Lecture Notes in Computer Science 5527

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

Maria Bras-Amorós Tom Høholdt (Eds.)

Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes

18th International Symposium, AAECC-18
Tarragona, Spain, June 8-12, 2009
Proceedings

Springer

Volume Editors

Maria Bras-Amorós
Universitat Rovira i Virgili
Departament d'Enginyeria Informàtica i Matemàtiques
Avinguda Països Catalans, 26, 43007 Tarragona, Catalonia, Spain,
E-mail: maria.bras@urv.cat

Tom Høholdt
The Technical University of Denmark
Department of Mathematics
Building 303, 2800 Lyngby, Denmark,
E-mail: tom@mat.dtu.dk

Library of Congress Control Number: Applied for

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 3-642-02180-8 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-02180-0 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12690084 06/3180 5 4 3 2 1 0

Preface

The AAECC symposia series was started in 1983 by Alain Poli (Toulouse), who, together with R. Desq, D. Lazard and P. Camion, organized the first conference. Originally the acronym AAECC stood for "Applied Algebra and Error-Correcting Codes." Over the years its meaning has shifted to "Applied Algebra, Algebraic Algorithms and Error-Correcting Codes," reflecting the growing importance of complexity, particularly for decoding algorithms. During the AAECC-12 symposium the Conference Committee decided to enforce the theory and practice of the coding side as well as the cryptographic aspects. Algebra was conserved, as in the past, but slightly more oriented to algebraic geometry codes, finite fields, complexity, polynomials, and graphs. The main topics for AAECC-18 were algebra, algebraic computation, codes and algebra, codes and combinatorics, modulation and codes, sequences, and cryptography.

The invited speakers of this edition were Iwan Duursma, Henning Stichtenoth, and Fernando Torres. We would like to express our deep regret for the loss of Professor Ralf Kötter, who recently passed away and could not be our fourth invited speaker.

Except for AAECC-1 (Discrete Mathematics 56, 1985) and AAECC-7 (Discrete Applied Mathematics 33, 1991), the proceedings of all the symposia have been published in Springer's *Lecture Notes in Computer Science* (Vols. 228, 229, 307, 356, 357, 508, 539, 673, 948, 1255, 1719, 2227, 2643, 3857, 4851).

It is a policy of AAECC to maintain a high scientific standard, comparable to that of a journal. This was made possible thanks to the many referees involved. Each submitted paper was evaluated by at least two international researchers. AAECC-18 received and refereed 50 submissions. Of these, 22 were selected for publication in these proceedings as regular papers and 7 were selected as extended abstracts.

The symposium was organized by Maria Bras-Amorós and Tom Høholdt, with the help of Jesús Manjón, Glòria Pujol, Jordi Castellà, Antoni Martínez, and Xavier Fernández under the umbrella of the CRISES group for Cryptography and Statistical Secrecy, at the Universitat Rovira i Virgili, led by Josep Domingo-Ferrer.

It was sponsored by the Catalan Government, the UNESCO Chair in Data Privacy located at Universitat Rovira i Virgili, and the Spanish Network on Mathematics of Information Society.

We would like to dedicate these proceedings to the memory of our colleague Ralf Kötter.

June 2009

Maria Bras-Amorós
Tom Høholdt

Applied Algebra, Algebraic Algorithms, and Error Correcting Codes - AAECC-18

General Chair

Maria Bras-Amorós	Universitat Rovira i Virgili, Catalonia, Spain

Co-chair

Tom Høholdt	Technical University of Denmark

Program Committee

Jacques Calmet	University of Karlsruhe, Germany
Claude Carlet	University of Paris 8, France
Gerard D. Cohen	Ecole Nationale Supérieure des Télécommunications, France
Cunsheng Ding	Hong Kong University of Science and Technology, China
Gui-Liang Feng	University of Southwestern Louisiana, USA
Marc Giusti	Ecole Polytechnique, France
Guang Gong	University of Waterloo, Canada
Joos Heintz	University of Buenos Aires, Argentina and University of Cantabria, Spain
Kathy Horadam	RMIT University, Australia
Hideki Imai	University of Tokyo, Japan
Navin Kashyap	Queen's University, Canada
Shu Lin	University of California, USA
Oscar Moreno	University of Puerto Rico
Wai Ho Mow	Southwest Jiaotong University, China
Harald Niederreiter	National University of Singapore
Michael E. O'Sullivan	San Diego State University, USA
Ferruh Ozbudak	Middle East Technical University, Turkey
Udaya Parampalli	University of Melbourne, Australia
Alain Poli	University P. Sabatier, France
S. Sandeep Pradhan	University of Michigan at Ann Arbor, USA
Asha Rao	Royal Melbourne Institute of Technology, Australia
Shojiro Sakata	Technical University of Denmark
Hong-Yeop Song	Yonsei University, Korea
Chaoping Xing	Nanyang Technological University, Singapore

Organizing Committee

Jesús Manjón	Universitat Rovira i Virgili, Catalonia, Spain
Glòria Pujol	Universitat Rovira i Virgili, Catalonia, Spain
Jordi Castellà-Roca	Universitat Rovira i Virgili, Catalonia, Spain
Antoni Martínez-Ballesté	Universitat Rovira i Virgili, Catalonia, Spain
Xavier Fernández	Universitat Rovira i Virgili, Catalonia, Spain

Table of Contents

Codes

Cryptography

Algebra

Extended Abstracts

The Order Bound for Toric Codes

Peter Beelen and Diego Ruano*

DTU-Mathematics, Technical University of Denmark,
Matematiktorvet, Building 303,
2800 Kgs. Lyngby, Denmark
{P.Beelen,D.Ruano}@mat.dtu.dk

Abstract. In this paper we investigate the minimum distance of generalized toric codes using an order bound like approach. We apply this technique to a family of codes that includes the Joyner code. For some codes in this family we are able to determine the exact minimum distance.

1 Introduction

In 1998 J.P. Hansen considered algebraic geometry codes defined over toric surfaces [7]. Thanks to combinatorial techniques of such varieties he was able to estimate the parameters of the resulting codes. For example, the minimum distance was estimated using intersection theory. Toric geometry studies varieties which contain an algebraic torus as a dense subset and where moreover the torus acts on the variety. The importance of such varieties, called toric varieties, resides in their correspondence with combinatorial objects, which makes the techniques to study the varieties (such as cohomology, intersection theory, resolution of singularities, etc) more precise and at the same time tractable [3,6].

The order bound gives a way to obtain a lower bound for the minimum distance of linear codes [1,4,5,9]. Especially for codes from algebraic curves this technique has been very successful. In this article we will develop a similar bound for toric codes. Actually our bound also works for the more general class of generalized toric codes (see Section 2). This will give a new way of estimating the minimum distance of toric codes that in some examples give a better bound than intersection theory. Another advantage is that known algorithms [4,5] can be used to decode the codes up to half the order bound. As an example we will compute the order bound for a family of codes that includes the Joyner codes [11]. For this reason we call these codes generalized Joyner codes. Also we will compute the exact minimum distance for several generalized Joyner codes. It turns out that a combination of previously known techniques and the order bound gives a good estimate of the minimum distance of generalized Joyner codes.

The paper is organized as follows. In Section 2 we will give an introduction to toric codes and generalized toric codes, while in Section 3 the order bound for

* The work of D. Ruano is supported in part by DTU, H.C. Oersted post doc. grant (Denmark) and by MEC MTM2007-64704 and Junta de CyL VA065A07 (Spain).

M. Bras-Amorós and T. Høholdt (Eds.): AAECC 2009, LNCS 5527, pp. 1–10, 2009.

these codes will be established. The last section of the paper will illustrate the theory by applying the results to generalized Joyner codes.

2 Toric Codes and Generalized Toric Codes

Algebraic geometry codes [9,19] are usually defined evaluating algebraic functions over a non-singular projective variety X defined over a finite field. The functions of $\mathcal{L}(D)$ are evaluated at certain rational points of the curve ($\mathcal{P} = \{P_1, \ldots, P_n\}$), where D is a divisor whose support does not contain any of the evaluation points. The zeros and poles of the functions of $\mathcal{L}(D)$ are bounded by D. More precisely, the algebraic geometry code $\mathcal{C}(X, D, \mathcal{P})$ is the image of the linear map:

$$\begin{aligned} \mathrm{ev} : \mathcal{L}(D) &\to \mathbb{F}_q^n \\ f &\mapsto (f(P_1), \ldots, f(P_n)) \end{aligned}$$

In this section we introduce toric codes, that is, algebraic geometry codes over toric varieties. One can define a toric variety and a Cartier divisor using a convex polytope, namely, a convex polytope is the same datum as a toric variety and Cartier divisor. Let M be a lattice isomorphic to $\mathbb{Z}^r$ for some $r \in \mathbb{Z}$ and $M_{\mathbb{R}} = M \otimes \mathbb{R}$. Let P be an r-dimensional rational convex polytope in $M_{\mathbb{R}}$ and let us consider X_P and D_P the toric variety and the Cartier divisor defined by P [15]. We may assume that X_P is non singular, in other case we refine the fan [6, Section 2.6]. Let $\mathcal{L}(D_P)$ be the $\mathbb{F}_q$-vector space of functions f over X_P such that $\mathrm{div}(f) + D_P \succeq 0$.

The toric code $\mathcal{C}(P)$ associated to P is the image of the linear evaluation map

$$\begin{aligned} \mathrm{ev} : \mathcal{L}(D_P) &\to \mathbb{F}_q^n \\ f &\mapsto (f(t))_{t \in T} \end{aligned}$$

where the set of points $\mathcal{P} = T$ is the algebric torus $T = (\mathbb{F}_q^*)^r$. Since we evaluate at $\#T$ points, $\mathcal{C}(P)$ has length $n = (q-1)^r$. One has that $\mathcal{L}(D_P)$ is the $\mathbb{F}_q$-vector space generated by the monomials with exponents in $P \cap M$

$$\mathcal{L}(D_P) = \langle \{ X^u = X_1^{u_1} \cdots X_r^{u_r} \mid u \in P \cap M \} \rangle \subset \mathbb{F}_q[X_1, \ldots, X_r]$$

The minimum distance of a toric code $\mathcal{C}(P)$ may be estimated using intersection theory [8,15]. Also, it can be estimated using a multivariate generalization of Vandermonde determinants on the generator matrix [13]. For plane polytopes, $r = 2$, one can estimate the minimum distance using the Hasse-Weil bound and combinatorial invariants of the polytope (the Minkowsky sum [12] and the Minkowsky length [17]).

An extension of toric codes are the so-called generalized toric codes [16]. The generalized toric code $\mathcal{C}(U)$ is the image of the $\mathbb{F}_q$-linear map

$$\begin{aligned} \mathrm{ev} : \mathbb{F}_q[U] &\to \mathbb{F}_q^n \\ f &\mapsto (f(t))_{t \in T} \end{aligned}$$

where $U \subset H = \{0, \ldots, q-2\}^r$ and $\mathbb{F}_q[U]$ is the $\mathbb{F}_q$-vector space

$$\mathbb{F}_q[U] = \langle X^u = X_1^{u_1} \cdots X_r^{u_r} \mid u = (u_1, \ldots, u_r) \in U \rangle \subset \mathbb{F}_q[X_1, \ldots, X_r].$$

Let $\overline{u}$ be $u \mod ((q-1)\mathbb{Z})^r$, that is $\overline{u} = (u_1 \mod (q-1), \ldots, u_r \mod (q-1))$, for $u \in \mathbb{Z}^r$, and $\overline{U} = \{\overline{u} \mid u \in U\}$. The dimension of the code $\mathcal{C}(U)$ is $k = \#\overline{U} = \#U$, since the evaluation map ev is injective.

By [16, Theorem 6], one has that the dual code of $\mathcal{C}(U)$ is $\mathcal{C}(U^\perp)$, where $U^\perp = H \setminus \overline{-U}$, with $\overline{-U} = \{\overline{-u} \,|\, u \in U\}$. Namely, we have

$$\mathrm{ev}(X^u) \cdot \mathrm{ev}(X^{u'}) = \begin{cases} 0 & \text{if } \overline{u+u'} \neq 0 \\ (-1)^r & \text{if } \overline{u+u'} = 0 \end{cases} \tag{1}$$

for $u, u' \in H$, where $\cdot$ denotes the inner product in $\mathbb{F}_q^n$.

The family of generalized toric codes includes the ones obtained evaluating polynomials of an arbitrary subalgebra of $\mathbb{F}_q[X_1, \ldots, X_r]$ at T, in particular toric codes. However, there is no estimate so far for the minimum distance in this more general setting, the order bound techniques in this paper will apply to generalized toric codes as well. From now on we will consider generalized toric codes but for the sake of simplicity, we will just call them toric codes.

3 The Order Bound for Toric Codes

In this section we follow the order bound approach to estimate the minimum distance of the dual code of a toric code $\mathcal{C}(U)$, for $U \subset H$.

Let $\mathcal{B}_1 = \{g_1, \ldots, g_n\}$ and $\mathcal{B}_2 = \{h_1, \ldots, h_n\}$ be two bases of $\mathbb{F}_q[H]$. For $c = \mathrm{ev}(f) \in \mathcal{C}(U)$, we consider the syndrome matrix $S(c) = (s_{i,j})_{1 \leq i,j \leq n}$, with $s_{i,j} = (\mathrm{ev}(g_i) * \mathrm{ev}(h_j)) \cdot \mathrm{ev}(f) = \mathrm{ev}(g_i h_j) \cdot \mathrm{ev}(f)$, where $*$ denotes the componentwise product. In other words, $S(c) = M_1 D(c) M_2^t$, where D is the diagonal matrix with c in the diagonal and M_1 and M_2 are the evaluation matrices given by

$$M_1 = \begin{pmatrix} g_1(t_1) & g_1(t_2) & \cdots & g_1(t_n) \\ g_2(t_1) & g_2(t_2) & \cdots & g_2(t_n) \\ \vdots & \vdots & \vdots & \vdots \\ g_n(t_1) & g_n(t_2) & \cdots & g_n(t_n) \end{pmatrix}, \quad M_2 = \begin{pmatrix} h_1(t_1) & h_1(t_2) & \cdots & h_1(t_n) \\ h_2(t_1) & h_2(t_2) & \cdots & h_2(t_n) \\ \vdots & \vdots & \vdots & \vdots \\ h_n(t_1) & h_n(t_2) & \cdots & h_n(t_n) \end{pmatrix}.$$

Here $t_1, \ldots, t_n$ denote the points of the algebraic torus. Note that M_1 and M_2 have full rank, since the evaluation map is injective. This implies that the rank of $S(c)$ equals $\mathrm{wt}(c)$. It is convenient to consider bases of $\mathbb{F}_q[H]$ consisting of monomials, that is, we set $g_i = X^{v_i}$ and $h_i = X^{w_i}$, for $i = 1, \ldots, n$, with $\{v_1, \ldots, v_n\} = \{w_1, \ldots, w_n\} = H$. Then, we can easily compute the syndrome matrix for a codeword using the following lemma.

Lemma 1. *Let $f = \sum_{u \in H} \lambda_u X^u$ and $S(\mathrm{ev}(f)) = (s_{i,j})_{1 \leq i,j \leq n}$ the syndrome matrix of $\mathrm{ev}(f)$. Then, one has that $s_{i,j} = (-1)^r \lambda_{\overline{-(v_i+w_j)}}$. In particular, $s_{i,j}$ is equal to zero if and only if $\overline{v_i + w_j} \notin \overline{-\mathrm{supp}(f)}$, where $\mathrm{supp}(f)$ denotes the support of f, $\mathrm{supp}(f) = \{u \in H \mid \lambda_u \neq 0\}$.*

Proof. By definition,

$$\begin{aligned} s_{i,j} &= \mathrm{ev}(X^{\overline{v_i+w_j}}) \cdot \mathrm{ev}(\sum_{u\in H} \lambda_u X^u) = \sum_{u\in H} \mathrm{ev}(X^{\overline{v_i+w_j}}) \cdot \mathrm{ev}(\lambda_u X^u) \\ &= (-1)^r \lambda_{\overline{-(v_i+w_j)}} \quad \text{(by (1))}. \end{aligned}$$

Therefore, $s_{i,j}$ is equal to zero if and only if $\overline{-(v_i+w_j)}$ is not in the support of f. Equivalently, $s_{i,j}=0$ if and only if $\overline{v_i+w_j} \notin \overline{-\mathrm{supp}(f)}$. □

To bound the minimum distance using order domain theory, we should give a lower bound for the rank of the syndrome matrix. Since the order bound gives an estimate for the minimum distance of the dual code, we begin by considering $\mathcal{C}(U)^\perp = \mathcal{C}(U^\perp)$ to get a bound for the minimum distance of $\mathcal{C}(U)$.

Let $H=\{u_1,\ldots,u_n\}$, with $U^\perp=\{u_1,\ldots,u_{n-k}\} \subset H$, notice that $U = \{\overline{-u_{n-k+1}},\ldots,\overline{-u_n}\}$. We are dealing with an arbitrary order on H, we only require, for the sake of simplicity, that the first $n-k$ elements of H are the elements of $U^\perp$. For $l \in \{0,\ldots,k-1\}$, we consider the following filtration of codes depending on the previous ordering

$$C \subsetneq C_1 \subsetneq C_2 \subsetneq \cdots \subsetneq C_l \subsetneq C_{l+1},$$

where $C=\mathcal{C}(U^\perp)$ and $C_m = \mathcal{C}(U^\perp \cup \{u_{n-k+1},\ldots,u_{n-k+m}\})$, for $m=1,\ldots,l+1$, and their dual codes,

$$C^\perp \supsetneq C_1^\perp \supsetneq C_2^\perp \supsetneq \cdots \supsetneq C_l^\perp \supsetneq C_{l+1}^\perp,$$

with $C^\perp = \mathcal{C}(U)$ and $C_m^\perp = \mathcal{C}(U \setminus \{\overline{-u_{n-k+1}},\ldots,\overline{-u_{n-k+m}}\})$, for $m=1,\ldots,l+1$, since $(U^\perp \cup \{u_{n-k+1},\ldots,u_{n-k+m}\})^\perp = U \setminus \{\overline{-u_{n-k+1}},\ldots,\overline{-u_{n-k+m}}\}$.

We wish to bound the weight of $c \in C_l^\perp \setminus C_{l+1}^\perp$. Let ν_l be the largest integer (in $\{1,\ldots,n\}$) such that

- $\overline{v_i+w_i} = u_{n-k+l+1}$, for $i=1,\ldots,\nu_l$.
- $\overline{v_i+w_j} \in U^\perp \cup \{u_{n-k+1},\ldots,u_{n-k+l}\}$, for $i=1,\ldots,\nu_l$ and $j<i$.

Proposition 1. *Let $c \in \mathcal{C}_l^\perp \setminus \mathcal{C}_{l+1}^\perp$, then $wt(c) \geq \nu_l$.*

Proof. Let $c=ev(f)$, then $f=\sum \lambda_u X^u$, where $u \in U \setminus \{\overline{-u_{n-k+1}},\ldots,\overline{-u_{n-k+l}}\}$. Notice that $\lambda_{\overline{-u_{n-k+l+1}}} \neq 0$, since $c \notin C_{l+1}^\perp$. Hence we have by Lemma 1 that,

- $s_{i,i} \neq 0$, for $i=1,\ldots,\nu_l$, since $\overline{v_i+w_i} = u_{n-k+l+1} \in \overline{-\mathrm{supp}(f)}$, for $i=1,\ldots,\nu_l$.
- $s_{i,j}=0$, for $i=1,\ldots,\nu_l$, since $\overline{v_i+w_j} \in U^\perp \cup \{u_{n-k+1},\ldots,u_{n-k+l}\}$, for $j<i$. That is, $\overline{v_i+w_j} \notin -\mathrm{supp}(f)$ because

$$H \setminus (U^\perp \cup \{u_{n-k+1},\ldots,u_{n-k+l}\}) = \overline{-U} \setminus \{u_{n-k+1},\ldots,u_{n-k+l}\}$$

Therefore, the submatrix of $S(c)$ consisting of the first ν_l rows and columns has full rank. In particular, the rank of $S(c)$ is at least ν_l and the result holds since the rank of $S(c)$ is equal to the weight of c. □

For every l in $\{0,\ldots,k-1\}$ we consider a filtration and we obtain a bound for the weight of a word in $\mathcal{C}_l^\perp \setminus \mathcal{C}_{l+1}^\perp$. Therefore, we have obtained the following bound for the minimum distance of $C^\perp = \mathcal{C}(U)$.

Theorem 1. *Let $\mathcal{C}(U)$ be a toric code with $U \subset H$. Then,*

$$d(\mathcal{C}(U)) \geq \min\{\nu_l \mid l = 0,\ldots,k-1\}.$$

Remark 1. We can apply known decoding algorithms [4,5] to decode a toric code $\mathcal{C}(U)$ up to half of the order bound obtained in the previous theorem.

In the next section we will use the above approach to estimate the minimum distance of a family of toric codes.

4 Generalized Joyner Codes

In this section we will introduce a class of toric codes that includes the well-known Joyner code [11, Example 3.9]. After introducing these codes, we will calculate a lower bound for their minimum distances using techniques from Section 3. Then we will calculate another lower bound for the minimum distance using a combination of the order bound and Serre's improvement of Hasse-Weil's theorem on the number of rational points on a curve [18]. In some cases we are able to compute the exact minimum distance. In this section we will always assume that $r = 2$, so that $H = \{0,\ldots,q-2\} \times \{0,\ldots,q-2\}$.

Definition 1. *Let q be a power of a prime and a an integer satisfying $2 \leq a \leq q-2$. We define the sets*

$$U_a = \{(u_1,u_2) \in H \mid u_1 + u_2 \leq a+1, u_1 - au_2 \leq 0, -au_1 + u_2 \leq 0\},$$

$$T_a = \{(u_1,u_2) \in H \mid u_1 + u_2 \leq a+1, u_1 \geq 1, u_2 \geq 1\},$$

$$V_a = U_a \setminus \{(1,a)\}, \text{ and } W_a = U_a \setminus \{(a,1)\}.$$

The set U_a consists of all elements of H lying in or on the boundary of the triangle with vertices $(0,0)$, $(1,a)$ and $(a,1)$. Note that the condition on a ensures that the points $(1,a)$ and $(a,1)$ are in H. Also note that the set U_a can be obtained by joining $(0,0)$ to the set T_a. All sets in the above definition are actually sets of integral points in a polytope, so the corresponding codes are classical toric codes.

We wish to investigate the toric code $\mathcal{C}(U_a)$ and begin by establishing some elementary properties:

Lemma 2. *The code $\mathcal{C}(U_a)$ is an $[(q-1)^2, 1 + a(a+1)/2, d]$ code over $\mathbb{F}_q$ and we have $d \leq (q-1)(q-a)$.*

Proof. Since the set U_a is contained in H, the dimension of the corresponding code is equal to the number of elements in U_a. Since U_a can be obtained by joining $\{(0,0)\}$ to the set T_a the formula for the dimension follows by a counting argument.

To prove the result on the minimum distance first note that the code $\mathcal{C}(T_a)$ is a subcode of $\mathcal{C}(U_a)$. It is well known, [8, Theorem 1.3], that $d(\mathcal{C}(T_a)) = (q-1)(q-a)$, so the result follows. □

The code $\mathcal{C}(U_4)$ is the Joyner code over $\mathbb{F}_q$, see [11]. It is known that for $q \geq 37$ its minimum distance is equal to $(q-1)(q-4)$, [17], meaning that the upper bound in the previous lemma is attained. In fact equality already holds for much smaller q. Using a computer one finds that $q = 8$ is the smallest value of q for which equality holds. It is conjectured that for all $q \geq 8$ one has that $d(\mathcal{C}(U_4)) = (q-1)(q-4)$. This behavior turns out to happen as well for other values of a. This is the reason we study these codes in this section. We proceed our investigation by calculating two lower bounds for the minimum distance of the codes $\mathcal{C}(U_a)$. The first one holds for any q, while the second one turns out to be interesting only for large q.

Proposition 2. *The minimum distance d of the $\mathbb{F}_q$-linear code $\mathcal{C}(U_a)$ satisfies $d \geq (q-1)(q-a-1)$.*

Proof. Since $d(\mathcal{C}(T_a)) = (q-1)(q-a)$, the proposition follows once we have shown that $\mathrm{wt}(c) \geq (q-1)(q-a-1)$ for any $c \in \mathcal{C}(U_a)\backslash\mathcal{C}(T_a)$. For such c it holds that $c = \mathrm{ev}(f)$ for some $f \in \mathbb{F}_q[U_a]$ satisfying that $(0,0) \in \mathrm{supp}(f)$. We will now use Proposition 1. Any number i between 0 and $(q-1)(q-a-1)-1$ can be written uniquely as $i = \beta_i \cdot (q-1) + \alpha_i$ with α_i and β_i integers satisfying $0 \leq \alpha_i \leq q-2$ and $0 \leq \beta_i \leq q-a-2$. For i between 0 and $(q-1)(q-a-1)-1$ we then define $v_i = (\alpha_i, \beta_i)$ and $w_i = \overline{-v_i}$. By construction of w_i it then holds that $\overline{v_i + w_i} = (0,0)$. On the other hand, if $j < i$, then $\overline{v_i + w_j} \neq (0,0)$ and $0 \leq \beta_i - \beta_j \leq q-a-2$ implying that $\overline{v_i + w_j} \notin \overline{-U_a}$. By Proposition 1, we get that $\mathrm{wt}(c) \geq (q-1)(q-a-1)$. □

For $q = 8$ and $a = 4$, the Joyner code case, we obtain that $d \geq 21$. An other method to obtain a lower bound for the minimum distance of toric codes is to use intersection theory. For the Joyner code over $\mathbb{F}_8$ one can prove in this way that $d \geq 12$, [14]. The bound we get compares favorably to it. Another advantage of the order bound techniques is that they are valid for generalized toric codes as well.

Now we obtain a second lower bound on the minimum distance of the code $\mathcal{C}(U_a)$. First we need a lemma.

Lemma 3. *Let c be a nonzero codeword from the code $\mathcal{C}(V_a)$ or the code $\mathcal{C}(W_a)$. Then $\mathrm{wt}(c) \geq (q-1)(q-a)$.*

Proof. Suppose that $c \in \mathcal{C}(V_a)$ (the case that $c \in \mathcal{C}(W_a)$ can be dealt with similarly by symmetry and will not be discussed below). If $c \in \mathcal{C}(T_a)$, we are done. Therefore we can suppose that $c \in \mathcal{C}(V_a)\backslash\mathcal{C}(T_a)$. Exactly as in the proof

of Proposition 2 we now define for i between 0 and $(q-1)(q-a)-1$ the tuple $u_i = (\alpha_i, \beta_i)$. The only difference is that now β_i is also allowed to be $q-a$, otherwise everything is the same. Further we also define $w_i = \overline{-v_i}$. Then we have that $\overline{v_i + w_i} = (0,0)$ and for $j < i$, we obtain that $\overline{v_i + w_j} \neq (0,0)$ and $0 \leq \beta_i - \beta_j \leq q-a-1$. This implies that $\overline{v_i + w_j} \notin \overline{-V_a}$. The lemma now follows. □

One can use Lemma 3 and the fact that $d(\mathcal{C}(T_a)) = (q-1)(q-a)$ [8], to restrict the number of possibilities for a non-zero codeword of weight less than $(q-1)(q-a)$. Namely, it has to be the evaluation of a function $f = \sum \lambda_u X^u$ with non-zero coefficients $\lambda_{(a,1)}, \lambda_{(0,0)}, \lambda_{(1,a)}$. We will use this in the following proposition. We distinguish cases between $a = 2$ and $a > 2$.

Proposition 3. *Let U_a be the set from Definition 1 and let $a > 2$. The minimum distance d of the code $\mathcal{C}(U_a)$ satisfies*

$$d \geq \min\left\{(q-1)(q-a), q^2 - 3q + 2 - \frac{a(a-1)}{2}\lfloor 2\sqrt{q}\rfloor\right\}.$$

Proof. Let $c \in \mathcal{C}(U_a)$ be a nonzero codeword and suppose that $c = \mathrm{ev}(f)$. If $\mathrm{supp}(f) \subset T_a$ then we know that $\mathrm{wt}(c) \geq (q-1)(q-a)$ from [8] as noted before. If $\mathrm{supp}(f) \subset V_a$ or $\mathrm{supp}(f) \subset W_a$ then $\mathrm{wt}(c) \geq (q-1)(q-a)$ by Lemma 3.

We are left with the case that $\{(0,0),(1,a),(a,1)\} \subset \mathrm{supp}(f)$. In this case the Newton-polygon of the polynomial f is Minkowski-indecomposable which implies that the polynomial f is absolutely irreducible. We can therefore consider the algebraic curve C_f defined by the equation $f = 0$. From Newton-polygon theory it follows that this curve has geometric genus at most $a(a-1)/2$ and that the edges from $(0,0)$ to $(1,a)$ and from $(0,0)$ to $(a,1)$ correspond to two rational points at infinity using projective coordinates (see [2, Remark 3.18 and Theorem 4.2]). We denote by N the total number of pairs $(\alpha,\beta) \in \mathbb{F}_q^2$ such that $f(\alpha,\beta) = 0$. We claim that $N \leq q - 1 + a(a-1)/2\lfloor\sqrt{q}\rfloor$. If the curve C has no singularities, all solutions (α,β) correspond one-to-one to all affine $\mathbb{F}_q$-rational points. Taking into account that there at least 2 rational points at infinity (in fact at least 3 if $a = 2$), the claim follows from Serre's bound (and can be slightly improved if $a = 2$). If there are singularities, a solution (α,β) may not correspond to a rational point on C, but for every such solution the genus will drop at least one, so the claim still follows from Serre's bound. The proposition now follows, since $\mathrm{wt}(c) \geq (q-1)^2 - N$. □

For $a = 2$ we can determine the exact minimum distance. We will do so in the following theorem. This theorem is formulated using the existence or non-existence of an elliptic curve with a certain number of points. The theorem is completely constructive, since it is very easy to determine if an elliptic curve defined over $\mathbb{F}_q$ with a certain number of points exists. To this end one can use the following fact [20, Theorem 4.1]:

Let q be a power of a prime p and let t be an integer. There exists an elliptic curve defined over $\mathbb{F}_q$ with $q+1+t$ points if and only if the following holds:

1. $p \not| t$ and $t^2 \leq 4q$,
2. e is odd and one of the following holds
   (a) $t = 0$,
   (b) $t^2 = 2q$ and $p = 2$,
   (c) $t^2 = 3q$ and $p = 3$,
3. e is even and one of the following holds
   (a) $t^2 = 4q$,
   (b) $t^2 = q$ and $p \not\equiv 1 \bmod 3$,
   (c) $t = 0$ and $p \not\equiv 1 \bmod 4$.

Theorem 2. *Denote by d the minimum distance of the $\mathbb{F}_q$-linear code $\mathcal{C}(U_2)$. Further let t be the largest integer such that*

1. $3 \mid q + 1 + t$,
2. *there exists an elliptic curve defined over $\mathbb{F}_q$ with $q + 1 + t$ points.*

Then we have: $d = q^2 - 3q + 3 - t$.

Proof. Analogously as in the previous proposition we only have to consider codewords coming from functions f such that $\{(0,0),(1,2),(2,1)\} \subset \mathrm{supp}(f)$. We again consider the curve C_f given by the equation $f = 0$. In this case all three edges of the Newton polygon of f correspond to rational points at infinity.

The polynomial f can be written as $\alpha + \beta X_1 X_2 + \gamma X_1 X_2^2 + \delta X_1^2 X_2$, where α, γ and δ are nonzero. We may assume that the curve does not have singularities, since otherwise the geometric genus of C_f is zero, which implies that the equation $f = 0$ has at most $1 + (q+1) - 3 = q - 1$ solutions in $\mathbb{F}_q^2$ (the first term represents the singular point which could have rational coordinates). This would give rise to a codeword of weight at least $(q-1)^2 - (q-1) = q^2 - 3q + 2$.

By changing variables to $U = -\gamma X_1 X_2/\delta$ and $V = \gamma X_1^2 X_2/\delta$ (or equivalently $X_1 = -V/U$ and $X_2 = \delta U^2/(\gamma V)$) one can show that the curve C_f is also given by the equation $V^2 - \beta UV/\delta + \alpha\gamma V/\delta^2 = U^3$. Since we have assumed that the curve C_f is nonsingular, it is an elliptic curve and we already found a Weierstrass equation for it. In (U, V) coordinates one sees that $(0, 0)$ is a point on the elliptic curve and using the addition formula one checks that this point has order three in the elliptic curve group.

Denote the total number of rational points on C_f by $q + 1 + t$, then clearly there exists an elliptic curve with $q + 1 + t$ points. Since the point $(0, 0)$ has order three, the total number of rational points has to be a multiple of three and it follows that $3|q + 1 + t$. Reasoning back we see that the total number of affine solutions to the equation $f = 0$ in $\mathbb{F}_q^2$ equals $q + 1 + t - 3 = q - 2 + t$. On the other hand there are no solutions with zero coordinates, so we have that $\mathrm{wt}(\mathrm{ev}(f)) = (q-1)^2 - (q-2+t)$.

It remains to be shown which values of t are possible when the polynomial f is varied. It is shown in [10, Section 4.2] that any elliptic curve having $(0, 0)$ as a point of order three has a Weierstrass equation of the form $V^2 + a_1 UV + a_3 V = U^3$, with $a_3 \neq 0$. This implies that t can be any value satisfying the two conditions stated in the formulation of the theorem. Choosing the maximal one among these values gives a codeword of lowest possible nonzero weight. This concludes the proof. □

If $a > 2$ the situation is more complicated. Given a fixed a the lower bound from Proposition 2 is good for relatively small q, while the lower bound from Proposition 3 becomes better as q becomes larger. A combination of the techniques from Section 3 and this section gives an in general good lower bound for the minimum distance. In some cases we are able to determine the minimum distance and we describe when this happens in the following theorem.

Theorem 3. *Let q be a prime power and consider a natural number a satisfying $2 < a \leq q-2$. Define the set U_a as in Definition 1. Then we have the following for the minimum distance d of the $\mathbb{F}_q$-linear code $\mathcal{C}(U_a)$:*

$$d = (q-1)(q-a) \text{ if } 2(a-2)(q-1) \geq (a-1)a\lfloor 2\sqrt{q} \rfloor.$$

Proof. If $(q-1)(q-a) \leq q^2 - 3q + 2 - a(a-1)\lfloor 2\sqrt{q} \rfloor/2$, then it follows from Lemma 2 and Proposition 3 that $d = (q-1)(q-a)$. The theorem follows after some manipulation of this inequality. □

For small values of a we obtain the following corollary. Note that the results for $a = 4$ are the same as in [17].

Corollary 1. *We have*
$d(\mathcal{C}(U_3)) = (q-1)(q-3)$ *if* $q \geq 37$,
$d(\mathcal{C}(U_4)) = (q-1)(q-4)$ *if* $q \geq 37$,
$d(\mathcal{C}(U_5)) = (q-1)(q-5)$ *if* $q = 41$ *or* $q \geq 47$,
$d(\mathcal{C}(U_6)) = (q-1)(q-6)$ *if* $q \geq 59$.

As in the case for the Joyner code it seems that the for many small values of q it also holds that $d(\mathcal{C}(U_a)) = (q-1)(q-a)$. We know by Corollary 1 and some computer calculations that $d(\mathcal{C}(U_3)) = (q-1)(q-3)$ if $q \geq 23$. It is known for the Joyner code that $d(\mathcal{C}(U_4)) = (q-1)(q-4)$ if $q \geq 8$. Finally we conjecture that $d(\mathcal{C}(U_5)) = (q-1)(q-5)$ if $q \geq 9$. It remains future work to prove this conjecture without the aid of a computer and to establish what happens for larger values of a.

References

1. Beelen, P.: The order bound for general algebraic geometric codes. Finite Fields Appl. 13(3), 665–680 (2007)
2. Beelen, P., Pellikaan, R.: The Newton polygon of plane curves with many rational points. Des. Codes Cryptogr. 21(1-3), 41–67 (2000)
3. Danilov, V.I.: The geometry of toric varieties. Russian Math. Surverys 33(2), 97–154 (1978)
4. Duursma, I.M.: Majority coset decoding. IEEE Trans. Inform. Theory 39(3), 1067–1070 (1993)
5. Feng, G.L., Rao, T.R.N.: Improved geometric Goppa codes. I. Basic theory. IEEE Trans. Inform. Theory 41(6, part 1), 1678–1693 (1995)
6. Fulton, W.: Introduction to toric varieties. Annals of Mathematics Studies, vol. 131. Princeton University Press, Princeton (1993)

7. Hansen, J.P.: Toric surfaces and error-correcting codes. In: Buchmann, J., Høholdt, T., Stichtenoth, H., Tapia-Recillas, H. (eds.) Coding theory, cryptography and related areas (Guanajuato, 1998), pp. 132–142. Springer, Berlin (2000)
8. Hansen, J.P.: Toric varieties Hirzebruch surfaces and error-correcting codes. Appl. Algebra Engrg. Comm. Comput. 13(4), 289–300 (2002)
9. Høholdt, T., van Lint, J.H., Pellikaan, R.: Algebraic geometry of codes. In: Handbook of coding theory, vol. I, II, pp. 871–961. North-Holland, Amsterdam (1998)
10. Husemöller, D.: Elliptic curves, 2nd edn. Graduate Texts in Mathematics, vol. 111. Springer, New York (2004)
11. Joyner, D.: Toric codes over finite fields. Appl. Algebra Engrg. Comm. Comput. 15(1), 63–79 (2004)
12. Little, J., Schenck, H.: Toric surface codes and Minkowski sums. SIAM J. Discrete Math. 20(4), 999–1014 (2006)
13. Little, J., Schwarz, R.: On toric codes and multivariate Vandermonde matrices. Appl. Algebra Engrg. Comm. Comput. 18(4), 349–367 (2007)
14. Martínez-Moro, E., Ruano, D.: Toric codes. In: Advances in Algebraic Geometry Codes. Series on Coding Theory and Cryptology, vol. 5, pp. 295–322. World Scientific, Singapore (2008)
15. Ruano, D.: On the parameters of r-dimensional toric codes. Finite Fields Appl. 13(4), 962–976 (2007)
16. Ruano, D.: On the structure of generalized toric codes. J. Symb. Comput. 44(5), 499–506 (2009)
17. Soprunov, I., Soprunova, E.: Toric surface codes and minkowski length of polygons. arXiv:0802.2088v1 [math. AG] (2008)
18. Serre, J.-P.: Sur le nombre des points rationnels d'une courbe algébrique sur un corps fini. C. R. Acad. Sci. Paris Sér. I Math. 296(9), 397–402 (1983)
19. Tsfasman, M.A., Vlăduţ, S.G.: Algebraic-geometric codes, Dordrecht. Mathematics and its Applications (Soviet Series), vol. 58 (1991)
20. Waterhouse, W.C.: Abelian varieties over finite fields. Ann. Sci. École Norm. Sup. 2(4), 521–560 (1969)

An Extension of the Order Bound for AG Codes

Iwan Duursma and Radoslav Kirov

Department of Mathematics, University of Illinois at Urbana-Champaign,
1409 W. Green Street (MC-382) Urbana, Illinois 61801-2975
http://www.math.uiuc.edu/{~duursma,~rkirov2}

Abstract. The most successful method to obtain lower bounds for the minimum distance of an algebraic geometric code is the order bound, which generalizes the Feng-Rao bound. By using a finer partition of the set of all codewords of a code we improve the order bounds by Beelen and by Duursma and Park. We show that the new bound can be efficiently optimized and we include a numerical comparison of different bounds for all two-point codes with Goppa distance between 0 and $2g-1$ for the Suzuki curve of genus $g = 124$ over the field of 32 elements.

Keywords: Algebraic geometric code, order bound, Suzuki curve.

1 Introduction

To obtain lower bounds for the minimum distance of algebraic geometric codes we follow [7] and exploit a relation between the minimum distance of algebraic geometric codes and the representation of divisor classes by differences of base point free divisors. Theorem 2 gives a new improved lower bound for the weight of vectors in a subset of the code. In Section 5 we outline methods to efficiently optimize lower bounds for the minimum distance using the theorem. Section 6 has tables that compare different bounds for two-point Suzuki codes over the field of 32 elements.

Let $X/\mathbb{F}$ be an algebraic curve (absolutely irreducible, smooth, projective) of genus g over a finite field $\mathbb{F}$. Let $\mathbb{F}(X)$ be the function field of $X/\mathbb{F}$. A nonzero rational function $f \in \mathbb{F}(X)$ has divisor $(f) = \sum_{P\in X} \mathrm{ord}_P(f)P = (f)_0 - (f)_\infty$, where the positive part $(f)_0$ gives the zeros of f and their multiplicities, and the negative part $(f)_\infty$ gives the poles of f and their multiplicities. A divisor $D = \sum_P m_P P$ is principal if it is of the form $D = (f)$ for some $f \in \mathbb{F}(X)$. Two divisors D and D' are linearly equivalent if $D' = D + (f)$ for some $f \in \mathbb{F}(X)$. Given a divisor D on X defined over $\mathbb{F}$, let $L(D)$ denote the vector space over $\mathbb{F}$ of nonzero functions $f \in \mathbb{F}(X)$ for which $(f) + D \geq 0$ together with the zero function. A point P is a base point for the linear system of divisors $\{(f) + D : f \in L(D)\}$ if $(f) + D \geq P$ for all $f \in L(D)$, that is to say if $L(D) = L(D - P)$.

We give the definition of an algebraic geometric code. For n distinct rational points $P_1, \ldots, P_n$ on X and for disjoint divisors $D = P_1 + \cdots + P_n$ and G, the geometric Goppa code $C_L(D, G)$ is defined as the image of the map

$$\alpha_L \ : \ L(G) \ \longrightarrow \ \mathbb{F}^n, \ \ f \mapsto (\, f(P_1), \ldots, f(P_n)\,).$$

M. Bras-Amorós and T. Høholdt (Eds.): AAECC 2009, LNCS 5527, pp. 11–22, 2009.

With the Residue theorem, the dual code $C_L(D,G)^\perp$ can be expressed in terms of differentials. Let $\Omega(X)$ be the module of rational differentials for $X/\mathbb{F}$. For a given divisor E on X defined over $\mathbb{F}$, let $\Omega(E)$ denote the vector space over $\mathbb{F}$ of nonzero differentials $\omega \in \Omega(X)$ for which $(\omega) \geq E$ together with the zero differential. Let K represent the canonical divisor class. The geometric Goppa code $C_\Omega(D,G)$ is defined as the image of the map

$$\alpha_\Omega \ : \ \Omega(G-D) \ \longrightarrow \ \mathbb{F}^n, \ \ \omega \mapsto (\,\mathrm{Res}_{P_1}(\omega), \ldots, \mathrm{Res}_{P_n}(\omega)\,).$$

The code $C_\Omega(D,G)$ is the dual code for the code $C_L(D,G)$. We use the following characterization of the minimum distance.

Proposition 1. *[7, Proposition 2.1] For the code $C_L(D,G)$, and for $C = D-G$,*

$$d(C_L(D,G)) = \min\{\deg A : 0 \leq A \leq D \mid L(A-C) \neq L(-C)\}.$$

For the code $C_\Omega(D,G)$, and for $C = G-K$,

$$d(C_\Omega(D,G)) = \min\{\deg A : 0 \leq A \leq D \mid L(A-C) \neq L(-C)\}.$$

Proof. There exists a nonzero word in $C_L(D,G)$ with support in A, for $0 \leq A \leq D$, if and only if $L(G-D+A)/L(G-D) \neq 0$. There exists a nonzero word in $C_\Omega(D,G)$ with support in A, for $0 \leq A \leq D$, if and only if $\Omega(G-A)/\Omega(G) \neq 0$ if and only if $L(K-G+A)/L(K-G) \neq 0$.

It is clear that in each case $d \geq \deg C$. The lower bound $d_{GOP} = \deg C$ is the Goppa designed minimum distance of a code.

2 Coset Bounds

For a point P disjoint from D, consider the subcodes $C_L(D,G-P) \subseteq C_L(D,G)$ and $C_\Omega(D,G+P) \subseteq C_\Omega(D,G)$.

Proposition 2. *[7, Proposition 3.5] Let $A = \mathrm{supp}(c)$ be the support of a codeword $c = (c_P : P \in D)$, with $0 \leq A \leq D$. For $C = D-G$,*

$$c \in C_L(D,G) \backslash C_L(D,G-P) \ \Rightarrow \ L(A-C) \neq L(A-C-P).$$

For $C = G-K$,

$$c \in C_\Omega(D,G) \backslash C_\Omega(D,G+P) \ \Rightarrow \ L(A-C) \neq L(A-C-P).$$

Proof. There exists a word in $C_L(D,G) \backslash C_L(D,G-P)$ with support in A, for $0 \leq A \leq D$, if and only if $C_L(D,G-(D-A)) \neq C_L(D,G-P-(D-A))$ only if $L(G-D+A) \neq L(G-D+A-P)$. There exists a word in $C_\Omega(D,G) \backslash C_\Omega(D,G+P)$ with support in A, for $0 \leq A \leq D$, if and only if $C_\Omega(A,G) \neq C_\Omega(A,G+P)$ only if $\Omega(G-A) \neq \Omega(G-A+P)$, which can be expressed as $L(K-G+A) \neq L(K-G+A-P)$.

The order bound for the minimum distance of an algebraic geometric code is motivated by the decoding procedures in [10], [8] and combines estimates for the weight of a word $c \in C_L(D,G) \setminus C_L(D,G-P)$, or $c \in C_\Omega(D,G) \setminus C_\Omega(D,G+P)$. The basic version (often referred to as the simple or first order bound [13], [4]) takes the form

$$d(C_L(D,G)) = \min_{i \geq 0}(\min\{\mathrm{wt}(c) : c \in C_L(D,G-iP) \setminus C_L(D,G-(i+1)P)\}),$$
$$d(C_\Omega(D,G)) = \min_{i \geq 0}(\min\{\mathrm{wt}(c) : c \in C_\Omega(D,G+iP) \setminus C_L(D,G+(i+1)P)\}).$$

The order bound makes it possible to use separate estimates for different subsets of codewords. The bound is successful if for each subset, i.e. for each i, we can find an estimate that is better than a uniform lower bound for all codewords. Methods that provide uniform lower bounds include the Goppa designed minimum distance and bounds of floor type [17], [16], [11], [7]. It follows from the Singleton bound that the minimum distance of an algebraic geometric code can not exceed its designed minimum distance by more than g, where g is the genus of the curve. This implies that the minimum in the order bound occurs for $i \in \{0, \ldots, g\}$.

For a curve X defined over the field $\mathbb{F}$, let $\mathrm{Pic}(X)$ be the group of divisor classes. Let $\Gamma = \{A : L(A) \neq 0\}$ be the semigroup of effective divisor classes. For a divisor class C, define

$$\Gamma(C) = \{A : L(A) \neq 0 \text{ and } L(A-C) \neq 0\},$$
$$\Gamma^*(C) = \{A : L(A) \neq 0 \text{ and } L(A-C) \neq L(-C)\}.$$

The semigroup $\Gamma(C)$ has the property that $A + E \in \Gamma(C)$ whenever $A \in \Gamma(C)$ and $E \in \Gamma$. With the extra structure $\Gamma(C)$ is a semigroup ideal. Similar for $\Gamma^*(C)$.

For a suitable choice of divisor class C, the subsets of coordinates A that support a codeword in an algebraic geometric code $C_L(D,G)$ or $C_\Omega(D,G)$ belong to the semigroup ideals $\Gamma^*(C) \subseteq \Gamma(C)$.

$$c \in C_L(D,G) \setminus \{0\} \;\Rightarrow\; A = \mathrm{supp}(c) \in \Gamma^*(C) \subseteq \Gamma(C),$$
$$c \in C_\Omega(D,G) \setminus \{0\} \;\Rightarrow\; A = \mathrm{supp}(c) \in \Gamma^*(C) \subseteq \Gamma(C).$$

Following [7], our approach from here on will be to estimate the minimal degree of a divisor $A \in \Gamma^*(C)$. For that purpose we no longer need to refer to the codes $C_L(D,G)$ or $C_\Omega(D,G)$ after we choose $C = D - G$ or $C = G - K$, respectively. We write $\mathcal{C}(C)$ to refer to any code with designed minimum support $D - G$ or $G - K$ in the divisor class C. In this short paper, we restrict ourselves to the case $\deg C > 0$ (i.e. to codes with positive Goppa designed minimum distance), so that $L(-C) = 0$, and $\Gamma^*(C) = \Gamma(C)$. The case $\deg C \leq 0$ is handled with a straightforward modification similar to that used in [7].

To apply the order bound argument we restate Proposition 2 in terms of semigroup ideals. For a given point $P \in X$, let $\Gamma_P = \{A : L(A) \neq L(A-P)\}$ be

the semigroup of effective divisor classes with no base point at P. For a divisor class C and for a point P, define the semigroup ideal

$$\Gamma_P(C) = \{A : L(A) \neq L(A-P) \text{ and } L(A-C) \neq L(A-C-P)\}.$$

The implications in Proposition 2 become

$$\begin{aligned} c \in C_L(D,G) \backslash C_L(D,G-P) &\Rightarrow A = \operatorname{supp}(c) \in \Gamma_P(C), \\ c \in C_\Omega(D,G) \backslash C_\Omega(D,G+P) &\Rightarrow A = \operatorname{supp}(c) \in \Gamma_P(C). \end{aligned}$$

Let $\Delta_P(C)$ be the complement of $\Gamma_P(C)$ in Γ_P,

$$\Delta_P(C) = \{A : L(A) \neq L(A-P) \text{ and } L(A-C) = L(A-C-P)\}.$$

Theorem 1. *(Duursma-Park [7]) Let $\{A_1 \leq A_2 \leq \cdots \leq A_w\} \subset \Delta_P(C)$ be a sequence of divisors with $A_{i+1} \geq A_i + P$, for $i = 1, \ldots, w-1$. Then $\deg A \geq w$, for every divisor $A \in \Gamma_P(C)$ with support disjoint from $A_w - A_1$.*

Proof. A more general version is proved in the next section.

3 Order Bounds

Following [1], we use the order bound with a sequence of points $\{Q_i : i \geq 0\}$. For $R_i = Q_0 + Q_1 + \cdots + Q_{i-1}$, $i \geq 0$,

$$\begin{aligned} c \in C_L(D,G-R_i) \backslash C_L(D,G-R_i-Q_i) &\Rightarrow A = \operatorname{supp}(c) \in \Gamma_{Q_i}(C+R_i), \\ c \in C_\Omega(D,G+R_i) \backslash C_\Omega(D,G+R_i+Q_i) &\Rightarrow A = \operatorname{supp}(c) \in \Gamma_{Q_i}(C+R_i). \end{aligned}$$

Repeated application of Theorem 1 gives the following bound for the minimum distance.

Corollary 1. *(Duursma-Park bound d_{DP} for the minimum distance) Let $\deg A \geq w$ for $A \in \Gamma_{Q_i}(C+R_i)$, $i \geq 0$ Then a code with designed minimum support in the divisor class C and with divisor D disjoint from $A_w - A_1$ as required by the application of Theorem 1, has minimum distance at least w.*

The Beelen bound d_B for the minimum distance [1, Theorem 7, Remark 5] is similar but assumes in the application of Theorem 1 that $A_w - A_1$ has support in the single point Q_i, for a seqeunce $\{A_1 \leq A_2 \leq \cdots \leq A_w\} \subset \Delta_{Q_i}(C+R_i)$.

The Beelen bound d_B has a weaker version (which we will denote by d_{B_0}) that assumes that moreover A_1 has support in the point Q_i. The simple or first order bound uses the further specialization that $Q_i = P$ for $i \geq 0$. For $G = K + C$, such that $\deg C > 0$, and for D disjoint from P, the simple order bound becomes

$$d(C_L(D,G)^\perp) \geq \min_{i \geq 0} \#(\Delta_P(C+iP) \cap \{jP : j \geq 0\}).$$

For $G = mP$ this is the original Feng-Rao bound.

The purpose of the order bound is to improve on uniform bounds such as the floor bound. In rare occasions the Beelen bound is less than bounds of floor type [1], [7] (this is the case for a single code in Table 1). Compared to the Beelen bound, the ABZ order bound $d_{ABZ'}$ [7, Theorem 6.6] allows $A_{j+1} - A_j \notin \{kQ_i : k > 0\}$ for a single j in the range $j = 1, \ldots, w-1$. With that modification the ABZ order bound always is at least the ABZ floor bound d_{ABZ} [7, Theorem 2.4] which is the best known bound of floor type. In general, $d_{B_0} \leq d_B \leq d_{ABZ'} \leq d_{DP}$ and $d_{ABZ} \leq d_{ABZ'}$.

4 Extension of the Order Bound

We seek to exploit the argument in the order bound a step further by using a partition

$$\begin{aligned} & C_L(D, G - iP) \backslash C_L(D, G - (i+1)P) \\ = \; & \cup_{j \geq 0} C_L(D, G - iP - jQ) \backslash \\ & \quad (C_L(D, G - (i+1)P - jQ) \cup C_L(D, G - iP - (j+1)Q). \end{aligned}$$

We apply the argument in the setting of the divisor semigroups. For a finite set S of rational points, let $\Gamma_S = \cap_{P \in S} \Gamma_P$, and let $\Gamma_S(C) = \{A \in \Gamma_S : A - C \in \Gamma_S\}$, so that $\Gamma_S(C) = \cap_{P \in S} \Gamma_P(C)$. Let $\Delta_S(C) = \cup_{P \in S} \Delta_P(C)$.

Proposition 3. *For a divisor class C, for a finite set of rational points S, and for $P \notin S$,*

$$\Gamma_S(C) \cap \Gamma_P = \cup_{i \geq 0} \Gamma_{S \cup P}(C + iP).$$

Proof. Since $P \notin S$, $P \in \Gamma_S$, and, for $A - C - iP \in \Gamma_S$, $A - C \in \Gamma_S$. Therefore

$$\Gamma_{S \cup P}(C + iP) = \Gamma_S(C + iP) \cap \Gamma_P(C + iP) \subseteq \Gamma_S(C) \cap \Gamma_P.$$

For the other inclusion, let $A \in \Gamma_S(C) \cap \Gamma_P$. Then $A - C \in \Gamma_S$ and $L(A - C) \neq 0$. Choose $i \geq 0$ maximal such that $L(A - C) = L(A - C - iP)$. Then $A - C - iP \in \Gamma_{S \cup P}$.

As a special case $\Gamma(C) \cap \Gamma_P = \cup_{i \geq 0} \Gamma_P(C + iP)$. Theorem 1 gives a lower bound for $\deg A$, for $A \in \Gamma_P(C)$. Combination of the lower bounds for $C \in \{C + iP : i \geq 0\}$ then gives a lower bound for $\deg A$, for $A \in \Gamma(C) \cap \Gamma_P$.

In combination with $\Gamma(C + iP) \cap \Gamma_Q = \cup_{j \geq 0} \Gamma_{\{P,Q\}}(C + iP + jQ)$, we obtain, for $S = \{P, Q\}$,

$$\Gamma(C) \cap \Gamma_S = \cup_{i,j \geq 0} \Gamma_S(C + iP + jQ).$$

The next theorem gives a lower bound for $\deg A$, for $A \in \Gamma_S(C)$. For $S = \{P, Q\}$, combination of the lower bounds for $C \in \{C + iP + jQ : i, j \geq 0\}$ then gives a lower bound for $\deg A$, for $A \in \Gamma(C) \cap \Gamma_S$.

Theorem 2. *(Main theorem) Let $\{A_1 \leq A_2 \leq \cdots \leq A_w\} \subset \Delta_S(C)$ be a sequence of divisors with $A_i \in \Delta_{P_i}(C)$, $P_i \in S$, for $i = 1, \ldots, w$, such that $A_i - P_i \geq A_{i-1}$ for $i = 2, \ldots, w$. Then $\deg A \geq w$, for every divisor $A \in \Gamma_S(C)$ with support disjoint from $A_w - A_1$.*

Lemma 1. *For $D' \in \Gamma_P(C)$, $\Delta_P(C) \subseteq \Delta_P(D')$.*

Proof. For $D' - C \in \Gamma_P$, if $A - C \notin \Gamma_P$ then $A - D' \notin \Gamma_P$.

Lemma 2. *Let $l_C(A) = l(A) - l(A - C)$. Then*

$$A \in \Delta_P(C) \;\Leftrightarrow\; l_C(A) - l_C(A - P) = 1.$$

Proof.

$$\begin{aligned} l_C(A) - l_C(A-P) = 1 &\Leftrightarrow (l(A) - l(A-P)) - (l(A-C) - l(A-C-P)) \;=\; 1 \\ &\Leftrightarrow l(A) - l(A-P) \;=\; 1 \wedge l(A-C) - l(A-C-P) \;=\; 0 \Leftrightarrow A \in \Delta_P(C). \end{aligned}$$

Proof. (Theorem 2) For $A = D' \in \Gamma_S(C) \subseteq \Gamma_{P_i}(C)$ and for $A_i \in \Delta_{P_i}(C)$, $A_i \in \Delta_{P_i}(D')$, by Lemma 1, and $l_{D'}(A_i) = l_{D'}(A_i - P_i) + 1$ by Lemma 2. With $A_i - P_i \geq A_{i-1}$, there exists a natural map

$$L(A_{i-1})/L(A_{i-1} - D') \longrightarrow L(A_i - P_i)/L(A_i - P_i - D').$$

With $(A_i - P_i) - A_{i-1}$ disjoint form D', the map is injective, since $L(A_{i-1}) \cap L(A_i - P_i - D') = L(A_{i-1} - D')$. So that $l_{D'}(A_i - P_i) \geq l_{D'}(A_{i-1})$. Iteration over i yields

$$\deg D' \geq l_{D'}(A_w) \geq l_{D'}(A_{w-1}) + 1 \geq \cdots \geq l_{D'}(A_1) + w - 1 \geq l_{D'}(A_1 - P_1) + w.$$

To obtain lower bounds with the theorem, we need to construct sequences of divisors in $\Delta_S(C)$. In the next section we discuss how this can be done effectively.

5 Efficient Computation of the Bounds

The main theorem can be used efficiently to compute coset distances, which in turn can be used to compute two-point code distances. To compute with a certain curve we use, for given points P and Q, a function $d_{P,Q}$ that encapsulates some geometric properties of the curve [7], see also [2], [14].

Lemma 3. *Let B be a divisor and let P, Q be distinct points. There exists a unique integer $k(B, P, Q)$ such that $B + k'P \in \Gamma_Q$ if and only if $k' \geq k(B, P, Q)$*

Proof. This amounts to showing that $B + kP \in \Gamma_Q$ implies $B + (k+1)P \in \Gamma_Q$. Now use that Γ_Q is a semigroup and that $P \in \Gamma_Q$.

Let us restrict our attention to two-point divisors. Using the previous notation define the following integer valued function.

$$d_{P,Q}(a) = k(aQ, P, Q) + a$$

Theorem 3. *For a divisor A with support in $\{P, Q\}$,*

$$A \in \Gamma_Q \;\Leftrightarrow\; \deg(A) \geq d_{P,Q}(A_Q).$$

Proof. Let $A = kP + aQ$, then $\deg(A) \geq d_{P,Q}(a)$ is equivalent to $k \geq k(aQ, P, Q)$, which by definition is $aQ + kP \in \Gamma_Q$.

This property makes the d function a powerful computational tool. Moreover, for m such that $mP \sim mQ$, d is defined modulo m. In general, the function d depends on the ordering of the points P and Q, but it is easy to see that the functions $d_{P,Q}$ and $d_{Q,P}$ satisfy the relation $d_{P,Q}(a) = a + b$ if and only if $d_{Q,P}(b) = a + b$. The function $d = d_{P,Q}$ and the parameter m are enough to compute sequences of two-point divisors $A_i \in \Delta_P(C)$ as required by Theorem 1. A simple application of the Riemann-Roch Theorem give us that we can restrict our search to a finite range of divisors, since, for $A \in \Delta_P(C)$,

$$\min\{0, \deg C\} \;\leq\; \deg A \;\leq\; \max\{2g - 1, \deg C + 2g - 1\}.$$

Using $mP \sim mQ$, we can assume moreover, for $A = A_P P + A_Q Q$, that $A_Q \in [0, \ldots, m - 1]$.

To find long sequences of divisors $A_i \in \Delta_P(C)$ we use a graph theory weight-maximizing algorithm on a rectangular grid T, such that $T_{i,j}$ is a path of longest length up to the divisor A with degree i and $A_Q = j$ (i.e. $A = iP + j(Q - P)$).

Computing bounds for the coset $\mathcal{C}(C)\backslash\mathcal{C}(C + P)$ with Theorem 1

1. Initialize the first row of T (corresponding to degree $i = \min\{0, \deg C\} - 1$) with 0.
2. Update each row of T successively by the rule

   $$T_{i,j} = \max\{T_{i-1,j-1}, T_{i-1,j} + BP_{i,j}\}$$

   where $BP_{i,j}$ is 1 if $A \in \Delta_P(C)$ and 0 otherwise, for the divisor A of degree i with $A_Q = j$. $BP_{i,j}$ is computed using Theorem 3.
3. Iterate up to the last row (corresponding to degree $i = \max\{2g - 1, \deg C + 2g - 1\}$).
4. Return the maximum value in the last row.

Using the algorithm we compute bounds for the cosets $\mathcal{C}(C, S)\backslash\mathcal{C}(C + P, S)$ and $\mathcal{C}(C, S)\backslash\mathcal{C}(C + Q, S)$ over all possible divisors C. We store them in arrays CP and CQ where the row denotes the degree of C and the column is $C_Q \pmod m$. For rational points P and Q such that $d_{P,Q} = d_{Q,P}$, we can save some work and obtain the table CQ from CP with the relabeling $CQ_{i,j} = CP_{i,j'}$ for $j' = i - j \pmod m$. After computing the coset bounds, we traverse all possible coset filtrations of all codes to find bounds for the minimum distances. We use a graph theory flow-maximizing algorithm on a rectangular grid D, such that $D_{i,j}$ is a bound for the minimum distance of a code $\mathcal{C}(C)$ with C of degree i and $C_Q = j$.

Computing bounds for the distances of all codes $\mathcal{C}(C)$ using P-coset and Q-coset tables

1. Initialize the last row of D (corresponding to degree $i = 2g$) with $2g$.

2. Update each row of D successively by the rule

$$D_{i,j} = \max\{\min\{D_{i+1,j}, CP_{i,j}\}, \min\{D_{i+1,j+1}, CQ_{i,j}\}\}$$

3. Iterate up to the first row (corresponding to degree $i = 0$).

Theorem 2 can be used to obtain improved lower bounds for the cosets $\mathcal{C}(C)\backslash\mathcal{C}(C+P)$ and $\mathcal{C}(C)\backslash\mathcal{C}(C+Q)$. The basic algorithm is the same as before with an extra step of using the table of all bounds for $\mathcal{C}(C,S)\backslash \cup_{P\in S}\mathcal{C}(C+P,S)$ to produce the P- and Q-coset tables.

Computing P- and Q-coset bounds using Theorem 2

1. Compute table CS with bounds for $\mathcal{C}(C,S)\backslash \cup_{P\in S}\mathcal{C}(C+P,S)$. This step is done in exactly the same way as the computation for Theorem 1, but the new table T for each C has the update rule

$$T_{i,j} = \max\{T_{i-1,j-1} + BQ_{i,i-j}, T_{i-1,j} + BPi, j\}$$

2. Initialize CP and CQ at the top row (corresponding to degree $i = 2g$) with $2g$.
3. Compute CP in decreasing row order using the rule $CP_{i,j} = \min\{CP_{i+1,j+1}, T_{i,j}\}$.
4. Compute CQ in decreasing row order using the rule $CQ_{i,j} = \min\{CQ_{i+1,j}, T_{i,j}\}$.

Once the P- and Q-coset bounds are known, exactly the same minimizing flow method as the one used for obtaining d_{DP} can be used for an improved bound, denoted d_{DK}.

6 Tables for the Suzuki Curve over $\mathbb{F}_{32}$

The Suzuki curve over the field of $q = 2q_0^2$ elements is defined by the equation $y^q + y = x_0^q(x^q + x)$. The curve has $q^2 + 1$ rational points and genus $g = q_0(q-1)$. The semigroup of Weierstrass nongaps at a rational point is generated by $\{q, q+q_0, q+2q_0, q+2q_0+1\}$. For any two rational points P and Q there exists a function with divisor $(q+2q_0+1)(P-Q)$. Let $m = q+2q_0+1 = (q_0+1)^2+q_0{}^2$, and let H be the divisor class containing $mP \sim mQ$. The canonical divisor $K \sim 2(q_0-1)H$. For the Suzuki curve over the field of 32 elements we use $q_0 = 4, q = 32, g = 124, m = 41, K \sim 6H$. The action of the automorphism group on the rational points of the curve is 2-transitive, so that $d_{P,Q}(a) = d(a)$ does not depend on the choice of the points P and Q. For the Suzuki curve with parameter q_0, the d function is given by

$$d(k) = (q_0 - a)(q-1)$$

where a,b are the unique integers such that $|a|+|b| \leq q_0$ and $k \equiv a(q_0+1)+bq_0 - q_0(q_0+1) \pmod{m}$. A detailed explanation of the geometry behind this result

can be found in [7]. To store the function d as a list, we go through all integers a and b with $|a| + |b| \leq q_0$. To compute $d(k)$ for a single value k, we may use

$$d(k) = (q-1)(2q_0 - q(k-1, 2q_0+1)+ \\ - q(r(k-1, 2q_0+1), q_0+1) - r(r(k-1, 2q_0+1), q_0+1)),$$

where $q(a, b)$ and $r(a, b)$ are the quotient and the remainder, respectively, when a is divided by b, and $k-1$ is taken modulo m.

Table 1. Number of improvements of one bound over another (top), and the maximum improvement (bottom), based on 10168 two-point codes for the Suzuki curve over $\mathbb{F}_{32}$

		Floor bounds			Order bounds				
	d_{BPT}	d_{LM}	d_{GST}	d_{ABZ}	d_{B_0}	d_B	$d_{ABZ'}$	d_{DP}	d_{DK}
d_{GOP}	6352	6352	6352	6352	6352	6352	6352	6352	6352
d_{BPT}	·	4527	4551	4597	5260	5264	5264	5264	5274
d_{LM}	·	·	2245	2852	4711	4729	4731	4731	4757
d_{GST}	·	·	·	2213	4711	4729	4731	4731	4757
d_{ABZ}	·	·	·	·	4665	4683	4685	4685	4711
d_{B_0}	·	1	1	1	·	176	374	412	1643
d_B	·	1	1	1	·	·	198	236	1565
$d_{ABZ'}$	·	·	·	·	·	·	·	38	1404
d_{DP}	·	·	·	·	·	·	·	·	1366
d_{GOP}	1	8	13	21	33	33	33	33	33
d_{BPT}	·	7	12	20	32	32	32	32	32
d_{LM}	·	·	7	15	28	28	28	28	28
d_{GST}	·	·	·	8	24	24	24	24	24
d_{ABZ}	·	·	·	·	24	24	24	24	24
d_{B_0}	·	1	1	1	·	1	5	5	6
d_B	·	1	1	1	·	·	5	5	6
$d_{ABZ'}$	·	·	·	·	·	·	·	1	6
d_{DP}	·	·	·	·	·	·	·	·	6

Table 2. Improvements of d_{DP} and d_{DK} over d_B for 10168 two-point codes

$d_{DK} - d_{DP} =$		0	1	2	3	4	5	6
$d_{DP} - d_B$	= 0	8603	656	356	198	50	6	63
	1	92	12	0	0	0	0	0
	2	33	4	0	0	0	0	0
	3	74	4	1	0	0	0	0
	4	0	0	0	0	0	0	0
	5	0	16	0	0	0	0	0

Table 3. Optimal codes (For given $\deg C = 2, \ldots, 124\,(= g)$, d_{DK} is the maximum lower bound for a two-point code defined with $C = C_P P + C_Q Q$, and C_Q gives values for which the maximum is achieved. Suppressed are divisors that define subcodes with the same minimum distance as already listed codes. Exchanging P and Q gives a similar code and listed are only divisors with $C_Q \pmod m \leq C_P \pmod m$. The last columns give the amount by which d_{DK} exceeds similarly defined maximum lower bounds for d_{DP} and d_B, respectively.

$\deg C$	d_{DK}	C_Q			$\deg C$	d_{DK}	C_Q			$\deg C$	d_{DK}	C_Q		
2	31	[5]	·	·	43	62	[5]	·	·	84	94	[1]	·	1
3	31		·	·	44	62		·	·	85	94	[0, 5]	·	·
4	31		·	·	45	63	[5]	·	·	86	96	[1]	·	1
5	32	[0]	·	·	46	64	[0]	·	·	87	96	[0]	·	·
6	32		·	·	47	64		·	·	88	96		·	·
7	32		·	·	48	64		·	·	89	96		·	·
8	38	[2]	·	3	49	68	[2]	·	2	90	99	[1, 2]	·	·
9	38		·	3	50	68		·	2	91	100	[2, 3]	·	·
10	38		·	3	51	70	[2, 4]	·	2	92	102	[3]	·	·
11	44	[1]	·	·	52	72	[1]	·	·	93	102		·	·
12	44	[6]	·	·	53	72	[4, 6]	·	·	94	103	[5]	·	·
13	44		·	·	54	72		·	·	95	104	[2]	·	·
14	44		·	·	55	74	[7]	2	2	96	106	[7]	·	·
15	48	[5]	·	1	56	75	[7]	2	3	97	106		·	·
16	48		·	·	57	75		·	2	98	106		·	·
17	48		·	·	58	75		·	·	99	108	[2, 6]	·	·
18	50	[8]	·	1	59	75		·	·	100	110	[7]	·	·
19	50		·	·	60	77	[8]	·	·	101	110		·	·
20	51	[9]	·	1	61	77		·	·	102	110		·	·
21	54	[10]	·	3	62	78	[10]	·	·	103	112	[6]	·	·
22	54		·	3	63	79	[10]	1	1	104	112	[2]	·	·
23	54		·	1	64	80	[11]	2	2	105	114	[7]	1	1
24	54		·	3	65	80	[8]	·	·	106	114	[9]	1	1
25	54		·	·	66	81	[5]	1	1	107	114		·	·
26	54		·	·	67	83	[10]	1	1	108	116	[2, 6, 7]	·	·
27	54		·	·	68	85	[11]	2	2	109	118	[7]	2	2
28	56	[12]	·	1	69	85		1	1	110	118		2	2
29	56		·	·	70	85	[5, 9]	1	1	111	118		1	1
30	58	[13]	2	2	71	87	[10]	3	3	112	120	[6]	1	1
31	58	[10]	2	2	72	87		2	2	113	120	[2]	1	1
32	58		2	2	73	87	[7, 16]	·	·	114	121	[7]	2	2
33	59	[12]	3	3	74	87		·	·	115	121	[2, 6, 9]	1	1
34	61	[13]	3	3	75	89	[10]	1	1	116	122	[2, 6, 7, 10, 15]	·	·
35	62	[10 14]	3	3	76	91	[10, 11]	3	3	117	123	[2, 6, 7]	1	1
36	62	[17]	3	3	77	91		1	1	118	124	[2, 6, 7]	2	2
37	62		3	3	78	91	[8]	1	1	119	124		1	1
38	62		2	2	79	91	[5, 7]	·	·	120	124	[1, 5, 10, 19]	·	·
39	62		2	2	80	93	[10, 11]	2	2	121	125	[1]	·	·
40	62		2	2	81	93		2	2	122	126	[1]	·	·
41	62		2	2	82	93		1	1	123	127	[1]	·	·
42	62		2	2	83	93	[7]	·	1	124	128	[0]	·	·

Table 1 compares the Goppa bound d_{GOP}, the base point bound d_{BPT} (an improvement of the Goppa bound by one whenever C has a base point), the floor bounds d_{LM} [16, Theorem 3], d_{GST} [11, Theorem 2.4], d_{ABZ} [7, Theorem 2.4], and the order bounds $d_{B_0}, d_B, d_{ABZ'}, d_{DP}, d_{DK}$. Each bound was optimized over the full parameter space corresponding to that bound. The computations for the Suzuki curve of genus $g = 124$ over $\mathbb{F}_{32}$ were very efficient, computations for the $2g \cdot m = 10168$ two-point codes with Goppa designed minimum distance in the range $0, \ldots, 2g-1$, took less than 10 minutes on a desktop PC for any given bound.

As can be seen, the Beelen bound d_B offers only slight improvement over the weaker Beelen bound d_{B_0}. Similar for the improvement of the Duursma-Park bound d_{DP} over the weaker ABZ bound $d_{ABZ'}$. Table 2 gives a further breakdown of the 236 codes for which d_{DP} improves d_B and the 1366 codes for which d_{DK} improves d_{DP}. The improvements are by at most 5 and 6, respectively.

The 63 codes with $d_{DK} = d_{DP} + 6$ all have $d_{DK} = 62$ which agrees with the actual minimum distance (namely realized by a choice of 62 points with $P_1 + \ldots + P_{62} \sim 31P + 31Q$). For each of the 63 codes, the coset with $C = 23P + 23Q$ is the unique coset where Theorem 1 fails to give a lower bound above 56. For the same coset, Theorem 2 gives a lower bound of 62, for example with a sequence $A_1, \ldots, A_{62}$ of type

$$\begin{aligned}\{A_1 = 24Q, \ldots, A_{24} = 114P + 24Q\}\ (\subseteq \Delta_P(C))\\ \cup \{A_{25} = 114P + 25Q, \ldots, A_{38} = 114P + 40Q\}\ (\subseteq \Delta_Q(C))\\ \cup \{A_{39} = 156P, \ldots, A_{62} = 270P\}\ (\subseteq \Delta_P(C)).\end{aligned}$$

Table 1 and Table 2 do not show whether the improvements occur for good codes or for poor codes. For Table 3 we select for each degree $\deg C$, i.e. for each given Goppa designed minimum distance, the optimal code with respect to each of the bounds d_B, d_{DP}, d_{DK} and we compare those. In this case, depending on the degree, the improvements of d_{DK}, obtained with Theorem 2, over the bounds d_B and d_{DP} vary between 0 and 3.

References

1. Beelen, P.: The order bound for general algebraic geometric codes. Finite Fields Appl. 13(3), 665–680 (2007)
2. Beelen, P., Tutaş, N.: A generalization of the Weierstrass semigroup. J. Pure Appl. Algebra 207(2), 243–260 (2006)
3. Bras-Amorós, M., O'Sullivan, M.E.: On semigroups generated by two consecutive integers and improved Hermitian codes. IEEE Trans. Inform. Theory 53(7), 2560–2566 (2007)
4. Campillo, A., Farrán, J.I., Munuera, C.: On the parameters of algebraic-geometry codes related to Arf semigroups. IEEE Trans. Inform. Theory 46(7), 2634–2638 (2000)
5. Carvalho, C., Torres, F.: On Goppa codes and Weierstrass gaps at several points. Des. Codes Cryptogr. 35(2), 211–225 (2005)

6. Chen, C.-Y., Duursma, I.M.: Geometric Reed-Solomon codes of length 64 and 65 over $\mathbb{F}_8$. IEEE Trans. Inform. Theory 49(5), 1351–1353 (2003)
7. Duursma, I., Park, S.: Coset bounds for algebraic geometric codes, 36 pages (2008) (submitted), arXiv:0810.2789
8. Duursma, I.M.: Majority coset decoding. IEEE Trans. Inform. Theory 39(3), 1067–1070 (1993)
9. Duursma, I.M.: Algebraic geometry codes: general theory. In: Munuera, C., Martinez-Moro, E., Ruano, D. (eds.) Advances in Algebraic Geometry Codes. Series on Coding Theory and Cryptography. World Scientific, Singapore (to appear)
10. Feng, G.L., Rao, T.R.N.: Decoding algebraic-geometric codes up to the designed minimum distance. IEEE Trans. Inform. Theory 39(1), 37–45 (1993)
11. Güneri, C., Sitchtenoth, H., Taskin, I.: Further improvements on the designed minimum distance of algebraic geometry codes. J. Pure Appl. Algebra 213(1), 87–97 (2009)
12. Hansen, J.P., Stichtenoth, H.: Group codes on certain algebraic curves with many rational points. Appl. Algebra Engrg. Comm. Comput. 1(1), 67–77 (1990)
13. Høholdt, T., van Lint, J.H., Pellikaan, R.: Algebraic geometry of codes. In: Handbook of coding theory, vol. I, II, pp. 871–961. North-Holland, Amsterdam (1998)
14. Kim, S.J.: On the index of the Weierstrass semigroup of a pair of points on a curve. Arch. Math. (Basel) 62(1), 73–82 (1994)
15. Kirfel, C., Pellikaan, R.: The minimum distance of codes in an array coming from telescopic semigroups. IEEE Trans. Inform. Theory 41(6, part 1), 1720–1732 (1995); Special issue on algebraic geometry codes
16. Lundell, B., McCullough, J.: A generalized floor bound for the minimum distance of geometric Goppa codes. J. Pure Appl. Algebra 207(1), 155–164 (2006)
17. Maharaj, H., Matthews, G.L.: On the floor and the ceiling of a divisor. Finite Fields Appl. 12(1), 38–55 (2006)
18. Matthews, G.L.: Weierstrass pairs and minimum distance of Goppa codes. Des. Codes Cryptogr. 22(2), 107–121 (2001)
19. Matthews, G.L.: Codes from the Suzuki function field. IEEE Trans. Inform. Theory 50(12), 3298–3302 (2004)

Sparse Numerical Semigroups

C. Munuera[1], F. Torres[2], and J. Villanueva[2]

[1] Dept. of Applied Mathematics, University of Valladolid
Avda Salamanca SN, 47012 Valladolid,
Castilla, Spain
[2] IMECC-UNICAMP, Cx.P. 6065, 13083-970, Campinas-SP, Brasil

Abstract. We investigate the class of numerical semigroups verifying the property $\rho_{i+1} - \rho_i \geq 2$ for every two consecutive elements smaller than the conductor. These semigroups generalize Arf semigroups.

1 Introduction

Let $\mathbf{N}_0$ be the set of nonnegative integers and $H = \{0 = \rho_1 < \rho_2 < \cdots\} \subseteq \mathbf{N}_0$ be a numerical semigroup of finite genus g. This means that the complement $\mathbf{N}_0 \setminus H$ is a set of g integers called *gaps*, $\mathrm{Gaps}(H) = \{\ell_1, \ldots, \ell_g\}$. Then $c = \ell_g + 1$ is the smallest integer such that $c + h \in H$ for all $h \in \mathbf{N}_0$. The number c is called the *conductor* of H and clearly $c = \rho_{c-g+1}$.

Semigroups play an important role in the theory of one point algebraic geometry codes. In fact, the computation of the *order bound* on the minimum distance of such a code $C(\mathcal{X}, D, mQ)$ –constructed from a curve $\mathcal{X}$, a divisor D on $\mathcal{X}$ and a point Q of $\mathcal{X}$– involves computations in the Weierstrass semigroup associated to Q (see Section 5 and [6] for more details). Some families of semigroups have been studied in deep from this point of view. Relevant for this article are Arf and inductive semigroups. Let us remember that the semigroup H is said to be *Arf* if $\rho_i + \rho_j - \rho_k \in H$ for $i \geq j \geq k \geq 0$. The study of Arf semigroups has been carried in the context of Coding Theory, [3],[5], and Local Algebra Theory, [1], [2], [7], among others (see [12] and the references therein). Inductive semigroups are a particular class of Arf semigroups ([5]). A sequence (S_n) of semigroups is called *inductive* if there exist sequences of positive integers (a_n) and (b_n) such that $S_1 = \mathbf{N}_0$ and for $n > 1$, $S_n = a_n S_{n-1} \cup \{m \in \mathbf{N}_0 : m \geq a_n b_n\}$. A semigroup is called *inductive* if it is a member of an inductive sequence.

Note that the Arf condition implies (see Corollary 1 below)

$$\ell_i - \ell_{i-1} \leq 2, \quad \text{for} \quad i = 2, \ldots, g, \tag{1}$$

or equivalently $\rho_{i+1} - \rho_i \geq 2$ for $i = 1, \ldots, c-g$. There are numerical semigroups satisfying (1) which are not Arf, e.g. those given in Example 1. This fact provides a motivation to introduce and study the class of semigroups for which (1) holds. These will be called *sparse*. Let us remark that there exist other generalizations of Arf semigroups, proposed by different authors. For example, in ([4]) the Arf property is extended to a widely class of semigroups via the concept of *patterns* of semigroups. The sparse and pattern properties are disjoint in general.

M. Bras-Amorós and T. Høholdt (Eds.): AAECC 2009, LNCS 5527, pp. 23–31, 2009.

From a geometrical point of view, sparse semigroups are closely related to Weierstrass semigroups arising in double covering of curves, cf. [14]. Its arithmetical structure is strongly influenced by the parity of its largest gap ℓ_g. Recall that for all semigroups we have $\ell_g \leq 2g-1$ ([9]). When equality holds then the semigroup is called *symmetric*. For even ℓ_g we consider the family

$$\mathcal{H}_{g,k} := \{H : H \text{ is a sparse semigroup of genus } g \text{ and } \ell_g = 2g-2k\}. \quad (2)$$

For each $H \in \mathcal{H}_{g,k}$ we prove that $g \leq 6k-3$ (Theorem 2); this was noticed by Barucci et al. [2, Thm. I.4.5] and Bras-Amorós [3, Prop. 3.2] when $k=1$. The approach used here is quite different from theirs. In particular we show that $\cup_g \mathcal{H}_{g,k}$ is finite. For ℓ_g odd, write $\ell_g = 2g-(2\gamma+1)$ with $\gamma \geq 0$ an integer. We prove (Theorem 3) that for g large enough with respect to γ, H is of the form

$$H = 2\bar{H} \cup \{h \in \mathbf{N} : h \geq 2g-2\gamma+1\}, \quad (3)$$

where $\bar{H}$ is a semigroup of genus γ uniquely determined by H. When $\gamma = 0$ then H is hyperelliptic, that is $2 \in H$ (which has already been noticed in [2], [3], [5]). Finally in Section 5 we observe that sparse semigroups of genus large enough satisfy the *acute* property in the sense of Bras-Amorós [3], and we obtain the order of such semigroups as a function of g and ℓ_g (Proposition 5).

2 Sparse Semigroups

Let $H = \{0 = \rho_1 < \rho_2 < \cdots\}$ be a semigroup of genus g and conductor c. Our starting point is the following result (see [5], Proposition 1).

Proposition 1. *H is Arf iff for all integers $i \geq j \geq 1$, it holds that $2\rho_i - \rho_j \in H$.*

Corollary 1. *The set of gaps of an Arf semigroup H of genus g satisfy (1).*

Proof. Let $\text{Gaps}(H) = \{\ell_1, \ldots, \ell_g\}$. Suppose there exists $i \in \{2, \ldots, g\}$ such that $\ell_i - \ell_{i-1} > 2$. Then $\ell_i - 2$ and $\ell_i - 1$ are non-gaps of H and Proposition 1 implies $2(\ell_i - 1) - (\ell_i - 2) = \ell_i \in H$, a contradiction.

Example 1. Let $g \geq 7$. The set

$$H = \mathbf{N}_0 \setminus \{1, \ldots, g-3, g-1, g+1, g+2\} = \{0, g-2, g, g+3, g+4, \ldots\}$$

is a semigroup of genus g satisfying (1). Since $2g-(g-2) = g+2 \notin H$ it is not Arf.

Definition 1. *A semigroup $H = \{0 = \rho_1 < \rho_2 < \ldots\}$ of genus g is called* sparse *if the set of its gaps,* $\text{Gaps}(H)$*, satisfies (1).*

Equivalently, H is sparse if $\rho_{i+1} - \rho_i \geq 2$ for $i = 1, \ldots, c-g$. Obviously, every Arf semigroup is sparse. Other remarkable examples of sparse semigroups are as follows.

Example 2. Let $\bar{H}$ be an arbitrary semigroup of conductor $\bar{c}$ and genus γ.

(a) Given two integers $a \geq 2$, $R \geq 0$, the semigroup $H := a\bar{H} \cup \{h \in \mathbf{N}_0 : h \geq R\}$ is sparse. In particular, inductive semigroups are sparse. Note that when $a = 2$ and $R \geq 2\bar{c}$, then H is Arf iff $\bar{H}$ is Arf.
(b) Let g be an integer such that $g \geq \gamma + \bar{c}$. Then $H = 2\bar{H} \cup \{h \in \mathbf{N} : h \geq 2g - 2\gamma + 1\}$ is a sparse semigroup of genus g with $\ell_g = 2g - (2\gamma + 1)$, whose number of even gaps is γ. In Section 4 we will show that every sparse semigroup with ℓ_g odd and genus large enough with respect to γ arise in this way.

For a semigroup H, let $m = m(H) = \rho_2 - 1$. When $H \neq \mathbf{N}_0$, m is the largest integer such that $\ell_m = m$. Then $m(H) = 1$ iff H is hyperelliptic.

Proposition 2. *Let H be a symmetric semigroup of genus $g > 0$. The following statements are equivalent:*

1. *H is Arf;*
2. *H is sparse;*
3. *$m(H) = 1$.*

Proof. The symmetry property implies $\ell_{g-1} = \ell_g - \rho_2$; then from inequalities (1) it follows that $\rho_2 = 2$. This shows that (2) implies (3). If $2 \in H$, assertion (1) follows from Example 2 as $H = 2\mathbf{N}_0 \cup \{n \in \mathbf{N} : n \geq 2g\}$.

Proposition 3. *Let H be a semigroup with $m(H) = 2$ and $g > 2$. The following statements are equivalent:*

1. *H is Arf;*
2. *H is sparse;*
3. *$H = \langle 3, \lfloor (3g+2)/2 \rfloor, \lfloor (3g+5)/2 \rfloor \rangle$.*

In this case $\ell_g = \lfloor (3g-1)/2 \rfloor$.

Proof. If H is sparse, by induction $\ell_i = \lfloor (3i-1)/2 \rfloor$ for $i = 1, \ldots, g$. Then $H = 3\mathbf{N}_0 \cup \{n \in \mathbf{N} : n \geq \lfloor (3g+2)/2 \rfloor\}$.

The following results show a strong restriction on the arithmetic of sparse semigroups. They generalize the statement (2) implies (3) in Proposition 2.

Lemma 1. *Let H be a sparse semigroup of genus g with* $\mathrm{Gaps}(H) = \{\ell_1, \ldots, \ell_g\}$. *If $\ell_{g-i} = \ell_{g-i+1} - 1$, then $2\ell_{g-i} < 3(g-i)$.*

Proof. The set $\{1, \ldots, \ell_{g-i}\}$ contains $g - i$ gaps and $t = \ell_{g-i} - (g-i)$ non-gaps. Write $\{1, \ldots, \ell_{g-i}\} \cap H = \{\rho_2, \ldots, \rho_{t+1}\}$. All elements in the set $\{\ell_{g-i} - \rho_2, \cdots, \ell_{g-i} - \rho_{t+1}\} \cup \{\ell_{g-i+1} - \rho_2, \cdots, \ell_{g-i+1} - \rho_{t+1}\}$ are distinct gaps. If otherwise $\ell_{g-i} - \rho_r = \ell_{g-i+1} - \rho_s = \ell_{g-i} + 1 - \rho_s$, then $\rho_r + 1 = \rho_s$ contradicting the fact of H being sparse. Thus $2(\ell_{g-i} - (g-i)) + 1 \leq g - i$ and hence $2\ell_{g-i} \leq 3(g-i) - 1$.

Let $m = m(H) = \rho_2 - 1$ be as above and let $K = K(H) = 2g - \ell_g$.

Theorem 1. *Let H be a sparse semigroup of genus g with* $\mathrm{Gaps}(H) = \{\ell_1, \ldots, \ell_g\}$. *Then*

1. *$\ell_i \geq 2i - K$ for all $i = 1, \ldots, g$. In particular $m \leq K$.*
2. *If $\ell_i = 2i - K$ then $\ell_j = 2j - K$ for $j = i, \ldots, g$.*
3. *If $g \geq 2K - 1$ then $\ell_i = 2i - K$ for all $i = 2K - 2, \ldots, g$.*

Proof. (1) By the sparse property, $\ell_g - \ell_i \leq 2(g - i)$ so that $\ell_i \geq 2i - K$. Then $2(j + 1) - K \leq \ell_{j+1} \leq \ell_j + 2$ and the result follows. (2) is clear. (3) Since $\ell_g = 2g - K$ by definition of K, it is enough to prove that $\ell_{i+1} = \ell_i + 2$ for $i = 2k - 2, \ldots, g - 1$. We use induction on $j = g - i$ and thus we have to prove the statement $\ell_{g-j} = \ell_{g-j+1} - 2$ for $j = 1, \ldots, g - 2K + 2$. If $\ell_{g-1} \neq \ell_g - 2$, the sparse property implies $\ell_{g-1} = \ell_g - 1$. Then, according to Lemma 1, $2(\ell_g - 1) = 2\ell_{g-1} < 3(g - 1)$ and so $g < 2K - 1$, a contradiction. Suppose that the assertion is true for all $t = 1, \ldots, j$ $(1 \leq j \leq g - 2K + 1)$ and let us prove it for $j + 1$. If $\ell_{g-j-1} = \ell_{g-j} - 1$, then from the Lemma it follows that $2(\ell_{g-j} - 1)) = 2\ell_{g-j-1} < 3(g - i - 1)$. By induction hypothesis we have $\ell_{g-j} = \ell_g - 2j$, hence $j > g - 2K + 1$, which is also a contradiction.

Example 3. Let H be a semigroup. According to the Theorem, $m(H) \leq K(H)$. When equality holds, then H is sparse iff $\ell_i = 2i - K$ for $i = K, \ldots, g$. Note that in this case H is also Arf. If $K = 2k$, then H is sparse iff H has exactly $(g - k)$ even gaps. Here observe that $2(2k + 1) \geq 2g - 2k + 2$ so that $g \leq 3k$. This example will be generalized in Theorem 2.

3 The Largest Gap: Even Case

As we have seen, the structure of sparse semigroups $H \neq \mathbf{N}_0$ is strongly influenced by its largest gap ℓ_g, and in particular by the parity of ℓ_g. Let us study now the case in which ℓ_g is even and thus $m = m(H) = \rho_2 - 1 \geq 2$.

Theorem 2. *Let H be a sparse semigroup of genus g with $\ell_g = 2g - K = 2g - 2k$. If $g \geq 4k - 1$, then $g \leq 6k - m(H) - 1$. In particular, the family of semigroups $\mathcal{H}_k = \bigcup_{g \geq 0} \mathcal{H}_{g,k}$, where $\mathcal{H}_{g,k}$ is as in (2), is finite.*

Proof. From Theorem 1(3), the gaps $6k - 2 = \ell_{4k-1} < \ldots < 2g - 2k = \ell_g$, are all of them even numbers. Thus the gaps $3k - 1 = \ell_{I-(g-4k+1)} < \cdots < g - k = \ell_I$ are consecutive integers and $I \leq 4k - 1$. By the sparse property, $\ell_{4k-1} - \ell_I \leq 2(4k - 1 - I)$, hence $2I \leq g + k$ $(*)$. If $m \geq g - k$, then $g \leq 3k$ by Theorem 1(1), so that $k = 1$ and $g = 3$. If $m < g - k$, then $m < 3k - 1$. By applying the sparse property once again, it follows that $3k - 1 - m \leq 2(I - g + 4k - 1 - m)$, i.e., $2g \leq 2I + 5k - m - 1$. Finally $(*)$ implies $g \leq 6k - m - 1$.

Example 4. In [2] and [3], Barucci et al. and Bras-Amorós proved that there exist only two Arf semigroups with $\ell_g = 2g - 2$, namely $H = \{0, 3, 4, \ldots\}$ and $H = \{0, 3, 5, 6, 7, \ldots\}$. By the above Theorem, the following statements are equivalent:

1. H is Arf with $\ell_g = 2g - 2$;
2. H is sparse with $\ell_g = 2g - 2$;
3. $g = 2$ and $H = \{0, 3, 4, \ldots\}$, or $g = 3$ and $H = \{0, 3, 5, 6, 7, \ldots\} = 3\mathbf{N}_0 \cup \{h \in \mathbf{N} : h \geq 5\}$.

Example 5. By using the above results we can compute all sparse semigroups with $\ell_g = 2g - 4$. First note that all these semigroups have genus $g \geq 4$. Also $g \leq 7$. If otherwise $g \geq 4k = 8$, Theorem 2 implies $g \leq 11 - m$ and hence $m = 2$ or $m = 3$. In both cases $12 \in H$. Since $\ell_7 = 10$, we have $12 \geq 2g$ which is a contradiction.

Thus $4 \leq g \leq 7$. Let $g > 4$. By Theorem 1(1) we have $2 \leq m \leq 4$. If $m = 2$ then Proposition 3 allows us to compute H. Let $m = 3$. We shall show that $g = 5$ and $H = \{0, 4, 7, 8, 9, 10, \ldots\} = 4\mathbf{N}_0 \cup \{h \in \mathbf{N} : h \geq 7\}$. In fact, here $\rho_2 = 4$, $\ell_4 = 5$. If $6 \in H$, then $\{4, 6, 8, 10, \ldots\} \subseteq H$, which implies $2g - 4 \leq 2$. Thus $\ell_5 = 6$. If $g > 5$, $8 \in \mathrm{Gaps}(H)$ by the sparse property. Finally the case $m = 4$ is computed via Example 3.

We subsume all these computations as follows. Let H be a semigroup with $\ell_g = 2g - 4$. The following statements are equivalent:

1. H is Arf;
2. H is sparse;
3. $g = 4$ and $\mathrm{Gaps}(H) = \{1, 2, 3, 4\}$, or $g = 5$ and $H = \langle 3, 8, 10\rangle$, or $g = 6$ and $H = \langle 3, 10, 11\rangle$, or $g = 7$ and $H = \langle 3, 11, 13\rangle$, or $g = 5$ and $H = \langle 4, 7, 9, 10\rangle$, or $m(H) = 4$ and hence H has exactly $(g - 2)$ even gaps for $g = 5, 6, 7$.

Remark 1. We can also study the cardinality of the family (2). Consider the semigroups H of genus g with $\ell_g(H) = 2g - 2k$ and $m(H) = 2k$. We have $2k \leq g \leq 3k$ and we obtain exactly $k + 1$ of such semigroups (cf. Example 3).

4 The Largest Gap: Odd Case

Let us study now the case ℓ_g odd. To do that we shall need some results concerning γ-hyperelliptic semigroups, which are those semigroups having exactly γ even gaps (cf. [13], [14]).

Proposition 4. *Let H be a semigroup. If there exists an integer $\gamma \geq 0$ such that $\rho_{2\gamma+2} = 6\gamma + 2$ and $g \geq 6\gamma + 4$, then H can be written in the form*

$$H = 2\bar{H} \cup \{u_\gamma, \ldots, u_1\} \cup \{n \in \mathbf{N} : n \geq 2g\},$$

where $\bar{H}$ is a semigroup of genus γ and $u_\gamma < \cdots < u_1 \leq 2g - 1$ are precisely the γ odd nongaps of H up to $2g$.

Proof. See [13], Theorem 2.1.

Observe that H in the above Proposition has exactly γ even nongaps and $\bar{H} = \{h/2 : h \in H, h \equiv 0 \pmod 2\}$. The main result of this section is the following.

Theorem 3. *Let H be a sparse semigroup with $\ell_g = 2g-(2\gamma+1)$. If $g \geq 6\gamma+4$ then $H = 2\bar{H} \cup \{u_\gamma, \ldots, u_1\} \cup \{n \in \mathbf{N} : n \geq 2g\}$, where $\bar{H}$ is a semigroup of genus γ and $u_i = 2g-2i+1$ for $i = 1, \ldots, \gamma$.*

Proof. By Theorem 1 we have $\ell_{2K-1} = 3K-2$ and $\ell_{2K} = 3K$ where $K = 2\gamma+1$. Thus $\rho_{K+1} = 3K-1$, hence $\rho_{2\gamma+2} = 6\gamma+2$ and the result follows.

A semigroup as in the Theorem will be called *ordinary γ-hyperelliptic*.

Remark 2. The lower bound on the genus in Theorem 3 is necessary as the following example shows. Let $\gamma \geq 1$ be an integer and consider the Arf semigroup $H = \langle 3, 6\gamma+2, 6\gamma+4 \rangle$. H is 2γ-hyperelliptic of genus $g = 4\gamma+1$ but $\ell_g = 2g-(2\gamma+1)$ (cf. Proposition 3).

Remark 3. There exist γ-hyperelliptic sparse semigroups of genus $g \geq 2\gamma+1$. Let H be the semigroup defined in Example 3. Here $\ell_i = i$ for $i = 1, \ldots, K$, $\ell_i = 2i-K$ for $i = K, \ldots, g$, and $K = 2\gamma+1$.

Corollary 2. *Let $\gamma \geq 0$ be an integer. Let H be a sparse semigroup of genus $g \geq 6\gamma+4$ with $\ell_g = 2g-(2\gamma+1)$. Let $\bar{H} = \{h/2 : h \in H, h \equiv 0 \pmod 2\}$. Then H is Arf if and only if so is $\bar{H}$.*

Proof. We can write $H = 2\bar{H} \cup \{h \in \mathbf{N} : h \geq 2g-2\gamma+1\}$. Since $R = 2g-2\gamma+1 \geq 2\bar{c}+1$ (note $2\gamma \geq \bar{c}$), the result follows from Example 2.

Example 6. If $g \geq 4$ and $\ell_g = 2g-1$, then $\gamma = 0$ and we find a new proof of Proposition 2.

Remember that there exists just one semigroup of genus 1, namely $\{0, 2, 3, \ldots\}$ and two semigroups of genus 2, $\{0, 2, 4, 5, \ldots\}$ and $\{0, 3, 4, 5, \ldots\}$. All of them are Arf semigroups.

Example 7. Let H be a semigroup of genus $g \geq 10$ with $\ell_g = 2g-3$. The following statements are equivalent:

1. H is Arf;
2. H is sparse;
3. H is ordinary 1-hyperelliptic, that is $H = 2\{0, 2, 3, \ldots\} \cup \{n \in \mathbf{N}_0 : n \geq 2g-1\}$.

Example 8. Let H be a semigroup of genus $g \geq 16$ with $\ell_g = 2g-5$. The following statements are equivalent:

1. H is Arf;
2. H is sparse;
3. H is ordinary 2-hyperelliptic, that is $H = 2\bar{H} \cup \{2g-3, 2g-1\}$, where $\bar{H}$ is a semigroup of genus 2.

Example 9. For every $\gamma \geq 3$, there exist semigroups $\bar{H}$ of genus γ which are not Arf. For example, $H := \mathbf{N}_0 \setminus \{1, 2, \ldots, \gamma-1, \gamma+2\}$. Thus for every $\gamma \geq 3$ there exist ordinary γ-hyperelliptic semigroups having largest gap ℓ_g odd, which are not Arf property (cf. Corollary 2).

5 On the Order of Semigroups

In this section we are interested in the *order bound* of sparse semigroups. Let us remember that for a semigroup $H = \{0 = \rho_1 < \rho_2 < \cdots\}$, we can consider the sets

$$A(\ell) = A[\rho_\ell] := \{(s,t) \in \mathbf{N}^2 : \ \rho_s + \rho_t = \rho_\ell\}$$

$\ell \in \mathbf{N}$, and their cardinals

$$\nu_\ell := |A(\ell+1)|$$

(or equivalently, $\nu_\ell = |\{\rho_{\ell+1} - \rho_1, \ldots, \rho_{\ell+1} - \rho_{\ell+1}\} \cap H|$). For $\ell \in \mathbf{N}_0$ we define the ℓ-th *order bound* of H as $d_\ell(H) := \min\{\nu_m : m \geq \ell\}$. The order bound was introduced in the context of the Theory of Error-Correcting codes. In fact, $d_\ell(H)$ gives a lower bound on the minimum distance of Algebraic Geometry codes related to H (see [6] for details). The order bound is often difficult to compute. For this reason, we define the *order number* of H as

$$o(H) := \min\{\ell \in \mathbf{N} : \nu_j \leq \nu_{j+1} \text{ for all } j \geq \ell\}\,.$$

It is clear that $d_\ell(H) = \nu_\ell$ for all $\ell \geq o(H)$.

Let us recall the concept of *acute* and *near-acute* semigroups (cf. [3], [8]). Write

$$H = \{0\} \cup [c_m, d_m] \cup \cdots \cup [c_1, d_1] \cup [c_0, \infty)\,, \tag{4}$$

where c_i, d_i are non-negative integers such that $d_{i+1} + 2 \leq c_i \leq d_i$ for $i = 1, \ldots, m$, $d_1 + 2 \leq c$, and $c_{m+1} = d_{m+1} = 0$, with c_0 equals the conductor c of H. Consider the following conditions:

(I) $c_0 - d_1 \leq c_1 - d_2$;
(II) $c_0 - d_1 \leq d_1 - d_2$;
(III) $2d_1 - c_0 + 1 \notin H$.

We say that H is acute if it satisfies (I); H is near-acute if it satisfies (II) or (III). Notice that (I) implies (II). Define the function ι on H by $\iota(\rho_i) = i$. In [8, Thm. 3.11] it is shown that $o(H) = \min\{\iota(2d_1+1), \iota(c_0+c_1-1)\}$ provided that H is near-acute (*). See also [10,11] for more information.

Proposition 5. *Let H be a sparse semigroup of genus g with $\ell_g = 2g - K$. If $2\ell_g \geq 3g$ then H is acute and $o(H) = 2\ell_g - g = 3g - 2K$.*

Proof. Notation as in equation (4). From Theorem 1, $c_0 = \ell_g + 1$, $c_1 = d_1 = \ell_g - 1$ and $c_2 = d_2 = \ell_g - 3$. We have $2d_1 + 1 = c_0 + c_1 - 1 = 2\ell_g - 1$ and thus H is acute. Now $2\ell_g - 1 = \rho_{2\ell_g - g}$ and the proof follows from the above formula $(*)$ for $o(H)$.

Example 10. For $g \geq 7$, let $H = \{0, g-2, g, g+3, g+4 \ldots\}$ be the sparse semigroup of Example 1 (thus $\ell_g = g + 2 = 2g - (g-2)$). It is easy to check that H is not near-acute. We claim that $o(H) = g - 1$ (in particular, $o(H)$ is not given by the formula $(*)$ above).

Since $r = 2c_0 - g - 1 = g + 5$, from [8, Thm. 2.6], the sequence $(\nu_\ell)_{\ell \geq g+5}$ is strictly increasing. Thus we have to consider the numbers ν_ℓ for $\ell \leq g+5$. Here $\rho_1 = 0$, $\rho_2 = g-2$, $\rho_3 = g$ and $\rho_\ell = g + \ell - 1$ for $\ell \geq 4$. Let $k \in \{1, \ldots, 6\}$ and $i \in [k+3, g+k-1]$. We have $\rho_{g+k} - \rho_i = g + k - i \in [1, g-3]$ and so

$$\nu_{g+k-1} = |\{\rho_{g+k} - \rho_2, \ldots, \rho_{g+k} - \rho_{k+2}\} \cap H| + 2\,.$$

If $k = 1$ then $\nu_g = |\{g+2, g\} \cap H| + 2 = 3$, otherwise

$$\begin{aligned} \rho_{g+k} - \rho_{k+2} &= g-2\,, && \text{if } k \geq 2 \\ \rho_{g+k} - \rho_{k+1} &= g-1 \notin H\,, && \text{if } k \geq 3 \\ \rho_{g+k} - \rho_k &= g\,, && \text{if } k \geq 4\,. \end{aligned}$$

In addition, if $k \geq 2$ we obtain $\rho_{g+k} - \rho_2 = g + k + 1 \in H$ and thus

$$\nu_{g+k-1} = |\{\rho_{g+k} - \rho_3, \ldots, \rho_{g+k} - \rho_{k+2}\} \cap H| + 3\,.$$

Therefore

$$\begin{aligned} \nu_{g+1} &= |\{g+1, g-2\} \cap H| + 3 = 4\,, \\ \nu_{g+2} &= |\{g+2, g-1, g-2\} \cap H| + 3 = 4\,, \\ \nu_{g+3} &= |\{g+3, g, g-1, g-2\} \cap H| + 3 = 6\,, \\ \nu_{g+4} &= |\{g+4, g+1, g, g-1, g-2\} \cap H| + 3 = 6\,, \\ \nu_{g+5} &= |\{g+5, g+2, g+1, g, g-1, g-2\} \cap H| + 3 = 6\,. \end{aligned}$$

Now, let $s \in \{0, 1\}$ and $j \in [4, g+s-1]$ then $\rho_{g-s} - \rho_j = g - s - j \in [1, g-4]$ and so

$$\nu_{g-s-1} = |\{\rho_{g-s} - \rho_2, \rho_{g-s} - \rho_3\} \cap H| + 2\,.$$

Then

$$\begin{aligned} \nu_{g-1} &= |\{g+1, g-1\} \cap H| + 2 = 2\,, \\ \nu_{g-2} &= |\{g, g-2\} \cap H| + 2 = 4\,. \end{aligned}$$

Thus $o(H) = g - 1$.

Acknowledgment. The authors wish to thank the reviewers for their detailed comments and suggestions.

References

1. Arf, C.: Une interpretation algébrique de la suite des ordres de multiplicité dúne branche algébrique. Proc. London Math. Soc. 50, 256–287 (1949)
2. Barucci, V., Dobbs, D.E., Fontana, M.: Maximality properties in numerical semigroups and applications to one-dimensional analytically irreducible local domains. Mem. Amer. Math. Soc. 125 (1997)

3. Bras-Amorós, M.: Acute semigroups, the order bound on the minimum distance, and the Feng-Rao improvement. IEEE Trans. Inform. Theory 50(6), 1282–1289 (2004)
4. Bras-Amorós, M., García, P.A.: Patterns on numerical semigroups. Linear Algebra and its Applications 414, 652–669 (2006)
5. Campillo, A., Farrán, J.I., Munuera, C.: On the parameters of algebraic-geometry codes related to Arf semigroups. IEEE Trans. Inform. Theory 46(7), 2634–2638 (2000)
6. Høholdt, T., van Lint, J.H., Pellikaan, R.: Algebraic Geometry codes. In: Pless, V., Huffman, C. (eds.) Handbook of Coding Theory, pp. 871–961. Elsevier, Amsterdam (1998)
7. Lipman, J.: Stable ideal and Arf semigroups. Amer. J. Math. 97, 791–813 (1975)
8. Munuera, C., Torres, F.: A note on the order bound on the minimum distance of AG codes and acute semigroups. Advances in Mathematics of Communications 2(2), 175–181 (2008)
9. Oliveira, G.: Weierstrass semigroups and the canonical ideal of non-trigonal curves. Manuscripta Math. 71, 431–450 (1991)
10. Oneto, A., Tamone, G.: On numerical Semigroups and the Order Bound. J. Pure Appl. Algebra 212, 2271–2283 (2008)
11. Oneto, A., Tamone, G.: On the order bound of one-point algebraic geometry codes. J. Pure Appl. Algebra (to appear)
12. Rosales, J.C., García-Sánchez, P.A., García-García, J.I., Branco, M.B.: Arf Numerical Semigroups. Journal of Algebra 276, 3–12 (2004)
13. Torres, F.: On γ-hyperelliptic numerical semigroups. Semigroup Forum 55, 364–379 (1997)
14. Torres, F.: Weierstrass points and double coverings of curves with applications: Symmetric numerical semigroups which cannot be realized as Weierstrass semigroups. Manuscripta Math. 83, 39–58 (1994)

From the Euclidean Algorithm for Solving a Key Equation for Dual Reed–Solomon Codes to the Berlekamp–Massey Algorithm*

Maria Bras-Amorós[1,**] and Michael E. O'Sullivan[2]

[1] Departament d'Enginyeria Informàtica i Matemàtiques
Universitat Rovira i Virgili
maria.bras@urv.cat
[2] Department of Mathematics and Statistics
San Diego State University
mosulliv@sciences.sdsu.edu

Abstract. The two primary decoding algorithms for Reed-Solomon codes are the Berlekamp-Massey algorithm and the Sugiyama et al. adaptation of the Euclidean algorithm, both designed to solve a key equation. This article presents a new version of the key equation and a way to use the Euclidean algorithm to solve it. A straightforward reorganization of the algorithm yields the Berlekamp-Massey algorithm.

1 Introduction

For correcting primal Reed-Solomon codes a useful tool are the so-called locator and evaluator polynomials. Once we know them, the error positions are determined by the inverses of the roots of the locator polynomial and the error values can be computed by a formula due to Forney [4] which uses the evaluator and the derivative of the locator evaluated at the inverses of the error positions.

Berlekamp presented in [1] a key equation determining the decoding polynomials for primal Reed-Solomon codes. Sugiyama et al. introduced in [9] their celebrated algoithm for solving this key equation based on the Euclidean algorithm for computing the greatest common divisor of two polynomials and the coefficients of the Bézout's identity. Another celebrated algorithm for solving the key equation is the algorithm by Berlekamp and Massey [1, 6] which is widely accepted to have better performance than the Sugiyama et al. algorithm, although its formulation is more difficult to understand. The connections between both algorithms were analyzed in [3, 5].

In this work we take the perspective of dual Reed Solomon codes. In Section 3 we present a key equation for dual Reed-Solomon codes and in Section 4 we

* An extended version is in preparation for being submitted for publication.

** This work was partly supported by the Catalan Government through a grant 2005 SGR 00446 and by the Spanish Ministry of Education through projects TSI2007-65406-C03-01 E-AEGIS and CONSOLIDER CSD2007-00004 ARES.

M. Bras-Amorós and T. Høholdt (Eds.): AAECC 2009, LNCS 5527, pp. 32–42, 2009.

introduce an Euclidean-based algorithm for solving it, following Sugiyama et al.'s idea. While the key equation for primal Reed-Solomon codes states that a linear combination of the decoding polynomials is a multiple of a certain power of x, in the key equation presented here the linear combination has bounded degree. Then, while in Sugiyama et al.'s algorithm the locator and evaluator polynomials play the role of one of the Bézout's coefficients and the residue respectively, in the Euclidean algorithm presented here the locator and evaluator polynomials play the role of the two coefficients of the Bézout's identity.

The decoding polynomials are now slightly different and in a sense more natural, since the error positions are given by the roots themselves of the locator polynomial rather than their inverses and that the error values are obtained by evaluating the evaluator polynomial and the derivative of the locator polynomial at the error positions rather than evaluating them at the inverses of the error positions. In addition the equivalent of the Forney formula does not have the minus sign.

In Section 5 we prove that the new Euclidean-based algorithm is exactly the Berlekamp-Massey algorithm. The connections between the Euclidean and the Berlekamp-Massey algorithms seem to be much more transparent for the dual Reed-Solomon codes than for the primal codes [3, 5]. In Section 6 we move back to primal Reed Solomon codes and see how all the developments of the previous sections can be applied to primal codes with minor modifications.

2 Settings and Notations

In this section we establish the notions and notations that we will use in the present work. A general reference is [8]. Let $\mathbb{F}$ be a finite field of size $q = p^m$ and let α be a primitive element in $\mathbb{F}$. Let $n = q - 1$. We identify the vector $u = (u_0, \ldots, u_{n-1})$ with the polynomial $u(x) = u_0 + \cdots + u_{n-1}x^{n-1}$ and denote $u(a)$ the evaluation of $u(x)$ at a.

Classically the (primal) Reed-Solomon code $C^*(k)$ of dimension k is the cyclic code generated by the polynomial $(x-\alpha)(x-\alpha^2)\cdots(x-\alpha^{n-k})$, and has generator matrix $G^*(k) = (\alpha^{ij})_j^i$, for $i = 0, \ldots, k-1$, and $j = 0, \ldots, n-1$, while the dual Reed-Solomon code $C(k)$ of dimension k is generated by the polynomial $(x - \alpha^{-(k+1)})\cdots(x-\alpha^{-(n-1)})(x-1)$ and has generator matrix $G^*(k) = (\alpha^{(i+1)j})_j^i$, for $i = 0, \ldots, k-1$, and $j = 0, \ldots, n-1$.

Both codes have minimum distance $d = n - k + 1$. Furthermore, $C(k)^\perp = C^*(n-k)$. There is a natural bijection from $\mathbb{F}^n$ to itself, $c \mapsto c^*$, that takes $C(k)$ to $C^*(k)$. Indeed, let i be a vector of dimension k and let $c = (c_0, c_1, \ldots, c_{n-1}) = iG(k) \in C(k)$, $c^* = (c_0^*, c_1^*, \ldots, c_{n-1}^*) = iG^*(k) \in C^*(k)$. Then,

$$c = (c_0^*, \alpha c_1^*, \alpha^2 c_2^*, \ldots, \alpha^{n-1} c_{n-1}^*). \tag{1}$$

In particular, $c(\alpha^i) = c_0^* + \alpha c_1^* \alpha^i + \alpha^2 c_2^* \alpha^{2i} + \cdots + \alpha^{n-1} c_{n-1}^* \alpha^{(n-1)i} = c^*(\alpha^{i+1})$.

3 Key Equations for Primal and Dual Reed-Solomon Codes

3.1 Polynomials for Correction of RS Codes

Suppose that a word $c^* \in C^*(k)$ is transmitted and that an error e^* occurred with $t \leq \frac{d-1}{2}$ non-zero positions, so that $u^* = c^* + e^*$ is received. Define the *error locator polynomial* Λ^* and the *error evaluator polynomial* Ω^* as

$$\Lambda^* = \prod_{i:e_i^* \neq 0} (1 - \alpha^i x), \qquad \Omega^* = \sum_{i:e_i^* \neq 0} e_i^* \alpha^i \prod_{j:e_j^* \neq 0, j \neq i} (1 - \alpha^i x),$$

and the *syndrome polynomial*

$$S^* = e^*(\alpha) + e^*(\alpha^2)x + e^*(\alpha^3)x^2 + \cdots + e^*(\alpha^n)x^{n-1}.$$

Notice that from the received word we only know $e^*(\alpha) = u^*(\alpha), \ldots, e^*(\alpha^{n-k}) = u^*(\alpha^{n-k})$. This is why we use the *truncated syndrome polynomial*

$$\bar{S}^* = e^*(\alpha) + e^*(\alpha^2)x + \cdots + e^*(\alpha^{n-k})x^{n-k-1}.$$

If Λ^* and Ω^* are known, the error positions can be identified as the indices i such that

$$\Lambda^*(\alpha^{-i}) = 0$$

and the error values can be computed by the Forney formula [4]

$$e_i^* = -\frac{\Omega^*(\alpha^{-i})}{\Lambda'^*(\alpha^{-i})}.$$

3.2 Polynomials for Correction of Dual RS Codes

Suppose that a word $c \in C(k)$ is transmitted and that an error e occurred with $t \leq \frac{d-1}{2}$ non-zero positions, so that $u = c + e$ is received. In this case define the *error locator polynomial* Λ and the *error evaluator polynomial* Ω as

$$\Lambda = \prod_{i:e_i \neq 0} (x - \alpha^i), \qquad \Omega = \sum_{i:e_i \neq 0} e_i \prod_{j:e_j \neq 0, j \neq i} (x - \alpha^i), \tag{2}$$

and the *syndrome polynomial*

$$S = e(\alpha^{n-1}) + e(\alpha^{n-2})x + \cdots + e(\alpha)x^{n-2} + e(1)x^{n-1}. \tag{3}$$

Notice that now, from the received word we only know $e(1) = u(1), \ldots, e(\alpha^{n-k-1}) = u(\alpha^{n-k-1})$. This is why we use the *truncated syndrome polynomial*

$$\bar{S} = e(\alpha^{n-k-1})x^k + e(\alpha^{n-k-2})x^{k+1} + \cdots + e(1)x^{n-1}. \tag{4}$$

In this case, if Λ and Ω are known, the error positions can be identified as the indices i such that

$$\Lambda(\alpha^i) = 0 \tag{5}$$

and the error values can be computed by the formula

$$e_i = \frac{\Omega(\alpha^i)}{\Lambda'(\alpha^i)}. \tag{6}$$

It is easy to check that when the error vectors e and e^* satisfy the relationship (1), then the polynomials Λ, Ω, S, S associated to the error e are related to the polynomials Λ^*, Ω^*, S^*, $\bar{S}^*$ associated to the error e^* as follows:

$$\begin{array}{ll} \Lambda = x^t \Lambda^*(1/x), & \Omega = x^{t-1}\Omega^*(1/x), \\ S = x^{n-1}S^*(1/x), & \bar{S} = x^{n-1}\bar{S}^*(1/x). \end{array}$$

3.3 Key Equations

A straightforward computation shows that

$$\Lambda S = (x^n - 1)\Omega \tag{7}$$

or, equivalently, $\Lambda^* S^* = (1 - x^n)\Omega^*$. See [7] for a proof. In particular, the polynomial $\Lambda^*\bar{S}^* + (x^n - 1)\Omega^* = \Lambda^*(\bar{S}^* - S^*)$ has only terms of order $n - k$ or larger and this implies

$$\boxed{\Lambda^*\bar{S}^* = \Omega^* \mod x^{n-k}}$$

This is the key equation introduced by Berlekamp [1]. Massey [6] gave an algorithm for solving linear feedback shift registers and its application to decoding BCH and thus RS codes. Sugiyama et al. [9] recognized that the Euclidean algorithm for finding the greatest common divisor of two polynomials and for solving Bézout's identity, could also be adapted to solve this kind of equation for Λ^* and Ω^*, starting with $r_{-2} = x^{n-k}$ and $r_{-1} = \bar{S}^*$. Their method is often referred to as the Euclidean decoding algorithm for Reed Solomon codes.

On the other hand, the polynomial $\Lambda\bar{S} - (x^n - 1)\Omega = \Lambda(\bar{S} - S)$ has degree at most $\lfloor \frac{d-1}{2} \rfloor + k - 1 = \lfloor \frac{d-1}{2} \rfloor + n - d = n - \lceil \frac{d+1}{2} \rceil$. That is

$$\boxed{\deg(\Lambda\bar{S} - (x^n - 1)\Omega) \leq n - \lceil \tfrac{d+1}{2} \rceil}$$

The aim of the present work is to deal with this alternative key equation on Λ and Ω and solving it by the Euclidean algorithm starting with $r_{-2} = x^n - 1$ and $r_{-1} = \bar{S}$.

4 Euclidean Algorithm for Dual Reed-Solomon Codes

The next theorem characterizes the decoding polynomials by means of the alternative key equation, the polynomial degrees, and their coprimality. It is the analogue of [2, Theorem 4.8] for the standard key equation, and the proof is similar.

Theorem 1. *Suppose that at most* $\lfloor \frac{d-1}{2} \rfloor$ *errors occurred. Then* Λ *and* Ω *are the unique polynomials* f, φ *satisfying the following properties.*

1. $deg(f\bar{S} - (x^n - 1)\varphi) \leq n - \lceil \frac{d+1}{2} \rceil$
2. $deg(\varphi) < deg(f) \leq \lceil \frac{d-1}{2} \rceil$
3. f, φ *are coprime.*
4. f *is monic*

Consider the following algorithm:

Initialize:
$r_{-2} = x^n - 1$, $f_{-2} = 0$, $\varphi_{-2} = -1$,
$r_{-1} = \bar{S}$, $f_{-1} = 1$, $\varphi_{-1} = 0$,
$i = -1$.
while deg$(r_i) > n - \lceil \frac{d+1}{2} \rceil$**:**
$q_i =$ **Quotient**(r_{i-2}, r_{i-1})
$r_i =$ **Remainder**(r_{i-2}, r_{i-1})
$\varphi_i = \varphi_{i-2} - q_i \varphi_{i-1}$
$f_i = f_{i-2} - q_i f_{i-1}$
end while
Return f_i/**LeadingCoefficient**$(f_i), \varphi_i$/**LeadingCoefficient**(f_i)

The polynomial φ has been initialized to negative 1 because we want that f_i, φ_i satisfy at each step $f_i\bar{S} - \varphi_i(x^n - 1) = r_i$.

The next theorem verifies the algorithm. Its proof has also been ommitted and will appear in the final form of this article.

Theorem 2. *If* $t \leq \frac{d-1}{2}$ *then the algorithm outputs* Λ *and* Ω.

5 From the Euclidean to the Berlekamp-Massey Algorithm

In this section we will perform a series of minor modifications to the previous algorithm that will lead to the Berlekamp-Massey algorithm. This will show that they are essentially the same.

First of all, notice that the updating step in the previous algorithm can be expressed in matrix form as

$$\begin{pmatrix} r_i & f_i & \varphi_i \\ r_{i-1} & f_{i-1} & \varphi_{i-1} \end{pmatrix} = \begin{pmatrix} -q_i & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} r_{i-1} & f_{i-1} & \varphi_{i-1} \\ r_{i-2} & f_{i-2} & \varphi_{i-2} \end{pmatrix}.$$

Furthermore, if $q_i = q_i^{(0)} + q_i^{(1)}x + \cdots + q_i^{(l)}x^l$ then

$$\begin{pmatrix} -q_i & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & -q_i^{(0)} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -q_i^{(1)}x \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -q_i^{(2)}x^2 \\ 0 & 1 \end{pmatrix} \cdots \begin{pmatrix} 1 & -q_i^{(l)}x^l \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

If we split all matrix multiplications we will need to add intermediate variables. Thus, it does not make sense anymore to consider matrices

$$\begin{pmatrix} r_i & f_i & \varphi_i \\ r_{i-1} & f_{i-1} & \varphi_{i-1} \end{pmatrix}.$$

We will use auxiliary polynomials $R_i, F_i, \Psi_i, \tilde{R}_i, \tilde{F}_i, \tilde{\Psi}_i$ and matrices

$$\begin{pmatrix} R_i & F_i & \Psi_i \\ \tilde{R}_i & \tilde{F}_i & \tilde{\Psi}_i \end{pmatrix}.$$

These matrices will satisfy at given stages of the algorithm (exactly when $\deg(R_i) < \deg(\tilde{R}_i)$) that R_i, F_i, Ψ_i are the remainder and the intermediate Bézout coefficients in one of the steps of the original Euclidean algorithm and that $\tilde{R}_i, \tilde{F}_i, \tilde{\Psi}_i$ are respectively the remainder and the intermediate Bézout coefficients previous to R_i, F_i, Ψ_i.

The previous algorithm can be expressed now:

Initialize:

$$\begin{pmatrix} R_{-1} & F_{-1} & \Psi_{-1} \\ \tilde{R}_{-1} & \tilde{F}_{-1} & \tilde{\Psi}_{-1} \end{pmatrix} = \begin{pmatrix} \bar{S} & 1 & 0 \\ x^n & -1 & 0 & -1 \end{pmatrix}$$

while deg$(R_i) > n - \lceil \frac{d+1}{2} \rceil$**:**

$$\begin{pmatrix} R_{i+1} & F_{i+1} & \Psi_{i+1} \\ \tilde{R}_{i+1} & \tilde{F}_{i+1} & \tilde{\Psi}_{i+1} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} R_i & F_i & \Psi_i \\ \tilde{R}_i & \tilde{F}_i & \tilde{\Psi}_i \end{pmatrix}$$

(quotient=0)

while deg$(R_i) \geq$ **deg**$(\tilde{R}_i)$:

$\mu =$ **LeadingTerm**(R_i)/**LeadingTerm**$(\tilde{R}_i)$

(quotient=quotient+μ**)**

$$\begin{pmatrix} R_{i+1} & F_{i+1} & \Psi_{i+1} \\ \tilde{R}_{i+1} & \tilde{F}_{i+1} & \tilde{\Psi}_{i+1} \end{pmatrix} = \begin{pmatrix} 1 & -\mu \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R_i & F_i & \Psi_i \\ \tilde{R}_i & \tilde{F}_i & \tilde{\Psi}_i \end{pmatrix}$$

end while

end while

Return F_i/**LeadingCoefficient**$(F_i), \Psi_i$/**LeadingCoefficient**(F_i)

5.1 Making Remainders Monic

A useful modification is to keep $\tilde{R}_i$ monic in all steps. This makes it easier to compute the μ's. It is enough to force this every time we get $\deg(R_i) < \deg(\tilde{R}_i)$. When $\deg(R_i) \geq \deg(\tilde{R}_i)$, then $\tilde{R}_i$ stays the same and so it remains monic. To this end we divide R_i by its leading coefficient $\tilde{\mu}$, and also $\tilde{F}_i$ and $\tilde{\Psi}_i$ in order to keep the properties. Analogously, we multiply R_i, F_i and Ψ_i by $-\tilde{\mu}$ in order to make the F_i's monic. Another modification is the use of an integer p keeping track of the degree difference between R_i and $\tilde{R}_i$ and that we take leading coefficients of R_i instead of leading terms.

With these modifications, the algorithm is now:

Initialize:

$$\begin{pmatrix} R_{-1} & R_{-1} & \Psi_{-1} \\ \tilde{R}_{-1} & \tilde{F}_{-1} & \tilde{\Psi}_{-1} \end{pmatrix} = \begin{pmatrix} \bar{S} & 1 & 0 \\ x^n - 1 & 0 & -1 \end{pmatrix}$$

while $\mathbf{deg}(R_i) > n - \lceil \frac{d+1}{2} \rceil$**:**

$\mu = \mathbf{LeadingCoefficient}(R_i)$

$p = \mathbf{deg}(R_i) - \mathbf{deg}(\tilde{R}_i)$

if $p < 0$ **then**

$$\begin{pmatrix} R_{i+1} & F_{i+1} & \Psi_{i+1} \\ \tilde{R}_{i+1} & \tilde{F}_{i+1} & \tilde{\Psi}_{i+1} \end{pmatrix} = \begin{pmatrix} 0 & -\mu \\ 1/\mu & 0 \end{pmatrix} \begin{pmatrix} R_i & R_i & \Psi_i \\ \tilde{R}_i & \tilde{F}_i & \tilde{\Psi}_i \end{pmatrix}$$

else

$$\begin{pmatrix} R_{i+1} & F_{i+1} & \Psi_{i+1} \\ \tilde{R}_{i+1} & \tilde{F}_{i+1} & \tilde{\Psi}_{i+1} \end{pmatrix} = \begin{pmatrix} 1 & -\mu x^p \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R_i & F_i & \Psi_i \\ \tilde{R}_i & \tilde{F}_i & \tilde{\Psi}_i \end{pmatrix}$$

end if

end while

Return F_i, Ψ_i

5.2 Joining Steps

It is easy to check that after each step corresponding to $p < 0$ the new p is exactly the previous one with opposite sign and so is μ. With this observation we can join each step corresponding to $p < 0$ with the following one:

Initialize:

$$\begin{pmatrix} R_{-1} & F_{-1} & \Psi_{-1} \\ \tilde{R}_{-1} & \tilde{F}_{-1} & \tilde{\Psi}_{-1} \end{pmatrix} = \begin{pmatrix} \bar{S} & 1 & 0 \\ x^n - 1 & 0 & -1 \end{pmatrix}$$

while $\mathbf{deg}(R_i) > n - \lceil \frac{d+1}{2} \rceil$**:**

$\mu = \mathbf{LeadingCoefficient}(R_i)$

$p = \mathbf{deg}(R_i) - \mathbf{deg}(\tilde{R}_i)$

if $p < 0$ **then**

$$\begin{pmatrix} R_{i+1} & F_{i+1} & \Psi_{i+1} \\ \tilde{R}_{i+1} & \tilde{F}_{i+1} & \tilde{\Psi}_{i+1} \end{pmatrix} = \begin{pmatrix} x^{-p} & -\mu \\ 1/\mu & 0 \end{pmatrix} \begin{pmatrix} R_i & F_i & \Psi_i \\ \tilde{R}_i & \tilde{F}_i & \tilde{\Psi}_i \end{pmatrix}$$

else

$$\begin{pmatrix} R_{i+1} & F_{i+1} & \Psi_{i+1} \\ \tilde{R}_{i+1} & \tilde{F}_{i+1} & \tilde{\Psi}_{i+1} \end{pmatrix} = \begin{pmatrix} 1 & -\mu x^p \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R_i & F_i & \Psi_i \\ \tilde{R}_i & \tilde{F}_i & \tilde{\Psi}_i \end{pmatrix}$$

end if

end while

Return F_i, Ψ_i

5.3 Skipping Remainders

Finally, the only reason to keep the polynomials R_i (and $\tilde{R}_i$) is that we need to compute their leading coefficients (the μ's). We now show that these leading coefficients may be obtained without reference to the polynomials R_i. This allows to compute the F_i, Ψ_i iteratively and dispense with the polynomials R_i.

On one hand, the remainder $R_i = F_i\bar{S} - \Psi_i(x^n - 1) = F_i\bar{S} - x^n\Psi_i + \Psi_i$ has degree at most $n-1$ for all $i \geq -1$. This means that all terms of $x^n\Psi_i$ cancel with terms of $F_i\bar{S}$ and that the leading term of R_i must be either a term of $F_i\bar{S}$ or a term of Ψ_i or a sum of a term of $F_i\bar{S}$ and a term of Ψ_i.

On the other hand, the algorithm only computes LeadingCoefficient(R_i) while $\deg(R_i) \geq n - \lceil \frac{d-1}{2} \rceil$. We want to see that in this case the leading term of R_i has degree strictly larger than that of Ψ_i. Indeed, one can check that for $i \geq -1$, $\deg(\Psi_i) < \deg(F_i)$ and that all F_i's in the algorithm have degree at most $\lceil \frac{d-1}{2} \rceil$. So $\deg(\Psi_i) < \deg(F_i) \leq \lceil \frac{d-1}{2} \rceil \leq d - \lceil \frac{d-1}{2} \rceil \leq n - \lceil \frac{d-1}{2} \rceil \leq \deg(R_i)$.

The previous algorithm can be transformed in a way such that the remainders are not kept but their degrees:

Initialize:
$d_{-1} = \mathbf{deg}(\bar{S})$ $d_{-2} = n$
$\begin{pmatrix} f_{-1} & \varphi_{-1} \\ F_{-1} & \Psi_{-1} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
while $d_i > n - \lceil \frac{d+1}{2} \rceil$**:**
$\mu = \mathbf{Coefficient}(f_i\bar{S}, d_i)$
$p = d_i - d_{i-1}$
if $\mu = 0$ **then**
$d_i = d_i - 1$
else if $p < 0$ **then**
$\begin{pmatrix} f_{i+1} & \varphi_{i+1} \\ F_{i+1} & \Psi_{i+1} \end{pmatrix} = \begin{pmatrix} x^{-p} & -\mu \\ 1/\mu & 0 \end{pmatrix} \begin{pmatrix} f_i & \varphi_i \\ F_i & \Psi_i \end{pmatrix}$
$d_i = d_{i-2} - 1$
else
$\begin{pmatrix} f_{i+1} & \varphi_{i+1} \\ F_{i+1} & \Psi_{i+1} \end{pmatrix} = \begin{pmatrix} 1 & -\mu x^p \\ 0 & 1 \end{pmatrix} \begin{pmatrix} f_i & \varphi_i \\ F_i & \Psi_i \end{pmatrix}$
$d_i = d_i - 1$
end if
end while
Return f_i, φ_i

This last algorithm is the Berlekamp-Massey algorithm that solves the linear recurrence $\sum_{j=0}^{t} \Lambda_j e(\alpha^{i+j-1}) = 0$ for all $i > 0$. This recurrence is derived from $\Lambda\frac{S}{x^n-1}$ being a polynomial (see (7)) and thus having no terms of negative order in its expression as a Laurent series in $1/x$, and from the equality

$$\frac{S}{x^n - 1} = \frac{1}{x}\left(e(1) + \frac{e(\alpha)}{x} + \frac{e(\alpha^2)}{x^2} + \cdots\right).$$

A proof of this last equality can be found in [7].

6 Moving Back to Primal Reed-Solomon Codes

It is well known that the words of a dual Reed-Solomon code are in bijection with those of the primal Reed-Solomon code of the same dimension. The

natural bijection was previously mentioned in (1). Suppose that a code word c^* from the primal Reed-Solomon code $C^*(k)$ is transmitted, and that an error e^* is added to it giving the received word $u^* = c^* + e^*$. Notice that if $c = (c_0^*, \alpha c_1^*, \alpha^2 c_2^*, \ldots, \alpha^{n-1} c_{n-1}^*)$ and $e = (e_0^*, \alpha e_1^*, \alpha^2 e_2^*, \ldots, \alpha^{n-1} e_{n-1}^*)$, then $c \in C(k)$ and $u := c + e = (u_0^*, \alpha u_1^*, \alpha^2 u_2^*, \ldots, \alpha^{n-1} u_{n-1}^*)$. So, e has the same non-zero positions as e^*, and the error values e_i^* added to u_i^* can be computed from the error values e_i added to u_i by $e_i^* = e_i/\alpha^i$.

Table 1. Steps of the classical Euclidean algorithm

$q(x) = 0$
$r(x) = x^9$
$f(x) = 2$
$\varphi(x) = 0$

$q(x) = 0$
$r(x) = \alpha^{14}x^8 + \alpha x^7 + \alpha^{10}x^6 + \alpha^{10}x^5 + \alpha^{17}x^4 + \alpha^5 x^3 + x^2 + \alpha^{11}x + 2$
$f(x) = 0$
$\varphi(x) = 1$

$q(x) = \alpha^{12}x + \alpha^{12}$
$r(x) = \alpha^3 x^7 + \alpha^{22}x^6 + 2x^5 + \alpha^6 x^4 + \alpha^{16}x^3 + \alpha^9 x^2 + \alpha^5 x + \alpha^{12}$
$f(x) = 2$
$\varphi(x) = \alpha^{25}x + \alpha^{25}$

$q(x) = \alpha^{11}x + \alpha^{21}$
$r(x) = \alpha^3 x^6 + x^5 + \alpha^{23}x^4 + \alpha^{19}x^3 + \alpha^{23}x^2 + \alpha^{24}x + \alpha^{17}$
$f(x) = \alpha^{11}x + \alpha^{21}$
$\varphi(x) = \alpha^{23}x^2 + \alpha^3 x + \alpha^4$

$q(x) = x + 2$
$r(x) = \alpha^{10}x^5 + \alpha^{11}x^4 + \alpha^{14}x^3 + \alpha x^2 + \alpha^{23}x + \alpha^3$
$f(x) = \alpha^{24}x^2 + \alpha^9 x + \alpha^2$
$\varphi(x) = \alpha^{10}x^3 + \alpha^{20}x^2 + \alpha^4 x + \alpha^{16}$

$q(x) = \alpha^{19}x + \alpha^{10}$
$r(x) = \alpha^4 x^3 + \alpha^4 x^2 + \alpha^{10}x + \alpha^{20}$
$f(x) = \alpha^4 x^3 + x^2 + \alpha^7 x + 2$
$\varphi(x) = \alpha^{16}x^4 + \alpha^4 x^3 + \alpha^{17}x^2 + \alpha^9 x + \alpha^7$

Now we can perform the previous versions of the algorithm using

$$\begin{aligned}\bar{S} &= e(\alpha^{n-k-1})x^k + e(\alpha^{n-k-2})x^{k+1} + \cdots + e(1)x^{n-1} \\ &= e^*(\alpha^{n-k})x^k + e^*(\alpha^{n-k-1})x^{k+1} + \cdots + e^*(\alpha)x^{n-1},\end{aligned}$$

which is known because it is equal to $u^*(\alpha^{n-k})x^k + u^*(\alpha^{n-k-1})x^{k+1} + \cdots + u^*(\alpha)x^{n-1}$.

Then, once we have the error positions, we can compute the error values as

$$e_i^* = \frac{\Omega(\alpha_i)}{\alpha^i \Lambda'(\alpha^i)}.$$

As an example, consider $\mathbb{F}_{27} = \mathbb{F}_2[x]/(x^3 + 2x + 1)$ and suppose that α is the class of x. Suppose $d = 10$ and $e^* = \alpha^4 x + \alpha^2 x^2 + \alpha^6 x^8 + \alpha^3 x^{24}$.

For the classical Euclidean algorithm, we need the polynomial

Table 2. Steps of the new Euclidean algorithm

$q(x) = 0$
$r(x) = x^{26} + 2$
$f(x) = 0$
$\varphi(x) = 2$

$q(x) = 0$
$r(x) = 2x^{25} + \alpha^{11}x^{24} + x^{23} + \alpha^5 x^{22} + \alpha^{17}x^{21} + \alpha^{10}x^{20} + \alpha^{10}x^{19} + \alpha x^{18} + \alpha^{14}x^{17}$
$f(x) = 1$
$\varphi(x) = 0$

$q(x) = 2x + \alpha^{24}$
$r(x) = \alpha^{14}x^{24} + \alpha^{16}x^{23} + \alpha^{25}x^{22} + \alpha^{17}x^{21} + \alpha^{20}x^{20} + \alpha^6 x^{19} + \alpha^7 x^{18} + \alpha^{25}x^{17} + 2$
$f(x) = x + \alpha^{11}$
$\varphi(x) = 2$

$q(x) = \alpha^{25}x + \alpha^{17}$
$r(x) = \alpha^{16}x^{23} + \alpha^{15}x^{22} + \alpha^7 x^{21} + \alpha^7 x^{20} + \alpha^{11}x^{19} + \alpha^{25}x^{18} + 2x^{17} + \alpha^{25}x + \alpha^{17}$
$f(x) = \alpha^{12}x^2 + \alpha x + \alpha^{25}$
$\varphi(x) = \alpha^{25}x + \alpha^{17}$

$q(x) = \alpha^{24}x + \alpha^6$
$r(x) = \alpha^{24}x^{22} + 2x^{21} + \alpha^{25}x^{20} + \alpha^9 x^{19} + \alpha^{21}x^{18} + \alpha^3 x^{17} + \alpha^{10}x^2 + \alpha^{24}x + \alpha^{11}$
$f(x) = \alpha^{23}x^3 + \alpha^9 x^2 + \alpha^{22}x + \alpha^{15}$
$\varphi(x) = \alpha^{10}x^2 + \alpha^{24}x + \alpha^{11}$

$q(x) = \alpha^{18}x + \alpha^{18}$
$r(x) = \alpha^8 x^{20} + \alpha^2 x^{19} + \alpha^{21}x^{18} + \alpha^{25}x^{17} + \alpha^{15}x^3 + \alpha^5 x^2 + \alpha^{25}x + \alpha^{25}$
$f(x) = \alpha^2 x^4 + \alpha^4 x^3 + \alpha^{12}x^2 + \alpha^{25}x + \alpha^{11}$
$\varphi(x) = \alpha^{15}x^3 + \alpha^5 x^2 + \alpha^{25}x + \alpha^{25}$

$\bar{S}^* = e^*(\alpha) + e^*(\alpha^2)x + \cdots + e^*(\alpha^{n-k})x^{n-k-1} = \alpha^{14}x^8 + \alpha x^7 + \alpha^{10}x^6 + \alpha^{10}x^5 + a^{17}x^4 + \alpha^5 x^3 + x^2 + \alpha^{11}x + 2.$

For the new Euclidean algorithm we will use $\bar{S} = e^*(\alpha^{n-k})x^k + e^*(\alpha^{n-k-1})x^{k+1} + \cdots + e^*(\alpha)x^{n-1} = 2x^{25} + \alpha^{11}x^{24} + x^{23} + \alpha^5 x^{22} + \alpha^{17}x^{21} + \alpha^{10}x^{20} + \alpha^{10}x^{19} + \alpha x^{18} + \alpha^{14}x^17.$

The steps of both algorithms are written in Table 1 and Table 2.

References

1. Berlekamp, E.R.: Algebraic coding theory. McGraw-Hill Book Co., New York (1968)
2. Cox, D.A., Little, J., O'Shea, D. (eds.): Using algebraic geometry, 2nd edn. Graduate Texts in Mathematics, vol. 185. Springer, New York (2005)
3. Dornstetter, J.-L.: On the equivalence between Berlekamp's and Euclid's algorithms. IEEE Trans. Inform. Theory 33(3), 428–431 (1987)
4. David Forney Jr., G.: On decoding BCH codes. IEEE Trans. Information Theory IT-11, 549–557 (1965)
5. Heydtmann, A.E., Jensen, J.M.: On the equivalence of the Berlekamp-Massey and the Euclidean algorithms for decoding. IEEE Trans. Inform. Theory 46(7), 2614–2624 (2000)
6. Massey, J.L.: Shift-register synthesis and BCH decoding. IEEE Trans. Information Theory IT-15, 122–127 (1969)

7. O'Sullivan, M.E., Bras-Amorós, M.: The Key Equation for One-Point Codes, ch. 3, pp. 99–152 (2008)
8. Roth, R.: Introduction to coding theory. Cambridge University Press, Cambridge (2006)
9. Sugiyama, Y., Kasahara, M., Hirasawa, S., Namekawa, T.: A method for solving key equation for decoding Goppa codes. Information and Control 27, 87–99 (1975)

Rank for Some Families of Quaternary Reed-Muller Codes*

Jaume Pernas, Jaume Pujol, and Mercè Villanueva

Dept. of Information and Communications Engineering,
Universitat Autònoma de Barcelona, Spain
{jaume.pernas,jaume.pujol,merce.villanueva}@autonoma.edu

Abstract. Recently, new families of quaternary linear Reed-Muller codes such that, after the Gray map, the corresponding $\mathbb{Z}_4$-linear codes have the same parameters and properties as the codes in the usual binary linear Reed-Muller family have been introduced. A structural invariant, the rank, for binary codes is used to classify some of these $\mathbb{Z}_4$-linear codes. The rank is established generalizing the known results about the rank for $\mathbb{Z}_4$-linear Hadamard and $\mathbb{Z}_4$-linear extended 1-perfect codes.

Keywords: Rank, quaternary codes, Reed-Muller codes, $\mathbb{Z}_4$-linear codes.

1 Introduction

Let $\mathbb{Z}_2$ and $\mathbb{Z}_4$ be the ring of integers modulo 2 and modulo 4, respectively. Let $\mathbb{Z}_2^n$ be the set of all binary vectors of length n and let $\mathbb{Z}_4^n$ be the set of all quaternary vectors of length n. Any nonempty subset C of $\mathbb{Z}_2^n$ is a binary code and a subgroup of $\mathbb{Z}_2^n$ is called a *binary linear code* or a $\mathbb{Z}_2$*-linear code*. Equivalently, any nonempty subset $\mathcal{C}$ of $\mathbb{Z}_4^n$ is a quaternary code and a subgroup of $\mathbb{Z}_4^n$ is called a *quaternary linear code*. Some authors also use the term "quaternary codes" to refer to additive codes over $GF(4)$ [1], but note that these are not the codes we are considering in this paper.

The *Hamming distance* $d_H(u, v)$ between two vectors $u, v \in \mathbb{Z}_2^n$ is the number of coordinates in which u and v differ. The *Hamming weight* of a vector $u \in \mathbb{Z}_2^n$, denoted by $w_H(u)$, is the number of nonzero coordinates of u. The *minimum Hamming distance* of a binary code C is the minimum value of $d_H(u, v)$ for $u, v \in C$ satisfying $u \neq v$.

The Gray map, $\phi : \mathbb{Z}_4^n \longrightarrow \mathbb{Z}_2^{2n}$ given by $\phi(v_1, \ldots, v_n) = (\varphi(v_1), \ldots, \varphi(v_n))$ where $\varphi(0) = (0, 0),\ \varphi(1) = (0, 1),\ \varphi(2) = (1, 1),\ \varphi(3) = (1, 0)$, is an isometry which transforms Lee distances over $\mathbb{Z}_4^n$ into Hamming distances over $\mathbb{Z}_2^{2n}$.

Let $\mathcal{C}$ be a quaternary linear code. Since $\mathcal{C}$ is a subgroup of $\mathbb{Z}_4^n$, it is isomorphic to an abelian structure $\mathbb{Z}_2^\gamma \times \mathbb{Z}_4^\delta$. Therefore, $\mathcal{C}$ is of type $2^\gamma 4^\delta$ as a group, it has $|\mathcal{C}| = 2^{\gamma+2\delta}$ codewords and $2^{\gamma+\delta}$ codewords of order two. The binary image

* This work was supported in part by the Spanish MEC and the European FEDER under Grant MTM2006-03250.

M. Bras-Amorós and T. Høholdt (Eds.): AAECC 2009, LNCS 5527, pp. 43–52, 2009.

$C = \phi(\mathcal{C})$ of any quaternary linear code $\mathcal{C}$ of length n and type $2^\gamma 4^\delta$ is called a $\mathbb{Z}_4$-*linear code* of binary length $N = 2n$ and type $2^\gamma 4^\delta$.

Two binary codes C_1 and C_2 of length n are said to be *isomorphic* if there is a coordinate permutation π such that $C_2 = \{\pi(c) \ : \ c \in C_1\}$. They are said to be *equivalent* if there is a vector $a \in \mathbb{Z}_2^n$ and a coordinate permutation π such that $C_2 = \{a + \pi(c) \ : \ c \in C_1\}$ [11]. Two quaternary linear codes $\mathcal{C}_1$ and $\mathcal{C}_2$ both of length n and type $2^\gamma 4^\delta$ are said to be *monomially equivalent*, if one can be obtained from the other by permuting the coordinates and (if necessary) changing the signs of certain coordinates. They are said to be *permutation equivalent* if they differ only by a permutation of coordinates [9]. Note that if two quaternary linear codes $\mathcal{C}_1$ and $\mathcal{C}_2$ are monomially equivalent, then the corresponding $\mathbb{Z}_4$-linear codes $C_1 = \phi(\mathcal{C}_1)$ and $C_2 = \phi(\mathcal{C}_2)$ are isomorphic.

Two structural invariants for binary codes are the rank and dimension of the kernel. The *rank* of a binary code C, denoted by r_C, is simply the dimension of $\langle C \rangle$, which is the linear span of the codewords of C. The *kernel* of a binary code C, denoted by $K(C)$, is the set of vectors that leave C invariant under translation, i.e. $K(C) = \{x \in \mathbb{Z}_2^n \ : \ C + x = C\}$. If C contains the all-zero vector, then $K(C)$ is a binary linear subcode of C. The dimension of the kernel of C will be denoted by k_C. These two invariants do not give a full classification of binary codes, since two nonisomorphic binary codes could have the same rank and dimension of the kernel. In spite of that, they can help in classification, since if two binary codes have different ranks or dimensions of the kernel, they are nonisomorphic.

It is well-known that an easy way to built the binary linear Reed-Muller family of codes, denoted by RM, is using the Plotkin construction [11]. In [14],[15], Pujol et al. introduced new quaternary Plotkin constructions to build new families of quaternary linear Reed-Muller codes, denoted by $\mathcal{RM}_s$. The quaternary linear Reed-Muller codes $\mathcal{RM}_s(r, m)$ of length 2^{m-1}, for $m \geq 1$, $0 \leq r \leq m$ and $0 \leq s \leq \lfloor \frac{m-1}{2} \rfloor$, in these new families satisfy that the corresponding $\mathbb{Z}_4$-linear codes have the same parameters and properties (length, dimension, minimum distance, inclusion and duality relationship) as the binary linear codes in the well-known RM family. In the binary case, there is only one family. In contrast, in the quaternary case, for each m there are $\lfloor \frac{m+1}{2} \rfloor$ families, which will be distinguished using subindexes s from the set $\{0, \ldots, \lfloor \frac{m-1}{2} \rfloor\}$.

The dimension of the kernel and rank have been studied for some families of $\mathbb{Z}_4$-linear codes [2], [4], [5], [10], [12]. In the RM family, the $RM(1, m)$ and $RM(m-2, m)$ binary codes are a linear Hadamard and extended 1-perfect code, respectively. Recall that a Hadamard code of length $n = 2^m$ is a binary code with $2n$ codewords and minimum Hamming distance $n/2$, and an extended 1-perfect code of length $n = 2^m$ is a binary code with 2^{n-m} codewords and minimum Hamming distance 4. Equivalently, in the $\mathcal{RM}_s$ families, the corresponding $\mathbb{Z}_4$-linear code of any $\mathcal{RM}_s(1, m)$ and $\mathcal{RM}_s(m-2, m)$ is a Hadamard and extended 1-perfect code, respectively [14],[15]. For the corresponding $\mathbb{Z}_4$-linear codes of $\mathcal{RM}_s(1, m)$ and $\mathcal{RM}_s(m-2, m)$, the rank were studied and computed in [5],[10].

Specifically,

$$r_H = \begin{cases} \gamma + 2\delta & \text{if } s = 0, 1 \\ \gamma + 2\delta + \binom{\delta-1}{2} & \text{if } s \geq 2 \end{cases} \quad \text{and} \tag{1}$$

$r_P = \bar{\gamma} + 2\bar{\delta} + \delta = 2^{m-1} + \bar{\delta}$ (except $r_P = 11$, if $m = 4$ and $s = 0$), where $H = \phi(\mathcal{RM}_s(1, m))$ of type $2^\gamma 4^\delta$ and $P = \phi(\mathcal{RM}_s(m-2, m))$ of type $2^{\bar{\gamma}} 4^{\bar{\delta}}$.

The dimension of the kernel was computed for all $\mathcal{RM}_s(r, m)$ codes in [13]. The aim of this paper is the study of the rank for these codes, generalizing the known results about the rank for the $\mathcal{RM}_s(r, m)$ codes with $r \in \{0, 1, m-2, m-1, m\}$ [5],[10]. The paper is organized as follows. In Section 2, we recall some properties related to quaternary linear codes and the rank of these codes. Moreover, we describe the construction of the $\mathcal{RM}_s$ families of codes. In Section 3, we establish the rank for all codes in the $\mathcal{RM}_s$ families with $s \in \{0, 1\}$. Furthermore, we establish the rank for the $\mathcal{RM}_s(r, m)$ codes with $r \in \{2, m-3\}$. In Section 4, we show that the rank allows us to classify the $\mathcal{RM}_s(r, m)$ codes with $r \in \{2, m-3\}$. Finally, the conclusions are given in Section 5.

2 Preliminaries

2.1 Quaternary Linear Codes

Let $\mathcal{C}$ be a quaternary linear code of length n and type $2^\gamma 4^\delta$. Although $\mathcal{C}$ is not a free module, every codeword is uniquely expressible in the form

$$\sum_{i=1}^{\gamma} \lambda_i u_i + \sum_{j=1}^{\delta} \mu_j v_j,$$

where $\lambda_i \in \mathbb{Z}_2$ for $1 \leq i \leq \gamma$, $\mu_j \in \mathbb{Z}_4$ for $1 \leq j \leq \delta$ and u_i, v_j are vectors in $\mathbb{Z}_4^n$ of order two and four, respectively. The vectors u_i, v_j give us a generator matrix $\mathcal{G}$ of size $(\gamma + \delta) \times n$ for the code $\mathcal{C}$. In [8], it was shown that any quaternary linear code of type $2^\gamma 4^\delta$ is permutation equivalent to a quaternary linear code with a canonical generator matrix of the form

$$\begin{pmatrix} 2T & 2I_\gamma & \mathbf{0} \\ S & R & I_\delta \end{pmatrix}, \tag{2}$$

where R, T are matrices over $\mathbb{Z}_2$ of size $\delta \times \gamma$ and $\gamma \times (n - \gamma - \delta)$, respectively; and S is a matrix over $\mathbb{Z}_4$ of size $\delta \times (n - \gamma - \delta)$.

The concepts of duality for quaternary linear codes were also studied in [8], where the inner product for any two vectors $u, v \in \mathbb{Z}_4^n$ is defined as $u \cdot v = \sum_{i=1}^n u_i v_i \in \mathbb{Z}_4$. Then, the *dual code* of $\mathcal{C}$, denoted by $\mathcal{C}^\perp$, is defined in the standard way $\mathcal{C}^\perp = \{v \in \mathbb{Z}_4^n : u \cdot v = 0 \text{ for all } u \in \mathcal{C}\}$. The corresponding binary code $\phi(\mathcal{C}^\perp)$ is denoted by $C_\perp$ and called the $\mathbb{Z}_4$-*dual code* of $C = \phi(\mathcal{C})$. Moreover, the dual code $\mathcal{C}^\perp$, which is also a quaternary linear code, is of type $2^\gamma 4^{n-\gamma-\delta}$.

Let $u * v$ denote the component-wise product for any $u, v \in \mathbb{Z}_4^n$.

Lemma 1 ([6],[7]). *Let $\mathcal{C}$ be a quaternary linear code of type $2^\gamma 4^\delta$ and let $C = \phi(\mathcal{C})$ be the corresponding $\mathbb{Z}_4$-linear code. Let $\mathcal{G}$ be a generator matrix of $\mathcal{C}$ and let $\{u_i\}_{i=1}^\gamma$ be the rows of order two and $\{v_j\}_{j=0}^\delta$ the rows of order four in $\mathcal{G}$. Then, $\langle C \rangle$ is generated by $\{\phi(u_i)\}_{i=1}^\gamma$, $\{\phi(v_j), \phi(2v_j)\}_{j=1}^\delta$ and $\{\phi(2v_j * v_k)\}_{1 \leq j < k \leq \delta}$.*

2.2 Quaternary Linear Reed-Muller Codes

Recall that a binary linear rth-order Reed-Muller code $RM(r,m)$ with $0 \leq r \leq m$ and $m \geq 2$ can be described using the Plotkin construction as follows [11]:

$$RM(r,m) = \{(u|u+v) \ : \ u \in RM(r,m-1),\, v \in RM(r-1,m-1)\},$$

where $RM(0,m)$ is the repetition code $\{\mathbf{0},\mathbf{1}\}$, $RM(m,m)$ is the universe code, and "|" denotes concatenation. For $m=1$, there are only two codes: the repetition code $RM(0,1)$ and the universe code $RM(1,1)$. This RM family of codes has length 2^m, minimum distance 2^{m-r} and dimension $\sum_{i=0}^{r} \binom{m}{i}$. Moreover, the code $RM(r-1,m)$ is a subcode of $RM(r,m)$ and the code $RM(r,m)$ is the dual code of $RM(m-1-r,m)$ for $0 \leq r < m$.

In the recent literature [2],[3],[8],[16],[17] several families of quaternary linear codes have been proposed and studied trying to generalize the RM family. However, when the corresponding $\mathbb{Z}_4$-linear codes are taken, they do not satisfy all the same properties as the RM family. In [14],[15], new quaternary linear Reed-Muller families, $\mathcal{RM}_s$, such that the corresponding $\mathbb{Z}_4$-linear codes have the parameters and properties of RM family of codes, were proposed. The following two constructions are necessary to generate these new $\mathcal{RM}_s$ families.

Definition 2 (Plotkin Construction). *Let $\mathcal{A}$ and $\mathcal{B}$ be two quaternary linear codes of length n, types $2^{\gamma_\mathcal{A}}4^{\delta_\mathcal{A}}$ and $2^{\gamma_\mathcal{B}}4^{\delta_\mathcal{B}}$, and minimum distances $d_\mathcal{A}$ and $d_\mathcal{B}$, respectively. A new quaternary linear code $\mathcal{PC}(\mathcal{A},\mathcal{B})$ is defined as*

$$\mathcal{PC}(\mathcal{A},\mathcal{B}) = \{(u|u+v) \ : \ u \in \mathcal{A}, v \in \mathcal{B}\}.$$

It is easy to see that if $\mathcal{G}_\mathcal{A}$ and $\mathcal{G}_\mathcal{B}$ are generator matrices of $\mathcal{A}$ and $\mathcal{B}$, respectively, then the matrix

$$\mathcal{G}_{PC} = \begin{pmatrix} \mathcal{G}_\mathcal{A} & \mathcal{G}_\mathcal{A} \\ 0 & \mathcal{G}_\mathcal{B} \end{pmatrix}$$

is a generator matrix of the code $\mathcal{PC}(\mathcal{A},\mathcal{B})$. Moreover, the code $\mathcal{PC}(\mathcal{A},\mathcal{B})$ is of length $2n$, type $2^{\gamma_\mathcal{A}+\gamma_\mathcal{B}}4^{\delta_\mathcal{A}+\delta_\mathcal{B}}$, and minimum distance $d = min\{2d_\mathcal{A}, d_\mathcal{B}\}$ [14],[15].

Definition 3 (BQ-Plotkin Construction). *Let $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ be three quaternary linear codes of length n; types $2^{\gamma_\mathcal{A}}4^{\delta_\mathcal{A}}$, $2^{\gamma_\mathcal{B}}4^{\delta_\mathcal{B}}$, and $2^{\gamma_\mathcal{C}}4^{\delta_\mathcal{C}}$; and minimum distances $d_\mathcal{A}$, $d_\mathcal{B}$, and $d_\mathcal{C}$, respectively. Let $\mathcal{G}_\mathcal{A}$, $\mathcal{G}_\mathcal{B}$, and $\mathcal{G}_\mathcal{C}$ be generator matrices of the codes $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$, respectively. A new code $\mathcal{BQ}(\mathcal{A},\mathcal{B},\mathcal{C})$ is defined as the quaternary linear code generated by*

$$\mathcal{G}_{BQ} = \begin{pmatrix} \mathcal{G}_\mathcal{A} & \mathcal{G}_\mathcal{A} & \mathcal{G}_\mathcal{A} & \mathcal{G}_\mathcal{A} \\ 0 & \mathcal{G}'_\mathcal{B} & 2\mathcal{G}'_\mathcal{B} & 3\mathcal{G}'_\mathcal{B} \\ 0 & 0 & \hat{\mathcal{G}}_\mathcal{B} & \hat{\mathcal{G}}_\mathcal{B} \\ 0 & 0 & 0 & \mathcal{G}_\mathcal{C} \end{pmatrix},$$

where $\mathcal{G}'_\mathcal{B}$ is the matrix obtained from $\mathcal{G}_\mathcal{B}$ after switching twos by ones in their $\gamma_\mathcal{B}$ rows of order two, and $\hat{\mathcal{G}}_\mathcal{B}$ is the matrix obtained from $\mathcal{G}_\mathcal{B}$ after removing their $\gamma_\mathcal{B}$ rows of order two.

The code $\mathcal{BQ}(\mathcal{A},\mathcal{B},\mathcal{C})$ is of length $4n$, type $2^{\gamma_{\mathcal{A}}+\gamma_{\mathcal{C}}}4^{\delta_{\mathcal{A}}+\gamma_{\mathcal{B}}+2\delta_{\mathcal{B}}+\delta_{\mathcal{C}}}$, and minimum distance $d = min\{4d_{\mathcal{A}}, 2d_{\mathcal{B}}, d_{\mathcal{C}}\}$ [14],[15].

Now, the quaternary linear Reed-Muller codes $\mathcal{RM}_s(r,m)$ of length 2^{m-1}, for $m \geq 1$, $0 \leq r \leq m$, and $0 \leq s \leq \lfloor \frac{m-1}{2} \rfloor$, will be defined. For the recursive construction it will be convenient to define them also for $r < 0$ and $r > m$. We begin by considering the trivial cases. The code $\mathcal{RM}_s(r,m)$ with $r < 0$ is defined as the zero code. The code $\mathcal{RM}_s(0,m)$ is defined as the repetition code with only the all-zero and all-two vectors. The code $\mathcal{RM}_s(r,m)$ with $r \geq m$ is defined as the whole space $\mathbb{Z}_4^{m-1}$. For $m = 1$, there is only one family with $s = 0$, and in this family there are only the zero, repetition and universe codes for $r < 0$, $r = 0$ and $r \geq 1$, respectively. In this case, the generator matrix of $\mathcal{RM}_0(0,1)$ is $\mathcal{G}_{0(0,1)} = \begin{pmatrix} 2 \end{pmatrix}$ and the generator matrix of $\mathcal{RM}_0(1,1)$ is $\mathcal{G}_{0(1,1)} = \begin{pmatrix} 1 \end{pmatrix}$.

For any $m \geq 2$, given $\mathcal{RM}_s(r,m-1)$ and $\mathcal{RM}_s(r-1,m-1)$ codes, where $0 \leq s \leq \lfloor \frac{m-2}{2} \rfloor$, the $\mathcal{RM}_s(r,m)$ code can be constructed in a recursive way using the Plotkin construction given by Definition 2 as follows:

$$\mathcal{RM}_s(r,m) = \mathcal{PC}(\mathcal{RM}_s(r,m-1), \mathcal{RM}_s(r-1,m-1)).$$

For example, for $m = 2$, the generator matrices of $\mathcal{RM}_0(r,2)$, $0 \leq r \leq 2$, are the following:

$$\mathcal{G}_{0(0,2)} = \begin{pmatrix} 2\ 2 \end{pmatrix}; \quad \mathcal{G}_{0(1,2)} = \begin{pmatrix} 0\ 2 \\ 1\ 1 \end{pmatrix}; \quad \mathcal{G}_{0(2,2)} = \begin{pmatrix} 1\ 0 \\ 0\ 1 \end{pmatrix}.$$

Note that when m is odd, the $\mathcal{RM}_s$ family with $s = \frac{m-1}{2}$ can not be generated using the Plotkin construction. In this case, for any $m \geq 3$, m odd and $s = \frac{m-1}{2}$, given $\mathcal{RM}_{s-1}(r,m-2)$, $\mathcal{RM}_{s-1}(r-1,m-2)$ and $\mathcal{RM}_{s-1}(r-2,m-2)$, the $\mathcal{RM}_s(r,m)$ code can be constructed using the BQ-Plotkin construction given by Definition 3 as follows:

$$\mathcal{RM}_s(r,m) = \mathcal{BQ}(\mathcal{RM}_{s-1}(r,m-2), \mathcal{RM}_{s-1}(r-1,m-2), \mathcal{RM}_{s-1}(r-2,m-2)).$$

For example, for $m = 3$, there are two families. The $\mathcal{RM}_0$ family can be generated using the Plotkin construction. On the other hand, the $\mathcal{RM}_1$ family has to be generated using the BQ-Plotkin construction. The generator matrices of $\mathcal{RM}_1(r,3)$, $0 \leq r \leq 3$, are the following: $\mathcal{G}_{1(0,3)} = \begin{pmatrix} 2\ 2\ 2\ 2 \end{pmatrix}$;

$$\mathcal{G}_{1(1,3)} = \begin{pmatrix} 1\ 1\ 1\ 1 \\ 0\ 1\ 2\ 3 \end{pmatrix}; \quad \mathcal{G}_{1(2,3)} = \begin{pmatrix} 1\ 1\ 1\ 1 \\ 0\ 1\ 2\ 3 \\ 0\ 0\ 1\ 1 \\ 0\ 0\ 0\ 2 \end{pmatrix}; \quad \mathcal{G}_{1(3,3)} = \begin{pmatrix} 1\ 1\ 1\ 1 \\ 0\ 1\ 2\ 3 \\ 0\ 0\ 1\ 1 \\ 0\ 0\ 0\ 1 \end{pmatrix}.$$

Table 1 shows the type $2^{\gamma}4^{\delta}$ of all these $\mathcal{RM}_s(r,m)$ codes for $m \leq 10$.

The following proposition summarizes the parameters and properties of these $\mathcal{RM}_s$ families of codes.

Proposition 4 ([14],[15]). *A quaternary linear Reed-Muller code $\mathcal{RM}_s(r,m)$, with $m \geq 1$, $0 \leq r \leq m$, and $0 \leq s \leq \lfloor \frac{m-1}{2} \rfloor$, has the following parameters and properties:*

1. *the length is* $n = 2^{m-1}$;
2. *the minimum distance is* $d = 2^{m-r}$;
3. *the number of codewords is* 2^k, *where* $k = \sum_{i=0}^{r} \binom{m}{i}$;
4. *the code* $\mathcal{RM}_s(r-1,m)$ *is a subcode of* $\mathcal{RM}_s(r,m)$ *for* $0 \leq r \leq m$;
5. *the codes* $\mathcal{RM}_s(1,m)$ *and* $\mathcal{RM}_s(m-2,m)$, *after the Gray map, are* $\mathbb{Z}_4$*-linear Hadamard and* $\mathbb{Z}_4$*-linear extended perfect codes, respectively;*
6. *the code* $\mathcal{RM}_s(r,m)$ *is the dual code of* $\mathcal{RM}_s(m-1-r,m)$ *for* $-1 \leq r \leq m$.

3 Rank for Some Infinite Families of $\mathcal{RM}_s(r,m)$ Codes

In this section, we will compute the rank for some infinite families of the quaternary linear Reed-Muller codes $\mathcal{RM}_s(r,m)$. The rank of $\mathcal{RM}_s(r,m)$ will be denoted by $r_{s(r,m)}$ instead of $r_{\mathcal{RM}_s(r,m)}$.

First of all, we will recall the result that gives us which of the $\mathcal{RM}_s(r,m)$ codes are binary liner codes after the Gray map. Note that if we have a quaternary linear code of type $2^\gamma 4^\delta$ which is a binary linear code after the Gray map, we can compute the rank as $\gamma + 2\delta$ [6],[7].

Proposition 5 ([13]). *For all* $m \geq 1$, *the corresponding* $\mathbb{Z}_4$*-linear code of the* $\mathcal{RM}_s(r,m)$ *code is a binary linear code if and only if*

$$\begin{cases} s = 0 \text{ and } r \in \{0,1,2,m-1,m\}, \\ s = 1 \text{ and } r \in \{0,1,m-1,m\}, \\ s \geq 2 \text{ and } r \in \{0,m-1,m\}. \end{cases}$$

Now, we will give an expression for the parameters γ and δ of a quaternary linear Reed-Muller code $\mathcal{RM}_s(r,m)$ of type $2^\gamma 4^\delta$, depending on s, r and m.

Lemma 6. *Let* $\mathcal{C}$ *be a quaternary linear Reed-Muller code* $\mathcal{RM}_s(r,m)$ *of type* $2^\gamma 4^\delta$. *Then, for* $s \geq 0$, $m \geq 2s+1$ *and* $0 \leq r \leq m$,

$$\gamma = \sum_{i=0}^{\lfloor \frac{r}{2} \rfloor} \binom{m-2s-1}{r-2i} \binom{s}{i} \quad \text{and} \quad \delta = \frac{1}{2} \sum_{i=0}^{r} \binom{m}{i} - \frac{\gamma}{2}.$$

The next proposition gives us an important result for these quaternary linear Reed-Muller codes. In some cases, we obtain two codes with the same rank, but different s. We will prove that these codes are equal. This proposition will be used for the classification of some of the $\mathcal{RM}_s(r,m)$ codes in Section 4, and it will also be used to calculate the rank of these codes as exceptions.

Proposition 7. *Given two codes* $\mathcal{RM}_s(r,m)$ *and* $\mathcal{RM}_{s-1}(r,m)$ *of type* $2^\gamma 4^\delta$ *and* $2^{\gamma'} 4^{\delta'}$, *respectively, such that* $m \geq 3$ *is odd,* $r \geq 2$ *is even, and* $s = \frac{m-1}{2}$, *then* $\mathcal{RM}_s(r,m) = \mathcal{RM}_{s-1}(r,m)$.

Proof. The generator matrix $\mathcal{G}_{s-1(r,m)}$ of $\mathcal{RM}_{s-1}(r,m)$ is obtained using the Plotkin construction from $\mathcal{RM}_{s-1}(r-1,m-1)$ and $\mathcal{RM}_{s-1}(r,m-1)$. Furthermore, the generator matrices of $\mathcal{RM}_{s-1}(r-1,m-1)$ and $\mathcal{RM}_{s-1}(r,m-1)$ can be obtained using Plotkin construction again from codes with $m-2$ value. So we can write the generator matrix $\mathcal{G}_{s-1(r,m)}$ as follows:

$$\mathcal{G}_{s-1(r,m)} = \begin{pmatrix} \mathcal{G}_{s-1(r,m-2)} & \mathcal{G}_{s-1(r,m-2)} & \mathcal{G}_{s-1(r,m-2)} & \mathcal{G}_{s-1(r,m-2)} \\ 0 & \mathcal{G}_{s-1(r-1,m-2)} & 0 & \mathcal{G}_{s-1(r-1,m-2)} \\ 0 & 0 & \mathcal{G}_{s-1(r-1,m-2)} & \mathcal{G}_{s-1(r-1,m-2)} \\ 0 & 0 & 0 & \mathcal{G}_{s-1(r-2,m-2)} \end{pmatrix}.$$

The generator matrix $\mathcal{G}_{s(r,m)}$ of $\mathcal{RM}_s(r,m)$ can be obtained using the BQ-Plotkin construction given by Definition 3. Since r is even and m is odd, $r-1$ and $m-2$ are odd. In this case any $\mathcal{RM}_{s-1}(r-1,m-2)$ code, where $s=\frac{m-1}{2}$, is of type $2^0 4^{\delta''}$. This result can be proved by induction on m and using the BQ-Plotkin construction. Since $\mathcal{G}_{s-1(r-1,m-2)}$ is of type $2^0 4^{\delta''}$, then $\mathcal{G}'_{s-1(r-1,m-2)} = \mathcal{G}_{s-1(r-1,m-2)}$ and $\hat{\mathcal{G}}_{s-1(r-1,m-2)} = \mathcal{G}_{s-1(r-1,m-2)}$. It is easy to find a linear combination of rows that transforms the matrix $\mathcal{G}_{s-1(r,m)}$ into the matrix $\mathcal{G}_{s(r,m)}$.

Now, we will give a recursive way to compute the rank for all Reed-Muller codes in the $\mathcal{RM}_0$ and $\mathcal{RM}_1$ families. Note that the first binary nonlinear code is $\mathcal{RM}_0(3,5)$. Thus, for $m<5$ the rank is $\gamma+2\delta$.

Proposition 8. *Let $\mathcal{C}$ be a quaternary linear Reed-Muller code $\mathcal{RM}_0(r,m)$. The rank of $\mathcal{C}$ for $m \geq 5$ is*

$$r_{0(r,m)} = r_{0(r,m-1)} + r_{0(r-1,m-1)} + \begin{cases} 0 & \text{if } r \in \{0,1,2,m-1,m\} \\ \binom{m-2}{2r-3} & \text{if } r \in \{3,\ldots,m-2\} \end{cases}$$

Proposition 9. *Let $\mathcal{C}$ be a quaternary linear Reed-Muller code $\mathcal{RM}_1(r,m)$. The rank of $\mathcal{C}$ for $m \geq 5$ is*

$$r_{1(r,m)} = r_{1(r,m-1)} + r_{1(r-1,m-1)} + \begin{cases} 0 & \text{if } r \in \{0,1,m-1,m\} \\ m-2 & \text{if } r = 2 \\ 2\binom{m-1}{2r-3} & \text{if } r \in \{3,\ldots,m-2\}. \end{cases}$$

The next proposition gives the rank for all quaternary linear Reed-Muller codes with $r \in \{0,1,m-3,m-2,m-1,m\}$ and any s.

Proposition 10. *Let $\mathcal{RM}_s(r,m)$ be a quaternary linear Reed-Muller code of type $2^\gamma 4^\delta$. The rank of $\mathcal{RM}_s(r,m)$ can be computed as*

$$r_{s(r,m)} = \begin{cases} \gamma+2\delta & \text{if } r \in \{0,m-1,m\} \\ \gamma+2\delta & \text{if } r=1 \text{ and } s \in \{0,1\} \\ \gamma+2\delta+\binom{\delta-1}{2} & \text{if } r=1 \text{ and } s \geq 2 \\ 2^{m-1}+\delta & \text{if } r=m-3 \text{ and } m>6 \\ 2^{m-1}+\delta & \text{if } r=m-2 \text{ and } m>4. \end{cases}$$

Finally, the next proposition gives a recursive way to compute the rank of all quaternary linear Reed-Muller codes with $r = 2$ and any s.

Proposition 11. *Let $\mathcal{RM}_s(2, m)$ be a quaternary linear Reed-Muller code of type $2^\gamma 4^\delta$. The rank of $\mathcal{RM}_s(2, m)$ can be computed as*

$$r_{s(2,m)} = r_{s(2,m-1)} + r_{s(1,m-1)} + 2s + \binom{s+1}{2}(m - s - 3),$$

except when m is odd and $s = \frac{m-1}{2}$, since the rank is $r_{s(2,m)} = r_{s-1(2,m)}$.

When m is odd and $s = \frac{m-1}{2}$, by Proposition 7, the codes $\mathcal{RM}_s(2, m)$ and $\mathcal{RM}_{s-1}(2, m)$ are equals. Thus, the rank is also the same. Note that, for $s = 0$ and $r = 2$, we have a binary linear code and the rank is $\gamma + 2\delta$.

4 Classification of Some Families of $\mathcal{RM}_s(r, m)$ Codes

In this section, we will show that this invariant, the rank, will allow us to classify these $\mathcal{RM}_s(r, m)$ codes in some cases depending on the parameter r. This classification was given for $r = 1$ and $r = m - 2$ [5], [10]. Now, we will extend this result for $r = 2$ and $r = m - 3$. We are close to generalize this result for all $0 \leq r \leq m$, but it is not easy to obtain a general form to compute the rank for all quaternary linear Reed-Muller codes $\mathcal{RM}_s(r, m)$.

Table 1 shows the type $2^\gamma 4^\delta$ and the rank of all these $\mathcal{RM}_s(r, m)$ codes for $m \leq 10$. In these examples, you can see that the rank is always different, except for the codes quoted in Proposition 7. If two codes have different rank, we can say that they are nonisomorphic. The next theorem proves that for a given $m \geq 4$, and $r \in \{2, m - 3\}2$, the $\mathcal{RM}_s(r, m)$ codes have different rank, so they are nonisomorphic. In some cases, there is an exception, but we know by Proposition 7 that the codes are equal.

Theorem 12. *For all $m \geq 4$ and $r \in \{2, m-3\}$, there are at least $\lfloor \frac{m+1}{2} \rfloor$ nonisomorphic binary codes with the same parameters as the code $RM(r, m)$, except when m is odd, and r is even. In this case, there are at least $\frac{m-1}{2}$ nonisomorphic binary codes with the same parameters as the code $RM(r, m)$.*

Proof. By Proposition 11, we know that $r_{s(2,m)} = r_{s(2,m-1)} + r_{s(1,m-1)} + 2s + \binom{s+1}{2}(m - s - 3)$. If $r = 1$, the code is Hadamard and $r_{s(1,m-1)}$ increases or is equal to, depending on s. For $m \geq 4$ the expression $2s + \binom{s+1}{2}(m - s - 3)$ also increases, depending on s. We can suppose that $r_{s(2,m-1)}$ is crescent on s for $m = 4$ and proceed by induction on m. Therefore, $r_{s(2,m)}$ is different for every s, except when m is odd, where we have two codes with the same rank. By Proposition 7, these two codes are equal.

By Proposition 10, we know that $r_{s(m-3,m)} = 2^{m-1} + \delta$. In Proposition 6, we can see a way to compute γ. Since $r = m - 3$, then the value of γ is decreasing on s. Thus, δ is crescent and the rank is also crescent, depending on s. When m is odd and r is even, we have again the case of two equal codes, solved in Proposition 7.

Table 1. Type $2^\gamma 4^\delta$ and rank $r_{s(r,m)}$ for all $\mathcal{RM}_s(r,m)$ codes with $m \leq 10$ and $r \in \{0,1,2,3\}$, showing them in the form $(\gamma,\delta)\ r_{s(r,m)}$

m	s \ r	0	1	2		$m-3$	$m-2$	$m-1$	m
1	0	(1,0) 1	(0,1) 2					(1,0) 1	(0,1) 2
2	0	(1,0) 1	(1,1) 3	(0,2) 4			(1,0) 1	(1,1) 3	(0,2) 4
3	0	(1,0) 1	(2,1) 4	(1,3) 7		(1,0) 1	(2,1) 4	(1,3) 7	(0,4) 8
	1	(1,0) 1	(0,2) 4	(1,3) 7		(1,0) 1	(0,2) 4	(1,3) 7	(0,4) 8
4	0	(1,0) 1	(3,1) 5	(3,4) 11		(3,1) 5	(3,4) 11	(1,7) 15	(0,8) 16
	1	(1,0) 1	(1,2) 5	(1,5) 13		(1,2) 5	(1,5) 13	(1,7) 15	(0,8) 16
5	0	(1,0) 1	(4,1) 6	(6,5) 16		(6,5) 16	(4,11) 27	(1,15) 31	(0,16) 32
	1	(1,0) 1	(2,2) 6	(2,7) 21		(2,7) 21	(2,12) 28	(1,15) 31	(0,16) 32
	2	(1,0) 1	(0,3) 7	(2,7) 21		(2,7) 21	(0,13) 29	(1,15) 31	(0,16) 32
6	0	(1,0) 1	(5,1) 7	(10,6) 22	(...)	(10,16) 47	(5,26) 58	(1,31) 63	(0,32) 64
	1	(1,0) 1	(3,2) 7	(4,9) 31	(...)	(4,19) 51	(3,27) 59	(1,31) 63	(0,32) 64
	2	(1,0) 1	(1,3) 8	(2,10) 35	(...)	(2,20) 52	(1,28) 60	(1,31) 63	(0,32) 64
7	0	(1,0) 1	(6,1) 8	(15,7) 29	(...)	(15,42) 106	(6,57) 121	(1,63) 127	(0,64) 128
	1	(1,0) 1	(4,2) 8	(7,11) 43	(...)	(7,46) 110	(4,58) 122	(1,63) 127	(0,64) 128
	2	(1,0) 1	(2,3) 9	(3,13) 53	(...)	(3,48) 112	(2,59) 123	(1,63) 127	(0,64) 128
	3	(1,0) 1	(0,4) 11	(3,13) 53	(...)	(3,48) 112	(0,60) 124	(1,63) 127	(0,64) 128
8	0	(1,0) 1	(7,1) 9	(21,8) 37	(...)	(21, 99) 227	(7,120) 248	(1,127) 255	(0,128) 256
	1	(1,0) 1	(5,2) 9	(11,13) 57	(...)	(11,104) 232	(5,121) 249	(1,127) 255	(0,128) 256
	2	(1,0) 1	(3,3) 10	(5,16) 75	(...)	(5,107) 235	(3,122) 250	(1,127) 255	(0,128) 256
	3	(1,0) 1	(1,4) 12	(3,17) 82	(...)	(3,108) 236	(1,123) 251	(1,127) 255	(0,128) 256
9	0	(1,0) 1	(8,1) 10	(28,9) 46	(...)	(28,219) 475	(8,247) 503	(1,255) 511	(0,256) 512
	1	(1,0) 1	(6,2) 10	(16,15) 73	(...)	(16,225) 481	(6,248) 504	(1,255) 511	(0,256) 512
	2	(1,0) 1	(4,3) 11	(8,19) 101	(...)	(8,229) 485	(4,249) 505	(1,255) 511	(0,256) 512
	3	(1,0) 1	(2,4) 13	(4,21) 118	(...)	(4,231) 487	(2,250) 506	(1,255) 511	(0,256) 512
	3	(1,0) 1	(0,5) 16	(4,21) 118	(...)	(4,231) 487	(0,251) 507	(1,255) 511	(0,256) 512
10	0	(1,0) 1	(7,1) 11	(36,10) 56	(...)	(36,466) 978	(9,502) 1014	(1,511) 1023	(0,512) 1024
	1	(1,0) 1	(5,2) 11	(22,17) 91	(...)	(22,473) 985	(7,503) 1015	(1,511) 1023	(0,512) 1024
	2	(1,0) 1	(3,3) 12	(12,22) 131	(...)	(12,478) 990	(5,504) 1016	(1,511) 1023	(0,512) 1024
	3	(1,0) 1	(1,4) 14	(6,25) 161	(...)	(6,481) 993	(3,505) 1017	(1,511) 1023	(0,512) 1024
	3	(1,0) 1	(1,5) 17	(4,26) 172	(...)	(4,482) 994	(1,506) 1018	(1,511) 1023	(0,512) 1024

5 Conclusions

In a recent paper [15], new families of quaternary linear codes, the $\mathcal{RM}_s(r,m)$ codes, are constructed in such a way that, after the Gray map, the $\mathbb{Z}_4$-linear codes fulfill the same properties and fundamental characteristics as the binary linear Reed-Muller codes. In this paper, a structural invariant for binary codes, the rank, is used to classify some of these new families of codes. Specifically, we classified the $\mathcal{RM}_s(r,m)$ codes with $r \in \{2, m-3\}$. The $\mathcal{RM}_s(r,m)$ codes with $r \in \{0,1,m-2,m-1,m\}$ were already classified using the rank [5],[10]. As a future research, it would be interesting to compute the rank for the $\mathcal{RM}_s(r,m)$ codes with $r \in \{3,\ldots,m-4\}$ and $s \geq 2$, in order to see whether it is possible to obtain a full classification of all these $\mathcal{RM}_s(r,m)$ codes using this invariant.

In this paper, we also proved that, when m is odd, $m \geq 5$, and r is even, there are two codes with the same rank, because these two codes are equal. Moreover, we also computed the rank for all codes in the $\mathcal{RM}_0$ and $\mathcal{RM}_1$ families.

References

1. Bierbrauer, J.: Introduction to coding theory. Chapman & Hall/CRC, Boca Raton (2005)
2. Borges, J., Fernández, C., Phelps, K.T.: Quaternary Reed-Muller codes. IEEE Trans. Inform. Theory 51(7), 2686–2691 (2005)
3. Borges, J., Fernández-Córdoba, C., Phelps, K.T.: "ZRM codes". IEEE Trans. Inform. Theory 54(1), 380–386 (2008)

4. Borges, J., Phelps, K.T., Rifà, J., Zinoviev, V.A.: On $\mathbb{Z}_4$-linear Preparata-like and Kerdock-like codes. IEEE Trans. Inform. Theory 49(11), 2834–2843 (2003)
5. Borges, J., Phelps, K.T., Rifà, J.: The rank and kernel of extended 1-perfect $\mathbb{Z}_4$-linear and additive non-$\mathbb{Z}_4$-linear codes. IEEE Trans. Inform. Theory 49(8), 2028–2034 (2003)
6. Fernández-Córdoba, C., Pujol, J., Villanueva, M.: On rank and kernel of $\mathbb{Z}_4$-linear codes. In: Barbero, A. (ed.) ICMCTA 2008. LNCS, vol. 5228, pp. 46–55. Springer, Heidelberg (2008)
7. Fernández-Córdoba, C., Pujol, J., Villanueva, M.: $\mathbb{Z}_2\mathbb{Z}_4$-linear codes: rank and kernel. Discrete Applied Mathematics (2008) (submitted), arXiv:0807.4247
8. Hammons, A.R., Kumar, P.V., Calderbank, A.R., Sloane, N.J.A., Solé, P.: The $\mathbb{Z}_4$-linearity of Kerdock, Preparata, Goethals and related codes. IEEE Trans. Inform. Theory 40, 301–319 (1994)
9. Huffman, W.C., Pless, V.: Fundamentals of Error-Correcting Codes. Cambridge University Press, Cambridge (2003)
10. Krotov, D.S.: $\mathbb{Z}_4$-linear Hadamard and extended perfect codes. In: International Workshop on Coding and Cryptography, Paris, France, January 8-12, pp. 329–334 (2001)
11. MacWilliams, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes. North-Holland Publishing Company, Amsterdam (1977)
12. Phelps, K.T., Rifà, J., Villanueva, M.: On the additive ($\mathbb{Z}_4$-linear and non-$\mathbb{Z}_4$-linear) Hadamard codes: rank and kernel. IEEE Trans. Inform. Theory 52(1), 316–319 (2006)
13. Pernas, J., Pujol, J., Villanueva, M.: Kernel dimension for some families of quaternary Reed-Muller codes. In: Calmet, J., Geiselmann, W., Müller-Quade, J. (eds.) Beth Festschrift. LNCS, vol. 5393, pp. 128–141. Springer, Heidelberg (2008)
14. Pujol, J., Rifà, J., Solov'eva, F.I.: Quaternary Plotkin constructions and quaternary Reed-Muller codes. In: Boztaş, S., Lu, H.-F(F.) (eds.) AAECC 2007. LNCS, vol. 4851, pp. 148–157. Springer, Heidelberg (2007)
15. Pujol, J., Rifà, J., Solov'eva, F.I.: Construction of $\mathbb{Z}_4$-linear Reed-Muller codes. IEEE Trans. Inform. Theory 55(1), 99–104 (2009)
16. Solov'eva, F.I.: On Z4-linear codes with parameters of Reed-Muller codes. Problems of Inform. Trans. 43(1), 26–32 (2007)
17. Wan, Z.-X.: Quaternary codes. World Scientific Publishing Co. Pte. Ltd., Singapore (1997)

Optimal Bipartite Ramanujan Graphs from Balanced Incomplete Block Designs: Their Characterizations and Applications to Expander/LDPC Codes

Tom Høholdt[1] and Heeralal Janwal[2]

[1] Institute for Mathematics, Technical University of Denmark, Denmark
T.Hoeholdt@mat.dtu.dk
[2] Department of Mathematics and Computer Science,
University of Puerto Rico (UPR), Rio Piedras Campus, P.O. Box: 23355,
San Juan, PR 00931 - 3355
hjanwa@upr.edu

Abstract. We characterize optimal bipartite expander graphs and give necessary and sufficient conditions for optimality. We determine the expansion parameters of the BIBD graphs and show that they yield optimal expander graphs that are also bipartite Ramanujan graphs. In particular, we show that the bipartite graphs derived from finite projective and affine geometries yield optimal Ramanujan graphs. This in turn leads to a theoretical explanation of the good performance of a class of LDPC codes.

Keywords: Bipartite graphs, expander graphs, eigenvalues of graphs, Ramanujan graphs, BIBD, finite geometries, LDPC and expander codes.

1 Introduction

An expander graph is a highly connected "sparse" graph (see, for example [51]). Expander graphs have numerous applications including those in communication science, computer science (especially complexity theory), network design, cryptography, combinatorics and pure mathematics (see the books and articles in the Bibliography).

Expander graphs have played a prominent role in recent developments in coding theory (LDPC codes, expander codes, linear time encodable and decodable codes, codes attaining the Zyablov bound with low complexity of decoding)

(see [55], [53], [54], [48], [57], [58], [38] [8],[27], [26], [4], [13], and others).

We shall consider graphs $\mathcal{X} = (V, E)$, where V is the set of vertices and E is the set of edges of $\mathcal{X}$. We will assume that the graph is undirected and connected and we shall only consider finite graphs. For $F \subset V$, the *boundary* ∂F is the set of edges connecting F to $V \setminus F$. The *expanding constant*, or *isoperimetric constant* of $\mathcal{X}$ is defined as,

$$h(\mathcal{X}) = \min_{\emptyset \neq F \subset V} \frac{|\partial F|}{\min\{|F|, |V \setminus F|\}}$$

M. Bras-Amorós and T. Høholdt (Eds.): AAECC 2009, LNCS 5527, pp. 53–64, 2009.

If $\mathcal{X}$ is viewed as the graph of a communication network, then $h(\mathcal{X})$ measures the "quality" of the network as a transmission network. In all applications, the larger the $h(\mathcal{X})$ the better, so we seek graphs (or families of graphs) with $h(\mathcal{X})$ as large as possible with some fixed parameters.

It is well-known that the expansion properties of a graph are closely related to the eigenvalues of the *adjacency matrix* A of the graph $\mathcal{X} = (V, E)$; it is indexed by pairs of vertices x, y of $\mathcal{X}$ and A_{xy} is the number of edges between x and y. When $\mathcal{X}$ has n vertices, A has n real eigenvalues, repeated according to multiplicities that we list in decreasing order

$$\mu_0 \geq \mu_1 \geq \ldots \geq \mu_{n-1}$$

It is also known that if $\mathcal{X}$ is D-regular, i.e. all vertices have degree D, then $\mu_0 = D$ and if morover the graph is connected $\mu_1 < D$. Also $\mathcal{X}$ is bipartite if and only if $-\mu_0$ is an eigenvalue of A. We recall the following

Theorem 1. *Let $\mathcal{X}$ be a finite, connected , D-regular graph then*

$$(D - \mu_1)/2 \leq h(\mathcal{X}) \leq \sqrt{2D(D - \mu_1)}$$

and

Theorem 2. *Let $(\mathcal{X}_m)_{m\geq 1}$ be a family of finite connected, D-regular graphs with $|V_m| \to +\infty$ as $m \to \infty$. Then*

$$\liminf_{N\to\infty} \mu_1(\mathcal{X}_m) \geq 2\sqrt{D-1}.$$

This leads to the following

Definition 1. *A finite connected, D-regular graph $\mathcal{X}$ is* Ramanujan *if, for every eigenvalue μ of A other than $\pm D$, one has*

$$\mu \leq 2\sqrt{D-1}.$$

We will also need

Definition 2 (Bipartite Ramanujan Graphs). *Let $\mathcal{X}$ be a (c, d)-regular bipartite graph. Then $\mathcal{X}$ is called a Ramanujan graph if $\mu_1(\mathcal{X}) \leq \sqrt{c-1} + \sqrt{d-1}$.*

In this article, we address the optimality of expander graphs and the property of being Ramanujan for:

A D-regular graphs;
B (c, d)-regular bipartite graphs;
C Irregular bipartite graphs.

We also show that the bipartite graphs of Balanced Incomplete Block Designs yield optimal Ramanujan graphs (meaning that we have equality in definition 1). Furthermore, we show that the bipartite graphs derived from finite projective and affine geometries (PG/EG) also yield optimal Ramanujan graphs.

2 Optimal Expander Graphs and Balanced Incomplete Block Designs

Unless otherwise specified, for background on block designs, we follow Hall [14].

Definition 3. *A balanced incomplete block design (BIBD) $\mathcal{D}$ with parameters (v, b, r, k, λ) is an incidence structure with a set $\mathcal{V}$ of v distinct varieties (or objects) denoted $a_1, \cdots, a_v$ and a set $\mathcal{B}$ of b distinct blocks denoted $B_1, \cdots, B_b$, such that each of the b blocks is incident with k varieties, and each of the v varieties is incident with r blocks, and every pair of varieties is incident with precisely λ blocks. The design is called symmetric if $b = v$ and its parameters are denoted by (v, k, r).*

Proposition 1. *For a (BIBD) $\mathcal{D}$ with parameters (v, b, r, k, λ),*

1. $b \cdot k = v \cdot r$
2. $r(k-1) = \lambda(v-1)$

Remark 1. *For a BIBD, Fisher's inequality implies that $b \geq v$ and hence $r \geq k$ and for symmetric designs $b = v$, and $r = k$. Furthermore, in any (v, k, r) symmetric block design, every pair of blocks is incident with precisely λ varieties, and if v is even then $k - \lambda$ is a square.*

Definition 4. *A BIBD can be described by a $v \times b$ incidence matrix $H := (h_{ij})$, where for $1 \leq i \leq v$ and $1 \leq j \leq b$, $h_{ij} = 1$ if a_i is incident with B_j, and $h_{ij} := 0$ else.*

Then $\mathcal{D}$ is a block design if and only if the following system of equations hold.

$$B := HH^T = (r - \lambda)I + \lambda J_v \qquad (1^v)^T H = k1^b \tag{1}$$

where J_v is the $v \times v$ all 1's matrix, and 1^v and 1^b are all 1's vectors of appropriate lengths.

2.1 The Bipartite Graph of a BIBD

Definition 5. *Let $\mathcal{D}$ be a BIBD. The bipartite graph $\mathcal{X}_\mathcal{D}$ has left set of vertices $\mathcal{V}$ and right set of vertices $\mathcal{B}$ and the adjacency of the left and right vertices is defined by the incidence structure of the design.*

It is clear that the left vertices of $\mathcal{X}_\mathcal{D}$ all have degree r and the right vertices all have degree k so the graph is what is called (r, k)*-regular*. In the symmetric case all vertices have degree r so the graph is r*-regular*.

The adjacency matrix of the bipartite graph $\mathcal{X}_\mathcal{D}$ is then,

$$A = \begin{bmatrix} \mathbf{0} & H \\ H^T & \mathbf{0} \end{bmatrix} \tag{2}$$

We will from the following proposition from [18] determine the eigenvalues of A.

Proposition 2. *Let M be a Hermitian matrix.*

1. *M is diagonalizable by a Unitary matrix.*
2. *The geometric multiplicity of an eigenvalue of M equals its algebraic multiplicity.*
3. *All the eigenvalues of M are real.*
4. *If $P_M(x) = \prod_{i=1}^{l}(x-\lambda_i)^{m_i}$ is the characteristic polynomial of M with distinct real eigenvalues λ_i with algebraic multiplicities m_i, then the minimal polynomial of M is $m_M(x) = \prod_{i=1}^{l}(x-\lambda_i)$*

With notations as above we can now prove

Lemma 1. *1. The characteristic and minimal polynomials of $B = HH^T$ are*
$P_B(x) = (x - r\cdot k)(x-(r-\lambda))^{v-1}$,
$m_B(x) = (x - r\cdot k)(x-(r-\lambda))$.
2. The characteristic and minimal polynomials of $C = H^TH$ are,
$P_C(x) = x^{b-v}(x - r\cdot k)(x-(r-\lambda))^{v-1}$,
$m_C(x) = x(x - r\cdot k)(x-(r-\lambda))$.

Proof. The matrix J_v has eigenvalue v with multiplicity 1 and corresponding eigenvector 1^v, and eigenvalue 0 with multiplicity $(v-1)$ with eigenspace $W = \{Y | Y \perp 1^v\}$. And all eigenvectors are also eigenvectors of I. From the relation 1, we conclude that $r\cdot k$ is an eigenvalue of B with eigenvector 1^v, and $(r-\lambda)$ with eigenspace W, and hence multiplicity $(v-1)$. Therefore, $P_B(X) = (x - r\cdot k)(x-(r-\lambda))^{v-1}$. Since B is a symmetric matrix, from Proposition 2, we have $m_B(x) = (x - r\cdot k)(x-(r-\lambda))$.

Since HH^T and H^TH have the same nonzero eigenvalues with the same multiplicities ([11] p.186) we get

$P_C(x) = x^{b-v}(x - r\cdot k)(x-(r-\lambda))^{v-1}$,

and from Propostion 2 $m_C(x) = x(x - r\cdot k)(x-(r-\lambda))$. Q.E.D.

This in turn leads to

Theorem 3. *The adjacency matrix of the A of the bipartite graph $\mathcal{X}_{\mathcal{D}}$ of a (v,b,r,k,λ) BIBD has characteristic polynomial $P_A(x) = (x-\sqrt{k\cdot r})(x+\sqrt{k\cdot r})(x-\sqrt{r-\lambda})^{v-1}(x+\sqrt{r-\lambda})^{v-1}x^{b-v}$, and minimal polynomial $m_A(x) = (x-\sqrt{k\cdot r})(x+\sqrt{k\cdot r})(x-\sqrt{r-\lambda})(x+\sqrt{r-\lambda})x$. In particular, the eigenvalues are $\sqrt{k\cdot r}$ with multiplicity 1, $\sqrt{r-\lambda}$ with multiplicity $v-1$, 0 with multiplicity $b-v$, $-\sqrt{r-\lambda}$ with multiplicity $v-1$, and $-\sqrt{k\cdot r}$ with multiplicity 1.*

Proof. First

$$AA^T = A^2 = \begin{bmatrix} HH^T & \mathbf{0} \\ \mathbf{0} & H^TH \end{bmatrix} \tag{3}$$

Therefore, the characteristic polynomial $P_{A^2}(x) = P_B(x)P_C(x)$. Therefore, from Lemma 1, $P_{A^2}(x) = (x - r\cdot k)^2(x-(r-\lambda))^{2(v-1)}x^{b-v}$. It is clear that

$P_{A^2}(x^2) = P_A(x)P_A(-x)$ and since the graph is bipartite we have $P_A(-x) = (-1)^{v+b}P_A(x)$ ([11] p. 31) and hence

$$(-1)^{v+b}(P_A(x))^2 = P_{A^2}(x^2) = (x^2 - r \cdot k)^2(x^2 - (r - \lambda))^{2(v-1)}x^{2(b-v)}$$

so we have

$$P_A(x) = \pm(x^2 - r \cdot k)(x^2 - (r - \lambda))^{v-1}x^{b-v} \tag{4}$$

and the theorem follows directly. Q.E.D.

Theorem 4. *(I) The bipartite graph $\mathcal{X}_{\mathcal{D}}$ of a (v, b, r, k, λ) BIBD is a (r, k)-regular bipartite Ramanujan graph with $\mu_1 = \sqrt{r - \lambda}$.*
(II) The r-regular graph of a symmetric BIBD is an r-regular bipartite Ramanujan graph with $\mu_1 = \sqrt{r - \lambda}$.

Proof. The proofs of (I) and (II) follow from Theorem 1 by applying the definition of Ramanujan graphs and following the inequalities. Q.E.D.

We also will show that the bipartite graph of a BIBD is an optimal expander graph for its parameters.

3 Characterization of Optimal Bipartite (c, d) Regular Expander Graphs

We recall that a matrix A with rows and columns indexed by a set X is called irreducible when it is not possible to find a proper subset of X so that $A(x, y) = 0$ whenever $x \in S$ and $y \in X \setminus S$. Equivalently, A is not irreducible if and only if it is possible to apply a simultaneous row and column permutation a matrix in a square block form so that one of the blocks is a zero block. For the following Lemma, see for example ([18] p. 363).

Lemma 2. *Let D be a finite graph. Then the adjacency matrix of A is irreducible if and only if D is connected.*

We shall also need

Proposition 3 (Perron-Frobenius). *Let A be an irreducible non-negative matrix. Then, there is up to scalar multiples, a unique non-negative eigenvector $\mathbf{a} := (a_1, a_2, \cdots, a_n)$ all of whose coordinates a_i are strictly positive. The corresponding eigenvalue μ_0 (called the dominant eigenvalue of A has algebraic multiplicity 1 and $\mu_0 \geq \mu_i$ for any eigenvalue μ_i of A.*

We recall the following special case of Courant-Fisher Theorem (called the Raleigh-Ritz Theorem) (see for example, ([18], Theorem 4.2.2))

Theorem 5. *Let A be an $n \times n$ Hermitian matrix over the complex field $\mathcal{C}$, then it is known that all its eigenvalues are real, with maximum eigenvalue $\mu_{\max}$. For $\mathbf{0} \neq X \in \mathcal{C}^n$, define the Raleigh quotient $R_X := \frac{X^{*T}AX}{X^{*T}X}$. Then $\mu_{\max} = \max_{X \neq \mathbf{0}} R_X$. Furthermore, $R_X \leq \mu_{\max}$ with equality if and only if $\mathcal{X}$ is an eigenvector corresponding to the eigenvalue $\mu_{\max}$.*

Definition 6. *Let $\mathcal{X}$ be a bipartite graph with average left degrees $\overline{c}$ and average right degree $\overline{d}$. Define $v := |E|/\overline{c}$ and $b := |E|/\overline{d}$. Let $\mu_1 := \max_\gamma\{|\gamma||\gamma \neq \mu_{\max}\}$, where the maximization is over the eigenvalues of $\mathcal{X}$, and $\mu_{\max}$ is the maximum eigenvalue of the adjancency matrix of $\mathcal{X}$.*

Theorem 6. *Let $\mathcal{X}$ be a connected bipartite graph with average left degrees $\overline{c}$ and average right degree $\overline{d}$ and maximum eigenvalue $\mu_{\max}$. Then $\sqrt{\overline{c}\overline{d}} \leq \mu_{\max}$ with equality if and only is $\mathcal{X}$ is a (c,d)-regular bipartite graph.*

PROOF. If $\mathcal{X}$ is a (c,d)-regular bipartite graph, with adjancy matrix

$$A = \begin{bmatrix} \mathbf{0} & H \\ H^T & \mathbf{0} \end{bmatrix}$$

then it is easy to see that $[1,1,\ldots,1,\tau,\ldots,\tau]$ where $\tau = \sqrt{\frac{d}{c}}$ is an eigenvector corresponding to the eigenvalue $\sqrt{cd}$ and then the inequality and equality follow from the Perron-Frobenius theorem (3). Conversely, assume that $\mathcal{X}$ is a connected bipartite graph on v left vertices of average degree $\overline{c}$ and b right vertices of average degree $\overline{d}$ and maximum eigenvalue $\mu_{\max}$. Define $Z := [1,1,\cdots,1,\tau,\cdots,\tau]^T$, in which the first v component are 1, and where $\tau := \sqrt{\frac{\overline{d}}{\overline{c}}}$.

Then one can see that $R_Z = \sqrt{\overline{c}\overline{d}}$ and therefore $\sqrt{\overline{c}\overline{d}} \leq \mu_{max}$. If we have equality then $R_Z = \mu_{\max}$, and from the Courant-Fisher Theorem Z is an eigenvector corresponding to the maximal eigenvalue R_Z. Therefore, $AZ = \mu_{\max}Z$. By assumption, then $AZ = \sqrt{\overline{c}\overline{d}}Z$.

By solving the simultaneous equations, we get that $c_{v_i} = \overline{c}$ for every left vertex v_i, and $d_{b_j} = \overline{d}$ for every right vertex b_j. Consequently, $\mathcal{X}$ is a (c,d) regular graph. Q.E.D.

Theorem 7. *Let $\mathcal{X}$ be a connected graph such that both $\pm\mu_{\max}$ are eigenvalues (i.e. it is a bipartite with say v left vertices and b right vertices). Suppose that*

$$\left(\frac{1}{2v}\sum_{\mu_i}{\mu_i}^2\right)\cdot\left(\frac{1}{2b}\sum_{\mu_i}{\mu_i}^2\right) = \mu_{\max}^2.$$

Then $\mathcal{X}$ is a (c,d)-regular graph, where $c := |E|/v$ and $d := |E|/b$

PROOF. By definition of the adjacency matrix, $Trace(AA^T) = 2|E|$. But $Trace$ $(AA^T) = \sum_{\mu_i}{\mu_i}^2$ Since, $\overline{c} := |E|/v$ and $\overline{d} = |E|/b$, by assumption we get $\overline{c}\overline{d} = \mu_{\max}^2$ and therefore by Theorem 6, $\mathcal{X}$ is a (c,d) regular graph with $c := |E|/v$ and $d := |E|/b$. Q.E.D.

Theorem 8. *Suppose that $\mathcal{X}$ is a bipartite graph with v left vertices and b right vertices where $b \geq v \geq 2$. Suppose the eigenvalues are $\pm\alpha$ with multiplicity 1, $\pm\beta$ with multiplicity $(v-1)$, 0 with multiplicity $(b-v)$ (and $\alpha > \beta > 0$). If*

$$\alpha^2 + (v-1)\beta^2 = \alpha^2 bv$$

then $\mathcal{X}$ is the graph of a balanced incomplete block design with parameters (b, v, r, k, λ), where
$r = \frac{\alpha^2+(v-1)\beta^2}{b}$, $k = \frac{\alpha^2+(v-1)\beta^2}{v}$ *and* $\lambda = \sqrt{r-\beta^2}$.

PROOF. With r and k as above we have that $r = \frac{|E|}{v}$ is the avearge degree of the left vertices and $k = \frac{|E|}{b}$ is the average degree of the right vertices. But then by the condition we get $\sqrt{rk} = \alpha$ so by Theorem 7, $\mathcal{X}$ is a bipartite (r, k)-regular graph.

Let $X = [\sqrt{r}, \sqrt{r}, \cdots, \sqrt{r}, \sqrt{k}, \sqrt{k}, \cdots, \sqrt{k}]$, where the multiplicity of $\sqrt{r}$ is v, and $\sqrt{k}$ is b.

Then we can verify that $AX = \sqrt{r \cdot k}X$.

Therefore,

$$A^T AX = \sqrt{r \cdot k}AX = (rk)X \tag{5}$$

Let $A = \begin{bmatrix} \mathbf{0} & H \\ H^T & \mathbf{0} \end{bmatrix}$. Then $AA^T = A^2 = \begin{bmatrix} HH^T & \mathbf{0} \\ \mathbf{0} & H^T H \end{bmatrix}$.

Let $B = HH^T$, where H is the left-right incidence matrix of the bipartite graph, and let $Q = H^T H$.

Hence, from the definition of H, we can confirm that $BY = r \cdot kY$, where $Y = [\sqrt{r}, \sqrt{r}, \cdots, \sqrt{r}]^T$. Consequently $Z = [1, 1, \cdots, 1]^T$ is an eigenvector corresponding to the eigenvalue $r \cdot k$.

Then by ([11] p.186), B has eigenvalues $\alpha = r \cdot k$ with multiplicity 1, β with multiplicity $(v-1)$.

Therefore, the minimal polynomial of B is $m_B(x) = (x-\alpha)q(x)$, where $q(x) = (x-\beta)$. (Since B is real symmetric, it is diagonalizable by an orthonormal basis, and therefore its minimal polynomial is composed of distinct factors).

Substituting B for X, we have $Bq(B) = r \cdot kq(B)$. Since α is a simple eigenvalue of B with eigenvector Z, $q(B)$ has columns that are multiples of Z. However $q(B)$ is symmetric, as B is symmetric, we conclude that all the column multiples are the same scalar c, i.e.

$$q(B) = cJ,$$

where J is the $v \times v$ matrix of all 1's. Hence $B - \beta^2 I_v = B - (r-\lambda)I_v = c \cdot J$. Hence, $B = (r-\lambda)I_v + c \cdot J_v$.

By taking Trace both sides, and since $Trace(B) = Trace(HH^T) = rv$, we conclude that $c = \lambda$. Hence

$$B = HH^T = (r-\lambda)I + \lambda J,$$

and $1^v H = k.1^b$ since the graph is a bipartite graph. Therefore, H is the incidence matrix of a (v, b, r, k, λ) BIBD. Q.E.D.

3.1 Bounds on the Eigenvalues of Irregular Bipartite Graphs

Theorem 9. *Let $\mathcal{X}$ be a connected bipartite graph with average left degree $\overline{c}$ and average right degree $\overline{d}$. Let $v := |E|/\overline{c}$ and $b := |E|/\overline{d}$ (we assume $b \geq v$) and maximum eigenvalue μ_{max}. Then*

$$\mu_1 \geq \left(\frac{|E| - \mu_{\max}^2}{v-1} \right)^{1/2}$$

with equality if and only if the eigenvalues are $\pm\mu_{\max}$ *(with multiplicity 1),* $\pm\mu_1$ *(with multiplicity* $(v-1)$*, and* 0 *with multiplicity* $b-v$*.*

PROOF. Since $\mathcal{X}$ is a connected bipartite graph, the set of eigenvalues is $S(\mathcal{X}) := \{\pm\mu_{\max}, \pm\mu_1, \pm\mu_2, \pm\cdots\pm\mu_{N-1}, 0\}$ with $2N$ non-zero eigenvalues, where the absolute values are in decreasing order.

Since $AA^T = A^2 = \begin{bmatrix} HH^T & \mathbf{0} \\ \mathbf{0} & H^TH \end{bmatrix}$

we get by taking the trace, $Trace(A^2) = Trace(HH^T) + Trace(H^TH) = v\overline{c} + b \cdot \overline{d} = |E| + |E| = 2|E|$. Therefore, $2|E| = Trace(A^2) = 2{\mu_{max}}^2 + 2{\mu_1}^2 + 2\sum_{i=2}^{N-1}[\mu_i^2 \leq 2{\mu_{max}}^2 + 2(N-1){\mu_1}^2$. Hence

$$\mu_1 \geq \left(\frac{|E| - \mu_{\max}^2}{N-1} \right)^{1/2} \geq \left(\frac{|E| - \mu_{\max}^2}{v-1} \right)^{1/2}$$

where the last inequality follows from the fact that $N \leq v$. It is clear that equality occurs if and only if the eigenvalues are $\pm\mu_{\max}$ with multiplicity 1, $\pm\mu_1$ with multiplicity $(v-1)$, and 0 with multiplicity $b-v$.

Q.E.D.

Corollary 1. *Let* $\mathcal{X}$ *be a connected* (c,d)*-regular bipartite graph. Define* $v := |E|/c$ *and* $b := |E|/d$*. Then* $\mu_{\max} = \sqrt{c \cdot d}$*, and*

$$\mu_1 \geq \left(\frac{|E| - \mu_{\max}^2}{v-1} \right)^{1/2}$$

with equality if the eigenvalues are $\pm\mu_{\max}$ *(with multiplicity 1),* $\pm\mu_1$ *(with multiplicity* $(v-1)$*), and* 0 *with multiplicity* $b-v$*.*

We can now derive the central theorem in this paper:

Theorem 10. *The bipartite graph* $\mathcal{X}_\mathcal{D}$ *of a* (v,b,r,k,λ) *BIBD is a* (r,k)*-regular bipartite Ramanujan graph with* $\mu_1(X) = \sqrt{r-\lambda}$*, and it is optimal expander graph with these parameters.*

4 Optimal Ramanujan Graphs from Finite Geometries

In this we consider special type of BIBD that are highly structured, namely those coming from Finite Affine and Projective Geometries. The corresponding class of bipartite graphs from $PG(n, \mathbb{F}_q)$ and $EG(n, \mathbb{F}_q)$ coming from subspaces, yield us, by way of Theorem 12 optimal bipartite expander graphs. The class has been used to construct good LDPC codes ([38], [39] , [40]) and our results on the eigenvalues and hence the expansion coefficient give a partial theoretical explanation of the good performance of these codes.

4.1 Optimal Bipartite Ramanujan Graphs of Projective Geometries

Let $\mathbb{F}_q$ be the finite field with $q = p^m$ elements. Let $n \geq 2$ be an integer, and let s be an integer such that $1 \leq s \leq n-1$.

We consider the incidence structure $\mathcal{X}(n, s, q) = (V, \mathcal{B})$, where $V =$ the points of the n-dimensional projective geometry $PG(n, q)$, and $\mathcal{B} = \{S|$ dim $S = s$ and S a subspace of $PG(n, q)\}$.

Proposition 4 (Hall [14]). *The graph of the incidence structure $\mathcal{X}(n, s, q)$ is a bipartite graph of a (v, b, r, k, λ) BIBD with parameters $b = \frac{(q^{n+1}-1)(q^{n+1}-q)\cdots(q^{n+1}-q^s)}{(q^{s+1}-1)(q^{s+1}-q)\cdots(q^{s+1}-q^s)}$, $v = \frac{(q^{n+1}-1)}{(q-1)}$, $k = \frac{(q^{s+1}-1)}{(q-1)}$, $r = \frac{(q^n-1)\cdots(q^n-q^{s-1})}{(q^s-1)\cdots(q^s-q^{s-1})}$, $\lambda = \frac{(q^{n+1}-q^2)\cdots(q^{n+1}-q^s)}{(q^{s+1}-q^2)\cdots(q^{s+1}-q^s)}$.*

Theorem 11. *The bipartite graph $\mathcal{X}(n, s, q)$ is a (r, k)-regular bipartite Ramanujan graph with $\mu_1(X) = \sqrt{r-\lambda}$, and it is optimal expander graph with these parameters.*

4.2 Optimal Bipartite Ramanujan Graphs of Affine Geometries

Let $\mathbb{F}_q$ be the finite field with $q = p^m$ elements. Let $n \geq 2$ be an integer, and let s be an integer such that $1 \leq s \leq n-1$.

We consider the incidence structure $\mathcal{Y}(n, s, q) = (V, \mathcal{B})$, where $V =$ the points of the n-dimensional affine geometry $EG(n, q)$, and $\mathcal{B} = \{S|$ dim $S = s$ and S a subspace of $EG(n, q)\}$.

Proposition 5 (Hall [14]). *The graph of the incidence structure $\mathcal{Y}(n, s, q)$ is a bipartite graph of a (v, b, r, k, λ) BIBD with parameters $b = \frac{q^n(q^n-1)(q^n-q)\cdots(q^{n+1}-q^{s-1})}{q^s(q^s-1)(q^{s+1}-q)\cdots(q^s-q^{s-1})}$, $v = q^n$, $b = \frac{(q^n-1)(q^n-q)\cdots(q^{n+1}-q^{s-1})}{(q^s-1)(q^{s+1}-q)\cdots(q^s-q^{s-1})}$, $k = q^s$, $\lambda = \frac{(q^n-q^2)\cdots(q^n-q^{s-1})}{(q^s-q)\cdots(q^{s+1}-q^{s-1})}$.*

Theorem 12. *The bipartite graph $\mathcal{Y}(n, s, q)$ is a (r, k)-regular bipartite Ramanujan graph with $\mu_1(X) = \sqrt{r-\lambda}$, and it is optimal expander graph with these parameters.*

5 Conclusion

In this article we have considered a special type of expander graphs coming from balanced incomplete block designs, in particular from finite geometries. We have shown that these yield optimal Ramanujan graphs.

In a forthcoming paper, Janwa and Høholdt [17], we study the expander codes and LDPC codes that are derived from BIBD and codes based on finite affine and projective geometric based bipartite graph and provide theoretical bounds on their parameters such as distance and rate, and derive some excellent codes.

References

1. Alon, N.: Eigenvalues and expanders. Combinatorica 6, 83–96 (1986)
2. Alon, N., Bruck, J., Naor, J., Naor, M., Roth, R.M.: Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs. IEEE Trans. on Inform. Theory 38(2), 509–516 (1992)
3. Alon, N., Lubotzky, A., Wigderson, A.: Semi-direct product in groups and zig-zag product in graphs: Connections and applications. In: Proc. of the 42nd FOCS, pp. 630–637 (2001)
4. Barg, A., Zemor, G.: Error exponent of expander codes. IEEE Trans. Inform. Theory 48(6), 1725–1729 (2002)
5. Bass, H.: The Ihara-Selberg zeta function of a tree lattice. International Journal of Mathematics 3(6), 717–797 (1992)
6. Biggs, N.: Algebraic Graph Theory, 2nd edn. Cambridge University Press, Cambridge (1994)
7. Cvetkovic, D.M., Doob, M., Sachs, H.: Spectra of Graphs: Theory and Applications. Academic Press, London (1979)
8. David Forney, G.: Codes on graphs: recent progress. Phys. A 302(104), 1–13 (2001)
9. Frey, B.J., Koetter, R., Forney, G.D., Kschischang, F.R., McEliece, R.J., Spielman, D.A. (eds.): Special Issue on Codes on Graphs and Iterative Algorithms. IEEE Trans. Inform. Theory 47 (2001)
10. Friedman, J. (ed.): Expanding Graphs. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 10. AMS (1993)
11. Godsil, C.D.: Algebraic Combinatorics. Chapman and Hall, Boca Raton (1993)
12. Guo, Q., Janwa, H.: Some properties of the zig-zag product of graphs. In: Presented at the 2nd NASA-EPSCoR meeting, November 10 (2003) (preprint)
13. Guruswami, V., Indyk, P.: Expander-Based Construction of Efficiently Decodable Codes. In: 42nd IEEE Symposium on Foundation of Computer Science, October 14-17, p. 658 (2001)
14. Hall Jr., M.: Combinatorial Theory, 2nd edn. Wiley Interscience in discrete Mathematics. Kluwer, Dordrecht (2003); Høholdt, T., Justesen, J.: Graph codes with Reed-Solomon component codes (submitted)
15. Høholdt, T., Justesen, J.: Graph codes with Reed-Solomon component codes. In: Proc. ISIT 2006, pp. 2002–2006 (2006)
16. Høholdt, T., Justesen, J.: Iterated analysis of iterated hard decision decoding of product codes with Reed-Solom component codes. In: Proc. ITW 2007 (September 2007)
17. Høholdt, T., Janwa, H.: Characterization of some optimal expander graphs and their application to expander/LDPC codes (in preparation)
18. Horn, R.A., Johnson, C.R.: Matrix Analysis. Cambridge University Press, Cambridge (1992)
19. Janwa Jr., H.: "Relations among Expander Graphs, Codes, Sequence Design, and Their Applications. In: Proceedings of the 4th World Multi-conference on Systemics, Cybernetics and Informatics (SCI 2000 and ISAS 2000), Orlando Florida, July 23-26, vol. XI, pp. 122–124 (2000)
20. Janwa, H.: New (Explicit) Constructions of Asymptotic Families of Constant Degree Expander Graphs from Algebraic Geometric (AG) Codes and Their Applications to Tanner Codes. ABSTRACTS OF THE AMS 26(1), 197 (2005); Joint AMS annual Meeting, Atlanta, 2005; Special session on Algebraic Geometry and Codes

21. Janwa, H.: New Constructions of Sparse Quasi-Random Graphs (in preparation)
22. Janwa, H.: Further examples of Ramanujan graphs from AG codes (2003) (in preparation)
23. Janwa, H.: Covering radius of codes, expander graphs, and Waring's problem over finite fields (in preparation)
24. Janwa, H.: A graph theoretic proof of the Delsarte bound on the covering radius of linear codes (preprint)
25. Janwa, H.: Good Expander Graphs and Expander Codes: Parameters and Decoding. In: Fossorier, M.P.C., Høholdt, T., Poli, A. (eds.) AAECC 2003. LNCS, vol. 2643, pp. 119–128. Springer, Heidelberg (2003)
26. Janwa, H., Lal, A.K.: "On Expander Graphs: Parameters and Applications (January 2001) (submitted), arXiv:cs.IT/04060-48v1
27. Janwa, H., Lal, A.K.: On Tanner Codes: Parameters and Decoding. Applicable Algebra in Engineering, Communication and Computing 13, 335–347 (2003)
28. Janwa, H.: Explicit Constructions of Asymptotic Families of Constant Degree Expander Graphs from Algebraic Geometric (AG) Codes. Congressus Numerantium 179, 193–207 (2006)
29. Janwa, H., Lal, A.K.: On the Generalized Hamming Weights and the Covering Radius of Linear Codes. LNCS, vol. 4871, pp. 347–356. Springer, Heidelberg (2007)
30. Janwa, H., Mattson Jr., H.F.: "The Projective Hypercube and Its Properties and Applications. In: Proceedings of the 2001 International Symposium on Information Theory (ISIT 2001), Washington D.C., USA (June 2001)
31. Janwa, H., Moreno, O.: Strongly Ramanujan graphs from codes, polyphase-sequences, and Combinatorics. In: Proceedings of the International Symposium on Information Theory (ISIT 1997), Ulm, Germany, p. 408 (1997)
32. Janwa, H., Moreno, O.: New Constructions of Ramanujan Graphs and Good Expander Graphs from Codes, Exponential Sums and Sequences. IMA Summer Program on Codes, Systems and Graphical Models (August 1999)
33. Janwa, H., Moreno, O.: Elementary constructions of some Ramanujan graphs. Congressus Numerantium (December 1995)
34. Janwa, H., Moreno, O.: "Coding theoretic constructions of some number theoretic Ramanujan graphs. Congressus Numerantium 130, 63–76 (1998)
35. Janwa, H., Moreno, O.: "Expander Graphs, Ramanujan Graphs, Codes, Exponential Sums, and Sequences (to be submitted)
36. Janwa, H., Moreno, O., Kumar, P.V., Helleseth, T.: Ramanujan graphs from codes over Galois Rings (in preparation)
37. Janwa, H., Rangachari, S.S.: Ramanujan Graphs. Lecture Notes. Preliminary Version (July 1994)
38. Kou, Y., Lin, S., Fossorier, M.P.C.: Low-density parity-check codes based on finite geometries: a rediscovery and new results. IEEE Trans. Inform. Theory 47(7), 2711–2736 (2001)
39. Lin, S., Kou, Y.: A Geometric Approach to the Construction of Low Density Parity Check Codes. In: Presented at the IEEE 29th Communication Theory Workshop, Haynes City, Fla., May 7-10 (2001)
40. Lin, S., Kou, Y., Fossorier, M.: Finite Geometry Low Density Parity Check Codes: Construction, Structure and Decoding. In: Proceedings of the ForneyFest. Kluwer Academic, Boston (2000)
41. Li, W.-C.W.: Character sums and abelian Ramanujan graphs. Journal of Number Theory 41, 199–217 (1992)
42. Lubotzky, A.: Discrete Groups, Expanding Graphs and Invariant Measures. Birkhäuser, Basel (1994)

43. Lubotzky, A., Phillips, R., Sarnak, P.: Ramanujan Graphs. Combinatorica 8, 261–277 (1988)
44. Margulis, G.A.: Explicit group theoretic constructions of combinatorial schemes and their applications for construction of expanders and super-concentrators. Journal of Problems of Information Transformation, 39–46 (1988)
45. Reingold, O., Vadhan, S., Wigderson, A.: Entropy waves, the zig-zag graph product, and new constant-degree expanders. Ann. of Math. 155(1), 157–187 (2002)
46. Richardson, T., Urbanke, R.: The capacity of low-density parity check codes under message-passing decoding. IEEE Trans. Inform. Theory 47(2), 569–618 (2001)
47. Richardson, T.J., Shokrollahi, M.A., Urbanke, R.: Design of capacity-approaching irregular low-density parity-check codes. IEEE Trans. Inform. Theory 47(2), 619–637 (2001)
48. Rosenthal, J., Vontobel, P.O.: Construction of LDPC Codes using Ramanujan Graphs and Ideas from Margulis. In: Allerton Conference (2000)
49. Roth, R.M., Skachek, V.: Improved nearly MDS codes. IEEE Trans. on Inform. Theory 52(8), 3650–3661 (2006)
50. Sarnak, P.: Some Applications of Modular Forms. Cambridge University Press, Cambridge (1990)
51. Sarnak, P.: What is... an Expander. Notices of the AMS 51(7), 762–763 (2004)
52. Shokrollahi, A.: Codes and graphs. In: Reichel, H., Tison, S. (eds.) STACS 2000. LNCS, vol. 1770, pp. 1–12. Springer, Heidelberg (2000)
53. Sipser, M., Spielman, D.A.: Expander codes. Codes and complexity. IEEE Trans. Inform. Theory 42(6, part 1), 1710–1722 (1996)
54. Spielman, D.A.: Linear-time encodable and decodable error-correcting codes. IEEE Trans. on Inform. Theory 42(6), 1710–1722 (1996)
55. Tanner, R.M.: Explicit concentrators from generalized N-gons. SIAM J. of Discrete and Applied Mathematics 5, 287–289 (1984)
56. van Lint, J.H., Wilson, R.M.: A Course in Combinatorics, 2nd edn. Cambridge University Press, Cambridge (2001)
57. Tanner, R.M.: Minimum-distance bounds by graph analysis. IEEE Trans. Inform. Theory 47(2), 808–821 (2001)
58. Zemor, G.: On Expander Codes. IEEE Trans. Inform. Theory. IT-47(2), 835–837 (2001)

Simulation of the Sum-Product Algorithm Using Stratified Sampling*

John Brevik[1], Michael E. O'Sullivan[2], Anya Umlauf[2], and Rich Wolski[3]

[1] California State University, Long Beach
1250 Bellflower Blvd.
Long Beach CA 90840
[2] Department of Mathematics and Statistics
San Diego State University
5500 Campanile Dr.
San Diego, CA, USA, 92182
[3] Department of Computer Science
University of California, Santa Barbara, CA 75275

Abstract. Using stratified sampling a desired confidence level and a specified margin of error can be achieved with smaller sample size than under standard sampling. We apply stratified sampling to the simulation of the sum-product algorithm on a binary low-density parity-check code.

Keywords: Sum-product algorithm, stratified sampling, simulation.

1 Introduction

One of the mysterious aspects of decoding low-density parity-check codes with the sum-product algorithm is the error floor appearing in the performance curve. As is now well known, the performance of the SPA, measured in either bit-error-rate or word-error-rate as a function of the signal-to-noise ratio, tends to have two regions: a "waterfall" portion, where the curve descends ever more steeply until it reaches the second region, the "error floor," where the curve flattens out considerably. The change in descent has been attrributed to "near-codewords" [4], also called trapping sets [5]. The error floor for ensembles of codes using erasure decoding and also maximum a posteriori (MAP) decoding are analyzed with sophisticated probabilistic techniques in [6, 3.24,4.14]. The analysis of the error floor for the SPA appears to be quite challenging. In particular, for individual codes, "the error floor does not concentrate [to the ensemble average] and significant differences can arise among the individual members of an ensemble. It is therefore important to indentify 'good elements' of an ensemble" [6, p.266].

The error floor is important because it limits the utility of an LDPC code. In practice, it may be desirable to have assurance that the (bit) error rate at a particular signal-to-noise ratio is lower than 10^{-10}, but simulation to achieve

* This research was supported by NSF CCF-Theoretical Foundations grants #0635382, #0635389 and #0635391.

M. Bras-Amorós and T. Høholdt (Eds.): AAECC 2009, LNCS 5527, pp. 65–72, 2009.

a reasonable level of confidence at this rate can be expensive. There are some methods to improve code constructions to lower the error floor see *e.g.* [3,1], but there is no clear theory to explain the error floor and no method to compute it for a particular LDPC code. Thus one might hope for improved methods of simulation.

Our approach to this problem is based on the observation that with high probability, a received vector for input in the SPA has a very small number of badly corrupted bits and will decode successfully; to look at it another way, the input vectors that fail to decode are concentrated in a region of relatively low probability. In this article, we summarize how stratified sampling may be used to greatly reduce the sample size required for a desired confidence level and margin of error. We apply stratified sampling to a particular LDPC code—a $(3, 6)$ regular code of length 26—that is small but still interesting. The method reduces the number of samples required by roughly a factor of 70, at SNR 9.0, a margin of error of 20% of the error rate (which is roughly 4×10^{-6}), and 95% confidence level.

2 Stratified Sampling

In this section, we give a brief resumé of the ideas behind *stratified sampling*, which is a method for reducing the variance in a sample (*Cf.* [2, Ch. 8] for a more complete account). We give a rather general treatment, provide some simple examples, and finally interpret the method in the context of block-error rates for LDPC codes.

2.1 Theoretical Overview

Let (Ω, p) be a probability space, and let X be a real-valued random variable on Ω, which we will assume to have finite mean μ and variance σ^2. We are interested in the problem of estimating the expected value $\mu = E(X)$ from a sample $(x_1, x_2 \ldots, x_r)$ of size r taken from X. Provided that X is relatively well behaved and that r is sufficiently large, the sample mean $\bar{X}$ has approximate distribution $N(\mu, \sigma^2/r)$. Again with the same provisions, σ^2 is adequately approximated by the sample variance s^2, so we can obtain a reasonable confidence interval for μ as $(\bar{x} - z^* s/\sqrt{r}, \bar{x} + z^* s/\sqrt{r})$, where $\bar{x}$ is the mean obtained from our particular sample and z^* is an appropriate standard-normal critical value.

Now suppose that Y is another random variable on Ω taking on the finite set $Y_1, Y_2, \ldots, Y_k$ of values; we will refer to the events $Y = Y_k$ as *strata* of Ω. Suppose further that $p_k = p(Y = Y_k)$ is known or can be approximated effectively. This suggests an alternate method for estimating μ from a sample of size r: Take a sample $(x_{j1}, x_{j2} \ldots, x_{jr_j})$ of size r_j from each of the variables $X_j = X|Y_j$, where $r = \sum r_j$, and take the weighted sum $\hat{x} = \sum p_j \bar{x}_j$ of sample means. The sample mean $\bar{X}_j$ has approximate distribution $N(E(X|Y_j), V(X|Y_j)/r_j)$; note that $\sum p_j E(X|Y_j) = \mu$. If we again assume that the j^{th} sample variance s_j^2 adequately approximates the true variance $V(X|Y_j)$, we find that the variance of $\sum p_j \bar{X}_j$ is approximately $\sum p_j^2 s_j^2 / r_j$.

To find the values of r_j, subject to $\sum r_j = r$ that minimize the variance, use Lagrange multipliers: We find that $-\frac{p_j^2 s_j^2}{r_j^2} = \lambda$ for all j, so $r_j = r \cdot \frac{p_j s_j}{\sum p_i s_i}$. The approximate value of the total variance of this estimator is thus

$$\operatorname{Var} \hat{x} \approx \sum \frac{p_j^2 s_j^2 \sum p_i s_i}{r p_j s_j} = \frac{(\sum p_j s_j)^2}{r}. \tag{1}$$

We note that the term "stratified sampling" is sometimes used for the specific case in which the r_i are chosen to be proportional to the p_i. In general, this approach also leads to some reduction in variance, simply because it controls one contributor to sampling variance, namely the variance in the sampling distribution of Y-values. The approach outlined above clearly yields superior reduction in variance, based as it is on optimization.

2.2 An Example

Suppose that we wish to obtain an accurate estimate for the incidence of a certain gene in a population. Suppose further that the population is 70% white and 30% non-white, that we have obtained the approximate values of .01% incidence in the white population and 17% in the non-white population, and that these approximations provide acceptable values for the purposes of estimating sample variances. This gives an approximate value of .05107 for the proportion of the population with this gene, and the proportion for a "raw" sample of size r would thus have variance approximately $\frac{.05107 \cdot (1 - .05107)}{r} \approx \frac{.04846}{r}$.

Now write Y_1 for the event that a member of the population is white and Y_2 for the event that s/he is non-white; then, in the notation established above, $p_1 = .7, s_1 = \sqrt{.0001 \cdot .9999} \approx .01$, so $p_1 s_1 \approx .007$; and $p_2 = .3, s_2 = \sqrt{.17 \cdot .83} \approx .3756$, so $p_2 s_2 \approx .1127$. The calculations above show that the optimal stratified sample of size r takes $r \cdot \frac{.007}{.007 + .1127} \approx .0585$ from the white population and therefore about 94.15% of the total sample from the non-white population. The sample variance in this scheme is $\frac{(.007 + .1127)^2}{r} = \frac{.0143}{r}$, an improvement by a factor of almost 3.4. In practical terms, the standard error (= standard deviation of the sample proportion) is about $\sqrt{.0143} \cdot .04846 \approx .54$ as large using stratified sampling as with the "raw" method. For a given sample size, then, stratification allows us to obtain an estimate with just over half the margin of error; alternatively, it allows us to use a much smaller sample – about .3 as large, since the standard error decreases as the square root of the sample size – to obtain the same level of accuracy.

We can gain even more advantage if the population can be further separated into strata with different characteristics. Suppose that the non-white population breaks down as 25% African-American and 5% Asian-American and that the gene has incidence about 1% in the African-American population and about 97% in the Asian-American population. An analogous calculation to the above

calls for a sample that is about 17.3% white, 65.6% African-American, and 21.1% Asian-American, and the total variance in this case comes out to about $\frac{.00163}{r}$ – a further improvement by a factor of about 9 over the first stratified example, and a further reduction by a factor of almost 3 in standard error. Compared to "raw" sampling, this final stratified sampling scheme reduces the necessary sample size for a given margin of error by a factor of about 29.7.

2.3 Stratified Sampling with LDPC Codes

We now consider a code of length n with code symbols ± 1 (*i.e.*, $\{(-1)^0, (-1)^1\}$), transmitted across a Gaussian memoryless binary symmetric channel. In this setting the probability space Ω consists of all vectors $(\ell_1, \ell_2, \ldots, \ell_n), 0 < \ell_j < 1$, where ℓ_j is the likelihood of the value -1 at the j^{th} bit, inferred by the decoder at the initial step based on the received value at that bit. For the purposes of simulation it is sufficient to restrict to the all-1 codeword; see [6][Sec. 4.3]; by our assumptions, the components of Ω are iid and the distribution on each component can be calculated using normal distributions.

Our random variable X of interest takes on the value 1 if the SPA decoder does not produce the all 1 codeword and 0 otherwise, so $E(X)$ is the *frame-error rate* (FER) of the code.

We can compute the density function for ℓ_i under the assumption that the all-1 codeword was sent.

Let T_1 stand for the input at a bit given that a 1 is transmitted. T_1 is distributed according to the Gaussian $N_1(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(t-1)^2}{2\sigma^2}}$. Similarly, if a -1 is transmitted, the input T_{-1} is distributed according to $N_{-1}(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(t+1)^2}{2\sigma^2}}$. Now, for a given input value t, the likelihood ratio $\dfrac{\ell_t(-1)}{\ell_t(1)}$ is given by $\dfrac{N_{-1}(t)}{N_1(t)} = e^{-\frac{2t}{\sigma^2}}$, and therefore the likelihood $r(t)$ that a -1 was sent is equal to

$$\frac{\ell_t(-1)}{\ell_t(-1) + \ell_t(0)} = \frac{e^{-\frac{2t}{\sigma^2}}}{e^{-\frac{2t}{\sigma^2}} + 1} = \frac{1}{1 + e^{\frac{2t}{\sigma^2}}}.$$

Note that $q = r(t)$ is a strictly **decreasing** function of t, and we can solve for t to obtain $t = \frac{\sigma^2}{2} \ln\left(\frac{1-q}{q}\right)$. Now, under the assumption that a 1 was actually transmitted, for any q in the interval $(0, 1)$, the distribution function $F_Q(q)$ is given by

$$F_Q(q) = P(Q \leq q) = P\left(T_1 \geq \frac{\sigma^2}{2} \ln\left(\frac{1-q}{q}\right)\right)$$

$$F_Q(q) = P\left(\frac{T_1 - 1}{\sigma} \geq \left(\frac{\sigma}{2} \ln\left(\frac{1-q}{q}\right) - \frac{1}{\sigma}\right)\right)$$

$$F_Q(q) = 1 - \left[\frac{1}{2}\left(1 + \operatorname{erf} \frac{\frac{\sigma}{2} \ln\left(\frac{1-q}{q}\right) - \frac{1}{\sigma}}{\sqrt{2}}\right)\right] \tag{2}$$

where erf is the the *error function,'* defined by $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_1^x e^{-u^2}\, du$ so that $\frac{1}{2}(1+\operatorname{erf}(X/\sqrt{2}))$ is the distribution function for the standard normal, any particular value of which can be well approximated.

Now, set some threshold value t_1 above which a likelihood ℓ_j is to be considered "high," and stratify the set of likelihood vectors by the number of high entries contained in each, so that our variable Y takes on the values $0, 1, \ldots, n$ corresponding to the number of high values in a vector. Note that $p_j = p(Y = j)$ is relatively straightforward to calculate from the distribution on ℓ and binomial coefficients. This situation is a prime candidate for stratified sampling, because the strata corresponding to large j have a much higher incidence of decoding failure than those with low j-values.

We then carry out sampling as follows: For a given j, choose randomly the j "high" positions; then sample from the conditional density $\ell > t_1$ in these places and from the conditional density $\ell < t_1$ in the others.

Of course, the threshold t_1 is a parameter in this scheme. More generally, in fact, one can define multiple thresholds $t_1 < t_1 < \cdots < t_m$, define $t_{m+1} = 1$, and stratify Ω by the m-tuple of integers $(y_1, y_2, \ldots, y_m)$ where y_j is the number of entries falling between t_j and t_{j+1}, so that $n - y_1 - \cdots - y_m$ is the number of entries between 0 and t_1.

Because the large sample sizes required to get meaningful estimates for error rates incurs a high cost in terms of time and compute cycles, our purpose for stratified sampling is not to obtain a superior estimate of block-error rate for a given sample size but rather to obtain an equally accurate estimate with a smaller sample size. In the next section, we show for a specific code the amount of sample-size reduction that is possible with stratified sampling.

3 Application

We applied stratified sampling to a $(3, 6)$ regular graph with 26 bit nodes and 13 check nodes. The girth of the graph is 6. We simulated the sum-product algorithm for signal to noise ratios from 5 to 9, over which range the output FER drops from roughly 10^{-2} to 5×10^{-6}. We want to estimate block error rate with a 95% confidence level and within 20% of the estimate $\hat{\theta}$. Let $\tilde{r}$ be the sample size needed for the stratified sampling, and let r be the sample size needed for standard sampling. We are interested in comparing these values.

We chose the threshold t_1 to be 0.5. As described above, the probability space Ω is stratified by the number j of high values in the vector, where j varies from 0 to 26. The probability of the jth stratum is $\binom{26}{j} q^j (1-q)^{26-j}$, where q is the probability, calculated as above, that any given received value is over 0.5. The vast majority of strata have very low probability and may be ignored.

For regular sampling, variance of $\hat{\theta}$ can be calculated with $\operatorname{Var}(\hat{\theta}) = \frac{\sigma^2}{r}$, where σ^2 is estimated by $\hat{\theta} \cdot (1 - \hat{\theta})$. Sample size required for the standard sampling can be calculated with $r = \frac{z_\alpha^2 \sigma^2}{M^2}$, where σ^2 is the variance of $\hat{\theta}$ and M is margin of error.

With stratified sampling, variance of $\hat{\theta}$ is found as in Equation 1 above: $\mathrm{Var}(\hat{\theta}) \approx \frac{(\sum_j p_j s_j)^2}{\tilde{r}}$, where $s_j = \sqrt{\hat{\theta}_j \cdot (1-\hat{\theta}_j)}$. The relative sample sizes for a given margin of error is then given by

$$\frac{r}{\tilde{r}} = \frac{\hat{\theta} \cdot (1-\hat{\theta})}{(\sum_j p_j s_j)^2}.$$

Note that this ratio is independent of the desired margin of error.

In order to use stratified sampling for the problem of estimating FER, we first need to obtain an approximate value for the FER within each stratum in

Table 1. Data for several strata at each SNR

SNR	$\hat{\theta}$	M^2	stratum j	p_j	$\hat{\theta}_j$	s_j	$p_j \times s_j$	r_j
5	9.50e-03	3.61e-06	0	3.68e-01	1.00e-10	1.00e-05	3.68e-06	1
			1	3.75e-01	1.04e-03	3.22e-02	1.21e-02	677
			2	1.84e-01	1.28e-02	1.12e-01	2.06e-02	1,155
			3	5.75e-02	6.19e-02	2.41e-01	1.39e-02	776
			4	1.29e-02	1.70e-01	3.76e-01	4.87e-03	273
			5	2.23e-03	4.05e-01	4.91e-01	1.20e-03	62
			6	3.06e-04	3.33e-01	4.71e-01	1.44e-04	9
Total							5.27e-02	2,953
6	3.36e-03	4.52e-07	0	5.46e-01	1.00e-10	1.00e-05	5.46e-06	2
			1	3.34e-01	3.50e-04	1.87e-02	6.25e-03	1,289
			2	9.84e-02	1.61e-02	1.26e-01	1.24e-02	2,553
			3	1.85e-02	6.67e-02	2.49e-01	4.62e-03	953
			4	2.51e-03	1.44e-01	3.51e-01	8.81e-04	182
			5	2.60e-04	2.46e-01	4.31e-01	1.12e-04	24
Total							2.43e-02	5,003
7	6.57e-04	1.73e-08	0	7.19e-01	1.00e-10	1.00e-05	7.19e-06	12
			1	2.38e-01	1.70e-04	1.30e-02	3.11e-03	5,170
			2	3.80e-02	7.31e-03	8.52e-02	3.24e-03	5,382
			3	3.87e-03	7.27e-02	2.60e-01	1.01e-03	1,674
			4	2.84e-04	1.90e-01	3.92e-01	1.12e-04	186
			5	1.59e-05	1.96e-01	3.97e-01	6.32e-06	11
Total							7.48e-03	12,435
8	8.65e-05	2.99e-10	0	8.55e-01	1.00e-10	1.00e-05	8.55e-06	175
			1	1.34e-01	3.31e-05	5.75e-03	7.73e-04	15,748
			2	1.01e-02	4.48e-03	6.68e-02	6.77e-04	13,806
			3	4.90e-04	6.86e-2	2.53e-01	1.24e-04	2,525
			4	1.70e-05	1.78e-01	3.83e-01	6.51e-06	133
Total							1.59e-03	32,387
9	4.84e-06	9.36e-13	0	9.39e-01	1.00e-10	1.00e-05	9.39e-06	9,610
			1	5.91e-02	7.57e-06	2.75e-03	1.63e-04	166,289
			2	1.79e-03	1.54e-03	3.93e-02	7.01e-05	71,753
			3	3.46e-05	4.36e-02	2.04e-01	7.06e-06	7,220
			4	4.81e-07	2.63e-01	4.40e-01	2.12e-07	217
Total							2.49e-04	255,089

Table 2. Summarized data for each SNR

SNR	$\hat{\theta}$	$(1-\hat{\theta})\cdot\hat{\theta}$	$(\sum_j p_j s_j)^2$	ratio	r	$\tilde{r}$
5	9.50e-03	9.41e-03	2.77e-03	3	10,011	2,953
6	3.36e-03	3.35e-03	5.89e-04	6	28,466	5,003
7	6.57e-04	6.57e-04	5.59e-05	12	146,065	12,435
8	8.65e-05	8.65e-05	2.52e-06	34	1,109,868	32,387
9	4.84e-06	4.84e-06	6.21e-08	78	19,855,359	255,089

order to set the sample sizes $r_j = (\frac{p_j s_j}{\sum_j p_j s_j})\tilde{r} = p_j s_j \frac{Z_\alpha^2}{M^2}(\sum_j p_j s_j)$. We do this by taking relatively small samples; note that any sampling error has only the "second-order" effect of changing the estimated standard error s_j^2, so we expect the final outcome to be fairly robust to this type of error.

Given these estimates, select r_j vectors from each stratum j at random. This can be achieved in practice by first randomly choosing j bit positions for "high" values and then sampling "high" values and "low" values by conditioning the distribution according to Equation 2.

The estimated value for the frame-error rate is $\hat{\theta} = \sum_j p_j \hat{\theta}_j$. For all included strata, sample sizes m_j are shown in Table 1.

Table 2 shows how much sample-size reduction can be achieved with stratified sampling, together with calculated sample sizes needed for standard and stratified sampling for a margin of error approximately 20% of the FER. At SNR 5 the reduction ratio is approximately 3; however, as SNR increases this ratio also increases. For example, standard sampling for SNR 9 requires almost 20 million units, whereas stratified sampling needs only 255 thousand, reducing sample size by a factor of approximately 77.

4 Conclusions and Future Work

The results shown here indicate that stratified sampling is an effective strategy for reducing sample size necessary to estimate decoder performance on an LDPC code at high SNR, within a specified margin of error. At this stage, there are a number of clear paths for future examination.

First, a length-26 code is far to short to have any real-world value, so it is necessary to validate these results with longer codes. We have begun testing codes of length 282 using massively parallel batch jobs on the Condor pool [1] [7],

[1] Condor provides (among many features) the ability to "harvest" computing cycles from idle computers that are connected to the Internet. For these experiments, we use a Condor pool located in the Computer Science Department at the University of California, Santa Barbara consisting of approximately 100 machines. In addition, the Condor Project at the University of Wisconsin allows UCSB's Condor pool to "offload" work that exceeds the available idle machine capacity; thus each experiment had available to it up to 300 machines for the purpose of computing each Monte Carlo simulation. The ability to harvest unused cycles and load-share between institutions made the investigation feasible in a short span of time.

and it is already evident from preliminary results that it will be necessary to adjust the cutoff value $q = 0.5$; while we have achieved a speedup by a factor of 5 or so using such adjustments, it is not clear that the straightforward approach used for the length-26 code will achieve such dramatic results.

However, there are a number of ways in which the stratification scheme can be made more sophisticated. As discussed earlier, one can use multiple thresholds, $t_1 < t_2 < \cdots < t_m$. For example, with $m = 2$, we could associate to each vector of likelihoods an ordered pair (a, b), where a stands for the number of values between, say, 0.5 and 0.9 and b stands for the number of values greater than 0.9. Probabilities of the various strata can still be calculated with binomial coefficients, and sampling can still be done with linear conditioning on random-number generators. Another possibility is to stratify based on the *proximity* of high-valued bits on the associated graph; intuitively, if high-valued bits are close together (*e.g.* sharing a check), the bits' inaccurate values tends to reinforce one another and affect nearby bit estimates in the SPA.

We note that our method can be adapted to bit-error rate (BER), which in some applications is a more suitable measure for decoding error than FER. In fact, a stratified method would likely work even better here, since the low-probability badly behaved strata will likely have much more variance in the number of errors per word than the high-probability strata and will therefore contribute proportionally more to the variance of the BER estimator.

As was kindly pointed out by one of the referees, the general framework of stratified sampling may well apply to a wide variety of communications systems, which typically lack precise analytical means for error analysis and therefore rely on sampling. While our experience is mainly in LDPC codes, exploring this broader applicability may lead to better and more general results.

References

1. Xu, J., Chen, L., Djurdjevic, I., Lin, S., Abdel-Ghaffar, K.: Construction of regular and irregular LDPC codes: geometry decomposition and masking. IEEE Trans. Inform. Theory 53, 121–134 (2007)
2. Ross, S.M.: Simulation, 4th edn. Elsevier Academic Press, Amsterdam (2006)
3. Tian, T., Jones, C., Villasenor, J.D., Wesel, R.D.: Construction of irregular LDPC codes with low error floors. In: IEEE International Conference on Communications, vol. 5, pp. 3125–3129 (2003)
4. MacKay, D.J.C., Postol, M.J.: Weaknesses of Margulis and Ramanujan–Margulis low-lensity parity-check codes. In: Proc. of MFCSIT 2002, Galway. Electronic Notes in Theoretical Computer Science, vol. 74. Elsevier, Amsterdam (2003)
5. Richardson, T.: Error floors of LDPC codes. In: Proc. of 41st Allerton Conf., Monticello, IL (September 2003)
6. Richardson, T., Urbanke, R.: Modern Coding Theory. Cambridge University Press, Cambridge (2008)
7. Tannenbaum, T., Litzkow, M.: The Condor Distributed Processing System. Dr. Dobbs Journal (February 1995)

A Systems Theory Approach to Periodically Time-Varying Convolutional Codes by Means of Their Invariant Equivalent*

Joan-Josep Climent[1], Victoria Herranz[2], Carmen Perea[2], and Virtudes Tomás[1,**]

[1] Departament de Ciència de la Computació i Intel·ligència Artificial
Universitat d'Alacant
Campus de Sant Vicent del Raspeig, Ap. Correus 99, E-03080 Alacant, Spain
[2] Centro de Investigación Operativa
Departamento de Estadística, Matemáticas e Informática
Universidad Miguel Hernández
Avenida de la Universidad s/n, E-03202 Elche - Alicante, Spain

Abstract. In this paper we construct (n, k, δ) time-variant convolutional codes of period τ. We use the systems theory to represent our codes by the input-state-output representation instead of using the generator matrix. The obtained code is controllable and observable. This construction generalizes the one proposed by Ogasahara, Kobayashi, and Hirasawa (2007). We also develop and study the properties of the time-invariant equivalent convolutional code and we show a lower bound for the free distance in the particular case of MDS block codes.

Keywords: Time-varying convolutional codes, controllability, observability, free distance.

1 Introduction

Convolutional codes [6, 10, 14] are an specific class of error correcting codes that can be represented as time-invariant discrete linear systems over a finite field [19]. They are used in phone data transmission, radio or mobile communication systems and in image transmissions from satellites [11, 13]. Convolutional coding is the main error correcting technique in data transmission applications due to its easy implementation and nice performance in random error channels [12, 23].

While it is usual for block codes to have the minimum distance guaranteed, for convolutional codes it is common to find out the minimum free distance by a

* This work was partially supported by Spanish grants MTM2008-06674-C02-01 and MTM2008-06674-C02-02.

** The work of this author was also supported by a grant of the Vicerectorat d'Investigació, Desenvolupament i Innovació of the Universitat d'Alacant for PhD students.

M. Bras-Amorós and T. Høholdt (Eds.): AAECC 2009, LNCS 5527, pp. 73–82, 2009.

code search. Since the required computation in a code search grows exponentially with the number of delay elements of a code, this code search of the minimum free distance can become difficult. As well, the computational requirements of the decoding increase with the amount of delay elements δ. In the construction proposed in this paper we will see that δ does not increase, what can reduce decoding computation [17].

The rest of the paper is organized as follows. In Section 2 we provide all the necessary concepts about convolutional codes to follow the rest of the paper. In Section 3 we give initial knowledge about periodically time variant convolutional codes and we construct the time-invariant equivalent one. In Subsection 3.1 we develop the minimality conditions for the constructed code, and in Subsection 3.2 we study sufficient conditions for the time-invariant code to be controllable and observable when this is formed by $(n, 1, 1)$ or $(n, n-1, 1)$ codes. In Section 4 we give a lower bound on the free distance. Finally, we provide some conclusions and future research lines in Section 5.

2 Preliminaries

Let $\mathbb{F}$ be a finite field. A rate k/n convolutional code $\mathcal{C}$ is a submodule of $\mathbb{F}^n[z]$ that can be described (see [21, 25]) as

$$\mathcal{C} = \left\{ \boldsymbol{v}(z) \in \mathbb{F}^n[z] \mid \boldsymbol{v}(z) = G(z)\boldsymbol{u}(z) \text{ with } \boldsymbol{u}(z) \in \mathbb{F}^k[z] \right\}$$

where $\boldsymbol{u}(z)$ is the **information vector** or **information word**, $\boldsymbol{v}(z)$ is the **code vector** or **code word** and $G(z)$ is an $n \times k$ polynomial matrix with rank k called **generator** or **encoding matrix** of $\mathcal{C}$.

The **complexity** δ of $\mathcal{C}$ is the maximum of the degrees of the determinants of the $k \times k$ submatrices of any generator matrix of $\mathcal{C}$. We call $\mathcal{C}$ an (n, k, δ) convolutional code (see [14]).

We can describe an (n, k, δ) convolutional code $\mathcal{C}$ by means of the system

$$\left.\begin{aligned} \boldsymbol{x}_{t+1} &= A\boldsymbol{x}_t + B\boldsymbol{u}_t \\ \boldsymbol{y}_t &= C\boldsymbol{x}_t + D\boldsymbol{u}_t \end{aligned}\right\}, \quad t = 0, 1, 2, \ldots; \quad \boldsymbol{x}_0 = \boldsymbol{0} \tag{1}$$

where $A \in \mathbb{F}^{\delta \times \delta}$, $B \in \mathbb{F}^{\delta \times k}$, $C \in \mathbb{F}^{(n-k) \times \delta}$, $D \in \mathbb{F}^{(n-k) \times k}$ and $\boldsymbol{v}_t = \begin{bmatrix} \boldsymbol{y}_t \\ \boldsymbol{u}_t \end{bmatrix}$.

The four matrices (A, B, C, D) are called the **input-state-output representation** of the code $\mathcal{C}$. This representation was introduced in [21] and has been used in the last years to analyze and construct convolutional codes [1, 4, 5, 9, 18, 19, 21, 25].

For each positive integer j let us define the matrices

$$\Phi_j(A, B) = \begin{bmatrix} B & AB & \cdots & A^{j-2}B & A^{j-1}B \end{bmatrix} \quad \text{and} \quad \Omega_j(A, C) = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{j-2} \\ CA^{j-1} \end{bmatrix}.$$

Definition 1. *Let A and B be matrices of sizes $\delta \times \delta$ and $\delta \times k$, respectively. The pair (A, B) is* ***controllable*** *if* $\operatorname{rank}(\Phi_\delta(A, B)) = \delta$. *If (A, B) is a controllable pair, then the smallest integer κ such that* $\operatorname{rank}(\Phi_\kappa(A, B)) = \delta$ *is the* ***controllability index*** *of (A, B).*

Definition 2. *Let A and C be matrices of sizes $\delta \times \delta$ and $(n-k) \times \delta$, respectively. The pair (A, C) is* ***observable*** *if* $\operatorname{rank}(\Omega_\delta(A, C)) = \delta$. *If (A, C) is an observable pair, then the smallest integer ν such that* $\operatorname{rank}(\Omega_\nu(A, C)) = \delta$ *is the* ***observability index*** *of (A, C).*

An important distance measures of a convolutional code is the **free distance**

$$d_{free}(\mathcal{C}) = \min_{\boldsymbol{u}_0 \neq 0} \left\{ \sum_{t=0}^{\infty} wt(\boldsymbol{u}_t) + \sum_{t=0}^{\infty} wt(\boldsymbol{y}_t) \right\}$$

where $wt(\cdot)$ denotes the Hamming weight of a vector.

Rosenthal and Smarandache [20] gave a generalization of the Singleton bound for block codes.

Theorem 1 (Theorem 2.2 of [20]). *If $\mathcal{C}$ is a (n, k, δ)-code over any field $\mathbb{F}$, then*

$$d_{free}(\mathcal{C}) \leq (n-k) \left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1 \right) + \delta + 1. \tag{2}$$

This bound is known as the **generalized Singleton bound**. Analogously to block codes, we say that a (n, k, δ)-code is *maximum distance separable (MDS)* if the free distance attains the generalized Singleton bound.

3 Periodically Time-Variant Convolutional Codes

In this section we define periodically time-varying convolutional codes and explain the concrete characteristics of our construction.

Let us assume that the matrices A_t, B_t, C_t and D_t at time t are of sizes $\delta \times \delta$, $\delta \times k$, $(n-k) \times \delta$ and $(n-k) \times k$, respectively. A time-variant convolutional code can be defined by means of the system

$$\left.\begin{aligned} \boldsymbol{x}_{t+1} &= A_t \boldsymbol{x}_t + B_t \boldsymbol{u}_t \\ \boldsymbol{y}_t &= C_t \boldsymbol{x}_t + D_t \boldsymbol{u}_t \end{aligned}\right\}, \quad t = 0, 1, 2, \ldots \quad \boldsymbol{x}_0 = \boldsymbol{0} \tag{3}$$

The defined time-variant convolutional code is a dynamical system which can have a **minimal representation**, that is, a representation where the matrices have the smallest size possible. It is known that the realization (A, B, C, D) of a linear system is minimal if and only if (A, B) is a controllable pair and (A, C) is an observable pair [2]. In Subsection 3.1 and 3.2 we will study the conditions for our code to have a minimal representation.

If the matrices change periodically with periods τ_A, τ_B, τ_C and τ_D respectively, (that is, $A_{\tau_A+t} = A_t$, $B_{\tau_B+t} = B_t$, $C_{\tau_C+t} = C_t$ and $D_{\tau_D+t} = D_t$ for

all t) then we have a periodically time-varying convolutional code of period $\tau = \operatorname{lcm}(\tau_A, \tau_B, \tau_C, \tau_D)$. Any periodic time-varying convolutional code is equivalent to an invariant one [3, 15, 16]. Relating every state and every output to previous states, we can always rewrite any block of τ iterations at a given time j as

$$\begin{aligned}
\boldsymbol{x}_{\tau(j+1)} &= \mathbb{A}_{\tau-1,0}\boldsymbol{x}_{\tau j} \\
&+ \begin{bmatrix} \mathbb{A}_{\tau-1,1}B_0 & \mathbb{A}_{\tau-1,2}B_1 & \dots & \mathbb{A}_{\tau-1,\tau-1}B_{\tau-2} & B_{\tau-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_{\tau j} \\ \boldsymbol{u}_{\tau j+1} \\ \vdots \\ \boldsymbol{u}_{\tau j+\tau-1} \end{bmatrix}, \\
\begin{bmatrix} \boldsymbol{y}_{\tau j} \\ \boldsymbol{y}_{\tau j+1} \\ \vdots \\ \boldsymbol{y}_{\tau j+\tau-1} \end{bmatrix} &= \begin{bmatrix} C_0 \\ C_1\mathbb{A}_{0,0} \\ C_2\mathbb{A}_{1,0} \\ \vdots \\ C_{\tau-1}\mathbb{A}_{\tau-2,0} \end{bmatrix} \boldsymbol{x}_{\tau j} \\
&+ \begin{bmatrix} D_0 & O & \dots & O & O \\ C_1B_0 & D_1 & \dots & O & O \\ C_2\mathbb{A}_{1,1}B_0 & C_2B_1 & \dots & O & O \\ \vdots & \vdots & & \vdots & \vdots \\ C_{\tau-2}\mathbb{A}_{\tau-3,1}B_0 & C_{\tau-2}\mathbb{A}_{\tau-3,2}B_1 & \dots & D_{\tau-2} & O \\ C_{\tau-1}\mathbb{A}_{\tau-2,1}B_0 & C_{\tau-1}\mathbb{A}_{\tau-2,2}B_1 & \dots & C_{\tau-1}B_{\tau-2} & D_{\tau-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_{\tau j} \\ \boldsymbol{u}_{\tau j+1} \\ \vdots \\ \boldsymbol{u}_{\tau j+\tau-1} \end{bmatrix}
\end{aligned}$$

where

$$\mathbb{A}_{i,j} = \begin{cases} A_iA_{i-1}\cdots A_{j+1}A_j, & \text{if } i \neq j, \\ A_i, & \text{if } i = j. \end{cases}$$

This system can be written as

$$\left.\begin{aligned} X_{j+1} &= \mathfrak{A}X_j + \mathfrak{B}U_j \\ Y_j &= \mathfrak{C}X_j + \mathfrak{D}U_j \end{aligned}\right\} \tag{4}$$

where

$$\mathfrak{A} = \mathbb{A}_{\tau-1,0}, \quad \mathfrak{B} = \begin{bmatrix} \mathbb{A}_{\tau-1,1}B_0 & \mathbb{A}_{\tau-1,2}B_1 & \dots & \mathbb{A}_{\tau-1,\tau-1}B_{\tau-2} & B_{\tau-1} \end{bmatrix},$$

$$\mathfrak{C} = \begin{bmatrix} C_0 \\ C_1\mathbb{A}_{0,0} \\ C_2\mathbb{A}_{1,0} \\ \vdots \\ C_{\tau-1}\mathbb{A}_{\tau-2,0} \end{bmatrix}, \quad \mathfrak{D} = \begin{bmatrix} D_0 & O & \dots & O & O \\ C_1B_0 & D_1 & \dots & O & O \\ C_2\mathbb{A}_{1,1}B_0 & C_2B_1 & \dots & O & O \\ \vdots & \vdots & & \vdots & \vdots \\ C_{\tau-2}\mathbb{A}_{\tau-3,1}B_0 & C_{\tau-2}\mathbb{A}_{\tau-3,2}B_1 & \dots & D_{\tau-2} & O \\ C_{\tau-1}\mathbb{A}_{\tau-2,1}B_0 & C_{\tau-1}\mathbb{A}_{\tau-2,2}B_1 & \dots & C_{\tau-1}B_{\tau-2} & D_{\tau-1} \end{bmatrix},$$

$$X_j = \boldsymbol{x}_{\tau j}, \quad Y_j = \begin{bmatrix} \boldsymbol{y}_{\tau j} \\ \boldsymbol{y}_{\tau j+1} \\ \vdots \\ \boldsymbol{y}_{\tau j+\tau-1} \end{bmatrix} \quad \text{and} \quad U_j = \begin{bmatrix} \boldsymbol{u}_{\tau j} \\ \boldsymbol{u}_{\tau j+1} \\ \vdots \\ \boldsymbol{u}_{\tau j+\tau-1} \end{bmatrix}.$$

System (4) is the time-invariant convolutional code equivalent to the periodic time-varying system (3). Our particular construction replaces matrices A_t and D_t by fixed matrices A and D, respectively, for all t. Then, expression (3) turns into

$$\left.\begin{aligned} \boldsymbol{x}_{t+1} &= A\boldsymbol{x}_t + B_t\boldsymbol{u}_t \\ \boldsymbol{y}_t &= C_t\boldsymbol{x}_t + D\boldsymbol{u}_t \end{aligned}\right\}, \quad t = 0, 1, 2, \ldots, \tag{5}$$

and matrices $\mathfrak{A}$, $\mathfrak{B}$, $\mathfrak{C}$ and $\mathfrak{D}$ of system (4) become

$$\mathfrak{A} = A^{\tau}, \quad \mathfrak{B} = \begin{bmatrix} A^{\tau-1}B_0 & A^{\tau-2}B_1 & \ldots & AB_{\tau-2} & B_{\tau-1} \end{bmatrix},$$

$$\mathfrak{C} = \begin{bmatrix} C_0 \\ C_1A \\ \vdots \\ C_{\tau-1}A^{\tau-1} \end{bmatrix}, \quad \mathfrak{D} = \begin{bmatrix} D & O & \ldots & O & O \\ C_1B_0 & D & \ldots & O & O \\ C_2AB_0 & C_2B_1 & \ldots & O & O \\ \vdots & \vdots & & \vdots & \vdots \\ C_{\tau-2}A^{\tau-3}B_0 & C_{\tau-2}A^{\tau-4}B_1 & \ldots & D & O \\ C_{\tau-1}A^{\tau-2}B_0 & C_{\tau-1}A^{\tau-3}B_1 & \ldots & C_{\tau-1}B_{\tau-2} & D \end{bmatrix}.$$

3.1 Minimality Conditions

In this subsection we study conditions for the controllability and observability of the new equivalent time-invariant convolutional code of our construction.

Theorem 2. *If $\tau k \geq \delta$ and the matrices B_j, for $j = 0, 1, \ldots, \tau - 1$, are such that $B_j = A^{-(\tau-j-1)}E_j$, being $E_j \in \mathbb{F}^{\delta\times k}$ with $E = \begin{bmatrix} E_0 & E_1 & \ldots & E_{\tau-1} \end{bmatrix}$ a full rank matrix, then the system defined by expression (4) is controllable.*

Proof. According to the form of the controllable matrix in (5) we have that

$$\Phi_\delta(\mathfrak{A}, \mathfrak{B}) = \begin{bmatrix} \mathfrak{B} & \mathfrak{A}\mathfrak{B} & \ldots & \mathfrak{A}^{\delta-1}\mathfrak{B} \end{bmatrix} = \begin{bmatrix} E & A^{\tau}E & \ldots & A^{(\delta-1)\tau}E \end{bmatrix}$$

which is clearly a full rank matrix. So, the system (4) is controllable. □

Similarly, we have the following result for the observability. Here we denote by M^T the transpose matrix of M.

Theorem 3. *If $\tau(n-k) \geq \delta$ and the matrices C_j, for $j = 0, 1, \ldots, \tau - 1$, are such that $C_j = F_jA^{-j}$, being $F_j \in \mathbb{F}^{(n-k)\times\delta}$ with $F = \begin{bmatrix} F_0^T & F_1^T & \ldots & F_{\tau-1}^T \end{bmatrix}^T$ a full rank matrix, then the system defined by expression (4) is observable.*

Note that if we choose B_j and C_j as in Theorems 2 and 3, then we ensure the minimality of the new invariant system.

Moreover, taking $A = I_\delta$, E as the parity check matrix of a block code, and F such that F_j is the submatrix of $\mathfrak{I} = \begin{bmatrix} I & I & I & \ldots \end{bmatrix}^T$ formed by the rows $j(n-k)+l$ for $l = 1, 2, \ldots, n-k$, then we obtain the periodically time-variant convolutional code proposed in [17]. So, our construction generalizes that case.

3.2 Minimality for Generic Periodic Time-Variant Convolutional Codes

Let us now return to the general case given by expression (3). Another representation of the time-invariant equivalent code is the following state-space form (see [7]):

$$\left.\begin{aligned}\mathcal{R}(\lambda)\widetilde{\boldsymbol{x}}_t(h) &= \mathcal{A}\widetilde{\boldsymbol{x}}_t(h) + \mathcal{B}\boldsymbol{u}_t(h)\\ \boldsymbol{y}_t(h) &= \mathcal{C}\widetilde{\boldsymbol{x}}_t(h) + \mathcal{D}\boldsymbol{u}_t(h)\end{aligned}\right\}$$

where

$$\boldsymbol{u}_t(h) = \begin{bmatrix}\boldsymbol{u}_{t+h\tau}\\ \boldsymbol{u}_{t+h\tau+1}\\ \vdots\\ \boldsymbol{u}_{t+h\tau+\tau-1}\end{bmatrix}, \quad \boldsymbol{y}_t(h) = \begin{bmatrix}\boldsymbol{y}_{t+h\tau}\\ \boldsymbol{y}_{t+h\tau+1}\\ \vdots\\ \boldsymbol{y}_{t+h\tau+\tau-1}\end{bmatrix}, \quad \widetilde{\boldsymbol{x}}_t(h) = \begin{bmatrix}\boldsymbol{x}_{t+h\tau}\\ \boldsymbol{x}_{t+h\tau+1}\\ \vdots\\ \boldsymbol{x}_{t+h\tau+\tau-1}\end{bmatrix}$$

and $\mathcal{R}(\lambda) = \begin{bmatrix} O & I_{(\tau-1)\delta} \\ \lambda I_\delta & O \end{bmatrix}$ where λ denotes the one-step-forward time operator in the variable h and O denotes the null matrix of the appropriate size. The matrices $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{D}$ are now block-diagonal matrices defined as

$$\mathcal{A} = \operatorname{diag}(A_0, A_1, \ldots, A_{\tau-1}), \quad \mathcal{B} = \operatorname{diag}(B_0, B_1, \ldots, B_{\tau-1}),$$
$$\mathcal{C} = \operatorname{diag}(C_0, C_1, \ldots, C_{\tau-1}) \quad \text{and} \quad \mathcal{D} = \operatorname{diag}(D_0, D_1, \ldots, D_{\tau-1}).$$

If we define $\mathcal{P}^i(\lambda) = \begin{bmatrix}\mathcal{A} - \mathcal{R}(\lambda) & \mathcal{B}\end{bmatrix}$ and $\mathcal{P}^0(\lambda) = \begin{bmatrix}\mathcal{A} - \mathcal{R}(\lambda)\\ \mathcal{C}\end{bmatrix}$, then the following theorem, which will be useful for the proofs of further results, holds also for finite fields.

Theorem 4 ([7]). *The periodic system (3) is controllable (respectively, observable) if and only if the matrix $\mathcal{P}^i(\lambda)$ (respectively, $\mathcal{P}^0(\lambda)$) has full row (respectively, column) rank for all $\lambda \in \mathbb{C}$.*

For the case where the subsystems forming the periodically time-varying convolutional code are $(n, 1, 1)$-codes or $(n, n-1, 1)$-codes we have the following result.

Theorem 5. *If the periodically time-varying convolutional code is generated by τ controllable and observable $(n, 1, 1)$-codes (or $(n, n-1, 1)$-codes), then the periodically time-varying convolutional code obtained is as well controllable and observable.*

Proof. We will develop the proof for the case $(n, 1, 1)$. In this case

$$A_t = [a_t], \quad B_t = [b_t], \quad C_t = \begin{bmatrix}c_{t,0}\\ c_{t,1}\\ \vdots\\ c_{t,n-2}\end{bmatrix}, \quad \text{and} \quad D_t = \begin{bmatrix}d_{t,0}\\ d_{t,1}\\ \vdots\\ d_{t,n-2}\end{bmatrix}.$$

Following the previous theorem, our periodic system is controllable if and only if the matrix

$$\mathcal{P}^i(\lambda) = \begin{bmatrix} \mathcal{A} - \mathcal{R}(\lambda) & \mathcal{B} \end{bmatrix} = \begin{bmatrix} a_0 & -1 & 0 & \dots & 0 & 0 & b_0 & 0 & \dots & 0 \\ 0 & a_1 & -1 & \dots & 0 & 0 & 0 & b_1 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & 0 & -1 & 0 & 0 & \dots & 0 \\ -\lambda & 0 & 0 & \dots & 0 & a_{\tau-1} & 0 & 0 & \dots & b_{\tau-1} \end{bmatrix}$$

has full row rank. To ensure that this matrix has full rank we will check that $b_i \neq 0$ for $i = 0, 1, \dots, \tau - 1$.

For every subsystem we have that controllability holds. Then by Popov-Belevitch-Hautus test (see [8]) it holds that $\text{rank}(\begin{bmatrix} zI - a_i & b_i \end{bmatrix}) = 1$ for every eigenvalue z of $[a_i]$ and $i = 0, 1, \dots, \tau - 1$. Since the only eigenvalue of $[a_i]$ is $z = a_i$ then it must hold that $b_i \neq 0$. Thus the matrix $\mathcal{P}^i(\lambda)$ has full rank and the periodic time-varying convolutional code is controllable.

On the other hand, by the previous result, our periodic system is observable if and only if the matrix

$$\mathcal{P}^0(\lambda) = \begin{bmatrix} \mathcal{A} - \mathcal{R}(\lambda) \\ \mathcal{C} \end{bmatrix} = \begin{bmatrix} a_0 & -1 & 0 & \dots & 0 & 0 \\ 0 & a_1 & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & -1 \\ -\lambda & 0 & 0 & \dots & 0 & a_{\tau-1} \\ c_{0,0} & 0 & 0 & \dots & 0 & 0 \\ c_{0,1} & 0 & 0 & & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ c_{0,n-2} & 0 & 0 & \dots & 0 & 0 \\ 0 & c_{1,0} & 0 & \dots & 0 & 0 \\ 0 & c_{1,1} & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & c_{1,n-2} & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & c_{\tau-1,0} \\ 0 & 0 & 0 & \dots & 0 & c_{\tau-1,1} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & c_{\tau-1,n-2} \end{bmatrix}$$

has full column rank. To ensure that this matrix has full rank we will check that for each $i = 0, 1, \dots, \tau - 1$ there exist at least one j such that $c_{i,j} \neq 0$.

For every subsystem we assumed that observability holds. Again by Popov-Belevitch-Hautus test it holds that $\text{rank}\left(\begin{bmatrix} zI - a_i \\ c_{i,0} \\ c_{i,1} \\ \vdots \\ c_{1,n-2} \end{bmatrix}\right) = 1$ for every eigenvalue

z of $[a_i]$ and $i = 0, 1, \ldots, \tau - 1$. Then it must exist at least one j such that $c_{i,j} \neq 0$. Thus the matrix $\mathcal{P}^0(\lambda)$ has full rank and the periodic time-varying convolutional code is observable.

The proof for the case $(n-1, n, 1)$ is analogous. □

4 Free Distance

In this section we obtain a lower bound on the free distance of the periodic time-varying code by means of the free distance of the invariant equivalent associated, due to the fact that both are the same (see [24]).

Theorem 6 (Theorem 3.1 of [22]). *Let $\mathcal{C}$ be an observable, rate k/n, degree δ, convolutional code defined through the matrices A, B, C and D. Let ν be the observability index of the pair (A, C) and suppose that there exists $d \in \mathbb{Z}^+$ such that $\Phi_{d\nu}(A, B)$ forms the parity-check matrix of a block code of distance d. Then the free distance of $\mathcal{C}$ is greater than or equal to d.*

Now we have the following result.

Theorem 7. *If the matrix $\Phi_{d\nu}(\mathfrak{A}, \mathfrak{B}) = \left[\mathfrak{B} \;\; \mathfrak{AB} \;\; \ldots \;\; \mathfrak{A}^{d\nu-1}\mathfrak{B}\right]$ represents the parity-check matrix of an MDS block code of distance d, then $d_{free} \geq \delta + 1$.*

Proof. Let $P = \left[A^{\tau-1}B_0 \;\; A^{\tau-2}B_1 \;\; \ldots \;\; B_{\tau-1}\right]$. The matrix

$$\left[\mathfrak{B} \;\; \mathfrak{AB} \;\; \ldots \;\; \mathfrak{A}^{d\nu-1}\mathfrak{B}\right] = \left[P \;\; A^{\tau}P \;\; \ldots \;\; A^{(d\nu-1)\tau}P\right]$$

has size $\delta \times \tau k d\nu$. Since the parity-check matrix of a block code is a matrix of size $(N-K)\times N$, we have that the parameters of the block code are $[\tau k d\nu, \tau k d\nu - \delta, d]$. The block code is MDS so the distance d achieves the Singleton bound $N-K+1$. Then we have

$$d_{free}(\mathcal{C}) \geq d = \tau k d\nu - \tau k d\nu + \delta + 1 = \delta + 1.$$ □

Note that if $d_{free}(\mathcal{C}) = \delta + 1$, then $\mathcal{C}$ cannot be an MDS convolutional code since, in order to attain the generalized Singleton bound, $n = k$ should hold.

5 Conclusion

In this paper we propose a new method for constructing periodically time-variant convolutional codes. We study the minimality conditions of the new system and state some results about the dependence between the controllability and observability of the subsystems and the minimality of the time-invariant equivalent one. We also study the general case for particular forming subsystems. A lower bound on the free distance is as well given using the fact that both codes have the same d_{free}.

As it is shown in this construction, the number of delay elements remains as δ. This fact makes that the decoding complexity of the new time-varying code

does not increase respect to the complexity of the subsystems. This gives us a lower complexity of the arithmetic circuits when implementing the model.

As future research lines we will study the properties of the time-invariant convolutional code depending on the properties of the subsystems forming the periodically time-variant convolutional one; we will attempt to construct MDS convolutional codes and maximum distance profile (MDP) convolutional codes, which are an optimum subclass.

References

[1] Allen, B.M.: Linear Systems Analysis and Decoding of Convolutional Codes. Ph.D thesis, Department of Mathematics, University of Notre Dame, Indiana, USA (June 1999)

[2] Antsaklis, P.J., Michel, A.N.: Linear Systems. McGraw-Hill, New York (1997)

[3] Balakirsky, V.B.: A necessary and sufficient condition for time-variant convolutional encoders to be noncatastrophic. In: Cohen, G.D., Litsyn, S., Lobstein, A., Zémor, G. (eds.) Algebraic Coding 1993. LNCS, vol. 781, pp. 1–10. Springer, Heidelberg (1994)

[4] Climent, J.J., Herranz, V., Perea, C.: A first approximation of concatenated convolutional codes from linear systems theory viewpoint. Linear Algebra and its Applications 425, 673–699 (2007)

[5] Climent, J.J., Herranz, V., Perea, C.: Linear system modelization of concatenated block and convolutional codes. Linear Algebra and its Applications 429, 1191–1212 (2008)

[6] Dholakia, A.: Introduction to Convolutional Codes with Applications. Kluwer Academic Publishers, Boston (1994)

[7] Grasselli, O.M., Longhi, S.: Finite zero structure of linear periodic discrete-time systems. International Journal of Systems Science 22(10), 1785–1806 (1991)

[8] Hautus, M.L.J.: Controllability and observability condition for linear autonomous systems. Proceedings of Nederlandse Akademie voor Wetenschappen (Series A) 72, 443–448 (1969)

[9] Hutchinson, R., Rosenthal, J., Smarandache, R.: Convolutional codes with maximum distance profile. Systems & Control Letters 54(1), 53–63 (2005)

[10] Johannesson, R., Zigangirov, K.S.: Fundamentals of Convolutional Coding. IEEE Press, New York (1999)

[11] Justesen, J.: New convolutional code constructions and a class of asymptotically good time-varying codes. IEEE Transactions on Information Theory 19(2), 220–225 (1973)

[12] Levy, Y., Costello Jr., D.J.: An algebraic approach to constructing convolutional codes from quasicyclic codes. DIMACS Series in Discrete Mathematics and Theoretical Computer Science 14, 189–198 (1993)

[13] Massey, J.L., Costello, D.J., Justesen, J.: Polynomial weights and code constructions. IEEE Transactions on Information Theory 19(1), 101–110 (1973)

[14] McEliece, R.J.: The algebraic theory of convolutional codes. In: Pless, V.S., Huffman, W.C. (eds.) Handbook of Coding Theory, pp. 1065–1138. Elsevier, North-Holland, Amsterdam (1998)

[15] Mooser, M.: Some periodic convolutional codes better than any fixed code. IEEE Transactions on Information Theory 29(5), 750–751 (1983)

[16] O'Donoghue, C., Burkley, C.: Catastrophicity test for time-varying convolutional encoders. In: Walker, M. (ed.) Cryptography and Coding 1999. LNCS, vol. 1746, pp. 153–162. Springer, Heidelberg (1999)
[17] Ogasahara, N., Kobayashi, M., Hirasawa, S.: The construction of periodically time-variant convolutional codes using binary linear block codes. Electronics and Communications in Japan, Part 3 90(9), 31–40 (2007)
[18] Rosenthal, J.: An algebraic decoding algorithm for convolutional codes. Progress in Systems and Control Theory 25, 343–360 (1999)
[19] Rosenthal, J.: Connections between linear systems and convolutional codes. In: Marcus, B., Rosenthal, J. (eds.) Codes, Systems and Graphical Models. The IMA Volumes in Mathematics and its Applications, vol. 123, pp. 39–66. Springer, Heidelberg (2001)
[20] Rosenthal, J., Smarandache, R.: Maximum distance separable convolutional codes. Applicable Algebra in Engineering, Communication and Computing 10, 15–32 (1999)
[21] Rosenthal, J., Schumacher, J., York, E.V.: On behaviors and convolutional codes. IEEE Transactions on Information Theory 42(6), 1881–1891 (1996)
[22] Rosenthal, J., York, E.V.: BCH convolutional codes. IEEE Transactions on Information Theory 45(6), 1833–1844 (1999)
[23] Tanner, R.M.: Convolutional codes from quasicyclic codes: A link between the theories of block and convolutional codes. Technical Report USC-CRL-87-21 (November 1987)
[24] Thommesen, C., Justesen, J.: Bounds on distances and error exponents of unit memory codes. IEEE Transactions on Information Theory 29(5), 637–649 (1983)
[25] York, E.V.: Algebraic Description and Construction of Error Correcting Codes: A Linear Systems Point of View. Ph.D thesis, Department of Mathematics, University of Notre Dame, Indiana, USA (May 1997)

On Elliptic Convolutional Goppa Codes*

José Ignacio Iglesias Curto

Departamento de Matemáticas, Universidad de Salamanca,
Plaza de la Merced 1, 37008 Salamanca, Spain

Abstract. The algebraic geometric tools used by Goppa to construct block codes with good properties have been also used successfully in the setting of convolutional codes. We present here this construction carried out over elliptic curves, yielding a variety of codes which are optimal with respect to different bounds. We provide a number of examples for different values of their parameters, including some explicit strongly MDS convolutional codes. We also introduce some conditions for certain codes of this class to be MDS.

1 Introduction

Goppa codes, introduced by V.D. Goppa in the late seventies [4,5] were the seed of a fruitful link between coding theory and algebraic geometry. A further use of algebraic geometric tools in coding theory resulted in many other code constructions but also in the study of related open questions, as for instance the number of rational points of a given curve.

Different algebraic elements have been used also to construct convolutional codes. A *convolutional code* of length n and dimension k is often defined as a k-dimensional subspace of $\mathbb{F}_q^n(z)$. However it is known that there exists always a polynomial generator matrix. As explained in [1], the term "convolutional" is used because the output sequences can be regarded as the convolution of the input sequences with the sequences in the encoder. Hence, the output at time t does not only depend on the input at time t, but also on those at previous time instants. The output vector at time t is precisely the coefficient of z^t. The amount of this dependance is a critical parameter of the convolutional code called the *degree* or the *complexity* of the code, δ, and can be computed as the sum of the row degrees of a polynomial generator matrix in row proper form. In fact convolutional codes of degree 0 are precisely linear block codes. In addition, the highest row degree of a polynomial generator matrix in row proper form stands for the number of previous inputs on which every output depends, and it is another invariant of the code known as its *memory*, m.

Since the components of convolutional codewords are not constant the weight function considered is slightly different than the one in block coding. The weight of a convolutional codeword is the sum of the non-zero coefficients at each of

* Research partially supported by the research contract MTM2006-076B of DGI and by Junta de Castilla y León under research project SA029A08.

M. Bras-Amorós and T. Høholdt (Eds.): AAECC 2009, LNCS 5527, pp. 83–91, 2009.

its components. In a similar way the distance of two codewords is defined, and the minimum among them is called the *free distance*, d_f, of the code. Similarly to block coding, several bounds on the free distance are used. Two of the most commonly ones used, to which we will refer later (and which are generalizations of bounds for block codes), are

$$d_{free} \leq S(n,k,\delta) = (n-k)\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1\right) + \delta + 1$$

generalized Singleton bound [11]

$$d_{free} \leq max\{d' \in \{1,\ldots,S(n,k,\delta)\} | \sum_{l=0}^{k(m+i)-\delta-1} \left\lceil \frac{d'}{q^l} \right\rceil \leq n(m+i), \text{ for all } i \in \hat{\mathbb{N}}\}$$

$$\hat{\mathbb{N}} = \begin{cases} \mathbb{N} \equiv \{1,2,\ldots\} & \text{if } km = \delta \\ \mathbb{N}_0 \equiv \{0,1,2,\ldots\} & \text{if } km > \delta \end{cases}.$$

Griesmer bound [3]

By analogy with block coding, convolutional codes attaining the generalized Singleton bound are known as Maximum Distance Separable, MDS, codes.

As convolutional codes are a generalization of linear block codes, one might wonder wether algebraic geometric tools, and in particular similar tools as the ones used by Goppa, could be also used to construct families of convolutional codes with certain good properties.

A first attempt to define convolutional Goppa codes was made in [10], where instead of a curve, a family of curves parameterized by the affine line was considered. Instead of points, disjoint sections of the projection of this family over the affine line were taken, and instead of divisors on a curve, a Cartier divisor and an invertible sheaf. An analogous construction to the classical one led to a family of convolutional codes "of Goppa type".

After that, a more general construction with simpler geometric tools has been given in [9].

The paper is structured as follows. In Section 2 we expose the use of algebraic geometric elements to construct convolutional Goppa codes. In Section 3 we apply this construction to the case in which the curve considered is elliptic. An explicit expression of the elements involved, in particular of the rational points taken, will allow to determine the generator matrices of the codes so obtained. In Section 4 we present several examples of optimal elliptic convolutional Goppa codes. Finally in Section 5 we address the problem of characterizing MDS codes.

2 Geometric Construction of Convolutional Goppa Codes

Let $\mathbb{F}_q$ be a finite field and $\mathbb{F}_q(z)$ the field of rational functions on one variable. Let X be a smooth projective curve over $\mathbb{F}_q(z)$ of genus g and let us assume that $\mathbb{F}_q(z)$ is algebraically closed in the field of rational functions of X. Both Riemann-Roch and the Residues theorems still hold under this hypothesis [6].

Let us take n different $\mathbb{F}_q(z)$-rational points $P_1, \ldots, P_n$ and the divisor $D = P_1 + \cdots + P_n$, with its associated invertible sheaf $\mathcal{O}_X(D)$. We have then the exact sequence of sheaves

$$0 \to \mathcal{O}_X(-D) \to \mathcal{O}_X \to Q \to 0\,, \tag{1}$$

where Q is a sheaf with support only at the points P_i.

Let G be another divisor on X with support disjoint from D. By tensoring the exact sequence (1) by the associated invertible sheaf $\mathcal{O}_X(G)$, we have

$$0 \to \mathcal{O}_X(G-D) \to \mathcal{O}_X(G) \to Q \to 0\,.$$

and by taking global sections we get the sequence

$$\begin{aligned} 0 \to H^0(X, \mathcal{O}_X(G-D)) \to H^0(X, \mathcal{O}_X(G)) \xrightarrow{\alpha} H^0(X, Q) \to \\ \to H^1(X, \mathcal{O}_X(G-D)) \to H^1(X, \mathcal{O}_X(G)) \to 0 \end{aligned}.$$

If we impose $\deg(G) < n = \deg(D)$, we have an injective $\mathbb{F}_q(z)$-linear map

$$\begin{array}{ccccccc} 0 & \longrightarrow & L(G) & \xrightarrow{\alpha} & \mathbb{F}_q(z) \times \overset{n}{\cdots} \times \mathbb{F}_q(z) & \longrightarrow & \cdots \\ & & s & \longmapsto & (s(P_1), \ldots, s(P_n)) & & \end{array}$$

Definition 1. *The* convolutional Goppa code $\mathcal{C}(D,G)$ *defined by the divisors* D *and* G *is the image of* $\alpha\colon L(G) \to \mathbb{F}_q(z)^n$. *Given a subspace* $S \subseteq L(G)$, *the convolutional Goppa code* $\mathcal{C}(D,S)$ *defined by* D *and* S *is the image of* $\alpha_{|_S}$.

We can use the Riemann-Roch theorem to calculate the dimension of a convolutional Goppa code.

Proposition 1 ([9]). $\mathcal{C}(D,G)$ *is a convolutional code of length* $n = deg(D)$ *and dimension* $k \geq deg(G) + 1 - g$. *Further, if* $deg(G) > 2g-2$ *then* $k = deg(G) + 1 - g$.

Proof. Since $deg(G) < n$, $\dim L(G-D) = 0$, hence the map α is injective and $k = \dim L(G)$. If $2g - 2 < deg(G)$, $\dim L(G) = 1 - g + deg(G)$ by the Riemann-Roch theorem.

The geometric tools to characterize the free distance of convolutional Goppa codes are much more sophisticated than the analogous ones in the block case, involving jets, osculating planes and an interpretation of the points P_i as sections over the affine line. Then, the calculus of the free distance could be interpreted as a problem of Enumerative Geometry over finite fields [9]. However a few conditions to obtain MDS convolutional codes will be given in Section 5.

Similarly, a convolutional code can be constructed by considering the dual morphisms of those defining $\mathcal{C}(D,G)$ and by means of the Residues Theorem it can be proven that the code so constructed is the dual code of $\mathcal{C}(D,G)$.

3 Convolutional Goppa Codes over Elliptic Curves

Let $X \subset \mathbb{P}^2_{\mathbb{F}_q(z)}$ be a plane elliptic curve over $\mathbb{F}_q(z)$. Without loss of generality we will assume that X has a rational point of order at least 4 (so that there are enough rational points to define a convolutional code). Then, in an affine plane containing this point, X can be written in Tate Normal form [7]

$$y^2 + axy + by = x^3 + bx^2 \tag{2}$$

being x, y the affine coordinates in this plane and $a, b \in \mathbb{F}_q(z)$. Let P_∞ be the point at infinity, $P_0 = (0, 0)$ and $P_1, \ldots, P_n$ n different rational points of X, with $P_i = (x_i, y_i)$ and $x_i, y_i \in \mathbb{F}_q(z)$. Consider the divisors $D = P_1 + \cdots + P_n$ and $G = rP_\infty$, with $0 < r < n$.

Recall that the divisors of the functions x, y are

$$(x) = P_0 + Q - 2P_\infty \, , \, (y) = 2P_0 + Q' - 3P_\infty$$

where Q, Q' are two rational points different from P_0, P_∞.

Then, a basis of $L(G)$ is given by $\{1, x, y, \ldots, x^i y^j, \ldots\}$, where i, j satisfy $2i + 3j \leq r = deg(G)$ (and to avoid linear dependencies $j = 0, 1$).

Since $r < n$ then the evaluation map

$$\begin{aligned} \alpha : L(G) &\longrightarrow \mathbb{F}_q(z)^n \\ x^i y^j &\longmapsto (x_1^i y_1^j, \ldots, x_n^i y_n^j) \end{aligned}$$

is an injective morphism and $Im\alpha$ defines the convolutional Goppa code $\mathcal{C}(D, G)$ with length n. As $g = 1$ and $deg(G) > 2g - 2$ the code has dimension $k = r = deg(G)$.

Let us consider now the more general case with $G = rP_\infty - sP_0$, where $r, s > 0$ and $0 < r - s < n$. Then $\{x^a y^b \mid s \leq a + 2b, \ 2a + 3b \leq r\}$ is a basis of $L(G)$. The code $\mathcal{C}(D, G)$ has length n and dimension $r - s$.

The explicit expression of a generator matrix of the code $\mathcal{C}(D, G)$ is

$$G = \begin{pmatrix} x_1^a y_1^b & x_2^a y_2^b & \ldots & x_n^a y_n^b \\ x_1^{a+1} y_1^b & x_2^{a+1} y_2^b & \ldots & x_n^{a+1} y_n^b \\ \vdots & \vdots & \ddots & \vdots \\ x_1^c y_1^d & x_2^c y_2^d & \ldots & x_n^c y_n^d \end{pmatrix}.$$

The following examples illustrate this construction, as well as the fact that it provides optimal codes.

Example 1. Let us consider the elliptic curve with Tate Normal form

$$y^2 + zxy + y = x^3 + x^2$$

over the field $F_2(z)$.

We take the divisor $D = P_1 + P_2 + P_3 + P_4$ with

$$P_1 = (1+z, z) \qquad P_2 = (1+z, 1+z^2)$$
$$P_3 = (\tfrac{1+z^3}{z^2}, \tfrac{1+z^3+z^4+z^5}{z^3}) \; P_4 = (\tfrac{1+z^3}{z^2}, \tfrac{1+z^2+z^4}{z^3})$$

and the divisor $G = 3P_\infty - P_0$, being therefore $L(G) = \langle\, x,\, y \rangle$. Then the convolutional Goppa Code defined by D and G is generated by the matrix

$$\begin{pmatrix} z^2 & z^2 & 1+z+z^2 & 1+z+z^2 \\ 1+z & 1+z^2+z^3 & 1+z+z^3 & 0 \end{pmatrix},$$

where the rows are the images of the functions $\frac{z^2}{1+z}x$ and $zy + \frac{1+z+z^2}{1+z}x$ respectively. $\mathcal{C}(D, G)$ has parameters $[n, k, \delta, m, d_{free}] = [4, 2, 5, 3, 8]$. The free distance of the code attains the Griesmer bound.

Example 2. We consider now the elliptic curve

$$y^2 + zxy + 2z^2y = x^3 + 2z^2x^2$$

over $\mathbb{F}_5(z)$, and the divisor $D = P_1 + P_2$, with support at the points

$$P_1 = (3z^2, 3z^2 + 2z^3) \; P_2 = (\tfrac{2z^2+3z^3+4z^4}{1+3z+z^2}, \tfrac{2z^4+2z^5+3z^6}{4+3z+2z^2+z^3})$$

We take the divisor $G = 2P_\infty - P_0$, and we have $L(G) = \langle x \rangle$.

A generator matrix of the code $\mathcal{C}(D, G)$ is

$$\begin{pmatrix} 2+z+2z^2, 3+2z+z^2 \end{pmatrix}$$

given by the image of the function $\frac{4+2z+4z^2}{z^2}x$. The code has parameters $[n, k, \delta, d_{free}] = [2, 1, 2, 6]$, and hence it is MDS.

Example 3. Now we take the curve

$$y^2 + (1+z+z^2)xy + (z^2+z^3)y = x^3 + (z^2+z^3)x^2$$

over $\mathbb{F}_q(z)$, with $q \neq 2^m$. We consider the divisor $D = P_1 + P_2 + P_3$ where

$$P_1 = (0, -z^3 - z^2)$$
$$P_2 = (z^2 - z, -z^4 - 2z^3 + z^2)$$
$$P_3 = (-z^2 - z, -z^3 + z)$$

Let us take $G = 2P_\infty$, then $L(G) = \langle 1, x \rangle$ and a generator matrix for the convolutional Goppa code $\mathcal{C}(D, G)$ is

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1-z & 1+z \end{pmatrix},$$

whose rows are images of the functions 1 and $-\frac{1}{z}x$. $\mathcal{C}(D, G)$, has parameters $[n, k, \delta, m, d_{free}] = [3, 2, 1, 1, 3]$. Then $\mathcal{C}(D, G)$ is an MDS convolutional code.

4 Some Optimal Convolutional Goppa Codes over Elliptic Curves

The previous examples illustrate the construction of elliptic convolutional Goppa codes with different parameters from their defining elements, i. e., the elliptic curve and the divisors D and G. They also show the existence of optimal convolutional codes defined over elliptic curves.

In this section we present a representative collection of convolutional Goppa codes defined over elliptic curves. A more extensive one can be found in [8]. The description of the codes is done by means of their generating elements. To characterize the elliptic curve over which each code is defined, let a, b be the parameters on its Tate Normal form (2). On this curve we take the points $\{P_i\}_{i=1}^{7} = \{2P, 3P, -3P, 4P, -4P, 5P, -5P\}$, where $P = (0, 0) = P_0$ is a rational point which belongs to the curve, and nP, $n \in \mathbb{Z}$ is obtained by the addition law defined over every elliptic curve. Together with them, the basis or the bases (when in the same curve and with the same divisor D there is more than one code for those parameters) of the space of functions that define each code are provided. We include for each code its parameters, the size p of the base field (which for the codes presented will be always prime) and the bound that it reaches so that it is considered optimal.

As shown on the table, for those codes that are optimal with respect to the Griesmer bound (i.e., there does not exist MDS codes over that field) there exist also MDS codes if the size of the base field is big enough.

4.1 Strongly MDS Convolutional Codes

The set of *strongly MDS* convolutional codes is a particularly interesting subset of MDS convolutional codes. They are characterized by the property that to decode the coefficients vector at time t, the smallest possible number of coefficients vectors of the received word are needed. This property is very convenient to develop iterative decoding algorithms, as the one in [2], with an error decoding capability per time interval similar to MDS block codes of a large length.

The family of convolutional Goppa codes defined over elliptic curves provides also a number of codes which are strongly MDS. A representative subset of them are presented in the following table. More examples can be found in [8].

We would like to stress two interesting facts. In [2] several examples of strongly MDS convolutional codes are presented, which are obtained by different methods. However all of them have in common that the length of the code and the characteristic of the base field must be coprime. This condition is not necessary in our construction and actually some of the codes obtained in this way don't fulfill it. Secondly, the already mentioned decoding algorithm for this kind of codes that is proposed in the same paper, has the drawback of needing a general syndrome decoding algorithm. The variety of examples presented suggests that elliptic convolutional Goppa codes are a promising way to obtain strongly MDS codes with a rich algebraic structure that would allow to adapt that decoding scheme in order to make it practical.

Table 1. Examples of optimal elliptic convolutional Goppa codes

$[n,k,\delta]$	p	d_{free}	(a,b)	$L(G)$	i,$D=\sum P_i$	Bound
[2,1,1]	3	4	$(\alpha_1 z, \alpha_1 z + 2z^2)$	$\{x\}$	1, 4	MDS
[2,1,1]	≥ 3	4	$(z, \alpha_2 z - \alpha_2 z^2)$	$\{x\}$	1, 4	MDS
[2,1,2]	2	5	$(1+z, 1)$	$\{y\}$, $\{xy\}$	4, 5	Griesmer
[2,1,2]	≥ 5	6	$(2z, 1)$	$\{x\}$	2, 6	MDS
[2,1,7]	7	15	$(z+2z^2, 1+4z+3z^2)$	$\{y\}$, $\{xy\}$	6, 7	Griesmer
[2,1,7]	13	16	$(3z^2, 1+2z+2z^2)$	$\{y\}$, $\{xy\}$	6, 7	MDS
[3,1,1]	≥ 5	6	$(z, 2-5z+3z^2)$	$\{y\}$, $\{xy\}$	1, 4, 5	MDS
[3,1,2]	2	8	(z, z^2)	$\{x\}$	1, 4, 5	Griesmer
[3,1,2]	7	9	$(z, 1+2z+4z^2)$	$\{y\}$	1, 4, 5	MDS
[3,1,6]	11	21	$(z, 1+3z^2)$	$\{xy\}$	1, 4, 5	MDS
[4,1,2]	5	12	$(z, 1+2z+2z^2)$	$\{y\}$	2, 3, 6, 7	MDS
[4,1,6]	11	28	$(z, 1+4z+6z^2)$	$\{xy\}$	2, 3, 6, 7	MDS
[4,1,8]	11	36	$(3z, 6+4z^2)$	$\{y\}$	2, 3, 6, 7	MDS
[3,2,1]	5	3	$(1+z, z+3z^2)$	$\{x,y\}$, $\{x^2,xy\}$	1, 4, 5	MDS
[3,2,1]	7	3	$(1+2z, 2z+z^2)$	$\{x,y\}$, $\{x^2,xy\}$	1, 4, 5	MDS
[3,2,3]	7	6	$(2z, 2+6z^2)$	$\{x,y\}$	1, 4, 5	MDS
[4,2,1]	≥ 3	4	$(z, -2+2z)$	$\{1,x\}$	0, 1, 2, 4	MDS
[4,2,1]	≥ 3	4	$(z, -2+3z-z^2)$	$\{1,x\}$	1, 2, 4, 6	MDS
[4,2,3]	7	8	$(2z, 3+2z^2)$	$\{x,y\}$	2, 3, 6, 7	MDS
[4,2,5]	2	8	$(z, 1)$	$\{x,y\}$	2, 3, 6, 7	Griesmer
[4,2,5]	13	12	$(z, 4+4z+5z^2)$	$\{x,y\}$	2, 3, 6, 7	MDS
[5,2,1]	≥ 3	5	$(z, -2+3z-z^2)$	$\{1,x\}$	0, 1, 2, 4, 6	MDS
[4,3,2]	7	4	$(5+\alpha_3 z, 3-\alpha_3 z)$	$\{1,x,y\}$	1, 3, 6, 7	MDS
[4,3,2]	11	4	$(6+\alpha_3 z, 6-\alpha_3 z)$	$\{1,x,y\}$	1, 3, 6, 7	MDS

$\alpha_2 \neq -1$ $\alpha_3 \neq 0$

Table 2. Examples of strongly MDS elliptic convolutional Goppa codes

$[n,k,\delta]$	p	d_{free}	(a,b)	$L(G)$	i,$D=\sum P_i$
[2,1,1]	5	4	$(z, 1+2z+2z^2)$	$\{x\}$	2, 6
[2,1,1]	7	4	$(z, 2+2z+3z^2)$	$\{x\}$	2, 6
[2,1,2]	11	6	$(0, 2+\alpha_1 z)$	$\{y\}$, $\{xy\}$	4, 5
[2,1,2]	11,13	6	$(2z, 3)$	$\{y\}$, $\{xy\}$	4, 5
[2,1,2]	13	6	$(2z, 5+3z)$	$\{y\}$, $\{xy\}$	6, 7
[3,1,2]	3	3	$(z^2, 1+z)$	$\{1,x\}$	1, 2, 6
[3,1,2]	≥ 5	3	$(z, 1-3z+2z^2)$	$\{x,y\}$,$\{x^2,xy\}$	1, 4, 5
[4,2,1]	5,11	4	$(2+\alpha_2 z, \alpha_3+\alpha_2\alpha_3 z)$	$\{1,x\}$	0, 1, 2, 4

$\alpha_1 = 1, \ldots, 6$, $\alpha_2 \geq 1$, $\alpha_3 \geq 2$

5 Conditions on Maximum Distance Separability

In [11] a condition is provided so that a codeword has weight bigger or equal to the value for the generalized Singleton bound, the so-called *weight property*. A codeword holds this property if every subset of k components has weight at

least $\delta + 1$. Hence a sufficient condition for a convolutional code to be MDS is that every codeword holds the weight property.

As it is shown, the weight property implies that at least $n-k+1$ components must have weight at least $\lfloor \delta/k \rfloor + 1$. In particular, since these components must be of course different from 0, the previous condition is also a sufficient condition to be a MDS block code over $\mathbb{F}_q(z)$.

It is easy to check that actually 1-dimensional MDS convolutional codes are also MDS as block codes. In particular, the necessary conditions for 1-dimensional MDS block codes are also of application in the convolutional case.

One of the necessary conditions that have to be verified is the following.

Proposition 2. *If $\mathcal{C}(D,G)$ is a 1-dimensional MDS elliptic code, then for any point P such that $G \sim P$, $P \notin supp(D)$.*

Proof. Since $G \sim P$, then there exists a function f such that $(f) = G - P$, i. e., $f \in L(G-P)$. On the other side, f may not have a zero on the support of D since we assume that $\mathcal{C}(D,G)$ is MDS. Then $P \notin supp(D)$.

In particular, for elliptic convolutional Goppa codes we have

Proposition 3. *If $\mathcal{C}(D,G)$ is a 1-dimensional MDS convolutional code, then for any function f such that $\alpha(f)$ is polynomial of degree δ, then $w(f(P_i)) = \delta + 1$ for all i.*

Proof. It is a direct consequence of generalized Singleton bound.

Note that these results cannot be extended to $(n-1)$-dimensional codes (as one would do for block codes) since the dual code of a MDS convolutional code may not be MDS.

6 Conclusions

The Goppa construction to define linear block codes can be also considered in a very natural way on the convolutional coding context. In particular an explicit expression of convolutional Goppa codes defined over elliptic curves can be given. This construction yields optimal codes over different fields and for different values of their parameters. It is of particular interest the set of strongly MDS convolutional codes of this family. However the problem of determining the free distance is far harder than in the block coding case. Some conditions to get 1-dimensional MDS convolutional codes have been given, but still remains unknown wether they can be extended for codes of any dimension.

References

1. Forney, G.D.: Convolutional codes I: Algebraic structure. IEEE Trans. Information Theory (1970)
2. Gluesing-Luerssen, H., Rosenthal, J., Smarandache, R.: Strongly MDS convolutional codes. IEEE Trans. Information Theory (52), 584–598 (2006)

3. Gluesing-Luerssen, H., Schmale, W.: Distance bounds for convolutional codes and some optimal codes (2003), arXiv:math/0305135v1
4. Goppa, V.D.: Codes associated with divisors. Probl. Peredachi Inform. 13(1), 33–39 (1977); Translation: Probl. Inform. Transmission 13, 22–26 (1977)
5. Goppa, V.D.: Codes on algebraic curves. Dokl. Adad. Nauk SSSR 259, 1289–1290 (1981); Translation: Soviet Math. Dokl. 24, 170–172 (1981)
6. Hartshorne, R.: Algebraic geometry. Grad. Texts in Math., vol. 52. Springer, New York (1977)
7. Husemoller, D.: Elliptic curves. Springer, New York (1987)
8. Iglesias Curto, J.I.: A study on convolutional codes. Classfication, new families and decoding, Ph.D. thesis, Universidad de Salamanca (2008)
9. Muñoz Porras, J.M., Dominguez Perez, J.A., Iglesias Curto, J.I., Serrano Sotelo, G.: Convolutional Goppa codes. IEEE Trans. Inform. Theory 52(1), 340–344 (2006)
10. Muñoz Porras, J.M., Domínguez Pérez, J.A., Serrano Sotelo, G.: Convolutional codes of Goppa type. AAECC 15, 51–61 (2004)
11. Rosenthal, J., Smarandache, R.: Maximum distance separable convolutional codes. AAECC 10(1), 15 (1999)

The Minimum Hamming Distance of Cyclic Codes of Length $2p^s$

Hakan Özadam and Ferruh Özbudak

Department of Mathematics and Institute of Applied Mathematics
Middle East Technical University, İnönü Bulvarı, 06531, Ankara, Turkey
{ozhakan,ozbudak}@metu.edu.tr

Abstract. We study cyclic codes of length $2p^s$ over $\mathbb{F}_q$, where p is an odd prime. Using the results of [1], we compute the minimum Hamming distance of these codes.

Keywords: Cyclic code; repeated-root cyclic code; Hamming distance.

1 Introduction

Although it has been shown in [1] that repeated-root cyclic codes over finite fields are "asymptotically bad", they remain to be interesting in some cases (see, for example, [6,7,8,9]). For instance, a sequence of binary repeated-root cyclic codes, that are optimal, was found in [8]. Moreover, in [6], it has been shown that the minimum Hamming distance of repeated-root cyclic codes over Galois rings can be determined by the minimum Hamming distance of codes of the same length over the residue finite field. Using this result of [6], the minimum Hamming distance of cyclic codes of length p^s over Galois rings of characteristic p^a is given in [7].

The minimum Hamming distance of cyclic codes of length p^s over a finite field of characteristic p is given in [2]. Later in [5], we have shown that the minimum Hamming distance of these codes can also be computed by using the results of [1] via simpler and more direct methods compared to that of [2]. In this study, we extend our methods to cyclic codes of length $2p^s$. Namely, we determine the minimum Hamming distance of all cyclic codes, of length $2p^s$, over a finite field of characteristic p, where p is an odd prime and s is an arbitrary positive integer.

This paper is organized as follows. In Section 2, we fix our notation and recall some preliminaries. In Section 3, using the results of [1], we compute the minimum Hamming distance of all cyclic codes of length $2p^s$ over a finite field of characteristic p, where p is an odd prime. We summarize our results in Table 1 at the end of Section 3.

2 Preliminaries

Let p be an odd prime and $\mathbb{F}_q$ be a finite field of characteristic p. Let n be a positive integer. Throughout this paper we identify a codeword $(a_0, a_1, \ldots, a_{n-1})$

M. Bras-Amorós and T. Høholdt (Eds.): AAECC 2009, LNCS 5527, pp. 92–100, 2009.

over $\mathbb{F}_q$ with the polynomial $a(x) = a_0 + a_1 x + \cdots + a_{n-1}x^{n-1} \in \mathbb{F}_q[x]$. We denote the minimum Hamming distance of a code C by $d_H(C)$.

Let $\mathfrak{a} = \langle x^n - 1 \rangle$ be an ideal of $\mathbb{F}_q[x]$ and let $\mathcal{R}_\mathfrak{a}$ be the finite ring given by $\mathcal{R}_\mathfrak{a} = \mathbb{F}_q[x]/\mathfrak{a}$. It is well-known that cyclic codes, of length n, over $\mathbb{F}_q$ are ideals of $\mathcal{R}_\mathfrak{a}$ (see, for example, Chapter 7 of [4]). Any element of $\mathcal{R}_\mathfrak{a}$ can be represented uniquely as $f(x) + \mathfrak{a}$ where $deg(f(x)) < n$. The codeword which corresponds to $f(x) + \mathfrak{a}$ is $(f_0, f_1, \ldots, f_{n-1})$, where $f(x) = f_0 + f_1 x + \cdots + f_{n-1}x^{n-1} \in \mathbb{F}_q[x]$. Since $\mathbb{F}_q[x]$ is a principal ideal domain, for any ideal I of $\mathcal{R}_\mathfrak{a}$, there exists a unique monic polynomial $g(x) \in F_q[x]$ with $\deg(g(x)) < n$ and $g(x)|x^n - 1$ such that $I = \langle g(x) \rangle$.

Let $h(x) \in \mathbb{F}_q[x]$ be a polynomial with $h(0) \neq 0$, then $h(x) \mid x^m - 1$ for some positive integer m (c.f. [3, Lemma 3.1]). Let e be the least positive integer with $h(x) \mid x^e - 1$. The *order* of $h(x)$ is defined to be e and it is denoted by $\mathrm{ord}(h) = e$.

3 Cyclic Codes of Length $2p^s$

In this section we determine the minimum Hamming distance of all cyclic codes of length $2p^s$ over $\mathbb{F}_q$, where s is a positive integer.

Let $0 \leq i, j \leq p^s$ be integers. Since $x^{2p^s} - 1 = (x^2 - 1)^{p^s} = (x-1)^{p^s}(x+1)^{p^s}$, all cyclic codes of length $2p^s$ over $\mathbb{F}_q$ are of the form $\langle (x-1)^i (x+1)^j \rangle$.

Let $\mathcal{R}_\mathfrak{a} = \mathbb{F}_q[x]/\langle x^{2p^s} - 1 \rangle$ and let $C = \langle (x-1)^i (x+1)^j \rangle \subset \mathcal{R}_\mathfrak{a}$. If $(i, j) = (0, 0)$, then $C = \mathcal{R}_\mathfrak{a}$ and if $(i, j) = (p^s, p^s)$, then $C = \{0\}$. These are the trivial cyclic codes of length $2p^s$ over $\mathbb{F}_q$. In the rest of this section, we compute the minimum Hamming distance of all non-trivial cyclic codes of length $2p^s$ over $\mathbb{F}_q$.

We partition the set $\{0, 1, \ldots, p^s\}$ into 4 subsets. The first one is the range $0, \ldots, p^{s-1}$, the second one is the range $p^{s-1} + 1, \ldots, (p-1)p^{s-1}$, the third one is the range $(p-1)p^{s-1} + 1, \ldots, p^s - 1$ and the last one just consists of p^s. These ranges arise naturally from technicalities of our computations explained below.

If $i = 0$, or $j = 0$, or $0 \leq i, j \leq p^{s-1}$, then the minimum Hamming distance of C can easily be determined as shown in Lemma 2 and Lemma 3 below.

We note that from Lemma 8 till Theorem 2, we consider only the cases, where $i \geq j$, explicitly. This is because the cases, where $j > i$, can be treated similarly as the corresponding case of $i > j$ and consequently, we have the analogous results in these cases. Finally, in Theorem 2, we state all of our results explicitly corresponding to each case including the ones where $j > i$.

First we give an overview of the results in this section. If $0 < j \leq p^{s-1}$ and $p^{s-1} + 1 \leq i \leq p^s$, then the minimum Hamming distance of C is computed in Lemma 8 and Lemma 9. Note that we use the results of [1] for our computations after Lemma 5. If $p^{s-1} + 1 \leq j \leq i \leq (p-1)p^{s-1}$, then the minimum Hamming distance of C is computed in Lemma 10. If $p^{s-1} + 1 \leq j \leq (p-1)p^{s-1} < i \leq p^s - 1$, then the minimum Hamming distance of C is computed in Lemma 11. If $(p-1)p^{s-1} + 1 \leq j \leq i \leq p^s - 1$, then the minimum Hamming distance of C is computed in Lemma 12 and Lemma 13. If $0 < j \leq p^{s-1}$ and $i = p^s$, then the minimum Hamming distance of C is computed in Lemma 14. If $p^{s-1} + 1 \leq j \leq (p-1)p^{s-1}$ and $i = p^s$, then the minimum Hamming distance of C is computed

in Lemma 15. Finally, if $(p-1)p^{s-1}+1 \leq j \leq p^s-1$ and $i=p^s$, then the minimum Hamming distance of C is computed in Lemma 16.

We begin to present and prove our results with the next lemma.

Lemma 1. *Let C be an ideal of $\mathcal{R}_\mathfrak{a}$ with $\{0\} \neq C \subsetneq \mathcal{R}_\mathfrak{a}$. Then $d_H(C) \geq 2$.*

Proof. This follows from the observation that αx^m is a unit in $\mathcal{R}_a$ for $\alpha \in \mathbb{F}_q \backslash \{0\}$ and $m \in \mathbb{Z}^+ \cup \{0\}$. □

Lemma 2. *Let $0 < i, j \leq p^s$ be integers, let $C = \langle (x-1)^i \rangle$ and $D = \langle (x+1)^j \rangle$. Then $d_H(C) = d_H(D) = 2$.*

Proof. The proof follows from Lemma 1 and the observation that $(x-1)^i | x^{p^s}-1$ and $(x+1)^j | x^{p^s}+1$. □

Lemma 3. *Let $C = \langle (x+1)^i (x-1)^j \rangle$, for some $0 \leq i, j \leq p^{s-1}$ with $(i,j) \neq (0,0)$. Then $d_H(C) = 2$.*

Proof. For this range of i and j, we have $(x+1)^i(x-1)^j | (x^{2p^{s-1}}-1)$. □

The following lemma shows that we can use the results of [1] for computing the minimum Hamming distance of the remaining codes.

Lemma 4. *Let $f(x) = (x-1)^i(x+1)^j \in \mathbb{F}_q[x]$ for some integers $0 \leq i, j \leq p^e$. Then $ord(f) = 2p^e$ if $j > p^{e-1}$, or if $j > 0$ and $i > p^{e-1}$.*

Proof. The proof follows from the fact that $\mathrm{ord}(x+1) = 2$, [3, Theorem 3.8] and [3, Theorem 3.9]. □

$$\begin{aligned} \text{Let} \quad G_1 &= \{(x-1)^i(x+1)^j : \quad p^s \geq i > 0 \quad \text{and} \quad p^s \geq j > p^{s-1}\}, \\ G_2 &= \{(x-1)^i(x+1)^j : \quad p^s \geq i > p^{s-1} \quad \text{and} \quad p^s \geq j > 0\} \end{aligned}$$

and let $g(x)$ be any element of $G_1 \cup G_2 \setminus \{(x-1)^{p^s}(x+1)^{p^s}\}$. By Lemma 4, $\mathrm{ord}(g(x)) = 2p^s$. Therefore we can use the results of [1] to determine the minimum Hamming distance of $\langle g(x) \rangle$.

Let $0 \leq t \leq p^s - 1$ be an integer. For $g(x) = (x-1)^i(x+1)^j \in G_1 \cup G_2 \setminus \{(x-1)^{p^s}(x+1)^{p^s}\}$, we define $C = \langle g(x) \rangle$,

$$e_{i,t} = \begin{cases} 1, \text{ if } \quad i > t, \\ 0, \text{ otherwise,} \end{cases} \quad \text{and} \quad e_{j,t} = \begin{cases} 1, \text{ if } \quad j > t, \\ 0, \text{ otherwise.} \end{cases}$$

Let $\bar{g}_t(x) = (x-1)^{e_{i,t}}(x+1)^{e_{j,t}}$ and let $\bar{C}_t = \langle \bar{g}_t(x) \rangle \subset \mathbb{F}_q[x]/\langle x^2-1 \rangle$ be the simple-root cyclic code depending on C and t. Following the conventions of [1], we set

$$d_H(\bar{C}_t) = \begin{cases} 2, & \text{if} \quad \bar{g}_t(x) = x-1 \quad \text{or} \quad \bar{g}_t(x) = x+1, \\ 1, & \text{if} \quad \bar{g}_t(x) = 1, \\ \infty, & \text{if} \quad \bar{g}_t(x) = x^2-1. \end{cases}$$

Then we have the following cases.

$$\begin{array}{l}\text{If} \quad i \geq j > t \quad \text{or} \quad j \geq i > t, \quad \text{then} \quad \bar{g}_t(x) = x^2 - 1 \\ \text{and therefore} \quad d_H(\bar{C}_t) = \infty.\end{array} \tag{1}$$

$$\begin{array}{l}\text{If} \quad i > t \geq j \quad \text{or} \quad j > t \geq i, \quad \text{then} \quad \bar{g}_t(x) = x - 1 \quad \text{or} \\ \bar{g}_t(x) = x + 1, \text{respectively, and therefore} \quad d_H(\bar{C}_t) = 2.\end{array} \tag{2}$$

$$\begin{array}{l}\text{If} \quad t \geq i \geq j \quad \text{or} \quad t \geq j \geq i, \quad \text{then} \quad \bar{g}_t(x) = 1 \\ \text{and therefore} \quad d_H(\bar{C}_t) = 1.\end{array} \tag{3}$$

For $0 \leq t \leq p^s - 1$, let $0 \leq t_0, t_1, \ldots, t_{s-1} \leq p - 1$ be the uniquely determined integers such that $t = t_0 + t_1 p + \cdots + t_{s-1} p^{s-1}$, and P_t be the positive integer given by

$$P_t = \prod_{m=0}^{s-1} (t_m + 1) \in \mathbb{Z}.$$

We define (cf. [1, page 339]) the set $T = \{t : t = (p-1)p^{s-1} + (p-1)p^{s-2} + \cdots + (p-1)p^{s-(j-1)} + rp^{s-j},\ 1 \leq j \leq s,\ 1 \leq r \leq p-1\} \cup \{0\}$.

In [1], it has been shown that we can express the minimum Hamming distance of C in terms of $d_H(\bar{C}_t)$ and P_t.

Lemma 5 ([1, Lemma 1]). *Let C, $\bar{C}_t$ and P_t be as above. The minimum Hamming distance of C satisfies $d_H(C) \leq P_t d_H(\bar{C}_t)$ for all $t \in \{0, 1, \ldots, p^s - 1\}$.*

Theorem 1 ([1, Theorem 1]). *Let C, $\bar{C}_t$ and P_t be as above. Then $d_H(C) = P_t d_H(\bar{C}_t)$ for some $t \in T$.*

Combining Lemma 5 and Theorem 1, we obtain

$$d_H(C) = \min\{P_t d_H(\bar{C}_t) : \quad t \in T\}. \tag{4}$$

Clearly, (1) implies that $d_H(C) = P_t d_H(\bar{C}_t)$ is impossible for $i \geq j > t$ and $j \geq i > t$. So we will consider only the cases (2) and (3) in our computations.

The following proposition is a useful tool for our computations.

Proposition 1. *Let $t, t' \in T$, then $t < t'$ if and only if $P_t < P_{t'}$.*

Proof. We have

$$\begin{aligned} t &= (p-1)p^{s-1} + (p-1)p^{s-2} + \cdots + (p-1)p^{s-(e-1)} + rp^{s-e} \quad \text{and} \\ t' &= (p-1)p^{s-1} + (p-1)p^{s-2} + \cdots + (p-1)p^{s-(e'-1)} + r'p^{s-e'} \end{aligned}$$

for some $1 \leq e, e' \leq s$ and $1 \leq r, r' \leq p-1$ as $t, t' \in T$. First assume that $t > t'$. Then either $e > e'$, or $e = e'$ and $r > r'$. If $e > e'$, then $p^{e-e'} \geq p$ so $p^{e-e'}(r+1) \geq 2p$. Moreover $r' \leq p-1$ implies $r' + 1 \leq p$. So we get $p^{e-e'}(r+1) - (r'+1) \geq p$. Therefore

$$P_t - P_{t'} = p^{e'}(p^{e-e'}(r+1) - (r'+1)) \geq p^{e'} p > 0.$$

Hence $P_t > P_{t'}$. If $e = e'$ and $r > r'$, then $P_t - P_{t'} = p^e((r+1) - (r'+1)) = p^e(r - r') > 0$ and hence $P_t > P_{t'}$. Using similar arguments, we obtain that if $P_t > P_{t'}$, then $t > t'$. This completes the proof. □

As an immediate consequence of Proposition 1, we have

$$\min\{P_t : \quad t \in T\ , \quad t \geq j\} = \min\{P_t : \quad t \in T\ , \quad i > t \geq j\}, \tag{5}$$

provided that these sets are nonempty. We use (5) from Lemma 8 on.

The minimum value of P_t is given in the following two lemmas, as t runs through certain sets. These sets are determined from the ranges of i and j in the definition of cyclic codes.

Note that if i is an integer satisfying $1 \leq i \leq (p-1)p^{s-1}$, then there exists a uniquely determined integer β such that $0 \leq \beta \leq p-2$ and $\beta p^{s-1} + 1 \leq i \leq (\beta+1)p^{s-1}$.

Lemma 6. *Let ℓ be an integer such that $\beta p^{s-1} + 1 \leq \ell \leq (\beta+1)p^{s-1}$, where β is an integer with $0 \leq \beta \leq p-2$. Then* $\min\{P_t : \quad t \geq \ell \quad \text{and} \quad t \in T\} = \beta + 2$.

Proof. If $t \geq \ell$ and $t \in T$, then $t \geq (\beta+1)p^{s-1}$. So we get $\min\{t \in T : \quad t \geq \ell\} = (\beta+1)p^{s-1}$. Using Proposition 1, we obtain $\min\{P_t : \quad t \geq \ell \quad \text{and} \quad t \in T\} = \beta + 2$. □

Note that $p^s - p^{s-1} < p^s - p^{s-2} < \cdots < p^s - p^{s-s} = p^s - 1$. Hence for an integer i satisfying $(p-1)p^{s-1} + 1 = p^s - p^{s-1} + 1 \leq i \leq p^s - 1$, there exists a uniquely determined integer k such that $1 \leq k \leq s-1$ and $p^s - p^{s-k} + 1 \leq i \leq p^s - p^{s-k-1}$. Moreover we have $p^s - p^{s-k} < p^s - p^{s-k} + p^{s-k-1} < p^s - p^{s-k} + 2p^{s-k-1} < \cdots < p^s - p^{s-k} + (p-1)p^{s-k-1}$ and $p^s - p^{s-k} + (p-1)p^{s-k-1} = p^s - p^{s-k-1}$. Therefore for such integers i and k, there exists a uniquely determined integer τ with $1 \leq \tau \leq p-1$ such that $p^s - p^{s-k} + (\tau-1)p^{s-k-1} + 1 \leq i \leq p^s - p^{s-k} + \tau p^{s-k-1}$.

Lemma 7. *Let τ and k be integers with $1 \leq \tau \leq p-1$ and $1 \leq k \leq s-1$. If ℓ is an integer with $p^s - p^{s-k} + (\tau-1)p^{s-k-1} + 1 \leq \ell \leq p^s - p^{s-k} + \tau p^{s-k-1}$, then* $\min\{P_t : \quad t \geq \ell \quad \text{and} \quad t \in T\} = (\tau+1)p^k$.

Proof. For $0 \leq \alpha \leq p-1$, we have $p^s - p^{s-k} + \alpha p^{s-k-1} = (p-1)p^{s-1} + (p-1)p^{s-2} + \cdots + (p-1)p^{s-k} + \alpha p^{s-k-1}$. So we get

$$\begin{aligned}&(p-1)p^{s-1} + (p-1)p^{s-2} + \cdots + (p-1)p^{s-k} + (\tau-1)p^{s-k-1} + 1 \leq \ell \\ &\leq (p-1)p^{s-1} + (p-1)p^{s-2} + \cdots + (p-1)p^{s-k} + \tau p^{s-k-1}.\end{aligned}$$

Now $t \geq \ell$ and $t \in T$ implies $t \geq (p-1)p^{s-1} + (p-1)p^{s-2} + \cdots + (p-1)p^{s-k} + \tau p^{s-k-1}$. Thus $\min\{t \in T : \quad t \geq \ell\} = (p-1)p^{s-1} + (p-1)p^{s-2} + \cdots + (p-1)p^{s-k} + \tau p^{s-k-1}$. Hence, using Proposition 1, we obtain $\min\{P_t : \quad t \in T \quad \text{and} \quad t \geq \ell\} = (\tau+1)p^{s-k}$. □

Now we are ready to obtain the minimum Hamming distance of the cyclic codes when $0 < j \leq p^{s-1} < i < p^s$ in the following two lemmas. Recall that from Lemma 8 till Theorem 2 we state the results when $j \leq i$ only. For $j > i$, the

analogous results are obtained by the same method, and their statements are included in Theorem 2.

Lemma 8. *Let $C = \langle (x-1)^i(x+1)^j \rangle$, where $2p^{s-1} \geq i > p^{s-1} \geq j > 0$. Then $d_H(C) = 3$.*

Proof. Note that $p^{s-1} \in T$ and $i > p^{s-1} \geq j$. Hence the set $\{P_t : \quad t \in T \quad \text{and} \quad i > t \geq j\}$ is nonempty. So, by Lemma 6, we get $\min\{P_t : \quad t \in T \quad \text{and} \quad i > t \geq j\} = 2$. If $i > t \geq j$, then $d_H(\bar{C}_t) = 2$ by (2), thus

$$\min\{P_t d_H(\bar{C}_t) : \quad t \in T \quad \text{and} \quad i > t \geq j\} = 4. \tag{6}$$

If $t \geq i > j$ and $2p^{s-1} \geq i > p^{s-1}$, then, by Lemma 6, we get $\min\{P_t : \quad t \in T \quad \text{and} \quad t \geq i > j\} = 3$. Note that $d_H(\bar{C}_t) = 1$ by (3), so

$$\min\{P_t d_H(\bar{C}_t) : \quad t \in T \quad \text{and} \quad t \geq i > j\} = 3. \tag{7}$$

Using (4), (6) and (7), we conclude $d_H(C) = 3$. □

Lemma 9. *Let $C = \langle (x-1)^i(x+1)^j \rangle$, where $p^s > i > 2p^{s-1}$ and $p^{s-1} \geq j > 0$. Then $d_H(C) = 4$.*

Proof. By Lemma 6, (2) and (3) we get

$$\min\{P_t d_H(\bar{C}_t) : \quad t \in T \quad \text{and} \quad i > t \geq j\} = 4, \tag{8}$$

$$\min\{d_H(\bar{C}_t) P_t : \quad t \in T \quad \text{and} \quad t \geq i > j\} \geq 4. \tag{9}$$

Using (4), (8) and (9), we obtain $d_H(C) = 4$. □

We consider the case $p^{s-1} + 1 \leq j \leq i \leq (p-1)p^{s-1}$ in the following lemma.

Lemma 10. *Let $j \leq i, 1 \leq \beta' \leq \beta \leq p-2$ be integers with $\beta p^{s-1} + 1 \leq i \leq (\beta+1)p^{s-1}$ and $\beta' p^{s-1} + 1 \leq j \leq (\beta'+1)p^{s-1}$. Let $C = \langle (x-1)^i(x+1)^j \rangle$, then $d_H(C) = \min\{\beta + 2, 2(\beta' + 2)\}$.*

Proof. First suppose that $\beta = \beta'$. Note that if $i > t \geq j$, then $t \notin T$. For $t \geq i \geq j$, by Lemma 6, we get $\min\{P_t : \quad t \in T \quad \text{and} \quad t \geq i \geq j\} = \beta + 2 = \beta' + 2$. For $t \geq i \geq j$, $d_H(\bar{C}_t) = 1$ by (3). So we have

$$\min\{P_t d_H(\bar{C}_t) : \quad t \in T \quad \text{and} \quad t \geq i \geq j\} = \beta + 2 = \beta' + 2. \tag{10}$$

Next we assume $\beta' < \beta$. Then, using Lemma 6, we get $\min\{P_t : \quad t \in T \quad \text{and} \quad i > t \geq j\} = (\beta' + 2)$. Since $i > t \geq j$, $d_H(\bar{C}_t) = 2$ by (2), therefore

$$\min\{P_t d_H(\bar{C}_t) : \quad t \in T \quad \text{and} \quad i > t \geq j\} = 2(\beta' + 2). \tag{11}$$

For $t \geq i \geq j$, using Lemma 6, we get $\min\{P_t : \quad t \in T \quad \text{and} \quad t \geq i \geq j\} = (\beta + 2)$. Since $t \geq i \geq j$, $d_H(\bar{C}_t) = 1$ by (3), thus

$$\min\{P_t d_H(\bar{C}_t) : \quad t \in T \quad \text{and} \quad t \geq i \geq j\} = \beta + 2. \tag{12}$$

Using (4), (10), (11) and (12), we obtain $d_H(C) = \min\{2(\beta' + 2), \beta + 2\}$. □

The following lemma deals with the case $p^{s-1}+1 \leq j < (p-1)p^{s-1} < i \leq p^s-1$.

Lemma 11. *Let $i, j, 1 \leq \tau \leq p-1, 1 \leq \beta \leq p-2$ and $1 \leq k \leq s-1$ be integers such that $p^s - p^{s-k} + (\tau-1)p^{s-k-1} + 1 \leq i \leq p^s - p^{s-k} + \tau p^{s-k-1}$ and $\beta p^{s-1} + 1 \leq j \leq (\beta+1)p^{s-1}$. Let $C = \langle (x-1)^i (x+1)^j \rangle$, then $d_H(C) = 2(\beta+2)$.*

Proof. By Lemma 6, (2), Lemma 7 and (3), we get

$$\min\{P_t d_H(\bar{C}_t) : \quad t \in T \quad \text{and} \quad i > t \geq j\} = 2(\beta+2), \tag{13}$$
$$\min\{P_t d_H(\bar{C}_t) : \quad t \in T \quad \text{and} \quad t \geq i > j\} = (\tau+1)p^k. \tag{14}$$

It follows from $\beta \leq p-2, 1 \leq \tau$ and $1 \leq k$ that $2(\beta+2) \leq 2p \leq (\tau+1)p^k$. So, using (4), (13) and (14), we obtain $d_H(C) = 2(\beta+2)$.. □

In the following two lemmas, we obtain the minimum Hamming distance when $(p-1)p^{s-1} + 1 \leq j \leq i \leq p^s - 1$.

Lemma 12. *Let $j \leq i$, $1 \leq k \leq s-1$, $1 \leq \tau' \leq \tau \leq p-1$ be integers such that*

$$p^s - p^{s-k} + (\tau-1)p^{s-k-1} + 1 \leq i \leq p^s - p^{s-k} + \tau p^{s-k-1} \quad \text{and}$$
$$p^s - p^{s-k} + (\tau'-1)p^{s-k-1} + 1 \leq j \leq p^s - p^{s-k} + \tau' p^{s-k-1}.$$

Let $C = \langle (x-1)^i (x+1)^j \rangle$, then $d_H(C) = \min\{2(\tau'+1)p^k, (\tau+1)p^k\}$.

Proof. First suppose that $\tau = \tau'$. Obviously, there exists no $t \in T$ with $i > t \geq j$. So we consider $t \geq i \geq j$. By Lemma 7 and (3) we get

$$\min\{P_t d_H(\bar{C}_t) : \quad t \in T \quad \text{and} \quad t \geq j \geq i\} = (\tau+1)p^k. \tag{15}$$

Next suppose that $\tau' < \tau$. By Lemma 7, (2) and (3) we get

$$\min\{P_t d_H(\bar{C}_t) : \quad t \in T \quad \text{and} \quad i > t \geq j\} = 2(\tau'+1)p^k, \tag{16}$$
$$\min\{P_t d_H(\bar{C}_t) : \quad t \in T \quad \text{and} \quad t \geq i \geq j\} = (\tau+1)p^k. \tag{17}$$

Using (4), (15), (16) and (17), we obtain $d_H(C) = \min\{2(\tau'+1)p^k, (\tau+1)p^k\}$. □

Lemma 13. *Let i, j, $1 \leq k' < k \leq s-1$, $1 \leq \tau', \tau \leq p-1$ be integers such that $p^s - p^{s-k} + (\tau-1)p^{s-k-1} + 1 \leq i \leq p^s - p^{s-k} + \tau p^{s-k-1}$ and $p^s - p^{s-k'} + (\tau'-1)p^{s-k'-1} + 1 \leq j \leq p^s - p^{s-k'} + \tau' p^{s-k'-1}$. Let $C = \langle (x-1)^i (x+1)^j \rangle$, then $d_H(C) = 2(\tau'+1)p^{k'}$.*

Proof. By Lemma 7, (2) and (3) we get

$$\min\{P_t d_H(\bar{C}_t) : \quad t \in T \quad \text{and} \quad i > t \geq j\} = 2(\tau'+1)p^{k'}, \tag{18}$$
$$\min\{P_t d_H(\bar{C}_t) : \quad t \in T \quad \text{and} \quad t \geq i > i\} = (\tau+1)p^k. \tag{19}$$

It follows from $k > k'$, $\tau + 1 \geq 2$ and $\tau' + 1 \leq p$ that $p^k(\tau+1) \geq 2p^{k'}(\tau'+1)$. So, using (4), (18) and (19) we obtain $d_H(C) = \min\{2(\tau'+1)p^{k'}, (\tau+1)p^k\} = 2(\tau'+1)p^{k'}$. □

Finally it remains to consider the cases $i = p^s$ and $0 < j \leq p^s - 1$. Clearly, for $i = p^s$ and $0 < j \leq p^s - 1$, there is no $t \in T$ with $t \geq i \geq j$, so we only consider $i > t \geq j$ in the following three lemmas. Their proofs follow from Lemma 6, Lemma 7, (2), (4) and similar arguments as above.

Lemma 14. *Let $0 < j \leq p^{s-1}$ be an integer and let $C = \langle (x-1)^{p^s}(x+1)^j \rangle$. Then $d_H(C) = 4$.*

Lemma 15. *Let $1 \leq \beta \leq p-2$, $\beta p^{s-1} + 1 \leq j \leq (\beta+1)p^{s-1}$ be integers. Let $C = \langle (x-1)^{p^s}(x+1)^j \rangle$, then $d_H(C) = 2(\beta+2)$.*

Lemma 16. *Let $1 \leq \tau \leq p-1$, $1 \leq k \leq s-1, j$ be integers such that $p^s - p^{s-k} + (\tau-1)p^{s-k-1} + 1 \leq j \leq p^s - p^{s-k} + \tau p^{s-k-1}$. Let $C = \langle (x-1)^{p^s}(x+1)^j \rangle$, then $d_H(C) = 2(\tau+1)p^k$.*

We summarize our results in the following theorem (see Table 1 also).

Theorem 2. *Let p be an odd prime and let a, s be arbitrary positive integers. For $q = p^a$, all cyclic codes of length $2p^s$ over $\mathbb{F}_q$ are of the form $\langle (x-1)^i(x+1)^j \rangle \subset \mathbb{F}_q[x]/\langle x^{2p^s} - 1 \rangle$ with $0 \leq i, j \leq p^s$. Let $C = \langle (x-1)^i(x+1)^j \rangle \subset \mathbb{F}_q[x]/\langle x^{2p^s} - 1 \rangle$.*

Table 1. The minimum Hamming distance of all non-trivial cyclic codes, of the form $\langle (x-1)^i(x+1)^j \rangle$, of length $2p^s$ over $\mathbb{F}_q$. The parameters $1 \leq \beta' \leq \beta \leq p-2$, $1 \leq \tau^{(2)} < \tau^{(1)} \leq p-1$, $1 \leq \tau, \tau^{(3)}, \tau^{(4)} \leq p-1$, $1 \leq k \leq s-1$, $1 \leq k'' < k' \leq s-1$ are integers. The cases with $i \geq j$ are given. For the cases with $i \leq j$, see Remark 1.

Case	**i**	**j**	$\mathbf{d_H(C)}$
1	$0 < i \leq p^s$	$j = 0$	2
2	$0 \leq i \leq p^{s-1}$	$0 \leq j \leq p^{s-1}$	2
3	$p^{s-1} < i \leq 2p^{s-1}$	$0 < j \leq p^{s-1}$	3
4	$2p^{s-1} < i \leq p^s$	$0 < j \leq p^{s-1}$	4
5	$\beta p^{s-1} + 1 \leq i \leq (\beta+1)p^{s-1}$	$\beta' p^{s-1} + 1 \leq j \leq (\beta' + 1)p^{s-1}$	$\min\{\beta+2, 2(\beta'+2)\}$
6	$p^s - p^{s-k} + (\tau-1)p^{s-k-1} + 1 \leq i \leq p^s - p^{s-k} + \tau p^{s-k-1}$	$\beta p^{s-1} + 1 \leq j \leq (\beta+1)p^{s-1}$	$2(\beta+2)$
7	$p^s - p^{s-k} + (\tau-1)p^{s-k-1} + 1 \leq i \leq p^s - p^{s-k} + \tau p^{s-k-1}$	$p^s - p^{s-k} + (\tau-1)p^{s-k-1} + 1 \leq j \leq p^s - p^{s-k} + \tau p^{s-k-1}$	$(\tau+1)p^k$
8	$p^s - p^{s-k} + (\tau^{(1)} - 1)p^{s-k-1} + 1 \leq i \leq p^s - p^{s-k} + \tau^{(1)} p^{s-k-1}$	$p^s - p^{s-k} + (\tau^{(2)} - 1)p^{s-k-1} + 1 \leq j \leq p^s - p^{s-k} + \tau^{(2)} p^{s-k-1}$	$\min\{2(\tau^{(2)}+1)p^k, (\tau^{(1)}+1)p^k\}$
9	$p^s - p^{s-k'} + (\tau^{(3)} - 1)p^{s-k'-1} + 1 \leq i \leq p^s - p^{s-k'} + \tau^{(3)} p^{s-k'-1}$	$p^s - p^{s-k''} + (\tau^{(4)} - 1)p^{s-k''-1} + 1 \leq j \leq p^s - p^{s-k''} + \tau^{(4)} p^{s-k''-1}$	$2(\tau^{(4)}+1)p^{k''}$
10	$i = p^s$	$\beta p^{s-1} + 1 \leq j \leq (\beta+1)p^{s-1}$	$2(\beta+2)$
11	$i = p^s$	$p^s - p^{s-k} + (\tau-1)p^{s-k-1} + 1 \leq j \leq p^s - p^{s-k} + \tau p^{s-k-1}$	$2(\tau+1)p^k$

If $(i,j) = (0,0)$*, then* C *is the whole space* $\mathbb{F}_q^{2p^s}$*, and if* $(i,j) = (p^s, p^s)$*, then* C *is the zero space* $\{\mathbf{0}\}$*. For the remaining values of* (i,j)*, the minimum Hamming distance of the cyclic code* C *is given in Table 1.*

Remark 1. We simplified Table 1 by giving the cases only with $i \geq j$ for some ranges of i and j. The corresponding case with $j \geq i$ have the same minimum Hamming distance by symmetry. For example in Case 1, the corresponding case is $i = 0$ and $0 \leq j \leq p^s$, and the minimum Hamming distance is 2.

Acknowledgements

The authors would like to thank anonymous referees for their valuable comments. The authors were partially supported by TÜBİTAK under Grant No. TBAG-107T826.

References

1. Castagnoli, G., Massey, J.L., Schoeller, P.A., von Seemann, N.: On repeated-root cyclic codes. IEEE Trans. Inform. Theory 37, 337–342 (1991)
2. Dihn, H.Q.: On the linear ordering of some classes of negacyclic and cyclic codes and their distance distributions. Finite Fields Appl. 14, 22–40 (2008)
3. Lidl, R., Niederreiter, H.: Finite Fields. Cambridge University Press, Cambridge (1997)
4. MacWilliams, F.J., Sloane, N.J.A.: The Theory Of Error Correcting Codes. North Holland, Amsterdam (1978)
5. Özadam, H., Özbudak, F.: A note on negacyclic and cyclic codes of length p^s over a finite field of characteristic p (2009) (submitted)
6. Sălăgean, A.: Repeated-root cyclic and negacyclic codes over a finite chain ring. Discrete Appl. Math. 154(2), 413–419 (2006)
7. López-Permouth, S.R., Szabo, S.: On the Hamming weight of repeated-root cyclic and negacyclic codes over Galois rings (2009) (preprint)
8. van Lint, J.H.: Repeated-root cyclic codes. IEEE Trans. Inform. Theory 37, 343–345 (1991)
9. Zimmerman, K.-H.: On generalizations of repeated-root cyclic codes. IEEE Trans. Inform. Theory 42, 641–649 (1996)

There Are Not Non-obvious Cyclic Affine-invariant Codes*

José Joaquín Bernal, Ángel del Río, and Juan Jacobo Simón

Departamento de Matemáticas, Universidad de Murcia, Spain

Abstract. We show that an affine-invariant code C of length p^m is not permutation equivalent to a cyclic code except in the obvious cases: $m = 1$ or C is either $\{0\}$, the repetition code or its dual.

Affine-invariant codes were firstly introduced by Kasami, Lin and Peterson [KLP2] as a generalization of Reed-Muller codes. This class of codes has received the attention of several authors because of its good algebraic and decoding properties [D, BCh, ChL, Ho, Hu]. It is well known that every affine-invariant code can be seen as an ideal of the group algebra of an elementary abelian group in which the group is identified with the standard base of the ambient space. In particular, if C is a code of prime length then C is permutation equivalent to a cyclic code. Other obvious affine-invariant cyclic codes are the trivial code, $\{0\}$, the repetition code and the code form by all the even-like words, provided its length is a prime power. In this paper we prove that these are the only affine-invariant codes which are permutation equivalent to a cyclic code.

Our main tools are an intrinsical characterization of group codes obtained in [BRS] and a description of the group of permutation automorphisms of non-trivial affine-invariant codes given in [BCh]. These results are reviewed in Section 1, where we also recall the definition and main properties of affine-invariant codes. In Section 2 we prove the main result of the paper.

1 Preliminaries

In this section we recall the definition of (left) group code and the intrinsical characterization given in [BRS]. We also recall the definition of affine-invariant code and the description of its group of permutation automorphisms given in [BCh].

All throughout $\mathbb{F}$ is a field of order a power of p, where p is a prime number. The finite field with p^s elements is denoted by $\mathbb{F}_{p^s}$. For a group G, we denote by $\mathbb{F}G$ the group ring of G with coefficients in $\mathbb{F}$. All the group theoretical notions used in this paper can be easily founded in [R].

Definition 1. *If E is the standard basis of $\mathbb{F}^n$, $C \subseteq \mathbb{F}^n$ is a linear code and G is a group (of order n) then we say that C is a* G-code *if there is a bijection*

* Research supported by D.G.I. of Spain and Fundación Séneca of Murcia.

M. Bras-Amorós and T. Høholdt (Eds.): AAECC 2009, LNCS 5527, pp. 101–106, 2009.

$\phi : E \to G$ such that the linear extension of ϕ to an isomorphism $\phi : \mathbb{F}^n \to \mathbb{F}G$ maps C to an ideal of $\mathbb{F}G$.

A group code *is a linear code which is a G-code for some group G.*

A cyclic group code (respectively, abelian group code, solvable group code, etc) is a linear code which is G-code for some cyclic group G (respectively, abelian group, solvable group, etc).

Let S_n denote the group of permutations of n symbols. Every $\sigma \in S_n$ defines an automorphism of $\mathbb{F}^n$ in the obvious way, i.e. $\sigma(x_1, \ldots, x_n) = (x_{\sigma^{-1}(1)}, \ldots, x_{\sigma^{-1}(n)})$. By definition, the group of permutation automorphisms of a linear code C of length n is

$$\mathrm{PAut}(C) = \{\sigma \in S_n : \sigma(C) = C\}.$$

An intrinsical characterization of group codes C in terms of $\mathrm{PAut}(C)$ has been obtained in [BRS]. Four our purposes we only need to consider abelian groups. So we record in the following theorem the specialization of [BRS, Theorem 1.2] to abelian groups.

Theorem 1. *[BRS] Let C be a linear code of length n over a field $\mathbb{F}$ and G a finite abelian group of order n. Then C is a G-code if and only if G is isomorphic to a transitive subgroup H of S_n.*

In the remainder of the paper we assume that $\mathcal{I} = \mathbb{F}_{p^m}$. Often we will be only using the underlying additive structure of $\mathcal{I}$; for example, $\mathbb{F}\mathcal{I}$ is the group algebra of this additive group with coefficients in $\mathbb{F}$. Let $S(\mathcal{I})$ denote the group of permutations of $\mathcal{I}$. Every element of $S(\mathcal{I})$ induces a unique $\mathbb{F}$-linear bijection of the group algebra $\mathbb{F}\mathcal{I}$. For an $\mathbb{F}$-subspace C of $\mathbb{F}\mathcal{I}$, let $\mathrm{PAut}(C) = \{\sigma \in S(\mathcal{I}) : \sigma(C) = C\}$.

An affine-invariant code is an $\mathbb{F}$-subspace C of $\mathbb{F}\mathcal{I}$ formed by even-like words such that $\mathrm{PAut}(C)$ contains the maps of the form $x \in \mathcal{I} \mapsto \alpha x + \beta$, with $\alpha \in \mathcal{I}^* = \{a \in \mathcal{I} : a \neq 0\}$ and $\beta \in \mathcal{I}$. These maps are called affine transformations of $\mathcal{I}$.

Observe that if $\mathbb{F}^{p^m}$ is identified with $\mathbb{F}\mathcal{I}$ via some bijection from $\{1, \ldots, p^m\}$ to $\mathcal{I}$, then the linear codes of length p^m correspond to subspaces of $\mathbb{F}\mathcal{I}$ in such a way that the groups of permutations automorphisms agree. Therefore if C is a subspace of $\mathbb{F}\mathcal{I}$ and G is a group then C is a left G-code if and only if $\mathrm{PAut}(C)$ contains a regular subgroup H of $S(\mathcal{I})$ isomorphic to G and it is a G-code if H can be selected such that $C_{S(\mathcal{I})}(H) \subseteq \mathrm{PAut}(C)$.

Affine-invariant codes can be seen as extended cyclic codes. Recall that a cyclic code C of length n over $\mathbb{F}$ is a subspace of $\mathbb{F}^n$ which is closed under cyclic permutations, that is if $(x_1, x_2, \ldots, x_{n-1}, x_n)$ is an element of C then so is $(x_n, x_1, x_2, \ldots, x_{n-1})$. Cyclic codes are cyclic group codes via the bijection $\phi : E \to G$ given by $\phi(e_i) = g^{i-1}$, where $E = \{e_1, \ldots, e_n\}$ is the standard basis of $\mathbb{F}^n$ and G is a cyclic group of order n generated by g. Conversely, any ideal of the group algebra $\mathbb{F}G$, with G a cyclic group of order n can be seen as a cyclic code with a suitable identification of the elements of G with the coordinates.

The zeroes of a cyclic code C of length n are the n-th roots of unity α such that $x_0 + x_1\alpha + x_2\alpha^2 + \cdots + x_{n-1}\alpha^{n-1} = 0$, for every $(x_0, x_1, \ldots, x_{n-1}) \in C$.

It is well known that every cyclic code is uniquely determined by its zeroes and conversely, if ζ is a primitive n-th root of unity and D is a union of q-cyclotomic classes modulo n then there is a unique q-ary cyclic code of length n whose set of zeroes is $\{\zeta^i : i \in D\}$.

Let $C \subseteq \mathbb{F}\mathcal{I}$ be an affine-invariant code and let C^* denote the code obtained by puncturing C at the coordinate labelled by 0. The permutation automorphisms of C which fix 0 induces permutation automorphisms of C^*. In particular, $\mathrm{PAut}(C^*)$ contains the maps of the form $x \to \alpha x$, for $\alpha \in \mathcal{I}^*$. This maps form a cyclic group isomorphic to the group of units of the field $\mathcal{I}$. So, C^* is a cyclic group code and C is the extended code obtained by adding a parity check coordinate.

However not every code obtained by extending a cyclic code of length $p^m - 1$ is affine-invariant. We recall a characterization of Kasami, Lin and Peterson of the extended cyclic codes which are affine-invariant in terms of the roots of the cyclic code [KLP1].

The p-adic expansion of a non-negative integer x is the list of integers $(x_0, x_1, \dots)$ with $0 \leq x_i < p$ and $x = \sum_{i \geq 0} x_i p^i$. The p-adic expansion yields a partial ordering in the set of positive integers by setting $x \preceq y$ if $x_i \leq y_i$, for every i, where (x_i) and (y_i) are the p-adic expansions of x and y, respectively.

Let $n = p^m - 1$ and let α be a primitive element of $\mathcal{I}$, i.e. a generator of $\mathcal{I}^*$. Identify the standard basis of $\mathbb{F}^n$, $E = \{e_1, \dots, e_n\}$, with $\mathcal{I}^*$ via the map $e_i \mapsto \alpha^{i-1}$. A cyclic code C^* of length n is determined by the following set

$$D_{C^*} = \{i : 0 \leq i < n, \alpha^i \text{ is a zero of } C^*\}.$$

Since C^* is a q-ary cyclic code, with $q = p^r$, the set D_{C^*} is invariant under multiplication by q modulo n, that is, D_{C^*} is a union of q-cyclotomic classes modulo n. Conversely, every union D of q-cyclotomic classes modulo n, yields to a uniquely defined cyclic code C^* of length n with $D = D_{C^*}$. If C^* is a cyclic code and C is its corresponding extended code then the defining set of C is by definition $D_C = D_{C^*} \cup \{0\}$ if $0 \notin D_C$, and $D_C = D_{C^*} \cup \{n\}$ if $0 \in D_{C^*}$.

Proposition 1. *[KLP1][Hu, Corollary 3.5] Let C^* be a cyclic code of length $n = p^m - 1$ and C the extended code of C^*. Then C is affine-invariant if and only if D_C satisfy the following condition for every $1 \leq s, t \leq n$:*

$$s \preceq t \text{ and } t \in D_C \quad \Rightarrow \quad s \in D_C. \tag{1}$$

The trivial code and the repetition code of length n are cyclic with defining sets $\{0, 1, 2, \dots, n-1\}$ and $\{1, 2, \dots, n-1\}$ respectively. Their duals, i.e. $\mathbb{F}^n$ and the space of all the even-like words, are also cyclic with defining sets $\emptyset$ and $\{0\}$. Recall that a word $(x_1, \dots, x_n)$ is even-like if $\sum_{i=1}^n x_i = 0$. When $n = p^m - 1$, except for the last one, all the others give rise to affine-invariant codes of length p^m: the trivial code, the repetition code and the code formed by the even-like words. These three codes are known as the trivial affine-invariant codes.

For future use we describe the affine-invariant codes of length 4.

Example 1 (Affine-invariant codes of length 4). Let D be the defining set of an affine-invariant code C of length 4 over $\mathbb{F}_{2^r}$. Then C is trivial as an affine-invariant code if and only if D is either $\{0\}, \{0,1,2\}$ or $\{0,1,2,3\}$. Since D is invariant by multiplication by 2^r modulo 3 and satisfies condition (1), if r is odd then there are not non trivial affine-invariant codes. However, if r is even then there are two non-trivial affine-invariant codes with defining sets $\{0,1\}$ and $\{0,2\}$ respectively.

If C is a trivial affine-invariant code then $\mathrm{PAut}(C) = S_n$, and therefore C is G-code for every group G of order p^m. So to avoid trivialities, in the remainder of the paper all the affine-invariant codes are suppose to be non-trivial. The group of permutations of a (non-trivial) affine-invariant code has been described by Berger and Charpin [BCh]. In order to present their description we need to introduce some notation.

We use the notation $N \rtimes G$ to represent a semidirect product of N by G via some action of G on N, which is going to be clear from the context. That is, N and G are groups and there is a group homomorphism $\sigma : G \to \mathrm{Aut}(N)$ associating $g \in G$ to σ_g. The underlying set of $N \rtimes G$ is the direct product $N \times G$ and the product is given by $(n_1, g_1)(n_2, g_2) = (n_1 \sigma_{g_1}(n_2), g_1 g_2)$.

For every $d|m$ let $\mathrm{GL}(\mathcal{I}_{\mathbb{F}_{p^d}})$ and $\mathrm{Aff}_d(\mathcal{I})$ denote the groups of linear and affine transformations of $\mathcal{I}$ as vector space over $\mathbb{F}_{p^d}$. The group of $\mathbb{F}_{p^d}$-automorphisms of the field $\mathcal{I}$ is denoted by $\mathrm{Gal}(\mathcal{I}/\mathbb{F}_{p^d})$. We identify every element $y \in \mathcal{I}$ with the translation $x \mapsto x + y$. Then $\mathrm{Aff}_d(\mathcal{I}) = \mathcal{I} \rtimes \mathrm{GL}(\mathcal{I}_{\mathbb{F}_{p^d}})$.

Given two divisors a and b of m with $b|a$, let

$$\mathcal{G}_{a,b} = \{f \in \mathrm{GL}(\mathcal{I}_{\mathbb{F}_{p^b}}) : f \text{ is } \tau - \text{semilinear for some } \tau \in \mathrm{Gal}(\mathbb{F}_{p^a}/\mathbb{F}_{p^b})\}.$$

We claim that $\mathcal{G}_{a,b} = \langle \mathrm{GL}(\mathcal{I}_{\mathbb{F}_{p^a}}), \mathrm{Gal}(\mathcal{I}/\mathbb{F}_{p^b})\rangle$. Indeed, if f is τ-semilinear with $\tau \in \mathrm{Gal}(\mathbb{F}_{p^a}/\mathbb{F}_{p^b})$ then τ is the restriction of σ for some $\sigma \in \mathrm{Gal}(\mathcal{I}/\mathbb{F}_{p^b})$ and $f\sigma^{-1} \in \mathrm{GL}(\mathcal{I}_{\mathbb{F}_{p^a}})$.

Theorem 2. *[BCh, Corollary 2] Let C be a non-trivial affine-invariant code of length p^m over $\mathbb{F}_q$, with $q = p^r$. Let*

$$a = a(C) = \min\{d|m : \mathrm{Aff}_d(\mathcal{I}) \subseteq \mathrm{PAut}(C)\},$$
$$b = b(C) = \min\{d \geq 1 : p^d D_C = D_C\}$$

Then $b|r$, $b|a|m$ and

$$\mathrm{PAut}(C) = \langle \mathrm{Aff}_a(\mathcal{I}), \mathrm{Gal}(\mathcal{I}/\mathbb{F}_{p^b})\rangle = \mathcal{I} \rtimes \mathcal{G}_{a,b}.$$

A method to compute $a(C)$ and $b(C)$ was firstly obtained by Delsarte [D]. Later, Berger and Charpin founded two alternative methods to calculate $a(C)$ and $b(C)$ which are sometimes computationally simpler [BCh].

2 Affine-invariant Cyclic Group Codes

Let C be an affine-invariant code. Then C is an $\mathcal{I}$-code, since the group of translations of $\mathcal{I}$ (which we have identified with the additive group $\mathcal{I}$) is a transitive

subgroup of $S(\mathcal{I})$ contained in $\mathrm{PAut}(C)$. So every affine-invariant code is an elementary abelian group code. In particular, if the length of C is prime then C is a cyclic group code. Next result shows that this one is the only type of non-trivial affine-invariant cyclic group codes.

Theorem 3. *A non-trivial affine-invariant code is permutation equivalent to a cyclic code if and only if it has prime length.*

Proof. Assume that C is a non trivial affine-invariant code of length p^m which is permutation equivalent to a cyclic code. Then C is a cyclic group code. By Theorem 1, this implies that $G = \mathrm{PAut}(C)$ contains a cyclic subgroup of order p^m or equivalently G contains an element of order p^m. Let $a = a(C)$, $b = b(C)$, as in Theorem 2 and $h = m/a$. Let p^t be the maximum p-th power dividing a/b, and p^u the minimum p-th power greater or equal than h. We first show that the existence of an element g of order p^m in G implies a strong relation on these parameters which reduces to some few cases.

By Theorem 2, $G = \mathcal{I} \rtimes \mathcal{G}_{a,b}$. Furthermore $H = \mathcal{I} \rtimes \mathrm{GL}(\mathcal{I}_{\mathbb{F}_{p^a}})$ is a normal subgroup of index a/b in G. Since the order of G is a p-th power and p^t is the maximum p-th power dividing a/b, g^{p^t} is an element of order p^{m-t} in H. Furthermore, H is isomorphic to $\mathbb{F}_{p^a}^h \rtimes \mathrm{GL}_h(\mathbb{F}_{p^a})$, where $\mathrm{GL}_h(\mathbb{F}_{p^a})$ is the group of invertible $h \times h$ matrices with entries in $\mathbb{F}_{p^a}$. Therefore there is an element $(x, A) \in \mathbb{F}_{p^a}^h \rtimes \mathrm{GL}_h(\mathbb{F}_{p^a})$ of order p^{m-t}. This implies that A has order p^k with $m-t-1 \leq k$. Since the order of A is a power of p, and the fields of characteristic p do not have elements of orden p, the only eigenvalue of A is 1. We may assume that A is given in Jordan form and hence $A = I + N$ where N is an upper triangular matrix with zeroes in the diagonal. Then $N^{p^u} = 0$ and therefore $A^{p^u} = I$. Thus $m - t - 1 \leq k \leq u = \lceil \log_p(h) \rceil$ and we conclude that

$$m \leq 1 + t + \lceil \log_p(h) \rceil < 2 + \log_p(a) + \log_p(h) = 2 + \log_p(m). \tag{2}$$

We have to prove that $m = 1$. Otherwise, (2) implies that either $m = 2$ or $m = 3$ and $p = 2$.

Case 1: $m = 2$. We claim that in this case $p = 2$. Indeed, if $p > 2$ then $0 < \log_p(2) < 1$ and $t = 0$, since p^t divides $m = 2$. Hence $2 = m \leq 1 + \lceil \log_p(h) \rceil \leq 1 + \lceil \log_p(m) \rceil = 2$ and so $h = 2$.

Hence (x, A) is an element of order p^2 in $\mathbb{F}_p^2 \rtimes \mathrm{GL}_2(\mathbb{F}_p)$. Thus there are $x_1, x_2, y \in \mathbb{F}_p$ such that $((x_1, x_2), A)^p \neq (0, 1)$, where $A = \begin{pmatrix} 1 & y \\ 0 & 1 \end{pmatrix}$. However

$$\begin{aligned} ((x_1, x_2), A)^p &= \left(\left(\textstyle\sum_{i=0}^{p-1} (x_1 + iyx_2), py \right), A^p \right) \\ &= \left(\left(\frac{p(p-1)}{2} yx_2, 0 \right), 1 \right) = (0, 1), \end{aligned}$$

which is the desired contradiction.

Hence $p = 2$ and so C has length 4. Since C is non-trivial as affine-invariant code, by Example 1, r is even and $D_C = \{0, 1\}$ or $\{0, 2\}$. By the definition of $b(C)$ we have $b = 2$ and hence $a = 2$. We deduce that $\mathrm{PAut}(C) = \mathcal{I} \rtimes \mathbb{F}_4^*$. Hence

$\mathcal{I} \simeq \mathbb{F}_p^2$ is the only subgroup of order 4 of $\mathrm{PAut}(C)$ and we conclude that C is not cyclic group code.

Case 2: $m = 3$ and $p = 2$. Then $t = 0$ and $3 \leq 1 + \lceil \log_2(h) \rceil \leq 1 + \lceil \log_2(m) \rceil = 3$, by (2). Thus $h = 3$, or equivalently $a = 1$ and hence $b = 1$. Thus (x, A) is an element of order 8 in $\mathbb{F}_2^3 \rtimes GL_3(\mathbb{F}_2)$. Then $u = \lceil \log_2(3) \rceil = 2$ and so $A^4 = 1$. If $A^2 = 1$ then $x^4 = 1$, a contradiction. Therefore A is a Jordan matrix of order 4 and thus

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

However $I + A + A^2 + A^3 = 0$. Then $(x, A)^4 = ((I + A + A^2 + A^3)x, A^4) = (0, 1)$, a contradiction. □

References

[BCh] Berger, T.P., Charpin, P.: The Permutation group of affine-invariant extended cyclic codes. IEEE Trans. Inform. Theory 42, 2194–2209 (1996)

[BRS] Bernal, J.J., del Río, Á., Simón, J.J.: An intrinsical description of group codes. Designs, Codes, Cryptog. (to appear), doi:10.1007/s10623-008-9261-z

[ChL] Charpin, P., Levy-Dit-Vehel, F.: On Self-dual affine-invariant codes, J. Comb. Theory, Series A 67, 223–244 (1994)

[D] Delsarte, P.: On cyclic codes that are invariant under the general linear group. IEEE Trans. Inform. Theory IT-16, 760–769 (1970)

[Ho] Hou, X.-D.: Enumeration of certain affine invariant extended cyclic codes. J. Comb. Theory, Series A 110, 71–95 (2005)

[Hu] Huffman, W.C.: Codes and groups. In: Pless, V.S., Huffman, W.C., Brualdi, R.A. (eds.) Handbook of coding theory, vol. II, pp. 1345–1440. North-Holland, Amsterdam (1998)

[KLP1] Kasami, T., Lin, S., Peterson, W.W.: Some results on cyclic codes which are invariant under the affine group and their applications. Information and Control 11, 475–496 (1967)

[KLP2] Kasami, T., Lin, S., Peterson, W.W.: New generalizations of the Reed-Muller codes part I: primitive codes. IEEE Trans. Inform. Theory IT-14, 189–199 (1968)

[R] Robinson, D.J.S.: A course in the theory of groups. Springer, Heidelberg (1996)

On Self-dual Codes over $\mathbf{Z}_{16}$

Kiyoshi Nagata*, Fidel Nemenzo, and Hideo Wada

Department of Business Studies and Informatics, Daito Bunka University, Tokyo, Japan
nagata@ic.daito.ac.jp
Institute of Mathematics, University of the Philippines, Quezon City, Philippines
fidel@math.upd.edu.ph
Faculty of Science and Technology, Sophia University, Tokyo, Japan
wada@mm.sophia.ac.jp

Abstract. In this paper, we look at self-dual codes over the ring $\mathbf{Z}_{16}$ of integers modulo 16. From any doubly even self-dual binary code, we construct codes over $\mathbf{Z}_{16}$ and give a necessary and sufficient condition for the self-duality of induced codes. We then give an inductive algorithm for constructing all self-dual codes over $\mathbf{Z}_{16}$, and establish the mass formula, which counts the number of such codes.

Keywords: Self-dual codes, Doubly even codes, Finite rings, Mass formula.

1 Introduction

There has been much interest in codes over the ring $\mathbf{Z}_m$ of integers modulo m and finite rings in general, since the discovery [5] of a relationship between non-linear binary codes and linear quaternary codes. By applying the Chinese Remainder Theorem [2] to self-dual codes over $\mathbf{Z}_m$, it suffices to look at codes over integers modulo prime powers.

We consider the problem of finding the mass formula, which counts the number of self-dual codes of given length. In [4], Gaborit gave a mass formula for self-dual quaternary codes. This was generalized in [1] to the ring $\mathbf{Z}_{p^2}$ and in [6] to $\mathbf{Z}_{p^3}$ for all primes p. In [7], we gave a mass formula over $\mathbf{Z}_{p^s}$ in [7] for any odd prime p and any integer $s \geq 4$. In this paper, we will give an inductive algorithm for constructing self-dual codes over $\mathbf{Z}_{16}$ from a given binary code. As a consequence, we obtain a mass formula for such codes.

A *code* of length n over a finite ring R is an R-submodule of R^n. Elements of codes are called *codewords*. Associated to a code $\mathcal{C}$ is a generator matrix, whose rows span $\mathcal{C}$ and are linearly independent in binary case. Two codewords $\boldsymbol{x} = (x_1, \ldots, x_n)$ and $\boldsymbol{y} = (y_1, \ldots, y_n)$ (considered as vectors) are *orthogonal* if their Euclidean inner product $\boldsymbol{x} \cdot \boldsymbol{y} = \sum_i x_i y_i$ is zero. The *dual* $\mathcal{C}^\perp$ of a code $\mathcal{C}$ over a ring R consists of all vectors of R^n which are orthogonal to every codeword in $\mathcal{C}$. A code $\mathcal{C}$ is said to be *self-dual* (resp. *self-orthogonal*) if $\mathcal{C} = \mathcal{C}^\perp$ (resp. $\mathcal{C} \subseteq \mathcal{C}^\perp$).

* Corresponding author.

M. Bras-Amorós and T. Høholdt (Eds.): AAECC 2009, LNCS 5527, pp. 107–116, 2009.

2 Construction of Self-dual Codes over $\mathbf{Z_{16}}$

Every code $\mathcal{C}$ of length n over $\mathbf{Z}_{16}$ has a generator matrix which, after a suitable permutation of coordinates, can be written as

$$\mathcal{C} = \begin{bmatrix} T_1 \\ 2T_2 \\ 2^2T_3 \\ 2^3T_4 \end{bmatrix} = \begin{bmatrix} I_{k_1} & A_{11} & A_{12} & A_{13} & A_{14} \\ 0 & 2I_{k_2} & 2A_{22} & 2A_{23} & 2A_{24} \\ 0 & 0 & 2^2I_{k_3} & 2^2A_{33} & 2^2A_{34} \\ 0 & 0 & 0 & 2^3I_{k_4} & 2^3A_{44} \end{bmatrix},$$

where I_{k_i} is the $k_i \times k_i$ identity matrix, and the other matrices A_{ij}'s ($1 \le i \le j \le 4$) are considered modulo 2^{j-i+1}. The columns are grouped in blocks of sizes k_1, k_2, k_3, k_4 and k_5, with $n = k_1 + k_2 + k_3 + k_4 + k_5$. In [7] we showed that $\mathcal{C}$ is self-dual if and only if $k_1 = k_5$, $k_2 = k_4$ and $T_iT_j^t \equiv 0 \pmod{2^{6-i-j}}$ for i and j such that $1 \le i \le j \le 4$ and $i + j \le 5$.

We showed in [7] that self-dual codes over the ring $\mathbf{Z}_{p^s}$ are induced from codes over the ring $\mathbf{Z}_{p^{s-2}}$, for any odd prime p and integer $s \ge 4$. In this paper we look at the case when $p = 2$ and show that self-dual codes over $\mathbf{Z}_{2^4}$ can be constructed from a code over $\mathbf{Z}_{2^2}$ represented by a generator matrix

$$\begin{bmatrix} \mathbf{T}_1 \\ T_2 \\ 2T_3 \end{bmatrix} \pmod{2^2},$$

where $\mathbf{T}_1$ has some generator vectors $\boldsymbol{t}_1, ..., \boldsymbol{t}_{k_1}$ such that $\boldsymbol{t}_i \cdot \boldsymbol{t}_i \equiv 0 \pmod{2^3}$ for all $i = 1, ..., k_1$. Such a representation of a quaternary code is called T_1-doubly even. We thus construct self-dual codes over $\mathbf{Z}_{2^4}$ from doubly even self-dual binary codes via the following constructive scheme diagram

$$\begin{bmatrix} T_1 \\ 2T_2 \\ 2^2T_3 \\ 2^3T_4 \end{bmatrix} \pmod{2^4} \leftarrow \begin{bmatrix} \mathbf{T}_1 \\ T_2 \\ 2T_3 \end{bmatrix} \pmod{2^2} \leftarrow \begin{bmatrix} \mathbf{T}_1 \\ \mathbf{T}_2 \end{bmatrix} \pmod{2},$$

where T_i's are composed of linearly independent binary codewords as row vectors and $[T_i]$ denotes the code generated by T_i. In the right of the diagram above, the boldface symbols denote doubly even binary codes. The boldface part in the middle of the diagram, $\mathbf{T}_1$, denotes the T_1-doubly even representation.

In [4] Gaborit gave the number of doubly even self-dual binary codes of size n and dimension $k_1 + k_2$, and from this we obtain a mass formula for counting the number of T_1-doubly even representations of quaternary codes induced from a given doubly even self-dual binary code.

We start with a doubly even self-dual binary code $\mathcal{C}_0$ of size n and of dimension $k_1 + k_2$, and divide it into two subspaces of dimension k_1 and k_2. The number of such divisions is $\rho(k_1 + k_2, k_1) = \prod_{i=0}^{k_1-1} \frac{2^{k_1+k_2-i} - 1}{2^{k_1-i} - 1}$. The code $\mathcal{C}_0$ can be represented as

$$\mathcal{C}_0 = \begin{bmatrix} T_1^{(0)} \\ T_2^{(0)} \end{bmatrix} = \begin{bmatrix} I_{k_1} & A_{11}^{(0)} & A_{12}^{(0)} & A_{13}^{(0)} & A_{14}^{(0)} \\ 0 & I_{k_2} & A_{22}^{(0)} & A_{23}^{(0)} & A_{24}^{(0)} \end{bmatrix},$$

where the $A_*^{(0)}$ are considered modulo 2.

Though $A_{11}^{(0)}$ is supposed to be 0 in a usual binary code basis expression, we should remark that some $A_{11}^{(0)}$ is necessary, since we consider it mod 2 in the lifted quaternary expression. These differences are counted in $\rho(k_1+k_2, k_1)$. By a suitable permutation of columns, we can assume that the binary square matrix $\begin{pmatrix} A_{13}^{(0)} & A_{14}^{(0)} \\ A_{23}^{(0)} & A_{24}^{(0)} \end{pmatrix}$ and $A_{14}^{(0)}$ are invertible.

Next we construct a quaternary code $\mathcal{C}_1$, represented by

$$\mathcal{C}_1 = \begin{bmatrix} T_1 \\ T_2 \\ 2T_3 \end{bmatrix} = \begin{bmatrix} I_{k_1} & A_{11}^{(0)} & A_{12}^{(0)} + 2A_{12}^{(1)} & A_{13}^{(0)} + 2A_{13}^{(1)} & A_{14}^{(0)} + 2A_{14}^{(1)} \\ 0 & I_{k_2} & A_{22}^{(0)} & A_{23}^{(0)} + 2A_{23}^{(1)} & A_{24}^{(0)} + 2A_{24}^{(1)} \\ 0 & 0 & 2I_{k_3} & 2A_{33}^{(0)} & 2A_{34}^{(0)} \end{bmatrix}.$$

Here we notice that $A_{12}^{(1)}$ is necessary, since $2I_{k_3}$ becomes $2^2 I_{k_3}$ in the extended code over $\mathbf{Z}_{2^4}$.

There are two conditions for $\mathcal{C}_1$ to be self-dual, and the first is

$$\begin{bmatrix} T_1 \\ T_2 \end{bmatrix} T_3^t \equiv 0 \pmod 2.$$

This implies that $\begin{pmatrix} A_{12}^{(0)} \\ A_{22}^{(0)} \end{pmatrix} I_{k_3} \equiv \begin{pmatrix} A_{13}^{(0)} & A_{14}^{(0)} \\ A_{23}^{(0)} & A_{24}^{(0)} \end{pmatrix} \begin{pmatrix} A_{33}^{(0)} & A_{34}^{(0)} \end{pmatrix}^t \pmod 2$, thus $\begin{pmatrix} A_{33}^{(0)} & A_{34}^{(0)} \end{pmatrix}$ is uniquely determined as $\left(\begin{pmatrix} A_{13}^{(0)} & A_{14}^{(0)} \\ A_{23}^{(0)} & A_{24}^{(0)} \end{pmatrix}^{-1} \begin{pmatrix} A_{12}^{(0)} \\ A_{22}^{(0)} \end{pmatrix} \right)^t$.

Let $\begin{bmatrix} T_1 \\ T_2 \end{bmatrix} = \begin{bmatrix} T_1^{(0)} \\ T_2^{(0)} \end{bmatrix} + 2 \begin{bmatrix} T_1^{(1)} \\ T_2^{(1)} \end{bmatrix}$ where $\begin{bmatrix} T_1^{(0)} \\ T_2^{(0)} \end{bmatrix}$ is a given doubly even self-dual binary code and $\begin{bmatrix} T_1^{(1)} \\ T_2^{(1)} \end{bmatrix} = \begin{bmatrix} 0\,0 & A_{12}^{(1)} & A_{13}^{(1)} & A_{14}^{(1)} \\ 0\,0 & 0 & A_{23}^{(1)} & A_{24}^{(1)} \end{bmatrix}$ is a binary code.

Then the self-dual quaternary code has a T_1-doubly even representation if

$$\begin{bmatrix} T_1^{(0)} \\ T_2^{(0)} \end{bmatrix} \begin{bmatrix} T_1^{(0)} \\ T_2^{(0)} \end{bmatrix}^t + 2 \widetilde{\begin{bmatrix} T_1^{(0)} \\ T_2^{(0)} \end{bmatrix} \begin{bmatrix} T_1^{(1)} \\ T_2^{(1)} \end{bmatrix}^t} + 2^2 \begin{bmatrix} T_1^{(1)} \\ T_2^{(1)} \end{bmatrix} \begin{bmatrix} T_1^{(1)} \\ T_2^{(1)} \end{bmatrix}^t \equiv 0 \qquad (1)$$

where $\widetilde{X} := X + X^t$ and the modulus of congruence is 8 only for the diagonal entries and 4 otherwise.

We describe some parts of $T_i^{(j)}$'s using binary row vectors such as

$$\begin{pmatrix} A_{12}^{(0)} & A_{13}^{(0)} & A_{14}^{(0)} \\ A_{22}^{(0)} & A_{23}^{(0)} & A_{24}^{(0)} \end{pmatrix} = \begin{pmatrix} \boldsymbol{a}_1 \\ \vdots \\ \boldsymbol{a}_{k'} \end{pmatrix} = \begin{pmatrix} * & \boldsymbol{a}'_1 \\ \vdots & \vdots \\ * & \boldsymbol{a}'_{k'} \end{pmatrix}, \quad \begin{pmatrix} A_{12}^{(1)} & A_{13}^{(1)} & A_{14}^{(1)} \\ 0 & A_{23}^{(1)} & A_{24}^{(1)} \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_{k'} \end{pmatrix},$$

where $\boldsymbol{a}$'s and $\boldsymbol{x}$'s are of size $k_3 + k_2 + k_1 = k_3 + k'$, $\boldsymbol{a}'$'s are of size $k_2 + k_1 = k'$. Put $\boldsymbol{x}_i = (\boldsymbol{0}\ \boldsymbol{x}'_i)$ for $k_1 + 1 \le i \le k'$ with $\boldsymbol{x}'_i$ of size k'.

Dividing the left hand side of congruence (1) by 2, we get

$$(f_{ij}) + (\boldsymbol{a}_i \cdot \boldsymbol{x}_j + \boldsymbol{a}_j \cdot \boldsymbol{x}_i) + 2(\boldsymbol{x}_i \cdot \boldsymbol{x}_j),$$

with $\begin{pmatrix} T_1^{(0)} \\ T_2^{(0)} \end{pmatrix} \begin{pmatrix} T_1^{(0)} \\ T_2^{(0)} \end{pmatrix}^t = (2f_{ij})$. Thus the self-dual quaternary code $\mathcal{C}_1$ has a T_1-doubly even representation if

$$\begin{cases} f_{ij} \equiv \boldsymbol{a}_i \cdot \boldsymbol{x}_j + \boldsymbol{a}_j \cdot \boldsymbol{x}_i & (i < j) \\ \frac{1}{2} f_{ii} \equiv \boldsymbol{a}_i \cdot \boldsymbol{x}_i + \boldsymbol{x}_i \cdot \boldsymbol{x}_i = (\boldsymbol{a}_i + \boldsymbol{1}) \cdot \boldsymbol{x}_i & (i = 1, ..., k_1) \end{cases}$$

where the congruence modulus is 2, and $\boldsymbol{1} = \boldsymbol{1}_{k_3+k'}$ denotes the all-1 vector of size $k_3 + k'$. The first condition together with the doubly even assumption for $\mathcal{C}_0$ is a necessary and sufficient condition for the self-duality of the quaternary code $\mathcal{C}_1$, while the second condition is for the T_1-doubly even presentation of $\mathcal{C}_1$.

We now consider the equation in the following matrix representation. To illustrate, we give such matrix representation for the case when $k_1 = 2$ and $k_2 = 1$. We have

$$\begin{pmatrix} \boldsymbol{a}_1 + \boldsymbol{1} & & & \\ & \boldsymbol{a}_2 + \boldsymbol{1} & & \\ \boldsymbol{a}_2 & \boldsymbol{a}_1 & & \\ \boldsymbol{a}_3 & & \boldsymbol{a}'_1 & \\ & \boldsymbol{a}_3 & \boldsymbol{a}'_2 & \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_1^t \\ \boldsymbol{x}_2^t \\ \boldsymbol{x}_3'^t \end{pmatrix} \equiv \begin{pmatrix} g_1 \\ g_2 \\ f_{12} \\ f_{13} \\ f_{23} \end{pmatrix},$$

where $g_i := \frac{1}{2} f_{ii}$ $(i = 1, 2)$ and the modulus of congruence is 2.

In general, if $\boldsymbol{M}$ is the coefficient matrix in the equation, then the number of row vectors in $\boldsymbol{M}$ is $\frac{1}{2}k'(k'-1) + k_1$ with the size of each row equal to $(k' + k_3) \times k_1 + k' \times k_2 = k'^2 + k_1 k_3$.

We compute the rank of $\boldsymbol{M}$. From the definition, the ranks of $(\boldsymbol{a}'_i)$ and $(\boldsymbol{a}_i)$ are both equal to k'. Any linearly dependent equation of row vectors in $\boldsymbol{M}$ must not contain any $\boldsymbol{a}'_i$'s and must contain at least one $\boldsymbol{a}_i + \boldsymbol{1}$. Thus we can assume that

$$\boldsymbol{a}_{i_1} + \cdots + \boldsymbol{a}_{i_m} \equiv \boldsymbol{1} \pmod 2$$

for some $1 \le i_1 < \cdots < i_m \le k_1$.

Going back to the starting code $\mathcal{C}_0$, we write it as

$$\mathcal{C}_0 = \begin{pmatrix} & & \boldsymbol{a}_1 \\ I_{k_1} & A_{11}^{(0)} & \vdots \\ & & \boldsymbol{a}_{k_1} \\ & & \boldsymbol{a}_{k_1+1} \\ 0 & I_{k_2} & \vdots \\ & & \boldsymbol{a}_{k'} \end{pmatrix} = \begin{pmatrix} \boldsymbol{e}_1 & \boldsymbol{a}_1^{(0)} & \boldsymbol{a}_1 \\ \vdots & \vdots & \vdots \\ \boldsymbol{e}_{k_1} & \boldsymbol{a}_{k_1}^{(0)} & \boldsymbol{a}_{k_1} \\ \boldsymbol{0} & \boldsymbol{e}_1 & \boldsymbol{a}_{k_1+1} \\ \vdots & \vdots & \vdots \\ \boldsymbol{0} & \boldsymbol{e}_{k_2} & \boldsymbol{a}_{k'} \end{pmatrix}.$$

The self-duality of this code implies that for any j $(1 \le j \le k_2)$

$$0 \equiv ((\boldsymbol{a}_{i_1}^{(0)}\ \boldsymbol{a}_{i_1}) + \cdots + (\boldsymbol{a}_{i_m}^{(0)}\ \boldsymbol{a}_{i_m})) \cdot (\boldsymbol{e}_j\ \boldsymbol{a}_{k_1+j}) \equiv (\boldsymbol{a}_{i_1}^{(0)} + \cdots + \boldsymbol{a}_{i_m}^{(0)}) \cdot \boldsymbol{e}_j + 1$$

using the equation $0 \equiv (\boldsymbol{e}_j^{(0)}\ \boldsymbol{a}_{k_1+j}) \cdot (\boldsymbol{e}_j^{(0)}\ \boldsymbol{a}_{k_1+j}) = 1 + \boldsymbol{a}_{k_1+j} \cdot \boldsymbol{a}_{k_1+j}$. As the index j ranges from 1 to k_2, we have

$$\boldsymbol{a}_{i_1}^{(0)} + \cdots + \boldsymbol{a}_{i_m}^{(0)} \equiv \mathbf{1}_{k_2},$$

and

$$(\boldsymbol{a}_{i_1}^{(0)}\ \boldsymbol{a}_{i_1}) + \cdots + (\boldsymbol{a}_{i_m}^{(0)}\ \boldsymbol{a}_{i_s}) \equiv \mathbf{1}_{k_2+(k_3+k')}.$$

Again from the self-duality of $\mathcal{C}_0$, we have for any j $(1 \le j \le k_1)$

$$\begin{aligned} 0 &\equiv ((\boldsymbol{e}_{i_1}\ \boldsymbol{a}_{i_1}^{(0)}\ \boldsymbol{a}_{i_1}) + \cdots + (\boldsymbol{e}_{i_m}\ \boldsymbol{a}_{i_m}^{(0)}\ \boldsymbol{a}_{i_m})) \cdot (\boldsymbol{e}_j\ \boldsymbol{a}_j^{(0)}\ \boldsymbol{a}_j) \\ &\equiv (\boldsymbol{e}_{i_1} + \cdots + \boldsymbol{e}_{i_m}) \cdot \boldsymbol{e}_j + 1. \end{aligned}$$

For the second congruence, we used $0 \equiv (\boldsymbol{e}_j\ \boldsymbol{a}_j^{(0)}\ \boldsymbol{a}_j) \cdot (\boldsymbol{e}_j\ \boldsymbol{a}_j^{(0)}\ \boldsymbol{a}_j) = 1 + (\boldsymbol{a}_j^{(0)}\ \boldsymbol{a}_j) \cdot (\boldsymbol{a}_j^{(0)}\ \boldsymbol{a}_j)$. As the index j ranges from 1 to k_1, we have

$$\boldsymbol{e}_{i_1} + \cdots + \boldsymbol{e}_{i_m} \equiv \mathbf{1}_{k_1}.$$

Therefore we see that $\{i_1, ..., i_m\} = \{1, ..., k_1\}$ and there is one possible linearly dependent equation in $\boldsymbol{M}$ only when

$$(\boldsymbol{e}_1\ \boldsymbol{a}_1^{(0)}\ \boldsymbol{a}_1) + \cdots + (\boldsymbol{e}_{k_1}\ \boldsymbol{a}_{k_1}^{(0)}\ \boldsymbol{a}_{k_1}) = \mathbf{1}_n,$$

which is equivalent to $\mathbf{1}_n \in T_1^{(0)}$. So the rank of $\boldsymbol{M}$ is $\frac{1}{2}k'(k'-1) + k_1 - \epsilon$ with

$$\epsilon = \begin{cases} 1 & \text{if } \mathbf{1} \in T_1^{(0)} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

In the case when $\epsilon = 1$, the doubly even quaternary code T_1 contains $\mathbf{1} + 2\boldsymbol{x}$ and

$$0 \equiv (\mathbf{1} + 2\boldsymbol{x}) \cdot (\mathbf{1} + 2\boldsymbol{x}) = n + 4(\mathbf{1} \cdot \boldsymbol{x} + \boldsymbol{x} \cdot \boldsymbol{x}) = n + 8\boldsymbol{x} \cdot \boldsymbol{x} \equiv n \pmod 8.$$

We need to check whether the simultaneous congruent equations represented in the matrix have a solution or not. We do this by computing the sum of

constants in the right-hand side corresponding to the rows in $\boldsymbol{M}$ which contain, for $i = 1, ..., k_1$, either $\boldsymbol{a}_i + \mathbf{1}$ or $\boldsymbol{a}_i$:

$$\begin{aligned}
&\sum_{i=1}^{k_1} g_i + \sum_{i=1}^{k_1-1} \sum_{j=i+1}^{k_1} f_{ij} \\
&\equiv \frac{1}{4}\left(\sum_{i=1}^{k_1} (\boldsymbol{e}_i\ \boldsymbol{a}_i^{(0)}\ \boldsymbol{a}_i) \cdot (\boldsymbol{e}_i\ \boldsymbol{a}_i^{(0)}\ \boldsymbol{a}_i) + 2 \sum_{1 \le i < j \le k_1} \boldsymbol{a}_i^{(0)} \cdot \boldsymbol{a}_j^{(0)} + 2 \sum_{1 \le i < j \le k_1} \boldsymbol{a}_i \cdot \boldsymbol{a}_j\right) \\
&\equiv \frac{1}{4} \sum_{i=1}^{k_1} (\boldsymbol{e}_i\ \boldsymbol{a}_i^{(0)}\ \boldsymbol{a}_i) \cdot \sum_{i=1}^{k_1} (\boldsymbol{e}_i\ \boldsymbol{a}_i^{(0)}\ \boldsymbol{a}_i) \equiv \frac{1}{4} \mathbf{1}_n \cdot \mathbf{1}_n \equiv \frac{1}{4} n \quad \equiv\ 0 \pmod 2.
\end{aligned}$$

We now have the following proposition.

Proposition 2.1. *Let $\mathcal{C}_0$ be a doubly even self-dual binary code with the following representation*

$$\mathcal{C}_0 = \begin{bmatrix} T_1^{(0)} \\ T_2^{(0)} \end{bmatrix} = \begin{bmatrix} I_{k_1} & A_{11}^{(0)} & A_{12}^{(0)} & A_{13}^{(0)} & A_{14}^{(0)} \\ 0 & I_{k_2} & A_{22}^{(0)} & A_{23}^{(0)} & A_{24}^{(0)} \end{bmatrix}.$$

If $\lambda(k_1, k_2, k_3)$ denotes the number of T_1-doubly even self-dual quaternary codes with representation

$$\mathcal{C}_1 = \begin{bmatrix} T_1 \\ T_2 \\ 2T_3 \end{bmatrix} = \begin{bmatrix} I_{k_1} & A_{11}^{(0)} & A_{12}^{(0)} + 2A_{12}^{(1)} & A_{13}^{(0)} + 2A_{13}^{(1)} & A_{14}^{(0)} + 2A_{14}^{(1)} \\ 0 & I_{k_2} & A_{22}^{(0)} & A_{23}^{(0)} + 2A_{23}^{(1)} & A_{24}^{(0)} + 2A_{24}^{(1)} \\ 0 & 0 & 2I_{k_3} & 2A_{33}^{(0)} & 2A_{34}^{(0)} \end{bmatrix} \tag{3}$$

then

$$\lambda(k_1, k_2, k_3) = \begin{cases} 0 & \text{if} \quad \mathbf{1} \in T_1^{(0)} \text{ and } 8 \not| n \\ 2^{\frac{1}{2}k'(k'+1) - k_1 + k_1 k_3 + \epsilon} & \text{otherwise} \end{cases},$$

where ϵ is defined as in (2).

Proof. If $\mathbf{1} \in T_1^{(0)}$ and $8 \not| n$ then there are no solutions to the matrix congruent equation. Otherwise, we are free to choose binary values for the entries in the solution column $(\boldsymbol{x}_i^t)$. If w denotes the size of matrix $\boldsymbol{M}$ minus the rank of $\boldsymbol{M}$, then $w = (k'^2 + k_1 k_3) - (\frac{1}{2}k'(k'-1) + k_1 - \epsilon) = \frac{1}{2}k'(k'+1) - k_1 + k_1 k_3 + \epsilon$. □

Once a T_1-doubly even self-dual quaternary code described in (3) is given, we can construct all the self-dual codes $\mathcal{C} = [2^i T_i]_{i=1,...,4}$ over $\mathbf{Z}_{2^4}$, where $T_1 = \mathbf{T}_1^{(1)} + 2^2(0\,0\,0\ A_{13}^{(2)} A_{14}^{(2)}) + 2^3(0\,0\,0\,0\ A_{14}^{(3)})$ $T_i = T_i^{(1)} + 2^{4-i}(0\,0\,0\,0\ A_{i4}^{(4-i)})$ for $i = 2$ and 3, $T_4 = (0\,0\,0\ I_{k_1} A_{44})$, the A_*'s are binary matrices and the $T_*^{(1)}$'s represent T_*'s in (3). From the self-duality of $\mathcal{C}$, we have $A_{44}^t = A_{14}^{(0)-1} A_{13}^{(0)}$, $A_{34}^{(1)t} = A_{14}^{(0)-1}(2^{-1}\mathbf{T}_1^{(1)} T_3^{(1)t})$ and $A_{24}^{(2)t} = A_{14}^{(0)-1}(2^{-2}\mathbf{T}_1^{(1)} T_2^{(1)t} + A_{13}^{(2)} A_{23}^{(0)t} + A_{14}^{(2)} A_{24}^{(0)t})$.

In the third equation $A_{13}^{(2)}$ and $A_{14}^{(2)}$ appear but these have not yet been given. We will consider the condition for them and for $A_{14}^{(3)}$ such as

$$\frac{1}{2^2}\mathbf{T}_1^{(1)}\mathbf{T}_1^{(1)t} + \widetilde{A'_{13}A_{13}^{(2)t}} + \widetilde{A'_{14}A_{14}^{(2)t}} + 2\widetilde{A_{14}^{(0)}A_{14}^{(3)t}} \equiv 0 \mod 2^2,$$

where $A'_{13} = A_{13}^{(0)} + 2A_{13}^{(1)}$ and $A'_{14} = A_{14}^{(0)} + 2A_{14}^{(1)}$.

Now we put $(d_{ij}) = \frac{1}{2^2}\mathbf{T}_1^{(1)}\mathbf{T}_1^{(1)t} + \widetilde{A_{13}^{(0)}A_{13}^{(2)t}}$ for any $k_1 \times k_2$ matrix $A_{13}^{(2)}$ and $(x_{ij}) = A_{14}^{(0)}A_{14}^{(2)t}$. Then $A_{14}^{(2)t}$ is uniquely determined by $A_{14}^{(0)-1}(x_{ij})$ such that $x_{ji} \equiv d_{ij} - x_{ij} \mod 2$ for $i < j$ and $x_{ii} \equiv \frac{1}{2}d_{ii} \pmod 2$. For $i > j$, the number of free binary values for x_{ij} is $\frac{1}{2}k_1(k_1-1)$.

Next we put $(f_{ij}) \equiv \frac{1}{2}\left(\frac{1}{2^2}\mathbf{T}_1^{(1)}\mathbf{T}_1^{(1)t} + \widetilde{A'_{13}A_{13}^{(2)t}} + \widetilde{A'_{14}A_{14}^{(2)t}}\right) \mod 2$ and $(y_{ij}) = A_{14}^{(0)}A_{14}^{(3)t}$. Then $A_{14}^{(3)t}$ is uniquely determined by $A_{14}^{(0)-1}(y_{ij})$ where $y_{ji} = f_{ij} - y_{ij}$ for $i < j$ and $\frac{1}{2}k_1(k_1+1)$ is the number of freely chosen binary values for y_{ij} $(i \geq j)$.

Therefore we have the following proposition.

Proposition 2.2. *For a fixed T_1-doubly even self-dual quaternary code described in (3), the number of induced codes over $\mathbf{Z}_{2^4}$ is*

$$2^{k_1(k_1+k_2)}.$$

3 Mass Formula for Self-dual Code over $\mathbf{Z_{16}}$

Now we establish the mass formula.

Theorem 3.1. *Let $N_{16}(n)$ denote the number of distinct self-dual codes over $\mathbf{Z}_{16}$ of length n.*

1. *If n is odd, then*

$$N_{16}(n) = \sum_{k'=0}^{\frac{n-1}{2}} 2^{\frac{1}{2}k'(k'+1)} \prod_{i=0}^{k'-1} \frac{2^{n-2i-2} + (-1)^{\delta}2^{\frac{n-1}{2}-i-1} - 1}{2^{i+1}-1}\rho(k'),$$

where

$$\delta = \begin{cases} 0 & n \equiv 1,\ 7 \\ 1 & n \equiv 3,\ 5 \end{cases} \pmod 8.$$

2. *If $n \equiv 2,\ 6 \pmod 8$, then*

$$N_{16}(n) = \sum_{k'=0}^{\frac{n}{2}-1} 2^{\frac{1}{2}k'(k'+1)} \prod_{i=0}^{k'-1} \frac{2^{n-2i-2} - 1}{2^{i+1}-1}\rho(k').$$

3. If $n \equiv 4 \pmod 8$, *then*

$$N_{16}(n) = \sum_{k'=0}^{\frac{n}{2}-2} 2^{\frac{1}{2}k'(k'+1)} \prod_{i=0}^{k'-1} \frac{2^{n-2i-2} - 2^{\frac{n}{2}-i-1} - 2}{2^{i+1}-1} \rho(k').$$

4. If $n \equiv 0 \pmod 8$, *then*

$$N_{16}(n) = \sum_{k'=0}^{\frac{n}{2}-1} 2^{\frac{1}{2}k'(k'+1)} \prod_{i=0}^{k'-1} \frac{2^{n-2i-2} + 2^{\frac{n}{2}-i-1} - 2}{2^{i+1}-1} \rho(k') + \sum_{k'=0}^{\frac{n}{2}} 2^{\frac{1}{2}k'(k'-1)} \prod_{i=0}^{k'-2} \frac{2^{n-2i-2} + 2^{\frac{n}{2}-i-1} - 2}{2^{i+1}-1} \rho'(k').$$

Here

$$\rho(k') = \sum_{k_1=0}^{k'} 2^{(n-k'-1)k_1} \rho(k', k_1), \text{ and } \rho'(k') = \sum_{k_1=0}^{k'} 2^{(n-k'-1)k_1} \frac{2^{k_1}-1}{2^{k'}-1} \rho(k', k_1),$$

$$\text{with } \rho(k', k_1) = \prod_{i=0}^{k_1-1} \frac{2^{k'-i}-1}{2^{k_1-i}-1}.$$

In the formulas above, we take the value of a product to be 1 whenever the starting index is greater than the limit.

Proof. First note that the length n is a multiple of 4 whenever a doubly even binary code containing $\mathbf{1}_n$ exists.

The number $\sigma_1(n,k)$ of distinct doubly even self-orthogonal binary codes of length n and dimension k containing $\mathbf{1}_n$ as well as the number $\sigma_2(n,k)$ of those which do not contain $\mathbf{1}_n$ are explicitly given in the proof of the Theorem 4.1 in [6]. We note that $\sigma_2(n, \frac{n}{2}) = 0$ for any even n and $\sigma_2(n, \frac{n}{2}-1) = 0$ for $n \equiv 4 \pmod 8$. For each binary code $\mathcal{C}_0$ of size n and the dimension k', the number of partitions $[T_i]$ is $\rho(k', k_1)$ if $\mathbf{1} \notin T_1$ and $\rho(k'-1, k_1-1)$ otherwise.

Over each binary code with a given partition, there exist $\lambda(k_1,k_2,k_3)$ T_1-doubly even representations for quaternary code $\mathcal{C}_1$. And from each such code, we can construct $2^{k_1k'}$ distinct self-dual codes (Proposition 2.2). Thus if $8|n$, the number of distinct self-dual codes of type (k_1,k_2,k_3) as described in the left part of the scheme diagram in Section 2 is

$$\left(\sigma_2(n,k')\rho(k',k_1) + 2\sigma_1(n,k')\rho(k'-1,k_1-1)\right) \lambda_{\epsilon=0}(k_1,k_2,k_3) \times 2^{k_1k'}.$$

Otherwise, the number of such codes is $\sigma_2(n,k')\rho(k',k_1)\lambda_{\epsilon=0}(k_1,k_2,k_3) \times 2^{k_1k'}$. This is because $\sigma_1(n,k')\lambda_{\epsilon=0}(k_1,k_2,k_3) = 0$ if $8 \not| n$, and $\lambda_{\epsilon=1}(k_1,k_2,k_3) = 2\lambda_{\epsilon=0}(k_1,k_2,k_3)$ if it is not 0. Except for the case of $\lambda = 0$, $\lambda_{\epsilon=0}(k_1,k_2,k_3) = 2^{\frac{1}{2}k'(k'+1)-k_1+k_1k_3}$ from Proposition 2.1. Then the calculation $-k_1 + k_1k_3 + k_1k' = (k_3 + k' - l)k_1 = (n-k'-1)k_1$ gives us the mass formulas. □

4 Example for $n = 8$, $k_1 = 4$, and $k_2 = k_3 = 0$

Let $\mathcal{C}_0 = \left[I_4 \quad A_{14}^{(0)} \right]$ with $A_{14}^{(0)} = \begin{pmatrix} 0\,1\,1\,1 \\ 1\,0\,1\,1 \\ 1\,1\,0\,1 \\ 1\,1\,1\,0 \end{pmatrix}$. Then $\mathcal{C}_0$ is a doubly even self-dual binary code with $d(\mathcal{C}_0) = 4$ and $\mathcal{C}_0 \ni \mathbf{1}_8$.

We then have $\mathcal{C}_1 = \left[I_4 \quad A_{14}^{(0)} + 2A_{14}^{(1)} \right]$ with $A_{14}^{(1)}$ equal to

$$\begin{pmatrix} 1 & \alpha_1+\alpha_3+\alpha_4+1 & \alpha_1+\alpha_3+\alpha_4+\alpha_6+\alpha_7 & \alpha_1 \\ \alpha_2+\alpha_3+\alpha_4+\alpha_5+\alpha_6 & 1 & \alpha_2+\alpha_3+\alpha_4+\alpha_6+\alpha_7+1 & \alpha_2 \\ \alpha_3+\alpha_5+\alpha_6+1 & \alpha_3 & 1 & \alpha_4 \\ \alpha_5 & \alpha_6 & \alpha_7 & 1 \end{pmatrix},$$

where α_* can be any binary value.

For instance, setting $\alpha_1 = \alpha_2 = \alpha_4 = \alpha_5 = \alpha_6 = \alpha_7 = 0$ and $\alpha_3 = 1$, we have $\mathcal{C}_1$ with $A_{14}^{(0)} + 2A_{14}^{(1)} = \begin{pmatrix} 2\,1\,3\,1 \\ 3\,2\,1\,1 \\ 1\,3\,2\,1 \\ 1\,1\,1\,2 \end{pmatrix}$, a doubly even self-dual quaternary code with Lee distance $d_L(\mathcal{C}_1) = 6$.

Computing $(d_{ij}) = \begin{pmatrix} 4\,3\,3\,2 \\ 3\,4\,3\,2 \\ 3\,3\,4\,2 \\ 2\,2\,2\,2 \end{pmatrix}$, we have

$$A_{14}^{(2)t} = A_{14}^{(0)-1} \begin{pmatrix} 0 & x_{12} & x_{13} & x_{14} \\ \overline{x_{12}} & 0 & x_{23} & x_{24} \\ \overline{x_{13}} & \overline{x_{23}} & 0 & x_{34} \\ -x_{14} & -x_{24} & -x_{34} & 1 \end{pmatrix}, \text{ where } \overline{x} = 1 - x.$$

From $(f_{ij}) \equiv \begin{pmatrix} 0\,0\,0\,1 \\ 0\,0\,0\,1 \\ 0\,0\,0\,1 \\ 1\,1\,1\,0 \end{pmatrix}$, we obtain the induced code $\mathcal{C}$ over $\mathbf{Z}_{16}$:

$$\mathcal{C} = \left[I_4 A_{14}^{(0)} + 2A_{14}^{(1)} + 2^2 A_{14}^{(2)} + 2^3 A_{14}^{(3)} \right]$$

with $A_{14}^{(3)t} = A_{14}^{(0)-1} \begin{pmatrix} y_{11} & y_{12} & y_{13} & y_{14} \\ y_{12} & y_{22} & y_{23} & y_{24} \\ y_{13} & y_{23} & y_{33} & y_{34} \\ \overline{y_{14}} & \overline{y_{24}} & \overline{y_{34}} & y_{44} \end{pmatrix}$.

References

1. Balmaceda, J., Betty, R., Nemenzo, F.: Mass formula for self-dual codes over $\mathbf{Z}_{p^2}$. Discrete Mathematics 308, 2984–3002 (2008)
2. Dougherty, S., Harada, M., Solé, P.: Self-dual codes over rings and the Chinese remainder theorem. Hokkaido Math. J. 28, 253–283 (1999)

3. Fields, J., Gaborit, P., Leon, J., Pless, V.: All self-dual $\mathbf{Z}_4$ codes of length 15 or less are known. IEEE Trans. Inform. Theory 44, 311–322 (1998)
4. Gaborit, P.: Mass formulas for self-dual codes over $\mathbf{Z}_4$ and $\mathbb{F}_q + u\mathbb{F}_q$ rings. IEEE Trans. Inform. Theory 42, 1222–1228 (1996)
5. Hammons, A., Kumar, P., Calderbank, A., Sloane, N., Solé, P.: The $\mathbf{Z}_4$ linearity of Kerdock, Preparata, Goethals and related codes. IEEE Trans. Inform. Theory 40, 301–319 (1994)
6. Nagata, K., Nemenzo, F., Wada, H.: The number of self-dual codes over $\mathbf{Z}_{p^3}$. Designs, Codes and Cryptography 50, 291–303 (2009)
7. Nagata, K., Nemenzo, F., Wada, H.: Constructive algorithm of self-dual error-correcting codes. In: Proc. of 11th International Workshop on Algebraic and Combinatorial Coding Theory, pp. 215–220 (2008) ISSN 1313-423X

A Non-abelian Group Based on Block Upper Triangular Matrices with Cryptographic Applications

Rafael Álvarez, Leandro Tortosa, José Vicent, and Antonio Zamora

Departamento de Ciencia de la Computación e Inteligencia Artificial, Universidad de Alicante, Campus de San Vicente, Ap. Correos 99, E–03080, Alicante, Spain
ralvarez@dccia.ua.es, tortosa@dccia.ua.es, jvicent@dccia.ua.es, zamora@dccia.ua.es

Abstract. The aim of this article is twofold. In the first place, we describe a special non-abelian group of block upper triangular matrices, and verify that choosing properly certain parameters, the order of the subgroup generated by one of these matrices can be as large as needed. Secondly, we propose and implement a new key exchange scheme based on these primitives. The security of the proposed system is based on discrete logarithm problem although a non-abelian group of matrices is used. The primary advantadge of this scheme is that no prime numbers are used and the efficiency is guaranteed by the use of a quick exponentiation algorithm for this group of matrices.

Keywords: Polynomial matrices, Block matrices, Quick exponentiation, Cryptography, Public Key.

1 Introduction

A lot of popular public-key encryption systems are based on number theory problems such as factoring integers or finding discrete logarithms. The underlying algebraic structures are, very often, abelian groups, as we can see in [3,4,16]; this is especially true in the case of the Diffie-Hellman method (DH), that was the first practical public key technique and introduced in 1976 (see [8]). In Crypto 2000 [12], Ko et al. proposed a new public cryptosystem based on Braid groups, which are non abelian groups, and it was the first practical public key cryptosystem based on non abelian groups.

The Discrete Logarithm Problem (DLP, see [7, 15, 20]) is, together with the Integer Factoring Problem (IFP, see [8]), one of the main problems upon which public-key cryptosystems are built. Thus, efficiently computable groups where the DLP is hard to break are very important in cryptography.

The purpose of this proposal is the analysis and implementation of a key exchange scheme based on a special non-abelian group of block upper triangular matrices. The main idea of this paper is to study the cryptographic behavior

M. Bras-Amorós and T. Høholdt (Eds.): AAECC 2009, LNCS 5527, pp. 117–126, 2009.

of products of the type $M_1^v M_2^w$, with v, w integers and M_1, M_2 elements of the group of matrices previously mentioned.

The paper is organized as follows: in Section 2 we present some properties concerning block upper triangular matrices defined over $\mathbb{Z}_p$. We will show that they can generate sets of large and known order if parameters are chosen adequately; due to this we study the order of the subgroup generated by a special matrix M; in Section 3 we describe the design of a key exchange scheme based on the non-abelian group of triangular matrices, for this, we use a especial quick exponentiation method; finally, some conclusions are given.

2 Preliminaries

Given p a prime number and $r, s \in \mathbb{N}$,we denote by $\mathrm{Mat}_{r\times s}(\mathbb{Z}_p)$ the matrices of size $r \times s$ with elements in $\mathbb{Z}_p$, and by $\mathrm{GL}_r(\mathbb{Z}_p)$ and $\mathrm{GL}_s(\mathbb{Z}_p)$, the invertible matrices of size $r \times r$ and $s \times s$ respectively.

We define

$$\Theta = \left\{ \begin{bmatrix} A & X \\ O & B \end{bmatrix}, A \in \mathrm{GL}_r(\mathbb{Z}_p), B \in \mathrm{GL}_s(\mathbb{Z}_p), X \in \mathrm{Mat}_{r\times s}(\mathbb{Z}_p) \right\}$$

Theorem 1. *The set Θ has a structure of non-abelian group for the product of matrices.*

The theorem that follows gives us a method for calculating the powers of this kind of matrices (for details see [6]).

Theorem 2. *Let $M = \begin{bmatrix} A & X \\ O & B \end{bmatrix}$ be an element of the set Θ, we consider the subgroup generated by the different powers of M.*

Taking h as a non negative integer then

$$M^h = \begin{bmatrix} A^h & X^{(h)} \\ O & B^h \end{bmatrix}, \tag{1}$$

where

$$X^{(h)} = \begin{cases} \boldsymbol{0} & if \quad h = 0 \\ \sum_{i=1}^{h} A^{h-i} X B^{i-1} & if \quad h \geq 1 \end{cases} \tag{2}$$

Also, if $0 \leq t \leq h$ then

$$X^{(h)} = A^t X^{(h-t)} + X^{(t)} B^{h-t} \tag{3}$$

$$X^{(h)} = A^{(h-t)} X^{(h)} + X^{(h-t)} B^t . \tag{4}$$

As a consequence, in the case $t = 1$ we have

$$X^{(h)} = A X^{(h-1)} + X B^{h-1} \text{ or } X^{(h)} = A^{h-1} X + X^{(h-1)} B$$

And, taking a, b non negative integers such as $a + b \geq 0$, we have

$$X^{(a+b)} = A^a X^{(b)} + X^{(a)} B^b \tag{5}$$

It is very important to achieve a high order of the group. With this aim, the following construction (see [11, 13]) guarantees a certain order.

Let $f(x) = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1} + x^n$ a monic polynomial in $\mathbb{Z}_p[x]$, whose companion $n \times n$ matrix is

$$\overline{A} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \\ -a_0 & -a_1 & -a_2 & \dots & -a_{n-2} & -a_{n-1} \end{bmatrix}.$$

If f is an irreducible polynomial in $\mathbb{Z}_p[x]$, then the order of the matrix $\overline{A}$ is equal to the order of any root of f in $\mathbb{F}_{p^n}$ and the order of $\overline{A}$ divides $p^n - 1$ (see [14]). Moreover, assuming that f is a primitive polynomial in $\mathbb{Z}_p[x]$, the order of $\overline{A}$ is exactly $p^n - 1$.

Odoni, Varadharajan and Sanders [19] propose an extended scheme based on the construction of block matrices like this

$$\overline{A} = \begin{bmatrix} \overline{A_1} & 0 & \dots & 0 \\ 0 & \overline{A_2} & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & \overline{A_k} \end{bmatrix},$$

where $\overline{A_i}$ is the companion matrix of f_i, and f_i, for $i = 1, 2, \ldots, k$, are different primitive polynomials in $\mathbb{Z}_p[x]$ of degree n_i, for $i = 1, 2, \ldots, k$, respectively. The order of $\overline{A_i}$ is $p^{n_i} - 1$, for $i = 1, 2, \ldots, k$. Therefore, the order of $\overline{A}$ is $lcm(p^{n_1} - 1, p^{n_2} - 1, \ldots, p^{n_k} - 1)$.

With the aim of using this type of matrix in a public key cryptosystem, Odoni, Varadharajan and Sanders conjugate this matrix $\overline{A}$ with an invertible matrix P of size $n \times n$, with $n = n_1 + n_2 + \ldots + n_k$, obtaining a new matrix $A = P\overline{A}P^{-1}$ that has the same order as $\overline{A}$. If we construct the blocks A and B of $M \in \Theta$ using primitive polynomials, we can guarantee a very high order.

Let $f(x) = a_0 + a_1 x + \cdots + a_{r-1} x^{r-1} + x^r$, $g(x) = b_0 + b_1 x + \cdots + b_{s-1} x^{s-1} + x^s$ be two primitive polynomials in $\mathbb{Z}_p[x]$ and $\overline{A}$, $\overline{B}$ the corresponding companion matrices. Let P and Q be two invertible matrices, such that $A = P\overline{A}P^{-1}$ and $B = Q\overline{B}Q^{-1}$. With this construction, the order of M is $lcm(p^r - 1, p^s - 1)$.

Using cyclotomic fields theory we know that the polynomial $x^n - 1$ is divided by $x^d - 1$ if $d|n$; therefore, if we choose r and s such that they are relatively prime, the number of common divisors is diminished and the $lcm(p^r - 1, p^s - 1)$ will be maximum.

Table 1 shows a comparison of the variation of the order of M depending on the value of p, r and s, where r and s are the sizes of blocks A and B respectively.

Table 1. Order of M depending of r, s and p

r	s	p	Order of M
2	3	2^{200}	2^{800}
3	4	2^{200}	2^{1200}
3	5	2^{200}	2^{1400}
4	5	2^{200}	2^{1600}
5	6	2^{200}	2^{2000}
31	32	29	2^{302}
47	48	29	2^{457}
60	61	29	2^{584}
130	131	29	2^{1264}
216	217	29	2^{2099}

Given the excellent cryptographic properties of block upper triangular matrices, we will see an aplication of this set to public key cryptography.

3 Cryptographic Application: A Key Exchange Scheme

The DLP ([17]) for a set G is finding, given $\alpha, \beta \in G$, a nonnegative integer x (if it exists) such that $\beta = \alpha^x$. The smallest such integer x is called the discrete logarithm of β to the base α, and is written $x = \log_\alpha \beta$. Clearly, the discrete logarithm problem for a general group G is exactly the problem of inverting the exponentiation function (see [18] for more information).

This is especially true for the Diffie-Hellman method ([8]), that was the first practical public key technique to be published. The security of this method for key exchanges, is based on the discrete logarithm problem, and it uses a prime number p and a primitive element $r \in \mathbb{Z}_p$.

Privacy or security of messages is not the only problem area in cryptology. It is also important that user identity can be authenticated. Digital signature is a property of asymmetric cryptography, that allows authentication. It consists of two processes: signing a message and verifying a message signature; and it must depend on the message to be signed.

The method presented in this section uses a non-abelian group based on the powers of a block upper triangular matrix, which is a very flexible technique.

Let $M_1 = \begin{bmatrix} A_1 & X_1 \\ \mathbf{0} & B_1 \end{bmatrix}$ and $M_2 = \begin{bmatrix} A_2 & X_2 \\ \mathbf{0} & B_2 \end{bmatrix}$ be two elements of the set Θ with orders m_1 and m_2 respectively.

We define the following notation for a pair of numbers $x, y \in \mathbb{N}$:

$$A_{xy} = A_1^x A_2^y,$$
$$B_{xy} = B_1^x B_2^y$$

and

$$C_{xy} = A_1^x X_2^{(y)} + X_1^{(x)} B_2^y.$$

If two users U and V wish to exchange a key, they may execute the following steps:

1. U and V agree on $p \in \mathbb{N}$ and $M_1, M_2 \in \Theta$, with m_1, m_2 the orders of M_1 and M_2, respectively.
2. U generates two random private keys $r, s \in \mathbb{N}$ such that

$$1 \leq r \leq m_1 - 1,\ 1 \leq s \leq m_2 - 1,$$

and computes A_{rs}, B_{rs}, C_{rs} constructing

$$C = \begin{bmatrix} A_{rs} & C_{rs} \\ \mathbf{0} & B_{rs} \end{bmatrix}.$$

3. U sends C to V.
4. V generates two random private keys $v, w \in \mathbb{N}$ such that

$$1 \leq v \leq m_1 - 1,\ 1 \leq w \leq m_2 - 1,$$

and computes A_{vw}, B_{vw}, C_{vw} constructing

$$D = \begin{bmatrix} A_{vw} & C_{vw} \\ \mathbf{0} & B_{vw} \end{bmatrix}.$$

5. V sends D to U.
6. The public keys of U and V are respectively the matrices C and D.
7. U computes

$$K_u = A_1^r A_{vw} X_2^{(s)} + A_1^r C_{vw} B_2^s + X_1^{(r)} B_{vw} B_2^s.$$

8. V computes

$$K_v = A_1^v A_{rs} X_2^{(w)} + A_1^v C_{rs} B_2^w + X_1^{(v)} B_{rs} B_2^w.$$

The following theorem shows that $K_u = K_v$.

Theorem 3. *If* $K_u = A_1^r A_{vw} X_2^{(s)} + A_1^r C_{vw} B_2^s + X_1^{(r)} B_{vw} B_2^s$ *and* $K_v = A_1^v A_{rs} X_2^{(w)} + A_1^v C_{rs} B_2^w + X_1^{(v)} B_{rs} B_2^w$, *then* $K_u = K_v$.

Proof 1. *We have*

$$C = \begin{bmatrix} A_{rs} & C_{rs} \\ \boldsymbol{0} & B_{rs} \end{bmatrix} = M_1^r M_2^s,\ D = \begin{bmatrix} A_{vw} & C_{vw} \\ \boldsymbol{0} & B_{vw} \end{bmatrix} = M_1^v M_2^w,$$

$$M_1^r = \begin{bmatrix} A_1^r & X_1^{(r)} \\ \boldsymbol{0} & B_1^r \end{bmatrix},\ M_1^v = \begin{bmatrix} A_1^v & X_1^{(v)} \\ \boldsymbol{0} & B_1^v \end{bmatrix},$$

$$M_2^s = \begin{bmatrix} A_2^s & X_2^{(s)} \\ \boldsymbol{0} & B_2^s \end{bmatrix} \text{ and } M_2^w = \begin{bmatrix} A_2^w & X_2^{(w)} \\ \boldsymbol{0} & B_2^w \end{bmatrix}.$$

Let

$$M_u = M_1^r D M_2^s = \begin{bmatrix} A_u & K_u \\ \boldsymbol{0} & B_u \end{bmatrix}$$

and

$$M_v = M_1^v C M_2^w = \begin{bmatrix} A_v & K_v \\ \boldsymbol{0} & B_v \end{bmatrix}.$$

Then

$$M_u = M_1^r D M_2^s = M_1^r M_1^v M_2^w M_2^s = M_1^v M_1^r M_2^s M_2^w = M_1^v C M_2^w = M_v$$

and, consequently, $K_u = K_v$.

As we have demonstrated in this theorem, now both U and V share a common and secret key,

$$K_u = K_v = P.$$

The private keys are r, s and v, w, respectively. These keys do not have to be prime numbers (we avoid primality tests).

In our key exchange scheme based on block upper triangular matrices, the usage of big powers of matrices is required (see [1]); so the implementation of an efficient and trustworthy quick exponentiation algorithm (see [2, 9, 10, 21]) becomes necessary for the accomplishment of this task.

Given $n \in \mathbb{N}$, then a ordered set of indices exist

$$I = \{i_1,\ i_2,\ i_3,\ i_4,\ \ldots, i_q\}$$

so that $n = 2^{i_1} + 2^{i_2} + 2^{i_3} + \ldots + 2^{i_q}$.

In order to compute the powers of A and B,

$$A^n = A^{2^{i_1}+2^{i_2}+\ldots+2^{i_q}} = A^{2^{i_1}} A^{2^{i_2}} \ldots A^{2^{i_q}}$$

$$B^n = B^{2^{i_1}+2^{i_2}+\ldots+2^{i_q}} = B^{2^{i_1}} B^{2^{i_2}} \ldots B^{2^{i_q}},$$

we use

$A^{2^0} = A = A_0$
$A^{2^1} = AA = A_1$
$A^{2^2} = A^2 A^2 = A_2$
. . . . . .
$A^{2^e} = A^{2^{e-1}} A^{2^{e-1}} = A_e$

That is to say, computing big powers of the blocks A, B of matrices M, is reduced to multiplying matrices quickly. So, for block X we have the following theorem.

Theorem 4. *Given an integer number* n*, whose binary decomposition is*

$$n = \sum_{j=1}^{q} 2^{i_j},$$

and a set of indices $I = \{i_1,\ i_2,\ i_3,\ i_4,\ \ldots, i_q\}$, *we have:*

$$X^{(n)} = \sum_{k=1}^{q} A^{n_1^{(k)}} X^{(n_2^{(k)})} B^{n_3^{(k)}} \tag{6}$$

Where

$$n_1^{(k)} = \sum_{j=1}^{q-k} 2^{i_j},\ \ for\ k = 1, 2, 3, \ldots q-1,\ \ and\ n_1^{(q)} = 0;$$

$$n_2^{(k)} = 2^{i_{q-k+1}}\ for\ k = 1, 2, 3, \ldots q\ ;$$

$$n_3^{(k)} = \sum_{j=q-k+2}^{q} 2^{i_j}\ \ for\ k = 2, 3, \ldots q,\ \ and\ n_3^{(1)} = 0.$$

In Alvarez et al. (see [2]) we prove this theorem and we show some examples of its use.

4 Security Analysis

Brute force attacks are infeasible if a sufficiently big order for M_1 and M_2 is chosen as, for example, 1024 bits. In the same way, we use big values for the private keys r, s, v, w (about 1024 bits) and due to this, we can avoid Meet-in-the-Middle-Attacks.

A widely used algorithm for the cryptanalysis of public key schemes based on matrix powers is due to Menezes and Wu (see [18]). It, basically, establishes the possibility of reducing the full discrete logarithm problem to a series of smaller discrete logarithms over finite fields. This algorithm is not viable for the presented scheme since no matrix powers are published.

Another technique for effectively cryptanalyzing some schemes based on block upper triangular matrices, has been developed by Climent, Gorla and Rosenthal [5] and is based on the Cayley-Hamilton theorem.

Theorem 5. *(Cayley-Hamilton Theorem). Let a matrix* $M \in GL_n(\mathbb{Z}_p)$ *and its characteristic equation*

$$q_M(\lambda) = \det(\lambda I_n - M) = a_0 + a_1\lambda + a_2\lambda^2 + \ldots + a_{n-1}\lambda^{n-1} + \lambda^n.$$

Then

$$q_M(M) = a_0 + a_1 M + a_2 M^2 + \ldots + a_{n-1}M^{n-1} + M^n = \boldsymbol{0}_n,$$

where I_n is the identity matrix of size n and $\mathbf{0}_n$ the null matrix of the same size.

This attack is not viable either, since two different matrices with different characteristic equations are employed. Let us analyze the inefficiency of this type of attack.

Consider

$$M_1 = \begin{bmatrix} A_1 & X_1 \\ \mathbf{0} & B_1 \end{bmatrix} \in \Theta \ , M_2 = \begin{bmatrix} A_2 & X_2 \\ \mathbf{0} & B_2 \end{bmatrix} \in \Theta,$$

be two matrices of sizes $n = r + s$, and assume that

$$\det(\lambda I - M_1) \neq 0 \text{ and } \det(\lambda I - M_2) \neq 0,$$

then

$$\begin{aligned} q_{M_1}(\lambda) &= \det\left(\begin{bmatrix} \lambda I - A_1 & -X_1 \\ \mathbf{0} & \lambda I - B_1 \end{bmatrix}\right) \\ &= \det(\lambda I - A_1) \cdot det(\lambda I - B_1) \\ &= q_{A_1}(\lambda) \cdot q_{B_1}(\lambda) \\ &= a_0 + a_1\lambda + a_2\lambda^2 + \ldots + a_{n-1}\lambda^{n-1} + \lambda^n, \end{aligned}$$

$$\begin{aligned} q_{M_2}(\lambda) &= \det\left(\begin{bmatrix} \lambda I - A_2 & -X_2 \\ \mathbf{0} & \lambda I - B_2 \end{bmatrix}\right) \\ &= \det(\lambda I - A_2) \cdot det(\lambda I - B_2) \\ &= q_{A_2}(\lambda) \cdot q_{B_2}(\lambda) \\ &= b_0 + b_1\lambda + b_2\lambda^2 + \ldots + b_{n-1}\lambda^{n-1} + \lambda^n \end{aligned}$$

with $q_{M_1}(\lambda) \neq q_{M_2}(\lambda)$.

The Cayley-Hamilton theorem guarantees that $q_{M_1}(M_1) = q_{M_2}(M_2) = 0$, so

$$a_0 I + a_1 M_1 + a_2 M_1^2 + \ldots + a_{n-1} M_1^{n-1} + M_1^n = \mathbf{0},$$

$$a_0 I + a_1 M_1 + a_2 M_1^2 + \ldots + a_{n-1} M_1^{n-1} = -M_1^n,$$

multiplying by M_1

$$a_0 M_1 + a_1 M_1^2 + a_2 M_1^3 + \ldots + a_{n-1} M_1^n = -M_1^{n+1}.$$

Replacing the value of M_1^n

$$\begin{aligned} a_0 M_1 + a_1 M_1^2 + a_2 M_1^3 + \ldots + a_{n-1}(-a_0 I - a_1 M_1 - a_2 M_1^2 - \ldots - a_{n-1} M_1^{n-1}) \\ = -M_1^{n+1}, \end{aligned}$$

and grouping terms we obtain

$$M_1^{n+1} = b_0 I + b_1 M_1 + b_2 M_1^2 + \ldots + b_{n-1} M_1^{n-1}.$$

with $b_0 = a_0 a_{n-1}$ and $b_i = a_{n-1} a_i - a_{i-1}$ for $i = 1, \ldots, n-1$

Following this process for a certain $p \geq n$, we have

$$M_1^p = c_0 I + c_1 M_1 + c_2 M_1^2 + \ldots + c_{n-1} M_1^{n-1}, \tag{7}$$

then

$$M_1^p = \begin{bmatrix} A_1^p & X_1^{(p)} \\ \mathbf{0} & B_1^p \end{bmatrix}$$
$$= c_0 \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & I \end{bmatrix} + c_1 \begin{bmatrix} A_1 & X_1 \\ \mathbf{0} & B_1 \end{bmatrix} + \ldots + c_{n-1} \begin{bmatrix} A_1^{n-1} & X_1^{(n-1)} \\ \mathbf{0} & B_1^{n-1} \end{bmatrix}.$$

Consequently,

$$\begin{aligned} A_1^p &= c_0 I + c_1 A_1 + c_2 A_1^2 + \ldots + c_{n-1} A_1^{n-1}, \\ X_1^{(p)} &= c_1 X_1 + c_2 X_1^{(2)} + \ldots + c_{n-1} X_1^{(n-1)}, \\ B_1^p &= c_0 I + c_1 B_1 + c_2 B_1^2 + \ldots + c_{n-1} B_1^{n-1}. \end{aligned}$$

If we proceed like in expression (7), we obtain

$$M_2^p = d_0 I + d_1 M_2 + d_2 M_2^2 + \ldots + d_{n-1} M_2^{n-1}.$$

In the scheme that we are analyzing, we know M_1 and M_2 so we can set up the following linear system

$$\begin{aligned} M_1^r &= e_1 M_1 + e_2 M_1^2 + \ldots + e_{n-1} M_1^{n-1}, \\ M_2^v &= f_1 M_2 + f_2 M_2^2 + \ldots + f_{n-1} M_2^{n-1}. \end{aligned}$$

Since r and s are private keys, coefficients $e_1, e_2, \ldots, e_{n-1}$ and $f_1, f_2, \ldots, f_{n-1}$, as well as the matrices M_1^r and M_2^v are unknown, rendering the system unsolvable. Therefore, the Climent, Gorla and Rosenthal technique (see [5]), based on the Cayley Hamilton theorem, cannot be used to obtain a system suitable for cryptanalysis. In order to cryptanalyze this scheme, we must apply square root algorithms to the calculation of discrete logarithms. We can choose the blocks A_1, A_2, B_1 and B_2 so that the inefficiency of this type of attack is guaranteed.

5 Conclusions

We propose a key exchange scheme, based on the behavior of matrix products of the type $M_1^v M_2^w$, where M_1, M_2 are elements of a non-abelian group of block upper triangular matrices with a big enough order, being v, w integers. One of the main advantages of this scheme is the absence of big prime numbers, avoiding the need for primality tests. Moreover, the proposed scheme is very efficient since it employs fast exponentiation algorithms for such type of matrices.

The presented algebraic structure can be applied to other cryptographic applications in addition to the proposal. It is based on the discrete logarithm problem for matrices and presents the advantage of reducing the required key length for a given level of security. Another advantage is the use of non-prime numbers, which prevents the primality tests that slow cryptographic protocols down.

References

1. Agnew, G.B., Mullin, R.C., Vanstone, S.A.: Fast Exponentiation in $GF(2^n)$. In: Günther, C.G. (ed.) EUROCRYPT 1988. LNCS, vol. 330, pp. 251–255. Springer, Heidelberg (1988)
2. Álvarez, R., Ferrández, F., Vicent, J.F., Zamora, A.: Applying quick exponentiation for block upper triangular matrices. Applied Mathematics and Computation 183, 729–737 (2006)
3. Anshel, I., Anshel, M., Goldfeld, D.: An algebraic method for public-key cryptography. Mathematical Research Letters 6, 287–291 (1999)
4. Blake, I., Seroussi, G., Smart, N.: Elliptic Curves in Cryptography. London Mathematical Society. Lecture Notes, vol. 265. Cambridge University Press, Cambridge (1999)
5. Climent, J.J., Gorla, E., Rosenthal, J.: Cryptanalysis of the CFVZ cryptosystem. Advances in Mathematics of Communications (AMC) 1, 1–11 (2006)
6. Climent, J.J.: Propiedades espectrales de matrices: el índice de matrices triangulares por bloques. La raíz Perron de matrices cocíclicas no negativas. Ph.D Thesis, Valencia. Sapin (1993)
7. Coppersmith, D., Odlyzko, A., Schroeppel, R.: Discrete logarithms in $GF(p)$. Algorithmica, 1–15 (1986)
8. Diffie, W., Hellman, M.: New directions In Cryptography. IEEE Trans. Information Theory 22, 644–654 (1976)
9. Gathen, J.: Efficient and Optimal Exponentiation in Finite Fields. Computational Complexity 1, MR 94a:68061, 360–394 (1991)
10. Gordon, D.M.: A Survey of Fast Exponentiation Methods. Journal of Algorithms 27, 129–146 (1998)
11. Hoffman, K., Kunze, R.: Linear Algebra. Prentice-Hall, New Jersey (1971)
12. Ko, K.H., Lee, S.J., Cheon, J.H., Han, J.W., Kang, J.: New Public Key Cryptosystem Unsing Braid Groups. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 166–183. Springer, Heidelberg (2000)
13. Koblitz, N.: A Course in Number Theory and Cryptography. Springer, Heidelberg (1987)
14. Lidl, R., Niederreiter, H.: Introduction to Finite Fields and Their Applications. Cambridge University Press, Cambridge (1994)
15. McCurley, K.: The discret logarithm problem. Cryptology and Computational Number Theory. In: Proceedings of Symposia in Applied Mathematics, vol. 42, pp. 49–74 (1990)
16. Menezes, A., Van Oorschot, P., Vanstone, S.: Handbook of Applied Cryptography. CRC Press, Florida (2001)
17. Menezes, A., Wu, Y.-H.: A polynomial representation for logarithms in $GF(q)$. Acta arithmetica 47, 255–261 (1986)
18. Menezes, A., Wu, Y.-H.: The Discrete Logarithm Problem in $GL(n, q)$. Ars Combinatoria 47, 22–32 (1997)
19. Odoni, R.W.K., Varadharajan, V., Sanders, P.W.: Public Key Distribution in Matrix Rings. Electronic Letters 20, 386–387 (1984)
20. Pohlig, S.C., Hellman, M.E.: An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance. IEEE Trans. Info. Theory 24, 106–110 (1978)
21. Shuhong, G., Gathen, J., Panario, D., Shoup, V.: Algorithms for Exponentiation in Finite Fields. Journal of Symbolic Computation 29-6, 879–889 (2000)

Word Oriented Cascade Jump $\sigma-$LFSR*

Guang Zeng, Yang Yang, Wenbao Han, and Shuqin Fan

Department of Applied Mathematics,
Zhengzhou Information Science and Technology Institute,
Zhengzhou 450002, China
sunshine_zeng@163.com, yangyang_wawa@sina.com,
wb.han@263.net, sq.fan@263.net

Abstract. Bit oriented cascade jump registers were recently proposed as building blocks for stream cipher. They are hardware oriented designed hence inefficient in software. In this paper word oriented cascade jump registers are presented based on the design idea of bit oriented cascade jump registers. Their constructions make use of special word oriented $\sigma-$LFSRs, which can be efficiently implemented on modern CPU and only require few memory. Experimental results show that one type of efficient word oriented cascade jump $\sigma-$LFSRs can be used as building blocks for software oriented stream cipher.

Keywords: Stream Cipher, Linear Feedback Shift Register(LFSR), Cascade Jump LFSR, $\sigma-$LFSR, Fast Software Encryption.

1 Introduction

Today stream ciphers are widely used in areas where the combination of security, performance and implementation complexity are of importance. One such area is wireless communication (GSM, 3G, IEEE802.11), where a low gate count in hardware implementation requirements prevail. Another area is efficient software encryption with speed about tens of gigabits per second. This opinion is supported by the eSTREAM project [1], which is an effort to identify new stream ciphers that might be interesting for widespread adoption.

LFSR is known to allow fast hardware implementation and produces sequences with a large period and good statistical properties. But the inherent linearity of these sequences results in susceptibility to algebraic attacks [2,3]. A well-known method to resist that is clock control. Due to the traditional clock control mechanism that either controls the LFSR irregularly clocking or shrinks or thins the output, clock controlled LFSRs have decreased rate of sequence generation since such LFSRs are usually stepped a few times to produce just one bit. A more effective method is dynamic feedback control, which is used in stream

* This work has been supported by a grant from the National High Technology Research and Development Program of China (No.2006AA01Z425), and National Natural Science Foundation of China (No.90704003, No.60503011), and National Basic Research Program of China (No.2007CB807902).

M. Bras-Amorós and T. Høholdt (Eds.): AAECC 2009, LNCS 5527, pp. 127–136, 2009.

cipher Pomaranch [4]. Pomaranch is a hardware oriented stream cipher and the core part is made of several bit oriented cascade jump registers [5], which was presented at SASC2004 Workshop. The main idea of cascade jump register [6] is to move the state of LFSR to another one over more than one step without having to step through all the intermediate states.

The nature of cascade jump register is bit oriented design, so it is suitable for hardware implementation while its software implementation is inefficient. The recent trend in software oriented stream cipher is towards word oriented design such as Sober [7], Snow [8], Sosemanuk [9] and etc. This allows them not only to be efficiently implemented in software but also to increase the throughput since words instead of bits output. $\sigma-$LFSRs [10,11] are one type of efficient word oriented LFSRs, which are constructed by few fundamental instructions of modern processor and hence have high software efficiency.

Combining the design ideas of bit oriented cascade jump register and word oriented $\sigma-$LFSR, we propose a cascade jump $\sigma-$LFSR. We examine one type of cascade jump $\sigma-$LFSRs on modern CPU architecture and make suggestions for design practices to maximize the speed, the security and decrease the implementation complexity of general cascade jump $\sigma-$LFSR. Finally, we give the jump index of our examples using Pollard Rho method.

2 Cascade Jump Register and $\sigma-$LFSR

Bit Oriented Cascade Jump Register. Linear Finite State Machine (LFSM) has been widely used in cryptography. Let $\mathbb{F}_2$ be the binary field and $\mathbb{F}_{2^m}$ be its extension field for some positive integer m. The state of the LFSM is represented by a vector $S_t = (r_{n-1}^t, \cdots, r_0^t) \in \mathbb{F}_{2^m}^n$ at time t, where $r_i^t \in \mathbb{F}_{2^m}$ and r_i^t denotes the content of the ith memory cell after t transitions. As the finite state machine is linear, transitions from one state to the next can be described by a multiplication of the state vector with a transition matrix $T \in M_{mn}(\mathbb{F}_2)$, i.e. $S_{t+1} = S_t \cdot T$, for $t \geq 0$. In practice, matrix T is generally chosen to have a special form for convenience of implementation.

If $m = 1$, the main idea of bit oriented cascade jump register is to find whether there exists an integer J, such that $T^J = T + I$ where I is a $mn \times mn$ identity matrix. If such a power of the transition matrix does exist, then one achieves the same effect as multiplying the state vector either by T^J or by $T + I$. Moreover, since changing T into $T+I$ is generally much simpler than rewiring T into T^J for an arbitrary transition matrix T, the LFSR is easy to jump. This modification of the transition matrix has very low complexity and is much more attractive and efficient than the method of rewiring the LFSR. Hence bit oriented cascade jump registers can be used as a dynamic part controlled by two-value sequence in clock control stream cipher.

Let $g(x) = det(xI + T)$ be the characteristic polynomial of T and $g^{\perp}(x) = g(x + 1)$ be the dual polynomial of $g(x)$. If there exists an integer J such that $T^J = T + I$, the value of J is called the jump index of g. The jump index always exists if $g(x)$ is a primitive polynomial over $\mathbb{F}_2$. Pomaranch uses 6 or 9 bit oriented

cascade jump registers for 80-bit and 128-bit key respectively. Both registers are built on 18 bits memory cells, and have primitive characteristic polynomial, i.e., when only clocked by zeros or ones they have a period of $2^{18}-1$. The transition matrix used in Pomaranch has a very special form, namely,

$$\begin{pmatrix} d_0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & d_1 & 0 & \cdots & 0 & t_1 \\ 0 & 1 & d_2 & \ddots & \vdots & \vdots \\ 0 & 0 & \ddots & \ddots & 0 & \vdots \\ \vdots & \vdots & \ddots & 1 & d_{16} & t_{16} \\ 0 & 0 & \cdots & 0 & 1 & d_{17}+t_{17} \end{pmatrix} \tag{1}$$

The major advantage of this type of transition matrix is the efficient hardware implementation. If wants to obtain word oriented cascade jump register, one only need choose $m \geq 2$, especially $m = 32$ or 64 for the word size of CPU correspondingly. But it should be noticed that the operation of $S_t \cdot T$ in computer is likely to be expensive unless the matrix T has a special form. In this case, $\sigma-$LFSRs are good choices because they are software oriented design.

Word Oriented $\sigma-$LFSR. The main idea of $\sigma-$LFSR [10,11] is to use a few the fundamental logic instructions, arithmetic instructions, or Single Instruction Multiple Data technique to construct high efficiency word oriented LFSR with good cryptographic properties. Let $C_0, \cdots, C_{n-1} \in M_m(\mathbb{F}_2)$ and the state at time t be $S_t = (r_{n-1}^t, \cdots, r_0^t)$, then the next state is $S_{t+1} = (r_{n-1}^{t+1}, \cdots, r_0^{t+1}) \in \mathbb{F}_{2^m}^n$ where $r_i^{t+1} = r_{i+1}^t$ for $0 \leq i \leq n-2$ and

$$r_{n-1}^{t+1} = C_0 r_0^t + C_1 r_1^t + \cdots + C_{n-1} r_{n-1}^t. \tag{2}$$

The system is called $\sigma-$LFSR of order n and matrix polynomial $f(x) = x^n + C_{n-1}x^{n-1} + \cdots + C_1 x + C_0 \in M_m(\mathbb{F}_2)[x]$ is called $\sigma-$polynomial of $\sigma-$LFSR. In fact, σ is a circular rotation operation defined in the following. Let $\alpha_0, \cdots, \alpha_{n-1}$ be a basis of $\mathbb{F}_{2^m}$ and for $\alpha \in \mathbb{F}_{2^m}$, there exists a vector $(a_0, a_1, \cdots, a_{m-1}) \in \mathbb{F}_2^m$ such that $\alpha = \sum_{i=0}^{m-1} a_i \alpha_i$. Operation σ over $\mathbb{F}_{2^m}$ is defined as

$$\sigma(\alpha) \triangleq a_{m-1}\alpha_0 + a_0\alpha_1 + \cdots + a_{m-2}\alpha_{m-1} \tag{3}$$

and σ is the Frobenius automorphism over $\mathbb{F}_{2^m}$ when the basis is normal, namely $\sigma(\alpha) = \alpha^2$. It is easy to see that σ is a linear transformation on $\mathbb{F}_{2^m}/\mathbb{F}_2$. From the view of linear transformation, we can obtain a new ring $\mathcal{A}_{2^m} = \mathbb{F}_{2^m}[\sigma]$ with adding σ to $\mathbb{F}_{2^m}$. Moreover we have $\mathcal{A}_{2^m} \cong M_m(\mathbb{F}_2)$ [11]. For the sake of obtaining fast speed implementation in software, we confine the coefficients in several special forms as follows.

1. AND Operation, $\wedge_V(\alpha) = \alpha \wedge V = \sum_{i=0}^{m-1} a_i c_i \alpha_i$, where $V = \sum_{i=0}^{m-1} c_i \alpha_i$;
2. Circular Rotation Operation, $\sigma^k(\alpha)$;
3. Left Shift Operation L, $\mathrm{L}(\alpha) = \sum_{i=1}^{m-1} a_i \alpha_{i-1}$

4. Right Shift Operation R, $\mathrm{R}(\alpha) = \sum_{i=0}^{m-2} a_i \alpha_{i+1}$;
5. LR Shift Combination Operation $\sqcup_{s,t} = \mathrm{L}^s + \mathrm{R}^t$.

If the output sequences generated by one $\sigma-$LFSR attain the maximal period, that $\sigma-$LFSR is called the primitive $\sigma-$LFSR. Suppose s^∞ is the sequence generated by primitive $\sigma-$LFSR, its bit coordinate sequences are all m-sequence with the same minimal polynomial[11]. In this case, the number of nonzero coefficients of the binary minimal polynomial is called the Hamming weight of $\sigma-$LFSR. This parameter is important, the closer to the half of the degree of its minimal polynomial, the better its properties. With these special operations above, we could find many primitive $\sigma-$LFSRs with good cryptographic properties. The following is a primitive $\sigma-$LFSR over $\mathbb{F}_{2^{16}}$.

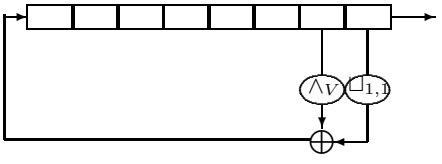


Fig. 1. $\sigma-$LFSR with $\sigma-$polynomial $x^8 + \wedge_{\mathtt{0x3fb6}} x + \sqcup_{1,1}$

The minimal polynomial of its coordinate sequences is $p(x)$ with Hamming weight 45, where $p(x) = x^{128} + x^{121} + x^{114} + x^{112} + x^{107} + x^{105} + x^{98} + x^{96} + x^{91} + x^{75} + x^{73} + x^{72} + x^{68} + x^{66} + x^{65} + x^{64} + x^{61} + x^{59} + x^{58} + x^{57} + x^{56} + x^{54} + x^{52} + x^{51} + x^{49} + x^{47} + x^{45} + x^{42} + x^{41} + x^{38} + x^{36} + x^{35} + x^{33} + x^{31} + x^{29} + x^{27} + x^{26} + x^{25} + x^{24} + x^{20} + x^{18} + x^{11} + x^6 + x^4 + 1$. Furthermore, this $\sigma-$LFSR has extremely fast speed, output speed is about over 25 Gbits/second on Pentium IV 3.0GHz CPU.

3 Word Oriented Cascade Jump $\sigma-$LFSR

Cascade jump register and $\sigma-$LFSR are effective methods in clock control design and word oriented LFSR design respectively. If two advantages could be combined together, we could strengthen the security of $\sigma-$LFSR and improve the performance of bit oriented cascade jump register. With this idea, we study the word oriented cascade jump $\sigma-$LFSR. The transition matrix of general word oriented cascade jump $\sigma-$LFSR is as follows.

$$T = \begin{pmatrix} 0 & 0 & 0 & 0 & C_0 \\ I & 0 & 0 & 0 & C_1 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & I & 0 & C_{n-2} \\ 0 & 0 & 0 & I & C_{n-1} \end{pmatrix}_{mn \times mn}$$

Only if T has maximal multiplicative order in $M_{mn}(\mathbb{F}_2)$, the output sequence generated by this $\sigma-$LFSR achieves the maximal period $2^{mn} - 1$, and in this

case, there exists a jump index J such that $T^J = T + I$. The following theorem may be helpful in searching for primitive T.

Theorem 1. *[10] Let $\sigma-$LFSR over $\mathbb{F}_{2^m}$ of order n with $\sigma-$polynomial $f(x) = x^n + C_{n-1}x^{n-1} + \cdots + C_1x + C_0 \in M_m(\mathbb{F}_2)[x]$ where $C_0 \in GL_m(\mathbb{F}_2)$ and $C_l = (c_l^{ij})_{m\times m}$ for $l = 0, 1, \cdots, n-1$. Let*

$$F(x) = (f^{ij}(x))_{m\times m} \in M_m(\mathbb{F}_2[x])$$

be the corresponding polynomial matrix of $f(x)$ where

$$f^{ij}(x) = \delta^{ij}x^n + \sum_{l=0}^{n-1} c_l^{ij}x^l \in \mathbb{F}_2[x],\ \delta^{ij} = \begin{cases} 1, & i = j; \\ 0, & i \neq j. \end{cases}$$

Then $\sigma-$LFSR has maximal period if and only if the determinant $|F(x)|$ is a primitive polynomial over $\mathbb{F}_2$ of degree mn.

A word oriented cascade jump $\sigma-$LFSR can be used under a binary control sequence, namely changing the transition matrix from T to $T + I$ according to the control sequence. We should choose $\sigma-$polynomial $f(x)$ to be primitive first, and furthermore $f^{\perp}(x) = f(x + 1)$ also be primitive. If $\sigma-$polynomial $f^{\perp}$ is not primitive, its output sequences will have short period with high probability and can not provide good properties in theory as maximal period sequence. This weakness will make this word oriented cascade jump $\sigma-$LFSR very dangerous if the control sequence is not balanced.

Apparently, the dual transition matrix of cascade jump $\sigma-$LFSRs will be identical to the transition matrix except for the entries on the main diagonal. Equivalently, adding ones to the entries on the main diagonal of the transition matrix equals to the jump index power of that matrix. Seeking for fast efficiency, we focus on one type of word oriented cascade $\sigma-$LFSRs with the following $\sigma-$polynomial over $M_{mn}(\mathbb{F}_2)$ as

$$f(x) = x^n + \wedge_V x^r + \sqcup_{s,t}, \tag{4}$$

where $V \in \mathbb{F}_{2^m}$, $0 < s, t < m-1$ and $(s+t)|m$. We have made exhaustive search for $m = 16$. Because of the symmetry of s, t, which means that if Eq. (4) is primitive, then $g(x) = x^n + \wedge_V x + \sqcup_{t,s}$ is also primitive, hence $s \leq t$ is required. Totally 6,815,744 $\sigma-$polynomials in the form of Eq. (4) are tested with $n = 8$. The following table lists the number of maximal period word oriented cascade jump $\sigma-$LFSRs.

Table 1. The Number of Primitive f and its Dual $f^{\perp}$

	$r=1$	$r=3$	$r=5$	$r=7$	$r=2,4,6$
f	2510	2415	2680	5666	0
$f^{\perp}$	2974	2334	2419	3570	0
Both	1382	48	54	46	0

Table 2. Hamming Weight of Some Good $\sigma-$polynomial

SN	$f(x)$	Weight(f)	Weight($f^{\perp}$)
1	$x^8 + \wedge_{\texttt{0x5806}} x + \sqcup_{3,5}$	9	33
2	$x^8 + \wedge_{\texttt{0xffc0}} x + \sqcup_{1,7}$	9	35
3	$x^8 + \wedge_{\texttt{0x9239}} x + \sqcup_{1,1}$	35	65
4	$x^8 + \wedge_{\texttt{0x29d8}} x + \sqcup_{1,1}$	33	75
5	$x^8 + \wedge_{\texttt{0x2ab9}} x^3 + \sqcup_{1,1}$	33	55
6	$x^8 + \wedge_{\texttt{0xdb73}} x^3 + \sqcup_{1,1}$	31	53
7	$x^8 + \wedge_{\texttt{0xb036}} x^5 + \sqcup_{1,3}$	21	57
8	$x^8 + \wedge_{\texttt{0xf00e}} x^5 + \sqcup_{1,1}$	29	65
9	$x^8 + \wedge_{\texttt{0x4e2c}} x^7 + \sqcup_{1,1}$	29	61
10	$x^8 + \wedge_{\texttt{0x623c}} x^7 + \sqcup_{1,1}$	39	69

There are total 1530 word oriented cascade jump $\sigma-$LFSRs satisfying our selection principle and the Hamming weight of $f^{\perp}(x)$ is superior to that of $f(x)$. In order to make the test results become visible, some examples with both f and $f^{\perp}$ primitive are listed below.

Example 1. The tenth word oriented cascade jump $\sigma-$LFSR in Table 2 has the best Hamming weight properties. Their software implementation are both very fast for simplicity of f and $f^{\perp}$. The original $\sigma-$LFSR and the modified cascade jump $\sigma-$LFSR are shown in Figure 2 and Figure 3 respectively.

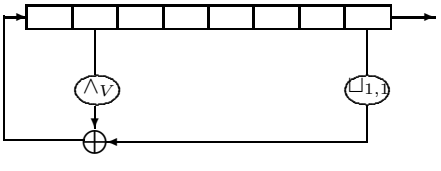


Fig. 2. $\sigma-$LFSR with $\sigma-$polynomial $x^8 + \wedge_{\texttt{0x623c}} x^7 + \sqcup_{1,1}$

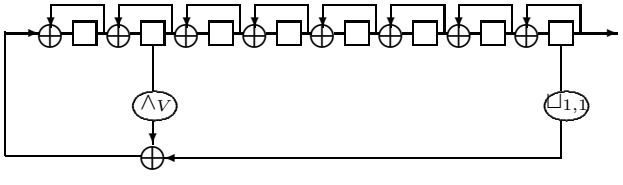


Fig. 3. The Modified Cascade Jump $\sigma-$LFSR in Fig.2

Searching for good word oriented $\sigma-$LFSRs as form in Eq. (4) is a time-consuming procedure. The following two necessary conditions may decrease the amount of search work a little.

Theorem 2. *If r is an even integer, $f(x)$ defined in Eq. (4) cannot be primitive.*

Proof: According Theorem 1, we need to calculate $|F(x)|$. Since every item in $f(x)$ is a square item over $\mathbb{F}_2[x]$, hence the determinant must be a square polynomial, not primitive polynomial. □

Theorem 3. *Polynomial $f(x)$ in Eq. (4) cannot be primitive if $s = t > 2$.*

Proof: Let $C_{m,s}$ be the $m \times m$ matrix as following

$$C_{m,s} = \begin{pmatrix} x_1 & 1 & & \\ & \ddots & 1 & \\ 1 & & \ddots & \ddots \\ & \ddots & & \ddots \\ & & 1 & x_m \end{pmatrix}_{m\times m} \tag{5}$$

In fact, with $x_i = x^n + c_i x^r$ for $0 \le i \le m-1$ in (5), we get $f(x)$ defined (4) in polynomial matrix expression. By Theorem 1, we need to calculate $|f(x)|$ and check whether it is a primitive polynomial. Next we will prove the determinant of $C_{m,s}$ is the multiplication of polynomials with the total number s when $s \ge 2$. Hence we know the determinant of $f(x)$ must be reducible, obviously not primitive.

For $C_{m,s}$ with $m \le 2s$, we can use the last $m-s$ rows to reduce the first $m-s$ rows, then use the Lapalace Theorem at first $m-s$ columns to compute the determinant, hence we can obtain a $s \times s$ diagonal matrix by suitable adjusting some columns. So the $|C_{m,s}|$ is the multiplication of s polynomials. This procedure is as follows

$$|C_{m,s}| = \begin{vmatrix} 0 & & x_1x_{s+1} & & \\ & \ddots & & & \\ & & & x_2x_{s+2} & \\ & 0 & & & \ddots \\ 1 & & x_{s+1} & & \\ & \ddots & & \ddots & \\ & 1 & & & x_m \end{vmatrix} = \begin{vmatrix} x_1x_{s+1} & & & \\ & x_2x_{s+2} & & \\ & & & \ddots \\ & & & \end{vmatrix}.$$

Next we will adopt mathematic induction method to prove. Suppose $|C_{m,s}|$ can be transformed to $s \times s$ diagonal matrix in the case $(k-1)s < m \le ks$ with some elementary row or column operations. Next we need to prove this is true for $ks < m \le (k+1)s$. In this case, we partition matrix $C_{m,s}$ as following where $C_{ks,s}$ is a $ks \times ks$ submatrix.

$$C_{m,s} = \begin{pmatrix} & & 0 & & \\ C_{ks,s} & & & \ddots & \\ & & & & 1 \\ 0 & 1 & x_{m-s+1} & & \\ & \ddots & \ddots & \ddots & \\ & 0 & 1 & & x_m \end{pmatrix}$$

According to our hypothesis, $C_{ks,s}$ can be transformed to $s \times s$ diagonal matrix during some elementary operations. We assume the result diagonal matrix is $diag(x'_1, \cdots, x'_s)$, hence after applying these transformations we have

$$|C_{m,s}| = \begin{vmatrix} x'_1 & & & * & & \\ & \ddots & & & \ddots & \\ & & x'_s & & & * \\ 1 & & & x_{m-s+1} & & \\ & \ddots & & & \ddots & \\ & & 1 & & & x_m \end{vmatrix}$$

Then using the similar elementary operations as that in the case of $m \leq 2s$, we can obtain another $s \times s$ diagonal matrix. Hence the determinant of $C_{m,s}$ is the multiplication of s polynomials. □

For finding other types of word oriented cascade jump $\sigma-$LFSRs, one should choose both $f(x)$ and $f^{\perp}(x)$ to be primitive. To be efficiently implemented in software, one needs to choose special matrix, which can be fast implemented on CPU. In the search procedure, one needs to carefully observe the test results to find some necessary condition for decreasing the search time. Finally, for the security aspect, one needs to pick out those whose Hamming weight are as large as possible.

4 Jump Index Calculation

The determination of the jump index of a given irreducible polynomial is a challenging task, especially for high degree polynomials, as it comes to finding the discrete logarithm in a large finite extension field. By employing a small amount of theory and a huge amount of computing power at the present time it is easy to determine the jump index of our examples as above.

Let $f(x)$ be the primitive $\sigma-$polynomial defined in Eq. (4), we briefly outline how we use the methods of Pohlig-Hellman and Pollard to compute $d = log_x(x+1)$ in finite field $\mathbb{F} = \mathbb{F}_{2^{128}}$. First we use the Pohlig-Hellman method [12] to reduce the DLP in $\mathbb{F}^*$ to the DLP in groups of prime group of order p, with p dividing $|\mathbb{F}^*| = 2^{128} - 1$. There are totally 9 subgroups due to $2^{128} - 1 = 3 * 5 * 17 * 257 * 641 * 65537 * 274177 * 6700417 * 67280421310721$. For each p dividing $|\mathbb{F}^*|$, we compute $d \bmod p$ as

$$(x+1)^{|\mathbb{F}^*|/p} = (x^{|\mathbb{F}^*|/p})^{d \bmod p}.$$

Each of these equations represents a DLP in the group of order p generated by $x^{|\mathbb{F}^*|/p}$. Having computed $d \bmod p$ for all prime factors p, we use the Chinese Remainder Theorem to determine d. To find $d \bmod p$ in subgroup we use the Pollard's Rho Method [13]. Assuming that G has group order $|G| = p$ with p prime, we divide G into four pairwise disjoint sets T_1, T_2, T_3, T_4 according to the

Table 3. Jump Indexes Result of Our Examples in Table 2

SN 1	22929362838758753427149407939876772168
dual of SN 1	17109370009926133818890776707951086976 2
SN 2	33632987543446883955251265142427660032 1
dual of SN 2	22167040042082436182152954769371836536 6
SN 3	13270467245833195151274491839174338983 6
dual of SN 3	63484811825393477931744641796814909436
SN 4	24269338571452150754055396087150413333 3
dual of SN 4	24217525397275954384661577393294632043 2
SN 5	30266368576397941998375310284498624875 4
dual of SN 5	30363583456125956220431035122577343503
SN 6	38103675587383110403116109629553749812
dual of SN 6	21701473805279016312844913568808878794
SN 7	29438726748767576609514068986360561044 0
dual of SN 7	33792940957167259832170208203377843788 6
SN 8	19636538106987442532043231829199232898 6
dual of SN 8	10608396032380298777638835482802539785 1
SN 9	33332060481899654896224778576908611522 0
dual of SN 9	38808670205409475437208962295677546997
SN 10	19920336790503412069861211140588601375 4
dual of SN 10	25867422615135782261650480803292578954 1

constant and x item coefficient. Let $f_P : G \to G$ as

$$f_P(y) \begin{cases} (x+1) * y, & y \in T_1; \\ y^2, & y \in T_2; \\ x * y, & y \in T_3; \\ x * (x+1) * y, & y \in T_4. \end{cases}$$

We choose a random number α in the range $\{1, 2, \cdots, |G|\}$, compute a starting element $y_0 = x^\alpha$, and put $y_{i+1} = f_P(y_i)$ for $i = 1, 2, \cdots$. This induces two triple integer sequences and we check whether y_{2i} matches y_i. If this is the case, we can obtain $d \bmod p$ with very high probability. If not, we repeat the whole computation with another starting value y_0; but this case is very rare for large group orders $|G| = p$. The expected value of iteration is about $1.229\sqrt{\pi|G|/2}$ and it take about one minute to compute one discrete logarithm on average.

The following table is the jump index of examples in Table 2. We list the jump index of $f(x)$ and its dual $f^{\perp}(x)$. From the table, we can see the jump indexes are extremely large, hence the cascade jump $\sigma-$LFSR is an effective method in clock control design indeed.

5 Conclusions

Modern processors have already gained much of better performance than their predecessors. This indicates the design criteria of software oriented cryptographic algorithm should fully benefit from these changes. We have illustrated one type word

oriented cascade jump $\sigma-$LFSR, which is software oriented design and hence has fast speed. Although not shown in great detail, it should be clear that these word oriented cascade jump $\sigma-$LFSR constructions can be used as building blocks in software oriented clock control stream cipher in a very efficient way.

References

1. ECRYPT, eSTREAM: ECRYPT Stream Cipher Project, `http://www.ecrypt.eu.org/stream/`
2. Courtois, N., Meier, W.: Algebraic attacks on stream ciphers with liners feedback. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 345–359. Springer, Heidelberg (2003)
3. Courtois, N.: Fast algebraic attacks on stream ciphers with linear feedback. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 176–194. Springer, Heidelberg (2003)
4. Jansen, C.J.A., Helleseth, T., Kholosha, A.: Cascade jump controlled sequence generator and Pomaranch stream cipher (Version 3), eSTREAM, ECRYPT Stream Cipher Project (2007)
5. Jansen, C.J.A.: Streamcipher design: Make your LFSRs jump! In: The State of the Art of Stream Ciphers, Workshop Record, Brugge, Belgium, October 2004, pp. 94–108 (2004)
6. Jansen, C.J.A.: Stream cipher design based on jumping finite state machines. Cryptology ePrint Archive, Report 2005/267 (2005), `http://eprint.iacr.org/2005/267/`
7. Hawkes, P., Rose, G.G.: Primitive Specification and Supporting Documentation for SOBER-t32 Submission to NESSIE. In: First NESSIE Workshop (2000)
8. Ekdahl, P., Johansson, T.: A New Version of the Stream Cipher SNOW. In: Nyberg, K., Heys, H.M. (eds.) SAC 2002. LNCS, vol. 2595, pp. 47–61. Springer, Heidelberg (2003)
9. Berbain, C., Billet, O., et al.: Sosemanuk, a fast software-oriented stream cipher. ECRYPT Stream Cipher Project (2007)
10. Zeng, G., Han, W., He, K.C.: High efficiency feedback shift register: $\sigma-$LFSR. Cryptology ePrint Archive, Report 2007/114 (2007)
11. Zeng, G., He, K.C., Han, W.: A trinomial type of $\sigma-$LFSR oriented toward software implementation. Science in China Series F-Information Sciences 50(3), 359–372 (2007)
12. Pohlig, S.C., Hellman, M.E.: An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance. IEEE Transactions on Information Theory 24, 106–110 (1978)
13. Pollard, J.M.: Monte Carlo methods for index computation (mod p). Mathematics of Computation 32(143), 918–924 (1978)
14. Teske, E.: Speeding Up Pollard's Rho Method for Computing Discrete Logarithms. In: Buhler, J.P. (ed.) ANTS 1998. LNCS, vol. 1423, pp. 541–554. Springer, Heidelberg (1998)

On Some Sequences of the Secret Pseudo-random Index j in RC4 Key Scheduling

Riddhipratim Basu[1], Subhamoy Maitra[1], Goutam Paul[2], and Tanmoy Talukdar[1]

[1] Indian Statistical Institute*, 203 B T Road, Kolkata 700 108, India
rpbasu.riddhi@gmail.com, subho@isical.ac.in, tanmoy.talukdar@gmail.com
[2] Department of Computer Science and Engineering, Jadavpur University, Kolkata 700032
goutam_paul@cse.jdvu.ac.in

Abstract. RC4 Key Scheduling Algorithm (KSA) uses a secret pseudo-random index j which is dependent on the secret key. Let S_N be the permutation after the complete KSA of RC4. It is known that the value of j in round $y+1$ can be predicted with high probability from $S_N[y]$ for the initial values of y and from $S_N^{-1}[y]$ for the final values of y. This fact has been exploited in several recent works on secret key recovery from S_N. In this paper, we perform extensive analysis of some special sequences of indices corresponding to the j values that leak useful information for key recovery. We present new theoretical results on the probability and the number of such sequences. As an application, we explain a new secret key recovery algorithm that can recover a 16 bytes secret key with a success probability of 0.1409. Our strategy has high time complexity at this point and requires further improvement to be feasible in practice.

Keywords: Bias, Cryptanalysis, Filter, Key Recovery, Permutation, RC4, Sequence, Stream Cipher.

1 Introduction

The RC4 stream cipher has been designed by Ron Rivest for RSA Data Security in 1987, and was a propriety algorithm until 1994. The RC4 cipher has two components, namely, the Key Scheduling Algorithm (KSA) and the Pseudo-Random Generation Algorithm (PRGA). The KSA expands a secret key $k[0 \ldots l-1]$ into an array $K[0 \ldots N-1]$ such that $K[y] = k[y \bmod l]$ for any y, $0 \leq y \leq N-1$. Using this key, an identity permutation $S[0 \ldots N-1]$ of $\{0, 1, \ldots N-1\}$ is scrambled using two indices i, j. The PRGA uses this permutation to generate pseudo-random keystream bytes $z_1, z_2, \ldots$, that are bitwise XOR-ed with the plaintext to generate the ciphertext at the sender end and bitwise XOR-ed with the ciphertext to get back the plaintext at the receiver end.

Any addition used related to the RC4 description is in general addition modulo N unless specified otherwise.

* The first and last authors worked for this paper during the winter break between the semesters (2008–2009) in their Bachelor of Statistics course.

M. Bras-Amorós and T. Høholdt (Eds.): AAECC 2009, LNCS 5527, pp. 137–148, 2009.

KSA(K)
Initialization:
For $i = 0, \ldots, N-1$
$S[i] = i$;
$j = 0$;
Scrambling:
For $i = 0, \ldots, N-1$
$j = (j + S[i] + K[i])$;
Swap$(S[i], S[j])$;

PRGA(S)
Initialization:
$i = j = 0$;
Keystream Generation Loop:
$i = i + 1$;
$j = j + S[i]$;
Swap$(S[i], S[j])$;
$t = S[i] + S[j]$;
Output $z = S[t]$;

We use subscript r to the variables S, i and j to denote their updated values in round r, where $1 \leq r \leq N$. According to this notation, S_N is the permutation after the completion of the KSA. By S_0 and j_0, we mean the initial permutation and the initial value of the index j respectively before the KSA begins. We use the notation S^{-1} for the inverse of the permutation S, i.e., if $S[y] = v$, then $S^{-1}[v] = y$. By $f_y(K)$, we denote the expression $\frac{y(y+1)}{2} + \sum_{x=0}^{y} K[x]$, $0 \leq y \leq N-1$.

RC4 is the most popular and widely deployed software stream cipher. It is used in network protocols such as SSL, TLS, WEP and WPA, and also in Microsoft Windows, Lotus Notes, Apple AOCE, Oracle Secure SQL etc. The cipher is simple in structure, easy to implement and efficient in throughput. Yet after more than twenty years of cryptanalysis, the cipher is still not completely broken and is of great interest to the cryptographic community. One may refer to [7,9,13] and the references therein for a detailed exposition on RC4 cryptanalysis.

In [12], it has been argued that the most likely value of the y-th element of the permutation after the KSA for the first few values of y is given by $S_N[y] = f_y(K)$. The experimental values of the probabilities $P(S_N[y] = f_y(K))$ for y from 0 to 47 were reported in [12] without any theoretical proof. The expressions for the probabilities $P(S_N[y] = f_y(K))$ for all values of the index y in $[0, N-1]$ were theoretically derived in [10, Section 2] and the work [11, Section 2.1] generalized it to derive $P(S_r[y] = f_y(K))$ for all rounds r, $1 \leq r \leq N$.

It has been shown in [6] that the bytes $S_N[y]$, $S_N[S_N[y]]$, $S_N[S_N[S_N[y]]]$, and so on, are biased to $f_y(K)$. In particular, they showed that $P(S_N[S_N[y]] = f_y(K))$ decreases from 0.137 for $y = 0$ to 0.018 for $y = 31$ and then slowly settles down to 0.0039 (beyond $y = 48$).

In [10, Section 3], for the first time an algorithm is presented to recover the complete key from the final permutation after the KSA using the Roos' biases, without any assumption on the key or IV. The algorithm recovers some secret key bytes by solving sets of independent equations of the form $S_N[y] = f_y(K)$ and the remaining key bytes by exhaustive search. Subsequently, the work [2] additionally considered differences of the above equations and reported better results. Recently, [1] has accumulated the ideas in the earlier works [2,6,10] along with some additional new results to devise a more efficient algorithm for key recovery. After the publication of [2,10], another work [11] which has been performed independently and around the same time as [1] shows that each byte

of S_N actually reveals secret key information. The key recovery algorithm of [11] sometimes outperform that of [2]. A recent work [3] starts with the equations of [2] and considers a bit-by-bit approach to key recovery.

Getting back the secret key from the final permutation after the KSA is an important problem in RC4 cryptanalysis. State recovery from keystream [4,8,14], another important attack on RC4, can be turned into a key recovery attack if key reconstruction from S_N is possible in a time complexity less than that of state recovery attacks. Moreover, in certain applications such as WEP [5], the key and the IV are combined in such a way that the secret key can be easily extracted from the session key. For these applications, if one can recover the session key from the permutation then it is possible to get back the secret key. In that case, for subsequent sessions where the same secret key would be used with different known IV's, the RC4 encryption would be completely insecure.

In this paper, we study sequences of j_{y+1} values corresponding to the rounds $y+1$ when i takes the value y, $0 \leq y \leq N-1$. We concentrate on a span equal to the key length at each end of the permutation S_N. Based on our sequence analysis in Section 2, we present a novel bidirectional search algorithm for secret key recovery in Section 3. Our algorithm can recover secret keys of length 16 bytes with a success probability of 0.1409, which is almost two times the currently best known value of 0.0745 reported in [1]. The time reported in [1] for probability 0.0745 is 1572 seconds, but it is not clear how the complexity of [1] would grow with increase in probability. Our analysis provides better probability and a method to achieve that, but further tuning is required to make our strategy feasible in practice as the time complexity is very high.

Our main idea is to guess $K[0], K[1], \ldots$ using the left end of the permutation and guess $K[l-1], K[l-2], \ldots$ using the right end of the permutation simultaneously. When some key bytes are guessed, the KSA may be run (in the forward or in the backward direction, depending on the location of the guessed key bytes) until a known j_{y+1} is encountered. If this matches with the computed j_{y+1}, then the guessing is continued, else the partial key is discarded. Thus, the indices y for which j_{y+1}'s are known act as "filters" for separating out the wrong keys from the correct candidates.

All the existing works on key recovery from the final permutation after the KSA try to guess the entire key together. Our bidirectional key retrieval makes use of the fact that once certain key bytes are known, the choice of possible values of the remaining key bytes become restricted. To our knowledge, the concept of "filtering technique" as well as the notion of bidirectional key search is a completely different and new approach towards secret key recovery of RC4.

2 Theoretical Analysis of Sequences of Filter Indices

In this section, we study the structure and algebraic properties of the sequence of indices that act as filters for validating the secret key guesses. For simplicity, we restrict l to be a factor of N.

The secret index j is pseudo-random and may be considered uniformly distributed in $[0, N-1]$. In general, if the values for a sequence of m many j's are guessed randomly, the success probability is N^{-m}. For $N = 256$ and $m = 8$, this value turns out to be 2^{-64}. Whereas, using our theoretical framework, we can guess a sequence of j values with very good probability (see Table 1).

Definition 1 (Event A_y). *For $0 \leq y \leq N-1$, event A_y occurs if and only if the following three conditions hold.*

1. $S_y[y] = y$ *and* $S_y[j_{y+1}] = j_{y+1}$.
2. $j_{y+1} \geq y$.
3. $S_N[y] = S_{y+1}[y]$.

Proposition 1. $p_y = P(A_y) = (\frac{N-y}{N}) \cdot (\frac{N-2}{N})^y \cdot (\frac{N-1}{N})^{N-1-y}$, $0 \leq y \leq N-1$.

Proof. Let us analyze the conditions mentioned in the definition of A_y.

1. $S_y[y] = y$ and $S_y[j_{y+1}] = j_{y+1}$ occur if $j_t \notin \{y, j_{y+1}\}$, for $1 \leq t \leq y$, the probability of which is $(\frac{N-2}{N})^y$.
2. $P(j_{y+1} \geq y) = \frac{N-y}{N}$.
3. $S_N[y] = S_{y+1}[y]$, if $j_t \neq y$, $\forall t \in [y+2, N]$. This happens with probability $(\frac{N-1}{N})^{N-y-1}$.

Multiplying the above three probabilities, we get the result. □

Note that the event A_y implies $j_{y+1} = S_N[y]$ and Proposition 1 is a variant of [10, Lemma 2] that relates j_{y+1} and $S_N[y]$.

Definition 2 (Event B_y). *For $0 \leq y \leq N-1$, event B_y occurs if and only if the following three conditions hold.*

1. $S_y[y] = y$.
2. $j_{y+1} \leq y$.
3. $S_N[j_{y+1}] = S_{y+1}[j_{y+1}]$.

Proposition 2. $p'_y = P(B_y) = (\frac{y+1}{N}) \cdot (\frac{N-1}{N})^{N-1}$, $0 \leq y \leq N-1$.

Proof. Let us analyze the conditions mentioned in the definition of B_y.

1. $S_y[y] = y$ occurs if $j_t \neq y$ for $1 \leq t \leq y$, the probability of which is $(\frac{N-1}{N})^y$.
2. $P(j_{y+1} \leq y) = \frac{y+1}{N}$.
3. $S_N[j_{y+1}] = S_{y+1}[j_{y+1}]$, if $j_t \neq j_{y+1}$, $\forall t \in [y+2, N]$. This happens with probability $(\frac{N-1}{N})^{N-y-1}$.

Multiplying the above three probabilities, we get the result. □

Note that the event B_y implies $j_{y+1} = S_N^{-1}[y]$, $0 \leq y \leq N-1$. B_y is the same as the event $E_1(y)$ defined in [11] and the proof of Proposition 2 has been presented as part of the proof of [11, Theorem 3].

Definition 3 (Filter). *An index y in $[0, N-1]$ is called a filter if either of the following two holds.*

1. $0 \leq y \leq \frac{N}{2} - 1$ *and event A_y occurs.*
2. $\frac{N}{2} \leq y \leq N-2$ *and event B_y occurs.*

Definition 4 (Bisequence). *Suppose j_N is known. Then a sequence of at least $(t+t'+1)$ many filters is called a (t, t')-bisequence if the following three conditions hold.*

1. *Exactly t many filters $0 \leq i_1 < i_2 < \ldots < i_t \leq \frac{l}{2} - 1$ exist in the interval $[0, \frac{l}{2} - 1]$.*
2. *Exactly t' many filters $N - 1 - \frac{l}{2} \leq i'_{t'} < \ldots < i'_1 \leq N-2$ exist in the interval $[N - 1 - \frac{l}{2}, N-2]$.*
3. *Either a filter i_{t+1} exist in the interval $[\frac{l}{2}, l-1]$ or a filter $i'_{t'+1}$ exist in the interval $[N-1-l, N-2-\frac{l}{2}]$.*

Lemma 1. *Given a set F_t of t many indices in $[0, \frac{l}{2} - 1]$, a set $B_{t'}$ of t' many indices in $[N-1-\frac{l}{2}, N-2]$ and an index x in $[\frac{l}{2}, l-1] \cup [N-1-l, N-2-\frac{l}{2}]$, the probability of the sequence of indices $F_t \cup B_{t'} \cup \{x\}$ to be a (t, t')-bisequence is*

$$\left(\prod_{i_u \in F_t} p_{i_u} \prod_{i_v \in [0, l-1] \setminus (F_t \cup \{x\})} q_{i_v} \prod_{i'_{u'} \in B_{t'}} p'_{i'_{u'}} \prod_{i'_{v'} \in [N-1-l, N-2] \setminus (B_{t'} \cup \{x\})} q'_{i'_{v'}} \right) \tilde{p}_x,$$

where $q_y = 1 - p_y$, $q'_y = 1 - p'_y$ for $0 \leq y \leq N-1$ and $\tilde{p}_x = p_x$ or p'_x according as $x \in [\frac{l}{2}, l-1]$ or $[N-1-l, N-2-\frac{l}{2}]$ respectively.

Proof. According to Definition 4, $F_t \cup B_{t'} \cup \{x\}$ would be an (t, t')-bisequence, if the indices in F_t and $B_{t'}$ and the index x are filters and the indices in $([0, l-1] \cup [N-1-l, N-2]) \setminus (F_t \cup B_{t'} \cup \{x\})$ are non-filters. Hence, the result follows from Propositions 1 and 2. □

Definition 5 (Critical Filters). *The last filter i_t within the first $\frac{l}{2}$ indices and the first filter i'_t within the last $\frac{l}{2}$ indices for an (t, t')-bisequence are called the left critical and the right critical filters respectively. Together, they are called the critical filters.*

Definition 6 (Favourable Bisequence). *A (t, t')-bisequence is called d-favourable, $d \leq \frac{l}{2}$, if the following seven conditions hold.*

1. $i_1 + 1 \leq d$.
2. $i_{u+1} - i_u \leq d$, $\forall u \in [1, t-1]$.
3. $\frac{l}{2} - 1 - i_t \leq d$.
4. $N - 1 - i'_1 \leq d$.
5. $i'_v - i'_{v+1} \leq d$, $\forall v \in [1, t'-1]$.
6. $i'_{t'} - (N - 1 - \frac{l}{2}) \leq d$.
7. $i_{t+1} - i_t \leq d$ *or* $i'_{t'} - i'_{t'+1} \leq d$.

Let us interpret the conditions mentioned in Definition 6. For ease of reference, we introduce a dummy index -1 to the left of index 0. Conditions 1 and 4 ensure that the first filter to the right of index -1 and the first filter to the left of index $N-1$ are at most at a distance d from indices -1 and $N-1$ respectively. Conditions 2 and 5 ensure that the distance between any two consecutive filters to the left of left critical filter and to the right of right critical filter is at most d. Conditions 3 and 6 ensure that the left and right critical filters are at most at a distance d from indices $\frac{l}{2}-1$ and $N-1-\frac{l}{2}$ respectively. Consider the first filter i_{t+1} to the right of the left critical filter and the first filter $i'_{t'+1}$ to the left of the right critical filter. For a (t,t')-bisequence, at least one of i_{t+1} and $i'_{t'+1}$ must exist. Condition 7 ensures that whichever exist (and at least one of the two, if both exist) is located at most at a distance d from the corresponding critical filter.

Lemma 2. *The number of distinct sequences of indices in* $[0,\frac{l}{2}-1]\cup[N-1-\frac{l}{2},N-2]$ *satisfying the conditions 1 through 6 of Definition 6 is* $\sum_{\delta\leq d,\delta'\leq d}\sum_{t\leq\frac{l}{2}-\delta,t'\leq\frac{l}{2}-\delta'}c(\delta,t)c(\delta',t')$, *where* $\delta=\frac{l}{2}-1-i_t$, $\delta'=i'_{t'}-(N-1-\frac{l}{2})$, *and* $c(\delta,t)$ *is the coefficient of* $x^{\frac{l}{2}-\delta-t}$ *in* $(1+x+\ldots+x^{d-1})^t$.

Proof. Let $x_1=i_1$, $x_u=i_u-i_{u-1}-1$ for $2\leq u\leq t$ be the number of non-filters between two consecutive filters in $[0,i_t]$. The total number of non-filters in the interval $[0,i_t]$ is $i_t-(t-1)=(\frac{l}{2}-1-\delta)-(t-1)=\frac{l}{2}-\delta-t$. Thus, the number of distinct sequences of indices in $[0,\frac{l}{2}-1]$ satisfying conditions 1, 2 and 3 of Definition 6 is the same as the number of non-negative integral solutions of $x_1+x_2+\ldots+x_t=\frac{l}{2}-\delta-t$, where $0\leq x_u\leq d-1,\forall u\in[1,t]$. The number of solutions is given by $c(\delta,t)$. Similarly, the number of distinct sequences of indices in $[N-1-\frac{l}{2},N-2]$ satisfying conditions 4, 5 and 6 of Definition 6 is $c(\delta',t')$. Hence, the number of distinct sequences of indices in $[0,\frac{l}{2}-1]\cup[N-1-\frac{l}{2},N-2]$ satisfying the conditions 1 through 6 is $\sum_{\delta\leq d,\delta'\leq d}\sum_{t\leq\frac{l}{2}-\delta,t'\leq\frac{l}{2}-\delta'}c(\delta,t)c(\delta',t')$. □

Theorem 1. *The probability of existence of a d-favourable* (t,t')*-bisequence in* $[0,l-1]\cup[N-1-l,N-2]$ *is*

$$\pi_d=\sum_{t,t'}\sum_{F_t,B_{t'}}\prod_{i_u\in F_t}p_{i_u}\prod_{i_v\in[0,\frac{l}{2}-1]\setminus F_t}q_{i_v}\prod_{i'_{u'}\in B_{t'}}p'_{i'_{u'}}$$
$$\prod_{i'_{v'}\in[N-1-\frac{l}{2},N-2]\setminus B_{t'}}q'_{i'_{v'}}\Big(1-\prod_{\substack{y\in[\frac{l}{2},i_t+d]\cup\\ [i_{t'}-d,N-2-\frac{l}{2}]}}\tilde{q}_y\Big),$$

where the sum is over all $t,t',F_t,B_{t'}$ *such that the sequence of indices* $F_t\cup B_{t'}$ *satisfy the conditions 1 through 6 in Definition 6 and* $\tilde{q}_y=q_y$ *or* q'_y *according as* $y\in[\frac{l}{2},i_t+d]$ *or* $[i_{t'}-d,N-2-\frac{l}{2}]$ *respectively.*

Proof. Immediately follows from Lemma 1. The term $(1-\prod_{y\in[i_t+1,i_t+d]\cup[i_{t'}-d,i_{t'}-1]}\tilde{q}_y)$ accounts for condition 7 of Definition 6. □

Using the definitions of $c(\delta,t)$, $c(\delta',t')$ introduced in Theorem 2, we can approximate the probability expression presented in Theorem 1 as follows.

Corollary 1. *If $l \leq 16$ (i.e., the key length is small), then we have*

$$\pi_d \approx \sum_{\delta \leq d, \delta' \leq d} \sum_{t \leq \frac{l}{2}-\delta, t' \leq \frac{l}{2}-\delta'} c(\delta, t) p^t q^{\frac{l}{2}-t} c(\delta', t') p'^{t'} q'^{\frac{l'}{2}-t'} (1 - q^{d-\delta} q'^{d-\delta'}),$$

where $p = \frac{2}{l} \sum_{y=0}^{\frac{l}{2}-1} p_y$, $p' = \frac{2}{l} \sum_{y=N-1-\frac{l}{2}}^{N-2} p'_y$, $q = 1 - p$, $q' = 1 - p'$.

Proof. Approximating each p_y in the left half by the average p of the first $\frac{l}{2}$ many p_y's and each p'_y in the right half by the average p' of the last $\frac{l}{2}$ many p'_y's, we get the result. The ranges of the variables in the summation account for conditions 3 and 6 of Definition 6. The term $c(\delta, t) p^t q^{\frac{l}{2}-t}$ accounts for the conditions 1 and 2, the term $c(\delta', t') p'^{t} q'^{\frac{l'}{2}-t'}$ accounts for the conditions 4 and 5, and the term $(1 - q^{d-\delta} q'^{d-\delta'})$ accounts for the condition 7 of Definition 6. □

In Table 1, we compare the theoretical estimates of π_d from Corollary 1 with the experimental values obtained by running the RC4 KSA with 10 million randomly generated secret keys of length 16 bytes.

Table 1. Theoretical and experimental values of π_d vs. d with $l = 16$

d	2	3	4	5	6
Theoretical	0.0055	0.1120	0.3674	0.6194	0.7939
Experimental	0.0052	0.1090	0.3626	0.6152	0.7909

The results indicate that our theoretical values match closely with the experimental values.

Theorem 2. *The number of distinct d-favourable (t, t')-bisequences containing exactly $f \leq 2l$ filters is*

$$\sum_{\delta \leq d, \delta' \leq d} \sum_{t \leq \frac{l}{2}-\delta, t' \leq \frac{l}{2}-\delta'} c(\delta, t) c(\delta', t') \sum_{s=1}^{2d-\delta-\delta'} \binom{2d-\delta-\delta'}{s} \binom{l-2d+\delta+\delta'}{f-t-t'-s}.$$

Proof. By Lemma 2, the number of distinct sequences of indices in $[0, \frac{l}{2} - 1] \cup [N - 1 - \frac{l}{2}, N - 2]$ satisfying the conditions 1 through 6 of Definition 6 is $\sum_{\delta \leq d, \delta' \leq d} \sum_{t \leq \frac{l}{2}-\delta, t' \leq \frac{l}{2}-\delta'} c(\delta, t) c(\delta', t')$. The justification for the two binomial coefficients with a sum over s is as follows. For condition 7 to be satisfied, at least one out of $2d - \delta - \delta'$ indices in $[\frac{l}{2}, \frac{l}{2} - 1 + d - \delta] \cup [N - 1 - \frac{l}{2} - (d - \delta'), N - 2 - \frac{l}{2}]$ must be a filter. Let the number of filters in this interval be $s \geq 1$. So the remaining $f - t - t' - s$ many filters must be amongst the remaining $l - 2d + \delta + \delta'$ many indices in $[\frac{l}{2} + d - \delta, l - 1] \cup [N - 1 - l, N - 2 - \frac{l}{2} - (d - \delta')]$ □

Corollary 2. *The number of distinct d-favourable (t, t')-bisequences containing at most $F \leq 2l$ filters is*

$$C_{d,F} = \sum_{\delta \leq d, \delta' \leq d} \sum_{t \leq \frac{l}{2}-\delta, t' \leq \frac{l}{2}-\delta'} c(\delta, t) c(\delta', t')$$

$$\sum_{s=1}^{2d-\delta-\delta'} \binom{2d-\delta-\delta'}{s} \sum_{r=0}^{F-t-t'-s} \binom{l-2d+\delta+\delta'}{r}.$$

Proof. In addition to $s \geq 1$ filters in $[\frac{l}{2}, \frac{l}{2}-1+d-\delta] \cup [N-1-\frac{l}{2}-(d-\delta'), N-2-\frac{l}{2}]$, we need r more filters in $[\frac{l}{2}+d-\delta, l-1] \cup [N-1-l, N-2-\frac{l}{2}-(d-\delta')]$, where $t+t'+s+r \leq F$. □

Corollary 3. *The number of distinct d-favourable (t,t')-bisequences in $[0, l-1] \cup [N-1-l, N-2]$ (containing at most $2l$ filters) is*

$$C_{d,2l} = \sum_{\delta \leq d, \delta' \leq d} \sum_{t \leq \frac{l}{2}-\delta, t' \leq \frac{l}{2}-\delta'} c(\delta,t)c(\delta',t')(1-2^{-2d+\delta+\delta'})2^l.$$

Proof. Substitute $F = 2l$ in Corollary 2 and simplify.

Alternatively, the number of distinct sequences satisfying the conditions 1 through 6 is $\sum_{\delta,\delta'} \sum_{t,t'} c(\delta,t)c(\delta',t')2^l$ and out of these the number of sequences violating the condition 7 is $\sum_{\delta,\delta'} \sum_{t,t'} c(\delta,t)c(\delta',t')2^{l-2d+\delta+\delta'}$. Subtracting, we get the result. Note that the term 2^r stands for the number of ways in which a set of r indices may be a filter or a non-filter. □

As an illustration, we present $C_{d,F}$ for different d and F values with $l = 16$ in Table 2.

Table 2. $C_{d,F}$ for different F and d values with $l = 16$

F	32	20	16	12	8
$d=2$	$2^{27.81}$	$2^{27.31}$	$2^{24.72}$	$2^{18.11}$	$2^{3.58}$
$d=3$	$2^{30.56}$	$2^{30.38}$	$2^{29.02}$	$2^{24.86}$	$2^{15.95}$
$d=4$	$2^{31.47}$	$2^{31.35}$	$2^{30.38}$	$2^{27.17}$	$2^{20.14}$

Though we have assumed that l divides N in our analysis, the same idea works when l is not a factor of N. One only needs to map the indices from the right end of the permutation to the appropriate key bytes.

3 Application of Filter Sequences in Bidirectional Key Search

Here, we demonstrate how the theoretical framework of filter sequences developed in the previous section can be used to devise a novel key retrieval algorithm.

Suppose that after the completion of the RC4 KSA, the final permutation S_N and the final value j_N of the index j is known. The update rule for KSA is $j_{y+1} = j_y + S_y[y] + K[y]$, $0 \leq y \leq N-1$. Since S_0 is identity permutation and $j_0 = 0$, we have $j_1 = K[0]$. Thus, if index 0 is a filter, then $K[0]$ is known. Again, if index $N-2$ is a filter, then j_{N-1} is known and hence $K[l-1] = K[N-1] = j_N - j_{N-1} - S_{N-1}[N-1] = j_N - j_{N-1} - S_N[j_N]$ is also known. Moreover, if two

consecutive indices $y-1$ and y are filters, $1 \leq y \leq N-2$, then j_y and j_{y+1} are known and $S_y[y] = y$ from the definition of filters. Thus, the correct value of the key byte $K[y \bmod l]$ can be derived as $K[y \bmod l] = j_{y+1} - j_y - y$.

For full key recovery, apart from the knowledge of the final permutation S_N and the final value j_N of the index j, we also need to assume that a *d-favourable* (t, t')-*bisequence* exist. Given such a sequence, our algorithm does not require to guess more than d many key bytes at a time. For faster performance, we like to keep d small (typically, 4).

Our basic strategy for full key recovery is as follows. First, we perform a partial key recovery using filters in the interval $[0, M-1] \cup [N-1-M, N-2]$, where $l \leq M \leq 2l$, such that the correct values of at least n_{corr} out of l key bytes are uniquely determined and at least n_{in} out of n_{corr} are found from the filters in $[0, l-1] \cup [N-1-l, N-2]$. Let $\mathcal{P}$ be the set of partially recovered key bytes, each of which is associated with a single value, namely, the correct value of itself.

Next, we build a frequency table for the $l - n_{corr}$ many unknown key bytes as follows. We guess one j_{y+1} from the 8 values

$$S_N[y],\ S_N^{-1}[y],\ S_N[S_N[[y]],\ S_N^{-1}[S_N^{-1}[y]],\ S_N[S_N[S_N[y]]],$$

$S_N^{-1}[S_N^{-1}[S_N^{-1}[y]]]$, $S_N[S_N[S_N[S_N[y]]]]$ and $S_N^{-1}[S_N^{-1}[S_N^{-1}[S_N^{-1}[y]]]]$. From two successive j values j_y and j_{y+1}, $8 \times 8 = 64$ candidates for the key byte $K[y]$ are obtained. This part is similar to the $6 \times 6 = 36$ candidate keys from up to three levels of nested indexing of S_N and S_N^{-1} as in [1]. These candidates are weighted according to their probabilities. We split the 256 possible values of each unknown key byte $K[y]$ into 4 sets of size 64 each. Let G_y be the set of values obtained from the above frequency table and let $H_{1,y}, H_{2,y}, H_{3,y}$ be the other sets, called *bad* sets. For $0 \leq y \leq l-1$, let $g_y = P(K[y] \in G_y)$. Since empirical results show that g_y is almost the same for each y, we would be using the average $g = \frac{1}{l}\sum_{y=0}^{l-1} g_y$ in place of each g_y.

After this, we perform a bidirectional search from both ends several times, each time with exactly one of the sets $\{G_y, H_{1,y}, H_{2,y}, H_{3,y}\}$ for each unknown key byte. We decide to use the bad sets for at most b of the unknown key bytes. The search algorithm, called *BidirKeySearch*, takes as inputs the final permutation S_N, the final value j_N, the partially recovered key set $\mathcal{P}$, a set $\mathcal{S}$ of $l - n_{corr}$ many candidate sets each of size 64 and a d-favourable (t, t')-bisequence $\mathcal{B}$.

For the left end, first we guess the key bytes $K[0], \ldots, K[i_1]$ except those in $\mathcal{P}$. Starting with S_0 and j_0, we run the KSA until we obtain S_{i_1+1} and j_{i_1+1}. Since the correct value of j_{i_1+1} is known, the tuples leading to the correct j_{i_1+1} form a set T_{i_1} of candidate solutions for $K[0 \ldots i_1]$. These tuples are said to "pass the filter" i_1. Similarly, starting with j_{i_1+1} and each state S_{i_1+1} associated with each tuple in T_{i_1}, we guess $K[i_1+1], \ldots, K[i_2]$ except those in $\mathcal{P}$. Thus, we obtain a set T_{i_2} of candidate solutions for $K[0 \ldots i_2]$ passing the filter i_2, and so on until we reach a stage where we have a set T_{i_t} of candidate solutions for $K[0 \ldots i_t]$ passing the *left critical filter* i_t. In the same manner, we start with S_N and j_N

and run the KSA backwards until we reach a stage where we have a set $T_{i'_t}$ of candidate solutions for $K[i'_{t'}+1 \ldots l-1]$ passing the *right critical filter* $i'_{t'}$.

Among the set of remaining key bytes $L = \{K[i_t+1], \ldots, K[i_{t+1}]\} \setminus \mathcal{P}$ on the left and the set of remaining key bytes $R = \{K[i'_{t'+1}+1], \ldots, K[i'_{t'}]\} \setminus \mathcal{P}$ on the right, some bytes are common. We first guess the smaller of these two sets and then guess only those key bytes from the larger set that are needed to complete the whole key. We can reduce the candidate keys by filtering the left half of the key using the right filters and the right half of the key using the left filters.

In the *BidirKeySearch* algorithm description, i_0 and i'_0 denote the two default filters -1 (a dummy filter) and $N-1$ respectively.

BidirKeySearch (S_N, j_N, $\mathcal{P}$, $\mathcal{S}$, $\mathcal{B}$)

Guess the Left Half-Key:

1. For $u = 1$ to t do
   - 1.1. Iteratively guess $\{K[i_{u-1}+1], \ldots, K[i_u]\} \setminus \mathcal{P}$.
   - 1.2. $T_{i_u} \leftarrow \big\{\{K[0], \ldots, K[i_u]\}$: the tuple pass the filter $i_u\big\}$.

Guess the Right Half-Key:

2. For $v = 1$ to t' do
   - 2.1. Iteratively guess $\{K[i'_v+1], \ldots, K[i'_{v-1}]\} \setminus \mathcal{P}$.
   - 2.2. $T_{i_v} \leftarrow \big\{\{K[i'_v+1], \ldots, K[l-1]\}$: the tuple pass the filter $i_v\big\}$.

Merge to Get the Full Key:

3. $L \leftarrow \{K[i_t+1], \ldots, K[i_{t+1}]\} \setminus \mathcal{P}$ and $R \leftarrow \{K[i'_{t'+1}+1], \ldots, K[i'_{t'}]\} \setminus \mathcal{P}$.
4. If $|L| < |R|$ then do
   - 4.1. Guess L.
   - 4.2. $T_{i_{t+1}} \leftarrow \big\{\{K[0], \ldots, K[i_{t+1}]\}$: the tuple pass the filter $i_{t+1}\big\}$.
   - 4.3. Guess $R \setminus T_{i_{t+1}}$ using the filter $i'_{t'+1}$.
5. Else
   - 5.1. Guess R.
   - 5.2. $T_{i'_{t'+1}} \leftarrow \big\{\{K[i'_{t'}+1], \ldots, K[l-1]\}$: the tuple pass the filter $i_v\big\}$.
   - 5.3. Guess $L \setminus T_{i'_{t'+1}}$ using the filter i_{t+1}.

Cross-Filtration:

6. If $K[0 \ldots m-1]$ is guessed from the left filters and $K[m \ldots l-1]$ is guessed from the right filters, then validate $K[m \ldots l-1]$ using the left filters up to $l-1$ and validate $K[0 \ldots m-1]$ using the right filters up to $N-1-l$.

Let m_1 be the complexity and α be the probability of obtaining a d-favourable (t, t') bisequence along with the partially recovered key set $\mathcal{P}$, satisfying the input requirements of *BidirKeySearch*. We require a search complexity of $m_2 = \binom{2(M-l)}{n_{corr}-n_{in}}$ for locating the $n_{corr} - n_{in}$ many correct key bytes from the filters in $[l, l+M-1] \cup [N-1-l, N-2-M]$. We need to run the *BidirKeySearch* algorithm a total of $m_3 = \sum_{r=0}^{b} \binom{n-n_{corr}}{r} 3^r$ times and this gives a success probability $\beta = \sum_{r=n-n_{corr}-b}^{n-n_{corr}} \binom{n-n_{corr}}{r} g^r (1-g)^{n-n_{corr}-r}$. If each run of the *BidirKeySearch* algorithm consumes τ time, the overall complexity for the full key recovery is $m_1 m_2 m_3 \tau 2^8$ and the success probability is $\alpha\beta$. The term 2^8 comes for guessing the correct value of j_N.

For $l = 16$, $d = 4$, $M = 20$, $n_{corr} = 6$, $n_{in} = 4$, our experiments with 10 million randomly generated secret keys reveal that $\alpha \approx 0.1537$ and $g \approx 0.8928$. Also, $m_1 \approx 2^{31}$ (see Table 2) and $m_2 \approx 2^5$. With $b = 2$, β and m_3 turn out to be 0.9169 and 2^9 (approx.) respectively. Thus, the success probability is $\alpha\beta \approx 0.1409$ and the complexity is approximately $2^{31+5+9+8}\tau = 2^{53}\tau$. Though the the exact estimation of τ is not feasible at this point, τ is expected to be small, since for wrong filter sequences the search is likely to terminate in a negligible amount of time. So far, the best known success probability for recovering a 16 bytes secret key has been 0.0745 [1]. The time reported in [1] for this probability is 1572 seconds, but it is not clear how the complexity of [1] would grow with increase in probability. We present better probability (almost two times that of [1]), but with very high time complexity, which is infeasible in practice unless further improvement is made.

4 Conclusion

In this paper, we have performed a detailed theoretical study on sequences of j indices in RC4 KSA. We have also demonstrated an application of our sequence analysis in secret key recovery from the final permutation after the KSA. Though our key retrieval algorithm has good success probability, it has high time complexity. Currently, we are working on some more special sequences to reduce the complexity parts from those reported in Table 2 and thereby improve the overall time complexity for key recovery.

References

1. Akgün, M., Kavak, P., Demirci, H.: New Results on the Key Scheduling Algorithm of RC4. In: Chowdhury, D.R., Rijmen, V. (eds.) INDOCRYPT 2008. LNCS, vol. 5365, pp. 40–52. Springer, Heidelberg (2008)
2. Biham, E., Carmeli, Y.: Efficient Reconstruction of RC4 Keys from Internal States. In: Nyberg, K. (ed.) FSE 2008. LNCS, vol. 5086, pp. 270–288. Springer, Heidelberg (2008)
3. Khazaei, S., Meier, W.: On Reconstruction of RC4 Keys from Internal States. In: Calmet, J., Geiselmann, W., Müller-Quade, J. (eds.) Mathematical Methods in Computer Science (MMICS). LNCS, vol. 5393, pp. 179–189. Springer, Heidelberg (2008)
4. Knudsen, L.R., Meier, W., Preneel, B., Rijmen, V., Verdoolaege, S.: Analysis Methods for (Alleged) RC4. In: Ohta, K., Pei, D. (eds.) ASIACRYPT 1998. LNCS, vol. 1514, pp. 327–341. Springer, Heidelberg (1998)
5. LAN/MAN Standard Committee. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, 1999 edition. IEEE standard 802.11 (1999)
6. Maitra, S., Paul, G.: New Form of Permutation Bias and Secret Key Leakage in Keystream Bytes of RC4. In: Nyberg, K. (ed.) FSE 2008. LNCS, vol. 5086, pp. 253–269. Springer, Heidelberg (2008); A revised and extended version with the same title is available at the IACR Eprint Server, eprint.iacr.org, number 2007/261 (January 9, 2009)

7. Mantin, I.: Analysis of the stream cipher RC4. Master's Thesis, The Weizmann Institute of Science, Israel (2001)
8. Maximov, A., Khovratovich, D.: New State Recovering Attack on RC4. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 297–316. Springer, Heidelberg (2008)
9. McKague, M.E.: Design and Analysis of RC4-like Stream Ciphers. Master's Thesis, University of Waterloo, Canada (2005)
10. Paul, G., Maitra, S.: Permutation after RC4 Key Scheduling Reveals the Secret Key. In: Adams, C., Miri, A., Wiener, M. (eds.) SAC 2007. LNCS, vol. 4876, pp. 360–377. Springer, Heidelberg (2007)
11. Paul, G., Maitra, S.: RC4 State Information at Any Stage Reveals the Secret Key. IACR Eprint Server, eprint.iacr.org, number 2007/2008 (January 9, 2009); This is an extended version of [10]
12. Roos, A.: A class of weak keys in the RC4 stream cipher. Two posts in sci.crypt, message-id 43u1eh$1j3@hermes.is.co.za and 44ebge$llf@hermes.is.co.za (1995)
13. Tews, E.: Attacks on the WEP protocol. IACR Eprint Server, eprint.iacr.org, number 2007/471, December 15 (2007)
14. Tomasevic, V., Bojanic, S., Nieto-Taladriz, O.: Finding an internal state of RC4 stream cipher. Information Sciences 177, 1715–1727 (2007)

Very-Efficient Anonymous Password-Authenticated Key Exchange and Its Extensions

SeongHan Shin[1,2], Kazukuni Kobara[1,2], and Hideki Imai[2,1]

[1] Research Center for Information Security (RCIS),
National Institute of Advanced Industrial Science and Technology (AIST),
1-18-13, Sotokanda, Chiyoda-ku, Tokyo, 101-0021 Japan
seonghan.shin@aist.go.jp
[2] Chuo University,
1-13-27, Kasuga, Bunkyo-ku, Tokyo, 112-8551 Japan

Abstract. An anonymous password-authenticated key exchange (anonymous PAKE) protocol is designed to provide both password-only authentication and user anonymity. In this paper, we propose a *very-efficient* anonymous PAKE (called, VEAP) protocol that provides the most efficiency among their kinds in terms of computation and communication costs. The VEAP protocol guarantees semantic security of session keys in the random oracle model under the chosen target CDH problem, and unconditional user anonymity against a semi-honest server. If the pre-computation is allowed, the computation cost of the VEAP protocol is the same as the well-known Diffie-Hellman protocol! In addition, we extend the VEAP protocol in two ways.

1 Introduction

Since the Diffie-Hellman protocol [8], many researchers have tried to design secure cryptographic algorithms/protocols for realizing secure channels. These algorithms/protocols are necessary because application-oriented protocols (e.g., web-mail, Internet banking/shopping, ftp) are frequently developed assuming the existence of such secure channels. The latter can be achieved by an authenticated key exchange (AKE) protocol at the end of which the involving parties authenticate and share a common session key to be used for subsequent cryptographic algorithms. For authentication, human-memorable passwords are commonly used rather than high-entropy keys because of their convenience. Many password-based AKE protocols (see [10]) have been extensively investigated so far. However, one should be very careful about two major attacks on passwords: on-line and off-line dictionary attacks. While on-line attacks are applicable to all of the password-based protocols equally, they can be prevented by having a server take appropriate countermeasures (e.g., lock-up accounts after consecutive failures of passwords). But, we cannot avoid off-line attacks by such countermeasures mainly because these attacks can be done off-line and independently of the party.

M. Bras-Amorós and T. Høholdt (Eds.): AAECC 2009, LNCS 5527, pp. 149–158, 2009.

1.1 Anonymous Password-Authenticated Key Exchange

A password-authenticated key exchange (PAKE) protocol allows a user, who remembers his/her password only, to authenticate with the counterpart server that holds the password or its verification data. In PAKE protocols (see [9]), a user should send his/her identity clearly in order to share an authenticated session key. Let us suppose an adversary who fully controls the networks. Though the adversary cannot impersonate any party in PAKE protocols, it is easy to collect a user's personal information about the communication history itself. For this problem, Viet et al., [15] proposed an anonymous PAKE protocol and its threshold construction[1] both of which simply combine a PAKE protocol [1] with an oblivious transfer (OT) protocol [7,14] for user's anonymity. The user anonymity is guaranteed against an outside adversary as well as a passive server, who follows the protocol honestly. Later, Shin et al., [13] proposed an anonymous PAKE protocol that is only based on the PAKE protocol [1], and showed that their protocol significantly improved the efficiency compared to [15]. Very recently, Yang and Zhang [16] proposed a new anonymous PAKE protocol that is based on a different PAKE (i.e., SPEKE [11,12]) protocol. The main idea of [16] is that the user and the server share a Difffie-Hellman-like key, by using the SPEKE protocol [11,12], and then run a sequential Diffie-Hellman protocol partially-masked with the shared key. To our best knowledge, all the anonymous PAKE protocols are constructed from PAKE protocols and, among them, the construction of [16] seems the most efficient in terms of computation and communication costs.

1.2 Our Contributions

In this paper, we propose a very-efficient anonymous PAKE (called, VEAP) protocol whose core part is based on the blind signature scheme [5]. The VEAP protocol guarantees semantic security of session keys in the random oracle model under the chosen target CDH problem, and unconditional user anonymity against a semi-honest server. In the VEAP protocol, the user and the server are required to compute only one modular exponentiation, respectively, if the pre-computation is allowed. Surprisingly, this is the same computation cost of the well-known Diffie-Hellman protocol [8] which does not provide authentication at all. Also, we extend the VEAP protocol in two ways: the first is designed to reduce the communication cost of the VEAP protocol and the second shows that stripping off anonymity parts from the VEAP protocol results in a *new* PAKE protocol.

Organization. In the next section, we explain some preliminaries to be used throughout this paper. In Section 3, we propose a very-efficient anonymous PAKE (called, VEAP) protocol with security proof and efficiency comparison. In Section 4, we extend the VEAP protocol in two ways.

[1] In their threshold construction, the "threshold" number of users should collaborate one another in order to be authenticated by the server. In this paper, we only focus on an anonymous PAKE protocol where the threshold $t = 1$.

2 Preliminary

2.1 Notation

Here, we explain some notation. Let $\mathbb{G}$ be a finite, cyclic group of prime order p and g be a generator of $\mathbb{G}$. The parameter $(\mathbb{G}, p, g)$ is public to everyone. In the aftermath, all the subsequent arithmetic operations are performed in modulo p unless otherwise stated. Let l and κ be the security parameters for hash functions (i.e., the size of the hashed value) and a symmetric-key encryption scheme (i.e., the length of the key), respectively. Let $\{0,1\}^*$ be the set of finite binary strings and $\{0,1\}^l$ be the set of binary strings of length l. Let "$\|$" be the concatenation of bit strings in $\{0,1\}^\star$. Let us denote $\mathbb{G}^* = \mathbb{G}\backslash\{1\}$ where 1 is the identity element of $\mathbb{G}$. The $\mathcal{G}$ is a full-domain hash (FDH) function that maps $\{0,1\}^\star$ to the elements of $\mathbb{G}^*$. While $\mathcal{F} : \{0,1\}^\star \rightarrow \{0,1\}^\kappa$, the other hash functions are denoted $\mathcal{H}_k : \{0,1\}^\star \rightarrow \{0,1\}^{l_k}$, for $k = 1, 2, 3$. Let $U = \{U_1, \cdots, U_n\}$ and S be the identities of a group of n users and server, respectively.

2.2 Formal Model

In this subsection, we introduce an extended model, built upon [4,2], and security notions for anonymous PAKE.

Model. In an anonymous PAKE protocol P, there are two parties U_i ($\in U$) and S where a pair of user U_i and server S share a low-entropy password pw_i, chosen from a small dictionary $\mathbb{D}_{\mathsf{password}}$, for i ($1 \leq i \leq n$). Each of U_i and S may have several instances, called oracles involved in distinct, possibly concurrent, executions of P. We denote U_i (resp., S) instances by U_i^μ (resp., S^ν) where $\mu, \nu \in \mathbb{N}$, or by I in case of any instance. During the protocol execution, an adversary $\mathcal{A}$ has the entire control of networks whose capability can be represented as follows:

- $\mathsf{Execute}(U_i^\mu, S^\nu)$: This query models passive attacks, where the adversary gets access to honest executions of P between U_i^μ and S^ν.
- $\mathsf{Send}(I, m)$: This query models active attacks by having $\mathcal{A}$ send a message m to an instance I. The adversary $\mathcal{A}$ gets back the response I generates in processing m according to the protocol P.
- $\mathsf{Reveal}(I)$: This query handles misuse of the session key by any instance I. The query is only available to $\mathcal{A}$, if the instance actually holds a session key, and at that case the key is released to $\mathcal{A}$.
- $\mathsf{RegisterUser}(U_j, pw_j)$: This query handles inside attacks by having $\mathcal{A}$ register a user U_j to server S with a password pw_j (i.e., $U_j \in U$).
- $\mathsf{Test}(I)$: This query is used to measure how much the adversary can obtain information about the session key. The Test-query can be asked at most once by the adversary $\mathcal{A}$ and is only available to $\mathcal{A}$ if the instance I is *fresh*[2].

[2] We consider an instance I that has accepted (holding a session key SK). Let I' be a partnered instance of I (refer to [4]). The instance I is fresh unless the following conditions are satisfied: (1) either $\mathsf{Reveal}(I)$ or $\mathsf{Reveal}(I')$ is asked by $\mathcal{A}$ at some point; or (2) $\mathsf{RegisterUser}(I, \star)$ is asked by $\mathcal{A}$ when $I = U_i$ for any i.

This query is answered as follows: one flips a private coin $b \in \{0,1\}$, and forwards the corresponding session key SK ($\mathsf{Reveal}(I)$ would output), if $b = 1$, or a random value with the same size except the session key, if $b = 0$.

Security Notions. The adversary $\mathcal{A}$ is provided with random coin tosses, some oracles and then is allowed to invoke any number of queries as described above, in any order. The AKE security is defined by the game $\mathbf{Game}^{\mathsf{ake}}(\mathcal{A}, P)$ where the goal of the adversary is to guess the bit b, involved in the Test-query, by outputting this guess b'. We denote the AKE advantage, by $\mathsf{Adv}_P^{\mathsf{ake}}(\mathcal{A}) = 2\Pr[b = b'] - 1$, as the probability that $\mathcal{A}$ can correctly guess the value of b. The protocol P is said to be (t, ε)-AKE-secure if $\mathcal{A}$'s advantage is smaller than ε for any adversary $\mathcal{A}$ running time t.

As [15], we consider a semi-honest server S, who honestly follows the protocol P, but it is curious about the involved user's identity. The user anonymity is defined by the probability distribution of messages in P. Let $P(U_i, S)$ (resp., $P(U_j, S)$) be the transcript of P between U_i (resp., U_j) and S. We can say that the protocol P is *anonymous* if, for any two users $U_i, U_j \in U$, $\mathsf{Dist}[P(U_i, S)] = \mathsf{Dist}[P(U_j, S)]$ where $\mathsf{Dist}[c]$ denotes c's probability distribution. This security notion means that the server S gets no information about the user's identity.

2.3 Computational Assumptions

Chosen Target CDH (CT-CDH) Problem [3,5]. Let $\mathbb{G} = \langle g \rangle$ be a finite cyclic group of prime order p with g as a generator. Let x be a random element of $\mathbb{Z}_p^\star$ and let $X \equiv g^x$. A (t, ε)-$\mathsf{CT\text{-}CDH}_{g,\mathbb{G}}$ attacker is a probabilistic polynomial time (PPT) machine $\mathcal{B}$ that is given X and has access to the target oracle $\mathcal{T}_{\mathbb{G}}$, returning random points W_i in $\mathbb{G}^*$, as well as the helper oracle $(\cdot)^x$. Let q_T (resp., q_H) be the number of queries $\mathcal{B}$ made to the target (resp., helper) oracles. The success probability $\mathsf{Succ}_{g,\mathbb{G}}^{\mathsf{ct-cdh}}(\mathcal{B})$ of $\mathcal{B}$ attacking the chosen-target CDH problem is defined as the probability to output a set of m pairs $((j_1, K_1), \cdots, (j_m, K_m))$ where, for all i $(1 \le i \le m)$, $\exists j_i$ $(1 \le j_i \le q_T)$ such that $K_i \equiv W_{j_i}^x$, all K_i are distinct and $q_H < q_T$. We denote by $\mathsf{Succ}_{g,\mathbb{G}}^{\mathsf{ct-cdh}}(t)$ the maximal success probability over every adversaries, running within time t. The $\mathsf{CT\text{-}CDH}$-Assumption states that $\mathsf{Succ}_{g,\mathbb{G}}^{\mathsf{ct-cdh}}(t) \le \varepsilon$ for any t/ε not too large. Note that, if $\mathcal{B}$ makes one query to the target oracle, then the chosen-target CDH assumption is equivalent to the computational Diffie-Hellman assumption.

A Symmetric-Key Encryption Scheme. A symmetric-key encryption scheme $\mathcal{SE}$ consists of the following two algorithms $(\mathcal{E}, \mathcal{D})$, with a symmetric-key $\mathcal{K}$ uniformly distributed in $\{0,1\}^\kappa$: (1) Given a message M and a key $\mathcal{K}$, $\mathcal{E}$ produces a ciphertext $C = \mathcal{E}_{\mathcal{K}}(M)$; and (2) Given a ciphertext C and a key $\mathcal{K}$, $\mathcal{D}$ recovers a message $M = \mathcal{D}_{\mathcal{K}}(C)$. The semantic security for $\mathcal{SE}$ is that it is infeasible for an adversary to distinguish the encryptions of two messages M_0 and M_1 (of the same length), even though the adversary has access to the en/decryption oracles. We denote by $\mathsf{Succ}_{\mathcal{SE}}^{\mathsf{ss}}(t, q)$ the maximal success probability of a distinguisher, running within time t and making at most q queries to the $\mathcal{E}$ and $\mathcal{D}$ oracles.

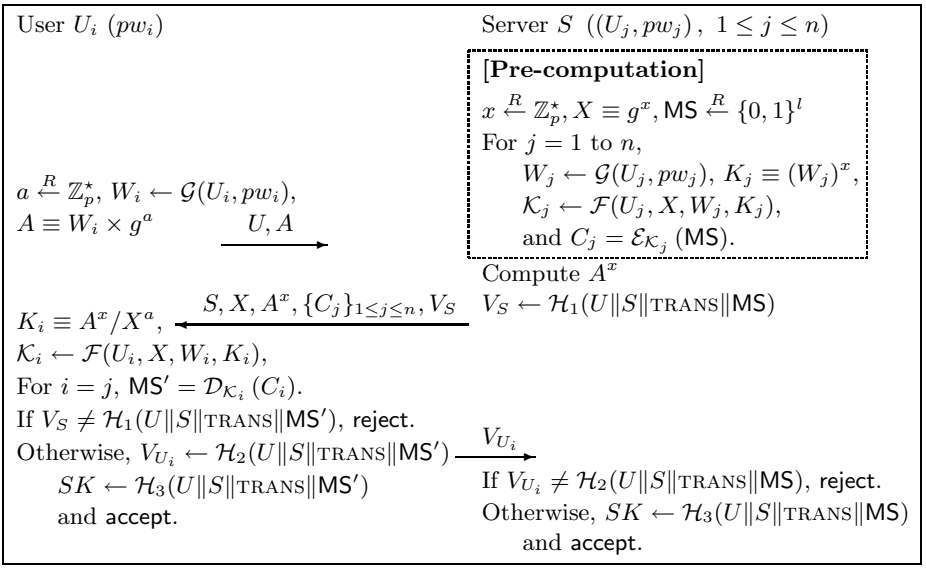


Fig. 1. The VEAP protocol where $\text{TRANS} = A\|A^x\|X\|\{C_j\}_{1\le j\le n}$

3 A Very-Efficient Anonymous PAKE Protocol

In this section, we propose a very-efficient anonymous PAKE (for short, VEAP) protocol that provides the most efficiency among their kinds in terms of computation and communication costs. We also show that the VEAP protocol guarantees not only AKE security against an active adversary but also unconditional user anonymity against a semi-honest server, who honestly follows the protocol.

3.1 The VEAP Protocol

We assume that each user U_i of the group $U = \{U_1, \cdots, U_n\}$ has registered his/her password pw_i to server S. For simplicity, we assign the users consecutive integer i $(1 \le i \le n)$ so that U_i can be regarded as the i-th user of U. In the VEAP protocol, any user U_i can share an authenticated session key with server S anonymously (see Fig. 1).

Step 0 (Pre-computation of server S): At first, server S chooses a random number x from $\mathbb{Z}_p^\star$ and a random master secret MS from $\{0,1\}^l$, and computes its Diffie-Hellman public value $X \equiv g^x$. Then, the server repeats the followings for all users U_j $(1 \le j \le n)$: 1) it computes $W_j \leftarrow \mathcal{G}(U_j, pw_j)$; 2) it computes $K_j \equiv (W_j)^x$ and derives a symmetric-key $\mathcal{K}_j \leftarrow \mathcal{F}(U_j, X, W_j, K_j)$; and 3) it generates a ciphertext $C_j = \mathcal{E}_{\mathcal{K}_j}(\mathsf{MS})$ for the master secret MS.

Step 1: The user U_i chooses a random number a from $\mathbb{Z}_p^\star$ and computes $W_i \leftarrow \mathcal{G}(U_i, pw_i)$. The W_i is masked with g^a so that the resultant value

A is computed in a way of $A \equiv W_i \times g^a$. The user sends A and the group U of users' identities to server S.

Step 2: For the received value A, server S computes A^x with the exponent x. Also, the server generates an authenticator $V_S \leftarrow \mathcal{H}_1(U\|S\|\text{TRANS}\|\mathsf{MS})$. Then, server S sends its identity S, X, A^x, $\{C_j\}_{1\le j\le n}$ and V_S to user U_i.

Step 3: After receiving the message from server S, user U_i first computes $K_i \equiv A^x/X^a$ and $\mathcal{K}_i \leftarrow \mathcal{F}(U_i, X, W_i, K_i)$. With $\mathcal{K}_i$, the user decrypts the i-th ciphertext $C_{i=j}$ as follows: $\mathsf{MS}' = \mathcal{D}_{\mathcal{K}_i}(C_i)$. If the received V_S is not valid, user U_i terminates the protocol. Otherwise, the user generates an authenticator $V_{U_i} \leftarrow \mathcal{H}_2(U\|S\|\text{TRANS}\|\mathsf{MS}')$ and a session key $SK \leftarrow \mathcal{H}_3(U\|S\|\text{TRANS}\|\mathsf{MS}')$. The authenticator V_{U_i} is sent to server S.

Step 4: If the received V_{U_i} is not valid, server S terminates the protocol. Otherwise, the server generates a session key $SK \leftarrow \mathcal{H}_3(U\|S\|\text{TRANS}\|\mathsf{MS})$.

RATIONALE. Unlike [13,15,16], the novelty of the VEAP protocol is that it does not use the existing PAKE protocols [9,10] at all. In order to provide the most efficiency, the VEAP protocol has the following rationale: 1) Instead of unmasking A, server S sends X and A^x which can be used to derive $K_{i=j}$ if user U_i has computed W_i with the correct U_i and pw_i. Importantly, this procedure makes it possible for server S to pre-compute K_j for all users U_j $(1 \le j \le n)$; 2) The server S generates only one Diffie-Hellman public value X and its exponent is used to compute all of the K_j's; 3) The server S sends $\{C_j\}_{1\le j\le n}$ by encrypting the master secret MS with $\mathcal{K}_j$. This is enough to guarantee unconditional user anonymity against a semi-honest server; and 4) Both U_i and pw_i are used to compute the verification data W_i. This is crucial since an adversary is enforced to make an on-line dictionary attack on a specific user, not the others.

3.2 Security

In this subsection, we prove that the VEAP protocol is AKE-secure under the chosen target CDH problem in the random oracle model [6] and provides unconditional user anonymity against a semi-honest server.

Theorem 1 (AKE Security). *Let P be the* VEAP *protocol where passwords are independently chosen from a dictionary of size N. For any adversary $\mathcal{A}$ within a polynomial time t, with less than q_{send} active interactions with the parties (*Send*-queries), q_{execute} passive eavesdroppings (*Execute*-queries) and asking q_{hashF}, q_{hashG}, q_{hashH} hash queries to $\mathcal{F}$, $\mathcal{G}$, any $\mathcal{H}_k$, for $k = 1, 2, 3$, respectively,*

$$\mathsf{Adv}_P^{\mathsf{ake}}(\mathcal{A}) \le 2nq_{\mathsf{sendS}} \times \mathsf{Succ}_{\mathcal{SE}}^{\mathsf{ss}}(t_1, q_1) + 16q_{\mathsf{sendS}} \times \mathsf{Succ}_{g,\mathbb{G}}^{\mathsf{ct-cdh}}(t_2 + q_{\mathsf{hashG}} \cdot \tau_e)$$

$$+\frac{6(q_{\mathsf{sendU}} + q_{\mathsf{sendS}})}{N} + \frac{2Q^2}{|\mathbb{G}|} + \frac{q_{\mathsf{hashF}}^2}{2^\kappa} + \frac{q_{\mathsf{hashH}}^2}{2^l} + \frac{2q_{\mathsf{sendU}}}{2^{l_1}} + \frac{2q_{\mathsf{sendS}}}{2^{l_2}} \tag{1}$$

where (1) q_{sendU} (resp., q_{sendS}) is the number of Send*-queries to U_i (resp., S) instance, (2) $Q = q_{\mathsf{execute}} + q_{\mathsf{send}}$ and $q_{\mathsf{send}} \le q_{\mathsf{sendU}} + q_{\mathsf{sendS}}$, (3) l_1 and l_2 are the output sizes of hash function $\mathcal{H}_1$ and $\mathcal{H}_2$, respectively, and (4) τ_e denotes the computational time for an exponentiation in $\mathbb{G}$.*

The main proof strategy is as follows. In order to "embed" X of the CT-CDH problem, the simulator selects a random value s in the set $\{1, \cdots, q_{\mathsf{sendS}}\}$ where q_{sendS} is the number of Send-queries to S instance. If this is the s-th instance of party S, the simulator uses the X and forwards A to the helper oracle $(\cdot)^x$, and then returns the received A^x. Let q_{T_1} and q_{T_2} be the first and the second $\mathcal{G}$-query, respectively. The simulator forwards q_{T_1} and q_{T_2} to the target oracle $\mathcal{T}_{\mathbb{G}}$ and returns the received W_{T_1} and W_{T_2}, respectively. For u ($3 \leq u \leq \mathsf{hashG}$) in the $\{q_{T_3}, \cdots, q_{T_{\mathsf{hashG}}}\}$ where hashG is the number of $\mathcal{G}$-queries, the simulator chooses a random number w_u from $\mathbb{Z}_p^\star$ and returns $W_{T_u} \equiv W_{T_1}^{w_u}$, if $u \equiv 1 \bmod 2$, or $W_{T_u} \equiv W_{T_2}^{w_u}$, if $u \equiv 0 \bmod 2$. That is, we simulate $\mathcal{G}$ oracle by answering with a randomized W_{T_1}, for half of $\mathcal{G}$-queries, and W_{T_2}, for the remaining queries. Note that $q_T = 2$ and $q_H = 1$. Whenever there is a collision on W_{j_i}, the simulator can find the solution to the CT-CDH problem with the probability $1/2q_{\mathsf{sendS}}$.

Theorem 2 (Anonymity). *The* VEAP *protocol provides unconditional user anonymity against a semi-honest server.*

Proof. Consider server S who honestly follows the VEAP protocol. It is obvious that server S cannot get any information about the user's identity since the A has a unique discrete logarithm of g and, with the randomly-chosen number a, it is the uniform distribution over $\mathbb{G}^*$. This also implies that the server cannot distinguish A_i (of user U_i) from A_j (of $U_{j\neq i}$) because they are completely independent each other. In addition, even if server S receives the authenticator V_{U_i} the A does not reveal any information about the user's identity since the probability, for all users, to get MS is equal. Therefore, $\mathsf{Dist}[P(U_i, S)] = \mathsf{Dist}[P(U_j, S)]$ for any two users $\{U_i, U_j\} \in U$. □

Remark 1. In [16], they discussed user anonymity against a malicious server who does not follow the protocol. However, the NAPAKE protocol [16] does not provide user anonymity against such type of attacks because, if there are only two users in the group, the malicious server can always determine which user is the one, with probability 1/2, by using different random values for the users.

3.3 Efficiency Comparison

This subsection shows efficiency comparison of the VEAP protocol and the previous anonymous PAKE protocols [15,13,16] (see Table 1). With respect to computation costs, we count the number of modular exponentiations of user U_i and server S. In Table 1, "Total" means the total number of modular exponentiations and "T−P" is the remaining number of modular exponentiations after excluding those that are pre-computable. With respect to communication costs, we measure the bit-length of messages where $|\cdot|$ indicate the bit-length.

As for computation cost of the VEAP protocol, user U_i (resp., server S) is required to compute 2 (resp., $n + 2$) modular exponentiations. When pre-computation is allowed, the remaining costs of user U_i (resp., server S) are only 1 (resp., 1) modular exponentiation. Note that this is the same computation cost

Table 1. Efficiency comparison of anonymous PAKE protocols in terms of computation and communication costs where n is the number of users

Protocols	Number of modular exponentiations				Communication costs *1
	User U_i		Server S		
	Total	T−P	Total	T−P	
APAKE [15]	6	4	$4n+2$	$3n+1$	$(n+2)\|p\|+(n+1)\|\mathcal{H}\|$
TAP [13]	3	2	$n+1$	n	$2\|p\|+(n+1)\|\mathcal{H}\|$
NAPAKE [16]	4	3 *2	$n+3$	2	$(n+3)\|p\|+\|\mathcal{H}\|$ *2
VEAP	2	1	$n+2$	1	$3\|p\|+2\|\mathcal{H}\|+n\|\mathcal{E}\|$

*1: The bit-length of identities is excluded

*2: In [16], they incorrectly estimated the efficiency of the NAPAKE protocol

of the Diffie-Hellman protocol [8]. One can easily see that the VEAP protocol is the most efficient in the number of modular exponentiations for both user and server. As for communication cost, the VEAP protocol requires a bandwidth of $(3|p|+2|\mathcal{H}|+n|\mathcal{E}|)$-bits, except the length of identities U and S, where the bandwidth for $|\mathcal{H}|$ and the modulus size $|p|$ are independent from the number of users. If we consider the minimum security parameters ($|p|=1024$, $|\mathcal{H}|=160$ and $|\mathcal{E}|=128$), the gap of communication costs between the VEAP protocol and the others [15,13,16] becomes larger as the number of users increases.

4 Its Extensions

4.1 Reducing Communication Cost of the VEAP Protocol

In this subsection, we show how to reduce the communication cost of the VEAP protocol, such that it is independent of the number of users, at the expense of slight increase of the computation cost (see Fig. 2). The main difference from the VEAP protocol is that server S fixes the values X and $\{C_j\}_{1\leq j\leq n}$ for a time period t. This obviously reduces the communication cost to be independent of the number of users. Instead, user U_i has to read a necessary information from server S's public bulletin board. Note that, if an adversary changes these values, this extended protocol results in "failure". However, one should be careful about the following problems: 1) session key privacy against other legitimate users; and 2) forward secrecy. In fact, the values $B^x \equiv X^b$ and $B^y \equiv Y^b$ in the hash functions play an important role to solve the above problems in the extended protocol.

4.2 A New PAKE Protocol

Here, we show that stripping off anonymity parts from the VEAP protocol is a new kind of PAKE protocol (see Fig. 3). To our best knowledge, this PAKE protocol is the first construction built from the blind signature scheme [5].

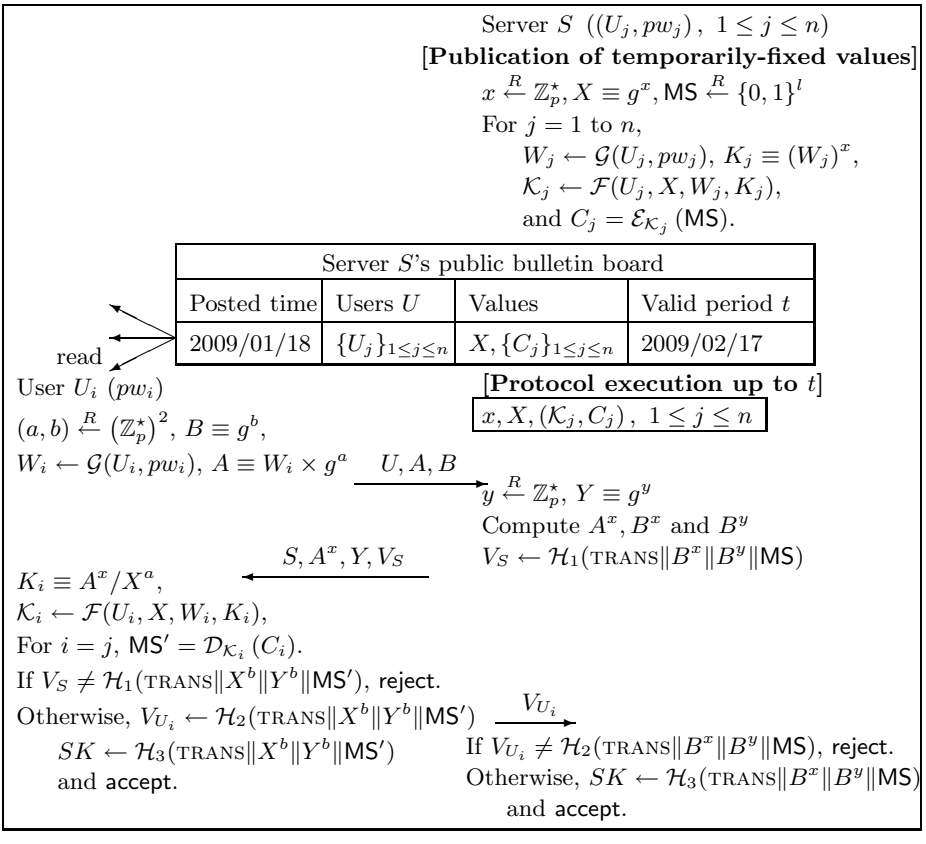


Fig. 2. An extended VEAP protocol where $\text{TRANS} = U\|S\|A\|A^x\|X\|B\|Y\|\{C_j\}_{1 \le j \le n}$

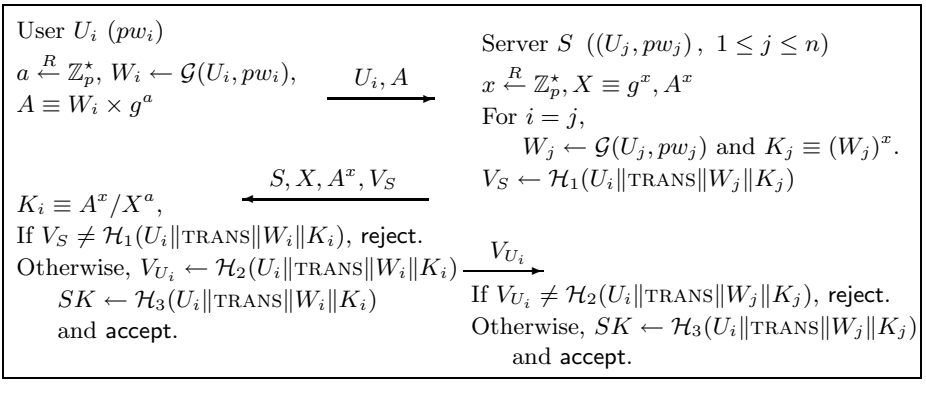


Fig. 3. A new PAKE protocol from the VEAP protocol where $\text{TRANS} = S\|A\|A^x\|X$

Acknowledgements

We appreciate the helpful comments of anonymous reviewers.

References

1. Abdalla, M., Pointcheval, D.: Simple Password-Based Encrypted Key Exchange Protocols. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 191–208. Springer, Heidelberg (2005)
2. Bresson, E., Chevassut, O., Pointcheval, D.: New Security Results on Encrypted Key Exchange. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 145–158. Springer, Heidelberg (2004)
3. Bellare, M., Namprempre, C., Pointcheval, D., Semanko, M.: The One-More-RSA-Inversion Problems and the Security of Chaum's Blind Signature Scheme. Journal of Cryptology 16(3), 185–215 (2003)
4. Bellare, M., Pointcheval, D., Rogaway, P.: Authenticated Key Exchange Secure against Dictionary Attacks. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 139–155. Springer, Heidelberg (2000)
5. Boldyreva, A.: Threshold Signatures, Multisignatures and Blind Signatures Based on the Gap-Diffie-Hellman-Group Signature Scheme. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 31–46. Springer, Heidelberg (2002)
6. Bellare, M., Rogaway, P.: Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In: Proc. of ACM CCS 1993, pp. 62–73. ACM Press, New York (1993)
7. Chu, C.K., Tzeng, W.G.: Efficient k-Out-of-n Oblivious Transfer Schemes with Adaptive and Non-adaptive Queries. In: Vaudenay, S. (ed.) PKC 2005. LNCS, vol. 3386, pp. 172–183. Springer, Heidelberg (2005)
8. Diffie, W., Hellman, M.: New Directions in Cryptography. IEEE Transactions on Information Theory IT-22(6), 644–654 (1976)
9. IEEE P1363.2: Password-Based Public-Key Cryptography, `http://grouper.ieee.org/groups/1363/passwdPK/submissions.html`
10. `http://www.jablon.org/passwordlinks.html`
11. Jablon, D.: Strong Password-Only Authenticated Key Exchange. ACM Computer Communication Review 26(5), 5–20 (1996)
12. Jablon, D.: Extended Password Key Exchange Protocols Immune to Dictionary Attacks. In: WET-ICE 1997 Workshop on Enterprise Security (1997)
13. Shin, S.H., Kobara, K., Imai, H.: A Secure Threshold Anonymous Password-Authenticated Key Exchange Protocol. In: Miyaji, A., Kikuchi, H., Rannenberg, K. (eds.) IWSEC 2007. LNCS, vol. 4752, pp. 444–458. Springer, Heidelberg (2007)
14. Tzeng, W.G.: Efficient 1-Out-n Oblivious Transfer Schemes. In: Naccache, D., Paillier, P. (eds.) PKC 2002. LNCS, vol. 2274, pp. 159–171. Springer, Heidelberg (2002)
15. Viet, D.Q., Yamamura, A., Tanaka, H.: Anonymous Password-Based Authenticated Key Exchange. In: Maitra, S., Veni Madhavan, C.E., Venkatesan, R. (eds.) INDOCRYPT 2005. LNCS, vol. 3797, pp. 244–257. Springer, Heidelberg (2005)
16. Yang, J., Zhang, Z.: A New Anonymous Password-Based Authenticated Key Exchange Protocol. In: Chowdhury, D.R., Rijmen, V., Das, A. (eds.) INDOCRYPT 2008. LNCS, vol. 5365, pp. 200–212. Springer, Heidelberg (2008)

Efficient Constructions of Deterministic Encryption from Hybrid Encryption and Code-Based PKE

Yang Cui[1,2], Kirill Morozov[1], Kazukuni Kobara[1,2], and Hideki Imai[1,2]

[1] Research Center for Information Security (RCIS),
National Institute of Advanced Industrial Science & Technology (AIST), Japan
{y-cui,kirill.morozov,k-kobara,h-imai}@aist.go.jp
http://www.rcis.aist.go.jp/
[2] Chuo University, Japan

Abstract. We build on the new security notion for deterministic encryption (PRIV) and the PRIV-secure schemes presented by Bellare et al at Crypto'07. Our work introduces: 1) A generic and efficient construction of deterministic length-preserving hybrid encryption, which is an improvement on the scheme sketched in the above paper; to our best knowledge, this is the first example of length-preserving hybrid encryption; 2) postquantum deterministic encryption (using the IND-CPA variant of code-based McEliece PKE) which enjoys a simplified construction, where the public key is re-used as a hash function.

Keywords: Deterministic encryption, hybrid encryption, code-based encryption, searchable encryption, database security.

1 Introduction

Background. The notion of security against *privacy adversary* (denoted as PRIV) for deterministic encryption was pioneered by Bellare et al [2] featuring an upgrade from the standard onewayness property. Instead of not leaking the whole plaintext, the ciphertext was demanded to leak, roughly speaking, no more than the plaintext statistics does. In other words, the PRIV-security definition (formulated in a manner similar to the semantic security definition of [7]) requires that a ciphertext must be essentially useless for adversary who is to compute some predicate on the corresponding plaintext. Achieving PRIV-security demands two important assumptions: 1) the plaintext space must be large enough and have a smooth (i.e. high min-entropy) distribution; 2) the plaintext and the predicate are independent of the public key.

Constructions satisfying two flavors of PRIV-security are presented in [2]: against chosen-plaintext (CPA) and chosen-ciphertext (CCA) attacks. The following three PRIV-CPA constructions are introduced in the random oracle (RO) model. The generic Encrypt-with-Hash (EwH) primitive features replacing of the coins used by the randomized encryption scheme with a hash of the public key concatenated with the message. The RSA deterministic OAEP (RSA-DOAEP) scheme provides us with length-preserving deterministic encryption.

M. Bras-Amorós and T. Høholdt (Eds.): AAECC 2009, LNCS 5527, pp. 159–168, 2009.

In the generic Encrypt-and-Hash (EaH) primitive, a "tag" in the form of the plaintext's hash is attached to the ciphertext of a randomized encryption scheme.

These results were extended by Boldyreva et al [4] and Bellare et al [3] presenting new extended definitions, proving relations between them, and introducing, among others, new constructions without random oracles.

Applications. The original motivation for this research comes from the demand on efficiently searchable encryption (ESE) in the database applications. Length-preserving schemes can also be used for encryption of legacy code and in the bandwidth-limited systems. Some more applications (although irrelevant to our work) to improving randomized encryption schemes were studied in [4, Sec. 8].

Motivation. The work [2, Sec. 5] sketches a method for encrypting long messages, but it is less efficient compared to the standard hybrid encryption, besides it is conjectured not to be length-preserving. Also, possible emerging of quantum computers raises demands for postquantum deterministic encryption schemes.

Our Contribution. In the random oracle model, we present a generic and efficient construction of length-preserving deterministic hybrid encryption. In a nutshell, we prove that the session key can be computed by concatenating the public key with the first message block and inputting the result into key derivation function. This is a kind of re-using the (sufficient) entropy of message, and it is secure due to the assumption that the message is high-entropy and independent of the key. Meanwhile, Bellare et al. employ the hybrid encryption in a conventional way, which first encrypts a random session key to further encrypt the data, obviously losing the length-preserving property. Hence, we show that the claim of Bellare et al [2, Sec. 5]: "However, if using hybrid encryption, RSA-DOAEP would no longer be length-preserving (since an encrypted symmetric key would need to be included with the ciphertext)" is overly pessimistic. To our best knowledge, this is the first example of length-preserving hybrid encryption.

For achieving postquantum deterministic encryption, we propose to plug in an IND-CPA secure variant [10] of the coding theory based (or code-based) McEliece PKE [9] into the generic constructions EaH and EwH, presented in [2, Sec. 5]. The McEliece PKE is believed to be resistant to quantum attacks, besides it has very fast encryption algorithm. Moreover, we point out a significant simplification: the public key (which is a generating matrix of some linear code) can be re-used as hash function.

Related Work. The deterministic hybrid encryption scheme is based on the same principle as the RSA-DOAEP scheme of [2, Sec. 5], we just fill the gap which was overlooked there.

Organization. The paper will be organized in the following way: Sec.2 provides the security definitions of deterministic encryption. Sec.3 gives the proposed generic and efficient construction of deterministic hybrid encryption, which leads to the first length-preserving construction, immediately. In Sec.4, we will provide deterministic encryption from the code-based PKE, which is postquantum secure and efficient due to the good property of the underlying PKE scheme. Next, in Sec.5, we further discuss how to extend the PRIV security to the chosen-ciphertext attack (CCA) scenario.

2 Preliminaries

Denote by "$|x|$" the cardinality of x. Denote by $\hat{\boldsymbol{x}}$ the vector and by $\hat{\boldsymbol{x}}[i]$ the i-th component of $\hat{\boldsymbol{x}}$ $(1 \leq i \leq |\hat{\boldsymbol{x}}|)$. Write $\hat{\boldsymbol{x}}||\hat{\boldsymbol{y}}$ for concatenation of vectors $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{y}}$. Let $x \leftarrow_R X$ denote the operation of picking x from the set X uniformly at random. Denote by $z \leftarrow A(x, y, ...)$ the operation of running algorithm A with input $(x, y, ...)$, to output z. Write $\log x$ as the logarithm with base 2. We also write $\Pr[A(x) = y : x \leftarrow_R X]$ the probability that A outputs y corresponding to input x, which is sampled from X. We say a function $\epsilon(k)$ is negligible, if for any constant c, there exists $k_0 \in \mathbb{N}$, such that $\epsilon < (1/k)^c$ for any $k > k_0$.

A public key encryption (PKE) scheme Π consists of a triple of algorithms $(\mathcal{K}, \mathcal{E}, \mathcal{D})$. The key generation algorithm $\mathcal{K}$ outputs a pair of public and secret keys $(\mathsf{pk}, \mathsf{sk})$ taking on input 1^k, a security parameter k in unitary notation. The encryption algorithm $\mathcal{E}$ on input pk and a plaintext $\hat{\boldsymbol{x}}$ outputs a ciphertext c. The decryption algorithm $\mathcal{D}$ takes sk and c as input and outputs the plaintext message $\hat{\boldsymbol{x}}$. We require that for any key pair $(\mathsf{pk}, \mathsf{sk})$ obtained from $\mathcal{K}$, and any plaintext $\hat{\boldsymbol{x}}$ from the plaintext space of Π, $\hat{\boldsymbol{x}} \leftarrow \mathcal{D}(\mathsf{sk}, \mathcal{E}(\mathsf{pk}, \hat{\boldsymbol{x}}))$.

Definition 1 (PRIV [2]). *Let a probabilistic polynomial-time (PPT) adversary $\mathcal{A}_{DE}$ against the privacy of the deterministic encryption $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, be a pair of algorithms $\mathcal{A}_{DE} = (\mathcal{A}_f, \mathcal{A}_g)$, where $\mathcal{A}_f, \mathcal{A}_g$ do not share any random coins or state. The advantage of adversary is defined as follows,*

$$\mathbf{Adv}^{priv}_{\Pi,\mathcal{A}_{DE}}(k) = \Pr[\mathbf{Exp}^{priv-1}_{\Pi,\mathcal{A}_{DE}}(k) = 1] - \Pr[\mathbf{Exp}^{priv-0}_{\Pi,\mathcal{A}_{DE}}(k) = 1]$$

where experiments are described as:

Experiment $\mathbf{Exp}^{priv-1}_{\Pi,\mathcal{A}_{DE}}(k)$:	**Experiment** $\mathbf{Exp}^{priv-0}_{\Pi,\mathcal{A}_{DE}}(k)$:
$(\mathsf{pk}, \mathsf{sk}) \leftarrow_R \mathcal{K}(1^k)$,	$(\mathsf{pk}, \mathsf{sk}) \leftarrow_R \mathcal{K}(1^k)$,
$(\hat{\boldsymbol{x}}_1, t_1) \leftarrow_R \mathcal{A}_f(1^k)$,	$(\hat{\boldsymbol{x}}_0, t_0) \leftarrow_R \mathcal{A}_f(1^k), (\hat{\boldsymbol{x}}_1, t_1) \leftarrow_R \mathcal{A}_f(1^k)$,
$c \leftarrow_R \mathcal{E}(1^k, \mathsf{pk}, \hat{\boldsymbol{x}}_1)$,	$c \leftarrow_R \mathcal{E}(1^k, \mathsf{pk}, \hat{\boldsymbol{x}}_0)$,
$g \leftarrow_R \mathcal{A}_g(1^k, \mathsf{pk}, c)$;	$g \leftarrow_R \mathcal{A}_g(1^k, \mathsf{pk}, c)$;
return 1 if $g = t_1$, *else return 0*	*return 1 if* $g = t_1$, *else return 0*

We say that Π is PRIV secure, *if $\mathbf{Adv}^{priv}_{\Pi,\mathcal{A}_{DE}}(k)$ is negligible, for any PPT $\mathcal{A}_{DE}$ with high min-entropy, where $\mathcal{A}_{DE}$ has a high min-entropy $\mu(k)$ means that $\mu(k) \in \omega(log(k))$, and $\Pr[\hat{\boldsymbol{x}}[i] = x : (\hat{\boldsymbol{x}}, t) \leftarrow_R \mathcal{A}_m(1^k)] \leq 2^{-\mu(k)}$ for all k, all $1 \leq i \leq |\hat{\boldsymbol{x}}|$, and any $x \in \{0,1\}^*$.*

In the underlying definition, the advantage of privacy adversary could be also written as

$$\mathbf{Adv}^{priv}_{\Pi,\mathcal{A}_{DE}}(k) = 2\Pr[\mathbf{Exp}^{priv-b}_{\Pi,\mathcal{A}_{DE}}(k) = b] - 1$$

where $b \in \{0,1\}$ and probability is taken over the choice of all of the random coins in the experiments.

Remarks. 1) The encryption algorithm Π need not be deterministic per se. For example, in a randomized encryption scheme, the random coins can be fixed in an appropriate way to yield a deterministic scheme (as explained in Sec.4);
2) As argued in [2], $\mathcal{A}_f$ has no access to the pk and $\mathcal{A}_g$ does not know the chosen plaintext input to encryption oracle by $\mathcal{A}_f$. This is required because the public key itself carries some non-trivial information about the plaintext if the encryption is deterministic.[1] Thus, equipping either $\mathcal{A}_f$ or $\mathcal{A}_g$ with both the public key and free choice of an input plaintext in the way of conventional indistinguishability notion [7] of PKE, the PRIV security cannot be achieved.

It is possible to build PRIV security from indistinguishability (IND) security, as observed in [2]. In the following, we recall the notion of IND security.

Definition 2 (IND-CPA). *We say a scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is IND-CPA secure, if the advantage $\mathbf{Adv}^{ind}_{\Pi,\mathcal{A}}$ of any PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ is negligible, (let s be the state information of $\mathcal{A}_1$, and $\hat{b} \in \{0,1\}$):*

$$\mathbf{Adv}^{ind}_{\Pi,\mathcal{A}}(k) = 2 \cdot \Pr\left[\begin{array}{l} \hat{b} = b : (\mathsf{pk}, \mathsf{sk}) \leftarrow_R \mathcal{K}(1^k), \\ (x_0, x_1, s) \leftarrow_R \mathcal{A}_1(1^k, \mathsf{pk}), \\ b \leftarrow_R \{0,1\}, c \leftarrow_R \mathcal{E}(1^k, \mathsf{pk}, x_b), \\ \hat{b} \leftarrow_R \mathcal{A}_2(1^k, c, s) \end{array}\right] - 1$$

Remark. IND security is required by a variety of cryptographic primitives. However, for an efficiently searchable encryption used in database applications, IND secure encryption may be considered as overkill. For such a strong encryption, it is not known how to arrange fast (i.e. logarithmic in the database size) search.

IND secure symmetric key encryption (SKE) has been carefully discussed in the literature, such as [6, Sec.7.2]. Given a key $K \in \{0,1\}^k$ and message m, an encryption algorithm outputs a ciphertext χ. Provided χ and K, a decryption algorithm outputs the message m uniquely. Note that for a secure SKE, outputs of the encryption algorithm could be considered uniformly distributed in the range, when encrypted under independent session keys. Besides, it is easy to build IND secure SKE.

Definition 3 (IND-CPA SKE). *A symmetric key encryption (SKE) scheme $\Lambda = (\mathcal{K}_{SK}, \mathcal{E}_{SK}, \mathcal{D}_{SK})$ with key space $\{0,1\}^k$, is indistinguishable against chosen plaintext attack (IND-CPA) if the advantage of any PPT adversary $\mathcal{B}$, $\mathbf{Adv}^{ind-cpa}_{\Lambda,\mathcal{B}}$ is negligible, where*

$$\mathbf{Adv}^{ind-cpa}_{\Lambda,\mathcal{B}}(k) = 2 \cdot \Pr\left[\begin{array}{l} \hat{b} = b : K \leftarrow_R \{0,1\}^k, b \leftarrow_R \{0,1\}, \\ \hat{b} \leftarrow_R \mathcal{B}^{\mathsf{LOR}(K,\cdot,\cdot,b)}(1^k) \end{array}\right] - 1,$$

where a left-or-right oracle $\mathsf{LOR}(K, M_0, M_1, b)$ returns $\chi \leftarrow_R \mathcal{E}_{SK}(K, M_b)$. Adversary $\mathcal{B}$ is allowed to ask LOR oracle, with two chosen message M_0, M_1 ($M_0 \neq M_1$, $|M_0| = |M_1|$).

[1] In other words, suppose that in Def. 1, $\mathcal{A}_f$ knows pk. Then, $\mathcal{A}_f$ can assign t_1 to be the ciphertext c, and hence $\mathcal{A}_g$ always wins the game (returns 1). Put it differently, although $\mathcal{A}_f$ and $\mathcal{A}_g$ are not allowed to share a state, knowledge of pk can help them to share it anyway.

Hybrid Encryption. In the seminal paper by Cramer and Shoup [6], the idea of hybrid encryption is rigorously studied. Note that typically, PKE is applied in key distribution process due to its expensive computational cost, while SKE is typically used for encrypting massive data flow using a freshly generated key for each new session. In hybrid encryption, PKE and SKE work in tandem: a randomly generated session key is first encrypted by PKE, then the plaintext is further encrypted on the session key by SKE. Hybrid encryption is more commonly used in practice than a sole PKE, since encryption/decryption of the former is substantially faster for long messages.

McEliece PKE. (denoted Π_M) Consists of the following triple of algorithms $(\mathcal{K}_M, \mathcal{E}_M, \mathcal{D}_M)$.

1. Key generation $\mathcal{K}_M$: On input λ, output $(\mathsf{pk}, \mathsf{sk})$. $n, t \in \mathbb{N}$, $t \ll n$
   - sk (Private Key): (S, φ, P)
     G': $l \times n$ generating matrix of a binary irreducible $[n, l]$ Goppa code which can correct a maximum of t errors. φ is an efficient bounded distance decoding algorithm of the underlying code, S: $l \times l$ non-singular matrix, P: $n \times n$ permutation matrix, chosen at random.
   - pk (Public Key): (G, t)
     G: $l \times n$ matrix given by a product of three matrices $SG'P$.
2. Encryption $\mathcal{E}_M$: Given pk and an l-bit plaintext m, randomly generate n-bit e with Hamming weight t, output ciphertext $c = mG \oplus e$.
3. Decryption $\mathcal{D}_M$: On input c, output m with private key sk.
   - Compute $cP^{-1} = (mS)G' \oplus eP^{-1}$, where P^{-1} is an inverse matrix of P.
   - Error correcting algorithm φ corresponding to G' applies to compute $mS = \varphi(cP^{-1})$.
   - Compute the plaintext $m = (mS)S^{-1}$.

IND-CPA security of the McEliece PKE can be achieved by padding the plaintext with a random bit-string r, $|r| = \lceil a \cdot l \rceil$ for some $0 < a < 1$. We refer to [10] for details.

3 Secure Deterministic Hybrid Encryption

In this section, we will present a generic composition of PKE and SKE to obtain deterministic hybrid encryption. Interestingly, the situation is different from conventional hybrid encryption. In that case, the overhead of communication cost includes at least the size of the session key, even if we pick the PKE scheme being a (length-preserving) one-way trapdoor permutation, e.g. RSA.

However, we notice that in PRIV security definition, both of public key and plaintext are not simultaneously known by $\mathcal{A}_f$ or $\mathcal{A}_g$. Hence, one can save on generating and encrypting a random session key. Instead, the secret session key could be extracted from the combination of public key and plaintext which are available to a legal user contrary to the adversary. As we show next, such an approach may need a little higher min-entropy, but it works in principle.

3.1 Generic Composition of PRIV-Secure PKE and IND-CPA Symmetric Key Encryption

Given a PRIV secure PKE scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, and an IND-CPA secure SKE scheme $\Lambda = (\mathcal{K}_{SK}, \mathcal{E}_{SK}, \mathcal{D}_{SK})$, we can achieve a deterministic hybrid encryption $HE = (\mathcal{K}_H, \mathcal{E}_H, \mathcal{D}_H)$. In the following, $H : \{0,1\}^* \mapsto \{0,1\}^k$ is a key derivation function (KDF), modeled as a random oracle. In the following section, we simply write input vector $\hat{\boldsymbol{x}}$ as x with length of $|\hat{\boldsymbol{x}}| = v$. Wlog, parse $x = \bar{x}||\underline{x}$, where the $|\bar{x}|$ and $|\underline{x}|$ are equivalent to the input domain of Π and Λ, respectively.

Table 1. Generic Construction of Deterministic Hybrid Encryption

$\mathcal{K}_H(1^k)$:	$\mathcal{E}_H(\mathsf{pk}, x)$:	$\mathcal{D}_H(\mathsf{sk}, c)$:
$(\mathsf{pk}, \mathsf{sk}) \leftarrow_R \mathcal{K}(1^k)$	Parses x to $\bar{x}\|\|\underline{x}$	Parse c to $\psi\|\|\chi$
Return $(\mathsf{pk}, \mathsf{sk})$	$\psi \leftarrow_R \mathcal{E}(1^k, \mathsf{pk}, \bar{x})$	$\bar{x} \leftarrow \mathcal{D}(\mathsf{sk}, \psi)$
	$K \leftarrow H(\mathsf{pk}\|\|\bar{x})$	$K \leftarrow H(\mathsf{pk}\|\|\bar{x})$
	$\chi \leftarrow_R \mathcal{E}_{SK}(K, \underline{x})$	$\underline{x} \leftarrow \mathcal{D}_{SK}(K, \chi)$
	Return $c = \psi\|\|\chi$	Return $x = \bar{x}\|\|\underline{x}$

In the Table 1, the proposed construction is simple, efficient, and can be generically built from any PRIV PKE and IND-CPA SKE. Note that the secret session key is required to have high min-entropy in order to deny a brute-force attack to SKE. However, thanks to the PRIV security, the high min-entropy requirement is inherently fulfilled for any PPT privacy adversary, so that we can build a reduction of security of the deterministic hybrid encryption to security of deterministic PKE. Next, we will provide a sketch of our proof.

3.2 Security Proof

Theorem 1. *In the random oracle model, given a PRIV PKE scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, and an IND-CPA SKE scheme $\Lambda = (\mathcal{K}_{SK}, \mathcal{E}_{SK}, \mathcal{D}_{SK})$, if there is a PRIV adversary $\mathcal{A}_H$ against the hybrid encryption $HE = (\mathcal{K}_H, \mathcal{E}_H, \mathcal{D}_H)$, then there exists PRIV adversary $\mathcal{A}$ or IND-CPA adversary $\mathcal{B}$, s.t.*

$$\mathbf{Adv}^{priv}_{HE,\mathcal{A}_H}(k) \leq \mathbf{Adv}^{priv}_{\Pi,\mathcal{A}}(k) + \mathbf{Adv}^{ind-cpa}_{\Lambda,\mathcal{B}}(k) + q_h v / 2^{\mu}$$

where q_h is an upper bound on the number of queries to random oracle H, v is the plaintext size of Π, μ is defined by high min-entropy of PRIV security of Π.

Proof. Since we assume a PPT adversary $\mathcal{A}_H = (\mathcal{A}_f, \mathcal{A}_g)$ against the HE scheme, according to the definition of PRIV, there must be a non-negligible advantage in the following experiments.

More precisely, if a successful adversary exists, then

$$\mathbf{Adv}^{priv}_{HE,\mathcal{A}_H}(k) = \Pr[\mathbf{Exp}^{priv-1}_{HE,\mathcal{A}_H}(k) = 1] - \Pr[\mathbf{Exp}^{priv-0}_{HE,\mathcal{A}_H}(k) = 1]$$

Experiment $\mathbf{Exp}^{priv-1}_{HE,\mathcal{A}_H}(k)$:
$(\mathsf{pk}, \mathsf{sk}) \leftarrow_R \mathcal{K}(1^k)$;
$(x_1, t_1) \leftarrow_R \mathcal{A}_f(1^k)$;
Parse x_1 to $\bar{x}_1||\underline{x}_1$;
$\psi \leftarrow_R \mathcal{E}(1^k, \mathsf{pk}, \bar{x}_1)$;
$K \leftarrow H(\mathsf{pk}||\bar{x}_1)$;
$\chi \leftarrow_R \mathcal{E}_{SK}(K, \underline{x}_1)$;
$c \leftarrow \psi||\chi$;
$g \leftarrow_R \mathcal{A}_g(1^k, \mathsf{pk}, c)$;
return 1 if $g = t_1$,
else return 0

Experiment $\mathbf{Exp}^{priv-0}_{HE,\mathcal{A}_H}(k)$:
$(\mathsf{pk}, \mathsf{sk}) \leftarrow_R \mathcal{K}(1^k)$;
$(x_0, t_0) \leftarrow_R \mathcal{A}_f(1^k)$, $(x_1, t_1) \leftarrow_R \mathcal{A}_f(1^k)$;
Parse x_0 to $\bar{x}_0||\underline{x}_0$;
$\psi' \leftarrow_R \mathcal{E}(1^k, \mathsf{pk}, \bar{x}_0)$;
$K' \leftarrow H(\mathsf{pk}||\bar{x}_0)$;
$\chi' \leftarrow_R \mathcal{E}_{SK}(K', \underline{x}_0)$;
$c' \leftarrow \psi'||\chi'$;
$g \leftarrow_R \mathcal{A}_g(1^k, \mathsf{pk}, c')$;
return 1 if $g = t_1$,
else return 0

is non-negligible for some $\mathcal{A}_H$. Next we present a simulator which gradually modifies the above experiments such that the adversary does not notice it. Our goal is to show that $\mathbf{Adv}^{priv}_{HE,\mathcal{A}_H}(k)$ is almost as big as the corresponding advantages defined for PRIV security of the PKE scheme and IND-CPA security of the SKE scheme, which are assumed negligible.

Because of the high min-entropy requirement of PRIV adversary, it is easy to see that $x_0 \neq x_1$, except with negligible probability. Thus, there must be $\bar{x}_0 \neq \bar{x}_1$ or $\underline{x}_0 \neq \underline{x}_1$, or both. Hence, we need to consider the following cases.

Case [$\bar{x}_0 \neq \bar{x}_1$] Since $x_0 \neq x_1$ and $\bar{x}_0 \neq \bar{x}_1$, the right part of x_b ($b \in \{0, 1\}$), could be equal or not.

- When $\underline{x}_0 = \underline{x}_1$, the adversary has two targets, such as Π and Λ in two experiments. First look at the SKE scheme Λ. In this case, the inputs to Λ in two experiments are the same, but still unknown to $\mathcal{A}_g$. The key derivation function H outputs $K \leftarrow H(\mathsf{pk}||\bar{x}_1)$ and $K' \leftarrow H(\mathsf{pk}||\bar{x}_0)$. Since $\bar{x}_0 \neq \bar{x}_1$, we have $K \neq K'$. Note that $\mathcal{A}_g$ does not know x_0 nor x_1, thus does not know K, K', either. Then, $\mathcal{A}_g$ must tell which of χ, χ' is the corresponding encryption under the unknown keys without knowing $\underline{x}_0, \underline{x}_1 (\underline{x}_0 = \underline{x}_1)$, which is harder than breaking IND-CPA security and that could be bounded by $\mathbf{Adv}^{ind-cpa}_{\Lambda,\mathcal{B}}(k)$.
  On the other hand, the adversary can also challenge the PKE scheme Π to distinguish two experiments, but it will break the PRIV security. More precisely, the advantage in distinguishing ψ, ψ' with certain K, K' is at most $\mathbf{Adv}^{priv}_{\Pi,\mathcal{A}}(k)$, since K, K' are not output explicitly and unavailable to adversary.
- when $\underline{x}_0 \neq \underline{x}_1$, this case is similar to the above, except that the inputs to Λ are different. $\mathcal{A}_g$ can do nothing given χ, χ' only, hence $\mathcal{A}_g$'s possible attack must be focused on Π, and its advantage can be bounded by $\mathbf{Adv}^{priv}_{\Pi,\mathcal{A}}(k)$.

Case [$\underline{x}_0 \neq \underline{x}_1$] Similarly, there must be either $\bar{x}_0 \neq \bar{x}_1$ or $\bar{x}_0 = \bar{x}_1$.

- when $\bar{x}_0 = \bar{x}_1$, the same session key $K \leftarrow H(\mathsf{pk}||\bar{x}_b)$ ($b \in \{0, 1\}$) is used for Λ. In this case, the ciphertexts ψ, ψ' are the same, adversary will focus on distinguishing the χ, χ'. Note that $\mathcal{A}_f$ cannot compute K even though he knows the $\bar{x}_0 = \bar{x}_1$, because pk is not known to him

(otherwise, it will break the PRIV security of Π immediately!). Thus, the successful distinguishing requires $\mathcal{A}_g$ to choose the same $\bar{x}_0 = \bar{x}_1$ when querying to the random oracle. Then, $\mathcal{A}_g$ has a harder game than IND-CPA (because it does not know $\underline{x}_0, \underline{x}_1$), whose advantage is bounded by $\mathbf{Adv}^{ind-cpa}_{\Lambda,\mathcal{B}}(k)$.
In order to be sure that adversary $(\mathcal{A}_f, \mathcal{A}_g)$ mounting a brute-force attack to find out the session key of Λ cannot succeed, the probability to find the key in searching all the random oracle queries should be taken into account as well. Suppose that adversary makes at most q_h queries to its random oracle, and the Π's plaintext size is v. Then, this probability could be upper bounded by $q_h v/2^{\mu}$ (Note that this bound is in nature similar to that in [2, Sec.6.1]).

– when $\bar{x}_0 \neq \bar{x}_1$, as we have discussed above, this will break the PRIV security of Π, and advantage of adversary could be bounded by $\mathbf{Adv}^{priv}_{\Pi,\mathcal{A}}(k)$.

Summarizing, we conclude that in all cases when $(\mathcal{A}_f, \mathcal{A}_g)$ intends to break the PRIV security of our HE scheme, its advantage of distinguishing two experiments is bounded by the sum of $\mathbf{Adv}^{priv}_{\Pi,\mathcal{A}}(k)$, $q_h v/2^{\mu}$ and $\mathbf{Adv}^{ind-cpa}_{\Lambda,\mathcal{B}}(k)$. □

Length-preserving Deterministic Hybrid Encryption. The first length-preserving PRIV PKE scheme is RSA-DOAEP due to [2]. The length-preserving property is important in practical use, such as bandwidth-restricted applications. RSA-DOAEP makes use of the RSA trapdoor permutation and with a modified 3-round Feistel network achieves the same sizes of input and output. As we have proved in Theorem 1, a construction proposed in Table 1 leads to a deterministic hybrid encryption.

In particular, RSA-DOAEP + IND-CPA SKE ⇒ a length-preserving deterministic hybrid encryption, because both RSA-DOAEP and IND-CPA SKE are length-preserving. Note that in [2, Sec.5.2], it is argued that RSA-DOAEP based hybrid encryption scheme cannot be length-preserving any more, because a random session key has to be embedded in RSA-DOAEP. However, by re-using the knowledge of public key pk and a part of the message, we can indeed build the first length-preserving deterministic hybrid encryption, which is not only convenient in practice, but also meaningful in theory.

4 Deterministic Encryption from Code-Based PKE

From a postquantum point of view, it is desirable to obtain deterministic encryption based on assumptions other than RSA or discrete log. Code-based PKE, such as McEliece PKE [9] is considered a promising candidate after being carefully studied for over thirty years.

To our surprise, it is not the only motivation to achieve deterministic encryption from code-based PKE. Another good property of the McEliece PKE and its variants is that its public key could be used as a hash function to digest the message, which is originally noted in Stern's paper [11], and recently designed by [1,8]. The advantage that public key itself is able to work as a hash function,

Table 2. Construction of EwH Deterministic Encryption

$\mathcal{K}(1^k)$:	$\mathcal{E}(\mathsf{pk}, H_N, x)$:	$\mathcal{D}(\mathsf{sk}, H_M, c)$:
$(\mathsf{pk}, \mathsf{sk}) \leftarrow_R \mathcal{K}_M(1^k)$	$R \leftarrow H_M(x)$	$x, r', e \leftarrow \mathcal{D}_M(\mathsf{sk}, c)$
$H_M \leftarrow \mathcal{H}(1^k, \mathsf{pk})$	Parse R to $r \| \| r_e$	Decode e to r'_e
Return $(\mathsf{pk}, H_M, \mathsf{sk})$	Encode r_e to e	$R' \leftarrow r' \| \| r'_e$, $R \leftarrow H_M(x)$
	$c \leftarrow \mathcal{E}_M(\mathsf{pk}, r \| \| x; e)$	Return x if $R = R'$
	Return c	Otherwise, return $\perp$

Table 3. Construction of EaH Deterministic Encryption

$\mathcal{K}(1^k)$:	$\mathcal{E}(\mathsf{pk}, H_M, x)$:	$\mathcal{D}(\mathsf{sk}, H_M, c \| \| T)$:
$(\mathsf{pk}, \mathsf{sk}) \leftarrow_R \mathcal{K}_M(1^k)$	$T \leftarrow H_N(x)$	$x, r, e \leftarrow \mathcal{D}_M(\mathsf{sk}, c)$
$H_N \leftarrow \mathcal{H}(1^k, \mathsf{pk})$	$r \leftarrow_R \{0,1\}^{l_p}$	$T' \leftarrow H_N(x)$
Return $(\mathsf{pk}, H_M, \mathsf{sk})$	$e \leftarrow_R \{0,1\}^n$, s.t. $Hw(e) = t$	Return x if $T = T'$
	$c \leftarrow \mathcal{E}_M(\mathsf{pk}, r \| \| x; e)$	Otherwise, return $\perp$
	Return $c \| \| T$	

can do us a favor to build efficient postquantum deterministic encryption. We call this Hidden Hash (HH) property of McEliece PKE. Henceforth, we assume that this function behaves as a random oracle.

In [2], two constructions satisfying PRIV security have been proposed: Encrypt-with-Hash (EwH) and Encrypt-and-Hash (EaH). Adapting the HH property of the McEliece PKE to the both constructions, we can achieve PRIV secure deterministic encryption. For proving PRIV security, we require the McEliece PKE to be IND-CPA secure, which has been proposed in [10]. (The proofs are deferred to the full version of this paper).

Construction of EwH. Let $\Pi_M = (\mathcal{K}_M, \mathcal{E}_M, \mathcal{D}_M)$ be the IND-CPA McEliece PKE as described in Section 2, based on $[n, l, 2t+1]$ Goppa code family, with l_p-bit padding where $l_p = \lceil a \cdot l \rceil$ for some $0 < a < 1$, and plaintext length $l_m = l - l_p$. Let $\mathcal{H}$ be a hash family defined over a set of public keys of the McEliece PKE. $H_M : \{0,1\}^{l_m} \mapsto \{0,1\}^{l_p + \log \sum_{i=1}^{t} \binom{n}{t}}$ and $H_N : \{0,1\}^{l_m} \mapsto \{0,1\}^{2k}$ are uniquely defined by 1^k and pk. Without knowledge of pk, there is no way to compute H_M or H_N (refer to [1,8] for details). e is an error vector, s.t. $|e| = n$ with Hamming weight $Hw(e) = t$. According to Cover's paper [5], it is quite efficient to find an injective mapping to encode the (short) bit string r_e into e, and vice versa.

Our EwH scheme is presented in Table 2.

Note that compared with the EwH scheme proposed by Bellare et al. [2], our scheme does not need to include pk into the hash, because hash function H_M itself is made of pk. Public key pk could be considered as a part of the algorithm of the hash function, as well. When we model H_M as a random oracle, we can easily prove the PRIV security in a similar way as Bellare et al's EwH.

A more favorable, efficiently searchable encryption (ESE) with PRIV security is EaH. EaH aims to model the practical scenario in database security, where a deterministic encryption of some keywords works as a tag attached to the encrypted data. To search the target data, it is only required to compute the

deterministic tag and compare it within the database, achieving a search time which is logarithmic in database size.

Construction of EaH. The description of McEliece PKE is similar to the above. EaH scheme is described in Table 3. The HH property is employed in order to achieve PRIV secure efficiently searchable encryption.

5 Concluding Remarks

Extension to Chosen-Ciphertext Security. Above, we have proposed several PRIV secure deterministic encryption schemes, in CPA case. A stronger attack scenario, CCA, requires a little more care. As commented in [2], PRIV-CCA could be obtained from PRIV-CPA scheme with some additional cost, such as one-time signatures or other authentication techniques to deny a CCA attacker. We can employ those techniques to lift up CPA to CCA. The important issue is that we have achieved very efficient PRIV-CPA secure building blocks which enjoy some advantages over previous works.

Open Question. Proving our constructions secure in the standard model is an open question and the topic of our future work.

References

1. Augot, D., Finiasz, M., Sendrier, N.: A Family of Fast Syndrome Based Cryptographic Hash Functions. In: Dawson, E., Vaudenay, S. (eds.) Mycrypt 2005. LNCS, vol. 3715, pp. 64–83. Springer, Heidelberg (2005)
2. Bellare, M., Boldyreva, A., O'Neill, A.: Deterministic and Efficiently Searchable Encryption. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 535–552. Springer, Heidelberg (2007)
3. Bellare, M., Fischlin, M., O'Neill, A., Ristenpart, T.: Deterministic Encryption: Definitional Equivalences and Constructions without Random Oracles. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 360–378. Springer, Heidelberg (2008)
4. Boldyreva, A., Fehr, S., O'Neill, A.: On Notions of Security for Deterministic Encryption, and Efficient Constructions without Random Oracles. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 335–359. Springer, Heidelberg (2008)
5. Cover, T.: Enumerative source encoding. IEEE IT 19(1), 73–77 (1973)
6. Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM Journal on Computing 33(1), 167–226 (2003)
7. Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences 28(2), 270–299 (1984)
8. Finiasz, M.: Syndrome Based Collision Resistant Hashing. In: Buchmann, J., Ding, J. (eds.) PQCrypto 2008. LNCS, vol. 5299, pp. 137–147. Springer, Heidelberg (2008)
9. McEliece, R.J.: A public-key cryptosystem based on algebraic coding theory. Deep Space Network Progress Rep. 42-44, 114–116 (1978)
10. Nojima, R., Imai, H., Kobara, K., Morozov, K.: Semantic Security for the McEliece Cryptosystem without Random Oracles. Designs, Codes and Cryptography 49(1-3), 289–305 (2008)
11. Stern, J.: A new identification scheme based on syndrome decoding. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 13–21. Springer, Heidelberg (1994)

Noisy Interpolation of Multivariate Sparse Polynomials in Finite Fields

Álvar Ibeas[1] and Arne Winterhof[2]

[1] University of Cantabria
E-39071 Santander, Spain
alvar.ibeas@unican.es
[2] Johann Radon Institute for Computational and Applied Mathematics
Austrian Academy of Sciences
Altenbergerstr. 69, 4040 Linz, Austria
arne.winterhof@oeaw.ac.at

Abstract. We consider the problem of recovering an unknown sparse multivariate polynomial $f \in \mathbb{F}_p[X_1, \ldots, X_m]$ over a finite field $\mathbb{F}_p$ of prime order p from approximate values of $f(t_1, \ldots, t_m)$ at polynomially many points $(t_1, \ldots, t_m) \in \mathbb{F}_p^m$ selected uniformly at random. Our result is based on a combination of bounds on exponential sums with the lattice reduction technique.

Keywords: Noisy interpolation, Sparse polynomials, Hidden number problem, Lattice reduction, Exponential sums.

1 Introduction

Let p be a prime. We will use the notation $\mathbb{F}_p$ for the finite field of p elements and identify it with the integers in the range $\{0, \ldots, p-1\}$. For an integer s we denote by $\lfloor s \rceil_p$ the remainder of s on division by p.

For a positive integer m we consider the problem of finding an unknown multivariate nonconstant polynomial $f \in \mathbb{F}_p[X_1, \ldots, X_m]$ of weight at most w from approximate values of $\lfloor f(t_1, \ldots, t_m) \rceil_p$ at polynomially many points $(t_1, \ldots, t_m) \in \mathbb{F}_p^m$. More precisely, using the abbreviations $\mathbf{X} = (X_1, \ldots, X_m)$ and $\mathbf{X}^{e_j} = X_1^{e_{j,1}} \cdots X_m^{e_{j,m}}$ if $e_j = e_{j,1} + e_{j,2}p + \cdots + e_{j,m}p^{m-1}$ with $0 \le e_{j,1}, e_{j,2}, \ldots, e_{j,m} < p$, the polynomial $f = f(\mathbf{X})$ is of the form

$$f(\mathbf{X}) = \sum_{j=1}^{w} \alpha_j \mathbf{X}^{e_j}, \tag{1}$$

where $1 \le e_1 < e_2 < \cdots < e_w < p^m$.

A polynomial time algorithm to recover a univariate sparse polynomial $f \in \mathbb{F}_p[X]$ from approximations to $\lfloor f(j) \rceil_p$ at random j was introduced and investigated in [3,4]. The case of a univariate polynomial $f(X) = \alpha X$ corresponds to the *hidden number problem*, see [2,5,6] and references therein. In the present paper we present an algorithm for arbitrary m.

M. Bras-Amorós and T. Høholdt (Eds.): AAECC 2009, LNCS 5527, pp. 169–178, 2009.

The algorithm is based on the *lattice reduction technique*. To be more precise, a certain lattice is constructed which contains a vector, associated with the coefficients of f, and which is close to a certain known vector. Using the lattice reduction technique one can hope to recover this vector. Then, in order to make these arguments rigorous, that is, to show that there are no other lattice vectors which are close enough to the known "target" vector, we need a certain uniform distribution property of f which is only guaranteed if the total degree of f is small enough. However, using a weaker uniform distribution property of f and ideas reminiscent to *Waring's problem in a finite field*, see for example [7], we can adapt the algorithm to polynomials of much larger degree.

We use $\log z$ to denote the binary logarithm of $z > 0$. For a prime p and $\tau \geq 0$ we denote by $\mathrm{MSB}_{\tau,p}(x)$ any integer u such that $|\lfloor x \rceil_p - u| \leq p/2^{\tau+1}$. Roughly speaking, $\mathrm{MSB}_{\tau,p}(x)$ gives the τ *most significant bits* of x, however this definition is more flexible and suits better our purposes. In particular we remark that τ need not be an integer. We use bold lowercase letters to denote vectors in $\mathbb{F}_p^m$ and use the analogue notation to $\mathbf{X}^e$: $\mathbf{x} = (x_1, \ldots, x_m)$ and $\mathbf{x}^e = x_1^{e_1} \cdots x_m^{e_m}$ if $e = e_1 + e_2 p + \cdots + e_m p^{m-1}$ with $0 \leq e_1, \ldots, e_m < p$.

2 Preliminaries

For a prime p we denote $\mathbf{e}_p(z) = \exp(2\pi i z/p)$. The following bound on exponential sums of a univariate polynomial is given in [1, Corollary 1.1]. It is nontrivial for polynomials of very large degree relative to p.

Lemma 1. *Let $F \in \mathbb{F}_p[X]$ be a nonconstant polynomial of degree D and weight w. For any $\varepsilon \in (0,1)$, if $D \leq \frac{p(\log(w \log p))^{1-\varepsilon}}{\log p}$, the following bound holds provided that p is large enough:*

$$\left| \sum_{x \in \mathbb{F}_p} \mathbf{e}_p(F(x)) \right| \leq p \left(1 - \frac{1}{(w \log p)^{1+\varepsilon}} \right).$$

The following result extends this bound to exponential sums of a multivariate polynomial.

Lemma 2. *Let $F \in \mathbb{F}_p[\mathbf{X}]$ be a nonconstant multivariate polynomial of total degree D and weight at most w. For any $\varepsilon \in (0,1)$ and $0 < c < 1$ with*

$$\min_{\substack{i=1,\ldots,m \\ \deg_{X_i}(F) > 0}} \deg_{X_i}(F) \leq \frac{p(\log \log p)^{1-\varepsilon}}{\log p} \quad \text{and} \quad D \leq cp, \tag{2}$$

the following bound holds provided that p is large enough:

$$\left| \sum_{\mathbf{x} \in \mathbb{F}_p^m} \mathbf{e}_p(F(\mathbf{x})) \right| \leq p^m \left(1 - \frac{1}{(w \log p)^{1+\varepsilon}} \right).$$

Proof. We may assume that $\deg_{X_m}(F) = \min_{i=1,\ldots,m} \deg_{X_i}(F) > 0$ and

$$F(\mathbf{X}) = \sum_{i=1}^{w_0} a_i(X_1, \ldots, X_{m-1}) X_m^{e_{j_i}}$$

for some $w_0 \le w$ and polynomials $a_1, \ldots, a_{w_0} \in \mathbb{F}_p[X_1, \ldots, X_{m-1}]$ not identically zero. Put $S = \left|\sum_{\mathbf{x} \in \mathbb{F}_p^m} \mathbf{e}_p(F(\mathbf{x}))\right|$. We have

$$S \le \sum_{x_1,\ldots,x_{m-1} \in \mathbb{F}_p} \left| \sum_{x_m \in \mathbb{F}_p} \mathbf{e}_p\left(\sum_{i=1}^{w_0} a_i(x_1, \ldots, x_{m-1}) x_m^{e_{j_i}} \right) \right|.$$

The polynomial a_{w_0} has at most $p^{m-2}D$ zeros and therefore, the univariate polynomials $F(x_1, \ldots, x_{m-1}, X_m)$ are constant for at most $p^{m-2}D$ choices of the values $x_1, \ldots, x_{m-1}$. Using Lemma 1 with a certain $\varepsilon' < \varepsilon$ for the rest, we get

$$\frac{S}{p^{m-1}} \le D + (p - D)\left(1 - \frac{1}{(w_0 \log p)^{1+\varepsilon'}}\right) \le p\left(1 - \frac{1-c}{(w_0 \log p)^{1+\varepsilon'}}\right),$$

and the result follows. □

For small total degree D we also use the Weil bound

$$\left| \sum_{\mathbf{x} \in \mathbb{F}_p} \mathbf{e}_p(F(\mathbf{x})) \right| < D p^{m-1/2}. \tag{3}$$

Let $\mathcal{S} \subseteq \mathbb{F}_p^m$ be a subset of cardinality s. For integers $h, k \ge 1$, r, and a polynomial $f(\mathbf{X}) = \sum_{i=1}^{w} \alpha_i \mathbf{X}^{e_i} \in \mathbb{F}_p[\mathbf{X}]$, we denote by $N(\mathcal{S}, k, f, h, r)$ the number of solutions in $\mathcal{S}^k$ of the "Waring-like" equation:

$$f(\mathbf{x}_1) + \cdots + f(\mathbf{x}_k) \equiv t \bmod p, \ r + 1 \le t \le r + h.$$

Let $\mathcal{P}_f$ be the set of polynomials $\sum_{i=1}^{w} \beta_i \mathbf{X}^{e_i}$ which are distinct to f. We say that the set $\mathcal{S}$ is (Δ, k, f)-*homogeneously distributed* modulo p if for every $g \in \mathcal{P}_f$

$$\max_{1 \le h, r \le p} \left| N(\mathcal{S}, k, f - g, h, r) - h s^k p^{-1} \right| \le \Delta s^k.$$

Lemma 3. *Let $f \in \mathbb{F}_p[\mathbf{X}]$ such that*

$$\left| \sum_{\mathbf{x} \in \mathbb{F}_p^m} \mathbf{e}_p(a(f(\mathbf{x}) - g(\mathbf{x}))) \right| \le B p^m,$$

for all $g \in \mathcal{P}_f$ and $a \in \mathbb{F}_p^*$. Then, for every integer $k \ge 1$, $\mathbb{F}_p^m$ is $(B^k \log p, k, f)$-homogeneously distributed modulo p.

Proof. By the well-known identity $\sum_{a=0}^{p-1} \mathbf{e}_p(au) = \begin{cases} 0, \text{ if } u \not\equiv 0 \bmod p, \\ p, \text{ if } u \equiv 0 \bmod p, \end{cases}$ we have

$$\begin{aligned} & N(\mathbb{F}_p^m, k, f-g, h, r) \\ &= \sum_{t=r+1}^{r+h} \sum_{\mathbf{x}_1,\ldots,\mathbf{x}_k \in \mathbb{F}_p^m} \frac{1}{p} \sum_{a=0}^{p-1} \mathbf{e}_p(a(f(\mathbf{x}_1) - g(\mathbf{x}_1) + \cdots + f(\mathbf{x}_k) - g(\mathbf{x}_k) - t)) \\ &= \frac{1}{p} \sum_{a=0}^{p-1} \left(\sum_{t=r+1}^{r+h} \mathbf{e}_p(-at) \right) \left(\sum_{\mathbf{x} \in \mathbb{F}_p^m} \mathbf{e}_p(a(f(\mathbf{x}) - g(\mathbf{x}))) \right)^k . \end{aligned}$$

The term corresponding to $a = 0$ is hp^{mk-1}. Applying the estimate

$$\max_{1 \le h \le p} \sum_{a=1}^{p-1} \left| \sum_{t=r+1}^{r+h} \mathbf{e}_p(at) \right| \le p \log p$$

to other terms we obtain the result. □

As in [3,4], the presented method uses lattice basis reduction techniques. We briefly review a result on lattices. Let $\{\mathbf{b}_1, \ldots, \mathbf{b}_s\}$ be a set of linearly independent vectors in $\mathbb{R}^r$. The set

$$\mathcal{L} = \{c_1\mathbf{b}_1 + \cdots + c_s\mathbf{b}_s \mid c_1, \ldots, c_s \in \mathbb{Z}\}$$

is called an s-dimensional lattice, and the set $\{\mathbf{b}_1, \ldots, \mathbf{b}_s\}$ is called a *basis* of $\mathcal{L}$. The search of elements in a lattice with small norm or close to a given one is a widely investigated problem. As in [3,4], we use the following result.

Lemma 4. *There exists a deterministic polynomial time algorithm which, for a given lattice $\mathcal{L} \subset \mathbb{R}^s$ and vector $\mathbf{r} = (r_1, \ldots, r_s) \in \mathbb{R}^s$, finds a lattice vector $\mathbf{v} = (v_1, \ldots, v_s)$ satisfying the inequality*

$$\begin{aligned} \sum_{i=1}^{s} (v_i - r_i)^2 \le & \exp\left(O\left(\frac{s \log^2 \log s}{\log s} \right) \right) \\ & \times \min\left\{ \sum_{i=1}^{s} (z_i - r_i)^2, \quad \mathbf{z} = (z_1, \ldots, z_s) \in \mathcal{L} \right\}. \end{aligned}$$

3 General Interpolation Result

Let $\tau > 0$ be a real number and $\mathbf{e} = (e_1, \ldots, e_w)$ a vector of integers such that $1 \le e_1 < \cdots < e_w < p^m$. For $\mathbf{t}_{1,1}, \ldots, \mathbf{t}_{1,k}; \ldots; \mathbf{t}_{d,1}, \ldots, \mathbf{t}_{d,k} \in \mathbb{F}_p^m$ we denote by $\mathcal{L}_{\tau,\mathbf{e},p}(\mathbf{t}_{1,1}, \ldots, \mathbf{t}_{d,k})$ the $(d+w)$-dimensional lattice generated by the rows of the

following matrix:

$$\begin{pmatrix} \mathbf{t}_{1,1}^{e_1}+\cdots+\mathbf{t}_{1,k}^{e_1} & \cdots & \mathbf{t}_{d,1}^{e_1}+\cdots+\mathbf{t}_{d,k}^{e_1} & 1/2^{\tau+1} & \cdots & 0 \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ \mathbf{t}_{1,1}^{e_w}+\cdots+\mathbf{t}_{1,k}^{e_w} & \cdots & \mathbf{t}_{d,1}^{e_w}+\cdots+\mathbf{t}_{d,k}^{e_w} & 0 & \cdots & 1/2^{\tau+1} \\ p & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & & \vdots \\ 0 & \cdots & p & 0 & \cdots & 0 \end{pmatrix}. \tag{4}$$

Lemma 5. *Let p be a sufficiently large n-bit prime and let $w \geq 1$ be an integer. We define*

$$d = 2\left\lceil (nw)^{1/2} \right\rceil \quad \text{and} \quad \eta = 0.5(nw)^{1/2} + 3.$$

Let $f \in \mathbb{F}_p[\mathbf{X}]$ be defined by (1) and k be a positive integer such that $\mathcal{S} \subseteq \mathbb{F}_p^m$ is $(2^{-\eta}, k, f)$-homogeneously distributed modulo p.

Assume that $\mathbf{t}_{1,1}, \ldots, \mathbf{t}_{d,k} \in \mathcal{S}$ are chosen uniformly and independently at random. Then with probability $P \geq 1 - 2^{-\eta}$ for any vector

$$\mathbf{s} = (s_1, \ldots, s_d, 0, \ldots, 0)$$

with

$$\left(\sum_{i=1}^{d} (\lfloor f(\mathbf{t}_{i,1}) + \cdots + f(\mathbf{t}_{i,k}) \rfloor_p - s_i)^2 \right)^{1/2} \leq 2^{-\eta} p,$$

all vectors $\mathbf{v} = (v_1, \ldots, v_d, v_{d+1}, \ldots, v_{d+w}) \in \mathcal{L}_{\tau,\mathbf{e},p}(\mathbf{t}_{1,1}, \ldots, \mathbf{t}_{d,k})$ satisfying

$$\left(\sum_{i=1}^{d} (v_i - s_i)^2 \right)^{1/2} \leq 2^{-\eta} p$$

are of the form

$$\mathbf{v} = \left(\left\lfloor \sum_{j=1}^{w} \beta_j (\mathbf{t}_{1,1}^{e_j} + \cdots + \mathbf{t}_{1,k}^{e_j}) \right\rfloor_p, \ldots, \left\lfloor \sum_{j=1}^{w} \beta_j (\mathbf{t}_{d,1}^{e_j} + \cdots + \mathbf{t}_{d,k}^{e_j}) \right\rfloor_p, \beta_1/2^{\tau+1}, \ldots, \beta_w/2^{\tau+1} \right)$$

with some integers $\beta_j \equiv \alpha_j \bmod p$, $j = 1, \ldots, w$.

Proof. We define the modular distance between two integers λ and μ as

$$\mathrm{dist}_{\mathrm{p}}(\lambda, \mu) = \min_{b \in \mathbb{Z}} |\lambda - \mu - bp| = \min \{ \lfloor \lambda - \mu \rfloor_p, p - \lfloor \lambda - \mu \rfloor_p \}.$$

Because of the homogeneous distribution property, we see that for any polynomial $g \in \mathcal{P}_f$ the probability $P(g)$ that

$$\operatorname{dist}_{\mathrm{p}}\left(g(\mathbf{t}_1) + \cdots + g(\mathbf{t}_k), f(\mathbf{t}_1) + \cdots + f(\mathbf{t}_k)\right) \le 2^{-\eta+1}p$$

for $\mathbf{t}_1, \ldots, \mathbf{t}_k \in \mathcal{S}$ selected uniformly at random is

$$P(g) \le 2^{-\eta+2} + 2^{-\eta} = \frac{5}{2^\eta}.$$

Therefore, for any $g \in \mathcal{P}_f$, the probability that there is $i \in [1, d]$ with

$$\operatorname{dist}_{\mathrm{p}}\left(g(\mathbf{t}_{i,1}) + \cdots + g(\mathbf{t}_{i,k}), f(\mathbf{t}_{i,1}) + \cdots + f(\mathbf{t}_{i,k})\right) > 2^{-\eta+1}p$$

equals $1 - P(g)^d \ge 1 - \left(\frac{5}{2^\eta}\right)^d$, where the probability is taken over $\mathbf{t}_{1,1}, \ldots, \mathbf{t}_{d,k} \in \mathcal{S}$ chosen uniformly and independently at random.

Since $\#\mathcal{P}_f = p^w - 1$, we obtain

$$\begin{aligned}\Pr\Big[\forall g \in \mathcal{P}_f,\ \exists i \in [1,d] \mid \\ \operatorname{dist}_{\mathrm{p}}\left(g(\mathbf{t}_{i,1}) + \cdots + g(\mathbf{t}_{i,k}), f(\mathbf{t}_{i,1}) + \cdots + f(\mathbf{t}_{i,k})\right) > 2^{-\eta+1}p\Big] \\ \ge 1 - (p^w - 1)\left(\frac{5}{2^\eta}\right)^d > 1 - 2^{-\eta}\end{aligned}$$

because

$$d(\eta - \log 5) > 2(nw)^{1/2}(0.5(nw)^{1/2} + 3 - \log 5) > nw + \eta \ge w \log p + \eta,$$

provided that p is large enough.

Indeed, we fix some $(\mathbf{t}_{1,1}, \ldots, \mathbf{t}_{d,k}) \in \mathcal{S}^{dk}$ with

$$\min_{g \in \mathcal{P}_f} \max_{i \in [1,d]} \operatorname{dist}_{\mathrm{p}}\left(g(\mathbf{t}_{i,1}) + \cdots + g(\mathbf{t}_{i,k}), f(\mathbf{t}_{i,1}) + \cdots + f(\mathbf{t}_{i,k})\right) > 2^{-\eta+1}p. \quad (5)$$

Let $\mathbf{v} \in \mathcal{L}_{\tau,\mathbf{e},p}(\mathbf{t}_{1,1}, \ldots, \mathbf{t}_{d,k})$ be a lattice point satisfying

$$\left(\sum_{i=1}^{d} (v_i - s_i)^2\right)^{1/2} \le 2^{-\eta}p.$$

Since $\mathbf{v} \in \mathcal{L}_{\tau,\mathbf{e},p}(\mathbf{t}_{1,1}, \ldots, \mathbf{t}_{d,k})$, there are some integers $\beta_1, \ldots, \beta_w, z_1, \ldots, z_d$ such that

$$\begin{aligned}\mathbf{v} = \Bigg(\sum_{j=1}^{w} \beta_j(\mathbf{t}_{1,1}^{e_j} + \cdots + \mathbf{t}_{1,k}^{e_j}) - z_1 p, \ldots, \sum_{j=1}^{w} \beta_j(\mathbf{t}_{d,1}^{e_j} + \cdots + \mathbf{t}_{d,k}^{e_j}) - z_d p, \\ \beta_1/2^{\tau+1}, \ldots, \beta_w/2^{\tau+1}\Bigg).\end{aligned}$$

If $\beta_j \equiv \alpha_j \bmod p$, $j = 1, \ldots, w$, then for all $i = 1, \ldots, d$ we have

$$\sum_{j=1}^{w} \beta_j(\mathbf{t}_{i,1}^{e_j} + \cdots + \mathbf{t}_{i,k}^{e_j}) - z_i p = \left\lfloor \sum_{j=1}^{w} \beta_j(\mathbf{t}_{i,1}^{e_j} + \cdots + \mathbf{t}_{i,k}^{e_j}) \right\rfloor_p = \lfloor f(\mathbf{t}_{i,1}) + \cdots + f(\mathbf{t}_{i,k}) \rfloor_p,$$

since otherwise there is $i \in [1, d]$ such that $|v_i - s_i| > 2^{-\eta} p$.

Now suppose that $\beta_j \not\equiv \alpha_j \bmod p$ for some $j \in [1, w]$. In this case we have

$$\begin{aligned}\left(\sum_{i=1}^{d} (v_i - s_i)^2\right)^{1/2} &\geq \max_{i \in [1,d]} \operatorname{dist}_{\mathrm{p}} \left(\sum_{j=1}^{w} \beta_j(\mathbf{t}_{i,1}^{e_j} + \cdots + \mathbf{t}_{i,k}^{e_j}), s_i\right) \\ &\geq \max_{i \in [1,d]} \Bigg(\operatorname{dist}_{\mathrm{p}} \left(f(\mathbf{t}_{i,1}) + \cdots + f(\mathbf{t}_{i,k}), \sum_{j=1}^{w} \beta_j(\mathbf{t}_{i,1}^{e_j} + \cdots + \mathbf{t}_{i,k}^{e_j})\right) \\ &\qquad - \operatorname{dist}_{\mathrm{p}} \left(s_i, f(\mathbf{t}_{i,1}) + \cdots + f(\mathbf{t}_{i,k})\right) \Bigg) \\ &> 2^{-\eta+1} p - 2^{-\eta} p = 2^{-\eta} p,\end{aligned}$$

contradicting our assumption. As we have seen, condition (5) holds with probability exceeding $1 - 2^{-\eta}$ and the result follows. □

Now we state a rather general result.

Theorem 1. *Let p be a sufficiently large n-bit prime and w and k be positive integers of polynomial size. We define*

$$\mu = (nw)^{1/2}, \quad \tau = \lceil \mu + \log n + \log k \rceil, \quad d = 2 \lceil \mu \rceil, \quad \text{and} \quad \eta = 0.5\mu + 3.$$

There exists a deterministic polynomial time algorithm $\mathcal{A}$ such that for any polynomial f, defined by (1), such that $\mathcal{S} \subseteq \mathbb{F}_p^m$ is $(2^{-\eta}, k, f)$-homogeneously distributed modulo p, given $kd(m+1)$ integers

$$\mathbf{t}_{i,j} \in \mathcal{S} \quad \text{and} \quad s_{i,j} = \mathrm{MSB}_{\tau,p}(f(\mathbf{t}_{i,j})), \qquad i = 1, \ldots, d, \quad j = 1, \ldots, k$$

its output satisfies

$$\Pr_{\mathbf{t}_{1,1}, \ldots, \mathbf{t}_{d,k} \in \mathbb{F}_p^m} \left[\mathcal{A}(\mathbf{t}_{1,1}, \ldots, \mathbf{t}_{d,k}; s_{1,1}, \ldots, s_{d,k}) = (\alpha_1, \ldots, \alpha_w)\right] \geq 1 + O(2^{-\eta}),$$

if $\mathbf{t}_{1,1}, \ldots, \mathbf{t}_{d,k}$ are chosen uniformly and independently at random from $\mathcal{S}$.

Proof. Let us consider the vector $\mathbf{s} = (s_1, \ldots, s_d, s_{d+1}, \ldots, s_{d+w})$ where $s_{d+j} = 0$, $j = 1, \ldots, w$, and $s_i = s_{i,1} + \cdots + s_{i,k}$, $i = 1, \ldots, d$. We have

$$|f(\mathbf{t}_{i,j}) - s_{i,j}| \leq p/2^{\tau+1}$$

and thus

$$|f(\mathbf{t}_{i,1}) + \cdots + f(\mathbf{t}_{i,k}) - s_i| \leq kp/2^{\tau+1}.$$

Next we have

$$f(\mathbf{t}_{i,1}) + \cdots + f(\mathbf{t}_{i,k}) - s_i = \lfloor f(\mathbf{t}_{i,1}) + \cdots + f(\mathbf{t}_{i,k}) \rfloor_p - \lfloor s_i \rfloor_p + \nu p$$

with $\nu \in \{-1, 0, 1\}$. If $\nu = 1$ then we have

$$\lfloor f(\mathbf{t}_{i,1}) + \cdots + f(\mathbf{t}_{i,k}) \rfloor_p - \lfloor s_i \rfloor_p + p = |f(\mathbf{t}_{i,1}) + \cdots + f(\mathbf{t}_{i,k}) - s_i| < kp/2^{\tau+1}$$

which is only possible if $\lfloor f(\mathbf{t}_{i,1}) + \cdots + f(\mathbf{t}_{i,k}) \rfloor_p < kp/2^{\tau+1}$. Similarly, we can show that $\nu = -1$ is only possible if $\lfloor f(\mathbf{t}_{i,1}) + \cdots + f(\mathbf{t}_{i,k}) \rfloor_p > p - kp/2^{\tau+1}$. Because of the homogeneous distribution property, the probability that $kp/2^{\tau+1} \le \lfloor f(\mathbf{t}_{i,1}) + \cdots + f(\mathbf{t}_{i,k}) \rfloor_p \le p - kp/2^{\tau+1}$ for all $i = 1, \ldots, d$ is $1 + O(dk/2^{\tau})$. Now we assume that $\nu = 0$ and thus

$$|\lfloor f(\mathbf{t}_{i,1}) + \cdots + f(\mathbf{t}_{i,k}) \rfloor_p - \lfloor s_i \rfloor_p| < kp/2^{\tau+1}.$$

Multiplying the jth row vector of the matrix (4) by α_j and subtracting a certain multiple of the $(w+j)$th vector, $j = 1, \ldots, w$, we obtain a lattice point $\mathbf{u} = (u_1, \ldots, u_d, \alpha_1/2^{\tau+1}, \ldots, \alpha_w/2^{\tau+1}) \in \mathcal{L}_{\tau,\mathbf{e},p}(\mathbf{t}_{1,1}, \ldots, \mathbf{t}_{d,k})$ such that

$$|u_i - s_i| < kp2^{-\tau-1}, \qquad i = 1, \ldots, d+w,$$

where $u_{d+j} = \alpha_j/2^{\tau+1}$, $j = 1, \ldots, w$. Therefore,

$$\sum_{i=1}^{d+w} (u_i - s_i)^2 \le (d+w)2^{-2\tau-2}k^2p^2.$$

We can assume that $w \le n$ because in the opposite case $\tau > n + \log k$ and the result is trivial. Therefore $d+w = O(\tau)$. Now we can use Lemma 4 to find in polynomial time a lattice vector $\mathbf{v} = (v_1, \ldots, v_d, v_{d+1}, \ldots, v_{d+w}) \in \mathcal{L}_{\tau,\mathbf{e},p}(t_{1,1}, \ldots, t_{d,k})$ such that

$$\begin{aligned}\sum_{i=1}^{d} (v_i - s_i)^2 &\le 2^{o(d+w)} \min\left\{ \sum_{i=1}^{d+w} (z_i - s_i)^2, \quad \mathbf{z} = (z_1, \ldots, z_{d+w}) \in \mathcal{L} \right\} \\ &\le 2^{-2\tau+o(\tau)}(d+w)k^2p^2 \le 2^{-2\tau+o(\tau)}k^2p^2 \le 2^{-2\eta-1}p^2,\end{aligned}$$

provided that p is large enough. We also have

$$\sum_{i=1}^{d} (u_i - s_i)^2 \le d2^{-2\tau-2}k^2p^2 \le 2^{-2\eta-2}p^2.$$

Therefore, $\sum_{i=1}^{d} (u_i - v_i)^2 \le 2^{-2\eta}p^2$. Applying Lemma 5, we see that $\mathbf{v} = \mathbf{u}_f$ with probability at least $1 - 2^{-\eta}$, and therefore the coefficients of f can be recovered in polynomial time. □

Applying Lemma 3 and Equation (3) to Theorem 1 we obtain ($k = 1$):

Corollary 1. *Let p be a sufficiently large n-bit prime and let $w \geq 1$ be an integer. We define $\mu = (nw)^{1/2}$, $\tau = \lceil \mu + \log n \rceil$, $d = 2\lceil \mu \rceil$, $\eta = 0.5\mu + 3$. There exists a deterministic polynomial time algorithm $\mathcal{A}$ such that for any polynomial f defined by (1), with known exponents $1 \leq e_1 < \cdots < e_w < p^m$ and total degree*

$$D \leq \frac{p^{1/2}}{2^\eta \log p};$$

given $d(m+1)$ integers $\mathbf{t}_i \in \mathbb{F}_p^m$ and $s_i = \mathrm{MSB}_{\tau,p}(f(\mathbf{t}_i))$, $i = 1, \ldots, d$, its output satisfies

$$\Pr_{\mathbf{t}_1,\ldots,\mathbf{t}_d \in \mathbb{F}_p^m} \left[\mathcal{A}(\mathbf{t}_1, \ldots, \mathbf{t}_d; s_1, \ldots, s_d) = (\alpha_1, \ldots, \alpha_w)\right] \geq 1 + O(2^{-\eta}),$$

if $\mathbf{t}_1, \ldots, \mathbf{t}_d$ are chosen uniformly and independently at random from $\mathbb{F}_p^m$.

For polynomials of higher degree, Lemma 2 gives:

Corollary 2. *Let p be a sufficiently large n-bit prime and let $w \geq 1$ be an integer. We define $\mu = (nw)^{1/2}$, $\tau = \lceil \mu + \log n + \log k \rceil$, $d = 2\lceil \mu \rceil$, $\eta = 0.5\mu + 3$, where $k = \left\lceil 3(w \log p)^{1+\varepsilon} \log p \right\rceil$ for $1 > \varepsilon > 0$. There exists a deterministic polynomial time algorithm $\mathcal{A}$ such that for any polynomial f defined by (1), with known exponents $1 \leq e_1 < \cdots < e_w < p^m$ and satisfying (2), given $kd(m+1)$ integers $\mathbf{t}_{i,j} \in \mathbb{F}_p^m$ and $s_{i,j} = \mathrm{MSB}_{\tau,p}(f(\mathbf{t}_{i,j}))$, $i = 1, \ldots, d$, $j = 1, \ldots, k$, its output satisfies*

$$\Pr_{\mathbf{t}_{1,1},\ldots,\mathbf{t}_{d,k} \in \mathbb{F}_p^m} \left[\mathcal{A}(\mathbf{t}_{1,1}, \ldots, \mathbf{t}_{d,k}; s_{1,1}, \ldots, s_{d,k}) = (\alpha_1, \ldots, \alpha_w)\right] \geq 1 + O(2^{-\eta}),$$

if $\mathbf{t}_{1,1}, \ldots, \mathbf{t}_{d,k}$ are chosen uniformly and independently at random from $\mathbb{F}_p^m$.

Remarks. Using the standard method for reducing incomplete exponential sums to complete ones we can easily extend Corollary 1 to subboxes of $\mathbb{F}_p^m$ of the form $\mathcal{S} = \{\mathbf{x}_0 + (x_1, \ldots, x_m) : 0 \leq x_i < K_i,\ i = 1, \ldots, m\}$ for some integers $1 \leq K_1, \ldots, K_m \leq p$ and $\mathbf{x}_0 \in \mathbb{F}_p^m$. Moreover, we can extend the results to the situation where the randomly chosen points are chosen from a sufficiently large subgroup of $\mathbb{F}_p^*$ of order T by substituting the variable X by $Y^{(p-1)/T}$.

The condition $e_1 \geq 1$ is not absolutely necessary. However, if we don't restrict ourselves to the case $f(0, \ldots, 0) = 0$ a modified algorithm gives only an approximation of the constant term $f(0, \ldots, 0)$.

References

1. Cochrane, T., Pinner, C., Rosenhouse, J.: Bounds on exponential sums and the polynomial Waring problem mod p. J. London Math. Soc. 67(2), 319–336 (2003)
2. Shparlinski, I., Winterhof, A.: A hidden number problem in small subgroups. Math. Comp. 74(252), 2073–2080 (2005)
3. Shparlinski, I., Winterhof, A.: Noisy interpolation of sparse polynomials in finite fields. Appl. Algebra Engrg. Comm. Comput. 16(5), 307–317 (2005)

4. Shparlinski, I.E.: Sparse polynomial approximation in finite fields. In: Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing, pp. 209–215. ACM, New York (2001) (electronic)
5. Shparlinski, I.E.: Playing 'hide-and-seek' with numbers: the hidden number problem, lattices and exponential sums. In: Public-key cryptography. Proc. Sympos. Appl. Math., vol. 62, pp. 153–177. Amer. Math. Soc., Providence (2005)
6. Shparlinski, I.E., Winterhof, A.: A nonuniform algorithm for the hidden number problem in subgroups. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 416–424. Springer, Heidelberg (2004)
7. Winterhof, A.: On Waring's problem in finite fields. Acta Arith. 87(2), 171–177 (1998)

New Commutative Semifields and Their Nuclei

Jürgen Bierbrauer

Department of Mathematical Sciences
Michigan Technological University
Houghton, Michigan 49931 (USA)

Abstract. Commutative semifields in odd characteristic can be equivalently described by planar functions (also known as PN functions). We describe a method to construct a semifield which is canonically associated to a planar function and use it to derive information on the nuclei directly from the planar function. This is used to determine the nuclei of families of new commutative semifields of dimensions 9 and 12 in arbitrary odd characteristic.

Keywords: PN functions, planar functions, presemifields, semifields, middle nucleus, kernel, Dembowski-Ostrom polynomial, isotopy, strong isotopy.

1 Introduction

Until recently the only known families of commutative semifields in arbitrary odd characteristic aside of the fields themselves were the classical constructions by Dickson [7] and Albert [1]. The first provably new such general constructions were given in Zha-Kyureghyan-Wang [11] and [2]. The families constructed in Budaghyan-Helleseth [3] may be new as well but this seems to remain unproved. The survey article of Kantor [9] gives more background information and comments on the scarcity of known commutative semifields in odd characteristic, in particular when the characteristic is > 3.

Definition 1. *A* **presemifield** *is a set F with two binary relations, addition and $*$, such that*

- *F is a commutative group with respect to addition, with identity* 0.
- *F^* is a loop under multiplication.*
- *$0 * a = 0$ for all a.*
- *The distributive laws hold.*

If moreover there is an element $e \in F$ such that $e * x = x * e = x$ for all x we speak of a **semifield.**

Definition 2. *Let $F = \mathbb{F}_{p^r}$ for an odd prime p. A function $f : F \longrightarrow F$ is* **perfectly nonlinear** *(PN), also called a* **planar function,** *if for each $0 \neq a \in F$ the directional derivative δ_a defined as $\delta_a(x) = f(x + a) - f(x)$ is bijective.*

M. Bras-Amorós and T. Høholdt (Eds.): AAECC 2009, LNCS 5527, pp. 179–185, 2009.

Let $f : F \longrightarrow F$ and write it as a polynomial $f(x) = \sum_{i=0}^{r-1} a_i x^i$. Then f is a **Dembowski-Ostrom** *(DO-)polynomial if all its monomials have p-weight ≤ 2 (the exponents are sums of two powers of p).*

In odd characteristic planar DO-polynomials are equivalent with commutative presemifields, see Coulter-Henderson [4]:

Theorem 1. *The following concepts are equivalent:*

- *Commutative presemifields in odd characteristic.*
- *Dembowski-Ostrom polynomials which are PN functions.*

The relation between those concepts is identical to the equivalence between quadratic forms and bilinear forms in odd characteristic, with the planar function in the role of the quadratic form. If $*$ is the presemifield product, then the corresponding planar function is $f(x) = x * x$. When the planar function is given, the corresponding semifield product is

$$x * y = (1/2)\{f(x+y) - f(x) - f(y)\}. \tag{1}$$

For an overview see also Section 9.3.2 of Horadam [8]. One way to construct a semifield from a commutative presemifield is the following: choose $0 \neq e \in F$ and define the new multiplication $\circ$ by

$$(x * e) \circ (y * e) = x * y.$$

Then $\circ$ describes a semifield with unit element $e * e$.

Definition 3. *Let $F = \mathbb{F}_p^r$ be the r-dimensional vector space over $\mathbb{F}_p$. Consider presemifields on F whose additions coincide with that of F. Two such presemifield multiplications $*$ and $\circ$ on F are* **isotopic** *if there exist $\alpha_1, \alpha_2, \beta \in GL(r,p)$ such that*

$$\beta(x \circ y) = \alpha_1(x) * \alpha_2(y)$$

always holds. They are **strongly isotopic** *if we can choose $\alpha_2 = \alpha_1$.*

This notion of equivalence is motivated by the fact that two presemifields are isotopic if and only if the corresponding projective planes are isomorphic. Let $F = \mathbb{F}_{p^r}$ be the field of order p^r. It is a commonly used method to replace a given commutative semifield of order p^r by an isotopic copy which is defined on F and shares the additive structure and the unit element 1 with F. The question is then to which degree the semifield structure can be made to coincide with the field structure. As associativity is the only field axiom that a commutative semifield does not satisfy it is natural that associativity will be in the center of interest.

Definition 4. *Let $F = \mathbb{F}_{p^r}$ and $(F, *)$ a commutative semifield with unit 1 whose additive structure agrees with that of the field F. Define*

$$\mathcal{S} = \{c \in F | c * x = cx \text{ for all } x \in F\}.$$

$$\mathcal{M} = \{c \in F | (x * c) * y = x * (c * y) \text{ for all } x, y \in F\}.$$

$$\mathcal{K} = \{c \in F | c * (x * y) = (c * x) * y \text{ for all } x, y \in F\}.$$

Here the dimensions of the **middle nucleus** $\mathcal{M}$ and of the **kernel** or **left nucleus** $\mathcal{K}$ of a commutative semifield are invariant under isotopy. The dimension of $\mathcal{S}$ depends on the embedding of the semifield in the field F. As mentioned in [5] we have $\mathcal{K} \subseteq \mathcal{M}$ (if $a*(x*y) = (a*x)*y$ for all x, y, then this also equals $(a*y)*x = (y*a)*x = (x*a)*y = x*(a*y)$). As $\mathcal{M}$ is closed under semifield multiplication and is associative it is a field. The semifield multiplication on $\mathcal{M}$ can therefore be made to coincide with field multiplication. The same is true of the vector space structure of F over its subfield $\mathcal{M}$. It follows that we can find a suitable isotope such that

$$\mathcal{K} \subseteq \mathcal{M} \subseteq \mathcal{S}.$$

Here are the constructions from [11] and [2]:

Theorem 2. *Let p be an odd prime, $q = p^s, q' = p^t, F = \mathbb{F}_{q^3}, s' = s/\gcd(s,t), t' = t/\gcd(s,t), s'$ odd. Let $f : F \longrightarrow F$ be defined by*

$$f(x) = x^{1+q'} - vx^{q^2+q'q} \text{ where } ord(v) = q^2 + q + 1.$$

Then f is a PN function in each of the following cases:

- $s' + t' \equiv 0 \pmod 3$.
- $q \equiv q' \equiv 1 \pmod 3$

Theorem 3. *Let p be an odd prime, $q = p^s, q' = p^t, K = \mathbb{F}_q \subset F = \mathbb{F}_{q^4}$ such that $2s/\gcd(2s,t)$ is odd, $q \equiv q' \equiv 1 \pmod 4$. Let $f : F \longrightarrow F$ be defined by*

$$f(x) = x^{1+q'} - vx^{q^3+q'q} \text{ where } ord(v) = q^3 + q^2 + q + 1.$$

Then f is a PN function.

The first family of Theorem 2 is constructed in [11], the second family of Theorem 2 and Theorem 3 are from [2]. In the generic case the first family of Theorem 2 is new as was shown in [11]. It was proved in [2] that the semifields of order p^{4s} isotopic to the special case $t = 2, s > 1$ odd of Theorem 3 are not isotopic to Dickson or Albert semifields.

Let $f(x)$ be a Dembowski-Ostrom polynomial which is a PN function and $*$ the corresponding presemifield product $(x*y = (1/2)\{f(x+y) - f(x) - f(y)\})$. In the following section we decribe a canonical construction of a (commutative) semifield strongly isotopic to $(F, *)$ which allows to read off information on the nuclei directly from $f(x)$. In the last section we continue studying low-dimensional subfamilies of the planar functions of Theorems 2,3. In particular we describe 12-dimensional semifields with middle nucleus of dimension 2 and kernel $\mathbb{F}_p$ as well as a new family of 9-dimensional semifields all of whose nuclei agree with the prime field. In the sequel p always denotes an odd prime.

2 From Commutative Presemifields to Semifields in Odd Characteristic

Definition 5. *Let $f(X)$ be a DO-polynomial defined on $F = \mathbb{F}_{p^r}$ for odd p. Let $G = Gal(F|\mathbb{F}_p) = \{g_0 = id, g_1, \ldots, g_{r-1}\}$ be the Galois group where $g_i(x) = x^{p^i}$. Write*

$$f(X) = \sum_{i=0}^{r-1} a_i g_i(X^2) + \sum_{j<k} b_{jk} g_j(X) g_k(X)$$

where $a_i, b_{jk} \in F$. If $f(X)$ is also a planar function, then the presemifield product defined by $f(X)$ is

$$x * y = \sum_i a_i g_i(xy) + \sum_{j<k} (b_{jk}/2)(g_j(x)g_k(y) + g_k(x)g_j(y))$$

Let $t_i(X) = X^{p^i} - X$.

Lemma 1. *$t_{mu}(X)$ is a polynomial in $t_m(X)$.*

Proof. Let $Q = p^m$. Then $t_{mu}(X) = t_m(X)^{Q^{u-1}} + t_m(X)^{Q^{u-2}} + \ldots + t_m(X) = g_{(u-1)m}(t_m(X)) + \ldots + t_m(X)$.

Proposition 1. *Let p odd, $F = \mathbb{F}_{p^r}$ and $(F, *)$ a commutative presemifield. Let $\alpha \in GL(r, p)$ and define a product $\circ$ by*

$$\alpha(1) * \alpha(x \circ y) = \alpha(x) * \alpha(y).$$

Then $(F, \circ)$ is a commutative semifield with unit 1. It is strongly isotopic to $(F, *)$.

Proof. Obviously $(F, \circ)$ is a commutative presemifield. It is related to $(F, *)$ by the strong isotopy $\beta(x \circ y) = \alpha(x) * \alpha(y)$ where $\beta(x) = \alpha(1) * \alpha(x)$. Choosing $y = 1$ shows $\alpha(1) * \alpha(x \circ 1) = \alpha(x) * \alpha(1)$. It follows $x \circ 1 = x$.

In case $\alpha = id$ we obtain $1 * (x \circ y) = x * y$. This is made explicit in the following definition and theorem.

Definition 6. *Let $F = \mathbb{F}_{p^r}$ for odd p and $f(x)$ a planar DO-polynomial on F. The* **associated semifield function** *is $B(f(x))$ where $B \in GL(r, p)$ is the inverse of $A(x) = x * 1$. The* **associated semifield product** *is the product $\circ$ defined by $B(f(x))$.*

Theorem 4. *Let $f(X) = \sum_{i=0}^{r-1} a_i g_i(X^2) + \sum_{j<k} b_{jk} g_j(X) g_k(X)$ be a planar function on $F = \mathbb{F}_{p^r}$ for odd p, with presemifield product $*$ and associated semifield product $\circ$ (see Definition 6). Let m be the greatest common divisor of r and the numbers $k - j$ where $j < k$ is such that $b_{jk} \neq 0$. Then $\mathbb{F}_{p^m} \subseteq \mathcal{M}(F, \circ) \cap \mathcal{S}(F, \circ)$.*

Proof. Let $x * y$ be the presemifield product defined by $f(X)$. We have

$$A(x) = x * 1 = \sum a_i g_i(x) + \sum_{j<k} (b_{jk}/2)(g_j(x) + g_k(x))$$

and

$$f(x) = A(x^2) + \sum_{j<k} (b_{jk}/2)(2g_j(x)g_k(x) - g_j(x^2) - g_k(x^2)).$$

The expression in parenthesis is

$$2g_j(x)g_k(x) - g_j(x^2) - g_k(x^2) = -(g_k(x) - g_j(x))^2 = -g_j(t_{k-j}(x)^2).$$

This yields the associated semifield function

$$B(f(x)) = x^2 - B(\sum_{j<k} (b_{jk}/2) g_j(t_{k-j}(x)^2))$$

and the associated semifield product

$$x \circ y = xy - B(\sum_{j<k} (b_{jk}/2) g_j(t_{k-j}(x) t_{k-j}(y)).$$

This follows from the linearity of Equation 1 and the fact that a term x^2 in the planar function turns into xy in the presemifield product. Observe that $t_{mu}(X)$ is a polynomial in $t_m(X)$ by Lemma 1. Let $c \in \mathbb{F}_{p^m}$. Then $c \circ x = cx$ as $t_m(c) = 0$. This shows $\mathbb{F}_{p^m} \subseteq \mathcal{S}(F, \circ)$. In order to show $\mathbb{F}_{p^m} \subseteq \mathcal{M}(F, \circ)$ it remains to be shown $(cx) \circ y = x \circ (cy)$ for all x, y. This also follows directly from the fact that $t_{k-j}(cx) = ct_{k-j}(x)$ for all k, j such that $b_{jk} \neq 0$.

Theorem 5. *In the situation of Theorem 4 let l be the greatest common divisor of r and the numbers i, j, k where $a_i \neq 0$ and $j < k$ such that $b_{jk} \neq 0$. Then the associated semifield has $\mathbb{F}_{p^l}$ in its left nucleus.*

Proof. Let $c \in \mathbb{F}_{p^l}$. We have to show $(cx) \circ y = c(x * y)$. This follows from the form of $B(f(x))$ as given in the proof of Theorem 4 and the fact that $A(x)$ and its inverse $B(x)$ are linear over $\mathbb{F}_{p^l}$.

3 Some Semifields and Their Nuclei

Theorem 6. *The semifields of order p^{12} associated to the presemifields in case $s = 3, t = 2$ of Theorem 3 have middle nucleus $\mathbb{F}_{p^2}$ and kernel $\mathbb{F}_p$.*

Proof. We have $p \equiv 1 \pmod 4$, $F = \mathbb{F}_{p^{12}}$ and $ord(v) = p^9 + p^6 + p^3 + 1$. The planar function is

$$f(x) = x^{1+p^2} - vx^{p^5+p^9}.$$

It follows from Theorem 4 that the middle nucleus $\mathcal{M}$ of the associated semifield $(F, \circ)$ has even dimension. It was shown in [2] that $dim(\mathcal{M})$ is not a multiple of

6. If $dim(\mathcal{M}) > 2$, then $dim(\mathcal{M}) = 4$. By a result of Menichetti [10] the semifield would be Albert which is not the case as we proved in [2]. It follows $\mathcal{M} = \mathbb{F}_{p^2}$. We have

$$x \circ y = xy - (1/2)B(t_2(x)t_2(y)) + (1/2)B(vg_5(t_4(x)t_4(y)))$$

(see the proof of Theorem 4) and $t_4(X) = t_2(X) + g_2(t_2(X))$. Let $K(X,Y)$ be the polynomial such that $x \circ y = xy + K(t_2(x), t_2(y))$. Then

$$K(X,Y) = -(1/2)B(XY - vg_5((X + X^{p^2})(Y + Y^{p^2})) =$$

$$= -(1/2)B(XY - v(XY)^{p^5} - v(XY)^{p^7} - vX^{p^5}Y^{p^7} - vX^{p^7}Y^{p^5}).$$

Assume $dim(\mathcal{K}) > 1$. Then $\mathcal{K} = \mathcal{M} = \mathbb{F}_{p^2}$. It has been proven in [5], Theorem 4.2, that this is equivalent with $K(X,Y)$ being a polynomial in X^{p^2} and Y^{p^2}. Although we do not know B explicitly it is obvious that this condition cannot be satisfied. In fact, let $B(x) = \sum_{i=0}^{11} \beta_i g_i(x)$. The absence of monomials $X^{p^i}Y^{p^{i+2}}$ for odd i shows $\beta_0 = \beta_2 = \ldots = \beta_{10} = 0$. As $(XY)^{p^i}$ is absent for odd i we have $0 = \beta_i - g_{i-5}(v)\beta_{i-5} - g_{i-7}(v)\beta_{i-7} = \beta_i$. This yields the contradiction $B \equiv 0$.

We turn to Theorem 2. The smallest dimension for which new planar functions may result is $r = 9$ for the second subfamily. Here t should not be a multiple of 3 as otherwise a field or an Albert twisted field is obtained. Up to obvious isotopy equivalences there are three cases, $f(x) = x^{1+p} - vx^{p^4+p^6}$, $f(x) = x^{1+p} - vx^{p^3+p^7}$, $f(x) = x^{1+p^2} - vx^{p^3+p^8}$. We show that those yield new semifields all of whose nuclei agree with the prime field:

Theorem 7. *Let $p \equiv 1 \pmod 3, q = p^3, K = \mathbb{F}_q \subset F = \mathbb{F}_{p^9}$ and $ord(v) = q^2 + q + 1$. The semifields of order p^9 associated to the planar functions*

$$f(x) = x^{1+p} - vx^{p^4+p^6},\ f(x) = x^{1+p} - vx^{p^3+p^7} \text{ or } f(x) = x^{1+p^2} - vx^{p^3+p^8}$$

have middle nucleus $\mathbb{F}_p$ and are not isotopic to a commutative Albert semifield.

Proof. Assume the middle nucleus $\mathcal{M}$ of a corresponding semifield has dimension > 1. Then the dimension is 3. By Menichetti [10] we are in the Albert case. It suffices therefore to prove that our presemifield is not isotopic to a commutative Albert presemifield. There are four cases to consider. Corresponding presemifields are described by the monomial planar functions X^{1+p^s} where $s \in \{1,2,3,4\}$. Here case $s = 3$ corresponds to the uniquely determined Albert semifield with nucleus of dimension 3, the remaining values of s are representatives of the three isotopism classes of commutative Albert presemifields whose corresponding semifields have nucleus of dimension 1. Assume we have isotopy with one of those commutative Albert presemifields. It follows from Coulter-Henderson [4], Corollary 2.8 that there is a strong isotopy. There exist invertible linear mappings

$$\alpha(x) = \sum_{i=0}^{8} a_i g_i(x),\ \beta(x) = \sum_{i=0}^{8} b_i g_i(x)$$

such that

$$\alpha(x)^{1+p^s} = \beta(f(x)).$$

We complete the proof for the first type $f(x)$. The proofs in the remaining cases are analogous. Observe that in the exponents modular distances (in the circle of length 9) $d = 0, d = 3, d = 4$ do not occur. In the case of distance $d = 0$ this yields $a_i a_{i+s} = 0$ for all i. The equations for $d = 3$ and $d = 4$ are the following:

$$a_i g_s(a_{i+3-s}) + a_{i+3} g_s(a_{i-s}) = 0.$$

$$a_i g_s(a_{i+4-s}) + a_{i+4} g_s(a_{i-s}) = 0.$$

Without restriction $a_0 \neq 0$. It follows $a_s = a_{-s} = 0$. Evaluating the $d = 3$ equation for $i \in \{0, -3, s, s-3\}$ and the $d = 4$ equation for $i \in \{0, -4, s, s-4\}$ shows $a_i = 0$ for $i \in \pm\{s, s-3, s-4, s+3, s+4\}$. For $s = 3$ or $s = 4$ this yields the contradiction $a_0 = 0$. For $s = 1$ or $s = 2$ the contradiction $a_i = 0$ for all $i \neq 0$ is obtained.

References

1. Albert, A.A.: On nonassociative division algebras. Transactions of the American Mathematical Society 72, 296–309 (1952)
2. Bierbrauer, J.: New semifields, PN and APN functions. Designs, Codes and Cryptography (submitted)
3. Budaghyan, L., Helleseth, T.: New perfect nonlinear multinomials over $\mathbb{F}_{p^{2k}}$ for any odd prime p,. In: Golomb, S.W., Parker, M.G., Pott, A., Winterhof, A. (eds.) SETA 2008. LNCS, vol. 5203, pp. 403–414. Springer, Heidelberg (2008)
4. Coulter, R.S., Henderson, M.: Commutative presemifields and semifields. Advances in Mathematics 217, 282–304 (2008)
5. Coulter, R.S., Henderson, M., Kosick, P.: Planar polynomials for commutative semifields with specified nuclei. Designs, Codes and Cryptography 44, 275–286 (2007)
6. Coulter, R.S., Matthews, R.W.: Planar functions and planes of Lenz-Barlotti class II. Designs, Codes and Cryptography 10, 167–184 (1997)
7. Dickson, L.E.: On commutative linear algebras in which division is always uniquely possible. Transactions of the American Mathematical Society 7, 514–522 (1906)
8. Horadam, K.J.: Hadamard matrices and their applications. Princeton University Press, Princeton (2007)
9. Kantor, W.M.: Commutative semifields and symplectic spreads. Journal of Algebra 270, 96–114 (2003)
10. Menichetti, G.: On a Kaplansky conjecture concerning three-dimensional division algebras over a finite field. Journal of Algebra 47, 400–410 (1977)
11. Zha, Z., Kyureghyan, G.M., Wang, X.: Perfect nonlinear binomials and their semifields. Finite Fields and Their Applications 15, 125–133 (2009)

Spreads in Projective Hjelmslev Geometries

Ivan Landjev

New Bulgarian University, 21 Montevideo str.,
1618 Sofia, Bulgaria
and
Institute of Mathematics and Informatics, BAS,
8 Acad. G. Bonchev str., 1113, Sofia, Bulgaria

Abstract. We prove a necessary and sufficient condition for the existence of spreads in the projective Hjelmslev geometries $\mathrm{PHG}(R_R^{n+1})$. Further, we give a construction of projective Hjelmslev planes from spreads that generalizes the familiar construction of projective planes from spreads in $\mathrm{PG}(n,q)$.

Keywords: Projective Hjelmslev geometry, projective Hjelmslev plane, spread, finite chain ring.

Subject Classifications: 51E26, 51E21, 51E22, 94B05.

1 Introduction

In this paper, we introduce spreads in the projective Hjelmslev geometries $\mathrm{PHG}(R_R^{n+1})$. There exists an extensive literature about spreads in the projective geometries $\mathrm{PG}(k,q)$ (cf. [5] and the references there). The same objects in ring geometries have attracted little or no attention despite the connections with interesting areas as linear codes over finite chain rings.

In what follows, we restrict ourselves mainly to spreads in geometries over chain rings of nilpotency index 2. This is a necessary step towards investigating geometries over chain rings of larger nilpotency index, because of the nested structure of the projective Hjelmslev geometries. On the other hand, geometries over rings have less regularities than the usual projective geometries, we settle for a problem that is tractable to some extent. Finally, there exists a complete classification for the chain rings R with $|R| = q^2$, $R/\mathrm{rad}R \cong \mathbb{F}_q$, so that in this case we have a description of all coordinate geometries.

This paper is organized as follows. In Section 2, we give some basic facts about finite chain rings and the structure of projective Hjelmslev geometries over such rings. In Section 3, we prove a necessary and sufficient condition for the existence of spreads in the projective Hjelmslev geometries $\mathrm{PHG}(R_R^{n+1})$, where R is a finite chain ring of nilpotency index 2. We sketch the proof of the main theorem for arbitrary chain rings. In Section 4, we present a construction of projective Hjelmslev planes from spreads in $\mathrm{PHG}(R_R^{n+1})$.

M. Bras-Amorós and T. Høholdt (Eds.): AAECC 2009, LNCS 5527, pp. 186–194, 2009.

2 Basic Facts on Projective Hjelmslev Geometries

A finite ring R (associative, with identity $1 \neq 0$, ring homomorphisms preserving the identity) is called a left (resp. right) chain ring if the lattice of its left (resp. right) ideals forms a chain. It turns out that every left ideal is also a right ideal. Moreover, if $N = \operatorname{rad} R$ every proper ideal of R has the form $N^i = R\theta^i = \theta^i R$, for any $\theta \in N \setminus N^2$ and some positive integer i. The factors N^i/N^{i+1} are one-dimensional linear spaces over R/N. Hence, if $R/N \cong \mathbb{F}_q$ and m denotes the nilpotency index of N, the number of elements of R is equal to q^m. For further facts about chain rings, we refer to [2,10,11].

As mentioned above, we consider chain rings of nilpotency index 2, i.e. chain rings with $N \neq (0)$ and $N^2 = (0)$. Thus we have always $|R| = q^2$, where $R/N \cong \mathbb{F}_q$. Chain rings with this property have been classified in [3,12]. If $q = p^r$ there are exactly $r+1$ isomorphism classes of such rings. These are:

- for every $\sigma \in \operatorname{Aut}\mathbb{F}_q$ the ring $R_\sigma \cong \mathbb{F}_q[X;\sigma]/(X^2)$ of so-called σ-dual numbers over $\mathbb{F}_q$ with underlying set $\mathbb{F}_q \times \mathbb{F}_q$, component-wise addition and multiplication given by $(x_0,x_1)(y_0,y_1) = \big(x_0y_0, x_0y_1 + x_1\sigma(y_0)\big)$;
- the Galois ring $\mathrm{GR}(q^2,p^2) \cong \mathbb{Z}_{p^2}[X]/(f(X))$, where $f(X) \in \mathbb{Z}_{p^2}[X]$ is a monic polynomial of degree r, which is basic irreducible (cf. [10]).

The rings R_σ with $\sigma \neq \mathrm{id}$ are noncommutative, while R_{id} is commutative. Moreover, $\operatorname{char} R_\sigma = p$ for every σ. The Galois ring $\mathrm{GR}(q^2,p^2)$ is commutative and has characteristic p^2. From now on we denote by R a finite chain ring of nilpotency index 2. The only exception will be Theorem 8, where R is a chain ring of an arbitrary nilpotency index.

Let R be a finite chain ring and consider the module $M = R_R^k$. Denote by M^* the set of all non-torsion vectors of M, i.e. $M^* = M \setminus M\theta$. Define sets $\mathcal{P}$ and $\mathcal{L}$ by

$$\mathcal{P} = \{xR; x \in M^*\},$$
$$\mathcal{L} = \{xR + yR; x,y \in M^*, x,y \text{ linearly independent}\},$$

respectively, and take as incidence relation $I \subseteq \mathcal{P} \times \mathcal{L}$ set-theoretical inclusion. Further, define a neighbour relation $\backsim$ on the sets of points and lines of the incidence structure $(\mathcal{P},\mathcal{L},I)$ as follows:

(N1) the points $X,Y \in \mathcal{P}$ are neighbours (notation $X \backsim Y$) if there exist two different lines incident with both of them;
(N2) the lines $s,t \in \mathcal{L}$ are neighbours (notation $s \backsim t$) if there exist two different points incident with both of them.

The incidence structure $\Pi = (\mathcal{P},\mathcal{L},I)$ with the neighbour relation $\backsim$ is called the $(k-1)$-dimensional (right) projective Hjelmslev geometry over R and is denoted by $\mathrm{PHG}(R_R^k)$.

The point set $\mathcal{S} \subseteq \mathcal{P}$ is called a Hjelmslev subspace (or simply subspace) of $\mathrm{PHG}(R_R^k)$ if for every two points $X,Y \in \mathcal{S}$, there exists a line l incident with X and Y that is incident only with points of $\mathcal{S}$. The Hjelmslev subspaces of $\mathrm{PHG}(R_R^k)$ are of the form $\{xR; x \in U^*\}$, where U is a free submodule of M. The (projective) dimension of a subspace is equal to the rank of the underlying module minus 1.

It is easily checked that $\backsim$ is an equivalence relation on each one of the sets $\mathcal{P}$ and $\mathcal{L}$. If $[X]$ denotes the set of all points that are neighbours to $X = xR$, then $[X]$ consists

of all free rank 1 submodules of $xR+M\theta$. Similarly, the class $[l]$ of all lines which are neighbours to $l=xR+yR$ consists of all free rank 2 submodules of $xR+yR+M\theta$.

More generally, two subspaces $\mathcal{S}$ and $\mathcal{T}$, $\dim\mathcal{S}=s$, $\dim\mathcal{T}=t$, $s\le t$, are neighbours if

$$\{[X];X\in\mathcal{S}\}\subseteq\{[X];X\in\mathcal{T}\}.$$

In particular, we say that the point X is a neighbour of the subspace $\mathcal{S}$ if there exists a point $Y\in\mathcal{S}$ with $X\backsim Y$. The neighbour class $[\mathcal{S}]$ contains all subspaces of dimension s that are neighbours to $\mathcal{S}$.

The next theorems give some insight into the structure of the projective Hjelmslev geometries $\mathrm{PHG}(R_R^k)$ and are part of more general results [1,4,6,7,8,9,13]. As usual, $\begin{bmatrix}n\\k\end{bmatrix}_q$ denotes the Gaussian coefficient:

$$\begin{bmatrix}n\\k\end{bmatrix}_q=\frac{(q^n-1)(q^{n-1}-1)\dots(q^{n-k+1}-1)}{(q^k-1)(q^{k-1}-1)\dots(q-1)}.$$

Theorem 1. *Let $\Pi=\mathrm{PHG}(R_R^k)$ where R is a chain ring with $|R|=q^2$, $R/N\cong\mathbb{F}_q$. Then*

(i) the number of points (hyperplanes) in Π is $q^{k-1}\begin{bmatrix}n\\k\end{bmatrix}_q=q^{k-1}\cdot\frac{q^k-1}{q-1}$;
(ii) every point (hyperplane) has q^{k-1} neighbours;
(iii) every subspace of dimension $s-1$ is contained in exactly $q^{(t-s)(k-t)}\begin{bmatrix}k-s\\t-s\end{bmatrix}_q$ subspaces of dimension $t-1$, where $s\le t\le k$;
(iv) given a point P and a subspace $\mathcal{S}$ of dimension $s-1$ containing P, there exist exactly q^{s-1} points in $\mathcal{S}$ that are neighbours to P.

Note that the Hjelmslev spaces $\mathrm{PHG}(R_R^k)$ are 2-uniform in the sense of [4]. Denote by η the natural homomorphism from R^k to $R^k/R^k\theta$ and by $\overline{\eta}$ the mapping induced by η on the submodules of R^k. It is clear that for every point X and every line l we have

$$[X]=\{Y\in\mathcal{P};\overline{\eta}(Y)=\overline{\eta}(X)\},$$

$$[l]=\{m\in\mathcal{L};\overline{\eta}(m)=\overline{\eta}(l)\}.$$

Let us denote by $\mathcal{P}'$ (resp. $\mathcal{L}'$) the set of all neighbour classes of points (resp. lines). The following result is straightforward.

Theorem 2. *The incidence structure $(\mathcal{P}',\mathcal{L}',I')$ with incidence relation I' defined by*

$$[X]\,I'\,[l]\iff\exists Y\in[X],\exists m\in[l]:\,Y\,I\,m$$

is isomorphic to the projective geometry $\mathrm{PG}(k-1,q)$

Let $\mathcal{S}_0$ be a fixed subspace in $\mathrm{PHG}(R_R^k)$ with $\dim\mathcal{S}_0=s$. Define the set $\mathfrak{P}$ of subsets of $\mathcal{P}$ by

$$\mathfrak{P}=\{\mathcal{S}\cap[X];X\backsim\mathcal{S}_0,\mathcal{S}\in[\mathcal{S}_0]\}.$$

The sets $\mathcal{S}\cap[X]$ are either disjoint or coincide. Define an incidence relation $\mathfrak{I}\subset\mathfrak{P}\times\mathcal{L}$ by

$$(\mathcal{S}\cap[X])\,\mathfrak{I}\,l\iff l\cap(\mathcal{S}\cap[X])\neq\emptyset.$$

Let $\mathcal{L}(\mathcal{S}_0)$ be the set of all lines in $\mathcal{L}$ incident with at least one point in $\mathfrak{P}$. For the lines $l_1, l_2 \in \mathcal{L}(\mathcal{S}_0)$ we write $l_1 \sim l_2$ if they are incident (under $\mathfrak{I}$) with the same elements of $\mathfrak{P}$. The relation $\sim$ is an equivalence relation under which $\mathcal{L}(\mathcal{S}_0)$ splits into classes of equivalent lines. Denote by $\mathfrak{L}$ a set of representatives of the equivalence classes of lines in $\mathcal{L}(\mathcal{S}_0)$. The set of representatives $\mathfrak{L}$ contains only two types of lines: lines l with $l \hookrightarrow \mathcal{S}_0$ and lines l with $l \not\hookrightarrow \mathcal{S}_0$.

Theorem 3. *The incidence structure* $(\mathfrak{P}, \mathfrak{L}, \mathfrak{I}\,|_{\mathfrak{P}\times\mathfrak{L}})$ *can be embedded into* $\mathrm{PG}(k-1,q)$.

A special case of this result is obtained if we take $\mathcal{S}_0$ to be a point. Given $\Pi = (\mathcal{P}, \mathcal{L}, I) = \mathrm{PHG}(R_R^k)$ and a point $P \in \mathcal{P}$, let $\mathcal{L}(P)$ be the set of all lines in $\mathcal{L}$ incident with points in $[P]$. For two lines $s,t \in \mathcal{L}(P)$ we write $s \sim t$ if s and t coincide on $[P]$. Denote by $\mathcal{L}_1$ a complete list of representatives of the lines from $\mathcal{L}(P)$ with respect to the equivalence relation $\sim$. Then we have the following result:

Theorem 4

$$([P], \mathcal{L}_1, I|_{[P]\times\mathcal{L}_1}) \cong \mathrm{AG}(k-1,q).$$

Finally, let two points X_1 and X_2 in $\Pi = \mathrm{PHG}(R_R^k)$ be neighbours. Then any two lines incident with X_1 and X_2 are neighbours and belong to the same class, $[l]$ say. In such case we say that the neighbour class $[l]$ has the direction of the pair (X_1, X_2).

3 The Existence of Spreads in Projective Hjelmslev Geometries

Definition 5. *An r-spread of the projective Hjelmslev geometry* $\mathrm{PHG}(R_R^{n+1})$ *is a set* $\mathcal{S}$ *of r-dimensional subspaces such that every point is contained in exactly one subspace of* $\mathcal{S}$.

Theorem 6. *Let R be a chain ring with* $|R| = q^2$, $R/\operatorname{rad} R \cong \mathbb{F}_q$. *There exists a spread* $\mathcal{S}$ *of r-dimensional spaces of* $\mathrm{PHG}(R_R^n)$ *if and only if* $r+1$ *divides* $n+1$.

Proof. The number of points in an r-dimensional subspace is $q^r \begin{bmatrix} r+1 \\ 1 \end{bmatrix}_q$. The existence of a spread of r dimensional subspaces implies that $q^r \begin{bmatrix} r+1 \\ 1 \end{bmatrix}_q$ divides $q^n \begin{bmatrix} n+1 \\ 1 \end{bmatrix}_q$, i.e. $\begin{bmatrix} r+1 \\ 1 \end{bmatrix}_q$ divides $\begin{bmatrix} n+1 \\ 1 \end{bmatrix}_q$, i.e. $r+1$ divides $n+1$.

Assume that $r+1$ divides $n+1$ and let s be determined by $n+1 = (s+1)(r+1)$. First we consider the case where $R = \mathrm{GR}(q^2, p^2)$. Take an algebra of dimension $r+1$ over $R = \mathrm{GR}(q^2, p^2)$, e.g. let this algebra be $R_{r+1} = R[X]/(f(X)) = \mathrm{GR}(q^{2(r+1)}, p^2)$, where f is a monic irreducible polynomial of degree $r+1$ over R. If α is a root of f in R_{r+1} then every element β from R_{r+1} can be written as

$$\beta = b_0 + b_1\alpha + \ldots + b_r\alpha^r, \;\; b_i \in R.$$

Clearly, R_{r+1}^{s+1}, R_{n+1} and R^{n+1} are isomorphic as modules over R. Thus each point in $\mathrm{PHG}(R_R^{n+1})$ can be represented by an $(s+1)$-tuple of elements from R_{r+1} or as a unit in R_{n+1}. In the same time, every $(s+1)$-tuple over R_{r+1} that has at least one coordinate that is a unit, can be viewed as a point in $\mathrm{PHG}(R_{r+1}^{s+1})$.

Let $(\gamma_0,\gamma_1,\dots,\gamma_s)\in (R_{r+1}^{s+1})^*$ be a nontorsion vector. Without loss of generality, let $\gamma_0\neq 0$. Consider the system

$$\left|\begin{array}{lllll} -\gamma_1 x_0+\gamma_0 x_1 & & & & =0\\ -\gamma_2 x_0+ & & +\gamma_0 x_2 & & =0\\ & & \ddots & & \\ \dots & & & & =0\\ -\gamma_s x_0+ & & & +\gamma_0 x_s & =0 \end{array}\right. . \tag{1}$$

The choice of the nonzero element is not essential. If we take $\gamma_j\neq 0$. The system (1) is equivalent to $-\gamma_i x_j+\gamma_j x_i=0$ for $i=0,1,\dots,s$, $i\neq j$. The solutions of (1) form a free submodule of rank 1 R_{r+1}^{s+1}, i.e. a point in $\mathrm{PHG}(R_{r+1}^{s+1})$. This rank 1 submodule can be considered as a free submodule of rank $(r+1)$ of R_R^{n+1}, i.e. a r-dimensional subspace of $\mathrm{PHG}(R_R^{n+1})$. Two $(r+1)$-dimensional subspaces in R_R^{n+1} obtained from different 1-dimensional subspaces of R_{r+1}^{s+1} do not have a common nontorsion vector.

Now consider two different points $(\gamma_0,\gamma_1,\dots,\gamma_s)$ and $(\delta_0,\delta_1,\dots,\delta_s)$ in $\mathrm{PHG}(R_{r+1}^{s+1})$. These points give rise to systems of the type (1) having as solutions different points of $\mathrm{PHG}(R_{r+1}^{s+1})$ (1-dimensional subspaces of R_{r+1}^{s+1}). Assume otherwise and let the $(s+1)$-tuple $(x_0,x_1,\dots,x_s)\neq(0,0,\dots,0)$ be a common solution of the two systems. Then

$$x_0=\lambda\gamma_0=\mu\delta_0, x_1=\lambda\gamma_1=\mu\delta_1,\dots,x_s=\lambda\gamma_s=\mu\delta_s,$$

where $\lambda,\mu\in R_{r+1}$, $\lambda,\mu\neq 0$. This is a contradiction since the points $(\gamma_0,\gamma_1,\dots,\gamma_s)$ and $(\delta_0,\delta_1,\dots,\delta_s)$ were assumed to be different.

It remains to prove that every point is contained in a r-dimensional subspace. The number of points in $\mathrm{PHG}(R_R^{n+1})$ is $q^n\begin{bmatrix}n+1\\1\end{bmatrix}_q$; the number of points in an r-dimensional subspace is $q^r\begin{bmatrix}r+1\\1\end{bmatrix}_q$ and the number of points in $\mathrm{PHG}(R_{r+1}^{s+1})$ is $q^{s(r+1)}\begin{bmatrix}s+1\\1\end{bmatrix}_{q^{r+1}}$. Now we have

$$q^r\begin{bmatrix}r+1\\1\end{bmatrix}_q\cdot q^{s(r+1)}\begin{bmatrix}s+1\\1\end{bmatrix}_{q^{r+1}}=q^r\frac{q^{r+1}-1}{q-1}\cdot q^{s(r+1)}\frac{q^{(s+1)(r+1)}-1}{q^{r+1}-1}=q^n\begin{bmatrix}n+1\\1\end{bmatrix}_q,$$

which means that the r-dimensional subspaces cover all points of $\mathrm{PHG}(R_R^{n+1})$.

Secondly, consider the case where R is the ring of σ dual numbers over the finite field $\mathbb{F}_q$, i.e. $R=R_\sigma=\mathbb{F}_q+t\mathbb{F}_q$. Denote by R' the ring of σ'-dual numbers $\mathbb{F}_{q^{r+1}}+t\mathbb{F}_{q^{r+1}}$, where $\sigma'|_{\mathbb{F}_{q^{r+1}}}=\sigma$. Similarly, let R'' be the ring of σ''-dual numbers $\mathbb{F}_{q^{n+1}}+t\mathbb{F}_{q^{n+1}}$, where $\sigma''|_{\mathbb{F}_{q^{n+1}}}=\sigma'$. The ring R is a subring of R' which in turn is a subring of R''. As above, $R_R'^{s+1}$ and R_R^{n+1} and R_R'' are isomorphic as (right) submodules over R.

Consider an arbitrary nontorsion vector $(\gamma_0,\gamma_1,\dots,\gamma_s)\in {}_RR'^{s+1}$. Fix a component which is a unit, γ_0 say, and consider the system of linear equations (1). The set of solutions of (1) is a free rank 1 submodule of $R_R'^{s+1}$ which can be viewed as a free rank r submodule of R_R^{n+1}. Further the proof is completed as for Galois rings.

Remark 7. *Assume $r+1$ divides $n+1$. We can prove the existence of a spread of r-dimensional subspaces using the nested structure of the projective Hjelmslev geometries.*

Let $\mathcal{H}_0$ be a fixed subspace in $\mathrm{PHG}(R_R^k)$ *with* $\dim \mathcal{H}_0 = r$. *Define the set $\mathfrak{P}$ of subsets of $\mathcal{P}$ by*

$$\mathfrak{P} = \{\mathcal{H} \cap [X]; X \hookrightarrow \mathcal{H}_0, \mathcal{H} \text{ is a subspace}, \dim \mathcal{H} = s, \mathcal{H} \in [\mathcal{H}_0]\}.$$

The sets $\mathcal{H} \cap [X]$ are either disjoint or coincide. Define an incidence relation $\mathfrak{I} \subset \mathfrak{P} \times \mathcal{L}$ by

$$(\mathcal{H} \cap [X])\,\mathfrak{I}\, l \Longleftrightarrow l \cap (\mathcal{H} \cap [X]) \neq \emptyset.$$

Let $\mathcal{L}(\mathcal{H}_0)$ be the set of all lines in $\mathcal{L}$ incident with at least one point in $\mathfrak{P}$. For the lines $l_1, l_2 \in \mathcal{L}(\mathcal{H}_0)$ we write $l_1 \sim l_2$ if they are incident (under $\mathfrak{I}$) with the same elements of $\mathfrak{P}$. The relation $\sim$ is an equivalence relation under which $\mathcal{L}(\mathcal{H}_0)$ splits into nonintersecting classes of equivalent lines. Denote by $\mathfrak{L}$ a set of representatives of the equivalence classes of lines in $\mathcal{L}(\mathcal{H}_0)$. The set of representatives $\mathfrak{L}$ contains only two types of lines: lines l with $l \hookrightarrow \mathcal{H}_0$ and lines l with $l \not\hookrightarrow \mathcal{H}_0$.

It is known from [6] that the incidence structure $(\mathfrak{P}, \mathfrak{L}, \mathfrak{I}\,|_{\mathfrak{P}\times\mathfrak{L}})$ can be embedded isomorphically into the projective geometry $\mathrm{PG}(k-1, q)$. *Hence we can construct a spread in* $\mathrm{PHG}(R_R^{n+1})$ *in the following way. We start with a spread in the factor geometry $PG(n,q)$. This spread defines a set of neighbourhood classes of projective r-subspaces. Each one of these classes is isomorphic in the sense of the above mentioned result to a projective geometry* $\mathrm{PG}(n,q)$ *with an $(n-r-1)$-dimensional space deleted. Now it suffices to take a spread which contains a spread of the deleted $(n-r-1)$-dimensional subspace.*

A spread with this property can be constructed, for instance, by repeating the construction from the proof of Theorem 6. The exceptional $(n-r-1)$-dimensional space can be taken as the space consisting of all points having zeros in the first $r+1$ positions.

Theorem 6 can be generalized to projective Hjelmslev geometries over arbitrary chain rings R.

Theorem 8. *Let R be a chain ring with $|R| = q^m$, $R/\operatorname{rad} R \cong \mathbb{F}_q$. The n-dimensional projective Hjelmslev geometry* $\mathrm{PHG}(R_R^{n+1})$ *has a spread of r-dimensional projective Hjelmslev subspaces if and only if $r+1$ divides $n+1$.*

Proof. We give only a sketch of a proof. For the sake of convenience, we set $N = \operatorname{rad} R = \theta R$. As before, the "only if"-part is straightforward. The proof of the "if"-part uses induction on m and n. This result is obviously true for $m = 1$ and 2. It is also trivial if $n = r$ for every m.

Consider the factor geometry having as points the $(m-1)$-neighbour classes on points. It is isomorphic to $\mathrm{PHG}((R^k/\theta^{m-i}R^k)_{R/N^{m-i}}$ (cf. [6]). By the induction hypothesis, it has a spread of r-dimensional projective Hjelmslev subspaces. The preimage of these subspaces are of the form $[\Delta]_{m-1}$ where Δ is an r-dimensional Hjelmslev subspace in $\mathrm{PHG}(R_R^{n+1})$. Here $[\Delta]_j$ is the class of all r-dimensional Hjelmslev subspaces that are j-th neighbours to Δ. Now $[\Delta]_j$ can be imbedded isomorphically in $\mathrm{PHG}((R/N)_{R/N}^{n+1}) \cong \mathrm{PG}(n,q)$ (cf. [6]) where the missing part is an $(n-r-1)$-dimensional subspace, $\mathcal{H}$ say. Since $r+1$ divides $n-r = (n+1)-(r+1)$, we have that $\mathcal{H}$ contains a spread by the induction hypothesis. Now it is enough to take a spread which contains as a subset the spread of the missing $(n-r-1)$-dimensional subspace.

4 Projective Hjelmslev Planes from Spreads

Spreads in $\Pi = \mathrm{PHG}(R_R^{n+1})$ can be used to construct projective Hjelmslev planes. Set

$$n = 2t-1,\ r = t-1,\ s = 1.$$

By Theorem 6 , there exists a spread $\mathcal{S}$ of r-dimensional subspaces of Π such that its image under the canonical map η is a (multiple of a) spread in $\mathrm{PG}(n,q)$.

The geometry Π can be imbedded in $\widehat{\Pi} = \mathrm{PHG}(R_R^{n+2})$, e.g. by taking by taking as points of Π all points of $\widehat{\Pi}$ with first coordinate 0. Hence Π can be considered as a hyperplane of $\widehat{\Pi}$. Denote by $[\Pi]$ the set of all neighbour hyperpalnes to Π in $\widehat{\Pi}$. Define a new incidence structure as follows:

Take as points:

(1) all points of $\widehat{\Pi}$ that are not incident with a point of $[\Pi]$. These are called proper points and their number is:

$$q^{n+1}\frac{q^{n+2}-1}{q-1} - q^{n+1}\frac{q^{n+1}-1}{q-1} = q^{2(n+1)} = q^{4t}.$$

(2) all subspaces of the form

$$\langle S,P\rangle \cap H,$$

where S is an r-dimensional subspace from $\mathcal{S}$, P is a point from $\widehat{\Pi}\setminus[\Pi]$ and H is a hyperplane of $\widehat{\Pi}$ contained in the neighbour class $[\Pi]$. We cann call these ideal points. The number of choices for the point P is $q^{4t} = q^{2(n+1)}$. The number of choices for $S \in \mathcal{S}$ is

$$|\mathcal{S}| = \frac{q^n\frac{q^{n+1}-1}{q-1}}{q^r\frac{q^{r+1}-1}{q-1}} = \frac{q^{2t-1}(q^{2t}-1)}{q^{t-1}(q^t-1)} = q^t(q^t+1).$$

The number of choices for $H \in [\Pi]$ is $q^{n+1} = q^{2t}$. For all points Q in $\langle S,P\rangle \setminus [\Pi]$, we have $\langle S,Q\rangle \cap H = \langle S,P\rangle \cap H$ i.e. we get the same point in the new incidence structure. Hence for

$$q^{r+1}\frac{q^{r+2}-1}{q-1} - q^{r+1}\frac{q^{r+1}-1}{q-1} = q^{2(r+1)} = q^{2t}.$$

different points P we get the same $(r+1)$-dimensional subspace $\langle S,P\rangle$.

As lines we take:

(1) all subspaces of the form $\langle S,P\rangle$, where $S \in \mathcal{S}$ and P is a point from $\widehat{\Pi}\setminus[\Pi]$, i.e. these are all $(r+1)$-dimensional subspaces through r-dimensional subspaces in the spread;
(2) all hyperplanes H from $[\Pi]$.

For the proper points neighbourhood is inherited from $\widehat{\Pi}$. For the ideal points, we have that

$$\langle S', P\rangle \cap H' \backsim \langle S'', P\rangle \cap H''$$

if and only if S' and S'' are neighbours in Π. By definition, two lines ℓ_1 and ℓ_2 are neighbours if for every point $X \in \ell_1$ there exists a point $Y \in \ell_2$ with $X \backsim Y$, and, conversely, for every $Y \in \ell_2$ there exists an $X \in \ell_1$ with $Y \backsim X$.

Lemma 9. *Let S be an r-dimensional subspace in Π and let P, Q be points from $\widehat{\Pi} \setminus [\Pi]$ with $P \backsim Q$. Then $\langle S, P\rangle \cap [\Pi] = \langle S, Q\rangle \cap [\Pi]$.*

Proof. Assume there exists a point $X \in [\Pi]$ with $X \in \langle S, Q\rangle$, but $X \notin \langle S, P\rangle$. The lines PY and QY are neighbours. Therefore $|PY \cap QY| = q$. The common points of both lines must be neighbours to Y. Hence the q common points must lie in $[\Pi]$, contradiction to the initial assumption.

Lemma 10. *The number of hyperplanes from $[\Pi]$ through a fixed r-dimensional flat $S \in \mathcal{S}$ ($S \subset \Pi$) is q^t.*

Proof. Any r-dimensional flat in an $(n+1)$-dimensional space can be given by a set of $(n+1) - r = t + 1$ equations. Without loss of generality, let S be given by $x_0 = x_1 = \ldots = x_t = 0$ and let Π be the hyperplane defined by $x_0 = 0$. An arbitrary hyperplane in $[\Pi]$ containing S satisfies an equation of the form:

$$x_0 + \theta(r_1 x_1 + r_2 x_2 + \ldots + r_t x_t) = 0. \tag{2}$$

We have $\theta r = \theta s$ if and only if $r - s \in \operatorname{rad} R$, therefore (2) describes all hyperplanes in $[\Pi]$ through S when $(r_1, r_2, \ldots, r_t)$ runs Γ^t, where Γ is a set of elements no two of which are congruent modulo $\operatorname{rad} R$. hence there are exactly q possibilities for each r_i and the number of hyperplanes in $[\Pi]$ through S is q^t.

According to Lemma 9 the number of the essentially different choices of P is

$$\frac{\frac{q^{n+2}-1}{q-1} - \frac{q^{n+1}-1}{q-1}}{\frac{q^{r+2}-1}{q-1} - \frac{q^{r+1}-1}{q-1}} = \frac{q^{n+1}}{q^{r+1}} = q^t.$$

The number of choices for S is $q^t(q^t+1)$ and the number of hyperplanes H from $[\Pi]$ is q^{2t}. On the other hand, by Lemma 10, we get the same intersection for q^t different hyperplanes in $[\Pi]$. Each ideal point of the second type can be obtained for q^t different subspaces $\langle S, P\rangle$. Therefore the number of all points of the second type is

$$\frac{q^t \cdot q^t(q^t+1) \cdot q^{2t}}{q^t \cdot q^t} = q^{3t} + q^{2t}.$$

Now it is a straightforward check that the defined incidence structure is indeed a projective Hjelmslev plane.

Acknowledgements. This research has been supported by the Strategic Development Fund of The New Bulgarian University.

References

1. Artmann, B.: Hjelmslev-Ebenen mit verfeinerten Nachbarschaftsrelationen. Mathematische Zeitschrift 112, 163–180 (1969)
2. Clark, W.E., Drake, D.A.: Finite chain rings. Abhandlungen aus dem mathematischen Seminar der Universität Hamburg 39, 147–153 (1974)
3. Cronheim, A.: Dual numbers, Witt vectors, and Hjelmslev planes. Geometriae Dedicata 7, 287–302 (1978)
4. Drake, D.A.: On n-Uniform Hjelmslev Planes. Journal of Combinatorial Theory 9, 267–288 (1970)
5. Hirschfeld, J.W.P.: Projective geometries over finite fields. Clarendon Press, Oxford (1998)
6. Honold, T., Landjev, I.: Projective Hjelmslev geometries. In: Proc. of the International Workshop on Optimal Codes, Sozopol, pp. 97–115 (1998)
7. Kreuzer, A.: Hjelmslev-Räume. Resultate der Mathematik 12, 148–156 (1987)
8. Kreuzer, A.: Projektive Hjelmslev-Räume, Dissertation, Technische Universität München (1988)
9. Kreuzer, A.: Hjelmslevsche Inzidenzgeometrie - ein Bericht, Bericht TUM-M9001, Technische Universität München, Beiträge zur Geometrie und Algebra Nr. 17 (January 1990)
10. McDonald, B.R.: Finite rings with identity. Marcel Dekker, New York (1974)
11. Nechaev, A.A.: Finite principal ideal rings. Mat. Sbornik of the Russian Academy of Sciences 20, 364–382 (1973)
12. Raghavendran, R.: Finite associative rings. Compositio Mathematica 21, 195–229 (1969)
13. Veldkamp, F.D.: Geometry over rings. In: Buekenhout, F. (ed.) Handbook of Incidence Geometry—Buildings and Foundations, pp. 1033–1084. Elsevier Science Publishers, Amsterdam (1995)

On the Distribution of Nonlinear Congruential Pseudorandom Numbers of Higher Orders in Residue Rings

Edwin D. El-Mahassni[1] and Domingo Gomez[2,⋆]

[1] Department of Computing, Macquarie University
North Ryde, NSW 2109, Australia
edwinelm@ics.mq.edu.au
[2] Faculty of Science, University of Cantabria
E-39071 Santander, Spain
domingo.gomez@unican.es

Abstract. The nonlinear congruential method is an attractive alternative to the classical linear congruential method for pseudorandom number generation. In this paper we present new discrepancy bounds for sequences of s-tuples of successive nonlinear congruential pseudorandom numbers of higher orders modulo a composite integer M.

1 Background

For an integer $M > 1$, we denote by $\mathbb{Z}_M$ the residue ring modulo M. In this paper we present some distribution properties of a generalization of pseudorandom number generators, first introduced in [7], defined by a recurrence

$$u_{n+1} \equiv f(g_1(u_n, \ldots, u_{n-r+1}), \ldots, g_r(u_n, \ldots, u_{n-r+1})) \pmod{M}, \quad (1)$$

where

$$f(X_1, \ldots, X_r),\ g_1(X_1, \ldots, X_r), \ldots,\ g_r(X_1, \ldots, X_r) \in \mathbb{Z}_M[X_1, \ldots, X_r]$$

for $n \geq r - 1$ with some initial values $u_0, \ldots, u_{r-1}$.

To study this pseudorandom number generator, we define the sequence of polynomials $f_k(\mathbf{X}) \in \mathbb{Z}_M[\mathbf{X}]$, with $\mathbf{X} = X_1, \ldots, X_r$ by the recurrence relation

$$f_k(\mathbf{X}) \equiv f_{k-1}(g_1(\mathbf{X}), \ldots, g_r(\mathbf{X})) \pmod{M}, \qquad k \geq 1 \quad (2)$$

where $f_0(\mathbf{X}) = f(\mathbf{X})$.

It is obvious that (1) becomes periodic with some period $t \leq M^r$. Throughout this paper we assume that this sequence is *purely periodic*, i.e. $u_n = u_{n+t}$ beginning with $n = 0$, otherwise we consider a shift of the original sequence.

⋆ Domingo Gomez is partially supported by the Spanish Ministry of Education and Science grant MTM20067088.

M. Bras-Amorós and T. Høholdt (Eds.): AAECC 2009, LNCS 5527, pp. 195–203, 2009.

Although the distribution of nonlinear congruential generators has been studied extensively, see [5,6,9] for instance, much less is known for its higher orders analogue. A result for a class of polynomials for prime moduli was established in [7], where this was later extended to a larger family of polynomials in [8]. In this paper we show a generalization of this result to a larger class of pseudorandom number generators.

1.1 Notation and First Results

This section begins with some notation. It will be assumed that $N,\ A_i,\ B_i$ and b_i represent integer positive numbers and $\mathbf{0}$ is the r-dimensional 0 vector. The elements of $\mathbb{Z}_M$ will be identified with the integers $\{0, \ldots,\ M-1\}$. For this reason, we can define $\mathbf{e}_M(z) = \exp(2\pi i z/M)$ for any element $z \in \mathbb{Z}_M$.

For a polynomial f, G is the gcd of the coefficients of nonconstant monomials with M.

We will denote $f^{(p)}(\mathbf{X}) \equiv f(\mathbf{X}) \pmod p$ of total degree d' for a polynomial f with integer coefficients and total degree d. Finally, we define $\deg_{X_r} f$, the degree of the coefficient X_r of the polynomial f, to be $\deg_{X_r}$ modulo every prime factor p of M.

Lemma 1. *Given a polynomial in the ring* $\mathbb{Z}_M[\mathbf{X}]$,

$$f(\mathbf{X}) \equiv \sum_{i_1=0}^{d_1} \cdots \sum_{i_r=0}^{d_r} b_{i_1,\ldots,i_r} X_1^{i_1} \ldots X_r^{i_r} \pmod M$$

with $\deg_{X_r} f \geq 1$ *, total degree* d*, then exist polynomials* $h_1(X_r), \ldots,\ h_{r-1}(X_r)$ *of degree less than* $d(\lceil \log d \rceil + 1)$ *such that*

$$g^{(p)}(X_r) = f^{(p)}(a_1 + h_1^{(p)}(X_r), \ldots, a_{r-1} + h_{r-1}^{(p)}(X_r), X_r) \tag{3}$$

is a nonconstant polynomial for any $p|M$, $p \not| G$ *and any values* $a_1, \ldots, a_{r-1} \in \mathbb{Z}_M$.

Proof. Let $p|M$ be a prime number not dividing G and $D = \lceil \log d \rceil + 1$. By the definition of G, we note that $f^{(p)}(\mathbf{X})$ is not a constant polynomial and it can be expressed as $f^{(p)}(\mathbf{X}) = h(X_1, \ldots,\ X_{r-1}) X_r^{d'} + f'(\mathbf{X})$ where $f'(\mathbf{X})$ is a polynomial of degree stricly less than d' in X_r. If $d' = 0$ then $f^{(p)}(\mathbf{X}) = h(X_1, \ldots,\ X_{r-1})$, but in any case h is a not a constant polynomial.

Let $\mathbb{F}$ be an extension field of degree D over $\mathbb{Z}_p$. By the cardinality of $\mathbb{F}$, there exist $\xi_1, \ldots,\ \xi_{r-1} \in \mathbb{F}$ such as $h(\xi_1, \ldots,\ \xi_{r-1}) \neq 0$.

It is easy to check that

$$f^{(p)}(X_1 + \xi_1 X_r, \ldots, X_{r-1} + \xi_{r-1} X_r,\ X_r) = h(\xi_1, \ldots,\ \xi_{r-1}) X_r^{d''} + f''(\mathbf{X}) \tag{4}$$

where $d \geq d'' > 0$ and $f''(\mathbf{X})$ is a polynomial with total degree less than $d'' - 1$ in X_r. Let $\mathbf{E}$ be a extension field of degree $d+1$ over $\mathbb{F}$ and let θ be a defining element of $\mathbf{E}$ over both $\mathbb{Z}_p$ and $\mathbf{E}$, i.e. $\mathbf{E} \equiv \mathbb{Z}_p(\theta) \equiv \mathbb{F}(\theta)$.

The evaluation of the polynomial in (4)

$$f^{(p)}(a_1 + \xi_1\theta, \ldots, a_{r-1} + \xi_{r-1}\theta,\ \theta) \neq 0, \qquad a_1, \ldots,\ a_{r-1} \in \mathbb{Z}_p \tag{5}$$

because the degree of the minimal polynomial of θ over $\mathbb{F}$ is $d+1$.

For $i = 1, \ldots, r-1$, each element $\xi_i\theta$ can be expressed as $h_i^{(p)}(\theta)$, where $h_i^{(p)}(X_r) \in \mathbb{Z}_p[X_r]$. Applying the Chinese Remainder Theorem to the different polynomials $h_i^{(p)}(X_r)$ for each prime $p|M$, we find the corresponding $h_i(X_r)$. By construction, $f^{(p)}(a_1 + h_1^{(p)}(X_r), \ldots, a_{r-1} + h_{r-1}^{(p)}(X_r),\ X_r)$ is not the zero polynomial by (5) for any integer values $a_1, \ldots,\ a_{r-1}$.

Now, we proceed to define a family of polynomials depending on $g_1(\mathbf{X}),\ldots,g_r(\mathbf{X})$ which will be the main subject of the article.

Let $\mathbb{K}$ be a field. We denote by $\mathcal{T}$ as the set of polynomials, f, such that $\sum_{j=0}^{s-1} a_j\,(f_{k+j}(\mathbf{X}) - f_{l+j}(\mathbf{X}))$ is nonconstant, where $f_i(\mathbf{X})$ are defined by (2) and $a_j \in \mathbb{K}$, with at least one $a_j \neq 0$ and $k \neq l$.

Here is a sufficient condition for a certain polynomial to be in class $\mathcal{T}$. To prove the result, we need some background.

We start defining a homomorphism of polynomial rings $\phi : \mathbb{K}[X_1, \ldots,\ X_r] \to \mathbb{K}[X_1, \ldots,\ X_r]$ with $\phi(X_i) = g_i(\mathbf{X})$.

Polynomials $g_1(\mathbf{X}), \ldots,\ g_r(\mathbf{X})$ are said to be algebraically independent if the application ϕ is injective. ϕ^k denotes the composition of the function ϕ k times with ϕ^0 being the identity map.

Lemma 2. *Let $f(\mathbf{X})$ be a polynomial in $\mathbb{K}[\mathbf{X}]$ and $g_1(\mathbf{X}), \ldots,\ g_r(\mathbf{X})$ be algebraically independent and $\mathbb{F}$ be an extension field of $\mathbb{K}$. Suppose that there exists $(b_1, \ldots,\ b_r)$, $(c_1, \ldots,\ c_r) \in \mathbb{F}^r$ two different zeros of the polynomials $g_1(\mathbf{X}), \ldots,\ g_r(\mathbf{X})$ with $f(c_1, \ldots,\ c_r) \neq f(b_1, \ldots,\ b_r)$, then $f \in \mathcal{T}$.*

Proof. Suppose that $k > l$, and exist $a_0, \ldots,\ a_{s-1} \in \mathbb{K}$ with $a_0 \neq 0$ satisfying; $\sum_{j=0}^{s-1} a_j\,(f_{k+j}(\mathbf{X}) - f_{l+j}(\mathbf{X})) = K$, where $K \in \mathbb{K}$.

Then

$$\sum_{j=0}^{s-1} a_j\,(f_{k+j}(\mathbf{X}) - f_{l+j}(\mathbf{X})) = \phi^{k-1}\left(\textstyle\sum_{j=0}^{s-1} a_j\,(f_{1+j}(\mathbf{X}) - f_{1-k+l+j}(\mathbf{X}))\right)$$

and this implies $K = \sum_{j=0}^{s-1} a_j\,((f_{1+j}(\mathbf{X})) - (f_{1-k+l+j}(\mathbf{X})))$ because ϕ is an injective map.

By equation (2), we notice that for $k \neq 0$, we have that $f_k(b_1, \ldots,\ b_r) = f_{k-1}(0, \ldots, 0) = f_k(c_1, \ldots,\ c_r)$, so substituting in the equation both points and subtracting the result, we get that $a_0 = 0$.

The last remark in this section is that conditions in this criterion can be tested using Groebner basis.

1.2 Exponential Sums and Previous Results

We start by listing some previous bounds on exponential sums which will be used to establish our main results.

The first Lemma is the well-known Hua-Loo Keng bound in a form which is a relaxation of the main result of [11] (see also Section 3 of [3] and Lemma 2.2 in [6]), followed by its multidimensional version.

Lemma 3. *For any polynomial* $f(X) = b_d X^d + \ldots + b_1 X + b_0 \in \mathbb{Z}_M[X]$ *of degree* $d \geq 1$*, there is a constant* $c_0 > 0$ *where the bound*

$$\left| \sum_{x \in \mathbb{Z}_M} \mathbf{e}_M(f(x)) \right| < e^{c_0 d} M^{1-1/d} G^{1/d}$$

holds, where $G = \gcd(b_d, \ldots, \ b_1, \ M)$.

Lemma 4. *Let* $f(\mathbf{X})$*, with total degree* $d \geq 2$ *and degree greater than one in* X_r*, be a polynomial with integer coefficients, with* $G = 1$*. Then the bound*

$$\left| \sum_{x_1, \ldots, x_r \in \mathbb{Z}_M} \mathbf{e}_M(f(x_1, \ldots, x_r)) \right| \leq e^{c_0 d^2 (\log d + 1)} M^{r - 1/(d^2(\log d + 1))}$$

holds, where c_0 *is some positive constant.*

Proof. We recall the univariate case that appears as Lemma 3. Then let

$$g(\mathbf{X}) = f(X_1 + h_1(X_r), X_2 + h_2(X_r), \ldots, X_r)$$

where $h_i(X_r)$ are the polynomials defined in Equation (3). It is easy to see that

$$\left| \sum \mathbf{e}_M(f(x_1, x_2 \ldots, x_r)) \right| = \left| \sum \mathbf{e}_M(g(x_1, x_2 \ldots, x_r)) \right|.$$

where the summations are taken over $x_1, x_2 \ldots, x_r \in \mathbb{Z}_M$ since $(x_1, \ldots, \ x_r) \to (x_1 + h_1(x_r), x_2 + h_2(x_r), \ldots, x_r)$ merely permutates the points. By Lemma 1, for any selection $x_1, \ldots, \ x_{r-1}$ this polynomial is not constant modulo p and the gcd of the coefficients of g and M are coprime. Hence, applying Lemma 3, we have

$$\begin{aligned} \left| \sum_{x_1, x_2 \ldots, x_r \in \mathbb{Z}_M} \mathbf{e}_M(g(x_1, x_2 \ldots, x_r)) \right| & \\ \leq \sum_{x_1, \ldots, x_{r-1} \in \mathbb{Z}_M} \left| \sum_{x_r \in \mathbb{Z}_M} \mathbf{e}_M(g(x_1, x_2 \ldots, x_r)) \right| & \\ \leq e^{c_0 d^2 (\log d + 1)} M^{r - 1/d^2(\log d + 1)}. & \end{aligned}$$

We obtain the last step by noting that the degree of g in X_r can be bounded by $d^2(\log d + 1)$ and so we are done.

This now allows us to state and prove the following Lemma.

Lemma 5. *Let $f(\mathbf{X})$ be a polynomial with integer coefficients with $\deg_{X_r} f \geq 1$ and total degree d. Recalling the definition of G,*

$$\left| \sum_{x_1 \ldots, x_r \in \mathbb{Z}_M} \mathbf{e}_M(f(x_1, \ldots, x_r)) \right| \leq e^{c_0 d^2 (\log d + 1)} M^r (G/M)^{1/d^2 (\log d + 1)}$$

Proof. We let

$$f_G(x_1, \ldots, x_r) = (f(x_1, \ldots, x_r) - f(\mathbf{0}))/G$$

and $m = M/G$.
Then,

$$\begin{aligned} \left| \sum_{x_1, \ldots, x_r \in \mathbb{Z}_M} \mathbf{e}_M(f(x_1, \ldots, x_r)) \right| &= \left| \sum_{x_1, \ldots, x_r \in \mathbb{Z}_M} \mathbf{e}_M(f(x_1, \ldots, x_r) - f(\mathbf{0})) \right| \\ &= G^r \left| \sum_{x_1, \ldots, x_r \in \mathbb{Z}_m} \mathbf{e}_m \left(f_G(x_1, \ldots, x_r)\right) \right| \end{aligned}$$

Now $f_G(x_1, \ldots, x_r)$ satisfies the conditions in Lemma 4, so:

$$G^r \left| \sum_{x_1, \ldots, x_r \in \mathbb{Z}_m} \mathbf{e}_m(f_G(x_1, \ldots, x_r)) \right| \leq G^r e^{c_0 d^2 (\log d + 1)} (m)^{r - 1/d^2 (\log d + 1)}$$

and so the result follows.

Lastly, we will make use of the following lemma, which is essentially the multi-dimensional version of Lemma 2.3 of [6].

Lemma 6. *Let $f(\mathbf{X}) \in \mathbb{Z}_M[\mathbf{X}]$ be a polynomial such that $f^{(p)} \in \mathcal{T}$ for every $p|M$ and let*

$$\sum_{j=0}^{s-1} a_j \left(f_{k+j}(\mathbf{X}) - f_{l+j}(\mathbf{X})\right) = \sum_{i_1=0}^{d_1} \cdots \sum_{i_r=0}^{d_r} b_{i_1, \ldots, i_r} X_1^{i_1} \ldots X_r^{i_r},$$

where $k \neq l$. Recalling the definition of G, the following equality $G = \gcd(a_0, \ldots, a_{s-1}, M)$ holds.

Proof. We put $A_j = a_j/G$ and $m = M/G$, $j = 0, \ldots, s-1$. In particular,

$$\gcd(A_0, \ldots, A_{s-1}, m) = 1. \tag{6}$$

It is enough to show that the polynomial

$$H(\mathbf{X}) = \sum_{j=0}^{s-1} A_j \left(f_{k+j}(\mathbf{X}) - f_{l+j}(\mathbf{X})\right)$$

is nonconstant modulo any prime $p|m$, for $k \neq l$.

By definition, we have

$$H^{(p)}(\mathbf{X}) \equiv \sum_{j=0}^{s-1} A_j \left(f_{k+j}^{(p)}(\mathbf{X}) - f_{l+j}^{(p)}(\mathbf{X}) \right) \pmod{p}$$

and $H^{(p)}(\mathbf{X})$ can not be a constant polynomial, since $f^{(p)} \in \mathcal{T}$ and so we are done.

1.3 Discrepancy

For a sequence of N points

$$\Gamma = (\gamma_{0,n}, \ldots, \gamma_{s-1,n})_{n=0}^{N-1} \tag{7}$$

of the half-open interval $[0,1)^s$, denote by Δ_Γ its discrepancy, that is,

$$\Delta_\Gamma = \sup_{B \subseteq [0,1)^s} \left| \frac{T_\Gamma(B)}{N} - |B| \right|,$$

where $T_\Gamma(B)$ is the number of points of the sequence Γ which hit the box

$$B = [\alpha_0, \beta_0) \times \ldots \times [\alpha_{s-1}, \beta_{s-1}) \subseteq [0,1)^s$$

and the supremum is taken over all such boxes.

For an integer vector $\mathbf{a} = (a_0, \ldots, a_{s-1}) \in \mathbb{Z}^s$ we put

$$|\mathbf{a}| = \max_{i=0,\ldots,s-1} |a_i|, \qquad r(\mathbf{a}) = \prod_{i=0}^{s-1} \max\{|a_i|, 1\}. \tag{8}$$

We need the *Erdös–Turán–Koksma inequality* (see Theorem 1.21 of [4]) for the discrepancy of a sequence of points of the s-dimensional unit cube, which we present in the following form.

Lemma 7. *There exists a constant $C_s > 0$ depending only on the dimension s such that, for any integer $L \geq 1$, for the discrepancy of a sequence of points (7) the bound*

$$\Delta_\Gamma < C_s \left(\frac{1}{L} + \frac{1}{N} \sum_{0 < |\mathbf{a}| \leq L} \frac{1}{r(\mathbf{a})} \left| \sum_{n=0}^{N-1} \exp\left(2\pi i \sum_{j=0}^{s-1} a_j \gamma_{j,n} \right) \right| \right)$$

holds, where $|\mathbf{a}|$, $r(\mathbf{a})$ are defined by (8) and the sum is taken over all integer vectors

$$\mathbf{a} = (a_0, \ldots, a_{s-1}) \in \mathbb{Z}^s$$

with $0 < |\mathbf{a}| \leq L$.

The currently best value of C_s is given in [2].

2 Discrepancy Bound

Let the sequence (u_n) generated by (1) be purely periodic with an arbitrary period t. For an integer vector $\mathbf{a} = (a_0, \ldots, a_{s-1}) \in \mathbb{Z}^s$ we introduce the exponential sum

$$S_{\mathbf{a}}(N) = \sum_{n=0}^{N-1} \mathbf{e}_M \left(\sum_{j=0}^{s-1} a_j u_{n+j} \right).$$

Theorem 1. *Let the sequence (u_n), given by (1) with a polynomial $f^{(p)}(\mathbf{X}) \in \mathcal{T}$, for every prime divisor p of M, with total degree d and $\deg_{X_r} f \geq 1$, be purely periodic with period t and $t \geq N \geq 1$. The bound*

$$\max_{\gcd(a_0,\ldots,a_{s-1},M)=G} |S_{\mathbf{a}}(N)| = O\left(N^{1/2} M^{r/2} (\log\log(M/G))^{-1/2}\right)$$

holds, where $G = \gcd(a_0, \ldots, a_{s-1}, M)$ and the implied constant depends only on s and d.

Proof. The proof follows a strategy first seen in [9].
Select any $\mathbf{a} = (a_0, \ldots, a_{s-1}) \in \mathbb{Z}^s$ with $\gcd(a_0, \ldots, a_{s-1}, M) = G$.

It is obvious that for any integer $k \geq 0$ we have

$$\left| S_{\mathbf{a}}(N) - \sum_{n=0}^{N-1} \mathbf{e}_M \left(\sum_{j=0}^{s-1} a_j u_{n+k+j} \right) \right| \leq 2k.$$

Therefore, for any integer $K \geq 1$,

$$K|S_{\mathbf{a}}(N)| \leq W + K^2,$$

where

$$W = \left| \sum_{n=0}^{N-1} \sum_{k=0}^{K-1} \mathbf{e}_M \left(\sum_{j=0}^{s-1} a_j u_{n+k+j} \right) \right| \leq \sum_{n=0}^{N-1} \left| \sum_{k=0}^{K-1} \mathbf{e}_M \left(\sum_{j=0}^{s-1} a_j u_{n+k+j} \right) \right|.$$

Accordingly, letting $\mathbf{x} = x_1, \ldots, x_r$, we obtain

$$\begin{aligned} W^2 &\leq N \sum_{n=0}^{N-1} \left| \sum_{k=0}^{K-1} \mathbf{e}_M \left(\sum_{j=0}^{s-1} a_j f_{k+j}\left(u_n, \ldots, u_{n-r+1}\right) \right) \right|^2 \\ &\leq N \sum_{\mathbf{x} \in \mathbb{Z}_M^r} \left| \sum_{k=0}^{K-1} \mathbf{e}_M \left(\sum_{j=0}^{s-1} a_j f_{k+j}\left(\mathbf{x}\right) \right) \right|^2 \\ &= N \sum_{k=0}^{K-1} \sum_{l=0}^{K-1} \sum_{\mathbf{x} \in \mathbb{Z}_M^r} \mathbf{e}_M \left(\sum_{j=0}^{s-1} a_j \left(f_{k+j}\left(\mathbf{x}\right) - f_{l+j}\left(\mathbf{x}\right) \right) \right). \end{aligned}$$

If $k = l$, then the inner sum is trivially equal to M^r. There are K such sums. Otherwise the polynomial $\sum_{j=0}^{s-1} a_j \left(f_{k+j}\left(\mathbf{X}\right) - f_{l+j}\left(\mathbf{X}\right)\right)$ is nonconstant since $f^{(p)} \in \mathcal{T}$. Hence we can apply Lemma 5 and Lemma 6 (so that we only need consider $a_j, j = 0, \ldots, s-1$, instead of the coefficients of f) to the inner sum, obtaining the upper bound

$$e^{c_0 d^{3(K+s-2)}} M^{r-1/d^{3(K+s-2)}} G^{1/d^{3(K+s-2)}}$$

for at most K^2 sums and positive constant c_0 and noting that
$d^2(\log d + 1) < d^3$.
Hence,

$$W^2 \leq KNM^r + K^2 N e^{c_0 d^{3(K+s-2)}} M^{r-1/d^{3(K+s-2)}} G^{1/d^{3(K+s-2)}}.$$

Now, without too much loss of generality we may assume $(d+1)^{3(K+s-2)} \geq 2$. Next we put $K = \lceil \log\log(M/G)/(3c\log(d+1)) \rceil$, for some $c > 2$ to guarantee that the first term dominates and the result follows.

Next, let $D_s(N)$ denote the discrepancy of the points given by

$$\left(\frac{u_n}{M}, \ldots, \frac{u_{n+s-1}}{M}\right), \qquad n = 0, \ldots, N-1,$$

in the s-dimensional unit cube $[0,1)^s$.

Theorem 2. *If the sequence (u_n), given by (1) with a polynomial $f^{(p)}(\mathbf{X}) \in \mathcal{T}$, for every prime divisor p of M, with total degree d and $\deg_{X_r} f \geq 1$ is purely periodic with period t with $t \geq N \geq 1$, then the bound*

$$D_s(N) = O\left(N^{-1/2} M^{r/2} (\log\log\log M)^s / (\log\log M)^{1/2}\right)$$

holds, where the implied constant depends only on s and d.

Proof. The statement follows from Lemma 7, taken with

$$L = \left\lceil N^{1/2} M^{-r/2} (\log\log M)^{1/2} \right\rceil$$

and the bound of Theorem 1, where all occurring $G = \gcd(a_0, \ldots, a_{s-1}, M)$ are at most L.

References

1. Arkhipov, G.I., Chubarikov, V.N., Karatsuba, A.A.: Trigonometric Sums in Number Theory and Analysis, de Gruyter Expositions in Mathematics, Berlin, vol. 39 (2004)
2. Cochrane, T.: Trigonometric approximation and uniform distribution modulo 1. Proc. Amer. Math. Soc. 103, 695–702 (1988)
3. Cochrane, T., Zheng, Z.Y.: A Survey on Pure and Mixed Exponential Sums Modulo Prime Numbers. Proc. Illinois Millenial Conf. on Number Theory 1, 271–300 (2002)

4. Drmota, M., Tichy, R.F.: Sequences, discrepancies and applications. Springer, Berlin (1997)
5. El-Mahassni, E.D., Shparlinski, I.E., Winterhof, A.: Distribution of nonlinear congruential pseudorandom numbers for almost squarefree integers. Monatsh. Math. 148, 297–307 (2006)
6. El-Mahassni, E.D., Winterhof, A.: On the distribution of nonlinear congruential pseudorandom numbers in residue rings. Intern. J. Number Th. 2(1), 163–168 (2006)
7. Griffin, F., Niederreiter, H., Shparlinski, I.: On the distribution of nonlinear recursive congruential pseudorandom numbers of higher orders. In: Fossorier, M.P.C., Imai, H., Lin, S., Poli, A. (eds.) AAECC 1999. LNCS, vol. 1719, pp. 87–93. Springer, Heidelberg (1999)
8. Gutierrez, J., Gomez-Perez, D.: Iterations of multivariate polynomials and discrepancy of pseudorandom numbers. In: Bozta, S., Sphparlinski, I. (eds.) AAECC 2001. LNCS, vol. 2227, pp. 192–199. Springer, Heidelberg (2001)
9. Niederreiter, H., Shparlinski, I.E.: On the distribution and lattice structure of nonlinear congruential pseudorandom numbers. Finite Fields and Their Appl. 5, 246–253 (1999)
10. Niederreiter, H., Shparlinski, I.E.: Exponential sums and the distribution of inversive congruential pseudorandom numbers with prime-power modulus. Acta Arith. 92, 89–98 (2000)
11. Stečkin, S.B.: An estimate of a complete rational exponential sum. Trudy Mat. Inst. Steklov. 143, 188–207 (1977) (in Russian)

Rooted Trees Searching for Cocyclic Hadamard Matrices over D_{4t}

Víctor Álvarez, José Ándrés Armario, María Dolores Frau, Félix Gudiel, and Amparo Osuna*

Dpto. Matemática Aplicada 1, E.T.S.I.Informática, Avda. Reina Mercedes s/n, 41012 Sevilla, Spain

Abstract. A new reduction on the size of the search space for cocyclic Hadamard matrices over dihedral groups D_{4t} is described, in terms of the so called *central distribution*. This new search space adopt the form of a forest consisting of two rooted trees (the vertices representing subsets of coboundaries) which contains all cocyclic Hadamard matrices satisfying the constraining condition. Experimental calculations indicate that the ratio between the number of constrained cocyclic Hadamard matrices and the size of the constrained search space is greater than the usual ratio.

Keywords: Hadamard matrix, cocyclic matrix, dihedral groups.

1 Introduction

Since Hadamard matrices (that is, $\{1, -1\}$-square matrices whose rows are pairwise orthogonal) were introduced at the end of the XIXth century, the interest in their construction has grown substantially, because of their multiple applications (see [7] for instance).

For this reason, many attempts and efforts have been devoted to the design of good construction procedures, the latest involving heuristic techniques (see [5], [1], [4] for instance). Even alternative theoretical descriptions characterizing Hadamard matrices have been proposed (for instance, Ito's works involving Hadamard graphs [11] in the middle eighties, and Hadamard groups [6,12] more recently). But no matter what one may think, Hadamard matrices keep on being elusive anyway.

The point is that though it may be easily checked that the size of a Hadamard matrix is to be 2 or a multiple of 4, there is no certainty whether such a Hadamard matrix exists for every size $4t$. This is the *Hadamard conjecture*, which remains unsolved for more than a century.

In fact, the design of a procedure which outputs a Hadamard matrix of the desired size has shown to be as important as solving the Hadamard conjecture itself.

* All authors are partially supported by the research projects FQM–296 and P07-FQM-02980 from Junta de Andalucía and MTM2008-06578 from Ministerio de Ciencia e Innovación (Spain).

M. Bras-Amorós and T. Høholdt (Eds.): AAECC 2009, LNCS 5527, pp. 204–214, 2009.

In the early 90s, a surprising link between homological algebra and Hadamard matrices [9] led to the study of cocyclic Hadamard matrices [10]. The main advantages of the cocyclic framework are that:

- On one hand, the Hadamard test for cocyclic matrices [10] runs in $O(t^2)$ time, better than the $O(t^3)$ algorithm for usual (not necessarily cocyclic) Hadamard matrices.
- On the other hand, the search space reduces drastically, though it still is often of exponential size (see [4,2] for details).

Among the groups for which cocyclic Hadamard matrices have been found, it seems that dihedral groups D_{4t} are more likely to give a more density of cocyclic Hadamard matrices, even for every order multiple of 4 (see [8,3,2] for instance).

Unfortunately, the task of explicitly construct cocyclic Hadamard matrices over D_{4t} is considerably difficult, since the search space inherits a exponential size. New ideas dealing with this problem are welcome.

The purpose of this paper is to describe a new reduction on the size of the search space for cocyclic Hadamard matrices over dihedral groups D_{4t}. The key idea is exploiting the notions of *i-paths* and *intersections* introduced in [2], in order to design a forest consisting of two rooted trees (the vertices representing subsets of coboundaries) which contains all cocyclic Hadamard matrices satisfying the so called *central distribution*. We will explain these notions in the following section.

We organize the paper as follows. Section 2 is devoted to preliminaries on cocyclic matrices. Section 3 describes the new search space for cocyclic Hadamard matrices over D_{4t}. We include some final remarks and comments.

2 Preliminaries on Cocyclic Matrices

Consider a multiplicative group $G=\{g_1=1, g_2, \ldots, g_{4t}\}$, not necessarily abelian. A cocyclic matrix M_f over G consists in a binary matrix $M_f = (f(g_i, g_j))$ coming from a 2-cocycle f over G, that is, a map $f : G \times G \to \{1, -1\}$ such that

$$f(g_i, g_j)f(g_ig_j, g_k) = f(g_j, g_k)f(g_i, g_jg_k), \qquad \forall\, g_i, g_j, g_k \in G.$$

We will only use normalized cocycles f (and hence normalized cocyclic matrices M_f), so that $f(1, g_j) = f(g_i, 1) = 1$ for all $g_i, g_j \in G$ (and correspondingly $M_f = (f(g_i, g_j))$ consists of a first row and column all of 1s).

A basis $\mathcal{B}$ for 2-cocycles over G consists of some representative 2-cocycles (coming from inflation and transgression) and some elementary 2-coboundaries ∂_i, so that every cocyclic matrix admits a unique representation as a Hadamard (pointwise) product $M = M_{\partial_{i_1}} \ldots M_{\partial_{i_w}} \cdot R$, in terms of some coboundary matrices $M_{\partial_{i_j}}$ and a matrix R formed from representative cocycles.

Recall that every *elementary coboundary* ∂_d is constructed from the characteristic set map $\delta_d : G \to \{\pm 1\}$ associated to an element $g_d \in G$, so that

$$\partial_d(g_i, g_j) = \delta_d(g_i)\delta_d(g_j)\delta_d(g_ig_j) \qquad \text{for} \qquad \delta_d(g_i) = \begin{cases} -1 & g_d = g_i \\ 1 & g_d \neq g_i \end{cases} \tag{1}$$

Although the elementary coboundaries generate the set of all coboundaries, they might not be linearly independent (see [3] for details). Moreover, since the elementary coboundary ∂_{g_1} related to the identity element in G is not normalized, we may assume that $\partial_{g_1} \notin \mathcal{B}$.

The cocyclic Hadamard test asserts that a cocyclic matrix is Hadamard if and only if the summation of each row (but the first) is zero [10]. In what follows, the rows whose summation is zero are termed *Hadamard rows*.

This way, a cocyclic matrix M_f is Hadamard if and only if every row $(M_f)_i$ is a Hadamard row, $2 \leq i \leq 4t$.

In [2] the Hadamard character of a cocyclic matrix is described in an equivalent way, in terms of *generalized coboundary matrices*, *i-walks* and *intersections*. We reproduce now these notions.

The *generalized coboundary matrix* $\bar{M}_{\partial_j}$ related to a elementary coboundary ∂_j consists in negating the j^{th}-row of the matrix M_{∂_j}. Note that negating a row of a matrix does not change its Hadamard character. As it is pointed out in [2], every generalized coboundary matrix $\bar{M}_{\partial_j}$ contains exactly two negative entries in each row $s \neq 1$, which are located at positions (s, i) and (s, e), for $g_e = g_s^{-1} g_i$. We will work with generalized coboundary matrices from now on.

A set $\{\bar{M}_{\partial_{i_j}} : 1 \leq j \leq w\}$ of generalized coboundary matrices defines an *i-walk* if these matrices may be ordered in a sequence $(\bar{M}_{l_1}, \ldots, \bar{M}_{l_w})$ so that consecutive matrices share exactly one negative entry at the i^{th}-row. Such a walk is called an *i-path* if the initial and final matrices do not share a common -1, and an *i-cycle* otherwise. As it is pointed out in [2], every set of generalized coboundary matrices may be uniquely partitioned into disjoint maximal i-walks.

From the definition above, it is clear that every maximal i-path contributes two negative occurrences at the i^{th}-row. This way, a characterization of Hadamard rows (consequently, of Hadamard matrices) may be easily described in terms of i-paths.

Proposition 1. *[2] The i^{th} row of a cocyclic matrix $M = M_{\partial_{i_1}} \ldots M_{\partial_{i_w}} \cdot R$ is Hadamard if and only if*

$$2c_i - 2I_i = 2t - r_i \tag{2}$$

where c_i denotes the number of maximal i-paths in $\{\bar{M}_{\partial_{i_1}}, \ldots, \bar{M}_{\partial_{i_w}}\}$, r_i counts the number of -1s in the i^{th}-row of R and I_i indicates the number of positions in which R and $\bar{M}_{\partial_{i_1}} \ldots \bar{M}_{\partial_{i_w}}$ share a common -1 in their i^{th}-row.

From now on, we will refer to the positions in which R and $\bar{M}_{\partial_{i_1}} \ldots \bar{M}_{\partial_{i_w}}$ share a common -1 in a given row simply as *intersections*, for brevity.

We will now focus on the case of dihedral groups.

3 Cocyclic Matrices over D_{4t}

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 [6] 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,

$$f(a^i, a^j b^k) = \begin{cases} A^{ij}, & i+j < 2t, \\ A^{ij}K, & i+j \geq 2t, \end{cases} \qquad f(a^i b, a^j b^k) = \begin{cases} A^{ij}B^k, & i \geq j, \\ A^{ij}B^k K, & i < j, \end{cases}$$

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 [2], and consists of the elementary coboundaries $\{\partial_a, \ldots, \partial_{a^{2t-3}b}\}$. This way, a basis for 2-cocycles over D_{4t} is given by $\mathcal{B} = \{\partial_a, \ldots, \partial_{a^{2t-3}b}, \beta_1, \beta_2, \gamma\}$.

Computational results in [6,2] suggest that the case $(A, B, K) = (1, -1, -1)$ contains a large density of cocyclic Hadamard matrices.

Furthermore, as it is pointed out in Theorem 2 of [2], cocyclic matrices over D_{4t} using $R = \beta_2\gamma$ are Hadamard matrices if and only if rows from 2 to t are Hadamard, so that the cocyclic test runs four times faster than usual.

From now on, we assume that $R = M_{\beta_2} \cdot M_\gamma = \begin{pmatrix} A & A \\ B & -B \end{pmatrix}$ for

$$A = \begin{pmatrix} 1 & 1 \cdots & 1 \\ 1 & \ddots & -1 \\ \vdots & \ddots & \ddots & \vdots \\ 1 & -1 \cdots & -1 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 1 & -1 \cdots & -1 \\ \vdots & \ddots & \ddots & \vdots \\ 1 & & \ddots & -1 \\ 1 & 1 \cdots & & 1 \end{pmatrix} \tag{3}$$

Now we would like to know how the 2-coboundaries in $\mathcal{B}$ have to be combined to form i-paths, $2 \leq i \leq t$. This information is given in Proposition 7 of [2].

Proposition 2. *[2] For $1 \leq i \leq 2t$, a maximal i-walk consists of a maximal subset in*

$$(M_{\partial_1}, \ldots, M_{\partial_{2t}}) \quad \textit{or} \quad (M_{\partial_{2t+1}}, \ldots, M_{\partial_{4t}})$$

formed from matrices $(\ldots, M_j, M_k, \ldots)$ which are cyclically separated in $i-1$ positions (that is $j \pm (i-1) \equiv k \bmod 2t$).

Notice that since $r_i = 2(i-1)$ for $2 \leq i \leq t$, the cocyclic Hadamard test reduces to check whether $c_i - I_i = t - i + 1$, for $2 \leq i \leq t$. Thus c_i uniquely determines I_i and reciprocally, $2 \leq i \leq t$.

In fact, the way in which intersections may be introduced at the i^{th}-row is uniquely determined. More explicitly

Lemma 1. *The following table gives a complete distribution of the coboundaries in $\mathcal{B}$ which may create an intersection at a given row. For clarity in the reading, we note the generalized coboundary $\bar{M}_{\partial_i}$ simply by i:*

row	*coboundaries*
2	$2t, 2t+1$
3	$2, 2t-1, 2t, 2t+1, 2t+2$
$4 \leq k \leq t$	$2, \ldots, k-1, 2t-k+2, \ldots, 2t+k-1, 4t-k+2, \ldots, 4t-2$

Proof
It may be seen by inspection, taking into account the distribution of the negative occurrences in R and the form of the generalized coboundary matrices. □

Lemma 2. *In particular, there are some coboundaries which do not produce any intersection at all, at rows $2 \leq k \leq t$, which we term free intersection coboundaries. More concretely,*

t	*coboundaries*
2	$2, 3, 6$
$t > 2$	$t, t+1, 3t, 3t+1$

Proof
It suffices to take the set difference between $\mathcal{B}$ and the set of coboundaries used in the lemma above. □

Lemma 3. *Furthermore, the following table distributes the coboundaries which produce a intersection at every row, so that coboundaries which produce the same negative occurrence at a row are displayed vertically in the same column.*

row	*coboundaries*
2	$2t$ $2t+1$
3	$2t$ $2t-1$ 2 $2t+1$ $2t+2$
$4 \leq k \leq t$	$2t-k+2$ $2t-k+3 \ \ldots \ 2t-1$ / $2 \ \ldots \ k-2$ $2t$ $k-1$ $2t+1$ $4t-k+2$ $2t+2 \ \ldots \ 2t+k-3$ / $4t-k+1 \ \ldots \ 4t-2$ $2t+k-2$ $2t+k-1$

Proof
It may be seen by inspection. □

Remark 1. Notice that:

- The set of coboundaries which may produce an intersection at the i^{th}-row is included in the analog set corresponding to the $(i+1)^{th}$-row.
- The boxed coboundaries do not produce any intersection at the precedent rows.

Now one could ask whether cocyclic Hadamard matrices exist for any formal distribution of pairs (c_i, I_i) satisfying the relations $c_i - I_i = t - i + 1$, for $2 \leq i \leq t$. Actually, this is not the case.

Proposition 3. *Not all of the formal sequences $[(c_2, I_2), \ldots, (c_t, I_t)]$ satisfying $c_i - I_i = t - i + 1$ give rise to cocyclic Hadamard matrices over D_{4t}, for $t \geq 3$.*

Proof

Proposition 10 of [2] bounds the number w of coboundaries in $\mathcal{B}$ to multiply with R so that a cocyclic Hadamard matrix is formed, so that $t-1 \leq w \leq 3t+2$.

In particular, for $t \geq 6$, we know that $5 \leq w$. Consequently, the case $I_2 = \ldots = I_t = 0$ is not feasible, since from Lemma 2 we know that only up to 4 coboundaries may be combined so that no intersection is generated at any row.

This proves the Lemma for $t \geq 6$. We now study the remaining cases.

Taking into account that $0 \leq I_i \leq r_i = 2i-2$, we may have a look in the way in which cocyclic Hadamard matrices are distributed regarding the number of intersections I_i, for those groups D_{4t} for which the whole set of cocyclic Hadamard matrices have been computed until now. These are precisely $t = 2, 3, 4, 5$.

For $t = 2$, we formally have 3 solutions for the equation $c_2 - I_2 = 1$,

c_2	1	2	3
I_2	0	1	2

Each of these solutions gives rise to some cocyclic Hadamard matrices M_f,

I_2	0	1	2
$\lvert M_f \rvert$	4	10	2

For $t = 3$, we formally have 15 solutions for the system $\begin{cases} c_2 - I_2 = 2 \\ c_3 - I_3 = 1 \end{cases}$ coming from the combination of any solution of each of the equations

c_2	2	3	4
I_2	0	1	2

c_3	1	2	3	4	5
I_3	0	1	2	3	4

Only 9 of the 15 hypothetical solutions are real solutions (there are no combinations of coboundaries meeting the other 6 "theoretical" solutions), distributed in the following way:

(I_2, I_3)	$(0,1)$	$(0,2)$	$(0,3)$	$(1,1)$	$(1,2)$	$(1,3)$	$(2,1)$	$(2,2)$	$(2,3)$
$\lvert M_f \rvert$	6	10	2	8	20	8	2	10	6

For $t = 4$, we formally have 105 solutions for the system $\begin{cases} c_2 - I_2 = 3 \\ c_3 - I_3 = 2 \\ c_4 - I_4 = 1 \end{cases}$ coming from the combination of any solution of each of the equations

c_2	3	4	5
I_2	0	1	2

c_3	2	3	4	5	6
I_3	0	1	2	3	4

c_4	1	2	3	4	5	6	7
I_4	0	1	2	3	4	5	6

Only 36 of the 105 hypothetical solutions are real solutions, distributed in the following way:

(I_2, I_3, I_4)	$(0,0,2)$	$(0,1,1)$	$(0,1,2)$	$(0,1,3)$	$(0,1,4)$	$(0,2,2)$	$(0,2,3)$	$(0,2,4)$	$(0,2,5)$
$\lvert M_f\rvert$	8	6	18	18	6	12	12	24	12
(I_2, I_3, I_4)	$(0,3,3)$	$(0,3,4)$	$(1,0,3)$	$(1,1,1)$	$(1,1,2)$	$(1,1,3)$	$(1,1,4)$	$(1,1,5)$	$(1,2,1)$
$\lvert M_f\rvert$	4	8	12	18	12	8	4	6	4
(I_2, I_3, I_4)	$(1,2,2)$	$(1,2,3)$	$(1,2,4)$	$(1,2,5)$	$(1,3,2)$	$(1,3,3)$	$(1,3,4)$	$(1,4,3)$	$(2,1,2)$
$\lvert M_f\rvert$	24	72	24	4	12	32	20	20	2
(I_2, I_3, I_4)	$(2,2,1)$	$(2,2,2)$	$(2,2,3)$	$(2,2,4)$	$(2,3,2)$	$(2,3,3)$	$(2,1,3)$	$(2,1,4)$	$(2,1,5)$
$\lvert M_f\rvert$	6	6	2	12	24	12	12	24	12

For $t = 5$, we formally have 945 solutions for the system $\begin{cases} c_2 - I_2 = 4 \\ c_3 - I_3 = 3 \\ c_4 - I_4 = 2 \\ c_5 - I_5 = 1 \end{cases}$ coming from the combination of any solution of each of the equations

c_2	4	5	6
I_2	0	1	2

c_3	3	4	5	6	7
I_3	0	1	2	3	4

c_4	2	3	4	5	6	7	8
I_4	0	1	2	3	4	5	6

c_5	1	2	3	4	5	6	7	8	9
I_4	0	1	2	3	4	5	6	7	8

Only 153 of the 945 hypothetical solutions are real solutions,

(I_2, I_3, I_4, I_5)	$(0,0,1,3)$	$(0,0,2,3)$	$(0,1,1,4)$	$(0,1,2,2)$	$(0,1,2,3)$	$(0,1,2,4)$
$\lvert M_f\rvert$	4	12	3	6	12	21
(I_2, I_3, I_4, I_5)	$(0,1,2,5)$	$(0,1,3,2)$	$(0,1,3,3)$	$(0,1,3,4)$	$(0,1,3,5)$	$(0,1,3,6)$
$\lvert M_f\rvert$	6	6	12	21	21	9
(I_2, I_3, I_4, I_5)	$(0,1,3,7)$	$(0,1,4,3)$	$(0,1,4,4)$	$(0,1,4,5)$	$(0,1,4,6)$	$(0,1,4,7)$
$\lvert M_f\rvert$	3	6	3	9	18	6
(I_2, I_3, I_4, I_5)	$(0,2,0,2)$	$(0,2,1,2)$	$(0,2,1,3)$	$(0,2,1,4)$	$(0,2,2,2)$	$(0,2,2,3)$
$\lvert M_f\rvert$	2	4	2	2	4	12
(I_2, I_3, I_4, I_5)	$(0,2,2,4)$	$(0,2,3,2)$	$(0,2,3,3)$	$(0,2,3,4)$	$(0,2,3,5)$	$(0,2,3,6)$
$\lvert M_f\rvert$	4	8	8	24	14	2
(I_2, I_3, I_4, I_5)	$(0,2,4,4)$	$(0,2,4,5)$	$(0,2,4,6)$	$(0,2,5,4)$	$(0,3,2,5)$	$(0,3,3,2)$
$\lvert M_f\rvert$	12	26	4	4	2	2
(I_2, I_3, I_4, I_5)	$(0,3,3,3)$	$(0,3,3,4)$	$(0,3,3,5)$	$(0,3,4,2)$	$(0,3,4,3)$	$(0,3,4,4)$
$\lvert M_f\rvert$	7	6	6	1	4	5
(I_2, I_3, I_4, I_5)	$(0,3,4,5)$	$(0,3,5,4)$	$(1,0,1,4)$	$(1,0,2,2)$	$(1,0,2,3)$	$(1,0,2,4)$
$\lvert M_f\rvert$	6	1	2	4	2	4
(I_2, I_3, I_4, I_5)	$(1,0,4,2)$	$(1,0,4,3)$	$(1,0,4,4)$	$(1,0,5,4)$	$(1,1,1,3)$	$(1,1,1,4)$
$\lvert M_f\rvert$	4	2	4	2	6	3
(I_2, I_3, I_4, I_5)	$(1,1,2,3)$	$(1,1,2,4)$	$(1,1,2,5)$	$(1,1,2,6)$	$(1,1,3,2)$	$(1,1,3,3)$
$\lvert M_f\rvert$	7	14	22	8	4	16
(I_2, I_3, I_4, I_5)	$(1,1,3,4)$	$(1,1,3,5)$	$(1,1,3,6)$	$(1,1,3,7)$	$(1,1,4,3)$	$(1,1,4,4)$
$\lvert M_f\rvert$	20	8	16	16	5	10
(I_2, I_3, I_4, I_5)	$(1,1,4,5)$	$(1,1,4,6)$	$(1,1,5,3)$	$(1,1,5,4)$	$(1,2,1,3)$	$(1,2,1,4)$
$\lvert M_f\rvert$	18	8	2	1	4	8

(I_2, I_3, I_4, I_5)	$(1,2,2,2)$	$(1,2,2,3)$	$(1,2,2,4)$	$(1,2,2,5)$	$(1,2,3,2)$	$(1,2,3,3)$
$\lvert M_f\rvert$	4	16	28	16	24	20
(I_2, I_3, I_4, I_5)	$(1,2,3,4)$	$(1,2,3,5)$	$(1,2,4,2)$	$(1,2,4,3)$	$(1,2,4,4)$	$(1,2,4,5)$
$\lvert M_f\rvert$	32	56	4	12	24	24
(I_2, I_3, I_4, I_5)	$(1,2,5,4)$	$(1,3,1,2)$	$(1,3,1,3)$	$(1,3,1,4)$	$(1,3,2,2)$	$(1,3,2,3)$
$\lvert M_f\rvert$	8	1	3	2	8	16
(I_2, I_3, I_4, I_5)	$(1,3,2,4)$	$(1,3,2,5)$	$(1,3,3,3)$	$(1,3,3,4)$	$(1,3,3,5)$	$(1,3,4,2)$
$\lvert M_f\rvert$	10	6	24	32	16	8
(I_2, I_3, I_4, I_5)	$(1,3,4,3)$	$(1,3,4,4)$	$(1,3,4,5)$	$(1,3,5,2)$	$(1,3,5,3)$	$(1,3,5,4)$
$\lvert M_f\rvert$	16	22	10	3	9	6
(I_2, I_3, I_4, I_5)	$(1,4,1,3)$	$(1,4,2,3)$	$(1,4,2,4)$	$(1,4,3,3)$	$(1,4,3,4)$	$(1,4,4,3)$
$\lvert M_f\rvert$	2	6	6	16	8	6
(I_2, I_3, I_4, I_5)	$(1,4,4,4)$	$(1,4,5,3)$	$(2,1,2,3)$	$(2,1,2,4)$	$(2,1,2,5)$	$(2,1,2,6)$
$\lvert M_f\rvert$	6	2	2	1	3	6
(I_2, I_3, I_4, I_5)	$(2,1,2,7)$	$(2,1,3,2)$	$(2,1,3,3)$	$(2,1,3,4)$	$(2,1,3,5)$	$(2,1,3,6)$
$\lvert M_f\rvert$	2	2	4	7	7	3
(I_2, I_3, I_4, I_5)	$(2,1,3,7)$	$(2,1,4,2)$	$(2,1,4,3)$	$(2,1,4,4)$	$(2,1,4,5)$	$(2,1,5,4)$
$\lvert M_f\rvert$	1	2	4	7	2	1
(I_2, I_3, I_4, I_5)	$(2,2,1,4)$	$(2,2,2,4)$	$(2,2,2,5)$	$(2,2,2,6)$	$(2,2,3,2)$	$(2,2,3,3)$
$\lvert M_f\rvert$	4	12	26	4	8	8
(I_2, I_3, I_4, I_5)	$(2,2,3,4)$	$(2,2,3,5)$	$(2,2,3,6)$	$(2,2,4,2)$	$(2,2,4,3)$	$(2,2,4,4)$
$\lvert M_f\rvert$	24	14	2	4	12	4
(I_2, I_3, I_4, I_5)	$(2,2,5,2)$	$(2,2,5,3)$	$(2,2,5,4)$	$(2,2,6,2)$	$(2,3,1,4)$	$(2,3,2,2)$
$\lvert M_f\rvert$	4	2	2	2	3	3
(I_2, I_3, I_4, I_5)	$(2,3,2,3)$	$(2,3,2,4)$	$(2,3,2,5)$	$(2,3,3,2)$	$(2,3,3,3)$	$(2,3,3,4)$
$\lvert M_f\rvert$	12	15	18	6	21	18
(I_2, I_3, I_4, I_5)	$(2,3,3,5)$	$(2,3,4,5)$	$(2,4,2,3)$			
$\lvert M_f\rvert$	18	6	12			

Attending to the tables above, we conclude that, for $2 \leq t \leq 5$, there is a large density of cocyclic Hadamard matrices in the case $c_i = t$ for $2 \leq i \leq t$, that is, $(I_2, \ldots, I_t) = (1, \ldots t-1)$. We call this case the *central distribution* for intersections and i-paths on D_{4t}.

We include now a table comparing the number *central* of cocyclic Hadamard matrices in the central distribution with the proportion $\% = \frac{|M_f|}{cases}$ of the amount $|M_f|$ of cocyclic Hadamard matrices over D_{4t} by the total number *cases* of valid distributions of intersections $(I_2, \ldots, I_t)$. The last column contains the number of cocyclic Hadamard matrices of the most prolific case:

t	$cases$	$\lvert M_f\rvert$	%	$central$	$best$
2	3	16	5.33	10	10
3	9	72	8	20	20
4	36	512	14.22	72	72
5	153	1400	9.15	32	56

It seems then reasonable trying to constraint the search for cocyclic Hadamard matrices over D_{4t} to the central distribution case.

The search space in the central distribution $(I_2, \ldots, I_t) = (1, \ldots, t-1)$ may be represented as a forest of two rooted trees of depth $t-1$. We identify each level of the tree to the correspondent row of the cocyclic matrix at which intersections are being counted, so that the roots of the trees are located at level 2 (corresponding to the intersections created at the second row of the cocyclic matrix).

This way the level i contains those coboundaries which must be added to the father configuration in order to get the desired $i-1$ intersections at the i^{th}-row, for $2 \leq i \leq t$.

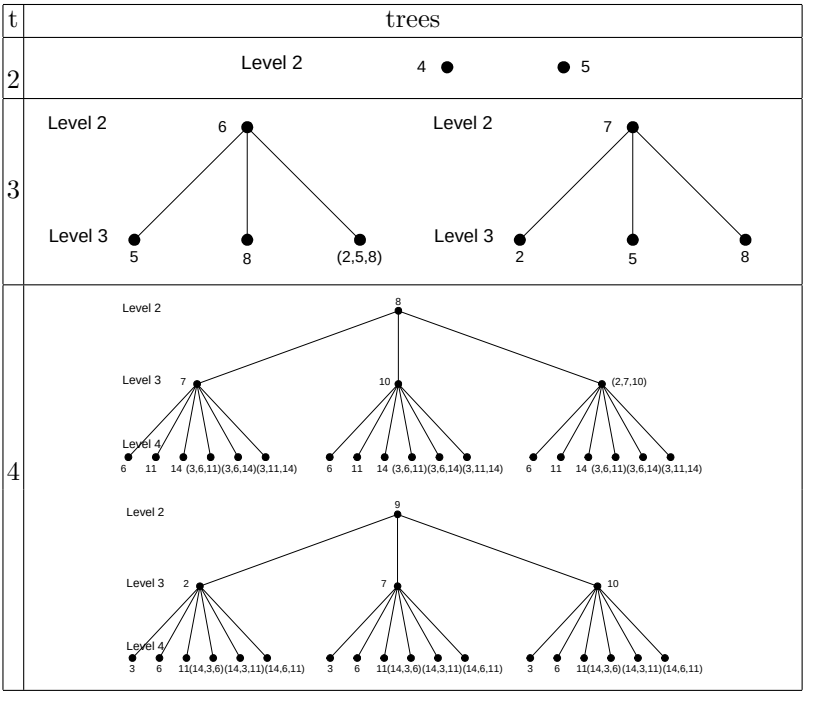
The root of the first tree is ∂_{2t}, whereas the root of the second tree is ∂_{2t+1}, since from Lemma 1 these are the only coboundaries which may give an intersection at the second row.

As soon as one of these coboundaries is used, the other one is forbidden, since otherwise a second intersection would be introduced at the second row.

Now one must add some coboundaries to get two intersections at the third row. Since ∂_{2t} is already used and ∂_{2t+1} is forbidden, there are only 3 coboundaries left (those boxed in the table of Lemma 3).

Successively, in order to construct the nodes at level k, one must add some of the correspondent boxed coboundaries of the table of Lemma 3, since the remaining coboundaries are either used or forbidden.

We include the forests corresponding to the cases $t = 2, 3, 4$ for clarity.



Every branch ending at level t gives a cocyclic matrix meeting the desired distribution of intersections $(I_2, \ldots, I_t)$. Now one has to check whether any of the 16 possible combinations with the free intersection coboundaries $\{\partial_t, \partial_{t+1}, \partial_{3t}, \partial_{3t+1}\}$ of Lemma 2 gives rise to a cocyclic Hadamard matrix (that is, to the desired distribution of i-paths, $(c_2, \ldots, c_t) = (t, \ldots, t)$).

We now give some properties of the trees above.

Proposition 4. *In the circumstances above, it may be proved that*

1. *The skeleton (i.e., the branches, forgetting about the indexes of the coboundaries used) of the trees related to D_{4t} are preserved to form the levels from 2 to t corresponding to the trees of $D_{4(t+1)}$.*
2. *Among the boxed coboundaries $\{\partial_{2t-k+2}, \partial_{k-1}, \partial_{4t-k+2}, \partial_{2t+k-1}$ to be added at level k of the trees, precisely one of them removes an intersection, whereas the remaining three adds one intersection each.*
3. *At each level, either just one or exactly three boxed coboundaries must be used, there is no other possible choice in order to meet the desired amount of intersections.*
4. *Consequently, a branch may be extended from level k to level $k+1$ if and only if $k - h_k \in \{-1, 1, 3\}$, where h_k denotes the number of intersections inherited from level k to level $k+1$.*
5. *Branches ending at levels above level t will never give rise to cocyclic Hadamard matrices meeting the central distribution. This will be more frequent the greater t is.*
6. *Both trees may have branches ending at level t which may not produce any cocyclic Hadamard matrix at all. This will be more frequent the greater t is.*

Proof

Most of the properties are consequences of the results explained in Lemma 1 through Lemma 3, and are left to the reader.

Property 3 comes as a result of a parity condition: there must be an odd number of intersections at odd levels, and an even number of intersections at even levels. Since boxed coboundaries either add an intersection each, or just one of them removes an intersection (added by a coboundary previously used), the parity condition leads to the result.

Concerning Property 5, we give a branch not reaching the last level for $t = 9$:

level	2	3	4	5	6	7	8
cob.	18	17	21	4, 22, 33	32	6, 13, 24	12, 25, 30

Concerning Property 6, we give a branch reaching the last level for $t = 9$, which do not give rise to any cocyclic Hadamard matrix at all:

level	2	3	4	5	6	7	8	9
cob.	19	17	21	22	14	6, 24, 31	12, 25, 30	29

□

So far, it is evident that the above trees reduce the search space for cocyclic Hadamard matrices over D_{4t}, constraining the solutions to the central distribution case.

There is only one question left. Is the new proportion $ratio_c$ of cocyclic Hadamard matrices in the central distribution case by the size of the reduced space greater than the proportion $ratio_g$ of general cocyclic Hadamad matrices by the size of the general search space?

It seems so, attending to the table below (we have followed the calculations of [2] about the size of the general search space in D_{4t}).

t	$\|M_f\|$	g. size	$ratio_g$	$\|M_f\|$	central size	$ratio_c$
2	16	32	0.5	10	16	0.625
3	72	492	0.146	20	96	0.208
4	512	8008	0.063	72	576	0.125

We claim that developing a heuristic search in the forest described above will produce some cocyclic Hadamard matrices over D_{4t} more likely than any other technique applied till now to the general case.

This will be the goal of our work in the near future.

References

1. Álvarez, V., Armario, J.A., Frau, M.D., Real, P.: A genetic algorithm for cocyclic Hadamard matrices. In: Fossorier, M.P.C., Imai, H., Lin, S., Poli, A. (eds.) AAECC 2006. LNCS, vol. 3857, pp. 144–153. Springer, Heidelberg (2006)
2. Álvarez, V., Armario, J.A., Frau, M.D., Real, P.: A system of equations for describing cocyclic Hadamard matrices. Journal of Comb. Des. 16(4), 276–290 (2008)
3. Álvarez, V., Armario, J.A., Frau, M.D., Real, P.: The homological reduction method for computing cocyclic Hadamard matrices. J. Symb. Comput. 44, 558–570 (2009)
4. Álvarez, V., Frau, M.D., Osuna, A.: A genetic algorithm with guided reproduction for constructing cocyclic Hadamard matrices. In: ICANNGA 2009 (2009) (sent)
5. Baliga, A., Chua, J.: Self-dual codes using image resoration techniques. In: Bozta, S., Sphparlinski, I. (eds.) AAECC 2001. LNCS, vol. 2227, pp. 46–56. Springer, Heidelberg (2001)
6. Flannery, D.L.: Cocyclic Hadamard matrices and Hadamard groups are equivalent. J. Algebra 192, 749–779 (1997)
7. Hedayat, A., Wallis, W.D.: Hadamard Matrices and Their Applications. Ann. Stat. 6, 1184–1238 (1978)
8. Horadam, K.J.: Hadamard matrices and their applications. Princeton University Press, Princeton (2006)
9. Horadam, K.J., de Launey, W.: Cocyclic development of designs. J. Algebraic Combin. 2(3), 267–290 (1993); Erratum: J. Algebraic Combin. 3(1), 129 (1994)
10. Horadam, K.J., de Launey, W.: Generation of cocyclic Hadamard matrices. In: Computational algebra and number theory, pp. 279–290. Math. Appl., Kluwer Acad. Publ, Dordrecht (1995)
11. Ito, N.: Hadamard Graphs I. Graphs Combin. 1(1), 57–64 (1985)
12. Ito, N.: On Hadamard Groups. J. Algebra 168, 981–987 (1994)

Interesting Examples on Maximal Irreducible Goppa Codes

Marta Giorgetti

Dipartimento di Fisica e Matematica, Universita' dell'Insubria

Abstract. A full categorization of irreducible classical Goppa codes of degree 4 and length 9 is given: it is an interesting example in the context of finding an upper bound for the number of Goppa codes which are permutation non-equivalent and irreducible and maximal with fixed parameters q, n and r ($\mathbb{F}_q$ is the field of the Goppa code, the Goppa polynomial has coefficients in $\mathbb{F}_{q^n}$ and its degree is r) using group theory techniques.

1 The Number of Non-equivalent Goppa Codes

Definition 1. *Let $g(x) = \sum g_i x^i \in \mathbb{F}_{q^n}[x]$ and let $L = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_N\}$ denote a subset of elements of $\mathbb{F}_{q^n}$ which are not roots of $g(x)$. Then the* ***Goppa code*** *$\mathcal{G}(L, g)$ is defined as the set of all vectors $c = (c_1, c_2, \ldots, c_N)$ with components in $\mathbb{F}_q$ which satisfy the condition: $\sum_{i=0}^{N} \frac{c_i}{x-\varepsilon_i} \equiv 0 \mod g(x)$.*

If $L = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_N\} = \mathbb{F}_{q^n}$ the Goppa code is said *maximal*; if the degree of $g(x)$ is r, then the Goppa code is called a Goppa code of degree r; if $g(x)$ is irreducible over $\mathbb{F}_{q^n}$ the Goppa code is called *irreducible* . In this case, a parity check matrix for $\mathcal{G}(L, g)$ can be $H_\alpha = \left(\frac{1}{\alpha-\varepsilon}, \frac{1}{\alpha-\varepsilon^2}, \ldots, \frac{1}{\alpha-\varepsilon^{q^n-1}}, \frac{1}{\alpha}\right)$, where $\alpha \in \mathbb{F}_{q^{nr}}$ is a root of $g(x)$ and $\langle\varepsilon\rangle = \mathbb{F}^*_{q^n}$. We denote the Goppa code $\mathcal{G}(L, g)$ as $C(\alpha)$ when a parity check matrix of type H_α is considered. It is important to stress that by using parity check matrix H_α to define $\mathcal{G}(L, g)$ we implicitly fix an order in L. We observe that the Goppa code $\mathcal{G}(L, g)$ is the subfield subcode of code over $\mathbb{F}_{q^{nr}}$ having as parity check matrix H_α. We denote by $\Omega = \Omega(q, n, r)$ the set of Goppa codes, with fixed parameters q, n, r, $\mathbb{S} = \mathbb{S}(q, n, r)$ the set of all elements in $\mathbb{F}_{q^{nr}}$ of degree r over $\mathbb{F}_{q^n}$ and $\mathbb{P} = \mathbb{P}(q, n, r)$ the set of irreducible polynomials of degree r in $\mathbb{F}_{q^n}[x]$.

In [4] the action on Ω is obtained by considering an action on $\mathbb{S}$ of an "semi-affine" group $T = AGL(1, q^n)\langle\sigma\rangle$ in the following way: for $\alpha \in \mathbb{S}$ and $t \in T$, $\alpha^t = a\alpha^{q^i} + b$ for some $a, b \in \mathbb{F}_{q^n}$, $a \neq 0$ and $i = 1 \ldots nr$. The action gives a number of orbits over $\mathbb{S}$ which is an upper bound for the number of non equivalent Goppa codes. An important result of [4] is the following:

Theorem 1. *[4] If $\alpha, \beta \in \mathbb{S}$ are related as it follows $\beta = \zeta\alpha^{q^i} + \xi$ for some $\zeta, \xi \in \mathbb{F}_{q^n}$, $\zeta \neq 0, i = 1 \ldots nr$, then $C(\alpha)$ is equivalent to $C(\beta)$.*

M. Bras-Amorós and T. Høholdt (Eds.): AAECC 2009, LNCS 5527, pp. 215–218, 2009.

In [2] the action of a group FG isomorphic to $A\Gamma L(1, q^n)$ on the q^n columns of the parity check matrix H_α is considered. We point out that columns of H_α are in bijective correspondence with the elements of $\mathbb{F}_{q^n}$. The group FG induces on Ω the same orbits which arise from the action introduced in [4]. This action does not describe exactly the orbits of permutation non equivalent Goppa codes, since in some cases the number of permutation non-equivalent Goppa codes is less than the number of orbits of T on $\mathbb{S}$.

The group FG acts faithfully on the columns of H_α: it can be seen as a subgroup of the symmetric group S_{q^n}. In [2] it has been proved that there exists exactly one maximal subgroup M (isomorphic to $AGL(nm, p)$) of S_{q^n} (A_{q^n}) containing FG ($q = p^m$). This suggests that one could consider the action of M on codes to reach the right bound. From this result one could hope that, when it is not possible to reach the exact number s of permutation non-equivalent Goppa codes by the action of FG, s is obtained by considering the group $AGL(nm, p)$. Unfortunately, this is not always true as it is shown in the next section. The following examples were introduced by Ryan in this PhD thesis [4]. In the next, we thoroughly analyze them, pointing out the group action of $AGL(nm, p)$.

Classification of $\Omega(3, 2, 4)$. Let $q = 3$, $\mathsf{n} = 2$, $r = 4$; let ε be a primitive element of $\mathbb{F}_{3^2}$ with minimal polynomial $x^2 + 2x + 2$; let $L = [\varepsilon, \varepsilon^2, \ldots, \varepsilon^7, 1, 0]$; let $\mathbb{P} = \mathbb{P}(3, 2, 4)$ be the set of all irreducible polynomials of degree 4 in $\mathbb{F}_9$, $|\mathbb{P}| = 1620$ and let $\mathbb{S} = \mathbb{S}(3, 2, 4)$ be the set of all elements of degree 4 over $\mathbb{F}_9$, $|\mathbb{S}| = 6480$. Let $\Gamma(g, L)$ be a maximal irreducible Goppa code of length 9 over $\mathbb{F}_3$, $g \in \mathbb{P}$. We denote by $S_{\mathbb{S}}$ the symmetric group on $\mathbb{S}$. We consider the action of T on $\mathbb{S}$: it gives 13 orbits on $\mathbb{S}$. It means that there are at most 13 classes of maximal irreducible Goppa codes. We choose a representative for each class.

Table 1 shows the thirteen classes: for each representative code Γ_i, we give the corresponding Goppa polynomial $g_i(x)$, the code parameters $[n, k, d]$ and the generator matrix M.

Table 1. Representatives of the 13 classes obtaining in the action of T over $\mathbb{S}$

Γ_i	$g_i(x)$	$[n, k, d]$	generator matrix$M$
Γ_1	$x^4 + f^3x^3 + fx + f^5$	$[9, 1, 9]$	[112212212]
Γ_2	$x^4 + f^7x^3 + x^2 + f^5x + f^3$	$[9, 1, 5]$	[010222010]
Γ_3	$x^4 + f^5x + f$	$[9, 1, 9]$	[122221112]
Γ_4	$x^4 + f^5x^2 + f^6x + f^2$	$[9, 1, 6]$	[120101202]
Γ_5	$x^4 + f^7x^2 + f^2x + f^5$	$[9, 1, 6]$	[112001011]
Γ_6	$x^4 + fx^3 + f^5x^2 + f^3x + f^6$	$[9, 1, 7]$	[001111122]
Γ_7	$x^4 + f^6x^3 + f^2x^2 + 2x + f^5$	$[9, 1, 5]$	[001220110]
Γ_8	$x^4 + 2x^3 + 2x^2 + 2x + f^7$	$[9, 1, 6]$	[121120200]
Γ_9	$x^4 + f^5x^3 + f^2x^2 + f^3$	$[9, 1, 6]$	[010021112]
Γ_{10}	$x^4 + 2x^3 + f^3x^2 + f^6$,	$[9, 1, 5]$	[120201200]
Γ_{11}	$x^4 + fx^3 + fx^2 + fx + f^2$,	$[9, 1, 7]$	[120220221]
Γ_{12}	$x^4 + f^3x^3 + f^2x^2 + 2x + f^3$,	$[9, 1, 6]$	[101012012]
Γ_{13}	$x^4 + f^3x^3 + f^5x^2 + x + f^6$	$[9, 1, 6]$	[011101202]

The analysis of parameters $[n, k, d]$ and generator matrices shows that these 13 code representantives can not be equivalent, since they have different minimum distances. We can observe that Γ_1 is permutation equivalent to Γ_3; Γ_2 and Γ_{10} are permutation equivalent to Γ_7; Γ_{11} is permutation equivalent to Γ_6; Γ_4 is permutation equivalent to Γ_8; Γ_9 are Γ_{12} are permutation equivalent to Γ_{13}. We can conclude that the number of different classes of permutation non equivalent codes is 6 and not 13 (Γ_5 composes a permutation equivalence class).

Moreover $\Gamma_5, \Gamma_4, \Gamma_8, \Gamma_9, \Gamma_{12}$ and Γ_{13} are monomially equivalent, so there are only 4 equivalence classes of non equivalent Goppa codes.

In Table 2 we summarize the results of the group actions as follows. The action of T on $\mathbb{S}$, $T \leq S_{\mathbb{S}}$, creates 13 orbits: we report the number of elements in each orbit $|\mathbb{S}^T|$ and we count the number of Goppa codes corresponding to these elements (by abuse of notation we write $|\Gamma_i^T|$). For each representative Γ_i, we consider its permutation group $\mathcal{P}(\Gamma_i)$: we obtain the number of codes permutation equivalent to it by computing $\frac{|S_9|}{|\mathcal{P}(\Gamma_i)|}$; the number of codes which are permutation equivalent to Γ_i under the actions of FG (and $AGL = AGL(2,3)$) is obtained as $\frac{|FG|}{|FG \cap \mathcal{P}(\Gamma_i)|}$ (and $\frac{|AGL|}{|ALG \cap \mathcal{P}(\Gamma_i)|}$, respectively). We use symbols ♣, ♢, ♡ and ♠ to denote the four monomial equivalence classes and symbols $\oplus$, $\odot$, $\otimes$ to denote the permutation classes when they are different from the monomial classes. We write P.E. to say Permutation Equivalent.

In this example, the action of the only maximal permutation group $AGL \leq S_{q^n}$, which contains FG, is not sufficient to unify disjoint orbits of non equivalent codes. Only the whole symmetric group S_{q^n} gives the right number of non equivalent Goppa codes.

Remark 1. It is interesting to analyzing polynomials in $\mathbb{P}$. We denote by $\mathbb{P}_{\clubsuit}$ the set of polynomials corresponding to Goppa codes in the ♣ equivalence class, and

Table 2. Different group actions

Γ_i			$\lvert\mathbb{S}^T\rvert$	$\lvert\Gamma_i^T\rvert$	$\lvert\mathcal{P}(\Gamma_i)\rvert$	$\frac{\lvert S_9\rvert}{\lvert\mathcal{P}(\Gamma_i)\rvert}$	$\frac{\lvert FG\rvert}{\lvert\mathcal{P}(\Gamma_i)\cap FG\rvert}$	$\frac{\lvert AGL\rvert}{\lvert\mathcal{P}(\Gamma_i)\cap ALG\rvert}$
Γ_1	♣		144	18	2880	126	18	54
Γ_3	♣		576	72	2880	$P.E.\Gamma_1$	72	72
Γ_2	♢		576	144	288	1260	144	432
Γ_{10}	♢		576	144	288	$P.E.\Gamma_2$	144	216
Γ_7	♢		576	144	288	$P.E.\Gamma_2$	144	432
Γ_6	♠		576	144	480	756	144	216
Γ_{11}	♠		288	72	480	$P.E.\Gamma_6$	72	108
Γ_5	♡	$\otimes$	576	144	720	504	144	216
Γ_4	♡	$\oplus$	576	144	432	840	144	432
Γ_8	♡	$\oplus$	576	144	432	$P.E.\Gamma_4$	144	216
Γ_9	♡	$\odot$	288	72	288	1260	72	108
Γ_{12}	♡	$\odot$	576	144	288	$P.E.\Gamma_9$	144	216
Γ_{13}	♡	$\odot$	576	144	288	$P.E.\Gamma_9$	144	216
			6480	1530		4746	1530	2934

so on for the others, hence $\mathbb{P} = \mathbb{P}_{\clubsuit} \cup \mathbb{P}_{\diamondsuit} \cup \mathbb{P}_{\spadesuit} \cup \mathbb{P}_{\heartsuit}$. We denote by $\mathbb{P}_{*,\Gamma_i}$, the set of polynomials in $\mathbb{P}_*$, $* \in \{\clubsuit, \diamondsuit, \heartsuit, \spadesuit\}$, corresponding to the codes in Γ_i^T. It is easy to check that if $g \in \mathbb{P}_{\clubsuit}$, g has the following shape $x^4 + \varepsilon^i x^3 + \varepsilon^j x + \varepsilon^k$ for some $\varepsilon^i, \varepsilon^j, \varepsilon^k \in \mathbb{F}_{q^n}$ and the x^2 coefficient is equal to zero. We know that $|\mathbb{P}_{\clubsuit,\Gamma_1}| = 36$ and $|\Gamma_1^T| = 18$. This means that more than one polynomial generates the same code. We can show that couples of polynomials in $\mathbb{P}_{\clubsuit,\Gamma_1}$ generate the same code. Moreover if $g_1, g_2 \in \Gamma_1^T$ generate the same Goppa code then they have the same coefficients except for the constant term: we can obtain one constant term from the other by arising to the q-th power. For example polynomials $x^4+\varepsilon^6 x^3+\varepsilon^2 x+\varepsilon^2$ and $x^4 + \varepsilon^6 x^3 + \varepsilon^2 x + \varepsilon^6$ generate the same Goppa code. A similar argument can conduce us to say that polynomials in $\mathbb{P}_{\clubsuit,\Gamma_3}$ are 576, but they generate 72 different Goppa codes. We have that 4 polynomials create the same Goppa code and we find the following relation: given a polynomial $g \in \mathbb{P}_{\clubsuit,\Gamma_3}$, $g = x^4+\varepsilon^i x^3+\varepsilon^j x+\varepsilon^k$, then the following tree polynomials generate the same Goppa code: $g' = x^4 + \varepsilon^{iq} x^3 + \varepsilon^{jq} x + \varepsilon^{kq}$, $g'' = x^4+\varepsilon^j x^3+\varepsilon^i x+\varepsilon^k$ and $g''' = x^4+\varepsilon^{jq} x^3+\varepsilon^{jq} x+\varepsilon^{kq}$. Analogous arguments can be used to describe set of polynomials in $\mathbb{P}_{\diamondsuit}$, $\mathbb{P}_{\heartsuit}$ and $\mathbb{P}_{\spadesuit}$.

Codes in $\Omega(2, 5, 6)$: let us consider the codes studied in [3]. Let $q = 2$, $n = 5$ $r = 6$ and let f be a primitive element of $\mathbb{F}_{q^n}$ with minimal polynomial x^5+x^2+1; let $L = [f, f^2, \ldots, f^{30}, 1, 0]$. We consider the following two polynomials $p_1 := x^6 + f^{22}x^5 + f^2x^4 + f^{25}x^3 + f^{10}x + f^3$ and $p_2 := x^6 + f^{20}x^5 + f^{19}x^4 + f^{19}x^3 + f^{12}x^2+f^4x+f^28$. They generate equivalent Goppa codes $\Gamma_1(L, p_1)$ and $\Gamma_2(L, p_2)$, but their roots are in different orbits under the action of T over $\mathbb{S} = \mathbb{S}(2, 5, 6)$. To know how many codes are in each orbits we take a representative code and we construct its orbit under the permutation group $FG \leq S_{32}$. We verify that the action of the maximal subgroup $AGL(2, 5)$ containing FG does not unify the two orbits. Also in this case, the only permutation group which gives the right number of non equivalent Goppa code is the whole symmetric group S_{q^n}.

References

1. Chen, C.-L.: Equivalent irreducible Goppa codes. IEEE Trans. Inf. Theory 24, 766–770 (1978)
2. Dalla Volta, F., Giorgetti, M., Sala, M.: Permutation equivalent maximal irreducible Goppa codes. Designs, Codes and Cryptography (submitted), `http://arxiv.org/abs/0806.1763`
3. Ryan, J.A., Magamba, K.: Equivalent irreducible Goppa codes and the precise number of quintic Goppa codes of length 32. In: AFRICON 2007, September 26-28, pp. 1–4 (2007)
4. Ryan, J.: Irreducible Goppa codes, Ph.D. Thesis, University College Cork, Cork, Ireland (2002)

Repeated Root Cyclic and Negacyclic Codes over Galois Rings

Sergio R. López-Permouth and Steve Szabo

Department of Mathematics, Ohio University, Athens, Ohio-45701, USA

Abstract. In this notice we describe the ideal structure of all cases of cyclic and negacyclic codes of length p^s over a Galois ring alphabet that have not yet been discussed in the literature. Unlike in the cases reported earlier in the literature by various authors, the ambient spaces here are never chain rings. These ambient rings do nonetheless share the properties of being local and having a simple socle.

Keywords: repeated root codes, cyclic codes, polynomial codes, Galois rings.

1 Introduction

While the literature on cyclic and negacylic codes over chain rings (such as Galois rings) has experienced much growth in recent years (see [1, 2, 3, 4, 5, 6]), in most instances the studies have been focused only on the cases where the characteristic of the alphabet ring is coprime to the code length, the so-called simple root codes. A few of the contributions to the study of the cases where the characteristic of the alphabet ring is not coprime to the code length (repeated root codes) are [7, 8, 9, 10, 11, 12].

In this paper we focus on the repeated root case where the code length is in fact p^s, a power of a prime p. In all papers dealing with this type of code so far, the codes correspond to principal ideals because in every case considered thus far the code ambient has been a chain ring. It turns out that in the remaining cases the code ambients are no longer chain rings and in fact, not even principal ideal rings. The authors are submitting a complete account of their research elsewhere [13]; the results announced here are proven there and methods to calculate the minimum distance of all negacyclic and cyclic codes considered are also given there.

The three cases of codes of length p^s not previously tackled in the literature are: cyclic and negacyclic codes over $GR(p^a, m)$ for odd prime p and $a > 1$ and cyclic codes over $GR(2^a, m)$ for $a > 1$. It is easy to see that the ambients $\frac{GR(p^a,m)[x]}{\langle x^{p^s}+1\rangle}$ and $\frac{GR(p^a,m)[x]}{\langle x^{p^s}-1\rangle}$ for negacyclic and cyclic codes for an odd prime p and $a > 1$ are isomorphic under the isomorphism sending x to $-x$. Hence, all the results about the lattice structure of their ideals can be done for one and translated into the other in a straightforward way. Surprisingly, the structure for the ambient ring $\frac{GR(2^a,m)[x]}{\langle x^{2^s}-1\rangle}$ of the cyclic codes over $GR(2^a, m)$ for $a > 1$

M. Bras-Amorós and T. Høholdt (Eds.): AAECC 2009, LNCS 5527, pp. 219–222, 2009.

is similar to the two other ambients in this paper and, in fact, the proofs of the corresponding results are highly parallel. For convenience, we opt to state all results on the cyclic case so all results can be stated with a single notation without restrictions on the parity of p. It is worthwhile noticing, however, that $\frac{GR(2^a,m)[x]}{\langle x^{2^s}-1\rangle}$ is not isomorphic to its negacyclic counterpart which is a chain ring [14]. One should also point out that in [13] where these results appear in their entirety including proofs, it was found to be better to work on the negacyclic case for the proofs when p is odd. In the case of $p = 2$, the results can be obtained using $x + 1$ or $x - 1$. For the reason mentioned, $x + 1$ was chosen to present the $p = 2$ case in [13] also. Here we will use $x - 1$ for the purposes of stating the results for arbitrary prime p in a concise way.

We use standard ring-theoretic terminology which we include here for the convenience of the reader. For further information, the reader may consult a standard reference such [15]. For an account focusing on finite rings, see [16]. A left module M of a ring R is *simple* if $M \neq 0$ and M has no R-submodules other than (0) and M. The *socle* of a ring R denoted by $soc(R)$, is the sum of of all minimal left ideals of R. The *Jacobson Radical* of ring R denoted by $J(R)$, is the intersection of all maximal left ideals of R. A *chain ring* is a ring whose ideals are linearly ordered by inclusion. Since all rings in this paper are commutative, the use of left modules, left ideals, etc. is unnecessary in our context and the reader may simply ignore the word "left" in each definition."

2 The Main Results

The following results hold for ambient rings $\frac{GR(p^a,m)[x]}{\langle x^{p^s}-1\rangle}$ for integers $a > 1$, $s > 0$ and p an arbitrary prime.

Proposition 1. *In* $\frac{GR(p^a,m)[x]}{\langle x^{p^s}-1\rangle}$, *the element* $(x - 1)$ *is nilpotent.*

Proposition 2. *The ambient ring* $\frac{GR(p^a,m)[x]}{\langle x^{p^s}-1\rangle}$ *is a local ring with radical* $J\left(\frac{GR(p^a,m)[x]}{\langle x^{p^s}-1\rangle}\right) = \langle p, x - 1\rangle$.

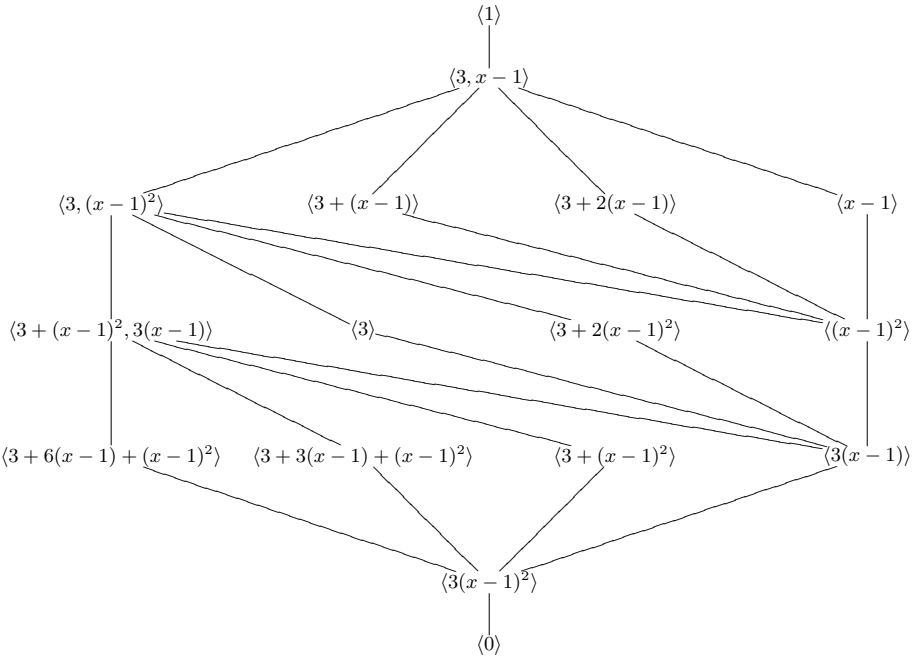
Proposition 3. *The socle* $soc\left(\frac{GR(p^a,m)[x]}{\langle x^{p^s}-1\rangle}\right)$ *of* $\frac{GR(p^a,m)[x]}{\langle x^{p^s}-1\rangle}$ *is the simple module* $\langle p^{a-1}(x-1)^{(p^s-1)}\rangle$.

Proposition 4. *In the ambient ring* $\frac{GR(p^a,m)[x]}{\langle x^{p^s}-1\rangle}$

1. $p \notin \langle x - 1\rangle$
2. $x - 1 \notin \langle p\rangle$
3. $\frac{GR(p^a,m)[x]}{\langle x^{p^s}-1\rangle}$ *is not a chain ring*
4. $\langle p, x - 1\rangle$ *is not a principal ideal*

Theorem 1. *The ambient ring* $\frac{GR(p^a,m)[x]}{\langle x^{p^s}-1\rangle}$ *is a finite local ring with simple socle that is not a chain ring.*

Example 1. To illustrate Theorem 1, we provide the following figure. It shows the ideal lattice of $\frac{Z_{3^2}[x]}{\langle x^3-1\rangle}$. Notice that the radical is $\langle 3, x-1\rangle$ and the socle is $\langle 3(x-1)^2\rangle$. More importantly, we see that the ring is not a chain ring.



3 Conclusion

The results in this paper shed light on a more general class of codes and their ambient structures, namely polynomial codes over Galois rings. It was shown here that such codes are not always principal. This abridged version of [13] should serve as insight for another paper by the authors which is in preparation at this time [17]. In that paper, the ambient rings for polynomial codes over Galois rings are studied which is the natural next step for the research presented here.

References

1. Calderbank, A.R., Sloane, N.J.A.: Modular and p-adic cyclic codes. Des. Codes Cryptogr. 6(1), 21–35 (1995)
2. Kanwar, P., López-Permouth, S.R.: Cyclic codes over the integers modulo p^m. Finite Fields Appl. 3(4), 334–352 (1997)
3. Kiah, H.M., Leung, K.H., Ling, S.: Cyclic codes over $\mathrm{GR}(p^2, m)$ of length p^k. Finite Fields Appl 14(3), 834–846 (2008)

4. Pless, V.S., Qian, Z.: Cyclic codes and quadratic residue codes over Z_4. IEEE Trans. Inform. Theory 42(5), 1594–1600 (1996)
5. Sălăgean, A.: Repeated-root cyclic and negacyclic codes over a finite chain ring. Discrete Appl. Math. 154(2), 413–419 (2006)
6. Wolfmann, J.: Negacyclic and cyclic codes over Z_4. IEEE Trans. Inform. Theory 45(7), 2527–2532 (1999)
7. Abualrub, T., Oehmke, R.: Cyclic codes of length 2^e over Z_4. Discrete Appl. Math. 128(1), 3–9 (2003); International Workshop on Coding and Cryptography (WCC 2001), Paris (2001)
8. Abualrub, T., Oehmke, R.: On the generators of $\mathbb{Z}_4$ cyclic codes of length 2^e. IEEE Trans. Inform. Theory 49(9), 2126–2133 (2003)
9. Castagnoli, G., Massey, J.L., Schoeller, P.A., von Seemann, N.: On repeated-root cyclic codes. IEEE Trans. Inform. Theory 37(2), 337–342 (1991)
10. Dougherty, S.T., Park, Y.H.: On modular cyclic codes. Finite Fields Appl. 13(1), 31–57 (2007)
11. Özadam, H., Özbudak, F.: A note on negacyclic and cyclic codes of length p^s over a finite field of characteristic p (2009) (preprint)
12. van Lint, J.H.: Repeated-root cyclic codes. IEEE Trans. Inform. Theory 37(2), 343–345 (1991)
13. López-Permouth, S.R., Szabo, S.: On the Hamming weight of repeated root cyclic and negacyclic codes over Galois rings. arXiv:0903.2791v1 (submitted)
14. Dinh, H.Q.: Negacyclic codes of length 2^s over Galois rings. IEEE Trans. Inform. Theory 51(12), 4252–4262 (2005)
15. Lam, T.Y.: A first course in noncommutative rings, 2nd edn. Graduate Texts in Mathematics, vol. 131. Springer, New York (2001)
16. McDonald, B.R.: Finite rings with identity. Pure and Applied Mathematics, vol. 28. Marcel Dekker Inc., New York (1974)
17. López-Permouth, S.R., Szabo, S.: Polynomial codes over Galois rings (in preparation)

Construction of Additive Reed-Muller Codes*

J. Pujol, J. Rifà, and L. Ronquillo

Department of Information and Communications Engineering, Universitat Autònoma de Barcelona, 08193-Bellaterra, Spain

Abstract. The well known Plotkin construction is, in the current paper, generalized and used to yield new families of $\mathbb{Z}_2\mathbb{Z}_4$-additive codes, whose length, dimension as well as minimum distance are studied. These new constructions enable us to obtain families of $\mathbb{Z}_2\mathbb{Z}_4$-additive codes such that, under the Gray map, the corresponding binary codes have the same parameters and properties as the usual binary linear Reed-Muller codes. Moreover, the first family is the usual binary linear Reed-Muller family.

Keywords: $\mathbb{Z}_2\mathbb{Z}_4$-Additive codes, Plotkin construction, Reed-Muller codes, $\mathbb{Z}_2\mathbb{Z}_4$-linear codes.

1 Introduction

The aim of our paper is to obtain a generalization of the Plotkin construction which gave rise to families of $\mathbb{Z}_2\mathbb{Z}_4$-additive codes such that, after the Gray map, the corresponding $\mathbb{Z}_2\mathbb{Z}_4$-linear codes had the same parameters and properties as the family of binary linear RM codes. Even more, we want the corresponding codes with parameters $(r, m) = (1, m)$ and $(r, m) = (m-2, m)$ to be, respectively, any one of the non-equivalent $\mathbb{Z}_2\mathbb{Z}_4$-linear Hadamard and $\mathbb{Z}_2\mathbb{Z}_4$-linear 1-perfect codes.

2 Constructions of $\mathbb{Z}_2\mathbb{Z}_4$-Additive Codes

In general, any non-empty subgroup $\mathcal{C}$ of $\mathbb{Z}_2^\alpha \times \mathbb{Z}_4^\beta$ is a $\mathbb{Z}_2\mathbb{Z}_4$-*additive code*, where $\mathbb{Z}_2^\alpha$ denotes the set of all binary vectors of length α and $\mathbb{Z}_4^\beta$ is the set of all β-tuples in $\mathbb{Z}_4$.

Let $\mathcal{C}$ be a $\mathbb{Z}_2\mathbb{Z}_4$-additive code, and let $C = \Phi(\mathcal{C})$, where $\Phi : \mathbb{Z}_2^\alpha \times \mathbb{Z}_4^\beta \longrightarrow \mathbb{Z}_2^n$ is given by the map $\Phi(u_1, \ldots, u_\alpha | v_1, \ldots, v_\beta) = (u_1, \ldots, u_\alpha | \phi(v_1), \ldots, \phi(v_\beta))$ where $\phi(0) = (0, 0)$, $\phi(1) = (0, 1)$, $\phi(2) = (1, 1)$, and $\phi(3) = (1, 0)$ is the usual Gray map from $\mathbb{Z}_4$ onto $\mathbb{Z}_2^2$.

Since the Gray map is distance preserving, the Hamming distance of a $\mathbb{Z}_2\mathbb{Z}_4$-linear code C coincides with the Lee distance computed on the $\mathbb{Z}_2\mathbb{Z}_4$-additive code $\mathcal{C} = \phi^{-1}(C)$.

* This work has been partially supported by the Spanish MICINN Grants MTM2006-03250, TSI2006-14005-C02-01, PCI2006-A7-0616 and also by the *Comissionat per a Universitats i Recerca de la Generalitat de Catalunya* under grant FI2008.

M. Bras-Amorós and T. Høholdt (Eds.): AAECC 2009, LNCS 5527, pp. 223–226, 2009.

A $\mathbb{Z}_2\mathbb{Z}_4$-additive code $\mathcal{C}$ is also isomorphic to an abelian structure like $\mathbb{Z}_2^\gamma \times \mathbb{Z}_4^\delta$. Therefore, $\mathcal{C}$ has $|\mathcal{C}| = 2^\gamma 4^\delta$ codewords and, moreover, $2^{\gamma+\delta}$ of them are of order two. We call such code $\mathcal{C}$ a $\mathbb{Z}_2\mathbb{Z}_4$-*additive code of type* $(\alpha, \beta; \gamma, \delta)$ and its binary image $C = \Phi(\mathcal{C})$ is a $\mathbb{Z}_2\mathbb{Z}_4$-*linear code of type* $(\alpha, \beta; \gamma, \delta)$.

Although $\mathcal{C}$ may not have a basis, it is important and appropriate to define a generator matrix for $\mathcal{C}$ as:

$$\mathcal{G} = \left(\begin{array}{c|c} B_2 & Q_2 \\ \hline B_4 & Q_4 \end{array}\right), \tag{1}$$

where B_2 and B_4 are binary matrices of size $\gamma \times \alpha$ and $\delta \times \alpha$, respectively; Q_2 is a $\gamma \times \beta$-quaternary matrix which contains order two row vectors; and Q_4 is a $\delta \times \beta$-quaternary matrix with order four row vectors.

2.1 Plotkin Construction

In this section we show that the well known Plotkin construction can be generalized to $\mathbb{Z}_2\mathbb{Z}_4$-additive codes.

Definition 1 (Plotkin Construction). *Let $\mathcal{X}$ and $\mathcal{Y}$ be any two $\mathbb{Z}_2\mathbb{Z}_4$-additive codes of types $(\alpha, \beta; \gamma_\mathcal{X}, \delta_\mathcal{X})$, $(\alpha, \beta; \gamma_\mathcal{Y}, \delta_\mathcal{Y})$ and minimum distances $d_\mathcal{X}$, $d_\mathcal{Y}$, respectively. If $\mathcal{G}_\mathcal{X}$ and $\mathcal{G}_\mathcal{Y}$ are the generator matrices of $\mathcal{X}$ and $\mathcal{Y}$, then the matrix*

$$\mathcal{G}_P = \begin{pmatrix} \mathcal{G}_\mathcal{X} & \mathcal{G}_\mathcal{X} \\ 0 & \mathcal{G}_\mathcal{Y} \end{pmatrix}$$

is the generator matrix of a new $\mathbb{Z}_2\mathbb{Z}_4$-additive code $\mathcal{C}$.

Proposition 2. *Code $\mathcal{C}$ defined above is a $\mathbb{Z}_2\mathbb{Z}_4$-additive code of type $(2\alpha, 2\beta; \gamma, \delta)$, where $\gamma = \gamma_\mathcal{X} + \gamma_\mathcal{Y}$, $\delta = \delta_\mathcal{X} + \delta_\mathcal{Y}$, binary length $n = 2\alpha + 4\beta$, size $2^{\gamma+2\delta}$ and minimum distance $d = \min\{2d_\mathcal{X}, d_\mathcal{Y}\}$.*

2.2 BA-Plotkin Construction

Applying two Plotkin constructions, one after another, but slightly changing the submatrices in the generator matrix, we obtain a new construction with interesting properties with regard to the minimum distance of the generated code. We call this new construction *BA-Plotkin construction.*

Given a $\mathbb{Z}_2\mathbb{Z}_4$-additive code $\mathcal{C}$ with generator matrix $\mathcal{G}$ we denote, respectively, by $\mathcal{G}[b_2]$, $\mathcal{G}[q_2]$, $\mathcal{G}[b_4]$ and $\mathcal{G}[q_4]$ the four submatrices B_2, Q_2, B_4, Q_4 of $\mathcal{G}$ defined in (1); and by $\mathcal{G}[b]$ and $\mathcal{G}[q]$ the submatrices of $\mathcal{G}$, $\left(\begin{array}{c|} B_2 \\ \hline B_4 \end{array}\right)$, $\left(\begin{array}{|c} Q_2 \\ \hline Q_4 \end{array}\right)$, respectively.

Definition 3 (BA-Plotkin Construction). *Let $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$ be any three $\mathbb{Z}_2\mathbb{Z}_4$-additive codes of types $(\alpha, \beta; \gamma_\mathcal{X}, \delta_\mathcal{X})$, $(\alpha, \beta; \gamma_\mathcal{Y}, \delta_\mathcal{Y})$, $(\alpha, \beta; \gamma_\mathcal{Z}, \delta_\mathcal{Z})$ and minimum distances $d_\mathcal{X}$, $d_\mathcal{Y}$, $d_\mathcal{Z}$, respectively. Let $\mathcal{G}_\mathcal{X}$, $\mathcal{G}_\mathcal{Y}$ and $\mathcal{G}_\mathcal{Z}$ be the generator matrices of the $\mathbb{Z}_2\mathbb{Z}_4$-additive codes $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$, respectively. We define a new code $\mathcal{C}$ as the $\mathbb{Z}_2\mathbb{Z}_4$-additive code generated by*

$$\mathcal{G}_{BA} = \left(\begin{array}{cc|cccccc} \mathcal{G}_{\mathcal{X}}[b] & \mathcal{G}_{\mathcal{X}}[b] & 2\mathcal{G}_{\mathcal{X}}[b] & \mathcal{G}_{\mathcal{X}}[q] & \mathcal{G}_{\mathcal{X}}[q] & \mathcal{G}_{\mathcal{X}}[q] & \mathcal{G}_{\mathcal{X}}[q] \\ 0 & \mathcal{G}_{\mathcal{Y}}[b_2] & \mathcal{G}_{\mathcal{Y}}[b_2] & 0 & 2\mathcal{G}'_{\mathcal{Y}}[q_2] & \mathcal{G}'_{\mathcal{Y}}[q_2] & 3\mathcal{G}'_{\mathcal{Y}}[q_2] \\ 0 & \mathcal{G}_{\mathcal{Y}}[b_4] & \mathcal{G}_{\mathcal{Y}}[b_4] & 0 & \mathcal{G}_{\mathcal{Y}}[q_4] & 2\mathcal{G}_{\mathcal{Y}}[q_4] & 3\mathcal{G}_{\mathcal{Y}}[q_4] \\ \mathcal{G}_{\mathcal{Y}}[b_4] & \mathcal{G}_{\mathcal{Y}}[b_4] & 0 & 0 & 0 & \mathcal{G}_{\mathcal{Y}}[q_4] & \mathcal{G}_{\mathcal{Y}}[q_4] \\ 0 & \mathcal{G}_{\mathcal{Z}}[b] & 0 & 0 & 0 & 0 & \mathcal{G}_{\mathcal{Z}}[q] \end{array}\right),$$

where $\mathcal{G}'_{\mathcal{Y}}[q_2]$ is the matrix obtained from $\mathcal{G}_{\mathcal{Y}}[q_2]$ after switching twos by ones in its $\gamma_{\mathcal{Y}}$ rows of order two, and considering the ones from the third column of the construction as ones in the quaternary ring $\mathbb{Z}_4$.

Proposition 4. *Code $\mathcal{C}$ defined above is a $\mathbb{Z}_2\mathbb{Z}_4$-additive code of type $(2\alpha, \alpha + 4\beta; \gamma, \delta)$ where $\gamma = \gamma_{\mathcal{X}} + \gamma_{\mathcal{Z}}$, $\delta = \delta_{\mathcal{X}} + \gamma_{\mathcal{Y}} + 2\delta_{\mathcal{Y}} + \delta_{\mathcal{Z}}$, binary length $n = 4\alpha + 8\beta$, size $2^{\gamma+2\delta}$ and minimum distance $d = \min\{4d_{\mathcal{X}}, 2d_{\mathcal{Y}}, d_{\mathcal{Z}}\}$.*

3 Additive Reed-Muller Codes

We will refer to $\mathbb{Z}_2\mathbb{Z}_4$-additive Reed-Muller codes as $\mathcal{ARM}$. Just as there is only one RM family in the binary case, in the $\mathbb{Z}_2\mathbb{Z}_4$-additive case there are $\lfloor \frac{m+2}{2} \rfloor$ families for each value of m. Each one of these families will contain any of the $\lfloor \frac{m+2}{2} \rfloor$ non-isomorphic $\mathbb{Z}_2\mathbb{Z}_4$-linear extended perfect codes which are known to exist for any m [1].

We will identify each family $\mathcal{ARM}_s(r, m)$ by a subindex $s \in \{0, \ldots, \lfloor \frac{m}{2} \rfloor\}$.

3.1 The Families of $\mathcal{ARM}(r, 1)$ and $\mathcal{ARM}(r, 2)$ Codes

We start by considering the case $m = 1$, that is the case of codes of binary length $n = 2^1$. The $\mathbb{Z}_2\mathbb{Z}_4$-additive Reed-Muller code $\mathcal{ARM}(0, 1)$ is the repetition code, of type $(2, 0; 1, 0)$ and which only has one nonzero codeword (the vector with only two binary coordinates of value 1). The code $\mathcal{ARM}(1, 1)$ is the whole space $\mathbb{Z}_2^2$, thus a $\mathbb{Z}_2\mathbb{Z}_4$-additive code of type $(2, 0; 2, 0)$. Both codes $\mathcal{ARM}(0, 1)$ and $\mathcal{ARM}(1, 1)$ are binary codes with the same parameters and properties as the corresponding binary $RM(r, 1)$ codes (see [2]). We will refer to them as $\mathcal{ARM}_0(0, 1)$ and $\mathcal{ARM}_0(1, 1)$, respectively.

The generator matrix of $\mathcal{ARM}_0(0, 1)$ is $\mathcal{G}_0(0, 1) = \left(\begin{array}{cc} 1 & 1 \end{array}\right)$ and the generator matrix of $\mathcal{ARM}_0(1, 1)$ is $\mathcal{G}_0(1, 1) = \left(\begin{array}{cc} 1 & 1 \\ 0 & 1 \end{array}\right)$.

For $m = 2$ we have two families, $s = 0$ and $s = 1$, of additive Reed-Muller codes of binary length $n = 2^2$. The family $\mathcal{ARM}_0(r, 2)$ consists of binary codes obtained from applying the Plotkin construction defined in Proposition 2 to the family $\mathcal{ARM}_0(r, 1)$. For $s = 1$, we define $\mathcal{ARM}_1(0, 2)$, $\mathcal{ARM}_1(1, 2)$ and $\mathcal{ARM}_1(2, 2)$ as the codes with generator matrices $\mathcal{G}_1(0, 2) = \left(\begin{array}{cc|c} 1 & 1 & 2 \end{array}\right)$, $\mathcal{G}_1(1, 2) = \left(\begin{array}{cc|c} 1 & 1 & 2 \\ 0 & 1 & 1 \end{array}\right)$ and $\mathcal{G}_1(2, 2) = \left(\begin{array}{cc|c} 1 & 1 & 2 \\ 0 & 1 & 0 \\ \hline 0 & 1 & 1 \end{array}\right)$, respectively.

3.2 Plotkin and BA-Plotkin Constructions

Take the family $\mathcal{ARM}_s$ and let $\mathcal{ARM}_s(r, m-1)$, $\mathcal{ARM}_s(r-1, m-1)$ and $\mathcal{ARM}_s(r-2, m-1)$, $0 \leq s \leq \lfloor \frac{m-1}{2} \rfloor$, be three consecutive codes with parameters $(\alpha, \beta; \gamma', \delta')$, $(\alpha, \beta; \gamma'', \delta'')$ and $(\alpha, \beta; \gamma''', \delta''')$; binary length $n = 2^{m-1}$; minimum distances 2^{m-r-1}, 2^{m-r} and 2^{m-r+1}; and generator matrices $\mathcal{G}_s(r, m-1)$, $\mathcal{G}_s(r-1, m-1)$ and $\mathcal{G}_s(r-2, m-1)$, respectively. By using Proposition 2 and Proposition 4 we can prove the following results:

Theorem 5. *For any r and $m \geq 2$, $0 < r < m$, code $\mathcal{ARM}_s(r, m)$ obtained by applying the Plotkin construction from Definition 1 on codes $\mathcal{ARM}_s(r, m-1)$ and $\mathcal{ARM}_s(r-1, m-1)$ is a $\mathbb{Z}_2\mathbb{Z}_4$-additive code of type $(2\alpha, 2\beta; \gamma, \delta)$, where $\gamma = \gamma' + \gamma''$ and $\delta = \delta' + \delta''$; binary length $n = 2^m$; size 2^k codewords, where $k = \sum_{i=0}^{r} \binom{m}{i}$; minimum distance 2^{m-r} and $\mathcal{ARM}_s(r-1, m) \subset \mathcal{ARM}_s(r, m)$.*

We consider $\mathcal{ARM}_s(0, m)$ to be the repetition code with only one nonzero codeword (the vector with 2α ones and 2β twos) and $\mathcal{ARM}_s(m, m)$ be the whole space $\mathbb{Z}_2^{2\alpha} \times \mathbb{Z}_4^{2\beta}$.

Theorem 6. *For any r and $m \geq 3$, $0 < r < m$, $s > 0$, use the BA-Plotkin construction from Definition 3, where generator matrices $\mathcal{G}_\mathcal{X}$, $\mathcal{G}_\mathcal{Y}$, $\mathcal{G}_\mathcal{Z}$ stand for $\mathcal{G}_s(r, m-1)$, $\mathcal{G}_s(r-1, m-1)$ and $\mathcal{G}_s(r-2, m-1)$, respectively, to obtain a new $\mathbb{Z}_2\mathbb{Z}_4$-additive $\mathcal{ARM}_{s+1}(r, m+1)$ code of type $(2\alpha, \alpha + 4\beta; \gamma, \delta)$, where $\gamma = \gamma' + \gamma'''$, $\delta = \delta' + \gamma'' + 2\delta'' + \delta'''$; binary length $n = 2^{m+1}$; 2^k codewords, where $k = \sum_{i=0}^{r} \binom{m+1}{i}$, minimum distance 2^{m-r+1} and, moreover, $\mathcal{ARM}_{s+1}(r-1, m+1) \subset \mathcal{ARM}_{s+1}(r, m+1)$.*

To be coherent with all notations, code $\mathcal{ARM}_{s+1}(-1, m+1)$ is defined as the all zero codeword code, code $\mathcal{ARM}_{s+1}(0, m+1)$ is defined as the repetition code with only one nonzero codeword (the vector with 2α ones and $\alpha + 4\beta$ twos), whereas codes $\mathcal{ARM}_{s+1}(m, m+1)$ and $\mathcal{ARM}_{s+1}(m+1, m+1)$ are defined as the even Lee weight code and the whole space $\mathbb{Z}_2^{2\alpha} \times \mathbb{Z}_4^{\alpha+4\beta}$, respectively.

Using both Theorem 5 and Theorem 6 we can now construct all $\mathcal{ARM}_s(r, m)$ codes for $m > 2$. Once applied the Gray map, all these codes give rise to binary codes with the same parameters and properties as the RM codes. Moreover, when $m = 2$ or $m = 3$, they also have the same codewords.

References

1. Borges, J., Rifà, J.: A characterization of 1-perfect additive codes. IEEE Trans. Inform. Theory 45(5), 1688–1697 (1999)
2. MacWilliams, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes. North-Holland Publishing Company, Amsterdam (1977)

Gröbner Representations of Binary Matroids*

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, Castilla, Spain
edgar@maf.uva.es

Abstract. Several constructions in binary linear block codes are also related to matroid theory topics. These constructions rely on a given order in the ground set of the matroid. In this paper we define the Gröbner representation of a binary matroid and we show how it can be used for studying different sets bases, cycles, activity intervals, etc.

1 Introduction

In this work we provide a representation of a binary matroid. We have tried to keep "Gröbner machinery" [11] out of the paper for those readers non familiar with it. Effective methods for computing the Gröbner representation that heavily depend of Gröbner basis techniques can be found in other papers of the authors [1,2,3,4]. The starting point of the research in Gröbner representations can be found in [5,11,12,16].

Let E be a finite set called *ground set* and $\mathcal{I} \subset 2^E$. We say that $\mathcal{M} = (E, \mathcal{I})$ is a *matroid* if the following conditions are satisfied: $(i)\, \emptyset \in \mathcal{I}$, (ii) If $A \in \mathcal{I}$ and $B \subseteq A$, then $B \in \mathcal{I}$ and (iii) If $A, B \in \mathcal{I}$ and $|A| = |B| + 1$, then there is an element $x \in A \setminus B$ so that $B \cup \{x\} \in \mathcal{I}$.

Let $\mathbb{F}_2$ be the finite field with 2 elements and G be a binary matrix with column index set E. Let $\mathcal{I}$ be the collection of sets $I \subseteq E$ such that the column submatrix indexed by I has $\mathbb{F}_2$ independent columns. If we consider $\emptyset \in \mathcal{I}$ then $\mathcal{M} = (E, \mathcal{I})$ is the *binary matroid* with ground set E and independent sets $\mathcal{I}$. A *base* B of $\mathcal{M}$ is a maximum cardinality set $B \in \mathcal{I}$ and a *circuit* C of $\mathcal{M}$ is a subset of E that indexes a $\mathbb{F}_2$-*minimal dependent* column submatrix of G. We call *cycles* to all the dependent sets (minimal and non minimal). G can be seen as a generator matrix of a code. From now on we consider only binary matrices that provide *projective codes*, i.e. those that any two columns of the generator matrix are different (equivalently, the dual code has minimum weight at least 3). Circuits (respectively cycles) of the matroid defined by G are in one to one correspondence with the minimum weight codewords (respectively codewords)

* Partially funded by "Agencia Española de Cooperación Internacional", Project A/016959/08.

** Partially funded by "Junta de Castilla y León" VA65A07 and "Ministerio Educación y Ciencia" I+D/I+D+I MTM2007-66842, MTM2007-64704.

M. Bras-Amorós and T. Høholdt (Eds.): AAECC 2009, LNCS 5527, pp. 227–230, 2009.

of the dual code. In matroids the greedy algorithm always succeeds in choosing a base of minimum weight, formally

Proposition 1. *Let $\mathcal{M} = (E, \mathcal{I})$ a matroid where the elements of E are totally ordered, then*

1. *there is a base $A = \{a_1, a_2, \ldots, a_k\}$ with $a_1 < a_2 < \ldots < a_k$, called the* first basis*, such that for any other base $X = \{x_1, x_2, \ldots, x_k\}$ with $x_1 < x_2 < \ldots < x_k$ we have that $a_i \leq x_i$.*
2. *there is a base $B = \{b_1, b_2, \ldots, b_k\}$ with $b_1 < b_2 < \ldots < b_k$, called the* last basis*, such that for any other base $X = \{x_1, x_2, \ldots, x_k\}$ with $x_1 < x_2 < \ldots < x_k$ we have that $b_i \geq x_i$.*

The first base is computed by a greedy algorithm, $\{a_1\}$ is the smallest independent set of size 1, then a_2 is the smallest element such that $\{a_1, a_2\}$ is independent, and so on. The last base is built in a similar way.

2 A Gröbner Representation of a Binary Matroid

Let $\{\mathbf{e}_i\}_{i=1}^n$ be the canonical basis of $\mathbb{F}_2^n$ (ie. $\mathbf{e}_i$ is the vector with 1 in position i and 0 elsewhere). We consider the following total order on the elements of $\mathbb{F}_2^n$. Let $\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^n$ we say that $\mathbf{x} \prec_e \mathbf{y}$ if $|\mathrm{supp}(\mathbf{x})| \leq |\mathrm{supp}(\mathbf{y})|$, or, if $|\mathrm{supp}(\mathbf{x})| = |\mathrm{supp}(\mathbf{y})|$ then $\mathbf{x} \prec \mathbf{y}$ where $\prec$ is the lexicographic ordering on the vectors of $\mathbb{F}_2^n$. Consider a binary matroid and $\mathcal{C}$ its associated projective code. We define the Gröbner representation of the matroid as follows,

Canonical forms. The set N is a transversal of $\mathbb{F}_2^n/\mathcal{C}^\perp$ such that for each $\mathbf{n} \in N \setminus \{\mathbf{0}\}$ there exists $\mathbf{e}_i$ $1 \leq i \leq n$ such that $\mathbf{n} = \mathbf{n}' + \mathbf{e}_i$ and $\mathbf{n}' \in N$.

Matphi. Let $\phi : N \times \{\mathbf{e}_i\}_{i=1}^n \to N$ be the function that maps each pair $(\mathbf{n}, \mathbf{e}_i)$ to the element in N representing the coset containing $\mathbf{n} + \mathbf{e}_i$.

We call the pair N, ϕ a *Gröbner representation* of $\mathcal{M}$. Suppose we consider the ideal given by $I(\mathcal{M}) = \langle \{\mathbf{x}^\mathbf{a} - \mathbf{x}^\mathbf{b} \mid (\mathbf{a} - \mathbf{b}) \in \mathcal{C}^\perp\} \rangle \subseteq \mathbb{F}_2[x_1, x_2, \ldots, x_n]$ where (abusing the notation) we consider an element $\mathbf{a}$ in $\mathbb{F}_2^n$ as a vector $\mathbf{a} = (a_1, a_2, \ldots, a_n) \in \mathbb{Z}^n$ then $\mathbf{x}^\mathbf{a}$ represents the monomial $\prod_{i=1}^n x_i^{a_n}$. A a Gröbner representation [11] of the ideal $I(\mathcal{M})$ is given by N, ϕ. An algorithm for computing a Gröbner basis of the ideal can be found in [1]. This greedy algorithm computes the first base given N, ϕ

1. $FB \leftarrow \mathbf{0}$, $\mathbf{v} \leftarrow \mathbf{0}$.
2. **for** $i = 1, \ldots, n$
   do (**if** $\phi(\mathbf{v}, \mathbf{e}_i) \neq \mathbf{0}$ then $FB \leftarrow FB + \mathbf{e}_i$, $\mathbf{v} \leftarrow \phi(\mathbf{v}, \mathbf{e}_i)$ **endif**) **enddo**
3. Return FB.

The returned vector is a 0-1 vector where the 1 index the elements on the first basis. Same algorithm applies for the last base just changing the order in the counter **for**. One can compute a trellis for a linear code isomorphic to Muder's minimal trellis [13] based on the first and last basis, see [6].

3 Activities

Let $\mathcal{M} = (E, \mathcal{I})$ be a matroid on a linearly ordered set E, and let $A \subseteq E$. We say that an element $e \in E$ is $\mathcal{M}$-*active* with respect to A if there is a circuit γ such that $e \in \gamma \subseteq A \cup \{e\}$ and e is the smallest element of the circuit. We denote by $\mathrm{Act}_{\mathcal{M}}(A)$ the set of $\mathcal{M}$-*active elements with respect to* A.

If $\mathcal{C}$ is the binary code associated to the matroid, then a codeword $\mathbf{c} = (c_1, \ldots, c_n)$ of the code $\mathcal{C}^{\perp}$ (i.e. a cycle of the matroid) is said to have the span interval $[i, j]$ where $i \leq j$, if $c_i c_j \neq 0$, and $c_l = 0$ if $l < i$ or $j < l$. In this case the codeword $\mathbf{c}$ is said to be *sup-active* in the interval $[i, j-1]$, and has span length $j - i + 1$. Sup-active words in binary codes are closely connected with the construction and analysis of trellis oriented generator matrix of $\mathcal{C}^{\perp}$ (see [8,10] and the references therein).

Given N, ϕ for a matroid $\mathcal{M}$, the nodes of the GR-graph (*Gröbner representation graph*) be the set N of canonical forms and there is an edge from node $\mathbf{n}_1$ to $\mathbf{n}_2$ if there is a i, $1 \leq i \leq n$ such that $\phi(\mathbf{n}_1, \mathbf{e}_i) = \mathbf{n}_2$, and the label of that edge is $\mathbf{e}_i$. Let $A \subseteq E = [1, n]$ we define the graph GR_A obtained from the GR-graph of the matroid by deleting all the edges labelled with $\mathbf{e}_i$, such that $i \in E \setminus A$ and element j is active if:

1. Either $j \in A$ and there is a circuit in GR_A such that all the labels $\mathbf{e}_l$ fulfill $l \geq j$ and the (binary) sum of all the labels is different from $\mathbf{0}$.
2. or $j \notin A$ and there is a circuit in $GR_{A \cup \{j\}}$ such that all the labels $\mathbf{e}_l$ fulfill $l \geq j$ and the (binary) sum of all the labels is different from $\mathbf{0}$.

Let A be an interval and let I_A the elimination ideal of $I = I(\mathcal{M})$ (see [7] for a definition) for the variables in $\{x_i\}_{i \in A}$. Let g the greatest element in A, then the cycles correspond to the non-zero binomials in $\left(\bigcup_{e<g} \left[I_{[e,g] \cap A} \setminus I_{[e+1,g] \cap A}\right]\right)$ and $\mathrm{Act}_{\mathcal{M}}(A)$ to the first elements of their leading terms. If we are given an interval $[i, j] \subseteq [1, n]$let $I_{[i,j]}$ the set of codewords sup-active in the interval $[i, j-1]$ is given as those cycles in the graph $GR_{[i,j]}$ where for all the labels $\mathbf{e}_l$ we have $l \in [i, j]$, two of them are $\mathbf{e}_i$ and $\mathbf{e}_j$ and the (binary) sum of all the labels is different from $\mathbf{0}$. If we consider the elimination ideal $I_{[i,j]}$ of I for the variables $x_i, x_{i+1}, \ldots, x_j$ then the cycles correspond to the non-zero binomials in $\left(I_{[i,j]} \setminus I_{[i+1,j]}\right) \setminus I_{[i,j-1]}$.

References

1. Borges-Quintana, M., Borges-Trenard, M.A., Fitzpatrick, P., Martínez-Moro, E.: On a Gröbner bases and combinatorics for binary codes. Appl. Algebra Engrg. Comm. Comput. 19-5, 393–411 (2008)
2. Borges-Quintana, M., Borges-Trenard, M.A., Martínez-Moro, E.: A general framework for applying FGLM techniques to linear codes. In: Fossorier, M.P.C., Imai, H., Lin, S., Poli, A. (eds.) AAECC 2006. LNCS, vol. 3857, pp. 76–86. Springer, Heidelberg (2006)

3. Borges-Quintana, M., Borges-Trenard, M.A., Martínez-Moro, E.: On a Gröbner bases structure associated to linear codes. J. Discrete Math. Sci. Cryptogr. 10-2, 151–191 (2007)
4. Borges-Quintana, M., Borges-Trenard, M.A., Martínez-Moro, E.: A Gröbner representation of linear codes. In: Advances in Coding Theory and Cryptography, pp. 17–32. World Scientific, Singapore (2007)
5. Borges-Quintana, M., Borges-Trenard, M.A., Mora, T.: Computing Gröbner Bases by FGLM Techniques in a Noncommutative Settings. J. Symb.Comp. 30, 429–449 (2000)
6. Cameron, P.: Codes, matroids and trellises, `citeseerx.ist.psu.edu/343758.html`
7. Cox, D., Little, J., O'Shea, D.: Ideals, varieties, and algorithms (1992)
8. Esmaeili, M., Gulliver, T., Secord, N.: Trellis complexity of linear block codes via atomic codewords. In: Chouinard, J.-Y., Fortier, P., Gulliver, T.A. (eds.) Information Theory 1995. LNCS, vol. 1133, pp. 130–148. Springer, Heidelberg (1996)
9. Faugère, J., Gianni, P., Lazard, D., Mora, T.: Efficient Computation of Zero-Dimensional Gröbner Bases by Change of Ordering. J. Symb. Comp. 16(4), 329–344 (1993)
10. Kschischang, F.R., Sorokine, V.: On the trellis structure of block codes. IEEE Trans. Inform. Theory 41, 1924–1937 (1995)
11. Mora, T.: Solving Polynomial Equation Systems II: Macaulay's Paradigm and Gröbner Technology. Cambridge Univ. Press, Cambridge (2005)
12. Mora, T.: The FGLM Problem and Möller's Algorithm on Zero-dimensional Ideals. In: [15]
13. Muder, D.J.: Minimal trellises for block codes. IEEE Trans. Inform. Theory 34(5, part 1), 1049–1053 (1988)
14. Oxley, J.G.: Matroid Theory. Oxford University Press, Oxford (1992)
15. Sala, M., Mora, T., Perret, L., Sakata, S., Traverso, C. (eds.): Gröbner Bases, Coding, and Cryptography. Springer, Heidelberg (2009) (to appear)
16. Reinert, B., Madlener, K., Mora, T.: A Note on Nielsen Reduction and Coset Enumeration. In: Proc. ISSAC 1998, pp. 171–178. ACM, New York (1998)

A Generalization of the Zig-Zag Graph Product by Means of the Sandwich Product*

David M. Monarres[1] and Michael E. O'Sullivan[2]

[1] Grossmont College
El Cajon, CA
[2] San Diego State University
San Diego, CA

Abstract. In this paper we develop a generalization of the *zig-zag* graph product created by Reingold, Vadhan, and Widgerson[8]. We do this by using a broader definition of directed and undirected graphs in which incidence is determined by functions from the edge set to the vertex set. We introduce the *sandwich product* of graphs and show how our general *zig-zag* product is a sandwich product.

1 Introduction

In this extended abstract we present a generalization of the *zig-zag* product of graphs which is defined in the manner outlined in the paper by Hoory et.al.[3]. For this definition let

- G be a m-regular graph on n verticies.
- H be a d-regular graph on m verticies.
- Let $v_1, v_2, \ldots, v_m$ be an enumeration of $V(H)$
- For each $u \in V(G)$, fix an enumeration $e_u^1, e_u^2, \ldots, e_u^m$ of the edges incident with u

Definition 1. *The* **zig-zag** *product of graphs G and H, denoted $G \textcircled{z} H$, is a graph on $V(G) \times V(H)$ for which (u, v_i) is adjacent to (w, v_j) if and only if there exists $k, l \in \{1, 2, \ldots, m\}$ such that both*

- $(v_i, v_k), (v_l, v_j) \in E(H)$, *and*
- $e_u^k = e_u^l$

The development of the zig-zag product was a seminal step in the explicit construction of *expander* graphs. These graphs have been used to address fundametal problems in computer science, such as network design[6,7] and complexity theory[11,10], and areas of pure mathematics, such as topology [1] and measure theory [5]. For a more general survey of the development of the construction of exapnder graphs see Hoory, et al. [3].

* This research was supported by NSF CCF- Theoretical Foundations grant #0635382.

M. Bras-Amorós and T. Høholdt (Eds.): AAECC 2009, LNCS 5527, pp. 231–234, 2009.

2 Graphs, Sandwiches, and the Zig-Zag Product

Our approach does not require the regularity of the first graph to be equal to the number of edges in the second graph, moreover, neither graph needs to be regular. We use definitions of digraph and of graph that allow for loops and parallel edges. The definition is perhaps not standard but is found in Harary[2] (where the term *net* is used instead of digraph) and is close to the definitions used by MacLane[4], and Serre [9]. We introduce the sandwich product of graphs and we show that the zig-zag product of graphs may be concisely presented as the sandwich product of two relatively simple graphs.

2.1 Directed and Undirected Graphs

In the following definitions, a graph is a digraph with additional structure.

Definition 2. *A* **directed graph** *(or* **digraph***), denoted by the letter G, is a collection of two sets $E(G)$ and $V(G)$ together with two functions σ_G and τ_G from $E(G)$ to $V(G)$ where*

- *$E(G)$ and $V(G)$ are known as the* **edge** *and* **vertex** *sets of the net G.*
- *σ_G and τ_G are known as the* **source** *and* **terminus** *functions of the net G.*

Definition 3. *An* ***undirected graph****, or just* ***graph****, is a digraph G in which there exists a unique involution $\rho_G : E \to E$ such that $\tau_G = \sigma_G \circ \rho_G$. We call the function ρ_G the* **rotation mapping** *of the graph G.*

The term rotation map was used in [8]. In the usual depiction of a graph an edge between two vertices represents two edges according to the preceding definition—one going each direction—and the two are images of each other under the involution ρ_G.

The tip to tail concatenation of the graphs used in the definitions of the zig-zag and replacement products has a connection to a more general construct, the *pullback* of mappings of sets.

Definition 4. *Let A, B, and C be sets and let $f : A \to C$ and $g : B \to C$ be functions. Then the pullback of set functions f and g is the set*

$$A \times_C B = \{(a, b) \,|\, f(a) = g(b)\}$$

together with the standard coordinate projections π_1 and π_2.

We can look at the tip to tail traversing of the edges of two graphs in the zig-zag product as really just the pullback of the terminus of one graph and the source of the other. More generally we may define the *concatenation* of any two graphs as follows:

Definition 5. *Let G and H be directed graphs with a common vertex set V and let $E(G) \times_V E(H)$ be the pullback of the mappings τ_G and σ_H. Then the* ***concatenation*** *of G with H, denoted by $G \textcircled{c} H$, is the directed graph with vertex set V, edge set $E(G) \times_V E(H)$, source map $\sigma = \sigma_G \circ \pi_1$ and terminus map $\tau = \tau_H \circ \pi_2$.*

While $G \textcircled{c} H$ is not guaranteed to be a graph, the double concatenation of the form $G \textcircled{c} H \textcircled{c} G$ always is. We will call this type of concatenation the *sandwich product*.

Definition 6. *Let G and H be graphs with common vertex set V. Then the* **sandwich product** *of G with H, denoted $G \textcircled{s} H$, is the graph defined by the triple concatination $G \textcircled{s} H = G \textcircled{c} H \textcircled{c} G$*

The sandwich product is a "zig-zag" like product but with the sole requirement that they have the same vertex set.

2.2 A Zig-Zag Sandwich

We now can display the zig-zag product of two graphs as the sandwich product of two simple graphs. The first factor we call the zig product and the second factor the zag product.

Definition 7. *Let G and H be graphs. The* ***zig product*** *of G with H, denoted $G \textcircled{i} H$, is $|V(G)|$ copies of the graph H defined by*

$$\begin{array}{c} \overset{id_{V(G)} \times \rho_H}{\curvearrowright} \\ V(G) \times E(H) \\ \Big\downarrow {\scriptstyle id_{V(G)} \times \sigma_H} \\ V(G) \times V(H) \end{array}$$

The zig product graph will lie at the beginning and end of our product much like how we start and end in a "cloud" when we traverse the replacement product graph in a "zig-zag" fashion. We now need to define our analog to the bridges between the "clouds" in the zig-zag. We will call this graph the *zag product*. The *zig-zag product* is the sandwich product of the zig and the zag.

Definition 8. *Let G and H be graphs and $\phi : E(G) \to V(H)$ be any mapping. Then the* **zag product** *of G and H, denoted $G \textcircled{@} H$, is the graph with vertex set $V(G) \times V(H)$, edge set $E(G \textcircled{@} H) = E(G)$, source mapping $\sigma_{\text{zag}} = \sigma_g \times \phi$ and rotation mapping $\rho_{\text{zag}} = \rho_G$ as depicted in the diagram*

$$\begin{array}{c} \overset{\rho_G}{\curvearrowright} \\ E(G) \\ \Big\downarrow {\scriptstyle \sigma_G \times \phi} \\ V(G) \times V(H) \end{array}$$

The **zig-zag** *product is then the sandwich product of the zig and the zag.*

$$G \textcircled{z} H = (G \textcircled{i} H) \textcircled{s} (G \textcircled{@} H)$$

The definition above requires only a mapping from the edge set of G to the vertex set of H. It is more general then the one in the paper by Reingold et al. [8] in which G must be regular with regularity equal to the number of vertices in H.

References

1. Gromov, M.: Spaces and questions. Geom. Funct. Anal. (Special Volume, Part I), 118–161 (2000); GAFA 2000 (Tel Aviv, 1999)
2. Harary, F., Norman, R.Z., Cartwright, D.: Structural Models: An Introduction to the Theorey of Directed Graphs. John Wiley & Sons, Inc., Chichester (1966)
3. Hoory, S., Linial, N., Wigderson, A.: Expander Graphs and their Applications. Bulletin of the American Mathematical Society 43(4), 439–561 (2006)
4. MacLane, S.: Categories for the working mathematician. Springer, Heidelberg (1971)
5. Lubotzky, A.: Discrete groups, expanding graphs and invariant measures. Progress in Mathematics, vol. 125. Birkhäuser, Basel (1994); With an appendix by Rogawski, J.D.
6. Pippenger, N.: Sorting and selecting in rounds. SIAM Journal on Computing 16(6), 1032–1038 (1987)
7. Pippenger, N., Yao, A.C.: Rearrangeable networks with limited depth. SIAM Journal of Algebraic and Discrete Methods 3, 411–417 (1982)
8. Reingold, O., Vadhan, S., Widgerson, A.: Entropy waves, the zig-zag graph product, and new constant-degree expanders. Annals of Mathmatics 155, 157–187 (2002)
9. Serre, J.-P.: Trees. Springer, Heidelberg (1980)
10. Valiant, L.G.: Graph-theoretic arguments in low-level complexity. In: Gruska, J. (ed.) MFCS 1977. LNCS, vol. 53, pp. 162–176. Springer, Heidelberg (1977)
11. Urquhart, A.: Hard examples for resolution. Journal of the Association for Computing Machinery 34(1), 209–219 (1987)

Novel Efficient Certificateless Aggregate Signatures

Lei Zhang[1], Bo Qin[1,3], Qianhong Wu[1,2], and Futai Zhang[4]

[1] UNESCO Chair in Data Privacy
Department of Computer Engineering and Mathematics
Universitat Rovira i Virgili
Av. Països Catalans 26, E-43007 Tarragona, Catalonia
{lei.zhang,bo.qin,qianhong.wu}@urv.cat
[2] Key Lab. of Aerospace Information Security and Trusted Computing
Ministry of Education, School of Computer, Wuhan University, China
[3] Dept. of Maths, School of Science, Xi'an University of Technology, China
[4] College of Mathematics and Computer Science
Nanjing Normal University, Nanjing, China
zhangfutai@njnu.edu.cn

Abstract. We propose a new efficient certificateless aggregate signature scheme which has the advantages of both aggregate signatures and certificateless cryptography. The scheme is proven existentially unforgeable against adaptive chosen-message attacks under the standard computational Diffie-Hellman assumption. Our scheme is also efficient in both communication and computation. The proposal is practical for message authentication in many-to-one communications.

1 Introduction

The notion of newly introduced aggregate signatures [2] allows an efficient algorithm to aggregate n signatures of n distinct messages from n different signers into one single signature. The resulting aggregate signature can convince a verifier that the n signers did indeed sign the n original messages. These properties greatly reduce the resulting signature size and make aggregate signatures very applicable to message authentication in many-to-one communications.

The inception of certificateless cryptography [1] efficiently addresses the key escrow problem in ID-based Cryptography. In certificateless cryptosystems, a trusted Key Generation Center (KGC) helps each user to generate his private key. Unlike ID-based cryptosysems, the KGC in certificateless cryptosystems merely determines a partial private key rather than a full private key for each user. Then the user computes the resulting private key with the obtained partial private key and a self-chosen secret value. As for the public key of each user, it is computed from the KGC's public parameters and the secret value chosen by the user. With this mechanism, certificateless cryptosystems avoid the key escrow problem in ID-based cryptosystems.

The advantages of certificateless cryptosystems motivate a number of further studies. The first certificateless signature scheme was presented by Al-Riyami

M. Bras-Amorós and T. Høholdt (Eds.): AAECC 2009, LNCS 5527, pp. 235–238, 2009.

and Paterson [1]. A security definition of certificateless signature was formalized in [7] and the Al-Riyami-Paterson scheme was analyzed in this model. The security model of CLS schemes was further enhanced in [6,8,9]. Two Certificateless Aggregate Signature (CLAS) schemes were recently presented [5] with security proofs in a weak model similar to that in [7]. Subsequently, a new CLAS scheme was proposed in [10] and proven secure in a stronger security model. As for efficiency, the existing schemes require a relatively large number of paring computations in the process of verification and suffer from long resulting signatures.

Our Contribution. In this paper, we propose a novel CLAS scheme which is more efficient than existing schemes. By exploiting the random oracle model, our CLAS scheme is proven existentially unforgeable against adaptive chosen-message attacks under the standard CDH assumption. It allows multiple signers to sign multiple documents in an efficient way and the total verification information (the length of the signature), consists only 2 group elements. Our scheme is also very efficient in computation and the verification procedure need only a very small constant number of pairing computations, independent of the number of aggregated signatures.

2 Our Certificateless Aggregate Signature Scheme

In this section, we propose a new certificateless aggregate signature scheme. Our scheme is realized in groups which allowing efficient bilinear maps [3].

2.1 The Scheme

The specification of the scheme is as follows.

- Setup: Given a security parameter ℓ, the KGC chooses a cyclic additive group G_1 which is generated by P with prime order q, chooses a cyclic multiplicative group G_2 of the same order and a bilinear map $e : G_1 \times G_1 \longrightarrow G_2$. The KGC also chooses a random $\lambda \in Z_q^*$ as the master-key and sets $P_T = \lambda P$, chooses cryptographic hash functions $H_1 \sim H_4 : \{0,1\}^* \longrightarrow G_1$, $H_5 : \{0,1\}^* \longrightarrow Z_q^*$. The system parameter list is params $= (G_1, G_2, e, P, P_T, H_1 \sim H_5)$.
- Partial-Private-Key-Extract: This algorithm is performed by KGC that accepts params, master-key λ and a user's identity $ID_i \in \{0,1\}^*$, and generates the partial private key for the user as follows.
  1. Compute $Q_{i,0} = H_1(ID_i, 0)$, $Q_{i,1} = H_1(ID_i, 1)$.
  2. Output the partial private key $(D_{i,0}, D_{i,1}) = (\lambda Q_{i,0}, \lambda Q_{i,1})$.
- UserKeyGen: This algorithm takes as input params, a user's identity ID_i, selects a random $x_i \in Z_q^*$ and sets his secret/public key as $x_i / P_i = x_i P$.
- Sign: To sign a message M_i using the signing key $(x_i, D_{i,0}, D_{i,1})$, the signer, whose identity is ID_i and the corresponding public key is P_i, first chooses a one-time-use string Δ then performs the following steps.

1. Choose a random $r_i \in Z_q^*$, compute $R_i = r_i P$.
2. Compute $T = H_2(\Delta), V = H_3(\Delta), W = H_4(\Delta)$.
3. Compute $h_i = H_5(M_i||\Delta||ID_i||P_i)$.
4. Compute $S_i = D_{i,0} + x_i V + h_i(D_{i,1} + x_i W) + r_i T$.
5. Output $\sigma_i = (R_i, S_i)$ as the signature on M_i.

– Aggregation: Anyone can act as an aggregate signature generater who can aggregate a collection of individual signatures that use the same string Δ. For an aggregating set (which has the same string Δ) of n users with identities $\{ID_1, \cdots, ID_n\}$ and the corresponding public keys $\{P_1, \cdots, P_n\}$, and message-signature pairs $(M_1, \sigma_1 = (R_1, S_1)), \cdots, (M_n, \sigma_n = (R_n, S_n))$ from $\{U_1, \cdots, U_n\}$ respectively, the aggregate signature generater computes $R = \sum_{i=1}^{n} R_i, S = \sum_{i=1}^{n} S_i$ and outputs the aggregate signature $\sigma = (R, S)$.

– Aggregate Verify: To verify an aggregate signature $\sigma = (R, S)$ signed by n users with identities $\{ID_1, ..., ID_n\}$ and corresponding public keys $\{P_1, ..., P_n\}$ on messages $\{M_1, ..., M_n\}$ under the same string Δ, the verifier performs the following steps.
   1. Compute $T = H_2(\Delta), V = H_3(\Delta), W = H_4(\Delta)$, and for all $i, 1 \leq i \leq n$ compute $h_i = H_5(M_i||\Delta||ID_i||P_i), Q_{i,0} = H_1(ID_i, 0), Q_{i,1} = H_1(ID_i, 1)$.
   2. Verify $e(S, P) \stackrel{?}{=} e(P_T, \sum_{i=1}^{n} Q_{i,0} + \sum_{i=1}^{n} h_i Q_{i,1}) e(T, R) e(W, \sum_{i=1}^{n} h_i P_i)$ $e(V, \sum_{i=1}^{n} P_i)$. If the equation holds, output *true*. Otherwise, output *false*.

In our scheme, each user in an aggregating set must use the same one-time-use string Δ when signing. As mentioned in [4], it is straightforward to choose such a Δ in certain settings. For example, if the signers have access to some loosely synchronized clocks, Δ can be chosen based on the current time. Furthermore, if Δ is sufficiently long, then it will be statistically unique. By exploiting a approach similar to that presented in [4], the one-time-use restriction on common reference string Δ in the above scheme can also be removed to achieve better applicability.

2.2 Security Analysis

Two types of adversaries, who can access to services in addition to those provided to the attacker against regular signatures, are considered in CL-PKC – Type I adversary and Type II adversary. A Type I adversary is not allowed to access to the master-key, but he can replace the public key of any user with a value of his choice. A Type II adversary can access to the master-key but he cannot replace the public key of any user. In a secure CLAS scheme, it is infeasible for Type I adversary and Type II adversary to forge a valid signature.

Under the standard computational Diffie-Hellman assumption, the proposed CLAS scheme is provably secure against both types of adversaries in the random model. The formal security proof is in the full version of this paper.

3 Conclusion

We presented an efficient certificateless aggregate signature scheme. To verify an aggregate signature signed by n users on n messages under the same string, a verifier only needs to compute four pairing operations. The proposal is provably secure in the random oracle model assuming that the computational Diffie-Hellman problem is hard. Our CLAS scheme can be applied to authentication in bandwidth limited scenarios such as many-to-one communications.

Acknowledgments and Disclaimer

This paper is partly supported by the Spanish Government through projects CONSOLIDER INGENIO 2010 CSD2007-00004 ARES, TSI2007-65406-C03-01 E-AEGIS and by the national nature science foundation of China (No. 60673070). The views of the author with the UNESCO Chair in Data Privacy do not necessarily reflect the position of UNESCO nor commit that organization.

References

1. Al-Riyami, S.S., Paterson, K.G.: Certificateless Public Key Cryptography. In: Laih, C.-S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 452–473. Springer, Heidelberg (2003)
2. Boneh, D., Gentry, C., Shacham, H., Lynn, B.: Aggregate and Verifiably Encrypted Signatures from Bilinear Maps. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 416–432. Springer, Heidelberg (2003)
3. Boneh, D., Franklin, M.: Identity-based Encryption from the Weil Pairing. SIAM J. Comput. 32, 586–615 (2003); a Preliminary Version Appeared. In: Kilian, J. (ed.): CRYPTO 2001. LNCS, vol. 2139, pp. 213-229. Springer, Heidelberg (2001)
4. Gentry, C., Ramzan, Z.: Identity-Based Aggregate Signatures. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T.G. (eds.) PKC 2006. LNCS, vol. 3958, pp. 257–273. Springer, Heidelberg (2006)
5. Gong, Z., Long, Y., Hong, X., Chen, K.: Two Certificateless Aggregate Signatures from Bilinear Maps. In: Proc. of the Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, pp. 188–193 (2007)
6. Hu, B.C., Wong, D.S., Zhang, Z., Deng, X.: Key Replacement Attack Against a Generic Construction of Certificateless Signature. In: Batten, L.M., Safavi-Naini, R. (eds.) ACISP 2006. LNCS, vol. 4058, pp. 235–246. Springer, Heidelberg (2006)
7. Huang, X., Susilo, W., Mu, Y., Zhang, F.: On the Security of Certificateless Signature Schemes from Asiacrypt 2003. In: Desmedt, Y.G., Wang, H., Mu, Y., Li, Y. (eds.) CANS 2005. LNCS, vol. 3810, pp. 13–25. Springer, Heidelberg (2005)
8. Huang, X., Mu, Y., Susilo, W., Wong, D.S., Wu, W.: Certificateless Signature Revisited. In: Pieprzyk, J., Ghodosi, H., Dawson, E. (eds.) ACISP 2007. LNCS, vol. 4586, pp. 308–322. Springer, Heidelberg (2007)
9. Zhang, Z., Wong, D.: Certificateless Public-Key Signature: Security Model and Efficient Construction. In: Zhou, J., Yung, M., Bao, F. (eds.) ACNS 2006. LNCS, vol. 3989, pp. 293–308. Springer, Heidelberg (2006)
10. Zhang, L., Zhang, F.: A New Certificateless Aggregate Signature Scheme. Computer Communications (2009), doi:10.1016/j.comcom.2008.12.042

Bounds on the Number of Users for Random 2-Secure Codes

Manabu Hagiwara[1,2], Takahiro Yoshida[1,2], and Hideki Imai[1,3]

[1] Research Center for Information Security, National Institute of Advanced Industrial Science and Technology, 1-18-13 Sotokanda, Chiyoda-ku, Tokyo 101-0021, Japan
hagiwara.hagiwara@aist.go.jp
[2] Center for Research and Development Initiative, Chuo University, 1-13-27 Kasuga, Bunkyo-ku, Tokyo, Japan
t-yoshida@imailab.jp
[3] Faculty of Science and Engineering, Chuo University, Kasuga, Bunkyo-ku, Tokyo. 112-8551, Japan
h-imai@aist.go.jp

1 Introduction

An illegal re-distribution problem is a problem that a regular user who received content re-distributes without legal permutation. This becomes a social problem all over the world. As it is indicated in [2], this problem has been known since a few or more handred years ago. Boneh and Shaw formalize this problem as collusion-secure fingerprinting (c-secure code) for digital data [2]. After their work, study of c-secure code has become one of popular research stream for information security [1]. One of remarkable results is called Tardos's code [5]. Tardos's code achieved to construct approximately optimal length c-secure codes with random coding method.

On the other hand, it is an interesting research way to restrict the number of colluders to small values [3]. Small colluder cases give us many knowledge, techniques, and examples for c-secure code theory. In [4], it is shown that coin-flipping codes are useful for 2-secure code which is a c-secure code restricted to against at most two colluders.

In this extended abstruct, we investigate the coin-flipping codes as 2-secure codes. We give three formulas to represent the success probability of simple tracing algorithm for the coin-flipping codes. It shows that even if the code length is short, e.g. 128 bits, it succeeds to trace one of colluders with high probability, e.g. 0.9999. A non-trivial achievable rate for our coin-flipping code is also given.

2 Notations and Definitions

Let n be a positive integer. First, the **administrator** divides a content to n or more parts, chose n parts from the divided contents.

A **user** i can requests to obtain a copy of the content one time. The administrator registers the person i. This is formalized as follows: The set of registered

M. Bras-Amorós and T. Høholdt (Eds.): AAECC 2009, LNCS 5527, pp. 239–242, 2009.

users $\mathcal{U}$ is defined as a set of integers. For ease, the set $\mathcal{U}$ is starting with 1 and is consecutive, i.e. $\mathcal{U} = \{1, 2, 3, \dots\}$.

The administrator flips coins n times. We assume that the probability to have face side is 0.5 and the one of the other side is 0.5 also. If jth coin is face side, we put $c_j^{(i)} = 0$. If otherwise, put $c_j^{(i)} = 1$. Therefore the administrator obtains n-bit sequence $c^{(i)} = (c_1^{(i)}, c_2^{(i)}, \dots, c_n^{(i)})$. The administrator memorizes $c^{(i)}$ to a database which the only administrator accesses. We call the set of codewords constructed by coin-flipping **coin-flipping code**. The bit $c_j^{(i)}$ is encrypted and is embedded to jth part of the requested content. Assume that any user cannot decrypt and cannot distinguish which bit is embedded.

The user i receives the requested content which the secret n-bit sequence $c^{(i)}$ is embedded by the administrator.

2.1 Collusion Attack under Marking-Assumption

We introduce the notion, called **marking-assumption**. For details, see the reference [2]. It is assumed that any user cannot delete the embedded codeword or a part of the codeword from a distributed content. The only attack users can is to shuffle the distributed content. We assumed that user cannot distinguish which bit is embedded to the distributed contents. However, it is possible for users a and b to find a part associated to jth part if the different bits $c_j^{(a)} \neq c_j^{(b)}$ are embedded by comparing their content.

2.2 Tracing Algorithm for Coin-Flipping Codes

If illegally re-distributed content is found somewhere, the administrator decrypts a bit sequence y from the content. The administrator tries to trace at least one of the colluders by analyzing y. A **tracing algorithm** $\mathcal{T}$ is an algorithm which inputs a bit sequence and outputs a subset of the users $\mathcal{U}$.

The details of our algorithm are the following:

For the input y and an user $i \in \mathcal{U}$, calculate Hamming distance $d(c^{(i)}, y)$. For i, denote the Hamming distance by s_i and call it the score of i for y. Since a value of Hamming distance is non-negative integer, we have minimum value s in $\{d(c^{(i)}, y) | i \in \mathcal{U}\}$. Define a set $U_s := \{i \in \mathcal{U} | d(c^{(i)}, y) = s\}$. Then output U_s. We denote the output U_s by $\mathcal{T}(y)$. Remark that the output U_s is not the empty set. We call the algorithm $\mathcal{T}(y)$ **Hamming distance tracing algorithm**.

3 Main Results

Let us introduce the numerical security indicator for coin-flipping codes as 2-secure codes. We propose the success probability $\mathbb{S}_{n,U}$ by putting:

$$\mathbb{S}_{n,U} = \text{Prob}[\mathcal{T}(y) \subset \{\text{colluders}\}, y : \text{ illegally re-distributed code}].$$

$$= \text{Prob}[c^{(1)}, c^{(2)}, \dots, \in_R \{0,1\}^n, a, b \in_R \mathcal{U}, y = \mathcal{S}_{a,b}(c^{(a)}, c^{(b)}), \mathcal{T}(y) \subset \{a, b\}],$$

where $\mathcal{S}_{a,b}(c^{(a)}, c^{(b)})$ is the best strategy for a, b with inputs $(c^{(a)}, c^{(b)})$ and Ex() is the expected value function.

Theorem 1. *Let n be the code length of a coin-flipping code and U the number of the users. Then the success probability $\mathbb{S}_{n,U}$ is written by the following two forms:*
1) $\frac{1}{2^n}\sum_{0\le s\le n, s:odd}\binom{n+1}{s}2^{-(U-2)h(n,(s-1)/2)}$,
2) $\mathrm{Ex}_{x\in\{0,1\}^n}[(\frac{1}{2})^{(U-2)h(n,\mathrm{wt}(x)/2)}]$,
where $h(n,a) := n - \log_2(2^n - \sum_{0\le t\le a}\binom{n}{t}))$ *and* $\mathrm{wt}(x)$ *is the Hamming weight of* x.

We call $\mathbb{S}_{n,U}$ a **security parameter**. How $\mathbb{S}_{n,U}$ should be small depends on the application. It is enough that the users give up re-distributing illegally.

It is easy to calculate $h(n,a)$ if n is not so huge, e.g. $n = 128$, and $0 \le a < n/2$ by using a math software, e.g. MuPAD, MATLAB, MAPLE, Mathematica and so on. Once we obtained a list of $h(n,a)$, it is easy to calculate $\mathbb{S}_{n,U}$ by the form in Theorem 1 (1). The following is a table of upper bounds on the maximal number of users for satisfying $\mathbb{S}_{n,U} > 0.99, 0.999, 0.9999$.

Table 1. The Upper Bounds on The Number of Users for Satisfying a Given Security Parameter and a Code Length

	$\mathbb{S}_{n,U} > 0.99$	$\mathbb{S}_{n,U} > 0.999$	$\mathbb{S}_{n,U} > 0.9999$
$n = 32$	3	2	2
$n = 64$	57	7	2
$n = 128$	68031	5363	516

From now, let introduce the following game G:

(1) Toss up n coins. Count the number of the face sides.
(2) Repeat step (1) $U-2$ times. Put the maximal number of the face sides and denote it by s_1.
(3) Toss up n coins again. Count the number of the faces and denote it by s_2.
(4) You win if $2s_1 + s_2 < 2n$. You lose if the otherwise.

Denote the winning probability of the above game by $\mathbb{W}_{n,U}$.

Theorem 2. *For $U \ge 3, n \ge 1$, $\mathbb{W}_{n,U} = \mathbb{S}_{n,U}$.*

It is interesting to know a theoretical capacity on the maximal number of the users with infinite length. Recently, a capacity of collusion secure code has been investigated by many researchers [1]. It is not trivial that there exist the capacity for our code. We investigate an achievable rate R for the coin-flipping code with Hamming distance tracing algorithm, where an achievable rate R is a constant such that $\lim_{n\to\infty}\mathbb{S}_{n,U} = 0$ for $R_0 > R$, where $U = 2^{R_0}$

Theorem 3. *For any $R < 1 - h(1/4)$, $\lim_{n\to\infty}\mathbb{S}_{n,U} = 1$, where $h(\cdot)$ is the binary entropy function, i.e. $h(x) = -x\log_2 x - (1-x)\log_2(1-x)$. In particular, the capacity is more than $1 - h(1/4)$, if it exists.*

4 Applications: Digital Cinema Distribution and Hard Paper Distribution

A recent illegally re-distribution problem is un-official copied DVD purchases. It is said that someone records a film by his own video camera in a cinema theater and he made illegally copied DVDs by him to sell them.

In this senario, the administrator is a provide company for films and the users are cinema theatersa and divides films to n or more sets of frames. If a cinema theater requests the film, the administrator embeds n-bit codeword to the associated n-set of frames. If an illegally re-distributed content is found, the administrator traces a cinema theater where the film was recorded. Then the administrator asks the cinema theater to pay more attention customers illegal recording. Corporation of the administrator and cinema theaters makes illegal recoding harder for pirate customers. At the early stage, it is possible for a pirate customer to find cinema theaters which does not pay so much attention to such illegal recording. However by repeating making efforts between the administrator and theaters, we expect to reduce such crime. There are not many film screens. For example, 3221 screens are in Japan, 2007 [6]. In Japan, the success probability of coin-flipping codes is more than 0.999 if the code length is 128.

The next application is to protect hard papers from away illegally re-distributions. The administrator is the president of a company or the chairman of the director's meeting. In this case, the number of the users is at most dozens. Furthermore the number of the colluders is at most a few by social reasons. The administrator embeds each bit of a fingerprinting codeword to each page of the material. If there is a mass of materials, it is hard to delete that from the point of cost.

References

1. Amiri, E., Trados, G.: High rate fingerprinting codes and the fingerprinting capacity. In: Proc. of the Twentieth Annual ACM-SIAM Symp. on Discrete Algorithms
2. Boneh, D., Shaw, J.: Collusion-secure fingerprinting for digital data. IEEE Transactions of Information Theory 44, 480–491 (1998)
3. Schaathun, H.G.: Fighting three pirates with scattering codes. In: Proc. Inter. Symp. on Information Theory, 2004. ISIT 2004, June-2 July, vol. 27, p. 202 (2004)
4. Hagiwara, M., et al.: A short random fingerprinting code against a small number of pirates. In: Fossorier, M.P.C., Imai, H., Lin, S., Poli, A. (eds.) AAECC 2006. LNCS, vol. 3857, pp. 193–202. Springer, Heidelberg (2006)
5. Tardos, G.: Optimal Probabilistic Fingerprint Codes. In: Proceedings of the 35th Annual ACM Symp. on Theory of Computing, pp. 116–125 (2003); J. of the ACM (to appear)
6. Website of Motion Picture Producers Association of Japan, Inc., `http://www.eiren.org/history_e/index.html`

Author Index