A Distinguisher for High Rate McEliece Cryptosystems

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 paper is to study the difficulty of the so-called Goppa Code Distinguishing (GD) problem introduced by Courtois, Finiasz and Sendrier in Asiacrypt 2001. GD 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. 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 McEliece cryptosystems. In this paper, we present an efficient distinguisher for alternant and Goppa codes of high rate over binary/non binary fields. Our distinguisher is based on a recent algebraic attack against compact variants of 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 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. We believe that this work permits to shed some light on the choice of secure parameters for McEliece cryptosystems; a topic thoroughly investigated recently. Our technique permits to indeed distinguish a public key of the CFS signature scheme for all parameters proposed by Finiasz and Sendrier at Asiacrypt 2009. Moreover, some realistic parameters of McEliece scheme also fit in the range of validity of our distinguisher.

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

1 Introduction

Code-based public key cryptography appeared with McEliece's pioneering work [23] where the author proposed to use 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, 28]. A binary Goppa code is defined by a polynomial Γ of degree $r \geq 1$ with coefficients in some extension field $\mathbb{F}_{2^m}$ 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 a randomly picked Γ which together with $\mathscr{L}$ provide all the information to decode efficiently. The public key is a generator matrix of a randomly 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 [26] 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 secure as McEliece's one. The first code-based signature scheme came out in [12] almost twenty years McEliece's proposal. The only difference between the encryption and the signature scheme lies in the choice of the parameters of the binary Goppa codes. For signature, Goppa codes 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 $k = n - rm$ where by definition r is the number of errors and generally n is chosen to be equal to 2^m. For instance according to [18], an 80-bit security signature scheme imposes $r = 10$ and $m = 21$ which leads to $R = 0.9999$.

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 [19, 20, 31, 10, 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 [21], the only known key-recovery attack is the exhaustive search of the secret polynomial Γ of the Goppa code, and applying the *Support Splitting Algorithm* (SSA) [29] 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 McEliece's proposal of using binary Goppa codes, one should not exclude the possibility of breakthrough in that field. The authors of [12] alleviated the McEliece assumptions by introducing the *Goppa Code Distinguishing (GD) problem*. They assume that no polynomial time algorithm exists that distinguishes a generator matrix of a Goppa code from a random generator matrix. This is a classical belief in code-based cryptography. For instance, according to [12], 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 [29], as explained below. The time complexity of this method is $\mathscr{O}(2^{mr})$ assuming 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 [27, 15, 8], the security in the random oracle model against existential forgery [12, 13] of the CFS signature [12] scheme, the provable security of several primitives such as a threshold ring signatures scheme [14], an identity-based identification scheme [11], 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 $\mathscr{D}$ *is an algorithm that takes as input a matrix* $\mathbf{G}$ *and returns a bit. We say that $\mathscr{D}$ solves the GD problem if it wins the following game:*

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

The probability that $\mathscr{D}$ outputs 1 *when* $\mathbf{G}$ *is chosen as a random binary generator matrix of a Goppa code is denoted by* $\Pr[\mathbf{G} \stackrel{R}{\leftarrow} \mathsf{Random}(n,k) : \mathscr{D}(\mathbf{G}) = 1]$ *and the probability that it outputs* 1 *when* $\mathbf{G}$ *is chosen randomly in* $\mathsf{Random}(n,k)$ *is denoted by* $\Pr[\mathbf{G} \stackrel{R}{\leftarrow} \mathsf{Random}(n,k) : \mathscr{D}(\mathbf{G}) = 1]$. *We define the advantage of a distinguisher $\mathscr{D}$ as:*

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

In this paper, we present a deterministic polynomial-time distinguisher for solving the GD problem defined below with advantage close to 1 for codes of high rate. Along the way, we also solve the code distinguishing problem for alternant codes. The key ingredient is a new algebraic technique introduced in [17] to attack two variants [1, 24] of McEliece. It has been observed [17] that a key recovery attack against these cryptosystems, as well as the genuine McEliece's system, can be reduced to solving the following algebraic set of equations:

$$\left\{ g_{i,1}Y_1X_1^j + \cdots + g_{i,n}Y_nX_n^j = 0 \;\middle|\; i \in \{1,\ldots,k\}, j \in \{0,\ldots,r-1\} \right\} \tag{1}$$

where the unknowns are the X_i's and the Y_i's and the $g_{i,j}$'s are known coefficients (with $1 \le i \le k, 1 \le j \le n$) which are nothing but the coefficients 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, 24], additional structures permit to drastically reduce the number of variables and solve (1) efficiently using dedicated Gröbner bases techniques [17]. For McEliece's cryptosystem, solving (1) seems to be out of the scope of such dedicated techniques.

However, this algebraic approach can be used to construct an efficient *distinguisher*. To do so, we consider the dimension of the solution space of a linear system deduced from (1). This linear system is obtained by linearization of the algebraic system (1). Linearization introduces many new unknowns. Consequently, this strategy makes sense if the number of equations k is greater than the number of newly introduced unknowns. This is for instance the case for the parameters proposed in CFS [12] but 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 codes and Goppa codes over any field. Note that the distinguisher is efficient since we only have to compute the rank of a linear system. Additionally, the distinguisher is highly discriminant. We provide in Section 5 explicit formulas for "generic" random, alternant, and Goppa code over any alphabet. We performed extensive experiments to compare our theoretical results on valid McEliece public keys. They confirm that the generic formula are accurate. 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 [27, 15, 8, 12–14, 11]. 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. Although our attack remains theoretical, we believe that this work also permits to shed some light on the choice of secure parameters for McEliece cryptosystems; a topic thoroughly investigated recently [6, 7, 25, 18]. Our technique permits to indeed distinguish a public key of the CFS signature scheme for all parameters proposed by Finiasz and Sendrier [18]. Moreover, some realistic parameters of McEliece scheme also fit in the range of validity of our distinguisher like a binary Goppa code of length $n = 2^{13}$ that corrects $r = 19$ errors. Fot these parameters, the scheme has a 90-bit security.

Organisation of Paper. In Section 2, we briefly recall the McEliece public-key cryptosystem as well as the Courtois-Finiasz-Sendrier CFS signature [12]. In Section 3, we recall several key features of Goppa and alternant codes. In Section 4, we precisely explain how we can mount an algebraic cryptanalysis against McEliece-like schemes *i.e.* namely how the algebraic system (1) is constructed. The distinguisher is presented in Section 5. Section 6 deals with the consequences of the existence of a distinguisher in code-based cryptography. Finally, in Section 7 we explain how the formulas used in Section 5 have been obtained. To do so, we use together combinatorial properties of the linearized system and distinguishing features of Alternant/Goppa codes.

2 Code-Based Public-Key Cryptography

The main cryptographic primitives in code-based public-key cryptography are the McEliece encryption and the CFS signature [12]. We recall that a linear *code* over a finite field $\mathbb{F}_q$ of q elements defined by a $k \times n$ matrix G (with $k \le n$) over $\mathbb{F}_q$ is the vector space $\mathscr{C}$ spanned by its rows *i.e.* $\mathscr{C} \stackrel{\text{def}}{=} \left\{ uG \mid u \in \mathbb{F}_q^k \right\}$. G is chosen as a full-rank matrix, so that the code is of dimension k. The *rate* of the code is given by the ratio $\frac{k}{n}$. Code-based public-key cryptography focuses on linear codes that have a polynomial time decoding algorithm. The role of decoding algorithms is to correct errors of prescribed weight. We say that a decoding algorithm corrects t errors if it recovers u from the knowledge of $uG+e$ for all possible $e \in \mathbb{F}_q^n$ of weight at most t.

Secret key: the triplet $(S, G_s, \mathbf{P})$ of matrices defined over a finite field $\mathbb{F}_q$ over q elements, with q being a power of two, that is $q = 2^s$. G_s is a full rank matrix of size $k \times n$, with $k < n$, S is of size $k \times k$ and is invertible. $\mathbf{P}$ is a permutation matrix of size $n \times n$. G_s is chosen in such a way that its associated linear code (that is the set of all possible uG_s with u ranging over $\mathbb{F}_q^k$) has a decoding algorithm which corrects in polynomial time t errors.

Public key: the matrix $G = SG_s\mathbf{P}$.

Encryption: A plaintext $u \in \mathbb{F}_q^k$ is encrypted by choosing a random vector e in $\mathbb{F}_q^n$ of weight at most t. The corresponding ciphertext is $\mathbf{c} = uG + e$.

Decryption: $\mathbf{c}' = \mathbf{c}\mathbf{P}^{-1}$ is computed from the ciphertext $\mathbf{c}$. Notice that $\mathbf{c}' = (uSG_s\mathbf{P} + e)\mathbf{P}^{-1} = uSG_s + e\mathbf{P}^{-1}$ and that $e\mathbf{P}^{-1}$ is of Hamming weight at most t. Therefore the aforementioned decoding algorithm can recover in polynomial time uS and therefore the plaintext u by multiplication by S^{-1}.

What is generally referred to as the McEliece cryptosystem is this scheme together with a particular choice of the code, which consists in taking a binary Goppa code. This class of codes belongs to a more general class of codes (see Section 3, namely the alternant code family ([22, Chap. 12, p. 365]). The main feature of this last class of codes is that they can be decoded in polynomial time.

Another important code-based cryptographic primitive is the CFS scheme [12], which is the first signature scheme based on the security of the McEliece cryptosystem. In this kind of scheme, a user whose public key is G and who wishes to sign a message $\mathbf{x} \in \mathbb{F}_2^k$ has to compute a string u such that the Hamming weight of $\mathbf{x} - uG$ is at most t. Anyone (a *verifier*) can publicly check the validity of a signature. Unfortunately, this approach can only provide signatures for messages $\mathbf{x}$ that are within distance t from a codeword uG. The CFS scheme suggests to modify the message by appending a counter incremented until the decoding algorithm can find such a signature. The efficiency of this scheme heavily depends on the number of trials. It is suggested in [12] to choose as in the McEliece cryptosystem, binary Goppa codes for this purpose with the following parameters $n = 2^m$ and $k = n - mt$. The number of trials is of order $t!$ in this case, which leads to choose a very small t and therefore to take a very large n in order to be secure. Notice that the code rate is then equal to $\frac{2^m - tm}{2^m} = 1 - \frac{mt}{2^m}$ which is for large n (that is for large values of 2^m) and moderate values of t quite close to 1. Thus, the major difference between the McEliece cryptosystem and the CFS scheme lies in the choice of the parameters. An 80-bit security CFS scheme requires $n = 2^{21}$ and $t = 10$ whereas the McEliece cryptosystem for the same security needs $n = 2^{11}$ and $t = 32$ ([18]). The code of the CFS scheme is of rate $1 - \frac{10 \times 21}{2^{21}} \approx 0.9999$. We see here that the CFS scheme depends on very high rate binary Goppa codes.

3 Basic Facts about Alternant and Goppa Codes

As explained in the previous section, the McEliece cryptosystem relies on Goppa codes which belong to the class of *alternant codes* and inherit an efficient decoding algorithm from this. It is convenient to describe this class through a *parity-check matrix* over an extension field $\mathbb{F}_{q^m}$ of $\mathbb{F}_q$ over which the code is defined. In other words, the parity check matrix is an $r \times n$ matrix $\mathbf{H}$ with coefficients in $\mathbb{F}_{q^m}$ and the associated alternant code $\mathscr{A}$ is the set of vectors of $\mathbb{F}_q^n$ which belong to the right kernel of $\mathbf{H}$, i.e.

$$\mathscr{A} = \{\mathbf{c} \in \mathbb{F}_q^n \mid \mathbf{H}\mathbf{c}^T = \mathbf{0}\}. \tag{2}$$

r satisfies in this case the condition $r \geq \frac{n-k}{m}$ where k is the dimension of $\mathscr{A}$. For alternant codes, there exists a parity-check matrix with a very special form related to Vandermonde matrices. For reasons which will be made clear in Section 4, it will be convenient to work with the projective plane $\overline{\mathbb{F}}_{q^m} \stackrel{\text{def}}{=} \mathbb{F}_{q^m} \cup \{\infty\}$ and to consider the class of *projective alternant codes* (which are slightly more general than classical alternant codes). More precisely, any projective alternant code has a parity check matrix which is of the form

$$V_r(\mathbf{x},\mathbf{y}) \stackrel{\text{def}}{=} \begin{pmatrix} y_1 & \cdots & y_n \\ y_1x_1 & \cdots & y_nx_n \\ \vdots & & \vdots \\ y_1x_1^{r-1} & \cdots & y_nx_n^{r-1} \end{pmatrix}. \tag{3}$$

where $\mathbf{x} = (x_1,\dots,x_n) \in (\overline{\mathbb{F}}_{q^m})^n$ and $\mathbf{y} = (y_1,\dots,y_n)$ in $(\mathbb{F}_{q^m})^n$. When $x_i = \infty$ we use the convention that the i-th column of $V_r(\mathbf{x},\mathbf{y})$ is equal to $\begin{pmatrix} 0 \\ \vdots \\ 0 \\ y_i \end{pmatrix}$.

Definition 2 (Projective and classical alternant code). *The projective alternant code of order r over $\mathbb{F}_q$ associated to $\mathbf{x} = (x_1,\dots,x_n) \in (\overline{\mathbb{F}}_{q^m})^n$ (where all x_i's are distinct) and $\mathbf{y} = (y_1,\dots,y_n) \in \left(\mathbb{F}^*_{q^m}\right)^n$, denoted by $\mathscr{A}_r(\mathbf{x},\mathbf{y})$, is defined by*

$$\mathscr{A}_r(\mathbf{x},\mathbf{y}) = \{\mathbf{c} \in \mathbb{F}_q^n | V_r(\mathbf{x},\mathbf{y})\mathbf{c}^T = \mathbf{0}\}. \tag{4}$$

A classical alternant code corresponds to the case where all x_i's are different from ∞.

The class of Goppa codes is a subfamily of alternant codes which are given by:

Definition 3 (Projective and classical Goppa codes). *The projective Goppa code $\mathscr{G}(\mathbf{x},\Gamma)$ over $\mathbb{F}_q$ associated to a polynomial $\Gamma(x)$ of degree r over $\mathbb{F}_{q^m}$ and a certain n-tuple $\mathbf{x} = (x_1,\dots,x_n)$ of distinct elements of $\overline{\mathbb{F}}_{q^m}$ satisfying $\Gamma(x_i) \neq 0$ for[4] all $i, 1 \le i \le n$, is the alternant code $\mathscr{A}_r(\mathbf{x},\mathbf{y})$ of order r with y_i being defined by $y_i = \Gamma(x_i)^{-1}$. A classical Goppa code corresponds to the case $x_i \in \mathbb{F}_{q^m}$ for all i in $\{1,\dots,n\}$.*

It should be noted that the public code in the McEliece cryptosystem is also an alternant code. This is a simple consequence of the fact that $\{uSG_s\mathbf{P} \mid u \in \mathbb{F}_q^k\}$ is obtained from the secret code $\{uG_s \mid u \in \mathbb{F}_q^k\}$ by permuting the coordinates in it with the help of $\mathbf{P}$, since multiplying by an invertible matrix S of size $k \times k$ leaves the code globally invariant.

4 Algebraic Cryptanalysis of McEliece-like Cryptosystems

In this part, we explain more precisely how we construct the algebraic system described in (1). This algebraic system is the main ingredient of the distinguisher. We recall a key feature of alternant codes.

Fact 1. *There exists a polynomial time algorithm decoding all errors of Hamming weight at most $\frac{r}{2}$ for an alternant code of order r once a parity-check matrix $\mathbf{H}$ of the form $\mathbf{H} = V_r(\mathbf{x},\mathbf{y})$ is given for it.*

The variants of McEliece's cryptosystem based on general alternant codes or on non binary Goppa codes, such as [1, 24] for instance, add errors which are of weight smaller than or equal to $r/2$. In this case, it is possible to break these variants by finding $\mathbf{x}^*$ and $\mathbf{y}^*$ in $\mathbb{F}_{q^m}^n$ such that:

$$\{\mathbf{x}G \mid \mathbf{x} \in \mathbb{F}_q^r\} = \{\mathbf{y} \in \mathbb{F}_q^n \mid V_r(\mathbf{x}^*,\mathbf{y}^*)\mathbf{y}^T = \mathbf{0}\}. \tag{5}$$

According to Fact 1, the knowledge of $V_r(x^*,y^*)$ permits to efficiently decode the public code, *i.e.* to recover u from $uG+e$. By the very definition of the public code G, we have:

$$V_r(x^*,y^*)G^T = 0.$$

[4] We define $\Gamma(\infty) \stackrel{\text{def}}{=} \gamma_r$ for $\Gamma(X) = \sum_{i=0}^r \gamma_i X^i$.

This is the key observation of the algebraic approach used in [17] to cryptanalyze dyadic and quasi-cyclic variants of McEliece. Let $X_1,\dots,X_n$ and $Y_1,\dots,Y_n$ be $2n$ variables corresponding to the x_i^*'s and the y_i^*'s. Observe that such x_i^*'s and y_i^*'s are a particular solution of the following system:

$$\left\{ g_{i,1}Y_1X_1^j+\cdots+g_{i,n}Y_nX_n^j=0 \;\middle|\; i\in\{1,\dots,k\}, j\in\{0,\dots,r-1\} \right\} \tag{6}$$

where the $g_{i,j}$'s are the entries of the known matrix G. In the cases of [1, 24], additional structures permit to drastically reduce the number of variables allowing to solve (1) efficiently using dedicated Gröbner bases techniques [17].

For binary Goppa codes, it is essential to recover its description as a Goppa code and not only the x_i's and the y_i's giving its description as an alternant code. This is a consequence of the following result.

Fact 2. *[28] There exists a polynomial time algorithm decoding all errors of Hamming weight at most r in a Goppa code $\mathscr{G}(\mathbf{x},\Gamma)$ when Γ has degree r and has no multiple roots, if $\mathbf{x}$ and Γ are known.*

If we recover only the x_i's and the y_i's we can decode only $r/2$ errors. The issue is now, once a possible description of a Goppa code has been found as an alternant code, that is once a solution $\mathbf{x}=(x_i)_{1\le i\le n}$ and $\mathbf{y}=(y_i)_{1\le i\le n}$ of the system (6) has been found, does there exist a polynomial $\Gamma(X)$ of degree r such that $y_i=\Gamma(x_i)^{-1}$ for all $i\in\{1,\dots,n\}$? If such a polynomial exists, it can be easily found by interpolation. The problem is that a Goppa code has multiple descriptions as an alternant code, *i.e.*, there are several $\mathbf{x},\mathbf{y}$'s for which $\mathscr{G}=\mathscr{A}_r(\mathbf{x},\mathbf{y})$. The solutions we are interested in are the ones for which $y_i=\Gamma(x_i)^{-1}$ for all i, and for some polynomial Γ of degree r.

This raises the fundamental issue of finding all possible descriptions of the form (4) of an alternant code $\mathscr{A}$, that is find all $\mathbf{x},\mathbf{y}$'s such that $\mathscr{A}=\mathscr{A}_r(\mathbf{x},\mathbf{y})$. When the extension field $\mathbb{F}_{q^m}$ is the same as the definition[5] field $\mathbb{F}_q$, *i.e.* if $m=1$, the problem was solved in [16]. This was the key of the cryptanalysis of McEliece's variant based on generalized Reed-Solomon codes [30].

The general case is still unsolved. However, the results of [16] basically show that the we have at least one degree of freedom for Y_i and three degrees of freedom for the X_i's in the system (6). First of all it is straightforward to notice that if $(X_i)_{1\le i\le n},(Y_i)_{1\le i\le n}$ is a solution of the algebraic Equation (6) then $(\alpha X_i)_{1\le i\le n},(\beta Y_i)_{1\le i\le n}$ is also a solution for any α,β in $\mathbb{F}_{q^m}$. Therefore, we can specialize one (X_i,Y_i) arbitrarily. It turns out we can fix more variables thanks to the following proposition.

Proposition 1. *Let $\mathbf{x}=(x_i)_{1\le i\le n}\in(\overline{\mathbb{F}}_{q^m})^n$ be an n-tuple formed by distinct elements and let $\mathbf{y}=(y_i)_{1\le i\le n}\in(\mathbb{F}_{q^m})^n$ be an n-tuple of nonzero elements. Let a,b,c,d be elements of $\mathbb{F}_{q^m}$ such that that $ad-bc\neq 0$. Then*

$$\mathscr{A}_r\left(\frac{a\mathbf{x}+b}{c\mathbf{x}+d},\mathbf{y}'\right)=\mathscr{A}_r(\mathbf{x},\mathbf{y}),\ \textit{where}$$

$$\frac{a\mathbf{x}+b}{c\mathbf{x}+d}\stackrel{def}{=}(x_i')_{1\le i\le n}\ \textit{with}\ x_i'=\frac{ax_i+b}{cx_i+d},\ \ \mathbf{y}'=(y_i')_{1\le i\le n}\ \textit{with}\ y_i'=y_i(cx_i+d)^{r-1}.$$

Remark 1. The proof is in Appendix A. Notice that either x_i or x_i' might be infinite. We used here the usual rules to evaluate the homography $z\mapsto\frac{az+b}{cz+d}$, namely $\frac{\alpha}{0}=\infty$, $\frac{\infty}{\alpha}=\infty$, $\frac{\alpha}{\infty}=0$, $\beta+\infty=\infty$, $0\times\infty=0$, $\frac{a\times\infty+b}{c\times\infty+d}=\frac{a}{c}$, where $\alpha\neq 0,\beta$ belong to $\mathbb{F}_{q^m}$.

This result explains that there is (at least) one degree of freedom for the Y_i's and three degrees of freedom for the X_i's. It is quite helpful to allow here x_i which can be infinite since even all of them are in $\mathbb{F}_{q^m}$, it might happen that cx_i+d is equal to zero. Therefore the corresponding image by the homography will be infinite. Finally, since the set of homographies acts 3-transitively over $\mathbb{F}_{q^m}\cup\{\infty\}$, we have:

[5] This means that the resulting code is a slight generalization of a generalized Reed-Solomon code known under the name of a Cauchy code.

Corollary 1. *We can specialize (almost) randomly one Y_i and three X_i's in* (1)*. As long as the X_i's are distinct, we still have a non-empty set of solutions for such modified system* (1)*.*

At first glance, the degree of freedom should be less for Goppa codes. Indeed, there is an additional crucial constraint for binary Goppa codes: a solution must verify $Y_i = \Gamma(X_i)^{-1}$ for a certain polynomial of degree r. Surprisingly, we can keep the same degree of freedom by considering a slight change of (6). Let $\tilde{\mathscr{G}}(\mathbf{x},\Gamma)$ be the subcode of the Goppa code $\mathscr{G}(\mathbf{x},\Gamma)$ formed by all codewords of even Hamming weight. Let $\tilde{G} = (\tilde{g}_{i,j})_{\substack{1\le i\le \tilde{k}\\ 1\le j\le n}}$ be a generator matrix of $\tilde{\mathscr{G}}(\mathbf{x},\Gamma)$, that is a matrix of full rank whose rows generate $\tilde{\mathscr{G}}(\mathbf{x},\Gamma)$. The dimension $\tilde{k}$ of this subspace is either k or $k-1$, where k is the dimension of the Goppa code $\mathscr{G}(\mathbf{x},\Gamma)$. This subcode is itself an alternant code.

Proposition 2. *[2] It holds that:*

$$\tilde{\mathscr{G}}(\mathbf{x},\Gamma) = \mathscr{A}_{r+1}(\mathbf{x},\mathbf{y})$$

for $\deg(\Gamma) = r$ *and where* $\mathbf{y} = (y_i)_i$ *with* $y_i = \Gamma(x_i)^{-1}$.

This implies that the x_i's and y_i's are a particular solution of:

$$\left\{ \tilde{g}_{i,1}Y_1X_1^j + \cdots + \tilde{g}_{i,n}Y_nX_n^j = 0 \;\middle|\; i \in \{1,\ldots,\tilde{k}\}, j \in \{0,\ldots,r\} \right\} \tag{7}$$

where the $\tilde{g}_{i,j}$'s are the entries of the known matrix $\tilde{G}$. Notice that this system is very similar to (6) with the exception that the powers of the X_i's can now be equal to r. The crucial result is now that

Proposition 3. *[2] Let $\mathbf{x} = (x_i)_{1\le i\le n}$ be an n-tuple of distinct elements of $\overline{\mathbb{F}}_{q^m}$ and Γ be a polynomial of degree r such that $\Gamma(x_i) \neq 0$ for all $i \in \{1,\ldots,n\}$. Let $\psi(z) = \frac{az+b}{cz+d}$ be an homography with $ad-bc \neq 0$ and $a,b,c,d \in \mathbb{F}_{q^m}$. Let $\mathbf{x}^\psi \stackrel{def}{=} (x_i^\psi)_{1\le i\le n}$ with $x_i^\psi \stackrel{def}{=} \psi^{-1}(x_i)$, $\Gamma^\psi(X) \stackrel{def}{=} (cx+d)^r\Gamma(\psi(x)) = \sum_{i=0}^r \gamma_i(aX+b)^i(cx+d)^{r-i}$, for $\Gamma(x) = \sum_{i=0}^r \gamma_i X^i$. Then*

$$\tilde{\mathscr{G}}(\mathbf{x},\Gamma) = \tilde{\mathscr{G}}(\mathbf{x}^\psi,\Gamma^\psi).$$

Once again, we can use that homographies have a 3-transitive action on $\overline{\mathbb{F}}_{q^m}$.

Corollary 2. *We can specialize in* (7) *one of the Y_i and three of the X_i's almost arbitrarily (with $Y_i \neq 0$ and such that the three X_i's are distinct) and still obtain a solution for which there exists a polynomial Γ of degree r such that $Y_i = \Gamma(X_i)^{-1}$ for all i in $\{1,\ldots,n\}$.*

To finish this discussion, it will be helpful to notice that in the case of binary Goppa codes, we have even more algebraic equations than the ones given in System (6). The starting point is the following result, which is essentially derived from a discussion in a paragraph about Goppa codes in [22, p.341].

Theorem 3. *A binary Goppa code $\mathscr{G}(\mathbf{x},\Gamma)$ associated to a Goppa polynomial $\Gamma(X)$ of degree r without multiple roots is equal to the alternant code $\mathscr{A}_{2r}(\mathbf{x},\mathbf{y})$, with $y_i = \Gamma(x_i)^{-2}$.*

In other words, $\mathbf{x}$ and $\mathbf{y}$ are solutions of the following algebraic system

$$\left\{ g_{i,1}Y_1X_1^j + \cdots + g_{i,n}Y_nX_n^j = 0 \;\middle|\; i \in \{1,\ldots,k\}, j \in \{0,\ldots,2r-1\} \right\}, \tag{8}$$

where (g_{il}) is a generator matrix of the Goppa code. Notice that the powers j are now in the range $\{0,1,\ldots,2r-1\}$ and not in $\{0,1,\ldots,r-1\}$, as was the case before.

Fig. 1. Systematic form of $\mathbf{G}$

5 A Distinguisher of Alternant and Goppa Codes

We present in this part the algebraic distinguisher. Let $\mathbf{G} = (g_{ij})_{\substack{1\le i\le k\\1\le j\le n}}$ be a generator matrix of the public code. We can assume without loss of generality that $\mathbf{G}$ is systematic in its k first positions. Such a form can be easily obtained by Gaussian elimination and by a suitable permutation of the columns. We describe now a simple way of using this particular form for solving (6). We assume that the rate of the public code is close to 1, i.e. $\frac{n-mr}{n} \approx 1$, which implies $mr \ll n$. From a cryptographic point of view, this means that the expansion ratio between the size of the ciphertext and the size of the message is close to 1. This kind of rate has been proposed in [18]. The strategy is as follows.

5.1 First step – expressing the $Y_iX_i^d$'s in terms of the $Y_jX_j^d$'s for $j \in \{k+1,\dots,n\}$.

Let $\mathbf{P} = (p_{ij})_{\substack{1\le i\le k\\k+1\le j\le n}}$ be the submatrix of $\mathbf{G}$ formed by its last mr columns (as in Figure 1). We can rewrite (6) as

$$\begin{cases} Y_i &=& \sum_{j=k+1}^n p_{i,j}Y_j \\ Y_iX_i &=& \sum_{j=k+1}^n p_{i,j}Y_jX_j \\ & \dots & \\ Y_iX_i^{r-1} &=& \sum_{j=k+1}^n p_{i,j}Y_jX_j^{r-1} \end{cases} \tag{9}$$

for all $i \in \{1,\dots,k\}$.

5.2 Second step – using the trivial identity $Y_iY_iX_i^2 = (Y_iX_i)^2$ and linearization.

Thanks to the trivial identity $Y_iY_iX_i^2 = (Y_iX_i)^2$ for all i in $\{1,\dots,k\}$, we get:

$$\sum_{j=k+1}^{n} p_{i,j}Y_j \sum_{j=k+1}^{n} p_{i,j}Y_jX_j^2 = \left(\sum_{j=k+1}^{n} p_{i,j}Y_jX_j\right)^2, \text{for all } i \in \{1,\dots,k\}.$$

It is possible to reorder this a little bit to obtain the following equations:

$$\sum_{j=k+1}^{n} \sum_{j'>j} p_{i,j}p_{i,j'}\left(Y_jY_{j'}X_{j'}^2 + Y_{j'}Y_jX_j^2\right) = 0 \tag{10}$$

We can now linearize this system by letting $Z_{jj'} \stackrel{\text{def}}{=} Y_jY_{j'}X_{j'}^2 + Y_{j'}Y_jX_j^2$. We obtain k linear equations involving the $Z_{jj'}$'s:

$$\left\{ \sum_{j=k+1}^{n} \sum_{j'>j} p_{i,j}p_{i,j'}Z_{jj'} = 0, i = 1\dots k \right\}. \tag{11}$$

To solve this system it is necessary that the number of equations is greater than the number of unknowns, *i.e.*:

$$k \ge \binom{mr}{2}$$

This approach works for alternant codes in general. However, for Goppa codes, it will be interesting to consider also a related system. It is obtained by applying the same approach described before but to the generator matrix $\tilde{\mathbf{G}}$ of the subcode of the public code consisting in codewords of even Hamming weight. The reason which makes this new system interesting will be explained in Subsection 7.2, it is related to Proposition 2. We denote by $\tilde{k}$ the dimension of this code. We have either $\tilde{k} = k$ or $\tilde{k} = k-1$.

As previously, we can suppose that $\tilde{\mathbf{G}}$ is in systematic form: $\tilde{\mathbf{G}} = \left(\tilde{\mathbf{I}}|\tilde{P}\right)$ where $\mathbf{I}$ is the identity matrix of size k or $k-1$ (depending on the dimension of the subcode). Finally, let $\tilde{p}_{ij}$ be the coefficient in the i-th row and j-th column of $\tilde{P}$. We can proceed similarly and obtain a new linear system of equations:

$$\left\{ \sum_{j=\tilde{k}+1}^{n} \sum_{j'>j} \tilde{p}_{i,j} \tilde{p}_{i,j'} Z_{jj'} = 0, i = 1 \dots \tilde{k} \right\}. \tag{12}$$

When $\tilde{k} = k-1$, the number of equations is smaller. It might be $k-1$ instead of k and the number of variables is also larger. It is equal to $\binom{n-\tilde{k}}{2} = \binom{mr+1}{2}$. However, we will see that due to Proposition 2, this system has also nice properties in the Goppa case.

5.3 Experimental behavior

Observe that the linear systems (11) and (12) have coefficients in $\mathbb{F}_q$ whereas solutions are sought in the extension field $\mathbb{F}_{q^m}$. In addition, the freedom of choosing three X_i's and one Y_i in order to reduce the number of unknowns in the linearized systems is not used. However, even if this additional knowledge is taken into account, the rank of the linear systems remains insufficient to solve the system. More precisely, the problem is that the dimension of the vector space solution of (11) is amazingly large. It even depends on whether or not the code with generator matrix G is chosen as a (generic) alternant code or as a Goppa code. Interestingly enough, when $\mathbf{G}$ is chosen at random, the dimension of the solution space is typically 0 when k is larger than the number of variables. Although these facts are an obstacle to break the McEliece cryptosystem, it can be used to distinguish the public generator from a random code. Let us denote by:

- $N \stackrel{\text{def}}{=} \binom{mr}{2}$ the number of variables in (11), $\tilde{N}$ the number of variables of (12),
- D_{random}, respectively $\tilde{D}_{\text{random}}$, the dimension of the vector space solution of (11), respectively (12) when the p_{ij}'s are chosen uniformly at random in $\mathbb{F}_q$,
- $D_{\text{alternant}}$, respectively $\tilde{D}_{\text{alternant}}$, the dimension of the vector space solution of (11), respectively (12) when $\mathbf{G}$ is chosen as a generator matrix of a random alternant code of degree r,
- D_{Goppa}, respectively $\tilde{D}_{\text{Goppa}}$ the dimension of the vector space solution of (11), respectively (12) when $\mathbf{G}$ is chosen as a generator matrix of a random Goppa code of degree r.

A thorough experimental study revealed that the dimension of the vector space over $\mathbb{F}_q$ of the solutions of (11) follows typically the following formulas:

Experimental fact 1 *Let D be in $\{D_{alternant}, \tilde{D}_{alternant}, D_{Goppa}, \tilde{D}_{Goppa}\}$. With very high probability and as long as $N - D < k$, the dimension D has the following value:*

$$D_{alternant} = \frac{m(r-1)}{2}\left((2\ell+1)r - 2\frac{q^{\ell+1}-1}{q-1}\right) \text{ for } \ell \stackrel{def}{=} \lfloor \log_q(r-1) \rfloor \tag{13}$$

$$\tilde{D}_{alternant} = D_{alternant} \text{ for } q > 2 \tag{14}$$

For $r < q-1$, it holds that

$$D_{Goppa} = \frac{m(r-1)(r-2)}{2} = D_{alternant} \tag{15}$$

$$\tilde{D}_{Goppa} = \frac{mr(r-1)}{2} \tag{16}$$

wheras for $r \geq q-1$, by denoting by ℓ the unique integer such that $q^\ell - 2q^{\ell-1} + q^{\ell-2} < r \leq q^{\ell+1} - 2q^\ell + q^{\ell-1}$, we obtain

$$D_{Goppa} = \frac{mr}{2}\left((2\ell+1)r - 2q^\ell + 2q^{\ell-1} - 1\right) \tag{17}$$

$$\tilde{D}_{Goppa} = \frac{mr}{2}\left((2\ell+1)r - 2q^\ell + 2q^{\ell-1} + 1\right) \tag{18}$$

We gathered samples of results we obtained through intensive computations with the Magma system [9] in order to confirm the formulas. We randomly generated alternant and Goppa codes over the field $\mathbb{F}_q$ with $q \in \{2,4,8,16,32\}$ for values of r in the range $\{3,\ldots,50\}$ and several m. The Goppa codes are generated by means of an irreducible Γ of degree r and hence Γ has no multiple roots. In particular, we can apply Theorem 3 in the binary case. We compare the dimensions of the solution space against the dimension D_{random} of the system derived from a random linear code. Table 1 and Table 2 give figures for the binary case with $m = 14$. We define $T_{\mathsf{alternant}}$ and T_{Goppa} respectively as the expected dimensions for an alternant and a Goppa code deduced from the formulas (13) and (15)-(17). We can check that D_{random} is equal to 0 for $r \in \{3,\ldots,12\}$ and $D_{\text{random}} = N - k$ as expected. We remark that $D_{\text{alternant}}$ is different from D_{random} whenever $r \leq 15$, and D_{Goppa} is different from D_{random} as long as $r \leq 25$. Finally we observe that our formulas for $T_{\mathsf{alternant}}$ fit as long as $k \geq N - T_{\mathsf{alternant}}$ which correspond to $r \leq 15$. This is also the case for binary Goppa codes since we have $T_{\mathsf{Goppa}} = D_{\text{Goppa}}$ as long as $k \geq N - T_{\mathsf{Goppa}}$ *i.e.* $r \leq 25$. We also give in Table 10 and Table 11 in Appendix B the examples that we obtained for $q = 4$ and $m = 6$ to check that the arguments also apply. We also compare binary Goppa codes and random linear codes for $m = 15$ in Table 4-6 and $m = 16$ in Table 7-9 (See Appendix B). We see that D_{random} and D_{Goppa} are different for $r \leq 33$ when $m = 15$ and for $m = 16$ they are different even beyond our range of experiment $r \leq 50$.

Table 1. $q = 2$ and $m = 14$

r	3	4	5	6	7	8	9	10	11	12	13	14	15	16
N	861	1540	2415	3486	4753	6216	7875	9730	11781	14028	16471	19110	21945	24976
k	16342	16328	16314	16300	16286	16272	16258	16244	16230	16216	16202	16188	16174	16160
D_{random}	0	0	0	0	0	0	0	0	0	0	269	2922	5771	8816
$D_{\text{alternant}}$	42	126	308	560	882	1274	1848	2520	3290	4158	5124	6188	7350	8816
$T_{\mathsf{alternant}}$	42	126	308	560	882	1274	1848	2520	3290	4158	5124	6188	7350	8610
D_{Goppa}	252	532	980	1554	2254	3080	4158	5390	6776	8316	10010	11858	13860	16016
T_{Goppa}	252	532	980	1554	2254	3080	4158	5390	6776	8316	10010	11858	13860	16016
$\tilde{N}$	903	1596	2485	3570	4851	6328	8001	9870	11935	14196	16653	19306	22155	25200
$\tilde{k}$	16341	16327	16313	16299	16285	16271	16257	16243	16229	16215	16201	16187	16173	16159
$\tilde{D}_{\text{random}}$	42	56	70	84	98	112	126	140	154	168	453	3120	5983	9041
$\tilde{D}_{\text{alternant}}$	84	182	378	644	980	1386	1974	2660	3444	4326	5306	6384	7560	9041
$\tilde{D}_{\text{Goppa}}$	294	588	1050	1638	2352	3192	4284	5530	6930	8484	10192	12054	14070	16240

6 Cryptographic Implications

The existence of a distinguisher for the specific case of binary Goppa codes has consequences for code-based cryptographic primitives because it is represents, and by far, the favorite choice in such primitives. One of the reasons for this, is the fact that this class has withstood many cryptographic attacks for more than thirty years now. We focus in this part on secure parameters that are within the range of validity of our distinguisher. In Section 5, we gave a general expression of the distinguisher for a Goppa code over any finite field $\mathbb{F}_q$. This expression can be easily simplified in the binary case ($q = 2$).

Proposition 4. *Let us define* $\ell \stackrel{def}{=} \lceil \log_2 r \rceil + 1$ *and* $N \stackrel{def}{=} \binom{mr}{2}$. *The formula for* D_{Goppa} *given in Equation* (17) *can be simplified to* $D_{Goppa} = \frac{mr}{2}\left((2\ell+1)r - 2^{\ell} - 1\right)$ *as long as* $N - D_{Goppa} < n - mr$.

This simple expression is therefore not true for any value of r and m but tends to be true for codes that have a code rate $\frac{n-mr}{n}$ that is close to one. This kind of codes are mainly encountered with the public keys of the

Table 2. $q = 2$ and $m = 14$

r	17	18	19	20	21	22	23	24	25	26	27	28	29	30
N	28203	31626	35245	39060	43071	47278	51681	56280	61075	66066	71253	76636	82215	87990
k	16146	16132	16118	16104	16090	16076	16062	16048	16034	16020	16006	15992	15978	15964
D_{random}	12057	15494	19127	22956	26981	31202	35619	40232	45041	50046	55247	60644	66237	72026
$D_{\text{alternant}}$	12057	15494	19127	22956	26981	31202	35619	40232	45041	50046	55247	60644	66237	72026
$T_{\mathsf{alternant}}$	10192	11900	13734	15694	17780	19992	22330	24794	27384	30100	32942	35910	39004	42224
D_{Goppa}	18564	21294	24206	27300	30576	34034	37674	41496	45500	50046	55247	60644	66237	72026
T_{Goppa}	18564	21294	24206	27300	30576	34034	37674	41496	45500	49686	54054	58604	63336	68250
$\tilde{N}$	28441	31878	35511	39340	43365	47586	52003	56616	61425	66430	71631	77028	82621	88410
$\tilde{k}$	16145	16131	16117	16103	16089	16075	16061	16047	16033	16019	16005	15991	15977	15963
$\tilde{D}_{\text{random}}$	12296	15747	19394	23237	27277	31512	35942	40569	45393	50411	55626	61037	66644	72447
$\tilde{D}_{\text{alternant}}$	12297	15747	19395	23238	27277	31511	35943	40570	45392	50412	55626	61038	66644	72447
$\tilde{D}_{\text{Goppa}}$	18802	21546	24472	27580	30870	34342	37996	41832	45850	50412	55626	61037	66644	72447

CFS signature scheme. We will show that there also exist public keys of the McEliece cryptosystem that can be distinguished for parameters considered as secure. We assume that the length n is equal to 2^m and we denote by r_{min} the smallest integer r such that $N - D_{\text{Goppa}} \geq 2^m - mr$. Recall that given a degree extension m over $\mathbb{F}_2$, any binary Goppa code defined with a polynomial $\Gamma(z)$ of degree $r \geq r_{\text{min}}$ cannot be distinguished from a random linear code by our technique. This value is gathered in Table 3 for different values of m. It provides therefore a lower bound for r in the choice of secure parameters if being unable to distinguish the public code from a random linear code is required. One can notice for instance that the McEliece key obtained with $m = 13$ and $r = 19$ and which corresponds to 90-bit of security, fits in the range of validity of our distinguisher. The values of r_{min} in Table 3 are checked by experimentations for $m \leq 16$ whereas those for $m \geq 17$ are obtained by solving the equation $\frac{mr}{2}\left((2\ell+1)r - 2^\ell - 1\right) = \frac{1}{2}mr(mr-1) - 2^m + mr$. Additionally, all the keys proposed in [18] (See therein Table 4) for the CFS scheme can be distinguished.

Table 3. Smallest order r of a binary Goppa code of length $n = 2^m$ for which our distinguisher does not work.

m	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
r_{min}	5	8	8	11	16	20	26	34	47	62	85	114	157	213	290	400

7 An Explanation for the Distinguisher

The goal of this section is to provide a theoretical explanation to the practical behavior observed in the previous section. We first consider the case of alternant codes and will explain the defect of rank observed in the linearized systems described previously.

7.1 The generic alternant case

As a general comment, we emphasize that it seems difficult to obtain a precise lower bound or upper bound on the dimension D, respectively $\tilde{D}$ of the vector space solution of (11), respectively (12) holding for all alternant codes. Indeed, it is always possible to have degenerate cases for particular $\mathbf{x}$ and $\mathbf{y}$ defining the alternant code

$\mathscr{A}_r(\mathbf{x},\mathbf{y})$. When $\mathbf{x}$ and $\mathbf{y}$ are chosen in a subfield $\mathbb{F}_{q^{m'}}$ with m' being a divisor of m, then the dimension D of the system is much smaller than predicted in experimental Fact 1. We have typically the same formula as in (13), but with m' replacing m there. On the other hand, when $\mathbf{y}$ is chosen accordingly to a Goppa code, then the dimension can be much larger.

However, there is a simple fact explaining what happens in the generic case for Formula (13), i.e. for "random" choices of $\mathbf{x}$ and $\mathbf{y}$. Indeed, to set up the linear system (11) or (12) we have used the trivial identity $Y_iY_iX_i^2 = (Y_iX_i)^2$. More generally, we can use any identity of the form $Y_iX_i^aY_iX_i^b = Y_iX_i^cY_iX_i^d$ with $a,b,c,d \in \{0,1,\dots,r-1\}$ such that $a+b=c+d$. It is straightforward to check that we obtain in the same way the algebraic system:

$$\sum_{j=k+1}^{n}\sum_{j'>j} p_{i,j}p_{i,j'}\left(Y_jX_j^aY_{j'}X_{j'}^b + Y_{j'}X_{j'}^aY_jX_j^b + Y_jX_j^cY_{j'}X_{j'}^d + Y_{j'}X_{j'}^cY_jX_j^d\right) = 0 \tag{19}$$

and

$$\sum_{j=\tilde{k}+1}^{n}\sum_{j'>j} \tilde{p}_{i,j}\tilde{p}_{i,j'}\left(Y_jX_j^aY_{j'}X_{j'}^b + Y_{j'}X_{j'}^aY_jX_j^b + Y_jX_j^cY_{j'}X_{j'}^d + Y_{j'}X_{j'}^cY_jX_j^d\right) = 0. \tag{20}$$

In other words:

$$Z_{a,b,c,d} \stackrel{\text{def}}{=} \left(Y_jX_j^aY_{j'}X_{j'}^b + Y_{j'}X_{j'}^aY_jX_j^b + Y_jX_j^cY_{j'}X_{j'}^d + Y_{j'}X_{j'}^cY_jX_j^d\right)_{\substack{1\le j\le mr\\ j'>j}}$$

is a solution of (11) whereas

$$\tilde{Z}_{a,b,c,d} \stackrel{\text{def}}{=} \left(Y_jX_j^aY_{j'}X_{j'}^b + Y_{j'}X_{j'}^aY_jX_j^b + Y_jX_j^cY_{j'}X_{j'}^d + Y_{j'}X_{j'}^cY_jX_j^d\right)_{\substack{1\le j\le n-\tilde{k}\\ j'>j}}$$

is a solution of (12). This yields many (presumably) independent vectors which are solution of (11) or (12). In other words, large dimension of the vector space solution of (11) or (12) is explained by the fact that *there are many different ways of combining the equations of the algebraic system* (10) *together yielding the same linearized systems* (11) *or* (12).

Observe that there are some relations among solutions, such as $Z_{a,b,c,d} + Z_{c,d,e,f} = Z_{a,b,e,f}$. However, if we define

$$S_t \stackrel{\text{def}}{=} \{\{a,b\} | a+b=t\},$$

then we expect to obtain $\sum_t(|S_t|-1)$ linearly independent solutions to (11) or (12) from this process. The term $|S_t|-1$ in the sum is a simple consequence of the following fact.

Fact 4. *Assume that we have ℓ independent (over $\mathbb{F}_2$) vectors $e_1,\dots,e_\ell$. Then the set $\{e_i+e_j : i,j \in \{1,\dots,\ell\}\}$ generates a vector space of dimension $\ell-1$ over $\mathbb{F}_2$.*

Finally, the solutions have coefficients over $\mathbb{F}_{q^m}$. By decomposing each coefficient over $\mathbb{F}_q$ we may finally have $m\sum_t(|S_t|-1)$ (potentially) independent vectors over $\mathbb{F}_q$. This accounts for a generating set of size:

$$\frac{m(r-1)(r-2)}{2}$$

which agrees with Formula (13) when $r \le q$.

For larger values of r, the automorphisms of $\mathbb{F}_{q^m}$ leaving $\mathbb{F}_q$ invariant have to be used. They are of the form $x \mapsto x^{q^l}$ for some $\ell \in \{0,\dots,m-1\}$. Notice that if we raise the equation $Y_iX_i = \sum p_{ij}Y_jX_j$ to the q-th power we get:

$$Y_i^qX_i^q = \sum p_{ij}Y_j^qX_j^q.$$

We can use the same trick for $Y_i = \sum p_{ij}Y_j$. From the trivial identity $Y_i(Y_iX_i)^q = Y_i^qY_iX_i^q$, we obtain a new algebraic equation which is

$$\sum_{j=k+1}^{n}\sum_{j'>j} p_{i,j}p_{i,j'}\left(Y_jY_{j'}^qX_{j'}^q + Y_{j'}Y_j^qX_j^q + Y_j^qY_{j'}X_{j'}^q + Y_{j'}^qY_jX_j^q\right) = 0. \tag{21}$$

To use $Y_i X_i^q = \sum p_{ij} Y_j X_j^q$, we need to have $r \geq q+1$. However it should be noticed that if $a+b=c+d$ then $Z_{a,b,c,d}$ and $Z_{qa,qb,qc,qd}$ only give m (potentially) independent vectors over $\mathbb{F}_q$ (and not $2m$) after decomposing their coefficients over $\mathbb{F}_q$. This comes from the fact that the Frobenius map $x \mapsto x^q$ is a $\mathbb{F}_q$-linear transform. Therefore, the only new vectors obtained in this way are of the form $Z_{a,q^j b,c,q^j d}$ with $0 \leq a,b,c,d < r$, $0 \leq j < m$ and $a + q^j b = c + q^j d$. This whole discussion leads to

Heuristic 1 *Let* $S_t^0 \stackrel{def}{=} \{\{a,b\} | 0 \leq a < r, 0 \leq b < r, a+b=t\}$ [6]. *For* j *in* $\{1,\dots,m-1\}$, *we set* $S_t^j \stackrel{def}{=} \{(a,q^j b) | 0 \leq a < r, 0 \leq b < r$
Then, for most choices of $\mathbf{x}$ *and* $\mathbf{y}$, *we have:*

$$D_{alternant} = m \sum_{\{t,j : S_t^j\} \neq \emptyset} (|S_t^j| - 1).$$

The sum appearing in the right-hand side has a very simple expression which is given by

Proposition 5.

$$\sum_{\{t,j : S_t^j\} \neq \emptyset} (|S_t^j| - 1) = \frac{r-1}{2} \left\{ (2\ell+1)r - 2\frac{q^{\ell+1}-1}{q-1} \right\} \tag{22}$$

with $\ell \stackrel{def}{=} \lfloor \log_q (r-1) \rfloor$.

This finishes to explain the first part of Experimental Fact 1. In order to prove Proposition 5, we first prove the following lemma.

Lemma 1.

$$|S_t^0| = \left\lceil \frac{t+1}{2} \right\rceil, \textit{ for } 0 \leq t \leq r-1, \tag{23}$$

$$|S_t^0| = \left\lceil \frac{2r-t-1}{2} \right\rceil, \textit{ for } r \leq t \leq 2r-2, \tag{24}$$

$$|S_t^0| = 0 \textit{ otherwise.} \tag{25}$$

$$|S_t^j| \leq 1, \textit{ if } q^j \geq r, \tag{26}$$

$$|S_t^j| = \min\left(r, \left\lfloor \frac{t}{q^j} \right\rfloor + 1\right) - \max\left(\left\lceil \frac{t-r+1}{q^j} \right\rceil, 0\right) \textit{ otherwise.} \tag{27}$$

Proof. The first three equations follow directly from the definition of S_t^0. Equation (26) is an easy consequence of the definition of S_t^j. Let us assume now that $r > q^j$. We now prove Equation (27). Let (a,b) be a couple of integers such that:

$$0 \leq \quad a \quad \leq r-1 \tag{28}$$

$$0 \leq \quad b \quad \leq r-1 \tag{29}$$

$$a + q^j b = t. \tag{30}$$

From (28), (29) and (30), we obtain $t - q^j b \leq r-1$, which implies $b \geq \left\lceil \frac{t-r+1}{q^j} \right\rceil$. Together with (29)

$$b \geq \max\left(\left\lceil \frac{t-r+1}{q^j} \right\rceil, 0\right). \tag{31}$$

On the other hand, we also have $b \leq r-1$ and $b \leq \left\lfloor \frac{t}{q^j} \right\rfloor$ since $a \geq 0$. This implies

$$b \leq \min\left(r-1, \left\lfloor \frac{t}{q^j} \right\rfloor\right). \tag{32}$$

All the b's between these upper and lower bounds are possible. Then, there is only one corresponding a each time. This yields Equation (27). □

[6] The notation $\{a,b\}$ refers to a multiset here. We may have $a=b$.

From this, we deduce:

Lemma 2. *It holds that:*

$$\sum_{t:S_t^0\neq\emptyset}(|S_t^0|-1) = \frac{(r-1)(r-2)}{2}, \tag{33}$$

$$\sum_{t:S_t^j\neq\emptyset}(|S_t^j|-1) = (r-1)(r-q^j) \textit{ for } r\geq q^j, \tag{34}$$

$$\sum_{t:S_t^j\neq\emptyset}(|S_t^j|-1) = 0 \textit{ otherwise.} \tag{35}$$

Proof. Let us first prove (33). By using Lemma 1, we obtain

$$\sum_{t:S_t^0\neq\emptyset}(|S_t^0|-1) = \sum_{t=0}^{r-1}\left(\left\lceil\frac{t+1}{2}\right\rceil-1\right)+\sum_{t=r}^{2r-2}\left(\left\lceil\frac{2r-t-1}{2}\right\rceil-1\right)$$

For r odd (say $r=2r'+1$), we notice that $\sum_{t=0}^{r-1}\left(\left\lceil\frac{t+1}{2}\right\rceil-1\right)=r'(r-1)+r'=r'^2$ and that $\sum_{t=r}^{2r-2}\left(\left\lceil\frac{2r-t-1}{2}\right\rceil-1\right)=r'(r-1)$. This implies that $\sum_{t:S_t^0\neq\emptyset}(|S_t^0|-1)=r'(2r'-1)=\frac{(r-1)(r-2)}{2}$. On the other hand, for r even, say $r=2r'$, we obtain $\sum_{t=0}^{r-1}\left(\left\lceil\frac{t+1}{2}\right\rceil-1\right)=r'(r'-1)$ and $\sum_{t=r}^{2r-2}\left(\left\lceil\frac{2r-t-1}{2}\right\rceil-1\right)=r'-1+(r'-1)(r'-2)=(r'-1)^2$. From this, we deduce that $\sum_{t:S_t^0\neq\emptyset}(|S_t^0|-1)=(r'-1)(2r'-1)=\frac{(r-1)(r-2)}{2}$. This proves (33).

To prove (34), we first notice that $|S_t^j|$ is positive if and only if t belongs to $\{0,1,\dots,(q^j+1)(r-1)\}$. Then, we use Lemma 1 again and we obtain

$$\begin{aligned}
\sum_{t:S_t^j\neq\emptyset}(|S_t^j|-1) &= \sum_{t=0}^{(q^j+1)(r-1)}(|S_t^j|-1) && (36)\\
&= \sum_{t=0}^{(q^j+1)(r-1)}\min\left(r,\left\lfloor\frac{t}{q^j}\right\rfloor+1\right)-\max\left(\left\lceil\frac{t-r+1}{q^j}\right\rceil,0\right)-1\\
&= \sum_{t=0}^{(q^j+1)(r-1)}\min\left(r-1,\left\lfloor\frac{t}{q^j}\right\rfloor\right)-\max\left(\left\lceil\frac{t-r+1}{q^j}\right\rceil,0\right)\\
&= \sum_{t=0}^{(q^j+1)(r-1)}\min\left(r-1,\left\lfloor\frac{t}{q^j}\right\rfloor\right)-\sum_{t=0}^{(q^j+1)(r-1)}\max\left(\left\lceil\frac{t-r+1}{q^j}\right\rceil,0\right). && (37)
\end{aligned}$$

Observe now that

$$\begin{aligned}
\sum_{t=0}^{(q^j+1)(r-1)}\min\left(r-1,\left\lfloor\frac{t}{q^j}\right\rfloor\right) &= \sum_{t=0}^{q^j(r-1)-1}\left\lfloor\frac{t}{q^j}\right\rfloor+\sum_{t=q^j(r-1)}^{q^j(r-1)-1}(r-1)\\
&= q^j(0+1+\cdots+r-2)+(r-1)r.
\end{aligned}$$

The other term appearing in the right-hand side of (37) is handled as follows

$$\begin{aligned}
\sum_{t=0}^{(q^j+1)(r-1)}\max\left(\left\lceil\frac{t-r+1}{q^j}\right\rceil,0\right) &= \sum_{t=r}^{(q^j+1)(r-1)}\left\lceil\frac{t-r+1}{q^j}\right\rceil\\
&= q^j(1+2+\cdots+r-1).
\end{aligned}$$

By plugging these two expressions in (37) we obtain

$$\sum_{t:S_t^j\neq\emptyset}(|S_t^j|-1)=q^j(0+1+\cdots+r-2)+(r-1)r-q^j(1+2+\cdots+r-1)=(r-1)r-q^j(r-1)=(r-1)(r-q^j).$$

□

Finally, we can now finish with the proof of Proposition 5.

Proof.

$$\sum_{t,j:S_t^j\neq\emptyset}(|S_t^j|-1) = \sum_{t:S_t^0\neq\emptyset}(|S_t^0|-1)+\sum_{j=1}^{m-1}\sum_{t,:S_t^j\neq\emptyset}(|S_t^j|-1)$$
$$= \frac{(r-1)(r-2)}{2}+\sum_{j:q^j<r}(r-1)(r-q^j)$$

Let ℓ be the largest integer such that $r > q^\ell$. We obtain

$$\sum_{t,j:S_t^j\neq\emptyset}(|S_t^j|-1) = \frac{r-1}{2}\left\{2\ell r+(r-2)-2\sum_{j=1}^{\ell}q^j\right\}$$
$$= \frac{r-1}{2}\left\{2(\ell+1)r-2\sum_{j=0}^{\ell}q^j\right\}$$
$$= \frac{r-1}{2}\left\{2(\ell+1)r-2\frac{q^{\ell+1}-1}{q-1}\right\}.$$

This concludes the proof. □

7.2 The Goppa case

The simplest way to understand why there is a difference between the generic alternant case and the Goppa case is to compare $\tilde{D}_{\text{Goppa}}$ with $\tilde{D}_{\text{alternant}}$. First of all, the same reasoning as in the previous subsection can be done for the subcode $\tilde{\mathscr{A}}_r(\mathbf{x},\mathbf{y})$ of even weights of an alternant code $\mathscr{A}_r(\mathbf{x},\mathbf{y})$. This leads in the same way to the conclusion that in general:

$$\tilde{D}_{\text{alternant}} = \frac{m(r-1)}{2}\left\{(2\ell+1)r-2\frac{q^{\ell+1}-1}{q-1}\right\},$$

with $\ell \stackrel{\text{def}}{=} \lfloor \log_q(r-1)\rfloor$. Notice, that from Proposition 2, we know that $\tilde{\mathscr{G}}(\mathbf{x},\Gamma)$ is an alternant code of degree $r+1$, when Γ is of degree r. Therefore, we have

$$\tilde{D}_{\text{Goppa}} \geq \frac{mr}{2}\left\{(2\ell+1)(r+1)-2\frac{q^{\ell+1}-1}{q-1}\right\}.$$

with $\ell \stackrel{\text{def}}{=} \lfloor \log_q(r)\rfloor$. This explains why $\tilde{D}_{\text{Goppa}}$ is significantly greater than $\tilde{D}_{\text{alternant}}$. If we we denote by $\tilde{D}_{\text{Goppa}}(r)$ the dimension of the solution space of (12) for a Goppa code associated to a polynomial of degree r (we fix the order m of the extension) and if we denote by $\tilde{D}_{\text{alternant}}(r)$ the dimension of the solution space of (11) for a generic alternant code $\mathscr{A}_r$ of degree r, then this explains why we have

$$\tilde{D}_{\text{Goppa}}(r) \geq \tilde{D}_{\text{alternant}}(r+1).$$

It should be added that for $r \leq q-2$, we actually have $\tilde{D}_{\text{Goppa}}(r) = \tilde{D}_{\text{alternant}}(r+1)$.

We do not have a general explanation for the formula observed for D_{Goppa} of non-binary Goppa codes. However, in the case of binary Goppa codes we can use Theorem 3. In this case, when the Goppa polynomial Γ has only simple roots, we know that $\mathscr{G}(\mathbf{x},\Gamma) = \mathscr{A}_{2r}(\mathbf{x},\mathbf{y}')$, where $r \stackrel{\text{def}}{=} \deg(\Gamma)$ and $y_i' = \Gamma(x_i)^{-2}$ where the x_i's are the coordinates of $\mathbf{x}$ and the y_i''s are the coordinates of $\mathbf{y}'$. This basically explains why the vector space solution of (11) is much greater for a binary Goppa code than for a binary alternant code of the same degree. This would suggest that $D_{\text{Goppa}}(r) \geq D_{\text{alternant}}(2r)$. However, this is not true. Now, there are linear relations among the vectors $Z_{a,b,c,d}$ which are solutions of (11). Providing a cleaner explanation of the formula obtained for binary Goppa codes is much more involved and is beyond the scope of this article.

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. Daniel J. Bernstein, Tanja Lange, Ruben Niederhagen, Christiane Peters, and Peter Schwabe. FSBday: Implementing Wagner's generalized birthday attack against the round-1 SHA-3 candidate FSB. In *INDOCRYPT*, pages 18–38, 2009.
7. Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Attacking and defending the McEliece cryptosystem. In *PQCrypto*, pages 31–46, 2008.
8. Bhaskar Biswas and Nicolas Sendrier. McEliece cryptosystem implementation: Theory and practice. In *PQCrypto*, pages 47–62, 2008.
9. 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.
10. 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.
11. Pierre-Louis Cayrel, Philippe Gaborit, David Galindo, and Marc Girault. Improved identity-based identification using correcting codes. *CoRR*, abs/0903.0069, 2009.
12. 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.
13. Léonard Dallot. Towards a concrete security proof of Courtois, Finiasz and Sendrier signature scheme. In *WEWoRC*, pages 65–77, 2007.
14. Léonard Dallot and Damien Vergnaud. Provably secure code-based threshold ring signatures. In *IMA Int. Conf.*, pages 222–235, 2009.
15. 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.
16. Arne Dür. The automorphism groups of Reed-Solomon codes. *Journal of Combinatorial Theory, Series A*, 44:69–82, 1987.
17. 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.
18. 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.
19. 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.
20. 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.
21. P. Loidreau and N. Sendrier. Weak keys in the mceliece public-key cryptosystem. *IEEE Transactions on Information Theory*, 47(3):1207–1211, 2001.
22. F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*. North–Holland, Amsterdam, fifth edition, 1986.
23. R. J. McEliece. *A Public-Key System Based on Algebraic Coding Theory*, pages 114–116. Jet Propulsion Lab, 1978. DSN Progress Report 44.
24. 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.
25. R. Niebuhr, M. Meziani, S. Bulygin, and J. Buchmann. Selecting parameters for secure McEliece-based cryptosystems. Technical Report 2010/271, IACR, 2010.
26. H. Niederreiter. A public-key cryptosystem based on shift register sequences. In *EUROCRYPT*, volume 219 of *LNCS*, pages 35–39, 1985.

27. 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.
28. N. Patterson. The algebraic decoding of Goppa codes. *IEEE Transactions on Information Theory*, 21(2):203–207, 1975.
29. N. Sendrier. Finding the permutation between equivalent linear codes: The support splitting algorithm. *IEEE Transactions on Information Theory*, 46(4):1193–1203, 2000.
30. 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.
31. 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 Proof of Proposition 1

Proof. Let $\mathbf{c} = (c_i)_{1 \leq i \leq n}$ be a codeword in $\mathscr{A}_r(\mathbf{x}, \mathbf{y})$. Consider a polynomial $P(X) = \sum_{j=0}^{r-1} a_j X^j \in \mathbb{F}_{q^m}[X]$ of degree at most $r-1$ and notice that

$$\begin{aligned} y'_i P\left(\frac{ax_i + b}{cx_i + d}\right) &= y_i (cx_i + d)^{r-1} \sum_{0 \leq j \leq r-1} a_j \left(\frac{ax_i + b}{cx_i + d}\right)^j \\ &= y_i \sum_{0 \leq j \leq r-1} a_j (ax_i + b)^j (cx_i + d)^{r-1-j} \\ &= y_i Q(x_i) \end{aligned}$$

where Q is a polynomial of degree at most $r-1$ which depends on a, b, c, d but not on i. By the very definition of $\mathscr{A}_r(\mathbf{x}, \mathbf{y})$, we know that

$$0 = \sum_{i=1}^{n} c_i y_i Q(x_i) = \sum_{i=1}^{n} c_i y'_i P\left(\frac{ax_i + b}{cx_i + d}\right).$$

In other words, we have just proved that $\mathbf{c} \in \mathscr{A}_r\left(\frac{a\mathbf{x}+b}{c\mathbf{x}+d}, \mathbf{y}'\right)$. This proves that

$$\mathscr{A}_r(\mathbf{x}, \mathbf{y}) \subset \mathscr{A}_r\left(\frac{a\mathbf{x} + b}{c\mathbf{x} + d}, \mathbf{y}'\right).$$

The inclusion in the other direction is proved similarly.

B Experimental Results

Table 4. $q = 2$ and $m = 15$

r	3	4	5	6	7	8	9	10	11	12	13	14	15	16
N	990	1770	2775	4005	5460	7140	9045	11175	13530	16110	18915	21945	25200	28680
k	32723	32708	32693	32678	32663	32648	32633	32618	32603	32588	32573	32558	32543	32528
D_{random}	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D_{Goppa}	270	570	1050	1665	2415	3300	4455	5775	7260	8910	10725	12705	14850	17160
T_{Goppa}	270	570	1050	1665	2415	3300	4455	5775	7260	8910	10725	12705	14850	17160
$\tilde{N}$	1035	1830	2850	4095	5565	7260	9180	11325	13695	16290	19110	22155	25425	28920
$\tilde{k}$	32722	32707	32692	32677	32662	32647	32632	32617	32602	32587	32572	32557	32542	32527
$\tilde{D}_{\text{random}}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\tilde{D}_{\text{Goppa}}$	315	630	1125	1755	2520	3420	4590	5925	7425	9090	10920	12915	15075	17400

Table 5. $q = 2$ and $m = 15$

r	17	18	19	20	21	22	23	24	25	26	27	28	29	30
N	32385	36315	40470	44850	49455	54285	59340	64620	70125	75855	81810	87990	94395	101025
k	32513	32498	32483	32468	32453	32438	32423	32408	32393	32378	32363	32348	32333	32318
D_{random}	0	3817	7987	12382	17002	21847	26917	32212	37732	43477	49447	55642	62062	68707
D_{Goppa}	19890	22815	25935	29250	32760	36465	40365	44460	48750	53235	57915	62790	67860	73125
T_{Goppa}	19890	22815	25935	29250	32760	36465	40365	44460	48750	53235	57915	62790	67860	73125
$\tilde{N}$	32640	36585	40755	45150	49770	54615	59685	64980	70500	76245	82215	88410	94830	101475
$\tilde{k}$	32512	32497	32482	32467	32452	32437	32422	32407	32392	32377	32362	32347	32332	32317
$\tilde{D}_{\text{random}}$	128	4088	8273	12683	17318	22178	27263	32573	38108	43868	49853	56063	62498	69158
$\tilde{D}_{\text{Goppa}}$	20145	23085	26220	29550	33075	36795	40710	44820	49125	53625	58320	63210	68295	73575

Table 6. $q = 2$ and $m = 15$

r	31	32	33	34	35	36	37	38	39	40	41	42	43	44
N	107880	114960	122265	129795	137550	145530	153735	162165	170820	179700	188805	198135	207690	217470
k	32303	32288	32273	32258	32243	32228	32213	32198	32183	32168	32153	32138	32123	32108
D_{random}	75577	82672	89992	97537	105307	113302	121522	129967	138637	147532	156652	165997	175567	185362
D_{Goppa}	78585	84240	90585	97537	105307	113302	121522	129967	138637	147532	156652	165997	175567	185362
T_{Goppa}	78585	84240	90585	97155	103950	110970	118215	125685	133380	141300	149445	157815	166410	175230
$\tilde{N}$	108345	115440	122760	130305	138075	146070	154290	162735	171405	180300	189420	198765	208335	218130
$\tilde{k}$	32302	32287	32272	32257	32242	32227	32212	32197	32182	32167	32152	32137	32122	32107
$\tilde{D}_{\text{random}}$	76043	83153	90488	98048	105833	113843	122078	130538	139223	148133	157268	166628	176213	186023
$\tilde{D}_{\text{Goppa}}$	79050	84720	91080	98048	105833	113843	122079	130539	139224	148134	157269	166628	176214	186024

Table 7. $q = 2$ and $m = 16$

r	3	4	5	6	7	8	9	10	11	12	13	14	15	16
N	1128	2016	3160	4560	6216	8128	10296	12720	15400	18336	21528	24976	28680	32640
k	65488	65472	65456	65440	65424	65408	65392	65376	65360	65344	65328	65312	65296	65280
D_{random}	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D_{Goppa}	288	608	1120	1776	2576	3520	4752	6160	7744	9504	11440	13552	15840	18304
T_{Goppa}	288	608	1120	1776	2576	3520	4752	6160	7744	9504	11440	13552	15840	18304
$\tilde{N}$	1176	2080	3240	4656	6328	8256	10440	12880	15576	18528	21736	25200	28920	32896
$\tilde{k}$	65487	65471	65455	65439	65423	65407	65391	65375	65359	65343	65327	65311	65295	65279
$\tilde{D}_{\text{random}}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\tilde{D}_{\text{Goppa}}$	336	672	1200	1872	2688	3648	4896	6320	7920	9696	11648	13776	16080	18560

Table 8. $q = 2$ and $m = 16$

r	17	18	19	20	21	22	23	24	25	26	27	28	29	30
N	36856	41328	46056	51040	56280	61776	67528	73536	79800	86320	93096	100128	107416	114960
k	65264	65248	65232	65216	65200	65184	65168	65152	65136	65120	65104	65088	65072	65056
D_{random}	0	0	0	0	0	0	2360	8384	14664	21200	27992	35040	42344	49904
D_{Goppa}	21216	24336	27664	31200	34944	38896	43056	47424	52000	56784	61776	66976	72384	78000
T_{Goppa}	21216	24336	27664	31200	34944	38896	43056	47424	52000	56784	61776	66976	72384	78000
$\tilde{N}$	37128	41616	46360	51360	56616	62128	67896	73920	80200	86736	93528	100576	107880	115440
$\tilde{k}$	65263	65247	65231	65215	65199	65183	65167	65151	65135	65119	65103	65087	65071	65055
$\tilde{D}_{\text{random}}$	0	0	0	0	0	0	2729	8769	15065	21617	28425	35489	42809	50385
$\tilde{D}_{\text{Goppa}}$	21488	24624	27968	31520	35280	39248	43424	47808	52400	57200	62208	67424	72848	78480

Table 9. $q = 2$ and $m = 16$

r	31	32	33	34	35	36	37	38	39	40	41	42	43
N	122760	130816	139128	147696	156520	165600	174936	184528	194376	204480	214840	225456	236328
k	65040	65024	65008	64992	64976	64960	64944	64928	64912	64896	64880	64864	64848
D_{random}	57720	65792	74120	82704	91544	100640	109992	119600	129464	139584	149960	160592	171480
D_{Goppa}	83824	89856	96624	103632	110880	118368	126096	134064	142272	150720	159408	168336	177504
T_{Goppa}	83824	89856	96624	103632	110880	118368	126096	134064	142272	150720	159408	168336	177504
$\tilde{N}$	123256	131328	139656	148240	157080	166176	175528	185136	195000	205120	215496	226128	237016
$\tilde{k}$	65039	65023	65007	64991	64975	64959	64943	64927	64911	64895	64879	64863	64847
$\tilde{D}_{\text{random}}$	58217	66305	74649	83249	92105	101217	110585	120209	130089	140225	150617	161265	172169
$\tilde{D}_{\text{Goppa}}$	84320	90368	97152	104176	111440	118944	126688	134672	142896	151360	160064	169008	178192

Table 10. $q = 4$ and $m = 6$

r	3	4	5	6	7	8	9	10	11	12	13	14	15	16
N	153	276	435	630	861	1128	1431	1770	2145	2556	3003	3486	4005	4560
k	4078	4072	4066	4060	4054	4048	4042	4036	4030	4024	4018	4012	4006	4000
D_{random}	0	0	0	0	0	0	0	0	0	0	0	0	0	560
$D_{\text{alternant}}$	6	18	60	120	198	294	408	540	690	858	1044	1248	1470	1710
$T_{\text{alternant}}$	6	18	60	120	198	294	408	540	690	858	1044	1248	1470	1710
D_{Goppa}	18	60	120	198	294	408	540	750	990	1260	1560	1890	2250	2640
T_{Goppa}	18	60	120	198	294	408	540	750	990	1260	1560	1890	2250	2640
$\tilde{N}$	171	300	465	666	903	1176	1485	1830	2211	2628	3081	3570	4095	4656
$\tilde{k}$	4077	4071	4065	4059	4053	4047	4041	4035	4029	4023	4017	4011	4005	3999
$\tilde{D}_{\text{random}}$	0	0	0	0	0	0	0	0	0	0	0	0	90	657
$\tilde{D}_{\text{alternant}}$	6	18	60	120	198	294	408	540	690	858	1044	1248	1470	1710
$\tilde{D}_{\text{Goppa}}$	36	84	150	234	336	456	594	810	1056	1332	1638	1974	2340	2736

Table 11. $q = 4$ and $m = 6$

r	17	18	19	20	21	22	23	24	25	26	27	28	29	30
N	5151	5778	6441	7140	7875	8646	9453	10296	11175	12090	13041	14028	15051	16110
k	3994	3988	3982	3976	3970	3964	3958	3952	3946	3940	3934	3928	3922	3916
D_{random}	1157	1790	2459	3164	3905	4682	5495	6344	7229	8150	9107	10100	11129	12194
$D_{\text{alternant}}$	2064	2448	2862	3306	3905	4682	5495	6344	7229	8150	9107	10100	11129	12194
$T_{\text{alternant}}$	2064	2448	2862	3306	3780	4284	4818	5382	5976	6600	7254	7938	8652	9396
D_{Goppa}	3060	3510	3990	4500	5040	5610	6210	6840	7500	8190	9107	10100	11129	12194
T_{Goppa}	3060	3510	3990	4500	5040	5610	6210	6840	7500	8190	8910	9660	10440	11250
$\tilde{N}$	5253	5886	6555	7260	8001	8778	9591	10440	11325	12246	13203	14196	15225	16290
$\tilde{k}$	3993	3987	3981	3975	3969	3963	3957	3951	3945	3939	3933	3927	3921	3915
$\tilde{D}_{\text{random}}$	1260	1899	2575	3285	4032	4816	5634	6489	7380	8307	9270	10269	11304	12375
$\tilde{D}_{\text{alternant}}$	2064	2448	2862	3306	4032	4815	5634	6489	7380	8307	9270	10269	11304	12375
$\tilde{D}_{\text{Goppa}}$	3162	3618	4104	4620	5166	5742	6348	6984	7650	8346	9270	10270	11305	12375