Lecture Notes in Computer Science 4851

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison
Lancaster University, UK

Takeo Kanade
Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler
University of Surrey, Guildford, UK

Jon M. Kleinberg
Cornell University, Ithaca, NY, USA

Friedemann Mattern
ETH Zurich, Switzerland

John C. Mitchell
Stanford University, CA, USA

Moni Naor
Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz
University of Bern, Switzerland

C. Pandu Rangan
Indian Institute of Technology, Madras, India

Bernhard Steffen
University of Dortmund, Germany

Madhu Sudan
Massachusetts Institute of Technology, MA, USA

Demetri Terzopoulos
University of California, Los Angeles, CA, USA

Doug Tygar
University of California, Berkeley, CA, USA

Moshe Y. Vardi
Rice University, Houston, TX, USA

Gerhard Weikum
Max-Planck Institute of Computer Science, Saarbruecken, Germany

Serdar Boztaş Hsiao-Feng (Francis) Lu (Eds.)

Applied Algebra, Algebraic Algorithms and Error-Correcting Codes

17th International Symposium, AAECC-17
Bangalore, India, December 16-20, 2007
Proceedings

Springer

Volume Editors

Serdar Boztaş
RMIT University, School of Mathematical and Geospatial Sciences
GPO Box 2476V, Melbourne 3001, Australia
E-mail: serdar.boztas@ems.rmit.edu.au

Hsiao-Feng (Francis) Lu
National Chung-Cheng University, Department of Communications Engineering
168 University Rd., Min-Hsiung, Chia-Yi, Taiwan
E-mail: francis@ccu.edu.tw

Library of Congress Control Number: 2007940905

CR Subject Classification (1998): E.4, I.1, E.3, G.2, F.2

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN 0302-9743
ISBN-10 3-540-77223-5 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-77223-1 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media

springer.com

© Springer-Verlag Berlin Heidelberg 2007
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12202058 06/3180 5 4 3 2 1 0

Preface

The AAECC Symposia Series was started in 1983 by Alain Poli (Toulouse), who, together with R. Desq, D. Lazard and P. Camion, organized the first conference. Originally the acronym AAECC meant "Applied Algebra and Error-Correcting Codes." Over the years its meaning has shifted to "Applied Algebra, Algebraic Algorithms and Error-Correcting Codes," reflecting the growing importance of complexity, particularly for decoding algorithms. During the AAECC-12 symposium the conference committee decided to enforce the theory and practice of the coding side as well as the cryptographic aspects. Algebra was conserved, as in the past, but slightly more oriented to algebraic geometry codes, finite fields, complexity, polynomials, and graphs.

For AAECC-17 the main subjects covered were:

- Block codes, including list-decoding algorithms
- Algebra and codes: rings, fields, algebraic geometry codes
- Algebra: rings and fields, polynomials, permutations, lattices
- Cryptography: cryptanalysis and complexity
- Computational algebra: algebraic algorithms and transforms
- Sequences and boolean functions

Seven invited speakers characterize the aim of AAECC-17:

- Ralf Koetter, "Error Correction for Network Coding Channels"
- Tor Helleseth, "New Attacks on the Filter Generator"
- Tanja Lange, "Arithmetic on Edwards Curves"
- Gary McGuire, "Spectra of Boolean Functions, Subspaces of Matrices, and Going up Versus Going Down"
- Priti Shankar, "Algebraic Structure Theory of Tail-biting Trellises"
- Henning Stichtenoth, "Nice Codes from Nice Curves"
- Manindra Agrawal, "Determinant versus Permanent"

In addition, an Invited List Decoding Session was organized by Madhu Sudan:

- Venkatesan Guruswami, "List Decoding and Pseudorandom Constructions"
- Tom Høholdt, "Iterative List decoding of LDPC Codes"
- Ralf Koetter, "Optimizing Multivariate Interpolation"
- Atri Rudra, "Efficient List Decoding of Explicit Codes with Optimal Redundancy"

Except for AAECC-1 (56, 1985) and AAECC-7 (33, 1991), the proceedings of all the symposia have been published in Springer's (Vols. 228, 229, 307, 356, 357, 508, 539, 673, 948, 1255, 1719, 2227, 2643, 3857). It is a policy of AAECC to maintain a high scientific standard, comparable to that of a journal.

This was made possible thanks to the many referees involved. Each submitted paper was evaluated by at least two international researchers.

AAECC-17 received and refereed 61 submissions. Of these, 1 was withdrawn and 33 were selected for publication in these proceedings.

The symposium was organized by P. Vijay Kumar, Tom Høholdt, Heeralal Janwa, Serdar Boztaş and Hsiao-feng (Francis) Lu, with the help of Govindar Rangarajan, C.E. Veni Madhavan and Priti Shankar, under the Indian Institute of Science Mathematics Initiative (IMI). It was sponsored by the Department of Science and Technology, India; the Defence Research and Development Organization, India; and Microsoft Research India.

We express our thanks to the Springer staff, especially Alfred Hofmann, for their help in the preparation of these proceedings.

October 2007

Serdar Boztaş
Hsiao-Feng (Francis) Lu

Organization

Steering Commitee

Conference Co-chairs	P. Vijay Kumar (Univ. of Southern California, USA)
	Tom Høholdt (Technical Univ. of Denmark, Denmark)
	Heeralal Janwa (Univ. of Puerto Rico, Puerto Rico)
Program Co-chairs	Serdar Boztaş (RMIT Univ., Australia)
	Hsiao-feng (Francis) Lu (National Chung Cheng University, Taiwan)

Conference Committee

J. Calmet	K. Horadam	O. Moreno
G. Cohen	H. Imai	H. Niederreiter
G.L. Feng	H. Janwa	A. Poli
M. Giusti	R. Kohno	T.R.N. Rao
J. Heintz	H.W. Lenstra, Jr.	S. Sakata
T. Høholdt	S. Lin	P. Solé

Program Committee

I.F. Blake	J. Heintz	F. Özbudak
J. Calmet	K. Horadam	A. Poli
C. Carlet	H. Imai	S.S. Pradhan
G. Cohen	N. Kashyap	A. Rao
C. Ding	S. Lin	S. Sakata
G-L. Feng	O. Moreno	H-Y. Song
M. Giusti	W.H. Mow	P. Udaya
G. Gong	H. Niederreiter	C. Xing

Local Organizing Committee

Govindar Rangarajan C.E. Veni Madhavan Priti Shankar

Sponsoring Institutions

Department of Science and Technology, India
Defence Research and Development Organization, India
Microsoft Research India

Table of Contents

Invited Contributions

List Decoding and Pseudorandom Constructions 1

A Survey of Recent Attacks on the Filter Generator 7

Iterative List Decoding of LDPC Codes 18

Inverted Edwards Coordinates 20

Spectra of Boolean Functions, Subspaces of Matrices, and Going Up Versus Going Down 28

Efficient List Decoding of Explicit Codes with Optimal Redundancy 38

Algebraic Structure Theory of Tail-Biting Trellises 47

Nice Codes from Nice Curves 48

Regular Contributions

Generalized Sudan's List Decoding for Order Domain Codes 50

Bent Functions and Codes with Low Peak-to-Average Power Ratio for Multi-Code CDMA 60

Determining the Nonlinearity of a New Family of APN Functions 72

An Improvement of Tardos's Collusion-Secure Fingerprinting Codes with Very Short Lengths 80

Space-Time Codes from Crossed Product Algebras of Degree 4 90

On Non-randomness of the Permutation After RC4 Key Scheduling 100

Correctable Errors of Weight Half the Minimum Distance Plus One for the First-Order Reed-Muller Codes 110

Fault-Tolerant Finite Field Computation in the Public Key Cryptosystems 120

A Note on a Class of Quadratic Permutations over $\mathbb{F}_{2^n}$ 130

Constructions of Orthonormal Lattices and Quaternion Division Algebras for Totally Real Number Fields 138

Quaternary Plotkin Constructions and Quaternary Reed-Muller Codes 148

Joint Source-Cryptographic-Channel Coding Based on Linear Block Codes 158

On the Key-Privacy Issue of McEliece Public-Key Encryption 168

Lattices for Distributed Source Coding: Jointly Gaussian Sources and Reconstruction of a Linear Function 178

Linear Complexity and Autocorrelation of Prime Cube Sequences 188

The "Art of Trellis Decoding" Is NP-Hard 198

On the Structure of Inversive Pseudorandom Number Generators 208

Subcodes of Reed-Solomon Codes Suitable for Soft Decoding 217

Normalized Minimum Determinant Calculation for Multi-block and Asymmetric Space-Time Codes 227

On the Computation of Non-uniform Input for List Decoding on Bezerra-Garcia Tower 237

Dense MIMO Matrix Lattices—A Meeting Point for Class Field Theory and Invariant Theory 247

Secure Cross-Realm Client-to-Client Password-Based Authenticated Key Exchange Against Undetectable On-Line Dictionary Attacks 257

Links Between Discriminating and Identifying Codes in the Binary Hamming Space 267

Construction of Rotation Symmetric Boolean Functions on Odd Number of Variables with Maximum Algebraic Immunity 271

A Path to Hadamard Matrices 281

The Tangent FFT 291

Novel Algebraic Structure for Cyclic Codes 301

Distribution of Trace Values and Two-Weight, Self-orthogonal Codes over $GF(p, 2)$ 311

Generalized Rotation Symmetric and Dihedral Symmetric Boolean Functions – 9 Variable Boolean Functions with Nonlinearity 242 321

On Quasi-cyclic Codes over Integer Residue Rings 330

Extended Norm-Trace Codes with Optimized Correction Capability 337

On Generalized Hamming Weights and the Covering Radius of Linear Codes 347

Homomorphic Encryptions of Sums of Groups 357

Author Index 367

List Decoding and Pseudorandom Constructions

Venkatesan Guruswami*

Department of Computer Science & Engineering
University of Washington
Seattle, WA 98195
venkat@cs.washington.edu

There is a rich interplay between coding theory and computational complexity theory that has enriched both disciplines over the years. In particular, list decoding and closely related notions have been instrumental in several advances in explicit constructions of combinatorial objects with strong "random-like" properties, such as expander graphs, randomness extractors, and pseudorandom generators. Our aim here is to present

(i) a unified list-decoding-centric view of the definition of these objects, and

(ii) the details of recent work due to the author, C. Umans, and S. Vadhan [3], where this viewpoint yields powerful results, namely the construction of unbalanced bipartite graphs with very strong expansion properties based on the list-decodable codes due to Parvaresh and Vardy [4]. In turn these expanders yield simple constructions of randomness extractors that are optimal up to constant factors.

A List Decoding Lens on Pseudorandom Objects

We begin with a discussion of how a variety of central combinatorial objects in the theory of pseudorandomness can be captured by an appropriate list-decoding-like property. The list decoding viewpoint has been implicitly or explicitly used in several works over the years, for example [8,7,5,6,1], and most recently, is explicitly discussed in [3, Sec. 2.1].

For an integer $M \geqslant 1$, let $[M]$ denote the set $\{1, 2, \ldots, M\}$. A code $C \subseteq \Sigma^D$ with N codewords[1] with encoding function $E : [N] \to \Sigma^D$ can be naturally viewed as a map $\Gamma : [N] \times [D] \to [D] \times \Sigma$ as follows:

$$\Gamma(x, i) = (i, E(x)_i)$$

where $E(x)_i$ is the i'th symbol of the codeword corresponding to message x. In an equivalent graph view, we think of Γ as specifying the vertex neighborhoods in a bipartite graph with N vertices on the left each of degree D.

* Currently on leave at the School of Mathematics, Institute for Advanced Study, Princeton, NJ 08540. Supported by NSF Career Award CCF-0343672, NSF CCR-0324906, and a Packard Fellowship.

[1] We are using symbols that are non-standard in coding theory to be consistent with the typical choices in the target pseudorandom objects.

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 1–6, 2007.
© Springer-Verlag Berlin Heidelberg 2007

We say that a code $C \subseteq \Sigma^D$ is (t, L)-list-decodable if for all $r \in \Sigma^D$, the number of codewords of C that agree with r on at least t locations is at most L. Here $D - t$ represents the number of "errors" that can be list decoded with an output list size of L. An equivalent view of the (t, L)-list-decodability property is that for all subsets $T \subseteq [D] \times \Sigma$ of size D of the form $T = \{(i, r_i) \mid i \in [D]\}$, we have $|\mathsf{LIST}(T)| \leqslant L$ where

$$\mathsf{LIST}(T) \stackrel{\text{def}}{=} \{x \in [N] \mid \ \#\{i \mid \Gamma(x, i) \in T\} \geqslant t\} \ . \tag{1}$$

(In words, $\mathsf{LIST}(T)$ is the set of vertices on the left at least t of whose neighbors belong to T.)

Turning to expander graphs, we say that a bipartite graph $G = (V_L, V_R, E)$ is a (K, A)-expander if for all $S \subseteq V_L$ with $|S| \leqslant K$, the neighborhood of S, $N(S) = \{\Gamma(s, i) \mid s \in S, \ i \in [D]\}$, satisfies $|N(S)| \geqslant A|S|$. Here A is the expansion factor, which is clearly at most D. Expanders where $A = D(1 - \varepsilon)$ (here $\varepsilon > 0$ is a parameter that can be picked to be an arbitrarily small constant) are called lossless expanders. The equivalent "list decoding" based definition of (K, A)-expanders is the following: The graph defined by Γ is a (K, A)-expander iff for all $K' \leqslant K$ and $T \subseteq [D] \times \Sigma$ with $|T| < AK'$, we have $|\mathsf{LIST}(T)| < K'$ where

$$\mathsf{LIST}(T) \stackrel{\text{def}}{=} \{x \in [N] \mid \ \forall i \in [D], \ \Gamma(x, i) \in T\} \ . \tag{2}$$

(In words, $\mathsf{LIST}(T)$ is the set of vertices on the left all of whose neighbors belong to T.)

The map Γ is a (k, ε) extractor if for all $T \subseteq [D] \times \Sigma$, we have $|\mathsf{LIST}(T)| < 2^k$ where

$$\mathsf{LIST}(T) \stackrel{\text{def}}{=} \{x \in [N] \mid \ \mathbf{Pr}_{i \in [D]}[\Gamma(x, i) \in T] \geqslant \frac{|T|}{D|\Sigma|} + \varepsilon\} \ . \tag{3}$$

(In words, $\mathsf{LIST}(T)$ is the set of vertices on the left which have ε fraction more neighbors in T than the density of T.)

Note that unlike the case of codes and expanders, for extractors we require a small $\mathsf{LIST}(T)$ for all subsets T on the right. In turn this means that for sets S of size at least 2^k on the left (k is called the min-entropy of the source distribution on the left), the distribution on the right induced by taking a random neighbor of a random element of S is within distance ε from the uniform distribution.

If we are able to guarantee a small $\mathsf{LIST}(T)$ (as defined in (3)) only for sets of bounded size, then we get a weaker object called a randomness condenser. A condenser's output need not be close to uniform, but must be close to a distribution with good min-entropy. (For this to be non-trivial the right hand side must be much smaller than the left, and the name condenser refers to the fact that the min-entropy of the distribution on the left is condensed, perhaps with some small loss, into a distribution over the smaller universe on the right.) For a formal description of this connection, see [3], but roughly, the condition "If the input has min-entropy $\log(L/\varepsilon)$, then the output is ε-close to having min-entropy $\log(Q/\varepsilon)$," is implied by the following list decoding condition: For

all $T \subseteq [D] \times \Sigma$ with $|T| \leqslant Q$, we have $|\mathsf{LIST}(T)| \leqslant L$ where

$$\mathsf{LIST}(T) \stackrel{\text{def}}{=} \{x \in [N] \mid \mathbf{Pr}_{i \in [D]}[\Gamma(x,i) \in T] \geqslant \varepsilon\} \ . \tag{4}$$

(In words, $\mathsf{LIST}(T)$ is the set of vertices on the left a fraction ε of whose neighbors belong to T.)

Note that all the above objects are captured by similar definitions, of the form: For all sets of T that obey a certain property, a suitably defined $\mathsf{LIST}(T)$, which can viewed as the list decoding of T, has small size. For codes, the sets T are very small (of size D) with additional special structure; for expanders and condensers, the sets T of interest are arbitrary sets of certain size; while for extractors, we need a list decoding guarantee for all subsets T on the right. For list-decodable error-correcting codes, one usually also demands an efficient list decoding algorithm to compute $\mathsf{LIST}(T)$. For the other pseudorandom objects, the "decoding" occurs only in the analysis and a combinatorial bound on $\mathsf{LIST}(T)$ is all that is needed.

A generalization of list decoding called [illegible] has been very influential in several recent works. Under list recovering, the input to the decoder is a set R_i of at most ℓ possible values for the i'th symbol for each i, and the goal is to output all codewords whose i'th symbol belongs to R_i for at least α fraction of the positions i (and there should be at most L such codewords). List recovering serves as a crucial primitive in decoding concatenated codes — for example, the best known explicit binary list-decodable codes use a strong list recovering algorithm for an outer folded Reed-Solomon code [2]. List recovering can also be clearly captured in the above framework. In fact, the list recovering requirement is very similar to the condenser requirement. In the latter we only restrict the union of the R_i's to be small instead of stipulating that each of them be small. It is worth remarking that the algebraic list recovering algorithms such as for Reed-Solomon codes and folded RS codes work just as well when the union of the R_i's is small.

We stress that though all these objects can be uniformly captured in a list decoding like set-up, there are key differences in the parameters of interest in these objects. (As a result, often different techniques are required to optimize the parameters in each setting.) For example, in extractors we want D to be small (this corresponds to a small seed length) and $|\Sigma|$ to be large (this corresponds to outputting many nearly uniform bits). Clearly for codes we want the alphabet size $|\Sigma|$ to be small (constant or polynomial in the block length). As another example, for list-decodable codes, the exact size of $|\mathsf{LIST}(T)|$ is not too crucial, and generally any bound that is polynomial in the message length is sufficient. For the lossless expander construction in the next section, the exact relation between $|\mathsf{LIST}(T)|$ and $|T|$ is crucial; a factor 2 increase in the bound on list size (for T of the same size) would change the expansion factor A from the near-optimal $(1-\varepsilon)D$ to $D/2$.

Yet, the intuition and constructions from one setting have often lead to progress in constructing other objects. Trevisan's breakthrough extractors were based on an insightful use of pseudorandom generators and list-decodable

codes [8]. Ta-Shma and Zuckerman [6] gave a construction of codes with very good list-recoverability properties, albeit over very large alphabets, using the above view of the Trevisan extractors, along with an "algorithmic" version of the analysis used to bound $|\mathsf{LIST}(T)|$. In [1], a similar framework was applied to an extractor construction due to Ta-Shma, Zuckerman, and Safra [7] along with other ideas to give a list-decodable code better than RS codes for low rates. Shaltiel and Umans [5] used list-decodability of Reed-Muller codes to construct extractors, as well as their computational counterpart, pseudorandom generators. In fact the similarity of their extractor to the folded Reed-Solomon codes from [2] (which achieved the optimal trade-off between rate and list-decoding radius) was the inspiration for our research leading to a new algebraic construction of unbalanced expanders [3], which we discuss in the next section.

There are several more fruitful connections between list decoding and other pseudorandom objects. As the next section shows, sometimes the argument underlying the construction of a particular object (a list-decodable code in our case) can be ported to give non-trivial constructions of one of the related objects (lossless bipartite expanders in our case).

Lossless Expanders from Parvaresh-Vardy Codes

We begin with a description of the Parvaresh-Vardy codes [4]. There are several parameters in this construction: integers n, m, h, a finite field $\mathbb{F}_q$, and an irreducible polynomial $E(X)$ of degree n over $\mathbb{F}_q$. The messages of the code belong to $\mathbb{F}_q^n$ which is identified in the obvious way with polynomials of degree at most $(n-1)$ over $\mathbb{F}_q$. The codewords have q symbols, one corresponding to each element of $\mathbb{F}_q$. Each codeword symbol is an m-tuple of symbols over $\mathbb{F}_q$. The map $\Gamma : \mathbb{F}_q^n \times \mathbb{F}_q \to \mathbb{F}_q \times \mathbb{F}_q^m$ is given by:

$$\Gamma(f(X), \alpha) = (\alpha, f(\alpha), f_1(\alpha), \ldots, f_{m-1}(\alpha)) \tag{5}$$

where for $i = 1, 2, \ldots, m-1$, $f_i(X) = f(X)^{h^i} \bmod E(X)$. (Note that each $f_i(X)$ is also a polynomial of degree less than n.)

Viewing the above map Γ as defining a degree q bipartite graph G with q^n nodes on the left and q^{m+1} nodes on the right, the following expansion property of G is proved in [3].

Theorem 1. [illegible] G [illegible] $(h^m, q - nmh)$ [illegible]

We will soon sketch the idea behind the proof of the above theorem. But first we discuss the implications to randomness extraction. The left degree D of the expander equals q, and thus if $q \geqslant nmh/\varepsilon$, the expansion factor $A = q - nmh$ satisfies $A \geqslant (1-\varepsilon)D$. Since sets of size $K = h^m$ expand by nearly a factor of q, the right hand side must have at least qh^m vertices. By picking $q \simeq h^{1+\delta}$ for a small constant $\delta > 0$, the right hand side has only about $DK^{1+\delta}$ vertices. It is known that lossless expanders (which expand by a $(1-\varepsilon)D$ factor) are equivalent to condensers that lose no entropy. In the condenser view, the small

right hand side of our construction implies that the entropy rate of the output distribution on the right is $\simeq 1/(1+\delta)$ and thus very close to 1. Since all the min-entropy of the distribution on the left is preserved, the above expander reduces the task of constructing an extractor for arbitrary min-entropy to the much easier task of construction an extractor for entropy rate 99%. Together with a back-end extractor that works for such high entropy rates, we get an extractor that achieves the best known parameters. We refer the reader to [3] for the detailed statements about the final extractor construction.

We conclude the paper with a brief discussion of the proof of Theorem 1. Let $K = h^m$ and $A = q - nmh$. With the list decoding view, we need to prove that for any $T \subseteq \mathbb{F}_q^{m+1}$ with $|T| \leqslant AK - 1$, the set $\mathsf{LIST}(T)$ defined in (2) satisfies $|\mathsf{LIST}(T)| \leqslant K - 1$. (We actually need to prove this for any $K' \leqslant K$, but the proof for this case uses similar ideas.) The proof consists of three steps.

1. Since $|T| \leqslant AK - 1$, there must exist a non-zero $(m+1)$-variate polynomial $Q \in \mathbb{F}_q[X, Z_1, Z_2, \ldots, Z_m]$ of degree at most $(h-1)$ in each of the Z_i's and degree at most $(A-1)$ in X such that $Q(\mathbf{a}) = 0$ for all $\mathbf{a} \in T$. This is because there are Ah^m monomials $X^j Z_1^{i_1} \cdots Z_m^{i_m}$ that obey the imposed degree restrictions, and only $AK - 1$ homogeneous linear constraints on the coefficients of these monomials.

2. Any $f(X) \in \mathsf{LIST}(T)$ must satisfy $Q(X, f(X), f_1(X), \ldots, f_{m-1}(X)) = 0$. This is because if $f(X) \in \mathsf{LIST}(T)$, then for every $\alpha \in \mathbb{F}_q$,

$$Q(\alpha, f(\alpha), f_1(\alpha), \ldots, f_{m-1}(\alpha)) = 0 \ .$$

   The univariate polynomial $Q(X, f(X), f_1(X), \ldots, f_{m-1}(X))$ thus has at least q roots, but on the other hand its degree is at most $A - 1 + (n-1)m(h-1) < A + nmh = q$. It must thus be the zero polynomial.

3. This is the most important step where the specifics of the construction (the choice of the correlated polynomials $f_i(X)$) plays a critical role. Recalling the definition of $f_i(X) = f(X)^{h^i} \bmod E(X)$, and viewing the polynomials $f(X)$ and $f_i(X)$ as elements of the extension field $\Lambda = \mathbb{F}_q[X]/(E(X))$, we observe that each $f(X) \in \mathsf{LIST}(T)$ must be a root of the univariate polynomial $Q^* \in \Lambda[Y]$ defined as

$$Q^*(Y) \stackrel{\text{def}}{=} Q(X, Y, Y^h, Y^{h^2}, \ldots, Y^{h^{m-1}}) \bmod E(X) \ .$$

   It can be argued that $Q^*(Y)$ is a non-zero polynomial. Therefore, we can bound $|\mathsf{LIST}(T)|$ from above by the degree of Q^*. This degree is clearly at most

$$(h-1) + (h-1)h + (h-1)h^2 + \cdots + (h-1)h^{m-1} = h^m - 1 = K - 1 \ ,$$

   leading to the desired bound $|\mathsf{LIST}(T)| \leqslant K - 1$.

References

1. Guruswami, V.: Better Extractors for Better Codes? In: 36th Annual ACM Symposium on Theory of Computing, pp. 436–444 (2004)
2. Guruswami, V., Rudra, A.: Explicit Capacity-Achieving List-Decodable Codes. In: 38th Annual ACM Symposium on Theory of Computing, pp. 1–10 (2006)
3. Guruswami, V., Umans, C., Vadhan, S.: Unbalanced Expanders and Randomness Extractors from Parvaresh-Vardy Codes. In: 22nd IEEE Conference on Computational Complexity, pp. 96–108 (2007)
4. Parvaresh, F., Vardy, A.: Correcting Errors Beyond the Guruswami-Sudan Radius in Polynomial Time. In: 46th Annual IEEE Symposium on Foundations of Computer Science, pp. 285–294 (2005)
5. Shaltiel, R., Umans, C.: Simple Extractors for All Min-Entropies and a New Pseudorandom Generator. J. ACM 52(2), 172–216 (2005)
6. Ta-Shma, A., Zuckerman, D.: Extractor Codes. IEEE Trans. Inform. Theory 50(12), 3015–3025 (2004)
7. Ta-Shma, A., Zuckerman, D., Safra, S.: Extractors from Reed-Muller codes. In: 42nd Annual Symposium on Foundations of Computer Science, pp. 638–647 (2001)
8. Trevisan, L.: Extractors and Pseudorandom Generators. J. ACM 48(4), 860–879 (2001)

A Survey of Recent Attacks on the Filter Generator

Sondre Rønjom[1], Guang Gong[2], and Tor Helleseth[1]

[1] The Selmer Center,
Department of Informatics, University of Bergen, PB 7803 N-5020 Bergen, Norway
[2] Department of Electrical and Computer Engineering
University of Waterloo, Waterloo, Ontario N2L 3G1, Canada

Abstract. The filter generator consists of a linear feedback shift register (LFSR) and a Boolean filtering function that combines bits from the shift register to create a key stream. The nonlinear combiner generator employs several (LFSRs) and a Boolean function that combines bit from all the registers to generate the key stream. A new attack on the filter generator has recently been described by Rønjom and Helleseth who also extended the attack to linear feedback shift registers over an extension field $GF(2^m)$. Some extensions and improvements of the attacks to the filter generator have been given by Rønjom, Gong and Helleseth. The purpose of this paper is to give a short overview of these attacks and to discuss how to extend these attacks to the nonlinear combiner generator.

Keywords: Boolean function, filter generator, nonlinear combiner generator, m-sequences, stream ciphers.

1 Introduction

The binary filter generator is an important building block in many stream ciphers. The generator consists of a linear feedback shift register of length n that generates a maximal linear sequence $\{s_t\}$ (an m-sequence) of period $2^n - 1$ and a Boolean function of degree d that combines bits from the shift register and produces an output bit z_t at any time t. An illustration of the filter generator is shown in Figure 1.

The sequence $\{s_t\}$ obeys the recursion

$$\sum_{j=0}^{n} c_j s_{t+j} = 0, \quad c_j \in \{0, 1\}$$

where $c_0 = c_n = 1$. The characteristic polynomial $g(x) = \sum_{j=0}^{n} c_j x^j$, of the linear recursion, is a primitive polynomial of degree n and period $2^n - 1$. The zeros of $g(x)$ are α^{2^i} for $i = 0, 1, \ldots, n-1$, where α is a primitive element in $GF(2^n)$, the finite field with 2^n elements. The m-sequence can be written as

$$s_t = Tr_1^n(\beta \alpha^t) \tag{1}$$

where $\beta \in GF(2^n)$ and $Tr_1^n(x) = \sum_{i=0}^{n-1} x^{2^i}$.

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 7–17, 2007.
© Springer-Verlag Berlin Heidelberg 2007

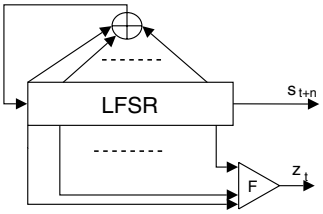


Fig. 1. Filter generator

The sequence $\{s_t\}$ is determined by the initial state $(s_0, s_1, \ldots, s_{n-1})$ and the characteristic polynomial $g(x)$. The 2^n sequences generated by $g(x)$, corresponding to the different initial states, form a vector space over $GF(2)$ denoted by $\Omega(g(x))$. For further information on linear shift registers the reader is referred to the recent book by Golomb and Gong [2]. By repeated use of the recursion we can write s_t as a linear combination of the n bits in the initial state. Thus, we have

$$s_t = \sum_{i=0}^{n-1} s_i l_{it} \tag{2}$$

for n binary sequences $\{l_{it}\}$ for $i = 0, 1, \ldots, n-1$. Note that each of these n sequences are nonzero and obey the same recursion as $\{s_t\}$ and thus are m-sequences.

At each time t, a keystream bit z_t is calculated as a function of certain bits in some positions $(e_0, e_1, \ldots, e_{m-1})$ in the LFSR state $(s_t, s_{t+1}, \ldots, s_{t+n-1})$ at time t using a Boolean polynomial function $f(x_0, x_1, \ldots, x_{m-1})$ of degree d in $m \leq n$ variables. The key stream is defined by $z_t = f(s_{t+e_0}, s_{t+e_1}, \ldots, s_{t+e_{m-1}})$. Since each s_t is determined by the initial state we will consider $f = f_0(s_0, s_1, \ldots, s_{n-1})$ as a polynomial of n variables $s_0, s_1, \ldots, s_{n-1}$.

By using the function $f(s_0, \ldots, s_{n-1})$ of degree d and expressing the sequence bits s_t as linear combinations of $s_0, s_1, \ldots, s_{n-1}$ given by (2), we define $f_t(s_0, s_1, \ldots, s_{n-1}) = f(s_t, s_{t+1}, \ldots, s_{t+n-1})$. This leads to the following system of nonlinear equations of degree d relating the n unknowns $s_0, s_1, \ldots, s_{n-1}$ to the keystream, $z_t = f_t(s_0, s_1, \ldots, s_{n-1})$ for $t = 0, 1, \ldots$ which has the initial state of the LFSR as a solution.

Various methods exist for solving the above nonlinear system of equations. The number of monomials of degree at most d over $GF(2)$ is given by the number $D = \sum_{i=1}^{d} \binom{n}{i}$. If D bits of the keystream is known, one may try to solve the above system directly using linear algebra techniques. General matrix reduction methods have complexity $O(D^\omega)$, where ω is commonly taken to be Strassen's reduction exponent $\log_2 7 \approx 2.807$. Thus for increasing n and d this approach becomes infeasable. If less keystream bits are known, one may instead try to experiment with Gröbner basis methods using for instance variations of Faugères algorithms, although the complexity then becomes more difficult to evalutate.

As described in Courtois and Meier [1], the keystream generator may be vulnerable to algebraic attacks even if the degree of the algebraic function is high. Let $S_t = (s_t, s_{t+1}, \ldots, s_{t+n-1})$ be the n-bit state of the linear shift register at time t and let $AN(f)$ denote the annihilator ideal of the Boolean function f in n variables, i.e.,

$$AN(f) = \{g \mid g(\mathbf{x})f(\mathbf{x}) = 0 \text{ for all } \mathbf{x} = (x_0, x_1, \ldots, x_{n-1})\}.$$

Thus any function g in $AN(f)$ leads to an equation of the form

$$f(S_t)g(S_t) = z_t g(S_t) = 0,$$

which when $z_t = 1$, implies that $g(S_t) = 0$. Similarly any function g' in $AN(1+f)$ leads to an equation of the form $g'(S_t) = 0$ whenever $z_t = 0$. This when the annihilators contain polynomials of small degree we may collect several equations of small degree and thus reduce the number of unknowns. Therefore one defines the algebraic immunity of the Boolean function f, $AI(f)$, as the smallest degree of a polynomial in $AN(f) \cup AN(1+f)$. It is therefore important to use Boolean functions with high algebraic immunity. Furthermore, is holds that $AI(f) \leq \lceil m/2 \rceil$ if m is the number of variables in f.

Most of the previous attacks on binary filter generators have considered the equation systems stemming from the filter generator as "random" or generic systems and have applied standard techniques for solving equations when analyzing the filter generator. The attacks described in the next sections utilize the structure of the finite field defined by the LFSR, and show that these systems are much simpler to solve than generic systems.

Since the paper contains an overview of some recent attacks several details will be omitted. The interested reader will find complete details including proofs and examples in [8], [7] and [10].

2 Attack Using Coefficient Sequences

For a subset $I = \{i_0, i_1, \ldots, i_{r-1}\}$ of $I_n = \{0, 1, \ldots, n-1\}$ define $s_I = s_{i_0} s_{i_1} \cdots s_{i_{r-1}}$. Let $K_{I,t}$ be the binary coefficient of s_I in the equation $z_t = f_t(s_0, s_1, \ldots, s_{n-1})$. Then we can represent the system of equations in a compact manner as

$$z_t = \sum_I s_I K_{I,t} \tag{3}$$

where the summation is taken over all subsets I of I_n. The binary sequence $\{K_{I,t}\}$ of coefficients of s_I is called the coefficient sequence. The main observation is that these sequences obey nice recursions so that when we add together equations according to these recursions we may remove the contribution of monomials of higher degree and arrive at a simple nonsingular system of n equations in n variables.

For simplicity, consider the contribution to the keystream from a the function consisting of a single monomial of degree r, say $f^* = x_{a_0} x_{a_1} \ldots x_{a_{r-1}}$ leading

to $z_t = s_{t+a_0}s_{t+a_1}\dots s_{t+a_{r-1}}$, where $0 \le a_0 < a_1 < \cdots < a_{r-1} < n$. Let $A = \{a_0, a_1, \dots, a_{r-1}\}$, then using (2) we obtain

$$z_t = s_{t+a_0}s_{t+a_1}\cdots s_{t+a_{r-1}} = \sum_I s_I K_{I,A,t}$$

where

$$K_{I,A,t} = \sum_{(i_0,i_1,\dots,i_{r-1}),I=\{i_0,i_1,\dots,i_{r-1}\}} l_{i_0,t+a_0}l_{i_1,t+a_1}\cdots l_{i_{r-1},t+a_{r-1}}. \tag{4}$$

The summation runs over all combinations of $i_0, i_1, \dots, i_{r-1}$ where the i_j's are in I_n and such that $I = \{i_0, i_1, \dots, i_{r-1}\}$.

The polynomial function f can in general be written as a sum of monomial terms as $f = \sum_A c_A x_A$. Note in particular that each subset A of I_n such that $|A| \ge |I|$ contributes to the coefficient sequence $\{K_{I,t}\}$. We therefore obtain

$$z_t = f(s_t, s_{t+1}, \dots, s_{t+n-1}) = \sum_I s_I K_{I,t}$$

where

$$K_{I,t} = \sum_{A,|A|\ge|I|} c_A K_{I,A,t}. \tag{5}$$

Lemma 1. *Let $wt(l)$ be the Hamming weight of the binary representation of l, and let*

$$g_q(x) = \prod_{l,wt(l)=q} (x + \alpha^l).$$

Let $|I| = k$ and let $\{K_{I,t}\}$ be the coefficient sequence corresponding to s_I for a Boolean function f of degree d. Then,

$$\{K_{I,t}\} \in \Omega(g_k(x)g_{k+1}(x)\cdots g_d(x)).$$

Proof (Sketch). The idea behind the proof is that from (4) it follows that $K_{A,I,t}$ is a linear combination of products of $r(\le d)$ shifted versions of the same m-sequence. Thus using (1) we get

$$K_{A,I,t} = \sum_{wt(J)\le d} b_J \alpha^{Jt}.$$

A detailed investigation shows that surprisingly $b_J = 0$ when $wt(J) < |I| = k$. Since $K_{I,t}$ is a linear combination of terms of the form $K_{A,I,t}$ the result holds for $K_{I,t}$. □

The main consequence of this lemma is that all coefficient sequences $\{K_{I,t}\}$, $|I| \ge 2$, for the nonlinear terms, obey the recursion with characteristic polynomial $p(x) = g_2(x)g_3(x)\cdots g_d(x)$. Thus using this recursion on the equation system leads to a linear system of n equations in n unknowns.

Algebraic Attack
1. Pre-compute $p(x) = g_2(x)g_3(x)\cdots g_d(x)$ and let $p(x) = \sum_{j=0}^{D-n} p_j x^j$.
2. Pre-compute the linear part of the equation system determined by the Boolean function $f(s_t, s_{t+1}, \ldots, s_{t+n-1}) = f_t(s_0, s_1, \ldots, s_{n-1})$ for $t = 0, 1, \ldots, D-1$. Compute the linear part of $f_0^* = \sum_{j=0}^{D-n} p_j f_j(s_0, s_1, \ldots, s_{n-1})$ from the linear parts of $f_j(s_0, s_1, \ldots, s_{n-1})$, and thereafter compute $f_1^*, f_2^*, \ldots, f_{n-1}^*$ (by increasing indices by 1 and replace s_n by its linear combination of $s_0, s_1, \ldots, s_{n-1}$).
3. For a given keystream z_t of D bits compute $z_t^* = \sum_{j=0}^{D-n} p_j z_{t+j}$. Determine the initial state (secret key) $(s_0, s_1, \ldots, s_{n-1})$ from the linear system of equations

$$z_t^* = f_t^*(s_0, s_1, \ldots, s_{n-1}) \quad \text{for } t = 0, 1, \ldots, n-1.$$

Note that if $f_0^* \neq 0$ then the coefficient matrix of the system will be nonsingular. This is due to the fact that the rows of the coefficient matrix can be considered to be n successive powers of α, where α is the primitive zero of the primitive polynomial $g(x)$ of degree n.

The best previous attacks have essentially been to reduce the problem to solve a nonlinear system of $D = \sum_{i=1}^{d} \binom{n}{i}$ equations in n unknowns, giving a complexity essentially $O(D^\omega)$ where $\omega = \log_2 7$. The new attack above provides an improved algorithm that breaks the filter generator in complexity $O(D)$ after a pre-computation of complexity $O(D(\log_2 D)^3)$ needed to find $p(x)$.

The case when $f_0^* = 0$ that, if the Boolean function is selected randomly, has a probability of about 2^{-n}. For $n = 128$ this is a small probability even though it is possible to compute such functions constructivly. In this case we need to modify the attack to avoid this (unlikely) problem. The modified attack do not need the properties of coordinate sequences. However, the overall complexity is essentially the same. The modifications are due to Rønjom, Gong and Helleseth and [7] and will be briefly described in Section 4.

Furthermore, the attack has been extended by Rønjom and Helleseth in [9] to special cases when the LFSR is over the extension field $GF(2^m)$.

3 More About the Coefficient Sequences

In this section we give a description of the nonlinear filter generator using a linear transformation. The coefficient sequences in the previous section are shown by Rønjom and Helleseth [10] to play a natural role in this linear transformation. The non-singular matrix

$$T_1 = \begin{pmatrix} 0 & 0 & \ldots & 0 & c_0 \\ 1 & 0 & \ldots & 0 & c_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & c_{n-1} \end{pmatrix},$$

is the companion matrix of $g(x)$ and also the characteristic polynomial of T_1 and thus it is known that

$$g(T_1) = T_1^n + c_{n-1}T_1^{n-1} + c_{n-2}T_1^{n-2} + \ldots + T_1^0 = 0.$$

Let $S_0 = (s_0, s_1, ..., s_{n-1})$ denote the initial state of the LFSR. Any state S_t at time t is found by taking appropriate powers of T_1 starting from the initial state

$$S_t = (s_t, s_{t+1}, \ldots, s_{t+n-1}) = (s_0, \ldots, s_{n-1})T_1^t,$$

and the consecutive states of the LFSR are

$$S_0, S_0T_1, S_0T_1^2, \ldots, S_0T_1^t, \ldots$$

which is an n-dimensional cyclic vector space.

Let $\hat{S}_t$ denote the vector with components s_{t+I} for $I \subset I_n$ in some ordering, say graded reverse lexicographic. We call $\hat{S}_t$ the (extended) state of the usual n-bit state $S_t = (s_t, s_{t+1}, \ldots, s_{t+n-1})$. We illustrate the definition with an example.

Example 1. Let $g_1(x) = g(x) = x^3 + x + 1$ be the generator polynomial for the LFSR. Then for $n = 3$ and $t = 0$ we have

$$\hat{S}_0 = (s_0, s_1, s_2, s_0s_1, s_0s_2, s_1s_2, s_0s_1s_2).$$

Using the linear recursion $s_{t+3} = s_{t+1} + s_t$ or $s_3 = s_1 + s_0$ we obtain the next (extended) state by increasing all indices by one. Thus the (extended) state at time $t = 1$ is

$$\hat{S}_1 = (s_1, s_2, s_3, s_1s_2, s_1s_3, s_2s_3, s_1s_2s_3).$$

Note that using the linear recursion of the LFSR each component in $\hat{S}_1$ is a linear combination of the components in $\hat{S}_0$. In this case we observe that the components in $\hat{S}_1$ not containing s_3 equals directly a component in $\hat{S}_0$, while the components involving s_3 can be written as

$$\begin{aligned} s_3 &= s_1 + s_0 \\ s_1s_3 &= s_1 + s_0s_1 \\ s_2s_3 &= s_0s_2 + s_1s_2 \\ s_1s_2s_3 &= s_0s_1s_2 + s_1s_2. \end{aligned}$$

Therefore the linear transformation that transforms $\hat{S}_0$ to $\hat{S}_1$ (or equivalently $\hat{S}_{t+1} = \hat{S}_tT$ for any integer t) can be described by the 7×7 matrix T given by

$$T = \begin{array}{c} \\ s_0 \\ s_1 \\ s_2 \\ s_0s_1 \\ s_0s_2 \\ s_1s_2 \\ s_0s_1s_2 \end{array} \begin{pmatrix} \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{array}{ccc} 4 & 5 & 6 \end{array} & \begin{array}{c} 7 \\ \\ \\ \hline 0 \\ 0 \\ 0 \\ \hline 1 \end{array} \\ \begin{array}{ccc} 0 & 0 & \mathbf{1} \\ \mathbf{1} & 0 & \mathbf{1} \\ 0 & \mathbf{1} & 0 \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 0 & 0 \end{array} & \begin{array}{ccc} 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 \\ \hline 0 & \mathbf{1} & 0 \\ 0 & 0 & \mathbf{1} \\ \mathbf{1} & 0 & \mathbf{1} \\ \hline 0 & 0 & 0 \end{array} & \begin{array}{c} 0 \\ 0 \\ 0 \\ \hline 0 \\ 0 \\ \mathbf{1} \\ \hline \mathbf{1} \end{array} \end{pmatrix}.$$

The columns are indexed as the rows but all indices are increased by one. For example the fifth column represents $s_1s_3 = s_1 + s_0s_1$.

For any subset $J = \{j_0, j_1, \ldots, j_{r-1}\} \subset I_n$, we define s_{t+J} to be $s_{t+J} = s_{t+j_0}s_{t+j_1} \cdots s_{t+j_{r-1}}$. The rows and columns are indexed by the subsets of I_n

and the value of T in position (I, J) is given by $K_{I,J,1}$, since this is the coefficient of s_I in s_{1+J}, i.e.,

$$s_{1+J} = \sum_I s_I K_{I,J,1}. \tag{6}$$

This matrix T also occurred in the paper by Hawkes and Rose [4] in their study of algebraic attacks. The $(2^n - 1) \times (2^n - 1)$ transformation matrix T given by $\hat{S}_{t+1} = \hat{S}_t T$ has more consequences for attacking the filter generator than anticipated in [4]. The interesting observation to be showed later is that the elements in the powers T^t of the matrix T are equal to the coefficient sequences $K_{I,J,t}$ defined by Rønjom and Helleseth in [8] as the coefficient of s_I in $s_{t+J} = s_{t+j_0} s_{t+j_1} \cdots s_{t+j_{r-1}}$ where $J = \{j_0, j_1, \ldots, j_{r-1}\}$, or in other words

$$s_{t+J} = \sum_I s_I K_{I,J,t}. \tag{7}$$

This is a consequence of the following theorem.

Theorem 1. *Let $T^t_{I,J}$ denote the element in row I and column J in T^t. Let $K_{I,J,t}$ be defined as the coefficient of s_I in the term s_{t+J}. Then $T^t_{I,J} = K_{I,J,t}$.*

Proof. The proof follows directly from (7) and $\hat{S}_t = \hat{S}_0 T^t$. □

Let v_f denote the binary vector of length $2^n - 1$ (we may assume without loss of generality there is no constant term in f) with component $v_{f,I}$ in position I being the coefficient of s_I in f, i.e., $f = \sum_I v_{f,I} s_I$. Then since, popular speacking, the effect of T is to increase the indices by one, this implies that the binary vector representation of $f_1(s_0, s_1, \ldots, s_{n-1}) = f_0(s_1, s_2, \ldots, s_n) (= f(s_1, s_2, \ldots, s_n))$ is related by

$$v_{f_1} = T v_{f_0}.$$

Therefore, in general each output bit z_t from the filter generator leads to the equation

$$z_t = \hat{S}_0 T^t v_{f_0} (= \hat{S}_0 v_{f_t}). \tag{8}$$

Let T_r be the submatrices along the diagonal of T, i.e., T_r equals T restricted to the $\binom{n}{r} \times \binom{n}{r}$ submatrix corresponding to the positions (I, J) where $|I| = |J| = r$. An interestinf property of T_r, proved in [10], is the following.

Theorem 2. *The minimal polynomial $m_{T_r}(x)$ and characteristic polynomial $c_{T_r}(x)$ of the square $\binom{n}{r} \times \binom{n}{r}$ matrix T_r are equal. Moreover, we have that*

$$c_{T_r}(x) = m_{T_r}(x) = g_r(x) = \prod_{e, wt(e)=r} (x + \alpha^e). \tag{9}$$

Consequently, we have that

$$m_T(x) = c_T(x) = \prod_{i=1}^{n} m_{T_i}(x) = g_1(x) g_2(x) \cdots g_n(x) = x^{2^n - 1} + 1.$$

Let v_f denote the length D support vector for a function $f(s_0, \ldots, s_{n-1})$ of degree d where the coefficients are ordered in the same order as the columns of T, and therefore in the same order as the expanded LFSR state S_t satisfying

$$\hat{S}_t T = \hat{S}_{t+1}, \hat{S}_t T^2 = \hat{S}_{t+2}, \ldots.$$

Since a keystream bit is given by $z_t = \hat{S}_t v_f$ and $z_{t+r} = \hat{S}_t T^r v_f = \hat{S}_{t+r} v_f$, a matrix relating sequence bits $s_t, \ldots s_{t+D-1}$ with keystream bits $z_t, \ldots, z_{t+D-1}$ is given by column vectors

$$A_t = \left(T^t v_f \middle| T^{t+1} v_f \middle| \ldots \middle| T^{t+D-1} v_f \right)$$

and thus $\hat{S}_0 A_t = \hat{S}_t A_0 = \hat{S}_0 T^t A_0 = [z_t, z_{t+1}, \ldots, z_{t+D-1}]$. The columns of the matrix A_t are the coefficient vectors of the functions studied in algebraic attacks.

Let as before $p(x) = g_2(x) g_3(x) \cdots g_d(x) = \sum_{j=0}^{D-n} p_j x^j$. The algebraic attack in the previous section can now be described by computing $f_t^* = \sum_{j=0}^{D-n} p_j f_{t+j}$ and

$$z_t^* = \sum_{j=0}^{D-n} p_j f_{t+j} = \sum_{j=0}^{D-n} p_j z_{t+j} = \hat{S}_0 T^t p_2(T) v_f = \hat{S}_0 v_{f_t^*},$$

where $v_{f_t^*} = T^t p(T) v_f$. Let $p(T) = T^{'}$, then

$$\hat{S}_0 p(T) A_t = \hat{S}_0 [v_{f_t^*}, T v_{f_t^*}, \ldots, T^{D-1} v_{f_t^*}] = [z_t^*, z_{t+1}^*, \ldots, z_{t+D-1}^*],$$

is a system of D linear equations. Note that $p(T)$ is only nonzero in the first n rows since $K_{I,J,t}$ is generated by $p(x)$ for any I when $|I| \geq 2$, due to the proofs of Lemma 1 and Theorem 1. Clearly, it therefore suffices to compute $v^{'} = p(T) v_f$ restricted to a length-n vector and then compute the columns of an $n \times n$ matrix given by $v^{'}, T_1 v^{'}, \ldots, T_1^{n-1} v^{'}$. Thus we have an system of n equations in the n unknown bits in the initial state $(s_0, s_1, \ldots, s_{n-1})$ which can therefore be determined.

4 Extending the Attack

Let $s_t = Tr_1^n(\beta \alpha^t)$ then we need to determine β to find the initial state of the LFSR used in the filter generator. We can write the bits in the key stream z_t in terms of its trace representation

$$z_t = \sum_k Tr_1^{m_k}(A_k (\beta \alpha^t)^k)$$

where the k's are (cyclotomic) coset leaders modulo $N = 2^n - 1$, and $m_k \,|\, n$ is the size of the coset $\{k, 2k, 2^2 k, \ldots\}$ mod N, which contains k. Here $wt(k) \leq d$ where d is the degree of the Boolean function f.

The main idea is to determine β directly from z_t. The attack in Rønjom and Helleseth [8] applied the shift operator to the key stream z_t using the polynomial $p(x) = g_2(x)g_3(x)\cdots g_d(x)$ with all zeros α^J of weight $2 \leq wt(J) \leq d$, leading to

$$p(E)z_t = \sum_{j=0}^{D-n} p_j z_{t+j} = \sum_k Tr_1^{m_k}(A_k\beta^k p(\alpha^k)\alpha^{tk}) = Tr_1^n(A_1\beta p(\alpha)\alpha^t).$$

The left hand side is linear in the bits in the initial state and thus leads to a linear equation system which is considered in Rønjom and Helleseth [8]. Furthermore, A_1 was explicitly given in [8].

In the case when $A_1 = 0$ we select another k such that $A_k \neq 0$ and $gcd(k, 2^n - 1) = 1$ and let instead $p(x)$ be defined to have all possible zeros α^J where $1 \leq wt(J) \leq d$, except for α^k. Then using the shift operator for this $p(x)$, we get

$$p(E)z_t = Tr_1^n(A_k\beta^k p(\alpha^k)\alpha^{tk}).$$

The aim is to calculate β. This is done in two steps.

Step 1. In the first step we determine $r = A_k\beta^k p(\alpha^k)$ from

$$u_t = p(E)z_t = Tr_1^n(r\alpha^{tk}) \text{ for } t = 0, 1, \ldots, n-1.$$

This is a linear equation system with n equations in n unknowns $x_i = r^{2^i}$ for $i = 0, 1, \ldots, n-1$. Since the coefficient matrix is a Vandermonde matrix and $(u_0, u_1, \ldots, u_{n-1})$ is known this gives us $r = A_k\beta^k p_k(\alpha^k)$ and therefore

$$\beta^k = rA_k^{-1}[p_k(\alpha^k)]^{-1}$$

where r and $p_k(\alpha^k)$ are known. Thus it remains to determine A_k.

Step 2. The second step is to find A_k. Note that $\{A_k\}$ is related to a discrete Fourier transform of $\{z_t\}$, which can be computed through expansion of z_t. An explicit formulae for A_k is given in Gong [3] or derived from results in [5] and [6]. For further details including a detailed example the reader is referred to [7].

Step 3. Compute the initial state by $s_t = Tr_1^n(\beta\alpha^t)$ for $t = 0, 1, \ldots, n-1$.

The complexity of this attack is asymptotically essentially the same as in [8], but also works in the case when $A_1 = 0$ (or equivalently $f_0^* = 0$) which needed some modifications in the original attack.

5 Attacking the Combiner Generator

The combiner generator uses several LFSRs, each generating a different m-sequence. The output from the different LFSRs are combined by a Boolean function to produce a key stream bit z_t. Usually one bit is taken from each register and a Boolean function f combines these bits to a key stream bit. The methods for analyzing the filter generator can be extended rather directly to the combiner case with minor changes. In this section we discuss this briefly.

For the filter generator the key stream z_t can be represented as

$$z_t = \sum_i \beta_i \alpha_i^t,$$

where α_i is a product of $\leq d(= deg(f))$ zeros from the LFSR. Thus the zeros are of the form α^J where the Hamming weight of the binary representation of J is at most d. The reason is that z_t is a sum of products of $\leq d$ products of shifted version of the same m-sequence.

In the linear combiner case the key stream can be represented similarly but now each α_i is a product of zeros from the characteristic polynomials of the different shift registers. For example if we have three LFSRs generating m-sequences, $\{a_t\}$, $\{b_t\}$ and $\{c_t\}$ and $f = x_1x_2x_3 + x_1$, and we select $x_1 = a_t$, $x_2 = b_t$ and $x_3 = c_t$, then the keystream can written

$$z_t = \sum_i \beta_i \alpha_i^t,$$

where each α_i is either a product of three elements being one zero from each of the characteristic polynomials or a zero from the characteristic polynomials generating the $\{a_t\}$ sequence.

In this case (when one variable enters linearly) we can define $p(x)$ to contain all these zeros except the zeros from the characteristic polynomial of $\{a_t\}$. Then we have

$$p(E)z_t = \sum_i \beta_i p(\alpha_i)\alpha_i^t = Tr_1^{n_a}(\beta_1 p(\alpha_1)\alpha_1^t)$$

where α_1 is a zero of the polynomial $a(x)$ of degree n_a. We can use methods similar to previous section to determine the initial state of $\{a_t\}$

We will consider the slightly more general case when the Boolean function may take more than one bit from each LFSR.

Example 2. Let $a(x) = x^4 + x + 1$ and $b(x) = x^5 + x^2 + 1$ be two LFSRs. Let $\{a_t\}$ and $\{b_t\}$ denote the m-sequences generated by the characteristic polynomials $a(x)$ and $b(x)$ respectively. Let f be the filter function

$$f(a_t, b_t) = f_t(a_0, a_1, a_2, a_3, b_0, b_1, b_2, b_3, b_4) = z_t.$$

For the $n = 9$ unknown variables:

$$(a_0, a_1, a_2, a_3, b_0, b_1, b_2, b_3, b_4) = (x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8).$$

Let $f(x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8) = x_0 + x_5 + x_0x_5 + x_1x_3 + x_5x_6$.

Then

$$\begin{aligned} z_t &= f(x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8) \\ &= f(a_t, a_{t+1}, a_{t+2}, a_{t+3}, b_t, b_{t+1}, b_{t+2}, b_{t+3}, b_{t+4}) \\ &= a_t + b_t + a_tb_t + a_{t+1}a_{t+2} + b_tb_{t+1}. \end{aligned}$$

In this case we can study the coordinate sequences for the polynomials f_t as a function of the coordinate sequences for a_I and b_J. The methods in the previous sections apply with minor adjustments.

Acknowledgements

This work was supported by the Norwegian Research Council.

References

1. Courtois, N., Meier, W.: Algebraic Attacks on Stream Ciphers with Linear Feedback. In: Biham, E. (ed.) EUROCRPYT 2003. LNCS, vol. 2656, pp. 345–359. Springer, Heidelberg (2003)
2. Golomb, S.W., Gong, G.: Signal Design for Good Correlation: For Wireless Communication, Cryptography and Radar. Cambridge University Press, Cambridge (2005)
3. Gong, G.: Analysis and Synthesis of Phases and Linear Complexity of Non-Linear Feedforward Sequences. Ph.D. thesis, University of Elec. Sci. and Tech. of China (1990)
4. Hawkes, P., Rose, G.: Rewriting Variables: The Complexity of Fast Algebraic Attacks on Stream Ciphers. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 390–406. Springer, Heidelberg (2004)
5. Herlestam, T.: On Functions of Linear Shift Register Sequences. In: Pichler, F. (ed.) EUROCRYPT 1985. LNCS, vol. 219, pp. 119–129. Springer, Heidelberg (1986)
6. Paterson, K.G.: Root Counting, the DFT and the Linear Complexity of Nonlinear Filtering. Codes and Cryptography 14, 247–259 (1998)
7. Rønjom, S., Gong, G., Helleseth, T.: On Attacks on Filtering Generators Using Linear Subspace Structures. In: SSC 2007, pp. 141–153 (2007)
8. Rønjom, S., Helleseth, T.: A New Attack on the Filter Generator. IEEE Trans. Inform. Theory 53(5), 1752–1758 (2007)
9. Rønjom, S., Helleseth, T.: Attacking the Filter Generator over $GF(2^m)$. In: WAIFI 2007. LNCS, vol. 4547, Springer, Heidelberg (2007)
10. Rønjom, S., Helleseth, T.: The Linear Vector Space Spanned by the Nonlinear Filter Generator. In: SSC 2007, pp. 141–153 (2007)
11. Rueppel, R.A.: Analysis and Design of Stream Ciphers. Springer, Heidelberg (1986)

Iterative List Decoding of LDPC Codes

Tom Høholdt[1] and Jørn Justesen[2]

[1] Department of Mathematics, The Technical University of Denmark Bldg.303
DK-2800 Lyngby, Denmark
T.Hoeholdt@mat.dtu.dk
[2] COM, The technical University of Denmark Bldg 343 DK-2800 Lyngby, Denmark
jju@com.dtu.dk

1 Extended Abstract

In the last decade two old methods for decoding linear block codes have gained considerable interest, *iterative decoding* as first described by Gallager in [1] and *list decoding* as introduced by Elias [2]. In particular iterative decoding of low-density parity-check (LDPC) codes, has been an important subject of research, see e.g. [3] and the references therein. "Good" LDPC codes are often randomly generated by computer, but recently codes with an algebraic or geometric structure have also been considered e.g [3] and [4]. The performance of the iterative decoder is typically studied by simulations and a theoretical analysis is more difficult.

In this paper we combine the two decoding methods and present an iterative list decoding algorithm. In particular we apply this decoder to a class of LDPC codes from finite geometries and show that the $(73, 45, 10)$ projective geometry code can be maximum likelihood decoded with low complexity. Moreover the list decoding approach enables us to give a complete analysis of the performance in this case. We also discuss the performance of the list bit-flipping algorithm for longer LDPC codes.

We consider hard-decision iterative decoding of a binary (n, k, d) code. For a received vector, y, we calculate an extended syndrome $s = Hy'$, where H is a parity check matrix, but usually has more than $n - k$ rows. Let r denote the length of the syndrome. The idea of using extended syndromes was also used in [5]. Our approach is based on one of the common versions of bit flipping (BF) [3], where the schedule is such that the syndrome is updated after each flip. In each step we flip a symbol chosen among those positions that reduce the weight of the extended syndrome, which we refer to briefly as the syndrome weight, u. A decoded word is reached when $u = 0$. In this paper we consider a variation of the common algorithm in the form of a tree-structured search. Whenever there is a choice between several bits, all possibilities are tried in succession. The result of the decoding algorithm is, in general, a list of codewords, obtained as leaves of the search tree. This form of the bit flipping algorithm leads naturally to a solution in the form of a list of codewords at the same smallest distance from y [6]. This list decoding concept is somewhat different from list decoding in the usual sense of all codewords within a certain distance from y. The paper is a continuation of [7] including results on long codes from [8].

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 18–19, 2007.
© Springer-Verlag Berlin Heidelberg 2007

References

1. Gallager, R.G.: Low-Density Parity-Check Codes. M.I.T. Press, Cambridge, MA (1963)
2. Elias, P.: List Decoding for Noisy Channel. Res. Lab. Electron., MIT, Cambridge, MA, Techn. Rep. 335 (1957)
3. Kou, Y., Lin, S., Fossorier, M.: Low-Density Parity-Check Codes Based on Finite Geometries: A Rediscovery and New Results. IEEE Trans. Inform. Theory 47, 2711–2736 (2001)
4. Liu, Z., Pados, D.A.: LDPC Codes from Generalized Polygons. IEEE Trans. Inform. Theory 51, 3890–3898 (2005)
5. Bossert, M., Hergert, F.: Hard-and Soft-Decision Decoding Beyond the Half Minimum Distance - An Algorithm for Linear Codes. IEEE Trans. Inform. Theory 32, 709–714 (1986)
6. Hjaltason, J.: List Decoding of LDPC Codes. M. Eng. Thesis, Department of Mathematics, Technical University of Denmark (2005)
7. Justesen, J., Høholdt, T., Hjaltason, J.: Iterative List Decoding of Some LDPC Codes. IEEE Trans. Inform. Theory. (to appear, 2007)
8. Kristensen, J.T.: List Decoding of LDPC Codes. M. Eng. Thesis, COM, Technical University of Denmark (2007)

Inverted Edwards Coordinates

Daniel J. Bernstein[1] and Tanja Lange[2,⋆]

[1] Department of Mathematics, Statistics, and Computer Science (M/C 249)
University of Illinois at Chicago, Chicago, IL 60607–7045, USA
djb@cr.yp.to
[2] Department of Mathematics and Computer Science
Technische Universiteit Eindhoven, P.O. Box 513, 5600 MB Eindhoven, Netherlands
tanja@hyperelliptic.org

Abstract. Edwards curves have attracted great interest for several reasons. When curve parameters are chosen properly, the addition formulas use only $10\mathbf{M} + 1\mathbf{S}$. The formulas are *strongly unified*, i.e., work without change for doublings; even better, they are *complete*, i.e., work without change for all inputs. Dedicated doubling formulas use only $3\mathbf{M} + 4\mathbf{S}$, and dedicated tripling formulas use only $9\mathbf{M} + 4\mathbf{S}$.

This paper introduces *inverted Edwards coordinates*. Inverted Edwards coordinates $(X_1 : Y_1 : Z_1)$ represent the affine point $(Z_1/X_1, Z_1/Y_1)$ on an Edwards curve; for comparison, standard Edwards coordinates $(X_1 : Y_1 : Z_1)$ represent the affine point $(X_1/Z_1, Y_1/Z_1)$.

This paper presents addition formulas for inverted Edwards coordinates using only $9\mathbf{M} + 1\mathbf{S}$. The formulas are not complete but still are strongly unified. Dedicated doubling formulas use only $3\mathbf{M} + 4\mathbf{S}$, and dedicated tripling formulas use only $9\mathbf{M} + 4\mathbf{S}$. Inverted Edwards coordinates thus save $1\mathbf{M}$ for each addition, without slowing down doubling or tripling.

Keywords: Elliptic curves, addition, doubling, explicit formulas, Edwards coordinates, inverted Edwards coordinates, side-channel countermeasures, unified addition formulas, strongly unified addition formulas.

1 Introduction

In [8] Edwards proposed a new normal form for elliptic curves and gave an addition law that is remarkably symmetric in the x and y coordinates. In [4], using coordinates $(X : Y : Z)$ to represent the point $(X/Z, Y/Z)$ on an Edwards curve, we showed that curve addition could be performed using only $10\mathbf{M} + 1\mathbf{S}$ (i.e., 11 field multiplications, of which 1 is a squaring) and that curve doubling could be performed using only $3\mathbf{M} + 4\mathbf{S}$. We presented a comprehensive survey

⋆ Permanent ID of this document: 0ef034ea1cdbb58a5182aaaefbea6754. Date of this document: 2007.10.03. This work has been supported in part by the European Commission through the IST Programme under Contract IST–2002–507932 ECRYPT. This work was carried out while the first author was visiting Technische Universiteit Eindhoven.

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 20–27, 2007.
© Springer-Verlag Berlin Heidelberg 2007

of speeds of our formulas and previous formulas for elliptic-curve arithmetic in various representations. The survey showed that Edwards curves provide the fastest additions and almost the fastest doublings. The only faster doublings were from doubling-oriented Doche/Icart/Kohel curves, which come with rather inefficient addition formulas.

One of the attractive features of the Edwards addition law is that it is *strongly unified*: the addition law works without change for doublings. We showed in [4] that, when curve parameters are chosen properly, the addition law is even *complete*: it works for all inputs, with no exceptional cases. Our fast addition formulas in [4] have the same features. See Section 2 of this paper for a more detailed review of Edwards curves.

In [2], together with Birkner and Peters, we showed that tripling on Edwards curves could be performed using only $9\mathbf{M} + 4\mathbf{S}$. We also analyzed the optimal combinations of additions, doublings, triplings, windowing methods, on-the-fly precomputations, curve shapes, and curve formulas, improving upon the analysis in [6] by Doche and Imbert. Hisil, Carter, and Dawson independently developed essentially the same tripling formulas; see [9].

New Contributions. This paper presents an even faster coordinate system for elliptic curves: namely, *inverted Edwards coordinates*, using coordinates $(X : Y : Z)$ to represent the point $(Z/X, Z/Y)$ on an Edwards curve. In Section 4 we present formulas for curve addition in inverted Edwards coordinates using only $9\mathbf{M} + 1\mathbf{S}$, saving $1\mathbf{M}$ compared to standard Edwards coordinates.

Inverted Edwards coordinates, unlike standard Edwards coordinates, do not have complete addition formulas: some points, such as the neutral element, must be handled separately. But our addition formulas still have the advantage of strong unification: they can be used without change to double a point.

In Sections 5 and 6 we present formulas for doubling and tripling in inverted Edwards coordinates using only $3\mathbf{M} + 4\mathbf{S}$ and $9\mathbf{M} + 4\mathbf{S}$, matching the speeds of standard Edwards coordinates.

All of the operation counts stated above assume small curve parameters and disregard the cost of multiplying by a curve parameter. Arbitrary curve parameters cost $1\mathbf{M}$ extra for each addition, each doubling, and each tripling. The penalty for standard Edwards coordinates is smaller: arbitrary curve parameters cost $1\mathbf{M}$ extra for addition but nothing for doubling or tripling.

In Section 7 we revisit the comparison from [4], analyzing the impact of inverted Edwards coordinates and other recent speedups.

2 Review of Edwards Curves

Let k be a field. Throughout this paper we assume that $2 \neq 0$ in k.

A curve in *Edwards form* is given by an equation

$$x^2 + y^2 = 1 + dx^2y^2,$$

where $d \notin \{0, 1\}$. Every Edwards curve is birationally equivalent to an elliptic curve in Weierstrass form. See [4, Section 3] for an explicit description of the equivalence.

One reason for the great interest in Edwards curves is that the Edwards addition law

$$(x_1, y_1), (x_2, y_2) \mapsto \left(\frac{x_1 y_2 + y_1 x_2}{1 + d x_1 x_2 y_1 y_2}, \frac{y_1 y_2 - x_1 x_2}{1 - d x_1 x_2 y_1 y_2} \right)$$

is *strongly unified*: it applies to doubling as well as to general addition, unlike the usual Weierstrass addition law. Strongly unified addition formulas had previously been published for Jacobi intersections, Jacobi quartics, and Weierstrass curves in projective coordinates, but the Edwards formulas are considerably faster.

We showed in [4, Theorem 3.3] that if d is not a square in k then the Edwards addition law has an even more attractive feature: it is *complete*. This means that there are no points $(x_1, y_1), (x_2, y_2)$ on the curve where the denominators vanish; the Edwards addition law produces the correct output for *every* pair of input points. The neutral element $(0, 1)$ does not cause any trouble. The Edwards curve has two singularities at infinity, corresponding to four points on the desingularization of the curve; but those four points are defined over $k(\sqrt{d})$, not over k.

To the best of our knowledge, the Edwards addition law is the only complete addition law stated in the literature. Previous addition laws have exceptional cases and require careful handling by the implementor to avoid the risk of incorrect results and to avoid the risk of leaking secret information through side channels. It should be possible to build a complete addition law for some Weierstrass curves starting from the formulas in [5], but we would not expect the resulting law to be nearly as fast as the Edwards addition law.

In [4] we suggested using homogeneous coordinates $(X_1 : Y_1 : Z_1)$, where $(X_1^2 + Y_1^2)Z_1^2 = Z_1^4 + dX_1^2 Y_1^2$ and $Z_1 \neq 0$, to represent the point $(X_1/Z_1, Y_1/Z_1)$ on the Edwards curve. Here $(X_1 : Y_1 : Z_1) = (\lambda X_1 : \lambda Y_1 : \lambda Z_1)$ for any $\lambda \neq 0$. In [4, Section 4] we presented explicit formulas for addition in this representation using $10\mathbf{M} + 1\mathbf{S} + 1\mathbf{D} + 7\mathbf{a}$, where $\mathbf{M}$ denotes the cost of a field multiplication, $\mathbf{S}$ the cost of a field squaring, $\mathbf{D}$ the cost of a multiplication by the curve parameter d, and $\mathbf{a}$ the cost of a field addition.

Implementations can gain speed, at the expense of simplicity, by using dedicated doubling formulas for additions where the inputs are known to be equal. In [4, Section 4] we presented explicit doubling formulas using $3\mathbf{M} + 4\mathbf{S} + 6\mathbf{a}$. Completeness remains beneficial in this situation: one does not need to check for other exceptions if the curve parameter d is not a square.

3 Inverted Edwards Coordinates

In this and the following sections we consider a different representation of points on an Edwards curve $x^2 + y^2 = 1 + dx^2 y^2$. We use three coordinates $(X_1 : Y_1 : Z_1)$, where

$$(X_1^2 + Y_1^2)Z_1^2 = X_1^2 Y_1^2 + dZ_1^4$$

and $X_1Y_1Z_1 \neq 0$, to represent the point $(Z_1/X_1, Z_1/Y_1)$ on the Edwards curve. We refer to these coordinates as *inverted Edwards coordinates*. As before, $(X_1 : Y_1 : Z_1) = (\lambda X_1 : \lambda Y_1 : \lambda Z_1)$ for any $\lambda \neq 0$.

It is easy to convert from standard Edwards coordinates $(X_1 : Y_1 : Z_1)$ to inverted Edwards coordinates: simply compute $(Y_1Z_1 : X_1Z_1 : X_1Y_1)$ with three multiplications. The same computation also performs the opposite conversion from inverted Edwards coordinates to standard Edwards coordinates.

For computations we use the vector (X_1, Y_1, Z_1) to represent the point $(X_1 : Y_1 : Z_1)$ in inverted Edwards coordinates.

Special points. The requirement $X_1Y_1Z_1 \neq 0$ means that inverted Edwards coordinates cannot represent points (x_1, y_1) on the Edwards curve that satisfy $x_1y_1 = 0$. There are four such points: the neutral element $(0, 1)$, the point $(0, -1)$ of order 2, and the points $(\pm 1, 0)$ of order 4. Additions that involve these points as inputs or outputs must be handled by separate routines.

The four points $(0, 1), (0, -1), (1, 0), (-1, 0)$ are $(0 : 1 : 1), (0 : -1 : 1)$, $(1 : 0 : 1), (-1 : 0 : 1)$ in standard Edwards coordinates. Applying the aforementioned conversion to inverted Edwards coordinates, and ignoring the requirement $X_1Y_1Z_1 \neq 0$, produces points at infinity on the projective curve $(X^2 + Y^2)Z^2 = X^2Y^2 + dZ^4$: specifically, $(1 : 0 : 0), (-1 : 0 : 0), (0 : 1 : 0), (0 : -1 : 0)$. But then the rule $(X_1 : Y_1 : Z_1) = (\lambda X_1 : \lambda Y_1 : \lambda Z_1)$ equates $(1 : 0 : 0)$ with $(-1 : 0 : 0)$, losing the distinction between $(0, 1)$ and $(0, -1)$, and similarly losing the distinction between $(1, 0)$ and $(-1, 0)$.

To have unique representations for the computations it is convenient to use the vectors $(1, 0, 0), (-1, 0, 0), (0, -1, 0), (0, 1, 0)$ to represent $(0, 1), (0, -1), (1, 0)$, $(-1, 0)$. Note that these representations are *not* homogeneous and that for algorithmic reasons $(\pm 1, 0)$ correspond to $(0, \mp 1, 0)$. One must be careful to check for $Z_1 = 0$ before adding $(X_1 : Y_1 : Z_1)$ to another point, and to check for $X_1Y_1 = 0$ before applying the conversions to and from standard Edwards coordinates.

In many applications one restricts attention to a subgroup of odd order, so the only special point is the neutral element and fewer checks are required. One can also randomize computations so that special points have a negligible chance of occurring; see [4, Section 8] for pointers to the literature.

Geometry. Recall that the desingularization of an Edwards curve has, over $k(\sqrt{d})$, four points that map to the two singularities at infinity on the curve. It also has four points that map without ramification to $(0, 1)$, $(0, -1)$, $(1, 0)$, and $(-1, 0)$.

Mapping the same desingularization to the projective curve $(X^2 + Y^2)Z^2 = X^2Y^2 + dZ^4$ takes the first four points without ramification to $(0 : \pm\sqrt{d} : 1)$ and $(\pm\sqrt{d} : 0 : 1)$, and takes the second four points to two singularities at infinity.

When d is not a square, the first map has no ramification points over k and allows a complete addition law on the Edwards curve. The second map always has ramification points, and in particular is ramified at the neutral element.

For mathematicians it is perhaps more satisfying to start from the projective curve $(X^2 + Y^2)Z^2 = X^2Y^2 + dZ^4$ and define an addition law on it, including

the points $(0 : \pm\sqrt{d} : 1)$ and $(\pm\sqrt{d} : 0 : 1)$, without mapping to an Edwards curve. We restricted to points $(X_1 : Y_1 : Z_1)$ with $X_1Y_1Z_1 \neq 0$ to maintain the link with Edwards curves and the Edwards addition law.

4 Addition

Obtaining more efficient addition formulas was our main goal in investigating inverted Edwards coordinates. Inspecting the addition formulas in [4, Section 4] one notices that the computations of the resulting X_3 and Y_3 each involve a multiplication by Z_1Z_2.

Inserting Z_i/X_i for x_i and Z_i/Y_i for y_i in the Edwards addition law (assuming $X_iY_iZ_i \neq 0$) we obtain

$$\left(\frac{Z_1}{X_1},\frac{Z_1}{Y_1}\right)+\left(\frac{Z_2}{X_2},\frac{Z_2}{Y_2}\right)=\left(\frac{(X_2Y_1+X_1Y_2)Z_1Z_2}{X_1X_2Y_1Y_2+dZ_1^2Z_2^2},\frac{(X_1X_2-Y_1Y_2)Z_1Z_2}{X_1X_2Y_1Y_2-dZ_1^2Z_2^2}\right)=\left(\frac{Z_3}{X_3},\frac{Z_3}{Y_3}\right)$$

where

$$\begin{aligned} X_3 &= (X_1X_2 - Y_1Y_2)(X_1X_2Y_1Y_2 + dZ_1^2Z_2^2) \\ Y_3 &= (X_2Y_1 + X_1Y_2)(X_1X_2Y_1Y_2 - dZ_1^2Z_2^2) \\ Z_3 &= (X_1X_2 - Y_1Y_2)(X_2Y_1 + X_1Y_2)Z_1Z_2. \end{aligned}$$

This shows the idea behind inverted Edwards coordinates, namely that in this representation only Z_3 needs to be multiplied with Z_1Z_2, which saves $1\mathbf{M}$ in total. Compared to the addition in Edwards coordinates the degree of these formulas is only 6 as opposed to 8 in that representation.

We then eliminate multiplications from these formulas, as in [4, Section 4], obtaining the following formulas to compute the sum $(X_3 : Y_3 : Z_3) = (X_1 : Y_1 : Z_1) + (X_2 : Y_2 : Z_2)$ in inverted Edwards coordinates, given $(X_1 : Y_1 : Z_1)$ and $(X_2 : Y_2 : Z_2)$:

$$\begin{aligned} &A = Z_1 \cdot Z_2;\ B = dA^2;\ C = X_1 \cdot X_2;\ D = Y_1 \cdot Y_2;\ E = C \cdot D; \\ &H = C - D; I = (X_1 + Y_1) \cdot (X_2 + Y_2) - C - D; \\ &X_3 = (E + B) \cdot H; Y_3 = (E - B) \cdot I;\ Z_3 = A \cdot H \cdot I. \end{aligned}$$

One readily counts $9\mathbf{M} + 1\mathbf{S} + 1\mathbf{D} + 7\mathbf{a}$, as advertised in the introduction. We have added these formulas to the EFD [3] for formal verification that the results coincide with the original Edwards addition law and that the formulas are strongly unified.

Restricted additions. *Mixed addition* means that Z_2 is known to be 1. There is an obvious saving of $1\mathbf{M}$ in this case since $A = Z_1 \cdot Z_2 = Z_1$, leading to a total cost of $8\mathbf{M} + 1\mathbf{S} + 1\mathbf{D} + 7\mathbf{a}$.

Readdition means that $(X_2 : Y_2 : Z_2)$ has been added to another point before. This means that computations depending only on $(X_2 : Y_2 : Z_2)$, such as X_2+Y_2,

can be cached from the previous addition. We have not found a way to save **M** or **S** in this case.

Special points. The above description of addition ignored the possibility of the special points $(0,1),(0,-1),(1,0),(-1,0)$ appearing as summands or as the sum. We now deal with that possibility. We represent these points as the vectors $(1,0,0),(-1,0,0),(0,-1,0),(0,1,0)$ respectively, as discussed in Section 3. We assume that d is not a square.

Special points as summands are easy to handle. If $Z_1 = 0$ or $Z_2 = 0$ then the sum of (X_1, Y_1, Z_1) and (X_2, Y_2, Z_2) is $(X_1X_2 - Y_1Y_2, X_2Y_1 + X_1Y_2, Z_1 + Z_2)$.

Even if neither summand is a special point, the sum could be a special point. If $I = 0$ and $Y_2Z_1 = Y_1Z_2$ then the sum is $(1,0,0)$. If $I = 0$ and $Y_2Z_1 = -Y_1Z_2$ then the sum is $(-1,0,0)$. If $H = 0$ and $Y_2Z_1 = -X_1Z_2$ then the sum is $(0,1,0)$. If $H = 0$ and $Y_2Z_1 = X_1Z_2$ then the sum is $(0,-1,0)$.

To derive these output rules, observe that two points (x_1, y_1) and (x_2, y_2) on the Edwards curve have sum $(0,1)$ if and only if $(x_2, y_2) = (-x_1, y_1)$. In this case $(Z_2/X_2, Z_2/Y_2) = (-Z_1/X_1, Z_1/Y_1)$ so, in the notation of our explicit formulas, $I = X_1Y_2 + Y_1X_2 = X_1Y_1Z_2/Z_1 - Y_1X_1Z_2/Z_1 = 0$ and $Y_2Z_1 = Y_1Z_2$. Similarly, two points (x_1, y_1) and (x_2, y_2) having sum $(0,-1)$ end up with $I = 0$ but with $Y_2Z_1 = -Y_1Z_2$; two points (x_1, y_1) and (x_2, y_2) having sum $(1,0)$ end up with $H = 0$ and $Y_2Z_1 = X_1Z_2$; two points (x_1, y_1) and (x_2, y_2) having sum $(-1,0)$ end up with $H = 0$ but with $Y_2Z_1 = -X_1Z_2$.

To see that the output rules are exclusive, suppose that $H = 0$ and $I = 0$. Then $X_1X_2 = Y_1Y_2$ and $X_1Y_2 + X_2Y_1 = 0$, so $X_1^2X_2 = X_1Y_1Y_2$ and $X_1Y_1Y_2 + X_2Y_1^2 = 0$, so $(X_1^2 + Y_1^2)X_2 = 0$; all variables are nonzero, so $X_1^2 + Y_1^2 = 0$. The curve equation $(X_1^2 + Y_1^2)Z_1^2 = X_1^2Y_1^2 + dZ_1^4$ now implies $0 = X_1^2(-X_1^2) + dZ_1^4$; i.e., $d = (X_1/Z_1)^4$, contradicting the assumption that d is not a square.

5 Doubling

Doubling refers to the case that the inputs $(X_1 : Y_1 : Z_1)$ and $(X_2 : Y_2 : Z_2)$ are known to be equal. If $X_1Y_1Z_1 = 0$ the special formulas from Section 4 apply. Otherwise inserting Z_1/X_1 for x_1 and x_2 and Z_1/Y_1 for y_1 and y_2 in the Edwards addition law we obtain

$$2(x_1, y_1) = \left(\frac{2X_1Y_1Z_1^2}{X_1^2Y_1^2 + dZ_1^4}, \frac{(X_1^2 - Y_1^2)Z_1^2}{X_1^2Y_1^2 - dZ_1^4}\right) = \left(\frac{2X_1Y_1}{X_1^2 + Y_1^2}, \frac{X_1^2 - Y_1^2}{X_1^2 + Y_1^2 - 2dZ_1^2}\right).$$

In the second equality we have used the curve equation to replace $X_1^2Y_1^2$ by $(X_1^2 + Y_1^2)Z_1^2 - dZ_1^4$, and then cancelled Z_1^2, reducing the overall degree of the formulas to 4. The resulting coordinates are

$$\begin{aligned} X_3 &= (X_1^2 + Y_1^2)(X_1^2 - Y_1^2) \\ Y_3 &= 2X_1Y_1(X_1^2 + Y_1^2 - 2dZ_1^2) \\ Z_3 &= 2X_1Y_1(X_1^2 - Y_1^2). \end{aligned}$$

The explicit formulas in this case need $3\mathbf{M} + 4\mathbf{S} + 1\mathbf{D} + 6\mathbf{a}$:

$$A = X_1^2;\ B = Y_1^2;\ C = A + B;\ D = A - B;\ E = (X_1 + Y_1)^2 - C;$$
$$Z_3 = D \cdot E;\ X_3 = C \cdot D;\ Y_3 = E \cdot (C - 2d \cdot Z_1^2).$$

6 Tripling

In Edwards coordinates tripling ($9\mathbf{M} + 4\mathbf{S} + 8\mathbf{a}$, or alternatively $7\mathbf{M} + 7\mathbf{S} + 16\mathbf{a}$) is faster than the sequential computation of a doubling ($3\mathbf{M} + 4\mathbf{S} + 6\mathbf{a}$) followed by an addition ($10\mathbf{M} + 1\mathbf{S} + 1\mathbf{D} + 7\mathbf{a}$). The main speedup comes from using the curve equation to reduce the degree of the tripling formulas. See Section 1 for credits and references.

For inverted Edwards coordinates with $X_1Y_1Z_1 \neq 0$ we now provide two sets of tripling formulas. Both sets have been added to the EFD [3] for formal verification. The first set needs $9\mathbf{M} + 4\mathbf{S} + 1\mathbf{D} + 10\mathbf{a}$:

$$A = X_1^2;\ B = Y_1^2;\ C = Z_1^2;\ D = A + B;\ E = 4(D - d \cdot C);$$
$$H = 2D \cdot (B - A);\ P = D^2 - A \cdot E;\ Q = D^2 - B \cdot E;$$
$$X_3 = (H + Q) \cdot Q \cdot X_1;\ Y_3 = (H - P) \cdot P \cdot Y_1;\ Z_3 = P \cdot Q \cdot Z_1.$$

The second set needs $7\mathbf{M} + 7\mathbf{S} + 1\mathbf{D} + 17\mathbf{a}$:

$$A = X_1^2;\ B = Y_1^2;\ C = Z_1^2;\ D = A + B;\ E = 4(D - d \cdot C);$$
$$H = 2D \cdot (B - A);\ P = D^2 - A \cdot E;\ Q = D^2 - B \cdot E;$$
$$X_3 = (H + Q) \cdot ((Q + X_1)^2 - Q^2 - A);\ Y_3 = 2(H - P) \cdot P \cdot Y_1;$$
$$Z_3 = P \cdot ((Q + Z_1)^2 - Q^2 - C).$$

The second set is faster if $\mathbf{S}/\mathbf{M}$ is small.

Triplings, like doublings, have similar speeds for inverted Edwards coordinates and standard Edwards coordinates. Inverted Edwards coordinates speed up addition by reducing the degree of the formulas, but the curve equation already appears to have produced the minimal degrees for doublings and triplings, so the lack of further improvements does not come as a surprise.

Special points. Tripling special points is very easy: $3(X_1, Y_1, 0) = (X_1, -Y_1, 0)$.

7 Comparison

The EFD [3] is meant to provide an up-to-date database with all curve forms and coordinate systems ever proposed. A comparison in a paper can only give a snapshot of what is known today. Most of the conclusions in [4] remain unchanged, but science has developed even in the short time since then!

Duquesne in [7] proposed what we call "extended Jacobi-quartic coordinates," now described in detail in the EFD. Duquesne's addition formulas use

$9\mathbf{M}+2\mathbf{S}+1\mathbf{D}$, saving $1\mathbf{M}-1\mathbf{S}$ compared to standard Edwards coordinates. These addition formulas are strongly unified but not complete: they can be used for doublings but have some exceptional cases. In the EFD we improve Duquesne's formulas to use $8\mathbf{M} + 3\mathbf{S} + 1\mathbf{D}$, saving another $1\mathbf{M} - 1\mathbf{S}$.

Hisil, Carter, and Dawson in [9] improved various elliptic-curve addition formulas, and in particular gave doubling formulas for extended Jacobi-quartic coordinates using $3\mathbf{M} + 4\mathbf{S}$. This is as fast as doubling in standard Edwards coordinates.

However, addition in inverted Edwards coordinates is even faster, saving an additional $2\mathbf{S}-1\mathbf{M}$, and has just as fast doublings (for small d). Inverted Edwards coordinates have the same advantage of being strongly unified.

The comparisons of different coordinate systems for scalar multiplications using DBNS in [2] have been updated to include the speeds of [7] and [9], and to include inverted Edwards coordinates. The comparison shows that, out of currently known methods for scalar multiplication on elliptic curves, inverted Edwards coordinates (with very few triplings) are the fastest.

To conclude we summarize the current situation: Edwards coordinates offer the only complete addition law stated in the literature. If completeness is not required then inverted Edwards coordinates are the new speed leader.

References

1. Barua, R., Lange, T. (eds.): INDOCRYPT 2006. LNCS, vol. 4329. Springer, Heidelberg (2006)
2. Bernstein, D.J., Birkner, P., Lange, T., Peters, C.: Optimizing Double-Base Elliptic-Curve Single-Scalar Multiplication. In: Srinathan, K., Pandu Rangan, C., Yung, M. (eds.) INDOCRYPT 2007. LNCS, vol. 4859, pp. 167–182. Springer, Heidelberg (2007)
3. Bernstein, D.J., Lange, T.: Explicit-Formulas Database, `http://www.hyperelliptic.org/EFD`
4. Bernstein, D.J., Lange, T.: Faster Addition and Doubling on Elliptic Curves. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 29–50. Springer, Heidelberg (2007), `http://cr.yp.to/newelliptic/`
5. Bosma, W., Lenstra Jr., H.W.: Complete Systems of Two Addition Laws for Elliptic Curves. J. Number Theory 53, 229–240 (1995)
6. Doche, C., Imbert, L.: Extended Double-Base Number System with Applications to Elliptic Curve Cryptography. In: Barua, R., Lange, T. (eds.) INDOCRYPT 2006. LNCS, vol. 4329, pp. 335–348. Springer, Heidelberg (2006)
7. Duquesne, S.: Improving the Arithmetic of Elliptic Curves in the Jacobi Model. Information Processing Letters 104, 101–105 (2007)
8. Edwards, H.M.: A Normal Form for Elliptic Curves. Bulletin of the American Mathematical Society 44, 393–422 (2007), `http://www.ams.org/bull/2007-44-03/S0273-0979-07-01153-6/home.html`
9. Hisil, H., Carter, G., Dawson, E.: New Formulae for Efficient Elliptic Curve Arithmetic. In: Srinathan, K., Pandu Rangan, C., Yung, M. (eds.) INDOCRYPT 2007. LNCS, vol. 4859, Springer, Heidelberg (2007)
10. Kurosawa, K. (ed.): ASIACRYPT 2007. LNCS, vol. 4833. Springer, Heidelberg (2007)

Spectra of Boolean Functions, Subspaces of Matrices, and Going Up Versus Going Down

Gary McGuire*

School of Mathematical Sciences
University College Dublin, Ireland
gary.mcguire@ucd.ie

Abstract. We will discuss two different but related topics. We first give a connection between the Fourier spectrum of Boolean functions and subspaces of skew-symmetric subspaces where each nonzero element has a lower bound on its rank. Secondly, we discuss some connections between bent and near-bent functions.

1 Introduction

Let V_n denote any n-dimensional vector space over $\mathbb{F}_2$. The Fourier transform of a function $f : V_n \longrightarrow V_m$ is defined by

$$\widehat{f}(a,b) := \sum_{x \in V_n} (-1)^{\langle b, f(x)\rangle + \langle a, x\rangle}$$

for $a \in V_n$ and $b \in V_m$, $b \neq 0$. The angular brackets $\langle\ ,\ \rangle$ denote any inner product on the relevant vector spaces. The Fourier spectrum of f is the subset of $\mathbb{Z}$ consisting of the set of values of $\widehat{f}$, over all a and b ($b \neq 0$), and is independent of the inner products used.

If $m = 1$ then $V_m = V_1 = \mathbb{F}_2$ and any function $f : V_n \longrightarrow \mathbb{F}_2$ is called a Boolean function.

Bent functions are Boolean functions which have Fourier spectrum $\{\pm 2^{n/2}\}$. Since the spectrum values are integers, bent functions can only exist when n is even. We shall call a Boolean function *near-bent* if its Fourier spectrum is $\{0, \pm 2^{(n+1)/2}\}$. Near-bent functions can only exist when n is odd. Near-bent functions have also been called "Gold-like" in the literature.

A function from $V_n \longrightarrow V_n$ is said to be *almost bent* if it has Fourier spectrum $\{0, \pm 2^{(n+1)/2}\}$. As for near-bent functions, almost bent functions can only exist when n is odd.

An important case is when the vector space is actually a field. For this paper let L denote $\mathbb{F}_{2^n}$, the finite field with 2^n elements. Let tr denote the trace map from L to $\mathbb{F}_2$. We usually use the inner product $\langle x, y\rangle = \mathrm{tr}(xy)$ when $V_n = L$.

* Research supported by the Claude Shannon Institute, Science Foundation Ireland Grant 06/MI/006.

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 28–37, 2007.
© Springer-Verlag Berlin Heidelberg 2007

For a function $f : L \longrightarrow L$ the formula for $\widehat{f}$ becomes

$$\widehat{f}(a,b) := \sum_{x \in L} (-1)^{\mathrm{tr}(bf(x)+ax)}. \tag{1}$$

In this context, f is almost bent if and only if each of the Boolean functions $\mathrm{tr}(bf(x))$ is near-bent, for all $b \in L$, $b \neq 0$. If f is a monomial permutation, say $f(x) = x^d$ where $(d, 2^n - 1) = 1$, then f is almost bent if and only if $\mathrm{tr}(f(x))$ is near-bent. This is because we may write any $b \in L$ as c^d, and then replacing x by x/c in (1) gives

$$\widehat{f}(a,b) = \sum_{x \in L} (-1)^{\mathrm{tr}(c^d x^d + ax)} = \sum_{x \in L} (-1)^{\mathrm{tr}(x^d + ac^{-1}x)} = \widehat{f}(ac^{-1}, 1).$$

It follows that the Fourier spectrum of $f(x) = x^d$ will be the same as the Fourier spectrum of the Boolean function $\mathrm{tr}(x^d)$, when $(d, 2^n - 1) = 1$. The most famous examples of this are the almost bent Gold functions $f(x) = x^{2^k+1}$ where k is relatively prime to n and n is odd. Bent and near-bent functions are discussed in section 4.

Let us next introduce the topic of subspaces of matrices where all nonzero matrices have a certain rank. In differential topology, one important problem is the construction of immersions from real projective space $\mathbb{P}^n(\mathbb{R})$ into Euclidean space $\mathbb{R}^{n+k}$. There are open problems dating from the early 1960s concerning the minimal possible k for which such an immersion exists. Let $M_{n,n}(F)$ denote the vector space of all $n \times n$ matrices over a field F. It can be shown that the highest possible dimension of a subspace of $M_{n,n}(\mathbb{R})$ not containing any elements of rank 1 is directly related to the question of which k are possible. It has also been shown that subspaces consisting of all symmetric matrices, or all skew-symmetric matrices, are of similar importance to the problem of constructing embeddings into Euclidean space. Also, connections have been found between the embedding problem and the immersion problem, so the symmetric case has implications for the immersion problem. More details can be found in [9].

Connections between subspaces of matrices with good rank properties and spacetime codes are studied in Calderbank et al [2] and Lu-Kumar [7].

Let $L(n,k,F)$ denote the maximal dimension of a subspace of $M_{n,n}(F)$ all of whose nonzero elements have rank at least k. Let $L_S(n,k,F)$ denote the maximal dimension of a subspace of $M_{n,n}(F)$ all of whose nonzero elements are skew-symmetric and have rank at least k. In section 2 we will discuss the case of $F = \mathbb{F}_2$ and k large. In particular, we discuss $L_S(n, n-1, \mathbb{F}_2)$ and $L_S(n, n-3, \mathbb{F}_2)$ when n is odd and its relationship to the Fourier spectrum of functions. These methods carry over easily to finite fields of odd characteristic, and are well known. We will discuss carrying over the methods to infinite fields.

2 The Connection Between Subspaces and Values

First we shall outline the connection between the values in the Fourier spectrum, and the ranks of the elements in some subspaces of matrices. The connection

goes through bilinear forms. This work is all implicit in Delsarte and Goethals [4]. They translate the results on bilinear forms into results in coding theory. It is known that such results in coding theory can be translated into results on the Fourier spectra of Boolean functions. We will directly translate results from Boolean functions to results on subspaces of matrices. Therefore, we are not going to present any new results in section 2.1, but we feel that it is useful to directly explain the connection without going through coding theory. In section 2.2 we will present a direction for future research, and a new result.

We recall some definitions for bilinear forms. Let $L = \mathbb{F}_{2^n}$ as before. A bilinear form $B : L \times L \longrightarrow \mathbb{F}_2$ is said to be symplectic if $B(x, x) = 0$ for all x. By definition the radical of B is

$$rad(B) = \{x \in L : B(x, y) = 0 \text{ for all } y \in L\}.$$

The rank of B is defined to be $n - \dim(rad(B))$, and a well known theorem states that the rank must be even.

Finally, let us state that although we only consider forms like $\mathrm{tr}(x^2y + xy^2)$ in characteristic 2, the arguments in this section carry over in a straightforward manner to alternating forms $\mathrm{tr}(x^py - xy^p)$ in characteristic p.

2.1 Background

Nothing in this section is new. We will use some motivating examples, which illustrate all the important ideas. In this section n is odd. For $a \in L$ the function

$$B_a(x, y) = \mathrm{tr}(a(x^2y + xy^2))$$

a symplectic bilinear form on L. The rank of B_a is $n - w_a$ where $w_a = \dim rad(B_a)$. By definition,

$$rad(B_a) = \{x \in L : \mathrm{tr}(a(x^2y + xy^2)) = 0 \ \ \forall y \in L\}$$

$$= \{x \in L : \mathrm{tr}((ax^2 + a^{2^{n-1}}x^{2^{n-1}})y) = 0 \ \ \forall y \in L\}.$$

Since the trace form is nondegenerate, x is in $rad(B_a)$ if and only if $ax^2 + a^{2^{n-1}}x^{2^{n-1}} = 0$. Squaring this gives

$$a^2x^4 + ax = 0. \tag{2}$$

Initially it appears possible that this equation could have 4 solutions in L. However this would imply that B_a has odd rank, since n is odd. Thus, the equation has two solutions in L. (Alternatively one can solve: if $ax \neq 0$ this implies $ax^3 = 1$, which has a unique solution for x.) Thus $w_a = 1$ for all $a \neq 0$. This also shows B_a is the zero form if and only if $a = 0$. Therefore, B_a has rank $n-1$ for all $a \neq 0$.

We note that the same argument works for any $\mathrm{tr}(a(x^\sigma y - xy^\sigma))$ where σ is a generating automorphism.

Now we introduce subspaces of skew-symmetric matrices. Observe that the B_a ($a \in L$) form a vector space over $\mathbb{F}_2$. Choosing a basis of L over $\mathbb{F}_2$, the matrices corresponding to these forms will yield an n-dimensional vector space of $n \times n$ (skew) symmetric matrices with zero diagonal such that all nonzero members have rank $n-1$. This is the maximum dimension for such a subspace, by a theorem in [4]. All this is well known over finite fields.

Next, we relate this to the Fourier spectrum of the function x^3. To see the connection it is best to review the calculation of the spectrum. The standard method is to square $\widehat{f}(a,b)$, perform a substitution, and rearrange to get

$$\widehat{f}(a,b)^2 = 2^n \sum_{u \in rad(B_a)} (-1)^{\mathrm{tr}(au^3+bu)}.$$

Now we see the connection to finding the radical of B_a. We computed the radical above and we saw that it has dimension 1. It is then clear that $\widehat{f}(a,b)^2$ is $2^n \pm 2^n$, and so is either 0 or 2^{n+1}.

In summary, the point we wish to make is that x^3 being an almost bent function is closely related to all nonzero elements in the vector space of skew-symmetric matrices B_a having rank $n-1$. In general the two facts are not *equivalent*, however. The ranks of the bilinear forms are the real connection, and although in this example this allowed us to determine the true values in the spectrum, in general more work has to be done in order to determine the precise spectrum.

Next, one could ask for subspaces where all ranks are $n-1$ or $n-3$. By [4], the maximum dimension for such a subspace is $2n$. A function with spectrum $\{0, \pm 2^{(n+1)/2}, \pm 2^{(n+3)/2}\}$ should correspond to such a subspace, under the connection we have illustrated. Here is an example (from [4]).

Consider the set of bilinear forms

$$B_{c,d}(x,y) = \mathrm{tr}(c(x^2y + xy^2) + d(x^4y + xy^4))$$

over all $c, d \in L$. This set of bilinear forms is an $\mathbb{F}_2$-vector space of dimension $2n$. We claim that each nonzero form has rank $n-1$ or $n-3$. This is the same as saying that the radicals have dimension 1 or 3. To show this, write

$$B_{c,d}(x,y) = \mathrm{tr}(y^4(c^4x^8 + c^2x^2 + d^4x^{16} + dx))$$

and then $x \in rad(B_{c,d})$ if and only if

$$c^4x^8 + c^2x^2 + d^4x^{16} + dx = 0.$$

Initially it appears possible that this equation could have 16 solutions in L. However, because the dimension of the solution space is odd (because the rank of $B_{c,d}$ is even), the dimension must be 1 or 3. We are done.

The same argument repeated for the forms $\mathrm{tr}(\sum_{i=1}^{k} c_i(x^{2^i}y + xy^{2^i}))$ will give kn-dimensional subspaces of matrices of ranks $n-1$, $n-3$, ... $n-2k+1$. This recovers a result of Delsarte and Goethals [4], which also appears in [6]. For example, in the $3n$-dimensional space of forms

$$\mathrm{tr}(c(x^2y + xy^2) + d(x^4y + xy^4) + e(x^8y + xy^8))$$

all nonzero elements have rank $n-1, n-3$ or $n-5$.

This $3n$-dimensional subspace contains three obvious $2n$-dimensional subspaces, consisting of all elements where one of c, d, e is 0. The $e = 0$ subspace has no elements of rank $n-5$, as shown above. What about the $d = 0$ subspace? This consists of forms $B_{c,e}(x, y) = \mathrm{tr}(c(x^2y + xy^2) + e(x^8y + xy^8))$. We try the same argument: x is in the radical of this form if and only if

$$c^8x^{16} + c^4x^4 + e^8x^{64} + ex = 0. \tag{3}$$

Since the rank of the form is even, it follows that this equation has 2^j solutions in L, where $j \in \{1, 3, 5\}$. It is true, but not obvious, that this equation cannot have 32 solutions in L. (This is proved as part of the calculation of the Fourier spectrum of Kasami-Welch functions – we give a more general proof in the next section.) This implies that the forms $B_{c,e}$ have rank $n-1$ or $n-3$. It is somewhat surprising that rank $n-5$ does not appear, and that the same result holds for the subspace of forms $\mathrm{tr}(c(x^2y + xy^2) + e(x^8y + xy^8))$ as holds for the subspace of forms $\mathrm{tr}(c(x^2y + xy^2) + d(x^4y + xy^4))$.

2.2 Future Work

Firstly, the known bounds on $L_S(n, k, \mathbb{F}_2)$ when n is odd due to Delsarte and Goethals have not been generalised to infinite fields. For example, the value of $L_S(n, n-1, F)$ is not known if F is an infinite field. The conjectured value is n, as in the finite field case. This is one area for future work.

Secondly, one can try to generalize the connections outlined in section 2.1. Gow and Quinlan [5] have generalised some results on bilinear forms over finite fields to arbitrary field extensions with a cyclic Galois group. In particular we quote the following theorem, which we will use.

Theorem 1. *Let L/K be a cyclic extension of degree n, with Galois group generated by σ. Let k be an integer with $1 \le k \le n$, and let w be a polynomial of degree k in $L[t]$. Let*

$$R = \{x \in L : w(\sigma)x = 0\}.$$

Then we have $\dim_K(R) \le k$.

We now present a result promised in the previous section.

Theorem 2. *Let L/K be a cyclic extension of degree n, n odd, with Galois group generated by σ. Consider the set of bilinear forms* $\mathrm{tr}(c(x^\sigma y + xy^\sigma) + e(x^{\sigma^3}y + xy^{\sigma^3}))$ *where $c, e \in L$. Then the ranks of these forms are $n-1$ or $n-3$.*

Proof: Let $B_{c,e} = \mathrm{tr}(c(x^\sigma y + xy^\sigma) + e(x^{\sigma^3}y + xy^{\sigma^3}))$. By definition,

$$rad(B_{c,e}) = \{x \in L : \mathrm{tr}(c(x^\sigma y + xy^\sigma) + e(x^{\sigma^3}y + xy^{\sigma^3})) = 0 \ \ \forall y \in L\}$$

$$= \{x \in L : \mathrm{tr}((cx^\sigma + c^{\sigma^{-1}}x^{\sigma^{-1}} + e^{\sigma^3}x^{\sigma^3} + e^{\sigma^{-3}}x^{\sigma^{-3}})y) = 0 \ \ \forall y \in L\}.$$

Since the trace form is nondegenerate, x is in $rad(B_a)$ if and only if

$$cx^{\sigma} + c^{\sigma^{-1}} x^{\sigma^{-1}} + e^{\sigma^3} x^{\sigma^3} + e^{\sigma^{-3}} x^{\sigma^{-3}} = 0.$$

Applying σ^3 to this equation gives

$$c^{\sigma^3} x^{\sigma^4} + c^{\sigma^2} x^{\sigma^2} + e^{\sigma^6} x^{\sigma^6} + ex = 0.$$

This can be written $w(\sigma)x = 0$ where $w(t)$ is the polynomial $c^{\sigma^3}t^4 + c^{\sigma^2}t^2 + e^{\sigma^6}t^6 + e$ in $L[t]$. Putting $u = t^2$, $w(t) = w'(u)$ where $w'(u) = c^{\sigma^3}u^2 + c^{\sigma^2}u + e^{\sigma^6}u^3 + e$. Letting $\tau = \sigma^2$, we may conclude

$$rad(B_{c,e}) = \{x \in L : w'(\tau)x = 0\}.$$

Since n is odd, τ also generates the Galois group of L/K. By Theorem 1, $rad(B_{c,e})$ has K-dimension at most 3. It follows that the rank of $B_{c,e}$ is $n-1$ or $n-3$. □

We remark that Theorem 1 applied directly to the forms in Theorem 2 would imply that ranks are $n-1, n-3$ and $n-5$. We also remark that the proof of Theorem 2 applies to the Kasami-Welch forms $\mathrm{tr}(c(x^{2^k}y + xy^{2^k}) + e(x^{2^{3k}}y + xy^{2^{3k}}))$ in the case $L = \mathbb{F}_{2^n}$, $(k, n) = 1$. In particular, this theorem proves that equation (3) cannot have 32 solutions, as we remarked in the previous section.

3 Even n

Suppose n is even. The situation is quite different with regard to subspaces of matrices.

It is well known that the Walsh spectrum of x^3 is $\{0, \pm 2^{n/2}, \pm 2^{(n+2)/2}\}$ in this case. It is no longer true that each function $\mathrm{tr}(bx^3)$ has the same Fourier spectrum. There are two types. If b is a cube, then we may do as in the n odd case and the spectrum of $\mathrm{tr}(bx^3)$ is the same as that of $\mathrm{tr}(x^3)$, which is $\{0, \pm 2^{(n+2)/2}\}$. However, if b is not a cube then the spectrum of $\mathrm{tr}(bx^3)$ is $\{\pm 2^{n/2}\}$. In other words, the subspace of bilinear forms $\mathrm{tr}(c(x^2y + xy^2))$ for $c \in L$ contains elements of rank n (when c is not a cube) and rank $n-2$ (when c is a cube).

Since the cubes (and the non-cubes) are not closed under addition, we do not get subspaces in the same way as when n is odd.

4 Going Up and Down

This section concerns a different topic. Because bent functions exist in even dimensions, and near-bent functions exist in odd dimensions, the possibility exists of moving up and down between bent and near-bent functions. In this section we will discuss each of the four possibilities.

4.1 Going Up From a Bent Function

Given a bent function on V_n, n even, we wish to consider adding one variable to create a near-bent function in $n+1$ variables. This is straightforward to prove.

Let $f(x)$ be a bent function on V_n, where n is even. Let y be a new Boolean variable, and consider the function $g(x,y) = f(x) + y$ on the $n+1$ dimensional vector space $V_n \oplus V_1$. It is easy to see that g is near-bent, as follows.

Any linear functional λ on $V_n \oplus V_1$ can be written as $\lambda(x,y) = \lambda'(x) + \delta y$ where λ' is a linear functional on V_n and δ is 0 or 1. Then

$$\widehat{g}(\lambda) = \sum_{(x,y)} (-1)^{g(x,y)+\lambda(x,y)} \tag{4}$$

$$= \sum_{(x,y)} (-1)^{f(x)+y+\lambda'(x)+\delta y} \tag{5}$$

$$= \begin{cases} 2 \sum_x (-1)^{f(x)+\lambda'(x)} & \text{if } \delta = 1 \\ \sum_{(x,y)} (-1)^{f(x)+y+\lambda'(x)} & \text{if } \delta = 0 \end{cases} \tag{6}$$

$$= \begin{cases} 2\widehat{f}(\lambda') & \text{if } \delta = 1 \\ 0 & \text{if } \delta = 0. \end{cases} \tag{7}$$

If f is bent, then clearly the spectrum of g is $\{0, \pm 2^{(n+2)/2}\}$ so g is near-bent.

We remark that g is what is called partially bent – the sum of a bent function and a linear function.

One might ask whether all near-bent functions arise in this way? The answer is no. Because, note that the function g is a bent function (namely f) when restricted to the hyperplane $y = 0$. There are near-bent functions that are not bent when restricted to any hyperplane – we have checked this by computer for some Kasami-Welch near-bent functions, for example. Such near-bent functions cannot arise from this construction.

4.2 Going Down from a Bent Function

Given a bent function on V_n, n even, we wish to consider restriction to a hyperplane to create a near-bent function in $n-1$ variables. This has been proved to be always true in Canteaut et. al. [1], see Theorem V.3 there. In that paper, the authors state that they do not know another way to prove that Dillon's PS_{ap} bent functions restrict to near-bent functions. We shall give such a proof now.

The construction of the PS_{ap} bent functions starts with a balanced function $g : K \longrightarrow \mathbb{F}_2$ where $K = \mathbb{F}_{2^t}$. Dillon's result states that the function $f(x,y) = g(xy^{2^n-2})$ is bent on $K \times K$ (actually the result is more general, concerning partial spread bent functions). Note that $g(xy^{2^n-2}) = g(x/y)$ if $y \neq 0$.

In $K \times K$ let H_a denote the line $\{(x, ax) : x \in K\}$ and let $H_\infty = \{(0,y) : y \in K\}$. These $2^t + 1$ lines intersect pairwise in $(0,0)$ and partition $K \times K$. The linear span of any two of these lines is $K \times K$.

Let H be a hyperplane in $K \times K$. We must show that the Fourier transform of $f|_H$ takes values $0, \pm 2^t$.

Let $H'_a := H_a \cap H$, for $a \in \mathbb{P}^1(K)$. Let λ be a linear functional on H. We shall break the sum over H up into sums over each H'_a, taking care to remove $(0,0)$ first.

$$
\begin{aligned}
&\widehat{f|_H}(\lambda) \\
&= \sum_{(x,y)\in H} (-1)^{f(x,y)+\lambda(x,y)} \\
&= 1 + \sum_{(x,y)\neq(0,0)\in H} (-1)^{f(x,y)+\lambda(x,y)} \\
&= 1 + \sum_{(x,y)\in H'_\infty \setminus \{0,0\}} (-1)^{\lambda(x,y)} + \sum_{a\neq\infty} \sum_{(x,y)\in H'_a \setminus \{0,0\}} (-1)^{g(x/y)+\lambda(x,y)} \\
&= \sum_{(x,y)\in H'_\infty} (-1)^{\lambda(x,y)} + \sum_{a\neq\infty} (-1)^{g(H_a)} \sum_{(x,y)\in H'_a \setminus \{0,0\}} (-1)^{\lambda(x,y)} \\
&= \sum_{(x,y)\in H'_\infty} (-1)^{\lambda(x,y)} + \sum_{a\neq\infty} (-1)^{g(H_a)} \left(\sum_{(x,y)\in H'_a} (-1)^{\lambda(x,y)} - 1 \right) \\
&= \sum_{(x,y)\in H'_\infty} (-1)^{\lambda(x,y)} - \sum_{a\neq\infty} (-1)^{g(H_a)} + \sum_{a\neq\infty} (-1)^{g(H_a)} \left(\sum_{(x,y)\in H'_a} (-1)^{\lambda(x,y)} \right) \\
&= \sum_{(x,y)\in H'_\infty} (-1)^{\lambda(x,y)} + \sum_{a\neq\infty} (-1)^{g(H_a)} \left(\sum_{(x,y)\in H'_a} (-1)^{\lambda(x,y)} \right) \\
&= \sum_{a\in\mathbb{P}^1(K)} (-1)^{g(H_a)} \left(\sum_{(x,y)\in H'_a} (-1)^{\lambda(x,y)} \right)
\end{aligned}
$$

where we used the fact that g is balanced, so $\sum_{a\neq\infty}(-1)^{g(H_a)} = 0$. We write $g(H_a)$ to denote the value of g at any element of H_a.

We must now distinguish some cases in order to finish the proof. If $\lambda = 0$ then it is easy to check that $\widehat{f|_H}(\lambda) = 2^t$. For the remainder, assume $\lambda \neq 0$. First we assume that H does not contain any H_a. Then each H'_a is a hyperplane in H_a. The inner summation is 0 unless λ vanishes on H'_a, and there are precisely two such a for any λ, as $\lambda \neq 0$. So $\widehat{f|_H}(\lambda) = \pm 2^{t-1} \pm 2^{t-1}$ which is 0 or $\pm 2^t$. Secondly, assume that H does contain one of the H_a, say H_ℓ. (H cannot contain two H_a since two H_a's generate the whole space $K \times K$.) Then the inner sum will be 0 unless λ is the unique linear functional whose kernel is H_ℓ. Thus the value of $\widehat{f|_H}(\lambda)$ in this case is $(-1)^{g(H_\ell)} 2^t$.

We thank John Dillon for discussions about these functions.

4.3 Going Up from an Near-Bent Function

Given a near-bent function $f(x)$ on V_n, n odd, we wish to consider adding one variable to create a bent function in $n+1$ variables. The same argument as in

section 4.1 does not work, because adding one variable results in a function of $n+1$ variables with Fourier spectrum $\{0, \pm 2^{(n+3)/2}\}$, which is therefore not bent.

However, it is sometimes possible to go up by other methods. Suppose there exists another near-bent function $h(x)$ on V_n, such that the support of $\widehat{h}$ does not intersect the support of $\widehat{f}$. (The supports both have cardinality 2^{n-1} and so they partition V_n.) In this case, let y be another Boolean variable, and define $g(x,y) = yf(x) + (y+1)h(x)$ on the $n+1$ dimensional vector space $V_n \oplus V_1$. Then

$$\begin{aligned}\widehat{g}(\lambda) &= \sum_{(x,y)\in V_n\oplus V_1} (-1)^{g(x,y)+\lambda(x,y)} \\ &= \sum_{(x,0)\in V_n\oplus V_1} (-1)^{h(x)+\lambda(x,0)} + \sum_{(x,1)\in V_n\oplus V_1} (-1)^{f(x)+\lambda(x,1)} \\ &= \widehat{h}(\lambda) + \widehat{f}(\lambda).\end{aligned}$$

Since $\widehat{h}$ and $\widehat{f}$ have disjoint support, and both have Fourier spectrum $\{0, \pm 2^{(n+1)/2}\}$, the values of $\widehat{g}(\lambda)$ are $\pm 2^{(n+1)/2}$, so g is bent.

An example of this is $f(x) = x^3$ and $h(x) = x^5 + x$, where $V_n = L$. The support of a Gold function such as x^3 (or x^5) is known to be the complement of the hyperplane H of trace 0 elements. It is easy to show that the support of h is H. Therefore, by the argument above, $yf(x)+(y+1)h(x) = yx^3+(y+1)(x^5+x)$ is a bent function (of algebraic degree 3). We do not know if this is a new bent function. Perhaps new bent functions can be constructed in this way.

4.4 Going Down from a Near-Bent Function

Given a near-bent function on V_n, n odd, we wish to consider restriction to a hyperplane to create a bent function in $n-1$ variables. This is sometimes possible, but not always possible. In [1] some conditions are given for the restriction to a hyperplane of a near-bent function to be bent. In [8] the restriction of Gold functions is considered. It is proved that the restriction of $f(x) = x^{2^k+1}$ (where $(k,n) = 1$) to a hyperplane $h^\perp$ is bent if and only if $\mathrm{tr}(h) = 1$. Here V_n is L, the finite field of order 2^n. We give a different proof here: we need the fact that the support of $\widehat{f}$ is the complement of the hyperplane H of trace 0 elements. Fix $h \notin H$. For $a \in L$, define g by

$$\widehat{g}(a) = \frac{1}{2}\big(\widehat{f}(a) + \widehat{f}(a+h)\big).$$

This is well-defined on the quotient space $L/\langle h\rangle$. Since exactly one of $a, a+h$ is in H, one of $\widehat{f}(a), \widehat{f}(a+h)$ is 0 and the other is $\pm 2^{(n+1)/2}$. Therefore $\widehat{g}(a) = \pm 2^{(n-1)/2}$, so g is bent.

Note that the proof did not require H to be a hyperplane; the argument only required that exactly one of $a, a+h$ is in H. For the Kasami-Welch function $k(x) = x^{4^k-2^k+1}$ with $3k \equiv \pm 1 \pmod{n}$, the support of $\widehat{k}$ is the set of $a \in L$ with $\mathrm{tr}(a^{2^k+1}) = 1$. Since exactly one of $a, a+1$ is in the support, the same

argument works to show that $k(x)$ is bent when restricted to the hyperplane $1^{\perp}$ (i.e., the trace 0 elements).

In [3] the Kasami-Welch functions $x^{4^k-2^k+1}$ are considered in greater detail.

Acknowledgements. We thank John Dillon, Carl Bracken, Philippe Langevin, Gregor Leander and Rod Gow for discussions which have helped this article.

References

1. Canteaut, A., Carlet, C., Charpin, P., Fontaine, C.: On Cryptographic Properties of the Cosets of $R(1, m)$. IEEE Trans. Inform. Theory 47(4), 1494–1513 (2001)
2. Calderbank, A.R., Diggavi, S.N., Al-Dhahir, N.: Space-Time Signaling Based on Kerdock and Delsarte-Goethals Codes. In: IEEE ICC 2004, vol. 1, pp. 483–487 (2004)
3. Dillon, J.F., McGuire, G.: Kasami-Welch Functions on a Hyperplane (submitted)
4. Delsarte, P., Goethals, J.M.: Alternating Bilinear Forms over GF(q). J. Comb. Th. Ser. A 19, 26–50 (1975)
5. Gow, R., Quinlan, R.: On the Vanishing of Subspaces of Alternating Bilinear Forms. Linear and Multilinear Algebra 54, 415–428 (2006)
6. Gow, R., Quinlan, R.: Galois Extensions and Subspaces of Alternating Bilinear Forms with Special Rank Properties (submitted)
7. Lu, H.F.F., Kumar, P.V.: Rate-Diversity Tradeoff of Space-Time Codes with Fixed Alphabet and Optimal Constructions for PSK Modulation. IEEE Trans. Inform. Theory 49(10), 2747–2751 (2003)
8. Lahtonen, J., McGuire, G., Ward, H.N.: Gold and Kasami-Welch Functions, Quadratic Forms, and Bent Functions. In: Advances in Mathematics of Communications (2007)
9. Petrovic, Z.: Nonsingular Bilinear Maps, Spaces of Matrices, Immersions and Embeddings. In: Contemporary Geometry and Related Topics, Belgrade (2006), http://www.emis.de/proceedings/CGRT2005/

Efficient List Decoding of Explicit Codes with Optimal Redundancy

Atri Rudra

Department of Computer Science and Engineering
University of Buffalo, State University of New York
Buffalo, 14260, USA
atri@cse.buffalo.edu

Abstract. Under the notion of list decoding, the decoder is allowed to output a small list of codeword such that the transmitted codeword is present in the list. Even though combinatorial limitations on list decoding had been known since the 1970's, there was essentially no algorithmic progress till the breakthrough works of Sudan [14] and Guruswami-Sudan [11] in the mid to late 1990's. There was again a lull in algorithmic progress till a couple of recent papers [12,8] closed the gap in our knowledge about combinatorial and algorithmic limitations of list decoding (for codes over large alphabets). This article surveys these latter algorithmic progress.

1 Introduction

Under the list decoding problem (introduced in [1,16]), given a code $C \subseteq \Sigma^n$, an error parameter $0 \leqslant \rho \leqslant 1$ and a received word $\mathbf{y} \in \Sigma^n$; the decoder should output all codewords in C that are within Hamming distance ρn of $\mathbf{y}$. Suppressing the motivation for considering such an error recovery model for the time being, let us consider the following natural trade-off: Given that one wants to correct ρ fraction of errors via list decoding, what is the maximum rate R that a code can have?

Before we address this question, let us formally define the notion of list decoding we will consider in this survey. For a real $0 \leqslant \rho \leqslant 1$ and an integer $L \geqslant 1$, we will call a code $C \subseteq \Sigma^n$ to be (ρ, L)-list decodable if for every received word $\mathbf{y} \in \Sigma^n$, $|\{c \in C | \Delta(c, \mathbf{y}) \leqslant \rho n\}| \leqslant L$ where $\Delta(c, \mathbf{y})$ denotes the Hamming distance between the vectors c and $\mathbf{y}$. Note that the problem is interesting only when L is small: in this survey L is considered to be small if it is polynomially bounded in n.

Using a standard random coding argument it can be show that there exists $(\rho, O(1/\varepsilon))$ list decodable codes over alphabets of size q with rate $R \geqslant 1 - H_q(\rho) - o(1)$ where $H_q(x) = -x \log_q \left(\frac{x}{q-1}\right) - (1-x)\log_q(1-x)$ is the q-ary entropy function (cf. [17,2]). Further, a simple counting argument shows that R must be at most $1 - H_q(\rho)$ (for $R > 1 - H_q(\rho)$ the list size L needs to be super-polynomial in n). In other words, the maximum fraction of errors that can be

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 38–46, 2007.
© Springer-Verlag Berlin Heidelberg 2007

corrected (via list decoding) using a rate R code (or the *list decoding capacity*), is given by the trade-off $H_q^{-1}(1-R)$. For $q = 2^{\Omega(1/\varepsilon)}$, $H_q^{-1}(1-R) \geqslant 1-R-\varepsilon$ (cf. [13]). In other words, for large enough alphabets, the list decoding capacity is $\rho_{\mathrm{cap}}(R) = 1-R$.

Now is a good time to compare the list decoding capacity with what can be achieved with "usual" notion of decoding for the worst-case noise model (called *unique decoding*), where the decoder has to always output the transmitted word. Note that list decoding is a relaxation where the decoder is allowed to output a list of codewords (with the guarantee that the transmitted codeword is in the list). It is well known that unique decoding can only correct up to half the minimum distance of the code, which along with the Singleton bound implies the following limit on the fraction of errors that can be corrected: $\rho_U(R) = (1-R)/2$. In other words, list decoding has the potential to correct *twice* as many errors than unique decoding.

However, in order to harness the real potential of list decoding, we need explicit codes along with efficient list decoding algorithms that can achieve the list decoding capacity. For this survey, a list decoding algorithm with a polynomial running time is considered to be efficient. (Note that this puts an *a priori* requirement that the worst case list size needs to be bounded by a polynomial in the block length of the code.) Even though the notion of list decoding was defined in the late 1950's, there was essentially no algorithmic progress in list decoding till the breakthrough works of Sudan [14] and Guruswami-Sudan [11] which can list decode Reed-Solomon codes up to the trade-off $\rho_{GS}(R) = 1-\sqrt{R}$. One can check that $\rho_{GS}(R) > \rho_U(R)$ for every rate R (with the gains being more pronounced for smaller rates). This fact lead to a spurt of research activity in list decoding including some surprising applications outside the traditional coding domain: see for example [15], [4, Chap. 12]. However, this result failed to achieve the list decoding capacity for *any* rate (with the gap being especially pronounced for larger rates).

The bound of ρ_{GS} resisted improvements for about seven years till in a recent breakthrough paper [12], Parvaresh and Vardy presented codes that are list-decodable beyond the $1-\sqrt{R}$ radius for low rates R. For any $m \geqslant 1$, they achieve the list-decoding radius $\rho_{\mathrm{PV}}^{(m)}(R) = 1 - \sqrt[m+1]{m^m R^m}$. For rates $R \to 0$, choosing m large enough, they can list decode up to radius $1 - O(R\log(1/R))$, which approaches the capacity $1-R$. However, for $R \geqslant 1/16$, the best choice of m is in fact $m = 1$, which reverts back to RS codes and the list-decoding radius $1-\sqrt{R}$. Building on works of Parvaresh and Vardy [12], Guruswami and Rudra [8] present codes that get arbitrarily close to the list decoding capacity $\rho_{\mathrm{cap}}(R)$ for every rate. In particular, for every $1 > R > 0$ and every $\varepsilon > 0$, they give *explicit* codes of rate R together with polynomial time list decoding algorithm that can correct up to a fraction $1-R-\varepsilon$ of errors. These are the first explicit codes (with efficient list decoding algorithms) that get arbitrarily close to the list decoding capacity for *any* rate. This article surveys the results of [12,8] and some of their implications for list decoding of explicit codes over small alphabets.

2 Folded Reed-Solomon Codes and the Main Results

The codes used in [8] are simple to state. They are obtained from the Reed-Solomon code by careful bundling together of codeword symbols (and hence, are called *folded Reed-Solomon codes*). We remark that the folded RS codes are a *special* case of the codes studied by [12]. However, for the ease of presentation, we will present all the results in terms of folded Reed-Solomon codes: this would be sufficient to highlight the algorithmic techniques used in [12]. See the survey [5] in these proceedings for a more detailed description of the Parvaresh-Vardy codes.

Consider a Reed-Solomon (RS) code $C = \mathsf{RS}_{\mathbb{F},\mathbb{F}^*}[n,k]$ consisting of evaluations of degree k polynomials over some finite field $\mathbb{F}$ at the set $\mathbb{F}^*$ of nonzero elements of $\mathbb{F}$. Let $q = |\mathbb{F}| = n+1$. Let γ be a generator of the multiplicative group $\mathbb{F}^*$, and let the evaluation points be ordered as $1, \gamma, \gamma^2, \ldots, \gamma^{n-1}$. Using all nonzero field elements as evaluation points is one of the most commonly used instantiations of Reed-Solomon codes.

Let $m \geqslant 1$ be an integer parameter called the *folding parameter*. For ease of presentation, it will assumed that m divides $n = q - 1$.

Definition 1 (Folded Reed-Solomon Code). *The m-folded version of the RS code C, denoted $\mathrm{FRS}_{\mathbb{F},\gamma,m,k}$, is a code of block length $N = n/m$ over $\mathbb{F}^m$. The encoding of a message $f(X)$, a polynomial over $\mathbb{F}$ of degree at most k, has as its j'th symbol, for $0 \leqslant j < n/m$, the m-tuple $(f(\gamma^{jm}), f(\gamma^{jm+1}), \cdots, f(\gamma^{jm+m-1}))$. In other words, the codewords of $C' = \mathrm{FRS}_{\mathbb{F},\gamma,m,k}$ are in one-one correspondence with those of the RS code C and are obtained by bundling together consecutive m-tuple of symbols in codewords of C.*

The following is the main result of Guruswami and Rudra.

Theorem 1 ([8]). *For every $\varepsilon > 0$ and $0 < R < 1$, there is a family of folded Reed-Solomon codes that have rate at least R and which can be list decoded up to a fraction $1 - R - \varepsilon$ of errors in time (and outputs a list of size at most) $(N/\varepsilon^2)^{O(\varepsilon^{-1}\log(1/R))}$ where N is the block length of the code. The alphabet size of the code as a function of the block length N is $(N/\varepsilon^2)^{O(1/\varepsilon^2)}$.*

The result of [8] also works in a more general setting called *list recovery*, which is defined next.

Definition 2 (List Recovery). *A code $C \subseteq \Sigma^n$ is said to be (ζ, l, L)-list recoverable if for every sequence of sets $S_1, \ldots, S_n$ where each $S_i \subseteq \Sigma$ has at most l elements, the number of codewords $c \in C$ for which $c_i \in S_i$ for at least ζn positions $i \in \{1, 2, \ldots, n\}$ is at most L.*

A code $C \subseteq \Sigma^n$ is said to (ζ, l)-list recoverable in polynomial time if it is $(\zeta, l, L(n))$-list recoverable for some polynomially bounded function $L(\cdot)$, and moreover there is a polynomial time algorithm to find the at most $L(n)$ codewords that are solutions to any $(\zeta, l, L(n))$-list recovery instance.

Note that when $l = 1$, $(\zeta, 1, \cdot)$-list recovery is the same as list decoding up to a $(1 - \zeta)$ fraction of errors. Guruswami and Rudra have the following result for list recovery.

Theorem 2 ([8]). *For every integer $l \geqslant 1$, for all R, $0 < R < 1$ and $\varepsilon > 0$, and for every prime p, there is an* explicit *family of folded Reed-Solomon codes over fields of characteristic p that have rate at least R and which can be $(R+\varepsilon, l)$-list recovered in polynomial time. The alphabet size of a code of block length N in the family is $(N/\varepsilon^2)^{O(\varepsilon^{-2}\log l/(1-R))}$.*

Theorem 2 will be put to good use in Section 4.

3 Informal Description of the Algorithms

In this section, we will give an overview of the list decoding algorithms that are needed to prove Theorem 1. Along the way we will encounter the main algorithmic techniques used in [14,11,12]. We start by stating more precisely the problem that needs to be solved for Theorem 1. We need list-decoding algorithms for the folded Reed-Solomon code $\mathrm{FRS}_{\mathbb{F}_q,\gamma,m,k}$ of rate R. More precisely, for every $1 \leqslant s \leqslant m$ and $\delta > 0$, given a received word $\mathbf{y} = \langle (y_0, \ldots, y_{m-1}), \ldots, (y_{n-m}, \ldots, y_{n-1}) \rangle$ (where recall $n = q-1$), we want to output all codewords in $\mathrm{FRS}_{\mathbb{F}_q,\gamma,m,k}$ that disagree with $\mathbf{y}$ in at most $1 - (1+\delta)\left(\frac{mR}{m-s+1}\right)^{s/(s+1)}$ fraction of positions in polynomial time. In other words, we need to output all degree k polynomials $f(X)$ such that for at least $(1+\delta)\left(\frac{mR}{m-s+1}\right)^{s/(s+1)}$ fraction of $0 \leqslant i \leqslant n/m - 1$, $f(\gamma^{im+j}) = y_{im+j}$ (for every $0 \leqslant j \leqslant m-1$). By picking the parameters m, s and δ carefully, we will get folded Reed-Solomon codes of rate R that can be list decoded up to a $1 - R - \varepsilon$ fraction of errors (for any $\varepsilon > 0$). We will now present the main ideas needed to design the required list-decoding algorithm.

For the ease of presentation we will start with the case when $s = m$. As a warm up, let us consider the case when $s = m = 1$. Note that for $m = 1$, we are interested in list decoding Reed-Solomon codes. More precisely, given the received word $\mathbf{y} = \langle y_0, \ldots, y_{n-1} \rangle$, we are interested in all degree k polynomials $f(X)$ such that for at least $(1+\delta)\sqrt{R}$ fraction of positions $0 \leqslant i \leqslant n-1$, $f(\gamma^i) = y_i$. We now sketch the main ideas of the algorithms in [14,11]. The algorithms have two main steps: the first is an *interpolation* step and the second one is a *root finding* step. In the interpolation step, the list-decoding algorithm finds a bivariate polynomial $Q(X, Y)$ that *fits* the input. That is,

> for every position i, $Q(\gamma^i, y_i) = 0$.

Such a polynomial $Q(\cdot, \cdot)$ can be found in polynomial time if we search for one with large enough total degree (this amounts to solving a system of linear equations). After the interpolation step, the root finding step finds all factors of $Q(X, Y)$ of the form $Y - f(X)$. The crux of the analysis is to show that

> for every degree k polynomial $f(X)$ that satisfies $f(\gamma^i) = y_i$ for at least $(1+\delta)\sqrt{R}$ fraction of positions i, $Y - f(X)$ is indeed a factor of $Q(X, Y)$.

However, the above is not true for every bivariate polynomial $Q(X,Y)$ that satisfies $Q(\gamma^i, y_i) = 0$ for all positions i. The main ideas in [14,11] were to introduce more constraints on $Q(X,Y)$. In particular, the work of Sudan [14] added the constraint that a certain weighted degree of $Q(X,Y)$ is below a fixed upper bound. Specifically, $Q(X,Y)$ was restricted to have a non-trivially bounded $(1,k)$-weighted degree. The $(1,k)$-weighted degree of a monomial X^iY^j is $i+jk$ and the $(1,k)$-weighted degree of a bivariate polynomial $Q(X,Y)$ is the maximum $(1,k)$-weighted degree among its monomials. The intuition behind defining such a weighted degree is that given $Q(X,Y)$ with weighted $(1,k)$ degree of D, the *univariate* polynomial $Q(X, f(X))$, where $f(X)$ is some degree k polynomial, has total degree at most D. The upper bound D is chosen carefully such that if $f(X)$ is a codeword that needs to be output, then $Q(X, f(X))$ has more than D zeroes and thus $Q(X, f(X)) \equiv 0$, which in turn implies that $Y - f(X)$ divides $Q(X,Y)$. To get to the bound of $1-(1+\delta)\sqrt{R}$, Guruswami and Sudan in [11], added a further constraint on $Q(X,Y)$ that requires it to have r roots at (γ^i, y_i), where r is some parameter (in [14] $r = 1$ while in [11], r is roughly $1/\delta$).

We now consider the next non-trivial case of $m = s = 2$ (the ideas for this case can be easily generalized for the general $m = s$ case). Note that now given the received word $\langle (y_0, y_1), (y_2, y_3), \ldots, (y_{n-2}, y_{n-1}) \rangle$ we want to find all degree k polynomials $f(X)$ such that for at least $(1+\delta)\sqrt[3]{2R^2}$ fraction of positions $0 \leqslant i \leqslant n/2 - 1$, $f(\gamma^{2i}) = y_{2i}$ and $f(\gamma^{2i+1}) = y_{2i+1}$. As in the previous case, we will have an interpolation and a root finding step. The interpolation step is a straightforward generalization of $m = 1$ case: we find a trivariate polynomial $Q(X,Y,Z)$ that fits the received word, that is, for every $0 \leqslant i \leqslant n/2 - 1$, $Q(\gamma^{2i}, y_{2i}, y_{2i+1}) = 0$. Further, $Q(X,Y,Z)$ has an upper bound on its $(1,k,k)$-weighted degree (which is a straightforward generalization of the $(1,k)$-weighted degree for the bivariate case) and has a multiplicity of r at every point. For the root finding step, it suffices to show that for every degree k polynomial $f(X)$ that needs to be output $Q(X, f(X), f(\gamma X)) \equiv 0$. This, however does not follow from weighted degree and multiple root properties of $Q(X,Y,Z)$. Here we will need two new ideas, the first of which is to show that for some irreducible polynomial $E(X)$ of degree $q-1$, $f(X)^q \equiv f(\gamma X) \mod (E(X))$ [8]. The second idea, due to Parvaresh and Vardy [12], is the following. We first obtain the bivariate polynomial (over an appropriate extension field) $T(Y,Z) \equiv Q(X,Y,Z) \mod (E(X))$. Note that by the first idea, we are looking for solutions on the curve $Z = Y^q$ (Y corresponds to $f(X)$ and Z corresponds to $f(\gamma X)$ in the extension field). The crux of the argument is to show that all the polynomials $f(X)$ that need to be output correspond to (in the extension field) some root of the equation $T(Y, Y^q) = 0$.

As was mentioned earlier, the extension of the $m = s = 2$ case to the general $m = s > 2$ case is fairly straightforward. To go from $s = m$ to any $s \leqslant m$ requires another simple idea from [8]: We will reduce the problem of list decoding folded Reed-Solomon code with folding parameter m to the problem of list decoding folded Reed-Solomon code with folding parameter s. We then use the algorithm outlined in the previous paragraph for the folded Reed-Solomon code with folding

parameter s. A careful tracking of the agreement parameter in the reduction, brings down the final agreement fraction (that is required for the original folded Reed-Solomon code with folding parameter m) from $(1+\delta)\sqrt[m+1]{mR^m}$ (which can be obtained without the reduction and is the bound achieved by [12]) to $(1+\delta)\sqrt[s+1]{\left(\frac{m}{m-s+1}\right)R^s}$.

4 Codes over Small Alphabets

To get within ε of capacity, the codes in Theorem 1 have alphabet size $N^{\Omega(1/\varepsilon^2)}$ where N is the block length. This leads to the following natural questions:

1. Can we achieve the list decoding capacity for smaller alphabets, say for $2^{\Omega(1/\varepsilon)}$ (for which the list decoding capacity as we saw in the introduction is $1-R$)?
2. Can we achieve list decoding capacity for codes over fixed alphabet sizes, for example, binary codes?

The best known answers to both of the questions above use the notion of *code concatenation* and Theorem 2. We now digress for a bit to talk about concatenated codes (and along the way motivate why list recovery is an important algorithmic task).

Concatenated codes were defined in the seminal thesis of Forney [3]. Concatenated codes are constructed from two different codes that are defined over alphabets of different sizes. Say we are interested in a code over $[q] \stackrel{def}{=} \{0,1,\ldots,q-1\}$ (in this section, we will think of $q \geqslant 2$ as being a fixed constant). Then the *outer code* C_{out} is defined over $[Q]$, where $Q = q^k$ for some positive integer k. The second code, called the *inner code* is defined over $[q]$ and is of dimension k (Note that the message space of C_{in} and the alphabet of C_{out} have the same size). The concatenated code, denoted by $C = C_{out} \circ C_{in}$, is defined as follows. Let the rate of C_{out} be R and let the block lengths of C_{out} and C_{in} be N and n respectively. Define $K = RN$ and $r = k/n$. The input to C is a vector $\mathbf{m} = \langle m_1, \ldots, m_K \rangle \in ([q]^k)^K$. Let $C_{out}(\mathbf{m}) = \langle x_1, \ldots, x_N \rangle$. The codeword in C corresponding to $\mathbf{m}$ is defined as follows

$$C(\mathbf{m}) = \langle C_{in}(x_1), C_{in}(x_2), \ldots, C_{in}(x_N) \rangle.$$

It is easy to check that C has rate rR, dimension kK and block length nN.

Notice that to construct a q-ary code C we use another q-ary code C_{in}. However, the nice thing about C_{in} is that it has small block length. In particular, since R and r are constants (and typically Q and N are polynomially related), $n = O(\log N)$. This implies that we can use up exponential time (in n) to search for a "good" inner code. Further, one can use the brute force algorithm to (list) decode C_{in}.

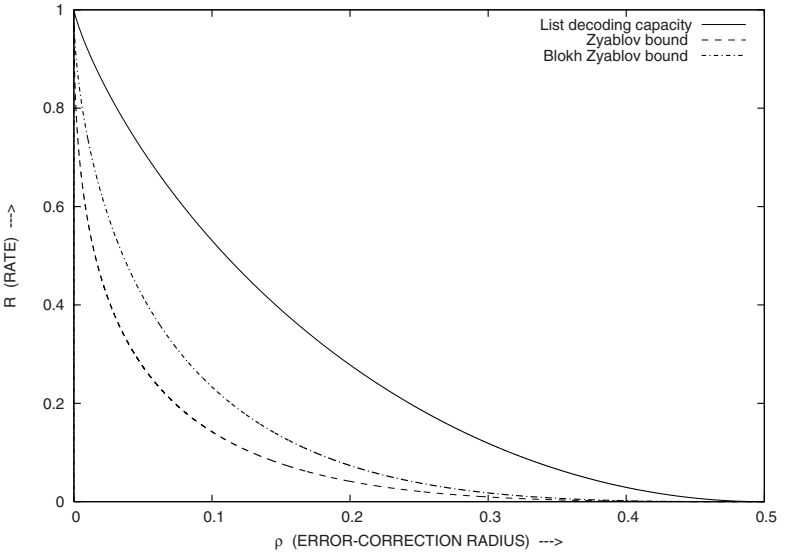


Fig. 1. Rate R of binary codes from [8,9] plotted against the list-decoding radius ρ of their respective algorithms. The best possible trade-off, i.e., list-decoding capacity, $\rho = H_2^{-1}(1-R)$ is also plotted.

Finally, we motivate why we are interested in list recovery. Consider the following natural decoding algorithm for the concatenated code $C_{out} \circ C_{in}$. Given a received word in $([q]^n)^N$, we divide it into N blocks from $[q]^n$. Then we use a decoding algorithm for C_{in} to get an intermediate received word to feed into a decoding algorithm for C_{out}. Now one can use unique decoding for C_{in} and list decoding for C_{out}. However, this loses information in the first step. Instead, one can use the brute force list-decoding algorithm for C_{in} to get a sequence of lists (each of which is a subset of $[Q]$). Now we use a list-recovery algorithm for C_{out} to get the final list of codewords.

By concatenating folded RS codes of rate close to 1 (that are list recoverable by Theorem 2) with suitable inner codes followed by redistribution of symbols using an expander graph (similar to a construction for linear-time unique decodable codes in [6]), one can get within ε of capacity with codes over an alphabet of size $2^{O(\varepsilon^{-4}\log(1/\varepsilon))}$ [8].

For binary codes, recall that the list decoding capacity is known to be $\rho_{\text{bin}}(R) = H_2^{-1}(1-R)$. No explicit constructions of binary codes that approach this capacity are known. However, concatenating the Folded RS codes with suitably chosen inner codes, one can obtain polynomial time constructable binary codes that can be list decoded up to the so called "Zyablov bound" [8]. Using a generalization of code concatenation to *multilevel code concatenation*, one can achieve codes that can be list decoded up to the so called "Blokh-Zyablov" bound [9]. See Figure 1 for a pictorial comparison of the different bounds.

5 Concluding Remarks

The results in [8] could be improved with respect to some parameters. The size of the list needed to perform list decoding to a radius that is within ε of capacity grows as $N^{O(\varepsilon^{-1}\log(1/R))}$ where N and R are the block length and the rate of the code respectively. It remains an open question to bring this list size down to a constant independent of N (recall that the existential random coding arguments work with a list size of $O(1/\varepsilon)$). The alphabet size needed to approach capacity was shown to be a constant independent of N. However, this involved a brute-force search for a rather large (inner) code, which translates to a construction time of about $N^{O(\varepsilon^{-2}\log(1/\varepsilon))}$ (instead of the ideal construction time where the exponent of N does not depend on ε). Obtaining a "direct" algebraic construction over a constant-sized alphabet, such as the generalization of the Parvaresh-Vardy framework to algebraic-geometric codes in [7], might help in addressing these two issues.

Finally, constructing binary codes (or q-ary codes for some fixed, small value of q) that approach the respective list decoding capacity remains a challenging open problem. In recent work [10], it has been shown that there *exist* q-ary linear concatenated codes that achieve list decoding capacity (in the sense that every Hamming ball of radius $H_q^{-1}(1-R-\varepsilon)$ has polynomially many codewords, where R is the rate). In particular, this results holds when the outer code is a folded RS code. This is somewhat encouraging news since concatenation has been the preeminent method to construct good list-decodable codes over small alphabets. But realizing the full potential of concatenated codes and achieving capacity (or even substantially improving upon the Blokh-Zyablov bound) with explicit codes and polynomial time decoding remains a huge challenge.

References

1. Elias, P.: List Decoding for Noisy Channels. Technical Report 335, Research Laboratory of Electronics, MIT (1957)
2. Elias, P.: Error-Correcting Codes for List Decoding. IEEE Trans. Inform. Theory 37(5), 5–12 (1991)
3. Forney, G.D.: Concatenated Codes. MIT Press, Cambridge, MA (1966)
4. Guruswami, V.: List Decoding of Error-Correcting Codes. LNCS, vol. 3282. Springer, Heidelberg (2004)
5. Guruswami, V.: List Decoding and Pseudorandom Constructions. In: Boztaş, S., Lu, H.F. (eds.) AAECC 2007. LNCS, vol. 4851, Springer, Heidelberg (2007)
6. Guruswami, V., Indyk, P.: Linear-Time Encodable/Decodable Codes with Near-Optimal Rate. IEEE Trans. Inform. Theory 51(10), 3393–3400 (2005)
7. Guruswami, V., Patthak, A.: Correlated Algebraic-Geometric Codes: Improved List Decoding over Bounded Alphabets. In: FOCS 2006, pp. 227–236 (2006)
8. Guruswami, V., Rudra, A.: Explicit Capacity-Achieving List-Decodable Codes. In: 38th Annual ACM Symposium on Theory of Computing, pp. 1–10 (2006)
9. Guruswami, V., Rudra, A.: Better Binary List-Decodable Codes Via Multilevel Concatenation. In: 11th International Workshop on Randomization and Computation. pp. 554–568 (2007)

10. Guruswami, V., Rudra, A.: Concatenated Codes Can Achieve List Decoding Capacity. In: 19th Annual ACM-SIAM Symposium on Discrete Algorithms (to appear, 2008)
11. Guruswami, V., Sudan, M.: Improved Decoding of Reed-Solomon and Algebraic-Geometric Codes. IEEE Trans. Inform. Theory 45, 1757–1767 (1999)
12. Parvaresh, F., Vardy, A.: Correcting Errors Beyond the Guruswami-Sudan Radius in Polynomial Time. In: 46th Annual IEEE Symposium on Foundations of Computer Science. pp. 285–294 (2005)
13. Rudra, A.: List Decoding and Property Testing of Error Correcting Codes. PhD thesis, University of Washington (2007)
14. Sudan, M.: Decoding of Reed-Solomon Codes Beyond the Error-Correction Bound. J. Complexity 13(1), 180–193 (1997)
15. Sudan, M.: List Decoding: Algorithms and Applications. SIGACT News 31, 16–27 (2000)
16. Wozencraft, J.M.: List Decoding. Quarterly Progress Report, Research Laboratory of Electronics. MIT 48, 90–95 (1958)
17. Zyablov, V.V., Pinsker, M.S.: List Cascade Decoding. Problems of Information Transmission 17(4), 29–34 (1981)

This page intentionally blank

Algebraic Structure Theory of Tail-Biting Trellises

Priti Shankar

Department of Computer Science and Automation
Indian Institute of Science
Bangalore, India 560012
priti@csa.iisc.ernet.in

It is well known that there is an intimate connection between algebraic descriptions of linear block codes in the form of generator or parity-check matrices, and combinatorial descriptions in the form of trellises. A conventional trellis for a linear code C is a directed labelled layered graph with unique start and final nodes, and all paths from the start to the final node spell out codewords. The trellis can be thought of as being laid out on a linear time axis. There is a rich theory of conventional trellises for linear block codes. Every linear block code has a unique minimal trellis, and several seemingly different constructions proposed, all yield this minimal trellis, which simultaneously minimizes all measures of trellis complexity. Tail-biting trellises are defined on circular time axes, and the underlying theory is a little more involved as there is no unique minimal trellis. Interestingly, the complexity of a tail-biting trellis can be much lower than that of the best possible conventional trellis. We extend the well-known BCJR construction for conventional trellises to linear tail-biting trellises, introducing the notion of a *displacement* matrix. This implicitly induces a coset decomposition of the code. The BCJR-like labeling scheme yields a very simple specification for the tail-biting trellis for the dual code, with the dual trellis having the same state-complexity profile as that of the primal code . We also show that the algebraic specification of Forney for state spaces of conventional trellises has a natural extension to tail-biting trellises. Finally we provide an automata-theoretic view of trellises and display some connections between well known results in finite automata and trellis theory.

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, p. 47, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Nice Codes from Nice Curves

Henning Stichtenoth

Sabancı University - FENS
Orhanli - Tuzla 34956 Istanbul, Turkey
henning@sabanciuniv.edu

The well-known Tsfasman-Vladut-Zink (TVZ) theorem states that for all prime powers $q = \ell^2 \geq 49$ there exist sequences of linear codes over $\mathbb{F}_q$ with increasing length whose limit parameters R and δ (rate and relative minimum distance) are better than the Gilbert-Varshamov bound. The basic ingredients in the proof of the TVZ theorem are sequences of modular curves (or their corresponding function fields) having many rational points in comparison to their genus (more precisely, these curves attain the so-called Drinfeld-Vladut bound). Starting with such a sequence of curves and using Goppa's construction of algebraic geometry (AG) codes, one easily obtains sequences of linear codes whose limit parameters beat the Gilbert-Varshamov bound.

However, this construction yields just *linear* codes, and the question arises if one can refine the construction to obtain good long codes with additional nice properties (e.g., codes with *many automorphisms*, *self-orthogonal* codes or *self-dual* codes). This can be done. We give a brief outline of some results in this direction.

Our starting point is the sequence of function fields $(F_i)_{i\geq 0}$ over $\mathbb{F}_q$ which are defined as

$$F_i = \mathbb{F}_q(x_0, x_1, ..., x_n) \quad \text{with the relation} \quad x_{i+1}^{\ell} - x_{i+1} = \frac{x_i^{\ell}}{1 - x_i^{\ell-1}}$$

for all $i \geq 0$. It is known that the curves corresponding to these function fields have many rational points; in fact they attain the Drinfeld-Vladut bound. The idea is now to replace the fields F_i by their Galois closure over some basis field (it is well-known in algebra that Galois extensions of fields have often much nicer properties than "ordinary" extensions).

We proceed as follows: we fix the element $u := (x_0^{\ell} - x_0)^{\ell-1} \in F_0 = \mathbb{F}_q(x_0)$ and consider the fields

$$E_i := \text{Galois closure of } F_i \text{ over } \mathbb{F}_q(u),\ i = 0, 1, 2, ...$$

This sequence $(E_i)_{i\geq 0}$ has particularly nice properties, e.g.

- all extensions $E_i/\mathbb{F}_q(u)$ are Galois,
- the corresponding curves attain the Drinfeld-Vladut bound,
- the Galois groups operate transitively on a large number of rational points,
- only 2 points of $\mathbb{F}_q(u)$ are ramified in E_i.

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 48–49, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Using these (and some other) properties of the function fields E_i, one can then construct AG codes in the usual manner and obtains:

Theorem 1. *The following classes of linear codes over $\mathbb{F}_q$ are better than the Gilbert-Varshamov bound for all $q = \ell^2$ with $\ell \geq 7$:*

1. *self-orthogonal codes,*
2. *self-dual codes,*
3. *transitive codes.*

Here a *transitive* code means one, whose automorphism group acts transitively on the coordinates. Note however that we cannot construct asymptotically good *cyclic* codes in this way (cyclic codes are a subclass of transitive codes).

The above theorem works over *quadratic* fields $\mathbb{F}_q$ (i.e., $q = \ell^2$). If one starts with a similar sequence of function fields over a *cubic* field $\mathbb{F}_q$ (i.e., $q = \ell^3$) one can prove an analogous result.

References

1. Stichtenoth, H.: Transitive and Self-Dual Codes Attaining the Tsfasman-Vladut-Zink Bound. IEEE Trans. Inform. Theory 52, 2218–2224 (2006)
2. Bassa, A., Garcia, A., Stichtenoth, H.: A New Tower over Cubic Finite Fields (preprint, 2007)
3. Bassa, A., Stichtenoth, H.: Asymptotic Bounds for Transitive and Self-Dual Codes over Cubic Finite Fields (in preparation, 2007)

Generalized Sudan's List Decoding for Order Domain Codes*

Olav Geil[1] and Ryutaroh Matsumoto[2]

[1] Department of Mathematical Sciences, Aalborg University, Denmark
olav@math.aau.dk
[2] Department of Communications and Integrated Systems, Tokyo Institute of Technology, Japan
ryutaroh@rmatsumoto.org

Abstract. We generalize Sudan's list decoding algorithm without multiplicity to evaluation codes coming from arbitrary order domains. The number of correctable errors by the proposed method is larger than the original list decoding without multiplicity.

1 Introduction

Høholdt et al. [6] proposed the new framework for algebraic code construction, which they called *evaluation codes*. Evaluation codes are defined by either generator matrices or parity check matrices. Evaluation codes defined by parity check matrices include many classes of algebraic codes, including generalized Reed-Muller, Reed-Solomon, and one-point geometric Goppa codes $C_\Omega(D, G)$, and they provided lower bounds on the minimum Hamming distance and decoding algorithms in a unified manner, while relatively little work was done for evaluation codes defined by generator matrices in [6]. The framework of evaluation codes and order domains was later generalized by O'Sullivan [7], Geil and Pellikaan [3].

Andersen and Geil [1] studied the evaluation codes defined by generator matrices, which also include generalized Reed-Muller, Reed-Solomon, and one-point geometric Goppa codes $C_\mathcal{L}(D, G)$, and they also provided lower bounds on the minimum Hamming distance in a unified manner. Their work [1] can be regarded as a generator matrix counterpart of [6]. In this paper we study evaluation codes defined by generator matrices.

On the other hand, Sudan [10] and Guruswami-Sudan [5] proposed the list decoding algorithms for Reed-Solomon and one-point geometric Goppa codes, and the latter method dramatically increased the number of correctable errors of the conventional bounded distance decoding algorithm, such as the Berlekamp-Massey algorithm. Following those work, Shokrollahi and Wasserman [9] generalized the Sudan method [10] to one-point geometric Goppa codes, and Pellikaan

* This research is in part supported by the Danish National Science Research Council Grant FNV-21040368 and the MEXT 21st Century COE Program: Photonics Nanodevice Integration Engineering.

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 50–59, 2007.
© Springer-Verlag Berlin Heidelberg 2007

and Wu [8] generalized the Guruswami-Sudan method [5] to generalized Reed-Muller codes as the first algorithm among three new list decoding algorithms in [8]. Augot and Stepanov [2] improved the estimation of error-correcting capability of the first algorithm in [8].

However, up to now, nobody has successfully generalized the list decoding algorithms [10,5] to evaluation codes from arbitrary order domains. The difficulty lies in the fact that existing methods [10,5,8] deal with codes coming from polynomial rings or their factor rings and utilize their polynomial structure such as the degree of a polynomial and the pole order of an algebraic function.

We will distill essential ingredients from Sudan's original decoding method [10], which allow us to carry over it to evaluation codes from arbitrary order domains. After that, we examine the error-correcting capability of the proposed generalization when we apply it to generalized Reed-Muller and one-point geometric Goppa codes, and show that the proposed method can correct more errors than [9] and the first algorithm in [8]. We have to note that the proposed method usually cannot correct more errors than the Guruswami-Sudan method [5] with multiplicity.

The paper is organized as follows. In Section 2 we present the modified Sudan decoding algorithm without multiplicity. Our description does not require that the reader has any previous experience with order domains. Some knowledge about generalized Reed-Muller and one-point geometric Goppa codes should do. In Section 3 we study decoding of generalized Reed-Muller codes. We compare our findings to the results by the first algorithm of Pellikaan and Wu in [8] and by Augot and Stepanov in [2]. Then in Section 4 we apply our method to some codes coming from norm-trace curves.

2 Decoding of Order Domain Codes

In this section we state the modified decoding algorithm for a large family of codes defined from order domains. We provide translations into the case of generalized Reed-Muller codes and one-point geometric Goppa codes. Our presentation relies on [1,3,7].

Definition 1. *Let R be an $\mathbb{F}_q$-algebra and let Γ be a subsemigroup of $\mathbb{N}_0^r$ for some r. Let $\prec_{\mathbb{N}_0^r}$ be a monomial ordering on $\mathbb{N}_0^r$. A surjective map $\rho : R \to \Gamma_{-\infty} := \Gamma \cup \{-\infty\}$ that satisfies the following six conditions is said to be a weight function*

(*W*.0) $\rho(f) = -\infty$ *if and only if* $f = 0$
(*W*.1) $\rho(af) = \rho(f)$ *for all nonzero* $a \in \mathbb{F}_q$
(*W*.2) $\rho(f+g) \preceq_{\mathbb{N}_0^r} \max\{\rho(f), \rho(g)\}$ *and equality holds when* $\rho(f) \prec_{\mathbb{N}_0^r} \rho(g)$
(*W*.3) *If* $\rho(f) \prec_{\mathbb{N}_0^r} \rho(g)$ *and* $h \neq 0$, *then* $\rho(fh) \prec_{\mathbb{N}_0^r} \rho(gh)$
(*W*.4) *If f and g are nonzero and* $\rho(f) = \rho(g)$, *then there exists a nonzero* $a \in \mathbb{F}_q$ *such that* $\rho(f - ag) \prec_{\mathbb{N}_0^r} \rho(g)$
(*W*.5) *If f and g are nonzero then* $\rho(fg) = \rho(f) + \rho(g)$.

An $\mathbb{F}_q$-algebra with a weight function is called an order domain over $\mathbb{F}_q$. The triple (R, ρ, Γ) is called an order structure and Γ is called the value semigroup of ρ.

We have the following two standard examples of weight functions.

Example 1. Consider the polynomial ring $R = \mathbb{F}_q[X_1, \ldots, X_m]$ and let $\prec_{\mathbb{N}_0^m}$ be the graded lexicographic ordering on $\mathbb{N}_0^m$ given by $(i_1, \ldots, i_m) \prec_{\mathbb{N}_0^m} (j_1, \ldots, j_m)$ if either $i_1 + \cdots + i_m < j_1 + \cdots + j_m$ holds or $i_1 + \cdots + i_m = j_1 + \cdots + j_m$ holds, but left most non-zero entry of $j_1 - i_1, \ldots, j_m - i_m)$ is positive. The map $\rho : R \to \mathbb{N}_0^m \cup \{-\infty\}$, $\rho(F) := \max_{\prec_{\mathbb{N}_0^m}} \{(i_1, \ldots, i_m) \mid X_1^{i_1} \cdots X_m^{i_m} \in \mathrm{Supp}(F)\}$ if $F \neq 0$ and $\rho(0) := -\infty$ is a weight function.

Example 2. Let Q be a rational place of a function field in one variable over $\mathbb{F}_q$. Then $R = \cup_{m=0}^{\infty} \mathcal{L}(mQ)$ is an order domain with a weight function given by $\rho(f) = -\nu_Q(f)$. Clearly, in this case the value semigroup Γ is simply the Weierstrass semigroup corresponding to Q and the monomial ordering is the unique monomial ordering on $\mathbb{N}_0$.

For the code construction we will need a few results.

Theorem 1. *Let (R, ρ, Γ) be an order structure. Then any set $\mathcal{B} = \{f_\gamma \mid \rho(f_\gamma) = \gamma\}_{\gamma \in \Gamma}$ constitutes a basis for R as a vector space over $\mathbb{F}_q$. In particular $\{f_\lambda \in \mathcal{B} \mid \lambda \preceq \gamma\}$ constitutes a basis for $R_\gamma := \{f \in R \mid \rho(f) \preceq \gamma\}$.*

A basis as in Theorem 1 is known in the literature as a well-behaving basis. In the remaining part of this section we will always assume that some fixed well-behaving basis has been chosen for the order domain under consideration.

Definition 2. *Let R be an $\mathbb{F}_q$-algebra. A surjective map $\varphi : R \to \mathbb{F}_q^n$ is called a morphism of $\mathbb{F}_q$-algebras if φ is $\mathbb{F}_q$-linear and $\varphi(fg) = \varphi(f) * \varphi(g)$ for all $f, g \in R$, where $*$ denotes the componentwise multiplication of two vectors.*

The class of codes $E(\lambda)$ below includes as we shall recall generalized Reed-Muller codes as well as one-point geometric Goppa codes.

Definition 3. *Consider an order domain R over $\mathbb{F}_q$ and a corresponding morphism $\varphi : R \to \mathbb{F}_q^n$. For $\lambda \in \Gamma$ we define $E(\lambda) := \varphi(R_\lambda)$.*

Example 3. This is a continuation of Example 1. Consider $\mathbb{F}_q^{q^m} = \{P_1, \ldots, P_{q^m}\}$ and let $\varphi : \mathbb{F}_q[X_1, \ldots, X_m] \to \mathbb{F}_q^{q^m}$ be given by $\varphi(F) = (F(P_1) \ldots, F(P_{q^m}))$. If we choose $\lambda = (u, 0, \ldots, 0)$ then $E(\lambda)$ is simply the generalized Reed-Muller code $\mathrm{RM}_q(u, m)$ no matter how the well-behaving basis for the order domain $R = \mathbb{F}_q[X_1, \ldots, X_m]$ has been chosen. For simplicity we choose in this paper always the well-behaving basis $\mathcal{B}$ to be the set of monomials in $X_1, \ldots, X_m$.

Example 4. This is a continuation of Example 2. Let $\{P_1, \ldots, P_n\}$ be rational places different from Q and consider the morphism $\varphi : R \to \mathbb{F}_q^n$ given by $\varphi(f) = (f(P_1), \ldots, f(P_n))$. The code $E(\lambda)$ is the one-point geometric Goppa code $C_{\mathcal{L}}(D, \lambda Q)$ where $D = P_1 + \cdots + P_n$.

We next consider some terminology from [1].

Definition 4. *Let* $\alpha(1) := 0$ *and define for* $i = 2, 3, \ldots, n$ *recursively* $\alpha(i)$ *to be the smallest element in* Γ *that is greater than* $\alpha(1), \alpha(2), \ldots, \alpha(i-1)$ *and satisfies* $\varphi(R_\gamma) \subsetneq \varphi(R_{\alpha(i)})$ *for all* $\gamma < \alpha(i)$. *Write* $\Delta(R, \rho, \varphi) = \{\alpha(1), \alpha(2), \ldots, \alpha(n)\}$.

Definition 5. *For* $\eta \in \Delta(R, \rho, \varphi) = \{\alpha(1), \alpha(2), \ldots, \alpha(n)\}$ *define*

$$M(\eta) := (\eta + \Gamma) \cap \Delta(R, \rho, \varphi)$$

where $\eta + \Gamma$ *means* $\{\eta + \lambda \mid \lambda \in \Gamma\}$. *Let* $\sigma(\eta) := \#M(\eta)$.

The first part of the following theorem plays a fundamental role in our modification of the Sudan decoding algorithm without multiplicity.

Theorem 2. *If* $\boldsymbol{c} \in E(\lambda)$ *but* $\boldsymbol{c} \notin E(\eta)$ *for any* η *with* $\eta \prec_{\mathbb{N}_0^r} \lambda$ *then* $w_H(\boldsymbol{c}) \geq \sigma(\lambda)$ *holds. In particular we have* $d(E(\lambda)) \geq \min\{\sigma(\eta) \mid \eta \in \Delta(R, \rho, \varphi), \eta \preceq \lambda\}$.

Example 5. The above bound gives the true minimum distances of generalized Reed-Muller codes and of Hermitian codes. For the case of one-point geometric Goppa codes the bound is an improvement to the usual bound by Goppa which states that the minimum distance of a one-point geometric Goppa code $C_{\mathcal{L}}(D, \lambda Q)$ is at least $n - \lambda$. More precisely, we have $\sigma(\lambda) \geq n - \lambda$ for any $\lambda \in \Delta(R, \rho, \varphi)$. For high dimensions the inequality is in general sharp.

Theorem 2 suggests the following improved code construction.

Definition 6. *Given any fixed basis* $\mathcal{B} = \{f_\gamma \mid \rho(f_\gamma) = \gamma\}_{\gamma \in \Gamma}$ *as in Theorem 1 we define* $\tilde{E}(\delta) := Span_{\mathbb{F}_q}\{\varphi(f_{\alpha(i)}) \mid \alpha(i) \in \Delta(R, \rho, \varphi)$ *and* $\sigma(\alpha(i)) \geq \delta\}$.

We have

Theorem 3. $d(\tilde{E}(\delta)) \geq \delta$.

The codes $\tilde{E}(\delta)$ are sometimes very much better than the corresponding codes $E(\lambda)$. This is for instance the case for the improved generalized Reed-Muller codes known as hyperbolic codes (or Massey-Costello-Justesen codes). Regarding one-point geometric Goppa codes the picture very much relies on which particular curve we consider, but the improvement may also in this case be significant. The idea of controlling the minimum distance of a code by choosing the functions f_λ to be used in the code construction in a clever way will be one of the main ingredients of our modified Sudan decoding algorithm without multiplicity.

We now describe the modified Sudan decoding algorithm without multiplicity for the codes $E(\lambda)$ and $\tilde{E}(\delta)$. To ease notation we state the algorithm for a larger class of codes, namely for any code C of the form

$$C = \mathrm{Span}_{\mathbb{F}_q}\{\varphi(f_{\lambda_1}), \ldots, \varphi(f_{\lambda_k})\} \quad \text{where} \quad \{\lambda_1, \ldots, \lambda_k\} \subseteq \Delta(R, \rho, \varphi). \tag{1}$$

The first part of the decoding algorithm is to find a proper interpolation polynomial $Q(Z)$ with coefficients from the order domain R. To set up the decoding procedure for a given fixed code C we first need to describe sets from which we will allow the coefficients to be chosen. To this end consider the following definition.

Definition 7. *Given a code C as above let E be some fixed value (representing the number of errors we would like to correct). For $s \in \mathbb{N}_0$ define*

$$L(E,s) := \{\lambda \in \Delta(R,\rho,\varphi) \mid \textit{for all } \ i_1,\ldots,i_s \in \{1,\ldots,k\} \ \textit{ we have}$$
$$f_\lambda \prod_{v=1}^{s} f_{\lambda_{i_v}} \in \mathit{Span}\{f_{\alpha(1)},\ldots,f_{\alpha(n)}\} \textit{ and} \tag{2}$$
$$\sigma(\lambda_i) > E \textit{ for all } \ f_{\lambda_i} \in \mathit{Supp}_{\mathcal{B}}(f_\lambda \prod_{v=1}^{s} f_{\lambda_{i_v}})\}, \tag{3}$$

where $\mathit{Supp}_{\mathcal{B}}(f)$ of $f \in R$ is the set of $g \in \mathcal{B}$ that appears in the unique linear combination of f by elements in $\mathcal{B}$.

Note, that there is no requirement that $i_1,\ldots,i_s$ are pairwise different. Note also that the set $L(E,s)$ relies on the actual choice of well-behaving basis $\{f_\lambda\}_{\lambda\in\Gamma}$. Further we observe that for large values of s we have $L(E,s) = \emptyset$. What we will need for the modified version of Sudan type decoding without multiplicity to work is a number E such that $\sum_{s=0}^{\infty} \#L(E,s) > n$. As indicated above the value E will be the number of errors we can correct and therefore we would of course like to find a large value of E such that the above condition is met. On the other hand the smallest value t such that

$$\sum_{s=0}^{t} \#L(E,s) > n \tag{4}$$

holds will to some extent reflect the complexity of the decoding algorithm. So in some situations it might be desirable to choose a smaller value of E than the largest possible one to decrease the complexity of the algorithm. Choosing parameters E and t and calculating the corresponding sets $L(E,0),\ldots,L(E,t)$ is something that is done when setting up the decoding system. Hence, the complexity of doing this is not of very high importance. However, as we will demonstrate in the case of generalized Reed-Muller codes, there are often tricks to ease the above procedure. We are now able to describe the modified Sudan decoding algorithm without multiplicity.

Algorithm 1
Input: A code C as in (1), parameters E,t such that (4) is met and corresponding sets $L(E,0),\ldots,L(E,t)$. A received word $\boldsymbol{r}$
Output: A list of at most t codewords that contains all codewords within distance at most E from $\boldsymbol{r}$
Step 1. Find $Q_0,\ldots,Q_t \in R$ not all zero such that $Q_s \in \mathrm{Span}_{\mathbb{F}_q}\{f_\lambda \mid \lambda \in L(E,s)\}$ for $s = 0,\ldots,t$ and such that $\sum_{s=0}^{t}(\varphi(Q_s)) * \boldsymbol{r}^s = \boldsymbol{0}$ holds. (Here $\boldsymbol{r}^s$ means the component wise product of $\boldsymbol{r}$ with itself s times and $\boldsymbol{r}^0 = \boldsymbol{1}$.)
Step 2. Factorize $\sum_{s=0}^{t} Q_s Z^s \in R[Z]$ and detect all possible $f \in R$ such that $Z - f$ appears as a factor, which can be done by the method of Wu [11].
Step 3. Return $\{\varphi(f) \mid f \ \text{ is a solution from step 2}\}$.

Theorem 4. *Algorithm 1 gives the claimed output.*

Proof: Condition (4) ensures that the set of linear equations in step 1 has more indeterminates than equations. Therefore $Q_0, \ldots, Q_t$ as described in step 1 indeed do exist.

Consider any code word $\boldsymbol{c}$. That is, let $\boldsymbol{c} = \varphi(f)$ where f is of the form $f = \sum_{v=1}^{k} \beta_v f_{\lambda_v}$. From the conditions (2) and (3) we get that

$$\sum_{i=0}^{s} Q_i f^i \in \mathrm{Span}\{f_{\alpha(1)}, \ldots, f_{\alpha(n)}\} \tag{5}$$

holds and that

$$\text{all } \ f_{\alpha(v)} \in \mathrm{Supp}_{\mathcal{B}}(\sum_{i=0}^{s} Q_i f^i) \ \text{ satisfies } \ \sigma(\alpha(v)) > E. \tag{6}$$

Assume now that $\boldsymbol{c} = \varphi(f)$ is a code word within Hamming distance at most E from $\boldsymbol{r}$. But then $\sum_{s=0}^{t}(\varphi(Q_s)) * (\varphi(f))^s$ differs from $\sum_{s=0}^{t}(\varphi(Q_s)) * \boldsymbol{r}^s = \boldsymbol{0}$ in at most E positions implying

$$w_H(\varphi(\sum_{s=0}^{t} Q_s f^s)) \leq E \tag{7}$$

Combining (5), (6) and (7) with the first part of Theorem 2 lead to the conclusion that $\varphi(\sum_{s=0}^{t} Q_s f^s) = 0$ must hold, and Eq. (2) implies $\sum_{s=0}^{t} Q_s f^s = 0$. That is, f is a zero of $Q(Z)$. But order domains are integral domains and therefore $\mathrm{Quot}(R)$ is a field. It follows that $Z - f$ divides $Q(Z) \in \mathrm{Quot}(R)[Z]$. As the leading coefficient of $Z - f$ is 1 we conclude that $Q(Z) = (Z - f)K(Z)$ for some $K(Z)$ with coefficients in R. Hence, indeed $Z - f$ appears in the factorization in step 2 of the algorithm. Finally, as $Q(Z)$ has degree at most t the list in step 3 is of length at most t. □

Remark 1. We have used the Hamming weight to ensure $Q(Z) = 0$ in the above argument. The conventional method [10,9] used the degree of a polynomial and the pole order of an algebraic function to ensure $Q(Z) = 0$. The use of Hamming weight allows us to list-decode codes from any order domains.

The following example illustrates the nature of our modification.

Example 6. Consider a one-point geometric Goppa code $E(\eta)$ where $\eta < n$. Let, g be the genus of the function field or equivalently let $g = \#\mathbb{N}_0 \backslash \Gamma$. The set

$$L'(E, s) = \{\lambda \in \Gamma \mid \lambda + s\eta < n - E\}$$

is easily calculated and we have $L'(E, s) \subseteq L(E, s)$. Replacing $L(E, s)$ with $L'(E, s)$ in Algorithm 1 gives the traditional algorithm [9] without multiplicity for the one-point geometric Goppa code $E(\eta)$. Hence, for one-point geometric Goppa codes the modified algorithm can correct at least as many errors as the original one and in cases where the sets $L(E, s)$ are larger than the sets $L'(E, s)$ we will be able to correct more errors by the modified algorithm.

3 Generalized Reed-Muller Codes

In this section we consider the implementation of Algorithm 1 to the case of generalized Reed-Muller codes of low dimensions. Recall, from Example 1 that we have a weight function $\rho : \mathbb{F}_q[X_1, \ldots, X_m] \to \mathbb{N}_0^m$ given by $\rho(F) = (i_1, \ldots, i_m)$ if $X^{i_1} \cdots X^{i_m}$ is the leading monomial of F with respect to the monomial ordering from Example 1. Recall from Example 3 that we always choose the well-behaving basis $\mathcal{B}$ of $\mathbb{F}_q[X_1, \ldots, X_m]$ to be simply the set of monomials in $X_1, \ldots, X_m$. From Definition 4, for the weight function under consideration the σ function is easily calculated as follows

$$\sigma\left((i_1, \ldots, i_m)\right) = \prod_{v=1}^{m} (q - i_v).$$

We get the following Lemma that significantly eases the job with finding $L(E, s)$.

Lemma 1. *Let $u < q$ and consider the generalized Reed-Muller code $RM_q(u, m)$. The description of $L(E, s)$ simplifies to*

$$\begin{aligned} L(E,s) = \{(l_1, \ldots, l_m) \in \mathbb{N}_0^m \mid & \\ & l_1 + su, \ldots, l_m + su < q, && (8)\\ & (q - l_1 - su)(q - l_2) \cdots (q - l_m) > E, \\ & \vdots && (9)\\ & (q - l_1) \cdots (q - l_{m-1})(q - l_m - su) > E\} \end{aligned}$$

Proof: To see that (9) corresponds to (3) we observe that the σ function from this section is concave. The fact that (8) corresponds to (2) follows from similar arguments. □

To decide how many errors our algorithm can correct we should according to (4) look for the largest possible E such that a t exists with $\sum_{s=0}^{t} \#L(E, s) > n = q^m$. Of course such an E can always be found by an extensive trial and error. For the case of $m = 2$ that is, codes of the form $\mathrm{RM}_q(u, 2)$ we now give an approximative trial and error method that requires only few calculations. It turns out that this approximative method is actually rather precise.

For a fixed s the conditions to be satisfied are

$$l_1 + su < q, \qquad l_2 + su < q \tag{10}$$

$$(q - l_1 - su)(q - l_2) > E, \qquad (q - l_1)(q - l_2 - su) > E \tag{11}$$

We make the (natural) assumption

$$0 \leq l_1, l_2 < q. \tag{12}$$

Equations (11) and (12) imply (10) which we therefore can forget about. When $E < q$, it is easy to lower-bound the number of solutions to (11) and (12). Under the assumption $E \geq q$ we now want to count the number of possible solutions

to (11) and (12). The number of such solutions is bounded below by the area in the first quadrant of the points that are under both the curve

$$l_2 = q - \frac{E}{q - l_1 - su} \tag{13}$$

as well as are under the curve

$$l_2 = q - su - \frac{E}{q - l_1} \tag{14}$$

By symmetry these two curves intersect in two points of the form (γ, γ). We have to use the point closer to the origin, which we calculate to be

$$\gamma = \frac{2q - su - \sqrt{s^2u^2 + 4E}}{2}.$$

Therefore (again by symmetry) the area is

$$\begin{aligned} &2\left(\int_0^{\gamma}(q - su - \frac{E}{q - l_1})dl_1 - \frac{1}{2}\gamma^2\right) \\ &= 2(\gamma(q - su) - E(\ln(q) - \ln(q - \gamma)) - \frac{1}{2}\gamma^2) \end{aligned}$$

A rougher but simpler estimate is found by approximating the above area with the area of the polygon with corners $(0,0), (0, q - \frac{E}{q} - su), (\gamma, \gamma), (q - \frac{E}{q} - su, 0)$. Here the second point is found by substituting $l_1 = 0$ in (14) and the fourth point is found by substituting $l_2 = 0$ in (13). The estimate can serve as a lower bound due to the fact that both functions in (13) and (14) are concave. The area of the polygon is found to be $\gamma(q - (E/q) - su)$. Whether we use the first estimate or the second estimate we would next like to know the largest value of t such that $L(E,t) \neq \emptyset$. But this is easily calculated from the requirement $\gamma \geq 0$ implying $t = \lfloor (q - (E/q))/u \rfloor$. Combining the above results with Theorem 4 we get.

Proposition 1. *Consider the code $RM_q(u,2)$ with $u < q$. For $E \geq q$ Algorithm 1 can correct at least E errors if the following holds*

$$\sum_{s=0}^{\lfloor (q - E/q)/u \rfloor} (2(\gamma(q - su) - E(\ln(q) - \ln(q - \gamma)) - \frac{1}{2}\gamma^2)) > q^2.$$

Corollary 1. *Consider the code $RM_q(u,2)$ with $u < q$. For $E \geq q$ Algorithm 1 can correct at least E errors if the following holds*

$$\sum_{s=0}^{\lfloor (q - E/q)/u \rfloor} (\gamma(q - \frac{E}{q} - su)) > q^2.$$

Augot and Stepanov in [2] gave an improved estimate of the sum of multiplicities in terms of the total degree of a multivariate polynomial as follows

Theorem 5. *The sum of multiplicities in $\mathbb{F}_q^m$ of an m-variate polynomial of total degree d is upper bounded by dq^{m-1}. The number of zeros with multiplicity r of such a polynomial is upper bounded by dq^{m-1}/r.*

The above bound is better than the combination of Lemmas 2.4 and 2.5 in [8]. As noted by Augot and Stepanov Theorem 5 allows us to use more monomials in the first list decoding algorithm in [8], and the resulting decoding algorithm has the larger error-correcting capability.

The error correcting capability of the modified list decoding algorithm with Theorem 5 is compared with ours and the original Pellikaan-Wu. The multiplicity used in Augot and Stepanov's estimate is 10. E_{PW}, E_{PWA}, E_{ours} are the error correcting capability of the original Pellikaan-Wu, Augot-Stepanov, and our method, respectively. Finally, E_{PWA1} respectively E_{PWA2} are the error correcting capability of the Augot-Stepanov modified the Pellikaan-Wu algorithm when multiplicity is 1 respectively 2. $q = 16$, $m = 2$, $n = 256$.

u	2	3	4	5	6	7	8	9	10	11	12
E_{PW}	63	46	34	26	19	14	10	7	5	3	2
E_{ours}	76	55	44	34	27	21	15	13	11	9	6
E_{PWA}	118	99	83	70	59	49	41	33	25	19	11
E_{PWA1}	47	31	15	-1	-17	-33	-33	-49	-49	-65	-65
E_{PWA2}	87	63	47	31	23	7	-1	-9	-17	-25	-25

Remark 2. The authors of the present paper have done a lot of computer experiments regarding the error correcting capability of the proposed decoding method for generalized Reed-Muller codes. In all of these experiments we were able to correct as many errors as Remark 2.1 in [8] guarantees Pellikaan-Wu algorithm (with multiplicity) to be able to.

4 One-Point Geometric Goppa Codes

As already mentioned our proposed decoding algorithm applies among other things to one-point geometric Goppa codes. In this section we will be concerned with codes defined from the norm-trace curve introduced in [4]. These are defined by the polynomial $X^{(q^r-1)/(q-1)} - Y^{q^{r-1}} - Y^{q^{r-2}} - \cdots Y \in \mathbb{F}_{q^r}[X, Y]$. We consider codes

$$C_{\mathcal{L}}(P_1 + \cdots + P_{q^{2r-1}}, sP_\infty) \tag{15}$$

where $P_1, \ldots, P_{q^{2r-1}}, P_\infty$ are the rational places of the corresponding function field and P_∞ is the unique place among these with $\nu_{P_\infty}(x) < 0$. We do not go into detail with how to implement the proposed algorithm but present only some examples.

Example 7. In this example we consider the norm-trace curve corresponding to $q = 2$ and $r = 6$. These are of length $n = 2^{11}$. In the table below s is the value used in (15) whereas E_{our} is the error correcting capability of the proposed

method and E_{GS1} is the error correcting capability of Sudan's algorithm [10] without multiplicity. By 900-929 we indicate that maximal performance is a number between 900 and 929. With multiplicity, Guruswami-Sudan's algorithm [5] outperform the proposed method.

s	64	96	192	288	480
E_{our}	1008	900-929	660-669	527	346
E_{GS1}	962	804	479	237	14

Example 8. In this example we consider the norm-trace curve corresponding to $q = 3$ and $r = 3$. These are of length $n = 3^5$. In the table below s is the value used in (15) whereas E_{our} is the error correcting capability of the proposed method and E_{GS1} is the error correcting capability of Sudan's algorithm [10] without multiplicity. With multiplicity, Guruswami-Sudan's algorithm [5] outperform the proposed method.

s	63	70	80	88
E_{our}	55	51	43	38
E_{GS1}	53	47	39	33

References

1. Andersen, H.E., Geil, O.: Evaluation Codes From Order Domain Theory. Finite Fields and Their Appl. (2007) doi:10.1016/j.ffa.2006.12.004
2. Augot, D., Stepanov, M.: Decoding Reed-Muller Codes with the Guruswami-Sudan's Algorithm. In: Slides of Talk Given by D. Augot at Workshop D1 Special Semester on Gröbner Bases and Related Methods, RICAM, Linz (2006), `http://www.ricam.oeaw.ac.at/specsem/srs/groeb/download/Augot.pdf`
3. Geil, O., Pellikaan, R.: On the Structure of Order Domains. Finite Fields and Their Appl. 8, 369–396 (2002)
4. Geil, O.: On Codes From Norm-Trace Curves. Finite Fields and Their Appl. 9, 351–371 (2003)
5. Guruswami, V., Sudan, M.: Improved Decoding of Reed-Solomon and Algebraic-Geometry Codes. IEEE Trans. Inform. Theory 45(4), 1757–1767 (1999)
6. Høholdt, T., van Lint, J., Pellikaan, R.: Algebraic Geometry Codes. In: Pless, V.S., Huffman, W.C. (eds.) Handbook of Coding Theory, pp. 871–961. Elsevier, Amsterdam (1998)
7. O'Sullivan, M.E.: New Codes for the Berlekamp-Massey-Sakata Algorithm. Finite Fields and Their Appl. 7, 293–317 (2001)
8. Pellikaan, R., Wu, X.-W.: List Decoding of q-ary Reed-Muller Codes. IEEE Trans. Inform. Theory 50, 679–682 (2004)
9. Shokrollahi, M.A., Wasserman, H.: List Decoding of Algebraic-Geometric Codes. IEEE Trans. Inform. Theory 45(2), 432–437 (1999)
10. Sudan, M.: Decoding of Reed Solomon Codes Beyond the Error Correction Bound. J. Complexity 13, 180–193 (1997)
11. Wu, X.-W.: An Algorithm for Finding the Roots of the Polynomials Over Order Domains. In: 2002 IEEE International Symposium on Information Theory, p. 202. IEEE Press, New York (2002)

Bent Functions and Codes with Low Peak-to-Average Power Ratio for Multi-Code CDMA

Jianqin Zhou[1,⋆], Wai Ho Mow[2], and Xiaoping Dai[1]

[1] Department of Computer Science,
Anhui University of Technology, Ma'anshan, 243002 China
zhou9@yahoo.com
[2] Dept. of Electrical & Electronic Engineering,
Hong Kong Univ. of Science and Technology, Clear Water Bay, Hong Kong

Abstract. In this paper, codes which reduce the peak-to-average power ratio (PAPR) in multi-code code division multiple access (MC-CDMA) communication systems are studied. It is known that using bent functions to define binary codewords gives constant amplitude signals. Based on the concept of quarter bent functions, a new inequality relating the minimum order of terms of a bent function and the maximum Walsh spectral magnitude is proved, and it facilitates the generalization of some known results. In particular, a new simple proof of the non-existence of the homogeneous bent functions of degree m in $2m$ boolean variables for $m > 3$ is obtained without invoking results from the difference set theory. We finally propose a new coding approach to achieve the constant amplitude transmission of codeword length 2^m for both even m as well as odd m.

Keywords: CDMA, multi-code, Walsh-Hadamard transform, PAPR, bent function.

1 Introduction

Code-Division Multiple-Access (CDMA) in one form or another is likely to be at the heart of future cellular wireless communications systems, third generation and beyond, and the orthogonal multi-code system has been drawing much attention in the last two decades.

The orthogonal multi-code system can achieve the code division multiplexing by assigning each orthogonal code to each user, and one user can utilize plural orthogonal code sequences. This means that the peak signal power in an MC-CDMA system can be as large as n times the average signal power. Typically $n = 2^m$ where m lies between 2 and 6 [1]. Thus, an MC-CDMA signal can have a significantly higher peak-to-average power ratio (PAPR) than a basic rate signal.

⋆ Corresponding author. The research was supported by the Chinese Natural Science Foundation (No. 60473142) and the Hong Kong Research Grants Council (No. 617706).

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 60–71, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Usually, the high power amplifier(HPA) that has non-linear characteristic is used to obtain the high power efficiency. Particularly, the high power efficiency is required on the reverse link (mobile to base station), where low cost components and low power consumption are vital. Thus, transmitting MC-CDMA signals without distortion requires either a more expensive power amplifier that is linear across a wider range of amplitudes, or the signal with low PAPR. In [1,8,9,10], it is shown that using bent functions to define binary codewords gives constant amplitude signals. Constant amplitude binary codes of length $n = 2^2$ and 2^4 are classified in [8], which can also be obtained by computer search. However, binary constant amplitude codes of length $n = 2^m$ exist only for m even.

In this paper, we first review a simple communication model proposed in [1] for MC-CDMA which captures the key features of an MC-CDMA reverse link in Section 2. The concept of quarter bent functions is presented in Section 3. Based on the new concept, a simpler method to find all 30 homogeneous bent functions of degree 3 in 6 boolean variables is given.

It is proved in [7] that homogeneous bent functions of degree m in $2m$ variables do not exist for $m > 3$. The proof uses a certain decomposition of a Menon difference set, which corresponds to any bent function. In section 4, it is proved that if the order of every term in a Boolean function $f(x)$ with $2m$ variables is more than $m - k$, and $|S_{(f)}(w)| \leq 2^{-m+t}$, where m, k and t are positive integers, then $\binom{2(m-k)}{m-k} \leq 2^{m+t}$, from which it follows that there do not exist homogeneous bent functions of degree m in $2m$ boolean variables for $m > 3$. Thus the result obtained here generalizes the main work in [7]. The property obtained here implies that the order of terms in spectrally bounded $f(x)$ over V_n is distributed flatly, or can not limited in a small range, under the condition that $\deg(f(x))$ is almost $\frac{n}{2}$. Thus it can be used to guide the construction of bent functions.

We finally present a new coding approach to achieve the constant amplitude transmission of codeword length 2^m for both even m as well as odd m in Section 5.

2 Preliminaries

In this section we first review the communication model of the reverse link of an MC-CDMA system. Throughout Section 2 and Section 5 n will be a power of 2. We write $n = 2^m$.

The Walsh-Hadamard matrix WH_n can be defined recursively by $WH_1 = (1)$ and

$$WH_{2^j} = \begin{pmatrix} WH_{2^{j-1}} & WH_{2^{j-1}} \\ WH_{2^{j-1}} & -WH_{2^{j-1}} \end{pmatrix}$$

This matrix is a $\{+1, -1\}$-matrix and is symmetric and orthogonal, so that: $WH_n \cdot WH_n = nI_n$, where I_n denotes the $n \times n$ identity matrix. Thus, the rows (or columns) of WH_n are orthogonal vectors of length n, called Walsh-Hadamard sequences.

It is easy to show by induction that

$$WH_{2^m} = ((-1)^{\sum_{k=0}^{m-1} j_k t_k})_{jt}$$

where $j = \sum_{k=0}^{m-1} j_k 2^k$ and $t = \sum_{k=0}^{m-1} t_k 2^k$ are radix-2 decompositions of j and t, respectively.

Let $\mathbf{j} = (j_0, j_1, \ldots, j_{m-1})$ and $\mathbf{t} = (t_0, t_1, \ldots, t_{m-1})$. Then,

$$WH_{2^m} = ((-1)^{\mathbf{j}\cdot\mathbf{t}^T})_{jt}$$

where superscript T denotes the transposition, subscripts j and t still denote integers.

We start with considering an MC-CDMA system without coding. We have n parallel streams of bits and the signal transmitted by a user on the reverse link corresponding to a vector $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1})$ of data bits (one bit $c_i \in \{0, 1\}$ from each stream) is the time-domain vector of real values $S(\mathbf{c}) = (S(\mathbf{c})_0, S(\mathbf{c})_1, \ldots, S(\mathbf{c})_{n-1})$ where

$$S(\mathbf{c})_t = \sum_{j=0}^{n-1} (-1)^{c_j} (WH_n)_{jt} = \sum_{j=0}^{n-1} (-1)^{c_j} (-1)^{\mathbf{j}\cdot\mathbf{t}^T} \tag{1}$$

Writing $(-1)^{\mathbf{c}} = ((-1)^{c_0}, (-1)^{c_1}, \ldots, (-1)^{c_{n-1}})$, we have $S(\mathbf{c}) = (-1)^{\mathbf{c}} \cdot WH_n$

In a real MC-CDMA system, the power required to transmit a signal is proportional to the square of the signal value. Since we are interested only in the peak-to-average power ratio, we define the instantaneous power of the signal $S(\mathbf{c})$ at time t to be $P(\mathbf{c})_t = S(\mathbf{c})_t^2$. From (1), the peak (i.e. largest) value of $P(\mathbf{c})_t$ can be as large as n^2.

$$\begin{aligned} \sum_{t=0}^{n-1} P(\mathbf{c})_t &= S(\mathbf{c})(S(\mathbf{c}))^T = (-1)^{\mathbf{c}} \cdot WH_n WH_n \cdot ((-1)^{\mathbf{c}})^T \\ &= n(-1)^{\mathbf{c}} \cdot ((-1)^{\mathbf{c}})^T = n \cdot n = n^2 \end{aligned}$$

It follows that the average value of $P(\mathbf{c})_t$ over $0 \le t < n$ is equal to n. Therefore we define the peak-to-average power ratio of the vector of data bits $\mathbf{c}$ (and the corresponding signal $S(\mathbf{c})$) to be

$$PAPR(\mathbf{c}) = \frac{1}{n} \max_{0 \le t < n} P(\mathbf{c})_t$$

From the above discussion we know that $1 \le PAPR(\mathbf{c}) \le n$.

Now we consider coding for MC-CDMA. We let C be an arbitrary binary code of length n and rate R, that is a set of 2^{nR} binary length n vectors. An encoder for C maps $k = nR$ information bits at a time onto vectors $\mathbf{c} \in C$. In MC-CDMA with coding, we have k parallel data streams that are fed into an

encoder for C and hence to a Walsh-Hadamard transform. Thus, only codewords c in C are selected for transmission, though (1) still describes the transmitted signal.

We define the PAPR of the code C to be $PAPR(C) = \max_{\mathbf{c}\in C} PAPR(\mathbf{c})$. A code C with $PAPR(C) = 1$ is called a constant amplitude code. Such a code attains the lowest and therefore best possible value of PAPR. The main focus of this paper is the construction of codes for MC-CDMA with low PAPR.

Let V_m be the vector space of k-tuples of element from $GF(2)$. Let $\alpha = (a_1, \cdots, a_m)$ and $\mathbf{x} = (x_1, \cdots, x_m)$. The inner product of x and α is defined as $\alpha \cdot \mathbf{x} = a_1x_1 \oplus \cdots \oplus a_mx_m$.

The Walsh-Hadamard transform of Boolean function $f : V_m \to V_1$ is defined as follows,

$$S_{(f)}(w) = 2^{-m} \sum_{x\in V_m} (-1)^{f(x)}(-1)^{w\cdot x}, \quad \text{where} \quad w \in V_m \ .$$

Definition 1. *The function $f(x)$ is bent if $|S_{(f)}(w)| = 2^{-\frac{m}{2}}$ for all $w \in V_m$. The function $f(x)$ is semi bent if $|S_{(f)}(w)| = 0$ or $2^{-\frac{m-1}{2}}$ for all $w \in V_m$.*

Comparing this definition with those of $S(\mathbf{c})$, we see that

$$S(\mathbf{c})_t = \hat{c}(t_0, t_1, \ldots, t_{m-1}), t = \sum_{k=0}^{m-1} t_k 2^k$$

The following lemma is now immediate [1].

Lemma 1. *Let c be a word of length $n = 2^m$. Then $PAPR(c) = \frac{1}{n} \max_{u\in V_m} |\hat{c}(u)|^2$. Moreover c has PAPR equal to 1 if and only if $\hat{c}$ is a bent function. In particular, constant amplitude codes of length $n = 2^m$ exist only for m even.*

In the discussion that follows, we will occasionally refer to a Boolean function $f : V_m \to V_1$ as a codeword of length 2^m.

3 The Quarter Bent Functions and Homogeneous Bent Functions

Let $f(x)$ be a boolean function: $V_n \to V_1$. The support set of its spectrum is defined as follows:

$$\text{supp}S_{(f)} = \{w \in V_n | S_{(f)}(w) \neq 0\}$$

Obviously, the support set of spectrum for a bent function $f(x)$ in n variables is V_n .

From Parseval identity, $\sum S^2_{(f)}(w) = 1$, we know that the support set of spectrum for a semi bent function $f(x)$ has 2^{n-1} elements, namely $|\ \text{supp}S_{(f)}| = 2^{n-1}$.

Lemma 2. *Let $X_1 = (x_1, x_2, \cdots, x_{n-1}), X = (X_1, x_n), g_1(X_1) = f(X_1, 0)$, and $g_2(X_1) = f(X_1, 1)$. Then $f(X)$ is a bent function if and only if both g_1 and g_2 are semi bent functions in $n-1$ variables, and g_1 and g_2 have disjoint support sets of spectrum.*

Proof. As $g_1(X_1) = f(X_1, 0), g_2(X_1) = f(X_1, 1)$, hence $f(X) = g_1(X_1)(x_n + 1) + g_2(X_1)x_n$.

Let $W_1 = (w_1, w_2, \cdots, w_{n-1}), W = (W_1, w_n), X_1 = (x_1, x_2, \cdots, x_{n-1})$, and $X = (X_1, x_n)$. Then,

$$\begin{aligned} S_{(f)}(W) &= 2^{-n} \sum_{X \in V_n} (-1)^{f(X)} (-1)^{W \cdot X} \\ &= 2^{-n} \sum_{X \in V_n} (-1)^{g_1(X_1)(x_n+1)+g_2(X_1)x_n} (-1)^{W_1 \cdot X_1 + w_n x_n} \\ &= 2^{-n} \Big(\sum_{X \in V_{n-1}} (-1)^{g_1(X_1)+W_1 \cdot X_1} + (-1)^{w_n} \sum_{X \in V_{n-1}} (-1)^{g_2(X_1)x_n + W_1 \cdot X_1} \Big) \\ &= 2^{-1} (S_{(g_1)}(W_1) + (-1)^{w_n} S_{(g_2)}(W_1)) \end{aligned} \tag{2}$$

Since $f(X)$ is a bent function, and w_n is 0 or 1, thus,

$$\begin{cases} |S_{(g_1)}(W_1) + S_{(g_2)}(W_1)| = 2 \cdot 2^{-\frac{n}{2}} \\ |S_{(g_1)}(W_1) - S_{(g_2)}(W_1)| = 2 \cdot 2^{-\frac{n}{2}}. \end{cases}$$

It is easy to show that $(|S_{(g_1)}(W_1)|, |S_{(g_2)}(W_1)|) = (2^{-\frac{n-2}{2}}, 0)$ or $(|S_{(g_1)}(W_1)|, |S_{(g_2)}(W_1)|) = (0, 2^{-\frac{n-2}{2}})$. Therefore, both g_1 and g_2 are semi bent functions in $n-1$ variables, and g_1 and g_2 have disjoint support sets of spectrum. On the other hand, if $(|S_{(g_1)}(W_1)|, |S_{(g_2)}(W_1)|) = (2^{-\frac{n-2}{2}}, 0)$ or $(|S_{(g_1)}(W_1)|, |S_{(g_2)}(W_1)|) = (0, 2^{-\frac{n-2}{2}})$, then it is easy to show that $f(X)$ is a bent function. □

Definition 2. *The function $f(x)$ is quarter bent if the Walsh-Hadamard transform $|S_{(f)}(w)| = 0$ or $2^{-\frac{n}{2}}$ or $2^{-\frac{n-2}{2}}$ for all $w \in V_n$.*

In a similar way to the above discussion, it is easy to prove the following lemmas.

Lemma 3. *Let $X_1 = (x_1, x_2, \cdots, x_{n-1}), X = (X_1, x_n), g_1(X_1) = f(X_1, 0), g_2(X_1) = f(X_1, 1)$, and $f(X)$ be a semi bent function, then both g_1 and g_2 are quarter bent functions in $n-1$ variables.*

Lemma 4. *Let $X_1 = (x_1, x_2, \cdots, x_{n-1}), X = (X_1, x_n), g_1(X_1) = f(X_1, 0), g_2(X_1) = f(X_1, 1)$, and both g_1 and g_2 be bent functions. Then $f(X)$ is a semi bent function.*

We now discuss the homogeneous bent functions of degree $k (k > 1)$ over V_n .

Let $f(X)$ be a bent function over V_n . It is known that the degree of $f(X)$ is not greater than $n/2$. If only we ensure that $f(X)$ does not contain the terms with a degree less than $n/2$, then $f(X)$ must be a homogeneous bent function.

Suppose that $f(X)$ has the following unique algebraic normal form:

$$f(x_1, x_2, \cdots, x_n) = a_0 + a_1x_1 + a_2x_2 + \cdots + a_nx_n + a_{1,2}x_1x_2 + \cdots + a_{n-1,n}x_{n-1}x_n + \cdots + a_{1,2,\cdots,n}x_1x_2\cdots x_n$$

Let $f(X)$ be a homogeneous bent functions of degree k, then a_0 =0, which is equivalent to $f(0, 0, \cdots, 0) = 0$.

Similarly, as a_0 =0, then a_1 =0 is equivalent to $f(1, 0, \cdots, 0) = 0$, a_2 =0 is equivalent to $f(0, 1, \cdots, 0) = 0, \cdots, a_n$ =0 is equivalent to $f(0, 0, \cdots, 1) = 0$.

As $a_0, a_1, a_2, \cdots, a_n$ are 0, then $a_{1,2} = 0$ is equivalent to $f(1, 1, 0\cdots, 0) = 0, a_{1,3} = 0$ is equivalent to $f(1, 0, 1, 0\cdots, 0) = 0, \cdots$.

Similarly, if one term with a degree less than k is not contained in $f(X)$, then there must exists a $\alpha \in V_n$, such that $f(\alpha) = 0$.

From Lemma 2, $f(X) = g_1(X_1)(x_n + 1) + g_2(X_1)x_n = g_1(X_1) + (g_1(X_1) + g_2(X_1))x_n$.

Let $f(X)$ be a homogeneous bent functions of degree k, then $g_1(X_1)$ does not contain any term with a degree less than k, correspondingly $g_1(X_1)$ is zero on those points of V_{n-1} ; $g_2(X_1)$ does not contain any term with a degree less than $k - 1$, correspondingly $g_2(X_1)$ is zero on those points of V_{n-1}.

Let $I(x_1, x_2, \cdots, x_n) = \sum_{i=1}^{n} x_i 2^{i-1}$, here the addition is in general meaning. Then $I : V_n \to \{0, 1, 2, \cdots, 2^n - 1\}$ is a one-to-one mapping. Therefore $(x_1, x_2, \cdots, x_n) \in V_n$ can be represented by an integer $I(x_1, x_2, \cdots, x_n)$. For example, (1,1) can be represented by 3.

Now we can discuss the homogeneous bent functions of degree 3 over V_6 .

Let f(X) be a homogeneous bent functions of degree 3. From the discussion above, the points belonging to {0,1,2,3,4,5,6, 8,9,10,12,16,17, 18,20,24, 32,33, 34,36,40,48} are zero points of $f(X)$. For example, as $f(0, 0, 0, 0, 1, 1) = 0$, thus 48 is one of the zero points of $f(X)$.

Let $f(X) = g_1(X_1) + (g_1(X_1) + g_2(X_1))x_6$. Then $g_1(X_1)$ does not contain any term with a degree less than 3, thus the points belonging to {0,1,2,3,4,5,6, 8,9,10,12,16,17,18,20,24} must be zero points of $g_1(X_1)$; $g_2(X_1)$ does not contain any term with a degree less than 2, thus the points belonging to {0,1,2,4, 8,16} must be zero points of $g_2(X_1)$.

From Lemma 2, we know that $g_1(X_1)$ is a semi bent function, therefore,

$$|S_{(g_1)}(0)| = 2^{-5} \sum_{X_1 \in V_5} (-1)^{g_1(X_1)}(-1)^{0 \cdot X_1} = 2^{-5} \sum_{X_1 \in V_5} (-1)^{g_1(X_1)} = 0 \text{ or } 2^{-2}$$

If $S_{(g_1)}(0) = -2^{-2} = -\frac{8}{32}$, then $g_1(X_1)$ must take on the value 1 with 20 points and take on the value 0 with 12 points. This contradicts the fact that the points belonging to {0,1,2,3,4,5,6,8,9,10,12,16,17,18,20,24} must be zero points of $g_1(X_1)$.

If $S_{(g_1)}(0) = 0$, then $g_1(X_1)$ must take on the value 1 with the points not belonging to {0,1,2,3,4,5,6,8,9,10,12,16,17, 18,20,24}. It is easy to show that the $g_1(X_1)$ with these conditions is not a semi bent function.

If $S_{(g_1)}(0) = 2^{-2} = \frac{8}{32}$, then $g_1(X_1)$ must take on the value 0 with 20 points and take on the value 1 with 12 points. It is easy to verify by computer that there are 15 cases that $g_1(X_1)$ with these conditions is a semi bent function.

As $S_{(g_1)}(0) = 2^{-2}$, from Lemma 1, hence $S_{(g_2)}(0) = 0$. Therefore $g_2(X_1)$ must take on the value 0 with 16 points and take on the value 1 with 16 points. It is easy to verify by computer that there are 64056 cases that $g_2(X_1)$ is a semi bent function with these conditions.

For every $g_1(X_1)$ of the 15 cases, there is only 2 $g_2(X_1)$ in the 64056 cases that $g_1(X_1)$ and $g_2(X_1)$ have disjoint support sets of spectrum. Therefore, for every $g_1(X_1)$ of the 15 cases, there are 2 homogeneous bent functions of degree 3. Thus there are totally 30 homogeneous bent functions of degree 3.

For example, let $g_1(X_1)$=$\{$7,11,13,14,19,21,23,26,27,28,29, 30$\}$. Then the spectrum set of $2^5 S_{(g_1)}(W)$ is $\{$8,8,8,0,8,0,8, -8, 8,8,0,-8,0,-8,-8,8, 8,0,0,0,0,0,-8,0, 0,-8,0,0,0,0,0,8$\}$, where W is in the sequence of $0, 1, 2, \cdots, 2^5 - 1$; $g_2(X_1)$ can be $\{$3,6,7,9,11,12,13,14,17, 18,20,21,22,24,25,26$\}$, the spectrum set of $2^5 S_{(g_2)}(W)$ is $\{$0,0,0,8,0,8,0,0, 0,0,8,0,8,0,0,0, 0,8,8, -8,8,8,0,-8, 8,0,8,-8,-8,-8,-8,0$\}$, where W is in the sequence of $0, 1, 2, \cdots, 2^5 - 1$.

Obviously, $g_1(X_1)$ and $g_2(X_1)$ have disjoint support sets of spectrum, hence constructing a homogeneous bent functions of degree 3, namely $f(X)$= $\{$7, 11, 13, 14, 19, 21, 23, 26, 27, 28, 29, 30, 35, 38, 39, 41, 43, 44, 45, 46, 49, 50, 52, 53, 54, 56, 57, 58$\}$, with the following unique algebraic normal form: $f(X) = x_1x_2x_3 + x_1x_2x_4 + x_1x_2x_5 + x_1x_2x_6 + x_1x_3x_4 + x_1x_3x_5 + x_1x_4x_6 + x_1x_5x_6 + x_2x_3x_4 + x_2x_3x_6 + x_2x_4x_5 + x_2x_5x_6 + x_3x_4x_5 + x_3x_4x_6 + x_3x_5x_6 + x_4x_5x_6$.

4 Spectrally Bounded Functions of Degree m over V_{2m}

It is easy to verify by computer that there are 28 homogeneous bent functions of degree 2 over V_4 . For example, $x_1x_2 + x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4 + x_3x_4$ =$\{$3, 5, 6, 7, 9, 10, 11, 12, 13, 14$\}$ is a homogeneous bent functions of degree 2.

We know that there are 30 homogeneous bent functions of degree 3 over V_6 .

However, we will prove that homogeneous bent functions of degree m over V_{2m} do not exist for $m > 3$.

Let $f(X)$ be a homogeneous bent function of degree m over V_n , where $n = 2m$. Then $f(X)$ does not contain the terms with a degree less than m.

Since $f(X) = g_1(X_1) + (g_1(X_1) + g_2(X_1))x_n$, then $g_1(X_1)$ is a semi bent function that does not contain the terms with a degree less than m.

From Lemma 3, $g_1(X_1) = h_1(X_2) + (h_1(X_2) + h_2(X_2))x_{n-1}$, where $X_2 = (x_1, \cdots, x_{n-2})$, $X_1 = (X_2, x_{n-1}), h_1(X_2) = g_1(X_2, 0), h_2(X_2) = g_1(X_2, 1)$, thus $h_1(X_2)$ is a quarter bent function that does not contain the terms with a degree less than m.

In $h_1(X_2)$, the biggest number of the terms with a degree 0 is 1;

The biggest number of the terms with a degree 1 is $\binom{n-2}{1} = \binom{2(m-1)}{1}$; $\cdots\cdots$; The biggest number of the terms with a degree $m-1$ is $\binom{2(m-1)}{m-1}$.

Therefore, the biggest number of the terms with a degree less than m is, $c_0 = \binom{2(m-1)}{0} + \binom{2(m-1)}{1} + \binom{2(m-1)}{2} + \cdots + \binom{2(m-1)}{m-1}$

Since $2^{2(m-1)} = (1+1)^{2(m-1)} = c_0 + \binom{2(m-1)}{m} + \cdots + \binom{2(m-1)}{2(m-1)} = 2c_0 - \binom{2(m-1)}{m-1}$ Thus

$$2c_0 = 2^{2(m-1)} + \binom{2(m-1)}{m-1} \tag{3}$$

Moreover, the fact that $h_1(X_2)$ does not contain a term with a degree less than m, is equivalent to that there exists a $\alpha \in V_{2(m-1)}$ that $h_1(\alpha)$=0, thus the number of points $\alpha \in V_{2(m-1)}$ that $h_1(\alpha)$=0 is not less than c_0 .

On the other hand, as $h_1(X_2)$ is a quarter bent function over $V_{2(m-1)}$, hence $2^{2(m-1)}S_{(h_1)}(0) = \sum_{X_2 \in V_{2(m-1)}} (-1)^{h_1(X_2)} \leq 2^{2(m-1)}2^{-m+2} = 2^m$.

Suppose the number of points $\alpha \in V_{2(m-1)}$ that $h_1(\alpha)$=0 is y, then $y - (2^{2(m-1)} - y) \leq 2^m$, namely $y \leq (2^{2(m-1)} + 2^m)/2$. Hence, the number of points $\alpha \in V_{2(m-1)}$ that $h_1(\alpha)$=0 is not more than $(2^{2(m-1)} + 2^m)/2$.

From equality (3), $(2^{2(m-1)} + \binom{2(m-1)}{m-1})/2 \leq (2^{2(m-1)} + 2^m)/2$, namely,

$$\binom{2(m-1)}{m-1} \leq 2^m \tag{4}$$

For $m = 3$, $\binom{2(m-1)}{m-1} = \binom{4}{2} = 6 < 2^3 = 8$, inequality (4) holds. For $m = 4$, $\binom{2(m-1)}{m-1} = \binom{6}{3} = 20 > 2^4 = 16$, inequality (4) does not hold.

Now we consider the case $m > 3$.

Since $2(m-1) - (m-2) = m \geq 4 = 2^2$, and $2m-2 = 2(m-1), 2m-3 > 2(m-2), \cdots$, so $\binom{2(m-1)}{m-1} = \frac{(2m-2)(2m-3)\cdots m}{(m-1)(m-2)\cdots 1} > 2^m$; thus inequality (4) does not hold.

Hence we have the following theorem.

Theorem 1. *Homogeneous bent functions of degree m over V_{2m} do not exist for $m > 3$.*

We now discuss a more general case. First we introduce a lemma.

Lemma 5. *If a Boolean function $f(x)$ over V_n , $n = 2m$, has a bounded Walsh-Hadamard transform, say $|S_{(f)}(w)| \leq 2^{-m+t}$. Let $g(x_1, \cdots, x_{2m-k})$=$f($ x_1, $\cdots$, x_{2m-k}, $0, \cdots, 0)$ be a Boolean function in $2m-k$ variables, where k is a positive integer. Then $|S_{(g)}(w_1, \cdots, w_{2m-k})| \leq 2^{-m+t+k}$.*

Proof. For $k = 1$, the proof is similar to that of Lemma 2

Clearly we can continue in this way. □

The main result of this section is the following.

Theorem 2. *If the order of every term in a Boolean function $f(x)$ over V_{2m} is more than $m-k$, and $|S_{(f)}(w)| \leq 2^{-m+t}$, then $\binom{2(m-k)}{m-k} \leq 2^{m+t}$.*

Proof. Let $g(x_1, \cdots, x_{2(m-k)}) = f(x_1, \cdots, x_{2(m-k)}, 0, \cdots, 0)$ be a Boolean function in $2(m-k)$ variables, where k is a positive integer. We know that

$|S_{(g)}(w_1, \cdots, w_{2(m-k)})| \leq 2^{-m+t+2k}$, and the order of every term in $g(x_1, \cdots, x_{2(m-k)})$ is more than $m-k$.

Very similar to the discussion of Theorem 1, we know that the biggest number of the terms in $g(x_1, \cdots, x_{2(m-k)})$ with a degree less than $m-k+1$ is $c_0 = \binom{2(m-k)}{0} + \binom{2(m-k)}{1} + \binom{2(m-k)}{2} + \cdots + \binom{2(m-k)}{m-k}$. Thus

$$2c_0 = 2^{2(m-k)} + \binom{2(m-k)}{m-k} \tag{5}$$

On the other hand, as $g(x_1, \cdots, x_{2(m-k)})$ is a function over $V_{2(m-k)}$, hence

$$2^{2(m-k)} S_{(g)}(0) = \sum_{X_2 \in V_{2(m-k)}} (-1)^{g(X_2)} \leq 2^{2(m-k)} 2^{-m+t+2k} = 2^{m+t}.$$

Furthermore, the number of points $\alpha \in V_{2(m-k)}$ that $g(\alpha)$=0 is not more than $(2^{2(m-k)} + 2^{m+t})/2$. From equality (5), we have $\binom{2(m-k)}{m-k} \leq 2^{m+t}$. □

Let $t = 0$ (here $|S_{(f)}(w)| = 2^{-m}$), and $k = 1$. Then $\binom{2(m-1)}{m-1} \leq 2^m$ implies that $m <= 3$. Therefore, Theorem 1 is a corollary of Theorem 2. Let $t = 0$ (here $|S_{(f)}(w)| = 2^{-m}$), and $k = 2$. Then $\binom{2(m-2)}{m-2} \leq 2^m$ implies that $m <= 5$.

5 A New Coding Approach to Achieve the Constant Amplitude Transmission

We now present a new coding approach to achieve the constant amplitude transmission of codeword length 2^m for both odd and even m.

Let us define the new code as a function $g : V_m \to \{-1, 0, 1\}$, which takes m boolean variables and assigns a value from $\{-1, 0, 1\}$. The ternary Walsh spectrum is defined as follows:

$$S_g(u) = \sum_{x \in V_m} g(x)(-1)^{x \cdot u}, u \in V_m$$

Let $S_g = (S_g(0, 0, \ldots, 0), S_g(1, 0, \ldots, 0), \ldots, S_g(1, 1, \ldots, 1))$, and $g = (g(0, 0, \ldots, 0), g(1, 0, \ldots, 0), \ldots, g(1, 1, \ldots, 1))$. Then $S_g^2(0, 0, \ldots, 0) + S_g^2(1, 0, \ldots, 0) + \ldots + S_g^2(1, 1, \ldots, 1) = S_g S_g^T = g(WH)_{2^m}(WH)_{2^m}^T g^T = 2^m g g^T$.

Let $weight(g)$=$|\{x | g(x) \neq 0$ and $x \in V_m\}|$. Then the average value of $S_g^2(u)$ over V_m is $gg^T = weight(g)$.

Now we can construct codes to achieve the constant amplitude transmission for both odd and even m. Given integer k, suppose that $2m \leq k$, and $f(x)$ is a bent binary function over V_{2m}. Let

$$g(x, x_{2m+1}, \ldots, x_k) = \begin{cases} (-1)^{f(x)}, & x_{2m+1} = \ldots = x_k = 0 \\ 0, & \text{otherwise} \end{cases}$$

where $x \in V_{2m}$. Then we have

$$\begin{aligned}
&\mathrm{PAPR}(g)\\
&= \max_{(u,u_{2m+1},\ldots,u_k)\in V_k} \Bigg(\sum_{(x,x_{2m+1},\ldots,x_k)\in V_k} g(x, x_{2m+1}, \ldots, x_k)\\
&(-1)^{(x,x_{2m+1},\ldots,x_k)(u,u_{2m+1},\ldots,u_k)^T} \Bigg)^2 / weight(g)\\
&= \max_{(u,u_{2m+1},\ldots,u_k)\in V_k} (\sum_{(x,0,\ldots,0)\in V_k} (-1)^{f(x)} (-1)^{xu^T})^2 / 2^{2m}\\
&= \max_{(u,u_{2m+1},\ldots,u_k)\in V_k} 2^{2m}/2^{2m} = 1
\end{aligned}$$

Now we consider the number of codewords with PAPR equal to 1. It is known that x_1x_2 is a bent function over V_2. $x_1x_2 + c_1x_1 + c_2x_2 + c_0$, where c_0, c_1, c_2 are in $GF(2)$, are totally 8 bent functions, and the minimum distance of these 8 codewords is 2. The distance between x_1x_2 and $x_1x_2 + 1$ is 4.

Let $m = 5$. Now we consider a code C of length $n = 2^5$. Suppose $g \in C$. Let f be a bent function over V_2. We define g as follows: Weight(g)=4; $g(u_0) = (-1)^{f(0,0)}$; $g(u_1) = (-1)^{f(1,0)}$; $g(u_2) = (-1)^{f(0,1)}$; $g(u_3) = (-1)^{f(1,1)}$. Here $\{u_0, u_1, u_2, u_3\} \subset V_5$ and $u_0 < u_1 < u_2 < u_3$, such that $\mathrm{PAPR}(g) = 1$.

Case A. Among $\{0, 1, 2, \ldots, 2^5-1\}$, select any two even integers $2i < 2j$. Then let $u_0 = 2i, u_1 = 2i+1, u_2 = 2j, u_3 = 2j+1$. There are totally $\binom{2^4}{2}$ such codewords.

Based on the following example, it is straightforward to give the detailed proof, which is omitted here.

Let $i = 2, j = 8$, $u_0 = 2, u_1 = 3, u_2 = 8$, $u_3 = 9$. Then the radix-2 decompositions of $\{u_0, u_1, u_2, u_3\}$ are the following,

$$\begin{pmatrix} 0\,1\,0\,0\,0 \\ 1\,1\,0\,0\,0 \\ 0\,0\,0\,1\,0 \\ 1\,0\,0\,1\,0 \end{pmatrix}$$

Thus, $S_g(w_0, w_1, w_2, w_3, w_4) = \sum_{(x_0,x_1,x_2,x_3,x_4)\in V_5} g(x_0, x_1, x_2, x_3, x_4) \cdot (-1)^{(x_0,x_1,x_2,x_3,x_4)(w_0,w_1,w_2,w_3,w_4)^T} = \sum_{(v_0,v_1)\in V_2} (-1)^{f(v_0,v_1)+w_1(1+v_1)}$. $(-1)^{(v_0,v_1)(w_0,w_3)^T}$. Since $f(v_0, v_1) + w_1(1 + v_1)$ is still a bent function over V_2, $(S_g(w_0, w_1, w_2, w_3, w_4))^2 = 2^2 = 4$ for any $(w_0, w_1, w_2, w_3, w_4) \in V_5$.

Note that there are $\binom{16}{2}$ ways to select two even integers $2i < 2j$ among $\{0, 1, 2, \ldots, 2^5 - 1\}$, hence it is easy to show that there are totally $\binom{16}{2}$ codewords in this case.

Case B. Among $\{0, 1, 2, \ldots, 2^5-1\}$, select any two integers divisible by 4, namely, $4i < 4j$, then let $4i \le u_0 < u_1 < 4j \le u_2 < u_3$. There are totally $\binom{2^3}{2} \times 2^3$ such codewords.

For example, let $i = 4, j = 16$, $u_0 = 4, u_1 = 6, u_2 = 16$, $u_3 = 18$. Then the radix-2 decompositions of $\{u_0, u_1, u_2, u_3\}$ are the following,

$$\begin{pmatrix} 0\,0\,1\,0\,0 \\ 0\,1\,1\,0\,0 \\ 0\,0\,0\,0\,1 \\ 0\,1\,0\,0\,1 \end{pmatrix}$$

Thus, $S_g(w_0, w_1, w_2, w_3, w_4) = \sum_{(v_0,v_1)\in V_2} (-1)^{f(v_0,v_1)+w_2(1+v_1)}(-1)^{(v_0,v_1)(w_1,w_4)^T}$. Hence PAPR$(g) = 1$.

Let $u_0 = 5, u_1 = 6, u_2 = 16$, $u_3 = 19$. Then the radix-2 decompositions of $\{u_0, u_1, u_2, u_3\}$ are the following,

$$\begin{pmatrix} 1\,0\,1\,0\,0 \\ 0\,1\,1\,0\,0 \\ 0\,0\,0\,0\,1 \\ 1\,1\,0\,0\,1 \end{pmatrix}$$

Thus,

$$\begin{aligned} &S_g(w_0, w_1, w_2, w_3, w_4) \\ &= \sum_{(v_0,v_1)\in V_2} (-1)^{f(v_0,v_1)+w_0(1+v_0+v_1)+w_2(1+v_1)}(-1)^{(v_0,v_1)(w_1,w_4)^T} \end{aligned}$$

Hence PAPR$(g) = 1$.

Suppose that $f(v_0, v_1)$ is a bent function over V_2. Then we know that $f(v_0, v_1) + w_0(c_0v_0 + c_1v_1 + c_2)$, $c_0, c_1, c_2 \in GF(2)$, is still a bent function over V_2. Thus, it is easy to show that there are totally $\binom{2^3}{2} \times 2^3$ codewords in this case.

Case C. Among $\{0, 1, 2, \ldots, 2^5-1\}$, select any two integers divisible by 8, namely, $8i < 8j$, then let $8i \le u_0 < u_1 < 8j \le u_2 < u_3$. There are totally $\binom{2^2}{2} \times (2^3)^2$ such codewords.

Case D. Among $\{0, 1, 2, \ldots, 2^5 - 1\}$, select any two integers divisible by 16, namely, $16i < 16j$, then let $16i \le u_0 < u_1 < 16j \le u_2 < u_3$. There are totally $\binom{2}{2} \times (2^3)^3$ such codewords.

Considering both Case A and Case B, it is easy to show Case C and Case D.

By using bent functions x_1x_2 and $x_1x_2 + 1$, it is easy to verify by computer that the size of C is $2 \times [\binom{2^4}{2} + \binom{2^3}{2} \times 8 + \binom{2^2}{2} \times \times (8)^2 + \binom{2}{2} \times (8)^3] = 2480$. The minimum distance of C is 4 and PAPR(C)=1.

In a similar way, for general n we can have the following theorem.

Theorem 3. *The following constant amplitude code can be constructed.*

Case 1. If only using $x_1x_2 + c_1x_1 + c_2x_2 + c_0$ and $x_1x_2 + c_1x_1 + c_2x_2 + c_0 + 1$, where c_0, c_1, c_2 are in $GF(2)$, then the size of C is $2 \times [\binom{2^{n-1}}{2} + \binom{2^{n-2}}{2} \times 8 + \cdots + \binom{2}{2} \times 8^{n-2}]$. The minimum distance of C is 4 and PAPR(C)=1.

Case 2. If using all $x_1x_2 + c_1x_1 + c_2x_2 + c_0$ *, where* c_0, c_1, c_2 *are in* $GF(2)$*, then the size of* C *is* $[8 \times [\binom{2^{n-2}}{2} + \binom{2^{n-2}}{2} \times 8 + \cdots + \binom{2}{2} \times 8^{n-2}]$*. The minimum distance of* C *is 2 and PAPR(*C*)=1.*

Let $m = 5$, f a bent function over V_4. Suppose $g \in C$. Let us define g as follows,

Weight(g)=16; $g(u_0) = (-1)^{f(0,0,0,0)}$; $g(u_1) = (-1)^{f(1,0,0,0)}$; $g(u_2) = (-1)^{f(0,1,0,0)}; \cdots; g(u_{15}) = (-1)^{f(1,1,1,1)}$. Here $\{u_0, u_1, u_2, \cdots, u_{15}\} \subset V_5$ and $u_0 < u_1 < u_2 < \cdots < u_{15}$, such that PAPR($g$) = 1. In a similar way, one can consider a code C of length $n = 2^5$.

It is shown in [8] that an information bit stream with 9 information bits can be transmitted with constant amplitude by using the Hadamard code sequences of length 2^4. It follows from the fact that the number of bent functions over V_4 is 896.

From Theorem 3, by using all 8 bent functions over V_2, we know that the number of constant amplitude codewords of length 2^4 is $8 \times [\binom{2^2}{2} + \binom{2^2}{2} \times 8 + \binom{2}{2} \times 8^2] = 1120$.

Note that $2^{10} < 1120 < 2^{11}$, thus 10 information bits can be transmitted with constant amplitude codewords of length 2^4.

Similarly, we know that the number of constant amplitude codewords of length 2^3 is $8 \times [\binom{2^2}{2} + \binom{2}{2} \times 8] = 112$.

Thus, 6 bits can be transmitted with constant amplitude.

References

1. Paterson, K.G.: On Codes with Low Peak-To-Average Power Ratio for Multi-Code CDMA. IEEE Trans. Inform. Theory 50(3), 550-559 (2004)
2. Paterson, K.G.: generalised Reed-Muller Codes and Power Control in OFDM Modulation. IEEE Trans. Inform. Theory 46, 104-120 (2000)
3. Paterson, K.G., Jones, A.E.: Effcient Decoding Algorithms for Generalised Reed-Muller Codes. IEEE Trans. Commun. 48(8), 1272-1285 (2000)
4. Paterson, K.G., Tarokh, V.: on the Existence and Construction of Good Codes with Low Peak-To-Average Power Ratios. IEEE Trans. Inform. Theory 46(6), 1974-1987 (2000)
5. Qu, C., Seberry, J., Pieprzyk, J.: Homogeneous Bent Functions. Discrete Applied Mathematics 102, 133-139 (2000)
6. Rothaus, O.S.: On "Bent" Functions. J. Combin. Theory Ser.A 20, 300-305 (1976)
7. Xia, T., Seberry, J., Pieprzyk, J., Charnes, C.: Homogeneous Bent Functions pf Degree n in $2n$ Variables Do Not Exist for $n > 3$. Discrete Applied Mathematics 142, 127-132 (2004)
8. Wada, T: Characteristic of Bit Sequences Applicable to Constant Amplitude Orthogonal Multicode Systems. IEICE Trans. Fundamentals E83-A(11), 2160-2164 (2000)
9. Wada, T., Yamazato, M., Ogawa, A.: A Constant Amplitude Coding for Orthogonal Multi-Code CDMA Systems. IEICE Trans. Fundamentals E80-A(12), 2477-2484 (1997)
10. Wada, T., Yamazato, T., Katayama, M., Ogawa, A.: Error Correcting Capability of Constant Amplitude Coding for Orthogonal Multi-Code CDMA Systems. IEICE Trans. Fundamentals E81-A(10), 2166-2169 (1998)

Determining the Nonlinearity of a New Family of APN Functions

Carl Bracken*, Eimear Byrne**, Nadya Markin**, and Gary McGuire**

School of Mathematical Sciences
University College Dublin
Ireland
carlbracken@yahoo.com, ebyrne@ucd.ie, nadyamarkin@gmail.com,
gary.mcguire@ucd.ie

Abstract. We compute the Walsh spectrum and hence the nonlinearity of a new family of quadratic multi-term APN functions. We show that the distribution of values in the Walsh spectrum of these functions is the same as the Gold function.

Key words: Almost perfect nonlinear, APN, almost bent, AB, nonlinearity, Walsh transform, Walsh spectrum, discrete binary Fourier transform.

1 Introduction

Let $L = GF(2^n)$ for some positive integer n and let $\mathrm{Tr}(x)$ denote the absolute trace map on L. Let $f : L \longrightarrow L$ be a function. The map f is said to be *almost perfect nonlinear* (APN) on L if the number of solutions in L of the equation

$$f(x+q) - f(x) = p \tag{1}$$

is at most 2, for all $p, q \in L$, $q \neq 0$. If Equation (1) has at most r solutions, then the function is called differentially r-uniform. Therefore APN functions are also called differentially 2-uniform. It is clear that Equation (1) cannot have just one solution in even characteristic: for any solution x_0, there is a corresponding solution $x_0 + q$. In odd characteristics, however, it is possible to have functions permitting just one solution in the equation for all $p, q \in L$, $q \neq 0$. Such functions are called *perfect nonlinear*.

Definition 1. *The Walsh transform of f at (a, b) is defined by*

$$f^W(a,b) := \sum_{x \in L} (-1)^{\mathrm{Tr}(ax+bf(x))},$$

for each $a, b \in L$.

* Research supported by Irish Research Council for Science, Engineering and Technology Postdoctoral Fellowship.

** Research supported by the Claude Shannon Institute, Science Foundation Ireland Grant 06/MI/006.

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 72–79, 2007.
© Springer-Verlag Berlin Heidelberg 2007

The Walsh transform of $f(x)$ at (a, b) is the discrete binary Fourier transform of $g(x) = (-1)^{\mathrm{Tr}(bf(x))}$ at a.

The Walsh spectrum of f is the set

$$\Lambda_f := \{f^W(a, b) : a, b \in L, b \neq 0\}.$$

A quantity related to the Walsh spectrum is the the *nonlinearity* of f, defined as

$$NL(f) := 2^{n-1} - \frac{1}{2}\max\{|f^W(a, b)| : a, b \in L, b \neq 0\}.$$

If n odd and $\Lambda_f = \{0, \pm 2^{\frac{n+1}{2}}\}$, then the function has largest possible nonlinearity $2^{n-1} - 2^{\frac{n-1}{2}}$, and we say that f is *almost bent* (AB) or *maximally nonlinear*.

When n is odd, every AB function on L is also APN [7]. If f is quadratic (so that each of its exponents is of the form 2^i+2^j for some integers i, j) and f is also APN, then f is necessarily an AB function [6]. However, for a quadratic APN function defined on a field of even degree, the APN property does not determine its nonlinearity.

Vectorial Boolean functions used as S-boxes in block ciphers must have high nonlinearity and low differential uniformity in order to be resistant to linear [7] and differential [11] cryptanalysis. The AES (advanced encryption standard) uses a differentially 4-uniform function on eight variables called the inverse function. Defined on a field of odd degree this function is APN, however, for implementation reasons S-boxes use functions defined on an even number of variables. In AES, the inverse function was chosen above the possible APN functions as it is a permutation and at present there are no known APN permutations defined on fields of even degree. This function also has the best known nonlinearity for vectors of even length, that is $2^{n-1} - 2^{\frac{n}{2}}$.

Carlet-Charpin-Zinoviev (CCZ) equivalence, introduced in [6], is a standard measure to determine whether or not two APN functions are essentially the same. This relation generalizes *extended affine* (EA) equivalence. A pair of CCZ equivalent functions have the same resistance to linear and differential cryptanalysis. A family of APN functions is determined to be new if its members are CCZ inequivalent to functions of any previously known family.

Until recently, all known APN functions had been found to be EA equivalent to one of a short list of monomial functions, namely the Gold, Kasami-Welch, inverse, Welch, Niho and Dobbertin functions. For some time it was conjectured that this list was the complete list of APN functions up to EA equivalence. The Gold and Kasami-Welch functions are APN on fields of even and odd degree. When the field has even degree both these functions have the same nonlinearity as the inverse function.

In 2006, new examples of APN functions began to appear in the literature. In [10] the function $x^3 + \theta x^{36}$, with θ having order 3, was shown to be APN on $GF(2^{10})$ and CCZ inequivalent to any power mapping. This function has not been generalised to an infinite family. An infinite family of APN binomials on fields $GF(2^n)$, where n is divisible by 3 but not 9, was presented in [2] and shown to be EA inequivalent to any power mapping, and CCZ inequivalent to the Gold, Kasami-Welch, inverse and Dobbertin functions in [3].

It is shown in [4] that if the Walsh spectrum of a quadratic APN function is limited to the five values $\{0, \pm 2^{\frac{n}{2}}, \pm 2^{\frac{n+2}{2}}\}$, then the distribution of these values is the same as the Gold function. Not all quadratic APN functions have this property: a counterexample is given in [9]. Let u be primitive in $GF(2^6)$. Then

$$g(x) = x^3 + u^{11}x^5 + u^{13}x^9 + x^{17} + u^{11}x^{33} + x^{48}$$

is a quadratic APN function on $GF(2^6)$ whose Walsh transform takes 7 distinct values.

In [1] two new APN families are presented, one a trinomial which generalizes the binomial from [2] and the other a multiterm polynomial defined on all fields of degree divisible by 2 but not 4. In this article we show that the second of these two new functions has the same Walsh spectrum (and indeed the same spectral distribution) as a Gold function.

2 A New Family of APN Functions

Let k, s be a pair of odd relatively prime integers. In [1] the following quadratic function was shown to be APN over the field $GF(2^{2k})$:

$$F(x) := \alpha x^{2^s+1} + \alpha^{2^k} x^{2^{k+s}+2^k} + \beta x^{2^k+1} + \sum_{i=1}^{k-1} \gamma_i x^{2^{k+i}+2^i}, \tag{2}$$

where α and β are primitive elements of $GF(2^{2k})$ and $\gamma_i \in GF(2^k)$ for each i.

Observe that since $GF(2^{2k})$ has even degree, the Walsh spectrum of F is not determined even though it is APN. Our main result is that F has the same Walsh spectrum as a Gold function. Before we compute the Walsh spectrum of this family of functions we need the following lemma and its corollary. While both of these results are well-known, we include proofs here for the convenience of the reader.

Let K be a field and let H_1, H_2 be extensions of K that are both subfields of a field extension M of K. We say that H_1 is linearly disjoint from H_2 over K if any set of K-linearly independent elements of H_1 forms a H_2-linearly independent set in M.

Lemma 1. *Let n, s be positive integers satisfying $(n, s) = 1$. Let K be a field and let H_1, H_2 be finite extensions of K of degrees n and s respectively. Let M be the compositum of H_1 and H_2. Then H_1 and H_2 are linearly disjoint over K.*

Proof. Let $\mathcal{S} = \{c_1, \ldots, c_t\}$ be a set of K-linearly independent elements of H_1. We'll show that $\mathcal{S}$ is linearly independent in M over H_2.

Let $\{a_1, \ldots, a_n\}$ be a K-basis of H_1 and let $\{b_1, \ldots, b_s\}$ a K-basis of H_2. Then the set $\{a_i \cdot b_j \mid 1 \le i \le n, 1 \le j \le s\}$ generates M as a vector space over K. It is clear that the set $\{a_1 \cdot b_1, \ldots, a_n \cdot b_1\}$ generates M as a vector space over the field H_2, and hence $[M : H_2] \le n$. On the other hand, since $(n, s) = 1$, $[M : H_2] = n$.

Moreover, without loss of generality we may assume that $b_1 = 1$ from which we conclude that $\{a_1, \ldots, a_n\}$ is a basis of M over H_2.

Now extend $\mathcal{S}$ to a basis $\{c_1, \ldots, c_t, \ldots, c_n\}$ of H_1 over K. Since this set forms a H_2-basis of M, its subset $\{c_1, \ldots, c_t\}$ is a fortiori linearly independent over H_2. □

Corollary 1. *Let d, n, s be positive integers satisfying $(n, s) = 1$ and let*

$$g(x) = \sum_{i=0}^{d} r_i x^{2^{si}} \in L[x].$$

Then the equation $g(x) = 0$ has at most 2^d solutions in L.

Proof. Let V denote the set of zeroes of $g(x)$ in L. We may assume that $V \neq \{0\}$. Since $g(x)$ is a linearized polynomial, V is a vector space over $GF(2)$ of finite dimension v for some positive integer v. Let $V' \subset GF(2^{sn})$ denote the vector space generated by the elements of V over the field $GF(2^s)$. Since $(n, s) = 1$, by Lemma 1, V' is a v-dimensional vector space over $GF(2^s)$. Furthermore, for all $c \in GF(2^s)$ and $w \in GF(2^{sn})$ we have $g(cw) = cg(w)$. Therefore all the elements of V' are also solutions of $g(x) = 0$. Since the dimension of V over $GF(2)$ is v, the size of V' is 2^{sv} and it follows that there are at least 2^{sv} zeroes of $g(x)$ in $GF(2^{sn})$. On the other hand, a polynomial of degree 2^{ds} can have at most 2^{ds} zeroes. We conclude that $v \leq d$. □

We now prove our main result.

Theorem 1. *Let $n = 2k$ and let $F(x)$ be defined on $L = GF(2^{2k})$ as in (2). The Walsh spectrum of $F(x)$ is $\{0, \pm 2^{\frac{n}{2}}, \pm 2^{\frac{n+2}{2}}\}$.*

Proof. By definition, we have

$$F^W(a, b) = \sum_{x \in L} (-1)^{\mathrm{Tr}(ax + bF(x))}.$$

The square of which is given by

$$|F^W(a, b)|^2 = \sum_{x \in L} \sum_{u \in L} (-1)^{\mathrm{Tr}(ax + bF(x) + a(x+u) + bF(x+u))},$$

An explicit representation of $F(x)$ in the first of these expressions gives

$$F^W(a, b) = \sum_{x \in L} (-1)^{\mathrm{Tr}(ax + b(\alpha x^{2^s+1} + \alpha^{2^k} x^{2^{k+s}+2^k} + \beta x^{2^k+1} + \sum \gamma_i x^{2^{k+i}+2^i}))}.$$

Using the fact that $\mathrm{Tr}(\theta) = \mathrm{Tr}(\theta^2)$ for any $\theta \in L$, this can be written as

$$F^W(a, b) = \sum_{x \in L} (-1)^{\mathrm{Tr}(ax + (b + b^{2^k})\alpha x^{2^s+1} + (b\beta + \sum b^{2^{-i}} \gamma_i^{2^{-i}}) x^{2^k+1})}.$$

Therefore, the square is given by

$$|F^W(a,b)|^2 = \sum_{u\in L}(-1)^{\mathrm{Tr}(au+bF(u))}\sum_{x\in L}(-1)^{\mathrm{Tr}(x\mathcal{L}_b(u))}$$

where

$$\mathcal{L}_b(u) = cu^{2^s} + c^{2^{-s}}u^{2^{-s}} + eu^{2^k}$$

for $c = (b + b^{2^k})\alpha$ and $e = b\beta + b^{2^k}\beta^{2^k} + \sum(b^{2^{-i}}\gamma_i{}^{2^{-i}} + b^{2^{k-i}}\gamma_i{}^{2^{k-i}})$. First we make some observations about the coefficients of this equation. As $e = e^{2^k}$, we can say that $e \in GF(2^k)$. Note also that as k is odd, all elements of $GF(2^k)$ are cubes in L. It is clear that $b + b^{2^k} \in GF(2^k)$ and is therefore a cube. If $b \notin GF(2^k)$ then $c = (b + b^{2^k})\alpha$ and hence c is not a cube since α is primitive. On the other hand, if $b \in GF(2^k)$ then $c = 0$.

Recall that for any character χ of a group G, we have

$$\sum_{g\in G}\chi(g) = \begin{cases} |G| & \text{if } \chi \text{ is trivial} \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

We apply this simple principle twice here. First note that for any $R \in L$, $x \mapsto (-1)^{\mathrm{Tr}(Rx)}$ is a character of L, which gives

$$\sum_{x\in L}(-1)^{\mathrm{Tr}(Rx)} = \begin{cases} 2^n & \text{if } R = 0 \\ 0 & \text{otherwise.} \end{cases}$$

and hence

$$|F^W(a,b)|^2 = 2^n \sum_{u\in\mathcal{K}_b}(-1)^{\mathrm{Tr}(au+bF(u))},$$

where $\mathcal{K}_b$ is the kernel of $\mathcal{L}_b(u)$. Secondly, it is easy to verify that

$$\chi_{a,b} : u \mapsto (-1)^{\mathrm{Tr}(au+bF(u))},$$

is a character of $\mathcal{K}_b$, from which we deduce that

$$|F^W(a,b)| = \begin{cases} \sqrt{2^n|\mathcal{K}_b|} & \text{if } \chi_{a,b}(u) = 1 \ \forall\ u \in \mathcal{K}_b \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Furthermore, as n is even and $|F^W(a,b)|$ is an integer, we know that $|\mathcal{K}_b|$ must be an even power of 2. Therefore, in order to demonstrate that $F(x)$ has a five valued spectrum, we need only to show that $|\mathcal{K}_b| < 16$.

First we demonstrate that $\mathcal{L}_b(u) = cu^{2^s} + c^{2^{-s}}u^{2^{-s}} + eu^{2^k}$ has no more than 4 zeroes when either c or e is zero. If $c = (b + b^{2^k})\alpha = 0$ then $b \in GF(2^k)$ and hence

$$\begin{aligned} e &= b\beta + b^{2^k}\beta^{2^k} + \sum_{i=0}^{k-1}(b^{2^{-i}}\gamma_i{}^{2^{-i}} + b^{2^{k-i}}\gamma_i{}^{2^{k-i}}) \\ &= b(\beta + \beta^{2^k}) + \sum_{i=0}^{k-1} b^{2^{-i}}(\gamma_i{}^{2^{-i}} + \gamma_i{}^{2^{k-i}}) = b(\beta + \beta^{2^k}), \end{aligned}$$

since $\gamma_i \in GF(2^k)$ for each i. Then $e \neq 0$ since otherwise $\beta \in GF(2^k)$, contradicting our choice of β as primitive in L. Then $\mathcal{L}_b(u) = eu^{2^k}$, which has only $u = 0$ as a root.

Now suppose $e = 0$. Then $c \neq 0$ and $\mathcal{L}_b(u) = cu^{2^s} + c^{2^{-s}}u^{2^{-s}}$. Then $\mathcal{L}_b(u) = 0$ implies $c^{2^s}u^{2^{2s}} + cu = 0$. As $(s, n) = 1$, this has no more than 4 solutions by Corollary 1.

We now assume that both c and e are non-zero. Let u be an arbitrary nonzero element of $\mathcal{K}_b$. Consider

$$u\mathcal{L}_b(u) = cu^{2^s+1} + c^{2^{-s}}u^{2^{-s}+1} + eu^{2^k+1} = 0.$$

It is clear that $eu^{2^k+1} \in GF(2^k)$. Hence $\mathcal{L}_b(u) = 0$ implies $cu^{2^s+1} + c^{2^{-s}}u^{2^{-s}+1} \in GF(2^k)$ and that

$$cu^{2^s+1} + c^{2^{-s}}u^{2^{-s}+1} = c^{2^k}u^{2^{k+s}+2^k} + c^{2^{k-s}}u^{2^{k-s}+2^k}.$$

Rearranging the terms gives

$$cu^{2^s+1} + c^{2^k}u^{2^{k+s}+2^k} = (cu^{2^s+1} + c^{2^k}u^{2^{k+s}+2^k})^{2^{-s}}.$$

Then $cu^{2^s+1} + c^{2^k}u^{2^{k+s}+2^k} \in GF(2^s)$ and as s and n are relatively prime, we have $cu^{2^s+1} + c^{2^k}u^{2^{k+s}+2^k} \in GF(2)$. If $cu^{2^s+1} + c^{2^k}u^{2^{k+s}+2^k} = 0$, then $cu^{2^s+1} \in GF(2^k)$ which implies it is a cube in $GF(2^{2k})$. However, u^{2^s+1} is a cube (as s is odd) which means c must be a cube (when $u \neq 0$), contradicting an earlier claim. Therefore, we may assume that u is a nonzero element of $\mathcal{K}_b$ if and only if it is a root of

$$G(u) := cu^{2^s+1} + c^{2^k}u^{2^{k+s}+2^k} + 1.$$

The following trick, taken from [8], is useful when dealing with an expression that contains non-linear terms but is known to have a solution set that forms an additive group. It exploits the fact that the term cu^{2^s} in $u^{-1}G(u)$ is a linear power and will therefore cancel in the computations. Now choose some fixed nonzero $v \in \mathcal{K}_b$ that is different from u and consider the expression

$$uv(u+v)(u^{-1}G(u) + v^{-1}G(v) + (u+v)^{-1}G(u+v)) = 0,$$

which is equal to

$$c^{2^k}(u^{2^{k+s}}v + v^{2^{k+s}}u)(u^{2^k}v + v^{2^k}u) + u^2 + v^2 + uv = 0. \tag{5}$$

For fixed nonzero $v \neq u$ in $\mathcal{K}_b$, the expression on the left in (5) is a polynomial in indeterminate u, whose set of zeroes $\mathcal{S}$ contains $\mathcal{K}_b$. Let $u = vw$ and divide (5) by v^2 to obtain

$$c^{2^k}v^{2^{k+s}+2^k}(w + w^{2^{k+s}})(w + w^{2^k}) + w^2 + w + 1 = 0. \tag{6}$$

Observe that there is a one-to-one correspondence between the solution set $v^{-1}\mathcal{S}$ of (6) and the solution set $\mathcal{S}$ of (5). We will obtain an upper bound on the size of $v^{-1}\mathcal{S}$ in L and hence on $|\mathcal{K}_b|$.

If $w_0 \in GF(2^k)$ is a solution to (6) then substituting ${w_0}^{2^k} = w_0$ in (6) gives ${w_0}^2 + w_0 + 1 = 0$, in which case $w_0 \in GF(4) \backslash GF(2)$. Since k is odd, this means $w_0 \notin GF(2^k)$, giving a contradiction. It follows that any solution w to (6) satisfies $w + w^{2^k} \neq 0$.

Raising (6) to the 2^k-th power and adding it to (6) we get

$$(w+w^{2^k})(cv^{2^s+1}(w^{2^k}+w^{2^s})+c^{2^k}v^{2^{k+s}+2^k}(w+w^{2^{k+s}}))+(w+w^{2^k})^2+(w+w^{2^k}) = 0. \tag{7}$$

The assumption $w + w^{2^k} \neq 0$ means we can divide (7) by $w + w^{2^k}$ to get

$$cv^{2^s+1}(w^{2^k} + w^{2^s}) + c^{2^k}v^{2^{k+s}+2^k}(w + w^{2^{k+s}}) + (w + w^{2^k}) + 1 = 0. \tag{8}$$

As $v \in \mathcal{K}_b$, it obeys the expression $G(v) = cv^{2^s+1} + c^{2^k}v^{2^{k+s}+2^k} + 1 = 0$, and so (8) becomes

$$cv^{2^s+1}(w + w^{2^s}) + c^{2^k}v^{2^{k+s}+2^k}(w^{2^k} + w^{2^{k+s}}) + 1 = 0. \tag{9}$$

Any element of $v^{-1}\mathcal{S}$ is a solution of (9), so an upper bound on the size of the solution set of (9) will give an upper bound on the size of $\mathcal{K}_b$.

Now return to our original equation $\mathcal{L}_b(u) = 0$ and replace u with vw to get

$$\mathcal{L}_b(vw) = cv^{2^s}w^{2^s} + c^{2^{-s}}v^{2^{-s}}w^{2^{-s}} + ev^{2^k}w^{2^k} = 0. \tag{10}$$

From this we derive two equations. First, a simple rearrangement of (10) gives

$$w^{2^k} = e^{-1}(cv^{2^s-2^k}w^{2^s} + c^{2^{-s}}v^{2^{-s}-2^k}w^{2^{-s}}). \tag{11}$$

Next take (11) to the power of 2^s to get

$$w^{2^{k+s}} = e^{-2^s}(c^{2^s}v^{2^{2s}-2^{k+s}}w^{2^{2s}} + cv^{1-2^{k+s}}w). \tag{12}$$

Substituting the expressions of w^{2^k} and $w^{2^{k+s}}$ given by (11) and (12) respectively, into (9), we obtain the following:

$$cv^{2^s+1}(w + w^{2^s}) + c^{2^k}v^{2^{k+s}+2^k}(e^{-1}(cv^{2^s-2^k}w^{2^s} + c^{2^{-s}}v^{2^{-s}-2^k}w^{2^{-s}})$$

$$+e^{-2^s}(c^{2^s}v^{2^{2s}-2^{k+s}}w^{2^{2s}} + cv^{1-2^{k+s}}w)) + 1 = 0.$$

Take a 2^s-th power of this equation to obtain an equation of the form

$$r_0w + r_1w^{2^s} + r_2w^{2^{2s}} + r_3w^{2^{3s}} + 1 = 0,$$

which has the same number of solutions as there are zeroes of the linearized polynomial

$$r_0w + r_1w^{2^s} + r_2w^{2^{2s}} + r_3w^{2^{3s}}. \tag{13}$$

Since, for example,

$$r_0 = c^{2^{k+s}+1} e^{-2^s} v^{2^{k+2s}-2^{k+s}+1} \neq 0,$$

for nonzero c, e and v, the polynomial in (13) is not identically zero. Then by Corollary 1, (13) can have no more than eight zeroes in w. Furthermore, as $|\mathcal{K}_b|$ must be an even power of 2, it follows that $|\mathcal{K}_b| \leq 4$ and we conclude from (4) that $F^W(a,b) \in \{0, \pm 2^{\frac{n}{2}}, \pm 2^{\frac{n+2}{2}}\}$. □

Recall that if the Walsh spectrum of a quadratic APN function is limited to the five values $\{0, \pm 2^{\frac{n}{2}}, \pm 2^{\frac{n+2}{2}}\}$, then the distribution of these values is the same as the Gold function. We therefore have the following immediate corollary to our theorem.

Corollary 2. *Let $F(x)$ be defined as in* (2)*. Then the Walsh spectrum of $F(x)$ has the same distribution as the Gold function.*

This is equivalent to saying that exactly $\frac{2}{3}(2^n - 1)$ of the Boolean functions $\mathrm{Tr}(bF(x))$ are bent, the fewest number possible.

References

1. Bracken, C., Byrne, E., Markin, N., McGuire, G.: New Families of Quadratic Almost Perfect Nonlinear Trinomials and Multinomials (preprint, 2007)
2. Budaghyan, L., Carlet, C., Felke, P., Leander, G.: An Infinite Class of Quadratic APN Functions Which Are Not Equivalent to Power Mappings. In: 2006 IEEE Internation Symposium on Information Theory, IEEE Press, New York (2006)
3. Budaghyan, L., Carlet, C., Leander, G.: A Class of Quadratic APN Binomials Inequivalent to Power Functions (preprint, 2007)
4. Carlet, C.: Vectorial Boolean Functions for Cryptography. In: Hammer, P., Crama, Y. (eds.) Boolean methods and models, Cambridge University Press, Cambridge
5. Canteaut, A., Charpin, P., Dobbertin, H.: Weight Divisibility of Cyclic Codes, Highly Nonlinear Functions on $GF(2^m)$ and Crosscorrelation of Maximum-Length Sequences. SIAM J. Discrete Mathematics 13(1), 105–138 (2000)
6. Carlet, C., Charpin, P., Zinoviev, V.: Codes, Bent Functions and Permutations Suitable for Des-Like Cryptosystems. Designs, Codes and Cryptography 15(2), 125–156 (1998)
7. Chabaud, F., Vaudenay, S.: Links Between Differential and Linear Cryptanalysis. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 356–365. Springer, Heidelberg (1995)
8. Dobbertin, H.: Another Proof of Kasami's Theorem. Designs, Codes and Cryptography 17, 177–180 (1999)
9. Dillon, J.: Polynomials over Finite Fields and Applications. Slides from talk given at Banff International Research Station (2006)
10. Edel, Y., Kyureghyan, G., Pott, A.: A New APN Function Which is not Equivalent to a Power Mapping. IEEE Trans. Inform. Theory 52(2), 744–747 (2006)
11. Nyberg, K.: Differentially Uniform Mappings for Cryptography. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 55–64. Springer, Heidelberg (1994)

An Improvement of Tardos's Collusion-Secure Fingerprinting Codes with Very Short Lengths*

Koji Nuida[1], Satoshi Fujitsu[2], Manabu Hagiwara[1], Takashi Kitagawa[1], Hajime Watanabe[1], Kazuto Ogawa[2], and Hideki Imai[1,3]

[1] Research Center for Information Security (RCIS), National Institute of Advanced Industrial Science and Technology (AIST); Akihabara-Daibiru Room 1102, 1-18-13 Sotokanda, Chiyoda-ku, Tokyo 101-0021, Japan
{k.nuida, hagiwara.hagiwara, t-kitagawa, h-watanabe, h-imai}@aist.go.jp
[2] Science & Technical Research Laboratories, Japan Broadcasting Corporation (NHK); 1-10-11 Kinuta, Setagaya-ku, Tokyo 157-8510, Japan
{fujitsu.s-hc, ogawa.k-cm}@nhk.or.jp
[3] Faculty of Science and Engineering, Chuo University 1-13-27 Kasuga, Bunkyo-ku, Tokyo 112-8551, Japan

Abstract. The code length of Tardos's collusion-secure fingerprinting code (STOC'03) is of theoretically minimal order with respect to the number of malicious users (pirates); however, the constant factor should be further reduced for practical implementation. In this paper we give a collusion-secure fingerprinting code by mixing recent two improvements of Tardos code and modifying their pirates tracing algorithms. Our code length is significantly shorter than Tardos code, especially in the case of fewer pirates. For example, the ratio of our length relative to Tardos code in some practical situation with 4 pirates is 4.33%; while the lowest among the preceding codes in this case (Škorić et al., 2007) is 9.87%.

1 Introduction

Recent development of computer and network technologies rapidly grows amount of trades of digital contents. This has increased not only convenience for both contents servers and users, but also risks of the contents being illegally copied and redistributed without permission. Although many DRM technologies have been introduced so far, most of them prevent copying of contents even by right users, so these may decrease the convenience for innocent users too much.

Fingerprinting codes are technology that is hoped to be an alternative solution. It does not prevent an innocent user to copy his contents for his own use; while any malicious user (called a *pirate*) illegally redistributing his contents will be detected from the *fingerprinting codeword* embedded into each content by the contents server beforehand. However, even if each codeword is embedded as being unrecognizable by a single user, there exists a strong *collusion-attack*

* This study has been sponsored by the Ministry of Economy, Trade and Industry, Japan (METI) under contract, New-generation Information Security R&D Program.

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 80–89, 2007.
© Springer-Verlag Berlin Heidelberg 2007

by two or more pirates [1]; they can recognize parts of the embedded codewords from differences of their contents and then modify or erase these parts. Thus any practical fingerprinting code should be prudently designed and be equipped with a *tracing algorithm*, which can detect at least one pirate even from the modified codeword (called a *pirated codeword*). A fingerprinting code is called *c-secure with ε-error* if the tracing algorithm fails (i.e. detects either no pirate or some innocent user) with probability at most ε when there are up to c pirates.

Tardos [9] recently gave a uniform construction of c-secure codes (Tardos codes) for all c, whose code length is of theoretically minimal order with respect to c. There have been several variants and improvements of Tardos codes; e.g. [2,3,5,6,7,8]. Their common idea in the tracing algorithm is to first assign a "score" to each user, which measures how his codeword is similar to the pirated codeword, and then output *all* users as pirates whose score exceeds a suitably determined threshold. However, such a scheme seems not to be optimal, since if the score of a pirate is much higher than the threshold and that of an innocent user is only slightly higher than the threshold, then the latter user is also accused though most suspect user is obviously the former.

In this paper, we modify such a tracing algorithm as outputting *just one* user with highest score instead. Then we investigate sufficient code lengths to make our code c-secure with ε-error; e.g. in some practical setting, it is about 3.01% of length of Tardos code for $c = 2$ and 4.33% of Tardos code for $c = 4$ (Table 4). Our numerical examples show that our code lengths are significantly shorter than the preceding improvements of Tardos code.

This paper is organized as follows. Section 2 summarizes our model for fingerprinting codes, with a slightly weaker version of the widely adopted Marking Assumption (cf. [1]), and construction of Tardos codes. Section 3 gives our code construction, which is a mixture of [7] and [8], and aforementioned tracing algorithm. Section 3 also gives a bound of tracing error probability and a formula of code lengths. Section 4 shows some numerical examples and comparison of code lengths with Tardos code and preceding improvements. Finally, an appendix is given in the last of this paper and devoted to the proof of our main results.

2 Preliminary

2.1 Our Model for Fingerprinting Codes

In our model, a contents server embeds a binary codeword $w_i = (w_{i,1}, \ldots, w_{i,m})$ of length m into the content distributed to i-th user u_i. When a pirated content involving *pirated codeword* $y = (y_1, \ldots, y_m)$ is found, the contents server perform some *tracing algorithm* with y and all the codewords w_i as input for detecting the pirates. It should be designed for decreasing as much as possible the tracing error probability, i.e. the probability that either no pirate or some innocent user is detected. Note that it is possible for some bits in y to be undecodable; such bits are denoted by '?'.

If there are two or more pirates, they can recognize some parts of the embedded codewords by finding differences of their contents, and then modify or erase

codewords in these positions by certain strategy. Here we put the following two assumptions for the pirates' strategy:

δ-Marking Assumption. The number of positions $1 \leq j \leq m$ with $w_{1,j} = w_{2,j} = \cdots = w_{\ell,j}$ (such positions are called *undetectable*; where $w_1, \ldots, w_\ell$ are the pirates' codewords) and $y_j \neq w_{1,j}$ does not exceed $m\delta$.

Pirates' Knowledge. Pirates have no knowledge about innocent (non-pirate) users' codewords; so y is chosen independently of those codewords.

Remark 1. Our δ-Marking Assumption is a relaxation of the classical Marking Assumption [1] (corresponding to the case $\delta = 0$). Our δ-Marking Assumption allows some undetectable bits in y to be flipped or erased. This is motivated by difficulty of the Marking Assumption being satisfied in practical situations.

A fingerprinting code is called *c-secure* (*with ε-error*) if the tracing error probability does not exceed a negligibly small value ε whenever the number ℓ of pirates is up to c; i.e. $\ell \leq c$.

2.2 Essence of Tardos Code and Its Generalizations

Essences of the construction and tracing algorithms of c-secure Tardos codes [9] and its generalizations [2,6,7] required here are summarized as follows. See the references for details.

First, the contents server chooses the random values $p^{(j)}$, $0 < p^{(j)} < 1$, independently for every $1 \leq j \leq m$ according to a given probability distribution $\mathcal{P}$ (referred to as the *bias distribution*). Tardos [9] used certain *continuous* bias distributions, while those used in [2,6,7] are *finite* and symmetric; i.e. outputting p and $1-p$ with the same probability. In this paper we follow the latter choice. The resulting sequence $P = (p^{(1)}, \ldots, p^{(m)})$ should be stored and be kept secret throughout the construction and the pirates tracing process (pirates are allowed to guess the values $p^{(j)}$ from their codewords and knowledge on the distribution $\mathcal{P}$, but not to know about the actual choices of $p^{(j)}$). Then, secondly, the server chooses bits $w_{i,j}$ of the codewords independently, in the following probabilistic manner: $\Pr(w_{i,j} = 1) = p^{(j)}$ and $\Pr(w_{i,j} = 0) = 1 - p^{(j)}$ for j-th position, where Pr denotes the probability.

In the tracing algorithm, the server computes a score S_i of each user u_i by $S_i = \sum_{j=1}^{n} S_i^{(j)}$, where $S_i^{(j)} = \sigma(p^{(j)})$ if $(y_j, w_{i,j}) = (1,1)$, $S_i^{(j)} = -\sigma(1-p^{(j)})$ if $(y_j, w_{i,j}) = (1,0)$ and $S_i^{(j)} = 0$ if $y_j \in \{0, ?\}$, with $\sigma(p) = \sqrt{(1-p)/p}$. The output of the tracing algorithm is the (possibly empty) set of *all* users u_i with $S_i \geq Z$, where Z is a suitably determined threshold parameter.

3 Our Contribution

3.1 Code Construction and Tracing Algorithm

Our c-secure fingerprinting code is constructed by slightly modifying the framework given in Sect. 2.2. First, we define the bias distribution $\mathcal{P} = \mathcal{P}_c$ as follows:

Definition 1. *Let* $L_k(t) = (\frac{d}{dt})^k(t^2-1)^k/(k!\,2^k)$ *be the* k*-th Legendre polynomial, and* $\widetilde{L}_k(t) = L_k(2t-1)$*. Then define* $\mathcal{P}_{2k-1} = \mathcal{P}_{2k}$ *to be the finite probability distribution whose values are the* k *zeroes of* $L_k(t)$*, with each value* p *taken with probability* $C \cdot \big(p(1-p)\big)^{-3/2}\widetilde{L}'_k(p)^{-2}$*, where* C *is the normalized constant.*

The above bias distribution was introduced in [6,7] for optimizing the memory amount required to record the sequence P. Note that we would have to use some approximation instead of the original $\mathcal{P}_c$ in practical implementation, since the values of $\mathcal{P}_c$ and corresponding probabilities are irrational numbers in general.

Secondly, the scoring rule in the tracing algorithm is modified as follows: the bitwise score $S_i^{(j)}$ is left unchanged if $y_j = 1$, but it is changed in the case $y_j \in \{0, ?\}$ so that $S_i^{(j)} = \sigma(1-p^{(j)})$ if $w_{i,j} = 0$ and $-\sigma(p^{(j)})$ if $w_{i,j} = 1$. (Such a "symmetric" scoring rule was introduced in [5,8] to reduce the lengths of Tardos codes). Again, we would have to use approximated values of these bitwise scores in practical implementation.

Moreover, in contrast with the original tracing algorithm that outputs all users with $S \geq Z$, our tracing algorithm does not use the threshold Z and outputs *just one* user whose score is the highest. (The way of choosing one user in the case that two or more users have the same highest score may be arbitrary, since it is not relevant to our security proof below.) Then a simple and easy argument can prove the following fact, which assures the merit of our modification.

Proposition 1. *The tracing error probability of our code with the modified tracing algorithm (i.e. without threshold) does not exceed that with the original tracing algorithm (i.e. using threshold).*

3.2 Tracing Error Probability and Code Lengths

From now, we give a bound of tracing error probability of our code, and then show a formula of code length which makes our code c-secure with ε-error.

Before stating the result, we prepare some further notations and terminology. Let p_0 denote the minimal possible output of a bias distribution $\mathcal{P}$, and put $\eta = \sigma(p_0)$. For $1 \leq \ell \leq c$ and $0 \leq x \leq \ell$, put

$$R_{\ell,x} = \max\left\{0, \mathrm{Ex}\left[p^x(1-p)^{\ell-x}\big(x\sigma(p) - (\ell-x)\sigma(1-p)\big)\right]\right\} \ ,$$

where the expectation value Ex is taken over the values p of $\mathcal{P}$. Let

$$\mathcal{R}_\ell = \ell \mathrm{Ex}\left[(1-p)^{\ell-1/2}p^{1/2}\right] - \sum_{x=1}^{\ell-1}\binom{\ell}{x}R_{\ell,x} \ .$$

Let δ' denote the approximation error of the bitwise scores, and put $\Delta = \delta' + 2\eta\delta$, which is referred to as the *error tolerance rate* of our code (where δ is the same as that in δ-Marking Assumption). Then take a value $\mathcal{R}$ such that $2c\Delta \leq \mathcal{R} \leq \mathcal{R}_\ell$ for all $1 \leq \ell \leq c$. Moreover, define the following functions

$$B_1(t) = \frac{e^{t\eta} + \eta^2 e^{-t/\eta}}{\eta^2+1}\,,\ B_{2,\ell}(t) = 1 + \frac{e^{t\ell\eta} - 1 - t\ell\eta}{\ell\eta^2} - 2t\mathcal{R}\,,\ \Phi(t) = t(1-\log t)\,,$$

where $\log = \log_e$ denotes the natural logarithm, and put

$$T_c = B_1(\beta c) B_{2,c}(\beta) e^{2\beta c \Delta} \ .$$

Now we state the following result, which will be proved in the appendix. Some numerical examples concerning this result will be provided below.

Theorem 1. *Assume that there are at most N users, involving up to c pirates. Let $0 < \varepsilon < 1$, and let $\beta > 0$ be chosen so that $N {T_c}^m < 1$.*

1. *If $T_c \leq T_0$ and $N {T_0}^m < 1$, then the tracing error probability of our code is less than or equal to $\Phi(N {T_0}^m)$. Hence our code is c-secure with ε-error if $\Phi(N {T_0}^m) \leq \varepsilon$.*
2. *Let $a > 1$ such that $\varepsilon \leq a e^{1-a}$ (e.g. $a = 10/9$ if $\varepsilon \leq 0.99$). Then our code is c-secure with ε-error if*

$$m \geq -\frac{1}{\log T_c} \left(\log \frac{N}{\varepsilon} + \log \frac{a}{a-1} + \log \log \frac{a}{\varepsilon} \right) \ . \tag{1}$$

Remark 2. If $\Delta = \delta' + 2\eta\delta$ is getting larger, then T_c also becomes larger, so Theorem 1 infers that our code length becomes longer as well.

3.3 Choice of the Parameter β

In order to reduce code lengths, the parameter β should be chosen so that the value T_c becomes as small as possible. Since it seems to be hopeless to express the optimal β in a closed form for general case, here we give a "pretty good" closed formula of β instead.

Let $j_1 = 2.40482\cdots$ be the smallest positive zero of the Bessel function $J_0(t) = \sum_{k=0}^{\infty} (-1)^k (t/2)^{2k} / (k!)^2$. Then our formula of β is

$$\beta_{\mathrm{formula}} = \frac{1}{\eta^2 j_1} \log \left(1 + \frac{2\eta}{c} (\mathcal{R} - \eta j_1 \Delta) \right) \ .$$

It can be shown that this formula becomes optimal in the limit case $c \to \infty$ (the proof is omitted here due to limited pages, and will appear in the full version of this paper). Moreover, the following numerical example suggests that this formula approximates the optimal β well, at least in the case $c \in \{2, 4, 6, 8\}$.

4 Numerical Example

4.1 Our Approximation of Bias Distribution

In this section, we consider the cases $c \in \{2, 3, 4, 6, 8\}$. We use the approximation $\mathcal{P} = \mathcal{P}_c$ of the bias distributions defined in Definition 1 in the former part of Table 1. Here columns p and q denote the values of $\mathcal{P}_c$ and the corresponding probabilities, respectively. On the other hand, the latter part of Table 1 gives

approximation of bitwise scores, where $p_0 < p_1 < \cdots$ are possible values of $\mathcal{P}$ and U_j denotes the approximated value of $\sigma(p_j)$. (Note that $U_{\lceil c/2 \rceil - 1 - j}$, where $\lceil x \rceil$ denotes the smallest integer n with $n \geq x$, is an approximation of $\sigma(1-p_j)$.) The approximation error is $\delta' = 0$ if $c = 1, 2$ and $\delta' = 10^{-5}$ if $3 \leq c \leq 8$. Moreover, the values $\mathcal{R}$ and approximation of η for these cases are given in Table 2.

Table 1. Approximations of bias distributions $\mathcal{P} = \mathcal{P}_c$ and bitwise scores

c	p	q
1, 2	0.50000	1.00000
3, 4	0.21132	0.50000
	0.78868	0.50000
5, 6	0.11270	0.33201
	0.50000	0.33598
	0.88730	0.33201

c	p	q
7, 8	0.06943	0.24833
	0.33001	0.25167
	0.66999	0.25167
	0.93057	0.24833

c	U_0	U_1	U_2	U_3
2	1			
4	1.93187	0.51763		
6	2.80590	1	0.35639	
8	3.66101	1.42485	0.70182	0.27314

Table 2. Auxiliary values for our example

c	2	3	4	6	8
$\mathcal{R}$	0.50000	0.40823	0.40823	0.37796	0.36291
η	1.00000	1.93188	1.93188	2.80591	3.66102

Table 3. Length comparison under δ-Marking Assumption

Here $\Delta = 0.01$. Lengths in parentheses are computed by using β_{formula}.

c		Case 1	Case 2	Case 3	Case 4	β_{optimal}
2	Ours	**403**	**444**	**273**		0.16921
		(404)	(444)	(274)		
	Tardos	12400	14000	8400		
	%	**3.25**	**3.17**	**3.25**	**2.97**	
3	Ours	**1514**	**1646**	**1014**		0.057404
		(1630)	(1771)	(1091)		
	Tardos	28800	31500	18900		
	%	**5.26**	**5.23**	**5.37**	**4.89**	
4	Ours	**2671**	**2879**	**1774**		0.034093
		(2672)	(2880)	(1775)		
	Tardos	51200	56000	33600		
	%	**5.22**	**5.14**	**5.28**	**4.81**	
6	Ours	**7738**	**8244**	**5079**		0.013798
		(7743)	(8249)	(5082)		
	Tardos	115200	126000	75600		
	%	**6.72**	**6.54**	**6.72**	**6.13**	
8	Ours	**16920**	**17879**	**11015**		0.0071633
		(16934)	(17894)	(11024)		
	Tardos	211200	224000	134400		
	%	**8.01**	**7.98**	**8.20**	**7.47**	

4.2 Calculation and Comparison of Code Lengths

Table 3 shows code lengths of our code under δ-Marking Assumption. Here the error tolerance late $\Delta = \delta' + 2\eta\delta$ is set to 0.01; so slightly fewer than $m/(200\eta)$ undetectable bits are allowed to be flipped or erased. We consider the following three cases: (1) $N = 100c$ and $\varepsilon = 10^{-11}$; (2) $N = 10^9$ and $\varepsilon = 10^{-6}$; (3) $N = 10^6$ and $\varepsilon = 10^{-3}$. Our code lengths are calculated from Theorem 1(1) (instead of slightly looser formula (1) in Theorem 1(2)) by using β_{formula} and the numerically searched optimal parameter β_{optimal}. The table also gives the percentages of our code lengths relative to lengths $100c^2\lceil\log(N/\varepsilon)\rceil$ of Tardos codes [9]. Moreover, Case 4 in this table gives the percentages in the limit case $N/\varepsilon \to \infty$ (i.e. $N \to \infty$ or $\varepsilon \to 0$); by Theorem 1(2), the percentage $m/\bigl(c^2\lceil\log(N/\varepsilon)\rceil\bigr)$ converges to $-\bigl(c^2 \log T_c\bigr)^{-1}$ when $N/\varepsilon \to \infty$. Table 4 is a similar table under the Marking Assumption; where Δ is equal to the approximation error δ' of bitwise scores.

These two tables show that our c-secure codes have lengths significantly shorter than Tardos codes and its preceding improvements [2,3,5,6,7,8], at least for the case of smaller c. For example, under the classical Marking Assumption, the code lengths in [7] for Case 1 are 6278, 19750, 41594 and 71552, respectively, when $c = 2$, 4, 6 and 8. On the other hand, in [8], Škorić et al. proved that the code lengths of Tardos codes under the Marking Assumption, with the symmetric scoring rule same as our code, can be reduced to $\pi^2 \approx 9.87\%$ of the

Table 4. Length comparison under Marking Assumption

Here $\Delta = \delta'$. Lengths in parentheses are computed by using β_{formula}.

c		Case 1	Case 2	Case 3	Case 4	β_{optimal}
2	Ours	**373**	**410**	**253**		0.17549
		(374)	(411)	(253)		
	Tardos	12400	14000	8400		
	%	**3.01**	**2.93**	**3.01**	**2.74**	
3	Ours	**1309**	**1423**	**877**		0.061345
		(1390)	(1511)	(931)		
	Tardos	28800	31500	18900		
	%	**4.55**	**4.52**	**4.64**	**4.23**	
4	Ours	**2190**	**2360**	**1454**		0.037405
		(2190)	(2360)	(1454)		
	Tardos	51200	56000	33600		
	%	**4.28**	**4.21**	**4.33**	**3.95**	
6	Ours	**5546**	**5909**	**3640**		0.016111
		(5547)	(5909)	(3641)		
	Tardos	115200	126000	75600		
	%	**4.81**	**4.69**	**4.81**	**4.39**	
8	Ours	**10469**	**11062**	**6815**		0.0089586
		(10469)	(11062)	(6816)		
	Tardos	211200	224000	134400		
	%	**4.96**	**4.94**	**5.07**	**4.62**	

original code lengths; and to $\pi^2/2 \approx 4.93\%$ under certain statistical assumption on scores of innocent users (see the reference for detail; see also [5]). Table 4 shows that our code lengths are shorter than the latter lengths in almost all cases considered here, without any statistical assumption.

5 Conclusion

In this paper, we give a c-secure fingerprinting code with very short code length. This is done by mixing two preceding improvements [7,8] of Tardos code, and by modifying its tracing algorithm so that it simply outputs one user with highest score and thus does not use a threshold any more. In case of smaller c, our code has indeed shorter length than Tardos code and its preceding improvements.

References

1. Boneh, D., Shaw, J.: Collusion-secure Fingerprinting for Digital Data. IEEE Trans. Inform. Theory 44, 1897–1905 (1998)
2. Hagiwara, M., Hanaoka, G., Imai, H.: A Short Random Fingerprinting Code Against a Small Number of Pirates. In: Fossorier, M.P.C., Imai, H., Lin, S., Poli, A. (eds.) AAECC 2006. LNCS, vol. 3857, pp. 193–202. Springer, Heidelberg (2006)
3. Isogai, T., Muratani, H.: Reevaluation of Tardos's Code. In: IEICE Technical Report, ISEC2006-96, pp. 7–12 (2006)
4. Carter, M., van Brunt, B.: The Lebesgue-Stieltjes Integral: A Practical Introduction. Springer, Heidelberg (2000)
5. Katzenbeisser, S., Škorić, B., Celik, M.U., Sadeghi, A.-R.: Combining Tardos Fingerprinting Codes and Fingercasting. In: IH 2007. LNCS, vol. 4567, Springer, Heidelberg (2007)
6. Nuida, K., Hagiwara, M., Watanabe, H., Imai, H.: Optimal Probabilistic Fingerprinting Codes Using Optimal Finite Random Variables Related to Numerical Quadrature, `http://www.arxiv.org/abs/cs/0610036`
7. Nuida, K., Hagiwara, M., Watanabe, H., Imai, H.: Optimization of Tardos's Fingerprinting Codes in a Viewpoint of Memory Amount. In: IH 2007. LNCS, vol. 4567, Springer, Heidelberg (2007)
8. Škorić, B., Katzenbeisser, S., Celik, M.U.: Symmetric Tardos Fingerprinting Codes for Arbitrary Alphabet Sizes, `http://eprint.iacr.org/2007/041`
9. Tardos, G.: Optimal Probabilistic Fingerprint Codes. J. ACM. In: 2003 ACM Symposium on Theory of Computing, pp. 116–125 (to appear)

Appendix: Proof of Theorem 1

This appendix is devoted to give an outline of the proof of Theorem 1; due to limited pages, details of the proof are omitted here and will appear in the full version of this paper.

We prepare the following lemmas, whose proofs follow the arguments in [2].

Lemma 1 (cf. [2], Lemma 1). *If $z \in \mathbb{R}$ and $\alpha > 0$, then for any fixed P and y, the probability that the score S of at least one innocent user satisfies $S \geq z$ is less than or equal to $\varphi(z) = \min\{NB_1(\alpha)^m e^{-\alpha z}, 1\}$.*

Proof (Sketch). Since there are at most N innocent users, this probability is less than or equal to $N\mathrm{Ex}\left[e^{\alpha S}\right]e^{-\alpha z}$ by the Markov's inequality. For bitwise scores S_j, by definition of p_0, an elementary analysis shows that

$$\begin{aligned}\mathrm{Ex}\left[e^{\alpha S_j}\right] &= \widetilde{p}^{(j)}e^{\alpha\sigma(\widetilde{p}^{(j)})} + (1-\widetilde{p}^{(j)})e^{-\alpha\sigma(1-\widetilde{p}^{(j)})} \\ &\leq p_0 e^{\alpha\sigma(p_0)} + (1-p_0)e^{-\alpha\sigma(1-p_0)} = B_1(\alpha)\ ,\end{aligned}$$

where $\widetilde{p}^{(j)} = p^{(j)}$ if $y_j = 1$ and $1 - p^{(j)}$ if $y_j \in \{0, ?\}$. Thus we have $\mathrm{Ex}\left[e^{\alpha S}\right] \leq B_1(\alpha)^m$, so the claim follows.

Lemma 2 (cf. [2], Lemma 2). *If $z \in \mathbb{R}$, $\beta > 0$ and there are ℓ pirates, then for any fixed pirates' strategy satisfying Marking Assumption, the probability that no pirate's score exceeds z is less than or equal to $F_\ell(z) = \min\{B_{2,\ell}(\beta)^m e^{\beta\ell z}, 1\}$.*

Proof (Sketch). This probability does not exceed the probability that the sum S_{psum} of ℓ pirates' scores is less than or equal to ℓz. By the Markov's inequality, the latter probability is less than or equal to $\mathrm{Ex}\left[e^{-\beta S_{\mathrm{psum}}}\right]e^{\beta\ell z}$. Now by a similar argument to [2,9], we have $\mathrm{Ex}\left[e^{-\beta S_{\mathrm{psum}}}\right] \leq \left(\sum_{x=0}^{\ell}\binom{\ell}{x}M_x\right)^m$, where $M_0 = N_{0,0}$, $M_\ell = N_{1,\ell}$, $M_x = \max\{N_{0,x}, N_{1,x}\}$ for $1 \leq x \leq \ell - 1$, with

$$N_{0,x} = \mathrm{Ex}\left[e^{\beta\mathcal{L}_{x,p}}p^x(1-p)^{\ell-x}\right]\ ,\ N_{1,x} = \mathrm{Ex}\left[e^{-\beta\mathcal{L}_{x,p}}p^x(1-p)^{\ell-x}\right]$$

(the last two expectation values are taken over the values p of $\mathcal{P}$) and $\mathcal{L}_{x,p} = x\sigma(p) - (\ell - x)\sigma(1-p)$. Since $|\mathcal{L}_{x,p}| \leq \ell\eta$, an elementary analysis shows that $e^{\pm\beta\mathcal{L}_{x,p}} \leq 1 \pm \beta\mathcal{L}_{x,p} + r(\beta\ell\eta)\beta^2{\mathcal{L}_{x,p}}^2$, respectively, where $r(t) = (e^t - 1 - t)/t^2$. Thus we have

$$\begin{aligned}M_x \leq\ & \mathrm{Ex}\left[p^x(1-p)^{\ell-x}\right] - \beta\mathrm{Ex}\left[p^x(1-p)^{\ell-x}\mathcal{L}_{x,p}\right] \\ & + r(\beta\ell\eta)\beta^2\mathrm{Ex}\left[p^x(1-p)^{\ell-x}{\mathcal{L}_{x,p}}^2\right] + 2\beta R_{\ell,x}\end{aligned}$$

for $1 \leq x \leq \ell - 1$; so by the fact that $\sum_{x=0}^{\ell}\binom{\ell}{x}p^x(1-p)^{\ell-x}{\mathcal{L}_{x,p}}^k = 1, 0, \ell$ for $k = 0, 1, 2$, respectively (cf. [2], Lemma 3), we have

$$\begin{aligned}\sum_{x=0}^{\ell}\binom{\ell}{x}M_x &\leq 1 + 2\beta\mathrm{Ex}\left[p^0(1-p)^{\ell-0}\mathcal{L}_{0,p}\right] + r(\beta\ell\eta)\beta^2\ell + 2\beta\sum_{x=1}^{\ell-1}\binom{\ell}{x}R_{\ell,x} \\ &= 1 + r(\beta\ell\eta)\beta^2\ell - 2\beta\mathcal{R}_\ell \leq B_{2,\ell}(\beta)\ .\end{aligned}$$

Hence we have $\mathrm{Ex}\left[e^{-\beta S_{\mathrm{psum}}}\right] \leq B_{2,\ell}(\beta)^m$, so the claim follows.

Now we come back to the proof of Theorem 1. Let y' be obtained by modifying y so that $y'_j = w_{1,j}$ whenever j-th bits $w_{1,j}, \ldots, w_{\ell,j}$ of the ℓ pirates coincide; i.e. y' satisfies the Marking Assumption. Then y and y' differ at up to $m\delta$ bits. Let S_{imax} denote the highest score of innocent users, and S'_{imax} be the highest score of innocent users which is calculated by using *precise* bitwise scores and the modified pirated codeword y' instead of y. Write the corresponding scores for

pirates by S_{pmax} and S'_{pmax}. Then we have $|S_{\mathrm{imax}} - S'_{\mathrm{imax}}| \le m\delta \cdot 2\eta + m\delta' = m\Delta$ and $|S_{\mathrm{pmax}} - S'_{\mathrm{pmax}}| \le m\Delta$ by definition of Δ; so the tracing error probability does not exceed $\Pr(S_{\mathrm{imax}} \ge S_{\mathrm{pmax}}) \le \Pr(S'_{\mathrm{imax}} + 2m\Delta \ge S'_{\mathrm{pmax}})$.

The following result is the key ingredient of our proof.

Lemma 3. *Put $G(z) = \Pr(S'_{\mathrm{pmax}} \le z)$ and $\widetilde{\varphi}(z) = \varphi(z - 2m\Delta)$. Then we have $\Pr(S'_{\mathrm{imax}} + 2m\Delta \ge S'_{\mathrm{pmax}}) \le \int_{\mathbb{R}} \widetilde{\varphi}\, dG$, where the last integral is the Lebesgue-Stieltjes integral with respect to the function G (cf. [4]).*

Proof (Sketch). Now we only give an intuitive argument, since the formal proof is too long to be included here (see the forthcoming full version of this paper for details). We evaluate the probability that $S'_{\mathrm{imax}} + 2m\Delta < S'_{\mathrm{pmax}}$; the probability that this event occurs and S'_{pmax} lies in a sufficiently minute interval $(z, z + dz]$ is $\ge \big(1 - \widetilde{\varphi}(z)\big)\big(G(z + dz) - G(z)\big)$ by Lemma 1. By taking the sum over these disjoint intervals covering the whole of $\mathbb{R}$, we have

$$\begin{aligned}\Pr(S'_{\mathrm{imax}} + 2m\Delta < S'_{\mathrm{pmax}}) &\ge \sum \big(1 - \widetilde{\varphi}(z)\big)\big(G(z + dz) - G(z)\big) \\ &= \sum \big(G(z + dz) - G(z)\big) - \sum \widetilde{\varphi}(z) \frac{G(z + dz) - G(z)}{dz} dz \\ &\overset{dz \to 0}{\to} 1 - \int_{-\infty}^{\infty} \widetilde{\varphi}(z) G'(z)\, dz = 1 - \int_{\mathbb{R}} \widetilde{\varphi} dG\end{aligned}$$

(Note that $\lim_{z \to \infty} G(z) = 1$, $\lim_{z \to -\infty} G(z) = 0$; while the function $G(z)$ is piecewise-linear, since now the number of the user's possible scores is finite). This infers the claim.

Moreover, since $\widetilde{\varphi} \ge 0$ is weakly decreasing and $G(z) \le F(z)$, we can derive the following fact from general properties of Lebesgue-Stieltjes integral.

Lemma 4. *We have $\int_{\mathbb{R}} \widetilde{\varphi}\, dG \le \int_{\mathbb{R}} \widetilde{\varphi}\, dF$ (see Lemma 2 for definition of F).*

Hence the tracing error probability is bounded by $\int_{\mathbb{R}} \widetilde{\varphi}\, dF$. Moreover, by putting $\alpha = \beta\ell$, a direct computation shows that $\int_{\mathbb{R}} \widetilde{\varphi}\, dF = \Phi(NT_\ell{}^m)$, where $T_\ell = B_1(\beta\ell) B_{2,\ell}(\beta) e^{2\beta\ell\Delta}$. Now Theorem 1(1) follows from the fact that $T_\ell \le T_c$ for any $1 \le \ell \le c$ and $\Phi(t)$ is increasing for $0 < t < 1$.

To prove Theorem 1(2), we consider the function $\Phi_\varepsilon(t) = \Phi(t) - \varepsilon$, which is increasing and concave up for $0 < t < 1$. Since $\lim_{t \to +0} \Phi_\varepsilon(t) = -\epsilon < 0$ and $\lim_{t \to 1-0} \Phi_\varepsilon(t) = 1 - \varepsilon > 0$, we have $\Phi_\varepsilon(t_0) = 0$ for a unique $0 < t_0 < 1$. Now if $a > 1$ and $\varepsilon \le a e^{1-a}$, then we have $\Phi_\varepsilon(\varepsilon/a) = (\varepsilon/a)\,(1 - \log(\varepsilon/a)) - \varepsilon \ge 0$ (note that $\log(\varepsilon/a) \le 1 - a$), so $t_0 \le \varepsilon/a < 1$. Then put

$$t_1 = \frac{\varepsilon}{a} - \frac{\Phi_\varepsilon(\varepsilon/a)}{\Phi'_\varepsilon(\varepsilon/a)} = \frac{a - 1}{a} \frac{\varepsilon}{\log(a/\varepsilon)} \ ,$$

which is the x-intercept of the tangent line of the curve $y = \Phi_\varepsilon(x)$ at $x = \varepsilon/a$. Since $\Phi_\varepsilon(t)$ is increasing and concave up, we have $t_1 \le t_0$ and so $\Phi_\varepsilon(t_1) \le 0$ (note that $t_1 > 0$). Thus we have $\Phi(NT_c{}^m) \le \Phi(t_1) \le \varepsilon$ whenever $NT_c{}^m \le t_1$, i.e. $m \ge -\log(N/t_1)/\log T_c$. Hence Theorem 1(2) is proved.

Space-Time Codes from Crossed Product Algebras of Degree 4*

Grégory Berhuy[1] and Frédérique Oggier[2]

[1] School of Mathematics
University of Southampton, UK
G.W.Berhuy@soton.ac.uk
[2] Department of Electrical Engineering
California Institute of Technology, USA
frederique@systems.caltech.edu

Abstract. We study crossed product algebras of degree 4, and present a new space-time code construction based on a particular crossed product division algebra which exhibits very good performance.

1 Introduction

Wireless systems are nowadays part of every day life. However, to answer the need of higher and higher data rate, researchers have started to investigate wireless systems where both the transmitter and receiver end are equipped with multiple antennas. This new kind of channel required new coding techniques, namely *space-time coding* [10]. Unlike classical coding, space-time coding involves the design of families of matrices, with the property, called *full diversity*, that the difference of any two distinct matrices is full rank.

Following the seminal work of Sethuraman et al. [7,8], codes based on division algebras have been investigated. This algebraic approach has generated a lot of interest, since division algebras naturally provide linear codes with full diversity. Quaternion algebras [1] and their maximal orders [3], cyclic algebras [8,4], Clifford algebras [9] and crossed product algebras [6] have been studied.

In this paper, we study crossed product algebras of degree 4, and, unlike in [6], we focus on the case where the Galois group is not cyclic. For this scenario, we derive conditions for crossed product algebras to be division algebras, which yields the full diversity property, and optimize the code design.

2 Crossed Product Algebras of Degree 4

Let L/K be a Galois extension. A central simple K-algebra is called a crossed product algebra over L/K if it contains L as a maximal commutative subfield. A crossed product algebra can be described nicely in terms of generators and

* This work was partly supported by the Nuffield Newly Appointed Lecturers Scheme 2006 NAL/32706, F. Oggier is now visiting RCIS, AIST, Tokyo, Japan.

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 90–99, 2007.
© Springer-Verlag Berlin Heidelberg 2007

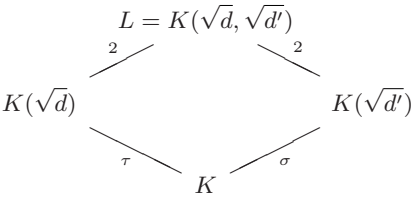


Fig. 1. A biquadratic extension of K

relations, and when L/K has cyclic Galois group, we recover the concept of cyclic algebra. Since in degree 2 and 3 Galois extensions have necessarily cyclic groups, the first interesting example of crossed product algebra arises in degree 4. This is the case we focus on in this work. For definitions and basic facts on crossed product algebras, the reader may refer to [2].

2.1 Definition and Examples

Consider a Galois extension L/K of degree 4. Its Galois group is either cyclic of order 4 or a product of two cyclic groups of order 2. We focus on the latter, and consider the case where L/K is a biquadratic extension (see Fig. 1), namely

$$L = K(\sqrt{d}, \sqrt{d'}).$$

We set $\mathcal{G} = \mathrm{Gal}(L/K) = \{1, \sigma, \tau, \sigma\tau\}$, where σ, τ are defined by

$$\sigma(\sqrt{d}) = \sqrt{d}, \sigma(\sqrt{d'}) = -\sqrt{d'}$$

$$\tau(\sqrt{d}) = -\sqrt{d}, \tau(\sqrt{d'}) = \sqrt{d'}.$$

In this case, using a suitable change of generators, one can show that a crossed product algebra $\mathcal{A}$ over L/K may be described as follows:

$$\mathcal{A} = L \oplus eL \oplus fL \oplus efL$$

with $e^2 = a,\ f^2 = b,\ fe = efu,\ \ \lambda e = e\sigma(\lambda),\ \ \lambda f = f\tau(\lambda)$ for all $\lambda \in L$, for some elements $a, b, u \in L^\times$ satisfying

$$\sigma(a) = a, \tau(b) = b, u\sigma(u) = \frac{a}{\tau(a)}, u\tau(u) = \frac{\sigma(b)}{b}. \tag{1}$$

Definition 1. *A crossed product algebra $\mathcal{A}$ over a biquadratic extension L/K will be called a* biquadratic crossed product algebra. *We write $\mathcal{A} = (a, b, u, L/K)$.*

Remark 1. Note from (1) that we have that $a \in K(\sqrt{d})$ and $b \in K(\sqrt{d'})$.

Example 1. Take $K = \mathbb{Q}(i)$, $d = 3$ and $d' = 5$, so that $L = \mathbb{Q}(i)(\sqrt{3}, \sqrt{5})$. The following choice of a, b, u is well defined:

$$a = \sqrt{3},\ b = \sqrt{5},\ u = i.$$

We need to verify that the conditions (1) are satisfied. Recall that here

$$\sigma(\sqrt{5}) = -\sqrt{5},\ \tau(\sqrt{3}) = -\sqrt{3}.$$

Clearly $\sigma(a) = a$ and $\tau(b) = b$. Finally

$$u\sigma(u) = -1 = \frac{\sqrt{3}}{-\sqrt{3}} \text{ and } u\tau(u) = -1 = \frac{-\sqrt{5}}{\sqrt{5}}.$$

Example 2. Take again $K = \mathbb{Q}(i)$, $d' = 5$, but now $d = 2$. Let ζ_8 be a primitive 8th root of unity. Note that $L = \mathbb{Q}(i)(\zeta_8, \sqrt{5})$ since $\zeta_8 = \frac{1}{\sqrt{2}}(1+i)$. We have

$$\sigma(\sqrt{5}) = -\sqrt{5},\ \tau(\sqrt{2}) = -\sqrt{2},\ \tau(\zeta_8) = -\zeta_8.$$

The following choice of a, b, u is also suitable:

$$a = \zeta_8,\ b = \sqrt{5},\ u = i.$$

Clearly $\sigma(a) = a$ and $\tau(b) = b$. Finally

$$u\sigma(u) = -1 = \frac{\zeta_8}{-\zeta_8} \text{ and } u\tau(u) = -1 = \frac{-\sqrt{5}}{\sqrt{5}}.$$

Remark 2. It is known that every central simple algebra over a number field is isomorphic to a cyclic algebra. However, for coding purposes, the algebra representation does matter, as it will be illustrated in the following (Remark 3).

2.2 Matrix Formulation and Encoding

In order to design codewords, we now explain how to identify $\mathcal{A}$ to a subalgebra of $\mathcal{M}_4(L)$, or in other words, how to get a correspondence between a matrix $\mathbf{X} \in \mathcal{M}_4(L)$, which will be a codeword, and an element $x \in \mathcal{A}$. This is done by associating to $\mathbf{X}$ its left multiplication matrix.

Proposition 1. *Let $x = x_1 + ex_\sigma + fx_\tau + efx_{\sigma\tau} \in \mathcal{A}$. Its left multiplication matrix $\mathbf{X}$ is given by*

$$\begin{pmatrix} x_1 & a\sigma(x_\sigma) & b\tau(x_\tau) & ab\tau(u)\sigma\tau(x_{\sigma\tau}) \\ x_\sigma & \sigma(x_1) & b\tau(x_{\sigma\tau}) & b\tau(u)\sigma\tau(x_\tau) \\ x_\tau & \tau(a)u\sigma(x_{\sigma\tau}) & \tau(x_1) & \tau(a)\sigma\tau(x_\sigma) \\ x_{\sigma\tau} & u\sigma(x_\tau) & \tau(x_\sigma) & \sigma\tau(x_1) \end{pmatrix}. \tag{2}$$

Proof. It is enough to do the computation on the basis elements. We have

$$ae = x_1e + ex_\sigma e + fx_\tau e + efx_{\sigma\tau}e = e\sigma(x_1) + a\sigma(x_\sigma) + fe\sigma(x_\tau) + efe\sigma(x_{\sigma\tau}).$$

Now we have $fe = efu$, and $efe = eefu = afu = f\tau(a)u$. Hence

$$ae = a\sigma(x_\sigma) + e\sigma(x_1) + f\tau(a)u\sigma(x_{\sigma\tau}) + efu\sigma(x_\tau).$$

We have also

$$af = x_1 f + e x_\sigma f + f x_\tau f + e f x_{\sigma\tau} f = f\tau(x_1) + ef\tau(x_\sigma) + b\tau(x_\tau) + eb\tau(x_{\sigma\tau}).$$

Hence,

$$af = b\tau(x_\tau) + eb\tau(x_{\sigma\tau}) + f\tau(x_1) + ef\tau(x_\sigma).$$

Finally,

$$\begin{aligned} aef &= x_1 ef + e x_\sigma ef + f x_\tau ef + e f x_{\sigma\tau} ef \\ &= ef\sigma\tau(x_1) + af\sigma\tau(x_\sigma) + fef\sigma\tau(x_\tau) + efef\sigma\tau(x_{\sigma\tau}). \end{aligned}$$

We have

$$fef = efuf = eb\tau(u),$$

and $efef = e(eb\tau(u)) = ab\tau(u)$. Thus,

$$aef = ef\sigma\tau(x_1) + f\tau(a)\sigma\tau(x_\sigma) + eb\tau(u)\sigma\tau(x_\tau) + ab\tau(u)\sigma\tau(x_{\sigma\tau}).$$

Therefore,

$$aef = ab\tau(u)\sigma\tau(x_{\sigma\tau}) + eb\tau(u)\sigma\tau(x_\tau) + f\tau(a)\sigma\tau(x_\sigma) + ef\sigma\tau(x_1).$$

For a matrix $\mathbf{X}$ of the form (2) to be a codeword, it further requires an encoding, that is a way to map the information symbols to be transmitted into the matrix $\mathbf{X}$. This can be easily done as follows. Let $\{\omega_1, \omega_2, \omega_3, \omega_4\}$ be a $\mathbb{Q}(i)$-basis of L. Let G be the matrix of the embeddings of the basis:

$$G = \begin{pmatrix} \omega_1 & \omega_2 & \omega_3 & \omega_4 \\ \sigma(\omega_1) & \sigma(\omega_2) & \sigma(\omega_3) & \sigma(\omega_4) \\ \tau(\omega_1) & \tau(\omega_2) & \tau(\omega_3) & \tau(\omega_4) \\ \sigma\tau(\omega_1) & \sigma\tau(\omega_2) & \sigma\tau(\omega_3) & \sigma\tau(\omega_4) \end{pmatrix}. \tag{3}$$

Let $\mathbf{x} = (x_1, x_2, x_3, x_4)$ be a vector containing 4 information symbols to be transmitted. Let $x = x_1\omega_1 + x_2\omega_2 + x_3\omega_3 + x_4\omega_4$ be an element of L, which can be seen as a linear combination of the 4 information symbols. We have

$$G\mathbf{x} = (x, \sigma(x), \tau(x), \sigma\tau(x))^T.$$

We can thus encode 16 information symbols into $\mathbf{X}$ as follows. Let

$$G\mathbf{x}_1 = (x_1, \sigma(x_1), \tau(x_1), \sigma\tau(x_1))^T.$$

$$G\mathbf{x}_\sigma = (x_\sigma, \sigma(x_\sigma), \tau(x_\sigma), \sigma\tau(x_\sigma))^T.$$

$$G\mathbf{x}_\tau = (x_\tau, \sigma(x_\tau), \tau(x_\tau), \sigma\tau(x_\tau))^T.$$

$$G\mathbf{x}_{\sigma\tau} = (x_{\sigma\tau}, \sigma(x_{\sigma\tau}), \tau(x_{\sigma\tau}), \sigma\tau(x_{\sigma\tau}))^T.$$

Let Γ_i, $i = 1, 2, 3, 4$, be given by $\Gamma_1 = \mathbf{I}_4$, the identity matrix, and

$$\Gamma_2 = \begin{pmatrix} 0 & a & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \tau(a) \\ 0 & 0 & 1 & 0 \end{pmatrix}, \Gamma_3 = \begin{pmatrix} 0 & 0 & b & 0 \\ 0 & 0 & 0 & b\sigma(u) \\ 1 & 0 & 0 & 0 \\ 0 & \sigma\tau(u) & 0 & 0 \end{pmatrix}, \Gamma_4 = \begin{pmatrix} 0 & 0 & 0 & ab\sigma(u) \\ 0 & 0 & b & 0 \\ 0 & \tau(a)\tau(u) & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

The codeword $\mathbf{X}$ is encoded as follows:

$$\mathbf{X} = \Gamma_1 diag(G\mathbf{x}_1) + \Gamma_2 diag(G\mathbf{x}_\sigma) + \Gamma_3 diag(G\mathbf{x}_\tau) + \Gamma_4 diag(G\mathbf{x}_{\sigma\tau}).$$

3 A Criterion for Full Diversity

For square codewords, the full diversity property [10] is given by

$$\det(\mathbf{X}' - \mathbf{X}'') \neq 0,\ \mathbf{X}' \neq \mathbf{X}'' \in \mathcal{A},$$

where $\mathcal{A}$ is identified with a subalgebra of a matrix algebra. Therefore, in order to satisfy this property, it is enough to require $\mathcal{A}$ to be a division algebra.

Theorem 1. *Let K be a number field, and let $\mathcal{A} = (a, b, u, L/K)$. Then the following conditions are equivalent:*

1. *$\mathcal{A}$ is a division algebra,*
2. *the quaternion algebra $(d, N_{K(\sqrt{d'})/K}(b))$ is not split,*
3. *the quaternion algebra $(d', N_{K(\sqrt{d})/K}(a))$ is not split.*

Proof. Since K is a number field, then the index of $\mathcal{A}$ is equal to its exponent. Thus $\mathcal{A}$ is a division algebra if and only if its exponent is not 1 or 2, that is $2[\mathcal{A}] \neq 0$ in the Brauer group $\mathrm{Br}(K)$. To conclude, it is enough to use the following equalities, that we will not prove here by lack of space:

$$2[\mathcal{A}] = (d, N_{K(\sqrt{d'})/K}(b)) = (d', N_{K(\sqrt{d})/K}(a)) \text{ in } \mathrm{Br}(K).$$

Lemma 1. *Let $u \in L$. The following conditions are equivalent:*

(1) $N_{L/K}(u) = 1$.
(2) There exists $a \in L^\times$ such that

$$\sigma(a) = a,\ u\sigma(u) = \frac{a}{\tau(a)}.$$

(3) There exists $b \in L^\times$ such that

$$\tau(b) = b,\ u\tau(u) = \frac{\sigma(b)}{b}.$$

Moreover, if u satisfies one of the above conditions, then whether the quaternion algebra $(d', N_{K(\sqrt{d})/K}(a))$ is split only depends on u, and not on the choice of a.

Proof. 1. If $N_{L/K}(u) = 1$, then

$$u\sigma(u)\tau(u)\sigma(\tau(u)) = 1,$$

so that

$$N_{K(\sqrt{d})/K}(u\sigma(u)) = 1,\ N_{K(\sqrt{d'})/K}(u\tau(u)) = 1,$$

and thus we get both (2) and (3) by Hilbert's 90. Now, if (2) holds, then

$$N_{L/K}(u) = u\sigma(u)\tau(u)\sigma(\tau(u)) = \frac{a}{\tau(a)}\tau\left(\frac{a}{\tau(a)}\right) = 1,$$

and similarly (3) implies (1).

2. Let u be given, and consider a, a' such that

$$u\sigma(u) = \frac{a}{\tau(a)} = \frac{a'}{\tau(a')},$$

so that

$$a'\tau(a) = a\tau(a') = \tau(a'\tau(a)).$$

Since we further have that

$$\sigma(a'\tau(a)) = \sigma(a')\tau(\sigma(a)) = a'\tau(a),$$

we conclude that $a'\tau(a) = \lambda \in K$, and thus

$$N_{K(\sqrt{d})/K}(a'\tau(a)) = \lambda^2.$$

In other words,

$$N_{K(\sqrt{d})/K}(a') = \left(\frac{\lambda}{N_{K(\sqrt{d})/K}(a)}\right)^2 N_{K(\sqrt{d})/K}(a)$$

and $(d', N_{K(\sqrt{d})/K}(a))$ is split if and only if $(d', N_{K(\sqrt{d})/K}(a'))$ is, which concludes the proof.

Lemma 2. *Let $u \in L$ such that $N_{L/K}(u) = 1$. If $u\sigma(u) = -1$, then we have*

$$u\sigma(u) = \frac{a}{\tau(a)}, \text{ where } a = \sqrt{d}, \text{ and } (d', N_{K(\sqrt{d})/K}(a)) = (-d, d'). \tag{4}$$

If $u\sigma(u) \neq -1$, then we have $u\sigma(u) = \dfrac{a}{\tau(a)}$, where

$$a = 1 + u\sigma(u), \text{ and } (d', N_{K(\sqrt{d})/K}(a)) = (2 + Tr_{K(\sqrt{d})/K}(u\sigma(u)), d'). \tag{5}$$

Proof. (4) is obvious. Now, assume that $u\sigma(u) \neq -1$ and set $a = 1 + u\sigma(u)$. We have that $u\sigma(u) + N_{L/K}(u) = u\sigma(u)\tau(1 + u\sigma(u))$, so that

$$u\sigma(u) = \frac{u\sigma(u) + N_{L/K}(u)}{\tau(1 + u\sigma(u))} = \frac{u\sigma(u) + 1}{\tau(1 + u\sigma(u))} = \frac{a}{\tau(a)}.$$

To conclude, for $a = u\sigma(u) + 1$, we have

$$\begin{aligned} N_{K(\sqrt{d})/K}(a) &= (u\sigma(u) + 1)\tau(u\sigma(u) + 1) \\ &= 1 + Tr_{K(\sqrt{d})/K}(u\sigma(u)) + N_{L/K}(u) \\ &= 2 + Tr_{K(\sqrt{d})/K}(u\sigma(u)). \end{aligned}$$

Example 3. Consider the algebra defined in Example 1, namely $K = \mathbb{Q}(i)$, $L = \mathbb{Q}(i)(\sqrt{3}, \sqrt{5})$ with

$$a = \sqrt{3},\ b = \sqrt{5},\ u = i.$$

Since $u\sigma(u) = -1$, by Lemma 2, we have to check whether $(-3, 5)$ is split. This is equivalent to check whether -3 is a norm in $\mathbb{Q}(i)(\sqrt{5})/\mathbb{Q}(i)$, namely whether $a^2 - 5b^2 = -3$ has a solution for $a, b \in \mathbb{Q}(i)$. If such a solution exists, then it is easy to see that the denominators of a and b are not divisible by $(2 + i)$. Therefore, reducing modulo $(2+i)$, we get that -3 is a square in $\mathbb{Z}[i]/(2+i)$. Since $5\mathbb{Z}[i] = (2+i)(2-i)$, the inertial degree [5, p.84] of $2+i$ is 1, and $\mathbb{Z}[i]/(2+i) \cong \mathbb{F}_5$. Since -3 is not a square modulo 5, we conclude that $(-3, 5)$ is not split.

Example 4. We now continue Example 2, where $K = \mathbb{Q}(i)$ and $L = \mathbb{Q}(i)(\zeta_8, \sqrt{5})$, with ζ_8 a primitive 8th root of unity. Furthermore, we have

$$a = \zeta_8,\ b = \sqrt{5},\ u = i.$$

Again $u\sigma(u) = -1$, and we have to check, by Lemma 2, whether $(-2, 5)$ is split. Since -2 is not a square modulo 5, we show as above that $(-2, 5)$ is not split.

4 Codes and Performance

From the above (see Examples 1, 3, 2 and 4), we now have two examples of division crossed product algebras:

1. $(a, b, u, L/K) = (\sqrt{3}, \sqrt{5}, i, \mathbb{Q}(i)(\sqrt{3}, \sqrt{5})/\mathbb{Q}(i))$,
2. $(a, b, u, L/K) = (\zeta_8, \sqrt{5}, i, \mathbb{Q}(i)(\sqrt{2}, \sqrt{5})/\mathbb{Q}(i))$.

We thus have two fully-diverse codes with a linear encoding. However, it is now known that this is not enough to get efficient codes, and a crucial other parameter is a *good shaping* (following the terminology of [4]), or in other words, the codes should be information lossless [6]. Both requirements can actually be shown to boil down to the same property: the matrices G and Γ_i, $i = 2, 3, 4$ used for the encoding (see Subsection 2.2) have to be unitary.

4.1 The Algebra on $\mathbb{Q}(i)(\sqrt{3}, \sqrt{5})/\mathbb{Q}(i)$

Recall that the encoding matrix Γ_3 is given by

$$\Gamma_3 = \begin{pmatrix} 0 & 0 & b & 0 \\ 0 & 0 & 0 & b\sigma(u) \\ 1 & 0 & 0 & 0 \\ 0 & \sigma\tau(u) & 0 & 0 \end{pmatrix}.$$

In order for Γ_3 to be unitary, we clearly need, since σ and τ commute with the complex conjugation, that $|u|^2 = 1,\ |b|^2 = 1$. Since $u = i$, we focus on $b = \sqrt{5}$. Of course b is not of modulus 1, but this can be remedied by normalizing it as follows:

$$b = \sqrt{\frac{1+2i}{1-2i}} = \frac{1+2i}{\sqrt{5}}.$$

Similarly, we need $|a|^2 = 1$ in order for Γ_2 to be unitary, where

$$\Gamma_2 = \begin{pmatrix} 0 & a & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \tau(a) \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Since such a normalization is not possible for 3, we focus on the other algebra.

4.2 The Algebra on $\mathbb{Q}(i)(\sqrt{2}, \sqrt{5})/\mathbb{Q}(i)$

As seen in the previous subsection, we need $|a|^2 = 1,\ |b|^2 = 1,\ |u|^2 = 1$. This is however fine here, since a and u are roots of unity, while b can be normalized. We thus finally take

$$a = \zeta_8,\ b = \sqrt{\frac{1+2i}{1-2i}},\ u = i.$$

Thus the encoding matrices $\Gamma_i,\ i = 2, 3, 4$ are unitary. We are thus left with making sure that G is unitary. Recall from (3) that G is given by:

$$G = \begin{pmatrix} \omega_1 & \omega_2 & \omega_3 & \omega_4 \\ \sigma(\omega_1) & \sigma(\omega_2) & \sigma(\omega_3) & \sigma(\omega_4) \\ \tau(\omega_1) & \tau(\omega_2) & \tau(\omega_3) & \tau(\omega_4) \\ \sigma\tau(\omega_1) & \sigma\tau(\omega_2) & \sigma\tau(\omega_3) & \sigma\tau(\omega_4) \end{pmatrix},$$

where $\{\omega_1, \omega_2, \omega_3, \omega_4\}$ is a basis of L. We can obtain a matrix G unitary by restricting to an ideal of L, as follows. Set

$$\theta = \frac{1+\sqrt{5}}{2},\ \alpha = 1 + i - i\theta.$$

Then the following basis

$$\omega_1 = \alpha,\ \omega_2 = \alpha\theta,\ \omega_3 = \alpha\zeta_8,\ \omega_4 = \alpha\theta\zeta_8$$

is such that $\dfrac{1}{\sqrt{10}}G$ is unitary. This can be easily checked since

$$G = G_2 \otimes G_1 \text{ with } G_1 = \begin{pmatrix} \alpha & \alpha\theta \\ \sigma(\alpha) & \sigma(\alpha)\sigma(\theta) \end{pmatrix},\ G_2 = \begin{pmatrix} 1 & \zeta_8 \\ 1 & \tau(\zeta_8) \end{pmatrix}$$

and G_1, G_2 satisfy

$$G_1 G_1^* = 5\mathbf{I}_2,\ G_2 G_2^* = 2\mathbf{I}_2.$$

Remark 3. Note that the crossed product algebra described in this subsection is isomorphic to the cyclic algebra $(i, \mathbb{Q}(i)(5^{1/4})/\mathbb{Q}(i), \sigma)$, where $\sigma(5^{1/4}) = i5^{1/4}$. However, the code construction is not available on the cyclic representation, since the orthonormal lattice does not exist.

4.3 Minimum Determinant and Simulations

Once a code satisfies the full diversity property and the shaping constraint, its performance is then governed by its *minimum determinant* [10], given by

$$\min_{\mathbf{X}\neq\mathbf{0}} |\det(\mathbf{X})|^2.$$

In the case of the code on $\mathbb{Q}(i)(\sqrt{2},\sqrt{5})/\mathbb{Q}(i)$, we have that [4]

$$\min_{\mathbf{X}\neq\mathbf{0}} |\det(\mathbf{X})|^2 = \frac{|N(\alpha)|^4}{\sqrt{10}^8}\frac{1}{|1-2i|^2} = \frac{1}{400}\frac{1}{5}$$

where the first equality comes from the following observations: the factor $\alpha\sigma(\alpha) = N(\alpha)$ appears squared in the determinant of $\mathbf{X}$, while the terms in $\sqrt{10}$ comes from the normalization of G, that is $\frac{1}{\sqrt{10}}G$. The term in $1-2i$ comes from the denominator of b. Note that 400 is actually the discriminant of $\mathbb{Q}(i)(\sqrt{2},\sqrt{5})/\mathbb{Q}(i)$.

The performance of this new code is shown in Fig. 2, compared to the best known code built on division algebras, namely on a cyclic division algebra [4], using a cyclic extension of discriminant 1125. The new code performs clearly better when using 4-QAM (that is $\pm 1 \pm i$ as information symbols). It looses a bit of its advantage when using 16-QAM. This can be easily explained. The discriminant of the new code is 400, and a further factor of 5 appears only when the term involving b is non-zero. So on average, the new code still performs better than the code based on cyclic algebra. However, when increasing the constellation size, the event of having the term in b non-zero occurs with smaller probability, and on average, the code still performs better, but with less advantage than in the 4-QAM case.

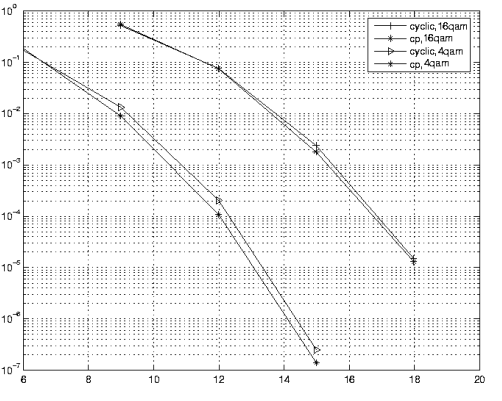


Fig. 2. New code from crossed product algebra, compared with the known code from cyclic algebra, using 4-QAM and 16-QAM

5 Conclusion

In this paper, we studied crossed product algebras of degree 4, in order to design new space-time code constructions. We provided conditions for crossed product algebras to be division algebras, and optimized the code design.

References

1. Belfiore, J.-C., Rekaya, G.: Quaternionic lattices for space-time coding. In: 2003 Information Theory Workshop, Paris (2003)
2. Draxl, P.K.: Skew fields. L.M.S.Lect. Note Serie, vol. 81. Cambridge Univ. Press, Cambridge (1982)
3. Hollanti, C., Lahtonen, J., Ranto, K., Vehkalahti, R.: Optimal Matrix Lattices for MIMO Codes from Division Algebras. In: 2006 IEEE Int. Symp. on Inform. Theory, Seattle (2006)
4. Oggier, F.E., Rekaya, G., Belfiore, J.-C., Viterbo, E.: Perfect Space-Time Block Codes. IEEE Trans. Inform. Theory 52(9), 3885–3902 (2006)
5. Samuel, P.: Théorie algébrique des nombres. Available in English. Hermann collection Méthodes, Paris (1967)
6. Vummintala, S., Sundar Rajan, B., Sethuraman, B.A.: Information-Lossless Space-Time Block Codes from Crossed-Product Algebras. IEEE Trans. Inform. Theory 52(9), 3913–3935 (2006)
7. Sethuraman, B.A., Sundar Rajan, B.: Full-Rank, Full-Rate STBCs from Division Algebras. In: 2002 Information Theory Workshop, Bangalore (2002)
8. Sethuraman, B.A., Sundar Rajan, B., Shashidhar, V.: Full-Diversity, High-Rate Space-Time Block Codes from Division Algebras. IEEE Trans. Inform. Theory 49(10), 2596–2616 (2003)
9. Susinder Rajan, G., Sundar Rajan, B.: STBCs from Representation of Extended Clifford Algebras. In: 2007 IEEE Int. Symp. on Inform. Theory, Nice (2007)
10. Tarokh, V., Seshadri, N., Calderbank, R.: Space-Time Codes for High Data Rate Wireless Communication: Performance Criterion and Code Construction. IEEE Trans. Inform. Theory 44, 744–765 (1998)

On Non-randomness of the Permutation After RC4 Key Scheduling

Goutam Paul[1], Subhamoy Maitra[2], and Rohit Srivastava[3]

[1] Department of Computer Science and Engineering, Jadavpur University, Kolkata 700 032, India
goutam_paul@cse.jdvu.ac.in
[2] Applied Statistics Unit, Indian Statistical Institute, 203, B T Road, Kolkata 700 108, India
subho@isical.ac.in
[3] Department of Computer Science and Engineering, Institute of Technology, Banaras Hindu University, Varanasi 221 005 (UP), India
rohit.engg@gmail.com

Abstract. Here we study a weakness of the RC4 Key Scheduling Algorithm (KSA) that has already been noted by Mantin and Mironov. Consider the RC4 permutation S of N (usually 256) bytes and denote it by S_N after the KSA. Under reasonable assumptions we present a simple proof that each permutation byte after the KSA is significantly biased (either positive or negative) towards many values in the range $0, \ldots, N-1$. These biases are independent of the secret key and thus present an evidence that the permutation after the KSA can be distinguished from random permutation without any assumption on the secret key. We also present a detailed empirical study over Mantin's work when the theoretical formulae vary significantly from experimental results due to repetition of short keys in RC4. Further, it is explained how these results can be used to identify new distinguishers for RC4 keystream.

Keywords: Bias, Cryptography, Cryptanalysis, Key Scheduling Algorithm, RC4, Stream Cipher.

1 Introduction

RC4, one of the most popular stream ciphers till date, was proposed by Rivest in 1987. The cipher gained its popularity from its extremely simple structure and substantially good strength in security, as even after lots of explored weaknesses in the literature (see [1,2,3,4,5,6,7,9,10,11,12,13,14] and the references in these papers), it could not be thoroughly cracked. Studying weaknesses of RC4 received serious attention in the literature and these studies are believed to be quite useful in further development of stream ciphers that exploit shuffle-exchange paradigm.

Before getting into our contribution, let us briefly present the Key Scheduling Algorithm (KSA) and the Pseudo Random Generation Algorithm (PRGA) of RC4. The data structure consists of (1) an array of size N (in practice 256

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 100–109, 2007.
© Springer-Verlag Berlin Heidelberg 2007

which is followed in this paper) which contains a permutation of $0, \ldots, N-1$, (2) two indices i, j and (3) the secret key array K. Given a secret key k of l bytes (typically 5 to 32), the array K of size N is such that $K[i] = k[i \bmod l]$ for any i, $0 \leq i \leq N-1$. All additions used in the description of the algorithm are modulo N additions.

Algorithm KSA	**Algorithm PRGA**
Initialization:	*Initialization*:
For $i = 0, \ldots, N-1$	$i = j = 0$;
$S[i] = i$;	*Output Keystream Generation Loop*:
$j = 0$;	$i = i + 1$;
Scrambling:	$j = j + S[i]$;
For $i = 0, \ldots, N-1$	Swap($S[i]$, $S[j]$);
$j = (j + S[i] + K[i])$;	$t = S[i] + S[j]$;
Swap($S[i], S[j]$);	Output $z = S[t]$;

RC4 KSA has been analysed deeply in [13,14,2,11]. All these works discuss the relationship of the permutation bytes after the KSA with the secret key. For a proper design, the permutation S after the KSA should not have any correlation with the secret keys. However, weaknesses of RC4 in this aspect have already been reported [13,14,2,11]. These weaknesses, in turn, leak information about RC4 secret key in the initial keystream output bytes [10].

Another approach of study is to look at the permutation after the KSA in a (secret) key independent manner and try to distinguish it from random permutations. In [9], the sign of the permutation after the KSA has been studied (see [9] for the definition of the sign of a permutation). There it has been shown that, after the KSA, the sign of the permutation can be guessed with probability 56%.

In [8, Chapter 6 and Appendix C] and later in [9], the problem of estimating $P(S_N[u] = v)$ has been discussed. A complete proof for these results has been presented in [8, Chapter 6 and Appendix C]. We present an independent proof technique in this paper which looks simpler. We argue in more detail in Section 2 how our technique is different from that in [8]. Due to the small keys (say 5 to 32 bytes) generally used in RC4, some of the assumptions differ from practice and hence the theoretical formulae do not match with the experimental results. We also detail this over the already identified anomalies in [8]. Further, we discuss applications to show how these results can be used to present new distinguishers for RC4. The distinguishers discussed in this paper are different from the earlier ones [1,3,5,7,12].

2 Bias in Each Permutation Byte

We denote the initial identity permutation by S_0 and the permutation at the end of the r-th round of the KSA by S_r, $1 \leq r \leq N$ (note that $r = i + 1$, for the deterministic index i, $0 \leq i \leq N-1$). Thus, the permutation after the

KSA will be denoted by S_N. By j_r, we denote the value of the index j after it is updated in round r. We consider the index j of each round to be distributed uniformly at random. Further, we replace the joint probabilities with the product of the probabilities of the individual events, assuming that the events under consideration are statistically independent.

Lemma 1. $P(S_2[0] = 1) = \frac{2(N-1)}{N^2}$.

Proof. In the first round, we have $i = 0$, and $j_1 = 0 + S[0] + K[0] = K[0]$. In the second round, $i = 1$ and $j_2 = j_1 + S_1[1] + K[1]$. We consider two mutually exclusive and exhaustive cases, namely, $K[0] = 1$ and $K[0] \neq 1$.

1. Take $K[0] = 1$. So, after the first swap, $S_1[0] = 1$ and $S_1[1] = 0$. Now, $j_2 = K[0] + 0 + K[1] = K[0] + K[1]$. Thus, after the second swap, $S_2[0]$ will remain 1, if $K[0] + K[1] \neq 0$. Hence the contribution of this case to the event $(S_2[0] = 1)$ is $P(K[0] = 1) \cdot P(K[0] + K[1] \neq 0) = \frac{1}{N} \cdot \frac{N-1}{N} = \frac{N-1}{N^2}$.
2. Take $K[0] \neq 1$. Then after the first swap, $S_1[1]$ remains 1. Now, $j_2 = K[0] + 1 + K[1] = K[0] + K[1] + 1$. Thus, after the second swap, $S_2[0]$ will get the value 1, if $K[0] + K[1] + 1 = 0$. Hence the contribution of this case to the event $(S_2[0] = 1)$ is $P(K[0] \neq 1) \cdot P(K[0] + K[1] + 1 = 0) = \frac{N-1}{N} \cdot \frac{1}{N} = \frac{N-1}{N^2}$.

Adding the two contributions, we get the total probability as $\frac{2(N-1)}{N^2}$. □

We here calculate $P(S_{v+1}[u] = v)$ for the special case $u = 0$, $v = 1$. Note that the form of $P(S_{v+1}[u] = v)$ for $v \geq u + 1$ in general (see Lemma 2 later) does not work for the case $u = 0, v = 1$ only. This will be made clear in Remark 1 after the proof of Lemma 2.

Proposition 1. $P(S_v[v] = v) = (\frac{N-1}{N})^v$, *for* $v \geq 0$.

Proof. In the rounds 1 through v, the deterministic index i touches the permutation indices $0, 1, \ldots, v-1$. Thus, after round v, $S_v[v]$ will remain the same as $S_0[v] = v$, if v has not been equal to any of the v many pseudo-random indices $j_1, j_2, \ldots, j_v$. The probability of this event is $(\frac{N-1}{N})^v$. So the result holds for $v \geq 1$. Furthermore, $P(S_0[0] = 0) = 1 = (\frac{N-1}{N})^0$. Hence, for any $v \geq 0$, we have $P(S_v[v] = v) = (\frac{N-1}{N})^v$. □

Proposition 2. *For* $v \geq u + 1$, $P(S_v[u] = v) = \frac{1}{N} \cdot (\frac{N-1}{N})^{v-u-1}$.

Proof. In round $u + 1$, the permutation index u is touched by the deterministic index i for the first time and the value at index u is swapped with the value at a random location based on j_{u+1}. Hence, $P(S_{u+1}[u] = v) = \frac{1}{N}$. The probability that the index u is not touched by any of the subsequent $v - u - 1$ many j values, namely, $j_{u+2}, \ldots, j_v$, is given by $(\frac{N-1}{N})^{v-u-1}$. So, after the end of round v, $P(S_v[u] = v) = \frac{1}{N} \cdot (\frac{N-1}{N})^{v-u-1}$. □

Lemma 2. *For* $v \geq u+1$ (except for the case "$u = 0$ and $v = 1$"), $P(S_{v+1}[u] = v) = \frac{1}{N} \cdot (\frac{N-1}{N})^{v-u} + \frac{1}{N} \cdot (\frac{N-1}{N})^v - \frac{1}{N^2} \cdot (\frac{N-1}{N})^{2v-u-1}$.

Proof. In round $v+1$, $i = v$ and $j_{v+1} = j_v + S_v[v] + K[v]$. The event $(S_{v+1}[u] = v)$ can occur in two ways.

1. $S_v[u]$ already had the value v and the index u is not involved in the swap in round $v+1$.
2. $S_v[u] \neq v$ and the value v comes into the index u from the index v (i.e., $S_v[v] = v$) by the swap in round $v+1$.

From Proposition 1, we have $P(S_v[v] = v) = (\frac{N-1}{N})^v$ and from Proposition 2, we have $P(S_v[u] = v) = \frac{1}{N} \cdot (\frac{N-1}{N})^{v-u-1}$. Hence, $P(S_{v+1}[u] = v)$
$= P(S_v[u] = v) \cdot P(j_v + S_v[v] + K[v] \neq u)$
$\quad + P(S_v[u] \neq v) \cdot P(S_v[v] = v) \cdot P(j_v + S_v[v] + K[v] = u)$
(except for the case "$u = 0$ and $v = 1$", see Remark 1)
$= \left(\frac{1}{N} \cdot (\frac{N-1}{N})^{v-u-1}\right) \cdot (\frac{N-1}{N}) + \left(1 - \frac{1}{N} \cdot (\frac{N-1}{N})^{v-u-1}\right) \cdot (\frac{N-1}{N})^v \cdot \frac{1}{N}$
$= \frac{1}{N} \cdot (\frac{N-1}{N})^{v-u} + \frac{1}{N} \cdot (\frac{N-1}{N})^v - \frac{1}{N^2} \cdot (\frac{N-1}{N})^{2v-u-1}$. □

Remark 1. Case 1 in the proof of Lemma 2 applies to Lemma 1 also. In case 2, i.e., when $S_v[u] \neq v$, in general we may or may not have $S_v[v] = v$. However, for $u = 0$ and $v = 1$, $(S_1[0] \neq 1) \iff (S_1[1] = 1)$, the probability of each of which is $\frac{N-1}{N}$ (note that there has been only one swap involving the indices 0 and $K[0]$ in round 1). Hence the contribution of case 2 except for "$u = 0$ and $v = 1$" would be $P(S_v[u] \neq v) \cdot P(S_v[v] = v) \cdot P(j_v + S_v[v] + K[v] = u)$, and for "$u = 0$ and $v = 1$" it would be $P(S_1[0] \neq 1) \cdot P(j_1 + S_1[1] + K[1] = 0)$ or, equivalently, $P(S_1[1] = 1) \cdot P(j_1 + S_1[1] + K[1] = 0)$.

Lemma 3. *Let $p_r^{u,v} = P(S_r[u] = v)$, for $1 \leq r \leq N$. Given $p_t^{u,v}$, i.e., $P(S_t[u] = v)$ for any intermediate round t, $max\{u, v\} < t \leq N$, $P(S_r[u] = v)$ after the r-th round of the KSA is given by*
$p_t^{u,v} \cdot (\frac{N-1}{N})^{r-t} + (1 - p_t^{u,v}) \cdot \frac{1}{N} (\frac{N-1}{N})^v \cdot \left(1 - (\frac{N-1}{N})^{r-t}\right)$, $t \leq r \leq N$.

Proof. After round t ($> max\{u, v\}$), there may be two different cases: $S_t[u] = v$ and $S_t[u] \neq v$. Both of these can contribute to the event $(S_r[u] = v)$ in the following ways.

1. $S_t[u] = v$ and the index u is not touched by any of the subsequent $r - t$ many j values. The contribution of this part is $P(S_t[u] = v) \cdot (\frac{N-1}{N})^{r-t}$ $= p_t^{u,v} \cdot (\frac{N-1}{N})^{r-t}$.
2. $S_t[u] \neq v$ and for some x in the interval $[t, r-1]$, $S_x[x] = v$ which comes into the index u from the index x by the swap in round $x+1$, and after that the index u is not touched by any of the subsequent $r-1-x$ many j values. So the contribution of the second part is given by
$P(S_t[u] \neq v) \cdot \left(\sum_{x=t}^{r-1} P(S_x[x] = v) \cdot P(j_{x+1} = u) \cdot (\frac{N-1}{N})^{r-1-x}\right)$.

Suppose, the value v remains in location v after round v. By Proposition 1, this probability, i.e., $P(S_v[v] = v)$, is $(\frac{N-1}{N})^v$. The swap in the next round

moves the value v to a random location $x = j_{v+1}$. Thus, $P(S_{v+1}[x] = v) = P(S_v[v] = v) \cdot P(j_{v+1} = x) = (\frac{N-1}{N})^v \cdot \frac{1}{N}$. For all $x > v$, until x is touched by the deterministic index i, i.e., until round $x+1$, v will remain randomly distributed. Hence, for all $x > v$, $P(S_x[x] = v) = P(S_{v+1}[x] = v) = \frac{1}{N}(\frac{N-1}{N})^v$ and

$$P(S_t[u] \neq v) \cdot \Big(\sum_{x=t}^{r-1} P(S_x[x] = v) \cdot P(j_{x+1} = u) \cdot (\tfrac{N-1}{N})^{r-1-x}\Big)$$

$$= (1 - p_t^{u,v}) \cdot \Big(\sum_{x=t}^{r-1} \tfrac{1}{N}(\tfrac{N-1}{N})^v \cdot \tfrac{1}{N} \cdot (\tfrac{N-1}{N})^{r-1-x}\Big)$$

$$= (1 - p_t^{u,v}) \cdot \tfrac{1}{N^2}(\tfrac{N-1}{N})^v \cdot \Big(\sum_{x=t}^{r-1} (\tfrac{N-1}{N})^{r-1-x}\Big) = (1 - p_t^{u,v}) \cdot \tfrac{1}{N^2}(\tfrac{N-1}{N})^v \cdot \Big(\tfrac{1-a^{r-t}}{1-a}\Big),$$

where $a = \frac{N-1}{N}$. Substituting the value of a and simplifying, we get the above probability as $(1 - p_t^{u,v}) \cdot \frac{1}{N}(\frac{N-1}{N})^v \cdot \Big(1 - (\frac{N-1}{N})^{r-t}\Big)$.

Now, combining the above two contributions, we get
$p_r^{u,v} = p_t^{u,v} \cdot (\frac{N-1}{N})^{r-t} + (1 - p_t^{u,v}) \cdot \frac{1}{N}(\frac{N-1}{N})^v \cdot \Big(1 - (\frac{N-1}{N})^{r-t}\Big)$. □

Corollary 1. *Given $p_t^{u,v}$, i.e., $P(S_t[u] = v)$ for any intermediate round t, $max\{u, v\} < t \leq N$, $P(S_N[u] = v)$ after the complete KSA is given by*
$p_t^{u,v} \cdot (\frac{N-1}{N})^{N-t} + (1 - p_t^{u,v}) \cdot \frac{1}{N}(\frac{N-1}{N})^v \cdot \Big(1 - (\frac{N-1}{N})^{N-t}\Big)$.

Proof. Substitute $r = N$ in Lemma 3. □

Theorem 1.
(1) *For $0 \leq u \leq N-2$, $u+1 \leq v \leq N-1$,*
$P(S_N[u] = v) = p_{v+1}^{u,v} \cdot (\frac{N-1}{N})^{N-1-v} + (1 - p_{v+1}^{u,v}) \cdot \frac{1}{N} \cdot \Big((\frac{N-1}{N})^v - (\frac{N-1}{N})^{N-1}\Big)$, *where*

$$p_{v+1}^{u,v} = \begin{cases} \frac{2(N-1)}{N^2} & \text{if } u = 0 \text{ and } v = 1; \\ \frac{1}{N} \cdot (\frac{N-1}{N})^{v-u} + \frac{1}{N} \cdot (\frac{N-1}{N})^v - \frac{1}{N^2} \cdot (\frac{N-1}{N})^{2v-u-1} & \text{otherwise.} \end{cases}$$

(2) *For $0 \leq v \leq N-1$, $v \leq u \leq N-1$,*
$P(S_N[u] = v) = \frac{1}{N} \cdot (\frac{N-1}{N})^{N-1-u} + \frac{1}{N} \cdot (\frac{N-1}{N})^{v+1} - \frac{1}{N} \cdot (\frac{N-1}{N})^{N+v-u}$.

Proof. First we prove item (1). Since $v > u$, so for any $t > v$, we will have $t > max\{u, v\}$. Substituting $t = v + 1$ in Corollary 1, we have
$P(S_N[u] = v) = p_{v+1}^{u,v} \cdot (\frac{N-1}{N})^{N-1-v} + (1 - p_{v+1}^{u,v}) \cdot \frac{1}{N}(\frac{N-1}{N})^v \cdot \Big(1 - (\frac{N-1}{N})^{N-1-v}\Big)$
$= p_{v+1}^{u,v} \cdot (\frac{N-1}{N})^{N-1-v} + (1 - p_{v+1}^{u,v}) \cdot \frac{1}{N} \cdot \Big((\frac{N-1}{N})^v - (\frac{N-1}{N})^{N-1}\Big)$. Now, from Lemma 2, we get $p_{v+1}^{u,v} = \frac{1}{N} \cdot (\frac{N-1}{N})^{v-u} + \frac{1}{N} \cdot (\frac{N-1}{N})^v - \frac{1}{N^2} \cdot (\frac{N-1}{N})^{2v-u-1}$, except for "$u = 0$ and $v = 1$". Also, Lemma 1 gives $p_2^{0,1} = \frac{2(N-1)}{N^2}$. Substituting these values of $p_{v+1}^{u,v}$, we get the result.

Now we prove item (2). Here we have $u \geq v$. So for any $t > u$, we will have $t > max\{u, v\}$. Substituting $t = u + 1$ in Corollary 1, we have
$P(S_N[u] = v) = p_{u+1}^{u,v} \cdot (\frac{N-1}{N})^{N-1-u} + (1 - p_{u+1}^{u,v}) \cdot \frac{1}{N}(\frac{N-1}{N})^v \cdot \Big(1 - (\frac{N-1}{N})^{N-1-u}\Big)$.

As $p_{u+1}^{u,v} = P(S_{u+1}[u] = v) = \frac{1}{N}$ (see proof of Proposition 2), substituting this in the above expression, we get
$P(S_N[u] = v) = \frac{1}{N} \cdot (\frac{N-1}{N})^{N-1-u} + (1 - \frac{1}{N}) \cdot \frac{1}{N} (\frac{N-1}{N})^{v} \cdot \left(1 - (\frac{N-1}{N})^{N-1-u}\right)$
$= \frac{1}{N} \cdot (\frac{N-1}{N})^{N-1-u} + \frac{1}{N} \cdot (\frac{N-1}{N})^{v+1} - \frac{1}{N} \cdot (\frac{N-1}{N})^{N+v-u}$. □

We like to mention that our final formulae in Theorem 1 are very close to the results presented in [8] apart from some minor differences as terms with N^2 in the denominator or a difference in 1 in the power. These differences are negligible and we have also checked by calculating the numerical values of the theoretical results that for $N = 256$, the maximum absolute difference between our results and the results of [8] is 0.000025 as well as the average of absolute differences is 0.000005.

However, our approach is different from that of [8]. In [8], the idea of relative positions is introduced. If the current deterministic index is i, then relative position a means the position $(i + 1 + a)$ mod N. The transfer function $T(a, b, r)$, which represents the probability that value in relative position a in S will reach relative position b in the permutation generated from S by executing r RC4 rounds, has the following explicit form by [8, Claim C.3.3]: $T(a, b, r) = p(q^a + q^{r-(b+1)} - q^{a+r-(b+1)})$ if $a \leq b$ and $T(a, b, r) = p(q^a + q^{r-(b+1)})$ if $a > b$, where $p = \frac{1}{N}$ and $q = (\frac{N-1}{N})$. This solution is obtained by solving a recurrence [8, Equation C.3.1] which expresses $T(a, b, r)$ in terms of $T(a - 1, b - 1, r - 1)$. Instead, we use the probabilities $P(S_t[u] = v)$ in order to calculate the probabilities $P(S_r[u] = v)$ which immediately gives $P(S_N[u] = v)$ with $r = N$. When $v > u$, we take $t = v + 1$ and when $v \leq u$, we take $t = u + 1$ (see Theorem 1). However, the values $u+1$ and $v+1$ are not special. If we happen to know the probabilities $P(S_t[u] = v)$ at any round t between $max\{u, v\} + 1$ and N, then we can arrive at the probabilities $P(S_r[u] = v)$ using Lemma 3. The recurrence relation in [8] is over three variables a, b and r, and at each step each of these three variables is reduced by one. On the other hand, our model has the following features.

1. It relates four variables u, v, t and r which respectively denote any index u in the permutation (analogous to b), any value $v \in [0, \ldots N - 1]$ (analogous to the value at a), any round $t > max\{u, v\}$ and a particular round $r \geq t$.
2. Though in our formulation we do not solve any recurrence relation and provide a direct proof, it can be considered analogous to a recurrence over a single variable r, the other two variables u and v remaining fixed.

3 Anomaly Pairs and New Distinguishers

To evaluate how closely our theoretical formulae tally with the experimental results, we use average percentage absolute error $\bar{\epsilon}$. Let $p_N^{u,v}$ and $q_N^{u,v}$ respectively denote the theoretical and the experimental value of the probability $P(S_N[u] = v)$, $0 \leq u \leq N - 1$, $0 \leq v \leq N - 1$. We define $\epsilon_{u,v} = \left(\frac{|p_N^{u,v} - q_N^{u,v}|}{q_N^{u,v}}\right) \cdot 100\%$

Table 1. The anomaly pairs for key length 32 bytes

u	v	$p_N^{u,v}$	$q_N^{u,v}$	$\lvert p_N^{u,v} - q_N^{u,v} \rvert$	$\epsilon_{u,v}$ (in %)
38	6	0.003846	0.003409	0.000437	12.82
38	31	0.003643	0.003067	0.000576	18.78
46	31	0.003649	0.003408	0.000241	7.07
47	15	0.003774	0.003991	0.000217	5.44
48	16	0.003767	0.003974	0.000207	5.21
66	2	0.003882	0.003372	0.000510	15.12
66	63	0.003454	0.002797	0.000657	23.49
70	63	0.003460	0.003237	0.000223	6.89
128	0	0.003900	0.003452	0.000448	12.98
128	127	0.003303	0.002440	0.000863	35.37
130	127	0.003311	0.003022	0.000289	9.56

and $\bar{\epsilon} = \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \epsilon_{u,v}$. We ran experiments for 100 million randomly chosen secret keys of 32 bytes and found that $\bar{\epsilon} = 0.22\%$. The maximum of the $\epsilon_{u,v}$'s was 35.37% and it occured for $u = 128$ and $v = 127$. Though the maximum error is quite high, we find that out of $N^2 = 65536$ (with $N = 256$) many $\epsilon_{u,v}$'s, only 11 ($< 0.02\%$ of 65536) exceeded the 5% error margin. These cases are summarized Table 1 below. We call the pairs (u, v) for which $\epsilon_{u,v} > 5\%$ as *anomaly pairs*.

The experimental values of $P(S_N[u] = v)$ match with the theoretical values given by our formula except at these few anomaly pairs. For example, $q_N^{38,v}$ follows the pattern predicted by $p_N^{38,v}$ for all v's, $0 \leq v \leq 255$ except at $v = 6$ and $v = 31$ as pointed out in Table 1.

We experimented with different key lengths (100 million random keys for each key length) and found that the location of the anomaly pairs and the total number of anomaly pairs vary with the key lengths in certain cases. Table 2 shows the number n_5 of anomaly pairs (when $\epsilon_{u,v} > 5\%$) for different key lengths l (in bytes) along with the average $\bar{\epsilon}$ and the maximum ϵ_{max} of the $\epsilon_{u,v}$'s. u_{max} and v_{max} are the (u, v) values which correspond to ϵ_{max}. Though for some key lengths there are more than a hundred anomaly pairs, most of them have $\epsilon_{u,v} \leq 10\%$. To illustrate this, we add the column n_{10} which shows how many of the anomaly pairs exceed the 10% error margin. The two rightmost columns show

Table 2. The number and percentage of anomaly pairs along with the average and maximum error for different key lengths

l	$\bar{\epsilon}$ (in %)	ϵ_{max} (in %)	u_{max}	v_{max}	n_5	n_{10}	n_5 (in %)	n_{10} (in %)
5	0.75	73.67	9	254	1160	763	1.770	1.164
8	0.48	42.48	15	255	548	388	0.836	0.592
12	0.30	21.09	23	183	293	198	0.447	0.302
15	0.25	11.34	44	237	241	2	0.368	0.003
16	0.24	35.15	128	127	161	7	0.246	0.011
20	0.20	5.99	30	249	3	0	0.005	0.000
24	0.19	4.91	32	247	0	0	0.000	0.000
30	0.19	6.54	45	29	1	0	0.002	0.000
32	0.22	35.37	128	127	11	6	0.017	0.009
48	0.18	4.24	194	191	0	0	0.000	0.000
64	0.26	35.26	128	127	6	4	0.009	0.006
96	0.21	4.52	194	191	0	0	0.000	0.000
128	0.34	37.00	128	127	3	2	0.005	0.003
256	0.46	2.58	15	104	0	0	0.000	0.000

what percentage of $256^2 = 65536$ (total number of (u, v) pairs) are the numbers n_5 and n_{10}.

These results indicate that as the key length increases, the proportion of anomaly pairs tends to decrease. With 256 bytes key, we have no anomaly pair with $\epsilon_{u,v} > 5\%$, i.e., $n_5 = 0$. It has also been pointed out in [8] that as the key length increases, the actual random behaviour of the key is demonstrated and that is why the number of anomaly pairs decrease and experimental results match the theoretical formulae. In [8, Section 6.3.2] the anomalies are discussed for rows and columns 9, 19 and also for the diagonal given short keys as 5 bytes. We now discuss these results with more details and how they can be applied to distinguish the RC4 keystream from random streams.

We denote the permutation after r-th round of PRGA by S_r^G for $r \geq 1$.

Lemma 4. *Consider $B \subset [0, \ldots, N-1]$ with $|B| = b$. Let $P(S_N[r] \in B) = \frac{b}{N} + \epsilon$, where ϵ can be positive or negative. Then $P(S_{r-1}^G[r] \in B) = \frac{b}{N} + \delta$, where $\delta = (\frac{b}{N} + \epsilon) \cdot \Big((\frac{N-1}{N})^{r-1} + \big(1 - (\frac{N-1}{N})^{r-1}\big) \cdot (\frac{b-1}{N-1} - \frac{b}{N}) \Big) - \frac{b}{N} \cdot (\frac{N-1}{N})^{r-1}$, $r \geq 1$.*

Proof. The event $(S_{r-1}^G[r] \in B)$ can occur in three ways.

1. $S_N[r] \in B$ and the index r is not touched by any of the $r-1$ many j values during the first $r-1$ rounds of the PRGA. The contribution of this part is $(\frac{b}{N} + \epsilon) \cdot (\frac{N-1}{N})^{r-1}$.
2. $S_N[r] \in B$ and index r is touched by at least one of the $r-1$ many j values during the first $r-1$ rounds of the PRGA. Further, after the swap(s), the value $S_N[r]$ remains in the set B. This will happen with probability $(\frac{b}{N} + \epsilon) \cdot \big(1 - (\frac{N-1}{N})^{r-1}\big) \cdot \frac{b-1}{N-1}$.
3. $S_N[r] \notin B$ and index r is touched by at least one of the $r-1$ many j values during the first $r-1$ rounds of the PRGA. Due to the swap(s), the value $S_N[r]$ comes to the set B. This will happen with probability $(1 - \frac{b}{N} - \epsilon) \cdot \big(1 - (\frac{N-1}{N})^{r-1}\big) \cdot \frac{b}{N}$.

Adding these contributions, we get the total probability as $(\frac{b}{N} + \epsilon) \cdot \Big((\frac{N-1}{N})^{r-1} + \big(1 - (\frac{N-1}{N})^{r-1}\big) \cdot (\frac{b-1}{N-1} - \frac{b}{N}) \Big) + \frac{b}{N} - \frac{b}{N} \cdot (\frac{N-1}{N})^{r-1}$. □

Lemma 5. *If $P(S_{r-1}^G[r] \in B) = \frac{b}{N} + \delta$, then $P(z_r \in C) = \frac{b}{N} + \frac{2\delta}{N}$, where $C = \{c' | c' = r - b'$ where $b' \in B\}$, $r \geq 1$.*

Proof. The event $(z_r \in C)$ can happen in two ways.

1. $S_{r-1}^G[r] \in B$ and $z_r = r - S_{r-1}^G[r]$. From Glimpse theorem [4,6], we have $P(z_r = r - S_{r-1}^G[r]) = \frac{2}{N}$ for $r \geq 1$. Thus, the contribution of this part is $\frac{2}{N}(\frac{b}{N} + \delta)$.
2. $S_{r-1}^G[r] \notin B$ and still $z_r \in C$ due to random association. The contribution of this part is $(1 - \frac{2}{N})\frac{b}{N}$.

Adding these two contributions, we get the result. □

Theorem 2. *If* $P(S_N[r] \in B) = \frac{b}{N} + \epsilon$, *then* $P(z_r \in C) = \frac{b}{N} + \frac{2}{N} \cdot \Big[(\frac{b}{N} + \epsilon) \cdot \Big((\frac{N-1}{N})^{r-1} + \big(1 - (\frac{N-1}{N})^{r-1}\big) \cdot (\frac{b-1}{N-1} - \frac{b}{N})\Big) - \frac{b}{N} \cdot (\frac{N-1}{N})^{r-1}\Big]$, *where* $C = \{c' | c' = r - b'$ *where* $b' \in B\}$, $r \geq 1$.

Proof. The proof immediately follows by combining Lemma 4 and Lemma 5. □

From the above results, it follows that for a single value v, if $P(S_N[r] = v) = \frac{1}{N} + \epsilon$, then $P(z_r = r - v) = \frac{1}{N} + \frac{2\delta}{N}$, where the value of δ can be calculated by substituting $b = 1$ in Lemma 5. This presents a non-uniform distribution of the initial keystream output bytes z_r for small r.

In [9, Section 6], it has been pointed out that z_1 (referred as z_0 in [9]) may not be uniformly distributed due to non-uniform distribution of $S_N[1]$. The experimental results presented in [9, Figure 6] show some bias which does not match with our theoretical as well as experimental results. According to our Theorem 2, if $P(S_N[1] = v) = \frac{1}{N} + \epsilon$, then $P\big(z_1 = (1 - v) \mod 256\big) = \frac{1}{N} + \frac{2\epsilon}{N}$ and this presents the theoretical distribution of z_1.

When the bias of $S_N[r]$ towards a single value v is propagated to z_r, the final bias at z_r is very small and difficult to observe experimentally. Rather, if we start with the bias of $S_N[r]$ towards many values in some suitably chosen set B, then a sum of $b = |B|$ many probabilities is propagated to z_r according to Theorem 2, making the bias of z_r empirically observable too. For example, given $1 \leq r \leq 127$, consider the set B as the set of integers $[r + 1, \ldots, r + 128]$, i.e., $b = |B| = 128$. The theoretical formulae as well as the experimental results give $P(S_N[r] \in B) > 0.5$, and in turn we get $P(z_r \in C) > 0.5$, which is observable at the r-th keystream output byte of RC4. We have experimented with key length 32 bytes and 100 million runs for different r's and the experimental results support this theoretical claim. It is important to note that the non-uniform distribution can be observed even at the 256-th output byte z_{256}, since the deterministic index i at round 256 becomes 0 and $S_N[0]$ has a non-uniform distribution as follows from Theorem 1. For random association, $P(z_r \in C)$ should be $\frac{b}{N}$, which is not the case here and thus all these results provide distinguishers for RC4.

We have earlier pointed out that for short key lengths, there exist many anomaly pairs. We can exploit these to construct some additional distinguishers by including in the set B those values which are far away from being random. We illustrate this in the two examples below. For 5 byte secret keys, we experimentally observe over 100 million runs that $P(S_N[9] \in B) = 0.137564$ (which is much less than the theoretical value 0.214785), where B is the set of all even integers greater than or equal to 128 and less than 256, i.e., $b = |B| = 64$ and $\frac{b}{N} = 0.25$. Using Theorem 2 we get $P(z_9 \in C) = 0.249530 < 0.25$, where $C = \{c' | c' = 9 - b'$ where $b' \in B\}$. Again, for 8 byte secret keys, we observe that $P(S_N[15] \in B) = 0.160751$ (which is much less than the theoretical value 0.216581), where B is the set of all odd integers greater than or equal to 129 and less than 256, i.e., $b = |B| = 64$ once again. Theorem 2 gives $P(z_{15} \in C) = 0.249340 < 0.25$, where $C = \{c' | c' = 15 - b'$ where $b' \in B\}$. Direct experimental observations also confirm these biases of z_9 and z_{15}. Further, given

the values of δ approximately -0.1 in the above two examples, one can get new linear distinguishers for RC4 with 5 byte and 8 byte keys.

It is interesting to note that since the anomaly pairs are different for different key lengths, by suitably selecting the anomaly pairs in the set B, one can also distinguish among RC4 of different key lengths.

Acknowledgments. We thank the anonymous reviewers for detailed comments that improved editorial as well as technical presentation of this paper.

References

1. Fluhrer, S.R., McGrew, D.A.: Statistical Analysis of the Alleged RC4 Keystream Generator. In: Schneier, B. (ed.) FSE 2000. LNCS, vol. 1978, pp. 19–30. Springer, Heidelberg (2001)
2. Fluhrer, S.R., Mantin, I., Shamir, A.: Weaknesses in the Key Scheduling Algorithm of RC4. In: Vaudenay, S., Youssef, A.M. (eds.) SAC 2001. LNCS, vol. 2259, pp. 1–24. Springer, Heidelberg (2001)
3. Golic, J.: Linear statistical weakness of alleged RC4 keystream generator. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 226–238. Springer, Heidelberg (1997)
4. Jenkins, R.J.: ISAAC and RC4 (1996), `http://burtleburtle.net/bob/rand/isaac.html`
5. Mantin, I., Shamir, A.: A Practical Attack on Broadcast RC4. In: Matsui, M. (ed.) FSE 2001. LNCS, vol. 2355, pp. 152–164. Springer, Heidelberg (2002)
6. Mantin, I.: A Practical Attack on the Fixed RC4 in the WEP Mode. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 395–411. Springer, Heidelberg (2005)
7. Mantin, I.: Predicting and Distinguishing Attacks on RC4 Keystream Generator. In: Cramer, R.J.F. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 491–506. Springer, Heidelberg (2005)
8. Mantin, I.: Analysis of the Stream Cipher RC4. Master's Thesis. The Weizmann Institute of Science, Israel (2001)
9. Mironov, I.: Random Shuffles of RC4. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 304–319. Springer, Heidelberg (2002)
10. Paul, G., Rathi, S., Maitra, S.: On Non-negligible Bias of the First Output Byte of RC4 towards the First Three Bytes of the Secret Key. In: 2007 International Workshop on Coding and Cryptography, pp. 285–294 (2007)
11. Paul, G., Maitra, S.: Permutation after RC4 Key Scheduling Reveals the Secret Key. In: SAC 2007. 14th Annual Workshop on Selected Areas in Cryptography, Ottawa, Canada (2007)
12. Paul, S., Preneel, B.: A New Weakness in the RC4 Keystream Generator and an Approach to Improve the Security of the Cipher. In: Roy, B., Meier, W. (eds.) FSE 2004. LNCS, vol. 3017, pp. 245–259. Springer, Heidelberg (2004)
13. Roos, A.: A class of weak keys in the RC4 stream cipher (1995), Available at `http://marcel.wanda.ch/Archive/WeakKeys`
14. Wagner, D.: My RC4 weak keys (1995), `http://www.cs.berkeley.edu/~daw/my-posts/my-rc4-weak-keys`

Correctable Errors of Weight Half the Minimum Distance Plus One for the First-Order Reed-Muller Codes

Kenji Yasunaga and Toru Fujiwara

Graduate School of Information Science and Technology,
Osaka University, Suita 565-0871, Japan
{k-yasunaga, fujiwara}@ist.osaka-u.ac.jp

Abstract. The number of correctable/uncorrectable errors of weight half the minimum distance plus one for the first-order Reed-Muller codes is determined. From a cryptographic viewpoint, this result immediately leads to the exact number of Boolean functions of m variables with nonlinearity $2^{m-2}+1$. The notion of *larger half* and *trial set*, which is introduced by Helleseth, Kløve, and Levenshtein to describe the monotone structure of correctable/uncorrectable errors, plays a significant role in the result.

Keywords: Syndrome decoding, Reed-Muller code, correctable error, Boolean function, nonlinearity, larger half.

1 Introduction

In syndrome decoding, the correctable errors are coset leaders of a code. The syndrome decoding performs maximum likelihood decoding if a minimum weight vector in each coset is taken as the coset leader. When there are two or more minimum weight vectors in a coset, we have choices of the coset leader. If the lexicographically smallest minimum weight vector is taken as the coset leader, then both the correctable errors and the uncorrectable errors have a monotone structure. That is, when $\boldsymbol{y}$ covers $\boldsymbol{x}$ (the support of $\boldsymbol{y}$ contains that of $\boldsymbol{x}$), if $\boldsymbol{y}$ is correctable, then $\boldsymbol{x}$ is also correctable, and if $\boldsymbol{x}$ is uncorrectable, then $\boldsymbol{y}$ is also uncorrectable [1]. Using this monotone structure, Helleseth, Kløve, and Levenshtein introduced *larger halves* of codewords and *trial sets* for codes to describe the monotone structure of errors and gave an improved upper bound on the number of uncorrectable errors using these notions [3].

The binary r-th order Reed-Muller code of length 2^m corresponds to the Boolean functions of m variables with degree at most r. The first-order Reed-Muller code of length 2^m, denoted by RM_m, corresponds to the set of affine functions of m variables. The *nonlinearity* of a Boolean function f of m variables is defined as the minimum distance between f and affine functions, and is equal to the weight of the coset leader in the coset f belongs to. Hence the weight distribution of coset leaders of RM_m represents the distribution of nonlinearity

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 110–119, 2007.
© Springer-Verlag Berlin Heidelberg 2007

of Boolean functions. When the number of coset leaders of weight i is p, the number of Boolean functions with the nonlinearity i is given by $p|\mathrm{RM}_m| = p2^{m+1}$. Nonlinearity is an important criterion for cryptographic system, in particular, block ciphers and stream ciphers. There has been much study of nonlinearity of Boolean functions in cryptography, see [4,5] and references therein. The weight distributions of the cosets of RM_5 are completely determined in [6]. In general, however, it is infeasible to obtain the weight distributions of the cosets (even only the coset leaders) of RM_m. Since the minimum distance of RM_m is 2^{m-1}, the problem is to know the number of the coset leaders of weight $\geq 2^{m-2}$. The explicit expression of the number of coset leaders of weight w, which is equal to the number of correctable errors of weight w, is given only for $w = 2^{m-2}$ [7].

In this paper, we determine the number of correctable/uncorrectable errors of weight $2^{m-2}+1$ for RM_m, from which the number of Boolean functions with nonlinearity $2^{m-2}+1$ is immediately obtained. To derive this result, we mainly use the properties of larger halves and trial sets.

2 Larger Halves and Trial Sets

Let $\mathbb{F}^n$ be the set of all binary vectors of length n. Let $C \subseteq \mathbb{F}^n$ be a binary linear code of length n, dimension k, and minimum distance d. Then $\mathbb{F}^n$ is partitioned into 2^{n-k} cosets $C_1, C_2, \ldots, C_{2^{n-k}}$; $\mathbb{F}^n = \bigcup_{i=1}^{2^{n-k}} C_i$ and $C_i \cap C_j = \emptyset$ for $i \neq j$, where each $C_i = \{\boldsymbol{v}_i + \boldsymbol{c} : \boldsymbol{c} \in C\}$ with $\boldsymbol{v}_i \in \mathbb{F}^n$. The vector $\boldsymbol{v}_i$ is called the coset leader of the coset C_i, and any vector in C_i can be taken as $\boldsymbol{v}_i$.

Let H be a parity check matrix of C. The syndrome of a vector $\boldsymbol{v} \in \mathbb{F}^n$ is defined as $\boldsymbol{v}H^T$. All vectors having the same syndrome are in the same coset. Syndrome decoding associates an error vector to each syndrome. The syndrome decoder presumes that the error vector added to the received vector $\boldsymbol{y}$ is the coset leader of the coset which contains $\boldsymbol{y}$. The syndrome decoding function $D : \mathbb{F}^n \to C$ is defined as

$$D(\boldsymbol{y}) = \boldsymbol{y} + \boldsymbol{v}_i \text{ if } \boldsymbol{y} \in C_i.$$

If each $\boldsymbol{v}_i$ has the minimum weight in its coset C_i, the syndrome decoder performs as a maximum likelihood decoder.

In this paper, we take as $\boldsymbol{v}_i$ the minimum element in C_i with respect to the following total ordering $\preceq$:

$$\boldsymbol{x} \preceq \boldsymbol{y} \text{ if and only if } \begin{cases} w(\boldsymbol{x}) < w(\boldsymbol{y}), & \text{or} \\ w(\boldsymbol{x}) = w(\boldsymbol{y}) \text{ and } v(\boldsymbol{x}) \leq v(\boldsymbol{y}), \end{cases}$$

where $w(\boldsymbol{x})$ denotes the Hamming weight of a vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ and $v(\boldsymbol{x})$ denotes the numerical value of $\boldsymbol{x}$:

$$v(\boldsymbol{x}) = \sum_{i=1}^{n} x_i 2^{n-i}.$$

We write $\boldsymbol{x} \prec \boldsymbol{y}$ if $\boldsymbol{x} \preceq \boldsymbol{y}$ and $\boldsymbol{x} \neq \boldsymbol{y}$.

Let $E^0(C)$ be the set of all coset leaders of C. In the syndrome decoding, $E^0(C)$ is the set of correctable errors and $E^1(C) = \mathbb{F}^n \setminus E^0(C)$ is the set of uncorrectable errors. Since we take the minimum element with respect to $\preceq$ in each coset as its coset leader, both $E^0(C)$ and $E^1(C)$ have the following well-known monotone structure, see [1, Theorem 3.11]. Let $\subseteq$ denote a partial ordering called "covering" such that

$$\boldsymbol{x} \subseteq \boldsymbol{y} \text{ if and only if } S(\boldsymbol{x}) \subseteq S(\boldsymbol{y}),$$

where

$$S(\boldsymbol{v}) = \{i : v_i \neq 0\}$$

is the support of $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)$. Consider $\boldsymbol{x}$ and $\boldsymbol{y}$ with $\boldsymbol{x} \subseteq \boldsymbol{y}$. If $\boldsymbol{y}$ is a correctable error, then $\boldsymbol{x}$ is also correctable. If $\boldsymbol{x}$ is uncorrectable, then $\boldsymbol{y}$ is also uncorrectable. For example, let $C = \{000, 001\}$ be a code. Then $E^0(C) = \{110, 100, 010\}$ and $E^1(C) = \{001, 101, 111\}$. In this case, even if we only know the fact that the vector 110 is correctable, we can deduce the vectors 100 and 010 are correctable, since they are covered by 110. A similar thing happens when we know 001 is uncorrectable. Using this structure, Zémor showed that the residual error probability after maximum likelihood decoding displays a threshold behavior [2]. Helleseth, Kløve, and Levenshtein [3] studied this structure and introduced *larger halves* and *trial sets*.

Since the set of uncorrectable errors $E^1(C)$ has a monotone structure, $E^1(C)$ can be characterized by *minimal uncorrectable errors* in $E^1(C)$. An uncorrectable error $\boldsymbol{y} \in E^1(C)$ is minimal if there exists no $\boldsymbol{x}$ such that $\boldsymbol{x} \subset \boldsymbol{y}$ in $E^1(C)$. If we know all minimal uncorrectable errors, all uncorrectable errors can be determined from them. We denote by $M^1(C)$ the set of all minimal uncorrectable errors in C. Larger halves of a codeword $\boldsymbol{c} \in C \setminus \{\boldsymbol{0}\}$ are introduced to characterize the minimal uncorrectable errors, and are defined as minimal vectors $\boldsymbol{v}$ with respect to covering such that $\boldsymbol{v} + \boldsymbol{c} \prec \boldsymbol{v}$. Any larger half $\boldsymbol{v}$ of a codeword $\boldsymbol{c}$ is an uncorrectable error, since $\boldsymbol{v} + \boldsymbol{c} \prec \boldsymbol{v}$ and they are in the same coset. The following condition is a necessary and sufficient condition that $\boldsymbol{v} \in \mathbb{F}^n$ is a larger half of $\boldsymbol{c} \in C \setminus \{\boldsymbol{0}\}$:

$$\boldsymbol{v} \subseteq \boldsymbol{c}, \tag{1}$$

$$w(\boldsymbol{c}) \leq 2w(\boldsymbol{v}) \leq w(\boldsymbol{c}) + 2, \tag{2}$$

$$l(\boldsymbol{v}) \begin{cases} = l(\boldsymbol{c}), & \text{if } \ 2w(\boldsymbol{v}) = w(\boldsymbol{c}), \\ > l(\boldsymbol{c}), & \text{if } \ 2w(\boldsymbol{v}) = w(\boldsymbol{c}) + 2, \end{cases} \tag{3}$$

where $l(\boldsymbol{x})$ is the smallest element in $S(\boldsymbol{x})$, that is, $l(\boldsymbol{x})$ is the leftmost non-zero coordinate in the vector $\boldsymbol{x}$. The proof of equivalence between the definition and the above condition is found in the proof of Theorem 1 of [3]. Let $LH(\boldsymbol{c})$ be the set of all larger halves of $\boldsymbol{c} \in C \setminus \{\boldsymbol{0}\}$. For a subset U of $C \setminus \{\boldsymbol{0}\}$, let

$$LH(U) = \bigcup_{\boldsymbol{c} \in U} LH(\boldsymbol{c}).$$

A trial set T for a code C is defined as follows:

$$T \subseteq C \setminus \{\mathbf{0}\} \text{ is a trial set for } C \text{ if } M^1(C) \subseteq LH(T). \tag{4}$$

A codeword $\boldsymbol{c}$ is called *minimal* if $\boldsymbol{c}' \subset \boldsymbol{c}$ for $\boldsymbol{c}' \in C$ implies $\boldsymbol{c}' = \mathbf{0}$. Let C^* be the set of all minimal codewords in C. It is shown that a trial set can consist of only minimal codewords [3, Corollary 5]. Therefore, C^* is a trial set of C.

In the rest of paper, for $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{F}^n$, we write $\boldsymbol{u} \cap \boldsymbol{v}$ as the vector in $\mathbb{F}^n$ whose support is $\mathrm{S}(\boldsymbol{u}) \cap \mathrm{S}(\boldsymbol{v})$.

3 Uncorrectable Errors of Weight $2^{m-2}+1$ for RM_m

In this section, we determine the number of correctable/uncorrectable errors of weight half the minimum distance plus one for the first-order Reed-Muller code of length $n = 2^m$, denoted by RM_m.

RM_m is a code of dimension $k = m + 1$, and minimum distance $d = 2^{m-1}$, and is defined recursively as

$$\mathrm{RM}_0 = \{\mathbf{0}, \mathbf{1}\},$$
$$\mathrm{RM}_m = \bigcup_{\boldsymbol{c} \in \mathrm{RM}_{m-1}} \{\boldsymbol{c} \circ \boldsymbol{c}, \boldsymbol{c} \circ \overline{\boldsymbol{c}}\},$$

where $\boldsymbol{u} \circ \boldsymbol{v}$ denotes the concatenation of $\boldsymbol{u}$ and $\boldsymbol{v}$, and $\overline{\boldsymbol{v}} \triangleq \mathbf{1} + \boldsymbol{v}$. Since all codewords in RM_m except all-zero and all-one codewords are minimum weight codewords, $\mathrm{RM}_m^* = \mathrm{RM}_m \setminus \{\mathbf{0}, \mathbf{1}\}$.

The weights of vectors in $LH(\mathrm{RM}_m^*)$ are 2^{m-2} and $2^{m-2}+1$ from the condition (2). Let $LH^-(\boldsymbol{c})$ and $LH^+(\boldsymbol{c})$ denote the sets of larger halves of $\boldsymbol{c} \in \mathrm{RM}_m^*$ of weight 2^{m-2} and $2^{m-2}+1$, respectively. Also let $LH^-(\mathrm{RM}_m^*) = \bigcup_{\boldsymbol{c} \in \mathrm{RM}_m^*} LH^-(\boldsymbol{c})$ and $LH^+(\mathrm{RM}_m^*) = \bigcup_{\boldsymbol{c} \in \mathrm{RM}_m^*} LH^+(\boldsymbol{c})$.

Let $E^1_{2^{m-2}+1}(\mathrm{RM}_m)$ be the set of uncorrectable errors of weight $d+1 = 2^{m-2}+1$ in RM_m. The set, $E^1_{2^{m-2}+1}(\mathrm{RM}_m)$, contains $LH^+(\mathrm{RM}_m^*)$, and $LH^+(\mathrm{RM}_m^*)$ contains all minimal uncorrectable errors of the weight from (4). Therefore, the remaining uncorrectable errors in $E^1_{2^{m-2}+1}(\mathrm{RM}_m)$ are non-minimal.

We will evaluate $|E^1_{2^{m-2}+1}(\mathrm{RM}_m)|$ by partitioning the set into two subsets. The first subset consists of the vectors that is covered by some codeword in RM_m^*. Any $\boldsymbol{v} \in \mathbb{F}^n$ of weight $2^{m-2}+1$ covered by $\boldsymbol{c} \in \mathrm{RM}_m$ is uncorrectable, since the coset to which $\boldsymbol{v}$ belongs contains the smaller weight vector $\boldsymbol{c} + \boldsymbol{v}$. The second one consists of the remaining non-minimal vectors.

Now, we evaluate the number of vectors in the first subset. It contains $\binom{2^{m-1}}{2^{m-2}+1}$ vectors for each codeword in RM_m^*, and all $|\mathrm{RM}_m^*| \cdot \binom{2^{m-1}}{2^{m-2}+1}$ such vectors are distinct. This is because, if $\boldsymbol{v} \subseteq \boldsymbol{c}_1$ and $\boldsymbol{v} \subseteq \boldsymbol{c}_2$ for a vector $\boldsymbol{v}$ in the set, then we have $w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2) \geq w(\boldsymbol{v}) = 2^{m-2}+1$, which contradicts the following Lemma 1.

Lemma 1. *Let $\boldsymbol{c}_1, \boldsymbol{c}_2 \in \mathrm{RM}_m^*$ with $\boldsymbol{c}_1 \neq \boldsymbol{c}_2$. Then, it holds that*

$$w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2) = \begin{cases} 2^{m-2}, & \text{if } \boldsymbol{c}_1 + \boldsymbol{c}_2 \neq \mathbf{1}, \\ 0, & \text{otherwise.} \end{cases}$$

Proof. The statement follows from the fact that $w(\boldsymbol{c}_1 + \boldsymbol{c}_2) = w(\boldsymbol{c}_1) + w(\boldsymbol{c}_2) - 2w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2)$. That is,

$$\begin{aligned} w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2) &= \frac{w(\boldsymbol{c}_1) + w(\boldsymbol{c}_2) - w(\boldsymbol{c}_1 + \boldsymbol{c}_2)}{2} \\ &= \frac{2^{m-1} + 2^{m-1} - w(\boldsymbol{c}_1 + \boldsymbol{c}_2)}{2} \\ &= \frac{2^m - w(\boldsymbol{c}_1 + \boldsymbol{c}_2)}{2}. \end{aligned}$$

□

Next, we evaluate the number of vectors in the second subset. The vectors in the subset are non-minimal uncorrectable errors that are not covered by any codeword in RM_m^*. Such a error covers a minimal uncorrectable error of weight 2^{m-2} in $LH^-(\mathrm{RM}_m^*)$, since 2^{m-2} is the smallest weight in uncorrectable errors. Therefore, we consider the set of vectors obtained by adding a weight-one vector to a larger half in $LH^-(\mathrm{RM}_m^*)$ that are not covered by any codeword in RM_m^*.

Let

$$\begin{aligned} &\mathbb{E}^n = \{\boldsymbol{e} \in \mathbb{F}^n : w(\boldsymbol{e}) = 1\}, \\ &\mathbb{E}^n(\boldsymbol{c}) = \{\boldsymbol{e} \in \mathbb{E}^n : \boldsymbol{e} \cap \boldsymbol{c} = \mathbf{0}\}, \quad \text{for } \boldsymbol{c} \in \mathrm{RM}_m^*. \end{aligned}$$

Then, the second subset can be represented as $\mathrm{X}_m \setminus \mathrm{Y}_m$, where

$$\begin{aligned} \mathrm{X}_m &= \{\boldsymbol{v} + \boldsymbol{e} : \boldsymbol{v} \in LH^-(\boldsymbol{c}) \text{ with } \boldsymbol{c} \in \mathrm{RM}_m^*, \boldsymbol{e} \in \mathbb{E}^n(\boldsymbol{c})\}, \\ \mathrm{Y}_m &= \{\boldsymbol{u} \in \mathrm{X}_m : \boldsymbol{u} \subseteq \boldsymbol{c} \text{ for some } \boldsymbol{c} \in \mathrm{RM}_m^*\}. \end{aligned}$$

From the above discussion, we have

$$|E^1_{2^{m-2}+1}(\mathrm{RM}_m)| = 2(2^m - 1)\binom{2^{m-1}}{2^{m-2}+1} + |\mathrm{X}_m \setminus \mathrm{Y}_m|. \tag{5}$$

For X_m and Y_m, we define the corresponding multisets $\tilde{\mathrm{X}}_m$ and $\tilde{\mathrm{Y}}_m$. That is, $\tilde{\mathrm{X}}_m$ is a multiset of vectors obtained by adding a weight-one vector $\boldsymbol{e}$ to larger halves $\boldsymbol{v} \in LH^-(\boldsymbol{c})$ satisfying $\boldsymbol{c} \cap \boldsymbol{e} = \mathbf{0}$ for each $\boldsymbol{c} \in \mathrm{RM}_m^*$. The set $\tilde{\mathrm{Y}}_m$ is a multiset of vectors in $\tilde{\mathrm{X}}_m$ that are covered by some codeword in RM_m^*. Then we have

$$\begin{aligned} |\tilde{\mathrm{X}}_m| &= |\mathrm{RM}_m^*| \cdot \binom{2^{m-1}-1}{2^{m-2}-1} \cdot 2^{m-1} \\ &= 2^{m-1}(2^m - 1)\binom{2^{m-1}}{2^{m-2}}, \end{aligned} \tag{6}$$

since the number of larger halves of each codeword is $\binom{2^{m-1}-1}{2^{m-2}-1}$ from (1)–(3).

We will evaluate $|\mathrm{X}_m \setminus \mathrm{Y}_m|$ by using $\tilde{\mathrm{X}}_m$ and $\tilde{\mathrm{Y}}_m$. First, we will show that the multiplicity of vectors in $\tilde{\mathrm{X}}_m \setminus \tilde{\mathrm{Y}}_m$ is not greater than 2 by using the following lemma.

Lemma 2. *Let $\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3$ be distinct codewords in RM_m^*. Then it holds that*

$$w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2 \cap \boldsymbol{c}_3) = \begin{cases} 2^{m-2}, & \text{if } \boldsymbol{c}_1 + \boldsymbol{c}_2 + \boldsymbol{c}_3 = \mathbf{1}, \\ 0, & \text{if } \boldsymbol{c}_i + \boldsymbol{c}_j = \mathbf{1} \text{ for some } i, j \text{ with } 1 \le i \ne j \le 3, \\ 2^{m-3}, & \text{otherwise.} \end{cases}$$

Proof. The statement follows from the fact that $w(\boldsymbol{c}_1 + \boldsymbol{c}_2 + \boldsymbol{c}_3) = w(\boldsymbol{c}_1) + w(\boldsymbol{c}_2) + w(\boldsymbol{c}_3) - 2(w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2) + w(\boldsymbol{c}_2 \cap \boldsymbol{c}_3) + w(\boldsymbol{c}_1 \cap \boldsymbol{c}_3)) + 4w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2 \cap \boldsymbol{c}_3)$ and Lemma 1. □

From the lemma, we see that $w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2 \cap \boldsymbol{c}_3) = 2^{m-3}$ if and only if $\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3, 1$ are linearly independent, that is, $a_1\boldsymbol{c}_1 + a_2\boldsymbol{c}_2 + a_3\boldsymbol{c}_3 + a_4\mathbf{1} = \mathbf{0}$ yields $a_1 = a_2 = a_3 = a_4 = 0$.

Lemma 3. *The multiplicity of any vector in $\tilde{\mathrm{X}}_m \setminus \tilde{\mathrm{Y}}_m$ is less than or equal to 2 for $m \ge 5$.*

Proof. Let $\boldsymbol{c}_1$, $\boldsymbol{c}_2$, $\boldsymbol{c}_3$ be distinct codewords in RM_m^*. For $1 \le i \le 3$, suppose there exist $\boldsymbol{v}_i, \boldsymbol{e}_i, \boldsymbol{u}$ such that $\boldsymbol{v}_i \in LH^-(\boldsymbol{c}_i)$, $\boldsymbol{e}_i \in \mathbb{E}^n(\boldsymbol{c}_i)$, $\boldsymbol{u} = \boldsymbol{v}_i + \boldsymbol{e}_i$, and there exists no $\boldsymbol{c}_4 \in \mathrm{RM}_m^*$ satisfying $\boldsymbol{u} \subseteq \boldsymbol{c}_4$. First note that $\boldsymbol{c}_1$, $\boldsymbol{c}_2$, $\boldsymbol{c}_3$, and $\mathbf{1}$ must be linearly independent for existing the above $\boldsymbol{v}_i, \boldsymbol{e}_i, \boldsymbol{u}$ for $1 \le i \le 3$ for $m \ge 4$.

If $\boldsymbol{v}_1 = \boldsymbol{v}_2$, then $\boldsymbol{v}_1 = \boldsymbol{c}_1 \cap \boldsymbol{c}_2 \subseteq \mathbf{1} + \boldsymbol{c}_1 + \boldsymbol{c}_2$ and $\boldsymbol{e}_1 = \boldsymbol{e}_2 \subseteq \mathbf{1} + \boldsymbol{c}_1 + \boldsymbol{c}_2$, and thus $\boldsymbol{v}_1 + \boldsymbol{e}_1 \subseteq \mathbf{1} + \boldsymbol{c}_1 + \boldsymbol{c}_2$, leading to the contradiction. Therefore $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, $\boldsymbol{v}_3$ are distinct, and so are $\boldsymbol{e}_1$, $\boldsymbol{e}_2$, $\boldsymbol{e}_3$. Then $w(\boldsymbol{v}_1 \cap \boldsymbol{v}_2 \cap \boldsymbol{v}_3) = 2^{m-2} - 2$, and thus $w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2 \cap \boldsymbol{c}_3) \ge w(\boldsymbol{v}_1 \cap \boldsymbol{v}_2 \cap \boldsymbol{v}_3) = 2^{m-2} - 2$. On the other hand, $w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2 \cap \boldsymbol{c}_3) = 2^{m-3}$ from Lemma 2. Thus we have $2^{m-3} \ge 2^{m-2} - 2$. The contradiction arises when $m \ge 5$. □

Thus, the size of $\mathrm{X}_m \setminus \mathrm{Y}_m$ is represented as follows.

$$|\mathrm{X}_m \setminus \mathrm{Y}_m| = |\tilde{\mathrm{X}}_m| - |\tilde{\mathrm{Y}}_m| - \frac{|\tilde{\mathrm{Z}}_m|}{2}, \tag{7}$$

where $\tilde{\mathrm{Z}}_m$ is the multiset defined as

$$\tilde{\mathrm{Z}}_m = \{\boldsymbol{v} \in \tilde{\mathrm{X}}_m : \boldsymbol{v} \not\subseteq \boldsymbol{c} \text{ for any } \boldsymbol{c} \in \mathrm{RM}_m^*, \text{ the multiplicity of } \boldsymbol{v} \text{ is } 2\}.$$

We will determine $|\tilde{\mathrm{Y}}_m|$ and $|\tilde{\mathrm{Z}}_m|$. The next lemma is useful to evaluate $|\tilde{\mathrm{Y}}_m|$.

Lemma 4. *Let $\boldsymbol{c}_1, \boldsymbol{c}_2 \in \mathrm{RM}_m^*$. Then*

1. *there exist $\boldsymbol{v} \in LH^-(\boldsymbol{c}_1)$, $\boldsymbol{e} \in \mathbb{E}^n(\boldsymbol{c}_1)$ such that $\boldsymbol{v} + \boldsymbol{e} \subseteq \boldsymbol{c}_2$ if and only if*

$$\boldsymbol{c}_1 \ne \boldsymbol{c}_2 \quad \text{and} \quad l(\boldsymbol{c}_1) \in S(\boldsymbol{c}_2); \tag{8}$$

2. if (8) holds,

$$\{(\boldsymbol{v}, \boldsymbol{e}) : \boldsymbol{v} \in LH^{-}(\boldsymbol{c}_1), \boldsymbol{e} \in \mathbb{E}^n(\boldsymbol{c}_1), \boldsymbol{v} + \boldsymbol{e} \subseteq \boldsymbol{c}_2\} = \{(\boldsymbol{c}_1 \cap \boldsymbol{c}_2, \boldsymbol{e}) : \boldsymbol{e} \in \mathbb{E}^n, S(\boldsymbol{e}) \subseteq S(\boldsymbol{c}_2) \setminus S(\boldsymbol{c}_1)\}. \quad (9)$$

Proof. (First part) The only if part is obvious. We prove the if part. Let $\boldsymbol{v} = \boldsymbol{c}_1 \cap \boldsymbol{c}_2$. Since $\boldsymbol{c}_1 \neq \boldsymbol{c}_2$ and $\boldsymbol{c}_1 + \boldsymbol{c}_2 \neq \mathbf{1}$ from (8), we have $w(\boldsymbol{v}) = 2^{m-2}$ from Lemma 1. We have $l(\boldsymbol{v}) = l(\boldsymbol{c}_1)$ from $l(\boldsymbol{c}_1) \in S(\boldsymbol{c}_2)$. Thus $\boldsymbol{v} \in LH^{-}(\boldsymbol{c}_1)$. Clearly, we can take $\boldsymbol{e} \in \mathbb{E}^n(\boldsymbol{c}_1)$ such that $\boldsymbol{v} + \boldsymbol{e} \subseteq \boldsymbol{c}_2$.

(Second part) The $\supseteq$ part is obvious, so we show the $\subseteq$ part. Since $\boldsymbol{v} \subseteq \boldsymbol{c}_1$ and $\boldsymbol{v} \subseteq \boldsymbol{c}_2$, it holds $w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2) \geq w(\boldsymbol{v}) = 2^{m-2}$. On the other hand, $w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2) = 2^{m-2}$. Therefore we have $\boldsymbol{v} = \boldsymbol{c}_1 \cap \boldsymbol{c}_2$. It immediately follows that $S(\boldsymbol{e}) \subseteq S(\boldsymbol{c}_2) \setminus S(\boldsymbol{c}_1)$ from $\boldsymbol{c}_1 \cap \boldsymbol{e} = \mathbf{0}$ and $\boldsymbol{v} + \boldsymbol{e} \subseteq \boldsymbol{c}_2$. □

From Lemma 4, $\boldsymbol{v} + \boldsymbol{e} \in \tilde{\mathrm{X}}_m$ is covered by every $\boldsymbol{c}_2 \in \mathrm{RM}_m^*$ satisfying (8). The number of codewords $\boldsymbol{c}_2$ satisfying (8) is $|\mathrm{RM}_m|/2 - 2 = 2^m - 2$. There are $|S(\boldsymbol{c}_2) \setminus S(\boldsymbol{c}_1)| = 2^{m-2}$ choices of $\boldsymbol{e}$ from (9). Thus we have

$$\begin{aligned} |\tilde{\mathrm{Y}}_m| &= |\mathrm{RM}_m^*| \cdot (2^m - 2) \cdot 2^{m-2} \\ &= 2^m(2^m - 1)(2^{m-1} - 1). \end{aligned} \quad (10)$$

The following lemma is useful to derive $|\tilde{\mathrm{Z}}_m|$.

Lemma 5. *Let $\boldsymbol{u} \in \tilde{\mathrm{X}}_m$ of multiplicity 2. That is, $\boldsymbol{u}$ is represented as $\boldsymbol{u} = \boldsymbol{v}_1 + \boldsymbol{e}_1 = \boldsymbol{v}_2 + \boldsymbol{e}_2$ where $\boldsymbol{v}_i \in LH^{-}(\boldsymbol{c}_i)$, $\boldsymbol{c}_i \in \mathrm{RM}_m^*$, $\boldsymbol{e}_i \in \mathbb{E}^n(\boldsymbol{c}_i)$ for $i = 1, 2$, and $\boldsymbol{c}_1 \neq \boldsymbol{c}_2$. Then,*

1. *for $m \geq 3$, $\boldsymbol{c}_1 + \boldsymbol{c}_2 \neq \mathbf{1}$,*
2. *for $m \geq 5$, there exists $\boldsymbol{c}_3 \in \mathrm{RM}_m^*$ such that $\boldsymbol{u} \subseteq \boldsymbol{c}_3$ if and only if $\boldsymbol{e}_1 = \boldsymbol{e}_2$.*

Proof. The first part holds, since $\boldsymbol{v}_1 + \boldsymbol{e}_1 = \boldsymbol{v}_2 + \boldsymbol{e}_2$ cannot hold for $m \geq 3$ if $\boldsymbol{c}_1 + \boldsymbol{c}_2 = \mathbf{1}$. Now we prove the second part.

(Only if part) We have $\boldsymbol{c}_1 \neq \boldsymbol{c}_3$ from $\boldsymbol{v}_1 + \boldsymbol{e}_1 \not\subseteq \boldsymbol{c}_1$ and $\boldsymbol{v}_1 + \boldsymbol{e}_1 \subseteq \boldsymbol{c}_3$. Since $\boldsymbol{v}_1 \subseteq \boldsymbol{c}_1$, and $\boldsymbol{v}_1 \subseteq \boldsymbol{c}_3$, we have $\boldsymbol{v}_1 = \boldsymbol{c}_1 \cap \boldsymbol{c}_3$. Equivalently, $\boldsymbol{v}_2 = \boldsymbol{c}_2 \cap \boldsymbol{c}_3$. Then $\boldsymbol{v}_1 \cap \boldsymbol{v}_2 = \boldsymbol{c}_1 \cap \boldsymbol{c}_2 \cap \boldsymbol{c}_3$, and hence $w(\boldsymbol{v}_1 \cap \boldsymbol{v}_2) = w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2 \cap \boldsymbol{c}_3)$. Since $\boldsymbol{c}_1$, $\boldsymbol{c}_2$, $\boldsymbol{c}_3$ are distinct, $w(\boldsymbol{c}_1 \cap \boldsymbol{c}_2 \cap \boldsymbol{c}_3)$ is either 2^{m-2}, 2^{m-3}, or 0. On the other hand, $w(\boldsymbol{v}_1 \cap \boldsymbol{v}_2)$ is 2^{m-2} if $\boldsymbol{v}_1 = \boldsymbol{v}_2$, and is $2^{m-2} - 2$ otherwise, since $\boldsymbol{v}_1 + \boldsymbol{e}_1 = \boldsymbol{v}_2 + \boldsymbol{e}_2$. Therefore $w(\boldsymbol{v}_1 \cap \boldsymbol{v}_2) = 2^{m-2}$ for $m \geq 5$, since $2^{m-3} \neq 2^{m-2} - 2$. Hence $\boldsymbol{v}_1 = \boldsymbol{v}_2$, and thus $\boldsymbol{e}_1 = \boldsymbol{e}_2$.

(If part) Since $\boldsymbol{e}_1 = \boldsymbol{e}_2$ and $\boldsymbol{c}_1 \neq \boldsymbol{c}_2$, we have $\boldsymbol{v}_1 = \boldsymbol{v}_2 = \boldsymbol{c}_1 \cap \boldsymbol{c}_2 \subseteq \mathbf{1} + \boldsymbol{c}_1 + \boldsymbol{c}_2$. Since $\boldsymbol{e}_1 \cap \boldsymbol{c}_1 = \boldsymbol{e}_2 \cap \boldsymbol{c}_2 = \boldsymbol{e}_1 \cap \boldsymbol{c}_2 = \mathbf{0}$, we have $\boldsymbol{e}_1 \subseteq \mathbf{1} + \boldsymbol{c}_1 + \boldsymbol{c}_2$. By taking $\boldsymbol{c}_3 = \mathbf{1} + \boldsymbol{c}_1 + \boldsymbol{c}_2$, we have $\boldsymbol{u} = \boldsymbol{v}_1 + \boldsymbol{e}_1 \subseteq \boldsymbol{c}_3$. □

From Lemma 5, for each $\boldsymbol{c}_1 \in \mathrm{RM}_m^*$, $|\tilde{\mathrm{Z}}_m|$ is obtained by counting all patterns in $\{\boldsymbol{v}_1 + \boldsymbol{e}_1 : \boldsymbol{v}_1 \in LH^{-}(\boldsymbol{c}_1), \boldsymbol{e}_1 \in \mathbb{E}^n(\boldsymbol{c}_1)\}$ such that $\boldsymbol{v}_1 + \boldsymbol{e}_1 = \boldsymbol{v}_2 + \boldsymbol{e}_2$ for some

$\boldsymbol{v}_2, \boldsymbol{e}_2$ with $\boldsymbol{v}_2 \in LH^-(\boldsymbol{c}_2), \boldsymbol{c}_2 \in \mathrm{RM}_m^* \setminus \{\boldsymbol{c}_1\}, \boldsymbol{e}_2 \in \mathbb{E}^n(\boldsymbol{c}_2)$ and $\boldsymbol{e}_1 \neq \boldsymbol{e}_2$. We will count such $\boldsymbol{v}_1 + \boldsymbol{e}_1$ for each $\boldsymbol{c}_1 \in \mathrm{RM}_m^*$.

We introduce some notations. Let $S_m = \{l(\boldsymbol{c}) : \boldsymbol{c} \in \mathrm{RM}_m\}$. From the definition of RM_m, $S_m = \{s_1, s_2, \ldots, s_k\}$, where

$$s_i = \begin{cases} 1, & \text{for } i = 1, \\ 2^{i-2} + 1, & \text{for } 2 \leq i \leq k = m + 1. \end{cases}$$

Also define

$$C_m(s_i) = \{\boldsymbol{c} \in \mathrm{RM}_m^* : l(\boldsymbol{c}) = s_i\}.$$

Then, we have

$$|C_m(s_i)| = \begin{cases} 2^m - 1, & \text{for } i = 1, \\ 2^{m+1-i}, & \text{for } 2 \leq i \leq m + 1. \end{cases} \tag{11}$$

Now we are ready to evaluate $|\tilde{\mathrm{Z}}_m|$. There are three cases to be considered.

1. When $l(\boldsymbol{c}_1) = l(\boldsymbol{c}_2)$; we choose $\boldsymbol{w}$ such that $\boldsymbol{w} \subseteq \boldsymbol{c}_1 \cap \boldsymbol{c}_2, w(\boldsymbol{w}) = 2^{m-2} - 1$, and $l(\boldsymbol{w}) = l(\boldsymbol{c}_1 \cap \boldsymbol{c}_2)$. We choose $\boldsymbol{e}_2$ so that $S(\boldsymbol{e}_2) \subseteq S(\boldsymbol{c}_1) \setminus S(\boldsymbol{c}_2)$, and choose $\boldsymbol{e}_1$ so that $S(\boldsymbol{e}_1) \subseteq S(\boldsymbol{c}_2) \setminus S(\boldsymbol{c}_1)$. Then letting $\boldsymbol{v}_1 = \boldsymbol{w} + \boldsymbol{e}_2$ and $\boldsymbol{v}_2 = \boldsymbol{w} + \boldsymbol{e}_1$ gives vectors as $\boldsymbol{v}_1 + \boldsymbol{e}_1 = \boldsymbol{v}_2 + \boldsymbol{e}_2$. There are $(2^{m-2} - 1) \cdot 2^{m-2} \cdot 2^{m-2}$ such $\boldsymbol{v}_1 + \boldsymbol{e}_1$.

   For each codeword $\boldsymbol{c}_1$ in $C_m(s_i)$, there are $|C_m(s_i)| - 1$ codewords $\boldsymbol{c}_2$ in RM_m^* satisfying $l(\boldsymbol{c}_1) = l(\boldsymbol{c}_2)$.
2. When $l(\boldsymbol{c}_1) > l(\boldsymbol{c}_2)$; since $\boldsymbol{v}_1 \in LH^-(\boldsymbol{c}_1)$ and $\boldsymbol{v}_2 \in LH^-(\boldsymbol{c}_2)$, the $l(\boldsymbol{c}_2)$-th bit of $\boldsymbol{e}_1$ is one.

   (a) If the $l(\boldsymbol{c}_1)$-th bit of $\boldsymbol{c}_2$ is one; we choose $\boldsymbol{w}$ such that $\boldsymbol{w} \subseteq \boldsymbol{c}_1 \cap \boldsymbol{c}_2, w(\boldsymbol{w}) = 2^{m-2} - 1$, and $l(\boldsymbol{w}) = l(\boldsymbol{c}_1 \cap \boldsymbol{c}_2)$. We choose $\boldsymbol{e}_2$ so that $S(\boldsymbol{e}_2) \subseteq S(\boldsymbol{c}_1) \setminus S(\boldsymbol{c}_2)$. Then letting $\boldsymbol{v}_1 = \boldsymbol{w} + \boldsymbol{e}_2$ and $\boldsymbol{v}_2 = \boldsymbol{w} + \boldsymbol{e}_1$ gives vectors as $\boldsymbol{v}_1 + \boldsymbol{e}_1 = \boldsymbol{v}_2 + \boldsymbol{e}_2$. There are $(2^{m-2} - 1) \cdot 2^{m-2}$ such $\boldsymbol{v}_1 + \boldsymbol{e}_1$.

   For each codeword $\boldsymbol{c}_1$ in $C_m(s_i)$ with $i \geq 2$, there are $\left(\left(\sum_{j<i} |C_m(s_j)| + 1\right)/2 - 1\right)$ codewords $\boldsymbol{c}_2$ in RM_m^* satisfying $l(\boldsymbol{c}_1) \in S(\boldsymbol{c}_2)$.

   (b) If the $l(\boldsymbol{c}_1)$-th bit of $\boldsymbol{c}_2$ is zero; then $\boldsymbol{e}_2$ must be the vector having one in the $l(\boldsymbol{c}_1)$-th bit. We choose $\boldsymbol{w}$ such that $\boldsymbol{w} \subseteq \boldsymbol{c}_1 \cap \boldsymbol{c}_2$ and $w(\boldsymbol{w}) = 2^{m-2} - 1$. Then letting $\boldsymbol{v}_1 = \boldsymbol{w} + \boldsymbol{e}_2$ and $\boldsymbol{v}_2 = \boldsymbol{w} + \boldsymbol{e}_1$ gives vectors as $\boldsymbol{v}_1 + \boldsymbol{e}_1 = \boldsymbol{v}_2 + \boldsymbol{e}_2$. There are 2^{m-2} such $\boldsymbol{v}_1 + \boldsymbol{e}_1$.

   For each codeword $\boldsymbol{c}_1$ in $C_m(s_i)$ with $i \geq 2$, there are $\left(\left(\sum_{j<i} |C_m(s_j)| + 1\right)/2 - 1\right)$ codewords $\boldsymbol{c}_2$ in RM_m^* satisfying $l(\boldsymbol{c}_1) \notin S(\boldsymbol{c}_2)$ and $\boldsymbol{c}_1 + \boldsymbol{c}_2 \neq \mathbf{1}$.
3. When $l(\boldsymbol{c}_1) < l(\boldsymbol{c}_2)$; the number of vectors we should count is equal to that for the second case.

From the above analysis, we have

$$\begin{aligned}|\tilde{\mathrm{Z}}_m| &= \sum_{i=1}^{m+1} |C_m(s_i)|(|C_m(s_i)|-1)\cdot(2^{m-2}-1)(2^{m-2})^2 \\ &\quad + 2\sum_{i=2}^{m+1} |C_m(s_i)| \left(\left(\sum_{j=1}^{i-1}|C_m(s_j)|+1\right)\cdot\frac{1}{2}-1\right)\cdot(2^{m-2}-1)2^{m-2} \\ &\quad + 2\sum_{i=2}^{m+1} |C_m(s_i)| \left(\left(\sum_{j=1}^{i-1}|C_m(s_j)|+1\right)\cdot\frac{1}{2}-1\right)\cdot 2^{m-2} \\ &= 2^{2m-3}\binom{2^m}{3}. \end{aligned} \tag{12}$$

From (5), (6), (7), (10), and (12), we can determine the number of uncorrectable errors of weight $2^{m-2}+1$ for RM_m.

Theorem 1. *For $m \geq 5$,*

$$|E^1_{2^{m-2}+1}(\mathrm{RM}_m)| = 4(2^m-1)(2^{m-3}+1)\binom{2^{m-1}}{2^{m-2}+1} - (4^{m-2}+3)\binom{2^m}{3}.$$

The number of correctable errors of weight $2^{m-2}+1$, $|E^0_{2^{m-2}+1}(\mathrm{RM}_m)|$, is obtained from the equation,

$$|E^0_{2^{m-2}+1}(\mathrm{RM}_m)| + |E^1_{2^{m-2}+1}(\mathrm{RM}_m)| = \binom{2^m}{2^{m-2}+1}.$$

On the number of Boolean functions with m variables of nonlinearity $2^{m-2}+1$, we have Corollary 1.

Corollary 1. *For $m \geq 5$, the number of Boolean functions with m variables of nonlinearity $2^{m-2}+1$ is equal to $2^{m+1}|E^0_{2^{m-2}+1}(\mathrm{RM}_m)|$, which is*

$$2^{m+1}\left(\binom{2^m}{2^{m-2}+1} - 4(2^m-1)(2^{m-3}+1)\binom{2^{m-1}}{2^{m-2}+1} + (4^{m-2}+3)\binom{2^m}{3}\right).$$

The results of the calculation of $|E^0_{2^{m-2}+1}(\mathrm{RM}_m)|$ and $|E^1_{2^{m-2}+1}(\mathrm{RM}_m)|$ for $5 \leq m \leq 9$ are listed in Table 1. These expressions can be approximated by Stirling's approximation, $n! \approx \sqrt{2\pi n}(n/e)^n$. Thereby,

$$|E^0_{2^{m-2}+1}(\mathrm{RM}_m)| \approx \sqrt{\frac{3}{2^{m-3}\pi}}\left(\frac{16}{3\sqrt{3}}\right)^{2^{m-1}},$$

$$|E^1_{2^{m-2}+1}(\mathrm{RM}_m)| \approx \sqrt{\frac{2^m}{\pi}}(2^m+8)2^{2^{m-1}}.$$

The ratio, $|E^1_{2^{m-2}+1}(\mathrm{RM}_m)|/|E^0_{2^{m-2}+1}(\mathrm{RM}_m)|$, approaches zero as m increases.

Table 1. The number of correctable/uncorrectable errors of weight $2^{m-2}+1$ for RM_m

			correctable	uncorrectable
m	n	k	$\lvert E^0_{2^{m-2}+1}(\mathrm{RM}_m)\rvert$	$\lvert E^1_{2^{m-2}+1}(\mathrm{RM}_m)\rvert$
5	32	6	21,288,320	6,760,480
6	64	7	1.378×10^{15}	1.283×10^{12}
7	128	8	4.299×10^{30}	1.535×10^{22}
8	256	9	5.625×10^{61}	7.938×10^{41}
9	512	10	1.329×10^{124}	7.605×10^{80}

4 Conclusion

In this paper, we have determined the number of correctable/uncorrectable errors of weight half the minimum distance plus one for the first-order Reed-Muller codes. We mainly use the notion of larger halves to derive this result.

Future work includes deriving the number of correctable errors of weight $\geq 2^{m-2}+2$ for RM_m using the larger half technique and applying the technique to other codes, for example, the second-order Reed-Muller codes and BCH codes.

References

1. Peterson, W.W., Weldon Jr., E.J.: Error-Correcting Codes, 2nd edn. MIT Press, Cambridge (1972)
2. Zémor, G.: Threshold Effects in Codes. In: Cohen, G., Lobstein, A., Zémor, G., Litsyn, S.N. (eds.) Algebraic Coding. LNCS, vol. 781, pp. 278–286. Springer, Heidelberg (1994)
3. Helleseth, T., Kløve, T., Levenshtein, V.: Error-Correction Capability of Binary Linear Codes. IEEE Trans. Infom. Theory 51(4), 1408–1423 (2005)
4. Canteaut, A., Carlet, C., Charpin, P., Fontaine, C.: On Cryptographic Properties of the Cosets of $R(1, m)$. IEEE Trans. Inform. Theory 47(4), 1513–1949 (2001)
5. Carlet, C.: Boolean Functions for Cryptography and Error Correcting Codes. In: Crama, Y., Hammer, P. (eds.) Boolean Methods and Models, Cambridge University Press, Cambridge (press)
6. Berlekamp, E.R., Welch, L.R.: Weight Distributions of the Cosets of the (32,6) Reed-Muller Code. IEEE Trans. Inform. Theory 18(1), 203–207 (1972)
7. Wu, C.K.: On Distribution of Boolean Functions with Nonlinearity $\leq 2^{n-2}$: Australasian. Journal of Combinatorics 17, 51–59 (1998)

Fault-Tolerant Finite Field Computation in the Public Key Cryptosystems

Silvana Medoš and Serdar Boztaş

School of Mathematical and Geospatial Sciences,
RMIT University, GPO Box 2476V, Melbourne 3001, Australia
{silvana.medos, serdar.boztas}@ems.rmit.edu.au

Abstract. In this paper, we propose a new method for fault tolerant computation over $GF(2^k)$ for use in public key cryptosystems. In particular, we are concerned with the active side channel attacks, i.e., fault attacks. We define a larger ring in which new computation is performed with encoded elements while arithmetic structure is preserved. Computation is decomposed into parallel, mutually independent, identical channels, so that fault effects do not spread to the other channels. By assuming certain fault models, our proposed model provides protection against their error propagation. Also, we provide an analysis of the error detection and correction capabilities of our proposed model.

1 Introduction and Motivation

The arithmetic structure of finite fields is utilized in *public key cryptography* (smart card technology, e-commerce, and internet security), as well as in *coding theory* (error-free communications and data storage). Public key cryptographic applications rely on computation in very large finite fields (with more than 2^{160} elements for Elliptic Curve Cryptography, and more than 2^{1024} elements for RSA). Unfortunately, a single fault in computation can yield an erroneous output, which can then be used by an adversary to break cryptosystem–we describe the details below. Since we require high reliability and robustness, fault tolerant finite field computation in the public key cryptosystems is crucial. Security of cryptosystems does not only depend on the mathematical properties; an adversary can attack the implementation rather than algorithmic specification. In this paper we are concerned with protecting finite field computation against active side channel attacks, i.e., fault attacks, where an adversary induces faults into a device, while it executes the correct program. After outlining the fault model, and discussing a specific example of how fault-inducing attacks can be catastrophic for public key cryptosystems, we make use of well-known error correcting codes in order to provide countermeasures to some fault-inducing attacks.

1.1 Security of Public Key Cryptosystems

To quantify the security of public key cryptosystems, the concept of *computational security* is widely used. Here, an adversary is assumed to have limited

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 120–129, 2007.
© Springer-Verlag Berlin Heidelberg 2007

computation time and memory available (polynomial in the input parameters) and the security of the cryptosystem is based on the fact that the problem of breaking the system is reducible to solving a problem that is strongly believed to be computationally hard, such as factoring a product of two large random primes and taking discrete logarithms in a large finite field. As a rule, one assumes that an adversary always has access to all data being transmitted by two communicating parties and exact knowledge of every aspect of the used cryptographic scheme, except for the secret key–this is referred to as the *Kerchoffs' Principle*. Moreover an adversary can request encryptions of polynomially many (in the size of the input parameters) chosen messages to achieve his objective. This scenario is usually referred to as a *black-box assumption*, since it allows purely theoretical proofs on paper. However, cryptosystems are used in the real world where cryptographic protocols are implemented in software or hardware, obeying laws of physics. The circuits used *leak* information, e.g., power and timing information, over side channels. Thus, one has a *gray box*, where an adversary has access to several side-channels.

2 Fault Models and Fault Attacks

2.1 Fault Models

Since smartcards are in exposed conditions and receive their power and clock signals from the smartcard reader, they can be subjected to physical attacks by using X-rays, heat/infrared radiation, power spikes, optical energy, etc. We refer the reader to the CHES proceedings (in the Springer LNCS series) for many papers illustrating such attacks. Our discussion of the fault models is broadly based on [9]. The crucial point is how to model the faults which result from such physical attacks. As a list of possibilities include:

- The attacker's control of the *location of the fault* (hardware memory cell, or software instruction) can be strong, weak, or nonexistent.
- The attacker's control of the *timing of the fault* can be precise, within an interval, or nonexistent.
- The *type of fault* can be, e.g., *random*, *bit-flip*, or *stuck-at*.
- The *fault duration* may be *permanent* or *transient*.

We now discuss some of these in more detail. Note that any fault induced in a variable x can be described by means of an additive error term $x \mapsto x' = x + e(x)$ but the error term $e(x)$ can itself take on quite different characteristics, depending on the type of the fault:

Stuck-at Faults. Let b be an arbitrary bit stored in memory. Assume that b is modified by a stuck-at fault. Then $b \mapsto b' = c$, where the constant $c = 1$ or $c = 0$. The value of the affected bit is not changed any more, even if a variable x, which uses these bits, is overwritten. Clearly stuck-at faults will have a noticeable effect only if the variable is overwritten at some point.

Bitflip Faults. Let b be an arbitrary bit stored in memory. Assume that b is modified by a bitflip fault. Then $b \mapsto b' = b + 1 \pmod 2$. The effect may be transient, permanent or destructive. A bitflip fault is easy to visualize, and always results in a fault on a variable using the bit which is faulty.

Random Faults. Let b be an arbitrary bit stored in memory. Assume that b is modified by a random fault. Then $b \mapsto b'$ where b' is a random variable taking on the values 0 or 1. The effect may be transient, permanent or destructive. Since several physical methods of fault induction are difficult to control precisely, random faults are considered to be the most realistic fault type. The random variable which models the fault may be uniform or non-uniform.

Note that the above faults can be considered for an arbitrary but unknown set of bits B, where assumptions about how the adversary controls the choice of B can also model different attack scenarios, but we do not proceed further in this direction due to space constraints.

2.2 An Attack on ElGamal Signatures

To generate a public/private key pair in this scheme, one first chooses a prime p and two integers g, x such that both are in $\mathbf{Z}_p$, and g is a generator of $\mathbf{Z}_p^*$. The private key is x while the public key is (y, g, p) where $y = g^x \pmod p$. The following attack is from [1].

To generate a signature on a message m the signer picks a random k with $gcd(k, p-1) = 1$, and computes

$$w = g^k \pmod p \qquad \text{and} \qquad s = (m - xw)/k \pmod{(p-1)}.$$

The signature is the pair (w, s) and to verify the signature, the verifier confirms that

$$y^w w^s = g^m \pmod p.$$

If a fault leads to a change in bit x_i of x by a bit-flip during the process of signing a message, a corrupted version x' of x will result and we will have the outputs.

$$w = g^k \pmod p \qquad \text{and} \qquad s' = (m - x'w)/k \pmod{(p-1)}.$$

Using w, s', m and the signer's public key (y, p, g) the attacker can now compute

$$T = y^w w^{s'} \pmod p = g^m g^{w(x-x')} \pmod p.$$

Let $R_i = g^{w2^i} \pmod p$ for $i = 0, 1, \ldots, t-1$ where t is the bitlength of x. Then

$$TR_i = g^m \pmod p, \qquad \text{if } x_i = 0,$$

(since $x_i = 0$ implies that $x - x' = -2^i$) and

$$TR_i^{-1} = g^m \pmod p, \qquad \text{if } x_i = 1,$$

(since $x_i = 1$ implies that $x - x' = 2^i$). The attacker can compute (TR_i, TR_i^{-1}) for $i = 0, 1, \ldots, t-1$ and checks to see if either equals $g^m \pmod p$ and if a match is found then one bit of x is found.

By using the fault tolerant computation technique–applied to multiplying field elements x', w defined above–discussed later on in this paper, it is possible to detect the fault which leads to the attack described above, which could be used to alert a higher level monitoring system that something has gone wrong in the signature computation.

In related work, [6] presents fault-tolerant computation over the integers based on the modulus replication residue number system, which allows modular arithmetic computations over identical channels. [11] presents multipliers for fields $GF(2^k)$ whose operations are resistant to errors caused by certain faults. They can correct single errors caused by one, or more faults in the multiplier circuits. [5] introduces scaled embedding for Fault-Tolerant Public Key Cryptography based on arithmetic codes and binary cyclic codes in order to achieve robust fault tolerant arithmetic in the finite field.

3 Finite Field Encoding

We want to protect computation over the field $GF(2^k)$, which can be represented as the set of polynomials modulo a primitive polynomial $f(x)$, $deg(f(x)) = k$, i.e., $GF(2)[x]/ < f(x) >= \{a_0 + \ldots + a_{k-1}x^{k-1} | a_i \in GF(2)\}$, and where $f(\alpha) = 0$, so that $GF(2^k) = \{0, 1, \alpha, \alpha^2, \ldots, \alpha^{2^k-2}\}$. The inputs to the computation are elements from the field $GF(2^k)$ represented as a polynomials. The input polynomials g_i from $GF(2)[x]/ < f(x) >$ are evaluated at the minimum required number of distinct elements from the set $T = \{\alpha_j | \alpha_j \in GF(2^k)\}$ such that there are enough values to represent the polynomial resulting from the computation. Evaluating input polynomials $g_i \in GF(2)[x]/ < f(x) >$ at distinct elements $\alpha_j \in T$ is same as taking remainder modulo $x - \alpha_j$. Let n be the expected degree of the output, which is not reduced modulo $f(x)$. Then, there exists a mapping ϕ

$$\phi : GF(2)[x]/ < f(x) > \rightarrow GF(2^k)[x]/ < x-\alpha_0 > \times \ldots \times GF(2^k)[x]/ < x-\alpha_n >,$$

such that each input polynomial $g_i(x) \in GF(2)[x]/ < f(x) >$ is evaluated at $n+1$ distinct elements from the set $T = \{\alpha_j | \alpha_j \in GF(2^k)\}$, i.e.,

$$g_i(x) \leftrightarrow (g_i(\alpha_0), g_i(\alpha_1), \ldots, g_i(\alpha_n)), \tag{1}$$

where, $g_i(\alpha_j) \in GF(2^k)$ (or equivalently $g_i(\alpha_j) \in GF(2)^k$) are evaluations of the input polynomials $g_i \in GF(2)[x]/ < f(x) >$ at distinct elements from the set T. Equivalently, $g_i(\alpha_j)$ is remainder of $g_i(x)$ on division by linear polynomial $(x - \alpha_j)$, i.e., $g_i(x) \equiv g_i(\alpha_j) \, mod \, (x - \alpha_j)$.

3.1 Computation in the Larger Ring

The computation of the finite field $GF(2^k)$ will be performed with encoded operands (as in (1)) in the direct product ring:

$$R = GF(2^k)[x]/ < x - \alpha_0 > \times \ldots \times GF(2^k)[x]/ < x - \alpha_n > \cong GF(2^k)^{n+1}, \quad (2)$$

while preserving arithmetic structure. Note that $R \cong GF(2^k)[x]/ < m(x) >$, where $m(x) = \prod_{i=0}^{n}(x-\alpha_i)$, such that $deg\,(m\,(x)) = 1+max\,\{deg\,(g(x) * h(x))\}$, where $g(x), h(x) \in GF(2)[x]/ < f(x) >$ are input polynomials, and $*$ is an operation (addition or multiplication) in the $GF(2^k)$ without modulo $f(x)$ reduction. By the well-known Lagrange Interpolation Theorem (LIT), interpolating $n+1$ output components $r(\alpha_j) \in GF(2^k)$ at distinct elements $\alpha_j \in GF(2^k)$ will determine a unique polynomial $r(x) \in GF(2^k)[x]/ < m(x) >$ of degree n.

4 Fault-Tolerant Computation

To protect computation in the finite field we add redundancy by adding more parallel channels than the minimum required to represent the output polynomial of a certain expected degree, i.e., see Figure 1. Thus, input polynomials are evaluated at additional distinct elements $\alpha_j \in GF(2^k)$. Let n be expected degree of the output polynomial without modulo $f(x)$ reduction. We use a total of $c > n+1$ evaluations so that computation now happens in the even larger direct product ring

$$R' = GF(2^k)[x]/ < x - \alpha_0 > \times \ldots \times GF(2^k)[x]/ < x - \alpha_{c-1} > \cong GF(2^k)^c.$$

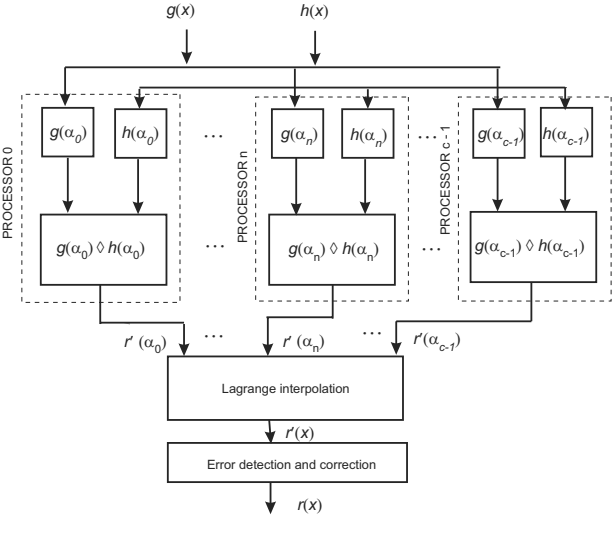


Fig. 1. Fault tolerant computation of the finite field $GF(2^k)$ in the ring R'

As before, let each input polynomial $g_i(x)$ be evaluated at $c > n+1$ distinct elements from the set $T = \{\alpha_j | \alpha_j \in GF(2^k)\}$, i.e.,

$$g_i(x) \rightarrow (g_i(\alpha_0), g_i(\alpha_1), \ldots, g_i(\alpha_n), g_i(\alpha_{n+1}), g_i(\alpha_{n+2}), \ldots, g_i(\alpha_{c-1})) \in R',$$

where $g_i(\alpha_0), g_i(\alpha_1), \ldots, g_i(\alpha_n)$ are *non-redundant components* of i-th input polynomial, and $g_i(\alpha_{n+1}), g_i(\alpha_{n+2}), \ldots, g_i(\alpha_{c-1})$ are *redundant components* of i-th input polynomial. Now, let $r' \in R'$ be an output vector of the computation which is in the form

$$r' = (r(\alpha_0), r(\alpha_1), \ldots, r(\alpha_n), r(\alpha_{n+1}), r(\alpha_{n+2}), \ldots, r(\alpha_{c-1})). \tag{3}$$

By the uniqueness of LIT, if there are no fault effects, the c output components $r(\alpha_j) \in GF(2^k)$ at distinct elements will determine a unique polynomial $r'(x) \in GF(2^k)[x]/ < m'(x) >$ of degree n with coefficients $a_i \in GF(2)$, otherwise, $n < deg(r'(x))$ with coefficients $a_i \in GF(2^k)$. This leads to:

Definition 1. *The set of correct results of computation, where n is expected degree of output polynomial of the computation without modulo $f(x)$ reduction, is*

$$C = \left\{ r'(x) \in GF\left(2^k\right)[x] / < m'(x) > \mid deg\left(r'(x)\right) < n+1,\, a_i \in GF(2) \right\}.$$

4.1 Complexity of Interpolation and Evaluation

Input polynomials are only evaluated at the beginning, while interpolation is performed at the end of the computation. We do modulo $f(x)$ reduction only if there are no errors.

Lemma 1. *Computational complexity of evaluating input polynomials $g_i \in GF(2)[x]/ < f(x) >$ at $c > n+1$ distinct elements from set T, where n is expected degree of the output polynomial without modulo $f(x)$ reduction is $O(ck)$, since the required number of operations in $GF(2^k)$ is $2c(k-1)$.*

Proof. Let $g_i(x) = \sum_{i=0}^{k-1} a_i x^i \in GF(2^k)$, and use Horner's rule

$$g_i(x) = (\ldots(a_{k-1}x + a_{k-2})x + \ldots + a_1)x + a_0.$$

Thus, g_i can be evaluated at a single point $\alpha_i \in T$ by $k-1$ addition and $k-1$ multiplications. Therefore, evaluating $g_i(x)$ at $c > n+1$ distinct elements from T it will require $2c(k-1)$ operations in $GF(2^k)$. So the computational complexity of input polynomial evaluation is $O(ck)$. □

Lemma 2. *Computational complexity of interpolating output vector $r' \in R'$ is $O\left(c^2\right)$, $c > n+1$.*

Proof. For a proof see, e.g., [4]. □

Theorem 1. *Total computational complexity of evaluating input polynomials $g_i(x) \in GF(2)[x]/ < f(x) >$ at the beginning of computation, and interpolation of the result of the computation at the end of computation is $O(c^2)$.*

Proof. Since the computational complexity of evaluating inputs g_i is $O(ck)$, where $k < c$, and complexity of interpolating the resulting vector is $O(c^2)$, then total complexity is $O(c^2)$. □

5 Error Detection and Correction

There is one processor per independent channel, i.e., see Figure 1. Let us assume that we have c processors, where processor i computes i-th polynomial evaluation and all processors perform operations over the finite field $GF(2^k)$.

We define a fault attack as any method and/or algorithm which when applied to the attacked processor return desired effects. We assume that a fault attack induces faults into processors by some physical set up, exposing the processor to a physical stress (x-rays, heat/infrared radiation, power spikes, clock glitches, etc.) An adversary can run the attack several times while inducing faults into structural elements of an attacked processor, till the desired effect occur. As a reaction, the attacked processor malfunctions, i.e., memory cells change their voltage, bus lines transmit different signals, or structural elements are damaged. The processor is now faulty, i.e., it does not compute the correct output given its input. We identify memory cells with their values, and we say that faults are induced into variables, or bits.

We are concerned with the effect of a fault as it manifests itself in a modified data, or a modified program execution. Therefore, we consider the following fault models (inspired by [9], see also Section 2 for more general background on modeling):

Random Fault Model (RFM) 2. *Assume that an adversary does not know much about his induced faults to know its effect, but he knows the affected polynomial evaluation. Therefore, we assume that affected polynomial evaluation $f(\alpha_i) \in GF(2^k)$ is changed to some random value from the finite field $GF(2^k)$, assumed to be uniformly distributed in that field.*

Arbitrary Fault Model (AFM) 3. *Assume that an adversary can target specific line of code, targeting specific channel, but without knowing the effects of the fault. This is modelled as the addition of an arbitrary and unknown element e_i to r_i.*

Since, computation is decomposed into parallel, mutually independent channels, adversary can use either RFM, or AFM on each channel. Assume that at most $c-n-1$ channels have faults. Let $r' \in R'$ be computed vector with c components as in (3), where $e_j \in GF(2^k)$ is the error at j -th position; then the computed component at the j-th positions is

$$r_j = r(\alpha_j) + e_j, \tag{4}$$

and each processor will have as an output component

$$r_j = \begin{cases} r(\alpha_j) + e_j, \, j \in \{j_1, \ldots, j_t\}, \\ r(\alpha_j), \qquad \text{else.} \end{cases}$$

Here, we have assumed that the set of error positions are $\{j_1, \ldots, j_t\}$, i.e., e_j is the effect of the fault in the channel j_i. By LIT, the computed vector $r' \in R'$

with corresponding set of c distinct elements $\alpha_j \in GF(2^k)$ gives as a output unique polynomial $r'(x) \in GF(2^k)[x]/ < m'(x) >$,

$$r'(x) = \sum_{0\le i\le c-1} r_i \prod_{0\le j\le c-1,\, i\neq j} \frac{x-\alpha_j}{\alpha_i-\alpha_j}$$
$$= r(x) + \sum_{1\le l\le t} e_{j_l} \prod_{0\le i\le c-1,\, j_l\neq i} \frac{x-\alpha_i}{\alpha_{j_l}-\alpha_i} = r(x) + e(x), \tag{5}$$

where $r(x)$ is correct expected polynomial of degree $\le n$ with coefficients from the ground field $GF(2)$, and $e(x)$ is the error polynomial which obeys the following:

Theorem 4. *Let effects of the fault $e_{j_1} \neq 0, \ldots, e_{j_t} \neq 0$ be any set of $1 \le t \le c-n-1$ elements of $GF(2^k)$, $c > n+1$, then $deg(e(x)) > n$ whose coefficients $a_i \in GF(2^k)$.*

Proof. We have that

$$e(x) = \sum_{1\le l\le t} e_{j_l} \prod_{0\le i\le c-1, j_l\neq i} \frac{x-\alpha_i}{\alpha_{j_l}-\alpha_i}$$
$$= \prod_{0\le i\le c-1} (x-\alpha_i) \left(\frac{e_{j_1}}{(x-\alpha_{j_1}) \prod_{0\le i\le c-1, j_1\neq i} (\alpha_{j_1}-\alpha_i)} + \ldots \right.$$
$$\left. \ldots + \frac{e_{j_t}}{(x-\alpha_{j_1}) \prod_{0\le i\le c-1,\, j_1\neq i} (\alpha_{j_t}-\alpha_{c-1})} \right)$$

Since, $deg\left(\frac{\prod_{0<i\le c-1}(x-\alpha_i)}{(x-\alpha_{j_1})}\right) = c-1, \ldots, deg\left(\frac{\prod_{0<i\le c-1}(x-\alpha_i)}{(x-\alpha_{j_t})}\right) = c-1, c > n+1$ then $deg\,(e(x)) = c-1 > n$ with coefficients $\frac{e_{j_k}}{(x-\alpha_{j_1}) \prod_{0\le i\le c-1\, j_1\neq i}(\alpha_{j_k}-\alpha_i)}$ in $GF(2^k)$. □

Therefore, faulty processors affect the result in an additive manner. From here on it is straightforward to appeal to standard coding theory results to show that:

Theorem 5. *(i) If the number of parallel, mutually independent, identical redundant channels is $d+t \le c-n-1$ $(d \ge t)$, then up to t faulty processors can be corrected, and up to d simultaneously detected. (ii) By adding $2t$ redundant independent channels at most t faulty processors can be corrected.*

While it is true that arbitrarily powerful adversaries can simply create faults in enough channels and overwhelm the system proposed here, it is part of the design process to decide on how much security is enough, since all security (i.e. extra channels) has a cost. We also remark that the Welch-Berlekamp algorithm is suitable for correcting the faults induced by the attacks described in this paper. Note that to specify the algorithm we choose a set of $n+1$ indices $K = \{0, 1, \ldots, n\}$, and $\overline{K} = \{0, \ldots, c-1\} \setminus K$.

Algorithm 1. Welch-Berlekamp Decoding of the Output Vector.

Inputs: output vector of computation $r' = (r_0, \ldots, r_n, r_{n+1}, \ldots, r_c)$, set of c distinct points $T = \{\alpha_j | \alpha_j \in GF(2^k)\}$, set of indices $K = \{0, 1, \ldots, n\}$, polynomial $g(x) = \prod_{i \in K} (x - x_i)$
Outputs: polynomials $d(x)$, $r'(x)$.

1. By *Lagrange interpolation*, interpolate output vector r' in order to get polynomial $r'(x)$, if $deg(r'(x)) \leq n$ and $a_i \in GF(2)$ then STOP, else
2. for $i \in K$ find $r'(x)$, where $deg(r') \leq n$,
3. evaluate $r'(x)$, at α_l, $l \in \overline{K}$,
4. determine syndromes $S_l = r_l - r'(x_l)$, $l \in \overline{K}$,
5. determine $y_l = \frac{S_l}{g(x_l)}$,
6. solve key equation $d(x_l)y_l = r'(x_l)$.

6 Conclusions and Current Work

We have described fault attacks on cryptosystems and proposed a means of protecting computation of the finite field $GF(2^k)$ against side-channel attacks, by decomposing computation over parallel, independent, identical channels. This offers a great advantage, since computations are mutually independent (fault effects do not spread to the other channels), and they are performed over the same field. Fault-tolerant computation is obtained by the use of redundancy. By adding $d + t$, $d \geq t$ redundant channels we can correct up to t faulty processors, and simultaneously detect d faulty processors. Either of two proposed fault models, *RFM*, or *AFM* can be used on each channel. Our method covers random and burst errors that can be caused by malicious fault insertion by an adversary, or transient faults. Also, efficient error correction is possible through the use of *Welch-Berlekamp decoding algorithm*. Moreover, it is part of the design process to decide on how much security is enough, since all security (i.e. extra channels) has a cost.

In current work, we are directly applying the method developed in this paper to the algorithm specific computations which are used in elliptic and hyperelliptic curve cryptosystems. Since the group addition in such cryptosystems is built up of a specific sequence of finite field additions and multiplications–to which the results of this paper directly apply–this is a natural progression in our research.

Acknowledgment

The authors would like to thank the Australian Research Council for its support through the ARC Linkage grant, LP0455324. The authors would also like to thank the anonymous referees whose comments vastly improved the presentation and content of the paper.

References

1. Bao, F., Deng, R.H., Han, Y., Jeng, A.B., Narasimhalu, A.D., Ngair, T-H.: Breaking Public Key Cryptosystems on Tamper Resistant Devices in the Presence of Transient Faults. In: Christianson, B., Lomas, M. (eds.) Security Protocols. LNCS, vol. 1361, pp. 115–124. Springer, Heidelberg (1998)
2. Beckmann, P.E., Musicus, B.R.: Fast Fault-Tolerant Digital Convolution Using a Polynomial Residue Number System. IEEE Trans. Signal Processing 41(7), 2300–2313 (1993)
3. Boneh, D., DeMilo, R.A., Lipton, R.J.: On the Importance of Eliminating Errors in Cryotographic Computations. J. Cryptology 14, 101–119 (2001)
4. Gathen, J., Gerhard, J.: Modern Computer Algebra. Cambridge University Press, UK (1999)
5. Gaubatz, G., Sunar, B.: Robust Finite Field Arithmetic for Fault-Tolerant Public-Key Cryptography. In: 2005 Workshop on Fault Diagnosis and Tolerance in Cryptography, Edinburgh, Scotland (2005)
6. Imbert, L., Dimitrov, L.S., Jullien, G.A.: Fault-Tolerant Computation Over Replicated Finite Rings. IEEE Trans. Circuits Systems-I: Fundamental Theory and Applications 50(7), 858–864 (2003)
7. Kocher, P.C., Jaffe, J., Jun, B.: Differential Power Analysis. In: Wiener, M.J. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999)
8. Lidl, R., Niederreiter, H.: Introduction to Finite Fields and Their Applications. Cambridge University Press, London (1986)
9. Otto, M.: Fault Attacks and Countermeasures. PhD Thesis (2004)
10. Reed, I.S., Solomon, G.: Polynomial Codes over Certain Finite Fields. J. Society for Industrial and Applied Mathematics 8(2), 300–304 (1960)
11. Reyhani-Masoleh, A., Hasan, M.A.: Towards Fault-Tolerant Cryptographic Computations over Finite Fields. ACM Trans. Embedded Computing Systems 3(3), 593–613 (2004)
12. Welch, L., Berlekamp, E.R.: Error Corrections for Algebraic Block Codes. U.S. Patent 4 633 470 (1983)
13. Wicker, S.B., Bhargava, V.K.: Reed-Solomon Codes and Their Applications. IEEE Press, New York (1994)

A Note on a Class of Quadratic Permutations over $\mathbb{F}_{2^n}$

Yann Laigle-Chapuy

INRIA, Domaine de Voluceau, BP 105,
78153 Rocquencourt, Le Chesnay Cedex, France
yann.laigle-chapuy@inria.fr

Abstract. Finding new classes of permutation polynomials is a challenging problem. Blockhuis at al. investigated the permutation behavior of polynomials of the form $\sum_{i=0}^{n-1} a_i X^{2^i+1}$ over $\mathbb{F}_{2^n}$. In this paper, we extend their results and propose as a new conjecture that if $n = 2^e$ then X^2 is the only unitary permutation polynomial of this type.

1 Introduction

Let $\mathbb{F}_{2^n}$ be the field of order 2^n and $\mathbb{F}_{2^n}[X]$ denote the ring of polynomials in the indeterminate X with coefficients in $\mathbb{F}_{2^n}$. A polynomial $P \in \mathbb{F}_{2^n}[X]$ which permutes $\mathbb{F}_{2^n}$ under evaluation is called a *permutation polynomial* over $\mathbb{F}_{2^n}$. For a general introduction to permutation polynomials, we refer to [1,2]. Discovering new classes of permutation polynomials is an old problem with applications in cryptography, coding theory and in combinatorial designs. For instance, Patarin introduced the HFE cryptosystem [3] based on *quadratic* polynomials, which are polynomials of the form

$$\sum_{0 \le i,j \le n-1} a_{i,j} X^{2^i+2^j}, \quad a_{i,j} \in \mathbb{F}_{2^n}.$$

In his paper, he raised the problem of finding quadratic *permutation* polynomials and stated that it seems to be difficult to characterize them. Only few families of such quadratic permutation polynomials are known. We can cite for example Dobbertin's permutation [4] over $\mathbb{F}_{2^n}$

$$X^{2^{m+1}+1} + X^3 + X, \quad \text{with } n = 2m+1.$$

Also families of binomials have been found recently by Budaghyan et al. [5]. Quadratic polynomials restricted to j equal to 0,

$$\sum_{0 \le i \le n-1} a_i X^{2^i+1}, \quad a_i \in \mathbb{F}_{2^n},$$

is an interesting subclass that has been introduced by Blockhuis et al. in [6] where they studied their permutation behavior.

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 130–137, 2007.
© Springer-Verlag Berlin Heidelberg 2007

The purpose of this paper is to extend their results. In Section 2 we state the definitions and notations. We will then define in Section 3 a new class of bilinear permutation polynomials which will lead us to a new conjecture. Finally, in Section 4, we discuss some related problems and give arguments supporting our conjecture.

2 Preliminaries

For most cryptographic purposes, compositions with a linear application will not change the properties of the function. We will therefore define *linear equivalence*. First, let us recall the shape of polynomials corresponding to linear applications, that is *linearized polynomials*.

Definition 1. *A* linearized *polynomial* $P \in \mathbb{F}_{2^n}[X]$ *is a polynomial of the shape*

$$P(X) = \sum_{i=0}^{n-1} a_i X^{2^i} \text{ with } a_i \in \mathbb{F}_{2^n}.$$

Let $\mathcal{L}(n)$ *be the set of all such polynomials.*

We will consider equivalence classes under the action of bijective linearized polynomials.

Definition 2. *Two polynomials* P *and* Q *in* $\mathbb{F}_{2^n}[X]$ *are* linearly equivalent *if there exists* L_1 *and* L_2 *linearized* permutation *polynomials in* $\mathcal{L}(n)$ *such that*

$$L_1 \circ P \circ L_2 = Q.$$

We will now present the class of polynomials which we will study. According to the work of Blockhuis et al., we will focus on a subclass of quadratic polynomials.

Definition 3. *A* bilinear *polynomial* $P \in \mathbb{F}_{2^n}[X]$ *is a polynomial of the shape*

$$P(X) = L_1(X)L_2(X) \text{ with } L_1, L_2 \in \mathcal{L}(n).$$

Moreover, as we are only interested in equivalence classes, we can extract an even smaller family.

Proposition 1 (cf. [6]). *Every* $P \in \mathbb{F}_{2^n}[X]$ *bilinear permutation is linearly equivalent to a polynomial* $XL(X)$ *with* $L \in \mathcal{L}(n)$.

Proof. To be a permutation, $P(X) = L_1(X)L_2(X)$ must have only 0 as a root. Therefore L_1 and L_2 , as they are linearized polynomials, must be permutations and thus invertibles.

Composing P with L_1^{-1} which is also *linearized*, we obtain an equivalent polynomial $P \circ L_1^{-1}(X) = XL(X)$ with $L = L_2 \circ L_1^{-1}$. □

This allows us to restrict ourselves to study only the permutation behavior of polynomials of the shape

$$XL(X) = \sum_{0 \leq i \leq n-1} a_i X^{2^i+1} \text{ with } a_i \in \mathbb{F}_{2^n}.$$

In the following, $\mathcal{B}(n)$ will denote the set of such polynomials.

Notice that all the terms of those polynomials are quadratic, except possibly a term of linear degree 2.

3 Permutations Amongst $\mathcal{B}(n)$

At the time being, only few permutations amongst $\mathcal{B}(n)$ are known. The only study of this class of polynomials is found in the article of Blokhuis et al. [6] and we will recall their results.

Theorem 1 (cf. [6]). *Let k and n be any integer and set $d = Gcd(n, k)$. The following three classes define permutations amongst $\mathcal{B}(n)$:*

(i) X^{2^k+1} *where n/d is odd.*
(ii) $X^{2^k+1} + aX^{2^{n-k}+1}$ *where n/d is odd and $a^{(2^n-1)/(2^d-1)} \neq 1$.*
(iii) $X^{2^{2k}+1} + (aX)^{2^k+1} + aX^2$ *where $n = 3k$ and $a^{(2^n-1)/(2^k-1)} \neq 1$.*

Moreover, (ii) and (iii) are linearly equivalent to (i).

All those classes are linearly equivalent to a monomial and the proof gives explicitly L_1 and L_2 such that

$$L_1 \circ XL(X) \circ L_2 = X^{2^k+1} \mod X^{2^n} + X.$$

Blokhuis et al. also introduced a last family of bilinear permutation using the trace function.

Definition 4. *Recall the notation for the field trace:*

$$\mathrm{Tr}_\ell^k(X) = \mathrm{Tr}_{\mathbb{F}_{2^{k\ell}}/\mathbb{F}_{2^\ell}}(X) = \sum_{i=0}^{k-1} X^{2^{i\ell}}$$

Theorem 2. *Let k be odd and ℓ be any positive integer. Set $n = k\ell$ and $a \in \mathbb{F}_{2^\ell} \setminus \mathbb{F}_2$. Then the following polynomial is a bilinear permutation over $\mathbb{F}_{2^n}$.*

(iv) $X\left(\mathrm{Tr}_\ell^k(X) + aX\right)$.

We will now give an extension to their results, constructing recursively new bilinear permutation polynomials.

Theorem 3 (a new class). *Let k be odd and ℓ be any positive integer. Set $n = k\ell$, $a \in \mathbb{F}_{2^\ell}$ a non zero element of the subfield and $L \in \mathcal{L}(\ell)$ a linearized polynomial over $\mathbb{F}_{2^\ell}$ such that $XL(X) \in \mathcal{B}(\ell)$ is a bilinear permutation over $\mathbb{F}_{2^\ell}$. Then the following polynomial is a bilinear permutation over $\mathbb{F}_{2^n}$.*

(v) $X\left(L(\mathrm{Tr}_\ell^k(X)) + a\mathrm{Tr}_\ell^k(X) + aX\right)$.

Proof. The case (iv) is deduced from (v), with $L(X) = X$, by applying the following transformation:

$$X\left(\mathrm{Tr}_\ell^k(X) + aX\right) = (a+1)X\left(\mathrm{Tr}_\ell^k(X) + \frac{a}{a+1}\mathrm{Tr}_\ell^k(X) + \frac{a}{a+1}X\right).$$

Let's now prove (v).
Let $P(X) = X\left(L(\mathrm{Tr}_\ell^k(X)) + a\mathrm{Tr}_\ell^k(X) + aX)\right) \in \mathbb{F}_{2^n}[X]$ with $Q(X) = XL(X)$ in $\mathcal{B}(\ell)$ a permutation over $\mathbb{F}_{2^\ell}$. For all $x \in \mathbb{F}_{2^n}$,

$$\mathrm{Tr}_\ell^k(P(x)) = \mathrm{Tr}_\ell^k(x)\mathrm{Tr}_\ell^k(L(\mathrm{Tr}_\ell^k(x))).$$

Moreover, as k is odd, $L(\mathrm{Tr}_\ell^k(x))$ which lies in $\mathbb{F}_{2^\ell}$ is equal to its trace. We thus obtain

$$\mathrm{Tr}_\ell^k(P(x)) = Q(\mathrm{Tr}_\ell^k(x)).$$

Let x and y be such that $P(x) = P(y)$. We have in particular

$$Q(\mathrm{Tr}_\ell^k(x)) = Q(\mathrm{Tr}_\ell^k(y)),$$

and since Q permutes $\mathbb{F}_{2^\ell}$,

$$\mathrm{Tr}_\ell^k(x) = \mathrm{Tr}_\ell^k(y).$$

Let t denote this trace.

$$\begin{aligned} P(x) = P(y) &\Leftrightarrow x(L(t) + at + ax) = y(L(t) + at + ay) \\ &\Leftrightarrow a(x+y)(a^{-1}L(t) + t + x + y) = 0 \end{aligned}$$

This implies that $x + y = a^{-1}L(t) + t$ or $x + y = 0$ which in both cases gives $x + y \in \mathbb{F}_{2^\ell}$.

Finally, applying the trace operator, $x+y = \mathrm{Tr}_\ell^k(x+y) = t+t = 0$ and $P(X)$ is a permutation polynomial, which conclude the proof.

In order to clarify these results, we will give a few examples.

Example 1. In $\mathbb{F}_{2^9}$, taking α as a primitive element, we obtain two classes of bilinear permutations non linearly equivalent neither to monomials nor to each other, one of type (iv) and one of type (v).

$$\begin{array}{lrcl} \text{type } (iv) & X^{65} + X^9 + \alpha^{73}X^2 & = & X\left(\mathrm{Tr}_3^3(X) + \alpha^{219}X\right) \\ \text{type } (v) & X^{129} + X^{65} + X^{17} + X^9 + X^3 & = & X\left(\mathrm{Tr}_3^3(X)^2 + \mathrm{Tr}_3^3(X) + X\right) \end{array}$$

Example 2. Starting from the type (iv) permutation polynomial in $\mathcal{B}(6)$

$$P_6(X) = X^{17} + X^5 + aX^2$$

where $a \in \mathbb{F}_4 \setminus \mathbb{F}_2$ and taking $b \in \mathbb{F}_{2^{15}}^\star$ non zero, we construct the following permutation of $\mathbb{F}_{2^{30}}$

$$\begin{aligned} P_{30}(X) &= X\left(\frac{P_6(X)}{X} \circ \mathrm{Tr}_6^5(X) + b\mathrm{Tr}_6^5(X) + bX\right) \\ &= X^{2^{28}+1} + X^{2^{26}+1} + X^{2^{22}+1} + X^{2^{20}+1} + X^{2^{16}+1} + X^{2^{14}+1} + X^{2^{10}+1} + X^{2^8+1} \\ &\quad + X^{2^4+1} + X^{2^2+1} + (a+b)\left(X^{2^{24}+1} + X^{2^{18}+1} + X^{2^{12}+1} + X^{2^6+1}\right) + aX^2. \end{aligned}$$

Example 3. In $\mathbb{F}_{2^{15}}$, the following polynomials are type (v) permutations.

$X^{2049} + aX^{1025} + X^{65} + aX^{33} + X^3$	$a \in \mathbb{F}_{32}^\star$
$X^{4097} + aX^{1025} + X^{129} + aX^{33} + X^5$	$a \in \mathbb{F}_{32}^\star$
$X^{8193} + aX^{1025} + X^{257} + aX^{33} + X^9$	$a \in \mathbb{F}_{32}^\star$
$X^{8193} + aX^{4097} + X^{1025} + aX^{513} + X^{129} + aX^{65} + X^{17} + aX^9 + X^3$	$a \in \mathbb{F}_8^\star$
$X^{16385} + aX^{1025} + X^{513} + aX^{33} + X^{17}$	$a \in \mathbb{F}_{32}^\star$
$X^{16385} + aX^{4097} + X^{2049} + aX^{513} + X^{257} + aX^{65} + X^{33} + aX^9 + X^5$	$a \in \mathbb{F}_8^\star$

Conjecture 1. The class (v) contains an infinite class of permutation polynomials non linearly equivalent to monomials.

We can see that all the families (i) to (v) need the degree of the extension n to have an odd factor. We also verified with an exhaustive search for $n \leq 7$ that, up to linear equivalence, there are no other bilinear permutations. This leads us to the two following conjectures.

Conjecture 2. Their is no other bilinear permutation than aX^2 in $\mathcal{B}(n)$ with $n = 2^e$.

Conjecture 3. Their is no non monomial bilinear permutation in $\mathcal{B}(p)$ with p prime.

Moreover, we will give in the following section a result in the direction of Conjecture 2.

4 Discussions

4.1 On Linearized Permutation

We would like to emphasize the role of linear permutations. They appear twice in our context. First, we use them to define linear equivalences, as cryptographic properties are mainly invariant under their action. Secondly, as stated in Proposition 1, every bilinear permutation comes from a linearized permutation polynomial. We can therefore deduce from Theorem 3 a class of linearized permutation.

Corollary 1 (new linearized permutations). *Let k be odd and ℓ be any positive integer. Set $n = k\ell$, $a \in \mathbb{F}_{2^\ell}$ a non zero element of the subfield and $L \in \mathcal{L}(\ell)$ a linearized polynomial over $\mathbb{F}_{2^\ell}$ such that $XL(X) \in \mathcal{B}(\ell)$ is a bilinear*

permutation over $\mathbb{F}_{2^\ell}$. Then the following polynomial is a linearized permutation polynomial over $\mathbb{F}_{2^n}$

$$L(\mathrm{Tr}_\ell^k(X)) + a\mathrm{Tr}_\ell^k(X) + aX.$$

It is also interesting to consider our problem as characterizing the permutation behavior of modified linearized permutations.

Our main result treats $XL(X)$. We will now look at $L(X)/X$ and $L(X)+aX$.

The first result is a theorem from Payne [7,8]. Originally dealing with ovoids in Desarguian planes, we can restate it as follows.

Theorem 4 (Payne). *Let $L(X) \in \mathcal{L}(n)$ be a linearized polynomial. Then*

$$P(X) = \frac{L(X)}{X} = \sum_{i=0}^{n} a_i X^{2^i-1}, \quad a_i \in \mathbb{F}_{2^n}$$

is a permutation polynomial if and only if

$$P(x) = a_0 + a_i X^{2^i-1}$$

with $a_i \neq 0$ and $Gcd(2^i - 1, 2^n - 1) = 1$.

We can deduce from this theorem a nice corollary.

Corollary 2. *Let $L(X) \in \mathcal{L}(n)$ be a linearized polynomial. Then there exists $a \in \mathbb{F}_{2^n}$ such that $L(X) + aX$ is a linearized permutation polynomial.*

Proof. We have to consider three cases.

- If $L(X) = a_0X$ then any $a \neq a_0$ is solution.
- If $L(X) = a_0X + a_iX^{2^i}$, $0 < i \leq n$ with $a_i \neq 0$, then taking $a = a_0$ we obtain
$$L(X) + aX = a_iX^{2^i}$$
which has clearly no other root than 0 and is thus bijective.
- If $L(X)$ is not of the form $a_0X + a_iX^{2^i}$, then from Theorem 4 we know that $P(X) = \frac{L(X)}{X}$ is not a permutation. If we choose $a \in \mathbb{F}_{2^n} \setminus Im(P)$ outside of its image, then
$$L(X) + aX = X\left(P(X) + a\right)$$
has its kernel reduced to $\{0\}$ and is therefore bijective.

4.2 The Case $\mathbb{F}_{2^{2^n}}$

We give here a result in the direction of Conjecture 2

Theorem 5. *Let n_0 be an integer such that the only unitary bilinear permutation over $\mathbb{F}_{2^{2^{n_0}}}$ is X^2.*

Then for all $n \geq n_0$, the only unitary bilinear permutation over $\mathbb{F}_{2^{2^n}}$ with coefficients in $\mathbb{F}_{2^{2^{n_0}}}$ is X^2.

Proof. Suppose that it is true until $n-1$, $n > n_0$. Set $t = 2^{n-1}$. Let

$$P(X) = \sum_{i=0}^{2t-1} \lambda_i X^{2^i+1} \quad \in \mathcal{B}(2^{2t})$$

with $\lambda_i \in \mathbb{F}_{2^{2^{n_0}}}$ be a permutation over $\mathbb{F}_{2^{2^n}}$. P must in particular permute $\mathbb{F}_{2^t}$. It follows from our hypothesis that $P \mod X^{2^t}+X$ must be equal to X^2 giving:

$$\sum_{i=0}^{t-1} (\lambda_i + \lambda_{i+t}) X^{2^i+1} = X^2.$$

This allows us to write

$$P(X) = XH(X) \text{ with } H(X) = X + \mathrm{Tr}_t^2\left(\sum_{i=1}^{t-1} \lambda_i X^{2^i}\right).$$

Note that H induces the identity over $\mathbb{F}_{2^t}$.
If $P(X) \neq X^2$, there exists $x \in \mathbb{F}_{2^{2t}} \setminus \mathbb{F}_{2^t}$ such that $H(x) + x = \beta \neq 0$. We then have

$$P(x+\beta) = P(x) + \beta\,(H(x) + x + \beta) = P(x)$$

proving that P is not a permutation polynomial.

The theorem follows by induction.

Corollary 3. *For all $n \geq 2$, the only unitary bilinear permutation over $\mathbb{F}_{2^{2^n}}$ with coefficients in $\mathbb{F}_{16}$ is X^2.*

Proof. For $n_0 = 2$, we can establish the result by exhaustive search. We then apply the previous theorem.

5 Conclusion

We described a new recursive family of quadratic permutation polynomials over $\mathbb{F}_{2^n}$. It enables us to construct easily many quadratic bilinear permutation polynomials over binary fields. Due to the recursive structure, the more odd factors n have, the more distincts permutation polynomials over $\mathbb{F}_{2^n}$ we will be able to construct.

On the other hand, if the degree of extension n is prime or if $n = 2^e$, we only obtain monomials. We thus conjecture that there exists no others. Moreover, for the case $n = 2^e$, we gave an argument supporting this conjecture.

beginthebibliography1

References

1. Lidl, R., Mullen, G.: When does a Polynomial over a Finite Field Permute the Elements of the Field? Amer. Math. Monthly 100, 71–74 (1993)
2. Lidl, R., Niederreiter, H.: Finite Fields, 2nd edn. Cambridge University Press, Cambridge (1997)

3. Patarin, J.: Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 33–48. Springer, Heidelberg (1996)
4. Dobbertin, H.: Almost Perfect Nonlinear Power Functions on GF(2N): The Niho Case. Inf. Comput. 151(1-2), 57–72 (1999)
5. Budaghyan, L., Carlet, C., Leander, G.: A Class of Quadratic APN Binomials Inequivalent to Power Functions. Cryptology ePrint Archive, Report 2006/445 (2006), http://eprint.iacr.org/
6. Blokhuis, A., Coulter, R.S., Henderson, M., O'Keefe, C.M.: Permutations Amongst the Dembowski-Ostrom Polynomials. In: 1999 Finite Fields and Applications, pp. 37–42. Springer, Berlin (2001)
7. Payne, S.: A Complete Determination of Translation Ovoids in Finite Desarguian Planes. Lincei - Rend. Sc. fis. mat. e nat. (1971)
8. Berger, T., Canteaut, A., Charpin, P., Laigle-Chapuy, Y.: Almost Perfect Nonlinear Functions. Technical Report RR-5774, INRIA Rocquencourt (2005), http://www.inria.fr/rrrt/rr-5774.html

Constructions of Orthonormal Lattices and Quaternion Division Algebras for Totally Real Number Fields*

B.A. Sethuraman[1] and Frédérique Oggier[2]

[1] Department of Mathematics
California State University, Northridge
al.sethuraman@csun.edu
[2] Department of Electrical Engineering
California Institute of Technology
frederique@systems.caltech.edu

Abstract. We describe some constructions of orthonormal lattices in totally real subfields of cyclotomic fields, obtained by endowing their ring of integers with a trace form. We also describe constructions of quaternion division algebras over such fields. Orthonormal lattices and quaternion division algebras over totally real fields find use in wireless networks in ultra wideband communication, and we describe the application.

1 Introduction

1.1 Algebraic Coding for Wireless Networks

We consider the problem of designing codes for a wireless relay network with $k+2$ nodes, each of them equipped with one antenna. Communication between the source node and the sink node is done with the help of k relay nodes. Several communication protocols have been proposed in the literature, and the one we will consider [1] belongs to the family of *amplify-and-forward* protocols, where each relay node just amplifies the signal it receives from the transmitter, before forwarding it to the receiver.

This protocol [1] is composed of k phases. During phase j, the source transmits in two steps. It sends a first signal to the jth relay and the destination. While the relay forwards the signal to the destination, the source further sends a second signal to the destination. This is repeated for each j, $j = 1, \ldots, k$.

For this protocol, the code design [16,2] consists of constructing *invertible* $2k \times 2k$ codewords, defined by

$$C = \mathrm{diag}(C_1, \ldots, C_k),$$

where C_j is a 2×2 matrix, $j = 1, \ldots, k$, containing $4k$ information symbols. The block diagonal form of C reflects the sequential nature of the protocol. Division algebras [13,10] have proved useful to design such invertible codewords.

* The first author is supported in part by NSF grant DMS-0700904.

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 138–147, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Codewords are usually (in narrow band systems) built over the complex field, but for ultra wideband communication, one needs to design them over the real field. Complex code constructions based on cyclic division algebras are proposed in [16]. In [2], examples of real codes are described for the case where the number of relays is at most 5. In this paper, we provide systematic code constructions for *arbitrary* number of relays, generalizing the approach in [2].

The general code design [2] consists of the following steps:

1. Choose a totally real number field F of degree k over $\mathbb{Q}$, which is cyclic, with Galois group generated by σ, and which is such that F and $\mathbb{Q}(\sqrt{5})$ are linearly disjoint over $\mathbb{Q}$. Let $\tau : \sqrt{5} \mapsto -\sqrt{5}$ be the generator of the Galois group of $\mathbb{Q}(\sqrt{5})/\mathbb{Q}$. Then $F(\sqrt{5})$ is Galois over $\mathbb{Q}$ with Galois group $\langle\sigma\rangle \times \langle\tau\rangle$. The Galois group of $F(\sqrt{5})/F$ is hence generated by τ.
2. Furthermore, choose F such that one can find a *trace lattice* (M, b_α) (see Subsection 1.2) inside the ring of integers of F that is isometric to the standard lattice $\mathbb{Z}^k \subseteq \mathbb{R}^k$. The orthonormal structure of M allows an efficient encoding [2] of information symbols, as detailed in Steps 4 and 5 below.
3. Now consider the cyclic algebra: $\mathcal{A} = (F(\sqrt{5})/F, \tau, \gamma)$, where γ is in F^*, and choose γ such that $\mathcal{A}$ is a division algebra. This will give us invertible codewords in Steps 4 and 5 below. Note that since $\mathcal{A}$ is a cyclic algebra of degree 2, it is also the quaternion algebra
$$\mathcal{A} = (5, \gamma). \tag{1}$$
4. Denote by C_0 a codeword from $\mathcal{A}$, that is of the form
$$C_0 = \begin{pmatrix} \sqrt{\alpha\lambda} & 0 \\ 0 & \sqrt{\alpha\tau(\lambda)} \end{pmatrix} \begin{pmatrix} a + b\nu & c + d\nu \\ \gamma(c + d\tau(\nu)) & a + b\tau(\nu) \end{pmatrix}$$
where $\nu = \frac{1+\sqrt{5}}{2}$, $\alpha \in \mathcal{D}_M^{-1}$ defines the trace form b_α and $\lambda = 2/(\sqrt{5}+5)$, chosen so that $(\sqrt{\lambda}, \sqrt{\lambda}\nu)$ and $(\sqrt{\tau(\lambda)}, \sqrt{\tau(\lambda)}\tau(\nu))$ are orthonormal in $\mathbb{R}^2$. Furthermore, if $\omega_1, \ldots, \omega_k$ is an orthonormal basis for (M, b_α), then $a = \sum_{i=1}^k \omega_i s_i$, $b = \sum_{i=1}^k \omega_i s_{k+i}$, $c = \sum_{i=1}^k \omega_i s_{2k+i}$, $d = \sum_{i=1}^k \omega_i s_{3k+i}$, where $s_1, \ldots, s_{4k}$ are information symbols from $\mathbb{Q}$.
5. We now define
$$C = \mathrm{diag}(\sigma(C_0), \ldots, \sigma^{k-1}(C_0), C_0)$$
where $\sigma(C_0)$ is obtained by applying σ to every entry of C_0. Furthermore
$$\tilde{vec}(C) = P \begin{pmatrix} \begin{pmatrix} \sqrt{\lambda} & \sqrt{\lambda}\nu \\ \sqrt{\tau(\lambda)} & \sqrt{\tau(\lambda)}\tau(\nu) \end{pmatrix} \otimes G & \mathbf{0} \\ \mathbf{0} & \begin{pmatrix} \gamma\sqrt{\lambda} & \gamma\sqrt{\lambda}\nu \\ \sqrt{\tau(\lambda)} & \sqrt{\tau(\lambda)}\tau(\nu) \end{pmatrix} \otimes G \end{pmatrix} \begin{pmatrix} s_1 \\ \vdots \\ s_{4k} \end{pmatrix}$$
where $\tilde{vec}(C)$ denotes the matrix C vectorized where the zero entries are removed, P is a permutation matrix, G is the generator matrix of M. Efficient encoding (or "shaping" [2,16]) requires the matrix that multiplies the

information symbols vector to be orthonormal, for which it is sufficient[1] that G be orthonormal.

To implement these steps above for an arbitrary number of relay nodes k, we thus need to find a totally real number field F of degree k over $\mathbb{Q}$ that is cyclic with generator σ, and that is linearly disjoint from $\mathbb{Q}(\sqrt{5})$, whose ring of integers allows the construction of an orthonormal trace lattice. Furthermore, we need to find a $\gamma \in F^*$ such that the algebra $(F(\sqrt{5})/F, \tau, \gamma)$ (which is the quaternion algebra $(5, \gamma)$) is a division algebra, where τ is the map that sends $\sqrt{5}$ to $-\sqrt{5}$.

We will discuss the cases where k is a power of 2 and a power of an odd prime separately in Sections 2 and 3, and then combine the cases in Section 4.

1.2 Trace Lattices

Let F be a totally real number field of degree k over $\mathbb{Q}$, and denote by $\mathcal{O}_F$ its ring of integers. Let $\sigma_1, \ldots, \sigma_k$ be the k embeddings of F into $\mathbb{R}$, and write $\mathrm{Tr}_{F/\mathbb{Q}}$ (or Tr_F when the context is clear) for the trace from F to $\mathbb{Q}$. We call an element $x \in F$ *totally positive* if $\sigma_i(x) \geq 0$, $i = 1, \ldots, k$. Let $M \subset \mathcal{O}_F$ be an integer lattice, that is, a subgroup of the additive group of $\mathcal{O}_F$. Being a $\mathbb{Z}$-submodule of a finitely generated and free $\mathbb{Z}$-module, M will also be finitely generated and free. We will focus on those M whose rank as a free $\mathbb{Z}$-module is exactly k (called *full lattices*). We denote by $\mathcal{D}_M^{-1}$ the codifferent of M, defined by

$$\mathcal{D}_M^{-1} = \{x \in F \mid \mathrm{Tr}_F(xM) \in \mathbb{Z}\}. \tag{2}$$

Definition 1. *A* trace lattice *is an integral lattice (M, b_α), where $M \subseteq \mathcal{O}_F$ is a (full) integer lattice, α is some totally positive element in $\mathcal{D}_M^{-1}$, and $b_\alpha : M \times M \to \mathbb{Z}$ is the bilinear form given by $b_\alpha(x, y) = Tr_F(\alpha xy)$. We refer to b_α as a trace form on M. We say that the trace lattice (M, b_α) is orthonormal if there exists a basis $\{\omega_1, \ldots, \omega_k\}$ such that $b_\alpha(\omega_i, \omega_j) = \delta_{i,j}$, in which case we say that the basis above is orthonormal.*

If $\{\omega_1, \ldots, \omega_k\}$ is a $\mathbb{Z}$-basis of M, then the trace lattice (M, b_α) can be embedded isometrically in $\mathbb{R}^k$ endowed with its standard bilinear form $\langle\ ,\ \rangle$ (the "dot product") by the map $\omega_i \mapsto f(\omega_i) = \big(\sqrt{\alpha_1}\sigma_1(\omega_i), \sqrt{\alpha_2}\sigma_2(\omega_i), \ldots, \sqrt{\alpha_k}\sigma_k(\omega_i)\big)$, where $\alpha_j = \sigma_j(\alpha)$, $j = 1, \ldots, k$ (note that α is totally positive). We may collect the $f(\omega_i)$ into a matrix known as the *generator matrix* of M, given by

$$G = \begin{pmatrix} \sqrt{\alpha_1}\sigma_1(\omega_1) & \sqrt{\alpha_2}\sigma_2(\omega_1) & \ldots & \sqrt{\alpha_k}\sigma_k(\omega_1) \\ \vdots & \vdots & \ldots & \vdots \\ \sqrt{\alpha_1}\sigma_1(\omega_k) & \sqrt{\alpha_2}\sigma_2(\omega_k) & \ldots & \sqrt{\alpha_k}\sigma_k(\omega_k) \end{pmatrix}. \tag{3}$$

One easily verifies that $GG^T = \{\mathrm{Tr}_F(\alpha\, \omega_i\omega_j)\}_{i,j=1}^k$, reflecting the fact that $b_\alpha(\omega_i, \omega_j) = \langle f(\omega_i), f(\omega_j)\rangle$. The basis $\{\omega_1, \ldots, \omega_k\}$ is an orthonormal basis if and only if GG^T is the identity matrix.

[1] γ should also be such that $|\gamma|^2 = 1$, which here prevents $\mathcal{A}$ to be a division algebra. This can be overcome, we refer the reader to [2, III.A.] for this discussion.

2 Totally Real Fields of Degree a Power of 2

Consider the cyclotomic field $L = \mathbb{Q}(\omega)$, where ω is the primitive 2^n-th root of unity $e^{2\pi\imath/n}$, for a positive integer $n \geq 3$. We write θ for the element $\omega + \omega^{-1}$, so that the maximal totally real subfield of L is given by $K = \mathbb{Q}(\theta)$. Note that $[L : \mathbb{Q}] = 2^{n-1}$ and $[K : \mathbb{Q}] = 2^{n-2}$. Let $k = 2^{n-2}$. We will work with the field K in this section. We will first construct an orthonormal lattice in $\mathcal{O}_K$, and then a suitable quaternion division algebra with center K.

2.1 $\mathcal{O}_K$ as an Orthonormal Lattice

We show here that $\mathcal{O}_K$ is an orthonormal lattice with respect to a suitable trace form. We have constructed this lattice after studying the $k = 2$ case presented in [2]. The existence of this lattice was sketched independently by Eva Bayer-Fluckiger and Gabriele Nebe in [5, Prop. 4.3]. We provide expanded proofs and some combinatorial remarks.

Note that $\mathcal{O}_K = \mathbb{Z}[\theta]$ (see [9, Exer. 35, Chap. 2] for instance). We write θ_j $(j = 0, 1, \ldots,)$ for the element $\omega^j + \omega^{-j}$; in particular, $\theta_1 = \theta$ and $\theta_0 = 2$. Expanding each power θ^s binomially and collecting terms we find

$$\theta^s = \begin{cases} \sum_{j=0}^{\lfloor s/2 \rfloor} \binom{s}{j} \theta_{s-2j} & \text{if } s \text{ is odd,} \\ \sum_{j=0}^{(s/2)-1} \binom{s}{j} \theta_{s-2j} + \binom{s}{s/2} & \text{if } s \text{ is even.} \end{cases} \tag{4}$$

It is easy to see that the relations (4) can inductively be inverted to write θ_s as a $\mathbb{Z}$-linear combination of θ^s. It follows that 1, $\theta_1 = \theta$, $\theta_2, \ldots$, θ_{k-1} is also a $\mathbb{Z}$-basis for $\mathcal{O}_K$.

We start by proving a property of the trace of the elements of the form θ_j.

Lemma 1. *For* $1 \leq j < 2k$,

$$Tr_{K/\mathbb{Q}}(\theta_j) = 0 \tag{5}$$

and for $1 \leq i, j \leq k - 1$

$$Tr_{K/\mathbb{Q}}(\theta_i \theta_j) = \begin{cases} 0 & \text{if } i \neq j \\ 2k & \text{if } i = j \end{cases} \tag{6}$$

Proof. First consider the case where j is odd. Since ω raised to any odd power is also a primitive 2^n-th root of unity, ω^j has minimal polynomial $x^k \pm \imath$ over $\mathbb{Q}(\imath)$, and consequently, ω^j has trace zero from L to $\mathbb{Q}(\imath)$. The same reasoning holds for $\omega^{-j} = (\omega^{-1})^j$ since ω^{-1} is also a primitive 2^n-th root of unity. It follows that $\mathrm{Tr}_{L/\mathbb{Q}(\imath)}(\theta_j) = 0$. Since $\mathrm{Tr}_{K/\mathbb{Q}}(\theta_j) = \mathrm{Tr}_{L/\mathbb{Q}(\imath)}(\theta_j)$, our result is proved when j is odd. (Notice that these arguments for odd j hold even if $j > 2k$.)

When j is even, we first assume that $j < k$. (This case is vacuous if $n = 3$.) If $j = 2m$, we write $2m$ as $2^e a$ for some $e \geq 1$ and odd integer a. Then ω^j is a

primitive 2^{n-e}-root of unity, and $[L : \mathbb{Q}(\omega^j)] = 2^e$. Since, by assumption, $e < n-2$, $\mathbb{Q}(\omega^j)$ strictly contains $\mathbb{Q}(\imath)$. Now, $\mathrm{Tr}_{L/\mathbb{Q}(\imath)}(\omega^j) = \mathrm{Tr}_{\mathbb{Q}(\omega^j)/\mathbb{Q}(\imath)}\mathrm{Tr}_{L/\mathbb{Q}(\omega^j)}(\omega^j) = 2^e\mathrm{Tr}_{\mathbb{Q}(\omega^j)/\mathbb{Q}(\imath)}(\omega^j)$. Just as in the previous paragraph, $\mathrm{Tr}_{\mathbb{Q}(\omega^j)/\mathbb{Q}(\imath)}(\omega^j)$ is zero since the minimal polynomial of ω^j is $x^{2^{n-e-2}} \pm \imath$. Since similar arguments hold for ω^{-j}, we find $\mathrm{Tr}_{L/\mathbb{Q}(\imath)}(\theta_j) = \mathrm{Tr}_{K/\mathbb{Q}}(\theta_j) = 0$.

Now assume $k \leq j < 2k$. Note that $\omega^k = \imath$ and $\omega^{-k} = -\imath$. Thus, when $j = k$, $\mathrm{Tr}_{L/\mathbb{Q}(\imath)}(\theta_j) = \mathrm{Tr}_{K/\mathbb{Q}}(\theta_j) = \imath - \imath = 0$. For $j > k$, $\omega^j = \imath\omega^{j-k}$, and by the considerations of the previous paragraph, $\mathrm{Tr}_{L/\mathbb{Q}(\imath)}(\omega^j) = \imath\mathrm{Tr}_{L/\mathbb{Q}(\imath)}(\omega^{j-k}) = 0$. Similarly, $\mathrm{Tr}_{L/\mathbb{Q}(\imath)}(\omega^{-j}) = 0$, so once again, $\mathrm{Tr}_{L/\mathbb{Q}(\imath)}(\theta_j) = \mathrm{Tr}_{K/\mathbb{Q}}(\theta_j) = 0$.

For the second assertion, note that $\theta_i\theta_j = \theta_{i+j} + \theta_{j-i}$, where we can assume without loss of generality that $j - i \geq 0$. The result immediately follows from the calculations of $\mathrm{Tr}_{K/\mathbb{Q}}(\theta_j)$ above, noting that $i + j < 2k$, and $\theta_0 = 2$.

Corollary 1. *For all x in $\mathcal{O}_K = \mathbb{Z}[\theta]$, the expression $Tr_{K/\mathbb{Q}}(1/k - \theta/2k)x$ takes values in $\mathbb{Z}$.*

Proof. Since trace is $\mathbb{Z}$-bilinear, this assertion can be checked for x coming from the basis 1, $\theta_1 = \theta$, θ_2, $\ldots$, θ_{k-1}. For such x the assertion is immediate from Lemma 1 above.

Write α for $1/k - \theta/2k$. Any element $\sigma \in Gal(K/\mathbb{Q})$ sends θ to θ_r for some odd r, so $\sigma(\theta)/2$ is a real number strictly between 1 and -1. Hence, α is totally positive, so as in Definition 1, we have the trace form $b_\alpha \colon \mathbb{Z}[\theta] \times \mathbb{Z}[\theta] \to \mathbb{Z}$ given by $b_\alpha(x, y) = \mathrm{Tr}_{K/\mathbb{Q}}(1/k - \theta/2k)xy$.

We first calculate the value of this bilinear form on the basis elements 1, $\theta_1 = \theta$, θ_2, $\ldots$, θ_{k-1}. (Note that this is really [5, Prop. 4.3], except that the authors in [5] work with the element $1/k + \theta/2k$.)

Lemma 2. *For $1 \leq j \leq i \leq k-1$, we have the formulas:*

$$b_\alpha(1, 1) = 1 \tag{7}$$

$$b_\alpha(1, \theta_i) = \begin{cases} -1 & \text{if } i = 1 \\ 0 & \text{if } i > 1 \end{cases} \tag{8}$$

$$b_\alpha(\theta_i, \theta_j) = \begin{cases} 2 & \text{if } j = i \\ -1 & \text{if } j = i + 1 \\ 0 & \text{if } j > i + 1 \end{cases} \tag{9}$$

Proof. The first two formulas arise from a direct application of the formulas in Lemma 1. For the third, we compute: $b_\alpha(\theta_i, \theta_j) = \mathrm{Tr}_{K/\mathbb{Q}}(1/k - \theta/2k)\theta_i\theta_j = (1/k)\mathrm{Tr}_{K/\mathbb{Q}}(\theta_i\theta_j) - (1/2k)\mathrm{Tr}_{K/\mathbb{Q}}(\theta\theta_i\theta_j)$. Now the formulas in Lemma 1 show that $(1/k)\mathrm{Tr}_{K/\mathbb{Q}}(\theta_i\theta_j)$ is zero except when $i = j$, in which case it is 2. As for the term $(\theta\theta_i\theta_j)$, note that like in the proof of Lemma 1, $\theta\theta_i\theta_j = \theta_1(\theta_{i+j} + \theta_{j-i}) = \theta_{i+j+1} + \theta_{i+j-1} + \theta_{j-i+1} + \theta_{j-i-1}$. When $i = j$ and when $j > i + 1$, Lemma 1 shows that $(1/2k)\mathrm{Tr}_{K/\mathbb{Q}}(\theta\theta_i\theta_j)$ is zero. When $i = j + 1$ the term $\theta_{j-i-1} = 2$ contributes $-(1/2k)2k$ to the trace. This establishes the formula. □

The lemma above immediately leads to the following (see the remark in [5] at the end of the proof of their Prop. 4.3):

Theorem 1. *The vectors* $w_0 = 1$, $w_1 = 1 + \theta_1$, $w_2 = 1 + \theta_1 + \theta_2$, $\ldots$, $w_{k-1} = 1 + \theta_1 + \theta_2 + \cdots + \theta_{k-1}$ *form an orthonormal basis for* $\mathcal{O}_K$ *with respect to the trace form* $b_\alpha(x, y)$ *described above.*

Proof. We prove this inductively. The assertion that $b_\alpha(w_0, w_0) = 1$ is just the first formula in Lemma 2 above. Now assume that we have proved that the vectors $w_0, \ldots, w_i$ are orthonormal. First, for a given $j < k$ and $l < k$, we expand w_j as $1 + \theta_1 + \cdots + \theta_j$ and using the bilinearity of b_α, we see that $b_\alpha(w_j, \theta_l) = 0$ whenever $l > j + 1$, and $b_\alpha(w_j, \theta_l) = -1$ if $l = j + 1$. From this and the induction assumption, it follows that for $j \leq i$, $b_\alpha(w_j, w_{i+1}) = b_\alpha(w_j, w_j) + b_\alpha(w_j, \theta_{j+1}) + \cdots + b_\alpha(w_j, \theta_{i+1}) = 1 - 1 = 0$. Also, $b_\alpha(w_{i+1}, w_{i+1}) = b_\alpha(w_i, w_i) + 2b_\alpha(w_i, \theta_{i+1}) + b_\alpha(\theta_{i+1}, \theta_{i+1}) = 1 - 2 + 2 = 1$. This proves the theorem. □

To compute the generator matrix for this lattice, note that the Galois group $Gal(K/\mathbb{Q})$ is generated by the action on K of $\sigma : \omega \mapsto \omega^r$, where r is some generator of the multiplicative group $(\mathbb{Z}/2^{n-1}\mathbb{Z})^*$. Thus, $\sigma(\theta_1) = \theta_r$, $\sigma(\theta_2) = \theta_{2r}$, $\sigma(1/k - \theta_1/2k) = 1/k - \theta_r/2k$ etc.

Some combinatorial remarks: There is a nice interplay between the two $\mathbb{Z}$-bases $1, \theta, \theta^2, \ldots, \theta^{k-1}$ (consisting of powers of θ), and the basis $1, \theta_1 = \theta, \theta_2, \ldots, \theta_{k-1}$, which leads to some interesting combinatorial considerations. For instance, we can compute the codifferent of $\mathcal{O}_K$ in terms of the two bases, and doing so, we are led to the *Hankel transform* of the binomial sequence $\binom{2n}{n}$: these have been studied by various authors ([12], [8],[15], for example) and is defined as the sequence h_n, $n = 1, 2, \cdots$, where h_n is the determinant of the $n \times n$ matrix

$$\begin{pmatrix} \binom{0}{0} & \binom{2}{1} & \cdots & \binom{2(n-1)}{n-1} \\ \binom{2}{1} & \binom{4}{2} & \cdots & \binom{2n}{n} \\ \vdots & \vdots & \vdots & \vdots \\ \binom{2(n-1)}{n-1} & \binom{2n}{n} & \cdots & \binom{4(n-1)}{2(n-1)} \end{pmatrix}. \tag{10}$$

We will be exploring this connection in [14].

In a different direction, one can check that the vectors w_i described in Theorem 1 above can be defined in terms of the powers θ^i by the following inductive scheme: $w_0 = 1$, $w_l = \sum_{s=0}^{l-1} a_s^{(l)} w_s + \theta^l$ for $l \geq 1$, where

$$a_s^{(l)} = \begin{cases} (-1)^{s+1}\binom{2t}{t - \lfloor \frac{s+1}{2} \rfloor}, & l = 2t; \\ (-1)^s \binom{2t+1}{t - \lfloor \frac{s}{2} \rfloor}, & l = 2t + 1. \end{cases} \tag{11}$$

(Indeed, this is the form in which we originally discovered our lattice. The various expressions on the right side of the definition of the $a_s^{(l)}$ above are all the binomial coefficients of the form $\binom{l}{j}$, starting from the middle and and working

towards both ends, taking one alternately on each side.) Proving the orthonormality of the w_i *directly* in this form without invoking Theorem 1 above leads to the following interesting combinatorial identities:

$$1 + \sum_{s=0}^{l} \left(a_s^{(l+1)}\right)^2 = \binom{2l+2}{l+1},$$

and, for $j > i$,

$$\sum_{s=0}^{i-1} a_s^{(i)} a_s^{(j)} - a_i^{(j)} = \begin{cases} -\binom{i+j}{(i+j+1)/2-1} & \text{if } i+j \text{ is odd,} \\ \binom{i+j}{(i+j)/2} & \text{if } i+j \text{ is even.} \end{cases} \tag{12}$$

2.2 A Quaternion Division Algebra over K

We now need to build a suitable quaternion divison algebra $\mathcal{A} = (5, \gamma)$ on K (see (1)). We will prove in this subsection the following result:

Theorem 2. *The algebra $\mathcal{A} = (5, 2-\theta)$ defined over K is a division algebra.*

Proof. We need to show that $2-\theta$ is not a norm from $K(\sqrt{5})$ to K. Observe that $2-\theta = (1-\omega)(1-\omega^{-1})$. It is a standard fact that there is a unique prime ideal $\tilde{P}$ in $\mathcal{O}_L$ that lies over 2, that it has ramification index $e = [L : \mathbb{Q}] = 2^{n-1}$ and inertial degree $f = 1$, and that it is generated by both $1-\omega$ and $1-\omega^{-1}$ (see for instance [9, Chap 3, Theo. 26]; note that ω^{-1} is also a primitive 2^n-th root of unity). It follows that there is a unique prime ideal lying over 2 in $\mathcal{O}_K$, call it P, and that $P\mathcal{O}_L = \tilde{P}^2$. But $\tilde{P}^2 = (1-\omega)\mathcal{O}_L(1-\omega^{-1})\mathcal{O}_L = (2-\theta)\mathcal{O}_L$. Since $2-\theta$ is already in $\mathcal{O}_K$, it follows that $P = \tilde{P}^2 \cap \mathcal{O}_K = (2-\theta)\mathcal{O}_K$.

Now we consider how P extends to $K(\sqrt{5})$. To do this, note that the prime 2 of $\mathbb{Z}$ stays prime in the field $\mathbb{Q}(\sqrt{5})$ (see [9, Chap. 3, Theo. 25] for instance.) Call this prime of $\mathcal{O}_{\mathbb{Q}(\sqrt{5})}$ P', so $e(P'|2\mathbb{Z}) = 1$ and $f(P'|2\mathbb{Z}) = 2$. Now if Q is any prime of $\mathcal{O}_{K(\sqrt{5})}$ lying over P, then $e(Q|2\mathbb{Z}) = e(Q|P)e(P|2\mathbb{Z}) \geq e(P|2\mathbb{Z}) = k$, and $f(Q|2\mathbb{Z}) = f(Q|P')f(P'|2\mathbb{Z}) \geq f(P'|2\mathbb{Z}) = 2$. Since $k \cdot 2$ already equals $[K(\sqrt{5}) : \mathbb{Q}]$, we find that Q is the unique prime in $K(\sqrt{5})$ lying over 2 and that $e(Q|2\mathbb{Z}) = k$ and $f(Q|2\mathbb{Z}) = 2$. In particular, this means that Q is the unique prime of $\mathcal{O}_K$ lying over P, and that $e(Q|P) = 1$ and $f(Q|P) = 2$.

Now assume that $2-\theta = N(x)$, for some $x \in K(\sqrt{5})$, where we have written N for the norm from $K(\sqrt{5})$ to K. Further writing $x = y/z$ for y and z in $\mathcal{O}_{K(\sqrt{5})}$, we find $N(z)(2-\theta) = N(y)$. Assume that the ideal $y\mathcal{O}_{K(\sqrt{5})}$ has the factorization $Q^l \cdot Q_1^{l_1} \cdots Q_r^{l_r}$ where the Q_i are primes other than Q and l and the l_i are nonnegative integers. Assume similarly that $z\mathcal{O}_{K(\sqrt{5})}$ has the factorization $Q^{l'} \cdot (Q_1')^{l_1'} \cdots (Q_{r'}')^{l_{r'}'}$. Then the ideal $N(y)\mathcal{O}_K$ in $\mathcal{O}_K$ has the factorization $P^{2l} \cdot P_1^{f_1 l_1} \cdots P_r^{f_r l_r}$, where the f_i are the inertial degrees of the primes Q_i, and $P_i = Q_i \cap \mathcal{O}_K$. (This follows, for instance from [9, Chap 3, Exer. 14]; note that we have used the fact that $f(Q|P) = 2$.) Similarly, $N(z)\mathcal{O}_K$ in $\mathcal{O}_K$ has the factorization $P^{2l'} \cdot (P_1')^{f_1' l_1'} \cdots (P_{r'}')^{f_{r'}' l_{r'}'}$. But then, since the ideal $(2-\theta)\mathcal{O}_K$ is

just P, we find that the powers of P in the associated factorization of ideals $N(y)\mathcal{O}_K = PN(z)\mathcal{O}_K$ do not match up, a contradiction. Hence, $(5, 2-\theta)$ is a division algebra over K.

3 Totally Real Fields of Odd Degree

3.1 An Orthonormal Lattice in $\mathcal{O}_K$

An example of an orthonormal lattice in totally real number fields K of degree p an odd prime was given by Erez ([7]). It was later pointed out in [6] that Erez' construction works, without any modification, for any odd degree k. We quote the construction from [6] with minor changes in notation:

- Pick a (guaranteed to exist) odd prime $p \equiv 1 \pmod{k}$.
- Set $\omega = \omega_p = e^{\frac{2\pi\imath}{p}}$ and let σ denote the generator of the cyclic Galois group $Gal(\mathbb{Q}(\omega)/\mathbb{Q})$.
- Find a primitive element r of the multiplicative group $(\mathbb{Z}/p\mathbb{Z})^*$.
- For $m = \frac{p-1}{2}$, create $\alpha = \prod_{j=0}^{m-1}(1-\omega^{r^j})$.
- Find a (guaranteed to exist) λ such that $\lambda(r-1) \equiv 1 \pmod{p}$ and let $z = \omega^{\lambda}\alpha(1-\omega)$.
- For $\sigma(\omega) = \omega^r$, let $x = \sum_{j=1}^{\frac{p-1}{k}} \sigma^{jk}(z)$.

The element x is hence in the field K, the subfield of $\mathbb{Q}(\omega)$ fixed by σ^k, of degree k over $\mathbb{Q}$. Then the matrix G given below is unitary:

$$G = \frac{1}{p}\begin{pmatrix} x & \sigma(x) & \cdots & \sigma^{k-2}(x) & \sigma^{k-1}(x) \\ \sigma(x) & \sigma^2(x) & \cdots & \sigma^{k-1}(x) & x \\ \sigma^2(x) & \sigma^3(x) & \cdots & x & \sigma(x) \\ & \vdots & & \vdots & \\ \sigma^{k-1}(x) & x & \cdots & \sigma^{k-3}(x) & \sigma^{k-2}(x) \end{pmatrix}. \tag{13}$$

Note that since k divides m as well, any element fixed by σ^k is also fixed by σ^m. Thus, K is contained in the fixed field of σ^m, which is the totally real field $\mathbb{Q}(\omega+\omega^{-1})$. K is hence totally real. Also, note that since z is integral over $\mathbb{Q}$, the element x is in $\mathcal{O}_k$. The fact that this matrix above is unitary says that the elements $x, \sigma(x), \ldots, \sigma^{k-1}(x)$ form an orthonormal basis for M with respect to the trace form $b_\gamma : M \times M \to \mathbb{Z}$ given by $b_\gamma(s,t) = \mathrm{Tr}_{K/\mathbb{Q}}(\gamma st)$, where $\gamma = 1/p^2$ (see the matrix G in the remark following Definition 1).

Remark: For the field $K = \mathbb{Q}(\omega+\omega^{-1})$, where ω is a primitive p^n-th root of unity, and p is an odd prime, it would be interesting to see if, just as for $p = 2$ in Subsection 2.1, there exists a suitable trace form for which $\mathcal{O}_K$ turns out to be an orthonormal lattice. Such a trace form is known to exist if $n = 1$ [4], but this construction does not hold for $n \geq 2$. The existence of such trace forms for general p and n is open as far as we know. For the special case of $K = \mathbb{Q}(\omega_9 + \omega_9^{-1})$, where we have written ω_9 for $e^{2\pi\imath/9}$, one can check that the vectors $-(1-\theta)\theta$, $-\theta$, $-1+\theta$ (where $\theta = \omega_9 + \omega_9^{-1}$) form an orthonormal basis

for $\mathcal{O}_K$ with respect to the trace form $b_\alpha(x,y) = Tr_K(\alpha xy)$, where α is the (totally positive) element $(16-\theta-5\theta^2)/9$.

3.2 A Quaternion Division Algebra over K

To construct a quaternion division algebra $\mathcal{A} = (5,\gamma)$ over K (as described in (1)), it is sufficient to take a quaternion division algebra over $\mathbb{Q}$ and consider it as an algebra over K: this follows from the result that if D is a division algebra of index m over a field F and if L/F is a field extension of degree n relatively prime to m, then $D \otimes_F L$ remains a division algebra ([11, Chap. 13, §4, Prop.]).

For this, note, for example, that $(5,2)$ is a division algebra over $\mathbb{Q}$. For, if 2 is the norm from $\mathbb{Q}(\sqrt{5})$ to $\mathbb{Q}$ of an element $x = (a+b\sqrt{5})/m$, where a, b, and m are integers, then we find $2m^2 = a^2 - 5b^2$. If m is divisible by 5, so must a, and then, so must b. Hence, we can repeatedly cancel 5^2 from both sides until m is not divisible by 5. Now reducing mod 5 and noting m is not zero mod 5, we find $2 = (a/m)^2$. But this is a contradiction as 2 is not a square mod 5. Hence, we may use $(5,2)$ as our quaternion division algebra over K.

4 Totally Real Fields of Arbitrary Degree

Finally, to construct lattices and quaternion division algebras over totally real number fields of arbitrary degree, we just have to combine the constructions in the previous two sections. Given an arbitrary positive integer $k \geq 2$, write $k = 2^m k'$, where k' is odd. We may assume that $m \geq 1$ and $k' \geq 3$, else we are in the situation of the previous sections. Write K_e for the field obtained in Section 2 of degree 2^m over $\mathbb{Q}$. Write M_e for the lattice obtained in that same section, b_{α_e} for its bilinear form, and G_e for the generator matrix that defines its isometric embedding in $\mathbb{R}^{2^m}$. Similarly, write K_o for the field obtained in Section 3 of degree k', M_o for the lattice obtained in that section, b_{α_o} for its bilinear form, and G_o for the generator matrix that defines its isometric embedding in $\mathbb{R}^{k'}$. Then, since the degrees of K_e and K_o are relatively prime, the compositum $K = K_eK_o$ has degree $k = 2^m k'$ over $\mathbb{Q}$. It is totally real since both K_e and K_o are totally real. (In fact, K is Galois over $\mathbb{Q}$ with Galois group $Gal(K_e/\mathbb{Q}) \times Gal(K_o/\mathbb{Q})$.)

If $\{c_i\}$ ($c_i \in K_e$) is an orthonormal basis for M_e, and if $\{d_j\}$ ($d_j \in K_o$) is an orthonormal basis for M_o, it is easy to see that the set $\{c_id_j\}$ is $\mathbb{Z}$-linearly independent, and hence generates a free submodule N of $\mathcal{O}_K$. We have the bilinear form $b_{\alpha_e\alpha_o}$, defined on the basis by

$$\begin{aligned} &b_{\alpha_e\alpha_o}(c_id_j, c_sd_t) = \mathrm{Tr}_{K/\mathbb{Q}}(\alpha_e\alpha_o c_id_jc_sd_t) = \mathrm{Tr}_{K_e/\mathbb{Q}}(\mathrm{Tr}_{K/K_e}(\alpha_e\alpha_o c_id_jc_sd_t)) \\ &= \mathrm{Tr}_{K_e/\mathbb{Q}}(\alpha_e c_ic_s\mathrm{Tr}_{K/K_e}(\alpha_o d_jd_t)) = \mathrm{Tr}_{K_e/\mathbb{Q}}(\alpha_e c_ic_s\mathrm{Tr}_{K_o/\mathbb{Q}}(\alpha_o d_jd_t)) \\ &= \mathrm{Tr}_{K_e/\mathbb{Q}}(\alpha_e c_ic_s)\mathrm{Tr}_{K_o/\mathbb{Q}}(\alpha_o d_jd_t) = b_e(c_i,c_s)b_o(d_j,d_t). \end{aligned}$$

The basis $\{c_id_j\}$ is orthonormal: $b_{\alpha_e\alpha_o}(c_id_jc_sd_t) = \delta_{(i,j),(s,t)}$. Since $Gal(K/\mathbb{Q}) \cong Gal(K_e/\mathbb{Q}) \times Gal(K_o/\mathbb{Q})$, we may write every element $\phi \in Gal(K/\mathbb{Q})$ as a product $\sigma\tau$ of elements $\sigma \in Gal(K_e/\mathbb{Q})$ and $\tau \in Gal(K_o/\mathbb{Q})$. Hence, $\phi(\alpha_e\alpha_o) =$

$\sigma(\alpha_e)\tau(\alpha_o)$, $\phi(c_i d_j) = \sigma(c_i)\tau(d_j)$, etc. Using this, it is easy to see that the orthonormal trace lattice $(N, b_{\alpha_e \alpha_o})$ embeds isometrically into $\mathbb{R}^k$ via the Kronecker product of the matrices G_e and G_o.

To obtain a quaternion division algebra over K, we simply consider the quaternion division algebra $\mathcal{A}$ obtained over K_e in Section 2 as an algebra over K. Since K is of odd degree over K_e, $\mathcal{A} \otimes_{K_e} K$ remains a division algebra by ([11, Chap. 13, §4, Prop.]).

References

1. Azarian, K., El Gamal, H., Schniter, P.: On the Achievable Diversity-Multiplexing Tradeoff in Half-Duplex Cooperative Channels. IEEE Trans. Inform. Theory 51(12), 4152–4172 (2005)
2. Abou-Rjeily, C., Daniele, N., Belfiore, J.-C.: Distributed Algebraic Space Time Codes for Ultra Wideband Communications. Kluwer Journal, Special Issue on Cooperative Diversity (2006)
3. Bayer-Fluckiger, E.: Lattices and Number Fields. Contemporary Mathematics 241, 69–84 (1999)
4. Bayer, E., Oggier, F., Viterbo, E.: New Algebraic Constructions of Rotated $\mathbb{Z}^n$ Lattice Constellations for the Rayleigh Fading Channel. IEEE Trans. Inform. Theory 50(4), 702–714 (2004)
5. Bayer-Fluckiger, E., Nebe, G.: On the Euclidean Minimum of Some Real Number Fields. J. Théo. Nombres Bordeaux 17, 437–454 (2005)
6. Elia, P., Sethuraman, B.A., Kumar, P.V.: Perfect Space-Time Codes with Minimum and Non-Minimum Delay for Any Number of Antennas. IEEE Trans. Inform. Theory (to appear)
7. Erez, B.: The Galois structure of the Trace Form in Extensions of Odd Prime Degree. J. of Algebra 118, 438–446 (1988)
8. Layman, J.W.: Then Hankel Transform and Some of Its Properties. J. Integer Sequences 4, Article 01.1.5 (2001)
9. Marcus, D.A.: Number Fields. Universitext. Springer, NY (1977)
10. Oggier, F.E., Rekaya, G., Belfiore, J.-C., Viterbo, E.: Perfect Space-Time Block Codes. IEEE Trans. Inform. Theory 52(9), 3885–3902 (2006)
11. Pierce, R.S: Associative Algebras. GTM88. Springer, NY (1982)
12. Radoux, C.: Calcul effectif de certains determinants de Hankel. Bull. Soc. Math. Belg. 31(1), 49–55 (1979)
13. Sethuraman, B.A., Rajan, B.S., Shashidhar, V.: Full-diversity, High-Rate Space-Time Block Codes from Division Algebras. IEEE Trans. Inform. Theory 49, 2596–2616 (2003)
14. Sethuraman, B.A., Oggier, F.E.: The Hankel Transform of the Central Binomial Coefficients and Orthonormal Lattices in Cyclotomic Fields (in preparation)
15. Spivey, M.Z., Steil, L.L.: The k-Binomial Transform and the Hankel Transform. J. Integer Sequences 9, Article 06.1.1 (2006)
16. Yang, S., Belfiore, J.-C.: Optimal Space-Time Codes For The Mimo Amplify-And-Forward Cooperative Channel. IEEE Trans. Inform. Theory 53(2), 647–663 (2007)

Quaternary Plotkin Constructions and Quaternary Reed-Muller Codes*

J. Pujol[1], J. Rifà[1], and F.I. Solov'eva[2]

[1] Department of Information and Communications Engineering,
Universitat Autònoma de Barcelona, 08193-Bellaterra, Spain
[2] Sobolev Institute of Mathematics,
Novosibirsk State University, Novosibirsk, Russia

Abstract. New quaternary Plotkin constructions are given and are used to obtain new families of quaternary codes. The parameters of the obtained codes, such as the length, the dimension and the minimum distance are studied. Using these constructions new families of quaternary Reed-Muller codes are built with the peculiarity that after using the Gray map the obtained $\mathbb{Z}_4$-linear codes have the same parameters as the codes in the classical binary linear Reed-Muller family.

Keywords: Quaternary codes, Plotkin constructions, Reed-Muller codes, Z4-linear codes.

1 Introduction

In [13] Nechaev introduced the concept of $\mathbb{Z}_4$-linearity of binary codes and later Hammons, Kumar, Calderbank, Sloane and Solé, see [7], showed that several families of binary codes are $\mathbb{Z}_4$-linear. In [7] it is proved that the binary linear Reed-Muller code $RM(r,m)$ is $\mathbb{Z}_4$-linear for $r = 0, 1, 2, m-1, m$ and is not $\mathbb{Z}_4$-linear for $r = m-2$ ($m \geq 5$). In a subsequent work, Hou, Lahtonen and Koponen, [8] proved that $RM(r,m)$ is not $\mathbb{Z}_4$-linear for $3 \leq r \leq m-2$.

In [7] the construction of Reed Muller codes, $\mathcal{QRM}(r,m)$, based on Z_4 linear codes is introduced such that after doing modulo two we obtain the usual binary linear Reed-Muller (RM) codes. In [2,3] such family of codes is studied and their parameters are computed as well as the dimension of the kernel and rank. In [15] some kind of Plotkin construction was used to build a family of additive Reed-Muller codes and also in [17] the Plotkin construction was utilized to obtain a sequence of quaternary linear Reed-Muller like codes. In both last quoted constructions, images of the obtained codes under the Gray map are binary codes with the same parameters as the classical binary linear RM codes.

Moreover, on the other hand, in [9,10] all the non-equivalent $\mathbb{Z}_4$-linear extended 1-perfect codes and their duals, the $\mathbb{Z}_4$-linear Hadamard codes, are classified. It is a natural question to ask if there exist families of quaternary linear

* This work has been partially supported by the Spanish MEC and the European FEDER Grant MTM2006-03250 and also by the UAB grant PNL2006-13.

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 148–157, 2007.
© Springer-Verlag Berlin Heidelberg 2007

codes such that, after the Gray map, the corresponding $\mathbb{Z}_4$-linear codes have the same parameters as the well known family of binary linear RM codes. In this new families, like in the usual $RM(r,m)$ family, the code with parameters $(r,m)=(1,m)$ should be a Hadamard code and the code with parameters $(r,m)=(m-2,m)$ should be an extended 1-perfect code.

It is well known that an easy way to built the RM family of codes is by using the Plotkin construction (see [12]). So, it seems a good matter of study to try to generalize the Plotkin construction to the quaternary codes and try to obtain new families of codes which contain the above mentioned $\mathbb{Z}_4$-linear Hadamard codes and $\mathbb{Z}_4$-linear extended 1-perfect codes and fulfil the same properties (from a parameters point of view) than the binary RM family.

The present paper is organized as follows. In Section 2 we introduce the concept of quaternary code and give some constructions that could be seen as quaternary generalizations of the well known binary Plotkin construction. In Section 3, we construct several families of $\mathbb{Z}_4$-linear RM codes and prove that they have similar parameters as the classical RM codes but they are not linear. Finally, in Section 4 we give some conclusions and further research in the same topic. The family of codes presented in the paper contains codes from [17].

2 Constructions of Quaternary Codes

2.1 Quaternary Codes

Let $\mathbb{Z}_2$ and $\mathbb{Z}_4$ be the ring of integers modulo two and modulo four, respectively. Let $\mathbb{Z}_2^n$ be the set of all binary vectors of length n and $\mathbb{Z}_4^N$ be the set of all quaternary vectors of length N. Any non-empty subset C of $\mathbb{Z}_2^n$ is a binary code and a subgroup of $\mathbb{Z}_2^n$ is called a *binary linear code* or a $\mathbb{Z}_2$*-linear code*. Equivalently, any non-empty subset $\mathcal{C}$ of $\mathbb{Z}_4^N$ is a quaternary code and a subgroup of $\mathbb{Z}_4^N$ is called a *quaternary linear code*. In general, any non-empty subgroup $\mathcal{C}$ of $\mathbb{Z}_2^\alpha \times \mathbb{Z}_4^\beta$ is an additive code.

The Hamming weight $w(v)$ of a vector v in $\mathbb{Z}_2^n$ is the number of its nonzero coordinates. The Hamming distance $d(u,v)$ between two vectors $u,v \in \mathbb{Z}_2^n$ is $d(u,v)=w(u-v)$. For quaternary codes it is more interesting to use the Lee metric (see [11]). In $\mathbb{Z}_2$ the Lee weight coincides with the Hamming weight, but in $\mathbb{Z}_4$ the Lee weight of their elements is $w_L(0)=0, w_L(1)=w_L(3)=1$, and $w_L(2)=2$. The Lee weight $w_L(v)$ of a vector v in $\mathbb{Z}_4^N$ is the addition of the Lee weight of all the coordinates. The Lee distance $d_L(u,v)$ between two vectors $u,v \in \mathbb{Z}_4^N$ is $d_L(u,v)=w_L(u-v)$.

Let $\mathcal{C}$ be an additive code, so a subgroup of $\mathbb{Z}_2^\alpha \times \mathbb{Z}_4^\beta$ and let $C=\Phi(\mathcal{C})$, where $\Phi : \mathbb{Z}_2^\alpha \times \mathbb{Z}_4^\beta \longrightarrow \mathbb{Z}_2^n$, $n=\alpha+2\beta$, is given by $\Phi(x,y)=(x,\phi(y))$ for any x from $\mathbb{Z}_2^\alpha$ and any y from $\mathbb{Z}_4^\beta$, where $\phi : \mathbb{Z}_4^\beta \longrightarrow \mathbb{Z}_2^{2\beta}$ is the usual Gray map, so $\phi(y_1,\ldots,y_\beta)=(\varphi(y_1),\ldots,\varphi(y_\beta))$, and $\varphi(0)=(0,0), \varphi(1)=(0,1), \varphi(2)=(1,1)$, $\varphi(3)=(1,0)$. Hamming and Lee weights, as well as Hamming and Lee distances, can be generalized, in a natural way, to vectors in $\mathbb{Z}_2^\alpha \times \mathbb{Z}_4^\beta$ by adding the corresponding weights (or distances) of the $\mathbb{Z}_2^\alpha$ part and the $\mathbb{Z}_4^\beta$ part.

Since $\mathcal{C}$ is a subgroup of $\mathbb{Z}_2^\alpha \times \mathbb{Z}_4^\beta$, it is also isomorphic to an abelian structure like $\mathbb{Z}_2^\gamma \times \mathbb{Z}_4^\delta$. Therefore, we have that $|\mathcal{C}| = 2^\gamma 4^\delta$ and the number of order two codewords in $\mathcal{C}$ is $2^{\gamma+\delta}$. We call such code $\mathcal{C}$ an *additive code of type* $(\alpha, \beta; \gamma, \delta)$ and the binary image $C = \Phi(\mathcal{C})$ a $\mathbb{Z}_2\mathbb{Z}_4$-*linear code of type* $(\alpha, \beta; \gamma, \delta)$. In the specific case $\alpha = 0$ the code $\mathcal{C}$ is quaternary linear and the code C is called a $\mathbb{Z}_4$-*linear code*. Note that the binary length of the binary code $C = \Phi(\mathcal{C})$ is $n = \alpha + 2\beta$.

The minimum Hamming distance d of a $\mathbb{Z}_2\mathbb{Z}_4$-linear code C is the minimum value of $d(u, v)$, where $u, v \in C$ and $u \neq v$. Notice that the Hamming distance of a $\mathbb{Z}_2\mathbb{Z}_4$-linear code C coincides with the Lee distance defined for the additive code $\mathcal{C} = \phi^{-1}(C)$. From now on, when we work with distances it must be understood that we are working with Hamming distances in the case of binary codes or Lee distances in the additive case.

Although $\mathcal{C}$ could not have a basis, it is important and appropriate to define a generator matrix for $\mathcal{C}$ as:

$$\mathcal{G} = \left(\begin{array}{c|c} B_2 & Q_2 \\ \hline B_1 & Q_1 \end{array}\right),$$

where B_2 is a $\gamma \times \alpha$ matrix; Q_2 is a $\gamma \times \beta$ matrix; B_1 is a $\delta \times \alpha$ matrix and Q_1 is a $\delta \times \beta$ matrix. Matrices B_1, B_2 are binary and Q_1, Q_2 are quaternary, but the entries in Q_2 are only zeroes or twos.

Two additive codes $\mathcal{C}_1$ and $\mathcal{C}_2$ both of the same length are said to be *monomial equivalent*, if one can be obtained from the other by permuting the coordinates and changing the signs of certain coordinates. Additive codes which differ only by a permutation of coordinates are said to be *permutational equivalent*.

We will use the following definition (see [16]) of the inner product in $\mathbb{Z}_2^\alpha \times \mathbb{Z}_4^\beta$:

$$\langle u, v \rangle = 2(\sum_{i=1}^{\alpha} u_i v_i) + \sum_{j=\alpha+1}^{\alpha+\beta} u_j v_j \in \mathbb{Z}_4, \tag{1}$$

where $u, v \in \mathbb{Z}_2^\alpha \times \mathbb{Z}_4^\beta$. Note that when $\alpha = 0$ the inner product is the usual one for vectors over $\mathbb{Z}_4$ and when $\beta = 0$ it is twice the usual one for binary vectors.

The *additive dual code* of $\mathcal{C}$, denoted by $\mathcal{C}^\perp$, is defined in the standard way

$$\mathcal{C}^\perp = \{u \in \mathbb{Z}_2^\alpha \times \mathbb{Z}_4^\beta \mid \langle u, v \rangle = 0 \text{ for all } v \in \mathcal{C}\}.$$

The corresponding binary code $\Phi(\mathcal{C}^\perp)$ is denoted by $C_\perp$ and called the $\mathbb{Z}_2\mathbb{Z}_4$-*dual code* of C. In the case $\alpha = 0$, $\mathcal{C}^\perp$ is also called the *quaternary dual code* of $\mathcal{C}$ and $C_\perp$ the $\mathbb{Z}_4$-*dual code* of C.

The additive dual code $\mathcal{C}^\perp$ is also an additive code, that is a subgroup of $\mathbb{Z}_2^\alpha \times \mathbb{Z}_4^\beta$. Its weight enumerator polynomial is related to the weight enumerator polynomial of $\mathcal{C}$ by the MacWilliams identity (see [6]). Notice that C and $C_\perp$ are not dual in the binary linear sense but the weight enumerator polynomial of $C_\perp$ is the McWilliams transform of the weight enumerator polynomial of C.

Given an additive code $\mathcal{C}$ it is well known the value of the parameters of the additive dual code (see [4] for additive codes with $\alpha \neq 0$ and [7] for additive codes with $\alpha = 0$).

From now on we focus our attention specifically to additive codes with $\alpha = 0$, so quaternary linear codes such that after the Gray map they give rise to $\mathbb{Z}_4$-linear codes. Given a quaternary linear code of type $(0, \beta; \gamma, \delta)$ we will write $(N; \gamma, \delta)$ to say that $\alpha = 0$ and $\beta = N$.

2.2 Plotkin Construction

In this section we show that the well known Plotkin construction can be generalized to quaternary codes. Let $\mathcal{A}$ and $\mathcal{B}$ be any two quaternary linear codes of types $(N; \gamma_{\mathcal{A}}, \delta_{\mathcal{A}})$ and $(N; \gamma_{\mathcal{B}}, \delta_{\mathcal{B}})$ and minimum distances $d_{\mathcal{A}}$, $d_{\mathcal{B}}$ respectively.

Definition 1 (Plotkin Construction). *Given the quaternary linear codes $\mathcal{A}$ and $\mathcal{B}$, we define a new quaternary linear code as*

$$\mathcal{C}^{2N} = \{(u|u+v) : u \in \mathcal{A}, v \in \mathcal{B}\}.$$

It is easy to see that if $\mathcal{G}_{\mathcal{A}}$ and $\mathcal{G}_{\mathcal{B}}$ are the generator matrices of $\mathcal{A}$ and $\mathcal{B}$ then the matrix

$$\mathcal{G}_P = \begin{pmatrix} \mathcal{G}_{\mathcal{A}} & \mathcal{G}_{\mathcal{A}} \\ 0 & \mathcal{G}_{\mathcal{B}} \end{pmatrix}$$

is the generator matrix of the code $\mathcal{C}^{2N}$.

Proposition 1. *The quaternary code $\mathcal{C}^{2N}$ defined above is a quaternary linear code of type $(2N; \gamma, \delta)$ where $\gamma = \gamma_{\mathcal{A}} + \gamma_{\mathcal{B}}$, $\delta = \delta_{\mathcal{A}} + \delta_{\mathcal{B}}$, binary length $n = 4N$, size $2^{\gamma+2\delta}$ and minimun distance $d = \min\{2d_{\mathcal{A}}, d_{\mathcal{B}}\}$.*

2.3 BQ-Plotkin Construction

Applying two Plotkin constructions, one after another but slightly changing the submatrices in the generator matrix we obtain a new construction with interesting properties regarding the minimum distance of the generated code. We call this new construction *BQ-Plotkin construction.*

Let $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ be any three quaternary linear codes of types $(N; \gamma_{\mathcal{A}}, \delta_{\mathcal{A}})$, $(N; \gamma_{\mathcal{B}}, \delta_{\mathcal{B}})$, $(N; \gamma_{\mathcal{C}}, \delta_{\mathcal{C}})$ and minimum distances $d_{\mathcal{A}}$, $d_{\mathcal{B}}$, $d_{\mathcal{C}}$ respectively.

Definition 2 (BQ-Plotkin Construction). *Let $\mathcal{G}_{\mathcal{A}}$, $\mathcal{G}_{\mathcal{B}}$ and $\mathcal{G}_{\mathcal{C}}$ be the generator matrices of the quaternary linear codes $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$. We define a new code $\mathcal{C}^{4N}$ as the quaternary linear code generated by*

$$\mathcal{G}_{BQ} = \begin{pmatrix} \mathcal{G}_{\mathcal{A}} & \mathcal{G}_{\mathcal{A}} & \mathcal{G}_{\mathcal{A}} & \mathcal{G}_{\mathcal{A}} \\ 0 & \mathcal{G}'_{\mathcal{B}} & 2\mathcal{G}'_{\mathcal{B}} & 3\mathcal{G}'_{\mathcal{B}} \\ 0 & 0 & \hat{\mathcal{G}}_{\mathcal{B}} & \hat{\mathcal{G}}_{\mathcal{B}} \\ 0 & 0 & 0 & \mathcal{G}_C \end{pmatrix},$$

where $\mathcal{G}'_{\mathcal{B}}$ is the matrix obtained from $\mathcal{G}_{\mathcal{B}}$ after switching twos by ones in their $\gamma_{\mathcal{B}}$ rows of order two and $\hat{\mathcal{G}}_{\mathcal{B}}$ is the matrix obtained from $\mathcal{G}_{\mathcal{B}}$ after removing their $\gamma_{\mathcal{B}}$ rows of order two.

Proposition 2. *The quaternary linear code generated by the BQ-Plotkin construction in Definition 2 is a quaternary code of type* $(4N; \gamma, \delta)$ *where* $\gamma = \gamma_{\mathcal{A}} + \gamma_{\mathcal{C}}$, $\delta = \delta_{\mathcal{A}} + \gamma_{\mathcal{B}} + 2\delta_{\mathcal{B}} + \delta_{\mathcal{C}}$, *binary length* $n = 8N$, *size* $2^{\gamma+2\delta}$ *and minimum distance* $d = \min\{4d_{\mathcal{A}}, 2d_{\mathcal{B}}, d_{\mathcal{C}}\}$.

3 Quaternary Reed-Muller Codes

The usual linear binary RM family of codes is one of the oldest and interesting family of codes. The codes in this family are easy to decode and their combinatorial properties are of great interest to produce new optimal codes from that.

For any integer $m \geq 1$ the family of binary linear RM codes is given by the sequence $RM(r, m)$, where $0 \leq r \leq m$; $RM(r, m)$ is called the r^{th} order binary Reed-Muller code of length $n = 2^m$ and

$$RM(0, m) \subset RM(1, m) \subset \cdots \subset RM(r-2, m) \subset RM(r-1, m) \subset RM(r, m).$$

Let $0 < r < m$, $m \geq 1$ and use the symbols $\mathbf{0}$, $\mathbf{1}$ for the all zeroes and the all ones vectors, respectively. According [12] the $RM(r, m)$ code of order r can be constructed by using the Plotkin construction in the following way:

$$\begin{array}{l} 1.\ RM(0, m) = \{\mathbf{0}, \mathbf{1}\}, RM(m, m) = \mathbb{F}^{2^m}, \\ 2.\ RM(r, m) = \{(u|u+v) : u \in RM(r, m-1),\ v \in RM(r-1, m-1)\}. \end{array} \tag{2}$$

It is important to note that if we fix m, once we know the sequence $RM(r, m)$ for all $0 \leq r \leq m$, then it is easy to obtain the new sequence $RM(r, m+1)$ using the Plotkin construction (2).

Codes in the RM family fulfil the basic properties summarized in the following Theorem (see [12]):

Theorem 1. *The binary linear Reed-Muller family of codes* $RM(r, m)$ *has the following properties:*

1. *length* $n = 2^m$;
2. *minimum distance* $d = 2^{m-r}$, $0 \leq r \leq m$;
3. *dimension* $k = \sum_{i=0}^{r} \binom{m}{i}$;
4. *each code* $RM(r-1, m)$ *is a subcode of* $RM(r, m)$, $r > 0$. $RM(0, m) = \{\mathbf{0}, \mathbf{1}\}$; $RM(m, m) = \mathbb{F}^{2^m}$ *and* $RM(r-1, m)$ *is the even code (so the code with all the vectors of even weight from* $\mathbb{F}^{2^m}$*);*
5. $RM(1, m)$ *is the binary linear Hadamard code and* $RM(r-2, m)$ *is the extended* 1*-perfect Hamming code of parameters* $(2^m, 2^m - m - 1, 4)$*;*
6. *the code* $RM(r, m)$ *is the dual code of* $RM(m-1-r, m)$ *for* $r < m$.

In the recent literature several families of quaternary linear codes has been proposed and studied [7,18,2,3] trying to generalize the RM codes, but when take the corresponding $\mathbb{Z}_4$-linear codes they do not satisfy all the above properties.

This is the main goal of the present work, to construct new families of quaternary linear codes such that, after the Gray map, we obtain $\mathbb{Z}_4$-linear codes with the parameters and properties quoted in Theorem 1 except for the duality.

We will refer to the quaternary linear Reed-Muller codes as $\mathcal{RM}$ to distinguish them from the binary linear Reed-Muller codes RM. Contrary to the linear binary case, where there is only one RM family, in the quaternary case we have $\lfloor \frac{m+1}{2} \rfloor$ families for each value of m. We will distinguish the families we are talking about by using subindexes s ($s \in \{0, \ldots, \lfloor \frac{m-1}{2} \rfloor\}$).

3.1 The Family of $\mathcal{RM}(r, 1)$ Codes

We start by considering the case of $m = 1$, so the case of codes of binary length $n = 2^1$. The quaternary linear Reed-Muller code $\mathcal{RM}(0, 1)$ is the repetition code with only one nonzero codeword (the vector with only one quaternary coordinate of value 2). This quaternary linear code is of type $(1; 1, 0)$. The code $\mathcal{RM}(1, 1)$ is the whole space $\mathbb{Z}_4^1$, so a quaternary linear code of type $(1; 0, 1)$.

These codes, $\mathcal{RM}(0, 1)$ and $\mathcal{RM}(1, 1)$, after the Gray map, give binary codes with the same parameters of the corresponding binary $RM(r, 1)$ codes and with the same properties described in Theorem 1. In this case, when $m = 1$, not only these codes have the same parameters, but they have the same codewords. We will refer to these codes as $\mathcal{RM}_0(0, 1)$ and $\mathcal{RM}_0(1, 1)$, respectively.

From now on and because we will need an specific representation for the above mentioned codes we will agree in to use the following matrices as the generator matrices for each one of them. The generator matrix of $\mathcal{RM}_0(0, 1)$ is $\mathcal{G}_0(0, 1) = \begin{pmatrix} 2 \end{pmatrix}$ and the generator matrix of $\mathcal{RM}_0(1, 1)$ is $\mathcal{G}_0(1, 1) = \begin{pmatrix} 1 \end{pmatrix}$.

3.2 Plotkin and BQ-Plotkin Constructions

The first important point is to apply the Plotkin construction to quaternary linear Reed-Muller codes.

Let $\mathcal{RM}_s(r, m-1)$ and $\mathcal{RM}_s(r-1, m-1)$, $0 \leq s \leq \lfloor \frac{m-1}{2} \rfloor$, be any two $\mathcal{RM}$ codes with parameters $(N; \gamma', \delta')$ and $(N; \gamma'', \delta'')$; binary length $n = 2^{m-1}$; number of codewords $2^{k'}$ and $2^{k''}$; minimum distance 2^{m-r-1} and 2^{m-r} respectively, where

$$k' = \sum_{i=0}^{r} \binom{m-1}{i}, \quad k'' = \sum_{i=0}^{r-1} \binom{m-1}{i}.$$

Using Proposition 1 we can prove the following result:

Theorem 2. *For any r and $m \geq 2$, $0 < r < m$, the code obtained by using the Plotkin construction:*

$$\mathcal{RM}_s(r, m) = \{(u|u+v) : u \in \mathcal{RM}_s(r, m-1),\ v \in \mathcal{RM}_s(r-1, m-1)\}$$

is a quaternary linear code of type $(2N; \gamma, \delta)$, where $\gamma = \gamma' + \gamma''$ and $\delta = \delta' + \delta''$; binary length $n = 2^m$; number of codewords 2^k, where $k = \sum_{i=0}^{r} \binom{m}{i}$, minimum distance 2^{m-r} and $\mathcal{RM}_s(r-1, m) \subset \mathcal{RM}_s(r, m)$.

For $r = 0$, $\mathcal{RM}_s(0, m)$ is the repetition code with only one nonzero codeword (the all twos vector). For $r = m$, the code $\mathcal{RM}_s(m, m)$ is the whole space $\mathbb{Z}_4^{(2^{m-1})}$.

Applying Theorem 2 and the above mentioned codes $\mathcal{RM}_0(r, m)$ with $m = 1$ we obtain the codes in Table 1a. The generator matrices for these codes are:

$$\mathcal{RM}_0(0,2) : \left(\underline{2\ 2}\right);\ \mathcal{RM}_0(1,2) : \begin{pmatrix} 0\ 2 \\ 1\ 1 \end{pmatrix}; \mathcal{RM}_0(2,2) : \left(\overline{\begin{matrix} 0\ 1 \\ 1\ 1 \end{matrix}}\right).$$

Table 1. $\mathcal{RM}_s(r, m)$ codes for (a: $m = 2$ and b: $m = 3$)

N	(r,m)	
	$(0,2)\ (1,2)\ (2,2)$	
	(γ, δ)	
2	$(1,0)\ (1,1)\ (0,2)$	$\mathcal{RM}_0(r,2)$

N	(r,m)	
	$(0,3)\ (1,3)\ (2,3)\ (3,3)$	
	(γ, δ)	
4	$(1,0)\ (2,1)\ (1,3)\ (0,4)$	$\mathcal{RM}_0(r,3)$
4	$(1,0)\ (0,2)\ (1,3)\ (0,4)$	$\mathcal{RM}_1(r,3)$

For $m = 3$ there exist two quaternary linear Hadamard codes. So, our goal is to find two families of quaternary Reed-Muller codes as it is shown in Table 1b.

Codes in the first row in Table 1b can be obtained using Plotkin construction from the codes in the first row of Table 1a. But, codes in the second row can not be obtained using only Plotkin constructions. It is in this point that we need to use the new BQ-Plotkin construction.

The constructions of additive codes whose images are binary codes with parameters of *RM* codes using the Plotkin construction were initiated in [15,17].

Let $\mathcal{RM}_{s-1}(r, m-2)$, $\mathcal{RM}_{s-1}(r-1, m-2)$ and $\mathcal{RM}_{s-1}(r-2, m-2)$, $0 < s \leq \lfloor \frac{m-1}{2} \rfloor$, $m \geq 3$, be any three $\mathcal{RM}$ codes with parameters $(N; \gamma', \delta')$, $(N; \gamma'', \delta'')$ and $(N; \gamma''', \delta''')$; binary length $n = 2^{m-2}$; number of codewords $2^{k'}$, $2^{k''}$ and $2^{k'''}$; minimum distances 2^{m-r-2}, 2^{m-r-1} and 2^{m-r} respectively, where

$$k' = \sum_{i=0}^{r} \binom{m-2}{i}, \quad k'' = \sum_{i=0}^{r-1} \binom{m-2}{i}, \quad k'' = \sum_{i=0}^{r-2} \binom{m-2}{i}.$$

Using Proposition 2 we are able to prove

Theorem 3. *For any r and $m \geq 3$, $0 < r < m-1$, the code $\mathcal{RM}_s(r, m)$, $s > 0$, obtained by using the BQ-Plotkin construction and with generator matrix $\mathcal{G}_s(r, m)$:*

$$\begin{pmatrix} \mathcal{G}_{s-1}(r, m-2) & \mathcal{G}_{s-1}(r, m-2) & \mathcal{G}_{s-1}(r, m-2) & \mathcal{G}_{s-1}(r, m-2) \\ 0 & \mathcal{G}'_{s-1}(r-1, m-2) & 2\mathcal{G}'_{s-1}(r-1, m-2) & 3\mathcal{G}'_{s-1}(r-1, m-2) \\ 0 & 0 & \hat{\mathcal{G}}_{s-1}(r-1, m-2) & \hat{\mathcal{G}}_{s-1}(r-1, m-2) \\ 0 & 0 & 0 & \mathcal{G}_{s-1}(r-2, m-2) \end{pmatrix}$$

is a quaternary linear code of type (4N; γ, δ), where $\gamma = \gamma' + \gamma'''$ and $\delta = \delta' + \gamma'' + 2\delta'' + \delta'''$; binary length $n = 2^m$; number of codewords 2^k, where $k = \sum_{i=0}^{r} \binom{m}{i}$, minimum distance 2^{m-r} and $\mathcal{RM}_s(r-1,m) \subset \mathcal{RM}_s(r,m)$.

To be coherent with all the notations, for $r = -1$, the code $\mathcal{RM}_s(-1,m)$ is defined as the all zero codeword code. For $r = 0$, the code $\mathcal{RM}_s(0,m)$ is defined as the repetition code with only one non zero codeword (the all twos quaternary vector). For $r = m-1$ and $r = m$, the codes $\mathcal{RM}_s(m-1,m)$ and $\mathcal{RM}_s(m,m)$ are defined as the even Lee weight code and the whole space $\mathbb{Z}_4^{(2^{m-1})}$, respectively.

Using both Theorems 2 and 3 we can construct the $\mathcal{RM}$ codes in two rows of Table 1b. We do not write the generator matrices for codes $\mathcal{RM}_0(r,3)$ because they can be directly obtained from the respective codes for $m = 2$ by using the Plotkin construction. For the codes in the family $\mathcal{RM}_3(r,3)$ we present the generator matrices as a direct application of Theorem 3:

$$\mathcal{RM}_3(0,3) : \left(\boxed{2\ 2\ 2\ 2}\right);\ \mathcal{RM}_3(1,3) : \left(\begin{array}{|cccc|}\hline 1&1&1&1\\ 0&1&2&3\\ \hline\end{array}\right);\ \mathcal{RM}_3(2,3) : \left(\begin{array}{|cccc|}\hline 2&0&0&0\\ \hline 1&1&1&1\\ 0&1&2&3\\ 0&0&1&1\\ \hline\end{array}\right)$$

the remaining code $\mathcal{RM}_3(3,3)$ in the family is the whole space $\mathbb{Z}_4^{(2^2)}$.

All these codes, after the Gray map, give binary codes with the same parameters as the $RM(r,3)$ codes and with the same properties described in Theorem 1. In this case under consideration, when $m = 3$, like in the case $m = 2$ not only these codes have the same parameters, but they have the same codewords. This is not in this way for all the other values of $m > 3$.

Now, from Table 1b and by using the Plotkin construction we can construct the two families of $\mathcal{RM}_s(r,4)$ codes for $s = 0, 1$, as it shown in Table 2.

Table 2. $\mathcal{RM}_s(r,m)$ codes for $m = 4$

	(r,m)	
	$(0,4)$ $(1,4)$ $(2,4)$ $(3,4)$ $(4,4)$	
N	(γ,δ)	
8	$(1,0)$ $(3,1)$ $(3,4)$ $(1,7)$ $(0,8)$	$\mathcal{RM}_0(r,4)$
8	$(1,0)$ $(1,2)$ $(1,5)$ $(1,7)$ $(0,8)$	$\mathcal{RM}_1(r,4)$

From codes in Table 1b and Table 2 applying the BQ-Plotkin and the Plotkin constructions, respectively, we can construct the three families of $\mathcal{RM}_s(r,5)$ for $s = 0, 1, 2$, as it is shown in Table 3.

As it is proved in Theorems 2 and 3 the constructed families of $\mathcal{RM}$ codes satisfy the same properties we stated for linear binary Reed-Muller codes in Theorem 1 except for the duality.

Table 3. $\mathcal{RM}_s(r,m)$ codes for $m=5$

	(r,m)						
	$(0,5)$	$(1,5)$	$(2,5)$	$(3,5)$	$(4,5)$	$(5,5)$	
N	(γ,δ)						
16	$(1,0)$	$(4,1)$	$(6,5)$	$(4,11)$	$(1,15)$	$(0,16)$	$\mathcal{RM}_0(r,5)$
16	$(1,0)$	$(2,2)$	$(2,7)$	$(2,12)$	$(1,15)$	$(0,16)$	$\mathcal{RM}_1(r,5)$
16	$(1,0)$	$(0,3)$	$(2,7)$	$(0,13)$	$(1,15)$	$(0,16)$	$\mathcal{RM}_2(r,5)$

Notice that the constructed $\mathcal{RM}$ families of quaternary linear Reed-Muller codes have not only the same parameters as the classical binary linear family of RM codes, but the characteristic codes $\mathcal{RM}_s(1,m)$ and $\mathcal{RM}_s(m-2,m)$ satisfy the following Lemma.

Lemma 1. *For any integer $m \geq 1$ and $0 \leq s \leq m$, the code $\mathcal{RM}_s(1,m)$ is a Hadamard quaternary linear code and the code $\mathcal{RM}_s(m-2,m)$ is an extended quaternary linear 1-perfect code.*

4 Conclusion

New constructions based on quaternary linear codes has been proposed such that, after the Gray map, the obtained $\mathbb{Z}_4$-linear codes fulfil the same properties and characteristics as the usual binary linear RM codes. Apart from the parameters characterizing each code an important property which remains in these new presented families is that the first order $\mathcal{RM}$ code is a Hadamard quaternary code and the $(m-2)$-th order $\mathcal{RM}$ code is a quaternary code which give rise to an extended 1-perfect code, like in the usual binary case. So the families of codes obtained in the paper contain the families of quaternary perfect and Hadamard codes from [9,10].

There are several questions and subjects related to this work where would be of great interest to go deeply. The first one is the generalization of the constructions of $\mathcal{RM}$ codes to the case of general additive codes, so the case of additive codes with $\alpha \neq 0$. It is known that there exist additive non $\mathbb{Z}_4$-linear 1-perfect codes [5] and the corresponding Hadamard additive dual codes. This observation could be taken as the starting point to produce the new families of Reed-Muller codes. Another important question is duality. It is well known that binary codes $RM(r,m)$ and $RM(m-r-1,m)$ are dual each other. The constructed $\mathcal{RM}$ families have a similar, but not exactly the same, property. Code $\mathcal{RM}(m-r-1,m)$ is equivalent, but not equal, to the additive dual of the code $\mathcal{RM}(r,m)$. Given any $\mathcal{RM}$ family would be interesting to find the dual family, in the sense that all the codes in the first family have the additive dual in the second family. Other open questions are related to uniqueness (up to equivalence) of obtained codes, their weight distribution, etc.

References

1. Bonnecaze, A., Solé, P., Calderbank, A.R.: Quaternary Quadratic Residue Codes and Unimodular Lattices. IEEE Trans. Inform. Theory 41, 366–377 (1995)
2. Borges, J., Fernandes, C., Phelps, K.T.: Quaternary Reed-Muller Codes. IEEE Trans. Inform. Theory 51(7), 2686–2691 (2005)
3. Borges, J., Fernandes, C., Phelps, K.T.: ZRM Codes. IEEE Trans. Inform. Theory (to appear)
4. Borges, J., Fernández, C., Pujol, J., Rifà, J., Villanueva, M.: On $\mathbb{Z}_2\mathbb{Z}_4$-Linear Codes and Duality. In: V Jornades de Matemàtica Discreta i Algorísmica, Soria, Spain, pp. 171–177 (2006)
5. Borges, J., Rifà, J.: A Characterization of 1-Perfect Additive Codes. IEEE Trans. Inform. Theory 45(5), 1688–1697 (1999)
6. Delsarte, P.: An Algebraic Approach to the Association Schemes of Coding Theory. Philips Research Rep. Suppl. 10 (1973)
7. Hammons, A.R., Kumar, P.V., Calderbank, A.R., Sloane, N.J.A., Solé, P.: The $\mathbb{Z}_4$-Linearity of Kerdock, Preparata, Goethals and Related Codes. IEEE Trans. Inform. Theory 40, 301–319 (1994)
8. Hou, X-D., Lahtonen, J.T., Koponen, S.: The Reed-Muller Code $R(r, m)$ Is Not $\mathbb{Z}_4$-Linear for $3 \leq r \leq m-2$. IEEE Trans. Inform. Theory 44, 798–799 (1998)
9. Krotov, D.S.: $\mathbb{Z}_4$-Linear Perfect Codes. Discrete Analysis and Operation Research, Novosibirsk, Institute of Math. SB RAS 7(4), 78–90 (2000)
10. Krotov, D.S.: $\mathbb{Z}_4$-Linear Hadamard and Extended Perfect Codes. In: 2001 Int. Workshop on Coding and Cryptography, Paris, France, pp. 329–334 (2001)
11. Lee, C.Y.: Some Properties of Nonbinary Error-Correcting Codes. IRE Trans. Inform. Theory 4(4), 77–82 (1958)
12. MacWilliams, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes. North-Holland Publishing Company, Amsterdam (1977)
13. Nechaev, A.A.: Kerdock Codes in a Cyclic Form. Disc. Math. 1(4), 123–139 (1989)
14. Plotkin, M.: Binary Codes with Specified Minimum Distances. IEEE Trans. Inform. Theory 6, 445–450 (1960)
15. Pujol, J., Rifà, J.: Additive Reed-Muller pCodes. In: 1997 Int. Symp. on Inform. Theory, Ulm, Germany, p. 508. IEEE Press, NewYork (1997)
16. Rifà, J., Pujol, J.: Translation Invariant Propelinear Codes. IEEE Trans. Inform. Theory 43, 590–598 (1997)
17. Solov'eva, F.I.: On Z4-Linear Codes with Parameters of Reed-Muller Codes. Problems of Inform. Trans. 43, 32–38 (2007)
18. Wan, Z.X.: Quaternary codes. World Scientific Publishing Co., Singapore (1997)

Joint Source-Cryptographic-Channel Coding Based on Linear Block Codes

Haruhiko Kaneko and Eiji Fujiwara

Graduate School of Information Science and Engineering,
Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku, Tokyo, 152-8552 Japan
hkaneko@fuji.cs.titech.ac.jp, fujiwara@cs.titech.ac.jp

Abstract. This paper proposes a joint coding with three functions: source coding, channel coding, and public-key encryption. A codeword is simply generated as a product of an encoding matrix and a sparse information word. This encoding method has much lower encoding complexity than the conventional coding techniques in which source coding, encryption, and channel coding are successively applied to an information word. The encoding matrix is generated by using two linear error control codes and randomly generated nonsingular matrices. Encryption is based on the intractableness of factorizing a matrix into randomly constructed factor matrices, and of decoding an error control code defined by a random parity-check matrix. Evaluation shows that the proposed joint coding gives a lower bit error rate and a superior compression ratio than the conventional codings.

1 Introduction

Compact communication devices will play an important role in future network systems such as sensor networks and ubiquitous computing networks. For an efficient and reliable data transmission, these devices should have source and channel coding capabilities. In addition, data should be encrypted when such devices are used in an insecure environment. Many data compression techniques are available for efficient source coding [1][2][3][4], and also strong error control codes have been developed for channel coding [5][6][7]. In addition, some encryption algorithms have been standardized for secure data transmission. Recent source and channel codings and encryption algorithms require considerable computational power for encoding and decoding. Compact communication devices, however, usually have limited computational resources. Therefore, low-complexity joint source-cryptographic-channel coding is preferable for such resource constrained devices.

Techniques for joint source-channel coding have been proposed aimed at decoding noisy compressed data as reliably as possible. Unequal error protection (UEP) coding can be used to protect important parts of compressed data, such as header information, from errors. UEP coding techniques have been proposed for several types of compressed data, such as compressed text [8] and video data [9].

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 158–167, 2007.
© Springer-Verlag Berlin Heidelberg 2007

These coding techniques, however, generally increase the encoding complexity. Another approach to this joint coding is to utilize source dependencies for a belief propagation decoding algorithm of low-density generator-matrix code [10]. MacKay-Neal (MN) Code [6] can also be used for the joint source-channel coding that efficiently encodes a sparse information word $\mathbf{m}$ to codeword $\mathbf{c} = \mathbf{Am}$, where $\mathbf{m}$ and $\mathbf{c}$ are column vectors, and $\mathbf{A}$ is a matrix generated from a low-density parity-check (LDPC) matrix. The MN code provides joint source-channel coding capability with simple exclusive-OR (EX-OR) operations in encoding. The above joint codings, however, do not have encryption capabilities.

This paper proposes a joint coding with three functions: source coding, channel coding, and public-key encryption. The proposed joint coding encodes a sparse information word $\mathbf{m}$ to a codeword $\mathbf{c} = \mathbf{Am}$, where $\mathbf{A}$ is defined as the product of a generator matrix, a parity-check matrix, and randomly constructed nonsingular matrices. Hence, the codeword $\mathbf{c}$ can be generated from the information word $\mathbf{m}$ by simple EX-OR operations, which means that the encoding complexity of the proposed joint coding is comparable to that of the conventional linear error control codes encoded by generator matrix or systematic parity-check matrix.

This paper is organized as follows. Section 2 reviews the related work of the joint coding. Section 3 presents a system model, and Section 4 shows a new joint coding under this model. Sections 5 and 6 demonstrate the security and the entropy conversion of this coding, respectively. Section 7 provides an evaluation of the proposed joint coding, and Section 8 concludes the paper.

2 Related Work

2.1 MN Code for Joint Source-Channel Coding

MN code [6] has been proposed for joint source-channel coding. Let $\mathbf{m} = (m_0, m_1, \ldots, m_{K-1})^T$ be an information word of length K bits, where m_i, $i \in \{0, 1, \ldots, K-1\}$, is the i-th information bit. In general, the conventional error control codes encode any input word $\mathbf{m}$ with arbitrary Hamming weight. On the other hand, MN code encodes sparse information word $\mathbf{m}$, i.e., the word with low Hamming weight. Let $\mathbf{H} = [\mathbf{C}_s|\mathbf{C}_n]$ be an $M \times N$ LDPC matrix, where $\mathbf{C}_s$ is an $M \times (N-M)$ matrix, and $\mathbf{C}_n$ is an $M \times M$ nonsingular matrix. Information word $\mathbf{m}$ with length $K = N - M$ bits is encoded as $\mathbf{c} = \left(\mathbf{C}_n^{-1}\mathbf{C}_s\right)\mathbf{m} = \mathbf{Am}$, where $\mathbf{C}_n^{-1}\mathbf{C}_s = \mathbf{A}$, the probability of information bit m_i being 1 is $q_1 < 1/2$, the matrix $[\mathbf{C}_s|\mathbf{C}_n]$ is an LDPC matrix over binary symmetric channel (BSC) with crossover probability $\epsilon = q_1$, and $\mathbf{c} = (c_0, c_1, \ldots, c_{M-1})^T$ is a codeword of length M bits. Let $\mathbf{c}' = \mathbf{c} + \mathbf{n}$ be a received word, where vector addition is performed over GF(2) and $\mathbf{n}$ is a noise vector of length M. The received word $\mathbf{c}'$ is decoded based on the following relation:

$$\mathbf{C}_n\mathbf{c}' = \mathbf{C}_n\mathbf{c} + \mathbf{C}_n\mathbf{n} = \mathbf{C}_s\mathbf{m} + \mathbf{C}_n\mathbf{n} = [\mathbf{C}_s|\mathbf{C}_n]\begin{pmatrix}\mathbf{m}\\ \mathbf{n}\end{pmatrix}.$$

From this, the information word $\mathbf{m}$ can be generated from $\mathbf{C}_n\mathbf{c}'$ by the sum-product algorithm [6] because $\mathbf{m}$ and $\mathbf{n}$ are sparse vectors and $\mathbf{H} = [\mathbf{C}_s|\mathbf{C}_n]$ is an LDPC matrix.

2.2 McEliece's Public-Key Cryptosystem (PKC) Using Linear Block Code

McEliece has proposed a PKC based on Goppa code [11]. This McEliece's PKC based on LDPC code has been proposed in [12]. Let $\mathbf{G}$ be a $K \times N$ generator matrix of $(N, K, 2t+1)$ linear code $\boldsymbol{C}$, where $\boldsymbol{C}$ is a random t-bit error correcting code. Let $\mathbf{Q}$ be an $N \times N$ random permutation matrix, and let $\mathbf{D}$ be a $K \times K$ random nonsingular matrix. Using the matrices $\mathbf{G}, \mathbf{Q}$, and $\mathbf{D}$ as a set of private-keys, the public-key $\mathbf{A}$ is generated as $\mathbf{A} = \mathbf{Q}\mathbf{G}^T\mathbf{D}$, where $\mathbf{A}$ is an $N \times K$ matrix. Binary plaintext $\mathbf{m} = (m_0, m_1, \ldots, m_{K-1})^T$ is encrypted using the public-key as $\mathbf{c} = \mathbf{A}\mathbf{m} + \mathbf{n}$, where $\mathbf{c}$ is a ciphertext expressed by a binary column vector of length N, and $\mathbf{n}$ is a random error vector with length N and Hamming weight t. The ciphertext $\mathbf{c}$ is decrypted by using the private-keys as follows:

1. Calculate $\mathbf{c}' = \mathbf{Q}^{-1}\mathbf{c} = \mathbf{G}^T\mathbf{D}\mathbf{m} + \mathbf{Q}^{-1}\mathbf{n}$.
2. Decode $\mathbf{c}'$ using the linear code $\boldsymbol{C}$ to correct errors $\mathbf{Q}^{-1}\mathbf{n}$, and generate the decoded word $\mathbf{u} = \mathbf{D}\mathbf{m}$.
3. Reconstruct the plaintext as $\mathbf{m} = \mathbf{D}^{-1}\mathbf{u}$.

Although McEliece's PKC is vulnerable to some practical attacks, modified versions of this PKC have proven to be semantically secure [13].

2.3 Niederreiter's PKC for Joint Source-Cryptographic Coding

Niederreiter's PKC [14] is also based on linear block error control codes. Unlike McEliece's PKC, which can encrypt a plaintext $\mathbf{m}$ with arbitrary Hamming weight, Niederreiter's PKC can only encrypt $\mathbf{m}$ with Hamming weight less than or equal to t. Let $\mathbf{H}$ be an $M \times N$ parity-check matrix of t-symbol error correcting code $\boldsymbol{C}$ over GF(q), such as Reed-Solomon code. Let $\mathbf{T}$ be an $N \times N$ random permutation matrix, and let $\mathbf{D}$ be an $M \times M$ random nonsingular matrix. Using the matrices $\mathbf{H}, \mathbf{T}$, and $\mathbf{D}$ as a set of private-keys, the public-key $\mathbf{A}$ is generated as $\mathbf{A} = \mathbf{D}\mathbf{H}\mathbf{T}$, where $\mathbf{A}$ is an $M \times N$ matrix. Information word $\mathbf{m} = (m_0, m_1, \ldots, m_{N-1})^T$ is encrypted by using the public-key as $\mathbf{c} = \mathbf{A}\mathbf{m}$, where the Hamming weight of $\mathbf{m}$ is less than or equal to t, and $\mathbf{c}$ is the ciphertext expressed by a column vector of length M. Note that deriving $\mathbf{m}$ directly from $\mathbf{A}$ and $\mathbf{c}$ is difficult because $\mathrm{rank}(\mathbf{A}) < N$ and $\mathbf{A}$ has no visible algebraic structure. The ciphertext $\mathbf{c}$ is decrypted using the private-keys as follows:

1. Calculate $\mathbf{c}' = \mathbf{D}^{-1}\mathbf{c} = \mathbf{H}\mathbf{T}\mathbf{m}$.
2. Find a column vector $\mathbf{u}$ that satisfies $\mathbf{c}' = \mathbf{H}\mathbf{u}$ and $w(\mathbf{u}) \leq t$ using a decoding algorithm for $\boldsymbol{C}$, where $w(\mathbf{u})$ is the Hamming weight of $\mathbf{u}$.
3. Reconstruct the plaintext as $\mathbf{m} = \mathbf{T}^{-1}\mathbf{u}$.

A security analysis has shown that McEliece's and Niederreiter's PKCs have equivalent security [15].

Table 1 summarizes the functions of the above coding techniques. Here, MN code and Niederreiter's PKC have a source coding function because sparser information word $\mathbf{m}$ gives shorter codeword $\mathbf{c}$.

Table 1. Functions of conventional coding techniques

Function	MN code	McEliece's PKC	Niederreiter's PKC
Source coding	Yes	No	Yes
Encryption	No	Yes	Yes
Channel coding	Yes	No	No

3 System Model

Recent communication and storage systems sometimes require three functions: source coding for data compression, cryptographic coding for data encryption, and channel coding for error correction/detection. Figure 1 (a) shows the conventional sequential encoding process for source, cryptographic, and channel codings, each performed independently, where the source coding contains preprocessing and entropy coding steps. The preprocessing is dependent on the type of input data. For example, still images are preprocessed by discrete cosine transform, quantization, zigzag scan, and run length coding [3]. Video data are first applied motion estimation/compensation, and then estimation errors are encoded in a similar way to still image coding [4]. Text data are usually preprocessed by dictionary coding [2] or block sorting. The preprocessing is usually followed by an entropy coding step, such as Huffman coding [1] and arithmetic coding. The compressed data is encrypted and then encoded by a channel code.

This paper proposes a new joint coding shown in Fig. 1 (b), where the conventional entropy, cryptographic, and channel codings are replaced by an entropy conversion and a joint coding based on an encoding matrix **A**. Unlike conventional PKC based on integer factoring problem or discrete logarithm problem which require many arithmetic operations, the proposed coding provides a PKC with simple EX-OR operations. In addition, this paper demonstrates in Section 7 that, for some cases, the joint coding provides superior data compression ratio and higher error correction capability than the conventional sequential coding.

This paper mainly focuses on the joint coding. The entropy conversion is briefly described in Section 6.

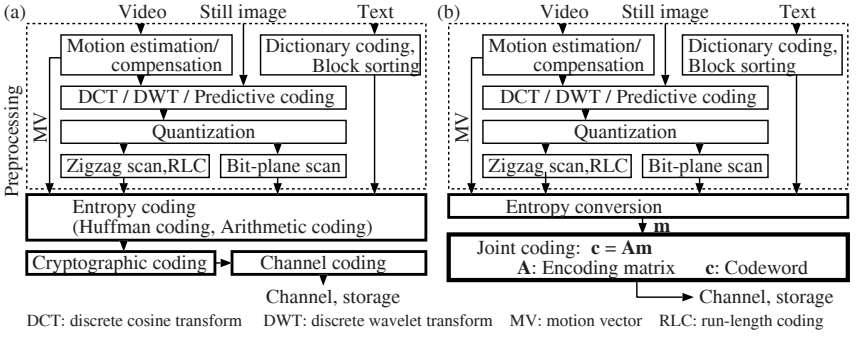


Fig. 1. (a) Conventional sequential coding. (b) Proposed joint coding.

4 Joint Source-Cryptographic-Channel Coding

This section demonstrates a joint source-cryptographic-channel coding that encodes a sparse binary information word $\mathbf{m} = (m_0, m_1, \ldots, m_{N_S-1})^T$ with Hamming weight t to a binary codeword $\mathbf{c} = (c_0, c_1, \ldots, c_{N_C-1})^T$, where the probability of $m_i, i \in \{0, 1, \ldots, N_S - 1\}$, being 1 is $q_1 = t/N_S < 1/2$. Here, the joint coding has source coding capability because the code rate N_S/N_C is determined based on both the source entropy $H(S) = -q_0 \log q_0 - q_1 \log q_1$ and the channel capacity $\mathcal{C}$, where $q_0 = 1 - q_1$.

4.1 Code Construction and Encoding

Let $\mathbf{H}_S$ be an $M_S \times N_S$ parity-check matrix of either a t-bit error correcting Goppa code or an LDPC code over BSC with crossover probability $\epsilon = q_1$. Let $\mathbf{H}_C$ be an $M_C \times N_C$ parity-check matrix of linear error correcting code $\boldsymbol{C}_C$ designed for a given communication channel C, where $K_C = N_C - M_C = M_S$. Generator matrix $\mathbf{G}_C$ for $\boldsymbol{C}_C$ is expressed as a binary $K_C \times N_C$ matrix. Square matrices $\mathbf{D}, \mathbf{Q}$, and $\mathbf{T}$ are defined as follows: $\mathbf{D}$ is an $M_S \times M_S$ random nonsingular matrix, $\mathbf{Q}$ is an $N_C \times N_C$ random permutation matrix, and $\mathbf{T}$ is an $N_S \times N_S$ random permutation matrix. The encoding matrix $\mathbf{A}$ is generated as follows:

$$\mathbf{A} = \mathbf{Q}^{-1}\mathbf{G}_C^T\mathbf{D}^{-1}\mathbf{H}_S\mathbf{T}^{-1},$$

where $\mathbf{A}$ is an $N_C \times N_S$ matrix. The matrix $\mathbf{A}$ is a public-key for encoding, and the other matrices are private-keys for decoding. Figure 2 illustrates how the encoding matrix $\mathbf{A}$ is generated.

Using the matrix $\mathbf{A}$, binary sparse information word $\mathbf{m} = (m_0, m_1, \ldots, m_{N_S-1})^T$ is encoded to a codeword $\mathbf{c} = (c_0, c_1, \ldots, c_{N_C-1})^T$ as $\mathbf{c} = \mathbf{A}\mathbf{m}$, where the probability of $m_i = 1, i \in \{0, 1, \ldots, N_S - 1\}$, is $q_1 < 1/2$.

4.2 Tandem Decoding

Let $\mathbf{c}' = (c'_0, c'_1, \ldots, c'_{N_C-1})^T = \mathbf{c} + \mathbf{n}$ be a received word, where $c'_i, i \in \{0, 1, \ldots, N_C - 1\}$, is an element of the channel output alphabet, and $\mathbf{n}$ is a noise vector

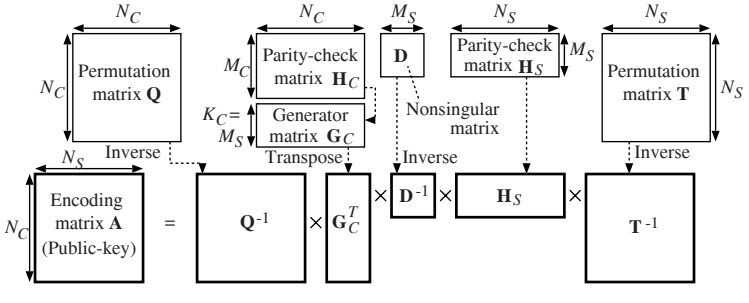


Fig. 2. Generation of encoding matrix $\mathbf{A}$

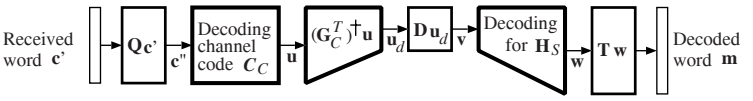


Fig. 3. Tandem decoding process

expressed as a column vector of length N_C. To reconstruct the original information word $\mathbf{m}$, the received word $\mathbf{c}'$ is decoded using the following relation:

$$\mathbf{c}' = \mathbf{c} + \mathbf{n} = \mathbf{Q}^{-1}\mathbf{G}_C^T\mathbf{D}^{-1}\mathbf{H}_S\mathbf{T}^{-1}\mathbf{m} + \mathbf{n}.$$

Firstly, the received word is permuted using $\mathbf{Q}$ as $\mathbf{c}'' = \mathbf{Q}\mathbf{c}'$. The column vector $\mathbf{c}''$ satisfies the following equation:

$$\mathbf{c}'' = \mathbf{Q}\mathbf{c}' = \mathbf{G}_C^T\mathbf{D}^{-1}\mathbf{H}_S\mathbf{T}^{-1}\mathbf{m} + \mathbf{Q}\mathbf{n}.$$

Since $\mathbf{G}_C^T\mathbf{D}^{-1}\mathbf{H}_S\mathbf{T}^{-1}\mathbf{m}$ is a codeword of $\boldsymbol{C}_C$, and $\mathbf{Q}\mathbf{n}$ is a permuted noise vector, errors in $\mathbf{c}''$ can be corrected by a decoding algorithm for $\boldsymbol{C}_C$. That is, the decoding removes the noise vector $\mathbf{Q}\mathbf{n}$ from the vector $\mathbf{c}''$. Then, $\mathbf{c}''$ turns out to be $\mathbf{u} = \mathbf{G}_C^T\mathbf{D}^{-1}\mathbf{H}_S\mathbf{T}^{-1}\mathbf{m}$. The generator matrix $\mathbf{G}_C^T$ is eliminated as

$$\mathbf{u}_d = (\mathbf{G}_C^T)^\dagger\mathbf{u} = \mathbf{D}^{-1}\mathbf{H}_S\mathbf{T}^{-1}\mathbf{m},$$

where $(\mathbf{G}_C^T)^\dagger\mathbf{G}_C^T = \mathbf{I}$. Here, $(\mathbf{G}_C^T)^\dagger$ is the $K_C \times N_C$ matrix generated by the method shown in [16]. Then, the column vector $\mathbf{v}$ is calculated as

$$\mathbf{v} = \mathbf{D}\mathbf{u}_d = \mathbf{H}_S\mathbf{T}^{-1}\mathbf{m} = \mathbf{H}_S\mathbf{w},$$

where $\mathbf{T}^{-1}\mathbf{m} = \mathbf{w}$. Since $\mathbf{w}$ is a sparse vector with Hamming weight t and $\mathbf{H}_S$ is a parity-check matrix of either a t-bit error correcting Goppa code or an LDPC code for BSC with $\epsilon = t/N_S$, $\mathbf{w}$ can be derived from $\mathbf{v}$ by using a decoding algorithm for $\mathbf{H}_S$. Finally, the original information word is reconstructed as $\mathbf{T}\mathbf{w} = \mathbf{T}\mathbf{T}^{-1}\mathbf{m} = \mathbf{m}$. Figure 3 illustrates the above successive decoding process, called tandem decoding.

4.3 Joint Decoding

If $\boldsymbol{C}_C$ is a systematic LDPC code, and also $\mathbf{H}_S$ and $\mathbf{D}$ are an LDPC matrix and a sparse nonsingular matrix, respectively, we can employ joint decoding using a three-layer Tanner graph, as shown in Fig.4(a). The top, middle, and bottom layers are the Tanner graphs for $\mathbf{H}_C$, $\mathbf{D}$, and $\mathbf{H}_S$, respectively, where the top and middle layers share K_C variable nodes (v-nodes), and the middle and bottom layers share $M_S = K_C$ check nodes (c-nodes). In comparison with the tandem decoding, the N_C v-nodes of the top layer correspond to $\mathbf{c}'' = \mathbf{Q}\mathbf{c}'$, the K_C v-nodes between the top and middle layers to $\mathbf{u}_d$, the M_S c-nodes between the middle and bottom layers to $\mathbf{v}$, and the N_S v-nodes of the bottom layer to $\mathbf{w} = \mathbf{T}^{-1}\mathbf{m}$. The permuted received word $\mathbf{c}'' = \mathbf{Q}\mathbf{c}'$ is decoded by the sum-product algorithm for Tanner graph shown in Fig. 4(b) whose node connections are identical to those in Fig. 4(a).

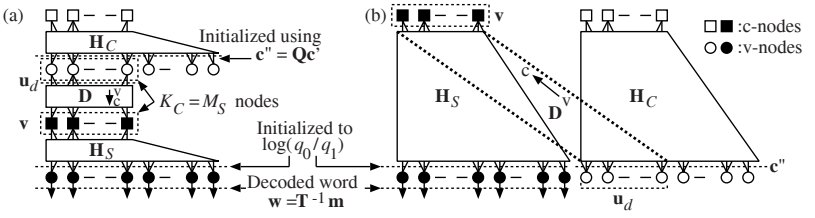


Fig. 4. (a) Three-layer Tanner graph for joint decoding. (b) Equivalent one-layer Tanner graph.

5 Security of the Proposed Joint Coding

Theorem 1. *The security of the proposed joint coding is equivalent to that of the Niederreiter's PKC.*

Proof. The encoding matrix of the Niederreiter's PKC is given as $\mathbf{A}_N = \mathbf{DHT}$, where $\mathbf{D}$ is a random nonsingular matrix, $\mathbf{H}$ a parity-check matrix, and $\mathbf{T}$ a random permutation matrix. The encoding matrix of the proposed joint coding is given as $\mathbf{A} = \mathbf{Q}^{-1}\mathbf{G}_C^T\mathbf{D}^{-1}\mathbf{H}_S\mathbf{T}^{-1}$, where $\mathbf{Q}$ and $\mathbf{T}$ are random permutation matrices, $\mathbf{G}_C$ is a generator matrix, $\mathbf{D}$ a random nonsingular matrix, $\mathbf{H}_S$ a parity-check matrix. By substituting $\mathbf{Q}^{-1}\mathbf{G}_C^T\mathbf{D}^{-1}$ by $\mathbf{D}'$, we have $\mathbf{A} = \mathbf{D}'\mathbf{H}_S\mathbf{T}^{-1}$, where $\mathbf{D}'$ is a random nonsingular matrix. From this, the Niederreiter's PKC and the proposed joint coding have equivalent security. □

It is proved that the Niederreiter's and the McEliece's PKCs have equivalent security [15], and hence the security of the proposed joint coding is equivalent to that of the McEliece's PKC. The security of the McEliece's PKC using Goppa code has been analyzed by Kobara and Imai [13], and they have shown that, without partial knowledge on the target plaintext or a decryption oracle, there exists no polynomial-time attack against ciphertext. Note that higher level of security, that is, indistinguishability against adaptive chosen-ciphertext attacks (IND-CCA2), can be achieved by appropriate preprocessings based on hash functions and random number generators [13].

6 Entropy Conversion

This section proposes an entropy conversion based on Huffman tree [1]. The following modified Huffman tree has a possibility to make an efficient entropy conversion. That is, compared to the conventional Huffman tree having only one edge type, this modified one has two edge types each determined according to the source symbol distribution. Figure 5(a) shows an example of the modified Huffman tree for 9-ary source symbols whose probability distribution is given by Fig. 5(b). The modified Huffman tree generates two binary output words: $\mathbf{m}_L$ of length N_S^L and $\mathbf{m}_H$ of length N_S^H, where $\mathbf{m}_L$ is a sparse word with $q_0 > q_1$, and $\mathbf{m}_H$ is a dense word (compared to $\mathbf{m}_L$) with $q_0 \simeq q_1$. Here, q_0 and q_1

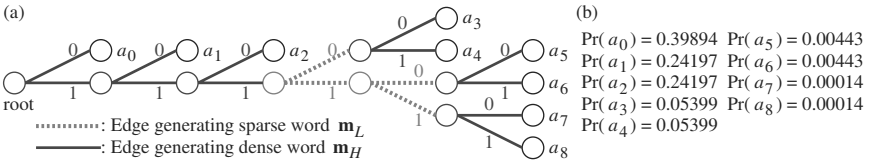


Fig. 5. (a) Modified Huffman tree. (b) Probability distribution of 9-ary source symbols.

are the probabilities of each bit having values 0 and 1, respectively. In Fig. 5, edges indicated by dotted lines generate $\mathbf{m}_L$, and those indicated by solid lines generate $\mathbf{m}_H$. Systematic construction of the tree is left to the future work.

In order to encode $\mathbf{m} = (\mathbf{m}_L, \mathbf{m}_H)^T$ by the encoding matrix $\mathbf{A}$, the matrices $\mathbf{H}_S$ and $\mathbf{T}$ in Section 4 are modified as follows:

$$\mathbf{H}_S = \left[\begin{array}{c|c} \mathbf{H}'_S & \mathbf{O} \\ \hline \mathbf{R} & \mathbf{D}' \end{array}\right], \quad \mathbf{T} = \left[\begin{array}{c|c} \mathbf{T}_L & \mathbf{O} \\ \hline \mathbf{O} & \mathbf{T}_H \end{array}\right],$$

where $\mathbf{H}'_S$ is an $M_S^L \times N_S^L$ parity-check matrix for compression of $\mathbf{m}_L$, $\mathbf{R}$ an $N_S^H \times N_S^L$ random matrix, $\mathbf{D}'$ an $N_S^H \times N_S^H$ random nonsingular matrix, $\mathbf{T}_L$ an $N_S^L \times N_S^L$ random permutation matrix, $\mathbf{T}_H$ an $N_S^H \times N_S^H$ random permutation matrix, and $\mathbf{O}$ a zero matrix. The remaining matrices of $\mathbf{H}_C, \mathbf{D}$, and $\mathbf{Q}$ are identical to those in Section 4.

7 Evaluation

This section evaluates the source and the channel coding capabilities of the proposed joint coding. Figure 6(a) shows a simulation flow for the evaluation, where nonbinary source sequences are transmitted over BSC. Note that the simulation is performed for the proposed joint coding adopting the entropy conversion described in the previous section. For comparison, the conventional sequential coding using Huffman code [1] for source coding and LDPC code for channel coding is also simulated according to Fig. 6(b).

Table 2 shows the compressed data size for 9-ary and 13-ary sources with length 100,000 symbols, where source symbols are generated according to the Gaussian and the Laplace distributions, and $\mathbf{H}_S$ used for the compression is a rate-1/2 irregular $M_S \times N_S$ LDPC matrix having degree distribution $0.275698x + 0.25537x^2 + 0.0765975x^3 + 0.392335x^8$ [17]. Note that the compressed data size of the joint coding is given by M_S. In both source sequences, the compressed data size of the joint coding is smaller than that of the Huffman coding in the sequential coding.

Figure 7 shows the relation between the crossover probability of BSC and the bit error rate (BER) of the decoded word, where rate-1/2 irregular LDPC codes having the same degree distribution as $\mathbf{H}_S$ are applied to the channel coding. Here, the channel code lengths for the Gaussian and Laplace distribution sources are 12,626 bits and 27,282 bits, respectively. This figure says that the joint coding

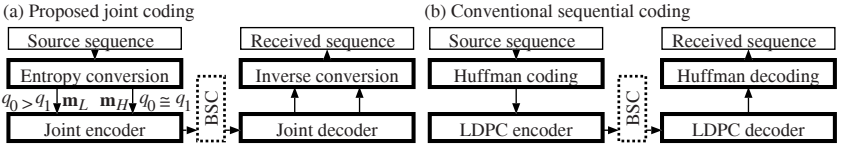


Fig. 6. Simulation flow

Table 2. Compressed data size

Source	Distribution	Source length (Symbols)	Entropy	Compressed data size (bits)	
				Proposed joint coding	Sequential coding (Huffman coding)
9-ary	Gaussian distribution	100,000	2.04715	214,040	215,437
13-ary	Laplace distribution	100,000	1.18892	139,641	143,253

gives lower BERs than the conventional sequential coding. This is because the joint coding can utilize source redundancies in $\mathbf{m}_L$ for channel error correction, as indicated in Fig. 4(b).

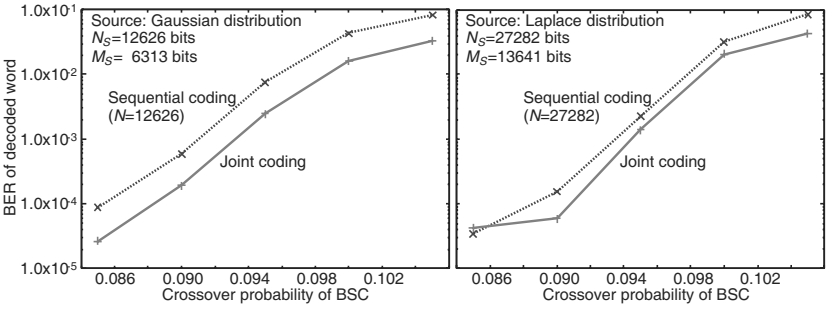


Fig. 7. Bit error rate of decoded word

8 Conclusion

This paper has proposed a joint source-cryptographic-channel coding using two linear block codes and nonsingular matrices. The encoding matrix is generated by multiplying several matrices, i.e., two permutation matrices, a code generator matrix, a nonsingular matrix, and a parity-check matrix, which leads to simple encoding. This paper has clarified that the cryptographic security is equivalent to the McEliece's PKC. Evaluation of the BER of the proposed coding over BSC has shown that the proposed joint coding gives lower BER than the conventional sequential coding. For a BSC with crossover probability $\epsilon = 0.090$ and code length 12,626 bits, the BER of the proposed joint coding gives 1.9×10^{-4}, while that of the conventional sequential coding 5.8×10^{-4}.

In future, we will improve the security level of the proposed joint coding. An efficient algorithm converting redundant input data, such as image data, into sparse information word is also left for future study.

References

1. Huffman, D.A.: A Method for the Construction of Minimum Redundancy Codes. Proc. of the IRE 40(9), 1098–1101 (1952)
2. Ziv, J., Lempel, A.: A Universal Algorithm for Sequential Data Compression. IEEE Trans. Inform. Theory 23(3), 337–343 (1977)
3. Wallace, G.K.: The JPEG Still Picture Compression Standard. Communications of the ACM 34(4), 30–44 (1991)
4. Wieqand, T., Sullivan, G.J., Bjntegaard, G., Luthra, A.: Overview of the H.264/AVC Video Coding Standard. IEEE Trans. Circuits and Systems for Video Technology 13(7), 560–576 (2003)
5. Fujiwara, E.: Code Design for Dependable Systems: Theory and Practical Applications. Wiley, Chichester (2006)
6. MacKay, D.J.C.: Good Error-Correcting Codes Based on Very Sparse Matrices. IEEE Trans. Inform. Theory 45(2), 399–431 (1999)
7. Richardson, T.J., Shokrollahi, M.A., Urbanke, R.L.: Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes. IEEE Trans. Inform. Theory 47(2), 619–637 (2001)
8. Fujiwara, E., Kitakami, M.: Unequal Error Protection in Ziv-Lempel Coding. IEICE Trans. Inform. and Systems E86-D E86-D(12), 2595–2600 (2003)
9. Horn, U., Stuhlmüller, K., Ling, M., Girod, B.: Robust Internet Video Transmission Based on Scalable Coding and Unequal Error Protection. Signal Processing: Image Communication 15(1-2), 77–94 (1999)
10. Zhong, W., Garcia-Frias, J.: LDGM Codes dor Channel Coding and Joint Source-Channel Coding of Correlated Sources. EURASIP J. Applied Signal Processing 2005(6), 942–953 (2005)
11. McEliece, R.J.: A Public-Key Cryptosystem Based on Algebraic Coding Theory. The Deep Space Network Progress Report, DSN PR, 42–44, 114–116 (1978)
12. Kabashima, Y., Murayama, T., Saad, D.: Cryptographical Properties of Ising Spin Systems. Physical Review Letters 84(9), 2030–2033 (2000)
13. Kobara, K., Imai, H.: Semantically Secure McEliece Public-Key Cryptosystem. IEICE Trans. Fundamentals 85(1), 74–83 (2002)
14. Niederreiter, H.: Knapsack-Type Cryptosystems and Algebraic Coding Theory. Problems of Control and Information Theory 15(2), 157–166 (1986)
15. Li, Y.X., Deng, R.H., Wang, X.M.: On the Equivalence of McEliece's and Niederreiter's Public-Key Cryptosystems. IEEE Trans. Inform, Theory 40(1), 271–273 (1994)
16. Fujiwara, E., Namba, K., Kitakami, M.: Parallel Decoding for Burst Error Control Codes. Electronics and Communications in Japan, Part. III 87(1), 38–48 (2004)
17. `http://lthcwww.epfl.ch/research/ldpcopt/`

On the Key-Privacy Issue of McEliece Public-Key Encryption

Shigenori Yamakawa[1], Yang Cui[2], Kazukuni Kobara[2], Manabu Hagiwara[2], and Hideki Imai[1,2]

[1] Chuo University, Japan
shigenori-yamakawa@imailab.jp
[2] Research Center for Information Security (RCIS), National Institute of Advanced Industrial Science & Technology (AIST), Japan
{y-cui, k-kobara, hagiwara.hagiwara, h-imai}@aist.go.jp

Abstract. The notion of key-privacy for encryption schemes was formally defined by Bellare, Boldyreva, Desai and Pointcheval in Asiacrypt 2001. This security notion has the application possibility in circumstances where anonymity is important. In this paper, we investigate the key-privacy issues of McEliece public-key encryption and its significant variants. To our best knowledge, it is the first time to consider key-privacy for such code-based public-key encryption, in the literature. We examine that the key-privacy is not available in the plain McEliece scheme, but can be achieved by some modification, with showing a rigorous proof. We believe that key-privacy confirmation will further magnify the application of McEliece and other code-based cryptography.

1 Introduction

As we all know, the McEliece cryptosystem [7] is based on coding theory and enjoys the merit of fast encryption and decryption. Besides that, McEliece public-key encryption (PKE) is believed to be secure against the adversary with a quantum computer (if it exists). Unlike popular RSA and El Gamal PKEs, the security of McEliece PKE is based on hardness of decoding problem, which is not known to be solvable by quantum computer in polynomial time. Therefore, it appears that McEliece PKE is a promising candidate for the post-quantum cryptography (i.e. If quantum computer is available, most of current PKEs collapse. For such a long-term security consideration of quantum algorithms, we call that post-quantum cryptography).

On the other hand, key-privacy as well as confidentiality (data-privacy) starts to get notice, because of the significance of anonymity in numerous applications. This issue seems necessary in such as, some authenticated key exchange, anonymous credential system, and electronic auction protocols [1]. Even a similar consideration exists in the block-cipher-based encryption scenario. Hence, it is worth looking over carefully the privacy of key as well as data.

Although data-privacy of McEliece PKE has been considered for nearly thirty years, to our best knowledge, its key-privacy issue has never been examined.

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 168–177, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Hence, in this paper, we provide the first rigorous investigation of key-privacy for McEliece PKE and its significant variants, in the literature.

In anonymous communication scenarios, the notion of key-privacy is important. As a sender transfers a ciphertext encrypted by using the receiver's public key, anonymous communication requires that an adversary cannot determine which user's public key has been used to generate the ciphertext that she sees. For example, the key-privacy doesn't exist in the plain RSA cryptosystem which is the most popularly used PKE. Because a ciphertext leaks the information about the utilized public-key (ciphertext distribution over modulo N). Therefore, it is proposed some way to lift up RSA PKE to hold the key-privacy in [1].

For McEliece PKE, assuming the same system parameters (n, k, t) (see Sec. 2.3 for details) will not suffice to imply the key-privacy. Actually, the distribution of the permuted error-correcting code (Goppa Code) plays a central role in our proofs. Besides, we also take advantage of stronger data-privacy to achieve an indistinguishable replacement of a random input for a chosen plaintext, which is the reason why we require the IND-CPA security (Def. 2).

Next, we first explain the preliminary notions and provide a proof to show no key-privacy is available for plain McEliece PKE, in Sec.2. Then, we further check two significant variants of McEliece PKE in Sec.3, and describe a rigorous proof in Sec.4.

2 Preliminaries

In the following, we first provide the security notion of key-privacy of public-key encryption according to [1]. After giving explanation of McEliece PKE, we show that the plain McEliece PKE actually has no key-privacy protection, in a rigorous way. For the simplicity, we only describe the "*indistinguishability of keys under chosen plaintext attack*" (IK-CPA). A stronger security notion defined in chosen ciphertext attack (IK-CCA) setting, can be defined in a similar way.

2.1 Key-Privacy

Definition 1 (IK-CPA). *([1]) Consider a PKE scheme which consists of a tuple of polynomial-time algorithms $PKE = (Gen, Enc, Dec)$. The security of key-privacy is defined as follows.*

1. *On input of security parameter κ, key generation algorithm $Gen(1^\kappa)$ outputs two independent sets of key pairs, (pk_0, sk_0), (pk_1, sk_1), at random.*
2. *Given (pk_0), (pk_1), a polynomial-time adversary $\mathcal{A}$ chooses a plaintext m, and sends it to the encryption oracle (algorithm).*
3. *Encryption oracle randomly flips coin $b \in \{0, 1\}$, to output $Enc_{pk_b}(m) = c$.*
4. *Given target ciphertext c, adversary $\mathcal{A}$ outputs b', where the advantage of success probability over random guess is defined as follows* [1]*,*

$$\mathbf{Adv}_{\mathcal{A}}^{ik-cpa}(\kappa) = |Pr[b = b'] - \frac{1}{2}|$$

[1] The definition of advantage we defined is as twice as the one in [1], where they actually tackle with the same essential issue.

If $\mathbf{Adv}_{\mathcal{A}}^{ik-cpa}(\kappa)$ *is negligible in* κ*, then, we say the underlying PKE is IK-CPA secure. Note that "negligible" means that for any constant* cons*, there exists* $k_0 \in \mathbf{N}$*, s.t. for any* $\kappa > k_0$*,* **Adv** *is less than* $(1/\kappa)^{\mathsf{cons}}$*.*

Remark. Note that in the above game, the adversary can choose whatever she likes to challenge the encryption oracle, even after observing the two given public keys. It immediately concludes that a deterministic (public-key) encryption can never obtain the key-privacy.

2.2 McEliece Public-Key Encryption

The original McEliece PKE was proposed by McEliece [7] in 1978. It is the first PKE based on assumptions other than factoring and discrete log problem, with *on-the-fly* encryption and decryption speed. The McEliece PKE scheme *McPKE* is described as follows.

McPKE=(Gen, Enc, Dec)

1. **Gen**: On input κ, output (pk, sk). $n, t \in \mathbf{N}$, $t \ll n$
   - sk (Private Key): (S, φ, P)
     G': $k \times n$ generating matrix of a binary irreducible $[n, k]$ Goppa code which can correct a maximum of t bits. φ is an efficient decoding algorithm of the underlying code.
     S: $k \times k$ non-singular matrix.
     P: $n \times n$ permutation matrix, chosen at random.
   - pk (Public Key): (G, t)
     G: $k \times n$ matrix given by a product of three matrices $SG'P$.
2. **Enc**: Given pk and a k-bit plaintext m, randomly generate n-bit e with Hamming weight t, output ciphertext c

$$c = mG \oplus e$$

3. **Dec**: On input c, output m with private key sk.
   - Multiply ciphertext c by inverse matrix P^{-1} of P.

$$cP^{-1} = (mS)G' \oplus eP^{-1}$$

   - Error correcting algorithm φ corresponding to G' applies to cP^{-1} to find mS.

$$mS = \varphi(cP^{-1})$$

   - Multiply the above mS by inverse matrix S^{-1} of S to find m.

$$m = (mS)S^{-1}$$

2.3 No Key-Privacy for Plain McEliece PKE

We can prove that key-privacy doesn't hold in the plain McEliece PKE, even though the McEliece PKE is secure. Note that anyone who can invert McEliece PKE can easily break its key-privacy. Thus, given two public keys and corresponding encryption pair (plaintext and ciphertext), to distinguish which key is used is an easier work than inverting McEliece PKE.

Since the different public parameters only lead to a trivial success of adversary, we only consider the case where public parameters (n, k, t) are the *same*, in the whole paper.

Proof. Assume two public keys are generated independently and randomly. It is well known that the Hamming weight t (the number of 1) of error vector e for encryption is small compared with n for typical setting of McEliece PKE. Thus, the random error e inverts the exact t-bit of mG, which makes mG and c only a little different.

On the other hand, as far as key-privacy is effected, ciphertext c must not leak any information about public-key. But, in this case, the ciphertext leaks some information about public-key utilized as a result. Because adversary can choose plaintext m and knows corresponding ciphertext c, it is possible to distinguish corresponding public-key G from mG.(i.e. c leaks mG.)

Let $wt(x)$ denote the Hamming weight of x. Given G_0 and G_1, the adversary chooses m s.t.

$$wt(mG_0 \oplus mG_1) \geq 2t + 1$$

Note that such m can be found easily. Now for given c, the following is true,

- if $b = b'$,

$$wt(c \oplus mG_{b'}) = wt(e) = t$$

- otherwise,

$$wt(c \oplus mG_{b'}) = wt(e \oplus mG_0 \oplus mG_1) \geq wt(mG_0 \oplus mG_1) - wt(e) \geq t + 1$$

It is easy to distinguish the Hamming weight of them with probability 1, i.e. $Pr[b = b'] = 1$.

From above, $\mathbf{Adv}_{\mathcal{A}}^{ik-cpa}(\kappa)$ is not negligible. So, we say that plain McEliece PKE is not IK-CPA secure. □

3 Key-Privacy of Modified McEliece PKE

Due to the lack of key-privacy in the plain McEliece PKE, it is important to find a way to guarantee anonymity as well as confidentiality, in some useful scenarios.

Luckily, it is common to use some security-enhanced variants of McEliece PKE rather than the plain one. Based on this stronger data-privacy, we next show the key-privacy is also available under appropriate assumptions.

3.1 Data Privacy

Definition 2 (IND-CPA). *[2] Consider a PKE scheme which consists of a tuple of polynomial-time algorithms, where* $PKE = (Gen, Enc, Dec)$.

1. *On input of security parameter* κ*, key generation algorithm* $Gen(1^\kappa)$ *outputs the set of private-key and public-key,* (pk, sk)
2. *Given* (pk, sk)*, a polynomial-time adversary* $\mathcal{A}$ *chooses two equal-length plaintexts* m_0, m_1 $(m_0 \neq m_1)$*, and sends them to the encryption oracle.*
3. *Encryption oracle (algorithm) randomly flips coin* $b \in \{0, 1\}$*, to encrypt* $Enc(pk, m_b) = c$.
4. *Given target ciphertext* c*, adversary* $\mathcal{A}$ *outputs* $b' \in \{0, 1\}$*, where the advantage of success probability over random guess is defined as follows,*

$$\mathbf{Adv}_{\mathcal{A}}^{ind-cpa}(\kappa) = |Pr[b = b'] - \frac{1}{2}|$$

If $\mathbf{Adv}_{\mathcal{A}}^{ind-cpa}(\kappa)$ *is negligible, then, we say underlying PKE is IND-CPA secure.*

Remark. IND-CPA means that indistinguishability of encrypted data is protected against the chosen plaintext attack (CPA) of adversaries. For the chosen ciphertext attack security (IND-CCA), the decryption oracle has to be considered additionally. We refer to [2] for a formal definition.

The reason why we need IND-CPA (resp. CCA) is that the complete control of input plaintext gives adversary too much freedom to employ an attack, in the IK-CPA (resp. CCA) notion. Our motivation is to deny such an advantage of adversary.

3.2 IND-CPA McEliece PKE in the Standard Model

We first examine a recently proposed variant [9] of McEliece PKE, which is provably secure in the standard model (i.e. without the assumption of ideal hash functions, so-called random oracle model [3]).

The IND-CPA security is derived from padding random number r with plaintext m, which makes difficult for adversaries to stay in control of the plaintext. Let $[r|m]$ denote the bit sequence concatenation of r and m. Then as what is explained in the following, the randomized McEliece cryptosystem [9] achieves IND-CPA (semantic security).

McPKE′ = (Gen′, Enc′, Dec′)

1. **Gen′**: On input κ, output (pk, sk). $n, t \in \mathbf{N}$, $t \ll n$
   - sk (Private Key): (S, φ, P)
   - pk (Public Key): (G, t), where $k = k_1 + k_2$, $G^T = [G_1^T | G_2^T]$
     (G_1: $k_1 \times n$ submatrix of G)
     (G_2: $k_2 \times n$ submatrix of G)
2. **Enc′**: Given pk and a k_2-bit plaintext m, generate k_1-bit r at random, and output ciphertext c

$$c = [r|m]G \oplus e = (rG_1 \oplus e) \oplus mG_2$$

3. **Dec′**: On input c, **Dec′** works as the same as **Dec**, except that it outputs k_2-bit m only.

The IND-CPA security of above scheme, relies on 1). the pseudorandomness of G and 2). the one-wayness of McEliece PKE. The former guarantees the padded r being masked and makes $rG_1 \oplus e$ look random; the latter ensures the r cannot be found by some message-inverting attack. It is worth noticing that r should be long enough, where security evaluation is given in [9].

In the formal proof, the indistinguishability of permuted code is defined. This fact is also used to build secure McEliece signature in [4].

Definition 3 (Pseudorandom Codes). *[4] Let $\mathcal{A}$ be a polynomial-time adversary, which outputs 1 with certain distribution, and 0 otherwise. Given a uniform probability distribution $\mathcal{C}(n, k)$ of all binary linear $[n, k]$ code $\mathbf{G}$, and any other probability distribution $\mathcal{F}(n, k)$. It is called a pseudorandom code if*

$$\mathbf{Adv}^{prc}_{\mathcal{A},\mathbf{G}}(\kappa) = |\Pr[\mathcal{A}^{\mathcal{C}(n,k)}(\kappa) = 1] - \Pr[\mathcal{A}^{\mathcal{F}(n,k)}(\kappa) = 1]|$$

is negligible.

Thus, it is easily concluded the following lemma,

Lemma 1. *[9] The underlying scheme* McPKE *is IND-CPA secure, if the $[n, k]$ code is pseudorandom and inverting McEliece PKE is infeasible in the polynomial time. More precisely,*

$$\mathbf{Adv}^{ind-cpa}_{\mathcal{A},\mathbf{Mc}'}(\kappa) \leq \mathbf{Adv}^{prc}_{\mathcal{A},\mathbf{G}}(\kappa) + \mathbf{Adv}^{ow}_{\mathcal{A},\mathbf{Mc}}(\kappa)$$

where, $\mathbf{Adv}^{ind-cpa}_{\mathcal{A},\mathbf{Mc}'}(\kappa)$ is the advantage of polynomial-time adversary $\mathcal{A}$ to distinguish input messages, and $\mathbf{Adv}^{ow}_{\mathcal{A},\mathbf{Mc}}(\kappa)$ is the success probability of inverting McEliece PKE.

Remark. Intuitively, the requirement of IND-CPA (resp. IND-CCA), comes from the motivation of blocking the free access of adversary to the input plaintext. As we have shown in Sec. 2.3, the complete control of input gives too much power to adversary, so that she can have enough room to cope with the target public keys.

IND-CPA (resp. IND-CCA) security, may guarantee the indistinguishability of input plaintexts, which means it is difficult for adversary to distinguish a chosen plaintext from a random message, under one public key. Then what left is to prove the similar holds even under two public keys, as done in our proof in Sec. 4.

3.3 IND-CCA McEliece PKE in the Random Oracle Model

Assuming the random oracle model [3], Kobara and Imai [6] first proposed two tailored conversions for McEliece cryptosystem to obtain IND-CCA security. In the following, we simply provide one of their McEliece PKE conversions.

$\mathbf{McPKE''} = (\mathbf{Gen''}, \mathbf{Enc''}, \mathbf{Dec''})$

---

1. $\mathbf{Gen''}$: On input κ, output (pk, sk). $n, t \in \mathbf{N}$, $t \ll n$
   - sk (Private Key): (S, φ, P)
   - pk (Public Key): (G, t), cryptographic hash functions $\mathcal{G}, \mathcal{H}, \mathcal{H}_E$
2. $\mathbf{Enc''}$: Given pk, a k-bit encoded message m, output ciphertext c.
   - Generate random number r, compute x_1, x_2 as follows.

$$x_1 = \mathcal{G}(r) \oplus m, \quad x_2 = r \oplus \mathcal{H}(x_1)$$

   - Define x_3, x_4 as, $(x_4 \| x_3) = (x_2 \| x_1)$
   - $\mathcal{H}_E$ maps an integer r into $\mathbf{Z}_{\binom{n}{t}}$. A bijective mapping $Conv$ converts $\mathcal{H}_E(r)$ to the corresponding error vector e.

$$e = Conv(\mathcal{H}_E(r))$$

   - Output x_4 with Encryption of (x_3, e).

$$c = x_4 \| \mathbf{Enc}_{pk}(x_3, e)$$

3. $\mathbf{Dec''}$: Simply reverse the $\mathbf{Enc''}$.

This scheme is IND-CCA secure. Note that it is a stronger security notion and implies IND-CPA security immediately.

4 Security Proof

4.1 IND-CPA McEliece PKE in Section 3.2 is IK-CPA

We confirm the presence of key-privacy based on the IND-CPA McEliece PKE in the standard model.

Theorem 1. *The underlying modified McEliece PKE, McPKE'=(Gen',Enc',Dec') is IK-CPA secure, in particular,*

$$\mathbf{Adv}^{ik-cpa}_{\mathcal{A},Mc'}(\kappa) \leq 2\mathbf{Adv}^{prc}_{\mathcal{A},\mathbf{G}}(\kappa) + 2\mathbf{Adv}^{ow}_{\mathcal{A},\mathbf{Mc}}(\kappa)$$

Proof. We define a sequence of games to link the IK-CPA security with IND-CPA security. Define $\Pr[E_i]$ the probability of event E_i that $b' = b$ in each corresponding game. For simplicity, let ϵ be $Adv^{ind-cpa}_{A}(\kappa)$.

G1. On input of security parameter κ, key generation algorithm generates randomly two pairs of keys $\langle (pk_0, sk_0), (pk_1, sk_1)\rangle$ (simply write as $\langle pk_0, pk_1\rangle$), and gives public keys to a polynomial-time adversary $\mathcal{A}$. $\mathcal{A}$ chooses m^* as she wants, to challenge the encryption oracle, and receive the corresponding ciphertext c, as follows. $\overset{R}{\leftarrow}$ means generate randomly and uniformly.

$$\begin{aligned} \langle pk_0, pk_1\rangle &\overset{R}{\leftarrow} \mathbf{Gen}'(1^\kappa). \\ m^* &\leftarrow \mathcal{A}(pk_0, pk_1, 1^\kappa). \\ c &\leftarrow \mathbf{Enc}'_{pk_b}(m^*). \\ b' &\leftarrow \mathcal{A}(pk_0, pk_1, c, m^*). \end{aligned}$$

It is easy to see that the above is the same as the Def. 1. Thus, $\Pr[E_1]$ should be the success probability of breaking IK-CPA game.

G2. **G2** is the same as the **G1**, except that a random plaintext m_R generated from message domain $\mathcal{M}$ is provided.

$$\begin{aligned} \langle pk_0, pk_1\rangle &\overset{R}{\leftarrow} \mathbf{Gen}'(1^\kappa). \\ m^* &\leftarrow \mathcal{A}(pk_0, pk_1, 1^\kappa). \\ m_R &\overset{R}{\leftarrow} \mathcal{M}. \\ c &\leftarrow \mathbf{Enc}'_{pk_b}(m^*). \\ b' &\leftarrow \mathcal{A}(pk_0, pk_1, c, m^*, m_R). \end{aligned}$$

Note that the success probability of adversary $\mathcal{A}$ does not change, because $\mathcal{A}$ can simply make use of m^*. Thus,

$$\Pr[E_2] = \Pr[E_1] \tag{1}$$

G3. **G3** is obtained from **G2** by modifying the encryption oracle query.

$$\begin{aligned} \langle pk_0, pk_1\rangle &\overset{R}{\leftarrow} \mathbf{Gen}'(1^\kappa). \\ m^* &\leftarrow \mathcal{A}(pk_0, pk_1, 1^\kappa). \\ m_R &\overset{R}{\leftarrow} \mathcal{M}. \\ c &\leftarrow \mathbf{Enc}'_{pk_b}(m_R). \\ b' &\leftarrow \mathcal{A}(pk_0, pk_1, c, m^*, m_R). \end{aligned}$$

It is easy to see that if a random m_R and a carefully-chosen m^* cannot be distinguished, then success probability of $\mathcal{A}$ will not change. Let F be the event that $\mathcal{A}$ correctly determines which plaintext is input to encryption oracle. The following holds,

$$\Pr[E_2] = \Pr[E_3|\bar{F}]$$

By the well-known difference lemma [10], it concludes that

$$|\Pr[E_3] - \Pr[E_2]| \leq \Pr[F] \tag{2}$$

Let us consider the probability of event F occurs. Assume adversary outputs $\delta = 1$ when m_R is detected to be sent to encryption oracle, and $\delta = 0$ when m^* is detected. Because the input m of encryption oracle is either m_R or m^*, a random plaintext m_R and a chosen plaintext m^* can be distinguished *at most* with the following probability,

$$\begin{aligned}\Pr[F] &\leq |\Pr[\delta = 1|m = m_R] + \Pr[\delta = 0|m = m^*] \\ &\qquad -(\Pr[\delta = 0|m = m_R] + \Pr[\delta = 1|m = m^*])| \\ &= |\Pr[\delta = 1|m = m_R] + \Pr[\delta = 0|m = m^*] \\ &\qquad -(1 - \Pr[\delta = 1|m = m_R] + \Pr[\delta = 0|m = m^*])| \\ &= 2|\Pr[\delta = 1|m = m_R] + \Pr[\delta = 0|m = m^*] - 1/2| \end{aligned} \tag{3}$$

Note that $(\Pr[\delta = 1|m = m_R] + \Pr[\delta = 0|m = m^*])$ is the success probability of IND-CPA adversary (Def.2), the right side of equation 3 is equal to 2ϵ. Hence, the probability of F is bounded by,

$$\Pr[F] \leq 2|\Pr[\delta = 1|m = m_R] + \Pr[\delta = 0|m = m^*] - 1/2| = 2\epsilon \tag{4}$$

Now we evaluate the distribution D_0 and D_1, in game **G3**.

$$D_0 = \{\langle pk_0, pk_1, \mathbf{Enc}'_{pk_0}(m_R)\rangle | (pk_0, sk_0), (pk_1, sk_1) \xleftarrow{R} \mathbf{Gen}'(1^\kappa)\}$$

$$D_1 = \{\langle pk_0, pk_1, \mathbf{Enc}'_{pk_1}(m_R)\rangle | (pk_0, sk_0), (pk_1, sk_1) \xleftarrow{R} \mathbf{Gen}'(1^\kappa)\}$$

It appears that with random input m_R and pseudorandom code of public keys of $McPKE'$, the above distributions are only looking random, and their distance is too close to be distinguished.

As a consequence, the best way to find $b' = b$ is to guess at random, which means that the probability $\Pr[E_3]$ is $1/2$. Summarizing all above equations. There is

$$\begin{aligned}\mathbf{Adv}^{ik-cpa}_{\mathcal{A},Mc'}(\kappa) &= |\Pr[E_1] - (1/2)| = |\Pr[E_1] - \Pr[E_2]| + |\Pr[E_2] - \Pr[E_3]| \\ &\leq 2\epsilon = 2\mathbf{Adv}^{ind-cpa}_{\mathcal{A},\mathbf{Mc'}}(\kappa) \end{aligned} \tag{5}$$

Combined with Lemma 1, it is easy to prove the theorem, hence finishes the proof. □

4.2 IND-CCA McEliece PKE in Section 3.3 is IK-CPA (resp.CCA)

In general, IND-CCA security places strict condition on the public-key cryptosystem than IND-CPA security. Intuitively, we can consider that IND-CCA is a special case of IND-CPA. In this sense, assuming the random oracle model, we can prove that IND-CCA McEliece PKE suffices IK-CPA, analogously. Furthermore, it is possibly to be proven IK-CCA secure, which employs similar proofs with some additional decryption simulation. We would like to show that in the full version of this paper.

5 Conclusion

In this paper, we have examined key-privacy issue against the chosen plaintext attack (CPA) for the plain McEliece PKE and its significant variants. We first show that plain McEliece public-key cryptosystem doesn't have key-privacy. Then we provide solutions based on IND-CPA McEliece PKE, and rigorously prove that these variants suffice IK-CPA. We believe that in more scenarios, anonymity is as crucial as confidentiality. Hence, the key-privacy issue of public-key encryption will play more important roles and attract more attentions.

Acknowledgement

We would like to thank anonymous reviewers for their helpful comments. Yang Cui would like to thank for the support of JSPS postdoctoral fellowship.

References

1. Bellare, M., Boldyreva, A., Desai, A., Pointcheval, D.: Key-Privacy in Public-Key Encryption. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 566–582. Springer, Heidelberg (2001)
2. Bellare, M., Desai, A., Pointcheval, D., Rogaway, P.: Relations Among Notions of Security for Public-Key Encryption Schemes. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 26–45. Springer, Heidelberg (1998)
3. Bellare, M., Rogaway, P.: Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In: 1993 ACM Conf. Computer and Communications Security, pp. 62–73 (1993)
4. Courtois, N., Finiasz, M., Sendrier, N.: How to Achieve a McEliece-Based Digital Signature Scheme. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 157–174. Springer, Heidelberg (2001)
5. Halevi, S.: A Sufficient Condition for Key-Privacy. Cryptology ePrint Archive: Report 2005/005 (2005)
6. Kobara, K., Imai, H.: Semantically Secure McEliece Public-Key Cryptosystems-Conversions for McEliece PKC. Public Key Cryptography, pp. 19–35 (2001)
7. McEliece, R.J.: A Public-Key Cryptosystem Based on Algebraic Coding Theory. Deep Space Network Progress Rep. (1978)
8. Niederreiter, H.: Knapsack-type Cryptosystems and Algebraic Coding Theory. Prob. of Control and Inf. Theory 15(2), 159–166 (1986)
9. Nojima, R., Imai, H., Kobara, K., Morozov, K.: Semantic Security for the McEliece Cryptosystem without Random Oracles. In: WCC 2007, pp. 257–268 (2007)
10. Shoup, V.: Sequences of Games: a Tool for Taming Complexity in Security Proofs. Cryptology ePrint Archive: Report 2004/332 (2004)

Lattices for Distributed Source Coding: Jointly Gaussian Sources and Reconstruction of a Linear Function*

Dinesh Krithivasan and S. Sandeep Pradhan

Department of Electrical Engineering and Computer Science,
University of Michigan, Ann Arbor, MI 48109, USA
dineshk@umich.edu, pradhanv@eecs.umich.edu

Abstract. Consider a pair of correlated Gaussian sources (X_1, X_2). Two separate encoders observe the two components and communicate compressed versions of their observations to a common decoder. The decoder is interested in reconstructing a linear combination of X_1 and X_2 to within a mean-square distortion of D. We obtain an inner bound to the optimal rate-distortion region for this problem. A portion of this inner bound is achieved by a scheme that reconstructs the linear function directly rather than reconstructing the individual components X_1 and X_2 first. This results in a better rate region for certain parameter values. Our coding scheme relies on lattice coding techniques in contrast to more prevalent random coding arguments used to demonstrate achievable rate regions in information theory. We then consider the case of linear reconstruction of K sources and provide an inner bound to the optimal rate-distortion region. Some parts of the inner bound are achieved using the following coding structure: lattice vector quantization followed by "correlated" lattice-structured binning.

1 Introduction

In this work, we present a coding scheme for distributed coding of a pair of jointly Gaussian sources. The encoders each observe a different component of the source and communicate compressed versions of their observations to a common decoder through rate-constrained noiseless channels. The decoder is interested in reconstructing a linear function of the sources to within a mean squared error distortion of D.

The problem of distributed source coding to reconstruct a function of the sources losslessly was considered in [1]. An inner bound was obtained for the performance limit which was shown to be optimal if the sources are conditionally independent given the function. In [2], the performance limit is given for the case of lossless reconstruction of the modulo-2 sum of two correlated binary sources and was shown to be tight for the symmetric case. This has been extended to several cases in [3] (see Problem 23 on page 400) and [4]. An improved

* This work was supported by NSF grant (CAREER) CCF-0448115.

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 178–187, 2007.
© Springer-Verlag Berlin Heidelberg 2007

inner bound was provided for this case in [5]. The key point to note is that the performance limits given in [2,4,5] are outside the inner bound provided in [1]. While [1] employs random vector quantization followed by independent random binning , the coding schemes of [2,4,5] instead use structured random binning based on linear codes on finite fields. Further, the binning operation of the quantizers of the sources are "correlated". This incorporation of structure in binning appears to give improvements in rates especially for those cases that involve reconstruction of a function of the sources.

With this as motivation, in this paper we consider a lossy distributed coding problem with K jointly Gaussian sources with one reconstruction. The decoder wishes to reconstruct a linear function of the sources with squared error as fidelity criterion. We consider a coding scheme with the following structure: sources are quantized using structured vector quantizers followed by "correlated" structured binning. The structure used in this process is given by lattice codes. We provide an inner bound to the optimal rate-distortion region. We show that the proposed inner bound is better for certain parameter values than an inner bound that can be obtained by using a coding scheme that uses random vector quantizers following by independent random binning. For this purpose we use the machinery developed by [9,10,11,12] for the Wyner-Ziv problem in the quadratic Gaussian case.

The paper is organized as follows. In Section 2, we give a concise overview of the asymptotic properties of high-dimensional lattices that are known in the literature and we use these properties in the rest of the paper. In Section 3, we define the problem formally for the case of two sources and present an inner bound to the optimal rate-distortion region given by a coding structure involving structured quantizers followed by "correlated" structured binning. Further, we also present another inner bound achieved by a scheme that is based on the Berger-Tung inner bound. Then we present our lattice based coding scheme and prove achievability of the inner bound. In Section 4, we consider a generalization of the problem that involves reconstruction of a linear function of an arbitrary finite number of sources. In Section 5, we provide a set of numerical results for the two-source case that demonstrate the conditions under which the lattice based scheme performs better than the Berger-Tung based scheme. We conclude with some comments in Section 6. We use the following notation throughout this paper. Variables with superscript n denote an n-length random vector whose components are mutually independent. However, random vectors whose components are not independent are denoted without the use of the superscript. The dimension of such random vectors will be clear from the context.

2 Preliminaries on High-Dimensional Lattices

2.1 Overview of Lattice Codes

Lattice codes play the same role in Euclidean space that linear codes play in Hamming space. Introduction to lattices and to coding schemes that employ lattice codes can be found in [9,10,11]. In the rest of this section, we will briefly

review some properties of lattice codes that are relevant to our coding scheme. We use the same notation as in [10] for these quantities.

An n-dimensional lattice Λ is composed of all integer combinations of the columns of an $n \times n$ matrix G called the generator matrix of the lattice. Associated with every lattice Λ is a natural quantizer namely one that associates with every point in $\mathbb{R}^n$ its nearest lattice point. This quantizer can be described by the function $Q_\Lambda(x)$. The quantization error associated with the quantizer $Q_\Lambda(\cdot)$ is defined by $x \bmod \Lambda = x - Q_\Lambda(x)$. This operation satisfies the useful distribution property

$$((x \bmod \Lambda) + y) \bmod \Lambda = (x + y) \bmod \Lambda \quad \forall\, x, y. \tag{1}$$

The basic Voronoi region $\mathcal{V}_0(\Lambda)$ of the lattice Λ is the set of all points closer to the origin than to any other lattice point. Let $V(\Lambda)$ denote the volume of the Voronoi region of Λ. The second moment of a lattice Λ is the expected value per dimension of the norm of a random vector uniformly distributed over $\mathcal{V}_0(\Lambda)$ and is given by

$$\sigma^2(\Lambda) = \frac{1}{n} \frac{\int_{\mathcal{V}_0(\Lambda)} \| x \|^2 \, \mathrm{dx}}{\int_{\mathcal{V}_0(\Lambda)} \mathrm{dx}} \tag{2}$$

The normalized second moment is defined as $G(\Lambda) \triangleq \sigma^2(\Lambda)/V^{2/n}(\Lambda)$.

In [12], the existence of high dimensional lattices that are "good" for quantization and for coding is discussed. The criteria used therein to define goodness are as follows:

- A sequence of lattices $\Lambda^{(n)}$ (indexed by the dimension n) is said to be a good channel σ_Z^2-code sequence if $\forall \epsilon > 0$, $\exists N(\epsilon)$ such that for all $n > N(\epsilon)$ the following conditions are satisfied for some $E(\epsilon) > 0$:

$$V(\Lambda^{(n)}) < 2^{n(\frac{1}{2}\log(2\pi e \sigma_Z^2) + \epsilon)} \quad \text{and} \quad P_e(\Lambda^{(n)}, \sigma_Z^2) < 2^{-nE(\epsilon)}. \tag{3}$$

  Here P_e is the probability of decoding error when the lattice points of $\Lambda^{(n)}$ are used as codewords in the problem of coding for the unconstrained AWGN channel with noise variance σ_Z^2 as considered by Poltyrev [13].
- A sequence of lattices $\Lambda^{(n)}$ (indexed by the dimension n) is said to be a good source D-code sequence if $\forall \epsilon > 0, \exists N(\epsilon)$ such that for all $n > N(\epsilon)$ the following conditions are satisfied:

$$\log(2\pi e G(\Lambda^{(n)})) < \epsilon \quad \text{and} \quad \sigma^2(\Lambda^{(n)}) = D. \tag{4}$$

2.2 Nested Lattice Codes

For lossy coding problems involving side-information at the encoder/decoder, it is natural to consider nested codes [10]. We review the properties of nested lattice codes here. Further details can be found in [10].

A pair of n-dimensional lattices (Λ_1, Λ_2) is nested, i.e., $\Lambda_2 \subset \Lambda_1$, if their corresponding generating matrices G_1, G_2 satisfy $G_2 = G_1 \cdot J$ where J is an $n \times n$ integer matrix with determinant greater than one. Λ_1 is referred to as the fine lattice while Λ_2 is the coarse lattice. In many applications of nested lattice codes, we require the lattices involved to be a good source code and/or a good channel code. We term a nested lattice (Λ_1, Λ_2) good if (a) the fine lattice Λ_1 is both a good δ_1-source code and a good δ_1-channel code and (b) the coarse lattice Λ_2 is both a good δ_2-source code and a δ_2-channel code. The existence of good lattice codes and good nested lattice codes (for various notions of goodness) has been studied in [11,12,14] which use the random coding method of [15]. Using the results of [11,12], it was shown in [14] that good nested lattices in the sense described above do exist.

3 Distributed Source Coding for the Two-Source Case

3.1 Problem Statement and Main Result

In this section we consider a distributed source coding problem for the reconstruction of the linear function $Z \triangleq F(X_1, X_2) = X_1 - cX_2$. Consideration of this function is enough to infer the behavior of any linear function $c_1X_1 + c_2X_2$ and has the advantage of fewer variables.

Consider a pair of correlated jointly Gaussian sources (X_1, X_2) with a given joint distribution $p_{X_1X_2}(x_1, x_2)$. The source sequence (X_1^n, X_2^n) is independent over time and has the product distribution $\prod_{i=1}^n p_{X_1X_2}(x_{1i}, x_{2i})$. The fidelity criterion used is average squared error. Given such a jointly Gaussian distribution $p_{X_1X_2}$, we are interested in the optimal rate-distortion region which is defined as the set of all achievable tuples (R_1, R_2, D) where achievability is defined in the usual Shannon sense. Here D is the mean squared error between the function and its reconstruction at the decoder. Without loss of generality, the sources can be assumed to have unit variance and let the correlation coefficient $\rho > 0$. In this case, $\sigma_Z^2 \triangleq \mathrm{Var}(Z) = 1 + c^2 - 2\rho c$.

We present the rate region of our scheme below.

Theorem 1. *The set of all tuples of rates and distortion (R_1, R_2, D) that satisfy*

$$2^{-2R_1} + 2^{-2R_2} \leq \left(\frac{\sigma_Z^2}{D}\right)^{-1} \tag{5}$$

are achievable.

Proof. See Section 3.2. □

We also present an achievable rate region based on ideas similar to Berger-Tung coding scheme [6,7].

Theorem 2. *Let the region $\mathcal{RD}_{in}$ be defined as follows.*

$$\mathcal{RD}_{in} = \bigcup_{(q_1,q_2)\in\mathbb{R}_+^2} \left\{(R_1, R_2, D)\colon R_1 \geq \frac{1}{2}\log\frac{(1+q_1)(1+q_2)-\rho^2}{q_1(1+q_2)},\right.$$

$$R_2 \geq \frac{1}{2}\log\frac{(1+q_1)(1+q_2)-\rho^2}{q_2(1+q_1)},\ R_1+R_2 \geq \frac{1}{2}\log\frac{(1+q_1)(1+q_2)-\rho^2}{q_1q_2}$$

$$D \geq \frac{q_1\alpha + q_2c^2\alpha + q_1q_2\sigma_Z^2}{(1+q_1)(1+q_2)-\rho^2}\Bigg\}. \tag{6}$$

where $\alpha \triangleq 1-\rho^2$ *and* $\mathbb{R}_+$ *is the set of positive reals. Then the rate distortion tuples* (R_1, R_2, D) *which belong to* $\mathcal{RD}_{in}^*$ *are achievable where* * *denotes convex closure.*

Proof. Follows directly from the application of Berger-Tung inner bound with the auxiliary random variables involved being Gaussian. □

For certain values of ρ, c and D, the sum-rate given by Theorem 1 is better than that given in Theorem 2. This implies that each rate region contains rate points which are not contained in the other. Thus, an overall achievable rate region for the coding problem can be obtained as the convex closure of the union of all rate distortion tuples (R_1, R_2, D) given in Theorems 1 and 2. A further comparison of the two schemes is presented in Section 5. Note that for $c < 0$, it has been shown in [8] that the rate region given in Theorem 2 is tight.

3.2 The Coding Scheme

In this section, we present a lattice based coding scheme for the problem of reconstructing the above linear function of two jointly Gaussian sources whose performance approaches the inner bound given in Theorem 1. In what follows, a nested lattice code is taken to mean a sequence of nested lattice codes indexed by the lattice dimension n.

We will require nested lattice codes $(\Lambda_{11}, \Lambda_{12}, \Lambda_2)$ where $\Lambda_2 \subset \Lambda_{11}$ and $\Lambda_2 \subset \Lambda_{12}$. We need the fine lattices Λ_{11} and Λ_{12} to be good source codes (of appropriate second moment) and the coarse lattice Λ_2 to be a good channel code. The proof of the existence of such nested lattices was shown in [14]. The parameters of the nested lattice are chosen to be

$$\sigma^2(\Lambda_{11}) = q_1,\ \sigma^2(\Lambda_{12}) = \frac{D\sigma_Z^2}{\sigma_Z^2 - D} - q_1, \quad \text{and } \sigma^2(\Lambda_2) = \frac{\sigma_Z^4}{\sigma_Z^2 - D} \tag{7}$$

where $0 < q_1 < D\sigma_Z^2/(\sigma_Z^2 - D)$. The coding problem is non-trivial only for $D < \sigma_Z^2$ and in this range, $D\sigma_Z^2/(\sigma_Z^2 - D) < \sigma^2(\Lambda_2)$ and therefore $\Lambda_2 \subset \Lambda_{11}$ and $\Lambda_2 \subset \Lambda_{12}$ indeed.

Let U_1 and U_2 be random vectors (dithers) that are independent of each other and of the source pair (X_1, X_2). Let U_i be uniformly distributed over the basic Voronoi region $\mathcal{V}_{0,1i}$ of the fine lattices Λ_{1i} for $i = 1, 2$. The decoder is assumed to share this randomness with the encoders. The source encoders use these nested lattices to quantize X_1 and cX_2 respectively according to equation

$$S_1 = (Q_{\Lambda_{11}}(X_1^n + U_1)) \mod \Lambda_2\,,\ S_2 = (Q_{\Lambda_{12}}(cX_2^n + U_2)) \mod \Lambda_2. \tag{8}$$

Note that the second encoder scales the source X_2 before encoding it. The decoder receives the indices S_1 and S_2 and reconstructs

$$\hat{Z} = \left(\frac{\sigma_Z^2 - D}{\sigma_Z^2}\right)([(S_1 - U_1) - (S_2 - U_2)] \bmod \Lambda_2)\,. \tag{9}$$

In general, the rate of a nested lattice encoder (Λ_1, Λ_2) with $\Lambda_2 \subset \Lambda_1$ is given by $R = \frac{1}{2}\log\frac{\sigma^2(\Lambda_2)}{\sigma^2(\Lambda_1)}$. Thus, the rates of the two encoders are given by

$$R_1 = \frac{1}{2}\log\frac{\sigma_Z^4}{q_1(\sigma_Z^2 - D)} \quad \text{and} \quad R_2 = \frac{1}{2}\log\frac{\sigma_Z^4}{D\sigma_Z^2 - q_1(\sigma_Z^2 - D)} \tag{10}$$

Clearly, for a fixed choice of q_1 all rates greater than those given in equation (10) are achievable. The union of all achievable rate-distortion tuples (R_1, R_2, D) over all choices of q_1 gives us an achievable region. Eliminating q_1 between the two rate equations gives the rate region claimed in Theorem 1. It remains to show that this scheme indeed reconstructs the function Z to within a distortion D. We show this in the following.

Using the distributive property of lattices described in equation (1), we can reduce the coding scheme to a simpler equivalent scheme by eliminating the first mod-Λ_2 operation in both the signal paths. The decoder can now be described by the equation

$$\hat{Z} = \left(\frac{\sigma_Z^2 - D}{\sigma_Z^2}\right)([(X_1^n + e_{q_1}) - (cX_2^n + e_{q_2})] \bmod \Lambda_2) \tag{11}$$

$$= \left(\frac{\sigma_Z^2 - D}{\sigma_Z^2}\right)([Z^n + e_{q_1} - e_{q_2}] \bmod \Lambda_2) \tag{12}$$

where e_{q_1} and e_{q_2} are dithered lattice quantization noises given by

$$e_{q_1} = Q_{\Lambda_{11}}(X_1^n + U_1) - (X_1^n + U_1)\,,\; e_{q_2} = Q_{\Lambda_{12}}(cX_2^n + U_2) - (cX_2^n + U_2). \tag{13}$$

The subtractive dither quantization noise e_{q_i} is independent of both sources X_1 and X_2 and has the same distribution as $-U_i$ for $i = 1, 2$ [10]. Since the dithers U_1 and U_2 are independent and for a fixed choice of the nested lattice e_{q_i} is a function of U_i alone, e_{q_1} and e_{q_2} are independent as well. Let $e_q = e_{q_1} - e_{q_2}$ be the effective dither quantization noise. The decoder reconstruction in equation (12) can be simplified as

$$\hat{Z} = \left(\frac{\sigma_Z^2 - D}{\sigma_Z^2}\right)([Z^n + e_q] \bmod \Lambda_2) \overset{\text{c.d}}{=} \left(\frac{\sigma_Z^2 - D}{\sigma_Z^2}\right)(Z^n + e_q) \tag{14}$$

$$= Z^n + \left(\left(\frac{\sigma_Z^2 - D}{\sigma_Z^2}\right)e_q - \frac{D}{\sigma_Z^2}Z^n\right) \triangleq Z^n + N. \tag{15}$$

The $\overset{\text{c.d}}{=}$ in equation (14) stands for equality under the assumption of correct decoding. Decoding error occurs if equation (14) doesn't hold. Let P_e be the

probability of decoding error. Assuming correct decoding, the distortion achieved by this scheme is the second moment per dimension[1] of the random vector N in equation (15). This can be expressed as

$$\frac{\mathbb{E} \parallel N \parallel^2}{n} = \left(\frac{\sigma_Z^2 - D}{\sigma_Z^2}\right)^2 \frac{\mathbb{E} \parallel e_q \parallel^2}{n} + \left(\frac{D}{\sigma_Z^2}\right)^2 \frac{\mathbb{E} \parallel Z^n \parallel^2}{n} \tag{16}$$

where we have used the independence of e_{q_1} and e_{q_2} to each other and to the sources X_1 and X_2 (and therefore to $Z = X_1 - cX_2$). Since e_{q_i} has the same distribution as $-U_i$, their expected norm per dimension is just the second moment of the corresponding lattice $\sigma^2(\Lambda_{1i})$. Hence the effective distortion achieved by the scheme is

$$\frac{1}{n}\mathbb{E}\|Z^n - \hat{Z}\|^2 = \left(\frac{\sigma_Z^2 - D}{\sigma_Z^2}\right)^2 \left(\frac{D\sigma_Z^2}{\sigma_Z^2 - D}\right) + \frac{D^2\sigma_Z^2}{\sigma_Z^4} = D. \tag{17}$$

Hence, the proposed scheme achieves the desired distortion provided correct decoding occurs at equation (14). Let us now prove that equation (14) indeed holds with high probability for an optimal choice of the nested lattice, i.e., there exists a nested lattice code for which $P_e \to 0$ as $n \to \infty$ where, $P_e = Pr\left((Z^n + e_q) \bmod \Lambda_2 \neq (Z^n + e_q)\right)$.

To this end, let us first compute the normalized second moment of $(Z^n + e_q)$.

$$\frac{1}{n}\mathbb{E} \parallel Z^n + e_q \parallel^2 = \sigma_Z^2 + q_1 + \frac{\sigma_Z^2 D}{\sigma_Z^2 - D} - q_1 = \sigma^2(\Lambda_2). \tag{18}$$

It was shown in [9] that as $n \to \infty$, the quantization noises e_{q_i} tend to a white Gaussian noise for an optimal choice of the nested lattice. It can be shown that, under these conditions, e_q also tends to a white Gaussian noise of the same variance as e_q. The proof involves entropy power inequality and is omitted.

We choose Λ_2 to be an exponentially good channel code in the sense defined in Section 2.1 (also see [10]). For such lattices, the probability of decoding error $P_e \to 0$ exponentially fast if $(Z^n + e_q)$ is Gaussian. The analysis in [11] showed that if $(Z^n + e_q)$ tends to a white Gaussian noise vector, the effect on P_e of the deviation from Gaussianity is sub-exponential and the overall error behavior is asymptotically the same. This implies that the reconstruction error $Z^n - \hat{Z}$ tends in probability to the random vector N defined in equation (15). Since all random vectors involved have finite normalized second moment, this convergence in probability implies convergence in second moment as well, i.e., $\frac{1}{n}\mathbb{E} \parallel Z^n - \hat{Z} \parallel^2 \to D$. Averaged over the random dithers U_1 and U_2, we have shown that the appropriate distortion is achieved. Hence there must exist a pair of deterministic dithers that also achieve distortion D and we have proved the claim of Theorem 1.

[1] We refer to this quantity also as the normalized second moment of the random vector N. This should not be confused with the normalized second moment of a lattice as defined in Section 2.1.

4 Distributed Source Coding for the K Source Case

In this section, we consider the case of reconstructing a linear function of an arbitrary number of sources. In the case of two sources, the two strategies used in Theorems 1 and 2 were direct reconstruction of the function Z and estimating the function from noisy versions of the sources respectively. In the presence of more than two sources, a host of strategies which are a combination of these two strategies become available. Some sets of sources might use the "correlated" binning strategy of Theorem 1 while others might use the "independent" binning strategy of Theorem 2. The union of the rate-distortion tuples achieved by all such schemes gives an achievable rate region for the problem.

Let the sources be given by $X_1, X_2, \ldots, X_K$ which are jointly Gaussian. The decoder wishes to reconstruct a linear function given by $Z = \sum_{i=1}^{K} c_i X_i$ with squared error fidelity criterion. The performance limit $\mathcal{RD}$ is given by the set of all rate-distortion tuples $(R_1, R_2, \ldots, R_K, D)$ that are achievable in the sense defined in Section 3.

For any set $A \subset \{1, \ldots, K\}$, let X_A denote those sources whose indices are in A, i.e., $X_A \triangleq \{X_i : i \in A\}$. Let Z_A be defined as $\sum_{i \in A} c_i X_i$. Let Θ be a partition of $\{1, \ldots, K\}$ with $\theta = |\Theta|$. Let $\pi_\Theta : \Theta \to \{1, \ldots, \theta\}$ be a permutation. One can think of π_Θ as ordering the elements of Θ. Each set of sources $X_A, A \in \Theta$ are decoded simultaneously at the decoder with the objective of reconstructing Z_A. The order of decoding is given by $\pi_\Theta(A)$ with the lower ranked sets of sources decoded earlier. Let $\mathcal{Q} = (q_1, \ldots, q_K) \in \mathbb{R}_+^K$ be a tuple of positive reals. For any partition Θ and ordering π_Θ, let us define recursively a positive-valued function $\sigma_\Theta^2 : \Theta \to \mathbb{R}^+$ as $\sigma_\Theta^2(A) = \mathbb{E}\left[(Z_A - f_A(S_A))^2\right]$ where $f_A(S_A) = \mathbb{E}(Z_A|S_A)$, $S_A = \{Z_B + Q_B : B \in \Theta, \pi_\Theta(B) < \pi_\Theta(A)\}$ and $\{Q_A : A \in \Theta\}$ is a collection of $|\Theta|$ independent zero-mean Gaussian random variables with variances given by $q_A = \mathrm{Var}(Q_A) \triangleq \sum_{i \in A} q_i$, and this collection is independent of the sources. Let $f(\{Z_A + Q_A : A \in \Theta\}) \triangleq \mathbb{E}\,(Z|\{Z_A + Q_A : A \in \Theta\})$.

Theorem 3. *For a given tuple of sources $X_1, \ldots, X_K$ and tuple of real numbers $(c_1, c_2, \ldots, c_K)$, we have $\mathcal{RD}_{in}^* \subset \mathcal{RD}$, where * denotes convex closure and*

$$\mathcal{RD}_{in} = \bigcup_{\Theta, \pi_\Theta, \mathcal{Q}} \Bigg\{ (R_1, \ldots, R_K, D) : R_i \geq \frac{1}{2} \log \frac{\sigma_\Theta^2(A) + q_A}{q_i} \text{ for } i \in A,$$

$$D \geq \mathbb{E}\left[\left(Z - f\left(\{Z_A + Q_A : A \in \Theta\}\right)\right)^2\right] \quad \Bigg\} \tag{19}$$

Proof. This inner bound to the optimal rate region can be proved by demonstrating a coding scheme that achieves the rates given. As in Section 3.2, we use "correlated" binning based on lattice codes. The basic idea of the proof is to use high dimensional lattices to mimic the Gaussian test channels used in the description of Theorem 3. The details are omitted. We remark that the general K-user rate region described above can be used to re-derive Theorems 1 and 2 by appropriate choices of the partition Θ.

5 Comparison of the Rate Regions

In this section, we compare the rate regions of the lattice based coding scheme given in Theorem 1 and the Berger-Tung based coding scheme given in Theorem 2 for the case of two users. The function under consideration is $Z = X_1 - cX_2$. To demonstrate the performance of the lattice binning scheme, we choose the sum rate of the two encoders as the performance metric.

In Fig. 1, we compare the sum-rates of the two schemes for $\rho = 0.8$ and $c = 0.8$. Fig. 1 shows that for small distortion values, the lattice scheme achieves a smaller sum rate than the Berger-Tung based scheme. We observe that the lattice based scheme performs better than the Berger-Tung based scheme for small distortions provided ρ is sufficiently high and c lies in a certain interval. Fig. 2 is a contour plot that illustrates this in detail. The contour labeled R encloses that region in which the pair (ρ, c) should lie for the lattice binning scheme to achieve a sum rate that is at least R units less than the sum rate of the Berger-Tung scheme for some distortion D. Observe that we get improvements only for $c > 0$.

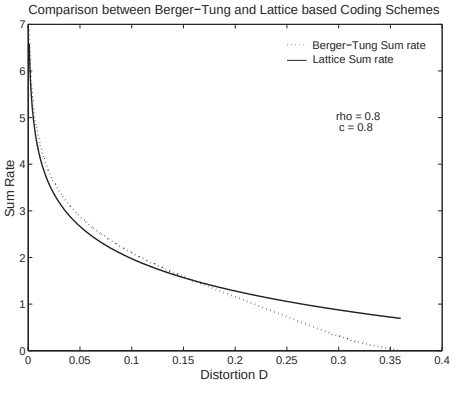


Fig. 1. Comparison of the sum-rates

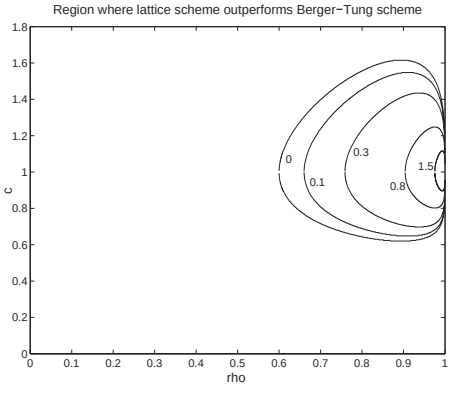


Fig. 2. (ρ, c) region for lower sum rate

6 Conclusion

We have thus demonstrated a lattice based coding scheme that directly encodes the linear function that the decoder is interested in instead of encoding the sources separately and estimating the function at the decoder. For the case of two users, it is seen that the lattice based coding scheme gives a lower sum-rate for certain values of ρ, c, D. Hence, using a combination of the lattice based and the Berger-Tung based coding schemes results in a better rate-region than using any one scheme alone. For the case of reconstructing a linear function of K sources, we have extended this concept to provide an inner bound to the optimal rate-distortion function. Some parts of the inner bound are achieved using a coding scheme that has the following structure: lattice vector quantization followed by "correlated" lattice-structured binning.

Acknowledgements

The authors would like to thank Dr. Ram Zamir and Dr. Uri Erez of Tel Aviv University for helpful discussions.

References

1. Gelfand, S., Pinsker, M.: Coding of Sources on the Basis of Observations with Incomplete Information. Problemy Peredachi Informatsii 15, 45–57 (1979)
2. Korner, J., Marton, K.: How to Encode the Modulo-Two Sum of Binary Sources. IEEE Trans. Inform. Theory 25, 219–221 (1979)
3. Csiszár, I., Korner, J.: Information Theory: Coding Theorems for Discrete Memoryless Systems. Academic Press, London (1981)
4. Han, T.S., Kobayashi, K.: A Dichotomy of Functions F(X,Y) of Correlated Sources (X,Y). IEEE Trans. on Inform. Theory 33, 69–76 (1987)
5. Ahlswede, R., Han, T.S: On Source Coding with Side Information via a Multiple-Access Channel and Related Problems in Multi-User Information Theory. IEEE Trans. on Inform. Theory 29, 396–412 (1983)
6. Berger, T.: Multiterminal Source Coding. Lectures presented at CISM summer school on the Inform. Theory approach to communications (1977)
7. Tung, S.-Y.: Multiterminal Source Coding. PhD thesis. Cornell University, Ithaca, NY (1978)
8. Wagner, A.B., Tavildar, S., Viswanath, P.: The Rate-Region of the Quadratic Gussian Two-Terminal Source-Coding Problem. arXiv:cs.IT/0510095
9. Zamir, R., Feder, M.: On Lattice Quantization Noise. IEEE Trans. Inform. Theory 42, 1152–1159 (1996)
10. Zamir, R., Shamai, S., Erez, U.: Nested Linear/Lattice Codes for Structured Multiterminal Binning. IEEE Trans. Inform. Theory 48, 1250–1276 (2002)
11. Erez, U., Zamir, R.: Achieving 1/2 log(1+SNR) on the AWGN Channel with Lattice Encoding and Decoding. IEEE Trans. Inform. Theory 50, 2293–2314 (2004)
12. Erez, U., Litsyn, S., Zamir, R.: Lattices Which Are Good for (Almost) Everything. IEEE Trans. Inform. Theory 51(10), 3401–3416 (2005)
13. Poltyrev, G.: On Coding Without Restrictions for the AWGN Channel. IEEE Trans. Inform. Theory 40, 409–417 (1994)
14. Krithivasan, D., Pradhan, S.S.: A Proof of the Existence of Good Nested Lattices, `http://www.eecs.umich.edu/techreports/systems/cspl/cspl-384.pdf`
15. Loeliger, H.A.: Averaging Bounds for Lattices and Linear Codes. IEEE Trans. Inform. Theory 43, 1767–1773 (1997)

Linear Complexity and Autocorrelation of Prime Cube Sequences

Young-Joon Kim, Seok-Yong Jin, and Hong-Yeop Song

Department of Electrical and Electronic Engineering
Yonsei University, Seoul, 121-749, Korea
{yj.kim, sy.jin, hysong}@yonsei.ac.kr

Abstract. We review a binary sequence based on the generalized cyclotomy of order 2 with respect to p^3, where p is an odd prime. Linear complexities, minimal polynomials and autocorrelation of these sequences are computed.

1 Introduction

Let $n \geq 2$ be a positive integer and Z_n^* be the multiplicative group of the integer ring Z_n. For a partition $\{D_i | i = 0, 1, \cdots, d-1\}$ of Z_n^*, if there exist elements $g_1, \cdots, g_d$ of Z_n^* satisfying $D_i = g_i D_0$ for all i where D_0 is a multiplicative subgroup of Z_n^*, the D_i are called *generalized cyclotomic classes* of order d. In 1998, Ding and Helleseth [1] introduced the new generalized cyclotomy with respect to $p_1^{e_1} \cdots p_t^{e_t}$ and defined a *balanced* binary sequence based on their own generalized cyclotomy, where $p_1, \cdots, p_t$ are distinct odd primes and $e_1, \cdots, e_t$ are positive integers. Before them, there have been lots of studies about cyclotomy, but they are only about ones with respect to p or p^2 or pq where p and q are distinct odd primes [1,4,7,8]. In [1] they also introduced how to construct a balanced binary sequence based on their generalized cyclotomy. Let it call the generalized cyclotomic sequences. Those sequences includes the binary quadratic residue sequences also known as Legendre Sequences because these sequences can be understood as the generalized cyclotomic sequences with respect to p.

In 1998, C. Ding [4] presented some cyclotomy sequences with period p^2 which are not balanced. They are defined in a slightly different way from the generalized cyclotomic sequences with respect to p^2. In that paper, he calculated the linear complexities with minor errors. Y.-H. Park and others [5] corrected the errors. The linear complexity of the sequence is not so good. In general, the linear complexity of a sequence is considered as good when it is not less than half of the period of the sequence. Recently, in [7], Yan *et al.* calculated the linear complexity and autocorrelation of generalized cyclotomic sequences of order 2 with respect to p^2.

In this paper, we compute the linear complexity and autocorrelation of the generalized cyclotomic sequences with respect to p^3. Hereafter we will call these sequences as *prime cube sequences*.

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 188–197, 2007.
© Springer-Verlag Berlin Heidelberg 2007

2 Prime Cube Sequences

Let p be an odd prime. Let g be a primitive root of p^2. Then it's well known that g is also a primitive root of p^k for $k \geq 1$[2]. The order of g modulo p is $p-1$, the order of g modulo p^2 is $p(p-1)$ and the order of g modulo p^3 is $p^2(p-1)$.

Define

$$\begin{array}{llll} D_0^{(p)} = (g^2) \pmod p & & D_1^{(p)} = gD_0^{(p)} \pmod p \\ D_0^{(p^2)} = (g^2) \pmod{p^2} & & D_1^{(p^2)} = gD_0^{(p^2)} \pmod{p^2} \\ D_0^{(p^3)} = (g^2) \pmod{p^3} & & D_1^{(p^3)} = gD_0^{(p^3)} \pmod{p^3} \end{array}$$

Then $Z_p^* = D_0^{(p)} \cup D_1^{(p)}$,$Z_{p^2}^* = D_0^{(p^2)} \cup D_1^{(p^2)}$ and $Z_{p^3}^* = D_0^{(p^3)} \cup D_1^{(p^3)}$. For $i = 0, 1, 2$, the $D_j^{(p^i)}$ are called generalized cyclotomic classes of order 2 with respect to p^j. Note that

$$Z_{p^3} = D_0^{(p^3)} \cup D_1^{(p^3)} \cup pD_0^{(p^2)} \cup pD_1^{(p^2)} \cup p^2D_0^{(p)} \cup p^2D_1^{(p)} \cup \{0\}.$$

Here and hereafter, $\frac{p^3}{p^i}D_j^{(p^i)}$ are sets of elements obtained by multiplying $\frac{p^3}{p^i}$ to the elements of $D_j^{(p^i)}$ over Z_{p^3} for $i = 0, 1, 2$ and $j = 0, 1$.

In [1], the authors define the binary prime cube sequence $\{s(n)\}$ as follows[1]:

$$s(i) = \begin{cases} 0, & \text{if } (i \bmod p^3) \in C_0 \\ 1, & \text{if } (i \bmod p^3) \in C_1 \ . \end{cases} \tag{1}$$

where $C_0 = \bigcup_{d|p^3, d>1} \frac{p^3}{d}D_0^{(d)}$ and $C_1 = \{0\} \cup \bigcup_{d|p^3, d>1} \frac{p^3}{d}D_1^{(d)}$.

3 Linear Complexity and Minimal Polynomial

Let $\{s(n)\}$ be a sequence of period L over a field F. The linear complexity of $\{s(n)\}$ is defined to be least positive integer l such that there are constants $c_0 = 1, c_1, \cdots, c_l \in F$ satisfying

$$-s(i) = c_1 s(i-1) + c_2 s(i-2) + \cdots + c_l s(i-l) \ \text{ for all } l \leq i < L$$

The polynomial $c(x) = c_0 + c_1 x + \cdots + c_l x^l$ is called a minimal polynomial of $\{s(n)\}$. Let $\{s(n)\}$ be a sequence of period L over a field F, and $S(x) = s(0) + s(1)x + \cdots + s(L-1)x^{L-1}$. It is well known that[3]

1. the mimimal polynomial of $\{s(n)\}$ is given by

$$c(x) = (x^L - 1)/\gcd(x^L - 1, S(x))$$

2. the linear complexity of $\{s(n)\}$ is given by

$$C_L = L - \deg(\gcd(x^L - 1, S(x)))$$

Lemma 1. *For $a \in Z_{p^3}^*$ and $1 \le i \le 3$*

$$aD_0^{(p^i)} = \begin{cases} D_0^{(p^i)}, & \text{if } a \in D_0^{(p^i)} \\ D_1^{(p^i)}, & \text{if } a \in D_1^{(p^i)} \end{cases}, \quad aD_1^{(p^i)} = \begin{cases} D_1^{(p^i)}, & \text{if } a \in D_0^{(p^i)} \\ D_0^{(p^i)}, & \text{if } a \in D_1^{(p^i)} \end{cases}.$$

Proof. It can be proved in the same way as [4].

Lemma 2. *Let b be any integer. Then $D_i^{(p^3)}+bp = D_i^{(p^3)}$ and $D_i^{(p^2)}+bp = D_i^{(p^2)}$ for $i = 0, 1$.*

Proof. It can also be proved in the same way as [4].

Lemma 3. *$-1 \pmod{p^3} \in D_0^{(p^3)}$ if and only if $-1 \pmod{p^2} \in D_0^{(p^2)}$ if and only if $-1 \pmod{p} \in D_0^{(p)}$ if and only if $p \equiv 1 \pmod 4$.*

Proof. It is well known that $-1 \pmod p \in D_0^{(p)}$ if and only if $p \equiv 1 \pmod 4$[2]. Using Lemma 2, we can show $-1 \pmod p \in D_0^{(p)}$ implies $-1 \pmod{p^2} \in D_0^{(p^2)}$ and $-1 \pmod{p^3} \in D_0^{(p^3)}$. The converse is obvious.

Lemma 4. *$2 \in D_i^{(p^3)}$ if and only if $2 \in D_i^{(p^2)}$ if and only if $2 \in D_i^{(p)}$ for $i = 0, 1$.*

Proof. It can be proved in the same way as [4].

Let m be the order of 2 modulo p^3 and θ a primitive p^3th root of unity in $GF(2^m)$. Define

$$S(x) = \sum_{i \in C_1} x^i = 1 + \Big(\sum_{i \in D_1^{(p^3)}} + \sum_{i \in pD_1^{(p^2)}} + \sum_{i \in p^2 D_1^{(p)}} \Big) x^i \in GF(2)[x].$$

Then $S(x)$ is generating function of the prime cube sequence $\{s(n)\}$ defined before. To compute $S(\theta)$, we use the generalized cyclotomic numbers of order 2 with respect to p^i for $i \ge 1$ defined by

$$(i,j)_{p^k} = |(D_i^{(p^k)} + 1) \cap D_j^{(p^k)}| \quad i, j = 0, 1, \text{ and } k = 0, 1, 2. \tag{2}$$

Lemma 5. *[1] If $p \equiv 3 \pmod 4$, then*

$$(1,0)_{p^k} = (0,0)_{p^k} = (1,1)_{p^k} = \frac{p^{k-1}(p-3)}{4}, \text{ and } (0,1)_{p^k} = \frac{p^{k-1}(p+1)}{4}.$$

If $p \equiv 1 \pmod 4$, then

$$(0,1)_{p^k} = (1,0)_{p^k} = (1,1)_{p^k} = \frac{p^{k-1}(p-1)}{4}, \text{ and } (0,0)_{p^k} = \frac{p^{k-1}(p-5)}{4}.$$

Note that

$$0=\theta^{p^3}-1=(\theta^{p^2})^p-1=(\theta^{p^2}-1)(1+\theta^{p^2}+\theta^{2p^2}+\cdots+\theta^{(p-1)p^2}). \tag{3}$$

It follows that

$$1+\theta^{p^2}+\theta^{2p^2}+\cdots+\theta^{(p-1)p^2}=1+\sum_{i\in p^2D_0^{(p)}}\theta^i+\sum_{i\in p^2D_1^{(p)}}\theta^i=0. \tag{4}$$

(3) can be rewritten as follows:

$$0=\theta^{p^3}-1=(\theta^p)^{p^2}-1=(\theta^p-1)(1+\theta^p+\cdots+\theta^{(p^2-1)p}).$$

It follows that

$$1+\theta^p+\cdots+\theta^{(p^2-1)p}=1+\sum_{i\in p^2D_0^{(p)}\cup p^2D_1^{(p)}\cup pD_0^{(p^2)}\cup pD_1^{(p^2)}}\theta^i=0. \tag{5}$$

From (4) and (5), we obtain

$$\sum_{i\in pD_0^{(p^2)}}\theta^i=\sum_{i\in pD_1^{(p^2)}}\theta^i. \tag{6}$$

Since $\sum_{i=0}^{p^3-1}\theta^i=0$, by (5) we obtain

$$\sum_{i\in D_0^{(p^3)}}\theta^i=\sum_{i\in D_1^{(p^3)}}\theta^i. \tag{7}$$

Assume $\theta_1=\theta^p,\theta_2=\theta^{p^2}$, then θ_1 is a primitive p^2th root of unity and θ_2 is a primitive pth root of unity in $GF(2^m)$. Define

$$t_1(\theta_1)=\sum_{i\in D_1^{(p^2)}}\theta_1^i \quad\text{and}\quad t_2(\theta_2)=\sum_{i\in D_1^{(p)}}\theta_2^i.$$

Lemma 6. *[5]* $\sum_{i\in pZ_p}\theta_1^i+\sum_{i\in D_1^{(p^2)}}\theta_1^i=0$ *if* p *is an odd prime.*

Lemma 7. $\sum_{i\in D_0^{(p^2)}}\theta_1^i=\sum_{i\in D_1^{(p^2)}}\theta_1^i=t_1(\theta_1)=0.$

Proof. From (4),(6) and Lemma 6, obvious.

Lemma 8. *[6]* $t_2(\theta_2)\in\{0,1\}$ *if and only if* $2\in D_0^{(p)}$ *if and only if* $p\equiv\pm1\pmod 8$

Lemma 9. *Let the symbols be the same as before,*

$$S(\theta^a)=\begin{cases}\frac{p+1}{2} \pmod 2, & \text{if } a=0\\ S(\theta), & \text{if } a\in D_0^{(p^3)}\\ S(\theta)+1, & \text{if } a\in D_1^{(p^3)}\\ \frac{p+1}{2}+t_2(\theta_2), & \text{if } a\in pD_0^{(p^2)}\\ \frac{p-1}{2}+t_2(\theta_2), & \text{if } a\in pD_1^{(p^2)}\\ 1+t_2(\theta_2), & \text{if } a\in p^2D_0^{(p)}\\ t_2(\theta_2), & \text{if } a\in p^2D_1^{(p)}.\end{cases}$$

Proof. For the case $a = 0$, we have $S(\theta^a) = S(1) = \frac{p^3+1}{2} \equiv \frac{p+1}{2} \pmod 2$. If $a \in D_0^{(p^3)}$, by definition there is an integer s such that $a = g^{2s}$. It follows that

$$\begin{aligned} aD_1^{(p^3)} &= \{g^{2s+2t+1} | t = 0, 1, \cdots, p^2(p-1) - 1\} = D_1^{(p^3)} \\ apD_1^{(p^2)} &= p\{g^{2s+2t+1} | t = 0, 1, \cdots, p(p-1) - 1\} = pD_1^{(p^2)} \\ ap^2 D_1^{(p)} &= p^2\{g^{2s+2t+1} | t = 0, 1, \cdots, (p-1) - 1\} = p^2 D_1^{(p^2)}. \end{aligned}$$

Hence

$$S(\theta^a) = 1 + (\sum_{i \in D_1^{(p^3)}} + \sum_{i \in pD_1^{(p^2)}} + \sum_{i \in p^2 D_1^{(p)}})\theta^{ai} = 1 + (\sum_{i \in D_1^{(p^3)}} + \sum_{i \in pD_1^{(p^2)}} + \sum_{i \in p^2 D_1^{(p)}})\theta^{i} = S(\theta).$$

If $a \in D_1^{(p^3)}$, then $aD_1^{(p^3)} = D_0^{(p^3)}, apD_1^{(p^2)} = pD_0^{(p^2)}, ap^2 D_1^{(p)} = p^2 D_0^{(p^2)}$. By (4), (6) and (7)

$$S(\theta^a) = 1 + (\sum_{i \in D_0^{(p^3)}} + \sum_{i \in pD_0^{(p^2)}} + \sum_{i \in p^2 D_0^{(p)}})\theta^{i} = S(\theta) + 1.$$

Note that $D_1^{(p^3)} \bmod p = D_1^{(p)}$, $|D_1^{(p^3)}| = p^2 |D_1^{(p)}|$, $\theta_1^{p^2} = 1$ and $\theta_2^p = 1$.
For $a = a_1 p, a_1 \in Z_{p^2}^* = D_0^{(p^2)} \cup D_1^{(p^2)}$, we have

$$\begin{aligned} S(\theta^a) &= 1 + (\sum_{i \in D_1^{(p^3)}} + \sum_{i \in pD_1^{(p^2)}} + \sum_{i \in p^2 D_1^{(p)}})\theta^{ai} \\ &= 1 + \sum_{i \in D_1^{(p^3)}} \theta^{a_1 pi} + \sum_{i \in pD_1^{(p^2)}} \theta^{a_1 pi} + \sum_{i \in p^2 D_1^{(p)}} \theta^{a_1 pi} \\ &= 1 + \sum_{i \in a_1 D_1^{(p^3)}} \theta_1^i + \sum_{i \in a_1 D_1^{(p^2)}} \theta_2^i + \frac{p-1}{2}. \end{aligned}$$

If $a_1 \in D_0^{(p^2)}$, $a_1 D_1^{(p^3)} = D_1^{(p^3)}$ and $a_1 D_1^{(p^2)} = D_1^{(p^2)}$. we have

$$\begin{aligned} S(\theta^a) &= \frac{p+1}{2} + \sum_{i \in D_1^{(p^3)}} \theta_1^i + \sum_{i \in D_1^{(p^2)}} \theta_2^i = \frac{p+1}{2} + p \sum_{i \in D_1^{(p^2)}} \theta_1^i + p \sum_{i \in D_1^{(p)}} \theta_2^i \\ &= \frac{p+1}{2} + t_1(\theta_1) + t_2(\theta_2) = \frac{p+1}{2} + t_2(\theta_2). \end{aligned}$$

If $a_1 \in D_1^{(p^2)}$, $a_1 D_1^{(p^3)} = D_0^{(p^3)}$ and $a_1 D_1^{(p^2)} = D_0^{(p^2)}$. we have

$$\begin{aligned} S(\theta^a) &= \frac{p+1}{2} + \sum_{i \in D_0^{(p^3)}} \theta_1^i + \sum_{i \in D_0^{(p^2)}} \theta_2^i \\ &= \frac{p+1}{2} + t_1(\theta_1) + 1 + t_2(\theta_2) = \frac{p-1}{2} + t_2(\theta_2). \end{aligned}$$

For $a = a_2 p^2, a_2 \in Z_p^* = D_0^{(p)} \cup D_1^{(p)}$, we have

$$\begin{aligned} S(\theta^a) &= 1 + (\sum_{i \in D_1^{(p^3)}} + \sum_{i \in pD_1^{(p^2)}} + \sum_{i \in p^2 D_1^{(p)}})\theta^{ai} \\ &= 1 + \sum_{i \in a_2 D_1^{(p^3)}} \theta^{p^2 i} + \sum_{i \in D_1^{(p^2)}} \theta_1^{a_2 p^2 i} + \sum_{i \in D_1^{(p)}} \theta_2^{a_2 p^2 i} \\ &= 1 + \sum_{i \in a_2 D_1^{(p^3)}} \theta_2^i + \frac{p^2 - p}{2} + \frac{p-1}{2}. \end{aligned}$$

If $a_2 \in D_0^{(p)}$, $a_2 D_1^{(p^3)} = D_1^{(p^3)}$ and $a_2 D_1^{(p^2)} = D_1^{(p^2)}$. we have

$$S(\theta^a) = \frac{p^2+1}{2} + \sum_{i \in D_1^{(p^3)}} \theta_2^i = \frac{p^2+1}{2} + p^2 \sum_{i \in D_1^{(p)}} \theta_2^i = 1 + t_2(\theta_2).$$

If $a_2 \in D_1^{(p^2)}$, $a_2 D_1^{(p^3)} = D_0^{(p^3)}$ and $a_2 D_1^{(p^2)} = D_0^{(p^2)}$. we have

$$S(\theta^a) = \frac{p^2+1}{2} + \sum_{i \in D_0^{(p^3)}} \theta_2^i = \frac{p^2+1}{2} + p^2 \sum_{i \in D_0^{(p)}} \theta_2^i = t_2(\theta_2).$$

Define $d_i^{(p^3)}(x) = \prod_{a \in D_i^{(p^3)}} (x - \theta^a)$, $d_i^{(p^2)}(x) = \prod_{a \in D_i^{(p^2)}} (x - \theta_1^a)$ and $d_i^{(p)}(x) = \prod_{a \in D_i^{(p)}} (x - \theta_2^a)$, $i = 0, 1$. Then

$$x^{p^3} - 1 = (x-1) d_0^{(p)}(x) d_1^{(p)}(x) d_0^{(p^2)}(x) d_1^{(p^2)}(x) d_0^{(p^3)}(x) d_1^{(p^3)}(x).$$

Lemma 10. *$d_i^{(p)}(x), d_i^{(p^2)}(x), d_i^{(p^3)}(x) \in GF(2)[x]$ if and only if $p \equiv \pm 1$ mod 8.*

Proof. Almost the same proof in [4] can be applied . If $p \equiv \pm 1 \bmod 8$, from Lemma 4 and 8, $2 \in D_0^{(p)} \cap D_0^{(p^2)} \cap D_0^{(p^3)}$. Then for $i = 0, 1, 2$, we have

$$(d_i^{(p^i)}(x))^2 = \prod_{a \in D_i^{(p^i)}} x^2 - \theta^{2p^i a}) = \prod_{a \in 2D_i^{(p^i)}} (x^2 - \theta^{p^i a}) = \prod_{a \in D_i^{(p^i)}} (x^2 - \theta^{p^i a}) = d_i^{(p^i)}(x^2).$$

Thus $d_i^{(p^i)}(x) \in GF(2)[x]$, $i = 0, 1, 2$. If $p \equiv \pm 3 \bmod 8$, from Lemma 4 and 8, $2 \in D_1^{(p)} \cap D_1^{(p^2)} \cap D_1^{(p^3)}$. Then for $i = 0, 1, 2$, we have

$$(d_i^{(p^i)}(x))^2 = \prod_{a \in D_{i+1 (mod\ 2)}^{(p^i)}} (x^2 - \theta^{p^i a}) = d_{i+1 (mod\ 2)}^{(p^i)}(x^2) \neq d_i^{(p^i)}(x^2).$$

Hence $d_i^{(p^i)}(x) \notin GF(2)[x]$, $i = 0, 1, 2$.

Theorem 1. *Let p be an odd prime and $\{s(n)\}$ be a prime cube sequence of period p^3. Then the linear complexity C_L of $\{s(n)\}$ is as follows:*

$$C_L = \begin{cases} \frac{p^3+1}{2}, & \text{if } p \equiv 1 \bmod 8 \\ p^3-1, & \text{if } p \equiv 3 \bmod 8 \\ p^3, & \text{if } p \equiv 5 \bmod 8 \\ \frac{p^3-1}{2}, & \text{if } p \equiv 7 \bmod 8. \end{cases}$$

Proof. If $p \equiv 1 \bmod 8$, from Lemmas 8, $t_2(\theta_2) \in \{0,1\}$. Furthermore, since $2 \in D_0^{(p)} \cap D_0^{(p^2)} \cap D_0^{(p^3)}$ by Lemma 4 and 8, $S(\theta^2) = S(\theta)$. Hence, $S(\theta) \in \{0,1\}$. Applying Lemma 9, we have

$$c(x) = \frac{x^{p^3}-1}{\gcd(x^{p^3}-1, S(x))} = \begin{cases} (x-1)d_1^{(p^3)}(x)d_0^{(p^2)}(x)d_0^{(p)}(x), & \text{if } (S(\theta),t_2(\theta_2)) = (0,0) \\ (x-1)d_1^{(p^3)}(x)d_1^{(p^2)}(x)d_1^{(p)}(x), & \text{if } (S(\theta),t_2(\theta_2)) = (0,1) \\ (x-1)d_0^{(p^3)}(x)d_0^{(p^2)}(x)d_0^{(p)}(x), & \text{if } (S(\theta),t_2(\theta_2)) = (1,0) \\ (x-1)d_0^{(p^3)}(x)d_1^{(p^2)}(x)d_1^{(p)}(x), & \text{if } (S(\theta),t_2(\theta_2)) = (1,1) \end{cases}$$

It follows that $C_L = \deg(c(x)) = 1 + \frac{p^3-p^2}{2} + \frac{p^2-p}{2} + \frac{p-1}{2} = \frac{p^3+1}{2}$.

For the cases of $p \equiv 3, 5$ and $7 \bmod 8$, we can reach easily by similar procedure with the case $p \equiv 1 \bmod 8$.

4 Autocorrelation

The periodic autocorrelation of a binary sequence $\{s(n)\}$ of period N is defined by $C_s(\tau) = \sum_{n=0}^{L} (-1)^{s(n+\tau)-s(n)}$ where $0 \le \tau < L$. Define $d_s(i,j;\tau) = |C_i \cap (C_j + \tau)|, \quad 0 \le \tau < L,\ i,j = 0,1$

Theorem 2. *Let p be an odd prime. Then the autocorrelation profile of the binary prime cube sequence of period p^3 which is defined at (1) is as follows:*

1. $p \equiv 1 \pmod 4$

$$C_s(\tau) = \begin{cases} p^3, & \tau = 0 \pmod{p^3} \\ p^3-p-3, & \tau \in p^2 D_0^{(p)} \\ p^3-p+1, & \tau \in p^2 D_1^{(p)} \\ p^3-p^2-p-2, & \tau \in p D_0^{(p^2)} \\ p^3-p^2-p+2, & \tau \in p D_1^{(p^2)} \\ -p^2-2, & \tau \in D_0^{(p^3)} \\ -p^2+2, & \tau \in D_1^{(p^3)} \end{cases}$$

2. $p \equiv 3 \pmod 4$

$$C_s(\tau) = \begin{cases} p^3, & \tau = 0 \pmod{p^3} \\ p^3-p-1, & \tau \in p^2 D_0^{(p)} \cup p^2 D_1^{(p)} \\ p^3-p^2-p, & \tau \in p D_0^{(p^2)} \cup p D_1^{(p^2)} \\ -p^2, & \tau \in D_0^{(p^3)} \cup D_1^{(p^3)}. \end{cases}$$

Proof. Since $C_s(\tau) = p^3 - 4d_s(1,0;\tau)$, we need to calculate $d_s(1,0;\tau)$. Note that

$$\begin{aligned} d_s(1,0;\tau) =& |C_1 \cap (C_0 + \tau)| \\ =& |C_1 \cap (p^2 D_0^{(p)} + \tau)| + |C_1 \cap (pD_0^{(p^2)} + \tau)| + |C_1 \cap (D_0^{(p^3)} + \tau)| \end{aligned} \tag{8}$$

Denote the first, the second and the third term in (8) as $A(\tau)$, $B(\tau)$ and $C(\tau)$, respectively. To begin with, we are going to compute $A(\tau)$. Note that

$$\begin{aligned} A(\tau) = |C_1 \cap (p^2 D_0^{(p)} + \tau)| = |\{0\} \cap (p^2 D_0^{(p)} + \tau)| + |p^2 D_1^{(p)} \cap (p^2 D_0^{(p)} + \tau)| \\ + |pD_1^{(p^2)} \cap (p^2 D_0^{(p)} + \tau)| + |D_1^{(p^3)} \cap (p^2 D_0^{(p)} + \tau)|. \end{aligned} \tag{9}$$

Denote the first, the second, the third and the fourth term in (9) as $A_1(\tau)$, $A_2(\tau)$, $A_3(\tau)$ and $A_4(\tau)$, respectively. Let us compute $A_1(\tau)$ first. When $\tau = 0$, $A_1(\tau) = |\{0\} \cap p^2 D_0^{(p)}| = 0$. When $\tau \in pD_i^{(p^2}$ for $i = 0, 1$, by Lemma 2, any element of $p^2 D_0^{(p)} + \tau$ is an element of $pD_i^{(p^2)}$ for $i = 0, 1$, respectively. Similarly, when $\tau \in D_i^{(p^3)}$ for $i = 0, 1$, any element of $p^2 D_0^{(p)} + \tau$ is an element of $D_i^{(p^3)}$ for $i = 0, 1$, respectively. Therefore, when $\tau \in \{0\} \cup pD_0^{(p^2)} \cup pD_1^{(p^2)} \cup D_0^{(p^3)} \cup D_1^{(p^3)}$, $A_1(\tau) = 0$. Next thing to do is to compute the value of $A_1(\tau)$ when τ belongs to the set $p^2 D_0^{(p)} \cup p^2 D_1^{(p)}$. From Lemma 1 and 3, if $p \equiv 1 \bmod 4$, $\tau \in p^2 D_i^{(p)}$ implies $-\tau \in p^2 D_i^{(p)}$ for $i = 0, 1$, respectively. Hence, in this case, $A_1(\tau) = 1$ if $\tau \in p^2 D_0^{(p)}$ and $A_1(\tau) = 0$ if $\tau \in p^2 D_1^{(p)}$. Likewise if $p \equiv 1 \bmod 4$, $\tau \in p^2 D_i^{(p)}$ implies $-\tau \in p^2 D_{i+1 \bmod 2}^{(p)}$ for $i = 0, 1$, respectively. Hence, $A_1(\tau) = 0$ if $\tau \in p^2 D_0^{(p)}$ and $A_1(\tau) = 1$ if $\tau \in p^2 D_1^{(p)}$. Summarizing these, we have

$$A_1(\tau) = \begin{cases} 0, \tau \in \{0\} \cup pD_0^{(p^2)} \cup pD_1^{(p^2)} \cup D_0^{(p^3)} \cup D_1^{(p^3)} \\ 1, \tau \in p^2 D_0^{(p)} \text{ and } p \equiv 1 \bmod 4 \\ 0, \tau \in p^2 D_0^{(p)} \text{ and } p \equiv 3 \bmod 4 \\ 0, \tau \in p^2 D_1^{(p)} \text{ and } p \equiv 1 \bmod 4 \\ 1, \tau \in p^2 D_1^{(p)} \text{ and } p \equiv 3 \bmod 4 \end{cases} \tag{10}$$

Next let us consider $A_2(\tau)$. Similarly $A_2(\tau) = 0$ if $\tau \in \{0\} \cup pD_0^{(p^2)} \cup pD_1^{(p^2)} \cup D_0^{(p^3)} \cup D_1^{(p^3)}$. When $\tau \in p^2 D_0^{(p)} \cup p^2 D_1^{(p)}$, $A_2(\tau) = |p^2 D_1^{(p)} \cap (p^2 D_0^{(p)} + \tau)| = |p^2 D_1^{(p)} \cap (p^2 D_0^{(p)} + p^2 a)|$ for some $a \in D_0^{(p)} \cup D_1^{(p)}$. Therefore $A_2(\tau) = |D_1^{(p)} \cap (D_0^{(p)} + a)| = |a^{-1} D_1^{(p)} \cap (a^{-1} D_0^{(p)} + 1)|$ and by Lemma 1 and the definition of the generalized cyclotomic numbers of order 2 with respect to p, we have

$$A_2(\tau) = \begin{cases} 0, & \tau \in \{0\} \cup pD_0^{(p^2)} \cup pD_1^{(p^2)} \cup D_0^{(p^3)} \cup D_1^{(p^3)} \\ (0,1)_p, & \tau \in p^2 D_0^{(p)} \\ (1,0)_p, & \tau \in p^2 D_1^{(p)} \end{cases}.$$

In the case of $A_3(\tau)$, $A_3(\tau) = 0$ if $\tau \in \{0\} \cup D_0^{(p^3)} \cup D_1^{(p^3)}$ with the same reason as $A_1(\tau)$ and $A_2(\tau)$. If $\tau \in \cup pD_0^{(p^2)} \cup pD_1^{(p^2)}$, then for $i = 0, 1$, any element

of $p^2 D_i^{(p)} + \tau$ is a multiple of p^2 mod p^3 so that it can not be an element of $pD_1^{(p^2)}$. Thus, in these cases, $A_3(\tau) = 0$. In the case of $\tau \in pD_i^{(p^2)}$ for $i = 0, 1$, we have $p^2 D_0^{(p)} + \tau \subset pD_i^{(p^2)}$. Therefore, $A_3(\tau) = |\emptyset| = 0$ if $\tau \in pD_0^{(p^2)}$ and $A_3(\tau) = |p^2 D_0^{(p)} + \tau| = \frac{p-1}{2}$ if $\tau \in pD_1^{(p^2)}$. Similarly, we can compute $A_4(\tau)$. Summarizing these calculation, we have

$$A_3(\tau) = \begin{cases} 0, & \tau \in Z_{p^3} \setminus pD_1^{(p^2)} \\ \frac{p-1}{2}, & \tau \in pD_1^{(p^2)} \end{cases}, \quad A_4(\tau) = \begin{cases} 0, & \tau \in Z_{p^3} \setminus D_1^{(p^3)} \\ \frac{p-1}{2}, & \tau \in D_1^{(p^3)} \end{cases}.$$

Combining the results of $A_1(\tau)$, $A_2(\tau)$, $A_3(\tau)$ and $A_4(\tau)$, we have

$$A(\tau) = \sum_{1 \le i \le 4} A_i(\tau) = \begin{cases} 0, & \tau = 0 \\ \frac{p+3}{4}, & \tau \in p^2 D_0^{(p)} \text{ and } p \equiv 1 \bmod 4 \\ \frac{p+1}{4}, & \tau \in p^2 D_0^{(p)} \text{ and } p \equiv 3 \bmod 4 \\ \frac{p-1}{4}, & \tau \in p^2 D_1^{(p)} \text{ and } p \equiv 1 \bmod 4 \\ \frac{p+1}{4}, & \tau \in p^2 D_1^{(p)} \text{ and } p \equiv 3 \bmod 4 \\ 0, & \tau \in pD_0^{(p^2)} \\ \frac{p-1}{2}, & \tau \in pD_1^{(p^2)} \\ 0, & \tau \in D_0^{(p^3)} \\ \frac{p-1}{2}, & \tau \in D_1^{(p^3)} \end{cases} \tag{11}$$

Next we are going to compute $B(\tau)$ and $C(\tau)$. Note that

$$\begin{aligned} B(\tau) = {} & |\{0\} \cap (pD_0^{(p^2)} + \tau)| + |p^2 D_1^{(p)} \cap (pD_0^{(p^2)} + \tau)| \\ & + |pD_1^{(p^2)} \cap (pD_0^{(p^2)} + \tau)| + |D_1^{(p^3)} \cap (pD_0^{(p^2)} + \tau)|. \end{aligned} \tag{12}$$

$$\begin{aligned} C(\tau) = {} & |\{0\} \cap (D_0^{(p^3)} + \tau)| + |p^2 D_1^{(p)} \cap (D_0^{(p^3)} + \tau)| \\ & + |pD_1^{(p^2)} \cap (D_0^{(p^3)} + \tau)| + |D_1^{(p^3)} \cap (D_0^{(p^3)} + \tau)|. \end{aligned} \tag{13}$$

Denote the first, the second, the third and the fourth term in (12) as $B_1(\tau)$, $B_2(\tau)$, $B_3(\tau)$ and $B_4(\tau)$, respectively. Likewise denote the first, the second, the third and the fourth term in (13) as $C_1(\tau)$, $C_2(\tau)$, $C_3(\tau)$ and $C_4(\tau)$, respectively. With almost the same way, we can reach the following:

$p \equiv 1 \bmod 4$	$B_1(\tau)$	$B_2(\tau)$	$B_3(\tau)$	$B_4(\tau)$	$B(\tau)$
$\tau \in pD_0^{(p^2)}$	1	$\frac{p-1}{2}$	$(0,1)_{p^2}$	0	$\frac{p^2+p+2}{4}$
$\tau \in pD_1^{(p^2)}$	0	0	$(1,0)_{p^2}$	0	$\frac{p(p-1)}{4}$
$\tau \in D_1^{(p^3)}$	0	0	0	$\frac{p^2-p}{2}$	$\frac{p^2-p}{2}$
otherwise	0	0	0	0	0

(14)

$p \equiv 3 \bmod 4$	$B_1(\tau)$	$B_2(\tau)$	$B_3(\tau)$	$B_4(\tau)$	$B(\tau)$
$pD_0^{(p^2)}$	0	0	$(0,1)_{p^2}$	0	$\frac{p(p+1)}{4}$
$\tau \in pD_1^{(p^2)}$	1	$\frac{p-1}{2}$	$(1,0)_{p^2}$	0	$\frac{p^2-p+2}{4}$
$\tau \in D_1^{(p^3)}$	0	0	0	$\frac{p^2-p}{2}$	$\frac{p^2-p}{2}$
otherwise	0	0	0	0	0

(15)

By doing the same procedure repeatedly, we can reach the following:

$p \equiv 1 \bmod 4$	$C_1(\tau)$	$C_2(\tau)$	$C_3(\tau)$	$C_4(\tau)$	$C(\tau)$
$\tau \in D_0^{(p^3)}$	1	$\frac{p-1}{2}$	$\frac{p^2-p}{2}$	$(0,1)_{p^3}$	$\frac{p^3+p^2+2}{4}$
$\tau \in D_1^{(p^3)}$	0	0	0	$(1,0)_{p^3}$	$\frac{p^3-p^2}{4}$
otherwise	0	0	0	0	0

$p \equiv 3 \bmod 4$	$C_1(\tau)$	$C_2(\tau)$	$C_3(\tau)$	$C_4(\tau)$	$C(\tau)$
$\tau \in D_0^{(p^3)}$	0	0	0	$(0,1)_{p^3}$	$\frac{p^3+p^2}{4}$
$\tau \in D_1^{(p^3)}$	1	$\frac{p-1}{2}$	$\frac{p^2-p}{2}$	$(1,0)_{p^3}$	$\frac{p^3-p^2+2}{4}$
otherwise	0	0	0	0	0

(16)

Combining (11),(14), and (16), we can compute $d_s(1,0;\tau)$. Since $C_s(\tau) = p^3 - 4d_s(1,0;\tau)$, it completes the proof.

References

1. Ding, C., Helleseth, T.: New Generalized Cyclotomy and Its Application. Finite Fields and Their Applications 4, 140–166 (1998)
2. Burton, D.M.: Elementary Number Theory, 4th edn. McGraw-Hill, New York (1998)
3. Golomb, S.W.: Shift Register Sequences, Revised edn. Aegean Park Press, Laguna Hills (1982)
4. Ding, C.: Linear Complexity of Some Generalized Cyclotomic Sequences. Int. J. Algebra and Computation 8, 431–442 (1998)
5. Park, Y.H., Hong, D., Chun, E.: On the Linear Complexity of Some Generalized Cyclotomic Sequences. Int. J. Algebra and Computation 14, 431–439 (2004)
6. Cusick, T., Ding, C., Renvall, A.: Stream Ciphers and Number Theory. Elservier Science, Amsterdam (1998)
7. Yan, T., Sun, R., Xiao, G.: Autocorrelation and Linear Complexity of the New Generalized Cyclotomic Sequences. IEICE Trans. Fundamentals E90-A, 857–864 (2007)
8. Bai, E., Liu, X., Xiao, G.: Linear Complexity of New Generalized Cyclotomic Sequences of Order Two of Length pq. IEEE Trans. Inform. Theory 51, 1849–1853 (2005)

The "Art of Trellis Decoding" Is NP-Hard*

Navin Kashyap

Dept. Mathematics and Statistics,
Queen's University, Kingston, ON, K7L 3N6, Canada
nkashyap@mast.queensu.ca

Abstract. Given a linear code $\mathcal{C}$, the fundamental problem of trellis decoding is to find a coordinate permutation of $\mathcal{C}$ that yields a code $\mathcal{C}'$ whose minimal trellis has the least state-complexity among all codes obtainable by permuting the coordinates of $\mathcal{C}$. By reducing from the problem of computing the pathwidth of a graph, we show that the problem of finding such a coordinate permutation is NP-hard, thus settling a long-standing conjecture.

1 Introduction

Maximum-likelihood (ML) decoding of a linear code can be implemented using the Viterbi algorithm on a trellis representation of the code. The run-time complexity of such an implementation depends on the complexity (size) of the trellis representation, and so it is desirable to find, for a given code $\mathcal{C}$, a low-complexity trellis representing $\mathcal{C}$. The theory of trellis representations of a linear code is well understood, and we refer the reader to the review by Vardy [11] for an excellent survey of this theory. A fundamental result of this theory is that a linear code has a unique minimal trellis that simultaneously minimizes several important measures of trellis complexity, including the number of states, the number of edges, and the so-called state-complexity of the trellis. There are several efficient algorithms known for determining the minimal trellis for a given linear code (again, see [11] and the references therein).

It is a somewhat surprising fact that permuting the coordinates of a code can result in a drastic change in the complexity of the minimal trellis. To be precise, if $\mathcal{C}'$ is a code obtained by permuting the coordinates of $\mathcal{C}$, then the minimal trellises of $\mathcal{C}$ and $\mathcal{C}'$ may have very different sizes. However, the simple action of coordinate permutation does not affect the performance of the code from an error-correction viewpoint. Therefore, given a code $\mathcal{C}$, one may as well use the code $\mathcal{C}'$ obtained by permuting the coordinates of $\mathcal{C}$, such that the minimal trellis of $\mathcal{C}'$ has the least complexity among the minimal trellises of codes obtained from $\mathcal{C}$ via coordinate permutations. The problem of determining the coordinate permutation of $\mathcal{C}$ that minimizes the complexity of the resulting minimal trellis has been termed the "art of trellis decoding" by Massey [8].

* This work was supported in part by a research grant from the Natural Sciences and Engineering Research Council (NSERC) of Canada.

S. Boztacs and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 198–207, 2007.
© Springer-Verlag Berlin Heidelberg 2007

It now matters which measure of trellis complexity is to be minimized, as the coordinate permutation of $\mathcal{C}$ that yields a minimal trellis with, say, the least state-complexity need not be the same as the coordinate permutation that yields a minimal trellis with the smallest number of states. The prior literature has most often focused on the problem of finding the coordinate permutation of a given code that minimizes the state-complexity of the resulting minimal trellis, and it has repeatedly been conjectured that this problem is NP-hard [5], [6], [11, Section 5]. To put it another way, the following decision problem was conjectured to be NP-complete:

Problem: TRELLIS STATE-COMPLEXITY

Let $\mathbb{F}_q$ be a fixed finite field.

Instance: An $m \times n$ generator matrix for a linear code $\mathcal{C}$ over $\mathbb{F}_q$, and an integer $w > 0$.

Question: Is there a coordinate permutation of $\mathcal{C}$ that yields a code $\mathcal{C}'$ whose minimal trellis has state-complexity at most w?

This decision problem was called "Maximum Partition Rank Permutation" in [5], and "Maximum Width" in [6]. Forney [4] has referred to the resolution of the aforementioned conjecture as the "only significant open problem" in the context of trellis representations.

In this paper, we settle the conjecture in the affirmative. We show that, for any fixed finite field $\mathbb{F}_q$, given an arbitrary code $\mathcal{C}$ over $\mathbb{F}_q$, the problem of finding the coordinate permutation of $\mathcal{C}$ that yields a minimal trellis with the least possible state-complexity is indeed NP-hard. Thus, TRELLIS STATE-COMPLEXITY is NP-complete. Our proof is by reduction from the problem of computing the pathwidth of a graph, which is known to be NP-hard [1],[2].

The rest of the paper is organized as follows. In Section 2, we lay down the definitions and notation necessary for our development. In Section 3, we sketch out a proof of the fact that for any fixed finite field $\mathbb{F}_q$, TRELLIS STATE-COMPLEXITY is NP-complete. We have had to omit some of the details of the proof due to space limitations; the complete proof can be found in our full paper [7]. We make some concluding remarks in Section 4.

2 Preliminaries

A *trellis* T for a length-n linear code $\mathcal{C}$ over a finite field $\mathbb{F}_q$ is an edge-labelled directed acyclic graph with certain properties. The vertex set, V, of T can be partitioned into $n+1$ disjoint subsets $V_0, V_1, \ldots, V_n$, such that each (directed) edge of T starts at V_i and ends at V_{i+1} for some $i \in \{0, 1, \ldots, n-1\}$. The set V_i is called the set of *states* at time index i. The set V_0 consists of a unique initial state v_0, and the set V_n consists of a unique terminal state v_n. It is further required that each state $v \in V$ lie on some (directed) path from v_0 to v_n. Note that each path from v_0 to v_n is of length exactly n. The edges of T are given labels from $\mathbb{F}_q$ in such a way that the set of all label sequences associated with paths from v_0 to v_n is precisely the code $\mathcal{C}$.

It turns out that if T is the minimal trellis for a linear code $\mathcal{C}$, then the cardinalities of the sets V_i are all powers of q. It is thus convenient to define the *state-complexity profile* of T to be the $(n+1)$-tuple $\mathbf{s} = (s_0, s_1, \ldots, s_n)$, where $s_i = \log_q(|V_i|)$. The *state-complexity* of T is then defined as $s_{\max} = \max_i s_i$. When T is the minimal trellis of $\mathcal{C}$, there is an explicit expression known for the s_i's. We will find it convenient to give this expression in terms of the connectivity function of $\mathcal{C}$, as defined below.

The set $[n] \stackrel{\text{def}}{=} \{1, 2, \ldots, n\}$ is taken to be the coordinate set of the length-n code $\mathcal{C}$. Given a subset $J \subset [n]$, we let $\mathcal{C}|_J$ denote the restriction of $\mathcal{C}$ to the coordinates with labels in J. In other words, $\mathcal{C}|_J$ is the code obtained by puncturing the coordinates in $J^c = [n] - J$. The *connectivity function* of the code $\mathcal{C}$ is the function $\lambda_{\mathcal{C}} : 2^{[n]} \to \mathbb{Z}$ defined by

$$\lambda_{\mathcal{C}}(J) = \dim(\mathcal{C}|_J) + \dim(\mathcal{C}|_{J^c}) - \dim(\mathcal{C}), \tag{1}$$

for each $J \subset [n]$. It is obvious that for any $J \subset [n]$, we have $\lambda_{\mathcal{C}}(J) \geq 0$ and $\lambda_{\mathcal{C}}(J) = \lambda_{\mathcal{C}}(J^c)$. Observe also that $\lambda_{\mathcal{C}}(\emptyset) = \lambda_{\mathcal{C}}([n]) = 0$. Furthermore, some elementary linear algebra suffices to verify that $\lambda_{\mathcal{C}}(J) = \lambda_{\mathcal{C}^\perp}(J)$ for any $J \subset [n]$.

The state-complexity profile of the minimal trellis of $\mathcal{C}$ can now be expressed as $\mathbf{s}(\mathcal{C}) = (s_0(\mathcal{C}), s_1(\mathcal{C}), \ldots, s_n(\mathcal{C}))$, where $s_0(\mathcal{C}) = s_n(\mathcal{C}) = 0$, and for $1 \leq i \leq n-1$,

$$s_i(\mathcal{C}) = \lambda_{\mathcal{C}}(\{1, 2, \ldots, i\}). \tag{2}$$

Thus, the state-complexity of the minimal trellis of $\mathcal{C}$ is given by $s_{\max}(\mathcal{C}) = \max_{i \in [n]} s_i(\mathcal{C})$. Note that since $\lambda_{\mathcal{C}}(J) = \lambda_{\mathcal{C}^\perp}(J)$ for any $J \subset [n]$, we have $\mathbf{s}(\mathcal{C}) = \mathbf{s}(\mathcal{C}^\perp)$, and hence, $s_{\max}(\mathcal{C}) = s_{\max}(\mathcal{C}^\perp)$.

As mentioned in Section 1, different coordinate permutations of the same code may result in codes with minimal trellises of very different complexities [11, Example 5.1]. Therefore, letting $[\mathcal{C}]$ denote the set of all codes that can be obtained from a code $\mathcal{C}$ by means of coordinate permutations, it is of interest to define the *trellis-width* of the family $[\mathcal{C}]$ as follows:

$$\text{tw}[\mathcal{C}] = \min_{\mathcal{C}' \in [\mathcal{C}]} s_{\max}(\mathcal{C}') = \min_{\mathcal{C}' \in [\mathcal{C}]} \max_{i \in [n]} s_i(\mathcal{C}'). \tag{3}$$

The main aim of this paper is to show that, given a code $\mathcal{C}$, the problem of computing the trellis-width of $[\mathcal{C}]$ is NP-hard. We accomplish this by reduction from the known NP-hard problem of computing the pathwidth of a graph.

3 NP-Hardness of Trellis-Width

The notion of graph pathwidth was introduced by Robertson and Seymour in [10]. Let $\mathcal{G}$ be a graph with vertex set V. An ordered collection $\mathcal{V} = (V_1, \ldots, V_t)$, $t \geq 1$, of subsets of V is called a *path-decomposition* of $\mathcal{G}$, if

(i) $\bigcup_{i=1}^{t} V_i = V$;
(ii) for each pair of adjacent vertices $u, v \in V$, we have $\{u, v\} \subset V_i$ for some $i \in [t]$; and

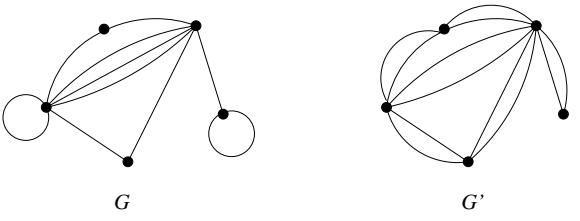


Fig. 1. Construction of $\mathcal{G}'$ from $\mathcal{G}$

(iii) for $1 \leq i < j < k \leq t$, $V_i \cap V_k \subset V_j$.

The *width* of such a path-decomposition $\mathcal{V}$ is defined to be $w_{\mathcal{G}}(\mathcal{V}) = \max_{i \in [t]} |V_i| - 1$. The *pathwidth* of $\mathcal{G}$, denoted by $\mathrm{pw}(\mathcal{G})$, is the minimum among the widths of all its path-decompositions. A path-decomposition $\mathcal{V}$ such that $w_{\mathcal{G}}(\mathcal{V}) = \mathrm{pw}(\mathcal{G})$ is called an *optimal* path-decomposition of $\mathcal{G}$.

Let $\mathbb{F}_q$ be an arbitrary finite field. Given a graph $\mathcal{G}$ with vertex set V, our aim is to produce, in time polynomial in $|V|$, a matrix A that generates a code $\mathcal{C}$ over $\mathbb{F}_q$ such that $\mathrm{pw}(\mathcal{G})$ can be directly computed from $\mathrm{tw}[\mathcal{C}]$. The NP-hardness of computing graph pathwidth then implies the NP-hardness of computing the trellis-width of $[\mathcal{C}]$ for an arbitrary code $\mathcal{C}$ over F_q. We now describe our construction of the matrix A.

Let $\mathcal{G}'$ be a graph defined on the same vertex set, V, as $\mathcal{G}$, having the following properties (see Figure 1):

(P1) $\mathcal{G}'$ is loopless;
(P2) a pair of distinct vertices is adjacent in $\mathcal{G}'$ iff it is adjacent in $\mathcal{G}$; and
(P3) in $\mathcal{G}'$, there are exactly two edges between each pair of adjacent vertices.

It is evident from the definition that $(V_1, \ldots, V_t)$ is a path-decomposition of $\mathcal{G}$ iff it is a path-decomposition of $\mathcal{G}'$. Therefore, $\mathrm{pw}(\mathcal{G}') = \mathrm{pw}(\mathcal{G})$.

Define $\overline{\mathcal{G}}$ to be the graph obtained by adding an extra vertex, henceforth denoted by x, to $\mathcal{G}'$, along with a pair of parallel edges from x to each $v \in V$ (see Figure 2). We will denote by $\overline{V}$ and $\overline{E}$ the vertex and edge sets, respectively, of $\overline{\mathcal{G}}$. Clearly, $\overline{\mathcal{G}}$ is constructible directly from $\mathcal{G}$ in $O(|V|^2)$ time. But more importantly, the desired matrix A can be readily obtained from the graph $\overline{\mathcal{G}}$. Indeed, letting $D(\overline{\mathcal{G}})$ be any directed graph obtained by arbitrarily assigning orientations to the edges of $\overline{\mathcal{G}}$, we simply take A to be the vertex-edge incidence matrix of $D(\overline{\mathcal{G}})$. This is the $|\overline{V}| \times |\overline{E}|$ matrix whose rows and columns are indexed by the vertices and directed edges, respectively, of $D(\overline{\mathcal{G}})$, and whose (i, j)th entry, $a_{i,j}$, is determined as follows:

$$a_{i,j} = \begin{cases} 1 & \text{if vertex } i \text{ is the tail of non-loop edge } j \\ -1 & \text{if vertex } i \text{ is the head of non-loop edge } j \\ 0 & \text{otherwise.} \end{cases}$$

Denote by $\overline{\mathcal{C}}$ the linear code over $\mathbb{F}_q$ generated by the matrix A. The trellis-width of $[\overline{\mathcal{C}}]$ relates very simply to the pathwidth of the original graph $\mathcal{G}$, as made precise by the following proposition.

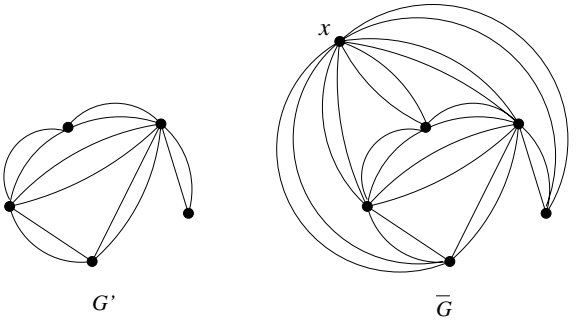


Fig. 2. Construction of $\overline{\mathcal{G}}$ from $\mathcal{G}'$

Proposition 1. $tw[\overline{\mathcal{C}}] = pw(\mathcal{G}) + 1$.

Before proving the above proposition, we observe that it yields the desired NP-hardness result. Indeed, it is easily checked that the matrix A can be constructed directly from $\mathcal{G}$ in $O(|V|^3)$ time. Now, suppose that there were a polynomial-time algorithm for computing the trellis-width of $[\mathcal{C}]$ for an arbitrary code $\mathcal{C}$ over $\mathbb{F}_q$, the code $\mathcal{C}$ being specified by some generator matrix. Then, given any graph $\mathcal{G}$, we can construct the matrix A, and then compute the trellis-width of $[\overline{\mathcal{C}}]$, all in polynomial time. Therefore, by Proposition 1, we have a polynomial-time algorithm to compute the pathwidth of $\mathcal{G}$. However, the graph pathwidth problem is NP-hard [1],[2]. So, if there exists a polynomial-time algorithm for it, then we must have $P = NP$. This implies our main result.

Theorem 1. *Let $\mathbb{F}_q$ be a fixed finite field. The problem of computing the trellis-width of an arbitrary linear code over $\mathbb{F}_q$, specified by any of its generator matrices, is NP-hard.*

Corollary 1. *For any fixed finite field $\mathbb{F}_q$, the decision problem* Trellis State-Complexity *is NP-complete.*

The remainder of this section is devoted to the proof of Proposition 1. Since $\mathrm{pw}(\mathcal{G}') = \mathrm{pw}(\mathcal{G})$, for the purpose of our proof, we may assume that $\mathcal{G}' = \mathcal{G}$. Thus, from now until the end of this section, we take $\mathcal{G}$ to be a loopless graph satisfying property (P3) above. Note that $\overline{\mathcal{G}}$ also satisfies (P3). For each pair of adjacent vertices u, v in $\mathcal{G}$ or $\overline{\mathcal{G}}$, we denote by l_{uv} and r_{uv} the two edges between u and v. Recall that V and E denote the sets of vertices and edges, respectively, of $\mathcal{G}$, and that $\overline{V}$ and $\overline{E}$ denote the corresponding sets of $\overline{\mathcal{G}}$. We thus have $\overline{V} = V \dot{\cup} \{x\}$, and $\overline{E} = E \dot{\cup} \left(\bigcup_{v \in V} \{l_{xv}, r_{xv}\}\right)$.

We will make much use of a basic fact, stated next, about the $|\overline{V}| \times |\overline{E}|$ matrix A whose construction was described above. For any $J \subset \overline{E}$, if $A|_J$ denotes the matrix obtained by restricting A to the columns indexed by the edges in J, then

$$\mathrm{rank}(A|_J) = \dim(\overline{\mathcal{C}}|_J) = r(J), \tag{4}$$

where rank and dim above are computed over the field $\mathbb{F}_q$, and $r(J)$ denotes the number of edges in any spanning forest of the subgraph of $\overline{\mathcal{G}}$ induced by J. To be precise, letting $\overline{\mathcal{G}}[J]$ denote the subgraph of $\overline{\mathcal{G}}$ induced by J, we have $r(J) = |V(\overline{\mathcal{G}}[J])| - \omega(\overline{\mathcal{G}}[J])$, where $\omega(\overline{\mathcal{G}}[J])$ is the number of connected components of $\overline{\mathcal{G}}[J]$. Equation (4) can be inferred from [9, Proposition 5.1.2].

We shall identify the set $\overline{E}$ with the coordinate set of the code $\overline{\mathcal{C}}$ generated by A. Given an ordering $\pi = (e_1, e_2, \ldots, e_n)$ of the elements of $\overline{E}$, we will denote by $\overline{\mathcal{C}}_\pi$ the code obtained by putting the coordinates of $\overline{\mathcal{C}}$ in the order specified by π. For any $J \subset \overline{E}$, and any ordering, π, of $\overline{E}$, we have by virtue of (4),

$$\begin{aligned}\lambda_{\overline{\mathcal{C}}_\pi}(J) = \lambda_{\overline{\mathcal{C}}}(J) &= r(J) + r(\overline{E} - J) - r(\overline{E}) \\ &= r(J) + r(\overline{E} - J) - |V|, \end{aligned} \tag{5}$$

the last equality above following from the fact that $\omega(\overline{\mathcal{G}}) = 1$ since $\overline{\mathcal{G}}$ is connected (each $v \in V$ is adjacent to x), so that $r(\overline{E}) = |\overline{V}| - 1 = |V|$.

We are now in a position to begin the proof of Proposition 1. We will first prove that $\mathrm{tw}[\overline{\mathcal{C}}] \leq \mathrm{pw}(\mathcal{G}) + 1$. Let $\mathcal{V} = (V_1, \ldots, V_t)$ be a path-decomposition of $\mathcal{G}$. We need the following fact about $\mathcal{V}$: for each $j \in [t]$,

$$\bigcup_{i \leq j} V_i \;\cap\; \bigcup_{k \geq j} V_k \;=\; V_j. \tag{6}$$

The above equality follows from the fact that a path-decomposition, by definition, has the property that for $1 \leq i < j < k \leq t$, $V_i \cap V_k \subset V_j$.

For $j \in [t]$, let F_j be the set of edges of $\mathcal{G}$ that have both their end-points in V_j. By condition (ii) in the definition of path-decomposition, $\bigcup_{j=1}^{t} F_j = E$. Now, let $\overline{F_j} = F_j \cup \left(\bigcup_{v \in V_j} \{l_{xv}, r_{xv}\}\right)$, so that $\bigcup_{j=1}^{t} \overline{F_j} = \overline{E}$.

Definition 1. *An ordering $(e_1, \ldots, e_n)$ of the edges of $\overline{\mathcal{G}}$ is said to* induce *an ordered partition $(E_1, \ldots, E_t)$ of $\overline{E}$ if for each $j \in [t]$, $\{e_{n_{j-1}+1}, e_{n_{j-1}+2}, \ldots, e_{n_j}\} = E_j$, where $n_j = \left|\bigcup_{i \leq j} E_i\right|$ (and $n_0 = 0$).*

Let $\pi = (e_1, \ldots, e_n)$ be any ordering of $\overline{E}$ that induces the ordered partition $(E_1, E_2, \ldots, E_t)$, where for each $j \in [t]$, $E_j = \overline{F_j} - \bigcup_{i<j} \overline{F_i}$. We claim that the state-complexity of the minimal trellis of $\overline{\mathcal{C}}_\pi$ is at most one more than the width of the path-decomposition $\mathcal{V}$.

Lemma 1. $s_{\max}(\overline{\mathcal{C}}_\pi) \leq w_{\mathcal{G}}(\mathcal{V}) + 1$.

Proof. Observe first that

$$\begin{aligned} s_{\max}(\overline{\mathcal{C}}_\pi) &= \max_{j \in [t]} \max_{1 \leq k \leq n_j - n_{j-1}} \lambda_{\overline{\mathcal{C}}_\pi}\left(\bigcup_{i<j} E_i \cup \{e_{n_{j-1}+1}, \ldots, e_{n_{j-1}+k}\}\right) \\ &\leq \max_{j \in [t]} \max_{E' \subset E_j} \lambda_{\overline{\mathcal{C}}_\pi}\left(\bigcup_{i<j} E_i \cup E'\right). \end{aligned} \tag{7}$$

Let $X = \bigcup_{i<j} E_i \cup E'$ for some $j \in [t]$ and $E' \subset E_j$. By (5), $\lambda_{\overline{\mathcal{C}}_\pi}(X) = r(X) + r(\overline{E} - X) - |V|$. If v is a vertex of $\overline{\mathcal{G}}$ incident with an edge in X, then $v \in \bigcup_{i\leq j} V_i \cup \{x\}$. So, the subgraph of $\overline{\mathcal{G}}$ induced by X has its vertices contained in $\bigcup_{i\leq j} V_i \cup \{x\}$. Therefore, $r(X) \leq \left|\bigcup_{i\leq j} V_i \cup \{x\}\right| - 1 = \left|\bigcup_{i\leq j} V_i\right|$.

Next, consider $\overline{E} - X = (\bigcup_{k>j} E_k) \cup (E_j - E')$. Reasoning as above, the subgraph of $\overline{\mathcal{G}}$ induced by $\overline{E} - X$ has its vertices contained in $\bigcup_{k\geq j} V_k \cup \{x\}$. Hence, $r(\overline{E} - X) \leq \left|\bigcup_{k\geq j} V_k\right|$.

Therefore, we have

$$\lambda_{\overline{\mathcal{C}}_\pi}(X) \;\leq\; \left|\bigcup_{i\leq j} V_i\right| + \left|\bigcup_{k\geq j} V_k\right| - |V| \;=\; \left|\bigcup_{i\leq j} V_i \cap \bigcup_{k\geq j} V_k\right| \;=\; |V_j|,$$

the last equality arising from (6). Hence, carrying on from (7),

$$s_{\max}(\overline{\mathcal{C}}_\pi) \leq \max_{j\in[t]} |V_j| = w_{\mathcal{G}}(\mathcal{V}) + 1,$$

as desired. □

The fact that $\mathrm{tw}[\overline{\mathcal{C}}] \leq \mathrm{pw}(\mathcal{G}) + 1$ easily follows from the above lemma. Indeed, we may choose $\mathcal{V}$ to be an optimal path-decomposition of $\mathcal{G}$. Then, by Lemma 1, there exists an ordering π of $\overline{E}$ such that $s_{\max}(\overline{\mathcal{C}}_\pi) \leq \mathrm{pw}(\mathcal{G}) + 1$. Hence, $\mathrm{pw}(M) \leq s_{\max}(\overline{\mathcal{C}}_\pi) \leq \mathrm{pw}(\mathcal{G}) + 1$.

We prove the reverse inequality in two steps, first showing that $\mathrm{pw}(\overline{\mathcal{G}}) = \mathrm{pw}(\mathcal{G}) + 1$, and then showing that $\mathrm{tw}[\overline{\mathcal{C}}] \geq \mathrm{pw}(\overline{\mathcal{G}})$.

Lemma 2. *$pw(\overline{\mathcal{G}}) = pw(\mathcal{G}) + 1$.*

Proof. Clearly, if $\mathcal{V} = (V_1, \ldots, V_t)$ is a path-decomposition of $\mathcal{G}$, then $\overline{\mathcal{V}} = (V_1 \cup \{x\}, \ldots, V_t \cup \{x\})$ is a path-decomposition of $\overline{\mathcal{G}}$. Hence, choosing $\mathcal{V}$ to be an optimal path-decomposition of $\mathcal{G}$, we have that $\mathrm{pw}(\overline{\mathcal{G}}) \leq w_{\overline{\mathcal{G}}}(\overline{\mathcal{V}}) = w_{\mathcal{G}}(\mathcal{V}) + 1 = \mathrm{pw}(\mathcal{G}) + 1$.

For the inequality in the other direction, we will show that there exists an optimal path-decomposition, $\widetilde{\mathcal{V}} = (\widetilde{V}_1, \ldots, \widetilde{V}_s)$, of $\overline{\mathcal{G}}$ such that $x \in \widetilde{V}_i$ for all $i \in [s]$. We then have $\mathcal{V} = (\widetilde{V}_1 - \{x\}, \ldots, \widetilde{V}_s - \{x\})$ being a path-decomposition of $\mathcal{G}$, and hence, $\mathrm{pw}(\mathcal{G}) \leq w_{\mathcal{G}}(\mathcal{V}) = w_{\overline{\mathcal{G}}}(\widetilde{\mathcal{V}}) - 1 = \mathrm{pw}(\overline{\mathcal{G}}) - 1$.

Let $\overline{\mathcal{V}} = (\overline{V}_1, \ldots, \overline{V}_t)$ be an optimal path-decomposition of $\overline{\mathcal{G}}$, and let $i_0 = \min\{i : x \in \overline{V}_i\}$ and $i_1 = \max\{i : x \in \overline{V}_i\}$. Since $\overline{V}_i \cap \overline{V}_k \subset \overline{V}_j$ for $i < j < k$, we must have $x \in \overline{V}_i$ for each $i \in [i_0, i_1]$.

We claim that $(\overline{V}_{i_0}, \overline{V}_{i_0+1}, \ldots, \overline{V}_{i_1})$ is a path-decomposition of $\overline{\mathcal{G}}$. We only have to show that $\bigcup_{i=i_0}^{i_1} \overline{V}_i = \overline{V}$, and that for each pair of adjacent vertices $u, v \in \overline{V}$, $\{u, v\} \subset \overline{V}_i$ for some $i \in [i_0, i_1]$. To see why the first assertion is true, consider any $v \in \overline{V}$, $v \neq x$. Since x is adjacent to v, and $\overline{\mathcal{V}}$ is a path-decomposition of $\overline{\mathcal{G}}$,

$\{x, v\} \subset \overline{V}_i$ for some $i \in [t]$. However, $x \in \overline{V}_i$ iff $i \in [i_0, i_1]$, and so, $\{x, v\} \subset \overline{V}_i$ for some $i \in [i_0, i_1]$. In particular, $v \in \overline{V}_i$ for some $i \in [i_0, i_1]$.

For the second assertion, suppose that u, v is a pair of vertices adjacent in $\overline{\mathcal{G}}$. Obviously, $\{u, v\} \subset \overline{V}_j$ for some $j \in [t]$. Suppose that $j \notin [i_0, i_1]$. We consider the case when $j > i_1$; the case when $j < i_0$ is similar. As $\bigcup_{i=i_0}^{i_1} \overline{V}_i = \overline{V}$, there exist $i_2, i_3 \in [i_0, i_1]$ such that $u \in \overline{V}_{i_2}$ and $v \in \overline{V}_{i_3}$. Without loss of generality (WLOG), $i_2 \leq i_3$. If $i_2 = i_3$, then there exists $i \in [i_0, i_1]$ such that $\{u, v\} \subset \overline{V}_i$. If $i_2 < i_3$, we have $u \in \overline{V}_{i_2} \cap \overline{V}_j$ and $i_2 < i_3 < j$. Hence, $u \in \overline{V}_{i_3}$ as well, and so once again, we have an $i \in [i_0, i_1]$ such that $\{u, v\} \in \overline{V}_i$.

Thus, $(\overline{V}_{i_0}, \overline{V}_{i_0+1}, \ldots, \overline{V}_{i_1})$ is a path-decomposition of $\overline{\mathcal{G}}$, with the property that $x \in \overline{V}_i$ for all $i \in [i_0, i_1]$. It must be an optimal path-decomposition, since it is a subsequence of the optimal path-decomposition $\overline{\mathcal{V}}$. □

To complete the proof of Proposition 1, it remains to show that $\mathrm{tw}[\overline{\mathcal{C}}] \geq \mathrm{pw}(\overline{\mathcal{G}})$. We introduce some notation at this point. Recall that the two edges between a pair of adjacent vertices u and v in $\overline{\mathcal{G}}$ (or $\mathcal{G}$) are denoted by l_{uv} and r_{uv}. We define

$$L_{\mathcal{G}} = \{l_{uv} : u, v \text{ are adjacent vertices in } \mathcal{G}\},$$
$$R_{\mathcal{G}} = \{r_{uv} : u, v \text{ are adjacent vertices in } \mathcal{G}\},$$

$L_x = \bigcup_{v \in V}\{l_{xv}\}$ and $R_x = \bigcup_{v \in V}\{r_{xv}\}$, where x is the distinguished vertex in $\overline{V} - V$. For $L \subset L_x$, define the *closure* of L to be the set $\mathrm{cl}(L) = L \cup \{r_{xu} : l_{xu} \in L\} \cup \{l_{uv}, r_{uv} : l_{xu}, l_{xv} \in L\}$. Note that $\mathrm{cl}(L_x) = \overline{E}$.

Our argument rests on the next lemma, whose somewhat technical proof we omit here. We refer the reader instead to the proof given in [7].

Lemma 3. *There exists an ordering $\pi = (e_1, \ldots, e_n)$ of $\overline{E}$ with the following properties:*

(a) $s_{\max}(\overline{\mathcal{C}}_\pi) = tw[\overline{\mathcal{C}}]$.

(b) The ordering π induces an ordered partition of $\overline{E}$ of the form

$$(L_1, A_1, B_1, R_1, L_2, A_2, B_2, R_2, \ldots, L_t, A_t, B_t, R_t),$$

where for each $j \in [t]$, $L_j \subset L_x$, $A_j \subset L_{\mathcal{G}}$, $B_j \subset R_{\mathcal{G}}$ and $R_j \subset R_x$. Moreover, for each $u, v \in \overline{V}$, $l_{uv} \in L_j \cup A_j$ iff $r_{uv} \in B_j \cup R_j$.

(c) For the ordered partition in (b), we have for each $j \in [t]$,

$$A_j \cup B_j \subset cl(\bigcup_{i \leq j} L_i) - cl(\bigcup_{i < j} L_i).$$

We can now furnish the last remaining piece of the proof of Proposition 1.

Lemma 4. *$tw[\overline{\mathcal{C}}] \geq pw(\overline{\mathcal{G}})$.*

Proof. Let $\pi = (e_1, \ldots, e_n)$ be an ordering of $\overline{E}$ having the properties guaranteed by Lemma 3. This ordering induces an ordered partition $(L_1, A_1, B_1, R_1, \ldots, L_t, A_t, B_t, R_t)$ of $\overline{E}$, as in Lemma 3(b). For $j \in [t]$, define $Y_j = \bigcup_{i<j}(L_i \cup A_i \cup B_i \cup R_i) \cup (L_j \cup A_j)$, and $Y'_j = \overline{E} - Y_j = \bigcup_{i>j}(L_i \cup A_i \cup B_i \cup R_i) \cup (B_j \cup R_j)$. Letting $\overline{\mathcal{G}}[Y_j]$ and $\overline{\mathcal{G}}[Y'_j]$ denote the subgraphs of $\overline{\mathcal{G}}$ induced by Y_j and Y'_j, respectively, set $V_j = V(\overline{\mathcal{G}}[Y_j]) \cap V(\overline{\mathcal{G}}[Y'_j])$. In other words, V_j is the set of vertices common to both $\overline{\mathcal{G}}[Y_j]$ and $\overline{\mathcal{G}}[Y'_j]$. It is easily checked that $\mathcal{V} = (V_1, \ldots, V_t)$ is a path-decomposition of $\overline{\mathcal{G}}$. Note that

$$|V_j| = |V(\overline{\mathcal{G}}[Y_j])| + |V(\overline{\mathcal{G}}[Y'_j])| - |\overline{V}|.$$

We next observe that $\overline{\mathcal{G}}[Y_j]$ and $\overline{\mathcal{G}}[Y'_j]$ are connected graphs. From Lemma 3(c), we have that $Y_j \subset \mathrm{cl}(\bigcup_{i\leq j} L_i)$. Therefore, for any edge l_{uv} (or r_{uv}) in $Y_j - \bigcup_{i\leq j} L_i$, both l_{xu} and l_{xv} must be in some L_i, $i \leq j$. Thus, in $\overline{\mathcal{G}}[Y_j]$, each vertex $v \neq x$ is adjacent to x, which shows that $\overline{\mathcal{G}}[Y_j]$ is connected.

Consider any vertex $v \neq x$ in $\overline{\mathcal{G}}[Y'_j]$, such that $r_{xv} \notin Y'_j$. Then, $r_{uv} \in Y'_j$ for some $u \neq x$. So, $r_{uv} \in B_k$ for some $k \geq j$. By Lemma 3(c), $r_{uv} \in \mathrm{cl}(\bigcup_{i\leq k} L_i) - \mathrm{cl}(\bigcup_{i<k} L_i)$. This implies that either $l_{xu} \in L_k$ or $l_{xv} \in L_k$. Hence, either $r_{xu} \in R_k$ or $r_{xv} \in R_k$. However, r_{xv} cannot be in R_k, since $r_{xv} \notin Y'_j$, and so, $r_{xu} \in R_k$. Thus, (r_{xu}, r_{uv}) forms a path in $\overline{\mathcal{G}}[Y'_j]$ from x to v. It follows that $\overline{\mathcal{G}}[Y'_j]$ is connected.

Therefore, by (5),

$$\begin{aligned}\lambda_{\overline{\mathcal{C}}_\pi}(Y_j) &= r(Y_j) + r(Y'_j) - |V| \\ &= (|V(\overline{\mathcal{G}}[Y_j])| - 1) + (|V(\overline{\mathcal{G}}[Y'_j])| - 1) - (|\overline{V}| - 1) = |V_j| - 1.\end{aligned}$$

Hence, from Lemma 3(a),

$$\mathrm{tw}[\overline{\mathcal{C}}] = s_{\max}(\overline{\mathcal{C}}_\pi) \geq \max_{j\in[t]} \lambda_{\overline{\mathcal{C}}_\pi}(Y_j) = \max_{j\in[t]} |V_j| - 1 = w_{\overline{\mathcal{G}}}(\mathcal{V}) \geq \mathrm{pw}(\overline{\mathcal{G}}),$$

which proves the lemma. □

The proof of Proposition 1 is now complete.

4 Concluding Remarks

The main contribution of this paper was to show that the decision problem Trellis State-Complexity is NP-complete, thus settling a long-standing conjecture. Now, the situation is rather different if we consider a variation of the problem in which the integer w is *not* taken to be a part of the input to the problem. In other words, consider the following problem:

Problem: Weak Trellis State-Complexity
Let $\mathbb{F}_q$ be a fixed finite field, and let w be a fixed positive integer.

Instance: An $m \times n$ generator matrix for a linear code $\mathcal{C}$ over $\mathbb{F}_q$.

Question: Is there a coordinate permutation of $\mathcal{C}$ that yields a code $\mathcal{C}'$ whose minimal trellis has state-complexity at most w?

There is good reason to believe that this problem is solvable in polynomial time. We again refer the reader to our full paper [7] for evidence in support of this belief.

Acknowledgment

The author would like to thank Jim Geelen for contributing some of his ideas to this paper.

References

1. Arnborg, S., Corneil, D.G., Proskurowski, A.: Complexity of Finding Embeddings in a K-Tree. SIAM J. Alg. Disc. Math 8, 277–284 (1987)
2. Bodlaender, H.L.: A Tourist Guide Through Treewidth. Acta Cybernetica 11, 1–23 (1993)
3. Forney Jr., G.D.: Dimension/Length Profiles and Trellis Complexity of Linear Block Codes. IEEE Trans. Inform. Theory 40(6), 1741–1752 (1994)
4. Forney Jr., G.D.: Codes on Graphs: Constraint Complexity of Cycle-Free Realizations of Linear Codes. IEEE Trans. Inform. Theory 49(7), 1597–1610 (2003)
5. Horn, G.B., Kschischang, F.R: On The Intractability of Permuting a Block Code to Minimize Trellis Complexity. IEEE Trans. Inform. Theory 42(6), 2042–2048 (1996)
6. Jain, K., Măndoiu, I., Vazirani, V.V.: The "Art of Trellis Decoding" is Computationally Hard — for Large Fields. IEEE. Trans. Inform. Theory 44(3), 1211–1214 (1998)
7. Kashyap, N.: Matroid Pathwidth and Code Trellis Complexity. SIAM J. Discrete Math. ArXiv e-print 0705.1384 (to appear)
8. Massey, J.L.: Foundation and Methods of Channel Encoding. In: 1978 Int. Conf. Inform. Theory and Systems, vol. 65, NTG-Fachberichte, Berlin, Germany (1978)
9. Oxley, J.G.: Matroid Theory. Oxford University Press, Oxford, UK (1992)
10. Robertson, N., Seymour, P.D.: Graph Minors. I. Excluding a Forest. J. Combin. Theory, Ser. B 35, 39–61 (1983)
11. Vardy, A.: Trellis Structure of Codes. In: Brualdi, R., Huffman, C., Pless, V. (eds.) Handbook of Coding Theory, Elsevier, Amsterdam, The Netherlands (1998)

On the Structure of Inversive Pseudorandom Number Generators

Harald Niederreiter[1] and Arne Winterhof[2]

[1] Department of Mathematics
National University of Singapore
2 Science Drive 2, Singapore 117543
Republic of Singapore
nied@math.nus.edu.sg
[2] Johann Radon Institute for Computational and Applied Mathematics
Austrian Academy of Sciences
Altenbergerstr. 69
4040 Linz, Austria
arne.winterhof@oeaw.ac.at

Abstract. We analyze the lattice structure and linear complexity of a new inversive pseudorandom number generator recently introduced by Niederreiter and Rivat. In particular, we introduce a new lattice test which is much stronger than its predecessors and prove that this new generator passes it up to very high dimensions. Such a result cannot be obtained for the conventional inversive generator with currently known methods. We also analyze the behavior of two explicit inversive generators under this new test and present lower bounds on the linear complexity profile of binary sequences derived from these three inversive generators.

1 Introduction

Let (η_n), $n = 0, 1, \ldots$, be a T-periodic sequence over the finite field $\mathbb{F}_q$ of q elements. For given integers $s \geq 1$, $0 < d_1 < d_2 < \ldots < d_{s-1} < T$, and $N \geq 2$, we say that (η_n) passes the *s-dimensional N-lattice test with lags $d_1, \ldots, d_{s-1}$* if the vectors $\{\underline{\eta}_n - \underline{\eta}_0 : 1 \leq n \leq N-1\}$ span $\mathbb{F}_q^s$, where

$$\underline{\eta}_n = (\eta_n, \eta_{n+d_1}, \ldots, \eta_{n+d_{s-1}}), \quad 0 \leq n \leq N-1.$$

In the case $d_i = i$ for $1 \leq i \leq s-1$, this test coincides essentially with the lattice test introduced in [6] and further analyzed in [4,5,6,7,11,24]. The latter lattice test is closely related to the concept of the linear complexity profile, see [6,7,22]. If additionally q is a prime and $N \geq T$, this special lattice test was proposed by Marsaglia [13].

If (η_n) passes the s-dimensional N-lattice test for all possible choices of lags, then it passes all s'-dimensional N-lattice tests for all possible choices of lags for $s' \leq s$ as well. Conversely, if (η_n) fails the s-dimensional N-lattice test for

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 208–216, 2007.
© Springer-Verlag Berlin Heidelberg 2007

a particular choice of lags, then it fails all s'-dimensional N-lattice tests with the same first s lags for $s' \geq s$. The greatest s such that (η_n) satisfies the s-dimensional N-lattice test for all lags $d_1, \ldots, d_{s-1}$ is denoted by $S(\eta_n, N)$, i.e.,

$$S(\eta_n, N) = \max\{s : \forall 0 < d_1 < \ldots < d_{s-1} < T : \\ \left\langle \left(\eta_n - \eta_0, \eta_{n+d_1} - \eta_{d_1}, \ldots, \eta_{n+d_{s-1}} - \eta_{d_{s-1}}\right),\ 1 \leq n < N \right\rangle = \mathbb{F}_q^s\}.$$

For given $\alpha \in \mathbb{F}_q^*$ and $\beta \in \mathbb{F}_q$, let ψ be the permutation of $\mathbb{F}_q$ defined by

$$\psi(\gamma) = \begin{cases} \alpha\gamma^{-1} + \beta & \text{if } \gamma \neq 0, \\ \beta & \text{if } \gamma = 0. \end{cases} \tag{1}$$

We can construct a sequence (γ_n) of elements of $\mathbb{F}_q$ by the recurrence relation

$$\gamma_n = \psi(\gamma_{n-1}) \quad \text{for } n = 1, 2, \ldots, \tag{2}$$

where γ_0 is the initial value. It is obvious that the sequence (2) is purely periodic with least period $t \leq q$. Conditions on α and β that guarantee the largest possible value $t = q$ are known (see [3,18]).

The map ψ in (1) is the basis for the family of inversive generators of pseudorandom numbers which can be traced back to the paper [8]. This family includes the recursive inversive generator, the compound inversive generator, the digital inversive generator, and others. Pseudorandom numbers produced by inversive generators have attractive distribution and structural properties. We refer to [10,21,23] for surveys on inversive and related generators.

In [12] and [20] the study of the (non-)linear complexity profile and the distribution properties of the sequence (2), respectively, used the sequence $R_0, R_1, \ldots$ of rational functions over $\mathbb{F}_q$ defined by

$$R_0(X) = X,\ R_n(X) = R_{n-1}(\alpha X^{-1} + \beta) \quad \text{for } n = 1, 2, \ldots,$$

where again $\alpha \in \mathbb{F}_q^*$ and $\beta \in \mathbb{F}_q$. It was proved in [20, Lemma 2] that there exist distinct elements $\varepsilon_1, \ldots, \varepsilon_{t-1}$ of $\mathbb{F}_q$ such that

$$R_n(X) = \frac{(\beta - \varepsilon_n)X + \alpha}{X - \varepsilon_n} \qquad \text{for } 1 \leq n \leq t-1.$$

In [19, Lemma 1] this result was extended by proving that the sequence of rational functions $R_0, R_1, \ldots$ is purely periodic with least period $T \leq q+1$ (obviously $t \leq T$), and that there exist distinct elements $\varepsilon_1, \ldots, \varepsilon_{T-1}$ of $\mathbb{F}_q$ such that the same equality above holds.

For $0 \leq n \leq t-1$, using the fact that ψ is a permutation, by (1) we have $R_n(\gamma) = \psi^n(\gamma)$ for all but at most n elements $\gamma \in \mathbb{F}_q$. For the purposes of the papers [12,20], it was possible to keep the influence of this exceptional set under control, but this is not always the case. Therefore for $1 \leq n \leq T-1$, Niederreiter and Rivat [19] introduced the alternative permutations of $\mathbb{F}_q$ defined by

$$\psi_n(\gamma) = \begin{cases} R_n(\gamma) & \text{if } \gamma \neq \varepsilon_n, \\ \beta - \varepsilon_n & \text{if } \gamma = \varepsilon_n, \end{cases}$$

and built from a seed $\gamma_0 \in \mathbb{F}_q$ a sequence (γ_n) of elements of $\mathbb{F}_q$, purely periodic of period T, by putting

$$\gamma_n = \psi_n(\gamma_0) \qquad \text{for } 1 \le n \le T-1. \tag{3}$$

The least period of this sequence can be smaller than T.

The advantage of this construction is that for $m, n \ge 0$, we have

$$\psi_m(\psi_n(\gamma)) = R_{m+n}(\gamma) \quad \text{for } \gamma \ne \varepsilon_n \text{ and } \psi_n(\gamma) \ne \varepsilon_m. \tag{4}$$

The price is a slightly more complicated algorithm to compute γ_n (see [19, Section 2]). The new construction allowed the authors of [19] to prove much stronger distribution and correlation properties of the generated sequences, as compared to the case of the standard generator (2). It will also allow us to prove a strong lower bound on $S(\gamma_n, N)$.

Let us observe that both constructions (2) and (3) need $\beta \ne 0$ to be of interest. We will assume this condition in the sequel. Moreover, it is reasonable to assume that

$$\gamma_0^2 \ne \beta\gamma_0 + \alpha. \tag{5}$$

Indeed if $\gamma_0^2 = \beta\gamma_0 + \alpha$, then for $1 \le n \le T-1$ such that $\varepsilon_n \ne \gamma_0$ we have

$$\gamma_n = \psi_n(\gamma_0) = R_n(\gamma_0) = \frac{(\beta - \varepsilon_n)\gamma_0 + \alpha}{\gamma_0 - \varepsilon_n} = \frac{\gamma_0^2 - \varepsilon_n\gamma_0}{\gamma_0 - \varepsilon_n} = \gamma_0,$$

so that the sequence (γ_n) generated by (3) is almost constant and not of much interest. In [19, Lemma 2] it was shown that for $\beta \in \mathbb{F}_q^*$ and γ_0 satisfying (5), the sequence (γ_n) defined by (3) is purely periodic with least period T and contains at least $T-1$ distinct elements of $\mathbb{F}_q$. It has least period $T = q+1$ if the polynomial $X^2 - \beta X - \alpha$ is primitive over $\mathbb{F}_q$ by [19, Theorem 1].

Our main goal in this paper is to study the behavior of the sequence defined by (3) under the new lattice test. Essentially the same method also provides lower bounds on the linear complexity profile over $\mathbb{F}_q$ of these sequences.

We recall that the *linear complexity profile* $L(\eta_n, N)$ of an infinite sequence (η_n), $n = 0, 1, \ldots$, over the field $\mathbb{F}$ is the function which for every integer $N \ge 1$ is defined as the length L of a shortest linear recurrence relation

$$\eta_{n+L} = a_{L-1}\eta_{n+L-1} + \cdots + a_0\eta_n, \quad 0 \le n \le N-L-1,$$

with $a_0, \ldots, a_{L-1} \in \mathbb{F}$, which is satisfied by the first N terms of this sequence. If (η_n) starts with $N-1$ zeros, then we define $L(\eta_n, N) = 0$ if $\eta_{N-1} = 0$ and $L(\eta_n, N) = N$ if $\eta_{N-1} \ne 0$.

In contrast to the generator (2), we are able to analyze the behavior of the following two explicit inversive generators under the new lattice test.

For a prime $p \ge 3$, the *explicit inversive congruential generator of period p* was introduced in [9] as the sequence (z_n) defined by

$$z_n = (an + b)^{p-2}, \quad n = 0, \ldots, p-1, \quad z_{n+p} = z_n, \ n \ge 0, \tag{6}$$

with $a, b \in \mathbb{F}_p$, $a \ne 0$. Its linear complexity profile was analyzed in [16].

Explicit inversive generators can also be defined for other periods, see [17] where also a lower bound on the linear complexity profile of these sequences is given. Let $\alpha, \beta, \gamma \in \mathbb{F}_q^*$ and $t|(q-1)$ be the order of γ in the group $\mathbb{F}_q^*$. We call a sequence (η_n) defined by

$$\eta_n = \overline{\alpha\gamma^n + \beta}, \quad n \geq 0, \tag{7}$$

an *explicit inversive pseudorandom number generator of least period* t, where

$$\overline{\eta} = \begin{cases} \eta^{-1} & \text{if } \eta \in \mathbb{F}_q^*, \\ 0 & \text{if } \eta = 0. \end{cases}$$

In [19, Section 5],[15], and [2], corresponding pseudorandom binary sequences were considered and bounds on their correlation measures in the sense of Mauduit and Sárközy [14] were established. We use a general inequality of [1] to derive a lower bound on the linear complexity profile of these binary sequences from the bound on their correlation measure. These bounds cannot be obtained for the conventional inversive generator (2) with current methods.

2 Lattice Structure

In the algorithm for the generation of the sequence (3), we continue to assume the conditions $\beta \neq 0$ and (5).

Theorem 1. *For the sequence of elements* (γ_n) *defined by* (3)*, we have*

$$S(\gamma_n, N) \geq \frac{N}{3} - 1 \quad \textit{for } 2 \leq N \leq T.$$

Proof. We assume that the sequence (γ_n) does not pass the s-dimensional N-lattice test for some lags $0 < d_1 < d_2 < \ldots < d_{s-1} < T$. Put

$$\underline{\gamma}_n = (\gamma_n, \gamma_{n+d_1}, \gamma_{n+d_2}, \ldots, \gamma_{n+d_{s-1}}) \quad \text{for } n \geq 0$$

and let V be the subspace of $\mathbb{F}_q^s$ spanned by all $\underline{\gamma}_n - \underline{\gamma}_0$ for $0 \leq n \leq N-1$. Then $\dim(V) < s$ and $\dim(V^\perp) \geq 1$. Take $\underline{0} \neq \underline{\alpha} \in V^\perp$, then

$$\underline{\alpha} \cdot (\underline{\gamma}_n - \underline{\gamma}_0) = 0 \quad \text{for } 0 \leq n \leq N-1,$$

and thus

$$\underline{\alpha} \cdot \underline{\gamma}_n = \underline{\alpha} \cdot \underline{\gamma}_0 =: b \quad \text{for } 0 \leq n \leq N-1,$$

where $\cdot$ denotes the usual inner product. If $\underline{\alpha} = (\alpha_0, \alpha_1, \ldots, \alpha_{s-1})$, then let j be the smallest index with $\alpha_j \neq 0$ (so $0 \leq j < s$). Then with $d_0 := 0$ if $j = 0$,

$$\alpha_j \gamma_{n+d_j} + \alpha_{j+1}\gamma_{n+d_{j+1}} + \cdots + \alpha_{s-1}\gamma_{n+d_{s-1}} = b \quad \text{for } 0 \leq n \leq N-1. \tag{8}$$

Next we show that for at least $N-2s$ different elements γ_{n+d_j}, $0 \leq n \leq N-1$, we have

$$\gamma_{n+d_i} = R_{d_i - d_j}(\gamma_{n+d_j}) \quad \text{for } j+1 \leq i \leq s-1.$$

If we have $\gamma_{n+d_i} \neq R_{d_i-d_j}(\gamma_{n+d_j})$ for some $j+1 \leq i \leq s-1$, then by (3) and (4) we have either

$$\psi_{n+d_i}(\gamma_0) \neq R_{n+d_i}(\gamma_0),\ \psi_{n+d_j}(\gamma_0) \neq R_{n+d_j}(\gamma_0),$$

$$\text{or } \psi_{d_i-d_j}(\gamma_{n+d_j}) \neq R_{d_i-d_j}(\gamma_{n+d_j}),$$

or equivalently

$$\gamma_0 = \varepsilon_{n+d_i},\ \gamma_0 = \varepsilon_{n+d_j},\ \text{or } \gamma_{n+d_j} = \varepsilon_{d_i-d_j}.$$

(Otherwise we had $R_{d_i-d_j}(\gamma_{n+d_j}) = \psi_{d_i-d_j}(\gamma_{n+d_j}) = \psi_{d_i-d_j}(\psi_{n+d_j}(\gamma_0)) = R_{n+d_i}(\gamma_0) = \psi_{n+d_i}(\gamma_0) = \gamma_{n+d_i}$.) Here $\varepsilon_m = \varepsilon_r$ if r is the least residue of $m \in \mathbb{Z}$ modulo T, with ε_0 arbitrary but not in $\mathbb{F}_q$ (since $\psi_0(X) = X$ has no pole, but the notation is more convenient if we use a fixed $\varepsilon_0 \notin \mathbb{F}_q$). For fixed i the first equation can occur for at most one n, giving at most $s-1-j$ exceptions. The second equation can occur at most once. The third equation leads again to at most $s-1-j$ exceptions. Summarizing we get at most $2(s-1-j)+1 \leq 2s-1$ exceptions. By a remark in Section 1, at least $N-1$ of the N terms γ_{n+d_j}, $0 \leq n \leq N-1$, of the sequence (γ_n) are distinct. Hence in view of (8),

$$\alpha_j \gamma_{n+d_j} + \alpha_{j+1} R_{d_{j+1}-d_j}(\gamma_{n+d_j}) + \cdots + \alpha_{s-1} R_{d_{s-1}-d_j}(\gamma_{n+d_j}) = b$$

for at least $(N-1)-(2s-1) = N-2s$ different elements γ_{n+d_j}. Using the explicit representation of the rational functions R_n in Section 1, we see that the polynomial of degree $s-j \leq s$ given by

$$F(X) = (\alpha_j X - b) \prod_{i=j+1}^{s-1} (X - \varepsilon_{d_i-d_j}) + \sum_{k=j+1}^{s-1} \alpha_k ((\beta - \varepsilon_{d_k-d_j})X + \alpha) \prod_{\substack{i=j+1 \\ i \neq k}}^{s-1} (X - \varepsilon_{d_i-d_j})$$

has at least $N-2s$ zeros. This implies $s \geq N-2s$, whence the result. □

Corollary 1. *For the sequence of elements (γ_n) defined by (3), the linear complexity profile over $\mathbb{F}_q$ satisfies*

$$L(\gamma_n, N) \geq \frac{N}{4} - 1 \quad \textit{for } 1 \leq N \leq T.$$

Proof. We start with a linear recurrence relation of length L, say

$$\alpha_j \gamma_{n+j} + \cdots + \alpha_L \gamma_{n+L} = 0 \quad \text{for } 0 \leq n \leq N-L-1,$$

where $\alpha_j \neq 0$ and $\alpha_L = -1$. Then as in the previous proof we can construct a nonzero polynomial of degree at most $L+1$ with at least $N-3(L+1)$ zeros and get $L+1 \geq N-3(L+1)$, which implies the result. □

Theorem 2. *For the sequences of elements* (z_n) *defined by* (6) *and* (η_n) *defined by* (7)*, we have*

$$S(z_n, N) \ge \frac{N}{2} - 1 \quad \text{for } 2 \le N \le p$$

and

$$S(\eta_n, N) \ge \frac{N}{2} - 1 \quad \text{for } 2 \le N \le t.$$

Proof. As in the proof of Theorem 1 we assume that (z_n) does not pass the s-dimensional N-lattice test for some lags $d_1, \ldots, d_{s-1}$, and we get with the definition $d_0 := 0$ a recurrence relation

$$\alpha_j z_{n+d_j} + \cdots + \alpha_{s-1} z_{n+d_{s-1}} = \beta \quad \text{for } 0 \le n \le N-1,$$

with some $\alpha_j \ne 0$. If $a(n+d_i)+b \ne 0$ for $j \le i \le s-1$, then we have

$$z_{n+d_i} = (a(n+d_i)+b)^{-1} \quad \text{for } j \le i \le s-1,$$

and we see easily that the polynomial

$$F(X) := -\beta \prod_{i=j}^{s-1} (a(X+d_i)+b) + \sum_{k=j}^{s-1} \alpha_k \prod_{\substack{i=j \\ i \ne k}}^{s-1} (a(X+d_i)+b)$$

of degree at most s has at least $N-s$ zeros. If $\beta \ne 0$, then the degree of $F(X)$ is exactly $s-j > 0$, and otherwise we have $F(-a^{-1}b - d_j) \ne 0$. In both cases $F(X)$ is a nonzero polynomial, and the first result follows. The second result is shown in an analogous way. □

3 Linear Complexity of Binary Sequences

We choose for the prime power q an odd prime p and identify $\mathbb{F}_p$ with the set $\{0, 1, \ldots, p-1\}$ of integers. Given an arbitrary sequence (η_n) of elements of $\mathbb{F}_p$, we build a binary sequence $e_0, e_1, \ldots$ of elements of $\{0, 1\}$ by writing

$$e_n := \begin{cases} 0 & \text{if } 0 \le \eta_n \le (p-1)/2, \\ 1 & \text{if } (p+1)/2 \le \eta_n \le p-1. \end{cases} \tag{9}$$

The *correlation measure of order* $k \ge 1$ of a binary sequence (e_n) of period T is defined as

$$C_k(e_n, N) = \max_{M,D} \left| \sum_{n=0}^{M-1} (-1)^{e_{n+d_1} + e_{n+d_2} + \cdots + e_{n+d_k}} \right| \quad \text{for } 1 \le N \le T,$$

where the maximum is taken over all $D = (d_1, d_2, \ldots, d_k)$ with nonnegative integers $d_1 < d_2 < \ldots < d_k$ and integers $M \ge 1$ such that $d_k \le N - M$. This correlation measure was introduced in [14].

For $\mathbb{F} = \mathbb{F}_2$ we have the following relation between $L(e_n, N)$ and the correlation measure which is obtained from the proof of [1, Theorem 1].

Lemma 1. *Let (e_n) be a T-periodic binary sequence. For $1 \le N \le T$ we have*

$$L(e_n, N) \ge N - \max_{1 \le k \le L(e_n,N)+1} C_k(e_n, N).$$

Now we take a sequence (γ_n) of elements of $\mathbb{F}_p$ generated by (3). We assume again $\beta \in \mathbb{F}_p^*$ and (5), so that the sequence is purely periodic with least period T by [19, Lemma 2].

Corollary 2. *The binary sequence (e_n) defined by (9) and (3) satisfies*

$$L(e_n, N) = \Omega\left(\frac{\log(Np^{-1/2})}{\log\log p}\right) \quad \text{for } T \ge N > p^{1/2}.$$

Proof. By [19, Theorem 5] we have the upper bound

$$C_k(e_n, N) < 2^k \left((14k)^{1/2} N^{1/2} p^{1/4} + kp^{1/2} + 8k\right) \left(\frac{4}{\pi^2}\log p + 1.72\right)^k + \frac{k2^k N}{p}$$

and the result follows from Lemma 1 after simple calculations. □

Corollary 3. *The binary sequence (e_n) defined by (9) and (6) satisfies*

$$L(e_n, N) = \Omega\left(\frac{\log(Np^{-1/2})}{\log\log p}\right) \quad \text{for } p \ge N > p^{1/2}.$$

Proof. By [15, Theorem 2] and the remarks following it, we have

$$C_k(e_n, N) = O(kp^{1/2}(\log p)^{k+1})$$

which implies the result after simple calculations. □

Corollary 4. *The binary sequence (e_n) defined by (9) and (7) satisfies*

$$L(e_n, N) = \Omega\left(\frac{\log(Np^{-1/2})}{\log\log p}\right) \quad \text{for } t \ge N > p^{1/2}.$$

Proof. From the proofs of [2, Theorem 2] and [25, Theorem 4] we get

$$C_k(e_n, N) = O(k2^k p^{1/2} (\log p)^k \log t)$$

which implies the result after simple calculations. □

Acknowledgments

The research of the first author is partially supported by the project NUGET of the Agence Nationale de la Recherche (France). The second author was supported by the Austrian Science Fund (FWF) under the grant P-19004-N18. This work was done during a pleasant visit by A. W. to the National University of Singapore whose hospitality is gratefully acknowledged.

References

1. Brandstätter, N., Winterhof, A.: Linear Complexity Profile of Binary Sequences With Small Correlation Measure. Period. Math. Hungar 52, 1–8 (2006)
2. Chen, Z.X.: Finite Binary Sequences Constructed by Explicit Inversive Methods. Finite Fields Appl. (to appear)
3. Chou, W.S.: The Period Lengths of Inversive Pseudorandom Vector Generations. Finite Fields Appl. 1, 126–132 (1995)
4. Dorfer, G.: Lattice Profile and Linear Complexity Profile of Pseudorandom Number Sequences. In: Mullen, G.L., Poli, A., Stichtenoth, H. (eds.) Finite Fields and Applications. LNCS, vol. 2948, pp. 69–78. Springer, Heidelberg (2004)
5. Dorfer, G., Meidl, W., Winterhof, A.: Counting Functions and Expected Values for the Lattice Profile at n. Finite Fields Appl. 10, 636–652 (2004)
6. Dorfer, G., Winterhof, A.: Lattice Structure and Linear Complexity Profile of Nonlinear Pseudorandom Number Generators. Appl. Algebra Engrg. Comm. Comput. 13, 499–508 (2003)
7. Dorfer, G., Winterhof, A.: Lattice Structure of Nonlinear Pseudorandom Number Generators in Parts of the Period. In: Niederreiter, H. (ed.) Monte Carlo and Quasi-Monte Carlo Methods 2002, pp. 199–211. Springer, Berlin (2004)
8. Eichenauer, J., Lehn, J.: A Non-Linear Congruential Pseudo Random Number Generator. Statist. Papers 27, 315–326 (1986)
9. Eichenauer-Herrmann, J.: Statistical Independence of a New Class of Inversive Congruential Pseudorandom Numbers. Math. Comp. 60, 375–384 (1993)
10. Eichenauer-Herrmann, J., Herrmann, E., Wegenkittl, S.: A Survey of Quadratic and Inversive Congruential Pseudorandom Numbers. In: Niederreiter, H., et al. (eds.) Monte Carlo and Quasi-Monte Carlo Methods 1996. Lecture Notes in Statistics, vol. 127, pp. 66–97. Springer, Heidelberg (1998)
11. Fu, F.-W., Niederreiter, H.: On the Counting Function of the Lattice Profile of Periodic Sequences. J. Complexity (to appear)
12. Gutierrez, J., Shparlinski, I.E., Winterhof, A.: On the Linear and Nonlinear Complexity Profile of Nonlinear Pseudorandom Number Generators. IEEE Trans. Inf. Theory 49, 60–64 (2003)
13. Marsaglia, G.: The Structure of Linear Congruential Sequences. In: Zaremba, S.K. (ed.) Applications of Number Theory to Numerical Analysis, pp. 249–285. Academic Press, New York (1972)
14. Mauduit, C., Sárközy, A.: On Finite Pseudorandom Binary Sequences. I. Measure of Pseudorandomness. The Legendre Symbol. Acta Arith. 82, 365–377 (1997)
15. Mauduit, C., Sárközy, A.: Construction of Pseudorandom Binary Sequences by Using the Multiplicative Inverse. Acta Math. Hungar. 108, 239–252 (2005)
16. Meidl, W., Winterhof, A.: On the Linear Complexity Profile of Explicit Nonlinear Pseudorandom Numbers. Inf. Process. Lett. 85, 13–18 (2003)
17. Meidl, W., Winterhof, A.: On the Linear Complexity Profile of Some New Explicit Inversive Pseudorandom Numbers. J. Complexity 20, 350–355 (2004)
18. Niederreiter, H.: Pseudorandom Vector Generation by the Inversive Method. ACM Trans. Modeling and Computer Simulation 4, 191–212 (1994)
19. Niederreiter, H., Rivat, J.: On the Correlation of Pseudorandom Numbers Generated by Inversive Methods. Monatsh. Math. (to appear)
20. Niederreiter, H., Shparlinski, I.E.: On the Distribution of Pseudorandom Numbers and Vectors Generated by Inversive Methods. Appl. Algebra Engrg. Comm. Comput. 10, 189–202 (2000)

21. Niederreiter, H., Shparlinski, I.E.: Recent Advances in the Theory of Nonlinear Pseudorandom Number Generators. In: Fang, K.T., Hickernell, F.J., Niederreiter, H. (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2000, pp. 86–102. Springer, Berlin (2002)
22. Niederreiter, H., Winterhof, A.: Lattice Structure and Linear Complexity of Nonlinear Pseudorandom Numbers. Appl. Algebra Engrg. Comm. Comput. 13, 319–326 (2002)
23. Topuzoğlu, A., Winterhof, A.: Pseudorandom Sequences. In: Garcia, A., Stichtenoth, H. (eds.) Topics in Geometry, Coding Theory and Cryptography, pp. 135–166. Springer, Dordrecht (2007)
24. Wang, L.-P., Niederreiter, H.: Successive Minima Profile, Lattice Profile, and Joint Linear Complexity Profile of Pseudorandom Multisequences. J. Complexity (to appear)
25. Winterhof, A.: On the Distribution of Some New Explicit Inversive Pseudorandom Numbers and Vectors. In: Niederreiter, H., Talay, D. (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2004, pp. 487–499. Springer, Berlin (2006)

Subcodes of Reed-Solomon Codes Suitable for Soft Decoding

Safitha J. Raj and Andrew Thangaraj

Department of Electrical Engineering
Indian Institute of Technology Madras, Chennai, India
andrew@iitm.ac.in

Abstract. Reed-Solomon (RS) codes over GF(2^m) have traditionally been the most popular non-binary codes in almost all practical applications. The distance properties of RS codes result in excellent performance under hard-decision bounded-distance decoding. In this work, we consider certain subcodes of RS codes over GF(q^m) whose q-ary traces are BCH codes over GF(q). The properties of these subcodes are studied and low-complexity hard-decision and soft-decision decoders are proposed. The decoders are analyzed, and their performance is compared with that of comparable RS codes. Our results suggest that these subcodes of RS codes could have some advantages when compared to RS codes.

1 Introduction

Reed-Solomon (RS) codes [1] are the most prevalent and commonly used codes today with applications ranging from satellite communications to computer drives. RS codes are popular, in theory, for their elegant algebraic construction. In practice, RS codes can be encoded and decoded with manageable complexity and high speed. RS codes continue to remain objects of active research with most recent interest being in list and soft-decision decoding [2][3].

Efficient soft decoding of RS codes has traditionally been a problem of importance. Early methods for soft decoding of RS codes included Chase decoding and Generalized Minimum Distance (GMD) decoding [4]. Other methods for soft decoding RS codes include [5][6]. Recently, the Koetter-Vardy algorithm [3] and belief-propagation-based iterative algorithm [7] have been proposed. Common themes in the above methods include (1) a coding gain of around 1dB, (2) an increase in complexity with size of the field, and (3) an increase in complexity for higher coding gain. As a result, efficient soft decoders are not readily available for high rate RS codes over large fields.

In this work, we study certain subcodes of q^m-ary RS codes that are more amenable to efficient decoding. Specifically, we consider subcodes whose traces are q-ary BCH codes. Suitable non-consecutive zeros are added to the set of zeros of a parent RS code to enable the trace to be a BCH code. Though the subcode is not typically maximum-distance-separable (MDS), our analysis shows that a large fraction of errors beyond minimum distance are correctable. Hence,

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 217–226, 2007.
© Springer-Verlag Berlin Heidelberg 2007

the performance of these subcodes of RS codes is comparable to that of a MDS RS code at the same rate. We refer to these select subcodes of RS codes as sub Reed-Solomon (SRS) codes in the rest of this article.

Because of the trace structure, the SRS codes are amenable to efficient soft-decision decoding. Since the image of a q^m-ary code is a concatenation of its q-ary trace, a soft decoder for the trace can be efficiently used to process soft input for the image. Using this idea, we propose simple soft decoders for SRS codes. Our simulations show that the proposed soft decoders for high-rate (> 0.9) SRS codes over large fields (GF(256)) perform close to other comparable soft decoders of MDS RS codes at the same rate. However, the complexity of soft decoding SRS codes is significantly lesser. Our results suggest that SRS codes could be competent alternatives to RS codes in certain situations.

2 Preliminaries

A finite field GF(q^m) (q: power of prime) is an m-dimensional vector space over GF(q). A set of m elements of GF(q^m) linearly independent over GF(q) form a basis for this vector space. See [8] for more details on the definitions and preliminary results in this section.

2.1 Definitions

Trace of an element $\alpha \in \mathrm{GF}(q^m)$ is a linear mapping $T_m : \mathrm{GF}(q^m) \rightarrow \mathrm{GF}(q)$ defined by $T_m(\alpha) = \sum_{i=0}^{m-1} \alpha^{q^i}$. If $\mathcal{C}$ is a code over GF(q^m), trace of $\mathcal{C}$ consists of the traces of all codewords of $\mathcal{C}$. Let $\mathcal{B} = \{\beta_1, \beta_2, \ldots, \beta_m\}$ be a basis for GF(q^m) over GF(q). Each element $\alpha \in \mathrm{GF}(q^m)$ can be represented as a linear combination of the elements in the basis. Let $\mathcal{B}' = \{\beta'_1, \beta'_2, \ldots, \beta'_m\}$ be the dual basis of $\mathcal{B}$. Each element $\alpha \in \mathrm{GF}(q^m)$ can be expanded as $\alpha = \sum_{i=1}^{m} a_i \beta_i$, where $a_i = T_m\{\alpha\beta'_i\}$. The element $\alpha \in \mathrm{GF}(q^m)$ can be viewed as the vector $[a_1\ a_2\ \ldots\ a_m]$ over GF(q) through expansion by basis $\mathcal{B}$. The vector $[a_1\ a_2\ \ldots\ a_m]$ is also called the image of $\alpha \in \mathrm{GF}(q^m)$ over GF(q). If $\mathcal{C}$ is a code over GF(q^m), image of $\mathcal{C}$, denoted by $\mathcal{C}_i$, consists of the images (with respect to a chosen basis) of all codewords of $\mathcal{C}$. Image of an (n, k, d) linear code over GF(q^m) will be an $(nm, km, \geq d)$ linear code over GF(q).

2.2 Preliminary Results

Let $\mathcal{C}$ be a linear code of length n over GF(q^m) and $\mathcal{C}_i$ be the image of $\mathcal{C}$ over GF(q) through expansion by basis $\mathcal{B}$. The image of any codeword in $\mathcal{C}$ can be viewed as an $n \times m$ matrix over GF(q). If c_i is the i^{th} component of the codeword, i^{th} row of the image matrix will be $[T_m\{c_i\beta'_1\}\ T_m\{c_i\beta'_2\}\ \ldots\ T_m\{c_i\beta'_m\}]$.

Proposition 1. *Each column of an image matrix in $\mathcal{C}_i$ will belong to the trace of the code $\mathcal{C}$.*

Proof. Let $\mathbf{c} = [c_1\ c_2\ \ldots\ c_n]^T \in \mathcal{C}$. The j^{th} column of the image matrix will be $[T_m\{c_1\beta'_j\}\ T_m\{c_2\beta'_j\}\ \ldots\ T_m\{c_n\beta'_j\}]^T$.

$$\mathbf{c} \in \mathcal{C}\ \Rightarrow\ \beta'_j\mathbf{c} \in \mathcal{C}\ . \tag{1}$$

Hence $[T_m\{c_1\beta'_j\}\ T_m\{c_2\beta'_j\}\ \ldots\ T_m\{c_n\beta'_j\}]^T$ will belong to the trace of $\mathcal{C}$. □

Let the subfield subcode of $\mathcal{C}$ over GF(q) be denoted by $\mathcal{C}_{ss}$.

Proposition 2. *Minimum distance of $\mathcal{C}_i$ is less than or equal to the minimum distance of $\mathcal{C}_{ss}$.*

Proof. $\mathcal{C}_{ss}$ consists of the set of all codewords of $\mathcal{C}$ with elements over GF(q). Suppose $\mathbf{c} = [c_1\ c_2\ \ldots\ c_n]^T \in \mathcal{C}_{ss} \subseteq \mathcal{C}$ is a minimum weight codeword of $\mathcal{C}_{ss}$. Since $c_i \in$ GF(q), image of $c_i\beta_1$ is

$$[c_iT_m\{\beta_1\beta'_1\}\ c_iT_m\{\beta_1\beta'_2\}\ \ldots\ c_iT_m\{\beta_1\beta'_m\}]^T = [c_i0\ldots0]^T\ . \tag{2}$$

Hence, weight of the image of $\beta_1\mathbf{c} \in \mathcal{C}$ is equal to the weight of $\mathbf{c}$. Since minimum distance of $\mathcal{C}_i$ is upper bounded by the weight of an arbitrary codeword such as the image of $\beta_1\mathbf{c}$, the result follows. □

In summary, if d, d_{ss} and d_i are the minimum distances of $\mathcal{C}$, $\mathcal{C}_{ss}$ and $\mathcal{C}_i$, respectively, we have $d \leq d_i \leq d_{ss}$.

3 Sub Reed-Solomon Codes

In this section, we discuss the construction and basic properties of sub Reed-Solomon (SRS) codes with a nontrivial trace. We restrict ourselves to images of GF(2^m) over GF(2) for simplicity. All results extend to the general case.

3.1 Construction

Let α be a primitive element of GF(2^m). Let $\mathcal{C}(t)$ be the $(n, n-2t, 2t+1)$ primitive, narrow-sense t-error correcting RS code of length $n = 2^m - 1$. The code has $2t$ consecutive powers of α as zeros. The zero set is $Z_{rs} = \{1, 2, \ldots, 2t\}$. The generator polynomial of the code, $\mathcal{C}(t)$ is given by $\prod_{i=1}^{2t}(x + \alpha^i)$.

A SRS code $\mathcal{C}(t, t')$ (for $t' \leq t$) is a subcode of $\mathcal{C}(t)$ with zero set $Z_{rs} \cup Z_{bch}$, where Z_{bch} is the zero set of the primitive, narrow-sense t'-error-correcting binary BCH code i.e.

$$Z_{bch} = C_1 \cup C_2 \cup \cdots \cup C_{2t}\ , \tag{3}$$

where C_i denotes the cyclotomic coset of i modulo $n = 2^m - 1$ under multiplication by 2.

Example 1. Let α be a primitive element of GF(256).

1. $\mathcal{C}(8, 1)$ is the subcode of the 8-error-correcting (255, 239, 17) RS code ($\mathcal{C}(8)$) with zeros $\{1, 2, \cdots, 16, 32, 64, 128\}$. $\mathcal{C}(8, 1)$ is a (255, 236, $\geq$ 17) code.

2. $\mathcal{C}(8,2)$ is the subcode of the 8-error-correcting (255, 239, 17) RS code with zeros $\{1, 2, \cdots, 16, 24, 32, 48, 64, 96, 128, 129, 192\}$. $\mathcal{C}(8,2)$ is a (255, 231, $\geq$ 17) code.
3. $\mathcal{C}(6,1)$ is the subcode of the 6-error-correcting (255, 243, 13) RS code with zeros $\{1, 2, \cdots, 12, 16, 32, 64, 128\}$. $\mathcal{C}(6,1)$ is a (255, 239, $\geq$ 13) code.

3.2 Properties

The following properties can be proved for the SRS code $\mathcal{C}(t,t')$ of length $n = 2^m - 1$ over GF(2^m).

Proposition 3. *The trace of $\mathcal{C}(t,t')$ is the t'-error-correcting binary BCH code.*

Proof. This follows from Delsarte's theorem [8, Chap. 7]. □

Thus, by Proposition 1, we see that when a codeword of the binary image of $\mathcal{C}(t,t')$ is written down as a $n \times m$ matrix, each column will belong to the t'-error-correcting binary BCH code.

Proposition 4. *The subfield subcode of the SRS code $\mathcal{C}(t,t')$ is the t-error-correcting primitive binary BCH code of length n. If the primitive t-error-correcting binary BCH code has minimum distance $2t+1$, then the minimum distance of $\mathcal{C}(t,t')$ is $2t+1$.*

Proof. The result follows from Proposition 2. □

As an example, consider the (255, 239, $\geq$ 13) code $\mathcal{C}(6,1)$ over GF(256). The trace of the code is the length-255 binary Hamming code. The subfield subcode is the 6-error-correcting length-255 binary BCH code with exact minimum distance 13 [9]. Hence, $\mathcal{C}(6,1)$ is a (255, 239, 13) code over GF(256).

4 Analysis of Error-Correcting Capability

Though an SRS code is not likely to be MDS in many cases of interest, simple decoders can be designed to correct a significant fraction of errors above half the minimum distance. We analyze the error-correcting capability by introducing and studying list decoders.

4.1 List Decoders

Consider the SRS code $\mathcal{C}(t,t')$ over GF(2^m). As seen before, every codeword of the binary image of $\mathcal{C}(t,t')$ can be written down as a $n \times m$ matrix with each column belonging to the t'-error-correcting binary BCH code.

The proposed list decoder works as follows. The input to the decoder is the $n \times m$ matrix R of received bits. Let R_i denote the i^{th} column of R. The first block of the decoder is a bounded-distance decoder for the t'-error correcting

binary BCH code of length n. The BCH decoder runs on each column R_i, $1 \leq i \leq m$. The output of the i^{th} BCH decoder is denoted $\hat{R}_i$. In case of decoder failure, $\hat{R}_i = R_i$. The next step in the decoding is performed by a bank of L t-error-correcting bounded-distance RS decoders. The i^{th} decoder $(1 \leq i \leq L)$ is parametrized by a set S_i, which is a subset of $\{1, 2, \cdots, m\}$. The input to the i^{th} RS decoder is a $n \times m$ matrix whose j^{th} column is $\hat{R}_j$ if $j \in S_i$ or R_j if $j \notin S_i$ $(1 \leq j \leq m)$. The matrix is converted to a $n \times 1$ vector over GF(2^m) for decoding by the i^{th} RS decoder.

Note that the set S_i specifies the columns that are decoded by the t'-error-correcting binary BCH decoder before input to the i^{th} RS decoder. Different RS decoders have different S_i. The output from the L RS decoders forms the list of possible codewords. The maximum list size is seen to be 2^m.

4.2 Analysis of the List Decoder

We devise an algorithm to calculate the fraction of weight-w errors correctable by $\mathcal{C}(t, t')$ using the proposed list decoder with list size set as 2^m. For $w \leq t$, the fraction is 1. The calculation is done for $w > t$.

Let $P_m(w)$ denote the set of partitions of w into not more than m parts. Let p be the partition given by $w = w_1 + w_2 + \cdots + w_l$ where $w_1 \geq w_2 \geq \cdots \geq w_l$. The numbers $w_1, w_2, \ldots, w_l$ denote the number of bit errors affecting l out of the m columns of the $n \times m$ codeword matrix. Equivalently, we can think of $w_1, w_2, \ldots, w_l$ as the weights of l out of the m columns of the $n \times m$ binary error matrix E.

For a given partition $p \equiv w_1 + w_2 + \cdots + w_l$ of w, an ensemble of error patterns $\mathcal{E}(p)$ exists with the column weight distribution $\{w_1, w_2, \ldots, w_l\}$. The size of the set $\mathcal{E}(p)$ is seen to be

$$|\mathcal{E}(p)| = \frac{l!}{n_1! n_2! \cdots n_r!} \binom{m}{l} \prod_{i=1}^{l} \binom{n}{w_i} \ , \tag{4}$$

where r is the number of distinct weights in the set of weights $\{w_1, w_2, \ldots, w_l\}$, and n_i is the number of times the i-th distinct weight occurs in the set of weights. For instance, if the set of weights is $\{4, 3, 3, 1, 1\}$, then $r = 3$, $n_1 = 1$, $n_2 = 2$, and $n_3 = 2$.

Thus, the fraction of correctable errors for weight w, denoted f_w is given by

$$f_w = \frac{\sum_p P_c(p) |\mathcal{E}(p)|}{\binom{nm}{w}} \ , \tag{5}$$

where $P_c(p)$ is the probability that an error vector with column weight distribution p is correctable.

To determine $P_c(p)$, the partitions in $P_m(w)$ are modified by deleting the parts that are lesser than t' to account for the BCH decoder. Since the list size is 2^m, there exists an RS decoder parametrized by the set of columns corresponding to the parts in p of weight less than t'. For example, let $t' = 1$ and $w = 9$. Let p be

the partition given by $9 = 4{+}3{+}1{+}1$; p is modified as $\hat{p}$ given by $\hat{p} \equiv 4{+}3$. Hence, a suitable RS decoder will see an error matrix with column weight distribution $\hat{p}$. Each partition in $P_m(w)$ is modified in a similar way to form a set $\hat{P}_m(w)$. Let $\hat{p}$ be given by $\hat{p} \equiv w_1 + w_2 + \cdots + w_k$. The sum $\hat{w} = w_1 + w_2 + \cdots + w_k$ need not be equal to w; it is less than or equal to w. Based on the modified partition $\hat{p}$, we have four different cases.

1. If $\hat{p}$ is empty, it implies that all elements in the partition p were $\leq t'$. A suitable RS decoder will output the correct codeword, and $P_c(p) = 1$.
2. If $\hat{w} \leq t$, then whatever way errors are distributed along different columns, the total number of rows affected cannot exceed t. A suitable RS decoder will output the correct codeword, and $P_c(p) = 1$.
3. If $w_1 > t \geq t'$, then more than t rows will be in error for all RS decoders. By the bounded-distance property, we assume that such error patterns can never be corrected, and $P_c(p) = 0$.
4. If $\hat{p}$ does not fall into any of the above three categories, the error pattern may or may not be correctable depending on how the errors are distributed along the columns. For this case, a more detailed analysis has been done to find the probability with which the given pattern is correctable. In this case, $0 < P_c(p) < 1$.

For Case 4 above, finding $P_c(p)$ is more involved. An error matrix $E \in \mathcal{E}(p)$ for $\hat{p} \equiv w_1 + w_2 + \cdots + w_k$ is modeled by a discrete random process that involves k steps. The i^{th} step corresponds to the random placement of w_i ones in one of the m columns. Let $\{Y_1, Y_2, \ldots, Y_k\}$ be a sequence of discrete random variables. The random variable Y_i denotes the total number of rows of E affected after the i^{th} step. For instance, Y_1 denotes the number of rows of E affected after the 1^{st} step, which will be w_1 with probability 1. Y_2 denotes the number of rows affected after the 2^{nd} step. Y_2 can take any value from w_1 to $(w_1 + w_2)$ with different probabilities. The probability mass function (pmf) of Y_2 can be determined from the pmf of Y_1 and the value w_2. Similarly, we can find the pmfs of all the random variables Y_1 to Y_k starting from the pmf of Y_1 and the values $w_1, w_2, \ldots, w_k$. Finally,

$$P_c(p) = \text{Prob}\{Y_k \leq t\} \ . \tag{6}$$

Fig. 1 shows a comparison of the 8-error-correcting (255, 239, 17) RS code ($\mathcal{C}(8)$) over GF(256) and the (255, 239, 13) SRS code ($\mathcal{C}(6, 1)$) over GF(256). The simulation was done over an AWGN channel with hard-decision decoding. We see that the analysis matches with the simulated list decoder, and the SRS code is competitive with the MDS RS code of same rate down to a block-error rate of 10^{-10}.

5 Soft-Input Decoders

Because of the special structure of SRS codes, several suboptimal soft decoders of varying complexity are possible. We propose three types of soft-input decoders

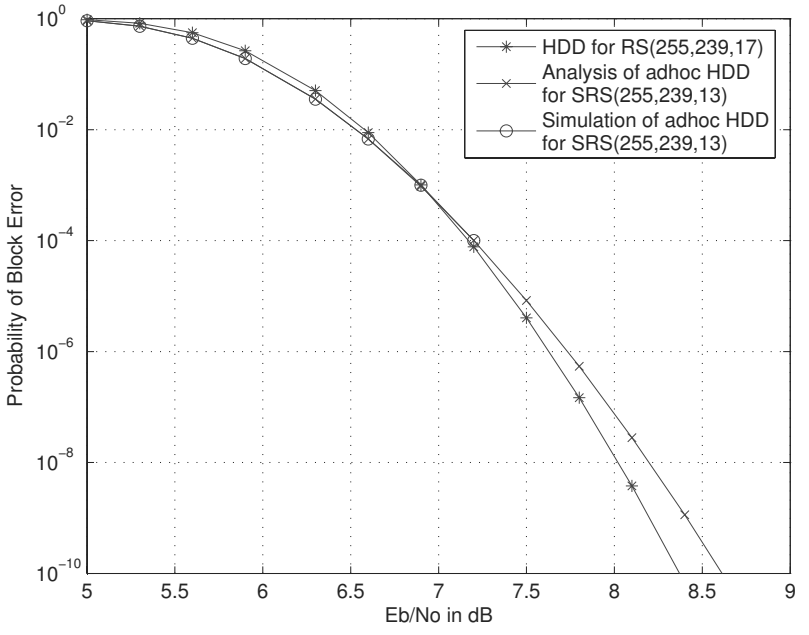


Fig. 1. Comparison of $\mathcal{C}(6,1)$ and $\mathcal{C}(8)$ over GF(256) by analysis and simulation

of increasing complexity. The codes $\mathcal{C}(6,1)$ and $\mathcal{C}(8)$ over GF(256) are chosen for comparison. Soft decoders for other codes yield similar gains.

We assume BPSK modulation over an AWGN channel. For an SRS code $\mathcal{C}(t,t')$ of length $n = 2^m - 1$ over GF(2^m), the received information R is a $n \times m$ real-valued matrix. The decoders work in two stages. The first stage decodes the columns of R according to the t'-error-correcting binary BCH code. We restrict ourselves to $t' = 1$ (Hamming code) for simplicity. The second stage decodes the output of the first stage according to the t-error-correcting RS code over GF(2^m).

5.1 Soft-Guided Decoders

In the first stage, hard-decision syndromes for the Hamming code are computed for all m columns of R. If the syndrome for the i^{th} column is 1 and the absolute received value at the error location (t' is assumed to be 1) is below a fixed threshold, the location is confirmed to be in error; otherwise, the location is assumed to be error-free. Hard decisions are made, and the confirmed error locations are flipped. The output is a $n \times m$ binary vector. The threshold is a parameter that needs to be fixed. Note that several other similar suboptimal first stages can be designed.

The second stage involves one t-error-correcting bounded-distance RS decoder on the output of the first stage. The performance of the soft-guided decoder is shown in Fig. 2. We see that the performance of a simple soft-guided decoder for

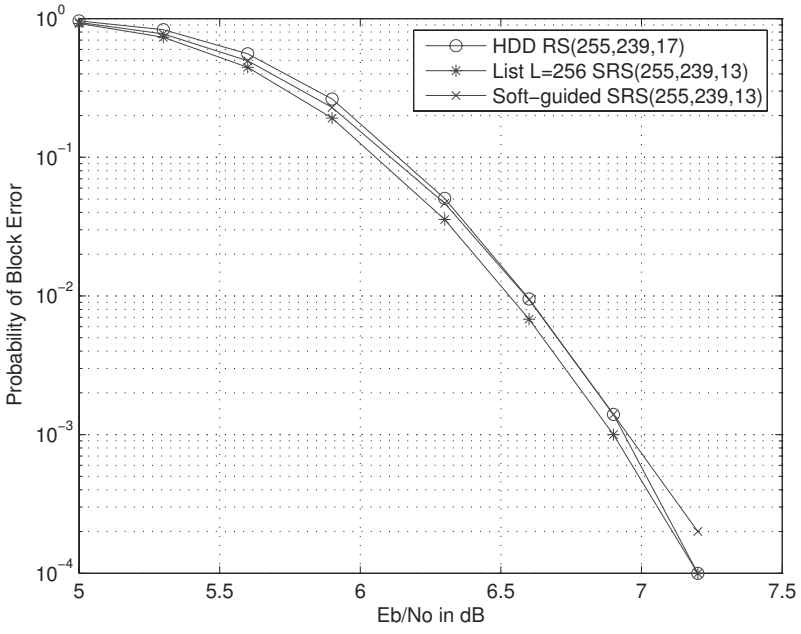


Fig. 2. Performance of soft-guided decoder

the SRS code is comparable to that of the hard-decision decoder for the MDS RS code at the same rate.

5.2 Hybrid Decoders

In hybrid soft-input decoders, the first stage is an optimal soft decoder for Hamming codes. An efficient implementation for bitwise-MAP decoders for Hamming codes can be found in [10]. The second stage is a t-error-correcting bounded-distance RS decoder. The complexity of the first stage in hybrid decoders is higher than that of soft-guided decoders.

The performance of hybrid decoders is shown in Fig. 3. We see that the hybrid decoders provide a coding gain of about 0.5 dB more than hard-decision decoders of MDS RS codes at the same rate. We also notice that additional gain is obtained by extending the SRS code.

5.3 Soft Decoders

The most complex among the soft-input decoders are the soft decoders. In the first stage, we employ the optimal bitwise MAP decoders for Hamming codes. In the second stage, the Koetter-Vardy (KV) soft-input decoder for RS codes presented in [3] is employed.

The performance of soft decoders is depicted in Fig. 4. We see that gains of about 0.9 dB over comparable hard-decoded RS codes are possible with soft

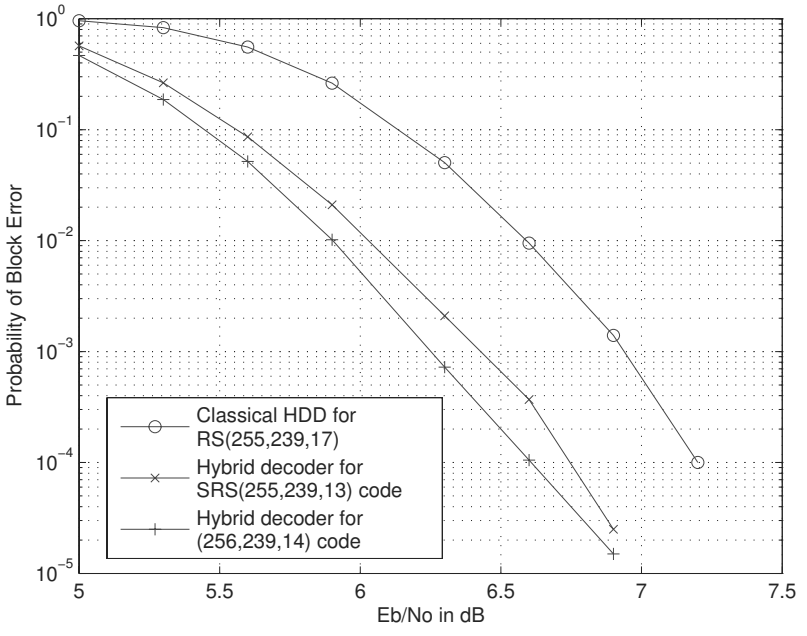


Fig. 3. Performance of hybrid decoder

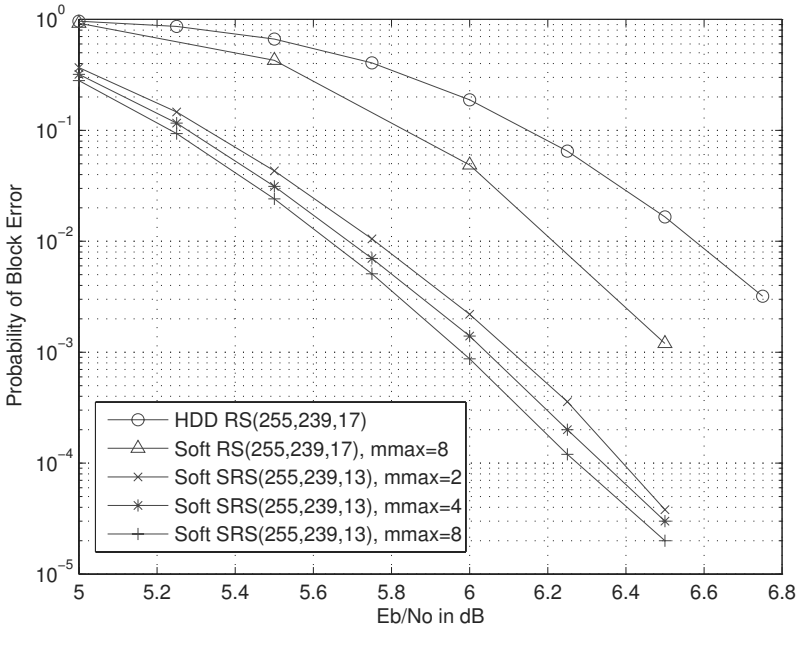


Fig. 4. Performance of soft decoder

decoders. Gains of about 0.5 dB are obtained over KV soft decoding of RS codes of same rate. The parameter 'mmax' (from [3]) indicates the complexity of the second stage.

6 Conclusion

We have studied Sub Reed-Solomon (SRS) codes, which are certain subcodes of Reed-Solomon codes with a nontrivial trace code. The trace structure results in the possibility of hard-decision list decoding beyond half the minimum distance and efficient soft-input decoding. The performance results, when compared to that of maximum-distance-separable Reed-Solomon codes, show a best-possible gain of about 0.9 dB. With reasonable complexity, gains of about 0.5 dB are possible.

References

1. Reed, I.S., Solomon, G.: Polynomial codes over certain finite fields. J. SIAM 8, 300–304 (1960)
2. Guruswami, V., Sudan, M.: Improved decoding of Reed-Solomon and algebraic-geometry codes. IEEE Trans. on Info. Theory 45(6), 1757–1767 (1999)
3. Koetter, R., Vardy, A.: Algebraic Soft-decision Decoding of Reed-Solomon Codes. IEEE Trans. Inform. Theory 49(11), 2809–2825 (2003)
4. Forney, D.: Generalized Minimum Distance Decoding. IEEE Trans. Inform. Theory 12(2), 125–131 (1966)
5. Vardy, A., Beery, Y.: Bit-level Soft-decision Decoding of Reed-Solomon Codes. IEEE Trans. on Comm. 39(3), 440–444 (1991)
6. Ponnampalam, V., Vucetic, B.: Soft Decision Decoding of Reed-Solomon Codes. IEEE Trans. on Comm. 50(11), 1758–1768 (2002)
7. Jiang, J., Narayanan, K.R.: Iterative Soft Decoding of Reed-Solomon Codes. IEEE Commun. Lett. 8(4), 244–246 (2004)
8. MacWilliams, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes. North-Holland, The Netherlands, Amsterdam (1977)
9. Augot, D., Charpin, P., Sendrier, N.: Studying the Locator Polynomials of Minimum Weight Codewords of BCH Codes. IEEE Trans. Inform. Theory 38(3), 960–973 (1992)
10. Ashikhmin, A., Litsyn, S.: Simple MAP Decoding of First-Order Reed-Muller and Hamming Codes. IEEE Trans. Inform. Theory 50(8), 1812–1818 (2004)

Normalized Minimum Determinant Calculation for Multi-block and Asymmetric Space-Time Codes

Camilla Hollanti[1] and Hsiao-feng (Francis) Lu[2]

[1] Department of Mathematics, FIN-20014 University of Turku, Finland
camilla.hollanti@utu.fi
[2] Department of Communication Engineering,
National Chung-Cheng University, Chia-yi, Taiwan
hsiaofel@commsci1.usc.edu

Abstract. The aim of this paper is to show the connection between certain, previously constructed multi-block and asymmetric space-time codes. The Gram determinants of the two constructions coincide, and hence the corresponding lattices share the same density. Using the notion of density, we define the normalized minimum determinant and give an implicit lower bound depending on the center of the cyclic division algebra in use. The calculation of the normalized minimum determinant is then performed in practice by using explicit code constructions.

Keywords: Asymmetric space-time block codes (ASTBCs), cyclic division algebras (CDAs), dense lattices, discriminants, diversity-multiplexing tradeoff (DMT), maximal orders, multi-block, multiple-input multiple-output (MIMO) channels, nonvanishing determinant (NVD).

1 Background

Previously, different methods for constructing asymmetric [1],[2] and multi-block [3] space-time codes have been proposed. *Asymmetric* codes are targeted at the code design for downlink transmission where the number of Rx antennas is strictly less than the number of Tx antennas. Typical examples of such situations are 3+G mobile phones and DVB-H (Digital Video Broadcasting-Handhelds) user equipment, where only a very small number of antennas fits at the end user site. The best code in [1] was shown to improve upon the punctured Perfect code [2] as well as the DjABBA code [2] in the BLER performance at the data rate 4 bpcu, hence proving that the methods proposed therein come into good use.

Multi-block codes, for their part, are used when one wishes to obtain vanishing error probability in addition to the D-M tradeoff optimality. In this work, we concentrate on the minimal delay multi-block construction given in [3] and the asymmetric construction given in [1] by *Method 1*. In [4] an approach similar to Method 1 was used for the MIMO amplify-and-forward cooperative channel.

Already in [1] we stated that Method 1 can be converted to produce multi-block ST codes [3] that do achieve the DMT. Here, we shall show this explicitly

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 227–236, 2007.
© Springer-Verlag Berlin Heidelberg 2007

and prove that maximizing the density (i.e. finding the most efficient packing in the available signal space) of asymmetric and multi-block codes arising from this method is equivalent to minimizing the discriminant of a certain order.

We define a *lattice* to be a discrete finitely generated free abelian subgroup $\mathbf{L}$ of a real or complex finite dimensional vector space, called the ambient space. In the space-time (ST) setting a natural ambient space is the space $\mathcal{M}_n(\mathbf{C})$ of complex $n \times n$ matrices. The *Gram matrix* is defined as

$$G(\mathbf{L}) = \left(\Re tr(x_i x_j^H)\right)_{1 \leq i,j \leq k}, \tag{1}$$

where H indicates the complex conjugate transpose of a matrix, tr is the matrix trace (=sum of the diagonal elements), and x_i, $i = 1, ..., k$, form a $\mathbf{Z}$-basis of $\mathbf{L}$. The rank k of the lattice is upper bounded by $2n^2$. The Gram matrix has a positive determinant equal to the squared measure of the fundamental parallelotope $m(\mathbf{L})^2$. A change of basis does not affect the measure $m(\mathbf{L})$.

Any lattice $\mathbf{L}$ with the *nonvanishing determinant* (NVD) property [5] can be scaled, i.e. multiplied by a real constant r, either to satisfy $\det_{min}(\mathbf{L}) = 1$ or to satisfy $m(\mathbf{L}) = 1$. This is because $\det_{min}(r\mathbf{L}) = r^n \det_{min}(\mathbf{L})$ and $m(r\mathbf{L}) = r^k m(\mathbf{L})$. As the minimum determinant determines the asymptotic pairwise error probability (PEP), this gives rise to natural numerical measures for the quality of a lattice. Following [6], we denote by $\delta(\mathbf{L})$ the *normalized minimum determinant* of the lattice $\mathbf{L}$, i.e. here we first scale $\mathbf{L}$ to have a unit size fundamental parallelotope. Dually we denote by $\rho(\mathbf{L}) = 1/m(\mathbf{L})$ the *normalized density* of the lattice $\mathbf{L}$, when we first scale the lattice to have unit minimum determinant, and only then compute the quantity $1/m(\mathbf{L})$.

It has been shown in [7] that CDA-based square ST codes with the NVD property achieve the diversity-multiplexing tradeoff (DMT) introduced in [8]. This result also extends over multi-block space-time codes [3].

For more information on matrix representations of division algebras and their use as MIMO STBCs the reader can refer to [9]-[13], just to name a few.

2 Cyclic Division Algebras and Orders

The theory of cyclic algebras and their representations as matrices are thoroughly considered in [9] and [14]. We are only going to recapitulate the essential facts here. For a more detailed introduction on orders, see [15].

In the following, we consider number field extensions E/F, where F denotes the base field and F^* (resp. E^*) denotes the set of the non-zero elements of F (resp. E). The rings of algebraic integers are denoted by $\mathcal{O}_F$ and $\mathcal{O}_E$ respectively. Let E/F be a cyclic field extension of degree n with Galois group $\mathrm{Gal}(E/F) = \langle\sigma\rangle$, where σ is the generator of the cyclic group. Let $\mathcal{A} = (E/F, \sigma, \gamma)$ be the corresponding cyclic algebra of degree n (n is also called the *index* of $\mathcal{A}$ and in practice it determines the number of transmitters), that is

$$\mathcal{A} = E \oplus uE \oplus u^2E \oplus \cdots \oplus u^{n-1}E,$$

with $u \in \mathcal{A}$ such that $eu = u\sigma(e)$ for all $e \in E$ and $u^n = \gamma \in F^*$. An element $x = x_0 + ux_1 + \cdots + u^{n-1}x_{n-1} \in \mathcal{A}$ has the following representation as a matrix

$$A = \begin{pmatrix} x_0 & \gamma\sigma(x_{n-1}) & \gamma\sigma^2(x_{n-2}) & \cdots & \gamma\sigma^{n-1}(x_1) \\ x_1 & \sigma(x_0) & \gamma\sigma^2(x_{n-1}) & & \gamma\sigma^{n-1}(x_2) \\ x_2 & \sigma(x_1) & \sigma^2(x_0) & & \gamma\sigma^{n-1}(x_3) \\ \vdots & & & & \vdots \\ x_{n-1} & \sigma(x_{n-2}) & \sigma^2(x_{n-3}) & \cdots & \sigma^{n-1}(x_0) \end{pmatrix}. \tag{2}$$

Definition 1. *An algebra $\mathcal{A}$ is called simple if it has no nontrivial ideals. A cyclic algebra $\mathcal{A} = (E/F, \sigma, \gamma)$ is central if its center $Z(\mathcal{A}) = \{x \in \mathcal{A} \mid xx' = x'x \text{ for all } x' \in \mathcal{A}\} = F$.*

All algebras considered here are finite dimensional associative central simple algebras over a field. From now on, we identify the element x of an algebra with its standard matrix representation defined above in (2).

Definition 2. *The determinant of the matrix A is called the reduced norm of the element $x \in \mathcal{A}$ and is denoted by $nr(x)$.*

Remark 1. The connection between the usual norm map $N_{\mathcal{A}/F}(a)$ and the reduced norm $nr(a)$ of an element $a \in \mathcal{A}$ is $N_{\mathcal{A}/F}(a) = (nr(a))^n$, where n is the degree of E/F.

In the following we give a condition when an algebra is a division algebra, i.e. each of its non-zero elements has a multiplicative inverse. For the proof, see [14, Theorem 11.12, p. 184].

Proposition 1. *An algebra $\mathcal{A} = (E/F, \sigma, \gamma)$ of index n is a division algebra if and only if the smallest factor $t \in \mathbf{Z}_+$ of n such that γ^t is the norm of some element in E^* is n.*

Let R (e.g. $R = \mathbf{Z}[i]$) denote a Noetherian integral domain with a quotient field F (e.g. $F = \mathbf{Q}(i)$), and let $\mathcal{A}$ be a finite dimensional F-algebra.

Definition 3. *An R-order in the F-algebra $\mathcal{A}$ is a subring Λ of $\mathcal{A}$, having the same identity element as $\mathcal{A}$, and such that Λ is a finitely generated module over R and generates $\mathcal{A}$ as a linear space over F.*

As usual, an R-order in $\mathcal{A}$ is said to be *maximal*, if it is not properly contained in any other R-order in $\mathcal{A}$.

Next we describe an order from where the elements are drawn in a typical CDA based MIMO space-time block code. For the proof of Proposition 2, see [15, Theorem 10.1, p. 125]. Some optimization to this can be done e.g. with the aid of ideals as in [10] or by using a maximal order [13].

Definition 4. *In any cyclic division algebra we can always choose the element $\gamma \in F^*$ determining the 2-cocycle in $H^2(E/F)$ to be an algebraic integer. We immediately see that the $\mathcal{O}_F$-module*

$$\Lambda_{NAT} = \mathcal{O}_E \oplus u\mathcal{O}_E \oplus \cdots \oplus u^{n-1}\mathcal{O}_E$$

is an $\mathcal{O}_F$-order in the cyclic algebra $(E/F, \sigma, \gamma)$. We refer to this $\mathcal{O}_F$-order as the natural order. An alternative appellation would be layered order, as the corresponding MIMO-lattice of this order has the layered structure described in [16].

Proposition 2. *For any non-zero element $x \in \Lambda_{NAT}$ its reduced norm $nr(x)$ is a non-zero element of the ring of integers $\mathcal{O}_F$ of the center F. In particular, if F is an imaginary quadratic number field or a cyclotomic field, then the minimum determinant of the lattice Λ_{NAT} is nonvanishing and equal to one. More generally, if x is an element of an R-order Λ, then $nr(x) \in R$.*

Remark 2. Note that if $\gamma \in F^*$ is not an algebraic integer, then an order Λ fails to be closed under multiplication. This may adversely affect the minimum determinant of the resulting matrix lattice as elements not belonging to an order may have non-integral and hence small norms. One of the motifs underlying the perfect codes [10] is the requirement that the variable γ should have a unit modulus. Relaxing this restriction on the size of γ will lead to an antenna power imbalance in both space and time domains. The measure of the fundamental parallelotope varies with different algebras. Hence, one has to keep in mind that, on the other hand, an algebra with a unit γ may still admit larger average energy than a different algebra with a non-unit γ so the size of γ is not the only parameter to stare at.

Definition 5. *Let $m = \dim_F \mathcal{A}$. The discriminant of the R-order Λ is the ideal $d(\Lambda/R)$ in R generated by the set*

$$\{\det tr(x_i x_j)_{i,j=1}^m \mid (x_1, ..., x_m) \in \Lambda^m\}.$$

In the interesting cases of $F = \mathbf{Q}(i)$, $i = \sqrt{-1}$ (resp. $F = \mathbf{Q}(\sqrt{-3})$) the ring $R = \mathbf{Z}[i]$ (resp. $R = \mathbf{Z}[\omega]$, $\omega = (-1+\sqrt{-3})/2$) is a Euclidean domain, so in these cases as well as in the case $R = \mathbf{Z}$ it makes sense to speak of the discriminant as an element of R rather than as an ideal. We simply compute the discriminant as $d(\Lambda/R) = \det tr(x_i x_j)_{i,j=1}^m$, where $\{x_1, \ldots, x_m\}$ is any R-basis of Λ.

Remark 3. It is readily seen that whenever $\Lambda \subseteq \Gamma$ are two R-orders, then $d(\Gamma/R)$ is a factor of $d(\Lambda/R)$. It also turns out (cf. [15, Theorem 25.3]) that all the maximal orders of a division algebra share the same discriminant. In this sense a maximal order has the smallest possible discriminant among all orders within a given division algebra, as all the orders are contained in the maximal one.

To conclude the section, we include the following simple but interesting result on maximal orders explaining why using a principal one-sided (left or right) ideal instead of the entire order will not change the density of the code. For the proof, see [13, Lemma 7.1].

Lemma 1. *Let Λ be a maximal order in a cyclic division algebra over an imaginary quadratic number field. Assume that the minimum determinant of the lattice Λ is equal to one. Let $x \in \Lambda$ be any non-zero element. Let $\rho > 0$ be a real parameter chosen such that the minimum determinant of the lattice $\rho(x\Lambda)$ is also equal to one. Then the fundamental parallelotopes of these two lattice have the same measure*

$$m(\Lambda) = m(\rho(x\Lambda)).$$

3 Block Diagonal Asymmetric ST Lattices

In this section, we recall *Method 1* from [1]. Let us rename this method as *Block Diagonal Method* (BDM).

Let us consider an extension tower $F \subseteq L \subseteq E$ with the degrees $[E : L] = r, [L : F] = m$ and with the Galois groups $\mathrm{Gal}(E/F) = \langle\tau\rangle$, $\mathrm{Gal}(E/L) = \langle\sigma = \tau^m\rangle$. Let $\mathcal{B} = (E/L, \sigma, \gamma) = E + \cdots + u^{r-1}E$ be an index r division algebra, where the center L is fixed by $\sigma = \tau^m$. We denote by $\#\mathrm{Tx} = n = rm$.

Note that if one has a symmetric, index $n = rm$ CDA based STBC, the algebra $\mathcal{B}$ can be constructed by just picking a suitable intermediate field $L \subseteq E$ of a right degree as the new center.

An element $b = x_0 + \cdots + u^{r-1}x_{r-1},\ x_i \in E,\ i = 0, ..., r-1$ of the algebra $\mathcal{B}$ has a representation as an $r \times r$ matrix $B = (b_{ij})_{1 \le i,j \le r}$ as given in (2).

However, we can afford an $n \times n$ packing as we are using n transmitters. This can be achieved by using the isomorphism τ. Let us denote by $\tau^k(\mathcal{B}) = (E/L, \sigma, \tau^k(\gamma)),\ k = 0, ..., m-1$ the m isomorphic copies of $\mathcal{B}$ and the respective matrix representations by

$$\tau^k(B) = (\tau^k(b_{ij}))_{1 \le i,j \le r},\ k = 0, ..., m-1. \tag{3}$$

The next proposition shows that by using these copies as diagonal blocks we obtain an infinite lattice with nonvanishing determinant. For the proof, see [1].

Proposition 3. ***(BDM)*** *Let $b \in \Lambda \subseteq \mathcal{B}$ and $F = \mathbf{Q}(\delta)$, where $\delta \in \{i, \omega\}$. Assume $\gamma \in \mathcal{O}_L$. The lattice*

$$\mathcal{C}(\Lambda) = \left\{ M = diag\left(B, \tau(B), \ldots, \tau^{m-1}(B)\right) \right\}$$

built from (3) has a nonvanishing determinant $\det \mathcal{C}(\Lambda) = \prod_{i=0}^{m-1} \det \tau^i(B) \in \mathbf{Z}[\delta]$. Thus, the minimum determinant is equal to one for all numbers of fading blocks m. The code rate equals $r^2m/rm = r$.

Now the natural question is how to choose a suitable division algebra. In [7] and [12] several systematic methods for constructing extensions E/L are provided. All of them make use of cyclotomic fields. In [1] we proved that, in the asymmetric scheme, maximizing the code density (i.e. minimize the volume of the fundamental parallelotope, see [13]) with a given minimum determinant is equivalent to minimizing a certain discriminant. In the next section we shall show that this also holds for the multi-block codes from [3].

First we need the following result. For the proof, see [15, p. 223].

Lemma 2. *Suppose $\Lambda \subseteq \mathcal{A} = (E/L, \tau, \gamma)$ is an $\mathcal{O}_F$-order and that $F \subseteq L$. The discriminants then satisfy*

$$d(\Lambda/\mathcal{O}_F) = N_{L/F}\left(d(\Lambda/\mathcal{O}_L)\right) d(\mathcal{O}_L/\mathcal{O}_F)^{dim_L \mathcal{A}}.$$

The same naturally holds in the commutative case when we replace $\mathcal{A}$ with E.

The definition of the discriminant closely resembles that of the Gram matrix of a lattice, so the following results are rather unsurprising. For the proof, see [1].

Proposition 4. *Assume that F is an imaginary quadratic number field and that $\{1, \rho\}$ forms a $\mathbf{Z}$-basis of its ring of integers $\mathcal{O}_F$. Let $r = [E : L]$, $m = [L : F]$, $n = rm$, and $s = |\Im\rho|^{mr^2}$. If the order $\mathcal{C}(\Lambda)$ defined as in Proposition 3 is a free $\mathcal{O}_F$-module (which is always the case if $\mathcal{O}_F$ is a principal ideal domain), then the measure of the fundamental parallelotope equals*

$$\begin{aligned} m(\mathcal{C}(\Lambda)) &= s|d(\Lambda/\mathcal{O}_F)| = s|d(\mathcal{O}_L/\mathcal{O}_F)^{r^2} N_{L/F} d(\Lambda/\mathcal{O}_L)| \\ &= s|d(\mathcal{O}_L/\mathcal{O}_F)^{r^2} \prod_{i=0}^{m-1} \tau^i(d(\Lambda/\mathcal{O}_L))|. \end{aligned}$$

Corollary 1. *In the case $F = \mathbf{Q}(i)$ we get $m(\mathcal{C}(\Lambda)) = |d(\Lambda/\mathbf{Z}[i])|$. For $F = \mathbf{Q}(\omega)$ the volume equals $m(\mathcal{C}(\Lambda)) = (\frac{\sqrt{3}}{2})^{mr^2} |d(\Lambda/\mathbf{Z}[\omega])|$.*

Now we can conclude that the extensions $E/L, L/F$ and the order $\Lambda \subseteq \mathcal{B}$ should be chosen such that the discriminants $d(\mathcal{O}_L/\mathcal{O}_F)$ and $d(\Lambda/\mathcal{O}_L)$ are as small as possible. By choosing a maximal order within a given division algebra we can minimize the norm of $d(\Lambda/\mathcal{O}_L)$ (cf. Remark 3). As in practice an imaginary quadratic number field F is contained in L, we know that L is totally complex. In that case the fact that

$$d(\Lambda/\mathcal{O}_L) \geq (P_1 P_2)^{r(r-1)}, \tag{4}$$

where P_1 and P_2 are prime ideals $\in \mathcal{O}_L$ with the smallest norms (to $\mathbf{Q}$) helps us in picking a good algebra (for the proof, see [13, Theorem 3.2]).

Remark 4. Note that as opposed to [13], here we do not achieve nice, explicit lower bounds for $d(\Lambda/\mathcal{O}_L)$. That is a consequence of the fact that the center L can now be almost anything that just contains $\mathbf{Z}[i]$ or $\mathbf{Z}[\omega]$. An exact lower bound of course exists, but we have not been searching for it yet. We hope to provide this lower bound in a forthcoming paper.

Remark 5. In [13] we have studied the use of maximal orders in the design of dense, symmetric, CDA based MIMO STBCs in more detail. The same ideas can be adapted to asymmetric and multi-block scheme as well.

4 Minimal Delay Multi-block ST Codes

The nTx+rRx antenna AST code from Proposition 3 can be transformed into an rTx+rRx antenna multi-block code [3] by an evident rearrangement of the blocks:

$$\operatorname{diag}\left(B, \tau(B), \ldots, \tau^{m-1}(B)\right) \leftrightarrow \left(B, \cdots, \tau^{m-1}(B)\right). \tag{5}$$

As the Gram matrices of an AST lattice and a multi-block ST lattice coincide, Lemma 4 also holds for multi-block ST codes with the same parameters. Let the notation be as in Section 3.

Proposition 5. *Let $b \in \Lambda \subseteq \mathcal{B}$ and $F = \mathbf{Q}(\delta)$, where $\delta \in \{i, \omega\}$. Assume $\gamma \in \mathcal{O}_L$. As the lattice*

$$\mathcal{C}'(\Lambda) = \left\{M = \left(B, \tau(B), \ldots, \tau^{m-1}(B)\right)\right\}$$

built from (3) satisfies the generalized nonvanishing determinant property (cf. [3],[11]), it is optimal with respect to the D-M tradeoff for all numbers of fading blocks m. Similarly as in Proposition 3, $\prod_{i=0}^{m-1} \det \tau^i(B) \geq 1$. The code rate equals $r^2 m/rm = r$.

Proof. For the proof, see [3].

Proposition 6. *The Gram determinants (cf. (1)) of the lattices $\mathcal{C}(\Lambda)$ and $\mathcal{C}'(\Lambda)$ coincide:*

$$\det G(\mathcal{C}(\Lambda)) = \det G(\mathcal{C}'(\Lambda)).$$

Proof. This is obvious, as $tr(diag(BB^H, .., \tau^{m-1}(B)\tau^{m-1}(B)^H)) =$
$= \sum_{i=0}^{m-1} tr(\tau^i(B)\tau^i(B)^H) = tr(\sum_{i=0}^{m-1}(\tau^i(B)\tau^i(B)^H))$.

An immediate consequence of Proposition 6 is

Corollary 2. *The lattices $\mathcal{C}(\Lambda)$ and $\mathcal{C}'(\Lambda)$ share the same density, i.e. Proposition 4 can be adapted as such to the multi-block scheme.*

5 Explicit Codes

In this section we provide explicit asymmetric constructions for the important case of 4Tx + 2Rx antennas. These codes can be modified for 2×2 multi-block use (cf. (5). The primitive nth root of unity will be denoted by ζ_n. The first three examples are given in terms of an asymmetric construction, whereas the last one is described as a multi-block code. However, with the aid of (5), an asymmetric code can always be transformed into a multi-block code and vice versa.

5.1 Perfect Algebra $\mathcal{PA}$

Let us consider an algebra with the same maximal subfield that was used for the 4×4 Perfect code in [10]. We have the nested sequence of fields $F \subseteq L \subseteq E$, where $F = \mathbf{Q}(i)$, $L = \mathbf{Q}(\sqrt{5}, i)$, and $E = \mathbf{Q}(\theta, i)$ with $\theta = \zeta_{15} + \zeta_{15}^{-1} = 2\cos(2\pi/15)$.

We denote this algebra by $\mathcal{PA} = (E/L, \sigma = \tau^2, \gamma) = E \oplus uE$, where $u^2 = \gamma = i$ and $\tau(\theta) = \theta^2 - 2$. As $\tau(\sqrt{5}) = -\sqrt{5}$, the field L is indeed fixed by $\sigma = \tau^2$. By embedding the algebra $\mathcal{PA}$ as in Proposition 3 we obtain the AST code

$$\mathcal{PA}_1 \subseteq \left\{ \begin{pmatrix} x_0 & i\sigma(x_1) & 0 & 0 \\ x_1 & \sigma(x_0) & 0 & 0 \\ 0 & 0 & \tau(x_0) & i\tau(\sigma(x_1)) \\ 0 & 0 & \tau(x_1) & \tau(\sigma(x_0)) \end{pmatrix} \;\middle|\; x_i \in \mathcal{O}_E \right\}.$$

As the center is L with $[L : \mathbf{Q}(i)] = 2$ and $\mathcal{O}_L = \mathbf{Z}[i, \mu = (1+\sqrt{5})/2]$, the elements x_i in the matrix are of the form $a_1 + a_2\mu + a_3\theta + a_4\mu\theta$, where $a_i \in \mathbf{Z}[i]$ for all i. Thus, the code transmits, on the average, 2 independent QAM symbols per channel use.

We can further improve the performance by taking the elements x_i from the ideal $a\mathcal{O}_E$, where $a = 1 - 3i + i\theta^2 \in \mathcal{O}_E$. Moreover, a change of basis given by

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -3 & 0 & 1 \\ -1 & -3 & 1 & 1 \end{pmatrix}$$

guarantees an orthogonal basis.

5.2 Cyclotomic Algebra $\mathcal{CA}$

The algebra $\mathcal{CA} = (E/L, \sigma = \tau^2 : \xi \mapsto -\xi, \gamma = 1 + s - i) = E \oplus uE$ (cf. [11], [13], [1]), for its part, has the nested sequence of fields $F \subseteq L \subseteq E$ with $F = \mathbf{Q}(i)$, $L = \mathbf{Q}(s = \zeta_8)$, and $E = \mathbf{Q}(\xi = \zeta_{16})$. As we have $\tau : \xi \mapsto i\xi, s \mapsto -s$, the field L is fixed by $\sigma = \tau^2$. Again by embedding the algebra $\mathcal{CA}$ as in Proposition 3, the AST code

$$\mathcal{CA}_1 \subseteq \left\{ \begin{pmatrix} x_0 & \gamma\sigma(x_1) & 0 & 0 \\ x_1 & \sigma(x_0) & 0 & 0 \\ 0 & 0 & \tau(x_0) & \tau(\gamma)\tau(\sigma(x_1)) \\ 0 & 0 & \tau(x_1) & \tau(\sigma(x_0)) \end{pmatrix} \;\middle|\; x_i \in \mathcal{O}_E \right\}$$

is obtained. The center is L with $[L : \mathbf{Q}(i)] = 2$ and $\mathcal{O}_L = \mathbf{Z}[s]$. The elements x_i in the matrix are of the form $a_1 + a_2 s + a_3\xi + a_4 s\xi$, where $a_i \in \mathbf{Z}[i]$ for all i. Hence the above code is transmitting again, on the average, 2 independent QAM symbols per channel use.

Note that we have chosen here a suitable non-norm element γ from $\mathcal{O}_L$ instead of $\mathcal{O}_F$ (cf. Section 3). We get some energy savings as $|1 + s - i| < |2 + i|$.

5.3 Algebra $\mathcal{IA}$ – An Improved Maximal Order

Similarly as in the two previous subsections, we obtain an rate-2 AST code $\mathcal{IA}_1$ by introducing yet another algebra $\mathcal{IA} = (E/L, \sigma = \tau^2, \gamma = \sqrt{-3})$, where $F = \mathbf{Q}(i)$, $L = \mathbf{Q}(i, \sqrt{3})$, $E = L(\sqrt{1+i})$, and $\tau : \sqrt{3} \mapsto -\sqrt{3}, \sqrt{1+i} \mapsto -\sqrt{1+i}$.

Among our example algebras, $\mathcal{IA}$ has the densest maximal order.

5.4 Algebra $\mathcal{QA}$ – An Improved Natural Order

Let us use the multi-block notation for a change. Here we consider another tower of number fields $F \subset L \subset E$, where $E = \mathbf{Q}(\zeta_5, i)$, $F = \mathbf{Q}(i)$, and where $L = \mathbf{Q}(\theta, i)$ with $\theta = \zeta_5 + \zeta_5^{-1}$. Clearly we have $\mathrm{Gal}(E/F) = \langle\tau\rangle$, $\tau(\zeta_5) = \zeta_5^2$, and $\tau(\theta) = \theta^2 - 2$. Thus we obtain the CDA $\mathcal{QA} = (E/L, \sigma = \tau^2, \gamma) = E \oplus uE$, and $\gamma = u^2 = i$ is a non-norm element. Embedding the algebra $\mathcal{QA}$ as in Proposition 3 yields the following multi-block ST code with coding over 2 consecutive fading blocks:

$$\mathcal{QA}_1 \subseteq \left\{ B = \begin{pmatrix} x_0 & i\sigma(x_1) \\ x_1 & \sigma(x_0) \end{pmatrix}, \tau(B) = \begin{pmatrix} \tau(x_0) & i\tau(\sigma(x_1)) \\ \tau(x_1) & \tau(\sigma(x_0)) \end{pmatrix} \middle| \, x_i \in \mathcal{O}_E \right\}.$$

The elements x_i in the above are of the form $x_i = \sum_{j=0}^{3} a_{i,j}\zeta_5^j$, where $a_{i,j} \in \mathbf{Z}[i]$, hence the above code transmits on the average, 2 independent QAM symbols per channel use.

Among our example algebras, $\mathcal{QA}$ has the densest natural order.

Table 1. Normalized minimum determinant δ and normalized density $\rho = 1/m(\Lambda)$ of natural and maximal orders of different algebras

	$\mathcal{PA}$	$\mathcal{PA}$	$\mathcal{CA}$	$\mathcal{CA}$	$\mathcal{IA}$	$\mathcal{IA}$	$\mathcal{QA}$
	Λ_{NAT}	Λ_{MAX}	Λ_{NAT}	Λ_{MAX}	Λ_{NAT}	$\boldsymbol{\Lambda_{MAX}}$	$\boldsymbol{\Lambda_{NAT=MAX}}$
δ	0.0298	0.0894	0.0361	0.1214	0.0340	**0.1361**	**0.0894**
ρ	$3^{-4} \cdot 5^{-6} =$ $7.9 \cdot 10^{-7}$	$5^{-6} =$ $6.4 \cdot 10^{-5}$	$2^{-16} \cdot 3^{-2} =$ $1.7 \cdot 10^{-6}$	$2^{-9} \cdot 3^{-2} =$ $2.2 \cdot 10^{-4}$	$2^{-10} \cdot 3^{-6} =$ $1.4 \cdot 10^{-6}$	$2^{-2} \cdot 3^{-6} =$ $\mathbf{3.4 \cdot 10^{-4}}$	$5^{-6} =$ $\mathbf{6.4 \cdot 10^{-5}}$

Example 1. Let us calculate the normalized minimum determinant of the algebra $\mathcal{IA}$ as an example (cf. Section 1, Definitions 4, 5, and Propositions 3 and 4). The other algebras can be treated likewise. In Table 1 we have listed the normalized minimum determinants δ and densities ρ of the natural and maximal orders of the algebras $\mathcal{PA}, \mathcal{CA}, \mathcal{IA}$, and $\mathcal{QA}$. Note that for $\mathcal{QA}$ these two actually coincide. We can conclude that among the natural orders, that of the algebra $\mathcal{QA}$ has the largest normalized minimum determinant, i.e. the highest density. The algebra $\mathcal{IA}$, for its part, has the densest maximal order. The corresponding numbers are shown **bold** in Table 1.

For the natural order of $\mathcal{IA}$ we have $\det_{min}(\mathcal{C}(\Lambda_{NAT})) = 1$ and $\rho^{-1} = m(\mathcal{C}(\Lambda_{NAT})) = 2^{10} \cdot 3^6$, hence $r = 2^{-5/8} \cdot 3^{-3/8}$. Now $m(r\mathcal{C}(\Lambda_{NAT})) = 1$ and the normalized minimum determinant is $\delta = \det_{min}(r\mathcal{C}(\Lambda_{NAT})) = 2^{-5/2} \cdot 3^{-3/2} \cdot 1 \approx 0.0340$.

The maximal order of $\mathcal{IA}$ has $\det_{min}(\mathcal{C}(\Lambda_{MAX})) = 1$ and $m(\mathcal{C}(\Lambda_{MAX})) = 2^2 \cdot 3^6$, thus $r = 2^{-1/8} \cdot 3^{-3/8}$ and $\delta = \det_{min}(r\mathcal{C}(\Lambda_{MAX})) = \frac{1}{3\sqrt{2}\sqrt{3}} \approx 0.1361$.

References

1. Hollanti, C., Ranto, K.: Asymmetric Space-Time Block Codes for MIMO Systems. In: 2007 IEEE ITW, Bergen, Norway, pp. 101–105 (2007)
2. Hottinen, A., Hong, Y., Viterbo, E., Mehlführer, C., Mecklenbraüker, C.F.: A Comparison of High Rate Algebraic and Non-Orthogonal STBCs. In: 2007 ITG/IEEE WSA 2007, Vienna, Austria (2007)
3. Lu, H.F.F.: Explicit Constructions of Multi-Block Space-Time Codes that Achieve the Diversity-Multiplexing Tradeoff. In: 2006 IEEE ISIT, Seattle, pp. 1149–1153 (2006)
4. Yang, S., Belfiore, J.-C.: Optimal Space-Time Codes for the MIMO Amplify-and-Forward Cooperative Channel. IEEE Trans. Inform. Theory 53, 647–663 (2007)
5. Belfiore, J.-C., Rekaya, G.: Quaternionic Lattices for Space-Time Coding. In: IEEE ITW 2003, Paris, France (2003)
6. Lahtonen, J.: Dense MIMO Matrix Lattices and Class Field Theoretic Themes in Their Construction. In: IEEE ITW 2007, Bergen, Norway, pp. 96–100 (2007)
7. Elia, P., Kumar, K.R., Pawar, S.A., Kumar, P.V., Lu, H.F.F.: Explicit Space-Time Codes Achieving the Diversity-Multiplexing Gain Tradeoff. IEEE Trans. Inf. Theory 52, 3869–3884 (2006)
8. Zheng, L., Tse, D.: Diversity and Multiplexing: A Fundamental Tradeoff in Multiple-Antenna Channels. IEEE Trans. Inform. Theory 49, 1073–1096 (2003)
9. Sethuraman, B.A., Rajan, B.S., Shashidhar, V.: Full-Diversity, High-Rate Space-Time Block Codes From Division Algebras. IEEE Trans. Inform. Theory 49, 2596–2616 (2003)
10. Belfiore, J.-C., Oggier, F., Rekaya, G., Viterbo, E.: Perfect Space-Time Block Codes. IEEE Trans. Inform. Theory 52, 3885–3902 (2006)
11. Kiran, T., Rajan, B.S.: STBC-Schemes with Non-Vanishing Determinant For Certain Number of Transmit Antennas. IEEE Trans. Inform. Theory 51, 2984–2992 (2005)
12. Lu, H.F.F., Elia, P., Kumar, K.R., Pawar, S.A., Kumar, P.V.: Space-Time Codes Meeting the Diversity-Multiplexing Gain Tradeoff with Low Signalling Complexity. In: 2005 CISS, Baltimore (2005)
13. Hollanti, C., Lahtonen, J., Ranto, K., Vehkalahti, R.: On the Densest MIMO Lattices from Cyclic Division Algebras. IEEE Trans. Inform. Theory (submitted 2006). http://arxiv.org/abs/cs.IT/0703052
14. Albert, A.A.: Structure of Algebras. AMS, New York (1939)
15. Reiner, I.: Maximal Orders. Academic Press, New York (1975)
16. El Gamal, H., Hammons Jr., A.R.: A New Approach to Layered Space-Time Coding and Signal Processing. IEEE Trans. Inform. Theory 47, 2321–2334 (2001)

On the Computation of Non-uniform Input for List Decoding on Bezerra-Garcia Tower

M. Prem Laxman Das and Kripasindhu Sikdar

Indian Statistical Institute
203 B.T. Road, Kolkata 700108, West Bengal, India
prem_r@isical.ac.in

Abstract. Guruswami and Patthak, among many results, gave a randomized algorithm for computing the evaluation of regular functions of the Garcia-Stichtenoth tower at a large degree place. An algorithm, along the same lines, for Bezerra-Garcia tower is given. This algorithm uses Kummer theorem.

1 Introduction

Algebraic-geometric codes are evaluation codes similar to Reed-Solomon codes. These codes are constructed over function fields, F, of transcendence degree one over a finite field. For more details refer [1]. Such codes are well-studied for their asymptotic properties. In fact, codes constructed on the tower of function fields introduced in [2] attain best known bounds. Encoding and decoding procedures for linear codes constructed on function fields have attracted much research in the last two decades. The encoding procedure involves finding a basis for Riemann-Roch spaces of divisors. The functions of $L(uQ)$ are evaluated at some places of degree one to obtain the code.

A list decoding algorithm for a code gives as output a small list of codewords, but corrects more errors than a classical algorithm can. Such an algorithm for one-point codes was given in [3] and a suitable representation of the data involved was discussed in [4]. The algorithm is a interpolate and root-find strategy. For a received word $y = (y_1, \ldots, y_n)$ a polynomial in one variable over F is found, such that each coefficient lies in $L(D)$, where D is the underlying divisor and the zeroes of this polynomial are the required words. Then the zeroes of the interpolation polynomial are found and those which lie sufficiently close to the received word are output. The zeros of the interpolation polynomial are known to be elements of $L(D)$ for the underlying divisor D. This data may be used to design efficient root finding algorithms over function fields. Here the focus is on the root finding step of the list decoding algorithm. In [4], the root-find step involves computation of a non-uniform input, which is a evaluation of the basis elements of $L(D)$ at a large degree place. Hence, the non-uniform input is independent of the received word.

In [5] the authors, among many other results, find the non-uniform input for the function fields of the Garcia-Stichtenoth tower [2]. They use the structure of

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 237–246, 2007.
© Springer-Verlag Berlin Heidelberg 2007

the quasi-regular functions used in the pole cancellation algorithm of [6]. Their procedure for finding the non-uniform input is randomized, making uniformly random choices for irreducible polynomials of a given degree over $\mathbb{F}_{q^2}$. A simple counting argument shows that there exist places of degree r of F_m lying above places of same degree of F_1. Here $F_1 \subset F_2 \subset F_3 \subset \ldots$ is the tower. The required non-uniform input is obtained as a solution to a system of linearized equations, using Kummer theorem(see [1, pp. 76]).

A similar procedure for the Bezerra-Garcia tower is given here. There is a unique of $x_1 \in F_m$, which is totally ramified throughout the tower. For construction of codes, divisors of the form uP_∞ are chosen. A nice dual basis for the ring of such regular functions exist, such that it is sufficient of determine the evaluations of the coordinate variables at a large degree place to evaluate the basis elements themselves. There exist places of F_m of degree r lying above a place of same degree of F_1 for large enough r. Also the set $\{1, y, \ldots, y^{q-1}\}$ is a integral basis for the large degree place. The required evaluations of the coordinate variables are obtained by solving a system of linearized equations, using Kummer's theorem.

The plan of the paper is as follows. First some preliminaries on the Bezerra-Garcia tower from [7] are recalled. Then some facts on number of places of a given degree of a function field $F/\mathbb{F}_q$ of genus g are recalled from [1]. List decoding procedure for one point codes is recalled. A bound on the number of places of F_1 of degree r lying below a place of the same degree of F_m is obtained. Hence, the probability that a place of degree r of F_1 chosen at random having the above property is calculated. Finally, the randomized algorithm for finding the non-uniform input on the function fields of the Bezerra-Garcia tower is given.

2 Preliminaries and Notations

Throughout $\mathbb{F}_q$ will denote a finite field of cardinality q having characteristic p. We will be concerned with function fields of transcendence degree one $F/\mathbb{F}_q$. The genus of F will be denoted by g. Places of F will be denoted by P, Q, R, etc. The discrete valuation associated with a place P is denoted by v_P and the valuation ring by $\mathcal{O}_P$. The set of places of F will be denoted by $\mathbb{P}(F)$. For $z \in F$, the divisor (z) denotes the principal divisor of z. For D a divisor $\deg(D)$ and $\dim(D)$ will denote the degree and dimension of the divisor respectively. Recall that for D, a divisor

$$L(D) = \{z \in F \mid (z) + D \geq 0\}.$$

Also $\mathrm{supp}(D)$ denotes the support of D, which is the set of places appearing in the expression for D. Further for $P' \mid P$ in a separable extension of function fields $e(P' \mid P)$, $f(P' \mid P)$ and $d(P' \mid P)$ will denote respectively the ramification index, the relative degree and the different exponent respectively.

2.1 The Bezerra-Garcia Tower

In this section, the tower studied in [7] is recalled. Some important properties of this tower are listed.

Definition 1. *Let $K = \mathbb{F}_{q^2}$ and let $F_1 := K(x_1)$, be the rational function field. For each $m \geq 1$, we have $F_{m+1} := F_m(x_{m+1})$, where x_{m+1} satisfies*

$$\frac{x_{m+1} - 1}{x_{m+1}^q} = \frac{x_m^q - 1}{x_m}. \tag{1}$$

Following facts regarding ramification of places lying above the pole and zeroes of x_1 and $x_1 - 1$ of F_1 may be recalled from [7].

Lemma 1. *The following hold for the function field F_m of tower.*

a. *The unique pole of x_1 in F_1 is totally ramified throughout the tower.*
b. *The unique zero of x_1 in F_1 is totally ramified throughout the tower.*

Proof. See [7, Lemma 2]. □

By regular functions, we mean functions of F_m having poles only at $P_\infty^{(m)}$. Such functions form a subring of F_m, denoted by R_m. This ring is the integral closure of $R_1 = \mathbb{F}_{q^2}[x_1]$ in F_m.

For a separable extension of function fields, given any basis, there exists a uniquely determined trace dual basis. Next, we state a simple result regarding the existence of a nice (trace)basis-dual basis.

Theorem 1. *Let $\rho_i = (x_1 - 1)^{q^i}$ for $i = 2, \ldots, m$. Let*

$$\mathcal{Z} := \prod_{i=2}^{m} \{1, \rho_i x_i, \rho_i x_i^2, \ldots, \rho_i x_i^{q-1}\}$$

and

$$\mathcal{Z}^* := \prod_{i=2}^{m} \left\{ -\frac{x_i - 1}{\rho_i x_i^q}, -\frac{x_i - 1}{\rho_i x_i^{q-1}}, \ldots, -\frac{x_i - 1}{\rho_i x_i^2}, \frac{1}{\rho_i x_i} \right\}$$

be the sets obtaining by taking $m-1$-fold products of the constituent sets. Then

$$\sum_{z \in \mathcal{Z}} R_1 z \subseteq R_m \subseteq \sum_{z^* \in \mathcal{Z}^*} R_1 z^*,$$

where the sums above are finite.

Hence, we have the following corollary.

Corollary 1. *Any element $\zeta \in F_m$ having poles only at P_∞ can be written as a (finite) sum*

$$\zeta = \sum_{\xi \in \mathcal{Z}^*} a_\xi(x_1)\xi,$$

where a_ξ is a polynomial in x_1.

The denominator of the dual basis for F_m/F_1 above involves only $x_1 - 1$ and x_j's. We shall use this result for finding the non-uniform input for this tower. The above result uses the proof [1, Theorem III.5.10] and some simple facts about the tower. This result and many other facts about the tower are dealt with elsewhere. This tower is interesting because it attains the Drinfeld-Vlăduţ bound. In fact, in [7] it is shown that this tower is a subtower of that in [2].

Lemma 2. *The genus of the mth function field g_m is given by*

$$(q-1)\cdot g_m = \begin{cases} \left(q^{\frac{m}{2}}-1\right)^2, & m \text{ even} \\ \left(q^{\frac{m-1}{2}}-1\right)\left(q^{\frac{m+1}{2}}-1\right), & m \text{ odd.} \end{cases} \tag{2}$$

The rational places of F_1 corresponding to the roots of $x_1^q + x_1 - 1 = 0$ are completely splitting throughout the tower. Hence the number of rational places for F_m, denoted by N_m, satisfies

$$N_m \geq q^m. \tag{3}$$

Hence, the tower attains Drinfeld-Vlăduţ bound.

2.2 Number of Places of a Given Degree

Let $F/\mathbb{F}_q$ be a function field of genus g. Here, we recall estimates on the number of places of a given degree of a function field over a finite field. Basic reference for this topic is [1, Chapter V]. Let $N = N(F)$ denote the number of places of F of degree one. Also, let N_r denote the number of places of degree one in the constant field extension $F_r = F\mathbb{F}_{q^r}$ for $r \geq 1$. Further, let B_r denote the number of places of F of degree r. The bound on B_r from [1, Corollary V.2.10] is recalled.

Proposition 1. *The estimate*

$$|B_r - \frac{q^r}{r}| < (2+7g)\frac{q^{r/2}}{r}.$$

This bound will be used to obtain an estimate of the number of places of degree r of F_1 lying below places of same degree of F_m of the tower.

2.3 Algebraic-Geometric Codes and Their List Decoding

In this section list decoding algorithm of [3] is outlined. Let us first recall the definition of one-point algebraic-geometric codes on a function field. Basic reference for this topic is the monograph [1].

Definition 2. *Let $F \supset \mathbb{F}_q$ be a function field of genus g. Let $P_1, \ldots, P_n$ be distinct places of degree 1, all distinct from a place Q. Let $G = P_1 + \ldots + P_n$ and uQ. Let*

$$C_L(u, G) = \{(f(P_1), \ldots, f(P_n)) \mid f \in L(uQ)\} \subseteq \mathbb{F}_q^n.$$

The code C_L is known as a (One-point)Algebraic-Geometric(AG) code.

The next lemma gives the parameters of the one-point codes.

Lemma 3. *Assume that $u < n$. Then $C_L(u, G)$ is an $[n, k, d]_q$ code with $k \geq u - g + 1$ and $d \geq n - u$.*

It is assumed henceforth that $u < n$, so that the above lemma holds. List decoding algorithm for such one-point codes was given in [3] and a suitable representation of the data involved was discussed in [4]. Suppose that the channel corrupts at most $n - t$ places of the sent word and $y = (y_1, \ldots, y_n)$ is received. The list decoding algorithm of [3] finds an interpolation polynomial for y as the first step. This polynomial has degree s for a suitably chosen parameter s and has coefficients in $L(D)$ for a suitably chosen divisor D. For more details consult [3]. The required list of decoded words comprises of those zeroes of the interpolation polynomial in $L(uQ)$ whose evaluations at P_i agree with y_i for at least t coordinates.

In [4] the representation issues related to the list decoding algorithm are discussed. A strategy for finding the zeroes of the interpolation polynomial is given. This strategy is based on finding a non-uniform input which doesn't depend on the received word. A basis for $L(D)$ is assumed to be computable. The non-uniform input is described below:

Non-Uniform Input: A place R in $\mathbb{P}(F)$ of degree r greater than $\deg D$ represented as a l-tuple $(\zeta_1^R, \ldots, \zeta_l^R)$ over $\mathbb{F}_{q^r}$, obtained by evaluating a increasing basis $(\Phi_1, \ldots, \Phi_l)$ of $L(D)$ at place R.

Let us begin by recalling [4, Lemma 5].

Lemma 4. *If $f_1, f_2 \in L(A)$ for $A \succeq 0$ and $f_1(R) = f_2(R)$ for some place R of degree bigger than $\deg(A)$. Then $f_1 = f_2$.*

The strategy now is to first reduce the interpolation polynomial $H(T)$ modulo R to obtain $h(T)$ over the underlying finite field and find the zeroes of the polynomial equation $h(T) = 0$ using some standard algorithm. Then for each root compute $\beta_i \in L(D)$, if any, such that $\beta_i(R) = \alpha_i$. This β_i, by Lemma 4, is unique. Those elements of the list $\beta_1, \ldots, \beta_t$ are output which meet the distance criterion. The root-find procedure of [4] is given below.

Algorithm 1 *(ROOT-FIND)*
Input: A degree d polynomial $H(T) = \sum_{i=0}^{d} a_i T^i \in F[T]$, where each $a_i \in L(D)$.
Output: All zeroes of H that lie in $L(D)$

1. *Reduce H modulo a place $R \in \mathbb{F}$ of large enough degree, say r, to obtain $h(T)$.*
2. *Compute the zeroes, say $\alpha_1, \ldots, \alpha_t$ of $h(T)$ using a procedure for factorization of polynomials over finite fields.*
3. *For each α_i find the unique $\beta_i \in L(D)$, if any, which evaluates to α_i at R.*

The correctness of the algorithm hinges on the following remark.

Remark 1. If $\beta_i = \sum_{j=1}^{l(D)} a_j \Phi_j$, then

$$\sum_{j=1}^{l(D)} a_j \Phi_j(R) = \alpha_i$$

may be considered as a system of linear equations with $a_1, \ldots, a_{l(D)}$ as indeterminate over $\mathbb{F}_q$ after fixing a representation for $\mathbb{F}_{q^r} \supset \mathbb{F}_q$. This system has a unique solution by Lemma 4.

From the above discussion, it is clear that given

1. the non-uniform input,
2. a root-finding algorithm over a large finite field and
3. a procedure for solving a system of linear equations over $\mathbb{F}_q$

the root finding algorithm may be efficiently implemented. There exist algorithms to perform the second and third tasks above. Hence, given the non-uniform input the entire root-find step of the list decoding algorithm may be efficiently implemented.

In [5] the authors, among many other results, find the non-uniform input for the function fields of the Garcia-Stichtenoth tower [2]. Suppose

$$F_1 \subset F_2 \subset F_3 \subset \ldots$$

denotes the tower and $P_\infty^{(m)}$ the unique pole of x_1 in F_m. In [6] a pole cancellation based algorithm for determining a basis for $L(uP_\infty^{(m)})$ is given, which uses regular functions defined there. The procedure of [5] makes use of the structure of quasi-regular functions. A simple counting argument of [5] shows that there exist places of degree r of F_m lying above places of F_1 of same degree. Their procedure for finding the non-uniform input is randomized, making uniformly random choices for irreducible polynomials of a given degree over $\mathbb{F}_q$. The required non-uniform input is obtained as a solution of a system of linearized equations using Kummer's theorem(see [1, pg. 76]).

3 Places of a Special Type of Degree r of the Tower

We restrict our attention to function fields over finite fields of the type $\mathbb{F}_{q^2}$. A bound on the number of places of F_1 of degree r lying below a place of the same degree of F_m is obtained. Hence, the probability that a place of degree r of F_1 chosen at random having the above property is calculated. Techniques used in this section are from [1, Chapter V].

In the following the superscript m denotes the function field F_m of the tower. Thus $B_r^{(m)}$ denotes the number of places of degree r of $F_m/\mathbb{F}_{q^2}$.3

For F_m, let $U_r^{(m)}$ denote the number of places of places of F_1 of degree r lying below a degree r place of F_m. Let

$$B_{r,1}^{(m)} := \text{the number of degree } r \text{ places of } F_m \text{ lying above a degree } r \text{ place of } F_1 \text{ and}$$

$$B_{r,2}^{(m)} := \text{the number of degree } r \text{ places of } F_m \text{ not lying above a degree } r \text{ place of } F_1.$$

Clearly we have $B_r^{(m)} = B_{r,1}^{(m)} + B_{r,2}^{(m)}$. We have

$$B_{r,1}^{(m)} \leq U_r^{(m)} \cdot [F_m : F_1]. \tag{4}$$

Now, we shall estimate $B_{r,2}^{(m)}$. We know that places of degree r of F_1 are in one-to-one correspondence with monic irreducible polynomials of degree r over $\mathbb{F}_{q^2}$. Also, if $P' \mid P$ then $\deg(P)$ divides $\deg(P')$. Hence $B_{r,2}^{(m)}$ is utmost the number of monic irreducible polynomials of degree utmost $r/2$ over $\mathbb{F}_{q^2}$. Thus

$$\begin{aligned} B_{r,2}^{(m)} &\leq \sum_{d=1}^{r/2} \frac{q^{2d} - q^2}{d} \\ &\leq q^{r+1}. \end{aligned} \tag{5}$$

Next, we state and prove a simple lemma.

Lemma 5. *For $r \geq m + 16$ the following holds*

$$q^{m-1} \cdot U_r^{(m)} \geq \frac{q^{2r}}{2r}.$$

Proof. Using Equations 4 and 5 and the bound on B_r in Proposition 1, we obtain

$$q^{m-1} \cdot U_r^{(m)} \geq \frac{q^{2r}}{r} - \frac{8 g_m q^r}{r} - q^{r+1}.$$

Using the fact that $g_m \leq q^m$, we obtain

$$q^{m-1} \cdot U_r^{(m)} \geq \frac{q^{2r}}{r} - \frac{8 q^{r+m}}{r} - q^{r+1}.$$

Consequently, for $r \geq m + 16$ the following holds

$$q^{m-1} \cdot U_r^{(m)} \geq \frac{q^{2r}}{2r}.$$

hence, the result. □

Finally we estimate the probability with which a degree r place of F_1 chosen uniformly at random has a degree r place of F_m above it. Notice that choosing a degree r place of F_1 is equivalent to choosing an irreducible polynomial of degree r over $\mathbb{F}_{q^2}$. The following is a easy corollary to the above lemma.

Corollary 2. *Let the notations be as in the previous lemma. Let $r \geq m + 16$. Then $p_{r,m}$, the probability that a place of F_1 of degree r chosen uniformly at random lies below a degree r place of F_m, satisfies*

$$p_{r,m} \geq \frac{1}{2rq^{m+1}}.$$

Thus with non-zero probability a degree r place of F_1 chosen uniformly at random has a degree r place of F_m above it. We use this fact to construct a randomized algorithm for finding the non-uniform input in the next section.

4 Non-uniform Input on Bezerra-Garcia Tower

In this section, a randomized procedure for finding the required non-uniform input is given. A basis for the underlying vector space $\Phi_1, \ldots, \Phi_l$ is assumed to be given. The procedure of [5] applies for this tower too. The procedure, initially, makes a random choice of an irreducible polynomial. The required data is obtained as a solution of a system of linearized equations, by Kummer's theorem. It is been shown in the last section that there exist places of F_1 having a place of F_m of same degree above them. Thus the procedure must terminate in expected polynomial time in the length of the code.

Recall that one-point codes are constructed by evaluating elements of a suitable Riemann-Roch space at places of degree one. For the Bezerra-Garcia tower, since the unique pole of x_1 is totally ramified throughout the tower, for each level, a divisor $D_m = u_m P_\infty^{(m)}$ is chosen. There are at least q^m places of degree one for F_m, not lying above zeroes and poles of $x_1(x_1 - 1)$. The code is obtained by evaluating elements of $L(u_m P_\infty^{(m)})$ at these q^m places. The sequence of codes thus obtained have asymptotically best properties.

There exist algorithms for finding a basis for the ring of regular functions on Garcia-Stichtenoth tower. See [6] for example. But such an explicit algorithm doesn't exist for the Bezerra-Garcia tower. So, the entire exercise assumes that a basis for the underlying vector space is given. The non-uniform input is calculated by evaluating these basis elements at a high degree place. However, the result in Lemma 1 guarantees that the non-uniform input may be effectively computed.

Recall that, list decoding one-point codes uses a non-uniform input for the root-finding step. Let r be chosen such that both:

(a). $r > u_m$ and
(b). $r \geq m + 16$

hold. A place of F_m of degree r may be constructed as follows. Places of degree r of F_1 are in one-to-one correspondence with monic irreducible polynomials of degree r over $\mathbb{F}_{q^2}$. Such a polynomial is chosen at random. Denote the place determined by this polynomial by ρ_1. Let $\gamma_2 = \left(\frac{x_1}{x_1^q - 1}\right)(\rho_1)$. Consider the system of linearized equations.

$$\begin{aligned} x_2^q - \frac{x_1}{x_1^q - 1} x_2 &= -\gamma_2 \\ x_3^q + \frac{x_2}{x_2^q - 1} x_3 &= -\frac{x_2}{x_2^q - 1} \\ &\vdots \\ x_m^q + \frac{x_{m-1}}{x_{m-1}^q - 1} x_m &= -\frac{x_{m-1}}{x_{m-1}^q - 1} \end{aligned} \tag{6}$$

A solution to this system gives a place of degree r, by Kummer's theorem(refer [1, pp. 76]). We first state the algorithm for finding the non-uniform input and then prove its correctness.

Algorithm 2 *(Non-uniform input)*
Input: m, r *and* $\Phi_1, \ldots, \Phi_l$
Output: $(\alpha_1, \ldots, \alpha_m)$

A. *Choose an irreducible polynomial* f *of degree* r *over* $\mathbb{F}_{q^2}$. *Let* ρ_1 *denote the place of* F_1 *with uniformizing parameter* $f(x_1)$.
B. *Set* $\alpha_1 = x_1(\rho_1)$ *and* $\gamma_2 = \left(\frac{x_1}{x_1^q - 1}\right)(\rho_1)$. *Find a solution of the system of Equations 6, say* $(\alpha_2, \ldots, \alpha_m)$.
C. *If a solution exists compute the evaluations of* $\Phi_1, \ldots, \Phi_l$ *at this place using* $(\alpha_1, \ldots, \alpha_m)$, *else report failure.*

Notice that only the choice of irreducible polynomial is random. Rest of the steps in the computation of the non-uniform input are deterministic. Thus with probability $p(r, m)$ the algorithm outputs the non-uniform input. The rest of the steps of the list decoding algorithm may be carried out efficiently once the non-uniform input is given, as discussed earlier. We start the proof of correctness of this algorithm with a simple technical lemma.

Lemma 6. *Let* P_j *and* P_{j-1} *be places of* F_j *and* F_{j-1} *with* $P_j \mid P_{j-1}$ *not lying above zeroes and poles of* $x_1(x_1 - 1) \in F_1$. *The set* $\{1, x_j, \ldots, x_j^{q-1}\}$ *is an integral basis for* $F_j/F_{j-1},\ j \geq 2$ *at* $P_j \mid P_{j-1}$.

Proof. By [1, Theorem III.5.10], the set $\{1, x_j, \ldots, x_j^{q-1}\}$ is an integral basis for $P_j \mid P_{j-1}$ if and only if $d(P_j \mid P_{j-1}) = v_{P_j}(\phi_j'(y))$. Here ϕ' denotes the formal derivative. We have

$$\begin{aligned} v_{P_j}(\phi_j'(y)) &= v_{P_j}\left(\frac{x_{j-1}}{x_{j-1}^q - 1}\right) \\ &= 0. \end{aligned}$$

By [7, Lemma 2], we have $P_j \mid P_{j-1}$ is unramified. Thus

$$d(P_j \mid P_{j-1}) = e(P_j \mid P_{j-1}) - 1 = 0,$$

by Dedekind's different theorem([1, Theorem III.5.1]). Thus $\{1, x_j, \ldots, x_j^{q-1}\}$ is an integral basis for the extension $F_j/F_{j-1},\ j \geq 2$ at $P_j \mid P_{j-1}$. □

We are now in a position to give the proof of correctness of the above algorithm.

Theorem 2. *The Algorithm 2 gives the required non-uniform input.*

Proof. For any level, we have shown that set $\{1, x_j, \ldots, x_j^{q-1}\}$ is an integral basis for $F_j/F_{j-1},\ j \geq 2$ at $P_j \mid P_{j-1}$. Notice that all the conditions of Kummer theorem are satisfied. The first equation of the system is the reduced form the defining equation. Also, if a solution to the system of linearized equations exists, then $(\alpha_1, \ldots, \alpha_m)$ is the evaluation of the coordinate variables at a degree r place of F_m. By Lemma 1, the basis elements may be evaluated using this tuple $(\alpha_1, \ldots, \alpha_m)$, since the denominator of the dual basis involves only $x_1 - 1$ and the x_j's. Hence the correctness of the algorithm is verified. □

Complexity: The main computational tasks involved in the procedure are the following:

1. checking whether a given polynomial is irreducible or not and
2. finding a solution to a system of linear equations.

There exist deterministic algorithms for performing both the tasks. Also, the procedure gives the required non-uniform input in expected polynomial time in the length of the code.

References

1. Stichtenoth, H.: Algebraic Function Fields and Codes. In: Universitext, Springer, Heidelberg (1993)
2. Garcia, A., Stichtenoth, H.: On the Asymptotic Behaviour of Some Towers of Function Fields over Finite Fields. Journal of Number Theory 61(2), 248–273 (1996)
3. Guruswami, V., Sudan, M.: Improved Decoding of Reed-Solomon and Algebraic-Feometric Codes. IEEE Trans. Inform. Theory 45(6), 1757–1767 (1999)
4. Guruswami, V., Sudan, M.: On Representations of Algebraic-Geometric Codes. IEEE Trans. on Inform. Theory 47(4), 1610–1613 (2001)
5. Guruswami, V., Patthak, A.: Correlated Algebraic-Geometric Codes: Improved List Decoding Over Bounded Alphabets. Mathematics of Computation (to appear)
6. Shum, K., Aleshnikov, I., Kumar, P.V., Stichtenoth, H., Deolalikar, V.: A Low-Complexity Algorithm for the Construction of Algebraic-Geometric Codes Better Than the GIlbert-VArshamov Bound. IEEE Trans. on Inform. Theory 47(6), 2225–2241 (2001)
7. Bezerra, J., Garcia, A.: A Tower with Non-GAlois Steps Which Attains the DRinfeld-VLadut Bound. Journal of Number Theory 106(1), 142–154 (2004)

Dense MIMO Matrix Lattices — A Meeting Point for Class Field Theory and Invariant Theory

Jyrki Lahtonen[1] and Roope Vehkalahti[2]

[1] University of Turku, Department of Mathematics, Finland, and Nokia Research Center, Radio Communications Lab
[2] University of Turku, Department of Mathematics, Finland and Turku Graduate School in Computer Science

Abstract. The design of signal constellations for multi-antenna radio communications naturally leads to the problem of finding lattices of square complex matrices with a fixed minimum squared determinant. Since [5] cyclic division algebras, their orders and related structures have become standard material for researchers seeking to construct good MIMO-lattices. In recent submissions [3], [8] we studied the problem of identifying those cyclic division algebras that have the densest possible maximal orders. That approach was based on the machinery of Hasse invariants from class field theory for classifying the cyclic division algebras. Here we will recap the resulting lower bound from [3], preview the elementary upper bounds from [4] and compare these with some suggested constructions. As the lattices of the shape E_8 are known to be the densest (with respect to the usual Euclidean metric) in an 8-dimensional space it is natural to take a closer look at lattices of 2x2 complex matrices of that shape. We derive a much tighter upper bound to the minimum determinant of such lattices using the theory of invariants.

1 Background

In the symmetric MIMO-case the received signal is

$$Y_{n\times n} = H_{n\times n}X_{n\times n} + N_{n\times n},$$

where H is the Rayleigh fading channel response, the elements of the noise matrix N are i.i.d. complex Gaussian random variables. Here n is the number of both transmitting and receiving antennas (= the symmetric case) and it is often assumed that the receiver knows the channel matrix H. An analysis of this situation gives rise to the so called determinant criterion: the receiver's ability to distinguish between signals X and X' is the better the larger the determinant of the matrix $(X - X')(X^H - X'^H)$. Thus a natural choice for a set of signals is a finite collection of low-energy matrices X within a lattice $\mathbf{L} \subset \mathcal{M}_n(\mathbf{C})$ with a large minimum determinant $\det_{min}(\mathbf{L})$, i.e. the infimum of the absolute values of the determinants of all non-zero matrices in $\mathbf{L}$. In this

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 247–256, 2007.
© Springer-Verlag Berlin Heidelberg 2007

note we restrict ourselves to lattices of maximal rank $k = 2n^2$. We refer the interested reader to [7] for some adaptations of this theory to the more general asymmetric case. Following [9] we shall insist on the non-vanishing determinant property $\det_{min}(\mathbf{L}) > 0$ guaranteeing that the resulting lattices achieve the diversity-multiplexing tradeoff bound from [10].

The measure, or hypervolume, $m(\mathbf{L})$ of the fundamental parallelotope of the lattice $\mathbf{L}$ is the reciprocal of its center density. Any lattice $\mathbf{L}$ can be scaled to satisfy $m(\mathbf{L}) = 1$. This gives rise to a natural numerical measure for the quality of a lattice. We shall denote by $\delta(\mathbf{L})$ the *normalized minimum determinant* of the lattice $\mathbf{L}$, i.e. here we first scale $\mathbf{L}$ to have a unit size fundamental parallelotope.

Definition 1. *Fix the index n of a MIMO-lattice. Let us define the optimal minimum determinant by*

$$\delta(n) = \sup_{\mathbf{L}} \delta(\mathbf{L}),$$

where the supremum is taken over the set of full rank lattices inside $\mathcal{M}_n(\mathbf{C})$ normalized to unit fundamental parallelotope.

It is worth emphasizing that we do not seek to optimize the minimum Euclidean distance of a lattice. The accumulated knowledge on that problem (cf. [1]) will come into use, but e.g. it isn't at all obvious, whether a lattice of rank 8 should be isometric to the root lattice E_8 in order to be optimal with respect to our criterion.

2 Cyclic Algebras and Orders

We refer the interested reader to [12] and [5] for an exposition of the theory of simple algebras, cyclic algebras, and their use in ST-coding. We only recall the basic definitions and notations here. Consider an extension E/F of number fields. In the interesting cases F is an imaginary quadratic field, usually either $\mathbf{Q}(i)$ or $\mathbf{Q}(\sqrt{-3})$. We assume that E/F is a cyclic field extension of degree n with Galois group $Gal(E/F) = \langle\sigma\rangle$. Let $\mathcal{A} = (E/F, \sigma, \gamma)$ be the corresponding cyclic algebra of index n:

$$\mathcal{A} = E \oplus uE \oplus u^2E \oplus \cdots \oplus u^{n-1}E.$$

Here $u \in \mathcal{A}$ is an auxiliary generating element subject to the relations $xu = u\sigma(x)$ for all $x \in E$ and $u^n = \gamma \in F^*$. An element $a = x_0 + ux_1 + \cdots + u^{n-1}x_{n-1} \in \mathcal{A}$ has the following representation as a matrix

$$a \mapsto \begin{pmatrix} x_0 & \sigma(x_{n-1}) & \sigma^2(x_{n-2}) & \cdots & \sigma^{n-1}(x_1) \\ \gamma x_1 & \sigma(x_0) & \sigma^2(x_{n-1}) & & \sigma^{n-1}(x_2) \\ \gamma x_2 & \gamma\sigma(x_1) & \sigma^2(x_0) & & \sigma^{n-1}(x_3) \\ \vdots & & & & \vdots \\ \gamma x_{n-1} & \gamma\sigma(x_{n-2}) & \gamma\sigma^2(x_{n-3}) & \cdots & \sigma^{n-1}(x_0) \end{pmatrix}.$$

The determinant (resp. trace) of the matrix A above is called the *reduced norm* (resp. *reduced trace*) of the element $a \in \mathcal{A}$ and is denoted by $nr(a)$ (resp. $tr(a)$).

Let R denote a Noetherian integral domain with a quotient field F, and let $\mathcal{A}$ be a finite dimensional F-algebra.

Definition 2. *An R-order in the F-algebra $\mathcal{A}$ is a subring Λ of $\mathcal{A}$, having the same identity element as $\mathcal{A}$, and such that Λ is a finitely generated module over R and generates $\mathcal{A}$ as a linear space over F. An order Λ is called maximal, if it isn't properly contained in another R-order.*

Example 1. In any cyclic division algebra we can always choose the element $\gamma \in F^*$ determining the 2-cocycle in $H^2(E/F)$ to be an algebraic integer. We immediately see that the $\mathcal{O}_F$-module

$$\Lambda = \mathcal{O}_E \oplus u\mathcal{O}_E \oplus \cdots \oplus u^{n-1}\mathcal{O}_E,$$

where $\mathcal{O}_E$ is the ring of integers, is an $\mathcal{O}_F$-order in the cyclic algebra $(E/F, \sigma, \gamma)$. We refer to this $\mathcal{O}_F$-order as the *natural order*.

For the purposes of constructing MIMO lattices the reason for concentrating on orders is summarized in the following proposition (e.g. [11, Theorem 10.1, p. 125]). We simply rephrase it here in the language of MIMO-lattices.

Proposition 1. *Let Λ be an order in a cyclic division algebra $(E/F, \sigma, \gamma)$. Then for any non-zero element $a \in \Lambda$ its reduced norm $nr(a)$ is a non-zero element of the ring of integers $\mathcal{O}_F$ of the center F. In particular, if F is an imaginary quadratic number field, then the minimum determinant of the lattice Λ is equal to one.*

Definition 3. *Let $m = dim_F\mathcal{A}$. The discriminant of the R-order Λ is the ideal $d(\Lambda/R)$ in R generated by the set*

$$\{det(tr(x_i x_j))_{i,j=1}^m \mid (x_1, ..., x_m) \in \Lambda^m\}.$$

An important fact is that all the maximal orders of a given cyclic division algebra have the same discriminant [11, Theorem 25.3]. The definition of the discriminant closely resembles that of the Gram matrix of a lattice, so the following results are unsurprising and probably well-known. We include them for easy reference. Sample proofs are given in [3].

Lemma 1. *Assume that F is an imaginary quadratic number field and that* 1 *and θ form a $\mathbf{Z}$-basis of its ring of integers R. Assume further that the order Λ is a free R-module (an assumption automatically satisfied, when R is a principal ideal domain). Then the measure of the fundamental parallelotope equals*

$$m(\Lambda) = |\Im\theta|^{n^2} |d(\Lambda/R)|.$$

Corollary 1. *Let $F = \mathbf{Q}(i), R = \mathbf{Z}[i]$, and assume that $\Lambda \subset (E/F, \sigma, \gamma)$ is an R-order. Then the determinant of the Gram matrix of the matrix representation of Λ is*

$$\det(G(\Lambda)) = |d(\Lambda/\mathbf{Z}[i])|^2,$$

and the normalized minimum determinant is thus $\delta(\Lambda) = 1/|d(\Lambda/\mathbf{Z}[i])|^{1/2n}$.

Corollary 2. *Let $\omega = (-1+\sqrt{-3})/2, F = \mathbf{Q}(\sqrt{-3}), R = \mathbf{Z}[\omega]$, and assume that $\Lambda \subset (E/F, \sigma, \gamma)$ is an R-order. Then the determinant of the Gram matrix of the matrix representation of Λ is*

$$\det(G(\Lambda)) = (3/4)^{n^2} |d(\Lambda/\mathbf{Z}[\omega])|^2,$$

and the normalized minimum determinant is $\delta(\Lambda) = (2/\sqrt{3})^{n/2}/|d(\Lambda/\mathbf{Z}[i])|^{1/2n}$.

So in both cases maximizing the density of the code is equivalent to minimizing the discriminant. From Proposition 1 we also get that the minimum determinants of any orders in any cyclic division algebra with center either $\mathbf{Q}(i)$ or $\mathbf{Q}(\sqrt{-3})$ are equal to one. Thus in order to maximize the normalized minimum determinant of a lattice we should use a maximal order. Furthermore, we need to look for division algebras that have a maximal order with as small a discriminant as possible. A point worth emphasizing is that using ideals of any order doesn't appear to improve the situation. This is because a cyclic submodule of any order shares the same normalized density with the 'mother code' of the maximal order. Also, when the center of the cyclic division algebra is either $\mathbf{Q}(i)$ or $\mathbf{Q}(\sqrt{-3}$ then Eichler's theorem [11, Theorem 34.9] says that all the one-sided ideals of a maximal order actually are cyclic. The use of ideals may change the shape of the lattice, and this was a point exploited in [6].

Example 2. In the case of the Golden algebra $(\mathbf{Q}(i, \sqrt{5})/\mathbf{Q}(i), \sigma, i)$ from [6] the natural order turns out to be maximal. As the ring of algebraic integers of $\mathbf{Q}(i, \sqrt{5})$ has basis $\{1, i, (1 + \sqrt{5})/2, i(1 + \sqrt{5})/2\}$ we can quickly compute that the discriminant of the Golden algebra is 25. We thus recover from Corollary 1 the fact [2] that the normalized minimum determinant of the Golden code is $\delta = 1/\sqrt{5}$.

3 Maximal Orders with Minimal Discriminants and Bounds on the Normalized Density

Let F be an algebraic number field that is finite dimensional over $\mathbf{Q}$, $\mathcal{O}_F$ its ring of integers. Let us next recall the Main Theorem of [3]. The proof therein uses the formula for the local discriminants for maximal orders in terms of the Hasse invariants, the result of global class field theory that the Hasse invariants must sum up to an integer, and some simple estimates. The relevant theoretical background is contained in e.g. [11] and [13].

Theorem 1 (Discriminant bound). *Assume that F is a totally complex number field, and that P_1 and P_2 are the two smallest prime ideals in $\mathcal{O}_F$. Then the smallest possible discriminant of all central division algebras over F of index n is $(P_1P_2)^{n(n-1)}$.*

For us the importance of this result is twofold. It proves the existence of fully multiplexing MIMO-lattices with a known normalized density and/or minimum determinant. It also proves that using orders of cyclic division algebras (and their cyclic submodules) one cannot do any better. The latter point was the upshot of [3] but here we benefit from the first point.

In the interesting cases $F = \mathbf{Q}(i)$ and $F = \mathbf{Q}(\sqrt{-3})$ Theorem 1 gives us the following two corollaries. They are directly from [3], but we have partially reformulated them in terms of the normalized minimum determinants. For the purposes of finding the optimal normalized minimum determinant the field $\mathbf{Q}(i)$ is not nearly as interesting as the denser $\mathbf{Q}(\sqrt{-3})$. We list the gaussian results here for reference, as the rectangular shapes enjoy certain practical advantages in radio communications. It is also worth remarking that the assumption about the center in Theorem 1 is essential. Indeed, the quaternionic division algebra with the real quadratic center $F = \mathbf{Q}(\sqrt{5})$ has the well known ring of icosians as a maximal order with unit discriminant. The difference comes from the fact that in this case the only non-trivial Hasse invariants are at the two infinite places, and they won't contribute to the discriminant.

Corollary 3 (Discriminant bound). *Let Λ be an order of a central division algebra of index n over the field $\mathbf{Q}(i)$. Then the normalized minimum determinant of the resulting lattice satisfies the inequality*

$$\delta(\Lambda) \leq 1/10^{(n-1)/4}.$$

Furthermore, there exist cyclic division algebras with center $\mathbf{Q}(i)$, whose maximal orders achieve this bound.

Corollary 4 (Discriminant bound). *Let Λ be an order of a central division algebra of index n over the field $\mathbf{Q}(\omega)$, $\omega = (-1+\sqrt{-3})/2$. Then the normalized minimum determinants of the lattice satisfies the inequality*

$$\delta(\Lambda) \leq (2/\sqrt{3})^{n/2}/12^{(n-1)/4}.$$

Furthermore, there exist cyclic division algebras with center $\mathbf{Q}(\omega)$, whose maximal orders achieve this bound.

The construction of algebras achieving the bounds in the two previous corollaries is done in [8]. These results can be viewed as giving a lower bound on the achievable normalized minimum determinant $\delta(n)$. Can we get upper bounds on the achievable normalized minimum determinant also? In general this is probably a difficult problem, but the following simple upper bounds from [4] are elementary to derive.

Lemma 2 (Hadamard bound). *Let A be an $n \times n$ complex matrix. Write $\|A\| = \sqrt{trA^H A}$ for its Frobenius-norm. We then have the inequality $|\det A| \leq \|A\|^n / n^{n/2}$.*

Proof. Let $A_j, j = 1, 2, \ldots, n$ be the rows of A. By the Hadamard inequality

$$|\det A| \leq \prod_{j=1}^{n} \|A_j\|.$$

Squaring this inequality and using the fact that $\|A\|^2 = \sum_{j=1}^{n} \|A_j\|^2$ together with the well-known inequality between the geometric and arithmetic means of positive numbers gives the claimed bound.

Proposition 2 (Hadamard bound). *For fully multiplexing 2×2 lattices we have the upper bound $\delta(2) \leq 1$.*

Proof. The root lattice E_8 has the best minimum distance among 8-dimensional lattices (cf. e.g. [1]). When we scale its fundamental parallelotope to have unit measure, the shortest vectors have length $\sqrt{2}$. In other words in any lattice $\mathbf{L}$ of rank 8 inside $\mathcal{M}_2(\mathbf{C})$ has a non-zero matrix A with $\|A\| \leq \sqrt{2}$. Lemma 2 then tells us that $|\det A| \leq 1$.

Proposition 3 (Rectangular Hadamard bound). *For any rectangular lattice $\mathbf{L} \subset \mathcal{M}_n(\mathbf{C})$*

$$\delta(\mathbf{L}) \leq \frac{1}{n^{n/2}}.$$

Proof. When a rectangular lattice has a fundamental parallelotope of unit measure, at least one of the vectors in an orthogonal basis has length at most 1. The determinant of such a matrix is at most $1/n^{n/2}$ by Lemma 2.

In order to get an idea how strong these bounds are let us consider the case of 2×2 lattices. The Golden code has $\delta = 1/\sqrt{5} = 0.4472$. By the bound of Proposition 3 any rectangular or hypercubical lattice cannot have normalized minimum determinant > 0.5, so in this sense the Golden code is very good. It would not surprise us, if the Golden code turned out to have the highest possible normalized minimum determinant among rectangular lattices.

On the other hand the non-rectangular lattice from the next section attains the bound of Corollary 3 and thus has normalized minimum determinant $\delta = 10^{-1/4} = 0.5623$. In light of this we might conclude that rectangular lattices have no hope of achieving the density of the lattice from the next section. In the same vein the hexagonal lattice in the next section achieves the bound of Corollary 4 and has a normalized minimum determinant $\delta = \sqrt{2}/3^{3/4} = 0.6204$. Somewhat surprisingly the resulting hexagonal 2×2 MIMO-codes outperform their rectangular cousins only at high data rates such as 8 bits per channel use.

All these constructions are still somewhat distant from the bound of Proposition 2. We believe that the actual value of $\delta(2)$ is most likely less than one, but dare not guess the exact value of $\delta(2)$.

In Table 1. we compare the normalized minimum determinant of the perfect lattices from [6] to the bounds of Proposition 3 and Proposition 4 and also the upper bounds of Proposition 3. The bound of Proposition 2 will be generalized for higher n in [4], but the general bound is more difficult to compute as it is expressed in terms of known bounds [1] for the minimum Euclidean distance of high dimensional lattices. Such a bound is listed in the last column. A quick summary of this table might be that the perfect codes are good within their constrained class of rectangular (hexagonal) lattices, but all the constructions are quite far away from the bound of Lemma 2. Most likely the simple bound is not tight at all.

Table 1. Normalized minimum determinant δ of selected lattices and bounds

n	Perfect	CDA/$\mathbf{Q}(i)$	CDA/$\mathbf{Q}(\omega)$	Rectangular bound	simple bound
2	0.447	0.562	0.620	0.500	1.00
3	0.143	0.316	0.358	0.192	1.16
4	0.0298	0.178	0.207	0.0625	1.61
5		0.100	0.119	0.0179	2.57
6	0.00255	0.0562	0.0689	0.00463	4.59

4 Dense Example Lattices

In [3] it is shown that the following cyclic division algebra achieves the bound of Corollary 3. Let λ be the square root of the complex number $2+i$ belonging to the first quadrant of the complex plane. Then the cyclic algebra $\mathcal{GA}+ = (\mathbf{Q}(\lambda)/\mathbf{Q}(i), \sigma, i)$, where the automorphism σ is determined by $\sigma(\lambda) = -\lambda$, is a division algebra.

In order to give a concrete description of a maximal order within $\mathcal{GA}+$ we describe it in terms of its $\mathbf{Z}[i]$-basis. A maximal order Λ consists of the matrices $aM_1 + bM_2 + cM_3 + dM_4$, where a, b, c, d are arbitrary Gaussian integers, M_1 is the identity matrix, and $M_i, i = 2, 3, 4$ are the following matrices.

$$M_2 = \begin{pmatrix} 0 & 1 \\ i & 0 \end{pmatrix}, \quad M_3 = \frac{1}{2}\begin{pmatrix} i+i\lambda & i-\lambda \\ -1+i\lambda & i-i\lambda \end{pmatrix}, \quad M_4 = \frac{1}{2}\begin{pmatrix} -1-i\lambda & i+i\lambda \\ -1+\lambda & -1+i\lambda \end{pmatrix}.$$

It is then straightforward to verify that the fundamental parallelotope of this lattice has measure 10, and thus $\delta(\Lambda) = 10^{-1/4}$.

Let $z = 3^{1/4}(1+i)/\sqrt{2}$ be the prescribed fourth root of -3, and

$$\rho = \frac{1}{2}\begin{pmatrix} 1+z & \omega(1+z) \\ (1+\omega)(1-z) & 1-z \end{pmatrix}.$$

From [3] we also get that the cyclic algebra constructed from the datum $E = \mathbf{Q}(z)$, $F = \mathbf{Q}(\omega)$, $\sigma(z) = -z$, $\gamma = -\omega$ is a division algebra whose maximal

orders achieve the bound $\delta = \sqrt{2}/3^{3/4}$ of Corollary 4. We are indebted to Camilla Hollanti for the extra piece of information that the set $\{1, \rho, z, z\rho, \omega, \omega\rho, \omega z, \omega z\rho\}$ forms a $\mathbf{Z}$-basis of one such maximal order. Do observe that in this listing, as is always the case with the elements of the maximal subfield E, the element z must be viewed as the diagonal matrix with entries $z, \sigma(z) = -z$.

5 A Sharper Bound for 2×2 Lattices with Shape E_8

Throughout this section we assume that $\mathbf{L}$ is a rank 8 lattice of 2×2 complex matrices. We identify such matrices with vectors of $\mathbf{R}^8$ via the natural mapping

$$f : (x_1, x_2, \ldots, x_8) \mapsto \begin{pmatrix} x_1 + ix_2 & x_3 + ix_4 \\ x_5 + ix_6 & x_7 + ix_8 \end{pmatrix}.$$

This mapping is an isometry with respect to the Euclidean norm of $\mathbf{R}^8$ and the Frobenius norm of complex matrices. Let us denote by $S(r)$ the sphere of radius r in the 8-dimensional space. Whenever convenient we identify it with its image in the matrix space. We shall be interested in the polynomial function

$$p(x_1, x_2, \ldots, x_8) = |\det(f(x_1, x_2, \ldots, x_8))|^2 .$$

Its space-consuming exact form doesn't interest us but we do observe that the polynomial

$$p(x_1, x_2, \ldots, x_8) - (x_1^2 + x_2^2)(x_7^2 + x_8^2) - (x_3^2 + x_4^2)(x_5^2 + x_6^2)$$

only contains terms that are products of 4 distinct coordinates x_i. Our immediate goal is to determine the average value of the polynomial p on the sphere $S(r)$.

It is well known (cf. e.g. [1]) that the lattice E_8 can be constructed as the set of vectors $\mathbf{x} = (x_1, x_2, \ldots, x_8) \in \mathbf{Z}^8$ such that after reduction mod 2 it becomes a word of the self-dual extended Hamming code of length 8, in other words

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix} \bar{\mathbf{x}}^T = 0.$$

This version of E_8 has minimum Euclidean distance 2, $m(E_8) = 16$ and it has 240 vector of minimal length 2: There are 16 vectors with a single ± 2 component together with seven zeros, and $14 \cdot 16 = 224$ vectors of 4 ± 1s and 4 zeros congruent (modulo 2) to one of the 14 words of weight 4 — every such word gives rise to 16 short vectors differing from each other by the combination of signs. We can scale E_8 to have any desired value $r > 0$ as the length of the shortest vectors, and we already noted that scaling $r = \sqrt{2}$ leads to a normalized version of E_8 in the sense that $m(E_8, r = \sqrt{2}) = 1$.

Let us review the concept of a spherical t-design. A finite set X on a sphere (centered at the origin) $S \subseteq \mathbf{R}^n$ is called a t-design, if for any polynomial

$q(x_1, x_2, \ldots, x_n) \in \mathbf{R}[x_1, \ldots., x_n]$ of degree at most t the average of the values attained by q on the set X equals the average of the values on all of S (with respect to the usual measure on S). From [1, p. 90] we pick up the crucial fact that the set of 240 shortest vectors in E_8 form a spherical 7-design $X(E_8)$. This is a consequence of a general result due to B. Venkov. The argument depends on the fact that the lattice E_8 has a very large group G of symmetries (the Weyl group of type E_8 from the theory of Lie algebras), and the fact that the only polynomial invariants of group G of degree less than 8 are polynomials of the squared Euclidean norm $x_1^2 + \cdots x_8^2$.

The polynomial $p(x_1, x_2, \ldots, x_8)$ has degree 4 so, in particular, we can compute the average value of p on any $S(r)$ by computing the same average value on an appropriately scaled version of $X(E_8)$. Furthermore, p is homogeneous of degree 4, so its average on $S(r)$ will be proportional to r^4. As any rotated version of $X(E_8)$ will do just as well, we can set $r = 2$ and do our calculations with the version of E_8 described above. It helps us to use the symmetries of the sphere: obviously the expected value of any monomial of the form $x_i x_j x_k x_\ell$, $i < j < k < \ell$ will be equal to zero as the sphere is symmetric under the mapping $x_i \mapsto -x_i$. Similarly any permutation of the coordinates will be an isometry of the sphere, so the averages of all the monomials $x_i^2 x_j^2, i < j$ will be equal to the average of $x_1^2 x_2^2$. There are exactly 3 words of weight 4 in the extended Hamming code that have non-zero components at both the first and the second position: 11110000, 11001100 and 11000011. Therefore the monomial $x_1^2 x_2^2$ assumes the value 1 at $3 \cdot 16 = 48$ points on $X(E_8)$ and the value zero at the remaining points. Hence the average value of $x_1^2 x_2^2$ on $S(r = 2)$ is $1/5 = r^4/80$. The polynomial p has altogether 8 monomial terms of the form $x_i^2 x_j^2$, so we have proven the following.

Theorem 2. *The average value of the squared absolute value of the determinant $p(x_1, x_2, \ldots, x_8)$ on the sphere $S(r)$ equals $r^4/10$. The same result holds for any rotated and scaled copy of the collection of 240 shortest non-zero vectors of the lattice E_8.*

Corollary 5. *The normalized minimum determinant of any rank 8 MIMO-lattice $\mathbf{L}$ of 2×2-matrices and shape E_8 is bounded from above by $\delta(\mathbf{L}) \leq \sqrt{2/5}$.*

Proof. Set $r = \sqrt{2}$ to achieve normalization. The minimum squared determinant on any rotated version of $X(E_8)$ cannot be higher than the average squared determinant.

A couple of closing remarks are due. The restricted upper bound $\sqrt{2/5} = 0.6325$ is suggestively close to the lower bound 0.6204 of Corollary 4. Thus in order to make significant improvement to that lower bound shapes other than E_8 are forced upon us. Of course, there are no guarantees that even that would help, and the very restricted upper bound of Corollary 5 may apply to a much larger set of MIMO-lattices.

The somewhat trivial averaging nature of the argument leading to Corollary 5 immediately asks the question of how tight is that bound. We know of no lattice

with shape E_8 that would have $\sqrt{2/5}$ as its normalized minimum determinant. The best known lattice with shape E_8 is a sublattice of index 64 in the Golden code (cf. [14]), but being a cyclic submodule of the Golden algebra that lattice shares the normalized minimum determinant of $1/\sqrt{5}$. On the other hand we strongly believe that there are lattices of shape E_8 that achieve the bound of Corollary 5 on the 'first layer' of the shortest 240 vectors. Before we discovered a proof for Corollary 5 we set up a computer search based on simulated annealing. The program found a copy of the first layer of E_8, where the squared minimum determinant was larger than 0.399. Thus the squared minimum determinant of $2/5$ will likely be achieved on the first layer.

References

1. Conway, J.H., Sloane, N.J.A.: Sphere Packings, Lattices and Groups. Springer, New York (1988)
2. Belfiore, J.-C., Rekaya, G., Viterbo, E.: The Golden Code: A 2x2 Full-Rate Space-Time Code With Non-vanishing Determinant. IEEE Trans. Inform. Theory 51(4), 1432–1436 (2005)
3. Hollanti, C., Lahtonen, J., Ranto, K., Vehkalahtid, R.: On the Densest MIMO Lattices from Cyclic Division Algebras, `http://arxiv.org/abs/cs/0703052`
4. Vehkalahti, R., Lahtonen, J.: Bounds on the Density of MIMO-lattices (in preparation)
5. Sethuraman, B.A., Rajan, B.S., Shashidhar, V.: Full-Diversity, High-Rate Space-Time Block Codes From Division Algebras. IEEE Trans. Inform. Theory 49, 2596–2616 (2003)
6. Belfiore, J.-C., Oggier, F., Rekaya, G., Viterbo, E.: Perfect Space-Time Block Codes. IEEE Trans. Inform. Theory 52, 3885–3902 (2006)
7. Hollanti, C.: Asymmetric Space-Time Block Codes for MIMO Systems. In: 2007 IEEE ITW, Bergen, Norway (2007)
8. Vehkalahti, R.: Constructing Optimal Division Algebras for Space-Time Coding. In: 2007 IEEE ITW, Bergen, Norway (2007)
9. Elia, P., Kumar, K.R., Pawar, S.A., Kumar, P.V., Lu, H.-F.: Explicit Space-Time Codes Achieving the Diversity-Multiplexing Gain Tradeoff. IEEE Trans. Inform. Theory 52, 3869–3884 (2006)
10. Zheng, L., Tse, D.: Diversity and Multiplexing: A Fundamental Tradeoff in Multiple-Antenna Channels. IEEE Trans. Inform. Theory 49, 1073–1096 (2003)
11. Reiner, I.: Maximal Orders. Academic Press, New York (1975)
12. Jacobson, N.: Basic Algebra II. W. H. Freeman and Company, San Francisco (1980)
13. Milne, J.S.: Class Field Theory, `http://www.jmilne.org/math/coursenotes/`
14. Hong, Y., Viterbo, E., Belfiore, J.-C.: Golden Space-Time Trellis Coded Modulation. arXiv:cs.IT/0604063v3
15. Elia, P., Sethuraman, B.A., Kumar, P.V.: Perfect Space-Time Codes with Minimum and Non-Minimum Delay for Any Number of Antennas. IEEE Trans. Inform. Theory (submitted), aXiv:cs.IT/0512023

Secure Cross-Realm Client-to-Client Password-Based Authenticated Key Exchange Against Undetectable On-Line Dictionary Attacks

Kazuki Yoneyama[1], Haruki Ota[2], and Kazuo Ohta[1]

[1] The University of Electro-Communications
[2] KDDI R&D Laboratories, Inc.
yoneyama@ice.uec.ac.jp

Abstract. The cross-realm client-to-client password-based authenticated key exchange (C2C-PAKE) is protocol which two clients in two different realms with different passwords exchange a session key through their corresponding servers. Recently, a provably secure cross-realm C2C-PAKE scheme with the optimal number of rounds for a client is pointed out that the scheme is insecure against an undetectable on-line dictionary attack and an unknown-key share attack. In this paper, we propose a new cross-realm C2C-PAKE scheme with the optimal number of rounds for a client, which has resistances to previously considered attacks which should be prevented, including undetectable on-line dictionary attacks and unknown-key share attacks. Moreover, our scheme assumes no pre-established secure channels between different realms, but just basic setups of ID-based systems.

Keywords: Authenticated key exchange, different password, C2C-PAKE, cross-realm setting, undetectable on-line dictionary attacks

1 Introduction

Recently, password-based authenticated key exchange (PAKE) protocols are received much attention as practical schemes in order to share a mutual session key secretly and reliably. Basic PAKE schemes enable two entities to authenticate each other and agree on a large session key from a human memorable password. Thus, PAKE schemes are regarded as practical key exchange schemes since entities do not have any pre-shared cryptographic symmetric key, certificate or support from a trusted third party. Such basic schemes which two entities pre-share a *common* password are classified into a model called same password-authentication (SPA) model. The SPA model is most cultivated PAKE model in previous studies and is usually used for *client-to-server* key exchanges. The concept of PAKE was first introduced by Bellovin and Merritt [1] in 1992 known as encrypted key exchange (EKE). First construction of password-only PAKE in SPA model was proposed by Jablon [2] in 1996 known as simple password

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 257–266, 2007.
© Springer-Verlag Berlin Heidelberg 2007

exponential key exchange (SPEKE). Formal definitions for this setting were first given by Bellare et al. [3] and Boyko et al. [4], and a concrete construction was also given in the random oracle (RO) model. And, various protocols have been proposed to achieve secure PAKE scheme [5,6,7,8,9,10] in SPA model.

On the other hand, with a variety of communication environments such as mobile network, it is considered as one of main concerns to establish a secure channel between clients with *different* passwords. Several schemes have been presented to provide PAKE between two entities with their different passwords. Such schemes are classified into a model called different password-authentication (DPA) model. In DPA model, entities carry out key exchange with the assistance of intermediate servers since entities have no secret common information and DPA model is usually used for *client-to-client* password-based authenticated key exchanges (C2C-PAKE).

First construction of C2C-PAKE in DPA model was introduced by Steiner et al. [11] in the *single-server* setting where two clients (or n clients) are in the *same* realm. In the single-server setting, the model consists of two clients A and B (or n clients) and a server S, where clients are in the realm of server S. Though several schemes embrace the single-server setting [12,13,14,15,16,17], there is a problem that it is unrealistic that clients trying to communicate each other are registered in the same server.

From this viewpoint, Byun et al. [18] proposed C2C-PAKE in the *cross-realm* setting where two clients are in two different realms and hence there existed two servers involved. In the cross-realm setting, the model consists of two clients A and B, and two servers SA and SB, where A and B are users of SA and SB, respectively. They also newly defined the security notions according to their framework for the special settings, and claimed their protocols' security under those definitions. However, some attacks are found against this scheme by Chen [19] which showed a dictionary attack by a malicious server in a different realm, Wang et al. [20] which showed three different dictionary attacks and Kim et al. [21] which showed Denning-Sacco-style attack (a variant of dictionary attack) by an insider with knowledge of the password of a client in a different realm. Though Kim et al. also proposed an improved cross-realm C2C-PAKE in [21], Phan and Goi [22] presented two unknown-key share attacks on it. To shake off vicious circle of attack-and-remedy procedures, Byun et al. [23] introduced a provably secure cross-realm C2C-PAKE scheme. However, it is also shown that this scheme falls to an undetectable on-line dictionary attack by any adversary and malicious servers can launch a successful man-in-the-middle attack by Phan and Goi [24]. Similarly, the undetectable on-line dictionary attack to [23] also works for lately proposed scheme [25].

On all the above schemes in the cross-realm setting, clients use their corresponding servers to obtain information for authentication and directly communicate for establishing their session key after obtaining these information. So, we call these schemes have *direct communication structure*. On the other hand, there are cross-realm C2C-PAKE schemes which have another structure, called *indirect communication structure*. Clients communicate only through their

corresponding servers in the indirect communication structure. The advantage of schemes which have the indirect communication structure is to be able to reduce the optimal number of rounds for a client, i.e, 2-rounds, than the existing schemes which have the direct communication structure, i.e., 4-rounds, and to remove communications of a client across different realms. So, the indirect communication structure can reduce loads of clients. Yin and Bao [26] proposed a first cross-realm C2C-PAKE scheme (Yin-Bao scheme) which have the indirect communication structure and is provably secure. However, to prove the security of this scheme it needs a strong setup assumption, i.e., pre-established secure channels between realms. Also, despite its provable security defects of their security model caused two attacks, an undetectable on-line dictionary attack by any adversary and an unknown-key share attack by a malicious client insider, which are found by Phan and Goi [24]. Ota et al. [27] proposed a general construction of universally composable cross-realm C2C-PAKE which has the indirect communication structure and a concrete construction. Though the universal composability provides a security-preserving composition property in concurrent execution environment, their concrete construction needs the large number of total rounds.

Our contribution. We construct a cross-realm C2C-PAKE scheme which has the indirect communication structure based on Yin-Bao scheme. Our scheme only needs the optimal number of rounds for clients and servers, i.e., 2-rounds between a client and a server, and 2-rounds between servers, as Yin-Bao scheme. So, our scheme is more efficient than [27]. Furthermore, we show that our scheme has resistances to previously considered attacks which should be prevented, including undetectable on-line dictionary attacks, by applying the technique of [12] with ID-based encryption (IBE) [28]. Therefore, the undetectable on-line dictionary attack and the unknown-key share attack to Yin-Bao scheme don't work for our scheme. Also, our scheme assumes no pre-established secure channels between different realms. Instead of that, we apply a secure message authentication code (MAC) based on Sakai et al.'s ID-based non-interactive key sharing (IDNIKS) [29]. To use of IBE and IDNIKS, we need just basic setups of ID-based systems, i.e., key extractions of servers by the trusted authority. That means we consider the security in more natural setup model than the model of Yin-Bao scheme. The comparison between previous schemes and our scheme is shown in Table 1.

Table 1. Comparison between previous schemes and our scheme

	number of total rounds	undetectable on-line dictionary attack	client's inside attack	universal composability
Yin and Bao [26]	4	insecure	insecure	unsatisfied
Ota et al. [27]	12	secure	secure	satisfied
Our scheme	4	secure	secure	unsatisfied

2 Preliminaries

2.1 Cross-Realm C2C-PAKE

Our cross-realm C2C-PAKE scheme contains four parties (two clients and two servers) who will engage in the protocol. In cross-realm setting, each client is in a realm and has a corresponding server belonging to the realm. Let each password be pre-shared between a client and a corresponding server and be uniformly and independently chosen from fixed low-entropy dictionary $\mathcal{D}$ of the size $|\mathcal{D}|$. An outside adversary or a malicious insider can obtain and modify messages on unauthenticated-links channels.

2.2 Security Properties

It is desirable for C2C-PAKE protocols to possess the following security properties:

- **Known-key security:** The protocol should still achieve its goal in the face of an adversary who has learned some other session keys - unique secret keys which each run of a key exchange protocol between clients should produce.

- **Forward secrecy:** If a password of a client and the corresponding server is compromised, secrecy of past session keys is not compromised.

- **Resistance to key-compromise impersonation:** When a client's password is compromised, it may be desirable that this event does not enable an outside adversary to impersonate other entities to the client.

- **Resistance to unknown-key share:** Client A should not be able to coerce into sharing a key with any client C including a malicious client insider when in fact he thinks that he is sharing the key with client B.

- **Resistance to undetectable on-line dictionary attacks:** There is no successful adversary as follows: The adversary attempts to use a guessed password in an on-line transaction. He verifies the correctness of his guess using responses of servers. If his guess fails he must start a new transaction with servers using another guessed password. A failed guess can not be detected and logged by servers, as servers are not able to depart an honest request from a malicious request.

- **Resistance to off-line dictionary attacks:** There is no successful adversary as follows: The adversary guesses a password and verifies his guess off-line. No participation of servers is required, so servers don't notice the attack. If his guess fails the adversary tries again with another password, until he finds the proper one.

– **No key control:** The secret session key between any two clients is determined by both users taking part in, and none of the two clients can influence the outcome of the secret session key, or enforce the session key to fall into a pre-determined interval.

3 Proposed Scheme

In this section, we show our cross-realm C2C-PAKE scheme.

3.1 Bilinear Map

Using the notation of Boneh and Franklin [28], we let $\mathbb{G}_1$ be an additive group of prime order q and $\mathbb{G}_2$ be a multiplicative group of the same order q. We assume the existence of an efficiently computable, non-degenerate, bilinear map $\hat{e}$ from $\mathbb{G}_1 \times \mathbb{G}_1$ to $\mathbb{G}_2$. Typically, $\mathbb{G}_1$ will be a subgroup of the group of points on an elliptic curve over a finite field, $\mathbb{G}_2$ will be a subgroup of the multiplicative group of a related finite field and the map $\hat{e}$ will be derived from either the Weil or Tate pairing on the elliptic curve. By $\hat{e}$ being bilinear, we mean that for any $Q, W \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_q$:

$$\hat{e}(aQ, bW) = \hat{e}(Q, W)^{ab} = \hat{e}(abQ, W).$$

By $\hat{e}$ being non-degenerate, we mean that for some element $P \in \mathbb{G}_1$, we have $\hat{e}(P, P) \neq 1_{\mathbb{G}_2}$.

3.2 Notation

Let p be a prime and let g be a generator of a large subgroup from $\mathbb{Z}_p^*$ of prime order p. Note that g is not element of bilinear groups. A and B are identity of two clients in two different realms, and SA and SB are identities of their corresponding servers respectively. A and SA (resp. B and SB) shared common secret password pw_A (resp. pw_B), and SA and SB received their private keys $sk_{SA} = s\bar{H}(SA)$ and $sk_{SB} = s\bar{H}(SB)$ from the trusted authority in the ID-based system as [28] and [29] in advance, where $s \in \mathbb{Z}_q$ is the master secret of the trusted authority and $\bar{H} : \{0,1\}^* \rightarrow \mathbb{G}_1$ is a collision-resistant hash function. $(\mathsf{Enc}, \mathsf{Dec})$ is Boneh-Franklin ID-based encryption (IBE) [28] with Fujisaki-Okamoto conversion [30], which satisfies semantically secure against adaptive chosen ciphertext attacks (ID-CCA), where $\mathsf{Enc}_{id}(m)$ is encryption algorithm of a message m using an identity id and $\mathsf{Dec}_{sk_{id}}(c)$ is decryption algorithm of a cipher-text c using a private key sk_{id}. MAC_{mk} is an existentially unforgeable MAC scheme against adaptively chosen message attacks where $mk \in \mathbb{G}_2$ is a MAC key. $H_1, H_2, H_3 : \{0,1\}^* \rightarrow \{0,1\}^k$ are hash functions modeled as random oracles, where k is a sufficiently large security parameter.

For simplicity, we omit "(mod p)" in this paper when computing the modular exponentiation. "$v \xleftarrow{R} V$" means randomly choosing an element v of a set V.

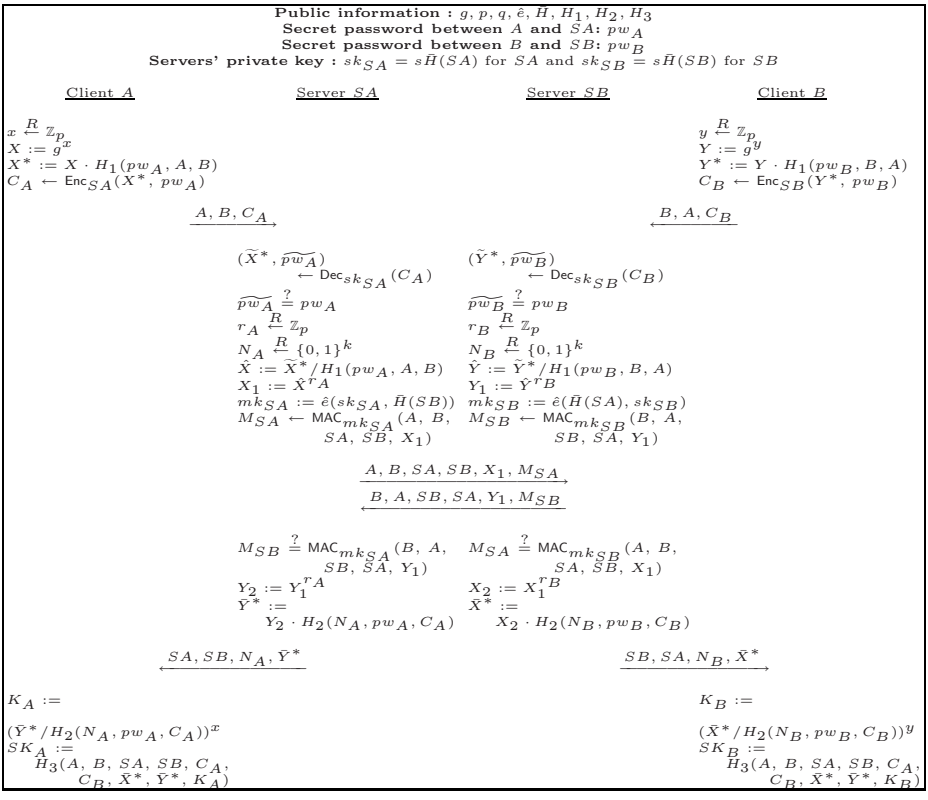


Fig. 1. A high-level overview of our protocol

3.3 Protocol Description

Here, we show the construction of our cross-realm C2C-PAKE scheme. Our protocol has the indirect communication structure as Yin-Bao scheme. But, all communication channels are unauthenticated-links unlike Yin-Bao scheme. A high-level overview of our protocol appears in Figure 1.

Then, our protocol is described as follows:

First, clients A and B choose $x, y \in \mathbb{Z}_p$ randomly, computes $X = g^x, Y = g^y$, and blind them as $X^* = X \cdot H_1(pw_A, A, B)$, $Y^* = Y \cdot H_1(pw_B, B, A)$ respectively. Also, they generate $C_A \leftarrow \mathsf{Enc}_{SA}(X^*, pw_A)$, $C_B \leftarrow \mathsf{Enc}_{SB}(Y^*, pw_B)$ by using their corresponding servers' identities SA and SB respectively, and A sends (A, B, C_A) to SA and B sends (B, A, C_B) to SB.

Secondly, servers SA and SB decrypt $(\widetilde{X}^*, \widetilde{pw_A}) \leftarrow \mathsf{Dec}_{sk_{SA}}(C_A)$, $(\widetilde{Y}^*, \widetilde{pw_B}) \leftarrow \mathsf{Dec}_{sk_{SB}}(C_B)$ by using sk_{SA} and sk_{SB} respectively.

If $\widetilde{pw_A} \neq pw_A$, then SA aborts the session, and if $\widetilde{pw_B} \neq pw_B$, then SB aborts the session too. Otherwise, SA computes $\hat{X} = \widetilde{X}^*/H_1(pw_A, A, B)$, blinds it as $X_1 := \hat{X}^{r_A}$ where r_A is SA's first random value from $\mathbb{Z}_p$, computes his MAC key $mk_{SA} = \hat{e}(sk_{SA}, \bar{H}(SB))$ by using Sakai et al.'s IDNIKS, and generates a

MAC $M_{SA} \leftarrow \mathsf{MAC}_{mk_{SA}}(A, B, SA, SB, X_1)$. SB also computes Y_1, mk_{SB} and generates M_{SB} similarly. Then SA and SB exchange $(A, B, SA, SB, X_1, M_{SA})$ and $(B, A, SB, SA, Y_1, M_{SB})$. After that, SA and SB verify M_{SB} and M_{SA} by using their MAC keys respectively. If MACs are invalid, they abort the session. Otherwise, SA blinds $Y_2 = Y_1^{r_A}$ and computes $\bar{Y}^* = Y_2 \cdot H_2(N_A, pw_A, C_A)$ where N_A is SA's second random value from $\{0,1\}^k$. SB performs similar operations and obtains $\bar{X}^*$. At the end, SA sends $\bar{Y}^*$, N_A to A, and SB sends $\bar{X}^*$, N_B to B.

Thirdly, A and B compute their ephemeral Diffie-Hellman keys $K_A = (\bar{Y}^* \,/\, H_2(N_A, pw_A, C_A))^x$ and $K_B = (\bar{X}^* \,/\, H_2(N_B, pw_B, C_B))^y$ respectively. Session keys are generated from the ephemeral Diffie-Hellman key and transcripts, $SK_A = H_3(A, B, SA, SB, C_A, C_B, \bar{X}^*, \bar{Y}^*, K_A)$ and $SK_B = H_3(A, B, SA, SB, C_A, C_B, \bar{X}^*, \bar{Y}^*, K_B)$. Note that transcripts of the protocol are public.

When session keys are honestly generated, $SK_A = SK_B$ since $K_A = (g^{y r_A r_B})^x$ and $K_B = (g^{x r_A r_B})^y$.

3.4 Design Principles

Our protocol can be viewed as an extension of Yin-Bao scheme. The main deference consists in the description of servers.

First, upon receiving an input from a client the corresponding server verifies the validity of encrypted password of the client and him. This procedure prevents undetectable on-line dictionary attacks as the technique of Lin et al. [12]. And, by using IBE, clients don't need to receive servers' certificates of public keys since IBE is able to encrypt a message with only the recipient's identity.

Also, servers exchange ephemeral keys with MACs by using MAC keys computed from their private keys. By using IDNIKS, servers are able to share the same MAC key non-interactively. Thus, authenticity of ephemeral keys is guaranteed even if different realms are connected by unauthenticated-links channel in our protocol.

Furthermore, when a client blinds X with his password, we make the client include the identities of both clients into the computation of the password-based blinding factors. This procedure prevents unknown-key share attacks by a malicious client insider as the technique of Choo et al. [31].

4 Analysis of Security

In this section, we show security properties of our scheme. For space limitation, we cannot give all detailed analyses here, only brief outline. In this paper, in particular, we show resistance to unknown-key share attacks and resistance to undetectable on-line dictionary attacks which Yin-Bao scheme does not satisfy. We will show all details in the full paper.

4.1 Resistance to Unknown-Key Share

In the case of that a malicious client insider C wants to convince a client B in the networks that B share a session key with C while in fact B shares the key

with another client A, C is required to know the password pw_A so that he could pass the verification of B. Otherwise, the attack hardly works.

So, we consider the case of that a malicious insider C wants to share a session key with a client B, while B believes that he shares the session key with a client A. Then, C cannot validly modify (B, A, C_B) into $(B, C, \widetilde{C_B})$ in the message to SB since C cannot compute $H_1(pw_B, B, C)$ instead of $H_1(pw_B, B, A)$ without the knowledge of pw_C. If C doesn't modify C_B, then B's session key is randomized by SB's operation to compute $\hat{Y}$ and C cannot obtain information of it. Also, C cannot validly modify $(A, B, SA, SB, X_1, M_{SA})$ to $(C, B, SC, SB, \widetilde{X_1}, \widetilde{M_{SA}})$ since he has no information of SA's MAC key. Thus, the probability of C successfully impersonates of A can be negligible.

4.2 Resistance to Undetectable On-Line Dictionary Attacks

A chance to attack is only when an adversary sends messages in the first phase as a client to servers since messages in other phases to servers contain no information of passwords. However, the adversary cannot replay to send messages in the first phase according to guessed passwords, since if he computes the ciphertext with a wrong password, then the server verifies it and will abort. Thus, since the adversary successfully continues on-line dictionary attacks only if the adversary successfully guesses the password, the success probability can be negligible close to $1/|\mathcal{D}|$.

5 Conclusion

We proposed a new cross-realm C2C-PAKE scheme which clients only need the optimal number of communication rounds and need no communication across different realms. Furthermore, we showed our scheme is secure against unknown-key share attacks and undetectable on-line dictionary attacks which successfully work to Yin-Bao scheme.

A remaining problem of further researches is to give provable security to our scheme. For proving formal security, since the formal security model of [26] has some defects, we have to cultivate the model. It will be achieved by referring to the recent formal model of authenticated key exchange, e.g., the model of LaMacchia et al. [32].

References

1. Bellovin, S.M., Merritt, M.: Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks. In: IEEE S&P 1992, pp. 72–84 (1992)
2. Jablon, D.P.: Strong Password-Only Authenticated Key Exchange. Computer Communication Review, ACM SIGCOMM 26(5), 5–26 (1996)
3. Bellare, M., Pointcheval, D., Rogaway, P.: Authenticated Key Exchange Secure against Dictionary Attacks. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 139–155. Springer, Heidelberg (2000)

4. Boyko, V., MacKenzie, P.D., Patel, S.: Provably Secure Password-Authenticated Key Exchange Using Diffie-Hellman. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 156–171. Springer, Heidelberg (2000)
5. Goldreich, O., Lindell, Y.: Session-Key Generation Using Human Passwords Only. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 408–432. Springer, Heidelberg (2001)
6. Katz, J., Ostrovsky, R., Yung, M.: Efficient Password-Authenticated Key Exchange Using Human-Memorable Passwords. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 475–494. Springer, Heidelberg (2001)
7. Gennaro, R., Lindell, Y.: A Framework for Password-Based Authenticated Key Exchange. In: Biham, E. (ed.) Advances in Cryptology – EUROCRPYT 2003. LNCS, vol. 2656, pp. 408–432. Springer, Heidelberg (2003)
8. Nguyen, M.H., Vadhan, S.P.: Simpler Session-Key Generation from Short Random Passwords. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 428–445. Springer, Heidelberg (2004)
9. Abdalla, M., Pointcheval, D.: Simple Password-Based Encrypted Key Exchange Protocols. In: Menezes, A.J. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 191–208. Springer, Heidelberg (2005)
10. Canetti, R., Halevi, S., Katz, J., Lindell, Y., MacKenzie, P.D.: Universally Composable Password-Based Key Exchange. In: Cramer, R.J.F. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 404–421. Springer, Heidelberg (2005)
11. Steiner, M., Tsudik, G., Waidner, M.: Refinement and Extension of Encrypted Key Exchange. ACM Operating Systems Review 29(3), 22–30 (1995)
12. Lin, C.L., Sun, H.M., Hwang, T.: Three-party Encrypted Key Exchange: Attacks and A Solution. ACM Operating Systems Review 34(4), 12–20 (2000)
13. Lee, T.F., Hwang, T., Lin, C.L.: Enhanced three-party encrypted key exchange without server public keys. Elsevier Computers & Security 23(7), 571–577 (2004)
14. Chang, Y.F., Chang, C.C.: Password-authenticated 3PEKE with Round Efficiency without Server's Public Key. In: CW 2005, pp. 340–344 (2005)
15. Abdalla, M., Fouque, P.A., Pointcheval, D.: Password-Based Authenticated Key Exchange in the Three-Party Setting. In: Public Key Cryptography 2005, pp. 65–84 (2005)
16. Byun, J.W., Lee, D.H.: N-Party Encrypted Diffie-Hellman Key Exchange Using Different Passwords. In: Ioannidis, J., Keromytis, A.D., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 75–90. Springer, Heidelberg (2005)
17. Lu, R., Cao, Z.: Simple three-party key exchange protocol. Elsevier Computers & Security 26(1), 94–97 (2007)
18. Byun, J.W., Jeong, I.R., Lee, D.H., Park, C.S.: Password-Authenticated Key Exchange between Clients with Different Passwords. In: Deng, R.H., Qing, S., Bao, F., Zhou, J. (eds.) ICICS 2002. LNCS, vol. 2513, pp. 134–146. Springer, Heidelberg (2002)
19. Chen, L.: A Weakness of the Password-Authenticated Key Agreement between Clients with Different Passwords Scheme. In: ISO/IEC JTC 1/SC27 N3716 (2003)
20. Wang, S., Wang, J., Xu, M.: Weaknesses of a Password-Authenticated Key Exchange Protocol between Clients with Different Passwords. In: Jakobsson, M., Yung, M., Zhou, J. (eds.) ACNS 2004. LNCS, vol. 3089, pp. 414–425. Springer, Heidelberg (2004)
21. Kim, J., Kim, S., Kwak, J., Won, D.: Cryptanalysis and Improvement of Password Authenticated Key Exchange Scheme between Clients with Different Passwords. In: Laganà, A., Gavrilova, M., Kumar, V., Mun, Y., Tan, C.J.K., Gervasi, O. (eds.) ICCSA 2004. LNCS, vol. 3043, pp. 895–902. Springer, Heidelberg (2004)

22. Phan, R.C.W., Goi, B.M.: Cryptanalysis of an Improved Client-to-Client Password-Authenticated Key Exchange (C2C-PAKE) Scheme. In: Ioannidis, J., Keromytis, A.D., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 33–39. Springer, Heidelberg (2005)
23. Byun, J.W., Lee, D.H., Lim, J.: Efficient and Provably Secure Client-to-Client Password-Based Key Exchange Protocol. In: Zhou, X., Li, J., Shen, H.T., Kitsuregawa, M., Zhang, Y. (eds.) APWeb 2006. LNCS, vol. 3841, pp. 830–836. Springer, Heidelberg (2006)
24. Phan, R.C.W., Goi, B.M.: Cryptanalysis of Two Provably Secure Cross-Realm C2C-PAKE Protocols. In: Barua, R., Lange, T. (eds.) INDOCRYPT 2006. LNCS, vol. 4329, pp. 104–117. Springer, Heidelberg (2006)
25. Gang, Y., Dengguo, F., Xiaoxi, H.: Improved Client-to-Client Password-Authenticated Key Exchange Protocol. In: IEEE ARES 2007, pp. 564–574 (2007)
26. Yin, Y., Bao, L.: Secure Cross-Realm C2C-PAKE Protocol. In: Batten, L.M., Safavi-Naini, R. (eds.) ACISP 2006. LNCS, vol. 4058, pp. 395–406. Springer, Heidelberg (2006)
27. Ota, H., Yoneyama, K., Kiyomoto, S., Tanaka, T., Ohta, K.: Universally Composable Client-to-Client General Authenticated Key Exchange. IPSJ Journal 48(9), 3073–3088 (2007)
28. Boneh, D., Franklin, M.K.: Identity-Based Encryption from the Weil Pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
29. Sakai, R., Ohgishi, K., Kasahara, M.: Cryptosystems based on pairing. In: SCIS 2000 (2000)
30. Fujisaki, E., Okamoto, T.: How to Enhance the Security of Public-Key Encryption at Minimum Cost. In: Imai, H., Zheng, Y. (eds.) PKC 1999. LNCS, vol. 1560, pp. 53–68. Springer, Heidelberg (1999)
31. Choo, K.K.R., Boyd, C., Hitchcock, Y.: Examining Indistinguishability-Based Proof Models for Key Establishment Protocols. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 585–604. Springer, Heidelberg (2005)
32. LaMacchia, B., Lauter, K., Mityagin, A.: Stronger Security of Authenticated Key Exchange. In: Provsec (to appear, 2007)

Links Between Discriminating and Identifying Codes in the Binary Hamming Space

Irène Charon, Gérard Cohen, Olivier Hudry, and Antoine Lobstein

GET - Télécom Paris & CNRS - LTCI UMR 5141
46, rue Barrault, 75634 Paris Cedex 13 - France
{irene.charon,gerard.cohen,olivier.hudry,antoine.lobstein}@enst.fr

Abstract. Let F^n be the binary n-cube, or binary Hamming space of dimension n, endowed with the Hamming distance, and $\mathcal{E}^n$ (respectively, $\mathcal{O}^n$) the set of vectors with even (respectively, odd) weight. For $r \geq 1$ and $x \in F^n$, we denote by $B_r(x)$ the ball of radius r and centre x. A code $C \subseteq F^n$ is said to be r-identifying if the sets $B_r(x) \cap C$, $x \in F^n$, are all nonempty and distinct. A code $C \subseteq \mathcal{E}^n$ is said to be r-discriminating if the sets $B_r(x) \cap C$, $x \in \mathcal{O}^n$, are all nonempty and distinct. We show that the two definitions, which were given for general graphs, are equivalent in the case of the Hamming space, in the following sense: for any odd r, there is a bijection between the set of r-identifying codes in F^n and the set of r-discriminating codes in F^{n+1}.

Keywords: Graph Theory, Coding Theory, Discriminating Codes, Identifying Codes, Hamming Space, Hypercube

1 Introduction

We define identifying and discriminating codes in a connected, undirected graph $G = (V, E)$, in which a *code* is simply a nonempty subset of vertices. These definitions can help, in various meanings, to unambiguously determine a vertex. The motivations may come from processor networks where we wish to locate a faulty vertex under certain conditions, or from the need to identify an individual, given its set of attributes.

In G we define the usual distance $d(v_1, v_2)$ between two vertices $v_1, v_2 \in V$ as the smallest possible number of edges in any path between them. For an integer $r \geq 0$ and a vertex $v \in V$, we define $B_r(v)$ the *ball* of radius r centred at v, as the set of vertices within distance r from v. Whenever two vertices v_1 and v_2 are such that $v_1 \in B_r(v_2)$ (or, equivalently, $v_2 \in B_r(v_1)$), we say that they *r-cover* each other. A set $X \subseteq V$ r-covers a set $Y \subseteq V$ if every vertex in Y is r-covered by at least one vertex in X.

The elements of a code $C \subseteq V$ are called *codewords*. For each vertex $v \in V$, we denote by

$$K_{C,r}(v) = C \cap B_r(v)$$

the set of codewords r-covering v. Two vertices v_1 and v_2 with $K_{C,r}(v_1) \neq K_{C,r}(v_2)$ are said to be *r-separated* by code C, and any codeword belonging to exactly one of the two sets $B_r(v_1)$ and $B_r(v_2)$ is said to *r-separate* v_1 and v_2.

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 267–270, 2007.
© Springer-Verlag Berlin Heidelberg 2007

A code $C \subseteq V$ is called *r-identifying* [10] if all the sets $K_{C,r}(v)$, $v \in V$, are nonempty and distinct. In other words, every vertex is r-covered by at least one codeword, and every pair of vertices is r-separated by at least one codeword. Such codes are also sometimes called *differentiating dominating sets* [8].

We now suppose that G is bipartite: $G = (V = I \cup A, E)$, with no edges inside I nor A — here, A stands for *attributes* and I for *individuals*. A code $C \subseteq A$ is said to be *r-discriminating* [4] if all the sets $K_{C,r}(i)$, $i \in I$, are nonempty and distinct. From the definition we see that we can consider only odd values of r.

In the following, we drop the general case and turn to the binary Hamming space of dimension n, also called the binary n-cube, which is a regular bipartite graph. First we need to give some specific definitions and notation.

We consider the n-cube as the set of binary row-vectors of length n, and as so, we denote it by $G = (F^n, E)$ with $F = \{0, 1\}$ and $E = \{\{x, y\} : d(x, y) = 1\}$, the usual graph distance $d(x, y)$ between two vectors x and y being called here the *Hamming distance* — it simply consists of the number of coordinates where x and y differ. The *Hamming weight* of a vector x is its distance to the all-zero vector, i.e., the number of its nonzero coordinates. A vector is said to be *even* (respectively, *odd*) if its weight is even (respectively, odd), and we denote by $\mathcal{E}^n$ (respectively, $\mathcal{O}^n$) the set of the 2^{n-1} even (respectively, odd) vectors in F^n. Without loss of generality, for the definition of an r-discriminating code, we choose the set A to be $\mathcal{E}^n$, and the set I to be $\mathcal{O}^n$. Additions are carried coordinatewise and modulo two.

Given a vector $x \in F^n$, we denote by $\pi(x)$ its parity-check bit: $\pi(x) = 0$ if x is even, $\pi(x) = 1$ if x is odd. Therefore, if $|$ stands for concatenation of vectors, $x|\pi(x)$ is an even vector. Finally, we denote by $M_r(n)$ (respectively, $D_r(n)$) the smallest possible cardinality of an r-identifying (respectively, r-discriminating) code in F^n.

In Section 2, we show that in the particular case of Hamming space, the two notions of r-identifying and r-discriminating codes actually coincide for all odd values of r and all $n \geq 2$, in the sense that there is a bijection between the set of r-identifying codes in F^n and the set of r-discriminating codes in F^{n+1}.

2 Identifying Is Discriminating

As we now show with the following two theorems, for any odd $r \geq 1$, any r-identifying code in F^n can be extended into an r-discriminating code in F^{n+1}, and any r-discriminating code in F^n can be shortened into an r-identifying code in F^{n-1}. First, observe that r-identifying codes exist in F^n if and only if $r < n$.

Theorem 1. *Let $n \geq 2, p \geq 0$ be such that $2p + 1 < n$, let $C \subseteq F^n$ be a $(2p + 1)$-identifying code and let*

$$C' = \{c|\pi(c) : c \in C\}.$$

Then C' is $(2p + 1)$-discriminating in F^{n+1}. Therefore,

$$D_{2p+1}(n + 1) \leq M_{2p+1}(n). \tag{1}$$

Proof. Let $r = 2p + 1$. By construction, C' contains only even vectors. We shall prove that (a) any odd vector $x \in \mathcal{O}^{n+1}$ is r-covered by at least one codeword of C'; (b) given any two distinct odd vectors $x, y \in \mathcal{O}^{n+1}$, there is at least one codeword in C' which r-separates them.

(a) We write $x = x_1|x_2$ with $x_1 \in F^n$ and $x_2 \in F$. Because C is r-identifying in F^n, there is a codeword $c \in C$ with $d(x_1, c) \leq r$. Let $c' = c|\pi(c)$.

If $d(x_1, c) \leq r - 1$, then whatever the values of x_2 and $\pi(c)$ are, we have $d(x, c') \leq r$; we assume therefore that $d(x_1, c) = r = 2p + 1$, which implies that x_1 and c have different parities. Since $x_1|x_2$ and $c|\pi(c)$ also have different parities, we have $x_2 = \pi(c)$ and $d(x, c') = r$. So the codeword $c' \in C'$ r-covers x.

(b) We write $x = x_1|x_2$, $y = y_1|y_2$, with $x_1, y_1 \in F^n$, $x_2, y_2 \in F$. Since C is r-identifying in F^n, there is a codeword $c \in C$ which is, say, within distance r from x_1 and not from y_1: $d(x_1, c) \leq r$, $d(y_1, c) > r$. Let $c' = c|\pi(c)$.

For the same reasons as above, x is within distance r from c', whereas obviously, $d(y, c') \geq d(y_1, c) > r$. So $c' \in C'$ r-separates x and y.

Inequality (1) follows. □

Theorem 2. *Let $n \geq 3, p \geq 0$ be such that $2p+2 < n$, let $C \subseteq \mathcal{E}^n$ be a $(2p+1)$-discriminating code and let $C' \subseteq F^{n-1}$ be any code obtained by the deletion of one coordinate in C. Then C' is $(2p+1)$-identifying in F^{n-1}. Therefore,*

$$M_{2p+1}(n-1) \leq D_{2p+1}(n). \tag{2}$$

Proof. Let $r = 2p + 1$. Let $C \subseteq \mathcal{E}^n$ be an r-discriminating code and $C' \subseteq F^{n-1}$ be the code obtained by deleting, say, the last coordinate in C. We shall prove that (a) any vector $x \in F^{n-1}$ is r-covered by at least one codeword of C'; (b) given any two distinct vectors $x, y \in F^{n-1}$, there is at least one codeword in C' which r-separates them.

(a) The vector $x|(\pi(x)+1) \in F^n$ is odd. As such, it is r-covered by a codeword $c = c'|u \in C \subseteq \mathcal{E}^n$: $c' \in C'$, $u = \pi(c')$, and $d(x|(\pi(x) + 1), c) \leq r$. This proves that x is within distance r from a codeword of C'.

(b) Both $x|(\pi(x) + 1)$ and $y|(\pi(y) + 1)$ are odd vectors in F^n, and there is a codeword $c = c'|u \in C \subseteq \mathcal{E}^n$, with $c' \in C'$, $u = \pi(c')$, which r-separates them: without loss of generality, $d(x|(\pi(x)+1), c) \leq r$ whereas $d(y|(\pi(y)+1), c)$, which is an odd integer, is at least $r+2$. Then obviously, $d(x, c') \leq r$ and $d(y, c') \geq r+1$, i.e., there is a codeword in C' which r-separates x and y.

Inequality (2) follows. □

Corollary 1. *For all $n \geq 2$ and $p \geq 0$ such that $2p + 1 < n$, we have:*

$$D_{2p+1}(n + 1) = M_{2p+1}(n).$$

□

3 Conclusion

We have shown the equivalence between discriminating and identifying codes; the latter being already well studied, this entails a few consequences on discriminating codes.

For example, the complexity of problems on discriminating codes is the same as that for identifying codes; in particular, it is known [9] that deciding whether a given code $C \subseteq F^n$ is r-identifying is co-NP-complete.

For yet another issue, constructions, we refer to, e.g., [1]–[3], [6], [9], [10] or [11]; visit also [12]. In the recent [7], tables for exact values or bounds on $M_1(n)$, $2 \leq n \leq 19$, and $M_2(n)$, $3 \leq n \leq 21$, are given.

Discriminating codes have not been thoroughly studied so far; let us simply mention [4] for a general introduction and [5] in the case of planar graphs.

References

1. Blass, U., Honkala, I., Litsyn, S.: On The Size of Identifying Codes. In: Fossorier, M.P.C., Imai, H., Lin, S., Poli, A. (eds.) AAECC-13. LNCS, vol. 1719, pp. 142–147. Springer, Heidelberg (1999)
2. Blass, U., Honkala, I., Litsyn, S.: On Binary Codes for Identification. J. of Combinatorial Designs 8, 151–156 (2000)
3. Blass, U., Honkala, I., Litsyn, S.: Bounds on Identifying Codes. Discrete Mathematics 241, 119–128 (2001)
4. Charbit, E., Charon, I., Cohen, G., Hudry, O.: Discriminating Codes in Bipartite Graphs. Electronic Notes in Discrete Mathematics 26, 29–35 (2006)
5. Charon, I., Cohen, G., Hudry, O., Lobstein, A.: Discriminating Codes in (Bipartite) Planar Graphs. European Journal of Combinatorics (to appear)
6. Exoo, G.: Computational Results on Identifying t-codes (preprint, 1999)
7. Exoo, G., Laihonen, T., Ranto, S.: Improved Upper Bounds on Binary Identifying Codes. IEEE Trans. Inform. Theory (to appear)
8. Gimbel, J., Van Gorden, B.D., Nicolescu, M., Umstead, C., Vaiana, N.: Location with Dominating Sets. Congressus Numerantium 151, 129–144 (2001)
9. Honkala, I., Lobstein, A.: On the Complexity of the Identification Problem in Hamming Spaces. Acta Informatica 38, 839–845 (2002)
10. Karpovsky, M.G., Chakrabarty, K., Levitin, L.B.: On a New Class of Codes for Identifying Vertices in Graphs. IEEE Trans. Inform. Theory 44(2), 599–611 (1998)
11. Ranto, S.: Identifying and Locating-Dominating Codes in Binary Hamming Spaces. Ph. D Thesis, University of Turku (2007)
12. `http://www.infres.enst.fr/~lobstein/bibLOCDOMetID.html`

Construction of Rotation Symmetric Boolean Functions on Odd Number of Variables with Maximum Algebraic Immunity

Sumanta Sarkar and Subhamoy Maitra

Applied Statistics Unit, Indian Statistical Institute
203 B T Road, Kolkata 700108, India
{sumanta_r, subho}@isical.ac.in

Abstract. In this paper we present a theoretical construction of Rotation Symmetric Boolean Functions (RSBFs) on odd number of variables with maximum possible algebraic immunity (AI) and further these functions are not symmetric. Our RSBFs are of better nonlinearity than the existing theoretical constructions with maximum possible AI. To get very good nonlinearity, which is important for practical cryptographic design, we generalize our construction to a construction cum search technique in the RSBF class. We find 7, 9, 11 variable RSBFs with maximum possible AI having nonlinearities 56, 240, 984 respectively with very small amount of search after our basic construction.

Keywords: Algebraic Immunity, Boolean Function, Nonlinearity, Nonsingular Matrix, Rotational Symmetry, Walsh Spectrum.

1 Introduction

Algebraic attack has received a lot of attention recently in studying the security of Stream ciphers as well as Block ciphers (see [1,2,3,4,5] and the references therein). One necessary condition to resist this attack is that the Boolean function used in the cipher should have good algebraic immunity (AI). It is known [2] that for any n-variable Boolean function, maximum possible AI is $\lceil \frac{n}{2} \rceil$.

So far a few theoretical constructions of Boolean functions with optimal AI have been presented in the literature. In [4], the first ever construction of Boolean functions with maximum AI was proposed. Later, the construction of symmetric Boolean functions with maximum AI was given in [6]. For odd number of input variables, majority functions are the examples of symmetric functions with maximum AI. Recently in [9], the idea of modifying symmetric functions to get other functions with maximum AI is proposed using the technique of [5].

An n-variable Boolean function which is invariant under the action of the cyclic group C_n on the set $V_n = \{0,1\}^n$ is called Rotation Symmetric Boolean functions (RSBFs). We denote the class of all n-variable RSBFs as $S(C_n)$. On the other hand, an n-variable symmetric Boolean function is one which is invariant under the action of the symmetric group S_n on the set V_n and we denote the

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 271–280, 2007.
© Springer-Verlag Berlin Heidelberg 2007

class of all n-variable symmetric Boolean functions as $S(S_n)$. The class $S(C_n)$ has been shown to be extremely rich as the class contains Boolean functions with excellent cryptographic as well as combinatorial significance (see [7,12] and the references therein). As for example, in [7], 9-variable Boolean functions with nonlinearity 241 have been discovered in $S(C_9)$ which had been open for a long period. Also an RSBF has a short representation which is interesting for the design purpose of ciphers. Since $C_n \subset S_n$, we have $S(S_n) \subset S(C_n)$. Therefore all the symmetric functions with maximum AI are also examples of RSBFs with maximum AI. The class $S(C_n) \setminus S(S_n)$ becomes quite huge for larger n. However, so far there has been no known construction method available which gives n-variable RSBFs belonging to $S(C_n) \setminus S(S_n)$, having the maximum AI. It has been proved in [10,13], that the majority function (upto complementation) is the only possible symmetric Boolean function on odd number of variables which has maximum AI. Hence, there is a need to get a theoretical construction method which provides new class of RSBFs with maximum AI, which are not symmetric.

In this paper we present a construction method (Construction 1) that generates RSBFs on odd variables (≥ 5) with maximum AI, which are not symmetric. Note that up to 3 variables, RSBFs are all symmetric, and that is the reason we concentrate on $n \geq 5$. In this construction, complement of n-variable majority function is considered and its outputs are toggled at the inputs of the orbits of size $\lfloor \frac{n}{2} \rfloor$ and $\lceil \frac{n}{2} \rceil$ respectively. These orbits are chosen in such a manner that a sub matrix associated to these points is nonsingular. This idea follows the work of [5], where the sub matrix was introduced to reduce the complexity for determining AI of a Boolean function. We also show that the functions of this class have nonlinearity $2^{n-1} - \binom{n-1}{\lfloor \frac{n}{2} \rfloor} + 2$ which is better than $2^{n-1} - \binom{n-1}{\lfloor \frac{n}{2} \rfloor}$, the lower bound [11] on nonlinearity of any n (odd) variable function with maximum AI; further the general theoretical constructions [4,6] could only achieve this lower bound so far.

We present a generalization of the Construction 1 in Construction 2 which is further generalized in Construction 3. In each of the generalizations we release the restrictions on choosing orbits and achieve better nonlinearity of the constructed RSBFs with maximum AI. We present instances of RSBFs having nonlinearities equal to or slightly less than $2^{n-1} - 2^{\frac{n-1}{2}}$ for odd n, $7 \leq n \leq 11$.

One may refer to [7,6] for basics of Boolean functions, and in particular, symmetric and rotation symmetric Boolean functions. Also [5] gives detailed description on algebraic immunity of a Boolean function.

2 Existing Results Related to Annihilators

We take the degree graded lexicographic order "$<^{dgl}$" on the set of all monomials on n-variables $\{x_{m_1} \dots x_{m_k} : 1 \leq k \leq n, 1 \leq m_1, \dots, m_k \leq n\}$.

Let $v_{n,d}(x) = (m_1(x), m_2(x), \dots, m_{\sum_{i=0}^{d} \binom{n}{i}}(x))$, where $m_i(x)$ is the i-th monomial as in the order ($<^{dgl}$) evaluated at the point $x = (x_1, x_2, \dots, x_n)$.

Definition 1. *For an n-variable Boolean function f, let* $M_{n,d}(f)$ *be an* $wt(f) \times \sum_{i=0}^{d} \binom{n}{i}$ *matrix defined as* $M_{n,d}(f) = \left[v_{n,d}(P_1), v_{n,d}(P_2), \ldots, v_{n,d}(P_{wt(f)})\right]^T$, *where* $0 \leq d \leq n$, $P_i \in supp(f)$, $1 \leq i \leq wt(f)$ *and* $P_1 <^{dgl} P_2 <^{dgl} \cdots <^{dgl} P_{wt(f)}$.

Let h be an n-variable Boolean function defined as $h(x) = 1$ if and only if $wt(x) \leq d$ for $0 \leq d \leq n$. Then from [5] we know that $M_{n,d}(h)^{-1} = M_{n,d}(h)$.

Let f be an n-variable Boolean function. Let a nonzero n-variable function g be an annihilator of f, i.e., $f(x_1, \ldots, x_n) * g(x_1, \ldots, x_n) = 0$ for all $(x_1, \ldots, x_n) \in V_n$. That means,

$$g(x_1, \ldots, x_n) = 0 \text{ if } f(x_1, \ldots, x_n) = 1. \tag{1}$$

If the degree of the function g is less than equal to d, then the ANF of g is of the form

$$g(x_1, \ldots, x_n) = a_0 + \sum_{i=0}^{n} a_i x_i + \cdots + \sum_{1 \leq i_1 < i_2 \cdots < i_d \leq n} a_{i_1,\ldots,i_d} x_{i_1} \cdots x_{i_d},$$

where $a_0, a_1, \ldots, a_{12}, \ldots a_{n-d+1,\ldots,n}$ are from $\{0, 1\}$ not all zero. Then the relation 1 gives a homogeneous linear equation

$$a_0 + \sum_{i=0}^{n} a_i x_i + \cdots + \sum_{1 \leq i_1 < i_2 \cdots < i_d \leq n} a_{i_1,\ldots,i_d} x_{i_1} \cdots x_{i_d} = 0, \tag{2}$$

with $a_0, a_1, \ldots, a_{12}, \ldots a_{n-d+1,\ldots,n}$ as variables for each input $(x_1, \ldots, x_n) \in supp(f)$ and thus $wt(f)$ homogeneous linear equations in total. If this system of equations has a nonzero solution, then g having the coefficients in its ANF which is the solution of this system of equations is an annihilator of f of degree less than or equal to d. Note that in this system of equations $M_{n,d}(f)$ is the coefficient matrix. Then it is clear that if the rank of $M_{n,d}(f)$ is equal to $\sum_{i=0}^{d} \binom{n}{i}$, f does not posses any annihilator. If for $d = \lfloor \frac{n}{2} \rfloor$, both of f and $1 \oplus f$ do not have any annihilator of degree less than or equal to d, then f has maximum algebraic immunity, i.e., $\lceil \frac{n}{2} \rceil$.

2.1 Existence of Boolean Functions with Maximum AI on Odd Variables

Let us start with a few available results on n-variable Boolean functions with maximum AI. Henceforth we will consider the $<^{dgl}$ ordering of the inputs of V_n unless stated.

Proposition 1. *[3] An odd variable Boolean function with maximum* AI *must be balanced.*

Proposition 2. *Let f be an n (odd) variable Boolean function. Then* AI *of f is* $\lceil \frac{n}{2} \rceil$ *if and only if f is balanced and* $M_{n,\lceil \frac{n}{2} \rceil - 1}(f)$ *has full rank.*

We denote by G_n, the n (odd) variable Boolean function with $G_n(X) = 1$ if $wt(X) \leq \lceil \frac{n}{2} \rceil - 1$, and $G_n(X) = 0$ if $wt(X) \geq \lceil \frac{n}{2} \rceil$.

The function G_n is a balanced symmetric Boolean function and it has been proved [6] that this function has maximum algebraic immunity, i.e., $\lceil \frac{n}{2} \rceil$. Then both of the matrices $M_{n,\lceil \frac{n}{2} \rceil - 1}(G_n)$ and $M_{n,\lceil \frac{n}{2} \rceil - 1}(1 \oplus G_n)$ are of the order $2^{n-1} \times 2^{n-1}$ and nonsingular. Now we take a look at a construction of an n-variable Boolean function having maximum AI by modifying some outputs of G_n.

Let $\{X_1, \ldots, X_{2^{n-1}}\}$ and $\{Y_1, \ldots, Y_{2^{n-1}}\}$ be the support of G_n and $1 \oplus G_n$ respectively. Suppose $X^j = \{X_{j_1}, \ldots, X_{j_k}\}$ and $Y^i = \{Y_{i_1}, \ldots, Y_{i_k}\}$. Construct the function F_n as

$$F_n(X) = \begin{cases} 1 \oplus G_n(X), \text{ if } X \subset X^j \cup Y^i, \\ G_n(X), \text{ elsewhere.} \end{cases}$$

In rest of the paper, we denote an n-variable Boolean function constructed as above by F_n. The next result follows from Proposition 2.

Proposition 3. *The function F_n has maximum* AI *if and only if the two k-sets X^j and Y^i be such that $M_{n,\lceil \frac{n}{2} \rceil - 1}(F_n)$ is nonsingular.*

This idea was first proposed in [5] and using this idea, a few examples of Boolean functions on odd variables with maximum AI have been demonstrated in [9]. However, this has not been studied in the domain of RSBFs.

Let's have a quick look at a result from linear algebra which is a consequence of the Steinitz Exchange Lemma [8].

Theorem 1. *Let V be a vector space over the field F of dimension τ and $\{\alpha_1, \ldots, \alpha_\tau\}$ and $\{\beta_1, \ldots, \beta_\tau\}$ are two bases of V. Then for any k $(1 \leq k \leq \tau)$, there will be a pair of k-sets $\{\beta_{a_1}, \ldots, \beta_{a_k}\}$ and $\{\alpha_{b_1}, \ldots, \alpha_{b_k}\}$ such that the set $\{\alpha_1, \ldots, \alpha_\tau\} \cup \{\beta_{a_1}, \ldots, \beta_{a_k}\} \setminus \{\alpha_{b_1}, \ldots, \alpha_{b_k}\}$ will be a basis of V.*

The row vectors $v_{n,\lfloor \frac{n}{2} \rfloor}(X_1), \ldots, v_{n,\lfloor \frac{n}{2} \rfloor}(X_{2^{n-1}})$ of $M_{n,\lfloor \frac{n}{2} \rfloor}(G_n)$ form a basis of the vector space $V_{2^{n-1}}$. Similarly the row vectors $v_{n,\lfloor \frac{n}{2} \rfloor}(Y_1), \ldots, v_{n,\lfloor \frac{n}{2} \rfloor}(Y_{2^{n-1}})$ of $M_{n,\lfloor \frac{n}{2} \rfloor}(1 \oplus G_n)$ also form a basis of the vector space $V_{2^{n-1}}$. By finding two k-sets $\{v_{n,\lfloor \frac{n}{2} \rfloor}(X_{j_1}), \ldots, v_{n,\lfloor \frac{n}{2} \rfloor}(X_{j_k})\}$ and $\{v_{n,\lfloor \frac{n}{2} \rfloor}(Y_{i_1}), \ldots, v_{n,\lfloor \frac{n}{2} \rfloor}(Y_{i_k})\}$ (which always exist by Theorem 1), one can construct an n-variable Boolean function F_n with maximum algebraic immunity if and only if the corresponding matrix $M_{n,\lfloor \frac{n}{2} \rfloor}(F_n)$ is nonsingular. Complexity of checking the nonsingularity of the matrix $M_{n,\lfloor \frac{n}{2} \rfloor}(F_n)$ is $O((\sum_{t=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{t})^3)$, i.e., this construction will take huge time for larger n. But this task can be done with lesser effort by forming a matrix, $W = M_{n,\lfloor \frac{n}{2} \rfloor}(1 \oplus G_n) \times (M_{n,\lfloor \frac{n}{2} \rfloor}(G_n))^{-1}$ and checking a sub matrix of it. Since $(M_{n,\lfloor \frac{n}{2} \rfloor}(G_n))^{-1} = M_{n,\lfloor \frac{n}{2} \rfloor}(G_n)$, then $W = M_{n,\lfloor \frac{n}{2} \rfloor}(1 \oplus G_n) \times M_{n,\lfloor \frac{n}{2} \rfloor}(G_n)$. We have the following proposition.

Proposition 4. *[5] Let A be a nonsingular $m \times m$ binary matrix where the row vectors are denoted as $v_1, \ldots, v_m$. Let U be a $k \times m$ matrix, $k \leq m$, where the vectors are denoted as $u_1, \ldots, u_k$. Let $Z = UA^{-1}$, be a $k \times m$ binary matrix.*

Consider that a matrix A' is formed from A by replacing the rows $v_{i_1}, \ldots, v_{i_k}$ of A by the vectors $u_1, \ldots, u_k$. Further consider the $k \times k$ matrix Z' is formed by taking the j_1-th, j_2-th, ..., j_k-th columns of Z. Then A' is nonsingular if and only if Z' is nonsingular.

From the construction of F_n it is clear that it is balanced. Now construct the matrix $W = M_{n,\lfloor \frac{n}{2} \rfloor}(1 \oplus G_n) \times M_{n,\lfloor \frac{n}{2} \rfloor}(G_n)$. Consider A to be the matrix $M_{n,\lfloor \frac{n}{2} \rfloor}(G_n)$ and let U be the matrix formed by i_1-th, ..., i_k-th rows of $M_{n,\lfloor \frac{n}{2} \rfloor}(1 \oplus G_n)$ which are the row vectors $v_{n,\lfloor \frac{n}{2} \rfloor}(Y_{i_1}), \ldots, v_{n,\lfloor \frac{n}{2} \rfloor}(Y_{i_k})$ respectively. Now replace the j_1-th, ..., j_k-th rows of $M_{n,\lfloor \frac{n}{2} \rfloor}(G_n)$ which are respectively the row vectors $v_{n,\lfloor \frac{n}{2} \rfloor}(X_{j_1}), \ldots, v_{n,\lfloor \frac{n}{2} \rfloor}(X_{j_k})$ by the rows of U and form the new matrix A'. Note that A' is exactly the $M_{n,\lfloor \frac{n}{2} \rfloor}(F_n)$ matrix. Let $W_{|Y^i| \times |X^j|}$ be the matrix formed by taking i_1-th, ..., i_k-th rows and j_1-th, ..., j_k-th columns of W. Then $M_{n,\lfloor \frac{n}{2} \rfloor}(F_n)$ is nonsingular if and only if $W_{|Y^i| \times |X^j|}$ is nonsingular. Thus we have the following theorem.

Theorem 2. *The function F_n has maximum algebraic immunity if and only if the sub matrix $W_{|Y^i| \times |X^j|}$ is nonsingular.*

The following proposition characterizes W.

Proposition 5. *[5] The (q,p)-th element of the matrix W is given by $W_{(q,p)} = 0$ if $WS(X_p) \not\subseteq WS(Y_q)$ and $W_{(q,p)} = \sum_{t=0}^{\lfloor \frac{n}{2} \rfloor - wt(X_p)} \binom{wt(Y_q) - wt(X_p)}{t} \bmod 2$, otherwise; where $WS((x_1, \ldots, x_n)) = \{i : x_i = 1\} \subseteq \{1, \ldots, n\}$.*

3 New Class of RSBFs with Maximum AI

Proposition 6. *Given odd n, all the orbits O_μ generated by $\mu = (\mu_1, \ldots, \mu_n) \in V_n$ of weight $\lfloor \frac{n}{2} \rfloor$ or $\lceil \frac{n}{2} \rceil$ have n elements.*

Proof. From [12], it is known that if $gcd(n, wt(\mu)) = 1$, then the orbit O_μ contains n elements. Since $gcd(n, \lfloor \frac{n}{2} \rfloor) = gcd(n, \lceil \frac{n}{2} \rceil) = 1$, the result follows. □

Construction 1

1. *Take odd $n \geq 5$.*
2. *Take an element $x \in V_n$ of weight $\lfloor \frac{n}{2} \rfloor$ and generate the orbit O_x.*
3. *Choose an orbit O_y by an element $y \in V_n$ of weight $\lceil \frac{n}{2} \rceil$ such that*

   for each $x' \in O_x$ there is a unique $y' \in O_y$ where $WS(x') \subset WS(y')$.

4. *Construct*

$$R_n(X) = \begin{cases} G_n(X) \oplus 1, & \text{if } X \in O_x \cup O_y, \\ G_n(X), & \text{elsewhere .} \end{cases}$$

Henceforth, we will consider R_n as the function on n (≥ 5 and odd) variables obtained from Construction 1. We have the following theorem.

Theorem 3. *The function R_n is an n-variable RSBF with maximum* AI.

Proof. R_n is obtained by toggling all outputs of G_n corresponding to the inputs belonging to the two orbits O_x and O_y. Therefore R_n is an RSBF on n variables. By Proposition 6, we have $|O_x| = |O_y|$. Also it is clear that $G_n(X) = 1$ for all $X \in O_x$ and $G_n(X) = 0$ for all $X \in O_y$. So $wt(R_n) = 2^{n-1} - |O_x| + |O_y| = 2^{n-1}$. Thus R_n is a balanced RSBF on n-variables.

Let us now investigate the matrix $W_{|O_y|\times|O_x|}$. We reorder the elements in O_x and O_y as $x^{(1)}, \ldots, x^{(|O_x|)}$ and $y^{(1)}, \ldots, y^{(|O_y|)}$ respectively where $WS(x^{(p)}) \subset WS(y^{(p)})$, for all $1 \leq p \leq |O_x| = |O_y|$. As $WS(x^{(p)}) \not\subseteq WS(y^{(q)})$ for all $q \in \{1, \ldots, |O_y|\} \setminus \{p\}$, then by Proposition 5, the value of $W_{(q,p)} = 0$, for all $q \in \{1, \ldots, |O_y|\} \setminus \{p\}$. Again by Proposition 5, the value of $W_{(p,p)}$ can be determined as $W_{(p,p)} = \sum_{t=0}^{\lfloor \frac{n}{2} \rfloor - wt(x^{(p)})} \binom{wt(y^{(p)}) - wt(x^{(p)})}{t} = \sum_{t=0}^{\lfloor \frac{n}{2} \rfloor - \lfloor \frac{n}{2} \rfloor} \binom{\lceil \frac{n}{2} \rceil - \lfloor \frac{n}{2} \rfloor}{t} = 1$. Thus the matrix $W_{|O_y|\times|O_x|}$ is a diagonal matrix where all the diagonal elements are all equal to 1. Hence $W_{|O_y|\times|O_x|}$ is nonsingular. Therefore Theorem 2 implies that R_n has maximum AI. □

Example 1. Take $n = 5$. Consider $x = (1, 0, 0, 1, 0)$ and $y = (1, 0, 0, 1, 1)$ and generate the orbits

$O_x = \{(1, 0, 0, 1, 0), (0, 1, 0, 0, 1), (1, 0, 1, 0, 0), (0, 1, 0, 1, 0), (0, 0, 1, 0, 1)\}$ and
$O_y = \{(1, 0, 0, 1, 1), (1, 1, 0, 0, 1), (1, 1, 1, 0, 0), (0, 1, 1, 1, 0), (0, 0, 1, 1, 1)\}$.

Here, for each $x' \in O_x$, there is a unique $y' \in O_y$ such that $WS(x') \subset WS(y')$. Therefore, by Theorem 3, the function

$$R_n(X) = \begin{cases} G_n(X) \oplus 1, \text{ if } X \in O_x \cup O_y, \\ G_n(X), \text{ elsewhere }, \end{cases}$$

is a 5-variable RSBF with maximum AI which is 3.

It is known [11] that for an n (odd) variable Boolean function f with maximum AI, we have $nl(f) \geq 2^{n-1} - \binom{n-1}{\lfloor \frac{n}{2} \rfloor}$. Therefore nonlinearity of the function R_n will be at least $2^{n-1} - \binom{n-1}{\lfloor \frac{n}{2} \rfloor}$. Let us now examine the exact nonlinearity of R_n.

Theorem 4. *The nonlinearity of the function R_n is $2^{n-1} - \binom{n-1}{\lfloor \frac{n}{2} \rfloor} + 2$.*

Proof. As per the assumptions of Construction 1, $n \geq 5$ and it is odd; and weights of the orbits O_x and O_y are respectively $\lfloor \frac{n}{2} \rfloor$ and $\lceil \frac{n}{2} \rceil$. Now G_n being a symmetric function, it is also RSBF. So R_n can be viewed as a function, which is obtained by toggling the outputs of the RSBF G_n corresponding to the orbit O_x and O_y. From [6], we know that $nl(G_n) = 2^{n-1} - \binom{n-1}{\lfloor \frac{n}{2} \rfloor}$. Also it is known that the maximum absolute Walsh spectrum value of G_n, i.e., $2\binom{n-1}{\lfloor \frac{n}{2} \rfloor}$ occurs at the inputs corresponding to the orbits of weight 1 and n. We denote an element of V_n by Λ^n. Note that when, $wt(\Lambda^n) = n$, the value of $W_{G_n}(\Lambda^n)$ is $-2\binom{n-1}{\lfloor \frac{n}{2} \rfloor}$ or $2\binom{n-1}{\lfloor \frac{n}{2} \rfloor}$ according as $\lfloor \frac{n}{2} \rfloor$ is even or odd, and for $wt(\Lambda^n) = 1$, $W_{G_n}(\Lambda^n) = -2\binom{n-1}{\lfloor \frac{n}{2} \rfloor}$.

Let us first find the relation between the values of $W_{R_n}(\Lambda^n)$ and $W_{G_n}(\Lambda^n)$.

$$\begin{aligned}
W_{R_n}(\Lambda^n) &= \sum_{\zeta\in V_n\setminus\{O_x\cup O_y\}} (-1)^{R_n(\zeta)}(-1)^{\zeta\cdot\Lambda^n} + \sum_{\zeta\in O_x} (-1)^{R_n(\zeta)}(-1)^{\zeta\cdot\Lambda^n} \\
&\quad + \sum_{\zeta\in O_y} (-1)^{R_n(\zeta)}(-1)^{\zeta\cdot\Lambda^n} \\
&= \sum_{\zeta\in V_n\setminus\{O_x\cup O_y\}} (-1)^{G_n(\zeta)}(-1)^{\zeta\cdot\Lambda^n} + \sum_{\zeta\in O_x} (-1)^{1\oplus G_n(\zeta)}(-1)^{\zeta\cdot\Lambda^n} \\
&\quad + \sum_{\zeta\in O_y} (-1)^{1\oplus G_n(\zeta)}(-1)^{\zeta\cdot\Lambda^n} \\
&= \sum_{\zeta\in V_n\setminus\{O_x\cup O_y\}} (-1)^{G_n(\zeta)}(-1)^{\zeta\cdot\Lambda^n} - \sum_{\zeta\in O_x} (-1)^{G_n(\zeta)}(-1)^{\zeta\cdot\Lambda^n} \\
&\quad - \sum_{\zeta\in O_y} (-1)^{G_n(\zeta)}(-1)^{\zeta\cdot\Lambda^n} \\
&= \sum_{\zeta\in V_n} (-1)^{G_n(\zeta)}(-1)^{\zeta\cdot\Lambda^n} - 2\sum_{\zeta\in O_x} (-1)^{1}(-1)^{\zeta\cdot\Lambda^n} - 2\sum_{\zeta\in O_y} (-1)^{0}(-1)^{\zeta\cdot\Lambda^n} \\
&= W_{G_n}(\Lambda^n) + 2\sum_{\zeta\in O_x} (-1)^{\zeta\cdot\Lambda^n} - 2\sum_{\zeta\in O_y} (-1)^{\zeta\cdot\Lambda^n} \qquad (3)
\end{aligned}$$

Consider that $wt(\Lambda^n) = 1$. It can be proved that for any two orbits O_μ and O_ν of weight $\lfloor\frac{n}{2}\rfloor$ and $\lceil\frac{n}{2}\rceil$ respectively, $\sum_{\zeta\in O_\mu}(-1)^{\zeta\cdot\Lambda} = 1$ and $\sum_{\zeta\in O_\nu}(-1)^{\zeta\cdot\Lambda} = -1$. Thus $\sum_{\zeta\in O_x}(-1)^{\zeta\cdot\Lambda} = 1$ and $\sum_{\zeta\in O_y}(-1)^{\zeta\cdot\Lambda} = -1$. Therefore from Equation 3 we get, $W_{R_n}(\Lambda^n) = -2\binom{n-1}{\lfloor\frac{n}{2}\rfloor} + 4$.

Let us now check the Walsh spectrum value $W_{R_n}(\Lambda^n)$ for $wt(\Lambda^n) = n$. We do it in the following two cases.

CASE I : $\lfloor\frac{n}{2}\rfloor$ is even.

We have, $\sum_{\zeta\in O_x}(-1)^{\zeta\cdot\Lambda^n} = |O_x| = n$, since $\zeta\cdot\Lambda^n$ is $\lfloor\frac{n}{2}\rfloor$ which is even. Again for $\zeta\in O_y$, we have, $\zeta\cdot\Lambda^n = \lceil\frac{n}{2}\rceil$ which is odd, so $\sum_{\zeta\in O_y}(-1)^{\zeta\cdot\Lambda^n} = |O_y| = -n$. Therefore from Equation 3, we get $W_{R_n}(\Lambda^n) = -2\binom{n-1}{\lfloor\frac{n}{2}\rfloor}+2n+2n = -2\binom{n-1}{\lfloor\frac{n}{2}\rfloor}+4n$.

CASE II : $\lfloor\frac{n}{2}\rfloor$ is odd.

Using the similar argument as applied in the previous case, we can show that $\sum_{\zeta\in O_x}(-1)^{\zeta\cdot\Lambda^n} = -n$ and $\sum_{\zeta\in O_y}(-1)^{\zeta\cdot\Lambda^n} = n$. Therefore from Equation 3, we get $W_{R_n}(\Lambda^n) = 2\binom{n-1}{\lfloor\frac{n}{2}\rfloor} - 2n - 2n = 2\binom{n-1}{\lfloor\frac{n}{2}\rfloor} - 4n$.

Note that $2\binom{n-1}{\lfloor\frac{n}{2}\rfloor} > 4n$, except for the case $n = 5$. Therefore for both of the cases and for $n \geq 7$, $|W_{R_n}(\Lambda^n)| = 2\binom{n-1}{\lfloor\frac{n}{2}\rfloor} - 4n$. Also $2\binom{n-1}{\lfloor\frac{n}{2}\rfloor} - 4n < 2\binom{n-1}{\lfloor\frac{n}{2}\rfloor} - 4$, for $n \geq 7$. This implies that $|W_{R_n}(\Lambda^n)| \leq |W_{R_n}(\Delta^n)|$ for $n \geq 7$, where $\Delta^n \in V_n$ is an input of weight 1. For $n = 5$, $2\binom{n-1}{\lfloor\frac{n}{2}\rfloor} = 12$ and thus, $W_{R_n}(\Lambda^n) = -8 = W_{R_n}(\Delta^n)$. Therefore, $|W_{R_n}(\Lambda^n)| \leq |W_{R_n}(\Delta^n)|$ for all $n \geq 5$.

Let us check the Walsh spectrum values of R_n at the other inputs, i.e., except inputs of weight 1 and n. For $n \geq 7$, the second maximum absolute value in the Walsh spectrum of G_n occurs at the inputs of weight 3 and $n-2$. The exact value at weight 3 input is $\mathcal{C} = [\binom{n-3}{\frac{n-1}{2}} - 2\binom{n-3}{\frac{n-1}{2}-1} + \binom{n-3}{\frac{n-1}{2}-2}]$, whereas at the input of weight $n-2$, the exact value is $\mathcal{C}$ when $\lfloor\frac{n}{2}\rfloor$ is even and it is $-\mathcal{C}$ when $\lfloor\frac{n}{2}\rfloor$ is odd. Equation 3 implies that when $wt(\Lambda^n) = 3$ or $n-2$, $|W_{R_n}(\Lambda^n)|$ can attain

value maximum up to $|W_{G_n}(\Lambda^n)| + 4n$, i.e., $\binom{n-3}{\frac{n-1}{2}} - 2\binom{n-3}{\frac{n-1}{2}-1} + \binom{n-3}{\frac{n-1}{2}-2} + 4n$. But it is clear that, $\binom{n-3}{\frac{n-1}{2}} - 2\binom{n-3}{\frac{n-1}{2}-1} + \binom{n-3}{\frac{n-1}{2}-2} + 4n \leq 2\binom{n-1}{\lfloor \frac{n}{2} \rfloor} - 4 = |W_{R_n}(\Delta^n)|$.

For $n = 5$, it can be verified that for any choice of a pair of orbits O_x and O_y assumed in Construction 1, the absolute Walsh spectrum value of R_n, for all the inputs Λ^n of weight 3 is 8 which is equal to $|W_{R_n}(\Delta^n)|$.

Therefore, for all $n \geq 5$, maximum absolute Walsh Spectrum value of R_n is $2\binom{n-1}{\lfloor \frac{n}{2} \rfloor} - 4$. Hence, $nl(R_n) = 2^{n-1} - \binom{n-1}{\lfloor \frac{n}{2} \rfloor} + 2$. □

4 Generalization of Construction 1

Construction 2. *Take orbits $O_{z_1}, \ldots, O_{z_k}$ with $G_n(z_i) = 1$, for $z_i \in V_n, 1 \leq i \leq k$ and $O_{w_1}, \ldots, O_{w_l}$ with $G_n(w_i) = 0$ for $w_i \in V_n, 1 \leq i \leq l$. Assume that,*

1. $\sum_{t=0}^{k} |O_{z_t}| = \sum_{t=0}^{l} |O_{w_t}|$.
2. *for each $x' \in \cup_{t=0}^{k} O_{z_t}$ there is a unique $y' \in \cup_{t=0}^{l} O_{w_t}$ s.t. $WS(x') \subset WS(y')$.*
3. $\sum_{t=0}^{\lfloor \frac{n}{2} \rfloor - wt(x')} \binom{wt(y') - wt(x')}{t}$ *is odd, for any $x' \in \cup_{t=0}^{k} O_{z_t}$ and corresponding $y' \cup_{t=0}^{l} O_{w_t}$ such that $WS(x') \subset WS(y')$. Then construct,*

$$R'_n(X) = \begin{cases} G_n(X) \oplus 1, & \text{if } X \in \{\cup_{t=0}^{k} O_{z_t}\} \bigcup \{\cup_{t=0}^{l} O_{w_t}\} \\ G_n(X), & \text{elsewhere .} \end{cases}$$

Theorem 5. *The function R'_n is an n-variable RSBF with maximum* AI.

Proof. Following the same argument as used in Theorem 3 we can prove that $W_{|\cup_{t=0}^{k} O_{z_t}| \times |\cup_{t=0}^{l} O_{w_t}|}$ is a diagonal matrix whose diagonal elements are all equal to 1, i.e., it is nonsingular. Hence the proof. □

Example 2. Take $n = 7$. Consider $z_1 = (0,0,0,1,1,0,1), z_2 = (0,0,1,0,1,0,1)$ and $w_1 = (0,0,0,1,1,1,1), w_2 = (0,0,1,0,1,1,1)$ and generate the orbits

$$O_{z_1} = \{(0,0,0,1,1,0,1), (0,0,1,1,0,1,0), (0,1,1,0,1,0,0), (1,1,0,1,0,0,0), (1,0,1,0,0,0,1), (0,1,0,0,0,1,1), (1,0,0,0,1,1,0)\};$$

$$O_{z_2} = \{(0,0,1,0,1,0,1), (0,1,0,1,0,1,0), (1,0,1,0,1,0,0), (0,1,0,1,0,0,1), (1,0,1,0,0,1,0), (0,1,0,0,1,0,1), (1,0,0,1,0,1,0)\};$$

$$O_{w_1} = \{(0,0,0,1,1,1,1), (0,0,1,1,1,1,0), (0,1,1,1,1,0,0), (1,1,1,1,0,0,0), (1,1,0,0,0,0,1), (1,1,0,0,0,1,1), (1,0,0,0,1,1,1)\};$$

$$O_{w_2} = \{(0,0,1,0,1,1,1), (0,1,0,1,1,1,0), (1,0,1,1,1,0,0), (0,1,1,1,0,0,1), (1,1,1,0,0,1,0), (1,1,0,0,1,0,1), (1,0,0,1,0,1,1)\}.$$

Here for each $x' \in O_{z_1} \cup O_{z_2}$, there exists a unique $y' \in O_{w_1} \cup O_{w_2}$ such that $WS(x') \subset WS(y')$ and $\sum_{t=0}^{\lfloor \frac{n}{2} \rfloor - wt(x')} \binom{wt(y') - wt(x')}{t}$ is odd. Then construct,

$$R'_n(X) = \begin{cases} G_n(X) \oplus 1, & \text{if } X \in \{O_{z_1} \cup O_{z_2}\} \bigcup \{O_{w_1} \cup O_{w_2}\} \\ G_n(X), & \text{elsewhere .} \end{cases}$$

Then by Theorem 5, R'_n is an 7-variable RSBF with maximum AI which is 4.

As in Construction 2, outputs of G_n are toggled at more inputs, one can expect better nonlinearity than the Construction 1.

For 7-variable functions with maximum AI 3, the lower bound on nonlinearity is 44 [11] and that is exactly achieved in the existing theoretical construction [4,6]. Our Construction 1 provides the nonlinearity 46. Further we used Construction 2 to get all possible functions R'_n and they provide the nonlinearity 48.

4.1 Further Generalization

Construction 3. *Take $n \geq 5$ and odd. Consider the orbits $O_{z_1}, \ldots, O_{z_k}$ and $O_{w_1}, \ldots, O_{w_k}$ such that the sub matrix $W_{|\cup_{t=0}^{k} O_{z_t}| \times |\cup_{t=0}^{l} O_{w_t}|}$ is nonsingular. Then construct,*

$$R''_n(X) = \begin{cases} G_n(X) \oplus 1, & \text{if } X \in \{\cup_{t=0}^{k} O_{z_t}\} \bigcup \{\cup_{t=0}^{l} O_{w_t}\} \\ G_n(X), & \text{elsewhere .} \end{cases}$$

Clearly, the function R''_n is an n-variable RSBF with maximum AI. Construction 3 will provide all the RSBFs with maximum AI. In this case we need a heuristic to search through the space of RSBFs with maximum AI as the exhaustive search may not be possible as number of input variables n increases. One may note that it is possible to use these techniques to search through the space of general Boolean functions, but that space is much larger (2^{2^n}) compared to the space of RSBFs ($\approx 2^{\frac{2^n}{n}}$) and getting high nonlinearity after a small amount of search using a heuristic is not expected. We present a simple form of heuristic as follows that we run for several iterations.

1. Start with an RSBF n having maximum AI using Construction 1.
2. Choose two orbits of same sizes having different output values and toggle the outputs corresponding to both the orbits (this is to keep the function balanced).
3. If the modified function is of maximum AI and having better nonlinearity than the previous ones, then we store that as the best function.

By this heuristic, we achieve 7, 9, 11 variable RSBFs with maximum possible AI having nonlinearities 56, 240, 984 respectively with very small amount of search. Note that these nonlinearities are either equal or close to $2^{n-1} - 2^{\frac{n-1}{2}}$.

5 Conclusion

In this paper, we present the construction (Construction 1) of Rotation Symmetric Boolean functions on $n \geq 5$ (odd) variables with maximum possible algebraic immunity. We determine the nonlinearity of the RSBFs constructed in Construction 1 and find that the nonlinearity is 2 more than the lower bound of nonlinearity of n (odd) variable Boolean functions with maximum algebraic immunity. Prior to our construction, the existing theoretical constructions could

achieve only the lower bound. We also included little amount of search with the construction method to get RSBFs having maximum possible AI and very high nonlinearity. With minor modifications, our method will work for RSBFs on even number of variables. This will be available in the full version of this paper.

References

1. Armknecht, F., Carlet, C., Gaborit, P., Kuenzli, S., Meier, W., Ruatta, O.: Efficient computation of algebraic immunity for algebraic and fast algebraic attacks. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, Springer, Heidelberg (2006)
2. Courtois, N., Meier, W.: Algebraic Attacks on Stream Ciphers with Linear Feedback. In: Biham, E. (ed.) EUROCRPYT 2003. LNCS, vol. 2656, pp. 345–359. Springer, Heidelberg (2003)
3. Dalai, D.K., Gupta, K.C., Maitra, S.: Results on Algebraic Immunity for Cryptographically Significant Boolean Functions. In: Canteaut, A., Viswanathan, K. (eds.) INDOCRYPT 2004. LNCS, vol. 3348, pp. 92–106. Springer, Heidelberg (2004)
4. Dalai, D.K., Gupta, K.C., Maitra, S.: Cryptographically Significant Boolean functions: Construction and Analysis in terms of Algebraic Immunity. In: Gilbert, H., Handschuh, H. (eds.) FSE 2005. LNCS, vol. 3557, pp. 98–111. Springer, Heidelberg (2005)
5. Dalai, D.K., Maitra, S.: Reducing the Number of Homogeneous Linear Equations in Finding Annihilators. In: Gong, G., Helleseth, T., Song, H.-Y., Yang, K. (eds.) SETA 2006. LNCS, vol. 4086, pp. 376–390. Springer, Heidelberg (2006)
6. Dalai, D.K., Maitra, S., Sarkar, S.: Basic Theory in Construction of Boolean Functions with Maximum Possible Annihilator Immunity. Design, Codes and Cryptography 40(1), 41–58 (2006)
7. Kavut, S., Maitra, S., Sarkar, S., Yücel, M.D.: Enumeration of 9-variable Rotation Symmetric Boolean Functions Having Nonlinearity > 240. In: Barua, R., Lange, T. (eds.) INDOCRYPT 2006. LNCS, vol. 4329, pp. 266–279. Springer, Heidelberg (2006)
8. Kurosh, A.G.: Theory of Groups. Chelsea Publishing Co., New York (1955)
9. Li, N., Qi, W.F.: Construction and Analysis of Boolean Functions of $2t+1$ Variables With Maximum Algebraic Immunity. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 84–98. Springer, Heidelberg (2006)
10. Li, N., Qi, W.F.: Symmetric Boolean functions Depending on an Odd Number of Variables with Maximum Algebraic Immunity. IEEE Trans. Inform. Theory 52(5), 2271–2273 (2006)
11. Lobanov, M.: Tight Bound Between Nonlinearity and Algebraic Immunity. Cryptology ePrint Archive no. 2005/441 (2005)
12. Stănică, P., Maitra, S.: Rotation Symmetric Boolean Functions – Count and Cryptographic Properties. Discrete Applied mathematics (to be published), http://dx.doi.org/10.1016/j.dam.2007.04.029
13. Qu, L., Li, C., Feng, K.: A Note on Symmetric Boolean Functions with Maximum Algebraic Immunity in Odd Number of Variables. IEEE Trans. Inform. Theory 53(8), 2908–2910 (2007)

A Path to Hadamard Matrices

P. Embury and A. Rao

School of Mathematics and Geospatial Sciences, RMIT University, Melbourne, Australia
s7603100@student.rmit.edu.au, asha@rmit.edu.au

Abstract. There are characteristics of Hadamard matrices that enable an exhaustive search using algorithmic techniques. The search derives primarily from the eigenvalues which are constant after the Hadamard matrix is multiplied by its transpose. Generally this would be a performance concern but there are additional properties that enable the eigenvalues to be predicted. Here an algorithm is given to obtain a Hadamard matrix from a matrix of 1s using optimisation techniques on a row-by-row basis.

Keywords: Hadamard Matrices, eigen values, optimization.

1 Introduction

Hadamard Matrices are named after Jacques Hadamard (1865-1963) and are square matrices of order 1, 2, 4n, n = $1 \ldots \infty$, whose entries are on the unit circle, and whose rows are all orthogonal to each other. That is,

$$H_n H_n^T = nI \tag{1}$$

where n is the order of matrix.

Hadamard Matrices currently are found using construction techniques which have been generally classified [3] as recursion theorems, "plug-in" methods or direct constructions. These are summarised in the following table:

Multiplicative or Recursive Techniques	Any Kronecker product of existing Hadamard matrices is itself a Hadamard matrix.
"Plug-in" methods	Named after James Sylvester, this is the earliest (1863) and simplest construction and is also based on the Kronecker product.
Direct Constructions	The Paley HMs are two constructions found using quadratic residues in a finite field.
Williamson	Also a "plug-in" algorithm where the HM is constructed from an existing HM plus 3 other HMs that satisfy specific conditions.

Full details of the construction methods mentioned above can be found in [3] while other methods can be found in [8] and are not included here. The important

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 281–290, 2007.
© Springer-Verlag Berlin Heidelberg 2007

aspect is that using any of the methods above, not all orders of Hadamard matrices have been verified that they can be created since they are constructed from matrices of lower orders. This is the long (since 1893) unsolved Hadamard conjecture. In addition, the nature of equivalence is not fully understood and only orders up to 28 have been fully explored, which has a known 427 equivalence classes.

Hadamard matrices are formed into equivalence classes where they are deemed to be equivalent if the following operations are performed on them to arrive at the same matrix:

1. Any permutation of the existing rows and/or permutation of the columns
2. The multiplication of row(s) or column(s) by -1.

More recently [6] has defined Q-classes to enable more manageable categorisation of the equivalence classes by "extracting" common sub-matrices or substructures of Hadamard matrices and classifying those matrices with these common sub-matrices as Q-equivalent.

2 Background

Instead of relying on construction techniques to construct, and sheer brute force to test for, Hadamard matrices, there are characteristics that enable Hadamard matrices be searched for using optimisation methods [4,7]. These characteristics also provide clues to the nature of Hadamard equivalence classes.

There are two properties of matrices that offer the opportunity to search for the existence of Hadamard matrices using optimisation techniques: the determinant and the eigenvectors and eigenvalues. These are, of course related measures.

The determinant for a Hadamard matrix is given by

$$detH = \pm n^{n/2}. \tag{2}$$

But one of the drawbacks in using the determinant is that it has sharp peaks and exhibits properties analogous to the Hamming cliffs encountered when using optimisation methods with binary strings as distinct from gray encoded strings. This primarily arises from the fact that the determinant is 0 when it becomes *singular* which can occur as soon as one row becomes identical to another. Also, there are multiple matrices that have the same determinant with little or no indication as to the direction a search could or should progress.

The eigenvectors of a Hadamard matrix all lie on the unit circle and the absolute value of the eigenvalues are constant and equal to n, where n is the order of the Hadamard matrix. Since the eigenvectors can be complex and half the eigenvalues are negative, it is easier to consider the RHS of (1) and the effects on it when looking at the optimisation path.

The question that arises is: what is the sensitivity of the eigenvalues to errors that would enable its use in optimisation?

It will be shown that the variance of the eigenvalues (EVV) of the RHS of (1) can provide a suitable optimisation path since:

1. The intra-row errors describe an EVV path that is represented by the parabolic equation:
$$8x\left(1-\frac{x}{N}\right) \tag{3}$$
where x is the number of cells within a row that are inverted and N is the order of the matrix. The variance of the eigenvalues is calculated as:
$$EVV=\sum_{i=1}^{n}\left(\frac{(\lambda_i-E\lambda)^2}{n}\right) \tag{4}$$
Note: $E\lambda=n$
2. The inter-row errors are described by the recursive equation:
$$EVV_x=\frac{y\,(y-1)}{2}M \tag{5}$$
where y is the number of non-orthogonal rows and M is the maximum **EVV** for each row which occurs where $x=N/2$ in (3). Eg, for order 8, $M=16$ and EVV for an entire order 8 matrix of ones = 448.

This information can be used to devise an optimisation path to a Hadamard matrix from a unit matrix since at any stage it is known how many steps a non-Hadamard matrix is from a Hadamard matrix.

It is inescapable that any algorithm that actually uses eigenvalues is not going to be very efficient. Hence there needs to be an improvement to any algorithm that does use the eigenvalues method that enables the eigenvalues to be predicted. Fortunately, for row-by-row optimisation, there are two criteria or tests that can be used to assist. This will be developed in the next section.

3 The Search for a Path

Consider the normalised Hadamard matrix H_4 of order 4 given by

$$H_4=\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$$

and the Hadamard matrix of order 8 H_8 given by:

$$\begin{bmatrix}H_4 & H_4\\ H_4 & -H_4\end{bmatrix}$$

As successive errors are introduced to a row, the eigenvalues (of the RHS of (1)) are affected according to the following tables and the variance follows an inverted parabolic path.

Number of cells inverted	0	1	2	3	4
Eigenvalues	4	0.5359	0	0.5359	4
	4	4	4	4	4
	4	4	4	4	4
	4	7.4641	8	7.4641	4
Variance (EVV)	0	6	8	6	0

Number of cells inverted	0	1	2	3	4	5	6	7	8
Eigenvalues	8	2.7085	1.0718	.2540	0	.2540	1.0718	2.7085	8
	8	8	8	8	8	8	8	8	8
	8	8	8	8	8	8	8	8	8
	8	8	8	8	8	8	8	8	8
	8	8	8	8	8	8	8	8	8
	8	8	8	8	8	8	8	8	8
	8	8	8	8	8	8	8	8	8
	8	13.2915	14.9282	15.7460	16	15.7460	14.9282	13.2915	8
Variance (EVV)	0	7	12	15	16	15	12	7	0

By the time all the entries in a row are inverted we have arrived at another (but equivalent) Hadamard matrix. Not only do the eigenvalues follow a predicable path, there is an indication of the number of rows that are not orthogonal to each other.

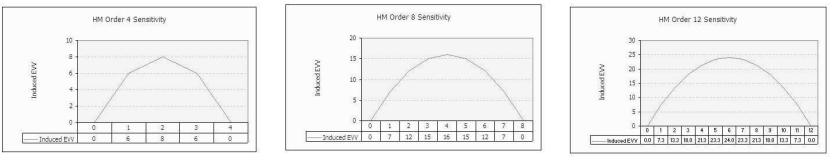


Fig. 1. Orders 4, 8 and 12 EVV as successive cells are inverted

What if there are errors in more than one row? In a simple situation where there is a single error in another row, and the particular column entries of the two rows have the same sign, then the errors aggregate, not only in that column, but also in other columns that have the same sign. When the column entries have opposite signs, they tend to compensate. (Figure 2)

But it is not so simple and with multiple errors, the situation needs to be viewed of sub-matrices of order 2 that include the respective rows. The EVV induced by errors on odd-weighted sub-matrices is double that of even weighted sub-matrices.

What if we overwrite two rows with ones within a Hadamard matrix (besides the first row)? Three rows, four rows....? What is the total EVV that can be found for a matrix consisting only of +1 in a matrix of a given order? The results are in the following table for those of order 8.

The total EVV of a square matrix of ones, (eg for order 8 is 448) which is the furthest a matrix can digress from Hadamard, can be broken down and it

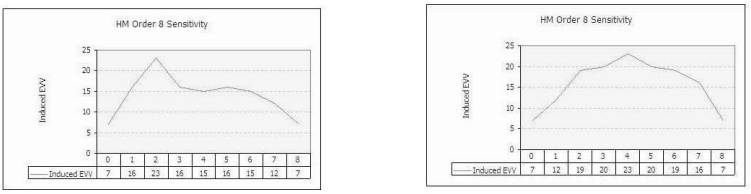


Fig. 2. Order 8 with same sign and opposite sign in another row as successive cells are inverted

Rows Non-orthogonal	EVV
1	0
2	16
3	48
4	96
5	160
6	240
7	336
8	448

Fig. 3. Successive EVV as more rows non-orthogonal

is possible to determine how many rows are not orthogonal to each other. In other words, every Hadamard matrix of any order must comply with (5) since replacing successive rows will automatically induce a known level of EVV until it reaches a maximum.

Putting the aforementioned characteristics into practice can be investigated in two separate ways.

1. The first method is to consider the matrix as a whole and successively test individual cells as to the effect on the EVV. In other words, each cell has a marginal contribution to the total EVV of the matrix. At each iteration, the entire matrix can be searched for the cell that when inverted, gives the greatest reduction in EVV, and then invert that cell. This will not always give a complete path to a Hadamard matrix since there are other criteria elaborated below that also need to be satisfied. Also, given that the eigenvalues are continually being calculated, this method is only suitable for matrices of smaller orders.
2. A second method is to optimise to a Hadamard matrix from a matrix of ones row-by-row. Exactly the opposite to successively substituting rows of a Hadamard matrix by rows of ones. The objective is to find which combination of cells within each row needs to be inverted to fit the expected EVV from (5).

By way of an example for the second method, start with a matrix of ones of the desired order, say 8. (Assume the the matrix will be normalised whereby

there will be exactly 4 -1's in every row or column except the first row and column which will be all 1's.)

1. Start with a matrix of ones.
2. Leave the first row and fill half the second row with -1s. Leave the first column so that normalisation is maintained.
3. Calculate the "Expected EVV" contributed by this row. That is, if this row is orthogonal to all the preceding rows, what is the EVV by all the subsequent rows. For the fourth row this is (using (3) where $N = 8$ and $x = 4$ and then (5) where $y = 5$): $5 * 4/2 * 16 = 160$.
4. Determine all the combinations of columns $2 \ldots 8$ in groups of 4 and calculate the new EVV if they were chosen. If the EVV matches the Expected EVV, then this combination forms part of a Hadamard matrix. Use it and move onto the next row. It needs to borne that there are

$$\binom{N-1}{\frac{n}{2}}$$

possible combinations for each row.

Working through an example, in figure 4 we have the situation where rows 1, 2 and 3 are complete. To assist in clarification the terminology has been changed and we will for the purposes of this example use '.' for a 1 and an X for -1. Any new row will require the resultant matrix to have an **EVV** of 160 (using (5)).

$$\left[\begin{array}{cccccccc}
. & . & . & . & . & . & . & . \\
. & . & . & . & X & X & X & X \\
. & X & X & . & X & X & . & . \\
. & . & . & . & . & . & . & . \\
 & & \vdots & & & & & \\
\hline
0 & 1 & 1 & 0 & 2 & 2 & 1 & 1
\end{array}\right]$$

Fig. 4. 3 Completed Rows

Note that the last row in the table is the sum of the "weights" or number of Xs already in each of the columns.

Any of the 8 rows in figure 5 are valid possible insertions for the fourth row since if they are inserted, the resultant EVV is 160 and each row has exactly 2 columns that match the -1s already in rows 2 and 3. Only one of these does not include column 4 which stands to reason since there is only one way of matching up exactly two columns with the existing rows 2 & 3. (Also note that this row is the difference between rows 2 and 3. This characteristic may offer a method of taking shortcuts in the algorithm or possibly lead to clues about the nature of equivalence.)

$$\begin{bmatrix}
. & X & X & . & . & . & X & X \\
. & X & . & X & X & . & X & . \\
. & X & . & X & X & . & . & X \\
. & X & . & X & . & X & X & . \\
. & . & X & X & X & . & X & . \\
. & . & X & X & X & . & . & X \\
. & . & X & X & . & X & X & . \\
. & . & X & X & . & X & . & X
\end{bmatrix}$$

Fig. 5. Possibilities for row 4

Now compare these rows with rows in figure 6 which will be rejected because they do not have the desired EVV (These are only 6 out of the possible 26.)

Rows	EVV	Row 2 Corr.	Row 3 Corr.
. X X X . . X .	164	1	2
. . . X X X . X	164	2	3
. X X X X . . .	168	1	3
. . X . X X . X	168	3	3
. X X . X X . .	176	4	0
. . . . X X X X	176	0	4

Fig. 6. Rejected Rows

The EVV column displays the resultant EVV of the matrix should that row be used as the new row 4. The last two columns show the difference between the candidate new row and the pre-existing rows 2 and 3 respectively. The last two candidate rows are the same as the rows already inserted and hence should be immediately rejected.

The important thing to note is that all the rows that result in the desired EVV of 160, have exactly $\left(\frac{n}{4}\right)$ matches between itself and all previous rows (not including the first row). In other words, for every possible column permutation for the next particular row, if there aren't $\left(\frac{n}{4}\right)$ matches then the combination can be rejected immediately.

The other noticeable facet concerns the number of all the Xs (see the bottom row of figure 4) or weights of the columns. Each successful combination or candidate row satisfies:

$$\sum Weights = (row - 2)\left(\frac{n}{4}\right) \tag{6}$$

This is another rule that can be used to fine-tune the algorithm. For example, the first successful candidate row tells us to insert Xs in columns 2, 3, 7 and 8. These four coulmns have weights (from figure (4)) of

$$1 + 1 + 1 + 1 = 4 = (4 - 2)\left(\frac{8}{4}\right) \tag{7}$$

If we assume that the second candidate row is selected (from figure 5) for row 4, what are the circumstances for the next iteration?

Figure 7 has the starting position and figure 8 has the new possible rows for row 5 (there are only 4 possibilities).

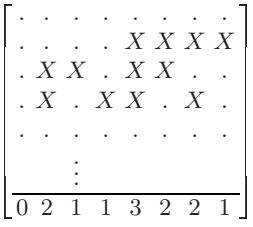


Fig. 7. 4 Completed Rows and new column weights

Rows	EVV	Row 2 Corr.	Row 3 Corr.	Row 4 Corr.
. X X . . . X X	96	2	2	2
. X . X . X . X	96	2	2	2
. . X X X . . X	96	2	2	2
. . X X . X X .	96	2	2	2

Fig. 8. Candidate Rows for Row 5

All these rows have:

- the new desired EVV of 96
- exactly 2 matches with every row preceding it except row 1
- the sum of column weights = 6 which satisfies (5)

In summary, there are 3 criteria to be satisfied for each row:

1. The desired EVV for each successive row needs to satsify $EVV = \frac{x(x-1)}{2}M$ where M is the maximum **EVV** for each row which occurs where $x = N/2$.
2. Each new row must have $\binom{n}{4}$ matches of X with every preceding row
3. The weights of each column must satisfy $\sum Weights = (row - 2)\binom{n}{4}$

Criteria 1 & 2 are synonymous while criterion 3 forms a type of a classical subset-sum problem [2] where the count of the weights subset is known and there are possible multiple solutions. The solutions are not necessarily known to exist.

3.1 An Incomplete Path

This is an example of when these methods will not allow us to proceed on our search for a Hadamard matrix. The following situation arises at the start of

optimising to an *order 12* matrix. After 3 rows we may have the following order 12 matrix:

$$\begin{bmatrix} . & . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & X & X & X & X & X & X \\ . & X & X & X & . & . & . & . & . & X & X & X \\ . & . & . & . & . & . & . & . & . & . & . & . \\ & & \vdots & & & & & & & & & \end{bmatrix}$$

The following rows are all possible for the fourth row (there are others)...

$$\begin{bmatrix} . & X & X & X & . & . & X & X & X & . & . & . \end{bmatrix}$$

$$\begin{bmatrix} . & X & X & . & X & . & X & . & . & X & X & . \end{bmatrix}$$

$$\begin{bmatrix} . & X & X & . & X & . & X & . & . & X & . & X \end{bmatrix}$$

$$\begin{bmatrix} . & X & X & . & X & . & . & X & . & X & X & . \end{bmatrix}$$

If the first row was selected then the situation could arise where we have...

$$\begin{bmatrix} . & . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & X & X & X & X & X & X \\ . & X & X & X & . & . & . & . & . & X & X & X \\ . & X & X & X & . & . & X & X & X & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . & . \\ & & \vdots & & & & & & & & & \end{bmatrix}$$

This is a situation which could occur but the problem is there is no further step. The EVV given by (5) is an expected 864 which satisfies criterion 1. Criterion 2 is satisfied since each row matches exactly 3 times against each previous row. However, criterion 3 fails, since the weights all = 2, and the required weights to proceed needs 6 columns to add up to 9 which is an impossibility.

In other words, an additional constraint is needed to ensure that the subsequent steps are valid which could lead to a large recursive tree for matrices of larger orders. This particular problem is easily circumvented by not choosing this particular row for row 4. This is the only exception encountered so far from other simulations on orders 8, 12, 16, 20 and 24 by randomly selecting any of the valid available rows.

4 Conclusions

The methods presented in this paper are only the start of possible alternatives using the eigenvalues and eigenvectors to search for Hadamard matrices. Can they be investigated further?

The Eigenvectors. How are they affected by disturbing a Hadamard matrix? Is there some clue in the direction they shift to that may help find the "way home" and may improve the algorithm?

Equivalence. At what stage of the traversal of the optimisation path is it decided whether a matrix belongs to another equivalence class or forms its own equivalence class?

The Determinant. Although this was discarded as a measure on its own, this may not necessarily be desirable because it is hard to discuss eigenvalues without acknowledging the determinant because of their direct relationship. Given that the calculation of the determinant is much more efficient to calculate for larger matrices, can it be used?

Although, the optimisation method is discussed as a "Hill-climb", the final solution forms a series of "Subset-sum" optimisation problems [2]. If these become too unwieldy, maybe alternatives as such "genetic algorithms" [1,5] could be used.

The one example found that led to an incomplete path described above is interesting because it may offer clues as to the existence of Hadamard matrices (or the non-existence).

References

1. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning, 1st edn. Addison-Wesley Longman Publishing Co., USA (1989)
2. Goodrich, M.T., Tamassia, R.: Algorithm Design: Foundations, Analysis, and Internet Examples. John Wiley and Sons, New York (2002)
3. Horadam, K.J.: Hadamard Matrices and Their Applications. Princeton University Press, Princeton, N.J. (2007)
4. Klima, R.E., Sigmon, N.P., Stitzinger, E.L.: Applications of Abstract Algebra with Maple^{TM} and Matlab®, 2nd edn. Chapman & Hall/CRC, Boca Raton (2006)
5. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs, 2nd edn. Springer, New York (1994)
6. Orrick, W.P.: Switching operations for Hadamard matrices (2007), `http://www.arxiv.org/abs/math.CO/0507515`
7. Snyman, J.A.: Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms. Springer, Cambridge, Massachusetts (2005)
8. Wallis, W.D., Street, A.P., Wallis, J.S.: Combinatorics: Room Squares, Sum-Free Sets, Hadmard Matrices. Springer, New York (1972)

The Tangent FFT*

Daniel J. Bernstein

Department of Mathematics, Statistics, and Computer Science (M/C 249)
University of Illinois at Chicago, Chicago, IL 60607–7045, USA
djb@cr.yp.to

Abstract. The split-radix FFT computes a size-n complex DFT, when n is a large power of 2, using just $4n \lg n - 6n + 8$ arithmetic operations on real numbers. This operation count was first announced in 1968, stood unchallenged for more than thirty years, and was widely believed to be best possible.

Recently James Van Buskirk posted software demonstrating that the split-radix FFT is *not* optimal. Van Buskirk's software computes a size-n complex DFT using only $(34/9 + o(1))n \lg n$ arithmetic operations on real numbers. There are now three papers attempting to explain the improvement from 4 to 34/9: Johnson and Frigo, *IEEE Transactions on Signal Processing*, 2007; Lundy and Van Buskirk, *Computing*, 2007; and this paper.

This paper presents the "tangent FFT," a straightforward in-place cache-friendly DFT algorithm having exactly the same operation counts as Van Buskirk's algorithm. This paper expresses the tangent FFT as a sequence of standard polynomial operations, and pinpoints how the tangent FFT saves time compared to the split-radix FFT. This description is helpful not only for understanding and analyzing Van Buskirk's improvement but also for minimizing the memory-access costs of the FFT.

Keywords: Tangent FFT, split-radix FFT, modified split-radix FFT, scaled odd tail, DFT; convolution,polynomial multiplication, algebraic complexity, communication complexity.

1 Introduction

Consider the problem of computing the size-n complex **DFT** ("discrete Fourier transform"), where n is a power of 2; i.e., evaluating an n-coefficient univariate complex polynomial f at all of the nth roots of 1. The input is a sequence of n complex numbers $f_0, f_1, \ldots, f_{n-1}$ representing the polynomial $f = f_0 + f_1 x + \cdots + f_{n-1} x^{n-1}$. The output is the sequence $f(1), f(\zeta_n), f(\zeta_n^2), \ldots, f(\zeta_n^{n-1})$ where $\zeta_n = \exp(2\pi i/n)$.

The size-n **FFT** ("fast Fourier transform") is a well-known algorithm to compute the size-n DFT using $(5+o(1))n \lg n$ arithmetic operations on real numbers. One can remember the coefficient 5 as half the total cost of a complex addition

* Permanent ID of this document: a9a77cef9a7b77f9b8b305e276d5fe25. Date of this document: 2007.09.19.

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 291–300, 2007.
© Springer-Verlag Berlin Heidelberg 2007

(2 real operations), a complex subtraction (2 real operations), and a complex multiplication (6 real operations).

The FFT was used for astronomical calculations by Gauss in 1805; see, e.g., [6, pages 308–310], published in 1866. It was reinvented and republished on several subsequent occasions and was finally popularized in 1965 by Cooley and Tukey in [2]. The advent of high-speed computers meant that users in the 1960s were trying to handle large values of n in a wide variety of applications and could see large benefits from the FFT.

The Cooley-Tukey paper spawned a torrent of FFT papers—showing, among other things, that Gauss had missed a trick. The original FFT is not the optimal way to compute the DFT. In 1968, Yavne stated that one could compute the DFT using only $(4+o(1))n \lg n$ arithmetic operations, specifically $4n \lg n - 6n + 8$ arithmetic operations (if $n \geq 2$), specifically $n \lg n - 3n + 4$ multiplications and $3n \lg n - 3n + 4$ additions; see [13, page 117]. Nobody, to my knowledge, has ever deciphered Yavne's description of his algorithm, but a comprehensible algorithm achieving exactly the same operation counts was introduced by Duhamel and Hollmann in [3], by Martens in [9], by Vetterli and Nussbaumer in [12], and by Stasinski (according to [4, page 263]). This algorithm is now called the **split-radix FFT**.

The operation count $4n \lg n - 6n + 8$ stood unchallenged for more than thirty years[1] and was frequently conjectured to be optimal. For example, [11, page 152] said that split-radix FFT algorithms did not have minimal multiplication counts but "have what seem to be the best compromise operation count." Here "compromise" refers to counting both additions and multiplications rather than merely counting multiplications.

In 2004, James Van Buskirk posted software that computed a size-64 DFT using fewer operations than the size-64 split-radix FFT. Van Buskirk then posted similar software handling arbitrary power-of-2 sizes using only $(34/9+o(1))n \lg n$ arithmetic operations. Of course, 34/9 is still in the same ballpark as 4 (and 5), but it is astonishing to see *any* improvement in such a widely studied, widely used algorithm, especially after 36 years of no improvements at all!

Contents of this paper. This paper gives a concise presentation of the **tangent FFT**, a straightforward in-place cache-friendly DFT algorithm having exactly the same operation counts as Van Buskirk's algorithm. This paper expresses the tangent FFT as a sequence of standard polynomial operations, and pinpoints how the tangent FFT saves time compared to the split-radix FFT. This description is helpful not only for understanding and analyzing Van Buskirk's improvement but also for minimizing the memory-access costs of the FFT.

[1] The 1998 paper [14] claimed that its "new fast Discrete Fourier Transform" was much faster than the split-radix FFT. For example, the paper claimed that its algorithm computed a size-16 real DFT with 22 additions and 10 multiplications by various sines and cosines. I spent half an hour with the paper, finding several blatant errors and no new ideas; in particular, Figure 1 of the paper had many more additions than the paper claimed. I pointed out the errors to the authors and have not received a satisfactory response.

There have been two journal papers this year—[8] by Lundy and Van Buskirk, and [7] by Johnson and Frigo—presenting more complicated algorithms with the same operation counts. Both algorithms can be transformed into in-place algorithms but incur heavier memory-access costs than the algorithm presented in this paper.

I chose the name "tangent FFT" in light of the essential role played by tangents as constants in the algorithm. The same name could be applied to all of the algorithms in this class. Lundy and Van Buskirk in [8] use the name "scaled odd tail," which I find less descriptive. Johnson and Frigo in [7] use the name "our new FFT ... our new algorithm ... our algorithm ... our modified algorithm" etc., which strikes me as suboptimal terminology; I have already seen three reports miscrediting Van Buskirk's 34/9 to Johnson and Frigo. All of the credit for these algorithms should be assigned to Van Buskirk, except in contexts where extra features such as simplicity and cache-friendliness play a role.

2 Review of the Original FFT

The remainder $f \bmod x^8 - 1$, where f is a univariate polynomial, determines the remainders $f \bmod x^4 - 1$ and $f \bmod x^4 + 1$. Specifically, if

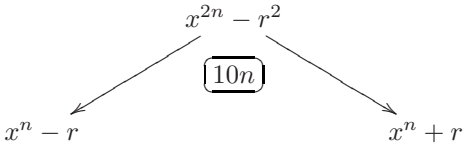
$$f \bmod x^8 - 1 = f_0 + f_1 x + f_2 x^2 + f_3 x^3 + f_4 x^4 + f_5 x^5 + f_6 x^6 + f_7 x^7,$$

then $f \bmod x^4 - 1 = (f_0 + f_4) + (f_1 + f_5)x + (f_2 + f_6)x^2 + (f_3 + f_7)x^3$ and $f \bmod x^4 + 1 = (f_0 - f_4) + (f_1 - f_5)x + (f_2 - f_6)x^2 + (f_3 - f_7)x^3$. Computing the coefficients $f_0 + f_4, f_1 + f_5, f_2 + f_6, f_3 + f_7, f_0 - f_4, f_1 - f_5, f_2 - f_6, f_3 - f_7$, given the coefficients $f_0, f_1, f_2, f_3, f_4, f_5, f_6, f_7$, involves 4 complex additions and 4 complex subtractions. Note that this computation is naturally carried out in place with one sequential sweep through the input. Note also that this computation is easy to invert: for example, the sum of $f_0 + f_4$ and $f_0 - f_4$ is $2f_0$, and the difference is $2f_4$.

More generally, let r be a nonzero complex number, and let n be a power of 2. The remainder $f \bmod x^{2n} - r^2$ determines the remainders $f \bmod x^n - r$ and $f \bmod x^n + r$, since $x^n - r$ and $x^n + r$ divide $x^{2n} - r^2$. Specifically, if

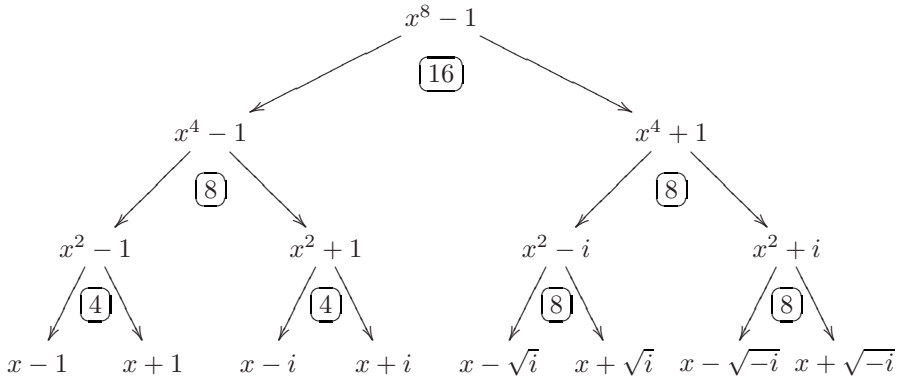
$$f \bmod x^{2n} - r^2 = f_0 + f_1 x + \cdots + f_{2n-1} x^{2n-1},$$

then $f \bmod x^n - r = (f_0 + r f_n) + (f_1 + r f_{n+1})x + \cdots + (f_{n-1} + r f_{2n-1})x^{n-1}$ and $f \bmod x^n + r = (f_0 - r f_n) + (f_1 - r f_{n+1})x + \cdots + (f_{n-1} - r f_{2n-1})x^{n-1}$. This computation involves n complex multiplications by r; n complex additions; and n complex subtractions; totalling $10n$ real operations. The following diagram summarizes the structure and cost of the computation:



Note that some operations disappear when multiplications by r are easy: this computation involves only $8n$ real operations if $r \in \{\sqrt{i}, -\sqrt{i}, \sqrt{-i}, -\sqrt{-i}\}$, and only $4n$ real operations if $r \in \{1, -1, i, -i\}$.

The same idea can be applied recursively:



The final outputs $f \bmod x - 1, f \bmod x + 1, f \bmod x - i, \ldots$ are exactly the (permuted) DFT outputs $f(1), f(-1), f(i), \ldots$, and this computation is exactly Gauss's original FFT. Note that the entire computation is naturally carried out in place, with contiguous inputs to each recursive step. One can further reduce the number of cache misses by merging (e.g.) the top two levels of recursion.

This view of the FFT, identifying each FFT step as a simple polynomial operation, was introduced by Fiduccia in [5]. Most papers (and books) suppress the polynomial structure, viewing each intermediate FFT result as merely a linear function of the input; but "$f \bmod x^n - r$" is much more concise than a matrix expressing the same function!

One might object that the concisely expressed polynomial operations in this section and in subsequent sections are less general than arbitrary linear functions. Is this restriction compatible with the best FFT algorithms? For example, does it allow Van Buskirk's improved operation count? This paper shows that the answer is yes. Perhaps some future variant of the FFT will force Fiduccia's philosophy to be reconsidered, but for the moment one can safely recommend that FFT algorithms be expressed in polynomial form.

3 Review of the Twisted FFT

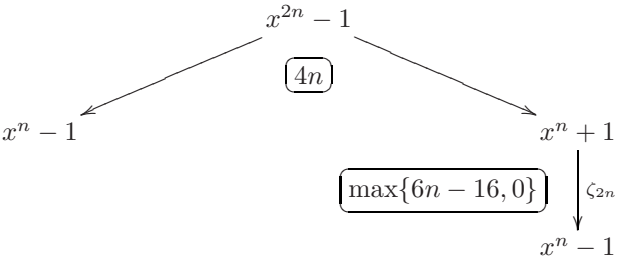
The remainder $f \bmod x^n + 1$ determines the remainder $f(\zeta_{2n}x) \bmod x^n - 1$. Specifically, if $f \bmod x^n + 1 = f_0 + f_1 x + \cdots + f_{n-1}x^{n-1}$, then

$$f(\zeta_{2n}x) \bmod x^n - 1 = f_0 + \zeta_{2n} f_1 x + \cdots + \zeta_{2n}^{n-1} f_{n-1} x^{n-1}.$$

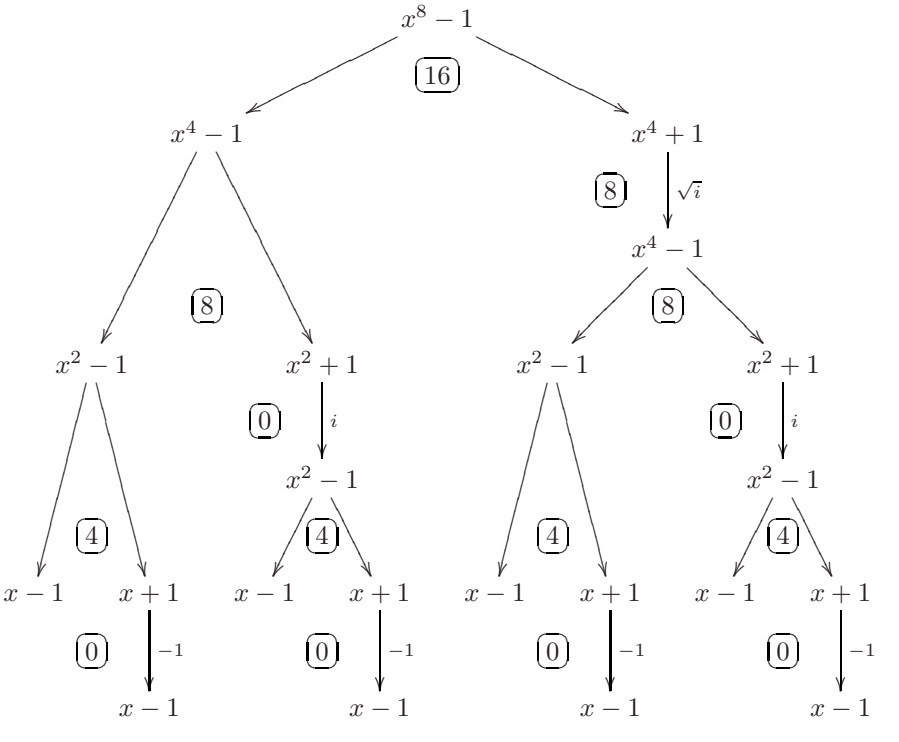
Computing the **twisted** coefficients $f_0, \zeta_{2n} f_1, \ldots, \zeta_{2n}^{n-1} f_{n-1}$ from the coefficients $f_0, f_1, \ldots, f_{n-1}$ involves one multiplication by ζ_{2n}, one multiplication by ζ_{2n}^2, and

so on through ζ_{2n}^{n-1}. These $n-1$ multiplications cost $6(n-1)$ real operations, except that a few multiplications are easier: 6 operations are saved for $\zeta_{2n}^{n/2}$ when $n \geq 2$, and another 4 operations are saved for $\zeta_{2n}^{n/4}, \zeta_{2n}^{3n/4}$ when $n \geq 4$.

The remainder $f \bmod x^{2n}-1$ determines the remainders $f \bmod x^n-1$ and $f \bmod x^n+1$, as discussed in the previous section. It therefore determines the remainders $f \bmod x^n-1$ and $f(\zeta_{2n}x) \bmod x^n-1$, as summarized in the following diagram:



The **twisted FFT** performs this computation and then recursively evaluates both $f \bmod x^n-1$ and $f(\zeta_{2n}x) \bmod x^n-1$ at the nth roots of 1, obtaining the same results as the original FFT. Example, for $n=8$:

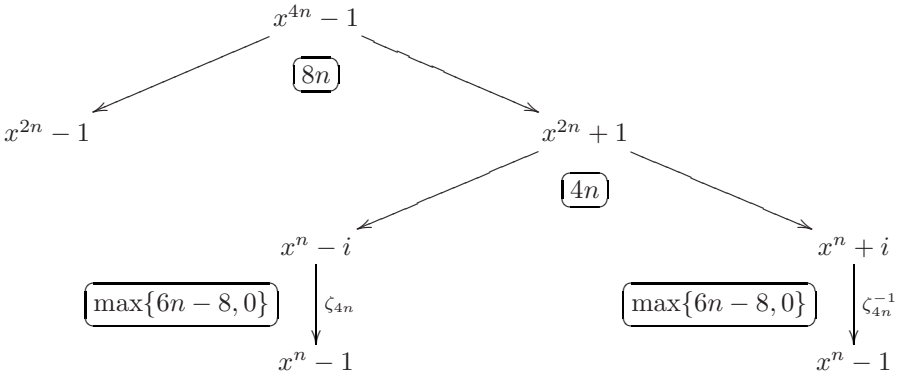


Note that the twisted FFT never has to consider moduli other than $x^n \pm 1$. The twisted FFT thus has a simpler recursive structure than the original FFT. The recursive step does not need to distinguish f from $f(\zeta_{2n}x)$: its job is simply to evaluate an input modulo $x^n - 1$ at the nth roots of 1.

One can easily prove that the twisted FFT uses the same number of real operations as the original FFT: the cost of twisting $x^n + 1$ into $x^n - 1$ is exactly balanced by the savings from avoiding $x^{n/4} - \sqrt{i}$ etc. In fact, the algorithms have the same number of multiplications by each root of 1. (One way to explain this coincidence is to observe that the algorithms are "transposes" of each other.) One might speculate at this point that *all* FFT algorithms have the same number of real operations; but this speculation is solidly disproven by the split-radix FFT, as discussed in Section 4.

4 Review of the Split-Radix FFT

The **split-radix FFT** applies the following diagram recursively:



The notation here is the same as in previous sections:

- from $f \bmod x^{4n} - 1$ compute $f \bmod x^{2n} - 1$ and $f \bmod x^{2n} + 1$;
- from $f \bmod x^{2n} + 1$ compute $f \bmod x^n - i$ and $f \bmod x^n + i$;
- from $f \bmod x^n - i$ compute $f(\zeta_{4n}x) \bmod x^n - 1$;
- from $f \bmod x^n + i$ compute $f(\zeta_{4n}^{-1}x) \bmod x^n - 1$;
- recursively evaluate $f \bmod x^{2n} - 1$ at the $2n$th roots of 1;
- recursively evaluate $f(\zeta_{4n}x) \bmod x^n - 1$ at the nth roots of 1; and
- recursively evaluate $f(\zeta_{4n}^{-1}x) \bmod x^n - 1$ at the nth roots of 1.

If $f \bmod x^n - i = f_0 + f_1x + \cdots + f_{n-1}x^{n-1}$ then $f(\zeta_{4n}x) \bmod x^n - 1 = f_0 + \zeta_{4n}f_1x + \cdots + \zeta_{4n}^{n-1}f_{n-1}x^{n-1}$. The $n - 1$ multiplications here cost $6(n - 1)$ real operations, except that 2 operations are saved for $\zeta_{4n}^{n/2}$ when $n \geq 2$. Similar comments apply to $x^n + i$.

The split-radix FFT uses only about $8n+4n+6n+6n = 24n$ operations to divide $x^{4n}-1$ into $x^{2n}-1, x^n-1, x^n-1$, and therefore only about $(24/1.5)n \lg n = 16n \lg n$

operations to handle $x^{4n}-1$ recursively. Here $1.5 = (2/4)\lg(4/2)+(1/4)\lg(4/1)+(1/4)\lg(4/1)$ arises as the entropy of $2n/4n, n/4n, n/4n$. An easy induction produces a precise operation count: the split-radix FFT handles x^n-1 using 0 operations for $n=1$ and $4n\lg n - 6n + 8$ operations for $n \geq 2$.

For the same split of $x^{4n}-1$ into $x^{2n}-1, x^n-1, x^n-1$, the twisted FFT would use about $30n$ operations: specifically, $20n$ operations to split $x^{4n}-1$ into $x^{2n}-1, x^{2n}-1$, and then $10n$ operations to split $x^{2n}-1$ into x^n-1, x^n-1, as discussed in Section 3. The split-radix FFT does better by delaying the expensive twists, carrying out only two size-n twists rather than one size-$2n$ twist and one size-n twist.

Most descriptions of the split-radix FFT replace $\zeta_{4n}, \zeta_{4n}^{-1}$ with ζ_{4n}, ζ_{4n}^3. Both ζ_{4n}^{-1} and ζ_{4n}^3 are nth roots of $-i$; both variants compute (in different orders) the same DFT outputs. There is, however, an advantage of ζ_{4n}^{-1} over ζ_{4n}^3 in reducing memory-access costs. The split-radix FFT naturally uses ζ_{4n}^k and ζ_{4n}^{-k} as multipliers at the same moment; loading precomputed real numbers $\cos(2\pi k/4n)$ and $\sin(2\pi k/4n)$ produces not only $\zeta_{4n}^k = \cos(2\pi k/4n) + i\sin(2\pi k/4n)$ but also $\zeta_{4n}^{-k} = \cos(2\pi k/4n) - i\sin(2\pi k/4n)$. Reciprocal roots also play a critical role in the tangent FFT; see Section 5.

5 The Tangent FFT

The obvious way to multiply $a+bi$ by a constant $\cos\theta + i\sin\theta$ is to compute $a\cos\theta - b\sin\theta$ and $a\sin\theta + b\cos\theta$. A different approach is to factor $\cos\theta + i\sin\theta$ as $(1+i\tan\theta)\cos\theta$, or as $(\cot\theta + i)\sin\theta$. Multiplying by a real number $\cos\theta$ is relatively easy, taking only 2 real operations. Multiplying by $1+i\tan\theta$ is also relatively easy, taking only 4 real operations.

This change does not make any immediate difference in operation count: either strategy takes 6 real operations, when appropriate constants such as $\tan\theta$ have been precomputed. But the change allows some extra flexibility: the real multiplication can be moved elsewhere in the computation. Van Buskirk's clever observation is that these real multiplications can sometimes be combined!

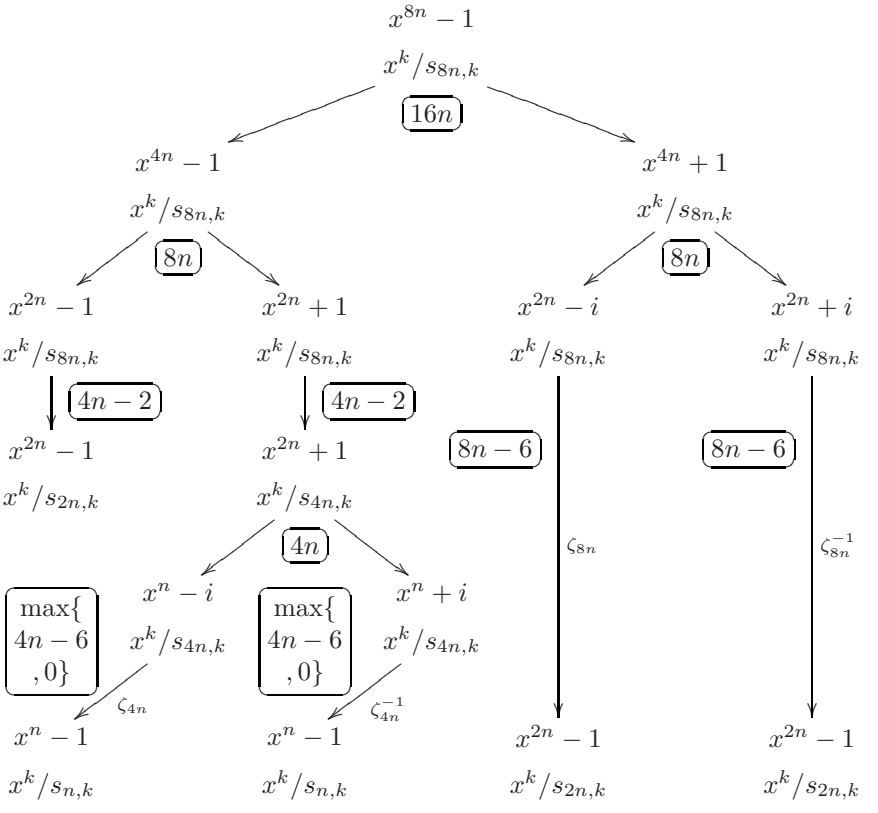
Specifically, let's change the basis $1, x, x^2, \ldots, x^{n-1}$ that we've been using to represent polynomials modulo x^n-1. Let's instead use a vector $(f_0, f_1, \ldots, f_{n-1})$ to represent the polynomial $f_0/s_{n,0} + f_1 x/s_{n,1} + \cdots + f_{n-1}x^{n-1}/s_{n,n-1}$ where

$$s_{n,k} = \prod_{\ell \geq 0} \max\left\{\left|\cos\frac{4^\ell 2\pi k}{n}\right|, \left|\sin\frac{4^\ell 2\pi k}{n}\right|\right\}.$$

This might appear at first glance to be an infinite product, but $4^\ell 2\pi k/n$ is a multiple of 2π once ℓ is large enough, so almost all of the terms in the product are 1.

This wavelet $s_{n,k}$ is designed to have two important features. The first is **periodicity**: $s_{4n,k} = s_{4n,k+n}$. The second is **cost-4 twisting**: $\zeta_{4n}^k(s_{n,k}/s_{4n,k})$ is $\pm(1+i\tan\cdots)$ or $\pm(\cot\cdots+i)$.

The **tangent FFT** applies the following diagram recursively:



This diagram explicitly shows the basis used for each remainder $f \bmod x^{\cdots} - \cdots$. The top node, $x^{8n}-1$ with basis $x^k/s_{8n,k}$, reads an input vector $(f_0, f_1, \ldots, f_{8n-1})$ representing $f \bmod x^{8n}-1 = \sum_{0\le k<8n} f_k x^k/s_{8n,k}$. The next node to the left, $x^{4n}-1$ with basis $x^k/s_{8n,k}$, computes a vector $(g_0, g_1, \ldots, g_{4n-1})$ representing $f \bmod x^{4n}-1 = \sum_{0\le k<4n} g_k x^k/s_{8n,k}$; the equation $s_{8n,k+4n} = s_{8n,k}$ immediately implies that

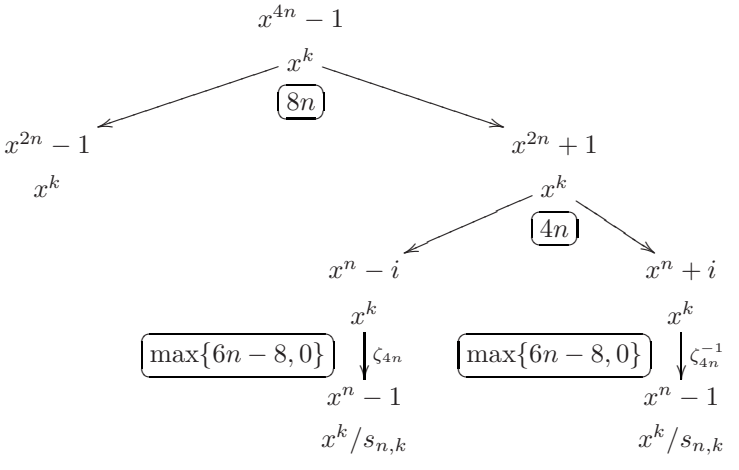
$$(g_0, g_1, \ldots, g_{4n-1}) = (f_0 + f_{4n}, f_1 + f_{4n+1}, \ldots, f_{4n-1} + f_{8n-1}).$$

The next node to the left, $x^{2n}-1$ with basis $x^k/s_{8n,k}$, similarly computes a vector $(h_0, h_1, \ldots, h_{2n-1})$ representing $f \bmod x^{2n}-1 = \sum_{0\le k<2n} h_k x^k/s_{8n,k}$. The next node after that, $x^{2n}-1$ with basis $x^k/s_{2n,k}$ (suitable for recursion), computes a vector $(h'_0, h'_1, \ldots, h'_{2n-1})$ representing $f \bmod x^{2n}-1 = \sum_{0\le k<2n} h'_k x^k/s_{2n,k}$; evidently $h'_k = h_k(s_{2n,k}/s_{8n,k})$, requiring a total of $2n$ real multiplications by the precomputed real constants $s_{2n,k}/s_{8n,k}$, minus 1 skippable multiplication by $s_{2n,0}/s_{8n,0} = 1$. Similar comments apply throughout the diagram: for example, moving from $x^{2n}-i$ with basis $x^k/s_{8n,k}$ to $x^{2n}-1$ with basis $x^k/s_{2n,k}$ involves cost-4 twisting by $\zeta_{8n}^k s_{2n,k}/s_{8n,k}$.

The total cost of the tangent FFT is about $68n$ real operations to divide $x^{8n}-1$ into $x^{2n}-1, x^{2n}-1, x^{2n}-1, x^n-1, x^n-1$, and therefore about $(68/2.25)n\lg n = (34/9)8n\lg n$ to handle $x^{8n}-1$ recursively. Here 2.25 is the entropy of $2n/8n, 2n/8n, 2n/8n, n/8n, n/8n$. More precisely, the cost $S(n)$ of handling x^n-1 with basis $x^k/s_{n,k}$ satisfies $S(1)=0$, $S(2)=4$, $S(4)=16$, and $S(8n)=60n-16+\max\{8n-12,0\}+3S(2n)+2S(n)$. The $S(n)$ sequence begins $0,4,16,56,164,444,1120,2720,6396,14724,33304,\ldots$; an easy induction shows that $S(n)=(34/9)n\lg n-(142/27)n-(2/9)(-1)^{\lg n}\lg n+(7/27)(-1)^{\lg n}+7$ for $n\ge 2$.

For comparison, the split-radix FFT uses about $72n$ real operations for the same division. The split-radix FFT uses the same $16n$ to divide $x^{8n}-1$ into $x^{4n}-1, x^{4n}+1$, the same $8n$ to divide $x^{4n}-1$ into $x^{2n}-1, x^{2n}+1$, the same $8n$ to divide $x^{4n}+1$ into $x^{2n}-i, x^{2n}+i$, and the same $4n$ to divide $x^{2n}+1$ into x^n-i, x^n+i. It also saves $4n$ changing basis for $x^{2n}-1$ and $4n$ changing basis for $x^{2n}+1$. But the tangent FFT saves $4n$ twisting $x^{2n}-i$, another $4n$ twisting $x^{2n}+i$, another $2n$ twisting x^n-i, and another $2n$ twisting x^n+i. The $12n$ operations saved in twists outweigh the $8n$ operations lost in changing basis.

What if the input is in the traditional basis $1, x, x^2, \ldots, x^{n-1}$? One could scale the input immediately to the new basis, but it is faster to wait until the first twist:



The coefficient of x^k in $f \bmod x^n-i$ is now twisted by $\zeta_{4n}^k s_{n,k}$, costing 6 real operations except for the easy cases $\zeta_{4n}^0 s_{n,0}=1$ and $\zeta_{4n}^{n/2} s_{n,n/2}=\sqrt{i}$.

The cost $T(n)$ of handling x^n-1 with basis x^k satisfies $T(1)=0$, $T(2)=4$, and $T(4n)=12n+\max\{12n-16,0\}+T(2n)+2S(n)$. The $T(n)$ sequence begins $0,4,16,56,168,456,1152,2792,6552,15048,33968,\ldots$; an easy induction shows that

$$T(n)=\frac{34}{9}n\lg n-\frac{124}{27}n-2\lg n-\frac{2}{9}(-1)^{\lg n}\lg n+\frac{16}{27}(-1)^{\lg n}+8$$

for $n\ge 2$, exactly matching the operation count in [7, Equation (1)].

References

1. 1968 Fall Joint Computer Conference. In: AFIPS conference proceedings, vol. 33, part one. See [13] (1968)
2. Cooley, J.W., Tukey, J.W.: An Algorithm for the Machine Calculation of Complex Fourier Series. Mathematics of Computation 19, 297–301 (1965)
3. Duhamel, P., Hollmann, H.: Split-Radix FFT algorithm. Electronics Letters 20, 14–16 (1984)
4. Duhamel, P., Vetterli, M.: Fast Fourier Transforms: a Tutorial Review and a State of the Art. Signal Processing 19, 259–299 (1990)
5. Fiduccia, C.M.: Polynomial Evaluation Via the Division Algorithm: the Fast Fourier Transform Revisited. In: [10], pp. 88–93 (1972)
6. Gauss, C.F.: Werke, Band 3 Königlichen Gesellschaft der Wissenschaften. Göttingen (1866)
7. Johnson, S.G., Frigo, M.: A Modified Split-Radix FFT with Fewer Arithmetic Operations. IEEE Trans. on Signal Processing 55, 111–119 (2007)
8. Lundy, T.J., Van Buskirk, J.: A New Matrix Approach to Real FFTs and Convolutions of Length 2^k. Computing 80, 23–45 (2007)
9. Martens, J.B.: Recursive Cyclotomic Factorization—A New Algorithm for Calculating the Discrete Fourier Transform. IEEE Trans. Acoustics, Speech, and Signal Processing 32, 750–761 (1984)
10. Rosenberg, A.L.: Fourth Annual ACM Symposium on Theory Of Computing. Association for Computing Machinery, New York (1972)
11. Sorensen, H.V., Heideman, M.T., Burrus, C.S.: On Computing the Split-Radix FFT. IEEE Trans. Acoustics, Speech, and Signal Processing 34, 152–156 (1986)
12. Vetterli, M., Nussbaumer, H.J.: Simple FFT and DCT Algorithms with Reduced Number of Operations. Signal Processing 6, 262–278 (1984)
13. Yavne, R.: An Economical Method for Calculating the Discrete Fourier Transform. In: [1], pp. 115–125 (1968)
14. Zhou, F., Kornerup, P.: A New Fast Discrete Fourier Transform. J. VLSI Signal Processing 20, 219–232 (1998)

Novel Algebraic Structure for Cyclic Codes

Dang Hoai Bac[1,2], Nguyen Binh[1], and Nguyen Xuan Quynh[1]

[1] Electronics Faculty, Posts & Telecommunications Institute of Technology (PTIT)-Vietnam. Km 10 Hanoi-Hadong Road
bacdh@ptit.edu.vn
[2] Mobile Telecommunication Research Division, Electronics & Telecommunications Research Institute (ETRI)-Korea

Abstract. The novel algebraic structure for the cyclic codes, Cyclic Multiplicative Groups (CMGs) over polynomial ring, is proposed in this paper. According to this algorithm, traditional cyclic codes can be considered as a subclass in these cyclic codes. With CMGs structure, more plentiful good cyclic code cosets can be found in any polynomial rings than other methods. An arbitrary polynomial in polynomial ring can generate cyclic codes in which length of codewords depend on order of the polynomial. Another advantage of this method is that a longer code can be generated from a smaller polynomial ring. Moreover, our technique is flexibly and easily implemented in term of encoding as well as decoding. As a result, the CMGs can contribute a new point of view in coding theory. The significant advantages of proposed cyclic code cosets can be applicable in the modern communication systems and crypto-systems.

1 Introduction

A very important class of codes is, arguably, the class of cyclic codes. Cyclic codes were explored by Prange [1] in the early history of coding theory. A cyclic code is a block code in which a cyclic shift of every code word yields another codeword belonging to the same code [2,3,4]. Currently, cyclic codes are used in a wide variety of communication systems, computer networks and data storage devices to provide inexpensive and effective error detection capabilities.

Although there are advantages of traditional cyclic codes, it can be seen that there is shortage in cyclic code generation. Specifically, the number of generator polynomials is limited because they depend on the number of ideals in polynomial ring, i.e. they depend on the ability of the factorization of $(x^k + 1)$ in polynomial ring $\mathbb{Z}_2[x]/(x^k + 1)$ [3,5].

To overcome this problem, the key idea in this paper is using Cyclic Multiplicative Group structure instead of the method in which cyclic codes generated from ideals, i.e. depend on factorization of polynomial rings. We investigate, for the first time, some properties of the Cyclic Multiplicative Groups in order to find more plentiful good cyclic cosets and their applications than the traditional methods, especially in polynomial rings with two cyclotomic cosets, i.e. the factorization in these rings has only two factors. Based on our knowledge, the cyclic codes in these rings have not been mentioned yet.

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 301–310, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Cyclic codes are built from an arbitrary polynomial in polynomial ring and these codes become independent of the factors of the factorization with CMGs structure. An arbitrary polynomial in polynomial ring can generate cyclic codes with length depended on order of polynomial. This is the main difference in comparison with the traditional method. Traditional cyclic codes can be considered as a subclass of the cyclic codes of the CMGs method.

The proposed method have significant advantages such as: the good constructed cyclic code cosets are larger than conventional; the longer cyclic code can be generated from smaller polynomial ring, leading to easier and faster for implementing. Furthermore, using properties of multiplicative group has much lower critical path delays, thus allowing much faster operating clock rates in implementing by VLSI, such as FPGA [6,7,10]. These properties are also suitable for high-speed cryptographic applications such as elliptic curve cryptography [12]. The effect of extension on the properties of CMGs can be used for further study in PN generation, peak to average power ratio (PAPR) reduction in OFDM, and building cypto-systems.

2 Cyclic Multiplicative Group over Polynomial Ring

2.1 Order of Polynomial

Based on the order of polynomials in polynomial ring, the cyclic multiplicative groups can be generated. CMGs play an important role in new algebraic structure mentioned in this paper.

Definition 1. *Order of polynomial* $a(x)$ *modulo* $(x^k + 1)$ *in polynomial ring* $\mathbb{Z}_2[x]/(x^k + 1)$ *is the positive integer* n *which satisfies:*

$$a(x)^{n+1} = a(x) \pmod{(x^k + 1)} \quad or \quad a(x)^n = e(x) \pmod{(x^k + 1)} \tag{1}$$

where, $e(x)$ *is an idempotent polynomial in this ring,* $e(x)$ *satisfies the property* $e(x) = e^2(x)$ *[2,3].*

Consider polynomial ring $\mathbb{Z}_2[x]/(x^k + 1)$, with k is an odd number, the formula of $x^k + 1$ is:

$$x^k + 1 \;=\; \prod_i f_i(x) \tag{2}$$

where $f_i(x)$ are distinctive irreducible polynomials. With $a(x) \in \mathbb{Z}_2[x]/(x^k + 1)$, the maximum order of $a(x)$ modulo $x^k + 1$ (denote $\max(ord\, a(x))$) is defined:

$$\max(ord\, a(x)) = 2^m - 1 \tag{3}$$

where $m = \max_i \deg f_i(x)$.

In the polynomial ring $\mathbb{Z}_2[x]/(x^k + 1)$, the available order of all polynomials n is $\max(ord\, a(x))$ or divisor of $\max(ord\, a(x))$.

Example 1. With $n = 5$, consider $a(x) \in \mathbb{Z}_2[x]/(x^5+1)$. According to (3), we have: $x^5+1 = f_1(x)\cdot f_2(x) = (1+x)(1+x+x^2+x^3+x^4)$ and $f_1(x) = (1+x) \Rightarrow \deg f_1(x) = m_1 = 1$.

$f_2(x) = (1+x+x^2+x^3+x^4) \Rightarrow \deg f_2(x) = m_2 = 4$.

Hence, $\max(ord\, a(x))$ is calculated as follows: ($a(x)$ is an arbitrary polynomial in polynomial ring $\mathbb{Z}_2[x]/(x^5+1)$.

According to (3), $\max(ord\, a(x)) = 2^m - 1 = 15 = 3\cdot 5$, thus the available orders of an arbitrary $a(x)$ in $\mathbb{Z}_2[x]/(x^5+1)$ are: $n = 1, 3, 5, 15$ ($n = 1$ is order of idempotents).

32 polynomials ($2^k - 1 = 32$) in $\mathbb{Z}_2[x]/(x^5+1)$ are divided into four groups as follow:

- The group of polynomials with order 1: {Zero, (0), (01234), (1234)}.
- The group of polynomials with order 3: {(023), (024), (14), (23)}.
- The group of polynomials with order 5: {(1),(2),(3),(4), (0123), (0124), (0134), (0234)}.
- The group of polynomials with order 15: {(01),(02),(03),(04), (012), (013), (024), (034)}.

Remark: (023) denotes $1 + x^2 + x^3$, (0) denotes 1 and zero = 0.

2.2 Cyclic Multiplicative Groups

Depending on the order of polynomial $a(x)$ modulo (x^k+1) in polynomial ring $\mathbb{Z}_2[x]/(x^k+1)$, we can construct the cyclic multiplicative group (CMGs).

Consider the following CMG:

$$A = \{a^i(x) \pmod{(x^k+1)}, i = \overline{1:n}\}. \tag{4}$$

Some characteristics of CMGs are presented as follows:

- All the elements of a CMG have the same parity of weight.
- The quantity of elements in A is equal order of $a(x) : |A = n$. P
- According to Lagranges theorem [2,3] n is divisor of $2^m - 1 : n|2^m - 1$.
- k is also divisor of $2^m - 1 : k|2^m - 1$.

CMG $I = \{x^i, i = 1, 2, ..., k\}$ is called unity CMG.

2.3 Symmetric Polynomial $\bar{a}(x)$ and Symmetric CMGs

Definition 2. *With $a(x) \in \mathbb{Z}_2[x]/(x^k+1)$, $\bar{a}(x)$ is called symmetric polynomial of $a(x)$, if $a(x) = \sum_{i\in U} x^i$ then $\bar{a}(x) = \sum_{j\in V} x^j$ where*

$$U \cap V = \emptyset \quad U \cup V = S = \{0, 1, 2, \cdots, k-1\}.$$

$e_0(x) = \sum_{i=0}^{k-1} x^i$ is called swallowing idempotent. $e_0(x)$ have the following characteristics:

– *If $a(x)$ is an arbitrary odd weight polynomial, then $a(x)e_0(x) = e_0(x)$.*
– *If $b(x)$ is an arbitrary event weight polynomial, then $b(x)e_0(x) = 0$.*

Based on these characteristics, we can define: $\bar{a}(x) = e_0(x) + a(x)$.

Lemma 1. *Consider $a(x)$ be generator element of CMG A. $\bar{a}(x)$ be generator of CMG $\bar{A}$. We have*

$$\begin{aligned} A &= \left\{a^i \pmod{x^k+1}\right\} \\ \bar{A} &= \left\{\bar{a}^i \pmod{x^k+1}\right\} \\ |A| &= \left|\bar{A}\right| \quad \textit{and} \quad \bar{a}^i(x) = \overline{a^i}(x). \end{aligned} \tag{5}$$

Proof. Consider $a(x)$ be odd weight polynomial, we have:

$$\begin{aligned} a^i(x) &= a^{i-1}(x)a(x) \\ \overline{a^i}(x) &= e_0(x) + a^i(x) \\ \bar{a}(x) &= e_0(x) + a(x) \\ \bar{a}^i(x) &= \bar{a}^{i-1}(x)\bar{a}(x) = \bar{a}^{i-1}(x)\left[e_0(x) + a(x)\right] = \bar{a}^{i-1}(x)a(x) \\ \bar{a}^i(x) &= \bar{a}(x)a^{i-1}(x) = \left[e_0(x) + a(x)\right]a^{i-1}(x) = e_0(x) + a^i(x) = \overline{a^i}(x) \end{aligned}$$

Similarly, we have: $|A| = |\bar{A}|$. The symmetric characteristic of CMGs is useful for constructing cyclic codes.

3 Cyclic Code Based on Cyclic Multiplicative Group

Consider CMG $A = \{a^i(x)\}$, the number of elements in A is: $|A| = n$. The cyclic codes can be built from CMG A in the way described as follows.

Definition 3. *Cyclic code with length n is called a code with its code digits that are elements of a CMG.*

Remark 1

– According to this definition, generator matrix has following form:

$$G = \left[a(x)a^2(x)\cdots a^n(x)\right].$$

– If $I = \{x^i\} \in A$ then cyclic code generated by A is symmetric code.
– If $a(x) = \sqrt[j]{x}$ then ith row of G is cyclic shift of $(i-1)$th row to the right with j.

For traditional cyclic codes, the generator matrix is built from the ideal of polynomial ring. The limitation of the traditional method is the small number of ideals. In some special polynomial rings, $x^k + 1$ have only two factors as follow:

$$x^k + 1 = (1+x)\left(\sum_{i=0}^{k-1} x^i\right) \tag{6}$$

These rings are called the polynomial rings with two cyclotomic cosets (with k = 3, 5, 11, 13, 17, 19 and etc., see Appendix). In these rings, we can not build good cyclic codes, except trivial codes which are repetition code (n, 1) and even parity check code (n, n-1). The cyclic codes in these rings have not been mentioned before. With CMGs structure mentioned above, many cyclic codes can be yielded in polynomial rings with two cyclotomic cosets, as can be seen in example below.

Example 2. Consider polynomial ring $\mathbb{Z}_2[x]/(x^5+1)$. If we choose an arbitrary polynomial $a(x)$ in this ring with order 15 (maximum order in this ring), such as:

$$a(x) = 1 + x^2 + x^4 \Leftrightarrow (024).$$

According to $a(x)$, we can define CMG A as follow:

$$\begin{aligned} A &= \left\{a^i(x)\right\} \\ &= \{(024),(034),(1),(013),(014),(2),(124),(012),(3),(023),(123),(4), \\ &\quad (134),(234),(0)\}. \end{aligned}$$

Cyclic code based on A is (15,5,7) code. This is a systematic code with the following generator matrix:

$$G = \begin{bmatrix} 1\,1\,0\,1\,1\,0\,0\,1\,0\,1\,0\,0\,0\,0\,1 \\ 0\,0\,1\,1\,1\,0\,1\,1\,0\,0\,1\,0\,1\,0\,0 \\ 1\,0\,0\,0\,0\,1\,1\,1\,0\,1\,1\,0\,0\,1\,0 \\ 0\,1\,0\,1\,0\,0\,0\,0\,1\,1\,1\,0\,1\,1\,0 \\ 1\,1\,0\,0\,1\,0\,1\,0\,0\,0\,0\,1\,1\,1\,0 \end{bmatrix}$$

In G , we can see that:

- The 1st column (10101) is $a(x) = (024)$.
- The 2nd column (10011) is $a^2(x) = (034)$.
- ...
- The 15th column (10000) is $a^{15}(x) = (0)$ and this element is the end of CMG A.

In CMG A , we have $(1 + x^2 + x^4)^3 = x \Leftrightarrow (024)^3 = (1)$. Thus, ith row of G is cyclic shift of $(i-1)$th row to the right with 3. The encoding algorithm of cyclic codes based on CMGs above can be given:

$$G = \left[a^i(x) \pmod{x^5+1}\right].$$

□

The decoding algorithm for this code can be performed by threshold algorithm with two levels as follows:

– Orthogonal checksum system with pair of information digits [0] + [1] at first level.

$$\begin{aligned}[0]+[1] &= [012]+[2]\\ &= [014]+[4]\\ &= [034]+[134]\\ &= [024]+[124]\\ &= [013]+[3]\\ &= [023]+[123]\end{aligned}$$

– Second level of threshold decode for information digit [0] in pair of [0]+[1].

$$\begin{aligned}[0] &= [04]+[4]\\ &= [04]+[34]+[3]\\ &= [01]+[12]+[2]\\ &= [01]+[1]\\ &= [023]+[23]\\ &= [34]+[034]\end{aligned}$$

In this code, the distant Hamming is: $d_0 = 7$. The detail schematic for decoding of this code is illustrated in Fig.1. In this scheme, clock rate of A register is equal to three times of clock rate of B register, M is threshold decoder.

The number of possible cyclic codes in this case is can be calculated:

$$N = 1 + 2 \cdot 15 + 1 = 32$$

According to CMGs above, the cyclic code (15,5,7) is built in $\mathbb{Z}_2[x]/(x^5+1)$ with simple way for encoding as well as decoding. Moreover, this cyclic code

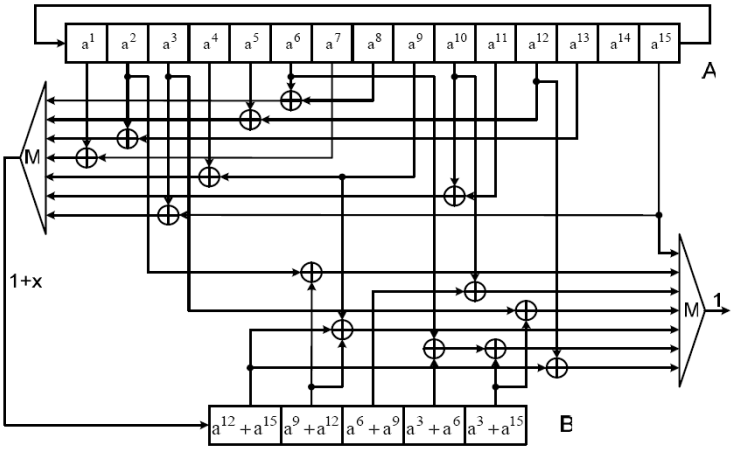


Fig. 1. Decoder scheme for cyclic codes (15, 5) in $\mathbb{Z}_2[x]/(x^5+1)$

equivalent to traditional cyclic codes with the same parameter in larger ring $\mathbb{Z}_2[x]/(x^5+1)$. Here is also significant advantage of this method, thus allow faster implementing.

Lemma 2. *Given a(x) has odd weight, if A generates a cyclic (n,k,d) then A generates a cyclic code (n,k -1,d +1).*

Example 3. In $\mathbb{Z}_2[x]/(x^5+1)$. If $a(x) = 1 + x^2 + x^4 \Leftrightarrow (024)$, according to definition 2, we have $\bar{a}(x) = x + x^3 \Leftrightarrow (13)$, and then the CMG A can be defined as: $\bar{A} = \{\bar{a}^i(x)\}$= {(13), (12),(0234), (24),(23),(0134), (03), (34), (0124), (14), (04), (0123), (02), (01), (1234)}

Cyclic code based on $\bar{A}$ is (15,4,8) code. Clearly, this is an optimal code satisfying Griesmer bound [3].

$\bar{A}$ is equivalent to $A' = \{a^i(x) \pmod{1 + x + x^2 + x^3 + x^4}\}$.

A' = {(13),(12),(1),(013),(23),(2),(03),(012),(3), (023), (123), (0123), (02), (01), (0)}

4 Cyclic Codes over CMGs and Traditional Cyclic Codes

Using CMGs brings more abilities to construct the cyclic codes than using traditional method. For supporting this opinion, we can consider again polynomial ring $\mathbb{Z}_2[x]/(x^5+1)$. In this ring, the factorization of (x^5+1) is:

$$x^5 + 1 = (1 + x)\left(\sum_{i=0}^{4} x^i\right)$$

where $(1 + x)$ and $\left(\sum_{i=0}^{4} x^i\right)$ are irreducible polynomials.

Based on the traditional cyclic code, only two trivial codes are can be built in this ring. With generator polynomial $f_1(x) = (1 + x)$, we yield (5, 4) code which is even parity check code with Hamming distance $d_0 = 2$. With generator polynomial $f_2(x) = (1 + x + x^2 + x^3 + x^4)$ we yield repetition code (5,5) with Hamming distance $d_0 = 5$. However, in $\mathbb{Z}_2[x]/(x^5+1)$, using the CMGs method, as can be seen from example 2, we can construct more cyclic codes with good properties than cyclic codes based on traditional method. This is because the ability of available cyclic code is dependent on a(x) and the order of a(x). Moreover, traditional cyclic codes over ideals I can be a special case of cyclic codes over CMGs.

Lemma 3. *Unity CMG according to modulo h(x) with $h(x)|(x^k+1)$ is a traditional cyclic code with the following generator polynomial $g(x)$*

$$g(x) \;=\; \left[\frac{x^k+1}{h(x)}\right]^* \tag{7}$$

where where $g^*(x)$ is reciprocal polynomial of g (x) :

$$g^*(x) = x^{\deg g(x)} g(x^{-1}) \tag{8}$$

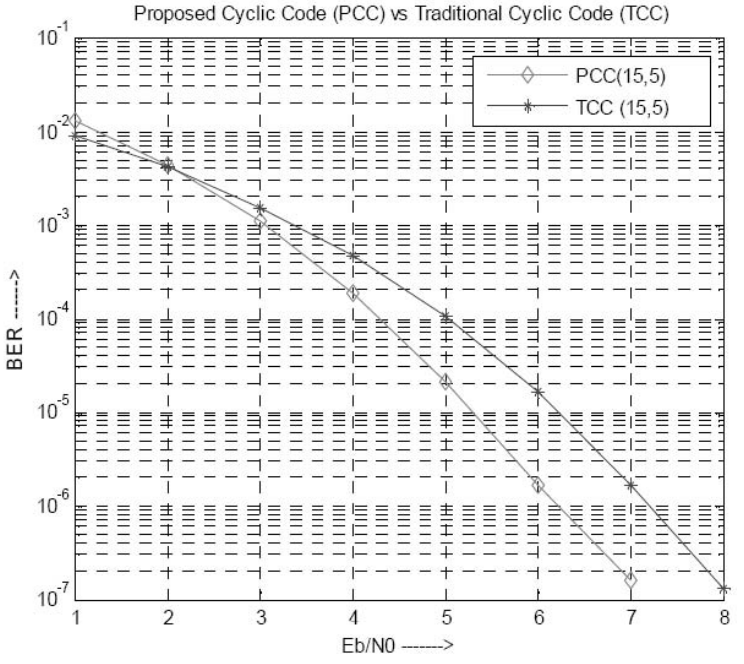


Fig. 2. Comparison of BER performance between proposed and traditional cyclic code (15,5)

Lemma 4. *In decomposition of polynomial ring according to unity CMG, coset G with leader g (x) is a traditional cyclic code with generator polynomial* $g^k(x)$

$$G = \left[g(x) \cdot x^i\right] \tag{9}$$

where g (x) is a divisor of $x^k + 1$ *:* $g(x)|x^k + 1$.

Example 4. Consider $\mathbb{Z}_2[x]/(x^7 + 1)$. If we choose $h(x) = 1 + x + x^2 + x^4$, we can define the unity CMG with modulo h(x) as follow:

$$\begin{aligned} I &= \left\{x^i \pmod{h(x)} : i = \overline{0:6}\right\} \\ &= \left\{1, x, x^2, x^3, 1 + x + x^2, x + x^2 + x^3, 1 + x + x^3\right\} \end{aligned}$$

Clearly, I is equivalent to cyclic code (7,4,3) with generator polynomial $g(x) = 1 + x^2 + x^3$.

Clearly, traditional cyclic codes are only a special case of cyclic codes over CMGs. Traditional cyclic codes are cyclic codes with clock x, cyclic codes over CMGs are cyclic codes with arbitrary clock a(x) . The number of traditional cyclic codes is limited and it depends on number of ideals. The number of cyclic codes over CMGs is greater than that. The simulation results for the comparison between BER performance of the proposed cyclic codes (PCC) based on CMG structure

and traditional cyclic codes (TCC) in an AWGN channel with BPSK modulation are shown in Fig. 2. In this investigation, the (15,5) cyclic code with 5e+7 input information bits is used. From Fig.2, it can be seen that the BER for the proposed cyclic codes is lower than the BER for the traditional cyclic codes. The BER of the proposed cyclic codes (15,5) at Eb/N0 = 7db has a good performance with value 1.6e-7.

5 Conclusion

We have proposed an approach for the efficient construction of cyclic codes from cyclic multiplicative group. Using the novel method, we can consider the traditional cyclic codes as subclass of CMGs cyclic codes. This leads to new point of view of cyclic codes with the ability to extend codewords generation. The novel algebraic structure has significant advantages, the good constructed cyclic code cosets are larger than conventional; the longer cyclic code can be generated from smaller polynomial ring and the better BER performance. Moreover, the characteristics of CMGs are very suitable for implementation of VLSI, such as FPGA with high speed calculation. Our future works will focus on investigation of application cyclic codes based on CMG for PN generation, PAPR reduction in OFDM and cryptography.

References

1. Prange, E.: Cyclic Error-Correcting Codes in Two Symbols. Electronics Research Directorate, Air Force Cambridge Res. Ctr. (1957)
2. MacWilliams, F.J., Sloane, N.J.A: The Theory of Error-Correcting Code. North-Holland, Amsterdam (1977)
3. Van Lint, J.H.: Introduction to Coding Theory, 3rd edn. Springer, Heidelberg (1999)
4. Blahut, R.E.: Theory and Practice of Error Control Coding. Addison-Wesley, Reading, MA (1983)
5. Moon, T.K.: Error Correction Coding: Mathematical Methods and Algorithm. John Wiley & Sons, Inc., Chichester (2005)
6. Pincin, A.: A New Algorithm for Multiplication in Finite Fields. IEEE Trans. Computer 38(1), 1045–1049 (1989)
7. Namin, A.H., Wu, H., Ahmadi, M.: Comb Architectures for Finite Field Multiplication in F_2^m. IEEE Trans. Computers 56(7), 909–916 (2007)
8. Katti, R., Brennan, J.: Low Complexity Multiplication in a Finite Field Using Ring Representation. IEEE Trans. Computers 52(4), 418–427 (2003)
9. Lidl, R., Niederreiter, H.: Introduction to Finite Fields and Their Applications, 2nd edn. Cambridge Univ. Press, Cambridge (1997)
10. Wang, C.C., Truong, T.K., Shao, H.M., Deutsch, L.J., Omura, J.K., Reed, I.S.: VLSI Architectures for Computing Multiplications and Inverses in $GF(2^m)$. IEEE Trans. Computers 34(8), 709–717 (1985)
11. Wu, H., Hasan, M.A., Blake, I.F., Gao, S.: Finite Field Multiplier Using Redundant Representation. IEEE Trans. Computers 51(11), 1306–1316 (2002)
12. Baodian, W., Liu, D., Ma, W., Wang, X.: Property of Finite Fields and Its Cryptography Application. Electron. Lett. 39, 655–656 (2003)

Appendix

The value of "k" so that $\mathbb{Z}_2[x]/(x^k+1)$ is a polynomial ring with two cyclotomic cosets.

k = 3, 5, 11, 13, 19, 29, 37, 53, 59, 61, 67, 83, 101, 107, 131, 139, 149, 163, 173, 179, 181, 197, 211, 227, 269, 293, 317, 347, 349, 373, 379, 389, 419, 421, 443, 461, 467, 491, 509, 523, 541, 547, 557, 563, 587, 613, 619, 653, 659, 661, 677, 701, 709, 757, 773, 787, 779, 821, 827, 829, 853, 859, 877, 883, 907, 941, 947.

Distribution of Trace Values and Two-Weight, Self-orthogonal Codes over $GF(p, 2)$

N. Pinnawala[1], A. Rao[1], and T.A. Gulliver[2]

[1] School of Mathematical and Geospatial Sciences, RMIT University, GPO Box 2476V, Melbourne, VIC - 3001, Australia
nimalsiri.pinnawala@rmit.edu.au,asha@rmit.edu.au
[2] Department of Electrical and Computer Engineering, University of Victoria, P.O. Box 3055, STN CSC, Victoria, B.C., Canada V8W 3P6
agullive@engr.uvic.ca

Abstract. The uniform distribution of the trace map lends itself very well to the construction of binary and non-binary codes from Galois fields and Galois rings. In this paper we study the distribution of the trace map with the argument ax^2 over the Galois field $GF(p, 2)$. We then use this distribution to construct two-weight, self-orthogonal, trace codes.

Keywords: Trace map, self-orthogonal, non-binary, two-weight, Galois fields.

1 Introduction

In [1] and [2] the trace map over Galois field $GF(p, m)$ and ring $GR(p^s, m)$ was used to construct linear codes over $\mathbb{Z}_{2^s}$ and $\mathbb{Z}_{p^s}$, respectively. At that time the distribution of the trace map was very intriguing and the question arose of whether this trace distribution was as straightforward when the argument was changed. One encounter of a different argument was in the search for mutually unbiased bases which can enable a quantum cryptosystem in d-dimensions [3].

The authors were unable to find any information in the literature about such distribution of the trace map other than the fundamental properties. It does turn out that this work is not straightforward and this paper looks at the distribution of $Tr(ax^2)$ over $GF(p, 2)$ for odd primes p. The two-weight self-orthogonal codes generated using this distribution are a by-product.

Let p be a prime and $\mathbb{Z}_p^n$ be the vector space of all n-tuples over the finite field $\mathbb{Z}_p$. If C is a k-dimensional subspace of $\mathbb{Z}_p^n$ then C is called an $[n, k]$ linear code over $\mathbb{Z}_p$. The generator matrix G of an $[n, k]$ code C is simply a matrix whose rows are linearly independent and span the code. The inner product of $x = (x_1, x_2, \ldots x_n), \quad y = (y_1, y_2, \ldots y_n) \in \mathbb{Z}_p^n$ is defined by $x \cdot y = \sum_{i=1}^{n} x_i y_i$. Using the inner product, the dual code $C^\perp$ of C is defined by $C^\perp = \{x \in \mathbb{Z}_p^n | x \cdot c = 0 \ \ \forall c \in C\}$. The code C is called self-orthogonal if $C \subseteq C^\perp$.

Many authors look at self-orthogonal codes, for example, [4,5,6,7]. Following are some preliminary results on self-orthogonal codes that are useful here:

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 311–320, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Lemma 1 (Theorem 1.4.3, [8]). *(i) If $x \in \mathbb{Z}_2^n$ then $w_H(x) \equiv x \cdot x \ (mod\ 2)$. (ii) If $x \in \mathbb{Z}_3^n$ then $w_H(x) \equiv x \cdot x \ (mod\ 3)$.*

Note that this result does not hold for $x \in \mathbb{Z}_p^n$ when $p > 3$, the reason being that when $x \in \mathbb{Z}_p^n$, $w_H(x) = \sum_{i=1}^{p-1} n_i$, where n_i is the number of non-zero i's in x, and $x \cdot x = \sum_{i=1}^{n} x_i^2 = n_1 + n_2 2^2 + n_3 3^2 + \ldots + n_{p-1}(p-1)^2$. This does not imply that $w_H(x) \equiv x \cdot x \ (\text{mod } p)$. Lemma 1 does tell us whether a given ternary code is self-orthogonal.

Lemma 2 (Theorem 1.4.10, [8]). *Let C be an $[n, k, d]$ code over $\mathbb{Z}_3$. C is self-orthogonal if and only if the weight of every non-zero codeword is divisible by 3.*

Again this result cannot check the self-orthogonality of codes over $\mathbb{Z}_p$ for $p > 3$. For this we need the following result.

Lemma 3 (Proposition 1 [7]). *Let p be an odd prime and C be a linear code over $\mathbb{Z}_p$. Then C is self-orthogonal if and only if $c \cdot c = 0 \ \forall \ c \in C$.*

An important invariant of a code is the minimum distance between codewords. The Hamming distance $d_H(x, y)$ between two vectors $x, y \in \mathbb{Z}_p^n$ is defined to be the number of coordinates in which x and y differ. The minimum distance of a code C is the smallest distance between distinct codewords, and is simply denoted by d. The higher the minimum distance, the greater the number of errors that can be corrected. If the minimum distance d of an $[n, k]$ code is known then C is an $[n, k, d]$ code.

The weight enumerator of C is the polynomial $W_C(x, y) = \sum_{i=0}^{n} A_i x^{n-i} y^i$, where A_i is the number of codewords of weight i. A code is called a two-weight code if $|\{i | i \neq 0 \ \text{ and } \ A_i \neq 0\}| = 2$. More details on two-weight codes can be found in [9,10,11], etc. and the references therein.

The trace map can be used to go down from a code defined over an extension field to a code defined over the ground field. Let $\mathbb{F}_q$ be the ground field of the extended field $\mathbb{F}_{q^r}$. Let C be an $\mathbb{F}_{q^r}$-linear code of length n and $Tr : \mathbb{F}_{q^r} \rightarrow \mathbb{F}_q$ be the trace. The code $Tr(C)$, defined as the set of all $(Tr(x_1), Tr(x_2), \ldots, Tr(x_n))$, is called the trace code, where $(x_1, x_2, \ldots, x_n) \in C$. We note that the codes found in this paper could be classed as trace codes, since they are found using a trace map. See [12] for example for details on trace codes.

We now have some of the tools required to classify the codes found in this paper. In the next section we study the distribution of the trace map over $GF(p, 2)$, using the argument ax^2. In Section 3 we construct our codes and study their properties. In the final section, we give some conclusions and detail further work.

2 Distribution of the $Tr(ax^2)$ over $GF(p, 2)$

Let $p(x)$ be a primitive polynomial of degree m over $\mathbb{Z}_p$. The Galois field of characteristic p is defined to be the quotient field $GF(p, m) = \mathbb{Z}_p[x]/(p(x))$. Let ζ be a root of $p(x)$ and therefore $GF(p, m) = \mathbb{Z}_p[\zeta]$. Any element in $GF(p, m)$

can be written as a polynomial of ζ over $\mathbb{Z}_p$, and further it is well known that $GF(p,m) = \{0, 1, \zeta, \zeta^2, \ldots, \zeta^{p^m-2}\}$.

Definition 1. *Let $GF(p,m)$ be the Galois field of characteristic p. The trace map $Tr : GF(p,m) \to \mathbb{Z}_p$ is defined by $Tr(x) = x + x^p + x^{p^2} + \ldots + x^{p^{m-1}}$.*

Theorem 1. *The trace map satisfies the following properties:*
(i) $Tr(x+y) = Tr(x) + Tr(y) \ \ \forall\, x, y \in GF(p,m)$.
(ii) $Tr(ax) = aTr(x) \ \ \forall\, a \in \mathbb{Z}_p,\ x \in GF(p,m)$.
(iii) $Tr(x^p) = Tr(x) \ \ \forall\, x \in GF(p,m)$.
(iv) $Tr(a) = am \ \ \forall\, a \in \mathbb{Z}_p$.
(v) $Tr(x) = 0$ if and only if $x = y^p - y$ for some $y \in GF(p,m)$.
(vi) As x ranges over $GF(p,m)$, $Tr(x)$ takes each element in $\mathbb{Z}_p$ equally often p^{m-1}-times.

Since every non-zero element of $GF(p,2)$ can be written as a power of the primitive element ζ, we first identify the powers of ζ that have trace zero.

Lemma 4. *Let Tr be the trace map over $GF(p,2)$ defined by $Tr(x) = x + x^p$. Let $\zeta^t \in GF(p,2)^* = GF(p,2) \setminus \{0\}$, where $0 \le t \le p^2 - 2$. Then*

i. $Tr(\zeta^{\frac{p+1}{2}}) = 0$.
ii. For $0 \le t < \frac{p+1}{2}$, $Tr(\zeta^t) \neq 0$.
iii. If $Tr(\zeta^t) = 0$ then $Tr(\zeta^{t(2k+1)}) = 0$, where $k = 0, 1, \ldots, p-2$.

Proof:

i. By using the definition of the trace map we have

$$Tr(\zeta^{\frac{p+1}{2}}) = \zeta^{\frac{p+1}{2}} + \left(\zeta^{\frac{p+1}{2}}\right)^p = \zeta^{\frac{p+1}{2}} \left(1 + \zeta^{\left(\frac{p^2-1}{2}\right)}\right).$$

Since ζ^{p^2-1} is the only element in $GF(p,2)^*$ such that $\zeta^{p^2-1} = 1$, we have $\zeta^{\left(\frac{p^2-1}{2}\right)} = -1$. Therefore $Tr(\zeta^{\frac{p+1}{2}}) = 0$.

ii. Let $Tr(\zeta^t) = 0$ for some t, $0 \le t < \frac{p+1}{2}$. This implies that

$$\zeta^t + \zeta^{tp} = 0 \Rightarrow \zeta^t = 0 \text{ or } \zeta^{(p-1)t} = -1$$

Since ζ is a primitive element of $GF(p,2)^*$, $\zeta^t \neq 0$ for any t. Thus $\zeta^{(p-1)t} = -1$ and $\zeta^{(p-1)2t} = 1$. Hence $(p^2-1) | (p-1)2t$, i.e., $2(p-1)t = (p^2-1)m$ for some $m \in \mathbb{Z}^+$. This implies that $t = \frac{(p+1)}{2}m$, a contradiction to the assumption. Therefore $Tr(\zeta^t) \neq 0$ for any t, $0 < t < \frac{p+1}{2}$ and the minimum value of t such that $Tr(\zeta^t) = 0$ is $t = \frac{p+1}{2}$.

iii. From the definition of the trace map if $Tr(\zeta^t) = 0$ then $\zeta^t + \zeta^{tp} = 0 \Rightarrow (\zeta^t)^{2k} = (\zeta^{tp})^{2k}$. Therefore $Tr(\zeta^{t(2k+1)}) = \zeta^{t(2k+1)} + \zeta^{tp(2k+1)} = \zeta^t \zeta^{2tk} + \zeta^{2tkp} \zeta^{tp} = \zeta^t \zeta^{2tkp} + \zeta^{2tkp} \zeta^{tp} = 0$. Thus if $Tr(\zeta^t) = 0$ then $Tr(\zeta^{t(2k+1)}) = 0$. From part (vi) of Theorem 1 there are $p-1$ elements in $GF(p,2)^*$ such that $Tr(x) = 0$. Hence if $Tr(\zeta^t) = 0$ then $Tr(\zeta^{t(2k+1)}) = 0$ for all $k = 0, 1, 2, \ldots, p-2$. □

Corollary 1. *For $x \in GF(p,2)^*$, $Tr(x) = 0$ if and only if $x = \zeta^{(\frac{p+1}{2})(2k+1)} = \zeta^{(p+1)k}\zeta^{\frac{(p+1)}{2}}$, where $k = 0, 1, 2, \ldots, p-2$.*

The base field $GF(p,1) \cong \mathbb{Z}_p$ is a subfield of the extended field $GF(p,2)$. The next lemma gives us those indices t for which $\zeta^t \in GF(p,1)^*$.

Lemma 5. *Let $\zeta^t \in GF(p,2)^*$, for some t, $0 \leq t \leq p^2-1$. If $\zeta^t \in GF(p,1)^*$ then $t = (p+1)k$.*

Proof: Let $\zeta^t \in GF(p,2)^*$, for some t, $0 \leq t \leq p^2-1$. Now $GF(p,1) \cong \mathbb{Z}_p$ is a subfield of $GF(p,2)$. Hence if $\zeta^t \in GF(p,1)^* \cong \mathbb{Z}_p \setminus \{0\}$ then $Tr(\zeta^t\zeta^{\frac{p+1}{2}}) = \zeta^t Tr(\zeta^{\frac{p+1}{2}}) = 0$, from part (ii) of Theorem 1 and part (i) of Lemma 4.

But from Corollary 1, if $x \in GF(p,2)^*$, such that $Tr(x) = 0$ then $x = \zeta^{(\frac{p+1}{2})(2k+1)} = \zeta^{(p+1)k}\zeta^{\frac{(p+1)}{2}}$. Hence $\zeta^t\zeta^{\frac{p+1}{2}} = \zeta^{(p+1)k}\zeta^{\frac{(p+1)}{2}} \Rightarrow \zeta^t = \zeta^{(p+1)k}$, since $\zeta^{\frac{(p+1)}{2}} \neq 0$. Therefore if $\zeta^t \in GF(p,1)^*$ then $t = (p+1)k, k = 0, 1, 2, \ldots, p-2$, i.e., ζ^t is an element of the subfield when $t = (p+1)k$, $k = 0, 1, 2, \ldots, p-2$. □

Thus far we have identified the elements $\zeta^t \in GF(p,2)^*$ which have trace 0 or are in the base field. We are now in a position to study the distribution of $Tr(ax^2)$, when both a and x range over $GF(p,2)$. A useful tool in this study is to list the elements of $GF(p,2)^*$ in a two-dimensional array based on the powers of a chosen primitive element ζ.

Let ζ be a primitive element of $GF(p,2)$. Then $GF(p,2)^* = \{1, \zeta, \zeta^2, \ldots, \zeta^{p^2-2}\}$ and $\zeta^{p^2-1} = \zeta^0 = 1$. Also $\zeta^{(\frac{p+1}{2})(2p-3)+(\frac{p+1}{2})} = \zeta^{\frac{2p^2-3p+2p-3+p+1}{2}} = \zeta^{\frac{2(p^2-1)}{2}} = \zeta^{p^2-1} = 1$. The elements in $GF(p,2)^*$ can now be listed by means of a $(p-1) \times (p+1)$ matrix: $\left[\zeta^{(\frac{p+1}{2})(2k+1)+d}\right]$, where $k = 0, 1, 2, \ldots, p-2$ ranges over the rows of the matrix creating $p-1$ rows and $d = 0, 1, 2, \ldots, p$ ranges over the columns of the matrix creating $p+1$ columns. This $(p-1) \times (p+1)$ matrix is given by

$$\begin{bmatrix}
\zeta^{(\frac{p+1}{2})} & \ldots & \zeta^{(\frac{p+1}{2})+d} & \ldots & \zeta^{(\frac{p+1}{2})+(\frac{p+1}{2})} & \ldots & \zeta^{(\frac{p+1}{2})+p} \\
\zeta^{(\frac{p+1}{2})3} & \ldots & \zeta^{(\frac{p+1}{2})3+d} & \ldots & \zeta^{(\frac{p+1}{2})3+(\frac{p+1}{2})} & \ldots & \zeta^{(\frac{p+1}{2})3+p} \\
\vdots & \ldots & \vdots & \ldots & \vdots & \ldots & \vdots \\
\zeta^{(\frac{p+1}{2})(2k+1)} & \ldots & \zeta^{(\frac{p+1}{2})(2k+1)+d} & \ldots & \zeta^{(\frac{p+1}{2})(2k+1)+(\frac{p+1}{2})} & \ldots & \zeta^{(\frac{p+1}{2})(2k+1)+p} \\
\vdots & \ldots & \vdots & \ldots & \vdots & \ldots & \vdots \\
\zeta^{(\frac{p+1}{2})(2p-3)} & \ldots & \zeta^{(\frac{p+1}{2})(2p-3)+d} & \ldots & \zeta^{p^2-1} = 1 & \ldots & \zeta^{(\frac{p+1}{2})(2p-3)+p}
\end{bmatrix},$$

This arrangement of the elements of $GF(p,2)^*$ enables us to better understand the distribution of the values of the trace map. For ease of reading let $a_k, k = 0, 1, \ldots p-2$, be a listing of the non-zero elements of the base field.

Lemma 6. *The trace of the elements of $GF(p,2)^*$ is distributed in the following manner:*

i. The trace of each element in the first column of the matrix representation of $GF(p,2)^*$ is zero.
ii. The trace of the elements in every other column of the matrix representation of $GF(p,2)^*$ takes every element in $\mathbb{Z}_p\backslash\{0\}$ once only.

Proof

i. From Corollary 1 it is clear that the trace of the elements in the first column of the matrix representation of $GF(p,2)^*$ is zero, i.e., $Tr\left(\zeta^{\left(\frac{p+1}{2}\right)(2k+1)}\right) = 0, \forall\ k = 0,1,2,\ldots,p-2$.

ii. From Lemma 5 the trace of the elements in the d^{th} column ($d \neq 0$) of the matrix is given by

$$\begin{aligned} Tr(\zeta^{\left(\frac{p+1}{2}\right)(2k+1)+d}) &= Tr(\zeta^{(p+1)k}\zeta^{\frac{p+1}{2}}\zeta^d) \\ &= Tr(a_k\zeta^{\frac{p+2d+1}{2}})(\text{ from Lemma 5}) \\ &= a_k Tr(\zeta^{\frac{p+2d+1}{2}})\ \ ; a_k \in GF(p,1)^* \equiv \mathbb{Z}_p\backslash\{0\} \end{aligned}$$

From Corollary 1 we know that for $x \in GF(p,2)^*$, $Tr(x) = 0$ if and only if $x = \zeta^{\left(\frac{p+1}{2}\right)(2k+1)}$, where $k = 0,1,2\ldots,p-2$ and therefore $Tr(\zeta^{\frac{p+2d+1}{2}}) \neq 0$ for all $d = 1,2,\ldots,p$, i.e., $Tr(\zeta^{\frac{p+2d+1}{2}})$ is fixed for each column. In addition, a_k represents every element in $\mathbb{Z}_p\backslash\{0\}$ for $k = 0,1,2,\ldots,p-2$. Consequently the trace of the elements in the d^{th} column of the matrix representation of $GF(p,2)^*$ takes each element in $\mathbb{Z}_p\backslash\{0\}$ exactly once. □

Example 1. Consider the primitive polynomial $p(x) = x^2 + x + 2$ over $\mathbb{Z}_5$. The elements in $GF(5,2)^* = \{1, \zeta, \zeta^2, \ldots, \zeta^{23}\}$ and their trace values are given in the following table:

x	$x = a_1\zeta + a_0$	$Tr(x)$	x	$x = a_1\zeta + a_0$	$Tr(x)$	x	$x = a_1\zeta + a_0$	$Tr(x)$
1	$0\zeta + 1$	2	ζ^8	$3\zeta + 1$	4	ζ^{16}	$2\zeta + 3$	4
ζ	$1\zeta + 0$	4	ζ^9	$3\zeta + 4$	0	ζ^{17}	$1\zeta + 1$	1
ζ^2	$4\zeta + 3$	2	ζ^{10}	$1\zeta + 4$	2	ζ^{18}	$0\zeta + 3$	1
ζ^3	$4\zeta + 2$	0	ζ^{11}	$3\zeta + 3$	3	ζ^{19}	$3\zeta + 0$	2
ζ^4	$3\zeta + 2$	1	ζ^{12}	$0\zeta + 4$	3	ζ^{20}	$2\zeta + 4$	1
ζ^5	$4\zeta + 4$	4	ζ^{13}	$4\zeta + 0$	1	ζ^{21}	$2\zeta + 1$	0
ζ^6	$0\zeta + 2$	4	ζ^{14}	$1\zeta + 2$	3	ζ^{22}	$4\zeta + 1$	3
ζ^7	$2\zeta + 0$	3	ζ^{15}	$1\zeta + 3$	0	ζ^{23}	$2\zeta + 2$	2

The matrix representation of $GF(5,2)^*$ is then:

$$GF(5,2)^* = \begin{bmatrix} \zeta^3 & \zeta^4 & \zeta^5 & \zeta^6 & \zeta^7 & \zeta^8 \\ \zeta^9 & \zeta^{10} & \zeta^{11} & \zeta^{12} & \zeta^{13} & \zeta^{14} \\ \zeta^{15} & \zeta^{16} & \zeta^{17} & \zeta^{18} & \zeta^{19} & \zeta^{20} \\ \zeta^{21} & \zeta^{22} & \zeta^{23} & \zeta^{24} = 1 & \zeta^{25} = \zeta & \zeta^{26} = \zeta^2 \end{bmatrix}_{4\times 6}$$

and the corresponding trace matrix is:

$$Tr(GF(5,2)^*)=\begin{bmatrix}0\,1\,4\,4\,3\,4\\0\,2\,3\,3\,1\,3\\0\,4\,1\,1\,2\,1\\0\,3\,2\,2\,4\,2\end{bmatrix}_{4\times 6}$$

It is clear that the first column is an all zero column and every non-initial column contains each non-zero element of $\mathbb{Z}_5$ exactly once.

We can now examine the trace distribution for the specific case considered in this paper: $Tr(ax^2)$.

Theorem 2. *Let Tr be the trace map over $GF(p,2)$. As x ranges over $GF(p,2)^*$ and for $a \in GF(p,2)^*$, $Tr(ax^2)$ takes each element in $\mathbb{Z}_p\backslash\{0\}$ equally often either $p+1$ times or $p-1$ times.*

In the matrix representation of $GF(p,2)^*$ (Lemma 6), we note that there are $\frac{p+1}{2}$ columns with odd powers of ζ and $\frac{p+1}{2}$ columns with even powers of ζ. We will label these columns as odd and even, respectively. We call the matrix obtained by taking the trace of each element in the matrix representation of $GF(p,2)^*$ as the trace matrix of $GF(p,2)^*$.

Before we can prove Theorem 2, we need to work out some more details of the trace matrix. We consider the two cases, $p \equiv 1 \ (mod\ 4)$ and $p \equiv 3 \ (mod\ 4)$ separately.

Case I: $p \equiv 1 \ (mod\ 4)$ In this case $\frac{p+1}{2}$ is odd. From Lemma 4, $Tr(\zeta^{(\frac{p+1}{2})(2k+1)}) = 0$ for all $k = 0, 1, 2, \ldots, p-2$. Hence the first odd column (which is the first column of the matrix representation of $GF(p,2)^*$) has trace zero. Therefore there are $\frac{p+1}{2} - 1 = \frac{p-1}{2}$ odd columns in the matrix representation of $GF(p,2)^*$ with non-zero trace.

From Lemma 6, the trace of the elements of each of these $\frac{p-1}{2}$ odd columns contain each element in $\mathbb{Z}_p\backslash\{0\}$ exactly once. Thus the trace of all the odd powers of ζ gives us each element in $\mathbb{Z}_p\backslash\{0\}$, $\frac{p-1}{2}$ times, and so the trace of all the even powers of ζ gives us each element in $\mathbb{Z}_p\backslash\{0\}$, $\frac{p+1}{2}$ times.

Case II: $p \equiv 3 \ (mod\ 4)$ Here $\frac{p+1}{2}$ is even. As in case I, $Tr(\zeta^{(\frac{p+1}{2})(2k+1)}) = 0$ for all $k = 0, 1, 2, \ldots, p-2$ and the first even column has trace zero. Therefore there are other $\frac{p+1}{2} - 1 = \frac{p-1}{2}$ even columns in the matrix representation of $GF(p,2)^*$ with non-zero trace and hence the trace of all the even powers of ζ gives us each element in $\mathbb{Z}_p \setminus \{0\}$, $\frac{p-1}{2}$ times. Consequently the trace of all the odd powers of ζ gives us each element in $\mathbb{Z}_p\backslash\{0\}$, $\frac{p+1}{2}$ times.

Proof of Theorem 2 Let $a \in GF(p,2)^*$ be an even (resp. odd) power of ζ and consider the set $\{Tr(ax^2) \mid x \in GF(p,2)^*\}$. This set can be written as two copies of the trace of the elements in the set $\{\zeta^{2h} \mid h = 0, 1, 2, \ldots, \frac{p^2-3}{2}\}$ (resp. $\{\zeta^{2h+1} \mid h = 0, 1, 2, \ldots, \frac{p^2-3}{2}\}$) or its cyclic shifts.

Suppose $p \equiv 1 \ (mod\ 4)$. If $a \in GF(p,2)^*$ is an odd power of ζ then from Case I above, as x ranges over $GF(p,2)^*$, $Tr(ax^2)$ takes each element in $\mathbb{Z}_p\backslash\{0\}$

equally often $p-1$ times. If $a \in GF(p,2)^*$ is an even power of ζ then $Tr(ax^2)$ takes each element in $\mathbb{Z}_p \setminus \{0\}$ equally often $p+1$ times.

Similarly if $p \equiv 3 \ (mod \ 4)$, when $a \in GF(p,2)^*$ is an even power of ζ then from Case II above, as x ranges over $GF(p,2)^*$, $Tr(ax^2)$ takes each element in $\mathbb{Z}_p \setminus \{0\}$ equally often $p-1$ times and when $a \in GF(p,2)^*$ is an odd power of ζ, $Tr(ax^2)$ takes each element in $\mathbb{Z}_p \setminus \{0\}$ equally often $p+1$ times. (See Examples 2 and 3.) □

3 Two-Weight Self-orthogonal Codes Via $Tr(ax^2)$ over $GF(p,2)$

Thus far we have studied the distribution of $Tr(ax^2)$ for x ranging over the Galois field $GF(p,2)$. In this section we apply this result to construct cyclic, two-dimensional, two-weight, self-orthogonal codes over $\mathbb{Z}_p$.

Theorem 3 (Codes from $Tr(ax^2)$)
Let $GF(p,2)$ be the Galois Field of characteristic $p \geq 3$. Let Tr be the trace map over $GF(p,2)$. Consider the matrix $H = [Tr(ax^2)]_{a,x \in GF(p,2)}$.

i. H is a linear code over $\mathbb{Z}_p$ with parameters $[n,k,d_H] = [p^2, 2, (p-1)^2]$, where d_H is the minimum Hamming distance.
ii. H is a two-weight code with Hamming weights p^2-1 and $(p-1)^2$.
iii. The code obtained by deleting the first column of H, denoted by H^*, is a cyclic code with parameters $[p^2-1, 2, (p-1)^2]$.
iv. For $p > 3$, H is a self-orthogonal code.

Proof

i. Let ζ be a primitive element of $GF(p,2)$ and c_i be any element in $GF(p,2)$. Consider the matrix

$$G_H = \begin{bmatrix} Tr(c_i^2), \ i = 1,2,\ldots,p^2 \\ Tr(\zeta c_i^2), \ i = 1,2,\ldots,p^2 \end{bmatrix}_{2 \times p^2}.$$

The two rows of G_H are linearly independent: For $a_0, a_1 \in \mathbb{Z}_p$, and for all $i = 1,2,\ldots,p^2$, $a_0 Tr(c_i^2) + a_1 Tr(\zeta c_i^2) = 0 \Rightarrow a_0 + a_1\zeta = 0$ since $c_i^2 \neq 0$ for some i $\Rightarrow a_0 = a_1 = 0$ since 1 and ζ are linearly independent over $\mathbb{Z}_p$.

Now consider all linear combinations of the two rows in G_H. This gives us $a_0 Tr(c_i^2) + a_1 Tr(\zeta c_i^2) = Tr((a_0 + a_1\zeta)c_i^2)$, $i = 1,2,\ldots,p^2$. Thus G_H is a generator matrix for H, and consequently the length n and the dimension k of H are p^2 and 2, respectively, and H is a linear code.

Now from Theorem 2 every non-zero row of H contains every non-zero element of $\mathbb{Z}_p$ equally often either $p+1$ times or $p-1$ times. Since there are $p-1$ non-zero elements in $\mathbb{Z}_p$, the minimum Hamming weight of H is $(p-1)^2$.

ii. Since every non-zero codeword of H contains each element in $\mathbb{Z}_p \setminus \{0\}$ equally often either $p+1$ times or $p-1$ times, the codewords have Hamming weights either p^2-1 or $(p-1)^2$, and H is a two-weight code over $\mathbb{Z}_p$.

iii. Let H^* be obtained by deleting the first column of H: $H^* =$

$$\begin{bmatrix} Tr(0) & Tr(0) & \dots & Tr(0) & Tr(0) & Tr(0) & \dots & Tr(0) \\ Tr(1) & Tr(\zeta^2) & \dots & Tr(\zeta^{2\frac{(p^2-3)}{2}}) & Tr(1) & Tr(\zeta^2) & \dots & Tr(\zeta^{2\frac{(p^2-3)}{2}}) \\ Tr(\zeta) & Tr(\zeta^3) & \dots & Tr(\zeta\zeta^{2\frac{(p^2-3)}{2}}) & Tr(\zeta) & Tr(\zeta^3) & \dots & Tr(\zeta\zeta^{2\frac{(p^2-3)}{2}}) \\ Tr(\zeta^2) & Tr(\zeta^4) & \dots & Tr(\zeta^2\zeta^{2\frac{(p^2-3)}{2}}) & Tr(\zeta^2) & Tr(\zeta^4) & \dots & Tr(\zeta^2\zeta^{2\frac{(p^2-3)}{2}}) \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ Tr(\zeta^{p^2-2}) & Tr(\zeta^{p^2}) & \dots & Tr(\zeta^{p^2-2}\zeta^{2\frac{(p^2-3)}{2}}) & Tr(\zeta^{p^2-2}) & Tr(\zeta^{p^2}) & \dots & Tr(\zeta^{p^2-2}\zeta^{2\frac{(p^2-3)}{2}}) \end{bmatrix}.$$

The second and third rows generate this code, the next consecutive two rows are the left cyclic shift by one element of the second and third rows, respectively, and so on. Thus H^* is a cyclic code.

iv. Let S be the dot product of every non-zero codeword of H with itself. Again from Theorem 2 every non-zero codeword of H contains each element in $\mathbb{Z}_p \backslash \{0\}$ equally often either $p+1$ times or $p-1$ times. Therefore either

$$S = (p+1)\sum_{i=1}^{p-1} i^2 = \frac{p}{6}(p+1)(2p^2-3p+1)$$

or

$$S = (p-1)\sum_{i=1}^{p-1} i^2 = \frac{p}{6}(p-1)(2p^2-3p+1)$$

If $p > 3$ we have $S \equiv 0 \bmod p$ and from Theorem 3 H is a self-orthogonal code over $\mathbb{Z}_p$ for $p > 3$. □

The following two examples illustrate Theorems 2 and 3.

Example 2. Consider the primitive polynomial $p(x) = x^2 + x + 2$ over $\mathbb{Z}_3$ and let ζ be a root of $p(x)$. The elements of $GF(3,2) = \mathbb{Z}_3[x]/(p(x)) = \mathbb{Z}_3[\zeta]$ can be listed as $\{0, 1, \zeta, \zeta^2, \dots, \zeta^7\}$. The following table provides the trace value of these elements and their squares.

x	$x = a_1\zeta + a_0$	$Tr(x)$	x^2	$Tr(x^2)$	x	$x = a_1\zeta + a_0$	$Tr(x)$	x^2	$Tr(x^2)$	x	$x = a_1\zeta + a_0$	$Tr(x)$	x^2	$Tr(x^2)$
0	$0\zeta + 0$	0	0	0	ζ^2	$2\zeta + 1$	0	ζ^4	1	ζ^5	$2\zeta + 0$	1	ζ^2	0
1	$0\zeta + 1$	2	1	2	ζ^3	$2\zeta + 2$	2	ζ^6	0	ζ^6	$1\zeta + 2$	0	ζ^4	1
ζ	$1\zeta + 0$	2	ζ^2	0	ζ^4	$0\zeta + 2$	1	1	2	ζ^7	$1\zeta + 1$	1	ζ^6	0

Taking $a, x \in GF(3,2) = \{0, 1, \zeta, \zeta^2, \dots, \zeta^7\}$, the 9×9 matrices $A = [(ax^2)]_{a,x \in GF(3,2)}$ and $H = [Tr(ax^2)]_{a,x \in GF(3,2)}$ are given by

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \zeta^0 & \zeta^2 & \zeta^4 & \zeta^6 & \zeta^0 & \zeta^2 & \zeta^4 & \zeta^6 \\ 0 & \zeta^1 & \zeta^3 & \zeta^5 & \zeta^7 & \zeta^1 & \zeta^3 & \zeta^5 & \zeta^7 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \zeta^6 & \zeta^0 & \zeta^2 & \zeta^4 & \zeta^6 & \zeta^0 & \zeta^2 & \zeta^4 \\ 0 & \zeta^7 & \zeta^1 & \zeta^3 & \zeta^5 & \zeta^7 & \zeta^1 & \zeta^3 & \zeta^5 \end{bmatrix}, \quad H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 1 & 0 & 2 & 0 & 1 & 0 \\ 0 & 2 & 2 & 1 & 1 & 2 & 2 & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 2 & 0 & 1 & 0 & 2 & 0 & 1 \\ 0 & 1 & 2 & 2 & 1 & 1 & 2 & 2 & 1 \end{bmatrix}.$$

A generator matrix for H is

$$G_H = \begin{bmatrix} 0\,2\,0\,1\,0\,2\,0\,1\,0 \\ 0\,2\,2\,1\,1\,2\,2\,1\,1 \end{bmatrix}_{2\times 9} \text{ while } \quad H^* = \begin{bmatrix} 0\,0\,0\,0\,0\,0\,0\,0 \\ 2\,0\,1\,0\,2\,0\,1\,0 \\ 2\,2\,1\,1\,2\,2\,1\,1 \\ \vdots\,\vdots\,\vdots\,\vdots\,\vdots\,\vdots\,\vdots\,\vdots \\ 0\,2\,0\,1\,0\,2\,0\,1 \\ 1\,2\,2\,1\,1\,2\,2\,1 \end{bmatrix}_{9\times 8} \text{ is obtained by}$$

deleting the first column of H. H is a linear code over $\mathbb{Z}_3$ with parameters $[9, 2, 4]$. The Hamming weight of each non-zero codeword is either 4 or 8. Thus H is a two-weight code. The punctured code H^*, obtained by deleting the first column of H, is an $[8, 2, 4]$ cyclic code over $\mathbb{Z}_3$. The weight of each non-zero codeword is not divisible by 3 and from Theorem 2, H is not a self-orthogonal code.

Example 3. Consider the primitive polynomial $p(x) = x^2 + x + 2$ over $\mathbb{Z}_5$ and let ζ be a root of $p(x)$. The elements of $GF(5,2) = \mathbb{Z}_5[x]/(p(x)) = \mathbb{Z}_5[\zeta]$ can be listed as $\{0, 1, \zeta, \zeta^2, \ldots, \zeta^{23}\}$. The following table provides the trace values of squares of these elements.

x	x^2	$Tr(x^2)$	x	x^2	$Tr(x^2)$	x	x^2	$Tr(x^2)$	x	x^2	$Tr(x^2)$	x	x^2	$Tr(x^2)$
0	0	0	ζ^4	ζ^8	4	ζ^9	ζ^{18}	1	ζ^{14}	ζ^4	1	ζ^{19}	ζ^{14}	3
1	1	2	ζ^5	ζ^{10}	2	ζ^{10}	ζ^{20}	1	ζ^{15}	ζ^6	4	ζ^{20}	ζ^{16}	4
ζ	ζ^2	2	ζ^6	ζ^{12}	3	ζ^{11}	ζ^{22}	3	ζ^{16}	ζ^8	4	ζ^{21}	ζ^{18}	1
ζ^2	ζ^4	1	ζ^7	ζ^{14}	3	ζ^{12}	1	2	ζ^{17}	ζ^{10}	2	ζ^{22}	ζ^{20}	1
ζ^3	ζ^6	4	ζ^8	ζ^{16}	4	ζ^{13}	ζ^2	2	ζ^{18}	ζ^{12}	3	ζ^{23}	ζ^{22}	3

Selecting $a, x \in GF(5,2) = \{0, 1, \zeta, \zeta^2, \ldots, \zeta^{23}\}$, the matrix $A = [(ax^2)]_{a,x\in GF(5,2)}$ is given by

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 \\ 0 & \zeta^0 & \zeta^2 & \zeta^4 & \zeta^6 & \zeta^8 & \zeta^{10} & \zeta^{12} & \zeta^{14} & \zeta^{16} & \zeta^{18} & \zeta^{20} & \zeta^{22} & \zeta^0 & \zeta^2 & \ldots & \zeta^{22} \\ 0 & \zeta^1 & \zeta^3 & \zeta^5 & \zeta^7 & \zeta^9 & \zeta^{11} & \zeta^{13} & \zeta^{15} & \zeta^{17} & \zeta^{19} & \zeta^{21} & \zeta^{23} & \zeta^1 & \zeta^3 & \ldots & \zeta^{23} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \zeta^{22} & \zeta^0 & \zeta^2 & \zeta^4 & \zeta^6 & \zeta^8 & \zeta^{10} & \zeta^{12} & \zeta^{14} & \zeta^{16} & \zeta^{18} & \zeta^{20} & \zeta^{22} & \zeta^0 & \ldots & \zeta^{20} \\ 0 & \zeta^{23} & \zeta^1 & \zeta^3 & \zeta^5 & \zeta^7 & \zeta^9 & \zeta^{11} & \zeta^{13} & \zeta^{15} & \zeta^{17} & \zeta^{19} & \zeta^{21} & \zeta^{23} & \zeta^1 & \ldots & \zeta^{21} \end{bmatrix}_{25\times 25},$$

and the matrix $H = [Tr(ax^2)]_{a,x\in GF(5,2)}$ is given by

$$H = \begin{bmatrix} 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,2\,2\,1\,4\,4\,2\,3\,3\,4\,1\,1\,3\,2\,2\,1\,4\,4\,2\,3\,3\,4\,1\,1\,3 \\ 0\,4\,0\,4\,3\,0\,3\,1\,0\,1\,2\,0\,2\,4\,0\,4\,3\,0\,3\,1\,0\,1\,2\,0\,2 \\ \vdots\,\vdots \\ 0\,3\,2\,2\,1\,4\,4\,2\,3\,3\,4\,1\,1\,3\,2\,2\,1\,4\,4\,2\,3\,3\,4\,1\,1 \\ 0\,2\,4\,0\,4\,3\,0\,3\,1\,0\,1\,2\,0\,2\,4\,0\,4\,3\,0\,3\,1\,0\,1\,2\,0 \end{bmatrix}_{25\times 25}.$$

The rows of H can be generated by

$$G_H = \begin{bmatrix} 0\,2\,2\,1\,4\,4\,2\,3\,3\,4\,1\,1\,3\,2\,2\,1\,4\,4\,2\,3\,3\,4\,1\,1\,3 \\ 0\,4\,0\,4\,3\,0\,3\,1\,0\,1\,2\,0\,2\,4\,0\,4\,3\,0\,3\,1\,0\,1\,2\,0\,2 \end{bmatrix}_{2\times 25}.$$

Therefore H is a linear code over $\mathbb{Z}_5$ and its parameters are $[25, 2, 16]$. The punctured code H^*, obtained by deleting the first column in H, is a $[24, 2, 16]$ cyclic code over $\mathbb{Z}_5$. The Hamming weight of each non-zero codeword of H is either 16 or 24. Thus H is a two-weight code. From part iv of Theorem 3, H is a self-orthogonal code.

4 Conclusions and Further Work

Even though much work has been done on the classification of nonbinary self-orthogonal codes ([4,5,13], these deal mostly with dimension 3 and larger. The codes we find here are 2-dimensional.

In this paper we studied the distribution of $Tr(ax^2)$ and used it to construct two-dimensional, two-weight, cyclic, self-orthogonal codes over $\mathbb{Z}_p$. The questions that arise are whether we can extend this construction to construct codes over $GF(p, 2)$ using $Tr(ax^\lambda)$ for any integer $\lambda > 0$ and in general over $GF(p, m)$. We are currently doing this research.

References

1. Pinnawala, N., Rao, A.: Cocyclic Simplex Codes of Type α Over $\mathbb{Z}_4$ and $\mathbb{Z}_2^s$. IEEE Trans. Inform. Theory 50(9), 2165–2169 (2004)
2. Rao, A., Pinnawala, N.: New Linear Codes over Z_p^s Via The Trace Map. In: 2005 IEEE International Symposium on Information Theory, Adelaide, Australia, pp. 124–126 (2005)
3. Cerf, N.J., Bourennane, M., Karlsson, A., Gisin, N.: Security of Quantum Key Distribution Using D-Level Systems. Physical Review Letters 88(127902) (2002)
4. Bouyukliev, I., Ostergard, P.R.J.: Classification of Self-Orthogonal Codes. Discrete Math. 19(2), 363–370 (2005)
5. Gupta, M.K., Glynn, D.G., Gulliver, T.A.: On Some Quaternary Self Orthogonal Codes. In: Bozta, S., Sphparlinski, I. (eds.) AAECC-14. LNCS, vol. 2227, pp. 112–121. Springer, Heidelberg (2001)
6. Harada, M., Ostergard, P.R.J.: Self- Dual and Maximal Self-Orthogonal Codes over f_7. Elsevier Disc. Math. 256, 471–477 (2002)
7. Wan, Z.X.: A Characteristic Property of Self-Orthogonal Codes and Its Application to Lattices. Bull. Belg. Maths. Soc. 5, 477–482 (1998)
8. Huffman, W.C., Pless, V.: Fundamentals of Error-Correcting Codes. Cambridge University Press, Cambridge (2003)
9. Calderbank, A.R., Kantor, W.M.: The Geometry of Two-Weight Codes. Bull. London Maths. Soc. 18, 97–122 (1986)
10. Dodunekova, R., Dodunekov, S.M.: Error Detection with a Class of Q-Ary Two-Weight Codes. In: IEEE ISIT 2005, pp. 2232–2235 (2005)
11. Helleseth, T.: Some Two-Weight Codes with Composite Parity-Check Polynomials. IEEE Trans. Inform. Theory 22(5), 631–632 (1976)
12. Bierbrauer, J.: Introduction to Coding Theory. Discrete Mathematics and its Applications. Chapman & Hall/CRC, New York (2005)
13. Chen, Z., Fan, P., Jin, F.: New Results on Self-Orthogonal Unequal Error Protection Codes. IEEE Trans. Inform. Theory 36(5), 1141–1144 (1990)

Generalized Rotation Symmetric and Dihedral Symmetric Boolean Functions – 9 Variable Boolean Functions with Nonlinearity 242

Selçuk Kavut and Melek Diker Yücel

Department of Electrical Engineering and Institute of Applied Mathematics,
Middle East Technical University (METU – ODTÜ), 06531, Ankara, Türkiye
{kavut, melekdy}@metu.edu.tr

Abstract. Recently, 9-variable Boolean functions having nonlinearity 241, which is strictly greater than the bent concatenation bound of 240, have been discovered in the class of Rotation Symmetric Boolean Functions (RSBFs) by Kavut, Maitra and Yücel. In this paper, we present several 9-variable Boolean functions having nonlinearity of 242, which we obtain by suitably generalizing the classes of RSBFs and Dihedral Symmetric Boolean Functions (DSBFs). These functions do not have any zero in the Walsh spectrum values, hence they cannot be made balanced easily. This result also shows that the covering radius of the first order Reed-Muller code $R(1, 9)$ is at least 242.

Keywords: Rotation symmetric boolean functions, dihedral symmetric boolean functions, nonlinearity.

1 Introduction

Constructing Boolean functions with high nonlinearity is a challenging problem in the area of cryptography and combinatorics. The problem is also related to the covering radius of the first order Reed-Muller code. The Boolean functions attaining maximum nonlinearity of $2^{n-1}-2^{\frac{n}{2}-1}$ are called bent [27] which occur only for even number of input variables n. For odd number of input variables n, an upper bound of nonlinearity is given as $2\lfloor 2^{n-2}-2^{\frac{n}{2}-1}\rfloor$ [13]. For odd n, one can get Boolean functions having nonlinearity $2^{n-1}-2^{\frac{n-1}{2}}$ by concatenating two bent functions on $(n-1)$ variables. That is the reason why the nonlinearity value of $2^{n-1}-2^{\frac{n-1}{2}}$ for odd n is known as the bent concatenation bound.

Recently, 9-variable Boolean functions with nonlinearity 241, which is greater than the bent concatenation bound of 240, have been discovered [15] in the RSBF class. The question of whether it is possible to exceed the bent concatenation bound for $n = 9, 11, 13$ was open for almost three decades. It was known for odd $n \leq 7$, that the maximum nonlinearity is equal to the bent concatenation bound, $2^{n-1}-2^{\frac{n-1}{2}}$; since the maximum nonlinearity of 5-variable Boolean functions was found [1] as 12 in 1972, and that of 7-variable Boolean functions was computed [23] as 56 in 1980. However, in 1983, 15-variable Boolean functions with

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 321–329, 2007.
© Springer-Verlag Berlin Heidelberg 2007

nonlinearity 16276 which exceeded the bent concatenation bound were demonstrated [24] and using this result, it became possible to get Boolean functions with nonlinearity $2^{n-1}-2^{\frac{n-1}{2}}+20\times 2^{\frac{n-15}{2}}$ for odd $n \geq 15$. Until 2006, there was a gap for $n = 9, 11, 13$ and the maximum nonlinearity known for these cases was $2^{n-1}-2^{\frac{n-1}{2}}$. In 2006, 9-variable functions, which belong to the class of Rotation Symmetric Boolean functions (RSBFs), with nonlinearity 241 ($=2^{9-1}-2^{\frac{9-1}{2}}+1$) were discovered [15]. Such functions were attained utilizing a steepest-descent based iterative heuristic that first appeared in [17], which was suitably modified in [15] for a search in the class of RSBFs.

The class of RSBFs is important in terms of their cryptographic and combinatorial properties [4,5,6,7,8,9,11,16,21,22,25,28,29]. The nonlinearity and correlation immunity of such functions have been studied in detail [4,11,16,21,22,28,29]. It is now clear that the RSBF class is quite rich in terms of these properties and the recently found 9-variable RSBFs [15] having nonlinearity 241 support this fact. In [20], a subspace of RSBFs called Dihedral Symmetric Boolean Functions (DSBFs), which are invariant under the action of the dihedral group are introduced. It has been shown that some of the 9-variable RSBFs having nonlinearity 241 also belong to this subspace, confirming the richness of DSBFs.

Since the space of the RSBF class is much smaller ($\approx 2^{\frac{2^n}{n}}$) than the total space of Boolean functions (2^{2^n}) on n variables, it is possible to exhaustively search the space of RSBFs up to a certain value of n. In [14], an exhaustive search is carried out for the whole space of 9-variable RSBFs exploiting some combinatorial results related to the Walsh spectra of RSBFs; and it has been shown that there is no RSBF having nonlinearity > 241. In order to find functions with higher nonlinearity, one needs to increase the search space. This motivated us to generalize the classes of RSBFs and DSBFs, and our search in the generalized DSBF and RSBF classes successfully ended up with 9-variable functions having nonlinearity 242. However, since these functions do not have any zero in the Walsh spectrum values, they cannot be made balanced easily.

Considering a Boolean function f as a mapping from $GF(2^n) \rightarrow GF(2)$, the functions for which $f(\alpha^2) = f(\alpha)$ for any $\alpha \in GF(2^n)$, are referred to as idempotents [8,9]. In [24], 15-variable Patterson-Wiedemann functions having nonlinearity $16276 = 2^{15-1} - 2^{\frac{15-1}{2}} + 20$ are identified in the idempotent class. As pointed out in [8,9], the idempotents can be seen as RSBFs with proper choice of basis.

In the following section, we will propose the generalized k-RSBFs, as functions which satisfy $f(\alpha^{2^k}) = f(\alpha)$, where $1 \leq k|n$. Note that if $\gcd(n, k) = 1$, the resulting functions are the same as idempotents; whereas for $k = n$ the entire space of n-variable Boolean functions is covered. In the space of generalized k-RSBFs, imposing the condition of invariance under the action of dihedral group, we obtain the class of generalized k-DSBFs as a subset of k-RSBFs.

2 Background

A Boolean function on n variables may be viewed as a mapping from $V_n = \{0,1\}^n$ into $\{0,1\}$. The *truth table* of a Boolean function $f(x_0, \ldots, x_{n-1})$ is a

binary string of length 2^n, $f = [f(0,0,\ldots,0), f(1,0,\ldots,0), f(0,1,\ldots,0), \ldots, f(1,1,\ldots,1)]$. The *Hamming weight* of a binary string S is the number of 1's in S denoted by $wt(S)$. An n-variable function f is said to be *balanced* if its truth table contains an equal number of 0's and 1's, i.e., $wt(f) = 2^{n-1}$. Also, the *Hamming distance* between equidimensional binary strings S_1 and S_2 is defined by $d(S_1, S_2) = wt(S_1 \oplus S_2)$, where $\oplus$ denotes the addition over $GF(2)$.

An n-variable Boolean function $f(x_0, \ldots, x_{n-1})$ can be considered to be a multivariate polynomial over $GF(2)$, called the *algebraic normal form* (ANF) of f. The number of variables in the highest order product term with nonzero coefficient is called the *algebraic degree* of f and denoted by $deg(f)$.

Functions of degree at most one are called *affine* functions. An affine function with constant term equal to zero is called a *linear* function. The set of all n-variable affine functions is denoted by $A(n)$. The nonlinearity of an n-variable function f is $nl(f) = \min_{g \in A(n)}(d(f,g))$, i.e., the minimum distance from the set of all n-variable affine functions.

Let $x = (x_0, \ldots, x_{n-1})$ and $w = (w_0, \ldots, w_{n-1})$ both belong to $\{0,1\}^n$ and $x \cdot w = x_0 w_0 \oplus \ldots \oplus x_{n-1} w_{n-1}$. Then the *Walsh transform* of the n-variable Boolean function $f(x)$ is a real valued function over $\{0,1\}^n$ which is defined as

$$W_f(w) = \sum\nolimits_{x \in \{0,1\}^n} (-1)^{f(x)} (-1)^{w \cdot x}.$$

In terms of the Walsh spectrum, the nonlinearity of f is given by

$$nl(f) = \frac{1}{2}(2^n - \max_{w \in \{0,1\}^n} |W_f(w)|).$$

The autocorrelation function of a Boolean function f is given by

$$r_f(d) = \sum\nolimits_{x \in \{0,1\}^n} (-1)^{f(x)} (-1)^{f(x \oplus d)}, \text{ where } d = (d_0, \ldots, d_{n-1}) \in \{0,1\}^n.$$

The autocorrelation value having maximum magnitude is also known as the absolute indicator [30] and denoted as $\triangle_f = \max_{d \neq (0,\ldots,0) \in \{0,1\}^n} |r_f(d)|$.

3 Generalized Rotation and Dihedral Symmetric Boolean Functions

After briefly summarizing RSBFs, we propose the generalized classes of k- RSBFs and k-DSBFs in Definition 2 and Definition 3 respectively. Letting $(x_0, x_1, \ldots, x_{n-1}) \in V_n$, the (left) k-cyclic shift operator ${\rho^k}_n$ on n-tuples is defined as ${\rho^k}_n(x_0, x_1, \ldots, x_{n-1}) = (x_{(0+k) \bmod n}, \ldots, x_{(n-1+k) \bmod n})$, for $1 \leq k \leq n$.

Definition 1. *A Boolean function f is called* rotation symmetric *if for each input $(x_0, \ldots, x_{n-1}) \in \{0,1\}^n$, $f({\rho^1}_n(x_0, \ldots, x_{n-1})) = f(x_0, \ldots, x_{n-1})$.*

That is, RSBFs are invariant under all cyclic rotations of the inputs. The inputs of a rotation symmetric Boolean function can be divided into *orbits* so

that each orbit consists of all cyclic shifts of one input. An orbit generated by $(x_0, x_1, \ldots, x_{n-1})$ is denoted by $G_n(x_0, x_1, \ldots, x_{n-1}) = \{{\rho^k}_n(x_0, x_1, \ldots, x_{n-1}) \mid 1 \leq k \leq n\}$ and the number of such orbits is g_n ($\approx 2^{\frac{2^n}{n}}$). More specifically, g_n is equal to $\frac{1}{n}\sum_{t|n}\phi(t)2^{\frac{n}{t}}$ [28], where $\phi(t)$ is the Euler's phi-function. The total number of n-variable RSBFs is 2^{g_n}.

In the following, we define the generalized RSBFs as k-rotation symmetric Boolean functions (k-RSBFs).

Definition 2. *Let $1 \leq k \leq n, k|n$. An n-variable Boolean function f is called* k-rotation symmetric *if for each input $(x_0, \ldots, x_{n-1}) \in \{0,1\}^n$, $f({\rho^k}_n(x_0, \ldots, x_{n-1})) = f(x_0, \ldots, x_{n-1})$.*

As can be seen, the k-rotation symmetric Boolean functions are invariant under k-cyclic rotations of inputs. Therefore, an orbit of a k-RSBF generated by $(x_0, x_1, \ldots, x_{n-1})$ is $G^k{}_n(x_0, x_1, \ldots, x_{n-1}) = \{{\rho^i}_n(x_0, x_1, \ldots, x_{n-1}) \mid i = k, 2k, 3k, \ldots, n\}$. For example, $G^3{}_9(001, 001, 111) = \{(001, 001, 111), (001, 111, 001), (111, 001, 001)\}$.

If $g_{n,k}$ is the number of distinct orbits in the class of k-RSBFs of n variables, one can show that $g_{n,k} = \frac{k}{n}\sum_{t|\frac{n}{k}}\phi(t)2^{\frac{n}{t}}$, where $\phi(t)$ is the Euler's phi function.

In [20], a subspace of RSBFs called Dihedral Symmetric Boolean Functions (DSBFs), which are invariant under the action of dihedral group D_n are introduced. In addition to the (left) k-cyclic shift operator ${\rho^k}_n$ on n-tuples, which is defined previously, the dihedral group D_n also includes the reflection operator $\tau_n(x_0, x_1, \ldots, x_{n-1}) = (x_{n-1}, \ldots, x_1, x_0)$. The $2n$ permutations of D_n are then defined as $\{{\rho^1}_n, {\rho^2}_n, \ldots, {\rho^{n-1}}_n, {\rho^n}_n, {\tau^1}_n, {\tau^2}_n, \ldots, {\tau^{n-1}}_n, {\tau^n}_n\}$. The dihedral group D_n generates equivalence classes in the set V_n [26]. Let d_n be the number of such partitions. The following proposition gives the exact count of d_n [10, page 184], [20].

Proposition 1. *Let d_n be the total number of orbits induced by the dihedral group D_n acting on V_n. Then $d_n = g_n/2 + l$, where, $g_n = \frac{1}{n}\sum_{t|n}\phi(t)2^{\frac{n}{t}}$ is the number of rotation symmetric classes [28], $\phi(t)$ is the Euler's phi-function and*

$$l = \begin{cases} \frac{3}{4}2^{\frac{n}{2}}, & \text{if } n \text{ is even,} \\ 2^{\frac{n-1}{2}}, & \text{if } n \text{ is odd.} \end{cases}$$

Since there are 2^{d_n} many n-variable DSBFs, a reduction in the size of the search space over the size of RSBFs is provided.

Definition 3. *Let $1 \leq k \leq n, k|n$. An n-variable Boolean function f is called* k-dihedral symmetric *if f is invariant under the group action ${D^k}_n = \{{\rho^i}_n, \tau_n{\rho^i}_n \mid i = k, 2k, 3k, ..., n\}$.*

As the class of DSBFs is a subspace of k-DSBFs, we call k-DSBFs "generalized dihedral symmetric Boolean functions". One should observe that k-DSBFs is a subspace of k-RSBFs.

When Proposition 1 is applied to k-dihedral symmetric functions, we obtain the following corollary.

Corollary 1. *Let $d_{n,k}$ be the number of distinct orbits, in the class of k-DSBFs of n variables. Then, $d_{n,k} = g_{n,k}/2 + l$, where, $g_{n,k} = \frac{k}{n}\sum_{t|\frac{n}{k}} \phi(t)2^{\frac{n}{t}}$ is the number of k-rotation symmetric classes, $\phi(t)$ is the Euler's phi-function and*

$$l = \begin{cases} 2^{\frac{n}{2}-1}, \text{ if } n \text{ is even, } k \text{ is even,} \\ \frac{3}{4}2^{\frac{n}{2}}, \text{ if } n \text{ is even, } k \text{ is odd,} \\ 2^{\frac{n-1}{2}}, \text{ if } n \text{ is odd.} \end{cases}$$

Table 1 compares the orbit counts of k-rotational classes, k-dihedral classes, RSBFs, and DSBFs for $k|n, n \leq 15$.

Table 1. Comparison of the orbit counts $g_n, d_n, g_{n,k}$ and $d_{n,k}$ for $n = 4, 6, \ldots, 15$, and all integers k, which divide n

n		k	2	3	4	5	6	7
4	$g_4 = 6$	$g_{4,k}$	10	–	–	–	–	–
	$d_4 = 6$	$d_{4,k}$	7	–	–	–	–	–
6	$g_6 = 14$	$g_{6,k}$	24	36	–	–	–	–
	$d_6 = 13$	$d_{6,k}$	16	24	–	–	–	–
8	$g_8 = 36$	$g_{8,k}$	70	–	136	–	–	–
	$d_8 = 30$	$d_{8,k}$	43	–	76	–	–	–
9	$g_9 = 60$	$g_{9,k}$	–	176	–	–	–	–
	$d_9 = 46$	$d_{9,k}$	–	104	–	–	–	–
10	$g_{10} = 108$	$g_{10,k}$	208	–	–	528	–	–
	$d_{10} = 78$	$d_{10,k}$	120	–	–	288	–	–
12	$g_{12} = 352$	$g_{12,k}$	700	1044	1376	–	2080	–
	$d_{12} = 224$	$d_{12,k}$	382	570	720	–	1072	–
14	$g_{14} = 1182$	$g_{14,k}$	2344	–	–	–	–	8256
	$d_{14} = 687$	$d_{14,k}$	1236	–	–	–	–	4224
15	$g_{15} = 2192$	$g_{15,k}$	–	6560	–	10944	–	–
	$d_{15} = 1224$	$d_{15,k}$	–	3408	–	5600	–	–

4 Search Strategy

Our search strategy (refer to [15,16] for details) uses a steepest-descent like iterative algorithm in the pre-chosen set of n-variable Boolean functions (3-DSBFs or 3-RSBFs for $n = 9$). Each iteration step accepts the function f and outputs the function f_{min}. Initiating the algorithm with a random Boolean function, at each iteration all costs are calculated within a pre-defined neighborhood of f, and the function having the smallest cost is chosen as the iteration output f_{min}. In some rare cases, the cost of f_{min} may be larger than or equal to the cost of f. This is the crucial part of the search strategy, which provides the ability to escape from local minima and its distinction from the steepest-descent algorithm. Hence, the algorithm minimizes the cost until a local minimum is attained; then it takes a step in the direction of nondecreasing cost. The deterministic step in

the reverse direction corresponds to the smallest possible cost increase within the pre-defined neighborhood of the preceding Boolean function. The choice of the cost function is also critical and it is chosen as:

$$Cost(f) = \sum_{w} |W_f(w)^2 - 2^n|^2,$$

which is also equal to the sum of squares of the autocorrelation values given by $\sum_{d,d\neq(0,\ldots,0)} {r_f}^2(d)$, excluding $r_f(0,\ldots,0)$.

5 Results

We apply our search strategy to 9-variable 3-DSBFs, where the size of the 3-DSBF search space is 2^{104} (see Table 1). We have found several unbalanced Boolean functions having nonlinearity 242. Among them there are two different absolute indicator values, which are 32 and 40. The following is the truth table of a 9-variable, 3-dihedral symmetric Boolean function having nonlinearity 242, absolute indicator value 40, and algebraic degree 7:

```
68B7EF2DA03B0D3EA00DB6A96DD99AEAFDB9C842B6D5DC8C4526CE0DD29020DB
B75FE3314568344E73688FF0CB2482E065231869E1AA4583765CC491F8A8DB12
```

And, the function below is another 9-variable 3-DSBF having nonlinearity 242, absolute indicator value 32, and algebraic degree 7:

```
125425D30A398F36508C06817BEE122E250D973314F976AED58A3EA9120DA4FE
0E4D4575C42DD0426365EBA7FC5F45BE9B2F336981B5E1863618F49474F6FE00
```

Using a computer system with Pentium IV 2.8 GHz processor and 256 MB RAM, and setting the maximum iteration number to $N = 60,000$, a typical run of the search algorithm takes 1 minute and 34 seconds. We have carried out 100 runs, each with $N = 60,000$. Out of 6 million distinct 3-DSBFs, 152 functions have the nonlinearity 241, and 36 many 3-DSBFs have the nonlinearity 242.

Additionally, we have applied the search strategy to 9-variable 3-RSBFs (the size of the search space is now 2^{176} as can be seen from Table 1), for which we initiate the search algorithm with a 9-variable 3-DSBF having nonlinearity 242. Then we have obtained some 9-variable 3-RSBFs having nonlinearity 242, absolute indicator 56, and algebraic degree 7. The following is the truth table of such a function:

```
3740B6A118A1E19642A85E2B7E2F3C3CB65FA0D95EC9DB1EA92BDB3666185AE0
087F5FE6E0757106A12FC918754C40E8A1BCCB7A714032A8961456E066E8A801
```

It is clear that using one of the above 9-variable functions (say f) and a 2-variable bent function (say g), the 11-variable function $g(y_0, y_1) \oplus f(x_0, \ldots, x_8)$ with highest -till date- nonlinearity of $2^{11-1} - 2^{\frac{11-1}{2}} + 4 = 996$, can be obtained.

Similarly $h(y_0, y_1, y_2, y_3) \oplus f(x_0, \ldots, x_8)$ is the most nonlinear 13-variable function known to date, with nonlinearity $2^{13-1} - 2^{\frac{13-1}{2}} + 8 = 4040$ where h is a 4-variable bent function and f is one of the above 9-variable functions with nonlinearity 242. We think this is a significant improvement on the results of [15]. However, since the nonlinearity $2^{n-1} - 2^{\frac{n-1}{2}} + 2 \times 2^{\frac{n-9}{2}}$, which can be obtained by bent concatenation of 9-variable functions with nonlinearity 242 is less than the nonlinearity $2^{n-1} - 2^{\frac{n-1}{2}} + 20 \times 2^{\frac{n-15}{2}}$ given in [24] for odd $n \geq 15$, this result is significant only for odd $13 \geq n \geq 9$.

6 Coding Theoretic Significance

The concept of *urcoset* was first presented in [12] and then in [2,3] as *orphan coset*. The set D defines an urcoset, if the union of the support of the leaders of D covers the full space; in other words, a coset D of the first order Reed-Muller code $R(1, n)$ with a set of coset leaders $L(D)$ is an urcoset [18], when $\cup_{g \in L(D)} supp(g) = \{0, 1, \ldots, 2^n - 1\}$.

In [9], orphan cosets having minimum weight of 240 have been reported, and in [14] it is confirmed that each of the cosets $f \oplus R(1, 9)$ is an orphan or urcoset, where f is any RSBF having nonlinearity 241.

We have checked by running a computer program that for any of the above functions f having nonlinearity 242, each of the cosets $f \oplus R(1, 9)$ is an orphan or urcoset. This is the first time orphan cosets having minimum weight 242 are demonstrated.

In [2], it is conjectured that the covering radius [19,24] of $R(1, n)$ is even. Our results for $n = 9$ show that the covering radius is at least 242 and it is an interesting open question to settle it. The upper bound presented in [13] for the covering radius of $R(1, 9)$ is 244.

References

1. Berlekamp, E.R., Welch, L.R.: Weight Distributions of the Dosets of the (32, 6) Reed-Muller code. IEEE Trans. Inform. Theory 18(1), 203–207 (1972)
2. Brualdi, R.A., Cai, N., Pless, V.: Orphan Structure of the First Order Reed-Muller Codes. Discrete Mathematics 102, 239–247 (1992)
3. Brualdi, R.A., Pless, V.S.: Orphans of the First Order Reed-Muller Codes. IEEE Trans. Inform. Theory 36(2), 399–401 (1990)
4. Clark, J., Jacob, J., Maitra, S., Stanica, P.: Almost Boolean Functions: The Design of Boolean Functions by Spectral Inversion. Computational Intelligence 20(3), 450–462 (2004)
5. Cusick, T.W., Stanica, P.: Fast Evaluation, Weights and Nonlinearity of Rotation-Symmetric Functions. Discrete Mathematics 258, 289–301 (2002)
6. Dalai, D.K., Gupta, K.C., Maitra, S.: Results on Algebraic Immunity for Cryptographically Significant Boolean Functions. In: Canteaut, A., Viswanathan, K. (eds.) INDOCRYPT 2004. LNCS, vol. 3348, pp. 92–106. Springer, Heidelberg (2004)
7. Dalai, D.K., Maitra, S., Sarkar, S.: Results on rotation symmetric Bent functions. In: BFCA 2006. 2nd International Workshop on Boolean Functions: Cryptography and Applications, pp. 137–156 (2006)

8. Filiol, E., Fontaine, C.: Highly nonlinear balanced Boolean Functions with a good Correlation-Immunity. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 475–488. Springer, Heidelberg (1998)
9. Fontaine, C.: On Some Cosets of the First-Order Reed-Muller Code With High Minimum Weight. IEEE Trans. Inform. Theory 45(4), 1237–1243 (1999)
10. Harary, F.: Graph Theory. Addison-Wesley Publishing Company, Reading (1972)
11. Hell, M., Maximov, A., Maitra, S.: On Efficient Implementation of Search Strategy For Rotation Symmetric Boolean Functions. In: ACCT 2004, Black Sea Coast, Bulgaria (2004)
12. Helleseth, T., Mattson Jr., H.F.: On the Cosets of the Simplex Code. Discrete Math. 56, 169–189 (1985)
13. Hou, X.-d.: On the Norm and Covering Radius of the First Order Reed-Muller codes. IEEE Trans. Inform. Theory 43(3), 1025–1027 (1997)
14. Kavut, S., Maitra, S., Sarkar, S., Yücel, M.D.: Enumeration of 9-variable Rotation Symmetric Boolean Functions having Nonlinearity > 240. In: Barua, R., Lange, T. (eds.) INDOCRYPT 2006. LNCS, vol. 4329, pp. 266–279. Springer, Heidelberg (2006)
15. Kavut, S., Maitra, S., Yücel, M.D.: Search for Boolean Functions with Excellent Profiles in the Rotation Symmetric Class. IEEE Trans. Inform. Theory 53(5), 1743–1751 (2007)
16. Kavut, S., Maitra, S., Yücel, M.D.: Autocorrelation Spectra of Balanced Boolean Functions on Odd Number Input Variables With Maximum Absolute Value $< 2^{\frac{n+1}{2}}$. In: BFCA 2006, University of Rouen, France, pp. 73–86 (2006)
17. Kavut, S., Yücel, M.D.: A New Algorithm for the Design of Strong Boolean Functions. In: First National Cryptology Symposium, METU, Ankara, Turkey, pp. 95–105 (2005)
18. Langevin, P.: On the Orphans and Covering Radius of the Reed-Muller Codes. In: Mattson, H.F., Rao, T.R.N., Mora, T. (eds.) AAECC-9. LNCS, vol. 539, pp. 234–240. Springer, Heidelberg (1991)
19. MacWillams, F.J., Sloane, N.J.A.: The Theory of Error Correcting Codes. North Holland, Amsterdam (1977)
20. Maitra, S., Sarkar, S., Dalai, D.K.: On Dihedral Group Invariant Boolean Functions. In: BFCA 2007, University of Rouen, France (2007)
21. Maximov, A.: Classes of Plateaued Rotation Symmetric Boolean functions under Transformation of Walsh Spectra. In: Ytrehus, Ø. (ed.) WCC 2005. LNCS, vol. 3969, pp. 325–334. Springer, Heidelberg (2006)
22. Maximov, A., Hell, M., Maitra, S.: Plateaued Rotation Symmetric Boolean Functions on Odd Number of Variables. In: BFCA 2005, University of Rouen, France (2005)
23. Mykkeltveit, J.J.: The Covering Radius of the (128, 8) Reed-Muller Code is 56. IEEE Trans. Inform. Theory 26(3), 359–362 (1980)
24. Patterson, N.J., Wiedemann, D.H.: The Covering Radius of the $(2^{15}, 16)$ Reed-Muller code is At Least 16276. IEEE Trans. Inform. Theory 29(3), 354–356 (1983)
25. Pieprzyk, J., Qu, C.X.: Fast Hashing and Rotation-Symmetric Functions. J. Universal Computer Science 5, 20–31 (1999)
26. Roberts, F.S.: Applied Combinatorics. Prentice-Hall, Inc., Englewood Ciffs, New Jersey

27. Rothaus, O.S.: On Bent Functions. Journal of Combinatorial Theory, Series A 20, 300–305 (1976)
28. Stanica, P., Maitra, S.: Rotation Symmetric Boolean Functions – Count and Cryptographic Properties. In: R. C. Bose Centenary Symposium on Discrete Mathematics and Applications. Electronic Notes in Discrete Mathematics, vol. 15, Elsevier, Amsterdam (2004)
29. Stanica, P., Maitra, S., Clark, J.: Results on Rotation Symmetric Bent and Correlation Immune Boolean Functions. In: Roy, B., Meier, W. (eds.) FSE 2004. LNCS, vol. 3017, pp. 161–177. Springer, Heidelberg (2004)
30. Zhang, X.M., Zheng, Y.: GAC – The Criterion for Global Avalanche Characteristics of Cryptographic Functions. J. Universal Computer Science 1(5), 316–333 (1995)

On Quasi-cyclic Codes over Integer Residue Rings

Maheshanand[1] and Siri Krishan Wasan[2]

[1] Centre for Development of Advanced Computing, Noida, India
maheshanand@cdacnoida.in
[2] Department of Mathematics, Jamia Millia Islamia, New Delhi, India
skwasan@yahoo.com

Abstract. In this paper we consider some properties of quasi-cyclic codes over the integer residue rings. A quasi-cyclic code over $\mathbb{Z}_k$, the ring of integers modulo k, reduces to a direct product of quasi-cyclic codes over $\mathbb{Z}_{p_i^{e_i}}$, $k = \prod_{i=1}^{s} p_i^{e_i}$, p_i a prime. Let T be the standard shift operator. A linear code $\mathcal{C}$ over a ring R is called an l-quasi-cyclic code if $T^l(c) \in \mathcal{C}$, whenever $c \in \mathcal{C}$. It is shown that if $(m, q) = 1$, $q = p^r$, p a prime, then an l-quasi-cyclic code of length lm over $\mathbb{Z}_q$ is a direct product of quasi-cylcic codes over some Galois extension rings of $\mathbb{Z}_q$. We have discussed about the structure of the generator of a 1-generator l-quasi-cyclic code of length lm over $\mathbb{Z}_q$. A method to obtain quasi-cyclic codes over $\mathbb{Z}_q$, which are free modules over $\mathbb{Z}_q$, has been discussed.

Keywords: Quasi-cyclic codes, circulant matrices, Galois rings, Hensel's lift.

1 Introduction

Quasi-cyclic codes form a remarkable generalization of cyclic codes [1], [5], [18]. They are asymptotically good as they meet a modified version of Gilbert-Varshamov bound [10]. They are closely linked to convolutional codes. Recently there has been a great interest for their applications in studying Turbo codes [19] and many Low-Density Parity Check (LDPC) codes [6]. It is well known that Turbo codes and LDPC codes have been proved to be capacity approaching codes.

In recent years, there has been a lot of interest in codes over finite rings since the revelation in 1994, in a breakthrough paper by Hammons et al. [7], that some non linear binary codes with very good parameters are actually binary images under a certain map, of some linear codes over $\mathbb{Z}_4$, the ring of integers modulo 4. Cyclic codes over some rings have been studied by a number of authors [2], [3], [15], [16], however, there has been a limited study on quasi-cyclic codes over rings [1], [12].

Let R be finite commutative ring with identity. A linear code $\mathcal{C}$ of length n over R is a submodule of the R-module R^n. Such a submodule is not necessarily

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 330–336, 2007.
© Springer-Verlag Berlin Heidelberg 2007

a free module. Let T be the standard shift operator. A linear code $\mathcal{C}$ over R is called a quasi-cyclic code of index l (or an l-quasi-cyclic code) if $T^l(c) \in \mathcal{C}$ whenever $c \in \mathcal{C}$. For $l = 1$, a quasi-cyclic code is simply a cyclic code. We assume that l divides the code length n and $n = lm$, for some positive integer m. Up to equivalence, the generator matrix of an l-quasi-cyclic code of length $n = lm$ can be expressed as a block matrix of $m \times m$ circulant matrices [4]. It is well known that a quasi-cyclic code of length $n = lm$ over R can be regarded as an $\frac{R[x]}{\langle x^m-1\rangle}$ -submodule of $\left(\frac{R[x]}{\langle x^m-1\rangle}\right)^l$ [1], [12]. If this module is generated by a single element $\boldsymbol{g(x)} \in \left(\frac{R[x]}{\langle x^m-1\rangle}\right)^l$, then we say that the code is a 1-generator quasi-cyclic code. The generator matrix of a 1-generator l-quasi-cyclic code of length $n = lm$ consists of a single row $G = [G_1, G_2, \ldots, G_l]$ of $m \times m$ circulant matrices. In this paper we consider some properties of quasi-cyclic codes over integer residue rings.

2 Quasi-cyclic Codes over $\mathbb{Z}_k$

Now we consider quasi-cyclic codes over $\mathbb{Z}_k$, the ring of integers modulo k. As usual, an l-quasi-cyclic code of length $n = lm$ over $\mathbb{Z}_k$ is a $\frac{\mathbb{Z}_k[x]}{\langle x^m-1\rangle}$-submodule of $\left(\frac{\mathbb{Z}_k[x]}{\langle x^m-1\rangle}\right)^l$. Let $k = \prod_{i=1}^{s} p_i^{e_i}$ be the canonical factorization of k. By the Chinese Remainder Theorem

$$\mathbb{Z}_k \cong \prod_{i=1}^{s} \mathbb{Z}_{p_i^{e_i}} \quad . \tag{1}$$

Which induces the isomorphism

$$\frac{\mathbb{Z}_k[x]}{\langle x^m-1\rangle} \cong \prod_{i=1}^{s} \frac{\mathbb{Z}_{p_i^{e_i}}[x]}{\langle x^m-1\rangle} \quad . \tag{2}$$

From (2), it follows that

$$\left(\frac{\mathbb{Z}_k[x]}{\langle x^m-1\rangle}\right)^l \cong \prod_{i=1}^{s} \left(\frac{\mathbb{Z}_{p_i^{e_i}}[x]}{\langle x^m-1\rangle}\right)^l \quad . \tag{3}$$

Every submodule of $\left(\frac{\mathbb{Z}_k[x]}{\langle x^m-1\rangle}\right)^l$ (l - quasi-cyclic code over $\mathbb{Z}_k$) is a direct product of submodules of $\left(\frac{\mathbb{Z}_{p_i^{e_i}}[x]}{\langle x^m-1\rangle}\right)^l$ (l - quasi-cyclic code over $\mathbb{Z}_{p_i^{e_i}}$), $i = 1, 2, \ldots, s$. Therefore, in order to study quasi-cyclic codes over $\mathbb{Z}_k$, it is sufficient to consider them over $\mathbb{Z}_{p^r}$, p a prime.

Let $q = p^r$, p a prime.

3 Quasi-cyclic Codes over $\mathbb{Z}_q$

Definition 1. *A polynomial $f(x) \in \mathbb{Z}_q[x]$ is said to be basic irreducible if $f(x)$ (modulo p) is irreducible in $\mathbb{Z}_p[x]$.*

Definition 2. *Let $f(x) \in \mathbb{Z}_q[x]$ is a monic basic irreducible polynomial of degree m. Then the Galois ring of degree m, over $\mathbb{Z}_q$, is defined by $GR(q,m) = \frac{\mathbb{Z}_q[x]}{\langle f(x)\rangle}$.*

Further, $GR(q,1) = \mathbb{Z}_q$ and $GR(p,m) = \mathbb{F}_{p^m}$, the finite field over $\mathbb{Z}_p$ with p^m elements. The Galois ring $R = GR(q,m) = GR(p^r,m)$ is a local ring with the maximal ideal $(p) = pR$ and the residue field $R/pR = \mathbb{F}_{p^m}$. If ξ is a root of $f(x)$, then $R = \mathbb{Z}_q(\xi)$.

Theorem 1 ([9]). *If $(m,q) = 1$, then $x^m - 1$ factorizes uniquely into monic pairwise coprime basic irreducible polynomials over $\mathbb{Z}_q$.*

Using Theorem (1) we can describe a structure of l-quasicyclic codes of length $n = lm$ over $\mathbb{Z}_q$. A more general structure of quasi-cyclic codes over finite chain rings has been discussed in [12].

Theorem 2. *If $(m,q) = 1$ then a quasi-cyclic code of length $n = lm$ and index l over $\mathbb{Z}_q$ is a direct product of linear codes over Galois extension rings of $\mathbb{Z}_q$.*

Proof. Let $\mathcal{C}$ be a quasi-cyclic code of length $n = lm$ and index l over $\mathbb{Z}_q$. Then $\mathcal{C}$ is a $\frac{\mathbb{Z}_q[x]}{\langle x^m-1\rangle}$ - submodule of $\left(\frac{\mathbb{Z}_q[x]}{\langle x^m-1\rangle}\right)^l$. Since $(m,q) = 1$, therefore, $x^m - 1$ factorizes uniquely into monic pairwise coprime basic irreducible polynomials over $\mathbb{Z}_q$. Let $x^m - 1 = f_1(x)f_2(x)...f_t(x)$ be the factorization of $x^m - 1$ into monic pairwise coprime basic irreducible polynomials over $\mathbb{Z}_q$. Then by the Chinese Remainder Theorem

$$\frac{\mathbb{Z}_q[x]}{\langle x^m - 1\rangle} = \prod_{i=1}^{t} \frac{\mathbb{Z}_q[x]}{\langle f_i(x)\rangle} \quad . \tag{4}$$

Since $f_i(x)$ is a basic irreducible polynomial in $\mathbb{Z}_q[x], i = 1,2,\ldots,t$, therefore, $\frac{\mathbb{Z}_q[x]}{\langle f_i(x)\rangle}$ is a Galois ring over $\mathbb{Z}_q$. Let $\frac{\mathbb{Z}_q[x]}{\langle f_i(x)\rangle} = GR(q,m_i)$, where $m_i = \deg(f_i(x))$. Then $\frac{\mathbb{Z}_q[x]}{\langle x^m-1\rangle} = \prod_{i=1}^{t} GR(q,m_i)$. From this, it follows that $\left(\frac{\mathbb{Z}_q[x]}{\langle x^m-1\rangle}\right)^l = \prod_{i=1}^{t} (GR(q,m_i))^l$. Then every submodule of $\left(\frac{\mathbb{Z}_q[x]}{\langle x^m-1\rangle}\right)^l$ is a direct product of submodules of $(GR(q,m_i))^l$. Since a submodule of $(GR(q,m_i))^l$ is a linear code of length l over $GR(q,m_i)$, therefore, $\mathcal{C}$ is a direct product of linear codes over the Galois rings $GR(q,m_i), i = 1,2,\ldots,t$. □

Thus, if $(m,q) = 1$ and $\mathcal{C}_i$ is a linear code of length l over the Galois ring $GR(q,m_i), i = 1,2,\ldots,t$, then $\prod_{i=1}^{t} \mathcal{C}_i$ is an l-quasi-cyclic code of length $n = lm$ over $\mathbb{Z}_q$ with $\prod_{i=1}^{t} |\mathcal{C}_i|$ codewords.

Example 1. Let $q = 2$, $l = 3$ and $m = 7$. Now $x^7 - 1 = (x-1)(x^3 + x + 1)(x^3 + x^2 + 1)$ is the factorization of $x^7 - 1$ into irreducible polynomials in $\mathbb{Z}_2[x]$. Let $\mathcal{C}$ be a 3-quasi-cyclic code of length 21 over $\mathbb{Z}_2$. Then $\mathcal{C} = \mathcal{C}_1 \bigoplus \mathcal{C}_2 \bigoplus \mathcal{C}_3$, where $\mathcal{C}_1$ is a linear code of length 3 over the Galois ring $GR(2,1) = \frac{\mathbb{Z}_2[x]}{\langle x-1\rangle} = \mathbb{Z}_2$, and $\mathcal{C}_2$ and $\mathcal{C}_3$ are linear codes of length 3 over the Galois ring $GR(2,3) = \frac{\mathbb{Z}_2[x]}{\langle x^3+x+1\rangle} = \frac{\mathbb{Z}_2[x]}{\langle x^3+x^2+1\rangle}$.

The structure of 1-generator quasi-cyclic codes over the finite fields is well known [17], [18]. We generalize the result for 1-generator quasi-cyclic codes over $\mathbb{Z}_q$.

Theorem 3. *Let $\mathcal{C}$ be a 1-generator l-quasi-cyclic code over $\mathbb{Z}_q$ of length $n = lm$. Then, a generator $\boldsymbol{g(x)} \in \left(\frac{\mathbb{Z}_q[x]}{\langle x^m-1\rangle}\right)^l$ of $\mathcal{C}$ has the form*

$$\boldsymbol{g(x)} = (g_1(x), g_2(x), \ldots, g_l(x))$$

where $g_i(x)$ is a generator polynomial of a cyclic code of length m over $\mathbb{Z}_q$.

Proof. Let $\mathcal{C}$ be a 1-generator l-quasi-cyclic code of length $n = lm$ over $\mathbb{Z}_q$. For a fixed i, $i = 1, 2, \ldots, l$, consider the projection $\left(\frac{\mathbb{Z}_q[x]}{\langle x^m-1\rangle}\right)^l \longrightarrow \frac{\mathbb{Z}_q[x]}{\langle x^m-1\rangle}$, defined by

$$(r_1(x), r_2(x), \ldots, r_l(x)) \longrightarrow r_i(x)$$

Let the image of $\mathcal{C}$ under this projection be $\mathcal{C}_i$, $i = 1, 2, \ldots, l$. Obviously $\mathcal{C}_i \subseteq \frac{\mathbb{Z}_q[x]}{\langle x^m-1\rangle}$. Now, since $\mathcal{C}$ is a $\frac{\mathbb{Z}_q[x]}{\langle x^m-1\rangle}$- submodule of $\left(\frac{\mathbb{Z}_q[x]}{\langle x^m-1\rangle}\right)^l$, therefore, for any $\boldsymbol{c(x)} = (c_1(x), c_2(x), \ldots, c_l(x)) \in \mathcal{C}$ and $a(x) \in \frac{\mathbb{Z}_q[x]}{\langle x^m-1\rangle}$, $a(x)\boldsymbol{c(x)} \in \mathcal{C}$. Consequently, for any $c_i(x) \in \mathcal{C}_i$, $a(x)c_i(x) \in \mathcal{C}_i$. It follows that $\mathcal{C}_i$ is an ideal of $\frac{\mathbb{Z}_q[x]}{\langle x^m-1\rangle}$. Therefore, $\mathcal{C}_i$ is a cyclic code of length m over $\mathbb{Z}_q$. Now, if $\boldsymbol{g(x)} = (g_1(x), g_2(x), \ldots, g_l(x))$ is a generator of $\mathcal{C}$, then the polynomial $g_i(x)$ generates $\mathcal{C}_i$, $i = 1, 2, \ldots, l$. Therefore $\mathcal{C}_i$ is a principal ideal of $\frac{\mathbb{Z}_q[x]}{\langle x^m-1\rangle}$ generated by $g_i(x)$. □

Now onward, we assume that $(m, q) = 1$. Then, $\frac{\mathbb{Z}_q[x]}{\langle x^m-1\rangle}$ is a principal ideal ring [3] [9]. Therefore, every cyclic code of length m over $\mathbb{Z}_q$ is generated by a single polynomial. However, unlike the finite field case, the generator polynomial of a cyclic code of length m over $\mathbb{Z}_q$ does not necessarily divide $x^m - 1$ in $\mathbb{Z}_q[x]$ [1], [20]. For the present case, (i. e. $(m, q) = 1$), an interesting class of quasi-cyclic codes occurs when the cyclic code $\mathcal{C}_i$, $i = 1, 2, \ldots, l$, in Theorem (3), is generated by a polynomial $g_i(x)$ dividing $x^m - 1$ in $\mathbb{Z}_q[x]$. $\mathcal{C}_i$ is then a free $\mathbb{Z}_q$ - module. In fact, the following result is true.

Theorem 4 ([22]). *Let $(n, q) = 1$ and $\mathcal{C}$ be a cyclic code of length n over $\mathbb{Z}_q$. Then $\mathcal{C}$ is a free $\mathbb{Z}_q$-module if and only if it is generated by a polynomial $g(x)$ which divides $x^n - 1$ in $\mathbb{Z}_q[x]$. In this case, the rank of $\mathcal{C}$ over $\mathbb{Z}_q$ is $n - deg\ g(x)$.*

In [20], it is shown that if a cyclic code $\mathcal{C}$ of length n over $\mathbb{Z}_4$ is generated by a polynomial $g(x) \mid (x^n - 1)$ then $g(x), xg(x), \ldots, x^{\mathrm{degg}(x)-1}g(x)$ form a basis for $\mathcal{C}$

over $\mathbb{Z}_4$. Applying the similar arguments, it is easy to show that if a cyclic code $\mathcal{C}$ of length n, $(n,q)=1$, over $\mathbb{Z}_q$, is generated by a polynomial $g(x) \mid (x^n-1)$ then $g(x), xg(x), \ldots, x^{\deg g(x)-1}g(x)$ form a basis for $\mathcal{C}$ over $\mathbb{Z}_q$. Further, as in the finite field case, it is straightforward to show that if a cyclic code of length n over $\mathbb{Z}_q$ is generated by a polynomial $g(x)$ dividing x^n-1 in $\mathbb{Z}_q[x]$, then it is also generated by a polynomial $f(x)g(x)$ with $\left(f(x), \frac{x^n-1}{g(x)}\right) = 1$ in $\mathbb{Z}_q[x]$. Therefore, for an l-quasi-cyclic code of length $n=lm$ over $\mathbb{Z}_q$ with a generator $\boldsymbol{g(x)} = (g_1(x), g_2(x), \ldots, g_l(x))$ with $g_i(x)$ dividing x^m-1 in $\mathbb{Z}_q[x]$, we can also take its generator as $\boldsymbol{g(x)} = (f_1(x)g_1(x), f_2(x)g_2(x), \ldots, f_l(x)g_l(x))$ with $\left(f_i(x), \frac{x^m-1}{g_i(x)}\right) = 1$.

Now we consider 1-generator l-quasi-cyclic codes of length $n=lm$ with generators of the form $\boldsymbol{g(x)} = (f_1(x)g(x), f_2(x)g(x), \ldots, f_l(x)g(x))$ with $g(x)$ dividing x^m-1 and $\left(f_i(x), \frac{x^m-1}{g(x)}\right) = 1$. In [1] it is proved that a 1-generator l-quasi-cyclic code of length $n=lm$, m odd, over $\mathbb{Z}_4$, with a generator of the above form is a free $\mathbb{Z}_4$-module of rank $m - \deg g(x)$. We generalize this result to quasi-cyclic codes over $\mathbb{Z}_q$.

Theorem 5. *Let $\mathcal{C}$ be a 1-generator l-quasi-cyclic code of length $n=lm$ over $\mathbb{Z}_q$ with a generator of the form*

$$\boldsymbol{g(x)} = (f_1(x)g(x), f_2(x)g(x), \ldots, f_l(x)g(x))$$

where $g(x) \mid (x^m-1)$ and $\left(f_i(x), \frac{x^m-1}{g(x)}\right) = 1$ for $i = 1, 2, \ldots, l$. Then $\mathcal{C}$ is a free $\mathbb{Z}_q$-module of rank $m-\deg\, g(x)$.

Proof. Once again, for a fixed i, $i = 1, 2, \ldots, l$, consider the projection $\left(\frac{\mathbb{Z}_q[x]}{\langle x^m-1\rangle}\right)^l \longrightarrow \frac{\mathbb{Z}_q[x]}{\langle x^m-1\rangle}$, defined by

$$(r_1(x), r_2(x), \ldots, r_l(x)) \longrightarrow r_i(x)$$

Let the image of $\mathcal{C}$ under this projection be $\mathcal{C}_i$, $i = 1, 2, \ldots, l$. Then, from Theorem (3), we know that $\mathcal{C}_i$ is a cyclic code of length m over $\mathbb{Z}_q$, generated by the polynomial $g_i(x) = g(x)$ for each $i = 1, 2, ..., l$. Now if a relation of the form $\sum_i a_i x^i g(x) = 0$, with $a_i \in \mathbb{Z}_q$ exists on $\frac{\mathbb{Z}_q[x]}{\langle x^m-1\rangle}$, then a similar relation $\sum_i a_i x^i \boldsymbol{g(x)} = 0$, with $a_i \in \mathbb{Z}_q$, holds in $\left(\frac{\mathbb{Z}_q[x]}{\langle x^m-1\rangle}\right)^l$. Also if $\sum_i a_i x^i g(x) \neq 0$ with $a_i \in \mathbb{Z}_q$ in $\frac{\mathbb{Z}_q[x]}{\langle x^m-1\rangle}$ then so is $\sum_i a_i x^i \boldsymbol{g(x)} \neq 0$ in $\left(\frac{\mathbb{Z}_q[x]}{\langle x^m-1\rangle}\right)^l$. Thus $g(x), xg(x), \ldots, x^s g(x)$, for some $s \geq 0$, are linearly independent over $\mathbb{Z}_q$, if and only if, $\boldsymbol{g(x)}, x\boldsymbol{g(x)}, \ldots, x^s\boldsymbol{g(x)}$ are linearly independent over $\mathbb{Z}_q$. Now $g(x)$ is a generator polynomial of the cyclic code $\mathcal{C}_i$ and $g(x)$ divides x^m-1 over $\mathbb{Z}_q$. Therefore, from Theorem (4), $\mathcal{C}_i$ is a free $\mathbb{Z}_q$-module and, $g(x), xg(x), \ldots, x^{m-(\deg g(x)-1)}g(x)$ form a basis for $\mathcal{C}_i$ over $\mathbb{Z}_q$. From above discussion, it follows that $\boldsymbol{g(x)}, x\boldsymbol{g(x)}, \ldots, x^{m-(\deg g(x)-1)}\boldsymbol{g(x)}$ form a basis for $\mathcal{C}$ over $\mathbb{Z}_q$. Hence $\mathcal{C}$ is a free $\mathbb{Z}_q$-module of rank $m - \deg g(x)$. □

Now we consider a method of generating quasi-cyclic codes over $\mathbb{Z}_q$.

Theorem 6 ([3]). *If $f_1(x) \in \mathbb{Z}_p[x]$ is a monic irreducible divisor of $x^m - 1$ over $\mathbb{Z}_p$, then there is a unique monic irreducible polynomial $f(x) \in \mathbb{Z}_q[x]$ which divides $x^m - 1$ over $\mathbb{Z}_q$ such that $f(x) \pmod p = f_1(x)$.*

The polynomial $f(x)$ in Theorem 6 is called the Hensel's lift of the polynomial $f_1(x)$.

Hensel's lift of a polynomial $f_1(x) \in \mathbb{Z}_2[x]$ to $\mathbb{Z}_4[x]$, can be obtained by the Graeffe's method [7], [8], described below. Let $f_1(x) = e(x) + o(x)$, where $e(x)$ contains only even powers of x and $o(x)$ contains only odd powers of x. Then the Hensel's lift $f(x) \in \mathbb{Z}_4[x]$ of $f_1(x)$ is obtained by $f(x) = \pm(e(x)^2 - o(x)^2)$, where the sign $\pm$ is chosen in such a way that the coefficient of the highest power of x is 1. This is illustrated by the following example.

Example 2. We know that, $x^7 - 1 = (x-1)(x^3 + x + 1)(x^3 + x^2 + 1)$ is the factorization of $x^7 - 1$ into irreducible polynomials in $\mathbb{Z}_2[x]$. Let $f_1(x) = x^3 + x^2 + 1$. Then $e(x) = x^2 + 1$ and $o(x) = x^3$. Let $f(x)$ be the Hensel's lift to $\mathbb{Z}_4[x]$, of $f_1(x)$. Then, $f(x^2) = -((x^2+1)^2 - (x^3)^2) = x^6 - x^4 + 2x^2 - 1$ and thus $f(x) = x^3 - x^2 + 2x - 1$.

To generate a 1-generator l-quasi-cyclic code $\mathcal{C}$ of length $n = lm$ over $\mathbb{Z}_q$ with a generator of the form $\boldsymbol{g(x)} = (f_1(x)g(x), f_2(x)g(x), \ldots, f_l(x)g(x))$, with $g(x) \mid (x^m - 1)$ and $\left(f_i(x), \frac{(x^m-1)}{g(x)}\right) = 1$, $i = 1, 2, \ldots, l$, we require a polynomial $g(x)$ which divides $x^m - 1$ over $\mathbb{Z}_q$. Such a polynomial can be obtained by the following procedure. Let $g_p(x)$ be a factor of $x^m - 1$ in $\mathbb{Z}_p[x]$. Let $g(x)$ be the Hensel's lift to $\mathbb{Z}_q[x]$ of $g_p(x)$. Then $g(x)$ divides $x^m - 1$ over $\mathbb{Z}_q$. Now $g(x)$ can be used to find the generator $\boldsymbol{g(x)}$ for $\mathcal{C}$.

Example 3. Let $q = 4$, $l = 3$ and $m = 7$. Now $x^7 - 1 = (x-1)(x^3 + x + 1)(x^3 + x^2 + 1)$ is the factorization of $x^7 - 1$ into irreducible polynomials in $\mathbb{Z}_2[x]$. Let $g_2(x) = x^3 + x + 1$. Then the Hensel's lift of $g_2(x)$, to $\mathbb{Z}_4[x]$, is the polynomial $g(x) = x^3 + 2x^2 + x - 1$. Let $f_1(x) = 1$, $f_2(x) = x + 1$ and $f_3(x) = x^3 + x$. Then, $\boldsymbol{g(x)} = (f_1(x)g(x), f_2(x)g(x), f_3(x)g(x)) = (x^3 + 2x^2 + x - 1, (x+1)(x^3 + 2x^2 + x - 1), (x^3 + x)(x^3 + 2x^2 + x - 1)) = (x^3 + 2x^2 + x - 1, x^4 + 3x^3 + 3x^2 - 1, x^6 + 2x^5 + 2x^4 + x^3 + x^2 - x)$, which generates a 3-quasi-cyclic code of length 21 and rank 4 over $\mathbb{Z}_4$.

4 Conclusion

In this paper we have considered some quasi-cyclic codes over integer residue rings. The study of quasi-cyclic codes over integer residue ring basically reduces to quasi-cyclic codes over $\mathbb{Z}_q$, $q = p^r$, p a prime. A decomposition of quasi-cyclic codes over $\mathbb{Z}_q$ into quasi-cyclic codes over some Galois extension rings of $\mathbb{Z}_q$ is given. Some properties of 1-generator quasi-cyclic codes over $\mathbb{Z}_q$ have been discussed.

References

1. Aydin, N., Ray-Chaudhuri, D.K.: Quasi-Cyclic Codes over $\mathbb{Z}_4$ and Some New Binary Codes. IEEE Trans. Inform. Theory 48, 2065–2069 (2002)
2. Blake, I.F.: Codes Over Certain Rings. Inform. Control 20, 396–404 (1972)
3. Calderbank, A.R., Sloane, N.J.A.: Modular and P-Adic Cyclic Codes. Designs Codes and Cryptography 6, 21–35 (1995)
4. Chen, C.L., Peterson, W.W., Weldon, E.J.: Some Results on Quasi-Cyclic Codes. Inform. Control 15, 407–423 (1969)
5. Daskalov, R., Hristov, P.: New Binary One-Generator Quasi-Cyclic Codes. IEEE Trans. Inform. Theory 49, 3001–3005 (2003)
6. Fossorier, M.P.C.: Quasi-Cyclic Low-Density Parity-Check Codes From Circulant Permutation Matrices. IEEE Trans. Inform. Theory 50, 1788–1793 (2004)
7. Hammons, A.R., Kumar, P.V., Calderbank, A.R., Sloane, N.J.A., Sole, P.: The $\mathbb{Z}_4$-Lnearity of Kerdock, Preparata, Goethal and Related Codes. IEEE Trans. Inform. Theory 40, 301–319 (1994)
8. Huffman, W.C., Pless, V.: Fundamentals Of Error Correcting Codes. Cambridge University Press, Cambridge (2003)
9. Kanwar, P., Lopez-Permouth, S.R.: Cyclic Codes Over the Integers Modulo p^m. Finite Fields and Their Applications 3, 334–352 (1997)
10. Kasami, T.: A Gibert-Varshamov Bound for Quasi-Cyclic Codes of Rate 1/2. IEEE Trans. Inform. Theory 20, 679–680 (1974)
11. Ling, S., Sole, P.: On the Algebraic Structures of Quasi-Cyclic Codes I: Finite Fields. IEEE Trans. Inform. Theory 47, 2751–2760 (2001)
12. Ling, S., Sole, P.: On the Algebraic Structures of Quasi-Cyclic Codes II: Chain Rings. Designs Codes and Cryptography 30, 113–130 (2001)
13. Ling, S., Sole, P.: On the Algebraic Structures of Quasi-Cyclic Codes III: Generator Theory. IEEE Trans. Inform. Theory 51, 2692–2700 (2005)
14. Macwilliams, F.J., Sloane, N.J.A.: The Theory Of Error Correcting Codes. North Holland Publishing Company, Amsterdam (1977)
15. Norton, G.H., Salagean-Mandache, A.: On the Structures of Linear and Cyclic Codes Over Finite Chain Rings. Applicable Algebra in Engineering Communication and Computing 10, 489–506 (2000)
16. Pless, V., Quian, J.: Cyclic Codes and Quadratic Residue Codes Over $\mathbb{Z}_4$. IEEE Trans. Of Inform. Theory 42, 1594–1600 (1996)
17. Seguin, G.E.: A Class of 1-Generator Quasi-Cyclic Codes. IEEE Trans. Inform. Theory 50, 1745–1753 (2004)
18. Siap, I., Aydin, N., Ray-Chaudhuri, D.K.: New Ternary Quasi-Cyclic Codes with Better Minimum Distances. IEEE Trans. Inform. Theory 46, 1554–1558 (2000)
19. Tanner, R.M.: Towards Algebraic Theory of Turbo Codes. In: 2nd Int. Symp. on Turbo Codes, Brest (2000)
20. Wan, Z.X.: Quaternary Codes. World Scientific, Singapore (1997)
21. Wasan, S.K.: On Codes over $\mathbb{Z}_m$. IEEE Trans. Inform. Theory 28, 117–120 (1982)
22. Woo, S.S.: Free Cyclic Codes Over Finite Local Rings. Bull. Korean Math. Soc. 43, 723–735 (2006)

Extended Norm-Trace Codes with Optimized Correction Capability*

Maria Bras-Amorós[1,**] and Michael E. O'Sullivan[2]

[1] Departament d'Enginyeria Informàtica i Matemàtiques
Universitat Rovira i Virgili
maria.bras@urv.cat
[2] Department of Mathematics and Statistics
San Diego State University
mosulliv@sciences.sdsu.edu

Abstract. We consider a generalization of the codes defined by norm and trace functions on finite fields introduced by Olav Geil. The codes in the new family still satisfy Geil's duality properties stated for norm-trace codes. That is, it is easy to find a minimal set of parity checks guaranteeing correction of a given number of errors, as well as the set of monomials generating the corresponding code. Furthermore, we describe a way to find the minimal set of parity checks and the corresponding generating monomials guaranteeing correction at least of *generic* errors. This gives codes with even larger dimensions.

1 Introduction

Let q be a prime power and r an integer greater than or equal to 2. The curve defined over $\mathbb{F}_{q^r}$ with affine equation $x^{\frac{q^r-1}{q-1}} = y^{q^{r-1}} + y^{q^{r-2}} + \cdots + y$ is called the *norm-trace curve* associated to q and r. In fact, the defining equation is equivalent to $\mathcal{N}_{\mathbb{F}_{q^r}/\mathbb{F}_q}(x) = \mathcal{T}_{\mathbb{F}_{q^r}/\mathbb{F}_q}(y)$, where, for $x \in \mathbb{F}_{q^r}$, $\mathcal{N}_{\mathbb{F}_{q^r}/\mathbb{F}_q}(x)$ denotes the norm of x over $\mathbb{F}_q$, and for $y \in \mathbb{F}_{q^r}$, $\mathcal{T}_{\mathbb{F}_{q^r}/\mathbb{F}_q}(y)$ denotes the trace of y over $\mathbb{F}_q$. Norm-trace curves were introduced by Geil in [2]. They are a natural generalization of Hermitian curves, these being norm-trace curves resulting from the field extension $\mathbb{F}_{q^2}/\mathbb{F}_q$. Norm-trace curves have a single rational point P_∞ at infinity and $n = q^{2r-1}$ proper rational points.

In this work we consider a somewhat broader family of curves that include norm-trace curves. For example, our family includes $x^u = \mathcal{T}_{\mathbb{F}_{q^r}/\mathbb{F}_q}(y)$, where u divides $(q^r-1)/(q-1)$, and it also includes the maximal curves derived from Hermitian curves studied in [3].

* Part of this work is in the manuscript [1] submitted for publication.

** This work was partly supported by the Catalan Government through a grant BE–2 2006 and a grant 2005 SGR 00446 and by the Spanish Ministry of Education through projects TSI2007-65406-C03-01 E-AEGIS and CONSOLIDER CSD2007-00004 ARES.

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 337–346, 2007.
© Springer-Verlag Berlin Heidelberg 2007

One can check that the new family of curves also fit the conditions for the list decoding algorithm for Hermitian codes using Gröbner basis presented by Lee and O'Sullivan [4].

The codes we obtain from these curves can be seen as codes from order domains. The theory of codes from order domains is developed in [5,6,7,8,9]. For these codes there is a lower bound on the minimum distance and an efficient generalization of the Berlekamp-Massey algorithm based on Sakata's algorithm for finding linear recurrence relations in multidimensional arrays [10]. Using the majority voting method of Feng, Rao, and Duursma, [11,12] the algorithm corrects up to half the above mentioned bound on the minimum distance. Improvements to code dimension, while maintaining a fixed decoding radius were discovered by Feng and Rao [11]. Our earlier work [13] considered a different improvement, based on the observation that the decoding algorithm corrects an error vector based not so much on the weight of the vector but rather the "footprint" of the error locations [14]. For some codes, most error vectors of a given weight t have a particular footprint. Those error vectors are called *generic* and are correctable with fewer check symbols than are required to correct all error vectors of weight t. These results may be combined with the Feng-Rao improvements to increase the dimension of a code while maintaining a desired correction capability.

We show how the codes from the new family of curves still satisfy Geil's duality properties stated for norm-trace codes [2]. That is, it is easy to find a minimal set of parity checks guaranteeing correction of a given number of errors, as well as the set of monomials generating the corresponding code.

Furthermore, following the results in [13] we describe a way to find the minimal set of parity checks and the corresponding generating monomials guaranteeing correction at least of generic errors. This gives codes with larger dimensions.

2 Codes from Extended Norm-Trace Curves

A linearized polynomial over $\mathbb{F}_q$—also called a q polynomial—is a polynomial over $\mathbb{F}_q$ whose terms all have degree a power of q [15, §3.4]. Let $L(y) = \sum_{i=0}^{d} a_i y^{q^i}$ be a linearized polynomial such that a_0, a_d are nonzero and such that $L(y) = 0$ has q^d distinct solutions in $\mathbb{F}_{q^r}$. Then L gives a q^d-to-one map from $\mathbb{F}_{q^r}$ into itself [15].

Let η be a primitive element of $\mathbb{F}_{q^r}$. Let v be any factor of $q^r - 1$ and let D be the powers of η^v, along with 0: $D = \{0\} \cup \{\eta^{vm} : m \in \{1, \ldots, \frac{q^r-1}{v}\}\}$. We will assume that $L(\mathbb{F}_{q^r}) \supseteq D$. Then for any u dividing v we consider the curve $x^u = L(y)$ whose coordinate ring is $A = \dfrac{\mathbb{F}_{q^r}[x, y]}{x^u - L(y)}$.

A basis of A as a vector space over $\mathbb{F}_{q^r}$ is given by the images in A of the set of monomials $\mathcal{B} = \{x^a y^b : b < q^d\}$. Consider $\ll$ the total ordering on A determined by the (q^d, u) weighted degree, $\deg_{q^d,u} x^a y^b = q^d a + ub$. One can check that any monomial in $\mathbb{F}_q[x, y]$ has the same weighted degree as exactly one monomial in $\mathcal{B}$. In particular, no two monomials from $\mathcal{B}$ have the same weighted degree. Let z_i be the ith monomial in $\mathcal{B}$ with respect to $\ll$, starting with $z_0 = 1$. Notice that

if z_i divides z_j then $i \leqslant j$. We say that a set of monomials W is *divisor-closed* if $z_i \in W$ whenever z_i divides a monomial in W.

Since u is coprime to q, this curve has a unique point at infinity. We consider all points (α, β) on the curve such that $\alpha^u = L(\beta) \in D$. Thus we have α is either 0 or $\alpha \in \{\eta^{mv/u} : m \in \{1, \ldots, \frac{(q^r-1)u}{v}\}\}$. There are $n = q^d(\frac{(q^r-1)u}{v} + 1)$ such points, $P_1, \ldots, P_n$.

Let $\varphi : A \longrightarrow \mathbb{F}_q^n$ be such that $f \mapsto (f(P_1), \ldots, f(P_n))$ and let W be a set of monomials in $\mathcal{B}$. We call E_W the linear code spanned by $\{\varphi(z_i) : z_i \in W\}$ and C_W the dual code.

Given W define W_φ as the set $\{z_i \in W : \varphi(z_i)$ is not in the span of $\varphi(z_j), j \in W, j < i\}$ and let $\mathcal{M} = \{x^a y^b : a \leqslant \frac{(q^r-1)u}{v}, b < q^d\}$. The next proposition shows how the set W defining a code can be reduced to its intersection with $\mathcal{M}$. For the sake of brevity, the proof of this proposition has been omitted.

Proposition 1. – *If $W \subseteq \mathcal{M}$ then $W_\varphi = W$.*
– *If W is divisor-closed then $W_\varphi = W \cap \mathcal{M}$.*

The next proposition shows how to find the set $W^\perp$ defining the dual code of the code defined by W. As before, for the sake of brevity, the proof of this proposition has been omitted. The proofs of both propositions is in [1].

Proposition 2. *Suppose that the prime divisor of q also divides $v-u$. If $W \in \mathcal{M}$ is divisor-closed then $C_W = E_{W^\perp}$ where*

$$W^\perp = \mathcal{M} \setminus \{x^{\frac{(q^r-1)u}{v}-a} y^{q^d-1-b} : x^a y^b \in W\}.$$

Example 1. Let q be a prime power, let i, j, k be positive integers with $j, k > 1$ and let r be a common multiple of ij and k. Consider the curve with affine equation $x^{\frac{q^{ij}-1}{q^i-1}} = y^{q^{k-1}} + y^{q^{k-2}} + \cdots + y^{q^2} + y^q + y$. We may write this as $\mathcal{N}_{\mathbb{F}_{q^{ij}}/\mathbb{F}_{q^i}}(x) = \mathcal{T}_{\mathbb{F}_{q^k}/\mathbb{F}_q}(y)$ which is illustrated in the diagram of fields in Figure 1.

This curve matches the former scheme with $v = \frac{q^r-1}{q-1}$ and $u = \frac{q^{ij}-1}{q^i-1}$. In this case $D = \{0\} \cup \{\eta^{vm} : m \in \{1, \ldots, q-1\}\} = \mathbb{F}_q$, and we can verify that u divides v since $v/u = \frac{q^r-1}{q-1} \cdot \frac{q^i-1}{q^{ij}-1} = \frac{q^r-1}{q^{ij}-1} \cdot \frac{q^i-1}{q-1}$ is an integer. Furthermore,

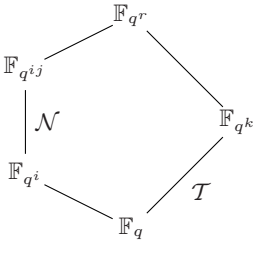


Fig. 1. Field extensions defining the curve of Example 1

both u and v are 1 modulo q so q divides $v-u$. Thus the assumptions of this section are fulfilled. Because of our choice of D, the only points on the curve that we are considering are in $\mathbb{F}_{q^{ij}} \times \mathbb{F}_{q^k}$ and there are $q^{k-1}(1+(q-1)u)$ of them. There may be other points on the curve, but we have not been able to show that the duality properties of this section can be extended to a larger set of points.

Norm-trace curves as defined in [2] correspond to the curves in this example when $i=1$ and $j=k=r$, and Hermitian curves correspond to the case when $i=1$ and $j=k=2$. Other curves can be obtained which are not Hermitian curves or norm-trace curves as in [2]. In Table 1 we list the possible norm and trace functions for all the extensions of finite fields contained in $\mathbb{F}_{2^8}$.

As an example, the curve with affine equation

$$x^3 = y^8 + y^4 + y^2 + y$$

over $\mathbb{F}_{2^4}$ is obtained taking $q=2$, $i=1$, $j=2$, $k=4$ and $r=4$. In this case $u=3$ and $v=15$ and the number of points in $\mathbb{F}_4 \times \mathbb{F}_{16}$ is 32. Considering $\mathbb{F}_{16} = \mathbb{Z}_2/(X^4+X+1)$ and taking $\alpha = [X]$, these 32 points are $(0,0)$, $(0,1)$, $(0,\alpha)$, $(0,\alpha^2)$, $(0,\alpha^4)$, $(0,\alpha^5)$, $(0,\alpha^8)$, $(0,\alpha^{10})$, $(1,\alpha^3)$, $(1,\alpha^6)$, $(1,\alpha^7)$, $(1,\alpha^9)$, $(1,\alpha^{11})$, $(1,\alpha^{12})$, $(1,\alpha^{13})$, $(1,\alpha^{14})$, (α^5,α^3), (α^5,α^6), (α^5,α^7), (α^5,α^9), (α^5,α^{11}), (α^5,α^{12}), (α^5,α^{13}), (α^5,α^{14}), (α^{10},α^3), (α^{10},α^6), (α^{10},α^7), (α^{10},α^9), $(\alpha^{10},\alpha^{11})$, $(\alpha^{10},\alpha^{12})$, $(\alpha^{10},\alpha^{13})$, $(\alpha^{10},\alpha^{14})$.

Table 1. For the field $\mathbb{F}_{256}$ the norm and trace functions appearing in Example 1 are listed. The curves considered are obtained for a fixed q by setting any norm function equal to any trace function.

$q=2$			
i,j	Norm functions on x	k	Trace functions on y
1,2	x^3	2	y^2+y
1,4	x^{15}	4	$y^8+y^4+y^2+y$
1,8	x^{255}	8	$y^{128}+y^{64}+y^{32}+y^{16}+$
2,2	x^5		$+y^8+y^4+y^2+y$
2,4	x^{85}		
4,2	x^{17}		

$q=4$			
i,j	Norm functions on x	k	Trace functions on y
1,2	x^5	2	y^4+y
1,4	x^{85}	4	$y^{64}+y^{16}+y^4+y$
2,2	x^{17}		

$q=16$			
i,j	Norm functions on x	k	Trace functions on y
1,2	x^{17}	2	$y^{16}+y$

3 Correction-Capability-Optimized Evaluation Codes

Given a field $\mathbb{F}$ and an $\mathbb{F}$-algebra A, an *order function* on A is a map $\rho : A \longrightarrow \mathbb{N}_{-1}$ which satisfies: i) the set $L_m = \{f \in A : \rho(f) \leqslant m\}$ is an $m+1$ dimensional vector space over $\mathbb{F}$; ii) if $f, g, z \in A$ and z is nonzero then $\rho(f) > \rho(g) \Longrightarrow \rho(zf) > \rho(zg)$ [6,5,16]. The pair A, ρ is often called an order domain. It is easy to show that ρ must be surjective. The ring A defined in Section 2 admits an order function in which z_i has order i. That is, we may define ρ by i) $\rho(0) = -1$, ii) $\rho(z_i) = i$, iii) $\rho(\sum_{i \in I} a_i z_i) = \max I$, where the $a_i's$ are assumed to be nonzero. Notice that in this case two elements f, g in A have the same order if and only if $v_{P_\infty}(f) = v_{P_\infty}(g)$.

Given an order function ρ an operation $\oplus$ in $\mathbb{N}_0$ can be well defined by $i \oplus j = \rho(fg)$ where f and g are such that $\rho(f) = i$ and $\rho(g) = j$. In our example, $i \oplus j = k$ is equivalent to $v_{P_\infty}(z_i) + v_{P_\infty}(z_j) = v_{P_\infty}(z_k)$. In fact $\mathbb{N}_0, \oplus$ is a commutative semigroup. We can define a partial ordering $\preccurlyeq$ on $\mathbb{N}_0$ by setting $i \preccurlyeq j$ if and only if there exists $k \in \mathbb{N}_0$ such that $i \oplus k = j$. When $i \preccurlyeq j$ we must also have $i \leqslant j$. An important parameter for decoding is $\nu_i = |\{j \in \mathbb{N}_0 : j \preccurlyeq i\}|$.

Codes from order domains are defined by means of a surjective map $\varphi : A \longrightarrow \mathbb{F}^n$. Consider a basis $\mathcal{B}$ of A with an element of each order. Given a subset W of $\mathcal{B}$, define the *order-prescribed evaluation code* related to W as the $\mathbb{F}$-subspace E_W generated by $\{\varphi(z_i) : z_i \in W\}$ and define C_W to be its dual code. For the codes considered here we take φ as the map which takes $f \in A$ to $(f(P_1), \ldots, f(P_n))$, where $P_1, \ldots, P_n$ are the points of the curve different than the one at infinity.

The subsets W can be defined in order to achieve optimal correction capability. The two results on decoding performance that we need are

Theorem 1. *[11] All error vectors of weight t can be corrected by C_W if W contains all elements z_i with $\nu_i < 2t + 1$.*

Theorem 2. *[13] All generic error vectors of weight t can be corrected by C_W if W contains all elements z_i with $i \notin \{j \oplus k : j, k \geqslant t\}$.*

Theorem 1 can be used to design an optimal order-prescribed evaluation code correcting t errors. Indeed, take $\widetilde{R}(t) = \{z_i \in \mathbb{N}_0 : \nu_i < 2t + 1\}$ and use the code $C_{\widetilde{R}(t)}$. This construction is due to Feng and Rao [11]. Theorem 2 can be used to design an optimal order-prescribed evaluation code correcting all generic errors of weight t. Indeed, take $C_{\widetilde{R}^*(t)}$ where $\widetilde{R}^*(t)$ is $\mathbb{N}_0 \setminus \{i \oplus j : i, j \geqslant t\}$. This construction was introduced in [13].

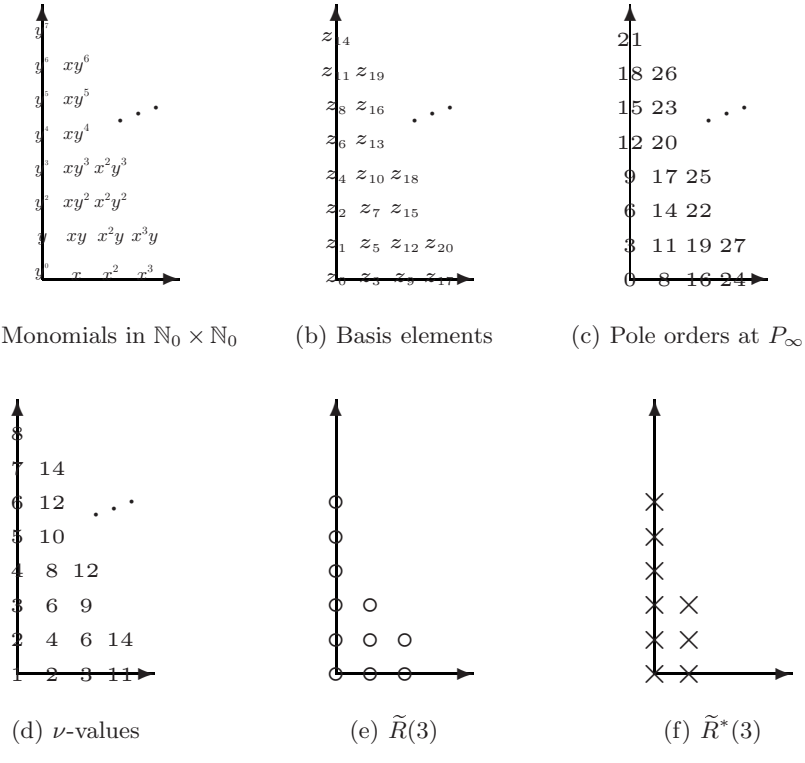
Example 2. Consider the codes over the curve with affine equation $x^3 = y^8 + y^4 + y^2 + y$ over $\mathbb{F}_{16}$. The monomials in A are ordered by their $(q^d, u) = (8, 3)$ graded degree, which in turn is the pole order of each monomial at infinity. Thus, $z_0 = 1$, $z_1 = y$, $z_2 = y^2$, $z_3 = x$, $z_4 = y^3$, $z_5 = xy$, etc.

The parity checks $\varphi(x^a y^b)$ can be represented by the corresponding monomials $x^a y^b$ and each monomial $x^a y^b$ can be represented by the point with coordinates (a, b) in the $\mathbb{N}_0 \times \mathbb{N}_0$ grid. This is illustrated in Figure 2(a) and Figure 2(b).

Figure 2(c) represents the pole order at infinity of each monomial represented in Figure 2(a). In this case the ν-value corresponding to the monomial z_i is the number of monomials z_j, z_k with $v_{P_\infty}(z_j) + v_{P_\infty}(z_k) = v_{P_\infty}(z_i)$. Figure 2(d) represents these ν-values.

Suppose we want to correct 3 errors. Theorem 1 says that the minimum set of parity checks that we need corresponds exactly to those monomials whose ν-value is at most 6. This gives the set $\widetilde{R}(3) = \{ z_0 = 1, z_1 = y, z_2 = y^2, z_3 = x, z_4 = y^3, z_5 = xy, z_6 = y^4, z_7 = xy^2, z_8 = y^5, z_9 = x^2, z_{12} = x^2 y\}$. These monomials are represented in Figure 2(e).

If we just want to guarantee correction of generic errors, Theorem 2 says that the minimum set of parity checks that we need corresponds exactly to those monomials whose pole order at infinity is not the sum of the pole orders of two monomials in $z_3, z_4, z_5, \ldots$. This gives the set $\widetilde{R}^*(3) = \{z_0 = 1, z_1 = y, z_2 = y^2, z_3 = x, z_4 = y^3, z_5 = xy, z_6 = y^4, z_7 = xy^2, z_8 = y^5\}$. These monomials are represented in Figure 2(f). Notice that $\widetilde{R}^*(3) \subseteq \widetilde{R}(3)$ and that $\widetilde{R}(3)$ has two more monomials than $\widetilde{R}^*(3)$.



(a) Monomials in $\mathbb{N}_0 \times \mathbb{N}_0$ (b) Basis elements (c) Pole orders at P_∞

(d) ν-values (e) $\widetilde{R}(3)$ (f) $\widetilde{R}^*(3)$

Fig. 2.

Dimension and Generating Matrices of Correction-Capability-Optimized Codes from Extended Norm-Trace Curves

In this subsection we see how the sets of check monomials giving correction-capability-optimized codes from extended norm-trace curves behave well in the sense that they are divisor-closed. Hence, we can find nice ways to determine the actual dimensions of the codes, their parity-check matrices and their generating matrices.

Lemma 1. *The sets $\widetilde{R}(t)$ and $\widetilde{R}^*(t)$ are divisor-closed.*

Proof. Notice that for a subset W being divisor-closed is equivalent to being $\preccurlyeq$-closed, that is, $i \in W$ for all $i \preccurlyeq j$ with $j \in W$. If $i \preccurlyeq j$ and $j \in \widetilde{R}(t)$ then $\nu_i \leqslant \nu_j < 2t+1$, so $i \in \widetilde{R}(t)$. Thus $\widetilde{R}(t)$ is closed under $\preccurlyeq$. To prove that $\widetilde{R}^*(t)$ is closed under $\preccurlyeq$ notice that, if $i \preccurlyeq j$ then $j = i \oplus s$ for some $s \in \mathbb{N}_0$. Suppose $i \notin \widetilde{R}^*(t)$. Then $i = k \oplus l$ with $k, l \geqslant t$ and $j = i \oplus s = k \oplus (l \oplus s)$ and so $j \notin \widetilde{R}^*(t)$.

Corollary 1. $\widetilde{R}_\varphi(t) = \widetilde{R}(t) \cap \mathcal{M}$, $\widetilde{R}^*_\varphi(t) = \widetilde{R}^*(t) \cap \mathcal{M}$.

Corollary 2. $C_{\widetilde{R}(t)} = C_{\widetilde{R}_\varphi(t)} = E_{\widetilde{R}_\varphi(t)^\perp}$, $C_{\widetilde{R^*}(t)} = C_{\widetilde{R^*}_\varphi(t)} = E_{\widetilde{R}^*_\varphi(t)^\perp}$.

Example 3. Consider the codes $C_{\widetilde{R}(3)}$ and $C_{\widetilde{R}^*(3)}$ over the curve $x^3 = y^8 + y^4 + y^2 + y$ represented in Figure 2(e) and Figure 2(f). In this case $\frac{(q^4-1)u}{v} = 3$ and $q^d - 1 = 7$, so $\mathcal{M} = \{x^a y^b : 0 \leqslant a \leqslant 3, 0 \leqslant b \leqslant 7\}$ (see Figure 3(a)). Since all checks in $\widetilde{R}(3)$ (resp. $\widetilde{R}^*(3)$) are inside $\mathcal{M}$, by Corollary 1, they are all linearly independent. So the dimension of $C_{\widetilde{R}(3)}$ is $32 - 11 = 21$ and the dimension of $C_{\widetilde{R}^*(3)}$ is $32 - 9 = 23$.

Now we can use Corollary 2 to derive, from $\widetilde{R}(3)$ and $\widetilde{R}^*(3)$, the set of monomials $x^a y^b$ such that the vectors $\varphi(x^a y^b)$ generate $C_{\widetilde{R}(3)}$ and $C_{\widetilde{R}^*(3)}$. In Figure 3(b) we represented all these sets. In Figure 4 and Figure 5 we give the explicit parity check matrices and generating matrices for these codes.

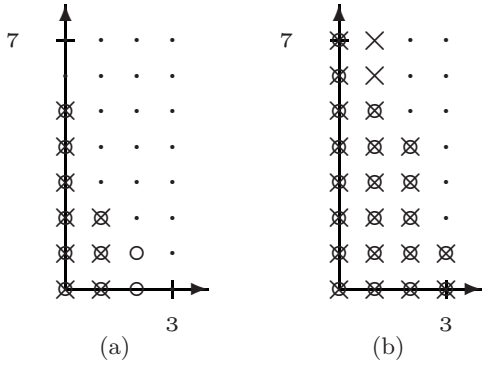


Fig. 3. Obtaining $\widetilde{R}(3)^\perp$ and $R^*_\varphi(3)^\perp$ from $\widetilde{R}_\varphi(3)$ and $R^*_\varphi(3)$, respectively

Fig. 4. Parity check matrix (above) and generating matrix (below) of $C_{\tilde{R}(3)}$

Fig. 5. Parity check matrix (above) and generating matrix (below) of $C_{\tilde{R}^*(3)}$

4 Conclusion

We described a new family of curves generalizing the norm-trace curves introduced by Geil. We showed how the associated correction-capability-optimized codes behave well in the sense that the set of defining check monomials is divisor-closed. This enables us to exactly determine the dimension of the codes and to construct a parity check matrix and a generating matrix.

References

1. Bras-Amorós, M., O'Sullivan, M.E.: Duality for Some Families of Correction Capability Optimized Evaluation Codes (2007)
2. Geil, O.: On Codes From Norm-Trace Curves. Finite Fields Appl. 9(3), 351–371 (2003)
3. Koetter, R.: On the Determination of Error Values for Codes From a Class of Maximal Curves. In: Proc. 35-th Allerton Conference on Communication, Control, and Computing, pp. 44–53 (1997)
4. Lee, K., O'Sullivan, M.E.: List Decoding of Hermitian Codes Using Groebner Bases (2006)
5. Hoeholdt, T., van Lint, J.H., Pellikaan, R.: Algebraic Geometry Codes. In: Handbook of Coding Theory, vol. I, pp. 871–961. North-Holland, Amsterdam (1998)
6. O'Sullivan, M.E.: New Codes for the Berlekamp-Massey-Sakata Algorithm. Finite Fields Appl. 7(2), 293–317 (2001)
7. Geil, O., Pellikaan, R.: On the Structure of Order Domains. Finite Fields Appl. 8(3), 369–396 (2002)
8. Geil, O.: Codes Based on an F_q-Algebra. PhD thesis, Aalborg University (1999)
9. Little, J.B.: The Ubiquity of Order Domains for the Construction Of Error Control Codes. Adv. Math. Commun. 1(1), 151–171 (2007)
10. Sakata, S.: Extension of Berlekamp-Massey Algorithm to n Dimensions. IEEE Trans. Inform. Theory 34(5), 1332–1340 (1988)
11. Feng, G.L., Rao, T.R.N.: Improved Geometric Goppa codes. I. Basic Theory. IEEE Trans. Inform. Theory 41(6, part 1), 1678–1693 (1995)
12. Duursma, I.M.: Majority Coset Decoding. IEEE Trans. Inform. Theory 39(3), 1067–1070 (1993)
13. Bras-Amorós, M., O'Sullivan, M.E.: The Correction Capability of the Berlekamp-Massey-Sakata Algorithm With Majority Voting. Appl. Algebra Engrg. Comm. Comput. 17(5), 315–335 (2006)
14. Geil, O., Hoeholdt, T.: Footprints or Generalized Bezout's Theorem. IEEE Trans. Inform. Theory 46(2), 635–641 (2000)
15. Lidl, R., Niederreiter, H.: Introduction to Finite Fields and Their Applications, 1st edn. Cambridge University Press, Cambridge (1994)
16. Bourbaki, N.: Commutative Algebra, ch. 1–7. Elements of Mathematics. Springer, Berlin (1998)

On Generalized Hamming Weights and the Covering Radius of Linear Codes

H. Janwa[1] and A.K. Lal[2]

[1] Department of Mathematics and Computer Science, University of Puerto Rico (UPR), Rio Piedras Campus, P.O. Box: 23355, San Juan, PR 00931 - 3355
hjanwa@uprrp.edu

[2] Department of Mathematics and Statistics, Indian Institute of Technology Kanpur, 208016, INDIA
arlal@iitk.ac.in

Abstract. We prove an upper bound on the covering radius of linear codes over $I\!F_q$ in terms of their generalized Hamming weights. We show that this bound is strengthened if we know that the codes satisfy the chain condition or a partial chain condition. We show that this bound improves all prior bounds. Necessary conditions for equality are also presented.

Several applications of our bound are presented. We give tables of improved bounds on the covering radius of many cyclic codes using their generalized Hamming weights. We show that most cyclic codes of length ≤ 39 satisfy the chain condition or partial chain condition up to level 5. We use these results to derive tighter bounds on the covering radius of cyclic codes.

Keywords: Generalized Hamming weights, covering radius, Griesmer bound, optimal codes, cyclic codes, chain condition, generalized Griesmer bound.

1 Introduction

Let C be an $[n, k, d]$ code over $I\!F_q$ (*i.e.,* a linear subspace of $I\!F_q^n$ of dimension k and Hamming distance d) with a check matrix H, and let $r = n - k$ be the redundancy of C (for terminology and standard results on coding theory, we refer to MacWilliaims and Sloane [22]).

The covering radius $\mathcal{R}(C)$ of C is defined by $\mathcal{R}(C) := \max_{x \in I\!F_q^n} \min_{c \in C} d(x, c)$, where $d(\cdot, \cdot)$ is the Hamming distance. For more details on the covering radius of codes and its applications, we refer to the book by Cohen *et. al.* [2]. An important open problem is to determine the covering radii of cyclic codes, as this class contains BCH codes, Reed-Solomon codes, extended Goppa codes (in general some important AG codes), quadratic-residue codes, some extended algebraic geometric codes, finite geometric codes, and punctured Reed-Muller codes. Covering radii of cyclic codes of length ≤ 64 and co-dimension ≤ 28 were determined by Dougherty and Janwa [3] using a highly efficient parallel algorithm implemented

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 347–356, 2007.
© Springer-Verlag Berlin Heidelberg 2007

on massively parallel computers (such as, 1024 node hypercube (at Caltech), Connection Machines at UCLA, and the Los Alamos National Laboratories). In [1,2], the authors have shown that the problem of computing covering radii of codes is known to be both NP-hard and Co-NP-hard. Indeed, this problem is strictly harder than any NP-Complete problem, unless NP = co-NP.

The complexity of computing covering radius of an $[n, k, d]$ linear code is of the order of $\mathcal{O}(2^{n-k})$. Thus, finding exact covering radius of high co-dimensional linear codes is very difficult, and finding good upper bounds on the covering radii of such codes is an important problem. In this article, we give tight upper bounds on the covering radii of q-ary linear codes in terms of their generalized Hamming weights.

Generalized Hamming weights (GHWs) were introduced by Wei [26] to study the linear coding scheme for the wire-tap channel of Type II. Ozarow and Wyner [24] had introduced a linear coding scheme on this channel in connection with Cryptography (wire-taping). Wei [26] has shown that the GHWs completely characterize the performance of a linear code when it is used on the above channel. The GHWs are also called the dimension/length profile and G.D. Forney [5] has used it in determining the trellis complexity of linear block codes. A connection between GHWs and list decoding was found by Guruswami [6]. Since covering radius gives the limits of complete decoding, the results in [6] give further evidence of connection between covering radii and GHWS.

Let $\mathcal{C}$ be $[n, k]$ linear code over $I\!F_q$. For $1 \leq r \leq k$, the rth *generalized Hamming weight* of $\mathcal{C}$, denoted $d_r(\mathcal{C})$, was defined by Wei [26] as

$$d_r(\mathcal{C}) = \min \{|\text{Supp}(U)| : \ U \subset \mathcal{C} \text{ and } \ \dim(U) = r \ \} \tag{1}$$

where $\text{Supp}(U) = \bigcup_{x \in U} \text{Supp}(x)$ and $\text{Supp}(x)$ is the support of the vector x, *i.e.,* the set of coordinates where x is not zero. Note that the minimum distance d of $\mathcal{C}$ is precisely $d_1(\mathcal{C})$. From now on we will use d_r in place of $d_r(\mathcal{C})$, and d in place of d_1.

In general, it is very difficult to compute the GHWs of arbitrary linear codes. An efficient algorithm to compute the GHWs of cyclic codes was given by Janwa and Lal [17]. That algorithm is efficient if the dimension of the code is small (and hence the co-dimension is large). Thus, we are able to give tight upper bound on the covering radii of high co-dimensional cyclic codes about which we do not have much information.

The paper is arranged as follows: general background and a list of known results are contained in Section 2. The main result is contained in Section III. Section IV briefly discusses some improvements of the bounds. Some applications of our results are contained in Section V and Table I.

2 Background

2.1 Preliminaries

In this section, we mention a few known results with their references. These results will be used in later sections.

Fact 1. *[26] Let $\mathcal{C}$ be an $[n,k]$ linear code over $\mathbb{F}_q$ and let $C^{\perp}$ be the dual code of $\mathcal{C}$. Then $\{d_r(\mathcal{C}) : 1 \le r \le k\} = \{1,2,\ldots,n\} \setminus \{n+1-d_r(\mathcal{C}^{\perp}) : 1 \le r \le n-k\}$.*

We now define the term "chain condition", which was introduced by Wei and Yang [27]. We also state a few results related to the chain condition. For more results on codes satisfying the chain condition, we refer the reader to [4,8,21].

Definition 1. Let $\mathcal{C}$ be an $[n,\ k]$ linear code with $\{d_1(\mathcal{C}), d_2(\mathcal{C}), \ldots, d_k(\mathcal{C})\}$ as the GHWs of $\mathcal{C}$. Suppose the code $\mathcal{C}$ has k linearly independent vectors $X_1, X_2, \ldots, X_k$ over $\mathbb{F}_q$, satisfying $d_r(\mathcal{C}) = |\bigcup_{i=1}^{r} Supp\ X_i|$ for $1 \le r \le k$. Then the code $\mathcal{C}$ is said to satisfy the CHAIN CONDITION.

Fact 2. *[27] If a linear code $\mathcal{C}$ satisfies the chain condition then so does its dual code, $\mathcal{C}^{\perp}$.*

Fact 3. *[27] Let $\mathcal{C}$ be an $[n,\ k]$ linear code over $\mathbb{F}_q$ satisfying the chain condition. Suppose the vectors $X_1, X_2, \ldots, X_k$ of $\mathcal{C}$ are linearly independent over $\mathbb{F}_q$ and $d_r(\mathcal{C}) = |\bigcup_{i=1}^{r} Supp\ X_i|$ for $1 \le r \le k$. Then, there exists a generator matrix G of $\mathcal{C}$ having X_i for $1 \le i \le k$ as its ith row.*

We now mention two results on the covering radius of codes.

Fact 4. *[14] The $[n,1,n]$ code over $\mathbb{F}_q$ has covering radius $\lfloor \frac{n(q-1)}{q} \rfloor$.*

Theorem 1. *[12,13] Let C be an $[n,k,d]$ linear code over $\mathbb{F}_q$. Then $R \le n - \sum_{i=1}^{k} \lceil \frac{d}{q^i} \rceil$.*

For $1 \le r \le k$, we define $\mathcal{H}_q(n,r,d) := n - \sum_{i=1}^{r} \lceil \frac{d}{q^i} \rceil$ and $g_q(r,d) := \sum_{i=1}^{r} \lceil \frac{d}{q^i} \rceil$. Also, for fixed positive integers k and d, let $n_q(k,d)$ denote the smallest possible length of any linear $[n,k,d]$ code over $\mathbb{F}_q$. Then in Theorem 1, the bound $\mathcal{H}_q(n,k,d)$ can be re-written either as

$$n - g_q(k,d) + d - \lceil \frac{d}{q^k} \rceil, \quad \text{or} \quad n - g_q(k+1,d) + d \quad \text{or} \quad n - g_q(k, \lceil \frac{d}{q} \rceil). \tag{2}$$

The results that give conditions under which $g_q(.,.)$ in (2) can be replaced by the function $n_q(.,.)$ are given in [12] and [20]. Indeed, $\mathcal{H}_q(n,k,d)$ can be replaced by $n - g_q(k, \lceil \frac{d}{q} \rceil)$ (see [20]) for complete proofs).

3 Upper Bounds on the Covering Radius in Terms of GHWs

In this section, we find upper bounds on the covering radius of linear codes in terms of their GHWs.

Let the generator matrix G of the code $\mathcal{C}$ be partitioned into

$$G = \begin{bmatrix} G_1 & 0 \\ A & G_2 \end{bmatrix}. \tag{3}$$

With this notation, we state the next two lemmas. The proof of the first lemma is immediate from the definition of covering radius and we give the proof of the second for the sake of completeness.

Proposition 1. *[14,23] Let $\mathcal{C}$ be a linear code with generator matrix as given in (3). If, for $i = 1, 2$, the matrix G_i generates the code $\mathcal{C}_i$, then*

$$\mathcal{R}(\mathcal{C}) \leq \mathcal{R}(\mathcal{C}_1) + \mathcal{R}(\mathcal{C}_2). \tag{4}$$

Lemma 1. *[18] Let $\mathcal{C}$ be a linear code with generator matrix as given in (3). Suppose that for $i = 1, 2$, the matrix G_i generates the code $\mathcal{C}_i$, rank $(G_1) = r$ and $|Supp(G_1)| = d_r(\mathcal{C})$. Then the minimum distance of the code $\mathcal{C}_2$, denoted $d(\mathcal{C}_2)$, satisfies*

$$d(\mathcal{C}_2) \geq d_{r+1} - d_r. \tag{5}$$

Furthermore, if the code $\mathcal{C}$ satisfies the chain condition then equality is attained in (5).

Proof. Let $\mathbf{x} \in \mathcal{C}_2$ be the code of minimum weight. Then by definition of GHWs $d_{r+1}(\mathcal{C}) \leq |\operatorname{Supp}(G_1) \cup \operatorname{Supp}(\mathbf{x})| = d_r(\mathcal{C}) + d(\mathcal{C}_2)$. Hence, $d(\mathcal{C}_2) \geq d_{r+1} - d_r$. Furthermore, if $\mathcal{C}$ satisfies the chain condition, then by Fact 3, for the new generator matrix G, $d_{r+1}(\mathcal{C}) = d_r(\mathcal{C}) + d(\mathcal{C}_2)$. Thus, the result follows. □

Let $\mathcal{C}$ be an $[n, k, d]$ code over $\mathbb{F}_q$ with its weight hierarchy $\{d_1, d_2, \ldots, d_k\}$. For $1 \leq r \leq k$, Helleseth et. al [11] defined the excess sequence $\{\epsilon_1, \epsilon_2, \ldots, \epsilon_k\}$ and the $\delta-$sequence $\{\delta_1, \delta_2, \ldots, \delta_k\}$ of $\mathcal{C}$, respectively, by

$$\epsilon_r := d_r - g_q(r, d) \quad \text{and} \quad \delta_r := \lceil \frac{d}{q^{r-1}} \rceil, \tag{6}$$

where $d = d_1(\mathcal{C})$. Using the observation, $d_r - d_{r-1} \geq g_q(r, d) - g_q(r-1, d)$, they proved that $\epsilon_r \geq \epsilon_{r-1} \geq 0$, for $2 \leq r \leq k$. For convention, let $\epsilon_0 = 0 = \delta_0 = d_0$. Then, we observe that

$$d_r - d_{r-1} = \epsilon_r + g_q(r, d) - (\epsilon_{r-1} + g_q(r-1, d)) = \epsilon_r - \epsilon_{r-1} + \delta_r. \tag{7}$$

The next result gives a bound on the covering radius of codes in terms of the excess and the δ sequence. A preliminary proof of this result appeared in [18].

Theorem 2. *Let $\mathcal{C}$ be an $[n, k, d]$ code over $\mathbb{F}_q$ satisfying the chain condition. Then with the convention $\epsilon_0 = \delta_0 = d_0 = 0$, we have*

$$\mathcal{R}(\mathcal{C}) \leq \sum_{r=1}^{k} \lfloor \frac{(d_r - d_{r-1})(q-1)}{q} \rfloor = n - \sum_{r=1}^{k} \lceil \frac{d_r - d_{r-1}}{q} \rceil \tag{8}$$

$$= n - \sum_{r=1}^{k} \lceil \frac{(\epsilon_r - \epsilon_{r-1}) + \delta_r}{q} \rceil \quad \textit{follows from} \quad (7). \tag{9}$$

Proof. Without loss of generality, suppose $\mathcal{C}$ does not have a zero coordinate. We use induction on the dimension of the subcode of $\mathcal{C}$. For $r = 1$, the result follows from Fact 4. Let the theorem be true for all subcodes D_r with $\dim(D_r) = r$ for $1 \leq r \leq k-1$. Consider the subcode D_{r+1}. Since the code $\mathcal{C}$ satisfies the chain condition, by Fact 3, the generator matrix of D_{r+1} can be partitioned as in (3), in such a way that $\dim(\mathcal{C}_1) = r, |\mathrm{Supp}(D_r)| = d_r$ and $\mathcal{C}_2$ is a linear code with parameters $[d_{r+1} - d_r, 1, \ldots]$. So, by the induction hypothesis and Proposition 1

$$\mathcal{R}(\mathcal{C}) \leq \mathcal{R}(\mathcal{C}_1) + \mathcal{R}(\mathcal{C}_2) \leq \sum_{i=1}^{r} \lfloor \frac{(d_i - d_{i-1})(q-1)}{q} \rfloor + \mathcal{R}(\mathcal{C}_2).$$

As the code satisfies the chain condition, using (5) and Fact 4, the result follows. □

As an immediate corollary, we show that the bound $n - \sum_{r=1}^{k} \lceil \frac{(\epsilon_r - \epsilon_{r-1}) + \delta_r}{q} \rceil$ on the covering radius is better than the bound on the covering radius given by Theorem 1. We also denote this new bound by $\mathcal{CH}_q(n,k,d_1,d_2,\ldots,d_k)$.

Corollary 1. *Let $\mathcal{C}$ be an $[n,k,d]$ code over $\mathbb{F}_q$ satisfying the chain condition. Then*

$$\mathcal{R}(\mathcal{C}) \leq \mathcal{CH}_q(n,k,d_1,d_2,\ldots,d_k) \leq \mathcal{H}_q(n,k,d). \tag{10}$$

Furthermore, $\mathcal{CH}_q(n,k,d_1,d_2,\ldots,d_k) = \mathcal{H}_q(n,k,d)$ only if

$$\lceil \frac{(\epsilon_r - \epsilon_{r-1}) + \delta_r}{q} \rceil = \lceil \frac{\delta_r}{q} \rceil, \qquad \forall\ r,\ 1 \leq r \leq k, \tag{11}$$

i.e., only if $\epsilon_r \leq \epsilon_{r-1} + (q-1)$, for all $r, 1 \leq r \leq k$. In particular, for $q = 2$, the necessary conditions for equality are: for $1 \leq r \leq k$, $\epsilon_r \leq \epsilon_{r-1} + 1$.

Proof. From Theorem 2 and (6), we get

$$\begin{aligned}\mathcal{R}(\mathcal{C}) &\leq n - \sum_{r=1}^{k} \lceil \frac{(\epsilon_r - \epsilon_{r-1}) + \delta_r}{q} \rceil \leq n - \sum_{r=1}^{k} \lceil \frac{\delta_r}{q} \rceil = n - \sum_{r=1}^{k} \lceil \frac{1}{q} \lceil \frac{d}{q^{r-1}} \rceil \rceil \\ &= n - \sum_{r=1}^{k} \lceil \frac{d}{q^r} \rceil.\end{aligned}$$

Hence, if $\mathcal{CH}_q\ (n,\ k,\ d_1,\ d_2,\ \cdots, d_k) = \mathcal{H}_q(n,k,d)$ then using (7), we have $\sum_{i=1}^{k} \lceil \frac{d}{q^i} \rceil = \sum_{r=1}^{k} \lceil \frac{(\epsilon_r - \epsilon_{r-1}) + \delta_r}{q} \rceil \leq \sum_{r=1}^{k} \lceil \frac{\delta_r}{q} \rceil = \sum_{i=1}^{k} \lceil \frac{d}{q^i} \rceil$. Therefore, we get the required results. □

The next two theorems are similar to Theorem 2. To prove them, we first need to partition the generator matrix G of the code $\mathcal{C}$ as in (3) and then proceed on the lines of Theorem 2. Thus, the proof is omitted.

Theorem 3. *Let $\mathcal{C}$ be an $[n,k,d]$ code over $\mathbb{F}_q$ with GHWs $\{d_1,d_2,\ldots,d_k\}$. Then,*

$$\mathcal{R}(\mathcal{C}) \leq n - \max_{1\leq r\leq k-1}\{\mathcal{H}_q(d_r,r,d_1)+\mathcal{H}_q(n-d_r,k-r,d_{r+1}-d_r)\}. \tag{12}$$

Theorem 4. *Let $\mathcal{C}$ be an $[n,k,d]$ code over $\mathbb{F}_q$ with GHWs $\{d_1,d_2,\ldots,d_k\}$. Suppose we know that the code $\mathcal{C}$ partially satisfies the chain condition. That is, for some l, $1\leq l\leq k$ we know $D_1\subset D_2\subset\cdots\subset D_l\subset\mathcal{C}$ with $|Supp\ (D_i)|=d_i$. Then*

$$\mathcal{R}(\mathcal{C}) \leq n - \sum_{i=1}^{l}\lceil\frac{d_i-d_{i-1}}{q}\rceil - \mathcal{H}_q(n-d_l,k-l,d_{l+1}-d_l). \tag{13}$$

In Table 1, $\mathcal{GH}_2(n,k,d_1,d_2,\ldots,d_k)$ denotes the expression on the right hand side of (12) and $\mathcal{PCH}_2(n,k,d_1,d_2,\ldots,d_k)$ denotes the expression on the right hand side of (13).

4 Further Improvements

Let $U_q(n,k,d)$ (respectively, $U_q(n,k,d_1,d_2,\ldots,d_k)$) denote the best known upper bound on the covering radius R of an arbitrary $[n,k,d]$ linear code over $\mathbb{F}_q$ (respectively, with GHWs $\{d_1,d_2,\ldots,d_k\}$). Then $U_q(n,k,d)\leq\min\{\ \mathcal{H}_q(n,k,d),\ n-n_q(k,\lceil\frac{d}{q}\rceil)\ \}$ and $U_q(n,k,d_1,d_2,\ldots,d_k)\leq\min\{\ \mathcal{GH}_q(n,k,d_1,d_2,\ldots,d_k),\ U_q(n,k,d)\ \}$. Also, if we know that $\mathcal{R}\leq d$, then $U_q(n,k,d)\leq\min\{\mathcal{H}_q(n,k,d),n-n_q(k+1,d)+d\}$. Therefore, from Theorem 3, we have the following result.

Theorem 5

$$\mathcal{R}\leq U_q(d_r,r,d_1,d_2,\ldots,d_r)+U_q(n-d_r,k-r,d_{r+1}-d_r).$$

Furthermore, if the code satisfies the chain condition, then the $\mathcal{GH}_q(\cdot)$ function can be replaced by the $\mathcal{CH}_q(\cdot)$ function.

5 Some Applications

5.1 Existence of Chains

Remark 1. Let $\mathcal{C}$ be an $[n,k]$ code with GHWs $\{d_1,d_2,\ldots,d_k\}$. Suppose that $n-\sum_{i=1}^{k}\lceil\frac{(d_i-d_{i-1})(q-1)}{q}\rceil<\mathcal{R}(\mathcal{C})$. Then the code $\mathcal{C}$ doesn't satisfy the chain condition as it contradicts Theorem 2.

For example, consider the code $\mathcal{C}$ generated by the matrix

$$G=\left[\begin{array}{c|c|c} 1 & 0\ 0\ 0\ 1 & 0\ 0\ 0\ 1 \\ 0 & 1\ 0\ 1\ 0 & 1\ 0\ 1\ 0 \\ 0 & 0\ 1\ 1\ 0 & 0\ 1\ 1\ 0 \end{array}\right]$$

which (after permuting the columns) can be written as

$$G_1 = \begin{bmatrix} 1\,1\,1\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,1\,1\,1\,1\,0\,0 \\ 0\,0\,0\,0\,0\,1\,1\,1\,1 \end{bmatrix}.$$

Using the matrix G_1 one easily observes that the GHWs of $\mathcal{C}$ are $\{3, 6, 9\}$. The code doesn't satisfy the chain condition as there does not exist vectors X_1, X_2 such that $|\mathrm{supp}(X_1)| = 3$ and $|\mathrm{supp}\,(\langle X_1, X_2\rangle)\,| = 6$. Observe that the matrix G is of the form $[C_1|C|C]$. Hence, from (3), $\mathcal{R}(\mathcal{C}) = 4$. Now bounding the covering radius with Theorem 2, we get $\mathcal{R}(\mathcal{C}) \leq 9 - \sum_{i=1}^{3} \lceil \frac{d_i - d_{i-1}}{2} \rceil = 9 - 6 = 3$ which contradicts the actual value of the covering radius. Therefore, the code above doesn't satisfy the chain condition.

We have seen that if $\mathcal{CH}_q(n, k, d_1, d_2, \ldots, d_k) = \mathcal{H}_q(n, k, d)$ then $\lceil \frac{(\epsilon_r - \epsilon_{r-1}) + \delta_r}{q} \rceil = \lceil \frac{\delta_r}{q} \rceil, \forall\ r,\ 1 \leq r \leq k$. Hence we have the following lemma.

Lemma 2. *Suppose that* $\mathcal{CH}_q(n, k, d_1, d_2, \ldots, d_k) = \mathcal{H}_q(n, k, d)$. *Then for each* $r,\ 1 \leq r \leq k$, *the* ϵ*-sequence and the* δ*-sequence satisfy the following condition: if* $\delta_r \equiv t \pmod{q}$ *for* $0 \leq t < q$ *then* $\epsilon_r - \epsilon_{r-1} \leq q - t$.

Therefore, $\mathcal{CH}_q(n, k, d_1, d_2, \ldots, d_k) \neq \mathcal{H}_q(n, k, d)$ whenever the above condition is violated for some $r,\ 1 \leq r \leq k$. For example, consider the Reed-Muller codes $R(u, m)$. It was shown (see [27]) that for all u and m, the codes $R(u, m)$ satisfy the chain condition. We also know (see [26]) that $d_r(R(1, m)) = 2^{m-1} + 2^{m-2} + \cdots + 2^{m-r}$ for $1 \leq r \leq m$, and $d_{m+1}(R(1, m)) = 2^m$. Therefore $\mathcal{R}(R(1, m)) \leq 2^{m-1} - 1$. For this code, it can be observed that

$$\lceil \frac{d_r - d_{r-1}}{2} \rceil = \lceil \frac{\delta_r}{2} \rceil, \qquad \text{for all } \ r,\ 1 \leq r \leq k,$$

and thus $\mathcal{H}_2(n, k, d) = \mathcal{CH}_2(n, k, d_1, d_2, \ldots, d_k)$. But for $q > 2$, $\mathcal{CH}_q(\ n, k,\ d_1,\ d_2,\ \ldots,\ d_k) < \mathcal{H}_q(n, k, d)$ (using results from [7]).

5.2 GHWs and the Covering Radius of Cyclic Codes

Wei's original paper [26] on the topic led to a tremendous interest in GHWs. In [17], we have given an efficient algorithm for computing the Weight hierarchy of cyclic codes. We use the results on the GHWs of cyclic codes from [17] to derive tight upper bounds on the covering radii of cyclic codes of odd lengths ≤ 39 for which the function $\mathcal{H}_2(\cdot)$, is strictly greater than the function, $\mathcal{GH}_2(\cdot)$. We obtain good bounds because most of the cyclic codes in our list satisfy the chain or the partial chain condition. The results are given in Table I. We compare our results with other known bounds listed in Section II.

The covering radius of cyclic codes of lengths ≤ 64 and co-dimension ≤ 28 were computed in [3]. We use the tables given in [3] to show that the bounds proved here come very close to equality for many cyclic codes. In fact, for 21 of the 85 cyclic codes listed in Table I, our bound attains equality.

Table 1. Comparisons of Various Bounds

n	Sl.no.	k	d_1	$\lfloor \frac{n}{2} \rfloor$	$\mathcal{H}_2(\cdot)$	$n - n_2(k, \lceil \frac{d}{2} \rceil)$	$\mathcal{GH}_2(\cdot)$	$\mathcal{CH}_2(\cdot)$	$\mathcal{R}$
9	1	3	3	4	5	5	4	3 =	3
15	3	9	3	7	5	5	4	4	3
15	5	8	4	7	6	6	5	5	4
15	8	7	3	7	7	7	6	5	3
15	12	5	3	7	9	9	8	5 =	5
15	14	4	6	7	8	8	7	6 =	6
15	15	3	5	7	9	9	7	6 =	6
21	5	13	3	10	7	7	6	6 **P**	3
21	7	12	4	10	8	8	7	7 **P**	4
21	10	12	3	10	8	8	7	6	3
21	12	11	4	10	9	9	8	7	4
21	13	10	5	10	8	7	7	7 **P**	6
21	14	10	4	10	10	10	9	8	4
21	16	9	4	10	11	11	10	9 =	9
21	17	9	6	10	9	8	8	8 **P**	5
21	18	9	3	10	11	11	10	7	5
21	19	8	6	10	10	9	9	9 **P**	7
21	20	8	6	10	10	9	9	8	6
21	21	7	8	10	10	9	9	9	6
21	22	7	3	10	13	13	12	7 =	7
21	23	6	8	10	11	10	10	10	9
21	24	6	6	10	12	11	11	8 =	8
21	25	6	7	10	11	10	9	9	6
21	27	4	9	10	10	10	9	9	8
21	29	3	7	10	14	14	11	9 =	9
27	1	9	3	13	17	17	16	9 =	9
27	2	8	6	13	16	15	15	10 =	10
27	3	7	6	13	17	16	14	11	10
27	4	6	6	13	18	17	15	12 =	12
27	5	3	9	13	17	17	14	12 =	12
31	9	16	7	15	11	9	10	10 **P**	5
31	10	16	6	15	12	10	11	10	5
31	11	16	5	15	12	10	11	10	5
31	13	15	8	15	12	10	11	11 **P**	7
31	14	15	6	15	13	11	12	11	9
31	15	15	8	15	12	10	11	11	6
31	19	11	10	15	13	11	12	12 **P**	8
31	22	10	10	15	14	12	13	13 **P**	11
33	1	23	3	16	9	9	8	7 **P**	3
33	3	21	3	16	11	11	10	8	5
33	5	20	6	16	10	8	9	9 **P**	6
33	7	13	10	16	13	11	12	12 **P**	8
33	8	13	3	16	19	19	18	11	8
33	9	12	10	16	14	12	13	13	11
33	10	12	6	16	18	16	17	12	9
33	11	11	11	16	14	12	13	13 **P**	10
33	12	11	3	16	21	21	20	11 =	11
33	13	10	12	16	15	13	14	14 **P**	11
33	14	10	6	16	20	19	19	12 =	12
33	15	3	11	16	22	22	18	15 =	≥ 15
35	6	22	4	17	12	12	11	10	5
35	7	20	6	17	12	10	11	11 **P**	5
35	9	19	6	17	13	11	12	12 **P**	7
35	10	19	4	17	15	15	14	11	6
35	11	19	4	17	15	15	14	13	6
35	12	18	4	17	16	17	15	13	8
35	13	17	6	17	15	13	14	11	7
35	14	16	6	17	16	14	15	12 **P**	8
35	15	16	7	17	15	13	14	13 **P**	7
35	16	16	4	17	18	18	17	14	7
35	17	15	8	17	16	141	15	14 **P**	9
35	18	15	4	17	19	19	18	15 =	15
35	19	13	8	17	18	16	17	15	9
35	20	12	8	17	19	17	18	16	15
35	21	11	5	17	21	20	18	14	10
35	22	10	10	17	18	16	17	14	11
35	23	8	7	17	23	22	19	14	12
35	24	7	14	17	18	17	16	16	13
35	25	7	5	17	25	24	22	14 =	14
35	26	6	10	17	22	19	19	15	≥ 11
35	27	5	7	17	26	25	22	15 =	≥ 15
35	28	4	14	17	21	19	18	16	≥ 15
35	29	4	15	17	20	20	17	16	≥ 15
39	1	27	3	19	11	11	10	8	4
39	3	25	3	19	13	13	12	10	5
39	5	24	6	19	12	10	11	11	6
39	7	15	10	19	17	14	16	16	10
39	8	15	3	19	23	23	22	13	9
39	9	14	10	19	18	15	17	17	13
39	10	14	6	19	22	20	21	14	10
39	11	13	12	19	18	16	17	17	11
39	12	13	3	19	25	25	24	13 =	13
39	13	12	12	19	19	17	18	17	15
39	14	12	6	19	24	22	23	14 =	14
39	15	3	13	19	25	26	20	18 =	18

The following abbreviations have been used:

- Sl.no. := if Sl.no.= l, then this code is the lth cyclic code of length n listed in [17].
- **P** := the particular code satisfies the partial chain condition up to certain level. In this case we use the bound $\mathcal{PCH}_2(\cdot)$.
- ' $=$' the bound equals the actual value of the covering radius.
- $\mathcal{H}_2(\cdot)$:= $\mathcal{H}_2(n, k, d)$, Theorem 1.
- $\mathcal{GH}_2(\cdot)$:= $\mathcal{H}_2(n, k, d_1, d_2, \ldots, d_k)$, Theorem 3.
- $\mathcal{CH}_2(\cdot)$:= bound derived using chain or partial chain, Theorem 2.
- $\mathcal{R}$:= Actual value of covering radius computed in [3].

The bounds are computed from the values of GHWs of cyclic codes given in [17] and the information regarding the chain condition.

For several examples of cyclic codes, our bounds improve other bounds that have appeared in literature, for example, those that depend upon the dual distance, and those that use powerful results from algebraic geometry. As an example, the GHWs of the duals of the three error-correcting BCH code of length 31 was determined in [17], and it turns out that it satisfies the partial chain condition. So, for this example, our bound on the covering radius is $R(BCH^{\perp}(3,5)) = 11$ (in Table I, this is the [31,15,8] code). This improves the bound of 12 (for $e = 3, m = 5$) obtained by the following theorem of Tietäväinen (see [2]). This bound was obtained by the use of powerful methods from algebraic geometry:

Theorem 6. *Let $BCH(e,m)$ be an e-error correcting BCH code of length $n = 2^m - 1$. Then*

$$R(BCH^{\perp}(e,m)) \leq 2^{m-1} - 1 - (\sqrt{e} - e^{1/e})\sqrt{2^m - e - 2}.$$

Acknowledgment

The authors thank Prof. H.F. Mattson, Jr. for helpful comments on the paper. The first author would like to thank Prof. T. Høholdt for enlightening discussions.

References

1. Berlekamp, E.R., McEliece, R.J., van Tilborg, H.C.A.: On the Inherent Intractability of Some Coding Problems. IEEE Trans. Inform. Theory 24(3), 384–386 (1996)
2. Cohen, G., Honkala, I., Litsyn, S., Lobstein, A.: Covering Codes. In: Sakata, S. (ed.) AAECC-8. LNCS, vol. 508, pp. 173–239. Springer, Heidelberg (1991)
3. Dougherty, R., Janwa, H.: Covering Radius Computations for Binary Cyclic Codes. Math. Comp. 57(195), 415–434 (1991)
4. Encheva, S., Kløve, T.: Codes Satisfying the Chain Condition. IEEE Trans. Inform. Theory 40(1), 175–180 (1994)
5. Forney, G.D.: Dimension/Length Profiles and Trellis Complexity of Linear Block Codes. IEEE Trans. Inform. Theory 40(6), 1741–1752 (1994)
6. Guruswami, V.: List Decoding From Erasures: Bounds and Code Constructions. IEEE Trans. Inform. Theory 49(11), 2826–2833 (2003)
7. Heijnen, P., Pellikaan, R.: Generalized Hamming Weights of q-ARY Reed-Muller Codes. IEEE Trans. Inform. Theory 44(1), 181–196 (1998)
8. Helleseth, T., Kløve, T., Ytrehus, Ø.: Codes, Weight Hierarchies, and Chains. In: 1992 ICCS/ISITA, Singapore, pp. 608–612 (1992)
9. Helleseth, T., Kløve, T., Ytrehus, Ø.: Generalized Hamming Weights of Linear Codes. IEEE Trans. Inform. Theory 38(3), 1133–1140 (1992)
10. Helleseth, T., Kløve, T., Levenshtein, V.I., Ytrehus, Ø.: Bounds on the Minimum Support Weights. IEEE Trans. Inform. Theory 41(2), 432–440 (1995)
11. Helleseth, T. , Kløve, T. , Levenshtein, V. I., Ytrehus, Ø.: Excess Sequences of Codes and the Chain Condition. In: Reports in Informatics, no. 65, Department of Informatics, University of Bergen (1993)

12. Janwa, H.: On the Optimality and Covering Radii of Some Algebraic Geometric Codes. In: Workshop on Coding Theory, IMA, University of Minnesota (1988)
13. Janwa, H.: Some New Upper Bounds on the Covering Radius of Binary Linear Codes. IEEE Trans. Inform. Theory 35, 110–122 (1989)
14. Janwa, H.: On the Covering Radii of q-ary Codes. In: 1990 ISIT, San Diego
15. Janwa, H.: Some Optimal Codes From Algebraic Geometry and Their Covering Radii. Europ. J. Combinatorics 11, 249–266 (1990)
16. Janwa, H.: On the Covering Radii of AG Codes (preprint, 2007)
17. Janwa, H., Lal, A.K.: On the Generalized Hamming Weights of Cyclic Codes. IEEE Trans. Inform. Theory 43(1), 299–308 (1997)
18. Janwa, H., Lal, A.K.: Bounds on the Covering Radii of Codes in Terms of Their Generalized Hamming Weights. MRI (preprint, 1997)
19. Janwa, H., Lal, A.K.: Upper Bounds on the Covering Radii of Some Important Classes of Codes Using Their Generalized Hamming Weights (preprint, 2007)
20. Janwa, H., Mattson Jr., H.F.: Some Upper Bounds on the Covering Radii of Linear Codes over F_q and Their Applications. Designs, Codes and Cryptography 18(1-3), 163–181 (1999)
21. Kløve, T.: Minimum Support Weights of Binary Codes. IEEE Trans. Inform. Theory 39(2), 648–654 (1993)
22. MacWilliaims, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes. North-Holland, Amsterdam (1977)
23. Mattson Jr., H.F.: An Improved Upper Bound on Covering Radius. In: Poli, A. (ed.) AAECC-2. LNCS, vol. 228, pp. 90–106. Springer, Heidelberg (1986)
24. Ozarow, L.H., Wyner, A.D.: Wire-Tap Channel-II. AT & T Bell Labs Tech J. 63, 2135–2157 (1984)
25. Pless, V.S., Huffman, W.C., Brualdi, R.A.: An Introduction to Algebraic Codes. In: Pless, V.S., Huffman, W.C. (eds.) Handbook of Coding Theory, pp. 3–139. Elsevier, Amsterdam (1998)
26. Wei, V.K.: Generalized Hamming Weights for Linear Codes. IEEE Trans. Inform. Theory 37(5), 1412–1418 (1991)
27. Wei, V.K., Yang, K.: The Feneralized Hamming Weights for Product Codes. IEEE Trans. Inform. Theory 39(5), 1709–1713 (1993)
28. Yang, K., Kumar, P.V., Stichtenoth, H.: On the Weight Hierarchy of Geometric Goppa Codes. IEEE Trans. Inform. Theory 40(3), 913–920 (1994)

Homomorphic Encryptions of Sums of Groups

Akihiro Yamamura

National Institute of Information and Communications Technology,
4-2-1, Nukui-Kitamachi, Koganei, Tokyo, 184-8795 Japan
aki@nict.go.jp

Abstract. We examine the mechanism of homomorphic encryptions based on the subgroup membership problem. Using the mechanism, we construct a homomorphic encryption of a direct sum of groups.

1 Introduction

A mapping between algebraic systems is called a *homomorphism* if it preserves the algebraic structures. In cryptography, a trapdoor one-way homomorphism between cyclic groups have been studied and applied to many cryptographic protocols. Such encryptions include ElGamal, Goldwasser-Micali, Paillier, Okamoto-Uchiyama cryptosystems and so on [2,4,5]. Homomorphic encryptions share many similarities, however, no uniform mechanism has been presented so far. In this paper, we study homomorphic encryptions from the standpoint of group theory, in particular, we use split exact sequences and the subgroup membership problem to explain the mechanism, constructions and the indistinguishability of homomorphic encryptions. We then construct a homomorphic encryption of a direct sum of groups. Algebraic structure is useful to encrypt structured data and relations among the data, and direct sums of cyclic groups possess richer structure than cyclic groups. Therefore, applications of homomorphic encryptions of direct sums of groups go beyond the encryptions between cyclic groups for some applications. For example, a general n-cryptographic counter can be constructed using a homomorphic encryption of a direct sum of n cyclic groups.

Our first contribution in this paper is to explain the mechanism of homomorphic encryptions using uniform design via exact sequences and the subgroup membership problem. This approach simplify the mechanism of numerous homomorphic encryptions and enable us to explain functionality of homomorphic encryptions in a mathematically sound way. Furthermore, the mechanism is wide enough to include encryptions whose set of plaintexts is a direct sum of groups. The second contribution is to construct a homomorphic encryption whose set of plaintexts is a direct sum of groups. The encryption satisfies IND-CPA provided the corresponding subgroup membership problem is intractable. We also define an operation among several encryption functions; it gives the way to create a new encryption function from the old ones in a way that the new one is closely related to the old ones.

S. Boztaş and H.F. Lu (Eds.): AAECC 2007, LNCS 4851, pp. 357–366, 2007.
© Springer-Verlag Berlin Heidelberg 2007

2 Mechanism of Homomorphic Encryptions

We describe the mechanism of homomorphic encryption functions. First, we recall that a sequence of homomorphisms $1 \longrightarrow H \xrightarrow{\delta} G \xrightarrow{d} P \longrightarrow 1$ is called *exact* if the kernel Ker$d$ coincides with the image Imδ. Following the mathematical convention, "1" stands for the trivial subgroup $\{1\}$. If the group operation is additive, we may denote it by 0. Note that $\delta : H \to G$ is an embedding and $G \to P$ is surjective. Furthermore, if there exists a homomorphism $\epsilon : P \to G$ such that $d \circ \epsilon$ is the identity mapping of P, then we say that the exact sequence *splits*. In such a case, G is isomorphic to a semidirect product of H by P.

Let k be the security parameter. For the input 1^k, a probabilistic polynomial time algorithm $\mathcal{IG}$, called an *instance generator*, outputs the description of a finite group P, the description of a finite group G, the description of a subgroup H of G, the couple of public and private keys, and the description of a probabilistic algorithm $\mathcal{SAM}$, called a sampling algorithm, that chooses randomly and uniformly an element of H. Elements in G and P are represented by binary strings and operations in the groups, multiplication and taking inverses, are efficiently computable. The subgroup H is called the *subgroup of randomizers*. The group P is called the *group of plaintexts*. The *encoder* ϵ is an isomorphism of P into G, and there is an algorithm to compute ϵ efficiently with the public key. The *decryption function* d is a homomorphism of G onto P such that $d \circ \epsilon = id_P$ and its kernel Ker d coincides with H. Furthermore, d is efficiently computable with the private key. In such a case, by the basic algebra, we have a split exact sequence $1 \longrightarrow H \xrightarrow{\delta} G \underset{\epsilon}{\overset{d}{\rightleftarrows}} P \longrightarrow 1$. Then $G = H\epsilon(P)$ and $H \cap \epsilon(P) = 1$. This implies that G is a semidirect product of H and $\epsilon(P)$. Furthermore, $G = \epsilon(P) \times H \cong P \times H$ and $P \cong G/H$ and $\epsilon(P)$ is the set of representatives of H in G, that is, $G = \epsilon(m_0)H \cup \epsilon(m_1)H \cup \cdots \cup \epsilon(m_n)H$, where $P = \{m_0, m_1, \ldots, m_n\}$ (if P is finite).

Encryption: The encryption function e is computed by

$$e(m) = \epsilon(m)r \ , \tag{1}$$

where r is an output of $\mathcal{SAM}$ and m is a plaintext in P. We note that each coset $\epsilon(m)H$ is the set of ciphertexts of the plaintext m. This means that e can be considered a probabilistic algorithm choosing an element randomly and uniformly from $\epsilon(m)H$ for each plaintext $m \in P$.

Decryption: The decryption is done just by computing d provided the private key (secret information) is given. Since Ker d coincides with H and $d \circ \epsilon = id_P$, we have $d(e(m)) = d(\epsilon(m)r) = d(\epsilon(m))d(r) = id_P(m) = m$ for every ciphertext $m \in P$. Hence, d decrypts the ciphertext $e(m)$. Note that we need the private key to compute d.

Assumption: Let G be a group, and let H be its subgroup. The membership problem is to decide whether or not a given element g in G belongs to H. A

computation problem is called *intractable* if no efficient algorithm exists. The efficiency is characterized by the asymptotic behavior of an algorithm with respect to the size of the input. For the input 1^k, where k is the security parameter, a probabilistic polynomial time algorithm $\mathcal{IG}$ outputs the description of a group G, the description of a subgroup H of G and the trapdoor that provides a polynomial time algorithm for the subgroup membership problem of H in G. The algorithm $\mathcal{IG}$ is called the *instance generator*. Every element of G is represented as a binary sequence of length k. Computation of the multiplication in G is performed in polynomial time in k. The predicate for the membership of a subgroup is denoted by Mem, that is, Mem is defined by $\mathrm{Mem}(G, H, x) = 1$ if $x \in H$ and 0 otherwise, where $\mathcal{IG}$ outputs the pair (G, H) for 1^k and x is in G. The *subgroup membership problem* is to compute Mem in polynomial time in k when we inputs 1^k and obtain a pair of groups (G, H) and an element g in G, which is uniformly and randomly chosen from H or $G \setminus H$ according to the coin toss $b \stackrel{R}{\leftarrow} \{0, 1\}$. If there does not exist a probabilistic polynomial time algorithm that computes Mem with a probability substantially larger than $\frac{1}{2}$, then we say that the membership problem is *intractable*. It is shown in [9] that the quadratic residue problem and the decision Diffie-Hellman problem can be characterized as a subgroup membership problem. We briefly review these two problems.

Quadratic Residue Problem: Let p, q be primes. Set $N = pq$. The primes p and q are trapdoor information for the quadratic residue problem, on the other hand, the integer N is a public information. Let G be the subgroup of $(\mathbb{Z}/(N))^*$ consisting of the elements whose Jacobi symbol is 1, and let H be the subgroup of G consisting of quadratic residues of G, that is, $H = \{x \in G \mid x = y^2 \bmod N \text{ for } y \in (\mathbb{Z}/(N))^*\}$. The *quadratic residue problem* (QR for short) of H in G is to decide whether or not, a given element $g \in G$, g belongs to H. We can effectively determine the membership of g in H provided that the information p and q are available. No polynomial time algorithm is known for the membership of a randomly chosen element of G in H without the information p and q. Hence, if we define an instance generator for the QR problem as a probabilistic algorithm that outputs two primes p and q of size k and a quadratic non-residue h whose Jacobi symbol is 1 for the input 1^k, then the QR problem is considered as a subgroup membership problem.

Decision Diffie-Hellman Problem: Let C be a cyclic group of prime order p. Let g be a generator of C. The *decision Diffie-Hellman problem* (DDH for short) is to decide whether or not $h_2 = g_2^a$ for the given quadruple (g_1, h_1, g_2, h_2) of elements in C with $h_1 = g_1^a$ for some $1 \leq a \leq p - 1$. If so, we say that (g_1, h_1, g_2, h_2) is a Diffie-Hellman quadruple. The integer a is the trapdoor of the DDH problem. Knowing the trapdoor a, we can efficiently decide whether or not $h_2 = g_2^a$. Now we set G to be the direct product $C \times C$. Then the input to the DDH problem is (x, y) where $x, y \in G$, that is, $x = (g_1, h_1)$ and $y = (g_2, h_2)$. It is obvious that (g_1, h_1, g_2, h_2) is a Diffie-Hellman quadruple if and only if y belongs to the subgroup $< x >$ of G generated by x. It follows

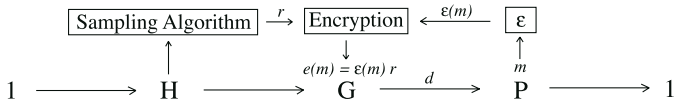


Fig. 1. Exact Sequence and Mechanism of Homomorphic Encryption

that the DDH problem for the cyclic group C is equivalent to the subgroup membership problem of the group $H =< x >$, where $x = (g_1, g_1^a)$, in the group $G = C \times C =< g_1 > \times < g_1 >$.

Homomorphic Property: For any ciphertexts $c_1 = \epsilon(m_1)r_1$ and $c_2 = \epsilon(m_2)r_2$, where r_1, r_2 are outputs of $\mathcal{SAM}$ and m_1, m_2 are plaintexts in P, we have $c_1c_2 = \epsilon(m_1)r_1\epsilon(m_2)r_2 = \epsilon(m_1m_2)r_1r_2$ since ϵ is a homomorphism. Note also that $r_1r_2 \in H$. Therefore, c_1c_2 belongs to $\epsilon(m_1m_2)H$ and it is a ciphertext of m_1m_2. Thus the encryption function e is homomorphic. In the language of group theory, the homomorphic property is a natural consequence of the quotient group G/H forms a group, that is, $c_1Hc_2H = c_1c_2H$ for all cosets c_1H, c_2H.

We summarize the mechanism of a homomorphic encryption in Fig. 1. The decryption d can be efficiently computed provided that the private key is given.

ElGamal Encryption: Let $C = \langle g \rangle$ be a cyclic group of prime order p. Let $P = C$ and $G = C \times C$. The encoder ϵ is defined to be the function $m \to (1, m) \in G$. It is clear that ϵ is an isomorphism of P into G. Suppose that the public key for the ElGamal encryption is (g, g^b), where b is uniformly and randomly chosen. Let $H = \langle (g, g^b) \rangle$ the subgroup of G generated by the element (g, g^b). We note that $\epsilon(P) \cap H = 1$ and $G = \epsilon(P)H$. Recall that a ciphertext of $m \in P$ is $e(m) = (g^a, g^{ab}m) = (1, m)(g, g^b)^a = \epsilon(m)r$, where $r = (g, g^b)^a$ is randomly and uniformly chosen from the subgroup H of randomizers, that is, a is randomly chosen, and $e(m)$ belongs to $\epsilon(m)H$. Since ϵ is an isomorphism, the encryption is homomorphic, that is, $e(m_1m_2) = e(m_1)e(m_2)$, or $\epsilon(m_1m_2)H = \epsilon(m_1)H\epsilon(m_2)H$. The decryption $d : G \to P$ is defined by $(g^x, g^y) \to g^{-xb}g^y$. Clearly d is a homomorphism. Moreover, it is easy to see that Ker$d$ is H and $d \circ \epsilon = id_P$. Hence, we have the split exact sequence $1 \longrightarrow \langle (g, g^b) \rangle \longrightarrow C \times C \overset{\epsilon}{\curvearrowleft} C \longrightarrow 1$. We recall that the semantic security of the ElGamal is equivalent to the DDH problem [7].

Goldwasser-Micali Encryption: Let G be the subgroup of $(\mathbb{Z}/(N))^*$, where $N = pq$, consisting of the elements whose Jacobi symbol is 1, and H be the subgroup of G consisting of quadratic residues of G. Goldwasser-Micali encryption [2] is characterized as follows. Let P be the cyclic group of order two, that is, $(\mathbb{Z}/2, +)$. The encoder $\epsilon : P \to G$ is defined by $m \to g^m$, where g is an element of $G \setminus H$ and the public key. The decryption $d : G \to P$ is defined by $d(x) = 0$ if $x \in H$ and $d(x) = 1$ otherwise. The message $m \in P$ is encrypted to be $e(m) = g^m r = \epsilon(m)r$, where r is uniformly and randomly chosen from H. Clearly d is a homomorphism. Moreover, evidently Ker$d$ is H and $d \circ \epsilon = id_P$.

Hence, we have the split exact sequence $1 \longrightarrow G^2 \longrightarrow G \overset{\epsilon}{\curvearrowright} (\mathbb{Z}/2, +) \longrightarrow 0$. We recall that the semantic security of the Goldwasser-Micali is equivalent to the quadratic residue problem [2].

The textbook RSA has the homomorphic property, that is, $e(m_1 m_2) = (m_1 m_2)^e = m_1^e m_2^e = e(m_1)e(m_2)$. In this case, the space of plaintexts does not form a group unless the user restricts the domain of the plaintexts to $(\mathbb{Z}/n)^*$. Instead, usually the domain of the plaintexts is just the semigroup $\mathbb{Z}/n$. Thus, the textbook RSA is not characterized as the scheme above.

3 Homomorphic Encryptions of Sums of Groups

In this section we introduce a homomorphic encryption whose group of plaintexts is a direct sum of (more than one) cyclic groups with distinct prime orders following the design of the encryption (1). The trivial method to construct encryptions of direct sums of groups is to concatenate several simple homomorphic encryptions based on cyclic groups. Ciphertexts of such encryptions can be easily tampered. For example, it is quite easy to exchange a part of the ciphertext by another ciphertext without the private key. Whereas it is hard to alter a ciphertext in the proposed scheme. In addition, we shall construct a new encryption $e_1 \circledcirc e_2$ from two encryptions e_1 and e_2 in Section 4. This property is desirable for some applications such as an electronic voting schemes.

We note that a finitely generated abelian group is a direct sum of finite number of cyclic groups. In particular, a finite abelian group is a direct sum of finite number of cyclic groups of finite order.

3.1 Okamoto-Uchiyama Logarithmic Function

Let us recall the logarithmic function L_p introduced by Okamoto and Uchiyama [4]. Suppose that p is a prime number of size k. Let Γ_p be the p-Sylow subgroup of the group $(\mathbb{Z}/(p^2))^*$ of units, that is, Γ_p is the maximal subgroup whose order is a power of p. The group $(\mathbb{Z}/(p^2))^*$ has order $\phi(p^2) = p(p-1)$. Thus $(\mathbb{Z}/(p^2))^*$ is an internal direct sum of Γ_p and the subgroup of order $p-1$. Since the mapping $x(\mathrm{mod}\ p^2) \mapsto x(\mathrm{mod}\ p)$ is a homomorphism of $(\mathbb{Z}/p^2)^*$ onto $(\mathbb{Z}/p)^*$, we have $(\mathbb{Z}/p^2)^*/\Gamma_p \cong (\mathbb{Z}/p)^*$. Therefore, the subgroup of order $p-1$ is isomorphic to $(\mathbb{Z}/(p))^*$ and so it is cyclic. On the other hand, Γ_p has order p and so it is cyclic. It follows that $(\mathbb{Z}/(p))^*$ is cyclic because p and $p-1$ are coprime.

We next show that if $x \equiv 1(\mathrm{mod}\ p)$, then we have $x^p \equiv 1(\mathrm{mod}\ p^2)$. Suppose $x \equiv 1(\mathrm{mod}\ p)$. Then $x = cp + 1$ for some c in $\mathbb{Z}$. We have $x^p = (cp+1)^p = \sum_{i=0}^{p} \binom{p}{i} (cp)^{p-i}$. Hence, $x^p = dp^2 + 1$ for some d in $\mathbb{Z}$. It follows that $x^p \equiv 1(\mathrm{mod}\ p^2)$ and $\Gamma_p = \{x \in (\mathbb{Z}/(p^2))^* | x \equiv 1(\mathrm{mod}\ p)\}$. Suppose now that x is an element of Γ_p. Then $x \equiv 1(\mathrm{mod}\ p)$ and so there uniquely exists an integer a such that $x - 1 = ap$. We define a mapping L_p by $L_p(x) = a\,(\mathrm{mod} p)$. Then L_p is a well-defined mapping of Γ_p into the additive group $(\mathbb{Z}/(p), +)$. Furthermore, L_p is an isomorphism of Γ_p onto $(\mathbb{Z}/(p), +)$, that is, we have

$L_p(ab) \equiv L_p(a) + L_p(b) (\mathrm{mod} p)$ for a, b in Γ_p. In particular, we have $L_p(y) = mL_p(x)$ for every x, y in Γ_p with $y = x^m$ $(m \in \mathbb{Z}/(p))$. Hence, $m = L_p(y)L_p(x)^{-1}$ unless $L_p(x) = 0$. Note that $L_p(x) = 0$ if and only if x is the identity element of Γ_p.

3.2 Proposed Scheme

We construct a homomorphic public key encryption whose plaintexts form the group $\mathbb{Z}/p_1 \bigoplus \mathbb{Z}/p_2 \bigoplus \cdots \bigoplus \mathbb{Z}/p_s$, where p_i are distinct primes of the same size. We discuss only the case that $s = 2$; the group P of plaintexts is $\mathbb{Z}/p_1 \bigoplus \mathbb{Z}/p_2$, where p_1 and p_2 are primes of the same size, say $|p_1| = k = |p_2|$. We may assume that p_1 is not a prime factor of $p_2 - 1$ and p_2 is not a prime factor of $p_1 - 1$ without loss of generality. The construction of an encryption for $s > 2$ is an immediate generalization of the case for $s = 2$. We set $n = p_1 \times p_2$ and then $|n| = 2k$. For an input 1^k, the instance generator $\mathcal{IG}$ outputs the descriptions of groups G, H, P, the sampling algorithm $\mathcal{SAM}$ of H, the encoder ϵ and the decryption function d as well as public keys and private keys. The groups G, H, P are defined to be $G = (\mathbb{Z}/n^2)^*, H = G^n = \{x^n \mid x \in G\}, P = \mathbb{Z}/p_1 \bigoplus \mathbb{Z}/p_2$, respectively. The pair (p_1, p_2) of the primes is the private key of the encryption. A public key is an ordered pair $(g_1, g_2) \in G \times G$ such that the order of $g_1 (\mathrm{mod}\ p_1^2)$ is p_1 and the order of $g_2 (\mathrm{mod}\ p_2^2)$ is p_2. Note that $G \cong (\mathbb{Z}/p_1^2)^* \times (\mathbb{Z}/p_2^2)^*$ and that $(\mathbb{Z}/p_1^2)^*$ and $(\mathbb{Z}/p_1^2)^*$ are the cyclic group of order $p_1(p_1 - 1)$ and $p_2(p_2 - 1)$, respectively. We also note that $P = \mathbb{Z}/p_1 \bigoplus \mathbb{Z}/p_2 \cong (\mathbb{Z}/n, +)$.

Key Generation: A public key (g_1, g_2) is established as follows. For primes p_1 and p_2, we find an element $g_1, g_2 \in G$ such that $|g_1 (\mathrm{mod}\ n^2)| = p_1$ and $|g_2 (\mathrm{mod}\ n^2)| = p_2$. Then we set (g_1, g_2) as a public key of the encryption. First, we choose randomly $h_i \in \mathbb{Z}/p_i^2$. Then we make sure that $h_i \in (\mathbb{Z}/p_i^2)^*$ by checking whether h_i is not divisible by p_i. Second, we check whether $h_i^{p_i - 1} \neq 1 (\mathrm{mod}\ p_i^2)$. If so, then the order of $h_i^{p_i - 1} (\mathrm{mod}\ p_i^2)$ is p_i. Third, using the Chinese remainder theorem algorithm, we obtain $g_i \in (\mathbb{Z}/n^2)^*$ $(i = 1, 2)$ such that $g_1 = h_1^{p_1 - 1} (\mathrm{mod}\ p_1^2)$, $g_1 = 1 (\mathrm{mod}\ p_2^2)$, $g_2 = h_2^{p_2 - 1} (\mathrm{mod}\ p_2^2)$, and $g_2 = 1 (\mathrm{mod}\ p_1^2)$. Then (g_1, g_2) satisfies $|g_1 (\mathrm{mod}\ n^2)| = p_1$ and $|g_2 (\mathrm{mod}\ n^2)| = p_2$ and is a public key of the encryption.

Sampling Algorithm: A sampling algorithm $\mathcal{SAM}$ is given as follows. Pick randomly and uniformly an element r_0 from $G = (\mathbb{Z}/n^2)^*$. We set $r = r_0^n$. This gives the probabilistic algorithm choosing an element from $H = G^n$ uniformly and randomly.

Encoder: Recall that the set of plaintexts is $P = \mathbb{Z}/p_1 \bigoplus \mathbb{Z}/p_2$. Suppose (g_1, g_2) is a public key. The encoder $\epsilon : P \to G$ is given by $\epsilon(x_1, x_2) = g_1^{x_1} g_2^{x_2}$ for $(x_1, x_2) \in \mathbb{Z}/p_1 \bigoplus \mathbb{Z}/p_2$. Note that the order of g_1 and g_2 are p_1 and p_2, respectively. Thus ϵ is well-defined.

Encryption: Suppose (m_1, m_2) is a plaintext in P $(m_1 \in \mathbb{Z}/p_1,\ m_2 \in \mathbb{Z}/p_2)$. The sampling algorithm $\mathcal{SAM}$ randomly and uniformly chooses an element r from H. Using the encryption (1), we make a ciphertext $e(m_1, m_2)$ by

$$e(m_1, m_2) = \epsilon(m_1, m_2) r = g_1^{m_1} g_2^{m_2} r \ . \tag{2}$$

Decryption: We now give the decryption function $d : G \to P$. Take an arbitrary element z from the group G. Then a mapping d of G into $P = \mathbb{Z}/p_1 \bigoplus \mathbb{Z}/p_2$ is defined by

$$d(z) = \left(\frac{L_{p_1}(z^{(p_1-1)p_2(p_2-1)} (\mathrm{mod}\ p_1^2))}{L_{p_1}(g_1^{(p_1-1)p_2(p_2-1)} (\mathrm{mod}\ p_1^2))}, \frac{L_{p_2}(z^{p_1(p_1-1)(p_2-1)} (\mathrm{mod}\ p_2^2))}{L_{p_2}(g_2^{p_1(p_1-1)(p_2-1)} (\mathrm{mod}\ p_2^2))}\right) . \quad (3)$$

It is a routine to see that $d \circ \epsilon$ is the identity mapping of P.

Take an arbitrary element $z \in H = G^n$. Then $z = w^n$ for some $w \in G$. Since the order of G is $p_1(p_1 - 1)p_2(p_2 - 1)$, we have $z^{(p_1-1)(p_2-1)} = w^{n(p_1-1)(p_2-1)} = w^{p_1(p_1-1)p_2(p_2-1)} = 1$. This implies that $d \circ \delta$ is the trivial mapping, that is, $d(\delta(z)) = (0 \,(\mathrm{mod}\ p_1), 0 \,(\mathrm{mod}\ p_2))$ for every $z \in G^n$ and so $\mathrm{Im}\delta \subset \mathrm{Ker}\, d$.

Let us now show the converse. Take z from $\mathrm{Ker}\, d$. Since G is a direct product of $(\mathbb{Z}/p_1^2)^*$ and $(\mathbb{Z}/p_2^2)^*$, there are generators f_1, f_2 of G such that $|f_1| = p_1(p_1 - 1)$ and $|f_2| = p_2(p_2 - 1)$. Then we have $z = f_1^a f_2^b$ for some a, b. Since $d(z) = (0, 0)$, we have $L_{p_1}(f_1^{a(p_1-1)}) = 0$ and so $(f_1^{p_1-1})^a = f_1^{a(p_1-1)} = 1$. This implies $p_1|a$ since $|f_1^{p_1-1}| = p_1$. Similarly we have $p_2|b$. Then $z = f_1^a f_2^b = f_1^{c_1 p_1} f_2^{c_2 p_2}$ for some c_1, c_2. Note that $|f_1^{p_1}| = p_1 - 1$ and $p_1 - 1$ and p_2 are coprime. Hence, $\mathrm{GCD(p_1(p_1 - 1), p_1 p_2)} = \mathrm{p_1}$ and so $p_1 = \alpha p_1(p_1 - 1) + \beta n$ for some α, β. Then $f_1^{\beta n} = f_1^{\alpha p_1(p_1-1)} f_1^{\beta n} = f_1^{p_1}$. Similarly $f_2^{\gamma n} = f_2^{p_2}$ for some γ. Consequently, we have $z = f_1^{c_1 p_1} f_2^{c_2 p_2} = (f_1^{c_1 \beta} f_2^{c_2 \gamma})^n \in \mathrm{Im}\delta$. Recall that δ is the inclusion of G^n into G. Therefore, we have $\mathrm{Im}\delta = \mathrm{Ker}\, d$. Now we have the exact sequence $1 \longrightarrow ((\mathbb{Z}/n^2)^*)^n \xrightarrow{\delta} (\mathbb{Z}/n^2)^* \overset{\epsilon}{\curvearrowright} P \longrightarrow 0$, where δ is the inclusion of $((\mathbb{Z}/n^2)^*)^n$ into $(\mathbb{Z}/n^2)^*$, and d is the homomorphism of $(\mathbb{Z}/n^2)^*$ onto $\mathbb{Z}/p_1 \bigoplus \mathbb{Z}/p_2$ defined above. On the other hand, ϵ is defined by $\epsilon(x_1 (\mathrm{mod}\ p_1), x_2 (\mathrm{mod}\ p_2)) = g_1^{x_1} g_2^{x_2}$ for $(x_1, x_2) \in \mathbb{Z}/p_1 \bigoplus \mathbb{Z}/p_2$. As shown in Section 2, e is a homomorphic encryption of $\mathbb{Z}/p_1 \bigoplus \mathbb{Z}/p_2$. It is also clear that d can be efficiently computed provided that the private key (p_1, p_2) is given.

3.3 Security

For an asymmetric key encryption, the indistinguishability under chosen plaintext attack (IND-CPA) [2], which is a standard requirement for encryption, is defined below. An adversary is modeled by a probabilistic polynomial time Turing machine, that is, the adversary participates in the game and yields a guess after polynomial time computation. The challenger generates a key pair (PK, SK) of public and private keys based on a security parameter k and publishes the public key PK to the adversary. On the other hand, the challenger does not publish the private key SK. The adversary is allowed to perform encryptions or other operations for his strategy to win the game. Eventually, the adversary submits two distinct chosen plaintexts m_0 and m_1 to the challenger. The challenger chooses a bit $b \in \{0, 1\}$ uniformly and randomly, and sends the ciphertext $c = e(PK, m_b)$ to the adversary. The bit b is made secret to the adversary. The adversary is allowed to perform additional computations to guess the bit b. Finally, it answers a guess for b. A cryptosystem is called *indistinguishable under chosen plaintext attack (IND-CPA)* if every probabilistic polynomial

time adversary has only a negligible advantage over random guessing, that is, if no adversary wins the game with probability significantly larger than $\frac{1}{2}$. The indistinguishability of the proposed encryption (2) is equivalent to the subgroup membership problem of the subgroup H of randomizers in G. We sketch the proof and the detailed proof will be given in the full version of the paper.

Theorem 1. *The proposed encryption e given by (2) satisfies IND-CPA if and only if the subgroup membership problem of H in G is intractable.*

Sketch of Proof. Suppose there exists an adversary who can attack the encryption with non-negligible probability. This implies that there exists a pair m_1, m_2 of messages in P such that the adversary can distinguish a ciphertext $e(m_b)$. Following the proof for the indistinguishability of ElGamal by Tsiounis and Yung [7], we use the Hoeffding inequality to obtain a message $m = (x_1, x_2) \in P$ whose encrypted message can be distinguishable from an encrypted message of a uniformly and randomly chosen message $m'(= (z_1, z_2))$ from P with non-negligible probability. Now we take an input to the subgroup membership problem of H in G, that is, $y = (y_1, y_2)$. So we would like to determine whether or not y belongs to G and so use the adversary as an oracle to solve this subgroup membership problem. Suppose the public key for the encryption is (g_1, g_2). We set $c' = y\epsilon(m) = y(g_1^{x_1}, g_2^{x_2})$. If $y \in H$, then $y\epsilon(m) = e(m)$. If $y \notin H$, then $y\epsilon(m) = e(m')$ for a certain uniformly distributed message m'. By our assumption, we can decide whether c' is a ciphertext of m or a ciphertext of a uniformly distributed plaintext m' with non-negligible probability. Therefore, we can decide whether or not the input y belongs to H and so we obtain an algorithm to solve the subgroup membership problem using the adversary as an oracle.

On the other hand, we suppose we have an algorithm to solve the subgroup membership problem. Let $m_1 = (0, 0)$ and $m_2 = (1, 1)$. Then we have $e(m_1) = r\epsilon(0, 0) = r$ for some $r \in H$. Thus, $e(m_1)$ always belongs to H. On the other hand, $e(m_2) = r\epsilon(1, 1) = r(g_1, g_2) \notin H$. Using the algorithm to solve the subgroup membership problem, we can determine whether given $e(m_i)$ is an encrypted message of $m1$ or m_2. Therefore, there exists an attack against the encryption scheme. □

It is clear that if the discrete logarithm problem in the underlying group of ElGamal is tractable, then it can be completely broken. On the other hand, the relationship between the discrete logarithm problem and the homomorphic encryption (2) is intricate. We should remark that solving the discrete logarithm problem does not give any trivial attacks against the encryption (2) in the generic group model by [6]. As a matter of fact, the security of the encryption is more closely related to the *multiple discrete logarithm problem* which is introduced in [8]. The multiple discrete logarithm problem is formulated as follows. Let G be a finite group isomorphic to $C \times D$, where C and D are cyclic group. Then, $G = \langle g_1, g_2 \rangle$ for some generators g_1 and g_2. The *multiple discrete logarithm problem* is to compute (x, y) for given $g \in G$, where $g = g_1^x g_2^y$. In the generic model, the multiple discrete logarithm problem is shown essentially harder than the discrete

logarithm problem [8]. This implies that an oracle of the discrete logarithm problem does not help to break the cryptosystem (2) in the generic model.

Every element in the underlying group G is generated by two elements, whereas the discrete logarithm oracle gives a correct answer only when it is given the correct pair of group elements a and a^x. The hardness of using the discrete logarithm oracle to solve the multiple discrete logarithm problem comes form the hardness of finding a non-trivial pair (h_1, h_2) of group elements which can be written as $h_1 = a$ and $h_2 = a^x$ for some a and x. The trapdoor of the scheme (2) is the Okamoto-Uchiyama logarithmic function with the primes p_1, p_2 and it helps to solve the multiple discrete logarithmic problem with respect to the public key. We should note that the generic model does not guarantee the security against the attack using the properties of the representations of the underlying group. We should also remark that the encryption can be completely broken if integer factoring is efficiently computed.

4 Products of Encryptions

We now discuss how to construct a new encryption from the old ones related to the proposed scheme (2). This implies that we define an operation on encryptions. Suppose that e_1 and e_2 are encryptions (2) related to the group $(\mathbb{Z}/n_1^2)^*$ and $(\mathbb{Z}/n_2^2)^*$, respectively, where n_1 and n_2 are composites of two primes of the same size. We may assume that $\mathrm{GCD}(\phi(\mathrm{n}_1), \mathrm{n}_2) = 1$ and $\mathrm{GCD}(\mathrm{n}_1, \phi(\mathrm{n}_2)) = 1$. Suppose the private key of e_1 is retained by Alice, whereas the private key of e_2 is retained by Bob.

Note that n_1 and n_2 are composites of primes of the same size, respectively, (say $n_1 = p_1p_2, n_2 = q_1q_2$). Then we can define the encryption related to n_1n_2, that is, $P = \mathbb{Z}/n_1n_2$ and $G = (\mathbb{Z}/(n_1n_2)^2)^*$. This is basically same as the general case of $s = 4$ in (2), however, there exists a big difference in the sense that the private key p_1, p_2 are retained by Alice whereas q_1, q_2 are retained by Bob. Therefore, the private keys are divided into two parts and each half is retained by each entities; Alice retains p_1, p_2 and Bob retains q_1, q_2. So it is not necessarily to appeal to a trusted third party to establish a new public key when the two entities agree to share a public key encryption. Alice and Bob can compute public keys for the new encryption without showing their private keys. Alice can compute $g_1, g_2 \in (\mathbb{Z}/(n_1n_2)^2)^*$ such that $|g_1| = p_1$ and $|g_2| = p_2$ using Chinese remainder algorithm, similarly Bob can compute $g_3, g_4 \in (\mathbb{Z}/(n_1n_2)^2)^*$ such that $|g_3| = q_1$ and $|g_4| = q_2$. Then a public key is (g_1, g_2, g_3, g_4). The encryption of a plaintext (x_1, x_2, x_3, x_4) is computed as $\epsilon(x_1, x_2, x_3, x_4)r = g_1^{x_1}g_2^{x_2}g_3^{x_3}g_4^{x_4}r$, where r is chosen uniformly and randomly from $((\mathbb{Z}/(n_1n_2)^2)^*)^{n_1n_2}$. The decryption is defined accordingly using Okamoto-Uchiyama logarithmic function. Let us denote the resulting encryption by $e_1 \circledcirc e_2$ and call it the *product* of e_1 and e_2.

The encryption $e_1 \circledcirc e_2$ has the following properties. The group of plaintexts is $\mathbb{Z}/p_1 \bigoplus \mathbb{Z}/p_2 \bigoplus \mathbb{Z}/q_1 \bigoplus \mathbb{Z}/q_2$ and ciphertexts lies in $(\mathbb{Z}/(n_1n_2)^2)^*$. Let us suppose $c = e_1 \circledcirc e_2(x_1, x_2, x_3, x_4)$. Then Alice can retrieve only x_1 and x_2, whereas Bob can retrieve only x_3 and x_4.

A ciphertext of $e_1 \circledcirc e_2$ is an element of $(\mathbb{Z}/(n_1 n_2)^2)^*$ and so it is hard to tamper a ciphertext. One can construct a new encryption just by concatenate two encryptions, that is, a new encryption is defined by $e(m_1, m_2) = (e_1(m_1), e_2(m_2))$ for two existing encryptions e_1 and e_2. It is quite easy to tamper with $e(m_1, m_2)$ to obtain $e(m_1, l_2)$ because one may just replace $e_2(m_2)$ by $e_2(l_2)$. On the other hand, $e_1 \circledcirc e_2$ is resistant to such an attack. This property is desired for construction of electronic voting schemes with multi-authorities. In such schemes, each authority would like to share homomorphic encryption, however, it would like to retain its own secret information.

A homomorphic encryption of a direct sum of groups is also desirable for multi-candidate election. Such a scheme is discussed in [1], in which a proof of validity must be provided in addition to ciphertext by a homomorphic encryption. A general n-bit cryptographic counter is constructed using 1-bit counter in [3]. The scheme also uses non-interactive zero-knowledge proof. It is possible to construct a cryptographic counter without an additional proof of validity using the proposed scheme. We shall discuss applications to election schemes and cryptographic counters in the full version of the paper.

References

1. Cramer, R., Gennaro, R., Schoenmakers, B.: A Secure and Optimally Efficient Multi-Authority Election Scheme. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 103–118. Springer, Heidelberg (1997)
2. Goldwasser, S., Micali, S.: Probabilistic Encryption. Jounal of Computer and System Sciences 28, 270–299 (1984)
3. Katz, J., Myers, S., Ostrovsky, R.: Cryptographic Counters and Applications to Electronic Voting. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 78–92. Springer, Heidelberg (2001)
4. Okamoto, T., Uchiyama, S.: A New Public-key Cryptosystem as Secure as Factoring. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 308–318. Springer, Heidelberg (1998)
5. Paillier, P.: Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)
6. Shoup, V.: Lower Bounds for Discrete Logarithms and Related Problems. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 256–266. Springer, Heidelberg (1997)
7. Tsiounis, Y., Yung, M.: On the security of ElGamal based encryption. In: Imai, H., Zheng, Y. (eds.) PKC 1998. LNCS, vol. 1431, pp. 117–134. Springer, Heidelberg (1998)
8. Yamamura, A., Kurosawa, K.: Generic Algorithms and Key Agreement Protocols Based on Group Actions. In: Eades, P., Takaoka, T. (eds.) ISAAC 2001. LNCS, vol. 2223, pp. 208–218. Springer, Heidelberg (2001)
9. Yamamura, A., Saito, T.: Private Information Retrieval Based on the Subgroup Membership Problem. In: Varadharajan, V., Mu, Y. (eds.) ACISP 2001. LNCS, vol. 2119, pp. 206–220. Springer, Heidelberg (2001)

Author Index

Bac, Dang Hoai 301
Berhuy, Grégory 90
Bernstein, Daniel J. 20, 291
Binh, Nguyen 301
Boztaş, Serdar 120
Bracken, Carl 72
Bras-Amorós, Maria 337
Byrne, Eimear 72

Charon, Irène 267
Cohen, Gérard 267
Cui, Yang 168

Dai, Xiaoping 60
Das, M. Prem Laxman 237

Embury, P. 281

Fujitsu, Satoshi 80
Fujiwara, Eiji 158
Fujiwara, Toru 110

Geil, Olav 50
Gong, Guang 7
Gulliver, T.A. 311
Guruswami, Venkatesan 1

Hagiwara, Manabu 80, 168
Helleseth, Tor 7
Hollanti, Camilla 227
Hudry, Olivier 267
Høholdt, Tom 18

Imai, Hideki 80, 168

Janwa, H. 347
Jin, Seok-Yong 188
Justesen, Jørn 18

Kaneko, Haruhiko 158
Kashyap, Navin 198
Kavut, Selçuk 321
Kim, Young-Joon 188
Kitagawa, Takashi 80
Kobara, Kazukuni 168
Krithivasan, Dinesh 178

Lahtonen, Jyrki 247
Laigle-Chapuy, Yann 130
Lal, A.K. 347
Lange, Tanja 20
Lobstein, Antoine 267
Lu, Hsiao-feng (Francis) 227

Maheshanand 330
Maitra, Subhamoy 100, 271
Markin, Nadya 72
Matsumoto, Ryutaroh 50
McGuire, Gary 28, 72
Medoš, Silvana 120
Mow, Wai Ho 60

Niederreiter, Harald 208
Nuida, Koji 80

O'Sullivan, Michael E. 337
Ogawa, Kazuto 80
Oggier, Frédérique 90, 138
Ohta, Kazuo 257
Ota, Haruki 257

Paul, Goutam 100
Pinnawala, N. 311
Pradhan, S. Sandeep 178
Pujol, J. 148

Quynh, Nguyen Xuan 301

Raj, Safitha J. 217
Rao, A. 281, 311
Rifà, J. 148
Rønjom, Sondre 7
Rudra, Atri 38

Sarkar, Sumanta 271
Sethuraman, B.A. 138
Shankar, Priti 47
Sikdar, Kripasindhu 237
Solov'eva, F.I. 148
Song, Hong-Yeop 188
Srivastava, Rohit 100
Stichtenoth, Henning 48

Thangaraj, Andrew 217

Vehkalahti, Roope 247

Wasan, Siri Krishan 330
Watanabe, Hajime 80
Winterhof, Arne 208

Yamakawa, Shigenori 168
Yamamura, Akihiro 357
Yasunaga, Kenji 110
Yoneyama, Kazuki 257
Yücel, Melek Diker 321

Zhou, Jianqin 60