Polynomial Time Attack on Wild McEliece Over Quadratic Extensions

Alain Couvreur, Ayoub Otmani, Jean-Pierre Tillich

Polynomial Time Attack on Wild McEliece Over Quadratic Extensions

Alain Couvreur, Ayoub Otmani, Jean-Pierre Tillich

***Abstract*—We present a polynomial-time structural attack against the McEliece system based on Wild Goppa codes defined over a quadratic finite field extension. We show that such codes can be efficiently distinguished from random codes. The attack uses this property to compute a filtration, that is to say, a family of nested subcodes which will reveal their secret algebraic description.**

***Index Terms*—McEliece cryptosystem, Wild Goppa code, cryptanalysis, Goppa Code Distinguishing problem.**

I. Introduction

THE McEliece encryption scheme [1] which dates back to the end of the seventies still belongs to the very few public key cryptosystems which remain unbroken. It is a code-based public-key cryptosystem that relies on binary Goppa codes. Several proposals which suggested to replace these codes with alternative families did not meet a similar fate. They all focus on a specific class of codes equipped with a decoding algorithm: generalized Reed–Solomon codes (GRS for short) [2] or large subcodes of them [3], Reed–Muller codes [4], algebraic geometry codes [5], LDPC and MDPC codes [6], [7] or convolutional codes [8], [9]. Most of them were successfully cryptanalyzed [10], [11], [12], [13], [14], [15], [16], [17], [18]. Each time a description of the underlying code suitable for decoding is efficiently obtained. But some of them remain unbroken, namely those relying on MDPC codes [7] and their cousins [6], the original binary Goppa codes of [1].

Slight variations of the binary Goppa codes family have also been proposed in order to reduce the key size, by replacing the binary Goppa codes by non binary Goppa codes [19], [20], or by considering quasi-cyclic, quasi-dyadic or quasi-monoidic versions of Goppa codes [21], [22], [23] (or more generally of alternant codes in [21]). The idea is here that what makes Goppa codes and the more general family of alternant codes suitable in the McEliece scheme, is that they display many properties of random codes. For instance, their weight distribution is close to the weight distribution of a random code [24]. However, in the quasi-cyclic/quasi-dyadic case it was shown in [25], [26], [27], [28] that the added structure allows a drastic reduction of the number of unknowns in algebraic attacks and most of the schemes proposed in [21], [22], [23] were broken by this approach. This kind of attack has exponential complexity and it can be thwarted by choosing smaller cyclic or dyadic blocks in this approach in order to increase the number of unknowns of the algebraic system. It is infeasible for the original McEliece scheme (the number of unknowns is linear in the length of the code) or the non-binary Goppa codes proposed in [29].

a) **Distinguisher for Goppa and Reed-Solomon codes:** None of the existing strategies is able to severely dent the security of [1] when appropriate parameters are taken. Consequently, it has even been advocated that the generator matrix of a Goppa code does not disclose any visible structure that an attacker could exploit. This is strengthened by the aforementioned fact that Goppa codes share many characteristics with random codes. Despite this fact, in [30], [31], an algorithm that manages to distinguish between a random code and a high rate Goppa code has been introduced.

b) **Component wise products of codes:** [32] showed that the distinguisher given in [30] has an equivalent but simpler description in terms of component-wise product of codes. This product allows in particular to define the square of a code. This square code operation can be used to distinguish a high rate Goppa code from a random one because the dimension of the square of the dual is much smaller than the one obtained with a random code. The notion of component-wise product of codes was first put forward to unify many different algebraic decoding algorithms [33], [34], then exploited in cryptology in [11] to break a McEliece variant based on random subcodes of GRS codes [3] and in [35], [36], [17], [37] to study the security of encryption schemes using algebraic geometry codes. Component-wise powers of codes are also studied in the context of secret sharing and secure multi-party computation [38], [39].

c) **Filtration key-recovery attacks:** The works [30], [31], without undermining the security of [1], prompts to wonder whether it would be possible to devise an attack exploiting the distinguisher. That was indeed the case in [15] for McEliece-like public-key encryption schemes relying on modified GRS codes [40], [41], [42]. Additionally, [15] has shown that the unusually low dimension of the square code of a generalized GRS code enables to compute a *filtration*, that is a nested sequence of subcodes, allowing the recovery of its algebraic structure. This gives an attack that is radically different from the Sidelnikov-Shestakov approach [10]. Notice

A. Couvreur is with Inria & LIX, CNRS UMR 7161 — École Polytechnique, 91128 Palaiseau Cedex, France. e-mail: alain.couvreur@lix.polytechnique.fr.

A. Otmani is with Normandie Univ, France; UR, LITIS, F-76821 Mont-Saint-Aignan, France. email: ayoub.otmani@univ-rouen.fr

J.P. Tillich is with Inria, 2 rue Simone Iff Paris 75012, France. email: jean-pierre.tillich@inria.fr.

Part of this work was supported by the Commission of the European Communities through the Horizon 2020 program under project number 645622 PQCRYPTO.

that the first step of the Sidelnikov-Shestakov attack which consists in computing the minimal codewords and then using this information for recovering the algebraic structure has been fruitful for breaking other families of codes: for instance binary Reed-Muller codes [12] or low-genus algebraic geometry codes [13]. This is not the approach we have followed here, because finding such codewords seems out of reach for the codes we are interested in, namely Goppa codes. Our filtration attack is really a new paradigm for breaking public key cryptosystems based on algebraic codes which in particular avoids the possibly very expensive computation of minimum weight codewords.

d) **Our contribution:** The purpose of this article is to show that the filtration attack of [15] which gave a new way of attacking a McEliece scheme based on GRS codes can be generalized to other families of codes. Notice that this filtration approach was also followed later on with great success to break all schemes based on algebraic geometry codes [17], whereas the aforementioned attack of Faure and Minder [13] could handle only the case of very low genus curves, due precisely to the expensive computation of the minimal codewords. A tantalizing project would be to attack Goppa code based McEliece schemes, or more generally alternant code based schemes. The latter family of codes are subfield subcodes defined over some field $\mathbb{F}_q$ of GRS codes defined over a field extension $\mathbb{F}_{q^m}$. Even for the smallest possible field extension, that is for $m = 2$, the cryptanalysis of alternant codes is a completely open question. Codes of this kind have indeed been proposed as possible improvements of the original McEliece scheme, under the form of *wild Goppa codes* in [19]. These are Goppa codes associated to polynomials of the form γ^{q-1} where γ is irreducible. Notice that all irreducible binary Goppa codes of the original McEliece system are actually wild Goppa codes. Interestingly enough, it turns out that these wild Goppa codes for $m = 2$ can be distinguished from random codes for a very large range of parameters by observing that the square code of some of their shortenings have a small dimension compared to squares of random codes of the same dimension. It should be pointed out that in the propositions of [19], the case $m = 2$ was particularly attractive since it provided the smallest key sizes.

We show here that this distinguishing property can be used to compute an interesting filtration of the public code, that is to say a family of nested subcodes of the public Goppa code such that each element of the family is an alternant code with the same support. This filtration can in turn be used to recover the algebraic description of the Goppa code as an alternant code, which yields an efficient key recovery attack. The attack is summarized in the heuristic below stated in Section VII. We say *heuristic* instead of *theorem* since a minor part of the tools used in the attack is not rigorously proved but are justified by heuristic arguments and confirmed by experiments.

Heuristic 1 *Let $\gamma \in \mathbb{F}_{q^2}[x]$ be an irreducible polynomial of degree r and $\boldsymbol{x}$ be an n–tuple of distinct elements of $\mathbb{F}_{q^2}$. Let $\mathscr{C}$ be the Goppa code $\mathscr{G}\left(\boldsymbol{x}, \gamma^{q-1}\right)$ used as a public key for the McEliece encryption scheme, then if*

$$n > 2q+4 \quad \text{and} \quad \binom{r(r+2)+2}{2} > 2r(q+1)+(q-r)-2,$$

there is a deterministic key-recovery attack of the scheme in $O(n^5)$ operations and a probabilistic one in $O(n^4)$.

This attack has been implemented in Magma [43] and allowed to break completely all the schemes with a claimed 128 bit security in Table 7.1 of [19] when $m = 2$ and the degree of γ is larger than 3. This corresponds precisely to the case where these codes can be distinguished from random codes by square code considerations. The filtration attack has a polynomial time complexity and basically boils down to linear algebra. This is the first time in the almost 40 years of existence of the MceEliece scheme that a polynomial time attack has been found on (non binary) Goppa codes. It questions the common belief that when Goppa codes are not GRS codes, i.e. when $m \geqslant 2$, they are immune to algebraic attacks on the key and only generic information-set-decoding attacks apply. It also raises the issue whether this algebraic distinguisher of Goppa and more generally alternant codes (see [31]) based on square code considerations can be turned into an attack in the other cases where it applies (for instance for Goppa codes of rate close enough to 1). Finally, it is worth pointing out that our attack works against codes without external symmetries confirming that the mere appearance of randomness is far from being enough to defend codes against algebraic attacks.

It should also be pointed out that subsequently to this work, it has been shown in [44] that one of the parameters of [19] that we have broken here can be attacked by Gröbner basis techniques by introducing an improvement of the algebraic modeling of [25] together with a new way of exploiting non-prime fields $\mathbb{F}_q$. Unlike our attack, it also applies to field extensions that are larger than 2 and to variations of the wild Goppa codes, called "wild Goppa incognito" in [20]. However this new attack is exponential in nature and can be thwarted by using either more conservative parameters or prime fields $\mathbb{F}_q$. Our attack on the other hand is much harder to avoid due to its polynomial complexity and choosing $m = 2$ together with wild Goppa codes seems now something that has to be considered with great care in a McEliece cryptosystem after our work.

e) **Outline of the article:** Our objective is to provide a self-contained article which could be read by cryptographers who are not aware with coding theory. For this reason, notation and many classical prerequisites are given in Section II. The core of our attack is presented in Sections III, IV. Section III presents a distinguisher on the public key i.e. a manner to distinguish such codes from random ones and Section IV uses this distinguisher to compute a family of nested subcodes of the public key providing information on the secret key. Section V is devoted to a short overview of the last part of the attack. Further technical details on the attack are given in Appendix E. The complexity of the attack is discussed in Section VI. Most of the attack is justified by mathematical proofs but a few facts which are subject to heuristic results

and have been experimentally verified. In Section VII are stated heuristics summarizing the reach of the attack and the unproved facts are listed. Section VI discusses the theoretical complexity of our algorithm and presents several running times of our Magma implementation of the attack.

f) **Note:** The material of this article was presented at the conference *EUROCRYPT 2014* (Copenhagen, Denmark) and published in its proceedings [45]. Due to space constraints, most of the proofs were omitted in the proceedings version. The present article is a long revisited version including all the missing proofs. Any proof which we did not consider as fundamental has been sent to the appendices. We encourage the reader first to read the article without these proofs and then read the appendices.

II. Notation, Definitions and Prerequisites

We introduce in this section notation we will use in the sequel. We assume that the reader is familiar with notions from coding theory. We refer to [46] for the terminology.

A. Vectors, matrices and Schur product

Vectors and matrices are respectively denoted in bold letters and bold capital letters such as $\boldsymbol{a}$ and $\boldsymbol{A}$. We always denote the entries of a vector $\boldsymbol{u} \in \mathbb{F}_q^n$ by $u_0, \ldots, u_{n-1}$. Given a subset $\mathcal{I} \subset \{0, \ldots, n-1\}$, we denote by $\boldsymbol{u}_{\mathcal{I}}$ the vector $\boldsymbol{u}$ *punctured* at $\mathcal{I}$, that is to say, *every entry with index in $\mathcal{I}$ is removed*. When $\mathcal{I} = \{j\}$ we allow ourselves to write $\boldsymbol{u}_j$ instead of $\boldsymbol{u}_{\{j\}}$. The component-wise product also called the Schur product $\boldsymbol{u} \star \boldsymbol{v}$ of two vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{F}_q^n$ is defined as:

$$\boldsymbol{u} \star \boldsymbol{v} \stackrel{\text{def}}{=} (u_0 v_0, \ldots, u_{n-1} v_{n-1}).$$

The i–th power $\boldsymbol{u} \star \cdots \star \boldsymbol{u}$ is denoted by $\boldsymbol{u}^i$. When every entry u_i of $\boldsymbol{u}$ is nonzero, we set

$$\boldsymbol{u}^{-1} \stackrel{\text{def}}{=} (u_0^{-1}, \ldots, u_{n-1}^{-1}),$$

and more generally for all i, we define $\boldsymbol{u}^{-i}$ in the same manner. The operation $\star$ has an identity element, which is nothing but the all-ones vector $(1, \ldots, 1)$ denoted by $\mathbf{1}$.

B. Polynomials

The ring of polynomials with coefficients in $\mathbb{F}_q$ is denoted by $\mathbb{F}_q[z]$, while the subspace of $\mathbb{F}_q[z]$ of polynomials of degree less than t is denoted by $\mathbb{F}_q[z]_{<t}$. For every rational fraction $P \in \mathbb{F}_q(z)$, with no poles at the elements $u_0, \ldots, u_{n-1}$, $P(\boldsymbol{u})$ stands for $(P(u_0), \ldots, P(u_{n-1}))$. In particular for all $a, b \in \mathbb{F}_q$, $a\boldsymbol{u} + b$ is the vector $(au_0 + b, \ldots, au_{n-1} + b)$.

The *norm* and *trace* from $\mathbb{F}_{q^m}$ to $\mathbb{F}_q$ can be viewed as polynomials and applied componentwise to vectors in $\mathbb{F}_{q^m}^n$. In the present article we focus in particular on quadratic extensions ($m = 2$) which motivates the following notation for all $\boldsymbol{x} \in \mathbb{F}_{q^2}^n$:

$$\begin{aligned} \mathrm{N}(\boldsymbol{x}) &\stackrel{\text{def}}{=} \left(x_0^{q+1}, \ldots, x_{n-1}^{q+1}\right) \\ \mathrm{Tr}(\boldsymbol{x}) &\stackrel{\text{def}}{=} \left(x_0^q + x_0, \ldots, x_{n-1}^q + x_{n-1}\right). \end{aligned}$$

Finally, to each vector $\boldsymbol{x} = (x_0, \ldots, x_{n-1}) \in \mathbb{F}_q^n$, we associate its *locator polynomial* denoted as $\pi_{\boldsymbol{x}}$ and defined as:

$$\pi_{\boldsymbol{x}}(z) \stackrel{\text{def}}{=} \prod_{i=0}^{n-1} (z - x_i).$$

C. Operations on codes

For a given code $\mathscr{D} \subseteq \mathbb{F}_q^n$ and a subset $\mathcal{I} \subseteq \{0, \ldots, n-1\}$ the *punctured* code $\mathcal{P}_{\mathcal{I}}(\mathscr{D})$ and *shortened* code $\mathcal{S}_{\mathcal{I}}(\mathscr{D})$ are defined as:

$$\begin{aligned} \mathcal{P}_{\mathcal{I}}(\mathscr{D}) &\stackrel{\text{def}}{=} \left\{(c_i)_{i \notin \mathcal{I}} \mid \boldsymbol{c} \in \mathscr{D}\right\}; \\ \mathcal{S}_{\mathcal{I}}(\mathscr{D}) &\stackrel{\text{def}}{=} \left\{(c_i)_{i \notin \mathcal{I}} \mid \exists \boldsymbol{c} = (c_i)_i \in \mathscr{D} \text{ s.t. } \forall i \in \mathcal{I},\ c_i = 0\right\}. \end{aligned}$$

Instead of writing $\mathcal{P}_{\{j\}}(\mathscr{D})$ and $\mathcal{S}_{\{j\}}(\mathscr{D})$ when $\mathcal{I} = \{j\}$ we rather use the notation $\mathcal{P}_j(\mathscr{D})$ and $\mathcal{S}_j(\mathscr{D})$. The following classical result will be used repeatedly.

Proposition 1. *Let $\mathscr{A} \subseteq \mathbb{F}_q^n$ be a code and $\mathcal{I} \subseteq \{0, \ldots, n-1\}$ be a set of positions. Then,*

$$\mathcal{S}_{\mathcal{I}}(\mathscr{A})^{\perp} = \mathcal{P}_{\mathcal{I}}\left(\mathscr{A}^{\perp}\right) \quad \text{and} \quad \mathcal{P}_{\mathcal{I}}(\mathscr{A})^{\perp} = \mathcal{S}_{\mathcal{I}}\left(\mathscr{A}^{\perp}\right).$$

Proof: See for instance [47, Theorem 1.5.7] ∎

Given a code $\mathscr{C}$ of length n over a finite field extension $\mathbb{F}_{q^m}$ of $\mathbb{F}_q$, the *subfield subcode* of $\mathscr{C}$ over $\mathbb{F}_q$ is the code $\mathscr{C} \cap \mathbb{F}_q^n$. The *trace code* $\mathrm{Tr}(\mathscr{C})$ is the image of $\mathscr{C}$ by the componentwise trace map $\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}$. We recall an important result due to Delsarte establishing a link between subfield subcodes and trace codes.

Theorem 2 (Delsarte Theorem [48, Theorem 2])**.** *Let $\mathscr{E}$ be a linear code of length n defined over $\mathbb{F}_{q^m}$. Then*

$$(\mathscr{E} \cap \mathbb{F}_q^n) = \mathit{Tr}(\mathscr{E}^{\perp})^{\perp}.$$

The following classical result is extremely useful in the next sections.

Proposition 3. *Let $\mathscr{A}$ be a code over $\mathbb{F}_{q^m}$ of length n and $\mathcal{I} \subseteq \{0, \ldots, n-1\}$, then, we have:*
(a) $\mathcal{P}_{\mathcal{I}}(\mathit{Tr}(\mathscr{A})) = \mathit{Tr}(\mathcal{P}_{\mathcal{I}}(\mathscr{A}))$;
(b) $\mathit{Tr}(\mathcal{S}_{\mathcal{I}}(\mathscr{A})) \subseteq \mathcal{S}_{\mathcal{I}}(\mathit{Tr}(\mathscr{A}))$;
(c) $\mathcal{S}_{\mathcal{I}}(\mathscr{A}) \cap \mathbb{F}_q^{n-|\mathcal{I}|} = \mathcal{S}_{\mathcal{I}}\left(\mathscr{A} \cap \mathbb{F}_q^n\right)$;
(d) $\mathcal{P}_{\mathcal{I}}\left(\mathscr{A} \cap \mathbb{F}_q^n\right) \subseteq \mathcal{P}_{\mathcal{I}}(\mathscr{A}) \cap \mathbb{F}_q^{n-|\mathcal{I}|}$.

Proof: The componentwise trace map and the puncturing map commute with each other, which proves (a). To prove (b), let $\boldsymbol{c} \in \mathscr{A}$ be a codeword whose entries with indexes in $\mathcal{I}$ are all equal to 0. We have $\mathrm{Tr}(c_{\mathcal{I}}) \in \mathrm{Tr}(\mathcal{S}_{\mathcal{I}}(\mathscr{A}))$. Moreover, the entries of $\mathrm{Tr}(\boldsymbol{c})$ with indexes in $\mathcal{I}$ are also equal to 0, hence $\mathrm{Tr}(\boldsymbol{c}_{\mathcal{I}}) = \mathrm{Tr}(\boldsymbol{c})_{\mathcal{I}} \in \mathcal{S}_{\mathcal{I}}(\mathrm{Tr}(\mathscr{A}))$. This proves (b). By duality, (c) and (d) can be directly deduced from (a) and (b) thanks to Proposition 1 and Theorem 2. ∎

D. Generalized Reed–Solomon and Alternant codes

Definition 1 (Generalized Reed-Solomon code)**.** Let q be a prime power and k, n be integers such that $1 \leqslant k < n \leqslant q$. Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be two n-tuples such that the entries of $\boldsymbol{x}$ are pairwise distinct elements of $\mathbb{F}_q$ and those of $\boldsymbol{y}$ are nonzero

elements in $\mathbb{F}_q$. The *generalized Reed-Solomon* code (GRS in short) $\mathbf{GRS}_k(\boldsymbol{x},\boldsymbol{y})$ of dimension k associated to $(\boldsymbol{x},\boldsymbol{y})$ is defined as

$$\begin{aligned}\mathbf{GRS}_k(\boldsymbol{x},\boldsymbol{y}) &\stackrel{\text{def}}{=} \left\{ \left(y_i p(x_i)\right)_{0\leqslant i<n} \;\middle|\; p\in\mathbb{F}_q[z]_{<k} \right\}\\ &= \left\{ \boldsymbol{y}\star p(\boldsymbol{x}) \;\middle|\; p\in\mathbb{F}_q[z]_{<k} \right\}.\end{aligned}$$

Reed-Solomon codes correspond to the case where $\boldsymbol{y}=\mathbf{1}$ and are denoted as $\mathbf{RS}_k(\boldsymbol{x})$. The vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are called the *support* and the *multiplier* of the code.

In the sequel, we will also use the terms support and multiplier without refering to a generalized Reed-Solomon code – this term will also appear in the context of Goppa and alternant codes. In this case, when we say that a vector $\boldsymbol{x}\in\mathbb{F}_q^n$ is a *support*, this means that all its entries are distinct. Likewise, when we say that a vector $\boldsymbol{y}\in\mathbb{F}_q^n$ is a *multiplier*, this means that all its entries are different from zero.

Proposition 4. *Let $\boldsymbol{x},\boldsymbol{y}$ be as in Definition 1. Then,*

$$\mathbf{GRS}_k(\boldsymbol{x},\boldsymbol{y})^\perp = \mathbf{GRS}_{n-k}(\boldsymbol{x},\boldsymbol{y}^{-1}\star\pi'_{\boldsymbol{x}}(\boldsymbol{x})^{-1})$$

where $\pi_{\boldsymbol{x}}$ is the locator polynomial of $\boldsymbol{x}$ as defined in § II-B and $\pi'_{\boldsymbol{x}}$ denotes its first derivative.

Proof: See for instance [49, Prop. 5.2 & Problems 5.6, 5.7]. ∎

This leads to the definition of alternant codes ([46, Chap. 12, § 2]).

Definition 2 (Alternant code)**.** Let $\boldsymbol{x},\boldsymbol{y}\in\mathbb{F}_{q^m}^n$ be a support and a multiplier. Let ℓ be a positive integer, the alternant code $\mathscr{A}_\ell(\boldsymbol{x},\boldsymbol{y})$ defined over $\mathbb{F}_q$ is defined as

$$\mathscr{A}_\ell(\boldsymbol{x},\boldsymbol{y}) \stackrel{\text{def}}{=} \mathbf{GRS}_\ell(\boldsymbol{x},\boldsymbol{y})^\perp\cap\mathbb{F}_q^n.$$

The integer ℓ is referred to as the *degree* of the alternant code and m as its *extension degree*.

Proposition 5 ([46, Chap. 12, § 2])**.** *Let $\boldsymbol{x},\boldsymbol{y}$ be as in Definition 2.*

1) $\dim_{\mathbb{F}_q}\mathscr{A}_\ell(\boldsymbol{x},\boldsymbol{y})\geqslant n-m\ell$*;*
2) $d_{min}(\mathscr{A}_\ell(\boldsymbol{x},\boldsymbol{y}))\geqslant\ell+1$*;*

where $d_{min}(\cdot)$ denotes the minimum distance of a code.

From Definition 2, it is clear that alternant codes inherit the decoding algorithms of the underlying GRS codes. The key feature of an alternant code is the following fact (see [46, Chap. 12, § 9]):

Fact 1. *There exists a polynomial time algorithm decoding all errors of Hamming weight at most $\lfloor\frac{\ell}{2}\rfloor$ once the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are known.*

The following description of alternant codes, will be extremely useful in this article.

Lemma 6. *Let $\boldsymbol{x}$, $\boldsymbol{y}$, ℓ be as in Definition 2. We have:*

$$\mathscr{A}_\ell(\boldsymbol{x},\boldsymbol{y}) = \left\{ \left(\frac{f(x_i)}{y_i\pi'_{\boldsymbol{x}}(x_i)}\right)_{0\leqslant i<n} \;\middle|\; f\in\mathbb{F}_{q^m}[z]_{<n-\ell} \right\}\cap\mathbb{F}_q^n.$$

E. Classical Goppa codes

Definition 3. Let $\boldsymbol{x}\in\mathbb{F}_{q^m}^n$ be a support and $\Gamma\in\mathbb{F}_{q^m}[z]$ be a polynomial such that $\Gamma(x_i)\neq 0$ for all $i\in\{0,\dots,n-1\}$. The classical Goppa code $\mathscr{G}(\boldsymbol{x},\Gamma)$ over $\mathbb{F}_q$ associated to Γ and supported by $\boldsymbol{x}$ is defined as

$$\mathscr{G}(\boldsymbol{x},\Gamma)\stackrel{\text{def}}{=}\mathscr{A}_{\deg\Gamma}(\boldsymbol{x},\Gamma(\boldsymbol{x})^{-1}).$$

We call Γ the *Goppa polynomial* and m the *extension degree* of the Goppa code.

As for alternant codes, the following description of Goppa codes, which is due to Lemma 6 will be extremely useful in this article.

Lemma 7. *Let $\boldsymbol{x},\Gamma$ be as in Definition 3. We have,*

$$\mathscr{G}(\boldsymbol{x},\Gamma) = \left\{ \left(\frac{\Gamma(x_i)f(x_i)}{\pi'_{\boldsymbol{x}}(x_i)}\right)_{0\leqslant i<n} \;\middle|\; f\in\mathbb{F}_{q^m}[z]_{<n-r} \right\}\cap\mathbb{F}_q^n,$$

where $r\stackrel{\text{def}}{=}\deg(\Gamma)$.

The interesting point about this subfamily of alternant codes is that under some conditions, Goppa codes can correct more errors than a general alternant code.

Theorem 8 ([50, Theorem 4])**.** *Let $\gamma\in\mathbb{F}_{q^m}[z]$ be a squarefree polynomial. Let $\boldsymbol{x}\in\mathbb{F}_{q^m}^n$ be a support, then*

$$\mathscr{G}\left(\boldsymbol{x},\gamma^{q-1}\right)=\mathscr{G}\left(\boldsymbol{x},\gamma^q\right).$$

Codes with such a Goppa polyomial are called *wild Goppa codes*. From Fact 1, wild Goppa codes correct up to $\lfloor\frac{qr}{2}\rfloor$ errors in polynomial-time instead of just $\lfloor\frac{(q-1)r}{2}\rfloor$ if viewed as $\mathscr{A}_{r(q-1)}(\boldsymbol{x},\gamma^{-(q-1)}(\boldsymbol{x})))$. On the other hand, these codes have dimension $\geqslant n-mr(q-1)$ instead of $\geqslant n-mrq$. Notice that when $q=2$, this amounts to double the error correction capacity. It is one of the reasons why binary Goppa codes have been chosen in the original McEliece scheme or why Goppa codes with Goppa polynomials of the form γ^{q-1} are proposed in [19], [20].

Remark 1. Actually, [50, Theorem 4] is more general and asserts that given irreducible polynomials $f_1,\dots,f_s\in\mathbb{F}_{q^m}[z]$, a polynomial g prime to $f_1\cdots f_s$ and positive integers $a_1,\dots,a_s$, then

$$\mathscr{G}\left(\boldsymbol{x},f_1^{a_1q-1}\cdots f_s^{a_sq-1}g\right)=\mathscr{G}\left(\boldsymbol{x},f_1^{a_1q}\cdots f_s^{a_sq}g\right).$$

F. Shortening Alternant and Goppa codes

The shortening operation will play a crucial role in our attack. For this reason, we recall the following classical result. We give a proof because of a lack of references.

Proposition 9. *Let $\boldsymbol{x}\in\mathbb{F}_{q^m}^n$ be a support and let $\boldsymbol{y}\in\mathbb{F}_{q^m}^n$ be a multiplier, then*

$$\mathcal{S}_{\mathcal{I}}\left(\mathscr{A}_r(\boldsymbol{x},\boldsymbol{y})\right)=\mathscr{A}_r(\boldsymbol{x}_{\mathcal{I}},\boldsymbol{y}_{\mathcal{I}}).$$

Proof: This proposition follows on the spot from the definition of the alternant code $\mathscr{A}_r(\boldsymbol{x},\boldsymbol{y})$: there is a parity-check $\boldsymbol{H}$ for it with entries over $\mathbb{F}_{q^m}$ which is the generating matrix of $\mathbf{GRS}_r(\boldsymbol{x},\boldsymbol{y})$. A parity-check matrix of the shortened code

$\mathcal{S}_{\mathcal{I}}\left(\mathscr{A}_r(\boldsymbol{x},\boldsymbol{y})\right)$ is obtained by throwing away the columns of $\boldsymbol{H}$ that belong to $\mathcal{I}$. That is to say, by puncturing $\mathbf{GRS}_r(\boldsymbol{x},\boldsymbol{y})$ at $\mathcal{I}$. This parity-check matrix is therefore the generator matrix of $\mathbf{GRS}_r(\boldsymbol{x}_{\mathcal{I}},\boldsymbol{y}_{\mathcal{I}})$ and the associated code is $\mathscr{A}_r(\boldsymbol{x}_{\mathcal{I}},\boldsymbol{y}_{\mathcal{I}})$. ∎

Corollary 10. *Let $\Gamma \in \mathbb{F}_{q^m}[z]$ and $\boldsymbol{x} \in \mathbb{F}_{q^m}^n$ be a support. Let $\mathcal{I} \subseteq \{0,\dots,n-1\}$, then*

$$\mathcal{S}_{\mathcal{I}}\left(\mathscr{G}\left(\boldsymbol{x},\Gamma\right)\right) = \mathscr{G}\left(\boldsymbol{x}_{\mathcal{I}},\Gamma\right).$$

G. McEliece encryption scheme

We recall here the general principle of McEliece public-key scheme [1]. The key generation algorithm picks a random $k \times n$ generator matrix $\boldsymbol{G}$ of a code $\mathscr{C}$ over $\mathbb{F}_q$ which is itself randomly picked in a family of codes for which t errors can be efficiently corrected. The *secret* key is the decoding algorithm $\mathcal{D}$ associated to $\mathscr{C}$ and the *public* key is $\boldsymbol{G}$. To encrypt $\boldsymbol{u} \in \mathbb{F}_q^k$, the sender chooses a random vector $\boldsymbol{e}$ in $\mathbb{F}_q^n$ of Hamming weight less than or equal to t and computes the ciphertext $\boldsymbol{c} = \boldsymbol{u}\boldsymbol{G} + \boldsymbol{e}$. The receiver then recovers the plaintext by applying $\mathcal{D}$ on $\boldsymbol{c}$.

McEliece based his scheme solely on binary Goppa codes. In [19], [20], it is advocated to use q-ary wild Goppa codes, *i.e.* codes with Goppa polynomials of the form γ^{q-1} because of their better error correction capability (Theorem 8). In this paper, we precisely focus on these codes but defined over quadratic extensions ($m = 2$). We shall see how it is possible to fully recover their secret structure under some mild condition on q and the degree of γ (further details in Table II).

III. A Distinguisher based on Square Codes

A. Square code

One of the keys for the distinguisher presented here and the attack outlined in the subsequent sections is a special property of certain alternant codes with respect to the component-wise product.

Definition 4 (Product of codes, square code)**.** Let $\mathscr{A}$ and $\mathscr{B}$ be two codes of length n. The *Schur product code* denoted by $\mathscr{A} \star \mathscr{B}$ is the vector space spanned by all products $\boldsymbol{a} \star \boldsymbol{b}$ for all $(\boldsymbol{a},\boldsymbol{b}) \in \mathscr{A} \times \mathscr{B}$. When $\mathscr{B} = \mathscr{A}$, $\mathscr{A} \star \mathscr{A}$ is called the *square code* of $\mathscr{A}$ and is denoted by $\mathscr{A}^{\star 2}$.

The dimension of the Schur product is easily bounded by:

Proposition 11. *Let $\mathscr{A}$ and $\mathscr{B}$ be two linear codes $\subseteq \mathbb{F}_q^n$ of dimensions k_A and k_B respectively, then if $k_{A\cap B}$ denotes the dimension of $\mathscr{A} \cap \mathscr{B}$ we have*

$$\dim\left(\mathscr{A}\star\mathscr{B}\right) \leqslant \min\left\{n,\ k_A k_B - \binom{k_{A\cap B}}{2}\right\} \quad (1)$$

$$\dim\left(\mathscr{A}^{\star 2}\right) \leqslant \min\left\{n,\ \binom{k_A+1}{2}\right\}. \quad (2)$$

Proof: Let $\{e_1,\dots,e_s\}$ be a basis of $\mathscr{A}\cap\mathscr{B}$. Complete it as two bases $B_{\mathscr{A}} = \{e_1,\dots,e_s,a_{s+1},\dots,a_k\}$ and $B_{\mathscr{B}} = \{e_1,\dots,e_s,b_{s+1},\dots,b_\ell\}$ of $\mathscr{A}$ and $\mathscr{B}$ respectively. The Schur products $\boldsymbol{u}\star\boldsymbol{v}$ where $\boldsymbol{u}\in B_{\mathscr{A}}$ and $\boldsymbol{v}\in B_{\mathscr{B}}$ span $\mathscr{A}\star\mathscr{B}$. The number of such products is $k\ell = \dim\mathscr{A}\dim\mathscr{B}$ minus the number of products which are counted twice, namely the products $e_i \star e_j$ with $i \neq j$ and their number is precisely $\binom{s}{2}$. This proves (1). The inequality given in (2) is a consequence of (1). ∎

It is proved in [51], [52] that, almost all codes of a given length and dimension reach these bounds while GRS codes behave completely differently when they have the same support.

Proposition 12. *Let $\boldsymbol{x} \in \mathbb{F}_q^n$ be a support and $\boldsymbol{y}$, $\boldsymbol{y}'$ be two multipliers in $\mathbb{F}_q^n$. Then,*

(i) $\mathbf{GRS}_k(\boldsymbol{x},\boldsymbol{y}) \star \mathbf{GRS}_{k'}(\boldsymbol{x},\boldsymbol{y}') = \mathbf{GRS}_{k+k'-1}(\boldsymbol{x},\boldsymbol{y}\star\boldsymbol{y}')$;

(ii) $\mathbf{GRS}_k(\boldsymbol{x},\boldsymbol{y})^{\star 2} = \mathbf{GRS}_{2k-1}(\boldsymbol{x},\boldsymbol{y}\star\boldsymbol{y})$.

This proposition shows that the dimension of $\mathbf{GRS}_k(\boldsymbol{x},\boldsymbol{y})\star\mathbf{GRS}_{k'}(\boldsymbol{x},\boldsymbol{y}')$ does not scale multiplicatively as kk' but additively as $k+k'-1$. It has been used the first time in cryptanalysis in [11] and appears for instance explicitly as Proposition 10 in [53]. We provide the proof here because it is crucial for understanding why the Schur products of GRS codes and some alternant codes behave in a non generic way.

Proof of Proposition 12: In order to prove (i), let $\boldsymbol{c} = (y_0 f(x_0),\dots,y_{n-1}f(x_{n-1})) \in \mathbf{GRS}_k(\boldsymbol{x},\boldsymbol{y})$ and $\boldsymbol{c}' = (y'_0 g(x_0),\dots,y'_{n-1}g(x_{n-1})) \in \mathbf{GRS}_{k'}(\boldsymbol{x},\boldsymbol{y}')$ where $\deg(f) \leqslant k-1$ and $\deg(g) \leqslant k'-1$. Then $\boldsymbol{c}\star\boldsymbol{c}'$ is of the form:

$$\begin{aligned}\boldsymbol{c}\star\boldsymbol{c}' &= (y_0y'_0 f(x_0)g(x_0),\dots,y_{n-1}y'_{n-1}f(x_{n-1})g(x_{n-1}))\\ &= (y_0y'_0 r(x_0),\dots,y_{n-1}y'_{n-1}r(x_{n-1}))\end{aligned}$$

where $\deg(r) \leqslant k+k'-2$. Conversely, any element $(y_0y'_0r(x_0),\dots,y_{n-1}y'_{n-1}r(x_{n-1}))$ where $\deg(r)\leqslant k+k'-2$, is a linear combination of Schur products of two elements of $\mathbf{GRS}_k(\boldsymbol{x},\boldsymbol{y})$. Statement (ii) is a consequence of (i) by putting $\boldsymbol{y}' = \boldsymbol{y}$ and $k' = k$. ∎

Since an alternant code is a subfield subcode of a GRS code, we might suspect that products of alternant codes have also low dimension compared to products of random codes. This is true but in a very attenuated form as shown by:

Theorem 13. *Let $\boldsymbol{x} \in \mathbb{F}_{q^m}^n$ be a support and $\boldsymbol{y},\boldsymbol{y}' \in \mathbb{F}_{q^m}^n$ be two multipliers. Then,*

$$\mathscr{A}_s(\boldsymbol{x},\boldsymbol{y}) \star \mathscr{A}_{s'}(\boldsymbol{x},\boldsymbol{y}') \subseteq \mathscr{A}_{s+s'-n+1}(\boldsymbol{x},\boldsymbol{y}''), \quad (3)$$

for $\boldsymbol{y}'' \stackrel{def}{=} \boldsymbol{y}\star\boldsymbol{y}'\star\pi'_{\boldsymbol{x}}(\boldsymbol{x})$.

Proof: Let $\boldsymbol{c},\boldsymbol{c}'$ be respective elements of $\mathscr{A}_s(\boldsymbol{x},\boldsymbol{y})$ and $\mathscr{A}_{s'}(\boldsymbol{x},\boldsymbol{y}')$. From Lemma 6,

$$\boldsymbol{c} = f(\boldsymbol{x})\star\boldsymbol{y}^{-1}\star\pi'_{\boldsymbol{x}}(\boldsymbol{x})^{-1} \quad \text{and} \quad \boldsymbol{c}' = g(\boldsymbol{x})\star\boldsymbol{y}'^{-1}\star\pi'_{\boldsymbol{x}}(\boldsymbol{x})^{-1}$$

for some polynomials f and g of degree $< n-s$ and $< n-s'$ respectively. This implies that

$$\boldsymbol{c}\star\boldsymbol{c}' = h(\boldsymbol{x})\star\boldsymbol{y}^{-1}\star\boldsymbol{y}'^{-1}\star\pi'_{\boldsymbol{x}}(\boldsymbol{x})^{-2}$$

where $h \stackrel{\text{def}}{=} fg$ is a polynomial of degree $< 2n-(s+s')-1$. Moreover, since $\boldsymbol{c}$ and $\boldsymbol{c}'$ have their entries in $\mathbb{F}_q$ then so has $\boldsymbol{c}\star\boldsymbol{c}'$. Consequently,

$$\boldsymbol{c}\star\boldsymbol{c}' \in \mathbf{GRS}_{2n-(s+s')-1}(\boldsymbol{x},\boldsymbol{y}^{-1}\star\boldsymbol{y}'^{-1}\star\pi'_{\boldsymbol{x}}(\boldsymbol{x})^{-2}) \cap \mathbb{F}_q^n.$$

From Definition 2, the above code equals $\mathscr{A}_{s+s'-n+1}(\boldsymbol{x},\boldsymbol{y}'')$ for $\boldsymbol{y}'' = \boldsymbol{y}\star\boldsymbol{y}'\star\pi'_{\boldsymbol{x}}(\boldsymbol{x})$. ∎

B. The particular case of wild Goppa codes over quadratic extensions

Theorem 13 generalizes Proposition 12 which corresponds to the particular case where the extension degree m is equal to 1. However, when $m > 1$, the right hand term of (3) is in general the full space $\mathbb{F}_q^n$. Indeed, assume that $m > 1$ and that the dimensions of $\mathscr{A}_s(\boldsymbol{x}, \boldsymbol{y})$ and $\mathscr{A}_{s'}(\boldsymbol{x}, \boldsymbol{y}')$ are equal to $n - sm$ and $n - s'm$ respectively. If we assume that both codes have non trivial dimensions then we should have $n - sm > 0$ and $n - s'm > 0$ which implies that both s and s' are $< n/m \leqslant n/2$ and hence:

$$(s + s') - n + 2 \leqslant 0$$

which entails that $\mathscr{A}_{s+s'-n+1}(\boldsymbol{x}, \boldsymbol{y}'')$ is the full space $\mathbb{F}_q^n$. However, in the case $m = 2$ and when either:

(i) $\mathscr{A}_s(\boldsymbol{x}, \boldsymbol{y})$ or $\mathscr{A}_{s'}(\boldsymbol{x}, \boldsymbol{y}')$ has dimension which exceeds the lower bound $n - sm$ or $n - s'm$
(ii) or when one of these codes is actually an alternant code for a larger degree *i.e.* $\mathscr{A}_s(\boldsymbol{x}, \boldsymbol{y}) = \mathscr{A}_{s''}(\boldsymbol{x}, \boldsymbol{y}')$ for $s'' > s$ and some multiplier $\boldsymbol{y}'$

then the right-hand term of (3) may be smaller than the full space. This is precisely what happens for wild Goppa codes of extension degree 2 as shown by the following statement.

Theorem 14 ([54])**.** *Let $\mathscr{G}\left(\boldsymbol{x}, \gamma^{q-1}\right)$ be a wild Goppa code of length n defined over $\mathbb{F}_q$ with support $\boldsymbol{x} \in \mathbb{F}_{q^2}^n$ where $\gamma \in \mathbb{F}_{q^2}[z]$ is irreducible of degree $r > 1$. Then,*

(i) $\mathscr{G}\left(\boldsymbol{x}, \gamma^{q-1}\right) = \mathscr{G}\left(\boldsymbol{x}, \gamma^{q+1}\right)$
(ii) $\dim(\mathscr{G}\left(\boldsymbol{x}, \gamma^{q+1}\right)) \geqslant n - 2r(q+1) + r(r+2)$
(iii) $\mathscr{G}\left(\boldsymbol{x}, \gamma^{q+1}\right) = \boldsymbol{u} \star \mathscr{A}_{r(q+1)}(\boldsymbol{x}, \mathbf{1})$ *for some multiplier* $\boldsymbol{u} \in \mathbb{F}_q^n$.

Proof: The results (i) and (ii) are straightforward consequences of Theorems 1 and 24 of [54]. Only (iii) requires further details. First, let us consider the case where $\boldsymbol{x}$ is a full–support, that is if $n = q^2$, then from [54, Corollary 10], we have

$$\mathscr{G}\left(\boldsymbol{x}, \gamma^{q+1}\right) = \boldsymbol{a} \star \left(\mathbf{RS}_{q^2-r(q+1)}(\boldsymbol{x}) \cap \mathbb{F}_q^n\right), \tag{4}$$

for some multiplier $\boldsymbol{a} \in \mathbb{F}_q^n$. Then, from Proposition 4, we have

$$\mathbf{RS}_{q^2-r(q+1)}(\boldsymbol{x}) = \mathbf{GRS}_{r(q+1)}(\boldsymbol{x}, \pi'_{\boldsymbol{x}}(\boldsymbol{x})^{-1})^{\perp}.$$

Since $\boldsymbol{x}$ is assumed to be full then $\pi'_{\boldsymbol{x}}(\boldsymbol{x}) = \mathbf{1}$. Therefore, from Definition 2 we see that (4) is equivalent to:

$$\begin{aligned} \mathscr{G}\left(\boldsymbol{x}, \gamma^{q+1}\right) &= \boldsymbol{u} \star (\mathbf{RS}_{r(q+1)}(\boldsymbol{x})^{\perp} \cap \mathbb{F}_q^n) && (5) \\ &= \boldsymbol{u} \star \mathscr{A}_{r(q+1)}(\boldsymbol{x}, \mathbf{1}), && (6) \end{aligned}$$

which yields (iii). The general case, i.e. when $\boldsymbol{x}$ is not full can be deduced from the full support case by shortening thanks to Proposition 9. ∎

C. Wild Goppa codes with non generic squares

In what follows, we prove that a wild Goppa code over a quadratic extension $\mathscr{G}\left(\boldsymbol{x}, \gamma^{q-1}\right)$ whose length belongs to some interval $[n_-, n_+]$ has a square with a non generic behaviour. The bounds n_- and n_+ of the interval depend only on the degree of γ and will be explicitly described. The corresponding wild Goppa codes have rather short length compared to the full support ones and very low rate (ratio k/n where k denotes the dimension).

We emphasize that public keys proposed for McEliece have not a length in the interval $[n_-, n_+]$. However, thanks to Corollary 10, a shortening of the public key is a wild Goppa code with the same Goppa polynomial but with a shorter length and a lower rate. This is the point of our "distinguisher by shortening" described in the subsequent sections.

1) Context: In what follows and until the end of the article, $\gamma \in \mathbb{F}_{q^2}[z]$ is an irreducible polynomial (actually squarefree is sufficient) and $\mathscr{C} = \mathscr{G}\left(\boldsymbol{x}, \gamma^{q-1}\right)$ denotes the corresponding wild Goppa code over the quadratic extension $\mathbb{F}_{q^2}/\mathbb{F}_q$. The public key of the wild McEliece encryption scheme is a certain description of this code (for instance a systematic generator matrix of it).

2) The parameters of wild Goppa codes with non generic squares: We look for a sufficient condition on the length of $\mathscr{C}$ for its square to have a non generic behavior. From Theorem 14, we have $\mathscr{C} = \mathscr{G}\left(\boldsymbol{x}, \gamma^{q+1}\right)$. Thus, it is an alternant code of degree $r(q+1)$ and from Theorem 13:

$$\mathscr{C}^{\star 2} \subseteq \mathscr{A}_{2r(q+1)+1-n}(\boldsymbol{x}, \boldsymbol{y}) \tag{7}$$

for some multiplier $\boldsymbol{y} \in \mathbb{F}_{q^2}^n$. Let $\mathscr{R}$ be a random code with the same length and dimension as $\mathscr{C}$. With high probability, we get

$$\dim \mathscr{R}^{\star 2} = \min\left\{n\ ,\ \binom{\dim \mathscr{C} + 1}{2}\right\}. \tag{8}$$

Therefore, $\mathscr{C}$ is distinguishable from $\mathscr{R}$ when the following conditions are both satisfied:

$$\dim \mathscr{A}_{2r(q+1)+1-n}(\boldsymbol{x}, \boldsymbol{y}) < n \tag{D1}$$

$$\dim \mathscr{A}_{2r(q+1)+1-n}(\boldsymbol{x}, \boldsymbol{y}) < \binom{\dim \mathscr{C} + 1}{2}. \tag{D2}$$

Using the very definition of alternant codes (Definition 2), one proves easily that an alternant code is different from its ambient space if and only if its degree is positive. Thus, (D1) is equivalent to:

$$n \leqslant 2r(q+1). \tag{9}$$

Next, from Theorem 14(ii), we have

$$\dim \mathscr{C} \geqslant n - 2r(q+1) + r(r+2). \tag{10}$$

Moreover, from Theorem 5(1) on the dimension of alternant codes, we have

$$\dim \mathscr{A}_{2r(q+1)+1-n}(\boldsymbol{x}, \boldsymbol{y}) \geqslant 3n - 4r(q+1) - 2. \tag{11}$$

Assume that the above lower bounds (10) and (11) on the dimensions of $\mathscr{C}$ and $\mathscr{A}_{2r(q+1)+1-n}(\boldsymbol{x}, \boldsymbol{y})$ are their actual dimension (which holds true in general). In such a case, (D2) becomes equivalent to

$$\binom{n - 2r(q+1) + r(r+2) + 1}{2} > 3n - 4r(q+1) - 2. \tag{12}$$

Therefore, if the length n of $\mathscr{C}$ satisfies both (9) and (12), then its square has a non generic dimension. Now, consider the map

$$\varphi : n \longmapsto \binom{n-2r(q+1)+r(r+2)+1}{2} - 3n + 4r(q+1) + 2.$$

Its first derivative is

$$\begin{aligned} \varphi'(n) &= n - 2r(q+1) + r(r+2) + \frac{1}{2} - 3 \\ &= n - 2r(q+1) + r(r+2) - \frac{5}{2}. \end{aligned}$$

Hence, for $n > 2r(q+1) - r(r+2) + \frac{5}{2}$, the map φ is increasing.

There are two cases to consider:

(i) either (12) is not satisfied for the largest possible value of n satisfying (9), namely $n = 2r(q+1)$ and then the fact that φ is increasing implies that it can not be satisfied by any value of n;

(ii) or (12) is satisfied for $n = 2r(q+1)$ and then the set of values n satisfying both (9) and (12) is an interval of the form $[n_-, n_+]$ where $n_+ = 2r(q+1)$ and n_- is the smallest n satisfying (12).

Notice that (12) is satisfied for $n = 2r(q+1)$ if and only if

$$\binom{r(r+2)+1}{2} > 2r(q+1) - 2. \tag{13}$$

a) **Conclusion:** Assuming that the above lower bounds (10) and (11) on the dimensions of $\mathscr{C}$ and $\mathscr{A}_{2r(q+1)+1-n}(\boldsymbol{x}, \boldsymbol{y})$ are their actual dimension and if (13) holds, then there is a nonempty interval $[n_-, n_+]$ of integers such that $\mathscr{C}^{\star 2}$ has a non generic behaviour for any n in $[n_-, n_+]$. Moreover,

1) $n_+ = 2r(q+1)$;
2) n_- is the least integer such that

$$\binom{n-2r(q+1)+r(r+2)+1}{2} > 3n - 4r(q+1) - 2.$$

For a length exceeding n_+, the square will probably be equal to the whole ambient space, while for a length less than n_-, the square will probably have a dimension equal to that of a square random code.

D. A distinguisher by shortening

It can easily be checked that proposed public keys for McEliece have a length far above the upper bound n_+ described in the previous section. However the previously described interval $[n_-, n_+]$ only depends on the degree r of γ. Moreover, according to Corollary 10, shortening $\mathscr{C}$ provides a shorter Goppa code with the same Goppa polynomial. This leads to the first fundamental result of this article.

Theorem 15. *Let $\mathscr{C}$ be the wild Goppa code $\mathscr{G}\left(\boldsymbol{x}, \gamma^{q-1}\right)$, where $\gamma \in \mathbb{F}_{q^2}[z]$ has degree $r < q$. If the following inequality holds,*

$$\binom{r(r+2)+1}{2} > 2r(q+1) - 2,$$

then there is a nonempty interval $[a_-, a_+] \subseteq [1, n]$ such that for all $\mathcal{I} \subseteq \{0, \ldots, n-1\}$ with $|\mathcal{I}| \in [a_-, a_+]$, the dimension of $\mathcal{S}_{\mathcal{I}}(\mathscr{C})^{\star 2}$ is less than that of almost all squares of random codes of the same length and dimension. Moreover,

1) $a_- = n - 2r(q+1)$;
2) a_+ is the largest integer such that

$$\binom{n-a_++r(r-2q)+1}{2} > 3(n-a_+) - 4r(q+1) - 2.$$

Proof: Apply the reasoning of § III-C2 to $\mathcal{S}_{\mathcal{I}}(\mathscr{C})$, *i.e.* replace everywhere n by $n - |\mathcal{I}|$. ∎

E. Experimental observation and example

Actually, in all our experiments we observed that $\mathcal{S}_{\mathcal{I}}(\mathscr{C})^{\star 2}$ has always codimension 1 in the code $\mathscr{A}_{2r(q+1)+1-n}(\boldsymbol{x}_{\mathcal{I}}, \boldsymbol{y}_{\mathcal{I}})$ (see (7)). This allows to replace the strict inequalities in (D1) and (D2) by large ones and provides a slightly larger distinguisher interval $[a_-, a_+]$, which turns out to be the actual distinguisher interval according to our experiments. Namely

1) $a_- = n - 2r(q+1) - 1$;
2) a_+ is the largest integer such that

$$\binom{n-a_++r(r-2q)+1}{2} > 3(n-a_+) - 4r(q+1) - 3.$$

This interval is nonempty as soon as:

$$\binom{r(r+2)+2}{2} > 2r(q+1). \tag{14}$$

We checked that this allows to distinguish from random codes all the wild Goppa codes of extension degree 2 suggested in [19] when $r > 3$. For instance, the first entry in [19, Table 7.1] is a wild Goppa code $\mathscr{C}$ defined over $\mathbb{F}_{29}$ of length 794, dimension 529 with a Goppa polynomial γ^{29} where $\deg \gamma = 5$. Table I shows that for a in the range $\{493, \ldots, 506\}$ the dimensions of $\mathcal{S}_{\mathcal{I}}(\mathscr{C})^{\star 2}$ differ from those of a random code with the same parameters. Note that for this example $a^- = 493$.

It is only when the degree of γ is very small and the field size large that we cannot distinguish the Goppa code in this way. In Table II, we gathered upper bounds on the field size for which we expect to distinguish $\mathscr{G}\left(\boldsymbol{x}, \gamma^{q-1}\right)$ from a random code in terms of the degree of γ.

TABLE I
DIMENSION OF $\mathcal{S}_{\mathcal{I}}(\mathscr{C})^{\star 2}$ WHEN $\mathscr{C}$ IS EITHER THE WILD GOPPA CODE IN THE FIRST ENTRY OF [19, TABLE 7.1], OR A RANDOM CODE OF THE SAME LENGTH AND DIMENSION FOR VARIOUS VALUES $|\mathcal{I}|$.

$\lvert\mathcal{I}\rvert$	493	494	495	496	497	498	499	500	501
Goppa	300	297	294	291	288	285	282	279	276
Random	301	300	299	298	297	296	295	294	293

$\lvert\mathcal{I}\rvert$	502	503	504	505	506	507	508	509	510
Goppa	273	270	267	264	261	253	231	210	190
Random	292	291	290	289	276	253	231	210	190

TABLE II
NUMERICAL ILLUSTRATION OF (14), I.E. LARGEST FIELD SIZE q FOR WHICH WE CAN EXPECT TO DISTINGUISH $\mathscr{G}\left(\boldsymbol{x}, \gamma^{q-1}\right)$ WHEN γ IS AN IRREDUCIBLE POLYNOMIAL IN $\mathbb{F}_{q^2}[z]$ OF DEGREE r.

r	2	3	4	5
q	9	19	37	64

IV. THE CODE FILTRATION

Remind that we are still in the context of Section III-C1. The crucial ingredient of our attack is the computation of a family of nested codes from the knowledge of the public key. According to the common terminology in commutative algebra, we call such a family a *filtration*. Roughly speaking, given the public code $\mathscr{C} = \mathscr{G}\left(\boldsymbol{x}, \gamma^{q-1}\right)$, we aim at computing a filtration:

$$\mathscr{C}_a(0) \supseteq \mathscr{C}_a(1) \supseteq \cdots \supseteq \mathscr{C}_a(s) \supseteq \cdots$$

such that $\mathscr{C}_a(0)$ is some puncturing of $\mathscr{C}$. Moreover, we wish the filtration to have a good behavior with respect to the Schur product. Ideally we would expect something like:

$$\text{“ } i+j=k+\ell \quad \Longrightarrow \quad \mathscr{C}_a(i) \star \mathscr{C}_a(j) = \mathscr{C}_a(k) \star \mathscr{C}_a(\ell).\text{”}$$

This is exactly what would happen if $\mathscr{C}$ is a GRS code (see §IV-A). Unfortunately, in the case of a wild Goppa code such a requirement is too strong and we will only have a weaker but sufficient version asserting that $\mathscr{C}_a(i) \star \mathscr{C}_a(j)$ and $\mathscr{C}_a(k) \star \mathscr{C}_a(\ell)$ are contained in a same alternant code. This is detailed further in Corollary 20.

Roughly speaking, the code $\mathscr{C}_a(j)$ (see Definition 5 below) consists in the codewords of $\mathscr{C}$ obtained from polynomials having a zero of order at least j at position a. The key point is that this filtration reveals a lot about the algebraic structure of $\mathscr{C}$. In particular, we will be able to recover the support from it. To understand the rationale behind such a filtration its computation and its use for cryptanalysis, let us start with an illustrative example on generalized Reed Solomon codes.

A. Illustrative example with GRS codes

Let $\boldsymbol{x}, \boldsymbol{y}$ be a support and a multiplier in $\mathbb{F}_{q^2}^n$. Let $k < n/2$. Assume that the codes $\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$ and $\mathbf{GRS}_{k-1}(\boldsymbol{x}, \boldsymbol{y})$ are known. We claim that from the single knowledge of these two codes, it possible to compute the whole filtration

$$\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y}) \supseteq \mathbf{GRS}_{k-1}(\boldsymbol{x}, \boldsymbol{y}) \supseteq \cdots \supseteq \mathbf{GRS}_1(\boldsymbol{x}, \boldsymbol{y}) \supseteq \{0\}. \tag{15}$$

Remark 2. In terms of polynomials, this filtration corresponds to:

$$\mathbb{F}_{q^2}[z]_{<k} \supseteq \mathbb{F}_{q^2}[z]_{<k-1} \supseteq \cdots \supseteq \cdots \supseteq \mathbb{F}_{q^2}[z]_{<1} \supseteq \{0\}. \tag{16}$$

Let us explain how we could compute $\mathbf{GRS}_{k-2}(\boldsymbol{x}, \boldsymbol{y})$. From Proposition 12, we obtain

$$\mathbf{GRS}_{k-2}(\boldsymbol{x}, \boldsymbol{y}) \star \mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y}) = \mathbf{GRS}_{k-1}(\boldsymbol{x}, \boldsymbol{y})^{\star 2} \tag{17}$$

and from this equality, one can prove that:

$$\mathbf{GRS}_{k-2}(\boldsymbol{x}, \boldsymbol{y}) = \left\{ \boldsymbol{c} \in \mathbf{GRS}_{k-1}(\boldsymbol{x}, \boldsymbol{y}) \mid \boldsymbol{c} \star \mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y}) \subseteq \mathbf{GRS}_{k-1}(\boldsymbol{x}, \boldsymbol{y})^{\star 2} \right\}. \tag{18}$$

Indeed, inclusion “$\subseteq$” is a direct consequence of (17). The converse inclusion can be obtained by studying the degrees in the associated spaces of polynomials (see [15, §6]). Thus, Equation (18) shows that $\mathbf{GRS}_{k-2}(\boldsymbol{x}, \boldsymbol{y})$ can be computed from the single knowledge of $\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$ and $\mathbf{GRS}_{k-1}(\boldsymbol{x}, \boldsymbol{y})$. By iterating this process, one can compute all the terms of the filtration (15). Finally, since the last nonzero term $\mathbf{GRS}_1(\boldsymbol{x}, \boldsymbol{y})$ is obtained by evaluation of constant polynomials, this space has dimension 1 and is spanned by $\boldsymbol{y}$. This yields $\boldsymbol{y}$ up to a multiplication by a scalar.

This is an illustration of how the computation of a filtration can provide crucial information on a code. On the other hand, there is no reason to know $\mathbf{GRS}_{k-1}(\boldsymbol{x}, \boldsymbol{y})$ especially in a cryptographic situation. However, some very particular subcodes of codimension 1 can be easily computed from the knowledge of $\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$. Namely, shortening $\mathcal{S}_i\left(\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})\right)$ at a single position $i \in \{0, \ldots, n-1\}$ can be computed by Gaussian elimination. This code corresponds to the space of polynomials vanishing at x_i, that is the space $(z - x_i)\mathbb{F}_q[z]_{<k-1}$, or, after a suitable change of variables, the space $z\mathbb{F}_q[z]_{<k-1}$ of polynomials vanishing at 0. Therefore, using the method described above, from $\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$ and its shortening at the i–th position, one can compute the filtration of codes corresponding to the spaces of polynomials:

$$\mathbb{F}_q[z]_{<k} \supseteq z\mathbb{F}_q[z]_{<k-1} \supseteq \cdots \supseteq z^{k-1}\mathbb{F}_q[z]_{<1} \supseteq \{0\}. \tag{19}$$

The computation of such filtrations permits a complete recovery of $\boldsymbol{x}, \boldsymbol{y}$ (see [15] for further details). This is exactly the spirit of our attack on wild Goppa codes.

Remark 3. From an algebraic geometric point of view, the filtrations (16) and (19) are very close to each other. Filtration (19) is the filtration associated to the valuation at 0 while filtration (16) is associated to the degree which can be regarded as a valuation at infinity. Thus, investigating a filtration like (19) is extremely natural.

B. The computation of particular subcodes

In the previous examplel and also in what follows, the computation of a term of a filtration can be done from the previous ones by solving a problem of the form:

Problem 1. *Given $\mathscr{A}$, $\mathscr{B}$, and $\mathscr{D}$ be three codes in $\mathbb{F}_q^n$ find the subcode $\mathscr{S}$ of elements $\boldsymbol{s}$ in $\mathscr{D}$ satisfying:*

$$\boldsymbol{s} \star \mathscr{A} \subseteq \mathscr{B}. \tag{20}$$

Such a code can be computed by linear algebra or equivalently by computing dual codes and Schur products. Namely, we have:

Proposition 16. *The solution space $\mathscr{S}$ of Problem 1 is:*

$$\mathscr{S} = \left(\mathscr{A} \star \mathscr{B}^{\perp}\right)^{\perp} \cap \mathscr{D}.$$

Proof: Let $\boldsymbol{s} \in \mathscr{S}$ then clearly $\boldsymbol{s} \in \mathscr{D}$. Let $\boldsymbol{a} \in \mathscr{A}$ and $\boldsymbol{b}^\perp \in \mathscr{B}^\perp$. Then,

$$\langle \boldsymbol{s}, \boldsymbol{a} \star \boldsymbol{b}^\perp \rangle = \sum_{i=0}^{n-1} z_i a_i b_i^\perp = \langle \boldsymbol{s} \star \boldsymbol{a}, \boldsymbol{b}^\perp \rangle$$

and this last term is zero by definition of $\mathscr{S}$. This proves $\mathscr{S} \subseteq \left(\mathscr{A} \star \mathscr{B}^\perp\right)^\perp \cap \mathscr{D}$. The converse inclusion is proved in the very same way. ∎

C. The filtration of alternant codes $\mathscr{C}_a(j)$

In the same spirit as the example of §IV-A, we will compute the terms of a filtration by solving iteratively problems of the form of Problem 1. The filtration we will compute is in some sense related to the polynomial spaces filtration:

$$\mathbb{F}_{q^2}[z]_{<k} \supseteq z\mathbb{F}_{q^2}[z]_{<k-1} \supseteq \cdots \supseteq z^{k-1}\mathbb{F}_{q^2}[z]_{<1} \supseteq \{0\}.$$

For that purpose, we introduce the following definition.

Definition 5. For all $a \in \{0, \ldots, n-1\}$ and for all $s \in \mathbb{Z}$, we define the code $\mathscr{C}_a(s)$ as the set of codewords of the form

$$\left(\frac{\gamma^{q+1}(x_i)(x_i - x_a)^s f(x_i)}{\pi'_{\boldsymbol{x}}(x_i)} \right)_{\substack{0 \leqslant i < n \\ i \neq a}}$$

which belong to $\mathbb{F}_q^{n-1}$ and where f belongs to $\mathbb{F}_{q^2}[z]_{<n-r(q+1)-s}$. Roughly speaking, for $s > 0$, the code $\mathscr{C}_a(s)$ is the subcode of $\mathcal{S}_a(\mathscr{C})$ obtained from rational fractions vanishing at x_a with order at least s.

The link with $\mathscr{C}$ becomes clearer if we use Theorem 14, which asserts that $\mathscr{C} = \mathscr{G}\left(\boldsymbol{x}, \gamma^{q+1}\right)$. Thanks to Lemma 7 on the description of Goppa codes as evaluation codes, we have:

$$\mathscr{C} = \left\{ \left(\frac{\gamma^{q+1}(x_i)}{\pi'_{\boldsymbol{x}}(x_i)} f(x_i) \right)_{0 \leqslant i < n} \middle| \; f \in \mathbb{F}_{q^2}[z]_{<n-r(q+1)} \right\} \cap \mathbb{F}_q^n. \tag{21}$$

From now on, we focus on the case $a = 0$ and assume that

Assumption 17. *(i)* $\mathcal{S}_0(\mathscr{C}) \neq \mathscr{C}$
(ii) $x_0 = 0,\ x_1 = 1$.

Discussion about these assumptions. If $\mathscr{C}$ is not the zero code, after possibly reordering the support we can always assume that the first position is not always equal to 0 in every codeword of $\mathscr{C}$ and therefore $\mathcal{S}_0(\mathscr{C}) \neq \mathscr{C}$. The second assumption can always be made, and this without reordering the support- this follows directly from Lemma 24.

Every statement in what follows could be reformulated for a general position a. This would however provide heavier notation which we have tried to avoid.

The following statement summarizes the properties of this filtration which are used in the attack. Since its IEEEproof is rather technical, we chose to postpone it in appendix.

Proposition 18. *Under Assumption 17 (i), we have*
(i) $\mathscr{C}_0(1) = \mathcal{S}_0(\mathscr{C})$;
(ii) $\mathscr{C}_0(0) = \mathcal{P}_0(\mathscr{C})$;
(iii) $\forall s \in \mathbb{Z},\ \dim \mathscr{C}_0(s) - \dim \mathscr{C}_0(s+1) \leqslant 2$;
(iv) $\mathscr{C}_0(q-r) = \mathscr{C}_0(q+1)$;
(v) $\forall s \in \mathbb{Z},\ \mathscr{C}_0(s) = \mathscr{A}_{r(q+1)+s-1}(\boldsymbol{x}_0, \boldsymbol{y}_0)$ *for*

$$\boldsymbol{y}_0 \stackrel{def}{=} \gamma^{-(q+1)}(\boldsymbol{x}_0) \star \boldsymbol{x}_0^{-(s-1)},$$

where we recall that $\boldsymbol{x}_0$ *denotes the vector* $\boldsymbol{x}$ *punctured at position* 0 *and that* r *denotes the degree of* γ.

Proof: Appendix A. ∎

Corollary 19. *For all* $s > 0$, *we have*

$$\dim \mathscr{C}_0(s) \geqslant n - 1 - 2r(q+1) - 2(s-1) + r(r+2).$$

Proof: The case $s = 1$ is a direct consequence of Proposition 18(i) since shortening at one position reduces the dimension from at most 1. Then the result is proved by induction on s using Proposition 18(iii). ∎

Corollary 20. *For all pair* s, s' *of integers,*

$$\mathscr{C}_0(s) \star \mathscr{C}_0(s') \subseteq \mathscr{A}_{2r(q+1)+s+s'-n}(\boldsymbol{x}_0, \boldsymbol{y}_0)$$

where $\boldsymbol{y}_0 = \gamma^{-2(q+1)}(\boldsymbol{x}_0) \star \boldsymbol{x}_0^{-(s+s'-2)} \star \pi'_{\boldsymbol{x}_0}(\boldsymbol{x}_0)$.

Proof: Apply Proposition 18(v) and Theorem 13 using the fact that $\mathscr{C}_0(s)$ and $\mathscr{C}_0(s')$ are of length $n-1$. ∎

D. The distinguisher intervals

The filtration $\left(\mathscr{C}_0(s)\right)_{s \in \mathbb{Z}}$ is strongly related to $\mathscr{C}$ since as explained in Proposition 18(i) and (ii), two elements of the filtration can easily be computed from the pulic key $\mathscr{C}$. Namely, the codes $\mathscr{C}_0(0)$ and $\mathscr{C}_0(1)$ are respectively obtained by puncturing and shortening $\mathscr{C}$ at position 0. The subsequent elements of the filtration will be computed iteratively by solving problems of the form of Problem 1 in the very same manner as in the example given in Section IV-A. For instance, we will compute $\mathscr{C}_0(2)$ from the "equation"

$$\text{“}\mathscr{C}_0(0) \star \mathscr{C}_0(2) \subseteq \mathscr{C}_0(1)^{\star 2}\text{”}$$

and more generally $\mathscr{C}_0(t)$, will be computed from

$$\text{“}\mathscr{C}_0(0) \star \mathscr{C}_0(t) \subseteq \mathscr{C}_0(\lfloor t/2 \rfloor) \star \mathscr{C}_0(\lceil t/2 \rceil)\text{”}. \tag{22}$$

Unfortunately, this relation is not strictly correct: according to Corollary 20, the right-hand term should be replaced by $\mathscr{A}_{2r(q+1)+t-n}(\boldsymbol{x}_0, \boldsymbol{y}_0)$ which is unknown. Moreover, as explained in §III the above Schur products fill in their ambient space. However, for some particular lengths it is possible to compute $\mathscr{C}_0(t)$ by solving a problem of the form of Problem 1. These lengths are those such that:

1) The alternant code $\mathscr{A}_{2r(q+1)+t-n}(\boldsymbol{x}_0, \boldsymbol{y}_0)$ does not fill in the ambient space.
2) The Schur product $\mathscr{C}_0(\lfloor t/2 \rfloor) \star \mathscr{C}_0(\lceil t/2 \rceil)$ should fill in $\mathscr{A}_{2r(q+1)+t-n}(\boldsymbol{x}_0, \boldsymbol{y}_0)$ or at least be a sufficiently large subcode of it.

Let $\mathscr{R}$ and $\mathscr{R}'$ be two random codes such that $\mathscr{R}' \subseteq \mathscr{R}$ and whose dimensions equal those of $\mathscr{C}_0(\lfloor t/2 \rfloor)$ and $\mathscr{C}_0(\lceil t/2 \rceil)$. For (2) to be satisfied, we expect that the dimension of $\mathscr{R} \star \mathscr{R}'$ exceeds that of $\mathscr{A}_{2r(q+1)+t-n}(\boldsymbol{x}_0, \boldsymbol{y}_0)$. Thus, the computation of $\mathscr{C}_0(t)$ is possible if the length of the codes is in

some particular interval. Therefore, it is possible to compute $\mathcal{S}_{\mathcal{I}}(\mathscr{C}_0(t))$ for a suitable set $\mathcal{I}$ and $|\mathcal{I}|$ is in some interval which is nothing but a distinguisher interval as computed in §III. We will compute these intervals in order to obtain $\mathscr{C}_0(t)$ by considering separately the cases of even and odd t.

1) The symmetric case: Assume that t is even:

$$t = 2s$$

for some positive integer s. From Corollary 19, we have

$$\dim \mathscr{C}_0(s) \geqslant (n-1) - 2(r(q+1)+s-1) + r(r+2).$$

Since, from Proposition 18(v), this code is alternant of degree $r(q+1)+s-1$. Then from Corollary 20:

$$\mathscr{C}_0(s)^{\star 2} \subseteq \mathscr{A}_{2(r(q+1)+s)-n}(\boldsymbol{x}_0, \boldsymbol{y}_0) \tag{23}$$

Thus, we are in the very same situation as in §III-D and the distinguisher interval for $\mathscr{C}_0(s)$ can be deduced from that of $\mathscr{C}$ by applying the changes of variables

$$\begin{array}{rcl} n & \longmapsto & n-1 \\ r(q+1) & \longmapsto & r(q+1)+s-1. \end{array}$$

Conclusion: If

$$\binom{r(r+2)+2}{2} > 2r(q+1)+t-2$$

then there is a nonempty interval $[b_-, b_+]$ such that for all $\mathcal{I} \subseteq \{1, \ldots, n-1\}$ with $|\mathcal{I}| \subseteq [b_-, b_+]$, the square of $\mathcal{S}_{\mathcal{I}}(\mathscr{C}_0(s))$ has a non generic behaviour. Moreover,

(1) $b_- = n - 2r(q+1) - t$;

(2) b_+ is the largest integer such that

$$\begin{aligned} \binom{n-b_+-t+2+r(r-2q)}{2} > \\ 3(n-1-b_+) - 4r(q+1) - 2t + 1. \end{aligned} \tag{24}$$

Remark 4. Actually, the above distinguisher interval, relies on an experimental observation similar to that of §III-E. Namely, we observed experimentally that (23) is a strict inclusion with codimension 1 as soon as the degree of the alternant code in the right hand term is non-negative.

2) The asymmetric case: As in §III-C2, we start by computing the interval for which the Schur product $\mathscr{C}_0(s) \star \mathscr{C}_0(s+1)$ has a non generic behaviour, then we can reduce to that case by shortening.

In the spirit of the distinguisher interval computed in §III. Instead of Equation (7), Corollary 20 yields

$$\mathscr{C}_0(s) \star \mathscr{C}_0(s+1) \subseteq \mathscr{A}_{2r(q+1)+t-n}(\boldsymbol{x}_0, \boldsymbol{y}_0). \tag{7'}$$

This leads to new distinguisher conditions:

$$\dim \mathscr{A}_{2r(q+1)+t-n}(\boldsymbol{x}_0, \boldsymbol{y}') < n-1 \tag{D1'}$$

$$\begin{aligned} \dim \mathscr{A}_{2r(q+1)+t-n}(\boldsymbol{x}_0, \boldsymbol{y}') < \dim \mathscr{C}_0(s) \dim \mathscr{C}_0(s+1) \\ - \binom{\dim \mathscr{C}_0(s+1)}{2}. \end{aligned} \tag{D2'}$$

According to Proposition 11(1), the right hand term of (D2') is the typical dimension of the Schur Product of two random codes of the same dimension as $\mathscr{C}_0(s)$ and $\mathscr{C}_0(s+1)$ if this product does not fill in the ambient space. From Corollary 19, we have

$$\begin{aligned} \dim \mathscr{C}_0(s) &\geqslant (n-1) - 2r(q+1) - 2s + 2 + r(r+2) \\ \dim \mathscr{C}_0(s+1) &\geqslant (n-1) - 2r(q+1) - 2s + r(r+2). \end{aligned}$$

Assuming that the above lower bounds are reached, which holds true in general, a computation from the formula of Proposition 11(1) gives

$$\dim \mathscr{C}_0(s) \dim \mathscr{C}_0(s+1) - \binom{\dim \mathscr{C}_0(s+1)}{2} = \frac{1}{2} d(d+5),$$

where

$$d \stackrel{\text{def}}{=} (n-1) - 2r(q+1) - 2s + r(r+2).$$

Conclusion: Proceeding as in §III-D and thanks to an experimental observation similar to Remark 4, we obtain that if

$$\frac{1}{2} r(r+2)\left(r(r+2)+5\right) > 2r(q+1) + t - 2,$$

then there exists an interval $[b_-, b_+]$ such that for $\mathcal{I}$ such that $|\mathcal{I}| \subseteq [b_-, b_+]$, the Schur product $\mathcal{S}_{\mathcal{I}}(\mathscr{C}_0(s)) \star \mathcal{S}_{\mathcal{I}}(\mathscr{C}_0(s+1))$ has a non generic behavior. Moreover,

1) $b_- = n - 2r(q+1) - t$;

2) b_+ is the largest integer such that

$$\frac{1}{2} d(d+5) > 3(n-1-b_+) - 4r(q+1) - 2t + 1,$$

where

$$d = (n-1-b_+) - 2r(q+1) - 2s + r(r+2).$$

Remark 5. From now on, in both situations (t even or odd), the corresponding interval will be referred to as the *distinguisher interval* for $\mathscr{C}_0(t)$.

E. A theoretical result on the multiplicative structure of the filtration

As explained previously, (22) does not hold in general even for the $\mathscr{C}_0(j)$'s even for shortenings at a set $\mathcal{I}$ such that $|\mathcal{I}|$ belongs to the distinguisher interval. However, we have the following Theorem. We explain in the sequel (see §IV-F) how to apply it practically. To avoid a huge amount of notation in its IEEEproof, we state it under a condition on the length of the $\mathscr{C}_0(j)$'s. It can then be applied in the general case to suitable shortenings of these codes.

Theorem 21. *Let $t > 1$ be an integer and assume that $n < 2r(q+1) + t$. Then,*

$$\mathscr{C}_0(t) = \left\{ \boldsymbol{c} \in \mathscr{C}_0(t-1) \mid \boldsymbol{c} \star \mathscr{C}_0(0) \subseteq \mathscr{A}_{2r(q+1)+t-n}(\boldsymbol{x}_0, \boldsymbol{y}_0) \right\}$$

where $\boldsymbol{y}_0 \stackrel{def}{=} \gamma^{-(2q+2)}(\boldsymbol{x}_0) \star \pi'_{\boldsymbol{x}_0}(\boldsymbol{x}_0)^{-1} \star \boldsymbol{x}_0^{-(t-2)}$.

Proof: Inclusion $\subseteq$ is a consequence of Corollary 20. Conversely, let $\boldsymbol{c} \in \mathscr{C}_0(t-1)$ be such that

$$\boldsymbol{c} \star \mathscr{C}_0(0) \subseteq \mathscr{A}_{2r(q+1)+t-n}(\boldsymbol{x}_0, \boldsymbol{y}_0). \tag{25}$$

Choose also an element $\boldsymbol{c}' \in \mathscr{C}_0(0) \backslash \mathscr{C}_0(1)$. From the definition of the $\mathscr{C}_0(j)$'s (Definition 5), these two codewords are of the form:

$$\begin{aligned} \boldsymbol{c} &= \gamma(\boldsymbol{x}_0)^{q+1} \star \boldsymbol{x}_0^{t-1} \star \pi'_{\boldsymbol{x}}(\boldsymbol{x}_0)^{-1} \star f(\boldsymbol{x}_0) \\ \boldsymbol{c}' &= \gamma(\boldsymbol{x}_0)^{q+1} \star \pi'_{\boldsymbol{x}}(\boldsymbol{x}_0)^{-1} \star g(\boldsymbol{x}_0), \end{aligned}$$

where

$$\deg(f) \leqslant n-r(q+1)-t \text{ and } \deg(g) \leqslant n-r(q+1)-1 \quad (26)$$

whereas and g does not vanish at 0. From Lemma 25 in Appendix A, we have $\pi'_{\boldsymbol{x}}(\boldsymbol{x}_0) = \boldsymbol{x}_0 \star \pi'_{\boldsymbol{x}_0}(\boldsymbol{x}_0)$ and hence,

$$\begin{aligned} \boldsymbol{c} \star \boldsymbol{c}' &= \gamma(\boldsymbol{x}_0)^{2(q+1)} \star \boldsymbol{x}_0^{t-3} \star \pi'_{\boldsymbol{x}_0}(\boldsymbol{x}_0)^{-2} \star f(\boldsymbol{x}_0) \star g(\boldsymbol{x}_0) \\ &= \boldsymbol{y}_0^{-1} \star \pi'_{\boldsymbol{x}_0}(\boldsymbol{x}_0)^{-1} \star \boldsymbol{x}_0^{-1} \star f(\boldsymbol{x}_0) \star g(\boldsymbol{x}_0). \end{aligned}$$

By (25), $\boldsymbol{c} \star \boldsymbol{c}'$ belongs to $\mathscr{A}_{2r(q+1)+t-n}(\boldsymbol{x}_0, \boldsymbol{y}_0)$. Using the description of alternant codes as evaluation codes given in Lemma 6, it can therefore be written as

$$\boldsymbol{c} \star \boldsymbol{c}' = \boldsymbol{y}_0^{-1} \star \pi'_{\boldsymbol{x}_0}(\boldsymbol{x}_0)^{-1} \star h(\boldsymbol{x}_0), \quad (27)$$

where

$$\deg(h) < n-1-2r(q+1)-t+n = 2n-1-2r(q+1)-t. \quad (28)$$

Putting (**??**) and (27) together, we get the vector equality $\boldsymbol{x}_0^{-1} \star f(\boldsymbol{x}_0) \star g(\boldsymbol{x}_0) = h(\boldsymbol{x}_0)$. Or, equivalently

$$\forall i \in \{1, \ldots, n-1\}, \ f(x_i)g(x_i) = x_i h(x_i).$$

From (26) and (28), the polynomial $f(z)g(z) - zh(z)$ has degree at most $2n - 2r(q+1) - t - 1$. Moreover, it has $n-1$ roots and, we assumed that $n < 2r(q+1)+t$ which entails $2n-2r(q+1)-t-1 < n-1$. Therefore, $f(z)g(z)-zh(z)$ has more roots than its degree, which proves the equality $f(z)g(z) = zh(z)$. Since by assumption, g does not vanish at zero, then z divides f, which entails that $\boldsymbol{c} \in \mathscr{C}_0(t)$. ∎

1) How to use this theorem?: Theorem 21 cannot be used directly in the cryptographic context for two reasons.

1) In general the inequality $n < 2r(q+1)+t$ is not satisfied by $\mathscr{C}$.
2) The alternant code $\mathscr{A}_{2r(q+1)+t-n}(\boldsymbol{x}_0, \boldsymbol{y}_0)$ is unknown.

Issue (1) is addressed by choosing suitable shortenings of the code. To address issue (2), despite $\mathscr{A}_{2r(q+1)+t-n}(\boldsymbol{x}_0, \boldsymbol{y}_0)$ is unknown, we know a possibly large subcode of it, namely

$$\mathscr{C}_0(\lfloor t/2 \rfloor) \star \mathscr{C}_0(\lceil t/2 \rceil).$$

Therefore, to use this result, one shortens the codes $\mathscr{C}_0(\lfloor t/2 \rfloor)$ and $\mathscr{C}_0(\lceil t/2 \rceil)$ at some set $\mathcal{I} \subseteq \{1, \ldots, n\}$ such that $|\mathcal{I}|$ lies in the distinguisher interval for computing $\mathscr{C}_0(t)$ defined in §IV-D. In this context, one can compute the subcode of $\mathcal{S}_{\mathcal{I}}(\mathscr{C}_0(t))$ defined as the set of elements $\boldsymbol{c}$ in $\mathcal{S}_{\mathcal{I}}(\mathscr{C}_0(t-1))$ that are such that

$$\boldsymbol{c} \star \mathcal{S}_{\mathcal{I}}(\mathscr{C}_0(0)) \subseteq \mathcal{S}_{\mathcal{I}}(\mathscr{C}_0(\lfloor t/2 \rfloor)) \star \mathcal{S}_{\mathcal{I}}(\mathscr{C}_0(\lceil t/2 \rceil)).$$

In all our experiments, this subcode turned out to be the whole $\mathcal{S}_{\mathcal{I}}(\mathscr{C}_0(t))$.

F. The algorithm

One expects to find $\mathcal{S}_{\mathcal{I}}(\mathscr{C}_0(t))$ by solving Problem 1. This allows to find several of these $\mathcal{S}_{\mathcal{I}}(\mathscr{C}_0(t))$'s associated to different subsets of indexes $\mathcal{I}$. It is straightforward to use such sets in order to recover $\mathscr{C}_0(t)$. Indeed, we clearly expect that

$$\mathcal{S}_{\mathcal{I} \cap \mathcal{J}}(\mathscr{C}_0(t)) = \mathcal{S}_{\mathcal{I}}(\mathscr{C}_0(t)) + \mathcal{S}_{\mathcal{J}}(\mathscr{C}_0(t)) \quad (29)$$

where with an abuse of notation we mean by $\mathcal{S}_{\mathcal{I}}(\mathscr{C}_0(t))$ and $\mathcal{S}_{\mathcal{J}}(\mathscr{C}_0(t))$ the codes $\mathcal{S}_{\mathcal{I}}(\mathscr{C}_0(t))$ and $\mathcal{S}_{\mathcal{J}}(\mathscr{C}_0(t))$ whose set of positions have been completed such as to also contain the positions belonging to $\mathcal{I} \setminus \mathcal{J}$ and $\mathcal{J} \setminus \mathcal{I}$ respectively and which are set to 0. Such an equality does not always hold of course, but apart from rather pathological cases it typically holds when $\dim(\mathcal{S}_{\mathcal{I}}(\mathscr{C}_0(t))) + \dim(\mathcal{S}_{\mathcal{I}}(\mathscr{C}_0(t))) \geqslant \dim(\mathcal{S}_{\mathcal{I} \cap \mathcal{J}}(\mathscr{C}_0(t)))$.

These considerations suggest Algorithm 1 for computing the codes $\mathscr{C}_0(s)$ for any $s > 1$. The value of $k(t)$ computed in line 3 for some $t > 2$ can also be obtained "offline" by computing the true dimension of a $\mathscr{C}_0(t)$ for an arbitrary choice of γ and $\boldsymbol{x}$. Algorithm 1 uses the knowledge of $\mathscr{C}_0(0)$ and $\mathscr{C}_0(1)$ (see Proposition 18). Observe that in line 5, the cardinality of $\mathcal{I}$ has to lie in the distinguisher interval as explained in Section IV-D. The instruction in line 10 should be understood as the addition of two codes having the "same" length where by abuse of notation, $\mathcal{S}_{\mathcal{I}}(\mathscr{C}_0(t))$ means the code $\mathcal{S}_{\mathcal{I}}(\mathscr{C}_0(t))$ to which 0's have been added in the positions belonging to $\mathcal{I}$.

Algorithm 1 Computation of $\mathscr{C}_0(s)$ with $s > 1$

1: **for** $t = 2$ to s **do**
2: $\quad \mathscr{C}_0(t) \leftarrow \{0\}$
3: $\quad k(t) \leftarrow (n-1) - 2r(q+1) - 2t + 2 + r(r+2)$
4: $\quad$ **while** $\dim \mathscr{C}_0(t) \neq k(t)$ **do**
5: $\quad\quad \mathcal{I} \leftarrow$ random subset of $\{1, \ldots, n-1\}$ such that $|\mathcal{I}| \in [b_-, b_+]$ {Section IV-D}
6: $\quad\quad \mathscr{A} \leftarrow \mathcal{S}_{\mathcal{I}}(\mathscr{C}_0(0))$
7: $\quad\quad \mathscr{B} \leftarrow \mathcal{S}_{\mathcal{I}}\left(\mathscr{C}_0(\lfloor \frac{t}{2} \rfloor)\right) \star \mathcal{S}_{\mathcal{I}}\left(\mathscr{C}_0(\lceil \frac{t}{2} \rceil)\right)$
8: $\quad\quad \mathscr{D} \leftarrow \mathcal{S}_{\mathcal{I}}(\mathscr{C}_0(t-1))$
9: $\quad\quad \mathcal{S}_{\mathcal{I}}(\mathscr{C}_0(t)) \leftarrow \mathscr{D} \cap \left(\mathscr{A} \star \mathscr{B}^{\perp}\right)^{\perp}$ {Solving of Problem 1}
10: $\quad\quad \mathscr{C}_0(t) \leftarrow \mathscr{C}_0(t) + \mathcal{S}_{\mathcal{I}}(\mathscr{C}_0(t))$
11: $\quad$ **end while**
12: **end for**
13: **return** $\mathscr{C}_0(s)$

V. An efficient Attack Using the Distinguisher

In this section, we sketch the complete attack we implemented. We chose to provide only a short description and to explain it in greater detail in Appendix E. We emphasize that the crucial aspects of the attack are the distinguisher and the computation of the filtration which are presented in §III and §IV. As soon as some terms of the filtration are computed, it is possible to derive some interesting information on the secret key. The attack we present here is one manner to recover the secret key and we insist on the fact that there might exist many other ways to recover it from the knowledge of some terms of the filtration. This is further discussed in Section VII-B.

Remind that we still stay in the context of Section III-C1 and $\mathscr{C}$ is the public key of a wild McEliece encryption scheme. Before describing the attack, we start with two key statements.

A. Key tools

The first statement is a very particular property of the space $\mathscr{C}_0(q+1)$. The fact that the $q+1$–th term has very particular properties is not surprising. Indeed, recall that in $\mathbb{F}_{q^2}$ the map $x \mapsto x^{q+1}$ is the norm over $\mathbb{F}_q$. In particular its sends $\mathbb{F}_{q^2}$ onto $\mathbb{F}_q$.

Proposition 22. *We have:*

$$\boldsymbol{x}_0^{-(q+1)} \star \mathscr{C}_0(q+1) \subseteq \mathscr{C}_0(0).$$

Proof: By definition $\boldsymbol{x}_0^{-(q+1)} \star \mathscr{C}_0(q+1)$ is the set of elements of the form

$$\begin{aligned} & \left(x_i^{-(q+1)} \frac{x_i^{q+1}\gamma^{q+1}(x_i)f(x_i)}{\pi'_{\boldsymbol{x}}(x_i)} \right)_{1 \leqslant i < n} \\ = & \left(\frac{\gamma^{q+1}(x_i)f(x_i)}{\pi'_{\boldsymbol{x}}(x_i)} \right)_{1 \leqslant i < n} \end{aligned}$$

such that

(i) $\left(\frac{x_i^{q+1}\gamma^{q+1}(x_i)f(x_i)}{\pi'_{\boldsymbol{x}}(x_i)} \right)_{1 \leqslant i < n}$ belongs to $\mathbb{F}_q^{n-1}$
(ii) $f \in \mathbb{F}_{q^2}[z]_{<n-(r+1)(q+1)}$.

Since $(q+1)$–th powers in $\mathbb{F}_{q^2}$ are norms over $\mathbb{F}_q$, we have $\boldsymbol{x}_0^{q+1} \in \mathbb{F}_q^{n-1}$. Therefore condition (i) is equivalent to asking for $\left(\frac{\gamma^{q+1}(x_i)f(x_i)}{\pi'_{\boldsymbol{x}}(x_i)} \right)_{1 \leqslant i < n}$ belonging to $\mathbb{F}_q^{n-1}$. By definition, $\mathscr{C}_0(0)$ is defined as the set

$$\left\{ \left(\frac{\gamma^{q+1}(x_i)f(x_i)}{\pi'_{\boldsymbol{x}}(x_i)} \right)_{1 \leqslant i < n} \;\middle|\; f \in \mathbb{F}_{q^2}[z]_{<n-r(q+1)} \right\} \cap \mathbb{F}_q^{n-1}.$$

These considerations imply that $\boldsymbol{x}_0^{-(q+1)} \star \mathscr{C}_0(q+1) \subseteq \mathscr{C}_0(0)$. ∎

The second statement asserts that the minimal polynomial over $\mathbb{F}_q$ of an element $t \in \mathbb{F}_{q^2}$ can be deduced from the single knowledge of the norms of t and $t-1$.

Lemma 23. *Let t be an element of $\mathbb{F}_{q^2}$ and*

$$P_t(z) \stackrel{def}{=} z^2 - (N(t) - N(t-1) - 1)z + N(t) \in \mathbb{F}_q[z].$$

Then, either $t \in \mathbb{F}_{q^2} \setminus \mathbb{F}_q$ and P_t is irreducible in which case is the minimal polynomial of t over $\mathbb{F}_q$, or $t \in \mathbb{F}_q$ and $P_t(z) = (z-t)^2$.

Proof: First, notice that

$$\begin{aligned} \mathrm{N}(t-1) &= (t-1)^{q+1} = (t-1)(t-1)^q = (t-1)(t^q-1) \\ &= t^{q+1} - t^q - t + 1 \\ &= \mathrm{N}_{\mathbb{F}_{q^2}/\mathbb{F}_q}(t) - \mathrm{Tr}_{\mathbb{F}_{q^2}/\mathbb{F}_q}(t) + 1. \end{aligned}$$

Therefore, $P_t(z) = z^2 - \mathrm{Tr}(t)z + \mathrm{N}(t)$, which is known to be the minimal polynomial of t whenever $t \in \mathbb{F}_{q^2} \setminus \mathbb{F}_q$. On the other hand, when $t \in \mathbb{F}_q$, then $P_t(z) = z^2 - 2tz + t^2$ which factorizes as $(z-t)^2$. ∎

B. Description of the attack

By the 2–transitivity of the affine group, one can assume without loss of generality that $x_0 = 0$ and $x_1 = 1$ (see Appendix A for further details).

Let us first assume that every element of $\mathbb{F}_{q^2}$ is an entry of $\boldsymbol{x}$. The general case: $n < q^2$, is discussed subsequently.

- **Step 1.** Compute $\mathscr{C}_0(q+1)$ using the method described in §IV. Notice that, thanks to Proposition 18(iv) it is sufficient to compute $\mathscr{C}_0(q-r)$.
- **Step 2.** Compute the set of solutions $\boldsymbol{c} \in \mathbb{F}_q^{n-1}$ of the problem

$$\begin{cases} \boldsymbol{c} \star \mathscr{C}_0(q+1) \subseteq \mathcal{P}_0(\mathscr{C}) \\ \\ \forall i \geqslant 1,\ c_i \neq 0, \quad (\text{i.e. } \boldsymbol{c} \text{ has full weight}) \\ \\ c_1 = 1. \end{cases} \tag{30}$$

  From Proposition 22, $\boldsymbol{x}_0^{-(q+1)}$ is one of the solutions of this problem. Indeed, it clearly satisfies the first equation, has full weight (0 has been removed) and its first entry is 1 since we assumed that $x_1 = 1$. One proves in Appendix F, that the space of words $\boldsymbol{c} \in \mathbb{F}_q^n$ such that $\boldsymbol{c} \star \mathscr{C}_0(q+1) \subseteq \mathcal{P}_0(\mathscr{C})$ has in general dimension 4 over $\mathbb{F}_q$. Moreover, with a high probability, this space has only 2 elements with full weight, namely, the vector $\boldsymbol{x}_0^{-(q+1)}$ (which has clearly full weight) and the all–one vector $\mathbf{1}$.

After these two steps, we know $\boldsymbol{x}^{q+1}$ which is unsufficient to deduce directly $\boldsymbol{x}$. However, we can re–apply Steps 1 and 2 replacing position 0 by position 1. By this manner, we compute $\mathscr{C}_1(q+1)$ and then solve a problem of the same form as (30) which yields $(\boldsymbol{x}-\mathbf{1})^{q+1}$.

- **Step 3.** Apply Lemma 23 to get the minimal polynomial of every entry x_i of the support $\boldsymbol{x}$. Now, the support is known up to Galois action.
- **Step 4.** One chooses an arbitrary support $\boldsymbol{x}'$ such that for all i, x_i and x'_i have the same minimal polynomial. That is, for all i either $x'_i = x_i$ or $x'_i = x_i^q$. Since $\mathscr{C} = \boldsymbol{u} \star \mathscr{A}_{r(q+1)}(\boldsymbol{x}, \mathbf{1})$ (Theorem 14), for some $\boldsymbol{u} \in (\mathbb{F}_q^\times)^n$, there exist a diagonal matrix $\boldsymbol{D}$ and a permutation matrix $\boldsymbol{P}$ such that

$$\mathscr{C} = \mathscr{A}_{r(q+1)}(\boldsymbol{x}, \mathbf{1})\boldsymbol{D}\boldsymbol{P}.$$

  The permutation $\boldsymbol{P}$ is the permutation that sends $\boldsymbol{x}$ onto $\boldsymbol{x}'$. Since it arises from Galois action, it is a product of transpositions with disjoint supports and the supports are known. Therefore, the matrix $\boldsymbol{D}\boldsymbol{P}$ is sparse and we know precisely the positions of the possible nonzero entries. The number of these unknown entries is $\approx 2q^2$ and the linear problem

$$\mathscr{C} = \mathscr{A}_{r(q+1)}(\boldsymbol{x}, \mathbf{1})\boldsymbol{M}$$

  whose unknowns the possible nonzero entries of $\boldsymbol{M}$ has more equations than unknowns and provide easily the matrix $\boldsymbol{D}\boldsymbol{P}$. From them we recover $\boldsymbol{x}$ and we have

$$\mathscr{C} = \mathscr{A}_{r(q+1)}(\boldsymbol{x}, \mathbf{1})\boldsymbol{D}.$$

  This concludes the attack.

The general case: When the support is not full, the main difficulty is that the resolution of (30) provides q^2-n other full-weight solutions. Thus we have q^2-n+1 candidates for $\boldsymbol{x}^{q+1}$ and q^2-n+1 for $(\boldsymbol{x}-\boldsymbol{1})^{q+1}$. A method is explained in Appendix F3 which permits to gather candidates by pairs $(\boldsymbol{a},\boldsymbol{b})$ where $\boldsymbol{a}$ is a candidate for $\boldsymbol{x}^{q+1}$ and $\boldsymbol{b}$ a candidate for $(\boldsymbol{x}-\boldsymbol{1})^{q+1}$. The good pair $(\boldsymbol{x}^{q+1},(\boldsymbol{x}-\boldsymbol{1})^{q+1})$ lies among these pairs.

Therefore, we have to iterate Steps 3 and 4 for every pair of candidates, which amounts to q^2-n iterations in the worst case. Step 4 is also a bit more complicated in the worst case but this has no influence on the complexity.

Remark 6. Notice that the computation of the Goppa polynomial is useless to attack the scheme. Actually, if the secret key is a wild Goppa code $\mathscr{G}\left(\boldsymbol{x},\gamma^{q-1}\right)$, then it is sufficient to find a pair of vectors $(\boldsymbol{x},\boldsymbol{y})$ such that:

$$\mathscr{G}\left(\boldsymbol{x},\gamma^{q-1}\right)=\mathscr{A}_{rq}(\boldsymbol{x},\boldsymbol{y}).$$

Indeed, such a representation as an alternant code allows to correct up to $\lfloor\frac{qr}{2}\rfloor$ errors (see Fact 1 and Theorem 8).

VI. Complexity

In what follows, by "$O(P(n))$" for some function $P:\mathbb{N}\to\mathbb{R}$, we mean "$O(P(n))$ operations in $\mathbb{F}_q$". We clearly have $n\leqslant q^2$ and we also assume that $q=O(\sqrt{n})$.

A. Computation of a code product

Given two codes $\mathscr{A},\mathscr{B}$ of length n and respective dimensions a and b, the computation of $\mathscr{A}\star\mathscr{B}$ consists first in the computation of a generator matrix of size $ab\times n$ whose computation costs $O(nab)$ operations. Then the Gaussian elimination costs $O(nab\min(n,ab))$. Thus the cost of Gaussian elimination dominates that of the construction step. In particular, for a code $\mathscr{A}$ of dimension $k\geqslant\sqrt{n}$, the computation of $\mathscr{A}^{\star 2}$ costs $O(n^2k^2)$. Thanks to Proposition 16, one shows that the dominant part of the resolution of Problem 1, consists in computing $\mathscr{A}\star\mathscr{B}^{\perp}$ and hence costs $O(na(n-b)\min(n,a(n-b)))$

B. Computation of the filtration

We first evaluate the cost of computing $\mathcal{S}_{\mathcal{I}}\left(\mathscr{C}_0(s+1)\right)$ from $\mathcal{S}_{\mathcal{I}}\left(\mathscr{C}_0(s)\right)$. The distinguisher interval described in §IV-D suggests that the dimension of $\mathcal{S}_{\mathcal{I}}\left(\mathscr{C}_0(s)\right)$ used to compute the filtration is in $O(\sqrt{n})$. From §VI-A, the computation of the square of $\mathcal{S}_{\mathcal{I}}\left(\mathscr{C}_0(s)\right)$ costs $O(n^3)$ operations in $\mathbb{F}_q$. Then, the resolution of Problem 16 in the context of Theorem 21, costs $O(na(n-b)\min(n,a(n-b)))$ where $a=\dim\mathcal{S}_{\mathcal{I}}\left(\mathscr{C}_0(s)\right)=O(\sqrt{n})$ and $b=\dim\mathscr{A}_{2r(q+1)+t-n+|\mathcal{I}|}(\boldsymbol{x}_{\mathcal{I}\cup\{0\}},\boldsymbol{y})$. We have $n-b=O(n)$, hence we get a cost of $O(n^3\sqrt{n})$.

The heuristic below Proposition 16 suggests that we need to perform this computation for $O(\sqrt{n})$ choices of $\mathcal{I}$. Since addition of codes is negligible compared to $O(n^3\sqrt{n})$ this leads to a total cost of $O(n^4)$ for the computation of $\mathscr{C}_0(s+1)$. This computation should be done $q+1$ times (actually $q-r$ times from Proposition 18(iv) and, we assumed that $q=O(\sqrt{n})$. Thus, the computation of $\mathscr{C}_0(q+1)$ costs $O(n^4\sqrt{n})$.

Remark 7. Actually, it is not necessary to compute all the terms of the filtration from $\mathscr{C}_0(1)$ to $\mathscr{C}_0(q-r)$, only $\log(q-r)$ of them are sufficient to get $\mathscr{C}_0(q-r)$ since $\mathscr{C}_0(q-r)$ is computed from $\mathscr{C}_0((q-r)/2)$. This reduces the complexity of this part to $O(n^4\log(n))$.

C. Other computations

The resolution of Problem (30) in Step 2, costs $O(n^4)$ (see Appendix E for further details). The solution space $\mathscr{D}$ of (30) has $\mathbb{F}_q$–dimension 4 (see Proposition 33 in Appendix E). Moreover, since we are looking for vectors of maximum weight in these solution spaces, it is sufficient to proceed to the search in the corresponding 3–dimensional projective spaces. Thus, the exhaustive search in these solution spaces costs $O(q^3)=O(n\sqrt{n})$ which is negligible. The computation of the pairs (see §F3) and that of minimal polynomials is also negligible. Finally, the resolution of the linear system in Step 4 costs $O(n^4)$ since it is very similar to Problem 1. Since the final step should be iterated q^2-n+1 times in the worst case, we see that the part of the attack after the computation of the filtration costs at worst $O(n^5)$. Thus, the global complexity of the attack is in $O(n^5)$ operations in $\mathbb{F}_q$.

D. Shortcuts

It is actually possible to reduce the complexity. Indeed, many linear systems we have to solve have b equations and a unknowns with $b\gg a$. For such systems it is possible to extract just slightly more that a equations chosen at random and solve this subsystem which has the same solution set with high probability. This probabilistic shortcut permits the computation of the square of a code in $O(n^3)$ and reduces the cost of Step 4 to $O(n^3)$. By this manner we have an overall complexity of $O(n^4)$.

VII. Main result, heuristics and extensions of the attack

A. The scope and limits of the attack

We are now able to state the main result which is partly proved mathematically and partly heuristic. In the following, we briefly list the intermediary results which are not proved but justified by heuristics.

Heuristic 1. *Let $\gamma\in\mathbb{F}_{q^2}[x]$ be an irreducible polynomial of degree r and $\boldsymbol{x}$ be an n–tuple of distinct elements of $\mathbb{F}_{q^2}$. Let $\mathscr{C}$ be the Goppa code $\mathscr{G}\left(\boldsymbol{x},\gamma^{q-1}\right)$ used as a public key for the McEliece encryption scheme, then if*

$$n>2q+4\quad\text{and}\quad\binom{r(r+2)+2}{2}>2r(q+1)+(q-r)-2,$$

there is a deterministic key-recovery attack of the scheme in $O(n^5)$ operations and a probabilistic one in $O(n^4)$.

Indeed, according to Lemma 35 and further discussions in Appendix E, the success of the Steps 2 to 4 only requires:

$$n>2q+4.$$

On the other hand, for the first step to work, the distinguisher intervals for the computation of $\mathscr{C}_0(2)$ up to $\mathscr{C}_0(q-r)$ should be non empty, i.e. from §IV-D1,

$$\binom{r(r+2)+2}{2} > 2r(q+1)+(q-r)-2.$$

Finally, the complexity of the attack is discussed in Section VI.

B. Extension of the attack

Actually, having a non empty distinguisher interval for the code seems sufficient to proceed to an attack, even if there is no distinguisher interval for the computation of $\mathscr{C}_0(q-r)$. Indeed, it is also possible to compute the "negative part" of the filtration, i.e. the codes $\mathscr{C}_0(-\ell)$'s for $\ell > 0$. The code $\mathscr{C}_0(-\ell)$ can be computed as

$$\left\{ \boldsymbol{c} \in \mathbb{F}_q^n \mid \boldsymbol{c} \star \mathscr{C}_0(0) \subseteq \mathscr{C}_0(\lfloor -\ell/2 \rfloor) \star \mathscr{C}_0(\lceil -\ell/2 \rceil) \right\}$$

or, if the Schur products fill in the ambient space, several suitable shortenings of $\mathscr{C}_0(-\ell)$ can be computed by this manner and then summed up to provide the whole $\mathscr{C}_0(-\ell)$. By this manner, as soon as we are able to compute two codes $\mathscr{C}_0(a)$ and $\mathscr{C}_0(b)$ such that $b-a = q+1$ then a statement of the form of Proposition 22 provides $\boldsymbol{x}_0^{q+1}$. This is for instance what we did for the $[851, 619]$ code over $\mathbb{F}_{32}$ presented in Section VIII. For this code it was not possible to compute $\mathscr{C}_0(q+1)$, thus we computed $\mathscr{C}_0(23)$ and $\mathscr{C}_0(-10)$.

As a conclusion, using a variant by computing some $\mathscr{C}_0(-\ell)$ we get the following extended heuristic.

Heuristic 2. *Let $\mathscr{C}$ be as in Heuristic 1. If*

$$n > 2q+4 \qquad \text{and} \qquad \binom{r(r+2)+2}{2} > 2r(q+1)-2,$$

then, there is a deterministic key-recovery attack of the scheme in $O(n^5)$ operations and a probabilistic one in $O(n^4)$.

C. Heuristic arguments

A large part of the previous statements is mathematically proved in this article. However, for some parts we have not been able to provide a mathematical proof. Nevertheless, every unproved fact is either discussed, justified by heuristics or confirmed by experimental results. Here we list the heuristics appearing along the text.

- **On the distinguisher interval**, Theorem 15 is proved but our experimental observations (Section III-E) show that conditions of Theorem 15 for the distinguisher interval to be non empty can be a little bit relaxed. We have no mathematical explanation of this small improvement.
- In Section IV-E, we give a proof of Theorem 21 but our attack relies on the assumption that (see Section IV-E1)

$$\mathcal{S}_{\mathcal{I}}\left(\mathscr{C}_0(\lfloor t/2 \rfloor)\right) \star \mathcal{S}_{\mathcal{I}}\left(\mathscr{C}_0(\lceil t/2 \rceil)\right) = \mathcal{S}_{\mathcal{I}}\left(\mathscr{C}_0(t)\right),$$

  which is confirmed by our experiments.
- In Section IV-F, we assume (see (29)) that the sum of two shortened alternant codes on set of indexes $\mathcal{I}, \mathcal{J}$ is the shortening at $\mathcal{I} \cap \mathcal{J}$. This is justified by a heuristic and confirmed by our experiments.
- In Appendix E, Proposition 32 is proved but the equality case is assumed to be true in the attack. This equality is discussed after the proof of the proposition. This holds also for the equality in the non full support case of Proposition 33.
- The final step of the attack G finishes by solving a large linear system whose solutions provide the support $\boldsymbol{x}$ and multiplier $\boldsymbol{y}$. Since this system has much more equations than unknowns we assume this system to have a very small space of solutions. This fact remains unproved but seems highly plausible. Moreover, in all our experiments the system had a linear space of solutions of dimension 2 (see Appendix G1 for further details).

VIII. Implementation

This attack has been implemented with Magma [43] and run over random examples of codes corresponding to the seven entries [19, Table 1] for which $m = 2$ and $r > 3$. For all these parameters, our attack succeeded. We summarize here the average running times for at least 50 random keys per 4–tuple of parameters, obtained with an Intel® Xeon 2.27GHz.

(q, n, k, r)	(29,781,516,5)	(29,791,575,4)	(29,794,529,5)
Aver. time	16min	19.5min	15.5min

(q, n, k, r)	(31, 795, 563, 4)	(31,813, 581,4)
Average time	31.5min	31.5min

(q, n, k, r)	(31, 851, 619, 4)	(32,841,601,4)
Average time	27.2min	49.5min

Remark 8. In the above table the code dimensions are not the ones mentioned in [19]. What happens here is that the formula for the dimension given [19, p.153,§1] is wrong for such cases: it understimates the true dimension for wild Goppa codes over quadratic extensions when the degree r of the irreducible polynomial γ is larger than 2 as shown by Theorem 14(ii).

All these parameters are given in [19] with a 128-bit security that is measured against information set decoding attack which is described in [19, p.151, Information set decoding §1] as the "*top threat against the wild McEliece cryptosystem for* $\mathbb{F}_3$, $\mathbb{F}_4$, *etc.*". It should be mentioned that these parameters are marked in [19] by the biohazard symbol ☣ (together with about two dozens other parameters). This corresponds, as explained in [19], to parameters for which the number of possible monic Goppa polynomials of the form γ^{q-1} is smaller than 2^{128}. The authors in [19] choose in this case a support which is significantly smaller than q^m (q^2 here) in order to avoid attacks that fix a support of size q^m and then enumerate all possible polynomials. Such attacks exploit the fact that two Goppa codes of length q^m with the same polynomial are *permutation equivalent*. We recall that the *support-splitting* algorithm [55], when applied to permutation equivalent codes, generally finds in polynomial time a permutation that sends one code onto the

other. The authors of [19] call this requirement on the length the *second defense* and write [19, p.152].

"*The strength of the second defense is unclear: we might be the first to ask whether the support-splitting idea can be generalized to handle many sets* $\{a_1, \ldots, a_n\}$ [1] *simultaneously, and we would not be surprised if the answer turns out to be yes.*" The authors also add in [19, p.154,§1] that "*the security of these cases*[2] *depends on the strength of the second defense discussed in Section 6*". We emphasize that our attack has nothing to do with the strength or a potential weakness of the second defense. Moreover, it does not exploit at all the fact that there are significantly less than 2^{128} Goppa polynomials. This is obvious from the way our attack works and this can also be verified by attacking parameters which were not proposed in [19] but for which there are more than 2^{128} monic wild Goppa polynomials to check. As an illustration, we are also able to recover the secret key in an average time of 24 minutes when the public key is a code over $\mathbb{F}_{31}$, of length 900 and with a Goppa polynomial of degree 14. In such case, the number of possible Goppa polynomials is larger than 2^{134} and according to Theorem 14, the public key has parameters $[n = 900, k \geqslant 228, d \geqslant 449]_{31}$. Note that security of such a key with respect to information set decoding [56] is also high (about 2^{125} for such parameters).

IX. Conclusion

The McEliece scheme based on Goppa codes has withstood all cryptanalytic attempts up to now, even if a related system based on GRS codes [2] was successfully attacked in [10]. Goppa codes are subfield subcodes of GRS codes and it was advocated that taking the subfield subcode hides a lot about the structure of the underlying code and also makes these codes more random-like. This is sustained by the fact that the distance distribution becomes indeed random [46] by this operation whereas GRS codes behave differently from random codes with respect to this criterion. This attack presented at the conference EUROCRYPT 2014 was the first example of a cryptanalysis which questions this belief by providing an algebraic cryptanalysis which is of polynomial complexity and which applies to many "reasonable parameters" of a McEliece scheme when the Goppa code is the $\mathbb{F}_q$-subfield subcode of a GRS code defined over $\mathbb{F}_{q^2}$. Subsequently to our attack, this uncertainty on the security of code based cryptosystems using wild Goppa codes has been strengthened by another cryptanalysis based on the resolution of a system of multivariate polynomial equations [44].

It could be argued that our attack applies to a rather restricted class of Goppa codes, namely wild Goppa codes of extension degree two. This class of codes also presents certain peculiarities as shown by Theorem 14 which were helpful for mounting an attack. However, it should be pointed out that the crucial ingredient which made this attack possible is the fact that such codes could be distinguished from random codes by square code considerations. A certain filtration of subcodes was indeed exhibited here and it turns out that shortened versions of these codes were related together by the star product. This allowed to reconstruct the filtration and from here the algebraic description of the Goppa code could be recovered. The crucial point here is really the existence of such a filtration whose elements are linked together by the star product. The fact that these codes were linked together by the star product is really related to the fact that the square code of certain shortened codes of the public code were of unusually low dimension which is precisely the fact that yielded the aforementioned distinguisher. This raises the issue whether other families of Goppa codes or alternant codes which can be distinguished from random codes by such square considerations [31] can be attacked by techniques of this kind. This covers high rate Goppa or alternant codes, but also other Goppa or alternant codes when the degree of extension is equal to 2. All of them can be distinguished from random codes by taking square codes of a shortened version of the dual code.

[1] $\{a_1, \ldots, a_n\}$ means here the support of the Goppa code.

[2] meaning here the cases marked with ☣.

References

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

[2] H. Niederreiter, "Knapsack-type cryptosystems and algebraic coding theory," *Problems of Control and Information Theory*, vol. 15, no. 2, pp. 159–166, 1986.

[3] T. P. Berger and P. Loidreau, "How to mask the structure of codes for a cryptographic use," *Des. Codes Cryptogr.*, vol. 35, no. 1, pp. 63–79, 2005.

[4] V. M. Sidelnikov, "A public-key cryptosytem based on Reed-Muller codes," *Discrete Math. Appl.*, vol. 4, no. 3, pp. 191–207, 1994.

[5] H. Janwa and O. Moreno, "McEliece public key cryptosystems using algebraic-geometric codes," *Des. Codes Cryptogr.*, vol. 8, no. 3, pp. 293–307, 1996.

[6] M. Baldi, M. Bodrato, and F. Chiaraluce, "A new analysis of the TMcEliece cryptosystem based on QC-LDPC codes," in *Proceedings of the 6th international conference on Security and Cryptography for Networks*, ser. SCN '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 246–262. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-85855-3_17

[7] R. Misoczki, J.-P. Tillich, N. Sendrier, and P. S. L. M. Barreto, "MDPC-McEliece: New McEliece variants from moderate density parity-check codes," in *Proc. IEEE Int. Symposium Inf. Theory - ISIT*, 2013, pp. 2069–2073.

[8] C. Löndahl and T. Johansson, "A new version of McEliece PKC based on convolutional codes," in *Information and Communications Security, ICICS*, ser. Lecture Notes in Comput. Sci., vol. 7168. Springer, 2012, pp. 461–470.

[9] D. Gligoroski, S. Samardjiska, H. Jacobsen, and S. Bezzateev, "McEliece in the world of Escher," IACR Cryptology ePrint Archive, Report2014/360, 2014, http://eprint.iacr.org/.

[10] V. M. Sidelnikov and S. Shestakov, "On the insecurity of cryptosystems based on generalized Reed-Solomon codes," *Discrete Math. Appl.*, vol. 1, no. 4, pp. 439–444, 1992.

[11] C. Wieschebrink, "Cryptanalysis of the Niederreiter public key scheme based on GRS subcodes," in *Post-Quantum Cryptography 2010*, ser. Lecture Notes in Comput. Sci., vol. 6061. Springer, 2010, pp. 61–72.

[12] L. Minder and A. Shokrollahi, "Cryptanalysis of the Sidelnikov cryptosystem," in *Advances in Cryptology - EUROCRYPT 2007*, ser. Lecture Notes in Comput. Sci., vol. 4515, Barcelona, Spain, 2007, pp. 347–360.

[13] C. Faure and L. Minder, "Cryptanalysis of the McEliece cryptosystem over hyperelliptic curves," in *Proceedings of the eleventh International Workshop on Algebraic and Combinatorial Coding Theory*, Pamporovo, Bulgaria, Jun. 2008, pp. 99–107.

[14] A. Otmani, J.-P. Tillich, and L. Dallot, "Cryptanalysis of two McEliece cryptosystems based on quasi-cyclic codes," *Special Issues of Mathematics in Computer Science*, vol. 3, no. 2, pp. 129–140, Jan. 2010.

[15] A. Couvreur, P. Gaborit, V. Gauthier-Umaña, A. Otmani, and J.-P. Tillich, "Distinguisher-based attacks on public-key cryptosystems using Reed-Solomon codes," *Des. Codes Cryptogr.*, vol. 73, no. 2, pp. 641–666, 2014. [Online]. Available: http://dx.doi.org/10.1007/s10623-014-9967-z

[16] G. Landais and J.-P. Tillich, "An efficient attack of a McEliece cryptosystem variant based on convolutional codes," in *Post-Quantum Cryptography'13*, ser. Lecture Notes in Comput. Sci., P. Gaborit, Ed., vol. 7932. Springer, Jun. 2013, pp. 102–117.

[17] A. Couvreur, I. Márquez-Corbella, and R. Pellikaan, "A polynomial time attack against algebraic geometry code based public key cryptosystems," in *Proc. IEEE Int. Symposium Inf. Theory - ISIT 2014*, Jun. 2014, pp. 1446–1450.

[18] A. Couvreur, A. Otmani, J. Tillich, and V. Gauthier-Umaña, "A polynomial-time attack on the BBCRS scheme," in *Public-Key Cryptography - PKC 2015*, ser. Lecture Notes in Comput. Sci., J. Katz, Ed., vol. 9020. Springer, 2015, pp. 175–193.

[19] D. J. Bernstein, T. Lange, and C. Peters, "Wild McEliece," in *Selected Areas in Cryptography*, ser. Lecture Notes in Comput. Sci., A. Biryukov, G. Gong, and D. Stinson, Eds., vol. 6544, 2010, pp. 143–158.

[20] ——, "Wild McEliece Incognito," in *Post-Quantum Cryptography 2011*, ser. Lecture Notes in Comput. Sci., B.-Y. Yang, Ed. Springer Berlin Heidelberg, 2011, vol. 7071, pp. 244–254. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-25405-5_16

[21] T. P. Berger, P.-L. Cayrel, P. Gaborit, and A. Otmani, "Reducing key length of the McEliece cryptosystem," in *Progress in Cryptology - AFRICACRYPT 2009*, ser. Lecture Notes in Comput. Sci., B. Preneel, Ed., vol. 5580, Gammarth, Tunisia, Jun. 21-25 2009, pp. 77–97.

[22] R. Misoczki and P. Barreto, "Compact McEliece keys from Goppa codes," in *Selected Areas in Cryptography*, Calgary, Canada, Aug. 13-14 2009.

[23] P. Barreto, R. Lindner, and R. Misoczki, "Monoidic codes in cryptography," in *Post-Quantum Cryptography 2011*, ser. Lecture Notes in Comput. Sci., vol. 7071. Springer, 2011, pp. 179–199.

[24] F. Levy-dit-Vehel and S. Litsyn, "Parameters of Goppa codes revisited," *IEEE Trans. Inform. Theory*, vol. 43, no. 6, pp. 1811–1819, Nov. 1997.

[25] J.-C. Faugère, A. Otmani, L. Perret, and J.-P. Tillich, "Algebraic cryptanalysis of McEliece variants with compact keys," in *Advances in Cryptology - EUROCRYPT 2010*, ser. Lecture Notes in Comput. Sci., vol. 6110, 2010, pp. 279–298.

[26] V. Gauthier-Umaña and G. Leander, "Practical key recovery attacks on two McEliece variants," 2009, iACR Cryptology ePrint Archive, Report2009/509.

[27] J.-C. Faugère, A. Otmani, L. Perret, F. de Portzamparc, and J.-P. Tillich, "Structural cryptanalysis of McEliece schemes with compact keys," *Des. Codes Cryptogr.*, vol. 79, no. 1, pp. 87–112, 2016. [Online]. Available: http://dx.doi.org/10.1007/s10623-015-0036-z

[28] ——, "Folding alternant and Goppa Codes with non-trivial automorphism groups," *IEEE Trans. Inform. Theory*, vol. 62, no. 1, pp. 184–198, 2016. [Online]. Available: http://dx.doi.org/10.1109/TIT.2015.2493539

[29] D. J. Bernstein, T. Lange, and C. Peters, "Smaller decoding exponents: ball-collision decoding," in *Advances in Cryptology - CRYPTO 2011*, ser. Lecture Notes in Comput. Sci., vol. 6841, 2011, pp. 743–760.

[30] J.-C. Faugère, V. Gauthier, A. Otmani, L. Perret, and J.-P. Tillich, "A distinguisher for high rate McEliece cryptosystems," in *Proc. IEEE Inf. Theory Workshop- ITW 2011*, Paraty, Brasil, Oct. 2011, pp. 282–286.

[31] ——, "A distinguisher for high rate McEliece cryptosystems," *IEEE Trans. Inform. Theory*, vol. 59, no. 10, pp. 6830–6844, Oct. 2013.

[32] I. Márquez-Corbella and R. Pellikaan, "Error-correcting pairs for a public-key cryptosystem," preprint, 2012, preprint.

[33] R. Pellikaan, "On decoding by error location and dependent sets of error positions," *Discrete Math.*, vol. 106–107, pp. 368–381, 1992.

[34] R. Kötter, "A unified description of an error locating procedure for linear codes," in *Proc. Algebraic and Combinatorial Coding Theory*, Voneshta Voda, 1992, pp. 113–117.

[35] I. Márquez-Corbella, E. Martínez-Moro, and R. Pellikaan, "Evaluation of public-key cryptosystems based on algebraic geometry codes," in *Proceedings of the Third International Castle Meeting on Coding Theory and Applications*, J. Borges and M. Villanueva, Eds., Barcelona, Spain, 2011, pp. 199–204.

[36] I. Márquez-Corbella, E. Martínez-Moro, and R. Pellikaan, "On the unique representation of very strong algebraic geometry codes," *Des. Codes Cryptogr.*, vol. 70, no. 1–2, pp. 215–230, 2014. [Online]. Available: http://dx.doi.org/10.1007/s10623-012-9758-3

[37] A. Couvreur, I. Márquez-Corbella, and R. Pellikaan, "Cryptanalysis of public-key cryptosystems that use subcodes of algebraic geometry codes," 2014, presented at the 4th International Castle Meeting on Coding Theory and Applications (4ICMCTA). Palmela (Portugal).

[38] I. Cascudo, H. Chen, R. Cramer, and C. Xing, "Asymptotically Good Ideal Linear Secret Sharing with Strong Multiplication over Any Fixed Finite Field," in *Advances in Cryptology - CRYPTO 2009*, ser. Lecture Notes in Comput. Sci., S. Halevi, Ed., vol. 5677. Springer Berlin Heidelberg, 2009, pp. 466–486.

[39] I. Cascudo, R. Cramer, and C. Xing, "The Torsion-Limit for Algebraic Function Fields and Its Application to Arithmetic Secret Sharing," in *Advances in Cryptology - CRYPTO 2011*, ser. Lecture Notes in Comput. Sci., P. Rogaway, Ed. Springer Berlin Heidelberg, 2011, vol. 6841, pp. 685–705.

[40] A. Bogdanov and C. Lee, "Homorphic encryption from codes," 2011, this paper was accepted for publication in the proceedings of the 44th ACM Symposium on Theory of Computing (STOC). The authors withdrew their paper after they learned that their scheme was threatened. It can be found on arXiv on `http://arxiv.org/abs/1111.4301`.

[41] M. Baldi, M. Bianchi, F. Chiaraluce, J. Rosenthal, and D. Schipani, "Enhanced public key security for the McEliece cryptosystem," submitted, 2011, arxiv:1108.2462v2[cs.IT].

[42] C. Wieschebrink, "Two NP-complete problems in coding theory with an application in code based cryptography," in *Proc. IEEE Int. Symposium Inf. Theory - ISIT*, 2006, pp. 1733–1737.

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

[44] J.-C. Faugère, L. Perret, and F. de Portzamparc, "Algebraic attack against variants of McEliece with Goppa polynomial of a special form," in *Advances in Cryptology - ASIACRYPT 2014*, ser. Lecture Notes in Comput. Sci., vol. 8873. Kaoshiung, Taiwan, R.O.C.: Springer, Dec. 2014, pp. 21–41.

[45] A. Couvreur, A. Otmani, and J.-P. Tillich, "Polynomial time attack on wild McEliece over quadratic extensions," in *Advances in Cryptology - EUROCRYPT 2014*, ser. Lecture Notes in Comput. Sci., P. Q. Nguyen and E. Oswald, Eds., vol. 8441. Springer Berlin Heidelberg, 2014, pp. 17–39. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-55220-5_2

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

[47] W. C. Huffman and V. Pless, *Fundamentals of error-correcting codes*. Cambridge University Press, Cambridge, 2003. [Online]. Available: http://dx.doi.org/10.1017/CBO9780511807077

[48] P. Delsarte, "On subfield subcodes of modified Reed-Solomon codes," *IEEE Trans. Inform. Theory*, vol. 21, no. 5, pp. 575–576, 1975.

[49] R. M. Roth, *Introduction to Coding Theory*. New York, NY, USA: Cambridge University Press, 2006.

[50] Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa, "Further results on Goppa codes and their applications to constructing efficient binary codes," *IEEE Trans. Inform. Theory*, vol. 22, pp. 518–526, 1976.

[51] I. Cascudo, R. Cramer, D. Mirandola, and G. Zémor, "Squares of random linear codes," *IEEE Trans. Inform. Theory*, vol. 61, no. 3, pp. 1159–1173, March 2015.

[52] H. Randriambololona, "Linear independence of rank 1 matrices and the dimension of *-products of codes," in *Proc. IEEE Int. Symposium Inf. Theory - ISIT*, Jun. 2015, pp. 196–200, `http://arxiv.org/abs/1501.05978`.

[53] I. Márquez-Corbella, E. Martínez-Moro, and R. Pellikaan, "The non-gap sequence of a subcode of a Generalized Reed–Solomon code," *Des. Codes Cryptogr.*, vol. 66, no. 1-3, pp. 317–333, 2012.

[54] A. Couvreur, A. Otmani, and J.-P. Tillich, "New identities relating wild Goppa codes," *Finite Fields Appl.*, vol. 29, pp. 178–197, 2014.

[55] N. Sendrier, "Finding the permutation between equivalent linear codes: The support splitting algorithm," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1193–1203, 2000.

[56] C. Peters, "Information-set decoding for linear codes over $\mathbf{F}_q$," in *Post-Quantum Cryptography 2010*, ser. Lecture Notes in Comput. Sci., vol. 6061. Springer, 2010, pp. 81–94.

[57] R. Lidl and H. Niederreiter, *Finite fields*, 2nd ed., ser. Encyclopedia of Mathematics and its Applications. Cambridge University Press, Cambridge, 1997, vol. 20, with a foreword by P. M. Cohn.

APPENDIX

REDUCING TO THE CASE $x_0 = 0$, $x_1 = 1$

The fact that we can choose x_0 to be equal to 0 and x_1 to be equal to 1 for the support $\boldsymbol{x}$ of $\mathscr{C}$ follows at once from the following lemma together with the 2-transitivity of the affine maps $x \mapsto ax + b$ over $\mathbb{F}_{q^m}$. This lemma is basically folklore,

but since we did not find a reference giving this lemma in exactly this form we have also provided a proof for it.

Lemma 24. *Consider a Goppa code $\mathscr{G}(\boldsymbol{x},\Gamma(x))$ defined over $\mathbb{F}_q$ and of extension degree m. Let $a,b\in\mathbb{F}_{q^m}$ with $a\neq 0$ and let $\psi(z)\stackrel{def}{=}az+b$. We have $\mathscr{G}(\boldsymbol{x},\Gamma(z))=\mathscr{G}\left(\psi(\boldsymbol{x}),\Gamma(\psi^{-1}(z))\right)$.*

Proof: We first observe that for any alternant code of some length n, degree r, extension degree m, defined over $\mathbb{F}_q$ we have

$$\mathscr{A}_r(\boldsymbol{x},\boldsymbol{y})=\mathscr{A}_r(a\boldsymbol{x}+b,\boldsymbol{y}). \tag{31}$$

This can be verified as follows. Let $\boldsymbol{c}=(c_i)_{0\leqslant i\leqslant n-1}$ be a codeword in $\mathscr{A}_r(\boldsymbol{x},\boldsymbol{y})$. We are going to prove that it also belongs to $\mathscr{A}_r(a\boldsymbol{x}+b,\boldsymbol{y})$. It suffices to prove that for any polynomial P in $\mathbb{F}_{q^m}[X]$ of degree at most $r-1$ we have $\sum_{i=0}^{n-1}c_iy_iP\left(ax_i+b\right)=0$. In order to prove this, let us first observe that we may write $P(ax+b)$ as a polynomial $Q(x)$ of degree at most $r-1$ which depends on a and b. This implies that

$$\sum_{i=0}^{n-1}c_iy_iP\left(ax_i+b\right)=\sum_{i=0}^{n-1}c_iy_iQ(x_i)=0.$$

where the last equality follows from the definition of $\mathscr{A}_r(\boldsymbol{x},\boldsymbol{y})$. In other words, we have just proved that $\boldsymbol{c}\in\mathscr{A}_r(a\boldsymbol{x}+b,\boldsymbol{y})$. This proves that

$$\mathscr{A}_r(\boldsymbol{x},\boldsymbol{y})\subseteq\mathscr{A}_r(a\boldsymbol{x}+b,\boldsymbol{y}). \tag{32}$$

The inclusion in the other direction by observing that by using (32) on $\mathscr{A}_r(a\boldsymbol{x}+b,\boldsymbol{y})$ with the affine map Ψ^{-1} we obtain

$$\mathscr{A}_r(a\boldsymbol{x}+b,\boldsymbol{y})\subseteq\mathscr{A}_r(\Psi^{-1}(a\boldsymbol{x}+b),\boldsymbol{y})=\mathscr{A}_r(\boldsymbol{x},\boldsymbol{y})$$

and this proves (31). This is used to finish the proof of Lemma 24 by observing that

$$\begin{aligned}\mathscr{G}(\boldsymbol{x},\Gamma(z)) &= \mathscr{A}_r(\boldsymbol{x},\boldsymbol{y})\\ &= \mathscr{A}_r(a\boldsymbol{x}+b,\boldsymbol{y})\\ &= \mathscr{G}\left(a\boldsymbol{x}+b,\Gamma(\psi^{-1}(z))\right)\end{aligned}$$

where $r=\deg\Gamma$ and $\boldsymbol{y}=\Gamma(\boldsymbol{x})^{-1}$. ∎

Further details on the behaviour of the filtration $\left(\mathscr{C}_0(s)\right)_{s\in\mathbb{Z}}$

The aim of this appendix is to give a complete proof of Proposition 18. For convenience, let us remind its statement.

Proposition 18 *Under Assumption 17 (i), we have*

(i) $\mathscr{C}_0(1)=\mathcal{S}_0(\mathscr{C})$;
(ii) $\mathscr{C}_0(0)=\mathcal{P}_0(\mathscr{C})$;
(iii) $\forall s\in\mathbb{Z},\ \dim\mathscr{C}_0(s)-\dim\mathscr{C}_0(s+1)\leqslant 2$;
(iv) $\mathscr{C}_0(q-r)=\mathscr{C}_0(q+1)$;
(v) $\forall s\in\mathbb{Z},\ \mathscr{C}_0(s)=\mathscr{A}_{r(q+1)+s-1}(\boldsymbol{x}_0,\boldsymbol{y}_0)$ *for*

$$\boldsymbol{y}_0\stackrel{def}{=}\gamma^{-(q+1)}(\boldsymbol{x}_0)\star\boldsymbol{x}_0^{-(s-1)}.$$

where we recall that $\boldsymbol{x}_0$ denotes the vector $\boldsymbol{x}$ punctured at position 0 and that r denotes the degree of γ.

The following lemma is useful in the proofs to follow.

Lemma 25. *For all $i\in\{1,\ldots,n-1\}$, we have*

$$\pi'_{\boldsymbol{x}}(x_i)=x_i\pi'_{\boldsymbol{x}_0}(x_i)$$

or, equivalently,

$$\pi'_{\boldsymbol{x}}(\boldsymbol{x}_0)=\boldsymbol{x}_0\star\pi'_{\boldsymbol{x}_0}(\boldsymbol{x}_0).$$

Proof: Recall that $x_0=0$ and therefore we have:

$$\pi_{\boldsymbol{x}}(z)=\prod_{i=0}^{n-1}(z-x_i)=z\prod_{i=1}^{n-1}(z-x_i)=z\pi_{\boldsymbol{x}_0}(z).$$

Therefore, $\pi'_{\boldsymbol{x}}(z)=z\pi'_{\boldsymbol{x}_0}(z)+\pi_{\boldsymbol{x}_0}(z)$ and since $\pi_{\boldsymbol{x}_0}(x_i)=0$ for all $i\in\{1,\ldots,n-1\}$, we get the result. ∎

A. proof of (v) and further results about the structure of the $\mathscr{C}_0(s)$'s

We will start by proving (v). Let $s\in\mathbb{Z}$. By definition $\mathscr{C}_0(s)$ is equal to the following set.

$$\left\{\left(\frac{\gamma^{q+1}(x_i)}{\pi'_{\boldsymbol{x}}(x_i)}x_i^sf(x_i)\right)_{1\leqslant i<n}\ \middle|\ f\in\mathbb{F}_{q^2}[z]_{<n-r(q+1)-s}\right\}\cap\mathbb{F}_q^{n-1}.$$

Then, from Lemma 25, we know that this set is equal to

$$\left\{\left(\frac{\gamma^{q+1}(x_i)x_i^{s-1}f(x_i)}{\pi'_{\boldsymbol{x}_0}(x_i)}\right)_{1\leqslant i<n}\ \middle|\ f\in\mathbb{F}_{q^2}[z]_{<(n-1)-r(q+1)-(s-1)}\right\}\cap\mathbb{F}_q^{n-1}.$$

The very definition of alternant codes (Definition 2) yields

$$\mathscr{C}_0(s)=\mathscr{A}_{r(q+1)+s-1}(\boldsymbol{x}_0,\boldsymbol{y}_0),$$

where

$$\boldsymbol{y}_0=\gamma^{-(q+1)}(\boldsymbol{x}_0)\star\boldsymbol{x}_0^{-(s-1)}.$$

It should be noted that the $\mathscr{C}_0(s)$'s can also be viewed as Goppa codes twisted by multiplying the positions by some fixed constants as explained by the following proposition which sheds some further light on the structure of the codes $\mathscr{C}_0(s)$:

Proposition 26. *Let $\boldsymbol{u}\stackrel{def}{=}\gamma^{q+1}(\boldsymbol{x}_0)\star\boldsymbol{x}_0^{-r(q+1)}$ where r denotes the degree of the polynomial γ such that $\mathscr{C}=\mathscr{G}\left(\boldsymbol{x},\gamma^{q+1}\right)$. Then $\boldsymbol{u}\in\mathbb{F}_q^{n-1}$ and for $s\geqslant -r(q+1)$:*

$$\mathscr{C}_0(s)=\boldsymbol{u}\star\mathscr{G}\left(\boldsymbol{x}_0,z^{r(q+1)+s-1}\right).$$

Proof: It has been discussed in § V-A, that $(q+1)$–th powers in $\mathbb{F}_{q^2}$ are norms and hence are elements of $\mathbb{F}_q$. Therefore, $\boldsymbol{u}\in\mathbb{F}_q^{n-1}$. Now, recall the definition of $\mathscr{C}_0(s)$ as

$$\left\{\left(\frac{\gamma^{q+1}(x_i)x_i^sf(x_i)}{\pi'_{\boldsymbol{x}_0}(x_i)}\right)_{1\leqslant i<n}\ \middle|\ f\in\mathbb{F}_{q^2}[z]_{<n-r(q+1)-s}\right\}\cap\mathbb{F}_q^{n-1}.$$

Since $\boldsymbol{u} = \gamma^{q+1}(\boldsymbol{x}_0) \star \boldsymbol{x}_0^{-r(q+1)}$ is a vector with entries in $\mathbb{F}_q$, it can get in the subfield subcode and $\boldsymbol{u}^{-1} \star \mathscr{C}_0(s)$ is nothing but

$$\left\{ \left(\frac{x_i^{r(q+1)+s-1} f(x_i)}{\pi'_{\boldsymbol{x}_0}(x_i)} \right)_{1 \leqslant i < n} \middle| \; f \in \mathbb{F}_{q^2}[z]_{<(n-1)-r(q+1)-s+1} \right\} \cap \mathbb{F}_q^{n-1}$$

and finally, thanks to the description of Goppa codes as evaluation codes in Lemma 7, the right hand term of the above equality is $\mathscr{G}\left(\boldsymbol{x}_0, z^{r(q+1)+s-1}\right)$, which concludes the proof. ■

B. Proof of (i)

From (v), applied to $s = 1$, we have

$$\mathscr{C}_0(1) = \mathscr{A}_{r(q+1)}(\boldsymbol{x}_0, \gamma^{-(q+1)}(\boldsymbol{x}_0)).$$

Thus, the very definition of Goppa codes (Definition 3) entails

$$\mathscr{C}_0(1) = \mathscr{G}\left(\boldsymbol{x}_0, \gamma^{q+1}\right).$$

Therefore, Corollary 10 on shortened Goppa codes asserts that $\mathscr{C}_0(1) = \mathcal{S}_0\left(\mathscr{C}\right)$.

C. Proof of (iii)

Let us bring in the family of GRS codes $\left(\mathscr{F}_s\right)_{s \in \mathbb{Z}}$ defined as the set

$$\left\{ \left(\frac{\gamma^{q+1}(x_i)}{\pi'_{\boldsymbol{x}_0}(x_i)} x_i^{s-1} f(x_i) \right)_{1 \leqslant i < n} \middle| \; f \in \mathbb{F}_{q^2}[z]_{<n-r(q+1)-s} \right\}. \tag{33}$$

We have

$$\forall s \in \mathbb{Z},\ \mathscr{C}_0(s) = \mathscr{F}_s \cap \mathbb{F}_q^{n-1}.$$

Moreover, it is readily seen that for all s, $\mathscr{F}_{s+1} \subseteq \mathscr{F}_s$ and $\dim \mathscr{F}_s - \dim \mathscr{F}_{s+1} \leqslant 1$, with equality if $\mathscr{F}_s$ is nonzero. Then, the proof of (iii) is a direct consequence of the following lemma.

Lemma 27. *Let $\mathscr{A}, \mathscr{B} \subseteq \mathbb{F}_{q^m}^n$ be two codes such that $\mathscr{A} \subseteq \mathscr{B}$. Then,*

$$\dim_{\mathbb{F}_q}(\mathscr{B} \cap \mathbb{F}_q^n) - \dim_{\mathbb{F}_q}(\mathscr{A} \cap \mathbb{F}_q^n) \leqslant m(\dim_{\mathbb{F}_{q^m}}(\mathscr{B}) - \dim_{\mathbb{F}_{q^m}}(\mathscr{A})).$$

Proof: Thanks to Delsarte's theorem (Theorem 2) it is equivalent to prove that

$$\dim_{\mathbb{F}_q} \mathrm{Tr}\left(\mathscr{A}^\perp\right) - \dim_{\mathbb{F}_q} \mathrm{Tr}\left(\mathscr{B}^\perp\right) \leqslant m(\dim_{\mathbb{F}_{q^m}} \mathscr{A}^\perp - \dim_{\mathbb{F}_{q^m}} \mathscr{B}^\perp).$$

To prove it, choose any direct summand $\mathscr{B}' \subseteq \mathbb{F}_{q^m}^n$ of $\mathscr{B}$ that is any code satisfying $\mathscr{A}^\perp = \mathscr{B}^\perp \oplus \mathscr{B}'$. Then, we clearly have

$$\dim_{\mathbb{F}_q} \mathrm{Tr}\left(\mathscr{A}^\perp\right) \leqslant \dim_{\mathbb{F}_q} \mathrm{Tr}\left(\mathscr{B}^\perp\right) + \dim_{\mathbb{F}_q} \mathrm{Tr}\left(\mathscr{B}'\right).$$

Finally, from [46, § 7.7], we get

$$\begin{aligned} \dim_{\mathbb{F}_q} \mathrm{Tr}(\mathscr{B}') &\leqslant m \dim_{\mathbb{F}_{q^m}} \mathscr{B}' \\ &\leqslant m\left(\dim_{\mathbb{F}_{q^m}} \mathscr{A}^\perp - \dim_{\mathbb{F}_{q^m}} \mathscr{B}^\perp\right), \end{aligned}$$

which yields the result. ■

D. Proof of (ii)

Statement (ii) is less obvious than it seems and far less obvious than (i). Indeed, let

$$\mathscr{F} \stackrel{\text{def}}{=} \left\{ \left(\frac{\gamma^{q+1}(x_i) f(x_i)}{\pi'_{\boldsymbol{x}}(x_i)} \right)_{0 \leqslant i < n} \middle| \; f \in \mathbb{F}_{q^2}[z]_{<n-r(q+1)} \right\}.$$

Using Lemma 25, one proves that $\mathscr{F}_0 = \mathcal{P}_0\left(\mathscr{F}\right)$ where $\mathscr{F}_0$ is defined in (33). Moreover, we have:

$$\mathscr{C} = \mathscr{F} \cap \mathbb{F}_q^n \quad \text{and} \quad \mathscr{C}_0(0) = \mathscr{F}_0 \cap \mathbb{F}_q^{n-1}.$$

Therefore,

$$\mathcal{P}_0\left(\mathscr{C}\right) = \mathcal{P}_0\left(\mathscr{F} \cap \mathbb{F}_q^n\right) \quad \text{while} \quad \mathscr{C}_0(0) = \mathcal{P}_0\left(\mathscr{F}\right) \cap \mathbb{F}_q^{n-1}.$$

Hence, from Proposition 3, we get

$$\mathcal{P}_0\left(\mathscr{C}\right) \subseteq \mathscr{C}_0(0) \tag{34}$$

and there is *a priori* no reason for the converse inclusion to be true. We will prove this by first observing that

Proposition 28.

$$\dim \mathscr{C}_0(0) - \dim \mathscr{C}_0(1) \geqslant 1. \tag{35}$$

Proof: First of all, notice that $\mathcal{P}_0\left(\mathscr{C}\right) = \mathscr{C}$ because $\mathscr{C}$ is of minimum distance > 1. Assumption 17 (i) tells us that $\mathcal{S}_0\left(\mathscr{C}\right) \neq \mathscr{C}$ and therefore $\mathcal{S}_0\left(\mathscr{C}\right)$ is strictly included in $\mathcal{P}_0\left(\mathscr{C}\right)$. In summary, thanks to (i) and (34), we have

$$\underbrace{\mathscr{C}_0(1)}_{=\mathcal{S}_0(\mathscr{C})} \subsetneq \mathcal{P}_0\left(\mathscr{C}\right) \subseteq \mathscr{C}_0(0) \tag{36}$$

and hence,

$$\dim \mathscr{C}_0(0) - \dim \mathscr{C}_0(1) \geqslant 1. \tag{37}$$

■

On the other hand, we can bound from above this difference of dimensions. This follows from

Proposition 29. *We have*

$$\dim \mathscr{C}_0(1) - \dim \mathscr{C}_0(-r) \leqslant 1.$$

Proof: From Proposition 26, we have

$$\mathscr{C}_0(1) = \boldsymbol{u} \star \mathscr{G}\left(\boldsymbol{x}_0, z^{r(q+1)}\right)$$

and

$$\begin{aligned} \mathscr{C}_0(-r) &= \boldsymbol{u} \star \mathscr{G}\left(\boldsymbol{x}_0, z^{r(q+1)-r-1}\right) \\ &= \boldsymbol{u} \star \mathscr{G}\left(\boldsymbol{x}_0, z^{rq-1}\right). \end{aligned}$$

From Theorem 8 and Remark 1, we have

$$\mathscr{G}\left(\boldsymbol{x}_0, z^{rq-1}\right) = \mathscr{G}\left(\boldsymbol{x}_0, z^{rq}\right).$$

In addition, from [54, Theorem 4], we have

$$\dim \mathscr{G}\left(\boldsymbol{x}_0, z^{rq}\right) - \dim \mathscr{G}\left(\boldsymbol{x}_0, z^{r(q+1)}\right) \leqslant 1.$$

This yields the result. ■

Conclusion: Putting inclusion sequence (36) in the filtration of the $\mathscr{C}_0(j)$'s, we get the inclusion sequence

$$\mathscr{C}_0(1) \subsetneq \mathcal{P}_0(\mathscr{C}) \subseteq \mathscr{C}_0(0) \subseteq \mathscr{C}_0(-1) \subseteq \cdots \subseteq \mathscr{C}_0(-r).$$

Using Proposition 29, we prove that, in the above inclusion sequence, every inclusion is an equality but the left-hand one. In particular,

$$\mathcal{P}_0(\mathscr{C}) = \mathscr{C}_0(0),$$

which concludes the proof of (ii).

Actually, we got other deep results namely, $\mathscr{C}_0(0) = \mathscr{C}_0(-r)$ and $\dim \mathscr{C}_0(0) - \dim \mathscr{C}_0(1) \leqslant 1$. Using Proposition 26, we obtain an interesting result on Goppa codes which clarifies [54, Theorem 4].

Corollary 30. *Let ℓ be a positive integer and $\boldsymbol{x} \in \mathbb{F}_{q^2}^n$ be a support, then:*
(i) $\mathscr{G}\left(\boldsymbol{x}, z^{\ell q-1}\right) = \mathscr{G}\left(\boldsymbol{x}, z^{\ell(q+1)-1}\right)$;
(ii) $\dim \mathscr{G}\left(\boldsymbol{x}, z^{\ell(q+1)-1}\right) - \dim \mathscr{G}\left(\boldsymbol{x}, z^{\ell(q+1)}\right) \leqslant 1.$

E. Proof of (iv)

Thanks to Proposition 26, it reduces to prove that

$$\mathscr{G}\left(\boldsymbol{x}_0, z^{(r+1)(q+1)-1}\right) = \mathscr{G}\left(\boldsymbol{x}_0, z^{(r+1)q-1}\right),$$

which is a direct consequence of Corollary 30(i) in the case $\ell = r+1$.

An in–depth presentation of the attack

Here we give a complete presentation of the attack in the general case, i.e. for a possibly non full support $\boldsymbol{x}$. As explained in §V, the attack divides into four steps:

- **Step 1.** Compute the terms of the filtrations $\left(\mathscr{C}_0(j)\right)_j$ and $\left(\mathscr{C}_1(j)\right)_j$ up to $\mathscr{C}_0(q+1)$ and $\mathscr{C}_1(q+1)$, using the methods presented in §IV.
- **Step 2.** Compute $\boldsymbol{x}^{q+1}$ and $(\boldsymbol{x}-\mathbf{1})^{q+1}$ thanks to Proposition 22.
- **Step 3.** Compute the minimal polynomials of every entry x_i of the support $\boldsymbol{x}$ using Lemma 23.
- **Step 4.** Compute a matrix $\boldsymbol{M}$ solution of the linear problem
$$\mathscr{C} = \mathscr{C}'\boldsymbol{M}$$
where $\boldsymbol{M}$ is a matrix with many prescribed zero entries and $\mathscr{C}' = \mathscr{A}_{r(q+1)}(\boldsymbol{x}, \mathbf{1})$ and obtain from $\boldsymbol{M}$ the whole structure of $\mathscr{C}'$.

Step 1 is explained in depth in §IV and Step 3 is straightforward (it is a direct application of Lemma 23). Thus, in this appendix, we give further details on Steps 2 and 4.

F. Further details on Step 2 of the attack

As explained in §V, the computation of $\boldsymbol{x}^{q+1}$ or, more precisely, that of $\boldsymbol{x}_0^{-(q+1)}$ reduces to solving Problem (30) which we recall here:

$$\begin{cases} \boldsymbol{c} \star \mathscr{C}_0(q+1) \subseteq \mathcal{P}_0(\mathscr{C}) \\ \\ \forall i \geqslant 1,\ c_i \neq 0, \quad \text{(i.e. } \boldsymbol{c} \text{ has full weight)} \\ \\ c_1 = 1. \end{cases} \tag{30}$$

Remind that, from Proposition 18(ii), we know that $\mathcal{P}_0(\mathscr{C}) = \mathscr{C}_0(0)$. Then, according to Proposition 16, the subspace of vectors $c \in \mathbb{F}_q^{n-1}$ such that $\boldsymbol{c} \star \mathscr{C}_0(q+1) \subseteq \mathcal{P}_0(\mathscr{C})$ is the space

$$\mathscr{D} \stackrel{\text{def}}{=} \left(\mathscr{C}_0(q+1) \star \mathscr{C}_0(0)^{\perp}\right)^{\perp}.$$

We will first investigate the structure of $\mathscr{D}$ and in particular its dimension. Then, we will study its set of full weight codewords. For this sake we will use repeatedly the following elementary lemma.

Lemma 31. *Let $\mathscr{A} \subseteq \mathbb{F}_q^n$ be a code and $\boldsymbol{u} \subseteq (\mathbb{F}_q^{\times})^n$, then*

$$(\boldsymbol{u} \star \mathscr{A})^{\perp} = \boldsymbol{u}^{-1} \star (\mathscr{A}^{\perp}).$$

Proof: Since $\boldsymbol{u}$ is invertible, then, clearly, both codes have the same dimension and it is sufficient to prove inclusion "$\supseteq$". Let $\boldsymbol{a} \in \mathscr{A}$ and $\boldsymbol{b} \in \mathscr{A}^{\perp}$, then

$$\langle \boldsymbol{u} \star \boldsymbol{a}, \boldsymbol{u}^{-1} \star \boldsymbol{b} \rangle = \sum_i u_i a_i u_i^{-1} b_i = \sum_i a_i b_i = \langle \boldsymbol{a}, \boldsymbol{b} \rangle = 0.$$

This concludes the proof. ∎

1) The structure of the code $\mathscr{D}$: We start with a rather technical statement which is fundamental in what follows.

Proposition 32. *We have*

$$\boldsymbol{x}_0^{-(q+1)} \star \left(\mathbf{RS}_{q+2}(\boldsymbol{x}_0) \cap \mathbb{F}_q^{n-1}\right) \subseteq \mathscr{D}.$$

Proof: First let us rewrite the codes $\mathscr{C}_0(0)$ and $\mathscr{C}_0(q+1)$ in a more convenient way. By definition $\mathscr{C}_0(q+1)$ is given by the set

$$\left\{ \left(\frac{\gamma^{q+1}(x_i)}{\pi'_{\boldsymbol{x}}(x_i)} x_i^{q+1} f(x_i) \right)_{1 \leqslant i < n} \middle| \ f \in \mathbb{F}_{q^2}[z]_{<n-(r+1)(q+1)} \right\} \cap \mathbb{F}_q^{n-1}.$$

In the very same way as in the proof of Proposition 26, since the $(q+1)$–th powers are norms and hence are in $\mathbb{F}_q$, they can get out of the subfield subcode which implies that $\mathscr{C}_0(q+1)$ is equal to the intersection of

$$\gamma^{q+1}(\boldsymbol{x}_0) \star \boldsymbol{x}_0^{q+1} \star \left\{ \left(\frac{1}{\pi'_{\boldsymbol{x}}(x_i)} f(x_i) \right)_{1 \leqslant i < n} \middle| \ f \in \mathbb{F}_{q^2}[z]_{<n-(r+1)(q+1)} \right\}$$

with $\mathbb{F}_q^{n-1}$. Since the codes have length $n-1$, it is more relevant to write $\mathbb{F}_{q^2}[z]_{<n-(r+1)(q+1)}$ as $\mathbb{F}_{q^2}[z]_{<(n-1)-(r+1)(q+1)+1}$. Then, thanks to Lemma 25, we get

$$\mathscr{C}_0(q+1) = \gamma^{q+1}(\boldsymbol{x}_0) \star \boldsymbol{x}_0^{q+1} \star \mathscr{A},$$

where $\mathscr{A}$ is the code

$$\left\{ \left(\frac{f(x_i)}{x_i \pi'_{\boldsymbol{x}_0}(x_i)} \right)_{1 \leqslant i < n} \middle| \ f \in \mathbb{F}_{q^2}[z]_{<(n-1)-(r+1)(q+1)+1} \right\} \cap \mathbb{F}_q^{n-1}.$$

Consequently, by the description of alternant codes as evaluation codes (Lemma 6), we obtain

$$\mathscr{C}_0(q+1) = \gamma^{q+1}(\boldsymbol{x}_0) \star \boldsymbol{x}_0^{q+1} \star \mathscr{A}_{(r+1)(q+1)-1}(\boldsymbol{x}_0, \boldsymbol{x}_0). \quad (38)$$

In the very same manner, we prove that

$$\mathscr{C}_0(0) = \gamma^{q+1}(\boldsymbol{x}_0) \star \mathscr{A}_{r(q+1)-1}(\boldsymbol{x}_0, \boldsymbol{x}_0). \quad (39)$$

From the definition of alternant codes (Definition 2) together with Delsarte's theorem (Theorem 2) and Lemma 31, we get

$$\mathscr{C}_0(0)^\perp = \gamma^{-(q+1)}(\boldsymbol{x}_0) \star \mathrm{Tr}\left(\mathbf{GRS}_{r(q+1)-1}(\boldsymbol{x}_0, \boldsymbol{x}_0)\right). \quad (40)$$

From (38) and (40),

$$\mathscr{C}_0(q+1) \star \mathscr{C}_0(0)^\perp = \boldsymbol{x}_0^{q+1} \star \mathscr{A}_{(r+1)(q+1)-1}(\boldsymbol{x}_0, \boldsymbol{x}_0) \star \mathrm{Tr}\left(\mathbf{GRS}_{r(q+1)-1}(\boldsymbol{x}_0, \boldsymbol{x}_0)\right).$$

Since the alternant code is defined over $\mathbb{F}_q$, it can get in the trace:

$$\mathscr{C}_0(q+1) \star \mathscr{C}_0(0)^\perp = \boldsymbol{x}_0^{q+1} \star \mathrm{Tr}\left(\mathscr{A}_{(r+1)(q+1)-1}(\boldsymbol{x}_0, \boldsymbol{x}_0) \star \mathbf{GRS}_{r(q+1)-1}(\boldsymbol{x}_0, \boldsymbol{x}_0)\right). \quad (41)$$

By definition of alternant codes (Definition 2) and by duality for GRS codes (Proposition 4),

$$\mathscr{A}_{(r+1)(q+1)-1}(\boldsymbol{x}_0, \boldsymbol{x}_0) = \mathbf{GRS}_{n-(r+1)(q+1)}(\boldsymbol{x}_0, \boldsymbol{x}_0^{-1} \star {\pi'_{\boldsymbol{x}_0}}^{-1}(\boldsymbol{x}_0)) \cap \mathbb{F}_q^{n-1}.$$

Therefore, since every code contains its subfield subcode,

$$\mathscr{A}_{(r+1)(q+1)-1}(\boldsymbol{x}_0, \boldsymbol{x}_0) \star \mathbf{GRS}_{r(q+1)-1}(\boldsymbol{x}_0, \boldsymbol{x}_0) \subseteq \mathbf{GRS}_{n-(r+1)(q+1)}(\boldsymbol{x}_0, \boldsymbol{x}_0^{-1} \star \pi'_{\boldsymbol{x}_0}(\boldsymbol{x}_0)^{-1}) \star \mathbf{GRS}_{r(q+1)-1}(\boldsymbol{x}_0, \boldsymbol{x}_0). \quad (42)$$

Thus, from Proposition 12(i) on products on GRS codes, we get

$$\mathscr{A}_{(r+1)(q+1)-1}(\boldsymbol{x}_0, \boldsymbol{x}_0) \star \mathbf{GRS}_{r(q+1)-1}(\boldsymbol{x}_0, \boldsymbol{x}_0) \subseteq \mathbf{GRS}_{(n-1)-(q+2)}(\boldsymbol{x}_0, \pi'_{\boldsymbol{x}_0}(\boldsymbol{x}_0)^{-1}). \quad (43)$$

Equations (41) and (43) yield

$$\mathscr{C}_0(q+1) \star \mathscr{C}_0(0)^\perp \subseteq \boldsymbol{x}_0^{q+1} \star \mathrm{Tr}\left(\mathbf{GRS}_{(n-1)-(q+2)}(\boldsymbol{x}_0, \pi'_{\boldsymbol{x}_0}(\boldsymbol{x}_0)^{-1})\right).$$

By dualizing and thanks to Lemma 31, to Delsarte Theorem (Theorem 2) and Proposition 4, we get

$$\mathscr{D} \supseteq \boldsymbol{x}_0^{-(q+1)} \star \left(\mathbf{RS}_{q+2}(\boldsymbol{x}_0) \cap \mathbb{F}_q^{n-1}\right).$$

■

a) **Discussion on the equality:** While Proposition 32 is only an inclusion, it turns out that in all our experiments, the inclusion was an equality. It is worth nothing that the reason why this equality typically holds is more or less the reason why our distinguisher works.

Indeed, for the equality to hold, (42) should be an equality. The right-hand product in (42) is a GRS code of dimension $(n-1)-(q+1)$ (see (43)), while the left hand one is a product of codes of respective (designed) dimensions $n-2(r+1)(q+1)$ and $r(q+1)-1$. From Proposition 11, the product of two random codes with these dimensions would be

$$\min\{n-1, (n-2(r+1)(q+1))(r(q+1)-1)\}.$$

For cryptographic sizes of parameters, the above min is $n-1$ and hence, with a very high probability, the left-hand product in (42) fills in the right-hand one. This explains, why the inclusion in Proposition 32 is almost always an equality.

Let us now investigate further the structure of the code $\mathbf{RS}_{q+2}(\boldsymbol{x}_0) \cap \mathbb{F}_q^{n-1}$.

Notation 1. Let α be a primitive element of $\mathbb{F}_{q^2}/\mathbb{F}_q$. In what follows we denote by $\mathscr{E}$ the following code which is used repeatedly

$$\mathscr{E} \stackrel{\text{def}}{=} \langle \mathbf{1}, \mathrm{Tr}(\boldsymbol{x}_0), \mathrm{Tr}(\alpha \boldsymbol{x}_0), N(\boldsymbol{x}_0) \rangle_{\mathbb{F}_q}.$$

Proposition 33. *We have,*

$$\mathscr{E} \subseteq \mathbf{RS}_{q+2}(\boldsymbol{x}_0) \cap \mathbb{F}_q^{n-1},$$

with equality when the support $\boldsymbol{x}$ is full.

Proof: We first prove the result under the assumption that $\boldsymbol{x}$ is full. Our goal is to describe the polynomials h in $\mathbb{F}_{q^2}[x]_{<q+2}$ satisfying

$$\forall x \in \mathbb{F}_{q^2}^\times,\ h(x) = h(x)^q.$$

Or equivalently,

$$h \equiv h^q \mod (x^{q^2-1} - 1). \quad (44)$$

Writing h as $h(x) = \sum_{i=1}^{q+1} h_i x^i$, Equation (44) yields the system

$$\begin{cases} h_0 = h_0^q \\ h_1 = h_q^q \\ h_{q+1} = h_{q+1}^q \\ h_i = 0, \quad \forall i \in \{2, \ldots, q-1\}. \end{cases} \quad (45)$$

Solving the above system yields the $\mathbb{F}_q$–basis of solutions: $1, x+x^q, \alpha x + \alpha^q x^q, x^{q+1}$, which concludes the proof. If $\boldsymbol{x}$ is non full then it is easy to see that the polynomials satisfying (45) provide words of $\mathbf{RS}_{q+1}(\boldsymbol{x}_0) \cap \mathbb{F}_q^{n-1}$ but there might exist other ones. ■

b) **Discussion on the non full–support case:** In all our experiments, the code $\mathbf{RS}_{q+1}(\boldsymbol{x}_0) \cap \mathbb{F}_q^n$ turned out to have dimension 4 even when the support is non full. This can be explained as follows. In terms of polynomials, the full support code is generated as the image of the $\mathbb{F}_q$–space of polynomials in $\mathbb{F}_{q^2}[z]_{<q+2}$ solution to the $\mathbb{F}_q$–linear system

$$\forall x_i \in \mathbb{F}_{q^2}^\times, \quad f(x_i)^q - f(x_i) = 0. \quad (46)$$

There are $q^2 - 1$ equations, while the $\mathbb{F}_q$–dimension of $\mathbb{F}_{q^2}[z]_{<q+2}$ is $2q+4$. The non full support case is obtained by removing equations in (46). Since this system is over-constrained, one can reasonably hope that removing some equations will have no incidence on the solution space as soon as $n > 2q+4$.

c) **Conclusion:** It is reasonable to hope — and this is exactly what happened in all our experiments (more than 600 tests) — that

$$\mathscr{D} = \boldsymbol{x}_0^{-(q+1)} \star \left(\mathbf{RS}_{q+1}(\boldsymbol{x}_0) \cap \mathbb{F}_q^{n-1}\right) = \boldsymbol{x}_0^{-(q+1)} \star \mathscr{E}.$$

2) The full weight codewords of $\mathscr{D}$: Since, the solution set of Problem (30) consists in full weight vectors $\boldsymbol{c}$ with $c_1 = 1$, it is sufficient to classify full weight vectors up to multiplication by a scalar. For this reason, in what follows, we will frequently consider vectors up to multiplication by a scalar. According to the previous discussions, one can assume that $\mathscr{D} = \boldsymbol{x}_0^{-(q+1)} \star \mathscr{E}$ and the study of full weight codewords of $\mathscr{D}$ reduces to that of $\mathscr{E}$.

Proposition 34. *Let*

$$U \stackrel{def}{=} \left\{ (\boldsymbol{x}_0 - a)^{q+1} \;\middle|\; a \in \mathbb{F}_{q^2} \setminus \{x_1, \ldots, x_{n-1}\} \right\}.$$

Then, the elements of U are full weight codewords of $\mathscr{E}$. Moreover, let $\mathbf{P}$ be the probability that every full weight codeword of $\mathscr{E}$ up to multiplication by a scalar is in U, then

$$\mathbf{P} \begin{cases} = 1, \text{ if } n \geqslant q^2 - q + 2 \\ \geqslant 1 - (q^3+q)\dfrac{\binom{q^2-q+1}{n-1}}{\binom{q^2}{n-1}}, \text{ else.} \end{cases}$$

Proof: Notice that the words $(\boldsymbol{x}_0 - a)^{q+1}$ for some $a \in \mathbb{F}_{q^2}$ are elements of $\mathscr{E}$. Indeed, expanding the word as

$$(\boldsymbol{x}_0 - a)^{q+1} = (\boldsymbol{x}_0 - a)^q(\boldsymbol{x}_0 - a) = \boldsymbol{x}_0^{q+1} - \mathrm{Tr}(a^q \boldsymbol{x}_0) + a^{q+1}.$$

Since a^q decomposes as $a_0 + \alpha a_1$, with $a_0, a_1 \in \mathbb{F}_q$, this provides a decomposition of $(\boldsymbol{x}_0 - a)^{q+1}$ as an $\mathbb{F}_q$–linear combination of the words $\mathbf{1}, \mathrm{Tr}(\boldsymbol{x}_0), \mathrm{Tr}(\alpha \boldsymbol{x}_0), \boldsymbol{x}_0^{q+1}$.

Let us investigate further the structure of the elements of $\mathscr{E}$. The codewords of $\mathscr{E}$ are given by evaluation of polynomials of the form

$$f(z) = \lambda_1 z^{q+1} + \lambda_2(z^q + z) + \lambda_3(\alpha^q z^q + \alpha z) + \lambda_4, \quad (47)$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ all belong to $\mathbb{F}_q$. Therefore, describing the full weight codewords of $\mathscr{E}$ reduces to understand which of these polynomials do not vanish at any entry of $\boldsymbol{x}_0$. Here, we can give a geometric interpretation of the set of roots in $\mathbb{F}_{q^2}$ of such a polynomial in terms of points of affine conics over $\mathbb{F}_q$. For that we proceed to a Weil descent. Namely, set $z = u + \alpha v$, where $u, v \in \mathbb{F}_q$. In addition, we choose $\alpha \in \mathbb{F}_{q^2} \setminus \mathbb{F}_q$ so that

$$\alpha^q = -\alpha, \quad \text{if } 2 \nmid q \quad or \quad \alpha^q = \alpha + 1 \quad \text{if } 2 \mid q.$$

Such an α always exists. Indeed,

- in odd characteristic, choose a non-square $d \in \mathbb{F}_q$ and let $\alpha \in \mathbb{F}_{q^2}$ be a square root of d.
- in even characteristic, choose $d \in \mathbb{F}_q$ such that $\mathrm{Tr}_{\mathbb{F}_q/\mathbb{F}_2}(d) \neq 0$, then the polynomial $z^2 + z + d$ is irreducible in $\mathbb{F}_q[z]$ and let α be one of its roots in $\mathbb{F}_{q^2}$.

Let us treat the odd characteristic case, the even characteristic can be treated in a very similar fashion. Set $x = u + \alpha v$, where $\alpha \in \mathbb{F}_{q^2}$ is a square root of a non-square $d \in \mathbb{F}_q$, then a simple computation from (47) transforms $f(x)$ as $\tilde{f}(u, v)$

$$\tilde{f}(u, v) = \lambda_1(u^2 - dv^2) + 2\lambda_2 u + 2d\lambda_3 v + \lambda_4, \quad (48)$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ all belong to $\mathbb{F}_q$. The set of pairs $(u, v) \in \mathbb{F}_q^2$ at which $\tilde{f}$ vanishes are in one-to-one correspondence with the set of zeros in $\mathbb{F}_{q^2}$ of f. Therefore, f provides a full-weight codeword in $\mathscr{E}$ if and only if f does not vanish on $\{x_1, \ldots, x_{n-1}\}$, that is if and only if the zero locus of $\tilde{f}$ in $\mathbb{F}_q^2$ is contained in

$$A \stackrel{\text{def}}{=} \left\{ (u, v) \;\middle|\; u + \alpha v \in \mathbb{F}_{q^2} \setminus \{x_1, \ldots, x_{n-1}\} \right\}.$$

Consequently, we need to understand the probability that A contains a conic whose equation is of the form (48). Let us analyze some particular cases of conics of the form (48).

(i) When $\lambda_1 = \lambda_2 = \lambda_3 = 0$. The conic is empty. In terms of codewords, it corresponds to multiples of the all-one word $\mathbf{1}$.

(ii) When, $\lambda_1 = 0$ the conic is nothing but an affine line. It has exactly q points over $\mathbb{F}_q$.

(iii) For $\lambda_1 \neq 0$. Since we consider words only up to multiplication by a scalar, one can assume that $\lambda_1 = 1$. Let us look for a criterion for the conic to be singular. Recall that a conic of equation $f(x, y)$ is said to be *singular* if $f, \frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$ have a common zero. The computation of the partial derivatives of $\tilde{f}$ yields:

$$\frac{\partial \tilde{f}}{\partial u} = 2u + 2\lambda_2 \quad \frac{\partial \tilde{f}}{\partial v} = -2dv + 2d\lambda_3$$

recall that we assumed $\lambda_1 = 1$. Then $\tilde{f}$ is singular if and only if $\tilde{f}(-\lambda_2, \lambda_3) = 0$ which is equivalent to

$$\lambda_4 = \lambda_2^2 - d\lambda_3^2 = (\lambda_2 + \alpha\lambda_3)^{q+1}.$$

In such a situation a computation to f from $\tilde{f}$ yields

$$\begin{aligned} f(z) &= z^{q+1} + \lambda_2 \mathrm{Tr}(z) + \lambda_3 \mathrm{Tr}(\alpha z) + (\lambda_2 + \alpha\lambda_3)^{q+1} \\ &= (z - (\lambda_2 + \alpha\lambda_3))^{q+1}. \end{aligned}$$

Therefore, the singular conics of the form (48) correspond to the words $(\boldsymbol{x}_0 - a)^{q+1}$ for $a \in \mathbb{F}_{q^2}$. In terms of codewords, either $a \in \{x_1, \ldots, x_n\}$ and the word $(\boldsymbol{x}_0 - a)^{q+1}$ has weight $n - 2$, or it is an element of U.

(iv) Finally, for $\lambda_1 = 1$ and $\lambda_4 \neq \lambda_2^2 + d\lambda_3^2$, the conic is nonsingular and it is well-known that affine nonsingular conics have at least $q - 1$ points (see for instance [57, Chapter 9.3]).

In summary, only cases (ii) and (iv) may provide codewords of full weight which are not in U. The number of lines coming from (ii) is the number of lines in the affine plane, namely $q^2 + q$. On the other hand, the number of conics coming from (iv) is the number of possible triples $(\lambda_2, \lambda_3, \lambda_4)$ with $\lambda_4 \neq \lambda_2^2 + d\lambda_3^2$. That is $q^3 - q^2$.

As a conclusion, there are $q^3 + q$ conics having at least $q - 1$ points which may be contained in A. The probability that such a conic is contained in A is therefore equal to 0 if $|A| < q - 1$ and is $\leqslant (q^3+q)\frac{\binom{q^2-q+1}{q^2-|A|}}{\binom{q^2}{q^2-|A|}}$ otherwise. Since $|A| = q^2 - (n-1)$, this yields the result. ∎

Remark 9. Actually, a further study proves that the nonsingular conics considered in the proof have all $q + 1$ points. This permits to obtain a sharper bound for the probability. Details are left out here.

q	29	31	31	31	31
n	791	892	851	813	795
U	$3\ 10^{-34}$	$2.3\ 10^{-33}$	$4.7\ 10^{-26}$	$1.06\ 10^{-21}$	$4.7\ 10^{-20}$

TABLE III
ESTIMATES OF THE UPPER BOUND U ON $1-\mathbf{P}$, WHERE $\mathbf{P}$ IS DEFINED IN PROPOSITION 34 FOR SOME EXPLICIT PARAMETERS.

3) Associating solutions by pairs: First remind that here again, words are considered only up to a multiplication by a scalar. In the previous subsections, we proved that with a very high probability, the inverses of the solutions of Problem (30) are

- $\boldsymbol{x}_0^{q+1}$ which is the solution we look for;
- the words $\boldsymbol{x}_0^{q+1} \star (\boldsymbol{x}_0 - a)^{-(q+1)}$, $a \in \mathbb{F}_{q^2} \setminus \{x_0, \ldots, x_{n-1}\}$.

In the very same manner, after computing the same filtration at position 1, one can then solve a linear problem of the form of Problem (30) whose inverse full weight solutions are

- $(\boldsymbol{x}_1 - 1)^{q+1}$ which is the one we look for;
- the words $(\boldsymbol{x}_1 - 1)^{q+1} \star (\boldsymbol{x}_1 - a)^{-(q+1)}$ $a \in \mathbb{F}_{q^2} \setminus \{x_0, \ldots, x_{n-1}\}$.

Basically, we have two sets of $q^2 - n + 1$ vectors (one can exclude the all-one vector $\mathbf{1}$ which is found easily). The first set contains $\boldsymbol{x}_0^{q+1}$ and the second one contains $(\boldsymbol{x}_1 - 1)^{q+1}$. But we do not know which ones they are. The first idea would be to iterate Steps 3 and 4 of the attack until the attack succeeds which represents in the worst case $(q^2-n+1)^2$ iterations. The point of this section is to explain how to reduce it to q^2-n+1 iterations in the worst case. For this purpose, let us bring in some notation.

Notation 2. Let $\boldsymbol{x}_{01}$ be the vector $\boldsymbol{x}$ punctured at positions $0, 1$. For all $a \in \mathbb{F}_{q^2} \setminus \{x_0, \ldots, x_{n-1}\}$, set

$$\begin{aligned} \boldsymbol{u}_0(a) &\stackrel{\text{def}}{=} \boldsymbol{x}_{01}^{q+1} \star (\boldsymbol{x}_{01} - a)^{-(q+1)} \\ \boldsymbol{u}_1(a) &\stackrel{\text{def}}{=} (\boldsymbol{x}_{01} - 1)^{q+1} \star (\boldsymbol{x}_{01} - a)^{-(q+1)}. \end{aligned}$$

Moreover, set

$$\boldsymbol{u}_0(\infty) \stackrel{\text{def}}{=} \boldsymbol{x}_{01}^{q+1} \quad \text{and} \quad \boldsymbol{u}_1(\infty) \stackrel{\text{def}}{=} (\boldsymbol{x}_{01} - 1)^{q+1},$$

which can be regarded as $\boldsymbol{u}_0(a)$ (resp. $\boldsymbol{u}_1(a)$) "when setting $a = \infty$". Finally, set

$$\begin{aligned} \mathcal{L}_0 &\stackrel{\text{def}}{=} \{\boldsymbol{u}_0(a) \mid a \in (\mathbb{F}_{q^2} \cup \{\infty\}) \setminus \{x_0, \ldots, x_{n-1}\}\} \\ \mathcal{L}_1 &\stackrel{\text{def}}{=} \{\boldsymbol{u}_1(a) \mid a \in (\mathbb{F}_{q^2} \cup \{\infty\}) \setminus \{x_0, \ldots, x_{n-1}\}\}. \end{aligned}$$

Lemma 35. *Assume that $n > 2q + 4$. Let $a, b, c, d \in (\mathbb{F}_{q^2} \cup \{\infty\}) \setminus \{x_0, \ldots, x_{n-1}\}$. Then, the vectors $\boldsymbol{u}_0(a) \star \boldsymbol{u}_1(b)$ and $\boldsymbol{u}_0(c) \star \boldsymbol{u}_1(d)$ are collinear if and only if*

$$\begin{array}{rlll} \text{either} & a = c & \text{and} & b = d \\ \text{or} & a = d & \text{and} & b = c. \end{array}$$

Proof: The "if" part is straightforward. Conversely, assume that $\boldsymbol{u}_0(a) \star \boldsymbol{u}_1(b)$ and $\boldsymbol{u}_0(c) \star \boldsymbol{u}_1(d)$ are collinear. Thus, there exists a nonzero scalar $\lambda \in \mathbb{F}_{q^2}$ such that

$$\boldsymbol{u}_0(a) \star \boldsymbol{u}_1(b) = \lambda \boldsymbol{u}_0(c) \star \boldsymbol{u}_1(d). \tag{49}$$

For convenience, we assume that a, b, c and d are all distinct from ∞. The cases when some of them equal ∞ are treated in the same way- we therefore omit to detail these cases here. From (49), we have that for all i in $\{2, \ldots, n-1\}$,

$$\left(\frac{x_i(x_i-1)}{(x_i-a)(x_i-b)}\right)^{q+1} = \lambda \left(\frac{x_i(x_i-1)}{(x_i-c)(x_i-d)}\right)^{q+1}.$$

This leads to

$$(x_i - c)^{q+1}(x_i - d)^{q+1} = \lambda (x_i - a)^{q+1}(x_i - b)^{q+1}, \tag{50}$$

From (50), the polynomial $P(z) \stackrel{\text{def}}{=} ((z-c)(z-d))^{q+1} - \lambda((z-a)(z-b))^{q+1}$ vanishes at x_i for all $i \in \{2, \ldots, n-1\}$, and hence has $n-2$ roots, while its degree is less than or equal to $2q+2$. Thus, under the assumption $n > 2q+4$, this polynomial has more roots than its degree and hence is zero. This yields the result. ∎

Proposition 36. *Assume that $n > 2q + 4$ Let $a, a' \in (\mathbb{F}_{q^2} \cup \{\infty\}) \setminus \{x_0, \ldots, x_{n-1}\}$. If we have the following equality of sets:*

$$\{\boldsymbol{u}_0(a) \star \boldsymbol{c} \mid \boldsymbol{c} \in \mathcal{L}_1\} = \{\boldsymbol{c}' \star \boldsymbol{u}_1(a') \mid \boldsymbol{c}' \in \mathcal{L}_0\},$$

where vectors are considered up to multiplication by a scalar, then, $a = a'$.

Proof: Clearly, if $a = a'$ then every element of the left hand set is of the form $\boldsymbol{u}_0(a) \star \boldsymbol{u}_1(b)$, for some $b \in (\mathbb{F}_{q^2} \cup \{\infty\}) \setminus \{x_0, \ldots, x_{n-1}\}$ and, from Lemma 35, this vector is collinear to $\boldsymbol{u}_0(b) \star \boldsymbol{u}_1(a)$.

Now, if $a \neq a'$, then let $b \in (\mathbb{F}_{q^2} \cup \{\infty\}) \setminus \{x_0, \ldots, x_{n-1}\}$ and $b \neq a, a'$. Then, Lemma 35 asserts that for all $c \in (\mathbb{F}_{q^2} \cup \{\infty\}) \setminus \{0, \ldots, n-1\}$, $\boldsymbol{u}_0(c) \star \boldsymbol{u}_1(a')$ is non collinear to $\boldsymbol{u}_0(a) \star \boldsymbol{u}_1(b)$. ∎

Proposition 36 allows to gather elements of $\mathcal{L}_0, \mathcal{L}_1$ by pairs $(\boldsymbol{u}_0(a), \boldsymbol{u}_1(a))$ without knowing a. We proceed as follows: we compute all the sets

$$\boldsymbol{a}_0 \star \mathcal{L}_1 \stackrel{\text{def}}{=} \{\boldsymbol{a}_0 \star \boldsymbol{c} \mid \boldsymbol{c} \in \mathcal{L}_1\}$$

for all $\boldsymbol{a}_0 \in \mathcal{L}_0$ and all the sets

$$\mathcal{L}_0 \star \boldsymbol{a}_1 \stackrel{\text{def}}{=} \{\boldsymbol{c}' \star \boldsymbol{a}_1 \mid \boldsymbol{c}' \in \mathcal{L}_0\}.$$

for all $\boldsymbol{a}_1 \in \mathcal{L}_1$.

Then, if two such sets match i.e. if $\boldsymbol{a}_0 \star \mathcal{L}_1 = \mathcal{L}_0 \star \boldsymbol{a}_1$, then we create the pair $(\boldsymbol{a}_0, \boldsymbol{a}_1)$. By Proposition 36 they correspond to pairs of the form $(\boldsymbol{u}_0(a), \boldsymbol{u}_1(a))$.By this manner, we create $q^2 - n + 1$ pairs of elements of $\mathcal{L}_0 \times \mathcal{L}_1$. One of them is the one we look for, namely the pair $(\boldsymbol{x}^{q+1}, (\boldsymbol{x} - 1)^{q+1})$.

G. Further details on Step 4 of the attack

Thanks to Lemma 23, one can compute the minimal polynomial P_{x_i} of every entry x_i of $\boldsymbol{x}$, that is to say that the support is known up to Galois action.

Fact 2. *From the very knowledge of these P_{x_i}'s, one can compute a (non unique) permutation σ such that $\boldsymbol{x}^\sigma \stackrel{\text{def}}{=} \sigma(\boldsymbol{x})$ is of the form*

$$\begin{aligned} \boldsymbol{x}^\sigma = (&u_0, u_1, \ldots, u_{\ell_1}, \\ &v_0, v_0^q, \ldots, v_{\ell_2-1}, v_{\ell_2-1}^q, w_0, \ldots, w_{\ell_3-1}) \end{aligned} \tag{51}$$

where

- *the u_i's list all the entries of $\boldsymbol{x}$ lying in $\mathbb{F}_q$;*
- *the v_i's list all the entries of $\boldsymbol{x}$ in $\mathbb{F}_{q^2} \setminus \mathbb{F}_q$ and whose conjugate is also an entry of $\boldsymbol{x}$.*
- *the w_i's list all the entries of $\boldsymbol{x}$ in $\mathbb{F}_{q^2} \setminus \mathbb{F}_q$ and whose conjugate is not an entry of $\boldsymbol{x}$.*

Therefore, one can compute a generator matrix of the code $\mathscr{C}^\sigma \stackrel{\text{def}}{=} \mathscr{G}\left(\boldsymbol{x}^\sigma, \gamma^{q+1}\right)$ by permuting the columns of a generator matrix of $\mathscr{C}$. Call $\boldsymbol{G}^\sigma \in \mathbb{F}_q^{k\times n}$ this matrix. In other words $\boldsymbol{G}^\sigma$ is obtaining by first picking the columns of a generator matrix $\boldsymbol{G}$ of $\mathscr{C}$ that correspond to the entries of $\boldsymbol{x}$ that belong to $\mathbb{F}_q$ and then putting together the columns of $\boldsymbol{G}$ that correspond to conjugate entries of $\boldsymbol{x}$ and finally put at the end the columns of $\boldsymbol{G}$ that are not of this kind.

It is worth noting that, even when σ is computed, the vector $\boldsymbol{x}^\sigma$ remains unknown since only the minimal polynomials of its entries are known. Afterwards, we introduce an extended support $\boldsymbol{x}^\sigma_{\mathbf{ext}}$ by inserting the conjugates of the w_j's which is of the form

$$\begin{array}{l}(u_0,\ldots,u_{\ell_1-1},\\ \quad v_0, v_0^q, \ldots, v_{\ell_2-1}, v_{\ell_2-1}^q, w_0, w_0^q, \ldots, w_{\ell_3-1}, w_{\ell_3-1}^q).\end{array} \tag{52}$$

This vector is also unknown, however, one can compute an arbitrary vector which equals $\boldsymbol{x}^\sigma_{\mathbf{ext}}$ up to a very particular permutation. One can namely compute an arbitrary vector $\boldsymbol{x}'_{\mathbf{ext}}$ of the form

$$\begin{array}{l}(u_0,\ldots,u_{\ell_1-1},\\ \quad v'_0, {v'_0}^q, \ldots, v'_{\ell_2-1}, {v'}^q_{\ell_2-1}, w'_0, {w'_0}^q \ldots w'_{\ell_3-1}, {w'}^q_{\ell_3-1})\end{array} \tag{53}$$

such that for all i the i-th entry of $\boldsymbol{x}'_{\mathbf{ext}}$ has the same minimal polynomial as that of $\boldsymbol{x}^\sigma_{\mathbf{ext}}$. Equivalently, the entries of $\boldsymbol{x}'_{\mathbf{ext}}$ equal those of $\boldsymbol{x}^\sigma_{\mathbf{ext}}$ up to Galois action. This can be interpreted in terms of permutations using:

Definition 6. Let $\mathfrak{T}$ be the subgroup of $\mathfrak{S}_{n+\ell_3}$ of products of transpositions with disjoint supports, each one permuting either the positions v_i, v_i^q the positions w_i, w_i^q in $\boldsymbol{x}^\sigma_{\mathbf{ext}}$. Every element $\tau \in \mathfrak{T}$ is represented by the matrix

$$\mathbf{R}_\tau \stackrel{\text{def}}{=} \begin{pmatrix} \mathbf{I}_{\ell_1} & (0) \\ (0) & \mathbf{B} \end{pmatrix}, \tag{54}$$

where $\mathbf{B} \in \mathfrak{M}_{2(\ell_2+\ell_3)}(\mathbb{F}_q)$ is 2×2 block–diagonal with blocks among $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$.

Lemma 37. *There exists $\tau \in \mathfrak{T}$ such that $\tau(\boldsymbol{x}^\sigma_{\mathbf{ext}}) = \boldsymbol{x}'_{\mathbf{ext}}$.*

We extend $\boldsymbol{G}^\sigma$ as a $k\times(n+\ell_3)$ matrix $\boldsymbol{G}^\sigma_{\mathbf{ext}}$ by inserting ℓ_3 zero columns at positions in one-to-one correspondence with the entries w_i^q in $\boldsymbol{x}^\sigma_{\mathbf{ext}}$ (see (52)). That is:

$$\boldsymbol{G}^\sigma = \begin{pmatrix} g_{11} & \ldots & g_{1n} \\ \vdots & & \vdots \\ g_{k1} & \ldots & g_{kn} \end{pmatrix} \tag{55}$$

$$\boldsymbol{G}^\sigma_{\mathbf{ext}} = \begin{pmatrix} g_{11} & \ldots & g_{1s} & 0 & g_{1,s+1} & 0 & \ldots & g_{1n} & 0 \\ \vdots & & & & & & & & \vdots \\ g_{k1} & \ldots & g_{ks} & 0 & g_{k,s+1} & 0 & \ldots & g_{kn} & 0 \end{pmatrix},$$

where $s = \ell_1 + 2\ell_2$ (see (51)). The corresponding code is referred to as $\mathscr{C}^\sigma_{\text{ext}}$. The key of this final step is the following statement.

Theorem 38. *There exists a matrix $\mathbf{M}$ such that*

$$\mathscr{C}^\sigma_{ext}\ \mathbf{M} \ \subseteq\ \mathscr{A}_{r(q+1)}(\boldsymbol{x}'_{\mathbf{ext}}, \mathbf{1}) \tag{56}$$

and $\mathbf{M} = \mathbf{R}_\tau \mathbf{D}$, where $\mathbf{D}$ is diagonal and invertible and $\mathbf{R}_\tau$ is the permutation matrix of some $\tau \in \mathfrak{T}$ as described in (54).

Proof: Recall that $\mathscr{C}^\sigma = \mathscr{G}\left(\boldsymbol{x}^\sigma, \gamma^{q+1}\right)$. Since from Definition 3, Goppa codes are alternant and hence from Proposition 9, the code $\mathscr{C}^\sigma$ is a shortening of $\mathscr{G}\left(\boldsymbol{x}^\sigma_{\mathbf{ext}}, \gamma^{q+1}\right)$. Consequently, we have

$$\mathscr{C}^\sigma_{\text{ext}} \subseteq \mathscr{G}\left(\boldsymbol{x}^\sigma_{\mathbf{ext}}, \gamma^{q+1}\right). \tag{57}$$

Second, from Lemma 37, there exists $\tau \in \mathfrak{T}$ such that $\tau(\boldsymbol{x}^\sigma_{\mathbf{ext}}) = \boldsymbol{x}'_{\mathbf{ext}}$, then

$$\mathscr{G}\left(\boldsymbol{x}^\sigma_{\mathbf{ext}}, \gamma^{q+1}\right)\mathbf{R}_\tau = \mathscr{G}\left(\boldsymbol{x}'_{\mathbf{ext}}, \gamma^{q+1}\right), \tag{58}$$

Next, from Theorem 14(iii), there exists $\boldsymbol{a} \in (\mathbb{F}_q^\times)^{n+\ell_3}$ such that

$$\mathscr{G}\left(\boldsymbol{x}'_{\mathbf{ext}}, \gamma^{q+1}\right) = \boldsymbol{a} \star \mathscr{A}_{r(q+1)}(\boldsymbol{x}'_{\mathbf{ext}}, \mathbf{1}). \tag{59}$$

Let $\mathbf{D}$ be the diagonal matrix whose diagonal equals $\boldsymbol{a}^{-1}$. From (57), (58) and (59), we get

$$\mathscr{C}^\sigma_{\text{ext}}\mathbf{R}_\tau\mathbf{D} \subseteq \mathscr{A}_{r(q+1)}(\boldsymbol{x}'_{\mathbf{ext}}, \mathbf{1}). \tag{60}$$

■

To finish the attack, we proceed as follows. We compute a permutation σ as in Fact 2. Then, we compute the matrix $\boldsymbol{G}^\sigma_{\mathbf{ext}}$ defined in (55). Afterwards, we compute an arbitrary vector $\boldsymbol{x}'_{\mathbf{ext}}$ and a parity–check matrix $\mathbf{H}$ of the code $\mathscr{A}_{r(q+1)}(\boldsymbol{x}'_{\mathbf{ext}}, \mathbf{1})$. Finally, we solve the problem

Problem 2. *Find the space of matrices $\mathbf{M} \in \mathfrak{M}_{n+\ell_3}(\mathbb{F}_q)$ of the form $\mathbf{M} = \begin{pmatrix} \mathbf{E} & (0) \\ (0) & \mathbf{F} \end{pmatrix}$, where $\mathbf{E}$ is $\ell_1 \times \ell_1$ and diagonal and $\mathbf{F}$ is $(2\ell_2+2\ell_3)\times(2\ell_2+2\ell_3)$ and 2×2–block–diagonal such that*

$$\mathbf{H}\ (\boldsymbol{G}^\sigma_{\mathbf{ext}}\ \mathbf{M})^T\ =\ 0.$$

The matrix $\mathbf{M}$ of Theorem 38 is a solution of Problem 2. Moreover, this problem is linear, has $\ell_1 + 2(\ell_2+\ell_3) \leqslant 4n$ unknowns and $(\dim\mathscr{C})(n-\dim\mathscr{A}_{r(q+1)}(\boldsymbol{x}'_{\mathbf{ext}}, \mathbf{1})) \geqslant k(n-k)$ equations. Thus, the number of unknowns is linear while the number of equations is quadratic. This provides an extremely small space of solutions.

Example 1. If we consider a $[841, 601]$ wild Goppa code over $\mathbb{F}_{32}$ (where $r = 4$), then we get less than 3364 unknowns and more than 120200 equations.

Experimentally, we observe that the solution space has dimension 2 and an exhaustive search of matrices which are the product of a diagonal matrix and a permutation matrix provide two solutions (see Lemma 39 for the rationale behind these two solutions). Choose a solution $\mathbf{M}$, then factorize it as $\mathbf{D}\mathbf{R}_\tau$ as in Theorem 38. This yields the permutation τ and

Algorithm 2 Algorithm of the attack.

Compute $\mathscr{C}_0(q+1)$, $\mathscr{C}_1(q+1)$ using Algorithm 1.
$\mathcal{L}_0 \leftarrow$ List of candidates for $\boldsymbol{x}_0^{q+1}$ (Obtained by solving (30))
$\mathcal{L}_1 \leftarrow$ List of candidates for $(\boldsymbol{x}_1 - 1)^{q+1}$
$\mathcal{P} \leftarrow$ the set of $q^2 - n + 1$ pairs $(\boldsymbol{a}_0, \boldsymbol{a}_1) \in \mathcal{L}_0 \times \mathcal{L}_1$ as explained in §F3.
$\boldsymbol{M}_0 \leftarrow 0$
while $\boldsymbol{M}_0 = 0$ and $L_0 \neq \emptyset$ **do**
  $(\boldsymbol{a}_0, \boldsymbol{a}_1) \leftarrow$ a random pair in $\mathcal{P}$.
  $\mathcal{P} \leftarrow \mathcal{P} \setminus \{(\boldsymbol{a}_0, \boldsymbol{a}_1)\}$
  Compute the minimal polynomials P_i of the positions using Lemma 23.
  Construct $\boldsymbol{G}_{\mathbf{ext}}^{\sigma}$, $\boldsymbol{x}'_{\mathbf{ext}}$ and a parity-check matrix $\boldsymbol{H}$ of the code $\mathscr{A}_{r(q+1)}(\boldsymbol{x}'_{\mathbf{ext}}, \mathbf{1})$ as described in Theorem 38.
  $V \leftarrow$ Space of solutions $\boldsymbol{M}$ of Problem 2
  if $\dim V > 0$ and $\exists \boldsymbol{M} \in V$ of the form $\mathbf{DR}_\tau$ as in Theorem 38 **then**
    $\boldsymbol{M}_0 \leftarrow M$
  end if
end while
if $\boldsymbol{M}_0 = 0$ **then**
  return "error"
else
  Recover $\boldsymbol{x}$ and $\boldsymbol{u}$ from $\boldsymbol{M}$ as in (61)
  return $\boldsymbol{x}, \boldsymbol{u}$
end if

hence the support $\boldsymbol{x}$. Second, the entries of $\mathbf{D}$ provide directly a vector $\boldsymbol{a}$ such that

$$\mathscr{C} = \boldsymbol{a} \star \mathscr{A}_{r(q+1)}(\boldsymbol{x}, \mathbf{1}), \tag{61}$$

which allows to correct up to $\lfloor \frac{r(q+1)}{2} \rfloor$ errors. Hence the scheme is broken.

1) The two solutions of the problem: The solutions of Problem 2 of the form $\mathbf{DR}_\tau$, where $\mathbf{D}$ is diagonal and invertible and $\mathbf{R}_\tau$ is a permutation matrix has cardinality 2. This is due to the fact that any alternant code of extension degree 2 has at least 2 pairs $(\boldsymbol{x}, \boldsymbol{y})$ to represent it. This explained in the following lemma.

Lemma 39. *Let $\boldsymbol{a} \in \mathbb{F}_{q^2}^n$ be a support and $\boldsymbol{b} \in \mathbb{F}_{q^2}^n$ be a multiplier. Then,*

$$\mathbf{GRS}_k(\boldsymbol{a}, \boldsymbol{b}) \cap \mathbb{F}_q^n = \mathbf{GRS}_k(\boldsymbol{a}^q, \boldsymbol{b}^q) \cap \mathbb{F}_q^n.$$

Proof: Let $f \in \mathbb{F}_{q^2}[x]_{<k}$ be a polynomial such that $(b_0 f(a_0), \ldots, b_{n-1} f(a_{n-1})) \in \mathbf{GRS}_k(\boldsymbol{a}, \boldsymbol{b}) \cap \mathbb{F}_q^n$. Writing f as $f_0 + f_1 x + \cdots + f_{k-1} x^{k-1}$, denote by $f^{(q)} \in \mathbb{F}_{q^2}[x]_{<k}$ the polynomial $f^{(q)} \stackrel{\text{def}}{=} f_0^q + f_1^q x + \cdots + f_{k-1}^q x^{k-1}$. Then it is easy to check that for all $i \in \{0, \ldots, n-1\}$, we have

$$b_i^q f^{(q)}(a_i^q) = (b_i f(a_i))^q.$$

In addition, since by assumption $b_i f(a_i) \in \mathbb{F}_q$, we have

$$\forall i \in \{0, \ldots, n-1\},\ b_i^q f^{(q)}(a_i^q) = b_i f(a_i)$$

Therefore,

$$(b_0 f(a_0), \ldots, b_{n-1} f(a_{n-1})) = (b_0^q f^{(q)}(a_0^q), \ldots, b_{n-1} f^{(q)}(a_{n-1}^q)) \in \mathbf{GRS}_k(\boldsymbol{a}^q, \boldsymbol{b}^q) \cap \mathbb{F}_q^n.$$

We proved that $\mathbf{GRS}_k(\boldsymbol{a}, \boldsymbol{b}) \cap \mathbb{F}_q^n \subseteq \mathbf{GRS}_k(\boldsymbol{a}^q, \boldsymbol{b}^q) \cap \mathbb{F}_q^n$ and the converse inclusion can be proved by the very same manner. ■

A. Couvreur received the Agrégation de Mathmatiques in 2004 and the Ph.D. degree in pure mathematics from the University of Toulouse (France) in 2008. He is presently INRIA junior researcher (chargé de recherche) at École Polytechnique (France).

Ayoub Otmani holds a Ph.D. degree in mathematics and its applications from the University of Limoges (France) since 2002, and the Habilitation à Diriger des Recherches degree from the University of Caen (France) since 2011. He is currently Professor at University of Rouen, France. His research lies in the fields of algebraic coding theory and cryptography.

Jean-Pierre Tillich 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 Supérieure des Télécommunications (ENST), Paris, in 1994. From 1997 to 2003, he was Assistant Professor at the University Paris XI. He is now a Researcher at the Institut de Recherche en Informatique et Automatique (INRIA), Paris, France.

From 2009 to 2012 he was an Associate Editor for Coding Theory for the IEEE TRANSACTIONS ON INFORMATION THEORY. His research interests include classical and quantum coding theory, cryptography and graph theory.