YACC'10

et Another Conference on Cryptography

Porquerolles, France October 4-8

BOOK OF ABSTRACTS

Program committee:

C. Carlet (laga Université Paris 8, France)
J.-C. Faugères (inria, LIP6, Université Paris 6, France)
G. McGuire (University College Dublin, Dublin)
F. Ozbudak (metu, Department of Mathematics, Ankara, Turkey)
D. Pointcheval (ens, Department of Computer Science, Paris, France)
P. Véron (Imath, Université du Sud Toulon-Var, France)

Invited speakers:

J.	Daemen	STMicroelectronics
A.	Pott	Otto-Von-Guericke University.
T.	Helleseth	University of Bergen.

Old and New Towers over Finite Fields

Henning Stichtenoth

Sabancı University, MDBF
34956 Tuzla, İstanbul, Turkey
henning@sabanciuniv.edu

Abstract. The theory of recursive towers of function fields over finite fields was developed by A. Garcia and the author since 1995. In this talk we describe some of the main ideas and results, and we present old and new examples of interesting recursive towers.

1

We consider towers of algebraic function fields, i.e. sequences $\mathcal{F} = (F_0, F_1, F_2, \ldots)$ of function fields F_i over a finite field $\mathbb{F}_q$ where $F_0 \subseteq F_1 \subseteq F_2 \subseteq \ldots$, all extensions F_{i+1}/F_i are separable and the genus $g(F_i)$ tends to infinity as $i \to \infty$. We denote by $N(F_i)$ the number of rational (i.e. degree one) places of F_i over $\mathbb{F}_q$. It is well-known that the limit

$$\lambda(\mathcal{F}) = \lim_{i\to\infty} \frac{N(F_i)}{g(F_i)}$$

exists, and it is bounded from above by the Drinfeld-Vladut bound

$$\lambda(\mathcal{F}) \leq \sqrt{q} - 1.$$

A tower $\mathcal{F}$ over $\mathbb{F}_q$ is said to be *asymptotically good*, if $\lambda(\mathcal{F}) > 0$, it is *asymptotically optimal* if $\lambda(\mathcal{F}) = \sqrt{q} - 1$, and it is *asymptotically bad* if $\lambda(\mathcal{F}) = 0$. The study of asymptotically good towers is of high interest because of their various applications in coding theory (Tsfasman-Vladut-Zink theorem), cryptography etc., but also as a mathematical challenge.

In order to determine the limit of a tower, it is useful to study the behaviour of the genus and of the number of rational places separately; i.e. one considers the limits

$$\gamma(\mathcal{F}/F_0) := \lim_{i\to\infty} \frac{g(F_i)}{[F_i : F_0]}\ , \quad \text{the } \textit{genus} \text{ of } \mathcal{F} \text{ over } F_0,$$

and

$$\nu(\mathcal{F}/F_0) := \lim_{i\to\infty} \frac{N(F_i)}{[F_i : F_0]}\ , \quad \text{the } \textit{splitting rate} \text{ of } \mathcal{F} \text{ over } F_0.$$

It is easily seen that $0 < \gamma(\mathcal{F}/F_0) \leq \infty$ and $0 \leq \nu(\mathcal{F}/F_0) < \infty$, and moreover,

$$\lambda(\mathcal{F}/F_0) = \frac{\nu(\mathcal{F}/F_0)}{\gamma(\mathcal{F}/F_0)}.$$

This implies that the tower $\mathcal{F}$ is asymptotically good if and only if its genus $\gamma(\mathcal{F}/F_0)$ is finite, and its splitting rate $\nu(\mathcal{F}/F_0)$ is strictly positive. We introduce the *ramification locus*

$$V(\mathcal{F}/F_0) := \{P \mid P \text{ is a place of } F_0 \text{ which is ramified in } F_k/F_0 \text{ for some } k > 0\}.$$

If $V(\mathcal{F}/F_0)$ is finite and all ramification in the tower is tame, it follows from the Hurwitz genus formula that the genus $\gamma(\mathcal{F}/F_0)$ is finite. The condition of tame ramification can be weakened as follows: suppose that for every place $P \in V(\mathcal{F}/F_0)$, there is a constant $c_P \in \mathbb{R}$ such that for all $k \geq 1$ and all places Q of F_k lying above P, the different exponent $d(Q|P)$ is bounded by

$$d(Q|P) \leq c_P \cdot e(Q|P)$$

(here we denote by $e(Q|P)$ the ramification index of $Q|P$). Then finiteness of the ramification locus implies again finiteness of the genus $\gamma(\mathcal{F}/F_0)$.

In a similar way, one can often make sure that the splitting rate $\nu(\mathcal{F}/F_0)$ is positive. To this aim, we define the *splitting locus*

$$Z(\mathcal{F}/F_0) = \{P \mid P \text{ is a place of } \ F_0 \text{ which splits completely in } \ \mathcal{F}\}.$$

Obviously, $\nu(\mathcal{F}/F_0) \geq |Z(\mathcal{F}/F_0)|$, and therefore a non-empty splitting locus implies that the splitting rate of the tower is positive. To conclude, we have the following result:

Lemma 1. *Suppose that the tower $\mathcal{F}$ satisfies the following conditions:*
(i) the ramification locus $V(\mathcal{F}/F_0)$ is finite,
(ii) for every $P \in V(\mathcal{F}/F_0)$, there is a constant $c_P \in \mathbb{R}$ such that $d(Q|P) \leq c_P \cdot e(Q|P)$ for all places Q above P,
(iii) the splitting locus of $\mathcal{F}/F_0$ is non-empty.
Then the tower is asymptotically good.

The conditions given in Lemma 1 are *sufficient* conditions for $\mathcal{F}$ to be asymptotically good. On the other hand, all examples of good towers that I know, satisfy all these conditions.

There are essentially three sources for towers: *class field towers* (first introduced by Serre [3]), *towers of modular curves* (first introduced by Ihara [2] and Tsfasman, Vladut and Zink [4]), and *recursive towers* (introduced by Garcia and the author [1]).

A tower $\mathcal{F} = (F_i)_{i \geq 0}$ is said to be *recursive* if there exist a non-zero polynomial $f(x, y) \in \mathbb{F}_q[x, y]$ and elements $x_0, x_1, x_2, \ldots$ such that for all $i \geq 0$,

$$F_0 = \mathbb{F}_q(x_0) \ , F_{i+1} = F_i(x_{i+1}) \text{ and } f(x_i, x_{i+1}) = 0 \ .$$

In this talk we discuss several examples of recursive towers, some of which are well-known, some are new. In all cases we focus on the question if the conditions of Lemma 1 are satisfied.

References

1. Garcia, A., Stichtenoth, H.: On the Asymptotic Behaviour of Some Towers of Function Fields over Finite Fields, J. Number Theory 61, 248-273 (1996)
2. Ihara, Y.: Some remarks on the number of rational points of algebraic curves over finite fields, J. Fac. Sci. Univ. Tokyo Sect. IA Math. 28, 721-724 (1981)
3. Serre, J.-P.: Sur le Nombre des Points Rationnels d'une Courbe Algebrique sur un Corps Finis, C. R. Acad. Sci. Paris 296, 397-402 (1983)
4. Tsfasman, M. A., Vladut, S. G., Zink, T.: Modular Curves, Shimura Curves, and Goppa Codes, Better than the Varshamov-Gilbert Bound, Math. Nachr. 109, 21-28 (1982)

ON THE TENSOR RANK OF MULTIPLICATION IN ANY EXTENSION OF $\mathbb{F}_2$

STÉPHANE BALLET AND JULIA PIELTANT

Abstract. In this paper, we obtain new bounds for the tensor rank of multiplication in any extension of $\mathbb{F}_2$. In particular, it also enables us to obtain the best known asymptotic bound. In this aim, we use the generalized algorithm of type Chudnovsky with derivative evaluations on places of degree one, two and four applied on the descent over $\mathbb{F}_2$ of a Garcia-Stichtenoth tower of algebraic function fields defined over $\mathbb{F}_{2^4}$.

1. Introduction

1.1. **Tensor rank of multiplication.** Let $\mathbb{F}_q$ be a finite field with q elements where q is a prime power and let $\mathbb{F}_{q^n}$ be a $\mathbb{F}_q$-extension of degree n. We denote by m the ordinary multiplication in the $\mathbb{F}_q$-vector space $\mathbb{F}_{q^n}$. The multiplication m is a bilinear map from $\mathbb{F}_{q^n} \times \mathbb{F}_{q^n}$ into $\mathbb{F}_{q^n}$, thus it corresponds to a linear map M from the tensor product $\mathbb{F}_{q^n} \bigotimes \mathbb{F}_{q^n}$ over $\mathbb{F}_q$ into $\mathbb{F}_{q^n}$. One can also represent M by a tensor $t_M \in \mathbb{F}_{q^n}^* \bigotimes \mathbb{F}_{q^n}^* \bigotimes \mathbb{F}_{q^n}$ where $\mathbb{F}_{q^n}^*$ denotes the dual of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$. Hence the product of two elements x and y of $\mathbb{F}_{q^n}$ is the convolution of this tensor with $x \otimes y \in \mathbb{F}_{q^n} \bigotimes \mathbb{F}_{q^n}$. If

$$t_M = \sum_{l=1}^{\lambda} a_l \otimes b_l \otimes c_l \tag{1}$$

where $a_l \in \mathbb{F}_{q^n}^*$, $b_l \in \mathbb{F}_{q^n}^*$, $c_l \in \mathbb{F}_{q^n}$, then

$$x \cdot y = \sum_{l=1}^{\lambda} a_l(x) b_l(y) c_l. \tag{2}$$

Every expression (2) is called a bilinear multiplication algorithm $\mathcal{U}$. The integer λ is called the multiplicative complexity $\mu(\mathcal{U})$ of $\mathcal{U}$.
Let us set

$$\mu_q(n) = \min_{\mathcal{U}} \mu(\mathcal{U}),$$

where $\mathcal{U}$ is running over all bilinear multiplication algorithms in $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$.
Then $\mu_q(n)$ corresponds to the minimum possible number of summands in any tensor decomposition of type (1), which is the rank of the tensor of multiplication in $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$. The tensor rank $\mu_q(n)$ is also called the bilinear complexity of multiplication in $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$.

1.2. **Known results.**

1.2.1. *General results.* From the existence of towers of algebraic function fields satisfying good properties, it was proved:

Theorem 1.1. *Let $q = p^r$ be a power of the prime p. The tensor rank $\mu_q(n)$ of multiplication in any finite field $\mathbb{F}_{q^n}$ is linear with respect to the extension degree, more precisely:*

$$\mu_q(n) \leq C_q n$$

Date: April 22, 2010.
2000 *Mathematics Subject Classification.* Primary 14H05; Secondaries 11Y16, 12E20.
Key words and phrases. Algebraic function fields, tower of function fields, tensor rank, algorithm, finite fields.

where C_q is the constant defined by:

$$C_q = \begin{cases} \text{if } q = 2 & \text{then } 54 & [2] \\ \text{else if } q = 3 & \text{then } 27 & [2] \\ \text{else if } q = p \geq 5 & \text{then } 3(1 + \frac{4}{q-3}) & [4] \\ \text{else if } q = p^2 \geq 25 & \text{then } 2(1 + \frac{2}{\sqrt{q}-3}) & [4] \\ \text{else if } q = p^{2k} \geq 16 & \text{then } 2(1 + \frac{p}{\sqrt{q}-3}) & [3] \\ \text{else if } q \geq 16 & \text{then } 3(1 + \frac{2p}{q-3}) & [7], [6] \text{ and } [5] \\ \text{else if } q > 3 & \text{then } 6(1 + \frac{p}{q-3}) & [3]. \end{cases}$$

In order to obtain these good estimates for the constant C_q, S. Ballet has given in [2] some easy to verify conditions allowing the use of the D.V. and G.V. Chudnovsky algorithm. Then S. Ballet and R. Rolland have generalized in [7] the algorithm using places of degree one and two.

Let us present the last version of this algorithm, which is a generalization of the algorithm of type Chudnovsky introduced by N. Arnaud in [1] and M. Cenk and F. Özbudak in [10]. This generalization uses several coefficients in the local expansion at each place P_i instead of just the first one. Due to the way to obtain the local expansion of a product from the local expansion of each term, the bound for the bilinear complexity involves the complexity notion $\widehat{M}_q(u)$ introduced by M. Cenk and F. Özbudak in [10] and defined as follows:

Definition 1.2. *We denote by $\widehat{M}_q(u)$ the minimum number of multiplications needed in $\mathbb{F}_q$ in order to obtain coefficients of the product of two arbitrary u-term polynomials modulo x^u in $\mathbb{F}_q[x]$.*

Now we introduce the generalized algorithm of type Chudnovsky described in [10].

Theorem 1.3. *Let*
- *q be a prime power,*
- *$F/\mathbb{F}_q$ be an algebraic function field,*
- *Q be a degree n place of $F/\mathbb{F}_q$,*
- *$\mathcal{D}$ be a divisor of $F/\mathbb{F}_q$,*
- *$\mathcal{P} = \{P_1, \ldots, P_N\}$ be a set of N places of arbitrary degree,*
- *$u_1, \ldots, u_N$ be positive integers.*

We suppose that Q and all the places in $\mathcal{P}$ are not in the support of $\mathcal{D}$ and that:

a) the application

$$Ev_Q : \left\{ \begin{array}{ccc} \mathcal{L}(\mathcal{D}) & \to & \mathbb{F}_{q^n} \simeq F_Q \\ f & \longmapsto & f(Q) \end{array} \right.$$

is onto,

b) the application

$$Ev_{\mathcal{P}} : \left\{ \begin{array}{ccc} \mathcal{L}(2\mathcal{D}) & \longrightarrow & \left(\mathbb{F}_{q^{\deg P_1}}\right)^{u_1} \times \left(\mathbb{F}_{q^{\deg P_2}}\right)^{u_2} \times \cdots \times \left(\mathbb{F}_{q^{\deg P_N}}\right)^{u_N} \\ f & \longmapsto & \big(\varphi_1(f), \varphi_2(f), \ldots, \varphi_N(f)\big) \end{array} \right.$$

is injective, where the application φ_i is defined by

$$\varphi_i : \left\{ \begin{array}{ccc} \mathcal{L}(2\mathcal{D}) & \longrightarrow & \left(\mathbb{F}_{q^{\deg P_i}}\right)^{u_i} \\ f & \longmapsto & \big(f(P_i), f'(P_i), \ldots, f^{(u_i-1)}(P_i)\big) \end{array} \right.$$

with $f = f(P_i) + f'(P_i)t_i + f''(P_i)t_i^2 + \ldots + f^{(k)}(P_i)t_i^k + \ldots$, the local expansion at P_i of f in $\mathcal{L}(2\mathcal{D})$, with respect to the local parameter t_i. Note that we set $f^{(0)} = f$.

Then

$$\mu_q(n) \leq \sum_{i=1}^{N} \mu_q(\deg P_i) \widehat{M}_{q^{\deg P_i}}(u_i).$$

Let us remark that the algorithm given in [11] by D.V. and G.V. Chudnovsky is the case $\deg P_i = 1$ and $u_i = 1$ for $i = 1, \ldots, N$. The generalization introduced here is useful: it allows us to use certain places many times, thus less places are necessary to get the injectivity of $Ev_{\mathcal{P}}$.

In particular, in [1] N. Arnaud has presented a simplified version of this algorithm where all integers $u_i = 2$ and places are of degree one and two. Applying his algorithm on Garcia-Stichtenoth towers, he obtained the two following bounds.

Theorem 1.4. *Let $q = p^r \geq 4$ be a prime power. Then*

$$\textit{(i)} \qquad \mu_{q^2}(n) \leq 2\left(1 + \frac{p}{q - 3 + (p-1)\left(1 - \frac{1}{q+1}\right)}\right) n,$$

$$\textit{(ii)} \qquad \mu_q(n) \leq 3\left(1 + \frac{2p}{q - 3 + 2(p-1)\left(1 - \frac{1}{q+1}\right)}\right) n.$$

1.2.2. *Asymptotic bounds for the extensions of $\mathbb{F}_2$.* From the asymptotic point of view, let us recall that I. Shparlinski, M. Tsfasman and S. Vladut have given in [13] many interesting remarks on the algorithm of D.V. and G.V. Chudnovsky. In particular, they considered the following asymptotic bounds for the bilinear complexity

$$M_q = \limsup_{k \to \infty} \frac{\mu_q(k)}{k}$$

and

$$m_q = \liminf_{k \to \infty} \frac{\mu_q(k)}{k}.$$

In [13], they claim that $M_2 \leq 27$ but I. Cascudo, R. Cramer and C. Xing recently noticed in [9] a gap in the proof about $M_2 \leq 27$ established by I. Shparlinsky, M. Tsfasman and S. Vladut. Consequently, we can consider this bound as not proved. Anyway, it is possible to obtain easily a better bound for M_2 from one of the bounds of N. Arnaud. Indeed, by using Bound (ii) of Theorem 1.4 we obtain:

Proposition 1.5.

$$M_2 \leq \frac{297}{13} \approx 22.85.$$

1.3. **New results established in this paper.** Our main result concerns an improvement of the asymptotic bound for the tensor rank of multiplication in any extension of $\mathbb{F}_2$. More precisely, we prove that:

$$M_2 \leq \frac{477}{26} \approx 18.35.$$

This result comes from a new bound for the tensor rank of multiplication in any extension of $\mathbb{F}_2$ that we also obtain in this paper, namely:

$$\mu_2(n) \leq \frac{477}{26} n + \frac{45}{2}.$$

In this aim, we use a modified Garcia-Stichtenoth's tower [12] studied in [3], [7], [6] and [4]. Specially, we apply an adapted version of the generalized algorithm of type Chudnosky which involves only places of degree one, two and four with derivative evaluations to the descent of the definition field of this Garcia-Stichtenoth's tower on the field $\mathbb{F}_2$ obtained in [8].

2. New bounds for the tensor rank

2.1. **Adapted algorithm of type Chudnovsky and associated complexity.** In this section, we use places of degree one, two and four to obtain new results for the tensor rank of multiplication in any extension of the finite field $\mathbb{F}_2$.

First of all, we specialize the general algorithm presented in Theorem 1.3 for places of degree one, two and four by using first derivative evaluations, i.e. with $u_i \leq 2$ for $i = 1, \ldots, N$.

Proposition 2.1. *Let*
- *q be a prime power,*
- *$F/\mathbb{F}_q$ be an algebraic function field,*
- *Q be a degree n place of $F/\mathbb{F}_q$,*
- *$\mathcal{D}$ be a divisor of $F/\mathbb{F}_q$,*
- *$\mathcal{P} = \{P_1, \ldots, P_{N_1}, P_{N_1+1}, \ldots, P_{N_1+N_2}, P_{N_1+N_2+1}, \ldots, P_{N_1+N_2+N_4}\}$ be a set of N_1 places of degree one, N_2 places of degree two and N_4 places of degree four.*
- *$0 \leq l_1 \leq N_1$, $0 \leq l_2 \leq N_2$ and $0 \leq l_4 \leq N_4$ be three integers.*

We suppose that Q and all the places in $\mathcal{P}$ are not in the support of $\mathcal{D}$ and that:

a) the application

$$Ev_Q : \mathcal{L}(\mathcal{D}) \to \mathbb{F}_{q^n} \simeq F_Q$$

is onto,

b) the application

$$Ev_{\mathcal{P}} : \begin{cases} \mathcal{L}(2\mathcal{D}) & \to & \mathbb{F}_q^{N_1} \times \mathbb{F}_q^{l_1} \times \mathbb{F}_{q^2}^{N_2} \times \mathbb{F}_{q^2}^{l_2} \times \mathbb{F}_{q^4}^{N_4} \times \mathbb{F}_{q^4}^{l_4} \\ f & \mapsto & \big(f(P_1), \ldots, f(P_{N_1}), f'(P_1), \ldots, f'(P_{l_1}), f(P_{N_1+1}), \ldots, f(P_{N_1+N_2}), \\ & & \quad f'(P_{N_1+1}), \ldots, f'(P_{N_1+l_2}), f(P_{N_1+N_2+1}), \ldots, f(P_{N_1+N_2+N_4}), \\ & & \quad f'(P_{N_1+N_2+1}), \ldots, f'(P_{N_1+N_2+l_4})\big) \end{cases}$$

is injective.

Then

$$\mu_q(n) \leq N_1 + 2l_1 + 3N_2 + 6l_2 + \mu_q(4)\big(N_4 + 2l_4\big).$$

This specialization yields the followings bounds for the tensor rank of multiplication in any extension of $\mathbb{F}_2$:

Theorem 2.2. *Let q be a prime power. Let $F/\mathbb{F}_q$ be an algebraic function field of genus g and N_i be a number of places of degree i in $F/\mathbb{F}_q$. Let l_1, l_2, l_4 be three integers such that $0 \leq l_1 \leq N_1$, $0 \leq l_2 \leq N_2$ and $0 \leq l_4 \leq N_4$. If*

i) $N_n > 0$ (or $2g + 1 \leq q^{\frac{n-1}{2}}(q^{\frac{1}{2}} - 1)$),

ii) $N_1 + l_1 + 2(N_2 + l_2) + 4(N_4 + l_4) > 2n + 2g + 6$,

then

$$\mu_q(n) \leq \frac{\mu_q(4)}{2}\big(n + g + 5\big) + \mu_q(4) l_4.$$

In particular,

$$\mu_2(n) \leq \frac{9}{2}\big(n + g + 5\big) + 9 l_4.$$

2.2. **Tensor rank in any extension of $\mathbb{F}_2$.** Now we apply the results of the preceding section to the tower of Garcia-Stichtenoth $T_3/\mathbb{F}_2$ presented in [8]. We obtain two kinds of result: one which uses derivative evaluations and an other which does not. In fact we obtain a better bound for M_2 using derivative evaluations but this utilization is more complicated in practice and leads to an increase of linear complexity which can be inconvenient; so we present both technics. Moreover, although the best results are obtained using derivative evaluations, we still get an improvement of the best known bound for M_2 using simple evaluations.

2.2.1. *Bound for the tensor rank without using derivative evaluation.* First of all, we apply the bound of Theorem 2.2 on the tower $T_3/\mathbb{F}_2$ with $l_1 = l_2 = l_4 = 0$. This yields:

Theorem 2.3. *For any integer $n \geq 2$, we have*

$$\mu_2(n) \leq \frac{45}{2} n + 85.5.$$

2.2.2. *Bound for the tensor rank using derivative evaluations.* Here, we apply results of Theorem 2.2 with an optimal number of derivative evaluations.

Theorem 2.4. *For any integer $n \geq 2$, we have*

$$\mu_2(n) \leq \frac{477}{26}n + \frac{45}{2}.$$

3. New asymptotic bounds for the tensor rank

Without using derivative evaluation, we obtain from Theorem 2.3 the following bound for M_2:

$$M_2 \leq 22.5,$$

which is better than the best known bound recalled in Proposition 1.5.

However, by using derivative evaluations, we obtain a better bound for M_2, namely:

Theorem 3.1.

$$M_2 \leq \frac{477}{26} \approx 18.35.$$

Proof. It follows from Theorem 2.4. □

References

[1] Nicolas Arnaud. *Évaluations dérivées, multiplication dans les corps finis et codes correcteurs*. PhD thesis, Université de la Méditerranée, Institut de Mathématiques de Luminy, 2006.

[2] Stéphane Ballet. Curves with many points and multiplication complexity in any extension of $\mathbb{F}_q$. *Finite Fields and Their Applications*, 5:364–377, 1999.

[3] Stéphane Ballet. Low increasing tower of algebraic function fields and bilinear complexity of multiplication in any extension of $\mathbb{F}_q$. *Finite Fields and Their Applications*, 9:472–478, 2003.

[4] Stéphane Ballet and Jean Chaumine. On the bounds of the bilinear complexity of multiplication in some finite fields. *Applicable Algebra in Engineering Communication and Computing*, 15:205–211, 2004.

[5] Stéphane Ballet and Dominique Le Brigand. On the existence of non-special divisors of degree g and $g-1$ in algebraic function fields over $\mathbb{F}_q$. *Journal on Number Theory*, 116:293–310, 2006.

[6] Stéphane Ballet, Dominique Le Brigand, and Robert Rolland. On an application of the definition field descent of a tower of function fields. In *Proceedings of the Conference Arithmetic, Geometry and Coding Theory (AGCT 2005)*, volume 21, pages 187–203. Société Mathématique de France, sér. Séminaires et Congrès, 2009.

[7] Stéphane Ballet and Robert Rolland. Multiplication algorithm in a finite field and tensor rank of the multiplication. *Journal of Algebra*, 272(1):173–185, 2004.

[8] Stéphane Ballet and Robert Rolland. Families of curves over any finite field with a class number greater than the Lachaud - Martin-Deschamps bounds. ArXiv:0906.5432v1, 2009.

[9] Ignacio Cascudo, Ronald Cramer, and Chaoping Xing. Torsion-limits for tower and asymptotically good special codes in secure computation and complexity. 2009.

[10] Murat Cenk and Ferruh Özbudak. On multiplication in finite fields. *Journal of Complexity*, doi:10.1016/j.jco.2009.11.002, 2009.

[11] David Chudnovsky and Gregory Chudnovsky. Algebraic complexities and algebraic curves over finite fields. *Journal of Complexity*, 4:285–316, 1988.

[12] Arnaldo Garcia and Henning Stichtenoth. A tower of Artin-Schreier extensions of function fields attaining the Drinfeld-Vladut bound. *Inventiones Mathematicae*, 121:211–222, 1995.

[13] Igor Shparlinski, Michael Tsfasman, and Serguei Vladut. Curves with many points and multiplication in finite fields. In H. Stichtenoth and M.A. Tsfasman, editors, *Coding Theory and Algebraic Geometry*, number 1518 in Lectures Notes in Mathematics, pages 145–169, Berlin, 1992. Springer-Verlag. Proceedings of AGCT-3 Conference, June 17-21, 1991, Luminy.

Institut de Mathématiques de Luminy, case 930, F13288 Marseille cedex 9, France

eRISCS Groupe de Recherche "Informatique des Systèmes Communicants Sécurisés"
E-mail address: stephane.ballet@univmed.fr

Institut de Mathématiques de Luminy, case 930, F13288 Marseille cedex 9, France

eRISCS Groupe de Recherche "Informatique des Systèmes Communicants Sécurisés"
E-mail address: julia.pieltant@univmed.fr

QUADRATIC FORMS OF CODIMENSION 2 OVER CERTAIN FINITE FIELDS OF EVEN CHARACTERISTIC

FERRUH ÖZBUDAK, ELİF SAYGI, ZÜLFÜKAR SAYGI

ABSTRACT. Let $\mathbb{F}_q$ be a finite field of characteristic 2, not containing $\mathbb{F}_4$. Let $k \geq 2$ be an even integer. We give a full classification of quadratic forms over $\mathbb{F}_{q^k}$ of codimension 2 provided that certain three coefficients are from $\mathbb{F}_4$.

Key words and phrases: Quadratic form, Artin-Schreier type curve, maximal curve, minimal curve

Let q be an integer which is a power of 2. Let $k \geq 2$ be an integer and $m = \left\lfloor \frac{k}{2} \right\rfloor$, the integer part of $k/2$. Let $\epsilon_0, \epsilon_1, \ldots, \epsilon_m \in \mathbb{F}_{q^k}$. The map

$$\begin{array}{rccl} Q : \mathbb{F}_{q^k} & \to & \mathbb{F}_q \\ x & \mapsto & \mathrm{Tr}\left(x\left(\epsilon_0 x + \epsilon_1 x^q + \cdots + \epsilon_m x^{q^m}\right)\right) \end{array} \tag{0.1}$$

is a quadratic form over $\mathbb{F}_{q^k}$. It is well known that the quadratic form Q is related to Artin-Schreier type curve

$$\chi : \quad y^q + y = x\left(\epsilon_0 x + \epsilon_1 x^q + \cdots + \epsilon_m x^{q^m}\right). \tag{0.2}$$

Here we assume that at least one of $\epsilon_1, \ldots, \epsilon_m$ is nonzero and hence the genus of χ is positive.

The radical of Q is an $\mathbb{F}_q$-linear subspace of $\mathbb{F}_{q^k}$. There is another invariant $\Lambda(Q)$ of Q, which is an integer in the set $\{-1, 0, 1\}$. By Hilbert's Theorem 90 (cf. [4, Theorem 2.25]) and the basic theory of quadratic forms, it is well known that the radical and the value of $\Lambda(Q)$ determine the number of $\mathbb{F}_{q^k}$-rational points of χ.

If the codimension of the radical is 0, then the invariant $\Lambda(Q)$ of Q can be determined, actually also when the characteristic of $\mathbb{F}_q$ is odd (cf. [6]). The next case in characteristic 2 is the case of codimension 2 as the codimension must be an even integer. Recently we have considered this problem when $\mathbb{F}_q$ is an extension of $\mathbb{F}_4$ under the extra condition that certain coefficients are in $\mathbb{F}_4$ (see [6]).

We still keep such a condition. However we note that the quadratic form is considered in $\mathbb{F}_{q^k}$. Hence if q is an odd power of 2 and k is even, then the map (0.1) is still a quadratic form and this problem is not considered in [6]. In this study we consider this problem and obtain again a full classification of such quadratic forms of codimension 2 when q is an odd power of 2 and k is even. Our result is an extension of [1]. We use some methods from [1], which become more complicated in our study.

We note that combining this result with the ones of [6] we complete the problem of classification of quadratic forms of codimension 2 over arbitrary finite fields provided that the first three coefficients are from $\mathbb{F}_4$ for $k \geq 4$. As these coefficients are from $\mathbb{F}_4$, we have $\mathbb{F}_4 \subseteq \mathbb{F}_{q^k}$. The case $\mathbb{F}_4 \subseteq \mathbb{F}_q$ is studied in [6] and here we study the remaining case that q is an odd power of 2 and k is even.

Ferruh Özbudak is with Department of Mathematics and Institute of Applied Mathematics, Middle East Technical University, İnönü Bulvarı, 06531, Ankara, Turkey; e-mail: ozbudak@metu.edu.tr,

Elif Saygı is with Department of Math. Education, Hacettepe University, Beytepe, 06532, Ankara, Turkey; e-mail: esaygi@hacettepe.edu.tr.

Zülfükar Saygı is with Department of Mathematics, TOBB University of Economics and Technology, Söğütözü 06530, Ankara, Turkey; e-mail: zsaygi@etu.edu.tr.

In the rest of this paper we state our main result in this study. Here we also note that as in [6] using our main result below we obtain new results on the classification of maximal and minimal curves of the form (0.2) (cf., for example, [2, 7, 5, 8, 3], and also the references in [6]), when q is an odd power of 2 and k is even. We will explain them in the full version of this paper.

First we introduce some notation. For certain even integers $k \geq 4$, we define six types of polynomials $A_1(\epsilon_0, \epsilon_1)$, $A_2(\epsilon_0, \epsilon_1)$, $A_3(\epsilon_0, \epsilon_1, \epsilon_2)$, $A_4(\epsilon_0, \epsilon_1)$, $A_5(\epsilon_0, \epsilon_1)$ and $A_6(\epsilon_0, \epsilon_1, \epsilon_2)$ in $\mathbb{F}_4[x]$. For simplicity of notation we indicate neither the variable x nor their dependence on k in denoting these polynomials. Let $m = \frac{k}{2}$.

- If $4|k$, then for $\epsilon_0, \epsilon_1 \in \mathbb{F}_4$ let
$$A_1(\epsilon_0, \epsilon_1) = \epsilon_0 x + \epsilon_1 \left(x^q + x^{q^3} + \cdots + x^{q^{m-1}} \right).$$
- If $3|k$, then for $\epsilon_0, \epsilon_1 \in \mathbb{F}_4$ let
$$A_2(\epsilon_0, \epsilon_1) = \epsilon_0 x + \epsilon_1 \left(x^q + x^{q^2} + x^{q^4} + x^{q^5} + \cdots + x^{q^{m-2}} + x^{q^{m-1}} \right).$$
- If $6|k$, then for $\epsilon_0, \epsilon_1, \epsilon_2 \in \mathbb{F}_4$ let
$$\begin{aligned} A_3(\epsilon_0, \epsilon_1, \epsilon_2) = \epsilon_0 x &+ \epsilon_1 \left(x^q + x^{q^5} + x^{q^7} + x^{q^{11}} + \cdots \right) \\ &+ \epsilon_2 \left(x^{q^2} + x^{q^4} + x^{q^8} + x^{q^{10}} + \cdots \right). \end{aligned}$$
where last terms in $A_3(\epsilon_0, \epsilon_1, \epsilon_2)$ are $\epsilon_1 \left(x^{q^{m-5}} + x^{q^{m-1}} \right)$ and $\epsilon_2 \left(x^{q^{m-4}} + x^{q^{m-2}} \right)$ if $k \equiv 0 \mod 12$, and $\epsilon_1 \left(x^{q^{m-4}} + x^{q^{m-2}} \right)$ and $\epsilon_2 \left(x^{q^{m-5}} + x^{q^{m-1}} \right)$ if $k \equiv 6 \mod 12$.
- If $12|k$, then for $\epsilon_0, \epsilon_1 \in \mathbb{F}_4$ let
$$A_4(\epsilon_0, \epsilon_1) = \epsilon_0 x + \left(\epsilon_1 x^q + x^{q^3} + \epsilon_1^2 x^{q^5} + \epsilon_1 x^{q^7} + x^{q^9} + \epsilon_1^2 x^{q^{11}} + \cdots + \epsilon_1^2 x^{q^{m-1}} \right).$$
- If $8|k$, then for $\epsilon_0, \epsilon_1 \in \mathbb{F}_4$ let
$$A_5(\epsilon_0, \epsilon_1) = \epsilon_0 x + \epsilon_1 \left(x^q + x^{q^2} + \epsilon_1 x^{q^3} + x^{q^5} + x^{q^6} + \epsilon_1 x^{q^7} + \cdots + \epsilon_1 x^{q^{m-1}} \right).$$
- If $8|k$, then for $\epsilon_0, \epsilon_1, \epsilon_2 \in \mathbb{F}_4$ let
$$\begin{aligned} A_6(\epsilon_0, \epsilon_1, \epsilon_2) = \epsilon_0 x &+ \epsilon_1 \left(x^q + \epsilon_1 x^{q^3} + x^{q^5} + \epsilon_1 x^{q^7} + \cdots + x^{q^{m-3}} + \epsilon_1 x^{q^{m-1}} \right) \\ &+ \epsilon_2 \left(x^{q^2} + x^{q^6} + x^{q^{10}} + \cdots + x^{q^{m-2}} \right). \end{aligned}$$

Now we state our main result.

Theorem 1. *Let $q \geq 2$ be an integer which is an odd power of 2. Let $k \geq 4$ be an even integer and set $m = \frac{k}{2}$. Let $\epsilon_0, \epsilon_1, \epsilon_2 \in \mathbb{F}_4$, and for $k \geq 8$ let $\epsilon_3, \ldots, \epsilon_{m-1}, \epsilon_m \in \mathbb{F}_{q^k}$. Let Q be the quadratic form from $\mathbb{F}_{q^k}$ to $\mathbb{F}_q$ defined as*

$$Q(x) = \mathrm{Tr}\left(x \sum_{i=0}^{m} \epsilon_i x^{q^i} \right), \tag{0.3}$$

where Tr *is the trace map from $\mathbb{F}_{q^k}$ to $\mathbb{F}_q$. Moreover let r be the positive odd integer with $q = 2^r$. If the $\mathbb{F}_q$-dimension of the radical of Q is $k-2$, then exactly one of the followings holds:*

a) *If $\epsilon_1 = 1$, then*
   a.1) *$4|k$, $\epsilon_0 = 0$, and for $1 \leq i \leq m-1$ we have*
$$\epsilon_i = \begin{cases} \epsilon_1 & \text{if } i \equiv 1 \mod 2, \\ 0 & \text{otherwise.} \end{cases}$$

In particular the polynomial $\sum_{i=0}^{m-1} \epsilon_i x^{q^i}$ *is equal to* $R(x) = A_1(\epsilon_0, \epsilon_1)$. *Moreover the invariant* $\Lambda(Q)$ *of* Q *is equal to 1 in this case.*

a.2) $4|k$, $\epsilon_0 \neq 0$, *and for* $1 \leq i \leq m-1$ *we have*

$$\epsilon_i = \begin{cases} \epsilon_1 & \text{if } i \equiv 1 \mod 2, \\ 0 & \text{otherwise.} \end{cases}$$

In particular the polynomial $\sum_{i=0}^{m-1} \epsilon_i x^{q^i}$ *is equal to* $R(x) = A_1(\epsilon_0, \epsilon_1)$. *Moreover the invariant* $\Lambda(Q)$ *of* Q *is equal to -1 in this case.*

a.3) $3|k$, $\epsilon_0 = \epsilon_1$, *and for* $1 \leq i \leq m-1$ *we have*

$$\epsilon_i = \begin{cases} \epsilon_1 & \text{if } i \equiv 1 \text{ or } 2 \mod 3, \\ 0 & \text{otherwise.} \end{cases}$$

In particular the polynomial $\sum_{i=0}^{m-1} \epsilon_i x^{q^i}$ *is equal to* $R(x) = A_2(\epsilon_0, \epsilon_1)$. *Moreover the invariant* $\Lambda(Q)$ *of* Q *is equal to -1 in this case.*

a.4) $3|k$, $\epsilon_0 \neq \epsilon_1$, *and for* $1 \leq i \leq m-1$ *we have*

$$\epsilon_i = \begin{cases} \epsilon_1 & \text{if } i \equiv 1 \text{ or } 2 \mod 3, \\ 0 & \text{otherwise.} \end{cases}$$

In particular the polynomial $\sum_{i=0}^{m-1} \epsilon_i x^{q^i}$ *is equal to* $R(x) = A_2(\epsilon_0, \epsilon_1)$. *Moreover the invariant* $\Lambda(Q)$ *of* Q *is equal to 0 in this case.*

a.5) $6|k$, $\epsilon_2 \notin \{0, \epsilon_1\}$, $\epsilon_0 = \epsilon_2$, *and for* $1 \leq i \leq m-1$ *we have*

$$\epsilon_i = \begin{cases} \epsilon_1 & \text{if } i \equiv 1 \text{ or } 5 \mod 6, \\ \epsilon_2 & \text{if } i \equiv 2 \text{ or } 4 \mod 6, \\ 0 & \text{otherwise.} \end{cases}$$

In particular the polynomial $\sum_{i=0}^{m-1} \epsilon_i x^{q^i}$ *is equal to* $R(x) = A_3(\epsilon_0, \epsilon_1, \epsilon_2)$. *Moreover the invariant* $\Lambda(Q)$ *of* Q *is equal to -1 in this case.*

a.6) $6|k$, $\epsilon_2 \notin \{0, \epsilon_1\}$, $\epsilon_0 \neq \epsilon_2$, *and for* $1 \leq i \leq m-1$ *we have*

$$\epsilon_i = \begin{cases} \epsilon_1 & \text{if } i \equiv 1 \text{ or } 5 \mod 6, \\ \epsilon_2 & \text{if } i \equiv 2 \text{ or } 4 \mod 6, \\ 0 & \text{otherwise.} \end{cases}$$

In particular the polynomial $\sum_{i=0}^{m-1} \epsilon_i x^{q^i}$ *is equal to* $R(x) = A_3(\epsilon_0, \epsilon_1, \epsilon_2)$. *Moreover the invariant* $\Lambda(Q)$ *of* Q *is equal to 0 in this case.*

b) *If* $\epsilon_1 \in \mathbb{F}_4 \setminus \mathbb{F}_2$, *then*

b.1) $12|k$, $\epsilon_0 = 0$, *and for* $1 \leq i \leq m-1$ *we have*

$$\epsilon_i = \begin{cases} \epsilon_1 & \text{if } i \equiv 1 \mod 6, \\ 1 & \text{if } i \equiv 3 \mod 6, \\ \epsilon_1^2 & \text{if } i \equiv 5 \mod 6, \\ 0 & \text{otherwise.} \end{cases}$$

In particular the polynomial $\sum_{i=0}^{m-1} \epsilon_i x^{q^i}$ *is equal to* $R(x) = A_4(\epsilon_0, \epsilon_1)$. *Moreover the invariant* $\Lambda(Q)$ *of* Q *is equal to -1 in this case.*

b.2) $12|k$, $\epsilon_0 \neq 0$, *and for* $1 \leq i \leq m-1$ *we have*

$$\epsilon_i = \begin{cases} \epsilon_1 & \text{if } i \equiv 1 \mod 6, \\ 1 & \text{if } i \equiv 3 \mod 6, \\ \epsilon_1^2 & \text{if } i \equiv 5 \mod 6, \\ 0 & \text{otherwise.} \end{cases}$$

In particular the polynomial $\sum_{i=0}^{m-1} \epsilon_i x^{q^i}$ *is equal to* $R(x) = A_4(\epsilon_0, \epsilon_1)$. *Moreover the invariant* $\Lambda(Q)$ *of* Q *is equal to 0 in this case.*

b.3) $8|k$, $\epsilon_0 = \epsilon_1^{r+1}$, *and for* $1 \leq i \leq m-1$ *we have*

$$\epsilon_i = \begin{cases} \epsilon_1 & \text{if } i \equiv 1 \text{ or } 2 \mod 4, \\ \epsilon_1^2 & \text{if } i \equiv 3 \mod 4, \\ 0 & \text{otherwise.} \end{cases}$$

In particular the polynomial $\sum_{i=0}^{m-1} \epsilon_i x^{q^i}$ *is equal to* $R(x) = A_5(\epsilon_0, \epsilon_1)$ *Moreover the invariant* $\Lambda(Q)$ *of* Q *is equal to 1 in this case.*

b.4) $8|k$, $\epsilon_0 = \epsilon_1^{r-1}$, *and for* $1 \leq i \leq m-1$ *we have*

$$\epsilon_i = \begin{cases} \epsilon_1 & \text{if } i \equiv 1 \text{ or } 2 \mod 4, \\ \epsilon_1^2 & \text{if } i \equiv 3 \mod 4, \\ 0 & \text{otherwise.} \end{cases}$$

In particular the polynomial $\sum_{i=0}^{m-1} \epsilon_i x^{q^i}$ *is equal to* $R(x) = A_5(\epsilon_0, \epsilon_1)$. *Moreover the invariant* $\Lambda(Q)$ *of* Q *is equal to -1 in this case.*

b.5) $8|k$, $\epsilon_0 \in \{0, \epsilon_1\}$, *and for* $1 \leq i \leq m-1$ *we have*

$$\epsilon_i = \begin{cases} \epsilon_1 & \text{if } i \equiv 1 \text{ or } 2 \mod 4, \\ \epsilon_1^2 & \text{if } i \equiv 3 \mod 4, \\ 0 & \text{otherwise.} \end{cases}$$

In particular the polynomial $\sum_{i=0}^{m-1} \epsilon_i x^{q^i}$ *is equal to* $R(x) = A_5(\epsilon_0, \epsilon_1)$. *Moreover the invariant* $\Lambda(Q)$ *of* Q *is equal to 0 in this case.*

b.6) $8|k$, $\big(r \equiv 1 \mod 4$ *and* $(\epsilon_0, \epsilon_2) \in \{(\epsilon_1, 1), (1, \epsilon_1^2)\}\big)$ *or* $\big(r \equiv 3 \mod 4$ *and* $(\epsilon_0, \epsilon_2) \in \{(\epsilon_1, \epsilon_1^2), (\epsilon_1^2, 1)\}\big)$ *and for* $1 \leq i \leq m-1$ *we have*

$$\epsilon_i = \begin{cases} \epsilon_1 & \text{if } i \equiv 1 \mod 4, \\ \epsilon_2 & \text{if } i \equiv 2 \mod 4, \\ \epsilon_1^2 & \text{if } i \equiv 3 \mod 4, \\ 0 & \text{otherwise.} \end{cases}$$

In particular the polynomial $\sum_{i=0}^{m-1} \epsilon_i x^{q^i}$ *is equal to* $R(x) = A_6(\epsilon_0, \epsilon_1, \epsilon_2)$. *Moreover the invariant* $\Lambda(Q)$ *of* Q *is equal to 1 in this case.*

b.7) $8|k$, $\big(r \equiv 1 \mod 4$ *and* $(\epsilon_0, \epsilon_2) \in \{(\epsilon_1, \epsilon_1^2), (\epsilon_1^2, 1)\}\big)$ *or*
$\big(r \equiv 3 \mod 4$ *and* $(\epsilon_0, \epsilon_2) \in \{(\epsilon_1, 1), (1, \epsilon_1^2)\}\big)$ *and for* $1 \leq i \leq m-1$ *we have*

$$\epsilon_i = \begin{cases} \epsilon_1 & \text{if } i \equiv 1 \mod 4, \\ \epsilon_2 & \text{if } i \equiv 2 \mod 4, \\ \epsilon_1^2 & \text{if } i \equiv 3 \mod 4, \\ 0 & \text{otherwise.} \end{cases}$$

In particular the polynomial $\sum_{i=0}^{m-1} \epsilon_i x^{q^i}$ *is equal to* $R(x) = A_6(\epsilon_0, \epsilon_1, \epsilon_2)$. *Moreover the invariant* $\Lambda(Q)$ *of* Q *is equal to -1 in this case.*

b.8) $8|k$, $\epsilon_2 \notin \{0, \epsilon_1\}$, $\epsilon_0 \in \{0, \epsilon_2\}$ *and for* $1 \leq i \leq m-1$ *we have*

$$\epsilon_i = \begin{cases} \epsilon_1 & \text{if } i \equiv 1 \mod 4, \\ \epsilon_2 & \text{if } i \equiv 2 \mod 4, \\ \epsilon_1^2 & \text{if } i \equiv 3 \mod 4, \\ 0 & \text{otherwise.} \end{cases}$$

In particular the polynomial $\sum_{i=0}^{m-1} \epsilon_i x^{q^i}$ *is equal to* $R(x) = A_6(\epsilon_0, \epsilon_1, \epsilon_2)$. *Moreover the invariant* $\Lambda(Q)$ *of* Q *is equal to 0 in this case.*

For each of these 14 cases above, the coefficient ϵ_m *is determined uniquely in* $\mathbb{F}_{q^k}$ *modulo* $\mathbb{F}_{q^m}$ *and hence map defined by* $\mathrm{Tr}\left(xR(x) + \gamma x^{q^m+1}\right)$, *where* γ *is an arbitrary element of* $\mathbb{F}_{q^k}$ *with* $(\gamma - \epsilon_m) \in \mathbb{F}_{q^m}$, *gives the same quadratic form as in (0.3).*

Conversely, for each of the 14 cases above, there exist $\epsilon_0, \epsilon_1, \epsilon_2, \cdots, \epsilon_{m-1}, \epsilon_m \in \mathbb{F}_4 \subseteq \mathbb{F}_{q^k}$ *satisfying the corresponding conditions. Therefore we have quadratic forms as in (0.3) from* $\mathbb{F}_{q^k}$ *to* $\mathbb{F}_q$, *even with coefficients from* $\mathbb{F}_4$, *such that their radicals are of codimension 2 and their invariants* $\Lambda(Q)$ *are given as in the 14 cases above.*

From the proof of Theorem 1 we easily obtain that if $k = 2$ and $\epsilon_0, \epsilon_1 \in \mathbb{F}_{q^k}$, then there is no quadratic form over $\mathbb{F}_{q^k}$ of codimension 2 in the form (0.1), when q is an odd power of 2.

Finally we note that the conditions on the main results in Theorem 1 above and [6, Theorem 3.1] are very different.

References

[1] R. W. Fitzgerald, Highly Degenerate Quadratic Forms over finite fields of characteristic 2, *Finite Fields Appl.*, vol. 11, pp. 165-181, (2005).

[2] J. W. P. Hirschfeld, *Projective Geometries over Finite Fields*, Oxford Univ. Press, New York, (1998).

[3] J. W. P. Hirschfeld, G. Korchmáros, F. Torres, *Algebraic Curves over a Finite Field*, Princeton Univ. Press, Princeton, (2008).

[4] R. Lidl and H. Niederreiter, *Finite fields*, Cambridge University Press, Cambridge, (1997).

[5] H. Niederreiter and C. Xing, *Rational Points on Curves over Finite Fields: Theory and Applications*, Cambridge Univ. Press, Cambridge, (2001).

[6] F. Özbudak, E. Saygı and Z. Saygı, Quadratic Forms of Codimension 2 over Finite Fields Containing $\mathbb{F}_4$ and Artin-Schreier Type Curves, submitted.

[7] H. Stichtenoth, *Algebraic Function Fields and Codes*, Springer, Berlin, (2008).

[8] M. A. Tsfasman, S. G. Vladut, D. Nogin *Algebraic Geometric Codes: Basic Notations*, Mathematical Surveys and monographs vol. 139, American Mathematical Society, Providence, RI, (2007).

DEMOCRACY—A Heuristic for Polynomial Systems of Equations over Finite Fields (Extended Abstract)

Gregory V. Bard

April 30, 2010

Abstract

This paper presents a heuristic for solving polynomial systems of equations on finite fields larger than GF(2), via stochastic local search (SLS). It was inspired by the SLS-based SAT-solvers G-SAT, Walk-SAT and Tabu-SAT. Called DEMOCRACY, the equations vote on which values for a given variable will satisfy as many as possible of them. Variables, one at a time, are thusly changed from an initial random setting, until all equations are satisfied.

We show that this algorithm can solve the Brent Equations, a cubic system of 64 equations and 84 unknowns, taken mod 7, in 2–8 minutes. On this system, MAGMA's F4 algorithm crashed for lack of memory, and Singular via SAGE timed out after 10 hours. Similar results have been observed for random quadratic systems of equations also.

Key Words: polynomial system, finite fields, stochastic local search algebraic cryptanalysis, the Brent equations.

1 Introduction

The study of the solution of polynomial systems of equations over GF(2) has attracted a great deal of attention in the past decade, particularly because of multivariate cryptography and block-cipher cryptanalysis. In fact, there are too many papers to cite. But this raises the question of other finite fields. Why should GF(3), GF(5) or GF(7) be so different? Many papers have been written about solving polynomial systems over finite fields in general, and the method-of-choice for such problems is Jean-Charles Faugere's algorithm known as F4 [**?**]. Furthermore, it is known that the implementation in MAGMA [MAG] is the best available at this time. Here is presented an alternative approach.

Several features of problems to be solved (via polynomial systems of equations) in multivariate cryptography, error-correcting codes, algebraic cryptanalysis, graph coloring, or combinatorial designs, are notable and unusual. The first and most pivotal is that one is usually interested only in those solutions inside the base field, not inside the algebraic closure. However, this can be easily remedied by adding the equation $x^q - x = 0$ for all variables x, where q is the size of the base field. Second, in most of the above cases, we know a solution exists and we merely wish to find it. The case of "no solution" does not occur—for example in algebraic cryptanalysis a message was indeed sent, and so it was sent under some key. Third, if there are multiple solutions, we need only one solution. It is often stated that a problem with a large but finite number of distinct solutions is the most difficult for any Gröbner Bases approach. This is discussed in the paper [Bar08]. Fourth, the problems are often huge, so the usage of memory can be the limiting factor in the feasibility of the computation.

Over GF(2), while F4 is often faster, SAT-solvers often address these issues. Here we present an interesting problem from the theory of matrix multiplication (which we call "The Brent Equations" in honor of Richard Brent, see Table 1.) The method in this paper—called DEMOCRACY—solves a system of equations

$$\sum_{k=1}^{k=s} \alpha_{wxk}\beta_{yzk}\gamma_{ijk} = \delta_{iw}\delta_{xy}\delta_{jz}$$

$$\forall \left\{ \begin{array}{ll} 1 \leq i \leq c & 1 \leq w \leq a \\ 1 \leq x \leq b & 1 \leq y \leq b \\ 1 \leq z \leq c & 1 \leq j \leq c \end{array} \right.$$

Table 1: The Brent Equations for Rectangular Shortcuts

related to this theory, in 8 minutes. The system has 64 equations, 84 unknowns, and is taken over GF(7). Each equation is a cubic form set to either zero or one. Meanwhile, on the same system, MAGMA crashes due to allocating too much memory, and SAGE/Singular [S^+] [sin] timed-out after 10 hours. Each of the four above conditions is represented here. First, that we only desire base-field solutions. But, the "field equations" of the form $x^7 - x = 0$ are problematic, because they are degree seven, namely 4 degrees above the equations of the system. Second, we know a solution exists (more detail will be given in Section 5.1). Third, one solution is as useful as knowing several solutions. Fourth, our method uses only 4 megabytes. Just like SAT-solvers, DEMOCRACY is not in need of much memory. These comparisons are not surprising, because this paper's approach was inspired by the SAT-solver community.

A SAT-solver takes a series of logic statements, called clauses, written in many variables. Clauses can be composed of "implies", "and", "or", "not", as well as "if and only if", but not "for all" nor "there exists. The SAT-solver tries to solve them by assigning a truth value (either true or false) to each variable, so that every clauses comes to evaluate as true. While this is an NP-Complete problem, for thousands of variables, and hundreds of thousands of clauses, many SAT-solvers can respond within minutes.

The world of SAT-solvers is divided into two categories. The first are the conflict-driven solvers, such as minisat [ES05], which have demonstrated their usefulness in solving polynomial equations mod 2 [BCJ06]. This category has dominated the SAT world for about a decade. The second are those that work off of stochastic local search, and it is these which the author sought to explore. These too have been tried in cryptanalysis, but with rather limited success [Mas99] [MM99].

Essentially, the first stochastic local search SAT-solver was G-SAT [SLM92], which evolved into Walk-SAT almost immediately [SKC93], and that had many descendants which were mostly minor variations. An ancestor of these is [Pap91]. Rather than convert polynomial systems mod odd primes into SAT problems, it soon became apparent that it would be easier to take the methodologies and principles of those algorithms and heuristics from the SLS-SAT world and apply them to the special case of polynomial systems of equations over finite fields larger than $\mathbb{GF}(2)$, by making a new, custom algorithm.

1.1 Target Problems

There are four main target problems in this research. The first, is to be able to conduct algebraic cryptanalysis (as well as hash-function attacks) on cryptosystems already based on finite fields of odd characteristic. The second is graph coloring. The third are the Brent Equations [Bre70], given in Table 1, explained later and in [Bar09]. The fourth are certain problems in combinatorial designs. Here the graph theory and design theory examples are omitted, as they are not near completion.

2 DEMOCRACY

In this section is described the basic heuristic. Later sections will add features to it and optimizations. Perhaps one would like to call DEMOCRACY an algorithm, but there is no reason to expect it to terminate in general, so one cannot. In fact, for systems with empty solution set, it will run forever. For this reason, we are compelled to call it a "heuristic".

2.1 Inspiration

Suppose one has a system of equations over a finite field. Fix a variable of interest. Perhaps

this variable, x, appears only with exponent 1 or 2 in the system. Then, for any equation that has an x, it can be rewritten as a quadratic or linear polynomial in x alone, in the following sense:

$$w^4y^3x+3x^2y^2+4wz^3x-x^2z+3wyz+w^5+w^2y^2$$

can be made to become

$$(3y^2-z)x^2+(w^4y^3+4wz^3)x+(3wyz+w^5+w^2y^2)$$

so if we treat x as unknown and the other variables as known, we have an ordinary quadratic equation.

Now if x appears with degree 4 or less, with the quadratic equation and the formulas of Gerolamo Cardano (1501–1576), we can solve for x explicitly—if all the other variables are known. Let us consider the quadratic equation:

$$ax^2 + bx + c = 0 \rightarrow \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

becomes in the finite field world

$$ax^2 + bx + c = 0 \rightarrow \left(-b + QR(b^2 - 4ac)\right)(2a)^{-1}$$

where $QR(n)$ is any of the set of elements in the finite field that square to n. If $n = 0$ this will be 0, otherwise there will be either 2 solutions or no solutions. It is easy to implement QR as an array lookup. Cardano's cubic equation is not quite as easy as the quadratic equation but we have implemented it, based on the entry in Dummit & Foote [DF03, Ch. 14.6], and the assistance of Kyle Kloster, a Fordham undergraduate in the mathematics department.

One might imagine that if x appears with degree 5, then because of the work of Évariste Galois (1811–1832) on the quintic equation, we do not have any formula that will work all the time. But one could, if desperate, loop through all finite field values. This is obviously computationally expensive. A must wiser alternative are two algorithms, one by Berlekamp [Ber67], and the other by Cantor–Zassenhaus [CZ81], which can solve polynomials over finite fields, regardless of degree, if in one variable. They are not polynomial time in general, but are for any fixed degree (this is called pseudo-polynomial time). Furthermore, the multiplicative groups of finite fields are cyclic, and so all Galois groups over them are solvable groups. Due to a result of David Dummit [Dum91], this can be used to solve the equations.

2.2 An Election

First, let us have some temporary assignment of values to the variables, which satisfy many but not all of the equations. Therefore the system of equations is not yet solved, and so perhaps we decide by some mechanism to be defined later that x is a variable that we would like to change.

We could solve each equation by the method given in the previous subsection, and see what we get for each one. There is one "ballot box" for each finite field value, and one for "absention". These are just mere counters, of course.

- If x does not appear in the equation, then the equation votes for abstention. This is because the equation either looks like $0 = 0$, and so is satisfied regardless of the value of x (and is thus already satisfied), or it is of the form $1 = 0$, and so is not satisfied and cannot become so.
- Note that x can appear with coefficients which "cancel out" or which become zero in the temporary assignment, and so the above bullet happens a bit often.
- If x appears with highest exponent as 1, 2, or 3, then simple algebra, the quadratic equation or Cardano's equation will suffice, because the resulting polynomial is degree 1, 2, or 3. The equation places a vote in the ballot box for each finite field value, such that if x is set to this, it would result in the equation being satisfied.
- At the moment, higher degree for x is impossible, as the Brent Equations are cubic, and multivariate cryptography is usually based on quadratic systems of equations. However, there are techniques to address higher degree, mentioned earlier.

This is a mathematical description of the process. In reality, the votes for any equation can

be computed *far more rapidly* via shortcuts for particular problems, and this is key to the efficiency of the heuristic. For example, in the Brent Equations, focusing on one variable results in a linear equation. A simple formula will yield the unique vote.

Why Do This? If you take the number of votes in the ballot box of the finite field value with the most votes (ignoring the abstentions box and the box for the current value of x), and subtract form it the number votes in the ballot box of the current value of x in the temporary solution, then you get the net number of newly satisfied equations. By that is meant the number that went from unsatisfied to satisfied, minus the number that went from satisfied to unsatisfied. Of course, this might be positive, zero, or negative, and we call this the "mandate" of the election. The finite field value with the most votes (excluding abstentions and the current value of x) is called the "winner" of the election. However, no other finite field value, besides the current value of x, would produce a higher mandate than the winner—though there may be ties.

2.3 An Iteration

The core of the heuristic is the following structure, which we call a "basic iteration." First, a random unsatisfied equation is selected. Second, an election is held for each variable in that equation. The election with the highest mandate is considered the "best election", and the variable of that election is set to a new value—the winner of that election. The iteration process then repeats, however many small counters and arrays that track minor features of the algorithm (for efficiency of other steps) are updated at this time.

Notes:

- The 'best election' is a global, not local, property. Indeed, the randomly chosen unsatisfied equation might not become satisfied by the changes made. What is ensured is that the number given by newly satisfied equations minus newly unsatisfied equations is as high as can be for any choice of x, except possibly its current value.

- The first paper of this type for SAT-solvers involved G-SAT [SLM92], which picks a random variable among those found in unsatisfied clauses. However, Walk-SAT [SKC93] choses an unsatisfied clause first, and then tries a random variable from that clause. The latter method, it turns out, is heavily biased among those variables which appear *frequently* versus *rarely* in unsatisfied clauses, (a very good feature) without actually tabulating that frequency! After much preliminary experimentation, we found it is better to be "greedy" and consider all variables in one unsatisfied clause, and take the one with the best mandate, though future experiments might be helpful in clarifying this.

- In the end, the best election has a mandate that is almost always -1 or better. Surely the iterations with best election mandate of 1 or more are ideal. After all, there are only finitely many equations, and if each iteration moves at least one closer to having all the equations satisfied, then eventually all equations are satisfied. Data on this point is tabulated by the program, and is available on request.

- Because the best mandate is often 0, numerous tie-breaking strategies have been explored. In fact, ties are very frequent. That will be described in a later section.

3 Assassinations

Walk-SAT [SKC93] also introduced noise into the general G-SAT structure [SLM92]. With some fixed probability, a randomly selected variable was simply flipped (recall, that is over $\mathbb{GF}(2)$. I also have a parameter for that probability, set at the command line. But after selecting a variable at random, an election is held. In $\mathbb{GF}(2)$ when you want to change a variable, there is no discussion as to what to change it

to—because there are only two values possible. But in larger finite fields, one must find a value to change it to. The electoral analog is that occasionally a politician dies in office, and so a special election must simply be called to fill the vacancy. A rate of approximately 25% turned out to be optimal.

4 Tie-Breakers

One possibility is that the algorithm could get stuck exploring a small region of the search space in an infinite loop. More precisely, perhaps the same set of three or four variables are being changed, while all other variables are not changing at all.

Traditionally, the approach used to prevent this is called Tabu-SAT [SHS03], which means that the last t variables to be changed, for some positive fixed constant t, cannot be the variable that gets changed (wins any election). The "runner-up" (second highest in votes) would be considered instead. This is particularly problematic if an equation has fewer than t clauses, and all variables in the equation have been taken recently and are thus tabu. But this was very far from effective in experiments.

Originally we defined the "age" of a variable in order to make the parameter t superfluous. The age of a variable is the number of iterations since it was last changed. If one assigns, among the n variables, the integers $\{-n+1, -n+2, -n+3, \ldots, 0\}$ as their default initial birthdays (randomly and without repeat), and one resets the birthday when the variable is changed, then one can entirely prevent ties. More plainly, we chose the election with the largest mandate, and in the event of a tie, we choose the older or younger variable to be changed, based on an age-bias parameter set at the command line. Now, further ties are impossible (since each variable always has a distinct age) and this prevents small infinite loops. Thus it is intuitive that the older-bias did experimentally better, avoiding stagnation of older variables and enlarging the search area.

Experiments proved birthdays vastly superior to tabu in running time for fixed problems. We attempt to explain this as follows. If one variable has been changed "recently" and has a mandate of $+500$, but another has not been changed in a while and has mandate -1, and all others are worse, then tabu would pass up the $+500$ and select instead the -1, while brithdays would not come into play because $+500 \neq -1$.

5 Experimental Results

Numerous experiments were performed to compare and contrast various tie-breaking strategies. The performance of the best choice was often an order of magnitude superior to the worst choice of strategy. Space forbids the details of such experiments appearing here.

5.1 The Brent Equations

The Brent Equations were solved by this method in the case $a = 2$, $b = 2$, $c = 2$ and $s = 7$. They can be taken over any field at all, and we chose GF(7). A solution of the Brent equations with $a = b = c = n$ yields an algorithm for $N \times N$ matrix multiplication in $\Theta(N^{\frac{\log s}{\log n}})$ time, over that same field and all its extension fields. Thus, our solution is an algorithm for matrix multiplication in fields of characteristic seven that runs in $\Theta(N^{\log_2 7}) = \Theta(N^{2.807})$ time—i.e. a repeat of Strassen's Algorithm. It is for this reason that we knew a solution existed to the equations, because Strassen's exists. It should be noted that DEMOCRACY, unlike F4, will loop forever on an unsolvable problem. Meanwhile, Singular via SAGE (running on computers provided by William[1] Stein) timed out after 10 hours. Furthermore, MAGMA, running under a limit of 8 gigabytes, ran out of memory and crashed. On the same machine as SAGE, the DEMOCRACY run was 2.5 minutes, and on the author's laptop, it was 8 minutes.

[1]Thanks for the National Science Foundation, Division of Mathematical Sciences, Grant No. DMS-0821725.

References

[Bar08] Gregory Bard, *Extending SAT-Solvers to low-degree extension fields of* $\mathbb{GF}(2)$, Presented at the Central European Conference on Cryptography, 2008, Available at `http://www.math.umd.edu/~bardg/extension_fields.pdf`.

[Bar09] ———, *New practical approximate matrix multiplication algorithms found via solving a system of cubic equations*, pre-print, 2009, Available at `http://www-users.math.umd.edu/~bardg/early_release.pdf`.

[BCJ06] Gregory Bard, Nicolas Courtois, and Christopher Jefferson, *Efficient methods for conversion and solution of sparse systems of low-degree multivariate polynomials over* $\mathbb{GF}(2)$ *via SAT-Solvers.*, Cryptology ePrint Archive, Report 2007/024, 2006, Available at `http://eprint.iacr.org/2007/024.pdf`.

[Ber67] Elwyn R. Berlekamp, *Factoring polynomials over finite fields*, Bell System Technical Journal **46** (1967).

[Bre70] Richard Brent, *Algorithms for matrix multiplication*, Tech. Report Report TR-CS-70-157, Department of Computer Science, Stanford, March 1970.

[CZ81] David G. Cantor and Hans Zassenhaus, *A new algorithm for factoring polynomials over finite fields*, Mathematics of Computation **36** (1981), 587–592.

[DF03] David Dummit and Richard Foote, *Abstract algebra*, third ed., Wiley, 2003.

[Dum91] David Dummit, *Solving solvable quintics*, Mathematics of Computation **57** (1991), no. 195, 387–401.

[ES05] Niklas Eén and Niklas Sörensson, *Minisat — a SAT solver with conflict-clause minimization*, Proc. Theory and Applications of Satisfiability Testing (SAT'05) (Fahiem Bacchus and Toby Walsh, eds.), Lecture Notes in Computer Science, vol. 3569, Springer-Verlag, 2005, pp. 61–75.

[MAG] *Magma*, Software Package, Available at `http://magma.maths.usyd.edu.au/magma/`.

[Mas99] Fabio Massacci, *Using Walk-SAT and Rel-SAT for cryptographic key search*, Proc. 16th International Joint Conference on Artificial Intelligence (Thomas Dean, ed.), Morgan Kaufmann Publishing, 1999, pp. 290–295.

[MM99] Laura Marraro and Fabio Massacci, *Towards the formal verification of ciphers: Logical cryptanalysis of DES*, Proc. Third LICS Workshop on Formal Methods and Security Protocols, Federated Logic Conferences (FLOC-99), 1999.

[Pap91] C. H. Papadimitriou, *On selecting a satisfying truth assignment*, Proceedings of the Conference on the Foundations of Computer Science (FOCS'91), IEEE, 1991, pp. 163–169.

[S$^+$] William A. Stein et al., *Sage mathematics software (version 4.3)*, Software Package, Available at `http://www.sagemath.org/`.

[SHS03] Kevin Smyth, Holger H. Hoos, and Thomas Stützle, *Iterated robust tabu search for max-sat*, Advances in Artificial Intelligence (Y. Xiang and Chaib-Draa, eds.), Lecture Notes in Computer Science, vol. 2671, Springer-Verlag, 2003, pp. 499–519.

[sin] *Singular*, Software Package, Available at `http://www.singular.uni-kl.de/`.

[SKC93] Bart Selman, Henry Kautz, and Bram Cohen, *Local search strategies for satisfiability testing*, Cliques, Coloring, and

Satisfiability: Second DIMACS Implementation Challenge (DIMACS'93) (David S. Johnson and Michael A. Trick, eds.), vol. 26, AMS, 1993.

[SLM92] B. Selman, H. J. Levesque, and D. G. Mitchell, *A new method for solving hard satisfiability problems*, 10th National Conference on Artificial Intelligence (AAAI'92), 1992, pp. 440–446.

The Equivalence Problem for Functions $F : \mathbb{F}_{p^n} \longrightarrow \mathbb{F}_{p^m}$

Alexander Pott

Institute for Algebra and Geometry, Faculty of Mathematics, Otto-von-Guericke-University Magdeburg, D-39016 Magdeburg, Germany

Let $F : \mathbb{F}_{p^n} \to \mathbb{F}_{p^m}$ be a function. Motivated by many applications, the following numbers are of interest, which measure the differential and the linear properties of F: The numbers

$$\delta_F(a,b) := \sharp\{x \in \mathbb{F}_{p^n} \ : \ F(x+a) - F(x) = b\}$$

measure the differential properties, and the numbers

$$\mathcal{W}_F(a,b) := \sum_{x \in \mathbb{F}_{p^n}, y \in \mathbb{F}_{p^m}} \zeta_p^{trace(ax+bF(x))}$$

describe the nonlinearity properties of F. The goal is that the maximum of all $\delta_F(a,b)$ $(a \neq 0)$ is small, and that the maximum of the $|\mathcal{W}_F(a,b)|$ is small $(b \neq 0)$.

Functions where these maxima are small include the important classes of bent functions (both p odd and $p = 2$), and perfect nonlinear (for p odd), almost perfect nonlinear as well as almost bent functions (for $p = 2$).

Definition 1. *A function $F : \mathbb{F}_{p^n} \to \mathbb{F}_{p^m}$ is* **bent** *if $\delta_F(a,b) = p^{n-m}$ for all $a \neq 0$. Bent functions with $n = m$ are called* **perfect nonlinear** *or* **planar**. *If $n = m$, $p = 2$ and $\delta_F(a,b) \in \{0,2\}$ for all $a \neq 0$, then F is called* **almost perfect nonlinear**.

There are several concepts of equivalence for functions F. We identify $\mathbb{F}_{p^n}$ with the vector space $\mathbb{F}_p^n$, and we define the **graph** G_F of F:

$$\{(x, F(x)) \ : \ x \in \mathbb{F}_p^n\} \subseteq \mathbb{F}_p^{n+m}.$$

Definition 2. *Two functions $F_1, F_2 : \mathbb{F}_p^n \to \mathbb{F}_p^m$ are called* **affine equivalent** *if there are linear permutations P and Q and an affine mapping L such that $F_1 = P \circ F_2 \circ Q + L$. The functions are called* **graph (or CCZ) equivalent** *if there is a linear permutation $H : \mathbb{F}_p^{n+m} \to \mathbb{F}_p^{n+m}$ such that $H(G_{F_1}) = G_{F_2} + v$ for some $v \in \mathbb{F}_p^{n+m}$, where $G_F + v := \{(x+a, F(x)+b) \ : \ x \in \mathbb{F}_p^n\}$, $v = (a,b)$.*

Any function defines a **design**, which is an incidence structure consisting of points and blocks (whcih are subsets of points): The points are the elements in $\mathbb{F}_p^{n+m}$, the lines are the "translates" $G_F + v$.

Definition 3. *Two functions F_1 and F_2 are called* **isomorphic** *if the corresponding designs are isomorphic.*

In general. isomorphism is an equivalence relation which is coarser than graph equivalence which is coarser than extended affine equivalence. However, in some cases the concepts coincide. I my talk, I will discuss relations between these concepts of equivalence for various classes of functions.

It may be possible to define even more general equivalence relations. There is a well known object ("Knuth cube") studied in connection with perfect nonlinear functions. I will discuss possible extensions of this concept to the almost perfect nonlinear case, which is related to the work of Taniguchi.

Another possible extension is the "switching equivalence" introduced by Edel and Pott.

The introduction of new concepts of equivalence is not just of academic interest in order to combine equivalence classes with respect to graph or extended affine equivalence. We may actually use them to construct new examples.

Part of the talk is based on joint work with my student Yue Zhou.

Asymptotic Behaviour of Boolean Functions Nonlinearity in Parallel with First-order Reed-Muller Code's Decoding Capability

Stéphanie DIB*

Institut de Mathématiques de Luminy, Marseille, France

Abstract. The aim of this work is to present an accurate distribution of Boolean functions according to the second-order nonlinearity. A lower bound being already proven for almost all Boolean functions, we specify their distribution by proving a concentration point for the nonlinearity of almost all Boolean functions. When considering the case of the (first) nonlinearity, our method meets the already proven result indeed, that is a concentration point as well. But more importantly, throughout this track, we were led to evaluate the error-correcting capability of the first-order Reed-Muller code below and beyond this concentration point. In turn, a decoding threshold is derived above which the probability of error-correcting vanishes in maximum-likelihood decoding.

Index terms - Boolean functions, nonlinearity, Reed-Muller codes, error-correction capability, decoding threshold.

1 Introduction

The nonlinearity $NL(f)$ of a Boolean function f of n variables is its Hamming distance to the set of affine functions, that is

$$NL(f) = \min_{h \text{ affine}} d_H(f, h).$$

It is used to measure the strength of cryptosystems when facing linear attacks. When dealing with low-degree approximation attacks [6, 11], we consider the higher-order nonlinearities. For every $0 \leq r \leq n$, we denote by $NL_r(f)$ the Hamming distance of f to the set of all functions whose algebraic degrees do not exceed r, that is the Reed-Muller code of order r. We have

$$NL_r(f) = \min_{g \in \mathrm{RM}(r,n)} d_H(f, g).$$

The nonlinearity of a Boolean function f of n variables is bounded from above by $2^{n-1} - 2^{n/2-1}$. When n tends to infinity, it was shown by C. Carlet [2], D. Olejàr and M. Stanek [7], that almost all Boolean functions have nonlinearities greater than $2^{n-1} - c \cdot 2^{\frac{n}{2}-1}\sqrt{2n \ln 2}$, where c is a real number greater than 1. Later on, the distribution of Boolean functions between the lower and upper bound was

* This work has been done with the support of the Région Provence-Côte d'Azur

established by F. Rodier [8–10] who proved that the nonlinearity of almost all Boolean functions lies in a small neighbourhood of $2^{n-1} - 2^{\frac{n}{2}-1}\sqrt{2n\log 2}$. This result has been proven as well later by S. Litsyn and A. Shpunt with a different approach [12].
As for the second-order nonlinearity, we recall that it is bounded from above by $2^{n-1} - \sqrt{15} \times 2^{n/2-1} + O(1)$ as proven by C. Carlet and S. Mesnager [4]. Asymptotically, C. Carlet [3] showed that the density of the set of functions satisfying

$$NL_2(f) > 2^{n-1} - c\ \sqrt{\binom{n}{2}\ln 2}\ 2^{\frac{n-1}{2}},$$

tends to 1 when n tends to infinity, if c is a real number greater than 1.

2 First result

Our result [5] shows that $2^{n-1} - \sqrt{\binom{n}{2}\ln 2}\ 2^{\frac{n-1}{2}}$ represents a concentration point of the second-order nonlinearity of Boolean functions.
The approach is basically the second moment method and applies mainly fundamental combinatorics and probability theory, whereas Rodier's approach involves harmonic analysis.
We address the problem as follows. We investigate the subset of Boolean functions whose second-nonlinearities are greater than $\delta = 2^{n-1} - c\ \sqrt{\binom{n}{2}\ln 2}\ 2^{\frac{n-1}{2}}$ where c is a strictly positive real number less than 1, δ being slightly greater than the lower bound given for almost all Boolean functions. By definition, these functions are those for which the Hamming distance to $\mathrm{RM}(2,n)$ exceeds δ. A geometrical viewpoint is the way to track these functions. Therefore, we consider the Hamming closed balls of radius δ centered at all codewords of $\mathrm{RM}(2,n)$, defined by $B_\delta[g] = \{f \in \mathcal{B}_n \mid d_H(f,g) \leq \delta\}$, where g ranges over $\mathrm{RM}(2,n)$. Notice that such functions are strictly those for which the sum of all indicator functions of the above-mentioned Hamming balls equals 0. Hence, over the set $\mathcal{B}_n$ of equiprobable Boolean functions, the problem of determining the probability of the event $\{NL_2(f) > \delta\}$ is equivalent to that of the event $\{\eta = 0\}$, where $\eta = \sum_{g\in\mathrm{RM}(2,n)} 1_{B_\delta[g]}$ is an integer-valued function defined on $\mathcal{B}_n$. This is followed by application of the Chebyshev's inequality [1, chapter IV]

$$P(\eta = 0) \leq P\big(|\eta - E(\eta)| \geq E(\eta)\big) \leq \frac{E(\eta^2) - E(\eta)^2}{E(\eta)^2}.$$

This upper bound allows us to prove a concentration point around
$2^{n-1} - \sqrt{\binom{n}{2}\ln 2}\ 2^{\frac{n-1}{2}}$, when n tends to infinity.
Through this path, we deal with the number of Boolean functions inside a Hamming ball and the intersection between two of them as well. In the (first) nonlinearity case, these estimates yield the following result.

3 Second result

Our second result shows that the concentration point related to the nonlinearity case, that is $2^{n-1} - 2^{\frac{n}{2}-1}\sqrt{2n \log 2}$, is moreover a decoding threshold of the first-order Reed-Muller code when considering maximum-likelihood decoding. Indeed, it is possible to decode correctly almost always errors of weight up to $2^{n-1} - 2^{\frac{n}{2}-1}\sqrt{2n \log 2}$ but the decoding fails almost always when error weight exceeds this value. This bound is unrelated to any algorithm but proves that one can't design algorithm to enhance the error-correction capability of first-order Reed-Muller code further from this threshold.

References

1. N. Alon, Noga; J. Spencer, The probabilistic method. Third edition. With an appendix on the life and work of Paul Erdös. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, Inc., Hoboken, NJ, 2008.
2. C. Carlet, On cryptographic complexity of Boolean functions, Proceedings of the Sixth Conference on Finite Fields with Applications to Coding Theory, Cryptography and Related Areas. Springer, G.L. Mullen, H. Stichtenoth and H. Tapia-Recillas Eds, pp. 53-69, 2002.
3. C. Carlet, The complexity of Boolean functions from cryptographic viewpoint, preprint.
4. C. Carlet, S. Mesnager, Improving the upper bounds on the covering radii of binary Reed-Muller codes. IEEE Trans. Inform. Theory 53 (2007), no. 1, 162–173.
5. S. Dib, Distribution of Boolean functions according to the second-order nonlinearity, to be published with the proccedings of the International Workshop on the Arithmetic of Finite Fields in the Springer lecture Notes in Computer Science.
6. L. Knudsen, M. Robshaw, Nonlinear approximations in linear cryptanalysis, EUROCRYPT 96, Lecture Notes in Computer Science, 1070, Springer Berlin / Heidelberg, 224-236.
7. D. Olejár et M. Stanek, On cryptographic properties of random Boolean functions, J.UCS 4 (1998), no. 8, 705–717.
8. F. Rodier, Sur la non-linéarité des fonctions booléennes, Acta Arithmetica, vol 115, (2004), 1-22, prétirage de l'IML n° 2002-07, disponible sur ArXiv: math.NT/0306395.
9. F. Rodier, On the nonlinearity of Boolean functions, Proceedings of WCC2003, Workshop on coding and cryptography 2003, sous la direction de D. Augot, P. Charpin, G. Kabatianski, INRIA, 2003, pp. 397 – 405.
10. F. Rodier, Asymptotic nonlinearity of Boolean functions, Designs, Codes and Cryptography, 40:1 2006.
11. T. Shimoyama1, T. Kaneko, Quadratic Relation of S-box and Its Application to the Linear Attack of Full Round DES, CRYPTO '98, Lecture Notes in Computer Science, 1462/, Springer Berlin / Heidelberg (1998), 129-147.
12. S. Litsyn, A. Shpunt, On the distribution of Boolean function nonlinearity. SIAM J. Discrete Math. 23 (2008/2009), no. 1, 79–95.

ON THE HIGHER TERM OF THE DIFFERENTIAL SPECTRUM OF A FUNCTION

YVES AUBRY

Abstract. We deal with the number of times that the differential uniformity of a function is reached with a connection to a conjecture of Voloch.

1. Introduction

For a function $f : \mathbb{F}_{2^n} \longrightarrow \mathbb{F}_{2^n}$, and for $\alpha \in \mathbb{F}_{2^n}$, we define the derivative of f with respect to α as the function $D_\alpha f : \mathbb{F}_{2^n} \longrightarrow \mathbb{F}_{2^n}$ defined by

$$D_\alpha f(x) = f(x+\alpha) + f(x), \ \forall x \in \mathbb{F}_{2^n}.$$

We define the differential uniformity $\delta(f)$ of the function f by:

$$\delta(f) = \max_{\alpha \neq 0, \beta} \sharp\{x \in \mathbb{F}_{2^n} \mid D_\alpha f(x) = \beta\}.$$

2. Voloch conjecture

Voloch has conjectured in [3] the following:

Conjecture 1. *For a given integer $m > 4$, there exists $\varepsilon_m > 0$ such that for all sufficiently large n, if f is a polynomial of degree m over $\mathbb{F}_{2^n}$ for at least $\varepsilon_m 2^{2n}$ values of $\alpha \neq 0$, $\beta \in \mathbb{F}_{2^n}$,*

$$\sharp\{x \in \mathbb{F}_{2^n} \mid f(x+\alpha) + f(x) = \beta\} = \delta(f).$$

A corollary of this conjecture would be that picking $\alpha \neq 0$, $\beta \in \mathbb{F}_{2^n}$ at random gives a probabilistic polynomial time algorithm for computing $\delta(f)$.

3. The quantity $\Delta(f)$

For a function $f : \mathbb{F}_{2^n} \longrightarrow \mathbb{F}_{2^n}$ and for $\alpha, \beta \in \mathbb{F}_{2^n}$, let us define $\delta_f(\alpha, \beta)$ or for convenience $\delta(\alpha, \beta)$ by:

$$\delta(\alpha, \beta) = \sharp\{x \in \mathbb{F}_{2^n} \mid f(x+\alpha) + f(x) = \beta\}.$$

Hence, the differential uniformity of f is just: $\delta(f) = \max_{\alpha \neq 0, \beta \in \mathbb{F}_{2^n}} \delta(\alpha, \beta)$.

Now we denote

$$\Delta(f) = \sharp\{(\alpha, \beta) \in \mathbb{F}_{2^n}^* \times \mathbb{F}_{2^n} \mid \delta(\alpha, \beta) = \delta(f)\}.$$

Date: May 12, 2010.

2000 *Mathematics Subject Classification.* 11R29,11R58,11R11,14H05.

Key words and phrases. Differential uniformity, power functions, APN functions.

The quantity $\Delta(f)$ corresponds to the number of 2-uplets which maximise the function $\delta(.,.)$.

The conjecture of Voloch can be rephrase as: For any $m \geq 5$, there exists $\varepsilon_m > 0$ such that for all sufficiently large n, if f is a polynomial of degree m over $\mathbb{F}_{2^n}$,

$$\Delta(f) \geq \varepsilon_m 2^{2n}.$$

We propose here to study $\Delta(f)$.

4. Differential spectrum

In analogy with the differential spectrum defined in [1] for power functions, we define the complete differential spectrum of $f : \mathbb{F}_{2^n} \longrightarrow \mathbb{F}_{2^n}$ by:

$$\mathrm{Spec}(f) = \{\Omega_0, \Omega_2, \ldots, \Omega_{\delta(f)}\}$$

where

$$\Omega_i = \sharp\{(\alpha, \beta) \in \mathbb{F}_{2^n}^* \times \mathbb{F}_{2^n} \mid \delta(\alpha, \beta) = i\}.$$

The quantity $\Delta(f)$ is nothing but $\Omega_{\delta(f)}$.

If f is a power function, namely $f(x) = x^d$, then for any nonzero $\alpha \in \mathbb{F}_{2^n}$ the equation $(x+\alpha)^d + x^d = \beta$ can be written

$$\alpha^d\Big((\frac{x}{\alpha} + 1)^d + (\frac{x}{\alpha})^d\Big) = \beta.$$

Thus $\delta(\alpha, \beta) = \delta(1, \beta/\alpha^d)$ for all $\alpha \neq 0$.

Hence, for all $\alpha \neq 0$, we have:

$$\sharp\{\beta \in \mathbb{F}_{2^n} \mid \delta(\alpha, \beta) = i\} = \sharp\{\beta \in \mathbb{F}_{2^n} \mid \delta(1, \beta) = i\},$$

i.e. it suffices to consider the derivative $D_1 f$ to get the complete differential spectrum of a power function f. In this sipirit, Blondeau, Canteaut and Charpin in [1] have defined the differential spectrum of a power function f as the set of ω_i, $i = 0, 2, \ldots, \delta(f)$ where

$$\omega_i = \sharp\{\beta \in \mathbb{F}_{2^n} \mid \delta(1, \beta) = i\}.$$

We have of course: $\Omega_i = 2^{n-1}\omega_i$.

5. First bounds

We have:

Proposition 1. *If $f : \mathbb{F}_{2^n} \longrightarrow \mathbb{F}_{2^n}$ then:*

$$\Delta(f) \leq (2^n - 1) \times \frac{2^n}{\delta(f)} \leq (2^n - 1) \times 2^{n-1}.$$

Moreover, if f is is a power function then:

$$\Delta(f) \geq 2^n - 1.$$

Remark 2. *Recall that a function f is called APN if $\delta(f) = 2$, i.e. the function $(\alpha, \beta) \longmapsto \delta(\alpha, \beta)$ has for image $\{0, 2\}$. Another way to say that is to say that all the derivatives $D_\alpha f$ of f are 2-to-1 (for any $a \in \mathbb{F}_{2^n}$), hence half of the elements of $\mathbb{F}_{2^n}$ have two preimages by $D_\alpha f$. Thus if f is APN, then $\Delta(f) = (2^n - 1)2^{n-1}$ and the upper bound of the previous proposition is reached.*

Remark 3. *Nyberg has shown in* [2] *that $f(x) = x^{2^n-2}$ over $\mathbb{F}_{2^n}$ has $\delta(f) = 4$ when n is even. Moreover, we can show that $\Delta(f) = 2^n - 1$. Indeed, the derivative $D_1 f$ is given by: $D_1 f(x) = \frac{1}{x(x+1)}$ if $x \neq 0$ and $x \neq 1$, and $D_1 f(0) = D_1 f(1) = 1$. If $\beta = 1$, then β has four preimages: 0, 1 and the two roots of $x^2 + x + 1 = 0$. If $\beta \neq 0$ and $\neq 1$, then a preimage x of β, if it exits, would satisfies the equation $x^2 + x + b = 0$ where $b = \beta^{-1}$. But, by Hilbert's Theorem 90, this equation has two solutions if $Tr(b) = 0$ and zero solution else. Here, Tr denotes the trace function of $\mathbb{F}_{2^n}$ over $\mathbb{F}_2$ which is a linear form, thus its kernel is an hyperplane and hence has 2^{n-1} elements. Thus $\Omega_0 = (2^n - 1)(2^{n-1} + 1)$, $\Omega_2 = (2^n - 1)(2^{n-1} - 2)$ and $\Delta(f) = \Omega_4 = 2^n - 1$. Thus, the lower bound of the previous proposition is reached.*

Example 1 : If $f(x) = x^m$ is a power permutation of $\mathbb{F}_{2^n}$ (i.e. $(2^n - 1, m) = 1$) and if $\delta(f) = 4$ then it is proved in [1] (corollary 4.2) that

$$\Delta(f)/(2^n - 1) = \kappa - 2^{n-2}$$

where $2^{n-2} < \kappa \leq 2^{n-1}$. Hence, we deduce that $\Delta(f) \leq (2^n - 1)2^{n-2}$. Note that for a Kasami exponent (i.e. $m = 2^{2k} - 2^k + 1$ with $(n, k) = 1$), we have $\Delta(f) = (2^n - 1)2^{n-2}$.

Example 2 : Let us consider functions f with a two-valued differential spectra i.e. such that $\delta(\alpha, \beta)$ takes two values, namely 0 and κ. As shown in [1], κ must be a power of 2 since in this case the function $x \longmapsto D_\alpha f(x)$ is $\delta(f)$-to-1 for any fixed α. Hence, we find:

$$\Delta(f) = (2^n - 1) \times \frac{2^n}{\delta(f)}.$$

We will study here the quantity $\Delta(f)$ and the related Voloch conjecture in different cases.

References

[1] C. Blondeau, A. Canteaut and P. Charpin, *Differential properties of power functions*, Int. J. Information and Coding Theory, Vol. 1, No. 2, 149-170, (2010).

[2] K. Nyberg, *Differentially uniform mappings for cryptography*, In Advances in Cryptology, Eurocrypt'93, LNCS vol. 765, p. 55-64, Springer-Verlag, 1993.

[3] J. F. Voloch, *Symmetric cryptography and algebraic curves, Algebraic Geometry and its Applications*, Series on Number Theory and Its Aplications, Vol. 5, World scientific, 135-141, (2008).

Institut de Mathématiques de Toulon, Université du Sud Toulon-Var, France

E-mail address: yves.aubry@univ-tln.fr

On the projection construction of planar and APN mappings

Jürgen Bierbrauer and Gohar M. Kyureghyan

May 12, 2010

Abstract

In the talk we present a method for constructing planar and APN mappings by a suitable choice of their coordinate functions. We expect to have more results on this subject by the dates of YACC.

Keywords: planar mapping, APN mapping, quadratic mapping

Let p be a prime number. Given a mapping $f : \mathbb{F}_{p^n} \to \mathbb{F}_{p^n}$ and a nonzero element $a \in \mathbb{F}_{p^n}$, we call

$$D_{f,a} : \mathbb{F}_{p^n} \to \mathbb{F}_{p^n}, \quad x \mapsto f(x+a) - f(x) - f(a)$$

the difference mapping of f defined by a. A mapping $f : \mathbb{F}_{p^n} \to \mathbb{F}_{p^n}$ is called planar (or perfect nonlinear) if all its difference mappings are bijective. Planar mappings do not exist in fields of even characteristic. Indeed, let $f : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$. Then the cardinality of the image set of a difference mapping $D_{f,a}$ is at most 2^{n-1}, since $D_{f,a}(x+a) = D_{f,a}(x)$ for any $x \in \mathbb{F}_{2^n}$. A mapping $f : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ is called almost perfect nonlinear (APN) if any its difference mapping has an image set of the maximal possible cardinality 2^{n-1}.

In [1] the first author introduces a method for constructing planar mappings, which meanwhile yields a unified explanation for several recent constructions of such mappings. In this note we continue study of this method.

Observe that the definition of planar and APN mappings uses only the additive group of the field, and hence such mappings can be defined in any group. The multiplicative structure of the field is beneficial in the study of planar and APN mappings, since it allows to construct explicit families of such mappings in polynomial forms. The monomial planar and APN mappings illustrate this. In this paper we show, that there are families of planar or APN mappings, which can be easier explained and better understood by studying their coordinate functions with respect to an appropriate chosen basis of $\mathbb{F}_{p^n}$ over a subfield.

In the rest of the paper we use the following notation: $n = m \cdot k$ and $\Omega := \{1, \omega_1, \ldots, \omega_{k-1}\} \subset \mathbb{F}_{p^n}$ is a basis of $\mathbb{F}_{p^n}$ over $\mathbb{F}_{p^m}$. Then, clearly, every mapping $F : \mathbb{F}_{p^n} \to \mathbb{F}_{p^n}$ can be uniquely represented by

$$F(x) = f_0(x) + \omega_1 f_1(x) + \ldots + \omega_{k-1} f_{k-1}(x),$$

where $f_i : \mathbb{F}_{p^n} \to \mathbb{F}_{p^m}$ for $0 \le i \le k-1$. We write $F = (f_0, \ldots, f_{k-1})$ and call the functions f_i the coordinate functions of F with respect to the basis Ω.

The following characterization of planar and APN mappings is straightforward:

Proposition 1 *The mapping $F : \mathbb{F}_{p^n} \to \mathbb{F}_{p^n}$ is planar (resp. APN) if and only if for every $a \in \mathbb{F}_{p^n}^*$ and $(b_0, \ldots, b_{k-1}) \in \mathbb{F}_{p^m}^k$ the following system of equations*

$$\begin{cases} f_0(x+a) - f_0(x) &= b_0 \\ & \vdots \\ f_{k-1}(x+a) - f_{k-1}(x) &= b_{k-1} \end{cases}$$

has exactly 1 (resp. 2) solutions in $\mathbb{F}_{p^n}$.

Remark 1 *Every mapping $g : \mathbb{F}_{p^n} \to \mathbb{F}_{p^m}$ can be represented as $g(x) = tr_{n/m}(G(x))$ where $tr_{n/m}$ is the trace mapping from $\mathbb{F}_{p^n}$ onto $\mathbb{F}_{p^m}$ and $G : \mathbb{F}_{p^n} \to \mathbb{F}_{p^n}$.*

A mapping $F : \mathbb{F}_{p^n} \to \mathbb{F}_{p^n}$ is called quadratic if it is described by a polynomial of form $\sum_{i,j} \alpha_{i,j} x^{p^i+p^j}$. For quadratic mappings Proposition 1 reduces to:

Corollary 1 *Let $n = k \cdot m$ and a mapping $F : \mathbb{F}_{p^n} \to \mathbb{F}_{p^n}$ be quadratic, i.e. $F(x) = \sum_{i,j} \alpha_{i,j} x^{p^i+p^j}$. Further, let $F = (f_0, \ldots, f_{k-1})$ with respect to some fixed basis of $\mathbb{F}_{p^n}$ over $\mathbb{F}_{p^m}$. Define*

$$l_i(x,y) = f_i(x+y) - f_i(x) - f_i(y).$$

Then F is a planar (resp. APN) if and only if for every fixed $a \in \mathbb{F}_{p^n}^*$ the subspace

$$\bigcap_{i=0}^{k-1} Kernel\big(l_i(x,a)\big)$$

is trivial (resp. 1-dimensional).

Proof. It follows from Proposition 1, since for a quadratic F the mappings $l_i(x,y)$ are bilinear. ◇

Next we show that Corollary 1 allows to classify all planar mappings of the shape $x^{p^m+1} + \omega\, tr_{n/m}(\alpha x^{p^i+p^j})$ when $n = 2m$.

Theorem 1 *Let p be an odd prime number, $n = 2m$ and $\omega \in \mathbb{F}_{p^n} \setminus \mathbb{F}_{p^m}$. Further, let $m, r \ge 1$, $p^m - 1 = 2^s k$ and $p^r - 1 = 2^t \ell$ with k, ℓ odd. Then the mapping $F : \mathbb{F}_{p^n} \to \mathbb{F}_{p^n}$ given by*

$$F(x) = x^{p^m+1} + \omega\, tr_{n/m}(\alpha x^{p^i(p^r+1)})$$

is planar if and only if

- α *is not a* (p^r+1)*th power in* $\mathbb{F}_{p^n}$*; and*

- $s \leq t$.

Proof. Note that $\{1, \omega\}$ is a basis of $\mathbb{F}_{p^n}$ over $\mathbb{F}_{p^m}$. Further, $F = (f_0, f_1)$ with $f_0(x) = x^{p^m+1}$ and $f_1(x) = tr_{n/m}(\alpha x^{p^i(p^r+1)})$. Since F is quadratic we apply Corollary 1. Here we have

$$l_0(x,y) = x^{p^m}y + xy^{p^m}$$

and

$$l_1(x,y) = tr_{n/m}\left(\alpha(xy^{p^r} + x^{p^r}y)^{p^i}\right).$$

Let $a \in \mathbb{F}_{p^n}^*$. Then

$$l_0(x,a) = a^{p^m+1}\left(\left(\frac{x}{a}\right)^{p^m} + \frac{x}{a}\right)$$

and thus $l_0(x,a) = 0$ if and only if $x/a \in \delta\,\mathbb{F}_{p^m}$, where u is a primitive element of $\mathbb{F}_{p^n}$ and $\delta = u^{(p^m+1)/2}$. Note that

$$\begin{aligned} l_1(x,a) &= tr_{n/m}\left(\alpha a^{p^i(p^r+1)}\left(\frac{x}{a} + \left(\frac{x}{a}\right)^{p^r}\right)^{p^i}\right) \\ &= tr_{n/m}\left(\beta\left(\frac{x}{a} + \left(\frac{x}{a}\right)^{p^r}\right)\right)^{p^i}, \end{aligned}$$

where we set $\beta = \left(\alpha a^{p^i(p^r+1)}\right)^{p^{n-i}}$.

Let $y = x/a \in \delta\,\mathbb{F}_{p^m}$. So $y = \delta z$ with $z \in \mathbb{F}_{p^m}$. Then $l_1(y,a) = 0$ if and only if

$$(\beta\delta + (\beta\delta)^{p^m})z + (\beta\delta^{p^r} + (\beta\delta^{p^r})^{p^m})z^{p^r} = 0.$$

By the choice of δ we have $\delta^{p^m} = -\delta$. So the last equality can be written as

$$(\beta - \beta^{p^m})\delta z + (\beta - \beta^{p^m})(\delta z)^{p^r} = (\beta - \beta^{p^m})(\delta z + (\delta z)^{p^r}) = 0. \tag{1}$$

Hence (1) holds if and only if one of the following conditions is satisfied

$$\beta^{p^m-1} = 1 \tag{2}$$

or

$$(\delta z)^{p^r} + \delta z = 0. \tag{3}$$

From the definition of β it follows that (2) is equivalent to

$$\alpha^{p^m-1}a^{p^i(p^r+1)(p^m-1)} = 1,$$

the latter equality holds for some $a \in \mathbb{F}_{p^n}$ if and only if α is a (p^r+1)th power in $\mathbb{F}_{p^n}$. Finally, using the definition of δ, condition (3) can be reduced to

$$u^{(p^m+1)\frac{p^r-1}{2}}z^{p^r-1} = -1 = u^{(p^m+1)\frac{p^m-1}{2}},$$

or equivalently

$$z^{p^r-1} = u^{(p^m+1)p^r \frac{p^{m-r}-1}{2}}.$$

The latter equation has a solution in $\mathbb{F}_{p^m}$ if and only if $\gcd(p^r-1, p^m-1) = 2^{\min(s,t)}\gcd(k,\ell)$ divides $\gcd((p^{m-r}-1)/2, p^m-1) = \gcd((p^r-1)/2, p^m-1) = 2^{\min(s-1,t)}\gcd(k,\ell)$.

$\diamond$

Remark 2 *Corollary 1 implies the classification of all APN mappings of the shape $x^{p^m+1} + \omega\, tr_{n/m}(\alpha x^{p^i+p^j})$ when $n = 2m$ as well.*

The following result follows from Theorem 1 and extends the construction of planar mappings given in [3, Main Theorem] based on the APN trinomials from [2, Corollary 1].

Corollary 2 *Let $n = 2m$, $m, r \geq 1$, $p^m - 1 = 2^s u$ and $p^r - 1 = 2^t v$ with u, v odd. Further, let $c \in \mathbb{F}_{p^n}$ be of multiplicative order k with $\gcd(k, p^m-1) = 1$. Define ℓ to satisfy $\ell(p^m-1) \equiv 1 \pmod{k}$. Then the mapping*

$$F(x) = bx^{p^m+1} + x^{p^i(p^r+1)} + cx^{p^{m+i}(p^r+1)}$$

is planar if and only if all of the following conditions are fulfilled:

- *$c^\ell b$ does not belong to $\mathbb{F}_{p^m}$;*
- *c^ℓ is not a (p^r+1)th power in $\mathbb{F}_{p^n}$; and*
- *$s \leq t$.*

Proof. Note that

$$\begin{aligned} c^\ell F(x) &= c^\ell b x^{p^m+1} + c^\ell x^{p^i(p^r+1)} + c^{\ell+1} x^{p^{m+i}(p^r+1)} \\ &= c^\ell b x^{p^m+1} + tr_{n/m}(c^\ell x^{p^i(p^r+1)}), \end{aligned}$$

since by choice of ℓ it holds $c^{\ell+1} = c^{\ell p^m}$. Hence Theorem 1 completes the proof. $\diamond$

More results on planar and APN mappings obtained via Corollary 1 will be presented in the extended version of this note.

References

[1] J. Bierbrauer: *New commutative semifields from projection mappings,* submitted to *Designs, Codes and Cryptography.*

[2] L. Budaghyan and C. Carlet, *Classes of quadratic APN trinomials and hexanomials and related structures*, *IEEE Trans. Inf. Theory*, 54(5) (2008) pp. 2354–2357.

[3] Z. Zha and X. Wang: *New families of perfect nonlinear polynomial functions, Journal of Algebra* (2009), doi:10.1016/j.algebra.2009.04.042.

Some More Functions That Are Not APN Infinitely Often

François Rodier

Abstract

We give new functions which cannot be APN over an infinity of field extensions.

Keywords: vector Boolean functions, almost perfect nonlinear functions, algebraic surface.

1 Introduction

The vector Boolean functions are used in cryptography to construct block ciphers and an important criterion on these functions is highly resistance to differential cryptanalysis.

Let $q = 2^n$ for some positive integer n. A function $f : \mathbb{F}_q \longrightarrow \mathbb{F}_q$ is said to be *almost perfect nonlinear* (APN) on $\mathbb{F}_q$ if the number of solutions in $\mathbb{F}_q$ of the equation

$$f(x + a) + f(x) = b$$

is at most 2, for all $a, b \in \mathbb{F}_q$, $a \neq 0$. Because $\mathbb{F}_q$ has characteristic 2, the number of solutions to the above equation must be an even number, for any function f on $\mathbb{F}_q$. This kind of function has a good resistance to differential cryptanalysis as was proved by Nyberg in [11].

So far, the study of APN functions has focused on power functions. Recently it was generalized to other functions, particularly polynomials (Carlet, Pott and al. [3, 7, 8]) or polynomials on small fields (Dillon [6]). On the other hand, several authors (Berger, Canteaut, Charpin, Laigle-Chapuy [2], Byrne and McGuire [4] or Jedlicka [10]) showed that APN functions did not exist in certain cases.

Thus, towards a full classification of all APN functions, one approach that already proved to be successful is to show that certain polynomials are not APN for infinitely many extensions of $\mathbb{F}_2$. So here one first fixes a

finite field $\mathbb{F}_q$ and a function $f : \mathbb{F}_q \to \mathbb{F}_q$ given as a polynomial in $\mathbb{F}_q[x]$ and asks the question whether this function can be APN on infinitely many extensions of $\mathbb{F}_q$. There is a variety of classes of functions for which it can be shown that each function is APN at most for a finite number of extensions.

Hernando and McGuire [9] showed a result on the classification of APN monomials which has been conjectured for 40 years: the only exponents such that the monomial x^d are APN over infinitely many extension of $\mathbb{F}_2$ are of the form $2^i + 1$ or $4^i - 2^i + 1$. One calls these exponents *exceptional exponents*. Then it is natural to formulate for polynomial functions the following conjecture.

Conjecture: (Aubry, McGuire and Rodier) a polynomial can be APN for an infinity of ground fields $\mathbb{F}_q$ only if it is CCZ equivalent (as was defined by Carlet, Charpin and Zinoviev in [5]) to a monomial x^t where t is an exceptional exponent.

One mean to prove this conjecture is to remark that the APN property is equivalent to the fact that the rational points of a certain algebraic surface $S(f)$ in a 3-dimensional space linked with the polynomial f defining the Boolean function are all in a surface V made of 3 planes and independent of f. We define the surface $S(f)$ in the three dimensional affine space $\mathbb{A}^3$ by

$$\phi(x, y, z) = \frac{f(x) + f(y) + f(z) + f(x + y + z)}{(x + y)(x + z)(y + z)}$$

which is a polynomial in $\mathbb{F}_q[x, y, z]$.

When this surface is irreducible (or when it has an irreducible component defined over the field of definition of f), a Weil's type bound may be used to approximate the number of rational points of this surface. When it is too large the surface is too big to be contained in the surface V, and the function f cannot be APN. A relation between the degree and the number of variables on a large number of Boolean function was obtained by this mean by the author [12]. This enabled him, with Aubry and McGuire to prove the conjecture in a number of cases.

2 The State of the Art

The best known examples of APN functions are the Gold functions x^{2^k+1} and the Kasami-Welch functions $x^{4^k-2^k+1}$. These functions are defined over $\mathbb{F}_2$, and are APN on any field $\mathbb{F}_{2^m}$ where $gcd(k, m) = 1$.

Theorem 2.1 (Aubry, McGuire and Rodier, [1]) *If the degree of the*

polynomial function f is odd and not a Gold or a Kasami-Welch number then f is not APN over $\mathbb{F}_{q^n}$ for all n sufficiently large.

In the even degree case, we can state the result when half of the degree is odd, with an extra minor condition.

Theorem 2.2 (Aubry, McGuire and Rodier, [1]) *If the degree of the polynomial function f is $2e$ with e odd, and if f contains a term of odd degree, then f is not APN over $\mathbb{F}_{q^n}$ for all n sufficiently large.*

We have some results on the polynomials of Gold degree $d = 2^k + 1$.

Theorem 2.3 (Aubry, McGuire and Rodier, [1]) *Suppose $f(x) = x^d + g(x)$ where $\deg(g) \leq 2^{k-1}+1$. Let $g(x) = \sum_{j=0}^{2^{k-1}+1} a_j x^j$. Suppose moreover that there exists a nonzero coefficient a_j of g such that $\phi_j(x, y, z)$ is absolutely irreducible (where $\phi_i(x, y, z)$ denote the polynomial $\phi(x, y, z)$ associated to x^i). Then f is not APN over $\mathbb{F}_{q^n}$ for all n sufficiently large.*

3 New results

3.1 Functions of degree $4e$ where e is odd

We have been interested in the degrees of the form $4e$, where e is an odd number. This case is far more intricate than the previous cases, because there are some polynomials which are CCZ equivalent to monomials.

Here are some partial results.

Theorem 3.1 *If the degree of the polynomial function f is even such that $\deg(f) = 4e$ with $e \equiv 3 \pmod 4$ and $e \geq 7$, then f is not APN over $\mathbb{F}_{q^n}$ for n large.*

Sketch of proof. We have in this case:

$$\phi_d(x, y, z) = \phi_e(x, y, z)^4((x + y)(x + z)(y + z))^3$$

where ϕ_d is the ϕ function associated to the monomial x^d. Hence X_∞ is not reduced where $X_\infty = H_\infty \cap \overline{S}(f)$, H_∞ is the plane at infinity of the space $\mathbb{A}^3$ and $\overline{S}(f)$ is the projective closure of $S(f)$.

Let X_0 a reduced absolutely irreducible component of $\overline{S}(f)$ which contains the line $x + y = 0$ in H_∞. By studying the various cases which may happen, one is reduced to the case where there are three components X_0,

X_1 and X_2 which contain this line, they are defined over the field $\mathbb{F}_{q^3}$, where $\mathbb{F}_q$ is the field of definition of f and the Galois group of $\mathbb{F}_{q^3}$ over $\mathbb{F}_q$ acts by cyclic permutations of X_0, X_1 and X_2. By studying the irreducibility of X_0, one is again reduced to the case where X_0, X_1 and X_2 are of degree 3 of the form $((x+y)(x+z)(y+z))^3 + P$ where P is a polynomial of degree at most 2. Because of the symmetry of the function ϕ associated to f in the variables x, y and z, one is reduced to the cases where the polynomial P is of the form

$$P(x,y,z) = c_1(x^2+y^2+z^2) + c_4(xy+xz+zy) + b_1(x+y+z) + d$$

Investigating these polynomials, one prove that it is impossible that they divide ϕ.

We have some other results when the degree e is a Gold degree.

Theorem 3.2 *If the degree of the polynomial f is 12, then either f is not APN over $\mathbb{F}_{q^m}$ for m large or f is CCZ equivalent to the Gold function x^3.*

Let $d \in \mathbb{F}_{q^3}$ such that $d + d^q + d^{q^2} = 0$. The polynomials CCZ equivalent to the Gold function x^3 are the polynomial functions

$$f(x) = L(x^3) \quad \text{or} \quad f(x) = L(x)^3$$

with

$$L(x) = x^4 + (d^{1+q} + d^{1+q^2} + d^{q+q^2})x^2 + d^{1+q+q^2}x$$

and the polynomials composed with f and an affine permutation.

3.2 Functions of Kasami degree

We can prove that a function $f(x) = x^d + g(x)$ where $d = 2^{2k} - 2^k + 1$ and $\deg(g)$ small enough and such that there exists a nonzero coefficient a_j of g such that $\phi_j(x,y,z)$ is absolutely irreducible. Then $\phi(x,y,z)$ is absolutely irreducible and f is not APN over $\mathbb{F}_{q^n}$ for all n sufficiently large.

This is proved by the same way than the case of polynomials of Gold degree in [1].

References

[1] Y. Aubry, G. McGuire, F. Rodier, *A few more functions that are not APN infinitely often*, to be published with the proceedings of the conference Fq9, Dublin, 2009. Contemporary Mathematics Series.

[2] T. Berger, A. Canteaut, P. Charpin, Y. Laigle-Chapuy *On almost perfect nonlinear functions over* F_2^n. IEEE Trans. Inform. Theory 52 (2006), no. 9, 4160–4170.

[3] L. Budaghyan and C. Carlet and P. Felke and G. Leander *An infinite class of quadratic APN functions which are not equivalent to power mappings,* Cryptology ePrint Archive, n° 2005/359

[4] Byrne E. and McGuire G., *On the Non-Existence of Quadratic APN and Crooked Functions on Finite Fields*, prepublication. http://www.maths.may.ie/sta/gmg/APNniceWeilEBGMG.pdf.

[5] C. Carlet, P. Charpin and V. Zinoviev. *Codes, bent functions and permutations suitable for DES-like cryptosystems.* Designs, Codes and Cryptography, 15(2), pp. 125-156, 1998.

[6] J. Dillon, *APN Polynomials: An Update.* Fq9, International Conference on Finite Fields and their Applications July 2009

[7] Y. Edel, G. Kyureghyan and A. Pott. *A new APN function which is not equivalent to a power mapping.* IEEE Trans. Inform. Theory 52 (2006), no. 2, 744–747.

[8] Y. Edel, A. Pott. *A new almost perfect nonlinear function which is not quadratic* Adv. Math. Commun. 3 (2009), no. 1, 59–81.

[9] F. Hernando and G. McGuire, *Proof of a conjecture on the sequence of exceptional numbers, classifying cyclic codes and APN functions, arXiv:0903.2016v3* [cs.IT], 2009.

[10] Jedlicka, D., *APN monomials over* GF(2^n) *for infinitely many* n, Finite Fields Appl. 13 (2007), no. 4, 1006–1028.

[11] K. Nyberg, *Differentially uniform mappings for cryptography, Advances in cryptology—Eurocrypt '93* (Lofthus, 1993), 55–64, Lecture Notes in Comput. Sci., Vol. 765, Springer, Berlin, 1994.

[12] F. Rodier, *Bornes sur le degré des polynômes presque parfaitement non-linéaires,* in Arithmetic, Geometry, Cryptography and Coding Theory, G. Lachaud, C. Ritzenthaler and M.Tsfasman editors, Contemporary Math. no 487, AMS, Providence (RI), USA, pp. 169-181, 2009. *arXiv:math/0605232v3* [math.AG].

On a Class of APN Permutation Polynomials

Ayça Çeşmelioğlu, Wilfried Meidl and Alev Topuzoğlu

Sabancı University, MDBF, Orhanlı, 34956 Tuzla, İstanbul, Turkey

Abstract

The permutations $\mathcal{P}(x) = (a_0x + a_1)^{p-2} + a_2$ over a prime field $\mathbb{F}_p$ are APN if and only if $p \equiv 5 \bmod 6$. We present results on the dispersion and p-dispersion of $\mathcal{P}(x)$, and relate this concept to Costas property. Results on the construction and enumeration of permutations $\mathcal{P}(x)$ with prescribed dispersion and cycle structure are given. Applications in coding theory are described.

Key Words: Permutation polynomial, APN permutation, inversive generator, dispersion, Costas permutation

Almost perfect nonlinear (APN) functions are mainly studied over fields of characteristic two, since there are no *perfect nonlinear* (PN) functions in characteristic 2. On the other hand PN "permutations" do not exist in odd characteristic either, hence APN permutations of $\mathbb{F}_p$ are of interest for odd p also.
This talk is concerned with the class of permutations $\mathcal{P}(x) \in \mathbb{F}_p[x]$, where

$$\mathcal{P}(x) = (a_0x + a_1)^{p-2} + a_2, \tag{1}$$

$a_0 \neq 0$ and p is an odd prime. We first observe that $\mathcal{P}(x)$ is APN if and only if $p \equiv 5 \bmod 6$.
These polynomials give rise to the well-known inversive pseudorandom number generators (u_n), which are defined recursively by $u_n = \mathcal{P}(u_{n-1}), u_0 \in \mathbb{F}_p$, for $a_1 = 0$. Since their *randomness* is well established, they are good candidates for use in turbo codes as interleavers. Indeed our results on the values of dispersion of these polynomials show that one can select the parameters a_0, a_1 and a_2 to yield permutations with prescribed dispersion (e.g. close to that of a random permutation). We recall that dispersion of a permutation P of the set $\{0, 1, \ldots, n-1\}$ is the cardinality $|D(P)|$ of the set

$$D(P) = \{(j - i, P(j) - P(i)) | 0 \leq i < j \leq n - 1\}.$$

Permutations with maximum possible dispersion $n(n-1)/2$ are, of course, those which satisfy *Costas property* (see [4, 6] for a recent survey on Costas

arrays). Accordingly the normalized dispersion is defined as

$$\gamma(P) = \frac{2\,|D(P)|}{n(n-1)},$$

and hence $\gamma(P) \leq 1$ for any permutation P of the set $\{0, 1, \ldots, n-1\}$. We note that the expected value of normalized dispersion of a random permutation is approximately 0.8, see, [3], [7]. Hence a permutation P can be considered as being random (and hence as a favourable interleaver) if $\gamma(P)$ is close to 0.8. We note that there are only a few non-empirical results on the normalized dispersion of permutation polynomials over finite fields, and only about monomials.
Since we need to calculate both in $\mathbb{Z}$ and in $\mathbb{F}_p$, we denote addition and subtraction in $\mathbb{F}_p$ by "$\oplus$", and "$\ominus$", to avoid confusion. Studying permutations of $\mathbb{F}_p$, we naturally consider the *p-dispersion* $|D(P, \mathbb{F}_p)|$ of $P \in \mathbb{F}_p[x]$, which we define as the cardinality of the set

$$D(P, \mathbb{F}_p) = \{(k, P(i \oplus k) \ominus P(i)) | 1 \leq k \leq p-1, 0 \leq i \leq p-1-k\}.$$

Our main results concerning dispersion of permutations $\mathcal{P}(x)$ are stated below. The element (the *pole*) $\ominus\frac{a_1}{a_0} \in \mathbb{F}_p$, is denoted by x_p. We denote the integer in $\{0, 1, \ldots, p-1\}$, which x_p is identified with, by x_I.

Theorem 1. *Let $p \equiv 5 \bmod 6$ and $\mathcal{P}(x) = (a_0 x \oplus a_1)^{p-2} \oplus a_2 \in \mathbb{F}_p[x]$, $a_0 \neq 0$. Then*

$$|D(\mathcal{P}, \mathbb{F}_p)| = \begin{cases} \frac{(p+3)(p-1)+4x_I(p-2x_I-2)}{4} & : \quad 0 \leq x_I \leq \frac{p-1}{2}, \\ \frac{(p+3)(p-1)+4(p-1-x_I)(2x_I-p)}{4} & : \quad \frac{p-1}{2} < x_I \leq p-1. \end{cases}$$

Theorem 2. *Let $p \equiv 5 \mod 6$ and let $\underline{\mathcal{P}}(x) = (a_0 x \oplus a_1)^{p-2} \in \mathbb{F}_p[x]$, $a_0 \neq 0$. The dispersion $|D(\underline{\mathcal{P}})|$ of $\underline{\mathcal{P}}(x)$ is then given by*

$$|D(\underline{\mathcal{P}})| = \begin{cases} \frac{(p+3)(p-1)+4x_1(p-2x_1-1)}{4} & : \quad 0 \leq x_1 \leq \frac{p-1}{2}, \\ \frac{(p+3)(p-1)+4(p-1-x_1)(2x_1-p+1)}{4} & : \quad \frac{p-1}{2} < x_1 \leq p-1. \end{cases}$$

Corollary 3. *Let $p \equiv 5 \bmod 6$ and let $\underline{\mathcal{P}}(x) = (a_0 x \oplus a_1)^{p-2} \in \mathbb{F}_p[x]$, $a_0 \neq 0$ The maximum value for the normalized dispersion $\gamma(\underline{\mathcal{P}})$ is $\gamma(\underline{\mathcal{P}}) = 0.75 + \frac{5}{4p}$. This value is attained for $x_I = \lfloor \frac{p-1}{4} \rfloor$ and $x_I = \lfloor \frac{3p-1}{4} \rfloor$.*

Corollary 4. *Let $p \equiv 5 \bmod 6$ and let $\underline{\mathcal{P}}(x) = (a_0 x \oplus a_1)^{p-2} \in \mathbb{F}_p[x]$, $a_0 \neq 0$. Then $|D(\underline{\mathcal{P}}, \mathbb{F}_p)| = |D(\underline{\mathcal{P}})|$ if and only if $\underline{\mathcal{P}}(x) = a_0 x^{p-2}$ or $\underline{\mathcal{P}}(x) = (a_0 x + a_0)^{p-2}$ where $a_0 \in \mathbb{F}_p^*$ is arbitrary.*

It is suggested that the cycle structure of a permutation P also affects the performance of P when used as an interleaver. Of particular interest are the permutations that decompose into cycles of the same length (see [8, 9]). We therefore present results on the cycle decomposition of $\mathcal{P}(x) = (a_0x \oplus a_1)^{p-2} \oplus a_2$ in relation to its dispersion. Since p-dispersion of $\mathcal{P}$ depends only on $\ominus a_1/a_0$, we show how to choose the remaining parameters in order to obtain a prescribed cycle structure, assuming $p \equiv 5 \bmod 6$. Furthermore we determine the number of permutations $\mathcal{P}(x) = (a_0x \oplus a_1)^{q-2} \oplus a_2 \in \mathbb{F}_p[x]$ with prescribed p-dispersion and cycle structure. The polynomial,

$$f(x) = x^2 \ominus (a_0a_2 \oplus a_1)x \ominus a_0. \tag{2}$$

associated to $\mathcal{P}(x)$, plays the central role in studying the cycle structure of $\mathcal{P}(x)$. Let $\alpha, \beta \in \mathbb{F}_{p^2}$ be the roots of $f(x)$, then the multiplicative order m of α/β either divides $p-1$ or $p+1$. The cycle structure of $\mathcal{P}(x)$ depends on this order m of α/β as can be seen in the following proposition. When a permutation P is decomposed into a product of disjoint cycles, we write $\mathcal{C}_P = [n_1 \times l_1, n_2 \times l_2, \ldots, n_s \times l_s]$ to indicate that the permutation is composed of n_1 cycles of length l_1, n_2 cycles of length l_2,...,n_s cycles of length l_s, $l_1 > l_2 > \cdots > l_s \geq 1$.

Proposition 5. *[1, Theorem 2] For a permutation $\mathcal{P}(x) = (a_0x \oplus a_1)^{p-2} \oplus a_2$, let f be the associated polynomial as in (2), and α, β be the roots of f. Suppose that m is the order of α/β in $\mathbb{F}_{p^2}$. Then $\mathcal{C}_\mathcal{P} = \mathcal{C}_\mathcal{P}(m)$ depends on m as follows:*

$$\mathcal{C}_\mathcal{P}(m) = \begin{cases} [(t-1) \times m, 1 \times (m-1)] & : \ m = (p+1)/t, \\ & \quad 1 \leq t < (p+1)/2, \\ [(t-1) \times m, 1 \times (m-1), 2 \times 1] & : \ m = (p-1)/t, \\ & \quad 1 \leq t < (p-1)/2, \\ [1 \times (p-1), 1 \times 1] & : \ m = 1, \\ [((p-1)/2 - \delta(f)) \times 2, (1 + 2\delta(f)) \times 1] & : \ m = 2, \end{cases}$$

with $\delta(f) = 0$ if f is irreducible and $\delta(f) = 1$ if f is reducible.

It follows that for $m = p \pm 1$ we have only one nontrivial cycle (of length > 1) having length $m-1$. If $2 < m < p-1$, all nontrivial cycles have the same length m, except for one of length $m-1$. If $m = 2$, all nontrivial cycles have length 2.

Theorem 6. *Let $p \equiv 5 \bmod 6$. For any $x_p \in \mathbb{F}_p$ and any integer m dividing $p-1$ or $p+1$, there is a permutation $\mathcal{P}(x) = (a_0x \oplus a_1)^{q-2} \oplus a_2 \in \mathbb{F}_p[x]$ with cycle decomposition $\mathcal{C}_\mathcal{P}(m)$ and $x_p = \ominus a_1/a_0$.*

Theorem 7. *Suppose* $p \equiv 5 \mod 6$. *Let* m *be a divisor of* $p-1$ *or* $p+1$ *and let* $\mathcal{N}(\mathcal{C}_{\mathcal{P}}(m), x_p)$ *be the number of permutations* $\mathcal{P}(x) = (a_0 x \oplus a_1)^{p-2} \oplus a_2 \in \mathbb{F}_p[x]$ *with given* $x_p = \ominus a_1/a_0 \in \mathbb{F}_p$ *and cycle decomposition* $\mathcal{C}_{\mathcal{P}}(m)$, *then*

$$\begin{aligned} \mathcal{N}(\mathcal{C}_{\mathcal{P}}(m), x_p) &= p-1 \quad \text{if } m = 1, 2, \quad \text{and} \\ \mathcal{N}(\mathcal{C}_{\mathcal{P}}(m), x_p) &= \frac{p-1}{2}\varphi(m) \quad \text{if } m > 2. \end{aligned}$$

An entirely different point of view regarding dispersion may help understanding the relation between APN and Costas permutations , see ([5]) for related work.
As mentioned above, the maximum possible dispersion is attained by permutations with Costas property. Recall that Costas permutation is precisely the one, which has no repetitions in the rows of its *difference triangle*, where the difference triangle of a permutation $P \in \mathbb{F}_p[x]$ is a triangular array $DT(P)$ of integers, which has $p-1$ *rows* $T_1(P), \ldots, T_{p-1}(P)$, with

$$T_k(P) = (P(k) - P(0), P(1+k) - P(1), \ldots, P(p-1) - P(p-k-1)),$$

for $k = 1, \ldots, p-1$. Calculating in $\mathbb{F}_p$, we obtain a p-difference triangle $DT_p(P)$ of P, whose rows are:

$$T_{p,k}(P) = (P(k) \ominus P(0), P(1 \oplus k) \ominus P(1), \ldots, P((p \ominus 1) \ominus P(p \ominus k \ominus 1)),$$

for $k = 1, \ldots, p-1$. Denoting the number of distinct integers appearing in $T_k(P)$ by $|T_k(P)|$ and distinct elements appearing in $T_{p,k}(P)$ by $|T_{p,k}(P)|$, one has $|D(P)| = \sum |T_k(P)|$ and $|D(P, \mathbb{F}_p)| = \sum |T_{p,k}(P)|$. There are no permutations of $\mathbb{F}_p$ with $|D(P, \mathbb{F}_p)| = p(p-1)/2$, see ([5]). Hence one can consider permutations with at most two repetitions in each row of their p-difference (or difference) triangles. By definition all APN permutations have at most two repetitions in each row of their difference triangles. See ([5]) for a study of the relation between APN and Costas permutations. In particular the APN permutations $\mathcal{P}$ we consider are not Costas for $p > 5$. On the other hand, though they are affine equivalent to x^{p-2}, the number of repeated terms in rows of their difference triangles show a great variety. In view of this fact we regard a permutation P as being close to having Costas property if $\gamma(P)$ is close to 1. We end this talk by describing how the choice of parameters a_0, a_1, a_2 affects $\mathcal{P}$ to being close to or far away from being Costas.
Further remarks and proofs of above results can be found in ([2]).

References

[1] A. Çeşmelioğlu, W. Meidl, A. Topuzoğlu, On the cycle structure of permutation polynomials, *Finite Fields Appl.* 14 (2008), 593–614.

[2] A. Çeşmelioğlu, W. Meidl, A. Topuzoğlu, On a class of APN permutation polynomials, preprint (2010).

[3] C.J. Corrada-Bravo, I.M. Rubio, Deterministic interleavers for Turbo codes with random-like performance and simple implementation, Proc. of the 3rd International Symposium on Turbo Codes and Related Topics, Brest, France, 2003, 555–558.

[4] K. Drakakis, A review on Costas arrays, J. Appl. Math. 2006.

[5] K. Drakakis, G. McGuire, APN permutations on $\mathbb{Z}_n$ and Costas arrays, Discrete Applied Mathematics 157, 3320–3326.

[6] S.W. Golomb, G. Gong, The status of Costas arrays, IEEE Trans. Inform. Theory 53 (2007), 4260–4265.

[7] C. Heegard, S.B. Wicker, Turbo Coding, Kluwer Academic Publishers, Dordrecht, 1999.

[8] I.M. Rubio, C.J. Corrada-Bravo, Cyclic decomposition of permutations of finite fields obtained using monomials, In: A. Poli, H. Stichtenoth H (Eds.), Proc. of Finite fields and applications 7, Lecture Notes in Computer Science Vol 2948., pp. 254–261, Springer-Verlag, Berlin (2004).

[9] I.M. Rubio, G.L. Mullen, C.J. Corrada, F.N. Castro, Dickson permutation polynomials that decompose in cycles of the same length, In: G.L. Mullen, D. Panario, I. Shparlinski (Eds.), Proc. of Finite fields and applications 8, Contemporary Mathematics Vol 461, 229–239.

On the Quasi-cyclicity of the Gray Image of Codes over Finite Chain Rings

H. Tapia-Recillas
Departamento de Matemáticas
Universidad Autónoma Metropolitana-I
09340 México, D.F., MEXICO
(htr@xanum.uam.mx)

Abstract

A characterization of λ-cyclic codes defined over a finite chain ring in terms of its Gray map image is given. This generalize the corresponding result over the ring $\mathbb{Z}_4$ of integers modulo 4, treated by J. Wolfmann.

1 Introduction

After the work of [3], [4] and [7] where several binary non-linear codes are described as the Gray map image of a linear code over the ring $\mathbb{Z}_4$ of integers modulo 4, the study of codes over finite rings, including $\mathbb{Z}_{p^n}$ where p is a prime and $n > 1$ and integer, Galois rings, finite chain rings and Frobenius rings has increased in recent years. In a series of papers J. Wolfmann undertook the study of various aspects of codes defined over the ring $\mathbb{Z}_4$ and, recently, other topics of Coding Theory. Since the ring $\mathbb{Z}_4$ is a particular case of the ring $\mathbb{Z}_{p^n}$ and these rings can be thought of as Galois rings, several results of codes over $\mathbb{Z}_4$ have been extended to codes over this latter ring. Furthermore, Galois rings are a subclass of the finite chain ring class, and a natural question to ask is to what extent the known results about codes defined over the just mentioned rings are valid for codes defined over finite chain rings. In this talk necessary and sufficient conditions for a code over a finite chain ring to be λ-cyclic is given in terms of its Gray image. This is an immediate consequence of a relation involving the Gray map, the usual cyclic shift and a mapping defined in terms of a special unit λ of the ring (see §2 for definitions). If $R = \mathbb{Z}_4$ the λ-cyclic codes are called negacyclic, which J. Wolfmann has studied extensively ([10]).

2 The result

Let R be a (commutative) finite chain ring of nilpotency index $t+1$. Let $\mathbb{F} = \mathbb{F}_q$ be its residue field, where $q = p^m$, p is a prime, $M = \langle \pi \rangle$ the maximal ideal, U the group of units and T a Teichmüller set of representatives of R (for details see [6]). Examples of such rings include $\mathbb{Z}_{p^n}$, the ring of integers modulo p^n; Galois rings $GR(p^n, m)$ of characteristic p^n and, residue polynomial rings $K[x]/\langle f(x)^k \rangle$, where K is a finite field and $f(x)$ is an irreducible polynomial.

For an integer $n > 1$ let

$$\Phi : R^n \longrightarrow \mathbb{F}^{nq^t},\ \Phi(A) = C_0 \otimes r_0(A) + \cdots + C_t \otimes r_t(A) \tag{1}$$

be the Gray map as introduced in [2].

Let $\lambda = 1 - \pi^t \in U$, $\sigma : R^n \longrightarrow R^n$ the usual cyclic shift, $\eta_\lambda : R^n \longrightarrow R^n$ defined as $\eta_\lambda(A) = (a_0, a_1, ..., \lambda a_{n-1})$ and let

$$\nu_\lambda : R^n \longrightarrow R^n,\ \nu_\lambda(A) = \sigma(\eta_\lambda(A)). \tag{2}$$

For integers $m, s > 1$, an element $\underline{a} \in \mathbb{F}^{ms}$ will be written as $\underline{a} = (a^{(1)}, ..., a^{(s)})$ where each $a^{(i)},\ i = 1, 2, ..., s$ has m entries. If σ is the usual cyclic shift, then.

$$\sigma^{\otimes s} : \mathbb{F}^{ms} \longrightarrow \mathbb{F}^{ms},\ \sigma^{\otimes s}(\underline{a}) = (\sigma(a^{(1)}), ..., \sigma(a^{(s)})), \tag{3}$$

i.e., σ is applied to each block $a^{(i)}$ of length m. Then we have the following:

Proposition 1 *With the notation as above,*

$$\Phi \circ \nu_\lambda = \sigma^{\otimes p^{mt-1}} \circ \Phi. \tag{4}$$

A code $\mathcal{C} \subset R^n$ is called λ-cyclic if $\nu_\lambda(\mathcal{C}) = \mathcal{C}$. Also, a code $C \subset \mathbb{F}^{sm}$ is said to be *quasi-cyclic of index* s if $\sigma^{\otimes s}(C) = C$. As a consequence of Proposition 1 we have,

Corollary 2 *Let $\mathcal{C}$ be a R-code of length n. Then $\mathcal{C}$ is λ-cyclic if and only if its Gray image $\Phi(\mathcal{C})$ is quasi-cyclic of index p^{mt-1}.*

Proof: If $\mathcal{C}$ is a λ-cyclic R-code of length n, from relation (4) it follows that

$$\Phi(\nu_\lambda(\mathcal{C})) = \Phi(\mathcal{C}) = \sigma^{\otimes p^{mt-1}}(\Phi(\mathcal{C})),$$

showing that $\Phi(\mathcal{C})$ is quasi-cyclic of index p^{mt-1}. The converse also follows from relation (4) and the fact that the Gray map Φ is injective. □

Observation. The case $R = \mathbb{Z}_4$ where $\lambda = 1 - 2 = -1$ was considered by J. Wolfmann in [9]; in this case the λ-cyclic codes are called *negacyclic*. These codes were

also studied over finite chain rings in [1]. The case of codes over the ring $R = \mathbb{Z}_{p^2}$ where $\lambda = 1 - p$ was treated in [5]. Several authors have obtained results related to λ-cyclicity over other rings including $\mathbb{Z}_{2^k}$ ([8]).

Due to constraints on space the full proof of Proposition 1 is not presented but an idea of it is given. Recall that if $A \in R^n$, then $\Phi(A) \in \mathbb{F}^{nq^t}$, so the right hand side of relation (4) says that $\Phi(A)$ can be thought of as $l = p^{mt-1}$ blocks, $\Phi(A) = (B_1, ..., B_l)$, each B_i having np entries. Similarly, if $\tilde{A} = \nu_\lambda(A)$, then $\Phi(\tilde{A}) = (\tilde{B}_1, ..., \tilde{B}_l)$ where each block $\tilde{B}_i$ also has np entries. Then

$$(\tilde{B}_1, ..., \tilde{B}_l) = \Phi(\tilde{A}) = \sigma^{\otimes p^{mt-1}}(\Phi(A)) = (\sigma(B_1), ..., \sigma(B_l)).$$

In order to prove Proposition 1, the blocks B_i and $\tilde{B}_i$ must be identified and then it must be shown that $\sigma(B_i) = \tilde{B}_i$ for all $i = 1, ..., l$.

For this purpose the elements of the residue field $\mathbb{F}$ will be taken in a special order and representation, as described below. Let $q = p^m$ and let $\{1, \omega, \omega^2, ..., \omega^{m-1}\}$ be a basis of $\mathbb{F} = \mathbb{F}_q$ over $\mathbb{F}_p$. Since there is a bijection between $\mathbb{Z}_{p^m}$ and $\mathbb{F}_{p^m}$ given by

$$h = h_0 + h_1 p + \cdots + h_{m-1} p^{m-1} \longrightarrow \omega_h = h_0 + h_1 \omega + \cdots + h_{m-1} \omega^{m-1},$$

where $0 \leq h_i \leq p - 1$ for $i = 0, 1, ..., m - 1$, the elements of the residue field $\mathbb{F}$ will be taken in the following order:

$$\mathbb{F}_q = \{\omega_h : h = 0, 1, 2, ..., p^m - 1\}.$$

Observe that,

$$\omega_{pi} + k = \omega_{pi+k} \text{ for } 0 \leq k \leq p - 1 \text{ and } 0 \leq i \leq p^{m-1}. \tag{5}$$

Let

$$\Omega_j = (\omega_{jp}, \omega_{jp+1}, ..., \omega_{jp+(p-1)}, \text{ for } j = 0, 1, ..., p^{m-1} - 1,$$

and let $u = (\omega_0, \omega_1, ..., \omega_{q-1}) = (\Omega_0, \Omega_1, ..., \Omega_{l-1}) \in \mathbb{F}_q^q$ be the vector that lists all the elements of the residue field in the above order. Let $v = (1, 1, ..., 1) = (1_p, ..., 1_p)$ be the all one-vector of length q^t where $1_p = (1, ..., 1)$ is the all-one vector of length p.

Let $A = (a_0, ..., a_{n-1}) \in R^n$ which can be written as $A = \rho_0(A) + \rho_1(A)\pi + \cdots + \rho_t(A)\pi^t$ where $\rho_j(A) = (\rho_j(a_0), ..., \rho_j(a_{n-1}))$ and $\rho_j(a_i)$ are those elements of the Teichmüller set T of R appearing in the π-adic representation of a_i. If $\rho(a) \in T$ and $\mu : R \longrightarrow \mathbb{F}$ is the natural residue mapping, let $r(a) = \mu(\rho(a))$, so that $r_j(A) = (r_j(a_0), ..., r_j(a_{n-1}))$.

From properties of the Kronecker product it follows that

$$\begin{aligned} C_0 \otimes r_0(A) &= (\Omega_i \otimes (v \otimes \cdots \otimes v \otimes r_0(A)))_{i=0}^{p^{m-1}-1} \\ &\vdots \\ C_t \otimes r_t(A) &= (1_p \otimes (v \otimes \cdots \otimes v \otimes r_0(A)))_{i=0}^{p^{m-1}-1} \end{aligned}$$

and

$$\begin{aligned} \Omega_i \otimes r_0(A) &= (\omega_{pi} r_0(A), ..., \omega_{pi+(p-1)} r_0(A)) \\ &\vdots \\ 1_p \otimes r_t(A) &= (r_t(A), ..., r_t(A)). \end{aligned}$$

Observe that each term $\Omega_i \otimes r_0(A)$,...,$1_p \otimes r_t(A)$ has np entries. From the definition of the Gray map it can be seen that the block B_i of $\Phi(A)$ for $i = 1, 2, ..., p^{mt-1}$ is:

$$B_i = \Omega_i \otimes r_0(A) + \cdots + 1_p \otimes r_t(A). \tag{6}$$

If $A' = \eta_\lambda(A)$ then $r_j(A') = r_j(A)$ for $0 \leq j \leq t-1$ and $r_t(A') = r_t(A) + \theta$ where $\theta = (0, ..., 0, (p-1)r_0(a_{n-1}))$. Since $\tilde{A} = \nu_\lambda(A) = \sigma(A')$ the corresponding block of $\Phi(A')$ is:

$$\tilde{B}_i = \Omega_i \otimes r_0(\sigma(A')) + \cdots + 1_p \otimes r_t(\sigma(A')). \tag{7}$$

Observing that σ is an automorphism and it commutes with the function r_i (in the appropriate context), then:

$$\tilde{B}_i = (\sigma(r_t(A') + \cdots + \omega_{pi} r_0(A')), ..., \sigma(r_t(A') + \cdots + \omega_{pi+(p-1)} r_0(A'))).$$

Since

$$r_t(A) + \cdots + \omega_{pi+j} r_0(A) = (r_t(a_0) + \cdots + \omega_{pi+j} r_0(a_0), ..., r_t(a_{n-1}) + \cdots + \omega_{pi+j} r_0(a_{n-1}))$$

for $0 \leq j \leq p-1$ then:

$$\tilde{B}_i = (\sigma(r_t(A) + \cdots + \omega_{pi} r_0(A) + \theta), \ldots, \sigma(r_t(A) + \cdots + \omega_{pi+(p-1)} r_0(A) + \theta)).$$

The relations $\omega_{pi} + k = \omega_{pi+k}$ for $0 \leq k \leq p-1$ (cf. (5)) induce a permutation on the np entries of $\tilde{B}_i$ in such a way that the (jp)-th entry goes to the $[(j+1)p]$-th entry for $1 \leq j \leq p-1$. In other words, we have a cyclic shift on the entries $(p, 2p, ..., (p-1)p)$ of $\tilde{B}_i$ leaving the other entries fixed. The composition of this permutation on the entries of $\tilde{B}_i$ together with the cyclic shift on each $\sigma(r_t(A) + \cdots + \omega_{pi+k} r_0(A) + \theta)$ for $0 \leq k \leq p-1$ determines the cyclic shift on the np entries of B_i, showing that

$$\tilde{B}_i = \sigma(B_i)$$

for $1 \leq i \leq p^{mt} - 1$ and hence proving Proposition 1.

References

[1] H.Q. Dinh and S. R. López-Permouth, "Cyclic and Negacyclic Codes Over Finite Chain Rings". *IEEE Trans. Inform. Theory*, vol.50, No.8, pp.1728-1744, 2004.

[2] M. Greferath and S.E. Schmidt , "Gray isometries for Finite Chain rings and a Nonlinear Ternary $(36, 3^{12}, 15)$ Code", IEEE *Trans. Inform. Theory*, vol. 45, pp.2522-2524, Nov., 1999.

[3] A. R. Hammons, Jr., P. V. Kumar, A. R. Calderbank, N. J. A. Sloane, and P. Solé, "The Z_4-linearity of Kerdock, Preparata, Goethals, and related codes," *IEEE Trans. Inform. Theory*, vol. 40, pp.301-319, 1994.

[4] W. Heise, T. Honold, and A. A. Nechaev, "Weighted modules and representations of codes," in *Proc. ACCT 6* (Pskov, Russia, 1998), pp.123-129.

[5] S. Ling and T. Blackford. "$\mathbb{Z}_{p^{k+1}}$-linear Codes. *IEEE Trans. Inform. Theory*, vol. 48, No.9, pp.2592-2605, 2002.

[6] B.R. McDonald, *Finite rings with identity*, Pure and Applied Mathematics. Marcel Deckker, vol. 28, New York, 1974.

[7] A.A. Nechaev, "The Kerdock code in a cyclic form". Math. Appl. vol.1, (1991), pp.365-384 (English translation of Diskret. Mat., 1989).

[8] H. Tapia-Recillas and G. Vega, "On the $\mathbb{Z}_{2^k}$-Linear and Quaternary Codes". *SIAM Journal on Discrete Mathematics*, vol.17, No.1, pp.103-113, 2003.

[9] J. Wolfmann, "Negacyclic and Cyclic Codes over $\mathbb{Z}_4$". *IEEE, Trans. Inform. Theory,* vol. 45, pp.2527-2532, 1999.

[10] J. Wolfmann, "Binary images of cyclic codes over $\mathbb{Z}_4$". *IEEE, Trans. Inform. Theory,* vol. 47, pp.1773-1779, 2001.

New Set of Codes for the Maximum-Likelihood Decoding Problem

M. Barbier

Abstract

The maximum-likelihood decoding problem is known to be NP-hard for general linear and Reed-Solomon codes [1, 4]. In this paper, we introduce the notion of $\mathcal{A}$-covered codes, that is, codes that can be decoded through a polynomial time algorithm $\mathcal{A}$ whose decoding bound is beyond the covering radius. For these codes, we show that the maximum-likelihood decoding problem is reachable in polynomial time in the code parameters. Focusing on binary BCH codes, we were able to find several examples of $\mathcal{A}$-covered codes, including two codes for which the maximum-likelihood decoding problem can be solved in quasi-quadratic time.

Keywords: Maximum-likelihood decoding, perfect codes, covering radius, list decoding.

1 Introduction

Berlekamp, McEliece and Van Tilborg showed in [1] that the maximum-likelihood decoding is a NP-hard problem for general linear codes. Guruswami and Vardy later proved in [4] that this problem applied to the family of Reed-Solomon codes is also NP-hard. We briefly recall below the maximum-likelihood problem.

Definition 1.1 (Maximum-likelihood decoding problem)**.** *Let $\mathcal{C}$ a linear code over $\mathbb{F}_q$ and v a $\mathbb{F}_q$-vector in the ambient space. The* maximum-likelihood decoding problem *is to find the codeword $w \in \mathcal{C}$ closest to v. Most precisely, to find $w \in \mathcal{C}$, such as*

$$d(w, v) = d(v, \mathcal{C}) = \min_{c \in \mathcal{C}} \{d(v, c)\}.$$

Clearly, if for a given code there exists an algorithm able to correct a number of errors at least equal to the covering radius, then this algorithm solves the maximum-likelihood decoding problem. We recall the covering radius definition, which is the largest distance between any vector in ambient space and the code.

Definition 1.2 (Covering radius)**.** *Let $\mathcal{C}$ a linear code over $\mathbb{F}_q$. Its ambient space is a $\mathbb{F}_q$-vector space V. Let $v \in V$, the* covering radius *R of $\mathcal{C}$ is given by*

$$R = \max_{v \in V} \{\min_{c \in \mathcal{C}} d(v, c)\}.$$

In light of Wu's recent algorithmic advances in list decoding [8], we proceed in a comparaison between covering radii and now achievable decoding bounds with such algorithm. This leads us to propose the new algorithmic notion of *$\mathcal{A}$-covered codes* for which maximum-likelihood decoding problem can be carried out in polynomial time, and provide some examples by focussing the family of binary BCH codes. We also exhibit two codes for which the maximum-likelihood decoding problem has quasi-quadratic complexity.

2 $\mathcal{A}$-covered codes

In the rest of this paper, we follow the standard notations of [2] and shall denote by R the covering radius of a code $\mathcal{C}$, and by $t \triangleq \lfloor \frac{d-1}{2} \rfloor$ its error correction capacity. We now recall the definition of a perfect code.

Definition 2.1 (Perfect code)**.** *A code $\mathcal{C}$ with capacity t and covering radius R is called a* perfect code *if and only if*

$$R = t.$$

These codes are of course very interesting from a decoding point of view since each element of their ambient spaces can be decoded. Linear perfect codes are completely classified and for each of them, we know a decoding algorithm up to $t = R$. The maximum-likelihood problem is consequently trivial for perfect codes. This very property prompts us to propose the notion of *$\mathcal{A}$-covered codes* in the context of list decoding. We first introduce the following definitions:

Definition 2.2 (List decoding algorithm)**.** *Let $\mathcal{C}$ a code and v a word in its ambient space. $\mathcal{A}$ is a* list decoding algorithm for $\mathcal{C}$ *up to $\tau_{\mathcal{A}}$ if and only if it returns all codewords $w \in \mathcal{C}$ such that $d(v, w) \leq \tau_{\mathcal{A}}$.*

Definition 2.3 (Polynomial time list decoding algorithm)**.** *Let $\mathcal{C}$ a code, n its length, v a word in its ambient space and $\mathcal{A}$ a list decoding algorithm up to $\tau_{\mathcal{A}}$. $\mathcal{A}$ is a* polynomial time list decoding algorithm *if it runs in $\mathcal{O}(f(n))$, where $f(X) \in \mathbb{R}[X]$.*

We can now present the notion of *$\mathcal{A}$-covered code*.

Definition 2.4 ($\mathcal{A}$-covered code)**.** *Let $\mathcal{C}$ a code with covering radius R and $\mathcal{A}$ a polynomial time list decoding algorithm which decodes $\mathcal{C}$ up to $\tau_{\mathcal{A}}$. $\mathcal{C}$ is an* $\mathcal{A}$-covered code *if and only if*

$$R \leq \tau_{\mathcal{A}}.$$

Remark 2.1. *Since this algorithm runs in polynomial time, the returned list is also of polynomial size.*

Proposition 2.1. *Let $\mathcal{C}$ an* $\mathcal{A}$-covered code*. The maximum-likelihood decoding problem for $\mathcal{C}$, (as given by Definition 1.1) is solvable in a time polynomial in the code parameters.*

As seen before, the notion of *$\mathcal{A}$-covered code* can be seen as a computational analogue to perfect codes, albeit in the list decoding context (see Figure 1).

Unique decoding	List decoding
Perfect code	$\mathcal{A}$-covered code
$R = t$	$R \leq \tau_{\mathcal{A}}$

Figure 1: *Perfect code vs $\mathcal{A}$-covered code*

3 Case of binary BCH codes

While still relatively recent, Wu's list decoding algorithm [8] is already regarded as a significant advance in the coding community. Compared to the Guruswami-Sudan algorithm [3], it exhibits an even better complexity. Moreover, when restricted to binary BCH codes, Wu's method allows decoding up to the binary Johnson bound $\frac{n}{2}(1-\sqrt{1-\frac{2d}{n}})$, whereas Guruswami-Sudan only reaches the smaller general Johnson bound $n(1-\sqrt{1-\frac{d}{n}})$, as shown in Figure 2.

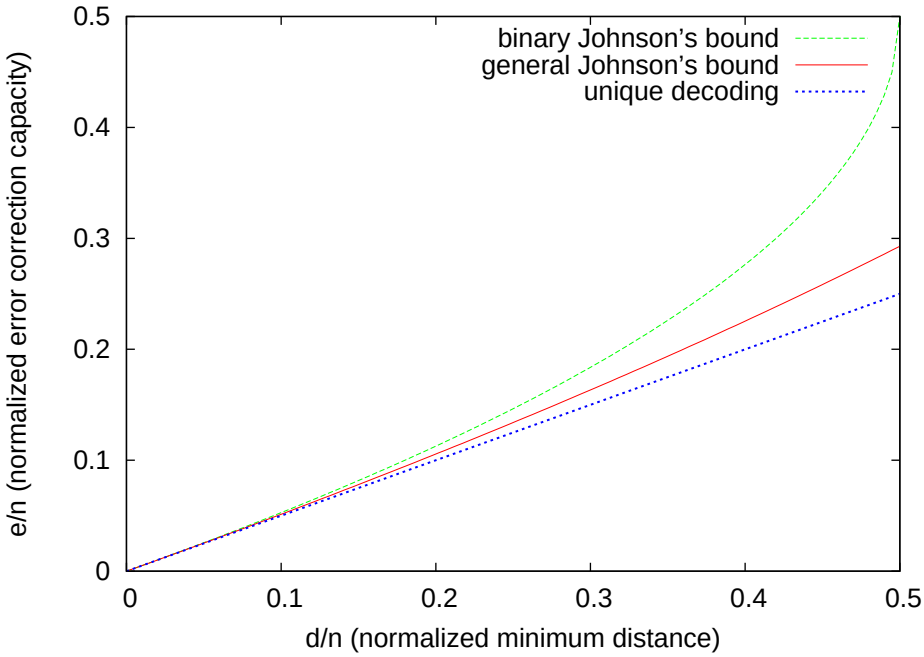


Figure 2: *General and binary Johnson's bound*

It is well known that 1-error-correcting BCH codes are perfect (these are Hamming codes) and 2-error-correcting codes are quasi-perfect [2]. Since Wu's method is a polynomial time list decoding algorithm [8], it is natural to compare their covering radii and the binary Johnson bound of other binary BCH codes. Unfortunately, classifying codes is a hard problem since it requires to compute the covering radii which usually is not an easy task [7]. Putting together and completing data from the literature [2], we still manage to obtain the list in Table 1. This table includes all primitive binary BCH codes of known covering radius. The non-primitive, Wu-covered binary BCH codes of length 17 and 23 were obtained by our own calculations.

Note that the BCH codes having 3 as their minimum distance are Hamming codes. Since those are perfect codes, the maximum-likelihood decoding problem is trivial. We also found two Reed-Muller codes of first order [5]. Since the dimensions of first order Reed-Muller codes are equal to the logarithm of their lengths, naively listing all closest codewords is already a polynomial time decoding algorithm. Hence, knowing that these codes are *Wu-covered* is of little pratical important in solving the maximum-likelihood decoding problem, since easier polynomial time methods are already available.

By contrast, the four codes $[15, 7, 5]$, $[17, 9, 5]$, $[23, 12, 7]$ and $[31, 11, 11]$ given in Table 1 do not fall into the two aforementioned families and we would expect the maximum-likelihood decoding problem to be asymptotically hard. However, the fact that they are *Wu-covered* implies that this problem is actually solvable in polynomial time only (in the code parameters).

n	k	d	R	τ	Comments
7	4	3	1	2	Hamming
15	11	3	1	1	Hamming
15	7	5	3	3	Wu-covered code
15	5	7	5	5	RM(1,4)*
17	9	5	3	3	Wu-covered code
23	12	7	3	4	Wu-covered code
31	26	3	1	1	Hamming
31	21	5	3	2	
31	16	7	5	4	
31	11	11	7	7	Wu-covered code
31	6	15	11	12	RM(1,5)*
63	57	3	1	1	Hamming
63	51	5	3	2	
63	45	7	5	3	
63	39	9	7	4	
63	36	11	9	6	

Table 1: *Table of covering radius and binary Johnson bound for some binary BCH codes.*

4 Quasi-quadratic list decoding of some binary BCH codes

Guruswami-Sudan's algorithm can decode up to Johnson's bound in polynomial time. As McEliece remarked in [6], if we accept to decode slightly less than this bound, the algorithm complexity is dramatically reduced. Under this relaxed constraint, Wu demonstrated in [8] that his algorithm runs in quasi-quadratic time.

Theorem 4.1. *Wu's list decoding algorithm decodes up to*

$$\tau = \lfloor \epsilon t + (1-\epsilon)\frac{n - n\sqrt{1-\frac{2d}{n}}}{2} \rfloor,$$

with multiplicity $m = \lfloor \epsilon^{-1} \rfloor$ *in* $\mathcal{O}(n^2 \lfloor \frac{1}{\epsilon} \rfloor^4)$.

Consequently, binary BCH codes having binary Johnson bound strictly greater than their covering radii, such as binary BCH $[31, 6, 15] = \mathrm{RM}(1, 5)^*$ and BCH $[23, 12, 7]$, can be decoded in quasi-quadratic time up to, and including, their covering radii.

5 Conclusion

Working purely from an algorithmic point of view, we proposed a new set of codes, the $\mathcal{A}$-*covered codes*, for which we showed that the maximum-likelihood decoding problem, known as NP-hard in the general case, is solvable in polynomial time.

The main difficulty in finding such codes lies in the computation of covering radii. However there may be quite a few of those codes as we exhibited nine binary BCH codes which

are *Wu-covered codes*, of which, four constitute a new result and two can be decoded in time quasi-quadratic in code parameters.

References

[1] Berlekamp, McEliece, and Tilborg. On the inherent intractability of certain coding problem. *IEEE Trans. on Inform. Theory*, IT-24(3):384–386, May 1978.

[2] Gérard Cohen, Iiro Honkala, Simon Litsyn, and Antoine Lobstein. *Covering codes*, volume 54 of *North-Holland Mathematical Library*. North-Holland Publishing Co., Amsterdam, 1997.

[3] Venkatesan Guruswami. *List Decoding of Error-Correcting Codes: Winning Thesis of the 2002 ACM Doctoral Dissertation Competition*. Lecture Notes in Computer Science. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[4] Venkatesan Guruswami and Alexander Vardy. Maximum-likelihood decoding of Reed-Solomon codes is NP-hard. In *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 470–478, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.

[5] F. J. Macwilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*. North-Holland Mathematical Library. North Holland, January 1988.

[6] R. J. McEliece. The Guruswami-Sudan decoding algorithm for Reed-Solomon codes. Technical report, The Interplanetary Network Progress Report 42-153, April 2003. http://ipnpr.jpl.nasa.gov/progress_report/42-153/153F.pdf.

[7] A. McLoughlin. The complexity of computing the covering radius of a code. *Information Theory, IEEE Transactions on*, 30(6):800–804, November 1984.

[8] Yingquan Wu. New list decoding algorithms for Reed-Solomon and BCH codes. *IEEE Trans. Inform. Theory*, 54(8):3611–3630, 2008.

A Distinguisher for High Rate McEliece Cryptosystem – Extended Abstract –

Jean-Charles Faugère[1], Ayoub Otmani[2,3], Ludovic Perret[1], and Jean-Pierre Tillich[2]

[1] SALSA Project - INRIA (Centre Paris-Rocquencourt)
UPMC, Univ Paris 06 - CNRS, UMR 7606, LIP6
104, avenue du Président Kennedy 75016 Paris, France
jean-charles.faugere@inria.fr, ludovic.perret@lip6.fr
[2] SECRET Project - INRIA Rocquencourt
Domaine de Voluceau, B.P. 105 78153 Le Chesnay Cedex - France
ayoub.otmani@inria.fr, jean-pierre.tillich@inria.fr
[3] GREYC - Université de Caen - Ensicaen
Boulevard Maréchal Juin, 14050 Caen Cedex, France.

Abstract. The purpose of this talk is to study the difficulty of the Goppa Code Distinguishing (GD) problem, which is the problem of distinguishing the public matrix in the McEliece cryptosystem from a random matrix. It is widely believed that this problem is computationally hard as proved by the increasing number of papers using this hardness assumption. One can consider that disproving/mitigating this hardness assumption is a breakthrough in code-based cryptography. In this paper, we present an efficient distinguisher for alternant and Goppa codes over binary/non binary fields. Our distinguisher is based on a recent algebraic attack against compact variants McEliece which reduces the key-recovery to the problem of solving an algebraic system of equations. We exploit a defect of rank in the (linear) system obtained by linearizing this algebraic system. It turns out that our distinguisher is also highly discriminant. Indeed, we are able to precisely quantify the defect of rank for "generic" binary and non-binary random, alternant and Goppa codes. We have verified these formulas with practical experiments, and a theoretical explanation for such defect of rank is also provided. To our knowledge, this is the first serious cryptographic weakness observed on McEliece since thirty years.

Keywords : public-key cryptography, McEliece cryptosystem, algebraic cryptanalysis, distinguisher.

1 Introduction

Code-based public key cryptography appeared with McEliece's pioneering work [21] where the author proposed one-way trapdoor functions based on irreducible binary Goppa codes. The class of Goppa codes represents one of the most important example of linear codes having an efficient decoding algorithm [4, 25]. A binary Goppa code requires in order to be defined a polynomial $g(z)$ of degree $r \geq 1$ with coefficients in some extension $\mathbb{F}_{2^m}$ of degree $m > 1$ over $\mathbb{F}_2$, and a n-tuple $\mathscr{L} = (x_1, \ldots, x_n)$ of distinct elements in $\mathbb{F}_{2^m}$ with $n \leq 2^m$. The trapdoor of the McEliece public-key scheme consists of the randomly picked $g(z)$ with $\mathscr{L}$ which together provide all the information to decode efficiently. The public key is a randomly picked generator matrix of the chosen Goppa code. A ciphertext is obtained by multiplying a plaintext with the public generator matrix and adding a random error vector of prescribed Hamming weight. The receiver decrypts the message thanks to the decoding algorithm that can be derived from the secrets. Niederreiter [23] brings a significant modification of the McEliece cryptosystem by proposing to describe public linear codes through parity-check matrices. The resulting public key cryptosystem is as as secure as the McEliece's one. The first code-based signature scheme came out in [10] almost twenty years after the McEliece's

proposal. The only difference between the encryption and the signature scheme rests on the choice of the parameters of the binary Goppa codes. In order to be efficient, Goppa codes for the signature have to be chosen such that they correct very few errors. This leads to a very high rate $R = k/n$ with n is its length and k being the dimension of the code. It holds that $k = n - tm$ where by definition t is the number of errors and generally n is chosen to be equal to 2^m. For instance [16], a 80-bit security signature scheme imposes $t = 10$ and $m = 21$ which leads to $R = 0.99$.

All these cryptographic primitives base their security under two assumptions: the intractability of decoding random linear codes [3], and the difficulty of recovering the private key or an equivalent one. The problem of decoding an unstructured code is a long-standing problem whose most effective algorithms [17, 18, 28, 8, 5] have an exponential time complexity. Thus, one may reasonably not expect much progress in this direction. On the other hand no significant breakthrough has been observed during the last thirty years regarding the problem of recovering the private key. Indeed, although some weak keys have been identified in [19], the only known key-recovery attack is the exhaustive search of the secret polynomial $\Gamma(z)$ of the Goppa code, and applying the *Support Splitting Algorithm* (SSA) [26] to check whether the Goppa code candidate is *permutation-equivalent* to the code defined by the public generator matrix. Despite the fact that there still does not exist a practical attack against the McEliece's proposal of using binary Goppa codes, one should not exclude the possibility of breakthrough in that field. The authors of [10] alleviated the McEliece assumptions by introducing the *Goppa Code Distinguishing (GD) problem*. It assumes that there exists no polynomial time algorithm that distinguishes a generator matrix of a Goppa code from a random generator matrix which is a classical belief in code-based cryptography. For instance, according to the authors of [10], proving or disproving the hardness of the GD problem will have a significant impact : *"Classification issues are in the core of coding theory since its emergence in the 50's. So far nothing significant is known about Goppa codes, more precisely there is no known property invariant by permutation and computable in polynomial time which characterizes Goppa codes. Finding such a property or proving that none exists would be an important breakthrough in coding theory and would also probably seal the fate, for good or ill, of Goppa code-based cryptosystems".* Currently, the only known algorithm that solves GD problem is based on the enumeration of Goppa codes and the SSA algorithm [26], as explained below. The time complexity of this method is $\mathcal{O}(2^{mt})$ if one assumes that the cost of the SSA algorithm is negligible (which is a reasonable assumption for Goppa codes, but not for all linear codes).

As a consequence, it is widely believed that distinguishing the public matrix in McEliece from a random matrix is computationally hard. Furthermore, the hardness of the Goppa Code Distinguishing (GD) problem is mandatory to prove the semantic and CCA2 security of McEliece in the random oracle model and in the standard model [24, 13, 6], the security in the random oracle model against existential forgery [10, 11] of the CFS signature [10] scheme, the provable security of several primitives such as a threshold ring signatures scheme [12], an identity-based identification scheme [9], which are build upon CFS. Therefore, showing that the Goppa Code Distinguishing problem is easier than expected will "unprove" most of the provable primitives based on McEliece, and more importantly will be the first serious cryptographic weakness observed on this scheme since thirty years. The purpose of this paper is to study the difficulty of the Goppa Code Distinguishing (GD) problem:

Definition 1 (Goppa Code Distinguishing (GD) Problem). *Let n and k be two integers such that $k \leq n$. We denote by $\mathsf{Goppa}(n,k)$ the set of $k \times n$ generator matrices of Goppa codes. Similarly, $\mathsf{Random}(n,k)$ is the set of $k \times n$ random generator matrices. A* distinguisher *$\mathcal{D}$ is an algorithm that takes as input a matrix $\mathbf{G}$ and returns a bit. We say that $\mathcal{D}$ solves the GD problem if it wins the following game:*

– $b \xleftarrow{R} \{0,1\}$ *If* $b = 0$ *then* $\mathbf{G} \xleftarrow{R} \mathsf{Goppa}(n,k)$ *otherwise* $\mathbf{G} \xleftarrow{R} \mathsf{Random}(n,k)$
– *If* $\mathscr{D}(\mathbf{G}) = b$ *then* $\mathscr{D}$ *wins the games else* $\mathscr{D}$ *loses.*

$\mathscr{D}$ *outputs* 1 *with probability* $\Pr[H \xleftarrow{R} \mathsf{Goppa}(n,k) : \mathscr{D}(H) = 1]$ *if* H *is a random binary parity check matrix of a Goppa code* $\mathsf{Goppa}(n,k)$ *and outputs* 1 *with probability* $\Pr[H \xleftarrow{R} \mathsf{Random}(n,k) : \mathscr{D}(H) = 1]$ *We call the advantage of a distinguisher* $\mathscr{D}$ *the following quantity:*

$$Adv^{GD}(\mathscr{D}) = \left|\Pr[H \xleftarrow{R} \mathsf{Goppa}(n,k) : \mathscr{D}(H) = 1] - \Pr[H \xleftarrow{R} \mathsf{Random}(n,k) : \mathscr{D}(H) = 1]\right|.$$

In this paper, we present a polynomial-time and deterministic distinguisher for solving the GD problem defined below with a maximal advantage. Along the way, we will also solve the code distinguishing problem for alternant codes. The key ingredient is to use a new algebraic technique introduced in [15] to attack two variants [1, 22] of McEliece. Namely, it has been observed [15] that a key recovery attack of these cryptosystems, as well as the genuine McEliece's system, can be reduced to solving the following algebraic set of equations:

$$\left\{ g_{i,1} Y_1 X_1^j + \cdots + g_{i,n} Y_n X_n^j = 0 \;\middle|\; i \in \{1,\ldots,k\}, j \in \{0,\ldots,r-1\} \right\} \tag{1}$$

where the unknowns are the X_i's and the Y_i's and the $g_{i,j}$'s are known coefficients (with $1 \leq i \leq k, 1 \leq j \leq n$) which are nothing but the coefficients (belonging to a certain field $\mathbb{F}_{2^s}$) of the public generator matrix of the scheme. Finally, k is equal to $n - mr$ here, where m is some divisor of s. In other words we have $2n$ unknowns and $rk = r(n - mr)$ polynomial equations. In the cases of [1, 22], additional structures permits to drastically reduce the number of variables and solve (1) efficiently using dedicated Gröbner bases techniques [15]. For the McEliece cryptosystem, solving (1) seems to be out of the scope of such dedicated techniques. However, we will see that the can use this algebraic approach to construct an efficient distinguisher.

Summary of the Results. In the first part of the talk, we will precisely explain how we can mount an algebraic attack against McEliece-like schemes, i.e. namely how the algebraic system (1) is constructed. Recall that solving (1) allows to recover the secret-key is McEliece-like schemes. The distinguisher that we will present is actually due to an unsuccessful attempt to solve the algebraic system (1) by linearization. Typically for the parameters proposed in CFS [10], there is a special method to linearize (1) yielding to a linear system with as many variables as equations. It turns out that the linearized system is not of full rank. Although this is an obstacle to break the system, this particular feature permits to construct an efficient distinguisher for alternant, binary and non-binary Goppa codes. Note that the distinguisher is efficient since we only have to compute the rank of a linear system. It turns out that our distinguisher is also highly discriminant. We will present explicit formulas on the dimension of the kernel of the linearized system for "generic" random, alternant, binary and non-binary Goppa codes. We performed extensive experiments to compare our theoretical results with practical results obtained on valid public matrices of McEliece. The results confirm that the generic formula obtained are accurate on concrete instances. As already pointed out, this is the first serious cryptographic weakness observed on Goppa codes since the initial proposal of McEliece 30 years ago. We emphasize that the Goppa Code Distinguishing problem has been widely considered as a hard problem in code-based cryptography as proved by the increasing number of papers using this assumption [24, 13, 6, 10–12, 9]. To our point of view, disproving/mitigating this hardness assumption is a breakthrough in code-based cryptography and may open a new direction to attack the McEliece cryptosystem. Finally, we will explain how the formulas have been obtained. To do so, we have used the combinatorial properties of the linearized system as well as the distinguishing features of Alternant/Goppa codes.

References

1. T. P. Berger, P.L. Cayrel, P. Gaborit, and A. Otmani. Reducing key length of the McEliece cryptosystem. In Bart Preneel, editor, *Progress in Cryptology - Second International Conference on Cryptology in Africa (AFRICACRYPT 2009)*, volume 5580 of *Lecture Notes in Computer Science*, pages 77–97, Gammarth, Tunisia, June 21-25 2009.
2. Thierry P. Berger. On the cyclicity of Goppa codes, parity-check subcodes of Goppa codes, and extended Goppa codes. *Finite Fields and their applications*, 6(3):255–281, 2000.
3. E. Berlekamp, R. McEliece, and H. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24(3):384–386, May 1978.
4. E. R. Berlekamp. Factoring polynomials over finite fields. In E. R. Berlekamp, editor, *Algebraic Coding Theory*, chapter 6. McGraw-Hill, 1968.
5. D. J. Bernstein, T. Lange, and C. Peters. Attacking and defending the McEliece cryptosystem. In *PQCrypto*, volume 5299 of *LNCS*, pages 31–46, 2008.
6. Bhaskar Biswas and Nicolas Sendrier. Mceliece cryptosystem implementation: Theory and practice. In *PQCrypto*, pages 47–62, 2008.
7. W. Bosma, J. J. Cannon, and Catherine Playoust. The Magma algebra system I: The user language. *J. Symb. Comput.*, 24(3/4):235–265, 1997.
8. A. Canteaut and F. Chabaud. A new algorithm for finding minimum-weight words in a linear code: Application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511. *IEEE Transactions on Information Theory*, 44(1):367–378, 1998.
9. Pierre-Louis Cayrel, Philippe Gaborit, David Galindo, and Marc Girault. Improved identity-based identification using correcting codes. *CoRR*, abs/0903.0069, 2009.
10. N. T. Courtois, M. Finiasz, and N. Sendrier. How to achieve a McEliece-based digital signature scheme. *Lecture Notes in Computer Science*, 2248:157–174, 2001.
11. Léonard Dallot. Towards a concrete security proof of courtois, finiasz and sendrier signature scheme. In *WEWoRC*, pages 65–77, 2007.
12. Léonard Dallot and Damien Vergnaud. Provably secure code-based threshold ring signatures. In *IMA Int. Conf.*, pages 222–235, 2009.
13. Rafael Dowsley, Jörn Müller-Quade, and Anderson C. A. Nascimento. A cca2 secure public key encryption scheme based on the mceliece assumptions in the standard model. In *CT-RSA*, pages 240–251, 2009.
14. Arne Dür. The automorphism groups of Reed-Solomon codes. *Journal of Combinatorial Theory, Series A*, 44:69–82, 1987.
15. Jean-Charles Faugère, Ayoub Otmani, Ludovic Perret, and Jean-Pierre Tillich. Algebraic cryptanalysis of McEliece variants with compact keys. In *Proceedings of Eurocrypt 2010*. Springer Verlag, 2010. to appear.
16. M. Finiasz and N. Sendrier. Security bounds for the design of code-based cryptosystems. In M. Matsui, editor, *Asiacrypt 2009*, volume 5912 of *LNCS*, pages 88–105. Springer, 2009.
17. P. J. Lee and E. F. Brickell. An observation on the security of McEliece's public-key cryptosystem. In *Advances in Cryptology - EUROCRYPT'88*, volume 330/1988 of *Lecture Notes in Computer Science*, pages 275–280. Springer, 1988.
18. J. S. Leon. A probabilistic algorithm for computing minimum weights of large error-correcting codes. *IEEE Transactions on Information Theory*, 34(5):1354–1359, 1988.
19. P. Loidreau and N. Sendrier. Weak keys in the mceliece public-key cryptosystem. *IEEE Transactions on Information Theory*, 47(3):1207–1211, 2001.
20. F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*. North–Holland, Amsterdam, fifth edition, 1986.
21. R. J. McEliece. *A Public-Key System Based on Algebraic Coding Theory*, pages 114–116. Jet Propulsion Lab, 1978. DSN Progress Report 44.
22. R. Misoczki and P. S. L. M. Barreto. Compact McEliece keys from Goppa codes. In *Selected Areas in Cryptography (SAC 2009)*, Calgary, Canada, August 13-14 2009.
23. H. Niederreiter. A public-key cryptosystem based on shift register sequences. In *EUROCRYPT*, volume 219 of *LNCS*, pages 35–39, 1985.
24. Ryo Nojima, Hideki Imai, Kazukuni Kobara, and Kirill Morozov. Semantic security for the mceliece cryptosystem without random oracles. *Des. Codes Cryptography*, 49(1-3):289–305, 2008.
25. N. Patterson. The algebraic decoding of Goppa codes. *IEEE Transactions on Information Theory*, 21(2):203–207, 1975.
26. N. Sendrier. Finding the permutation between equivalent linear codes: The support splitting algorithm. *IEEE Transactions on Information Theory*, 46(4):1193–1203, 2000.
27. V.M. Sidelnikov and S.O. Shestakov. On the insecurity of cryptosystems based on generalized Reed-Solomon codes. *Discrete Mathematics and Applications*, 1(4):439–444, 1992.
28. J. Stern. A method for finding codewords of small weight. In G. D. Cohen and J. Wolfmann, editors, *Coding Theory and Applications*, volume 388 of *Lecture Notes in Computer Science*, pages 106–113. Springer, 1988.

A NOTE ON YAO'S THEOREM ABOUT PSEUDO-RANDOM GENERATORS

STÉPHANE BALLET AND ROBERT ROLLAND

1. Introduction

The study of provable security notions can be done from different points of view:

(1) the study can be done in a static context, with given parameters. In this case, the sizes of the objects are fixed.
(2) on the contrary the study can be done in a dynamic context, namely the system depends on a variable parameter (the so-called security parameter) growing to infinity.

In the paper [Yao(1982)], the study is done in an asymptotic context an for one pseudo-random generator for each value of the security parameter. In this paper, we present modified versions of Yao's results, in particular a static version for a pseudo-random generator and for a finite family of pseudo-random generators. In each case, we give detailed proofs and we compute the exact cost of the reduction between the notion of indistinguishability and the notion of unpredictability of the next bit. We stress that this point of view can be of interest for a practical study of concrete pseudo-random generators with a fixed length. Finally we derive from the preceding results an asymptotic version which generalizes the classical Yao's theorem to a family of pseudo-random generators having the same security parameter k, as k tends to infinity.

2. Pseudo-random generators

2.1. Definition of a pseudo-random generator.

Definition 2.1. *A pseudo-random generator is a deterministic function f defined on a subset $\mathbf{U} \subseteq \{0,1\}^k$ into $\{0,1\}^l$ (where $k < l$) which maps a seed $X_0 \in \mathbf{U}$ (a secret seed) to a finite sequence of l bits:*

$$f(X_0) = (x_1, x_2, \cdots, x_l).$$

3. The security of a pseudo-random generator

3.1. Indistinguishable pseudo-random generator.

Let us consider the following pseudo-random generator:

$$f : \mathbf{U} \subset \{0,1\}^k \to \{0,1\}^l \text{ where } k < l.$$

If $\mathcal{A}$ is a probabilistic algorithm which outputs one bit, We will denote by $\mu_{\mathcal{A}}(e)$ the probability of the output 1 when the input of $\mathcal{A}$ is e.

Key words and phrases. complexity, cryptography, distinguishable, prediction, pseudo-randomness, pseudo-random number generator, Yao's theorem.

The following random experiment, involves a probabilistic algorithm $\mathcal{A}$ having for input a vector $Y \in \{0,1\}^l$ and which outputs one bit. It has for aim to recognize if a sequence Y is the output of the pseudo-random generator f or of a truly random generator.

Let us fix an integer s such that $0 \leq s \leq l$.

$\mathbf{Expt}_{\mathbf{f,s}}^{\mathbf{dist}}(\mathcal{A})$
  $X_0 \leftarrow \mathbf{U} \subseteq \{0,1\}^k$
  $X \leftarrow f(X_0)$
    (notation : $X = (x_1, \cdots, x_l)$)
  $Y_1 \leftarrow (x_1, x_2, \cdots, x_s)$
  $Y_2 \leftarrow \{0,1\}^{l-s}$
  $Y \leftarrow Y_1 || Y_2$
  $b \leftarrow \mathcal{A}(Y)$
  return b
End.

Let $q_{f,s}$ be the probability that the experiment

$$\mathbf{Expt}_{\mathbf{f,s}}^{\mathbf{dist}}(\mathcal{A})$$

returns $b = 1$.

In particular, $q_{f,0}$ is the probability of the following event: we draw at random an element of $\{0,1\}^l$ according to the uniform distribution, we run the algorithm $\mathcal{A}$ on this element, and the output is 1. The probability $q_{f,l}$ is the probability of the following event: we draw at random a seed X_0 in $\mathbf{U}$, we apply f to obtain an element of $\{0,1\}^l$ which becomes the input of the algorithm $\mathcal{A}$, and the output is 1.

Let us recall now the notion of advantage which permits to quantify the ability of $\mathcal{A}$ to distinguish f.

Definition 3.1. *The advantage of the algorithm $\mathcal{A}$ to distinguish f is*

$$Adv_f^{dist}(\mathcal{A}) = |q_{f,l} - q_{f,0}|.$$

Then we define a (T, ϵ)-distinguisher:

Definition 3.2. *Let f be a pseudo-random generator. Let T and ϵ be positive real numbers. A (T, ϵ)-distinguisher for f is a probabilistic algorithm $\mathcal{A}$ such that*

1. *the maximal running time of $\mathcal{A}$ is $\leq T$,*
2. *the input of $\mathcal{A}$ is an element of $\{0,1\}^l$,*
3. *the output of $\mathcal{A}$ is a bit b,*
4. *the algorithm $\mathcal{A}$ can distinguish the pseudo-random generator from the uniform distribution, namely*

$$Adv_f^{dist}(\mathcal{A}) > \epsilon.$$

3.2. **Unpredictable pseudo-random generator.** Let us consider the following pseudo-random generator:

$$f : \mathbf{U} \subset \{0,1\}^k \to \{0,1\}^l.$$

Let $1 \leq s < l$. The following random experiment involves a probabilistic algorithm $\mathcal{B}$ having for input a sequence of s bits and for output a bit. Roughly speaking, this

algorithm tries to predict the next bit produced by the pseudo-random generator f, namely the bit of index $s+1$.

$\mathbf{Expt}_{\mathbf{f,s}}^{\mathbf{pred}}(\mathcal{B})$
  $X_0 \leftarrow \mathbf{U} \subseteq \{0,1\}^k$
  $X \leftarrow f(X_0)$
    (notation : $X = (x_1, \cdots, x_l)$)
  $Y \leftarrow (x_1, x_2, \cdots, x_s)$
  $b \leftarrow \mathcal{B}(Y)$
  if $b = x_{s+1}$
    then return 1
    else return 0
  fi
End.

Let $r_{f,s}$ be the probability that the experiment $\mathbf{Expt}_{\mathbf{f,s}}^{\mathbf{pred}}(\mathcal{B})$ returns 1.

Definition 3.3. *The advantage of the algorithm $\mathcal{B}$ to predict the bit of index $(s+1)$ computed by f is:*

$$Adv_{f,s}^{pred}(\mathcal{B}) = \left| r_{f,s} - \frac{1}{2} \right|.$$

We can now define the notion of (T, s, ϵ)-prediction algorithm.

Definition 3.4. *Let f be a pseudo-random generator. Let T and ϵ be positive real numbers and s be an integer such that $1 \leq s < l$. A (T, s, ϵ)-prediction algorithm $\mathcal{B}$ is a probabilistic algorithm such that:*

(1) *the maximal running time of $\mathcal{B}$ is $\leq T$,*
(2) *the input of $\mathcal{B}$ is an element of $\{0,1\}^s$,*
(3) *the output of $\mathcal{B}$ is a bit,*
(4) *the algorithm $\mathcal{B}$ can predict the next bit, namely*

$$Adv_{f,s}^{pred}(\mathcal{B}) > \epsilon.$$

3.3. **Yao's theorem, static version.** The Yao's theorem relates the notion of security to the notion of unpredictability of the next bit. We express it in its non-asymptotic form. In this case, we give two results which can be considered respectively as a necessary condition and a sufficient condition to have the security of a generator f.

Theorem 3.5. *We consider the following pseudo-random generator:*

$$f : \mathbf{U} \subset \{0,1\}^k \rightarrow \{0,1\}^l.$$

If we have a (T, s, ϵ)-prediction algorithm for f, we can build a $(T+c, \epsilon)$-distinguisher where c is the constant time needed to compare two bits.

Theorem 3.6. *Let f be a pseudo-random generator:*

$$f : \mathbf{U} \subset \{0,1\}^k \rightarrow \{0,1\}^l.$$

Let us suppose that for all s such that $1 \leq s < l$, it does not exist any (T, s, ϵ)-prediction algorithm. Then f is $(T - (c_1 l + c_2), l\,\epsilon)$-secure where c_1 is the constant time needed to draw one bit at random, and c_2 is the constant time needed to test the value of a bit and then depending upon the value of this bit to return a bit or its complementary.

4. The security of a family of pseudo-random generators with same given size

We have considered the case of one pseudo-random generator f. But even in the non-asymptotic case where k and l are fixed, we have to study not only one, but a family (a finite family because k and l are fixed) Γ of function f defined on a subset $\mathbf{U}_f$ (which can depend on f) of $\{0,1\}^k$ with images in $\{0,1\}^l$.

4.1. Revisiting the previous notions in the case of a family of pseudo-random generators with same size.

In a realistic situation we must, in the random experiment which defines the attacker's advantage, draw at random the function f in the family Γ according to a probability δ. So, we replace now the algorithms $\mathcal{A}$ and $\mathcal{B}$ of the previous section by algorithms whose inputs are a function $f \in \Gamma$ and an a vector. The random experiments $\mathbf{Expt}^{\mathbf{dist}}_{\mathbf{f,s}}(\mathcal{A})$ and $\mathbf{Expt}^{\mathbf{pred}}_{\mathbf{f,s}}(\mathcal{B})$ are replaced by the random experiments $\mathbf{Expt}^{\mathbf{dist}}_{\mathbf{\Gamma,s}}(\mathcal{A})$ and $\mathbf{Expt}^{\mathbf{pred}}_{\mathbf{\Gamma,s}}(\mathcal{B})$ where we draw at random not only the seed X_0, but also the function f itself. Then we extend the previous definitions for the case of a family of pseudo-random generator and we get a an adapted version of the Yao's theorem.

4.2. Yao's theorem.

Theorem 4.1. *Let Γ be a family of pseudo-random generators having the same size where each $f \in \Gamma$ is a function*

$$f : \mathbf{U}_f \subset \{0,1\}^k \to \{0,1\}^l.$$

If we have a (T, s, ϵ)-prediction algorithm for Γ, we can build a $(T+c, \epsilon)$-distinguisher for Γ where c is the constant time needed to compare two bits.

Theorem 4.2. *Let Γ be a family of pseudo-random generators having the same size where each $f \in \Gamma$ is a function*

$$f : \mathbf{U}_f \subset \{0,1\}^k \to \{0,1\}^l.$$

Let us suppose that for all s such that $1 \leq s < l$, it does not exist any (T, s, ϵ)-prediction algorithm. Then f is $(T - (c_1 l + c_2), l\,\epsilon)$-secure where c_1 is the constant time needed to draw one bit at random, and c_2 is the constant time needed to test the value of a bit and then depending upon the value of this bit to return a bit or its complementary.

5. Asymptotic behavior

As a consequence of the previous results for fixed k and l, we can deduce results on the asymptotic theory of the pseudo-random generators. Let k be a positive integer (the security parameter) and $l(k)$ a polynomial function of k such that $l(k) > k$. For any k we have a set Γ_k of deterministic functions such that

1. if $f \in \Gamma_k$ then f is a function from a subset $\mathbf{U}_f$ of $\{0,1\}^k$ into $\{0,1\}^{l(k)}$;
2. there exist a polynomial function $t(k)$ such that for any k, any $f \in \Gamma_k$ and any $X \in U_f$ the computation time of $f(X)$ is upper-bounded by $t(k)$;
3. for any k we provide a probability δ_k on the set Γ_k.

Definition 5.1. *The family $\Gamma = (\Gamma_k)_{k>0}$ of sets of pseudo-random generators is said asymptotically secure if for any polynomial $S(k)$, any integer u and any*

distinguisher $\mathcal{A}$ with running time $\leq S(k)$, the advantage of the algorithm $\mathcal{A}_k$ is a negligible function of $\frac{1}{k^u}$, namely

$$\lim_{k\to+\infty} k^u \, Adv^{dist}_{\Gamma_k}(\mathcal{A}_k) = 0.$$

Let $s = (s_k)_{k\geq 1}$ a sequence of integers such that $1 \leq s_k < l(k)$. We define now a s-prediction algorithm to be a probabilistic polynomial algorithm $\mathcal{B}$ having for inputs the security parameter k, a function $f \in \Gamma_k$ and a vector $Z \in \{0,1\}^{s_k}$, and which outputs a bit. Let k be an integer, we will denote by $\mathcal{B}_k$ the probabilistic algorithm obtained from $\mathcal{B}$ by fixing the first entry to the value k.

Definition 5.2. *The family $\Gamma = (\Gamma_k)_{k>0}$ of sets of pseudo-random generators is said asymptotically unpredictable if for any polynomial $S(k)$, any sequence s and any s-prediction algorithm $\mathcal{B}$ with running time $\leq S(k)$, the advantage of the s_k-prediction algorithm $\mathcal{B}_k$ is a negligible function of $\frac{1}{k^u}$, namely*

$$\lim_{k\to+\infty} k^u \, Adv^{pred}_{\Gamma_k,s_k}(\mathcal{A}_k) = 0.$$

The two notions are related by the following theorem:

Theorem 5.3. *Let $l(k)$ be a polynomial function of one integer variable k such that $l(l) > k$. Let $\Gamma = (\Gamma_k)_{k>0}$ a family of sets, where any set Γ_k is a probability set of random generators mapping a subset of $\{0,1\}^k$ into $\{0,1\}^{l(k)}$ (more precisely, each $f \in \Gamma_k$ has its own definition subset $\mathbf{U}_f \subseteq \{0,1\}^k$). The family Γ is asymptotically secure if and only if it is asymptotically unpredictable.*

References

[Goldreich(1999)] Goldreich O (1999) Modern Cryptography, Probabilistic Proofs and Pseudo-randomness. No. 17 in Algorithms and Combinatorics, Springer

[Goldreich(2001)] Goldreich O (2001) The Foundations of Cryptography, Volume I. Cambridge University Press

[Luby(1996)] Luby M (1996) Pseudorandomness and Cryptographic Applications. Princeton University Press

[Stinson(2005)] Stinson D (2005) Cryptography: Theory and Practice, Third Edition. CRC Press

[Yao(1982)] Yao AC (1982) Theory and Applications of Trapdoor Functions. In: Proceedings of the 23rd IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, pp 80–91

Institut de Mathématiques de Luminy and eRISCS, case 930, F13288 Marseille cedex 9, France

E-mail address: ballet@iml.univ-mrs.fr

Institut de Mathématiques de Luminy and eRISCS, Campus de Luminy, Case 907, 13288 MARSEILLE Cedex 9

E-mail address: robert.rolland@acrypta.fr

Cross Correlation of Sequences, Generalized Bent Functions, Jacobsthal Sums and Related Topics

Tor Helleseth and Alexander Kholosha

The Selmer Center
Department of Informatics, University of Bergen
P.O. Box 7800, N-5020 Bergen, Norway
{Tor.Helleseth,Alexander.Kholosha}@uib.no

1 Introduction

Niho in [1, Theorem 3-7] and Helleseth in [2] studied the cross correlation between two binary m-sequences that differ by the decimation $2^{3k} - 2^{2k} + 2^k + 1$. They proved that the cross-correlation function is four-valued and found the distribution. We take $n = 4k$, an odd prime p and examine p-ary functions having the form $f(x) = \mathrm{Tr}_n\left(ax^d + bx^2\right)$ with $a, b, x \in \mathrm{GF}(p^n)$ and $d = p^{3k} + p^{2k} - p^k + 1$. Note that d is cyclotomic equivalent to the Niho exponent (with 2 changed to p) and that $\gcd(d, p^n - 1) = 2$ since $d = (p^{2k} - 1)(p^k + 1) + 2$.

Given a function $f(x)$ mapping $\mathrm{GF}(p^n)$ to $\mathrm{GF}(p)$, the direct and inverse *Walsh transform* operations on f are defined at a point by the following respective identities:

$$S_f(y) = \sum_{x \in \mathrm{GF}(p^n)} \omega^{f(x) - \mathrm{Tr}_n(yx)} \quad \text{and} \quad \omega^{f(x)} = \frac{1}{p^n} \sum_{y \in \mathrm{GF}(p^n)} S_f(y) \omega^{\mathrm{Tr}_n(yx)}$$

where $\mathrm{Tr}_n() : \mathrm{GF}(p^n) \to \mathrm{GF}(p)$ denotes the absolute trace function, $\omega = e^{\frac{2\pi i}{p}}$ is the complex primitive p^{th} root of unity and elements of $\mathrm{GF}(p)$ are considered as integers modulo p.

According to [3], $f(x)$ is called a *p-ary bent function* (or *generalized bent function*) if all its Walsh coefficients satisfy $|S_f(y)|^2 = p^n$. A bent function $f(x)$ is called *regular* (see [3, Definition 3] and [4, p. 576]) if for every $y \in \mathrm{GF}(p^n)$ the normalized Walsh coefficient $p^{-n/2} S_f(y)$ is equal to a complex p^{th} root of unity, i.e., $p^{-n/2} S_f(y) = \omega^{f^*(y)}$ for some function f^* mapping $\mathrm{GF}(p^n)$ into $\mathrm{GF}(p)$. A bent function $f(x)$ is called *weakly regular* if there exists a complex u having unit magnitude such that $up^{-n/2} S_f(y) = \omega^{f^*(y)}$ for all $y \in \mathrm{GF}(p^n)$. Recently, weakly regular bent functions were shown to be useful for constructing certain combinatorial objects such as partial difference sets, strongly regular graphs and association schemes (see [5, 6]). This justifies why the classes of (weakly) regular bent functions are of independent interest. For a comprehensive reference on monomial and quadratic p-ary bent functions we refer reader to [7].

Taking $a = b = 1$, results in a weakly regular bent function and the exact value of its Walsh transform coefficients (and value distribution) can be found.

This gives the infinite class of nonquadratic generalized bent functions built over the fields of an arbitrary odd characteristic.

Theorem 1. *Let $n = 4k$. Then p-ary function $f(x)$ mapping $\mathrm{GF}(p^n)$ to $\mathrm{GF}(p)$ and given by*

$$f(x) = \mathrm{Tr}_n\left(x^{p^{3k}+p^{2k}-p^k+1} + x^2\right)$$

is a weakly regular bent function. Moreover, for $y \in \mathrm{GF}(p^n)$ the corresponding Walsh transform coefficient of $f(x)$ is equal to

$$S_f(y) = -p^{2k}\omega^{\mathrm{Tr}_k(x_0)/4} \ ,$$

where x_0 is a unique root in $\mathrm{GF}(p^k)$ of the polynomial

$$y^{p^{2k}+1} + (y^2+X)^{(p^{2k}+1)/2} + y^{p^k(p^{2k}+1)} + (y^2+X)^{p^k(p^{2k}+1)/2} \ .$$

In particular, if $y^2 \in \mathrm{GF}(p^{2k})$ then $x_0 = -\mathrm{Tr}_k^{2k}(y^2)$. Also, every value $-p^{2k}\omega^i$ with $i = \{1, \ldots, p-1\}$ occurs $p^{2k-1}(p^{2k}+1)$ times in the Walsh spectrum of $f(x)$ and $-p^{2k}$ occurs $(p^{2k-1}-1)(p^{2k}+1)+1$ times.

The general case when $a, b \in \mathrm{GF}(p^n)$ is much more complicated. It seems to be hard to find the Walsh transform coefficients of $f(x)$ at an arbitrary point, so we calculate the exponential sum of $f(x)$, i.e., $S_f(0)$. This is equal to the cross-correlation function between two sequences of length $(p^n-1)/2$ obtained by the decimation of an m-sequence by d and 2 or can be seen as a codeword weight in the corresponding p-ary linear code. We relate this value to the number of zeros, a particular polynomial has in a cyclic subgroup of order $p^{2k}+1$ of the multiplicative group of $\mathrm{GF}(p^n)$. Moreover, we show that if either $a^{p^k(p^k+1)} \neq b^{p^k+1}$ or $a^2 = b^d$ with $b \neq 0$ then the exponential sum of $f(x)$ is three-valued. Some steps towards finding the distribution of these values are made but the exact distribution remains an open problem. For the remaining options for choosing (a, b), we show that $S_f(0)$ can be expressed using the Jacobsthal sum of order p^k+1 which has the number of possible values growing with k.

2 Related Results

Let ν be a primitive element of $\mathrm{GF}(p^{2k})$ and let C_t $(t = 0, \ldots, p^k)$ denote the *cyclotomic classes* of order p^k+1 in $\mathrm{GF}(p^{2k})^*$, i.e., $C_t = \{\nu^{(p^k+1)i+t} \mid i = 0, \ldots, p^k-2\}$. The number of elements $x \in C_i$ such that $x+1 \in C_j$ is called the *cyclotomic number* and denoted (i, j). Since $-1 \in C_0$ in our case, we can also take $x-1$ in the definition of the cyclotomic numbers. Note that since the cyclotomic numbers of order p^k+1 are *uniform* (see [8]), their values can easily be determined.

Lemma 1. *For any $i, j = 0, \ldots, p^k$, the cyclotomic numbers of order p^k+1 in $\mathrm{GF}(p^{2k})$ are*

$$(i,j) = \begin{cases} 1, & \text{if} \quad i \neq j \text{ and } ij \neq 0 \\ p^k-2, & \text{if} \quad i = j = 0 \\ 0, & \text{otherwise} \ . \end{cases}$$

Following [9, Definition 5.49], for any $a \in \mathrm{GF}(q)^*$, define a *Jacobsthal sum* of order n as

$$H_n(a) = \sum_{x \in \mathrm{GF}(q)} \eta(x^{n+1} + ax) \ ,$$

where $\eta(\cdot)$ is the quadratic character of $\mathrm{GF}(q)$ extended by setting $\eta(0) = 0$. Define also a companion sum

$$I_n(a) = \sum_{x \in \mathrm{GF}(q)^*} \eta(x^n + a) \ .$$

It is well known (see, e.g., [9, Theorem 5.50]) that $I_{2n}(a) = I_n(a) + H_n(a)$.

In our case, $q = p^{2k}$ and we consider $n = p^k + 1$. If $a \in \mathrm{GF}(p^k)$ then, obviously, $I_{p^k+1}(a) = I_{2(p^k+1)}(a) = p^{2k} - 1$ and $H_{p^k+1}(a) = 0$. For any $a \in \mathrm{GF}(p^{2k}) \setminus \mathrm{GF}(p^k)$, we compute

$$I_{p^k+1}(a) = -(p^k + 1)(\eta(a) + 1) \ .$$

Calculating $H_{p^k+1}(a)$ (that is equivalent to calculating $I_{2(p^k+1)}(a)$) is not that easy.

Theorem 2. *For any $a \in \mathrm{GF}(p^{2k}) \setminus \mathrm{GF}(p^k)$, assume $a = a_0 + 2\mu^{1/2}a_1$ with $a_0, a_1 \in \mathrm{GF}(p^k)$ and μ being a primitive element of $\mathrm{GF}(p^k)$. Then*

$$H_{p^k+1}(a) = (N - p^k)(p^k + 1) \quad \textit{and} \quad |H_{p^k+1}(a)| \le 2p^{k/2}(p^k + 1) \ ,$$

where N denotes the number of points on the elliptic curve $f^2 = z^3 - a_0 z^2 + \mu a_1^2 z$ over $\mathrm{GF}(p^k)$ excluding the point at infinity.

3 Calculating the Exponential Sum of $f(x)$

In this section, we consider the function $f(x)$ with arbitrary coefficients $a, b \in \mathrm{GF}(p^n)$. If n is even, let U denote a cyclic subgroup of order $p^{n/2} + 1$ of the multiplicative group of $\mathrm{GF}(p^n)$ (generated by $\xi^{p^{n/2}-1}$, where ξ is a primitive element of $\mathrm{GF}(p^n)$).

Theorem 3. *Let $n = 4k$. For any $a, b \in \mathrm{GF}(p^n)$, define the following p-ary function mapping $\mathrm{GF}(p^n)$ to $\mathrm{GF}(p)$*

$$f(x) = \mathrm{Tr}_n\left(ax^{p^{3k}+p^{2k}-p^k+1} + bx^2\right) \ .$$

Then the Walsh transform coefficient of $f(x)$ evaluated at point zero is equal to

$$S_f(0) = p^{2k}(2N(a, b) - 1) \ ,$$

where $2N(a, b)$ is the number of zeros in U of the polynomial

$$L(X) = b^{p^{2k}} X + aX^{p^k} + bX^{p^{2k}} + a^{p^{2k}} X^{p^{3k}} \ . \tag{1}$$

In what follows, we consider separately the cases when either $a^{p^k(p^k+1)} \neq b^{p^k+1}$ or $a^2 = b^d$ with $b \neq 0$; and when $a^{p^k(p^k+1)} = b^{p^k+1}$ with $a^2 \neq b^d$, where $d = p^{3k} + p^{2k} - p^k + 1$. This covers all the value space for the pairs $(a, b) \neq (0, 0)$.

Proposition 1. *Let $n = 4k$ and take any $a, b \in \mathrm{GF}(p^n)$ such that either $a^2 = b^d$ with $b \neq 0$ or $a^{p^k(p^k+1)} \neq b^{p^k+1}$. Then polynomial $L(X)$ defined in (1) has none, two or four zeros in U, i.e., $N(a, b) \in \{0, 1, 2\}$.*

Proposition 2. *Let $n = 4k$ and take any $a, b \in \mathrm{GF}(p^n)$ such that $a^{p^k(p^k+1)} = b^{p^k+1}$ and $a^2 \neq b^d$. If $2N(a, b)$ is the number of zeros in U of the polynomial $L(x)$ defined in (1) then*

$$N(a,b) = \#\left\{c \in \mathrm{GF}(p^k) \mid (cg)^2 - b^{p^{2k}+1} \text{ is a nonsquare in } \mathrm{GF}(p^{2k})\right\} \ ,$$

where g is any element in $\mathrm{GF}(p^{2k})^*$ with $g^{p^k-1} = -b^{p^{3k}}/a$. Moreover,

$$2N(a,b) = p^k - \frac{H_{p^k+1}\big(-b^{p^{2k}+1}/g^2\big)}{p^k+1} + 1 \ .$$

Thus, the value distribution of $S_f(0)$ when $a^{p^k(p^k+1)} = b^{p^k+1}$ with $a^2 \neq b^d$, is defined by the value distribution of the Jacobsthal sums order $p^k + 1$ and is directly related to the number of elliptic curves with the prescribed number of points. This is the first direct connection between sequences and Jacobsthal sums we are aware of. We find it interesting and believe that this gives an important link between sequences/codes and classical character sums. In the following corollary, we list some basic properties of $N(a, b)$.

Corollary 1. *Under the conditions of Proposition 2,*

(i) $N(a,b) = \begin{cases} N(a^{-1}, b^{-1}), & \text{if } b^{(p^n-1)/2} = 1 \\ p^k + 1 - N(a^{-1}, b^{-1}), & \text{otherwise} \ ; \end{cases}$

(ii) $N(a,b) + N(-a,b) = N(a,b) + N(a,-b) = p^k + 1$;

(iii) $N(-a,b) = \begin{cases} p^k + 1 - N(a^{-1}, b^{-1}), & \text{if } b^{(p^n-1)/2} = 1 \\ N(a^{-1}, b^{-1}), & \text{otherwise} \ ; \end{cases}$

(iv) if $b^{(p^n-1)/2} = 1$ (resp. $b^{(p^n-1)/2} = -1$) then $N(a, b)$ is an even (resp. odd) number;

(v) $\big|N(a,b) - \frac{p^k+1}{2}\big| \leq p^{k/2}$, in particular, $N(a, b)$ is positive and, if $k > 2$ then $N(a, b) > 8$;

(vi) if $p \equiv -1 \pmod 4$, k is odd and $b^{(p^n-1)/2} = 1$ then $N(a,b) = N(-a,b) = (p^k+1)/2$ for $a = \nu^{(p^{2k}-1)/4} b^{d/2}$, where ν is a primitive element of $\mathrm{GF}(p^{2k})$;

(vii) for any $b \in \mathrm{GF}(p^n)^*$,

$$\sum_{a \in \mathrm{GF}(p^n):\ a^{p^k(p^k+1)} = b^{p^k+1},\ a^2 \neq b^d} N(a,b) = (p^k+1)\big(p^k - b^{(p^n-1)/2}\big)/2 \ . \quad (2)$$

References

1. Niho, Y.: Multi-Valued Cross-Correlation Functions Between Two Maximal Linear Recursive Sequences. PhD thesis, University of Southern California, Los Angeles (1972)
2. Helleseth, T.: A note on the cross-correlation function between two binary maximal length linear sequences. Discrete Math. **23**(3) (1978) 301–307
3. Kumar, P.V., Scholtz, R.A., Welch, L.R.: Generalized bent functions and their properties. J. Combin. Theory Ser. A **40**(1) (September 1985) 90–107
4. Hou, X.D.: p-Ary and q-ary versions of certain results about bent functions and resilient functions. Finite Fields Appl. **10**(4) (October 2004) 566–582
5. Pott, A., Tan, Y., Feng, T., Ling, S.: Association schemes arising from bent functions. In Kholosha, A., Rosnes, E., Parker, M., eds.: WCC 09 Preproceedings - The International Workshop on Coding and Cryptography, Bergen (2009) 48–61
6. Tan, Y., Pott, A., Feng, T.: Strongly regular graphs associated with ternary bent functions. J. Combin. Theory Ser. A **117**(6) (August 2010) 668–682
7. Helleseth, T., Kholosha, A.: Monomial and quadratic bent functions over the finite fields of odd characteristic. IEEE Trans. Inf. Theory **52**(5) (May 2006) 2018–2032
8. Baumert, L., Mills, W., Ward, R.L.: Uniform cyclotomy. J. Number Theory **14**(1) (February 1982) 67–82
9. Lidl, R., Niederreiter, H.: Finite Fields. Volume 20 of Encyclopedia of Mathematics and its Applications. Cambridge University Press, Cambridge (1997)

On The Characteristic Polynomial of Frobenius of Supersingular Abelian Varieties Of Dimension up to 7 over Finite Fields

Vijaykumar Singh, Alexey Zatysev, Gary McGuire

May 12, 2010

Abstract

In this article, we derive a list of possible characteristic polynomials of the Frobenius of simple supersingular abelian varieties of dimension $1,\ 2,\ 3,\ 4,\ 5, 6, 7$ over $\mathbb{F}_q$ where $q = p^n,\ \ n$ odd.

1 Introduction

Supersingular abelian varieties have applications in cryptography and coding theory. Identity based ecryption and computation of weights of some Reed Muller codes are some of them. The important (isogeny) invariant which carries most of the information about supersingular curves is the characteristic polynomial of the Frobenius. Here we give a list of such polynomials up to dimension 7 over $\mathbb{F}_q$ where $q = p^n,\ \ n$ odd. We also give the procedure, which extends to all dimensions.

Let A be an abelian variety of dimension g over $\mathbb{F}_q$ where $q = p^n$. The characteristic polynomial $P(X) \in \mathbb{Z}[X]$ of the Frobenius endomorphism is given by

$$P_A(X) = X^{2g} + a_1X^{2g-1} + \cdots + a_gX^g + qa_{g-1}X^{g-1} \cdots + q^g.$$

An abelian variety A is simple if it is not isogenous to a product of abelian varieties of lower dimensions. In that case $P_A(X)$ is either irreducible over $\mathbb{Z}$ or $P(X) = h(X)^e$ where $h(x) \in \mathbb{Z}[X]$ is irreducible, see [8]. An elliptic curve E over $\mathbb{F}_q$ is supersingular if $E(\bar{\mathbb{F}_q})$ has no points of order p. An abelian variety A over $\mathbb{F}_q$ is called supersingular if A is isogenous over $\bar{\mathbb{F}_q}$ to a product of supersingular elliptic curves. Given $P_A(X)$ for an abelian variety of dimension g, we have the following test from Stichtenoth and Xing [5] for supersingularity.

Theorem 1.1. *Suppose $q = p^n$ and suppose that A is an abelian variety of dimension g over $\mathbb{F}_q$. Let $P_A(X) = X^{2g} + a_1X^{2g-1} + \cdots + a_gX^g + qa_{g-1}X^{g-1} \cdots + q^g$ be the characteristic polynomial of the Frobenius endomorphism on A. Then A is supersingular if and only if, for all $1 \leq j \leq g$,*

$$p^{\lceil \frac{jn}{2} \rceil} | a_j.$$

We can factor $P(X)$ over the complex numbers as $P(X) = \prod_{i=1}^{2g}(X - \alpha_i)$ where the α_i are algebraic integers. If A is supersingular then Manin [6] and Oort [4] proved that numbers $\frac{\alpha_i}{\sqrt{q}}$ are roots of unity.

In this article we are interested in the characterization of the Frobenius for simple supersingular abelian varieties of the dimension g over $\mathbb{F}_{q^n}$, where $1 \leq g \leq 7$ and n is odd. We write $P(x)$ instead of $P_A(x)$ when the underlying abelian variety is understood.

2 Frobenius Polynomial

In this section we give the results and procedure for computing the characteristic polynomial of Frobenius.

Theorem 2.1. *If $P(X)$ is irreducible with $a_{2i+1} \neq 0$ for some i, then the a_i's satisfy*

$$(t^{2g} + \frac{a_2 t^{2g-2}}{q} + \ldots + \frac{a_{2i} t^{2g-2i}}{q^i} + \ldots + 1)^2 - \frac{1}{q}(a_1 t^{2g-1} + \frac{a_3 t^{2g-3}}{q} + \ldots + a_1 t)^2 = 0.$$

Proof. By dividing the characteristic polynomial of Frobenius by q^g we get,

$$\frac{P(X)}{q^g} = \frac{X^{2g}}{q^g} + a_1 \frac{X^{2g-1}}{q^g} + \ldots + a_g \frac{X^g}{q^g} + q^{g-1} + \frac{X}{q^g} + 1 = 0.$$

Doing the transformation $\frac{X}{\sqrt{q}} \to t$ over $\mathbb{Q}(\sqrt{q})$ and rearranging we get

$$G(t) := (t^{2g} + \frac{a_2 t^{2g-2}}{q} + \ldots + \frac{a_{2i} t^{2g-2i}}{q^i} + \ldots + 1) + \frac{1}{\sqrt{q}}(a_1 t^{2g-1} + \frac{a_3 t^{2g-3}}{q} + \ldots + a_1 t) = 0.$$

Since $q = p^n$, n odd, $Gal(\mathbb{Q}(\sqrt{q})/\mathbb{Q}) = \{1, \sigma\}$ where $\sigma(\sqrt{q}) = -\sqrt{q}$. Therefore

$$H(t) := G(t)G(t)^\sigma = (t^{2g} + \frac{a_2 t^{2g-2}}{q} + \ldots + \frac{a_{2i} t^{2g-2i}}{q^i} + \cdots + 1)^2 - \frac{1}{q}(a_1 t^{2g-1} + \frac{a_3 t^{2g-3}}{q} + \ldots + a_1 t)^2 = 0.$$

□

The polynomial $H(t)$ defined above has only even powers of t. We proved the following.

Proposition 2.2. *Let $G(t) \in \mathbb{Q}(\sqrt{q})[t]$ (as defined above) be reducible over $\mathbb{Q}(\sqrt{q})$. Then $G(t) = F_1(t)F_2(t)$ where F_1, F_2 are irreducible over $\mathbb{Q}(\sqrt{q})[t]$ with $\deg F_1, \deg F_2 = g$.*

Proposition 2.3. *Assume $a_{2i+1} \neq 0$.*

1. *If $G(t)$ is irreducible over $\mathbb{Q}(\sqrt{q})$ then $H(t)$ is an irreducible cyclotomic factor of $t^m - 1$ of degree $4g$ where $\phi(m) = 4g$ over $\mathbb{Q}$.*

2. *If $G(t)$ is reducible over $\mathbb{Q}(\sqrt{q})$ then $G(t) = F_1(t)F_2(t)$ where at least one of F_1, $F_2 \in \mathbb{Q}(\sqrt{q})[t] \setminus \mathbb{Q}[t]$. If $F_i \in \mathbb{Q}[t]$ then it is an irreducible cyclotomic factor of $t^m - 1$ of degree g where $\phi(m) = g$, else $F_i F_i^\sigma \in \mathbb{Q}[t]$ is an irreducible cyclotomic factor of $t^m - 1$ of degree $2g$ where $\phi(m) = 2g$.*

Proposition 2.4. *If $a_{2i+1} = 0$ for all i then*

1. *If $G(t)$ is irreducible over $\mathbb{Q}$, it is an irreducible cyclotomic factor of $t^m - 1$ of degree g where $\phi(m) = 2g$.*

2. *If $G(t)$ is reducible over $\mathbb{Q}$, then $G(t) = F_1F_2$ where F_i are irreducible cyclotomic factor of $t^m - 1$ of degree g with $\phi(m) = g$.*

Using the above propositions we have the following procedure to derive the characteristic polynomial.

2.1 Procedure

1. Suppose $P(x)$ is irreducible.
   (a) If $a_{2i+1} \neq 0$ for some i.
      i. If $G(t)$ is irreducible, then as in proposition 2.3 find appropriate $H(t)$ and solve for the a_i's.
      ii. If $G(t)$ is reducible then use proposition 2.3 to find appropriate $H(t) = G(t)G(t)^\sigma$ and solve for the a_i's .
   (b) If $a_{2i+1} = 0$ for all i.
      i. If $G(t)$ is irreducible over $\mathbb{Q}$ then use proposition 2.4 to find appropriate cyclotomic factors and compare with $G(t)$ to solve for a_i's.
      ii. If $G(t)$ is reducible over $\mathbb{Q}$ then use proposition 2.4 to compute $G(t) = F_1(t)F_2(t)$ and compare coefficients to solve for a_i's.

2. Suppose $P(x)$ is reducible ie; $P(x) = h(x)^e$ with $e|g$. Then for $\frac{h(x)}{q^{\frac{g}{e}}}$, doing the transformation $\frac{x}{\sqrt{q}} \to t$, we get a new polynomial $G(t)$ whose roots are roots of unity, and using above tools we can find $h(x)$.

Using the procedure above we have proved the following.

Theorem 2.5. *The characteristic polynomial of a simple supersingular abelian variety over $\mathbb{F}_q$, $q = p^n$, n odd is one of the following*

1. dimension 1
   (a) $p = 2 : x^2 \pm \sqrt{2q}x + q$
   (b) $p = 3 : x^2 \pm \sqrt{3q}x + q$
   (c) $x^2 + q$

2. dimension 2
   (a) $p \neq 3 : x^4 - qx^2 + q^2$
   (b) $x^4 + qx^2 + q^2$
   (c) $p = 2 : x^4 \pm \sqrt{pq}x^3 + qx^2 \pm \sqrt{pq}x + q^2$

(d) $p = 5 : x^4 \pm \sqrt{pq}x^3 + 3qx^2 \pm \sqrt{pq}x + q^2$

(e) $(x^2 - q)^2$

(f) $p \neq 2 : x^4 + q^2$

3. dimension 3

(a) $p = 3 : x^6 \pm q\sqrt{pq}x^3 + q^3$

(b) $p = 7 : x^6 \pm \sqrt{pq}x^5 + 3qx^4 \pm q\sqrt{pq}x^3 + 3q^2x^2 \pm q^2\sqrt{pq}x + q^3$

4. dimension 4

(a) $p = 2 : x^8 \pm \sqrt{2q}x^7 + qx^6 - q^2x^4 + q^3x^2 \pm q^3\sqrt{2q}x + q^4$

(b) $p = 3 : x^8 \pm \sqrt{3q}x^7 + 2qx^6 \pm q\sqrt{3q}x^5 + q^2x^4 \pm q^2\sqrt{3q}x^3 + 2q^3x^2 \pm q^3\sqrt{3q}x^+q^4$

(c) $x^8 + q^4$

(d) $x^8 - qx^6 + q^2x^4 - q^3x^2 + q^4$

(e) $p \neq 5 : x^8 + qx^6 + q^2x^4 + q^3x^2 + q^4$

(f) $p \neq 2 : x^8 - q^2x^4 + q^4$

(g) $p = 5 : x^8 \pm \sqrt{5q}x^7 + 2qx^6 \pm q\sqrt{5q}x^5 + 3q^2x^4 \pm q^2\sqrt{5q}x^3 + 2q^3x^2 + \pm q^3\sqrt{5q}x^+q^4$

5. dimension 5

(a) $p = 11 : X^{10} \pm \sqrt{11q}X^9 + 5qX^8 \pm q\sqrt{11q}X^7 - q^2X^6 \pm q^2\sqrt{11q}X^5 - q^3X^4 \pm q^3\sqrt{11q}X^3 + 5q^4X^2 \pm q^4\sqrt{11q}X + q^5.$

6. dimension 6

(a) $p = 2 : x^{12} \pm \sqrt{2q}x^{11} + qx^{10} - q^2x^8 - q^2(\pm\sqrt{2q})x^7 - q^3x^6 - q^3(\pm\sqrt{2q})x^5 - q^4x^4 + q^5x^2 \pm q^5\sqrt{2q}x + q^6$

(b) $p = 2 : x^{12} \pm q\sqrt{2q}x^9 + q^3x^6 \pm q^4\sqrt{2q}x^3 + q^6$

(c) $p = 3 : x^{12} \pm \sqrt{3q}x^{11} + 2qx^{10} \pm 3q\sqrt{3q}x^9 + q^2x^8 - q^3x^6 + q^4x^4 \pm 3q^4\sqrt{3q}x^3 + 2q^5x^2 \pm q^5\sqrt{3q}x + q^6$

(d) $p = 7 : x^{12} \pm \sqrt{7q}x^{11} + 4qx^{10} \pm q\sqrt{7q}x^9 - q^2x^8 - 2q^2(\pm\sqrt{7q})x^7 - 7q^3x^6 - 2q^3(\pm\sqrt{7q})x^5 - q^4x^4 \pm q^4\sqrt{7q}x^3 + 4q^5x^2 \pm q^5\sqrt{7q}x + q^6$

(e) $p = 7 : x^{12} \pm 2\sqrt{7q}x^{11} + 13qx^{10} \pm 8q\sqrt{7q}x^9 + 29q^2x^8 \pm 14q^2\sqrt{7q}x^7 + 41q^3x^6 \pm 14q^3\sqrt{7q}x^5 + 29q^4x^4 \pm 8q^4\sqrt{7q}x^3 + 13q^5x^2 \pm 2q^5\sqrt{7q}x + q^6$

(f) $p = 13 : x^{12} \pm \sqrt{13q}x^{11} + 7qx^{10} \pm 3q\sqrt{13q}x^9 + 15q^2x^8 \pm 5q^2\sqrt{13q}x^7 + 19q^3x^6 \pm 5q^3\sqrt{13q}x^5 + 15q^4x^4 \pm 3q^4\sqrt{13q}x^3 + 7q^5x^2 \pm q^5\sqrt{13q}x + q^6$

(g) $p = 3 : x^{12} \pm 2q\sqrt{3q}x^9 + 5q^3x^6 \pm 2q^4\sqrt{3q}x^3 + q^6$

(h) $x^{12} + qx^{10} + q^2x^8 + q^3x^6 + q^4x^4 + q^5x^2 + q^6$

(i) $p \neq 7 : x^{12} - qx^{10} + q^2x^8 - q^3x^6 + q^4x^4 - q^5x^2 + q^6$

(j) $p \neq 3 : x^{12} - q^3x^6 + q^6$

(k) $x^{12} + q^3x^6 + q^6$

7. dimension 7*: None*

Remarks:

The procedure above can be extended to any genus.

Work is in progress for n even.

An important question is whether the above actually occur as the characteristic polynomial of the Frobenius. Work is in progess on this question. For genus 1, 2 and 3 all of the polynomials listed occur, see [2], [7] and [9] respectively. For genus 4 the following occur as Jacobians of hyperelliptic curves of genus 4 over $\mathbb{F}_q$ where $q = 2^n, n$ odd. An example of each is given here for $q = 2^5$ where α is a primitive root. These are cases (a), (c), (e) of the theorem.

$P(x)$	Curve
$x^8 + \sqrt{2q}x^7 + qx^6 - q^2x^4 + q^3x^2 + q^3\sqrt{2q}x + q^4$	$y^2 + y = x^9 + \alpha^2x^5 + \alpha^9x^3$
$x^8 - \sqrt{2q}x^7 + qx^6 - q^2x^4 + q^3x^2 - q^3\sqrt{2q}x + q^4$	$y^2 + y = x^9 + \alpha^2x^5 + \alpha^{25}x^3$
$x^8 + q^4$	$y^2 + y = x^9 + x^5 + \alpha^3x^3$
$x^8 + qx^6 + q^2x^4 + q^3x^2 + q^4$	$y^2 + y = x^9 + x^5 + \alpha x^3$

References

[1] H-G Ruck,Abelian Surfaces and Jacobian varieties over finite fields, Compositio Math. 76(1990),351-366.

[2] Waterhouse Abelian Varieties over finite fields, ANn.Sci.Ecole Norm.Sup (4) 2 (1969),521-560.

[3] C.P.Xing, On supersingular abelian varieties of dimension two over finite fields, Finite fields and their app. 2, 407-421 (1996).

[4] Frans Oort, Subvarieties of Moduli Spaces, Invent.Math., 24:95-119, 1974.

[5] H. Stichtenoth and C. Xing, On the structure of the divisor class group of a class of curves over finite fields, Arch. Math., Vol. 65 (1995) 141-150.

[6] Yu. I. Manin, The theory of commutative formal groups over fields of finite characteristic, Russ. Math. Surv., 18, No. 6 (1963) 1983.

[7] D. Maisner, E. Nart, appendix of E. W. Howe. Abelian surfaces over finite fields as jacobians.Experiment. Math.. Vol 11, p. 321-337, 2002.

[8] Waterhouse W. C., Milne J. S. Abelian varieties over finite fields. Proceedings of Symposia in Pure Mathematics (1971) 20:5364

[9] E. Nart, C. Ritzenthaler. Jacobians in isogeny classes of supersingular threefolds in characteristic 2. Finite Fields and Their Applications. no 14, p. 676-702, 2008.

COMPLETE ADDITION LAWS ON ABELIAN VARIETIES OVER FINITE FIELDS

CHRISTOPHE ARENE, DAVID KOHEL, AND CHRISTOPHE RITZENTHALER

Keywords: completeness, addition law.

Abstract. The first results on completeness of addition laws, on a given abelian variety A over a field k, were obtained by Lange and Ruppert in 1985 [LR85]. Subsequent work [LR85, LR87, BL95] considered abelian varieties, particularly elliptic curves, over algebraically closed fields. One of their main results is the non existence of a single addition law describing the group law m of A embedded in a projective space. The study of the completeness of Edwards curves, initiated by Bernstein and Lange [BL07], naturally leads us to consider the arithmetic properties of addition laws. In this paper, we give a proof of the existence of a single addition law describing m on the k-rational points of A, provided k has an extension of degree greater than the dimension of A. We also describe the construction of such a law for elliptic curves and Jacobians of genus 2 curves over a finite field. The study of complete addition laws is motivated by their role in avoiding side channel attacks.

1. Introduction

Let A be an abelian variety of dimension g over a field k. We assume that A is embedded in some projective space $\mathbb{P}^n$ over k by a very ample line bundle L and we denote $\iota : A \hookrightarrow \mathbb{P}^n$ the corresponding morphism. We also assume the embedding is projectively normal, but not mention it in the sequel, so that the sections we use further come from polynomials.
The group law $m : (x, y) \mapsto x + y$ can be described locally by bihomogenous polynomials. More precisely, an *addition law* of bidegree (μ, ν) on $\iota(A) \subset \mathbb{P}^n$ is a nonempty open set $U \subset A \times A$ together with an $(n+1)$-tuple of polynomials

$$\mathfrak{p} = (p_0, \ldots, p_n) \in k[X_0, \ldots, X_n, Y_0, \ldots, Y_n]^{n+1},$$

not all zero, bihomogeneous of degree μ in $X_0, \ldots, X_n$ and of degree ν in $Y_0, \ldots, Y_n$ such that we have

$$\iota(x + y) = (p_0(\iota(x), \iota(y)) : \ldots : p_n(\iota(x), \iota(y)))$$

for all $(x, y) \in U$.

Definition 1.1. A set S of addition laws is called *complete* if for any point $(x, y) \in (A \times A)(k)$ there is a law in S defined on an open set U containing (x, y). A set S of addition laws is said *geometrically complete* if the previous

The authors acknowledge the financial support by grant ANR-09-BLAN-0020-01 from the French ANR, the AXA Research Fund for the PhD grant of the first author and of the grant MTM2006-11391 from the Spanish MEC for the third author.

property is true over $\bar{k}$. An addition law $\mathfrak{p}$ is called complete if $S = \{\mathfrak{p}\}$ is complete.

Remark 1.2. We call attention to the change of notation towards the original paper [LR85] where original *complete* means here *geometrically complete*.

In [LR85], the authors give the interpretation of the possible addition laws in terms of the section of a certain line bundle.

Proposition 1.3 ([LR85, Lem.2.1])**.** *Let $p_1, p_2 : A \times A \to A$ be the projection maps on the first and second factor. There is an addition law (resp. a geometrically complete set of addition laws) of bidegree (μ, ν) on A with respect to the embedding in $\mathbb{P}^n$ determined by L if and only if*

$$H^0(A \times A, M) \neq 0$$

where $M = m^*L^{-1} \otimes p_1^*L^\mu \otimes p_2^*L^\nu$ (resp. the line bundle M is base point-free).

To a nonzero section $w \in H^0(A \times A, M)$ one associates an addition law defined on the open set $A \times A \setminus (w)_0$.

Note that asking the embedding to be symmetric is natural to have a good behaviour of the inversion as we see below. Another interest is to guarantee the existence of sections of bidegree (2,2) on M (see 2.2).

Proposition 1.4. *If A is embedded in a $\mathbb{P}^n$ by a very ample symmetric line bundle L, then the inversion is induced by a linear automorphism of $\mathbb{P}^n$.*

Proof. If L is a very ample symmetric line bundle, $H^0(A, L)$ can be generated by even or odd sections s_i [BL04]. Let ι th embedding associated to L. Hence

$$\iota(-z) = (s_0(-z) : \cdots : s_n(-z)) = (\pm s_0(z) : \cdots : \pm s_n(z)).$$

□

2. Geometrically complete set of addition laws

Theorem 2.1. *Let A an abelian variety and a symmetric embedding $\iota : A \hookrightarrow \mathbb{P}^n$. If S is a geometrically complete set of addition laws on A then $\#S > g$.*

Proof. Up to a projective change of coordinates, assume the image of $\mathcal{O}$ is $(0 : \cdots : 0 : 1)$.
Let $(U_j, \mathfrak{p}_j)_{1 \leq j \leq s}$ be a complete system of addition on A and a the injection $a : A \to A \times A$, $x \mapsto (x, -x)$. Consider the morphisms

$$A \xrightarrow{\ a\ } A \times A \overset{\mathfrak{p}_j}{\dashrightarrow} A$$

Note that for all x such that $a(x) \in U_j$, $\mathfrak{p}_j \circ a(x) = \iota(O)$ and, in particular, $\mathfrak{p}_n^{(j)} \circ a \neq 0$. Considering g hypersurfaces defined by the equations $\mathfrak{p}_n^{(j)} \circ a(\iota(x)) = 0$, $j = 1, \ldots, g$, the intersection between A and g hypersurfaces in $\mathbb{P}^n$ being non empty, the addition laws $\{\mathfrak{p}_1, \ldots, \mathfrak{p}_g\}$ are not defined at the points in this intersection. We hence get that $\#S = s > g$. □

We would like to have an upper bound on S as well. This is an easy consequence of the following results of [LR85].

Proposition 2.2 ([LR85, Prop.2.2,Prop.2.3])**.** *Let L be an ample line bundle on A. If $(\mu,\nu)=(2,2)$ then*

(1) if L is not symmetric then $H^0(A\times A,M)=\{0\}$.
(2) if L is symmetric, then $M=s^*L$ where $s:A\times A\to A$ is the difference map $(x,y)\mapsto x-y$. In particular M is base point-free and $h^0(M)=h^0(L)$.

If $(\mu,\nu)\neq(2,2)$ then $h^0(M)=h^0(L)^2(\mu\nu-\mu-\nu)^g$.

Proof. For $(\mu,\nu)=(2,2)$, only the equality $h^0(M)=h^0(L)$ has to be clarified. It is proven in [LR85, Prop.2.2] that $M|_{K(M)_0}$ is trivial. As L is ample his index in zero and, according to [Kem, Theorem 1(ii) p95], $H^0(A\times A,M)\cong H^0(A,L)$.
For the case $(\mu,\nu)\neq(2,2)$, one proves it like for the case $(\mu,\nu)=(2,3)$ of [LR85, Prop.2.3]. □

Corollary 2.3. *Let A be a principally polarized abelian variety. There exists an embedding of A for which a geometrically complete set of addition laws of order less or equal to 3^g exists.*

Proof. The line bundle L associated to the principal polarization is ample symmetric and has degree 1, so L^3 provides an embedding $A\hookrightarrow\mathbb{P}^{3^g}$. □

Remark 2.4. For $g=1$, our bounds show that there exists a geometrically complete set of addition laws of order 2 or 3. An explicit set of order 3 is given in [LR85, Sec.3] and in [LR87] in all characteristic and an explicit set of order 2 is given in [BL95].

3. Complete set of addition laws

Let L be a very ample symmetric line bundle defined by an effective rational divisor D on A. Let v be a section of $H^0(A,L)$ such that $(v)_0=D$ and let $w=s^*(v)$. As we have seen in Section 1, w is a section of $H^0(A\times A,M)$ with $M=m^*L^{-1}\otimes p_1^*L^2\otimes p_2^*L^2$ and an addition law (of bidegree $(2,2)$) is defined on the complementary of $(w)_0=s^*D$. Hence it is sufficient that D has no rational points for the group law to be complete. Note that it is also necessary since a rational point x on D gives the rational point $(x,0)$ on s^*D.

Theorem 3.1. *Let A an abelian variety over a non algebraically closed field and an embedding $\iota_0:A\hookrightarrow\mathbb{P}^n$. If there exists $\alpha\in\bar{k}$ whose degree d over k is greater than n then there exists an embedding $\iota:A\hookrightarrow\mathbb{P}^{n(2d-1)}$ and a complete addition law on $\iota(A)$.*

Proof. Let $\alpha_0,\ldots,\alpha_{d-1}$ the Galois conjugates of α in $\bar{k}$.
For $i=0,\ldots,d-1$, let H_i be the hyperplane

$$H_i: X_0+\alpha_iX_1+\ldots+\alpha_i^nX_n=0.$$

Since $d>n$, $\{1,\alpha_i,\ldots,\alpha_i^n\}$ are linearly independent for every i over k and $H_i(k)$ is empty. Now $\sum H_i$ is a divisor over k, let $D_0=\iota_0^*(\sum H_i)$ and $D=D_0+(-1)^*D_0$. D is a symmetric rational divisor on A without rational points. Denoting L_0 the line bundle associated to ι_0, the line bundle $L=\mathcal{L}(D)$ is linearly equivalent to L_0^{2d}, so L is very ample and provides an embedding $A\hookrightarrow\mathbb{P}^{n(2d-1)}$ with a complete addition law. □

We want to minimize the dimension of the projective space in which embed the variety, considering a finite map $\varphi : A \to \mathbb{P}^n$. Indeed $\varphi^*\mathcal{O}_{\mathbb{P}^n}(1)$ is ample (and conversely). Moreover, such a map exists for $n = g$ (got from Noether's normalisation theorem). Hence this proves the following theorem:

Theorem 3.2. *Let A an abelian variety of dimension g over a non algebraically closed field. If there exists in $\bar{k}$ an element of degree d greater than g then there exists an embedding ι of A in $\mathbb{P}^{g(2d-1)}$ and an arithemetically complete addition law on $\iota(A)$.*

Over finite fields, every abelian variety of dimension g can be embedded in a projective space of dimension $g(2g+1)$ with a complete addition law.

4. Example in genus one and two

In the previous section, the complete addition law is proved to exist but for a large embedding of the abelian variety. When A over a finite field has dimension 1 or 2, we make explicit the construction of a desired divisor.

4.1. **Elliptic curves.** First, let us consider a particular case.

Theorem 4.1. *Let $E : y^2 = f(x)$ be an elliptic curve embedded in $\mathbb{P}^2$. If f is irreducible there exists a complete addition law on $E \subset \mathbb{P}^2$.*

Proof. Consider the zero divisor of the section y on E. It is a very ample symmetric rational divisor without rational points which provides a complete addition law on $E \subset \mathbb{P}^2$. □

Theorem 4.2. *If $E/\mathbb{F}_q$ is an elliptic curve with a point whose field of definition is $\mathbb{F}_{q^2}$, then there is an embedding on E in $\mathbb{P}^3$ on which one can define a complete addition law.*

Proof. Let $P \in E(\overline{k})$ as in the statement. Then, $D := D_0 + (-1)^* D_0$, where $D_0 = P + P^\sigma$, is the divisor we want to. □

Remark 4.3. Previously we used the fact that D symmetric divisor implies $\mathcal{L}(D)$ symmetric line bundle. But the converse is not true: if $D \sim D'$ with D not symmetric and D' symmetric divisors then $\mathcal{L}(D) \sim \mathcal{L}(D')$ are symmetric line bundles. In particular, this is the case if the points of the Galois orbit lie on a line. That's what we use in the theorem 4.4.

Theorem 4.4. *Let $E/\mathbb{F}_q, q \geq 5$, an elliptic curve defined by $y^2 = f(x)$. There exists an embedding on E in $\mathbb{P}^2$ on which we can define a complete addition law.*

Proof. Consider the group homomorphism $N : E(\mathbb{F}_{q^3}) \to E(\mathbb{F}_q)$, $P \mapsto P + P^\sigma + P^{\sigma^2}$, where σ denotes here the Frobenius. $ker(N)$ restricts on $E(\mathbb{F}_q)$ to the $\mathbb{F}_q$-rational 3-torsion points on E. Now $\#\ker(N) \geq \frac{\#E(\mathbb{F}_{q^3})}{\#E(\mathbb{F}_q)} \geq \frac{q^3+1-2\sqrt{q^3}}{q+1+2\sqrt{q}} > 9$. Hence there exists $P_0 \in E(\mathbb{F}_{q^3})$ non $\mathbb{F}_q$-rational whose Galois orbit provides the aligned points we searched for. □

Remark 4.5. If the trace of E is positive the points in the Galois orbit can be chosen on a horizontal line.

4.2. **Genus 2 curves.** Let $C/\mathbb{F}_q, q \geq 5$, be a genus 2 curve. Assume C has a model over k of the form $y^2 = f(x)$ with f of degree 5. Let us denote by $\overline{\ }$ the canonical involution on C, *i.e.* if $P = (x, y)$, $\overline{P} = (x, -y)$ and $\Theta = W_1$. Θ is an ample symmetric rational divisor on $\mathrm{Jac}\, C$ which defines the canonical principal polarization on $\mathrm{Jac}\, C$. For any $z \in \mathrm{Jac}(C)(\bar{k})$, we denote $\Theta_z = \Theta + z$ the translation of Θ by z.

Lemma 4.6 ([Mum75, p.272],[Deb86])**.** *If $z \sim P - Q \in (\mathrm{Jac}\, C)(\bar{k})$ then*

$$\Theta \cdot \Theta_z = \{(P) - (P_\infty)\} \cup \left\{(\overline{Q}) - (P_\infty)\right\}.$$

Lemma 4.7. *There exists $x_0 \in \mathbb{F}_{q^2}$ such that $f(x_0)$ is not a square in $\mathbb{F}_{q^2}$.*

Proof. If it were not the case, there would be at least $2(q^2 - q) + \#C(\mathbb{F}_q)$ $\mathbb{F}_{q^2}-$rational points on C which contradicts the Hasse-Weil bounds. □

Theorem 4.8. *Let C be a genus two curve as above. There exists an embedding ι of $\mathrm{Jac}\, C$ in $\mathbb{P}^{15}$ such that there is a complete group law on $\iota(\mathrm{Jac}\, C)$.*

Proof. Let x_0 be as in Lemma 4.7, let $P = (x_0, y_0) \in C(\mathbb{F}_{q^4})$ where y_0 is a choice of square root of $f(x_0)$. The Galois orbit of P is $\{P, P^\sigma, \overline{P}, \overline{P}^\sigma\}$ where σ is the Frobenius.
Now we construct points $\alpha_0, \alpha_1, \alpha_2, \alpha_3$ such that $\Theta_{\alpha_0} \cdot \Theta_{\alpha_1} \cdot \Theta_{\alpha_2} \cdot \Theta_{\alpha_3} = \emptyset$. We take $\alpha_0 = P + P^\sigma, \alpha_1 = P^\sigma + \overline{P}, \alpha_2 = \overline{P} + \overline{P}^\sigma, \alpha_3 = \overline{P}^\sigma + P$ in $\mathrm{Jac}\, C$. They verify the relations

$$\Theta_{\alpha_0} \cdot \Theta_{\alpha_1} = t^*_{\alpha_0}(\Theta \cdot \Theta_{\overline{P}-P}) = t^*_{\alpha_0}(\overline{P}),$$

$$\Theta_{\alpha_0} \cdot \Theta_{\alpha_3} = t^*_{\alpha_0}(\Theta \cdot \Theta_{\overline{P}^\sigma - P^\sigma}) = t^*_{\alpha_0}(\overline{P}^\sigma).$$

So the divisor $D = \sum \Theta_{\alpha_i}$ is ample, symmetric, rational by construction, and without rational points according to relations above. □

References

[BL95] W. Bosma and H.W.jun. Lenstra. Complete systems of two addition laws for elliptic curves. *J. Number Theory*, 53(2):229–240, 1995.

[BL04] Christina Birkenhake and Herbert Lange. *Complex abelian varieties*, volume 302 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, second edition, 2004.

[BL07] Daniel J. Bernstein and Tanja Lange. Faster addition and doubling on elliptic curves. Kurosawa, Kaoru (ed.), Advances in cryptology – ASIACRYPT 2007. 13th international conference on the theory and application of cryptology and information security, Kuching, Malaysia, December 2-6, 2007. Proceedings. Berlin: Springer. Lecture Notes in Computer Science 4833, 29-50 (2007)., 2007.

[Deb86] O. Debarre. Sur la démonstration de A. Weil du théorème de Torelli pour les courbes. *Compositio Math.*, 58(1):3–11, 1986.

[Kem] G. Kempf. Appendix to *"Varieties defined by quadratic equations"* by D. Mumford. In: Questions on algebraic varieties. CIME.

[LR85] H. Lange and W. Ruppert. Complete systems of addition laws on abelian varieties. *Invent. Math.*, 79:603–610, 1985.

[LR87] H. Lange and W. Ruppert. Addition laws on elliptic curves in arbitrary characteristics. *J. Algebra*, 107(1):106–116, 1987.

[Mum75] David Mumford. *Curves and their Jacobians*. The University of Michigan Press, Ann Arbor, Mich., 1975.

Elliptic Curve Cryptography on Standard Curves Using the Edwards Addition Law

Vincent Verneuil[1,2]

[1] Inside Contactless,
41, parc du Golf, 13 856 Aix-en-Provence cedex 3, France
vverneuil@insidefr.com
[2] Institut de Mathématiques de Bordeaux,
351, cours de la Libération, 33 405 Talence cedex, France

Abstract. In this paper we address the issue of performing scalar multiplication over standard curves using the Edwards unified addition law. In particular we detail the required mapping for curves having cofactor 1 and discuss the efficiency of this method for scalar multiplication.

Keywords: Elliptic Curve Cryptography, Scalar Multiplication, Edwards Curves, Standard Curves, Side-Channel Countermeasures.

1 Introduction

Scalar multiplication on elliptic curves is the main operation of many cryptographic protocols using elliptic curves such as ECDSA, ECDH, EC ElGamal, etc. Indeed it is often the most time consuming operation of these protocols and many studies have focused on speeding-up this operation. Moreover it may manipulate secret keys and thus be threatened by the side-channel analysis [16, 17] when implemented on embedded devices such as smart cards.

Short Weierstrass equation elliptic curves $y^2 = x^3 + ax + b$ over a large characteristic prime field $\mathbb{F}_p$ require to distinguish points additions (i.e. $P_1 + P_2$, $P_1 \neq P_2$) from point doublings ($P_1 + P_1$) which may yield to side channel leakages if no particular countermeasure is applied. Homogeneous and Jacobian projective coordinates are generally employed to avoid the expensive inversions in $\mathbb{F}_p$ of the affine group law [6] but still have different formulas for points additions and doublings.

Counterfeiting side-channel analysis may be achieved through several ways. Firstly, regular algorithms may be used, these algorithms either hide sensitive conditional branching by adding dummy point operations [7] or have a naturally regular structure [13–15]. Another countermeasure consists in using a unified addition law when available, which performs point additions and doublings using the same sequence of field operations and thus cannot be distinguished from one another[3]. Finally, the atomicity principle may be used to protect any algorithm by

[3] Such formulas has been proposed by Brier and Joye in [4] for affine and homogeneous projective coordinates but are very expensive.

expressing formulas using sequences of a single field operation pattern. Its exact cost varies for each algorithm but is generally reasonable [5, 18, 10].

More recently Edwards presented in [8] a unified addition law for elliptic curves of equation $x^2+y^2 = c^2(1+x^2y^2)$. Then Bernstein and Lange generalized this law in [3] to the larger group of curves $x^2+y^2 = 1+dx^2y^2$ which covers about 1/4 of all the elliptic curves over non-binary finite fields. Finally, Bernstein et al. presented in [2] the twisted Edwards curves of equation $ax^2 + y^2 = 1 + dx^2y^2$ with the corresponding unified addition law. This equation covers even more curves than the previous one and has a comparable addition cost. Moreover when the unified addition law is not necessary, Edwards and twisted Edwards curves provide a very fast doubling formula. Those curves are thus of the utmost interest for scalar multiplication implementors.

On the other hand, the implementation of elliptic curve cryptography in industrial products has required the development of standards. There are currently two main standards: NIST FIPS 186-3 [9] and the German Brainpool standard [19]. Indeed, other standards such as ANSI X9.62 [1], ISO 15946 [12], IEEE P1363 [11] and SECG [20] mainly provide pointers to NIST curves.

Fast and secure implementations of the elliptic curve scalar multiplication for industrial products should thus be optimized for NIST and/or Brainpool curves. There are 5 NIST curves over $\mathbb{F}_p$ with parameters of sizes 192, 224, 256, 384, and 521 and 7 Brainpool curves over $\mathbb{F}_p$ with parameters of sizes 160, 192, 224, 256, 320, 384, and 512. All have cofactor 1.

We study in this paper the explicit mappings to perform scalar multiplications of points lying on standard curves over $\mathbb{F}_p$ using Edwards and twisted Edwards addition laws. These mappings are described in [3, 2] but i) we thought that descriptions were not explicit enough for implementors desiring to use them, and ii) we did not found in the literature if they effectively lead to fast implementations on standard curves.

2 Edwards Curves

In [3] Bernstein and Lange mention that every elliptic curve having a point of order 4 has some quadratic twist birationally equivalent to an Edwards curve. Unfortunately all NIST and Brainpool curves have cofactor 1, nonetheless we can consider these curves over an extension field $\mathbb{F}_{p^e}$ in which they have a point of order 4.

Assume that a point $P = (x_1, y_1)$ lie on an elliptic curve

$$E_1 : y^2 = x^3 + ax + b$$

with $p > 3$ and $4a^3 + 27b^2 \in \mathbb{F}_p^*$. Assume also that $E_1(\mathbb{F}_p)$ has prime cardinal $n > 2$. We aim at computing the scalar multiplication $Q = k \cdot P$, $k \in \mathbb{Z}$.

The roots of the polynomial $\psi_4^* = \psi_4/\psi_2 = x^6 + 5ax^4 + 20bx^3 - 5a^2x^2 - 4abx - a^3 - 8b^2$ give the x-coordinates of the points of order 4 of E_1. Since this polynomial has no root over $\mathbb{F}_p$ we need to consider an extension field $\mathbb{F}_{p^e}$ to find such roots. The degree of ψ_4^* ensures that $e \leq 6$. More precisely if ψ_4^* is irreducible over $\mathbb{F}_p$ then $e = 6$. This is the case for all NIST and Brainpool curves, except NIST P-224 for which ψ_4^* splits in two polynomials of degree 3.

Consider an irreducible polynomial $A(t)$ of degree e and r a root of ψ_4^* in $\mathbb{F}_p[t]/A(t)$. Notice that ψ_4^* is a suitable choice for A in most cases, which gives also $r = \bar{t}$. Then compute s a square root of $r^3 + ar + b$. If $s \notin \mathbb{F}_{p^e}$ it is necessary to consider an extension field of degree $2e$ instead of e.

Fact 1 *The minimal extension field to find a point of order 4 (r, s) is $\mathbb{F}_{p^6}$ for all NIST and Brainpool curves over $\mathbb{F}_p$.*

Let $(r_2, 0)$ denote $2(r, s)$ from the elliptic curve points group law and having order 2. We shall translate the curve E_1 to ensure that the point $(0, 0)$ belongs to it. This is done by considering the isomorphic curve obtained with the map $(x, y) \mapsto (x - r_2, y)$. This gives

$$E_2 : y^2 = x^3 + \lambda_1 x^2 + \lambda_2 x$$

on which lie $(r_1, s_1) = (r - r_2, s)$ the point of order 4, and $(x_2, y_2) = (x_1 - r_2, y_1)$ the point corresponding to P.

Let compute $d = 1 - 4{r_1}^3/{s_1}^2$ and $\lambda_3 = r_1/(1 - d)$. Then consider E_3 the following quadratic twist of E_2 :

$$E_3 : \lambda_3 y^2 = x^3 + \lambda_1 x^2 + \lambda_2 x$$

If λ_3 is a square in $\mathbb{F}_{p^6}$ then E_3 is isomorphic to E_2 and thus to E_1. This is the case of all NIST and Brainpool curves. The base point on E_3 becomes $(x_3, y_3) = (x_2, y_2/\sqrt{\lambda_3})$. Notice that $\lambda_1 = 2r_1(1 + d)/(1 - d)$ and $\lambda_2 = r_1^2$.

Now we consider the elliptic curve obtained by the substitution $(x, y) \mapsto (x/r_1, y/r_1)$. Since $r_1 \neq 0$ – otherwise (r_1, s_1) would be of order 2 – E_3 is isomorphic to the so obtained elliptic curve :

$$E_4 : \frac{1}{1-d} y^2 = x^3 + \frac{2(1+d)}{1-d} x^2 + x$$

The base point thus becomes $(x_4, y_4) = (x_3/r_1, y_3/r_1)$.

Finally we consider the Edwards elliptic curve E_5:

$$E_5 : x^2 + y^2 = 1 + dx^2y^2$$

Bernstein and Lange have shown that E_4 – and therefore E_1 – is birationally equivalent to E_5. The point corresponding to P is given by $(x_5, y_5) = (2x_4/y_4, (x_4 - 1)/(x_4 + 1))$.

It is now possible to compute $(u_5, v_5) = k \cdot (x_5, y_5)$ using the Edwards addition formula in $\mathbb{F}_{p^6}$. Notice that d is a square for all NIST and Brainpool curves, thus the Edwards addition law is not *complete* – i.e. some special inputs may require particular handling.

The corresponding results on E_4,E_3 and E_2 are respectively obtained as $(u_4, v_4) = ((1+v_5)/(1-v5), 2(1+v_5)/u_5(1-v5))$, $(u_3, v_3) = (u_4 r_1, v_4 r_1)$, $(u_2, v_2) = (u_3, v_3\sqrt{\lambda_3}))$. Finally $Q = (u_2 + r_2, v_2)$.

This method allowed to successfully perform scalar multiplications on the 5 NIST curves and the 7 Brainpool curves defined over $\mathbb{F}_p$ using the Edwards addition law. However it implies handling coordinates in $\mathbb{F}_{p^6}$ during the scalar multiplication and thus increases operations cost by a factor at least 6 – which is very optimistic. On the other hand, the speed-up provided by the Edwards formula depends on the exact context but cannot reasonably exceed 50%. Consequently the cost of handling coordinates in $\mathbb{F}_{p^6}$ clearly overcomes the gain of using the Edwards addition law.

3 Twisted Edwards curves

In [2] Bernstein et al. mention that every elliptic curve having three points of order 2 is isogenous to a twisted Edwards curve. We thus experiment to find such points in an extension of $\mathbb{F}_p$ in order to perform the scalar multiplication on a twisted Edwards curve.

Let E_1, P and Q be defined as previously. Points of order 2 have by definition their y-coordinate to 0. We thus need to find r_0, r_1, r_2 the roots of the polynomial $x^3 + ax + b$.The three points of order 2 are then $(r_0, 0)$, $(r_1, 0)$ and $(r_2, 0)$.

Fact 2 *The minimal extension field to find three points of order 2 is $\mathbb{F}_{p^3}$ for all NIST and Brainpool curves over $\mathbb{F}_p$.*

As previously we shall translate the curve E_1 to ensure that the point of order 2 $(0, 0)$ belongs to it. Again, this is done by considering the isomorphic curve obtained by applying $(x, y) \mapsto (x - r_0, y)$. This yields

$$E_2 : y^2 = x^3 - (r_1 + r_2)x^2 + (r_1 r_2)x$$

on which lies $(x_2, y_2) = (x_1 - r_0, y_1)$ the point corresponding to P.

Then we consider the isogenous curve E_3 given by :

$$E_3 : y^2 = x^3 + 2(r_1 + r_2)x^2 + (r_1 - r_2)^2 x$$

on which lies $(x_3, y_3) = ({y_2}^2/{x_2}^2, y_2(r_1 r_2 - {x_2}^2)/{x_2}^2)$.

Then we consider the Montgomery curve E_4, isomorphic to E_3, given by :

$$E_4 : \frac{1}{r_1 - r_2} y^2 = x^3 + \frac{2(r_1 + r_2)}{r_1 - r_2} x^2 + x$$

on which lies $(x_4, y_4) = (x_3/(r_1 - r_2), y_3/(r_1 - r_2))$.

Bernstein et al. have shown that every Montgomery curve is birationally equivalent to a twisted Edwards curve, in particular E_4 is birationally equivalent to :

$$E_5 : 4r_1x^2 + y^2 = 1 + 4r_2x^2y^2$$

on which lies $(x_5, y_5) = (x_4/y_4, (x_4 - 1)/(x_4 + 1))$ the point corresponding to P.

The scalar multiplication on E_5 becomes $(u_5, v_5) = (k/2 \mod n'/4) \cdot (x_5, v_5)$ with $n' = |E_1(\mathbb{F}_{p^3})|$. From the Weil formula we have $n' = p^3 + 1 - t(t^2 - 3p)$ where $t = p + 1 - n$.

Remark 1. Notice that $k/2 \mod n'/4$ has the size of k if k is even but is 3 times longer if k is odd. However this is not a real issue since one may only consider even scalars or compute $(k-1) \cdot P$ on the twisted Edwards curve and add P on the Weierstrass curve.

Notice also that the twisted Edwards addition law is complete if r_1 is a square in $\mathbb{F}_{p^3}$ but not r_2. It means that among the three roots of $x^3 + ax + b$ at least one has to be a square and at least one a non-square. This property was always fulfilled except for NIST P-224.

The corresponding results on E_4,E_3 and E_2 can be respectively recovered as $(u_4, v_4) = ((1+v_5)/(1-v5), (1+v_5)/u_5(1-v5))$, $(u_3, v_3) = (u_4(r_1-r_2), v_4(r_1-r_2))$, $(u_2, v_2) = ({v_3}^2/4{u_3}^2, v_3((r_1 - r_2)^2 - {u_3}^2)/8{u_3}^2)$. Finally $Q = (u_2 + r_0, v_2)$.

We successfully performed scalar multiplications on the 12 NIST and Brainpool curves over $\mathbb{F}_p$ using the twisted Edwards addition law. However, it implies computations in $\mathbb{F}_{p^3}$, whose cost is lower than previously but still clearly overcomes the gain of using the twisted Edwards formula.

4 Conclusion

We pointed out in this article that standard elliptic curves over $\mathbb{F}_p$ require to perform scalar multiplication over an extension field of degree 6 to apply the Edwards addition law or degree 3 for the twisted Edwards addition law. However such an extension degree is not affordable with regard to the speed-up provided by Edwards and twisted Edwards formulas. This is especially the case in the embedded devices context which is leading in the industrial elliptic curves cryptography development due to its memory and power constraints.

In conclusion we guess that, as long as there will be no standard elliptic curves over $\mathbb{F}_p$ with cofactor 4, speeding-up the scalar multiplication of Weierstrass curves will remain a topic of interest in the industrial context.

References

1. ANSI X9.62–2005. *Public Key Cryptography for The Financial Service Industry : The Elliptic Curve Digital Signature Algorithm (ECDSA)*. American National Standards Institute, November 2005.
2. Daniel J. Bernstein, Peter Birkner, Marc Joye, Tanja Lange, and Christiane Peters. Twisted edwards curves. In Serge Vaudenay, editor, *AFRICACRYPT*, volume 5023 of *Lecture Notes in Computer Science*, pages 389–405. Springer, 2008.
3. Daniel J. Bernstein and Tanja Lange. Faster addition and doubling on elliptic curves. Cryptology ePrint Archive, Report 2007/286, 2007. `http://eprint.iacr.org/`.
4. O. Billet and M. Joye. The Jacobi Model of an Elliptic Curve and Side-Channel Analysis. Cryptology ePrint Archive, Report 2002/125, 2002.
5. Benoît Chevallier-Mames, Mathieu Ciet, and Marc Joye. Low-cost Solutions for Preventing Simple Side-Channel Analysis: Side-Channel Atomicity. *IEEE Transactions on Computers*, 53(6):760–768, 2004.
6. Henri Cohen, Gerhard Frey, et al. *Handbook of Elliptic and Hyperelliptic Curve Cryptography*. Chapman & Hall/CRC, 2005.
7. J.-S. Coron. Resistance against Differential Power Analysis for Elliptic Curve Cryptosystems. In Ç.K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES '99*, volume 1717 of *Lecture Notes in Computer Science*, pages 292–302. Springer, 1999.
8. Harold M. Edwards. A normal form for elliptic curves. *Bulletin of the American Mathematical Society*, 44:393–422, 2007.
9. FIPS PUB 186-3. *Digital Signature Standard*. National Institute of Standards and Technology, March 2006. Draft.
10. C. Giraud and V. Verneuil. Atomicity Improvement for Elliptic Curve Scalar Multiplication. In Julien Iguchi-Cartigny, Dieter Gollmann, and Jean-Louis Lanet, editors, *CARDIS 2010*, volume 6035 of *Lecture Notes in Computer Science*, pages 80–101. Springer, 2010.
11. IEEE Std 1363a-2004. *IEEE Standard Specifications for Public-Key Cryptography – Amendment 1: Additional Techniques*. IEEE Computer Society, September 2004.
12. ISO/IEC 15946-1. *Information Technology – Security Techniques – Cryptographic techniques Based on Elliptic Curves – Part 1: General*, 2002.
13. M. Joye and S.-M. Yen. The Montgomery Powering Ladder. In B.S. Kaliski Jr., Ç.K. Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*, pages 291–302. Springer, 2002.
14. Marc Joye. Highly regular right-to-left algorithms for scalar multiplication. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems – CHES 2007*, volume 4727 of *Lecture Notes in Computer Science*, pages 135–147. Springer, 2007.
15. Marc Joye. Highly regular m-ary powering ladders. In *16th Annual Workshop on Selected Areas in Cryptography (SAC 2009)*, August 2009.
16. P. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In N. Koblitz, editor, *Advances in Cryptology – CRYPTO '96*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer, 1996.
17. P. Kocher, J. Jaffe, and B. Jun. Introduction to Differential Power Analysis and Related Attacks. Technical report, Cryptography Research Inc., 1998.
18. Patrick Longa. Accelerating the Scalar Multiplication on Elliptic Curve Cryptosystems over Prime Fields. Master's thesis, School of Information Technology and Engineering, University of Ottawa, Canada, 2007.
19. M. Lochter and J. Merkle. *Elliptic Curve Cryptography Brainpool Standard Curves and Curve Generation*. ECC Brainpool, March 2010. RFC 5639 `http://tools.ietf.org/html/rfc5639`.
20. Standards for Efficient Cryptography Group (SECG). *SEC 2 : Recommended Elliptic Curve Domain Parameters*. Certicom Research, 2000.

A Numerical Example

We provide in the following two examples of scalar multiplication on the NIST P-192 curve using Edwards and twisted Edwards formula with the following base point and scalar :

$P = (602046282375688656758213480587526111916698976636884684818, 174050332293622031404857552280219410364023488927386650641)$

$k = 764793464287780488015619147195181618135290770882198355914$

A.1 Edwards Mapping

$A(t) = t^6+6277101735386680763835789423207666416083908700390324961264t^4+5163398772472111007955535234149537078279480026049155003234 7t^3+6277101735386680763835789423207666416083908700390324961234t^2+4353459610560602757540369289538755752893661896372995011664t+556572562867275638562069244750930080159497890832495480428 7$

$(r, s) = (t, 123288533732987672110865078112302685534011569685174821871 6t^5+4932477686389072668016991071412418223708629683233024004 48t^4+3738804120027750459147372311576249545169533275370852887435t^3+2985549502162143631678647934517474989370195960933744006916t^2+345127056729561339752586048651174548525442418081292243533 5t+3675696037813148084070050989945142303640407213286917992 43)$

$d = 342033898252281290806442380795827393029436030539879436831 2t^5+7096293228683146697806937966593386891854953651283738782 41t^4+6276253110032103724812950305489023427888477883448250281073t^3+8826093320057595705768052216495766672348376319910524707 06t^2+3744415841762234276757594919031605484022217101057518417 395t+2544405952845063473758844807435704955368111104330496027457$

$(x_3, y_3) = (6042801227311487248891532886824760140143018265622682125527t^5+4999440895681090877477447164012672236561385433529378727 18t^4+3280207113052709209219595093606878631724660867469997005 28t^3+4443428579974747537705574162279619493381428119369885465088t^2+2669751631708896080503479091367749654876457936583305717 70t+4908438192344199888037793697631965658617568918402742032 92, 5448940144146238690395392987575249102457906876388411728494t^5+6253498282324623590425802397073442417403878977744368969 42t^4+5371398455506405768885608633602548599109111181777341154649t^3+1062401218587368023903413873640100662769657989816371162819t^2+3592207122684893417023860007954255099937160514158788855726t+5238832436750337898023115609617977527979118313938908189912)$

$(x_5, y_5) = (2411620920747559806654530833406361680285397433517820138225t^5+2178194511121855954145423061971735686308488106140904616061t^4+3035145219164513855423694820214289866360537862491721611532t^3+1575841831888966463076829213240133390198510497312856768131t^2+3433775095752019020811384990062716869320098942070497063477t+6055993466415172059541564274034851443862312849531737220508, 9344763454052934094154325193271636227285836422065916841 20t^5+4901588814074284966087421037449470933422314692300849183648t^4+1560145556924236026462482284446476973220445880317716383270t^3+5934114755194262414936404990219501543462599522373770705893t^2+3433266300379969589597679547146430322784102918299141574256t+3714568610352283246178132491546257131387479417673263847505)$

$(u_5, v_5) = (3338563021775073422109629931257855172669205872938547929456t^5+1806851729074167634573774551481175278099574410842660666947t^4+4673054178789749506142787495338016851836251851521115922164t^3+1270682325639584583001564867621533011954970211690961310882t^2+5096188360313330717937411142038755611374198189517746443465t+4793145063747189037226039589343742100627088263323126198918, 1815426887389087292654708598859930356510180535301706364309t^5+3857849846656926882814158216759927508517260009708024804663t^4+4318015937391865862699834399829172507629462535198720840343t^3+5074827600207293986247307542569191531430258921898515155881t^2+5479129751516854254694027158300638119981336424525611064498t+1024342340679804401361714485156265354582960566628481184047)$

$(u_3, v_3) = (6042801227311487248891532886824760140143018265622682125527t^5+4999440895681090877477447164012672236561385433529378727 18t^4+3280207113052709209219595093606878631724660867469997005 28t^3+4443428579974747537705574162279619493381428119369885465088t^2+2669751631708896080503479091367749654876457936583305717 70t+5870565252401868281583577845557231684877110298695$

$7855839828, 503857869404974462517173507902893313084295202651243908284 8t^5 + 483228421729943134577849006818371435796536536841361272353 8t^4 + 4954764612336757239071117165647861458182358460617028160 45t^3 + 338781382644238461878506715481915781560490964047990841113 9t^2 + 4817448041364297834094169058023603474411555045759680740322t + 1760621086073593505120109802912135356157779095265669943544)$

$Q = (59817677155431369495380119563815612309320523834944663213 54, 19694698228932324251488692553304937635125530119952253131 32)$

A.2 Twisted Edwards Mapping

$A(t) = t^3 + 2t + 1$

$r_0 = 60831036468885443179070946508185830119109217414945422567 75 * t^2 + 436845175979817440468957250799300878860915776776543903275 6 * t + 60184376173891655025975297266888885438532594218626146 88607$

$r_1 = 45568658286979911428214980751179633067434880479939320938 81 * t^2 + 14751836579137902284369262564313990772339292305849537748 9 * t + 60758211049306548570953307668239510756579840639919094 58508$

$r_2 = 19142339951868260669429861204787865135134076112921755719 02 * t^2 + 17611316097971273363025242895715177197513580095663905510 34 * t + 45994474845354116797871835290249321265657391492612 5775443$

$(x_3, y_3) = (15677742798915387947985843459700308825927105129653906851 02t^2 + 624122645127030662417816848448246703918296188949972969638 2t + 33742873642907774849777431734834510632722998141038049813 33, 30837045630525735949199682036143197013289916097630302518 14t^2 + 933408426399387677114666626873122855519733694475392049 152t + 41641848990057491152384814723428239570468893099159579899 81)$

$(x_5, y_5) = (33262451083402663256988362747517432643771010021932106064 69t^2 + 637178476784103578925455693539414920658734859435971111 57t + 50709225157952492460913218918411024830077370813431391043 72, 84641810906354508041587914501320230545206796582530079906 8t^2 + 32127218550829972703606549224623076520739041381447689639 55t + 27864447807438396297671335736818287282742720653699672978 85)$

$n' = 24733040147310453406050252101964719003295083292832501924229687865064402806584051392906234240119484788372433875038556510890373644895961069259546024013600118489382431286334160 4$

$(u_5, v_5) = (58964859110855725886067674976462723276502347648364697018 47t^2 + 178367726290130774410094024365932736530174220227985137 4217t + 10478271487255780757628793650831115649760273891637196122 6, 21776538101634411610828988053875689297781744526844009620 65t^2 + 65489481594431246334515243402296698388041872238262008995 5t + 14272798214772469130689553606915709822280433354740603502 76)$

$(u_3, v_3) = (39382591455911263356635803170582900561732786663111766201 51t^2 + 571207562273103144353980929107696156132006851222677284896 1t + 46130716527428634112124012374247721371865543934290704502 96, 51430882855314985966375391532967873106889452049250767435 98t^2 + 394673150414784673832602976087836811786466655196533920844 1t + 57324769175843665733650647971518375658434791270246847247 03)$

$Q = (59817677155431369495380119563815612309320523834944663213 54, 19694698228932324251488692553304937635125530119952253131 32)$

About the security of pairing based cryptography

Nadia EL MRABET

Université de Caen - GREYC
nadia.el_mrabet@info.unicaen.fr

Abstract. Since 2001, the interest of cryptographers in pairing based cryptography is very high. Pairings allow construction of original protocols and simplification of existing protocols. Several system of coordinates or elliptic curves are used in pairing based cryptography, it seems fair to wonder which system of coordinates and elliptic curves assure the best security.
keyword: pairing based cryptography, side channel attacks, security.

1 Introduction

Pairings are mathematical tools which were introduced in cryptography during the 90'. At the beginning, it was used to build attacks on cryptosystems, transferring the discrete logarithm problem on elliptic curves, to a discrete logarithm problem on finite fields. The first was the MOV [10] attack in 1993.

Since 2001, pairings are used in a constructive way, simplifying existing protocols [8], [3] or creating new ones [2]. One of the most interesting could be the identity based cryptography. In an identity based protocol the public key of a user is his identity and a trusted authority construct the private key related to his identity. When using pairings in identity based cryptography, the problem of side channel attacks must be taken into account. Indeed, there is a secret to protect: the private key of users.

In [12, 14, 6, 5], the authors study security of pairing based cryptography overs Weirstrass elliptic curves. It has been proved that without protection, the pairing based cryptography in Weierstrass coordinates is vulnerable to Differential Power Attack (DPA), Correlation and fault attacks, all of this attacks are side channel attacks.

Recently, the Edwards elliptic curves were used to compute pairings [1]. Edwards curves became interesting for elliptic curve cryptography when it was proved by Bernstein and Lange in that they provide addition and doubling formulas faster than all addition formulas known at that time. The advantage of Edwards coordinates is that the addition law can be complete (i.e. the formulas for adding or doubling two points are the same) and thus the exponentiation in Edwards coordinates is naturally protected against side channel attacks. Unfortunately, pairing in Edwards coordinates are also vulnerable to side channel attacks.

Considering, different system of coordinates, for different elliptic curves there is different countermeasure possibles. We are going to investigate which one is more secure and less expensive to implement. Indeed, a secure algorithm is not interesting if its implementation is too expensive in time and place.

The outline of the article is the following, first we recall what is the pairing based cryptography in Section 2. Then in Section 3 we describe the differentiel power analysis attack in pairing

based cryptography and the possible countermeasure, and in Section 4 toward fault attacks. We conclude in Section 6.

2 pairing based cryptography

2.1 A short introduction to pairings

In this section we give a brief overview of the definitions of pairings on elliptic curves and of Miller's algorithm [11] used in pairing computation. Let q be a prime power not divisible by 2, E an elliptic curve over $\mathbb{F}_q$ and r a prime factor of $\#(E(\mathbb{F}_q))$. Suppose r^2 does not divide $\#(E(\mathbb{F}_q))$ and let k be the *embedding degree* with respect to r, i.e. the smallest integer such that r divides $q^k - 1$. We denote $\mathcal{O}$ the point at infinity of the elliptic curve.

Definition 1. *A pairing is a bilinear and non degenerate function:*

$$\begin{aligned} e : \mathbb{G}_1 \times \mathbb{G}_2 &\to \mathbb{G}_3 \\ (P, Q) &\to e(P, Q) \end{aligned}$$

where $\mathbb{G}_1$ and $\mathbb{G}_2$ are subgroups of order r on the elliptic curve and $\mathbb{G}_3$ is generally μ_r, the subgroup of the r-th roots of unity in $\mathbb{F}_{q^k}$. In general we take $\mathbb{G}_1 = E(\mathbb{F}_q)[r]$ and $\mathbb{G}_2 \subset E(\mathbb{F}_{q^k})[r]$, where we denote by $E(K)[r]$ the subgroup of K-rational points of order r of the elliptic curve E. We also denote $E[r]$ the subgroup of points of order r defined over the algebraic closure of $\mathbb{F}_q$. Let $P \in \mathbb{G}_1$, $Q \in \mathbb{G}_2$. The goal of Miller's algorithm is to construct a rational function $f_{s,P}$ associated to the point P and to some integer s and to evaluate this function at the point Q (in fact at a divisor associated to this point). The function $f_{s,P}$ is such that the divisor associated to it is:

$$\mathrm{div}(f_{s,P}) = s(P) - (sP) - (s-1)(\mathcal{O}).$$

Suppose that we want to compute the sum of iP and jP. Take h_1 the equation of the line going through iP and jP and h_2 the equation of the vertical line through $(i+j)P$. Miller's idea is to make use of the following relation

$$f_{i+j,P} = f_{i,P} f_{j,P} \frac{h_1}{h_2}, \tag{1}$$

in order to compute $f_{s,P}$ iteratively. Moreover, Miller's algorithm uses the double-and-add method to compute $f_{s,P}$ in $\log_2(s)$ operations [11].

The reduced Tate pairing

In order to illustrate the notion of pairings, we give here the definition of the reduced Tate pairing. The reduced Tate pairing, denoted $\widehat{e}_{Tate}$, is defined by:

$$\begin{aligned} &\mathbb{G}_1 \times \mathbb{G}_2 \mapsto \mathbb{G}_3 \\ &(P, Q) \mapsto \widehat{e}_{Tate}(P, Q) = f_{r,P}(Q)^{\frac{p^k - 1}{r}}. \end{aligned}$$

2.2 pairing based cryptography

The most useful property in pairing based cryptography is bilinearity:

$$\forall (n,m) \in \mathbb{Z}^2,\ e([n]P,[m]Q) = e(P,Q)^{nm}.$$

Pairings permit several protocol simplifications and original scheme creation, for example identity based cryptography protocols. A recent book [9] is dedicated to idendity based cryptography.

The general scheme of an identity based encryption is described in [2].

2.3 side channel attacks in pairing based cryptography

The important point in identity based cryptography is that to decipher a message using an Identity Based Protocol, a computation of a pairing involving the private key and the message is done. Side channel attacks can be therefore applied to find this secret. The particularity of identity based cryptography is that an attacker can know the algorithm used, the number of iterations and the exponent. The secret is only one of the arguments of the pairing.

3 Differential Analysis Family Attack

The aim of this family of attacks is to use the interaction during the execution of an algorithm of the secret and known elements. In the family of differential analysis attack, we include for example the differential power and electromagnetic attacks [4], or the correlation attack [13]. In order to simplify the explanation, we describe the Differential Power Analysis (DPA) Attack.

3.1 Description of Differential Power Analysis Attack

We now introduce some theoretical issues that allow the reader to understand the principle underlying the DPA attack. We consider the output of a gate whose state depends on both the plain text under ciphering (primary inputs) and the secret key. It is called the target node. We consider now a sequence of input patterns $P_0, P_1, \ldots, P_n$ that generate the transitions $T_1(P_0 \rightarrow P_1), T_2(P_1 \rightarrow P_2), \ldots, T_n(P_{n-1} \rightarrow P_n)$ on the circuit primary inputs. A logic simulation of the circuit while monitoring the target node allows classifying these input transitions in two sets, according to a guess on the key:

- P_A, composed by the transitions that make the target node to commute from 0 to 1 and therefore that make the target gate to consume current;
- P_B, composed by the transitions that do not lead the target gate to participate to the power consumed by the circuit (i.e., transitions from 0 to 0, 1 to 1, and 1 to 0 on the target node).

Once we split in two the curves, we compute the average consumption curve in each set P_A and P_B and compute the difference. If any pic of consumption is find then the hypothesis was correct, if not, then it was false.

4 Fault attack Family

4.1 Description of the fault attack

The goal of a fault injection attack is to provoke mistakes during the calculation of an algorithm, for example by modifying the internal memory, in order to reveal sensitive data. This attack needs a very precise positioning and an expensive apparatus to be performed. Nevertheless, current technologies could allow for this attack [7].

We follow the scheme of attack described and completed in [5]. We assume that the pairing is used during an Identity Based Protocol, the secret point is introduced in a smart card or an electronic device and is a parameter of the pairing. In order to find the secret, we modify the number of iterations in the Miller's algorithm by the following way.

First of all, we have to find the flip-flops belonging to the counter of the number of iterations (i.e. $\log_2(s)$) in the Miller's algorithm. This step can be done by using reverse engineering procedures. In classical architecture, the counter is divided into small pieces of 8 bits. To find it, we make one normal execution of the algorithm, without any fault. Then we choose one piece of the counter, and provoke disturbances in order to modify it and consequently the number of iterations of the Miller's algorithm. For example the disturbance can be induced by a laser. Nowadays lasers are thin enough to make this attack realistic [7]. Counting the clock cycles, we are able to know how many iterations the Miller loop has done. Each time, we record the value of the Miller loop and the number of iterations we made. The aim is to obtain a couple $(\tau, \tau+1)$ of two consecutive values, corresponding to τ and $\tau+1$ iterations during the Miller's algorithm.

We denote the two results by $F_{\tau,P}(Q)$ and $F_{\tau+1,P}(Q)$. To conclude the attack, we consider the ratio $\frac{F_{\tau+1,P}(Q)}{F_{\tau,P}(Q)^2}$. By identification in the basis of $\mathbb{F}_{q^k}$, we are lead to a system which can reveal the secret point, which is described in Section **??**.

The probability for obtaining two consecutive numbers is sufficiently large to make the attack possible. In fact, for an 8-bits architecture only 15 tests are needed to obtain a probability larger than one half, $P(15, 2^8) = 0.56$, and only 28 for a probability larger than 0.9.

5 Countermeasure to these attacks

To protect an identity based protocol against side channel attacks, several countermeasure where introduced.

5.1 The third Coron's countermeasure

This countermeasure is efficient only for projective and Jacobien coordinates. It consists in using homogeneity of the coordinates. For exemple, in projective coordinates, a point $P = (X_P, Y_P, Z_P)$ is also the point $(\lambda X_P, \lambda Y_P, \lambda Z_P)$ for λ a non zero integer.

5.2 The bilinearity countermeasure

There are several countermeasure using the bilinearity of pairings. We give here some of them:

- take random $\lambda \neq 0$ and compute $e(\lambda P, \frac{1}{\lambda}Q)$,
- take λ and γ such that $\lambda \times \gamma = 1 \mathrm{mod}(r)$ and compute $e(\lambda P, \gamma Q)$,
- take a random point R and compute $\frac{e(P+R,Q)}{e(R,Q)}$.

6 Conclusion

We present side channel attacks in pairing based cryptography and the countermeasures to prevent them. The less expensive countermeasure is the third Coron's countermeasure. This countermeasure is very easy to introduce in a pairing algorithm and cost only three multiplications in $\mathbb{F}_p$, the drawback is this countermeasure is efficient only for projective and Jacobien coordinates. The other techniques are efficient for every system of coordinates but they are much more expensive. The countermeasure using integers needs a point multiplication over $E(\mathbb{F}_p)$ and $E(\mathbb{F}_{p^k})$. The countermeasure using a random point involves two pairing computations. Consequently, it seems more interesting to implement pairing in projective or jacobien coordinates and then use the third Coron's countermeasure.

References

1. C. Arne, T. Lange, M. Naehrig, and C. Ritzenhaler. Faster Pairing Computation of the Tate pairing. Cryptology ePrint Archive, Report 2009/155, `http://eprint.iacr.org/2009/155`, 2009.
2. D. Boneh and M.K. Franklin. Identity-Based Encryption from the Weil Pairing. In *CRYPTO '01: Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, pages 213–229, London, UK, 2001. Springer-Verlag.
3. D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. In Colin Boyd, editor, *Advances in ASIACRYPT 2001*, volume 2248 of *mylncs*, pages 514–532. Springer Verlag, 2001.
4. E. De Mulder, S. B. Örs, B. Preneel, and I. Verbauwhede. Differential power and electromagnetic attacks on a fpga implementation of elliptic curve cryptosystems. *Comput. Electr. Eng.*, 33(5-6):367–382, 2007.
5. N. El Mrabet. What about vulnerability to a fault attack of the Miller's algorithm during an identity based protocol? In *Advances in Information Security and Assurance*, volume 5576, pages 122–134, London, UK, 2009. Springer-Verlag.
6. N. El Mrabet, G. Di Natale, and M.L. Flottes. A practical differential power analysis attack against the miller algorithm. In *PRIME 2009 - 5th Conference on Ph.D. Research in Microelectronics and Electronics, Circuits and Systems Magazine*, IEEE Xplore, 2009.
7. D.H Habing. The use of lasers to simulate radiation-induced transients in semiconductor devices and circuits. In *IEEE Transactions On Nuclear Science*, volume 39, 1992.
8. A. Joux. A one round protocol for tripartite Diffie-Hellman. In *ANTS-IV: Proceedings of the 4th International Symposium on Algorithmic Number Theory*, pages 385–394, London, UK, 2000. Springer-Verlag.
9. M. Joye and G. Neven. *Identity-Based Cryptography.* vol. 2 of Cryptology and Information Security Series, IOS Press, 2009.
10. A. Menezes, T. Okamoto, and S. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. In *STOC '91: Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 80–89, New York, NY, USA, 1991. ACM.
11. V. S. Miller. The Weil pairing, and its efficient calculation. In *Jounal of Cryptology*, volume 17, pages 235–261, Secaucus, NJ, USA, 2004. Springer-Verlag New York, Inc.
12. D. Page and F. Vercauteren. Fault and side channel attacks on pairing based cryptography. In *IEEE Transactions on Computers*, 2006.
13. C. Whelan and M. Scott. Side channel analysis of practical pairing implementation: Which path is more secure? In *VietCrypt 2006*, Lecture Notes in Computer Science, 2006.
14. C. Whelan and M. Scott. The importance of the final exponentiation in pairings when considering fault attacks. In *Pairing-Based Cryptography Pairing 2007*, volume 4575 of *Lecture Notes in Computer Science*, pages 225–246, 2007.

WEIGHT ESTIMATES FOR CYCLIC CODES VIA WEIL-SERRE BOUND FOR ADDITIVE EQUATIONS

Cem Güneri
Faculty of Engineering and Natural Sciences
Sabancı University
Tuzla, 34956, İstanbul, Turkey
e-mail: guneri@sabanciuniv.edu

Ferruh Özbudak
Department of Mathematics
Middle East Technical University
İnönü Bulvarı, 06531, Ankara, Turkey
e-mail: ozbudak@metu.edu.tr

Abstract. Wolfmann gave a general weight estimate for a large class of cyclic codes using the Weil-Serre bound. His estimate can be applied to any cyclic code over a prime finite field $\mathbb{F}_p$ but there are codes over extension fields to which Wolfmann's bound does not apply. We extend Wolfmann's result to all cyclic codes over any finite field. Interestingly, the problem of estimating weights of cyclic codes reduces to computation of degrees of left greatest common divisors in the ring of additive polynomials with coefficients from a finite field.

Keywords: Cyclic code, trace representation, additive polynomial, left greatest common divisor, Weil-Serre bound.

1. Introduction to Wolfmann's Result

Weights in cyclic codes can be determined by the number of solutions of certain equations over finite fields. Let $\mathbb{F}_q$ be the finite field with $q = p^e$ elements ($e \geq 1$) and $\mathbb{F}_{q^m}$ be its extension of degree $m \geq 2$. Let α be a primitive element of $\mathbb{F}_{q^m}$ and C be a q-ary cyclic code of length $n = q^m - 1$ such that $\{t_1, t_2, \ldots, t_s\} \subseteq \{1, \ldots, n-1\}$ is a *basic zero set* for the dual cyclic code $C^\perp$. In other words, the generator polynomial of $C^\perp$ is $\prod_{j=1}^{s} f_j(x)$, where $f_j(x) \in \mathbb{F}_q[x]$ is the minimal polynomial of α^{t_j} for $j = 1, \ldots, s$. If Tr denotes the trace map from $\mathbb{F}_{q^m}$ to $\mathbb{F}_q$, then one has the following representation for C ([6, Proposition 2.1]):

$$C = \left\{ \left(\mathrm{Tr}(f(\alpha^0)), \ldots, \mathrm{Tr}(f(\alpha^{n-1})) \right) : f(x) = \sum_{j=1}^{s} \lambda_j x^{t_j} \in \mathbb{F}_{q^m}[x] \right\} \tag{1.1}$$

Then by Hilbert's Theorem 90 ([3, Theorem 2.25]), we have that the weight of the codeword $c_{\boldsymbol{\lambda}} \in C$, determined by $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_s) \in \mathbb{F}_{q^m}^s$ as above, is

$$w(c_{\boldsymbol{\lambda}}) = q^m - \frac{N_{\boldsymbol{\lambda}}}{q}, \tag{1.2}$$

where $N_{\boldsymbol{\lambda}}$ is the number of solutions of the following equation in $\mathbb{F}_{q^m} \times \mathbb{F}_{q^m}$:

$$y^q - y = \lambda_1 x^{t_1} + \cdots + \lambda_s x^{t_s} \tag{1.3}$$

If we assume that each exponent t_j in (1.3) is relatively prime to p, then the equation defines an irreducible Artin-Schreier type curve for any choice of coefficients $\lambda_1, \ldots, \lambda_s$, as long as not all

$\lambda_j = 0$. Genus of such a curve is given by

$$g = \frac{(q-1)(t-1)}{2},$$

where $t := \max\{t_j : \lambda_j \neq 0\}$ (see [5, Proposition III.7.10] for these facts). One can estimate the number of $\mathbb{F}_{q^m}$-rational points of such a curve by the Weil-Serre estimate ([5, Theorem V.3.1]).

Using these observations, Wolfmann wrote a general algebraic geometric bound on the weights of a q-ary cyclic code of length $q^m - 1$ under the assumption that the dual code's basic zeros $\{t_1, t_2, \ldots, t_s\}$ are all relatively prime to the characteristic p (see [6]). Note that an element t_j in a basic zero set for a cylic code can be replaced by any other element in its q-cyclotomic coset $\{t_j, qt_j, q^2t_j, \ldots\}$ mod $(q^m - 1)$. When $q = p$ is a prime, the smallest member of the cyclotomic coset is necessarily relatively prime to p. Hence, Wolfmann's bound applies to any cyclic code of length $p^m - 1$ over a prime finite field $\mathbb{F}_p$.

2. Dealing with Reducible Additive Equations and an Extension of Wolfmann's Bound to All Cyclic Codes

The following examples show that equations of the form (1.3), which are related to cyclic codes, need not define irreducible curves when q is not prime.

Example 2.1. Let $q = 4$ and $n = 4^3 - 1 = 63$. Consider the 4-ary cyclic code C of length n whose dual's basic zero set is $\{1, 2\}$. In other words, if α is a primite element of $\mathbb{F}_{64}$, then the generator polynomial of the dual code $C^\perp$ is $f_1(x)f_2(x)$, where $f_i(x)$ denotes the minimal polynomial of α^i over $\mathbb{F}_4$.

Observe that the 4-cyclotomic coset mod 63 containing 2 is $\{2, 8, 32\}$ and none of the elements in the coset is relatively prime to the characteristic 2.

Here is the related family of curves we have to study to estimate weights of codewords in C:

$$\mathcal{F} = \left\{y^4 - y = \lambda_1 x + \lambda_2 x^2 : \ \lambda_1, \lambda_2 \in \mathbb{F}_{64}\right\}$$

If only one of the coefficients λ_1 or λ_2 is nonzero, the resulting curve is rational with 64 (affine) rational points over $\mathbb{F}_{64}$. However, for the equation $y^4 - y = x + x^2$, we have

$$\begin{aligned} y^4 + y + x^2 + x &= y^4 + y^2 + x^2 + y^2 + y + x \\ &= \left(y^2 + y + x\right)^2 + \left(y^2 + y + x\right) \\ &= \left(y^2 + y + x\right)\left(y^2 + y + x + 1\right) \end{aligned}$$

Hence the curve defined by $y^4 + y = x^2 + x$ has 2 irreducible components both of which are rational curves. Therefore, there are 128 (affine) rational points over $\mathbb{F}_{64}$.

Example 2.2. Let us present another explicit decomposition of a reducible curve of the form (1.3) when $q = 16$:

$$\begin{aligned} y^{16}+y+x^8+x^2 &= (y^8+y^4+y^2+y+x^4+x^2)^2 \\ &\quad +(y^8+y^4+y^2+y+x^4+x^2) \\ &= (y^8+y^4+y^2+y+x^4+x^2) \\ &\quad (y^8+y^4+y^2+y+x^4+x^2+\underbrace{1}_{\nu^2+\nu}) \\ &= (y^4+y+x^2)(y^4+y+x^2+1) \\ &\quad (y^4+y+x^2+\nu)(y^4+y+x^2+\nu+1) \end{aligned}$$

Here, ν is defined in such a way that $\mathbb{F}_4=\mathbb{F}_2(\nu)$ with $\nu^2+\nu+1=0$. Each of the four factors in the last step above defines a rational curve over the field of definition, which is some extension of $\mathbb{F}_{16}$.

Motivated by such examples, we studied factorizations of "reducible equations" of the form (1.3) in [1, 2]. Naturally, explicit factorization as in the above examples is rather complicated and can only be done in specific situations. In particular, we could write weight bounds for all cyclic codes over prime fields $\mathbb{F}_p$ and their square extensions $\mathbb{F}_{p^2}$ by carrying out explicit factorizations of related reducible equations (cf. [1, Theorem 3.2, Theorem 3.3]). Our code bound was an extension of Wolfmann's bound. A more effective result that applies to all cyclic codes over all finite fields required a less computational and non-explicit approach. In order to describe this result, let us introduce some terminology.

A polynomial of the form

$$A(T)=a_nT^{p^n}+a_{n-1}T^{p^{n-1}}+\cdots+a_1T^p+a_0T$$

with coefficients in a field K of characteristic p is called an *additive polynomial*. We further assume that K is a perfect field. Our main objects of study are equations of the form

$$A(T)=z \tag{2.1}$$

for $z\in K(x)$. We will refer to (2.1) as an *additive equation*. Note that (1.3) is an example of an additive equation. By abuse of language we will call (2.1) an *irreducible additive equation* if the polynomial $A(T)-z$ is irreducible over $K(x)$. Otherwise, (2.1) will be called a *reducible additive equation*.

Let $A(T)$ and $B(T)$ be two additive polynomials in $K[T]$ defined by

$$\begin{aligned} A(T) &= a_nT^{p^n}+a_{n-1}T^{p^{n-1}}+\cdots+a_1T^p+a_0T, \\ B(T) &= b_mT^{p^m}+b_{m-1}T^{p^{m-1}}+\cdots+b_1T^p+b_0T, \end{aligned}$$

where $a_nb_m\neq 0$. Note that the ordinary product of A and B is not an additive polynomial but their composition

$$A(T)\circ B(T):=A(B(T)) \tag{2.2}$$

produces an additive polynomial. Under summation and composition, the set of additive polynomials forms a subring of $K[T]$. We will denote this ring by $\mathcal{R}$ throughout.

We say that A is *left divisible* by B if there exists another additive polynomial $C(T) \in \mathcal{R}$ such that $A = B \circ C$. If A is not left divisible by B then, assuming that $\deg A \geq \deg B$, one can carry out a left division of A by B as

$$A(T) = B(T) \circ \left(\left(\frac{a_n}{b_m} \right)^{1/p^m} T^{p^{n-m}} + \cdots \right) + R(T),$$

where $R(T) \in \mathcal{R}$ is of degree less than $\deg B$, unless $R = 0$. Hence, over a perfect field K, the ring $\mathcal{R}$ has a left Euclidean algorithm and any two additive polynomials $A, B \in \mathcal{R}$ have a monic *left greatest common divisor* $\mathrm{lgcd}(A, B)$ in $\mathcal{R}$.

Assume that A is monic, separable and splits completely in K with the zero set W. Hence, $W \subseteq K$ and $|W| = p^n$. It is well-known that the set W is an $\mathbb{F}_p$-linear subspace of K. There is another $\mathbb{F}_p$-subspace of K associated with A. Namely, let $\overline{W}$ be the set of elements $u \in K$ such that

$$A(T) = \left(T^p - \frac{1}{u^{p-1}} T \right) \circ Q(T) \tag{2.3}$$

for some $Q(T) \in \mathcal{R}$. The space $\overline{W}$ is called the *adjoint space* of A. Its dimension is equal to that of W (i.e. $\dim_{\mathbb{F}_p}(\overline{W}) = n$). We refer to [4, Theorems 14, 15 and 16] for the proofs of these facts.

The following is our main theorem in [2], which determines when an additive equation is reducible and if so, describes its factorization to irreducible components.

Theorem 2.3. *Let $A(T), B_1(T), \ldots, B_t(T) \in \mathcal{R}$ be nonzero additive polynomials and assume that A is monic, separable and splits in K. Let $r_1, \ldots, r_t$ be distinct positive integers with $\gcd(p, r_i) = 1$, for all $1 \leq i \leq t$. Set*

$$L(T) := \mathrm{lgcd}(A(T), B_1(T), B_2(T), \ldots, B_t(T)).$$

Let $\hat{A}(T), \hat{B}_1(T), \ldots, \hat{B}_t(T) \in \mathcal{R}$ be additive polynomials such that

$$\begin{aligned} A(T) &= L(T) \circ \hat{A}(T) \quad \text{and} \\ B_i(T) &= L(T) \circ \hat{B}_i(T) \quad \text{for } 1 \leq i \leq t. \end{aligned}$$

Let W be the set of roots of $L(T)$. Then,

i)

$$A(T) - \sum_{i=1}^{t} B_i(x^{r_i}) \text{ is irreducible over } K(x) \text{ if and only if } L(T) = T.$$

ii) The following polynomial identity holds in the ring $K(x)[T]$:

$$A(T) - \sum_{i=1}^{t} B_i(x^{r_i}) = \prod_{w \in W} \left(\hat{A}(T) - \sum_{i=1}^{t} \hat{B}_i(x^{r_i}) - w \right)$$

iii) for each $w \in W$, the polynomial

$$\hat{A}(T) - \sum_{i=1}^{t} \hat{B}_i(x^{r_i}) - w \in K(x)[T]$$

is irreducible.

If K is a finite field, Theorem 2.3 yields a Weil-Serre type bound for a large class of (reducible or irreducible) additive equations.

Corollary 2.4. *Let $K = \mathbb{F}_q$ be a finite field of characteristic p in Theorem 2.3 and let N denote the number of solutions $(x, y) \in \mathbb{F}_q \times \mathbb{F}_q$ of*

$$A(y) = \sum_{i=1}^{t} B_i(x^{r_i}). \tag{2.4}$$

If $\deg A = p^n$ and $\deg L = p^\mu$, then

$$|N - qp^\mu| \leq \frac{(p^n - p^\mu)(R-1)}{2} \lfloor 2\sqrt{q} \rfloor, \tag{2.5}$$

where $R = \max\{r_1, \ldots, r_t\}$.

When these results are applied to (reducible or irreducible) additive equations with $A(T) = T^q - T$ (cf (1.3)), we obtain the necessary estimates for determining weight bounds for arbitrary cyclic codes. Our method requires the degrees of certain left greatest common divisor polynomials. It can be shown by examples that our bounds perform very well against the BCH bound and they yield the exact minimum distance in some cases.

References

[1] C. Güneri and F. Özbudak, "Cyclic codes and reducible additive equations", *IEEE Trans. Inform. Theory*, vol. 53, no. 2, 848-853, 2007.

[2] C. Güneri and F. Özbudak, "Weil-Serre type bounds for cyclic codes", *IEEE Trans. Inform. Theory*, vol. 54, no. 12, 5381-5395, 2008.

[3] R. Lidl and H. Niederreiter, *Finite Fields*, Cambridge University Press, Cambridge, 1997.

[4] O. Ore, "On a special class of polynomials", *Trans. Amer. Math. Soc.*, vol. 35, no. 3, 559-584, 1933.

[5] H. Stichtenoth, *Algebraic Function Fields and Codes*, Springer-Verlag, Berlin, 1993.

[6] J. Wolfmann, "New bounds on cyclic codes from algebraic curves", in: Lecture Notes in Computer Science, vol. 388, pp. 47-62, New York: Springer-Verlag, 1989.

Inside KECCAK (Extended Abstract)

Guido Bertoni[1], Joan Daemen[1], Michaël Peeters[2], and Gilles Van Assche[1]

[1] STMicroelectronics
[2] NXP Semiconductors

Abstract. In this extended abstract we summarize the algebraic properties of the building blocks of KECCAK relevant in cryptanalysis and implementation. KECCAK is a sponge function family that makes use of an underlying set of fixed-length iterated permutations called KECCAK-f. The round function of KECCAK-f consists of 5 step mappings chosen for their ease of implementation and strong long-term diffusion and nonlinearity. We discuss the individual properties of these step mappings such as invertibility, translation-invariance and (when applicable) non-linearity, their interactions and the implications for resistance to attacks such as linear and differential cryptanalysis.

Keywords: Keccak, algebraic properties

1 Introduction

KECCAK is a parameterized sponge function that uses as a building block a set of 7 related permutations of different sizes, called KECCAK-f. There are 7 KECCAK-f permutations, indicated by KECCAK-$f[b]$, where $b = 25 \times 2^\ell$ and ℓ ranges from 0 to 6. KECCAK-$f[b]$ is a permutation over $s \in \mathbb{Z}_2^b$.

The permutation KECCAK-$f[b]$ operates on a state a that is a three-dimensional array of elements of GF(2) denoted by $a[x][y][z]$. The size of the array depends on ℓ and is given by $a[5][5][w]$. It consists of $w = 2^\ell$ *slices*, one for each value of z. A slice contains 25 bits arranged in 5 rows, one for each value of y, and 5 columns, one for each value of x. A state can also be seen as 25 *lanes*, orthogonal to the slices. For the different values of ℓ, the *lane size* w ranges from 1 to 64 in powers of two.

KECCAK-$f[b]$ is an iterated permutation, consisting of a sequence of n_r rounds with $n_\mathrm{r} = 12 + 2\ell$. For KECCAK-$f[1600]$ this results in 24 rounds.

KECCAK is specified in [1] and the properties of its components is described in detail in [2]. In this abstract we give an overview of the algebraic properties of the components of the KECCAK-f round function and refer to [2] for the details. We first list the design principles underlying the KECCAK-f permutations. This is followed by a discussion on the algebraic properties of its components. Finally we explain what this means for the interaction between the different components relevant in cryptanalysis and implementation.

2 Design principles

As a consequence of the underlying design principles, the step mappings have a number of remarkable properties, quite rare for components of symmetric cryptographic primitives. These design principles are the following:

- **Minimalism:** each step mapping has a simple structure and description with operations in GF(2). Each step mapping has a particular purpose and omitting one of them would result in a collapse of security.
- **Symmetry:** the processing of the different bits of the state depends as little as possible on the bit positions. This is reflected in the property of translation-invariance [2, Section 6.1].
- **Transferability of analysis:** analysis of smaller versions of KECCAK-f is relevant for the larger versions. This is reflected in the Matryoshka structure [2, Section 6.2].
- **Efficiency, but only where needed:** the round function is easy to implement on a wide range of platforms and is suited for side-channel resistant implementations. There are no such requirements for the inverse of the round function (as opposed to a block ciphers).

3 The step mappings and their properties

A round consists of five step mappings:

- χ: nonlinearity
- θ: linear mixing for diffusion
- π: bit transposition in the xy direction for *dispersion* (moving nearby bits to remote positions)
- ρ: bit transposition in the z direction for dispersion
- ι: round constants addition (in GF(2)) for destruction of symmetry

In the following subsections we will summarize the algebraic properties of the first four step mappings. For a more detailed treatment we refer to [2, Section 6.3]

3.1 The nonlinear mapping χ

χ is the nonlinear mapping of the round function and operates on the rows independently. It is translation-invariant in any direction and its operation is independent of the lane size w. χ simply consists of adding to each bit the product (AND) of two other statebits, with one of them complemented. It follows that its algebraic degree is 2. Its invertibility is however not trivial and its inverse has algebraic degree 3.

The fact that the algebraic degree is two has consequences for the difference propagation through χ and correlation across χ.

Given a difference pattern a at the input of χ, the output difference patterns form a linear affine variety. This can be represented by an offset and a basis with $w(a)$ elements and hence has $2^{w(a)}$ elements, with $w(a)$ the *restriction weight* of the difference pattern a. For all the output difference patterns, the differential probability (DP) is equal to $2^{-w(a)}$ and is fully determined by the input difference pattern a. A difference propagation (a, b) imposes w linear conditions on state bits at the input of χ.

Given a bit parity at the output of χ defined by a *(linear) mask* u, the masks of the bit parities at the input of χ that have a non-zero correlation with it, form a linear affine variety. This can

be represented by an offset and a basis with $w(u)$ elements and hence has $2^{w(u)}$ elements, with $w(u)$ the *correlation weight* of the mask u. For all the compatible input masks the magnitude of the correlation is equal to $2^{-w(a)/2}$ and is fully determined by the output mask.

3.2 The linear mixing layer θ

θ is the linear mixing layer of the round function. It is translation-invariant in any direction and operates independently of the lane size w if $w > 1$. θ simply consists of the addition to each bit of the state the parity of two columns. Every output bit of θ is the bitwise sum of 11 input bits. Computing the inverse of θ is a non-trivial exercise and requires calculations in ideals over polynomial rings in three variables. It turns out that every input bit of θ is the bitwise sum of about half of its output bits. The inverse of θ depends on the lane size w.

For the subset of states with an even number of 1-bits in each column, θ is the identity mapping. We call this subset the (*column parity*) *kernel*. In general, the column parity (i.e., a two-dimensional array containing the parities of all columns) of state a fully determines the column parity of $\theta(a)$. The column parity imposes lower bounds on the total number of bits equal to 1 (active bits) in a and $\theta(a)$ jointly: a parity-specific branch number.

3.3 The transposition of lanes π

π is one of the two dispersion layers and its purpose is to change the relative positions of bits in the slices. It is translation-invariant in the direction of the z-axis and operates independently of the lane size w. It moves the bits within a slice to other positions where the new position (X, Y) is computed from the old position (x, y) by a multiplication with a 2×2 matrix M with elements in GF(5). The inverse of π is defined by the inverse matrix.

The multiplication by M is a linear mapping with a number of interesting properties. It leaves the origin $(0, 0)$ invariant and moves the other 24 positions around in a single cycle. The relative position of two points before π fully determines their relative position after π. As such it respects the 6 possible partitions of the 25 (x, y) positions according to lines parallel to lines through the origin. As such, the 5 positions in a row are mapped to a column, a falling 2-slope, a rising 2-slope, a falling 1-slope, a rising 1-slope and back to a row again in 6 iterations.

3.4 The intra-lane translation ρ

ρ is one of the two dispersion layers and its purpose is to move bits in the same slice to different slices. It is translation-invariant in the direction of the z-axis and its operation depends on the lane size w. It just (cyclically) translates the bits within the lane in the direction of the z-axis, by a different offset for each lane. The translation offsets are the same for all values of w, but as the translation are cyclic, in practice their values are reduced modulo w.

The translation offsets are defined by the sequence $i(i+2)/2$. For any lane size 2^ℓ the sequence $i(i+2)/2 \bmod 2^\ell$ has period $2^{\ell+1}$ and runs through all integers smaller than 2^ℓ.

4 Interaction between the step mappings

4.1 Differential and linear cryptanalysis

The fact that χ is the only nonlinear step mapping in a round, makes that the difference patterns at the input of a round defines a linear affine variety of possible difference patterns at the input of

the next round. Similarly, a mask at the output of a round defines a linear affine variety of possible masks at the output of the previous rounds.

These linear affine varieties can be seen as the product a linear affine variety in the kernel and a linear affine variety ortogonal to the kernel. As such, difference propagations with small branch number can be identified both in the case of differential and linear cryptanalysis. this can be used in proving lower bounds for the weight of linear and differential trails.

All step mappings are translation-invariant in the direction of the z-axis. This implies that for any differential or linear trail, there are w equivalent ones that are just z-translated versions. On the other hand, as a consequence of the Matryoshka structure of Keccak-f, any trail for a given lane size w also defines one for w with twice the weight [2, Section 6.5.2].

Exploiting the kernel allows finding low weight trails for 2 and 3 rounds, but not beyond [2, Sections 6.5.3 to 6.5.6].

4.2 Implementation

Keccak-f is well suited to be implemented in software with bitwise logical operations and cyclic shifts. These cyclic shifts implement the translations in the direction of the z-axis and require the word size of the CPU to be equal to the lane size w. However, the high level of symmetry and the translation-invariance in the direction of the z-axis allow the use of cyclic shifts also when the CPU word length divides the lane size. This can be done with the technique of bit interleaving. In its simplest form, a lane is stored in two words: one word containing the bits with even z coordinates and one the bits with odd z coordinates [2, Section 9.2.2].

The fact that the round function of Keccak-f only has degree 2 makes it very well suited for implementations that provide resistance against differential power analysis. This low degree makes it suitable for implementing secret sharing methods, both in software and in hardware [2, Sections 9.3.4 and 9.4.4].

References

1. G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche, Keccak *specifications, version 2*, NIST SHA-3 Submission, September 2009, **jvvr<11mgeecmOpqgmgqpOqti1**.
2. _____, Keccak *sponge function family main document*, NIST SHA-3 Submission (updated), June 2010, **jvvr<11mgeecmOpqgmgqpOqti1**.

Towards a Spectral Approach for the Design of Self-Synchronizing Stream-Ciphers

Jérémy Parriaux[1], Philippe Guillot[2], and Gilles Millérioux[1]

[1] Nancy University, CNRS,
Research Center for Automatic Control of Nancy (CRAN UMR 7039), France,
jeremy.parriaux@esstin.uhp-nancy.fr, gilles.millerioux@esstin.uhp-nancy.fr,
[2] Université Paris 8, France
Laboratoire Analyse, Géométrie et Applications (LAGA UMR 7539),
philippe.guillot@univ-paris8.fr

Abstract. This paper addresses the problem of self-synchronization of a class of dynamical systems involving Boolean functions. We motivate the work in the context of cryptography with the perspective of designing self-synchronizing stream ciphers and assessing their efficiency in terms of security. The self-synchronization is tackled through the notion of influence of variables. We propose a spectral characterization of the self-synchronization property through the Walsh transform. We discuss two kinds of self-synchronization: the finite time and the statistical one.

1 Introduction

Stream-ciphers are cryptosystems well suited when the data to be encrypted are generated sequentially or when the encryption speed is a concern. Indeed, their simple structure usually allows higher throughput than block ciphers. Unlike block ciphers, they involve a time varying transformation applied to the plain-text giving rise to the interesting property that two identical plain-texts may have different cipher-texts. An important requirement for proper decryption is to guarantee that the decipher is well synchronized with the cipher. This is often ensured by resorting to synchronization protocols which introduce overheads in the stream packets conveyed through the channel. Such an approach applied for the so-called synchronous stream ciphers. A special class of stream-ciphers has the interesting property of self-synchronization. By self-synchronization, it is meant an intrinsic (also called structural) synchronization between the decipher and the cipher rendering the synchronization protocols useless. They are called self-synchronizing stream-ciphers (SSSC for short). And yet, it must be stressed that few attention has been paid on them. The reader may refer to [1] for details about the design of SSSC.

In this paper, we focus on the self-synchronization property for systems involving Boolean functions [2][3] having in mind the cryptographic context and more specifically the use of self-synchronizing stream ciphers. The self-synchronization is tackled through the notion of influence of variables which is developed herein. Then, we propose a spectral characterization of the self-synchronization property. The layout is the following: Section 2 is devoted to the problem statement. Canonical equations of SSSC are recalled. In Section 3, we expose the main results which

essentially consist in a spectral characterization of the self-synchronization property and the notion of influence of variables.

2 Problem Statement

Concealment in stream-ciphers is usually done by performing the exclusive-OR (denoted by $\oplus$) between the successive plain-text bits (denoted by p_k) and the bits of a key-stream (denoted by K_k). The index k stands for the discrete-time. The cipher and the decipher can be respectively modeled as dynamical systems governed by the sets of equations (1) and (2).

$$\begin{cases} x_k &= f(c_k, x_{k-1}) \\ K_k &= g(x_k, c_{k-1}) \\ c_k &= K_k \oplus p_k \end{cases} \qquad \text{(cipher equation)} \tag{1}$$

$$\begin{cases} \widehat{x}_k &= f(c_k, \widehat{x}_{k-1}) \\ \widehat{K}_k &= g(\widehat{x}_k, c_{k-1}) \\ \widehat{p}_k &= \widehat{K}_k \oplus c_k \end{cases} \qquad \text{(decipher equation)} \tag{2}$$

Let us call a (n, m)–function a function from the vector space $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$. x^k and $\widehat{x}_k$ are respectively the state vectors of the cipher and of the decipher. The next-state function f is an $(n+1, n)$ function depending on x_{k-1} and the cipher-text c_k. g is an $(n, 1)$–function which delivers the key-streams K_k or $\widehat{K}_k$. Finally, $\widehat{p}_k$ stands for the recovered plain-text. For a proper decryption of the cryptogram c_k, the states x_k and $\widehat{x}_k$ must coincide at each discrete-time k. Roughly speaking, the compound system (1) and (2) is self-synchronizing if the coincidence $x_k = \widehat{x}_k$ is achieved whatever are the initial states x_0 and $\widehat{x}_0$, possibly after a transient time. In this sense, the synchronization is an intrinsic (or structural) property of the system. It can easily be seen that this property depends exclusively on the function f. Now, let us formally define the self-synchronization property. We denote by (c) the sequence $(c_0, \ldots, c_i)$.

Definition 1 (Self-synchronizing sequence). *A sequence (c) is self-synchronizing for f if there exists an integer k_c so that for all initial state x_0 and $\widehat{x}_0$*

$$\forall k \geq k_c,\ x_k = \widehat{x}_k \tag{3}$$

Definition 2 (Finite time self-synchronization). *The system (1)-(2) is finite time self-synchronizing if the minimum value k_c is upper bounded over all possible sequences (c). The upper bound k_c is called the self-synchronization delay of f.*

Remark 1. If (c) is a random sequence, then k_c is a random variable. In such a case, it should be denoted K_c.

Definition 3 (Statistical self-synchronization). *The system (1)-(2) is statistically self-synchronizing if $\lim_{k \to +\infty} \text{Prob}(K_c \leq k) = 1$. K_c is called the random synchronization delay for the random sequence (c).*

3 Main Results

This section aims at stating a spectral characterization of the self-synchronizing property through the Walsh transform. Such a property is expressed in terms of influence of variables which is developed herein.

3.1 Self-Synchronization and Influence

The function f involved in (1)-(2) is considered as a mapping $f(c,x) : \mathbb{F}_2 \times \mathbb{F}_2^n \longrightarrow \mathbb{F}_2^n$. We define, for any positive integer i, the $(n+i+1,n)$–function ϕ_i which is the iterated function of f. It expresses the internal state after $i+1$ iterations and thereby depends on the initial internal state and the cipher-text sequence. For $x_0 \in \mathbb{F}_2^n$ and $(c) = (c_0, \ldots, c_i) \in \mathbb{F}_2^{i+1}$, $\phi_i(c,x)$ is defined as:

$$\phi_i(c,x) = f(c_i, f(c_{i-1}, f(\cdots, f(c_0, x_0)\cdots))) \tag{4}$$

The self-synchronization property can be directly related to the notion of influence. Indeed, it is worth pointing out that (1)-(2) has the self-synchronization property if there exists an integer i so that ϕ_i does not depend on x_0 anymore.
Therefore, we have the following Proposition

Proposition 1. *The system* (1)-(2) *is self-synchronizing if and only if* x_0 *has no influence over* ϕ_i*.*

The function ϕ_i playing a central role and having in mind a spectral characterization of the influence, we must state some properties of the corresponding Walsh transform.
Let us denote denote by f^0 (respectively f^1) the (n,n)–function which is the restriction of f to the input $c = 0$ (respectively to $c = 1$). For a given fixed input sequence (c), we denote by ϕ_i^c the (n,n)–function that expresses the internal state after $i+1$ iterations: $\phi_i^c : x \longmapsto \phi_i(c,x)$. The spectrum of the function ϕ_i restricted to the input sequence (c) is the Walsh matrix $W_{\phi_i^c}$ of ϕ_i^c.

Proposition 2. *The Walsh matrix of* ϕ_i^c *is*

$$W_{\phi_i^c} = \frac{1}{2^{n+i}} W_{f^{c_i}} W_{f^{c_{i-1}}} \times \cdots \times W_{f^{c_0}}. \tag{5}$$

The expression of the spectrum of ϕ_i can be expressed with the spectrum of f. For two vectors $u = (u_0, \ldots u_i)$ and $v = (v_0, \ldots, v_{n-1})$, their concatenation, denoted $u|v$ is by definition the $(n+i+1)$–dimensional vector $u|v = (u_0, \ldots n_i, v_0, \ldots, v_{n-1})$. We define the operator τ which transforms a binary vector $z = (z_0, \cdots, z_\ell)$ to the integer $\tau(z) = \sum_{i=0}^{\ell} 2^{\ell-i} z_i$.

Proposition 3. *The entry of the* $2^n \times 2^{n+i+1}$ *dimensional Walsh matrix of the iterated function* ϕ_i *is*

$$w_{t,\tau(u|v)}^{\phi_i} = \sum_{c \in \mathbb{F}_2^{i+1}} (-1)^{u \cdot c} w_{t,v}^{\phi_i^c}. \tag{6}$$

3.2 Spectral Characterization of the Influence

Throughout the literature, see in particular [4] and [5], the influence of a set S of variables over a function is often defined as the probability that there exists at least one way to change the value of the function by changing the value of the variables in the set, the variables not in the set being chosen at random. The fact that this definition has no usable spectral expression has a major drawback. Indeed, it does not take into account the number of possibilities of choosing the variables for changing the value of the function. We therefore rather introduce another definition of the influence that takes this point into account. A spectral characterization is more appropriate to this end.

Definition 4. *The influence on a function f of a set S of variables is the mean of the probabilities that $f(x)$ changes when x is randomly chosen, the mean being computed for all possibles changes of variables not in the set S.*

Recall that the support of a vector $u \in \mathbb{F}_2^n$ is by definition $\mathrm{supp}(u) = \{i \in \{1, \cdots, n\} | u_i \neq 0\}$.

Definition 5. *Let f be a $(n,1)$–function of a vector $x \in \mathbb{F}_2^n$ and S a set of ℓ components of x. The influence $I_f(S)$ is:*

$$I_f(S) = \frac{1}{2^n(2^\ell - 1)} \sum_{x \in \mathbb{F}_2^n} \sum_{u \in \mathbb{F}_2^n / u \neq 0,\ \mathrm{supp}(u) \subset S} [f(x) \oplus f(x \oplus u)] \tag{7}$$

It can be shown that $I_f(S)$ can be expressed by means of $\widehat{f_\chi}$ which is the Walsh transform of f. Notice that the correspondence between $\widehat{f_\chi}$ and the Walsh matrix fulfills $w_{1,\tau(v)} = \widehat{f_\chi}(v)$.

Proposition 4. *Let f be a $(n,1)$–function of a vector $x \in \mathbb{F}_2^n$ and S a set of ℓ components of x,*

$$I_f(S) = \frac{2^{\ell-1}}{2^{2n}(2^\ell - 1)} \sum_{v \in \mathbb{F}_2^n / \,\mathrm{supp}(v) \cap S \neq \emptyset} \widehat{f_\chi}^2(v) \tag{8}$$

We extend Definition 5 to a vectorial Boolean function that is to a (n,m)–function with $m \geq 1$.

Definition 6. *The influence of a set of variables S over a vectorial Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$ is*

$$I_f(S) = \frac{1}{m} \sum_{j=1}^{m} I_{f_j}(S) \tag{9}$$

where f_j is the j^{th} component function of f.

3.3 Spectral Characterization of the Self-Synchronization Property

In this section, we characterize the self-synchronizing property based of the results stated above.

Finite Time Self-Synchronization We recall that the self-synchronization property is related to the existence of an integer i so that $\phi_i(c, x_0)$ does not depend on x_0 or equivalently to the fact that the initial state x_0 has no influence over ϕ_i. By virtue of (8) and (9), the influence is proportional to the sum of some squared Walsh coefficients. This yields the following proposition.

Proposition 5. *Let S_{x_0} be the set of n components of x_0 the initial state of the functions ϕ_i. The system (1)-(2) is finite time self-synchronizing if and only if there exists a corresponding iterated function ϕ_i so that,*

$$\forall v \in \mathbb{F}_2^{n+i+1} / \operatorname{supp}(v) \cap S_{x_0} \neq \emptyset,\ \forall j \in [0, 2^{n-1} - 1],\ w^{\phi_i}_{j,\tau(v)} = 0 \tag{10}$$

Statistical Self-Synchronization We now consider the case of the statistical self-synchronization property. If a system is statistically self-synchronizing, there exists at least one self-synchronizing sequence. It is possible to characterize such a sequence using the following proposition.

Proposition 6. *The sequence (c) of length $i+1$ is a self-synchronizing sequence if and only if the Walsh matrix of the function ϕ_i^c is of the form*

$$W_{\phi_i^c} = \begin{pmatrix} 2^n & 0 & \cdots & 0 \\ \pm 2^n & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ \pm 2^n & 0 & \cdots & 0 \end{pmatrix} \tag{11}$$

4 Concluding Remarks

We have focus on the self-synchronization property for systems involving Boolean functions. It is motivated by the fact that such a property is central when addressing cryptographic applications involving the special class of stream ciphers, namely the self-synchronizing ones. We expect that this work constitutes a first step towards a systematic methodology for the design of such ciphers. Indeed, the design amounts to finding suitable Boolean functions f so that the iterated function fulfills the spectral conditions provided in this note.

References

1. G. Millérioux and P. Guillot. Self-synchronizing stream ciphers and dynamical systems: state of the art and open issues. *International Journal of Bifurcation and Chaos, 20(9)*, 2010.
2. *Boolean Models and Methods in Mathematics, Computer Science, and Engineering*, chapter Boolean Functions for Cryptography and Error-Correcting Codes. Cambridge Press, 2010.
3. *Boolean Models and Methods in Mathematics, Computer Science, and Engineering*, chapter Vectorial Boolean Functions for Cryptography. Cambridge Press, 2010.
4. J. Kahn, G. Kalai, and N. Linial. The influence of variables on boolean functions. In *SFCS '88: Proceedings of the 29th Annual Symposium on Foundations of Computer Science*, pages 68–80, Washington, DC, USA, 1988. IEEE Computer Society.
5. Nathan Keller. On the influence of variables on boolean functions in product spaces, May 2009. Availaible at: http://arxiv.org/PS_cache/arxiv/pdf/0905/0905.4216v1.pdf.

De Bruijn Sequences and Complexity of Symmetric Functions

Christelle Rovetta and Marc Mouffron

EADS Defence and Security
Cyber Security Customer Solutions Centre, France

Abstract. A multivalued function is a function from a set E_q^n to a set E_m, where E_k is a set which contains k elements. These functions are used in cryptography: cipher design, hash function design and in theoretical computer science. In this paper, we study the representation of these functions with Multivalued Decision Diagrams (MDD). This representation can be used both to measure complexity and to implement efficiently functions in hardware.
We are especially interested in symmetric functions whose MDDs are shown to have much lower size than classical functions MDDs. One major result is to determine exactly their MDD's maximum size. Notably, we highlight the links between De Bruijn sequences and the most complex symmetric functions and exhibit new functions for $q = 2$ and any m. Enumeration of these functions are supplied, they are numerous enough to allow many applications.

Key words: Symmetric functions, functions over finite sets, hardware implementation, MDD, De Bruijn sequences.

1 Introduction

Today cryptography is spreading everywhere in a lot of devices and peculiarly small, mobile, low energy and low cost pieces of equipment such that Bluetooth earpieces, RFID tags, sensors. To design and implement cryptographic algorithms on these devices, there is an eager need of small footprint Boolean or multivalued functions achieving a good trade-off in term of complexity, implementation and cryptographic properties.

Some works already deal with complexity issues of the partially symmetric Boolean functions [5] and the symmetric multivalued functions [3,6]. They show that decision diagrams [2] are well suited to benefit symmetries and we enhance this further on with excellent results for symmetric multivalued functions. Decision Diagrams (either binary, BDD, or multivalued, MDD) are able both to provide a measure of the complexity of these functions and to achieve an efficient implementation. This is why our analysis makes huge use of this tool.

An asset of symmetric functions is that their MDD have bounds [3,6] of small order, but no result expresses exactly the maximum size. No result neither provides functions achieving this maximum. We investigate this in order to establish the exact maximum value of these MDD, with different levels of reductions [2].

In this paper, we study the structure of symmetric functions MDD and prove their maximum size. We exhibit the functions reaching this maximum value (the "hard" symmetric functions) and give an external characterization of such functions linked to De Bruijn sequences. We then derive some counting results on these "hard" symmetric functions. In this short note some results are shortened and some proofs are omitted, they will be provided in the final paper.

2 Definitions and Notations

2.1 Multivalued Functions

Let us consider $E_k = \{0, 1, \ldots, k-1\}$ the set of k elements, and n, q, m positive integers. Let C_n^k be the choose k among n coefficient. Card(E) stands for the cardinal of set E.

Definition 1. *Given any positive integers n, q, m, we call multivalued function any function from E_q^n to E_m.*

The set of multivalued functions is denoted by $M_n(q, m)$. It contains m^{q^n} functions. Any function $f \in M_n(q, m)$ is characterized by a vector $f_v \in E_m^{q^n}$ called its *value vector*, consisting in the evaluations of the function at every possible input:

$$f_v = (f(0, \ldots, 0), f(0, \ldots, 0, 1), f(0, \ldots, 0, 2), \ldots, f(q-1, \ldots, q-1)) \quad . \tag{1}$$

2.2 Symmetric Multivalued Functions

Definition 2. *A multivalued function $f : E_q^n \longrightarrow E_m$, is symmetric, if f is invariant under any permutation of its input's variables:*
$\forall \sigma \in S_n,\ f(x_1, \ldots, x_n) = f(x_{\sigma(1)}, \ldots, x_{\sigma(n)})$.

The set of these functions is denoted by $SM_n(q, m)$.

Definition 3. *We call symmetry class of $x = (x_1, \ldots, x_n) \in E_q^n$, the set $P_{n,q}(x)$ defined by*

$$P_{n,q}(x) := \left\{ y \in E_q^n \ ; \exists \text{ a permutation } \sigma \text{ such that } y = (y_1, \ldots, y_n) = (x_{\sigma(1)}, \ldots, x_{\sigma(n)}) \right\} \quad (2)$$

We set as representative of the class of symmetry, the smallest element in the lexicographical order.

According to the lemma of Andrew [1], we deduce that there are C_{n+q-1}^{q-1} symmetry classes in E_q^n. We call j^{th} symmetry class of E_q^n the class of symmetry whose representative is classified j^{th} among all representatives according to the lexicographical order.

A symmetric multivalued function can be represented by a vector with values in E_m, whose length equals C_{n+q-1}^{q-1}. The components of the vector are the evaluations of the function for each representative of the symmetry classes. The j^{th} component of the vector corresponds to the evaluation of the function for the j^{th} symmetry class. We call this vector a *simplified vector* of the function:

$$f_{sv} = (f(0, \ldots, 0), \ldots, f(0, \ldots, 0, q-1), f(0, \ldots, 0, 1, 1), \ldots, f(q-1, \ldots, q-1)) \ . \quad (3)$$

There are $m^{C_{n+q-1}^{q-1}}$ possible simplified vectors, i.e. as many as symmetric multivalued functions.

2.3 Multivalued Decision Diagrams

A Multivalued Decision Diagram (MDD) is a generalization of a Binary Decision Diagram (BDD). In his famous article " Graph-Based Algorithms for Boolean Function Manipulation " [2], Bryant defined a reduction procedure *reduce* which reduces a BDD in a single and optimal way. This procedure applies two rules: the fusion rule and the suppression rule. Bryant's procedures can easily be generalized to deal with the MDD.

Definition 4. *The fusion rule says that two nodes are fusioned if their subgraphs are isomorphic. The suppression rule says that a node is deleted if it has only one child node.*

Definition 5. *A Reduced Ordered Multivalued Decision Diagram (ROMDD) is a MDD reduced by the full reduction procedure (i.e. fusion and suppression rules).*

Definition 6. *A Quasi Reduced Ordered Multivalued Decision Diagram (QROMDD) is a MDD reduced by using only the fusion rule.*

2.4 Complexity

The number of nodes of a MDD is called the *size* of the MDD. We call *height* of a node its distance to the top. The set of nodes having a same height $k \in \{0, 1, \ldots, n\}$ is called *level* k of the MDD.

Definition 7. *Let $f \in M_n(q, m)$, we define:*

- *its complexity, noted $c(f)$, the size of its QROMDD.*
- *its reduced complexity, noted $c_R(f)$, the size of its ROMDD.*

3 Symmetric Functions Representation by Simplified MDD

We put forward an optimized representation of symmetric functions by MDD called a *simplified MDD* which is a QROMDD. The idea is to associate a symmetry class to each node of the MDD.

The *simplified MDD* of a symmetric function $f \in SM_n(q, m)$ is defined as follows.
Let $n_{k,j}$ be the j^{th} node from the left of level $k \in \{0, \ldots, n\}$ and $j \in \{0, \ldots, k+1\}$, then $n_{k,j}$ represents the j^{th} symmetry class of the set E_q^k.

- If $k = n$ then $n_{k,j}$ is terminal. Its value is equal to $f(x_j)$ where $x \in E_q^n$ is the representative of the j^{th} class of symmetry of the set E_q^n.
- Else, $n_{k,j}$ is not terminal and it has q distinct sons. Each son represents a distinct symmetry class of the set E_q^{k+1}.

A simplified MDD has a height equal to $n+1$ and has C_{k+q-1}^{q-1} nodes on each of its level k.
Thus, the simplified MDD has C_{q+n}^{q} nodes, which enables us to deduce an upper bound for the complexity of any symmetric multivalued function.

Lemma 1. *Let $f \in SM_n(q,m)$, then $c(f) \leq C_{n+q}^{q}$.*

4 Maximum Complexity of Symmetric Functions

A symmetric multivalued function $f \in SM_n(q,m)$ is called *symmetric (multivalued) hard* if its complexity is the largest among the functions of its set, i.e. if

$$c(f) = \max\{c(g); g \in SM_n(q,m)\} \quad . \tag{4}$$

We denote by $HSM_n(q,m)$ the set of the hard symmetric multivalued functions and by $SC_n(q,m)$ the complexity of the functions in $HSM_n(q,m)$, i.e.

$$SC_n(q,m) := \max\{c(f), f \in SM_n(q,m)\} \tag{5}$$

and

$$HSM_n(q,m) := \{f \in SM_n(q,m); c(f) = SC_n(q,m)\} \quad . \tag{6}$$

For all integer $k \in \{0,1,\ldots,n\}$, we define $SR_{n,q,m}(k) = \min\left(C_{k+q-1}^{q-1}, m^{C_{n-k+q-1}^{q-1}}\right)$.

We call *symmetric inflection level* $h(n,q,m)$, the integer $h \leq n$ such that if $n \geq 2$

$$\begin{cases} C_{h+q-2}^{q-1} < m^{C_{n-h+q-2}^{q-1}} \\ C_{h+q-1}^{q-1} \geq m^{C_{n-h+q-1}^{q-1}} \end{cases} \quad \text{i.e.} \quad \begin{cases} SR_{n,q,m}(h-1) = C_{h+q-2}^{q-1} \\ SR_{n,q,m}(h) \quad\quad = m^{C_{n-h+q-1}^{q-1}} \end{cases}$$

We take by convention $h(0,q,m) = 0$ and $h(1,q,m) = 1$.

For all integer $n \geq 1, q \geq 2, m \geq 2$, we call *pair of symmetric inflection* associated with the parameters, the pair $(SR_{n,q,m}(h(n,q,m)), SR_{n,q,m}(h(n,q,m)-1))$.
In the multivalued case, we notice that:

- C_{k+q-1}^{q-1} is the number of nodes of a simplified MDD on the level k,
- $m^{C_{n-k+q-1}^{q-1}}$ is the number of symmetric functions with $n-k$ variables.

Theorem 1. $\forall n \geq 1,\ SC_n(q,m) = \sum_{k=0}^{n} SR_{n,q,m}(k) = C_{q+h(n,q,m)-1}^{q} + \sum_{k=q+h(n,q,m)-1}^{n} m^{C_{n-k+q-1}^{q-1}}$.

5 Simplified Value Vector in the Case $q = 2$ and any m

5.1 General Results for any n

For the particular case $q = 2$ and any m, the simplified value vector can be read directly on the last level of the simplified MDD.

Theorem 2. *Let (l,k) be the inflection symmetric pair associated to the parameters $(n,2,m)$. When n takes all the values between $a + m^a - 2$ and $a + m^{a+1} - 1$, i.e. $n = a + m^a + b - 2$, for all $b \in \{0,\ldots,(m-1)m^a\}$, then*

(i) $h(n,2,m) = m^a + b + 1 = n - a + 1$,
(ii) l *is equal to* m^a,
(iii) k *is equal to* $h(n,2,m)$.

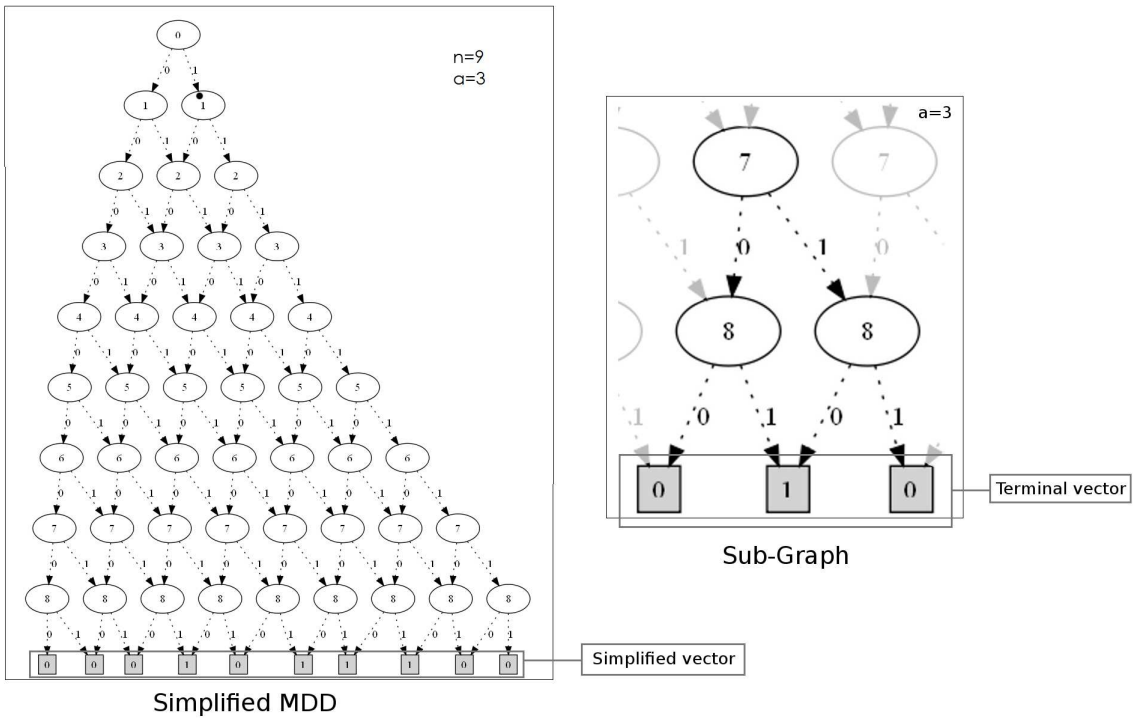


Fig. 1. Simplified MDD and sub-graph of $f \in SM_9(2,2)$. In this case, $a = 3$

Let G_f be the simplified MDD associated to the symmetric function $f \in SM_n(q,m)$. We call *sub-graph* sG_f of G_f any sub-graph of height equals to $n+1-h(n,2,m) = a$ and whose root is of level $h(n,2,m)$ in G_f. We call *terminal vector* into a sub-graph sG_f, the values of the terminal nodes read from left to right. The terminal vector of a subgraph has a length equal to a.

By definition, a simplified MDD has $C^1_{h(n,2,m)+1} = h(n,2,m)+1$ sub-graphs sG_f in the case $q = 2$.

Lemma 2. *A function is hard if and only if the number of its sub-graphs sG_f being not isomorphic is maximum.*

Theorem 3. *Let $n = a + m^a + b - 2$ where $a \geq 0$, $b \in \{0, \ldots, (m-1)m^a\}$. Then a function $f \in SM_n(2,m)$ is hard if and only if in its simplified vector, it appears m^a subsequences of a consecutive different letters.*

Theorem 4. *Let's $f \in HSM_n(2,m), n \geq 1, a \geq 1, b \geq 0$, such that*
$n = a + m^a - 2 + b$ *then* $C_R(f) = m + \sum_{k=0}^{n-1} min(k+1, m^{n+1-k} - m)$

5.2 De Bruijn Sequences and Terminal Vectors

According to Theorem 3, for $n = m^a + a - 2$, $h(n,2,m) + 1 = m^a$, i.e. the simplified MDD of a hard symmetric function has the same number of nodes at the level $h(n,2,m)$ as its QROMDD. So we deduce the following corollary.

Corollary 1. *Let n and a be two positive integers such that $n = a + m^a - 2$ and let $f \in SM_n(2,m)$. Then f is hard if and only if in its simplified vector, it appears exactly m^a distinct subsequences with length equal to a.*

This property on the simplified vector is pertaining to De Bruijn sequences [7]. Let A be an alphabet of m letters. We call a De Bruijn sequence $B(m,a)$ a cyclic sequence such that each subsequence with length equal to a appears exactly once.
A simplified MDD of a symmetric function with $n = m^a + a - 2$ variables contains $m^a + a - 1$ terminal nodes. The following theorem establishes the link between the De Bruijn sequences and the simplified vectors of the hard symmetric functions.

Theorem 5. *Let $n \geq 1$ and $a \geq 0$ be integers such that $n = a + m^a - 2$. Then the simplified value vector of $f \in HSM_n(2, m)$ is a rotation of a De Bruijn sequence $B(m, a)$ at the end of which one the $a - 1$ first letters of the sequence are concatenated.*

Corollary 2. *Let $n \geq 1$ and $a \geq 0$ be integers such that $n = a + m^a - 2$. Then the number of hard symmetric functions of parameters $(n, 2, m)$ is equal to $m!^{m^{a-1}}$.*

Proposition 1. *Let $n \geq 1$ be an integer such that there exists $a \geq 0$ and $b \geq 0$ with $n = a + m^a - 2 + b$. Then the number of hard symmetric functions of parameters $(n, 2, m)$ is greater or equal to: $m!^{m^{a-1}} \times m^b$.*

5.3 Algebraic Degree of Hard Symmetric Boolean Functions

The following theorem establishes the link between the periodicity of the simplified vector of a symmetric Boolean function and its algebraic degree [4].

Theorem 6. *[4] Let $f \in SM_n(2, 2)$, then its simplified value vector, $v_s = (v_s(0), \ldots, v_s(n))$ is periodic with period 2^t, $2^t < n$, if and only if $\deg(f) \leq 2^t - 1$.*

For $n = a + 2^a - 2$, the simplified value vector v_s of a hard symmetric function is periodic with period 2^a, and by properties of De Bruijn sequence this vector cannot be periodic with period 2^k, where $k > a$. Thus, according to the theorem 6 and its contraposition, we obtain the following result.

Theorem 7. *The hard symmetric Boolean functions with $n = a + 2^a - 2$ variables have degree belonging to interval $[2^{a-1}, \ldots, 2^a - 1]$.*

6 Conclusion

We were interested primarily in the multivalued symmetric functions from a set E_q^n to a set E_m and their representations by QROMDD. We initially set out an efficient way to represent these functions by a MDD, then we gave a formula for the exact value of the complexity of the hard symmetric functions. For $q = 2$ (and any m) we could generalize the results concerning the simplified vectors of these hard functions. We highlighted the links between De Bruijn sequences and these simplified vectors from the hard symmetric functions when $n = a + m^a - 2$. We thus could count these hard functions in this particular case. For the general value of n, we stated another link with De Bruijn sequences but we were only able to give a lower bound of the number of functions.
These hard symmetric functions can be a good compromise for a use in cryptography; being symmetric they have a low number of nodes but being hard they also appear among the most robust. We have also shown that their algebraic degree takes interesting values. The further characterization of more detailed cryptographic properties of these functions will be of great interest. The generalization of our results to higher values of q can offer good new functions.

References

1. Andrews G.E.: The theory of partitions. Encyclopedia of Mathematics and Its Applications **2** Addison-Wesley Publishing Company Reading, Mass. (1976).
2. Bryant R.E.: Graph-Based Algorithms for Boolean Function Manipulation. IEEE Transactions on Computers C35 **8** 677–691 (1986) .
3. Butler Jon T., Herscovici David S., Sasao Tsutomu, and Barton Robert J.: "Average and Worst Case Number of Nodes in Decision Diagrams of Symmetric Multiple-Valued Functions", IEEE Transaction on Computers, Vol. 46, N° 4, (April 1997).
4. Canteaut A., Videau M.: Symmetric Boolean Functions. IEEE Transactions on Information Theory **51** N° 8 2791–2811 (2005).
5. Heinrich-Litan L. and Molitor P.: "Least Upper Bounds for the Size of OBDDs Using Symmetry Properties", IEEE Transaction on Computers, Vol. 49, N° 4, 271–281, (April 2000).
6. Mouffron M.: "Balanced Alternating and Symmetric Functions over Finite Sets", Workshop on Boolean Functions Cryptography and Applications, (BFCA08), Copenhagen, Denmark, 27–44, (2008).
7. Rosenfeld V.R.: Enumerating De Bruijn Sequences. MATCH Commun. Math. Comput. Chem. **45** 71-83 (2002).

A APPENDIX: Algebraic Degree of Super Hard and Hard Symmetric Boolean Functions

n	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	2	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	4	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4		8	6	4	2	4	8	8	0	0	0	0	0	0	0	0	0	0	0	0	0
5			4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6				8	6	12	20	16	0	0	0	0	0	0	0	0	0	0	0	0	0
7					4	8	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8						32	30	20	20	20	16	12	8	4	8	32	32	0	0	0	0
9							64	46	28	14	4	0	0	0	0	0	0	0	0	0	0
10								104	64	32	14	0	0	0	0	0	0	0	0	0	0
11									140	92	60	40	28	20	40	136	128	0	0	0	0
12										136	46	24	12	4	8	32	32	0	0	0	0
13											148	72	32	16	40	192	192	0	0	0	0
14												116	48	20	48	200	192	0	0	0	0
15													96	32	64	32	0	0	0	0	0
16														128	336	448	444	372	324	328	296
17															608	720	772	736	644	516	340
18																1408	1540	1560	1264	916	666
19																	3560	3378	2856	2308	1664
20																		6688	5856	4804	3584
21																			10368	7806	5464
22																				16896	12340
23																					24324
TOTAL	8	14	14	12	12	56	126	194	252	294	288	264	224	224	1152	3200	6892	12734	21312	33574	48678

Table 1. Algebraic degree of Super Hard Symmetric Functions for $n = 3, \ldots, 23$

n	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	4	4	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3		4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4			0	2	4	2	0	0	0	0	0	0	0	0	0	0	0	0	0
5				12	10	4	0	0	0	0	0	0	0	0	0	0	0	0	0
6					24	22	16	16	16	0	0	0	0	0	0	0	0	0	0
7						20	8	0	0	0	0	0	0	0	0	0	0	0	0
8							16	0	4	12	16	12	12	8	4	0	0	0	0
9								0	12	28	30	24	8	4	0	0	0	0	0
10									48	66	64	44	20	10	4	0	0	0	0
11										104	100	100	92	84	84	64	64	64	0
12											192	136	88	42	8	0	0	0	0
13												328	236	148	88	64	64	64	0
14													396	248	164	128	128	128	0
15														384	144	64	0	0	0
16															416	128	0	16	44
17																256	0	64	184
18																	0	208	532
19																		800	1128
20																			2056
TOTAL	6	8	4	18	38	48	40	16	80	210	402	644	852	928	912	704	256	1344	3944

Table 2. Algebraic degree of Hard Symmetric Functions for $n = 2, \ldots, 20$

B APPENDIX: Complexities of Symmetric Functions from E_2^n

n	$max\ c_R(f)$	Number of super hard functions	$max\ c(f)$	Number of hard functions
1	3	2	3	2
2	5	2	5	6
3	7	8	8	8
4	10	14	12	4
5	14	14	16	18
6	19	12	21	38
7	25	12	27	48
8	31	56	34	40
9	38	126	42	16
10	46	194	50	80
11	55	252	59	210
12	65	294	69	402
13	76	288	80	644
14	88	264	92	852
15	101	224	105	928
16	115	224	119	912
17	129	1152	134	704
18	144	3200	150	256
19	160	6892	166	1344
20	177	12734	183	3944
21	195	21312	201	9276
22	214	33574	220	To be completed
23	234	48678	240	To be completed
24	255	65040	261	To be completed
25	277	81348	283	To be completed
26	300	94376	306	To be completed
27	324	103944	330	To be completed
28	349	107744	355	To be completed
29	375	99744	381	To be completed
30	402	95232	408	To be completed
31	430	81408	436	To be completed
32	459	61440	465	To be completed
33	489	61440	495	To be completed

Table 3. Complexity of Symmetric Functions from E_2^n