A Distinguisher for High Rate McEliece Cryptosystems

Jean-Charles Faugère, Valérie Gauthier-Umaña, Ayoub Otmani, Ludovic Perret, Jean-Pierre Tillich

Abstract

The Goppa Code Distinguishing (GD) problem consists in distinguishing the matrix of a Goppa code from a random matrix. The hardness of this problem is an assumption to prove the security of code-based cryptographic primitives such as McEliece's cryptosystem. Up to now, it is widely believed that the GD problem is a hard decision problem. We present the first method allowing to distinguish alternant and Goppa codes over any field. Our technique can solve the GD problem in polynomial-time provided that the codes have sufficiently large rates. The key ingredient is an algebraic characterization of the key-recovery problem. The idea is to consider the rank of a linear system which is obtained by linearizing a particular polynomial system describing a key-recovery attack. It appears that this dimension depends on the type of code considered. Explicit formulas derived from extensive experimentations for the rank are provided for "generic" random, alternant, and Goppa codes over any field. Finally, we give theoretical explanations of these formulas in the case of random codes, alternant codes over any field of characteristic two and binary Goppa codes.

Index Terms

McEliece cryptosystem, CFS signature, Algebraic cryptanalysis, Goppa Code Distinguishing problem.

I. Introduction

THIS paper[1] investigates the difficulty of the Goppa Code Distinguishing (GD) problem which first appeared in [2]. This is a decision problem that aims at recognizing a generator matrix of a binary Goppa code from a randomly drawn binary matrix. Up to now, it is assumed that no polynomial time algorithm exists that distinguishes a generator matrix of a Goppa code from a randomly picked generator matrix.

The main motivation for introducing the GD problem is to formally relate the problem of decoding a random linear code to the security of the McEliece public-key cryptosytem [3]. Since its apparition, this cryptosystem has withstood many attacks and after more than thirty years now, it still belongs to the very few unbroken public key cryptosystems. This situation substantiates the claim that inverting the encryption function, and in particular recovering the private key from public data, is intractable. The classical methods for inverting the McEliece encryption function without finding a trapdoor all resort to the use of the best general decoding algorithms [4]–[11]. All these algorithms, whose time complexity is exponential (in the length), attempt to solve the long-standing problem of decoding random linear code [12]. They also assume (implicitly or explicitly) that there does not exist an algorithm that is able to decode more efficiently McEliece public keys. Note that if ever such an algorithm exists, it would permit to solve the GD problem.

On the other hand, no significant breakthrough has been observed with respect to the problem of recovering the private key [13], [14]. This has led to state that the generator matrix of a binary Goppa code does not disclose any visible structure that an attacker could exploit. This is strengthened by the fact that Goppa codes share many characteristics with random codes. For instance they asymptotically meet the Gilbert-Varshamov bound. They also have a trivial permutation group, *etc.* Hence, the hardness of the GD problem has become a classical belief, and as a consequence, a *de facto* assumption to prove the semantic security in the standard model (IND-CPA in [15] and IND-CCA2 in [16]), and the security in the random oracle model against existential forgery [2], [17] of the signature scheme [2].

We present a deterministic polynomial-time distinguisher for codes whose rate is close to 1. This includes in particular codes encountered with the signature scheme CFS ([2], [18]). We emphasize that our method can distinguish codes also used in McEliece's encryption scheme. For instance, the binary Goppa code obtained with $m = 13$ and $r = 19$ corresponding to a 90-bit security key is distinguishable. We provide an asymptotic formula for the smallest rate R_{crit} for which one can

J.-C. Faugère and L. Perret are with
INRIA, Paris-Rocquencourt Center, POLSYS Project,
UPMC Univ Paris 06, UMR 7606, LIP6, F-75005, Paris, France,
CNRS, UMR 7606, LIP6, F-75005, Paris, France,
e-mail: jean-charles.faugere@inria.fr, ludovic.perret@lip6.fr
Valérie Gauthier-Umaña is with Departamento de Matemáticas, Universidad de los Andes, Bogotá, Colombia, email: ve.gauthier@uniandes.edu.co
A. Otmani is with LITIS - Université de Rouen, Avenue de l'Université, F-76801 Saint-Étienne-du-Rouvray Cedex, e-mail: ayoub.otmani@univ-rouen.fr
J.-P. Tillich is with SECRET Project – INRIA Rocquencourt, Domaine de Voluceau, B.P. 105, 78153 Le Chesnay Cedex - France, e-mail: jean-pierre.tillich@inria.fr

[1]A preliminary version [1] of this paper has been presented in IEEE Information Theory Workshop (ITW 2011), Paraty, Brazil, Oct. 16-20 2011.

distinguish a q-random code from a q-ary alternant or Goppa code (Theorem 3). If q is fixed and assuming that the length is q^m then when m tends to infinity, we have:

$$R_{\text{crit}} = 1 - \sqrt{\frac{2\,m\,\log q}{q^m \log m}}\big(1 + o(1)\big).$$

where all logarithms are taken to base 2.

Our distinguisher is based on the algebraic attack developed against compact variants of McEliece [19]. In this approach, the key-recovery problem is transformed into the one of solving an algebraic system. By using a linearization technique, we are able to derive a linear system whose rank is different from what one would expect in the random case. More precisely, we observe experimentally that this defect in the rank is directly related to the type of codes. We provide explicit formulas for "generic" random, alternant, and Goppa codes over any alphabet. We performed extensive experiments to confirm that the formulas are accurate. Eventually, we prove the formula in the random case and give explanations in the case of alternant codes over any field of characteristic two and binary Goppa codes.

However, the existence of our distinguisher does not undermine the security of primitives based on Goppa codes, but basically, it proves that the GD assumption is false for some parameters, and consequently should be used with great care as an assumption for a security reduction.

The paper is organized as follows. After recalling basic notions in coding-theory in Section II, we introduce in Section III our algebraic distinguisher which is basically the dimension of the solution space of a linear system that is deduced by linearization from the algebraic system that any McEliece cryptosystem must satisfy. We then provide explicit formulas that predicts the behavior of the distinguisher coming from experiments. In Section IV and Section V, we give explanations of the formulas for alternant and binary Goppa codes. In Section VI, we give a proof of its typical behavior in the random case. Lastly, we conclude over the cryptographic implications the distinguisher induces and we deduce an asymptotic formula for the smallest rate for which we can distinguish a random code from an alternant code or a Goppa code.

II. Code-Based Public-Key Cryptography

The problem of decoding random linear codes is a potential candidate for building public-key cryptographic primitives such as an encryption scheme. McEliece [3] was the first to use this problem in public-key cryptography. The idea is to start from afamily of codes equipped with a polynomial-time decoding algorithm. The fundamental concept of is to consider two equivalent representations of a code: one should facilitate the decoding, whereas the decoding should be infeasible from the other one. Although his design principle is general, he explicitly advocated to use binary Goppa codes [20].

A. *Coding Theory Background*

Code-based public-key cryptography focuses on linear codes that have a polynomial time decoding algorithm. We recall that a q-ary (linear) *code* $\mathscr{C}$ over the finite field $\mathbb{F}_q$ of q elements defined by a $k \times n$ matrix $\boldsymbol{G}$ (with $k \leqslant n$) whose entries belong to $\mathbb{F}_q$ is the vector space spanned by its rows *i.e.*,

$$\mathscr{C} \stackrel{\text{def}}{=} \Big\{\boldsymbol{u}\boldsymbol{G} \mid \boldsymbol{u} \in \mathbb{F}_q^k\Big\}.$$

The length of $\mathscr{C}$ is n and its rate is the ratio $R \stackrel{\text{def}}{=} k/n$. The role of decoding algorithms is to correct errors of prescribed weight. We say that a decoding algorithm corrects r errors if it recovers $\boldsymbol{u}$ from the knowledge of $\boldsymbol{u}\boldsymbol{G} + \boldsymbol{e}$ for all possible $\boldsymbol{e} \in \mathbb{F}_q^n$ of weight at most r.

One famous family of codes is the one of binary Goppa codes. It belongs to the more general class of alternant codes ([21, Chap. 12, p. 365]). The main well-known feature of an alternant code is the possibility of being decoded in polynomial time. It is more 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. We recall that a parity-check matrix $\boldsymbol{H}$ of a q-ary code $\mathscr{C}$ is defined as a matrix such that:

$$\mathscr{C} \stackrel{\text{def}}{=} \Big\{\boldsymbol{c} \in \mathbb{F}_q^n \mid \boldsymbol{H}\boldsymbol{c}^T = \boldsymbol{0}\Big\}.$$

where the symbol ^{T} means the transpose operation. For q-ary alternant codes of length $n \leqslant q^m$, there exists a parity-check matrix with a very special form related to rectangular Vandermonde matrices:

$$\boldsymbol{V}_r(\boldsymbol{x}, \boldsymbol{y}) \stackrel{\text{def}}{=} \begin{pmatrix} y_1 & \cdots & y_n \\ y_1 x_1 & \cdots & y_n x_n \\ \vdots & & \vdots \\ y_1 x_1^{r-1} & \cdots & y_n x_n^{r-1} \end{pmatrix}$$

where $\boldsymbol{x} = (x_1, \ldots, x_n)$ and $\boldsymbol{y} = (y_1, \ldots, y_n)$ are in $\mathbb{F}_{q^m}^n$.

Definition 1 (Alternant code): A q-ary alternant code of order r associated to $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{F}_{q^m}^n$ where all x_i's are distinct and $\boldsymbol{y} = (y_1, \ldots, y_n) \in \left(\mathbb{F}_{q^m}^*\right)^n$ denoted by $\mathscr{A}_r(\boldsymbol{x}, \boldsymbol{y})$ is

$$\left\{\boldsymbol{c} \in \mathbb{F}_q^n \mid \boldsymbol{V}_r(\boldsymbol{x}, \boldsymbol{y})\boldsymbol{c}^T = \boldsymbol{0}\right\}.$$

It is well-known that the dimension k of an alternant codes of degree r satisfies $k \geqslant n - rm$. Moreover, a key feature about them is the following property.

Proposition 1: An alternant codes of degree r can decode in polynomial time all errors of weight at most $\frac{r}{2}$ whenever there exists a parity-check matrix in the form $\boldsymbol{V}_r(\boldsymbol{x}^*, \boldsymbol{y}^*)$ for some vectors $\boldsymbol{x}^*$ and $\boldsymbol{y}^*$.

Definition 2 (Goppa codes): A q-ary Goppa code $\mathscr{G}(\boldsymbol{x}, \gamma)$ associated to a polynomial $\gamma(z) \stackrel{\text{def}}{=} \sum_{i=0}^{r} \gamma_i x^i$ of degree r over $\mathbb{F}_{q^m}$ and an n-tuple $\boldsymbol{x} = (x_1, \ldots, x_n)$ of distinct elements of $\mathbb{F}_{q^m}$ satisfying $\gamma(x_i) \neq 0$ for all $i, 1 \leqslant i \leqslant n$, is the q-ary alternant code $\mathscr{A}_r(\boldsymbol{x}, \boldsymbol{y})$ of order r with $y_i = \gamma(x_i)^{-1}$.

Goppa codes, viewed as alternant codes, inherit a decoding algorithm that corrects up to $\frac{r}{2}$ errors. But in the case of *binary* Goppa codes, it is possible to correct twice as many errors. The starting point is the following result given in [21, p. 341].

Theorem 1: A binary Goppa code $\mathscr{G}(\boldsymbol{x}, \gamma)$ associated to a Goppa polynomial $\gamma(z)$ of degree r without multiple roots is equal to the alternant code $\mathscr{A}_{2r}(\boldsymbol{x}, \boldsymbol{y})$ with $y_i = \gamma(x_i)^{-2}$.

Corollary 1 ([22]): There exists a polynomial time algorithm decoding all errors of weight at most r for any Goppa code $\mathscr{G}(\boldsymbol{x}, \gamma)$ where $\gamma(z)$ is of degree r and has no multiple roots.

It is worthwhile recalling that the only requirement for decoding a binary Goppa $\mathscr{G}$ is either to know $\boldsymbol{x}$ and $\gamma(z)$ *or* to know two vectors $\boldsymbol{x}^*$ and $\boldsymbol{y}^*$ such that:

$$\mathscr{G} = \mathscr{A}_{2r}(\boldsymbol{x}^*, \boldsymbol{y}^*). \tag{1}$$

B. Cryptographic Primitives Based on Binary Goppa Codes

The two most important public schemes that use binary Goppa codes are McEliece's encryption function and Courtois-Finiasz-Sendrier (CFS) [2] signature algorithm. We briefly recall here the general principle of McEliece's scheme. The key generation algorithm picks at random one $k \times n$ generator matrix $\boldsymbol{G}$ of a randomly picked binary Goppa code of $\mathscr{G}$ of degree r. The *secret* key is the decoding algorithm $\mathcal{D}$ associated to $\mathscr{G}$ and the *public* key is $\boldsymbol{G}$. To encrypt $\boldsymbol{u} \in \mathbb{F}_2^k$, the sender has to choose a random vector $\boldsymbol{e}$ in $\mathbb{F}_2^n$ of weight r and computes the ciphertext $\boldsymbol{c} \stackrel{\text{def}}{=} \boldsymbol{uG} + \boldsymbol{e}$. The receiver then recovers the plaintext by applying $\mathcal{D}$ on $\boldsymbol{c}$.

The CFS scheme also relies on binary Goppa codes. A user whose public key is $\boldsymbol{G}$ and who wishes to sign a message $\boldsymbol{x} \in \mathbb{F}_2^k$ has to compute a string $\boldsymbol{u}$ such that the Hamming weight of $\boldsymbol{x} - \boldsymbol{uG}$ is at most r. Anyone (a *verifier*) can publicly check the validity of a signature. Unfortunately, this approach can only provide signatures for messages $\boldsymbol{x}$ that are within distance r from a codeword $\boldsymbol{uG}$. The CFS scheme prompts 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. With a binary Goppa codes of length $n = 2^m$ and and dimension $k = n - mr$, the number of trials is of order $r!$. So one has to choose a very small r and therefore take a very large n in order to be secure. The code rate is then equal to $1 - \frac{1}{2^m}mr$ which is quite close to 1 for large n (that is for large values of m) and moderate values of r. For instance, a 80-bit security CFS scheme requires to take $n = 2^{21}$ and $r = 10$ whereas the McEliece cryptosystem for the same security needs to choose $n = 2^{11}$ and $r = 32$ ([18]). Thus one major difference between the McEliece cryptosystem and the CFS scheme lies in the choice of the parameters of the codes.

C. Goppa Code Distinguishing Problem

The minimum requirement for an encryption function is that it should be infeasible from a given ciphertext $\boldsymbol{c}$ and public data[2] like the public key, ciphertexts, *etc.* to recover the corresponding plaintext $\boldsymbol{x}$. This issue is directly linked to the following computational problem.

Definition 3 (McEliece Problem): Let $\boldsymbol{G}$ be a generator matrix of a binary Goppa code of length $n \leqslant 2^m$ and dimension $k = n - rm$ where m and t are positive integers. Let $\boldsymbol{x}$ be a vector from $\mathbb{F}_{2^m}^k$ and let $\boldsymbol{e}$ be a vector from $\mathbb{F}_{2^m}^n$ of weight t. Finally, we set $\boldsymbol{c} \stackrel{\text{def}}{=} \boldsymbol{xG} + \boldsymbol{e}$. Then the *McEliece Problem* asks to find $\boldsymbol{x}$ and $\boldsymbol{e}$ only from $\boldsymbol{G}$ and $\boldsymbol{c}$.

One obvious way of solving this problem consists in devising a method that recovers the private key. But, it is also possible to recover a plaintext from a specific ciphertext without resorting to a key-recovery attack. In particular, an attacker against the McEliece scheme would find the plaintext by applying general decoding methods like [3]–[11], [23]–[27] on the public matrix $\boldsymbol{G}$. Such attacks are called *decoding attacks*. It is possible $\mathcal{G}_{n,k}$ to precise formally all these notions. For any integers n and k with $k \leqslant n$, we denote by $\mathcal{G}_{n,k}$ the set of $k \times n$ generator matrices of binary Goppa codes. Similarly, $\mathcal{R}_{n,k}$ is the set of binary random generator $k \times n$ matrices. The set of words of weight t is denoted by $\mathcal{S}(\boldsymbol{0}, t)$.

[2]This kind of attack is called a *Chosen Plaintext Attack* (CPA).

Definition 4: An (T, ε)-*adversary* $\mathbb{A}$ against the McEliece cryptosystem is an algorithm that runs in time at most T such that:

$$\texttt{Suc}_{\mathsf{McE}}(\mathbb{A}) \stackrel{\text{def}}{=} \mathbf{Pr}_{\boldsymbol{G},\boldsymbol{m},\boldsymbol{e}}\Big\{\mathbb{A}(\boldsymbol{G}, \boldsymbol{m}\boldsymbol{G}+\boldsymbol{e}) = (\boldsymbol{m},\boldsymbol{e})\Big\} \leqslant \varepsilon$$

where the random choices are made such that $\boldsymbol{G} \in \mathcal{G}_{n,k}$, $\boldsymbol{m} \in \mathbb{F}_2^k$ and $\boldsymbol{e} \in \mathcal{S}(\boldsymbol{0}, t)$.

An algorithm $\mathbb{A}$ is a (T, ε)-*decoder* if it runs in time at most T such that:

$$\texttt{Suc}_{\mathsf{Rand}}(\mathbb{A}) \stackrel{\text{def}}{=} \mathbf{Pr}_{\boldsymbol{G},\boldsymbol{m},\boldsymbol{e}}\Big\{\mathbb{A}(\boldsymbol{G}, \boldsymbol{m}\boldsymbol{G}+\boldsymbol{e}) = (\boldsymbol{m},\boldsymbol{e})\Big\} \leqslant \varepsilon$$

where $\boldsymbol{G} \in \mathcal{R}_{n,k}$, $\boldsymbol{m} \in \mathbb{F}_2^k$ and $\boldsymbol{e} \in \mathcal{S}(\boldsymbol{0}, t)$.

Currently, the only known methods that aim to solve the McEliece problem are based either on an exhaustive search of the private key or on applying very general decoding methods. Both approaches run in exponential time on the length when the rate is fixed. But this situation is a still unsatisfactory because there is no certitude that there does not exist a better way to solve it.

A classical stance is to claim that binary Goppa codes look like random linear codes. It amounts to say that there does not exist a polynomial-time computable quantity which behaves differently depending on whether the code is a Goppa or a random code. Currently, it is an open problem to establish a formal proof that would substantiate the claim that a binary Goppa code is *indistinguishable* from a random code. This assumption is attractive because it enables to rely on the hardness of decoding a random linear code to prove the security of the McEliece function. This reasoning does make sense because binary Goppa codes share several common aspects[3] with a randomly picked linear code. Furthermore, all the general decoding algorithms do not exploit the information, even partially, that a matrix describes a "hidden" Goppa code. Based on this, the authors of [2] defined the *Goppa Code Distinguishing (GD) problem* and stated that *"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"*. We state now precisely the GD problem.

Definition 5 (Goppa Code Distinguishing (GD) Problem): An algorithm $\mathcal{D} : \mathbb{F}_2^{k\times n} \to \{0,1\}$ that takes as input a $k \times n$ matrix $\boldsymbol{G}$ and returns a bit solves the GD problem if it wins the following game:

1) $b \leftarrow \{0,1\}$
2) If $b = 1$ then $\boldsymbol{G} \leftarrow \mathcal{G}_{n,k}$ else $\boldsymbol{G} \leftarrow \mathcal{R}_{n,k}$
3) If $\mathcal{D}(\boldsymbol{G}) = b$ then $\mathcal{D}$ wins else $\mathcal{D}$ loses.

An algorithm $\mathcal{D} : \mathbb{F}_2^{k\times n} \to \{0,1\}$ is a (T, ε)-distinguisher if it runs in time at most T and such that:

$$\texttt{Adv}(\mathcal{D}) \stackrel{\text{def}}{=} \Big|\mathbf{Pr}_{\boldsymbol{G}\in\mathcal{G}_{n,k}}\left\{\mathcal{D}(\boldsymbol{G}) = 1\right\} - \mathbf{Pr}_{\boldsymbol{G}}\left\{\mathcal{D}(\boldsymbol{G}) = 1\right\}\Big|.$$

There is simple way to construct a distinguisher from an attacker (T, ε)-attacker $\mathbb{A}$ against the McEliece cryptosystem. We denote it by $\mathcal{D}_{\mathbb{A}}$ and it works as follows. On a given input $\boldsymbol{G}$, it randomly picks a couple $(\boldsymbol{m}, \boldsymbol{e})$ among $\mathbb{F}_2^k \times \mathcal{S}(\boldsymbol{0}, t)$ and then outputs $\mathcal{D}_{\mathbb{A}}(\boldsymbol{G}) = 1$ if $\mathbb{A}(\boldsymbol{G}, \boldsymbol{m}\boldsymbol{G} + \boldsymbol{e}) = (\boldsymbol{m}, \boldsymbol{e})$ and $\mathcal{D}_{\mathbb{A}}(\boldsymbol{G}) = 0$ otherwise. The running time of $\mathcal{D}_{\mathbb{A}}(\boldsymbol{G})$ is therefore uppe-bounded by the running time of $\mathbb{A}$ and its advantage is then equal to [28]:

$$\texttt{Adv}(\mathcal{D}_{\mathbb{A}}) = \Big|\texttt{Suc}_{\mathsf{Rand}}(\mathbb{A}) - \texttt{Suc}_{\mathsf{McE}}(\mathbb{A})\Big|.$$

This shows that if $\texttt{Adv}(\mathcal{D}_{\mathbb{A}}$ is very small (or negligible) then the chances that an attacker recovers a plaintext are also very small provided that the problem of decoding a random linear code is hard. So the difficulty of the GD problem guarantees that there is no polynomial-time algorithm that solves the McEliece problem.

Until our recent work in [1] and this paper, the only known algorithm for solving GD enumerates binary Goppa codes and tests the code equivalence thanks to the *Support Splitting* algorithm [29]. This approach runs in time $O\left(\frac{1}{mr}n^{r-1}\right)$ for binary Goppa codes of degree r and length n with $m \leqslant \log_2 n$. Another possible approach was proposed in [30] which shows that Quantum Fourier Sampling (QFS) can also be used for solving the GD problem. However, it turns out that QFS has a negligible advantage against the GD problem. Thus GD problem seems immune against QFS unlike classical cryptographic problems such as factoring integers and discrete logarithms which can be solved in (quantum) poly-time thanks to QFS [31]. This does not contradict the existence of a (classical) poly-time algorithm solving GD. In this paper, we show how to exploit the algebraic structure of Goppa codes to construct a classical poly-time distinguisher with optimal advantage under some conditions on the rate of the code.

[3]Similarly to random codes, Goppa codes asymptotically meet the Gilbert-Varshamov bound. They have also a trivial permutation group like random codes.

D. Semantically Secure Conversions

The fundamental issue when dealing with cryptographic primitives is to prove its security. A first approach is to show that the primitive resists to the best known attacks. However, this does not guarantee that there will not appear one day a better attack that renders the primitive insecure. The methodology of *security proof by reduction* addresses this question by linking a security notion that a cryptographic primitive should verify to an algorithmic problem widely considered as hard. The approach is similar to the one that proves the NP-Completeness of a given problem. Such a "security proof" proves that if an attacker exists then it can be used as a subroutine to solve a hard problem. In other words, such an attacker has little chances to exist.

These simple facts prompt to design conversions that would lead to an IND-CCA secure encryption scheme. The first article to propose such a conversion for the McEliece cryptosystem is [32] which proposes a conversion resulting into an IND-CCA2 in the *Random Oracle Model* under the assumption that the problem of decoding random linear codes is difficult. This work was then followed by [33] which proposes another modification while providing an IND-CPA secure encryption scheme in the standard model[4] under the assumptions that both decoding random linear codes *and* distinguishing Goppa codes are difficult problems. Finally, under the same assumptions, [34] proposed (a modified) McEliece cryptosystem that is IND-CCA2 in the *standard model*.

III. A Distinguisher of Alternant and Goppa Codes

The McEliece cryptosystem relies on binary Goppa codes which belong to the class of *alternant codes*. We are now able to construct an algebraic system as explained in [19] for a key-recovey. This algebraic system will be the main ingredient for building a distinguisher. We assume that the public matrix is a $k \times n$ generator matrix $\boldsymbol{G}$ where by assumption $k = n - rm$ and such that it defines an alternant code of degree r. We know that the knowledge of a matrix $\boldsymbol{V}_r(\boldsymbol{x}^*, \boldsymbol{y}^*)$ for some vectors $\boldsymbol{x}^*$ and $\boldsymbol{y}^*$ allows to efficiently decode the public code defined by $\boldsymbol{G}$. Furthermore, from the definition of $\boldsymbol{G}$, we also know:

$$\boldsymbol{V}_r(\boldsymbol{x}^*, \boldsymbol{y}^*)\boldsymbol{G}^T = \boldsymbol{0}.$$

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 [19] of the following polynomial system:

$$\bigcup_{e=0}^{r-1} \left\{ \sum_{j=1}^{n} g_{i,j} Y_j X_j^e = 0 \;\mid\; 1 \leqslant i \leqslant k \right\} \tag{2}$$

where the $g_{i,j}$'s are the entries of the known matrix $\boldsymbol{G}$. Clearly, solving this system would lead to a possibly equivalent private key. For compact variants [35], [36] of McEliece [3], additional structures permit to drastically reduce the number of variables allowing to solve (2) for a large set of parameters in polynomial-time using dedicated Gröbner bases techniques [19]. But the general case is currently a major *open question*. However, we describe a simple way for *partially* solving (2). It basically consists in deriving a linear system from the polynomial system (2). Note that this operation is actually the first step performed during the computation of Gröbner bases algorithms such as by F_4 or F_5 [37], [38]. From now on, we will always assume that $q = 2^s$ with $s \geqslant 1$. We can assume that $\boldsymbol{G} = (g_{ij})$ with $1 \leqslant i \leqslant k$ and $1 \leqslant j \leqslant n$ is in reduced row echelon form over its k first positions $\boldsymbol{G} = (\boldsymbol{I}_k \mid \boldsymbol{P})$ where $\boldsymbol{P} = (p_{ij})$ for $1 \leqslant i \leqslant k$ and $k+1 \leqslant j \leqslant n$ is the submatrix of $\boldsymbol{G}$ formed by its last $n - k = mr$ columns. Next, for any $i \in \{1, \dots, k\}$ and $e \in \{0, \dots, r-1\}$, we can rewrite (2) as $Y_i X_i^e = \sum_{j=k+1}^{n} p_{i,j} Y_j X_j^e$. Then, thanks to the trivial identity $Y_i(Y_i X_i^2) = (Y_i X_i)^2$ for all i in $\{1, \dots, k\}$, we get:

$$\sum_{j=k+1}^{n} p_{i,j} Y_j \sum_{j=k+1}^{n} p_{i,j} Y_j X_j^2 = \left(\sum_{j=k+1}^{n} p_{i,j} Y_j X_j \right)^2.$$

We thus obtain a linear system $\mathcal{L}_{\boldsymbol{P}}$ of k equations involving $\binom{mr}{2}$ variables $Z_{jj'} \stackrel{\text{def}}{=} Y_j Y_{j'} X_{j'}^2 + Y_{j'} Y_j X_j^2$ which is as follows:

$$\mathcal{L}_{\boldsymbol{P}} \stackrel{\text{def}}{=} \begin{cases} \displaystyle\sum_{j=k+1}^{n-1} \sum_{j'>j}^{n} p_{1,j} p_{i,j'} Z_{jj'} = 0 \\ \qquad\vdots \\ \displaystyle\sum_{j=k+1}^{n-1} \sum_{j'>j}^{n} p_{k,j} p_{i,j'} Z_{jj'} = 0 \end{cases} \tag{3}$$

Definition 6: For any integer $r \geqslant 1$ and $m \geqslant 1$, the number of variables $\binom{mr}{2}$ in the linear system $\mathcal{L}_{\boldsymbol{P}}$ as defined in (3) is denoted by N and its rank by $\texttt{rank}(\mathcal{L}_{\boldsymbol{P}})$. We denote by $\texttt{Ker}(\mathcal{L}_{\boldsymbol{P}})$ the kernel of $\mathcal{L}_{\boldsymbol{P}}$ and its dimension as a $\mathbb{F}_q$-vector space is denoted by D.

[4]There is no hash function in this model.

Let us recall that $\mathtt{Ker}(\mathcal{L}_{\boldsymbol{P}})$ is necessarily a $\mathbb{F}_q$-vector space since the linear system (3) have coefficients in $\mathbb{F}_q$ but the solutions of (2) are sought in the extension field $\mathbb{F}_{q^m}$. Furthermore, we obviously have $D = N - \mathtt{rank}(\mathcal{L}_{\boldsymbol{P}})$. Hence, in order to recover the solutions of (2), it is necessary that $\mathtt{rank}(\mathcal{L}_{\boldsymbol{P}})$ is almost equal to the number of variables $N = \binom{mr}{2}$. For a random system, this is likely to happen when the number k of equations in (3) is greater than the number of unknowns, that is to say $k \geqslant N$. It appears experimentally that D is amazingly *large even in the case where* $k \geqslant N$. It even depends on whether or not the code with generator matrix $\boldsymbol{G}$ is chosen as a (generic) alternant code or as a Goppa code. Interestingly enough, when $\boldsymbol{G}$ is chosen at random, $\mathtt{rank}(\mathcal{L}_{\boldsymbol{P}})$ is equal to $\min\{k, N\}$ with very high probability. In particular, the dimension of the solution space is typically 0 when k is larger than the number of variables N as one would expect. This will be proved in Section VI. Although this *defect* in the rank is an obstacle to break the McEliece cryptosystem, it can be used to distinguish the public generator of a *structured* code from a random code.

We consider three cases. First, when the p_{ij}'s are chosen uniformly and independently at random in $\mathbb{F}_q$ then we denote by D_{random} the dimension of $\mathtt{Ker}(\mathcal{L}_{\boldsymbol{P}})$. When $\boldsymbol{G}$ is chosen as a generator matrix of a random alternant (*resp.* Goppa) code of degree r, we denote it by $D_{\text{alternant}}$ (*resp.* D_{Goppa}). We carried out intensive computations with Magma [39] by randomly generating alternant and Goppa codes over the field $\mathbb{F}_q$ with $q \in \{2, 4, 8, 16, 32\}$ for r in the range $\{3, \dots, 50\}$ and several values of m. Furthermore, in our probabilistic model, a random alternant code is obtained by picking uniformly and independently at random two vectors $(x_1, \dots, x_n)$ and $(y_1, \dots, y_n)$ from $(\mathbb{F}_{q^m})^n$ such that the x_i's are all different and the y_i's are all nonzero. A random Goppa code is obtained by taking a random vector $(x_1, \dots, x_n)$ in $(\mathbb{F}_{q^m})^n$ with all the x_i's different and a random *irreducible* polynomial $\gamma(z) = \sum_i \gamma_i z^i$ of degree r. Our experiments have revealed that the dimension of $\mathtt{Ker}(\mathcal{L}_{\boldsymbol{P}})$ is *predictable* and follows formulas.

Experimental Fact 1 (Alternant Case): As long as $N - D_{\text{alternant}} < k$, $D_{\text{alternant}}$ is equal with high probability to:

$$T_{\text{alternant}} \stackrel{\text{def}}{=} \frac{1}{2} m(r-1)\left((2e+1)r - 2\frac{q^{e+1}-1}{q-1}\right) \tag{4}$$

with $e \stackrel{\text{def}}{=} \left\lfloor \log_q(r-1) \right\rfloor$.

Experimental Fact 2 (Goppa Case): As long as $N - D_{\text{Goppa}} < k$ then D_{Goppa} is equal with high probability to T_{Goppa} which is defined when $r < q - 1$ as:

$$T_{\text{Goppa}} \stackrel{\text{def}}{=} \frac{1}{2} m(r-1)(r-2) = T_{\text{alternant}} \tag{5}$$

and when $r \geqslant q - 1$:

$$T_{\text{Goppa}} \stackrel{\text{def}}{=} \frac{1}{2} mr\Big((2e+1)r - 2(q-1)q^{e-1} - 1\Big) \tag{6}$$

with e being the unique integer such that:

$$(q-1)^2 q^{e-2} < r \leqslant (q-1)^2 q^{e-1}.$$

We gathered in Appendix A some experimental results obtained through intensive computations with the Magma system [39].

IV. Alternant Case

The goal of this section is to explain the value of the dimension $D_{\text{alternant}}$ of $\mathtt{Ker}(\mathcal{L}_{\boldsymbol{P}})$ for q-ary alternant codes of degree r. We shall see that this dimension will be obtained by first identifying a $\mathbb{F}_{q^m}$-basis of $\mathtt{Ker}(\mathcal{L}_{\boldsymbol{P}})$ when viewed as a linear system with coefficients in $\mathbb{F}_{q^m}$. To set up the linear system $\mathcal{L}_{\boldsymbol{P}}$ as defined in (3), we have used the trivial identity $Y_i Y_i X_i^2 = (Y_i X_i)^2$. The fundamental remark is that we can use any identity $Y_i X_i^a Y_i X_i^b = Y_i X_i^c Y_i X_i^d$ with $a, b, c, d \in \{0, 1, \dots, r-1\}$ such that $a + b = c + d$. Such identities lead to the same algebraic system $\mathcal{L}_{\boldsymbol{P}}$:

$$\begin{aligned} \sum_{(j,j') \in J} p_{i,j} p_{i,j'} \Big(& Y_j X_j^a Y_{j'} X_{j'}^b + Y_{j'} X_{j'}^a Y_j X_j^b \\ & + Y_j X_j^c Y_{j'} X_{j'}^d + Y_{j'} X_{j'}^c Y_j X_j^d \Big) = 0 \end{aligned} \tag{7}$$

where we have set :

$$J \stackrel{\text{def}}{=} \left\{ (j, j') \in \mathbb{N} \times \mathbb{N} \mid k+1 \leqslant j < j' \leqslant n \right\}.$$

The fact that *there are many different ways of combining equations together yielding the same linear system* $\mathcal{L}_{\boldsymbol{P}}$ explains why the dimension of $\mathtt{Ker}(\mathcal{L}_{\boldsymbol{P}})$ is large. In what follows, we exhibit further elements of $\mathtt{Ker}(\mathcal{L}_{\boldsymbol{P}})$ thanks to the automorphisms $x \longmapsto x^{q^\ell}$ where ℓ is in $\{0, \dots, m-1\}$. Indeed, we can also consider the identity:

$$(Y_i X_i^a)^{q^{\ell'}} (Y_i X_i^b)^{q^\ell} = (Y_i X_i^c)^{q^{\ell'}} (Y_i X_i^d)^{q^\ell} \tag{8}$$

for any integers a, b, c, d, ℓ and ℓ' such that $aq^{\ell'} + bq^{\ell} = cq^{\ell'} + dq^{\ell}$. We get again the linear system $\mathcal{L}_{\boldsymbol{P}}$. However, assuming that $\ell' \leqslant \ell$, solutions obtained from such equations are exactly those coming from the identity:

$$Y_i X_i^a Y_i^{q^{\ell-\ell'}} X_i^{bq^{\ell-\ell'}} = Y_i X_i^c Y_i^{q^{\ell-\ell'}} X_i^{dq^{\ell-\ell'}}. \tag{9}$$

We can focus on vectors that satisfy equations obtained with $0 \leqslant a, b, c, d < r$, $0 \leqslant \ell < m$ and $a + q^{\ell} b = c + q^{\ell} d$. Without loss of generality, we can assume that $d > b$ and let us set $\delta = d - b$. Moreover, the equality $a + q^{\ell} b = c + q^{\ell} d$ implies that $a = c + q^{\ell}\delta$.

We now try to determine the number of linearly independent solutions induced by such identities. For the sake of simplicity, we denote by $\boldsymbol{Z} = (Z_{j,j'})_{(j,j')\in J}$ the vector that is obtained from the identity $a + q^{\ell} b = c + q^{\ell} d$. One can show that:

$$Z_{j,j'} = \left(X_j^{\delta} + X_{j'}^{\delta}\right)^{q^{\ell}} \left(Y_j Y_{j'}^{q^{\ell}} X_j^c X_{j'}^{q^{\ell} b} + Y_{j'} Y_j^{q^{\ell}} X_{j'}^c X_j^{q^{\ell} b}\right).$$

Thus any solution obtained by the tuple (a, b, c, d, ℓ) is uniquely described by (b, c, δ, ℓ) by setting $d = b + \delta$ and $a = c + q^{\ell}\delta$ provided that $1 \leqslant \delta \leqslant r - 1 - b$, $0 \leqslant b \leqslant r - 2$ and $0 \leqslant c + q^{\ell}\delta \leqslant r - 1$. In the sequel, we set $e \stackrel{\text{def}}{=} \lfloor \log_q(r-1) \rfloor$ so that $0 \leqslant \ell \leqslant e$. We will show that $\boldsymbol{Z}$ can be expressed as a linear combination of some solutions obtained thanks to very specific identities. Firstly, let us denote by $\boldsymbol{B}(b, c, \ell)$ *the solution* obtained form (b, c, δ, ℓ) with $\delta = 1$. We define:

$$\mathcal{B}_r \stackrel{\text{def}}{=} \bigcup_{\substack{0 \leqslant b \leqslant r-2 \\ 0 \leqslant \ell \leqslant e}} \left\{ \boldsymbol{B}(b, c, \ell) \neq \boldsymbol{0} \;\mid\; 0 \leqslant c \leqslant r - 1 - q^{\ell} \right\}.$$

It is easy to see that if $\boldsymbol{B}(b, c, 0)$ belongs to $\mathcal{B}_r$ then $0 \leqslant b < c \leqslant r-2$. We denote by $< \mathcal{B}_r >$ the vector space spanned by $\mathcal{B}_r$. We are now in position to show that $\boldsymbol{Z}$ belongs to $< \mathcal{B}_r >$. First notice from Equation (7) that if $\boldsymbol{W}$ is the solution obtained from $a + q^{\ell} b = g + q^{\ell} f$ for some integers g and f then $\boldsymbol{Z} + \boldsymbol{W}$ is the vector that would be obtained from $c + q^{\ell} d = g + q^{\ell} f$. In particular, the identity $c + q^{\ell}\delta + bq^{\ell} = c + q^{\ell}(b + \delta)$ can also be rewritten as $c + q^{\ell}\delta + bq^{\ell} = c + q^{\ell}(\delta - 1) + (b + 1)q^{\ell}$. Therefore, we see that $\boldsymbol{Z} = \boldsymbol{B}(b, c + q^{\ell}(\delta - 1), \ell) + \boldsymbol{V}$ where $\boldsymbol{V}$ is the solution obtained from the identity

$$c + q^{\ell}(\delta - 1) + (b + 1)q^{\ell} = c + q^{\ell}(b + \delta).$$

Hence by induction on δ we can prove that:

$$\boldsymbol{Z} = \sum_{i=1}^{\delta} \boldsymbol{B}\Big(b + i - 1, c + q^{\ell}(\delta - i), \ell\Big). \tag{10}$$

Proposition 2: For all $r \geqslant 3$, let us denote by $|\mathcal{B}_r|$ the cardinality of $\mathcal{B}_r$ then:

$$T_{\text{alternant}} = m|\mathcal{B}_r|.$$

Proof: The number of elements in $\mathcal{B}_r$ is given by the number of possible tuples (b, c, ℓ) that is to say:

$$\begin{aligned} |\mathcal{B}_r| &= \frac{1}{2}(r-1)(r-2) + \sum_{\ell=1}^{e}\sum_{b=0}^{r-2}(r - q^{\ell}) \\ &= \frac{1}{2}(r-1)\left(r - 2 + 2er - 2\sum_{\ell=1}^{e} q^{\ell}\right) = \frac{1}{m} T_{\text{alternant}}. \end{aligned}$$

■

This proposition shows that $\mathcal{B}_r$ is a $\mathbb{F}_{q^m}$-basis that provides a $\mathbb{F}_q$-basis of $\mathtt{Ker}(\mathcal{L}_{\boldsymbol{P}})$ allowing us to suggest an heuristic: consider an arbitrary decomposition of the elements of $\mathbb{F}_{q^m}$ in a $\mathbb{F}_q$ basis. Let $\pi_i : \mathbb{F}_{q^m} \to \mathbb{F}_q$ be the function giving the i-th coordinate in this decomposition with $1 \leqslant i \leqslant m$. By extension we denote for $\boldsymbol{z} = (z_j)_{1 \leqslant j \leqslant n} \in (\mathbb{F}_{q^m})^n$ by $\pi_i(\boldsymbol{z})$ the vector $(\pi_i(z_j))_{1 \leqslant j \leqslant n} \in \mathbb{F}_q^n$.

Heuristic 1: For any j such that $1 \leqslant j \leqslant n$ and for random choices of x_j's and y_j's then $\bigcup_{1 \leqslant i \leqslant m} \left\{ \pi_i(\boldsymbol{Z}) \;\mid\; \boldsymbol{Z} \in \mathcal{B}_r \right\}$ forms a basis of $\mathtt{Ker}(\mathcal{L}_{\boldsymbol{P}})$.

V. BINARY GOPPA CASE

In this section, we will investigate the case of binary Goppa codes. Notice that for q-ary Goppa codes of degree $r < q - 1$ we have observed that $T_{\text{alternant}} = T_{\text{Goppa}}$ because (4) simplifies to:

$$\frac{1}{2}m(r-1)(r-2) \stackrel{\text{def}}{=} T_{\text{Goppa}}.$$

This is due to the fact that $e = 0$ when $r < q-1$. We leave as an open question the proof that q-ary Goppa codes of degree $r < q-1$ behave for our distinguisher as alternant codes. We focus on the classical case in code-based cryptography of binary Goppa codes.

The goal is to identify a basis of $\mathtt{Ker}(\mathcal{L}_{\boldsymbol{P}})$ for binary Goppa codes of degree r. We assume therefore that $q = 2$. In that special case, the theoretical expression T_{Goppa} (Experimental Fact 2) has a simpler expression.

Proposition 3: Let $e = \lceil \log_2 r \rceil + 1$. When $q = 2$ then (6) can be simplified to:

$$T_{\text{Goppa}} = \frac{1}{2} mr \Big((2e+1)r - 2^e - 1 \Big).$$

Theorem 1 shows that a binary Goppa code of degree r can be regarded as a binary alternant code of degree $2r$. This seems to indicate that we should have

$$D_{\text{Goppa}}(r) = T_{\text{alternant}}(2r).$$

This is not the case though because it turns out that $D_{\text{Goppa}}(r)$ is significantly smaller than this. In our experiments, we have found out that the vectors of $\mathcal{B}_{2r}$ still form a generating set for $\mathtt{Ker}(\mathcal{L}_{\boldsymbol{P}})$. Unfortunately, they are not independent anymore. Our goal is therefore to identify the additional dependencies occuring in $\mathcal{B}_{2r}$. We will see that many of them come from $\mathbb{F}_{2^m}$-relations induced by the Goppa polynomial $\gamma(z) = \sum_{i=0}^{r} \gamma_i z^i$ with $\gamma_r \neq 0$. Recall that by definition $Y_i = \gamma(X_i)^{-2}$. This fact will allow to derive two types of linear dependencies. The first type of linear relations is rather natural, whilst the second type is more subtle. In the sequel we set $u \stackrel{\text{def}}{=} \lfloor \log_2(2r-1) \rfloor$.

A. Linear Dependencies over $\mathbb{F}_{2^m}$

We derive a first set of linear dependencies induced by the Goppa polynomial $\gamma(X)$.

Proposition 4: Let t, ℓ and c be such that $0 \leqslant t \leqslant r-2$, $1 \leqslant \ell \leqslant u$ and $0 \leqslant c \leqslant 2r - 2^\ell - 1$ then it holds:

$$\begin{aligned} &\sum_{b=0}^{r} \gamma_b^{2^\ell} \boldsymbol{B}(t+b, c, \ell) = \\ &\boldsymbol{B}(2t, c + 2^{\ell-1}, \ell-1) + \boldsymbol{B}(2t+1, c, \ell-1). \end{aligned} \tag{11}$$

Proof: Let us set $\boldsymbol{V} = (V_{i,j'})_{(i,j') \in J}$ as the following:

$$\boldsymbol{V} \stackrel{\text{def}}{=} \sum_{b=0}^{r} \gamma_b^{2^\ell} \boldsymbol{B}(t+b, c, \ell).$$

The equality $Y_j \gamma(X_j)^2 = 1$ clearly implies that $\sum_{b=0}^{r} \gamma_b^2 Y_j^2 X_j^{2(b+t)} = Y_j X_j^{2t}$ and therefore:

$$V_{j,j'} = (X_j + X_{j'})^{2^\ell} \left(Y_j Y_{j'}^{2^{\ell-1}} X_j^c X_{j'}^{2^\ell t} + Y_{j'} Y_j^{2^{\ell-1}} X_{j'}^c X_j^{2^\ell t} \right).$$

One can check that $\boldsymbol{V}$ is exactly the vector that we would obtain from the identity $a^* + 2^{\ell-1} b^* = c + 2^{\ell-1}(b^*+2)$ with $a^* \stackrel{\text{def}}{=} c + 2^\ell$ and $b^* \stackrel{\text{def}}{=} 2t$. Hence by (10) the vector $\boldsymbol{V}$ can be written as:

$$\boldsymbol{V} = \boldsymbol{B}(2t, c + 2^{\ell-1}, \ell-1) + \boldsymbol{B}(2t+1, c, \ell-1).$$

This last equality terminates the proof. ■

Consequently $\mathcal{B}_{2r}$ cannot provide a basis for $\mathtt{Ker}(\mathcal{L}_{\boldsymbol{P}})$ as in the alternant case. It even possible to count the number of linear dependencies predicted by Proposition 4. Indeed, each equation is defined by a *unique* (t, c, ℓ) such that $0 \leqslant t \leqslant r-2, 1 \leqslant \ell \leqslant u$ and $0 \leqslant c \leqslant 2r - 2^\ell - 1$. So if N_L is the number of equations of the form (11) then clearly we have:

$$N_L = 2(r-1)\left(ru + 1 - 2^u\right). \tag{12}$$

B. Additional Relations Inducing Dependencies over $\mathbb{F}_2$

We exhibit new linear dependencies between some elements of $\mathcal{B}_{2r}$ and the vectors $\boldsymbol{B}^2(b, c, \ell)$.

Proposition 5: For any ℓ, b and t such that $0 \leqslant \ell \leqslant u-1$, $0 \leqslant b \leqslant 2r-2$ and $0 \leqslant t \leqslant r - 1 - 2^\ell$, we have:

$$\boldsymbol{B}(b, 2t, \ell+1) = \sum_{c=0}^{r} \gamma_c^2 \boldsymbol{B}^2(b, t+c, \ell). \tag{13}$$

Proof: Let us set $\boldsymbol{V} = (V_{i,j'})_{(i,j') \in J}$ as the vector:

$$\boldsymbol{V} \stackrel{\text{def}}{=} \sum_{c=0}^{r} \gamma_c^2 \boldsymbol{B}^2(b, t+c, \ell).$$

On can see that for all $(j,j') \in J$:

$$V_{j,j'} = (X_j + X_{j'})^{2^{\ell+1}} \Big(Y_j Y_{j'}^{2^{\ell+1}} X_j^{2t} X_{j'}^{2^{\ell+1}b} + Y_{j'} Y_j^{2^{\ell+1}} X_{j'}^{2t} X_j^{2^{\ell+1}b} \Big). \tag{14}$$

We recognize that $\boldsymbol{V}$ is exactly the expression of $\boldsymbol{B}(b, 2t, \ell+1)$ and hence the proposition is proved. ■

We can count the number of linearly dependencies predicted by Proposition 5. Let N_Q be the number of vectors of $\mathcal{B}_{2r}$ satisfying Equation (13). By Proposition 5 N_Q is exactly the number of (ℓ, b, t) so that:

$$N_Q = (2r-1)(ru - 2^u + 1). \tag{15}$$

C. Counting the Exact Number of Linear Dependencies over $\mathbb{F}_2$

We now want to count the number of linear dependencies induced by Proposition 5 and Proposition 4. The difficulty is that some of the N_Q vectors of $\mathcal{B}_{2r}$ are counted twice because they appear both in linear relations of the form (11) and "quadratic" equations of the form (13). Let $N_{L\cap Q}$ be the number of such vectors. More precisely, let $\mathcal{Q}_{2r}$ be the subset of vectors $\boldsymbol{B}(b,c,\ell)$ of $\mathcal{B}_{2r}$ which are involved in an Equation of type (13). Remark in that case c has to be even and $\ell \geqslant 1$. Furthermore, there are equations of type (11) which involve only vectors of $\mathcal{Q}_{2r}$. Let N_1 be their number. Moreover, it is possible by adding two equations of type (11) involving at least one vector which is not in $\mathcal{Q}_{2r}$ to obtain an equation which involves only vectors of $\mathcal{Q}_{2r}$. Let N_0 be the number of such sums and let $N_{L\cap Q} \stackrel{\text{def}}{=} N_1 + N_0$. Our goal is to prove that:

$$N_{L\cap Q} = (r-1)\Big((u - \frac{1}{2})r - 2^u + 2\Big). \tag{16}$$

Proof: We will consider vectors of $\mathcal{Q}_{2r}$, that is to say vectors of $\mathcal{B}_{2r}$ that satisfy Equation (13), such that there exists a linear relation that link them. In other words, we consider all the linear relations of the form:

$$\sum_i \alpha_i \boldsymbol{B}(b_i, c_i, \ell_i) = 0$$

with α_i in $\mathbb{F}_{2^m}$ and where each $\boldsymbol{B}(b_i, c_i, \ell_i)$ is equal to a linear relation of the form (13).

One can observe that for such vectors we necessarily have c_i even and $1 \leqslant \ell_i \leqslant u$. In particular, an equation of the form of Equation (11) will involve only vectors of $\mathcal{Q}_{2r}$ if and only if $\ell - 1 \geqslant 1$ and c is even. Let us recall that the number of such equations is N_1, and consequently, its value is $\sum_{t=0}^{r-2}\sum_{\ell=2}^{u}\frac{1}{2}(2r-2^\ell)$, namely:

$$N_1 = (r-1)\Big((u-1)r - 2^u + 2\Big). \tag{17}$$

On the other hand when $\ell = 1$ and c is even, say for instance $c = 2t'$ then Equation (11) becomes:

$$\sum_{b=0}^{r} \gamma_b^2 \boldsymbol{B}(t+b, 2t', 1) = \boldsymbol{B}(2t, 2t'+1, 0) + \boldsymbol{B}(2t+1, 2t', 0).$$

Notice that we always have $\boldsymbol{B}(b,c,0) = \boldsymbol{B}(c,b,0)$. So if $t' = t$ then $\boldsymbol{B}(2t, 2t+1, 0) + \boldsymbol{B}(2t+1, 2t, 0) = \boldsymbol{0}$ and when $t \neq t'$ we obviously have:

$$\sum_{b=0}^{r} \gamma_b^2 \boldsymbol{B}(t+b, 2t', 1) = \sum_{b=0}^{r} \gamma_b^2 \boldsymbol{B}(t'+b, 2t, 1).$$

In both case, we get new linear dependencies between vectors of $\mathcal{Q}_{2r}$ different from those already obtained.In conclusion, N_0 is exactly the number of sets $\{t, t'\}$. By assumption t and t' have to satisfy $0 \leqslant t \leqslant r-2$ and $c = 2t'$ with $0 \leqslant c \leqslant 2r-3$, which implies that $0 \leqslant t' \leqslant r-2$. N_0 is therefore the number of couples (t,t') such that $0 \leqslant t \leqslant t' \leqslant r-2$. By gathering all the cases we have proved that:

$$N_{L\cap Q} = (r-1)\left((u-1)r - 2^u + 2\right) + \frac{1}{2}(r-1)r. \tag{18}$$

■

Proposition 6: For any integer $r \geqslant 2$, we have:

$$\begin{aligned} \frac{1}{m} T_{\text{Goppa}}(r) &= |\mathcal{B}_{2r}| - N_L - N_Q + N_{L\cap Q} \\ &= \frac{1}{2} r \left((2u+3)r - 2^{u+1} - 1\right). \end{aligned}$$

Proof: From Equation (4) and gathering all the equalities:

$$|\mathcal{B}_{2r}| - (N_L + N_Q - N_{L\cap Q}) = r\left((u + \frac{3}{2})r - 2^u - \frac{1}{2}\right).$$

Moreover, Proposition 3 gives:

$$\frac{1}{m}T_{\text{Goppa}}(r) = \frac{1}{2}r\left((2e+1)r - 2^e - 1\right)$$

where $e = \lceil \log_2 r \rceil + 1$. Using the basic inequality $2r-1 < 2r < 2(2r-1)$, we have therefore $\log_2(2r-1) < \log_2(r)+1 < \log_2(2r-1)+1$. This implies $\lceil \log_2 r \rceil = u$ and thus $\frac{1}{m}T_{\text{Goppa}}(r) = \frac{1}{2}r\left((2u+3)r - 2^{u+1} - 1\right)$. ∎

VI. Random Case

The purpose of this section is to study the behavior of D_{random}, namely the dimension of $\mathtt{Ker}(\mathcal{L}_{\boldsymbol{P}})$ as $\mathbb{F}_q$-vector space when the entries of the matrix P are drawn independently from the uniform distribution over $\mathbb{F}_q$. In this case, we can show that:

Theorem 2: Assume that $N \leqslant k$ and that the entries of P are drawn independently from the uniform distribution over $\mathbb{F}_q$. Then for any function $\omega(x)$ tending to infinity as x goes to infinity, we have

$$\mathbf{Pr}\Big(D_{\text{random}} \geqslant mr\omega(mr)\Big) = o(1),$$

as mr goes to infinity.

Notice that if we choose $\omega(x) = \log(x)$ for instance, then asymptotically the dimension D_{random} of the solution space is with very large probability smaller than $mr\log(mr)$. When m and r are of the same order (which is generally chosen in practice) this quantity is smaller than $D_{\text{alternant}}$ or D_{Goppa} which are of the form $\Omega(mr^2)$.

The main ingredient for proving Theorem 2 consists in analyzing a certain (partial) Gaussian elimination process on the matrix:

$$M \stackrel{\text{def}}{=} \Big(p_{ij}p_{ij'}\Big)_{\substack{1\leqslant i\leqslant k\\ k+1\leqslant j<j'\leqslant n}}.$$

We can see the matrix M in block form, each block consists of the matrix:

$$B_j = \Big(p_{i,k+j}p_{i,k+j'}\Big)_{\substack{1\leqslant i\leqslant k\\ 1\leqslant j<j'\leqslant n-k}}.$$

Each block B_j is of size $k \times (rm-j)$. Notice that in B_j, the rows for which $p_{i,k+j} = 0$ consist only of zeros. To start the Gaussian elimination process with B_1, we will therefore choose $rm-1$ rows for which $p_{i,k+1} \neq 0$. This gives a square matrix M_1. We perform Gaussian elimination on M by adding rows involved in M_1 to put the first block B_1 in standard form. We continue this process with B_2 by picking now $rm-2$ rows which have not been chosen before and which correspond to $p_{i,k+2} \neq 0$. This yields a square submatrix M_2 of size $rm-2$ and we continue this process until we reach the last block. The key observation is that:

$$\mathtt{rank}(M) \geqslant \mathtt{rank}(M_1) + \cdots + \mathtt{rank}(M_{rm-1}).$$

A rough analysis of this process yields Theorem 2. The important point is that what happens for different blocks are independent processes and it corresponds to looking at different rows of the matrix P. We give all the previous results that we need in order to prove Theorem 2.

It will be convenient to assume that the columns of M are ordered lexicographically. The index of the first column is $(j,j') = (k+1,k+2)$, the second one is $(j,j') = (k+1,k+3)$, while the last one is $(j,j') = (n-1,n)$. The matrices M_i's which are involved in the Gaussian elimination process mentioned above are defined inductively as follows. Let E_1 be the subset of $\{1,\ldots,k\}$ of indices s such that $p_{s,k+1} \neq 0$. Let F_1 be the subset of E_1 formed by its first $rm-1$ elements (if these elements exist). Now , we set

$$M_1 \stackrel{\text{def}}{=} \left(p_{s,k+1}p_{s,j}\right)_{\substack{s\in F_1\\ k+1<j\leqslant n}}. \tag{19}$$

Let r_1 be the rank of M_1. To simplify the discussion, we assume that:

1) $F_1 = \{1,2,\ldots,rm-1\}$,
2) the submatrix N_1 of M_1 formed by its first r_1 rows and columns is of full rank.

Note that we can always assume this by performing suitable row and column permutations. In other words M has the following block structure:

$$M = \begin{pmatrix} N_1 & B_1 \\ A_1 & C_1 \end{pmatrix}.$$

We denote:

$$M^{(1)} \stackrel{\text{def}}{=} \begin{pmatrix} N_1^{-1} & O \\ -A_1N_1^{-1} & I \end{pmatrix} M,$$

where O is a matrix of size $r_1 \times (k-r_1)$ with only zero entries and I is the identity matrix of size $k-r_1$. Notice that $M^{(1)}$ takes the block form:

$$M^{(1)} = \begin{pmatrix} I & B_1' \\ O & C_1' \end{pmatrix}.$$

This is basically performing Gaussian elimination on M in order to have the first r_1 columns in standard form. We then define inductively the $E_i, F_i, M_i, M^{(i)}$ and N_i as follows:

$$E_i \stackrel{\text{def}}{=} \left\{s \mid 1 \leqslant s \leqslant k, p_{s,k+i} \neq 0\right\} \setminus \bigcup_{u=1}^{i-1} F_{i-u},$$

$$F_i \stackrel{\text{def}}{=} \text{the first } rm - i \text{ elements of } E_i.$$

M_i is the submatrix of $M^{(i-1)}$ obtained from the rows in F_i and the columns associated to the indices of the form $(k+i, j')$ where j' ranges from $k+i+1$ to n. $M^{(i)}$ is obtained from $M^{(i-1)}$ by first choosing a square submatrix N_i of M_i of full rank and with the same rank as M_i and then by performing Gaussian elimination on the rows in order to put the columns of $M^{(i-1)}$ involved in N_i in standard form (i.e., the submatrix of $M^{(i-1)}$ corresponding to N_i becomes the identity matrix while the other entries in the columns involved in N_i become zero). It is clear that the whole process leading to $M^{(rm-1)}$ amounts to perform (partial) Gaussian elimination to M. Hence:

Lemma 1: When $|E_i| \geqslant rm - i$, for all $i \in \{1, \dots, rm-1\}$, we have:

$$\mathtt{rank}(M) \geqslant \sum_{i=1}^{rm-1} \mathtt{rank}(M_i).$$

Another observation is that M_i is equal to the sum of the submatrix $(p_{s,k+i}p_{s,j})_{\substack{s \in F_i \\ k+i<j\leqslant n}}$ of M and a certain matrix which is some function on the entries $p_{t,k+i}p_{t,j}$ where t belongs to $F_1 \cup \dots F_{i-1}$ and j ranges over $\{k+i+1, n\}$. Since by definition of F_i, $p_{s,k+i}$ is different from 0 for s in F_i. In addition, the rank of M_i does not change by multiplying each row of index s by $p_{s,k+i}^{-1}$. Then, it turns out that the rank of M_i is equal to the rank of a matrix which is the sum of the matrix $(p_{s,j})_{\substack{s \in F_i \\ k+i<j\leqslant n}}$, another matrix depending on the $p_{t,k+i}p_{t,j}$'s (where t ranges over $F_1 \cup \dots F_{i-1}$) and the $p_{s,k+1}$'s with $s \in F_i$. This proves that:

Lemma 2: Assume that $|E_i| \geqslant rm - i$ for all $i \in \{1, \dots, rm-1\}$. Then, the random variables $\mathtt{rank}(M_i)$ are independent and $\mathtt{rank}(M_i)$ is distributed as the rank of a square matrix of size $rm - i$ with entries drawn independently from the uniform distribution on $\mathbb{F}_q$.

Another essential ingredient for proving Theorem 2 is the following well known lemma (see for instance [40][Theorem 1])

Lemma 3: There exist two positive constants A and B depending on q such that the probability $p(s, \ell)$ that a random $\ell \times \ell$ matrix over $\mathbb{F}_q$ is of rank $\ell - s$ (where the coefficients are drawn independently from each other from the uniform distribution on $\mathbb{F}_q$) satisfies

$$\frac{A}{q^{s^2}} \leqslant p(s, \ell) \leqslant \frac{B}{q^{s^2}}.$$

This enables to control the exponential moments of the defect of a random matrix.For a square matrix M of size $\ell \times \ell$, we define the defect $d(M)$ by $d(M) \stackrel{\text{def}}{=} \ell - \mathtt{rank}(M)$.

Lemma 4: If M is random square matrix whose entries are drawn independently from the uniform distribution over $\mathbb{F}_q$, then there exists some constant K such that for every $\lambda > 0$,

$$\mathbb{E}\left(q^{\lambda d(\boldsymbol{M})}\right) \leqslant K q^{\frac{\lambda^2}{4}},$$

$\mathbb{E}(.)$ denoting the expectation.

Proof: By using Lemma 3, we obtain:

$$\mathbb{E}\left(q^{\lambda d(M)}\right) \leqslant \sum_{d=0}^{\infty} q^{\lambda d} \frac{B}{q^{d^2}} \leqslant B \sum_{d=0}^{\infty} q^{\lambda d - d^2}.$$

Observe that the maximum of the function $d \mapsto q^{\lambda d - d^2}$ is reached for $d_0 = \frac{\lambda}{2}$ and is equal to $q^{\frac{\lambda^2}{4}}$. Then, we can write the sum above as:

$$\sum_{d=0}^{\infty} q^{\lambda d - d^2} = \sum_{d \leqslant d_0} q^{\lambda d - d^2} + \sum_{d > d_0} q^{\lambda d - d^2}.$$

Finally, we notice that:

$$\frac{q^{\lambda(d+1)-(d+1)^2}}{q^{\lambda d - d^2}} \leqslant \frac{q^{\lambda(d_0+1)-(d_0+1)^2}}{q^{\lambda d_0 - d_0^2}} = \frac{1}{q} \text{ for } d > d_0,$$

$$\frac{q^{\lambda(d-1)-(d-1)^2}}{q^{\lambda d - d^2}} \leqslant \frac{q^{\lambda(d_0-1)-(d_0-1)^2}}{q^{\lambda d_0 - d_0^2}} = \frac{1}{q} \text{ for } d \leqslant d_0.$$

This leads to claim that $\sum_{d=0}^{\infty} q^{\lambda d - d^2} = O\left(q^{\frac{\lambda^2}{4}}\right)$ since we have:

$$\sum_{d=0}^{\infty} q^{\lambda d-d^2} \quad \leqslant \quad \sum_{d \leqslant d_0} q^{d-\lfloor d_0 \rfloor} q^{\frac{\lambda^2}{4}} + \sum_{d > d_0} q^{\lceil d_0 \rceil - d} q^{\frac{\lambda^2}{4}}.$$

■

We can use now the previous lemma together with Lemma 1 and Lemma 2 to derive the following lemma.

Lemma 5: Assuming that $|E_i| \geqslant rm - i$ for all $i \in \{1, \ldots, t\}$, we get:

$$\mathbf{Pr}\left(\sum_{i=1}^{t} d(M_i) \geqslant u\right) \leqslant K^t q^{-\frac{u^2}{t}}$$

where K is the constant appearing in Lemma 4.

Proof: Let $D \stackrel{\text{def}}{=} \sum_{i=1}^{t} d(\boldsymbol{M}_i)$. Using Markov's inequality:

$$\mathbf{Pr}(D \geqslant u) \leqslant \frac{\mathbb{E}(q^{\lambda D})}{q^{\lambda u}} \tag{20}$$

for some well chosen $\lambda > 0$. The exponential moment appearing at the numerator is upper-bounded with the help of the previous lemma and by using the independence of the random variables $q^{\lambda d(\boldsymbol{M}_i)}$ *i.e.*:

$$\begin{aligned} \mathbb{E}(q^{\lambda D}) \quad &= \quad \mathbb{E}\left(q^{\lambda \sum_{i=1}^{t} d(M_i)}\right) = \prod_{i=1}^{t} \mathbb{E}\left(q^{\lambda d(\boldsymbol{M}_i)}\right) \\ &\leqslant \quad K^t q^{\frac{t\lambda^2}{4}}. \end{aligned} \tag{21}$$

Using now (21) in (20), we obtain $\mathbf{Pr}(D \geqslant \alpha t) \leqslant K^t \frac{q^{\frac{t\lambda^2}{4}}}{q^{\lambda u}}$ which implies:

$$\mathbf{Pr}(D \geqslant \alpha t) \leqslant K^t q^{\frac{t\lambda^2}{4} - \lambda u}.$$

We choose $\lambda = \frac{2u}{t}$ to minimize this upper-bound leading to:

$$\mathbf{Pr}(D \geqslant u) \leqslant K^t q^{-\frac{u^2}{t}}.$$

■

The last ingredient for proving Theorem 2 is a bound on the probability that E_i is too small to construct F_i.

Lemma 6: Let $u_i \stackrel{\text{def}}{=} \binom{mr}{2} - \frac{1}{2}(2rm - i)(i-1)$ and F be the event "$|F_j| = rm - j$ for $j \in \{1, \ldots, i-1\}$" then

$$\mathbf{Pr}\left(|E_i| < rm - i \mid \ F\right) \leqslant e^{-\frac{2}{u_i}\left(\frac{q-1}{q} u_i - rm - i + 1\right)^2}.$$

Proof: When all the sets F_j are of size $rm - j$ for j in $\{1, \ldots, i-1\}$, it remains

$$N - \sum_{j=1}^{i-1}(rm - j) = N - \frac{1}{2}(2rm - i)(i-1) = u_i$$

rows which can be picked up for E_i. Let S_t be the sum of t Bernoulli variables of parameter $\frac{q-1}{q}$. We obviously have

$$\mathbf{Pr}(|E_i| < rm - i \mid \ F) = \mathbf{Pr}(S_{u_i} < rm - i).$$

It remains to use the Hoeffding inequality on the binomial tails to finish the proof. ■

We are ready now to prove Theorem 2:

TABLE I
VALUES OF $r_{\max}$ AND r_{CRIT} FOR A BINARY GOPPA CODE OF LENGTH $n = 2^m$.

m	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
$r_{\max}$	5	8	8	11	16	20	26	34	47	62	85	114	157	213	290	400
$\lceil r_{\text{crit}} \rceil$	5	6	8	11	14	19	25	34	46	62	84	114	156	214	293	402

Proof of Theorem 2: Let $u = \lceil \sqrt{mr\omega(mr))} \rceil$. We observe now that if all E_j's are of size at least $rm-j$ for $j \in \{1, \dots, u\}$, we can write that $D = N - \texttt{rank}(M)$:

$$
\begin{aligned}
D &\leqslant N - \sum_{i=1}^{rm-u} \texttt{rank}(M_i) \quad \text{(by Lemma 1)} \\
&= \sum_{i=1}^{rm-1} (rm-i) - \sum_{i=1}^{rm-u} \texttt{rank}(M_i) \\
&= \sum_{i=1}^{rm-u} d(M_i) + \sum_{i=rm-u+1}^{rm-1} (rm-i) \\
&= \sum_{i=1}^{rm-u} d(M_i) + \frac{u(u-1)}{2} \\
&< \sum_{i=1}^{rm-u} d(M_i) + \frac{1}{2} mr\omega(mr).
\end{aligned}
$$

From this we deduce that:

$$
\mathbf{Pr}\big(D_{\text{random}} \geqslant mr\omega(mr)\big) \leqslant \mathbf{Pr}(A \cup B) \leqslant \mathbf{Pr}(A) + \mathbf{Pr}(B)
$$

where A is the event "$\sum_{i=1}^{rm-u} d(M_i) \geqslant \frac{1}{2} mr\omega(mr)$" and B is the event "for at least one E_j with $j \in \{1, \dots, rm-u\}$ we have $|E_j| < rm - j$". We use now Lemma 5 to prove that $\mathbf{Pr}(A) = o(1)$ as rm goes to infinity. We finish the proof by noticing that the probability of the complementary set of B satisfies

$$
\begin{aligned}
\mathbf{Pr}(\bar{B}) &= \mathbf{Pr}\left(\bigcap_{i=1}^{rm-u} |E_i| \geqslant rm - i \right) \\
&= \prod_{i=1}^{rm-u} \mathbf{Pr}\left(|E_i| \geqslant rm - i \mid \ F \right) \\
&= 1 - o(1) \quad \text{(by Lemma 6)}.
\end{aligned}
$$

■

VII. CONCLUSION AND CRYPTOGRAPHIC IMPLICATIONS

The existence of a distinguisher for the specific case of binary Goppa codes is not valid for any value of r and m but tends to be true for codes that have a rate $\frac{n-mr}{n}$ very close to one. We will elaborate on this point below. This kind of codes are mainly encountered with the signature scheme [2]. If we assume that the length n is equal to 2^m and we denote by $r_{\max}$ the smallest integer r such that $N - T_{\text{Goppa}} \geqslant 2^m - mr$ then any binary Goppa code of degree $r < r_{\max}$ can be distinguished (Table I). For example, the binary Goppa code obtained with $m = 13$ and $r = 19$ corresponding to a 90-bit security McEliece public key is distinguishable. More interestingly, all the keys proposed in [18] for the CFS signature scheme can be distinguished.

A. Asymptotic Behaviour

When the length n of the code goes to infinity an asymptotic formula can be derived for the smallest rate R_{crit} allowing distinguish a random code from an alternant code or a Goppa code. We derive such a formula when we assume for simplicity that the cardinality q of the base field is fixed and n is chosen as $n = q^m$ (in practice n is chosen either in this way or at least of the same order as q^m). We also assume that the dimension k of the code satisfies $k = n - rm$. Finally, we also make the assumption that the dimensions $D_{\text{alternant}}$ and D_{Goppa} are given by their theoretical values $T_{\text{alternant}}$ and T_{Goppa} respectively and that the dimension of D_{random} is given by $T_{\text{random}} \stackrel{\text{def}}{=} \max\left(0, \binom{mr}{2} - k\right)$. This critical rate r_{crit} corresponds to the smallest value of r for which T_{random} becomes bigger than $T_{\text{alternant}}$ (asymptotically there will be no difference between Goppa codes or alternant codes). It holds that:

$$r_{\text{crit}} \stackrel{\text{def}}{=} \min\left\{ r > 0 \ \mid T_{\text{random}} \geqslant T_{\text{alternant}} \right\}.$$

We let $R_{\text{crit}} \stackrel{\text{def}}{=} \dfrac{n - r_{\text{crit}} m}{n} = 1 - \dfrac{r_{\text{crit}} m}{n}$. Our claim (whose proof is postponed in the appendix) is that

Theorem 3: Let us assume that $n = q^m$. When q is fixed and m tends to infinity then $r_{\text{crit}} = \sqrt{\frac{2\, q^m \log q}{m \log m}} \big(1 + o(1)\big)$ and

$$R_{\text{crit}} = 1 - \sqrt{\frac{2\, m\, \log q}{q^m \log m}} \Big(1 + o(1)\Big),$$

where all logarithms are taken to base 2.

In Table I, we have computed the value of $\left\lceil \sqrt{\frac{2q^m \log q}{m \log m}} \right\rceil$ for several m (q is equal to 2). This shows that our approximation is rather close to $r_{\max}$ computed in practice even for small values of m.

B. Concluding Remarks

We emphasize that the existence of such a distinguisher does not undermine the security of McEliece [3] or CFS [2]. It only shows that the GD assumption should be used with great care. It has also been observed in [41] that the value of D can be equivalently determined by considering the dimension of the square code of the dual of the public code. The square code construction relies on the component-wise product of vectors. For any vectors $\boldsymbol{a} = (\boldsymbol{a}_1, \dots, \boldsymbol{a}_n)$ and $\boldsymbol{b} = (\boldsymbol{b}_1, \dots, \boldsymbol{b}_n)$ we denote it by $\boldsymbol{a} \star \boldsymbol{b} \stackrel{\text{def}}{=} (\boldsymbol{a}_1\boldsymbol{b}_1, \dots, \boldsymbol{a}_n\boldsymbol{b}_n)$. For a code $\mathscr{A}$, we denote by $\mathscr{A}^2$ its square code which is the linear space spanned by $\boldsymbol{a}_i \star \boldsymbol{a}_j$ where $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$ describe a basis of $\mathscr{A}$. If we denote by $\mathscr{D}$ the dual of the public code then it turns out [41] that $\dim(\mathscr{D}^2) = \binom{\dim(\mathscr{D})+1}{2} - D$. It should be added that this notion has been used recently to successfully attack several cryptographic schemes relying on (modified) generalized Reed-Solomon codes [42]. More generally, a natural open question is to investigate the hardness of GD for others codes having a polynomial-time decoding algorithm (for instance, LDPC, Reed-Muller,…).

Lastly, the recent work [30] shows that the natural reduction of GD to a hidden subgroup problem yields negligible information. As a consequence, it rules out the direct analogue of a quantum attack using the so-called Quantum Fourier Sampling (QFS) which breaks number theoretic problems [31]. More exactly, [30] shows that QFS has a negligible advantage against GD when the rate is $\geqslant R_{\text{QFS}}$ where $R_{\text{QFS}} \stackrel{\text{def}}{=} 1 - \frac{\log_q(n)^{3/2}}{\sqrt{5n}} = 1 - \frac{m^{3/2}}{\sqrt{5 \cdot q^{m/2}}}$. Whilst our result is somewhat contradictory with [30], it is interesting to observe that R_{crit} and the critical rate R_{QFS} share some similarities.

REFERENCES

[1] J. Faugère, V. Gauthier-Umana, A. Otmani, L. Perret, and J. Tillich, "Distinguisher for high rate McEliece cryptosystems," in *Proceedings of the 2011 IEEE Information Theory Workshop (ITW 2011)*, Paraty, Brazil, Oct. 16-20 2011.

[2] N. T. Courtois, M. Finiasz, and N. Sendrier, "How to achieve a McEliece-based digital signature scheme," in *ASIACRYPT*, vol. 2248, 2001, pp. 157–174.

[3] R. J. McEliece, *A Public-Key System Based on Algebraic Coding Theory*. Jet Propulsion Lab, 1978, pp. 114–116, dSN Progress Report 44.

[4] P. J. Lee and E. F. Brickell, "An observation on the security of McEliece's public-key cryptosystem," in *Advances in Cryptology - EUROCRYPT'88*, ser. Lecture Notes in Computer Science, vol. 330/1988. Springer, 1988, pp. 275–280.

[5] J. S. Leon, "A probabilistic algorithm for computing minimum weights of large error-correcting codes," *IEEE Transactions on Information Theory*, vol. 34, no. 5, pp. 1354–1359, 1988.

[6] J. Stern, "A method for finding codewords of small weight," in *Coding Theory and Applications*, ser. Lecture Notes in Computer Science, G. D. Cohen and J. Wolfmann, Eds., vol. 388. Springer, 1988, pp. 106–113.

[7] 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*, vol. 44, no. 1, pp. 367–378, 1998.

[8] D. J. Bernstein, T. Lange, and C. Peters, "Attacking and defending the McEliece cryptosystem," in *PQCrypto*, ser. LNCS, vol. 5299, 2008, pp. 31–46.

[9] ——, "Smaller decoding exponents: Ball-collision decoding," in *CRYPTO*, ser. Lecture Notes in Computer Science, P. Rogaway, Ed., vol. 6841. Springer, 2011, pp. 743–760.

[10] A. May, A. Meurer, and E. Thomae, "Decoding random linear codes in $\tilde{O}(2^{0.054n})$," in *ASIACRYPT*, ser. Lecture Notes in Computer Science, D. H. Lee and X. Wang, Eds., vol. 7073. Springer, 2011, pp. 107–124.

[11] A. Becker, A. Joux, A. May, and A. Meurer, "Decoding random binary linear codes in $2^{n/20}$: How 1 + 1 = 0 improves information set decoding," in *EUROCRYPT*, ser. Lecture Notes in Computer Science, D. Pointcheval and T. Johansson, Eds., vol. 7237. Springer, 2012, pp. 520–536.

[12] E. Berlekamp, R. McEliece, and H. van Tilborg, "On the inherent intractability of certain coding problems," *IEEE Transactions on Information Theory*, vol. 24, no. 3, pp. 384–386, May 1978.

[13] J. Gibson, "Equivalent Goppa codes and trapdoors to McElieces public key cryptosystem," in *Advances in Cryptology EUROCRYPT 91*, ser. Lecture Notes in Computer Science, D. Davies, Ed. Springer Berlin / Heidelberg, 1991, vol. 547, pp. 517–521.

[14] P. Loidreau and N. Sendrier, "Weak keys in the McEliece public-key cryptosystem," *IEEE Transactions on Information Theory*, vol. 47, no. 3, pp. 1207–1211, 2001.

[15] R. Nojima, H. Imai, K. Kobara, and K. Morozov, "Semantic security for the McEliece cryptosystem without random oracles," *Des. Codes Cryptography*, vol. 49, no. 1-3, pp. 289–305, 2008.

[16] R. Dowsley, J. Müller-Quade, and A. C. A. Nascimento, "A CCA2 secure public key encryption scheme based on the McEliece assumptions in the standard model," in *CT-RSA*, 2009, pp. 240–251.

[17] L. Dallot, "Towards a concrete security proof of Courtois, Finiasz and Sendrier signature scheme," in *WEWoRC*, 2007, pp. 65–77.

[18] M. Finiasz and N. Sendrier, "Security bounds for the design of code-based cryptosystems," in *Asiacrypt 2009*, ser. LNCS, M. Matsui, Ed., vol. 5912. Springer, 2009, pp. 88–105.

[19] J.-C. Faugère, A. Otmani, L. Perret, and J.-P. Tillich, "Algebraic cryptanalysis of McEliece variants with compact keys," in *EUROCRYPT*, ser. Lecture Notes in Computer Science, H. Gilbert, Ed., vol. 6110. Springer, 2010, pp. 279–298.

[20] V. D. Goppa, "A new class of linear correcting codes," *Probl. Peredachi Inf.*, vol. 6, no. 3, 1970.

[21] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, 5th ed. Amsterdam: North–Holland, 1986.

[22] N. Patterson, "The algebraic decoding of Goppa codes," *IEEE Transactions on Information Theory*, vol. 21, no. 2, pp. 203–207, 1975.

[23] J. van Tilburg, "On the McEliece public-key cryptosystem," in *CRYPTO '88: Proceedings of the 8th Annual International Cryptology Conference on Advances in Cryptology*. London, UK: Springer-Verlag, 1990, pp. 119–131.

[24] A. Canteaut and H. Chabanne, "A further improvement of the work factor in an attempt at breaking McEliece's cryptosystem," in *EUROCODE 94*. INRIA, 1994, pp. 169–173.

[25] A. Canteaut and F. Chabaud, "Improvements of the attacks on cryptosystems based on error-correcting codes," INRIA, Tech. Rep. 95–21, 1995.

[26] I. Dumer, "Suboptimal decoding of linear codes : partition techniques," *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1971–1986, 1996.

[27] A. Canteaut and N. Sendrier, "Cryptanalysis of the original McEliece cryptosystem," in *Advances in Cryptology - ASIACRYPT'98*, ser. Lecture Notes in Computer Sscience, no. 1514. Springer-Verlag, 1998, pp. 187–199.

[28] N. Sendrier, "On the use of structured codes in code based cryptography," in *Coding Theory and Cryptography III*, ser. Contactforum, S. Nikova, B. Preneel, and L. Storme, Eds. Koninklijke Vlaamse Academie van België voor Wetenschaeppen en Kunsten, 2009, pp. 59–68.

[29] ——, "Finding the permutation between equivalent linear codes: The support splitting algorithm," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1193–1203, 2000.

[30] H. Dinh, C. Moore, and A. Russell, "McEliece and Niederreiter cryptosystems that resist quantum Fourier sampling attacks," in *CRYPTO*, ser. Lecture Notes in Computer Science, P. Rogaway, Ed., vol. 6841. Springer, 2011, pp. 761–779.

[31] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM J. Comput.*, vol. 26, no. 5, pp. 1484–1509, 1997.

[32] K. Kobara and H. Imai, "Semantically secure McEliece public-key cryptosystems-conversions for McEliece PKC," in *Public Key Cryptography, 4th International Workshop on Practice and Theory in Public Key Cryptography, PKC 2001, Cheju Island, Korea, February 13-15, 2001, Proceedings*, ser. Lecture Notes in Computer Science, K. Kim, Ed., vol. 1992. Springer, 2001, pp. 19–35.

[33] R. Nojima, H. Imai, K. Kobara, and K. Morozov, "Semantic security for the McEliece cryptosystem without random oracles," *Des. Codes Cryptography*, vol. 49, no. 1-3, pp. 289–305, 2008.

[34] R. Dowsley, J. Müller-Quade, and A. C. A. Nascimento, "A CCA2 secure public key encryption scheme based on the McEliece assumptions in the standard model," in *CT-RSA*, 2009, pp. 240–251.

[35] T. P. Berger, P. Cayrel, P. Gaborit, and A. Otmani, "Reducing key length of the McEliece cryptosystem," in *Progress in Cryptology - Second International Conference on Cryptology in Africa (AFRICACRYPT 2009)*, ser. Lecture Notes in Computer Science, B. Preneel, Ed., vol. 5580, Gammarth, Tunisia, Jun. 21-25 2009, pp. 77–97.

[36] R. Misoczki and P. S. L. M. Barreto, "Compact McEliece keys from Goppa codes," in *Selected Areas in Cryptography (SAC 2009)*, Calgary, Canada, Aug. 13-14 2009.

[37] J.-C. Faugère, "A new efficient algorithm for computing gröbner bases (f4)," *Journal of Pure and Applied Algebra*, vol. 139(1-3), pp. 61–88, 1999.

[38] ——, "A new efficient algorithm for computing gröbner bases without reduction to zero : F5," in *ISSAC'02*. ACM press, 2002, pp. 75–83.

[39] W. Bosma, J. J. Cannon, and C. Playoust, "The Magma algebra system I: The user language," *J. Symb. Comput.*, vol. 24, no. 3/4, pp. 235–265, 1997.

[40] C. Cooper, "On the distribution of rank of a random matrix over a finite field," *Random Struct. Algorithms*, vol. 17, no. 3-4, pp. 197–212, 2000.

[41] I. Marquez-Corbella and R. Pellikaan, "Error-correcting pairs for a public-key cryptosystem," in *CBC 2012, Code-based Cryptography Workshop 2012*, 2012, paper available on `http://www.win.tue.nl/ ruudp/paper/60.pdf` .

[42] A. Couvreur, P. Gaborit, V. Gauthier, A. Otmani, and J. Tillich, "Distinguisher-based attacks on public-key cryptosystems using reed-solomon codes," in *Proceedings of the 2013 International Workshop on Coding and Cryptography (WCC 2013)*, Bergen, Norway, 2013, pp. 181–193.

[43] P. Rogaway, Ed., *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, ser. Lecture Notes in Computer Science, vol. 6841. Springer, 2011.

Appendix

A. Experimental Results

We gathered in Table II-XI some results obtained through intensive computations with the Magma system [39]. 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 $\{3,\dots,50\}$ and several m. The Goppa codes are generated by means of an irreducible $\gamma(z)$ of degree r and hence $\gamma(z)$ has no multiple roots. In particular, we can apply Theorem 1 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 II and Table III give figures for the binary case with $m = 14$. We can check that D_{random} is equal to 0 for $r \in \{3,\dots,12\}$ and $D_{\text{random}} = N - k$ as expected. We remark that $D_{\text{alternant}}$ is different from D_{random} whenever $r \leqslant 15$, and D_{Goppa} is different from D_{random} as long as $r \leqslant 25$. Finally we observe that our formulas for $T_{\mathsf{alternant}}$ fit as long as $k \geqslant N - D_{\text{alternant}}$ which correspond to $r \leqslant 15$. This is also the case for binary Goppa codes since we have $T_{\mathsf{Goppa}} = D_{\text{Goppa}}$ as long as $k \geqslant N - D_{\text{Goppa}}$ *i.e.,* $r \leqslant 25$. We also give in Table X and Table XI the examples 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 IV-VI and $m = 16$ in Table VII-IX. We see that D_{random} and D_{Goppa} are different for $r \leqslant 33$ when $m = 15$ and for $m = 16$ they are different even beyond our range of experiment ($r \leqslant 50$).

TABLE II
$q = 2$ 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
$T_{\text{alternant}}$	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

TABLE III
$q = 2$ 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
$T_{\text{alternant}}$	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

TABLE IV
$q = 2$ 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

TABLE V
$q = 2$ 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

TABLE VI
$q = 2$ 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

B. Proof of Theorem 3

To prove Theorem 3 we will first use the following observation

TABLE VII
$q = 2$ 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

TABLE VIII
$q = 2$ 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

TABLE IX
$q = 2$ 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

TABLE X
$q = 4$ 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
$T_{\mathsf{alternant}}$	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

TABLE XI
$q = 4$ 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
$T_{\mathsf{alternant}}$	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

Lemma 7: Let $T_{\text{alternant}}$ be as defined in (4). Let also T_{Goppa} be as defined in (6). There exists constants K_1 and K_2 (resp. K_1' and K_1' and K_2') such that:

$$mr^2\big(\log_q(r) + K_1\big) \leqslant T_{\text{alternant}} \leqslant mr^2\big(\log_q(r) + K_2\big), \tag{22}$$

$$mr^2(\log_q(r) + K_1') \leqslant T_{\text{Goppa}} \leqslant mr^2(\log_q(r) + K_2'). \tag{23}$$

Proof: We recall that:

$$T_{\text{alternant}} \stackrel{\text{def}}{=} \frac{1}{2}m(r-1)\left((2e+1)r - 2\frac{q^{e+1}-1}{q-1}\right),$$

where $e \stackrel{\text{def}}{=} \lfloor \log_q(r-1) \rfloor$. There exists some absolute constants K_3 and K_4 such that for all integers $r \geqslant 2$:

$$\begin{cases} 2r\log_q(r) + K_3 r & \leqslant & (2e+1)r - 2\dfrac{q^{e+1}-1}{q-1} \\ 2r\log_q(r) + K_4 r & \geqslant & (2e+1)r - 2\dfrac{q^{e+1}-1}{q-1} \end{cases} \tag{24}$$

The upper bound is clear since:

$$(2e+1)r - 2\frac{q^{e+1}-1}{q-1} \leqslant 2r\log_q(r) + 2r.$$

For the lower bound, we remark that:

$$e \geqslant \log_q(r-1) - 1 = \log_q(r) + \log_q(1-1/r) - 1.$$

In addition:

$$\frac{q^{e+1}-1}{q-1} \leqslant q \cdot q^e \leqslant rq.$$

As a consequence $\left((2e+1)r - 2\frac{q^{e+1}-1}{q-1}\right) \geqslant 2r\log_q(r) + r\big(2\log_q(1-1/r) - 1 - 2q\big)$. Remark that $\log_q(1-1/r)$ can be bounded from above by some (negative) constant. So, it holds that:

$$\left((2e+1)r - 2\frac{q^{e+1}-1}{q-1}\right) \geqslant 2r\log_q(r) + K_3 r,$$

for some constant K_3. Observe now that $\frac{1}{2}m(r-1)\left(2\log_q(r) + K_3 r\right) = \frac{1}{2}(mr-m)\left(2\log_q(r) + K_3 r\right)$ is equal to:

$$\frac{1}{2}\left(2mr\log_q(r) + K_3 mr^2 - 2m\log_q(r) - K_3 mr\right). \tag{25}$$

The lower bound on $T_{\text{alternant}}$ follows immediately from this. The expression can be lower bounded (resp. upper bounded) by a term of the form $K_1 mr^2$ (resp. $K_2 mr^2$) for some constant K_1 (resp. K_2). This holds for all positive integers r. Finally, we recall that when $r < q-1$:

$$T_{\text{Goppa}} = T_{\text{alternant}} = \frac{1}{2}m(r-1)(r-2)$$

and when $r \geqslant q-1$:

$$T_{\text{Goppa}} = \frac{1}{2}mr\Big((2e+1)r - 2(q-1)q^{e-1} - 1\Big).$$

with e being the unique integer such that:

$$(q-1)^2 q^{e-2} < r \leqslant (q-1)^2 q^{e-1}.$$

The bound (23) on can be proved in the same way. ■

From this lemma, we deduce that:

Lemma 8: There exist two constants C_1 and C_2 such that for every r satisfying $\binom{mr}{2} \geqslant n - mr$ we have:

$$mr^2\left(m/2 - \log_q(r) + C_1\right) - q^m \leqslant T_{\text{random}} - T_{\text{alternant}}, \tag{26}$$

$$mr^2\left(m/2 - \log_q(r) + C_2\right) - q^m \geqslant T_{\text{random}} - T_{\text{alternant}}. \tag{27}$$

We also have the same inequalities when we replace $T_{\text{random}} - T_{\text{alternant}}$ with $T_{\text{random}} - T_{\text{Goppa}}$.

Proof: For all positive integer values of r such that $\binom{mr}{2} \geqslant n - mr$, we have:

$$\begin{aligned} T_{\text{random}} &= N - k = \binom{mr}{2} - q^m + mr \\ &= m^2r^2/2 - q^m + mr/2. \end{aligned}$$

We can then conclude using Lemma 7. ■

From this lemma, we derive the following estimate for r_{crit}:

Lemma 9: When m goes to infinity we have

$$r_{\text{crit}} = \sqrt{\frac{2q^m \log q}{m\log m}}\big(1 + o(1)\big).$$

Proof: From Lemma 8, we know that:

$$T_{\text{random}} - T_{\text{alternant}} = mr^2 \left(m/2 - \log_q(r)\right) - q^m + O(mr^2).$$

Let $r_0 \stackrel{\text{def}}{=} \sqrt{\frac{2q^m \log q}{m \log m}}$. It holds that:

$$\begin{aligned} 2\log_q(r_0) &= \log_q(2q^m \log q) - \log_q(m \log m), \\ &= m + \log_q(2\log q) - \log_q(m) - \log_q(\log m). \end{aligned}$$

Thus by writing that $mr_0^2 \left(m/2 - \log_q r_0\right) - q^m$ is equal to:

$$\frac{2q^m \log q}{\log m}\Big(m/2 - m/2 \\ - \frac{\log(2\log q)}{2\log q} + \frac{\log m}{2\log q} + \frac{\log\log m}{2\log q}\Big) - q^m$$

which in turn equals $\frac{q^m}{\log m}\left(\log\log m - \log(2\log q)\right)$. We also observe that $mr_0^2 = \frac{2q^m \log q}{\log m}$ is negligible compared to $\frac{q^m}{\log m}\left(\log\log m - \log(\right.$ when m goes to infinity. This can be used to show that $T_{\text{random}} - T_{\text{alternant}}$ is positive for $r = \lceil r_0 \rceil$ when m is large enough. Therefore for m large enough, we have $r_{\text{crit}} \leqslant \lceil r_0 \rceil$. On the other hand, let α be any positive constant < 1. We set:

$$r_\alpha \stackrel{\text{def}}{=} \sqrt{\frac{2\alpha q^m \log q}{m \log m}}.$$

Notice that the function $f(x) = mx^2 \left(m/2 - \log_q x\right) - q^m$ can be shown to be increasing in the range $(0, r_\alpha)$. Therefore for every $r \leqslant r_\alpha$, we have $mr^2 \left(m/2 - \log_q r\right) - q^m$ is less than or equal to:

$$-q^m + \frac{2\alpha q^m \log q}{\log m}\Big(m/2 - m/2 + \frac{\log m}{2\log q} \\ + \frac{\log\log m}{2\log q} - \frac{\log(2\alpha\log q)}{2\log q}\Big),$$

This last quantity is upper-bounded by:

$$(\alpha - 1)q^m + \frac{2\alpha q^m \log q}{\log m}\left(\frac{\log\log m}{2\log q} - \frac{\log(2\alpha\log q)}{2\log q}\right)$$

which in turn is less than or equal to:

$$(\alpha - 1)q^m + q^m \frac{\alpha \log\log m}{\log m}.$$

Since $mr^2 \leqslant \frac{2\alpha q^m \log q}{\log m}$, it follows that any function of the form $mr^2 \left(m/2 - \log_q r\right) - q^m + O(mr^2)$ will be negative for m large enough in the range $(0, r_\alpha)$. This implies that $r_{\text{crit}} \geqslant r_\alpha = \sqrt{\frac{2\alpha q^m \log q}{m \log m}}$ for m large enough. We deduce from this fact which holds for any $0 < \alpha < 1$ and from the upper bound $r_{\text{crit}} \leqslant \lceil r_0 \rceil = \left\lceil \sqrt{\frac{2q^m \log q}{m \log m}} \right\rceil$ that $r_{\text{crit}} = \sqrt{\frac{2q^m \log q}{m \log m}}(1 + o(1))$ when m goes to infinity. Finally, the proof of Theorem 3 is now obtained by remarking:

$$R_{\text{crit}} = \frac{q^m - mr_{\text{crit}}}{q^m} = 1 - \sqrt{\frac{2\, m\, \log q}{q^m \log m}}\left(1 + o(1)\right).$$

■

Jean-Charles Faugère Jean-Charles Faugère graduated from École Normale Supérieure, Paris, France, and received a Ph.D. degree from the University of Paris 6 in 1994. He was a CNRS Researcher until 2006; presently, he is a INRIA Senior Researcher and the head of the project-team PolSys which is common between INRIA (Paris-Rocquencourt), UPMC (LIP6) and CNRS (LIP6). His research interests are in computer algebra, polynomial systems solving, Gröbner basis, cryptography and other applications of Computer Algebra.

Valérie Gauthier-Umaña was born in Lyon, France. She received a M.Sc. degree (ALGANT program) in mathematics from the University of Bordeaux I, France and the Univertià degli study di Padova, Padova, Italy in 2008, and a Ph.D. degree in mathematics from the Technical University of Denmark, Lyngby, Denmark in 2011. In 2012 she was a postdoctoral Researcher at the University of Caen, France. She is currently a postdoctoral Researcher at the Universidad de los Andes, Bogotá, Colombia. Her research lies in cryptography and coding theory.

Ayoub Otmani holds a Ph.D. degree in mathematics and its applications from the University of Limoges (France) since 2002. From 2004 through 2012 he was associate professor at Ensicaen (Caen, France). From 2009 to 2011, he was in secondment as a Researcher at the Institut de Recherche en Informatique et Automatique (INRIA - Rocquencourt) He is currently Professor at University of Rouen, France. His research lies in the fields of algebraic coding theory and cryptography.

Ludovic Perret holds a Ph.D. degree in computer science from the University of Marne-la-Vallée (France) since 2005. He is associate professor since 2007 at UPMC (Paris, France) in the project-team PolSys which is common between UPMC/LIP6 and INRIA-Rocquencourt. His research area is at the intersection of cryptography and symbolic computation.

Jean-Pierre Tillich (M'06) received the Engineer degree from École des Mines de Paris, Paris, France, in 1989 and the Ph.D. degree in computer science from École Nationale Suprieure des Télćommunications (ENST), Paris, in 1994. From 1997 to 2003, he was an Assistant Professor at the University Paris XI. He is now a Researcher at the Institut de Recherche en Informatique et Automatique (INRIA), Rocquencourt, Le Chesnay, France. His research interests include classical and quantum coding theory, cryptography and graph theory.