Marc Fossorier
Hideki Imai
Shu Lin
Alain Poli (Eds.)

LNCS 3857

Applied Algebra, Algebraic Algorithms and Error-Correcting Codes

16th International Symposium, AAECC-16
Las Vegas, NV, USA, February 2006
Proceedings

Springer

Lecture Notes in Computer Science 3857

Marc Fossorier Hideki Imai
Shu Lin Alain Poli (Eds.)

Applied Algebra, Algebraic Algorithms and Error-Correcting Codes

16th International Symposium, AAECC-16
Las Vegas, NV, USA, February 20-24, 2006
Proceedings

Springer

Volume Editors

Marc Fossorier
University of Hawaii, Department of Electrical Engineering
2540 Dole St., Holmes Hall 483, Honolulu, HI 96822, USA
E-mail: marc@spectra.eng.hawaii.edu

Hideki Imai
University of Tokyo, Institute of Industrial Science
Komaba, Meguro-ku, Tokyo 153-8505, Japan
E-mail: imai@iis.u-tokyo.ac.jp

Shu Lin
University of California, Department of Electrical and Computer Engineering
Davis One Shields Avenue, Davis, CA 95615, USA
E-mail: shulin@ece.udavis.edu

Alain Poli
University Paul Sabatier, AAECC/IRIT
118 route de Narbonne, 31067 Toulouse cedex, France
E-mail: poli@cict.fr

Library of Congress Control Number: 2005938816

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 3-540-31423-7 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-31423-3 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11617983 06/3142 5 4 3 2 1 0

Preface

The AAECC symposium was started in June 1983 by Alain Poli (Toulouse), who, together with R. Desq, D. Lazard, and P. Camion, organized the first conference. The meaning of the acronym AAECC changed from "Applied Algebra and Error Correcting Codes" to "Applied Algebra, Algebraic Algorithms, and Error Correcting Codes." One reason was the increasing 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 is conserved as in the past, but slightly more oriented to algebraic geometry codes, finite fields, complexity, polynomials, and graphs.

For AAECC-16 the main subjects covered were:

- Block codes.
- Algebra and codes: rings, fields, AG codes.
- Cryptography.
- Sequences.
- Algorithms, decoding algorithms.
- Iterative decoding: code construction and decoding algorithms.
- Algebra: constructions in algebra, Galois group, differential algebra, polynomials.

Four invited speakers characterize the outlines of AAECC-16:

- C. Carlet ("On Bent and Highly Nonlinear Balanced/Resilient Functions and their Algebraic Immunities").
- S. Gao ("Grobner Bases and Linear Codes").
- R.J. McEliece ("On Generalized Parity Checks").
- T. Okamoto ("Cryptography Based on Bilinear Maps").

Except for AAECC-1 (Discrete Mathematics, 56, 1985) and AAECC-7 (Discrete Mathematics, 33, 1991), the proceedings of all the symposia have been published in Springer's *Lecture Notes in Computer Science* series (vol. 228, 229, 307, 356, 357, 508, 673, 948, 1255, 1719, 2227, 2643). It is a policy of AAECC to maintain a high scientific standard. This has been made possible thanks to the many referees involved. Each submitted paper was evaluated by at least two international researchers.

AAECC-16 received 32 submissions; 25 were selected for publication in these proceedings while 7 additional works contributed to the symposium as oral presentations. In addition to the four invited speakers, five invited papers also contributed to these proceedings.

The symposium was organized by Marc Fossorier, Shu Lin, Hideki Imai and Alain Poli, with the help of the 'Centre Baudis' in Toulouse.

We express our thanks to Springer staff, especially to Alfred Hofmann and Anna Kramer, as well as to the referees.

November 2005

M. Fossorier
S. Lin
H. Imai
A. Poli

Organization

Steering Committee

Conference General Chairman:	Shu Lin (Univ. of Davis, USA)
Conference Co-chairmen:	H. Imai (Univ. of Tokyo, Japan), Alain Poli (Univ. of Toulouse, France)
Publication:	Marc Fossorier (Univ. of Hawaii, USA)
Local Arrangements:	Fay Horie (Univ. of Hawaii, USA)

Conference Committee

J. Calmet
G. Cohen
S.D. Cohen
G.L. Feng
M. Giusti
J. Heintz
T. Hoehold
H. Imai
H. Janwa
J.M. Jensen
R. Kohno
H.W. Lenstra Jr.
S. Lin
O. Moreno
H. Niederreiter
A. Poli
T.R.N. Rao
S. Sakata
P. Sole

Program Committee

T. Berger
E. Biglieri
J. Calmet
C. Carlet
D. Costello
T. Ericson
P. Farrell
M. Fossorier
J. Hagenauer
S. Harari
T. Helleseth
E. Kaltofen
T. Kasami
L.R. Knudsen
S. Lietsyn
R.J. McEliece
R. Morelos-Zaragoza
H. Niederreiter
P. Sole
H. Tilborg

Table of Contents

On Bent and Highly Nonlinear Balanced/Resilient Functions and Their Algebraic Immunities

Claude Carlet*

INRIA, Projet CODES, BP 105 - 78153, Le Chesnay Cedex, France
claude.carlet@inria.fr

Abstract. Since the introduction of the notions of nonlinearity in the mid-70's (the term has been in fact introduced later), of correlation immunity and resiliency in the mid-80's, and of algebraic immunity recently, the problem of efficiently constructing Boolean functions satisfying, at high levels, one or several of these criteria has received much attention. Only few primary constructions are known, and secondary constructions are also necessary to obtain functions achieving or approaching the best possible cryptographic characteristics. After recalling the background on cryptographic criteria and making some general observations, we try to give a survey of all these constructions and their properties. We then show that a nice and simple property of Boolean functions leads to a general secondary construction building an n-variable function from three known n-variable functions. This construction generalizes secondary constructions recently obtained for Boolean bent functions and also leads to secondary constructions of highly nonlinear balanced or resilient functions, with potentially better algebraic immunities than the three functions used as building blocks.

Keywords: stream cipher, Boolean function, algebraic degree, resiliency, nonlinearity, algebraic attack.

1 Introduction

Boolean functions, that is, F_2-valued functions defined on the vector space F_2^n of all binary words of a given length n, are used in the S-boxes of block ciphers and in the pseudo-random generators of stream ciphers. They play a central role in their security. The generation of the keystream consists, in many stream ciphers, of a linear part, producing a sequence with a large period, usually composed of one or several LFSR's, and a nonlinear combining or filtering function f which produces the output, given the state of the linear part. The main classical cryptographic criteria for designing such function f are balancedness (f is balanced if its Hamming weight equals 2^{n-1}) to prevent the system from leaking statistical information on the plaintext when the ciphertext is known, a high algebraic

[*] Also member of the University of Paris 8 (MAATICAH).

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 1–28, 2006.

degree (that is, a high degree of the algebraic normal form of the function) to prevent the system from Massey's attack by the Berlekamp-Massey algorithm, a high order of correlation immunity (and more precisely, of resiliency, since the functions must be balanced) to counter correlation attacks (at least in the case of combining functions), and a high nonlinearity (that is, a large Hamming distance to affine functions) to withstand correlation attacks (again) and linear attacks.

The recent algebraic attacks have led to further characteristics of Boolean functions. These attacks recover the secret key by solving an overdefined system of multivariate algebraic equations. The scenarios found in [26], under which low degree equations can be deduced from the knowledge of the nonlinear combining or filtering function, have led in [48] to a new parameter, the (basic) algebraic immunity, which must be high. This condition is itself not sufficient, since a function can have sufficiently high algebraic immunity and be weak against fast algebraic attacks [25]. A further criterion strengthening the basic notion of algebraic immunity can be defined accordingly.

The problems of designing numerous bent functions (that is, functions with highest possible nonlinearity) and of efficiently constructing highly nonlinear balanced (or, if necessary, resilient) functions with high algebraic degrees have been receiving much attention for several years. They are relevant to several domains: mainly cryptography, but also combinatorics, design theory, coding theory ... Few primary constructions (in which the functions are designed *ex nihilo*) are known, and secondary constructions (which use already defined functions to design new ones) are also necessary to obtain functions, on a sufficient number of variables, achieving or approaching the best possible cryptographic characteristics. We can say that research has obtained limited but non-negligible success in these matters. However, the problem of meeting all of these characteristics at sufficient levels and, also, achieving high algebraic immunities, with functions whose outputs can be fastly computed (this is also a necessary condition for using them in stream ciphers) shows some resistance. The most efficient primary construction in this matter has been obtained in [29] (the authors present their result as a secondary construction, but as they observe themselves, their construction is just a direct sum of a function taken as a building block, with a function that they design and which corresponds to a primary construction). It leads to functions in any even numbers of variables and with optimal algebraic immunities. And as shown in [19], their algebraic degrees are very high and their output can be very fastly computed. They are not balanced, but any function! can be made balanced by adding one variable. The remaining problem is in their insufficient nonlinearities, which makes them unusable in cryptosystems. Used as a secondary construction, their method does not give full satisfaction either, for the same reason. Hence, this secondary construction represents a very nice but still partial step towards a good tradeoff between nonlinearity, resiliency and algebraic immunity.

Most classical primary or secondary constructions of highly nonlinear functions seem to produce insufficient algebraic immunities. For instance, the

10-variable Boolean function used in the LILI keystream generator (a submission to NESSIE European call for cryptographic primitives) is built following [56] by using classical constructions; see [59]. It has algebraic immunity 4 and is responsible for the lack of resistance of LILI to algebraic attacks, see [26].

As shown in [48], taking random balanced functions on sufficiently large numbers of variables could suffice to withstand algebraic attacks on the stream ciphers using them. It would also withstand fast algebraic attacks (this can be checked with the same methods as in [48]). As shown in [49], it would moreover give reasonable nonlinearities. But such solution would imply using functions on large numbers of variables, whose outputs would be computable in much too long time. This would not allow acceptable efficiency of the corresponding stream ciphers. It would not allow nonzero resiliency orders either.

The present paper tries to present the state of the art on Boolean cryptographic functions and to suggest several directions for further research. At the end of the paper, a construction (first presented in [17]) of functions on F_2^n from functions on F_2^n is presented, which combined with the classical primary and secondary constructions can lead to functions achieving high algebraic degrees, high nonlinearities and high resiliency orders, and which also allows attaining potentially high algebraic immunity. The same principle allows constructing bent functions too.

2 Preliminaries and General Observations

In some parts of this paper, we will deal in the same time with sums modulo 2 and with sums computed in $\mathbb{Z}$. We denote by $\oplus$ the addition in F_2 (but we denote by $+$ the addition in the field F_2 and in the vector space F_2^n, since there will be no ambiguity) and by $+$ the addition in $\mathbb{Z}$. We denote by $\bigoplus_{i\in\ldots}$ (resp. $\sum_{i\in\ldots}$) the corresponding multiple sums. Let n be any positive integer. Any Boolean function f on n variables admits a unique algebraic normal form (A.N.F.):

$$f(x_1,\ldots,x_n)=\bigoplus_{I\subseteq\{1,\ldots,n\}} a_I \prod_{i\in I} x_i,$$

where the a_I's are in F_2. The terms $\prod_{i\in I} x_i$ are called *monomials*. The *algebraic degree* $d^\circ f$ of a Boolean function f equals the maximum degree of those monomials with nonzero coefficients in its algebraic normal form. *Affine functions* are those Boolean functions of degrees at most 1.

Another representation of Boolean functions is also very useful. The vector space F_2^n can be endowed with the structure of the field F_2 , since this field is an n-dimensional F_2-vector space. The function $(u,v)\mapsto tr(u\,v)$, where $tr(u)=u+u^2+u^{2^2}+\cdots+u^{2\quad -1}$ is the *trace function*, is an inner product in F_2 . Every Boolean function can be written in the form $f(x)=tr(F(x))$ where F is a mapping from F_2 into F_2 , and this leads to the *trace representation*: $f(x)=tr\left(\sum_{i=0}^{2\quad -1}\beta_i\,x^i\right)$, where $\beta_i\in F_2$. Thanks to the fact that $tr(u^2)=tr(u)$ for every $u\in F_2$, we can restrict the exponents i with nonzero coefficients β_i so that there is at most one such exponent in each cyclotomic class $\{i \times 2^j\,[\bmod\ (2^n-1)]\,;\ j \in N\}$.

The *Hamming weight* $w_H(f)$ of a Boolean function f on n variables is the size of its support $\{x \in F_2^n;\ f(x)=1\}$. The *Hamming distance* $d_H(f,g)$ between two Boolean functions f and g is the Hamming weight of their difference $f \oplus g$. The *nonlinearity* of f is its minimum distance to all affine functions. Functions used in stream or block ciphers must have high nonlinearities to resist the attacks on these ciphers (correlation and linear attacks, see [4, 40, 41, 58]). The nonlinearity of f can be expressed by means of the discrete Fourier transform of the "sign" function $\chi_f(x) = (-1)^{f(x)}$, equal to $\widehat{\chi_f}(s) = \sum_{x\in F_2}(-1)^{f(x)\oplus x\cdot s}$ (and which is called the *Walsh transform*, or Walsh-Hadamard transform): the distance $d_H(f,l)$ between f and the affine function $l(x) = s\cdot x \oplus \epsilon$ $(s \in F_2^n;\ \epsilon \in F_2)$ and the number $\widehat{\chi_f}(s)$ are related by:

$$\widehat{\chi_f}(s) = (-1)^{\epsilon}(2^n - 2d_H(f,l)) \tag{1}$$

and the nonlinearity N_f of any Boolean function on F_2^n is therefore related to the Walsh spectrum of χ_f via the relation:

$$N_f = 2^{n-1} - \frac{1}{2}\max_{s\in F_2} |\widehat{\chi_f}(s)|. \tag{2}$$

It is upper bounded by $2^{n-1} - 2^{n/2-1}$ because of the so-called Parseval's relation $\sum_{s\in F_2}\widehat{\chi_f}^2(s) = 2^{2n}$.

A Boolean function is called *bent* if its nonlinearity equals $2^{n-1} - 2^{n/2-1}$, where n is necessarily even. Then, its distance to every affine function equals $2^{n-1} \pm 2^{n/2-1}$, according to Parseval's relation again and to (1).

A Boolean function f is bent if and only if all of its *derivatives* $D_af(x) = f(x)\oplus f(x+a)$ are balanced, (see [53]). Hence, f is bent if and only if its support is a *difference set* (cf. [30]).

If f is bent, then the *dual* Boolean function $\widetilde{f}$ defined on F_2^n by $\widehat{\chi_f}(s) = 2^{\frac{n}{2}}\chi_{\widetilde{f}}(s)$ is bent. The dual of $\widetilde{f}$ is f itself. The mapping $f \mapsto \widetilde{f}$ is an isometry (the Hamming distance between two bent functions is equal to that of their duals).

The notion of bent function is invariant under linear equivalence and it is independent of the choice of the inner product in F_2^n (since any other inner product has the form $\langle x, s\rangle = x \cdot L(s)$, where L is an auto-adjoint linear isomorphism).

Rothaus' inequality [53] states that any bent function has algebraic degree at most $n/2$. Algebraic degree being an important complexity parameter, bent functions with high degrees are preferred from cryptographic viewpoint.

The class of bent functions, whose determination or classification is still an open problem, is relevant to cryptography (cf. [47]), to algebraic coding theory (cf. [45]), to sequence theory (cf. [51]) and to design theory (any difference set can be used to construct a symmetric design, cf. [1], pages 274-278). More information on bent functions can be found in the survey paper [10] or in the more recent chapter [18].

The class of bent functions is included in the class of the so-called *plateaued* functions. This notion has been introduced by Zheng and Zhang in [62]. A function is called plateaued if its Walsh transform takes at most three values 0 and $\pm\lambda$ (where λ is some positive integer, that we call the *amplitude* of the plateaued function). Because of Parseval's relation, λ must be of the form 2^r where $r \geq \frac{n}{2}$, and the suppport $\{s \in F_2^n / \widehat{\chi_f}(s) \neq 0\}$ of the Walsh transform of a plateaued function of amplitude 2^r has size 2^{2n-2r}.

Bent functions cannot be *balanced*, i.e. have uniformly distributed output. Hence, they cannot be used without modifications in the pseudo-random generator of a stream cipher, since this would leak statistical information on the plaintext, given the ciphertext[1]. Finding balanced functions with highest known nonlinearities is an important cryptographic task, as well as obtaining the best possible upper bounds on the nonlinearities of balanced functions. A nice way of designing highly nonlinear balanced functions is due to Dobbertin [33]: taking a bent function f which is constant on an $n/2$-dimensional flat A of F_2^n and replacing the values of f on A by the values of a highly nonlinear balanced function on A (identified to a function on $F_2^{n/2}$). The problem of similarly modifying bent functions into resilient functions (see definition below) has been studied in [46].

After the criteria of balancedness, high algebraic degree and high nonlinearity, which are relevant to all stream ciphers, another important cryptographic criterion for Boolean functions is resiliency. It plays a central role in their security, at least in the case of the standard model – the combination generator (cf. [57]). In this model, the vector whose coordinates are the outputs to n linear feedback shift registers is the input to a Boolean function. The output to the function during N clock cycles produces the keystream (of length N, the length of the plaintext), which is then (as in any stream cipher) bitwise xored with the message to produce the cipher. Some divide-and-conquer attacks exist on this method of encryption (cf. [4, 40, 41, 58]). To withstand these *correlation attacks*, the distribution probability of the output to the function must be unaltered when any m of its inputs are fixed [58], with m as large as possible. This property, called *m-th order correlation-immunity* [57], is characterized by the set of zero values in the Walsh spectrum [61]: f is m-th order correlation-immune if and only if $\widehat{\chi_f}(u) = 0$, for all $u \in F_2^n$ such that $1 \leq w_H(u) \leq m$, where $w_H(u)$ denotes the Hamming weight of the n-bit vector u, (the number of its nonzero components). Balanced m-th order correlation-immune functions are called *m-resilient* functions. They are characterized by the fact that $\widehat{\chi_f}(u) = 0$ for all $u \in F_2^n$ such that $0 \leq w_H(u) \leq m$.

The notions of correlation immune and resilient functions are not invariant under linear equivalence; they are invariant under translations $x \mapsto x + a$, since, if $g(x) = f(x + a)$, then $\widehat{\chi_g}(u) = \widehat{\chi_f}(u)(-1)^{a \cdot u}$, under permutations of the input coordinates, and when n is even, under an additional involution (see [38]).

Siegenthaler's inequality [57] states that any m-th order correlation immune function on n variables has degree at most $n - m$, that any m-resilient function

[1] However, as soon as n is large enough (say $n \geq 20$), the bias $\frac{2^{\ /2-1}}{2^{\ }-1}$ between their weights and the weight of balanced functions is quite small.

$(0 \leq m < n-1)$ has algebraic degree smaller than or equal to $n-m-1$ and that any $(n-1)$-resilient function has algebraic degree 1. We shall call *Siegenthaler's bound* this property.

Sarkar and Maitra have shown that the Hamming distance between any m-resilient function and any affine function is divisible by 2^{m+1} (this divisibility bound is improved in [11, 23] for functions with specified algebraic degrees). This leads to an upper bound on the nonlinearity of m-resilient functions (also partly obtained by Tarannikov and by Zhang and Zheng): the nonlinearity of any m-resilient function is smaller than or equal to $2^{n-1} - 2^{m+1}$ if $\frac{n}{2} - 1 < m + 1$, to $2^{n-1} - 2^{\frac{n}{2}-1} - 2^{m+1}$ if n is even and $\frac{n}{2} - 1 \geq m + 1$ and to $2^{n-1} - 2^{m+1} \left\lceil 2^{n/2-m-2} \right\rceil$ if n is odd and $\frac{n}{2} - 1 \geq m + 1$. We shall call this set of upper bounds *Sarkar et al.'s bound*. A similar bound exists for correlation immune functions, but we do not recall it since non-balanced correlation immune functions present little cryptographic interest.

Two kinds of constructions, providing resilient functions with degrees and nonlinearities approaching or achieving the known bounds, can be identified. Some constructions give direct definitions of Boolean functions. There are few such *primary* constructions and new ideas for designing them are currently lacking. Except for small values of the number of variables, the only known primary construction of resilient functions which leads to a wide class of such functions, the Maiorana-McFarland's construction, does not allow designing balanced or resilient functions with high degrees and high nonlinearities (see e.g. [12, 13]), for which the trade-off between these parameters achieve the bounds recalled above. Moreover, the stream ciphers using the constructed functions are subject to the time-memory-data trade-off attack (see [42]). Modifications and generalizations of this construction have been proposed (see e.g. [12, 16, 50, 55]), but these generalizations lead to classes with roughly the same properties as the original class. *Secondary* constructions use previously defined functions (that we shall call "building blocks") to build new ones. Most of them design n-variable functions from m-variable functions with $m < n$ and lead in practice to recursive constructions.

Until recently, these criteria were the only requirements needed for the design of the function f used in a stream cipher as a combining function or as a filtering one (in the filter model, a single LFSR of greater length is used and the input to the n-variable Boolean function is given by a subset of n positions in this LFSR). The recent algebraic attacks [25, 26] have changed this situation by adding new criteria of considerable importance to this list. Algebraic attacks exploit multivariate relations involving key/state bits and output bits of f. If one such relation (or, better, several) is found that is of low degree in the key/state bits, algebraic attacks are very efficient. It is demonstrated in [26] that low degree relations and thus successful algebraic attacks exist for several well known constructions of stream ciphers that are immune to all previously known attacks. These low degree relations are obtained by multiplying the Boolean function f by a well chosen low degree nonzero function g, such that the product function

fg (that is, the function which support equals the intersection of the supports of f and g) has also low degree.

The scenarios found in [26], under which functions $g \neq 0$ and h of degrees at most d exist such that $fg = h$, have been simplified in [48] into two scenarios: (1) there exists a nonzero Boolean function g of degree at most d whose support is disjoint from the support of f, i.e. such that $fg = 0$ (such a function g is called an *annihilator* of f); (2) there exists a nonzero annihilator, of degree at most d, of $f \oplus 1$ (we write then: $g \preceq f$).

The (basic) *algebraic immunity* $AI(f)$ of a Boolean function f is the minimum value of d such that f or $f \oplus 1$ admits a nonzero annihilator of degree d. Obviously, $AI(f)$ is upper bounded by the degree $d^\circ f$. It should be high enough (at least equal to 7).

When the total number $1 + \ldots + \binom{n}{d}$ of monomials of degrees at most d is strictly greater than 2^{n-1}, these monomials and their products with f cannot be linearly independent. This proves, as observed in [26], that the algebraic immunity of any function f satisfies $AI(f) \leq \lceil n/2 \rceil$. This implies that Boolean functions used in stream ciphers must have at least 13 variables. In fact, 13 is very probably insufficient.

Another upper bound on $AI(f)$, which involves the nonlinearity of f, has been proved in [28]: $\sum_{i=0}^{AI(f)-2} \binom{n}{i} \leq N_f$. It is a consequence of the double inequality $\sum_{i=0}^{AI(f)-1} \binom{n}{i} \leq w_H(f) \leq \sum_{i=0}^{n-AI(f)} \binom{n}{i}$, which also implies that a function f such that $AI(f) = \frac{n+1}{2}$ (n odd) must be balanced.

There is more generally a relationship between $AI(f)$ and the minimum distance $N_f^{(r)}$ between f and all Boolean functions of degrees at most r (the so-called Reed-Muller code of order r), that we shall call the r-th order nonlinearity of f. We have $\sum_{i=0}^{AI(f)-r-1} \binom{n}{i} \leq N_f^{(r)}$ (see [19]). Moreover:

Proposition 1. *If* $AI(f) \leq r$ *and if* f *is balanced, then we have* $N_f^{(r)} \leq 2^{n-1} - 2^{n-r}$.

Proof. By hypothesis, there exists a nonzero function g of degree at most r such that $g \preceq f$ or $g \preceq f \oplus 1$. Since g is nonzero and belongs to the Reed-Muller code of order r, it has weight at least the minimum distance of this code, that is 2^{n-r}. If $g \preceq f$, then $d_H(f, g) = w_H(f \oplus g) = w_H(f) - w_H(g) \leq 2^{n-1} - 2^{n-r}$. If $g \preceq f \oplus 1$, then $d_H(f, g \oplus 1) = w_H(f \oplus g \oplus 1) = w_H(f \oplus 1) - w_H(g) \leq 2^{n-1} - 2^{n-r}$. This implies in both cases that $N_f^{(r)} \leq 2^{n-1} - 2^{n-r}$. ◇

This observation opens a direction for research: finding balanced functions with r-th order nonlinearity strictly greater than $2^{n-1} - 2^{n-r}$ for some high value of r. A combinatorial argument shows that such functions exist almost surely as soon as $r \leq .17 \times n$. Indeed, the number of n-variable Boolean functions of algebraic degrees at most r equals $2^{1+n+\binom{}{2}+\ldots+\binom{}{}}$. Such a function h being given, the number of those Boolean functions f such that the Hamming distance $d_H(f, h)$ satisfies $d_H(f, h) \leq 2^{n-1} - R$ for some positive number R equals $\displaystyle\sum_{0 \leq i \leq 2^{\;\,-1} - R} \binom{2^n}{i}$.

It is known (see [45], page 310) that, for every integer N and every $\delta < 1/2$, the number $\sum_{0\leq i\leq \delta N}\binom{N}{i}$ is upper bounded by $2^{NH_2(\delta)}$, and it is noticed in [15] that $2^{NH_2(\delta)} < 2^{N-2N(\frac{1}{2}-\delta)^2\,\log_2 e}$. Hence, $\sum_{0\leq i\leq 2\ \ ^{-1}-R}\binom{2^n}{i}$ is upper bounded by $2^{2\ \ -2^{-\ \ +1}R^2\,\log_2 e}$, and the number of those Boolean functions such that $N_f^{(r)}\leq 2^{n-1}-R$ is therefore smaller than $2^{1+n+\binom{}{2}+\ldots+\binom{}{}+2\ \ -2^{-\ \ +1}R^2\,\log_2 e}$. According to [45] again, we have: $1+n+\binom{n}{2}+\ldots+\binom{n}{r}\leq 2^{nH_2(r/n)}$. The probability that a random n-variable Boolean function f satisfies $N_f^{(r)}\leq 2^{n-1}-2^{n-r}$ is then smaller than $2^{2\ \ \ ^{2(\ /\ )}-2\ \ ^{(1-2\ /\ )+1}\,\log_2 e}$. It is a simple matter to show that, when $r/n\leq .17$, this probability tends to 0 when n tends to infinity.

A high value of $AI(f)$ is not a sufficient property for a resistance to algebraic attacks, because of fast algebraic attacks [25], in which h can have a greater degree than g. Indeed, while the complexity of the standard algebraic attack is roughly $O\left(\binom{n}{AI(f)}^3\right)$, the complexity of the fast algebraic attack, when functions $g\neq 0$ and h have been found such that $fg=h$, is roughly $O\left(\binom{n}{d^\circ g}\binom{n}{d^\circ h}\log_2\left(\binom{n}{d^\circ h}\right)+\binom{n}{d^\circ g}^3+\binom{n}{d^\circ h}{\log_2}^2\left(\binom{n}{d^\circ h}\right)\right)$ [36]. Similarly as above, when the number of monomials of degrees at most e, plus the number of monomials of degrees at most d, is strictly greater than 2^n – that is, when $d^\circ g+d^\circ h\geq n$ – there exist $g\neq 0$ of degree at most e and h of degree at most d such that $fg=h$. An n-variable function f is then optimal with respect to fast! algebraic attacks if there do not exist two functions $g\neq 0$ and h such that $fg=h$ and $d^\circ g+d^\circ h<n$. Very little research in this direction has been done already.

3 The Known Constructions of Bent Functions and of Resilient Functions and the Corresponding Degrees, Nonlinearities and Algebraic Immunities

3.1 Primary Constructions

Maiorana-McFarland Constructions. Maiorana-McFarland class (cf. [31]) is the set of all the (bent) Boolean functions on $F_2^n=\{(x,y),x,y\in F_2^{\frac{\ }{2}}\}$ (n even) of the form :

$$f(x,y)=x\cdot\pi(y)\oplus g(y) \tag{3}$$

where π is any permutation on $F_2^{\frac{\ }{2}}$ and g is any Boolean function on $F_2^{\frac{\ }{2}}$.

The dual of f is then $\widetilde{f}(x,y)=y\cdot\pi^{-1}(x)\oplus g(\pi^{-1}(x))$. Notice that the degree of f can be $n/2$, i.e. be optimal.

In [3] is introduced a generalization leading to balanced and resilient functions: let m and $n=r+s$ be any integers such that $r>m\geq 0$, $s>0$, g any Boolean function on F_2^s and ϕ a mapping from F_2^s to F_2^r such that every element in $\phi(F_2^s)$ has Hamming weight strictly greater than m, then the function:

$$f(x,y) = x\cdot\phi(y)\oplus g(y),\ x\in F_2^r,\ y\in F_2^s \tag{4}$$

is m-resilient, since we have

$$\widehat{\chi_f}(a,b) = 2^r \sum_{y\in\phi^{-1}(a)} (-1)^{g(y)\oplus b\cdot y}. \tag{5}$$

The degree of f (which is upper bounded by $s+1$) and its nonlinearity have been studied in [12, 13]. The functions of the form (4), for $\frac{n}{2}-1 < m+1$, can have high nonlinearities. However, optimality or sub-optimality with respect to Siegenthaler's and Sarkar et al's bounds could be obtained with this construction only with functions in which r was large and s was small. These functions having then low degrees, they are not suitable for practical use. In the case $\frac{n}{2}-1\geq m+1$, no function belonging to Maiorana-McFarland's class and having nearly optimal nonlinearity could be constructed, except in the limit case $\frac{n}{2}-1 = m+1$.

It has been shown in [22] that, under an assumption on ϕ which seems highly probable, unless r is much greater than s (case that we must exclude for degree reasons), every highly nonlinear function (4) satisfies $AI(f)\leq s$. This has been also checked by computer experiment. Table 1, from [22], gives the AI of some resilient functions built by the Maiorana-McFarland. The notation 'Const' is for the type of construction used: the classical construction of [3] is denoted by 'a', the case where exactly two elements of F_2^s have the same image of weight at least w but with different values for the function g is denoted by 'b'.

Generalizations of Maiorana-McFarland's functions have been studied in [12, 16]. They have the respective forms $f(x,y) = \bigoplus_{i=1}^{\lfloor r/2\rfloor} x_{2i-1}x_{2i}\,\psi_i(y)\oplus x\cdot\phi(y)\oplus$

Table 1. Computation of some characteristics for Boolean functions built by the Maiorana-McFarland construction

n	r	s	degree	Const.	w	resiliency	nonlinearity	alg. immunity
8	4	4	5	b	2	2	112	3
9	5	4	5	b	3	3	224	3
9	5	4	5	a	3	2	240	4
10	5	5	6	b	3	3	480	4
10	6	4	5	a	4	3	480	4
11	6	5	6	b	4	4	960	4
11	6	5	6	a	3	2	992	5
12	6	6	7	b	4	4	$2^{11}-2^6$	5
12	7	5	6	a	4	3	$2^{11}-2^6$	5
13	7	6	7	a	4	3	$2^{11}-2^6$	5
13	7	6	7	b	4	4	$2^{12}-2^7$	5
13	8	5	6	a	5	4	$2^{12}-2^7$	5
14	7	7	8	b	4	4	$2^{13}-2^7$	5
14	8	6	7	b	6	6	$2^{13}-2^8$	5
14	8	6	7	a	5	4	$2^{13}-2^7$	5
14	8	6	7	a	5	4	$2^{13}-2^7$	5
14	9	5	6	a	7	6	$2^{13}-2^8$	5

$g(y)$ and $f(x,y)=\prod_{i=1}^{\varphi(y)}(x\cdot\phi_i(y)\oplus g_i(y)\oplus 1)\oplus x\cdot\phi(y)\oplus g(y)$. The first one has more or less the same behavior with respect to resiliency and nonlinearity as the original construction, but allows achieving some particular tradeoffs that seemed impossible to achieve before. Its degree is upper bounded by $s+2$, and, *under the same reasonable assumption on ψ as the one evoked above for ϕ, we have $AI(f)\leq s+1$. The degree and the algebraic immunity of the second form have potential better behavior. Further work on this subject will have to be made in the future.* Modifications have also been proposed (see e.g. [52], in which some affine functions, at least one, are replaced by suitably chosen nonlinear functions) but it is shown in [48] that the algebraic immunities of these functions are often low.

Effective Partial-Spreads Constructions. In [31] is also introduced the class of bent functions called $\mathcal{PS}_{ap}$ (a subclass of the so-called Partial-Spreads class), whose elements are defined the following way:
$F_2^{\frac{}{2}}$ is identified to the Galois field $F_{2^{\frac{}{2}}}$ and F_2^n is identified to $F_{2^{\frac{}{2}}}\times F_{2^{\frac{}{2}}}$; $\mathcal{PS}_{ap}$ (or more precisely an extended version of the original one given by Dillon) is the set of all the functions of the form $f(x,y)=g(x\,y^{2^{\frac{}{2}}-2})$ (i.e. $g(\frac{x}{y})$ with $\frac{x}{y}=0$ if $x=0$ or $y=0$) where g is a balanced Boolean function on $F_2^{\frac{}{2}}$. We have then $\widetilde{f}(x,y)=g(\frac{y}{x})$. The degree of f is optimal, even if g is affine (see e.g. [21]).

An alternative representation of these functions is as follows. F_{2} equals $F_{2\ /2}+\omega F_{2\ /2}$, where $\omega\in F_{2}\ \setminus F_{2\ /2}$. A function f belongs to $\mathcal{PS}_{ap}$ if and only if it has weight $2^{n-1}\pm 2^{n/2-1}$ and satisfies $f(\beta x)=f(x)$, for every $\beta\in F^*_{2\ /2}$. This last condition is equivalent to $f(\alpha^{2\ /2+1}x)=f(x)$, where α is a primitive element of F_{2} . Indeed, $\alpha^{2\ /2+1}$ is a primitive element of $F_{2\ /2}$.

It is proved in [35] that, almost surely, any function in this class satisfies $AI(f)=d^\circ(f)=n/2$.

The idea of this construction is used in [9] to obtain a construction of correlation-immune functions:
Let s and r be two positive integers and $n=r+s$, g a function from F_{2} to F_2, ϕ a linear mapping from F_2^s to F_{2} and u an element of F_{2} such that $u+\phi(y)\neq 0$, $\forall y\in F_2^s$. Let f be the function from $F_{2}\ \times F_2^s\sim F_2^n$ to F_2 defined by:

$$f(x,y)=g\left(\frac{x}{u+\phi(y)}\right)\oplus v\cdot y, \qquad (6)$$

where $v\in F_2^s$. If, for every z in F_{2} , $\phi^*(z)\oplus v$ has weight greater than m, where $\phi^*: F_{2}\ \mapsto F_2^s$ is the adjoint of ϕ, then f is m-resilient.

The same observations as for Maiorana-McFarland's construction on the ability of these functions to have nonlinearities near Sarkar-Maitra's bound can be made. This construction generates a small number of functions (compared to the Mariorana-McFarland construction). But it may be able to reach better algebraic immunities and *it should be studied further for this reason.*

Functions with Few Terms in Their Trace Representation. The so-called Gold function $tr\left(\alpha x^{2\ +1}\right)$ ($r\in\mathbb{N}$, n even) is bent if and only if $\alpha\notin\{x^{2\ +1};\, x\in$

$F_{2\ \ }$ }. The Dillon function $tr\left(\alpha x^{2\ \ /2-1}\right)$ is bent if and only if the Kloosterman sum $\sum_{x\in F_{2\ \ /2}}(-1)^{tr\ \ /2(1/x+\alpha x)}$ is null, where $tr_{n/2}$ is the trace function on $F_{2\ \ /2}$, see [30]. Recent results prove the bentness of other functions with few terms in their trace representation. Namely, the functions:

- $tr\left(\alpha x^{4\ \ -2\ \ +1}\right)$, where (k,n)=1, n is not divisible by 3 and $\alpha \notin \{x^3;\, x \in F_{2\ \ }\}$, cf. [32];
- $tr\left(\alpha x^{2\ \ /2+2\ \ /4+1+1}\right)$, where $n \equiv 4$ [mod 8], $\alpha = \beta^5$, $\beta^4+\beta+1=0$, cf. [44];
- $tr\left(\alpha_1 x^{2\ \ -1+2\ \ /2-1}+\alpha_2 x^{3(2\ \ /2-1)+1}\right)$, where $n \equiv 4$ [mod 8], $\left(\alpha_1+{\alpha_1}^{2\ \ /2}\right)^2$ $= {\alpha_2}^{2\ \ /2+1}$ and $\alpha_2 \in \{x^5;\, x \in F_{2\ \ }^{*}\}$, cf. [34];
- $tr\left(\alpha_1 x^{2\ \ -1+2\ \ /2-1}+\alpha_2 x^{3(2\ \ /2-1)+1}\right)$, where $\left(\alpha_1+{\alpha_1}^{2\ \ /2}\right)^2 = {\alpha_2}^{2\ \ /2+1}$, $\alpha_2 \in F_{2\ \ }^{*}$ and $n \equiv 0$ or 2 or 6 [mod 8], cf. [34];
- $tr\left(\alpha_1 x^{2\ \ -1+2\ \ /2-1}+\alpha_2 x^{(2\ \ /2-1+2\ \ /2-2+1)(2\ \ /2-1)+1}\right)$, where $n \equiv 2$ [mod 4], $\left(\alpha_1+{\alpha_1}^{2\ \ /2}\right)^2 = {\alpha_2}^{2\ \ /2+1}$, cf. [34];
- $tr\left(\alpha_1 x^{2\ \ -1+2\ \ /2-1}+\alpha_2 x^{\frac{2\ \ /2-1+1}{3}(2\ \ /2-1)+1}\right)$, where n is divisible by 4 and $\left(\alpha_1+{\alpha_1}^{2\ \ /2}\right)^2 = {\alpha_2}^{2\ \ /2+1}$, cf. [34].

A last function, with more terms in its trace representation, and that we do not recall, is given in [43].

Computer experiment has been reported in [22] giving highly nonlinear balanced functions having high AI's. In Table 2, is computed the algebraic immunity of the function $tr(x^{2\ \ -2})$ (recall that the inverse function $x^{2\ \ -2}$ is used as S-box in the AES) for $7 \le n \le 14$. This table shows that this fonction, even if good, is not optimal.

In Table 3 are listed balanced functions of three types: (1) balanced functions equal to the traces of power functions; (2) functions, denoted by *, obtained

Table 2. Computation of the nonlinearity and algebraic immunity for the inverse function for $6 \le n \le 14$

n	d	weight	degree	nonlinearity	alg. immunity
6	-1	32	5	24	3
7	-1	64	6	54	4
8	-1	128	7	112	4
9	-1	256	8	234	4
10	-1	512	9	480	5
11	-1	1024	10	980	5
12	-1	2048	11	1984	5
13	-1	4096	12	4006	6
14	-1	8192	13	8064	6

Table 3. Computation of the nonlinearity, algebraic degree and algebraic immunity for certain power functions $tr(x^d)$

n	d	weight	degree	nonlinearity	alg. immunity
8	31	128	5	112	4
8	39 (Kasami)	128^*	6	114	4
9	57 (Kasami)	256	4	224	4
9	59	256	5	240	5
9	115	256	5	240	5
10	241 (Kasami)	512	5	480	5
10	362	512	5	480	5
10	31 (Dillon)	512^*	9	486	5
10	339 (Dobbertin)	512^*	9	480	5
11	315	1024	6	992	6
12	993 (Kasami)	2048^*	11	2000	6
12	63 (Dillon)	2048^*	11	2000	6
12	636	2048^*	11	2000	6
13	993 (Kasami)	4096	6	4032	6
13	939	4096^{**}	12	4030	7
14	4033 (Kasami)	8192	7	8064	7
14	127 (Dillon)	8192^{**}	13	8088	7

from traces of power functions, which are not balanced (they have weight $2^{n-1} - 2^{n/2-1}$) and which are made balanced by replacing the first $2^{n/2-1}$ 0's by 1's (usually this construction leads to a function with a higher algebraic degree than the starting function); (3) functions, denoted by **, of the same kind as the previous ones, but for which were additionally inverted a small number of bits from 0 to 1 and reciprocally from 1 to 0 (this small modification does not affect too much the nonlinearity but may increase the AI by 1 in the case when the dimension of the annihilator of the Boolean function f or $1 + f$ is small).

3.2 Secondary Constructions

We shall call constructions with extension of the number of variables those constructions using functions on F_2^m, with $m < n$, to obtain functions on F_2^n.

General Constructions with Extension of the Number of Variables. All known secondary constructions of bent functions are particular cases of a general construction given in [8]:
Let m and r be two positive even integers. Let f be a Boolean function on F_2^{m+r} such that, for any element x' of F_2^r, the function on F_2^m:

$$f_{x'} : x \rightarrow f(x, x')$$

is bent. Then f is bent if and only if for any element u of F_2^m, the function

$$\varphi_u : x' \rightarrow \widetilde{f_{x'}}(u)$$

is bent on F_2^r.

A particular case of the general construction of bent functions given above is a construction due to Rothaus in [53]. We describe it because it will be related to the construction studied at the end of the present paper: if f_1, f_2, f_3 and $f_1 \oplus f_2 \oplus f_3$ are bent on F_2^m (m even), then the function defined on any element (x_1, x_2, x) of F_2^{m+2} by:

$$f(x_1, x_2, x) =$$

$$f_1(x)f_2(x) \oplus f_1(x)f_3(x) \oplus f_2(x)f_3(x) \oplus [f_1(x) \oplus f_2(x)]x_1 \oplus [f_1(x) \oplus f_3(x)]x_2 \oplus x_1x_2$$

is bent.

The classical secondary constructions of resilient functions are the following:

Direct Sums of Functions: if f is an r-variable t-resilient function and if g is an s-variable m-resilient function, then the function:

$$h(x_1, \ldots, x_r, x_{r+1}, \ldots, x_{r+s}) = f(x_1, \ldots, x_r) \oplus g(x_{r+1}, \ldots, x_{r+s})$$

is $(t + m + 1)$-resilient. This comes from the easily provable relation $\widehat{\chi_h}(a, b) = \widehat{\chi_f}(a) \times \widehat{\chi_g}(b)$, $a \in F_2^r$, $b \in F_2^s$. We have also $d^\circ h = \max(d^\circ f, d^\circ g)$ and, thanks to Relation (2), $N_h = 2^{r+s-1} - \frac{1}{2}(2^r - 2N_f)(2^s - 2N_g) = 2^r N_g + 2^s N_f - 2N_f N_g$.

We clearly have $\max(AI(f), AI(g)) \leq AI(h) \leq AI(f) + AI(g)$, since the restriction to an affine subspace E of the annihilator of a function is the annihilator of the restriction to E of the function (note that in the present case, at least one restriction is actually nonzero if the annihilator is nonzero), and since every product of an annihilator of $f + \epsilon$ with an annihilator of $g + \eta$ ($\epsilon, \eta \in F_2$) is an annihilator of $h + \epsilon + \eta$ (and, here, the direct product of a nonzero r-variable annihilator of f with a nonzero s-variable annihilator of g is nonzero since the two annihilators depend on disjoint sets of variables). The question seems open of determining general conditions under which the inequality $AI(h) \leq AI(f) + AI(g)$ can be proved to be an equality (which is clearly false in some cases, e.g. when $AI(f) + AI(g) > \max(d^\circ(f), d^\circ(g))$).

Note that, when the sum is not direct, the inequality $AI(f \oplus g) \leq AI(f) + AI(g)$ can be false: let h be an n-variable Boolean function and let l be an n-variable nonzero linear function, then the functions $f = hl$ and $g = h(l \oplus 1)$ have algebraic immunities at most 1, since $f(l \oplus 1) = gl = 0$, and their sum equals h. If $AI(h) > 2$, we obtain a counter-example. However, it involves non-balanced functions. A counter-example with balanced functions is as follows: let h be an n-variable balanced Boolean function and let l and l' be two distinct n-variable nonzero linear functions, such that the functions hll', $hl(l' \oplus 1)$, $h(l \oplus 1)l'$ and $h(l \oplus 1)(l' \oplus 1)$ are balanced. Then the functions $f = hll' \oplus (h \oplus 1)l(l' \oplus 1) + (l \oplus 1)(l' \oplus 1)$ and $g = l(l' \oplus 1) + h(l \oplus 1)l' + (h \oplus 1)(l \oplus 1)(l' \oplus 1)$ have algebraic immunities at most 2, since $f(l \oplus 1)l' = gll' = 0$, they are balanced and their sum equals h. If $AI(h) > 4$, we obtain a counter-example.

The secondary construction recently introduced in [29] consists in the direct sum of the starting function f and of a function g_k on $2k$ variables.

Siegenthaler's Construction: Let f and g be two Boolean functions on F_2^r. Consider the function

$$h(x_1, \ldots, x_r, x_{r+1}) = (x_{r+1} \oplus 1)f(x_1, \ldots, x_r) \oplus x_{r+1}g(x_1, \ldots, x_r)$$

on F_2^{r+1}. Then:

$$\widehat{\chi_h}(a_1,\dots,a_r,a_{r+1}) = \widehat{\chi_f}(a_1,\dots,a_r) + (-1)^{a\ \ +1}\,\widehat{\chi_g}(a_1,\dots,a_r).$$

Thus, if f and g are m-resilient, then h is m-resilient; moreover, if for every $a \in F_2^r$ of Hamming weight $m+1$, we have $\widehat{\chi_f}(a)+\widehat{\chi_g}(a)=0$, then h is $(m+1)$-resilient. And we have: $N_h \geq N_f + N_g$. If f and g achieve maximum possible nonlinearity $2^{r-1}-2^{m+1}$ and if h is $(m+1)$-resilient, then the nonlinearity $2^r - 2^{m+2}$ of h is the best possible. If the supports of the Walsh transforms of f and g are disjoint, then we have $N_h = 2^{r-1} + \min(N_f, N_g)$; thus, if f and g achieve maximum possible nonlinearity $2^{r-1}-2^{m+1}$, then h achieves best possible nonlinearity $2^r - 2^{m+1}$.

The algebraic immunity of h has been studied in [29]:

- If $AI(f) \neq AI(g)$ then $AI(h) = \min\{AI(f), AI(g)\}+1$.
- If $AI(f) = AI(g) = d$, then $d \leq AI(h) \leq d+1$, and $AI(h) = d$ if and only if there exist $f_1, g_1 \in B_n$ of algebraic degree d such that $\{f * f_1 = 0, g * g_1 = 0\}$ or $\{(1+f)*f_1 = 0, (1+g)*g_1 = 0\}$ and $\deg(f_1+g_1) \leq d-1$.

We cannot say that Siegenthaler's construction is good or is bad in terms of algebraic immunity, since:

- a good construction is supposed to gain 1 (resp k) for the algebraic immunity when we add 2 (resp $2k$) variables, here we add only one;
- the construction is very general since every function can be obtained from it.

In practice, we could not obtain good algebraic immunity with it.

Siegenthaler [57] proposed, as a particular case of its (iterated) construction, to add to a given function f a linear function on disjoint variables for increasing its resiliency order. This does not allow achieving good algebraic immunity, since adding a linear function to f can increase the AI at most by one (an annihilator of f, multiplied by $l+1$ gives an annihilator of $f+l$).

Tarannikov's Construction: Let g be any Boolean function on F_2^r. Define the Boolean function h on F_2^{r+1} by

$$h(x_1,\dots,x_r,x_{r+1}) = x_{r+1} \oplus g(x_1,\dots,x_{r-1},x_r \oplus x_{r+1}).$$

The Walsh transform $\widehat{\chi_h}(a_1,\dots,a_{r+1})$ is equal to

$$\sum_{x_1,\dots,x\ \ +1\in F_2} (-1)^{a\cdot x\oplus g(x_1,\dots,x\ \ )\oplus a\ \ x\ \ \oplus(a\ \ \oplus a\ \ +1\oplus 1)x\ \ +1}$$

where we write $a=(a_1,\dots,a_{r-1})$ and $x=(x_1,\dots,x_{r-1})$; it is null if $a_{r+1}=a_r$ and it equals $2\,\widehat{\chi_g}(a_1,\dots,a_{r-1},a_r)$ if $a_r = a_{r+1}\oplus 1$. Thus: $N_h = 2\,N_g$; If g is m-resilient, then h is m-resilient. If, additionally, $\widehat{\chi_g}(a_1,\dots,a_{r-1},1)$ is null for every vector $(a_1,\dots,a_{r-1})$ of weight at most m, then h is $(m+1)$-resilient.

Generalizations: Tarannikov in [60], and after him, Pasalic et al. in [54] used this construction to design a more complex one, that we call *Tarannikov et al.'s construction*, and which allowed maximum tradeoff between resiliency, algebraic degree and nonlinearity. This construction uses two $(n-1)$-variable m-resilient functions f_1 and f_2 achieving Siegenthaler's and Sarkar et al.'s bounds to design an $(n+3)$-variable $(m+2)$-resilient function h also achieving these bounds, assuming that $f_1 + f_2$ has same degree as f_1 and f_2 and that the supports of the Walsh transforms of f_1 and f_2 are disjoint. The two restrictions $h_1(x_1, \ldots, x_{n+2}) = h(x_1, \ldots, x_{n+2}, 0)$ and $h_2(x_1, \ldots, x_{n+2}) = h(x_1, \ldots, x_{n+2}, 1)$ have then also disjoint Walsh supports, and these two functions can then be used in the places of f_1 and f_2. This leads to an infinite class of functions achieving Sarkar et al.'s and Siegenthaler's bounds. It has been proved in [2] that the n-variable functions constructed by this method attain $\boldsymbol{\Omega}(\sqrt{n})$ algebraic immunity (which is unfortunately bad).

Tarannikov et al.'s construction has been in its turn generalized (see [14]):

Theorem 1. *Let r, s, t and m be positive integers such that $t < r$ and $m < s$. Let f_1 and f_2 be two r-variable t-resilient functions. Let g_1 and g_2 be two s-variable m-resilient functions. Then the function $h(x,y) = f_1(x) \oplus g_1(y) \oplus (f_1 \oplus f_2)(x)\,(g_1 \oplus g_2)(y)$, $x \in F_2^r, y \in F_2^s$ is an $(r+s)$-variable $(t+m+1)$-resilient function. If f_1 and f_2 are distinct and if g_1 and g_2 are distinct, then the algebraic degree of h equals $\max(d^\circ f_1, d^\circ g_1, d^\circ(f_1 \oplus f_2) + d^\circ(g_1 \oplus g_2))$; otherwise, it equals $\max(d^\circ f_1, d^\circ g_1)$. The Walsh transform of h takes value*

$$\widehat{\chi_h}(a,b) = \frac{1}{2}\widehat{\chi_{f_1}}(a)\left[\widehat{\chi_{g_1}}(b) + \widehat{\chi_{g_2}}(b)\right] + \frac{1}{2}\widehat{\chi_{f_2}}(a)\left[\widehat{\chi_{g_1}}(b) - \widehat{\chi_{g_2}}(b)\right]. \tag{7}$$

If the Walsh transforms of f_1 and f_2 have disjoint supports as well as those of g_1 and g_2, then

$$N_h = \min_{i,j \in \{1,2\}} \left(2^{r+s-2} + 2^{r-1} N_g \; + 2^{s-1} N_f \; - N_f \, N_g \;\right). \tag{8}$$

In particular, if f_1 and f_2 have (optimum) nonlinearity $2^{r-1} - 2^{t+1}$ and have disjoint Walsh supports, if g_1 and g_2 have (optimum) nonlinearity $2^{s-1} - 2^{m+1}$ and have disjoint Walsh supports, if $f_1 + f_2$ has degree $r - t - 1$ and if $g_1 + g_2$ has degree $s - m - 1$, then h has degree $r + s - t - m - 2$ and nonlinearity $2^{r+s-1} - 2^{t+m+2}$, and thus achieves Siegenthaler's and Sarkar et al.'s bounds.

Note that function h, defined this way, is the concatenation of the four functions f_1, $f_1 \oplus 1$, f_2 and $f_2 \oplus 1$, in an order controled by $g_1(y)$ and $g_2(y)$. The proof of this theorem and examples of such pairs (f_1, f_2) (or (g_1, g_2)) can be found in [14]. This construction being very general since it generalizes all known secondary constructions, it is difficult to give bounds on the algebraic immunity of the resulting functions.

Other Constructions: There exists a secondary construction of resilient functions from bent functions (see [9]): let r be a positive integer, m a positive even integer and f a function such that, for any element x', the function: $f_{x'} : x \to f(x, x')$

is bent. If, for every element u of Hamming weight at most t, the function $\varphi_u : x' \rightarrow \widetilde{f_{x'}}(u)$ is $(t - w_H(u))$-resilient, then f is t-resilient (the converse is true).

Rothaus' construction has been modified in [9] into a construction of resilient functions: if f_1 is t-resilient, f_2 and f_3 are $(t-1)$-resilient and $f_1 \oplus f_2 \oplus f_3$ is $(t-2)$-resilient, then $f(x_1, x_2, x)$ is t-resilient (the converse is true). This construction does not seem able to produce functions with higher algebraic immunities than the functions used as building blocks.

Constructions Without Extension of the Number of Variables. Such constructions, by modifying the support of highly nonlinear resilient functions without decreasing their characteristics, may be appropriate for trying to increase the algebraic immunities of such functions, previously obtained by classical constructions. There exist, in the literature, four such constructions.

Modifying a Function on a Subspace: Dillon proves in [31] that if a binary function f is bent on F_2^n (n even) and if E is an $\frac{n}{2}$-dimensional flat on which f is constant, then, denoting by 1_E the indicator (i.e. the characteristic function) of E, the function $f \oplus 1_E$ is bent too. This is generalized in [6]:

Let $E = b \oplus E'$ be any flat in F_2^n (E', the direction of E, is a linear subspace of F_2^n). Let f be any bent function on F_2^n. The function $f^\star = f \oplus 1_E$ is bent if and only if one of the following equivalent conditions is satisfied :

1. for any x in $F_2^n \setminus E'$, the function: $y \mapsto f(y) \oplus f(x \oplus y)$ is balanced on E;
2. for any a in F_2^n, the restriction of the function $\widetilde{f}(x) \oplus b \cdot x$ to the flat $a \oplus E'^\perp$ is either constant or balanced.

If $f^\star$ is bent, then E has dimension greater than or equal to $r = n/2$ and the degree of the restriction of f to E is at most $dim(E) - r + 1$. If E has dimension r, then this last condition (i.e., the fact that the restriction of f to E is affine) is also sufficient and the function $\widetilde{f^\star}(x)$ is equal to :

$$\widetilde{f}(x) \oplus 1_{E'^\perp}(u \oplus x),$$

where u is any element of F_2^n such that for any x in $E : f(x) = u \cdot x \oplus \epsilon$.

This construction has been adapted to correlation-immune functions in [9]: let t, m and n any positive integers and f a t-th order correlation-immune function from F_2^n to F_2^m; assume there exists a subspace E of F_2^n, whose minimum nonzero weight is greater than t and such that the restriction of f to the orthogonal of E (i.e. the subspace of F_2^n: $E^\perp = \{u \in F_2^n \,|\, \forall x \in E,\, u \cdot x = 1\}$) is constant. Then f remains t-th order correlation-immune if we change its constant value on $E^\perp$ into any other one.

Hou-Langevin Construction: X.-D. Hou and P. Langevin have made in [39] a very simple observation: Let f be a Boolean function on F_2^n, n even. Let $\sigma = (\sigma_1, \cdots, \sigma_n)$ be a permutation on F_2^n such that

$$d_H\left(f, \sum_{i=1}^{n} a_i\, \sigma_i\right) = 2^{n-1} \pm 2^{\frac{}{2}-1};\ \forall a \in F_2^n.$$

Then $f \circ \sigma^{-1}$ is bent.

A case of application of this fact, pointed out in [37], is when f belongs to Maiorana-McFarland class (3), with $\pi = id$ and when the coordinate functions of σ are all of the form $x_{i_1} y_{j_1} \oplus \ldots \oplus x_{i}\ \ y_{j}\ \ \oplus l(x,y) \oplus h(y)$, where $k < n/2$ and $i_l < j_l$ for every $l \leq k$; the function h is any Boolean function on $F_2^{n/2}$ and l is affine.

Another case of application is given in [39] when f has degree at most 3: assume that for every $i = 1, \cdots, n$, there exists a subset U_i of F_2^n and an affine function h_i such that:

$$\sigma_i(x) = \sum_{u \in U} (f(x) \oplus f(x \oplus u)) \oplus h_i(x).$$

Then $f \circ \sigma^{-1}$ is bent.

Only examples of potentially new bent functions have been deduced by Hou and Langevin from these results.

This idea of construction can be adapted to resilient functions:

If $d_H(f, \sum_{i=1}^{n} a_i\, \sigma_i) = 2^{n-1}$ for every $a \in F_2^n$ of weight at most k, then $f \circ \sigma^{-1}$ is k-resilient. This secondary construction needs strong hypothesis on the function used as buiding block to produce resilient functions. Further work seems necessary for designing functions for stream ciphers by using it.

Two Recent Constructions have been introduced in [24]. They will be recalled at Subsection 4.3.

4 A New Secondary Construction of Boolean Functions

4.1 A Modification of Rothaus' Construction

Rothaus' construction was the first non-trivial construction of bent functions to be obtained in the literature. It is still one of the most interesting known constructions nowadays, since the functions it produces can have degrees near $n/2$, even if the functions used as building blocks don't. But the constructed functions have a very particular form. It is possible to derive a construction having the same nice property but having not the same drawback, thanks to the following observation.

Given three Boolean functions f_1, f_2 and f_3, there is a nice relationship between their Walsh transforms and the Walsh transforms of two of their elementary symmetric related functions:

Lemma 1. *Let f_1, f_2 and f_3 be three Boolean functions on F_2^n. Let us denote by σ_1 the Boolean function equal to $f_1 \oplus f_2 \oplus f_3$ and by σ_2 the Boolean function equal to $f_1 f_2 \oplus f_1 f_3 \oplus f_2 f_3$. Then we have $f_1 + f_2 + f_3 = \sigma_1 + 2\sigma_2$. This implies*

$$\widehat{\chi_{f_1}} + \widehat{\chi_{f_2}} + \widehat{\chi_{f_3}} = \widehat{\chi_{\sigma_1}} + 2\widehat{\chi_{\sigma_2}}. \qquad (9)$$

Proof. The fact that $f_1+f_2+f_3=\sigma_1+2\sigma_2$ (recall that these sums are calculated in $\mathbb{Z}$ and not mod 2) can be checked easily and directly implies $\chi_{f_1}+\chi_{f_2}+\chi_{f_3}=\chi_{\sigma_1}+2\chi_{\sigma_2}$, thanks to the equality $\chi_f=1-2f$ (valid for every Boolean function). The linearity of the Fourier transform with respect to the addition in $\mathbb{Z}$ implies then Relation (9). ◇

4.2 Deduced Constructions of Resilient Functions

We begin with resilient functions because the application of Lemma 1 is easy in this case. In the following theorem, saying that a function f is 0-order correlation immune does not impose any condition on f and saying it is 0-resilient means it is balanced.

Theorem 2. *Let n be any positive integer and k any non-negative integer such that $k \leq n$. Let f_1, f_2 and f_3 be three k-th order correlation immune (resp. k-resilient) functions. Then the function $\sigma_1 = f_1 \oplus f_2 \oplus f_3$ is k-th order correlation immune (resp. k-resilient) if and only if the function $\sigma_2 = f_1 f_2 \oplus f_1 f_3 \oplus f_2 f_3$ is k-th order correlation immune (resp. k-resilient). Moreover:*

$$N_{\sigma_2} \geq \frac{1}{2}\left(N_{\sigma_1} + \sum_{i=1}^{3} N_f\right) - 2^{n-1} \tag{10}$$

and if the Walsh supports of f_1, f_2 and f_3 are pairwise disjoint (that is, if at most one value $\widehat{\chi_f}\,(s)$, $i=1,2,3$ is nonzero, for every vector s), then

$$N_{\sigma_2} \geq \frac{1}{2}\left(N_{\sigma_1} + \min_{1\leq i\leq 3} N_f\right). \tag{11}$$

Proof. Relation (9) and the fact that for every nonzero vector a of weight at most k we have $\widehat{\chi_f}\,(a)=0$ for $i=1,2,$ 3 imply that $\widehat{\chi_{\sigma_1}}(a)=0$ if and only if $\widehat{\chi_{\sigma_2}}(a)=0$. Same property occurs for $a=0$ when f_1, f_2 and f_3 are resilient. Relation (9) implies the relation

$$\max_{s\in F_2}|\widehat{\chi_{\sigma_2}}(s)| \leq \frac{1}{2}\left(\sum_{i=1}^{3}\left(\max_{s\in F_2}|\widehat{\chi_f}\,(s)|\right) + \max_{s\in F_2}|\widehat{\chi_{\sigma_1}}(s)|\right)$$

and Relation (2) implies then Relation (10). If the Walsh supports of f_1, f_2 and f_3 are pairwise disjoint, then Relation (9) implies the relation

$$\max_{s\in F_2}|\widehat{\chi_{\sigma_2}}(s)| \leq \frac{1}{2}\left(\max_{1\leq i\leq 3}\left(\max_{s\in F_2}|\widehat{\chi_f}\,(s)|\right) + \max_{s\in F_2}|\widehat{\chi_{\sigma_1}}(s)|\right)$$

and Relation (2) implies then Relation (11). ◇

Remark: We have $\sigma_2 = f_1 \oplus (f_1 \oplus f_2)(f_1 \oplus f_3)$. Hence, another possible statement of Theorem 2 is: if f_1, $f_1 \oplus f_2$ and $f_1 \oplus f_3$ are k-th order correlation immune (resp. k-resilient) functions, then the function $f_1 \oplus f_2 \oplus f_3$ is k-th order correlation immune (resp. k-resilient) if and only if the function $f_1 \oplus f_2 f_3$ is k-th order correlation immune (resp. k-resilient).

We use now the invariance of the notion of correlation-immune (resp. resilient) function under translation to deduce an example of application of Theorem 2.

Proposition 2. *Let n be any positive integer and k any non-negative integer such that $k \leq n$. Let f and g be two k-th order correlation immune (resp. k-resilient) functions on F_2^n. Assume that there exist $a, b \in F_2^n$ such that $D_a f \oplus D_b g$ is constant. Then the function $h(x) = f(x) \oplus D_a f(x)(f(x) \oplus g(x))$, that is, $h(x) = \begin{cases} f(x) \text{ if } D_a f(x) = 0 \\ g(x) \text{ if } D_a f(x) = 1 \end{cases}$ is k-th order correlation immune (resp. k-resilient). Moreover:*

$$N_h \geq N_f + N_g - 2^{n-1} \tag{12}$$

and if the Walsh support of f is disjoint of that of g, then

$$N_h \geq \min\left(N_f, N_g\right). \tag{13}$$

Note that finding hihgly nonlinear resilient functions with disjoint supports is easy, by using Tarannikov et al.'s construction.

Proof. Let $D_a f \oplus D_b g = \epsilon$. Taking $f_1(x) = f(x)$, $f_2(x) = f(x+a)$ and $f_3(x) = g(x)$, the hypothesis of Theorem 2 is satisfied, since $\sigma_1(x) = D_a f(x) \oplus g(x) = D_b g(x) \oplus \epsilon \oplus g(x) = g(x+b) \oplus \epsilon$ is k-th order correlation immune (resp. k-resilient). Hence, $h(x) = f(x) \oplus D_a f(x)(f(x) \oplus g(x))$ is k-th order correlation immune (resp. k-resilient). Relation (12) is a direct consequence of Relation (10). Note that the Walsh support of f_2 equals that of $f_1 = f$, since we have $\widehat{\chi_{f_2}}(s) = (-1)^{a \cdot s} \widehat{\chi_f}(s)$ and that the Walsh support of σ_1 equals that of $f_3 = g$. Hence, if the Walsh support of f is disjoint of that of g, then Relation (9) implies the relation

$$\max_{s \in F_2} |\widehat{\chi_h}(s)| \leq \max\left(\max_{s \in F_2} |\widehat{\chi_f}(s)|, \max_{s \in F_2} |\widehat{\chi_g}(s)|\right)$$

and Relation (2) implies then Relation (13). ◇

Remarks:

1. The notion of resilient function being also invariant under any permutation of the input coordinates $x_1, \ldots, x_n$, Proposition 2 is also valid if we replace $D_a f$ by $f(x_1, \ldots, x_n) \oplus f(x_{\tau(1)}, \ldots, x_{\tau(n)})$ and $D_b g$ by $g(x_1, \ldots, x_n) \oplus g(x_{\tau'(1)}, \ldots, x_{\tau'(n)})$, where τ and τ' are two permutations of $\{1, \ldots, n\}$.
2. Computer experiment shows that the secondary construction of Theorem 2 and its particular case given in Proposition 2 can increase the algebraic immunity, while keeping the same resiliency order and the same nonlinearity. The reason is in the fact that the support of σ_2 (resp. h) is, in general, more complex than those of f_1, f_2 and f_3 (resp. f and g). *It would be nice finding a provable result illustrating this.*

A Deduced Primary Construction of Resilient Functions

Proposition 3. *let t and $n = r + s$ be any positive integers ($r > t > 0$, $s > 0$). Let g_1, g_2 and g_3 be any boolean functions on F_2^s and ϕ_1, ϕ_2 and ϕ_3 any mappings from F_2^s to F_2^r such that, for every element y in F_2^s, the vectors $\phi_1(y)$, $\phi_2(y)$, $\phi_3(y)$ and $\phi_1(y) \oplus \phi_2(y) \oplus \phi_3(y)$ have Hamming weights greater than t. Let us denote $f_1(x) = x \cdot \phi_1(y) \ \oplus \ g_1(y)$, $f_2(x) = x \cdot \phi_2(y) \ \oplus \ g_2(y)$ and $f_3(x) = x \cdot \phi_3(y) \ \oplus \ g_3(y)$. Then the function:*

$$f(x, y) = f_1(x)\, f_2(x) \ \oplus f_1(x)\, f_3(x) \oplus \ f_2(x)\, f_3(x)$$

is t-resilient.

Note that, if the sets $\phi_1(F_2^s)$, $\phi_2(F_2^s)$, and $\phi_3(F_2^s)$ are disjoint, Relation (5) implies that the Walsh supports of f_1, f_2 and f_3 are disjoint. Relation (11) of Theorem 2 is then satisfied. This implies that f can be (nearly) optimum with respect to Siegenthaler's and Sarkar et al.'s bounds. We have seen that a Maiorana-McFarland (nearly) optimum function has low degree and still lower AI. But here, the algebraic degree of f and its algebraic immunity may be higher than those of Maiorana-McFarland's (nearly) optimum functions. For instance, we obtained in [22] a balanced 14-variable function with algebraic degree 7, nonlinearity 7808, order of resiliency 5 and AI 6 by considering ϕ_1, ϕ_2, ϕ_3 from F_2^6 to F_2^8 such that for any $i \in \{1, 2, 3\}$ and any $x \in F_2^6$: $w_H(\phi_i(x)) \geq 6$ and such that $w_H(\phi_1(x) + \phi_2(x) + \phi_3(x)) \geq 6$.

Remark: We can also apply Theorem 2 to the class of resilient functions derived from the $\mathcal{PS}_{ap}$ construction: Let n and m be two positive integers, g_1, g_2 and g_3 three functions from F_2 to F_2, ϕ a linear mapping from F_2^n to F_2 and a an element of F_2 such that $a \oplus \phi(y) \neq 0$, $\forall y \in F_2^n$.

Let b_1, b_2 and $b_3 \in F_2^n$ such that, for every z in F_2 , $\phi^*(z) \oplus b_i$, $i = 1, 2, 3$ and $\phi^*(z) \oplus b_1 \oplus b_2 \oplus b_3$ have weight greater than t, where ϕ^* is the adjoint of ϕ, then the function

$$\begin{gathered} f(x, y) = \\ \left(g_1\left(\frac{x}{a \oplus \phi(y)}\right) \oplus b_1 \cdot y\right)\left(g_2\left(\frac{x}{a \oplus \phi(y)}\right) \oplus b_2 \cdot y\right) \oplus \\ \left(g_1\left(\frac{x}{a \oplus \phi(y)}\right) \oplus b_1 \cdot y\right)\left(g_3\left(\frac{x}{a \oplus \phi(y)}\right) \oplus b_3 \cdot y\right) \oplus \\ \left(g_2\left(\frac{x}{a \oplus \phi(y)}\right) \oplus b_2 \cdot y\right)\left(g_3\left(\frac{x}{a \oplus \phi(y)}\right) \oplus b_3 \cdot y\right) \end{gathered}$$

is t-resilient. The complexity of the support of this function may permit getting a good algebraic immunity.

4.3 Constructing Bent Functions by Using Lemma 1

Applying Lemma 1 to the construction of bent functions is slightly less simple than for resilient functions. Nevertheless, we will deduce here again a secondary construction (we shall see that it generalizes a secondary construction obtained recently) and a primary construction.

Theorem 3. *Let n be any positive even integer. Let f_1, f_2 and f_3 be three bent functions. Denote by σ_1 the function $f_1 \oplus f_2 \oplus f_3$ and by σ_2 the function $f_1 f_2 \oplus f_1 f_3 \oplus f_2 f_3$. Then:*
1. if σ_1 is bent and if $\widetilde{\sigma_1} = \widetilde{f_1} \oplus \widetilde{f_2} \oplus \widetilde{f_3}$, then σ_2 is bent and $\widetilde{\sigma_2} = \widetilde{f_1}\widetilde{f_2} \oplus \widetilde{f_1}\widetilde{f_3} \oplus \widetilde{f_2}\widetilde{f_3}$;
2. if σ_2 is bent, or if more generally $\widehat{\chi_{\sigma_2}}(a)$ is divisible by $2^{n/2}$ for every a (e.g. if σ_2 is plateaued), then σ_1 is bent.

Proof. By hypothesis, we have for $i = 1, 2$, 3 and for every vector a: $\widehat{\chi_f}\,(a) = (-1)^{\widetilde{f}\,(a)}\, 2^{n/2}$.
1. If σ_1 is bent and if $\widetilde{\sigma_1} = \widetilde{f_1} \oplus \widetilde{f_2} \oplus \widetilde{f_3}$, then we have:

$$\widehat{\chi_{\sigma_1}}(a) = (-1)^{\widetilde{f_1}(a) \oplus \widetilde{f_2}(a) \oplus \widetilde{f_3}(a)}\, 2^{n/2}.$$

Relation (9) implies:

$$\begin{aligned}\widehat{\chi_{\sigma_2}}(a) &= \left[(-1)^{\widetilde{f_1}(a)} + (-1)^{\widetilde{f_2}(a)} + (-1)^{\widetilde{f_3}(a)} - (-1)^{\widetilde{f_1}(a) \oplus \widetilde{f_2}(a) \oplus \widetilde{f_3}(a)}\right] 2^{(n-2)/2} \\ &= (-1)^{\widetilde{f_1}(a)\widetilde{f_2}(a) \oplus \widetilde{f_1}(a)\widetilde{f_3}(a) \oplus \widetilde{f_2}(a)\widetilde{f_3}(a)} 2^{n/2}.\end{aligned}$$

2. If $\widehat{\chi_{\sigma_2}}(a)$ is divisible by $2^{n/2}$ for every a, then the number $\widehat{\chi_{\sigma_1}}(a)$, equal to $\left[(-1)^{\widetilde{f_1}(a)} + (-1)^{\widetilde{f_2}(a)} + (-1)^{\widetilde{f_3}(a)}\right] 2^{n/2} - 2\widehat{\chi_{\sigma_2}}(a)$, is congruent with $2^{n/2}$ mod $2^{n/2+1}$ for every a. This is sufficient to imply that σ_1 is bent, according to Lemma 1 of [7]. ◇

Remark: Here again, it is possible to state Theorem 3 slightly differently. For instance, if f_1, $f_1 \oplus f_2$ and $f_1 \oplus f_3$ are three bent functions such that $f_1 \oplus f_2 f_3$ has Walsh spectrum divisible by $2^{n/2}$, then $\sigma_1 = f_1 \oplus f_2 \oplus f_3$ is bent. Notice that a sufficient condition for $f_1 \oplus f_2 f_3$ having Walsh spectrum divisible by $2^{n/2}$ is that $f_2 f_3 = 0$ or that $f_2 \preceq f_3$ (i.e. that the support of f_3 includes that of f_2). In particular, if f is a bent function and if E and F are two disjoint $(n/2)$-dimensional flats on which f is affine, the function $f \oplus 1_E \oplus 1_F$ is bent.

Theorem 3 and Lemma 1 imply as particular cases two secondary constructions of bent functions, recently obtained in [24]:

Corollary 1. *[24] Let f and g be two bent functions on F_2^n (n even). Assume that there exists $a \in F_2^n$ such that $D_a f = D_a g$. Then the function $f(x) \oplus D_a f(x)(f(x) \oplus g(x))$ is bent and has dual $\widetilde{f}(x) \oplus (a \cdot x)(\widetilde{f}(x) \oplus \widetilde{g}(x))$.*

Indeed, taking $f_1(x) = f(x)$, $f_2(x) = f(x+a)$ and $f_3(x) = g(x)$, the hypothesis of Alinea 1 of Theorem 3 is satisfied: $\sigma_1(x) = D_a f(x) \oplus g(x) = D_a g(x) \oplus g(x) = g(x+a)$ is bent and we have $\widetilde{\sigma_1}(x) = a \cdot x \oplus \widetilde{g}(x) = \widetilde{f_1}(x) \oplus \widetilde{f_2}(x) \oplus \widetilde{f_3}(x)$. Hence, $\sigma_2(x) = f(x) \oplus D_a f(x)(f(x) \oplus g(x))$ is bent (note that the dual of f_2 equals $\widetilde{f_1} \oplus a \cdot x$). ◇

Remarks:

1. Applying Corollary 1 to the duals of f and g gives that, if f and g are two bent functions on F_2^n such that there exists $a \in F_2^n$ such that $D_a \widetilde{f} = D_a \widetilde{g}$, then the function $f(x) \oplus (a \cdot x)(f(x) \oplus g(x))$ is bent.

2. More generally than in Corollary 1, let L be an affine automorphism of F_2^n. We know that, for every bent function f, the function $f \circ L$ is bent and admits as dual $\widetilde{f} \circ L^*$, where L^* is the adjoint operator of L^{-1} (such that, for every $x, y \in F_2^n$, we have $x \cdot L^{-1}(y) = L^*(x) \cdot y$). Then if f and g are two bent functions such that $\begin{cases} f(x) \oplus f \circ L(x) \oplus g(x) \oplus g \circ L(x) = 0, \forall x \in F_2^n \\ \widetilde{f}(x) \oplus \widetilde{f} \circ L^*(x) \oplus \widetilde{g}(x) \oplus \widetilde{g} \circ L^*(x) = 0, \forall x \in F_2^n; \end{cases}$, then the function $fg \oplus (f \oplus g)(f \circ L)$ is bent and its dual equals $\widetilde{f}\widetilde{g} \oplus (\widetilde{f} \oplus \widetilde{g})(\widetilde{f} \circ L^*)$. Indeed, taking $f_1 = f$, $f_2 = f \circ L$ and $f_3 = g$, we have $\sigma_1 = g \circ L$ and therefore $\widetilde{\sigma}_1 = \widetilde{g} \circ L^* = \widetilde{f}_1 \oplus \widetilde{f}_2 \oplus \widetilde{f}_3$ and $\sigma_2 = fg \oplus (f \oplus g)(f \circ L)$.

Proposition 4. *[24] Let n be any positive even integer. Let f and g be two plateaued functions of the same amplitude $2^{n/2+1}$, whose Walsh transform's supports S_f and S_g are two distinct cosets of the same vector subspace E of F_2^n. Let a be an element of F_2^n such that the cosets $a + S_f$ and $a + S_g$ are both distinct of S_f and S_g. Then the function $f(x) \oplus (a \cdot x)(f(x) \oplus g(x))$ is bent.*

Proof. Set $f_1(x) = f(x)$, $f_2(x) = f(x) \oplus a \cdot x$ and $f_3(x) = g(x)$. We have: $\sigma_1(x) = a \cdot x \oplus g(x)$. Hence, f_1, f_2, f_3 and σ_1 are four plateaued functions of amplitude $2^{n/2+1}$, whose Walsh transform's supports equal $S_f, a + S_f, S_g$ and $a + S_g$. The cosets $S_f, S_g, a + S_f$ and $a + S_g$ constituting a partition of F_2^n (note that E has necessarily co-dimension 2), Relation (9) implies that $\sigma_2(x) = f(x) \oplus (a \cdot x)(f(x) \oplus g(x))$ is bent. ◇

An Example Related to Proposition 4: The Kerdock Code. Partially-bent functions (see [5]) give a way of constructing plateaued functions[2]; they are defined as follows: two vector subspaces E (of even dimension) and F are chosen in F_2^n such that their sum is direct and equals F_2^n; for every $x \in E$ and every $y \in F$, we define $f(x + y) = g(x) \oplus l(y)$, where g is bent on E and l is linear on F. All quadratic functions (that is, functions of algebraic degrees at most 2) are of this type (F is then the kernel of their associated symplectic form; see [45, 5]). If F (often called the kernel of f) has dimension 2, then f is plateaued with amplitude $2^{n/2+1}$ and its Walsh transform's support S_f is a flat (of co-dimension 2) whose direction equals $F^\perp$. Hence, we can choose two vectors a and b such that $\{S_f, a+S_f, b+S_f, a+b+S_f\}$ is a partition of F_2^n. We define then $f_1(x) = f(x)$, $f_2(x) = f(x) \oplus a \cdot x$, and $f_3(x) = f(x) \oplus b \cdot x$. We have $(f_1 \oplus f_2 \oplus f_3)(x) = f(x) \oplus (a+b) \cdot x$ and the hypothesis of Lemma 1 (that is, the hypothesis of Proposition 4 with $g(x) = f(x) \oplus b \cdot x$) is satisfied. We deduce that the function $f(x) \oplus (a \cdot x)(b \cdot x)$ is bent. In the sequel, we shall call *Kerdock-like construction* this construction $(f, a, b) \mapsto \sigma_2$. The fact that it always provides bent functions is not new, but this is exactly how the bent functions in the Kerdock code are constructed (see [45]). We show now how (revisiting an observation from [24]). Denoting $m = n - 1 = 2t + 1$, the elements of F_2^n are identified to ordered pairs (x, ϵ) with $x \in F_2$ and $\epsilon \in F_2$. Then we define $f(x, \epsilon) = tr(\sum_{i=1}^{t} x^{2\ +1})$, where tr is the trace function from F_2 to F_2. This function is quadratic and its kernel F (more precisely here, the kernel of its associated symplectic form)

[2] Another way is by using Maiorana-McFarland construction (4) with ϕ injective.

equals (see [45]) the vector space $\{(x,\epsilon) \in F_2 \ \ \times F_2 /\, x + tr(x) = 0\} = \{\mathbf{0},\mathbf{1}\} \times F_2$, where $\mathbf{0},\mathbf{1} \in F_2$. This kernel has dimension 2 and the Walsh transform's support S_f of f is therefore a flat of dimension $n-2$ (whose direction equals $F^{\perp}$). So we can apply Kerdock-like construction. Recall that the notion of bent function is independent of the choice of the inner product. So we can choose $(x,\epsilon)\cdot(y,\eta) = tr(xy) \oplus \epsilon\eta$. The choice of $a = (\mathbf{0},1)$, $b = (\mathbf{1},0)$ in the Kerdock-like construction shows that the function $\sigma_2(x,\epsilon) = tr(\sum_{i=1}^{t} x^{2\ +1}) \oplus \epsilon tr(x)$ is bent. Obviously, for every $u \in F_2^{*}$, the function $(x,\epsilon) \mapsto \sigma_2(ux,\epsilon)$ is also bent (note that it is obtained through the Kerdock-like construction from $f_u(x,\epsilon) = tr(\sum_{i=1}^{t}(ux)^{2\ +1})$, $a = (\mathbf{0},1)$ and $b = (u,0)$). A property which is specific to Kerdock codes (and that could not be obtained with non-quadratic functions until now) is that the sum $(x,\epsilon) \mapsto \sigma_2(ux,\epsilon) \oplus \sigma_2(vx,\epsilon)$ of two distinct such functions is still bent. Let us check this: the quadratic function $f_u \oplus f_v$ has kernel $\{(x,\epsilon) \in F_2 \ \ \times F_2 /\, (u^2+v^2)x + utr(ux) + vtr(vx) = 0\} = E_{u,v} \times F_2$, where $E_{u,v}$ has dimension at most 2 (since the equation $(u^2+v^2)x + utr(ux) + vtr(vx) = 0$ has at most 4 solutions). Since we know that the kernel of a quadratic function must have even co-dimension (and hence, here, even dimension), the dimension of $E_{u,v}$ must equal 1. The function $\sigma_2(ux,\epsilon) \oplus \sigma_2(vx,\epsilon)$ can then be obtained through the Kerdock-like construction from the function $f_u \oplus f_v$ and the vectors $a = (\mathbf{0},1)$ and $b = (u+v,0)$. The hypothesis of Proposition 4 is satisfied thanks to the fact that b does not belong to $E_{u,v}^{\perp}$ (this can be checked by showing that $E_{u,v}^{\perp} = \{(u^2+v^2)y + utr(uy) + vtr(vy);\ y \in F_2 \ \ \})$.

A Primary Construction of Bent Functions Deduced from Theorem 3

Proposition 5. *Let n be any positive even integer. Let π_1, π_2, π_3 be three permutations on $F_2^{n/2}$ such that $\pi_1 \oplus \pi_2 \oplus \pi_3$ is also a permutation and such that the inverse of $\pi_1 \oplus \pi_2 \oplus \pi_3$ equals $\pi_1^{-1} \oplus \pi_2^{-1} \oplus \pi_3^{-1}$. Then the function*

$$f(x,y) = [x\cdot\pi_1(y)]\,[x\cdot\pi_2(y)] \oplus [x\cdot\pi_1(y)]\,[x\cdot\pi_3(y)] \oplus [x\cdot\pi_2(y)]\,[x\cdot\pi_3(y)]$$

is bent.

The proof is a direct consequence of the first alinea of Theorem 3 and of the properties of Maiorana-McFarland's class recalled above. Note that the result is still valid if an affine function g in y is added to the $x\cdot\pi_i(y)$'s in the expression of $f(x,y)$.

It is also easy to apply Theorem 3 to class $\mathcal{PS}_{ap}$: the condition on the dual of σ_1 is automatically satisfied if σ_1 is bent. But this does not lead to new functions, since if $f_i(x,y) = g_i(x\, y^{2^{\overline{2}}-2})$ for $i = 1,2,3$, then σ_1 and σ_2 have the same forms.

4.4 A Generalization of Lemma 1

Lemma 1 can be generalized to more than 3 functions. This leads to further methods of constructions.

Proposition 6. *Let f_1, …, f_m be Boolean functions on F_2^n. For every positive integer l, let σ_l be the Boolean function defined by*

$$\sigma_l = \bigoplus_{1\leq i_1<\ldots<i\ \leq m} \prod_{j=1}^{l} f_i \qquad \textit{if } l \leq m \textit{ and } \sigma_l = 0 \textit{ otherwise.}$$

Then we have $f_1 + \ldots + f_m = \sum_{i\geq 0} 2^i\,\sigma_2$. Denoting by $\widehat{f}$ the Fourier transform of f, that is, $\widehat{f}(s) = \sum_{x\in F_2}\ f(x)(-1)^{x\cdot s}$, this implies $\widehat{f_1}+\ldots+\widehat{f_m} = \sum_{i\geq 0} 2^i\,\widehat{\sigma_2}$. Moreover, if $m+1$ is a power of 2, say $m+1 = 2^r$, then

$$\widehat{\chi_{f_1}} + \ldots + \widehat{\chi_{f}\ } = \sum_{i=0}^{r-1} 2^i\,\widehat{\chi_{\sigma_2}\ } . \tag{14}$$

Proof. Let x be any vector of F_2^n and $j = \sum_{k=1}^{m} f_k(x)$. According to Lucas' Theorem (cf. [45]), the binary expansion of j is $\sum_{i\geq 0}\left[2^i\ \left(\binom{j}{2}\ [\text{mod } 2]\right)\right]$. It is a simple matter to check that $\binom{j}{2}$ [mod 2] $= \sigma_2\ (x)$. Thus, $f_1 + \ldots + f_m = \sum_{i\geq 0} 2^i\,\sigma_2$. The linearity of the Walsh transform with respect to the addition in $\mathbb{Z}$ implies then directly $\widehat{f_1} + \ldots + \widehat{f_m} = \sum_{i\geq 0} 2^i\,\widehat{\sigma_2}$.

If $m+1 = 2^r$, then we have $m = \sum_{i=0}^{r-1} 2^i$. Thus, we deduce $\chi_{f_1} + \ldots + \chi_{f}\ \ = \sum_{i=0}^{r-1} 2^i\,\chi_{\sigma_2}$ from $f_1 + \ldots + f_m = \sum_{i=0}^{r-1} 2^i\,\sigma_2$. The linearity of the Walsh transform implies then relation (14). ◇

Corollary 2. *Let n be any positive integer and k any non-negative integer such that $k \leq n$. Let f_1, …, f_7 be k-th order correlation immune (resp. k-resilient) functions. If two among the functions $\sigma_1 = f_1 \oplus \ldots \oplus f_7$, $\sigma_2 = f_1 f_2 \oplus f_1 f_3 \oplus \ldots \oplus f_6 f_7$ and $\sigma_4 = \bigoplus_{1\leq i_1<\ldots<i_4\leq 7} \prod_{j=1}^{l} f_i$ is k-th order correlation immune (resp. k-resilient) then the third one is k-th order correlation immune (resp. k-resilient).*

The proof is similar to the proof of Theorem 2.

Corollary 3. *Let n be any positive even integer and f_1, …, f_m ($m \leq 7$) be bent functions on F_2^n.*

- *Assume that σ_1 is bent, and that, for every $a \in F_2^n$, the number $\widehat{\chi_{\sigma_4}}(a)$ is divisible by $2^{n/2}$. Then:*
  - *if $m = 5$ and $\widetilde{\sigma_1} = \widetilde{f_1} \oplus \ldots \oplus \widetilde{f_5} \oplus 1$ then σ_2 is bent;*
  - *if $m = 7$ and $\widetilde{\sigma_1} = \widetilde{f_1} \oplus \ldots \oplus \widetilde{f_7}$, then σ_2 is bent;*
- *Assume that $m \in \{5, 7\}$ and that, for every $a \in F_2^n$, the number $\widehat{\chi_{\sigma_4}}(a)$ is divisible by $2^{n/2-1}$ and the number $\widehat{\chi_{\sigma_2}}(a)$ is divisible by $2^{n/2}$, then σ_1 is bent.*

Proof. By hypothesis, we have for $i = 1, \ldots, m$ and for every vector $a \neq 0$: $\widehat{\chi_{f}}\ (a) = -2\widehat{f}_i(a) = (-1)^{\widetilde{f}\ (a)}\,2^{n/2}$.

- If σ_1 is bent and, for every $a \in F_2^n$, the number $\widehat{\chi_{\sigma_4}}(a)$ is divisible by $2^{n/2}$, then $\widehat{\chi_{\sigma_2}}(a)$ is congruent with $\left[(-1)^{\widetilde{f_1}(a)} + \ldots + (-1)^{\widetilde{f}\ (a)} - (-1)^{\widetilde{\sigma_1}(a)}\right]$ $2^{n/2-1}$ modulo $2^{n/2+1}$, for every $a \neq 0$.

  If $m = 5$ and $\widetilde{\sigma_1} = \widetilde{f_1} \oplus \ldots \oplus \widetilde{f_5} \oplus 1$ then, denoting by k the Hamming weight of the word $(\widetilde{f_1}(a), \ldots, \widetilde{f_5}(a))$, the number $\widehat{\chi_{\sigma_2}}(a)$ is congruent with $[5 - 2k + (-1)^k]\, 2^{n/2-1}$ modulo $2^{n/2+1}$.

  If $m = 7$ and $\widetilde{\sigma_1} = \widetilde{f_1} \oplus \ldots \oplus \widetilde{f_7}$ then, denoting by k the Hamming weight of the word $(\widetilde{f_1}(a), \ldots, \widetilde{f_7}(a))$, the number $\widehat{\chi_{\sigma_2}}(a)$ is congruent with $[7 - 2k - (-1)^k]\, 2^{n/2-1}$ modulo $2^{n/2+1}$. So, in both cases, we have $\widehat{\chi_{\sigma_2}}(a) \equiv 2^{n/2}$ [mod $2^{n/2+1}$], and σ_2 is bent, according to Lemma 1 of [7] (which is equivalent to saying that a Boolean function f is bent if and only if $\widehat{\chi_f}(a)$ is congruent with $2^{n/2}$ modulo $2^{n/2+1}$, for every $a \neq 0$; indeed, $a \neq 0$ is sufficient thanks to Parseval's relation).
- If, for every $a \in F_2^n$, the number $\widehat{\chi_{\sigma_4}}(a)$ is divisible by $2^{n/2-1}$ and the number $\widehat{\chi_{\sigma_2}}(a)$ is divisible by $2^{n/2}$, then, for every $a \neq 0$, the number $\widehat{\chi_{\sigma_1}}(a)$ is congruent with $\left[(-1)^{\widetilde{f_1}(a)} + \ldots + (-1)^{\widetilde{f}\ (a)}\right]$ $2^{n/2}$ mod $2^{n/2+1}$. Since $m \in \{5, 7\}$, it is then congruent with $2^{n/2}$ mod $2^{n/2+1}$ and σ_1 is bent, according to Lemma 1 of [7]. ◇

References

1. E.F. Assmus and Key, J. D. *Designs and their Codes*, Cambridge Univ. Press.
2. A. Botev and Y. Tarannikov. Lower bounds on algebraic immunity for recursive constructions of nonlinear filters. Preprint 2004.
3. P. Camion, C. Carlet, P. Charpin, N. Sendrier, On correlation-immune functions, *Advances in Cryptology: Crypto '91, Proceedings, Lecture Notes in Computer Science*, V. 576, pp. 86-100, 1991.
4. A. Canteaut and Trabbia, M. Improved fast correlation attacks using parity-check equations of weight 4 and 5, *Advanced in Cryptology-EUROCRYPT 2000. Lecture notes in computer science* 1807, pp. 573-588, 2000.
5. C. Carlet. Partially-bent functions, *Designs Codes and Cryptography*, 3, 135-145 (1993) and proceedings of CRYPTO' 92, Advances in Cryptology, Lecture Notes in Computer Science 740, Springer Verlag, pp. 280-291, 1993.
6. C. Carlet. Two new classes of bent functions, *EUROCRYPT' 93, Advances in Cryptology, Lecture Notes in Computer Science* 765, Springer Verlag, pp. 77-101, 1994.
7. C. Carlet. Generalized Partial Spreads, *IEEE Transactions on Information Theory*, vol 41, number 5, pp. 1482-1487, 1995.
8. C. Carlet. A construction of bent functions. *Finite Fields and Applications, London Mathematical Society*, Lecture Series 233, Cambridge University Press, pp. 47-58, 1996.
9. C. Carlet. More correlation-immune and resilient functions over Galois fields and Galois rings. *Advances in Cryptology, EUROCRYPT' 97, Lecture Notes in Computer Science* 1233, 422-433, Springer Verlag, 1997.
10. C. Carlet. Recent results on binary bent functions. *International Conference on Combinatorics, Information Theory and Statistics; Journal of Combinatorics, Information and System Sciences*, Vol. 24, Nos. 3-4, pp. 275-291, 1999.

11. C. Carlet. On the coset weight divisibility and nonlinearity of resilient and correlation-immune functions, *Proceedings of SETA'01* (Sequences and their Applications 2001), Discrete Mathematics and Theoretical Computer Science, Springer, pp. 131-144, 2001.
12. C. Carlet. A larger Class of Cryptographic Boolean Functions via a Study of the Maiorana-McFarland Construction. *Advances in Cryptology - CRYPT0 2002, no. 2442 in Lecture Notes in Computer Science*, pp. 549-564, 2002.
13. C. Carlet. On the confusion and diffusion properties of Maiorana-McFarland's and extended Maiorana-McFarland's functions. *Special Issue "Complexity Issues in Coding and Cryptography", dedicated to Prof. Harald Niederreiter on the occasion of his 60th birthday, Journal of Complexity* 20, pp. 182-204, 2004.
14. C. Carlet. On the secondary constructions of resilient and bent functions. Proceedings of the Workshop on Coding, Cryptography and Combinatorics 2003, published by Birkhäuser Verlag, K. Feng, H. Niederreiter and C. Xing Eds., pp. 3-28, 2004.
15. C. Carlet. On the degree, nonlinearity, algebraic thickness and non-normality of Boolean functions, with developments on symmetric functions. *IEEE Transactions on Information Theory*, vol. 50, pp. 2178-2185, 2004.
16. C. Carlet. Concatenating indicators of flats for designing cryptographic functions. *Design, Codes and Cryptography* volume 36, Number 2, pp.189 - 202, 2005.
17. C. Carlet. Designing bent functions and resilient functions from known ones, without extending their number of variables. Proceedings of International Symposium on Information Theory 2005.
18. C. Carlet. Boolean Functions for Cryptography and Error Correcting Codes. Chapter of the monography *Boolean Methods and Models*, Y. Crama and P. Hammer eds, Cambridge University Press, to appear (winter 2005-2006).
19. C. Carlet, D. K. Dalai, K. C. Gupta and S. Maitra. Algebraic Immunity for Cryptographically Significant Boolean Functions: Analysis and Construction. Preprint.
20. C. Carlet, H. Dobbertin and G. Leander. Normal extensions of bent functions. *IEEE Transactions on Information Theory*, vol. 50, no. 11, pp. 2880-2885, 2004.
21. C. Carlet and P. Gaborit. Hyper-bent functions and cyclic codes. To appear in the Journal of Combinatorial Theory, Series A, 2005.
22. C. Carlet and P. Gaborit. On the construction of balanced Boolean functions with a good algebraic immunity. Proceedings of International Symposium on Information Theory 2005. To appear.
23. C. Carlet and P. Sarkar. Spectral Domain Analysis of Correlation Immune and Resilient Boolean Functions. *Finite fields and Applications* 8, pp. 120-130, 2002.
24. C. Carlet and J.L. Yucas. Piecewise Constructions of Bent and Almost Optimal Boolean Functions. To appear in *Designs, Codes and Cryptography*, 2005.
25. N. Courtois. Fast Algebraic Attacks on Stream Ciphers with Linear Feedback. *Advances in cryptology–CRYPTO 2003, Lecture Notes in Computer Science* 2729, pp. 177-194, Springer, 2003.
26. N. Courtois and W. Meier. Algebraic Attacks on Stream Ciphers with Linear Feedback. *Advances in cryptology–EUROCRYPT 2003, Lecture Notes in Computer Science* 2656, pp. 346-359, Springer, 2002.
27. N. Courtois and J. Pieprzyk. Cryptanalysis of block ciphers with overdefined systems of equations. *Advances in cryptology–ASIACRYPT 2002, Lecture Notes in Computer Science* 2501, pp. 267-287, Springer, 2003.
28. D. K. Dalai, K. C. Gupta and S. Maitra. Results on Algebraic Immunity for Cryptographically Significant Boolean Functions. Indocrypt 2004, pp. 92–106, number 3348 in Lecture Notes in Computer Science, Springer Verlag, 2004.

29. D. K. Dalai, K. C. Gupta and S. Maitra. Cryptographically Significant Boolean functions: Construction and Analysis in terms of Algebraic Immunity. *Fast Software Encryption 2005, Lecture Notes in Computer Science* 3557, pp. 98-111, 2005.
30. J. F. Dillon. *Elementary Hadamard Difference sets.* Ph. D. Thesis, Univ. of Maryland, 1974.
31. J. F. Dillon. Elementary Hadamard Difference sets, *Proc. Sixth S-E Conf. Comb. Graph Theory and Comp.*, F. Hoffman et al. (Eds), Winnipeg Utilitas Math, pp. 237-249, 1975.
32. J. F. Dillon and H. Dobbertin. New cyclic difference sets with Singer parameters. Finite Fields and Their Applications 10, pp. 342-389, 2004.
33. H. Dobbertin. Construction of bent functions and balanced Boolean functions with high nonlinearity. *Fast Software Encryption, Second International Workshop*, Lecture Notes in Computer Science 1008, pp. 61-74, 1995.
34. H. Dobbertin, G. Leander, A. Canteaut, C. Carlet, P. Felke and P. Gaborit. Construction of Bent Functions via Niho Power Functions. To appear in the *Journal of Combinatorial Theory, Series A*, 2005.
35. G. Gong. On Existence and Invariant of Algebraic Attacks. Technical report, 2004. http://www.cacr.math.uwaterloo.ca/techreports/2004/corr2004-17.pdf
36. P. Hawkes and G. G. Rose. Rewriting variables: the complexity of fast algebraic attacks on stream ciphers. Cryptology ePrint Archive, Report 2004/081, 2004. http://eprint.iacr.org/2004/081.
37. X.-D. Hou. New constructions of bent functions, *International Conference on Combinatorics, Information Theory and Statistics; Journal of Combinatorics, Information and System Sciences*, Vol. 24, Nos. 3-4, pp. 275-291, 1999.
38. X.-D. Hou. Group actions on binary resilient functions. Appl. Algebra Eng. Commun. Comput. 14(2), pp. 97-115, 2003.
39. X.-D. Hou and P. Langevin. Results on bent functions, *Journal of Combinatorial Theory, Series A*, 80, pp. 232-246, 1997.
40. T. Johansson and Jönsson, F. Improved fast correlation attack on stream ciphers via convolutional codes. *Advances in Cryptology - EUROCRYPT'99, number 1592 in Lecture Notes in Computer Science*, pp. 347-362, 1999.
41. T. Johansson and Jönsson, F. Fast correlation attacks based on turbo code techniques. *Advances in Cryptology - CRYPTO'99, number 1666 in Lecture Notes in Computer Science*, pp. 181-197, 1999.
42. K. Khoo, G.-E. Tan, H.-K. Lee and G. Gong. Comparision of Boolean function design. Proceedings of International Symposium on Information Theory 2005.
43. G. Leander. Bent functions with 2^r Niho exponents. *Proceedings of the Workshop on Coding and Cryptography* 2005, Bergen, pp. 454-461, 2005.
44. G. Leander. Monomial bent functions. *Proceedings of the Workshop on Coding and Cryptography* 2005, Bergen, pp. 462-470, 2005.
45. F. J. Mac Williams and N. J. Sloane. *The theory of error-correcting codes*, Amsterdam, North Holland, 1977.
46. S. Maity and S. Maitra. Minimum distance between bent and 1-resilient Boolean functions. *Proceedings of Fast Software Encryption* 2004, LNCS 3017, pp. 143-160, 2004.
47. W. Meier and O. Staffelbach. Nonlinearity Criteria for Cryptographic Functions, *Advances in Cryptology, EUROCRYPT' 89, Lecture Notes in Computer Science 434, pp. 549-562, Springer Verlag*, 1990.
48. W. Meier, E. Pasalic and C. Carlet. Algebraic attacks and decomposition of Boolean functions. *Advances in Cryptology, EUROCRYPT 2004, Lecture Notes in Computer Science, Springer Verlag* 3027, pp. 474-491, 2004.

49. D. Olejár and M. Stanek. ”On cryptographic properties of random Boolean functions.” Journal of Universal Computer Science, vol. 4, No.8, pp. 705-717, 1998.
50. E. Pasalic and S. Maitra. A Maiorana-McFarland type construction for resilient Boolean functions on n variables (n even) with nonlinearity $> 2^{n-1} - 2^{n/2} + 2^{n/2-2}$. *Proceedings of the Workshop on Coding and Cryptography 2003*, pp. 365-374, 2003.
51. Olsen, J. D., Scholtz, R. A. and L. R. Welch. Bent function sequences, *IEEE Trans. on Inf. Theory* , vol IT- 28, n° 6, 1982.
52. E. Pasalic. Degree optimized resilient Boolean functions from Maiorana-McFarland class. In *9th IMA Conference on Cryptography and Coding*, 2003.
53. O. S. Rothaus. On “bent” functions, *J. Comb. Theory*, 20A, 300-305, 1976.
54. E. Pasalic, T. Johansson, S. Maitra and P. Sarkar. New constructions of resilient and correlation immune Boolean functions achieving upper bounds on nonlinearity. *Proceedings of the Workshop on Coding and Cryptography* 2001, published by *Electronic Notes in Discrete Mathematics*, Elsevier, vo. 6, pp. 425-434, 2001.
55. P. Sarkar and S. Maitra. Construction of nonlinear Boolean functions with important cryptographic properties. *Advances in Cryptology - EUROCRYPT 2000*, no. 1807 in Lecture Notes in Computer Science, Springer Verlag, pp. 485-506, 2000.
56. P. Sarkar and S. Maitra. Nonlinearity Bounds and Constructions of Resilient Boolean Functions. *CRYPTO 2000, LNCS* Vol. 1880, ed. Mihir Bellare, pp. 515-532, 2000.
57. T. Siegenthaler. Correlation-immunity of nonlinear combining functions for cryptographic applications. *IEEE Transactions on Information theory*, V. IT-30, No 5, pp. 776-780, 1984.
58. Siegenthaler, T. Decrypting a Class of Stream Ciphers Using Ciphertext Only. *IEEE Transactions on Computer, V. C-34*, No 1, pp. 81-85, 1985.
59. L. Simpson, E. Dawson, J. Golic and W. Millan. LILI Keystream generator, *P roceedings of SAC'2000, Lecture Notes in Computer Science* 1807, Springer, pp. 248-261, 2001; cf. www.isrc.qut.edu.au/lili/.
60. Y. V. Tarannikov. On resilient Boolean functions with maximum possible nonlinearity. *Proceedings of INDOCRYPT 2000, Lecture Notes in Computer Science* 1977, pp. 19-30, 2000.
61. Xiao Guo-Zhen and Massey, J. L. A Spectral Characterization of Correlation-Immune Combining Functions. *IEEE Trans. Inf. Theory*, Vol IT 34, n° 3, pp. 569-571, 1988.
62. Y. Zheng and X. M. Zhang. Plateaued functions. *ICICS'99, Lecture Notes in Computer Science*, Heidelberg, Ed., Springer-Verlag, vol. 1726, pp. 284-300, 1999.

On Generalized Parity Checks*

Robert J. McEliece and Edwin Soedarmadji

California Institute of Technology, Department of Electrical Engineering,
MC 136-93, Pasadena, CA 91125, USA
{rjm, edwin}@systems.caltech.edu

Abstract. An ordinary parity-check is an extra bit p appended to a block $(x_1, \ldots, x_k)$ of k information bits such that the resulting codeword $(x_1, \ldots, x_k, p)$ is capable of detecting one error. The choices for p are

$$\begin{aligned} p_0 &= x_1 + \cdots + x_k \pmod 2 \quad \text{(even parity)} \\ p_1 &= x_1 + \cdots + x_k + 1 \pmod 2 \quad \text{(odd parity)} \end{aligned}$$

In this paper we consider defining a parity-check if the underlying alphabet is nonbinary. The obvious definition is of course

$$p = x_1 + \cdots + x_k + \alpha \pmod q.$$

We shall show that this obvious choice is the only choice for $q = 2$, and up to a natural equivalence the only choice for $q = 3$. For $q \geq 4$, however, the situation is much more complicated.

Notation.

X: a set with q elements
X^k: The set of k-tuples of elements of X,
$d_H(\boldsymbol{x}, \boldsymbol{y})$ The Hamming distance between $\boldsymbol{x}$ and $\boldsymbol{y}$.
$B(\boldsymbol{u}) = \{\boldsymbol{x} : d_H(\boldsymbol{u}, \boldsymbol{x}) \leq 1\}$. The unit ball with center $\boldsymbol{u}$.
S_X The set of all permutations of the set X.
$F(n, k)$ the set of all (k, q) Generalized Parity Checks.

1 Introduction. Existence

Let X be a finite set with q elements. A function $p : X^k \to X$ such that $p(\boldsymbol{x}) \neq p(\boldsymbol{y})$ whenever $\boldsymbol{x}$ and $\boldsymbol{y}$ are adjacent, i.e., $d_H(\boldsymbol{x}, \boldsymbol{y}) = 1$ is called an (k, q) generalized parity check. Alternatively, $p(\boldsymbol{x})$ is the "color" assigned to $\boldsymbol{x}$, with the requirement that two adjacent vertices must be assigned distinct colors. The following theorem establishes a link to classical combinatorial coding theory.

Theorem 1.1. *$p(\boldsymbol{x})$ is an (k, q) GPC iff $\left\{(\boldsymbol{x}, p(\boldsymbol{x})) : \boldsymbol{x} \in X^k\right\}$ is an $(k+1, q^k, 2)$ (MDS) code.*

* This work was sponsored by NSF grant CCF-0514881, Qualcomm, Sony, and the Lee Center for Advanced Networking.

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 29–34, 2006.

Proof. Immediate. □

For $q = 2$ it is presumably well known that the only generalized parity checks are the ordinary even and odd parity checks $p_0(x_1, \ldots, x_k) = x_1 + \cdots + x_k \pmod 2$ and $p_1(x_1, \ldots, x_k) = x_1 + \cdots + x_k + 1 \pmod 2$. However, for $q > 2$ the situation is much more interesting. As a start, the next theorem guarantees the existence of at least one GPC for every (k, q).

Theorem 1.2. *If $X = G$ is a finite group, written multiplicatively,*

$$p_G(\boldsymbol{x}) = x_1 \cdots x_k$$

is an (k, q)-GPC. In particular, if $X = Z_q$, the cyclic group of order q, written additively, then

$$p_q(\boldsymbol{x}) = x_1 + \cdots + x_k \pmod q$$

is a (k, q) GPC.

Proof. Suppose $d_H(\boldsymbol{x}, \boldsymbol{y}) = 1$ but $x_1 \cdots x_k = y_1 \cdots y_k$. If $x_1 = y_1$, the cancellation law implies $x_2 \cdots x_k = y_2 \cdots y_k$ and the proof proceeds by induction on k. If $x_1 \neq y_1$, then $x_k = y_k$ and again applying the cancellation law, $x_1 \cdots x_{k-1} = y_1 \cdots y_{k-1}$ and the proof proceeds as before. □

Example 1.1. For $q = 4$ there are two nonisomorphic groups, viz., the cyclic group Z_4 and the elementary Abelian group E_4. The corresponding addition tables are given below. Thus there are (at least) two essentially different ways of assigning parity if the underlying alphabet has size 4.

$$Z_4 = \begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 0 & 1 & 2 & 3 \\ 1 & 1 & 2 & 3 & 0 \\ 2 & 2 & 3 & 0 & 1 \\ 3 & 3 & 0 & 1 & 2 \end{array} \qquad E_4 = \begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 0 & 1 & 2 & 3 \\ 1 & 1 & 0 & 3 & 2 \\ 2 & 2 & 3 & 0 & 1 \\ 3 & 3 & 2 & 1 & 0 \end{array}$$

Example 1.2. For $q = 5$ there is only one group, viz., Z_5. But there exist a nonassociative algebraic structure called a *loop*, which can also be used to construct GPC's.

$$L_5 = \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 1 & 2 & 3 & 4 & 5 \\ 2 & 2 & 4 & 1 & 5 & 3 \\ 3 & 3 & 5 & 4 & 2 & 1 \\ 4 & 4 & 1 & 5 & 3 & 2 \\ 5 & 5 & 3 & 2 & 1 & 4 \end{array}$$

2 Equivalent GPC's

Given one GPC, it is easy to manufacture more. A quite general method is to use a monomial transformation. A monomial transformation of X^k is of the form

$$M\boldsymbol{x} = (\pi_1(x_1), \ldots, \pi_k(x_k))P,$$

$k =$	*1*	*2*	*3*	*4*	
$q =$ *2*	1	1	1	1	√
3	1	1	1	1	√
4	1	4	4^3	**3460**	
5	1	56	40246		
6	1	9408			

$\mathfrak{m}_{k,q}$–isotopy classes
($N(k,q)/q(q-1)!^k$)

$k =$	*1*	*2*	*3*	*4*	
$q =$ *2*	1	1	1	1	√
3	1	1	1	1	√
4	1	2	**4**	**7**	
5	1	2			
6	1	12			

$\mathfrak{M}_{k,q}$–equivalence classes
(Nonisomorphic GPCs)

Fig. 1. Summary of Results. A checkmark indicates a pattern that continues indefinitely.

where P is an $k \times k$ permutation matrix and $\pi_1 \ldots, \pi_k \in S_X$. The set of all $k!q!^k$ monomial transformations forms a group under the composition operation and is denoted by $\mathfrak{M}_{k,q}$.

Theorem 2.1. *If $p(\boldsymbol{x})$ is an (k, q) GPC, and M is a monomial transformation of X^k, then*

$$p'(\boldsymbol{x}) = p(M\boldsymbol{x})$$

is a (k, q)-GPC. If $q(\boldsymbol{x}) = p(M\boldsymbol{x})$ for some M, we say that p and q are monomially equivalent.

Proof. If $p'(\boldsymbol{x}) = p'(\boldsymbol{y})$, then $d_H(M\boldsymbol{x}, M\boldsymbol{y}) \geq 2$. But It is easy to see that every monomial transformation is an isometry, i.e., $d_H(\boldsymbol{x}, \boldsymbol{y}) = d_H(M\boldsymbol{x}, M\boldsymbol{y})$. □

As we shall see below, for $q = 2$ and $q = 3$ all GPC's are monomially equivalent, but the next theorem guarantees that for most values of q, there exist more than one monomial equivalence class.

Theorem 2.2. *If G and H are nonisomorphic groups of order q, then for $k \geq 2$, p_G and p_H are not monomially equivalent.*

Proof. Omitted. □

Let us denote by $N(k, q)$ the number of nonisomorphic (k, q) GPC's. The following table summarizes what we know about this number.

3 Reduced GPC's

A *reduced* monomial transformation of X^k is of the form

$$m\boldsymbol{x} = (\pi_1(x_1), \ldots, \pi_k(x_k)) \quad \pi_1(0) = \cdots = \pi_k(0).$$

The set of all $q(q-1)!^k$ reduced monomial transformations is a subgroup of $\mathfrak{M}_{k,q}$ denoted by $\mathfrak{m}_{k,q}$. $\mathfrak{m}_{n,q}$ is the symmetry group of the unit ball $B(\boldsymbol{u})$ (see Fig. 2). Alternatively, it is the stabilizer of $B(\boldsymbol{u})$ in the group $\mathfrak{M}_{k,q}$.

Theorem 3.1. *Let $f, g \in F(k,q)$, $\boldsymbol{u} \in X^k$. Then there exists a unique $\alpha \in \mathfrak{m}_{k,q}$ such that $f(\alpha^{-1}\boldsymbol{x}) = g(\boldsymbol{x})$ for all $\boldsymbol{x} \in B(\boldsymbol{u})$.*

Proof. Let $B(\boldsymbol{u}) = \{b_{i,j}(\boldsymbol{u})\}_{i=1}^{k}{}_{j=0}^{q-1}$, i.e., $b_{i,j}(\boldsymbol{u})$ is the k-vector whose ith component is j, all other components match $\boldsymbol{u}$. Since $\{b_{i,0}, \ldots, b_{i,q-1}\}$ is a clique, $\{f(b_{i,0}), \ldots, f(b_{i,q-1})\} = \{g(b_{i,0}), \ldots, g(b_{i,q-1})\} = \{0, 1, \ldots, q-1\}$, and we can define a permutation $\pi_i(j)$ as follows.

$$g(b_{i,\pi\ (j)}) = f(b_{i,j})$$

Then $\boldsymbol{\pi}(b_{i,j}) = b_{i,\pi\ (j)}$, so $f(\boldsymbol{\pi}^{-1}b_{i,j}) = g(b_{i,j})$ for all i and j. □

Corollary 3.1. *Every GPC in $F(k,q)$ is equivalent to one that satisfies*

$$f(\boldsymbol{x}) = \boldsymbol{x} \quad \text{for all } \boldsymbol{x} \text{ of weight} \leq 1.$$

This is called a reduced GPC.

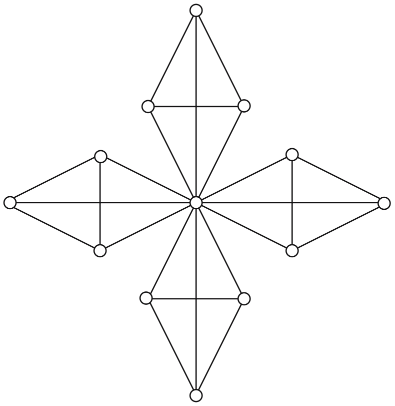

Fig. 2. The $n = 4$, $q = 4$ unit ball. There are 4 $3!^4$ ways to 4-color it. By appropriately choosing π_1, π_2, π_3, and π_4 from S_4, one can convert any one of these colorings to any other.

4 Uniqueness

Theorem 1.2 guarantees the existence of GPC's for all q and k. In this section we will show that these GPC's are unique for $q = 2$ and unique (up to monomial equivalence) for $q = 3$.

Theorem 4.1. *Let $X = \{0,1\}$. The only two $(k,2)$ GPCs are given by*

$$f_0(\boldsymbol{x}) = x_1 + \cdots + x_k \pmod 2$$
$$f_1(\boldsymbol{x}) = x_1 + \cdots + x_k + 1 \pmod 2.$$

Proof. Let $(\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{M-1})$ be a Gray Code, i.e., a list of the $M = 2^k$ elements of X^k such that $\boldsymbol{x}_0 = 0$ and $d_H(\boldsymbol{x}_i, \boldsymbol{x}_{i+1}) = 1$ for $i = 0, 1, \ldots, M-2$. If $f(\boldsymbol{x}_0) = 0$, then $f(\boldsymbol{x}_1) = 1$ etc., and $f = f_0$. Otherwise $f(\boldsymbol{x}_0) = 1$, $f(\boldsymbol{x}_1) = 0$ etc., and $f = f_1$. □

Theorem 4.2. *Let $X = \{0, 1, 2\}$. Every $(k, 3)$ GPC is monomially equivalent to*

$$f_0(\boldsymbol{x}) = x_1 + \cdots + x_k \pmod 3.$$

Alternatively, every $(k, 3)$ GPC is of the form

$$f(\boldsymbol{x}) = a_1 x_1 + \cdots + a_k x_k + b,$$

where $a_1, \ldots, a_k$ are from $GF(3)^*$, and $b \in GF(3)$.

Proof. We will show that if $f(\boldsymbol{x}) \in F(k, 3)$, and $f(\boldsymbol{x}) = f_0(\boldsymbol{x})$ for all $\boldsymbol{x} \in B(0)$, that $f(\boldsymbol{x}) = f_0(\boldsymbol{x})$ for all $\boldsymbol{x} \in X^k$. We will induct on $w_H(\boldsymbol{x})$, the cases $w = 0$ and $w = 1$ being the hypothesis.

Let $w(\boldsymbol{x}) = j + 1$. Then $\boldsymbol{x}$ has $j + 1$ neighbors of weight j., viz., the vectors $\boldsymbol{y}$ obtained from $\boldsymbol{x}$ by changing one nonzero component to zero. By induction, $f(\boldsymbol{y}) = f_0(\boldsymbol{y})$ for each of these neighbors, so $f(\boldsymbol{x}) \neq f_0(\boldsymbol{y}) = f_0(\boldsymbol{x}) - x_i$. This determines $f(\boldsymbol{x})$ except when $\boldsymbol{x}$ is a two-valued vector, say $\boldsymbol{x} = (1111000)$. But then $\boldsymbol{x}$ belongs to a clique $\{(0111000), (1111000), (2111000)\}$ in which every vector except $\boldsymbol{x}$ itself is already colored, either because it has weight j or because it has weight $j + 1$ and is three-valued. □

5 GPC's and Latin Squares

A Latin square of order q is a $q \times q$ matrix with entries from X, such that every element of X appears exactly once in each row and column [3]. It is a classically difficult combinatorial problem to determine the number and type of Latin squares. Thus the following theorem demonstrates that the GPC problem, being essentially the Latin hypercube problem, is even harder.

Theorem 5.1. *There is a one-to-one correspondence between $(2, q)$ GPC's and $q \times q$ Latin Squares.*

Proof. Let $p(x, y)$ be a $(2, q)$ GPC. Then the matrix whose (x, y)th entry is $p(x, y)$ is a Latin square. □

Example 5.1. Here are the inequivalent Latin squares of orders 2, 3, 4, and 5; alternatively the inequivalent (2,2), (2,3), (2,4), and (2,5) GPC's.

The 2×2 Latin square (the addition table for the cyclic group Z_2).

$$Z_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

The 3 × 3 Latin square (the addition table for the cyclic group Z_3).

$$Z_3 = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \end{bmatrix}$$

The two inequivalent 4 × 4 Latin squares (the addition table for the cyclic group Z_4 and the elementary Abelian group E_4).

$$Z_4 = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & 0 & 1 & 2 \end{bmatrix} \qquad E_4 = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 3 & 2 \\ 2 & 3 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{bmatrix}$$

The two inequivalent 5 × 5 Latin squares (the addition table for the cyclic group Z_5 and the loop L_5).

$$Z_5 = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 0 \\ 2 & 3 & 4 & 0 & 1 \\ 3 & 4 & 0 & 1 & 2 \\ 4 & 0 & 1 & 2 & 3 \end{bmatrix} \qquad L_5 = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 3 & 0 & 4 & 2 \\ 2 & 4 & 3 & 1 & 0 \\ 3 & 0 & 4 & 2 & 1 \\ 4 & 2 & 1 & 0 & 3 \end{bmatrix}$$

References

1. F. Gray. Pulse code communication, March 17, 1953. U.S. patent no. 2,632,058.
2. W.C. Huffman and V. Pless, *Fundamentals of Error-Correcting Codes*, Cambridge: Cambridge University Press, 2003.
3. J. H. van Lint and R. M. Wilson, *A Course in Combinatorics, 2nd ed.* Cambridge: Cambridge University Press, 2001.
4. Eric W. Weisstein. "Latin Square." From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/LatinSquare.html

Cryptography Based on Bilinear Maps

Tatsuaki Okamoto

NTT Laboratories, Nippon Telegraph and Telephone Corporation,
1-1 Hikarino-oka, Yokosuka 239-0847, Japan
okamoto.tatsuaki@lab.ntt.co.jp

Abstract. The bilinear mapping technique that uses the (Weil and Tate) pairings over elliptic (or hyperelliptic) curves represents a great breakthrough in cryptography. This paper surveys this new trend in cryptography, and emphasizes the design of efficient cryptographic primitives that are provably secure in the standard model (i.e., without the random oracle model).

1 Introduction

Elliptic curves have been applied to practical cryptographic designs for two decades. The advantage of elliptic curve based cryptosystems, ECC, over other public-key cryptosystems is their short key size, high processing throughput, and low bandwidth. For example, the typical key size of ECC that guarantees the security comparable to that of 1024 bit key size with the RSA cryptosystems is considered to be just 160 bits. Therefore, several of the most efficient public-key encryption schemes and digital signatures are ECC such as EC-ElGamal (the elliptic curve version of ElGamal) and EC-DSA.

The reason why ECC has such short key lengths is that the *index calculus* technique is considered to be ineffective for computing the discrete logarithm (DL) of the elliptic curve group over finite fields, while it can effectively compute integer factoring and DL of the multiplicative group of a finite field.

However, the mathematical features that are specific to elliptic curve groups compared with multiplicative groups are not only the inapplicability of the index calculus. The most characteristic property of an elliptic curve group is its group structure, which is isomorphic to the product of *two cyclic groups*.

The pairing on an elliptic curve to be introduced in this paper employs this group structure, and is specific to elliptic curve groups (and the generalizations such as hyperelliptic curve groups). In this sense, two decades after we started applying elliptic curves to cryptography, we have finally reached the application of the pairing to cryptographic design, the most essential and natural application of elliptic curves in cryptography.

2 Elliptic Curve Cryptosystems

The application of elliptic curves to cryptography uses elliptic curves defined over finite fields.

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 35–50, 2006.

We now introduce some notations. $E(\mathbb{F}_q)$ is a set of $\mathbb{F}_q$-rational points of elliptic curve E over finite field $\mathbb{F}_q$. That is, $E(\mathbb{F}_q)$ is a set of points satisfying $y^2 = x^3 + ax + b$ (other equations are used for finite field $\mathbb{F}_q$ with characteristic 2 and 3) and the special point $\mathcal{O}$ called the infinity point.

A group operation is defined over $E(\mathbb{F}_q)$ and $\mathcal{O}$ is the identity. We now express the group operation by $+$. The discrete logarithm (DL) problem of $E(\mathbb{F}_q)$ is to compute $x \in \mathbb{N}$, given (G, Y), where G is a base point of $E(\mathbb{F}_q)$ and $Y = xG$, which is $G + \cdots + G$ (G is added x times). (After Section 5, we will use the multiplicative form for the group operations in place of the conventional additive form here.)

Elliptic curve cryptosystems (ECC) are constructed on the group of $E(\mathbb{F}_q)$. The security of ECC depends on the difficulty of computing the DL problem of $E(\mathbb{F}_q)$. An ECC scheme can be designed in a manner similar to that of a scheme based on the multiplicative DL problem. For example, EC-DH, EC-ElGamal and EC-DSA are constructed over $E(\mathbb{F}_q)$ in a manner analogous to that of DH, ElGamal and DSA.

Cryptosystems based on bilinear maps (of elliptic curves) are a class of elliptic curve cryptosystems, but have very different features than the conventional ECC.

3 Pairing

The Weil pairing is defined over elliptic curves as follows: Let $E/\mathbb{F}_q$ be an elliptic curve defined over $\mathbb{F}_q$ and m be an integer coprime to q. Let $E[m]$ be the set of m torsion points of $E/\mathbb{F}_q$ (i.e., $E[m] = \{P \mid P \in \overline{\mathbb{F}}_q \wedge mP = \mathcal{O}\}$). $E[m]$ is isomorphic to $\mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}$. The Weil pairing, $e_m(P, Q) \in \overline{\mathbb{F}}_q^*$, is defined for two points, P and Q, in $E[m]$, and has the following properties:

(1) For any $P, Q \in E[m]$, $(e_m(P, Q))^m = 1$.
(2) For all $P \in E[m]$, $e_m(P, P) = 1$.
(3) Bilinear: for any $P, Q, P_1, P_2, Q_1, Q_2 \in E[m]$,

$$e_m(P_1 + P_2, Q) = e_m(P_1, Q)e_m(P_2, Q),$$
$$e_m(P, Q_1 + Q_2) = e_m(P, Q_1)e_m(P, Q_2).$$

(4) Alternating: for any $P, Q \in E[m]$, $e_m(P, Q) = e_m(Q, P)^{-1}$.
(5) Non-degenerate: if $e_m(P, Q) = 1$ for any $P \in E[m]$, then $Q = \mathcal{O}$.

That is, $e_m(P, Q)$ bilinearly maps two points, P and Q, in $E[m]$ to an m-th root of unity in $\overline{\mathbb{F}}_q^*$.

Note that there exists an extension field, $\mathbb{F}_q$, such that $E(\mathbb{F}_q\ )$ includes $E[m]$. Then $e_m(P, Q)$ is an m-th root of unity in $\mathbb{F}_q^*$.

The Weil pairing can be efficiently calculated by Miller's algorithm. The Tate pairing also has similar properties, and is often employed in cryptographic applications, since it is faster to compute a Tate pairing than a Weil pairing in typical implementations.

Historically, bilinear mapping was first used to attack elliptic curve cryptosystems on supersingular curves in the early 1990's [29] (the attack is often called the MOV reduction). However, in the recent application of bilinear maps to cryptography, they are used not for a negative purposes (i.e., attacking cryptographic schemes) but for positive purposes (i.e., designing cryptographic schemes).

4 Variant of the Weil/Tate Pairing

When we apply the Weil/Tate pairing to a general elliptic curve, we have to use an extension field $\mathbb{F}_q$ with huge extension degree k (in general k is exponentially large in $|q|$). One of the most suitable curves for the application of Weil/Tate pairing to cryptography is *supersingular curves*, since the extension degree is at most 6 for supersingular curves. (It is ironical that supersingular curves were considered to be unsuitable for application to cryptography as evidenced by the MOV reduction.)

There is another merit of supersingular curves when employing bilinear maps for cryptography. That is, a supersingular curve has (efficiently computable) isomorphism, ϕ, called the *distortion map.*

Let E be a supersingular curve over $\mathbb{F}_q$ and the order of point $G_1 \in E(\mathbb{F}_q)$ be m. Then, there exists an extension degree $k(\leq 6)$ and $G_2 \in E(\mathbb{F}_q\ )$ such that $E[m] \cong \langle G_1 \rangle \times \langle G_2 \rangle$, and ϕ is the isomorphism from $\langle G_1 \rangle$ to $\langle G_2 \rangle$, where $G_2 = \phi(G_1)$. We can then define a variant of the Weil pairing $\hat{e}_m$ over two points, P and Q, in $E(\mathbb{F}_q)$ as follows:

$$\hat{e}_m(P, Q) = e_m(P, \phi(Q)) \in \mathbb{F}_q^* \ .$$

Here note that $\hat{e}_m(P, P) \neq 1$ and $\hat{e}_m(P, Q) = \hat{e}_m(Q, P)$, while $e_m(P, P) = 1$ and $e_m(P, Q) = e_m(Q, P)^{-1}$. So, this variant of Weil pairing $\hat{e}_m$ is called a *symmetric* pairing, while the original Weil pairing e_m is called an *asymmetric* pairing.

The advantage of this Weil pairing variant $\hat{e}_m : \langle G_1 \rangle \times \langle G_1 \rangle \rightarrow \mathbb{F}_q^*$ is that it is defined over two points in $E(\mathbb{F}_q)$ (two elements in $\langle G_1 \rangle$), while e_m is defined over a point in $E(\mathbb{F}_q)$ and another point in $E(\mathbb{F}_q\ )$. (For example, if the size of an element of $\mathbb{F}_q$ is 200 bits and extension degree k is 6, then the size of an element of $\mathbb{F}_q$ and the size of $\hat{e}_m(P, Q)$ and $e_m(P, Q')$ are 1200 bits.)

In some applications, however, a general curve (not a supersingular curve) may be more suitable, since a general curve offers more freedom in selecting the extension degree and other properties. Some methods have been proposed to efficiently select a general curve that has a low extension degree applicable to the pairing [28, 31].

5 Cryptography Based on Bilinear Maps

5.1 Bilinear Groups

Hereafter, we will consider only this symmetric variant of Weil pairing (not the original Weil pairing) as a bilinear map, but almost all schemes that we will introduce in this paper can be also realized with the original Weil pairing.

For simplicity of description, we express the symmetric pairing $\hat{e}_m : \langle G_1 \rangle \times \langle G_1 \rangle \to \mathbb{F}_q^*$ by bilinear map e from a multiplicative group, $\mathbb{G}$, to another multiplicative group, $\mathbb{G}_T$, i.e., $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ such that:

1. $\mathbb{G}$ is a cyclic group of prime order p,
2. g is a generator of $\mathbb{G}$,
3. e is a non-degenerate bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, where $|\mathbb{G}| = |\mathbb{G}_T| = p$, and
4. e and the group action in $\mathbb{G}$ and $\mathbb{G}_T$ can be computed efficiently.

5.2 Brief Overview of Bilinear-Map-Based Cryptography

Around 2000, application of the bilinear maps to cryptography was initiated by Verheul [35], Joux [27], and Sakai, Ohgishi and Kasahara [33]. Verheul introduced the above-mentioned Weil pairing variant, and Joux proposed a key distribution system among three parties (three party version of the Diffie-Hellman key distribution) by using the Weil pairing variant. Sakai, Ohgishi and Kasahara solved the problem on the identity-based encryption (IBE) that had been open since 1984 when Shamir proposed the concept of IBE.

Following these pioneer works, Boneh and others drastically exploited the possibility of applying bilinear maps to cryptography. Boneh and Franklin [11] formalized the security of IBE as the IND-ID-CCA2 (indistinguishable against adaptively chosen-ciphertext attacks under chosen identity attacks) security and proposed an IND-ID-CCA2 secure IBE scheme in the random oracle model [4]. Boneh, Lynn and Shacham [13] proposed a new signature scheme whose signatures are shorter than those of any previous scheme. The security proof is also based on the random oracle model.

Then, more than two or three hundred papers on bilinear-map-based cryptography have been published for the last few years, and they cover very broad areas of cryptography [2].

One of the most interesting applications of bilinear maps to cryptography is to construct practical encryption/signature schemes that are proven to be secure in the standard model (without the random oracle model). Previously only a few such schemes (e.g., Cramer-Shoup schemes [20, 21, 22]) were proposed.

Interestingly IBE plays a key role of constructing practical secure schemes in the standard model. That is, a secure IBE scheme in the standard model can be used to construct secure public-key encryption/signature schemes in the standard model. (In addition, hierarchical IBE (HIBE) [25] is used to construct forward-secure public-key encryption schemes and CCA2 secure IBE schemes [18, 19].)

Hereafter we will introduce how bilinear maps are applied to constructing secure IBE/encryption/signature schemes in the standard model.

6 Computational Assumptions

Let $\mathbb{G}$ be a bilinear group of prime order p and g be a generator of $\mathbb{G}$. Here we review several computational assumptions on the bilinear maps, which are assumed in the bilinear-map-based cryptographic schemes to be introduced in this paper.

6.1 Bilinear Diffie-Hellman Assumption

The Bilinear Diffie-Hellman (BDH) problem [11, 27] in $\mathbb{G}$ is as follows: given a tuple g, g^a, g^b, $g^c \in \mathbb{G}$ as input, output $e(g,g)^{abc} \in \mathbb{G}_T$. The advantage of adversary $\mathcal{A}$ for the BDH problem is

$$\Pr[\mathcal{A}(g, g^a, g^b, g^c) = e(g,g)^{abc}].$$

Similarly, the advantage of adversary $\mathcal{B}$ for the Decisional BDH (DBDH) problem is

$$|\Pr[\mathcal{B}(g, g^a, g^b, g^c, e(g,g)^{abc}) = 0] - \Pr[\mathcal{B}(g, g^a, g^b, g^c, T) = 0]|,$$

where T is randomly selected from $\mathbb{G}_T$.

Definition 1. *We say that the (Decisional) BDH assumption holds in $\mathbb{G}$ if any probabilistic polynomial time adversary has negligible advantage for the (Decisional) BDH problem.*

6.2 Bilinear Diffie-Hellman Inversion Assumption

The q Bilinear Diffie-Hellman Inversion (q-BDHI) problem [7] is defined as follows: given the $(q+1)$-tuple $(g, g^x,\ g^{x^2}, \ldots, g^{x}\ \ ) \in (\mathbb{G})^{q+1}$ as input, compute $e(g,g)^{1/x} \in \mathbb{G}_T$. The advantage of an adversary $\mathcal{A}$ for q-BDHI is

$$\Pr[\mathcal{A}(g, g^x, g^{x^2}, \ldots, g^{x}\ \ ) = e(g,g)^{1/x}].$$

Similarly, the advantage of adversary $\mathcal{B}$ for the Decisional q-BDHI (q-DBDHI) problem is

$$|\Pr[\mathcal{B}(g, g^x, g^{x^2}, \ldots, g^{x}\ \ , e(g,g)^{1/x}) = 0] - \Pr[\mathcal{B}(g, g^x, g^{x^2}, \ldots, g^{x}\ \ , T) = 0]|,$$

where T is randomly selected from $\mathbb{G}_T$.

Definition 2. *We say that the (Decisional) q-BDHI assumption holds in $\mathbb{G}$ if any probabilistic polynomial time adversary has negligible advantage for the (Decisional) q-BDHI problem.*

It is not known if the q-BDHI assumption, for $q > 1$, is equivalent to BDH.

In this paper, we often drop the q and refer to the (Decisional) BDHI assumption.

6.3 Strong Diffie-Hellman Assumption

The q Strong Diffie-Hellman (q-SDH) problem [8] is defined as follows: given the $(q+1)$-tuple $(g, g^x, g^{x^2}, \ldots, g^{x}\ \ ) \in (\mathbb{G})^{q+1}$ as input, compute $(g^{1/(x+c)}, c) \in \mathbb{G} \times \mathbb{N}$. The advantage of an adversary $\mathcal{A}$ for q-SDH is

$$\Pr[\mathcal{A}(g, g^x, g^{x^2}, \ldots, g^{x}\ \ ) = (g^{1/(x+c)}, c)].$$

Definition 3. *We say that the q-SDH assumption holds in $\mathbb{G}$ if any probabilistic polynomial time adversary has negligible advantage for the q-SDH problem.*

In this paper, similarly to the BDHI assumption, we often drop the q and refer to the SDH assumption.

7 Identity-Based Encryption (IBE)

Identity-based encryption (IBE) [34] is a variant of public-key encryption (PKE), where the identity of a user is employed in place of the user's public-key. In this concept,

Setup: A trusted party (authority) generates a pair of secret-key x (master secret key) and public-key y (system parameter).

Extract: The trusted party also generates A's secret decryption key, s_A, from the identity of A and securely sends s_A to A.

Encrypt: When B encrypts a message m to A, B utilizes A's identity, ID_A (in place of A's public-key in PKE). Let c_A be a ciphertext of m encrypted by ID_A.

Decrypt: A can decrypt ciphertext c_A by using A's decryption key s_A.

Although IBE itself is a very useful primitive in cryptography, here we will review IBE as a building block of designing practical secure PKE/signature schemes in the standard model.

7.1 Security of IBE

Boneh and Franklin [11] define the security, IND-ID-CCA2 (indistinguishable against adaptively chosen-ciphertext attacks under chosen identity attacks), for IBE systems. We now informally introduce the definition as follows:

Definition 4. *(Security of IBE: IND-ID-CCA2) Let us consider the following experiment between an adversary, $\mathcal{A}$, and the challenger, $\mathcal{C}$.*

1. *First, $\mathcal{C}$ generates a system parameter of IBE and sends it to $\mathcal{A}$.*
2. *$\mathcal{A}$ is allowed to ask two types of queries, extraction queries and decryption queries, to $\mathcal{C}$. Here, an extraction query is an identity, ID_i, to which $\mathcal{C}$ replies the corresponding decryption key, d_i, and a decryption query is a ciphertext, c_j, along with an identity, ID_j, to which $\mathcal{C}$ replies with the corresponding plaintext, m_j.*

3. *$\mathcal{A}$ is also allowed to adaptively choose an identity, ID*, and two messages, m_0 and m_1, that $\mathcal{C}$ wishes to attack, then $\mathcal{C}$ replies with a ciphertext, c^*, of m_b (b is randomly chosen from $\{0,1\}$) with respect to identity ID*.*
4. *Finally $\mathcal{A}$ outputs a bit, b^*. Let Advantage be $|2\Pr[b = b^*] - 1|$.*

An IBE scheme is IND-ID-CCA2 if, for any probabilistic polynomial-time $\mathcal{A}$, Advantage is negligibly small.

In the above-mentioned definition of IND-ID-CCA2, $\mathcal{A}$ is allowed to adaptively choose the challenge identity, ID*, that it wishes to attack.

Canetti, Halevi, and Katz [18, 19] define a weaker notion of security in which the adversary $\mathcal{A}$ commits ahead of time to the challenge identity ID* it will attack. We refer to this notion as *selective identity* adaptively chosen-ciphertext secure IBE (IND-sID-CCA2). In addition, they also define a weaker security notion of IBE, *selective-identity* chosen-plaintext secure IBE (IND-sID-CPA).

7.2 Boneh-Franklin IBE Scheme

The Boneh-Franklin IBE scheme [11] is proven to be secure in the random oracle model (not in the standard model). We now introduce this scheme as a typical example of bilinear-map-based secure cryptosystems in the random oracle model (and as a bench mark to evaluate the efficiency of secure IBE schemes in the standard model).

Setup: Given $(\mathbb{G}, \mathbb{G}_T, p, k$ $(k = |p|)$, a trusted party randomly selects a generator g in $\mathbb{G}$ as well as four hash functions, $H_1, \ldots, H_4$. The trusted party also randomly selects $x \in (\mathbb{Z}/p\mathbb{Z})^*$, and computes $y = g^x$. The system parameter is $(g, y, H_1, \ldots, H_4)$ and the (secret) master key is x.

Extract: Given ID$_A$ of user A, ID$_A$ is mapped (through H_1) to an element of $\mathbb{G}$, h_A, and A's secret key, $s_A = h_A^x$ is computed.

Encrypt: To encrypt a message $m \in \{0,1\}^k$ under ID$_A$, randomly select $\sigma \in \{0,1\}^k$, and compute

$$C = (g^r, \sigma \oplus H_2(e(h_A, y)^r), m \oplus H_4(\sigma)),$$

where $r = H_3(\sigma, m)$.

Decrypt: Let $C = (C_1, C_2, C_3)$ be a ciphertext encrypted using ID$_A$. To decrypt C, compute

$$\sigma = C_2 \oplus H_2(e(s_A, C_1)), \quad \text{and} \quad m = C_3 \oplus H_4(\sigma).$$

Set $r = H_3(\sigma, m)$ and check whether $C_1 = g^r$ holds. If not, rejects the decryption. Otherwise, output m.

Security: The Boneh-Franklin IBE scheme is IND-ID-CCA2 in the random oracle model (i.e., assuming $H_1, \ldots, H_4$ are truly random functions) under the BDH assumption.

7.3 Boneh-Boyen IBE Scheme

There are three Boneh-Boyen IBE schemes that are secure in the standard model (two are in [7] and one is in [9]).

One of the two schemes in [7] is IND-sID-CPA secure, and the other is IND-sID-CCA2 secure. The IND-sID-CCA2 secure scheme [7] is constructed by converting from the IND-sID-CPA secure basic scheme through the conversion technique of [19]. The scheme in [9] is fully secure (IND-ID-CCA2 secure) (through the conversion technique of [19]).

The IND-sID-CPA secure scheme in [7] is much more efficient than the others. Since an IND-sID-CPA secure IBE scheme is sufficient as a building block to construct an IND-CCA2 PKE (Section 8.1), we now introduce the IND-sID-CPA secure IBE scheme in [7] as follows (another reason why we introduce this scheme is that it is closely related to the Boneh-Boyen signature scheme [8] in Section 9.1):

Setup: Given $(\mathbb{G}, \mathbb{G}_T, p, k)$ $(k = |p|)$, a trusted party randomly selects a generator g in $\mathbb{G}$ and $x, y \in (\mathbb{Z}/p\mathbb{Z})^*$, and computes $X = g^x$ and $Y = g^y$. The system parameter is (g, X, Y) and the (secret) master key is (x, y).

Extract: Given $v \in (\mathbb{Z}/p\mathbb{Z})^*$ as ID_A of user A, pick a random $r \in \mathbb{Z}/p\mathbb{Z}$, compute $K = g^{1/(v+x+ry)} \in \mathbb{G}$, and set A's secret key $d_A = (r, K)$.

Encrypt: To encrypt a message $m \in \mathbb{G}_T$ under ID_A (i.e., v), pick a random $s \in \mathbb{Z}/p\mathbb{Z}$ and output the ciphertext

$$C = (g^{sv} X^s, Y^s, e(g, g)^s m).$$

Decrypt: Let $C = (C_1, C_2, C_3)$ be a ciphertext encrypted using ID_A. To decrypt C using $d_A = (r, K)$, compute

$$\frac{C_3}{e(C_1 C_2^r, K)},$$

which is m when C is valid.

For a valid ciphertext we have

$$\begin{aligned} \frac{C_3}{e(C_1 C_2^r, K)} &= \frac{C_3}{e(g^{sv} X^s Y^{sr}, g^{1/(v+x+ry)})} \\ &= \frac{C_3}{e(g^{s(v+x+ry)}, g^{1/(v+x+ry)})} = \frac{C_3}{e(g, g)^s} = m. \end{aligned}$$

Security: The above-mentioned Boneh-Boyen IBE scheme is IND-sID-CPA (selective-identity chosen-plaintext secure) under the Decisional BDHI (DBDHI) assumption.

7.4 Waters IBE Scheme

The Waters IBE scheme [36] is the most efficient IND-ID-CCA2 secure IBE in the standard model. Similarly to the Boneh-Boyen IBE scheme [9], the Waters

IBE scheme is converted from the IND-ID-CPA secure basic scheme (the *Waters basic IBE scheme*) through the conversion technique of [19].

Efficient secure (IND-CCA2) PKE and secure (EUF-CMA) signatures in the standard model are constructed from the Waters basic IBE scheme (Sections 8.2 and 9.2). The Waters basic IBE scheme is as follows:

Setup: Given $(\mathbb{G}, \mathbb{G}_T, p, k)$ $(k = |p|)$, a trusted party randomly selects generators, g and g_2, in $\mathbb{G}$ and $\alpha \in \mathbb{Z}/p\mathbb{Z}$, and computes $g_1 = g^{\alpha}$. Additionally the party randomly selects $u' \in \mathbb{G}$ and k-length vector $(u_1, \ldots, u_k) \in \mathbb{G}^k$, The public parameter is $(g, g_1, g_2, u', u_1, \ldots, u_k)$. The master secret key is g_2^{α}.

Extract: Let v be an k bit string representing an identity ID_A of user A, v_i denote the i-th bit of v, and $\mathcal{V} \subseteq \{1, \ldots, k\}$ be the set of all i for which $v_i = 1$. A's secret key, d_A, for identity v is generated as follows. First, a random $r \in \mathbb{Z}/p\mathbb{Z}$ is chosen. Then the secret key is constructed as:

$$d_A = (g_2^{\alpha}(u' \prod_{j \in \mathcal{V}} u_j)^r, g^r).$$

Encrypt: To encrypt a message $m \in \mathbb{G}_T$ under ID_A (i.e., v), pick a random $s \in (\mathbb{Z}/p\mathbb{Z})^*$ and output the ciphertext

$$C = ((u' \prod_{j \in \mathcal{V}} u_j)^s, g^s, e(g_1, g_2)^s m).$$

Decrypt: Let $C = (C_1, C_2, C_3)$ be a ciphertext encrypted using ID_A (i.e., v). To decrypt C using $d_A = (d_1, d_2)$, compute

$$C_3 \frac{e(d_2, C_1)}{e(d_1, C_2)}$$

which is m when C is valid.

For a valid ciphertext we have

$$C_3 \frac{e(d_2, C_1)}{e(d_1, C_2)} = (e(g,g)^s m) \frac{e(g^r, (u' \prod_{j \in \mathcal{V}} u_j)^s)}{e(g_2^{\alpha}(u' \prod_{j \in \mathcal{V}} u_j)^r, g^s)}$$

$$= (e(g,g)^s m) \frac{e(g, (u' \prod_{j \in \mathcal{V}} u_j))^{rs}}{e(g_1, g_2)^s e((u' \prod_{j \in \mathcal{V}} u_j), g)^{rs}} = m.$$

Security: The Waters basic IBE scheme is IND-ID-CPA under the Decisional BDH (DBDH) assumption.

8 Public-Key Encryption

The desirable security of a public-key encryption (PKE) scheme is formulated as semantic security against adaptively chosen-ciphertext attacks (IND-CCA2) [3].

Although there are several ways to construct practical IND-CCA2 secure PKE schemes in the *random oracle model* [4], only a few practical schemes such as the Cramer-Shoup PKE scheme [20, 22] were proven to be secure in the *standard model.*

Bilinear maps are exploiting a new methodology to design a practical IND-CCA2 secure PKE schemes in the standard model. The new methodology uses transformation from an IBE scheme to a PKE scheme.

8.1 Canetti-Halevi-Katz Construction

Canetti, Halevi and Katz [19] have shown how to construct an IND-CCA2 secure PKE scheme from any IND-sID-CPA secure IBE scheme. In the construction, a one-time signature scheme is also employed. Since this construction is efficient, we can construct an efficient IND-CCA2 secure PKE scheme in the standard model using the Boneh-Boyen IBE scheme [7].

We now show the construction of a PKE scheme as follows:

Key Generation: Run the setup process of IBE to obtain a pair of system parameter and master key. The public key, PK, is the system parameter and the secret key, SK, is the master key.

Encrypt: To encrypt message m using public key PK (IBE's system parameter), the sender first generates a pair of verification key vk and signing key sk of a one-time signature scheme. The sender then computes IBE's ciphertext C of message m with respect to identity vk, and signature σ of C by using signing key sk. The ciphertext is (vk, C, σ).

Decrypt: To decrypt ciphertext (vk, C, σ) using secret key SK (IBE's master key), the receiver first checks whether σ is a valid signature of C with respect verification key vk. If not, the receiver outputs $\perp$. Otherwise, the receiver computes IBE's decryption key d_{vk} for identity vk, and output m decrypted from C by d_{vk}.

Security: If the underlying IBE scheme is IND-sID-CPA and the one-time signature scheme is strongly unforgeable (see [8] for the definition of strong unforgeability) then the Canetti-Halevi-Katz construction of PKE is IND-CCA2.

If the underlying one-time signature scheme is efficient, the Canetti-Halevi-Katz PKE scheme from the Boneh-Boyen IBE scheme [7] is relatively as efficient as (but less efficient than) Cramer-Shoup. The major advantage of this construction over Cramer-Shoup is that the validity of a ciphertext can be verified publicly, while a ciphertext should be verified secretly (i.e., the verification requires the secret key) in Cramer-Shoup. This property is useful in constructing a threshold PKE scheme like [10].

Boneh and Katz [12] improved the Canetti-Halevi-Katz construction by using a message authentication code in place of a one-time signature. The Boneh-Katz construction however is not publicly verifiable.

8.2 Boyen-Mei-Waters PKE Scheme

Boyen, Mei and Waters [14] presented a new way (inspired by [19]) of constructing IND-CCA2 secure PKE schemes in the standard model. Their construction is based on two efficient IBE schemes, the Boneh-Boyen and Waters basic IBE schemes. Unlike the Canetti-Halevi-Katz and Boneh-Katz constructions that use IBE as a black box, the Boyen-Mei-Waters construction directly uses the underlying IBE structure, and requires no cryptographic primitive other than the IBE scheme itself. In addition, the validity of ciphertexts can be checked publicly.

We now introduce the Boyen-Mei-Waters PKE scheme based on the Waters basic IBE scheme.

Key Generation: A user's public/private key pair generation algorithm proceeds as follows. Given $(\mathbb{G}, \mathbb{G}_T, p, k)$ $(k = |p|)$, randomly select a generator g in $\mathbb{G}$ and $\alpha \in \mathbb{Z}/p\mathbb{Z}$, and computes $g_1 = g^\alpha$ and $z = e(g, g_1) = e(g, g)^\alpha$. Next, choose a random $y' \in \mathbb{Z}/p\mathbb{Z}$ and a random k-length vector $(y_1, \ldots, y_n)$, whose elements are chosen at random from $\mathbb{Z}/p\mathbb{Z}$. Then calculate $u' = g^{y'}$ and $u_i = g^y$ for $i = 1$ to k. Finally, an injective encoding $H_0 : \mathbb{G} \times \mathbb{G}_T \rightarrow \{0, 1\}^k$ is chosen. The published public key is

$$(z = e(g, g_1), u' = g^{y'}, u_1 = g^{y_1}, \ldots, u_k = g^{y} \ , H_0),$$

and the private key is

$$(g_1 = g^\alpha, y', y_1, \ldots, y_k).$$

Encrypt: A message $m \in \mathbb{G}_T$ is encrypted as follows. First, a value $s \in \mathbb{Z}/p\mathbb{Z}$ is randomly selected. Then compute $C_2 = g^s$ and $C_3 = z^s m = e(g, g_1)^s m = e(g, g)^{\alpha s} m$. Next, compute $w = H_0(C_2, C_3)$ and $w_1 w_2 \ldots w_k$ denote the binary expansion of w, where each bit $w_i \in \{0, 1\}$. Let $\mathcal{W} \subseteq \{1, \ldots, k\}$ be the set of all i for which $w_i = 1$. Finally compute $C_1 = (u' \prod_{i=1}^k u_i^w \ )^s$. The ciphertext is

$$C = (C_1, C_2, C_3) = ((u' \prod_{j \in \mathcal{W}} u_j)^s, g^s, e(g, g_1)^s m).$$

Decrypt: Given ciphertext $C = (C_1, C_2, C_3)$, first compute $w = H_0(C_2, C_3)$, expressed in binary as $w_1 w_2 \ldots w_k$. Next, compute $w' = y' + \sum_{i=1}^k y_i w_i \bmod p$, and check whether $(C_2)^{w'} = C_1$. If not, output $\perp$. Otherwise, the ciphertext is valid, and decrypt the message as

$$\frac{C_3}{e(C_2, g_1)} = m.$$

The Boyen-Mei-Waters PKE scheme is almost as efficient as the Cramer-Shoup PKE and the variants, and the validity of a ciphertext is publicly verifiable, where the check of $(C_2)^{w'} = C_1$ using private information w' is replaced by the equivalent check with using the bilinear map and public information. Due to the public verifiability, an efficient threshold PKE scheme in the standard model can be constructed on this PKE scheme [10]. Therefore, this scheme is considered to be the most practical PKE scheme in the standard model.

Security: Let H_0 be an efficiently computable injection. Then the Boyen-Mei-Waters PKE scheme is IND-CCA2 under the Decisional BDH (DBDH) assumption.

9 Digital Signatures

The current status on designing secure digital signatures in the standard model is fairly similar to that on designing secure PKE schemes in the standard model.

The desirable security of a digital signature scheme is formulated as existential unforgeability against adaptively chosen-message attacks (EUF-CMA) [26]. Although there are several ways to construct practical EUF-CMA secure signature schemes in the random oracle models [5, 6, 13], only a few practical schemes were proven to be secure in the standard model (the Cramer-Shoup signature scheme etc. [16, 21, 24]).

Similarly to PKE, bilinear maps are exploiting a new methodology to design practical EUF-CMA secure signature schemes in the standard model. There are two ways in the new methodology; one is to directly design (and prove the security of) a signature scheme from bilinear maps (the Boneh-Boyen signature scheme etc. [8, 32, 37]), and the other is to convert an IND-ID-CPA secure IBE scheme to a signature scheme (e.g., the Waters signature scheme [36]).

The Boneh-Boyen signature scheme may be considered to be converted from the Boneh-Boyen IBE scheme [7] in Section 7.3, but it is a bit different from the case of the Waters signature scheme. Since the Waters basic IBE scheme is IND-ID-CPA, the converted signature scheme is EUF-CMA under the same assumption as that for the IBE scheme. On the other hand, since the Boneh-Boyen IBE scheme is IND-sID-CPA, the converted signature scheme is not guaranteed to be EUF-CMA under the same assumption. Actually, the assumption (SDH) for the Boneh-Boyen signature scheme is different from that (DBDHI) for the Boneh-Boyen IBE scheme.

9.1 Boneh-Boyen Signature Scheme

Boneh and Boyen presented a very practical signature scheme that is EUF-CMA secure in the standard model. Signatures in their scheme are much shorter and simpler than the previous secure signature schemes in the standard model.

The Boneh-Boyen signature scheme [8] is as follows:

Key Generation: Given $(\mathbb{G}, \mathbb{G}_T, p, k)$ $(k = |p|)$, randomly select a generator g in $\mathbb{G}$ and $x, y \in (\mathbb{Z}/p\mathbb{Z})^*$, and computes $u = g^x$ and $v = g^y$. The public key is (g, u, v). The secret key is (x, y).

Sign: Given a secret key (x, y) and a message $m \in (\mathbb{Z}/p\mathbb{Z})^*$, pick a random $r \in (\mathbb{Z}/p\mathbb{Z})^*$ and compute

$$\sigma = g^{1/(x+m+yr)}.$$

Here $1/(x + m + yr)$ is computed modulo p. The signature is (σ, r).

Verify: Given a public key (g, u, v), a message $m \in (\mathbb{Z}/p\mathbb{Z})^*$, and a signature (σ, r), verify that

$$e(\sigma, ug^m v^r) = e(g, g).$$

If the equality holds the result is valid; otherwise the result is invalid.

Security: The Bone-Boyen signature scheme is EUF-CMA under the strong DH (SDH) assumption.

9.2 Waters Signature Scheme

The Waters signature scheme is converted from the Waters basic IBE scheme.

Key Generation: Given $(\mathbb{G}, \mathbb{G}_T, p, k)$ $(k = |p|)$, randomly select generators, g and g_2, in $\mathbb{G}$ and $\alpha \in \mathbb{Z}/p\mathbb{Z}$, and compute $g_1 = g^\alpha$. Randomly select $u' \in \mathbb{G}$ and k-length vector $(u_1, \ldots, u_k) \in \mathbb{G}^k$. The public key is $(g, g_1, g_2, u', u_1, \ldots, u_k)$. The secret key is g_2^α.

Sign: Let m be an k-bit message to be signed and m_i denotes the ith bit of m, and $\mathcal{M} \subseteq \{1, \ldots, k\}$ be the set of i for which $m_i = 1$. A signature of m is generated as follows. First, a random r is chosen. Then the signature is constructed as:

$$\sigma = (g_2^\alpha (u' \prod_{j \in \mathcal{M}} u_j)^r, g^r).$$

Verify: Given a public-key $(g, g_1, g_2, u', u_1, \ldots, u_k)$, a message $m \in \{0, 1\}^k$, and a signature $\sigma = (\sigma_1, \sigma_2)$, check

$$\frac{e(\sigma_1, g)}{e(\sigma_2, u' \prod_{j \in \mathcal{M}} u_j)} = e(g_1, g_2).$$

If it holds, the verification result is valid; otherwise the result is invalid.

Security: The Waters signature scheme is EUF-CMA under the Decisional BDH (DBDH) assumption.

10 Concluding Remarks

This paper introduced how bilinear maps are used to design efficient IBE/PKE/ signatures that are provably secure in the standard model. The methodology of using bilinear maps will be applied to more wide areas of secure cryptosystems and protocols. For example, it is applied to more protocol-oriented primitives like group signatures [1, 15], blind signatures [32], threshold PKE [10] and verifiable random functions [23].

Acknowledgements

The author would like to thank Goichiro Hanaoka for his valuable comments on the preliminary manuscript.

References

1. Ateniese, G., Camenisch, J., de Medeiros, B. and Hohenberger, S., Practical Group Signatures without Random Oracles, IACR ePrint Archive, 2005/385, http://eprint.iacr.org/2005/385 (2005)
2. Barreto, P., The Pairing-Based Crypto Lounge, http://paginas.terra.com.br/informatica/paulobarreto/pblounge.html.
3. Bellare, M., Desai, A., Pointcheval, D. and Rogaway, P., Relations Among Notions of Security for Public-Key Encryption Schemes, Adv. in Cryptology – Crypto 1998, LNCS 1462, Springer-Verlag, pp. 26-45 (1998).
4. Bellare, M. and Rogaway, P., Random Oracles are Practical: a Paradigm for Designing Efficient Protocols, Proceedings of the 1st ACM Conference on Computer and Communications Security, CCS 1993, ACM, pp. 62–73 (1993).
5. Bellare, M. and Rogaway, P., The Exact Security of Digital Signatures: How to Sign with RSA and Rabin, Adv. in Cryptology – Eurocrypt 1996, LNCS 1070, Springer-Verlag, pp. 399-416 (1996).
6. Boldyreva, A., Threshold Signature, Multisignature and Blind Signature Schemes Based on the Gap-Diffie-Hellman-Group Signature Scheme, Proceedings of PKC 2003, LNCS 2567, Springer-Verlag, pp.31-46 (2003).
7. Boneh, D. and Boyen, X., Efficient Selective-ID Secure Identity Based Encryption Without Random Oracles, Adv. in Cryptology – Eurocrypt 2004, LNCS 3027, Springer-Verlag, pp. 223-238 (2004).
8. Boneh, D. and Boyen, X., Short Signatures Without Random Oracles, Adv. in Cryptology – Eurocrypt 2004, LNCS 3027, Springer-Verlag, pp. 56–73 (2004).
9. Boneh, D. and Boyen, X., Secure Identity Based Encryption Without Random Oracles, Adv. In Cryptology – Crypto 2004, LNCS 3152, Springer-Verlag, pp. 443–459 (2004).
10. Boneh, D., Boyen, X. and Halevi, S., Chosen Ciphertext Secure Public Key Threshold Encryption Without Random Oracles, to appear in Proceedings of CT-RSA 2006, Springer-Verlag (2006). Available at http://crypto.stanford.edu/˜dabo/abstracts/threshold.html.
11. Boneh, D. and Franklin, M., Identity-Based Encryption from the Weil Pairing, Adv. in Cryptology – Crypto 2001, LNCS 2139, Springer-Verlag, pp.213–229 (2001). Journal version in SIAM Journal of Computing, 32(3), pp. 586–615 (2003).
12. Boneh, D. and Katz, J., Improved Efficiency for CCA-Secure Cryptosystems Built Using Identity Based Encryption, Proceedings of CT-RSA 2005, LNCS 3376, Springer-Verlag, pp.87-103 (2005).
13. Boneh, D., Lynn, B. and Shacham, H., Short Signatures from the Weil Pairing, Adv. in Cryptology – Asiacrypt 2001, LNCS 2248, Springer-Verlag, pp.514–532 (2001).
14. Boyen, X., Mei, Q. and Waters, B., Direct Chosen Ciphertext Security from Identity-Based Techniques, Proceedings of the 12th ACM Conference on Computer and Communications Security, CCS 2005, ACM (2005). Full version available at http://www.cs.stanford.edu/ xb/ccs05/.
15. Boyen, X. and Waters, B., Compact Group Signatures Without Random Oracles, IACR ePrint Archive, 2005/381, http://eprint.iacr.org/2005/381 (2005)
16. Camenisch, J. and Lysyanskaya, A., A Signature Scheme with Efficient Protocols, Security in communication networks, LNCS 2576, Springer-Verlag, pp. 268-289 (2002).

17. Camenisch, J. and Lysyanskaya,A., Signature Schemes and Anonymous Credentials from Bilinear Maps, Adv. In Cryptology – Crypto 2004, LNCS 3152, Springer-Verlag, pp.56–72 (2004)
18. Canetti, R., Halevi, S. and Katz, J., A Forward-Secure Public-Key Encryption Scheme, Adv. in Cryptology – Eurocrypt 2003, LNCS, Springer-Verlag, pp.255-271 (2003). Full version available at http://eprint.iacr.org/2003/083.
19. Canetti, R., Halevi, S. and Katz, J., Chosen-Ciphertext Security from Identity-Based Encryption, Adv. in Cryptology – Eurocrypt 2004, LNCS 3027, Springer-Verlag, pp. 207-222 (2004). Full version available at http://eprint.iacr.org/2003/182.
20. Cramer, R. and Shoup, V., A Practical Public Key Cryptosystem Provably Secure Against Chosen Ciphertext Attack, Adv. in Cryptology – Crypto 1998, LNCS 1462, Springer-Verlag, pp. 13-25 (1998).
21. Cramer, R. and Shoup, V., Signature Schemes Based on the Strong RSA Assumption, ACM TISSEC, 3(3), pp.161–185 (2000). Extended abstract in Proc. 6th ACM CCS (1999).
22. Cramer, R. and Shoup, V., Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption, Adv. in Cryptology – Eurocrypt 2002, LNCS 2332, Springer-Verlag, pp. 45-64 (2002).
23. Dodis, Y. and Yampolskiy, A., A Verifiable Random Function with Short Proofs and Keys, Proceedings of PKC 2005, LNCS 3386, Springer-Verlag, pp.416–431 (2005).
24. Fischlin, M., The Cramer-Shoup Strong-RSA Signature Scheme Revisited, Proceedings of PKC 2003, LNCS 2567, Springer-Verlag, pp.116–129 (2003).
25. Gentry, C. and Silverberg, A., Hierarchical Identity-Based Cryptography, Adv. in Cryptology – Asiacrypt 2002, LNCS 2501, Springer-Verlag, pp. 548-566 (2002).
26. Goldwasser, S., Micali, S. and Rivest, R., A Digital Signature Scheme Secure against Adaptive Chosen-Message Attacks, SIAM J. Computing 17(2): 281-308 (1988).
27. Joux, A., A One Round Protocol for Tripartite Diffie-Hellman, Proceedings of Algorithmic Number Theory Symposium IV, LNCS 1838, Springer-Verlag, pp.385–394 (2000).
Journal version in Journal of Cryptology, 17(4), pp.263–276 (2004).
28. Koblitz, N. and Menezes, A., Pairing-Based Cryptography at High Security Levels, IACR ePrint Archive, 2005/076, http://eprint.iacr.org/2005/076 (2005)
29. Menezes, A., Okamoto, T., and Vanstone, S., Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field, IEEE Transactions on Information Theory 39, pp. 1639–1646 (1993).
30. Miller, V., The Weil Pairing, and its Efficient Calculation, Journal of Cryptology, 17(4) (2004).
31. Miyaji, A., Nakabayashi, M. and Takano, S., New Explicit Conditions of Elliptic Curve Traces for FR-reduction, IEICE Trans. Fundamentals, E84-A(5) (2001).
32. Okamoto,T., Efficient Blind and Partially Blind Signatures Witout Random Oracles, to appear in Proceedings of TCC 2006, LNCS, Springer-Verlag (2006).
33. Sakai, R., Ohgishi, K. and Kasahara, M., Cryptosystems Based on Pairings, In Symposium on Cryptography and Information Security, SCIS 2000, Japan (2000).
34. Shamir, A., Identity-Based Cryptosystems and Signature Schemes, Adv. in Cryptology – Crypto 1984, LNCS 196, Springer-Verlag, pp. 47-53 (1984).

35. Verheul, E., Self-blindable Credential Certificates from the Weil Pairing, Adv. in Cryptology – Asiacrypt 2001, LNCS 2248, pp. 533–551, Springer-Verlag (2002).
36. Waters, B., Efficient Identity-Based Encryption Without Random Oracles, Adv. in Cryptology – Eurocrypt 2005, LNCS 3494, pp. 114-127, Springer-Verlag (2005). Available at http: //eprint.iacr.org/2004/180
37. Zhang, F., Chen, X., Susilo, W. and Mu, Y., A New Short Signature Scheme Without Random Oracles from Bilinear Pairings, IACR ePrint Archive, 2005/386, http://eprint.iacr.org/2005/386 (2005)

The Merit Factor Problem for Binary Sequences

Tom Høholdt

Department of Mathematics, The Technical University of Denmark,
Bldg. 303, DK-2800 Lyngby, Denmark
T.Hoeholdt@mat.dtu.dk

Abstract. Binary sequences with small aperiodic correlations play an important role in many applications ranging from radar to modulation and testing of systems. In 1977, M. Golay introduced the *merit factor* as a measure of the goodness of the sequence and conjectured an upper bound for this. His conjecture is still open. In this paper we survey the known results on the Merit Factor problem and comment on the recent experimental results by R.A.Kristiansen and M. Parker and by P. Borwein,K.-K.S.Choi and J. Jedwab.

1 Introduction

Let $x_0, x_1 \ldots, x_{N-1}$ be a sequence of N elements of value $+1$ or -1. The *aperiodic correlations* are defined by

$$c_k = \sum_{j=0}^{N-k-1} x_j x_{j+k} \qquad k = 1, \ldots, N-1 \tag{1}$$

and the *merit factor* of the sequence introduced by M. Golay [1] is defined by

$$M = \frac{N^2}{2\sum_{k=1}^{N-1} c_k^2} \tag{2}$$

The significance of the merit factor comes from the relation between the merit factor and the spectral properties of the signal corresponding to the sequence. Let

$$Q(e^{i\omega}) = \sum_{j=0}^{N-1} x_j e^{ij\omega} \tag{3}$$

be the *Fourier transform* of the sequence x_j. An easy calculation gives

$$2\sum_{k=1}^{N-1} c_k^2 = \frac{1}{2\pi}\int_0^{2\pi} (\left|Q(e^{i\omega})\right|^2 - N)^2 d\omega. \tag{4}$$

Hence the denominator in the merit factor measures — in terms of power — how much the amplitude spectrum of the signal deviates from the constant value N, and a sequence with maximal merit factor M gives a signal with maximally flat

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 51–59, 2006.

spectrum for a fixed N. The problem is then to find sequences with large merit factors.

We also get from the above expression

$$2\sum_{k=1}^{N-1} c_k^2 + N^2 = \frac{1}{2\pi}\int_0^{2\pi} \left|Q(e^{i\omega})\right|^4 d\omega \tag{5}$$

This means that finding sequences with large merit factor is the same as finding sequences with small L^4 norm. In this setting the problem have been discussed by Littlewood [24] and by Newmann and Byrness [2].

In [14] M. Golay used what he called the "ergodicity postulate" saying that for asymptotic results the aperiodic correlations can be treated as independent random variables to arrive at the formula (see below) for Legendre sequences. He also used this to conjecture that for every length $N > 13$ the merit factor is bounded by 12.32, and that this is the maximal asymptotic value.

For lenghts N up to 40 a complete search was carried out by Lindner [24]. Except for $N = 11$ and $N = 13$, the maximal merit factor is in the interval from 3.3 to 9.85. For $N = 11$ one gets 12.1 and for $N = 13$ the maximum is 14.08. In both cases these sequences are Barker sequences (binary sequences for which all aperiodic corellations are either 0, $+1$, or -1). It is known that Barker sequences do not exist for odd lenghts > 13 and for even lenghts there is a lot of evidence for their nonexistence for $N > 4$, see e.g. Pott and Jungnickel [4].

There are a number of results of partial searches for larger N [1], [8], [5], [6], [7], [19] and [20] but for N larger than 300 these give a merit factor of at most 5.

The survey in [23] contains most of the known results up to 1999.

Recently R. Kristiansen and M.Parker [21] and P.Borwein, K.-K.S. Choi and J. Jedwab [18] constructed sequences by rotation of Legendre sequences and appending approximately a part of it to obtain long seqences with merit factor greater than 6.34.

A new survey by J.Jedwab [22] contains a list of challenges concerning the merit factor problem.

In this paper we will briefly recall what is known and discuss the possibility of proving some of the new experimental results.

2 The Known Results

In this section we recall the known results on the merit factor problem.

2.1 The Merit Factor of a Random Sequence

Theorem 1. *If*

$$x_j, j = 0, \ldots, N-1$$

is a random sequence then

$$E(M) = 1 + \frac{1}{N-1}.$$

Here $E(M)$ denotes the expected value of the merit factor, for a proof see [2] or [23]. This result somehow indicates why it is so difficult to find sequences with large (significantly larger than 1) merit factor.

2.2 A Class of Sequences Where the Merit Factor Can Be Explicitely Calculated

Define a sequence of length $N = 2^m$ recursively by

$$x_0 = 1$$

$$x_{2\ +j} = (-1)^{j+f(i)} x_{2\ -j-1}, 0 \leq j \leq 2^i - 1, i = 0, 1, \ldots, m-1,$$

where f is any function mapping the natural numbers into $\{0, 1\}$.

This was treated in [9] and if $f(0) = f(2k-1) = 0$ and $f(2k) = 1$, $k > 0$, we get the first 2^m elements of the Rudin-Shapiro sequence [10].

If $M(m)$ denotes the meritfactor of the sequence we have

$$M(m) = 3 / \left(1 - \left(-\frac{1}{2}\right)^m\right)$$

so as $m \to \infty$, $M(m) \to 3$.

2.3 A General Method

In [13] we presented a general method for calculation of the merit factor for sequences of odd length. Since this is still the only such method known we recall it. The method uses the discrete Fourier transform of the sequence, i.e.

$$Q(\varepsilon_j) = \sum_{k=0}^{N-1} x_k \varepsilon_j^k$$

where $\varepsilon_j = \exp(j 2\pi i / N)$. A straight forward calculation yields

$$\begin{aligned} |Q(\varepsilon_j)|^2 &= N + c_1(\varepsilon_j + \varepsilon_j^{-1}) + c_2(\varepsilon_j^2 + \varepsilon_j^{-2}) \\ &\quad + \cdots + c_{N-1}(\varepsilon_j^{N-1} + \varepsilon_j^{-(N-1)}) \\ &= N + \sum_{k=1}^{N-1} \Theta_k(\varepsilon_j^k) \end{aligned}$$

and therefore

$$\sum_{j=0}^{N-1} |Q(\varepsilon_j)|^4 = N^3 + 2N \sum_{k=1}^{N-1} c_k c_{N-k} + 2N \sum_{k=1}^{N-1} c_k^2$$

so if N is odd we get

$$2 \sum_{k=1}^{N-1} c_k^2 = \frac{1}{2N} \left(\sum_{j=0}^{N-1} |Q(\varepsilon_j)|^4 + \sum_{j=0}^{N-1} |Q(-\varepsilon_j)|^4 \right) - N^2.$$

We put $S = \sum_{j=0}^{N-1} |Q(\varepsilon_j)|^4 + \sum_{j=0}^{N-1} |Q(-\varepsilon_j)|^4$ and obtain

$$\frac{1}{M} = \frac{S}{2N^3} - 1.$$

One way to treat the second part of the sum S is to use a well-known interpolation formula [15–p. 89] which gives

$$Q(-\varepsilon_j) = \frac{2}{N} \sum_{k=0}^{N-1} \frac{\varepsilon_j}{\varepsilon_j + \varepsilon_k} Q(\varepsilon_k)$$

and therefore

$$\sum_{k=0}^{N-1} |Q(-\varepsilon_j)|^4 = \frac{16}{N^4} \sum_{k=0}^{N-1} \left| \sum_{j=0}^{N-1} \frac{\varepsilon_j}{\varepsilon_j + \varepsilon_k} Q(\varepsilon_j) \right|^4 .$$

It turns out, see [13], that

$$\sum_{k=0}^{N-1} \left| \sum_{j=0}^{N-1} \frac{\varepsilon_j}{\varepsilon_j + \varepsilon_k} Q(\varepsilon_j) \right|^4 = A + B + C + D$$

where

$$A = \frac{1}{16} \left(\frac{1}{3} N^4 + \frac{2}{3} N^2 \right) \sum_{a=0}^{N-1} |Q(\varepsilon_a)|^4$$

$$B = \frac{N^2}{8} \sum_{\substack{a,b=0 \\ a \neq b}} 2|Q(\varepsilon_a)|^2 (\overline{Q}(\varepsilon_a) Q(\varepsilon_b) \varepsilon_b + Q(\varepsilon_a) \overline{Q}(\varepsilon_b) \varepsilon_a) \left(\frac{\varepsilon_a + \varepsilon_b}{(\varepsilon_a - \varepsilon_b)^2} \right)$$

$$C = -\frac{N^2}{4} \sum_{\substack{a,b,c=0 \\ b \neq a \neq c}} 2|Q(\varepsilon_a)|^2 \left(\frac{Q(\varepsilon_b) \overline{Q}(\varepsilon_c) \varepsilon_a \varepsilon_b + \overline{Q}(\varepsilon_b) Q(\varepsilon_c) \varepsilon_a \varepsilon_c}{(\varepsilon_b - \varepsilon_a)(\varepsilon_c - \varepsilon_a)} \right)$$

$$-\frac{N^2}{4} \sum_{\substack{a,b,c=0 \\ b \neq a \neq c}} \frac{Q^2(\varepsilon_a) \overline{Q}(\varepsilon_b) \overline{Q}(\varepsilon_c) \varepsilon_a^2 + \overline{Q}^2(\varepsilon_a) Q(\varepsilon_b) Q(\varepsilon_c) \varepsilon_b \varepsilon_c}{(\varepsilon_b - \varepsilon_a)(\varepsilon_c - \varepsilon_a)}$$

$$D = -\frac{N^2}{2} \cdot \frac{1}{2} \sum_{\substack{a,b=0 \\ a \neq b}} (4|Q(\varepsilon_a)|^2 |Q(\varepsilon_b)|^2 \varepsilon_a \varepsilon_b + Q^2(\varepsilon_b) \overline{Q}^2(\varepsilon_a) \varepsilon_b^2$$

$$+ Q^2(\varepsilon_a) \overline{Q}^2(\varepsilon_b) \varepsilon_a^2) / (\varepsilon_a - \varepsilon_b)^2.$$

This at first hand looks very unpleasent but it turns out that one can estimate these sums in a number of cases, in particular for sequences from difference sets where $|Q(\varepsilon_a)|^2$, $a \neq 0$ is a constant.

2.4 Sequences from Difference Sets

We recall the definition of a cyclic difference set [11]: A set $D = \{i_1, i_2, \ldots, i_k\}$ of k residues modulo v is called a (v, k, λ)-difference set, if the equation $x - y \equiv j \pmod{v}$ has exactly λ solutions $(x, y) \in D \times D$ for each $j \in \{1, 2, \ldots, v-1\}$. If D is a difference set, we construct a binary sequence x_j, $j = 0, \ldots, v-1$ by

$$x_j = \begin{cases} +1 \text{ if } n \in D \\ -1 \text{ if } n \notin D. \end{cases}$$

The periodic correlations, $\Theta_j = \sum_{l=0}^{v-1} x_l x_{l+j}$, where the indices are calculated modulo v, are given by

$$\Theta_j = \begin{cases} v & \text{if } j = 0 \\ v - 4(k - \lambda) & \text{if } j \in \{1, \ldots, v-1\}. \end{cases}$$

Thus the periodic correlations take on only two values. This in fact characterizes the sequences coming from cyclic difference sets [11].

The periodic correlations are connected to the aperiodic correlations by $\Theta_j = C_j + C_{N-j}$, so there is no obvious reason why sequences where the magnitudes of the periodic correlations are small should give sequences with small aperiodic correlations and therefore a large merit factor, but this however turns out to be the case for some sequences from cyclic difference sets and for closely related sequences.

2.5 Legendre Sequences

Let N be an odd prime. A *Legendre sequence* of length N is defined by the Legendre symbols

$$x_j = \left(\frac{j}{N}\right), j = 0, 1, \ldots, N-1$$

which gives

$$x_0 = 1, x_j = \begin{cases} 1 & \text{if } j \text{ is a square} \pmod{N} \\ -1 & \text{if } j \text{ is a nonsquare} \pmod{N}. \end{cases}$$

An "offset" sequence is one in which a function f of its length is chopped off at one end of the sequence and appended at the other, in other words, a cyclic shift of $t = fN$ places. For Legendre sequences it is known that $Q(1) = 1$ and

$$Q(\varepsilon_j) = \begin{cases} 1 + x_j\sqrt{N} & \text{if } N \equiv 1 \pmod{4} \\ 1 + i x_j\sqrt{N} & \text{if } N \equiv 3 \pmod{4} \end{cases}$$

so if $N \equiv 3 \pmod{4}$, $|Q(\varepsilon_j)|$ is independent of $j = 1, 2, \ldots, N-1$ corresponding to the fact that the quadratic residues form a difference set. If we denote by $Q_t(\varepsilon_j)$ the discrete Fourier transform of the sequence cyclic shifted t places we have $Q_t(\varepsilon_j) = \varepsilon_j^{-t} Q(\varepsilon_j)$.

The fact that $|Q(\varepsilon_j)|$ is independent of $j = 1, 2, \ldots, N-1$ in the case $N \equiv 3 \pmod 4$ greatly faciliates the calculation of the sums A,B,C, and D, and it turns out that the case $N \equiv 1 \pmod 4$ can be treated asymptotically in the same manner.

In [13] it was proved that if M is the merit factor for $N \to \infty$ of an offset Legendre sequence corresponding to the fraction f, then

$$\frac{1}{M} = \frac{2}{3} - 4|f| + 8f^2, |f| \le \frac{1}{2}.$$

This gives the highest merit factor $M = 6$ for $|f| = \frac{1}{4}$, a result earlier conjectured by M. Golay [14].

2.6 Maximal Length Shift Register Sequences

Let α be a primitive element of the finite field $F_2\ $ and β a fixed element of $F_2\ $. A *maximal length shift register sequence* — an ML sequence — can be defined as

$$x_j = (-1)^{\mathrm{tr}(\beta\alpha\ )}, j = 0, 1, \ldots, N-1 = 2^m - 2$$

where $\mathrm{tr}(x)$ denotes the trace map from $F_2\ $ to F_2. For the basic facts of these sequences see [16]. It is known that these sequences arise from Singer difference sets with parameters $(2^m - 1, 2^{m-1} - 1, 2^{m-2} - 1)$ and hence it follows that $Q(1) = 1$ and $|Q(\varepsilon_j)|^2 = 2^m = N + 1$, $j = 0, 1, \ldots, N-1$. These two properties alone are not sufficient to determine the merit factor using the method of section 2.3. We have to also use the shift-and-add-property of these sequences. This states [16–Th. 10.6] that there exists a permutation π of $\{1, 2, \ldots, N-1\}$ such that $x_m x_{m+s} = x_{m+\pi(s)}$ for all $m \in \{0, 1, \ldots, N-1\}$ and $s \neq 0$. Moreover this permutation satisfies

$$\alpha^{\pi(s)} = \alpha^s + 1 \text{ for } s \in \{1, \ldots, N-1\}$$

so $\pi(s)$ is what is sometimes called the Zech logarithm. Using this and some character sum estimates we proved in [12] the following result.

Theorem 2. *The asymptotic merit factor of an ML sequence is* 3.

2.7 Jacobi Sequences

Let p and q be different primes and $N = pq$. The Jacobi symbol $\left(\frac{j}{N}\right)$ is defined by

$$\left(\frac{j}{N}\right) = \left(\frac{j}{p}\right)\left(\frac{j}{q}\right)$$

where the factors on the right hand side are Legendre symbols.

A *Jacobi sequence* $Z = z_0 z_1 \ldots z_{N-1}$ of length $N = pq$ is defined by

$$z_l = \left(\frac{l}{N}\right), l = 0, 1, \ldots, N-1.$$

The Jacobi sequences are a special case of a more general construction, namely the product construction of [24] which we recall here.

Definition 1. *Let* $X = x_0, x_1, \ldots, x_{N_1-1}$ *and* $Y = y_0, y_1, \ldots, y_{N_2-1}$ *be two binary sequences with* $\gcd(N_1, N_2) = 1$*. The* product sequence $Z = z_0, z_1, \ldots, z_{N-1}$ *of length* $N = N_1 N_2$ *is defined by*

$$z_l = x_{l_1} y_{l_2}, l = 0, 1, \ldots, N-1,$$

where $l_1 = l \pmod{N_1}$ *and* $l_2 = l \pmod{N_2}$.

The nice properties of a product sequence are that

- 1. The period correlations satisfy

$$\Theta_j(z) = \Theta_{j_1}(x)\Theta_{j_2}(y), j = 0, 1, \ldots, N-1$$

where $j_1 = j \pmod{N_1}$ and $j_2 = j \pmod{N_2}$ and

- 2.

$$Q_z(\varepsilon_j) = Q_x\left(\varepsilon_{tN_2 j_1}\right) Q_y\left(\varepsilon_{sN_1 j_2}\right)$$

where J_1 and J_2 are as above and s, t are integers such that $sN_1 + tN_2 = 1$.

Based on this and the result for the Legendre sequences one can prove [12].

Theorem 3. *Let* $N = pq$*, where* p *and* q *are different primes, goto infinity such that* $(p+q)^5 \log^4 N/N^3 \to 0$*. Then the asymptotic merit factor* M *of the corresponding Jacobi sequence shifted* t *places is given by the formula*

$$\frac{1}{M} = \frac{2}{3} - 4|f| + 8f^2, |f| \leq \frac{1}{2}$$

where $f = t/N$.

This implies that the maximal asymptotic merit factor is 6 as for Legendre sequences.

3 The Recent Results

The idea of Kristiansen and Parker [21] and of Borwein, Choi and Jedwab [18] based on observations by Kirilusha and Narayanaswamy [25] is to take a sequence with known good merit factor and append a fraction of it. If X_r denote a Legendre sequence rotated rN places $(X)^t$ the first tN elements of the sequence (X), where r and t are real numbers in the interval $[0, 1]$ extensive numerical calculations suggest that:

1. For large N the merit factor of the sequence $X_{\frac{1}{4}};(X_{\frac{1}{4}})^t$ is greater than 6.2 when $t \approx 0.03$.
2. For large N the merit factor of $X_r;(X_r)^t$ is greater than 6.34 for $r \approx 0.22$ and $r \approx 0.72$ when $t \approx 0.06$.

The lengths of the considered sequences are 20.000 to 260.000 indicating that there is hope that the maximal asymptotic merit factor is indeed larger than 6.34 (or maybe even 7). To date there is no proof of this but [18] give a relation between the merit factor of the appended sequence and the merit factor of the truncated sequences $(X_r))^t$ and $(X_{r+t})^{1-t})$. One could however hope that a suitable adaption of the method described in section 2.3 eventually will give a proof.

References

1. M.J.E.Golay :Sieves for low autocorrelation binary sequences, IEEE Trans. Inform. Theory, vol. IT-23, no. 1, pp. 43–51, Jan. 1977.
2. D.J.Newman and J.S.Byrnes :The L^4 Norm of a Polynomial with Coefficients ±1, Amer. Math. Monthly, vol. 97, no. 1, pp. 42–45, Jan. 1990.
3. J. Lindner: Binary sequences up to length 40 with best possible autocorrelation function, Electron. Lett., vol. 11, p. 507, Oct. 1975.
4. D.Jungnickel and A.Pott .Perfect and almost perfect sequences. Discrete Applied Math.vol.95 /1 pp.331- 359 July 1999.
5. C.de Groot, D.Würtz, and K.H.Hoffmann :Low autocorrelation binary sequences: Exact enumeration and optimization by evolutionary strategies. Optimization 23, pp. 369–384, 1991.
6. S.Mertens :Exhaustive search for low-autocorrelation binary sequences. J. Phys. A, vol. 29, pp. 473–481, 1996.
7. J. Bernasconi :Low autocorrelation binary sequences: Statistical mechanics and configuration space analysis. J. Phys. vol. 48, pp. 559–567, April 1987.
8. G.F.M.Bencker, T.A.C.M.Claasen and P.W.C.Heimes :‘Binary sequences with a maximally flat amplitude spectrum. Phillips J. Res., vol. 40, no. 5, pp. 289–304, 1985.
9. T.Høholdt, H.Elbrønd Jensen, and J.Justesen :Aperiodic Correlations and the Merit Factor of a class of Binary Sequences. IEEE Trans. Inform. Theory, vol. IT-31, no. 4, pp. 549-552, July 1985.
10. W.Rudin :Some theorems on Fourier coefficients.
Proc. Amer. Math. Soc. vol. 10, pp. 855-859, Dec. 1959.
11. L.D.Baumert: Cyclic Difference Sets Berlin: Springer, Lecture Notes in Mathematics, vol. 189, 1971.
12. J.M.Jensen, H.Elbrønd Jensen, and T. Høholdt :The Merit Factor of Binary Sequences Related to Difference Sets. IEEE Trans. Inform. Theory, vol. 37, no. 3, pp. 617–626, May 1991.
13. Tom Høholdt and Helge Elbrønd Jensen :Determination of the Merit Factor of Legendre Sequences. IEEE Trans. Inform. Theory, vol. 34, no. 1, pp. 161-164, Jan. 1988.
14. M.J.E.Golay :The merit factor of Legendre sequences. IEEE Trans. Inform. Theory, vol. IT-29, no. 6, pp. 934–936, Nov. 1982.
15. G.Polya and G.Szegö : Aufgeben und Lehrsätze aus der Analyse II , Berlin: Springer 1925.
16. R.J.McEliece : Finite Fields for Computer Scientists and Engineers. Boston: Kluwer Academics, 1987.
17. H.D.Lüke: Sequences and arrays with perfect periodic correlation. IEEE Trans. Aerospace Electron. Systems, vol. 24, no. 3, pp. 287–294, May 1988.
18. P.Borwein, K.-K.S.Choi, and J.Jedwab: Binary sequences with merit factor greater than 6.34. IEEE-Trans.Inform. Theory vol 50 pp.3224-3249 2004.
19. P. Borwein, R. Ferguson, and J. Knauer: The merit factor of binary sequences. In preparation
20. J. Knauer: Merit Factor Records. Online Available: ⟨http://www.cecm.sfu.ca/˜jknauer/labs/records.html⟩
21. R.Kristiansen and M.Parker: Binary sequences with merit factor >6.3. IEEE-Trans.Inform. Theory vol.50 pp 3385-3389 2004.

22. J.Jedwab: A Survey of the Merit Factor Problem for Binary Sequences. Preprint December 2004.
23. Tom Høholdt:The Merit Factor of Binary Sequences. In A.Pott et al. editors: Difference Sets, Sequences and Their Correlation Properties, vol. 542 of NATO Science Series C pp.227-237 Kluwer Academic Publishers, Dordrecht 1999.
24. J.E.Littlewood: On polynomials $\sum^n \pm z^m, \sum^n e^{\alpha}\ {}^{i}z^m, z = e^{\theta i}$. J.London Math.Soc vol.41 pp. 367-376 1966
25. A.Kirilusha and G.Narayanaswamy : Construction of new asymptotic classes of binary sequences based on existing asymptotic classes. Summer Science Program Technical Report, Dept. Math.Comput. Science, University of Richmond, July 1999.

Quantum Period Reconstruction of Binary Sequences

Florina Piroi and Arne Winterhof

Johann Radon Institute for Computational and Applied Mathematics,
Austrian Academy of Sciences,
Altenberger Str. 69, A-4040 Linz, Austria
firstname.lastname@oeaw.ac.at

Abstract. We consider the problem of determining the period of a binary sequence. For sequences with small autocorrelation we prove the existence of a polynomial time quantum algorithm for the above problem based on an algorithm of Hales and Hallgren. We apply this result to several concrete examples for which the autocorrelation can be estimated using known bounds on character sums.

Keywords: Period finding, quantum algorithm, binary sequences, autocorrelation, finite fields.

1 Introduction

According to *Kerckhoff's principle*, the security of a cryptosystem shall not be based on keeping the encryption algorithm secret but solely on keeping the encryption key secret. The security of many cryptographic sequences is only based on a secret period. Investigating the vulnerability of the secret key is of great importance for their choice.

We focus on the most important case of binary sequences and consider the problem of recovering the period T of a periodic sequence $\mathcal{S} = (s_n)_{n\geq 0}$ over $\mathbb{F}_2 = \{0, 1\}$ using a quantum algorithm.

Since the mapping $n \mapsto s_n$, $0 \leq n < T$, is not bijective, T cannot be recovered by the well-known algorithm of Shor [9]. Here we show that a result of Hales and Hallgren [4] is quite adequate for our purpose if the given sequence $\mathcal{S}$ has a small autocorrelation, which is an essential feature of cryptographic sequences.

We apply our result to several concrete examples:

- Generalisations of *Legendre sequences*;
- Generalisations of *Sidelnikov sequences*;
- Generalisations of *trace sequences*;
- *Elliptic curve trace sequences.*

As far as the authors are aware of, no classical algorithms are known that tackle with the above problems. We remark, however, that most of the results of this paper can be generalised to nonbinary sequences, see [10].

The main mathematical result of this paper is given in the proof of Theorem 1 in Section 3 and states that if the autocorrelation of a binary sequence is small then its distance from any sequence of smaller period is large.

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 60–67, 2006.

2 Preliminary Results

2.1 Autocorrelation

We recall the definition of the autocorrelation of a periodic binary sequence.

Let $\mathcal{S} = (s_n)_{n\geq 0}$ be a sequence over $\mathbb{F}_2$ and let $T > 1$ be the period of $\mathcal{S}$. The *autocorrelation function* AC of the sequence $\mathcal{S}$ with respect to the shift t is defined by the following relation:

$$\mathrm{AC}(\mathcal{S}, t) = \frac{1}{T}\sum_{n=0}^{T-1}(-1)^{s\ \ +s\ \ +}, \quad 1 \leq t < T.$$

We need the following simple lemma.

Lemma 1. *Let $\mathcal{S} = (s_n)_{n\geq 0}$ be a sequence over $\mathbb{F}_2$, T the period of $\mathcal{S}$, and let $B \geq 0$ be fixed, such that*

$$\max_{1\leq t<T} |\mathrm{AC}(\mathcal{S}, t)| \leq BT^{-1}.$$

For a given t, $1 \leq t < T$, we denote with N_t the cardinality of the set

$$\{s_n \mid s_n = s_{n+t}, \quad 0 \leq n < T\}.$$

Then, for any t, $1 \leq t < T$, we have

$$\left|N_t - \frac{T}{2}\right| \leq \frac{B}{2}.$$

Proof. We clearly have that

$$|2N_t - T| = T \cdot |\mathrm{AC}(\mathcal{S}, t)| \leq B,$$

and the result of the lemma follows immediately.

2.2 Quantum Period Finding Algorithm

Given two periodic sequences $\mathcal{S}^1 = (s_n^1)_{n\geq 0}$ and $\mathcal{S}^2 = (s_n^2)_{n\geq 0}$ with periods T and t, respectively, we denote by $D(\mathcal{S}^1, \mathcal{S}^2)$ the number of integers $n \in [0, Tt-1]$ with $s_n^1 \neq s_n^2$. The following result can be immediately obtained from [4–Theorem 2].

Lemma 2. *For any constant $c > 0$, there is a quantum algorithm which computes in polynomial time, with probability at least $3/4$, the period of any sequence $\mathcal{S}^1$ of period T satisfying*

$$D(\mathcal{S}^1, \mathcal{S}^2) \geq \frac{Tt}{(\log T)^c},$$

for any sequence $\mathcal{S}^2$ of period $t < T$.

3 Reconstruction of the Period

In this section we state and prove the main theorem of this paper. Given a periodic binary sequence, the theorem below gives a condition which, when fulfilled, ensures the existence of a quantum algorithm for the reconstruction of the binary sequence's period.

Theorem 1. *Let $\mathcal{S} = (s_n)_{n\geq 0}$ be a sequence over $\mathbb{F}_2$ and T the period of $\mathcal{S}$, for which*

$$\max_{1\leq t<T} |\mathrm{AC}(\mathcal{S},t)| \leq 1 - \frac{4}{(\log T)^c}$$

for some $c > 0$. Then there exists a quantum algorithm which computes T in polynomial time, with exponentially small probability of failure.

Proof. Let $\mathcal{S}^1 = (s^1_n)_{n\geq 0}$ be a sequence of period $t < T$ and let $\mathcal{K}_t$ be the set

$$\{s_n \mid s_n = s^1_n \text{ and } s_{n+t} = s^1_{n+t}, \quad 0 \leq n \leq Tt - 1\}.$$

Considering the definition of $D(\mathcal{S},\mathcal{S}^1)$ we know that

$$Tt - 2D(\mathcal{S},\mathcal{S}^1) \leq |\mathcal{K}_t|.$$

Also, for each $n \in \mathcal{K}_t$ we can write $s_n = s^1_n = s^1_{n+t} = s_{n+t}$ and thus $s_n = s_{n+t}$. Using the result of Lemma 1 with the bound $B = T(1 - 4(\log T)^{-c})$, we get

$$|\mathcal{K}_t| \leq tN_t \leq \frac{tT}{2}\left(2 - \frac{4}{(\log T)^c}\right).$$

We have now the following sequence of inequalities

$$Tt - 2D(\mathcal{S},\mathcal{S}^1) \leq |\mathcal{K}_t| \leq \frac{tT}{2}\left(2 - \frac{4}{(\log T)^c}\right).$$

From here, we arrive at

$$D(\mathcal{S},\mathcal{S}^1) \geq \frac{Tt}{(\log T)^c}.$$

The result of the theorem follows, then, from the application of Lemma 2.

The above theorem ensures us that a quantum algorithm for computing the period of a binary sequence exists, provided that the maximum autocorrelation of the sequence is small enough. The concrete description of the actual quantum algorithm and a proof of its correctness are not in the scope of this paper. We direct the interested reader to consult [4].

4 Applications

In this section we give some examples how Theorem 1 can be used to give the existence of quantum algorithms for recovering the period of special families of binary sequences. For each sequence in the examples below we give a bound for the maximum autocorrelation of the given sequence and then the condition for the existence of the quantum algorithm. Each of the corollaries formulated below follow immediately from Theorem 1.

4.1 Legendre and Related Sequences

We recall that a *Legendre sequence* $\mathcal{L} = (l_n)_{n\geq 0}$ is defined by

$$l_n = \begin{cases} 1 \text{ if } \left(\frac{n}{p}\right) = -1, \\ 0 \text{ otherwise}, \end{cases} \qquad n \geq 0,$$

where p is an odd prime and $\left(\frac{\div}{p}\right)$ denotes the *Legendre symbol.*

Now, given an odd prime p and a polynomial $f(X)$ over $\mathbb{F}_p$, we define the *generalised Legendre sequence* $\mathcal{L} = (l_n)_{n\geq 0}$ *of period* p*, with the polynomial* $f(X)$ as follows:

$$l_n = \begin{cases} 1, \text{ if } \left(\frac{f(n)}{p}\right) = -1, \\ 0, \text{ otherwise}, \end{cases} \qquad n \geq 0.$$

The following lemma can be immediately proved using Weil's bound for multiplicative character sums; see [6–Theorem 5.41].

Lemma 3. *For a generalised Legendre sequence* $\mathcal{L}$ *with a polynomial* $f(X) \in \mathbb{F}_p[X]$ *and period* p *such that, for any* $1 \leq t < p$*,* $f(X)f(X+t)$ *is not a square we have*

$$\max_{1\leq t<p} |\mathrm{AC}(\mathcal{L}, t)| \leq (2\deg(f) - 1)p^{-1/2} + 2\deg(f)p^{-1}.$$

Proof. Note that

$$(-1)^l \; = \left(\frac{f(n)}{p}\right) \quad \text{if} \quad f(n) \neq 0$$

and we have $f(n) = 0$ or $f(n+t) = 0$ for at most $2\deg(f)$ different n, with $0 \leq n < p$. Hence, for $1 \leq t < p$ and using the multiplicativity of the Legendre symbol, we have

$$p|\mathrm{AC}(\mathcal{L}, t)| \leq \left|\sum_{n=0}^{p-1} \left(\frac{f(n)f(n+t)}{p}\right)\right| + 2\deg(f)$$

and the result follows using the Weil bound.

The above lemma naturally holds for the classical case of Legendre sequences. This can be easily checked by instantiating the polynomial $f(X)$ with $f(X) = X$.

We state now the following existence result.

Corollary 1. *Let* $\mathcal{L} = (l_n)_{n\geq 0}$ *be a generalised Legendre sequence of period* p*, with the polynomial* $f(X) \in \mathbb{F}_p[X]$ *of degree at most*

$$\frac{p^{1/2}}{2}\left(1 - \frac{4}{(\log p)^c}\right)$$

for some $c > 0$*, such that, for any* $1 \leq t < p$*,* $f(X)(f(X+t))$ *is not a square for any* $1 \leq t < p$*. Assume that we are given a black-box that outputs* l_n *for every input integer* n*. Then there exists a quantum algorithm which computes* p *with an exponentially small probability in polynomial time.*

The result in the above corollary is an immediate consequence of Theorem 1.

In the currently available literature, some generalised Legendre sequences for particular polynomials f have been studied. For example, in the case $f(X) = X + s$, where s is a shift, quantum algorithms for finding the period p and the shift s are given in [2]. In the case p is known then $f(X)$ can be recovered in the general case using an algorithm of quantum query complexity $O(\deg(f))$; see [8].

For considerations on the autocorrelation for extensions of Legendre sequences of period q, with q an odd prime power, which are defined over the field $\mathbb{F}_q$ which a special, somewhat natural, ordering of the elements and the quadratic character of $\mathbb{F}_q$, see [7]. For sequences of period pq with two primes p and q see [1, 3].

4.2 Generalised Sidelnikov Sequences

Classically, a Sidelnikov sequence $\mathcal{S} = (s_n)_{n\geq 0}$ is defined by

$$s_n = \begin{cases} 1 \text{ if } \eta(g^n + 1) = -1, \\ 0 \text{ otherwise,} \end{cases} \qquad n \geq 0,$$

where g is a *primitive element* and η denotes the *quadratic character* of the finite field $\mathbb{F}_q$ of odd order q.

Let q be an odd prime power, $f(X)$ a polynomial over the finite field $\mathbb{F}_q$ of q elements, and $g \in \mathbb{F}_q$ an element of order T. Then a *generalised Sidelnikov sequence* $\mathcal{S} = (s_n)_{n\geq 0}$ *of period* T*, with an element* g *of order* T *and a polynomial* f is defined by

$$s_n = \begin{cases} 1, \text{ if } \eta(f(g^n)) = -1, \\ 0, \text{ otherwise,} \end{cases} \qquad n \geq 0,$$

where η is, as before, the quadratic character of the field $\mathbb{F}_q$.

The following result is again based on the Weil bound.

Lemma 4. *Let* $\mathcal{S}$ *be a generalised Sidelnikov sequence of period* T*, with an element* $g \in \mathbb{F}_q$ *of order* T *and a polynomial* $f(X) \in \mathbb{F}_q[X]$ *such that, for any* $1 \leq t < T$*,* $f(X)f(g^tX)$ *is, up to a constant, not a square. Then we have*

$$\max_{1\leq t<T} |\mathrm{AC}(\mathcal{S}, t)| < 2\deg(f)(q^{1/2} + 1)T^{-1}.$$

Proof. The conclusion of the lemma follows immediately from the Weil's theorem. Namely, we have

$$\begin{aligned}
&T|A(\mathcal{S}, t)| \\
&\leq \left|\sum_{n=0}^{T-1} \eta(f(g^n)f(g^tg^n))\right| + 2\deg(f) \\
&\leq \frac{T}{q-1}\left(\left|\sum_{x\in\mathbb{F}} \eta(f(x^{(q-1)/T})f(g^tx^{(q-1)/T}))\right| + 1\right) + 2\deg(f) \\
&\leq \frac{T}{q-1}(2\deg(f)(q-1)/T - 1)q^{1/2} + 2\deg(f) \\
&< 2\deg(f)(q^{1/2} + 1).
\end{aligned}$$

As it was the case for Legendre sequences, Lemma 4 holds also for the classical case of Sidelnikov sequences. In order to check this we have to take $f(X) = X + 1$.

Corollary 2. *Let $\mathcal{S} = (s_n)_{n\geq 0}$ be a generalised Sidelnikov sequence of period T, with $g \in \mathbb{F}_q$ of order T and a polynomial $f(X) \in \mathbb{F}_q[X]$ of degree at most*

$$\frac{T}{2(q^{1/2}+1)}\left(1 - \frac{4}{(\log T)^c}\right)$$

for some $c > 0$, such that, for any $1 \leq t < T$, $f(X)(f(g^t X))$ is, up to a constant, not a square. Assume that we are given a black-box which, for every integer n, outputs s_n. Then there exists a quantum algorithm which computes T in polynomial time with an exponentially small probability of failure.

4.3 Generalised Trace Sequences

Let us now look at generalisations of trace sequences. A *trace sequence* $\mathcal{T} = (t_n)_{n\geq 0}$ is defined by

$$t_n = \mathrm{Tr}\,(g^n), \quad n \geq 0,$$

where g is a primitive element of $\mathbb{F}_2$ and Tr denotes the *absolute trace* of $\mathbb{F}_2$, for some $r \geq 1$.

Let $f(X) \in \mathbb{F}_2\ [X]$, and $g \in \mathbb{F}_2$ an element of order T. We define the *generalised trace sequence $\mathcal{T} = (t_n)_{n\geq 0}$ of period T, with the polynomial f and element g* by

$$t_n = \mathrm{Tr}\,(f(g^n)), \quad n \geq 0.$$

The following result is based on Weil's bound for additive character sums; see, e.g., [6–Theorem 5.38].

Lemma 5. *For any generalised trace sequence $\mathcal{T} = (t_n)_{n\geq 0}$ of period T, with $g \in \mathbb{F}_2$ of order T and any polynomial $f(X) \in \mathbb{F}_2\ [X]$ such that $f(X) + f(g^t X)$ is not of the form $h(X)^2 + h(X) + c$ for any $1 \leq t < T$, we have*

$$\max_{1\leq t<T} |\mathrm{AC}(\mathcal{T}, t)| < \deg(f) 2^{r/2} T^{-1}.$$

Corollary 3. *Let $\mathcal{T} = (t_n)_{n\geq 0}$ be a generalised trace sequence of period T, with $g \in \mathbb{F}_2$ of order T and any polynomial $f(X) \in \mathbb{F}_2\ [X]$ of degree at most*

$$\frac{T}{2^{r/2}}\left(1 - \frac{1}{(\log T)^c}\right),$$

for some $c > 0$ and such that $f(X) + f(g^t X)$ is not of the form $h(X)^2 + h(X) + c$, for any $1 \leq t < T$. Assume that we are given a black-box which, for every integer n, gives t_n. Then there exists a quantum algorithm which computes T in polynomial time with an exponentially small probability of failure.

For some certain cases of trace sequences we can recover the period T also by combining the Berlekamp-Massey and the Shor algorithm. The Berlekamp-Massey algorithm delivers the coefficients $c_0, \ldots, c_L \in \mathbb{F}_2$ of the shortest linear recurrence relation

$$\sum_{l=0}^{L} c_l t_{n+l} = 0, \quad n \geq 0,$$

satisfied by $\mathcal{T}$. For example, if $f(X) = X$ this leads to

$$\mathrm{Tr}\left(g^n \sum_{l=0}^{L} c_l g^l \right) = 0, \quad n \geq 0.$$

We denote the sum above with b. If g is a defining element of $\mathbb{F}_2$, i.e., $\{1, \ldots, g^{r-1}\}$ is a basis of $\mathbb{F}_2$, then by the linearity of the trace $\mathrm{Tr}\,(bg^n) = 0$, $0 \leq n < r$, we know that $\mathrm{Tr}\,(bx) = 0$, $x \in \mathbb{F}_2$, and thus $b = 0$. A root finding algorithm can be used to determine g and, finally, Shor's algorithm can be applied to calculate T.

4.4 Elliptic Curve Trace Sequences

Let E be an elliptic curve over $\mathbb{F}_2$ and P a rational point on E of order T. For a function f in the function field $\mathbb{F}_2$ (E) the sequence $\mathcal{E} = (e_n)_{n \geq 0}$ defined by $e_n = \mathrm{Tr}\,(f(nP))$, $n \geq 0$, has the period T.

The following result follows from [5–Corollary 1].

Lemma 6. *For any function f in the function field $\mathbb{F}_2$ (E) such that $f(nP) - f((n+t)P)$ is not constant for $1 \leq t < T$ and $n \geq 0$, the sequence $\mathcal{E} = (e_n)_{n \geq 0}$ satisfies*

$$\max_{1 \leq t < T} |\mathrm{AC}(\mathcal{E}, t)| \leq 4 \deg(f) 2^{r/2} T^{-1}.$$

For example, the function $f(Q) = x(Q)$, where $x(Q)$ is the first coordinate of $Q = (x(Q), y(Q)) \in E$, satisfies the condition that $s_n - s_{n+t} = f(nP) - f((n+t)P)$ is not constant, for $1 \leq t < T$.

Corollary 4. *Let $\mathcal{E} = (e_n)_{n \geq 0}$ be a sequence of period T defined as in Lemma 6 with* $\deg(f)$ *at most*

$$\frac{T}{4 \cdot 2^{r/2}} \left(1 - \frac{4}{(\log 2^r)^c} \right)$$

for some $c > 0$. Assume that we are given a black-box which for every integer n outputs e_n. Then there exists a quantum algorithm which computes T in polynomial time with an exponentially small probability of failure.

Acknowledgements. A. W. was supported in part by the Austrian Academy of Sciences and by FWF grant S8313. The authors wish to thank Igor Shparlinski for helpful comments.

References

1. N. Brandstätter and A. Winterhof. Some notes on the two-prime generator of order 2. *IEEE Trans. Inform. Theory*, **51**, 3654–3657, 2005.
2. W. van Dam, S. Hallgren and L. Ip. Quantum algorithms for some hidden shift problems. *Proc. of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms, Baltimore, 2003*, 489–498, ACM, New York, 2003.
3. C. Ding. Autocorrelation values of generalized cyclotomic sequences of order two. *IEEE Trans. Inform. Theory*, **44**, 1699–1702, 1998.
4. L. Hales and S. Hallgren. An improved quantum Fourier transform algorithm and applications. *Proc. 41st IEEE Symp. on Found. of Comp. Sci.*, 515–525, 2000.
5. D. R. Kohel and I. E. Shparlinski. Exponential sums and group generators for elliptic curves over finite fields. *Proc. of the 4th International Symposium on Algorithmic Number Theory*, Lecture Notes in Computer Science, **1838**, 395–404, Springer-Verlag, London, UK, 2000.
6. R. Lidl and H. Niederreiter, *Finite Fields*, Cambridge University Press, Cambridge, 1997.
7. W. Meidl and A. Winterhof. On the autocorrelation of cyclotomic generators. *Finite Fields and Applications: 7th International Conference, Fq7, Toulouse, France, May 5-9, 2003*, Lecture Notes in Computer Science, **2948**, 1–11, Springer, Berlin, 2004.
8. A. Russell and I. E. Shparlinski. Classical and quantum function reconstruction via character evaluation. *J. Complexity* **20**, 404–422, 2004.
9. P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comp.*, **26**, 1484–1509, 1997.
10. I. Shparlinski and A. Winterhof. Quantum period reconstruction of noisy sequences. *Proc. ERATO Conf. on Quantum Inform. Sci.*, Tokyo, 2005, 7–8.

The Vector Key Equation and Multisequence Shift Register Synthesis

Li-Ping Wang

Temasek Laboratories, National University of Singapore,
5 Sports Drive 2, Singapore 117508, Republic of Singapore
tslwlp@nus.edu.sg

Abstract. We introduce the vector key equation for cyclic codes with more than one set of consecutive roots. Using the lattice basis reduction multisequence shift register synthesis algorithm [7, 8], we find the minimal solution to the vector equation key and also give an important parameter of the multiple sequences, that is, characteristic sequence. Using it, we give a simpler sufficient and necessary condition for the uniqueness of the minimal polynomial of multiple sequences than that in [7]. The new approach enables us not only to decode cyclic codes up to HT and Roos bounds and but also to find out certain error patterns which can be decoded beyond the two bounds.

1 Introduction

The central computation in decoding alternant codes (including BCH, RS and 1-variable Goppa codes) is to solve the so-called key equation, i.e.,

$$C(x)s(x) \equiv D(x) \bmod x^{-N-1}, \tag{1}$$

where $s(x)$ is the corresponding formal negative power series of the syndrome sequence with length N, $C(x)$, $D(x)$ represent the error-locator and error-evaluator polynomials, respectively, they are relatively prime and $\deg(D(x)) < \deg(C(x)) \leq N$.

In this paper we consider the cyclic codes with more than one set of consecutive roots. The reader is referred to [2] for notations and background.

Let C be a cyclic code of length n over a finite field $\mathbb{F}_q$ generated by a polynomial $g(x)$. Let d_{BCH}, d_{HT} and d_{Roos} denote the BCH bound, the HT bound and Roos bound, respectively, on the minimum distance d of the code. Let β be a primitive nth root of unity in $\mathbb{F}_q$ with g.c.d.$(n, q) = 1$. If $\beta^{b+ir_1+hr_2}$ are roots of $g(x)$ for $i = 1, 2, \ldots, N$, $h = 1, 2, \ldots, m$, where $(r_1, n) = (r_2, n) = 1$, then

$$d \geq d_{\text{HT}} = N + m. \tag{2}$$

When $m = 1$, the HT bound reduces to the BCH bound. It is Hartmann and Tzeng's original generalization of the BCH bound [3].

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 68–75, 2006.

If $\beta^{b+ir_1+j\ \ r_2}$ are roots of $g(x)$ for $i = 1, 2, \ldots, N$, $h = 1, 2, \ldots, m$, where $1 = j_1 < j_2 < \cdots < j_m$, $(j_m - m) < N$, and $(r_1, n) = (r_2, n) = 1$, then

$$d \geq d_{\mathrm{Roos}} = N + m. \tag{3}$$

It is noted that the above Roos bound reduces to the HT bound when $j_h = h$ for $h = 1, 2, \ldots, m$.

Let $c(x)$ be a codeword polynomial, e(x) an error polynomial and $r(x)$ a received polynomial such that $r(x) = c(x)+e(x)$. Let $s_i^{(h)}$ be syndrome defined by

$$s_i^{(h)} = r(\beta^{b+ir_1+j\ \ r_2}) = e(\beta^{b+ir_1+j\ \ r_2}), \text{ for } i = 1, 2, \ldots, N, h = 1, 2, \ldots, m. \tag{4}$$

Our vector key equation has the form

$$C(x)(s^{(1)}(x), s^{(2)}(x), \ldots, s^{(m)}(x)) = (D_1(x), D_2(x), \ldots, D_m(x)) \mod x^{-N-1}. \tag{5}$$

where $C(x)$ is the monic error-locator polynomial, $D_i(x) \in \mathbb{F}_q[x]$, $1 \leq i \leq m$, are the error-evaluator polynomial, and

$$s^{(h)}(x) = \sum_{i=1}^{N} s_i^{(h)} x^{-i}, 1 \leq h \leq m$$

is the formal negative power series of syndrome sequences.

When $m = 1$, it is the famous key equation.

In [2] the authors generalized the famous Berlekamp-Massey algorithm and applied this generalized algorithm to decoding cyclic codes up to HT and Roos bounds. However, the proof about the uniqueness of the error-locator polynomial is very complicated.

Using a lattice basis reduction algorithm, in [7] we proposed a new multisequence shift register synthesis algorithm (LBRMS algorithm) and in [8] we gave its refined algorithm. In this paper we propose a new definition of characteristic sequence of the multiple sequences. By means of the new concept, we give a simpler sufficient and necessary condition for the uniqueness of the minimal polynomial of multiple sequences than that in [7]. The new approach enables us not only to decode cyclic codes up to HT and Roos bounds easily and but also to find out certain error patterns which can be decoded beyond the two bounds. Finally, we give such an example.

2 The Vector Key Equation

In this section our goal is to write the series $s^{(h)}(x)$ described in the Introduction as a quotient of two relatively prime polynomials and so we introduce the vector key equation.

Let t be the number of errors and a_μ, Y_μ for $\mu = 1, 2, \ldots, t$ be the locations and the values of the errors, where $0 \leq a_\mu < n$, $Y_\mu \neq 0$ and $Y_\mu \in \mathbb{F}_q$. Then

$$e(x) = \sum_{\mu=1}^{t} Y_\mu x^{a\ } \ . \tag{6}$$

Let $X_\mu = \beta^{a}$, we have

$$s_i^{(h)} = \sum_{\mu=1}^{t} Y_\mu X_\mu^{b+ir_1+j\ \ r_2}, \text{ for } i = 1, 2, \ldots, N, h = 1, 2, \ldots, m. \tag{7}$$

The problem of decoding cyclic codes is to determine Y_μ's and X_μ's from the multiple syndrome sequences. Let

$$C(x) = \prod_{\mu=1}^{t} (x - X_\mu^{r_1}) = x^t + c_{t-1}x^{t-1} + \cdots + c_1 x + c_0 \tag{8}$$

be the error-locator polynomial. Then

$$\begin{aligned} s_j^{(h)} &+ c_{t-1}s_{j-1}^{(h)} + \cdots + c_0 s_{j-t}^{(h)} = 0 \\ &\text{for } j = t+1, t+2, \ldots, N, \text{ and } h = 1, 2, \ldots, m. \end{aligned} \tag{9}$$

Let $\mathbf{s} = (\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(m)})$ be an m-fold multisequence over a finite field $\mathbb{F}_q$ such that each single sequence $\mathbf{s}^{(h)} = (s_1^{(h)}, s_2^{(h)}, \ldots, s_N^{(h)})$ for $h = 1, 2, \ldots, m$. A polynomial is called a characteristic polynomial of $\mathbf{s}$ if (9) holds. A minimal polynomial of $\mathbf{s}$ is defined by the characteristic polynomial with least degree. Thus $C(x)$ is a minimal polynomial of $\mathbf{s}$ and its degree is called the joint linear complexity of the given m sequences, denoted by $LC(\mathbf{s})$. So the problem of decoding cyclic codes is reduced to finding the minimal polynomial of the syndrome sequences $\mathbf{s}$.

Next we see the expression of $s^{(h)}(x)$. We have

$$\frac{1}{x - X_\mu^{r_1}} = \frac{x^{-1}}{1 - X_\mu^{r_1}x^{-1}} = x^{-1} + X_\mu^{r_1}x^{-2} + X_\mu^{2r_1}x^{-3} + \cdots \tag{10}$$

Multiplying $Y_\mu X_\mu^{b+r_1+j\ \ r_2}$ on (10), then

$$\sum_{\mu=1}^{t} \frac{Y_\mu X_\mu^{b+r_1+j\ \ r_2}}{x - X_\mu^{r_1}} = s_1^{(h)}x^{-1} + s_2^{(h)}x^{-2} + \cdots + s_N^{(h)}x^{-N} + \cdots \tag{11}$$

For $h, h = 1, ..., m$, set

$$D_h(x) = \sum_{\mu=1}^{t} Y_\mu X_\mu^{b+r_1+j\ \ r_2} \prod_{\substack{=\ 1 \\ \neq}}^{t} (x - X_k^{r_1}) \tag{12}$$

Then it is straightforward to obtain the following lemma.

Lemma 1. *For each h, $h = 1, ..., m$, we have*

$$\frac{D_h(x)}{C(x)} = s_1^{(h)}x^{-1} + s_2^{(h)}x^{-2} + \cdots + s_N^{(h)}x^{-N} + \cdots. \tag{13}$$

Theorem 1. *(Vector key equation) For each h, $h = 1, ..., m$, we have*
1) $\deg(D_h(x)) < \deg(C(x))$.
2) g.c.d.$(C(x), D_h(x)) = 1$.
3)

$$Y_\mu = \frac{D_h(X_\mu^{r_1})}{X_\mu^{b+r_1+j\ \ r_2} \prod^t_{\underset{\neq}{=}\ 1} (X_\mu^{r_1} - X_k^{r_1})}, \quad \mu = 1, \ldots, t.$$

4) $C(x)(s^{(1)}(x), s^{(2)}(x), \ldots, s^{(m)}(x)) = (D_1(x), D_2(x), \ldots, D_m(x))$ *mod* x^{-N-1}.

The polynomial $C(x)$ with least degree is called a minimal solution of the vector key equation.

3 The LBRMS Algorithm

In this section we first review the LBRMS algorithm briefly and then give a simpler sufficient and necessary condition about the uniqueness of the minimal polynomial of multiple sequences than that in [7].

We will identify the sequence $\mathbf{s}^{(h)}$ with the formal power series $s^{(h)}(x) = \sum_{i=1}^{\infty} s_i^{(h)} x^{-i}$ for $1 \leq h \leq m$ and so need to introduce the Laurent series field

$$K = \mathbb{F}_q((x^{-1})) = \left\{ \sum_{i=i_0}^{\infty} a_i x^{-i} | \ i_0 \in \mathbb{Z},\ a_i \in \mathbb{F}_q \right\}.$$

Clearly, there is a valuation v on K, that is, for $\alpha = \sum_{i=i_0}^{\infty} a_i x^{-i} \in K$, $v(\alpha) = \min\{i \in \mathbb{Z} | \ a_i \neq 0\}$ if $\alpha \neq 0$, and $v(\alpha) = \infty$ if $\alpha = 0$.

The following two mappings (order function and projection) on the vector space K^{m+1} will be used in the description of the LBRMS algorithm.

$$V : K^{m+1} \to \mathbb{Z} \cup \{\infty\} : \gamma = \ (\alpha_i)_{1 \leq i \leq m+1} \mapsto \begin{cases} \infty, & \text{if } \gamma = \mathbf{0}, \\ \min\{v(\alpha_i) | \, 1 \leq i \leq m+1\}, & \text{otherwise}, \end{cases}$$

$$\theta_k : K^{m+1} \to \mathbb{F}^{m+1} : \gamma = (\alpha_i)_{1 \leq i \leq m+1} \mapsto (a_{1,k}, \ldots, a_{m+1,k})^T, \text{ for } k \in \mathbb{Z},$$

where $\alpha_i = \sum_{j=j_0}^{\infty} a_{i,j} x^{-j}$, $1 \leq i \leq m+1$, T denotes the transpose of a vector.

In the sequel $\theta_{V(\gamma)}(\gamma)$ is often used and simply denoted by $\theta(\gamma)$.

A subset Λ of K^{m+1} is called an $\mathbb{F}_q[x]$-lattice if there exists a basis $\omega_1, \ldots$, ω_{m+1} of K^{m+1} such that

$$\Lambda = \sum_{i=1}^{m+1} \mathbb{F}_q[x]\, \omega_i = \left\{ \sum_{i=1}^{m+1} f_i\, \omega_i \mid f_i \in \mathbb{F}_q[x],\ i = 1, \ldots, m+1 \right\}.$$

In this situation we say that $\omega_1, \ldots, \omega_{m+1}$ form a basis for Λ and we often denote the lattice by $\Lambda(\omega_1, \ldots, \omega_{m+1})$. A basis $\omega_1, \ldots, \omega_{m+1}$ is reduced if $\theta(\omega_1), \ldots,$

$\theta(\omega_{m+1})$ are linearly independent over $\mathbb{F}_q$. The determinant of the lattice is defined by $\det(\Lambda(\omega_1,\ldots,\omega_{m+1})) = v(\det(\omega_1,\ldots,\omega_{m+1}))$.

Next we construct a special lattice $\Lambda(\varepsilon_1,\cdots,\varepsilon_m,\alpha)$ in K^{m+1} spanned by the vectors $\varepsilon_1 = (1,0,\ldots,0),\ldots,\varepsilon_m = (0,\ldots,0,1,0)$, $\alpha = (s^{(1)}(x),\ldots,s^{(m)}(x), x^{-N-1})$.

Let π_i, $i = 1,\ldots,m+1$, denote the ith component of a vector in $\mathbb{F}^{m+1}$, and Γ the set of all characteristic polynomials of $\mathbf{s}$.

The mapping $\eta : \Lambda(\varepsilon_1,\ldots,\varepsilon_m,\alpha) \to \mathbb{F}[x]$ is given by $\gamma = D_1(x)\varepsilon_1 + \cdots + D_m(x)\varepsilon_m + C(x)\alpha \mapsto C(x)$. Put

$$S(\Lambda(\varepsilon_1,\ldots,\varepsilon_m,\alpha)) = \{\gamma \in \Lambda(\varepsilon_1,\ldots,\varepsilon_m,\alpha)|\ \pi_{m+1}(\theta(\gamma)) = 1\}.$$

By Theorem 1, a mapping $\varphi : S(\Lambda(\varepsilon_1,\ldots,\varepsilon_m,\alpha)) \to \Gamma$ is well defined by $\gamma \mapsto \eta(\gamma)$. Likewise, a mapping $\psi : \Gamma \to S(\Lambda(\varepsilon_1,\cdots,\varepsilon_m,\alpha))$ given by $C(x) \mapsto \sigma(C(x))$ is well-defined. It is easy to verify that $\varphi\psi = 1_\Gamma$ and $\psi\varphi = 1_{S(\Lambda(\varepsilon_1,\ldots,\varepsilon\ \ ,\alpha))}$. Hence φ is a bijection. Furthermore, we define two natural total orderings, namely, $S(\Lambda(\varepsilon_1,\ldots,\varepsilon_m,\alpha))$ is ordered by the orders of elements and Γ by the degrees of polynomials. For any two elements γ_1, $\gamma_2 \in S(\Lambda(\varepsilon_1,\ldots,\varepsilon_m,\alpha))$ with $V(\gamma_1) \leq V(\gamma_2)$, we have $\deg(\varphi(\gamma_1)) \geq \deg(\varphi(\gamma_2))$.

Thus we have proved the following theorem.

Theorem 2. *(cf. [7], Theorem 2) The mapping φ is an inverse-order preserving one-to-one correspondence between $S(\Lambda(\varepsilon_1,\ldots,\varepsilon_m,\alpha))$ and Γ.*

So far, the problem about minimal polynomials of the given sequences is reduced to finding an element γ of $S(\Lambda(\varepsilon_1,\ldots,\varepsilon_m,\alpha))$ such that its order is maximum.

By means of a lattice basis reduction algorithm [6], we can transform a basis for the lattice $\Lambda(\varepsilon_1,\cdots,\varepsilon_m,\alpha)$ into a reduced one $\omega_1,\omega_2,\ldots,\omega_{m+1}$.

Especially, in the LBRMS algorithm the reduced basis satisfies the following properties [8].

1. $\theta(\omega_1),\ldots,\theta(\omega_m)$ are linearly independent over $\mathbb{F}$ and $\pi_{m+1}(\theta(\omega_i))$, $i = 1,\ldots, m$, is zero.
2. $\pi_{m+1}(\theta(\omega_{m+1})) = 1$.
3. $\sum_{i=1}^{m+1} V(\omega_i) = \det(\Lambda(\varepsilon_1,\ldots,\varepsilon_m,\alpha)) = N+1$.

Therefore $\eta(\omega_{m+1})$ is a minimal polynomial of $\mathbf{s}$ and $V(\omega_{m+1}) = N+1 - LC(\mathbf{s})$, see [7, 8] for details.

If $\omega_1,\ldots,\omega_m,\omega_{m+1}$ form a reduced basis for any lattice Λ, the set $\{V(\omega_1),\ldots, V(\omega_{m+1})\}$ is completely determined by the lattice and does not depend on the particular choice of reduced basis $\omega_1,\ldots,\omega_m,\omega_{m+1}$ [6]. For the lattice $\Lambda(\varepsilon_1,\ldots, \varepsilon_m,\alpha)$, it means that the set is completely determined by the prescribed multiple sequences and so we give a new definition.

Definition 1. *The sequence $\pi(\mathbf{s}) = \{V(\omega_1),\ldots,V(\omega_m)\}$ is called the characteristic sequence of the multiple sequences $\mathbf{s}$, where $\omega_1,\ldots,\omega_m,\omega_{m+1}$ is a reduced basis for the lattice $\Lambda(\varepsilon_1,\ldots,\varepsilon_m,\ \alpha)$, $V(\omega_1) \geq V(\omega_2) \geq \cdots \geq V(\omega_m)$ and $V(\omega_{m+1}) = N+1-LC(\mathbf{s})$.*

By property 3, we have

Theorem 3. *For any multiple sequences* $\mathbf{s}$*, we have*

$$\sum_{a \in \pi(\mathbf{s})} a = LC(\mathbf{s}).$$

In the following theorem we simplify a sufficient and necessary condition about the uniqueness about minimal polynomials in [7].

Theorem 4. *The minimal polynomial of* $\mathbf{s}$ *is unique if and only if the first element of* $\pi(\mathbf{s})$ *is less than* $N + 1 - LC(\mathbf{s})$*, where* N *is the length of these sequences.*

Proof. Let $\omega_1, \omega_2, \ldots, \omega_{m+1}$ be a reduced basis of $\Lambda(\varepsilon_1, \ldots, \varepsilon_m, \alpha)$ obtained from the LBRMS algorithm satisfying the above three properties. It is easily seen from Theorem 2 that it suffices to show the shortest length vector in $S(\Lambda(\varepsilon_1, \ldots, \varepsilon_m, \alpha))$ is unique if and only if $V(\omega_{m+1}) > V(\omega_i)$, for all $1 \leq i \leq m$. Suppose there exists a vector $\gamma = f_1(x)\omega_1 + \cdots + f_{m+1}\omega_{m+1} \in S(\Lambda(\varepsilon_1, \ldots, \varepsilon_m, \alpha))$ such that $V(\gamma) = V(\omega_{m+1})$. Since the basis $\omega_1, \ldots, \omega_{m+1}$ is reduced, we get $\gamma = f_{m+1}(x) \cdot \omega_{m+1}$. Because of $V(\gamma) = V(\omega_{m+1})$ and $\pi_{m+1}(\theta(\gamma)) = \pi_{m+1}(\theta(\omega_{m+1}) = 1$, we know $f_{m+1}(x) = 1$, i.e., $\gamma = \omega_{m+1}$. Therefore the minimal polynomial is unique. Conversely, suppose that there exists some j, $1 \leq j \leq m$, such that $V(\omega_{m+1}) \leq V(\omega_j)$. Setting $\gamma = x^{V(\omega\ )-V(\omega\ \ +1)}\omega_j + \omega_{m+1}$, then $\gamma \in S(\Lambda(\varepsilon_1, \ldots, \varepsilon_m, \alpha))$ and $V(\gamma) = V(\omega_{m+1})$. But since $\gamma \neq \omega_{m+1}$, then $\eta(\gamma) \neq \eta(\omega_{m+1})$, a contradiction to the uniqueness of minimal polynomial. □

Because of $\pi(\mathbf{s}) = \{LC(\mathbf{s})\}$ when $m = 1$, the following two conditions are equivalent.

1. $LC(\mathbf{s}) < N + 1 - LC(\mathbf{s})$.
2. $2LC(\mathbf{s}) \leq N$.

So the above theorem is the famous condition about the uniqueness of the minimal polynomial in the case of single sequence.

In addition, we give an upper bound about the errors number in decoding such codes using multisequence linear feedback shift register technique.

Corollary 1. *If the minimal polynomial of* $\mathbf{s}$ *is unique, then*

$$LC(\mathbf{s}) < \frac{m}{m+1}(N+1).$$

Proof. By Theorem 4, we have

$$N + 1 - LC(\mathbf{s}) > a, \text{ for any } a \in \pi(\mathbf{s}).$$

By Theorem 3, we get

$$m(N + 1 - LC(\mathbf{s})) > \sum_{a \in \pi(\mathbf{s})} a = LC(\mathbf{s}).$$

and so

$$m(N+1-LC(\mathbf{s})) > LC(\mathbf{s}).$$

Thus we get the desired results. □

Usually we assume $m \leq N$, so we have

Corollary 2.

$$\frac{2m}{m+1}(N+1) > N+m-1.$$

Proof. If we want to have this result, then we need to show that

$$\begin{aligned}
&\Leftarrow 2m(N+1) > (m+1)(N+m-1)\\
&\Leftarrow (m-1)N > (m+1)(m-1) - 2m\\
&\Leftarrow N > m-1-\frac{2}{m-1}\\
&\Leftarrow N \geq m-1.
\end{aligned}$$

So the result holds. □

From Corollary 2, this upper bound is absolutely greater than the HT bound and the Roos bound, which makes it possible to decode the cyclic codes beyond the two bounds.

4 Decoding BCH Codes Using Multiple Syndrome Sequences

In [2] the authors proved the uniqueness about the error-locator polynomial up to HT and Roos bounds with additional requirement for the latter by matrix theory. In this section we simply solve the uniqueness of decoding problem by a new viewpoint. However, we have to add the condition, i.e. $V(\omega_i) > 0$ for $1 \leq i \leq m$.

Theorem 5. *If $2t \leq N+m-1$, where t is the number of errors and $V(\omega_i) > 0$ for $1 \leq i \leq m$, then the minimal polynomial of the syndrome sequences $\mathbf{s}$ is unique.*

Proof. Let $\omega_1, \omega_2, \ldots, \omega_{m+1}$ be a reduced basis obtained by the LBRMS algorithm. Because of $t = LC(\mathbf{s})$ and $V(\omega_{m+1}) = N+1-t \geq t-m+2$, $V(\omega_i) \leq t-m+1$ for $1 \leq i \leq m$, by Theorem 4 we have the conclusion. □

Finally, we give a special kind of error patterns which can be decoded beyond the above two bounds. From Theorem 4, it is straightforward that an error pattern can be decoded beyond HT and Roos bounds if the characteristic sequence of its syndrome sequences satisfies the condition in Theorem 4.

The following is such an example.

Example 1. Let $n = 31$, $g(x) = m_1(x)m_3(x)m_5(x)m_{11}(x)$. We have two consecutive roots $\beta, \beta^2, \ldots, \beta^6$, and $\beta^8, \beta^9, \ldots, \beta^{13}$, where β is a primitive element in $\mathbb{F}_{2^5}$ such that $\beta^5 + \beta^2 + 1 = 0$. Here the BCH bound is 7, the HT bound is 8, and the actual minimum distance is 11.

Assume we obtain two syndrome sequences

$$\mathbf{s}^{(\mathbf{1})} = (\beta^{12}, \beta^{24}, \beta^{20}, \beta^{17}, \beta, \beta^9)$$
$$\mathbf{s}^{(\mathbf{2})} = (\beta^3, \beta^8, \beta^2, \beta^{13}, \beta^{18}, \beta^{21})$$

Using the refined LBRMS algorithm, we have $V(\omega_1) = 2$, $V(\omega_2) = 2$, and $\omega_3 = 3$. So we obtain a unique minimal polynomial $C(x) = x^4 + \beta^{12}x^3 + \beta^{21}x^2 + \beta^8 x + \beta = (x-1)(x-\beta)(x-\beta^3)(x-\beta^5)$. Hence the error polynomial is

$$e(x) = 1 + x + x^3 + x^5.$$

So we can correct 4 errors beyond HT and BCH bounds.

Acknowledgement

The author would like to thank Professor Harald Neiderreiter for his valuable comments and suggestions.

References

[1] E. R. Berlekamp, Algebraic Coding Theory. New York, McGrawHill, 1969.

[2] G. L. Feng and K. K. Tzeng, A generalization of the Berlekamp-Massey algorithm for multisequence shift-register synthesis with applications to decoding cyclic codes, IEEE Trans. Inform. Theory, vol. IT-37 (1991) 1274-1287.

[3] C. R. P. Hartmann and K. K. Tzeng, Generalizations of the BCH bound, Inform. Contr., vol. 20, no. 5 (1972) 489-498.

[4] J. L. Massey, Shift-register synthesis and BCH decoding. IEEE Trans. Inform. Theory, vol. IT-15(1)(1969) 122-127.

[5] G. H. Norton, Some decoding applications of minimal realization, In Cryptography and Coding, LNCS 1025 (1995) 53-62.

[6] W. M. Schmidt, Construction and estimation of bases in function fields, J. Number Theory 39, no. 2 (1991) 181-224.

[7] L.-P. Wang, Y.-F. Zhu, $F[x]$-lattice basis reduction algorithm and multisequence synthesis, Science in China (series F), vol. 44 (2001) 321-328.

[8] L.-P. Wang, Y.-F. Zhu and D.-Y. Pei, On the lattice basis reduction multisequence synthesis algorithm, IEEE Trans. Inform. Theory, vol. 50, No. 11 (2004) 2905-2910.

A General Framework for Applying FGLM Techniques to Linear Codes

M. Borges-Quintana[1,⋆], M.A. Borges-Trenard[1], and E. Martínez-Moro[2,⋆⋆]

[1] Dpto. de Matemática, FCMC, U. de Oriente, Santiago de Cuba, Cuba
mijail@mbq.uo.edu.cu, mborges@mabt.uo.edu.cu
[2] Dpto. de Matemática Aplicada, U. de Valladolid, Spain
edgar@maf.uva.es

Abstract. We show herein that a pattern based on FGLM techniques can be used for computing Gröbner bases, or related structures, associated to linear codes. This Gröbner bases setting turns out to be strongly related to the combinatorics of the codes.

1 Introduction

It is well known that the complexity of Gröbner bases computation heavily depends on the term orderings, moreover, elimination orderings often yield a greater complexity. This remark led to the so called FGLM convertion problem, i.e., **given** a Gröbner basis with respect to a certain term ordering, **find** a Gröbner basis of the same ideal with respect to another term ordering. One of the efficient approaches for solving this problem, in the zero-dimensional case, is the FGLM algorithm (see [11]).

The key ideas of this algorithm were successfully generalized in [12] with the objective of computing Gröbner bases of zero-dimensional ideals that are determined by functionals. In fact, the pioneer work of FGLM and [12] was the Buchberger-Möller's paper (cf. [9]). Authors of [1] used the approach of [9] and some ideas of [11] for an efficient algorithm to zero-dimensional schemes in both affine and projective spaces. In [4] similar ideas of using a generalized FGLM algorithm as a pattern algorithm were presented in order to compute Gröbner basis of ideals of free finitely generated algebras. In particular, it is introduced the pattern algorithm for monoid and group algebras . In [3, 13] a more general pattern algorithm which works on modules is introduced, many things behind of this idea of using linear algebra are formalized, notions like "Gröbner technology" and "Gröbner representations" are used. There are other approches which also generalized similar ideas to some settings, behind of all these works is the essential fact of using linear algebra techniques to compute in "Gröbner bases schemes".

⋆ This author is partially supported by Spanish MEC grant SB 2003-0287.
⋆⋆ Partially supported by Spanish MEC MTM2004-00876 and MTM2004-00958.

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 76–86, 2006.

The main goal of this paper is to show the application of techniques to linear codes like the ones in FGLM and subsequent works, which comes from an specification of the pattern algorithm for monoid algebras given in [4], i.e. by taking an algebra associated to a linear code.

2 Preliminaries

The case of the algebra associated to a linear code we are going to introduce is connected with an ideal of a free commutative algebra; therefore, we will restric ourselves to the formulation of a pattern algorithm for a free commutative algebra. Similar settings can be performed in a free associated algebra or over modules (see [4, 3, 13]).

Let $X := \{x_1, \ldots, x_n\}$ be a finite set of variables, $[X]$ the free commutative monoid on X, K a field, I an ideal of $K[X]$, $I(F)$ the ideal of $K[X]$ generated by $F \subset K[X]$, $K[X]/I$ the residue class algebra of $K[X]$ module I. Let us denote by 1 the empty word in $[X]$, $L(u)$ the length of the word $u \in [X]$, and $Card(C)$ the cardinal of the set C. Let now $\prec$ be a semigroup total well ordering on $[X]$ (such an ordering is also called admissible), then for $f \in K[X] \setminus \{0\}$, $T_\prec(f)$ is the maximal term of f with respect to $\prec$, $LC_\prec(f)$ is the leading coefficient of f with respect to $\prec$. Similarly, for $F \subset K[X]$, $T_\prec\{F\}$ is the set of maximal terms of non-zero polynomials in F, $T_\prec(F)$ is the semigroup ideal generated by $T\{F\}$. Moreover, for the sake of simplicity in notation, $U_\prec(F)$ will be used instead of $U(T_\prec(F))$, where U lies in $\{G, N, B, I\}$. Of course, given an ideal I and two different admissible orderings $\prec_1$ and $\prec_2$, in general we have $U(T_{\prec_1}(I)) \neq U(T_{\prec_2}(I))$. Notwithstanding this strong dependency on $\prec$, while a single admissible ordering $\prec$ is considered, so that no confusion arise, we will often simply write $U(F)$ for $U_\prec(F)$.

Let $\tau \subset [X]$ be a semigroup ideal of $[X]$, i.e., for $u \in [X]$ and $t \in \tau$, $tu \in \tau$. Then, it is well known that τ has a unique subset $G(\tau)$ of irredundant generators (probably infinite). In the case of I a zero-dimensional ideal, for $\tau = T(I)$, $G(\tau)$ is always finite. We are going to introduce for τ some notation and terminology, which are similar to those introduced in [12].

$Pred(w) := \{u \in [X] \mid \exists\, x \in X\, (w = ux)\}$ (the set of *predecessors of* w),
$N(\tau) := \{s \in [X] \mid s \notin \tau\}$ (*outside of* τ),
$B(\tau) := \{w \in \tau \mid Pred(w) \subset N(\tau)\}$ (*border of* τ),
$I(\tau) := \tau \setminus B(\tau)$ (*interior of* τ).

We remark that $w \in \tau$ lies in $G(\tau)$ if and only if all its proper divisors are in $N(\tau)$ (that is if $Pred(w) \subset N(\tau)$). In the following proposition, some basic results concerning τ and its regions are summarized. Although they are very easy to prove, their importance is crucial for FGLM techniques.

Proposition 1 (Properties of the semigroup ideal regions).

i. For each $w \in \tau$ *there exist* $u \in [X]$ *and* $v \in B(\tau)$ *s.t.* $w = vu$.
ii. For $x \in X$*:*

  (a) If $u \in N(\tau)$, then $ux \in N(\tau) \cup B(\tau)$.
  (b) If $u \in B(\tau)$, then $ux \in B(\tau) \cup I(\tau)$.
  (c) If $u \in I(\tau)$, then $ux \in I(\tau)$.

iii. $N(\tau), N(\tau) \cup G(\tau), N(\tau) \cup B(\tau)$ are order ideals, i.e., if u belongs to one of these subsets and v divides u, then v also belongs to the corresponding sets.

Theorem 1 (The vector space of canonical forms modulo an ideal). *Let $Span_K(N_{\prec}(I))$ be the K-vector space whose basis is $N_{\prec}(I)$. Then the following holds:*

i. $K\langle X\rangle = I \oplus Span_K(N_{\prec}(I))$ (this sum is considered as a direct sum of vector spaces).
ii. For each $f \in K[X]$ there is a unique polynomial of $Span_K(N_{\prec}(I))$, denoted by $Can(f, I, \prec)$ such that $f - Can(f, I, \prec) \in I$; moreover:
  (a) $Can(f, I, \prec) = Can(g, I, \prec)$ if and only if $f - g \in I$.
  (b) $Can(f, I, \prec) = 0$ if and only if $f \in I$.
iii. There is a K-vector space isomorphism between $K[X]/I$ and $Span_K(N_{\prec}(I))$ (the isomorphism associates the class of f modulo I with the canonical form $Can(f, I, \prec)$ of f modulo I).

$Can(f, I, \prec)$ is called the canonical form of f modulo I. We use simply $Can(f, I)$ if the ordering used is clear from the context.

We assume the reader to be familiar with definition and properties of Gröbner bases (see [2] for an easy to read introduction to Gröbner bases).

Proposition 2 (Characterization of zero-dimensional ideals). *Let G be a Gröbner basis of I with respect to $\prec$. Then, I is a zero-dimensional ideal (i.e. $dim_K K[X]/I < \infty$) if and only if $N_{\prec}(G)$ is finite. Moreover, in such a case, $dim_K K[X]/I = Card(N_{\prec}(G))$.*

Definition 1 (Border basis). *The border basis of I with respect to $\prec$ is the subset $\mathcal{B}(I, \prec) \subset I$ defined by:*

$$\mathcal{B}(I, \prec) := \{w - Can(w, I, \prec) \mid w \in B_{\prec}(I)\} \quad \textit{(the } \mathcal{B}\textit{-basis of } I).$$

Note that the $\mathcal{B}$-basis of I is a Gröbner basis of I that contains the reduced Gröbner basis.

2.1 Matphi Matrices and Gröbner Representation

The word Matphi appears by the first time in [11] to denote a procedure that computes a set of matrices (called matphi matrices) s.t. there is one matrix for each variable in X and they describe the multiplication structure of the quotient algebra $K[X]/I$, where I is a zero dimensional ideal. We often refer to this set of matrices as the matphi structure.

Definition 2 (Gröbner representation, Matphi structure). *Let I be a zero-dimensional ideal of $K[X]$, let $s = dim(K[X]/I)$. A Gröbner representation of I is a pair (N, ϕ) consisting of*

i. $N = \{N_1, \ldots, N_s\}$ *s.t.* $K[X]/I = Span_K(N)$, *and*
ii. $\phi := \{\phi(k) \mid 1 \leq k \leq n\}$, *where* $\phi(k)$ *are the square matrices* $\phi(k) := (a^k_{ij})_{ij}$ *s.t. for all* $1 \leq i \leq s$, $N_i x_k \equiv_I \sum_j a^k_{ij} N_j$.

ϕ *is called the matphi structure and the* $\phi(k)$*'s the matphi matrices.*

See [3, 13] for a more general treatment of these concepts. Note that the matphi structure is indepent of the particular set N of representative elements of the quotient $K[X]/I$. In addition, the matphi matrices allow to obtain the class of any product of the form $N_i x_k$ as a combination of the representative elements (i.e. as a linear combination of the basis N for the vector space $K[X]/I$).

3 The FGLM Pattern Algorithm

In this section we present a generalization of the FGLM algorithm for free commutative algebras, which allows to solve many different problems and not only the clasic FGLM convertion problem. The procedure we are presenting is based on a sort of black box pattern: in fact, the description of the steps 5 and 6 is only made in terms of their input and output. More precisely, we are assuming that a term ordering $\prec_1$ is fixed on $[X]$, I is a zero-dimensional ideal (without this restriction the algorithm does not terminate), and that the K-vector space $Span_K(N_{\prec_1}(I))$ is represented by giving

- a K-vector space E which is endowed of an *effective* function

  $$\textbf{LinearDependency}[v, \{v_1, \ldots, v_r\}]$$

  which, for each finite set $\{v_1, \ldots, v_r\} \subset E$ of linearly independent vectors and for each vector $v \in E$, returns the value defined by

  $$\begin{cases} \{\lambda_1, \ldots, \lambda_r\} \subset K & \text{if } v = \sum_{i=1}^r \lambda_i v_i, \\ \textbf{False} & \text{if } v \text{ is not a linear combination of } \{v_1, \ldots, v_r\}. \end{cases}$$

- an injective morphism $\xi : Span_K(N_{\prec_1}(I)) \mapsto E$.

This informal approach allows a free choice of a suitable representation of the space $Span_K(N_{\prec_1}(I))$ regarding an efficient implementation of these techniques and a better complexity. Moreover, as an aside effect, it enables us to present this generalization in such a way that it can be applied on several more particular patterns and helps to make key ideas behind the FGLM algorithm easier to understand. Let us start making some references to some subroutines of the algorithm.

InsertNexts$[w, List, \prec]$ inserts properly the products wx (for $x \in X$) in *List* and sorts it by increasing ordering with respect to the ordering $\prec$. The reader should remark that **InsertNexts** could count the number of times that an element w is inserted in *List*, so $w \in N_\prec(I) \cup T_\prec\{G\}$ if and only if this number coincide with the number of variables in the support of w, otherwise, it means that $w \in T_\prec(I) \setminus T_\prec\{G\}$, see [11], this criteria can be used to know the boolean value of the test condition in Step 4 of the Algorithm 1.

NextTerm[*List*] removes the first element from *List* and returns it.

Algorithm 1 (FGLM pattern algorithm)
Input: $\prec_2$*, a term ordering on* $[X]$*;* $\xi : Span_K(N_{\prec_1}(I)) \mapsto E$*.*
Output: *rGb*$(I, \prec_2)$*, the reduced Gröbner basis of* I *w.r.t. the ordering* $\prec_2$*.*

```
1. $G := \emptyset$; $List := \{1\}$; $N := \emptyset$; $r := 0$;
2. While $List \neq \emptyset$ do
3.   $w :=$ NextTerm$[List]$;
4.   If $w \notin T_{\prec_2}(G)$ (if $w$ is not a multiple of any element in $G$) then
5.      $v := \xi(Can(w, I, \prec_1))$;
6.      $\Lambda :=$ LinearDependency$[v, \{v_1, \ldots, v_r\}]$;
7.      If $\Lambda \neq$ False then $G := G \cup \{w - \sum_{i=1}^{r} \lambda_i w_i\}$ (where $\Lambda = (\lambda_1, \ldots, \lambda_r)$)
8.                           else $r := r + 1$;
9.                                $v_r := v$;
10.                               $w_r := w$; $N := N \cup \{w_r\}$;
11.                               $List :=$ InsertNexts$[w_r, List, \prec_2]$;
12. Return$[G]$.
```

Remark 1.

i. A key idea in algorithms like FGLM is to use the relationship between membership to an ideal I and linear dependency modulo I, namely $\forall\ c_i \in K, s_i \in K[X]$ we have $\sum_{i=1}^{r} c_i s_i \in I \setminus \{0\} \iff \{s_1, \ldots, s_r\}$ is linearly dependent modulo I. This connection with linear algebra was used for the firts time in Gröbner bases theory since the very begining (see [8]).

ii. Since each element of $N_{\prec_2}(I) \cup B_{\prec_2}(I)$ belongs to *List* at some moments of the algorithm and $List \subset N_{\prec_2}(I) \cup B_{\prec_2}(I)$ at each iteration of the algorithm, it is clear that one can compute $\mathcal{B}(I, \prec_2)$ or the Gröbner representation $(N_{\prec_2}(I), \phi)$ of I just by eliminating Step 4 of the algorithm and doing from Step 5 to Step 11 with very little changes in order to built those structures instead of *rGb*$(I, \prec_2)$.

iii. Note that Step 5 and 6 depends on the particular setting. In Step 5 it is necessary to have a way of computing $Can(w, I, \prec_1)$ and the corresponding element in E, while in Step 6 we need an effective method to decide linear dependency.

iv. Complexity analysis of this pattern algorithm can be found in [4] for the more general case of free associative algebras, and for a more general setting in [3, 13]. Of course, having a pattern algorithm as a model, it is expected that for particular applications, one could do modification and specification of the steps in order to improve the speed and decrease the complexity of the algorithm by taking advantage of the particular structures involved.

3.1 The Change of Orderings: A Particular Case

Suppose we have an initial ordering $\prec_1$ and the reduced Gröbner basis of I for this ordering, now we want to compute by the FGLM algorithm the new reduced Gröbner basis for a new ordering $\prec_2$. Then the vector space E is K^s, where $s = dim(K[X]/I)$. In Step 5, $Can(w, I, \prec_1)$ can be computed using the reduced Gröbner basis $rGb(I, \prec_1)$ and the coefficients of this canonical form build the vector of E corresponding to this element (the image by the morphism ξ). Then Step 6 is perfomed using pure linear algebra.

4 FGLM Algorithm for Monoid Rings

The pattern algorithm is presented in [4] for the free monoid algebra, we will restrict here to the commutative case. Let M be a finite commutative monoid generated by $g_1, \ldots, g_n$; $\xi : [X] \to M$, the canonical morphism that sends x_i to g_i; $\sigma \subset [X] \times [X]$, a presentation of M defined by ξ ($\sigma = \{(w, v) \mid \xi(w) = \xi(v)\}$). Then, it is known that the monoid ring $K[M]$ is isomorphic to $K[X]/I(\sigma)$, where $I(\sigma)$ is the ideal generated by $P(\sigma) = \{w - v \mid (w, v) \in \sigma\}$; moreover, any Gröbner basis G of $I(\sigma)$ is also formed by binomials of the above form. In addition, it can be proved that $\{(w, v) \mid w - v \in G\}$ is another presentation of M.

Note that M is finite if and only if $I = I(\sigma)$ is zero-dimensional. We will show that in order to compute $rGb(I)$, the border basis or the Gröbner representation of I, one only needs to have M given by a concrete representation that allows the user to multiply words on its generators; for instance: M may be given by permutations, matrices over a finite field, or by a more abstract way (a complete or convergent presentation). Accordingly, we are going to do the necessary modifications on Algorithm 1 for this case.

We should remark that in this case $\prec_1 = \prec_2$, then at the begining of the algorithm the set $N_{\prec_1}(I)$ is unkown (which is not the case of the change of orderings). It could be precisely a goal of the algorithm to compute a set of representative elements for the quotient algebra.

Now consider the natural extension of ξ to an algebra morphism ($\xi : K[X] \mapsto K[M]$), note that the restriction of ξ to $Span_K(N_{\prec_1}(I))$ ($\xi : Span_K(N_{\prec_1}(I)) \mapsto K[M]$) is an injective morphism; moreover, $\xi(w) = \xi(Can(w, I, \prec_1))$, for all $w \in [X]$. Therefore, the image of $Can(w, I, \prec_1)$ can be computed as $\xi(w)$, and the linear dependency checking will find out whether w is a new canonical form (i.e. $w \in N_{\prec_1}(I)$) or not (i.e. $w \in T_{\prec_1}(rGb(I, \prec_1))$). Hence, Step 5 will be

$$v := \xi(u) g_i, \text{ where } u \in Pred(w) \text{ and } u x_i = w.$$

Moreover, let $w_1, \ldots, w_r$ be elements of $N_{\prec_1}(I)$ and $v_i = \xi(w_i)$, for $1 \leq i \leq r$. Then **LinearDependency**$[v, \{v_1, \ldots, v_r\}]$ can be computed as

$$\begin{cases} v_j & \text{if } v = v_j, \text{ for some } j \in [1, r], \\ \textbf{False} & \text{otherwise.} \end{cases}$$

Finally, Step 7 changes into **If** $\Lambda \neq$ **False** **then** $G := G \cup \{w - w_j\}$.

Remark 2.

i. This example shows that the capability of the K-vector space E w.r.t. **LinearDependency**, that is demanded in the Algorithm 1, is required only on those sets of vectors $\{v_1, \ldots, v_r, v\}$ that are built in the algorithm, which means in this case that **LinearDependency** is reduced to the **Member** checking, i.e., v is linear dependent of $\{v_1, \ldots, v_r\}$ if and only if it belongs to this set.

ii. When a word w is analyzed by the algorithm, all the elements in $Pred(w)$ have been already analyzed ($\xi(u)$ is known for any $u \in Pred(w)$), this is the case whenever $\prec_1$ is an admissible ordering. Therefore, the computation of $\xi(w)$ is immediate.

We will show the case of linear codes as a concrete setting for an application of the FGLM pattern algorithm for monoid rings, where the monoid is given by a set of generators and a way of multiply them.

5 FGLM Algorithm for Linear Codes

For the sake of simplicity we will stay in the case of binary linear codes, where more powerfull structures for applications are obtainned as an output of the corresponding FGLM algorithm (for a general setting see [5, 7]). From now on we will refer to linear codes simply as codes.

Let $\mathbb{F}_2$ be the finite field with 2 elements. Let $\mathcal{C}$ be a binary code of dimension k and length n ($k \leq n$), so that the $n \times (n-k)$ matrix H is a *parity check matrix* ($c \cdot H = 0$ if and only if $c \in \mathcal{C}$). Let d be the minimum distance of the code, and t the error-correcting capability of the code ($t = \left[\frac{d-1}{2}\right]$, where $[x]$ denotes the greater integer less than x). Let $B(\mathcal{C}, t) = \{y \in \mathbb{F}_2^n \mid \exists\, c \in \mathcal{C}\ (d(c, y) \leq t)\}$, it is well known that the equation $eH = yH$ has a unique solution e with weight$(e) \leq t$, for $y \in B(\mathcal{C}, t)$.

Let us consider the free commutative monoid $[X]$ generated by the n variables $X := \{x_1, \ldots, x_n\}$. We have the following map from X to $\mathbb{F}_2^n$: $\psi : X \to \mathbb{F}_2^n$, where $x_i \mapsto e_i$ (the i-th coordinate vector). The map ψ can be extended in a natural way to a morphism from $[X]$ onto $\mathbb{F}_2^n$, where $\psi(\prod_{i=1}^{n} x_i^{\beta}) = (\beta_1 \bmod 2, \ldots, \beta_n \bmod 2)$.

A binary code $\mathcal{C}$ defines an equivalence relation $R_{\mathcal{C}}$ in $\mathbb{F}_2^n$ given by $(x, y) \in R_{\mathcal{C}}$ if and only if $x - y \in \mathcal{C}$. If we define $\xi(u) := \psi(u)H$, where $u \in [X]$, the above congruence can be translated to $[X]$ by the morphism ψ as $u \equiv_{\mathcal{C}} w$ if and only if $(\psi(u), \psi(w)) \in R_{\mathcal{C}}$, that is, if $\xi(u) = \xi(w)$. The morphism ξ represents the transition of the syndromes from $\mathbb{F}_2^n$ to $[X]$; therefore, $\xi(w)$ is the "syndrome" of w, which is equal to the syndrome of $\psi(w)$.

Definition 3 (The ideal associated with a binary code). *Let $\mathcal{C}$ be a binary code. The ideal $I(\mathcal{C})$ associated with $\mathcal{C}$ is $I(\mathcal{C}) := \langle \{w - u \mid \xi(w) = \xi(u)\} \rangle \subset K[X]$.*

The Algorithm for Binary Codes. The monoid M is set to be $\mathbb{F}_2^{n-k}$ (where the syndromes belong to). Doing $g_i := \xi(x_i)$, note that $M = \mathbb{F}_2^{n-k} = \langle g_1, \ldots, g_n \rangle$. Moreover, $\sigma := R_{\mathcal{C}}$, hence $I(\sigma) = I(\mathcal{C})$. Let $\prec$ be an admissible ordering. Then the FGLM algorithm for linear codes can be used to compute the reduced Gröbner basis, the border basis, or the Gröbner representation for $\prec$.

Algorithm 2 (FGLM for binary codes)
Input: n, H *the parameters for a given binary code,* $\prec$ *an admissible ordering.*
Output: $rGb(I(\mathcal{C}), \prec)$.

```
$List := \{1\}, N := \emptyset, r := 0, G = \{\}$;
While $List \neq \emptyset$ do
    $w :=$ NextTerm$[List]$;
    If $w \notin T(G)$;
    $v := \xi(w)$;
    $\Lambda :=$ Member$[v, \{v_1, \ldots, v_r\}]$;
    If $\Lambda \neq False$ then $G := G \cup \{w - w_j\}$;
     else $r := r + 1$;
        $v_r := v'$;
        $w_r := w,\ N := N \cup \{w_r\}$;
        $List :=$ InsertNext$[w_r, List]$;
Return$[G]$.
```

In many cases of FGLM applications a good choice of the ordering $\prec$ is a crucial point in order to solve a particular problem. In the following theorem it is shown the importance of using a total degree compatible ordering (for example the Degree Reverse Lexicographic). Let us denote by $<_T$ a total degree compatible ordering.

Theorem 2 (Canonical forms of the vectors in $B(C, t)$). *Let $\mathcal{C}$ be a code and let G_T be the reduced Gröber basis with respect to $<_T$. If $w \in [X]$ satisfies* $\text{weight}(\psi(Can(w, G_T))) \leq t$ *then $\psi(Can(w, G_T))$ is the error vector corresponding to $\psi(w)$. On the other hand, if* $\text{weight}(\psi(Can(w, G_T))) > t$ *then $\psi(w)$ contains more than t errors.*

Proof. If we assume that $\text{weight}(\psi(Can(w, G_T))) \leq t$ then, we can infer at once that $\psi(w) \in B(\mathcal{C}, t)$ and $\psi(Can(w, G_T))$ is its error vector (notice that $\xi(w) = \xi(Can(w, G_T))$ and the unicity of the error vector).

Now, if $\text{weight}(\psi(Can(w, G_T))) > t$, we have to prove that $\psi(w) \notin B(\mathcal{C}, t)$. It is equivalent to show that $\text{weight}(\psi(Can(w, G_T))) \leq t$ if $\psi(w) \in B(\mathcal{C}, t)$. Let $\psi(w)$ be an element of $B(\mathcal{C}, t)$ and let e be its error vector then, $\text{weight}(e) \leq t$. Let w_e be the squarefree representation of e. Note that $\text{weight}(e)$ coincides with the total degree of w_e; accordingly, $L(\text{w}_e) \leq t$. On the other hand, $Can(w, G_T) <_T \text{w}_e$, which implies that $L(Can(w, G_T)) \leq L(\text{w}_e)$ (because $<_T$ is degree compatible). Hence, $\text{weight}(\psi(Can(w, G_T))) \leq L(Can(w, G_T)) \leq t$. □

The computation of the error-correcting cability of the code t can be done in the computing process of Algorithm 2 (see Example 1 and [5]). The previous theorem allows us to use the computed reduced Gröbner basis for solving the decoding problem in general binary codes, but also with such a powerful tool available, it is expected to be able to study the structure of the codes, like some combinatorics properties. Some possible examples are the permutation-equivalence of codes (see [5]), and some problems related with binary codes associated with the set of cycles in a graph (finding the set of minimal cycles and a minimal cycle basis of the cycles of a graph), see [6].

To generalize Theorem 2 for non binary linear codes have some conflicts with the needed ordering; however, the FGLM algorithm can be still used to compute the border basis or a Gröbner representation for the ideal $I(\mathcal{C})$ and it will be possible to solve the problems that one can solve with the reduced Gröbner basis in the case of binary codes. Those problems are explained in [5]. In addition, [5] contains some results and examples about the application of this setting to general linear codes and, in binary codes, for studying the problems of decoding and the permutation-equivalence.

Remark 3 (Complexity considerations). In contrast with a brute-force syndrome decoding method, which enumerates the 2^n elements in $\mathbb{F}_2^n$, computes their images by the parity check matrix and stores the word of smallest weight for each image, Algorithm 2 computes a reduced Gröbner basis associated with a given binary code (also a Gröbner representation or the border basis) by analyzing $2^{n-k}n$ candidates (the numbers of elements that belong to *List*) in $\mathbb{F}_2^n$ for the 2^{n-k} representative elements of the quotient $K[X]/I(\mathcal{C})$. Each iteration of the Algorithm can be arranged to be linear in the number of variables; thus, *Algorithm 2 performs $\mathcal{O}(2^{n-k}n^2)$ operations.*

Regarding the computation of the canonical form, it is known that the border basis or matphi gives a very efficient reduction algorithm, in the case of codes, the canonical form can be obtained at most after n reductions. However, the border basis or matphi needs a memory space proportional to $\mathcal{O}(2^{n-k}n)$. A reduced Gröbner basis can be substantially smaller than the border basis although in general no better bound can be given. It is kown to be non efficient the computation of the canonical form using the reduced Gröbner basis, although in the case associeted with binary codes, a better result could be achieved. For a detailed complexity analysis see [5, 7].

Example 1. Let $\mathcal{C}$ be the linear code over $\mathbb{F}_2^6$ determined by the parity check matrix $H^T = \begin{vmatrix} 1, & 1, & 0, & 1, & 0, & 0 \\ 1, & 0, & 1, & 0, & 1, & 0 \\ 1, & 1, & 1, & 0, & 0, & 1 \end{vmatrix}$ The minimum distance is $d = 3$, so, $t = 1$, the numbers of variables is 6, $<_T$ is set to be the Degree Reverse Lexicographic ordering with $x_{i+1} >_T x_i$.

Application of Algorithm 2. (Only essential parts of the computation will be described.) $List := \{1\}$; $N := \{\}$; $r := 0$; $w := 1$; $\xi(1) = (0,0,0)$; $N := N \cup \{1\} = \{1\}$; $\xi(N) := \{(0,0,0)\}$; $List := \{x_1, x_2, x_3, x_4, x_5, x_6\}$; $w := x_1$; $\xi(x_1) = (1,1,1)$; $N := \{1, x_1\}$; $\xi(N) := \{(0,0,0), (1,1,1)\}$;
After analyzing x_6 we are at the following stage:

$N := \{1, x_1, x_2, x_3, x_4, x_5, x_6\}$, and $List = \{x_1^2, x_1x_2, x_1x_3, x_1x_4, x_1x_5, x_1x_6, x_2^2, x_2x_3, x_2x_4, x_2x_5, x_2x_6, x_3^2, x_3x_4, x_3x_5, x_3x_6, x_4^2, x_4x_5, x_4x_6, x_5^2, x_5x_6, x_6^2\}$.

There is still one element left in N because there are 7 elements in N of a total of 8 (2^{6-3}). Taking the elements of *List* from x_1^2 to x_1x_5 they are a linear combination of elements already in N (their syndromes are in the list of syndromes computed $\xi(N)$). Therefore, $G := \{x_1^2 - 1, x_1x_2 - x_5, x_1x_3 - x_4, x_1x_4 - x_3, x_1x_5 - x_2\}$, for example $x_1x_2 - x_5$ is obtained, bacause when $w = x_1x_2$, first note that $Pred(w) \subset N$, which means that it is either a new irreducible element or a head of a binomial of the reduced basis. Then $\xi(x_1x_2)$ is computed and we got that $\xi(x_1x_2) = \xi(x_5)$. This means that $x_1x_2 - x_5$ belongs to G. Also x_1x_2 is the first minimal representation which is not in N, this implies that $t = \text{weight}(\psi(x_1x_2)) - 1$ (see [5]). The next element in *List*, $w = x_1x_6$, is the last element that will be included in N and the corresponding multiples will be included in *List*. From this point, the algorithm will just take elements from *List* and it analyzes in each case whether it is in $T\{rGb(I(\mathcal{C}), <_T)\}$ (like x_2x_3) or in $T(rGb(I(\mathcal{C}), <_T) \setminus T\{rGb(I(\mathcal{C}), <_T)\}$ (like $x_1x_2x_6$), this process is executed until the *List* is empty when the last element $x_1x_6^2$ of the list is analyzed. Finally, the reduced Gröbner basis for $<_T$ is

$$\begin{aligned} G := \{&x_1^2 - 1, x_1x_2 - x_5, x_1x_3 - x_4, x_1x_4 - x_3, x_1x_5 - x_2, x_2^2 - 1, x_2x_3 - x_1x_6,\\ &x_2x_4 - x_6, x_2x_5 - x_1, x_2x_6 - x_4, x_3^2 - 1, x_3x_4 - x_1, x_3x_5 - x_6, x_3x_6 - x_5,\\ &x_4^2 - 1, x_4x_5 - x_1x_6, x_6x_4 - x_2, x_5^2 - 1, x_5x_6 - x_3, x_6^2 - 1\}. \end{aligned}$$

Assume the vector $y = (1, 1, 1, 0, 1, 0)$ is received, the corresponding word is $w = x_1x_2x_3x_5$. Compute $w_e = Can(w, G) = x_3$. As $\text{weight}(\psi(w_e)) = 1$ the error vector is $e = (0, 0, 1, 0, 0, 0)$, and the codeword is $c = (1, 1, 0, 0, 1, 0)$.

References

1. J. Abbott, M. Kreuzer, L. Robiano. Computing Zero-Dimensional Schemes. *J. Symb. Comp.* 39(1), p. 31-49, 2005.
2. W.W. Adams, Ph. Loustaunau. *An introduction to Gröbner bases.* Graduate Studies in Mathematics, 3. American Mathematical Society, Providence, RI, 1994.
3. M.E. Alonso, M.G. Marinari, T. Mora. The Big Mother of all Dualities: Möller Algorithm. *Comm. Algebra* 31(2), 783-818, 2003.
4. M. A. Borges-Trenard, M. Borges-Quintana, and T. Mora. Computing Gröbner bases by FGLM techniques in a non-commutative setting. *J. Symb. Comp.* 30(4), p. 429–449, 2000.
5. M. Borges-Quintana, M. Borges-Trenard and E. Martínez-Moro. On a Gröbner bases structure associated to linear codes. Submitted to J. Discrete Math. Sci. Cryptogr. *Arxiv Preprint*, `http://arxiv.org/abs/math.AC/0506045`.
6. M. Borges-Quintana, M. Borges-Trenard, P. Fitzpatrick and E. Martínez-Moro. Gröbner bases and combinatorics for binary codes. Submitted to Appl. Algebra Engrg. Comm. Comput., 2005.
7. M. Borges-Quintana, F. Winkler, and M. Borges-Trenard. FGLM Techniques Applied to Linear Codes – An Algorithm for Decoding Linear Codes. *Techn. Rep.*, RISC-Linz, RISC - 00-14, J. Kepler Univ., Linz, Austria, 2000.
8. B. Buchberger. An Algorithmic Criterion for the Solvability of a System of Algebraic Equations (German). *Aequationes Mathematicae* 4, p. 374-383, 1970. (English translation in [10]).
9. B. Buchberger, H.M. Möller. The construction of Multivariate Polynomials with Preassigned Zeros". In: EUROCAM'82, Marseille. *LNCS.* 144, p. 24-31, 1982.

10. B. Buchberger, F. Winkler (eds). *Gröbner Bases and Applications* (Proc. of the Conference 33 Years of Gröbner Bases). London Mathematical Society Lecture Notes Series 251, C.U.P., 1998.
11. J. Faugère, P. Gianni, D. Lazard, T. Mora. Efficient Computation of Zero-Dimensional Gröbner Bases by Change of Ordering. *J. Symb. Comp.* 16(4), p. 329-344, 1993.
12. M.G. Marinari, H.M. Möller, T. Mora. *Gröbner Bases of Ideals Defined by Functionals with an Application to Ideals of Projective Points.* Appl. Algebra Engrg. Comm. Comput. 4(2), p. 103-145, 1993.
13. T. Mora. *Solving Polynomial Equation Systems II: Macaulay's Paradigm and Gröbner Technology.* Encyclopedia of Maths. and its Applications 99. C.U.P., 2005.

A Theory of Highly Nonlinear Functions

K.J. Horadam

Royal Melbourne Institute of Technology, Melbourne, VIC 3001, Australia
Kathy.Horadam@ems.rmit.edu.au

Abstract. Highly nonlinear functions are important as sources of low-correlation sequences, high-distance codes and cryptographic primitives, as well as for applications in combinatorics and finite geometry.

We argue that the theory of such functions is best seen in terms of splitting factor pairs. This introduces an extra degree of freedom, through the pairing of a normalised function $\phi : G \to N$ between groups with a homomorphism $\varrho : G \to \mathrm{Aut}(N)$.

From this perspective we introduce a new definition of equivalence for functions, relative to ϱ, and show it preserves their difference distributions. When $\varrho \equiv 1$ it includes CCZ and generalised linear equivalence, as well as planar and linear equivalence.

More generally, we use splitting factor pairs to relate several important measures of nonlinearity. We propose approaches to both linear approximation theory and bent functions, and to difference distribution theory and perfect nonlinear functions, which encompass the current approaches.

The purpose of this paper is to argue that the proper context in which to set the general theory of nonlinearity for functions of significance in combinatorics, finite geometry, coding, sequences or cryptography is the *splitting factor pairs.*

Factor pairs generalise the family of functions known as cocycles. The simplest class of factor pairs — the *splitting* factor pairs — generalise the simplest class of cocycles — the coboundaries — in a way which allows us to study all normalised functions between arbitrary (not necessarily abelian) groups. In Section 1 this fundamental relationship between functions defined on groups and splitting factor pairs is demonstrated. Each pair consisting of a normalised function $\phi : G \to N$ between groups and a homomorphism ϱ from G to $\mathrm{Aut}(N)$, the group of automorphisms of N, defines a splitting factor pair, and vice versa. The ϱ-twisted homomorphisms between groups are introduced and orthogonality for a factor pair is defined.

In Section 2, we use equivalence of transversals to to determine equivalence classes (bundles) of splitting factor pairs, generalising Theorem 3.2 of [8]. These in turn define equivalence classes of normalised functions, relative to ϱ, which we also call *bundles* (Definition 7).

We argue that these bundles are the natural equivalence classes for functions, for several reasons. For instance, the difference distribution of a function, from

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 87–100, 2006.

which important measures of its nonlinearity are derived, is an invariant of its bundle (Theorem 4).

Only the bundles for the case $\varrho \equiv 1$ are likely to be of practical interest for some time. They are specified in Theorem 3. When $\varrho \equiv 1$ the underlying equivalence relation includes the expected cases of equivalence of planar functions [5] and linear equivalence of cryptographic functions, as well as CCZ equivalence [3] and generalised linear equivalence [1] of functions. In particular, a permutation and its inverse are in the same bundle (Corollary 3).

However, by considering non-trivial homomorphisms ϱ, a framework for understanding the nature and interrelationships of many different measures of nonlinearity may be constructed.

Perera and Horadam [15] proved that orthogonal cocycles, central semiregular relative difference sets, cocyclic generalised Hadamard matrices and divisible designs with a regular group of automorphisms in which a central subgroup acts class regularly, are all equivalent. These four structures are shown by Hughes [10] also to be equivalent to a type of low correlation function he calls a base sequence. Under these equivalences, an orthogonal coboundary corresponds to a perfect nonlinear, or planar, or bent function, depending on the particular groups employed.

Galati [7] has extended the work of Perera and Horadam maximally, proving that orthogonal factor pairs, semiregular relative difference sets, coupled cocyclic generalised Hadamard matrices and class regular divisible designs with a regular group action, are all equivalent. Similarly, the author [9–Chapter 7] has extended Hughes' base sequences to this most general case. These five mutual equivalences are called the *Five-fold Constellation*.

In Section 3, we describe this Five-fold Constellation in the simplest (splitting) case. The object corresponding to a base sequence is a *perfect nonlinear function* $\phi : G \to N$, *relative to a homomorphism* $\varrho : G \to \mathrm{Aut}(N)$ (Theorem 5). A consequence for combinatorial applications is a better understanding of what 'splitting' really means for relative difference sets (Definition 10) than that which follows from the traditional definition.

With this perspective, in Section 4 we develop a framework for a general theory of nonlinearity for normalised functions between groups. This theory is based on observation in numerous contexts of the relationship between perfect nonlinear, planar and bent functions, and the rows of a group-developed generalised Hadamard matrix. In particular, new definitions are proposed whereby a normalised function $\phi : G \to N$ between groups is bent, or maximally nonlinear, or a flat perfect array, relative to ϱ.

The following conventions are observed. Groups G of order v and N of order w are written multiplicatively unless otherwise specified. For $a \in N$, $\overline{a}$ denotes the inner automorphism $\overline{a}(n) = ana^{-1}$ for all $n \in N$. For $\sigma_1, \sigma_2 \in \mathrm{Aut}(N)$, multiplication $\sigma_1\sigma_2$ is given by the "opposite" action $n^{\sigma_1\sigma_2} = \sigma_1(\sigma_2(n)),\ n \in N$.

We will restrict our study of nonlinearity to functions $\phi : G \to N$ which are *normalised*; that is, $\phi(1) = 1$. The group of normalised functions $\{\phi : G \to N,\ \phi(1) = 1\}$ under pointwise multiplication is denoted $C^1(G, N)$.

We begin with a brief summary of the cohomological background, necessary to describe splitting factor pairs and their properties. For more details, see [7] or [9].

Definition 1. *A (normalised)* factor pair *of N by G is a pair (ψ, ε) of functions $\psi : G \times G \to N$ (the* factor set*) and $\varepsilon : G \to \mathrm{Aut}(N)$ (the* coupling*) satisfying, for all $x, y, z \in G$,*

$$\varepsilon(x)\varepsilon(y) = \overline{\psi(x,y)}\varepsilon(xy), \tag{1}$$

$$\psi(x,y)\psi(xy,z) = \psi(y,z)^{\varepsilon(x)}\psi(x,yz), \tag{2}$$

$$\psi(x,1) = 1 = \psi(1,x). \tag{3}$$

The set of all factor pairs of N by G is denoted $F^2(G,N)$.

For each $\phi \in C^1(G,N)$ and $(\psi,\varepsilon) \in F^2(G,N)$ there is an equivalent factor pair $(\psi', \varepsilon') \sim_\phi (\psi, \varepsilon) \in F^2(G,N)$.

Definition 2. *A factor pair (ψ', ε') of N by G is* equivalent *to (ψ,ε) via ϕ, written $(\psi',\varepsilon') \sim_\phi (\psi,\varepsilon)$, if there exists a function $\phi \in C^1(G,N)$ such that, for all $x, y \in G$,*

$$\varepsilon'(x) = \overline{\phi(x)}\varepsilon(x), \text{ and} \tag{4}$$

$$\psi'(x,y) = \phi(x)\phi(y)^{\varepsilon(x)}\psi(x,y)\phi(xy)^{-1}. \tag{5}$$

The equivalence class containing (ψ,ε) is denoted $[\psi,\varepsilon]$.

Each factor pair (ψ,ε) in $F^2(G,N)$ determines a canonical extension $N \overset{\iota}{\rightarrowtail} E_{(\psi,\varepsilon)} \overset{\kappa}{\twoheadrightarrow} G$ of N by G, where the underlying set of the *extension group* $E_{(\psi,\varepsilon)}$ is $N \times G$, where $\iota(a) = (a,1)$ and $\kappa(a,x) = x$, and where the multiplication in $E_{(\psi,\varepsilon)}$ is

$$(a,x)(b,y) = (ab^{\varepsilon(x)}\psi(x,y), xy), \; a, b \in N, \; x, y \in G. \tag{6}$$

Conversely, if $N \overset{\imath}{\rightarrowtail} E \overset{\pi}{\twoheadrightarrow} G$ is an extension of N by G, every transversal $T = \{t_x : x \in G\}$ of $\imath(N)$ in E determines a factor pair $(\psi\ \ , \varepsilon\ \ )$ in $F^2(G,N)$

$$\varepsilon_T(x) = \imath^{-1} \circ \overline{t_x} \circ \imath \tag{7}$$

$$\psi_T(x,y) = \imath^{-1}(t_x t_y t_{xy}^{-1}). \tag{8}$$

Every factor pair in $[\psi\ \ , \varepsilon\ \ ]$ derives from such a transversal.

In particular, $T = \{t_x = (1,x) : x \in G\} \subseteq E_{(\psi,\varepsilon)}$ is a transversal of $\iota(N) = N \times \{1\}$ in $E_{(\psi,\varepsilon)}$ with $(\psi\ \ , \varepsilon\ \ ) = (\psi, \varepsilon)$, and if $(\psi', \varepsilon') \sim_\phi (\psi,\varepsilon)$, there is an isomorphism $\beta : E_{(\psi',\varepsilon')} \to E_{(\psi,\varepsilon)}$ defined by

$$\beta((a,x)) = (a\phi(x), x). \tag{9}$$

1 Splitting Factor Pairs

If (ψ, ε) is a factor pair and $\psi \equiv 1$ then by (1) the coupling ε is a group homomorphism ϱ, and any factor pair equivalent to $(1, \varrho)$ is called *splitting.*

The splitting factor pairs are the focus of this paper. The coupling ϱ brings an extra degree of freedom to the study of normalised functions $\phi : G \to N$.

Definition 3. *$(\psi, \varepsilon) \in F^2(G, N)$ is a* splitting *factor pair if there exist $\phi \in C^1(G, N)$ and a homomorphism $\varrho : G \to \mathrm{Aut}(N)$ such that $(\psi, \varepsilon) \sim_\phi (1, \varrho)$. It has the form $(\psi, \varepsilon) = (\partial\phi^{-1}, \overline{\phi}\varrho)$, where*

$$(\overline{\phi}\varrho)(x) = \overline{\phi(x)}\varrho(x) \tag{10}$$

$$\partial\phi^{-1}(x, y) = \phi(x)\,\phi(y)^{\varrho(x)}\,\phi(xy)^{-1},\ \ x, y \in G. \tag{11}$$

In particular, splitting factor pairs determine split extensions of N by G and split transversals. The corresponding extension groups $E_{(\psi,\varepsilon)}$ are semidirect products.

Lemma 1. *Let $\phi \in C^1(G, N)$, let $\varrho : G \to \mathrm{Aut}(N)$ be a homomorphism and let $(\psi, \varepsilon) \in F^2(G, N)$.*

Then $(\psi, \varepsilon) \sim_\phi (1, \varrho) \Leftrightarrow E_{(\partial\phi^{-1},\overline{\phi}\varrho)} \cong E_{(1,\varrho)} = N \rtimes_\varrho G$. In particular, $(\psi, \varepsilon) \sim_\phi (1, 1) \Leftrightarrow E_{(\psi,\varepsilon)} \cong N \times G$.

In this case if $\beta : E_{(\partial\phi^{-1},\overline{\phi}\varrho)} \cong E_{(1,\varrho)}$ is the isomorphism of (9) and $T = \{(1, x) : x \in G\} \subseteq E_{(\partial\phi^{-1},\overline{\phi}\varrho)}$ then $\beta(T) = \{(\phi(x), x) : x \in G\} \subseteq N \rtimes_\varrho G$.

For each homomorphism $\varrho : G \to \mathrm{Aut}(N)$ there is a surjection $\partial_\varrho : C^1(G, N) \twoheadrightarrow [1, \varrho] \subset F^2(G, N)$ defined by

$$\partial_\varrho(\phi) = (\partial\phi^{-1}, \overline{\phi}\varrho). \tag{12}$$

The preimage of $(1, \varrho)$ under ∂_ϱ is a group which is important for our analysis. It is a generalisation of the group of homomorphisms $\mathrm{Hom}(G, N)$, which is the preimage of $(1, 1)$; that is, the case $\varrho \equiv 1$. (In the special case that G is abelian with exponent m and N is the cyclic group of complex m^{th} roots of unity, $\mathrm{Hom}(G, N)$ is the character group of G.)

Definition 4. *Let $\varrho : G \to \mathrm{Aut}(N)$ be a homomorphism. Then $\chi \in C^1(G, N)$ is a ϱ-*twisted homomorphism *if $\overline{\chi}\varrho = \varrho$ and $\chi(xy) = \chi(x)\,\chi(y)^{\varrho(x)},\ x, y \in G$. Denote the subgroup of ϱ-twisted homomorphisms in $C^1(G, N)$ by $\mathrm{Hom}_\varrho(G, N)$.*

Though $\mathrm{Hom}_\varrho(G, N)$ may not be a normal subgroup of $C^1(G, N)$, its (left) cosets $\phi\,\mathrm{Hom}_\varrho(G, N)$ are the preimages of the distinct elements in $[1, \varrho]$. The coset mapping $\widehat{\partial}_\varrho(\phi\ \mathrm{Hom}_\varrho(G, N)) = \partial_\varrho(\phi)$ induced by (12) is a set isomorphism,

$$\widehat{\partial}_\varrho : \{\phi\,\mathrm{Hom}_\varrho(G, N) : \phi \in C^1(G, N)\} \overset{\cong}{\rightarrowtail} [1, \varrho]. \tag{13}$$

However, if N is abelian, factor pairs are the rather more familiar functions known as *cocycles*, and $\mathrm{Hom}_\varrho(G, N)$ is a normal subgroup of the abelian group $C^1(G, N)$.

Lemma 2. *Let N be abelian. Then a factor pair (ψ,ε) is a cocycle $\psi \in Z^2_\varepsilon(G,N)$ with coefficients in the G-module (N,ε). If $(\psi,\varepsilon) \sim_\phi (1,\varrho)$, then $\varepsilon = \overline{\phi}\varrho = \varrho$, $\psi = \partial\phi^{-1} = (\partial\phi)^{-1} \in B^2_\varrho(G,N)$ is a coboundary, and $\widehat{\partial}_\varrho : C^1(G,N)/\mathrm{Hom}_\varrho(G,N) \rightarrowtail [1,\varrho] = B^2_\varrho(G,N)$ is a group isomorphism.*

The final notion, orthogonality, introduced in this section will subsequently be shown (Theorem 5) to correspond to perfect nonlinearity for normalised functions. Galati [7–Definition 4.1] defines $(\psi,\varepsilon) \in F^2(G,N)$ to be (v,w,k,λ)-*orthogonal with respect to* a k-subset D of G if, for each $x \in G\backslash\{1\}$, in the group ring $\mathbb{Z}N$

$$\sum_{y\in D\cap x^{-1}D} \psi(x,y) \ = \ \lambda \sum_{a\in N} a. \tag{14}$$

In this case, D is an ordinary $(v,k,w\lambda)$-difference set in G. When $k = v$, so $D = G$ and $\lambda = v/w$, (ψ,ε) is termed *orthogonal.* When N is abelian, $\varepsilon \equiv 1$ and $k = v$, this specialises to the original definition of an orthogonal cocycle due to the author and Perera [15]. Orthogonality is the property of factor pairs which characterises the corresponding transversals as semiregular relative difference sets (RDSs).

2 Bundles of Factor Pairs and Equivalence of Functions

Equivalence between transversals is defined from that for RDSs. Here we use it to derive equivalence classes in $C^1(G,N)$ which we claim are the natural classes for studying functions of interest as S-box functions, low-correlation sequences, planar functions or high-distance codes.

Definition 5. [8–Definition 2.1] *Let T, T' be transversals of the isomorphic normal subgroups K, K', respectively, in E. Then T and T' are* equivalent *if there exist $\alpha \in \mathrm{Aut}(E)$ and $e \in E$ such that $\alpha(K) = K'$ and $\alpha(T) = e\,T'$.*

Every transversal is equivalent to a normalised one (i.e. containing 1).

In [8] the author maps equivalent normalised transversals T, T' of isomorphic central subgroups C, C' in E, to a corresponding pair of cocycles, thereby defining an equivalence relation between cocycles. The resulting equivalence classes of cocycles are termed *bundles.* For equivalent transversals of normal subgroups we have to do the same for the corresponding factor pairs.

Theorem 1. *Let E be an extension group of N by G and let T and T' be normalised transversals in E of the normal subgroups $K \cong K' (\cong N)$, respectively, for which $E/K \cong E/K' \cong G$. Let $(\psi,\varepsilon), (\psi',\varepsilon') \in F^2(G,N)$ be the factor pairs corresponding to T, T', respectively.*

There exist $\alpha \in \mathrm{Aut}(E)$ and $e \in E$ such that $\alpha(K') = K$ and $\alpha(T') = e\,T$ if and only if there exist $\gamma \in \mathrm{Aut}(N)$, $\theta \in \mathrm{Aut}(G)$ and $s \in G$ such that

1. $(\psi',\varepsilon') = \big(\gamma \circ (\psi \cdot s) \circ (\theta \times \theta),\ \gamma \circ ((\varepsilon \cdot s) \circ \theta) \circ \gamma^{-1}\big)$, *where*
2. $\big(\psi \cdot s,\ \varepsilon \cdot s\big) \sim_{\psi|}\ (\psi,\varepsilon)$ *and where*

3. $\psi|_s \in C^1(G,N)$ *is defined by*

$$\psi|_s(x) = \psi(s, s^{-1})^{-1}\psi(s, s^{-1}x), \quad x \in G. \tag{15}$$

Proof. Straightforward adaptation of [8–Theorem 3.2] to the normal subgroup case applies, on writing $e = k\,t_s$ uniquely for $k \in K$, $t_s \in T$ and $s \in G$, using (6) and showing by normalisation that $k = \iota(\psi(s,s^{-1})^{-1})$.

Definition 6. *Let* $(\psi, \varepsilon) \in F^2(G,N)$*. The* bundle $\mathcal{B}((\psi,\varepsilon))$ of (ψ,ε) *is the set*

$$\mathcal{B}((\psi,\varepsilon)) = \big\{\big(\gamma\circ(\psi\cdot s)\circ(\theta\times\theta),\ \gamma\circ((\varepsilon\cdot s)\circ\theta)\circ\gamma^{-1}\big) : \gamma\in\mathrm{Aut}(N), \theta\in\mathrm{Aut}(G), s\in G\big\}.$$

The splitting case of Theorem 1 may now be extracted without much difficulty.

Theorem 2. *Let* $\phi \in C^1(G,N)$ *and let* $\varrho : G \to \mathrm{Aut}(N)$ *be a homomorphism. Suppose* $(\psi,\varepsilon) = (\partial\phi^{-1}, \overline{\phi}\varrho) \sim_\phi (1,\varrho)$*. Let* s*,* θ *and* γ *be as in Theorem 1, and let* $(\psi',\varepsilon') = \big(\gamma\circ(\psi\cdot s)\circ(\theta\times\theta),\ \gamma\circ((\varepsilon\cdot s)\circ\theta)\circ\gamma^{-1}\big)$*. Define*

$$(\phi\cdot s)(x) = \big(\phi(s^{-1})^{-1}\phi(s^{-1}x)\big)^{\varrho(s)},\ x\in G \tag{16}$$

$$\phi'(x) = (\gamma\circ(\phi\cdot s)\circ\theta)(x),\ x\in G. \tag{17}$$

Then $\phi\cdot s,\ \phi' \in C^1(G,N)$ *and*

1. $(\psi\cdot s,\ \varepsilon\cdot s) = (\partial(\phi\cdot s)^{-1},\ \overline{(\phi\cdot s)}\,\varrho) \sim_{\phi\cdot s} (1,\ \varrho)$;
2. $(\psi',\varepsilon') = (\partial(\phi')^{-1}, \overline{(\phi')}\,\varrho') \sim_{\phi'} (1,\ \varrho')$*, where* $\varrho' : G \to \mathrm{Aut}(N)$ *is the homomorphism defined by* $\varrho'(x) = \gamma\circ\varrho(\theta(x))\circ\gamma^{-1},\ x\in G$*.*

Proof. Since $(\psi,\varepsilon)\sim_\phi(1,\varrho)$ and $(\psi\cdot s,\ \varepsilon\cdot s)\sim_{\psi|}\ (\psi,\varepsilon)$ by Theorem 1, it follows that $(\psi\cdot s,\ \varepsilon\cdot s)\sim_{\psi|\ \phi}(1,\varrho)$ and by (11), $(\psi|_s\phi)(x) = (\phi(s^{-1})^{-1}\phi(s^{-1}x))^{\varrho(s)} = (\phi\cdot s)(x)$, giving part 1. Then part 2 follows because $\overline{\phi'(x)} = \overline{\gamma\circ(\phi\cdot s)(\theta(x))}$ $= \gamma\circ\overline{\phi}(\theta(x))\circ\gamma^{-1}$.

It follows from Definition 6 that the bundle of a splitting factor pair consists entirely of splitting factor pairs: $\mathcal{B}((\partial\phi^{-1},\overline{\phi}\varrho)) =$

$$\big\{\big(\partial(\phi')^{-1},\ \overline{\phi'}\,(\gamma\circ\varrho(\theta)\circ\gamma^{-1})\big) : \phi' = \gamma\circ(\phi\cdot s)\circ\theta, \gamma\in\mathrm{Aut}(N), \theta\in\mathrm{Aut}(G), s\in G\big\} \tag{18}$$

Thus the set of splitting factor pairs partitions into disjoint bundles. However, it is important to recognise that a bundle of splitting factor pairs cuts across equivalence classes of splitting factor pairs, and vice versa. In fact, for a particular homomorphism $\varrho : G\to\mathrm{Aut}(N)$ we have

$$\bigvee_{\theta,\gamma}[1,\gamma\circ\varrho(\theta)\circ\gamma^{-1}] = \bigvee_{\phi}\mathcal{B}(\partial_\varrho(\phi)).$$

These bundles are the heavy machinery we use, together with (13), to arrive at a natural definition of equivalence of functions $\phi\in C^1(G,N)$ relative to ϱ.

Two mappings $\varphi,\phi \in C^1(G,N)$ are *equivalent relative to* ϱ if there exist $\theta\in\mathrm{Aut}(G)$ and $\gamma\in\mathrm{Aut}(N)$ such that $\mathcal{B}(\partial_{\varrho'}(\varphi)) = \mathcal{B}(\partial_\varrho(\phi))$, where $\varrho' = \gamma\circ\varrho(\theta)\circ\gamma^{-1}$; that is, by (12), if and only if there exist $\theta\in\mathrm{Aut}(G)$ and $\gamma\in\mathrm{Aut}(N)$ such that $(\partial\varphi^{-1},\overline{\varphi}\varrho')\in\mathcal{B}((\partial\phi^{-1},\overline{\phi}\varrho))$, where $\varrho' = \gamma\circ\varrho(\theta)\circ\gamma^{-1}$. By (18) and (13), we obtain the following more convenient definition.

Definition 7. *Let $\varrho : G \to \mathrm{Aut}(N)$ be a homomorphism. Two mappings $\varphi, \phi \in C^1(G,N)$ are* equivalent relative to *ϱ if there exist $\theta \in \mathrm{Aut}(G)$, $\gamma \in \mathrm{Aut}(N)$, $f \in \mathrm{Hom}_{\gamma\circ\varrho(\theta)\circ\gamma^{-1}}(G,N)$ and $s \in G$ such that*

$$\varphi = (\gamma \circ (\phi \cdot s) \circ \theta)\ f. \tag{19}$$

The function $\phi \cdot s$ in (19) is termed the shift *of ϕ by s. The equivalence class of ϕ relative to ϱ, denoted $\mathbf{b}(\phi,\varrho)$, is called its* bundle relative to *ϱ. That is, $\mathbf{b}(\phi,\varrho) =$*

$$\big\{(\gamma \circ (\phi \cdot s) \circ \theta)\ f : f \in \mathrm{Hom}_{\gamma\circ\varrho(\theta)\circ\gamma^{-1}}(G,N),\ \theta \in \mathrm{Aut}(G),\ \gamma \in \mathrm{Aut}(N),\ s \in G\big\}. \tag{20}$$

In particular, if $\varrho \equiv 1$, so $\gamma \circ \varrho(\theta) \circ \gamma^{-1} \equiv 1$, the bundle $\mathbf{b}(\phi) = \mathbf{b}(\phi, 1)$ *of ϕ is*

$$\mathbf{b}(\phi) = \big\{(\gamma \circ (\phi \cdot s) \circ \theta)\ f : f \in \mathrm{Hom}(G,N),\ \theta \in \mathrm{Aut}(G),\ \gamma \in \mathrm{Aut}(N),\ s \in G\big\}. \tag{21}$$

Corollary 1. *Let $\varrho : G \to \mathrm{Aut}(N)$ be a homomorphism and let $\phi \in C^1(G,N)$. For every $s \in G$, $\theta \in \mathrm{Aut}(G)$ and $\gamma \in \mathrm{Aut}(N)$,*

$$\begin{aligned} \mathbf{b}(\phi,\ \varrho) &= \mathbf{b}(\phi \cdot s,\ \varrho), \\ \mathbf{b}(\phi,\ \varrho) &= \mathbf{b}(\gamma \circ \phi \circ \theta,\ \gamma \circ \varrho(\theta) \circ \gamma^{-1}). \end{aligned}$$

For each homomorphism $\varrho : G \to \mathrm{Aut}(N)$, the group of all normalised functions $C^1(G,N)$ therefore partitions into disjoint bundles relative to ϱ

$$C^1(G,N) = \bigvee_{\phi} \mathbf{b}(\phi, \varrho).$$

Corollary 2. *Let $\varrho : G \to \mathrm{Aut}(N)$ be a homomorphism. The mapping $\natural$ defined on bundles by $\natural\,(\mathbf{b}(\phi,\varrho)) = \mathcal{B}(\partial_\varrho(\phi))$ is a set isomorphism*

$$\natural : \{\mathbf{b}(\phi,\varrho) : \phi \in C^1(G,N)\} \xrightarrow{\cong} \{\,\natural\,(\mathbf{b}(\phi,\varrho)) = \mathcal{B}(\partial_\varrho(\phi)) : \phi \in C^1(G,N)\}.$$

Next, we give the case $\varrho \equiv 1$ of Theorem 2 and Corollary 2. These are the only bundles which are likely to be of practical interest for some time. In this case, by Lemma 1 we have an isomorphism $\beta : E_{(\partial\phi^{-1},\overline{\phi})} \to N \times G$, under which the transversal $\{(1,x) : x \in G\}$ of $N \times \{1\}$ in $E_{(\partial\phi^{-1},\overline{\phi})}$ maps to transversal $T = \{(\phi(x),\ x) : x \in G\}$ of $N \times \{1\}$ in $N \times G$. By (7) and (8), the splitting factor pair determined by T is $(\partial\phi^{-1}, \overline{\phi})$, and $(\partial\phi^{-1}, \overline{\phi}) \sim_\phi (1,1)$.

Theorem 3. *(The case $\varrho \equiv 1$.) The following statements are equivalent:*

1. *The functions $\phi,\ \varphi \in C^1(G,N)$ are equivalent;*
2. *$\mathbf{b}(\phi) = \mathbf{b}(\varphi)$;*
3. *there exist $s \in G$, $\theta \in \mathrm{Aut}(G)$, $\gamma \in \mathrm{Aut}(N)$, $f \in \mathrm{Hom}(G,N)$ such that*

$$\varphi = (\gamma \circ (\phi \cdot s) \circ \theta)\ f, \quad \textit{where}$$

$$(\phi \cdot s)(x) = \phi(s^{-1})^{-1}\phi(s^{-1}x),\ x \in G,$$

4. the transversals $T_\varphi = \{(\varphi(x),\, x) : x \in G\}$ *and* $T_\phi = \{(\phi(x),\, x) : x \in G\}$ *of* $N \times \{1\}$ *in* $N \times G$ *are equivalent; that is, there exist* $\alpha \in \mathrm{Aut}(N \times G)$ *and* $s \in G$ *such that*

$$\alpha(N \times \{1\}) = N \times \{1\} \ \text{ and}$$

$$\alpha(T_\varphi) = (\phi^{-1}(s), s)\, T_\phi.$$

We illustrate with an application in the case $\varrho \equiv 1$ and $G = N$.

Corollary 3. *Let* $\varrho \equiv 1$, $G = N$ *and suppose* $\phi \in C^1(G,G)$ *is a permutation with inverse* $inv(\phi)$. *Then* ϕ *and* $inv(\phi)$ *are equivalent; that is,* $\mathbf{b}(\phi) = \mathbf{b}(inv(\phi))$.

Proof. Clearly, $T_\phi = \{(\phi(x), x) : x \in G\}$ is a normalised transversal of $G \times \{1\}$ in $G \times G$ and $T' = \{(x, \phi(x)) : x \in G\}$ is a normalised transversal of $\{1\} \times G$ in $G \times G$. The splitting factor pair determined by T_ϕ is $(\partial\phi^{-1}, \overline{\phi})$. Let $\tau(x,y) = (y,x)$ for all $x, y \in G$. Then $\tau \in \mathrm{Aut}(G \times G)$, $\tau(\{1\} \times G) = G \times \{1\}$ and $\tau(T') = T_\phi$, so T' and T_ϕ are equivalent. By Theorem 1 and Definition 6, the corresponding splitting factor pairs lie in the same bundle $\mathcal{B}((\partial\phi^{-1}, \overline{\phi}))$. But as a transversal of $G \times \{1\}$ in $G \times G$, $T' = T_{inv\phi} = \{(inv(\phi)(x), x) : x \in G\}$, so it determines the splitting factor pair $(\partial(inv(\phi))^{-1},\, \overline{inv(\phi)})$. By Theorem 3, $\mathbf{b}(\phi) = \mathbf{b}(inv(\phi))$.

To support our contention that bundles are the natural equivalence classes for functions, we will present three arguments.

Firstly, consider the (difference) distribution of ϕ relative to ϱ, which is significant in several measures of nonlinearity. Of fundamental importance is its invariance within $\mathbf{b}(\phi, \varrho)$ and under ∂_ϱ.

The *distribution of* $\phi \in C^1(G,N)$ *relative to* ϱ is the multi-set of all frequencies $\mathcal{D}(\phi,\varrho) = \{n_{(\phi,\varrho)}(x,a) = |\{y \in G : \phi(y)^{\varrho(x)}\phi(xy)^{-1} = a\}| : x \in G,\, a \in N\}$. Similarly, the *distribution of* $(\psi,\varepsilon) \in F^2(G,N)$ is the multi-set of all frequencies $\mathcal{D}((\psi,\varepsilon)) = \{N_{(\psi,\varepsilon)}(x,a) = |\{y \in G : \psi(x,y) = a\}| : x \in G,\, a \in N\}$.

Theorem 4. *For each* $\varrho : G \to \mathrm{Aut}(N)$, *the distribution of a function is an invariant of its bundle relative to* ϱ. *If* $\phi \in C^1(G,N)$ *is a homomorphism and* $\mathbf{b}(\phi,\varrho) = \mathbf{b}(\varphi,\varrho')$, *then* $\mathcal{D}(\phi,\varrho) = \mathcal{D}(\varphi,\varrho')$. *Furthermore,* $\mathcal{D}(\phi,\varrho) = \mathcal{D}(\partial_\varrho(\phi))$.

Proof. If $\varphi = \phi f$, where $f \in \mathrm{Hom}_\varrho(G,N)$, then $n_{(\varphi,\varrho)}(x,a) = n_{(\phi,\varrho)}(x, f(x)\,a)$. If $\varphi = \phi \cdot s$ for $s \in G$, then

$$n_{(\varphi,\varrho)}(x,a) \;=\; n_{(\phi,\varrho)}(s^{-1}xs,\ \phi(s^{-1})^{\varrho(s^{-1}xs)}a^{\varrho(s^{-1})}\phi(s^{-1})^{-1}).$$

The other statements are straightforward to substantiate.

Secondly, we demonstrate that bundle equivalence is familiar when $\varrho \equiv 1$ and $G = N$. It includes equivalence of planar functions and linear equivalence of cryptographic functions.

We specialise to the case $\varrho \equiv 1$ and $G = N = (GF(p^n), +)$, written additively. Every $\phi \in C^1(G,G)$ is the evaluation map of some polynomial $\phi(x) \in GF(p^n)[x]$ of degree less than p^n, with $\phi(0) = 0$. The homomorphisms $\mathrm{Hom}(G,G)$ are the linearised polynomials. The elements of $C^1(G,G)/\text{Hom}(G,G)$ (see Lemma 2) are represented by those $\phi(x) \in GF(p^n)[x]$ with no linearised summand.

Bundle equivalence is known implicitly to finite geometers, because planar functions equivalent by Definition 7 will determine isomorphic planes [5]. Planarity of $\phi(x)$ is preserved by the operations of linear transformation, addition of a linearised polynomial of G or pre- or post-composition with a linearised permutation polynomial, all of which are bundle equivalences. (In particular, if $s \in G$, then the linear transformation $\phi(x+s) - \phi(s)$ is the shift $(\phi \cdot s)(x)$.)

Bundle equivalence includes the linear equivalence used in cryptography and probably in other contexts as well. Two functions $\phi, \varphi \in C^1(G,G)$ are *linearly equivalent* [1–p. 80] if there exist invertible linear transformations β, γ of G and $f \in \text{Hom}(G,G)$ such that

$$\varphi(x) = (\gamma \circ \phi \circ \beta)(x) + f(x).$$

The nature of equivalence for cryptographic functions has attracted considerable attention recently, and competing definitions have been proposed [3, 1, 2]. These have been prompted by the observation that if ϕ is invertible, then $inv(\phi)$ has the same cryptographic robustness as ϕ, so the inverse of a function is also quoted as being equivalent to it.

Both [3–Proposition 3] for $p = 2$ (as cited in [2]) and [1] appear to have arrived independently at the same extension of linear equivalence which will include permutations and their inverses in the same equivalence class. Both extensions are the case $E = G \times G$ of Definition 5. In [2] the transversal $T = \{(\phi(x), x) : x \in G\}$ is called the *graph* of ϕ and translation is on the right. In [1] it is called the *implicit embedding* and no translation is included. By Corollary 3, bundle equivalence explains and unifies these ideas.

Thirdly, in the next section we argue that bundle equivalence is indeed the fundamental equivalence relation on functions, by virtue of the number of related equivalences.

3 The Splitting Five-Fold Constellation

The splitting factor pairs define the optimal generalisation of group developed (or, equally, group-invariant) matrices with entries in a group (c.f. [7–Theorem 10.1]).

Definition 8. *Let M be a $v \times v$ matrix with entries in N. Then M is a* coupled G-developed matrix over N *if there is an ordering $G = \{x_1, \ldots, x_v\}$, a mapping $\phi \in C^1(G,N)$ and a homomorphism $\varrho : G \to \text{Aut}(N)$ such that $M = M_{(\phi,\varrho)}$ where*

$$M_{(\phi,\varrho)} = [\,\phi(x_i x_j)^{\varrho(x^{-1})}\,]_{1 \le i,j \le v}. \tag{22}$$

If the coupling $\varrho \equiv 1$, we say M is G-developed. *The row of M indexed by 1 (without loss of generality, the top row) is always $(\phi(x_1), \phi(x_2), \ldots, \phi(x_v))$.*

Such top rows determine an important equivalence. The perfect binary arrays PBA and their quaternary counterparts PQA, defined for *abelian* groups G and for $N = \{\pm 1\}$ and $N = \{\pm 1, \pm i\}$, respectively, are used for signal array correlation. The existence of a PBA is equivalent to the existence of a Menon-Hadamard difference set in G and to the existence of a Menon Hadamard matrix with the PBA as top row. The existence of a PQA is equivalent to the existence of a G-developed complex Hadamard matrix with the PQA as top row. The top rows of G-developed generalised Hadamard matrices $GH(4, v/4)$ are the *flat* PQAs [11].

When G and N are abelian, the top rows of G-developed generalised Hadamard matrices over N are also familiar to cryptographers, where, following Nyberg, the defining functions $G \to N$ are called PN (perfect nonlinear). Her original definition [14–Def. 3.1] of PN functions has $G = \mathbb{Z}_r^n$ and $N = \mathbb{Z}_r^m$, $n \geq m$, and for $r = 2$ they are precisely the vectorial bent functions.

It is obvious that the function defining the top row of a coupled G-developed generalised Hadamard matrix $GH(w, v/w)$ is a most interesting object for study.

Definition 9. *Suppose $w|v$ and let ϕ, ϱ and $M_{(\phi,\varrho)}$ be as in Definition 8.*

The function ϕ is perfect nonlinear (PN) relative to ϱ *if $M_{(\phi,\varrho)}$ is a $GH(w, v/w)$ over N. If $\varrho \equiv 1$ we say ϕ is perfect nonlinear (PN).*

Let R be a ring with unity for which char R *does not divide v, $N \leq R^*$ and $\sum_{u \in N} u = 0$ in R. The sequence $(\phi(x),\ x \in G)$ is a* flat perfect array (FPA) over R relative to ϱ *if ϕ is PN relative to ϱ. If $\varrho \equiv 1$ we say it is a flat perfect array (FPA) over R.*

Next, we combine Definition 9 with the well-known equivalence between relative difference sets (RDS) and divisible designs. It has been traditional to call an RDS in E relative to N 'splitting' if $E \cong N \times G$, so that any splitting RDS with N abelian necessarily has N central in E. However, Lemma 1 shows that the following definition coincides with the traditional definition in the central case and provides a more general interpretation for splitting RDSs in the non-central case.

Definition 10. [7–p. 287] *An RDS R in E relative to N is a* splitting *RDS if E splits over N, that is, if there is a subgroup $H \leqslant E$ with $E = NH$ and $N \cap H = \{1\}$ (equivalently, if E is isomorphic to a semidirect product $N \rtimes_\rho E/N$).*

We derive the important splitting case of the Five-fold Constellation.

Theorem 5. *(Splitting Five-fold Constellation) Suppose $w|v$. Let $N \overset{\imath}{\rightarrowtail} E \overset{\pi}{\twoheadrightarrow} G$ be a split extension of N by G with associated equivalence class $[1, \varrho]$ of factor pairs, so $E \cong E_{(1,\varrho)} \cong N \rtimes_\varrho G$. The following five statements are equivalent:*

1. *the function $\phi : G \to N$ is PN relative to ϱ;*
2. *the splitting factor pair $(\partial\phi^{-1}, \overline{\phi}\varrho) \sim_\phi (1, \varrho)$ is orthogonal;*
3. *the transversal $R_\phi = \{(\phi(x), x) : x \in G\} \subseteq E_{(1,\varrho)}$ is a splitting $(v, w, v, v/w)$-RDS in $E_{(1,\varrho)}$ relative to $N \times \{1\}$;*

4. *the $(v, w, v, v/w)$-divisible design* $\mathrm{dev}(R_\phi)$*, class regular with respect to* $\imath(N)$*, has regular group* $N \rtimes_\varrho G$*;*
5. *the coupled G-developed matrix* $M_{(\phi,\varrho)} = [\,\phi(xy)^{\varrho(x^{-1})}\,]_{x,y\in G}$ *is a* $GH(w, v/w)$ *over* N*.*

Proof. $1 \Leftrightarrow 5$ is Definition 9. $5 \Leftrightarrow 2$ is the splitting case of [7–Theorem 10.1]. $2 \Leftrightarrow 3$ is the semiregular splitting case of [7–Theorem 5.1 and Corollary 5.1]. $3 \Leftrightarrow 4$ is an application of [12–Theorem 2.7].

When $\varrho \equiv 1$, the splitting equivalences $3 \Leftrightarrow 4 \Leftrightarrow 5$ of Theorem 5 are the original G-developed case of Jungnickel [12], using the traditional definition of splitting RDS. As we now see, Definition 10 is more appropriate.

There are pre-existing definitions of equivalence for factor pairs (Definition 2), for transversals (Definition 5), and for matrices with entries from a group, all arising naturally from theoretical considerations in each area. Even when comparison is possible, as with the items in Theorem 5, the types of equivalence do not coincide.

For instance, Theorem 2 shows that a bundle of splitting factor pairs is not, in general, contained in an equivalence class of splitting factor pairs.

However, it is relatively easy to verify that functions in the same bundle determine coupled G-developed matrices over N in the same equivalence class. That is,

$$\mathbf{b}(\phi, \varrho) = \mathbf{b}(\varphi, \varrho') \Rightarrow M_{(\phi,\varrho)} \sim M_{(\varphi,\varrho')}. \tag{23}$$

The equivalence operations on $M_{(\phi,\varrho)}$ determined by bundle action on $\partial_\varrho(\phi)$ are restricted (for instance, not all possible row or column permutations are applied) so that a single equivalence class of coupled G-developed matrices over N could contain the images of two, or more, distinct bundles of functions in $C^1(G, N)$.

The critical equivalence relation for our purposes is that for transversals. Theorem 1 and Corollary 2 then give us the following one-to-one mapping of bundles around four ‘stars’ of the Splitting Five-fold Constellation.

Corollary 4. *Under the conditions of Theorem 5, the mappings*

$$\mathbf{b}(\phi, \varrho) \leftrightarrow \mathcal{B}(\partial_\varrho(\phi)) \leftrightarrow [R_\phi] \leftrightarrow \{\mathrm{dev}(R_\varphi) : \varphi \in \mathbf{b}(\phi, \varrho)\}$$

define one-to-one correspondences between the corresponding sets of bundles of PN functions relative to ϱ*, bundles of orthogonal splitting factor pairs in* $F^2(G, N)$*, equivalence classes of semiregular RDSs in* E *relative to* N*, and equivalence classes of semiregular divisible designs with regular group* E*, class regular with respect to* N*.*

4 A Theory of Nonlinear Functions

The purpose of this section is to relate Definition 9 to the literature on PN and other nonlinear functions, and expand it to a framework for developing a general theory of nonlinear functions.

The situation for *abelian* PN functions (G and $N = C$ are abelian) is surveyed in [4] and [16]. The equivalence of abelian PN functions and G-developed generalised Hadamard matrices over C was first observed by de Launey [6].

In the binary case, when PN functions exist, they are also characterised by bentness, that is, they are maximally distant (in a specific sense) from linear functions. The measuring instrument is the Walsh-Hadamard Transform (WHT), and the function $\phi : V(n,2) \to V(m,2)$, with even $n \geq 2m$, is PN if and only if for each $c \neq 0 \in V(m,2)$ the component ϕ_c is bent; that is, if and only if for each $c \neq 0 \in V(m,2)$ the WHT $\widehat{\phi_c}$ of component ϕ_c takes only the values $\pm 2^{n/2}$. The analogue of this result holds for abelian PN functions $\phi : G \to C$, if the rôle of the WHT is taken by the Fourier Transform (FT) for the abelian group C and if, for each $c \in C$, the *component* ϕ_c is defined to be $\phi_c = \chi_c \circ \phi$.

Definition 11. [13] *Let C be a finite abelian group and suppose $\varphi : C \to \mathbb{C}$ takes values in the complex unit circle. Then φ is* bent *if its FT $\widehat{\varphi}$ has constant magnitude $|\widehat{\varphi}(x)| = \sqrt{|C|}$ for every $x \in C$.*

Pott [16] extends the definition of *maximal nonlinearity* from the binary case to the abelian case. As for bentness, this is a character-theoretic definition, which Pott gives in terms of the characters of a transversal of C in $C \times G$.

Definition 12. *Let G and C be finite abelian groups, let $\widehat{C \times G}$ be the character group of $C \times G$, let $\phi : G \to C$ and let $T_\phi = \{(\phi(x), x) :\ x \in G\} \subset C \times G$. The* maximum nonlinearity *of ϕ is $\mathcal{L}(\phi) = \max\{|\chi(T_\phi)| : \chi \neq \chi_0 \in \widehat{C \times G}\}$ and ϕ is* maximally nonlinear *if it attains the minimum possible value for $\mathcal{L}(\phi)$ for functions from G to C.*

Pott shows that $\mathcal{L}(\phi) \geq \sqrt{|G|}$. When $|C|$ divides $|G|$, he shows that functions with maximum nonlinearity coincide with PN functions by proving the transversal T_ϕ is a splitting abelian RDS. His proof invokes the dual definition, in terms of its characters, of an abelian RDS. He concludes that the transversal $T_\phi = \{(\phi(x), x) :\ x \in G\}$ is the correct instrument for measuring the nonlinear behaviour of any $\phi : G \to C$ between abelian groups.

Theorem 6. *Let G and C be abelian groups of orders v and w, respectively, where $w|v$, and $\phi \in C^1(G, C)$. The following are equivalent:*

1. *ϕ is PN;*
2. [4–Theorem 16] *for every $c \neq 1 \in C$ the component $\phi_c = \chi_c \circ \phi$ is bent; that is, its FT $\widehat{\phi_c}$ has magnitude $\widehat{\phi_c}(x) = \sqrt{v}$ for every $x \in G$;*
3. [16–Theorem 8] *ϕ is maximally nonlinear with maximal nonlinearity $\sqrt{v}$.*

These two characterisations of abelian PN functions (additional to Theorem 5) should still somehow hold true for our most general form of PN function. By (14) and Theorem 5.2, ϕ is PN relative to ϱ if and only if, in the group ring $\mathbb{Z}N$,

$$\forall\ x \neq 1 \in G, \quad \sum_{y \in G} \phi(y)^{\varrho(x)} \phi(xy)^{-1} = (v/w) \sum_{a \in N} a, \tag{24}$$

that is, if and only if, for every $x \neq 1 \in G$ and $a \in N$, the frequency

$$n_{(\phi,\varrho)}(x,a) = |\{y \in G : \ \phi(y)^{\varrho(x)}\phi(xy)^{-1} = a\}| = v/w. \tag{25}$$

However, if ϕ is a ϱ-twisted homomorphism, the left-hand side of (25) takes only two values: 0 (if $\phi(x)^{-1} \neq a$) and v (if $\phi(x)^{-1} = a$), so the frequency distributions, as $x \neq 1$ runs through G, are at opposite extremes: a sequence of delta-functions for twisted homomorphisms but of uniform distributions for PN functions.

How are we to capture this optimal difference of PN functions with respect to ϱ from ϱ-twisted homomorphisms?

Character theoretic techniques begin to falter when N is nonabelian, but we can replace the character group by $\mathrm{Hom}_\varrho(G,N)$ and test function $\phi : G \to N$ directly against all the ϱ-twisted homomorphisms $\chi : G \to N$. In matrix terms (c.f. (22)), we would compute $[\chi(xy)^{\varrho(x^{-1})}][\phi(y)^{-1}]^\top$.

Definition 13. *Let $w|v$, let $\varrho : G \to \mathrm{Aut}(N)$ be a homomorphism and let $\phi \in C^1(G,N)$. Then ϕ is* bent relative to ϱ *if, for all $x \neq 1 \in G$ and $\chi \in \mathrm{Hom}_\varrho(G,N)$, $\langle\chi,\phi\rangle(x) = \sum_{y\in G}\chi(xy)^{\varrho(x^{-1})}\phi(y)^{-1} = (v/w)\sum_{a\in N} \ a$.*

However the advantages of Fourier inversion and Transform may be lost if there is not a set of mutually orthogonal ϱ-twisted homomorphisms to work with.

Research Problem 1. *Develop the linear approximation (LA) theory of functions ϕ relative to ϱ, with bentness defined in Definition 13. How consistent is it with other approaches to this problem?*

Theorem 5 and Pott's approach suggest that maximal nonlinearity could reasonably be defined by existence of a splitting RDS, with the set to be measured for nonlinearity being the transversal $T_\phi = \{(\phi(x),x) : x \in G\}$ of N in an appropriate split extension of N by G. Then the optimal cases are given by [7–Theorem 5.1 and Corollary 5.1] (and Theorem 5 when $k = v$).

Definition 14. *Let $\phi \in C^1(G,N)$ and let $\varrho : G \to \mathrm{Aut}(N)$ be a homomorphism. Then ϕ is* maximally nonlinear relative to ϱ *if for some $k > 1$ there exists a k-subset D of G such that $R_\phi = \{(\phi(x),x) : x \in D\} \subset E_{(1,\varrho)}$ is a splitting (v,w,k,λ)-RDS relative to $N \times \{1\}$ lifting D.*

Research Problem 2. *Develop the difference distribution (DD) theory of functions ϕ relative to ϱ, with maximality defined in Definition 14. How consistent is it with other approaches to this problem?*

Acknowledgement. My thanks are due to the editors, who recommended enlarging the paper to make it more self contained, and to Alexander Pott, whose request for clarification of the relation between bundles and equivalent transversals when $\varrho \equiv 1$ led to inclusion of Theorem 3.

References

1. L. Breveglieri, A. Cherubini and M. Macchetti, On the generalized linear equivalence of functions over finite fields, in: ASIACRYPT 2004, ed. P. J. Lee, LNCS 3329 (2004) 79–91.
2. L. Budaghyan, C. Carlet and A. Pott, New classes of almost bent and almost perfect nonlinear polynomials, preprint, 2005.
3. C. Carlet, P. Charpin and V. Zinoviev, Codes, bent functions and permutations suitable for DES-like cryptosystems, *Des., Codes Cryptogr.* 15 (1998) 125–156.
4. C. Carlet and C. Ding, Highly nonlinear mappings, *J. Complexity* 20 (2004) 205–244.
5. R. S. Coulter and R. W. Matthews, Planar functions and planes of Lenz-Barlotti Class II, *Des., Codes Cryptogr.* 10 (1997) 167–184.
6. W. de Launey, Generalised Hadamard matrices which are developed modulo a group, *Discrete Math.* 104 (1992) 49–65.
7. J. C. Galati, A group extensions approach to relative difference sets, *J. Combin. Designs* 12 (2004) 279–298.
8. K. J. Horadam, Equivalence classes of central semiregular relative difference sets, *J. Combin. Des.* 8 (2000) 330–346.
9. K. J. Horadam, Hadamard Matrices, Princeton University Press, 9 Chapters, approx. 250pp, under review.
10. G. Hughes, Characteristic functions of relative difference sets, correlated sequences and Hadamard matrices, in: AAECC-13, LNCS 1719, Springer, Berlin, 1999, 346–354.
11. G. Hughes, The equivalence of certain auto-correlated quaternary and binary arrays, *Australas. J. Combin.*, 22 (2000) 37–40.
12. D. Jungnickel, On automorphism groups of divisible designs, *Can. J. Math.* 34 (1982) 257–297.
13. O. A. Logachev, A. A. Salnikov and V. V. Yashchenko, Bent functions on a finite abelian group, *Discrete Math. Appl.* 7 (1997) 547–564.
14. K. Nyberg, Perfect nonlinear S-boxes, in: EUROCRYPT-91, LNCS 547, Springer, New York, 1991, 378–385.
15. A. A. I. Perera and K. J. Horadam, Cocyclic generalised Hadamard matrices and central relative difference sets, *Des., Codes Cryptogr.* 15 (1998) 187–200.
16. A. Pott, Nonlinear functions in abelian groups and relative difference sets, *Discr. Appl. Math.* 138 (2004) 177–193.

The Solutions of the Third Power Sum Equation for Niho Type Decimations

Kalle Ranto* and Petri Rosendahl

Department of Mathematics, University of Turku,
20014 Turku, Finland
{kara, perosen}@utu.fi

Abstract. We will completely describe the solutions of the equation $(x+1)^d = x^d + 1$ in the field $GF(q^2)$, where $q = p^k$ and d is of Niho type, i.e., $d \equiv 1 \pmod{q-1}$. Our results have applications in the theory of cross-correlation functions of m-sequences and in the theory of cyclic codes.

1 Introduction

For the theory of finite fields in general we refer to [10]. For background in m-sequences and cyclic codes the reader should consult e.g. [7] and [13].

The finite field of order $q = p^k$, where p is a prime, will be denoted by $GF(q)$. The multiplicative group of $GF(q)$ will be denoted by $GF(q)^\times$.

In this paper, we will deal with the equation

$$(x+1)^d = x^d + 1 \tag{1}$$

in finite fields $GF(q^2)$. We will assume that d is of Niho type, i.e., it satisfies the so called Niho condition

$$d \equiv 1 \pmod{q-1}.$$

Cross-correlation functions of m-sequences corresponding to this type of decimations were first studied by Niho in his thesis [11], and hence the name. Recently, this kind of decimations (and exponents and corresponding power functions) have attracted great attention. For instance a new family of four-valued cross-correlation functions corresponding to Niho type decimations was found in [9]. In [2] it was proved that binary cyclic codes of length $2^n - 1$ with two non-zeros α^{-1} and α^{-d}, where α is a primitive element of $GF(2^n)$ and d is of Niho type, have at least four non-zero weights. As the last example we mention that properties of Niho type power functions were exploited in construction of bent functions in [6].

1.1 Motivation

To justify the study of the equation (1), we show how it is related to cross-correlation of m-sequences and cyclic codes. Many more examples can be given;

* Research supported by the Academy of Finland, grant 108238.

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 101–107, 2006.

see e.g. [5] for a connection to non-linearity properties of power functions and [4] for a connection to difference sets.

The polynomial $(x+1)^d - x^d - 1$ (1) has been studied before, especially in the binary case. However, not very many results of general nature are known. The interested reader should see e.g. [1] and [3].

Cross-Correlation of *m*-Sequences. The cross-correlation function $C_d(\tau)$ of two p-ary m-sequences u_i and u_{di}, $i = 0, 1, 2, \ldots$, of period $p^n - 1$ that differ by a decimation d is defined for $\tau = 0, 1, \ldots, p^n - 2$ by

$$C_d(\tau) = \sum_{i=0}^{p \ \ -2} \zeta^{u \ \ -u \ \ (\ + \ )},$$

where ζ is a complex primitive p-th root of unity.

To calculate the function $C_d(\tau)$ is essentially the same as to evaluate the character sums $\sum_{x \in GF(p \ \ )^\times} \chi(x + yx^d)$, where χ is the well known canonical additive character of the field $GF(p^n)$. This connection leads to the following result, which is useful in finding the distribution of the values. For a proof we refer to [11] and [7].

Theorem 1. *We have*

(i) $\sum_{\tau=0}^{p \ \ -2} (C_d(\tau) + 1) = p^n$
(ii) $\sum_{\tau=0}^{p \ \ -2} (C_d(\tau) + 1)^2 = p^{2n}$
(iii) $\sum_{\tau=0}^{p \ \ -2} (C_d(\tau) + 1)^3 = p^{2n} b,$

where b is the number of $x \in GF(q)$ such that

$$(x+1)^d = x^d + 1.$$

Thus the equation (1) occurs naturally in this context.

Cyclic Codes. Assume that a binary cyclic code of length $2^n - 1$ has a parity check matrix

$$\begin{pmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{2 \ \ -2} \\ 1 & \alpha^d & \alpha^{2d} & \cdots & \alpha^{(2 \ \ -2)d} \end{pmatrix},$$

where α is a primitive element of the field $GF(2^n)$.

The equation (1) is related to the number of codewords of weight three in the following way. Assume we have a codeword of weight three and shift it cyclically in such way that the first coordinate is 1. Now a codeword with 1 in the coordinates 1, i, and j occurs if and only if we have both $1 + \alpha^i + \alpha^j = 0$ and $1 + \alpha^{id} + \alpha^{jd} = 0$. To have a solution here is the same as to have a non-trivial solution for the equation (1).

1.2 Preliminary Facts and Notation

From now on we concentrate on the equation

$$(x+1)^d = x^d + 1. \tag{2}$$

We will assume that d is of Niho type, i.e.,

$$d \equiv 1 \pmod{q-1},$$

and we wish to find the solutions in the field $GF(q^2)$, where $q = p^k$ for a prime p. No further assumptions are made, that is, p is arbitrary and we do not assume $\gcd(d, q^2-1) = 1$ (as one would in the context of m-sequences).

The conjugate of an element $x \in GF(q^2)$ over $GF(q)$ will be denoted by $\overline{x}$, i.e.,

$$\overline{x} = x^q.$$

The conjugation operation has properties similar to ordinary complex conjugation. We have for example $\overline{x+y} = \overline{x} + \overline{y}$.

An important role will be played by the set

$$S = \{x \in GF(q^2) : x\overline{x} = 1\}.$$

Note that S is a cyclic group of order $q+1$.

2 The Third Power Sum for Niho Type Exponents

In what follows, we will give a complete solution to the equation (2). The techniques we use were essentially developed in [8] and [12], where some special cases of our main result were presented.

The first thing we note is that if x is in the subfield $GF(q)$ then it automatically satisfies (2). This is because $x^{q-1} = 1$ for $x \in GF(q)^\times$.

Lemma 2. *Assume that $d = (q-1)s+1$ and that $x \in GF(q^2)^\times$ is a solution to*

$$(x+1)^d = x^d + 1.$$

Then $z = x^{q-1}$ satisfies $z^s = 1$ or $z^{s-1} = 1$.

Proof. Since

$$(x+1)^d = x^d + 1,$$

we also have

$$(\overline{x}+1)^d = \overline{x}^d + 1.$$

Multiplying these equations gives

$$(x\overline{x} + x + \overline{x} + 1)^d = (x\overline{x})^d + x^d + \overline{x}^d + 1. \tag{3}$$

Clearly $x\overline{x}, x+\overline{x} \in GF(q)$ and therefore also $x\overline{x}+x+\overline{x}+1 \in GF(q)$. Since for $a \in GF(q)$ we have $a^d = a$, (3) implies

$$x + \overline{x} = x^d + \overline{x}^d.$$

Divided by x this becomes

$$1 + x^{q-1} = x^{d-1} + x^{qd-1}.$$

Let $z = x^{q-1}$. Since $z^{q+1} = 1$, we get

$$1 + z = z^s + z^{1-s},$$

which is equivalent to

$$(z^s - 1)(z^{s-1} - 1) = 0, \tag{4}$$

from which the claim follows.

The key idea here is that z is an element of S. Therefore (4) implies in fact that $z^{\gcd(s,q+1)} = 1$ or $z^{\gcd(s-1,q+1)} = 1$.

Lemma 3. *Assume that q is fixed.*

- *(i) Let $x \in GF(q^2) \setminus \{0,-1\}$, and denote $z = x^{q-1}$ and $w = (x+1)^{q-1}$. Then x is a solution to (2) if and only if $z^s = w^s = 1$ or $z^{s-1} = w^{s-1} = 1$.*
- *(ii) The set of solutions to (2) depends only on the pair $\{gcd(s,q+1), \gcd(s-1,q+1)\}$, not on the specific choice of s. More precisely, let $e = (q-1)t+1$ and assume that either*

$$\begin{cases} \gcd(s,q+1) = \gcd(t,q+1) \\ \gcd(s-1,q+1) = \gcd(t-1,q+1) \end{cases}$$

or

$$\begin{cases} \gcd(s,q+1) = \gcd(t-1,q+1) \\ \gcd(s-1,q+1) = \gcd(t,q+1). \end{cases}$$

Then $x \in GF(q^2)$ is a solution to (2) if and only if x satisfies $(x+1)^e = x^e + 1$.

Proof. (*i*) If $x \neq 0$ is a solution to (2), then by the previous lemma $z^s = 1$ or $z^{s-1} = 1$.

Assume first that $z^s = 1$. Then $x^d = xz^s = x$ and we get from $(x+1)^d = x^d + 1$ that $w^s = 1$.

If in turn $z^{s-1} = 1$, we have $x^d = \overline{x}z^{s-1} = \overline{x}$. Therefore $(x+1)^d = x^d + 1$ implies $(x+1)^{d-q} = 1$, i.e., $w^{s-1} = 1$.

If $z^s = w^s = 1$ or $z^{s-1} = w^{s-1} = 1$, then obviously x satisfies (2).

(*ii*) This follows easily from (*i*) and the fact that both z and w are elements of S. As an illustration, assume that $\gcd(s,q+1) = \gcd(t,q+1)$ and $\gcd(s-1,q+1) = \gcd(t-1,q+1)$. If $x \neq 0,-1$ satisfies (2) then $z^s = w^s = 1$ or $z^{s-1} = w^{s-1} = 1$. We then have $z^t = w^t = 1$ or $z^{t-1} = w^{t-1} = 1$, because of the assumption on the greatest common divisors. From (*i*) we deduce $(x+1)^e = x^e + 1$. The remaining details are left to the reader.

Lemma 4. *If $x \in GF(q^2) \setminus GF(q)$ satisfies the equation* (2) *then x can be represented as*

$$x = \frac{w-1}{z-w}$$

for some $z, w \in S \setminus \{1\}$ such that

(*i*) *either $z^s = w^s = 1$ or $z^{s-1} = w^{s-1} = 1$*
(*ii*) *$z \neq w$*

Proof. We denote $x^{q-1} = z$ and $(x+1)^{q-1} = w$. Then clearly $\overline{x} = xz$ and

$$\overline{x+1} = (x+1) \cdot w,$$

which implies $\overline{x} = wx + w - 1$. Equating the two forms for $\overline{x}$ we get $xz = wx + w - 1$. Solving x gives the desired form.

The condition (*i*) is obvious because of Lemma 3. Also, $z = w$ would imply $z = w = 1$ and then $x \in GF(q)$ contrary to assumption.

The latter part of Lemma 3 says that s and $s-1$ are in symmetrical roles. Thus we consider s only in what follows.

Lemma 5. *Let $w, z \in S \setminus \{1\}$, $z \neq w$ and assume $z^s = w^s = 1$. Set*

$$x_0 = \frac{w-1}{z-w}$$

Then x_0 is a solution to the equation (2) *and $x_0 \notin GF(q)$.*

Proof. We have

$$\overline{x_0} = \frac{\overline{w}-1}{\overline{z}-\overline{w}} = \frac{w^{-1}-1}{z^{-1}-w^{-1}} = \frac{z-zw}{w-z} = x_0 z,$$

and hence $x_0^{q-1} = z$.

Now let

$$x_1 = x_0 + 1 = \frac{z-1}{z-w}.$$

A similar computation as above shows that

$$x_1^{q-1} = w.$$

Now

$$x_0^d = x_0^{s(q-1)+1} = x_0 \cdot z^s = x_0$$

and similarly $x_1^d = x_1$. Hence

$$(x_0+1)^d = x_1^d = x_1 = x_0 + 1 = x_0^d + 1,$$

which shows that x_0 satisfies (2).

To end the proof we only have to note that $x_0 \notin GF(q)$, but this is clear from $x_0^q = x_0 z$ since $z \neq 1$.

Lemma 6. *The elements of the form*

$$x = \frac{w-1}{z-w}$$

are distinct whenever $z, w \in S \setminus \{1\}$ and $z \neq w$.

Proof. Assume that we have

$$x = \frac{w-1}{z-w} = \frac{v-1}{u-v} \tag{5}$$

for some $z, w, u, v \in S \setminus \{1\}$ with $z \neq w$ and $u \neq v$.

Then

$$\overline{x} = \frac{\frac{1}{w} - 1}{\frac{1}{z} - \frac{1}{w}} = \frac{zw - z}{z - w},$$

and similarly

$$\overline{x} = \frac{uv - u}{u - v}.$$

We now calculate the element $\overline{x}/x$ in two ways. Firstly,

$$\frac{\overline{x}}{x} = \frac{zw - z}{z - w} \cdot \frac{z - w}{w - 1} = z$$

and similarly

$$\frac{\overline{x}}{x} = u.$$

Hence we have $z = u$.

By cross-multiplying we get from (5) that

$$uw - u - vw + v = zv - z - vw + w,$$

and using $z = u$ we get immediately that $w = v$.

We are now ready to state the main result of the paper.

Theorem 7. *Let $d = (q-1) \cdot s + 1$, $r_0 = \gcd(s, q+1)$, and $r_1 = \gcd(s-1, q+1)$. Consider the equation*

$$(x+1)^d = x^d + 1$$

in the field $GF(q^2)$.

(*i*) *The solutions to the equation are exactly the elements of the subfield $GF(q)$ and the elements of the form*

$$x = \frac{w-1}{z-w},$$

where $w, z \in S \setminus \{1\}$ satisfy $z \neq w$ and either $z^s = w^s = 1$ or $z^{s-1} = w^{s-1} = 1$.

(ii) The number of solutions to the equation is

$$b = q + (r_0 - 1)(r_0 - 2) + (r_1 - 1)(r_1 - 2)\,.$$

Proof. The first part is already proven. The second part is an elementary counting argument. We only have to mention that $z^s = 1$ (resp. $z^{s-1} = 1$) implies $z^{r_0} = 1$ (resp. $z^{r_1 - 1} = 1$), and that the equations $z^s = 1$ and $z^{s-1} = 1$ have no common solutions other than 1.

A similar result holds of course when $d \equiv p^i \pmod{q-1}$ for some i. In fact the solutions remain the same, only their multiplicities change.

References

1. A. Canteaut, P. Charpin, and H. Dobbertin. Weight divisibility of cyclic codes, highly nonlinear functions on $\mathbf{F}_2$, and crosscorrelation of maximum-length sequences. *SIAM J. Discrete Math.*, 13(1):105–138, 2000.
2. P. Charpin. Cyclic codes with few weights and Niho exponents. *J. Combin. Theory Ser. A*, 108(2):247–259, 2004.
3. P. Charpin, A. Tietäväinen, and V. A. Zinoviev. On binary cyclic codes with minimum distance three. *Problems of Information Transmission*, 33(4):287–296, 1997.
4. J. F. Dillon and H. Dobbertin. New cyclic difference sets with Singer parameters. *Finite Fields Appl.*, 10(3):342–389, 2004.
5. H. Dobbertin. Almost perfect nonlinear power functions on GF(2^n): the Welch case. *IEEE Transactions on Information Theory*, 45(4):1271–1275, 1999.
6. H. Dobbertin, G. Leander, A. Canteaut, C. Carlet, P. Felke, and P. Gaborit. Construction of bent functions via Niho power functions. Submitted.
7. T. Helleseth. Some results about the cross-correlation function between two maximal linear sequences. *Discrete Mathematics*, 16(3):209–232, 1976.
8. T. Helleseth, J. Lahtonen, and P. Rosendahl. On Niho type cross-correlation functions of m-sequences. To appear in Finite Fields and Their Applications.
9. T. Helleseth and P. Rosendahl. New pairs of m-sequences with four-level cross-correlation. To appear in Finite Fields and Their Applications.
10. R. Lidl and H. Niederreiter. *Finite Fields*, volume 20 of *Encyclopedia of Mathematics and Its Applications*. Addison-Wesley, Reading, 1983.
11. Y. Niho. *Multivalued cross-correlation functions between two maximal linear recursive sequences*. PhD thesis, University or Southern California, 1972.
12. P. Rosendahl. *Niho type cross-correlation functions and related equations*. PhD thesis, University of Turku, 2004. Available at http://www.tucs.fi/.
13. J. H. van Lint. *Introduction to coding theory*, volume 86 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, third edition, 1999.

On Constructing AG Codes Without Basis Functions for Riemann-Roch Spaces

Drue Coles

Department of Mathematics, Computer Science, and Statistics,
Bloomsburg University
dcoles@bloomu.edu

Abstract. The standard construction of linear error-correcting codes on algebraic curves requires determining a basis for the Riemann-Roch space $\mathcal{L}(G)$ associated to a given divisor G, often a hard problem. Here we consider the problem of constructing the code without any knowledge of such a basis. We interpret the columns of a generator matrix as points on an embedded copy of the curve, and show that in certain cases these points can be realized in principle as the images of a set of vector bundles under a standard map to a class of repartitions.

1 Introduction

Let C denote a smooth projective algebraic curve of genus γ defined over a finite field k. Fix a divisor $D = P_1 + \cdots + P_n$ on C, where each point P_i is rational (over k), and let G be another divisor of degree α with rational support disjoint from that of D. The algebraic-geometric (AG) code given by these two divisors is defined to be

$$C(D, G) = \{(f(P_1), \ldots, f(P_n)) : f \in \mathcal{L}(G)\}\,.$$

The code is linear over k, and if $\alpha \geq 2\gamma - 1$ then it has dimension $\alpha - \gamma + 1$ by the Riemann-Roch theorem. The minimum distance is at least $n - \alpha$, since a non-zero function in $\mathcal{L}(G)$ has at most α zeros. AG codes were discovered by Goppa [5] in the early 1980s, and since that time many important practical and theoretical advances have been made, including efficient decoding algorithms (surveyed, for example, in [8]) and polynomial-time constructable codes that beat the Gilbert-Varshamov bound [15, 6, 13].

Usually the divisor G is taken as a multiple of a single point, and here as well we let $G = \alpha Q$ for a rational point $Q \in C$. We assume $\alpha > 2\gamma$ so that the rational map $\varphi : C \to \mathbb{P}^m$ $(m = \alpha - \gamma)$ given by the complete linear system $|\alpha Q|$ is an embedding [12–Ch. III, Sect. 6.6].

In practice, a linear code is constructed by computing a generator matrix for it. If $\{f_i\}$ is a basis for $\mathcal{L}(G)$, then we get the rows of a generator matrix for $C(D, G)$ by computing the linearly independent codewords $(f_i(P_1), \ldots, f_i(P_n))$. Equivalently, we can view the points $\varphi(P_i)$ as columns of this matrix.

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 108–117, 2006.

Computing a basis for a Riemann-Roch space, however, is often a difficult problem. This note describes the theoretical framework for an alternative method of one-point AG code construction. It applies only for certain choices of the point Q, but from a coding standpoint the choice of Q is immaterial since the dimension and distance of an AG code depend only on the *degree* of the divisors used.

The basic idea is to map extensions of lines bundles determined by Q and the points P_i into a class of repartitions via standard sheaf cohomology; elements of the class can be uniquely expressed as a linear combination of certain fixed repartitions of a very simple form, and the coefficients in that combination are precisely the coordinates of $\varphi(P_i)$.

The next section reviews some background material and establishes notation. After proving the main result in the third section, we point out that it can be used to determine the Weierstrass non-gaps at Q, and we demonstrate this fact with an example on the Klein curve. The last section looks at computational aspects of the cohomology maps used in the main result and the algorithmic details that would need to be worked out for an explicit implementation.

2 Background and Notation

Fix a smooth projective curve C of genus γ over a finite field k for the rest of the paper. Let $\bar{k}$ denote the algebraic closure of k. We refer to k-rational points simply as rational points for brevity.

Fix a rational point $Q \in C$ and an integer $\alpha > 2\gamma$, and let φ denote the embedding of the curve determined by the complete linear system $|\alpha Q|$. The goal is to compute $\varphi(P)$ for rational points $P \neq Q$ on the curve without knowing a basis for $\mathcal{L}(\alpha Q)$. In that way, we get the columns of a generator matrix for $C(D, \alpha Q)$, where as usual D is the formal sum of all rational points other than Q. The code has length $n = |Supp(D)|$, and we assume $n > \alpha$.

For $f \in \bar{k}(C)$ and $P \in C$, $\nu_P(f)$ denotes the order of f at P, and $\mathcal{O}_P$ denotes the local ring at P.

We enumerate the Weierstrass non-gaps at Q by $\mu_0, \mu_1, \mu_2, \ldots$

2.1 Extension Spaces

For any rank-2 vector bundle $E \to C$ of degree at least γ, there is line bundle L such that the sequence $0 \to \mathcal{O}_C \to E \to L \to 0$ is exact. Here E is called an extension of L by $\mathcal{O}_C$. Another such extension E' is isomorphic to E if there is an isomorphism of exact sequences

$$\begin{tikzcd}
0 \arrow[r] & \mathcal{O}_C \arrow[r] \arrow[d, equal] & E \arrow[r] \arrow[d, "\cong"'] & L \arrow[r] \arrow[d, equal] & 0 \\
0 \arrow[r] & \mathcal{O}_C \arrow[r] & E' \arrow[r] & L \arrow[r] & 0.
\end{tikzcd}$$

We denote by $Ext(L, \mathcal{O}_C)$ the space of extension classes of L by $\mathcal{O}_C$. This has the structure of a linear space over $\bar{k}$ [3–Sect. 5.7].

An extension $E \in Ext(L, \mathcal{O}_C)$ corresponds to an element of $H^1(L^{-1})$ as follows: twist by L^{-1} to get an exact sequence $0 \to L^{-1} \to E \otimes L^{-1} \to \mathcal{O}_C \to 0$, form the associated long exact sequence and take the image of the identity element in $H^0(\mathcal{O}_C)$. Some authors actually define first cohomology directly in terms of extensions, and there is a geometric way of defining a group operation on them [3–Sect. 5.7].

Now $H^1(L^{-1}) \cong H^0(\omega_C \otimes L)^*$ by Serre duality, so we have identified the space $Ext(L, \mathcal{O}_C)$ (modulo scalers) with projective space $\mathbb{P}(H^0(\omega_C \otimes L)^*)$.

2.2 The Segre Invariant and Secant Varieties

The s-invariant of a non-split rank-2 vector bundle E on a smooth projective curve is defined by

$$s(E) = \deg E - 2 \max\{\deg M : M \hookrightarrow E\},$$

where M runs over all line subbundles of E. Let (e) denote the rank-2 extension E viewed as a point of projective space $\mathbb{P}$. Lange and Narasimhan [10], following Atiyah [1], showed that $s(E)$ is determined by the smallest integer j such that (e) is contained in the j-secant variety of the curve in $\mathbb{P}$. This picture was also described by Bertram [2], and later Trygve Johnsen observed that it leads to an interpretation of decoding AG codes in terms of vector bundles on the underlying curve [9].

It turns out that $(e) = \varphi(P)$ if and only if $\mathcal{O}_C(P)$ is a quotient line bundle of E. This fact is really just a special case of [10–Proposition 1.1], made more explicit by [9–Proposition 2.5]. Summarizing with the notation established at the beginning of this section, we have:

Proposition 1. *Let $L = \mathcal{O}_C(\alpha Q - K)$. For any point $P \in C$, there is a unique extension $E \in Ext(L, \mathcal{O}_C)$ with quotient line bundle $\mathcal{O}_C(P)$, and E corresponds to $\varphi(P)$ as a point of projective space with respect to the embedding φ determined by $|\alpha Q|$.*

2.3 Repartitions

A repartition r associates to each point $P \in C$ a function $r_P \in \bar{k}(C)$, with $r_P \in \mathcal{O}_P$ for all but finitely many points. R denotes the ring of repartitions, and $\bar{k}(C)$ is viewed as a subring by identifying $f \in \bar{k}(C)$ with the repartition that assigns f to each point of the curve. If $f \in \bar{k}(C)$ and $Q \in C$, then we write f/Q for the repartition that assigns f to Q and zero to every other point.

Given a divisor A, $R(A)$ denotes the additive subgroup of repartitions $r \in R$ satisfying $\nu_P(r_P) + \nu_P(A) \geq 0$ for every point $P \in C$. There is a canonical isomorphism

$$H^1(\mathcal{O}_C(A)) \cong R / \left(R(A) + \bar{k}(C)\right).$$

Serre proves this fact [14–Proposition II.3] and uses it to prove the duality theorem for curves.

For a divisor G, we therefore have

$$H^0(\mathcal{O}_C(G))^* \;\cong\; H^1(\mathcal{O}_C(K-G)) \;\cong\; R/\big(R(K-G)+\bar{k}(C)\big)$$

When G is a multiple of a single point, there is a basis for $R/(R(K-G)+\bar{k}(C))$ consisting of repartitions of an especially simple form; using our notation, this result can be stated as:

Proposition 2. *For a local parameter t at Q, define the differential $\omega = dt$ in a neighborhood of Q and let $K = (\omega)$. Then the set $\{t^{\mu\ \ -1}/Q : 0 \leq i \leq m\}$ is a basis for the vector space $R/(R(K-\alpha Q)+\bar{k}(C))$ over $\bar{k}$.*

This was used to compute a transition matrix for the rank-2 extension that corresponds as a point in projective space to the syndrome of a corrupted codeword [4–Proposition 1]. The proof is repeated in the appendix of this paper.

3 Constructing a Generator Matrix

Recall that the columns of a generator matrix for $C(D,\alpha Q)$ are given explicitly by the points $\varphi(D)$, where φ is the embedding of C determined by $|\alpha Q|$.

Theorem 1. *For a local parameter t at Q, define the differential $\omega = dt$ in a neighborhood of Q and let $K = (\omega)$. Suppose there is a basis $S = \{f_1,\ldots,f_\gamma\}$ for $\mathcal{L}(K)$ consisting of functions that have a single non-zero term of degree less than α when expanded in powers of t. Then for any point $P \neq Q$ on the curve, $\varphi(P) = (c_0 : \cdots : c_m)$ if and only if the repartition $(c_0 t^{\mu_0 -1} + \cdots + c_m t^{\mu\ \ -1})/Q$ corresponds to the unique extension $E \in Ext(\mathcal{O}_C(\alpha Q - K), \mathcal{O}_C)$ with quotient line bundle $\mathcal{O}_C(P)$.*

Proof. Since ω is regular and non-zero at Q by definition, any function in $\mathcal{L}(K)$ is regular at Q. Distinct functions $f_i, f_j \in S$ have distinct orders at Q, for otherwise $\nu_Q(f_i - f_j) \geq \alpha$ by the hypothesized property of elements of S, which would not be possible since $f_i - f_j \in \mathcal{L}(K)$ can have at most $2\gamma - 2 < \alpha$ zeros.

Let $n_i = \nu_Q(f_i)$. We will show that $t^{n\ } /Q \in R(K-\alpha Q)+\bar{k}(C)$ for $1 \leq i \leq \gamma$. Multiplying each f_i by a constant if necessary, we can write

$$f_i \;=\; t^i + \sum_{j=\alpha}^{\infty} b_{ij} \cdot t^j$$

with uniquely determined coefficients $b_{ij} \in \bar{k}$. Now for each function $f_i \in S$, we define a repartition r_i by

$$(r_i)_P \;=\; \begin{cases} f_i - t^{n} & : \; P = Q \\ \quad -f_i & : \; P \neq Q. \end{cases}$$

We have $\nu_Q((r_i)_Q) \geq \alpha$, and since $f_i \in \mathcal{L}(K)$ it follows that $r_i \in R(K-\alpha Q)$. Now $r_i + f_i = t^{n\ } /Q$, so $t^{n\ } /Q \in R(K-\alpha Q)+\bar{k}(C)$ as claimed.

Let $A = \{t^i/Q : 0 \leq i \leq \alpha\}$. We have just shown that there are γ elements of A that are zero in the space $R/R(K - \alpha Q) + \bar{k}(C))$, namely each $t^{n} \ /Q$. On the other hand, Proposition 2 provides a basis for this space consisting of the remaining $m + 1 = \alpha - \gamma + 1$ elements of A. Consequently, any element of $R/R(K - \alpha Q) + \bar{k}(C))$ can be uniquely expressed as a linear combination of the particular $m + 1$ elements of A specified in the proposition.

By Proposition 1, there is a unique extension $E \in Ext(\mathcal{O}_C(\alpha Q - K), \mathcal{O}_C)$ with quotient bundle $\mathcal{O}_C(P)$; the extension corresponds via Serre duality to the point $\varphi(P) \in \mathbb{P}(H^0(\mathcal{O}_C(\alpha Q))^*)$, and the isomorphism

$$H^0(\mathcal{O}_C(\alpha Q))^* \quad \xrightarrow{\sim} \quad R/\left(R(K - \alpha Q) + \bar{k}(C)\right)$$

is realized by $(c_0, \ldots, c_m) \mapsto \left(c_0 t^{\mu_0 - 1} + \cdots + c_m t^{\mu \quad -1}\right)/Q$. □

3.1 Computing Weierstrass Non-gaps

Suppose that for some fixed positive integer $i < \alpha$, the repartition corresponding to a given point $\varphi(P)$ is simply t^i/Q. This means that $\varphi(P)$ has a single non-zero coordinate. But we do not know *which* coordinate is non-zero unless we also happen to know where the integer i lies among the exponents $\mu_0 - 1, \ldots, \mu_m - 1$, or in other words, unless we know the Weierstrass non-gaps at Q.

Consider, however, the set of all points $\varphi(P_1), \ldots, \varphi(P_n)$. It is not possible that all of them are zero at the same coordinate, for otherwise there would exist a non-zero function in $\mathcal{L}(\alpha Q)$ that vanishes at $n > \alpha Q$ points. We can therefore think of Theorem 1 as *determining* the non-gaps at Q. This is illustrated below. Of course, this observation does not translate directly into an effective algorithm for computing the non-gaps at certain points since the cohomology maps that we are using have been described in a completely abstract fashion. Section 4 discusses the algorithmic aspects of Theorem 1.

3.2 Illustration

Let C denote the Klein curve of genus 3 defined by $x^3y + y^3z + z^3x = 0$. We will use Theorem 1 with $\alpha = 5$ (the highest possible non-gap) to compute the Weierstrass non-gaps at $Q_1 = (1 : 0 : 0)$. Let $Q_2 = (0 : 1 : 0)$ and $Q_3 = (0 : 0 : 1)$. These latter two points also lie on the curve. Computing the intersection divisors of the three lines $xyz = 0$ with the curve, we have:

$$\begin{aligned} div(x) &= 3Q_3 + Q_2 \\ div(y) &= 3Q_1 + Q_3 \\ div(z) &= 3Q_2 + Q_1 \end{aligned}$$

We can use this information to obtain the order of a monomial at any of the points Q_i. In particular, we see that $t = z/x$ is a local parameter at Q_1.

Define the differential $\omega = dt$ in the open set $U = \{(x : y : z) \in C : x \neq 0\}$, and let $K = (\omega)$.

Lemma 1. $K = 3Q_3 + Q_2$.

Proof. A point $P \in U$ has the form $P = (1 : b : c)$. Note that $t' = (z + cx)/x$ is a local parameter at P, and $dt' = dt$. It follows that ω has no zeros on U, so the support of K must be contained in $C \setminus U = \{Q_2, Q_3\}$. A canonical divisor on a plane quartic is the intersection divisor of a line with the curve, and the only such divisor supported by Q_2 and Q_3 is the intersection divisor of the line $x = 0$ with the curve; that is, $3Q_3 + Q_2$. □

Lemma 2. *The set* $\{1,\ z/x,\ y/x\}$ *is a basis for* $\mathcal{L}(K)$.

Proof. The dimension of $\mathcal{L}(K)$ is 3 (genus), and we see that the given functions are contained in $\mathcal{L}(K)$ by checking the intersection divisors of the lines $xyz = 0$ with the curve. For linear independence, note that the functions have distinct orders at Q_1. □

We want to verify that the basis given by the preceding lemma satisfies the hypothesis of Theorem 1; that is, the basis functions have a single non-zero term of degree less than $\alpha = 5$ when expanded in powers of $t = z/x$. Obviously, we only need to look at y/x, which vanishes at Q_1 with order 3. The first term in the expansion of y/x is t^3, and to determine the next term we compute

$$y/x - t^3 \;=\; (x^3y + xz^3)/x^4 \;=\; y^3z/x^4.$$

Since $\nu_{Q_1}(y^3z/x^4) = 10$, the expansion of y/x in powers of t has exactly one non-zero term of degree less than α. The basis functions therefore satisfy the hypothesis of Theorem 1. They vanish at Q_1 with orders 0, 1 and 3, and the proof of Theorem 1 show that in this case

$$\{1/Q_1,\ t/Q_1,\ t^3/Q_1\} \;\subset\; R/R(K - 5Q_1) + \overline{k}(C)).$$

On the other hand, $\{t^{\mu\ -1} : 0 \leq i \leq 2\}$ is a basis for $R/(R(K - 5Q) + \overline{k}(C))$ according to Proposition 2, and the values 0, 1 and 3 have been excluded as possible values for $\mu_i - 1$, leaving -1, 2 and 4. The non-gaps μ_i at Q_1 are therefore 0, 3 and 5.

Indeed, the set $\{1,\ x/y,\ xz/y^2\}$ is a basis for $\mathcal{L}(5Q_1)$, and the functions in this basis have pole orders 0, 3, and 5 at Q_1.

4 Algorithmic Questions

Theorem 1 provides the theoretical framework for computing the points $\varphi(P)$ via cohomology maps, but three main computational problems must be solved to apply the theorem in practice:

1. Compute a concrete representation of the unique extension

$$E \in Ext(\mathcal{O}_C(\alpha Q - K), \mathcal{O}_C)$$

with quotient bundle $\mathcal{O}_C(P)$ for each rational point $P \neq Q$.

2. Compute the image of E under the map
$$\psi \ : \ Ext\,(\mathcal{O}_C(K-\alpha Q), \mathcal{O}_C) \ \xrightarrow{\sim} \ R/\left(R(K-\alpha Q) + \bar{k}(C)\right).$$
3. If $\psi(E) \in R/(R(K-\alpha Q)+\bar{k}(C))$ is not of the form $\sum_{i=0}^{\alpha} c_i t^{\mu\ -1}/Q$, then translate it into the unique representative of its equivalence class having that form.

We look briefly at each of these questions in turn.

4.1 Concrete Representations of Rank-2 Extensions

The problem is to find the extension $E \in Ext(\mathcal{O}_C(\alpha Q - K), \mathcal{O}_C)$ with quotient line bundle $\mathcal{O}_C(P)$, or equivalently with line subbundle $\mathcal{O}_C(\alpha Q - K - P)$. Here we quickly review the idea of a vector bundle as an abstract algebraic variety, and a subbundle as an embedding of varieties. The basic facts can be looked up in Shafarevich [12–Chap. VI]. Then following the definitions, we translate the problem of finding the desired rank-2 bundle into a search for rational functions on the curve satisfying a certain linear relation.

Consider a line bundle $L \to C$, say $L = \mathcal{O}_C(A)$ for a divisor A. Since the base space C is a curve, there is a covering by two open sets with L trivial over each. Fix such a covering (U_1, U_2), and let $U_{12} = U_1 \cap U_2$. Then L is realized as an abstract algebraic variety by glueing the two affine varieties $U_i \times \bar{k}$ along their intersection. In particular, it is represented by a transition function $h \in \mathcal{O}_C(U_{12})^*$ that for each $x \in U_{12}$ identifies the point $(x, a) \in U_1 \times \bar{k}$ with the point $(x, h(a)) \in U_2 \times \bar{k}$. If A has local equations h_i in U_i, then we may take $h = h_2/h_1$. Then an extension $E \in Ext(L, \mathcal{O}_C)$ is represented by a transition matrix
$$M = \begin{pmatrix} 1 & 0 \\ g & h \end{pmatrix},$$
where $g \in \bar{k}(C)$ depends on the class of E. This matrix determines the glueing relation for the affine varieties $U_1 \times \bar{k}^2$ and $U_2 \times \bar{k}^2$.

Considering vector bundles as abstract algebraic varieties, an embedding
$$\varphi \ : \ L \to E$$
is a regular map of varieties that preserves fibers, and on each fiber induces a linear map $\bar{k} \to \bar{k}^2$. This means that there are regular functions r_i and s_i on U_i $(i = 0, 1)$ such that $\varphi|_{U\ \times k} : (x, a) \mapsto (x, r_i(x) \cdot a, s_i(x) \cdot a)$, and these functions preserve the gluing relation of E; in our case, it means
$$f \cdot (r_2, s_2) = \begin{pmatrix} 1 & 0 \\ g & h \end{pmatrix} \begin{pmatrix} r_1 \\ s_1 \end{pmatrix}$$
on the intersection U_{12}, where f is a transition function for $L = \mathcal{O}_C(\alpha Q - K - P)$. We may take $f = hp^{-1}$, where p is a transition function for $\mathcal{O}_C(P)$. Note that for suitable choices of the trivializing cover $\{U_i\}$, it is easy to obtain local equations for the rational points of the curve, or for any divisor with rational support.

In summary, we get a transition matrix for $E \in Ext(\mathcal{O}_C(\alpha Q - K), \mathcal{O}_C)$ with quotient line bundle $\mathcal{O}_C(P)$ by finding $g \in \bar{k}(C)$ and $r_i, s_i \in \mathcal{O}_C(U_i)$ that

$$\begin{aligned} f \cdot r_2 &= r_1 \\ f \cdot s_2 &= g \cdot r_1 + h \cdot s_1 \end{aligned}$$

on the intersection U_{12}.

4.2 Cohomology Maps

Our map ψ is a composition of three maps

$$\begin{aligned} Ext(\mathcal{O}_C(\alpha Q - K), \mathcal{O}_C) &\longrightarrow Ext(\mathcal{O}_C, \mathcal{O}_C(K - \alpha Q)) && (1) \\ &\longrightarrow H^1(\mathcal{O}_C(K - \alpha Q)) && (2) \\ &\longrightarrow R/(R(K - \alpha Q) + \bar{k}(C)) && (3) \end{aligned}$$

The first map takes an extension of the form $0 \to \mathcal{O}_C \to E \to \mathcal{O}_C(\alpha Q - K) \to 0$ and twists by $\mathcal{O}_C(K - \alpha Q)$ to obtain $0 \to \mathcal{O}_C(K - \alpha Q) \to E' \to \mathcal{O}_C \to 0$. What this means concretely, in terms of transition matrices, is multiplying each entry of a transition matrix for E by a transition function for $\mathcal{O}_C(\alpha Q - K)$ to obtain a transition matrix for E'.

The second map arises by forming the associated long exact sequence and taking the image of the identity element under $H^0(\mathcal{O}_C) \to H^1(\mathcal{O}_C(K - \alpha Q))$. But here the picture seems too abstract for direct computation. It may simplify matters to use the Cech cohomology, but care is needed to choose a suitable open cover so that the Cech cohomology agrees with the derived functor cohomology; see Hartshorne [7–Chap. III] on this point.

The map to repartitions (3) is described by Serre [12–Proposition II.3], and there also one may need to recast the details in terms of Cech cohomology for a more computationally concrete picture.

4.3 Translating Repartitions

Let us define the *support* of a repartition r to be the set of points P at which $r_P \neq 0$. We begin by noting that for any divisor A, a repartition with infinite support is equivalent modulo $R(A)$ to one with finite support: for example, if $r' \in R$ is defined by

$$r'_P = \begin{cases} r_p & : \ P \notin Supp\,(A) \text{ and } \nu_P(r_p) \geq 0 \\ 0 & : \ \text{otherwise} \end{cases}$$

then $r - r' \equiv r$ has finite support. Recall that we identify $f \in \bar{k}(C)$ with the repartition that assigns f to every point. If we take $f = r_Q$, then $r - f$ does not have Q in its support and it is equivalent to r modulo $R(K - \alpha Q) + \bar{k}(C)$. We may therefore assume that a given repartition r, viewed as an element of $R/(R(K - \alpha Q) + \bar{k})$, is supported by finitely many points, Q not among them.

Proposition 1 says that any $r \in R$ is equivalent modulo $R(K - \alpha Q) + \bar{k}(C)$ to a linear combination of the repartitions $t^{\mu\ -1}/Q$ for $0 \leq i \leq m$. This means that there are unique coefficients $c_i \in \bar{k}$ and a function $f \in \bar{k}(C)$ such that

$$r - f + \left(\sum_i c_i t^{\mu\ -1}/Q \right) \in R(K - \alpha Q).$$

Equivalently, assuming $r_Q = 0$ as discussed above, the coefficients c_i and the function f must satisfy:

1. $\nu_P(r_P - f) \geq -\nu_P(K)$ for all $P \neq Q$;
2. $\nu_Q(\sum_i c_i t^{\mu\ } - f) \geq \alpha$.

The existence of f satisfying (1) is guaranteed by the Strong Approximation Theorem [11–Chap. 12]; moreover, since $\alpha > m$, (2) implies that f is regular at Q and the coefficients c_i can be obtained from the initial part of its expansion in powers of t.

References

1. M. Atiyah. *Complex fibre bundles and ruled surfaces*, Proc. London Math. Soc., vol. 5, pp. 407–434, 1955.
2. A. Bertram. *Moduli of rank-2 vector bundles, theta divisors, and the geometry of curves in projective space*, Journal of Diff. Geometry, vol. 35, pp. 429–469, 1992.
3. F. Bogomolov and T. Petrov. *Algebraic Curves and One-Dimensional Fields*, Providence, RI: Amer. Math. Soc., 2002, Courant Lecture Notes in Mathematics.
4. D. Coles. *Vector Bundles and Codes on the Hermitian Curve*, IEEE Trans. Inf. Theory, vol. 51, no. 6, pp. 2113–2120, 2005.
5. V.D. Goppa. *Codes on algebraic curves*, Soviet Math. Dokl., vol. 24, pp. 170–172, 1981.
6. A. Garcia and H. Stichtenoth. *A Tower of Artin-Schreier extensions of function fields attaining the Drinfeld-Vlăduţ bound*, Invent. Math., vol. 121, pp. 211–222, 1995.
7. R. Hartshorne. *Algebraic Geometry*, Springer-Verlag, New York, 1977.
8. T. Høholdt and R. Pellikaan. *On the decoding of algebraic-geometric codes*, IEEE Trans. Inf. Theory, vol. 41, no. 6, pp. 1589–1614, 1995.
9. T. Johnsen. *Rank two bundles on Algebraic Curves and decoding of Goppa Codes*, International Journal of Pure and Applied Mathematics, vol. 4, no. 1, pp. 33–45, 2003.
10. H. Lange and M.S. Narasimhan. *Maximal Subbundles of Rank Two Vector Bundles on Curves*, Math. Ann., vol. 266, pp. 55–72, 1983.
11. O. Pretzel. *Codes and Algebraic Curves*, Oxford University Press, 1998.
12. I. R. Shararevich. *Basic Algebraic Geometry 1-2*, 2nd edition, Springer-Verlag, New York, 1994. Translated by M. Reid.
13. K. W. Shum, I. Aleshnikov, P. V. Kumar, H. Stichtenoth, V. Deolalikar. *A Low-Complexity Algorithm for the Construction of Algebraic-Geometric Codes Better Than the Gilbert-Varshamov Bound*, IEEE Trans. Inf. Theory, vol. 47, no. 6, pp. 2225–2241, 2001.

14. J.-P. Serre. *Algebraic Groups and Class Fields*, Springer-Verlag, New York, 1988.
15. M.A. Tsfasman, S.G. Vlăduţ, and T. Zink. *Modular curves, Shimura curves and Goppa codes better than the Varshamov-Gilbert bound*, Mathematische Nachrichten, vol. 109, pp. 21–28, 1982.

A The Proof of Proposition 2

This comes directly from [4], with minor rewording.

Applying Serre's proof of the duality theorem for curves [14–Proposition II.3] to our situation, we note that $\Omega^1(K - \alpha Q) \cong H^0(\alpha Q)$ is put in duality with $R/(R(K - \alpha Q) + \bar{k}(C))$ by the pairing

$$\langle \omega, r \rangle \;=\; \sum_{P \in C} Res_P(r_P \cdot \omega). \tag{4}$$

Fix a basis $\{f_i\}$ for $\mathcal{L}(\alpha Q)$. We may assume $\nu_Q(f_i) = \mu_i$ (the i-th Weierstrass non-gap at Q). Let $a(i,j) \in \bar{k}$ denote the coefficient of t^j in the expansion of f_i in powers of t; that is,

$$f_i \;=\; \sum_{r=-\mu}^{\infty} a(i,r) \cdot t^r.$$

Now suppose for a moment that there are distinct indices i and j for which $a(i, -\mu_j) \neq 0$. Thus while f_j has a pole of order μ_j at Q, there is a non-zero coefficient of $t^{-\mu}$ in the expansion of some other function f_i with a higher pole order at Q. Since $\nu_Q(f_i - a(i, -\mu_j) \cdot f_j) = -\mu_i$, we may replace f_i with $f_i - a(i, -\mu_j) \cdot f_j$ to obtain another basis for $\mathcal{L}(\alpha Q)$; moreover, the coefficient of $t^{-\mu}$ in the expansion of this new function is zero. We may therefore assume that $a(i, -\mu_j) = 0$ if and only if $i \neq j$. Letting $\omega_i = f_i \cdot dt$ in an open neighborhood of Q, we can write $Res_Q(t^{n\ -1} \cdot \omega_i) \;=\; a(i, -n_j)$; that is,

$$Res_Q(t^{n\ -1} \cdot \omega_i) = 0 \;\longleftrightarrow\; i \neq j.$$

Combining this with (4), we have

$$\langle \omega_i,\, t^{n\ -1}/\, Q \rangle = 0 \;\longleftrightarrow\; i \neq j.$$

Referring again to (4), we see that every differential $\omega \in \Omega^1(K - \alpha Q)$ defines a linear functional $\langle \omega, \cdot \rangle$ on $R/(R(K - \alpha Q) + \bar{k}(C))$. Indeed, $\langle \omega, r \rangle = 0$ for all $r \in R(K - \alpha Q)$ since $r_P \cdot \omega$ has no poles for any point $P \in C$, and $\langle \omega, r \rangle = 0$ for all repartitions $r \in \bar{k}(C)$ by the Residue Theorem.

We have contructed a basis $\{t^{\mu\ -1}/Q\}$ in the standard way for the space $R/(R(K - \alpha Q) + \bar{k}(C))$ in terms of a basis $\{\omega_i\}$ for the dual space. That is, $\langle \omega_i,\, t^{n\ -1} \rangle = 0$ if and only if $i \neq j$. □

Computing Gröbner Bases for Vanishing Ideals of Finite Sets of Points

Jeffrey B. Farr[1] and Shuhong Gao[2]

[1] Simon Fraser University, Burnaby, B.C. V5A 1S6, Canada
jfarr@cecm.sfu.ca
http://www.cecm.sfu.ca/~jfarr/
[2] Clemson University, Clemson, SC 29634-0975, USA
sgao@ces.clemson.edu
http://www.math.clemson.edu/~sgao/

Abstract. We present an algorithm to compute a Gröbner basis for the vanishing ideal of a finite set of points in an affine space. For distinct points the algorithm is a generalization of univariate Newton interpolation. Computational evidence suggests that our method compares favorably with previous algorithms when the number of variables is small relative to the number of points. We also present a preprocessing technique that significantly enhances the performance of all the algorithms considered. For points with multiplicities, we adapt our algorithm to compute the vanishing ideal via Taylor expansions.

1 Introduction

Suppose $P_1, \dots, P_n$ are distinct points in the m-dimensional vector space over a field $\mathbb{F}$. The set of polynomials in $\mathbb{F}[x_1, \dots, x_m]$ that evaluate to zero at each P_i form a zero-dimensional ideal called the vanishing ideal of the points. The problem that we consider is how to compute the reduced Gröbner basis for the vanishing ideal of any finite set of points under any given monomial order. This problem arises in several applications; for example, see [16] for statistics, [13] for biology, and [18, 11, 12, 6] for coding theory.

A polynomial time algorithm for this problem was first given by Buchberger and Möller (1982) [2], and significantly improved by Marinari, Möller and Mora (1993) [14], and Abbott, Bigatti, Kreuzer and Robbiano (2000) [1]. These algorithms perform Gauss elimination on a generalized Vandermonde matrix and have a polynomial time complexity in the number of points and in the number of variables. O'Keeffe and Fitzpatrick (2002) [9] studied this problem from a coding theory point of view. They present an algorithm that is exponential in the number of variables, and the Gröbner basis which they compute is not reduced.

We present here a variation of the O'Keeffe-Fitzpatrick method. Our approach does, though, compute the *reduced* Gröbner basis and is essentially a generalization of Newton interpolation for univariate polynomials. Even though the time complexity of our algorithm is still exponential in the number of variables, its practical performance improves upon both the O'Keeffe-Fitzpatrick algorithm

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 118–127, 2006.

and the linear algebra approach if the number of variables is relatively small compared to the number of points.

The rest of the paper is organized as follows. In Section 2, we present our algorithm for distinct points. We also show how multivariate interpolation is a special case of computing vanishing ideals. Section 3 presents experimental time comparisons along with a sorting heuristic for the points. Finally, Section 4 shows how to handle the case for points with multiplicity. Some of the material presented here is surveyed in our recent paper [7], which gives a broader view on how Gröbner basis theory can be used in coding theory.

2 Distinct Points

Throughout this section we fix an arbitrary monomial order (also called term order by some authors) on the polynomial ring $\mathbb{F}[x_1, \ldots, x_m]$. Then each polynomial $f \in \mathbb{F}[x_1, \ldots, x_m]$ has a leading term, denoted by $\mathrm{LT}(f)$, and each ideal has a unique reduced Gröbner basis. For any subset $G \subset \mathbb{F}[x_1, \ldots, x_m]$, we define

$$\mathcal{B}(G) = \{\mathbf{x}^\alpha : \alpha \in \mathbb{N}^m \text{ and } \mathbf{x}^\alpha \text{ is not divisible by } \mathrm{LT}(g) \text{ for any } g \in G\},$$

where $\mathbb{N} = \{0, 1, 2, \ldots\}$ and $\mathbf{x}^\alpha = x_1^{\alpha_1} \cdots x_m^{\alpha}$ for $\alpha = (\alpha_1, \ldots, \alpha_m)$. A basic theorem in Gröbner basis theory tells us that, for each ideal $\mathbf{I} \subset \mathbb{F}[x_1, \ldots, x_m]$, the monomials in $\mathcal{B}(\mathbf{I})$ form a basis for the quotient ring $\mathbb{F}[x_1, \ldots, x_m]/\mathbf{I}$ as a vector space over $\mathbb{F}$ (see Section 3 in [4]). This basis is called a monomial basis, or a standard basis, for $\mathbf{I}$ under the given monomial order. For $V \subseteq \mathbb{F}^m$, let $\mathbf{I}(V)$ denote the vanishing ideal of V; that is,

$$\mathbf{I}(V) = \{f \in \mathbb{F}[x_1, \ldots, x_m] : f(P) = 0, \text{ for all } P \in V\}.$$

If $V = \{P_1, \ldots, P_n\}$, $\mathbf{I}(V)$ is also written as $\mathbf{I}(P_1, \ldots, P_n)$.

Lemma 1. *For $g_1, \ldots, g_s \in \mathbf{I} = \mathbf{I}(P_1, \ldots, P_n)$, $\{g_1, \ldots, g_s\}$ is a Gröbner basis for $\mathbf{I}$ if and only if $|\mathcal{B}(g_1, \ldots, g_s)| = n$.*

Proof. By definition $g_1, \ldots, g_s \in \mathbf{I}$ form a Gröbner basis for $\mathbf{I}$ if and only if $\mathcal{B}(g_1, \ldots, g_s) = \mathcal{B}(\mathbf{I})$. One can show by interpolation that $\dim \mathbb{F}[x_1, \ldots, x_m]/\mathbf{I} = n$. But the monomials in $\mathcal{B}(\mathbf{I})$ form a basis for the quotient ring $\mathbb{F}[x_1, \ldots, x_m]/\mathbf{I}$ viewed as a vector space over $\mathbb{F}$. The lemma follows immediately. □

Lemma 2. *Suppose $G = \{g_1, \ldots, g_s\}$ is a Gröbner basis for $\mathbf{I}(V)$, for a finite set $V \subset \mathbb{F}^m$. For a point $P = (a_1, \ldots, a_m) \notin V$, let g_i denote the polynomial in G with smallest leading term such that $g_i(P) \neq 0$, and define*

$$\begin{aligned} \tilde{g}_j &:= g_j - \frac{g_j(P)}{g_i(P)} \cdot g_i, \qquad && j \neq i, \text{ and} \\ g_{ik} &:= (x_k - a_k) \cdot g_i, \qquad && 1 \leq k \leq m. \end{aligned}$$

Then

$$\widetilde{G} = \{\tilde{g}_1, \ldots, \tilde{g}_{i-1}, \tilde{g}_{i+1}, \ldots, \tilde{g}_s, g_{i1}, \ldots, g_{im}\}$$

is a Gröbner basis for $\mathbf{I}(V \cup \{P\})$.

Proof. At least one polynomial in G must be nonzero when evaluated at P since $P \notin V$; hence, a suitable g_i exists.

Certainly, $\widetilde{G} \subseteq \mathbf{I}(V \cup \{P\})$ as the new and modified polynomials evaluate to zero at all points in $V \cup \{P\}$. Denote $\mathrm{LT}(g_i)$ by $\mathbf{x}^\alpha$. We claim that

$$\mathcal{B}(\widetilde{G}) = \mathcal{B}(G) \cup \{\mathbf{x}^\alpha\}. \tag{1}$$

By the choice of i, $\mathrm{LT}(\tilde{g}_j) = \mathrm{LT}(g_j)$, for all $j \neq i$. Also, since g_i was replaced in $\widetilde{G}$ by $g_{i1}, g_{i2}, \ldots, g_{im}$, whose leading terms are $\mathbf{x}^\alpha x_1$, $\mathbf{x}^\alpha x_2$, $\ldots$, $\mathbf{x}^\alpha x_m$, we know that $\mathbf{x}^\alpha$ is the only monomial not in $\mathcal{B}(\mathbf{I}(V))$ that is in $\mathcal{B}(\mathbf{I}(V \cup \{P\}))$. Thus, (1) is satisfied, and $|\mathcal{B}(\widetilde{G})| = |\mathcal{B}(G)| + 1$. Since G is a Gröbner basis for $\mathbf{I}(V)$, we have $|\mathcal{B}(G)| = |V|$, and the conclusion follows from Lemma 1. □

Notice that some of the $\mathrm{LT}(g_{ik})$ may be divisible by the leading term of another polynomial in $\widetilde{G}$. In such a case, g_{ik} may be omitted from $\widetilde{G}$ and $\widetilde{G} \setminus \{g_{ik}\}$ is still a Gröbner basis. In fact, we can check for this property before computing g_{ik} so that we save ourselves needless computation. In so doing, we also guarantee that the resulting $\widetilde{G}$ is a minimal Gröbner basis for $\mathbf{I}(V \cup \{P\})$.

To get a reduced Gröbner basis, we still need to reduce the new polynomials g_{ik}. We order the variables in increasing order, say $x_1 < x_2 < \ldots < x_m$, and reduce the polynomials from g_{i1} up to g_{im} . Thus, in Algorithm 1 the polynomials in G are always stored so that the leading terms of its polynomials are in increasing order. This will make sure that each g_{ik} need only be reduced once. Also, $\mathrm{Reduce}(h, G)$ is the unique remainder of h when reduced by polynomials in G.

Algorithm 1

```
Input: $P_1, P_2, \ldots, P_n \in \mathbb{F}^m$, and a monomial order
  We assume that the variables are labelled so that $x_1 < \ldots < x_m$.
Output: $G$, the reduced Gröbner basis for $\mathbf{I}(P_1, \ldots, P_n)$, in increasing order.

$G := \{1\}$;          /* the ith polynomial in $G$ is denoted $g_i$ */
FOR $k$ from 1 to $n$ DO
   Find the smallest $i$ so that $g_i(P_k) \neq 0$;
   FOR $j$ from $i+1$ to $|G|$ DO   $g_j := g_j - \frac{g\ (P\ )}{g\ (P\ )} \cdot g_i$;   END FOR;
   $G := G \setminus \{g_i\}$;
   FOR $j$ from 1 to $m$ DO
      IF $x_j \cdot \mathrm{LT}(g_i)$ not divisible by any leading term of $G$ THEN
         Compute $h := \mathrm{Reduce}((x_j - a_j) \cdot g_i, G)$;
         Insert $h$ (in order) into $G$;
      END IF;
   END FOR;
END FOR;

RETURN $G$.
```

Lemma 2 and the subsequent remarks imply the following theorem.

Theorem 1. *For a finite set $V \subseteq \mathbb{F}^m$ and a given monomial order, Algorithm 1 returns the reduced Gröbner basis for $\mathbf{I}(V)$.*

A related question is multivariate interpolation, and it can easily be solved using Algorithm 1. The interpolation problem is: given the points $P_1, \ldots, P_n \in \mathbb{F}^m$ and any values $r_1, \ldots, r_n \in \mathbb{F}$, find a "smallest" f so that

$$f(P_i) = r_i, \quad 1 \leq i \leq n. \tag{2}$$

Multivariate polynomial interpolation has been extensively studied in the past 30 years (see the survey [10]). The property of being "smallest" is addressed by introducing an appropriate monomial order on $\mathbb{F}[x_1, \ldots, x_m]$. Then there is a unique polynomial $f \in Span_{\mathbb{F}}(\mathcal{B})$ satisfying (2), and it will be the smallest such polynomial under the given monomial order. One strategy for finding this polynomial f is given in [17] that uses separator polynomials. The following theorem, which follows directly from Lemma 1, tells us that any algorithm for computing vanishing ideal can be easily used to solve the interpolation problem.

Theorem 2. *Let G be the reduced Gröbner basis for $\mathbf{I} = \mathbf{I}(P_1, \ldots, P_n)$ under the fixed monomial order $<$ on $\mathbb{F}[x_1, \ldots, x_m]$, and let $\mathcal{B} = \mathcal{B}(\mathbf{I})$ be the corresponding monomial basis. Introduce a new variable z and an elimination order for z that extends $<$. Then the reduced Gröbner basis for $\mathbf{I}((P_1, r_1), \ldots, (P_n, r_n))$, is of the form $G \cup \{z - f\}$, where f is the unique polynomial in $Span_{\mathbb{F}}(\mathcal{B})$ satisfying (2).*

One can easily generalize Theorem 2 to the case when there are more than one z-coordinate. Also, in the case when $m = 1$ Theorem 2 appears in the literature as the "Shape Lemma" (see Exercise 16 in Section 2.4 in [5]).

This same strategy can be modified for multivariate rational function interpolation. In this case the Gröbner basis computation is performed for a submodule of rank two rather than for an ideal. The major hurdle that has to be overcome before applying a modified Algorithm 1 is the selection of an appropriate term order. We refer the reader to [6] for more details.

3 Time Complexity

3.1 The Cost of Reduction

All the steps in Algorithm 1 are straightforward to analyze except the reduction step in line 11. We use standard Buchberger reduction (i.e., repeated division). This reduction has a worst-case time complexity that may be exponential in the number m of variables. It is possible to make this step polynomial time by using the border-basis reduction technique introduced in [8]. The border Gröbner basis computed, however, is quite large in general. For example, the reduced Gröbner basis for the vanishing ideal of a random set of 500 points from $\mathbb{F}_2^{10}$ under *lex* order usually contains around 100 polynomials, while the border basis typically contains over 2000. So the running time and memory usage of Algorithm 1 using border-basis reduction are much worse than the original. For these reasons we ignore the theoretical "improvements" that border-basis reduction provides.

3.2 Running Time Comparison

As we mentioned earlier, the methods in [1, 2, 14] are based on Gauss elimination and have a polynomial time complexity $O(n^3m^2)$. We compare our Algorithm 1 particularly with the algorithm (MMM) of Marinari, Möller and Mora [14]. Although the algorithm of [1] has an excellent implementation in the computer algebra system CoCoA, it is not appropriate for us to compare this compiled code with our interpreted code (see specs below).

The Gröbner basis found via the algorithm (O'K-F) of O'Keeffe and Fitzpatrick [9] is minimal in the sense that the number of polynomials in the basis is the smallest, but the length of the polynomials computed may grow exponentially in the number m of variables. So, most of the computing time in O'K-F is taken up with dealing with large polynomials, and most of the time in Algorithm 1 involves the reduction step, *i.e.*, computing Reduce(g_{ij}, G).

Table 1 presents running times for the algorithms for various point sets. We have chosen three high-dimensional vector spaces and three low-dimensional vector spaces to highlight the significance of the dimension. The times are the average running times in seconds for randomly chosen point sets from the specified vector space (based on 100 experiments for $n = 250$, 10 experiments for

Table 1. Average running times for 250, 500 and 1000 random points from $\mathbb{F}_q^m$

q	m	MMM		Algorithm 1		O'K-F	
		grlex	*lex*	*grlex*	*lex*	*grlex*	*lex*
11	3	2.440	1.404	1.182	0.356	1.006	0.705
31	3	2.762	1.481	1.418	0.312	1.213	0.395
101	3	2.867	1.423	1.512	0.220	1.289	0.218
2	10	2.542	1.321	2.201	1.142	3.015	9.832
2	12	4.954	1.894	4.207	2.381	4.037	14.43
2	15	7.568	2.875	7.592	7.565	8.066	22.54

q	m	MMM		Algorithm 1		O'K-F	
		grlex	*lex*	*grlex*	*lex*	*grlex*	*lex*
11	3	14.72	10.50	5.726	1.789	5.126	9.547
31	3	19.33	10.53	9.357	1.688	8.236	3.043
101	3	20.08	10.66	11.16	1.141	9.467	1.371
2	10	12.10	7.773	8.849	4.314	18.81	332
2	12	24.43	11.62	21.68	13.49	31.90	367
2	15	64.69	16.79	63.74	33.42	53.38	522

q	m	MMM		Algorithm 1		O'K-F	
		grlex	*lex*	*grlex*	*lex*	*grlex*	*lex*
31	3	143	80.98	71.04	10.80	68.05	39.59
101	3	149	86.05	107	6.852	92.92	11.52
2	12	173	75.62	146	74.36	307	10321
2	15	315	117	312	218	471	16609

$n = 500, 1000$). The algorithms were implemented in Magma version 2.11 and run on an Apple Power Macintosh G5 computer, 2.5 GHz CPU, 2 GB RAM.

The timings indicate that Algorithm 1 is faster than MMM—by a factor of two for *grlex*, a factor approaching ten for *lex*—if the dimension m (the number of variables) is small relative to the number n of points. In comparison to O'K-F, Algorithm 1 takes roughly the same time for small m and n, but O'K-F slows down somewhat when n increases and slows down quickly when m increases.

3.3 Sorting the Points

A clever ordering of the points can improve the running time of Algorithm 1, O'K-F and, somewhat surprisingly, MMM. The significance of improvement depends on both the chosen monomial order and the geometric structure of the points.

The details of this ordering are quite simple. If $x_1 < \ldots < x_m$, then group the points first according to the x_1-coordinate; these groups are ordered by the number of elements, largest to smallest (specifically, nonincreasingly). Within each

Table 2. Running times for 250, 500 and 1000 random points (sorted) from $\mathbb{F}_q^m$

q	m	MMM		Algorithm 1		O'K-F	
		grlex	*lex*	*grlex*	*lex*	*grlex*	*lex*
11	3	2.004	0.714	1.066	0.277	0.890	0.286
31	3	2.735	0.956	1.418	0.273	1.189	0.264
101	3	2.867	1.076	1.516	0.206	1.277	0.193
2	10	1.557	0.746	1.363	0.575	1.534	0.646
2	12	3.342	1.131	3.257	1.091	2.544	1.158
2	15	5.061	1.802	5.360	3.689	5.283	2.118

q	m	MMM		Algorithm 1		O'K-F	
		grlex	*lex*	*grlex*	*lex*	*grlex*	*lex*
11	3	10.31	3.558	4.978	1.277	4.426	1.421
31	3	19.017	5.693	9.298	1.418	8.184	1.406
101	3	20.11	6.996	11.158	1.049	9.394	0.962
2	10	5.461	2.498	4.534	1.681	6.308	2.426
2	12	13.70	5.250	13.34	6.022	16.19	7.850
2	15	46.06	8.058	53.42	11.16	34.39	12.39

q	m	MMM		Algorithm 1		O'K-F	
		grlex	*lex*	*grlex*	*lex*	*grlex*	*lex*
31	3	139	34.41	70.66	8.754	70.35	9.041
101	3	149	47.38	107	6.037	91.97	5.471
2	12	90.44	22.45	88.74	23.63	121	45.02
2	15	169	40.91	182	71.13	217	114

of the groups, repeat the process, but according to the x_2-coordinate. Continue for $x_3, \ldots, x_m$.

A comparison of Table 2 with Table 1 indicates the sizable impact of reordering. Essentially, this sorting decreases the amount of reduction that Algorithm 1 needs to do. Further, although O'K-F does not involve reduction, it is also helped since the Gröbner basis remains comparatively small. In MMM, reordering the points corresponds to a favorable reordering of the columns in an implicit matrix to which Gauss elimination is applied.

Gröbner bases under *lex* order experience the greatest benefit since they typically require the most reduction and are prone to exponential growth without reduction. Gröbner bases under *grlex* order with points from a low-dimensional vector space experience little or no speedup.

4 Points with Multiplicities

We now consider the case in which some points in the vanishing set have multiplicity. A general notion of *algebraic multiplicity* is described in [14] and [15]. We will adopt a special form used by Cerlienco and Mureddu [3] that is general enough for most applications.

Let $\mathbf{v} = (v_1, \ldots, v_m) \in \mathbb{Z}^m$. We define a differential operator $D^{\mathbf{v}}$ by

$$D^{\mathbf{v}} = \frac{1}{v_1! \cdots v_m!} \cdot \frac{\partial^{v_1 + \ldots + v}}{\partial {x_1}^{v_1} \cdots \partial {x_m}^{v}}.$$

We note that $D^{\mathbf{v}}$ is a linear map on functions with the m variables $x_1, \ldots, x_m$. Let $P \in \mathbb{F}^m$ and f be any function on $x_1, \ldots, x_m$. We employ the notation

$$[D^{\mathbf{v}} f](P) = D^{\mathbf{v}} f|_{\mathbf{x}=P}, \tag{3}$$

where $P = (a_1, \ldots, a_m) \in \mathbb{F}^m$. Then, under reasonable conditions (analytic or algebraic) on f, we have

$$f(\mathbf{x} + P) = \sum_{\mathbf{v} \in \mathbb{N}} [D^{\mathbf{v}} f](P) \cdot \mathbf{x}^{\mathbf{v}}. \tag{4}$$

We call the right-hand side of (4) the Taylor expansion of f at P, denoted by $T(f, P)$. Note that (4) is equivalent to

$$f(\mathbf{x}) = \sum_{\mathbf{v} \in \mathbb{N}} [D^{\mathbf{v}} f](P) \cdot (\mathbf{x} - P)^{v} = \sum_{\mathbf{v} \in \mathbb{N}} [D^{\mathbf{v}} f](P) \cdot (x_1 - a_1)^{v_1} \cdots (x_m - a_m)^{v} \ ,$$

which is the more typically referred to form of Taylor expansion.

A subset $\Delta \subseteq \mathbb{N}^m$ is called a *delta set* (or a *Ferrers diagram*, or an *order ideal*), if it closed under the division order; that is, if $\mathbf{u} \in \Delta$ then $\mathbf{v} \in \Delta$ for all $\mathbf{v} = (v_1, \ldots, v_m) < \mathbf{u} = (u_1, \ldots, u_m)$ componentwise. Define

$$T(f, P, \Delta) = \sum_{\mathbf{v} \in \Delta} [D^{\mathbf{v}} f](P) \cdot \mathbf{x}^{\mathbf{v}}. \tag{5}$$

$T(f,P)$ denotes the full (possibly infinite if f is not a polynomial) Taylor expansion of f, while $T(f,P,\varDelta)$ is truncated to consider only those coefficients corresponding to monomials with exponents in $\varDelta$. For any nonzero polynomial $f \in \mathbb{F}[x_1,\ldots,x_m] = \mathbb{F}[\mathbf{x}]$, a point $P \in \mathbb{F}^m$ and a delta set $\varDelta \subset \mathbb{N}^m$, f is said to vanish at P with multiplicity $\varDelta$ if $T(f,P,\varDelta) = 0$.

On the other hand, f is said to have *arithmetic multiplicity* m_0 at P if $[D^{\mathbf{v}} f](P) = 0$ for all v with $|v| = v_1 + v_2 + \ldots + v_m < m_0$. In terms of the algebraic definition, this implies that the multiplicity set $\varDelta$ is restricted to a triangular shape. The algebraic definition clearly subsumes the arithmetic one.

With the algebraic definition of multiplicity in mind, we generalize Algorithm 1 to compute the vanishing ideal of a set of points $\{P_1,\ldots,P_n\}$, each point having multiplicity defined by the sets $\varDelta_1,\ldots,\varDelta_n$. Denote this ideal by

$$\mathbf{I}\left((P_1,\varDelta_1),\ldots,(P_n,\varDelta_n)\right) = \{f \in \mathbb{F}[x_1,\ldots,x_m] : T(f,P_i,\varDelta_i) = 0, \quad 1 \le i \le n\}.$$

Since the $\varDelta_i$'s are delta sets, one can show that this set is indeed an ideal in $\mathbb{F}[x_1,\ldots,x_m]$. (This is not true if the $\varDelta_i$'s are not all delta sets.)

Algorithm 2 varies from Algorithm 1 in the following way. Instead of evaluating each $f \in G$ at P_i, we need to compute the truncated Taylor expansion $T(f,P_i,\varDelta_i)$; we denote the set of these expansions by $\mathcal{T}$. The points in each $\varDelta_i$ must be ordered in nondecreasing order to ensure that these Taylor expansions may be computed efficiently and to ensure that G at each iteration (*i.e.*, at the start of line 8) is a Gröbner basis for the vanishing ideal of the points $P_1,\ldots,P_k$ with multiplicities $\varDelta_1,\ldots,\varDelta_{k-1}$ and the subset of points in $\varDelta_k$ up to $\mathbf{v}$.

Like Algorithm 1, Algorithm 2 is an iterative method; in fact, not only does the algorithm build the Gröbner basis for the vanishing ideal "one point at a time" but it also builds it "one multiplicity at a time." That is, when a new point is introduced, the algorithm updates the Gröbner basis by stepping through the corresponding multiplicity set element by element. Of course if each multiplicity set is trivial ($|\varDelta_i| = 1$), then Algorithm 2 is equivalent to Algorithm 1.

The following analogue to Lemma 1 is necessary to establish the correctness of Algorithm 2. We omit the proof, noting only that the key step that established Algorithm 1 is the same for this algorithm. Namely, at each step in the algorithm, we add exactly one element from a multiplicity set and exactly one element to the monomial basis. This ensures that our basis G is always Gröbner.

Lemma 3. *Fix a monomial order on* $\mathbb{F}[\mathbf{x}]$*, and let* $V = \{(P_1,\varDelta_1),\ldots,(P_n,\varDelta_n)\}$*, where* $P_i \in \mathbb{F}^m$ *are distinct and* $\varDelta_i \subset \mathbb{N}^m$ *are delta sets. Then* $\{g_1,\ldots,g_s\} \subset \mathbf{I}(V)$ *is a Gröbner basis for* $\mathbf{I}$ *if and only if* $|\mathcal{B}(g_1,\ldots,g_s)| = \sum_{j=1}^{n} |\varDelta_j|$.

Proof. We know (Lemma 3.8 in [14]) that $g_1,\ldots,g_s \in \mathbf{I}(V)$ form a Gröbner basis if and only if $|\mathcal{B}(g_1,\ldots,g_s)| = \dim \mathbb{F}[x_1,\ldots,x_m]/\mathbf{I}(V)$. We just need to show that the latter has dimension equal to $\sum_{j=1}^{n} |\varDelta_j|$. To see this, let $I_j = \mathbf{I}(P_j,\varDelta_j)$, the vanishing ideal of P_j with multiplicity $\varDelta_j$. Then $\mathbf{I}(V) = I_1 \cap \cdots \cap I_n$ and

$$\mathbb{F}[x_1,\ldots,x_m]/\mathbf{I}(V) \cong \bigoplus_{j=1}^{n} \mathbb{F}[x_1,\ldots,x_m]/I_j,$$

Table 3. Algorithm for computing the reduced Gröbner basis for the vanishing ideal of a set of points with multiplicities

```
Algorithm 2

Input: $P_1, \ldots, P_n \in \mathbb{F}^m$; $\Delta_1, \ldots, \Delta_n \subset \mathbb{N}^m$; and a monomial order.
Output: $G$, the reduced Gröbner basis for $\mathbf{I}((P_1, \Delta_1), \ldots, (P_n, \Delta_n))$,
           in increasing order.

$G := \{1\}$;      /* $g_i$ is the ith polynomial in $G$, in increasing order */
Order the variables so that $x_1 < x_2 < \ldots < x_m$;
Order the elements in each $\Delta_k$ in nondecreasing order under the
           division order;
FOR $k$ from 1 to $n$ DO
   Compute $\mathcal{T} = \{T_j = T(g_j, P_k, \Delta_k) : g_j \in G\}$, the set of
         (truncated) Taylor expansions;
   FOR $\mathbf{v}$ in $\Delta_k$ DO
      Find the smallest $i$ so that coeff$(T_i, \mathbf{x}^{\mathbf{v}}) \neq 0$;
      FOR $j$ from $i + 1$ to $|G|$ DO
         $\delta :=$ coeff$(T_j, \mathbf{x}^{\mathbf{v}})$ / coeff$(T_i, \mathbf{x}^{\mathbf{v}})$;
         $g_j = g_j - \delta \cdot g_i$;
         $T_j = T_j - \delta \cdot T_i$;
      END FOR;
      $G := G \setminus \{g_i\}$ and $\mathcal{T} := \mathcal{T} \setminus \{T_i\}$;
      FOR $j$ from 1 to $m$ DO
         IF $x_j \cdot$ LT$(g_i)$ not divisible by any LT of $G$ THEN
            Compute $h :=$ Reduce$((x_j - a_j) \cdot g_i, G)$;
            $T_h := x_j \cdot T_i$    (truncated);
            Insert (in order) $h$ into $G$ and $T_h$ into $\mathcal{T}$;
         END IF;
      END FOR;
   END FOR;
END FOR;

RETURN $G$.
```

as rings over $\mathbb{F}$. Note that $\{\mathbf{x}^\alpha : \alpha \in \Delta_j\}$ forms a basis for $\mathbb{F}[x_1, \ldots, x_m]/I_j$ as a vector space over $\mathbb{F}$, so its dimension is $|\Delta_j|$. The lemma follows immediately. □

5 Final Remarks

Algorithm 1 is included in MAPLE10 under the command `VanishingIdeal` in the `PolynomialIdeals` package. MAPLE code for the application of this algorithm to multivariate polynomial and rational function interpolation may be downloaded from [19]. A GAP implementation of Algorithm 1 by Joyner [20] is also available.

References

1. Abbott, J., Bigatti, A., Kreuzer, M., Robbiano, L.: Computing ideals of points. J. Symbolic Comput. **30** (2000), 341-356
2. Buchberger, B., Möller, H. M.: The construction of multivariate polynomials with preassigned zeros. Computer algebra, EUROCAM '82, pp. 24-31, Lecture Notes in Comput. Sci., vol. 144, Springer, Berlin-New York, 1982
3. Cerlienco, L. and Mureddu, M.: From algebraic sets to monomial linear bases by means of combinatorial algorithms. Formal power series and algebraic combinatorics (Montreal, PQ, 1992). Discrete Math. **139** (1995), no. 1-3, 73-87
4. Cox, D., Little, J., O'Shea, D.: Ideals, varieties, and algorithms, 2nd ed. Undergraduate Texts in Mathematics, Springer-Verlag, New York, 1997
5. Cox, D., Little, J., O'Shea, D.: Using algebraic geometry. Graduate Texts in Mathematics, 185, Springer-Verlag, New York, 1998
6. Farr, Jeffrey B., Gao, Shuhong: Gröbner bases and generalized Padé approximation. Math. Comp. (to appear)
7. Farr, Jeffrey B., Gao, Shuhong: Gröbner bases, Padé approximation, and decoding of linear codes. Coding Theory and Quantum Computing (Eds. D. Evans et al.), 3–18, Contemp. Math., 381, Amer. Math. Soc., Providence, RI, 2005
8. Faugere, J., Gianni, P., Lazard, D., Mora, T.: Efficient computation of zero-dimensional Gröbner bases by change of ordering. J. Symbolic Comput. **16** (1993), 329-344
9. Fitzpatrick, P., O'Keeffe, H.: Gröbner basis solutions of constrained interpolation problems. Fourth special issue on linear systems and control. Linear Algebra Appl. **351/352** (2002), 533-551
10. Gasca, M., Sauer, T.: Polynomial interpolation in several variables, in Multivariate polynomial interpolation. Adv. Comput. Math. **12** (2000), no. 4, 377-410
11. Guruswami, V., Sudan, M.: Improved decoding of Reed-Solomon and algebraic-geometric codes. IEEE Transactions on Information Theory **46** (1999), no. 6, 1757–1767
12. Koetter, R., Vardy, A.: Algebraic Soft-Decision Decoding of Reed-Solomon Codes. IEEE Transactions on Information Theory **49**(2003), 2809–2825
13. Laubenbacher, R., Stigler, B.: A computational algebra approach to the reverse engineering of gene regulatory networks. Journal of Theoretical Biology **229** (2004), 523-537
14. Marinari, M. G., Möller, H. M., Mora, T.: Gröbner bases of ideals defined by functionals with an application to ideals of projective points. Appl. Algebra Engrg. Comm. Comput. **4** (1993), no. 2, 103-145
15. Marinari, M. G., Möller, H. M., Mora, T.: On multiplicities in polynomial system solving. Trans. Amer. Math. Soc. **348** (1996), no. 8, 3283-3321
16. Pistone, G., Riccomagno, E., Wynn, H. P.: Algebraic Statistics: Computational Commutative Algebra in Statistics. Monographs on Statistics & Applied Probability 89, Chapman & Hall/CRC, 2001
17. Robbiano, L.: Gröbner bases and statistics. *Gröbner bases and applications* (Linz, 1998), 179-204, London Math. Soc. Lecture Note Ser., vol. 251, Cambridge Univ. Press, Cambridge, 1998
18. Sudan, M.: Decoding of Reed Solomon codes beyond the error-correction bound. J. Complexity **13** (1997), no. 1, 180–193
19. Shuhong Gao's webpage. http://www.math.clemson.edu/˜sgao/
20. David Joyner's webpage. http://cadigweb.ew.usna.edu/˜wdj/gap/curves/

A Class of Fermat Curves for which Weil-Serre's Bound Can Be Improved

Francis N. Castro[1], Ernesto Gomez[2], and Oscar Moreno[3]

[1] Department of Mathematics, University of Puerto Rico, Rio Piedras
fcastro@goliath.cnnet.clu.edu
[2] Department of Computer Science,
California State University, San Bernardino
egomez@csci.causb.edu
[3] Department of Computer Sciences,
University of Puerto Rico, Rio Piedras
moreno@uprr.pr

Abstract. In this paper we introduce the class of semiprimitive Fermat curves, for which Weil-Serre's bound can be improved using Moreno-Moreno p-adic techniques. The basis of the improvement is a technique for giving the exact divisibility for Fermat curves, by reducing the problem to a simple finite computation.

1 Summary of p-Adic Bounds for Curves

In this paper we are going to present new curves satisfying Theorem 1 below and using it we obtain our improved Weil-Serre's bound.

In the present section we recall how O. Moreno and C. Moreno combine Serre's techniques with the Moreno-Moreno improved Ax-Katz estimate (see [3])to produce a p-adic version of Serre's estimate. For Fermat curves considered here, we can formulate the best possible Moreno-Moreno type p-adic Serre Bound.

Let

$$aX^d + bY^d = cZ^d, \ (abc \neq 0) \tag{1}$$

be a Fermat curve over $\mathbb{F}_{p^f}$ and let $|N|$ be the number of affine points of $aX^d + bY^d = cZ^d$ over $\mathbb{F}_{p^f}$. Note that the Fermat curves are nonsingular curves. Hence we can apply to them the Weil's Theorem.

Now we apply the p-adic estimate of [1] to the curve (1). Note that the genus of a Fermat equation is less than or equal to $(d-1)(d-2)/2$, where d is the degree of the Fermat equation.

Theorem 1. *Let $aX^d + bY^d = cZ^d$ be an equation over $\mathbb{F}_{p^f}$ and let μ be a positive integer satisfying $|N(\mathbb{F}_{p^{fm}})| \equiv 0 \bmod p^{\mu m}\ \forall\ m > 0$. Then the number of solutions $|\tilde{N}|$ of $aX^d + bY^d = cZ^d$ in $\mathbb{P}^2(\mathbb{F}_{p^{mf}})$ satisfies the following bound:*

$$|\,|\tilde{N}| - (p^{mf}+1)| \leq \frac{1}{2}(d-1)(d-2)p^{\mu m}[2p^{mf/2}p^{-\mu m}].$$

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 128–135, 2006.

Remark 1. Note that in order to obtain in the above theorem a non-trivial improvement, m and f must both be odd. That is the reason why throughout the paper, and in particular in Tables 1, 2 and 3, f and m are always odd.

Also note that in order to apply Theorem 1 we need curves where the divisibility grows upon extensions or $|N(\mathbb{F}_{p^{fm}})| \equiv 0 \bmod p^{\mu m} \; \forall \; m > 0$.

Remark 2. In general, it is difficult to find curves satisfying the property of divisibility of Theorem 1. This is to find curves $\mathcal{C}$ over $\mathbb{F}_q$ and $\mu > 0$ such that $p^{m\,\mu}$ divides the number of rational points of $\mathcal{C}$ over $\mathbb{F}_{q^m}$ for $m = 1, 2 \ldots$ (Artin-Schreier's curves satisfy this property.).

In the following section we are going to present new families of curves satisfying Remark 2. Hence we obtain an improved p-adic bound for their number of rational points.

2 Divisibility of Fermat Curves

In this section we are going to reduce the estimation of the divisibility of Fermat curves to a computational problem. Let $|N|$ be the number of solutions of the Fermat curve $aX^d + bY^d = cZ^d$ over the finite field $\mathbb{F}_{p^f}$. Note that that if $(p^f - 1, d) = k$, then the number of solutions of $aX^d + bY^d = cZ^d$ is equal to the number of solutions of $aX^k + bY^k = cZ^k$ over $\mathbb{F}_{p^f}$. Hence, we assume that d divides $p^f - 1$.

Let n be a positive integer $n = a_0 + a_1 p + a_2 p^2 + \cdots + a_l p^l$ where $0 \leq a_i < p$ we define the p-weight of n by $\sigma_p(n) = \sum_{i=0}^{l} a_i$.

Following the techniques of [3, Theorem 22], we associate to equation (1) the following system of modular equations:

$$\begin{aligned} dj_1 &\equiv 0 \bmod p^f - 1 \\ dj_2 &\equiv 0 \bmod p^f - 1 \\ dj_3 &\equiv 0 \bmod p^f - 1 \\ j_1 + j_2 + j_3 &\equiv 0 \bmod p^f - 1, \end{aligned} \tag{2}$$

where $1 \leq j_1, j_2, j_3 \leq q - 1$.

This modular system of equations determines the p-divisibility of $|N|$, i.e., if

$$\mu = \min_{\substack{(j_1, j_2, j_3) \\ is\ solution\ of\ (2)}} \{\frac{\sigma_p(j_1) + \sigma_p(j_2) + \sigma_p(j_3)}{p - 1}\} - f, \tag{3}$$

then p^μ divides $|N|$. This implies that any solution of the modular equation $dj_i \equiv 0 \bmod p^f - 1$ is of the form $c_i \cdot \frac{p^f - 1}{d}$ where $1 \leq c_i \leq d$. We are going to use the following results of [3]: for any positive integer k

$$\sigma_p((p^f - 1)k) \geq \sigma_p(p^f - 1) = (p - 1)f. \tag{4}$$

Now we state one of the main theorem of [3, Theorem 25],

Theorem 2. *Consider the family of polynomial equations:*

$$\mathcal{G} = \{aX^d + bY^d = cZ^d \,|\, a, b, c \in \mathbb{F}_{p^f}^{\times} \}.$$

Then there exists a polynomial $G \in \mathcal{G}$ such that the number of solutions of G is divisible by p^{μ} but not divisible by $p^{\mu+1}$, where μ is defined in (3).

Now we consider 3-tuples $(c_1, c_2, c_3) \in \mathbf{N}^3$ satisfying:

$$\frac{c_1}{d} + \frac{c_2}{d} + \frac{c_3}{d} \tag{5}$$

is a positive integer, where $1 \le c_i \le d$. The following Lemma gives a simpler way to compute μ of (3).

Lemma 1. *Let $q = p^f$ and d be a divisor of $q-1$. Let $aX^d + bY^d = cZ^d$ be a polynomial over $\mathbb{F}_q$. Then μ defined in (3) satisfies*

$$\mu = \min_{\substack{(c_1,c_2,c_3) \\ satisfies\ (5)}} \frac{\sum_{i=1}^{3} \sigma_p(c_i(q-1)/d)}{p-1} - f. \tag{6}$$

Proof. We know that the solutions of (2) are of the form $(c_1(p^f-1)/d, c_2(p^f-1)/d, c_3(p^f-1)/d)$. We obtain from the last congruence of (2) the following:

$$\frac{c_1(p^f-1)}{d} + \frac{c_2(p^f-1)}{d} + \frac{c_3(p^f-1)}{d} = (\frac{c_1}{d} + \frac{c_2}{d} + \frac{c_3}{d})(p^f-1) = k(p^f-1).$$

Therefore $\frac{c_1}{d} + \frac{c_2}{d} + \frac{c_3}{d}$ is positive integer.

The following Lemma is the one that allows us to apply Theorem 1.

Lemma 2. *Let q be power of a prime and d divides $q-1$. Then $\sigma_p(c(q^m-1)/d) = m\,\sigma_p(c(q-1)/d)$, where $1 \le c \le d-1$.*

Proof. Note that $c(q^m-1) = c(q-1)(q^{m-1} + \cdots + q + 1)$. Hence

$$\begin{aligned} \sigma_p(c(q^m-1)/d) &= \sigma_p(c\tfrac{q-1}{d}(q^{m-1} + \cdots + q + 1)) \\ &= m\,\sigma_p(\tfrac{c(q-1)}{d}) \end{aligned}$$

Combining the above two lemmas, we obtain the following proposition.

Proposition 1. *Let $q = p^f$ and d be a divisor of $q-1$. Let $aX^d + bY^d = cZ^d$ be a polynomial over $\mathbb{F}_{q^m}$. Then μ defined in (3) satisfies*

$$\mu = (\min_{\substack{(c_1,c_2,c_3) \\ satisfies\ (5)}} \frac{\sum_{i=1}^{3} \sigma_p(c_i(q^m-1)/d)}{p-1} - f) = m(\min_{\substack{(c_1,c_2,c_3) \\ satisfies\ (5)}} \frac{\sum_{i=1}^{3} \sigma_p(c_i(q-1)/d)}{p-1} - f).$$

Remark 3. Note that using Proposition 1, we only need to do one computation to estimate the divisibility of (1), the smallest $q-1$ such that d divides $q-1$. Consequently we have reduced the problem of finding the divisibility of Fermat Curves to a finite computation. Proposition 1 gives the exact divisibility in the sense that there are coefficients a', b', c' in $\mathbb{F}_{q^m}$ such that the number of solutions of $aX^d + bY^d = cZ^d$ over $\mathbb{F}_{q^m}$ is divisible by p^μ but not by $p^{\mu+1}$. In some sense this theorem completely solves the problem of divisibility for Fermat curves. Furthermore, the property of Lemma 1 is very important since from it we obtain a best possible Moreno-Moreno's p-adic Serre bound (see Theorem 1).

Our next theorem shows how or system of modular equations (2) can in some cases be reduced to a single equation. This considerably lowers the complexity of our computational problem.

Proposition 2. *Let d be a divisor of $p^f - 1$. Consider the diagonal equation $aX^d + bY^d = cZ^d$ over $\mathbb{F}_{p^{mf}}$. Let*

$$\lambda = \min_{1 \le c \le d-1} \sigma_p(c(p^f - 1)/d).$$

Then $p^{(\frac{3\lambda}{p-1} - f)m}$ divides the number of solutions of $aX^d + bY^d = cZ^d$ over $\mathbb{F}_{p^{fm}}$.

Proof. Note that if $\sigma_p(c(p^f - 1)/d) \ge \lambda$ for $1 \le c \le d$. Then $\sigma_p(j_1) + \sigma_2(j_2) + \sigma(j_3) \ge 3\lambda$.

Remark 4. In many cases we have that $\min_{1 \le c \le d-1} \sigma_p(c(p^f - 1)/d) = \sigma_p((p^f - 1)/d)$.

Example 1. Let $d = 23$ and $\mathbb{F}_{2^f} = \mathbb{F}_{2^{11}}$. In this case we compute

$$\min_{1 \le c \le 22} \sigma_2(c(2^{11} - 1)/23).$$

We have that $\sigma_2(c(2^{11}-1)/23) = 4$ for $c \in \{1, 2, 3, 4, 6, 8, 9, 12, 13, 16, 18\}$. Hence $\min_{1 \le c_i \le 22} \sigma_2(c_1(2^{11} - 1)/23) + \sigma_2(c_2(2^{11} - 1)/23)) + \sigma_2(c_2(2^{11} - 1)/23)) = 12$ since $c_1 = 1, c_2 = 4$ and $c_3 = 18$ gives a solution of (5). Applying Proposition 1 and Theorem 2, we obtain the best divisibility for the families curves $\mathcal{G} = \{aX^{23} + bY^{23} = cZ^{23} \,|\, a, b, c \in \mathbb{F}^{\times}_{2^{11m}}\}$. Hence there is an equation $a_0X^{23} + b_0Y^{23} = c_0Z^{23} \in \mathcal{G}$ with exact divisibility 2^m.

Example 2. Let $d = 151$ and $\mathbb{F}_{2^f} = \mathbb{F}_{2^{15}}$. In this case we compute

$$\min_{1 \le c \le 150} \sigma_2(c(2^{15} - 1)/151).$$

We have that $\sigma_2(c(2^{15} - 1)/151) = 5$. Hence $\min_{1 \le c_i \le 150} \sigma_2(c_1(2^{15} - 1)/151) + \sigma_2(c_2(2^{15} - 1)/151)) + \sigma_2(c_2(2^{15} - 1)/151)) = 15$ since $c_1 = 57, c_2 = 19$ and $c_3 = 4(\sigma_2(c_i(2^{15} - 1)/151) = 5$ for $i = 1, 2, 3)$ gives a solution of (5). Applying Proposition 1 and Theorem 2, we obtain the best divisibility for the families curves $\mathcal{G} = \{aX^{151} + bY^{151} = cZ^{151} \,|\, a, b, c \in \mathbb{F}^{\times}_{2^{15m}}\}$. Hence there is an equation $a_0X^{151} + b_0Y^{151} = c_0Z^{151} \in \mathcal{G}$ where 2 does not divide its number of solutions over $\mathbb{F}_{2^{15m}}$.

Example 3. Let $d = 23^2 = 529$ and $\mathbb{F}_{2^f} = \mathbb{F}_{2^{253}}$ (The first finite field of characteristic 2 satisfying that 529 divides $2^f - 1$ is $\mathbb{F}_{2^{253}}$). In this case we compute

$$\min_{1 \le c \le 528} \sigma_2(c(2^{253} - 1)/529).$$

We have that $\sigma_2(c(2^{253} - 1)/151) = 92$. We have that $\sigma_2(c(2^{253} - 1)/529) = 92$ for

$$c \in \{23, 46, 69, 92, 138, 184, 207, 276, 299, 368, 414, 500\}.$$

Hence $\min_{1 \le c_i \le 529} \sigma_2(c_1(2^{253} - 1)/529) + \sigma_2(c_2(2^{253} - 1)/529)) + \sigma_2(c_2(2^{253} - 1)/259)) = 276$ since $c_1 = 23, c_2 = 92$ and $c_3 = 414(\sigma_2(c_i(2^{253} - 1)/529) = 5$ for $i = 1, 2, 3$) gives a solution of (5). Applying Proposition 1 and Theorem 2, we obtain the best divisibility for the families curves $\mathcal{G} = \{aX^{529} + bY^{529} = cZ^{529} \mid a, b, c \in \mathbb{F}^{\times}_{2^{253m}}\}$. Hence there is an equation $a_0X^{529} + b_0Y^{529} = c_0Z^{529} \in \mathcal{G}$ with exact divisibility 2^{23m}.

Example 3 is an example where $\min_{1 \le c \le d-1} \sigma_p(c(p^f - 1)/d) \ne \sigma_p((p^f - 1)/d)$. Also note that in Example 3 we computed μ for a large finite field.

3 Tables

In the following tables, we are going to calculate μ for the curves $aX^d + bY^d = cZ^d$ over $\mathbb{F}_{p^f}$,where f is odd, in order to apply Theorem 1.

In Table 1 we compute μ for the first f such that d divides $2^f - 1$. Recall that if we know μ for the first f such that d divides $2^f - 1$, the we know μ for all the extensions of $\mathbb{F}_{2^f}$ (see Proposition 1). Note that we can assume that d is odd since the characteristic of $\mathbb{F}_{2^f}$ is 2.

Table 1. Best Divisibility of $aX^d + bY^d = cZ^d$ over $\mathbb{F}_{2^f}$

d	smallest f such that d divides $2^f - 1$.	μ
23	11	1
47	23	4
71	35	7
529	253	23

Table 2. Best Divisibility of $aX^d + bY^d = cZ^d$ over $\mathbb{F}_{3^f}$

d	smallest f such that d divides $3^f - 1$.	μ
11	5	1
23	11	1
46	11	1
47	23	4
59	29	10

In Table 2 we compute μ for the first f such that d divides $3^f - 1$. Recall that if we know μ for the first f such that d divides $3^f - 1$, then we know μ for all the extensions of $\mathbb{F}_{3^f}$ (see Proposition 1). Note that we can assume that d is not divisible by 3 since the characteristic of $\mathbb{F}_{3^f}$ is 3.

Theorem 3. *Let $aX^d + bY^d = cZ^d$ be a Fermat curve of the tables. Then $aX^d + bY^d = cZ^d$ satisfies Theorem 1, where μ is given by the table.*

4 Semiprimitive Fermat Curves

In this section we obtain a general family of Fermat curves satisfying Theorem 1, generalizing the results of Tables 1,2,3.

Now we are going to consider odd primes l for which p is of order exactly $(l-1)/2$, i.e., the smallest positive integer k for which $p^k \equiv 0 \bmod l$. We call p a semiprimitive root for such l. Note that 2 is a semiprimitive root for $l = 7, 23, 47, 71$. We would obtain a new family of Fermat curves that satisfy Theorem 1.

Let $g(j)$ be the Gauss sum defined by:

$$g(j) = \sum_{x \in \mathbb{F}_q^\times} \chi^{-j}(x)\psi(x),$$

where χ is multiplicative character of order $q-1$ and ψ is an additive character of $\mathbb{F}_q$. In [2], Moreno-Moreno proved that

$$S(l) = \sum_{x \in \mathbb{F}_q} (-1)^{Tr(x^l)} = \frac{l-1}{2}\{g(\frac{q-1}{l}) + g(q-1-\frac{q-1}{l})\}. \tag{7}$$

This implies that 2^λ divides $S(l)$, where $l = \min\{\sigma_2((q-1)/l), \sigma_2((q-1)-((q-1)/l))\}$. They proved the above identity for finite fields of characteristic 2. The proof for arbitrary characteristic follows from their proof using $g(j) = g(p^a j)$.

Table 3. Best Divisibility of $aX^d + bY^d = cZ^d$ over $\mathbb{F}_{p^f}$

d	smallest f such that d divides $p^f - 1$.	μ
11	$\mathbb{F}_{5^5}$	1
38	$\mathbb{F}_{5^9}$	2
20	$\mathbb{F}_{7^7}$	2
31	$\mathbb{F}_{7^{15}}$	3
37	$\mathbb{F}_{7^9}$	3
58	$\mathbb{F}_{7^7}$	1
43	$\mathbb{F}_{11^7}$	2
23	$\mathbb{F}_{13^{11}}$	1
46	$\mathbb{F}_{13^{11}}$	1
53	$\mathbb{F}_{13^{53}}$	5
19	$\mathbb{F}_{17^9}$	3
38	$\mathbb{F}_{19^9}$	2

Lemma 3. *Let $q = p^{(l-1)/2}$ and let p be a prime for which p is a semiprimitive root for l. Given $aX^l + bY^l = cZ^l$ over $\mathbb{F}_{q^m}$, the μ of (3) is such that $\mu > 0$, whenever 3 does not divide $(l-1)(p-1)/2$.*

Proof. Using Proposition 1, we need only estimates μ of (3) for the finite field $\mathbb{F}_q$. Let $f = (l-1)/2$. First we consider the solutions of $aX^l + bY^l = cZ^l$ over $\mathbb{F}_q$. We have the following modular system associated to $aX^l + bY^l = cZ^l$:

$$\begin{aligned} lj_1 &\equiv 0 \bmod q-1 \\ lj_2 &\equiv 0 \bmod q-1 \\ lj_3 &\equiv 0 \bmod q-1 \\ j_1 + j_2 + j_3 &\equiv 0 \bmod q-1 \end{aligned} \tag{8}$$

By the identity (7), we have that $\sigma_2(c(q-1)/l)) = \sigma_2((q-1)/l)$ or $\sigma_2(q-1-((q-1)/l))$. Note that $\sigma_2((q-1)/l) + \sigma_2(q-1-((q-1)/l)) = f(p-1)$. If $\sigma_2(j_{k_1}) \neq \sigma_2(j_{k_2})$, then $\sigma_2(j_{k_1}) + \sigma_2(j_{k_2}) + \sigma_2(j_{k_3}) > (p-1)f$. Hence we can assume that the minimal solution of (8) satisfies $\sigma_2(j_1) = \sigma_2(j_2) = \sigma_2(j_3)$. Applying the function σ_2 to the last modular equation of (8), we obtain $\sigma_2(j_1) + \sigma_2(j_2) + \sigma_2(j_3) \geq f(p-1)$. Therefore

$$\mu = \min \sigma_2(j_1) + \sigma_2(j_2) + \sigma_2(j_3) = 3 \min \sigma_2(j_1) \geq f(p-1).$$

Hence $\mu \geq 1$ whenever 3 does not divide $(l-1)(p-1)/2$. Hence at least p^μ divides $|N(\mathbb{F}_q)|$. Then by Lemma 2, we obtain that $p^{\mu m}$ divides $|N(\mathbb{F}_{q^m})|$.

Now we state a p-adic Serre bound for the Fermat curves of Lemma 3.

Theorem 4. *Let $q = p^{(l-1)/2}$ and let l be an odd prime for which p is a semiprimitive root for l. Let μ be as defined in (3) for the curve $aX^l + bY^l = cZ^l$ over $\mathbb{F}_{q^m}$. Then*

$$||\tilde{N}| - (q^m + 1)| \leq \frac{(p-1)(p-2)}{2} p^{\mu m} [q^{m/2} p^{1-\mu m}],$$

whenever 3 does not divide $(l-1)(p-1)/2$.

Futhermore, we have $\mu \geq 1$ by Lemma 3 .

Proof. Combining Lemma 3 and Theorem 1, we obtain the result.

We apply Theorem 4 to some semiprimitive primes.

Example 4. Note 2 is a semiprimitive root for 23 and $\mu = 1$. Applying Theorem 4, we obtain

$$||\tilde{N}| - (2^{11m} + 1)| \leq 231 \times 2^m [2^{(9m+2)/2}].$$

Example 5. Note 2 is a semiprimitive root for 47 and $\mu = 4$. Applying Theorem 4, we obtain

$$||\tilde{N}| - (2^{23m} + 1)| \leq 1035 \times 2^{4m} [2^{(15m+2)/2}].$$

In particular, for the finite field $\mathbb{F}_{2^{69}}$, Serre improvement to Weil's bound gives $1035 \times [2 \times 2^{69/2}] = 50292728269650$ and our improvement gives $1035 \times 2^{12} \times [2^{47/2}] = 50292727418880$.

Example 6. Note 2 is a semiprimitive root for 71 and $\mu = 7$. Applying Theorem 4, we obtain

$$||\tilde{N}| - (2^{35m} + 1)| \leq 2415 \times 2^{7m}[2^{(21m+2)/2}].$$

Remark 5. Using our computations of Table 1 we have obtained the above best bounds. Notice that each example of μ gives a family of bounds.

5 Conclusion

The main result of this paper is obtaining a general class(the semiprimitive case presented in the last section) of Fermat curves for which Weil-Serre's bound can be improved using Moreno-Moreno p-adic techniques. We also prove that for each particular case, the best bound μ is computed in a simple computation which is presented in the second section.

Acknowledgment

We would like to thank the referee for his careful reading and for his generous suggestions that helped improved the paper considerably.

References

1. O. Moreno and C. J. Moreno, A p-adic Serre Bound, *Finite Fields and Their Applications,* **4:**(1998), pp. 241-244.
2. O. Moreno and C.J. Moreno, The MacWilliams-Sloane Conjecture on the Tightness of the Carlitz-Uchiyama Bound and the Weights of Duals of BCH Codes, *IEEE Trans. Inform. Theory*, **4:**6(1994), pp. 1894-1907.
3. O. Moreno, K. Shum, F. N. Castro and P.V. Kumar, Tight Bounds for Chevalley-Warning-Ax Type Estimates, with Improved Applications, *Proc. of the London Mathematical Society*, **4** (2004) pp. 201-217.

Nonbinary Quantum Codes from Hermitian Curves

Pradeep Kiran Sarvepalli and Andreas Klappenecker

Texas A&M University, Department of Computer Science,
College Station, TX 77843-3112
klappi@cs.tamu.edu

In memory of Thomas Beth

Abstract. The extreme sensitivity of quantum information to ambient noise has prompted the study of quantum error-correcting codes. In this paper two families of quantum error-correcting codes are constructed based on Hermitian curves. Unlike the case of classical codes, it is a difficult, sometimes impossible, task to puncture a quantum stabilizer code. The puncture code of Hermitian codes is computed that allows one to determine the admissible puncturings. A large number of punctured Hermitian codes are derived by exploiting known facts about the weight distribution of these puncture codes.

1 Introduction

Since the discovery of powerful quantum algorithms, most notably a polynomial time algorithm for factorization by Shor and a quantum search algorithm by Grover, quantum computing has received a lot of interest. While a practical implementation of these algorithms is far away, it has nonetheless become clear that some form of error correction is required to protect quantum data from noise. This was the motivation for the development of quantum error-correcting codes. Many quantum codes can be derived from classical linear codes using the quantum stabilizer code formalism. Therefore, it is natural to use algebraic geometric codes to develop quantum codes. Quantum algebraic geometric codes have been studied, for instance, in [2, 11, 12].

In this paper, we consider quantum stabilizer codes that are derived from Hermitian codes (see [3, 14, 15, 16] for some background on classical Hermitian codes). Quantum codes derived from Hermitian curves for the binary case have been investigated in [12] and for prime alphabets in [11].

We hope to present a more elementary treatment of nonbinary Hermitian quantum codes, taking advantage of the approach developed in [9] for classical Hermitian codes. We present a new family of quantum Hermitian codes that are derived from classical codes which are self-orthogonal with respect to a Hermitian inner product. Furthermore, we study the puncturing of Hermitian quantum codes and derive, in closed form, the so-called puncture code. The puncture

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 136–143, 2006.

code is a combinatorial object that contains the information about the admissible lengths to which one can shorten a quantum stabilizer code. A result by Stichtenoth on the weight distribution of classical Hermitian codes enables us to derive numerous punctured Hermitian codes.

The paper is organized as follows. After reviewing briefly the essentials of quantum codes, we review the classical Hermitian codes. Following this we proceed with the construction of the quantum codes. In section 4 we shall take up the task of puncturing these quantum codes.

Notation: The euclidean inner product of two vectors x and y in $\mathbf{F}_q^n$ is denoted by $\langle x|y\rangle = x_1y_1 + \cdots x_ny_n$ while the Hermitian inner product of two vectors x and y in $\mathbf{F}_{q^2}^n$ is denoted by $\langle x|y\rangle_h = x_1y_1^q + \cdots x_ny_n^q$. We write $C^{\perp}$ for the euclidean dual of a code $C \subseteq \mathbf{F}_q^n$, and $D^{\perp}$ for the Hermitian dual of a code $D \subseteq \mathbf{F}_{q^2}^n$.

2 Preliminaries

2.1 Quantum Codes

In this section we quickly review the essentials of quantum stabilizer codes. The reader is referred to [4] and [7] for an introduction to quantum error correction; all necessary background on nonbinary stabilizer codes is contained in [1, 10, 13]. Every quantum stabilizer code is a subspace in the q^n-dimensional complex vector space $\mathbf{C}^q$. Let G denote all the matrix operators on $\mathbf{C}^q$. A quantum stabilizer code is given by a joint $+1$ eigenspace of a finite abelian subgroup of G. This subgroup is called the stabilizer of the code, whence the name stabilizer codes.

For the purposes of this paper, it is sufficient to know that a stabilizer code can be associated with a self-orthogonal classical code. In the next two lemmas, we recall two simple constructions of quantum stabilizer codes. Essentially, these lemmas show that the existence of certain classical codes ensures the existence of quantum stabilizer codes. A stabilizer code which encodes k q-ary bits into n q-ary bits is denoted by $[[n, k, d]]_q$. It is a q^k-dimensional subspace of $\mathbf{C}^q$ and is able to detect $< d$ errors. The first construction makes use of a pair of nested linear codes over $\mathbf{F}_q$.

Lemma 1 (CSS Construction). *Let C_1 and C_2 be linear codes over the finite field $\mathbf{F}_q$ respectively with parameters $[n, k_1, d_1]_q$ and $[n, k_2, d_2]_q$. If $C_1 \subset C_2$ and $d = \min \mathrm{wt}\{(C_2 \setminus C_1) \cup (C_1^{\perp} \setminus C_2^{\perp})\}$, then there exists an $[[n, k_2 - k_1, d]]_q$ quantum code, which is pure if $\mathrm{wt}(C_2 \setminus C_1) = \mathrm{wt}(C_2)$ and $\mathrm{wt}(C_1^{\perp} \setminus C_2^{\perp}) = \mathrm{wt}(C_1^{\perp})$.*

Proof. See [4] for the CSS construction of binary codes and for q-ary generalizations see [6, Theorem 3] or [10]. For details on the definition of pure quantum codes see [4] or [7].

In the above Lemma the code $C = C_1 \oplus C_2^{\perp}$ is self-orthogonal with respect to the symplectic product [10]. The second construction makes use of a classical code over $\mathbf{F}_{q^2}$ that is self-orthogonal with respect to the Hermitian inner product.

Lemma 2. *Let C be a linear $[n,k]_{q^2}$ contained in its Hermitian dual, $C^{\perp}$, such that $d = \min\{wt(C^{\perp} \;\setminus C)\}$. Then there exists an $[[n, n-2k, d]]_q$ quantum code.*

Proof. See [6, Corollary 2].

2.2 Hermitian Codes

We will give a brief description of the classical Hermitian codes (cf. [9]). Consider the Hermitian curve $h(x,y) = x^{q+1} - y^q - y$, where x, y are in $\mathbf{F}_{q^2}$. Let $\{P_1, P_2, \ldots, P_n\}$ denote the zeros of $h(x,y) = 0$. It can be shown that $n = q^3$ [9, page 152]. Now let $x^a y^b$ be an element in $\mathbf{F}_{q^2}[x,y]$. Then we define the order of $x^a y^b$ as $\rho(x^a y^b) = aq + b(q+1)$. Given a function $f(x,y) = \sum_{i,j} f_{ij} x^i y^j$ in $\mathbf{F}_{q^2}[x,y]$ we define the order of $f(x,y)$ as

$$\rho\left(f(x,y)\right) = \max_{f\ \neq 0}\{\rho(x^i y^j)\}. \tag{1}$$

Let $M(s) = \{x^i y^j \mid 0 \le i, 0 \le j \le q-1, 0 \le \rho(x^i y^j) \le s\}$ and let $\mathbf{L}(s)$ be the $\mathbf{F}_{q^2}$ subspace spanned by the monomials in $M(s)$.

The q^2-ary Hermitian code of order s denoted by $\mathbf{H}_{q^2}(s)$ is an $[n, k(s), d(s)]_{q^2}$ code defined as

$$\mathbf{H}_{q^2}(s) = \{(f(P_1), f(P_2), \ldots, f(P_n)) \mid f \in \mathbf{L}(s)\}. \tag{2}$$

Let $\tilde{s}$ be defined as follows.

$$\tilde{s} = \max\{l \mid l = iq + j(q+1) \le s, 0 \le i, 0 \le j \le q-1\}. \tag{3}$$

Then the dimension and the distance of the Hermitian code for various values of s are given in the following table [17].

s	$k(s)$	$d(s)$
$0 \le s \le q^2 - q - 2$ $\tilde{s} = aq + b$ $0 \le b \le a \le q-1$	$a(a+1)/2 + b + 1$	$n - \tilde{s}$
$q^2 - q - 2 < s < n - q^2 + q$	$s + 1 - q(q-1)/2$	$n - s$
$n - q^2 + q \le s < n$ $s = n - q^2 + aq + b$ $0 \le a, b \le q-1$	$s + 1 - q(q-1)/2$	$n - s$ if $a < b$ $n - s + b$ if $a \ge b$
$n \le s \le n + q^2 - q - 2$ $s^{\perp} = n + q^2 - q - 2 - s$ $\tilde{s}^{\perp} = aq + b$ $0 \le b \le a \le q-1$	$n - a(a+1)/2 - b - 1$	$a + 2$ if $b = a$ $a + 1$ if $b < a$

(4)

Further, the dual code is also a Hermitian code. Let $s^{\perp}$ be the order of the dual code defined as

$$s^{\perp} = n + q^2 - q - 2 - s. \tag{5}$$

Then the dual code of $\mathbf{H}_{q^2}(s)$ is given by [3], [9, page 154]

$$\mathbf{H}_{q^2}(s)^{\perp} = \mathbf{H}_{q^2}(s^{\perp}) = \mathbf{H}_{q^2}(n + q^2 - q - 2 - s). \tag{6}$$

3 Quantum Hermitian Codes

In this section we will construct some quantum codes from the classical Hermitian codes.

Theorem 3. *For* $0 \leq s_1 < s_2 \leq n+q^2-q-2$*, there exists an* $[[n, k(s_2)-k(s_1), \geq \min\{d(s_2), d(s_1^\perp)\}]]_{q^2}$ *quantum code of length* $n = q^3$*, where* $k(s)$ *and* $d(s)$ *are given by equation (4).*

Proof. From the definition of the Hermitian codes we can see that if $s_1 \leq s_2$ $C_1 = \mathbf{H}_{q^2}(s_1) \subseteq \mathbf{H}_{q^2}(s_2) = C_2$. We can see that $\mathrm{wt}(C_2 \setminus C_1) \geq d(s_2)$ and $\mathrm{wt}(C_1^\perp \setminus C_2^\perp) \geq d(s_1^\perp)$. By Lemma 1 we have an $[[n, k(s_2)-k(s_1), \geq \min\{d(s_2), d(s_1^\perp)\}]]_{q^2}$ quantum code, where $k(s)$ and $d(s)$ are dimension and distance of $\mathbf{H}_{q^2}(s)$ respectively as given by equation (4).

For certain values of s_1, s_2 we can be more precise about the distance of the quantum code and also the purity of the quantum code.

Corollary 4. *If* $2q^2 - 2q - 2 < s_1 < s_2 < n - q^2 + q$ *then there exists a pure* $[[n, s_2 - s_1, \min\{n - s_2, 2 + s_1 + q - q^2\}]]_{q^2}$ *quantum code, with* $n = q^3$*.*

Proof. If $q^2 - q - 2 < s < n - q^2 + q$, then the distance of the Hermitian code is given by $n - s$. Further if $s_1 < s_2$, then $d(s_2) < d(s_1)$ which implies that $\mathrm{wt}(C_2 \setminus C_1) = d(s_2)$. Since $2q^2 - 2q - 2 < s_1 < n - q^2 + q$, $2q^2 - 2q - 2 < s_1^\perp < n - q^2 + q$, which implies $d(C_1^\perp \setminus C_2^\perp) = d(s_1^\perp)$. Further $k(s_2) = s_2 + 1 - (q^2 - q)/2$ and $k(s_1) = s_1 + 1 - (q^2 - q)/2$. Thus from Lemma 1 we see that there exists a pure $[[n, s_2 - s_1, \min\{n - s_2, 2 + s_1 + q - q^2\}]]_{q^2}$ quantum code.

Remark. The previous work on Hermitian codes restricts the range of s_1, s_2. We have been able to extend this using the results of [17].

Our next goal will be to construct quantum codes from self-orthogonal codes over $\mathbf{F}_{q^2}$. First we need the following result on the order functions.

Lemma 5. *Suppose that* $f(x, y)$ *and* $g(x, y)$ *are two arbitrary functions in* $\mathbf{F}_{q^2}[x, y]$*. We have*

$$\rho(f(x, y)g(x, y)) \leq \rho(f(x, y)) + \rho(g(x, y))$$

for all zeros (x, y) *of the curve* $x^{q+1} - y^q - y = 0$*.*

Proof. Let $f(x, y) = \sum_{i,j} f_{i,j} x^i y^j$, such that $\rho(f(x, y)) = s_1$ and $g(x, y) = \sum_{a,b} g_{a,b} x^a y^b$, such that $\rho(g(x, y)) = s_2$. Let $x^a y^b$, $x^i y^j$ be any two monomials such that $x^a y^b$ is in $M(s_1)$ and $x^i y^j$ is in $M(s_2)$. Then we have $aq + b(q+1) \leq s_1$, $iq + j(q+1) \leq s_2$, where $0 \leq i, a$ and $0 \leq b, j \leq q-1$. We can find $\rho(x^a y^b x^i y^j) = \rho(x^{a+i} y^{b+j})$ as follows. If $b+j \leq q-1$, then $\rho(x^{a+i} y^{b+j}) = q(a+i) + (q+1)(b+j) \leq s_1 + s_2$. If on the other hand $q \leq b + j \leq 2q - 2$, then because we are evaluating on the curve $x^{q+1}-y^q-y=0$, we can write $y^{b+j}=y^{q+\beta}=(x^{q+1}-y)y^\beta$, where $0\le\beta\le q-2$. Then

$$\begin{aligned}\rho(x^{a+i}y^{b+j}) &= \rho(x^{a+i}(x^{q+1}-y)y^\beta)=\rho(x^{a+i+q+1}y^\beta-x^{a+i}y^{\beta+1}),\\ &=\max\{q(a+i+q+1)+\beta(q+1),q(a+i)+(\beta+1)(q+1)\},\\ &=\max\{q(a+i)+(\beta+q)(q+1),q(a+i)+(\beta+1)(q+1)\},\\ &=q(a+i)+(\beta+q)(q+1)=q(a+i)+(q+1)(b+j),\\ &=\rho(x^ay^b)+\rho(x^iy^j)\le s_1+s_2=\rho(f(x,y))+\rho(g(x,y)). \qquad (7)\end{aligned}$$

We have $f(x,y)g(x,y)=\sum_{i,j}\sum_{a,b}f_{i,j}g_{a,b}x^{i+a}y^{j+b}$, hence

$$\begin{aligned}\rho(f(x,y)g(x,y)) &= \rho(\sum_{i,j}\sum_{a,b}f_{i,j}g_{a,b}x^{i+a}y^{j+b}),\\ &\le \max_{(i,j);(a,b)}\rho(x^{i+a}y^{j+b}),\\ &\le \rho(f(x,y))+\rho(g(x,y)),\end{aligned}$$

where the last inequality is due to equation (7).

Corollary 6. *If $f(x,y)$ is in $\boldsymbol{L}(s_1)$ and $g(x,y)$ is in $\boldsymbol{L}(s_2)$, then $f(x,y)g(x,y)$ is in $\boldsymbol{L}(s_1+s_2)$.*

Lemma 7. *For $0\le s\le q^2-2$ the Hermitian codes are self-orthogonal with respect to the Hermitian inner product.*

Proof. Consider any monomial x^iy^j in $M(s)$. Then by Lemma 5, $\rho((x^iy^j)^q)\le qs$. So for any monomial x^iy^j in $M(s)$, its conjugate $x^{qi}y^{qj}$ is present in $M(s^\perp)$ if $\rho(x^{qi}y^{qj})\le s^\perp$. This means

$$\begin{aligned}qs &\le s^\perp=q^3+q^2-q-2-s,\\ s&\le q^2-2.\end{aligned}$$

Let *ev* be the evaluation map on f, so that *ev* $f=(f(P_1),f(P_1),\ldots,f(P_n))$. For any two monomials $x^{i_1}y^{j_1}$, $x^{i_2}y^{j_2}$ in $M(s)$ we have

$$\langle ev\ x^{i_1}y^{j_1}\mid ev\ x^{i_2}y^{j_2}\rangle_h=\langle ev\ x^{i_1}y^{j_1}\mid ev\ x^{qi_2}y^{qj_2}\rangle=0,$$

where the inner product vanishes because *ev* $x^{qi_2}y^{qj_2}$ is in $\mathbf{H}_{q^2}(s)^\perp$ for $s\le q^2-2$. Since the monomials span $\mathbf{L}(s)$, $\mathbf{H}_{q^2}(s)$ is self-orthogonal with respect to the Hermitian inner product for $0\le s\le q^2-2$.

Theorem 8. *For $0\le s\le q^2-2$ there exist quantum codes $[[n,n-2k(s),\ge d(s^\perp)]]_q$, where $n=q^3$ and $k(s),d(s)$ are given by equation (4).*

Proof. From Lemma 7 we know that $\mathbf{H}_{q^2}(s)$ is self-orthogonal with respect to the Hermitian product when $0\le s\le q^2-2$. The minimum distance of $\mathbf{H}_{q^2}(s)^\perp$ is the same as that of $\mathbf{H}_{q^2}(s)^\perp$. From Lemma 2 we can conclude that there exists an $[[n,n-2k(s),\ge k(s^\perp)]]_q$ quantum code where $d(s^\perp)$ and $k(s)$ are given by equation (4).

Remark. None of the previously cited work make use of the Hermitian inner product to derive quantum codes from Hermitian curves.

4 Puncturing the Hermitian Codes

In this section we will show how to reduce the length (and the dimension) of the previously constructed quantum codes while preserving the minimum distance. In the literature on quantum codes it is common to refer to this operation as puncturing or shortening, though strictly speaking these are distinct operations in classical coding theory. The key tool for finding when puncturing is possible is determined by the puncture code proposed by Rains [13]. Here we will briefly review the idea of puncture code.

Given a code $C \subseteq \mathbf{F}_q^{2n}$ such that $C = C_1 \oplus C_2^{\perp}$ we define the puncture code of C as [6, Theorem 7]

$$P(C) = \{(a_i b_i)_{i=1}^n \mid a \in C_1, b \in C_2^{\perp}\}^{\perp}. \tag{8}$$

Curiously, the weight distribution of $P(C)$ determines the lengths to which we can puncture the quantum code formed from C_1, C_2 using the CSS construction. In general we are hampered by the fact that $P(C)$ is not known in closed form and even if it is known its weight distribution is not known in general. For the Hermitian codes however we are able to compute a subcode of $P(C)$. Finding the weight distribution of any code in general is a difficult. In a short paper on Hermitian codes [14] Stichtenoth discovered a rather curious result on the allowable weights in Hermitian codes. These two results will allow us to derive many punctured Hermitian codes.

Lemma 9. *The Hermitian code $\boldsymbol{H}_{q^2}(s)$ has a vector of weight $r \geq d(s)$ if $n-r \leq \min\{n-q^2, s\}$ can be expressed as $n-r = iq$ or $n-r = iq + j(q+1)$, where $0 \leq i, 0 \leq j \leq q-1$.*

Proof. See propositions 2,3 in [14].

Theorem 10. *For $0 \leq s_1 < s_2 \leq n + q^2 - q - 2$ and $0 \leq \sigma \leq s_2 - s_1$, if $\boldsymbol{H}_{q^2}(\sigma)$ has a vector of weight r, then there exists an $[[r, \geq (k(s_2) - k(s_1) - (n-r)), \geq d]]_{q^2}$ quantum code, where $n = q^3$ and $d = \min\{d(s_2), d(s_1^{\perp})\}$. It is possible to puncture to a length $r = d(\sigma)$ and all r such that $n - r \leq \min\{n-q^2, \sigma\}$ and $n - r = iq + j(q+1)$ where $0 \leq i, 0 \leq j \leq q-1$.*

Proof. Let $C_i = \mathbf{H}_{q^2}(s_i)$ with $0 \leq s_1 \leq s_2 < n + q^2 - q - 2$, where $n = q^3$. An $[[n, k(s_2) - k(s_1), d]]_q$ quantum code $\mathcal{Q}$, with $d = \min\{d(s_2), d(s_1^{\perp})\}$ exists by Lemma 3. Define C to be the direct sum of $C_1, C_2^{\perp}$ viz. $C = C_1 \oplus C_2^{\perp}$. Then the puncture code $P(C)$ is given by

$$P(C) = \{(a_i b_i)_{i=1}^n \mid a \in C_1, b \in C_2^{\perp}\}^{\perp}. \tag{9}$$

By Corollary 6 we see that $P(C)^{\perp} = \mathbf{H}_{q^2}(s_1 + s_2^{\perp})$, so

$$P(C) = \mathbf{H}_{q^2}(n + q^2 - q - 2 - s_1 - s_2^{\perp}) = \mathbf{H}_{q^2}(s_2 - s_1). \tag{10}$$

If there exists a vector of weight r in $P(C)$, then there exists a quantum code of length r and distance $d' \geq d$ obtained by puncturing $\mathcal{Q}$ (cf. [6]). Since

$P(C) = \mathbf{H}_{q^2}(s_2 - s_1) \supseteq \mathbf{H}_{q^2}(\sigma)$ for all $0 \le \sigma \le s_2 - s_1$, the weight distributions of $\mathbf{H}_{q^2}(\sigma)$ give all the lengths to which $\mathcal{Q}$ can be punctured. By Lemma 9 vectors with weight r exist in $\mathbf{H}_{q^2(\sigma)}$, provided $n-r = iq+j(q+1) \le \min\{n-q^2, \sigma\}$ and $0 \le i, 0 \le j \le q-1$. In particular $P(C)$ will contain codewords whose weight $r = d(\sigma)$ which is equal to the minimum weight of $\mathbf{H}_{q^2}(\sigma)$. Thus there exist punctured quantum codes with the parameters $[[r, \ge (k(s_2) - k(s_1) - (n-r)), \ge d]]_{q^2}$.

Slightly redefining the puncture code allows us to puncture quantum codes constructed via Theorem 8.

Theorem 11. *Let $C = \boldsymbol{H}_{q^2}(s)$ with $0 \le s \le q^2 - 2$ and $(q+1)s \le \sigma \le n + q^2 - q - 2$. If the puncture code $P(C) \supseteq \boldsymbol{H}_{q^2}(\sigma)^\perp|_{\mathbf{F}}$ contains a vector of weight r, then there exists an $[[r, \ge (n - 2k(s) - (n-r)), \ge d(s^\perp)]]_q$ quantum code.*

Proof. For a linear code $C \subseteq \mathbf{F}_{q^2}^n$, the puncture code $P(C)$ is given as [6, Theorem 13]

$$P(C) = \{\mathrm{tr}_{q^2/q}(a_i b_i^q)_{i=1}^n \mid a, b \in C\}^\perp. \qquad (11)$$

Let us first consider the code D given by

$$D = \{(a_i b_i^q)_{i=1}^n \mid a, b \in C\}. \qquad (12)$$

If $C = \mathbf{H}_{q^2}(s)$, then by Corollary 6 we can see that $D \subseteq \mathbf{H}_{q^2}(\sigma)$ for $(q+1)s \le \sigma \le n + q^2 - q - 2$ and $P(C)^\perp = \mathrm{tr}_{q^2/q}(D) \subseteq \mathbf{H}_{q^2}(\sigma)$. By Delsarte's theorem [5] the dual of the trace code is the restriction of the dual code. That is

$$\mathrm{tr}_{q^2/q}(D)^\perp = D^\perp|_{\mathbf{F}} \qquad (13)$$

$$P(C) = D^\perp|_{\mathbf{F}} \supseteq \mathbf{H}_{q^2}(\sigma)^\perp|_{\mathbf{F}} \qquad (14)$$

$$P(C) \supseteq \mathbf{H}_{q^2}(\sigma)^\perp|_{\mathbf{F}} = \mathbf{H}_{q^2}(n + q^2 - q - 2 - \sigma)|_{\mathbf{F}} \qquad (15)$$

From Theorem 8 there exists an $[[n, n - 2k(s), d(s^\perp)]]$ quantum code $\mathcal{Q}$ for $0 \le s \le q^2 - 2$. By [6, Theorem 11] $\mathcal{Q}$ can be punctured to all lengths which have nonzero distribution in $P(C)$. Thus if $P(C) \supseteq \mathbf{H}_{q^2}(\sigma)|_{\mathbf{F}}$ contains a vector of weight r, then there exists an $[[n - (n-r), \ge (n - 2k(s) - (n-r)), \ge d(s^\perp)]]_q$ code.

Remark. It is possible to extend the above theorem by making use of Lemma 9 as in Theorem 10.

Acknowledgment. The authors would like to thank the referees whose comments, especially on puncturing, helped to improve the paper. This research was supported by NSF grant CCR 0218582, NSF CAREER award CCF 0347310, a Texas A&M TITF initiative, and a TEES Select Young Faculty award.

References

1. Ashikhmin, A., Knill, E.: Nonbinary quantum stabilizer codes. IEEE Trans. on Inform. Theory, vol 47, 7, pp. 3065-3072, 2001.
2. Ashikhmin, A., Litsyn, S., Tsfasman, M. A.: Asymptotically good quantum codes. Phy. Rev. A, vol. 63, 032311, 2001.
3. Blake, I., Heegard, C., Høholdt, T., Wei, V.: Algebraic-geometry codes. IEEE Trans. Inform. Theory, vol. 44, 6, pp. 2596-2618 Oct. 1998.
4. Calderbank, A. R., Rains, E. M., Shor, P. W., Sloane, N. J. A.: Quantum error correction via codes over GF(4). IEEE Trans. Inform. Theory, vol 44, 4, pp. 1369-1387, Jul. 1998.
5. Delsarte, P.: On subfield subcodes of Reed-Solomon codes. IEEE Trans. Inform. Theory, vol. 21, 5, pp. 575-576, 1975.
6. Grassl, M., Beth, T.: On optimal quantum codes. International Journal of Quantum Information, vol. 2, 1, pp. 757-775, 2004.
7. Gottesman, D.: An introduction to quantum error-correction, in Quantum Computation: A Grand Mathematical Challenge for the Twenty-First Century and the Millennium, ed. Lomonaco, Jr., S. J., pp. 221-235, AMS, Rhode Island, 2002.
8. Høholdt, T., van Lint, J. H., R. Pellikaan, R.: Algebraic geometry codes, in the Handbook of coding theory, vol 1, pp. 871-961, Elsevier, Amsterdam, 1998.
9. Justesen, J., Høholdt, T.: A course in error-correcting codes. European Mathematical Society, Textbooks in Mathematics, 2004.
10. Ketkar, A., Klappenecker, A., Kumar, S., Sarvepalli, P. K.: Nonbinary stabilizer codes over finite fields. eprint quant-ph/0508070.
11. Kim, J.-L., Walker, J.: Nonbinary quantum error-correcting codes from algebraic curves. submitted to a special issue of Com^{2}MaC Conference on Association Schemes, Codes and Designs in Discrete Math, 2004.
12. Lee, C. H.: Quantum algebraic-geometric codes. Part III essay in dept. of Applied Mathematics and Theoretical Physics, University of Cambridge.
13. Rains, E.M.: Nonbinary quantum codes. IEEE Trans. Inform. Theory, vol. 45, 6, pp. 1827-1832, 1999.
14. Stichtenoth, H.: A note on Hermitian codes over $GF(q^2)$. IEEE Trans. Inform. Theory, vol. 34, 5, pp. 1345-1348 Jul. 1988.
15. Tiersma, H. J.: Remarks on codes from Hermitian curves. IEEE Trans. Inform. Theory, vol. 33, 4, pp. 605-609 Jul. 1987.
16. van Lint, J. H., Springer, T. A.: Generalized Reed-Solomon codes from algebraic geometry. IEEE Trans. Inform. Theory, vol. 33, 4, pp. 305-309 May 1987.
17. Yang, K., Kumar, P. V.: On the true minimum distance of Hermitian codes. Lecture notes in Mathematics, Coding theory and algebraic geometry, Proceedings, Luminy, 1991.

A Genetic Algorithm for Cocyclic Hadamard Matrices*

V. Álvarez, J.A. Armario, M.D. Frau, and P. Real

Dpto. Matemática Aplicada I, Universidad de Sevilla,
Avda. Reina Mercedes s/n 41012 Sevilla, Spain
{valvarez, armario, mdfrau, real}@us.es

Abstract. A genetic algorithm for finding cocyclic Hadamard matrices is described. Though we focus on the case of dihedral groups, the algorithm may be easily extended to cover any group. Some executions and examples are also included, with aid of MATHEMATICA 4.0.

1 Introduction

Over the past decade considerable effort has been devoted to computations of cocycles and (Hadamard) cocyclic matrices. On one hand, using classical methods involving the Universal Coefficient Theorem, Schur multipliers, inflation and transgression, two algorithms for finding 2-cocycles representing 2-dimensional cohomology classes and their correspondent cocyclic matrices have been worked out. The first one constitutes the foundational work on the subject [6, 7], and is applied over abelian groups. The second one [8] is applied over groups G for which the word problem is solvable.

On the other hand, Homological Perturbation Theory [12, 13, 16] provides computational short-cuts in a straightforward manner, by means of the so-called *(co)homological models*. This technique has been exploited in [11] from a cohomological point of view and more recently in [1, 2, 3] from a homological point of view.

From past literature it is evident that the search of Hadamard (cocyclic) matrices inherits high computational difficulty. In fact, though the use of the cocyclic construction of Hadamard matrices has permitted cut-downs in the search time, the search space still grows exponentially.

The work in [4] attempts to make an adaptation of image-processing techniques for the restoration of damaged images for the purpose of sampling the search space systematically. As a matter of fact, this approximation reveals to work whenever enough Hadamard cocyclic matrices are already known, from which the performance is then feasible.

Our aim is to provide a tool for generating Hadamard cocyclic matrices in an easy and (hopefully) fast way, which will complement in turn the work in [4]. Here

* All authors are partially supported by the PAICYT research project FQM–296 from Junta de Andalucía (Spain).

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 144–153, 2006.

we look at an adaptation of classical Genetic Algorithms [14, 15] for this purpose. Though it actually works on any group, we will focus on an improved version running on dihedral groups. Both of the genetic algorithms and all the executions and examples of the last section have been worked out with aid of Mathematica 4.0, running on a *Pentium IV 2.400 Mhz DIMM DDR266 512 MB*. It is a remarkable fact that, as far as we know, none of the algorithms already known has produced some Hadamard cocyclic matrices of large order (say $4t \geq 40$). The examples in Section 6 include some Hadamard cocyclic matrices of order 52. This way, our method seems to provide some cocyclic Hadamard matrices of larger order than those previously obtained with other algorithms.

We organize the paper as follows. Section 2 summarizes the classical methods already known for finding (Hadamard) cocyclic matrices. Section 3 is a brief introduction to Genetic Algorithms. The genetic algorithm itself for finding Hadamard cocyclic matrices is described in Section 4. The following section includes an improved version of the algorithm for the case of dihedral groups. As a matter of fact, both the search time and the search space are substantially optimized thanks to the work of the authors in [2, 3]. The last section is devoted to some examples.

2 Generating Cocyclic Matrices

The foundational work on cocyclic matrices is [6, 7], where a basis for 2-cocycles is codified in terms of a *development table*. Horadam and de Launey's method is based on an explicit version of the well-known Universal Coefficient Theorem, which provides a decomposition of the second cohomology group of G into the direct sum of two summands,

$$H^2(G,C) \cong Ext(G/[G,G],C) \oplus Hom(H_2(G),C).$$

The $Ext(G/[G,G],C)$ factor is referred as the *symmetric part*, and is completely determined from a presentation of G and the primary invariant decomposition of the abelianization $G/[G,G]$. The $Hom(H_2(G),C)$ factor is referred as the *commutator part* and turns out to be the difficult one to compute. The case of abelian groups is described in [6, 7]. Once a set of generators for both the symmetric and commutator parts is determined, it suffices to add a basis for 2-coboundaries of G, so that a basis for 2-cocycles is finally achieved.

Another method is described in [8], whenever the word problem is solvable in G. This method has already been implemented in [10], using the symbolic computational system MAGMA. Flannery calculates $H^2(G;C) \cong I \oplus T$ as the images of certain embeddings (called *inflation, I,* and *transgression*, T) which are complementary. Calculation of representative 2-cocycles associated to $Ext(G/[G,G],C)$ (inflation) is again canonical. However, calculation of a complement of the image by the embeddings of inflation I in $H^2(G,C)$ as the image of transgression is usually not canonical. As a matter of fact, it depends on the choice of a Schur complement of I in $H^2(G;C)$. If $|I|$ and $|T|$ are not coprime, there will be more than one complement of I in $H^2(G;C)$. This is a potential source of difficulties in computation of representative 2-cocycles associated with elements of $Hom(H_2(G),C)$. The case of dihedral groups and central extensions is described in [8,9,10].

Using the proper techniques of Homological Perturbation Theory [12,13,16], Grabmeier and Lambe present in [11] alternate methods for calculating representative 2-cocycles for all finite p–groups. They compute $H^2(G;C)$ straightforwardly from a *cohomological model* K of G. That is, a structure K such that $H^2(G;C) \cong H^2(K;C)$ and the computation of $H^2(K;C)$ is much simpler than that of $H^2(G;C)$.

One more approximation to this question, the so-called *homological reduction method*, is developed in another work of the authors [1,2,3]. Here *homological models* K for G are determined instead of cohomological models, in the sense that $H_*(K) \cong H_*(G)$ and $H_*(K)$ is computed substantially more easily than $H_*(G)$. The method developed in these papers covers any iterated product of central extensions and semidirect product of groups, so that dihedral groups D_{4t} are included. The genetic algorithm to be described in Section 4 is performed upon the calculations that the homological reduction method provides when it is applied over D_{4t}.

3 Preliminaries in Genetic Algorithms

Genetic algorithms (more briefly, GAs in the sequel) are appropriate for searching through large spaces, where exhaustive methods cannot be employed.

The father of the original Genetic Algorithm was John Holland who invented it in the early 1970's [14]. We next include a brief introduction to the subject. The interested reader is referred to [15] for more extensive background on GAs.

The aim of GAs is to mimic the principle of evolution in order to find an optimum solution for solving a given optimization problem. More concretely, starting from an initial "population" of potential solutions to the problem (traditionally termed *chromosomes*), some transformations are applied (may be just to some individuals or even to the whole population), as images of the "mutation" and "crossover" mechanisms in natural evolution. Mutation consists in modifying a "gene" of a chromosome. Crossover interchanges the information of some genes of two chromosomes.

Only some of these individuals will move on to the next generation (the more fit individuals, according to the optimization problem, in terms of the measure of an "evaluation function"). Here "generation" is a synonymous of iteration. The mutation and crossover transformations are applied generation through generation, and individuals go on striving for survival. After some number of iterations, the evaluation function is expected to measure an optimum solution, which solves the given problem. Although no bounds are known on the number of iterations which are needed to produce the fittest individual, it is a remarkable fact that GAs usually converge to an optimum solution significantly faster than exhaustive methods do. Indeed, GAs need not to explore the whole space.

4 Finding Hadamard Cocyclic Matrices by Means of GAs

We now set the directives for a genetic algorithm looking for Hadamard cocyclic matrices over a group G. In the following section we improve the design of the algorithm in the particular case of the dihedral group D_{4t}.

Let G be a group and $\mathcal{B} = \{\delta_1, \ldots, \delta_c, \beta_1, \ldots, \beta_s, \gamma_1, \ldots, \gamma_t\}$ a basis for 2-cocycles (according to the Hadamard pointwise product $\bullet$ of matrices). Here δ_i denote 2-coboundaries, β_i denote representative symmetric 2-cocycles (coming from inflation, i.e. factor $Ext(G/[G,G], \mathbb{Z}_2)$) and γ_i denote representative not symmetric 2-cocycles (coming from transgression, i.e. factor $Hom(H_2(G), \mathbb{Z}_2)$). Notice that in these circumstances the whole space of 2-cocycles consists of 2^{c+s+t} elements, and precisely 2^c of them are 2-coboundaries. Moreover, every 2-cocycle f may be uniquely expressed as a binary $(c+s+t)$-tuple $(f_1, \ldots, f_{c+s+t})_{\mathcal{B}}$ such that

$$f = \delta_1^{f_1} \bullet \ldots \bullet \delta_c^{f} \;\bullet \beta_1^{f\;\;+1} \bullet \ldots \bullet \beta_s^{f\;\;+} \;\;\bullet \gamma_1^{f\;\;+\;\;+1} \bullet \ldots \bullet \gamma_t^{f}$$

A genetic algorithm for finding Hadamard cocyclic matrices may be designed as follows.

The population consists of the whole space of *normalized* cocyclic matrices over G, $M_f = (f(g_i, g_j))$, f being a 2-cocycle. The term “normalized” means that the first row is formed all by 1. Each of the individuals f of the population (i.e. potential solutions to the problem) is identified to a binary $(c+s+t)$-tuple $(f_1, \ldots, f_{c+s+t})_{\mathcal{B}}$, the coordinates of the 2-cocycle f with regards to the basis $\mathcal{B}$. This way, the coordinates f_k are the genes of the individual $f = (f_1, \ldots, f_{c+s+t})_{\mathcal{B}}$.

The initial population P_0 is formed by some binary $(c+s+t)$-tuples randomly generated. Assuming that $|G| = 4t$ (remember that only 2×2 Hadamard matrices exist whose sizes are not multiple of 4), we consider $4t$ individuals for instance. Special care must be taken in generating the population, so that the population space does not grow exponentially with the order of the group G.

The population is expected to evolve generation through generation until an optimum individual (i.e. a Hadamard cocyclic matrix) is located. We now describe how to form a new generation P_{i+1} from an old one P_i:

1. Firstly, we must evaluate the fitness of every individual (i.e. 2-cocycle f) of P_i. It is common-knowledge that a computationally cheap test exists [7] to check if f gives rise to a Hadamard cocyclic matrix M_f. Concretely, it suffices to check whether the sum of every row in M_f but the first is zero. Define a *Hadamard row* to be a row whose summation is zero.

   From the property above an evaluation function for individuals is derived immediately: the fitness of M_f grows with the number of its Hadamard rows. Thus, the more fit an individual is, the more Hadamard rows it possess, and vice versa. The optimum is reached when all the rows but the first (i.e. rows from 2 to $4t$) are Hadamard rows. That is, whenever M_f reveals to be a Hadamard cocyclic matrix itself.

2. Once the evaluation is finished, the crossover comes into play. All individuals are paired at random, so that crossover combines the features of two parent chromosomes to form two similar offspring by swapping corresponding segments of the parents. Each time, the break point n is chosen at random, so that two couples of different parents are swapped with possibly different break points.

$$\left.\begin{array}{ll} j^{th}\text{-individual} & \overbrace{\bullet\bullet\bullet\cdots\bullet\bullet\bullet}^{n}\overbrace{\bullet\cdots\bullet}^{4t-n} \\ (j+1)^{th}\text{-individual} & \overbrace{\circ\circ\circ\cdots\circ\circ\circ}^{n}\overbrace{\circ\cdots\circ}^{4t-n} \end{array}\right\} \Rightarrow$$

$$\underset{\Rightarrow}{\text{children}} \left\{\begin{array}{l} \overbrace{\bullet\bullet\bullet\cdots\bullet\bullet\bullet}^{n}\overbrace{\circ\cdots\circ}^{4t-n} \\ \overbrace{\circ\circ\circ\cdots\circ\circ\circ}^{n}\overbrace{\bullet\cdots\bullet}^{4t-n} \end{array}\right.$$

3. Next we apply the mutation operator. Mutation arbitrarily alters just one gene of a selected individual (i.e. just one coordinate of the corresponding $(c+s+t)$-tuple, swapping 0 to 1 or 1 to 0, as it is the case), by a random change with a probability equal to the mutation rate (for instance, 1%).
4. Now individuals strive for survival: a selection scheme, biased towards fitter individuals (according to the number of their hadamard rows), selects the next generation. In case that an optimum individual exists, the algorithm stops. Otherwise the population P_{i+1} is constructed from a selection of $4t$ of the fittest individuals, in the following sense. Assume that n_k indicates the number of individuals in P_i which consists of exactly k Hadamard rows. Furthermore, assume that the fittest individuals in P_i consist of precisely r Hadamard rows (so that every individual in P_i possess at most r Hadamard rows, possibly less). The selection scheme firstly selects the n_r individuals with r Hadamard rows. If $n_r < 4t$, then all n_{r-1} individuals with exactly $r-1$ Hadamard rows are selected. And so on. This process continues until at least $4t$ individuals have been selected. Eventually, if the number of selected individuals exceeds from $4t$, some of the last individuals to be incorporated must be randomly deleted, in order to keep exactly $4t$ individuals in generation P_{i+1}.

The process goes on generation through generation until an optimum is reached. In spite of its simplicity, the method has surprisingly shown to work over several groups, though the number of required generations grows significantly with the size of the matrices. We next discuss the case of dihedral groups, where some significant improvements are introduced.

5 Genetic Algorithm on Dihedral Groups

Denote by D_{4t} the dihedral group $\mathbb{Z}_{2t} \times_\chi \mathbb{Z}_2$ of order $4t$, $t \geq 1$, given by the presentation

$$< a, b | a^{2t} = b^2 = (ab)^2 = 1 >$$

and ordering

$$\{1=(0,0), a=(1,0), \ldots, a^{2t-1}=(2t-1,0), b=(0,1), \ldots, a^{2t-1}b=(2t-1,1)\}$$

In [9] a representative 2-cocycle f of $[f] \in H^2(D_{4t}, \mathbb{Z}_2) \cong \mathbb{Z}_2^3$ is written interchangeably as a triple (A,B,K), where A and B are the inflation variables and K is the transgression variable. All variables take values ±1. Explicitly,

$$f(a^i, a^j b^k) = \begin{cases} A^{ij}, & i+j < 2t, \\ A^{ij}K, & i+j \geq 2t, \end{cases}$$

$$f(a^i b, a^j b^k) = \begin{cases} A^{ij}B^k, & i \geq j, \\ A^{ij}B^k K, & i < j, \end{cases}$$

Let β_1, β_2 and γ denote the representative 2-cocycles related to $(A,B,K) = (1,-1,1), (-1,1,1), (1,1,-1)$ respectively.

A basis for 2-coboundaries is described in [3]. Let $\partial_x : D_{4t} \rightarrow \mathbb{F}_2$ denote the characteristic set map associated to x, such that $\partial_x(y) = 1$ for $y \neq x$ and $\partial_x(x) = -1$. Let δ_x denote the 2-coboundary naturally associated to ∂_x, such that $\delta_x(s,t) = \partial_x(s)\partial_x(t)\partial_x(s \cdot t)$. According to the ordering above, a basis for 2-coboundaries may be constructed straightforwardly. It suffices to drop coboundaries $\delta_1, \delta_{a^2\ -2b}, \delta_{a^2\ -1b}$ from the whole set of coboundaries naturally associated to the elements in D_{4t}, as it is shown in [3]. Consequently, there are 2^{4t-3} different 2-coboundaries. Furthermore, there are 2^{4t} different 2-cocycles, and $\mathcal{B} = \{\delta_a, \ldots, \delta_{a^2\ -3b}, \beta_1, \beta_2, \gamma\}$ is a basis for 2-cocycles.

Once a basis for 2-cocycles over D_{4t} has been determined, we turn towards cocyclic Hadamard matrices.

A condition for the existence of a cocyclic Hadamard matrix over D_{4t} is detailed in [9]. Cocyclic Hadamard matrices developed over D_{4t} can exist only in the cases $(A,B,K) = (1,1,1), (1,-1,1), (1,-1,-1), (-1,1,1)$ for t odd. We focus in the case $(A,B,K) = (1,-1,-1)$, since computational results in [9, 3] suggest that this case contains a large density of cocyclic Hadamard matrices. Anyway, the techniques presented in this paper can be adapted easily for other cases of (A,B,K), or even other finite groups rather than D_{4t}, as the examples in Section 6 illustrate.

At this time, we may assume that individuals of our population consists of binary $(4t-3)$-tuples (better than $4t$-tuples), corresponding to generators from the basis for 2-coboundaries. Furthermore, computational results in [3] suggest that tuples formed from $2t-1$ to $2t+1$ ones gives rise to a significantly large density of cocyclic Hadamard matrices.

So that we may assume that individuals of our initial population consists of tuples that meet these bounds. That is, tuples of length $4t-3$ which consists of k ones and $4t-3-k$ zeros, for $2t-1 \leq k \leq 2t+1$. Consequently, the search space reduces in turn, from $2^{4t-3} = \sum_{i=0}^{4t-3} \binom{4t-3}{i}$ individuals to precisely $\binom{4t-3}{2t-1} + \binom{4t-3}{2t} + \binom{4t-3}{2t+1}$ individuals. We do not care about missing many potential solutions (from among those tuples which do not meet the impossed bounds). Computational evidences of this fact are discussed in [3]. Anyway, crossover and mutation operators will eventually introduce individuals out from the impossed bounds, which will also strive for survival, so that good behaviuors outside the reduced search space may be ocassionally incorporated.

In these circumstances, the evaluation function for fitness may be redesigned to check precisely rows from 2 to t. This is a straightforward consequence of a result in [3]: a cocyclic matrix over D_{4t} of the type $\left(\prod_{i\in I}\delta_i\right)\beta_1\gamma$ is Hadamard if and only if rows from 2 to t are Hadamard rows. Consequently, the Hadamard test runs 4 times faster each time. This way, when the genetic algorithm runs on D_{4t} we are able to reduce not only the search space but also the search time.

6 Examples

Both of the genetic algorithms and all the executions and examples of this section have been worked out with aid of Mathematica 4.0, running on a *Pentium IV 2.400 Mhz DIMM DDR266 512 MB*. We include here some Hadamard cocyclic matrices of order $4t$ for $6 \leq t \leq 13$. Apparently, our method seems to provide some cocyclic Hadamard matrices of larger order than those previously obtained with other algorithms.

Calculations in [5, 9, 2] suggest that $G_1^t = \mathbb{Z}_t \times \mathbb{Z}_2^2$ and $G_2^t = D_{4t}$ give rise to a large number of Hadamard cocyclic matrices.

This behavior has also been observed on a third family of groups [2],

$$G_3^t = (\mathbb{Z}_t \times_f \mathbb{Z}_2) \times_\chi \mathbb{Z}_2$$

Here f denotes the normalized 2-cocycle $f : \mathbb{Z}_2 \times \mathbb{Z}_2 \to \mathbb{Z}_t$ such that

$$f(-1,-1) = \lceil \frac{t}{2} \rceil + 1$$

And $\chi : \mathbb{Z}_2 \times (\mathbb{Z}_t \times_f \mathbb{Z}_2) \to \mathbb{Z}_t \times_f \mathbb{Z}_2$ denotes the dihedral action, such that $\chi(-1,x) = -x$. Notice that G_3^t is a slight modification of $G_2^t = D_{4t}$, since f becomes a 2-coboundary precisely for odd $t = 2k+1$. Thus $G_3^{2k+1} = G_2^{2k+1}$ and $G_3^{2k} \neq G_2^{2k}$.

However the search space for cocyclic Hadamard matrices over the families G_i^t above grows exponentially with t (according to the dimensions of the basis $\mathcal{B}_i$ for 2-cocycles), so that exhaustive search is only possible in low orders (up to $t = 5$). Each of the matrices is represented as a tuple with regards to some basis $\mathcal{B}_i = \{\delta_k|\beta_j|\gamma_n\}$ for 2-cocycles over G_i^t. At this point, we only indicate how many generators of each type (coboundaries, inflation and transgression) appear in $\mathcal{B}_i$ (see [2, 3] for details):

– For odd t, $\mathcal{B}_1$ consists of $4t-3$ coboundaries δ_k, 2 cocycles β_j coming from inflation and 1 cocycle γ coming from transgression. For even t, $\mathcal{B}_1$ consists of $4t-4$ coboundaries δ_k, 3 cocycles β_j coming from inflation and 3 cocycles γ_n coming from transgression.

– $\mathcal{B}_2$ is the basis $\mathcal{B}$ described at Section 5, which consists of $4t-3$ coboundaries δ_k, 2 cocycles β_j coming from inflation and 1 cocycle γ coming from transgression.

– $\mathcal{B}_3$ coincides with $\mathcal{B}_2$ for odd t. We have not identified a general behavior for even t, so we analyze the cases $t = 2, 4, 6, 8$ independently:
  - If $t = 2$, $\mathcal{B}_3$ consists of 4 coboundaries δ_k, 3 cocycles β_j coming from inflation and 3 cocycles γ_n coming from transgression.
  - If $t = 4$, $\mathcal{B}_3$ consists of 13 coboundaries δ_k, 2 cocycles β_j coming from inflation and 1 cocycle γ coming from transgression.
  - If $t = 6$, $\mathcal{B}_3$ consists of 20 coboundaries δ_k, 3 cocycles β_j coming from inflation and 3 cocycles γ_n coming from transgression.
  - If $t = 8$, $\mathcal{B}_3$ consists of 29 coboundaries δ_k, 2 cocycles β_j coming from inflation and 1 cocycle γ coming from transgression.

Now we show some executions of the genetic algorithm (in its general version) running on these families. The tables below show some Hadamard cocyclic matrices over G_i^t, and the number of iterations and time required (in seconds) as well. Notice that the number of generations is not directly related to the size of the matrices. Do not forget about randomness of the genetic algorithm.

t	iter.	time	product of generators of 2-cocycles over G_1^t
1	0	$0''$	$(1,0,0,0)$
2	0	$0''$	$(1,0,0,1,0,1,0,0,1,1)$
3	1	$0.14''$	$(0,1,1,0,1,0,0,0,1,1,1,1)$
4	7	$1.89''$	$(0,1,1,0,0,1,1,1,1,1,1,0,1,1,1,0,0,1)$
5	30	$17.08''$	$(1,0,1,1,0,0,0,1,0,1,1,1,0,1,1,1,0,1,1,1)$
6	3	$3.69''$	$(0,0,0,1,0,0,0,1,1,1,0,0,1,1,0,1,1,1,0,1,1,0,1,0,0,1)$
7	584	$21'33''$	$(1,0,1,1,0,1,1,0,1,0,1,0,0,0,0,0,1,1,0,0,1,1,0,0,0,1,1,1)$
8	239	$14'33''$	$(0,0,0,1,0,0,0,0,1,1,1,1,0,1,0,1,0,1,0,0,0,1,0,1,1,0,1,0,0,1,0,0,0,1)$

t	iter.	time	product of generators of 2-cocycles over G_2^t
1	0	$0''$	$(0,1,1,1)$
2	0	$0''$	$(1,0,1,1,1,1,0,0)$
3	3	$0.25''$	$(1,0,1,1,0,0,1,0,0,1,0,1)$
4	0	$0''$	$(0,1,1,1,1,1,1,0,0,0,1,0,1,1,0,1)$
5	3	$1.42''$	$(0,0,1,0,1,1,0,1,0,1,0,0,0,1,0,1,1,1,0,1)$
6	31	$34.87''$	$(1,1,0,1,1,0,1,0,0,0,0,0,0,1,0,1,0,0,1,1,0,1,0,1)$
7	102	$5'17''$	$(1,0,0,1,0,1,1,0,0,1,1,1,0,1,0,0,0,0,0,0,0,0,1,1,0,1,0,1)$
8	98	$6'27''$	$(0,0,1,0,1,1,0,1,1,0,1,1,1,0,1,1,0,1,1,1,1,0,0,0,0,0,0,1,0,1,0,1)$

t	iter.	time	product of generators of 2-cocycles over G_3^t
1	0	$0''$	$(0, 1, 1, 1)$
2	0	$0''$	$(1, 1, 0, 0, 0, 0, 1, 0, 0, 0)$
3	0	$0''$	$(0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1)$
4	6	$1.20''$	$(0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0)$
5	18	$10.33''$	$(1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0)$
6	15	$19.49''$	$(1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1)$
7	6	$12.39''$	$(0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1)$
8	153	$9'45''$	$(1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0)$

As it is expected, the improved version of the genetic algorithm for $G_2^t = D_{4t}$ provides not only faster outputs but also larger sizes on the matrices.

t	iter.	time	product of generators of 2-cocycles over D_{4t} (improved version)
6	0	$0''$	$(1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1)$
7	4	$0.69''$	$(0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1)$
8	3	$1.18''$	$(0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1)$
9	7	$5.09''$	$(1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1)$
10	43	$48.03''$	$(0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1)$
11	471	$13'15''$	$(1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1)$
12	279	$11'16''$	$(0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1)$
13	970	$53'44''$	$(0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1)$

The authors are convinced that improved versions of the algorithm are still to be implemented, attending to refinements on the crossover operator. We are yet to find a systematic way of doing crossover more suitably for our purposes.

Acknowledgments

The authors want to express their gratitude to the referees for their many valuable advices and suggestions, which have led to a number of improvements of the paper.

References

1. V. Álvarez, J.A. Armario, M.D. Frau and P. Real. An algorithm for computing cocyclic matrices developed over some semidirect products. *Proceedings AAECC'14*. Eds. S. Boztas, I.E. Shparlinski. *Springer Lecture Notes in Computer Science*, Springer-Verlag, Heidelberg, (2001).
2. V. Álvarez, J.A. Armario, M.D. Frau and P. Real. Homological reduction method for constructing cocyclic Hadamard matrices. To appear.

3. V. Álvarez, J.A. Armario, M.D. Frau and P. Real. How to bound the search space looking for cocyclic Hadamard matrices. To appear.
4. A. Baliga and J. Chua. Self-dual codes using image resoration techniques. *Proceedings AAECC'14*. Eds. S. Boztas, I.E. Shparlinski. *Springer Lecture Notes in Computer Science*, Springer-Verlag, Heidelberg, (2001).
5. A. Baliga and K.J. Horadam. Cocyclic Hadamard matrices over $\mathbb{Z}_t \times \mathbb{Z}_2^2$. *Australas. J. Combin.*, **11**, 123–134, (1995).
6. W. de Launey and K.J. Horadam. Cocyclic development of designs. *J. Algebraic Combin.*, **2** (3), 267–290, 1993. Erratum: *J. Algebraic Combin.*, (1), pp. 129, 1994.
7. W. de Launey and K.J. Horadam. Generation of cocyclic Hadamard matrices. *Computational algebra and number theory* (Sydney, 1992), volume **325** of *Math. Appl.*, 279–290. Kluwer Acad. Publ., Dordrecht, (1995).
8. D.L. Flannery. Calculation of cocyclic matrices. *J. of Pure and Applied Algebra*, **112**, 181–190, (1996).
9. D.L. Flannery. Cocyclic Hadamard matrices and Hadamard groups are equivalent. *J. Algebra*, **192**, 749–779, (1997).
10. D.L. Flannery and E.A. O'Brien. Computing 2-cocycles for central extensions and relative difference sets. *Comm. Algebra*, **28(4)**, 1939–1955, (2000).
11. J. Grabmeier, L.A. Lambe. Computing Resolutions Over Finite p-Groups. *Proceedings ALCOMA'99*. Eds. A. Betten, A. Kohnert, R. Lave, A. Wassermann. *Springer Lecture Notes in Computational Science and Engineering*, Springer-Verlag, Heidelberg, (2000).
12. V.K.A.M. Gugenheim and L.A. Lambe. Perturbation theory in Differential Homological Algebra, I. *Illinois J. Math.,* **33**, 556–582, (1989).
13. V.K.A.M. Gugenheim, L.A. Lambe and J.D. Stasheff. Perturbation theory in Differential Homological Algebra II, *Illinois J. Math.,* **35** (3), 357–373, (1991).
14. J.H. Holland. Adaptation in natural and artificial systems. *University of Michigan Press*, Ann Arbor (1975).
15. Z. Michalewicz. Genetic algorithms + data structures = evolution programs. *Springer-Verlag* (1992).
16. P. Real. Homological Perturbation Theory and Associativity. *Homology, Homotopy and Applications*, **2**, 51–88, (2000).

Unconditionally Secure Chaffing-and-Winnowing: A Relationship Between Encryption and Authentication

Goichiro Hanaoka[1], Yumiko Hanaoka[2], Manabu Hagiwara[1],
Hajime Watanabe[1], and Hideki Imai[1,3]

[1] Research Center for Information Security,
National Institute of Advanced Industrial Science and Technology,
1102 Akihabara Daibiru, 1-18-13 Sotokanda, Chiyoda-ku, Tokyo 101-0021, Japan
{hanaoka-goichiro, hagiwara.hagiwara, h-watanabe}@aist.go.jp
[2] NTT DoCoMo, Inc., 3-5 Hikarino-oka, Yokosuka 239-8536, Japan
yamamotoyumi@nttdocomo.co.jp
[3] Institute of Industrial Science, University of Tokyo,
4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, Japan
imai@iis.u-tokyo.ac.jp

Abstract. A chaffing-and-winnowing is a cryptographic scheme which does not require encryption but instead use a message authentication code (MAC) to provide the same function as encryption. In this paper, we discuss and introduce some new insights in the relationship between unconditionally secure authentication codes (A-code) and unconditionally secure encryption schemes through observing the mechanisms of chaffing-and-winnowing. Particularly, we show through chaffing-and-winnowing that an A-code with a security level considerably low stands equivalently for an encryption scheme with perfect secrecy, and a fully secure authentication scheme implies both perfect secrecy and non-malleability for an encryption scheme in the unconditionally secure setting.

1 Introduction

1.1 Background and Motivation

In 1998, Rivest proposed a novel and interesting cryptographic technique called "*chaffing-and-winnowing*" [15]. Remarkable property of this cryptographic technique is that it can provide data confidentiality by using authentication when sending data over an insecure channel. In other words, chaffing-and-winnowing is *an encryption scheme without encryption*. As Rivest also made a point that, as chaffing-and-winnowing is not categorized as an encryption, this may be one solution to getting a way around existing encryption legislation. Realistically, though, the efficiency of chaffing-and-winnowing still remains to be a problem, and if we leave out the merit of being able to bypass the law enforcement,

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 154–162, 2006.

chaffing-and-winnowing, for this moment, does not measure up to what conventional encryption schemes offer especially regarding its practicality. However, to the extent that encryption export regulations are going to be a long-term issue, chaffing-and-winnowing is still marked as an interesting development methodology which uses authentication mechanisms to send confidential data, and if you dig down a bit more, interesting relations between encryption and authentication through chaffing-and-winnowing can be found.

In our paper, we discuss and introduce some new insights in the relationship between an authentication scheme and an encryption scheme through observing the mechanism of chaffing-and-winnowing. Particularly, we consider the case in which the security is guaranteed for unbounded computational resources (i.e. unconditionally-secure setting). Namely, we show through chaffing-and-winnowing, in an unconditionally secure setting, an authentication scheme with a security level considerably low, in fact, stands equivalently for an encryption scheme with perfect secrecy, and a fully secure authentication scheme implies security of a much higher level (i.e. satisfies both perfect secrecy and non-malleability). In addition, we compare the constructions of encryption and authentication schemes and show some interesting relationships between the two schemes and see how closely the security for an impersonation attack for authentication, and perfect secrecy for encryption are related to each other. We follow then by comparing the relationship between the security for substitution attack for authentication and non-malleability for encryption as well.

1.2 Related Works

Bellare and Boldyreva analyzed the security for chaffing-and-winnowing from a computationally-secure context [1]. They have shown in their paper that, with a message authentication code (MAC) based on pseudo-random function (PRF) as an authentication scheme, combining an appropriately chosen *all-or-nothing transform* (AONT) [14] (e.g., OAEP [4, 5]) gives a semantically secure encryption scheme [9]. It should be noticed that any PRF is a good MAC [8, 3] and not vice versa. In contrast, in our paper, we look into the relation from an information-theoretically perspective, and show that with an authentication code (A-code) [7, 18] (which corresponds to MAC in the computationally-secure setting), it gives an encryption scheme with perfect secrecy even for a *weak* A-code. If we have a stronger (fully-secure) A-code, then in an information-theoretically-secure setting, an even stronger encryption scheme (i.e. with perfect secrecy and non-malleability) can be constructed. There has been a proposal of combining an A-code with information-theoretically secure encryption called *A-code with secrecy* [7, 17], and also, the secrecy property of A-code has been discussed in [18, 19], however, it is novel and never been discussed before through observing the mechanism of chaffing-and-winnowing to show the correspondence relationship of the security requirement of an A-code and an encryption.

2 Preliminaries

2.1 Unconditionally Secure Authentication Code (A-code)

In an A-code [7, 18], there are three participants, a sender S, a receiver R and a trusted initializer TI. TI generates secret information u and v for R and S, respectively. In order to send a plaintext m to R, S generates an authenticated message (m, α) from m by using u and transmits (m, α) to R. R verifies the validity of α using m and v. We note that S and/or R may generate u and v themselves to remove TI.

Definition 1. Let $\mathcal{U}, \mathcal{V}, \mathcal{M}$ and $\mathcal{A}$ denote the random variables induced by u, v, m and α, respectively. We say that $(\mathcal{U}, \mathcal{V}, \mathcal{M}, \mathcal{A})$ is *p-impersonation secure (p-*Imp*)* if

1. Any outsiders (which do not include S, R or TI) can perform impersonation with probability at most p. Namely,

$$\max_{(m,\alpha)} \Pr[R \text{ accepts } (m, \alpha)] \leq p,$$

it is also *p-substitution secure (p-*Sub*)* if

2. Any set of outsiders can perform substitution with probability at most p. Namely, letting (m', α') be an authenticated message generated by S,

$$\max_{(m',\alpha')} \max_{(m,\alpha)(\neq(m',\alpha'))} \Pr[R \text{ accepts } (m, \alpha)|(m', \alpha')] \leq p.$$

We say that $(\mathcal{U}, \mathcal{V}, \mathcal{M}, \mathcal{A})$ is *p-impersonation&substitution secure (p-*Imp&Sub*)* if it is both p-Imp and p-Sub secure.

Construction methods for A-codes are given in, for example, [7, 18, 10, 16]. In the rest of the paper, for simplicity, we let $f : \mathcal{M} \times \mathcal{U} \to \mathcal{A}$ denote a mapping such that $f(m, u) = \alpha$.

2.2 Unconditionally Secure Encryption

In the model of unconditionally secure encryption, there are three participants, a senders S, a receiver R and a trusted initializer TI. TI generates an encryption key e for S, and a decryption key d for R. TI, after distributing these keys, deletes them from his memory. To send a plaintext m to R with confidentiality, S encrypts m by using e and transmits the ciphertext c to R. R decrypts c by using d and recovers m. We note that in order to remove TI, S and/or R may generate e and d themselves. Throughout this paper, we let a random variable be $\mathcal{X}$ and $H(\mathcal{X})$ denote the entropy of $\mathcal{X}$. For $\mathcal{X}$, let $X := \{x | \Pr[\mathcal{X} = x] > 0\}$. $|X|$ is the cardinality of X.

Definition 2. Let $\mathcal{E}$, $\mathcal{D}$, $\mathcal{M}$ and $\mathcal{C}$ denote the random variables induced by e, d, m and c, respectively. We say that $(\mathcal{E}, \mathcal{D}, \mathcal{M}, \mathcal{C})$ has *perfect secrecy* (PS) if

1. R correctly decrypts m from c, that is, $H(\mathcal{M}|\mathcal{C},\mathcal{D}) = 0$.
2. No outsider obtain any information on m from c, that is, $H(\mathcal{M}|\mathcal{C}) = H(\mathcal{M})$.

It will also satisfy *perfect non-malleablity* (NM) if

3. No outsider generate a ciphertext whose plaintext is meaningfully related to m, that is,

$$H(\hat{\mathcal{M}}|\mathcal{C},\hat{\mathcal{C}},\mathcal{M}) = H(\hat{\mathcal{M}}|\mathcal{C},\mathcal{M}),$$

where $\hat{c}(\neq c)$ be another ciphertext which can be generated by S instead of c, $\hat{m}(\neq m)$ be a plaintext corresponding $\hat{c}$, and $\hat{\mathcal{C}}$ and $\hat{\mathcal{M}}$ denote random variables induced by $\hat{c}$ and $\hat{m}$, respectively.

Then, we say that $(\mathcal{E},\mathcal{D},\mathcal{M},\mathcal{C})$ has PS&NM if it satisfies both PS and NM.

The notion of "non-malleability" is a concept proposed by Dolev, Dwork and Naor [6]. The discussion that followed after their original proposal was mainly given from a computationally secure perspective [2]. The first formalization of an information-theoretically secure scheme with non-malleability was given recently in [11] by Hanaoka, Shikata, Hanaoka and Imai, and the idea was then extended by McAven, Safavi-Naini and Yung [13]. It is obvious that a classical one-time pad does not provide perfect non-malleability.

2.3 Chaffing-and-Winnowing

In brief, chaffing-and-winnowing can be constructed as follows. Start by each sender S and receiver R prepare themselves each a key for message authentication. When S sends a plaintext m to R, S adds "dummy" plaintext m' (with an invalid authentication tag) so that "dummy" m' obscure the intended message m, so that only the authorized receiver R can distinguish the "real" from the "dummy". On receiving the message, R removes the dummy m' by checking its tag. As long as an adversary do not distinguish a valid tag from the invalid tag, adversary cannot tell which one of m and m' is real and not real. Chaffing-and-winnowing is a technique which consists of adding dummy messages to a message, so that it becomes unintelligible for anyone to distinguish the message except for the authorized receiver. Chaffing-and-winnowing is not an encryption and is not a technique which tries to hide the plaintext itself (like encryption).

3 Unconditionally Secure Chaffing-and-Winnowing

Here, we show the construction of an unconditionally secure chaffing-and-winnowing (USCW) which is secure against any adversary with unlimited computational power by using an A-code. Take notice that straightforward construction (i.e. if the dummy is not generated appropriately) will not be secure and the information on the plaintext will be more likely to leak. We give careful consideration on this point when we construct our scheme. We first show that we can construct an USCW with PS from a p-Imp A-code.

Unconditionally Secure Chaffing-and-Winnowing with PS

Key Generation. For a given A-code $(\mathcal{U}, \mathcal{V}, \mathcal{M}, \mathcal{A})$, TI generates $u \in U$ and $v \in V$ as an encryption key and a decryption key, respectively. Let the plaintext space be M. TI gives u and v to S and R, respectively.

Encryption. Let a plaintext be $m^* \in M$. S sets $\alpha_{m^*} := f(m^*, u)$. S picks $|M|$–1 distinct keys $(u'_m)_{m \in M \backslash \{m^*\}}$ from $U \backslash \{u\}$ and sends $c := (m||\alpha_m)_{m \in M}$ to R, where $\alpha_m := f(m, u'_m)$ for $m \in M \backslash \{m^*\}$.

Decryption. On receiving c', R parses c' as $c' := (m||\alpha_m)_{m \in M}$ and selects m' such that m' is accpeted as valid (by using v). Finally, R outputs m'.

Next, we show that p-Imp A-codes imply USCW with PS. For simplicity, we consider only an *optimal* p-Imp A-code such that $p = 1/|A| = 1/|U|$. It should be noticed that if an A-code is p-Imp, then it is $|A| \geq 1/p$ and $|U| \geq 1/p$ [12]. Many of such optimal A-codes have been known so far.

Theorem 1. *Suppose* $(\mathcal{U}, \mathcal{V}, \mathcal{M}, \mathcal{A})$ *is* $1/|M|$-Imp *A-code. Then, the above USCW is* PS.

Proof. Due to symmetric property of u and $(u'_m)_{m \in M \backslash \{m^*\}}$, it is sufficient to prove that for all $m \in M \backslash \{m^*\}$, R rejects (m, α_m) as invalid (by v).

For each $m \in M$, there exists only one $\alpha \in A$ which will be accepted by R since $p = 1/|A|$. This means that it is sufficient to prove that $f(m, u) \neq f(m, u')$ for all $u, u'(\neq u) \in U$ and $m \in M$. We prove this by contradition.

Assume that there exist u, u' and m such that $f(m, u) = f(m, u')(= \alpha)$. Then, an adversary can launch an impersonation attack by sending (m, α). In this case, R accepts (m, α) as valid if S's key is either u or u'. Hence, the attack succeeds with probability $2/|U| = 2p$. This is a contradiction. □

We next show p-$\mathsf{Imp\&Sub}$ A-codes imply USCW with $\mathsf{PS\&NM}$. For simplicity, we consider only an *optimal* p-$\mathsf{Imp\&Sub}$ A-code such that $p = 1/|A| = 1/|U|^{1/2}$. It should be noticed that if an A-code is p-$\mathsf{Imp\&Sub}$, then $|A| \geq 1/p$ and $|U| \geq 1/p^2$ [12]. Many of such optimal A-codes have been known.

Unconditionally Secure Chaffing-and-Winnowing with $\mathsf{PS\&NM}$

Key Generation. For a given A-code $(\mathcal{U}, \mathcal{V}, \mathcal{M}, \mathcal{A})$, TI generates $u \in U$ and $v \in V$ as an encryption key and a decryption key, respectively. Let the plaintext space be M. TI gives u and v to S and R, respectively. S picks $|M|$ distinct keys $u_1, ..., u_{|M|}$ from $U \backslash \{u\}$ such that

$$\forall u_i, u_j (\neq u_i),\ \forall m \in M,\ f(m, u_i) \neq f(m, u_j).$$

ENCRYPTION. Let a plaintext be $m^* \in M$. S sets $\alpha := f(m^*, u)$ and finds u_i such that $f(m^*, u_i) = \alpha$. Then, S sends $c := (m||\alpha_m)_{m \in M}$ to R, where $\alpha_m := f(m, u_i)$.

DECRYPTION. On receiving c', R parses c' as $c' := (m||\alpha_m)_{m \in M}$ and selects m' such that m' is accpeted as valid (by using v). Finally, R outputs m'.

Before we begin the security proof, we first show that the above $u_1, ..., u_{|M|}$ in fact always exist if the given A-code is $1/|M|$-Imp&Sub.

Lemma 1. *If $(\mathcal{U}, \mathcal{V}, \mathcal{M}, \mathcal{A})$ is $1/|M|$-Imp&Sub, then, for all $u \in U$ there exist $u_1, ..., u_{|M|} \in U \backslash \{u\}$ such that for all $u_i, u_j (\neq u_i) \in \{u_1, ..., u_{|M|}\}$ and $m \in M$, $f(m, u_i) \neq f(m, u_j)$.*

Proof. Here, we show how $\{u_1, ..., u_{|M|}\}$ is chosen for any given u. First, pick u_1 from $U \backslash \{u\}$ randomly, and set $U_{1,m} := \{u | f(m, u_1) = f(m, u)\}$ for all $m \in M$. Since the given A-code is $1/|M|$-Imp&Sub, it is clear that $|U_{1,m}| \leq |M|$. This implies that

$$|U \backslash \cup_{m \in M} U_{1,m}| \geq |M| - 1.$$

Next, pick distinct $u_2, ..., u_{|M|} \in U \backslash \cup_{m \in M} U_{1,m}$, and set $U_{i,m} := \{u | f(m, u_i) = f(m, u)\}$ for $i = 2, ..., |M|$ and $m \in M$. Assume that there exist $u_{i_0}, u_{i_1} (\neq u_{i_0}) \in \{u_2, ..., u_{|M|}\}$ such that $f(m_0, u_{i_0}) = f(m_0, u_{i_1})$ for some $m_0 \in M$, i.e. $U_{i_0,m_0} = U_{i_1,m_0}$. This implies that $| \cup_{2 \leq i \leq |M|} U_{i,m_0}| \leq (|M| - 2)|M|$. On the other hand, it is obvious that $U = \cup_{\alpha \in A} \{u | f(u, m_0) = \alpha\}$, and consequently, we have

$$|U| = | \cup_{\alpha \in A} \{u | f(u, m_0) = \alpha\}| = | \cup_{1 \leq i \leq |M|} U_{i,m_0}| \leq (|M| - 1)|M|.$$

This is a contradiction since $|U| = |M|^2$. Hence, for all $i_0, i_1 (\neq i_0)$ and m, $f(m_0, u_{i_0}) \neq f(m_0, u_{i_1})$. □

Next, we prove for such $u_1, ..., u_{|M|}$ and any $m \in M$, only one u_i exists, such that $f(m, u) = f(m, u_i)$.

Lemma 2. *For any $u \in U$, any $u_1, ..., u_{|M|}$ chosen as in above, and any $m \in M$, $|\{u_i | f(m, u_i) = f(m, u),\ u_i \in \{u_1, ..., u_{|M|}\}\}| = 1$.*

Proof. Assume this lemma is false. Then, there exist $u_{i_0}, u_{i_1} \in \{u_1, ..., u_{|M|}\}$ such that $f(m, u_{i_0}) = f(m, u_{i_1})$ which is a contradiction. □

Lemma 1 and 2 guarantee that the proposed USCW will always work properly for any u and m.

Finally, we prove the security.

Lemma 3. *The proposed USCW has* PS, *i.e. $H(\mathcal{M}^*) = H(\mathcal{M}^* | \mathcal{C})$, where $\mathcal{M}^*$ is a random variable induced by m^*.*

Proof. Let $\mathcal{U}_i$ denote a random variable induced by u_i such that $f(m^*, u) = f(m^*, u_i)$. Then, it is clear that $H(\mathcal{C}|\mathcal{U}_i) = 0$. Consequently, we have that

$$H(\mathcal{M}^*|\mathcal{U}_i) - H(\mathcal{M}^*|\mathcal{C},\mathcal{U}_i) = H(\mathcal{M}^*|\mathcal{U}_i) - H(\mathcal{M}^*|\mathcal{U}_i) = 0.$$

If $H(\mathcal{M}^*|\mathcal{U}_i) = H(\mathcal{M}^*)$, then we have

$$\begin{aligned} H(\mathcal{M}^*) - H(\mathcal{M}^*|\mathcal{C}) &\leq H(\mathcal{M}^*) - H(\mathcal{M}^*|\mathcal{C},\mathcal{U}_i) \\ &= H(\mathcal{M}^*|\mathcal{U}_i) - H(\mathcal{M}^*|\mathcal{C},\mathcal{U}_i) \\ &= 0. \end{aligned}$$

This means that to prove the lemma, it will be sufficient to prove $H(\mathcal{M}^*|\mathcal{U}_i) = H(\mathcal{M}^*)$.

For a given u_i, we have only that $u \notin \{u_1, ..., u_{|M|}\}$ and $f(m^*, u) = f(m^*, u_i)$ for some m^*, and therefore, $u \in \{u|\exists m \in M,\ f(m, u) = f(m, u_i),\ u \in U\backslash\{u_i\}\} = \cup_{m\in M} U_{i,m}\backslash\{u_i\}$. Since $|U_{i,m}| = |M|$ for all $m \in M$,

$$\max_{m^*,u} \max_{m'} \Pr[m' = m^*|u_i] = \frac{|U_{i,m'}\backslash\{u_i\}|}{|\cup_{m\in M} U_{i,m}\backslash\{u_i\}|} = \frac{|M| - 1}{|M|(|M| - 1)} = \frac{1}{|M|}.$$

Hence, $H(\mathcal{M}^*|\mathcal{U}_i) = H(\mathcal{M}^*)$. □

Lemma 4. *The proposed USCW has* NM*, i.e.* $H(\hat{\mathcal{M}}|\mathcal{C}, \hat{\mathcal{C}}, \mathcal{M}^*) = H(\hat{\mathcal{M}}|\mathcal{C}, \mathcal{M}^*)$*, where* $\hat{c}(\neq c)$ *is another ciphertext which can be generated by* S *instead of* c*,* $\hat{m}(\neq m^*)$ *be a plaintext underlying* $\hat{c}$*, and* $\hat{\mathcal{C}}$ *and* $\hat{\mathcal{M}}$ *denote random variables induced by* $\hat{c}$ *and* $\hat{m}$*, respectively.*

Proof. From m^* and c, an adversary only knows that $u \in U_{i,m^*}\backslash\{u_i\}$. Now, we prove that all $\tilde{m} \in M\backslash\{m^*\}$ are equally possible even if it is known that $u \in U_{i,m^*}\backslash\{u_i\}$.

Let $\hat{c} := (m||f(m, u_j))_{m\in M}$, $j \neq i$, then $\tilde{m} \in M$ can be the plaintext only if $|U_{i,m^*} \cap U_{j,\tilde{m}}| \neq 0$.

Claim 1. *For any* $\tilde{m} \in M\backslash\{m^*\}$*,* $|U_{i,m^*} \cap U_{j,\tilde{m}}| = 1$*.*

Proof. Assume that $|U_{i,m^*} \cap U_{j,\tilde{m}}| = 0$. Then, for an authenticated message (m^*, α^*) where $\alpha^* = f(m^*, u_i)$, an adversary can launch a substitution attack by generating $(\tilde{m}, \tilde{\alpha})$ where $\tilde{\alpha}$ is randomly chosen from $A\backslash\{f(\tilde{m}, u_j)\}$. Notice that the adversary can generate $(\tilde{m}, \tilde{\alpha})$ if he knows only (m^*, α^*). It is clear that $f(\tilde{m}, u_j)$ is not the correct authentication tag for $\tilde{m}$, and consequently, we have

$$\Pr[R \text{ accepts } (\tilde{m}, \tilde{\alpha})|(m^*, \alpha^*)] \geq \frac{1}{|M| - 1}.$$

Since the given A-code is $1/|M|$-Imp&Sub, this is a contradiction.

Next, assume that $|U_{i,m^*} \cap U_{j,\tilde{m}}| \geq 2$. Then, for an authenticated message (m^*, α^*) where $\alpha^* = f(m^*, u_i)$, an adversary can launch a substitution attack by generating $(\tilde{m}, \tilde{\alpha})$ where $\tilde{\alpha} = f(\tilde{m}, u_j)$. It is clear that there exist at least two keys in U_{i,m^*} such that the correct authentication tag for $\tilde{m}$ is determined as $f(\tilde{m}, u_j)$, and consequently, we have

$$\Pr[R \text{ accepts } (\tilde{m}, \tilde{\alpha})|(m^*, \alpha^*)] \geq \frac{2}{|M|},$$

which is a contradiction. Hence, $|U_{i,m^*} \cap U_{j,\tilde{m}}| = 1$. □

Claim 2. *For any $\tilde{m}_0, \tilde{m}_1 (\neq \tilde{m}_0) \in M \backslash \{m^*\}$, $U_{j,\tilde{m}_0} \cap U_{j,\tilde{m}_1} = \{u_j\}$.*

Proof. It is obvious that $u_j \in U_{j,\tilde{m}_0} \cap U_{j,\tilde{m}_1}$, and hence, it is sufficient to prove that $|U_{j,\tilde{m}_0} \cap U_{j,\tilde{m}_1} \backslash \{u_j\}| = 0$. Assume that there exists $u \in U \backslash \{u_j\}$ such that $f(\tilde{m}_0, u) = f(\tilde{m}_0, u_j)$ and $f(\tilde{m}_1, u) = f(\tilde{m}_1, u_j)$. Then, for an authenticated message $(\tilde{m}_0, \tilde{\alpha}_0)$ where $\tilde{\alpha}_0 = f(\tilde{m}_0, u_j)$, an adversary can launch a substitution attack by generating $(\tilde{m}_1, \tilde{\alpha}_1)$ where $\tilde{\alpha}_1 = f(\tilde{m}_1, \tilde{u})$ and $\tilde{u}$ is randomly chosen from $U_{j,\tilde{m}_0}$. It is clear that there exist at least two keys in $U_{j,\tilde{m}_0}$ such that the correct authentication tag for $\tilde{m}_1$ is determined as $f(\tilde{m}_1, u_j)$, and consequently, we have

$$\Pr[R \text{ accepts } (\tilde{m}_1, \tilde{\alpha}_1)|(\tilde{m}_0, \tilde{\alpha}_0)] \geq \frac{2}{|M|},$$

which is a contradiction. □

From Claims 1 and 2, we have that

$$\begin{aligned}\max_{\hat{m}, m^*, u} \max_{\tilde{m}} \Pr[\tilde{m} = \hat{m}|m^*, u_i] &= \frac{|U_{j,\hat{m}} \backslash \{u_j\}|}{|\cup_{m \in M \backslash \{m^*\}} U_{j,m} \backslash \{u_j\}|} \\ &= \frac{|M| - 1}{(|M| - 1)(|M| - 1)} = \frac{1}{|M| - 1},\end{aligned}$$

which proves the lemma. □

4 An Observation

The above results show a new insight in the relationship between impersonation security of an A-code and perfect secrecy of an unconditionally-secure encryption, and also substitution security of an A-code and non-malleability of an unconditionally-secure encryption.

Here, in this section, we look at this from another angle as well. To give an actual example of an unconditionally secure encryption with perfect secrecy, we like to consider here, a classical one-time pad. Let $GF(q)$ be a finite field with q elements, $k \in GF(q)$ be a shared key between S and R, and $c := m + k$ where $m \in GF(q)$. Obviously, (m, c) can also be used as a $1/q$-Imp secure A-code. Next, similarly, we consider an example of an unconditionally secure encryption with perfect secrecy and non-malleability [11]. Let $(k_1, k_2) \in GF(p) \backslash \{0\} \times GF(p)$ be a shared key between S and R, and $c := k_1 m + k_2$. Hence, this can also be used as a $1/(q-1)$-$\mathsf{Imp}\&\mathsf{Sub}$ secure A-code. What these observations are telling is that, impersonation security and perfect security, as well as, substitution security and non-malleability, are deeply related to each other.

Acknowledgement

We would like to thank anonymous reviewers for their invaluable comments.

References

1. M. Bellare and A. Boldyreva, "The security of chaffing and winnowing," Proc. of Asiacrypt'00, LNCS 1976, Springer-Verlag, pp.517-530, 2000.
2. M. Bellare, A. Desai, D. Pointcheval and P. Rogaway, "Relations among notions of security for public-key encryption schemes," Proc. of CRYPTO'98, LNCS 1462, Springer-Verlag, pp.26-45, 1998.
3. M. Bellare, J. Killian and P. Rogaway, "The security of cipher block chaining," Proc. of CRYPTO'94, LNCS 839, Springer-Verlag, pp.341-358, 1994.
4. M. Bellare and P. Rogaway, "Optimal asymmetric encryption - How to encrypt with RSA," Proc. of Eurocrypt'94, LNCS 950, Springer-Verlag, pp.92-111, 1994.
5. V. Boyko, "On the security properties of OAEP as an all-or-nothing transform," Proc. of CRYPTO'99, LNCS 1666, Springer-Verlag, pp.503-518, 1999.
6. D. Dolev, C. Dwork and M. Naor, "Non-malleable cryptography," Proc. of 23rd ACM Symposium on the Theory of Computing (STOC), pp.542-552, 1991.
7. E.N. Gilbert, F.J. MacWilliams and N.J.A. Sloane, "Codes which detect deception," Bell System Technical Journal, 53, pp.405-425, 1974.
8. O. Goldreich, S. Goldwasser and S. Micali, "How to construct random functions," Journal of the ACM, vol.33, no.4, pp.210-217, 1986.
9. S. Goldwasser and S. Micali, "Probabilistic encryption," Journal of Computer and System Science, vol.28, pp.270-299, 1984.
10. G. Hanaoka, J. Shikata, Y. Zheng and H. Imai, "Unconditionally secure digital signature schemes admitting transferability," Proc. of Asiacrypt'00, LNCS 1976, Springer-Verlag, pp.130-142, 2000.
11. G. Hanaoka, J. Shikata, Y. Hanaoka and H. Imai, "Unconditionally secure anonymous encryption and group authentication," Proc. of Asiacrypt'02, LNCS 2501, Springer-Verlag, pp.81-99, 2002.
12. U.M. Maurer, "A unified and generalized treatment of authentication theory," Proc. of STACS'96, LNCS 1046, Springer-Verlag, pp.387-398, 1996.
13. L. McAven, R. Safavi-Naini and M. Yung, "Unconditionally secure encryption under strong attacks," Proc. of ACISP'04, LNCS 3108, Springer-Verlag, pp.427-439, 2004.
14. R. Rivest, "All-or-nothing encryption and the package transform," Proc. of FSE'97, LNCS 1267, Springer-Verlag, pp.210-218, 1997.
15. R. Rivest, "Chaffing and winnowing: confidentiality without encryption," http://theory.lcs.mit.edu/~rivest/publication.html.
16. J. Shikata, G. Hanaoka, Y. Zheng and H. Imai, "Security notions for unconditionally secure signature schemes," Proc. of Eurocrypt'02, LNCS 2332, Springer-Verlag, pp.434-449, 2002.
17. J. Shikata, G. Hanaoka, Y. Zheng, T. Matsumoto and H. Imai, "Unconditionally secure authenticated encryption," IEICE Trans., vol.E87-A, no.5, pp.1119-1131, 2004.
18. G.J. Simmons, "Authentication theory/coding theory," Proc. of CRYPTO'84, LNCS 196, Springer-Verlag, pp.411-431, 1984.
19. D.R. Stinson, "A construction for authentication/secrecy codes from Certain combinatorial designs," J. Cryptology vol.1, no.2, pp.119-127, 1988.

A Fast Calculus for the Linearizing Attack and Its Application to an Attack on KASUMI

Nobuyuki Sugio[1], Shunichi Nambu[2], and Toshinobu Kaneko[2]

[1] NTT DoCoMo, 3-5 Hikari-no-oka, Yokosuka, Kanagawa 239-8536, Japan
sugio@nttdocomo.co.jp
[2] Tokyo University of Science, 2641, Yamazaki, Noda, Chiba 278-8510, Japan
j7303655@ed.noda.tus.ac.jp, kaneko@ee.noda.tus.ac.jp

Abstract. This paper describes a linearizing attack with fast calculus for higher order differential attack. The linearizing attack, proposed by Shimoyama et al. [13], [15], linearizes the attack equation and determines the key by Gaussian elimination. The cost of calculating the coefficient matrix is dominant overhead in this attack. We improve the algorithm used to calculate the coefficient matrix by applying a bit-slice type implementation [3]. We apply this method to five-round KASUMI and show that it need $2^{27.5}$ chosen plaintexts and 2^{34} KASUMI encryptions.

1 Introduction

Higher order differential attack is a well-known attack against block ciphers. It exploits the properties of the higher order differentials of functions and derives an attack equation to determine the key. Jakobsen et al. applied it to $\mathcal{KN}$ cipher [8]. They used exhaustive search to solve the attack equation. Shimoyama et al. proposed an effective method of solving the attack equation [15] and Moriai et al. generalized it for the attack on CAST cipher [13]. Their method, which we call *linearizing attack* in this paper, linearizes the attack equation and solves the key by using Gaussian elimination. Hatano et al. proposed an optimization for linearizing attack [6] that is based on linear dependency between unknowns in the attack equation; it decreases the number of independent variables.

In the linearizing attack, the major computational cost is estimated to be the cost of calculating the coefficient matrix of unknown variables.[1] In this paper, we propose a fast calculus for an enhanced linearizing attack. We apply the bit-slice type implementation proposed by Biham [3] to the attack equation to calculate the coefficient matrix. We analyze elements of the coefficient matrix of unknown variables and calculate it using the T function proposed in this paper instead of a round function.

We apply the fast calculus to attack the 64-bit block cipher KASUMI. KASUMI [1] is based on the known block cipher MISTY1 [11] and is optimized for

[1] If the size of coefficient matrix is small, this computational cost ignores the complexity of solving the system of equations [13].

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 163–172, 2006.

Table 1. Comparison to previous attacks on KASUMI

Cipher	Rounds	Complexity		Comments
		Data	Time	
KASUMI	4^{*1}	$2^{10.5}$	$2^{22.11}$	Higher Order Differential Attack [18]
	5	$2^{39.4}$	2^{117}	Higher Order Differential Attack [19]
	5	$2^{22.1}$	$2^{60.7}$	Higher Order Differential Attack [16]
	5	$2^{27.5}$	$2^{39.9}$	Higher Order Differential Attack [20]
	5	$2^{27.5}$	2^{34}	**This paper**

*1-this attack is on a version of the cipher without FL functions.
Time complexity is measured in encryption units.

implementation in hardware. It is used in the confidentiality and integrity algorithm of 3GPP mobile communications. Table 1 lists the known attacks on KASUMI. Our method needs $2^{27.5}$ chosen plaintexts and 2^{34} KASUMI encryptions.

2 Preliminaries

2.1 Higher Order Differential [10]

Let $F(\cdot)$ be an encryption function as follows.

$$Y = F(X; K) \tag{1}$$

where $X \in \mathrm{GF}(2)^n$, $Y \in \mathrm{GF}(2)^m$, and $K \in \mathrm{GF}(2)^s$. X, K, and Y denote a plaintext, a key and a ciphertext, respectively. Let $\{A_1, \cdots, A_i\}$ be a set of linearly independent vectors in $\mathrm{GF}(2)^n$ and $V^{(i)}$ be the sub-space spanned by these vectors. The i-th order differential is defined as follows.

$$\Delta^{(i)}_{V^{(\,)}} F(X; K) = \bigoplus_{A \in V^{(\,)}} F(X \oplus A; K) \tag{2}$$

In the following, $\Delta^{(i)}$ denotes $\Delta^{(i)}_{V^{(\,)}}$, when it is clearly understood.

In this paper, we use the following properties of the higher order differential.

Property 1. If the degree of $F(X; K)$ with respect to X equals N, then

$$deg_X\{F(X; K)\} = N \Rightarrow \begin{cases} \Delta^{(N+1)} F(X; K) = 0 \\ \Delta^{(N)} F(X; K) = const \end{cases} \tag{3}$$

Property 2. The higher order differential has linear property on Exclusive-OR sum.

$$\Delta^{(N)}\{F(X_1; K_1) \oplus F(X_2; K_2)\} = \Delta^{(N)} F(X_1; K_1) \oplus \Delta^{(N)} F(X_2; K_2) \tag{4}$$

2.2 Attack of a Block Cipher

Consider an R-round block cipher. Let $H_{R-1}(X) \in \mathrm{GF}(2)^m$ be a part of the $(R-1)$-th round output and $C(X) \in \mathrm{GF}(2)^m$ be the ciphertext for the plaintext $X \in \mathrm{GF}(2)^n$. $H_{R-1}(X)$ is described as follows.

$$H_{R-1}(X) = F_{R-1}(X; K_1, \cdots, K_{R-1}) \tag{5}$$

Let K_i be an i-th round key and $F_i(\cdot)$ be a function of $\mathrm{GF}(2)^n \times \mathrm{GF}(2)^{s \times i} \rightarrow \mathrm{GF}(2)^m$.

If the degree of $F_{R-1}(\cdot)$ with respect to X is $N-1$, we have

$$\Delta^{(N)} H_{R-1}(X) = 0 \tag{6}$$

Let $\tilde{F}(\cdot)$ be a function that outputs $H_{R-1}(X)$ from the ciphertext $C(X) \in \mathrm{GF}(2)^m$.

$$H_{R-1}(X) = \tilde{F}(C(X); K_R) \tag{7}$$

where $K_R \in \mathrm{GF}(2)^s$ denotes the round key to decode $H_{R-1}(X)$ from $C(X)$. From Eq. (6), (7) and Property 1, the following equation holds.

$$0 = \Delta^{(N)} \tilde{F}(C(X); K_R) \tag{8}$$

In the following, we refer to Eq. (8) as the attack equation.

2.3 Linearizing Attack

Shimoyama et al. proposed an effective method of solving attack Eq. (8) [13], [15]. This method, called linearizing attack in this paper, linearizes the attack equation by treating every higher order variable like $k_i k_j$ with new independent variables like k_{ij}. In the following, we use the term *linearized attack equation* to refer to an attack equation that is regarded as a linear equation.

Let L be the number of unknowns in the linearized attack equation of Eq. (8). Since the attack Eq. (8) is derived by using an m-bit sub-block, we can rewrite it as follows.

$$\mathbf{A}\mathbf{k} = \mathbf{b}\ ,\ \mathbf{k} = {}^t(k_1, k_2, \ldots, k_1 k_2, \ldots, k_1 k_2 k_3, \cdots) \tag{9}$$

where $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{k}$ are the $m \times L$ coefficient matrix, the m-dimensional vector, and the L-dimensional vector over GF(2), respectively. $\mathbf{k}$ denotes linearized unknowns that are expressed as monomials of the R-th round key K_R.

We can obtain m linearized attack equations from one N-th order differential because Eq. (8) is an m-bit equation. Therefore, we need $\lfloor L/m \rfloor$ sets of the N-th order differential to determine a unique solution.

Since one set of N-th order differential requires 2^N chosen plaintexts, the number of plaintexts, M, needed to determine the key is estimated as

$$M = 2^N \times \left\lfloor \frac{L}{m} \right\rfloor \tag{10}$$

If we use the same technique shown in [13], [15], Eq. (9) requires $2^N \times (L+1)$ times $\tilde{F}(\cdot)$ calculations. Since we have to prepare $\lfloor L/m \rfloor$ sets of N-th order differentials to determine $\mathbf{k}$, the computational cost is estimated as

$$T = 2^N \times (L+1) \times \left\lfloor \frac{L}{m} \right\rfloor \tag{11}$$

3 Fast Calculus for the Linearizing Attack

Each element of the matrix $\mathbf{A}$ and the vector $\mathbf{b}$ in Eq. (9) can be expressed as a Boolean expression of ciphertext $C(X) = (c_1, c_2, \ldots, c_m)$ like $c_1 + \cdots + c_1c_2 + \cdots + c_1c_2c_3 + \cdots$. Let $\mathbf{a}_j$ $(j = 1, 2, \cdots, L+1)$ be a m-dimensional column vector of $\mathbf{A}$ and $\mathbf{b}$. $\mathbf{a}_j$ is calculated by using N-th order differentials, and is defined as follows.

$$\mathbf{a}_j = \Delta^{(N)} \mathbf{A}_j \mathbf{c} \ , \ \mathbf{c} = {}^t(c_1, \cdots, c_1c_2, \cdots, c_1c_2c_3, \cdots) \tag{12}$$

where $\mathbf{A}_j$ is an $m \times D$ constant matrix determined from Eq. (8) and $\mathbf{c}$ is a D-dimensional vector. The elements of $\mathbf{c}$ are ciphertext monomials which include higher order degrees. We can rewrite Eq. (12) as follows.

$$\mathbf{a}_j = \mathbf{A}_j \Delta^{(N)} \mathbf{c} \tag{13}$$

$\mathbf{c}$ is determined from ciphertexts. Since we calculate $\Delta^{(N)}\mathbf{c}$ for each set of N-th order differential, we are able to determine $\mathbf{a}_j$ by calculating Eq. (13) without using the $\tilde{F}(\cdot)$ function. Therefore, we can determine coefficient matrix $\mathbf{A}$ and vector $\mathbf{b}$ from Eq. (13).

Consider the derivation of $\mathbf{c}$ by using T function to calculate ciphertexts. We take T to be a D-bit output function that outputs elements of $\mathbf{c}$ and implement T by using the bit-slice method [3]. Since S-boxes are generally implemented as tables in an encryption function, we embed T as a table in the same way. If we implement it on a 32-bit processor, we need $\lfloor D/32 \rfloor$ table look-ups to retrieve D-bit elements. In this paper, we consider that the computational costs of table S-box and T function look-ups as being the same.

In the following, we introduce an algorithm for key derivation and estimate the necessary number of chosen plaintexts and the computational cost.

Algorithm for key derivation
Step 0: Prepare $\lfloor L/m \rfloor$ sets of N-th order differentials.
Step 1: Calculate $\Delta^{(N)}\mathbf{c}$ using one set of N-th order differential and repeat the calculation for $\lfloor L/m \rfloor$ sets.
Step 2: Calculate $\mathbf{a}_j$ $(j = 1, 2, \cdots, L+1)$ from Eq. (13).
Step 3: Determine the key by solving Eq. (9) with a method such as Gaussian elimination.

The necessary number of chosen plaintexts M' for key derivation is the same as Eq. (10).

$$M' = 2^N \times \left\lfloor \frac{L}{m} \right\rfloor \tag{14}$$

We estimate the computational cost for each step of the algorithm for key derivation as follows.

Step 1: It needs $2^N \times \lfloor D/32 \rfloor$ table look-ups to calculate $\Delta^{(N)}\mathbf{c}$ for each N-th order differential. Thus Step 1 has computational cost of T'_{Step1} as follows.

$$T'_{Step1} = 2^N \times \left\lfloor \frac{D}{32} \right\rfloor \times \left\lfloor \frac{L}{m} \right\rfloor \tag{15}$$

Step 2: In calculating Eq. (13), we calculate inner products of m sets of row vectors of $\mathbf{A}_j$ and $\Delta^{(N)}\mathbf{c}$. This needs $2\times\lfloor D/32\rfloor\times m\times(L+1)$ table look-ups. Since we prepare $\lfloor L/m \rfloor$ sets of N-th order differentials, the necessary computational cost of Step 2 is estimated to be

$$T'_{Step2} = 2 \times \left\lfloor \frac{D}{32} \right\rfloor \times m \times (L+1) \times \left\lfloor \frac{L}{m} \right\rfloor \approx 2 \times \left\lfloor \frac{D}{32} \right\rfloor \times L^2 \tag{16}$$

Step 3: Solving Eq. (9) with a method such as Gaussian elimination is generally estimated to cost about L^3. In this paper, since we evaluate computational cost assuming the use of a 32-bit processor, Step 3 costs $T1_{Step3}$ as follows.

$$T'_{Step3} = \left\lfloor \frac{L^3}{32} \right\rfloor \tag{17}$$

Therefore the necessary computational cost, T', of this algorithm is evaluated as follows.

$$T' = T'_{Step1} + T'_{Step2} + T'_{Step3} \tag{18}$$

4 Higher Order Differential Attack on KASUMI

4.1 KASUMI

KASUMI is a Feistel type block cipher with 64-bit data block and 128-bit secret key. It is based on MISTY1 [11] which has provable security against linear and differential cryptanalysis [4], [12]. In 2000, the 3rd Generation Partnership Project (3GPP)[2] selected KASUMI as the mandatory cipher in Wideband Code Division Multiple Access (W-CDMA). It is used in the confidentiality and integrity algorithm of 3GPP mobile communications. Fig. 1 outlines its block diagrams with equivalent FO and FI functions; we call it KASUMI hereafter.

4.2 Previous Results

Tanaka et al. proposed the first attack on 5-round KASUMI with a 32-nd order differential by using a bijective round function feature [19]. Sugio et al. searched for an effective chosen plaintext by computer simulations and reduced

[2] 3GPP is a consortium that standardize the 3rd Generation Mobile System.

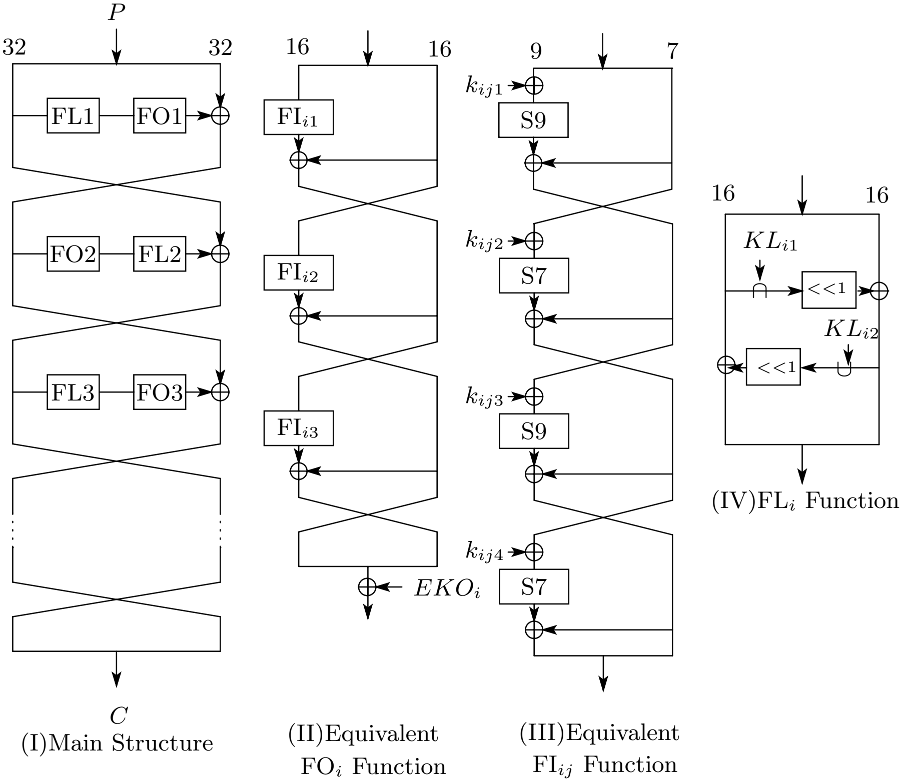

Fig. 1. KASUMI

the necessary number of plaintexts and computational cost by using a 16-th order differential [16]. In the following, we introduce the outline of [16].

Let $H_i = (h_{i4}, h_{i3}, h_{i2}, h_{i1})$ where $h_{i4}, h_{i2} \in \mathrm{GF}(2)^7$ and $h_{i3}, h_{i1} \in \mathrm{GF}(2)^9$ are the right half of the i-th round output. With KASUMI, plaintext X is divided into eight sub-blocks as follows.

$$X = (X_7, X_6, \ldots, X_0) \quad X_i \in \begin{cases} \mathrm{GF}(2)^9 \ (i = \mathrm{odd}) \\ \mathrm{GF}(2)^7 \ (i = \mathrm{even}) \end{cases} \tag{19}$$

The following plaintext, obtained by computer simulations, is the effective chosen plaintext that enables us to reduce the necessary number of chosen plaintexts and computational cost.

$$X \in (\mathcal{C}, \mathcal{C}, \mathcal{C}, \mathcal{C}, \mathcal{X}_2, \mathcal{X}_1, \mathcal{C}, \mathcal{C}) \quad \mathcal{X}_1, \mathcal{X}_2 : \text{variable}, \ \mathcal{C} : \text{fixed sub-block} \tag{20}$$

Using the above chosen plaintext, we have a constant value that denotes $\Delta^{(16)} h_{33} = 0 \in \mathrm{GF}(2)^9$. Accordingly, we derive the attack equation as follows.

$$\Delta^{(16)}\{\mathrm{FO}_5^9(\mathrm{FL5}(C_L; KL_5); KO_5) \oplus C_R^9\} = 0, \tag{21}$$

where C_L and C_R denote the left and right 32-bit ciphertext, respectively, and $\mathrm{FO}_5^9(\cdot)$ and C_R^9 denote the 9-bits corresponding to h_{33}. Eq. (21) has 82-bit equivalent keys[3]. Sugio et al. estimated the key by combining exhaustive search with the linearizing attack [16]. It needs 2^{22} chosen plaintexts and 2^{63} (FO+FL) function operations.

Nambu et al. analyzed unknown variables $L = 26,693$ in linearized attack equations using the computer software REDUCE. They estimated 82 equivalent key bits by the linearizing attack. It needs $2^{27.5}$ chosen plaintexts and $2^{42.2}$ (FO+FL) function operations [20].

4.3 Application of Fast Calculus to an Attack on KASUMI

In the following, we will demonstrate an application of fast calculus to an attack on KASUMI. We linearize Eq. (21) and express it as follows.

$$\mathbf{A}\mathbf{k} = \mathbf{b} \tag{22}$$

where $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{k}$ are the $9 \times 26,693$ coefficient matrix, the 9-dimensional vector, and the 26,693-dimensional vector, respectively. If we determine the coefficient matrix $\mathbf{A}$ and the vector $\mathbf{b}$ by calculating Eq. (13), we need to analyze the constant matrixes $\mathbf{A}_j$ and the vector $\mathbf{c}$. Therefore, we analyzed $\mathbf{A}_j$ and $\mathbf{c}$ by expanding the Boolean expressions of Eq. (21) with the computer software REDUCE. We show the number of elements of $\mathbf{c}$ in Table 2.

Table 2. Analysis of the number of elements of **c**

bit position	# of elements of **c**
16-th bit	6537
17-th bit	6686
18-th bit	6237
19-th bit	6433
20-th bit	6419
21-th bit	6713
22-th bit	6569
23-th bit	6493
24-th bit	6854
all	9109

'*all*' denotes the number of all elements of **c** in 9-bits.

As a result, we determined $\mathbf{A}_j$ $(j = 1, 2, \ldots, 26,694)$ as the 9×9109 matrixes and $\mathbf{c}$ as the 9109-dimensional vector. In the following, we estimate the number of chosen plaintexts needed and the computational cost for the fast calculus.

[3] $KL_5 = (KL_{51}, KL_{52})$ 32 bits and $KO_5 = (k_{511}, k_{512}, k_{513}, k_{521}, k_{522}, k_{523})$ 50 bits.

Estimation of Complexity

Since unknown variables $L = 26,693$ exist in the linearized system of equations, we need $\lfloor 26,693/9 \rfloor$ sets of 16-th order differentials to determine the key. Therefore, the necessary number of chosen plaintexts is estimated as follows.

$$M' = 2^{16} \times \left\lfloor \frac{26,693}{9} \right\rfloor \approx 2^{27.5} \tag{23}$$

We can estimate the computational cost, T', by calculating Eq. (15),···,(18).

$$T'_{Step1} = 2^{16} \times \left\lfloor \frac{9109}{32} \right\rfloor \times \left\lfloor \frac{26,693}{9} \right\rfloor \approx 2^{35.69} \tag{24}$$

$$T'_{Step2} \approx 2 \times \left\lfloor \frac{9109}{32} \right\rfloor \times 26,693^2 \approx 2^{38.56} \tag{25}$$

$$T'_{Step3} = \left\lfloor \frac{26,693^3}{32} \right\rfloor \approx 2^{39.11} \tag{26}$$

$$T' = T'_{Step1} + T'_{Step2} + T'_{Step3} = 2^{39.94} \tag{27}$$

We compare Eq. (27) to the previous results. In Eq. (27), we estimate the computational cost as the number of table look-ups of the T function and matrix calculations. According to Fig. 1, each FI function has two S9-boxes and two S7-boxes and so each FO function has (S9 × 2 + S7 × 2) × 3 = 12 S-boxes. Therefore, we regard the computational cost of Eq. (27) as $2^{39.94}/12 \approx 2^{36.4}$ (FO+FL) function operations and this is equivalent to $2^{36.4}/5 \approx 2^{34}$ KASUMI encryptions.[4] We summarize the results of this fast calculus in Table 1.

5 Conclusion

In this paper we applied higher order differential attack to five-round KASUMI. We proposed a linearizing attack with fast calculus that can reduce the complexity incurred in calculating the coefficient matrix **A** and the vector **b**. Our attack requires $2^{27.5}$ chosen plaintexts and 2^{34} encryptions.

In the linearizing attack, we solve the system of linearized equations by using Gaussian elimination. If the number of unknown variables L is large, we can't ignore the computational cost of Gaussian elimination. Therefore, if we decrease the number of unknown variables, we can diminish the total computational cost, T'. We outlined a technique that eliminates unknown variables for the fast calculus in the appendix. We will be able to reduce the computational cost for the key derivation by using this elimination technique.

[4] We discuss here an attack on a 5-round variant.

References

1. 3GPP TS 35202. "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 2: KASUMI Specification", http://www.3gpp.org/tb/other/algorithms.htm
2. S. Babbage and L. Frisch. "On MISTY1 higher order differential cryptanalysis", proceedings of 3rd International Conference on Information Security and Cryptology, Lecture Notes in Computer Science 2015, pp. 22-36, Springer-Verlag, 2000.
3. E. Biham. "A Fast New DES Implementation in Software", proceedings of Fast Software Encryption 4th International Workshop, Lecture Notes in Computer Science 1267, pp. 260-272, Springer-Verlag, 1997.
4. E. Biham, A.Shamir. "Differential Cryptanalysis of DES-Like Cryptosystems", Journal of Cryptology, 4(1), pp.3-72, 1991.
5. Y. Hatano, H. Sekine, T. Kaneko. "Higher Order Differential Attack of Camellia (II)", proceedings of Selected Areas in Cryptography 2002 9th Annual International Workshop, pp. 129-146, Lecture Notes in Computer Science 2595 Springer-Verlag 2003.
6. Y. Hatano, H.Tanaka, T.Kaneko. "Optimization for the algebraic method and its application to an attack of MISTY1", IEIEC TRANS. FUNDAMENTALS, Vol.E87-A, No.1, pp.18-27, January, 2004.
7. T. Iwata, K. Kurosawa. "Probabilistic Higher Order Differential Attack and Higher Order Bent Functions", proceedings of Advances in Cryptology - ASIACRYPT '99, International Conference on the Theory and Applications of Cryptology and Information Security, Lecture Notes in Computer Science 1716, pp. 62-74, Springer-Verlag 1999.
8. T. Jakobsen and L. R. Knudsen. "The Interpolation Attack on Block Cipher", proceedings of Fast Software Encryption 4th International Workshop, Lecture Notes in Computer Science 1267, pp. 28-40, Springer-Verlag, 1997.
9. L. R. Knudsen and D. Wagner. "Integral Cryptanalysis", proceedings of Fast Software Encryption 9th International Workshop, Lecture Notes in Computer Science 2365, pp. 112-127, Springer-Verlag, 2002.
10. X. Lai. "Higher Order Derivatives and Differential Cryptanalysis", proceedings of Communications and Cryptography, pp.227-233, Kluwer Academic Publishers, 1994.
11. M. Matsui. "New Block Encryption Algorithm MISTY", proceedings of Fast Software Encryption 4th International Workshop, Lecture Notes in Computer Science 1267, pp. 54-67, Springer-Verlag, 1997.
12. M. Matsui. "Linear Cryptanalysis Method for DES Cipher", Advances in Cryptology, proceedings of EUROCRYPT'93, Lecture Notes in Computer Science 765, pp. 386-397, Springer-Verlag, 1994.
13. S. Moriai, T. Shimoyama, and T. Kaneko. "Higher Order Differential Attack of a CAST Cipher", proceedings of Fast Software Encryption 5th International Workshop, Lecture Notes in Computer Science 1372, pp. 17-31, Springer-Verlag, 1998.
14. S. Moriai, T. Shimoyama, T. Kaneko. "Higher Order Differential Attack Using Chosen Higher Order Differences", proceedings of Selected Areas in Cryptography 1998, pp. 106-117, Lecture Notes in Computer Science 1556 Springer-Verlag 1999.
15. T. Shimoyama, S. Moriai, T. Kaneko, and S. Tsujii. "Improving Higher Order Differential Attack and Its Application to Nyberg-Knudesen's Designed Block Cipher", IEIEC Trans. Fundamentals, Vol.E82-A, No.9, pp. 1971-1980, September, 1999.

16. N. Sugio, H. Tanaka, and T. Kaneko. "A Study on Higher Order Differential Attack of KASUMI", proceedings of International Symposium on Information Theory and its Applications 2002, pp. 755-758, 2002.
17. H. Tanaka, K. Hisamatsu, T. Kaneko. "Strength of MISTY1 without FL Function for Higher Order Differential Attack", proceedings of Applied Algebra, Algebraic Algorithms and Error-Correcting Codes 13th International Symposium, Lecture Notes in Computer Science 1719, pp. 221-230, Springer-Verlag 1999.
18. H. TanakaCC. IshiiCT. Kaneko. "On the Strength of KASUMI without FL Functions against Higher Order Differential Attack", proceedings of 3rd International Conference on Information Security and Cryptology, Lecture Notes in Computer Science 2015, pp. 14-21, Springer-Verlag, 2000.
19. H. TanakaCC. IshiiCT. Kaneko. "On the strength of block cipher KASUMI and MISTY", proceedings of Symposium on Cryptography and Information Security 2001 (in Japanese), pp. 647-652, 2001.
20. S. Nambu, T. Kaneko. "A Study on Higher Order Differential Attack of KASUMI (Ⅲ)", proceedings of The 27th Symposium on Information Theory and Its Applications 2004 (in Japanese), pp.45-48, 2004.

A A Technique for Eliminating Unknown Variables for the Fast Calculus

We will outline an elimination technique of unknown variables for the fast calculus by using lots of the linearized attack equation.

We linearize the attack equation in the same way as Eq. (9) and divide unknown variables L into L' and L''. Let $\mathbf{a}_j$ $(j = 1, 2, \ldots, L+1)$ be a m-dimensional column vector of $\mathbf{A}$ and $\mathbf{b}$, and let the elements of $\mathbf{a}_j$ $(j = 1, 2, \ldots, L')$ be D' ciphertext monomials which include higher order degrees and elements of $\mathbf{a}_j$ $(j = L' + 1, \ldots, L + 1)$ be the same as those of $\mathbf{c}$. Therefore, we can rewrite Eq. (13) as

$$\mathbf{a}_j = \begin{cases} \mathbf{A}_j \Delta^{(N)} \mathbf{c}'_i & (i = 1, 2, \ldots\ ,\ j = 1 \sim L') \\ \mathbf{A}_j \Delta^{(N)} \mathbf{c}_i & (i = 1, 2, \ldots\ ,\ j = L' + 1 \sim L + 1), \end{cases} \tag{28}$$

where $\mathbf{c}'_i$ is a D'-dimensional vector that is composed of a part of the elements of D-dimensional vector $\mathbf{c}_i$. Therefore, if we prepare $Q(> D')$ sets of N-th order differentials and calculate each $\Delta^{(N)}\mathbf{c}'_i$ $(i = 1, 2, \ldots, Q)$, we can determine $\Delta^{(N)}\mathbf{c}'_i = \mathbf{0}$ $(i = D' + 1, \ldots, Q)$ by using linear dependency of $\Delta^{(N)}\mathbf{c}'_i$. In Eq. (28), if $\Delta^{(N)}\mathbf{c}'_i$ equals $\mathbf{0}$, we can determine $\mathbf{a}_j = \mathbf{0}$. Since $\mathbf{a}_j$ $(j = 1, 2, \ldots, L')$ that correspond to unknown variables L' equals $\mathbf{0}$, it is not necessary to estimate L'.

If many unknown variables L exist in the linearized attack equation, we will be able to reduce the computational cost for the key derivation by using this elimination technique.

On Achieving Chosen Ciphertext Security with Decryption Errors

Yang Cui[1], Kazukuni Kobara[2], and Hideki Imai[2]

[1] Dept. of Information & Communication Engineering, University of Tokyo
cuiyang@imailab.iis.u-tokyo.ac.jp
[2] Institute of Industrial Science, University of Tokyo,
Komaba 4-6-1, Meguro-Ku, Tokyo, 153-8505, Japan
{kobara, imai}@iis.u-tokyo.ac.jp
http://imailab-www.iis.u-tokyo.ac.jp/imailab.html

Abstract. Perfect decryption has been always assumed in the research of public key encryption, however, this is not true all the time. For some public key encryption primitives, like NTRU [9] or Ajtai-Dwork [1], the decryption process may not obtain the corresponding message even the encryption and decryption are run correctly. Furthermore, such a kind of decryption errors will lead to some dangerous attacks against the underlying primitive. Another interesting point is that, those primitives are not based on the factoring, nor the discrete log problem which are subject to the Shor's algorithm [18] with quantum computers. This kind of primitives may be promising in the post-quantum cryptography. Therefore, the decryption errors deserve much attention and should be coped with carefully.

In this paper, our main technique is not to use any error-correcting codes to *eliminate* the errors, but to use some padding (transform) to *hide* "bad" errors from attacker's control. We 1) efficiently enhance these error-prone public key encryption primitives to the chosen ciphertext security, even in the presence of the decryption errors, and 2) show that the solution is more generic, rather than some specific padding methods previously presented, to thwart the decryption errors based attacks successfully.

1 Introduction

Public key encryption (PKE) is a crucial building block in cryptography, widely used in many security protocols and schemes. Whilst various PKEs are proposed to fulfill with the requirement in different scenarios, one property of PKE is always assumed, which is the perfect decryption. It means that any validly encrypted ciphertext will lead to the same message corresponding to the ciphertext for certainty. However, there exists a family of PKE that has good performance in the implementation, but fails to have perfect decryption sometime, such as NTRU [9] and Ajtai-Dwork [1] etc.

Even though their decryption errors do not occur often, they do have been affected greatly. Indeed the decryption errors we care about are not only possible

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 173–182, 2006.

to reduce the efficiency of PKE, but might also give additional useful information to potential attackers, thus lead to a fatal attack, such as secret key exposure [16].

Given a PKE (a randomly generated public and secret key pair), for some message and randomness pair, the encryption algorithm may lead to a mapping which is not injective. This would be inevitable if some specific message pairs are chosen, and further it gives the opportunity for the attacker to know the truth - some ciphertexts are corresponding to decryption error message, which is never desired to be known by the attacker with strong power, such as adaptively chosen ciphertext attacker [17].

Although perfect decryption has not been achieved, this kind of PKE is so meaningful after the Shor's factoring algorithm [18]. Its significance lies in that they are not based on the common number-theoretic problems of factoring or discrete log, like RSA or ElGamal, but on the lattice problem which is believed hard to be solved even that the quantum computer is built in the future. Additionally, since the fast implementation of them can be compared with RSA, this family may be a promising replacement of the commonly used PKE, if it could be made immune to the decryption errors.

1.1 Related Work

There are some related work in this context, Goldreich et al [7] proposed a solution to the decryption problem of Ajtai-Dwork [1], but failed to make the scheme secure [12]. Later, Dwork et al [5] generalized the theoretical solution to solve any infrequent decryption errors, using several totally impractical techniques as parallel repetition, hard core bit and direct product. By these only theoretically meaningful techniques, the error probability could be found with only a tiny probability, i.e. the attack using decryption errors is made impossible to run efficiently.

Although some efficient work by Howgrave-Graham et al. [10] provided an exclusive use padding scheme for NTRU, called NAEP, to enhance the security of NTRU even in the presence of decryption errors, it was especially designed and thus not useful for any other PKE. As NAEP did, in the random oracle model [2] [1], an efficient solution in [5] also used a padding to enhance the security practically. However, this transform appears a little complex and not good at bandwidth overhead, having several padding schemes together, like Fujisaki-Okamoto [6] combined with PSS-E [4], where a symmetric encryption is also required.

1.2 Main Contributions

From a practical viewpoint, we expect the padding methods be generic so that it could deal with many other PKEs, no matter whether there exist decryption errors or not. In addition, the efficiency is also important, otherwise it will be too expensive for this kind of error-prone encryptions.

[1] A useful tool to design and analyze the cryptosystem, is widely used in both theory and practice.

Our main method is not to correct the errors, but to hide the errors from the attacker's control. We point out that actually, some existed generic padding may help deterministic primitive immunize the attack from the decryption errors, such as 3-round OAEP [14]. And we provide a new variant of it to cope with probabilistic primitive as well. Note that both of them are generic to adapt to many other PKEs, and very efficient, especially in bandwidth.

Next, we will first explain the security notions and the attack by Proos, then show that some error-prone PKEs could be enhanced to chosen ciphertext security provably when decryption failures occur.

2 Notions and Notations

In the following paper, we define $\mathcal{M}$, $\mathcal{R}$ as the message and randomness space respectively, and $\mathcal{C}$ is the ciphertext space, where $\mathcal{C} = \mathcal{M} \times \mathcal{R}$. $\Pr[\texttt{operation}|\cdot]$ represents the probability of event "$\cdot$" under the corresponding operation. And we say that $\texttt{negl}(k)$ is negligible, if for any constant c, there exists $k_0 \in \mathbf{N}$, s.t. $\texttt{negl}(k) < (1/k)^c$ for any $k > k_0$.

2.1 Public Key Encryption

Definition 1. *Public key encryption Π is defined by a triple of algorithms, ($\mathcal{K}$, $\mathcal{E}$, $\mathcal{D}$):*

- *the key generation algorithm $\mathcal{K}$: on a secret input 1^k ($k \in \mathbf{N}$), in polynomial time in k, it produces a pair of keys (pk, sk), public and secret known respectively.*
- *the encryption algorithm $\mathcal{E}$: on input of message $m \in \mathcal{M}$ and public key pk, the algorithm $\mathcal{E}(m, r)$ produces the ciphertext c of m, $c \in \mathcal{C}$. (random coins $r \in \mathcal{R}$).*
- *the decryption algorithm $\mathcal{D}$: By using a ciphertext c and the secret key sk, $\mathcal{D}$ returns the plaintext m, s.t.*

$$\Pr[\mathcal{D}_{\mathsf{sk}}(\mathcal{E}_{\mathsf{pk}}(m, r)) = m] = 1$$

or when it is an invalid ciphertext, outputs $\perp$. This algorithm is deterministic.

Definition 2. *Error-prone Public key encryption $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$*

- *$\mathcal{K}'$ is equivalent to $\mathcal{K}$, except that there may exist such a pair (pk, sk), corresponding to the $\mathcal{D}'$ but not to $\mathcal{D}$.*
- *$\mathcal{E}'$ is equivalent to $\mathcal{E}$, except that there may exist pairs (m, r) which do not fit the algorithm $\mathcal{D}$.*
- *$\mathcal{D}'$ decrypt the ciphertext $c \in \mathcal{C}$ with overwhelming probability, though,*

$$\Pr[\mathcal{D}_{\mathsf{sk}}(\mathcal{E}_{\mathsf{pk}}(m, r)) \neq m] \leq \texttt{negl}(k)$$

Definition 3. *A public key encryption scheme is said to be* OW-PCA *secure, if any polynomial-time adversary $\mathcal{A}$, with the public data and the help of the plaintext-checking oracle $\mathcal{O}_{\mathsf{pca}}$, can get the whole preimage of the ciphertext with* *at most q queries to $\mathcal{O}_{\mathsf{pca}}$, in a time bound t and a winning probability no more than negligible:*

$$\Pr\left[\begin{array}{l}(\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{K}(1^k)\\ m \leftarrow \mathcal{M}, r \overset{R}{\leftarrow} \Omega\\ c \leftarrow \mathcal{E}_{\mathsf{pk}}(m;r), m' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{pca}}}(c)\end{array}\middle| m' = m\right] \leq \mathtt{negl}(k)$$

Remark. Naturally, the security of OW-PCA primitive is dependent on the inverting the cipher even with the help of plaintext-checking oracle, which is a polynomial-time turing machine able to decide whether a cipher and a message is the corresponding encryption pair, or not, which is firstly introduced by Okamoto and Pointcheval [13]. The reason why we introduce the notion hereby is, some famous padding like REACT [13] has been used to enhance the security of error-prone primitive, e.g. NTRU, without concerning the decryption errors. The result is rigorously proved though, it loses the security as soon as decryption error based attack is employed. Furthermore, if we could show that some transforms are possible to rescue the provable security based on the OW-PCA even in the presence of decryption errors, we may successfully enhance lots of the public key encryption primitives, since almost all commonly used PKEs are in OW-PCA security.

Beyond the one-wayness, the *polynomial indistinguishability* [8] of the encryption can make the leakage of any partial information as hard as that of the whole plaintext. In order to make sense in the strongest attack scenario, the IND should be considered in the CCA model, called IND-CCA [17], which has become the *de facto* requirement of the public key cryptosystem, as follows.

Definition 4. *A public key encryption scheme is* IND-CCA *secure, if there exists no polynomial-time adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ who, under the help of the decryption oracle, can distinguish the encryption of two equal-length, distinct plaintexts, with the probability significantly greater than 1/2 (the only restriction is that the target ciphertext cannot be sent to the decryption oracle directly). More formally, the scheme is* IND-CCA *secure, if with the time bound t, decryption oracle querying bound q, the following is satisfied:*

$$\Pr_{\substack{\leftarrow\{0,1\}\\ \leftarrow}}\left[\begin{array}{l}(\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{K}(1^k)\\ (m_0, m_1, s) \leftarrow \mathcal{A}_1^{\mathcal{O}}(\mathsf{pk})\\ c \leftarrow \mathcal{E}_{\mathsf{pk}}(m_b;r)\\ \hat{b} \leftarrow \mathcal{A}_2^{\mathcal{O}}(c, m_0, m_1, s, \mathsf{pk})\end{array}\middle| \hat{b} = b\right] \leq \frac{1}{2} + \mathtt{negl}(k)$$

Remark. IND-CCA security is such a strong security notion that it is considered to leak no single bit of the useful information against even very dangerous attack. On the other hand, however, it is subject to the decryption errors as well. For example, the famous Naor-Yung paradigm [11], which uses two independent public keys to encrypt one same message, together with some proof that the message two ciphertext encrypted is the same, is denied as long as decryption errors occur. Thus, we can find that the failure of decryption leads to not only efficiency lost, but also security flaw. Similarly for Ajtai-Dwork scheme [1], the decryption errors were claimed to be eliminated by Goldreich et al [7], however, later was pointed put to be insecure or totally impractical by Nguyen [12]. Very recently, Proos gave a successful attack based on decryption failure of NTRU, which denied the provable security of many previous transforms, such as REACT-NTRU, and OAEP-NTRU.

3 Proos's Attack

In 2003, Proos [16] provided an attack which for an error-prone public key encryption $(\mathcal{K}', \mathcal{E}', \mathcal{D}')$, can break the scheme totally, i.e. to find the secret key, whereas the scheme remains IND-CCA secure, if with perfect decryption$(\mathcal{K}, \mathcal{E}, \mathcal{D})$.

If an encryption scheme has the perfect decryption, the act of decrypting a valid ciphertext will provide no useful information to attackers. However, if the error-prone decryption is employed, the error occurred may give useful information for attackers to determine the information of the secret key, such as whether a valid ciphertext is correctly encrypted or not. Note that even a valid ciphertext is encrypted correctly, the secret information is still possible to leak due to the imperfect decryption. Next we will explain the attack by Proos [2].

3.1 Decipherable Ciphertext Attacks

Let $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ be an error-prone public key encryption primitive. Given a randomly generated key pair (pk,sk), and a decipherable oracle, $DC_{(\mathsf{pk},\mathsf{sk})}$ is an oracle which on input $(x,\ r,\ y)$ s.t. $\mathcal{E}_{\mathsf{pk}}(x, r) = y$ returns whether or not $\mathcal{D}_{\mathsf{sk}}(y) = x$. That is, a DC oracle can be used to determine if a valid ciphertext encrypted using pk can be correctly decrypted using sk. An attack using the public information and a DC oracle will be named as a decipherable ciphertext attack (DCA). Since naturally, a DC oracle gives additional information on the decryption, DCA is stronger than plain chosen plaintext attack. As it is also able to be simulated by decryption oracle, DCA is no stronger than CCA, obviously. And it is also adapted to the perfect decryption case, though seems a little redundant.

3.2 Attack on IND-CCA Transform

The encryption primitive may not be IND-CCA secure originally, however, there are many ways to enhance its security to the "appropriate" level, such as by the Optimal Asymmetric Encryption Padding (OAEP) [3], or Rapid Enhanced-security Asymmetric Cryptosystem Transform (REACT) [13] in the random oracle model.

Unfortunately, by the Proos's attack [16], guaranteed security for perfect decryption transform is not available any more for imperfect decryption ones. There exists such a scheme which can be proven secure in the perfect decryption scenario, but fails to hold the security in the imperfect decryption scenario, due to

[2] Due to the page limit, we omit the introduction of NTRU. Please refer to [9, 10] for the details why decryption errors occur.

leakage of useful information when answering the query of decipherable ciphertext attack. We will explain it in the following.

Take the OW-PCA secure PKE $(\mathcal{K}', \mathcal{E}', \mathcal{D}')$ as an example, the enhanced security $\Pi^R = (\mathcal{K}^R, \mathcal{E}^R, \mathcal{D}^R)$ as the following:

- $\mathcal{K}^R = \mathcal{K}'$.
- $\mathcal{E}^R_{\mathsf{pk}}(m, s, r)$, for a message m, choose randomness s and r, let $c_1 = \mathcal{E}_{\mathsf{pk}}(s, r)$; and use cryptographic hash functions G and H to compute $c_2 = G(s) \oplus m$, with $c_3 = H(s, m, c_1, c_2)$. At last, define the ciphertext $c = (c_1, c_2, c_3)$.
- $\mathcal{D}^R_{\mathsf{sk}}(c_1, c_2, c_3)$, for a ciphertext $c = (c_1, c_2, c_3)$, $s' = \mathcal{D}_{\mathsf{sk}}(c_1)$, $m' = G(s') \oplus c_2$ and $c_3' = H(s, m', c_1, c_2)$. If $s' \in \mathcal{M}$ and $c_3' = c_3$ then output m', otherwise output $\perp$.

The above encryption is able to be proven the IND-CCA security without the presence of the decryption error, due to [13], while we hereby would like to consider another situation.

It turns out that when Π^R with the decryption errors, the IND-CCA security loses, due to the fact that OW-PCA PKE will not return $\perp$ for all invalid ciphertexts, which would provide a convenience for the attacker to break the PKE totally.

Consider a PKE with decryption errors $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$, and assume that the attacker has found k invalid ciphertext $y_1, y_2 ..., y_k$ of Π', then the attacker could build $\Pi^{R'}$ by using REACT transform as follows:

- $\mathcal{K}^{R'} = \mathcal{K}'$
- $\mathcal{E}^{R'}_{\mathsf{pk}}(m, r) = \mathcal{E}'_{\mathsf{pk}}(m, r)$
-

$$\mathcal{D}^{R'}_{\mathsf{sk}}(y) = \begin{cases} \mathcal{D}'_{\mathsf{sk}}(y) & \text{if } y \notin \{y_1, y_2, ..., y_k\} \\ x & \text{if } y = y_i \text{, and the i-th bit of } \mathsf{sk} \text{ is } 1 \\ \perp & \text{otherwise} \end{cases}$$

Clearly, the new $\Pi^{R'}$ and original Π' are indistinguishable to a PCA attacker, thus $\Pi^{R'}$ is also OW-PCA secure.

The following attack could be applied to $\Pi^{R'}$ to recover the secret key. For $1 \leq i \leq k$ form a $\Pi^{R'}$ ciphertext $y_i^{R'}$ with $s = x$ and c_1 replaced by y_i. The $y_i^{R'}$ then could be sent to the DCA oracle with the knowledge that $y_i^{R'}$ will decrypt to $\perp$ if and only if the i-th bit of sk is zero. Thus $\Pi^{R'}$ is no more IND-CCA.

4 Generic Transforms

Since the underlying attack seems not able to be prevented by the previous scheme, [10] presented a new exclusive use padding for NTRU. However, we find that it is able to employ currently existed generic transform, 3-round OAEP [14] and its variant for this mission. The merit of the schemes is that they are not only useful to error-prone PKE, but also applicable to other common PKE;

furthermore, the 3-round OAEP has a good performance in efficiency, i.e. saves the bandwidth and is less redundant.

4.1 Concrete Construction

The scheme is described in the following:

Setup. On the security parameter, key generation algorithm randomly output a pair of $(\mathsf{pk},\mathsf{sk})$. Assume the random oracle family $\mathcal{H}$, and $F, G, H \overset{R}{\leftarrow} \mathcal{H}$,

$$\begin{aligned} F &: \{0,1\}^k \mapsto \{0,1\}^n, \\ G &: \{0,1\}^n \mapsto \{0,1\}^k, \\ H &: \{0,1\}^k \mapsto \{0,1\}^n. \end{aligned}$$

"$||$" represents bit concatenation. Let a sequence of bit zero be k_0-bit long, then the message length will be $n - k_0$.

Construction (3-round OAEP). The transform for deterministic encryption is defined as the following:

```
**Encryption** Enc(m)
    $w := [m||0^{k_0}] \oplus F(r)$
    $t := r \oplus G(w)$
    $s := H(t) \oplus w$
    $c := \mathcal{E}_{\mathsf{pk}}(s||t)$
```

```
**Decryption** Dec(c)
    $\mathcal{D}_{\mathsf{sk}}(c) := (s||t)$
    $w := H(t) \oplus s$
    $r := G(s) \oplus t$
    $m'||\mathsf{o} := w \oplus F(r)$
    If $\mathsf{o} = 0^{k_0}$, then $m = m'$
    otherwise, return $\perp$
```

A New Proposal. When the primitive is a probabilistic encryption scheme, the transform shall be changed, correspondingly. Use one more random oracle $H' \overset{R}{\leftarrow} \mathcal{H}$, $H' : \{0,1\}^{n+k} \mapsto \{0,1\}^{k'}$, and $r' = H'(m||r)$, be used as the required randomness of the probabilistic encryption. And the sequence of bit zero becomes not necessary. Others hold the same.

```
**Encryption** Enc(m)
    $w := m \oplus F(r)$
    $t := r \oplus G(w)$
    $s := H(t) \oplus w$
    $r' := H'(m||r)$
    $c := \mathcal{E}_{\mathsf{pk}}(s||t; r')$
```

```
**Decryption** Dec(c)
    $\mathcal{D}_{\mathsf{sk}}(c) := (s||t; r')$
    $w := H(t) \oplus s$
    $r := G(s) \oplus t$
    $m' := w \oplus F(r)$
    If $r' = H'(m'||r)$, then $m = m'$
    otherwise, return $\perp$
```

Remark. Actually, 3-round OAEP [14] has been proposed for its nice property of size efficiency. However, another property of this transform that it is inherently immune to the attack based on decryption failures, was not carefully studied and analyzed. Besides, since 2-round OAEP is widely used now, this is a good candidate for promoting uses. And we still provide a slight modification of 3-round OAEP, which adapts to the probabilistic encryption with decryption errors.

4.2 Security Analysis with Decryption Errors

We first give the explanation that error-prone encryptions combined with transforms will be immune to the decryption errors attack, and then check the chosen ciphertext security of both transforms.

Thwarting Decryption Errors. The above transforms could be combined with error-prone PKE Π' with sparse distributions of errors, and then decrease the probability of finding errors by the attacker. For the sake of analysis, we let Π' has an error probability α, where the probability is over the choice of (M, R) message and randomness pair. Thus, we could define the error probability of M and R as α where α is negligible when the message and randomness is chosen randomly [3].

Since we are going to reduce the successful probability of attacker to find such a "bad" pair that leads to the DCA attack, we just analyze that probability before and after the transform is applied. We start with the 3-round OAEP transform. Given some message randomness pair, we at first modify the message gradually, and change the randomness r due to the relation of paddings. The goal of the attacker is to control the input of Π', i.e. $(s||t)$, but only has access to m. But this is obviously difficult, because F, G, H are random oracles, the value passing through them becomes randomly. Therefore, the best strategy of the attacker, rather than randomly guessing, is to query the random oracle and check all the answers to find appropriate $(s||t)$ and their corresponding (m, r). We assume the queries to three random oracles are q_F, q_G, q_H respectively.

By analyzing 3-round OAEP, we have the following fact. Let us first see the t part of the input of Π', we have $t = r \oplus G(w)$, where $w = (m||0^{k_0}) \oplus F(r)$. For searching the appropriate t, the attacker should use three lists to record the query and answer to F, G and H, such as $(r_1, ..., r_q\ )$, $(f_1, ..., f_q\ )$ of oracle F, $(w_1, ..., w_q\ )$, $(g_1, ..., g_q\ )$ of oracle G and $(t_1, ..., t_q\ )$, $(h_1, ..., h_q\ )$ of oracle H. Then, we try to find some "bad" t, where there are corresponding g and h in the lists, s.t. $t = r_i \oplus g_j$, and further choose $s = w_j \oplus h_k$, thus get a candidate pair of (m, r) which leads to an fault decryption. Since the error probability is assumed as α, we can compute the possible probability is bounded by $\alpha \cdot q_G q_H$. From another view, error probability is fixed at first, then all $s||t$ candidates should fit the requirement of $m||0^{k_0}$. Since $m||0^{k_0} = f_i \oplus w_j$, after querying both oracles, $1 - (1 - 1/2^{k_0})^{q\ \ q} \approx q_F q_G / 2^{k_0}$.

According to our analysis, the 3-round OAEP could decrease the error probability occurred $\Pr[Error]_1$ at most

$$\Pr[Error]_1 \leq \alpha \cdot q_G q_H + \frac{q_F q_G}{2^{k_0}}$$

Note that we are able to adjust the parameter to let the above probability tiny enough.

On the probabilistic 3-round OAEP, the situation is likely, except that one more hash oracle is introduced. Hence we have to count this probability as well. Besides similar analysis as above, the attacker has to make the chosen message

[3] Attacker will use a more smart strategy to choose its target.

passing the check by H' hash function. Since it is querying all the possible value to the H' oracle, this probability will be bounded by $q_{H'}/2^{k'}$. The total bound is as the following.

$$\Pr[Error]_2 \leq \alpha \cdot q_G q_H + \frac{q_{H'}}{2^{k'}}$$

On the IND-CCA Security. The proof that 3-round OAEP is fulfilling with IND-CCA seems quite natural to understand after the work by [14]. We will refer to their paper. On the second transform, it seems that this modified version has not been proved yet, although another probabilistic version has been studied in [15], without achieving the exact IND-CCA security.

We just describe the proof strategy of the second transform, and refer to the full version of this paper for detailed proof. The original 3-round OAEP is provable with deterministic one-way permutation, however, it is not possibly to be proved with the probabilistic encryption. The reason is that for probabilistic encryption, even the input message is the same, the ciphertext could be different due to distinct randomness used. Thus for the oracle simulation process, the exact IND-CCA security (definition 4) will be lost easily. We apply one random oracle to check the validity of message and randomness pair, then all the possible pairs from attacker must be contained in the queries of this H' oracle (otherwise, we just simply reject the request). The above problem of original 3-round OAEP can be overcome. The security of this transform bases on OW-PCA (definition 3) security.

Remark. From above analysis, it may be raised a question that why not just use more hash functions and build more rounds. It is obvious that they are redundant and expensive. More importantly, the 2-round OAEP has been proved insecure against decryption errors attack [16], thus we naturally conclude that 3-round is the best efficient in the presence of decryption errors, from the viewpoint of bandwidth.

5 Conclusion

In this paper (extended abstract), we explain that existing generic transform is suitable for PKEs without or with imperfect decryption, and propose a new variant as well. We present the error probability bound, which decreases much capability of attackers to control the message and ciphertext pair in the CCA attack, and finally contribute to immunize the decryption failure for error-prone PKEs.

References

1. M. Ajtai and C. Dwork, "A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence", in *STOC 1997*, pp. 284-293, 1997.
2. M. Bellare and P. Rogaway, "Random Oracles Are Practical: A paradigm for designing efficient protocols", in *Proc. First Annual Conference on Computer and Communications Security*, ACM, 1993.

3. M. Bellare and P. Rogaway, “Optimal Asymmetric Encryption - How to Encrypt with RSA”, in *Eurocrypt’94*, LNCS 950, Springer-Verlag, pp. 92-111, 1995.
4. J.S. Coron, M. Joye, D. Naccache and P. Paillier, “Universal padding schemes for RSA”, in *Crypto’02*, LNCS 2442, Springer-Verlag, pp. 226-241, 2002.
5. C. Dwork, M. Naor and O. Reingold, “Immunizing Encryption Schemes from Decryption Errors”, in *Eurocrypt’04*, LNCS 3027, Springer-Verlag, pp. 342-360, 2004.
6. E. Fujisaki and T. Okamoto, “How to Enhance the Security of Public-Key Encryption at Minimum Cost”, in *PKC’99*, LNCS 1560, Springer-Verlag, pp. 53-68, 1999.
7. O. Goldreich, S. Goldwasser and S. Halevi, “Eliminating Decryption Errors in the Ajtai-Dwork Cryptosystem”, in *Crypto’97*, LNCS 1294, Springer-Verlag, pp. 105-111, 1997.
8. S. Goldwasser and S. Micali, “Probabilistic encryption”, *Journal of Computer Security*, 28:270–299, 1984.
9. J. Hoffstein, J. Pipher and J.H. Silverman, “NTRU: A Ring-Based Public Key Cryptosystem”, in *ANTS’98*, LNCS 1423, Springer-Verlag, pp. 267-288, 1998.
10. N. Howgrave-Graham, P. Nguyen, D. Pointcheval, J. Proos, J.H. Silverman, A. Singer and W. Whyte, “The Impact of Decryption Failures on the Security of NTRU Encryption”, in *Crypto’03*, LNCS 2729, Springer-Verlag, pp. 226-246, 2003.
11. M. Naor and M. Yung, “Public-key Cryptosystems Provably Secure against Chosen Ciphertext Attacks”, *STOC 1990*, pp. 427-437, 1990.
12. P. Nguyen, “Cryptanalysis of the Goldreich-Goldwasser-Halevi Cryptosystem from Crypto ’97”, in *Crypto’99*, LNCS 1666, Springer-Verlag, pp. 288-304, 1999.
13. T. Okamoto and D. Pointcheval, “REACT: Rapid Enhanced-Security Asymmetric Cryptosystem Transform”, in *CT-RSA’01*, LNCS 2020, Springer-Verlag, pp 159-175, 2001.
14. D.H. Phan and D. Pointcheval, “Chosen-Ciphertext Security without Redundancy”, in *Asiacrypt’03*, LNCS 2894, Springer-Verlag, pp. 1-18, 2003.
15. D.H. Phan and D. Pointcheval, “OAEP 3-Round: A Generic and Secure Asymmetric Encryption Padding”, in *Asiacrypt’04*, LNCS 3329, pages 63-77, Springer-Verlag, 2004.
16. J. Proos, “Imperfect Decryption and an Attack on the NTRU Encryption Scheme”, Cryptology ePrint Archive: Report 2003/002.
17. C. Rackoff and D. Simon, “Non-interactive Zero-knowledge Proof of Knowledge and Chosen Ciphertext Attack”, in *Crypto’91*, LNCS 576, Springer-Verlag, pp. 433-444, 1992.
18. P.W. Shor, “Algorithms for Quantum Computation: Discrete Logarithms and Factoring”, *FOCS 1994*, pp. 124-134, 1994.

Applying Fujisaki-Okamoto to Identity-Based Encryption

Peng Yang[1], Takashi Kitagawa[2], Goichiro Hanaoka[2], Rui Zhang[1],
Kanta Matsuura[1], and Hideki Imai[1,2]

[1] Institute of Industrial Science, University of Tokyo,
4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, Japan
{pengyang, kanta, imai}@iis.u-tokyo.ac.jp,
zhang@imailab.iis.u-tokyo.ac.jp

[2] Research Centre for Information Security (RCIS),
National Institute of Advanced Industrial Science and Technology (AIST),
1102 Akihabara-Daibiru, 1-18-13 Sotokanda, Chiyoda-ku, Tokyo 101-0021, Japan
{t-kitagawa, hanaoka-goichiro}@aist.go.jp

Abstract. The Fujisaki-Okamoto (FO) conversion is widely known to be able to generically convert a weak public key encryption scheme, say one-way against chosen plaintext attacks (OW-CPA), to a strong one, namely, indistinguishable against adaptive chosen ciphertext attacks (IND-CCA). It is not known that if the same holds for identity-based encryption (IBE) schemes, though many IBE and variant schemes are in fact specifically using the FO conversion. In this paper, we investigate this issue and confirm that the FO conversion is generically effective also in the IBE case. However, straightforward application of the FO conversion only leads to an IBE scheme with a loose (but polynomial) reduction. We then propose a simple modification to the FO conversion, which results in considerably more efficient security reduction.

1 Introduction

Background. Identity based encryption (IBE) [11] is a public key encryption scheme where the encryption public key can be an arbitrarily string, such as the recipient's identity, thus the distribution of public key certificates can be avoided for an IBE scheme. This was first motivated by applications to encrypt emails under the recipient's email address, however, it found more applications ever since, e.g. [8, 4].

It has been shown [1, 7] that the strongest security notion for IBE is *indistinguishability against adaptive chosen ID and adaptive chosen ciphertext attacks* (IND-ID-CCA). Nevertheless, many IBE schemes, other than (IND-ID-CCA), first build a "basic scheme" which is *one-way against adaptive chosen ID and chosen plaintext attacks* (OW-ID-CPA), then *specifically* use the famous Fujisaki-Okamoto (FO) conversion [6] to upgrade the basic scheme to a scheme with IND-ID-CCA security. However, it is still unknown whether the FO conversion can *generically* upgrade OW-ID-CPA security to IND-ID-CCA security.

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 183–192, 2006.

It is crucial to note that the FO conversion is a generic conversion to enhance a public key encryption scheme with security of *one-wayness under chosen plaintext attacks* (OW-CPA) to security of *indistinguishability against adaptive chosen ciphertext attacks* (IND-CCA) [10] in the random oracle model. Many practical PKE schemes are based on it.

Our Contributions. Our contributions are three-fold:
First, we investigate the generic security of the IBE obtained by applying the FO conversion to an underlying OW-ID-CPA secure IBE and confirm the IND-ID-CCA security of the IBE can be polynomially reduced to the OW-ID-CPA security of the underlying IBE.

Additionally, we find that the straightforward application of the FO conversion yields a significantly inefficient reduction cost. To be more precise, the simulator's time complexity is more than $2^{100} (> 2^{80})$ times re-encryption computation (in addition to an IND-ID-CCA adversary's running time).

Finally, we slightly modify the FO conversion so that the simulator's time complexity is reduced to be $2^{60} (< 2^{80})$ times re-encryption computation (in addition to an adversary's running time) which can be dealt with in practice.

2 Preliminary

In this section, we present the definitions of IBE, OW-ID-CPA, IND-ID-CCA and γ-uniformity.

ID-Based Encryption. ID-Based encryption (IBE) [11] is a public key encryption scheme where the encryption public keys can be arbitrary strings. It is formally defined as follows:

Definition 1 (ID-Based Encryption). *Formally, an identity-based encryption (IBE) scheme $\Pi = \{\mathcal{S}, \mathcal{X}, \mathcal{E}, \mathcal{D}\}$ consists of the four algorithms.*

- $\mathcal{S}$, the setup algorithm, takes security parameter $k \in \mathbf{Z}$ as input, and outputs system parameters params and the master-key $\mathsf{master\text{-}key}$. $\mathcal{S}$ is a probabilistic algorithm. params consists of descriptions of a finite message space MSPC, and a finite ciphertext space CSPC.
- $\mathcal{X}$, the extract algorithm, takes as inputs params, $\mathsf{master\text{-}key}$ and an arbitrary $\mathsf{ID} \in \{0,1\}^*$, and outputs a private key d. ID is an arbitrary string and used as a public key. d is the corresponding private key(decryption key). This algorithm extracts a private key corresponding to ID.
- $\mathcal{E}$, the encryption algorithm, takes as input params, ID and $M \in \mathsf{MSPC}$. Let $\mathsf{COIN}(k) \subseteq \{0,1\}^*$ be a finite set. $\mathcal{E}$ chooses a random string $coin \in \mathsf{COIN}(k)$ and outputs a ciphertext $C \in \mathsf{CSPC}$. $\mathcal{E}$ is a probabilistic algorithm. We denote the result of running this algorithm $\mathcal{E}(\mathsf{params}, \mathsf{ID}, M; coin)$.
- $\mathcal{D}$, the decryption algorithm, takes as input params, $C \in \mathsf{CSPC}$ and a private key d, and outputs $M \in \mathsf{MSPC}$. The algorithm decrypts a ciphertext C using the private key d.

These algorithms must satisfy the standard consistency constraint,

$$\forall M \in \mathsf{MSPC},\ \mathcal{D}(\mathsf{params}, d, C) = M \text{ where } C = \mathcal{E}(\mathsf{params}, \mathsf{ID}, M).$$

One-Way Identity-Based Encryption. A notion of security called one-way encryption(OWE) is an even weaker notion. Roughly speaking, this notion means that when given the encryption of a random plaintext the adversary cannot produce the plaintext in its entirety. Originally OWE is defined for standard public key encryption schemes. Boneh and Franklin [3] extended the definition of OWE for IBE schemes. An IBE scheme is an one-way encryption scheme if no polynomial adversary $\mathcal{A}$ has a non-negligible advantage against the challenger in the following game:

Setup: The challenger takes a security parameter k and runs the setup algorithm $\mathcal{S}$. It gives the adversary the resulting system parameters params. It keeps the $\mathsf{master\text{-}key}$ to itself.

Phase 1: The adversary issues private key extraction queries $\mathsf{ID}_1, \ldots, \mathsf{ID}_m$. The challenger responds by running $\mathcal{X}$ to extract the private key d_i corresponding to the public key ID_i. It sends d_i to the adversary. These queries may be asked adaptively.

Challenge: Once the adversary decides that Phase 1 is over it outputs a public key $\mathsf{ID} \notin \{\mathsf{ID}_1, \ldots, \mathsf{ID}_m\}$ on which it wishes to be challenged. The challenger picks a random $M \in \mathsf{MSPC}$ and encrypts M using ID as the public key. It then sends the resulting ciphertext C to the adversary.

Phase 2: The adversary issues more extraction queries $\mathsf{ID}_{m+1}, \ldots, \mathsf{ID}_n$. The only constraint is that $\mathsf{ID}_i \neq \mathsf{ID}$. The challenger responds as in Phase 1.

Guess: Finally, the adversary outputs a guess $M' \in \mathsf{MSPC}$ and wins the game if $M = M'$.

We refer to such an adversary $\mathcal{A}$ as an $\mathsf{OW\text{-}ID\text{-}CPA}$ adversary. $\mathcal{A}$'s advantage in attacking the scheme is defined as: $Adv_{\mathcal{A}}(k) = \Pr[M = M']$. The probability is taken over the random bits used by the challenger and the adversary.

Definition 2 ($\mathsf{OW\text{-}ID\text{-}CPA}$). *We say that an IBE scheme is secure in the sense of* $\mathsf{OW\text{-}ID\text{-}CPA}$ *if* $Adv_{\mathcal{A}}$ *is negligible for any polynomial time algorithm* $\mathcal{A}$.

Chosen Ciphertext Security. Boneh and Franklin [3] defined chosen ciphertext security for IBE systems. In their model, security for an IBE system is defined by the following $\mathsf{IND\text{-}ID\text{-}CCA}$ game:

Setup: The challenger takes a security parameter k and runs setup algorithm $\mathcal{S}$. It gives the adversary the resulting system parameters params and keeps the $\mathsf{master\text{-}key}$ to itself.

Phase 1: The adversary issues queries $q_1, \cdots, q_m$ where query q_i is one of:

- Extraction query $\langle \mathsf{ID}_i \rangle$. The challenger responds by running algorithm $\mathcal{E}$ to generate decryption key d_i which corresponds to the public key $\langle \mathsf{ID}_i \rangle$. It sends d_i to the adversary.

- Decryption query $\langle \mathsf{ID}_i, C_i \rangle$. The challenger responds by running algorithm $\mathcal{E}$ to generate the decryption key d_i corresponding to the public key $\langle \mathsf{ID}_i \rangle$. Then it runs algorithm $\mathcal{D}$ to decrypt the ciphertext C_i using d_i. It sends the adversary the resulting plaintext.

The query may be asked adaptively, that is, each query q_i may depends on the replies to $q_1, \cdots, q_{i-1}$.

Challenge: Once the adversary decides that Phase 1 is over it outputs two equal length plaintext $M_0, M_1 \in \mathsf{MSPC}$ and an ID on which it wishes to be challenged. The only constraint is that the ID did not appear in any Extraction query in Phase 1. The challenger picks a random bit $b \in \{0,1\}$ and sets $C = \mathcal{E}(\mathsf{params}, \mathsf{ID}, M_b)$. It sends C to the adversary.

Phase 2: The adversary issues more queries $q_{m+1}, \cdots, q_n$ where query q_i is one of:

- Extraction query $\langle \mathsf{ID}_i \rangle$ where $\mathsf{ID}_i \neq \mathsf{ID}$. The challenger responds as in Phase 1.
- Decryption query $\langle \mathsf{ID}_i, C_i \rangle$ where $\langle \mathsf{ID}_i, C_i \rangle \neq \langle \mathsf{ID}, C \rangle$. The challenger responds as in Phase 1.

These queries may be asked adaptively as in Phase 1.

Guess: Finally, the adversary outputs a guess $b' \in \{0,1\}$ and wins the game if $b' = b$.

We refer to such an adversary $\mathcal{A}$ as an $\mathsf{IND\text{-}ID\text{-}CCA}$ adversary. An advantage of an $\mathsf{IND\text{-}ID\text{-}CCA}$ adversary is defined as follows: $Adv_{\mathcal{A}}(k) = |\Pr[b = b'] - \frac{1}{2}|$. The probability is taken over the random bits used by the challenger and the adversary.

Definition 3 ($\mathsf{IND\text{-}ID\text{-}CCA}$). *We say that an IBE system is secure in sense of* $\mathsf{IND\text{-}ID\text{-}CCA}$ *if $Adv_{\mathcal{A}}$ is negligible for any polynomial time algorithm $\mathcal{A}$.*

γ-Uniformity. A property γ-uniformity is originally defined for conventional public key encryption schemes [6]. Here, we define γ-uniformity for IBE schemes.

Definition 4 (γ-uniformity). *Let $\Pi = \{\mathcal{S}, \mathcal{X}, \mathcal{E}, \mathcal{D}\}$ be an IBE scheme. For a given* $\mathsf{ID} \in \{0,1\}^*$*, the corresponding decryption key d, $x \in \mathsf{MSPC}$ and $y \in \mathsf{CSPC}$, define*

$$\gamma(x, y) = \Pr[h \leftarrow_R \mathsf{COIN}(k) : y = \mathcal{E}(\mathsf{params}, \mathsf{ID}, x; h)].$$

We say that Π is γ-uniform, if, for any $\mathsf{ID} \in \{0,1\}^*$*, any $x \in \mathsf{MSPC}$ and any $y \in \mathsf{CSPC}$, $\gamma(x, y) \leq \gamma$.*

3 Fujisaki-Okamoto Conversion for IBE Schemes

In this section, we discuss the security of the FO conversion for $\mathsf{OW\text{-}ID\text{-}CPA}$ secure IBE. As far as we know, this is the first formal analysis which proves that FO generically converts any $\mathsf{OW\text{-}ID\text{-}CPA}$ secure IBE into an $\mathsf{IND\text{-}ID\text{-}CCA}$ secure IBE. We also give an observation that the straightforward application of FO to achieve a strong security is insufficient.

Straightforward Application of FO. Let $\Pi = \{\mathcal{S}, \mathcal{X}, \mathcal{E}, \mathcal{D}\}$ be an $\mathsf{OW\text{-}ID\text{-}CPA}$ IBE. Then, we can construct another IBE $\Pi' = \{\mathcal{S}', \mathcal{X}', \mathcal{E}', \mathcal{D}'\}$ as follows:

Let l_1 be a bit length of a plaintext of Π, l_2 be a bit length of a plaintext of Π' and $\mathsf{COIN}(k)$ be Π's coin-flipping space.

- $\mathcal{S}'$, the setup algorithm. It is as $\mathcal{S}$. In addition, we pick two hash functions $G : \{0,1\}^{l_1} \times \{0,1\}^{l_2} \to \mathsf{COIN}(k)$ and $H : \{0,1\}^{l_1} \to \{0,1\}^{l_2}$.
- $\mathcal{X}'$, the extraction algorithm. It is as $\mathcal{X}$.
- $\mathcal{E}'$, the encryption algorithm. It is defined as follows:

$$\mathcal{E}'(\mathsf{params}, \mathsf{ID}, M; \sigma) = \mathcal{E}\big(\mathsf{params}, \mathsf{ID}, \sigma; G(\sigma, M)\big) \| H(\sigma) \oplus M$$

- $\mathcal{D}'$, the decryption algorithm. Let $C = C_1 \| C_2$ be a ciphertext to decrypt. Algorithm $\mathcal{D}'$ works in the following steps:
  1. Computes $\mathcal{D}(\mathsf{params}, d, C_1) = \sigma$.
  2. Computes $H(\sigma) \oplus C_2 = M$
  3. Sets $r = G(\sigma, M)$. Tests that $\mathcal{E}(\mathsf{params}, \mathsf{ID}, \sigma; r) = C_1$. If not, outputs "reject".
  4. Outputs M as the decryption of C

Theorem 1. *Suppose the hash functions G and H are random oracles and Π is a γ-uniform IBE encryption scheme. Let $\mathcal{B}$ be an* IND-ID-CCA *adversary which has advantage $\epsilon(k)$ against Π' and it runs in time at most $t(k)$. Suppose $\mathcal{B}$ makes at most q_H H queries, q_G G queries, q_E Extraction queries and q_D Decryption queries. Suppose that running time of $\mathcal{E}$ is at most τ. Then there is an* OW-ID-CPA *adversary $\mathcal{A}$ which has advantage at least $\frac{1}{q \quad +q}\big(2\epsilon(k) - q_D\gamma - q_D/2^{l_2}\big)$ against Π. Its running time is $t(k) + q_G \cdot q_D \cdot \tau$.*

Proof. We show how to construct adversary $\mathcal{A}$ by using adversary $\mathcal{B}$ as an oracle. The challenger starts an OW-ID-CPA game by executing $\mathcal{S}$ and generates params and $\mathsf{master\text{-}key}$. The $\mathsf{master\text{-}key}$ is kept secret by the challenger. $\mathcal{A}$ works by interacting with $\mathcal{B}$ in an IND-ID-CCA game as follows:

Setup: $\mathcal{A}$ gives params to $\mathcal{B}$.

Responses to G-Queries: $\mathcal{A}$ maintains a list of tuples $\langle \sigma_i, M_i, g_i \rangle$ as explained below. We refer to this list as the G^{list}. The list is initially empty. When $\mathcal{B}$ queries $G(\sigma_i, M_i)$, $\mathcal{A}$ responds as follows:
1. If the query σ_i and M_i already appears on the G^{list} in a tuple $\langle \sigma_i, M_i, g_i \rangle$ then $\mathcal{A}$ responds with $G(\sigma_i, M_i) = g_i$.
2. Otherwise, $\mathcal{A}$ picks a random element g_i from $\mathsf{COIN}(k)$ of Π.
3. $\mathcal{A}$ adds the tuple $\langle \sigma_i, M_i, g_i \rangle$ to the G^{list} and returns g_i.

Responses to H-Queries: $\mathcal{A}$ maintains a list of tuples $\langle M_i, h_i \rangle$ to respond the queries. We refer to this list as H^{list}. The list is initially empty. When $\mathcal{B}$ queries $H(M_i)$, $\mathcal{A}$ responds as following:
1. If the query M_i already appears on the H^{list} in a tuple $\langle M_i, h_i \rangle$ then $\mathcal{A}$ responds with $H(M_i) = h_i$.
2. Otherwise, $\mathcal{A}$ picks a string h_i from $\{0,1\}^{l_2}$ randomly.
3. $\mathcal{A}$ adds the tuple $\langle M_i, h_i \rangle$ to the H^{list} and returns h_i.

Responses to Extraction Queries: Let $\langle \mathsf{ID}_i \rangle$ be an Extraction query issued by $\mathcal{B}$. $\mathcal{A}$ inputs $\langle \mathsf{ID}_i \rangle$ to its own extraction oracle and gets the corresponding decryption key d_i. $\mathcal{A}$ passes d_i to $\mathcal{B}$ as the answer of the query.

Responses to Decryption Queries: Let $\langle \mathsf{ID}_i, C_i \rangle$ be a Decryption query issued by $\mathcal{B}$. $\mathcal{A}$ responds as follows:

1. Find a pair of tuples $\langle \sigma, M, g \rangle$ and $\langle \sigma, h \rangle$ from the G^{list} and H^{list}, respectively, such that $\mathcal{E}(\mathsf{params}, \mathsf{ID}_i, \sigma; g) \| h \oplus M_j = C_i$.
2. Outputs M if there exists such a pair of tuples, or outputs "reject" otherwise.

Challenge: Once $\mathcal{B}$ decides that Phase 1 is over it outputs a public key ID and two messages M_0, M_1 on which it wishes to be challenged. $\mathcal{A}$ sends ID to the challenger and receives a ciphertext C. Then, $\mathcal{A}$ generates $C_{ch1} \| C_{ch2}$ where $C_{ch1} = C$ and C_{ch2} is a random string whose length is l_2. $\mathcal{A}$ gives $C_{ch1} \| C_{ch2}$ as the challenge to $\mathcal{B}$.

Guess: Once $\mathcal{B}$ decides that Phase 2 is over it outputs a guess b'.

After $\mathcal{B}$ outputs the guess b', $\mathcal{A}$ chooses a tuple $\langle \sigma, M, g \rangle$ or $\langle \sigma, h \rangle$ from the G^{list} or the H^{list}, respectively. Then, $\mathcal{A}$ outputs σ in the tuple as the answer of the $\mathsf{OW\text{-}ID\text{-}CPA}$ game.

We first define the following three events:

SuccB the event that $\mathcal{B}$ wins the $\mathsf{IND\text{-}ID\text{-}CCA}$ game.

AskB the event that $\mathcal{B}$ asks a query for $G(\mathcal{D}(\mathsf{params}, d, C_{ch1}), *)$ or $H(\mathcal{D}(\mathsf{params}, d, C_{ch1}))$ at some point during the game, where $d := \mathcal{X}(\mathsf{params}, \mathsf{master\text{-}key}, \mathsf{ID})$ and $*$ denotes any l_2-bit string.

Fail the event that the simulation fails before $\mathcal{B}$ submits a query for $G(\mathcal{D}(\mathsf{params}, d, C_{ch1}), *)$ or $H(\mathcal{D}(\mathsf{params}, d, C_{ch1}))$.

Then, we have that

$$\Pr[\mathsf{SuccB} | \neg\mathsf{Fail}] \cdot \Pr[\neg\mathsf{Fail}] \geq \epsilon(k) + \frac{1}{2} - \Pr[\mathsf{Fail}].$$

Since $\Pr[\mathsf{SuccB} | \neg\mathsf{Fail}, \neg\mathsf{AskB}] = 1/2$, we also have

$$\begin{aligned}\Pr[\mathsf{SuccB} | \neg\mathsf{Fail}] &= \Pr[\mathsf{SuccB} | \neg\mathsf{Fail} \wedge \mathsf{AskB}] \cdot \Pr[\mathsf{AskB}] + \frac{1}{2}\Big(1 - \Pr[\mathsf{AskB}]\Big) \\ &\leq \frac{1}{2}\Pr[\mathsf{AskB}] + \frac{1}{2}.\end{aligned}$$

Hence, we have that

$$\Big(\frac{1}{2}\Pr[\mathsf{AskB}] + \frac{1}{2}\Big) \cdot \Pr[\neg\mathsf{Fail}] \geq \epsilon(k) + \frac{1}{2} - \Pr[\mathsf{Fail}],$$

and therefore,

$$\Pr[\mathsf{AskB}] \geq 2\epsilon(k) - \Pr[\mathsf{Fail}].$$

Next, we estimate $\Pr[\mathsf{Fail}]$. The event Fail occurs only when either

Case 1. $\mathcal{B}$ submits a Decryption query $\langle \mathsf{ID}, C_1 \| H(\sigma) \oplus M \rangle$ such that $C_1 = \mathcal{E}(\mathsf{params}, \mathsf{ID}, \sigma; G(\sigma, M))$ without asking $G(\sigma, M)$, or

Case 2. $\mathcal{B}$ submits a Decryption query $\langle \mathsf{ID}, \mathcal{E}(\mathsf{params}, \mathsf{ID}, \sigma; G(\sigma, M)) \| C_2 \rangle$ such that $C_2 = H(\sigma) \oplus M$ without asking $H(\sigma)$.

Case 1 and **2** happen with probability at most γ and $1/2^{l_2}$, respectively, and therefore, we have that $\Pr[\mathsf{Fail}] \leq 1 - (1 - \gamma - 1/2^{l_2})^q$.

Hence, we have that

$$\begin{aligned} Adv_{\mathcal{A}}(k) &\geq \frac{1}{q_G + q_H} \Pr[\mathsf{AskB}] \\ &\geq \frac{1}{q_G + q_H} \left(2\epsilon(k) - \left(1 - \left(1 - \gamma - \frac{1}{2^{l_2}} \right)^q \right) \right) \\ &\simeq \frac{1}{q_G + q_H} \left(2\epsilon(k) - q_D \gamma - \frac{q_D}{2^{l_2}} \right). \end{aligned}$$

Finally, we estimate $\mathcal{A}$'s running time. Since in addition to $\mathcal{B}$'s running time, $\mathcal{A}$ has to run $\mathcal{E}$ for q_G times for responding to each Decryption query, $\mathcal{A}$'s running time is estimated as $t(k) + q_G \cdot q_D \cdot \tau$. □

Discussion: Running Time of $\mathcal{A}$. As shown in Theorem 1, there exists a polynomial time reduction from $\mathcal{B}$ to $\mathcal{A}$, and consequently, any polynomial time adversary cannot break Π' in IND-ID-CCA sense if any polynomial time adversary cannot break Π in OW-ID-CPA sense. However, this result does not immediately imply that any realistic adversary cannot break Π' in IND-ID-CCA sense if any realistic adversary cannot break Π in OW-ID-CPA sense. Suppose that $\mathcal{A}$'s computational time is significantly larger than $\mathcal{B}$'s. Then, it might be still infeasible to break Π in practice even if $\mathcal{B}$ can break Π' in IND-ID-CCA sense. Bellare and Rogaway [2] proposed the notion of *exact security* for formally dealing with this issue.

Now, we focus on the running times of $\mathcal{A}$ and $\mathcal{B}$ (rather than their advantages). As in Theorem 1, $\mathcal{A}$'s running time is estimated as $t(k) + q_G \cdot q_D \cdot \tau$, where $t(k)$ denotes $\mathcal{B}$'s running time. This means that $\mathcal{A}$ has to run the encryption algorithm $\mathcal{E}$ for $q_G \cdot q_D$ times in addition to $\mathcal{B}$'s running time. Consequently, assuming that q_G and q_D are estimated as 2^{60} and 2^{40} respectively, $\mathcal{A}$ has to run $\mathcal{E}$ for 2^{100} times! (Notice that a Decryption query requires on-line computation while a G-query only requires off-line hash computation.) It is believed that more than 2^{80} operations are computationally infeasible in the real world, and therefore, $\mathcal{A}$ cannot break OW-ID-CPA security of Π in practice (even if $\mathcal{B}$ works in a practical time).

Hence, the above straightforward application of the FO conversion is insufficient for achieving a strong security. In the next section, we propose an improved version of the FO conversion for IBE, which provides an efficient simulator with less time complexity.

4 Modified Fujisaki-Okamoto for IBE Schemes

In this section, we propose a modified FO conversion with an improved reduction cost, i.e. the simulator needs shorter running time but still obtains the same advantage when compared with the simulator in the straightforward FO. The difference between our modification and the original FO is only that we take σ, M and ID as input to G instead of σ and M.

Basic Idea. The huge running time of $\mathcal{A}$ in Theorem 1 is caused by the following reason. In order to respond to a Decryption query $\langle \mathsf{ID}, C\rangle$, $\mathcal{A}$ has to find a pair of tuples from G^{list} and H^{list} such that its corresponding ciphertext with public key ID is identical to C. Since $\mathcal{A}$ does not know ID in advance, it is required to carry out re-encryption with public key ID for all tuples in G^{list} for every Decryption query. This results in $q_G \cdot q_D$ times of re-encryption operations. For solving this problem, we add ID as one of the inputs to G.

Modified FO Conversion. Let $\Pi = \{\mathcal{S}, \mathcal{X}, \mathcal{E}, \mathcal{D}\}$ be an IBE scheme which is secure in the sense of $\mathsf{OW\text{-}ID\text{-}CPA}$. We denote the new encryption scheme as $\Pi'' = \{\mathcal{S}'', \mathcal{X}'', \mathcal{E}'', \mathcal{D}''\}$. Let l_1 be a bit length of a plaintext of Π, l_2 be a bit length of a plaintext of Π'' and $\mathsf{COIN}(k)$ be Π's coin-flipping space.

- $\mathcal{S}''$, the setup algorithm. It is as $\mathcal{S}$. In addition we pick two hash functions $G : \{0,1\}^{l_1} \times \{0,1\}^{l_2} \times \{0,1\}^* \to \mathsf{COIN}(k)$ and $H : \{0,1\}^{l_1} \to \{0,1\}^{l_2}$.
- $\mathcal{X}''$, the extraction algorithm. It is as $\mathcal{X}$.
- $\mathcal{E}''$, the encryption algorithm. It takes system parameter params, public key $\mathsf{ID} \in \{0,1\}^*$, random coin $\sigma \in \{0,1\}^{l_1}$ and a message $M \in \{0,1\}^{l_2}$.
  It is defined as follows:

$$\mathcal{E}''(\mathsf{params}, \mathsf{ID}, \sigma, M) = \mathcal{E}\big(\mathsf{params}, \mathsf{ID}, \sigma; G(\sigma, M, \mathsf{ID})\big) \| H(\sigma) \oplus M$$

- $\mathcal{D}''$, the decryption algorithm. Let $C = C_1 \| C_2$ be a ciphertext to decrypt. This algorithm works in the following four steps:
  1. Computes $\mathcal{D}(\mathsf{params}, d, C_1) = \sigma$
  2. Computes $H(\sigma) \oplus C_2 = M$
  3. Sets $r = G(\sigma, M, \mathsf{ID})$. Test that $\mathcal{E}(\mathsf{params}, \mathsf{ID}, M; r) = C_1$. If not, outputs "reject".
  4. Outputs M as the decryption of C.

Theorem 2. *Suppose the hash functions G and H are random oracles and Π is γ-uniform IBE encryption scheme. Let $\mathcal{B}$ be an* $\mathsf{IND\text{-}ID\text{-}CCA}$ *adversary which has advantage $\epsilon(k)$ against Π'' and it runs in time at most $t(k)$. Suppose $\mathcal{B}$ makes at most q_G G-queries, q_H H-queries, q_E Extraction queries and q_D Decryption queries. Suppose that encrypting one message needs time τ. Then there is an* $\mathsf{OW\text{-}ID\text{-}CPA}$ *adversary $\mathcal{A}$ which has advantage at least $\frac{1}{q \quad +q}\big(2\epsilon(k) - q_D\gamma - q_D/2^{l_2}\big)$ against Π. Its running time is $t(k) + q_G \cdot \tau$*

Proof. To prove Theorem 2, almost same strategy as the proof of Theorem 1 can be used. That is, assuming $\mathsf{IND\text{-}ID\text{-}CCA}$ adversary $\mathcal{B}$ for Π'', constructing $\mathsf{OW\text{-}ID\text{-}CPA}$ adversary $\mathcal{A}$ for Π which uses $\mathcal{B}$ as an oracle.

There are two different points between the proof of Theorem 1 and 2. The points are how to answer G-queries and Decryption-queries in the $\mathsf{IND\text{-}ID\text{-}CCA}$ game between $\mathcal{A}$ and $\mathcal{B}$. Due to the space limitation, we describe only these different points.

Responses to G-Queries: $\mathcal{A}$ maintains a list of tuples $\langle \sigma_i, M_i, \mathsf{ID}_i, g_i, C_i \rangle$ as explained below. We refer to this list as the G^{list}. The list is initially empty. When $\mathcal{B}$ queries $G(\sigma_i, M_i, \mathsf{ID}_i)$, $\mathcal{A}$ responds as follows:

1. If the query σ_i, M_i and ID_i already appears on the G^{list} in a tuple $\langle \sigma_i, M_i, \mathsf{ID}_i, g_i, C_i \rangle$ then $\mathcal{A}$ responds with $G(\sigma_i, M_i, \mathsf{ID}_i) = a_i$.
2. Otherwise, $\mathcal{A}$ picks a random element g_i from $\mathsf{COIN}(k)$.
3. $\mathcal{A}$ generates a ciphertext $C_i = \mathcal{E}(\mathsf{params}, \mathsf{ID}_i, \sigma_i; g_i) \| H(\sigma_i) \oplus M_i$.
4. $\mathcal{A}$ adds the tuple $\langle \sigma_i, M_i, \mathsf{ID}_i, g_i, C_i \rangle$ to the G^{list} and responds to $\mathcal{B}$ with $G(\sigma_i, M_i, \mathsf{ID}_i) = g_i$.

Responses to Decryption Queries: Let $\langle \mathsf{ID}_i, C_i \rangle$ be a decryption query issued by $\mathcal{B}$. $\mathcal{A}$ responds this query in the following steps:

1. Finds a tuple $\langle \sigma_j, M_j, \mathsf{ID}_j, g_j, C_j \rangle$ from the G^{list} such that $\mathsf{ID}_i = \mathsf{ID}_j$ and $C_i = C_j$.
2. Outputs M_j if there exists such a tuple, or outputs "reject" otherwise.

After $\mathcal{B}$ outputs the guess b', $\mathcal{A}$ chooses a tuple $\langle \sigma, M, \mathsf{ID}, g, C \rangle$ or $\langle \sigma, h \rangle$ from the G^{list} or the H^{list} randomly and outputs σ in the tuple as the answer of the OW-ID-CPA game.

The advantage of $\mathcal{A}$ can be evaluate in the same way as in Theorem 1. So, we omit to describe the detail of the evaluation here.

Finally, we estimate $\mathcal{A}$'s running time. In addition to $\mathcal{B}$'s running time, $\mathcal{A}$ has to run $\mathcal{E}$ for q_G times to make the G^{list}. Thus, $\mathcal{A}$'s running time is estimated as $t(k) + q_G \cdot \tau$. □

Comparison. Here, we compare the running times of simulators for Π' and Π''. In the comparison, we especially focus on times to run the encryption algorithm $\mathcal{E}$ which is required for each simulation. It is believed that if a simulator has to run $\mathcal{E}$ for more than 2^{80} times, then it does not properly work in a realistic time. Now, we have that

$$\#_{\mathcal{E}}(\Pi')(= 2^{100}) \gg 2^{80} \gg \#_{\mathcal{E}}(\Pi'')(= 2^{60}),$$

where $\#_{\mathcal{E}}(\cdot)$ denotes the times to run $\mathcal{E}$ in the simulation. This implies that the running time of the simulator for Π'' is considered realistic, and on the other hand, that for Π' is not.

However, it should be noticed that existence of an adversary which can break Π'' does not always imply existence of another adversary which can break Π in practice. This is due to its non-tight reduction cost in terms of advantage, i.e. $\frac{2}{q \quad +q}\epsilon(k)$.

5 Conclusion

In this paper, we confirmed the generic security of FO conversion in IBE schemes, and investigated the fact that there exists a significantly inefficient reduction cost in the straightforward application, say, the additional 2^{100} times re-encryption computation. Under this circumstance, we modified FO and reduced the additional time down to 2^{60} times re-encryption computation.

Our discussion started from the OW-ID-CPA schemes, and we can also address the case starting from the IND-ID-CPA schemes. When we apply REACT [9] and the *PKC '99* version of FO [5] to IBE, some similar but more interesting results will appear. We will present them in the full version of this paper.

Acknowledgements

Rui Zhang is supported by a JSPS Fellowship. We appreciate the anonymous referees of AAECC for helpful comments and suggestions. And we are also grateful to Yang Cui for the constructive discussions.

References

1. N. Attrapadung, Y. Cui, G. Hanaoka, H. Imai, K. Matsuura, P. Yang, and R. Zhang. Relations among notions of security for identity based encryption schemes. Cryptology ePrint Archive, Report 2005/258, 2005. `http://eprint.iacr.org/2005/258`.
2. M. Bellare and P. Rogaway. The exact security of digital signatures - how to sign with rsa and rabin. In *Advances in Cryptology - EUROCRYPT '96*, volume 1070 of *LNCS*, pages 399–416. Springer, 1996.
3. D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In *Advances in Cryptology - CRYPTO '01*, volume 2139 of *LNCS*, pages 213–229. Springer, 2001.
4. R. Canetti, S. Halevi, and J. Katz. A forward-secure public-key encryption scheme. In *Advances in Cryptology - EUROCRYPT '03*, volume 2656 of *LNCS*, pages 255–271. Springer, 2003.
5. E. Fujisaki and T. Okamoto. How to enhance the security of public-key encryption at minimum cost. In *Public Key Cryptography - PKC '99*, volume 1560 of *LNCS*, pages 53–68. Springer, 1999.
6. E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Advances in Cryptology - CRYPTO '99*, volume 1666 of *LNCS*, pages 537–554. Springer, 1999.
7. D. Galindo and I. Hasuo. Security notions for identity based encryption. Cryptology ePrint Archive, Report 2005/253, 2005. `http://eprint.iacr.org/2005/253`.
8. C. Gentry and A. Silverberg. Hierarchical id-based cryptography. In *ASIACRYPT '02*, volume 2501 of *LNCS*, pages 548–566. Springer, 2002.
9. T. Okamoto and D. Pointcheval. React: Rapid enhanced-security asymmetric cryptosystem transform. In *Topics in Cryptology – CT-RSA '01*, volume 2020 of *Lecture Notes in Computer Science*, pages 159–174. Springer, 2001.
10. C. Rackoff and D.R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Advances in Cryptology - CRYPTO '91*, volume 576 of *LNCS*, pages 433–444. Springer, 1991.
11. A. Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology - CRYPTO '84*, volume 196 of *LNCS*, pages 47–53. Springer, 1985.

A Short Random Fingerprinting Code Against a Small Number of Pirates

Manabu Hagiwara[1], Goichiro Hanaoka[1], and Hideki Imai[1,2]

[1] Research Center for Information Security (RCIS),
Advanced Industrial Science and Technology (AIST),
Akihabara-Daibiru Room 1102,
1-18-13 Sotokanda, Chiyoda, Tokyo 101-0021, Japan
{hagiwara.hagiwara, hanaoka-goichiro, h-imai}@aist.go.jp
http://unit.aist.go.jp/rcis/

[2] Institute of Industrial Science, University of Tokyo,
4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, Japan
imai@iis.u-tokyo.ac.jp

Abstract. In this paper, we propose a variant of Tardos code which is practical for various applications against a small number of pirates. As an example of our results, for $c = 5$, the code length becomes only $1500 \log(1/\epsilon)$ bits while the conventional Tardos code requires $2500 \log(1/\epsilon)$ bits, where ϵ is a security parameter.

Furthermore our codes do not need a continuous distribution which is needed to construct the original Tardos codes. Our codes are based on a simple random variable drawn from a small set. It implies that it makes to implement and to perform a simulation extremely easier than the original one.

1 Introduction

1.1 Background

Pointing out a few of many information leakage incidents happened recently includes the exposure of credit card users' information incident by one of eminent credit card companies of the world which shocked us public how fragile the world's top most security system could be [1]. Other similar situations have also been seen in government and major corporate cases. Protecting information leakage is now a significantly important issue for the network society.

In many cases, such information leakage are performed by a few of persons involved in. Therefore, it is thought that the importance of the fingerprinting technology against a small number of pirates is improved. In fact, research for a fingerprinting code against a few pirates is one of hot topics for recent studies of information security. There are lots of papers, e.g. [2], [4], [5], [6], [7], [9] on the topic. In particular, Tardos [8] and Schaathun [6] give very short fingerprinting codes.

In this paper, we propose a variant of Tardos code [6] against a small number of pirates. As an example of our results, for $c = 5$, the code length becomes only

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 193–202, 2006.

$1500 \log(1/\epsilon)$ bits while the conventional Tardos code requires $2500 \log(1/\epsilon)$ bits, where ϵ is a security parameter. Though a continuous distribution is needed to construct the original Tardos code, our code is based on a discrete random variable drawn with a small set. For example, only 5 elements distribution is required to construct 5-secure codes. It is needless to say that to implement and to perform a simulation become wonderfully simple.

1.2 Our Contribution

In this paper, we propose a variant of Tardos code against a small number of pirates. As an example of our results, for $c = 5$, the code length becomes only $1500 \log(1/\epsilon)$ bits while the conventional Tardos code requires $2500 \log(1/\epsilon)$ bits, where ϵ is a security parameter. Though a continuous distribution is needed to construct the original Tardos code, our code is based on a discrete random variable drawn with a small set. For example, only 5 elements distribution is required to construct 5-secure codes. It is needless to say that to implement and to perform a simulation become wonderfully simple.

2 A Generalisation of Tardos Code with a Symmetric Probabilistic Variable

In this section, we set up our model and introduce how to construct our code.

2.1 The Model

Data Distribution. Let M be the number of total users. When data are distributed to users (buyers) U_i $(1 \leq i \leq M)$, a special user, we call him a "codeword distributor", embed 0 or 1 to the original data. The embedded data 0 or 1 must be encrypted not to distinguish them for users. The codeword distributor keep all of bits distributed to users. We call distributed data "codeword" and denote by c_i and denote its j-th element by $c_{i,j}$. We denote the length of our code by n.

Assumption for an Attack by Colluders. Some users challenge to change their data to distribute other people who is not regular. We call such a dishonest user a "pirate". And we call a data made by some pirates a "false codeword". We assume that an attack by pirates performs under the Marking Assumption [3]. It means that if all of the j-th element of the pirates are the same c_j then the j-th element of a false codeword must be c_j.

And we assume that all of elements of a false codeword is 0 or 1 i.e. any coalition of pirates cannot make any other symbols.

Tracing Algorithm. After a false codeword is found out, a codeword distributor makes a score for each users. Supposing an user's score is over the threshold Z, a codeword distributor considers that the user is a member of pirates. A way to make a score in detail is introduced with §2.3. A threshold parameter Z is obtained in Remark 1.

2.2 Construction of Our Code

For $0 < q < 1$, we denote P_q a random variable such that $P_q = 1$ with probability q or $P_q = 0$ with probability $1 - q$. Sometimes we denote P_q by q for simplicity. Let $\mathcal{P}$ be a random variable which outputs a random variable P_q $(0 < q < 1)$. In this paper, we call $\mathcal{P}$ a *Symmetric Random Variable (SRV)* if $\Pr[\mathcal{P} = p] = \Pr[\mathcal{P} = 1 - p]$.

A codeword distributor fixes a SRV $\mathcal{P}$ which will become a basis of the code. Before a codeword distributor constructs codewords for users, a codeword distributor takes out a random variable $p_j \in \mathcal{P}$ and keeps $p = (p_1, p_2, \ldots, p_n)$ in a memory.

For a user U_i, the codeword distributor determines an element $c_{i,j}$, according to the random variable p_j, of U_i's codeword $c_i = (c_{i,1}, c_{i,2}, \ldots, c_{i,n})$: i.e. $c_{i,j} = 0$ if the outcome of p_j is 1 or $c_{i,j} = 0$ if otherwise.

Our codes are based on one of three SRVs $\mathcal{P}_2, \mathcal{P}_3$ and $\mathcal{P}_5$ introduced in the next section.

2.3 How to Make a Score

Let $y = (y_1, y_2, \ldots, y_n)$ be a false codeword assembled by pirates.

A codeword distributor makes a score S_i for each user U_i by the following. Let $c_i = (c_{i,1}, c_{i,2}, \ldots, c_{i,n})$ be the U_i's codeword. For each position j, the codeword distributor adds $\sqrt{\frac{1-p}{p}}$ points to S_i if $c_{i,j} = 1$ and $y_j = 1$. The codeword distributor adds $-\sqrt{\frac{p}{1-p}}$ points to S_i if $c_{i,j} = 0$ and $y_j = 1$. Otherwise, $y_j = 0$, the codeword distributor adds zero point to S_i. In other words, the score S_i is formulated by:

$$S_i := \sum_{1 \le j \le M : y\ =1} \left\{ \sqrt{\frac{1 - p_j}{p_j}} \delta_{1,c\ ,} - \sqrt{\frac{p_j}{1 - p_j}} \delta_{0,c\ ,} \right\},$$

where δ is a delta function, i.e. $\delta_{x,y} = 1$ (if $x = y$) or $\delta_{x,y} = 0$ (otherwise).

3 Our Contribution: Our Proposal SRVs and Lengths

In this section, we propose three SRVs $\mathcal{P}_2, \mathcal{P}_5$ and $\mathcal{P}_5$ which our 2-secure code, 3-secure code and 5-secure code are base on respectively. We denote $\Pr[\mathcal{P} = p] = q$ by $(p, q) \in \mathcal{P}$. Then put

$$\mathcal{P}_2 := \{(1/2, 1)\},$$

$$\mathcal{P}_3 = \{((\sqrt{3} - 1)/2\sqrt{3}, 1/2), ((\sqrt{3} + 1)/2\sqrt{3}, 1/2)\},$$

and

$$\mathcal{P}_5 := \{(1/10, 0.29225), (1/4, 0.05925), (1/2, 0.2970), (3/4, 0.05925), (9/10, 0.29225)\}.$$

In §3.1 and §3.2, we compare the length of our codes based on $\mathcal{P}_2, \mathcal{P}_3$ and $\mathcal{P}_5$ with Tardos codes [8] and Schaathun codes [6].

3.1 Comparison with Tardos Codes i.e. $\epsilon_1 = \epsilon, \epsilon_2 = \epsilon^{c/4}$

The code length of the original Tardos codes ([8]) is given by

$$n = 100c^2 \log(1/\epsilon).$$

If parameters $\epsilon_1 = \epsilon, \epsilon_2 = \epsilon^{c/4}$ are specified and written about the formula above, it can express below:

$$n = \frac{200c^2}{3} \log(1/\epsilon_1) + \frac{400c}{3} \log(1/\epsilon_2).$$

Applied the same parameters $\epsilon_1 = \epsilon, \epsilon_2 = \epsilon^{c/4}$ to, then our codes associated to $\mathcal{P}_2, \mathcal{P}_3$ and $\mathcal{P}_5$ have the lengths $151.8497 \log(1/\epsilon)$, $517.4193 \log(1/\epsilon)$ and $1497.7922 \log(1/\epsilon)$ respectively. Thus our lengths are about 0.6-time length compared with original Tardos ones.

3.2 3-Secure Code: $\epsilon_1 = \epsilon/M, \epsilon_2 = \epsilon$, Where $\epsilon = 10^{-j}$ and $M = 2^i$

Next we compare our 3-secure code with Schaathun codes which are concatenated fingerprinting codes with scattering codes ([6]). If we assume that the number of buyers (users) is $M = 2^i$ and the security parameters are $\epsilon_1 = \epsilon/M, \epsilon_2 = \epsilon$ where $\epsilon = 10^{-j}$, then the length n of our code is formulated by:

$$n = i \times 311.9162 + j \times 1244.8242.$$

Table 1 shows that our codes are shorter than Schaathun codes under the parameters (i, j) listed below.

Table 1. Lengths of 3-secure codes of [6] and ours

(j, i)	$(25, 13)$	$(75, 15)$	$(11, 18)$	$(16, 18)$	$(53, 21)$	$(148, 40)$
[6]	57337	229369	110565	57330	114681	458745
Ours	35176	98041	19308	25532	72537	196711

4 Code Length

In this section, we analyze the length of our code based on a SRV (Theorem 1). Before beginning the analysis of the length, we introduce two following lemmas. The proofs will be performed in Appendix.

Lemma 1. *Let x_1 and r_1 be positive numbers such that $r_1 = (e_1^x - 1 - x)/x^2$. Let α be a positive number. If $\alpha\sqrt{(1-p)/p} < x_1$ holds for any $p \in \mathcal{P}$, then the probability that the score S of an innocent user will exceed the threshold Z is the following:*

$$\Pr[S > Z] < e^{r_1 \alpha^2 n - \alpha Z}.$$

Lemma 2. *Let S_l be the sum of the scores of l pirates who assembled a false codeword. Let $\mathcal{P}$ be a SRV. Put $R_{l,x} := \max\{0, \mathrm{Ex}[p^x(1-p)^{l-x}\{x\sqrt{(1-p)/p} - (l-x)\sqrt{p/(1-p)}\}]\}$, $\mathcal{R}_{l,\mathcal{P}} := \{lEX[(1-p)^{l-1/2}p^{1/2}] - \sum_{1\le x\le l-1}\binom{l}{x}R_{l,x}\}$ and $s := -r_2/b + \mathcal{R}_{l,P}$. Then*

$$\mathrm{Ex}_{p\in\mathcal{P},U}[e^{-\beta S}\ ] \le e^{-\beta sn}.$$

In this paper, we introduce two security parameters ϵ_1 and ϵ_2 to estimate the security level of our codes. The parameter ϵ_1 expresses the upper bound of the probability to accuse an innocent user and ϵ_2 expresses the upper bound of the probability not to detect one of pirates, either. Applying $r_1\alpha^2 n - \alpha Z = \log\epsilon_1$ and $-s\beta n + c\beta Z = \log\epsilon_2$ to Lemmas 1 and 2 respectively, we immediately obtain the following inequalities:

$$\Pr[S > Z] < e^{r_1\alpha^2 n - \alpha Z} = \epsilon_1, \tag{1}$$

$$\Pr[S_l < lZ] < e^{-s\beta n + c\beta Z} = \epsilon_2. \tag{2}$$

By two relations (1) and (2), we have the following equations:

$$\begin{pmatrix} n \\ Z \end{pmatrix} = \begin{pmatrix} r_1\alpha^2 & -\alpha \\ -s\beta & c\beta \end{pmatrix}^{-1} \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \end{pmatrix} = \frac{1}{r_1 c\alpha^2\beta - s\alpha\beta} \begin{pmatrix} c\beta & \alpha \\ s\beta & r\alpha^2 \end{pmatrix} \begin{pmatrix} \log\epsilon_1 \\ \log\epsilon_2 \end{pmatrix}$$

Observing the length n, we have the following formula:

$$n = \frac{c\beta\log(1/\epsilon_1) + \alpha\log(1/\epsilon_2)}{s\alpha\beta - r_1 c\alpha^2\beta}. \tag{3}$$

For ease, put $\alpha = 1/ac, \beta = 1/bc$. (Note that $a = 10$ and $b = 20$ have been used to construct Tardos codes). Then (3) can be re-formulated by

$$n = \frac{a^2c^2\log(1/\epsilon_1) + abc\log(1/\epsilon_2)}{sa - r_1} = \frac{a^2\log(1/\epsilon_1)}{sa - r_1}c^2 + \frac{ab\log(1/\epsilon_2)}{sa - r_1}c.$$

In order to shorten the length n, it is important to choose a, b, s, r_1 well. By Remark 3, 4, we have conditions:

$$s = -\frac{r_2}{b} + \mathcal{R}_{l,P},$$

$$r_1 = \frac{e^{x_1} - 1 - x_1}{x_1^2}, \quad r_2 = \frac{e^{x_2} - 1 - x_2}{x_2^2},$$

$$ax_1 \ge \sqrt{(1-p_0)/p_0}/c, \quad bx_2 \ge \sqrt{(1-p_0)/p_0}.$$

Now $p_0, c, \mathcal{R}_{l,\mathcal{P}}$ are given, so we put

$$\frac{e^{x_1} - 1 - x_1}{x_1} = \frac{\mathcal{R}_{l,\mathcal{P}}\sqrt{1-p_0}}{4c\sqrt{p_0}}, \quad \frac{e^{x_2} - 1 - x_2}{x_2} = \frac{\mathcal{R}_{l,\mathcal{P}}\sqrt{1-p_0}}{2\sqrt{p_0}}$$

and

$$b := \sqrt{\frac{1-p_0}{p_0}}\frac{1}{x_2}\left(= \frac{2r_2}{\mathcal{R}_{l,\mathcal{P}}}\right), \quad a := \sqrt{\frac{1-p_0}{p_0}}\frac{1}{x_1 c}\left(= \frac{4r_1}{\mathcal{R}_{l,\mathcal{P}}}\right).$$

Then we have the following theorem:

Theorem 1. *Let $\mathcal{P}$ be a SRV. Let c be the total number of pirates and l the total number of pirates that a false codeword was assembled by l (out of c) pirates . Put $\mathcal{R}_{l,\mathcal{P}} := l\mathrm{Ex}[(1-p)^{l-1/2}p^{1/2}] - \sum_{1\leq x\leq l-1}\binom{l}{x}R_{l,x}$. Let $p_0 \in \mathcal{P}$ be minimum. Then it is enough if the code length $n_{c,l,\mathcal{P}}$ satisfies*

$$\begin{aligned} n_{c,l,\mathcal{P}} &\geq 4c\frac{\sqrt{1-p_0}}{x_1\mathcal{R}_{l,\mathcal{P}}\sqrt{p_0}}\log(1/\epsilon_1) + 4c\frac{\sqrt{1-p_0}}{x_2\mathcal{R}_{l,\mathcal{P}}\sqrt{p_0}}\log(1/\epsilon_2) \\ &= \frac{1-p_0}{(e^{x_1}-1-x_1)p_0}\log(1/\epsilon_1) + \frac{2c(1-p_0)}{(e^{x_2}-1-x_2)p_0}, \end{aligned}$$

where $\frac{e^{\ 1}-1-x_1}{x_1} = \frac{\mathcal{R}_{\ ,\mathcal{P}}\sqrt{1-p_0}}{4c\sqrt{p_0}}$ and $\frac{e^{\ 2}-1-x_2}{x_2} = \frac{\mathcal{R}_{\ ,\mathcal{P}}\sqrt{1-p_0}}{2\sqrt{p_0}}$.

Remark 1. Then the threshold parameter Z associated to is

$$\begin{aligned} Z &= \frac{2\sqrt{1-p_0}}{x_1\sqrt{p_0}}\log(1/\epsilon_1) + \frac{\sqrt{1-p_0}}{x_2\sqrt{p_0}}\log(1/\epsilon_2) \\ &= \frac{\mathcal{R}_{l,\mathcal{P}}(1-p_0)}{2c(e^{x_1}-1-x_1)p_0}\log(1/\epsilon_1) + \frac{\mathcal{R}_{l,\mathcal{P}}(1-p_0)}{2(e^{x_2}-1-x_2)p_0}\log(1/\epsilon_2). \end{aligned}$$

5 c-Indistinguishable SRV

In this section, we introduce a notion "c-indistinguishable" to an SRV. In fact, our codes are based on a c-indistinguishable SRV.

We fix the total number of pirates and denote the number by c. Thus a false codeword will be assembled by at most c pirates.

We call an SRV $\mathcal{P}$ *c-indistinguishable* if $\sum_{1\leq x\leq l-1}\binom{l}{x}R_{l,x} = 0$ for any $2 \leq l \leq c$. We note that if $\mathcal{P}$ is c-indistinguishable then we have $\mathcal{R}_{l,\mathcal{P}} = l\mathrm{Ex}[(1-p)^{l-1/2}p^{1/2}]$.

Proposition 1. *Any SRV $\mathcal{P}$ is 2-indistinguishable.*

Proof. Since $c = 2$, it is enough to show that

$$R_{2,1} = \mathrm{Ex}[p^{0.5}(1-p)^{1.5} - p^{1.5}(1-p)^{0.5}] = 0.$$

Since $\mathcal{P}$ is a SRV, $(p,q) \in \mathcal{P}$ implies $(1-p,q) \in \mathcal{P}$. Hence $p^{0.5}(1-p)^{1.5} - p^{1.5}(1-p)^{0.5} = -((1-p)^{0.5}p^{1.5} - (1-p)^{0.5}p^{1.5})$. It follows that $R_{2,1} = 0$.

We give examples of c-indistinguishable SRV and calculate its associated length.

$\mathcal{P}_2$. Then $\mathcal{R}_{2,\mathcal{P}} = 1/2$, $p_0 = 1/2, x_1 = 0.12005$ and $x_2 = 0.76279$.
Thus

$$\begin{aligned} n &= 133.2796\log(1/\epsilon_1) + 37.1402\log(1/\epsilon_2), \\ Z &= 16.6600\log(1/\epsilon_1) + 2.3213\log(1/\epsilon_2). \end{aligned}$$

$\mathcal{P}_3$**.** There is no 3-indistinguishable SRV if $\mathcal{P}$ consists of one random variable i.e. $\mathcal{P} = \{(p), 1)\}$ and $\mathcal{P}_3$ is a unique 3-indistinguishable SRV $\mathcal{P}$ with two elements. Its length is

$$n = 450.7279 \log(1/\epsilon_1) + 89.8924 \log(1/\epsilon_2).$$

$\mathcal{P}_5$**.** In fact, $\mathcal{P}_5$ is 5-indistinguishable. Then $\mathcal{R}_{5,\mathcal{P}} = \mathcal{R}_{4,\mathcal{P}} = \mathcal{R}_{3,\mathcal{P}} = \mathcal{R}_{2,\mathcal{P}} =$ 0.375158 and

$$n = 1473.8529 \log(1/\epsilon_1) + 19.1515 \log(1/\epsilon_2).$$

References

1. Search keywords“customer”, “data” and “loss” on internet search engines.
2. A. Bark, G.R. Blakley, and G.A. Kabatiansky. Digital fingerprinting codes: Problem statements, constructions, identification of traitors. IEEE Trans. Inform. Theory, 49(4):852-865, April 2003.
3. D. Boneh and J. Shaw: Collusion-secure fingerprinting for digital data. IEEE Transactions of Information Theory 44 (1998), 480-491.
4. V. Dong Tô, R. Safavi-Naini, and Y. Wang: A 2-Secure Code with Efficient Tracing Algorithm. Lecture Notes in Computer Science, Vol. 2551. Springer-Verlag, Berlin Heidelberg New York (2002), pp. 149-163.
5. M. Fernandez and M. Soriano: Fingerprinting Concatenated Codes with Efficient Identification. Lecture Notes in Computer Science, Vol. 2433. Springer-Verlag, Berlin Heidelberg New York (2002), pp. 459-470.
6. H.G. Schaathun: Fighting three pirates with scattering codes. Information Theory, 2004. ISIT 2004. Proceedings. International Symposium on Volume, Issue, 27 June-2 July 2004 Page(s): 202-.
7. H.G. Schaathun M. Fermandez: Boneh-Shaw fingerprinting and soft decision decoding. IEEE ITSOC ITW 2005, Technical report no. 289 from Department of Informatics, UiB. January 2005.
8. G. Tardos: Optimal Probabilistic Fingerprint Codes. Proceedings of the 35th Annual ACM Symposium on Theory of Computing, 2003, pp. 116-125. Journal of the ACM, to appear.
9. J. Yoshida, K. Iwamura, H. Imai: A Coding Method for Collusion-Secure Watermark and Less Decline. SCIS'98.

A Appendix: Lemmas and Remarks

A.1 Formulas

Let l be a positive integer and x a variable. Let p be a positive number such that $0 < p < 1$. Put $\mathcal{L}_{(p,x)} := x\sqrt{(1-p)/p} - (l-x)\sqrt{p/(1-p)}$.

Lemma 3.

$$\sum_{x=0}^{l} \binom{l}{x} p^x (1-p)^{l-x} \mathcal{L}_{(p,x)} = 0, \sum_{x=0}^{l} \binom{l}{x} p^x (1-p)^{l-x} \mathcal{L}_{(p,x)}^2 = l.$$

Lemma 4 (Markov Bound). *Let Y be a random variable and t a positive number. If Y is non-negative,*

$$\Pr[Y \geq t] \leq \mathrm{Ex}[Y]/t,$$

where $\mathrm{Ex}[Y]$ is the expected value of Y.

Remark 2. Let er_1 be the probability to accuse a single innocent user and er_M the probability to accuse one of M innocent users. Then $\mathrm{er}_M \leq \mathrm{er}_1 \times M$, since

$$\mathrm{er}_M = 1 - (1 - \mathrm{er}_1)^M < 1 - (1 - \mathrm{er}_1 M) = \mathrm{er}_1 M.$$

A.2 Proof of Lemma 1

Proof (Proof of Lemma 1). In general,

$$\Pr[S > Z] = \Pr[\alpha S > \alpha Z] = \Pr[e^{\alpha S} > e^{\alpha Z}].$$

Since we assume $\alpha > 0$ and by Markov bound (Lemma 4), $\Pr[e^{\alpha S} > e^{\alpha Z}] < \frac{\mathrm{Ex}[e\quad]}{e}$.

From now, we show $\mathrm{Ex}[e^{\alpha S}] < e^{r\alpha^2 n}$. Put $u_i = \sqrt{\frac{1-p}{p}}\delta_{1,c\ ,} - \sqrt{\frac{p}{1-p}}\delta_{0,c\ ,}$, appeared in §2.3, then

$$\mathrm{Ex}[e^{\alpha S}] = \mathrm{Ex}[e^{\alpha \sum\ :\ \ =1\ u}\ ].$$

Since u_i are bitwise-independent, $\mathrm{Ex}[e^{\alpha \sum\ :\ \ =1\ u}\ ] = \prod_{i:y\ =1} \mathrm{Ex}[e^{\alpha u}\ ]$. Thus

$$\mathrm{Ex}[e^{\alpha u}\ ] \leq \mathrm{Ex}[1 + \alpha u_i + r_1\alpha^2 u_i^2] = 1 + \alpha\mathrm{Ex}[u_i] + r_1\alpha^2\mathrm{Ex}[u_i^2].$$

It is easy to verify that $\mathrm{Ex}[u_i] = 0, \mathrm{Ex}[u_i^2] = 1$, hence

$$\mathrm{Ex}[e^{\alpha u}\ ] \leq 1 + r_1\alpha^2.$$

In general, $1 + x \leq e^x$. Therefore $\mathrm{Ex}[e^{\alpha u}\ ] \leq 1 + r_1\alpha^2 \leq e^{r_1\alpha^2}$. Thus $\mathrm{Ex}[e^{\alpha S}] \leq \prod_{i:y\ =1} e^{r_1\alpha^2} \leq \prod_i e^{r_1\alpha^2} = e^{r_1\alpha^2 n}$. Now we conclude that

$$\Pr[S > Z] \leq \frac{e^{r_1\alpha^2 n}}{e^{\alpha Z}} = e^{r_1\alpha^2 n - \alpha Z}.$$

Remark 3. We assumed two conditions in the statement of Lemma 1:

$$r_1 = (e^{x1} - 1 - x_1)/x_1^2$$

and

$$^\forall p \in \mathcal{P}, ax_1 \geq \sqrt{(1-p)/p}/c,$$

where $\alpha = 1/ac$. The second condition is equivalent to

$$ax_1 \geq \sqrt{(1-p_0)/p_0}/c,$$

where p_0 is the minimum value of $\mathcal{P}$.

A.3 Proof of Lemma 2: The Probability NOT to Accuse a Pirate

Let c be the maximal number of pirates. Let y be a given false codeword and let l the number of users who made y. We note that it is easy to trace a pirate if $l = 1$. By Marking assumption, y is the codeword of him. Thus we can assume that $2 \leq l \leq c$ to analyze the length of our code, without the loss of generality.

Lemma 5. *Let $\mathcal{P}$ be a SRV. For $0 \leq x \leq l$, put*

$$o_{l,0,x} := \mathrm{Ex}_{p\in\mathcal{P}}[p^x(1-p)^{l-x}],$$
$$o_{l,1,x} := \mathrm{EX}_{p\in\mathcal{P}}[p^x(1-p)^{l-x}e^{-\beta\mathcal{L}_{(\ ,\ )}}].$$

And put

$$M_{l,x} := \begin{cases} o_{0,0}, & x = 0 \\ \max\{o_{0,x}, o_{1,x}\}, & 1 \leq x \leq M-1 \\ o_{1,l}, & x = M \end{cases}$$

Then we have

$$\mathrm{Ex}_{p\in\mathcal{P},U}[e^{-\beta S}\ ] \leq \left(\sum_{0\leq x\leq l}\binom{l}{x}M_x\right)^n,$$

where U is the set of all $l \times n$-matrices constructed by the codewords of the coalition of l-pirates.

Proof. Indeed, it is proved by a similar argument in [8] Equation (2). Hence we omit the proof.

Let x_2, r_2 be positive numbers satisfy $e^{x_2} = 1 + x_2 + r_2x_2^2$. Now $-\beta\mathcal{L}_{(p,x)} \leq -\beta\mathcal{L}_{(p,0)} = \beta l\sqrt{p/(1-p)}$. If $\beta l\sqrt{p/(1-p)} < x_2$ for all $p \in \mathcal{P}$, then we have the following:

$$p^x(1-p)^{l-x}e^{-\beta\mathcal{L}_{(\ ,\ )}} < p^x(1-p)^{l-x}(1 - \beta\mathcal{L}_{(p,x)} + r_2\beta^2\mathcal{L}^2_{(p,x)}).$$

For $1 \leq x \leq l-1$, we define

$$o_{l,2,x} := \mathrm{EX}[p^x(1-p)^{l-x}] - \beta\mathrm{EX}[p^x(1-p)^{l-x}\mathcal{L}_{(p,x)}] + r_2\beta^2\mathrm{EX}[p^x(1-p)^{l-x}] + \beta R_{l,x},$$

where

$$R_{l,x} := \max\{0, \mathrm{EX}[p^x(1-p)^{l-x}\mathcal{L}_{(p,x)}]\}.$$

By the definition of $o_{l,2,x}$, we have $o_{l,0,x}, o_{l,1,x} \leq o_{l,2,x}$.

Proof (Proof of Lemma 2). It is known that $\mathrm{Ex}_{p\in\mathcal{P},U}[e^{-\beta S}\ ] \leq (\sum_{0\leq x\leq l}\binom{l}{x}M_x)^n$ (Lemma 5).

$$\begin{aligned} \sum_{0\leq x\leq l}\binom{l}{x}M_x &\leq o_{0,0} + o_{1,1} + \sum_{1\leq x\leq l-1}\binom{l}{x}o_{2,x} \\ &= \sum_{0\leq x\leq l}\binom{l}{x}\mathrm{EX}[p^x(1-p)^{l-x}] + r_2\beta^2\sum_{1\leq x\leq l}\binom{l}{x}\mathrm{EX}[p^x(1-p)^{l-x}\mathcal{L}^2_{(p,x)}] \\ &\quad -\beta\sum_{1\leq x\leq l}\binom{l}{x}\mathrm{EX}[p^x(1-p)^{l-x}\mathcal{L}_{(p,x)}] + \beta\sum_{1\leq x\leq l-1}\binom{l}{x}R_{l,x}. \end{aligned} \tag{4}$$

By Lemma 3,

$$\begin{aligned}
(4) &= 1 + r_2\beta^2(l - \mathrm{EX}[(1-p)^l \mathcal{L}^2_{(p,0)}]) \\
&\quad -\beta(0 - EX[(1-p)^l \mathcal{L}_{(p,0)}]) + \beta \sum_{1 \le x \le l-1} \binom{l}{x} R_{l,x} \\
&\le 1 + \beta \left\{ r_2 l\beta - lEX[(1-p)^{l-1/2}p^{1/2}] + \sum_{1 \le x \le l-1} \binom{l}{x} R_{l,x} \right\} \\
&\le 1 - \beta \left\{ -r_2/b + lEX[(1-p)^{l-1/2}p^{1/2}] - \sum_{1 \le x \le l-1} \binom{l}{x} R_{l,x} \right\}. \qquad (5)
\end{aligned}$$

Remember $1 + x \le e^x$. Then

$$\begin{aligned}
(5) &\le \exp\left(-\beta \left\{ -r_2/b + l\mathrm{EX}[(1-p)^{l-1/2}p^{1/2}] - \sum_{1 \le x \le l-1} \binom{l}{x} R_{l,x} \right\} \right) \\
&= \exp(-\beta\{-r_2/b + \mathcal{R}_{l,\mathcal{P}}\}) = \exp(-\beta s).
\end{aligned}$$

Remark 4. We assumed the following conditions in the statement of Lemma 2:

$$e^{x_2} = 1 + x_2 + r_2 x_2$$

and

$$^{\forall}p \in \mathcal{P}, \beta l \sqrt{p/(1-p)} \le x_2.$$

The second condition is equivalent to

$$bx_2 \ge \sqrt{(1-p_0)/p_0} l/c,$$

where $\beta = 1/bc$ and the minimum $p_0 \in \mathcal{P}$. In particular, it is sufficient to

$$bx_2 \ge \sqrt{(1-p_0)/p_0},$$

since $l \le c$.

A General Formulation of Algebraic and Fast Correlation Attacks Based on Dedicated Sample Decimation

Miodrag J. Mihaljević[1], Marc P.C. Fossorier[2], and Hideki Imai[3]

[1] Mathematical Institute, Serbian Academy of Sciences and Arts,
Kneza Mihaila 35, 11001 Belgrade, Serbia and Montenegro
miodragm@turing.mi.sanu.ac.yu

[2] Department of Electrical Engineering, University of Hawaii,
2540 Dole St., Holmes Hall 483, Honolulu, HI 96822, USA
marc@spectra.eng.hawaii.edu

[3] University of Tokyo, Institute of Industrial Science,
4-6-1, Komaba, Meguro-ku, Tokyo 153-8505, Japan
imai@iis.u-tokyo.ac.jp

Abstract. This paper proposes a novel approach for cryptanalysis of certain cryptographic pseudorandom sequence (keystream) generators consisting of the composition of a linear finite state machine (LFSM) and nonlinear mapping. The proposed approach includes a dedicated decimation of the sample for cryptanalysis based on the following: Suppose certain B bits of the LFSM initial state as known and identify time instances where certain arguments of the nonlinear function depend only on these B bits and are equal to zero. As opposed to previously reported methods, the proposed one also identifies and uses certain characteristics of the LFSM state-transition matrix in order to reduce the nonlinearity of the system of overdefined equations employed in an algebraic attack scenario, or to reduce the noise introduced by the linearization of the nonlinear function which corrupts the linear equations employed in a correlation attack scenario.

Keywords: overdefined systems of nonlinear equations, decimation, decoding, stream ciphers, keystream generators, state transition matrix, LFSRs, cryptanalysis, algebraic attacks, fast correlation attack.

1 Introduction

This paper points out novel algebraic and correlation attack techniques for cryptanalysis of certain keystream generators for stream ciphers known as the nonlinear combination generators (see [11], for example).

Algebraic and correlation attacks are well recognized as the general purpose tools for security evaluation and cryptanalysis of these generators. A general paradigm of the algebraic and correlation attacks is based on establishing and processing a system of overdefined equations which are:

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 203–214, 2006.

- nonlinear and (mainly) error free in the case of algebraic attacks;
- linear and (very) noisy in the case of correlation attacks (assuming that a noisy equation denotes an equation which is satisfied with a certain, known, probability).

Recently, algebraic attacks have appeared as a powerful tool for cryptanalysis and security evaluation of certain encryption schemes and particularly stream ciphers including the nonlinear filter based keystream generators. Some early algebraic attacks on stream and related ciphers have been reported in [4] as well as in [17] and [18]. Very recently, a number of algebraic attacks have been reported in [5], [6], [13], [1], [8] and [19]. An algebraic attack can be roughly summarized as follows: (i) Describe the secret key by a largely overdefined system of (low-degree) nonlinear algebraic equations; (ii) If the number of equations exceeds the number of terms, linearize the system; i.e. treat each term as an independent variable and solve this (huge) system of linear equations, or (iii) Try to solve the system by other appropriate techniques (Grobner basis, ...).

On the other hand, a correlation attack can be summarized as follows: (i) Describe the secret key as a largely overdefined system of noisy linear algebraic equations; (ii) Employ an appropriate decoding oriented procedure for finding a solution. All contemporary correlation attacks originate from [22] where this cryptanalytic approach was introduced, and [12] where the first fast correlation attack algorithm was proposed. The fast correlation attack is usually modeled as the problem of recovering a LFSR initial state when its output sequence is observable via a binary symmetric channel (BSC) with crossover probability equal to p. The modeling of a BSC is a consequence of the linearization of the keystream generator. Accordingly, the fast correlation attack can be addressed as the decoding of an appropriate code related to the LFSR output sequence. As underlying codes, certain block and convolutional codes have been considered, and the employed decoding techniques include two main approaches: one pass decoding and iterative decoding. The reported iterative block decoding approaches include [14], [15], and the non-iterative approaches include those reported in [16] and [3], for example. The most efficient techniques include a search over all hypotheses on a subset of the information bits. The convolutional code based approaches for fast correlation attack have been considered in a number of papers including the ones recently reported in [9] and [21].

Motivation for the Work. The general powerful algebraic attacks that have been recently reported are based on the construction of an overdefined system of nonlinear equations employing only certain characteristics of the nonlinear function. Accordingly, the performance of these attacks strongly depends on the nonlinear part, and if this part does not have certain characteristics appropriate for cryptanalysis, the attacks could become very complex or even not feasible. A goal of this paper is to address the following issue: Find a way to include into the algebraic attack certain characteristics of the linear part in order to obtain more powerful attacks against certain nonlinear functions (which could be heavily resistant against the reported algebraic attacks). An additional motivation for this work was a complementary extension of the algebraic attack approach reported in [19].

The paradigm of contemporary fast correlation attacks could be considered as consisting of the following main steps: (i) assuming that certain secret key bits are known, specification of an overdefined system of noisy linear equations; (ii) solving the specified system as a decoding problem via hypothesis testing and evaluation of parity checks. The noise involved in the system of equations is a consequence of the linearization of a nonlinear system of equations which describes the considered stream cipher. As a result, this noise is not an usual random one and could be an objective for adjustment attempts. Accordingly, motivations for this work include consideration of the possibilities for specifying the systems of equations with a noise level lower than the one obtained by a simple straightforward linearization of the initial system of nonlinear equations related to the nonlinear filter.

Finally, a motivation for this work was to generalize the attacking approach on a particular keystream generator reported in [20].

Organization of the Paper. The model of the keystream generators under consideration is given in Section 2. The framework for the dedicated decimation based cryptanalysis employing algebraic and fast correlation attack approaches is proposed in Section 3. Following this framework, novel algebraic and fast correlation attacks are proposed and analyzed in Sections 4 and 5, respectively.

2 Model of the Keystream Generators Under Consideration

2.1 Preliminaries

An m-variable Boolean function $f(x_1, x_2, \ldots, x_m)$ can be considered as a multivariate polynomial over GF(2). This polynomial can be expressed as a sum of products of all distinct r-th order products $(0 \leq r \leq m)$ of the variables as follows:

$$f(x_1, x_2, \ldots, x_m) = \bigoplus_{u \in GF(2\ \ )} \lambda_u \prod_{i=1}^{m} x_i^u \ , \ \lambda_u \in GF(2) \ , \ u = (u_1, u_2, \ldots, u_m)$$

or

$$f(x_1, x_2, \ldots, x_m) = a_0 \oplus_{1 \leq i \leq m} a_i x_i \oplus_{1 \leq i < j \leq m} a_{ij} x_i x_j \oplus \cdots \oplus a_{12 \ldots m} x_1 x_2 \ldots x_m \ , \tag{1}$$

where the coefficients $a_0, a_i, a_{ij}, \ldots, a_{12 \ldots m} \in \mathrm{GF}(2)$. This representation of $f(\cdot)$ is called the algebraic normal form (ANF) of f. The algebraic degree of f, denoted by $deg(f)$ or simply d, is the maximal value of the Hamming weight of u such that $\lambda_u \neq 0$, or the number of variables in the highest order term with nonzero coefficient.

Note that the ANF of $f(\cdot)$ directly specifies one multivariate equation between the function arguments and its output which has the nonlinearity order equal to the algebraic degree of $f(\cdot)$, but in many cases additional multivariate equations with a lower nonlinearity order can be specified as well. When a linear combining of the equations is allowed, the linear combination can be with a lower degree than the component equations, assuming that enough equations are available for the combining (see [6], for example).

A binary linear finite state machine (LFSM) can be described as $\mathbf{X}_t = \mathbf{A}\mathbf{X}_{t-1}$, where $\mathbf{A}$ is the **state transition matrix** (over GF(2)) of the considered LFSM. Let $\mathbf{X}_0$ be the column $(L \times 1)$ matrix $[X_{L-1}, \ldots, X_0]^T$ representing the initial contents or initial state of the LFSM, and let $\mathbf{X}_t = [X_{L-1}^{(t)}, \ldots, X_0^{(t)}]^T$, be the L-dimensional column vector over GF(2) representing the LFSM state after t clocks, where $\mathbf{X}^T$ denotes the transpose of the L-dimensional vector $\mathbf{X}$. We define

$$\mathbf{X}_t = \mathbf{A}\mathbf{X}_{t-1} = \mathbf{A}^t\mathbf{X}_0, \qquad \mathbf{A}^t = \begin{bmatrix} \mathbf{A}_1^{(t)} \\ \cdot \\ \mathbf{A}_L^{(t)} \end{bmatrix}, \quad t = 1, 2, \ldots, \tag{2}$$

where $\mathbf{A}^t$ is the t-th power over GF(2) of the $L \times L$ state transition binary matrix $\mathbf{A}$, and each $\mathbf{A}_i^{(t)}$, $i = 1, 2, \ldots, L$, represents a $1 \times L$ matrix (a row-vector).

2.2 Basic Model

For simplicity of presentation, the novel algebraic and correlation based attack techniques proposed in this paper for cryptanalysis of certain keystream generators are introduced via the nonlinear filter keystream generator model (see [11], for example). The developed approach is also applicable for attacking certain keystream generators belonging to the class of nonlinear combination generators which consist of a number of LFSMs whose outputs are combined by a nonlinear Boolean function (see [11], for example).

Accordingly, the basic model of the keystream generators under cryptanalytic consideration is depicted in Fig. 1 where LFSM denotes a known LFSM with only the initial state $\mathbf{X}_0$ determined by the secret key, and $f(\cdot)$ denotes a known nonlinear memoryless function of m arguments $\{x_j\}_{j=1}^m$ specified by the state $\mathbf{X}_t$ of LFSM as follows:

$$x_j = X_{i(j)}^{(t)}, \quad i(j) \geq j, \quad i(j) < i(j+1), \;\; j = 1, 2, \ldots, m. \tag{3}$$

3 Underlying Ideas for the Decimated Sample Based Cryptanalysis

The developed approach for cryptanalysis is based on the following framework.

- *Pre-Processing*: Assuming that a certain subset of the secret bits is known, decimate the sample so that at the selected points the nonlinear function degenerates into a more suitable one for the cryptanalysis.

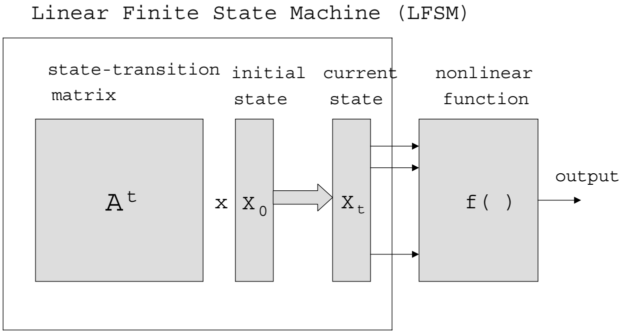

Fig. 1. Basic model of the keystream generators under consideration: The nonlinear filter

- *Processing*: Perform the main steps of cryptanalysis taking into account only the sample elements selected in the pre-processing phase.

Accordingly, the nonlinear function $f(\cdot)$ is considered as:

$$f(x_1, x_2, \ldots, x_m) = x_j \oplus g(x_1, x_2, \ldots, x_m) \ , \ j \in \{1, 2, \ldots, m\} \ . \tag{4}$$

Obviously, the function $g(x_1, x_2, \ldots, x_m)$ does not contain the linear term x_j and it has the same algebraic degree as $f(\cdot)$.

One of the main objectives of the pre-processing phase is to identify an appropriate sample decimation so that one of the following two goals is achieved:

(a) at the decimated points, $g(\cdot)$ is equal to zero or it has the algebraic degree $d^* << d$;
(b) at the decimated points, $g(\cdot)$ reduces to a nonlinear function $g^*(\cdot)$ which can be approximated by noise that corresponds to p^*, which is much smaller than the noise defining p determined by a direct approximation of $f(\cdot)$ with x_j.

The required decimation is based on the consideration of the state-transition matrix powers in order to find powers t for which at certain indices i, in each vector $\mathbf{A}_i^{(t)}$ the all zero pattern appears at pre-specified positions. Accordingly, we introduce the following definition.

Definition 1. The sets $\mathcal{T}$ and $\mathcal{I}$ are sets of the values t and i, respectively, determined by the vectors $\mathbf{A}_i^{(t)}$ with the last $L - B$ elements equal to zero. The cardinalities of $\mathcal{T}$ and $\mathcal{I}$ are $|\mathcal{T}|$ and $|\mathcal{I}|$, respectively.

Note that the set $\mathcal{I}$ is a subset of the indices $\{i(j)\}_{j=1}^{m}$ from (3).

Regarding the above proposed basic framework we have the following.

- The implementation of the framework includes a preprocessing phase which is independent of a particular sample (i.e. it should be done only once), and a processing phase which recovers the secret key based on the given sample.

- Assuming a nonlinear function suitable for the proposed attack, the gain in the processing phase is a consequence of the following:
  - a highly reduced nonlinearity of the related system of equations in the case of algebraic attacks;
  - a highly reduced correlation noise in the case of fast correlation attacks.

4 A Novel Algebraic Attack

This section proposes and analyzes a particular algorithm which targets the case $g(\cdot) = 0$, i.e. $d^* = 0$. In a similar manner an algorithm which targets the case when the algebraic degree $d^* << d$ can be developed.

The proposed algorithm follows the framework pointed out in the previous section, as well as an approach based on classes of equivalent hypotheses defined by the following.

Definition 2. For given $\mathbf{A}_i^{(t)}$ with all zeros in the last $L - B$ positions and w ones at the first B positions, a class of equivalent hypotheses on B bits of the LFSM initial state $\mathbf{X}_0$ is a set of B-dimensional binary vectors where a certain pattern corresponds to the w positions of ones in $\mathbf{A}_i^{(t)}$ and with all possible 2^{B-w} binary patterns at the remaining $B - w$ positions.

4.1 Algorithm I

- **Pre-Processing Phase**
  - *Input*: (i) the generator elements L, $\mathbf{A}$, $f(\cdot)$, $g(\cdot)$; and (ii) the parameter B which depends on the values of N and L, and fulfills $L - (\lfloor L/B \rfloor + 1)B < (B/2) + \log_2 B + |\mathcal{I}|$.
  - *Pre-Processing Steps*
    - ∗ Identify a minimal sub-set of the arguments x_i, $i \in \mathcal{I}$, such that their equality to zero implies $g(\cdot) = 0$ as well.
    - ∗ For each of $\lfloor L/B \rfloor + 1$ different suitably selected non-overlapping positions P_B of the B bits do the following:
      1. Searching over the powers of the state transition matrix $\mathbf{A}$, for the given set $\mathcal{I}$ determine the set $\mathcal{T}$ specified by Definition 1.
      2. For each $t \in \mathcal{T}$ specify the set of eligible hypotheses as a subset of all possible 2^B hypotheses for which the condition $g(\cdot) = 0$ is fulfilled.
      3. For each $t \in \mathcal{T}$ categorize all the eligible hypotheses into the classes $\mathcal{H}_q^{(t)}$, $q = 1, 2, \ldots, Q$, of equivalent hypotheses specified by Definition 2.
      4. Implement a suitable list of all 2^B hypotheses which support removal from the list of all the hypotheses from a class of equivalent hypotheses via one-step parallel processing.
  - *Output*: Outputs of the steps 3 and 4.

- **Processing Phase**
  - *Input*: (i) Inputs and outputs of the Pre-Processing Phase; and (ii) the generator output sequence $\{z_i\}_{i=1}^{N}$.
  - *Processing Steps*
    1. For each $\lfloor L/B \rfloor + 1$ sets P_B do the following:
       (a) For each $t \in \mathcal{T}$ and for each class of equivalent hypotheses $\mathcal{H}_q^{(t)}$ perform the following:
          i. Select an arbitrary hypothesis from $\mathcal{H}_q^{(t)}$ on the B bits of the LFSM initial state;
          ii. For the considered hypothesis evaluate $\hat{z}_t = \mathbf{A}_j^{(t)} \mathbf{X}_0$;
          iii. If $\hat{z}_t \neq z_t$, $t \in \mathcal{T}$, remove from the list all the hypotheses of the considered class of equivalent hypotheses.
    2. For each of the remaining hypotheses on B bits and all possible hypotheses on the remaining $L - (\lfloor L/B \rfloor + 1)B$ bits of the LFSM initial state $\mathbf{X}_0$, do the following:
       (a) Evaluate $\hat{z}_t = f(\{\mathbf{A}_j^{(t)} \mathbf{X}_0\}_{j=1}^{m})$, $t = 1, 2, \ldots, 2L$;
       (b) If $\hat{z}_t = z_t$, $t = 1, 2, \ldots, 2L$, accept the considered candidate as the correct one and go to the output.
  - *Output*: Recovered LFSM initial state.

4.2 Analysis of the Complexity

Required Sample
The required sample depends on the employed LFSM, and in general, only the upper bound can be claimed. Due to the decimation approach, it is expected to be relatively long.

Proposition 1. According to the preprocessing steps 1 and 2, the upper bound on required sample length N is $O(2^{L-B+\mathcal{I}})$.

Space Complexity
The space complexity is determined by the space requirements for storing the list of all 2^B hypotheses and for storing all the lists of equivalent hypotheses for all decimated sample points. As a consequence of the binomial distribution of the weights (number of ones in a binary vector) of each $\mathbf{A}_i^{(t)}$, for each sampled position, there are on average $2^{B/2}$ classes of the equivalent hypotheses. On the other hand due to the condition $g(\cdot) = 0$ the expected length of each list of equivalent hypotheses is equal to $2^{(B/2)-|\mathcal{I}|}$.

Proposition 2. According to the pre-processing steps 2-4 and the structure of the entire algorithm the space complexity of the algorithm is $O(B2^B)$.

Time Complexity

Proposition 3. According to the pre-processing steps 1 and 2 the time complexity of pre-processing is $O(2^{L-B+|\mathcal{I}|}) + O(2^B)$.

Table 1. Comparison of the proposed algebraic attack and the algebraic attack based on the results reported in [6], [1] and [8] assuming in the second case that the algebraic degree of the employed function $f(\cdot)$ is d

	pre-processing time complexity	processing time complexity	upper bound on required sample	required memory
algebraic attack based on [6], [1] and [8]	$O\left(\binom{L}{d}^{2.7}\right)$	$O(\binom{L}{d}L^2)$	$O(\binom{L}{d})$	$O(\binom{L}{d}L^2)$
proposed Algorithm I	$O\left(2^{L-B+\lvert\mathcal{I}\rvert}\right)+O(2^B)$	$O\left(2^{(B/2)+\log_2 B+\lvert\mathcal{I}\rvert}\right)$	$O(2^{L-B+\mathcal{I}})$	$O(B2^B)$

For consideration of the time complexity of processing we assume the following.

Assumption 1. The complexity of removing a class of equivalent hypotheses from the list of all possible hypotheses is $O(1)$.

The implementation of Assumption 1 includes a dedicated memory access management. Particularly note that in a dedicated implementation, removal of a class of equivalent hypotheses from the list of all possible hypotheses does not require random memory access but the static one. A possible implementation framework could be as follows. Assign to each class of equivalent hypotheses $\mathcal{H}_q^{(t)}$ a variable $\alpha_q^{(t)}$ which is initially set to 1, and changes to 0 if the class $\mathcal{H}_q^{(t)}$ is rejected. To each of the 2^B hypotheses assign: (i) a number of pointers corresponding to all $\alpha_q^{(t)}$ regarding $\mathcal{H}_q^{(t)}$ to which the considered hypothesis belongs; and (ii) an AND logic gate with inputs $\alpha_q^{(t)}$ provided by the pointers. Note that after consideration of all the sets $\mathcal{H}_q^{(t)}$ it is expected that only one hypothesis will remain which has all assigned $\alpha_q^{(t)}$ equal to one, and this hypothesis can be identified with time complexity $\log(2^{B+\mathcal{I}})$. The described framework has space complexity $O(B2^B)$.

Proposition 4. According to the processing step 1(a), the time complexity of processing is $O(2^{(B/2)+\log_2 B+|\mathcal{I}|})$.

According to Propositions 1–4, Table 1 summarizes the performance of the proposed Algorithm I and compares it with related previously reported algorithms.

5 A Novel Fast Correlation Attack

5.1 Framework

Instead of employing just a straightforward linearization of the nonlinear function $f(\cdot)$, i.e. approximation of the nonlinear function $g(\cdot)$ by a constant, the approach proposed in this section takes into account certain (suitable) characteristics of the linear part which provide a possibility for establishing relevant linear equations corrupted by a lower noise. Consequently, the recovery of the secret key via the correlation decoding approach becomes easier. The LFSM initial state is considered as consisting of two suitably determined parts: one which should be determined by an exhaustive search, and the other which can be recovered via an appropriately designed correlation attack, assuming that the first part is known due to the employed hypotheses.

When certain necessary conditions are fulfilled, in general, the correlation attack based algorithms for cryptanalysis under development consist of the following main steps:

(i) Suppose that certain B secret key bits (variables) will be recovered by exhaustive search;
(ii) Determine a set $\mathcal{T}$ of time instances t where a subset S_g of arguments of $g(\cdot)$ depend only on the assumed B bits;
(iii) For each assumed pattern of B bits, identify a subset $\mathcal{T}_B$ of $\mathcal{T}$ where all the arguments in S_g are equal to zero; As a result the approximation of $g(\cdot)$ by zero at these time instances implies correctness of each parity check equations with probability $1-p^*$, where $p^* \leq p$, and $1-p$ is the correctness of the same parity-check equations when a straightforward linearization of $f(\cdot)$ is applied without taking into account a suitable decimation;
(iv) Recover a candidate for the secret key via solving the system of overdefined noisy linear equations related only to the time instances $t \in \mathcal{T}_{\mathcal{B}}$ employing an appropriate decoding technique, and check correctness of the candidate.

5.2 Algorithm II

- **Pre-Processing Phase**
  - *Input*: (i) the generator elements L, $\mathbf{A}$, $f(\cdot)$, $g(\cdot)$; and (ii) the parameter B (which depends on the values of N and L).
  - *Pre-Processing Steps*
    1. Identify a minimal subset of arguments x_i, $i \in \mathcal{I}$, such that their equality to zero implies that $g(\cdot)$ can be approximated by a constant introducing a lower approximation (correlation) noise than direct linearization of $f(\cdot)$.
    2. Searching over the powers of the state transition matrix $\mathbf{A}$, for the given set $\mathcal{I}$, determine the set $\mathcal{T}$ specified by Definition 1.
    3. Employing particular characteristics of LFSM states at the positions $t \in \mathcal{T}$ construct the basic parity check equations and via linear combining up to w of these parity check equations (in the manner employed in [16] and [3], for example) for each of the not assumed $L-B$ elements of $\mathbf{X}_0$ specify a set Ω_i of the related parity checks of the weight $B+w+1$, $i = B+1, B+2, \ldots, L$.
  - *Output*: Outputs of the steps 2 and 3.

- **Processing Phase**
  - *Input*: (i) Inputs and outputs of the Pre-Processing Phase; and (ii) the generator output sequence $\{z_i\}_{i=1}^{N}$.
  - *Processing Steps*:
    1. Select a previously not considered B-bit pattern.
    2. For each $i = B+1, B+2, \ldots, L$, select a subset $\Omega_i(B)$ of Ω_i with the parity-check equations related only to the positions $t \in \mathcal{T}_B$ where the arguments from the subset $\mathcal{I}$ are equal to zero.
    3. Perform the appropriate decoding procedure (like those reported in [16] and [3], for example), and recover the values of the $L-B$ initial state bits under the assumption that the B-bits in the processing Step 1 are correct.
    4. Compare the output sequence generated based on the initial state specified by the processing Step 1 and Step 3, and the one given as the algorithm input.
    5. If the sequences under comparison in Step 4 are at the Hamming distance lower than the certain threshold, accept the initial state determined by the current hypothesis on B bits and the recovered $L-B$ bits as the correct one, recover the secret key and go to the processing phase Output. Otherwise, go to the processing Step 1.
  - *Output*: Recovered secret key.

Remark 1. The given Algorithm II is a basic form of the proposed fast correlation attack approach based on dedicated sample decimation. The concept of eligible and equivalent hypotheses used in Algorithm I, as well as certain elements of the techniques reported in [3] and [10], can be involved into a more sophisticated version of the algorithm for cryptanalysis, but these issues are out the scope of this paper.

5.3 Analysis of the Complexity

Assumption 2. A hypothesis on B out of the L secret bits of the LFSM initial state, under the condition that in the decimated sample one input of the nonlinear function is always equal to zero, decreases the correlation noise to p^*, $p^* < p$. This implies that M^* parity-checks of weight w per bit provide the correct decoding of that bit with probability close to 1.

Remark 2. According to the results presented in [7] for given p^* and w, the required number M^* of parity checks can be estimated as $O((1-2p^*)^{-2w})$. Regarding the parameter w see [16], [3] for the trade-off between the required sample size and pre-processing/processing complexity.

Remark 3. For each $i = B+1, B+2, \ldots, L$, the expected cardinality of $\Omega_i(B)$ is $2^{-w}|\Omega_i|$, and so the expected cardinality of Ω_i should be $2^w M^*$.

Table 2. Parameters, complexity and required sample of the proposed fast correlation attack versus corresponding the most powerful reported one - A simplified comparison

	required number of parity-check equations per a bit	pre-processing complexity	processing complexity	required sample
reported attack [16]	M	$O(2^{L-B+\log_2(L-B)+\log_2 M})$	$O((L-B)2^B M)$	N
proposed Algorithm II	$M^* < M$	$O(2^{L-B+\log_2(L-B)+w+\log_2 M^*})$	$O((L-B)2^B M^*)$	$N^* > N$

Accordingly, the structure of the proposed fast correlation attack implies the following statements.

Proposition 5. When Assumption 2 holds, the expected sample N^* required for the proposed fast correlation attack satisfies $2^{B-L}(L-B)\binom{2\ -\ N^*}{w} > 2^w M^*$ implying that required length of N^* is $O(2^{L-B+w^{-1}(L-B-\log_2(L-B)+w+\log_2 M^*)})$.

Proposition 6. When Assumption 2 holds, the expected time complexity of the proposed fast correlation attack pre-processing is $O(2^{L-B+\log_2(L-B)+w+\log_2 M^*})$.

Proposition 7. When Assumption 2 holds, the expected time complexity of the proposed fast correlation attack processing is $O((L-B)2^B M^*)$.

Proposition 8. When Assumption 2 holds, the expected space complexity of the proposed fast correlation attack processing is $O((L-B)2^w M^*)$.

According to Propositions 5–7, Table 2 summarizes the performance of the proposed Algorithm II and compares it with the related previously reported algorithm.

References

1. F. Armknecht, "Improving fast algebraic attacks", FSE 2004, *Lecture Notes in Computer Science*, vol. 3017, pp. 65-82, 2004.
2. A. Braeken, V. Nikov, S. Nikova and B. Preneel, "On Boolean functions with generalized cryptographic properties", INDOCRYPT 2004, *Lecture Notes in Computer Science*, vol. 3348, pp. 120-135, 2004.
3. P. Chose, A. Joux and M. Mitton, "Fast Correlation Attacks: An Algorithmic Point of View", EUROCRYPT 2002, *Lecture Notes in Computer Science*, vol. 2332, pp. 209-221, 2002.
4. N.T. Courtois, "Higher order correlation attacks, XL algorithm and cryptanalysis of Toyocrypt", ICISC2002, *Lecture Notes in Computer Science*, vol. 2587, pp. 182-199, 2003.

5. N.T. Courtois and W. Meier, "Algebraic attacks on stream ciphers with linear feedback", EUROCRYPT2003, *Lecture Notes in Computer Science*, vol. 2656, pp. 345-359, 2003.
6. N.T. Courtois, "Fast algebraic attacks on stream ciphers with linear feedback", CRYPTO2003, *Lecture Notes in Computer Science* vol. 2729, pp. 176-194, 2003.
7. M. P. C. Fossorier, M. J. Mihaljević and H. Imai, "A Unified Analysis on Block Decoding Approaches for the Fast Correlation attack," *2005 IEEE Int. Symp. Inform. Theory - ISIT 2005*, Adelaide, Australia, September 2005, Proceedings, 4 pages (accepted for publication).
8. P. Hawkes and G. Rose, "Rewriting variables: the complexity of Fast algebraic attacks on stream ciphers", CRYPTO 2004, *Lecture Notes in Computer Science*, vol. 3152, pp. 390-406, 2004.
9. T. Johansson and F. Jonsson, " Theoretical Analysis of a Correlation Attack Based on Convolutional Codes," *IEEE Trans. Information Theory*, vol. 48, pp. 2173-2181, August 2002.
10. P. Lu, "A new correlation attack on LFSR sequences with high error tolerance", in *Coding, Cryptography and Combinatorics*, Eds. K. Feng, H. Niederreiter and C. Xing: Birkhauser Verlag AG, May 2004.
11. A. Menezes, P.C. van Oorschot and S.A. Vanstone, *Handbook of Applied Cryptography*. Boca Roton: CRC Press, 1997.
12. W. Meier and O. Staffelbach, "Fast correlation attacks on certain stream ciphers," *Journal of Cryptology*, vol. 1, pp. 159-176, 1989.
13. W. Meier, E. Pasalic and C. Carlet, "Algebraic attacks and decomposition of Boolean functions", EUROCRYPT 2004, *Lecture Notes in Computer Science*, Vol. 3027, pp. 474-491, 2004.
14. M. J. Mihaljević, M. P. C. Fossorier and H. Imai, "On decoding techniques for cryptanalysis of certain encryption algorithms," *IEICE Trans. Fundamentals*, vol. E84-A, pp. 919-930, Apr. 2001.
15. M. J. Mihaljević and J. Dj. Golić, "A method for convergence analysis of iterative probabilistic decoding," *IEEE Trans. Inform. Theory*, vol. 46, pp. 2206-2211, Sept. 2000.
16. M. J. Mihaljević, M. P. C. Fossorier and H. Imai, "Fast Correlation Attack Algorithm with List Decoding and an Application", FSE 2001, *Lecture Notes in Computer Science*, vol. 2355, pp. 196-210, 2002.
17. M. J. Mihaljević and H. Imai, "Cryptanalysis of TOYOCRYPT-HS1 stream cipher", *IEICE Transactions on Fundamentals*, vol. E85-A, pp. 66-73, Jan. 2002.
18. M. J. Mihaljević and R. Kohno, "Cryptanalysis of fast encryption algorithm for multimedia FEA-M", *IEEE Commun. Lett.*, vol. 6, pp. 382-384, Sept. 2002.
19. M. J. Mihaljević and H. Imai, "The decimated sample based improved algebraic attacks on the nonlinear filters", SCN 2004, *Lecture Notes in Computer Science*, vol. 3352, pp. 310-323, Jan. 2005.
20. M. J. Mihaljević, M. P. C. Fossorier and H. Imai, "Cryptanalysis of keystream generator by decimated sample based algebraic and fast correlation attacks", INDOCRYPT2005, *Lecture Notes in Computer Science*, vol. 3707, pp. 155-168, Dec. 2005.
21. H. Molland, J.E. Mathiassen and T. Helleseth, "Improved Fast Correlation Attack using Low Rate Codes", Cryptography and Coding 2003, *Lecture Notes in Computer Science*, vol. 2898, pp. 67-81, 2003.
22. T. Siegenthaler, "Decrypting a class of stream ciphers using ciphertext only," *IEEE Trans. Comput.*, vol. C-34, pp. 81-85, 1985.

Traitor Tracing Against Powerful Attacks Using Combinatorial Designs

Simon McNicol, Serdar Boztaş, and Asha Rao

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

Abstract. This paper addresses the problem of threshold traitor tracing for digital content where, by embedding appropriate digital patterns into the distributed content, it is possible to trace and identify the source of unauthorised redistribution.

We use a set of marking assumptions where the adversaries have varying powers to change or erase coordinates of the fingerprint where their individual fingerprints differ–and consider the implications. We propose new codes derived from combinatorial designs–and develop a method for concatenating these codes to filter out the false positives and defend against some of the attacks considered.

Keywords: Traitor tracing, fingerprinting, digital rights management, coding theory, block designs, Reed-Solomon codes.

1 Introduction

Fingerprinting (see [4, 2, 1]) is a technique that aims to prevent the unauthorized redistribution of digital content. Very close copies of a digital document are made available to a large number of authorized users. The locations where the copies differ are where the fingerprint has been embedded into the digital object. Malicious users attempt to discover the fingerprint and alter it to construct rogue copies which will still "function". The document is assumed to be a string over a finite alphabet, with the fingerprint being a randomly spread-out substring of the former, but much shorter than the document itself. *Traitor Tracing* (see [5, 3]) schemes enable the tracing of the user(s) whose fingerprints were used to construct the rogue copies. Here we are interested in tracing more powerful attacks as in the example below.

Example 1. *For two coalitions drawn from codewords* 01234*,* 00224 *and* 10323 *we exhibit some descendants below:*

codeword		*codeword*	
*traitor*1	0 1 2 3 4	*traitor*1	0 1 2 3 4
*traitor*2	0 0 2 2 4	*traitor*3	1 0 3 2 3
narrow	0 0 2 3 4	*narrow*	1 1 3 3 3
wide	0 4 2 # 4	*wide*	# 0 3 # 4
erasure	0 # 2 # 4	*erasure*	# # # # #
hybrid	0 # 2 3 4	*hybrid*	1 # 2 3 #
type	*descendant*	*type*	*descendant*

(1)

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 215–224, 2006.

In a *narrow attack*, the attackers can choose any symbol which already appears in their copies at that location–see Tardos [11] for an efficient solution, and [9, 10] for earlier work. In an *erasure attack*, the attackers erase symbols at locations where their copies mismatch. In a *hybrid attack*, the attackers can switch between the narrow and erasure attacks, position by position. In a *wide attack* they can choose any symbol from the alphabet or erase the locations where their copies differ. Clearly an attacker who can carry out the weaker attacks can also carry out a combination of them, so this model is reasonable.

2 Attacks on Fingerprinting Schemes

The symbol $\mathbf{v}$ denotes a vector; $v^{(i)}$ is its i-th coordinate. We assume the finite alphabet Q to be F_q, wherever convenient.

Definition 1. *A code $\mathcal{M}$ is a subset of $\mathbb{F}_q^N$. A linear code is a vector subspace of $\mathbb{F}_q^N$.*

2.1 Attack Taxonomy

We identify each user with the unique codeword that she has been assigned. A coalition is a set of attackers. We then ask: (i) *What can a coalition do?* (ii) *How large can a coalition be?*

Definition 2. *(a) Let $\mathcal{C}(\mathcal{M}, \omega)$ denote the collection of all possible coalitions of $\mathcal{M}$ with size at most ω: $\mathcal{C}(\mathcal{M}, \omega) = \{A \subseteq \mathcal{M} : |A| \leq \omega\}$.*

(b) Let $A \subseteq F_q^N$. The spectrum of A at the z-th coordinate is $\mathrm{spec}(A, z) = \left\{v^{(z)} : \mathbf{v} \in A\right\}$.

(c) Let $Q^{\#} = Q \cup \{\#\}$ be the extension of the finite alphabet Q. Let $\mathcal{T} \in \mathcal{C}(\mathcal{M}, \omega)$. The word $\mathbf{d} = (d^{(1)}, \ldots, d^{(N)})$ is a wide descendant of the coalition $\mathcal{T}$, denoted $d \in$ $\mathrm{Wdesc}_\omega(\mathcal{T})$

$$d^{(i)} \in \begin{cases} \mathrm{spec}(\mathcal{T}, i), & \text{if } |\,\mathrm{spec}(\mathcal{T}, i)| = 1 \\ Q^{\#} & \text{else.} \end{cases}$$

(d) For any code $\mathcal{M}$, the set of all wide descendants from coalitions of size $\leq \omega$ is denoted:

$$\mathrm{WDesc}_\omega(\mathcal{M}) = \bigcup_{\mathcal{T} \in C(\mathcal{M}, \omega)} \mathrm{Wdesc}_\omega(\mathcal{T}) \tag{2}$$

The special cases of the wide attack have been informally defined in Example 1 due to space constraints. The narrow, erasure and hybrid attacks need to be defended against if we are to defend against the wide attack. We have recently considered this hierarchy of attacks [7, 8] and reduced the wide attack to a hybrid attack–by the method of alphabet boosting–at the cost of expanding the codelength. Here, we focus on the details of code design for tracing the hybrid attack, after a brief overview of the attack hierarchy, and refer the reader to [8] for the details of tracing the erasure attack and the wide attack reduction.

3 Defending Against the Various Attacks

3.1 Tracing the Hybrid Attack

We first define some concepts from the erasure attack.

Definition 3 (Coalition Erasure Profile)**.** *Consider a coalition $\mathcal{T}$; $E_{\mathcal{T}}$ is defined by,* $E_{\mathcal{T}} = \{(i, \mathrm{spec}(\mathcal{T}, i)) : |\mathrm{spec}(\mathcal{T}, i)| = 1\}$.

The erasure profile $E_{\mathcal{T}}$ also obeys $E_{\mathcal{T}} = \bigcap_{f \in \mathcal{T}} E_f$. Note that, $E_f = E_{\{f\}} = \left\{(i, f^{(i)}) \ : \ 1 \le i \le N\right\}$, i.e., E_f is the erasure profile of a coalition containing only the codeword f. Given $\mathbf{d} \in \mathrm{Edesc}(\mathcal{T})$, the erasure profile is simply the sets of pairs $(i, d^{(i)})$ of the descendant that haven't been erased:

$$E_{\mathcal{T}} = \left\{(i, d^{(i)}) : d^{(i)} \neq \#, 1 \le i \le N\right\}. \tag{3}$$

To construct a code that is resistant to an erasure attack we require: (i) $E_{\mathcal{T}}$ *must be nonempty for all coalitions* $\mathcal{T}$; (ii) *To ensure that every coalition can be uniquely defined by* $E_{\mathcal{T}}$*, we impose the simple condition:*

$$\mathcal{T}_1 \neq \mathcal{T}_2 \Leftrightarrow E_{\mathcal{T}_1} \neq E_{\mathcal{T}_2}, \ \forall \mathcal{T}_1, \mathcal{T}_2 \in \mathcal{C}(\mathcal{M}, \omega). \tag{4}$$

Definition 4 (Partial Trace)**.** *Given the Erasure profile of a coalition* $E_{\mathcal{T}}$*, the partial trace for the coalition* $\mathcal{T}$ *on the pairs* $(i, x) \in E_{\mathcal{T}}$ *is*
$B_{i,x} = \left\{\mathbf{f} \in \mathcal{M} : f^{(i)} = x\right\}$.

For a hybrid descendant $\mathbf{d}$, we define the corresponding profile as the "*tracing profile*".

Definition 5 (Tracing Profile)**.** *For any descendant* $\mathbf{d} \in \mathrm{HDesc}_\omega(\mathcal{M})$*, the tracing profile of* $\mathbf{d}$ *is given by,*

$$I_{\mathbf{d}} = \left\{(i, d^{(i)}) : d^{(i)} \neq \#, 1 \le i \le N\right\}. \tag{5}$$

Note that $E_{\mathcal{T}}$ represents the set of (coordinate, value) that a coalition **can't** erase, whereas $I_{\mathbf{d}}$ is the set of (coordinate, value) that a coalition **has not** deleted.

Definition 6 (Trace of hybrid attack)**.** *The trace of a descendant* $\mathbf{d} \in \mathrm{HDesc}_\omega(\mathcal{M})$ *is the intersection of all coalitions* $\mathcal{T} \in \mathcal{C}(\mathcal{M}, \omega)$ *which can construct* $\mathbf{d}$*. A coalition* $\mathcal{T}$ *can construct* $\mathbf{d}$ *provided* $\mathcal{T} \cap B_{i,z} \neq \emptyset$ *for all* $(i, z) \in I_{\mathbf{d}}$*. The intersection of all such coalitions gives all codewords required to construct the descendant* $\mathbf{d}$*, i.e.,*

$$\mathrm{trace}(\mathbf{d}) = \bigcap_{\{\mathcal{T}:\ \mathbf{d} \in \mathrm{Hdesc}(\mathcal{T})\}} \mathcal{T}. \tag{6}$$

It has been noted in [7] that the hybrid attack can be defended against by using codes described in the next section, which have been designed against the erasure attack, for slightly smaller coalition sizes. We don't consider this further in this paper.

4 Vector Space Block Design Codes (VSBDCs)

A VSBDC is a type of Resolvable BIBD [12], constructed by using the set of all $(\omega - 1)-$dimensional quotient vector spaces of a vector space $\mathcal{V}$. For this construction, we need a method of mapping the partitions of a resolvable design to coordinates of the codewords.

Definition 7 (Partition Mapping)**.** *Let A be a finite set with lexicographical ordering $\{a_1, a_2, \ldots, a_n\}$ and let $\mathcal{P}$ be a partition on A with segments $\{S_1, S_2, \ldots, S_m\}$ which ordered by size and lexicographically. The map $\zeta_{\mathcal{P}} : A \mapsto \mathbb{I}_m = \{1, \ldots, m\}$ is defined by*

$$\forall a \in A, a \in S_i \Longleftrightarrow \zeta_{\mathcal{P}}(a) = i. \tag{7}$$

Given a partition mapping $\zeta : A \to \mathbb{I}_m$, we can construct a set of codewords or "*partition vectors*" from a Resolvable BIBD.

Definition 8 (Partition Vectors)**.** *Let A be a finite set with a set of N ordered partitions $\mathcal{B} = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_N\}$. A partition vector is a vector of length N generated from A and $\mathcal{B}$ and is denoted $\mathbf{v}_{\mathcal{B}}(a)$. Each coordinate i of $\mathbf{v}_{\mathcal{B}}(a)$ specifies the segment (by index) that contains the element a in the partition $\mathcal{P}_i \in \mathcal{B}$. i.e,*

$$\mathbf{v}_{\mathcal{B}}(a) = \left(\zeta_{\mathcal{P}_1}(a), \zeta_{\mathcal{P}_2}(a), \ldots, \zeta_{\mathcal{P}}\;\,(a)\right). \tag{8}$$

We can construct a partition vector for each $a \in A$. We call the set of all partition vectors a "block design code".

Definition 9 (Block Design Code)**.** *Given an ordered set of partitions $\mathcal{B} = \{\mathcal{P}_1, \mathcal{P}_1, \ldots, \mathcal{P}_N\}$ of a finite set A, a block design code denoted $\mathrm{BDC}(A, \mathcal{B})$ is the set of all partition vectors $\mathbf{v}_{\mathcal{B}}(a)$ where $a \in A$. i.e.,*

$$\mathrm{BDC}(A, \mathcal{B}) = \{\mathbf{v}_{\mathcal{B}}(a) : a \in A\}. \tag{9}$$

The type of block design code we are interested in will be called a "*vector space block design code*".

Definition 10 (Vector Space Block Design Codes)**.** *A Vector Space Block Design Code (VSBDC), denoted $\mathrm{BDC}(\mathcal{V}, \mathcal{B}_{\mathcal{V},k})$, is a block design code where the set of treatments is a vector space $\mathcal{V}$ and the block design is the set of quotient vector spaces $\mathcal{B}_{\mathcal{V},k}$ of dimension k.*

The set of quotient vector spaces $\mathcal{B}_{\mathcal{V},k}$ is a resolvable balanced incomplete block design (RBIBD). We now give an example of a VSBDC with $\omega = 3$.

Example 2. *Let $\mathcal{V} = Z_2^3$. The VSBDC $\mathcal{M} = \mathrm{BDC}(\mathcal{V}, \mathcal{B}_{\mathcal{V},2})$ is constructed from the set of all vector subspaces of $\mathcal{V}$ with dimension 2, denoted $S_{\mathcal{V},2}$. The codewords in $\mathrm{BDC}(\mathcal{V}, \mathcal{B}_{\mathcal{V},2})$ are*

$$\begin{array}{ll}
\mathbf{v}_{\mathcal{B}}(000) = (1,1,1,1,1,1,1), & \mathbf{v}_{\mathcal{B}}(001) = (1,1,1,2,2,2,2), \\
\mathbf{v}_{\mathcal{B}}(010) = (1,2,2,1,1,2,2), & \mathbf{v}_{\mathcal{B}}(100) = (2,1,2,1,2,1,2), \\
\mathbf{v}_{\mathcal{B}}(011) = (1,2,2,2,2,1,1), & \mathbf{v}_{\mathcal{B}}(101) = (2,1,2,2,1,2,1), \\
\mathbf{v}_{\mathcal{B}}(110) = (2,2,1,1,2,2,1), & \mathbf{v}_{\mathcal{B}}(111) = (2,2,1,2,1,1,2).
\end{array} \tag{10}$$

The set $A_{\mathcal{T}}$, defined below, is the $\mathcal{V}$-representation of the set of partition vectors in the coalition $\mathcal{T}$.

Definition 11. *Consider a coalition $\mathcal{T}$ of ω codewords from the vector space block design code* $\mathrm{BDC}(\mathcal{V}, \mathcal{B}_{\mathcal{V},\omega-1})$. *Then $A_{\mathcal{T}}$ consists of all vectors $\mathbf{a}$ in $\mathcal{V}$ that satisfy: $A_{\mathcal{T}} = \left\{\mathbf{a} : \mathbf{v}_{\mathcal{B}_{\mathcal{V},\ -1}}(\mathbf{a}) \in \mathcal{T},\ \ \mathbf{a} \in \mathcal{V}\right\}$.*

We define the partial trace $B_{i,x}$ as the set which can be used to construct the descendant, but includes codewords which may not have actually taken part in the coalition.

Definition 12. *Let $\mathcal{M}$ be the code* $\mathrm{BDC}(\mathcal{V}, \mathcal{B}_{\mathcal{V},k})$. *We define the partial trace $B_{i,x}$ to be the vectors in $\mathcal{V}$ that construct the codewords in $\mathcal{T}$ instead of the codewords themselves. i.e., $B_{i,x} = \left\{\mathbf{a} : \mathbf{v}^{(i)}_{\mathcal{B}_{\mathcal{V},}}(\mathbf{a}) = x,\ \ \mathbf{a} \in \mathcal{V}\right\}$, where $\mathbf{v}^{(i)}_{\mathcal{B}_{\mathcal{V},}}(\mathbf{a})$ is the i-th coordinate of the partition vector $\mathbf{v}_{\mathcal{B}_{\mathcal{V},}}(\mathbf{a})$.*

We characterize the partial trace below.

Theorem 13. *Let $\mathcal{M}$ be the vector space block design code* $\mathrm{BDC}(\mathcal{V}, \mathcal{B}_{\mathcal{V},\omega-1})$, *where $\mathcal{V} = F_q^n$. Let $\mathcal{T} \in \mathcal{C}(\mathcal{M}, \omega)$. The partial trace of $E_{\mathcal{T}}$ is the coset generator $[A_{\mathcal{T}}]_{\mathcal{V}}$. i.e.,*

$$\mathrm{ptrace}(E_{\mathcal{T}}) = \bigcap_{(i,x)\in E_{\mathcal{T}}} B_{i,x} = [A_{\mathcal{T}}]_{\mathcal{V}}\,. \tag{11}$$

Proof. The erasure profile $E_{\mathcal{T}}$ lists all the blocks that contain the coalition $A_{\mathcal{T}}$. If the coset generator has $|[A_{\mathcal{T}}]_{\mathcal{V}}| = q^{\omega-1}$ elements, then $[A_{\mathcal{T}}]_{\mathcal{V}}$ must be one of the blocks $B_{j,z}$ in the block design $\mathcal{B}_{\mathcal{V},\omega-1}$. Now, each block in $\mathcal{B}_{\mathcal{V},\omega-1}$ is unique, therefore the erasure profile $E_{\mathcal{T}}$ can only contain one pair (j,z), which corresponds to the coset generator $[A_{\mathcal{T}}]_{\mathcal{V}}$. Thus it follows that:

$$\mathrm{ptrace}(E_{\mathcal{T}}) = \bigcap_{(i,x)\in E_{\mathcal{T}}} B_{i,x} = B_{j,z} = [A_{\mathcal{T}}]_{\mathcal{V}}\,. \tag{12}$$

If $|[A_{\mathcal{T}}]_{\mathcal{V}}| = q^k < q^{\omega-1}$ then it must be a vector subspace of $\mathcal{V}$ with dimension k. Let $\langle H\rangle = [A_{\mathcal{T}}]_{\mathcal{V}}$ be that vector subspace. Moreover, every block in the design $\mathcal{B}_{\mathcal{V},\omega-1}$ that contains the vector subspace $\langle H\rangle$, must also be a vector subspace of $\mathcal{V}$, since it must contain the zero vector. All the blocks $B_{i,x}$ that contain $\langle H\rangle$ (which themselves are vector subspaces of $\mathcal{V}$) can be expressed as a direct sum

$$B_{i,x} = \langle H\rangle + \langle T_{i,x}\rangle,\ \ \langle T_{i,x}\rangle \cap \langle H\rangle = \{\underline{0}\}, \tag{13}$$

where $T_{i,x}$ extends $\langle H\rangle$ to $B_{i,x}$. So the set $\{B_{i,x} : (i,x) \in E_{\mathcal{T}}\}$, is the set of all vector subspaces in $\mathcal{V}$ containing the subspace $\langle H\rangle$ with dimension $\omega - 1$. The intersection of all subspaces $B_{i,x}$ is

$$\bigcap_{(i,x)\in E_{\mathcal{T}}} B_{i,x} = \bigcap_{(i,x)\in E_{\mathcal{T}}} (\langle H\rangle + \langle T_{i,x}\rangle) = \langle H\rangle + \bigcap_{(i,x)\in E_{\mathcal{T}}} \langle T_{i,x}\rangle = \langle H\rangle, \tag{14}$$

where intersection of all allowable $\langle T_{i,x}\rangle$ is the trivial vector subspace $\{\underline{0}\}$. □

Note that the VSBDC has the desirable property that any coalition of size ω must have at least one coordinate where all codewords are equal (since every coalition $\mathcal{T}$ must have a non empty erasure profile $E_{\mathcal{T}}$). This property enables a partial trace of the descendant which will contain the coalition plus other codewords (spoof words). We use concatenated codes to get around this problem.

Definition 14 (Concatenated Vector Space Block Design Code)**.** *A CVSBDC $\mathcal{M}$ consists of an inner code $\mathcal{M}_I = \mathrm{BDC}(F_q^N\ , \mathcal{B}_{F\quad ,k})$, an outer code $\mathcal{M}_O \subset F_q^N$ and a scalar preserving homomorphism $\varphi : F_q^N \mapsto (F_q\quad , +)$. Its codewords are constructed as follows:*

$$\mathcal{M} = \left\{ \mathbf{v}_{\mathcal{B}}\left(\varphi^{-1}(f^{(1)})\right), \ldots, \mathbf{v}_{\mathcal{B}}\left(\varphi^{-1}(f^{(N\ )})\right) : \mathbf{f} \in \mathcal{M}_O \right\}$$

where $\mathcal{B}$ is the set of quotient vector spaces $\mathcal{B}_{\mathcal{V},k}$, and
$\mathbf{f} = (f^{(1)}, f^{(2)}, \ldots, f^{(N\ )})$.

We cannot use a linear code as $\mathcal{M}_O$ as this re-introduces the problem of spoof words at the outer code level. In [8, 7], we have introduced δ-nonlinear codes–in the form of modified GRS codes–to get around this problem. These are codes where the sum of δ or fewer codewords is *not a codeword*, and are obtained by applying a special type of permutation polynomial to Generalized Reed Solomon (GRS) codewords–see [6] for details on GRS codes–which are MDS codes attaining the Singleton bound, and have length N, dimension k and minimum distance $N - k + 1$.

Definition 15 (Modified GRS Code)**.** *Let $X = (x_1, x_2 \ldots, x_N)$, where x_i are distinct elements of $\mathbb{F}_q$, and let $\Pi : \mathbb{F}_q[x] \longrightarrow \mathbb{F}_q[x]$. The Modified Generalized Reed-Solomon code, denoted by $\mathrm{MGRS}_k(\mathbb{F}_q, X, \Pi)$, consists of all vectors,*

$$\mathbf{f} = (\Pi(f)(x_1), \Pi(f)(x_2), \ldots, \Pi(f)(x_N))) \,, \tag{15}$$

where $f(x)$ ranges over all polynomials of degree $\leq k$ with coefficients from $\mathbb{F}_q$.

The δ-nonlinearity property ensures that the partial trace of the concatenated code would only contain the coalition $\mathcal{T}$ plus other non-codewords which can be discarded leaving only the coalition–see the concluding section for further discussion. We give an example of a concatenated vector space block design code (CVSBDC) below.

Example 3. *Let $\mathcal{V}$ be the vector space $\mathbb{F}_2^2$. There are 3 vector subspaces $\mathcal{H}_1, \mathcal{H}_2$, $\mathcal{H}_3$ of $\mathcal{V}$ with dimension 1. The set of all quotient vector spaces with dimension 1, $\mathcal{B}_{\mathcal{V},1}$, is:*

$$\mathcal{B}_{\mathcal{V},1} = \left\{ \underbrace{\left\{ \overbrace{\{00, 01\}}^{\mathcal{H}_1}, \{10, 11\} \right\}}_{\mathcal{V}/\mathcal{H}_1}, \underbrace{\left\{ \overbrace{\{00, 10\}}^{\mathcal{H}_2}, \{01, 11\} \right\}}_{\mathcal{V}/\mathcal{H}_2}, \underbrace{\left\{ \overbrace{\{00, 11\}}^{\mathcal{H}_3}, \{01, 10\} \right\}}_{\mathcal{V}/\mathcal{H}_3} \right\}.$$

From $\mathcal{B}_{\mathcal{V},1}$, we generate the inner code $\mathcal{M}_I = \mathrm{BDC}(\mathcal{V}, \mathcal{B}_{\mathcal{V},1})$;

$$\mathcal{M}_I = \left\{\mathbf{v}_{\mathcal{B}\ \ ,1}(00),\ \ \mathbf{v}_{\mathcal{B}\ \ ,1}(01),\ \ \mathbf{v}_{\mathcal{B}\ \ ,1}(10),\ \ \mathbf{v}_{\mathcal{B}\ \ ,1}(11)\right\} = \{111,\ 122,\ 212,\ 221\}\,.$$

Here $\mathbf{v}_{\mathcal{B}\ \ ,1}(01) = 122$ is the partition vector of $01 \in \mathcal{V}$. Now, let $\mathcal{M}_O \subseteq \mathbb{F}_4^4$ be the outer code with a codeword $\mathbf{f} = (0, \alpha, 1, \alpha^2)$ (α is a primitive element of $\mathbb{F}_4$). A suitable choice of scalar preserving homomorphism $\varphi : \mathbb{F}_2^2 \longrightarrow (\mathbb{F}_4, +)$ is:

$x \in \mathbb{F}_2^2$	$\varphi(x) \in \mathbb{F}_{2^2}$	$\varphi(x)$ as α^i
00	0	0
01	1	α^0
10	α	α^1
11	$\alpha + 1$	α^2

(16)

We construct the concatenated codeword as follows: The first coordinate of $\mathbf{f}$ is $f^{(1)} = 0$, applying the inverse of φ we get the vector 00. The corresponding partition vector is $\mathbf{v}_{\mathcal{B}_{\mathcal{V},1}}(00) = 111$. The second, third and fourth coordinates give us the partition vectors 212, 122 and 221. The concatenated codeword is then (111, 212, 122, 221). We can apply this process to all the codewords in $\mathcal{M}_O$ to construct $\mathcal{M}$.

Given a concatenated code $\mathcal{M}$, and a descendant $\mathbf{d} \in \mathrm{EDesc}_\omega(\mathcal{M})$, what is the partial trace of the erasure profile? If we view the concatenated code at the inner level, then the partial trace of the concatenated code is simply the sequence of partial traces for each inner code segment. We call this the "*concatenated partial trace*".

Definition 16 (Concatenated Partial Trace)**.** *Let $\mathcal{M}$ be a CVSBDC with outer code $\mathcal{M}_O$ and inner code $\mathcal{M}_I$. Let $\mathcal{T} \in \mathcal{C}(\mathcal{M}, \omega)$ be a coalition of the concatenated code. Let $\mathcal{T}_O$ be the codewords in $\mathcal{M}_O$ that generate the concatenated codewords in $\mathcal{T}$. Let $\mathcal{T}^{(i)} \in \mathcal{C}(\mathcal{M}_I, \omega)$ be the coalition of inner codewords that are inserted into each codeword of $\mathcal{T}_O$ at the i-th coordinate. The concatenated partial trace of the erasure profile $E_\mathcal{T}$ denoted* $\mathrm{Cptrace}(E_\mathcal{T})$*, is the sequence of N_O partial traces for each inner coalition $\mathcal{T}^{(i)}$. i.e.,* $\mathrm{Cptrace}(E_\mathcal{T}) = (\mathrm{ptrace}(E_{\mathcal{T}^{(1)}}), \ldots, \mathrm{ptrace}(E_{\mathcal{T}^{(\ \ \ )}}))$.

Note that, if the inner code $\mathcal{M}_I$ is a VSBDC, then by Theorem 13 the concatenated partial trace is $\mathrm{Cptrace}(E_\mathcal{T}) = \left([\mathcal{T}^{(1)}]_{\mathbb{F}\ \ }, \ldots, [\mathcal{T}^{(N\ \ )}]_{\mathbb{F}\ \ }\right)$. The concatenated partial trace is useful for viewing the code at the inner level; at the outer level, we need to use a "*coset image*".

Definition 17 (Coset Image)**.** *Consider $\mathbb{F}_q^n$ and $\mathbb{F}_{q\ \ }$. For a given scalar preserving homomorphism $\varphi : \mathbb{F}_q^n \longrightarrow (\mathbb{F}_{q\ \ }, +)$ and a given subset $A \subseteq \mathbb{F}_{q\ \ }$, a coset image $[A]_\varphi$ is $[A]_\varphi = \varphi\left([\varphi^{-1}(A)]_{\mathbb{F}\ \ }\right)$ is:*

$$[A]_\varphi = \left\{a + \sum_{b \in A} \lambda_b (b - a) : \lambda_b \in \mathbb{F}_q, b \in A\right\}, \text{ for any } a \in A, \tag{17}$$

where $\varphi^{-1}(A) = \{\varphi^{-1}(a) : a \in A\} \subset \mathbb{F}_q^n$.

The components λ_b are over the subfield $\mathbb{F}_q$ and not the field $\mathbb{F}_q$. The coset image contains $\leq q^{|A|-1}$ elements of $\mathbb{F}_q$. We now show that we can construct the concatenated partial trace by using a coset image on the outer code $\mathcal{M}_O$.

Definition 18 (Coset Generator)**.** *Let $\mathcal{V}$ be an n–dimensional vector space over the finite field $\mathbb{F}_q$. Also, let $A \subset \mathcal{V}$ with $\mathbf{v} \in A$. The set defined $[A]_{\mathcal{V},\mathbf{v}}$ consists of all vectors $\mathbf{v} + \langle A - \mathbf{v} \rangle$, where $\langle A - \mathbf{v} \rangle$ is a vector subspace of $\mathcal{V}$ generated by the span of $A - \mathbf{v}$. The set of vectors $[A]_{\mathcal{V},\mathbf{v}}$ is a coset of the vector subspace $\langle A - \mathbf{v} \rangle$.*

Theorem 19. *Let $\mathcal{M}$ be a CVSBDC as defined in Definition 14. Let $\mathcal{T} \in \mathcal{C}(\mathcal{M}, \omega)$ be a coalition and let $\mathcal{T}_O$ and $\mathcal{T}^{(i)}$ be as defined in Definition 14. The coset image $[\mathcal{T}_O]_\varphi$ can generate every partial trace of the erasure profiles $E_{\mathcal{T}(\ )}$ via* $\operatorname{ptrace}(E_{\mathcal{T}(\ )}) = [\mathcal{T}^{(i)}]_{\mathbb{F}} \;\; = \varphi^{-1}(\operatorname{spec}([\mathcal{T}_O]_\varphi, i))$.

Proof. The coalitions $\mathcal{T}^{(i)}$ are obtained from the spectrum of $\mathcal{T}_O$ at the i-th coordinate;

$$\mathcal{T}^{(i)} = \varphi^{-1}(\operatorname{spec}(\mathcal{T}_O, i)), \;\; 1 \leq i \leq N_O. \tag{18}$$

Using the definitions of a coset generator (see Definition 18), a coset image (see Definition 17) and some algebraic manipulation, we obtain:

$$\begin{aligned}[\mathcal{T}^{(i)}]_{\mathbb{F}} &= \left[\varphi^{-1} \operatorname{spec}(\mathcal{T}_O, i)\right]_{\mathbb{F}} \\ &= \varphi^{-1}\varphi\left[\varphi^{-1} \operatorname{spec}(\mathcal{T}_O, i)\right]_{\mathbb{F}} \\ &= \varphi^{-1}\left[\operatorname{spec}(\mathcal{T}_O, i)\right]_{\varphi} \\ &= \varphi^{-1} \operatorname{spec}([\mathcal{T}_O]_\varphi, i)\end{aligned} \tag{19}$$

This gives the required result. □

We now have a coset image $[\mathcal{T}_O]_\varphi$ that can be used to construct the concatenated partial trace in Definition 16. But for it to be of any use for tracing purposes, we require that it be the *only* coset image that constructs the concatenated partial trace.

Theorem 20. *Let $\mathcal{M}_O$ be a linear code with codelength N_O and maximum sharing $\mathrm{s}_{\mathcal{M}}$. For any two coalitions $\mathcal{T}_O, \mathcal{T}'_O \in \mathcal{C}(\mathcal{M}, \omega)$, if the codelength $N_O > q^\omega \mathrm{s}_{\mathcal{M}}$ and* $\operatorname{spec}([\mathcal{T}_O]_\varphi, i) = \operatorname{spec}([\mathcal{T}'_O]_\varphi, i)$ *for $1 \leq i \leq N_O$, then $[\mathcal{T}_O]_\varphi = [\mathcal{T}'_O]_\varphi$.*

Proof. The coset images $[\mathcal{T}_O]_\varphi, [\mathcal{T}'_O]_\varphi$ contain up to $q^{\omega-1}$ codewords (see Definition 17). Let $N_O > q^{\omega-1}\mathrm{s}_{\mathcal{M}}$, $[\mathcal{T}_O]_\varphi \neq [\mathcal{T}'_O]_\varphi$ and $\operatorname{spec}([\mathcal{T}_O]_\varphi, i) = \operatorname{spec}([\mathcal{T}'_O]_\varphi, i)$ for $1 \leq i \leq N_O$. Then there must be a codeword $\mathbf{v} \in [\mathcal{T}_O]_\varphi \backslash [\mathcal{T}'_O]_\varphi$. We immediately have that $\operatorname{share}([\mathcal{T}'_O]_\varphi, \mathbf{v}) = N_O$. But, since $\mathbf{v} \notin [\mathcal{T}'_O]_\varphi$ we can show by a simple counting argument that $\operatorname{share}([\mathcal{T}'_O]_\varphi, \mathbf{v}) \leq q^{\omega-1}\mathrm{s}_{\mathcal{M}}$. But $N_O > q^{\omega-1}\mathrm{s}_{\mathcal{M}}$ therefore we have a contradiction: it must be concluded that $[\mathcal{T}_O]_\varphi = [\mathcal{T}'_O]_\varphi$. □

5 Conclusions and Discussion

We have investigated *powerful attacks against fingerprinting schemes* used for securing digital content and described *new types of codes which can be used to*

Table 1. Some suitable Permutation Polynomials for δ-Nonlinear Codes. Here $\mathbb{F}_2$ is the alphabet, δ is the nonlinearity threshold, $\mathrm{s}_{\mathcal{M}}$ is the maximum degree of the MGRS code, $|\mathcal{M}|$ is the number of codewords and $\xi(x)$ is the minimum degree permutation polynomial found which can be used to construct a δ-nonlinear code, as in Theorem 9 of [8].

$\mathbb{F}_2$	δ	$\mathrm{s}_{\mathcal{M}}$	$\|\mathcal{M}\|$	$\xi(x)$
$\mathbb{F}_{2^4}$	3	1	2^8	$x^6 + \alpha^4 x^5 + \alpha^{12} x^3 + \alpha^{10} x^2 + x$
$\mathbb{F}_{2^5}$	3	4	2^{25}	$x^3 + x^2 + x$
$\mathbb{F}_{2^5}$	4	2	2^{15}	$x^5 + x^3 + x$
$\mathbb{F}_{2^6}$	3	4	2^{30}	$x^{11} + x^{10} + x^9 + x^8 + x^3 + x^2 + x$
$\mathbb{F}_{2^6}$	5	3	2^{24}	$x^{11} + x^{10} + x^9 + x^8 + x^3 + x^2 + x$
$\mathbb{F}_{2^7}$	3	20	2^{147}	$x^3 + x^2 + x$
$\mathbb{F}_{2^7}$	5	7	2^{56}	$x^5 + x^4 + x^3 + x^2 + x$
$\mathbb{F}_{2^7}$	7	3	2^{28}	$x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x$

defend against such attacks. We have made use of resolvable BIBDs (Balanced Incomplete Block Designs) in order to construct new code families we call VS-BDC (Vector Space Block Design) codes. These codes, used with concatenation techniques, are resistant to narrow, erasure, and as far as experimentally observed, to hybrid attacks.

There are efficient algorithms for decoding Reed Solomon codes, which could be applied to the δ-nonlinear Generalized Reed Solomon codes. The algorithms for decoding the BIBD based component of the concatenated codes could be based on linear algebra operations for computing cosets and subspaces, and thus have a priori complexity no worse than a small power of the codelength, which is given by an appropriate $q-$binomial coefficient. More precisely, if $\mathcal{V}$ is an n-dimensional vector space over the field $\mathbb{F}_q$, the number of vector subspaces with dimension k, $|S_{\mathcal{V},k}|$, is given by the q-Binomial Coefficient:

$$|S_{\mathcal{V},k}| = \begin{bmatrix} n \\ k \end{bmatrix}_q = \prod_{i=0}^{k-1} \frac{q^{n-i} - 1}{q^{i+1} - 1}, \tag{20}$$

and a small power–between 2 and 3–of this quantity will determine the maximum complexity of the operations used–see also Definition 14. It remains an open problem to obtain more efficient decoding algorithms for the design based codes.

One question raised during the review process of this paper is the complexity of generating the VSBDCs and the δ-nonlinear codes–which together generate the CVSBDCs. The VSBDCs can be generated recursively by generating all the relevant vector subspaces, hence by complexity essentially proportional to the size of the above q-Binomial coefficient. It has turned out that a set of special permutation polynomials, which are used to obtain the δ-nonlinear codes can also be easily generated. More precisely, for the MGRS codes in Definition 15, we can use the mapping

$$\Pi : f(x) \mapsto \frac{\xi(f(x) \cdot x)}{x} \tag{21}$$

where ξ is a special type of permutation polynomial, in order to preserves the distance between all modified codewords and obtain the desired δ-nonlinearity property. We omit the details due to the limitations on space but exhibit a small table of such polynomials, which are obtained by a simple algorithmic procedure and after an efficient randomized procedure. See section V in [8] for more details. Here, we display a table with suitable choices of permutation polynomials to construct δ-nonlinear codes.

The main contribution of this paper is an explicit–nonrandomized–coding construction for addressing erasure and other attacks in digital fingerprinting.

Acknowledgment

The authors would like to thank the referees whose questions and comments contributed to improving the paper. The first author was supported by an APA scholarship from the Australian Research Council.

References

1. A. Barg, G.R. Blakely, and G.A. Kabatiansky. Digital finger printing codes: problem statements, constructions, identification of traitors. *IEEE Trans. Inform. Theory*, 49:852–865, 2003.
2. A. Barg, G. Cohen, S. Encheva, G. Kabatiansky, and G. Zemor. A hypergraph approach to the identifying parent property: the case of multiple parents. *SIAM Journal of Discrete Math.*, 14(3):423–431, 2001.
3. D. Boneh and M. Franklin. An efficient public key traitor tracing scheme. *Lecture Notes in Computer Science*, 1666:338–353, 1999.
4. D. Boneh and J. Shaw. Collusion-secure fingerprinting for digital data. *Lecture Notes in Computer Science*, 963:452–465, 1995.
5. B. Chor, A. Fiat, and M. Naor. Tracing traitors. *Lecture Notes in Computer Science*, 839:257–270, 1999.
6. F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error Correcting Codes.* North-Holland Publishing Company., 1977.
7. S. McNicol. *Traitor Tracing Using Generalized Reed-Solomon Codes and Combinatorial Designs.* PhD thesis, RMIT University, July 2005.
8. S. McNicol, S. Boztaş, and A. Rao. Traitor tracing against powerful attacks. *Proceedings of the IEEE International Symposium on Information Theory*, pages 1878–1882, 2005.
9. J. Staddon, D. R. Stinson, and R. Wei. Combinatorial properties of frameproof and traceability codes. *IEEE Transactions on Information Theory*, 47:1042–1049, 2001.
10. D. Stinson, T. van Trung, and R. Wei. Secure frameproof codes, key distribution patterns, group testing algorithms and related structures. *Planning and Inference*, 86(2):595–617, 2000.
11. Gábor Tardos. Optimal probabilistic fingerprint codes. In *STOC '03: Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 116–125, New York, NY, USA, 2003. ACM Press.
12. W.D. Wallis. *Combinatorial Designs.* Marcel Decker Inc., 1988.

New Bounds on the Capacity of Multi-dimensional RLL-Constrained Systems

Moshe Schwartz and Alexander Vardy

University of California San Diego, La Jolla, CA 92093-0407, USA
moosh@everest.ucsd.edu, vardy@kilimanjaro.ucsd.edu

Abstract. We examine the well-known problem of determining the capacity of multi-dimensional run-length-limited constrained systems. By recasting the problem, which is essentially a combinatorial counting problem, into a probabilistic setting, we are able to derive new lower and upper bounds on the capacity of $(0,k)$-RLL systems. These bounds are better than all previously-known bounds for $k \geqslant 2$, and are even tight asymptotically. Thus, we settle the open question: what is the rate at which the capacity of $(0,k)$-RLL systems converges to 1 as $k \to \infty$? While doing so, we also provide the first ever non-trivial upper bound on the capacity of general (d,k)-RLL systems.

1 Introduction

A (d,k)-RLL constrained system is the set of all binary sequences in which every two adjacent 1's are separated by at least d zeroes, and no more than k 0's appear consecutively. The study of these systems was initiated by Shannon [10, 11] who defined the capacity of a constrained system S as

$$\mathsf{cap}(S) = \lim_{n\to\infty} \frac{\log_2 |S(n)|}{n} \quad ,$$

where $S(n)$ denotes the number of sequences of S of length exactly n.

Constrained systems are widely used today in all manners of storage systems [7, 8]. However, the emergence of two-dimensional recording systems brought to light the need for two-dimensional and even multi-dimensional constrained systems. A two-dimensional (d,k)-RLL constrained system is the set of all binary arrays in which every row and every column obeys the one-dimensional (d,k)-RLL constraint. The generalization to the D-dimensional case is obvious, and we denote such a system as $S^D_{d,k}$. Though we consider in this paper only symmetrical constrains, i.e., the same d and k along every dimension, the results generalize easily to asymmetrical RLL constraints as well.

In the one-dimensional case it is well known that $\mathsf{cap}(S^1_{d,k})$, for $0 \leqslant d \leqslant k$, is the logarithm in base 2 of the largest positive root of the polynomial

$$x^{k+1} - x^{k-d} - x^{k-d-1} - \cdots - x - 1 \ .$$

However, unlike the one-dimensional case, almost nothing is known about the two-dimensional case, and even less in the multi-dimensional case. In [1], Calkin

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 225–234, 2006.

and Wilf gave a numerical estimation method for the capacity of the two-dimensional (0, 1)-RLL constraint which gives,

$$0.5878911617 \leqslant \mathsf{cap}(S^2_{0,1}) \leqslant 0.5878911618 \ .$$

Their method ingeniously uses the fact that the transfer matrix is symmetric, but unfortunately, this happens only for the case of (0, 1)-RLL (and by inverting all the bits, the equivalent $(1, \infty)$-RLL case). Using the same method in the three-dimensional case, it was shown in [9] that

$$0.522501741838 \leqslant \mathsf{cap}(S^3_{0,1}) \leqslant 0.526880847825 \ .$$

Some general bounds on the capacity were given in [5]. Using bit-stuffing encoders, the best known lower bounds on two-dimensional (d, ∞)-RLL were shown in [2]. Amazingly, we still do not know the exact capacity of the multi-dimensional RLL-constraint except when it is zero [3].

The bounds we improve upon in this work are those of two-dimensional $(0, k)$-RLL, $k \geqslant 2$. These are given in the following three theorems:

Theorem 1 (Theorem 3, [5]). *For every positive integer k,*

$$\mathsf{cap}(S^2_{0,k}) \geqslant 1 - \frac{1 - \mathsf{cap}(S^2_{0,1})}{\lceil k/2 \rceil} \ .$$

Theorem 2 ([12]). *For all integers $k \geqslant 8$,*

$$\mathsf{cap}(S^2_{0,k}) \geqslant 1 + \frac{\log_2(1 - (\lfloor k/2 \rfloor + 1)2^{-(\lfloor k/2 \rfloor - 1)})}{(\lfloor k/2 \rfloor + 1)^2} \ .$$

Theorem 3 (Theorem 7, [5]). *For every positive integer k,*

$$\mathsf{cap}(S^2_{0,k}) \leqslant 1 - \frac{1}{k+1} \log_2 \left(\frac{1}{1 - 2^{-(k+1)}} \right) \ .$$

Our new bounds are given in Theorem 6 and Theorem 13. A numerical comparison with the previously-best bounds for $2 \leqslant k \leqslant 10$ is given in Table 1. Furthermore, our lower and upper bounds agree asymptotically, thus settling the open question of the rate of convergence to 1 of $\mathsf{cap}(S^D_{0,k})$ as $k \to \infty$ by showing it to be $\frac{D \log_2 e}{4 \cdot 2}$.

Our approach to the problem of bounding the capacity is to recast the problem from a combinatorial counting problem to a probability bounding problem. Suppose we randomly select a sequence of length n with uniform distribution. Let A^S_n denote the event that this sequence is in the constrained system S. Then the total number of sequences in S of length n may be easily written as

$$|S(n)| = \Pr[A^S_n] \cdot 2^n \ .$$

It follows that

$$\mathsf{cap}(S) = \lim_{n \to \infty} \frac{\log_2 |S(n)|}{n} = \lim_{n \to \infty} \frac{\log_2(\Pr[A^S_n]2^n)}{n} = \lim_{n \to \infty} \frac{\log_2 \Pr[A^S_n]}{n} + 1 \ .$$

Table 1. Comparison of lower bounds (LB) and upper bounds (UB) on $\mathsf{cap}(S^2_{0,k})$, for $2 \leqslant k \leqslant 10$. Lower and upper bounds are rounded down and up, respectively, to six decimal digits.

k	LB by [5]	LB by [12]	LB by Theorem 6	UB by Theorem 13	UB by [5]
2	0.587891		0.758292	0.904373	0.935785
3	0.793945		0.893554	0.947949	0.976723
4	0.793945		0.950450	0.970467	0.990840
5	0.862630		0.976217	0.983338	0.996214
6	0.862630		0.988383	0.990816	0.998384
7	0.896972		0.994268	0.995068	0.999295
8	0.896972	0.943398	0.997155	0.997410	0.999687
9	0.917578	0.943398	0.998583	0.998663	0.999860
10	0.917578	0.981164	0.999293	0.999318	0.999936

This translates in a straightforward manner to higher dimensions as well. By calculating or bounding $\Pr[A^S_n]$, we may get the exact capacity or bounds on it, which is the basis for what is to follow.

The work is organized as follows. In Section 2 we use monotone families to achieve lower bounds on $\mathsf{cap}(S^D_{0,k})$ and an upper bound on $\mathsf{cap}(S^D_{d,k})$. While this method may also be used to lower bound $\mathsf{cap}(S^D_{d,\infty})$, the resulting bound is extremely weak. We continue in Section 3 by deriving an upper bound on $\mathsf{cap}(S^D_{0,k})$ using a large-deviation bound for sums of nearly-independent random variables. We conclude in Section 4 by discussing the asymptotics of our new bounds and comparing them with the case of (d,∞)-RLL.

2 Bounds from Monotone Families

We can use monotone increasing and decreasing families to find new lower bounds on the capacity of $(0,k)$-RLL, and a new upper bound on the capacity of (d,k)-RLL, $d \geqslant 1$. We start with the definition of these families.

Definition 4. *Let $n > 0$ be some integer, and $[n]$ denote the set $\{1,2,\dots,n\}$. A family $\mathcal{F} \subseteq 2^{[n]}$ is said to be* monotone increasing *if when $A \in F$ and $A \subseteq A' \subseteq [n]$, then $A' \in F$. It is said to be* monotone decreasing *if when $A \in F$ and $A' \subseteq A$, then $A' \in F$.*

The following theorem is due to Kleitman [6]:

Theorem 5. *Let $\mathcal{A}$, $\mathcal{B}$ be monotone increasing families, and $\mathcal{C}$, $\mathcal{D}$ be monotone decreasing families. Let X be a random variable describing a uniformly-distributed random choice of subset of $[n]$ out of the 2^n possible subsets. Then,*

$$\Pr[X \in \mathcal{A} \cap \mathcal{B}] \geqslant \Pr[X \in \mathcal{A}] \cdot \Pr[X \in \mathcal{B}] \ , \tag{1}$$

$$\Pr[X \in \mathcal{C} \cap \mathcal{D}] \geqslant \Pr[X \in \mathcal{C}] \cdot \Pr[X \in \mathcal{D}] \ , \tag{2}$$

$$\Pr[X \in \mathcal{A} \cap \mathcal{C}] \leqslant \Pr[X \in \mathcal{A}] \cdot \Pr[X \in \mathcal{C}] \ . \tag{3}$$

We can now apply Kleitman's theorem to $(0,k)$-RLL constrained systems:

Theorem 6. *For all integers $k \geqslant 0$, $\mathsf{cap}(S^2_{0,k}) \geqslant 2\mathsf{cap}(S^1_{0,k}) - 1$.*

Proof. The constrained system we examine is $S = S^2_{0,k}$, and with our notation, A^S_n denotes the event that a randomly chosen $n \times n$ array is $(0,k)$-RLL.

We now define two closely related constraints. Let R denote the set of two-dimensional arrays in which every **row** is $(0,k)$-RLL, and C denote the set of two-dimensional arrays in which every **column** is $(0,k)$-RLL. Similarly we define the events A^R_n and A^C_n. By definition,

$$A^S_n = A^R_n \cap A^C_n \ .$$

It is easy to verify that both constraints R and C are monotone increasing families. Hence, by Theorem 5,

$$\Pr[A^S_n] = \Pr[A^R_n \cap A^C_n] \geqslant \Pr[A^R_n] \Pr[A^C_n] \ .$$

It follows that,

$$\mathsf{cap}(S) = \lim_{n\to\infty} \frac{\log_2 \Pr[A^S_n]}{n^2} + 1 \geqslant \lim_{n\to\infty} \frac{\log_2(\Pr[A^R_n]\Pr[A^C_n])}{n^2} + 1 \ . \qquad (4)$$

Now, both $\Pr[A^R_n]$ and $\Pr[A^C_n]$ may be easily expressed in terms of one-dimensional constrained systems. An $n \times n$ binary array chosen randomly with uniform distribution is equivalent to a set of n^2 i.i.d. random variables for each of the array's bits, each having a "1" with probability 1/2. Thus,

$$\Pr[A^R_n] = \Pr[A^C_n] = \left(\Pr[A^{S'}_n]\right)^n \ ,$$

where $S' = S^1_{0,k}$ is the one-dimensional $(0,k)$-RLL constraint. Plugging this back into (4) we get

$$\mathsf{cap}(S^2_{0,k}) \geqslant \lim_{n\to\infty} \frac{2\log_2 \Pr[A^{S'}_n]}{n} + 1 = 2\mathsf{cap}(S^1_{0,k}) - 1 \ .$$

□

This is generalized to higher dimensions in the following corollary.

Corollary 7. *Let $D_1, D_2 \geqslant 1$ be integers, then*

$$\mathsf{cap}(S^{D_1+D_2}_{0,k}) \geqslant \mathsf{cap}(S^{D_1}_{0,k}) + \mathsf{cap}(S^{D_2}_{0,k}) - 1 \ .$$

We note that similar lower bounds may be given for the (d,∞)-RLL constraint, since such arrays form a monotone decreasing family. However, the resulting bounds are very weak. We can also mix monotone increasing and decreasing families to get the following result.

Theorem 8. *Let $D \geqslant 1$ be some integer, and $k \geqslant d$ also integers, then*

$$\mathsf{cap}(S^D_{d,k}) \leqslant \mathsf{cap}(S^D_{d,\infty}) + \mathsf{cap}(S^D_{0,k}) - 1 \ .$$

Proof. Omitted. □

3 New Upper Bounds

In this section we present upper bounds on the capacity of $(0,k)$-RLL. Unlike the previous section, these bounds are explicit. For this purpose we introduce a new probability bound. It is derived from the bound by Janson [4], but by requiring some symmetry, which applies in our case, we can make the bound stronger.

Suppose that ξ_i, $i \in [n]$, is a family of independent 0–1 random variables. Let $\mathcal{S} \subseteq [n]^{\leqslant k}$, where $[n]^{\leqslant k}$ denotes the set of all subsets of $[n]$ of size at most k. We then define the following indicator random variables,

$$I_A = \begin{cases} \prod_{i \in A} \xi_i & A \in \mathcal{S} \ , \\ 0 & A \notin \mathcal{S} \ . \end{cases}$$

For $A, B \in \mathcal{S}$, we denote $A \sim B$ if $A \neq B$ and $A \cap B \neq \emptyset$. Let $X = \sum_{A \in \mathcal{S}} I_A$, and define

$$\varDelta = \sum_{A} \sum_{B \sim A} \Pr[I_A = 1 \wedge I_B = 1] \ .$$

Janson [4] gave the following bound:

Theorem 9. *With the setting as defined above, let $\mu = E(X) = \sum_A E(I_A)$, then*

$$\Pr[X = 0] \leqslant e^{-\frac{\ 2}{\ +\ }} \ .$$

Our goal is to use Theorem 9 to show an upper bound on the capacity of two-dimensional $(0,k)$-RLL systems. If $S(n,m)$ denotes the number of two-dimensional $(0,k)$-RLL arrays of size $n \times m$ then by definition,

$$\mathsf{cap}(S^2_{0,k}) = \lim_{n,m \to \infty} \frac{\log_2 |S(n,m)|}{nm} \ .$$

However, it would be more convenient to work in a more symmetric setting. In a sense, positions which are close enough to the edge of the array are "less constrained" than others lying within the array. We overcome this difficulty by considering cyclic $(0,k)$-RLL arrays.

We say that a binary $n \times m$ array $\mathcal{A}$ is *cyclic* $(0,k)$*-RLL* if there does not exist $0 \leqslant i \leqslant n-1$, $0 \leqslant j \leqslant m-1$ such that $\mathcal{A}_{i,j} = \mathcal{A}_{i+1,j} = \cdots = \mathcal{A}_{i+k,j} = 0$ or $\mathcal{A}_{i,j} = \mathcal{A}_{i,j+1} = \cdots = \mathcal{A}_{i,j+k} = 0$, where the indices are taken modulo n and m respectively. We denote the set of all such $n \times m$ arrays as $S_{\mathrm{c}}(n,m)$. The next lemma shows that by restricting ourselves to cyclic $(0,k)$-RLL arrays, we do not change the capacity.

Lemma 10. *For all positive integers k,*

$$\mathsf{cap}(S^2_{0,k}) = \lim_{n,m \to \infty} \frac{\log_2 |S_{\mathrm{c}}(n,m)|}{nm} \ .$$

Proof. Omitted. □

We start by considering a random $n \times n$ binary array, chosen with uniform distribution, which is equivalent to saying that we have an array of n^2 i.i.d. 0–1 random variables $\xi_{i,j}$, $0 \leqslant i, j \leqslant n-1$, with $\xi_{i,j} \sim Be(1/2)$.

For the remainder of this section, we invert the bits of the array, or equivalently, we say that an array is $(0,k)$-RLL if it does not contain $k+1$ consecutive 1's along any row or column. Furthermore, by Lemma 10, we consider only cyclic $(0,k)$-RLL arrays. Suppose we define the following subsets of coordinates of the arrays:

$$\begin{aligned}
\mathcal{S}_{\mathrm{V}} &= \{\{(i,j),(i+1,j),\dots,(i+k,j)\} \mid 0 \leqslant i,j \leqslant n-1\} \ , \\
\mathcal{S}_{\mathrm{H}} &= \{\{(i,j),(i,j+1),\dots,(i,j+k)\} \mid 0 \leqslant i,j \leqslant n-1\} \ , \\
\mathcal{S} &= \mathcal{S}_{\mathrm{V}} \cup \mathcal{S}_{\mathrm{H}} \ ,
\end{aligned}$$

where all the coordinates are taken modulo n. We now define the following indicator random variables

$$I_A = \prod_{(i,j)\in A} \xi_{i,j} \qquad\qquad \text{for all } A \in \mathcal{S} \ .$$

If $I_A = 1$ for some $A \in \mathcal{S}$, we have a forbidden event of $k+1$ consecutive 1's along a row or a column. Finally, we count the number of forbidden events in the random array by defining $X = \sum_{A\in\mathcal{S}} I_A$. It is now clear that the probability that this random array is $(0,k)$-RLL is simply

$$\Pr[A_n^{S^2_{0,}}\ ] = \Pr[X=0] \ .$$

It is easy to be convinced that this setting agrees with the requirements of Theorem 9. All we have to do now to upper bound $\Pr[X=0]$, is to calculate μ and Δ. We note that X is the sum of $2n^2$ indicator random variables, so by linearity of expectation,

$$\mu = E(X) = \frac{1}{2^{k+1}} \cdot 2n^2 = \frac{n^2}{2^k} \ ,$$

since each of the indicator random variables has probability exactly $1/2^{k+1}$ of being 1. Calculating Δ is equally easy,

$$\begin{aligned}
\Delta &= \sum_A \sum_{B\sim A} \Pr[I_A = 1 \wedge I_B = 1] = 2n^2 \left((k+1)^2 \frac{1}{2^{2k+1}} + 2\sum_{i=1}^{k} \frac{1}{2^{k+1+i}} \right) \\
&= n^2 \left(\frac{(k+1)^2}{2^{2k}} + \frac{2}{2^k}\left(1 - \frac{1}{2^k}\right)\right) \ .
\end{aligned}$$

By Theorem 9,

$$\Pr[X=0] \leqslant e^{-\frac{2}{\ +\ }} = e^{-\frac{2}{3\cdot 2\ \ +(\ \ +1)^2-2}} \ ,$$

which immediately gives us:

$$\mathsf{cap}(S^2_{0,k}) \leqslant 1 - \frac{\log_2 e}{3\cdot 2^k + (k+1)^2 - 2} \ . \qquad (5)$$

The bound of (5) is already better than the best known bounds for $k \geqslant 2$ given in [5]. But we can do even better by improving the bound of Theorem 9. This is achieved by assuming some more symmetry than the general setting of the theorem. Given some $A \in \mathcal{S} \subseteq [n]^{\leqslant k}$, let $X_A = I_A + \sum_{B \sim A} I_B$. We define

$$\varGamma_A = \sum_i \frac{\Pr[X_A = i \mid I_A = 1]}{i} \quad .$$

If $\varGamma_A$ does not depend on the choice of $A \in \mathcal{S}$, we simply denote it as $\varGamma$.

Theorem 11. *With the setting as defined above, let $\mu = E(X) = \sum_A E(I_A)$. If the distribution of X_A given $I_A = 1$ does not depend on the choice of A, then*

$$\Pr[X = 0] \leqslant e^{-\mu\varGamma} \quad .$$

Proof. Omitted. □

It is obvious that the symmetry requirements of Theorem 11 hold in our case. So now, in order to apply Theorem 11 we have to calculate $\varGamma$, which is a little more difficult than calculating $\varDelta$. Since $\varGamma$ does not depend on the choice of A, we arbitrarily choose the horizontal set of coordinates

$$A = \{(0,0), (0,1), \ldots, (0,k)\} \quad .$$

We now have to calculate $\Pr[X_A = i \mid I_A = 1]$. We note that we can partition the set $\{B \mid B \sim A\}$ into the following disjoint subsets:

$$\{B \mid B \sim A\} = \mathcal{S}_{\mathrm{HL}} \cup \mathcal{S}_{\mathrm{HR}} \cup \mathcal{S}_{\mathrm{V},0} \cup \mathcal{S}_{\mathrm{V},1} \cup \cdots \cup \mathcal{S}_{\mathrm{V},k} \quad ,$$

where

$$\begin{aligned}
\mathcal{S}_{\mathrm{HL}} &= \{B \in \mathcal{S}_{\mathrm{H}} - \{A\} \mid (0,0) \in B\} \quad , \\
\mathcal{S}_{\mathrm{HR}} &= \{B \in \mathcal{S}_{\mathrm{H}} - \{A\} \mid (0,k) \in B\} \quad , \\
\mathcal{S}_{\mathrm{V},j} &= \{B \in \mathcal{S}_{\mathrm{V}} \mid (0,j) \in B\} \quad , \qquad\qquad \text{for all } 0 \leqslant j \leqslant k \quad .
\end{aligned}$$

We define $X_{\mathrm{HL}} = \sum_{B \in \mathcal{S}_{\mathrm{HL}}} I_B$, and in a similar fashion, X_{HR} and $X_{\mathrm{V},j}$ for all $0 \leqslant j \leqslant k$. Since the indicators for elements from different subsets are independent given $I_A = 1$ because their intersection contains only coordinates from A, it follows that X_{HL}, X_{HR} and $X_{\mathrm{V},j}$, $0 \leqslant j \leqslant k$, are independent given $I_A = 1$.

The distribution of X_{HL} and X_{HR} given $I_A = 1$ is easily seen to be

$$\Pr[X_{\mathrm{HL}} = i \mid I_A = 1] = \Pr[X_{\mathrm{HR}} = i \mid I_A = 1] = \begin{cases} \frac{1}{2^{\ +1}} & 0 \leqslant i \leqslant k-1 \\ \frac{1}{2} & i = k \end{cases}$$

since the 0 closest to A determines the number of runs of 1's of length $k+1$. We denote

$$f_k^{\|}(i) = 2^k \Pr[X_{\mathrm{HL}} = i \mid I_A = 1] = 2^k \Pr[X_{\mathrm{HR}} = i \mid I_A = 1] \quad .$$

For the distribution of $X_{\mathrm{V},j}$ we need the following lemma.

Lemma 12. *Let $f_k^{\perp}(i)$ denote the number of binary strings of length $2k+1$ with their middle position a 1, and which contain exactly $0 \leqslant i \leqslant k+1$ runs of $k+1$ 1's. Then,*

$$f_k^{\perp}(i) = \begin{cases} 2^{2k} - (k+2)2^{k-1} & i = 0 \\ (k-i+4)2^{k-i-1} & 1 \leqslant i \leqslant k \\ 1 & i = k+1 \ . \end{cases}$$

Proof. Omitted. □

Using this lemma, we can now say that

$$\Pr[X_{\mathrm{V},j} = i \mid I_A = 1] = \frac{f_k^{\perp}(i)}{2^{2k}} \ .$$

Since $X_A = X_{\mathrm{HL}} + X_{\mathrm{HR}} + \sum_{j=0}^{k} X_{\mathrm{V},j} + I_A$, we have that

$$\begin{aligned} &\Pr[X_A = i \mid I_A = 1] \\ &= \sum_{\substack{i\ \ +i\ \ +i_0+\ldots+i\ \ =i-1 \\ 0\leqslant i\ \ ,i\ \ \leqslant k \\ 0\leqslant i_0,\ldots,i\ \ \leqslant k+1}} \Pr[X_{\mathrm{HL}} = i_L \mid I_A = 1] \Pr[X_{\mathrm{HR}} = i_R \mid I_A = 1] \\ &\quad \cdot \prod_{j=0}^{k} \Pr[X_{\mathrm{V},j} = i_j \mid I_A = 1] \ . \end{aligned}$$

It follows that

$$\mathit{\Gamma} = \sum_{i \geqslant 1} \frac{1}{i} \sum_{\substack{i\ \ +i\ \ +i_0+\ldots+i\ \ =i-1 \\ 0\leqslant i\ \ ,i\ \ \leqslant k \\ 0\leqslant i_0,\ldots,i\ \ \leqslant k+1}} \frac{f_k^{\|}(i_L) f_k^{\|}(i_R)}{2^{2k}} \prod_{j=0}^{k} \frac{f_k^{\perp}(i_j)}{2^{2k}} \ . \tag{6}$$

We can now apply Theorem 11 and get that

$$\Pr[X = 0] \leqslant e^{-n^2 \mathit{\Gamma}/2} \quad ,$$

where $\mathit{\Gamma}$ is given by (6). This immediately gives us the following theorem.

Theorem 13. *Let $k \geqslant 1$ be some integer, then*

$$\mathsf{cap}(S_{0,k}^2) \leqslant 1 - \frac{\log_2 e}{2^k} \mathit{\Gamma} \ ,$$

where $\mathit{\Gamma}$ is given by (6)

We can make the bound of Theorem 13 weaker for small values of k, but more analytically appealing for an asymptotic analysis. This is achieved by noting that $f_k^{\perp}(0)/2^{2k}$ is almost 1 for large values of k.

Theorem 14. *Let $k \geqslant 1$ be some integer, then*

$$\mathsf{cap}(S^2_{0,k}) \leqslant 1 - \frac{\log_2 e}{2^k}\left(\frac{1}{2} - \frac{1}{2^{k+1}}\right)(1-(k+2)2^{-(k+1)})^{k+1} \ .$$

Proof. Omitted. □

We can generalize both Theorem 13 and Theorem 14, and for simplicity, show just the latter in the following theorem.

Theorem 15. *Let $D \geqslant 2$ and $k \geqslant 1$ be some integers, then*

$$\mathsf{cap}(S^D_{0,k}) \leqslant 1 - \frac{D\log_2 e}{2 \cdot 2^k}\left(\frac{1}{2} - \frac{1}{2^{k+1}}\right)(1-(k+2)2^{-(k+1)})^{(D-1)(k+1)} \ .$$

4 Conclusion

In this work we showed new lower and upper bounds on the multi-dimensional capacity of $(0,k)$-RLL systems, as well as a new upper bound on the capacity of (d,k)-RLL systems. We conclude with an interesting comparison of the asymptotes of our new bounds with those of the best previously known bounds. We examine the rate of convergence to 1 of $\mathsf{cap}(S^2_{0,k})$ as $k \to \infty$. The best asymptotic bounds were given in [5]:

$$\frac{\log_2 e}{2(k+1)2^k} < 1 - \mathsf{cap}(S^2_{0,k}) \leqslant \frac{4\sqrt{2}\log_2 e}{(k+1)2^{k/2}} + \frac{8}{2^k} \ ,$$

for sufficiently large k. Our bounds, given in Theorem 6 and Theorem 14, show:

$$\frac{\log_2 e}{2^k}\left(\frac{1}{2} - \frac{1}{2^{k+1}}\right)(1-(k+2)2^{-(k+1)})^{k+1} \leqslant 1 - \mathsf{cap}(S^2_{0,k}) \leqslant 2(1 - \mathsf{cap}(S^1_{0,k}))$$

for all integers $k \geqslant 1$. As mentioned in [5], the one-dimensional capacity of $(0,k)$-RLL converges to 1 when $k \to \infty$ as $\frac{\log_2 e}{4 \cdot 2}$. Hence, our lower and upper bounds agree asymptotically and the rate of convergence to 1 of $\mathsf{cap}(S^2_{0,k})$ as $k \to \infty$ is $\frac{\log_2 e}{2 \cdot 2}$. In the D-dimensional case this rate becomes $\frac{D\log_2 e}{4 \cdot 2}$.

It is also interesting to make a comparison with (d,∞)-RLL. While $\mathsf{cap}(S^2_{d,\infty})$ converges to 0 as $\frac{\log_2 d}{d}$, just as it does in one dimension, for D-dimensional $(0,k)$-RLL the capacity converges to 1 slower than the one-dimensional case by a factor of D.

References

1. N. Calkin and H. Wilf. The number of independent sets in the grid graph. *SIAM J. Discrete Math.*, 11:54–60, 1998.
2. S. Halevy, J. Chen, R. M. Roth, P. H. Siegel, and J. K. Wolf. Improved bit-stuffing bounds on two-dimensional constraints. *IEEE Trans. on Inform. Theory*, 50(5):824–838, May 2004.

3. H. Ito, A. Kato, Z. Nagy, and K. Zeger. Zero capacity region of multidimensional run length constraints. *Elec. J. of Comb.*, 6, 1999.
4. S. Janson. Poisson approximation for large deviations. *Random Structures and Algorithms*, 1:221–230, 1990.
5. A. Kato and K. Zeger. On the capacity of two-dimensional run-length constrained channels. *IEEE Trans. on Inform. Theory*, 45:1527–1540, July 1999.
6. D. J. Kleitman. Families of non-disjoint subsets. *J. Combin. Theory*, 1:153–155, 1966.
7. B. H. Marcus, P. H. Siegel, and J. K. Wolf. Finite-state modulation codes for data storage. *IEEE J. Select. Areas Commun.*, 10:5–37, January 1992.
8. Brian H. Marcus, Ron M. Roth, and Paul H. Siegel. *Constrained systems and coding for recording channels.* V. S. Pless and W. C. Huffman (Editors), Elsevier, Amsterdam, 1998.
9. Zsigmond Nagy and Kenneth Zeger. Capacity bounds for the three-dimensional $(0, 1)$ run length limited channel. *IEEE Trans. on Inform. Theory*, 46(3):1030–1033, May 2000.
10. C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, July 1948.
11. C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:623–656, October 1948.
12. R. Talyansky. Coding for two-dimensional constraints. M.Sc. thesis, Computer Science Dep., Technion – Israel Institute of Technology, Haifa, Israel, 1997. (in Hebrew).

LDPC Codes for Fading Channels: Two Strategies

Xiaowei Jin, Teng Li, Thomas E. Fuja, and Oliver M. Collins

University of Notre Dame, Notre Dame, IN 46556, USA
{xjin, tli, tfuja, ocollins}@nd.edu

Abstract. This paper compares two approaches to reliable communication over fading channels using low-density parity check (LDPC) codes. The particular model considered is the block fading channel with independent channel realizations. The first approach uses a block interleaver to create independent sub-channels that are encoded using irregular LDPC codes with rates specified by the appropriate capacity values; this first approach uses a decision feedback structure wherein decoded data are used as pilots to estimate the channel prior to the next round of decoding. The second approach uses a combined graph describing both the channel and the code to iteratively estimate the channel and decode data. For comparable channels, it is shown that the decision-feedback approach provides better performance when a long delay is acceptable, while the iterative receiver provides better performance when more stringent delay constraints are imposed.

1 Introduction

Modern error correcting codes such as low-density parity-check (LDPC) codes [1] and turbo codes [2] provide excellent performance over fading channels. In such systems, the receiver must estimate the characteristics of the fading – i.e., the channel state information (CSI) – to effectively decode the data. The optimal approach is to carry out joint channel estimation and decoding; however, the complexity of optimal joint channel estimation and decoding can be prohibitive.

A popular alternative to joint estimation/decoding is to design *iterative receivers* that iteratively estimate the channel and decode the data. An iterative receiver provides a good approximation to the optimal approach with reasonable complexity. A unified approach for designing iterative receivers on factor graphs was proposed by Worthen et. al [5]; this unified approach makes it possible to employ the iterative *sum-product algorithm* on factor graphs [3] describing the iterative receiver. Performance analysis of the resulting receiver is possible by means of density evolution [4]; this also leads to the design of good codes that are well-matched to the receiver. Examples of iterative receivers employing LDPC and turbo codes can be found in [6,7,8]. Although the iterative receiver approach is suboptimal, it achieves good performance.

More recently, a receiver employing decision feedback based successive decoding has been proposed for channels with memory [9]. This approach decomposes

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 235–244, 2006.

a fading channel (or a channel with memory) into a bank of memoryless sub-channels with a block interleaver, and each sub-channel is protected with an LDPC code. The receiver is composed of a decoder and a channel estimator. The LDPC codes are decoded successively, and the decoded symbols are fed back to the channel estimator, which uses the feedback to estimate the channel and then decode the LDPC code for the next sub-channel. It has been shown that this approach is optimal [9]. However, it can incur a long delay in order to achieve optimal performance.

The goal of this paper is to compare the performance of these two approaches. The channel model considered in this paper is a block fading channel with independent channel realizations between blocks. The transmitted signal is subject to frequency-flat, time-selective fading with both amplitude fluctuation and phase rotation. The complex fading coefficient remains constant for T channel uses and then changes to a new (independent) value for the next T channel uses. (In this paper, we refer to each group of T channel uses over which the channel is constant as a *block*.) Neither the transmitter nor the receiver are assumed to know the channel realization. The effect of a delay constraint is examined for each receiver structure.

2 Channel Model

The receiver produces samples of the matched filter output at the symbol rate. The equivalent discrete-time complex channel model is given by

$$y_{i,k} = c_i x_{i,k} + w_{i,k}, \quad i = 1, 2, \ldots N, \quad k = 1, 2, \ldots, T, \tag{1}$$

where the fading coefficients $\{c_i\}$ are i.i.d. complex Gaussian random variables with distribution $\sim \mathcal{CN}(0, 1)$ and the additive noise $\{w_{i,k}\}$ are similarly i.i.d. complex Gaussian with distribution $\mathcal{CN}(0, N_0)$. Here N_0 is the noise variance per two dimensions. In the equation above, $x_{i,k}$ is the kth transmitted symbol of the ith block and $y_{i,k}$ is the corresponding received symbol. For simplicity, we assume binary phase shift keying (BPSK) modulation, so $x_{i,k} \in \mathcal{S} = \{+1, -1\}$.

3 The Successive Decoding Receiver

To transmit data over a block fading channel with coherence time T, the transmitter de-multiplexes the user data into T streams. Each stream is then individually encoded using a block code of length N and rate R_k for $k = 1, \ldots, T$. The kth codeword is denoted $\overline{\mathbf{x}}_k = [x_{1,k}, \ldots, x_{N,k}]^T$ for $k = 1, \ldots, T$. These codewords are stored column-wise in the following block structure:

$$\begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,k} & \cdots & x_{1,T} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,k} & \cdots & x_{2,T} \\ \vdots & \vdots & & \vdots & & \vdots \\ x_{i,1} & x_{i,2} & \cdots & x_{i,k} & \cdots & x_{i,T} \\ \vdots & \vdots & & \vdots & & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,k} & \cdots & x_{N,T} \end{pmatrix}. \tag{2}$$

The transmitter sends the data in 2 row by row. We will also use $\overline{\mathbf{x}}_{i,:}$ to denote the ith row and $\overline{\mathbf{x}}_{:,j}$ to denote the jth columns in (2).

3.1 Estimation and Decoding

The receiver employs a successive decoding strategy that was proposed for channels with memory in [9]. It operates on the block structure in (2), starting from the leftmost column and proceeding to the right. The codeword $\overline{\mathbf{x}}_{:,1}$ is decoded first, and $\overline{\mathbf{x}}_{:,2}$ is decoded second with the assistance of the decoded symbols corresponding to $\overline{\mathbf{x}}_{:,1}$. More specifically, the decoded symbols corresponding to $\overline{\mathbf{x}}_{:,1}$ are used to estimate the channel realizations that affect each symbol in $\overline{\mathbf{x}}_{:,2}$. This approach is used to permit decoding to proceed from left to right.

The estimation and the decoding of a codeword are performed sequentially. Take the kth codeword as an example, where $1 \leq k \leq T$. At this point, all the previous $k-1$ codewords are decoded, and the decoded symbols have been fed back to the receiver. First, the receiver estimates the *a posteriori* probability (APP) of the ith bit in the kth codeword as

$$\mathrm{APP}(x_{i,k} = a) = P\big(x_{i,k} = a | \overline{\mathbf{y}}_{i,:}, x_{i,1}, \cdots, x_{i,k-1}\big) \tag{3}$$

for $a \in \{+1, -1\}$ and $i = 1, \ldots, N$. In (3), the bits $x_{i,1}, \ldots, x_{i,k-1}$ are treated as training symbols. After the receiver calculates the log-likelihood ratios (LLRs) $\{\xi(1,k), \ldots, \xi(N,k)\}$, where

$$\xi(i,k) = \log \frac{\mathrm{APP}(x_{i,k} = +1)}{\mathrm{APP}(x_{i,k} = -1)}, \tag{4}$$

the decoder uses the LLRs to decode the kth LDPC codeword.

3.2 Optimality

The receiver structure described above is optimal, i.e., it is information lossless if the decisions fed back at each stage are correct. This was shown in [9] for any channels with memory. The rest of this section will briefly illustrate the result for the block fading channel.

The main idea is that the block transmission structure effectively decomposes the original block fading channel into a bank of T sub-channels. These sub-channels are memoryless, but they interact with each other via the decision feedback. Thus, the bits in a codeword are transmitted over a memoryless sub-channel, and separate estimation and decoding is optimal. To see this, we write the constrained channel capacity of a block fading channel as

$$C = \frac{1}{TN} I(\overline{\mathbf{x}}_1, \ldots, \overline{\mathbf{x}}_T; \overline{\mathbf{y}}_1, \ldots, \overline{\mathbf{y}}_T) \tag{5}$$

where N is assumed to be sufficiently large. Now define the kth sub-channel as follows: it has a scalar inputs $x_{i,k}$, a vector output $\overline{\mathbf{y}}_{i,:}$ and a vector of decision feedbacks $[x_{i,1}, \ldots, x_{i,k-1}]^T$. The capacity of the kth sub-channel is given by

$$C_k = \frac{1}{N} I\Big(\overline{\mathbf{x}}_k; \overline{\mathbf{y}}_1, \ldots, \overline{\mathbf{y}}_T \Big| [x_{1,1}, \ldots, x_{1,k-1}]^T, \ldots, [x_{T,1}, \ldots, x_{T,k-1}]^T\Big). \tag{6}$$

From the chain rule of mutual information, we have

$$C = \frac{1}{T} \sum_{k=1}^{T} C_k \tag{7}$$

which means the original channel can be decomposed into T sub-channels without loss of mutual information. Furthermore, due to the independent nature of the original channel, the sub-channel is memoryless, i.e.,

$$C_k = \frac{1}{N} \sum_{i=1}^{N} I(x_{i,k}; \overline{\mathbf{y}}_{i,:} | x_{i,1}, \ldots, x_{i,k-1}) \tag{8}$$

Finally, the APP value in (3) is a sufficient statistics for the sub-channel. Therefore, the estimation and decoding scheme in Section 3.1 is optimal.

Intuitively, the block fading channel with coherence time T and i.i.d. inputs can be viewed as a multiple access channel with T independent users and a vector channel output. Using this analogy, the successive interference cancellation scheme, which is optimal for a multiple access channel, becomes a successive decoding scheme, wherein the decision feedback serve as training symbols.

3.3 The APP Calculation

This section describes how the APP in (3) can be computed. Since the techniques for estimating the APP values are the same for any row of (2), we will only consider the first row and drop the time index i for the rest of the paper. In what follows, x_1^k is used to denote the vector $[x_1, x_2, \ldots, x_k]$, and y_1^k is defined similarly. Since

$$P(x_k = a | y_1^T, x_1^{k-1}) \propto P\big(y_1^T | x_1^{k-1}, x_k = a\big), \tag{9}$$

we will consider the computation of likelihood function (9). Minimum mean square error (MMSE) channel estimation uses the decision feedback to obtain an MMSE estimate of the channel state and then enumerates all possible values of the unknown (or future) symbols to obtain the probability 9. Mathematically,

$$\begin{aligned}
P\big(y_1^T | x_1^k\big) &= \sum_{x_{\ +1} \in \mathcal{S}^{\ -}} P(x_{k+1}^T) P\big(y_1^T | x_1^T\big) \\
&= \sum_{x_{\ +1} \in \mathcal{S}^{\ -}} P(x_{k+1}^T) P(y_1^{k-1} | x_1^{k-1}) P(y_k^T | y_1^{k-1}, x_1^T) && (10) \\
&\propto \sum_{x_{\ +1} \in \mathcal{S}^{\ -}} P(y_k^T | y_1^{k-1}, x_1^T) && (11) \\
&= \sum_{x_{\ +1} \in \mathcal{S}^{\ -}} \frac{1}{|\pi \Sigma|} \exp\Big(- \big(y_k^T - x_k^T \hat{c}\big)^H \Sigma^{-1} \big(y_k^T - x_k^T \hat{c}\big)\Big) && (12)
\end{aligned}$$

where from linear estimation theory [10], the conditional mean and variance are given by

$$\hat{c} = \frac{1}{k-1+N_0} \sum_{i=1}^{k-1} y_i x_i^* \quad \text{and} \tag{13}$$

$$\Sigma = \frac{N_0}{k-1+N_0} x_k^T (x_k^T)^H + N_0 \mathbf{I}_{T-k+1}. \tag{14}$$

3.4 Channel Capacity

Using the simplified notation, the constrained capacity of sub-channel k is

$$C_k = I(x_k; y_1^T | x_1^{k-1}). \tag{15}$$

From the definition of mutual information and entropy, (15) becomes

$$C_k = H(x) - \mathrm{E}[-\log \mathrm{APP}(x_k)], \tag{16}$$

where the APP value can be computed using (9) and (12). The expectation in (16) can be evaluated using Monte Carlo integration.

Due to the increasing number of training symbols, the sequence of sub-channel capacity is monotonic increasing, i.e.,

$$C_1 < C_2 < \cdots < C_T. \tag{17}$$

The kth sub-channel is coded by a particular LDPC codes of rate R_k. Here, we set the code rate to be equal to the sub-channel capacity, i.e.,

$$R_k = C_k, \quad \text{for } k = 1, \ldots, T. \tag{18}$$

This paper used irregular LDPC codes optimized for the AWGN channel as component codes.

4 The Iterative Receiver

This section derives an algorithm that carries out iterative channel estimation and LDPC decoding on a joint factor graph. Since the channel is a complex fading channel, pilot symbols are used to assist in estimating the channel states. The basic idea of the iterative receiver is to permit the channel state estimator and the iterative decoder to share preliminary information about the transmitted symbols; after several iterations of LDPC decoding, the decoded symbols are fed back to the channel estimator as additional pilots to help refine the channel estimation.

4.1 Factor Graph

The system factor graph is shown in Figure 1. The LDPC decoder is represented by a bipartite graph in which the variable nodes V represent transmitted symbols and the factor nodes C represent parity checks. One pilot symbol (designated

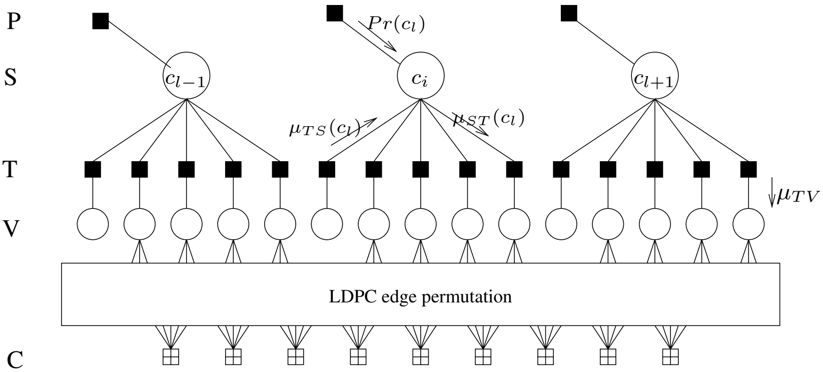

Fig. 1. A fraction of factor graph for a block fading channel with LDPC code, where channel states are complex fading coefficients and independent with each other

x_{l0} for all $l \in \{1, 2, \ldots, N\}$) is transmitted in each fading block; the value of this pilot symbol is known to the receiver and is not part of any LDPC codeword. The joint graph is obtained by concatenating the code graph and the channel graph. In the channel graph, the channel states are denoted by variable nodes S. The factor nodes $T_{lk} = f(y_{lk}|x_{lk}, c_l)$ represent the input-output characteristic of the channel. The factor nodes P represent the prior information about the distribution of the channel state.

In the following, we use notation $\mathcal{CN}(x, m, \sigma^2)$ to represent the complex Gaussian distribution of x with mean m and variance σ^2 per two dimensions. The message from the channel to the decoder and the message from the decoder to the channel are denoted by $\mu_{T\ \to V}\ (y_{lk})$ and $\mu_{V\ \to T}\ (y_{lk})$, respectively; they represent log-likelihood ratios associated with x_{lk}. In contrast, the messages in the channel graph are probability density functions (PDFs). The message from T_{lk} to S_l is $\mu_{T\ \to S}\ (c_l)$, which is the PDF of y_{lk} given the channel state c_l:

$$\mu_{T\ \to S}\ (c_l) = Pr(y_{lk}|c_l, \hat{x}_{lk}) \propto \mathcal{CN}(y_{lk}, c_l\hat{x}_{lk}, \sigma_n^2) \propto \mathcal{CN}(c_l, y_{lk}/\hat{x}_{lk}, \sigma_n^{)}. \quad (19)$$

Here $\hat{x}_{lk}$ represents the decisions made by the channel decoder for the transmitted symbols.

The message from S_l to T_{lk} is the PDF of c_l given the pilots, and is denoted by $\mu_{S\ \to T}\ (c_l)$. Note that, in the first channel estimation iteration, only one pilot symbol is sent for each fading block; however, in the subsequent iterations, additional pilots are obtained by taking hard decisions about the transmitted symbols from the LDPC decoder. Finally the message from P_l to S_l is basically the prior distribution of the channel state c_l, which is $Pr(c_l) = \mathcal{CN}(c_l, 0, 1)$.

4.2 The Iterative Sum-Product Algorithm

Once the messages on the combined factor graph have been defined, it is straightforward to derive the message passing algorithm that iteratively estimates the channel and decodes the LDPC code. Because LDPC decoding via message passing is well understood and widely known, this section will emphasize the aspects of the receiver dealing with channel estimation.

The iterative algorithm works as follows. First, the channel estimator obtains the initial estimate of the channel using the pilot symbols. The LLRs of the channel ($\mu_{T \quad \to V \quad}(y_{lk})$) are then calculated based on this channel estimate and are provided to the LDPC decoder. After several LDPC decoding iterations, new LLRs ($\mu_{V \quad \to T \quad}(y_{lk})$) are calculated by the decoder, and hard decisions ($\hat{x}_{lk}, k \neq 0$) of the transmitted symbols are obtained by the channel estimator based on $\mu_{V \quad \to T \quad}(y_{lk})$. The following hard decision rule is applied,

$$\hat{x}_{lk} = \begin{cases} 1, & \mu_{V \quad \to T \quad} > T \\ -1, & \mu_{V \quad \to T \quad} < -T \\ 0, & \text{otherwise.} \end{cases} \tag{20}$$

If T is sufficiently large, then the code symbols with messages with absolute values that are greater than T are highly reliable, and the resulting non-zero values of $\hat{x}_{lk}$ can be treated as "pseudo-pilots" to help re-estimate the channel.

According to the sum-product rule, the message produced by a state variable node is the product of the input messages, Thus, the message from the channel state node S_l to the factor node T_{lk} is

$$\begin{aligned} \mu_{S \quad \to T \quad}(c_l) &= Pr(c_l) \prod_{j=0, j\neq k, \hat{x} \quad \neq 0}^{j=N-1} \mu_{T \quad \to S}(c_l) \\ &= \mathcal{CN}(c_l, 0, 1) \prod_{j=0, j\neq k, \hat{x} \quad \neq 0}^{j=N-1} \mathcal{CN}(c_l, y_{lk}/\hat{x}_{lk}, \sigma_n^2) \\ &\propto \mathcal{CN}(c_l, \hat{m}_c, \hat{\sigma}_c^2). \end{aligned} \tag{21}$$

The expressions for $\hat{m}_c$ and $\hat{\sigma}_c^2$ can be obtained by applying the product rule for Gaussian PDFs (see Appendix A of [3]) and are omitted here. Also by the sum-product rule, the message produced by a factor node is the product of the input messages with the local factor, marginalized for the destination variable. Thus the message out of the factor node T_{lk} is

$$\begin{aligned} Pr(y_{lk}|x_{lk}) &= \int_c \mu_{S \quad \to T \quad}(c_l) Pr(y_{lk}|c_l, x_{lk}) dc_l \\ &= \int_c \mathcal{CN}(c_l, \hat{m}_c, \hat{\sigma}_c^2) \mathcal{CN}(y_{lk}, x_{lk}c_l, \sigma_n^2) dc_l \\ &\propto \mathcal{CN}(y_{lk}, x_{lk}\hat{m}_c, \hat{\sigma}_c^2 + \sigma_n^2). \end{aligned} \tag{22}$$

Equation (22) comes from the the integration rule for Gaussian PDFs. (See Appendix of [3]). Since the messages from the channel to the decoder are LLRs, we have

$$\mu_{T\;\;\to V}\;\;(y_{lk}) = \log \frac{Pr(y_{lk}|x_{lk} = 1)}{Pr(y_{lk}|x_{lk} = -1)} = \frac{2\mathrm{Re}\{y_{lk}^{*}\hat{m}_c\}}{\hat{\sigma}_c^2 + \sigma_n^2}. \tag{23}$$

5 Simulation Results and Conclusions

We first considered a block fading channel with block length $T = 5$.In this case, the successive decoding receiver uses five codes of rates $R_1, \ldots, R_5$ set to 0, 0.4948, 0.5643, 0.5917, 0.6058, respectively. The overall rate is 0.4513. For the iterative receiver, a code of rate 0.5641 is used, so taking into account the pilots the overall rate is also 0.4513. The channel capacity in terms of $Eb/N0$ is 5.6 dB. To compare the performance of the two receivers, the overall delay is set to be the same. We set the delay to be 200k, 50k, and 6k bits, respectively. For the successive receiver, the codeword length of each sub-channel is 40k, 10k and 1.2k. For the iterative receiver, the codeword length is 160k, 40k, and 4.8k.

The results of the $T = 5$ block fading channel are plotted in Fig. 2. When the delay is large, the successive decoding scheme outperforms the iterative receiver, while at a short delay, the iterative receiver performs better. Intuitively, when a long delay is acceptable, the successive decoding receiver, proven to be optimal by preserving channel mutual information under the assumption that the fed back decoded symbols are correct, will always performs at least as well as the iterative receiver. On the other hand, since the iterative receiver uses a single code, as compared to T codes used in the successive decoding receiver, the block length of the LDPC code in the iterative receiver is T times that of constituent codes in the successive receiver. (Taking into account the one-bit training symbol for each fading block in the iterative receiver, the exact ratio of the component codeword length is $T - 1$.) When the over all delay is relatively short, this difference in codeword length has significant impact on system performance, as clearly demonstrated in the 6k bits delay curve in Fig. 2. In a moderate delay constraint of 50k bits, the performances of the two approaches are rather close.

We also simulated a $T = 10$ block fading channel. In the simulation, the successive receiver uses 10 codes of rates 0, 0.5177, 0.5869, 0.6109, 0.6229, 0.6302, 0.6364, 0.6397, 0.6430 and 0.6453. The iterative receiver uses a code of rate 0.5014. The overall rate of both system is 0.4513. The performance comparison for delay constraints of 10k, 100k and 200k bits results are shown in Fig. 3. The results are similar to the case of $T = 5$. Note that the performance gap between the iterative and successive receiver increases to around 1 dB in the long delay case. This is due to the fact that if delay is fixed, longer channel memory means less channel diversity, which degrades the performance of the iterative receiver.

In conclusion, the decision-feedback based successive receiver has better capacity approaching performance if long delay is acceptable, while iterative receiver is more robust to delay constraints. Currently we are looking at the comparison of two approaches on more practical channel models. We are also investigating the possibility of combining the two design philosophies into one receiver design, that takes the advantages of both approaches.

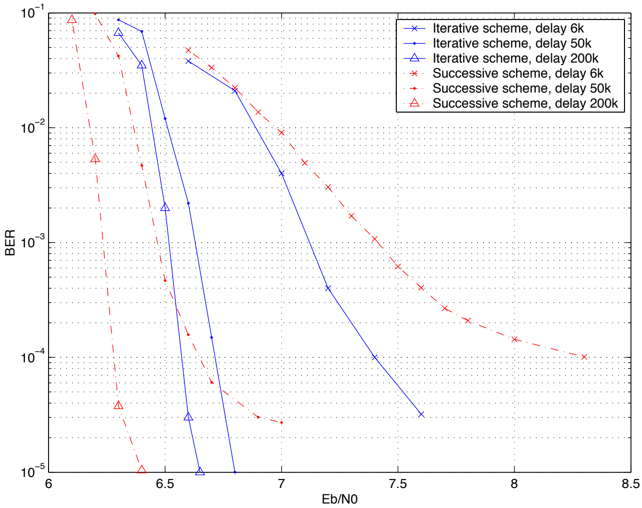

Fig. 2. Performance comparison of successive and iterative schemes for a $T = 5$ independent block fading channel under different delay constraints

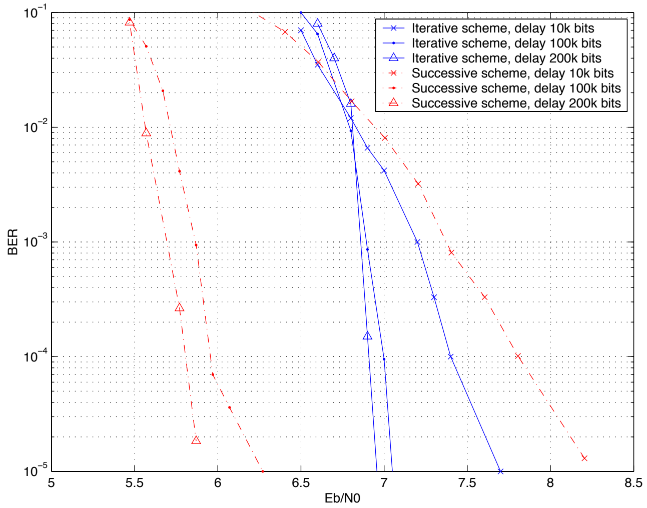

Fig. 3. Performance comparison of successive and iterative schemes for a $T = 10$ independent block fading channel under different delay constraints

References

1. R. G. Gallager: Low Density Parity Check Codes, MIT Press (1963)
2. G. Berrou, A. Gla vieux, P. Thitimajshima: Near shannon limit error-correcting coding: Turbo codes. *Proc. 1993 Int. Conf. Commun.*, pp. 1064-1070, 1993.

3. F. R. Kschischang, B. J. Frey, H.-A. Loeliger: Factor Graphs and the Sum-Product Algorithm. *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498–519, 2001.
4. T. J. Richardson, R. L. Urbanke: The capacity of Low-Density Parity-Check codes under message-passing decoding. *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 599–618 2001
5. A. P. Worthen, W. E. Stark: Unified design of iterative receivers using factor graphs. *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 843–849, 2001.
6. R.-R. Chen, R. Koetter, U. Madhow, D. Agrawal: Joint demodulation and decoding for the noncoherent block fading channel: a practical framework for approaching Shannon capacity. *IEEE Trans. Commun.*, vol. 51, no. 10, pp. 1676-1689, Oct. 2003.
7. K. Fu and A. Anastasopoulos: Analysis and design of LDPC codes for time-selective complex-fading channels. *IEEE Trans. Wireless Communications*, vol. 4, no. 3, pp. 1175-1185, May 2005.
8. X. Jin, A. W. Eckford, T. E. Fuja: Analysis and design of low density parity-check codes for non-coherent block fading channels. *Proc. of the Int. Symp. on Inform. Theory,* Adelaide, Australia, Sep. 2005.
9. T. Li, O. M. Collins: A successive deocding strategy for channels with memory. *Proc. of the Int. Symp. on Inform. Theory*, Adelaide, Australia, Sep. 2005.
10. S. M. Kay: Fundamentals of statistical signal processing estimation theory. Prentice Hall, 1993.

Low-Floor Tanner Codes Via Hamming-Node or RSCC-Node Doping*

Shadi Abu-Surra[1], Gianluigi Liva[2], and William E. Ryan[1]

[1] Electrical and Computer Engineering Department, University of Arizona
ryan@ece.arizona.edu, shadia@ece.arizona.edu
[2] Dipartimento di Elettronica, Informatica e Sistemistica, Universita di Bologna
gliva@deis.unibo.it

Abstract. We study the design of structured Tanner codes with low error-rate floors on the AWGN channel. The design technique involves the "doping" of standard LDPC (proto-)graphs, by which we mean Hamming or recursive systematic convolutional (RSC) code constraints are used together with single-parity-check (SPC) constraints to construct a code's protograph. We show that the doping of a "good" graph with Hamming or RSC codes is a pragmatic approach that frequently results in a code with a good threshold and very low error-rate floor. We focus on low-rate Tanner codes, in part because the design of low-rate, low-floor LDPC codes is particularly difficult. Lastly, we perform a simple complexity analysis of our Tanner codes and examine the performance of lower-complexity, suboptimal Hamming-node decoders.

1 Introduction

An LDPC code, as first proposed by Gallager in [1], is defined as an (n, k) linear block code with a low density of non-zero elements in its parity check matrix H. The $m \times n$ matrix H can be represented as a bipartite graph (Tanner graph) with n variable nodes and m single-parity-check (SPC) nodes. A generalization of these codes was suggested by Tanner in [2], for which subsets of the variable nodes obey a more complex constraint than an SPC constraint, such as a Hamming code constraint. There are at least two advantages to employing constraint nodes with constraints more complex than a simple parity check. First, more complex constraints tend to lead to larger minimum distances. Second, because a complex constraint node can encapsulate multiple SPC constraints, the resulting Tanner graph will contain fewer edges so that deleterious graphical properties are more easily avoided. Both of these advantages lead to a lower error-rate floor. One successful instance of a Tanner code is the turbo product code (TPC) [3]. Another special case of Tanner codes was studied in [4] and [5], where the constraint nodes correspond to Hamming codes. Also, in [6] codes are built by applying BCH or Reed-Solomon code constraints to variable node

* This work was funded by NASA-JPL grant 1264726 and by the University of Bologna, Progetto Pluriennale.

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 245–254, 2006.

subsets, and in [7] recursive systematic convolutional (RSC) codes are used as constraints. The RSC-LDPC codes in this work are more general in the sense that different constraint nodes can be used to construct codes and the graph structure is generally more flexible.

Liva and Ryan in [8], [9] present a more general case of Tanner codes in [5] called Hamming-doped LDPC codes (HD-LDPCC). This generalization allows more than one type of constraint node in the graph as well as irregularity among the node degrees. The doping refers to the fact that the design approach involves inserting Hamming constraint nodes into a Tanner graph or a protograph [10] in place of selected SPC nodes. (A protograph will be defined in Section III.) In this paper, we consider the doping of protographs using either Hamming nodes or RSC nodes; we will call the latter code type RSC-LDPC codes. When referring generically to such a code, we will use doped LPDC code and Tanner code interchangeably. We will also refer to a code that resides at a constraint node as a component code (in contrast with Tanner's "subcode"), and we use constraint node and component-code node interchangeably.

We demonstrate via computer simulations that both HD-LDPC and RSC-LPDC codes exhibit very low error floors, even for code lengths less than 1000 bits. Of course, since our doping technique replaces SPC nodes of code rate $p/(p+1)$ by lower-rate codes, the resulting doped LDPC codes are low-rate codes. Thus, our code design technique provides an approach to designing structured, short (or long), low-rate graphical codes with very low floors, a difficult task if one were restricted to standard LDPC codes [11].

The paper proceeds as follows. In the next section, we present an overview of the construction of Hamming- and RSCC-doped LDPC codes based on protographs. Section III presents four example code family designs. In Section 4, we discuss the iterative decoders which are used to decoder the doped LDPC codes, and analyze their complexity. In Section 5, we present simulation results of the codes we have designed.

2 Overview of the Design Technique

The graph of a Tanner code has n variable nodes and m_c constraint nodes. The connections between the set of variable nodes and constraint nodes V and C is given by an $m_c \times n$ adjacency matrix Γ. For an LDPC code, the adjacency matrix Γ and the parity-check matrix H are identical. For a Tanner code, knowledge of the parity-check matrices of the component codes is also required.

In this paper, we consider only Hamming or RSC component codes in addition to the more common SPC component codes. Because the parity-check matrices for SPC and Hamming codes are straightforward, we discuss only the parity-check matrices for (possibly punctured) rate-1/2 finite-length RSC codes which will be used to dope graphs. For a memory-ν, rate-1/2 RSC code with generator polynomials $g_1(D) = g_{10} + g_{11}D + \cdots + g_{1\nu}D^{\nu}$ and $g_2(D) = g_{20} + g_{21}D + \cdots + g_{2\nu}D^{\nu}$, the corresponding parity-check matrix is

$$H(D) = \left[g_2(D) \; g_1(D) \right] . \tag{1}$$

Because we consider finite block lengths, the binary parity-check matrix for such a code is given by

$$H = \begin{bmatrix} g_{20} & g_{10} & 0 & 0 & 0 & 0 & \cdots \\ g_{20} & g_{10} & g_{20} & g_{10} & 0 & 0 & \cdots \\ \vdots & \vdots & g_{20} & g_{10} & & & \\ g_{2\nu} & g_{1\nu} & \vdots & \vdots & & & \\ 0 & 0 & g_{2\nu} & g_{1\nu} & & & \\ 0 & 0 & 0 & 0 & & \ddots & \\ \vdots & \vdots & \vdots & \vdots & & & \end{bmatrix}, \qquad (2)$$

To find the rate of a doped graph with n variable nodes and m_c constraint nodes, note that each component-code contributes $(1 - R_i)n_i$ redundant bits, where n_i and R_i are the length and rate of the i^{th} component-code, respectively. Consequently, the total number of redundant bits in the code cannot exceed $m = \sum_{i=1}^{m}(1 - R_i)n_i$, and so the number of information bits in the code will be at least $n - m$. This implies that the code rate satisfies $R_c \geq 1 - \frac{m}{n}$, with equality when the check equations are independent.

The parameters in standard LDPC code design which most affect code performance are the degree distributions of the node types, the topology of the graph (e.g., to maximize girth), and the minimum distance, d_{min}. For the design of Tanner codes, decisions must also be made on the types and multiplicities of component codes to be used. The choice of component code types and their multiplicities is dictated by the code rate and complexity requirements. Regarding complexity, we consider only Hamming codes for which the number of parity bits is $(1 - R_i)n_i \leq 4$ and only RSC codes for which the number of trellis states is at most eight. Note that this constraint on the Hamming code family limits the number of states in the time-varying BCJR trellis [12] to be at most 16.

As for LDPC codes, the topology of the graph for a Tanner code should be free of short cycles. Obtaining optimal or near-optimal degree distributions for the graphs of Tanner codes can proceed as for LDPC codes, using EXIT charts [13], for example. In this paper, we instead follow the pragmatic design approach introduced in [8], [9], which starts with a protograph that is known to have a good decoding threshold and replaces selected SPC nodes with either Hamming or RSC nodes. Although we provide no proof, the substitution of these more complex nodes tends to increase minimum distance as shown by simulations. Further, it leads to a smaller adjacency matrix since multiple SPC nodes are replaced by a single component code node. The implication of a smaller adjacency matrix is that short cycles and other deleterious graphical properties are more easily avoided.

3 Example Doped LDPC Code Designs

A protograph [14], [10] is a relatively small bipartite graph from which a larger graph can be obtained by a copy-and-permute procedure: the protograph is copied q times, and then the edges of the individual replicas are permuted among the replicas (under restrictions described below) to obtain a single, large graph. Of course, the edge connections are specified by the adjacency matrix Γ.

Note that the edge permutations cannot be arbitrary. In particular, the nodes of the protograph are labeled so that if variable node A is connected to constraint node B in the protograph, then variable node A in a replica can only connect to one of the q replicated B constraint nodes. Doing so preserves the decoding threshold properties of the protograph. A protograph can possess parallel edges, i.e., two nodes can be connected by more than one edge. The copy-and-permute procedure must eliminate such parallel connections in order to obtain a derived graph appropriate for a parity-check matrix.

It is convenient to choose an adjacency matrix Γ as an $M_c \times n_c$ array of $q \times q$ weight-one circulant matrices (some of which may be the $q \times q$ zero matrix). We will call each row of permutation matrices a *block row* which we observe has q rows and $n = qn_c$ columns. We note that there is one block row for each constraint node of the protograph. We note also that the number of nonzero permutation matrices in a block row is simultaneously equal to the degree of its corresponding constraint nodes and the common length of the nodes' component codes.

Since there is one matrix H_i for each block row of Γ (for the i^{th} component code), we need only discuss the i^{th} block row. Let H_i be $m_i \times n_i$. Then for each row in the i^{th} block row, replace the n_i ones in the row by the corresponding n_i columns of H_i. This expands the i^{th} block row from $q \times n$ to $qm_i \times n$. (For the special case of an SPC constraint node, $m_i = 1$ and the row block is not expanded.) Once this process has been applied to each block row, the resulting parity-check matrix H for the Tanner code will be $\sum_i qm_i \times n$. Because Γ is block circulant, the resulting matrix H can also be put in a block-circulant form (thus, the Tanner code will be quasi-cyclic) [9].

For the case when Γ is not an array of circulants, the H matrix can be obtained via a process analogous to the one above. Γ in this case corresponds to a random permutation on the edges of the protograph replicas, but two constraints are taken in considerations: the protograph structure and the girth of the graph.

In the remainder of this section, we present several HD-LDPC and RSC-LDPC codes whose design relies on doping protographs. In Section 5 we present selected simulation results for these codes on the AWGN channel.

Code 1: Rate-1/6 HD-LDPC Code. The doped protograph for a rate-1/6 HD-LDPC code is shown in Figure 1. The protograph displays a single information bit, u_0, five parity bits p_0 to p_4, two SPC nodes, and a (6,3) shortened Hamming code. The initial protograph that we doped was a rate-1/4 ARA protograph [15], but with minor modification.

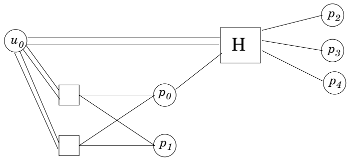

Fig. 1. Rate-1/6 HD-LDPC protograph

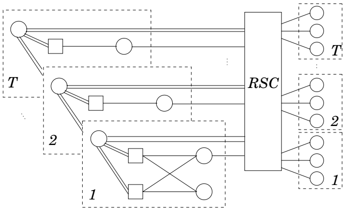

Fig. 2. Rate-1/6 RSC-LDPC protograph

The (6,3) Hamming code was selected because it leads to the targeted rate of 1/6, it has a low-complexity BCJR decoder, and its H matrix-based graph is free of 4-cycles so that belief propagation is an option. Note also that the addition of the Hamming node has the effect of amplifying the minimum distance of the eventual code (after copying and permuting). This is because there will be q copies of the Hamming node whose codewords have a minimum distance of three. Section 5 presents an example code based on this protograph together with its performance (a pseudo-random adjacency matrix is used).

Code 2: Rate-1/6 RSC-LDPC Code. The idea of adding a component code node to amplify weight (hence, d_{min}) led us to consider RSC nodes, particularly since RSC codes produce large weight for low-weight inputs. Since a rate-1/2 RSC code can have any even length, we must consider in the design of an RSC-doped protograph what this length should be. Figure 2 accommodates an unterminated ($6T$, $3T$) RSC component code, where T is a design parameter, so that the overall protograph has T inputs and $6T$ outputs. The $6T$ outputs are represented by all of the circles in Figure 2, some of which are obscured; the RSC node in Figure 2 has $3T$ inputs and $3T$ outputs. Notice that this figure contains T equivalent *sub-protographs*. In the copy-and-permute procedure, we ignore the fact that these were formerly protographs, and apply the copy-and-permute rules only to the overall protograph.

We point out that codes based on this protograph are turbo-like [16] in the sense that copies of the information bits are permuted and distributed over T accumulators, and then part of their outputs together with the remaining information bits are permuted and fed to the RSC code encoder. One major difference, however, is that the present code uses multiple short RSC code blocks rather than one or two long RSC code blocks. The rate-1/6 RSC-LDPC codes presented in Section 5 utilize (pseudo-)random adjacency matrices.

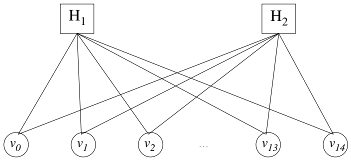

Fig. 3. Rate-1/2 HD-LDPC protograph

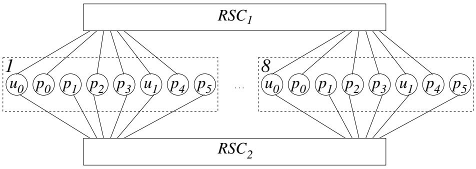
RSC_1
1
u_0 p_0 p_1 p_2 p_3 u_1 p_4 p_5
...
8
u_0 p_0 p_1 p_2 p_3 u_1 p_4 p_5
RSC_2

Fig. 4. Rate-1/4 RSC-LDPC protograph

Code 3: Rate-1/2 HD-LDPC Code. The protograph in Figure 3 corresponds to a rate-1/2 HD-LDPC code. It consists of two (15,11) Hamming component codes and 15 variable nodes. One of the protograph's variable nodes is punctured to achieve the desired rate. Further, the two code-components are not identical. Specifically,

$$H_1 = [M_1 \; M_2] = \begin{bmatrix} 1\,0\,1\,0\,1\,0\,1\,0 & 1\,0\,1\,0\,1\,0\,1 \\ 0\,1\,1\,0\,0\,1\,1\,0 & 0\,1\,1\,0\,0\,1\,1 \\ 0\,0\,0\,1\,1\,1\,1\,0 & 0\,0\,0\,1\,1\,1\,1 \\ 0\,0\,0\,0\,0\,0\,0\,1 & 1\,1\,1\,1\,1\,1\,1 \end{bmatrix}, \tag{3}$$

and $H_2 = [M_2 \; M_1]$, where the definitions of M_1 and M_2 are evident. The benefit of permuting the bits of identical component codes was pointed out by Tanner [2].

A rate-1/2 (2044,1022) Tanner code can be constructed from the protograph of Figure 3 as follows. First, make $q = 146$ total replicas of the protograph. This yield a graph with $n = (15)(146) = 2190$ bit nodes and $m_c = 292$ check nodes. The number of parity bits for the code is $m = 292(15 - 11) = 1168$ so that the resulting code is (2190,1022). For the code presented in Section 5, $\mathit{\Gamma}$ is an array of $q \times q$ circulants, in which case, the code quasi-cyclic. A rate-1/2 (2044,1022) quasi-cyclic Tanner code can be obtained by puncturing the first 146 bits of each codeword (corresponding to the first column of circulants of $\mathit{\Gamma}$).

Code 4: Rate 1/4 RSC-LDPC Code. As depicted in Figure 4, we can obtain a rate-1/4 RSC-LDPC protograph which resembles the protograph of Figure 3, with two different rate-1/2 RSC nodes (of length 48) used in place of the Hamming nodes. Note that the two RSC component-codes form 48 parity check equations, which necessitate the existence of 64 variable nodes in the protograph in order to achieve a rate-1/4 code. Moreover, the number of information bits among these 64 bits is 16 and each 64-bit word must satisfy these 48 check equations. In Figure 4, we divided the variable nodes into eight similar groups (enclosed in the dash boxes), with six connections to each RSC code. Each group contains two information bits, u_0 and u_1, and six parity bits, p_0 to p_5, which are ordered in a sequence relevant to the decoder.

The rate-1/2 RSC component codes have two different polynomial sets; one has polynomials $(17, 15)_8$ and the other has polynomials $(3, 2)_8$. Assuming that both have unterminated trellises; the resultant code has rate 1/4. However, we have to terminate one of the two component codes to obtain good performance. (Terminating both of them also works, but at the cost of code rate.) In this code, the $(17, 15)_8$ RSC code trellis has been terminated. Since $\nu = 3$ and the rate is 1/2 for this component code, 6 code bits are related to these termination bits.

From Figure 4, the last six bits of each of the RSC component codes include two information bits. Consequently, trellis termination process reduces the rate from 16/64 to 14/64. In order to obtain rate 1/4, we puncture eight of the 64 bits, four degree-one variable nodes from each RSC code.

Finally, we constructed a (16352,4088) RSC-LDPC code by making 292 copies of the above protograph. A block-circulant adjacency matrix was used in our simulations. In summary, $n = 18\,688$, $m_c = 584$, $M_c = 2$, and $q = 292$.

4 Doped-LDPC Code Iterative Decoder

For LDPC codes in this paper, we used the standard sum-product algorithm (SPA). For the Tanner codes which have more complex constraint nodes, a soft-input soft-output (SISO) decoder is used to compute the soft-output messages. The choice of the SISO decoder for non-SPC constraint codes depends on the code type. For RSC codes we use the BCJR decoder [17].

In HD-LDPC codes, the Hamming constraints can be replaced by their equivalent SPC equations. However, except for the (6,3) shortened Hamming code, the large number of 4-cycles the resultant graph degrades the performance of the SPA decoder. Alternatively, for the Hamming nodes, we can use the BCJR decoder applied to the BCJR trellis [12]. We also consider the modified Chase algorithm [18] and the cyclic-2 pseudo-maximum likelihood (PML) decoder [19].

The modified Chase and cyclic-2 PML decoders are both SISO list-based decoders. Cyclic-2 has an advantage over modified Chase in term of complexity as it uses a list that refers to nearby codewords, which are independent of its input, resulting in fewer addition operations. The complexity reduction factor from using either of these decoders instead of the BCJR decoder depends on the number of the states in the code's trellis. As an example, in the decoding of 10^7 codewords of the (32, 26) extended Hamming code, we observed that the cyclic-2 decoder was 9 times faster than BCJR decoder, and the modified Chase decoder was 4.5 times faster than BCJR.

Lastly, to gain insight on the decoding complexity of the HD-LDPC and RSC-LDPC codes compared with that of standard regular LDPC we consider the following rate 1/6 codes. The first is an HD-LDPC code constructed from W copies of the protograph in Figure 1. The second is an RSC-LDPC code based on one copy of the protograph in Figure 2, using the RSC code polynomials $(5, 7)_8$. The last code is an LDPC code derived from the previous HD-LDPC code, where the Hamming constraint is replaced by its SPC constraints.

The number of additions per iteration are $131W$, $50W$, and $20W$ for HD-LDPC, RSC-LDPC, and LDPC codes, respectively. This calculation is based on the following (η is the relevant block length, $N_{s,total}$ is the total number of trellis states in the finite-length trellis): (1) For a standard LDPC codes, the number of additions equals to the number of ones in its parity-check matrix. (2) The number of additions in the HD-LDPC BCJR is given by $2N_{s,total}$ $+4\eta$. (3) For the RSC-LDPC BCJR, the number of additions is $2N_{s,total}$ $+5\eta/2$ because the number of stages in a rate-1/2 RSC trellis is half the block length, but it has to compute two values at each stage; hence, 5 instead of 4.

5 Simulation Results

In this section we present several simulation results for different doped-LDPC codes. First, we designed a (600, 100) HD-LDPC code based on the protograph in Figure 1 and three (600, 100) RSC-LDPC codes based on the protograph in Figure 2. The three RSC-LDPC codes correspond to three different values of the parameter T: $T = 2$, 4, and 8. All of these codes were constructed using random permutations on the edges of their protographs, but two constraints are taken into consideration: the protograph structure and the girth of the graph. The progressive edge growth construction in [20] is used to give the required girth, which is eight for all loops that have only SPC nodes. On the other hand, loops that include Hamming or RSC nodes can be of length less than eight.

A comparison between the frame error rate (FER) curves of these codes and the (600,100) random coding bound (RCB) is presented in Figure 5. The iterative decoder described above was used, where BCJR decoders are used to decode the Hamming and RSC component codes. The maximum number of iterations is $I_{max} = 50$ and 20 error events were collected at each E_b/N_0 value on each curve, except for the point at 4.5 dB of the $T = 8$ RSC-LDPC curve where only three error events occurred during the decoding of 7.26×10^8 codewords. Note that the floor for almost every code is quite low, even though the code length is 600. Note also the lowest floor occurs for the $T = 8$ RSC-LDPC code, which shows no evidence of a floor down to FER $\approx 10^{-9}$. This code is about 1.3 dB from the random coding bound at FER=10^{-4}.

Figure 6 shows the error rate performance curves of the (2044, 1022) quasi-cyclic HD-LDPC code. The Hamming component codes were decoded using the BCJR decoder and the overall decoder employed a maximum of $I_{max} = 50$ iterations. The code performance is within 1 dB of the random coding bound and has no floor down to FER $\approx 5 \times 10^{-8}$.

The performance of the rate-1/4 RSC-LDPC code ($I_{max} = 20$) constructed in Section 2 is presented in Figure 7. Its performance is compared to that of

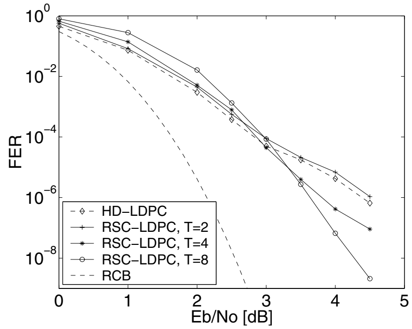

Fig. 5. Frame error rate comparison between (600, 100) HD-LDPC code and RSC-LDPC codes at different T, $I_{max} = 50$

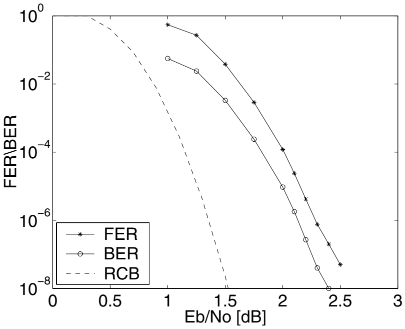

Fig. 6. Performance of (2044, 1022) HD-LDPC code compared to the random coding bound. $I_{max} = 50$.

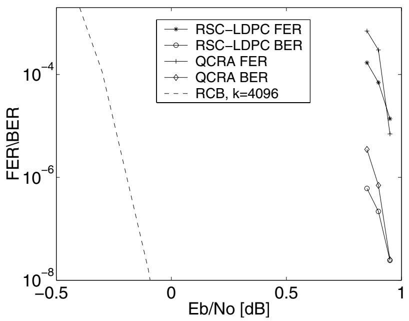

Fig. 7. Performance of (16352, 4088) RSC-LDPC code compared to that of (16384, 4096) QCRA code. $I_{max} = 20$.

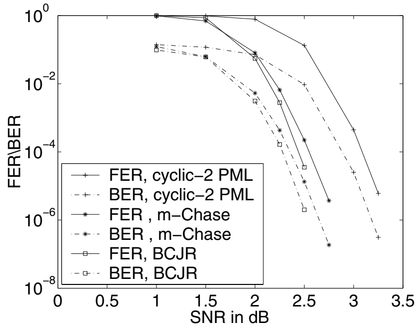

Fig. 8. Comparison between the performance of BCJR decoder, and the other sub-optimal decoders. The (2048, 1024) HD-LDPC components are (32, 26) extended Hamming codes.

the quasi-cyclic repeat-accumulate code (QCRA) in [21] as well as the random coding bound. The curves show that our code is superior to the QCRA code at low E_b/N_0 values. But at higher E_b/N_0 values, the QCRA code has a slightly better FER than the RSC-LDPC. We noticed that by increasing I_{max} from 20 to 50 in RSC-LDPC code, the FER at $E_b/N_0 = 0.8$ dB reduced to around 2×10^{-6}.

Finally, we examined the performance of a (2048, 1024) HD-LDPC code, constructed from the (32, 26) extended Hamming code, using the BCJR decoder, the Chase decoder (radius 6), and the cyclic-2 PML decoder. Note in Figure 8 that the performance curves of the modified Chase and the BCJR decoders are almost the same, and about 0.5 dB better than that of the cyclic-2 PML decoder. On the other hand, cyclic-2 PML decoder is about twice as fast as the Chase decoder and about nine times as fast as the BCJR decoder.

Acknowledgments

We would like to acknowledge Professor Keith Chugg of the University of Southern California and Professor Marc Fossorier of the University of Hawaii for helpful discussions on finite code length performance bounds, and Prof. Chugg for his program which computes such bounds.

References

1. R. G. Gallager, “Low-density parity-check codes,” *IRE Transactions on Information Theory*, vol. 8, pp. 21–28, January 1962.
2. R. M. Tanner, “A recursive approach to low complexity codes,” *IEEE Transactions on Information Theory*, vol. 27, pp. 533–547, September 1981.
3. P. Elias, “Error free coding,” *IRE Transactions on Information Theory*, vol. PGIT-4, pp. 29–37, September 1954.

4. M. Lentmaier and K. S. Zigangirov, “Iterative decoding of generalized low-density parity-check codes,” in *IEEE International Symposium on Information Theory*, p. 149, August 1998.
5. J. Boutros, O. Pothier, and G. Zemor, “Generalized low density (Tanner) codes,” in *IEEE International Conference on Communications, ICC ’99*, pp. 441–445, June 1999.
6. N. Miladinovic and M. Fossorier, “Generalized LDPC codes with Reed-Solomon and BCH codes as component codes for binary channels,” in *IEEE Global Telecommunications Conference, GLOBECOM ’05*, November 2005.
7. S. Vialle and J. Boutros, “A Gallager-Tanner construction based on convolutional codes,” in *Proceedings of International Workshop on Coding and Cryptography, WCC’99*, pp. 393–404, January 1999.
8. G. Liva and W. E. Ryan, “Short low-error-floor Tanner codes with Hamming nodes,” in *IEEE Military Communications Conference, MILCOM ’05*, 2005.
9. G. Liva, W. E. Ryan, and M. Chiani, “Design of quasi-cyclic Tanner codes with low error floors,” in *4th International Symposium on Turbo Codes, ISTC-2006*, April 2006.
10. J. Thorpe, “Low-density parity-check (LDPC) codes constructed from protographs,” Tech. Rep. 42-154, IPN Progress Report, August 2003.
11. Y. Mao and A. H. Banihashemi, “A heuristic search for good low-density parity-check codes at short block lengths,” in *IEEE International Conference on Communications, ICC 2001*, pp. 41–44, June.
12. R. J. McEliece, “On the BCJR trellis for linear block codes,” *IEEE Transactions on Information Theory*, vol. 42, pp. 1072–1092, July 1996.
13. S. ten Brink, G. Kramer, and A. Ashikhmin, “Design of low-density parity-check codes for modulation and detection,” *IEEE Transactions on Communications*, vol. 52, pp. 670–678, April 2004.
14. I. D. S. Lin, J. Xu and H. Tang, “Hybrid construction of LDPC codes,” in *Proc. of the 40th Annual Allerton Conference on Communication, Control, and Computing, Illinois*, October 2002.
15. A. Abbasfar, D. Divsalar, and K. Yao, “Accumulate repeat accumulate codes,” in *IEEE Global Telecommunications Conference, GLOBECOM ’04*, pp. 509–513, November 2004.
16. D. Divsalar, H. Jin, and R. McEliece, “Coding theorems for “turbo-like” codes,” in *Proc. of 36th Allerton Conf.*, September 1998.
17. L. R. Bahl, J. cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Transactions on Information Theory*, vol. 20, pp. 284–287, March 1974.
18. R. M. Pyndiah, “Near-optimum decoding of product codes: Block turbo codes,” *IEEE Transactions on Communications*, vol. 46, pp. 1003–1010, August 1998.
19. “Block decoding with soft output information,” Patent 5930272, United States Patent, July 1999.
20. X. Y. Hu, E. Eleftheriou, and D. M. Arnold, “Progressive edge-growth Tanner graphs,” in *IEEE Global Telecommunications Conference, GLOBECOM ’01*, pp. 995–1001, November 2001.
21. R. M. Tanner, “On quasi-cyclic repeat-accumulate codes,” in *Proc. of the 37th Annual Allerton Conference on Communication, Control, and Computing, Monticello, Illinois*, September 1999.

Algebraic Constructions of Quasi-cyclic LDPC Codes – Part I: For AWGN and Binary Random Erasure Channels*

Lan Lan, Lingqi Zeng, Ying Y. Tai, Lei Chen, Shu Lin,
and Khaled Abdel-Ghaffar

Department of Electrical and Computer Engineering,
University of California, Davis, CA 95616
{squash, lqzeng, yytai, leichen, shulin,
ghaffar}@ece.ucdavis.edu

Abstract. This paper is the first part of a sequence of two papers that present algebraic constructions of quasi-cyclic LDPC codes for AWGN, binary random and burst erasure channels. In this paper, a class of quasi-cyclic LDPC codes for both AWGN and binary random erasure channels is constructed based on finite fields and special vector representations of finite field elements.

1 Introduction

LDPC codes, discovered by Gallager in 1962 [1], were rediscovered and shown to form a class of Shannon capacity approaching codes in the late 1990's [2, 3]. Ever since their rediscovery, design, construction, decoding, efficient encoding, and applications of these codes in digital communication and storage systems have become focal points of research. Many methods for constructing these codes have been proposed. Based on the methods of construction, LDPC codes can be classified into two general categories: (1) random-like codes [4, 5] that are generated by computer search based on certain design guidelines and required structural properties of their Tanner graphs [6]; and (2) structured codes that are constructed based on algebraic and combinatorial tools [7, 8, 9, 10, 11, 12, 13, 14, 15, 16].

Most of the proposed constructions of LDPC codes are for the AWGN channel, however only a few of them for other types of channels. In this and a succeeding papers, we present two algebraic methods for constructing quasi-cyclic (QC) LDPC codes for AWGN, binary random and burst erasure channels. QC-LDPC codes have encoding advantage over the other types of LDPC codes. They can be encoded with simple shift-registers with linear complexity [17]. It has been shown that well designed QC-LDPC codes decoded with iterative decoding perform very well over the AWGN channel and close to the Shannon theoretical limit [10, 14]. In this and next papers, we show that well designed QC-LDPC codes decoded with iterative decoding also perform well over binary random and burst erasure channels.

* This research was supported by NASA under the Grant NNG05GD13G and NSF under the Grant CCR-0117891 and ECS-0121469.

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 255–264, 2006.

A binary regular LDPC code [1] is defined as the null space of a sparse parity-check matrix **H** over GF(2) with the following structural properties: (1) each row has constant weight ρ; (2) each column has constant weight γ; (3) no two rows (or two columns) have more than one 1-component in common; and (4) both ρ and γ are small compared with the code length. **H** is said to be (γ,ρ)-*regular* and the code given by the null space of **H** is called a (γ,ρ)-regular LDPC code. Property (3) is referred to as the *column-row* (*RC*) *constraint*. The RC-constraint ensures that: (1) the minimum distance of the code is at least $\gamma+1$; and (2) the Tanner graph of the code is *free* of cycles of length 4 [7]. An LDPC code is said to be *irregular* if its parity-check matrix has *varying column weights and/or varying row weights*. A QC-LDPC code is given by the null space of an *array of sparse circulants* [7, 10, 14].

The performance of an LDPC code decoded with iterative decoding is measured by its bit-error probability, block-error probability, error-floor and rate of decoding convergence, collectively. Structured LDPC codes in general have a lower error-floor which is important in digital communication and storage systems, where very low error rates are required. Structured LDPC codes with large minimum distances can be constructed much easier than computer generated random-like LDPC codes.

The performance of an LDPC code over the AWGN channel with iterative decoding depends on a number of code structural properties besides its minimum distance. One such structural property is the *girth* of the code that is defined as the length of the shortest cycle in the code's Tanner graph. For an LDPC code to perform well over the AWGN channel with iterative decoding, its Tanner graph must not contain short cycles. The shortest cycles that affect code performance the most are cycles of length 4. Therefore, cycles of length 4 must be prevented in LDPC code construction for the AWGN channel. For an LDPC code to perform well over the binary random erasure channel, its Tanner graph must also be free of cycles of length 4 [18, 19].

2 LDPC Codes for the Binary Random Erasure Channel

For transmission over the binary random erasure channel, a symbol, 0 or 1, is either correctly received with probability $1-p$ or erased with probability p (called *erasure probability*), and there is no transmission error. The output of the binary random erasure channel consists of three symbols, 0, 1, and ?, where the symbol "?" denotes a transmitted symbol being erased, called an *erasure*. Suppose a codeword $\boldsymbol{x} = (x_0, x_1, \ldots, x_{n-1})$ from a binary code $\mathcal{C}$ of length n is transmitted and $\boldsymbol{y} = (y_0, y_1, \ldots, y_{n-1})$ is the corresponding received sequence. Let $\mathcal{E} = \{j_1, j_2, \ldots, j_t\}$ be the set of locations in $\boldsymbol{y}$ with $0 \leq j_1 < j_2 < \ldots < j_t < n$, where the transmitted symbols are being erased. Let $[n] \triangleq \{0, 1, \ldots, n-1\}$. Define $\bar{\mathcal{E}} \triangleq [n] \backslash \mathcal{E}$. Then $\bar{\mathcal{E}}$ is the set of locations in **y** where the transmitted symbols are correctly received, i.e., $y_i = x_i$ for $i \in \bar{\mathcal{E}}$. The set $\mathcal{E}$ displays the *pattern* of erased symbols in **y** and is called an *erasure pattern*. Decoding $\boldsymbol{y}$ is to determine the value of each erasure in $\mathcal{E}$. An erasure pattern $\mathcal{E}$ is said to be *recoverable* (*resolvable* or *correctable*) if the value of each erasure in $\mathcal{E}$ can be uniquely determined.

Consider an LDPC code $\mathcal{C}$ of length n given by the null space of a $J \times n$ sparse matrix **H**. Then a binary n-tuple $\boldsymbol{x} = (x_0, x_1, \ldots, x_{n-1})$ is a codeword in $\mathcal{C}$ if and only if $\boldsymbol{x}\mathbf{H}^T = \mathbf{0}$. Suppose a codeword $\boldsymbol{x}$ is transmitted and $\boldsymbol{y} = (y_0, y_1, \ldots, y_{n-1})$ is the corresponding received sequence. Let $\mathcal{E} = \{j_1, j_2, \ldots, j_t\}$ be the erasure pattern contained in $\boldsymbol{y}$. Let $\mathbf{H}_\varepsilon$ be the submatrix that consists of the columns of $\mathbf{H}$ corresponding to the locations of the erased symbols given in $\mathcal{E}$ and $\mathbf{H}_{\bar{\varepsilon}}$ be the submatrix that consists of the columns of $\mathbf{H}$ corresponding to the locations of the correctly received code symbols in $\bar{\mathcal{E}}$. Let $\boldsymbol{y}_\varepsilon$ denote the subsequence that consists of the erased symbols in $\boldsymbol{y}$ and $\boldsymbol{y}_{\bar{\varepsilon}}$ denote the subsequence that consists of the known symbols in $\boldsymbol{y}$ at the locations given in $\bar{\mathcal{E}}$. The symbols in $\boldsymbol{y}_\varepsilon$ are unknown. For $\boldsymbol{y}$ to be a codeword in $\mathcal{C}$, we must have $\boldsymbol{y}\mathbf{H}^T = \mathbf{0}$. This constraint can be put in the form:

$$\boldsymbol{y}_\varepsilon \mathbf{H}_\varepsilon^T = \boldsymbol{y}_{\bar{\varepsilon}} \mathbf{H}_{\bar{\varepsilon}}^T. \tag{1}$$

The right-hand side of (1) is known and can be computed from $\boldsymbol{y}_{\bar{\varepsilon}}$ and $\mathbf{H}_{\bar{\varepsilon}}$. The left-hand side of this equation (or $\boldsymbol{y}_{\bar{\varepsilon}}$) is unknown. Then decoding $\boldsymbol{y}$ is to solve (1). An iterative method for solving (1) was proposed in [18].

Let $\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_J$ be the rows of $\mathbf{H}$. For $1 \leq i \leq J$, let $\boldsymbol{h}_i = (h_{i,0}, h_{i,1}, \ldots, h_{i,n-1})$. Then a codeword $\boldsymbol{y} = (y_0, y_1, \ldots, y_{n-1})$ must satisfy the condition, $s_i \triangleq y_0 h_{i,0} + y_1 h_{i,1} + \ldots + y_{n-1} h_{i,n-1} = 0$ for $1 \leq i \leq J$, which is called a *check-sum*. The code symbol y_j is said to be checked by the sum s_i if $h_{i,j} = 1$, i.e., y_j is included in the sum s_i. Then y_j can be determined from other code bits that are checked by $\boldsymbol{h}_i$ as follows:

$$y_j = \sum_{k=0, k \neq j}^{n-1} y_k h_{i,k}. \tag{2}$$

For each erased position j_l in an erasure pattern $\mathcal{E} = \{j_1, j_2, \ldots, j_t\}$ with $1 \leq l \leq t$, if there exists a row $\boldsymbol{h}_i$ in $\mathbf{H}$ that checks *only* the erased symbol y_j and not any of the other $t-1$ erased symbols in $\mathcal{E}$, then it follows from (2) that the value of each erased symbol in $\mathcal{E}$ can be determined by the correctly received symbols in $\bar{\mathcal{E}}$ as follows:

$$y_j = \sum_{k \in \bar{\mathcal{E}}} y_k h_{i,k}. \tag{3}$$

Such an erasure pattern is said to be *resolvable* in *one step* (or *one iteration*). However, there are erasure patterns that are not resolvable in one step but resolvable in *multiple steps* iteratively. Given an erasure pattern $\mathcal{E}$, we first determine the values of those erased symbols that can be resolved in one step using (3). Then we remove the known erased symbols from $\mathcal{E}$. This results in a new erasure pattern $\mathcal{E}_1$ of smaller size. Next we determine the values of erased symbols in $\mathcal{E}_1$ that are resolvable using (3). Removing the known erased symbols from $\mathcal{E}_1$, we obtain an erasure pattern $\mathcal{E}_2$ of size smaller than that of $\mathcal{E}_1$. We repeat the above process iteratively until either all the erased symbols in $\mathcal{E}$ are resolved or an erasure pattern $\mathcal{E}_m$ is obtained such that no erasure in $\mathcal{E}_m$ can be resolved using (3). In the latter case, some erasures can not be recovered.

The above decoding process is iterative in nature and can be formulated as an algorithm [18]. To initialize the decoding process, we first set $k = 0$ and $\mathcal{E}_0 = \mathcal{E}$. Then we execute the following steps iteratively:

(1) Determine $\mathcal{E}_k$. If $\mathcal{E}_k$ is empty, stop decoding, otherwise go to Step 2.
(2) Form $\mathbf{H}_\varepsilon$, $\mathbf{H}_{\bar{\varepsilon}}$, $\boldsymbol{y}_\varepsilon$, and $\boldsymbol{y}_{\bar{\varepsilon}}$.

(3) Compute $\boldsymbol{y}_{\bar{\varepsilon}}\ \mathbf{H}^T_{\bar{\varepsilon}}$.

(4) Find the rows in $\mathbf{H}_\varepsilon$ such that each contains only one 1-component. Determine the erasures in $\mathcal{E}_k$ that are checked by these rows. Determine the values of these erasures by application of (3) and go to Step 5. If there is no row in $\mathbf{H}_\varepsilon$ that contains only one 1-component, stop decoding.

(5) Remove the erasures resolved at the step 4 from $\mathcal{E}_k$. Set $k = k+1$ and go to Step 1.

If decoding stops at Step 1, all the erasures in the erasure pattern $\mathcal{E}$ are resolved and the decoding is successful. If decoding stops at Step 4, some erasures in $\mathcal{E}$ can not be recovered.

The performance measure of an LDPC code over the binary random erasure channel is the error probability. Di et. al. [18] have derived the *threshold* for regular LDPC codes with given Tanner degree distribution pair (γ, ρ) (or column and row weight pair of a (γ,ρ)-regular parity-check matrix) using the above iterative decoding algorithm. The threshold is a small probability $\varepsilon(\gamma, \rho)$ associated with an ensembles of regular LDPC codes whose Tanner graphs have degree distribution pair (γ,ρ). The implication of threshold $\varepsilon(\gamma, \rho)$ is as follows: over all binary random erasure channels with erasure probability p smaller than $\varepsilon(\gamma, \rho)$, information can be reliably transmitted by using a sufficiently long LDPC code with degree distribution pair (γ, ρ). Reliable transmission of information is not possible if the erasure probability p is larger than the threshold $\varepsilon(\gamma, \rho)$.

The performance of an LDPC code over the binary random erasure channel is determined by the *stopping sets* of its Tanner graph $\mathcal{T}$ [18]. Let $\mathcal{V}$ be a set of variable nodes in $\mathcal{T}$ and $\mathcal{S}$ be the set of check nodes in $\mathcal{T}$ such that each check node in $\mathcal{S}$ is connected to at least one variable node in $\mathcal{V}$. The nodes in $\mathcal{S}$ are called the neighbors of the nodes in $\mathcal{V}$. A set $\mathcal{V}$ of variable nodes is called a stopping set of $\mathcal{T}$ if each check node in the neighbor check set $\mathcal{S}$ of $\mathcal{V}$ is connected to at least two nodes in $\mathcal{V}$. If an erasure pattern $\mathcal{E}$ corresponds to a stopping set in the Tanner graph of an LDPC code, then a checksum that checks an erasure in $\mathcal{E}$ also checks *at least one other erasure* in $\mathcal{E}$. As a result, no erasure in $\mathcal{E}$ can be determined with Eq. (3) (or Eq. (1)) and $\mathcal{E}$ is an irrecoverable erasure pattern.

A set $\mathcal{Q}$ of variable nodes in $\mathcal{T}$ may contain many stopping sets. It is clear that: (1) the union of two stopping sets in $\mathcal{Q}$ is also a stopping set in $\mathcal{Q}$; and (2) the union of all the stopping sets in $\mathcal{Q}$ gives the maximum stopping set in $\mathcal{Q}$. A set $\mathcal{V}_{ssf}$ of variable nodes in $\mathcal{T}$ is said to be *stopping-set-free* (SSF) if it does not contain any stopping set. The following theorem [18] characterizes the significance of stopping sets for correcting erasures: Suppose an LDPC code $\mathcal{C}$ is used for correcting erasures using iterative decoding. Let $\mathbf{y}$ be a received sequence that contains an erasure pattern $\mathcal{E}$. Then the erasures contained in the maximum stopping set of $\mathcal{E}$ cannot be recovered. This theorem says that any erasure pattern $\mathcal{E}$ is recoverable if it is SSF.

Let $\mathcal{B}$ be a stopping set of minimum size in the Tanner graph of an LDPC code, called a *minimal stopping set* (not unique). If the code symbols corresponding to the variable nodes in $\mathcal{B}$ are being erased, it follows from the above theorem that $\mathcal{B}$ forms an irrecoverable erasure pattern of minimum size. Therefore, for random erasure correction with iterative decoding, it is desired to construct codes with largest possible minimal stopping sets in their Tanner graphs. A good LDPC code for erasure correction must have no or very few small stopping sets. A stopping set always contains cycles. In [19], it has been proved that the size of a minimal stopping set of a Tanner graph with girth 4 is two. The size of a minimal stopping set of a Tanner graph with girth 6 is $\gamma + 1$ and is 2γ for girth 8, where γ is the degree of a variable node (or the column weight of the parity-check matrix of the code). Hence for iterative decoding of an LDPC code over the binary random erasure channel, the most critical cycles in the code's Tanner graph are cycles of length 4. Therefore, in code construction for the binary random erasure channel, cycles of length 4 must be avoided in the Tanner graph of a code. It is proved in [20] that for a code with minimum distance d_{min}, it must contain a stopping set of size d_{min}. Therefore, in the construction of a code for erasure correction, we need to keep its minimum distance large. For a regular LDPC code whose parity-check matrix has column weight γ and satisfies the RC-constraint, the size of a minimal stopping set in the code's Tanner graph is at least $\gamma + 1$.

3 A Class of QC-LDPC Codes Constructed Based on Finite Fields

Consider the Galois field GF(q) where q is a power of a prime. Let α be a primitive element of GF(q). Then $\alpha^{-\infty} \triangleq 0, \alpha^0 = 1, \alpha, \ldots, \alpha^{q-2}$ form all the elements of GF(q) and $\alpha^{q-1} = 1$. The $q-1$ nonzero elements of GF(q) form the multiplicative group of GF(q) under the multiplication operation. For each nonzero element α^i with $0 \leq i \leq q-2$, we form a $(q-1)$-tuple over GF(2), $\mathbf{z}(\alpha^i) = (z_0, z_1, \ldots, z_{q-2})$, whose components correspond to the $q-1$ nonzero elements of GF(q), where the ith component $z_i = 1$ and all the other $q-2$ components are equal to 0. This $(q-1)$-tuple $\mathbf{z}(\alpha^i)$ is referred to as the *location vector* of α^i with respective to the multiplicative group of GF(q). We call $\mathbf{z}(\alpha^i)$ the location-vector of α^i. The location-vectors of two different nonzero elements of GF(q) are different. The location vector of the 0 element of GF(q) is defined as the all-zero $(q-1)$-tuple, $(0, 0, \ldots, 0)$. Let β be a nonzero element in GF(q), then the location-vector $\mathbf{z}(\alpha\beta)$ of $\alpha\beta$ is the *cyclic-shift* (*one place to the right*) of the location-vector $\mathbf{z}(\beta)$ of β. Form a $(q-1) \times (q-1)$ matrix $\mathbf{A}$ over GF(2) with the location-vectors of $\beta, \alpha\beta, \ldots, \alpha^{q-2}\beta$ as rows. Then $\mathbf{A}$ is a *circulant permutation matrix*.

Form the following $(q-1) \times (q-1)$ matrix over GF(q):

$$\mathbf{M} = \begin{bmatrix} \mathbf{w}_0 \\ \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_{q-2} \end{bmatrix} = \begin{bmatrix} \alpha^0 - 1 & \alpha - 1 & \cdots & \alpha^{q-2} - 1 \\ \alpha - 1 & \alpha^2 - 1 & \cdots & \alpha^{q-1} - 1 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{q-2} - 1 & \alpha^{q-1} - 1 & \cdots & \alpha^{2(q-2)} - 1 \end{bmatrix}. \quad (4)$$

Matrix $\mathbf{M}$ has the following structural properties: (1) any two rows (or two columns) differ in all positions; (2) all the entries in a row (or a column) are different elements in GF(q); and (3) each row (or column) contains one and only one zero element.

Lemma 1. *For* $0 <= i, j, k, l < q-1$ *with* $i \neq j$, *the two* $(q-1)$*-tuples* $\alpha^k \mathbf{w}_i$ *and* $\alpha^l \mathbf{w}_j$ *can not have more than one position with identical components, i.e., they differ in at least* $q-2$ *positions.*

Proof. Suppose there are two different positions, say s and t with $0 \leq s, t < q-1$, where $\alpha^k \mathbf{w}_i$ and $\alpha^l \mathbf{w}_j$ have identical components. Then $\alpha^k(\alpha^{i+s} - 1) = \alpha^l(\alpha^{j+s} - 1)$

and $\alpha^k(\alpha^{i+t}-1)=\alpha^l(\alpha^{j+t}-1)$. These two equalities imply that either $i=j$ or $s=t$ that contradicts the assumptions that $i\neq j$ and $s\neq t$. This proves the theorem.

For each row $\mathbf{w}_i$ of $\mathbf{M}$ given by (4) with $0\leq i<q-1$, we form the following $(q-1)\times(q-1)$ matrix over GF(q) with $\mathbf{w}_i,\alpha\mathbf{w}_i,\ldots,\alpha^{q-2}\mathbf{w}_i$ as rows:

$$\mathbf{M}_i=\begin{bmatrix}\mathbf{w}_i\\ \alpha\mathbf{w}_i\\ \vdots\\ \alpha^{q-2}\mathbf{w}_i\end{bmatrix}=\begin{bmatrix}\alpha^i-1 & \alpha^{i+1}-1 & \cdots & \alpha^{i+q-2}-1\\ \alpha(\alpha^i-1) & \alpha(\alpha^{i+1}-1) & \cdots & \alpha(\alpha^{i+q-2}-1)\\ \vdots & \vdots & \ddots & \vdots\\ \alpha^{q-2}(\alpha^i-1) & \alpha^{q-2}(\alpha^{i+1}-1) & \cdots & \alpha^{q-2}(\alpha^{i+q-2}-1)\end{bmatrix}. \tag{5}$$

We label the column of $\mathbf{M}_i$ from 0 to $q-2$. We readily see that: (1) any two rows differ in every position, except the $(q-1-i)$th position, where they both have the 0 element of GF(q); and (2) the $q-1$ entries of each column of $\mathbf{M}_i$ form the $q-1$ nonzero elements of GF(q), except the entries of the $(q+1-i)$th column, which are all zeros.

Replacing each entry in $\mathbf{M}_i$ by its location-vector, we obtain a $(q-1)\times(q-1)^2$ matrix over GF(2), $\mathbf{B}_i=[\mathbf{A}_{i,0}\mathbf{A}_{i,1}\ldots\mathbf{A}_{i,q-2}]$, which consists of a row of $q-1$ $(q-1)\times(q-1)$ square submatrices, where $\mathbf{A}_{i,j}$ is formed with the location-vectors of the $q-1$ entries of the jth column of $\mathbf{M}_i$, $\alpha^{i+j}-1,\alpha(\alpha^{i+j}-1),\ldots,\alpha^{q-2}(\alpha^{i+j}-1)$, as rows. All the submatrices of $\mathbf{B}_i$ are $(q-1)\times(q-1)$ circulant permutation matrices, except $\mathbf{A}_{i,q-1-i}$, which is a $(q-1)\times(q-1)$ zero matrix. All the circulant permutation matrices in $\mathbf{B}_i$ are different. Form the following $(q-1)\times(q-1)$ array of $(q-1)\times(q-1)$ circulant permutation and zero matrices:

$$\mathbf{H}=\begin{bmatrix}\mathbf{B}_0\\ \mathbf{B}_1\\ \vdots\\ \mathbf{B}_{q-2}\end{bmatrix}=\begin{bmatrix}\mathbf{A}_{0,0} & \mathbf{A}_{0,1} & \cdots & \mathbf{A}_{0,q-2}\\ \mathbf{A}_{1,0} & \mathbf{A}_{1,1} & \cdots & \mathbf{A}_{1,q-2}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{A}_{q-2,0} & \mathbf{A}_{q-2,1} & \cdots & \mathbf{A}_{q-2,q-2}\end{bmatrix}, \tag{6}$$

which is a $(q-1)^2\times(q-1)^2$ matrix over GF(2) with both column and row weight $q-2$. It follows from Lemma 1 and the structural properties of matrices $\mathbf{M}$ and $\mathbf{M}_i$ that $\mathbf{H}$ satisfies the RC-constraint.

For any pair of positive integers with $1\leq\gamma,\rho<q$, let $\mathbf{H}(\gamma,\rho)$ be a $\gamma\times\rho$ subarray of $\mathbf{H}$. $\mathbf{H}(\gamma,\rho)$ is a $\gamma(q-1)\times\rho(q-1)$ matrix over GF(2) which also satisfies the RC-constraint. If $\mathbf{H}(\gamma,\rho)$ does not contain zero submatrices of $\mathbf{H}$, it has constant column and row weights γ and ρ, respectively. The null space of $\mathbf{H}(\gamma,\rho)$ gives a (γ,ρ)-regular QC-LDPC code $\mathcal{C}_{qc}$ of length $\rho(q-1)$, rate at least $(\rho-\gamma)/\rho$ and minimum distance at least $\gamma+1$, whose Tanner graph has a girth of at least 6. Since $\mathbf{H}(\gamma,\rho)$ is an array of permutation matrices, no odd number of columns can be added to the zero column vector $\mathbf{0}$, and hence the minimum distance of $\mathcal{C}_{qc}$ must be even. Consequently, the minimum distance of $\mathcal{C}_{qc}$ is at least $\gamma+2$ for even γ and $\gamma+1$ for odd γ. Since the girth of the Tanner graph of $\mathcal{C}_{qc}$ is at least 6, the size of a minimal stopping set in the Tanner graph is at least $\gamma+1$ [19]. If $\mathbf{H}(\gamma,\rho)$ contains some zero submatrices of $\mathbf{H}$, then $\mathbf{H}(\gamma,\rho)$ has two column weights, $\gamma-1$ and γ, and possibly two row weights $\rho-1$ and ρ. In this case, the null space of $\mathbf{H}(\gamma,\rho)$ gives a near-regular QC-LDPC code with minimum distance at least γ for even γ and at least $\gamma+1$ for odd γ. The size of a

minimal stopping set in the code's Tanner graph is either at least γ or $\gamma + 1$. The above construction gives a class of QC-LDPC codes whose Tanner graph have girth at least 6.

4 An Example

In the following, we use an example to illustrate the method of construction of QC-LDPC codes described in Section III and to demonstrate the performances of a code over the AWGN and binary random erasure channels. For the AWGN channel, the code is decoded with the *sum-product algorithm* (SPA) [3,4,7], and its performance is compared with the Shannon limit. For the binary random erasure channel, the code is decoded with the iterative decoding algorithm given in Section II and its performance is compared with the threshold $\epsilon(\gamma, \rho)$ for the degree pair (γ,ρ) of its Tanner graph. We set the maximum number of decoding iterations to 100. We also assume BPSK signaling.

Let GF(73) be the field for code construction. Using this field, we can construct a 72×72 array $\mathbf{H}$ of 72×72 circulant permutation and zero matrices. Set $\gamma = 6$ and $\rho = 72$. We take a 6×72 subarray $\mathbf{H}(6, 72)$ from array $\mathbf{H}$ (the first 6 rows of submatrices of $\mathbf{H}$). Each of the first 6 columns of submatrices of $\mathbf{H}(6, 72)$ contains a single 72×72 zero matrix. Hence $\mathbf{H}(6, 72)$ is a 432×5184 matrix over GF(2) with constant row weight 71 and two column weights, 5 and 6. The null space of $\mathbf{H}(6, 72)$ gives a $(5184, 4752)$ QC-LDPC code with rate 0.917. The performance and the rate of decoding convergency of this code over the AWGN channel are shown in Figure 1. We see that the decoding of this code converges very fast. At the BER of 10^{-6}, the performance gap between 5 iterations and 100 iterations is within 0.2dB. At BER of

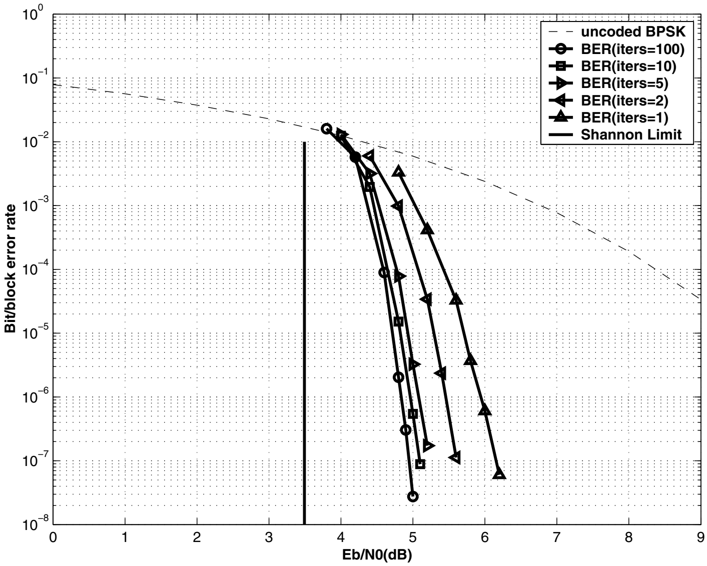

Fig. 1. Performance and the rate of decoding convergence of the (5184,4752) QC-LDPC code given in Section 4 over the AWGN channel

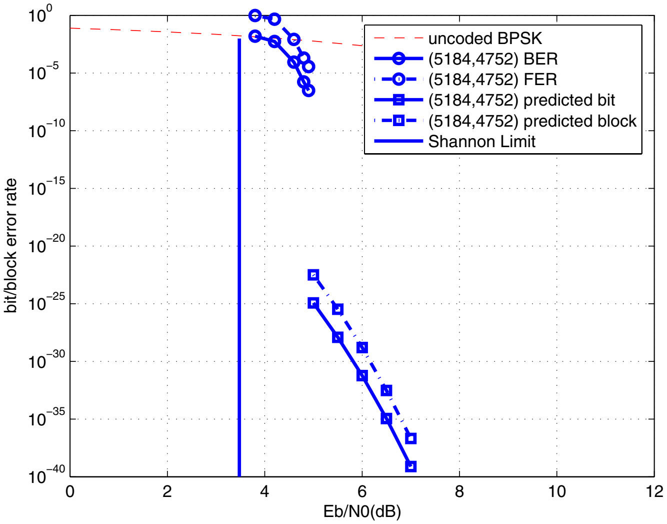

Fig. 2. Estimated error floor of the (5184,4752) QC-LDPC code given in Section 4 over the AWGN channels

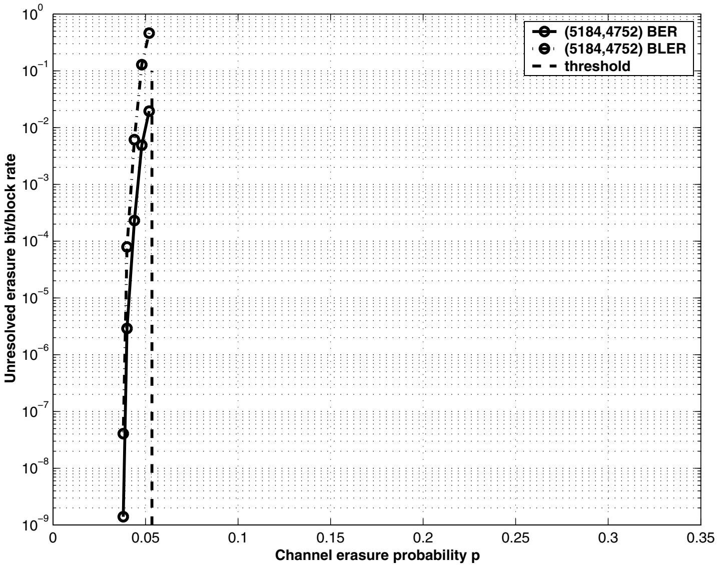

Fig. 3. Performance of the (5184,4752) QC-LDPC code given in Section 4 over the binary random erasure channel

10^{-6} with 100 iterations, the code performs 1.3 dB from the Shannon limit. The error-floor of this code is estimated below the BER of 10^{-25} and the block-error rate (BLER) of 10^{-22} as shown in Figure 2 (using the method given in [21]). The estimated minimum distance of this code is 19. The error performance of this code for the binary random erasure channel is shown in Figure 3. At the BER of 10^{-6}, the code performs 0.002 from the threshold $\epsilon(6, 72) = 0.0528$. Figures 1 to 3 demonstrates that the (5184,4752) QC-LDPC code constructed based on GF(73) performs well on both the AWGN and binary erasure channels.

5 Conclusion

In this paper, we have presented a method for constructing a class of QC-LDPC codes based on finite fields and location-vector representations of finite field elements. The Tanner graphs of the codes in this class have girth of at least 6. For a given finite field, a family of QC-LDPC codes with various lengths, rates, minimum distances and sizes of minimal stopping sets can be constructed. The proposed construction of QC-LDPC codes may be regarded parallel to the construction of BCH codes [22]. A QC-LDPC code was constructed to show that it performs very well over both the AWGN and binary random erasure channels with iterative decoding. It has a very low error-floor. In a succeeding paper, we will use the RC-constrained arrays of circulant permutation matrices constructed based on finite fields together with a masking technique to construct QC-LDPC codes for AWGN, binary random and burst erasure channels.

References

1. R. G. Gallager, "Low density parity-check codes," *IRE Trans. Inform. Theory*, IT-8, pp. 21-28, Jan. 1962.
2. D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity-check codes," *Electro. Lett.*, vol. 32, pp. 1645-1646, Aug. 1996.
3. D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, pp. 399-432, Mar. 1999.
4. T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 599-618, Feb., 2001.
5. T. Richardson, A. Shokrollahi, and R. Urbanke, "Design of capacity-approaching low desnsity parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 619-637, Feb., 2001.
6. R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. 27, pp. 533-547, Sept. 1981.
7. Y. Kou, S. Lin, and M. Fossorier, "Low density parity-check codes based on finite geometries: a discovery and new results," *IEEE Trans. Inform. Theory*, vol. 47, pp. 2711-2736, Nov. 2001.
8. S. J. Johnson and S. R. Weller, "Codes for iterative decoding from partial geometries," *IEEE Trans. Commun.*, vol. 52, no. 2, pp. 236-247, Feb. 2003.
9. I. Djurdjevic, J. Xu, K. Abdel-Ghaffar and S. Lin, "Construction of low-density parity-check codes based on Reed-Solomon codes with two information symbols," *IEEE Commun. Lett.*, vol. 7, no. 7, pp. 317-319, July 2003.
10. L. Chen, J. Xu, I. Djurdjevic, and S. Lin, "Near Shannon limit quasi-cyclic low-density parity-check codes," *IEEE Trans. Commun.*, vol. 52, no. 7, pp. 1038-1042, July 2004.
11. B. Vasic and O. Milenkovic, "Combinatorial construction of low density parity-check codes for iterative decoding," *IEEE Trans. Inform. Theory*, vol 50, no. 6, pp. 1156-1176, June 2004.
12. H. Tang, J. Xu, Y. Kou, S. Lin, and K. Abdel-Ghaffar, "On algebraic construction of Gallager and circulant low density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 50, no. 6, pp 1269-1279, June 2004.

13. B. Ammar, B. Honary, Y. Kou, J. Xu, and S. Lin, "Construction of low density parity-check codes based on balanced imcomplete block designs," *IEEE Trans. Inform. Theory*, vol. 50, vo. 6, pp. 1257-1268, June 2004.
14. L. Chen, L. Lan, I. Djurdjevic, S. Lin, K. Abdel-Ghaffar, "An algebraic method for constructing quasi-cyclic LDPC codes," *Proc. Int. Symp. Inform. Theory and Its Applications*, ISITA2004, pp. 535-539, Parma, Italy, Oct. 10-13, 2004.
15. H. Tang, J. Xu, S. Lin, and K. Abdel-Ghaffar, "Codes on finite geometries," *IEEE Trans. Inform. Theory*, vol. 51, no. 2, pp. 572-596, Feb. 2005.
16. J. Xu, L. Chen, L. -Q. Zeng, L. Lan, and S. Lin, "Construction of low-density parity-check codes by superposition," *IEEE Trans. Commun.*, vol.53, no. 2, pp. 243-251, Feb. 2005.
17. Z. -W. Li, L. Chen, S. Lin, W. Fong, and P. -S. Yeh, "Efficient encoding of quasi-cyclic low-density parity-check codes," *IEEE Trans. Commun.*, (accepted) 2005.
18. C. Di, D. Proietti, I. E. Teletar, T. j. Richarson, and R. L. Urbanke, "Finite length analysis of low-density parity-check codes on binary erasure channel," *IEEE Trans. Inform. Theory*, vol. 48, no. 6, pp. 1570-1579, June 2002.
19. A. Orlitsky, R. L. Urbanke, K. Viswanathan, and J. Zhang, "Stopping sets and the girth of Tanner graph," *Proc. Int. Symp. Inform. Theory*, p.2, Lausanne, Switzerland, June 30-July 5, 2002.
20. T. Tian, C. Jones, J. D. Villasensor, and R. D. Wesel,"Construction of irregular LDPC codes with low error floors," *IEEE ICC'03 Int. Conf. on Commun.*, pp. 3125-3129, 2003.
21. T. Richardson, "Error floors of LDPC codes,"*Proc. Allerton Conf. on Communication, Control and Computing*, pp. 1426-1435, Monticello, IL., Oct. 2003.
22. S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*, 2nd edition, Prentice Hall, Upper Saddle River, NJ., 2004.

Algebraic Construction of Quasi-cyclic LDPC Codes – Part II: For AWGN and Binary Random and Burst Erasure Channels*

Ying Y. Tai, Lingqi Zeng, Lan Lan, Shumei Song, Shu Lin, and Khaled Abdel-Ghaffar

Department of Electrical and Computer Engineering,
University of California, Davis, CA 95616
{yytai, lqzeng, squash, ssmsong, shulin,
ghaffar}@ece.ucdavis.edu

Abstract. This paper is the second part of a sequence of two papers that present several algebraic methods for constructing quasi-cyclic (QC) LDPC codes for AWGN, binary random and burst erasure channels. In the first paper, we presented a class of QC-LDPC codes for both the AWGN and binary random erasure channels. The construction of this class of QC-LDPC codes is based on finite fields and location vector representations of finite field elements. In this paper, we presented two other algebraic methods for constructing QC-LDPC codes for the AWGN, binary random and burst erasure channels.

1 Introduction

This paper is the second part of a sequence of two papers devoted in construction of QC-LDPC codes for three types of channels, namely the AWGN, binary random and burst erasure channels. In the first paper [1], we presented a method based on finite fields and location vector representations of field elements to construct RC-constrained arrays of circulant permutation matrices. From these RC-constrained arrays of circulant permutation matrices, we constructed a class of QC-LDPC codes for both the AWGN and binary random erasure channels. In this paper, we use these arrays and a masking technique to construct QC-LDPC codes for all three types of channels mentioned above.

2 Construction of QC-LDPC Codes by Masking

Consider the RC-constrained arrays of circulant permutation matrices constructed in the first paper. These arrays are highly structured and their component circulant permutation matrices are densely packed. The density of an array can be reduced by replacing a set of circulant permutation matrices by zero matrices. We referred to this replacement of circulant permutation matrices by zero matrices as *masking* [2,3]. Masking results in a new array whose Tanner graph has fewer edges and hence has fewer short cycles and possibly larger girth.

* This research was supported by NASA under the Grant NNG05GD13G and NSF under the Grant CCR-0117891 and ECS-0121469.

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 265–274, 2006.

Consider a $\gamma \times \rho$ array $\mathbf{H}(\gamma, \rho) = [\mathbf{A}_{i,j}]$ of $(q-1) \times (q-1)$ circulant permutation matrices constructed based on the Galois field GF(q). The masking operation can be mathematically formulated as a *special matrix product*. Let $\mathbf{D}(\gamma, \rho) = [d_{i,j}]$ be a $\gamma \times \rho$ matrix over GF(2). Define the following matrix product:

$$\mathbf{M}(\gamma, \rho) = \mathbf{D}(\gamma, \rho) \otimes \mathbf{H}(\gamma, \rho) = [d_{i,j}\mathbf{A}_{i,j}], \tag{1}$$

where $d_{i,j}\mathbf{A}_{i,j} = \mathbf{A}_{i,j}$ for $d_{i,j} = 1$ and $d_{i,j}\mathbf{A}_{i,j} = \mathbf{O}$ (a $(q-1) \times (q-1)$ zero matrix) for $d_{i,j} = 0$. In this product operation, a set of permutation matrices in $\mathbf{H}(\gamma, \rho)$ is masked by the set of 0-entries of $\mathbf{D}(\gamma, \rho)$, We call $\mathbf{D}(\gamma, \rho)$ the *masking matrix*, $\mathbf{H}(\gamma, \rho)$ the *base matrix* (or array), and $\mathbf{M}(\gamma, \rho)$ the *masked matrix* (or array). The distribution of the circulant permutation matrices in the masked matrix $\mathbf{M}(\gamma, \rho)$ is identical to the distribution of the 1-entries in the masking matrix $\mathbf{D}(\gamma, \rho)$. The masked matrix $\mathbf{M}(\gamma, \rho)$ is an array of circulant permutation and zero matrices. Since the base array $\mathbf{H}(\gamma, \rho)$ satisfies the RC-constraint, the masked array $\mathbf{M}(\gamma, \rho)$ also satisfies the RC-constraint regardless of the masking matrix $\mathbf{D}(\gamma, \rho)$. Hence, the Tanner graph of $\mathbf{M}(\gamma, \rho)$ has a girth of at least 6. If the girth of the masking matrix $\mathbf{D}(\gamma, \rho)$ is $g \geq 6$, the girth of the Tanner graph of the masked matrix $\mathbf{M}(\gamma, \rho)$ is at least g. If the size of a masking matrix is not very large, it is quite easy to construct masking matrices with girth, 8 to 12, either by computer search or using the algorithm given in [4, 5].

The null space of the masked matrix $\mathbf{M}(\gamma, \rho)$ gives a QC-LDPC code, whose Tanner graph has a girth of at least 6. If the masking matrix $\mathbf{D}(\gamma, \rho)$ is a regular matrix, then the null space of $\mathbf{M}(\gamma, \rho)$ gives a regular QC-LDPC code, otherwise it gives an irregular QC-LDPC code. Masking is very effective for constructing long LDPC codes, regular or irregular. The performance of an LDPC code constructed by masking depends on the choice of the masking matrix. How to design masking matrices to optimize the performance of codes is a challenging research problem that needed to be studied. Regular masking matrices can be constructed using algebraic or combinatorial methods. An irregular masking matrix can be constructed by computer search based on the variable- and check-node degree distributions of a code graph derived by the evolution of the probability densities of the messages passed between the two types of nodes as proposed in [6].

Example 1. In this example, we construct three regular QC-LDPC codes using the masking technique presented in this section. For code construction, we use the prime field GF(257). Based on this field, we construct a 256×256 array $\mathbf{H}$ of 256×256 circulant permutation matrices. We choose three pairs of integers, (γ, ρ)'s, as follows: (1) $\gamma = 8$ and $\rho = 32$; (2) $\gamma = 8$ and $\rho = 64$; and (3) $\gamma = 8$ and $\rho = 128$. Based on these choices of (γ, ρ)'s, we form three subarrays, $\mathbf{H}(8, 32)$, $\mathbf{H}(8, 64)$, and $\mathbf{H}(8, 128)$, of $\mathbf{H}$ as the base arrays for masking. In forming these arrays, we avoid the inclusion of zero matrices in $\mathbf{H}$. Next we design three regular masking matrices, $\mathbf{D}(8, 32)$, $\mathbf{D}(8, 64)$, and $\mathbf{D}(8, 128)$, which are rows of four, eight, and sixteen 8×8 circulants, respectively, each circulant having both column and row weights 4. The first rows (called the *generators*) of the circulants in each masking matrix are given in Table 1. Masking the three base arrays with the designed masking matrices, we obtain three masked matrices, $\mathbf{M}(8, 32)$, $\mathbf{M}(8, 64)$, and $\mathbf{M}(8, 128)$. The null spaces of these masked matrices give three regular QC-LDPC codes that are: $(8192, 6145)$, $(16384, 14337)$, and $(32768, 30721)$ codes

Table 1. Generators of circulants in the masking matrices of Example 1

$\mathbf{D}(8,32)$	$\boldsymbol{d}_1 = (01101010)$	$\boldsymbol{d}_2 = (10101010)$
	$\boldsymbol{d}_3 = (11001100)$	$\boldsymbol{d}_4 = (00110110)$
$\mathbf{D}(8,64)$	$\boldsymbol{d}_1 = (10011010)$	$\boldsymbol{d}_2 = (11011000)$
	$\boldsymbol{d}_3 = (00111010)$	$\boldsymbol{d}_4 = (01100110)$
	$\boldsymbol{d}_5 = (01111000)$	$\boldsymbol{d}_6 = (11100010)$
	$\boldsymbol{d}_7 = (11010010)$	$\boldsymbol{d}_8 = (01010110)$
$\mathbf{D}(8,128)$	$\boldsymbol{d}_1 = (10100100)$	$\boldsymbol{d}_2 = (01101010)$
	$\boldsymbol{d}_3 = (10101100)$	$\boldsymbol{d}_4 = (10100110)$
	$\boldsymbol{d}_5 = (01011100)$	$\boldsymbol{d}_6 = (10111000)$
	$\boldsymbol{d}_7 = (01010110)$	$\boldsymbol{d}_8 = (01110010)$
	$\boldsymbol{d}_9 = (10010110)$	$\boldsymbol{d}_{10} = (01011010)$
	$\boldsymbol{d}_{11} = (00011110)$	$\boldsymbol{d}_{12} = (11000110)$
	$\boldsymbol{d}_{13} = (00111010)$	$\boldsymbol{d}_{14} = (01011010)$
	$\boldsymbol{d}_{15} = (00111010)$	$\boldsymbol{d}_{16} = (11001100)$

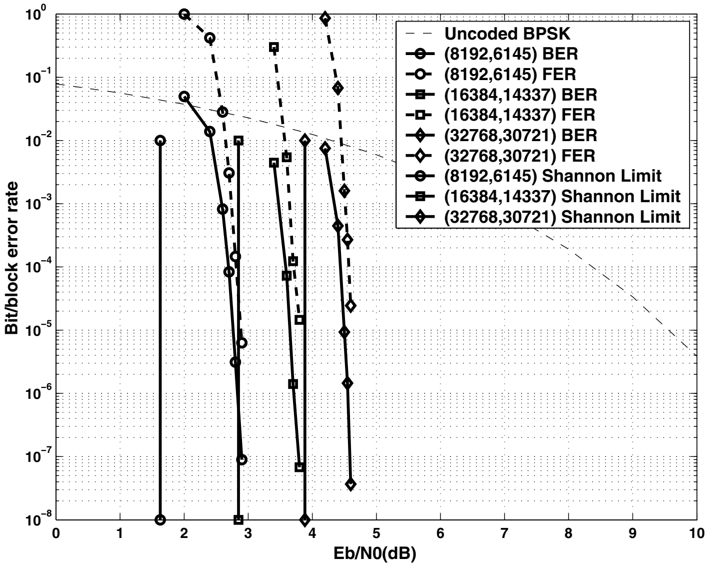

Fig. 1(a). Performances of the three regular QC-LDPC codes given in Example 1 over the AWGN channel

with rates 0.7501, 0.8750, and 0.9375, respectively. Their performances on the AWGN and binary random erasure channels are shown in Figures 1(a) and 1(b), respectively. We see that they perform well on both channels. For example, at the BER of 10^{-6}, the $(32768, 30721)$ code performs 0.65 dB from the Shannon limit for the AWGN channel

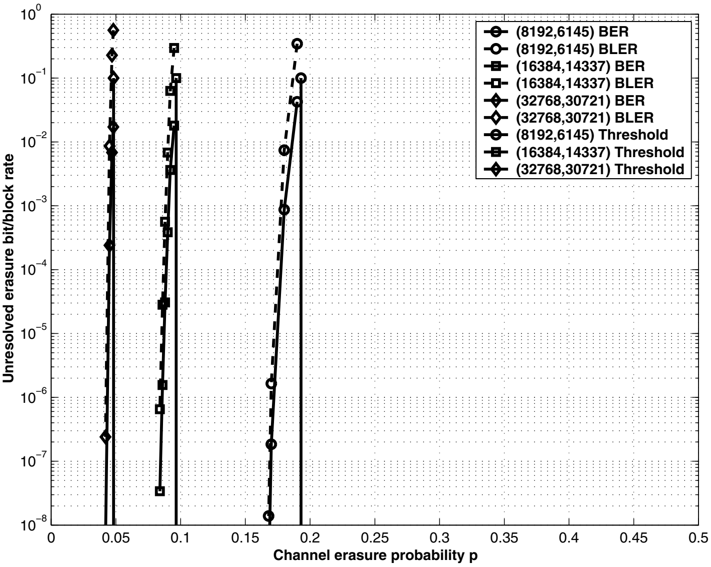

Fig. 1(b). Performances of the three regular QC-LDPC codes given in Example 1 over the binary random erasure channel

and 0.009 from the threshold $\varepsilon(4, 64) = 0.0482$ for the degree distribution pair $(4, 64)$ for the binary random erasure channel. $\triangle\triangle$

An irregular LDPC code is given by the null space of a sparse matrix $\mathbf{H}$ with varying column weights and/or varying row weights. Consequently, its Tanner graph $\mathcal{T}$ has varying variable-node degrees and/or varying check-node degrees. The *degree distributions* of these two type of nodes are expressed in terms of two polynomials [6], $\boldsymbol{v}(X) = \sum_{i=1}^{d_v} v_i X^{i-1}$ and $\boldsymbol{c}(X) = \sum_{i=1}^{d_c} c_i X^{i-1}$, where v_i and c_i denote the fractions of variable- and check-nodes in $\mathcal{T}$ with degree i, respectively, d_v and d_c denote the maximum variable- and check-node degrees, respectively. Since the variable- and check-nodes of $\mathcal{T}$ correspond to the columns and rows of the parity-check matrix $\mathbf{H}$, $\boldsymbol{v}(X)$ and $\boldsymbol{c}(X)$ also give the *column and row weight distributions* of $\mathbf{H}$. It has been shown that the error performance of an irregular LDPC code depends on the variable- and check-node degree distributions of its Tanner graph [6] and Shannon limit approaching codes can be designed by optimizing the two degree distributions based on density evolution. In code construction, once the degree distributions $\boldsymbol{v}(X)$ and $\boldsymbol{c}(X)$ have been derived, a code graph is constructed by connecting the variable-nodes and check-nodes with edges based on the degree distributions. Since the selection of edges in the construction of a code graph is not unique, edge selection is carried out in a random manner by computer search. During the edge selection process, effort must be made to ensure that the code graph does not contain short cycles. Once a code graph is constructed, the corresponding parity-check matrix $\mathbf{H}$ is formed based on the edges that connect the variable- and the check-nodes of $\mathcal{T}$. The null space of $\mathbf{H}$ gives a random irregular LDPC code.

Irregular QC-LDPC codes can be constructed based on masking an array of circulant permutation matrices. First we design the degree distributions, $\boldsymbol{v}(X)$ and $\boldsymbol{c}(X)$, of the variable- and check-nodes of the Tanner graph of a code with desired rate R based on density evolution. Then choose proper γ, ρ, and GF(q) that give the desired code length and rate R (or close to the desired code length and rate). The condition $\gamma \geq d_v$ and $\rho \geq d_c$ must be met. By computer search, we construct a masking matrix $\mathbf{D}(\gamma, \rho)$ with column and row weight distributions identical (or close to) $\boldsymbol{v}(X)$ and $\boldsymbol{c}(X)$. Construct a base array $\mathbf{H}(\gamma, \rho)$ of circulant permutation matrices using the method given in part-I [1]. Masking the base array $\mathbf{H}(\gamma, \rho)$ with $\mathbf{D}(\gamma, \rho)$, we obtain a masked matrix $\mathbf{M}(\gamma, \rho)$ with column and row weight distributions identical (or close to) $\boldsymbol{v}(X)$ and $\boldsymbol{c}(X)$. Then the null space of the masked matrix gives a QC-LDPC code whose Tanner graph has degree distributions identical (or close to) $\boldsymbol{v}(X)$ and $\boldsymbol{c}(X)$. This masking construction not only gives an irregular QC-LDPC code that can be efficiently encoded but also simplifies the code construction significantly. Since the Tanner graph of the base array is already free of cycles of length 4, the Tanner graph of the resultant code is also free of cycles of length 4. However, in random construction, a large random bipartite graph based on the degree distributions must first be constructed. In the process of constructing a code graph by computer, effort must be made to avoid cycles of length 4.

Example 2. Suppose we want to construct an irregular QC-LDPC code of length equal or close to $10^{13} = 8196$ with rate about $7/8$ based on the degree distributions, $\boldsymbol{v}(X) = 0.09375X + 0.0625X^2 + 0.84375X^3$ and $\boldsymbol{c}(X) = 0.25X^{28} + 0.5X^{29} + 0.25X^{30}$. First we use the field GF(2^8) to construct an RC-constrained 255×255 array $\mathbf{H}$ of 255×255 circulnat permutation matrices. The largest degrees in $\boldsymbol{v}(X)$ and $\boldsymbol{c}(X)$ are 4 and 31,

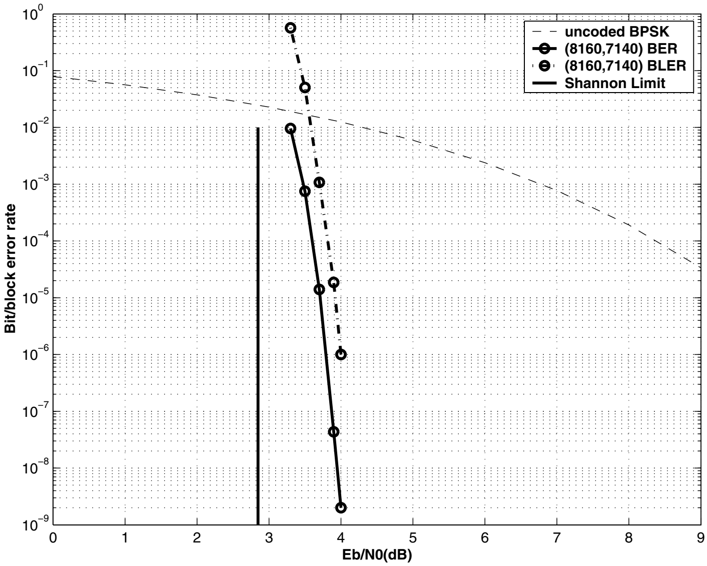

Fig. 2. Performance of the $(8160, 7140)$irregular LDPC code constructed by masking in Example 2 over the AWGN channel

respectively. We set $\gamma = 4$ and $\rho = 32$. Choose a 4×32 subarray $\mathbf{H}(4, 32)$ from $\mathbf{H}$, avoiding the zero matrices in $\mathbf{H}$. $\mathbf{H}(4, 32)$ is a 1020×8160 matrix with column and row weights, 4 and 32, respectively. Next we construct a 4×32 masking matrix $\mathbf{D}(4, 32)$ with column and row weight distributions identical to $\boldsymbol{v}(X)$ and $\boldsymbol{c}(X)$. Masking $\mathbf{H}(4, 32)$ with $\mathbf{D}(4, 32)$, we obtain a 1020×8160 masked matrix $\mathbf{M}(4, 32)$ with column and row weight distribution identical to $\boldsymbol{v}(X)$ and $\boldsymbol{c}(X)$. The null space of $\mathbf{M}(4, 32)$ gives a $(8160, 7140)$ irregular QC-LDPC code with rate 0.875. The performance of this code over the AWGN channel is shown in Figure 2. At the BER of 10^{-6}, the code performs 0.9 dB from the Shannon limit. △△

3 LDPC Codes for Binary Burst Erasure Channels

There are binary erasure channels over which erasures cluster in bursts. Such channels are called burst erasure channels. An erasure pattern $\mathcal{E}$ is called an erasure-burst of length l if the erasures in $\mathcal{E}$ are confined to l consecutive locations, the first and last of which are erasures. Erasure bursts occur in recording and fading channels. In this section, we consider designing LDPC codes for correcting erasure-bursts.

Let $\boldsymbol{x} = (x_0, x_1, \cdots, x_{n-1})$ be a nonzero n-tuple over GF(2). The first (or the *left-most*) 1-component of $\boldsymbol{x}$ is called the *leading*-1 of $\boldsymbol{x}$ and the last (or the *right-most*) 1-component of $\boldsymbol{x}$ is called the *trailing*-1 of $\boldsymbol{x}$. If $\boldsymbol{x}$ has only 1-component, then the leading-1 and trailing-1 of $\boldsymbol{x}$ are the same. A *zero-span* of $\boldsymbol{x}$ is defined as a sequence of *consecutive zeros* between two 1-components. The zeros to the right of the trailing-1 of $\boldsymbol{x}$ together with the zeros to the left of leading-1 of $\boldsymbol{x}$ also form a zero-span, called the *end-around-zero-span*. The number of zeros in a zero-span is called the *length* of the zero-span. Given a nonzero binary n-tuple $\boldsymbol{x}$, we can determine all its zero-spans and compute their lengths. A zero-span in $\boldsymbol{x}$ with the longest length is called a *maximal* zero-span of $\boldsymbol{x}$ (not unique). Consider the binary 16-tuple, $\boldsymbol{x} = (0010001000000100)$. It has three zero-spans with lengths, 3, 6, and 4, respectively. The end-around zero-span, $(001 \ldots 100)$, has a length of 4.

Consider a (γ, ρ)-regular LDPC code $\mathcal{C}$ given by the null space of an RC-constrained $J \times n$ parity-check matrix $\mathbf{H}$ with constant column weight γ. Label the columns of $\mathbf{H}$ from 0 to $n - 1$. For $0 \leq j < n$, there are γ rows of $\mathbf{H}$, each having a "1" in the j-th column of $\mathbf{H}$. For each of these γ rows, we find its zero-span starting from the $(j+1)$-th column to the next 1-component and compute its length. If the 1-component of a row at the position j (or in the j-th column of $\mathbf{H}$) is the trailing-1 of the row, we determine its end-around zero-span. Among the zero-spans of the γ rows with a "1" in the j-th column of $\mathbf{H}$, the *longest one*, denoted δ_j, is called the *zero-span* of column j. Define $\delta \triangleq min_{0 \leq j < n} \delta_j$. This parameter δ is called the *zero-span of the parity-check matrix* $\mathbf{H}$. The next lemma gives a lower bound on the erasure-burst-correction capability of an LDPC code in terms of the zero-span of its parity-check matrix.

Lemma 1. *An LDPC code $\mathcal{C}$ given by the null space of a parity-check matrix $\boldsymbol{H}$ over $GF(2)$ with zero-span δ is capable of correcting any erasure-burst of length at least up to $\delta + 1$.*

Proof. Consider an erasure-burst $\mathcal{E}$ of length $\delta + 1$ or shorter whose first erasure occurs at position j with $0 \leq j < n$. Since the zero-span of the parity-check matrix $\mathbf{H}$ is δ, there exists a row in **H** that checks the erasure at the position j and no other erasures in $\mathcal{E}$. From (3) of Part-I, the value of the erasure at position j can be uniquely determined. Once the value of the erasure at the position j is determined, the erasure-burst $\mathcal{E}$ is shortened. Let $\mathcal{E}_1$ be the new erasure-burst. The length of $\mathcal{E}_1$ is at most δ. Determine the position, say k, of the first erasure in $\mathcal{E}_1$. Since the length of $\mathcal{E}_1$ is at most δ, there exists at least a row in **H** that checks the erasure at position k and no other erasures in $\mathcal{E}_1$. Find this row and determine the value of the erasure at position k. This results in another shorter erasure-burst $\mathcal{E}_2$. We repeat the above correction process step by step until the values of all the erasures in $\mathcal{E}$ are determined. It follows from the definition of the zero-span of the parity-check matrix **H** that at each step of decoding, there always exists a row in **H** that checks only the first erasure of the erasure-burst obtained in the previous step.

The parameter $\delta + 1$ gives a lower bound on the erasure-burst correction capability of a regular LDPC code whose parity-check matrix has a zero-span δ. Lemma 1 implies that no $\delta + 1$ or fewer consecutive variable nodes in the Tanner graph of an LDPC code cannot form a stopping set. The zero-span δ of a parity-check matrix and Lemma 1 can be generalized to irregular LDPC codes.

Decoding an erasure-burst based on the zero-spans of columns of the parity-check matrix of an LDPC code can be carried out iteratively as follows:

(1) If there are erasures in the received vector $\boldsymbol{y}$, determine the starting position of the erasure-burst in $\boldsymbol{y}$, say position j, and go to Step 2. If there is no erasure in $\boldsymbol{y}$, stop the decoding.
(2) Determine the length of the erasure-burst, say l. If $l \leq \delta_j + 1$, go to Step 3, otherwise stop the decoding.
(3) Determine the value of the erasure at the position j and go to Step 1.

The above decoding algorithm actually corrects many erasure-bursts of lengths longer than $\delta + 1$. The iterative decoding algorithm for correcting random erasures given in Section II of Part-I can also be used for decoding erasure-bursts, but the above algorithm is simpler.

4 A Class of QC-LDPC Codes for Correcting Erasure-Bursts

A class of QC-LDPC codes for erasure-burst correction can be constructed by masking the arrays presented in Section 3 of Part-I using a special class of masking matrices.

For $k, l \geq 2$, we form l k-tuples, $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots \boldsymbol{u}_l$ over GF(2) where: (1) $\boldsymbol{u}_1 = (100\ldots0)$ consists of a single 1-component at the first position followed by $k - 1$ consecutive zeros; (2) $\boldsymbol{u}_l = (011\ldots1)$ consists of a single 0-component at the first position followed by $k - 1$ consecutive 1-components; and (2) the other k-tuples, $\boldsymbol{u}_2$ to $\boldsymbol{u}_{l-1}$ are zero k-tuples, i.e., $\boldsymbol{u}_2 = \ldots = \boldsymbol{u}_{l-1} = (00\ldots0)$. Let $\boldsymbol{u} = (\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_l)$. Then $\boldsymbol{u}$ is a kl-tuple with a single 1-component at the left-end and $k - 1$ one-components at the right-end and has weight k. This kl-tuple has one and only one zero-span of nonzero length that is $k(l - 1)$.

Form a $kl \times kl$ circulant **Q** with $\boldsymbol{u}$ as the first row and $kl - 1$ right cyclic-shifts of $\boldsymbol{u}$ as the other $kl - 1$ rows. As a circulant, each row of **Q** is the right cyclic-shift of the row above it and the first row is the right cyclic-shift of the last row; and each column of $\mathbf{Q}$ is the downward cyclic-shift of the column on its left and the first column is the downward cyclic-shift of the last column. It follows from the cyclic-shift structure of the rows and columns of $\mathbf{Q}$ that the zero-span of $\mathbf{Q}$ is $k(l-1)$. For $1 \leq m$, we form a $kl \times klm$ matrix over GF(2),

$$\mathbf{D}(kl, klm) = [\mathbf{Q}\ \mathbf{Q} \ldots \mathbf{Q}], \tag{2}$$

that consists of a row of m $\mathbf{Q}$-matrices. We readily see that each column of $\mathbf{D}(kl, klm)$ is downward cyclic-shift of the of the column on its left (including the transition across of the boundary of two neighbor $\mathbf{Q}$-matrices) and the first column of $\mathbf{D}(kl, klm)$ is the downward cyclic-shift of its last column. Each row has m zero-spans, each with length $(l-1)k$. Consequently, the zero-span of $\mathbf{D}(kl, klm)$ is $(l-1)k$.

Take a $kl \times klm$ subarray $\mathbf{H}(kl, klm)$ of $(q-1) \times (q-1)$ circulant permutation matrices from the array $\mathbf{H}$ constructed based on the field GF(q) such that $klm < q$ and $\mathbf{H}(kl, klm)$ does not contains any zero matrix of $\mathbf{H}$, Masking $\mathbf{H}(kl, klm)$ with $\mathbf{D}(kl, klm)$, we obtained a $kl \times klm$ masked array $\mathbf{M}(kl, klm)$ of $(q-1) \times (q-1)$ circulant permutation and zero matrices. In each column of $\mathbf{M}(kl, klm)$ (as an array), there is a circulant permutation matrix that is followed by $(l-1)k$ $(q-1) \times (q-1)$ zero matrices. Hence $\mathbf{M}(kl, klm)$ is a $kl(q-1) \times klm(q-1)$ RC-constrained matrix over GF(2) with a zero-span of length at least $k(l-1)(q-1)$. The column and row weights of $\mathbf{M}(kl, klm)$ are k and km, respectively. The null space of $\mathbf{M}(kl, klm)$ gives a regular QC-LDPC code $\mathcal{C}$ of length $klm(q-1)$ with rate at least $(m-1)/m$ and minimum distance at least $k+1$ or $k+2$, whose Tanner graph has a girth of at least 6. The code is capable of correcting any erasure-burst of length at least up to $k(l-1)(q-1)+1$. If $\mathbf{H}(kl, klm)$ contains zero matrices of $\mathbf{H}$, then $\mathbf{M}(kl, klm)$ may contain two different column weights, k and $k-1$, and two row weights, km and $km-1$. In this case, the null space of $\mathbf{M}(kl, klm)$ gives a near-regular QC-LDPC code. The erasure-burst correction capability of the code is still at least $k(l-1)(q-1)+1$. The number of parity-check bits of $\mathcal{C}$ is at most $kl(q-1)$. The ratio σ of the erasure-burst correction capability of a code to the number of its parity-check bits is defined as the *erasure-burst correction efficiency*. The erasure-burst correction efficiency of the QC-LDPC code $\mathcal{C}$ constructed above is lower bounded by $(l-1)/l$. We see that the efficiency approaches to 1 for long codes with large l.

Example 3. Consider the 72×72 array $\mathbf{H}$ of 72×72 circulant permutation and zero matrices constructed based on the multiplicative group of the prime field GF(73). Take the first 8 rows of $\mathbf{H}$ and remove the first and the last 7 columns. This results in an 8×64 subarray $\mathbf{H}(8, 64)$ of 72×72 circulant permutation matrices (no zero matrices). Let $k = 4$, $l = 2$ and $m = 8$. Based on (2), we construct an 8×64 masking matrix $\mathbf{D}(8, 64)$ that consists of a row of eight 8×8 $\mathbf{Q}$-matrices. The zero-span of $\mathbf{D}(8, 64)$ is 4. Masking $\mathbf{H}(8, 64)$ with $\mathbf{D}(8, 64)$, we obtain a 576×4608 masked matrix $\mathbf{M}(8, 64)$ with a zero-span at least 288. The column and row weights of $\mathbf{M}(8, 64)$ are 4 and 32, respectively. The null space of $\mathbf{M}(8, 64)$ gives a $(4608, 4033)$ QC-LDPC code with rate 0.8752 that is capable of correcting any erasure-burst of length up to at least 289. By computer search, we find the that code can actually correct any erasure-burst of length up to 375. Thus the erasure-burst correction efficiency of the code is 0.652. The performances of

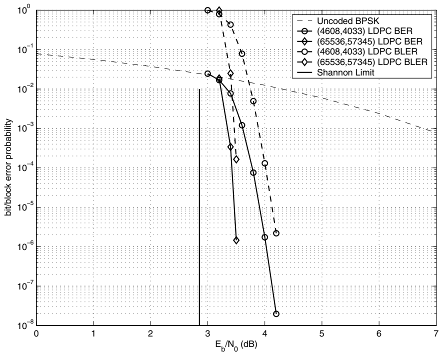

Fig. 3(a). Performances of the (4608,4033) and (65536,57345) QC-LDPC codes given in Examples 3 and 4 over the AWGN channel

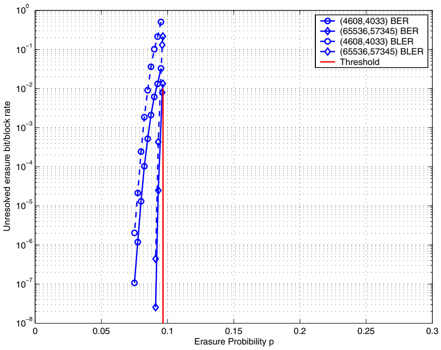

Fig. 3(b). Performances of the (4608,4033) and (65536,57345) QC-LDPC codes given in Examples 3 and 4 over the binary random erasure channel

this code with iterative decoding over the AWGN and binary random erasure channels are shown in Figures 3(a) and 3(b). For the AWGN channel, it performs 1.15 dB from the Shannon limit at the BER of 10^{-8}. For the BEC channel, it performs 0.018 from the threshold $\epsilon(4, 32) = 0.0966$ for the degree pair $(4, 32)$ at the BER of 10^{-6}. The code performs well for all three types of channels. △△

Example 4. Suppose we use the prime field GF(257) for code construction. Based on this field, we construct a 256×256 array $\mathbf{H}$ of 256×256 circulant permutation matrices. Set $k = 4$, $l = 8$ and $m = 8$. Based on (2), we construct a 32×256 masking matrix $\mathbf{D}(32, 256)$. Take a 32×256 subarray $\mathbf{H}(32, 256)$ from $\mathbf{H}$. Masking $\mathbf{H}(32, 256)$ with $\mathbf{D}(32, 256)$, we obtain a 32×256 masked array $\mathbf{M}(32, 256)$ of 256×256 circulant permutation and zero matrices. The zero-span of $\mathbf{M}(32, 256)$ is at least 7168. The null space of $\mathbf{M}(32, 256)$ gives a $(65536, 57345)$ regular QC-LDPC code with rate 0.875. The code is capable of correcting any erasure-burst of length at least up to 7169. Its erasure-burst correction efficiency is at least 0.875. This code also perform well over the AWGN and binary erasure channels as shown in Figures 3(a) and 3(b). At the BER of 10^{-6}, it performs 0.6 dB away from the Shannon limit for the AWGN channel and 0.00357 from the threshold $\varepsilon(4, 32) = 0.0966$ for the degree distribution pair (4,32). △△

5 Conclusion

In this paper, we first constructed a class of QC-LDPC codes for the AWGN and binary random erasure channels based on masking RC-constrained arrays of circulant permutation matrices. Then we discussed erasure-burst correction capability of an LDPC code in terms of the zero-span of its parity-check matrix. A simple iterative method for recovering an erasure-burst was presented. Finally, a class of QC-LDPC codes for correcting erasure-burst was constructed based on masking RC-constrained arrays of circulant permutation matrices with a special class of masking matrices.

References

1. Lan Lan, Lingqi Zeng, Ying Y. Tai, Lei Chen, and Shu Lin, "Algebraic Construction of Quasi-Cyclic LDPC Codes - Part I: For AWGN and Binary Random Erasure Channels," *The proceedings of the 16th Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, Las Vegas, Nevada, Feb.20-24, 2006, Springer-Verlag, 2006.
2. J. Xu, L. Chen, I. Djurdjevic, S. Lin, and K. Abdel-Ghaffar, "Construction of regular and irregular LDPC codes: geometry decomposition and masking," submitted to *IEEE Trans. Inform. Theory*, 2004 (in revision).
3. L. Chen, I. Djurdjevic, J. Xu, S. Lin, and K. Abdel-Ghaffar, "Construction of QC-LDPC codes based on the minimum-weight codewords of Reed-Solomon codes," *Proc. IEEE Int. Symp. Inform. Theory*, p. 239, Chicago, IL., June 27-July 2, 2004.
4. X. -Y. Hu, E. Eleftheriou, and D. -M. Arnold, "Progressive edge-growth Tanner graphs," *Proc. IEEE GLOBECOM*, San Antonio, TX, Nov. 2001.
5. L. Lan, Y. Y. Tai, L. Chen, s. Lin, and Abdel-Ghaffar, "A trellis-based method for removing cycles from bipartite graphs and construction of low density parity check codes," *IEEE Commun. Lett.*, vol. 8, no. 7, pp. 443-445, Jul. 2004.
6. T. Richardson, A. Shokrollahi, and R. Urbanke, "Design of capacity-approaching low desnsity parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 619-637, Feb., 2001.

New Constructions of Quasi-cyclic LDPC Codes Based on Two Classes of Balanced Incomplete Block Designs: For AWGN and Binary Erasure Channels

Lan Lan[1], Ying Yu Tai[1], Shu Lin[1], Behshad Memari[2], and Bahram Honary[2]

[1] Department of Electrical and Computer Engineering,
University of California, Davis, CA 95616
[2] Department of Communication Engineering,
Lancaster University, Lancaster, LA14YW, UK

Abstract. This paper presents two new methods for constructing quasi-cyclic LDPC codes using certain special classes of balanced incomplete block designs constructed based on finite fields. The codes constructed perform well with iterative decoding over both AWGN and binary erasure channels.

1 Introduction

In 1939, Bose presented a method [1], to construct several classes of balanced incomplete block designs (BIBD's) [2] based on finite fields. These BIBD's were recently used to construct LDPC codes [3] for the AWGN channel [4]. LDPC codes of practical lengths constructed from these BIBD's with iterative decoding using the *sum-product algorithm* (SPA) [5]-[8] perform very well over the AWGN channel and close to the Shannon limit. In this paper, we present two new methods for constructing quasi-cyclic (QC) LDPC codes based on two classes of BIBD's constructed by Bose for both AWGN and binary erasure channels. QC-LDPC codes have encoding advantage over other types of LDPC codes. Encoding of a QC-LDPC code can be accomplished using simple shift-registers with linear complexity [9]. Well designed QC-LDPC codes can perform just as well as random LDPC codes in terms of bit-error probability, block-error probability, error-floor, and rate of decoding convergence, collectively [10]-[14].

A binary regular LDPC code [3] is given by the null space of a sparse parity-check matrix $\mathbf{H}$ with the following structural properties: (1) each row has constant weight ρ; (2) each column has constant weight γ; (3) no two rows (or two columns) have more than one 1-component in common; and (4) ρ and γ are small compared to the length of the code. Property (3) is referred to as the *row-column (RC) constraint*. The RC-constraint ensures that the Tanner graph [5] of the code is free of cycles of length 4 and the minimum distance of the code is at least $\gamma + 1$ [15], [16]. Since the parity-check matrix has constant column and row weights γ and ρ, respectively, we say that the parity-check matrix $\mathbf{H}$ is (γ, ρ)-regular and the code given by the null space of $\mathbf{H}$ is called a (γ, ρ)-regular LDPC code. An LDPC code is *quasi-cyclic* if its parity-check matrix is an *array of sparse circulants* of the same size [10], [15], [17].

The performance of an LDPC code over an AWGN channel with iterative decoding depends on a number of code structural properties besides its minimum distance. One

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 275–284, 2006.

such structural property is the *girth* of the code that is defined as the length of the shortest cycle in the code's Tanner graph. For a code to perform well with iterative decoding, its Tanner graph must not contain short cycles. The shortest cycles that affect code performance the most are cycles of length 4. In code construction, these cycles must be avoided. A code that is free of cycles of length 4 in its Tanner graph has a girth of at least 6.

Let $\mathcal{V}$ be a set of variable nodes in the Tanner graph $\mathcal{T}$ of an LDPC code and $\mathcal{S}$ be a set of check nodes in $\mathcal{T}$ that are adjacent to the nodes in $\mathcal{V}$, i.e., each check node in $\mathcal{S}$ is connected to at least one variable node in $\mathcal{V}$. The nodes in $\mathcal{S}$ are called the neighbors of nodes in $\mathcal{V}$. A set $\mathcal{V}$ of variable nodes in $\mathcal{T}$ is called a *stopping set* of $\mathcal{T}$ if each check node in the neighbor set $\mathcal{S}$ of $\mathcal{V}$ is connected to at least two variable nodes in $\mathcal{V}$ [18]. A stopping set of minimum size is called a *minimal stopping set* (not unique). The performance of an LDPC code over a binary erasure channel (BEC) with iterative decoding is determined by the distribution of stopping sets in its Tanner graph, especially the size of its minimal stopping sets [18]. A stopping set corresponds to an erasure pattern that can not be recovered. For an LDPC code to achieve good performance over the BEC, the size of its minimal stopping sets should be made as large as possible. The size of a minimal stopping set in a Tanner graph is proved to be related to the girth of the Tanner graph. It has been proved [19] that the sizes of minimal stopping sets of Tanner graphs with girths 4, 6 and 8 are 2, $\gamma + 1$ and 2γ, respectively. For a (γ, ρ)-regular LDPC code whose parity-check matrix satisfies the RC-constraint, the size of a minimal stopping set is at least $\gamma + 1$.

2 Bose-BIBD's

Let $X = \{x_1, x_2, \ldots, x_q\}$ be a set of q objects. A BIBD of X [1], [2] is a collection of n g-subsets of X, denoted $B_1, B_2, \ldots, B_n$, called *blocks*, such that the following conditions hold: (1) each object x_i appears in exactly r of the n blocks; (2) every two objects appear together in exactly λ of the n blocks; and (3) the number g of objects in each block is small compared to the total number of objects in X. Since a BIBD is characterized by 5 parameters, n, q, r, g, and λ, it is also called a (n, q, r, g, λ)-configuration (or BIBD). For the *special case* $\lambda = 1$, each pair (x_i, x_j) of objects in X appears in *exactly one block*. Consequently, any two blocks can not have more than one object in common. BIBD's of this special type have been used for constructing LDPC codes whose Tanner graphs are free of cycles of length 4 [4].

In his 1939 paper [1], Bose constructed several classes of $(n, q, r, g, 1)$-BIBD's based on finite fields and a technique, called *symmetrically repeated differences*. In this section, we present two classes of Bose-BIBD's which will be used to construct QC-LDPC codes in the next two sections for both the AWGN and BEC channels.

2.1 Class-I Bose-BIBD's

Let t be a positive integer such that $12t + 1$ is a prime. Then there exists a finite field GF$(12t + 1) = \{0, 1, \ldots, 12t\}$ with $12t + 1$ elements under modulo-$(12t + 1)$ addition and multiplication. Let the elements of GF$(12t + 1)$ represent the objects for which

Table 1

Prime Field GF($12t+1$)		Prime Field GF($20t+1$)	
t	(α, c)	t	(α, c)
1	(2,1)	2	(6,3)
6	(5,33)	3	(2,23)
8	(5,27)	12	(7,197)
9	(6,71)	14	(3,173)
15	(2,13)	21	(2,227)
19	(6,199)	30	(7,79)
20	(7,191)	32	(3,631)
23	(5,209)	33	(2,657)
28	(10,129)	35	(2,533)
34	(21,9)	41	(2,713)

a BIBD is to be constructed. Suppose GF($12t + 1$) has a primitive element α such that $\alpha^{4t} - 1 = \alpha^c$, where c is an odd integer less than $12t + 1$. Then there exists a $(n, q, r, g, 1)$-BIBD with $n = t(12t + 1)$, $q = 12t + 1$, $r = 4t$, $g = 4$ and $\lambda = 1$. Since α is a primitive element of GF($12t + 1$), then $\alpha^0 = 1, \alpha, \ldots, \alpha^{12t-1}$ form all the $12t$ nonzero elements of GF($12t + 1$) and $\alpha^{12t} = 1$. Table 1 gives a list of t's that satisfy the condition, $\alpha^{4t} - 1 = \alpha^c$ and $12t + 1$ is a prime.

To form the BIBD with the above parameters, n, q, r, g, and 1, we first form t *base blocks* which are given as follows: $B_i = \{0, \alpha^{2i}, \alpha^{2i+4t}, \alpha^{2i+8t}\}$, for $0 \leq i < t$. From each base block B_i, we can form $12t+1$ blocks by adding each element of GF($12t+1$) in turn to the elements in B_i. This results in $t(12t + 1)$ blocks. For $0 \leq i < t$ and $0 \leq j \leq 12t$, let $B_{i,j} \triangleq \{0 + j, \alpha^{2i} + j, \alpha^{2i+4t} + j, \alpha^{2i+8t} + j\}$ be the block obtained by adding the element j to the elements of the base block B_i.

2.2 Class-II Bose-BIBD's

Let t be a positive integer such that $20t + 1$ is a prime. Then there exists a finite field GF($20t + 1$) $= \{0, 1, \ldots, 20t\}$ with $20t + 1$ elements. Suppose there exists a primitive element α in GF($20t + 1$) such that $\alpha^{4t} + 1 = \alpha^c$, where c is a positive odd integer less than $20t + 1$. Then there exists a $(n, q, r, g, 1)$-BIBD with $n = t(20t + 1)$, $q = 20t + 1$, $r = 5t$, $g = 5$ and $\lambda = 1$. Table 1 gives a list of t's that satisfy the condition, $\alpha^{4t} + 1 = \alpha^c$. To form this BIBD, we first form t base blocks, $B_i = \{\alpha^{2i}, \alpha^{2i+4t}, \alpha^{2i+8t}, \alpha^{2i+12t}, \alpha^{2i+16t}\}$ for $0 \leq i < t$. From each base block B_i, we form $20t + 1$ blocks by adding each element of GF($20t + 1$) in turn to the elements in B_i. This results in $t(20t+1)$ blocks. For $0 \leq j \leq 20t$, the block obtained by adding the element j to the elements in B_i is $B_{i,j} \triangleq \{\alpha^{2i} + j, \alpha^{2i+4t} + j, \alpha^{2i+8t} + j, \alpha^{2i+12t} + j, \alpha^{2i+16t} + j\}$.

3 A New Construction of QC-LDPC Codes Based on Bose-BIBD's

Let p be a prime and GF(p)=$\{0, 1, \ldots, p - 1\}$ be a prime field under the modulo-p addition and multiplication. For an element i in GF(p), we define a p-tuple over GF(2), $\mathbf{z}(i) = (z_0, z_1, z_2, \ldots, z_{p-1})$, whose components correspond to the p elements of GF(p), where the ith component z_i is set to "1" and all the other components are set to "0". This p-tuple is called the *location vector* of element i (the concept of location vectors of elements of a finite field was first introduced in [20]). The 1-components of the location vectors of two different field elements are at two different positions. For any element i in GF(p), the location vector $\mathbf{z}(i+1)$ of element $i+1$ is the *cyclic-shift* (one place to the right) of the location vector $\mathbf{z}(i)$ of element i. It is clear that the elements $i, i+1, \ldots, i+p-1$ under modulo-p addition give all the p elements of GF(p). Let $\mathbf{A}$ be a $p \times p$ matrix over GF(2) with the location vectors of field elements, $i, i+1, \ldots, i+p-1$ as rows. Then $\mathbf{A}$ is a *circulant permutation matrix.*

3.1 Class-I BIBD QC-LDPC Codes

For $0 \leq i < t$, form a $(12t+1) \times 4$ matrix over GF($12t+1$) with the elements of the $12t+1$ blocks, $B_{i,0}, B_{i,1}, \ldots, B_{i,12t}$, of a Class-I Bose-BIBD as rows as follows:

$$\mathbf{Q}_i = \begin{bmatrix} 0 & \alpha^{2i} & \alpha^{2i+4t} & \alpha^{2i+8t} \\ 1 & \alpha^{2i}+1 & \alpha^{2i+4t}+1 & \alpha^{2i+8t}+1 \\ \vdots & \vdots & \ldots & \vdots \\ 12t & \alpha^{2i}+12t & \alpha^{2i+4t}+12t & \alpha^{2i+8t}+12t \end{bmatrix}. \tag{1}$$

Matrix $\mathbf{Q}_i$ has the following structural properties: (1) all the $12t+1$ entries in a column are different and form all the $12t+1$ elements of GF($12t+1$); and (2) any two columns (or two rows) differ in every position. Since each row of $\mathbf{Q}_i$ is a block of a Bose-BIBD, it follows from the structural property of a $(n, q, r, g, 1)$-BIBD that two rows from two different matrices $\mathbf{Q}_i$ and $\mathbf{Q}_j$ have only one element in common. We label the rows of $\mathbf{Q}_i$ from 0 to $12t$ and the columns from 0 to 3. Replacing each entry in $\mathbf{Q}_i$ by its location vector, we obtain a $(12t+1) \times 4(12t+1)$ matrix $\mathbf{M}_i$ over GF(2) that consists of a row of four $(12t+1) \times (12t+1)$ circulant permutation matrix, $\mathbf{M}_i = [\mathbf{A}_{i,0}\ \mathbf{A}_{i,1}\ \mathbf{A}_{i,2}\ \mathbf{A}_{i,3}]$, where $\mathbf{A}_{i,j}$ is formed with the location vectors of the entries of the jth column of $\mathbf{Q}_i$ as rows for $0 \leq j \leq 3$. Next we form the following $t \times 4$ array $\mathbf{Z}_1$ of $(12t+1) \times (12t+1)$ circulant permutation matrices:

$$\mathbf{Z}_1 = \begin{bmatrix} \mathbf{M}_0 \\ \mathbf{M}_1 \\ \vdots \\ \mathbf{M}_{t-1} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{0,0} & \mathbf{A}_{0,1} & \mathbf{A}_{0,2} & \mathbf{A}_{0,3} \\ \mathbf{A}_{1,0} & \mathbf{A}_{1,1} & \mathbf{A}_{1,2} & \mathbf{A}_{1,3} \\ \vdots & \vdots & \ldots & \vdots \\ \mathbf{A}_{t-1,0} & \mathbf{A}_{t-1,1} & \mathbf{A}_{t-1,2} & \mathbf{A}_{t-1,3} \end{bmatrix}. \tag{2}$$

It is a $t(12t+1) \times 4(12t+1)$ matrix over GF(2) with column weight t and row weigh 4. Since the rows of $\mathbf{Z}_1$ correspond to the blocks of a Class-I Bose-BIBD, it follows from the structured properties of $\mathbf{Q}_i$'s that no two rows (or two columns) of $\mathbf{Z}_1$ have more than one 1-component in common. Hence, $\mathbf{Z}_1$ satisfies the RC-constraint.

Let $\mathbf{H}_1$ be the transpose of $\mathbf{Z}_1$, i.e., $\mathbf{H}_1 \triangleq \mathbf{Z}_1^T$. Then $\mathbf{H}_1$ is a $4 \times t$ array of $(12t+1) \times (12t+1)$ circulant permutation matrices. It is a $4(12t+1) \times t(12t+1)$ matrix over GF(2) with column and row weights 4 and t, respectively. Clearly $\mathbf{H}_1$ also satisfies the RC-constraint. Let ρ be a positive integer such that $1 \leq \rho \leq t$. Let $\mathbf{H}_1(4, \rho)$ be a $4 \times \rho$ subarray of $\mathbf{H}_1$. It is a $4(12t+1) \times \rho(12t+1)$ matrix over GF(2) with column and row weights 4 and ρ, respectively. The null space of $\mathbf{H}_1(4,\rho)$ gives a binary $(4,\rho)$-regular QC-LDPC code $\mathcal{C}_{qc,1}$ of length $\rho(12t+1)$ and rate at least $(\rho-4)/\rho$, whose Tanner graph has a girth of at least 6. Since $\mathbf{H}_1(4,\rho)$ is an array of permutation matrices, no odd number of columns of $\mathbf{H}_1(4,\rho)$ can be added to 0. As a results, the minimum distance of $\mathcal{C}_{qc,1}$ must be even. Since $\mathbf{H}_1(4,\rho)$ satisfies the RC-constraint, the minimum distance is at least $4+1=5$. Consequently, the minimum distance of $\mathcal{C}_{qc,1}$ is at least 6. The size of a minimal stopping set in the Tanner graph of $\mathcal{C}_{qc,1}$ is at least 5. The above construction gives a class of $(4,\rho)$-regular QC-LDPC codes whose Tanner graphs have girth at least 6.

Example 1. Let $t=15$. Then $12t+1=181$ is a prime. The element $\alpha=2$ is a primitive element of the prime field GF(181) that satisfies the condition $\alpha^{4t}-1=\alpha^c$ with $c=13$. Based on this field, we can construct a class-I Bose-BIBD with $n=2715$,

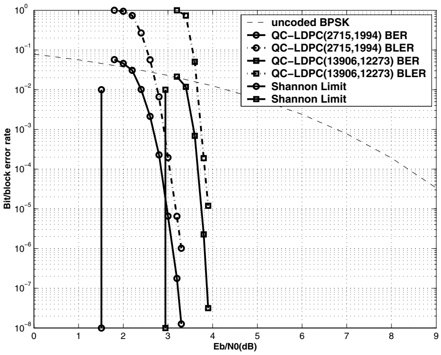

Fig. 1. Error performances of the $(2715,1994)$ and $(13906,12273)$ QC-LDPC codes given in Examples 1 and 2 over the AWGN channel

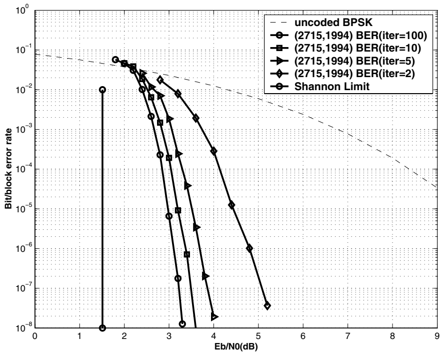

Fig. 2. Rate of decoding convergence of the $(2715,1994)$ QC-LDPC code given in Example 1 over the AWGN channel

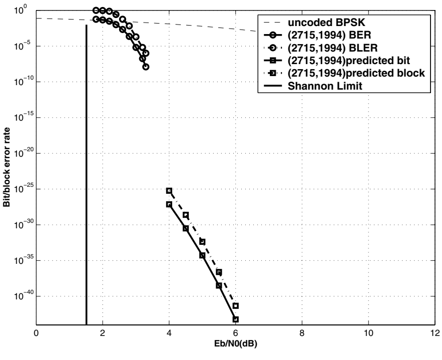

Fig. 3. Estimated error-floor of the $(2715,1994)$ QC-LDPC code given in Example 1 over the AWGN channel

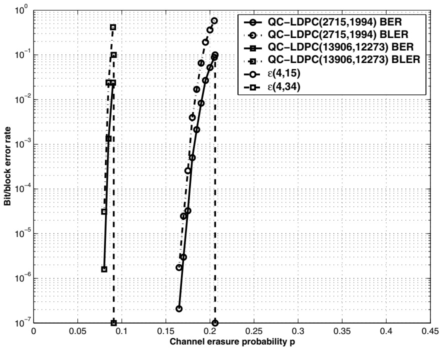

Fig. 4. Error performances of the $(2715,1994)$ and $(13906,12273)$ QC-LDPC codes given in Examples 1 and 2 over the BEC

$q = 181, r = 60, g = 4$ and $\lambda = 1$. Based on this BIBD, we can form a 4×15 array $\mathbf{H}_1$ of 181×181 circulant permutation matrices. Choose $\rho = 15$. Then $\mathbf{H}_1(4, 15) = \mathbf{H}_1$ which is a 724×2715 matrix with column and row weights 4 and 15, respectively. The null space of $\mathbf{H}_1(4, 15)$ gives a $(4, 15)$-regular $(2715, 1994)$ QC-LDPC code with rate 0.7344. Assume BPSK transmission. The performance of this code over the AWGN channel with iterative decoding using the SPA is shown in Figure 1 (maximum number of decoding iterations is set to 100). At the BER of 10^{-6}, the code performs 1.6 dB from the Shannon limit and there is no error-floor down to the BER of 10^{-8}. Figures 2 and 3 show the rate of decoding convergence and the estimated error-floor of the code (using a modified technique given in [21]). We see that the iterative decoding of the code converges very fast and the code has a very low-error floor. The performance of the code over the BEC using the iterative decoding given in [18] is shown in Figure 4. The Tanner graph of this code has degree pair $(4, 15)$. For this degree pair, the threshold $\epsilon(4, 15)$ is 0.2059 [18]. From Figure 4, we see that the code perform 0.036 from the threshold.

Example 2. If we choose $t = 34$, we can construct a 4×34 array $\mathbf{H}_1$ of 409×409 circulant permutation matrices based on the prime field GF(409) (the primitive element $\alpha = 21$ of GF(409) satisfies the condition $\alpha^{4t} - 1 = \alpha^c$ with $c = 9$). Let $\rho = 34$. Then $\mathbf{H}_1(4, 34) = \mathbf{H}_1$. The null space of $\mathbf{H}_1(4, 34)$ gives a $(4, 34)$-regular $(13906, 12273)$ QC-LDPC code with rate 0.8826. The performances of this code over the AWGN and binary erasure channels are also shown in Figures 1 and 4, respectively. For the AWGN channel channel, the code performs 0.85 dB from the Shannon limit at the BER of 10^{-6} has no error-floor down to 5×10^{-8}. For the BEC, the code performs 0.0135 from the threshold $\epsilon(4, 34) = 0.0909$ for the degree pair $(4, 34)$ of its Tanner graph.

3.2 Class-II QC-LDPC Codes

For each t such that $20t + 1$ is a prime, if there is a primitive element α in GF($20t + 1$) which satisfies the condition $\alpha^{4t} + 1 = \alpha^c$, we can construct a Class-II Bose-BIBD. Based on a class-II Bose-BIBD, we can construct a $5 \times t$ array $\mathbf{H}_2$ of $(20t + 1) \times (20t + 1)$ circulant permutation matrices. The construction of this array is the same as the construction of array $\mathbf{H}_1$ based on a Class-I Bose-BIBD. For $3 \leq \gamma \leq 5$ and $3 \leq \rho \leq t$, take a $\gamma \times \rho$ subarray $\mathbf{H}_2(\gamma, \rho)$ from $\mathbf{H}_2$. Then the null space of $\mathbf{H}_2(\gamma, \rho)$ gives a (γ, ρ)-regular QC-LDPC code of length $\rho(20t + 1)$.

Example 3. Let $t = 21$. Then $20t + 1 = 421$ is a prime. Based on the prime field GF(421), we can construct a Bose-BIBD with the following parameters: $n = 8841$, $q = 421, r = 105, g = 5$ and $\lambda = 1$. Based on this BIBD, we can construct a 5×21 array $\mathbf{H}_2$ of 421×421 circulant permutation matrices. Choose $\gamma = 5$ and $\rho = 20$. Take the first 20 columns of circulant permutation matrices to form a 5×20 subarray $\mathbf{H}_2(5, 20)$ of 421×421 circulant permutation matrices. $\mathbf{H}_2(5, 20)$ is a 2105×8420 matrix over GF(2) with column and row weights 5 and 20, respectively. The null space of $\mathbf{H}_2(5, 20)$ gives a $(8420, 6319)$ QC-LDPC code with rate 0.7504. The performances of this code over the AWGN and binary erasure channels with iterative decoding are shown in Figures 5 and 6, respectively.

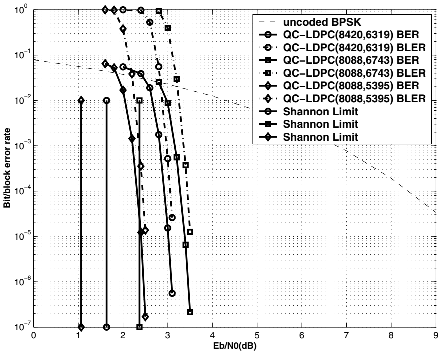

Fig. 5. Error performances of the $(8420, 6319)$, $(8088, 6743)$ and $(8088, 5395)$ QC-LDPC codes given in Examples 3 and 4 over the AWGN channel

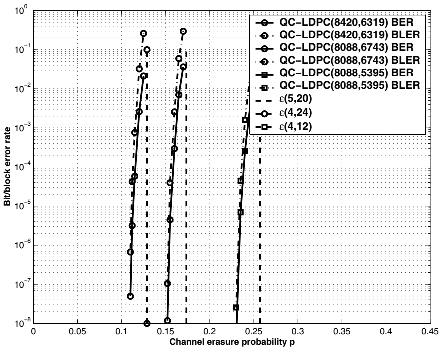

Fig. 6. Error performances of the $(8420, 6319)$, $(8088, 6743)$ and $(8088, 5395)$ QC-LDPC codes given in Examples 3 and 4 over the BEC

4 Construction of QC-LDPC Codes by Circular Dispersion

The component circulant permutation matrices of an array $\mathbf{H}_e$ with $e = 1$ or 2 can be dispersed (or spreaded) to form a new array of circulant permutation and zero matrices. Dispersion reduces the density of an array and hence reduces the number of short cycles in its Tanner graph and may even increase its girth. Dispersion results in a larger array of circulant permutation and zero matrices that still satisfies the RC-constraint.

In this section, we present a *circular dispersion* that maintains the regularity structure of $\mathbf{H}_e$, i.e., constant column weight and constant row weight. Circular dispersion results in QC-LDPC codes that are also capable of correcting bursts of erasures.

For illustration of the dispersion technique, we use a subarray of the array $\mathbf{H}_1$ constructed based on a class-I Bose-BIBD. Let k be a positive integer such that $8k \leq t$. Take a $4 \times 8k$ subarray $\mathbf{H}_1(4, 8k)$ from the array $\mathbf{H}_1$ of circulant permutation matrices constructed based on a Class-I Bose-BIBD. This subarray consists of 4 rows and $8k$ columns of circulant permutation matrices. For $1 \leq i \leq 4$, we label all the circulant permutation matrices in the ith row with integer i. Divide $\mathbf{H}_1(4, 8k)$ into k 4×8 subarrays, $\mathbf{D}_1^{(1)}, \mathbf{D}_2^{(1)}, \ldots, \mathbf{D}_k^{(1)}$, each consisting of 8 consecutive columns of $\mathbf{H}_1(4, 8k)$. The circulant permutation matrices of each subarray $\mathbf{D}_i^{(1)}$ are dispersed to form an 8×8 array of $(12t + 1) \times (12t + 1)$ circulant permutation and zero matrices as follows:

$$\mathbf{E}_i^{(1)} = \left[\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 2 & 2 & 0 & 0 & 0 & 0 & 2 & 2 \\ 3 & 3 & 3 & 0 & 0 & 0 & 0 & 3 \\ 4 & 4 & 4 & 4 & 0 & 0 & 0 & 0 \\ - & - & - & - & - & - & - & - \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 2 & 2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 3 & 3 & 3 & 3 & 0 \\ 0 & 0 & 0 & 0 & 4 & 4 & 4 & 4 \end{array}\right], \tag{3}$$

where a zero in $\mathbf{E}_i^{(1)}$ represents a $(12t+1)\times(12t+1)$ zero matrix. Array $\mathbf{E}_i^{(1)}$ is called a *dispersion array* (or simply dispersion) of $\mathbf{D}_i^{(1)}$. The dispersions of all the k subarrays, $\mathbf{D}_1^{(1)}, \mathbf{D}_2^{(1)}, \ldots, \mathbf{D}_k^{(1)}$, of $\mathbf{H}_1(4, 8k)$ are of the same structure. $\mathbf{E}_i^{(1)}$ consists of four 4×4 subarrays of permutation and zero matrices. $\mathbf{E}_i^{(1)}$ is an $8(12t+1)\times 8(12t+1)$ matrix over GF(2) with both column and row weights 4.

The above dispersion of the k subarrays of $\mathbf{H}_1(4, 8k)$ results in an $8 \times 8k$ array $\mathbf{W}^{(1)}(8, 8k) = [\mathbf{E}_1^{(1)}\mathbf{E}_2^{(1)} \ldots \mathbf{E}_k^{(1)}]$ of $(12t+1) \times (12t+1)$ circulant permutation and zero matrices. It is an $8(12t+1)\times 8k(12t+1)$ matrix over GF(2) with column and row weights 4 and $4k$, respectively. The null space of $\mathbf{W}^{(1)}(8, 8k)$ gives a $(4, 4k)$-regular QC-LDPC code with rate at least $(k-1)/k$, whose Tanner graph has a girth of at least 6.

So far, we have only presented QC-LDPC codes for AWGN and binary random erasure channels. Over a binary random erasure channel, erasures occur at random locations. However, there are erasure channels over which erasures cluster into bursts, such as recording, jamming and some fading channels. An erasure pattern $\mathcal{E}$ is called an erasure-burst of length l if the erasures in $\mathcal{E}$ are confined to l consecutive locations, the first and the last of which are erasures. QC-LDPC codes constructed using circular dispersion are also effective for correcting erasure-bursts.

From (3), we readily see that for each column of the array $\mathbf{W}^{(1)}(8, 8k)$, there is a $(12t+1) \times (12t+1)$ circulant permutation matrix that is followed by four $(12t+1) \times (12t+1)$ zero matrices, including the end-around case. This implies that for any column j in the matrix $\mathbf{W}^{(1)}(8, 8k)$, there exists a row with a 1-component at the jth position that is followed by at least $4(12t+1)$ zeros. As a results, if an erasure-burst of length $4(12t+1)+1$ or shorter occurs and starts at the position j, then this row gives a check-sum that contains the erasure at the position j but not other erasures. From this check-sum, we can determine the value of the erasure at position j. In the same manner, we can determine the values of the other erasures in the burst. Therefore, the code given by the null space of $\mathbf{W}^{(1)}(8, 8k)$ is capable of correcting any erasure-burst of length at least $4(12t+1)+1$, including the end-around erasure-bursts.

Example 4. Consider the 4×28 array $\mathbf{H}_1$ of 337×337 circulant permutation matrices constructed based on the prime field GF($12t+1$) with $t = 28$. Take the first 24 columns of circulant permutation matrices of $\mathbf{H}_1$ to form a 4×24 subarray $\mathbf{H}_1(4, 24)$ of $\mathbf{H}_1$. The code given by the null space of $\mathbf{H}_1(4, 24)$ is an $(8088, 6743)$ QC-LDPC code with rate 0.834 whose performances over the AWGN and binary erasure channels are shown in Figures 5 and 6. Divide $\mathbf{H}_1(4, 24)$ into three 4×8 subarrays, $\mathbf{D}_1^{(1)}$, $\mathbf{D}_2^{(1)}$, and $\mathbf{D}_3^{(1)}$. Disperse these three subarrays of $\mathbf{H}_1(4, 24)$ into three 8×8 arrays of 337×337 circulant permutation and zero matrices, $\mathbf{E}_1^{(1)}$, $\mathbf{E}_2^{(1)}$, and $\mathbf{E}_3^{(1)}$, based on (3). The dispersion of $\mathbf{H}_1(4, 24)$ results in an 8×24 array of 337×337 circulant permutation and zero matrices, $\mathbf{W}^{(1)}(8, 24) = [\mathbf{E}_1^{(1)}\mathbf{E}_2^{(1)}\mathbf{E}_3^{(1)}]$. $\mathbf{W}^{(1)}(8, 24)$ is a 2696×8088 matrix over GF(2) with column and row weights 4 and 12, respectively. The null space of $\mathbf{W}^{(1)}(8, 24)$ gives a $(8088, 5395)$ QC-LDPC code with rate 0.667. The performances of this code over AWGN and binary erasure channels are also shown in Figures 5 and 6, respectively. The code is also capable of correcting any erasure-burst of length at least 1349. We see that the code performs well over AWGN, binary random and burst erasure channels.

In a similar manner, we can disperse subarrays of arrays $\mathbf{H}_2$ constructed based on class-II Bose-BIBD's.

5 Conclusion

In this paper, we have presented two new methods for constructing QC-LDPC codes based on two classes of Bose-BIBD's that are constructed from prime fields. The Tanner graphs of codes in these classes have girth at least 6. Some example codes of various lengths were given. Experimental results showed that the example codes perform very well over the AWGN and binary erasure channels. One class of codes is also capable of correcting bursts of erasures. The proposed methods can be used to construct QC-LDPC codes from BIBD's with $\lambda = 1$ that are constructed based on finite fields.

References

1. R. C. Bose, "On the construction of balanced incomplete designs," *Ann. Eugenics* 9, pp. 353-399, 1939.
2. H. B. Mann, "Analysis and Design of Experiments", Dover Publishers, New York, NY, 1949.
3. R. G. Gallager, "Low density parity-check codes," *IRE Trans. Inform. Theory*, IT-8, pp. 21-28, Jan. 1962.
4. B. Ammar, B. Honary, Y. Kou, J. Xu, and S. Lin, "Construction of low density parity-check codes based on balanced imcomplete block designs," *IEEE Trans. Inform. Theory*, vol. 50, vo. 6, pp. 1257-1268, June 2004.
5. R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. 27, pp. 533-547, Sept. 1981.
6. D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, pp. 399-432, Mar. 1999.
7. T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 599-618, Feb., 2001.
8. F. R. Kschinschang, B. J. Frey, and H. -A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, pp. 498-519. Feb. 2001.
9. Z. -W. Li, L. Chen, S. Lin, W. Fong, and P. -S. Yeh, "Efficient encoding of quasi-cyclic low-density parity-check codes," *IEEE Trans. Commun.*, (accepted) 2005.
10. L. Chen, J. Xu, I. Djurdjevic, and S. Lin, "Near Shannon limit quasi-cyclic low-density parity-check codes," *IEEE Trans. Commun.*, vol. 52, no. 7, pp. 1038-1042, July 2004.
11. H. Tang, J. Xu, Y. Kou, S. Lin, and K. Abdel-Ghaffar, "On algebraic construction of Gallager and circulant low density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 50, no. 6, pp 1269-1279, June 2004.
12. L. Chen, I. Djurdjevic, J. Xu, S. Lin, and K. Abdel-Ghaffar, "Construction of QC-LDPC codes based on the minimum-weight codewords of RS codes," *Proc. IEEE Int. Symp. Inform. Theory*, pp. 239, Chicago, IL, June 27-July 2, 2004.
13. L. Chen, L. Lan, I. Djurdjevic, S. Lin, K. Abdel-Ghaffar, "An algebraic method for constructing quasi-cyclic LDPC codes," *Proc. Int. Symp. Inform. Theory and Its Applications*, ISITA2004, pp. 535-539, Parma, Italy, Oct. 10-13, 2004.
14. J. Xu, L. Chen, L. Zeng, L. Lan, and S. Lin, "Construction of low-density parity-check codes by superposition,", *IEEE Trans. Commun.,* vol. 53, no. 2, pp. 243-251, Feb. 2005.
15. Y. Kou, S. Lin, and M. Fossorier, "Low density parity-check codes based on finite geometries: a discovery and new results," *IEEE Trans. Inform. Theory*, vol. 47, pp. 2711-2736, Nov. 2001.

16. S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*, 2nd edition, Prentice Hall, Upper Saddle River, NJ., 2004.
17. R. M. Tanner, "Spectral graphs for quasi-cyclic LDPC codes," *Proc. IEEE int. Symp. on Inform. Theory*, p. 226, Washington D.C., June 24-29, 2001.
18. C. Di, D. Proietti, I. E. Teletar, T. j. Richarson, and R. L. Urbanke, "Finite length analysis of low-density parity-check codes on binary erasure channel," *IEEE Trans. Inform. Theory*, vol. 48, no. 6, pp. 1570-1579, June 2002.
19. A. Orlitsky, R. L. Urbanke, K. Viswanathan, and J. Zhang, "Stopping sets and the girth of Tanner graph," *Proc. Int. Symp. Inform. Theory*, p.2, Lausanne, Switzerland, June 30-July 5, 2002.
20. I. Djurdjevic, J. Xu, K. Abdel-Ghaffar, and S. Lin, "Construction of low-density parity-check codes based on shortened Reed-Solomon codes with two information symbols," *IEEE Commun. Lett.,* no. 7, pp. 317-319, July 2003.
21. T. Richardson, "Error floors of LDPC codes,"*Proc. Allerton Conf. on Communication, Control and computing,* pp. 1426-1435, Monticello, IL., Oct. 2003.

Long Extended BCH Codes Are Spanned by Minimum Weight Words

Tali Kaufman[1] and Simon Litsyn[2]

[1] MIT Computer Science and Artificial Intelligence Laboratory (CSAIL),
32 Vassar Street, Cambridge, MA 02139, USA
kaufmant@mit.edu

[2] Department of Electrical Engineering-Systems, Tel Aviv University,
Tel Aviv 69978, Israel
litsyn@eng.tau.ac.il

Abstract. It is shown that long enough extended t-error correcting BCH codes $\mathcal{B}_t$ are spanned by its lightest words of weight $2t+2$. The proof follows from an upper bound on the number of words of weight $2t+2$ in any subcode of $\mathcal{B}_t$ of codimension 1.

1 Introduction

The (binary, primitive, narrow-sense) BCH codes are among the most studied algebraic codes. It is not always simple to determine their combinatorial parameters such as minimum distance, covering radius, and distance distribution. However, when the length of the considered codes is large enough these parameters "stabilize", and exact results can be derived. For instance, it was shown by Farr (see [23]) that the actual minimum distance of long t-error correcting BCH codes, $\mathcal{C}_t$, equals to its designed distance, $2t+1$. Skorobogatov and Vladuts [27] (see also [5, 22, 30, 31]) proved that the covering radius of long codes $\mathcal{C}_t$ is exactly $2t-1$. As for the distance distribution of BCH codes, it was proved by Sidel'nikov [26] (see also [7, 14, 16, 18, 20, 28, 32]) that when the length of the code grows the distance distribution of $\mathcal{C}_t$ converges to a normalized binomial distribution.

The question whether a linear code has a basis consisting of minimum weight words has been addressed for different classes of codes. For instance, it was central in considerations of [33] where such a property was related to existence of a minimal-change ordering of code. In [9] the codes having a minimum-weight basis were called "high-visible". In [3] the defined property was a necessary condition for existence of a minimum possible path for a traveling salesman on the Hamming cube. Some applications to data compression were discussed in [3]. In [17] it was shown that low-weight trellis-based iterative soft-decision decoding algorithms are very efficient for codes that are spanned by their minimum weight words.

The question of existence of a minimum-weight basis has got a special attention in the case of Reed-Muller codes and their generalizations [6, 8, 10, 11, 13, 15].

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 285–294, 2006.

However, in the case of BCH codes the problem has not been addressed earlier. The only result we are aware of is due to Augot, Charpin and Sendrier [1], who showed that in the BCH codes of length $2^m - 1$ and minimum distance $2^{m-2} - 1$, there is no minimum-weight basis.

In this paper we study extended (binary, primitive, narrow-sense) BCH codes, $\mathcal{B}_t$, having length $n = 2^m$ and minimum distance $2t + 2$. The main result is that long enough codes $\mathcal{B}_t$ have a basis consisting only of words of weight $2t + 2$. The proof is based on checking a condition on the weight distribution of a linear subcode of the code. Namely, we demonstrate that in any subcode of codimension 1 the number of codewords of weight $2t + 2$ is always less than the one in the ambient code. The proof involves a combination of a Johnson-like and Karamata inequalities, and some properties of Krawtchouk polynomials.

2 Preliminaries and Useful Coding Tools

Let $F = \{0, 1\}$, and x be a vector in F^n of weight $w(x)$. For a *linear* code $C \subseteq F^n$ and $v \in F^n$ such that $v \notin C$, the *v-coset* of C is

$$C + v \overset{\text{def}}{=} \{c + v | c \in C\}.$$

Note that $|C + v| = |C|$. Let

$$C \cup v \overset{\text{def}}{=} C \cup (C + v) = \{c | c \in C \vee c \in C + v\},$$

$|C \cup v| = 2|C|$. The *covering radius* of C, $R(C)$, is the maximum distance from the code to a vector in the ambient space [4]. The following basic facts (see e.g. [23]) will be used throughout the paper.

For a linear code $C \subseteq F^n$ the *distance distribution (spectrum) of* C,

$$B^C = (B_0^C, B_1^C, \cdots, B_n^C),$$

is defined as follows:

$$B_i^C = |\{c \in C | w(c) = i\}|, \quad i = 0, 1, \ldots, n.$$

It coincides with the weight distribution. For a coset $[C + v]$ we define its weight distribution as

$$B_i^{[C+v]} = |\{c \in [C + v] | w(c) = i\}|, \quad i = 0, 1, \ldots, n.$$

Claim. [Johnson Bound] For any $v \in F^n$ and a code C of length n and distance d,

$$B_i^{[C+v]} \leq \frac{dn}{2i^2 - 2ni + dn} \ . \qquad \square$$

The distance distribution $B^{C^\perp}$ of the dual code $C^\perp$ is called the *dual spectrum* of C. It is uniquely determined by B^C.

Claim. [MacWilliams Transform] For a linear binary code C of length n,

$$B_j^{C^\perp} = \frac{1}{|C|}\sum_{i=0}^{n} B_i^C P_j(i), \quad j = 0, 1, \ldots, n,$$

where

$$P_j(i) = \sum_{\ell=0}^{j}(-1)^\ell \binom{i}{\ell}\binom{n-i}{j-\ell}$$

is the *Krawtchouk polynomial* of degree j. □

For a coset $[C+v]$ we formally define its MacWilliams transform,

$$B_j^{[C+v]^\perp} \overset{\text{def}}{=} \frac{1}{|C|}\sum_{i=0}^{n} B_i^{[C+v]} P_j(i), \quad j = 0, 1, \ldots, n.$$

For properties of the Krawtchouk polynomials see e.g. survey [21]. Following is a useful simple bound on values of Krawtchouk polynomials [21].

Claim. For $k = 0, 1, \ldots, n$,

$$P_k(i) \leq \frac{|n-2i|^k}{k!}. \qquad \square$$

Extended (binary primitive narrow-sense) BCH codes, $\mathcal{B}_t$, have length $n = 2^m$ and minimum distance at least $2t+2$. The possible non-zero distances in $\mathcal{B}_t^\perp$ are restricted by the following result.

Claim. [A.Weil-Carlitz-Uchiyama Bound] For $c \in \mathcal{B}_t^\perp$, $c \notin \{0^n, 1^n\}$:

$$\frac{n}{2} - (t-1)\sqrt{n} \leq w(c) \leq \frac{n}{2} + (t-1)\sqrt{n}.$$

Moreover,

$$B_0^{\mathcal{B}^\perp} = B_n^{\mathcal{B}^\perp} = 1,$$
$$B_i^{\mathcal{B}^\perp} = B_{n-i}^{\mathcal{B}^\perp}, \quad i = 1, \ldots, n-1. \qquad \square$$

Clearly this bound holds for the minimum distance $d_t^\perp$ of $\mathcal{B}_t^\perp$,

$$d_t^\perp \geq \frac{n}{2} - (t-1)\sqrt{n}. \tag{1}$$

Finally, we will be using the following result from [20].

Claim. In $\mathcal{B}_t$ the number of codewords of weight $2i$, $B_{2i}^{\mathcal{B}}$, is

$$B_{2i}^{\mathcal{B}} = \frac{\binom{n}{2i}}{2n^t}\left(1 + O\left(\frac{1}{n}\right)\right). \qquad \square$$

Throughout we assume that t is a constant while n is large enough.

3 Coset Property and a Minimum-Weight Basis for Codes

In the following we provide a sufficient condition for existence of a minimum-weight basis for a linear code.

Definition 1. *A linear code C of length n has* Coset-Property$(k, \delta(\epsilon, k))$ *if for every ϵ,*

$$\frac{1}{n} \leq \epsilon \leq \frac{R(C)}{n},$$

there exists a non-decreasing (in ϵ) $\delta = \delta(\epsilon, k) > 0$, such that for every vector $v_{\epsilon n}$ being at distance ϵn from C,

$$B_k^{[C+v \quad]^{\perp}} \leq (1-2\delta) B_k^{C^{\perp}}. \qquad \square$$

Indeed for a vector $v_{\epsilon n} \notin C$, its distance from C is between 1 to $R(C)$. Thus, ϵ is chosen to cover all possible distances of $v_{\epsilon n}$ from C.

Note that the condition

$$B_k^{[C+v \quad]^{\perp}} \leq (1-2\delta) B_k^{C^{\perp}}$$

implies

$$B_k^{[C \cup v \quad]^{\perp}} \leq (1-\delta) B_k^{C^{\perp}}.$$

Indeed,

$$\begin{aligned} B_k^{[C \cup v \quad]^{\perp}} &= \frac{1}{|C \cup v_{\epsilon n}|} \sum_{i=0}^{n} B_i^{C \cup v \quad} P_k(i) \\ &= \frac{1}{2|C|} \sum_{i=0}^{n} B_i^{C} P_k(i) + \frac{1}{2|C|} \sum_{i=0}^{n} B_i^{C+v \quad} P_k(i) \\ &= \frac{B_k^{C^{\perp}}}{2} + \frac{B_k^{[C+v \quad]^{\perp}}}{2}. \end{aligned}$$

Next we show that the coset-property of C implies that $C^{\perp}$ is spanned by minimum-weight words.

Theorem 1. *If C has* Coset-Property$(k, \delta(\epsilon, k))$ *then $C^{\perp}$ is spanned by words of weight k.*

Proof. Assume that the weight k codewords $C_k^{\perp}$ do not span $C^{\perp}$. Then, there exists C^* that contains $C_k^{\perp}$ and perhaps some other codewords from $C^{\perp}$, such that its size is half of the size of $C^{\perp}$. Hence, we conclude that $C^{*\perp}$, that is dual to C^*, is $C^{*\perp} = C \cup v_{\epsilon n}$, where $v_{\epsilon n} \in C^{\perp}$. However, due to Coset-Property$(k, \delta(\epsilon, k))$ we know that the number of words of weight k in $C^* = [C \cup v_{\epsilon n}]^{\perp}$ is at most $(1-\delta) B_k^{C^{\perp}}$, and this contradicts to the fact that C^* contains $B_k^{C^{\perp}}$ words of weight k. Hence, $C^{\perp}$ is spanned by its weight k words.

The main result of the paper is the following theorem.

Theorem 2. *For a constant $t \geq 1$ and n large enough, $\mathcal{B}_t$ is spanned by its words of weight $2t+2$.* □

To prove it we show that $\mathcal{B}_t^{\perp}$ has the Coset-Property$(2t+2, \epsilon)$, i.e. we chose $\delta(\varepsilon, 2t+2) = \varepsilon$. The idea we use is as follows. Note that the Coset-Property$(2t+2, \epsilon)$ for $\mathcal{B}_t^{\perp}$ is obtained if for any $\epsilon \in \left[\frac{1}{n}, R(\mathcal{B}_t^{\perp})\right]$ and any vector $v_{\epsilon n}$ that is at distance ϵn from $\mathcal{B}_t^{\perp}$,

$$B_{2t+2}^{[\mathcal{B}^{\perp}+v \quad]^{\perp}} \leq (1-2\epsilon) B_{2t+2}^{\mathcal{B}}.$$

Using the MacWilliams transform the last can be reformulated as follows:

$$\sum_{i=0}^{n} B_i^{\mathcal{B}^{\perp}+v} \quad P_{2t+2}(i) \leq (1-2\epsilon) \sum_{i=0}^{n} B_i^{\mathcal{B}^{\perp}} P_{2t+2}(i).$$

Recalling that $B^{\mathcal{B}^{\perp}}$, $B^{\mathcal{B}^{\perp}+v}$ and $P_{2t+2}(i)$ are symmetric around $\frac{n}{2}$, we conclude that the last inequality can be deduced from the following one:

$$\sum_{i=\epsilon n}^{n/2} B_i^{\mathcal{B}^{\perp}+v} \quad P_{2t+2}(i) \leq (1-2\epsilon) \left(\binom{n}{2t+2} + \sum_{i=d^{\perp}}^{n/2-1} B_i^{\mathcal{B}^{\perp}} P_{2t+2}(i) \right).$$

The intuition justifying validity of the last claim is as follows. Since all non-zero and non-one codewords of $\mathcal{B}_t^{\perp}$, are of weight close to $\frac{n}{2}$, and as the corresponding Krawtchouk polynomial has small values around $\frac{n}{2}$, we conclude that the term $\binom{n}{2t+2}$ corresponding to the zero word contributes significantly to the summation in the right-hand side, while other codewords have negligible effect. The zero word does not appear in the coset of $\mathcal{B}_t^{\perp} + v_{\epsilon n}$ since the minimum weight of $\mathcal{B}_t^{\perp} + v_{\epsilon n}$ is ϵn. Since the rest of codewords do not contribute much to the summation we obtain the desired property.

In order to bound the left-hand side expression we use the following strategy. We partition it into two sums that we address as **head** (first sum) and **tail** (second sum),

$$\sum_{i=\epsilon n}^{\frac{}{2}} B_i^{\mathcal{B}^{\perp}+v} \quad P_{2t+2}(i) = \sum_{i=\epsilon n}^{\frac{}{2}-a} B_i^{\mathcal{B}^{\perp}+v} \quad P_{2t+2}(i) + \sum_{i=\frac{}{2}-a}^{\frac{}{2}} B_i^{\mathcal{B}^{\perp}+v} \quad P_{2t+2}(i).$$

Here a, to be defined later in the proof, is close to $\frac{n}{2} - \sqrt{n}$. Hence, we will conclude that the contribution of the tail is negligible. To deal with the head we use some generalization of the Johnson bound, to estimate $\sum_{i=\epsilon n}^{\frac{}{2}-a} B_i^{\mathcal{B}^{\perp}+v}$. Then we use the Karamata inequality to deduce a bound on the head. Note that Karamata inequality was used earlier for derivation of an upper bound on code's size as a function of minimum distance by Tietäväinen [29]. In what follows we consider the case $t > 1$. Note that for $t = 1$, it is known that the extended Hamming codes (= Reed-Muller codes of length 2^m and order $m-2$) are spanned by its minimum weight vectors [23].

4 Proof of Theorem 2

In this section we provide a proof to the central result.

4.1 Two Useful Lemmas

Lemma 1. [Karamata Inequality, see [24]]: *Let* $x_1, \cdots, x_s$*,* $y_1, \cdots y_s$ *be two non-negative sequences, such that for every* $q < s$*,*

$$\sum_{i=1}^{q} x_i \leq \sum_{i=1}^{q} y_i,$$

and such that

$$\sum_{i=1}^{s} x_i = \sum_{i=1}^{s} y_i.$$

If $f(\cdot)$ *is a convex function then,*

$$\sum_{i=1}^{s} f(x_i) \leq \sum_{i=1}^{s} f(y_i). \qquad \square$$

Lemma 2. [A generalized Johnson Bound]: *Let*

$$a = n^{\frac{1}{2}+\frac{1}{4(2\ +3)}},$$

and let c_i *for* $i = 1, \cdots, 2n^t$*, be the codewords of* $\mathcal{B}_t^{\perp} + v_{\epsilon n}$*. Denote* $x_i = n - 2w(c_i)$ *and assume that the order of the codewords is such* $x_1 \geq x_2 \geq \cdots \geq x_{2n}$ *. Denote*

$$Q = \sum_{\epsilon n}^{\frac{}{2}-a} B_i^{\mathcal{B}^{\perp}+v} \quad .$$

Then

$$Q \leq n^{1-\frac{1}{(2\ +2)}}(1+o(1)).$$

Moreover,

for $s \leq Q$*, such that* $s \leq n^{\frac{1}{2}-\frac{1}{100}}$*,*

$$\sum_{i=1}^{s} x_i \leq n\sqrt{s}(1+o(1)),$$

for $s \leq Q$*, such that* $s \geq n^{\frac{1}{2}+\frac{1}{100}}$*,*

$$\sum_{i=1}^{s} x_i \leq n^{\frac{3}{4}} s(1+o(1)),$$

for $s \leq Q$*, such that* $n^{\frac{1}{2}-\frac{1}{100}} \leq s \leq n^{\frac{1}{2}+\frac{1}{100}}$*,*

$$\sum_{i=1}^{s} x_i \leq n^{1+\frac{1}{100}}\sqrt{s}(1+o(1)).$$

4.2 Sketch of the Proof of Theorem 2

Consider $\mathcal{B}_t^{\perp}$, $t > 1$, $\epsilon > 0$. Recall that by Theorem 1 it is sufficient to show that $\mathcal{B}_t^{\perp}$ has the Coset-Property($2t + 2$,ϵ).

We next show that for a vector $v_{\epsilon n}$ that is at distance ϵn from $\mathcal{B}_t^{\perp}$,

$$B_{2t+2}^{[\mathcal{B}^{\perp}+v\quad]^{\perp}} \leq (1-2\epsilon)B_{2t+2}^{\mathcal{B}}.$$

As we showed it is sufficient to show that:

$$\sum_{i=\epsilon n}^{\frac{}{2}} B_i^{\mathcal{B}^{\perp}+v}\quad P_{2t+2}(i)$$

$$\leq (1-2\epsilon)\sum_{i=\epsilon n}^{\frac{}{2}-a} B_i^{\mathcal{B}^{\perp}+v}\quad P_{2t+2}(i) + (1-2\epsilon)\sum_{i=\frac{}{2}-a}^{\frac{}{2}} B_i^{\mathcal{B}^{\perp}+v}\quad P_{2t+2}(i).$$

In the following we bound the left-hand side. Choose $a = n^{\frac{1}{2}+\frac{1}{(2\ +2)}}$. Note that $a^{2t+2} = o(n^{t+2})$. By the bound from Claim 2, and by the bound on the number of codewords in $\mathcal{B}_t^{\perp}$ the following is true:

$$\sum_{i=\epsilon n}^{\frac{}{2}} B_i^{\mathcal{B}^{\perp}+v}\quad P_{2t+2}(i)$$

$$\leq \sum_{i=\epsilon n}^{\frac{}{2}-a} B_i^{\mathcal{B}^{\perp}+v}\quad P_{2t+2}(i) + (n+1)^t \max_{\frac{}{2}-a<i\leq\frac{}{2}}(P_{2t+2}(i))$$

$$\leq \frac{1}{(2t+2)!}\left(\sum_{i=\epsilon n}^{\frac{}{2}-a} B_i^{\mathcal{B}^{\perp}+v}\quad (n-2i)^{2t+2} + (2a)^{2t+2}(n+1)^t\right). \qquad (2)$$

In the following we present two properties, whose validity is sufficient for the proving the theorem.

Head Bound:

$$\sum_{i=\epsilon n}^{\frac{}{2}-a} B_i^{\mathcal{B}^{\perp}+v}\quad (n-2i)^{2t+2} \leq (1-3\epsilon)n^{2t+2} + O(n^{2t+1}).$$

Tail Bound:

$$(2a)^{2t+2}\cdot(n+1)^t \leq \frac{1}{2}(\epsilon n^{2t+2} + O(n^{2t+1})).$$

We will provide different proofs for different ranges of values of ϵ. We deal with three cases:

Case 1: $\frac{1}{4} \leq \epsilon \leq \frac{1}{2}$.

Case 2: $\frac{1}{n^{\frac{1}{2}-\frac{1}{4(2\ +2)}}} \leq \epsilon \leq \frac{1}{4}$.

Case 3: $\frac{1}{n} \leq \epsilon \leq \frac{1}{n^{\frac{1}{2}-\frac{1}{4(2\ +2)}}}$.

Note that by the bounds on the covering radius of the code (see e.g. [4]), ϵ is at most $\frac{1}{2}$. Hence, the three cases cover the possible values of ϵ. The first two cases are proved along the same lines. The proof is based on the Karamata inequality presented in Lemma 1. In either of these cases we shall show that the **Head Bound** and **Tail Bound** apply. In the proof of the third case we use a different strategy based on analysis of the relevant values of Krawtchouk polynomials.

In the proof of the first case we use the fact that by the Johnson bound the number of words of weight exactly ϵn in $\mathcal{B}_t^{\perp} + v_{\epsilon n}$ is at most

$$\frac{dn}{2(\epsilon n)^2 - 2n(\epsilon n) + dn} \leq \frac{1}{(1-2\epsilon)^2}(1 + o(1)).$$

In the second case we use the fact that there can be only one word c_i in $\mathcal{B}_t^{\perp} + v_{\epsilon n}$ of weight ϵn. The rest of the words of $\mathcal{B}_t^{\perp} + v_{\epsilon n}$ are of weight at least $(\frac{1}{2} - \epsilon)n$. By the Johnson bound the number of words of weight $(\frac{1}{2} - \epsilon)n$ in this code is at most

$$\frac{dn}{2(\frac{1}{2}-\epsilon)^2 - 2n(\frac{1}{2}-\epsilon) + dn} \leq \frac{1}{(2\epsilon)^2}(1 + o(1)).$$

The complete proofs will be published elsewhere.

5 Conclusion

We proved that long enough extended BCH codes are spanned by a subset of their minimum weight codewords. It is still an open question if the same claim is true for non-extended codes. Though we believe this is the case we were not able to prove it. As well it would be interesting to obtain estimates on the minimum length from which Theorem 2 is true. The example from [1] demonstrates that this result cannot be correct for all lengths. It is easy to see that in our arguments we used only knowledge of the minimum distance, dual minimum distance, and the size of codes. Therefore, our consideration can be easily extended to other codes with similar to BCH parameters, e.g. Goppa codes. It would be interesting to further expand the list of codes for which there is a minimum-weight basis.

Acknowledgment

We are grateful to Noga Alon, Oded Goldreich, Michael Krivelevich, Dana Ron, Amir Shpilka, Madhu Sudan, and Sergei Yekhanin for constructive comments and discussions.

References

1. D. Augot, P. Charpin and N. Sendrier, Studying the locator polynomials of minimum weight codewords of BCH codes, *IEEE Trans. Inform. Theory*, vol. 38, 3, 1992, pp.960–973.
2. E. R. Berlekamp, *Algebraic Coding Theory*, McGraw-Hill, 1968.
3. G. Cohen, S. Litsyn and G. Zémor, On the traveling salesman problem in binary Hamming spaces, *IEEE Trans. Inform. Theory*, vol. 42, 4, 1996, pp.1274–1276.
4. G. Cohen, I. Honkala, S. Litsyn and A. Lobstein, *Covering Codes,* Elsevier, 1997.
5. S. D. Cohen, The length of primitive BCH codes with minimal covering radius, *Designs, Codes, and Cryptography*, vol. 10, 1, 1997, pp.5–16.
6. Ph. Delsarte, J. M. Goethals and F. J. MacWilliams, On generalized Reed-Muller codes and their relatives, *Inform. and Control*, vol.16, 1970, pp.403–442.
7. Y. Desaki, T. Fujiwara and T. Kasami, The weight distributions of extended binary primitive BCH codes of length 128, *IEEE Trans. Inform. Theory*, vol. 43, 4, 1997, pp.1364–1371.
8. P. Ding and J. D. Key, Minimum-weight codewords as generators of generalized Reed-Muller codes, *IEEE Trans. Inform. Theory*, vol. 46, 6, 2000, pp.2152–2158.
9. J. Francke and J. J. H. Meijers, Super-visible codes, *Discrete Math.* vol. 110, 1–3, 1992, pp.119–134.
10. S. Gao and J. D. Key, Bases of minimum-weight vectors for codes from designs, *Finite Fields Appl.* vol. 4, 1, 1998), pp.1–15.
11. J. N. Gutie'rrez and H. Tapia-Recillas, A minimum weight basis for the binary Reed-Muller code, *Congr. Numer.* 154 (2002), 195–199.
12. *Handbook of Coding Theory*, North Holland, 1997.
13. C. S. Jutla, A. C. Patthak, A. Rudra and D. Zuckerman, Testing low-degree polynomials over prime fields, in book: *Proc. IEEE Symposium on Foundations of Computer Science*, 2004.
14. T. Kasami, T. Fujiwara, S. Lin, An approximation to the weight distribution of binary linear codes, *IEEE Trans. Inform. Theory*, vol. 31, 6, 1985, pp.769–780.
15. T. Kaufman and D. Ron, Testing polynomials over general fields, in book: *Proc. IEEE Symposium on Foundations of Computer Science*, 2004.
16. O. Keren and S. Litsyn, More on the distance distribution of BCH codes, *IEEE Trans. Inform. Theory*, vol. 45, 1, 1999, pp. 251–255.
17. T. Koumoto, T. Takata, T. Kasami, and S. Lin, A low-weight trellis-based iterative soft-decision decoding algorithm for binary linear block codes, *IEEE Trans. Inform. Theory*, vol.45, 2, 1999, pp.731–741.
18. I. Krasikov and S. Litsyn, On spectra of BCH codes, *IEEE Trans. Inform. Theory,* vol.41, 3, 1995, pp.786–788.
19. I. Krasikov and S. Litsyn, Linear programming bounds for codes of small size, *European J. Combin.*, vol.18, 6, 1997), pp.647–654.
20. I. Krasikov, and S. Litsyn, On the distance distributions of BCH codes and their duals, *Des. Codes Cryptogr.*, vol. 23, 2, 2001, pp.223–231.
21. I. Krasikov and S. Litsyn, Survey of binary Krawtchouk polynomials, in book: *Codes and Association Schemes,* DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol.56, 2001, pp. 199–211.
22. F. Levy-dit-Vehel and S. Litsyn, More on the covering radius of BCH codes, *IEEE Trans. Inform. Theory*, vol. 42, 1996, pp.1023–1028.
23. F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error Correcting Codes.* North Holland, 1977.

24. A. W. Marshall and I. Olkin. *Inequalities: Theory of Majorization and its Applications.* Academic Press, New York, 1979.
25. R. J. McEliece, H. Rumsey,Jr. Sphere-packing in the Hamming metric, *Bull. Amer. Math. Soc.*, vol.75, 1969, pp. 32–34.
26. V. M. Sidel'nikov, The spectrum of weights of binary Bose-Chaudhuri-Hocquenghem codes, *Problemy Peredachi Informatsii*, vol.7, 1, 1971, pp.14–22.
27. A. N. Skorobogatov and S. G. Vladuts, The covering radius of long binary BCH codes, *Problemy Peredachi Informatsii*, vol. 25, 1, 1989, pp.38-45.
28. P. Solé, A limit law on the distance distribution of binary codes, *IEEE Trans. Inform. Theory*, vol. 36, 1990, pp.229–232.
29. A. Tietäväinen, Bounds for binary codes just outside the Plotkin range, *Inform. and Control*, vol. 47, 2, 1980, pp.85–93.
30. A. Tietäväinen, On the covering radius of long binary BCH codes, *Discr. App. Math.*, vol. 16, 1987, pp. 75-77.
31. A. Tietäväinen, An asymptotic bound on the covering radius of binary BCH codes, *IEEE Trans. Inform. Theory*, vol. 36, 1990, pp.211-213.
32. S. Vladuts and A. Skorobogatov, On spectra of subcodes over subfields of algebraic-geometric codes, *Probl. Peredachi Inform.*, vol. 27, 1, 1991, pp.24–36.
33. A. J. van Zanten, Minimal-change order and separability in linear codes, *IEEE Trans. Inform. Theory*, vol. 39, 6, 1993, pp.1988–1989.

On the Feng-Rao Bound for Generalized Hamming Weights

Olav Geil and Christian Thommesen

Aalborg University, 9220 Aalborg Øst, Denmark
olav@math.aau.dk, cthom@math.aau.dk

Abstract. The Feng-Rao bound gives good estimates of the minimum distance of a large class of codes. In this work we are concerned with the problem of how to extend the Feng-Rao bound so that it deals with all the generalized Hamming weights. The problem was solved by Heijnen and Pellikaan in [7] for a large family of codes that includes the duals of one-point geometric Goppa codes and the q-ary Reed-Muller codes, but not the Feng-Rao improved such ones. We show that Heijnen and Pellikaan's results holds for the more general class of codes for which the traditional Feng-Rao bound can be applied. We also establish the connection to the Shibuya-Sakaniwa bound for generalized Hamming weights ([15], [16], [17], [18], [19] and [20]). More precisely we show that the Shibuya-Sakaniwa bound is a consequence of the extended Feng-Rao bound. In particular the extended Feng-Rao bound gives always at least as good estimates as does the Shibuya-Sakaniwa bound.

1 Introduction

In [3] and [4] Feng and Rao showed how to estimate the minimum distance of a large class of algebraically defined codes by considering certain relations between the rows in the corresponding parity check matrices. This result is known today as the Feng-Rao bound. Using the bound Feng and Rao were able to improve a large class of well-known codes by leaving out certain rows in the corresponding parity check matrices. Since the emergence of the Feng-Rao bound quite a lot of research has been done on the subject. In the present paper we will present a new point of view on how to extend the Feng-Rao bound so that it also deals with generalized Hamming weights. This in particular will allow us to establish the connection between various results in the literature.

The literature gives several interpretations of the Feng-Rao bound. In [14] and [9] Kirfel and Pellikaan introduced the concept of an error-correcting array. Using this concept they reformulated the Feng-Rao bound for a large class of codes that includes the duals of one-point geometric Goppa codes, the q-ary Reed-Muller codes and the cyclic codes. Another interpretation was given by Høholdt, van Lint and Pellikaan in [8]. Here they introduced the concept of an order function acting on what is known today as an order domain ([6]). They reformulated some of the most important results by Feng and Rao into this new setting. The code constructions described by Høholdt et al. includes the set of

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 295–306, 2006.

duals of one-point geometric Goppa codes, the set of Feng-Rao improved such ones, the set of q-ary Reed-Muller codes and the set of Feng-Rao improved such ones (the hyperbolic codes). In the PhD thesis [11] and the papers [12] and [13] Miura independently took on more or less the same point of view as done by Høholdt et. al. Furthermore Miura showed how to interpret the Feng-Rao bound into the setting of any linear code over $\mathbb{F}_q$ defined by means of its paritycheck matrix. This point of view was taken a little further by Matsumoto and Miura in [10]. The work by Matsumoto and Miura is very much related to the work by Kirfel and Pellikaan. Matsumoto and Miura's formulation of the Feng-Rao bound is the most general version of all previous proposed interpretations.

In [7] Heijnen and Pellikaan showed how to derive the generalized Hamming weights of a family of codes related to order domains. This family of codes consists of the duals of one-point geometric Goppa codes, the q-ary Reed-Muller codes and a large class of codes defined from order domains of transcendence degree more than one. However, it was not described in [7] how to deal with the Feng-Rao improved codes. In the series of papers [15], [16], [17], [18], [19], [20], Shibuya, Sakaniwa et. al derived a bound on the generalized Hamming weights of linear codes defined by means of their parity check matrices. We will refer to this bound as the *Shibuya-Sakaniwa bound*. In the first paper they consider only affine variety codes, but in the later papers their results are generalized into the setting of any linear codes using the concepts introduced by Miura in [11] and [12] and using to some extend the concepts introduced by Matsumoto and Miura in [10]. The very fact that Shibuya, Sakaniwa et. al use the concept introduced by Matsumoto and Miura indicates that there should be a strong connection between the Shibuya-Sakaniwa bound and the Feng-Rao bound. This connection is to some extent investigated in the work by Shibuya, Sakaniwa et. al, but it is left as an open problem to establish the precise and general connection ([18, p. 1094], [20, p. 3141]). In the present paper we suggest an extension of the Feng-Rao bound so that it deals with the generalized Hamming weights of any linear codes defined by means of their paritycheck matrices. From our bound it is clear what is the connection between the work by Heijnen, Pellikaan by Matsumoto, Miura and by Shibuya, Sakaniwa et. al. Our bound can be viewed as a generalization and to some extend improvement of all the above bounds.

2 The New Bound

Generalized Hamming weights were introduced by Wei in [21] for cryptographically purposes. We start this section by reminding the reader of their definition. Recall that the support of a set S, $S \subseteq \mathbb{F}_q^n$ is defined by

$$\mathrm{Supp}(S) := \{i \mid c_i \neq 0 \ \text{ for some } \ \boldsymbol{c} = (c_1, \ldots, c_n) \in S\}.$$

The tth generalized Hamming weight of a code C is defined by

$$d_t(C) := \min\{\#Supp(S) \mid S \text{ is a linear subcode of } \ C \ \text{ of dimension } \ t\}.$$

Clearly $d_1(C)$ is just the well-known minimum distance. Consider the following definition of a linear code.

Definition 1. *Let $B = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n\}$ be a basis for $\mathbb{F}_q^n$ and let $G \subseteq B$. We define the $\#G$ dimensional code $C(B,G)$ by $C(B,G) := span_{\mathbb{F}}\ \{\boldsymbol{b} \mid \boldsymbol{b} \in G\}$. The dual code (of dimension $n - \#G$) is denoted $C^{\perp}(B,G)$. If $\mathbb{F}_{q'}$ is a subfield of $\mathbb{F}_q$ then the corresponding subfield-subcode of $C^{\perp}(B,G)$ is denoted $C_{q'}^{\perp}(B,G)$*

We next introduce a number of concepts that play a central role in the following.

Definition 2. *For $\boldsymbol{u} = (u_1, \ldots, u_n), \boldsymbol{v} = (v_1, \ldots, v_n) \in \mathbb{F}_q^n$ define the component-wise (or Schur or Hadamard) product $\boldsymbol{u} * \boldsymbol{v} := (u_1 v_1, \ldots, u_n v_n)$. Consider the basis $B = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n\}$ for $\mathbb{F}_q^n$ and define $\boldsymbol{b}_0 := \boldsymbol{0} \in \mathbb{F}_q^n$. Define $L_{-1} := \emptyset$ and $L_l := span_{\mathbb{F}}\ \{\boldsymbol{b}_0, \boldsymbol{b}_1, \ldots, \boldsymbol{b}_l\}$ for $l = 0, \ldots, n$.*

We have a chain of spaces $L_0 \subsetneq L_1 \subsetneq \cdots \subsetneq L_{n-1} \subsetneq L_n = \mathbb{F}_q^n$ and $\dim(L_i) = i$ holds for $i = 0, 1, \ldots, n$. Hence, the following definition makes sense.

Definition 3. *Define $\bar{\rho} : \mathbb{F}_q^n \to \{0, 1, \ldots, n\}$ by $\bar{\rho}(\boldsymbol{v}) = l$ if $\boldsymbol{v} \in L_l \backslash L_{l-1}$.*

The concept of a well-behaving ordered pair plays a central role in this paper. We recall this concept and introduce a new concept called *one-way well-behaving.*

Definition 4. *Consider two bases $B = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n\}$ and $B' = \{\boldsymbol{b}'_1, \ldots \boldsymbol{b}'_n\}$ for $\mathbb{F}_q^n$ (we may or may not have $B = B'$). Let $I := \{1, 2, \ldots, n\}$. An ordered pair $(i,j) \in I^2$ is said to be well-behaving (WB) if $\bar{\rho}(\boldsymbol{b}_u * \boldsymbol{b}'_v) < \bar{\rho}(\boldsymbol{b}_i * \boldsymbol{b}'_j)$ for all u and v with $1 \leq u \leq i, 1 \leq v \leq j$ and $(u,v) \neq (i,j)$. Less restrictive an ordered pair $(i,j) \in I^2$ is said to be one-way well-behaving (OWB) if $\bar{\rho}(\boldsymbol{b}_u * \boldsymbol{b}'_j) < \bar{\rho}(\boldsymbol{b}_i * \boldsymbol{b}'_j)$ for $u < i$.*

In the literature (e.g. [10] and [9]) one also finds the concept of weakly well-behaving (WWB). This concept can be interpreted as follows. An ordered pair (i,j) is said to be WWB if both (i,j) and (j,i) are OWB. Clearly, WB implies OWB and also WWB implies OWB. The results in the present paper are all stated using the concept of OWB. As a consequence of the above observations all results holds if OWB is replaced by either WB or WWB.

Definition 5. *Given bases B, B' as above consider for $l = 1, 2, \ldots, n$ the following sets*

$$V_l := \{i \in I \mid \bar{\rho}(\boldsymbol{b}_i * \boldsymbol{b}'_j) = l \ \ for\ some\ \ \boldsymbol{b}'_j \in B'\ with\ \ (i,j)\ OWB\ \} \qquad (1)$$

$$\Lambda_i := \{l \in I \mid \bar{\rho}(\boldsymbol{b}_i * \boldsymbol{b}'_j) = l\ for\ some\ \boldsymbol{b}'_j \in B'\ with\ \ (i,j)\ OWB\} \qquad (2)$$

Definition 6. *For $\{l_1, \ldots, l_t\} \subseteq I$ and $\{i_1, \ldots, i_t\} \subseteq I$ define*

$$\bar{\mu}(l_1, \ldots l_t) := \#\left((\cup_{s=1,\ldots,t} V_l\ ) \cup \{l_1, \ldots, l_t\}\right) \qquad (3)$$

$$\bar{\sigma}(i_1, \ldots i_t) := \#\left((\cup_{s=1,\ldots,t} \Lambda_i\ ) \cup \{i_1, \ldots, i_t\}\right) \qquad (4)$$

Our main result is (5) below.

Theorem 1. *Let $G \subseteq B$ be fixed. For $1 \leq t \leq \#G$ respectively $1 \leq t \leq n - \#G$ we have*

$$\begin{aligned}&d_t(C(B,G)) \geq \min\{\bar{\sigma}(a_1,\ldots,a_t) \mid a_i \neq a_j \textit{ for } i \neq j \textit{ and } \{\boldsymbol{b}_{a_1},\ldots,\boldsymbol{b}_{a}\ \} \subseteq G\}\\ &d_t(C^{\perp}(B,G)) \geq\\ &\qquad \min\{\bar{\mu}(a_1,\ldots,a_t) \mid a_i \neq a_j \textit{ for } i \neq j \textit{ and } \{\boldsymbol{b}_{a_1},\ldots,\boldsymbol{b}_{a}\ \} \subseteq B\backslash G\}.\end{aligned} \tag{5}$$

Given a subfield $\mathbb{F}_{q'}$ of $\mathbb{F}_q$ the bound (5) also holds if for t, $1 \leq t \leq \dim(C_{q'}^{\perp}(B,G))$ one replaces $d_t(C^{\perp}(B,G))$ with $d_t(C_{q'}^{\perp}(B,G))$.

The result concerning $C(B,G)$ is from [1]. The results concerning the codes $C^{\perp}(B,G))$ and $C_{q'}^{\perp}(B,G)$ are new, but are very much related to the Shibuya-Sakaniwa bound. We postpone a discussion of these connections to the next section. Obviously the result concerning the code $C_{q'}^{\perp}(B,G)$ follows immediately from (5). For the proof of (5) we will need the following definition and a lemma.

Definition 7. *For any $\boldsymbol{c} \in \mathbb{F}_q^n\backslash\{\boldsymbol{0}\}$ we define $m(\boldsymbol{c})$ to be the unique number m such that $\boldsymbol{c} \in L_{m-1}^{\perp}$ but $\boldsymbol{c} \notin L_m^{\perp}$. In other words*

$$m(\boldsymbol{c}) = \min\{m \mid \boldsymbol{c}\cdot\boldsymbol{b}_m \neq 0, \boldsymbol{c}\cdot\boldsymbol{b}_1 = \cdots = \boldsymbol{c}\cdot\boldsymbol{b}_{m-1} = 0\}.$$

Lemma 1. *Consider $G = \{\boldsymbol{b}_{i_1},\ldots,\boldsymbol{b}_{i}\ \} \subseteq B$. Let S, $S \subseteq C^{\perp}(B,G)$ be a linear space of dimension t. There exist a basis $\{\boldsymbol{c}_1,\ldots,\boldsymbol{c}_t\}$ for S with*

$$m(\boldsymbol{c}_1) < \cdots < m(\boldsymbol{c}_t). \tag{6}$$

We have

$$m(\boldsymbol{c}_i) \in I\backslash\{i_1,\ldots,i_s\}, i = 1,\ldots,t. \tag{7}$$

Proof. We first observe that by the very definition of the function m for any $\boldsymbol{c} \in C^{\perp}(B,G)\backslash\{\boldsymbol{0}\}$ we have $m(\boldsymbol{c}) \in \{1,2,\ldots,n\}\backslash\{i_1,\ldots,i_s\}$. Hence, if a basis exists that satisfies (6) then it will certainly also satisfy (7). Let $\{\boldsymbol{c}_1,\ldots,\boldsymbol{c}_t\}$ be a basis for S. If $m(\boldsymbol{c}_1),\ldots,m(\boldsymbol{c}_t)$ are pairwise different we are through. Assume $m(\boldsymbol{c}_u) = m(\boldsymbol{c}_v) =: m$ for some u, v with $1 \leq u < v \leq t$. Define $\beta_u := \boldsymbol{c}_u \cdot \boldsymbol{b}_m \neq 0$, $\beta_v := \boldsymbol{c}_v \cdot \boldsymbol{b}_m \neq 0$. Consider $\boldsymbol{c}_v' := \beta_v\boldsymbol{c}_u - \beta_u\boldsymbol{c}_v$. As $\boldsymbol{c}_v' \cdot \boldsymbol{r} = \beta_v(\boldsymbol{c}_u \cdot \boldsymbol{r}) - \beta_u(\boldsymbol{c}_v \cdot \boldsymbol{r})$ for any $\boldsymbol{r}$ we conclude that $\boldsymbol{c}_v' \cdot \boldsymbol{b}_i = 0$ for $i = 1,2,\ldots,m$. If we replace $\boldsymbol{c}_v$ with $\boldsymbol{c}_v'$ in the basis $\{\boldsymbol{c}_1,\ldots,\boldsymbol{c}_t\}$ we get a new basis. In particular $\boldsymbol{c}_v' \neq \boldsymbol{0}$ and therefore $m(\boldsymbol{c}_v')$ is well defined. We therefore have $m(\boldsymbol{c}_v') > m$. The lemma now follows by induction.

Proof of (5). Let S, $S \subseteq C^{\perp}(B,G)$ be a space of dimension t. Let $\{\boldsymbol{c}_1,\ldots,\boldsymbol{c}_t\}$ be a basis for S as in Lemma 1. Denote $m_1 := m(\boldsymbol{c}_1),\ldots,m_t := m(\boldsymbol{c}_t)$. Denote $\gamma := \bar{\mu}(m_1,\ldots,m_t)$ and write

$$\begin{aligned}&\{i_1,\ldots,i_\gamma\} =\\ &\cup_{s=1,\ldots,t}\left(\{i \in I \mid \exists \boldsymbol{b}_j' \in B' \text{ with } \bar{\rho}(\boldsymbol{b}_i * \boldsymbol{b}_j') = m_s \text{ and } (i,j) \text{ OWB }\} \cup \{m_s\}\right).\end{aligned}$$

We may assume $i_1 < \ldots < i_\gamma$. Let $1 \leq h \leq \gamma$ and consider any vector

$$\boldsymbol{r}_h = \sum_{v=1}^{h} \alpha_v \boldsymbol{b}_i\ , \alpha_v \in \mathbb{F}_q, \alpha_h \neq 0.$$

If $i_h \in \{m_1, \ldots, m_t\}$ then it follows from the definition of the function m (Definition 7) that $\boldsymbol{r}_h \cdot \boldsymbol{c}_h \neq 0$ and in particular that $\boldsymbol{r}_h * \boldsymbol{c}_h \neq \boldsymbol{0}$. If $i_h \notin \{m_1, \ldots, m_t\}$ then it is because there exists a j and an m_u, $u \in \{1, \ldots, t\}$ such that $\bar{\rho}(\boldsymbol{b}_i \ * \boldsymbol{b}'_j) = m_u$ with (i_h, j) OWB. From the definition of the function m and from the OWB property of (i_h, j) we know that $(\boldsymbol{r}_h * \boldsymbol{b}'_j) \cdot \boldsymbol{c}_{\boldsymbol{u}} \neq 0$. But then $\boldsymbol{r}_h * \boldsymbol{c}_{\boldsymbol{u}} \neq \boldsymbol{0}$ holds again. All together for every $\boldsymbol{r}_h$ there exist a $\boldsymbol{c} \in S$ with $\boldsymbol{r}_h * \boldsymbol{c} \neq \boldsymbol{0}$.

This contradicts that $\#\mathrm{Supp}(S) < \gamma$, $\mathrm{Supp}(S) \subseteq \{1, \ldots, \gamma - 1\}$ say, which is seen by selecting $\boldsymbol{r} = \sum_{v=1}^{\gamma} \beta_v \boldsymbol{b}_i$, $(\beta_1, \ldots, \beta_\gamma) \in \mathbb{F}_q^\gamma \backslash \{\boldsymbol{0}\}$ such that $\mathrm{Supp}(\{\boldsymbol{r}\}) \subseteq \{\gamma, \gamma + 1, \ldots, n\}$ and observe that $\boldsymbol{r} * \boldsymbol{c} = \boldsymbol{0}$ for all $\boldsymbol{c} \in S$. The proof of (5) is complete.

Clearly, Theorem 1 holds in particular for the special case $B = B'$. More or less all known code constructions for which the Feng-Rao bound is known to be interesting corresponds to the case $B = B'$. As an exception, to deal with the cyclic codes we will need two different bases B, B' (see [9, Ex. 2.4] and [20, Sec. 4]). The results in Theorem 1 concerning the codes $C^\perp(B, G)$ and $C_{q'}^\perp(B, G)$ can be generalized to deal not only with two, but with many different bases for $\mathbb{F}_q^n$. In this way one can in particular extend the traditional Feng-Rao bound as stated by Matsumoto and Miura in [10] so that it also deals with generalized Hamming weights. Actually, by the following remark one can generalize even further.

Remark 1. From the proof of (5) it is clear that it is of no significance that B' is a basis for $\mathbb{F}_q^n$ and therefore B' can be any indexed subset of $\mathbb{F}_q^n$.

The use of the OWB concept prior to the WWB concept is already justified by the above remark. We further note that it is possible to give examples where OWB gives better estimates than WWB does.

3 The Connection to the Work by Shibuya, Sakaniwa et al.

In [19] Shibuya and Sakaniwa considered the set-up with only one basis B (that is, the setup in Theorem 1 with $B = B'$). In [19] they were concerned with the WWB property. In [20] Shibuya and Sakaniwa considered the set-up with two bases B, B' but were only concerned with the WB property. As we will show below our bound is at least as good as their bound even if we replace their WB as well as their WWB with OWB.

In the following let B and B' be bases as in the previous section. Assume $G \subseteq B$ and denote $G = \{\boldsymbol{b}_{i_1}, \ldots, \boldsymbol{b}_i \ \}$. The Shibuya-Sakaniwa bound from [19] and [20] can be interpreted as follows (we have replaced their WB respectively WWB with OWB).

Definition 8. *For $T \subseteq \{i_1, \ldots, i_s\}$ let*

$$\Lambda_T := \cup_{i \in T} \Lambda_i \qquad \Lambda_T^* := (I \backslash \{i_1, \ldots, i_s\}) \backslash \Lambda_T$$
$$\eta_t := s - \max\{\#T \mid T \subseteq \{i_1, \ldots, i_s\} \ \textit{ such that } \ \#\Lambda_T^* \geq t\}.$$

Theorem 2. *For $t = 1, \ldots, n - \#G$ we have*

$$d_t\left(C^{\perp}(B, G)\right) \geq \eta_t + t \tag{8}$$

Given a subfield $\mathbb{F}_{q'}$ of $\mathbb{F}_q$ the bound (8) also holds if for t, $1 \leq t \leq \dim\left(C^{\perp}_{q'}(B, G)\right)$ one replaces $d_t(C^{\perp}(B, G))$ with $d_t(C^{\perp}_{q'}(B, G))$.

The connection to the theory in the present paper is easily read of the proof of the following proposition.

Proposition 1. *The bound (5) in Theorem 1 is at least as tight as the Shibuya-Sakaniwa bound (8).*

Proof. From Definition 6 we have

$$\begin{aligned}
&\min_{\in^{1}\ \backslash\{\ _{1}, \overset{\cdots}{\ldots}, \quad \}} \{\bar{\mu}(a_1, \ldots, a_t)\} \\
&= \min_{\in^{1}\ \backslash\{\ _{1}, \overset{\cdots}{\ldots}, \quad \}} \{\#[\cup_{s=1}^{t}\{i \in I \mid \\
&\qquad \bar{\rho}(\boldsymbol{b}_i * \boldsymbol{b}'_j) = a_s \text{ for some } \boldsymbol{b}'_j \in B' \text{ with } (i, j) \text{ OWB}\} \cup \{a_1, \ldots, a_t\}]\} \\
&\geq \min_{\in^{1}\ \backslash\{\ _{1}, \overset{\cdots}{\ldots}, \quad \}} \{\# \cup_{s=1}^{t} \{i \in \{i_1, \ldots, i_s\} \mid \\
&\qquad \bar{\rho}(\boldsymbol{b}_i * \boldsymbol{b}'_j) = a_s \text{ for some } \boldsymbol{b}'_j \in B' \text{ with } (i, j) \text{ OWB}\}\} + t \\
&= s - \max\{\#T \mid T \subseteq \{i_1, \ldots, i_s\} \text{ such that } \#\Lambda^{*}_{T} \geq t\} + t = \eta_t + t
\end{aligned}$$

This concludes the proof of Proposition 1.

In Section 5 below we demonstrate that the bound (5) in Theorem 1 can actually be sharper than the bound (8) in Theorem 2.

Next we will be concerned with the complexity of calculating the two bounds (5) and (8). By k we will denote the dimension of $C^{\perp}(B, G)$. That is, $\#G = n - k$. The bound (8) can be calculated with a worst case complexity of

$$O\left(ki \sum_{i=1}^{n-k} \binom{n-k}{i}\right).$$

At a first glance it seems as if the worst case complexity of the bound (5) is

$$O\left(nt\binom{k}{t}\right). \tag{9}$$

However, due to the generalized Singleton bound $d_t \leq n - k + t$ one need in (5) only consider the a_i's with $\bar{\mu}(a_i) \leq n - k + t$. The number of such a_i's are in general much smaller than k. So the value k in (9) should be replaced with a much smaller value. Hence, for large t combined with small dimensions the new bound (5) is by far the fastest. Whereas, for large t combined with large dimensions the picture is not so clear. For small values of t the estimation of

d_t will be fastest by using the bound (5). Fortunately we may sometimes do without the above calculations. Recall, that a code is called tth rank MDS if the tth generalized Hamming weight attains the generalized Singleton bound. In [20] Shibuya and Sakaniwa gave an easily calculate-able criterion under which the code $C^{\perp}(B, G)$ is guaranteed to be tth rank MDS.

Theorem 3. *Let $G = \{\boldsymbol{b}_{i_1}, \ldots, \boldsymbol{b}_{i}\ \}$ and $I = \{1, 2, \ldots, n\}$ and define*

$$g(B, G) := \max_{i \in \{i_1, \ldots, i\ \}} \{\# \left(I \backslash (\Lambda_i \cup \{i_1, \ldots, i_s\})\right)\}. \tag{10}$$

For t with $g(B, G) + 1 \leq t \leq n - s$ the code $C^{\perp}(B, G)$ is tth rank MDS.

In [20, Sec. 4] Shibuya and Sakaniwa presented a BCH type bound for the generalized Hamming weights of cyclic codes. To establish this bound they considered two bases B and B'. The proof in [20] is not very complicated, however with the bound (5) in hand the proof gets even shorter.

4 Codes from Order Domains

In [7] Heijnen and Pellikaan showed how to estimate the generalized Hamming weights of a family of codes related to order domains. This family consists of the duals of one-point geometric Goppa codes and their generalizations to order domains of transcendence degree more than one, including the q-ary Reed-Muller codes. Heijnen and Pellikaan did not describe how to deal with the Feng-Rao improved codes. In this section we will apply the bound (5) to the case of codes defined from order domains. We will see that Heijnen and Pellikaan's bound can be viewed as a consequence of (5) and as a special case of our new bound. In our presentation we will consider only order functions that are also weight functions. These seems to be the only order functions that are relevant for coding theoretical purposes. From[6] we have the following definition and theorem.

Definition 9. *Let R be an $\mathbb{F}_q$-algebra, let Γ be a subsemigroup of $\mathbb{N}_0^r$ for some r and let $\prec$ be a monomial ordering on $\mathbb{N}_0^r$. A surjective map $\rho : R \rightarrow \Gamma_{-\infty} := \Gamma \cup \{-\infty\}$ that satisfies the following five conditions is said to be a weight function over the order domain R*

(*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) *$\rho(f + g) \preceq \max\{\rho(f), \rho(g)\}$ and equality holds when $\rho(f) \prec \rho(g)$*
(*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 $\rho(f - ag) \prec \rho(g)$*
(*W*.5) *If f and g are nonzero then $\rho(fg) = \rho(f) + \rho(g)$.*

Theorem 4. *Given a weight function 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 $R_\gamma := \{f \in R \mid \rho(f) \preceq \gamma\}$.*

In the following we will assume that a basis $\mathcal{B}$ as above has been chosen.

Definition 10. *Let R be an $\mathbb{F}_q$-algebra. A surjective map $\varphi : R \rightarrow \mathbb{F}_q^n$ 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 $f, g \in R$.*

From [1] we have the following definition.

Definition 11. *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 $\Delta(R, \rho, \varphi) = \{\alpha(1), \alpha(2), \ldots, \alpha(n)\}$.*

We are now in the position that we can describe bases $B = B'$ for $\mathbb{F}_q^n$ for which the bound (5) is very much applicable. From [1] we have.

Theorem 5. *Let $\Delta(R, \rho, \varphi) = \{\alpha(1), \alpha(2), \ldots, \alpha(n)\}$ be as in Definition 11. The set*

$$B := \{\boldsymbol{b}_1 := \varphi(f_{\alpha(1)}), \ldots, \boldsymbol{b}_n := \varphi(f_{\alpha(n)})\} \tag{11}$$

constitutes a basis for $\mathbb{F}_q^n$ as a vector space over $\mathbb{F}_q$. For any $\boldsymbol{c} \in \mathbb{F}_q^n$ there exists a unique ordered set $(\beta_1, \ldots, \beta_n)$, $\beta_i \in \mathbb{F}_q$ such that $\boldsymbol{c} = \varphi\left(\sum_{i=1}^n \beta_i f_{\alpha(i)}\right)$. The function $\bar{\rho} : \mathbb{F}_q^n \rightarrow \{0, 1, \ldots, n\}$ corresponding to B is given by

$$\bar{\rho}(\boldsymbol{c}) = \begin{cases} 0 & \textit{if } \boldsymbol{c} = 0 \\ \max\{i \mid \beta_i \neq 0\} & \textit{otherwise.} \end{cases}$$

The following proposition from [1] helps us dealing with the concept of WB.

Proposition 2. *Let $B = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n\}$ be the basis in (11). If $\alpha(i), \alpha(j), \alpha(l) \in \Delta(R, \rho, \varphi)$ are such that $\rho(f_{\alpha(i)} f_{\alpha(j)}) = \alpha(l)$ then $\bar{\rho}(\boldsymbol{b}_i * \boldsymbol{b}_j) = l$ and $(i, j) \in I^2$ is WB. Consider $\alpha(l) \in \Delta(R, \rho, \varphi)$ and assume $\beta_1, \beta_2 \in \Gamma$ satisfies $\rho(f_{\beta_1} f_{\beta_2}) = \alpha(l)$. Then $\beta_1, \beta_2 \in \Delta(R, \rho, \varphi)$ holds.*

We have motivated the following definition.

Definition 12. *For $\lambda \in \Gamma$ define $N(\lambda) := \{\alpha \in \Gamma \mid \lambda - \alpha \in \Gamma\}$ and $\mu(\lambda) := \#N(\lambda)$ if $N(\lambda)$ is finite and $\mu(\lambda) := \infty$ if not. In larger generality consider $\{\lambda_1, \ldots, \lambda_t\} \subseteq \Gamma$ and define $N(\lambda_1, \ldots, \lambda_t) := \cup_{s=1}^t N(\lambda_s)$. Define $\mu(\lambda_1, \ldots, \lambda_t) := \#N(\lambda_1, \ldots, \lambda_t)$ if $N(\lambda_1, \ldots, \lambda_t)$ is finite and $\mu(\lambda_1, \ldots, \lambda_t) := \infty$ if not.*

As an immediate consequence of Proposition 2 we have.

Proposition 3. *Consider the set $\Delta(R, \rho, \varphi) = \{\alpha(1), \ldots \alpha(n)\}$ and the basis $B = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n\}$ from Definition 5. Let $\alpha(s) \in \Delta(R, \rho, \varphi)$. For $i = 1, \ldots, n$ we have $\bar{\mu}(i) \geq \mu(\alpha(i))$. In larger generality for $\{a_1, \ldots, a_t\} \subseteq I$ we have $\bar{\mu}(a_1, \ldots, a_t) \geq \mu(\alpha(a_1), \ldots, \alpha(a_t))$.*

The results concerning the generalized Hamming weights in (5) are now easily translated into the setting of codes from order domains. We consider only two particular choices of subsets G of B.

Definition 13. *Given a basis $\mathcal{B}$ as in Theorem 4 and a morphism φ let*

$$C(\lambda) := \{\boldsymbol{c} \in \mathbb{F}_q^n \mid \boldsymbol{c} \cdot \varphi(f_\gamma) = 0 \textit{ for all } \gamma \preceq \lambda\}$$
$$\tilde{C}(\delta) := \{\boldsymbol{c} \in \mathbb{F}_q^n \mid \boldsymbol{c} \cdot \varphi(f_{\alpha(i)}) = 0 \textit{ for all } \alpha(i) \in \Delta(R,\rho,\varphi) \textit{ with } \mu(\alpha(i)) < \delta\}$$

For the case of order domains of transcendence degree 1 the set of codes $C(\lambda)$ are the set of duals to one-point geometric Goppa codes. For larger transcendence degree the set of codes $C(\lambda)$ includes the q-ary Reed-Muller codes but also many other codes. The codes $\tilde{C}(\delta)$ are examples of Feng-Rao improved codes. The theorem below is an immediate consequence of (5) and the above discussion.

Theorem 6. *For* $1 \leq t \leq \dim(C(\lambda))$ *respectively* $1 \leq t \leq \dim(\tilde{C}(\delta))$ *we have*

$$d_t(C(\lambda)) \geq \min\{\mu(\eta_1,\ldots,\eta_t) \mid \{\eta_1,\ldots,\eta_t\} \subseteq \Delta(R,\rho,\varphi), \lambda \prec \eta_s \textit{ for } s = 1,\ldots,t\} \tag{12}$$

$$d_t(\tilde{C}(\delta)) \geq \min\{\mu(\eta_1,\ldots,\eta_t) \mid \{\eta_1,\ldots,\eta_t\} \subseteq \Delta(R,\rho,\varphi), \mu(\eta_s) \geq \delta \textit{ for } s = 1,\ldots,t\} \tag{13}$$

The bound (12) for the codes $C(\lambda)$ is identical to the bound given by Heijnen and Pellikaan in [7, Th. 3.14]. It is known that (12) gives the actual values of the t generalized Hamming weights of the q-ary Reed-Muller codes (see [7]). It is also known that (12) gives the actual values of the tth generalized Hamming weights of the Hermitian codes (see [2]). For the case of hyperbolic codes (improved q-ary Reed-Muller codes) (13) gives exactly the same estimates as was found in [5]. We note that the result concerning the condition for tth rank MDS from the previous section is easily translated into the setting of the present section. Also we note that one can show that applying the Shibuya-Sakaniwa bound (8) to the codes of this section would produce the same estimates as is found by using(12) and(13).

5 Examples

The following two examples deals with codes coming from the Hermitian curve.

Example 1. Consider the factorring $R = \mathbb{F}_{q^2}[X,Y]/I$ where $I := \langle X^{q+1} - Y^q - Y\rangle$ means the ideal generated by $X^{q+1} - Y^q - Y$. The set $\mathcal{B} = \{X^aY^b + I \mid 0 \leq a, 0 \leq b < q\}$ constitutes a basis for R as a vectorspace over $\mathbb{F}_{q^2}$. Consider the map $\rho : \mathcal{B} \rightarrow \mathbb{N}_0$, $\rho(X^aY^b + I) = qa + (q+1)b$. This map is easily extended to a weight function ρ on R by applying the rules (W.0), (W.1) and (W.2) from Definition 9. With this weight function, the basis $\mathcal{B}$ can be indexed to be of the form described in Theorem 4. The polynomial $X^{q+1} - Y^q - Y$ has q^3 zeros $P_1,\ldots,P_{q^3}$ which give rise to the following morphism $\varphi : R \rightarrow \mathbb{F}_{q^2}^{q^3}$ $\varphi(G(X,Y) + I) = (G(P_1),\ldots,G(P_{q^3}))$. We get

$$\Delta(R,\rho,\varphi) = \{\alpha(1),\ldots,\alpha(q^3)\} = \{qa + (q+1)b \mid 0 \leq a < q^2, 0 \leq b < q\}.$$

The basis B that we should use for the code construction is $B = \{\boldsymbol{b}_i \mid i = 1,\ldots,q^3\}$ where $\boldsymbol{b}_i := \varphi(X^aY^b + I)$ with $0 \leq a < q^2$, $0 \leq b < q$ and $qa + (q +$

$1)b = \alpha(i)$. Using (12) and (13) we calculate the following parameters for some Hermitian codes and improved such ones (recall that we noted above that the bound (12) is sharp for the non-improved Hermitian codes). The codes in the first two arrays are defined over $\mathbb{F}_{16}$ and are of length $n = 64$. The codes in the last two arrays are defined over $\mathbb{F}_{64}$ and are of length $n = 512$. A bolded number means that the generalized Singleton bound is attained. We note that Theorem 3 predicts exactly the bolded numbers.

	k	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9
$\tilde{C}(6)$	55	6	8	9	11	12	14	15	16	**18**
$C(14)$	55	4	8	9	12	13	14	**16**	**17**	**18**

	k	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8
$\tilde{C}(9)$	51	9	12	14	15	17	18	19	**21**
$C(18)$	51	8	12	13	16	17	18	**20**	**21**
$C(19)$	50	9	13	14	17	18	19	**21**	**22**

	k	d_1	d_2	d_3	d_4	d_5	d_6	d_7
$\tilde{C}(18)$	476	18	21	24	26	27	30	32
$C(63)$	476	9	17	18	25	26	27	33
$C(72)$	467	18	26	27	34	35	36	42

	k	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9
$\tilde{C}(5)$	504	5	6	7	8	9	12	13	14	15
$C(25)$	504	4	5	6	7	8	11	12	13	14
$C(27)$	502	5	6	7	8	9	13	14	15	16

We take a closer look at the code $\tilde{C}(6)$ and $C(14)$ over $\mathbb{F}_{16}$. These codes are of the same dimension $k = 55$ and $\tilde{C}(6)$ is indeed an improved code as $d(\tilde{C}(6)) \geq 6 > 4 = d(C(14))$ holds. Nevertheless we observe that the estimated value of d_7 respectively d_8 of $\tilde{C}(6)$ are smaller than $d_7(C(14))$ respectively $d_8(C(14))$. A similar phenomenon occurs for the the codes $\tilde{C}(9)$ and $C(18)$ over $\mathbb{F}_{16}$.

In the next example we consider not only one, but two different bases B and B' related to the Hermitian curve. This will allow us to demonstrate that the bound (5) can actually be better than the Shibuya-Sakaniwa bound (8).

Example 2. Consider $R = \mathbb{F}_4[X,Y]/I$ where $I = \langle X^3 + Y^2 + Y \rangle$. Let φ be as in the previous example and consider the following two bases for $\mathbb{F}_4^8$.

$$\begin{aligned} B = \{&\boldsymbol{b}_1 = \varphi(1+I), \boldsymbol{b}_2 = \varphi(X+I), \boldsymbol{b}_3 = \varphi(Y+I), \boldsymbol{b}_4 = \varphi(X^2+I), \\ &\boldsymbol{b}_5 = \varphi(XY+I), \boldsymbol{b}_6 = \varphi(X^3+I), \boldsymbol{b}_7 = \varphi(X^2Y+I), \boldsymbol{b}_8 = \varphi(X^3Y+I)\} \\ B' = \{&\boldsymbol{b}'_1 = \varphi(1+I), \boldsymbol{b}'_2 = \varphi(X+I), \boldsymbol{b}'_3 = \varphi(XY+X^2+Y+I), \\ &\boldsymbol{b}'_4 = \varphi(XY+X^2+I), \boldsymbol{b}'_5 = \varphi(XY+I), \boldsymbol{b}'_6 = \varphi(X^2Y+X^3+I), \\ &\boldsymbol{b}'_7 = \varphi(X^2Y+I), \boldsymbol{b}'_8 = \varphi(X^3Y+I)\} \end{aligned}$$

Given a monomial X^aY^b we define the weight of X^aY^b to be $w(X^aY^b) := 2a+3b$. The following observations will play an important role to us:

$$\left.\begin{array}{l} w(X^aY^b) = 0 \Rightarrow \varphi(X^aY^b+I) \in L_1 \\ w(X^aY^b) = s \Rightarrow \varphi(X^aY^b+I) \in L_s \backslash L_{s-1} \text{ for } s = 2,3,4,5,6,7 \\ w(X^aY^b) = 9 \Rightarrow \varphi(X^aY^b+I) \in L_8 \backslash L_7 \end{array}\right\} \tag{14}$$

Consider the Hermitian code $C(3)$ (made from B). Clearly, this code has parameters $[n,k,d] = [8,5,3]$. We now show that for the particular choice of B' our new bound (5) will give us at least $d(C(3)) \geq 2$ whereas the Shibuya-Sakaniwa bound will only give $d(C(3)) \geq 1$.

The bound (5) calls for an estimation of the values $\bar{\mu}(4), \ldots, \bar{\mu}(8)$. By use of (14) we get the following estimates: $\bar{\mu}(4) \geq 2$ as $\boldsymbol{b}_2 * \boldsymbol{b}'_2, \boldsymbol{b}_4 * \boldsymbol{b}'_1 \in L_4 \backslash L_3$ and $(2,2),(4,1)$ are OWB. $\bar{\mu}(5) \geq 3$ as $\boldsymbol{b}_1 * \boldsymbol{b}'_3, \boldsymbol{b}_3 * \boldsymbol{b}'_2, \boldsymbol{b}_5 * \boldsymbol{b}'_1 \in L_5 \backslash L_4$ and $(1,3),(3,2),(5,1)$ are OWB. $\bar{\mu}(6) \geq 2$ as $\boldsymbol{b}_4 * \boldsymbol{b}'_2, \boldsymbol{b}_6 * \boldsymbol{b}'_1 \in L_6 \backslash L_5$ and $(4,2),(6,1)$ are OWB. $\bar{\mu}(7) \geq 4$ as $\boldsymbol{b}_1 * \boldsymbol{b}'_6, \boldsymbol{b}_2 * \boldsymbol{b}'_3, \boldsymbol{b}_5 * \boldsymbol{b}'_2, \boldsymbol{b}_7 * \boldsymbol{b}'_1 \in L_7 \backslash L_6$ and $(1,6),(2,3),(5,2),(7,1)$ are all OWB. $\bar{\mu}(8) \geq 5$ as $\boldsymbol{b}_1 * \boldsymbol{b}'_8, \boldsymbol{b}_2 * \boldsymbol{b}'_6, \boldsymbol{b}_4 * \boldsymbol{b}'_3, \boldsymbol{b}_7 * \boldsymbol{b}'_2, \boldsymbol{b}_8 * \boldsymbol{b}'_1 \in L_8 \backslash L_7$ and $(1,8),(2,6),(4,3),(7,2),(8,1)$ are all OWB. Hence, from (5) we get $d(C(3)) \geq 2$. We next apply Definition 8 and (8) in Theorem 2. We will show that for $T = \{1,2,3\}$ we have $\{6\} \subseteq \Lambda_T^*$. From this we can conclude that $\eta_1 = 3 - 3 = 0$ and therefore (8) becomes $d(C(3)) \geq 0 + 1 = 1$. To establish $\{6\} \subseteq \Lambda_T^*$ we will in the following show that there is no pair (i,j), $i \in \{1,2,3\}$, $j \in \{1,\ldots,8\}$ such that $\boldsymbol{b}_i * \boldsymbol{b}'_j \in L_6 \backslash L_5$. By use of (14) we get $\boldsymbol{b}_1 * \boldsymbol{b}'_1 \in L_1$, $\boldsymbol{b}_1 * \boldsymbol{b}'_2 \in L_2$, $\boldsymbol{b}_1 * \boldsymbol{b}'_3, \boldsymbol{b}_1 * \boldsymbol{b}'_4, \boldsymbol{b}_1 * \boldsymbol{b}'_5 \in L_5$, $\boldsymbol{b}_1 * \boldsymbol{b}'_6, \boldsymbol{b}_1 * \boldsymbol{b}'_7 \in L_7 \backslash L_6$, $\boldsymbol{b}_1 * \boldsymbol{b}'_8 \in L_8 \backslash L_7$, $\boldsymbol{b}_2 * \boldsymbol{b}'_1 \in L_2$, $\boldsymbol{b}_2 * \boldsymbol{b}'_2 \in L_4$, $\boldsymbol{b}_2 * \boldsymbol{b}'_3, \boldsymbol{b}_2 * \boldsymbol{b}'_4, \boldsymbol{b}_2 * \boldsymbol{b}'_5 \in L_7 \backslash L_6$, $\boldsymbol{b}_2 * \boldsymbol{b}'_6, \boldsymbol{b}_2 * \boldsymbol{b}'_7 \in L_8 \backslash L_7$, $\boldsymbol{b}_3 * \boldsymbol{b}'_1 \in L_3$, $\boldsymbol{b}_3 * \boldsymbol{b}'_2 \in L_5$.

It remains to study the incidents (i,j), $i \in \{1,2,3\}$ for which (14) does not immediately apply. We get

$$\begin{aligned}
&\boldsymbol{b}_2 * \boldsymbol{b}'_8 = \varphi(X^4Y + I) = \varphi(XY + I) \in L_5 \\
&\boldsymbol{b}_3 * \boldsymbol{b}'_3 = \varphi(XY^2 + X^2Y + Y^2 + I) = \varphi(X + XY + X^2Y + X^3 + Y + I) \in L_7 \backslash L_6 \\
&\boldsymbol{b}_3 * \boldsymbol{b}'_4 = \varphi(XY^2 + X^2Y + I) = \varphi(X + XY + X^2Y + I) \in L_7 \backslash L_6 \\
&\boldsymbol{b}_3 * \boldsymbol{b}'_5 = \varphi(XY^2 + I) = \varphi(X + XY + I) \in L_5 \\
&\boldsymbol{b}_3 * \boldsymbol{b}'_6 = \varphi(X^2Y^2 + X^3Y + I) = \varphi(X^2 + X^2Y + X^3Y + I) \in L_8 \backslash L_7 \\
&\boldsymbol{b}_3 * \boldsymbol{b}'_7 = \varphi(X^2Y^2 + I) = \varphi(X^2 + X^2Y + I) \in L_7 \backslash L_6 \\
&\boldsymbol{b}_3 * \boldsymbol{b}'_8 = \varphi(X^3Y^2 + I) = \varphi(X^3 + X^3Y + I) \in L_8 \backslash L_7.
\end{aligned}$$

We have shown that there is no pair (i,j), $i \in \{1,2,3\}$, $j \in \{1,\ldots,8\}$ such that $\boldsymbol{b}_i * \boldsymbol{b}'_j \in L_6 \backslash L_5$ and therefore by the above discussion (8) becomes $d(C(3)) \geq 1$.

Acknowledgments

The authors wish to thank the anonymous referees for their valuable remarks.

References

1. H. E. Andersen, O. Geil, The Missing Evaluation Codes from Order Domain Theory, (2004), submitted.
2. A. I. Barbero, C. Munuera, The Weight Hierarchy of Hermitian Codes, *Siam J. Discrete Math.,* **13** (2000), 79–104.
3. G.-L. Feng and T.R.N. Rao, Decoding of algebraic geometric codes up to the designed minimum distance, *IEEE Trans. Inf. Theory*, **39**, (1993) 37-46.
4. G.-L. Feng and T.R.N. Rao, Improved Geometric Goppa Codes, Part I:Basic theory, *IEEE Trans. Inf. Theory*, **41**, (1995), 1678-1693.
5. O. Geil and T. Høholdt, On Hyperbolic Codes, Proc. AAECC-14, *Lecture Notes in Comput. Sci. 2227*, Springer, Berlin, 2001, 159-171.

6. O. Geil and R. Pellikaan, On the Structure of Order Domains, *Finite Fields and their Applications*, **8**, (2002), 369-396.
7. P. Heijnen, R. Pellikaan, Generalized Hamming weights of q-ary Reed-Muller codes, *IEEE Trans. Inf. Theory*, **44**, (1998), 181-196.
8. T. Høholdt, J. van Lint and R. Pellikaan, Algebraic Geometry Codes, Chapter 10 in "Handbook of Coding Theory," (V.S. Pless and W.C. Huffman, Eds.), vol. 1, Elsevier, Amsterdam, 1998, 871-961.
9. C. Kirfel and R. Pellikaan, The minimum distance of codes in an array coming from telescopic semigroups, *IEEE Trans. Inf. theory*, **41**, (1995), 1720-1732.
10. R. Matsumoto and S. Miura, On the Feng-Rao Bound for the $\mathcal{L}$-Construction of Algebraic Geometry Codes, *IEICE Trans. Fund.,* **E83-A**, no. 5 (2000), 923-927.
11. S. Miura, On error correcting codes based on algebraic geometry, Ph.D. thesis, Univ. Tokyo, May 1997, (in Japanese).
12. S. Miura, Linear Codes on Affine Algebraic Varieties, *Trans. IEICE*, **J81-A**, no. 10 (1998), 1386-1397 (in Japanese).
13. S. Miura, Linear Codes on Affine Algebraic Curves, *Trans. IEICE*, **J81-A**, no. 10 (1998), 1398-1421 (in Japanese).
14. R. Pellikaan, On the efficient decoding of algebraic-geometric codes, *Eurocode 92*, edited by P. Camion, P. Charpin and S. Harari, Udine, CISM Courses and Lectures, **339**, Springer-Verlag, (1993), 231-253.
15. T. Shibuya, J. Mizutani, K. Sakaniwa, On generalized Hamming Weights of codes constructed on Affine algebraic sets, Proc. AAECC-12, *Lecture Notes in Computer Science*, **vol. 1255**, Springer-Verlag, (1997), 311-320.
16. T. Shibuya, J. Mizutani, K. Sakaniwa, On generalized Hamming Weights of codes constructed on Affine algebraic sets, *IEICE Trans. Fund.*, **E81-A** (1998), 1979–1989.
17. T. Shibuya and K. Sakaniwa, Lower bound on generalized Hamming weights in terms of a notion af well-behaving, Proc. ISIT 1998, Cambridge, USA, (1998), 96.
18. T. Shibuya, R. Hasagawa, K. Sakaniwa, A Lower Bound for Generalized Hamming Weights and a Condition for t-th Rank MDS, *IEICE Trans. Fund.*, **E82-A**, (1999), 1090-1101.
19. T. Shibuya and K. Sakaniwa, A Dual of Well-Behaving Type Designed Minimum Distance, *IEICE Trans. Fund.*, **E84-A**, (2001), 647–652.
20. T. Shibuya and K. Sakaniwa, A note on a Lower Bound for General Hamming Weights, *IEICE Trans. Fund.*, **E84-A**, (2001), 3138–3145.
21. V. K. Wei, Generalized Hamming weights for linear codes, *IEEE Trans. Inf. Theory*, **37**, (1991), 1412-1418.

Nested Codes for Constrained Memory and for Dirty Paper

Hans Georg Schaathun[1] and Gérard D. Cohen[2]

[1] Dept. Informatics, University of Bergen,
Pb. 7800, N-5020 Bergen, Norway
[2] Dept. Informatique et Reseaux,
Ecole Nationale Supérieure des Télécommunications,
46, rue Barrault, F-75634 Paris Cedex 13, France

Abstract. Dirty paper coding are relevant for wireless networks, multiuser channels, and digital watermarking. We show that the problem of dirty paper is essentially equivalent to some classes of constrained memories, and we explore the binary so-called nested codes, which are used for efficient coding and error-correction on such channels and memories.

Keywords: dirty paper, constrained memory, nested codes, covering codes.

The motivation of this paper is the dirty paper channel introduced by Costa [3]. This channel has received increased attention [4] in recent years, due to applications in wireless multiuser networks and digital fingerprinting [5].

We show that the dirty paper channel is practically equivalent to writing on reluctant memories, and we make a few improvements on the existing results for such channels. Our interest is mainly in the binary dirty paper channel (BDP).

1 Dirty Paper and Constrained Memory Coding

The dirty paper channel is depicted in Figure 1. There are two independent noise sources which are added to the transmitted signal to form the received signal. The first noise vector, which we will call the *state* of the channel is known to the sender but not to the receiver. The second noise vector, which we will refer to as *noise* is unknown to both.

The sender is subject to a power constraint $||\mathbf{x}|| \leq P$ on the transmitted signal. For a binary channel $||\cdot||$ is usually the Hamming norm; for a continuous channel it is usually the Euclidean norm.

Costa [3] introduced this channel with Gaussian sources for both the state and the noise. His surprising result was that the channel capacity depends only on the intensity of the noise; the intensity of the state does not change capacity. In more recent years, his results have been generalised to other source distributions. We will consider the binary dirty paper channel (BDP).

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 307–316, 2006.

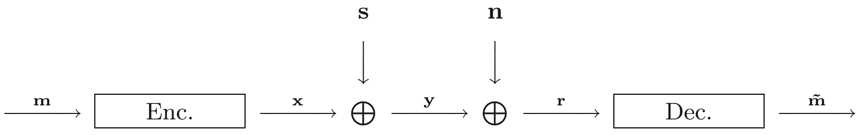

Fig. 1. The dirty paper channel

In a constrained memory, there are restrictions on writing to the memory, such that starting in one memory state, some states are reachable in one write operation and others are not. For each memory state, there is a feasible region of words which may be used to represent the next message. In this case the state is given by the previous message stored in memory.

Dirty paper coding and constrained memory coding are similar, in fact BDP channels are practically equivalent to WRM (write reluctant memories) with error-correction [2]. In WRM, one write operation cannot change more than a certain number P of bits. This corresponds to the power constraint in BDP; if $\mathbf{s}$ is the state (previous contents), $\mathbf{x}$ is the change, and $\mathbf{y} = \mathbf{s} + \mathbf{x}$ is the memory contents after writing, then $w(\mathbf{x}) \leq P$.

The state on dirty paper channels is externally given, whereas in constrained memories it is the old codeword (with possible errors). The state, together with power constraints, defines the feasible region of vectors $\mathbf{y}$ which can be generated. For BDP/WRM, the feasible region is a Hamming sphere around the state.

Remark 1. Occasionnally, in constrained memories, one assumes that $\mathbf{s}$ is a codeword with few errors, since nobody would write rubbish to the memory. We will not make this assumption, for two reasons. Primarily, it does not extend to BDP. Also, we know of no cases where results can be improved due to this assumption. Furthermore, by avoiding such assumption, the system can recover after an error pattern which could not be corrected.

Example 1. Another example of constrained memory is the Write Isolated Memory (WIM), where two consecutive memory bits cannot be changed in the same operation. In other words, the feasible region is the set $\{\mathbf{x}+\mathbf{s} : \mathbf{x} = (x_1, \ldots, x_n), x_i = 1 \Rightarrow x_{i-1} = x_{i+1} = 0\}$, where $\mathbf{s}$ is the memory state and $x_0 = x_{n+1} = 0$ by convention.

BDP (WRM) and WIM both fall into a class of channels, where the feasible regions are translation invariant, permitting some common techniques. By this we mean that if $F_{\mathbf{s}}$ is the feasible region from $\mathbf{s}$, then $F_{\mathbf{s}'} = F_{\mathbf{s}} - \mathbf{s} + \mathbf{s}'$. Let us call this class CCTIR (constrained channels with translation invariant regions).

2 Some Coding Theory

An $(n, M)_q$ code C is an M-set of n-tuples over a q-ary alphabet. When $q = 2$ we may suppress the subscript. The Hamming distance $d(\mathbf{x}, \mathbf{y})$ is the number of positions where the two tuples differ. The minimum distance $d = d(C)$ of C is the least distance between two different codewords. We say that C is an $(n, M, d)_q$ code. The covering radius r of C is the largest distance between a vector $\mathbf{y} \in Q^n$ and the code.

The problem of covering codes amounts to finding codes minimising r given n and M, whereas the problem of error-correcting codes is about maximising d given n and M.

We also define normalised measures, which will be useful when moving to asymptotic codes. We define the rate $\log_q M/n$, the distance $\delta = d/n$, and the covering radius $\rho = r/n$.

3 Codes for CCTIR

In order to make a successful code for CCTIR, we need for every state $\mathbf{s}$ and every message $\mathbf{m}$, to have at least one codeword $\mathbf{x}$ corresponding to $\mathbf{m}$ in the feasible region of $\mathbf{s}$. Furthermore, we require any capability for error-correction that we may need. We will study e-error correcting CCTIR codes.

Lemma 1. *For CCTIR, if* $\mathbf{x} \in F_{\mathbf{y}}$ *then* $\mathbf{y} \in F_{\mathbf{x}}$.

Let B_i be the set of words corresponding to message i. We require that for any $\mathbf{s}$, $F_{\mathbf{s}} \cap B_i \neq \emptyset$. By the lemma above, this is equivalent to

$$\bigcup_{\mathbf{b} \in B} F_{\mathbf{b}} = \mathbb{F}^n, \tag{1}$$

i.e. that the feasible regions around the words of B_i cover the space. If the set of possible messages is $i = 1, \dots, M$, then we define

$$C_F = \bigcup_{i=1}^{M} B_i.$$

When the feasible regions are spheres of radius ρ, this is to say that B_i must be a covering code of covering radius ρ or smaller. For other feasible regions it is a more general *covering by* F*-shapes.*

In order to correct up to e errors, we require that if $i \neq j$, then $d(B_i, B_j) > 2e$. It is sufficient to require that C_F has minimum distance $2e{+}1$ or more; i.e. that C_F is e-error correcting. Furthermore as a necessary condition, if there are two codewords with distance at most $2e$ apart, they must fall in the same set B_i.

In a sense, we try to pack the space with coverings B_i such that we maintain a minimum distance of $2e + 1$, a problem studied in [2].

We say that a CCTIR code $(B_1, \dots, B_M)$ is linear if C_F is a linear e-error-correcting code, B_j is a subcode satisfying (1) for some j, and the B_i are cosets of B_j in C_F. Clearly by linearity, B_i satisfies (1) whenever B_j does.

Let $a_n = \#F_{\mathbf{0}}$. For CCTIR, all the feasible regions clearly have the same size.

Lemma 2. *For an (n, M) CCTIR code, we have*

$$M \leq a_n$$

Lemma 3. *For dirty paper codes, we have*

$$a_n = V(n, R) = \sum_{i=0}^{n} \binom{n}{i}.$$

In the case of WRM and dirty paper channels, a linear CCTIR code is also called a nested code. We call C_F the fine code and $C_C \subseteq C_F$ the coarse code. The nested code is the quotient $C = C_F/C_C$, and we say that C is an $[n, K; d_1, \rho]$ code, where $K = k_F - k_C$ is the dimension of C. The following lemma is well known.

Lemma 4 (Supercode lemma). *For any $[n, K; d_1, \rho]$ nested code, we have $\rho \geq d_1$.*

4 Asymptotic Existence

Definition 1 (Entropy). *The (binary) entropy of a discrete stochastic variable X drawn from a set $\mathcal{X}$ is defined as*

$$H(X) = -\sum_{x \in \mathcal{X}} P(X = x) \log P(X = x).$$

The conditional entropy of X with respect to another discrete stochastic variable Y from $\mathcal{Y}$ is

$$H(X|Y) = -\sum_{y \in \mathcal{Y}} P(Y = y) \sum_{x \in \mathcal{X}} P(X = x|Y = y) \log P(X = x|Y = y).$$

The following general theorem appears in [2].

Theorem 1. *For n large enough, there are θn-error correcting codes for CCTIR with rate*

$$\kappa(\theta) \geq \kappa_0 - H(2\theta),$$

where κ_0 is the maximum rate for a non-error-correcting code for the same constrained channel.

The proof is by greedy techniques, as follows.

Proof. We write

$$S(B, i) = \bigcup_{\mathbf{b} \in B} \{\mathbf{x} : d(\mathbf{x}, \mathbf{b} \leq i\}.$$

First we make a code C_C of rate $1 - \kappa$ without error-correction. Let $S_0 = S(C_C, 2\theta n - 1)$.

We start with $B = \{\mathbf{0}\}$, and construct a code C_C by the following greedy algorithm. In each step we take a random vector $\mathbf{y} \in S \backslash S(B + C_C, 2\theta n - 1)$, and update B to be the linear span of $\mathbf{y}$ and the vectors of B. We proceed until $S \backslash S(B + C_C, 2\theta n - 1)$ is empty. Since each word included in B excludes at most $\#S(C_C, 2\theta n - 1)$ elements from S_0, we get that

$$\#B \geq \frac{2^n}{\#C_C \#S(\{\mathbf{0}\}, 2\theta n - 1)} \geq \frac{2^{\kappa n}}{\#S(\{\mathbf{0}\}, 2\theta n - 1)}.$$

Assymptotically, we have $\#B \approx 2^{(\kappa - H(2\theta))n}$, Let $C_F = B + C_C$, so that $C = C_F / C_C \equiv B$. Clearly the rate of B and C is $\kappa - H(2\theta)$ as required.

In the case of dirty paper channel, $\kappa_0 = 1 - R_\rho$ where R_ρ is the minimum rate for a covering code with appropriate ρ.

Theorem 2. *For dirty paper codes with no error-correction, we can obtain rate* $\kappa_0 = H(\rho)$.

Observe that whenever $\rho > \delta$, we get asymptotic codes with non-zero rate from the above theorems. For $\rho = \delta$, however, the guaranteed rate is just zero.

Problem 1. *Are there asymptotic families of nested codes with* $R > 0$ *and* $\rho = d$*?*

5 Some Small Constructions

Lemma 5. *For any* $[n, K; 1, 1]$ *nested code with even* n*, we have* $K \leq \log n$*.*

Proof. For a $[n, k_C]1$ covering code, we have $k_C \leq n - \log n$ when n is even, and for an $[n, k_F, 1]$ code, we have $k_F \leq n$. Hence $K = k_F - k_C \leq \log n$.

Lemma 6. *There is a* $[2^K - 1, K; 1, 1]$ *nested code for any* K*.*

Proof. Let the coarse code be the $[2^K - 1, 2^K - 1 - K, 3]1$ Hamming code, and let the fine code be the $[2^K - 1, 2^K - 1, 1]$ code.

Table 1. Some nested codes for $n = 3$

Parameters	Coarse code	Fine code
$[3, 1; 1, 1]$	$\begin{bmatrix}110\\101\end{bmatrix}$	$\begin{bmatrix}100\\010\\001\end{bmatrix}$
$[3, 1; 2, 2]$	$\begin{bmatrix}110\end{bmatrix}$	$\begin{bmatrix}110\\011\end{bmatrix}$
$[3, 2; 1, 2]$	$\begin{bmatrix}110\end{bmatrix}$	$\begin{bmatrix}100\\010\\001\end{bmatrix}$

Table 2. Some nested codes for $n = 4, 5, 6$

Parameters	Coarse code	Fine code
$[4, 2; 1, 1]$	$\begin{bmatrix}1110\\1001\end{bmatrix}$	$\begin{bmatrix}1000\\0100\\0010\\0001\end{bmatrix}$
$[4, 2; 2, 2]$	$\begin{bmatrix}1111\end{bmatrix}$	$\begin{bmatrix}1100\\1010\\1001\end{bmatrix}$
$[4, 3; 1, 2]$	$\begin{bmatrix}1111\end{bmatrix}$	$\begin{bmatrix}1000\\0100\\0010\\0001\end{bmatrix}$
$[5, 2; 1, 1]$	$\begin{bmatrix}11100\\10011\\00110\end{bmatrix}$	$[5, 5, 1]$ full code
$[5, 4; 1, 2]$	$\begin{bmatrix}11111\end{bmatrix}$	$[5, 5, 1]$ full code
$[5, 2; 2, 2]$	$\begin{bmatrix}11111\end{bmatrix}$	$\begin{bmatrix}11000\\10100\\10010\end{bmatrix}$
$[6, 2; 2, 2]$	$\begin{bmatrix}111000\\000111\end{bmatrix}$	$\begin{bmatrix}111000\\000111\\100100\\010010\end{bmatrix}$
$[6, 4; 1, 2]$	$\begin{bmatrix}111000\\000111\end{bmatrix}$	$[6, 6, 1]$ full code
$[6, 4; 2, 3]$	$\begin{bmatrix}111111\end{bmatrix}$	$[6, 5, 2]$ even weight

Lemma 7. *There are $[2^K, K; 1, 1]$ and $[2^K - 1, K; 1, 1]$ nested codes.*

Proof. The fine code is the $[n, n, 1]$ full space. The coarse codes are the Hamming codes, and the $[2^K, 2^K - K, 1]$ direct sum of a Hamming code padded with a zero column, and the code generated by a single word of weight one.

Lemma 8. *There are $[2^K, K; 2^{K-1}, 2^{K-1}]$ and $[2^K - 1, K; 2^{K-1} - 1, 2^{K-1} - 1]$ nested codes for any positive K.*

Proof. Let the coarse code be the $[2^K, 1, 2^K]$ repetition code, and let the fine code be the $[2^K, K, 2^{K-1}]$ Reed-Muller code. The second set of parameters comes from puncturing the above code.

Lemma 9. *If there is an $[n, K; d, \rho]$ code, then there is an $[n - 1, K; d - 1, \rho]$ code by puncturing and an $[n - 1, K; d, \rho + 1]$ code by shortening.*

Proof. This follows easily from the standard results on puncturing and shortening of error-correcting and covering codes.

Lemma 10 ([2]). *The $[2^m - 1, 2^m - 1 - 2m, 5]$ BCH code has $\rho = 3$ for $m \geq 3$.*

Table 3. Some nested codes for $n \geq 7$

Parameters	Coarse code	Fine code
$[7,3;1,1]$	$[7,4;3]1$ Hamming	$[7,7,1]$
$[7,3;3,3]$	$[7,1,7]3$ repetition	$[7,4,3]$ Hamming
$[8,3;1,1]$	$\begin{bmatrix}10001110\\01000110\\00101010\\00010110\\00000001\end{bmatrix}$	$[8,8,1]$
$[8,3;4,4]$	$[8,1,8]4$ repetition	$[8,4,4]$ ext. Hamming
$[15,4;3,3]$	$[15,7,5]3$ BCH(2)	$[15,11,3]$ Hamming
$[15,2;5,5]$	$[15,5,7]5$ BCH(3)	$[15,7,5]3$ BCH(2)
$[15,2;7,7]$	$[15,1,15]7$ repetition	$[15,3,7]$ BCH(3)
$[15,4;7,7]$	$[15,1,15]7$ repetition	$[15,5,7]$ punctured Reed-Muller
$[16,6;4,6]$	$[16,5,8]6$ RM$(1,4)$	$[16,11,4]$ RM$(2,4)$
$[16,4;8,8]$	$[16,1,16]8$ repetition	$[16,5,8]$ Reed-Muller
$[27,6;11,13]$	$[27,1,27]13$ repetition	$[27,7,11]$ [1]
$[28,6;12,14]$	$[28,1,28]14$ repetition	$[28,7,12]$ [1]
$[31,5;3,3]$	$[31,21,5]3$ BCH(2)	$[31,26,3]$ Hamming
$[31,5;5,5]$	$[31,16,7]5$ BCH(3)	$[31,21,5]3$ BCH(2)
$[31,5;7,7]$	$[31,11,11]7$ BCH(4)	$[31,16,7]5$ BCH(3)
$[31,5;11,11]$	$[31,6,15]11$ BCH(6)	$[31,11,11]7$ BCH(4)
$[31,5;15,15]$	$[31,1,31]15$ repetition	$[31,6,15]$ punctured Reed-Muller
$[32,5;2,2]$	$[32,26,4]2$ RM$(3,5)$	$[32,31,2]$ RM$(4,5)$
$[32,10;4,6]$	$[32,16,8]6$ RM$(2,5)$	$[32,26,4]$ RM$(3,5)$
$[32,10;8,12]$	$[32,6,16]12$ RM$(1,5)$	$[32,16,8]$ RM$(2,5)$
$[36,20;4,13]$	$[36,8,16]\rho$ $\rho \leq 13$	$[36,28,4]$ $C_C^{\perp}$
$[49,9;20,24]$	$[49,1,49]24$ repetition	$[49,10,20]$ [1]
$[63,6;1,1]$	$[63,57,3]1$ BCH(1)	$[63,63,1]$ full code
$[63,6;3,3]$	$[63,51,5]3$ BCH(2)	$[63,57,3]1$ BCH(1)
$[63,6;5,5]$	$[63,45,7]5$ BCH(3)	$[63,51,5]3$ BCH(2)
$[63,6;7,7]$	$[63,39,9]7$ BCH(4)	$[63,45,7]5$ BCH(3)
$[63,3;9,9]$	$[63,36,11]9$ BCH(5)	$[63,39,9]7$ BCH(4)
$[64,15;4,8]$	$(u,u+v)$ construction	
$[64,15;16,28]$	$[64,7,32]28$ RM$(1,6)$	$[64,22,16]$ RM$(2,6)$

Corollary 1. *There is a* $[2^m-1,m;3,3]$ *nested code for every* $m \geq 3$.

Proof. The coarse code is the $[2^m-1, 2^m-1-2m, 5]3$ BCH(2) code, and the fine code is the Hamming code.

Lemma 11 ([2]). *The* $[2^m-1, 2^m-1-3m, 7]$ *BCH code has* $\rho = 5$ *for* $m \geq 4$.

Corollary 2. *There is a* $[2^m-1,m;5,5]$ *nested code for every* $m \geq 4$.

Proof. The coarse code is the $[2^m-1, 2^m-1-3m, 7]5$ BCH(3) code, and the fine code is the $[2^m-1, 2^m-1-2m, 5]3$ BCH(2) code.

Lemma 12. *There are* $[2^{2m+1}-2^m, 2m+2; 2^{2m}-2^m, 2^{2m}]$ *and* $[2^{2m+1}-2^m-1, 2m+2; 2^{2m}-2^m-1, 2^{2m}-1]$ *nested codes.*

Proof. The coarse code is a repetition code. The fine code is a $[2^{2m+1}-2^m, 2m+3, 2^{2m}-2^m]$ code [1] or a punctured version of it.

Lemma 13. *There is no* $[6,4;2,2]$ *nested code, so the* $[6,3;2,2]$, $[6,4;1,2]$ *and* $[6,4;2,3]$ *codes are optimal.*

Proof. The smallest covering code of $\rho = 2$ and $n = 6$ has $k_C = 2$, so to get $K = 4$, we would need $k_F \geq 6$, which would give $d = 1$.

6 Some Upper Bounds on the Nested Code Dimension

Lemma 14. *For an* $[n, K; d, d]$ *nested code, we have*

$$2^K \leq \binom{n}{d} + 1.$$

Proof. Consider the points of C_C and the balls of radius $\rho = d$ around these points. Because ρ is the covering radius of C_C, these balls cover the space. Since C_F has minimum distance $d = \rho$, it can only contain points on the border of these balls, besides the points of C_C. Hence

$$\#C_F \leq \#C_C \cdot \left(\binom{n}{d} + 1\right),$$

and hence

$$\#(C_F/C_C) \leq \left(\binom{n}{d} + 1\right),$$

as required.

We have seen that this bound can be met with equality for $\rho = 1$. For $\rho > 1$ except $\rho = n = 2$ we have inequality; let's see this for $\rho = 2$ first.

Proposition 1. *For an* $[n, K; 2, 2]$ *nested code with* $n > 2$*, we have*

$$2^K < \binom{n}{2} + 1.$$

Proof. Suppose the bound were met with equality. Since C_C is a covering code of $\rho = 2$, we have

$$2^n \leq 2^{k} \ \left(1 + \binom{n}{1} + \binom{n}{2}\right) \leq 2^{k} \ (2^K + n) \leq 2^{k} \ \ + 2^{k} \ n.$$

For all $n > 2$, we have

$$n < 1 + \binom{n}{2},$$

which is equal to 2^K by assumption. This gives

$$2^n < 2^{k \ \ +1},$$

and clearly $n \geq k_F$, so we get $n = k_F$; but then $d = 1 < 2$, giving a contradiction.

We do have degenerate $[n, 1; n, n]$ nested codes for all n. They have only the zero word for C_C, an $[n, 1, n]$ repetition code for C_F.

Proposition 2. *For an $[n, K; d, d]$ nested code, we have*

$$2^K \leq A(n, d, d) + 1.$$

It is readily seen that this bound is stronger than Lemma 14 when $\rho > 2$.

Proof. We start as we did proving Lemma 14 with the balls of radius ρ around the points of C_C. The border of the ball around $\mathbf{x}$ are the points $\mathbf{x}+\mathbf{y}$ where $\mathbf{y}$ has weight ρ. Obeying the distance requirement, the $\mathbf{y}$ that we choose for C_F from this ball, will have to form a constant weight code of weight and distance $\rho = d$.

Generalising, we get the following proposition, for which we ommit the proof.

Proposition 3. *For an $[n, K; d, \rho]$ nested code, we have*

$$2^K \leq 1 + \sum_{w=d}^{\rho} A(n, d, w).$$

7 Some Constructions

Theorem 3. *Let $U = U_F/U_C$ and $V = V_F/V_C$ be $[n, K_U; d_U, \rho_U]$ and $[n, K_V; d_V, \rho_V]$ nested codes. Let $U_i \circ V_i$ denote the $(\mathbf{u}, \mathbf{u}+\mathbf{v})$ composition of U_I and U_V. Then we can form a nested code $C = U \circ V = (U_F \circ V_F)/(U_C \circ U_F)$, and C is a $[2n, K_U + K_V; d, \rho]$ nested code with $\rho \leq \rho_U + \rho_V$ and $d = \min\{2d_V, d_U\}$.*

The proof is obvious from fundamental results on the parameters of the component codes.

Acknowledgements

The authors are grateful for codes contributed by Carl Bracken of Dublin, and for the interest of and discussions with Wolfgang Willems of Magdeburg.

References

1. Carl Bracken. Private communication. 2004.
2. Gérard Cohen, Iiro Honkala, Simon Litsyn, and Antoine Lobstein. *Covering codes*, volume 54 of *North-Holland Mathematical Library*. North-Holland Publishing Co., Amsterdam, 1997.
3. Max H. M. Costa. Writing on dirty paper. *IEEE Trans. Inform. Theory*, 29(3):439–441, 1983.
4. Frank Kschischang and David Tse, editors. *Proc. IEEE Intern. Symp. Inform. Theory*, pages 533–536. June 2004. A four-talk session on dirty paper coding.
5. M.L. Miller, G.J. Doerr, and I.J. Cox. Applying informed coding and embedding to design a robust high-capacity watermark. *IEEE Transactions on Image Processing*, 13(6):792–807, June 2004.

Complementary Sets and Reed-Muller Codes for Peak-to-Average Power Ratio Reduction in OFDM*

Chao-Yu Chen[1], Chung-Hsuan Wang[2], and Chi-chao Chao[1]

[1] Institute of Communications Engineering, National Tsing Hua University,
Hsinchu 30013, Taiwan, R.O.C.
d919609@Oz.nthu.edu.tw, ccc@ee.nthu.edu.tw

[2] Department of Communication Engineering, National Chiao Tung University,
Hsinchu 30010, Taiwan, R.O.C.
chwang@mail.nctu.edu.tw

Abstract. One of the disadvantages of orthogonal frequency division multiplexing (OFDM) systems is the high peak-to-average power ratio (PAPR) of OFDM signals. Golay complementary sets have been proposed to tackle this problem. In this paper, we develop several theorems which can be used to construct Golay complementary sets and multiple-shift complementary sets from Reed-Muller codes. We show that the results of Davis and Jedwab on Golay complementary sequences and those of Paterson and Schmidt on Golay complementary sets can be considered as special cases of our results.

1 Introduction

Orthogonal frequency division multiplexing (OFDM) [1] is a technique to transmit data over a number of subcarriers. OFDM offers many advantages, and in particular, it can provide resistance to dispersion caused by multipath delay spread. Consequently, OFDM has been adopted for many types of wireless applications. One of the drawbacks in implementation is the high peak-to-average power ratio (PAPR) of OFDM signals. High PAPR leads to the consequence that power amplifiers with a large linear range are required, which results in high cost and limits widespread applications.

Many coding approaches [2]–[8] have been proposed to cope with this problem. In [3], Davis and Jedwab provided a connection between Golay complementary sequences [9] and Reed-Muller codes, along with a code which guarantees low PAPR. And Paterson proposed a theory linking Golay complementary sets [10] and second-order cosets of the first-order Reed-Muller codes in [4]. The results in [3] on Golay complementary sequences can be regarded as a special case of those proposed in [4] on complementary sets. However, only second-order cosets are considered in [4]. In [8], Schmidt and Finger generalized Paterson's work to cosets

* This work was supported by the National Science Council of the Republic of China under Grant NSC 93–2213–E–007–021.

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 317–327, 2006.

of arbitrary order, but the upper bound on PAPR is not very tight. In this paper, we propose a more powerful theory to link all the cosets of the first-order Reed-Muller codes and a tighter upper bound on PAPR. Furthermore, we provide a relationship between Reed-Muller codes and multiple shift complementary sets [11]–[13] which are an extension of Golay complementary sets.

2 Signal Model

For an even integer q, we denote $\boldsymbol{c} = (c_0, c_1, \ldots, c_{n-1})$, a $\mathbb{Z}_q$-valued sequence of length n, where c_i is in the ring $\mathbb{Z}_q = \{0, 1, \ldots, q-1\}$, and $\boldsymbol{x} = (x_0, x_1, \ldots, x_{n-1})$ $= (\xi^{c_0}, \xi^{c_1}, \ldots, \xi^{c \ -1})$, a q-PSK modulated sequence, where $\xi = e^{2\pi j/q}$ is a complex qth root of unity. For the sake of convenience, we use the equivalent complex baseband notation. Let an OFDM signal be given by

$$X_{\boldsymbol{c}}(t) = \sum_{i=0}^{n-1} x_i e^{j2\pi i t}, \qquad 0 \le t \le 1$$

where n is the number of subcarriers. The instantaneous power of an OFDM signal is then given by $P_{\boldsymbol{c}}(t) = |X_{\boldsymbol{c}}(t)|^2$. The average power P_{av} is n for equal energy constellations. The PAPR of a sequence $\boldsymbol{c}$ is defined as

$$\mathrm{PAPR}(\boldsymbol{c}) = \max_{0 \le t \le 1} \frac{P_{\boldsymbol{c}}(t)}{P_{av}}.$$

3 Golay Complementary Sets

Before describing Golay complementary sets, we introduce the relationship between the instantaneous power and the aperiodic autocorrelation. It is easy to show [4] that

$$P_{\boldsymbol{c}}(t) = n + 2 \sum_{u=1}^{n-1} \Re\left\{A_{\boldsymbol{c}}(u) e^{j2\pi u t}\right\}$$

where $\Re\{\cdot\}$ denotes the real part. The aperiodic autocorrelation function $A_{\boldsymbol{c}}(u)$ of $\boldsymbol{c}$ at displacement u is defined as

$$A_{\boldsymbol{c}}(u) = \sum_{k=0}^{n-1-u} x_{k+u} x_k^* = \sum_{k=0}^{n-1-u} \xi^{c \ + \ -c}, \qquad 0 \le u \le n-1.$$

Definition 1. *[10] A set of N length-n sequences $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_N$ is called a Golay complementary set if*

$$A_{\boldsymbol{c}_1}(u) + A_{\boldsymbol{c}_2}(u) + \cdots + A_{\boldsymbol{c}}\ (u) = \begin{cases} 0, & u \neq 0 \\ Nn, & u = 0. \end{cases}$$

If $N = 2$, the set is called a Golay complementary pair, and any sequence which is a member of such a pair is called a Golay complementary sequence. It has been shown in [14], [15], and [4] that the PAPR of any sequence of a Golay complementary set of size N is at most N.

4 Complementary Sets from Reed-Muller Codes

In this section, we will provide a relationship between Golay complementary sets and cosets of the first-order Reed-Muller codes. We denote the rth-order Reed-Muller code of length 2^m over $\mathbb{Z}_q$ by $\mathrm{RM}_q(r,m)$. Let the 2^m-tuple vectors

$$\boldsymbol{v}_i = (\underbrace{00\cdots0}_{2^{\ -1}}\underbrace{11\cdots1}_{2^{\ -1}}\underbrace{00\cdots0}_{2^{\ -1}}\cdots\underbrace{11\cdots1}_{2^{\ -1}}), \quad i=1,2,\ldots,m \tag{1}$$

and

$$\boldsymbol{v}_0 = (11\cdots1) \tag{2}$$

the all-one vector. For vectors $\boldsymbol{a} = (a_0, a_1, \ldots, a_{n-1})$ and $\boldsymbol{b} = (b_0, b_1, \ldots, b_{n-1})$, we denote $\boldsymbol{ab} = (a_0 \cdot b_0, a_1 \cdot b_1, \ldots, a_{n-1} \cdot b_{n-1})$ where "$\cdot$" represents the product. Then $\mathrm{RM}_q(r,m)$ is a linear code over $\mathbb{Z}_q$ generated by the following vectors [16]:

$$G_{\mathrm{RM}}(r,m) = \{\boldsymbol{v}_0, \boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m, \boldsymbol{v}_1\boldsymbol{v}_2, \ldots, \boldsymbol{v}_{m-1}\boldsymbol{v}_m,$$
$$\ldots, \text{ up to products of degree } r\}.$$

We will show that any codeword of arbitrary coset of $\mathrm{RM}_q(1,m)$ is contained in a Golay complementary set, and, as a result, we can derive an upper bound on PAPR for cosets of $\mathrm{RM}_q(1,m)$. Some of the proofs and derivations are sketched or omitted hereinafter, due to the length limitation.

Theorem 1. *For any even integer q, for any choice of $a_{i,J} \in \mathbb{Z}_q$, for any integer $k = 0,1,\ldots,m-1$, and for any permutation π of the symbols $\{1,2,\ldots,m\}$, if we denote*

$$\begin{aligned}\boldsymbol{Q} = \frac{q}{2} &\sum_{i\in\{1,2,\ldots,m-1\}\setminus\{t_1,t_4,\ldots,t_{3\ -2}\}} \boldsymbol{v}_{\pi(i)}\boldsymbol{v}_{\pi(i+1)} \\ &+\sum_{J\subseteq S}\ \sum_{i\in\bigcap_{\ \in}\ I} a_{i,J}\boldsymbol{v}_{\pi(i)}\left(\prod_{j\in J}\boldsymbol{v}'_j\right) \\ &+\frac{q}{2}\sum_{d=0}^{2\ -1}\left(\prod_{i=1}^{k}\boldsymbol{v}^{d}_{\pi(t_3\ )}(\boldsymbol{v}_0-\boldsymbol{v}_{\pi(t_3\ )})^{1-d}\ \right)\boldsymbol{Q}_d(\boldsymbol{v}_{\pi(1)},\boldsymbol{v}_{\pi(2)},\ldots,\boldsymbol{v}_{\pi(t_1)})\end{aligned} \tag{3}$$

where $\mathbf{d} = (d_1, d_2, \ldots, d_k) \in \mathbb{Z}_2^k$ is the binary representation of $d \in \mathbb{Z}_{2\ }$, $\boldsymbol{v}_i$, $i = 1,2,\ldots,m$, and $\boldsymbol{v}_0$ are defined in (1) and (2), respectively, $\boldsymbol{v}'_j = \boldsymbol{v}_{\pi(t\ )}$ for $j = 2,3,\ldots,3k$, $\boldsymbol{v}'_{-1} = \sum_{d=0}^{2\ -1}\boldsymbol{v}_{\pi\ (1)}\prod_{i=1}^{k}\boldsymbol{v}^{d}_{\pi(t_3\ )}(\boldsymbol{v}_0-\boldsymbol{v}_{\pi(t_3\ )})^{1-d}\ $, $\boldsymbol{v}'_1 = \sum_{d=0}^{2\ -1}\boldsymbol{v}_{\pi\ (t_1)}\prod_{i=1}^{k}\boldsymbol{v}^{d}_{\pi(t_3\ )}(\boldsymbol{v}_0-\boldsymbol{v}_{\pi(t_3\ )})^{1-d}$ where each π_d is a permutation of $\{1,2,\ldots,m\}$ with $\pi_d(i) = \pi(i)$ for $i = t_2, t_2+1, \ldots, m$, $S = \{-1,1,\ldots,3k\}$, $I_{3p-1} = \bigcup_{l=p+1}^{k}\{t_{3l-1}, t_{3l-1}+1, \ldots, t_{3l}\}$ for $p = 0,1,\ldots,k-1$, $I_{3p-2} = \bigcup_{l=p}^{k}\{t_{3l}, t_{3l}+1, \ldots, t_{3l+1}\}$ for $p = 1,2,\ldots,k$, $I_{3p} = \{1,2,\ldots,t_{3p-2}\}$ for $p = 1,2,\ldots,k$, and $1 = t_{-1} \le t_0 = t_1 = t_2 - 1 < t_3 < \cdots < t_{3k-2} = t_{3k-1} - 1 < t_{3k} \le m = t_{3k+1}$. For each d, if

$$\sum_{i=0}^{t_1-1}\boldsymbol{v}_{\pi(i)}\boldsymbol{v}_{\pi(i+1)} + \boldsymbol{Q}_d$$

is of the form

$$\sum_{i=0}^{t_1-1} \boldsymbol{v}_{\pi\ \ (i)}\boldsymbol{v}_{\pi\ \ (i+1)}$$

then for any choice of $g_i \in \mathbb{Z}_q$*,*

$$G = \left\{ \boldsymbol{Q} + \sum_{i=0}^{m} g_i \boldsymbol{v}_i + \frac{q}{2}\left(\sum_{j=0}^{k} d_j \boldsymbol{v}'_{3j-1}\right) : d_j \in \{0,1\} \right\}$$

is a Golay complementary set of size 2^{k+1}*.*

Proof. We have to show

$$\sum_{\boldsymbol{c}\in G}\sum_{i=0}^{n-1-u} \xi^{c\ +\ \ -c\ } = \sum_{i=0}^{n-1-u}\sum_{\boldsymbol{c}\in G} \xi^{c\ +\ \ -c\ } = 0, \quad \text{for } u \neq 0.$$

For an integer i, denote $j = i + u$ and let $(i_1, i_2, \ldots, i_m)$ and $(j_1, j_2, \ldots, j_m)$ be the binary representations of i and j, respectively.

(i) If $i_{\pi(t_3\ \ -1)} \neq j_{\pi(t_3\ \ -1)}$ for $p \in \{1, 2, \ldots, k\}$, then for any sequence $\boldsymbol{c} \in G$, there exists $\boldsymbol{c}' = \boldsymbol{c} + \frac{q}{2}\boldsymbol{v}_{\pi(t_3\ \ -1)} \in G$ such that

$$c_j - c_i - c'_j + c'_i = \frac{q}{2}(i_{\pi(t_3\ \ -1)} - j_{\pi(t_3\ \ -1)}) \equiv \frac{q}{2} \pmod{q}.$$

So we have

$$\xi^{c\ \ -c\ \ } / \xi^{c'\ -c'} = \xi^{q/2} = -1$$

which implies $\xi^{c\ \ -c\ \ } + \xi^{c'\ -c'} = 0$. Hence, we have

$$\sum_{\boldsymbol{c}\in G} \xi^{c\ +\ \ -c\ } = 0.$$

(ii) If $i_{\pi(t_3\ \ -1)} = j_{\pi(t_3\ \ -1)}$ for $p = 1, 2, \ldots, k$, we denote h as the smallest integer such that $i_{\pi(h)} \neq j_{\pi(h)}$ for $t_{3k-1} < h \leq t_{3k}$. Let $(i_1, i_2, \ldots, 1 - i_{\pi(h-1)}, \ldots, i_m)$ and $(j_1, j_2, \ldots, 1 - j_{\pi(h-1)}, \ldots, j_m)$ be the binary representations of i' and j', respectively. For any sequence $\boldsymbol{c} \in G$, we have

$$\begin{aligned} c_{i'} - c_i = {} & \frac{q}{2} i_{\pi(h-2)} + \frac{q}{2} i_{\pi(h)} \\ & + \sum_{J \subseteq \{-1, 2, \ldots, 3k-4\}} a_{h-1,J} \left(\prod_{p\in J} i_{\pi(t\ \ )} \right) \\ & + g_{\pi(h-1)} i_{\pi(h-1)}. \end{aligned}$$

Therefore,

$$\begin{aligned} c_j - c_i - c_{j'} + c_{i'} &= \frac{q}{2}\big(i_{\pi(h-2)} - j_{\pi(h-2)}\big) + \frac{q}{2}\big(i_{\pi(h)} - j_{\pi(h)}\big) \\ &\qquad + \sum_{J \subseteq \{-1,2,\ldots,3k-4\}} a_{h-1,J} \left(\prod_{p\in J} i_{\pi(t\ \ )} - j_{\pi(t\ \ )} \right) \\ &\qquad + g_{\pi(h-1)}\big(i_{\pi(h-1)} - j_{\pi(h-1)}\big) \\ &\equiv \frac{q}{2} \pmod{q} \end{aligned}$$

since $i_{\pi(h-2)} = j_{\pi(h-2)}, i_{\pi(h-1)} = j_{\pi(h-1)}, i_{\pi(h)} \neq j_{\pi(h)}$, and $i_{\pi(t\ \ )} = j_{\pi(t\ \ )}$ for $p \in \{-1, 2, \ldots, 3k-4\}$. So we have $\xi^{c\ \ -c\ \ }/\xi^{c\ \ '-c\ \ '} = \xi^{q/2} = -1$ which implies $\xi^{c\ \ -c\ \ } + \xi^{c\ \ '-c\ \ '} = 0$. Hence, we have

$$\sum_{\boldsymbol{c}\in G} \xi^{c\ \ +\ \ -c\ \ } + \xi^{c\ \ '+\ \ -c\ \ '} = 0.$$

We assume that $i_{\pi(r)} = j_{\pi(r)}$ for $r = t_{3l-1}, t_{3l-1}+1, \ldots, t_{3l}$ for $l = p+1, p+2, \ldots, k$. Let h' be the smallest integer such that $i_{\pi(h')} \neq j_{\pi(h')}$ for $t_{3p-1} < h' \leq t_{3p}$. Let $(i_1, i_2, \ldots, 1 - i_{\pi(h'-1)}, \ldots, i_m)$ and $(j_1, j_2, \ldots, 1 - j_{\pi(h'-1)}, \ldots, j_m)$ be the binary representations of i' and j', respectively. For any sequence $\boldsymbol{c} \in G$, we can also obtain $c_j - c_i - c_{j'} + c_{i'} = q/2$. Therefore,

$$\sum_{\boldsymbol{c}\in G} \xi^{c\ \ +\ \ -c\ \ } + \xi^{c\ \ '+\ \ -c\ \ '} = 0.$$

By induction, if $i_{\pi(p)} \neq j_{\pi(p)}$ for $p \in \{t_{3l-1}+1, t_{3l-1}+2, \ldots, t_{3l} : l \in \{0, 1, \ldots, k\}\}$, we have

$$\sum_{\boldsymbol{c}\in G} \xi^{c\ \ +\ \ -c\ \ } + \xi^{c\ \ '+\ \ -c\ \ '} = 0$$

for a particular i'.

(iii) For any $\boldsymbol{d} \in \mathbb{Z}_2^k$, let $(i_{\pi(t_3)}, i_{\pi(t_6)}, \ldots, i_{\pi(t_3\ \ )}) = \boldsymbol{d}$. If $i_{\pi\ \ (1)} \neq j_{\pi\ \ (1)}$, then for any sequence $\boldsymbol{c} \in G$, there exists $\boldsymbol{c}' = \boldsymbol{c} + \frac{q}{2}\boldsymbol{v}'_{-1} \in G$ such that

$$c_j - c_i - c'_j + c'_i = \frac{q}{2}\big(i_{\pi\ \ (1)} - j_{\pi\ \ (1)}\big) \equiv \frac{q}{2} \pmod{q}.$$

If $i_{\pi\ \ (1)} = j_{\pi\ \ (1)}$, we denote h as the smallest integer such that $i_{\pi\ \ (h)} \neq j_{\pi\ \ (h)}$ for $1 < h \leq t_1$. Let $(i_1, i_2, \ldots, 1 - i_{\pi\ \ (h-1)}, \ldots, i_m)$ and $(j_1, j_2, \ldots, 1 - j_{\pi\ \ (h-1)}, \ldots, j_m)$ be the binary representations of i' and j', respectively. We can also obtain

$$\sum_{\boldsymbol{c}\in G} \xi^{c\;+\quad -c} \;+ \xi^{c\;'+\quad -c\;'} = 0.$$

(iv) We assume that $i_{\pi(p)} = j_{\pi(p)}$ for $p{=}t_{3l-1}, t_{3l-1}{+}1, \ldots, t_{3l}$ for $l{=}0, 1, \ldots, k$. If $i_{\pi(h)} \neq j_{\pi(h)}$ for $h \in \{t_{3l}+1, t_{3l}+2, \ldots, t_{3l+1} : l{\in}\{1, 2, \ldots, k\}\}$, then by induction similar to (ii), we can also obtain

$$\sum_{\boldsymbol{c}\in G} \xi^{c\;+\quad -c} \;+ \xi^{c\;'+\quad -c\;'} = 0$$

for a particular i'.

From (i), (ii), (iii), and (iv), we have

$$\sum_{\boldsymbol{c}\in G} \sum_{i=0}^{n-1-u} \xi^{c\;+\quad -c} = 0, \quad \text{for } u \neq 0.$$

which completes the proof. □

Note that all the sequences of the same Golay complementary set mentioned in the above theorem lie in the same coset $\boldsymbol{Q} + \mathrm{RM}_q(1, m)$. Furthermore, from the fact that the PAPR of any sequence in a Golay complementary set of size 2^{k+1} is at most 2^{k+1}, we can obtain a corollary to this theorem.

Corollary 1. *If $\boldsymbol{Q}$ is the same as that defined in Theorem 1, then the coset $\boldsymbol{Q} + RM_q(1, m)$ has PAPR at most 2^{k+1}.*

The connection between Golay complementary sets and second-order cosets of $\mathrm{RM}_q(1, m)$ was provided in [3] and [4]. Theorem 1 and [8] both generalize the construction of complementary sets in [4] and give a connection between Golay complementary sets and cosets of $\mathrm{RM}_q(1, m)$ of arbitrary order. But, however, the upper bound in Corollary 1 is tighter than those in [4] and [8]. Also by taking $t_2 = t_3 = t_4 = m - k + 1, t_5 = t_6 = t_7 = m - k + 2, \ldots$, and $t_{3k-1} = t_{3k} = t_{3k+1} = m$ in Theorem 1, the results of [8] on Golay complementary sets can be regarded as a special case of Theorem 1. Moreover, if $|J| = 1$, Theorem 1 can be reduced to the results of [4] on Golay complementary sets. In [4], an example was provided: for length $n = 2^5 = 32$, let $\boldsymbol{Q} = \boldsymbol{v}_1\boldsymbol{v}_2 + \boldsymbol{v}_1\boldsymbol{v}_5 + \boldsymbol{v}_2\boldsymbol{v}_5 + \boldsymbol{v}_3\boldsymbol{v}_5 + \boldsymbol{v}_4\boldsymbol{v}_5$, and then the coset $\boldsymbol{Q} + \mathrm{RM}_q(1, 5)$ has PAPR equal to 3.449. However, the theorems in [4] and [8] both give a bound of only 8, while our theorem gives a tight bound of 4 by taking $k = 1$, $\pi(1) = 1$, $\pi(2) = 2$, $\pi(3) = 4$, $\pi(4) = 5$, $\pi(5) = 3$, $t_{-1} = 1$, $t_1 = 2$, $t_2 = 3$, $t_3 = 4$, $t_4 = 5$, and $\boldsymbol{Q}_d = \boldsymbol{0}$ for $d \in \mathbb{Z}_2$ in Theorem 1. Any codeword in this coset lies in a complementary set of size 4, so this coset has PAPR at most 4. Consider one more example: for $n = 2^m$ where m is even, let $\boldsymbol{Q} = \sum_{i=1, i\neq m/2}^{m-1} \boldsymbol{v}_i\boldsymbol{v}_{i+1}$, [4] and [8] both give a bound of $2^{m/2+1}$ which is very loose as m is large, while Theorem 1 gives a tight bound of 4 by taking $k = 1$, $t_1 = m/2$, $\pi(i) = i$ for $i = 1, 2, \ldots, m$, and $\boldsymbol{Q}_d = \boldsymbol{0}$ for $d \in \mathbb{Z}_2$ whether m is large or not. In [4] it was asked that "What is the strongest possible generalization of Theorem 12?" Theorem 1 could be a partial answer to this question.

5 Multiple-Shift Complementary Sets

We define a set of N length-n sequences $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_N$ to be called a multiple L-shift complementary set [12] if the following property is satisfied:

$$A_{\boldsymbol{c}_1}(u) + A_{\boldsymbol{c}_2}(u) + \cdots + A_{\boldsymbol{c}\;\;}(u) = \begin{cases} 0, & u \bmod L = 0 \\ Nn, & u = 0. \end{cases}$$

From the above definition, it is obvious that the family of multiple L-shift complementary sets is an extension of that of Golay complementary sets. When $N = 1$, such a sequence is called a multiple L-shift auto-orthogonal sequence. A multiple L-shift complementary set of size 2 is called a multiple L-shift complementary pair, and the sequences in this set are called multiple L-shift complementary sequences. In [13], it was shown that any multiple L-shift complementary sequence has PAPR at most $2L$. In this section, we provide a relationship between cosets of the first-order Reed-Muller codes and multiple L-shift complementary sets of length $n = 2^m$. We first denote the set of all sequences which lie in some multiple L-shift complementary sets of size N as $G_N^{(L)}$, and then we derive the following theorem about $G_N^{(L)}$ when both of L and N are a power of 2.

Theorem 2. *For $L = 2^l$ and $N = 2^k$, where $l, k \geq 0$, if S is a multiple L-shift complementary set of size N, then $S + a\boldsymbol{v}_i$ is also a multiple L-shift complementary set of size N, for $i = 1, 2, \ldots, l$, where $a \in \mathbb{Z}_q$ and $\boldsymbol{v}_i$'s are the same as those defined in Theorem 1. Moreover, if we denote $S_p = \Big\{\boldsymbol{a} + \frac{q}{2}\sum_{i=l-p+1}^{l} d_i\boldsymbol{v}_i : \boldsymbol{a} \in S,$ $d_i \in \{0, 1\}\Big\}$, then S_p is a multiple 2^{l-p}-shift complementary set of size 2^{k+p}, for $p = 1, 2, \ldots, l$. Therefore, $G_2^{(2\;)} \subseteq G_{2\;+1}^{(2\;-1)} \subseteq \cdots \subseteq G_{2\;+\;}^{(1)}$.*

Proof. (a) For a sequence $\boldsymbol{c} \in S$, we denote $\hat{\boldsymbol{c}} = \boldsymbol{c} + a\boldsymbol{v}_p \in S + a\boldsymbol{v}_p$ for $p \in \{1, 2, \ldots, l\}$ where $a \in \mathbb{Z}_q$. By definition, we have

$$\sum_{\boldsymbol{c} \in S} \sum_{i=0}^{n-1-u} \xi^{c\;+\;\;-c\;} = 0, \quad \text{for } u \bmod L = 0.$$

For an integer i, denote $j = i + u$ and let $(i_1, i_2, \ldots, i_m)$ and $(j_1, j_2, \ldots, j_m)$ be the binary representations of i and j, respectively. Since $u \mod 2^l = 0$, we have $i_h = j_h$ for $h = 1, 2, \ldots, l$. So

$$\hat{c}_{i+u} - \hat{c}_i = c_{i+u} + aj_p - c_i - ai_p = c_{i+u} - c_i.$$

Hence, we have

$$\begin{aligned}\sum_{\hat{\boldsymbol{c}}\in S+a\boldsymbol{v}} A_{\hat{\boldsymbol{c}}}(u) &= \sum_{\hat{\boldsymbol{c}}\in S+a\boldsymbol{v}} \sum_{i=0}^{n-1-u} \xi^{\hat{c}\ \ +\ \ -\hat{c}} \\ &= \sum_{\boldsymbol{c}\in S} \sum_{i=0}^{n-1-u} \xi^{c\ \ +\ \ -c} \\ &= 0, \quad \text{for } u \bmod L = 0\end{aligned}$$

which implies $S + a\boldsymbol{v}_p$ is a multiple L-shift complementary set of size N.

(b) We have to show

$$\sum_{\boldsymbol{c}\in S} A_{\boldsymbol{c}}(u) = \sum_{i=0}^{n-1-u} \sum_{\boldsymbol{c}\in S} \xi^{c\ \ +\ \ -c} = 0, \quad \text{for } u \bmod 2^{l-p} = 0$$

if $p = 1, 2, \ldots, l$. For an integer i, denote $j = i + u$ and let $(i_1, i_2, \ldots, i_m)$ and $(j_1, j_2, \ldots, j_m)$ be the binary representations of i and j, respectively. Since $u \bmod 2^{l-p} = 0$, we have $i_h = j_h$ for $h = 1, 2, \ldots, l - p$.

(i) If $i_h \neq j_h$ for $h \in \{l-p+1, l-p+2, \ldots, l\}$, then for any sequence $\boldsymbol{c} \in S_p$, there exists $\boldsymbol{c}' = \boldsymbol{c} + \frac{q}{2}\boldsymbol{v}_h \in S_p$ such that

$$c_j - c_i - c_j' + c_i' = \frac{q}{2}(i_h - j_h) \equiv \frac{q}{2} \pmod{q}.$$

So we have

$$\xi^{c\ \ -c\ \ } / \xi^{c'\ -c'} = \xi^{q/2} = -1$$

which implies $\xi^{c\ \ -c\ \ } + \xi^{c'\ -c'} = 0$. Hence, we have

$$\sum_{\boldsymbol{c}\in S} \xi^{c\ \ +\ \ -c} = 0.$$

(ii) If $i_h = j_h$ for $h = 1, 2, \ldots, l$, then $u = j - i \bmod 2^l = 0$. From (a), we have $\hat{S} = S + \frac{q}{2}\sum_{r=l-p+1}^{l} d_r\boldsymbol{v}_r$ is a multiple L-shift complementary set of size N for $d_r \in \{0, 1\}$ since S is a multiple L-shift complementary set of size N. Therefore,

$$\begin{aligned}\sum_{\boldsymbol{c}\in S} A_{\boldsymbol{c}}(u) &= \sum_{d\ _-\ _{+1}\in\{0,1\}} \cdots \sum_{d\ \in\{0,1\}} \sum_{\boldsymbol{c}\in\hat{S}} A_{\boldsymbol{c}}(u) \\ &= \sum_{d\ _-\ _{+1}\in\{0,1\}} \cdots \sum_{d\ \in\{0,1\}} 0 \\ &= 0, \quad \text{for } u \bmod 2^l = 0.\end{aligned}$$

From (i) and (ii), we have

$$\sum_{\boldsymbol{c}\in S} A_{\boldsymbol{c}}(u) = \sum_{i=0}^{n-1-u} \sum_{\boldsymbol{c}\in S} \xi^{c\ \ +\ \ -c} = 0, \quad \text{for } u \bmod 2^{l-p} = 0.$$

(c) From (b), it can be obtained that $G_2^{(2\ )} \subseteq G_{2\ +1}^{(2\ ^{-1})}$ by taking $p = 1$. Therefore, we have $G_2^{(2\ )} \subseteq G_{2\ +1}^{(2\ ^{-1})} \subseteq \cdots \subseteq G_{2\ +}^{(1)}$.

□

From Theorem 2, we know that $G_2^{(2\ )} \subseteq G_{2\ +}^{(1)}$, where $G_{2\ +}^{(1)}$ is the family of Golay complementary sets of size 2^{k+l}, so any sequence in a multiple 2^l-shift complementary set of size 2^k has PAPR at most 2^{k+l}. With a proof similar to that of Theorem 1, we can obtain the following theorem which can be used to construct multiple L-shift complementary sets from cosets of the first-order Reed-Muller codes.

Theorem 3. *Let $L = 2^l$ be a power of 2. Suppose all the parameters are the same as those defined in Theorem 1, except an additional condition. If $V_l \subseteq T$, where $V_l = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_l\}$ and $T = \{\boldsymbol{v}'_{-1}, \boldsymbol{v}'_2, \ldots, \boldsymbol{v}'_{3k-1}\}$, then for any choice of $g_i \in \mathbb{Z}_q$,*

$$G = \left\{ \boldsymbol{Q} + \sum_{i=0}^{m} g_i \boldsymbol{v}_i + \frac{q}{2} \left(\sum_{\boldsymbol{w} \in T \setminus V} d_{\boldsymbol{w}} \cdot \boldsymbol{w} \right) : d_{\boldsymbol{w}} \in \{0, 1\} \right\}$$

is a multiple L-shift complementary set of size 2^{k-l+1}.

In the above discussion, we only consider the case when L is a power of 2. However, we know that multiple L-shift complementary sequences exist for $L \neq 2^l$ by [13] and computer search results. Hence, we also provide two theorems to construct multiple L-shift complementary sequences when $L = 2^l + 1$ and $L = 2^l - 1$.

Theorem 4. *Let q be an even integer, for $L = 2^l + 1$ where l is a positive integer, and for any permutation π of the symbols $\{1, 2, \ldots, m\}$, if $\pi(m) \equiv \pi(m-1)$ (mod $2l$), let*

$$\begin{aligned} \boldsymbol{Q} = \frac{q}{2} \sum_{i=1}^{m-2} \boldsymbol{v}_{\pi(i)} \boldsymbol{v}_{\pi(i+1)} &+ a \boldsymbol{v}_{\pi(m-2)} \boldsymbol{v}_{\pi(m-1)} \boldsymbol{v}_{\pi(m)} \\ &+ b \boldsymbol{v}_{\pi(m-1)} \boldsymbol{v}_{\pi(m)} \end{aligned}$$

where a, $b \in \mathbb{Z}_q$. If $\pi(m) \not\equiv \pi(m-1)$ (mod $2l$), let

$$\begin{aligned} \boldsymbol{Q} = \frac{q}{2} \sum_{i=1}^{m-3} \boldsymbol{v}_{\pi(i)} \boldsymbol{v}_{\pi(i+1)} &+ \frac{q}{2} a \boldsymbol{v}_{\pi(m-2)} \boldsymbol{v}_{\pi(m-1)} \\ &+ \frac{q}{2} (1 \oplus a \oplus b) \boldsymbol{v}_{\pi(m-2)} \boldsymbol{v}_{\pi(m-1)} \boldsymbol{v}_{\pi(m)} \\ &+ \frac{q}{2} b \boldsymbol{v}_{\pi(m-2)} \boldsymbol{v}_{\pi(m)} + c \boldsymbol{v}_{\pi(m-1)} \boldsymbol{v}_{\pi(m)} \end{aligned}$$

where $a, b \in \mathbb{Z}_2$, $c \in \mathbb{Z}_q$, and $\oplus$ denotes the mod-2 addition. For any codeword $\boldsymbol{c} \in \boldsymbol{Q} + RM_q(1, m)$, $(\boldsymbol{c}, \boldsymbol{c} + \frac{q}{2} \boldsymbol{v}_{\pi(1)})$ is a pair of multiple L-shift complementary sequences. Hence, the coset $\boldsymbol{Q} + RM_q(1, m)$ has PAPR at most $2L$.

Theorem 5. *Let q be an even integer, for $L = 2^l - 1$ where $l \geq 3$, and for any permutation π of the symbols $\{1, 2, \ldots, m\}$, we denote*

$$\begin{aligned}\boldsymbol{Q} = \frac{q}{2} \sum_{i=1}^{m-2} \boldsymbol{v}_{\pi(i)} \boldsymbol{v}_{\pi(i+1)} + a \boldsymbol{v}_{\pi(m-2)} \boldsymbol{v}_{\pi(m-1)} \boldsymbol{v}_{\pi(m)} \\ + b \boldsymbol{v}_{\pi(m-1)} \boldsymbol{v}_{\pi(m)}\end{aligned}$$

where a, $b \in \mathbb{Z}_q$. For any codeword $\boldsymbol{c} \in \boldsymbol{Q} + RM_q(1, m)$, $(\boldsymbol{c}, \boldsymbol{c} + \frac{q}{2} \boldsymbol{v}_{\pi(1)})$ is a pair of multiple L-shift complementary sequences. Hence, the coset $\boldsymbol{Q} + RM_q(1, m)$ has PAPR at most $2L$.

6 Conclusion

In this paper, we have shown how to obtain complementary sets and multiple-shift complementary sets from first-order Reed-Muller codes of length $n = 2^m$. In addition, we also provide some results which can be used to derive multiple L-shift complementary sequences for some $L \neq 2^l$. Only the case that the length of the complementary set is a power of 2 is considered. However, complementary sets also exist for length $n \neq 2^m$. Work still in progress includes the study of complementary sets of length n which can be any integer and that of multiple L-shift complementary sets for any integer L.

References

1. van Nee, R., Prasad, R.: OFDM for Wireless Multimedia Communications. Artech House, Boston (2000)
2. van Nee, R.: OFDM Codes for Peak-to-Average Power Reduction and Error Correction. Proc. IEEE Globecom, London, U.K. (1996) 740–744
3. Davis, J.A., Jedwab, J.: Peak-to-Mean Power Control in OFDM, Golay Complementary Sequences, and Reed-Muller Codes. IEEE Trans. Inform. Theory **45** (1999) 2397–2417
4. Paterson, K.G.: Generalized Reed-Muller Codes and Power Control in OFDM Modulation. IEEE Trans. Inform. Theory **46** (2000) 104–120
5. Paterson, K.G., Tarokh, V.: On the Existence and Construction of Good Codes with Low Peak-to-Average Power Ratios. IEEE Trans. Inform. Theory **45** (2000) 1974–1987
6. Chiu, M.-C., Chen, H.-S.: Reduction of the Peak to Average Power Ratio in OFDM Systems with Convolutional Codes. Proc. IEEE Int. Symp. Inform. Theory, Lausanne, Switzerland (2002) 246
7. Chen, C.-Y., Wang, C.-H., Chao, C.-C.: Convolutional Codes for Peak-to-Average Power Ratio Reduction in OFDM. Proc. IEEE Int. Symp. Inform. Theory, Yokohama, Japan (2003) 5
8. Schmidt, K.-U., Finger, A.: New Codes for OFDM with Low PMEPR. Proc. IEEE Int. Symp. Inform. Theory, Adelaide, Australia (2005) 1136–1140
9. Golay, M.J.E.: Complementary Series. IRE Trans. Inform. Theory **IT-7** (1961) 82–87

10. Tseng, C.-C., Liu, C.L.: Complementary Sets of Sequences. IEEE Trans. Inform. Theory **IT-18** (1972) 644–652
11. Taki, Y., Miyakawa, H., Hatori, M., Namba, S.: Even-Shift Orthogonal Sequences. IEEE Trans. Inform. Theory **IT-15** (1969) 295–300
12. Suehiro, N., Hatori, M.: N-Shift Cross-Orthogonal Sequences. IEEE Trans. Inform. Theory **34** (1988) 143–146
13. Xin, Y., Fair, I.J.: Multiple-Shift Complementary Sequences and Their Peak-to-Average Power Ratio Values. Proc. IEEE Int. Symp. Inform. Theory, Chicago, IL (2004) 121
14. Boyd, S.: Multitone Signals with Low Crest Factor. IEEE Trans. Circuits Syst. **CAS-33** (1986) 1018–1022
15. Popović, B.M.: Synthesis of Power Efficient Multitone Signals with Flat Amplitude Spectrum. IEEE Trans. Commun. **39** (1991) 1031–1033
16. Lin, S., Costello, D.J.Jr.: Error Control Coding. 2nd edn. Pearson Prentice Hall, Upper Saddle River, NJ (2004)

Hadamard Codes of Length $2^t s$ (s Odd). Rank and Kernel[⋆]

Kevin T. Phelps[1], Josep Rifà[2], and Mercè Villanueva[2]

[1] Dept. of Mathematics and Statistics, Auburn University,
Auburn, Al 36849-5307
phelpkt@auburn.edu

[2] Dept. of Information and Communications Engineering,
Universitat Autònoma de Barcelona, Spain
{josep.rifa, merce.villanueva}@autonoma.edu

Abstract. The rank, r, and the dimension of the kernel, k, for binary Hadamard codes of length 2^t were studied in [12], constructing such codes for all possible pairs (r, k). Now, we will focus on Hadamard codes of length $2^t \cdot s$, $s > 1$ odd. As long as there exists a Hadamard code of length $4s$, constructions of Hadamard codes of length $n = 2^t \cdot s$ ($t \geq 3$) with any rank, $r \in \{4s + t - 3, \ldots, n/2\}$, and any possible dimension of the kernel, $k \in \{1, \ldots, t - 1\}$, are given.

1 Introduction

Let $\mathbb{F}^n$ denote the set of all binary vectors of length n. The Hamming distance between two vectors $x, y \in \mathbb{F}^n$, denoted by $d(x, y)$, is the number of coordinates in which x and y differ. The Hamming weight of x is given by $wt(x) = d(x, \mathbf{0})$, where $\mathbf{0}$ is the all-zero vector. The support of a vector $x \in \mathbb{F}^n$ is the set of nonzero coordinate positions of x and is denoted by $supp(x)$.

A *(binary)* (n, M, d)*-code* is a subset, C, of $\mathbb{F}^n$ such that $|C| = M$ and $d(c_1, c_2) \geq d$ for all pairs $c_1, c_2 \in C$. The elements of a code are called *codewords* and d is called *minimum distance*. We will write $\langle C \rangle$ to denote the binary linear span of C.

Two codes $C_1, C_2 \in \mathbb{F}^n$ are *equivalent* if there exists a vector $a \in \mathbb{F}^n$ and a permutation π such that $C_2 = \{a + \pi(c) \mid c \in C_1\}$. Two structural properties of nonlinear codes are the rank and kernel. The *rank* of a binary code C, $r = rank(C)$, is simply the dimension of the linear span of C. By the binary orthogonal code of the nonlinear code C, denoted by $C^\perp$, we mean the dual of the subspace spanned by C having dimension $n - r$. The *kernel* of a binary code C is defined as $K(C) = \{x \in \mathbb{F}^n \mid x + C = C\}$. If the zero word is in C, then $K(C)$ is a linear subspace of C. In general, C can be written as the union of cosets of $K(C)$ and $K(C)$ is the largest such linear code for which this is true (see [3]). We will denote the dimension of the kernel of C by $k = ker(C)$.

[⋆] Research partially supported by CICYT Grants TIC2003-08604-C04-01, TIC2003-02041 and by Catalan DURSI Grant 2001SGR 00219.

M. Fossorier et al. (Eds.): AAECC 2006, LNCS 3857, pp. 328–337, 2006.

A *Hadamard matrix* H of size n is an $n \times n$ matrix of $+1$'s and -1's such that $HH^T = nI$, where I is the $n \times n$ identity matrix. In other words, the real inner product of any row with itself is n and distinct rows are orthogonal. Since $nH^{-1} = H^T$, we also have $H^T H = nI$, thus the columns have the same properties and the transpose of any Hadamard matrix, H, is also a Hadamard matrix, which is not necessary equivalent to H. We know that if a Hadamard matrix H of size n exists, then n is 1, 2 or a multiple of four (see [4, 8]).

Two Hadamard matrices are *equivalent* if one can be obtained from the other by permuting rows and/or columns and multiplying rows and/or columns by -1. We can change the first row and column of H into $+1$'s and we obtain an equivalent Hadamard matrix which is called *normalized*.

From now on, we will use H' to denote a normalized Hadamard matrix of size n. If $+1$'s are replaced by 0's and -1's by 1's, H' is changed into a *(binary) Hadamard matrix* $c(H')$. Since the rows of H' are orthogonal, any two rows of $c(H')$ agree in $n/2$ places and differ in $n/2$ places, and so have Hamming distance $n/2$ apart. The binary $(n, 2n, n/2)$-code consisting of the rows of $c(H')$ and their complements is called a *(binary) Hadamard code* (see [8]) and we will use H to denote it.

Now, we recall some results, that we will need, about the rank and the kernel dimension of Hadamard codes constructed using the *Kronecker product construction*. That is, if $H' = (h_{ij})$ is any $n \times n$ Hadamard matrix, and $B_1, B_2, \dots, B_n$ are any $k \times k$ Hadamard matrices, then the following matrix

$$H' \otimes [B_1, B_2, \dots, B_n] = \begin{pmatrix} h_{11}B_1 & h_{12}B_1 & \cdots & h_{1n}B_1 \\ h_{21}B_2 & h_{22}B_2 & \cdots & h_{2n}B_2 \\ \vdots & \vdots & \vdots & \vdots \\ h_{n1}B_n & h_{n2}B_n & \cdots & h_{nn}B_n \end{pmatrix}$$

is a $nk \times nk$ Hadamard matrix, [1]. If $B_1 = B_2 = \cdots = B_n = B$, we write $H' \otimes [B_1, B_2, \dots, B_n] = H' \otimes B$.

Let S be the Hadamard matrix $\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$. Starting from a Hadamard matrix H' of size n, we can construct a Hadamard matrix of size $2n$, $S \otimes H'$. So, if there exists a Hadamard matrix of size $4s$, where s is an odd number, there exists a Hadamard matrix of size $2^t \cdot s$, $\forall t \geq 3$. Currently, the first size n for which it is not known whether or not there exists a Hadamard matrix is $n = 668$, [5].

Lemma 1. *[11, 12] Let H' be a Hadamard matrix and H its Hadamard code. The kernel dimension of the corresponding Hadamard code of $S \otimes H'$ is $ker(H) + 1$ and the rank is $rank(H) + 1$.*

Lemma 2. *[1] Let H'_1 ,H'_2 be two Hadamard matrices and H_1, H_2 their Hadamard codes. The rank of the corresponding Hadamard code of $S \otimes [H'_1, H'_2]$ is $rank(H_1) + rank(H_2) + 1 - dim(\langle H_1 \rangle \cap \langle H_2 \rangle)$ or, equivalently, $dim(\langle H_1 \cup H_2 \rangle) + 1$.*

Lemma 3. *[11, 12] Let H'_1, H'_2 be two Hadamard matrices. Let H_1, H_2 be their Hadamard codes and $K(H_1)$, $K(H_2)$ their kernels. If for all v, $H_1 \neq v + H_2$,* *then the kernel K of the corresponding Hadamard code of $S \otimes [H_1', H_2']$ is $K = \{(x,x) \mid x \in K(H_1) \cap K(H_2)\}$.*

The rank and the dimension of the kernel for Hadamard codes of length a power of two were studied in [1, 6, 9, 10, 11, 12]. In [11, 12], exact lower and upper bounds for the rank and dimension of the kernel of a Hadamard code of length $n = 2^t$ were established. Moreover, Hadamard codes for all possible ranks and dimension of kernels between these bounds were constructed.

In this paper, we will focus on the rank and the dimension of the kernel for Hadamard codes of length $n = 2^t \cdot s$, $s > 1$ odd. The paper is arranged as follows. In Section 2, we establish that Hadamard codes of length $2^t \cdot s$ ($s > 1$ odd) have kernel of dimension k in the range $\{1, \ldots, t-1\}$. In Section 3, we prove that if there exists a Hadamard code of length $4s$ (s odd), we can construct Hadamard codes of length $2^t \cdot s$ with rank r, for all r between $4s+t-3$ and the upper bound $n/2$. Finally, in Section 4, we construct Hadamard codes of length $2^t \cdot s$ with parameters (r, k) for all of the above values, as long as there exists a Hadamard code of length $4s$ (s odd).

2 Dimension of the Kernel of Hadamard Codes

In [11, 12], it was proved that the Hadamard codes of length $n = 2^t$ have kernels of dimension $k \in \{1, 2, \ldots, t-1, t+1\}$. For these Hadamard codes of length 2^t, there always exists the linear one of dimension $t+1$, denoted by S_t. We can assume S_t is generated by the binary vectors $\mathbf{1}, v_1, v_2, \ldots, v_t$ of length 2^t, where the vectors v_i, $\forall i \in \{1, \ldots, t\}$, are as follows:

$$\begin{aligned}
v_1 &= (\underbrace{1,1,1,1,1,\ldots,1}_{n/2},\underbrace{0,0,0,0,0,\ldots,0}_{n/2}),\\
v_2 &= (\underbrace{1,1,\ldots,1}_{n/4},\underbrace{0,\ldots,0}_{n/4},\underbrace{1,\ldots,1}_{n/4},\underbrace{0,\ldots,0}_{n/4}),\\
&\vdots\\
v_t &= (\underbrace{1,\ldots,1}_{n/2},\underbrace{0,\ldots,0}_{n/2},\underbrace{1,\ldots,1}_{n/2},\ldots,\underbrace{0,\ldots,0}_{n/2}).
\end{aligned} \tag{1}$$

In general, if $n = 2^t \cdot s$, we can also consider the vectors $\mathbf{1}, v_1, v_2, \ldots, v_t$ of length $n = 2^t \cdot s$ constructed in the same way. It is known that not always $\langle \mathbf{1}, v_1, v_2, \ldots, v_t \rangle \subseteq \langle H \rangle$ [2], however if a Hadamard code H has $ker(H) = k$, it is straightforward to see that the kernel is generated by k (independent) vectors from $\mathbf{1}, v_1, v_2, \ldots, v_t$, so we can assume that $K(H) = \langle \mathbf{1}, v_1, v_2, \ldots, v_{k-1} \rangle$, up to equivalence.

Now, we will show that we can construct a new Hadamard code by puncturing some coordinates of a given Hadamard code. Let H be a Hadamard code of length $n = 2^t \cdot s$ ($t \geq 2$), where $s \neq 1$ is an odd number, let $c(H')$ be the corresponding binary Hadamard matrix and let

$$S = \{i \,|\, x_i \neq 0, \ \forall x \in K(H) \cap c(H')\}.$$

If $k > 1$, then $|S| < n$ and we can construct the multi-punctured code L, obtained from H by deleting all the coordinates out of S and avoiding repeated vectors. The length of L is $|S| = 2^{t-(k-1)} \cdot s$.

Lemma 4. *Let H be a Hadamard code of length $n = 2^t \cdot s$ ($t \geq 2$), where $s \neq 1$ is an odd number. If the dimension of the kernel of H is $k > 1$, then $t \geq 3$.*

Proof. Let H' be the corresponding normalized Hadamard matrix and $c(H')$ be the binary Hadamard matrix. If $k > 1$, there exists $v \in K(H) \cap c(H')$ and $v \neq \mathbf{0}$. Let $x, y \in c(H') \backslash \{\mathbf{0}\}$, such that $x \neq y + v$, which coincide in α coordinates of value 1 in *supp*(v) and coincide in β coordinates of value 1 in *supp*$(\mathbf{1} + v)$. We know that any two vectors in $c(H') \backslash \{\mathbf{0}\}$ have $n/4$ coordinates in which they share ones. Then, $\alpha + \beta = n/4$. Since $v \in K(H)$, we have that $\mathbf{1} + v \in K(H)$ and $y + \mathbf{1} + v \in H$. There are $\alpha + (n/4 - \beta) = n/4$ coordinates where x and $y + \mathbf{1} + v$ share ones. Hence, $\alpha = \beta = n/8$ and so $t \geq 3$.

Lemma 5. *Let H be a Hadamard code of length $n = 2^t \cdot s$ ($t \geq 2$), where $s \neq 1$ is an odd number. If the dimension of the kernel of H is $k \geq 1$, then $t \geq k + 1$.*

Proof. For $k = 1$ it is trivial and for $k = 2$ it follows directly from Lemma 4. For the general case, we will use induction, so we assume the result is true for $k > 1$ and we will prove it for $k + 1$.

Let $v \in K(H) \backslash \{\mathbf{0}, \mathbf{1}\}$. Take H and puncture the code eliminating all the coordinates where the vector v is zero. There are exactly two copies of each vector, since for any $x \in H$, the vector $x + \mathbf{1} + v \in H$ is the same as x in the support of v. Let L be the new code without repeated vectors. The code L has length $n = 2^{t-1} \cdot s$. The dimension of the kernel is greater or equal to $k - 1$, since the independent vectors in the kernel of H are still independent in the kernel of L, with the possible exception of the vector v, which coincides with $\mathbf{1}$.

We will see that L is a Hadamard code since $t - 1 \geq 2$, so $4|n$ and, moreover, if we take any two vectors in H at distance apart $n/2$, then the restrictions of these two vectors to the support of v are at distance apart $n/4$. Let $x, y \in c(H') \backslash \{\mathbf{0}\}$, such that $x \neq y + v$, which coincide in α coordinates of value 1 in *supp*(v) and coincide in β coordinates of value 1 in *supp*$(\mathbf{1} + v)$. We know by the proof of Lemma 4 that $\alpha = \beta = n/8$. The vectors x and y coincide in γ coordinates of value 0 in *supp*(v) and coincide in δ coordinates of value 0 in *supp*$(\mathbf{1} + v)$. Moreover, $\alpha + \delta = \beta + \gamma = n/4$ [1, theorem 7.2.2], so $\gamma = \delta = n/8$. The distance from x to y restricted to the support of v is $n/2 - \alpha - \gamma = n/4$. Finally, the code L is a Hadamard code of length $2^{t-1} \cdot s$ and dimension of the kernel greater or equal to $k - 1$, so by using the induction hypothesis $t - 1 \geq k$ or, equivalently, $t \geq k + 1$.

Theorem 1. *A Hadamard code of length $n = 2^t \cdot s$ ($t \geq 2$), where $s \neq 1$ is an odd number, has kernel of dimension $k \in \{1, 2, \ldots, t - 1\}$.*

Proof. For a Hadamard code, the minimum dimension of the kernel is 1, since the complement of any codeword is in the code. By Lemma 5, we have that $k \leq t - 1$.

3 Rank of Hadamard Codes

In [11, 12], it was proved that there exist a Hadamard code of length $n = 2^t$ for any possible rank, $r \in \{t+1, \dots, n/2\}$. For Hadamard codes of length $4s$ and $8s$, where $s \neq 1$ is an odd number, the next result is well-known.

Proposition 1. *[1] The Hadamard codes of length $4s$ and $8s$, where s is an odd number, have rank $4s-1$ and $4s$, respectively.*

It is also a well-known result that for any binary matrix A of size $n \times n$, in which all the rows are nonzero and distinct, the rank is lower bounded by $log_2 n$ with equality if and only if A is a (binary) Hadamard matrix whose associate Hadamard code is linear, [7]. For Hadamard codes of length $n = 2^t \cdot s$, where $s \neq 1$ is an odd number, this logarithm does not give us the exact lower bound. However, the following theorem shows the existence of these Hadamard codes with rank r, for all r between $4s + t - 3$ and the exact upper bound $n/2$.

Lemma 6. *Let H' be a Hadamard matrix of size $n = 2^t \cdot s$ $(t \geq 3)$ and H its Hadamard code. The minimum weight in the linear span $\langle H \rangle$ is greater or equal to four.*

Proof. The minimum weight in $H^\perp$ is at least 3, since H does not contain equal columns. As $H \subset H^\perp$, we have that $\langle H \rangle \subset H^\perp$ and the weight of the vectors in H is even. So, the minimum weight in $\langle H \rangle$ is greater or equal to four.

Theorem 2. *If there exists a Hadamard code of length $4s$, where $s \neq 1$ is an odd number, for all $t \geq 3$ there exists a Hadamard code of length $n = 2^t \cdot s$ with rank r, $\forall r \in \{4s + t - 3, \dots, n/2\}$.*

Proof. By Proposition 1 and Lemma 1, the corresponding Hadamard code of $S \otimes H'$, where H' is the Hadamard matrix of size $4s$, has length $8s$ and rank $4s$, so the result is true when $t = 3$. Let K' be a Hadamard matrix of size $n = 2^{t-1} \cdot s$ $(t \geq 4)$ and K its Hadamard code with $rank(K) = r$ and such that the (last) r column vectors of K are the independent ones. We will see how to construct Hadamard matrices of size $2^t \cdot s$ with different ranks.

First, the rank of the corresponding Hadamard code of $S \otimes K'$ is $r + 1$, by Lemma 1. Now, consider $L'_1 = \pi_{0,1}(K')$ the matrix formed by switching columns n and $n-1$ in K', (i.e. $\pi_{0,1} = (n-1, n)$) and let L_1 be its Hadamard code. The independent vectors in $\langle K \cup L_1 \rangle$ include those in K as well as, the vector $u_{r-1,r} = (0, \dots, 0, 1, 1) = (x, 01) + (x, 10)$ for some $(x, 01) \in K$ and $(x, 10) \in L_1$. By Lemma 6, $u_{r-1,r}$ is independent from $\langle K \rangle$. Hence the rank of the corresponding Hadamard code of $S \otimes [K', L'_1]$ is $dim(\langle K \cup L_1 \rangle) + 1 = r + 2$.

We can continue in this way taking $L'_2 = \pi_{0,2}(L'_1)$ the matrix formed by switching columns n and $n-2$ in L'_1, $\pi_{0,2} = (n-2, n)$ or, equivalently, by a cyclic shift $L'_2 = (n, n-1, n-2)K'$. The independent vectors in $\langle K \cup L_2 \rangle$ include those in K and, moreover, vectors $u_{r-1,r} = (0, \dots, 0, 1, 1)$ and $u_{r-2,r} = (0, \dots, 0, 1, 0, 1)$ which are independent from $\langle K \rangle$ by Lemma 6. There exist some vectors in $\langle K \rangle$ with different values in the last three coordinates (e.g. $(x, 001), (x, 010), (x, 100)$), such that adding pairwise these vectors to the corresponding vectors in L_2 we find vectors $u_{r-1,r}$ and $u_{r-2,r}$. So, the rank of the corresponding Hadamard code of $S \otimes [K', L_2']$ is $dim(\langle K \cup L_2 \rangle) + 1 = r + 3$.

In the same way, we can form matrices $L_i' = \pi_{0,i}(L_{i-1}')$ or equivalently by taking cyclic shifts $L_i' = (n, n-1, \ldots, n-i)K'$, of $i+1 \leq r$ independent columns in K'. Hence if you assume we have a Hadamard code of length $n = 2^{t-1} \cdot s$ and rank $r \in \{4s+t-4, \ldots, 2^{t-2} \cdot s\}$ we can construct new Hadamard codes of length $2^t \cdot s$ and rank from $r+1$ to $2r$ which, in general, gives us Hadamard codes of rank from $4s+t-3$ to $2^{t-1} \cdot s$.

4 Hadamard Codes with a Given Rank and Dimension of the Kernel

First, for Hadamard codes of any length, we will give an upper bound on the rank, in terms of the dimension of the kernel. This result is a generalization of a result in [11, 12] for length a power of two.

Proposition 2. *A (nonlinear) Hadamard code of length $n = 2^t{\cdot}s$ ($t \geq 3$), where s is an odd number, with rank r and a kernel of dimension k satisfies*

$$r \leq \begin{cases} 2^{t+1-k}{\cdot}s + k - 1 & \textit{if}\, 3 \leq k \leq t-1 \\ 2^{t-1}{\cdot}s & \textit{if}\, 1 \leq k \leq 2 \end{cases}$$

Proof. Let H be a Hadamard code of length $n = 2^t{\cdot}s$ with rank r and a kernel of dimension k. We know that $K(H)$ is the largest linear subspace into H such that H can be written as the union of cosets of $K(H)$ and that the cosets of $K(H)$ form a partition of H. There are $2^{t+1-k} \cdot s$ cosets in H. When each coset has an independent vector, the rank is maximum, so $r \leq 2^{t+1-k}{\cdot}s + k - 1$. For $k = 1$ and $k = 2$, $2^{t+1-k}{\cdot}s + k - 1 > 2^{t-1}{\cdot}s$, but since these codes are self-orthogonal, $r \leq n/2 = 2^{t-1}{\cdot}s$ [1], so in these two cases the upper bound is $2^{t-1}{\cdot}s$.

In [11, 12] it was proved that apart from the linear Hadamard code, we can construct Hadamard codes of length 2^t ($t > 4$) with rank r and kernel of dimension k, for all the possible pairs (r, k) between the theoretical lower and upper bounds:

$$\begin{cases} t+2 \leq r \leq 2^{t+1-k} + k - 1 & \text{if } 3 \leq k \leq t-1 \\ t+3 \leq r \leq 2^{t-1} & \text{if } 1 \leq k \leq 2 \end{cases} \tag{2}$$

For example, in Table 1, we show the different ranks and dimension of the kernels for which there exist a Hadamard code of length 32. For length 16, apart from the linear one, there exist four more with each one of the parameters (r, k) $\in \{(6,3), (7,2), (8,2), (8,1)\}$.

Now, we will also show how to construct Hadamard codes of length $2^t \cdot s$ ($t \geq 3$), where $s \neq 1$ is an odd number, with any rank between $4s+t-3$ and the upper bound (see Proposition 2), given any possible dimension of the kernel (see Theorem 1). These constructions will work, as long as there exists a Hadamard code of length $4s$, which will have rank $4s-1$ and dimension of the kernel 1, by Proposition 1 and Theorem 1, respectively.

Table 1. Dimension of the kernels and ranks of Hadamard codes of length $n = 32$

	rank(C)										
ker(C)	6	7	8	9	10	11	12	13	14	15	16
6	⋆										
4		⋆									
3		●	⋆	⋆	●						
2			⋆	⋆	⋆	⋄	⋄	⋄	⋄	⋄	⋄
1			●	⋆	∗	∗	∗	∗	∗	∗	∗

Proposition 3. *If there exists a Hadamard code of length* $4s$*, where* $s \neq 1$ *is an odd number, there exist two (non-equivalent) Hadamard codes of length* $8s$ *with kernel of dimensions 1 and 2, and both with rank* $4s$*.*

Proof. By Proposition 1 and Theorem 1, a Hadamard code H of length $4s$ has rank $4s - 1$ and dimension of the kernel 1. By Lemma 1, the corresponding Hadamard code of $S \otimes H'$ has length $8s$, rank $4s$ and kernel of dimension 2. Now, consider the Hadamard matrix $L' = \pi(H')$, where π is any transposition, and L its Hadamard code. Since $\langle H \rangle$ is generated by all the vectors of weight 2, $\langle H \cup L \rangle = \langle H \rangle$ and the corresponding Hadamard code of $S \otimes [H', \pi(H')]$ has also rank $4s$ by Lemma 2, and kernel of dimension 1 by Lemma 3.

Lemma 7. *Given a nonlinear Hadamard code* H *of length* $2^t \cdot s$ $(t \geq 3)$ *with rank* r *and kernel of dimension* k*, there exist Hadamard codes of length* $n = 2^{t+1} \cdot s$ *with rank* $r + 1 + \delta$ *and kernel of dimension* $k + 1 - \delta$*,* $\forall \delta \in \{0, \ldots, k\}$*.*

Proof. By Lemma 1 the corresponding Hadamard code of $S \otimes H'$ has rank $r + 1$ and kernel of dimension $k + 1$. By the same argument as in the proof of Proposition 2, for each $\delta \in \{1, \ldots, k\}$ there exists a permutation π_δ such that the corresponding Hadamard code of $C' = S \otimes [H', \pi_\delta(H')]$ has rank $r + 1 + \delta$. These permutations represent a cyclic shift of $\delta + 1$ independent columns in H'. We can choose these columns in the following way. If $\delta = 1$, π_1 is a transposition that fixes $K(H)$. Note that π_1 always exists since we stated that H is a nonlinear code and, in this case, following the notation in (1), we can take as π_1 the transposition of two coordinates with the same value in all the vectors v_i. Hence, in the case $\delta = 1$, by Lemma 3, the Hadamard code C has kernel of dimension $k + 1 - 1$. If $\delta \in \{2, \ldots, k\}$, π_δ effects $\delta - 1$ vectors in $K(H)$, so C has kernel of dimension $k - (\delta - 1) = k + 1 - \delta$.

Lemma 8. *If there exists a Hadamard code of length* $4s$*, where* $s \neq 1$ *is an odd number, there exist Hadamard codes of length* $n = 2^t \cdot s$ $(t \geq 3)$ *with kernel of dimension 1 and rank* r*,* $\forall r \in \{4s + 2(t - 3), \ldots, n/2\}$*.*

Proof. By Proposition 3, it is true for $t = 3$. Let H be a Hadamard code of length $2^{t-1} \cdot s$, with rank $4s + 2(t - 4)$ and kernel of dimension 1. By Lemma 7 $(\delta = 1)$, there exists a Hadamard code of length $n = 2^t \cdot s$, with rank $4s + 2(t - 4) + 2 = 4s + 2(t - 3)$ and kernel of dimension 1. The result follows using Lemma 3 and the same argument as in the proof of Proposition 2.

Lemma 9. *If there exists a Hadamard code of length* $4s$*, where* $s \neq 1$ *is an odd number, there exist Hadamard codes of length* $n = 2^t \cdot s$ $(t \geq 4)$ *with kernel of dimension 2 and rank* r*,* $\forall r \in \{2^{t-2} \cdot s + 3, \ldots, n/2\}$*.*

Proof. The Hadamard codes considered in Lemma 8 have a kernel of dimension 1 and are constructed using the Kronecker product. In the corresponding Hadamard matrix H', after a normalization, we can always assume there exists a column c with all the coordinates one and so another column d with half the coordinates equal to one and the other half equal to zero. If we take the transposed matrix, we obtain a new Hadamard matrix L'. The corresponding Hadamard code L will have kernel of dimension at least two. The two independent rows c^T and d^T are in the kernel, because of the Kronecker product construction.

From Proposition 2 we know there does not exist any Hadamard code with dimension of the kernel greater than two and rank greater or equal to $2^{t-2} \cdot s + 3$. Hence, when the rank has these values we conclude that the dimension of the kernel is 2.

Lemma 10. *If there exists a Hadamard code of length* $4s$*, where* $s \neq 1$ *is an odd number, there exists a Hadamard code of length* $n = 2^t \cdot s$ $(t \geq 2)$ *with kernel of dimension 1 and minimum rank* $4s + t - 3$*.*

Proof. Let H_t and L_t be Hadamard codes of length $2^t \cdot s$ with rank $4s + t - 3$ and kernel of dimension 1 and 2, respectively. We know this result is true for $t = 2$, so there exists H_2. By Proposition 3, there also exist H_3 and L_3.

We suppose there exists H_{t-1} $(t \geq 4)$, which has rank $4s + t - 4$. By Lemma 7 and from a Hadamard code H_{t-2} with rank $4s + t - 5$, we can construct L_{t-1} which will have rank $4s+t-4$ and kernel of dimension 2. Then, the corresponding Hadamard code of $S \otimes [H_{t-1}, L_{t-1}]$ will have length $2^t \cdot s$, rank $4s + t - 3$ and kernel of dimension 1, by Lemmas 2 and 3. So there exists $H_t = S \otimes [H_{t-1}, L_{t-1}]$.

Lemma 11. *If there exists a Hadamard code of length* $4s$*, where* $s \neq 1$ *is an odd number, there exists a Hadamard code of length* $n = 2^t \cdot s$ $(t \geq 4)$ *with rank* $r = 2^{t-2} \cdot s + 2$ *and dimension of the kernel* $k = 3$*.*

Proof. Let H be a Hadamard code (which exists by Lemma 7) of length $n = 2^t \cdot s$ $(t \geq 4)$ with rank $2^{t-2} \cdot s + 1$ and dimension of the kernel 3. Assume (after a coordinate permutation if it is needed) the basis vectors for the kernel $K = K(H)$ are $\mathbf{1}, v_1, v_2$, as they are defined in equation (1). Let L be a code $H \backslash (K+x) \cup (K+x+v_1 v_2)$, where $x \in H \backslash K$ and $v_1 v_2 = (11\ldots1, 00\ldots0)$ with 1's in the first $n/4$ coordinates. We claim that L is a Hadamard code. It is clear that L has also $2n$ codewords, since $(K + x + v_1 v_2) \cap (K + y) = \emptyset$, $\forall y \in H \backslash (K + x)$. To prove that the minimum distance between codewords is $n/2$, it suffices to show that this is the minimum weight for the words of type $K + x + y + v_1 v_2$. Let $z = (z_0, z_1, z_2, z_3)$ be any word of $K + x + y + v_1 v_2$ of minimum weight $n/2$, where in each z_i there are $n/4$ coordinates of z. Since $y \notin K + x$, we have that $z \notin K = \langle 1, v_1, v_2 \rangle$. Each z_i $(i \in 0, 1, 2, 3)$ and $z_0 + 11\ldots1$ have weight $n/8$, so $z + v_1 v_2$ has weight $n/2$. The kernel of L is K and $rank(L) = rank(H) + 1$

because $\langle L\rangle = \langle H, v_1v_2\rangle$. Hence, code L is a Hadamard code of length $n = 2^t \cdot s$ ($t \geq 4$) with rank $2^{t-2} \cdot s + 2$ and dimension of the kernel 3.

Finally, using these last lemmas, we have established the next theorem.

Theorem 3. *If there exists a Hadamard code of length $4s$, where $s \neq 1$ is an odd number, there exist Hadamard codes of length $n = 2^t{\cdot}s$ ($t \geq 3$) with kernel of dimension k and rank r for all r such that*

$$4s + t - 3 \leq r \leq \begin{cases} 2^{t+1-k}{\cdot}s + k - 1 & \mathit{if}\, 3 \leq k \leq t-1 \\ 2^{t-1}{\cdot}s & \mathit{if}\, 1 \leq k \leq 2 \end{cases} \qquad (3)$$

For example, since we know that there exists a Hadamard code of length $n = 12$ with rank 11 and kernel of dimension 1, we can construct two (non-equivalent) Hadamard codes of length $n = 24$ with kernel of dimension 1 and 2, respectively. The corresponding Hadamard matrices H_1 and H_2 of these codes, constructed using Proposition 3, are (in hexadecimal notation):

H_1=[000000, $FFF000, C8B374, 4F1B0E, 64D9B2, 3A9C56, 8EA8EA, 6A66A6, 56C56C, 3FFC0, D252DA, B0EB0E, 1C7E38, 59A59A, E38E38, C56C56, 9B49B4, FC0FC0, 372372, 9596A6, A9556A, 7138EC, A6359C, 2DC2DC$],

H_2=[000000, $FFF000, 8EA8EA, A9356C, 4F1B0E, 3A9C56, 6A66A6, 56C56C, 59A59A, 2DC2DC, B0EB0E, 9596A6, 1C7E38, A6559A, 372372, D232DC, E38E38, 64B9B4, C56C56, 9B49B4, 3FFC0, C8D372, FC0FC0, 7158EA$].

Continuing with the example, we can also construct (non-equivalent) Hadamard codes with the ranks and the dimension of the kernels given in Table 2 for length $n = 48$ and $n = 96$. These codes are constructed using Lemmas 7, 8, 9, 10 and 11 and are denoted by $\star$, $*$, $\diamond$, $\circ$ and $\bullet$, respectively.

Table 2. Dimension of the kernels and ranks of Hadamard codes of length $n = 48$ and $n = 96$, respectively

$n = 48$

	$rank(C)$					
$ker(C)$	13	14	15	16	$\cdots$	24
3	$\star$	$\bullet$				
2	$\star$	$\star$	$\diamond$	$\diamond$	$\cdots$	$\diamond$
1	$\circ$	$*$	$*$	$*$	$\cdots$	$*$

$n = 96$

	$rank(C)$									
$ker(C)$	14	15	16	$\cdots$	25	26	27	28	$\cdots$	48
4	$\star$	$\star$								
3	$\star$	$\star$	$\star$	$\cdots$	$\star$	$\bullet$				
2	$\star$	$\star$	$\star$	$\cdots$	$\star$	$\star$	$\diamond$	$\diamond$	$\cdots$	$\diamond$
1	$\circ$	$\star$	$*$	$\cdots$	$*$	$*$	$*$	$*$	$\cdots$	$*$

5 Conclusions

In this paper we studied the 2-rank and the dimension of the kernel for Hadamard codes of length $n = 2^t \cdot s$ ($s \neq 1$ odd). These two parameters can be used to distinguish between non-equivalent Hadamard matrices, since equivalent ones have codes with the same parameters.

We proved the existence of Hadamard codes of length $n = 2^t \cdot s$ ($s \neq 1$ odd) with rank r and kernel of dimension k, for all $r \in \{4s+t-3, \ldots, n/2\}$ and $k \in \{1, \ldots, t-1\}$, provided that there exists a Hadamard code of length $4s$. It is still an open problem to establish the exact lower bound for the rank of these codes. However, if the dimension of the kernel is $t-1$ or $t-2$, then $4s+t-3$ is the exact lower bound. We claim that this value is always the lower bound. To prove this it is enough to show the non-existence of Hadamard codes with $k = 1$ and $r < 4s+t-3$. The non-existence of a Hadamard code of length $n = 48$, with $k = 1$ and $r < 13$ is the smallest unknown case.

References

1. E. F. Assmus Jr. and J. D. Key, *Designs and their codes*, Cambridge University Press, Great Britain (1992).
2. E. F. Assmus Jr. and J. D. Key, *Designs and Codes: an update*, Designs, Codes and Cryptography, Vol. 9 (1996) pp. 7-27.
3. H. Bauer, B. Ganter and F. Hergert, *Algebraic techniques for nonlinear codes*, Combinatorica, Vol. 3 (1983) pp. 21-33.
4. J. Hadamard, *Résolution d'une question relative aux déterminants*, Bulletin des Sciences Mathématiques, 17 (1893) pp. 240-246.
5. H. Kharaghani and B. Tayleh-Rezaie, *A Hadamard matrix of order 428*, preprint, 2004. http://math.ipm.ac.ir/tayfeh-r/papersandpreprints/h428.pdf
6. D. S. Krotov, *$\mathbb{Z}_4$-linear Hadamard and extended perfect codes*, Procs. of the International Workshop on Coding and Cryptography, Paris (France), Jan. 8-12 (2001) pp. 329-334.
7. T. S. Michael, *The rigidity theorems of Hamada and Ohmori, revisited*, in Coding Theory and Cryptography: From the Geheimschreiber and Enigma to Quantum Theory. (Annapolis, MD, 1998), Springer, Berlin (2000) pp. 175-179.
8. F. I. MacWilliams and N. J. Sloane, *The theory of Error-Correcting codes*, North-Holland, New York (1977).
9. K. T. Phelps, J. Rifà and M. Villanueva, *Rank and Kernel of additive ($\mathbb{Z}_4$-linear and non-$\mathbb{Z}_4$-linear) Hadamard codes*, Proceedings of ACCT'04 conference. Kranevo, Bulgary, June 19-25 (2004) pp. 327-332.
10. K. T. Phelps, J. Rifà and M. Villanueva, *On the additive ($\mathbb{Z}_4$-linear and non-$\mathbb{Z}_4$-linear) Hadamard codes. Rank and Kernel*, to appear in IEEE Transactions on Information Theory.
11. K. T. Phelps, J. Rifà and M. Villanueva, *Binary Hadamard Codes of length 2^t. Rank and Kernel.*, Proceedings of OC'05 conference (Optimal Codes and Related Topics). Pamporovo, Bulgary, June 17-23 (2005) pp. 244-247.
12. K. T. Phelps, J. Rifà and M. Villanueva, *Rank and Kernel dimension of binary Hadamard codes*, to appear in IEEE Transactions on Information Theory.
13. K. T. Phelps and M. Villanueva, *On Perfect Codes: Rank and Kernel*, Designs, Codes and Cryptography, Vol. 27 (2002) pp. 183-194.

Author Index