Polynomial Time Attack on Wild McEliece Over Quadratic Extensions

Alain Couvreur[1], Ayoub Otmani[2], and Jean–Pierre Tillich[3]

[1] GRACE Project — INRIA Saclay & LIX, CNRS UMR 7161 — École Polytechnique, 91120 Palaiseau Cedex, France. alain.couvreur@lix.polytechnique.fr
[2] Normandie Univ, France; UR, LITIS, F-76821 Mont-Saint-Aignan, France. ayoub.otmani@univ-rouen.fr
[3] SECRET Project — INRIA Rocquencourt, 78153 Le Chesnay Cedex, France. jean-pierre.tillich@inria.fr

Abstract. We present a polynomial time structural attack against the McEliece system based on Wild Goppa codes from a quadratic finite field extension. This attack uses the fact that such codes can be distinguished from random codes to compute some filtration, that is to say a family of nested subcodes which will reveal their secret algebraic description.

Keywords: public-key cryptography, wild McEliece cryptosystem, filtration, cryptanalysis.

1 Introduction

The McEliece cryptosystem and its security. The McEliece encryption scheme [35] which dates backs to the end of the seventies still belongs to the very few public-key cryptosystems which remain unbroken. It is based on the famous Goppa codes family. Several proposals which suggested to replace binary Goppa 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) [38] or subcodes of them [4], Reed–Muller codes [43], algebraic geometry codes [22], LDPC and MDPC codes [2, 37] or convolutional codes [28]. Most of them were successfully cryptanalyzed [44, 48, 36, 19, 39, 13, 24, 14]. 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 [37] and their cousins [2], the original binary Goppa codes of [35] and their non-binary variants as proposed in [6, 7].

Concerning the security of the McEliece proposal based on Goppa codes, weak keys were identified in [21, 27] but they can be easily avoided. There also exist generic attacks by exponential decoding algorithms [25, 26, 45, 10, 5, 34, 3]. More recently, it was shown in [18, 20] that the secret structure of Goppa codes can be recovered by an algebraic attack using Gröbner bases. This attack is of exponential nature and is infeasible for the original McEliece scheme (the number of unknowns is linear in the length of the code), whereas for variants using Goppa codes with a quasi-dyadic or quasi-cyclic structure it was feasible due to the huge reduction of the number of unknowns.

Distinguisher for Goppa and Reed-Solomon codes. None of the existing strategies is able to severely dent the security of [35] 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 fact that Goppa codes share many characteristics with random codes. However, in [16, 17], an algorithm that manages to distinguish between a random code and a high rate Goppa code has been introduced.

Code product. [33] showed that the distinguisher given in [16] 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 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 [40, 23], then exploited in cryptology in [48] to break a McEliece variant based on random subcodes of GRS codes [4] and in [30, 32] to study the security of encryption schemes using algebraic-geometric codes. Component-wise powers of codes are also studied in the context of secret sharing and secure multi-party computation [11, 12].

Distinguisher-based key-recovery attacks. The works [16, 17], without undermining the security of [35], prompts to wonder whether it would be possible to devise an attack exploiting the distinguisher. That was indeed the case in [13] for McEliece-like public-key encryption schemes relying on modified GRS codes [8, 1, 47]. Additionnally, [13] has shown that the unusually low dimension of the square code of a generalized GRS code enables to compute a nested sequence of subcodes – we call this a filtration – allowing the recovery of its algebraic structure. This gives a completely different attack from [44] of breaking GRS-based encryption schemes. In particular, compared to the attack of [44] on GRS codes and to the attack of [36, 19] on binary Reed–Muller codes and low-genus algebraic geometry codes, this new way of cryptanalyzing does not require as a first step the computation of minimum weight codewords, which is polynomial in time only for the very specific case of GRS codes.

Our contribution. The purpose of this article is to show that the filtration attack of [13] which gave a new way of attacking a McEliece scheme based on GRS codes can be generalized to other families of codes. It leads for instance to a successful attack of McEliece based on high genus algebraic geometry codes [14]. 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 the smallest field extension, that is $m = 2$, for which these subfield subcodes are not GRS 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 [6]. Such codes 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 an abnormally small dimension.

We show here that this distinguishing property can be used to compute a filtration of the public code, that is to say a family of nested subcodes of the public Goppa code. 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. This attack has been implemented in Magma [9] and allowed to break completely all the schemes with a claimed 128 bit security in Table 7.1 of [6] corresponding to $m = 2$ when 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 35 years of existence of the McEliece scheme based on Goppa codes that a polynomial time attack has been found on it. It questions the common belief that GRS codes are weak for a cryptographic use while Goppa codes are secure as soon as $m \geqslant 2$ and that for the latter only generic information-set-decoding attacks apply. It also raises the issue whether this algebraic distinguisher of Goppa and more generally alternant codes (see [17]) 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.

Note that due to space constraints, the results are given here without proofs. For more details we refer to a forthcoming paper.

2 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 [29] for the terminology.

Star 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, *indexes that are in* $\mathcal{I}$ *are removed*. When $\mathcal{I} = \{j\}$ we allow ourselves to write $\boldsymbol{u}_j$ instead of $\boldsymbol{u}_{\{j\}}$. The component-wise 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 denote by $\boldsymbol{u}^{-1} \stackrel{\text{def}}{=} (u_0^{-1}, \ldots, u_{n-1}^{-1})$, 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 $\mathbf{1}$. To a vector $\boldsymbol{x} \in \mathbb{F}_q^n$, we associate the set $\mathcal{L}_{\boldsymbol{x}} \stackrel{\text{def}}{=} \{x_i \mid i \in \{0, \ldots, n-1\}\}$ which is defined as the set of entries of $\boldsymbol{x}$. We always have $|\mathcal{L}_{\boldsymbol{x}}| \leqslant n$ and equality holds when the entries of $\boldsymbol{x}$ are pair-wise distinct.

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 polynomial $P \in \mathbb{F}_q[z]$, $P(\boldsymbol{u})$ 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 $(au_0 + b, \ldots, au_{n-1} + b)$. To each vector $\boldsymbol{x} = (x_0, \ldots, x_{n-1}) \in \mathbb{F}_q^n$, we associate its *locator polynomial* denoted as $\pi_{\boldsymbol{a}}$ and defined as $\pi_{\boldsymbol{x}}(z) \stackrel{\text{def}}{=} \prod_{i=0}^{n-1}(z - x_i)$. Its first derivative is denoted as $\pi'_{\boldsymbol{x}}$ and one shows easily that its evaluation at the entries of $\boldsymbol{x}$ yields the vector $\pi'_{\boldsymbol{x}}(\boldsymbol{x}) = \left(\prod_{j \neq i}(x_i - x_j)\right)_{0 \leqslant i < n}$.

The *norm* and *trace* from $\mathbb{F}_{q^2}$ to $\mathbb{F}_q$ when applied to any $\boldsymbol{x} \in \mathbb{F}_{q^2}^n$ are respectively $\mathrm{N}(\boldsymbol{x})$ and $\mathrm{Tr}(\boldsymbol{x})$ with by definition $\mathrm{N}(\boldsymbol{x}) \stackrel{\text{def}}{=} \left(x_0^{q+1}, \ldots, x_{n-1}^{q+1}\right)$ and $\mathrm{Tr}(\boldsymbol{x}) \stackrel{\text{def}}{=} \left(x_0^q + x_0, \ldots, x_{n-1}^q + x_{n-1}\right)$.

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

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

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 results will be used repeatedly.

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

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

Diagonal equivalence of codes. Two q-ary codes $\mathscr{A}, \mathscr{B} \subset \mathbb{F}_q^n$ are said to be $\mathbb{F}_q$*–diagonally equivalent*, and we will write $\mathscr{B} \sim_{\mathbb{F}_q} \mathscr{A}$, if there exists $\boldsymbol{u} \in \left(\mathbb{F}_q^{\times}\right)^n$ such that:

$$\mathscr{B} = \boldsymbol{u} \star \mathscr{A} = \{\boldsymbol{u} \star \boldsymbol{a} \mid \boldsymbol{a} \in \mathscr{A}\}.$$

It is equivalent to say that $\mathscr{A}$ and $\mathscr{B}$ are $\mathbb{F}_q$–equivalent if $\mathscr{B}$ is the image of $\mathscr{A}$ by an invertible diagonal matrix whose diagonal is $\boldsymbol{u}$.

Generalized Reed–Solomon, Alternant and Classical Goppa 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 $\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$ of dimension k associated to $(\boldsymbol{x}, \boldsymbol{y})$ is the k-dimensional vector space*

$$\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y}) \stackrel{def}{=} \Big\{ \big(y_0 p(x_0), \ldots, y_{n-1} p(x_{n-1})\big) \;\mid\; p \in \mathbb{F}_q[z]_{<k} \Big\}.$$

Reed-Solomon codes correspond to the case where $y_i = 1$ for all $i \in \{0, \ldots, n-1\}$ and are denoted as $\mathbf{RS}_k(\boldsymbol{x})$. The vector $\boldsymbol{x}$ is called the support of the code.

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

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

This leads to the definition of alternant codes ([29, Chap. 12, §2]).

Definition 2 (Alternant code). *Let $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}_{q^m}^n$ be two vectors such that the entries of $\boldsymbol{x}$ are pairwise distinct and those of $\boldsymbol{y}$ are all nonzero. The alternant code $\mathscr{A}_r(\boldsymbol{x}, \boldsymbol{y})$ defined over $\mathbb{F}_q$ where $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}_{q^m}^n$ is the subfield subcode over $\mathbb{F}_q$ of the code $\mathbf{GRS}_r(\boldsymbol{x}, \boldsymbol{y})^\perp$ defined over $\mathbb{F}_{q^m}$, that is:*

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

The integer r is referred to as the degree *of the alternant code, the integer m as its* extension degree *and the vector $\boldsymbol{x}$ as its* support.

From this definition, 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 [29, Chap. 12, §9]):

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

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

Proposition 2.

$$\begin{aligned}\mathscr{A}_r(\boldsymbol{x}, \boldsymbol{y}) &= \left\{ \left(\frac{f(x_i)}{y_i \pi'_{\boldsymbol{x}}(x_i)} \right)_{0 \leqslant i < n} \;\middle|\; f \in \mathbb{F}_{q^m}[z]_{<n-r} \right\} \cap \mathbb{F}_q^n \\ &= \left\{ f(\boldsymbol{x}) \star \boldsymbol{y}^{-1} \star \pi'_{\boldsymbol{x}}(\boldsymbol{x})^{-1} \;\middle|\; f \in \mathbb{F}_{q^m}[z]_{<n-r} \right\} \cap \mathbb{F}_q^n.\end{aligned}$$

Definition 3 (Classical Goppa code). *Let $\boldsymbol{x}$ be an n–tuple of distinct elements of $\mathbb{F}_{q^m}$, let r be a positive integer and $\Gamma \in \mathbb{F}_{q^m}[z]$ be a polynomial of degree r such that $\Gamma(x_i) \neq 0$ for all $i \in \{0, \ldots, n-1\}$. The classical Goppa code $\mathscr{G}(\boldsymbol{x}, \Gamma)$ over $\mathbb{F}_q$ associated to Γ and supported by $\boldsymbol{s}$ is defined as*

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

We call Γ the Goppa polynomial, *$\boldsymbol{x}$ the* support *and m the* extension degree *of the Goppa code.*

As for alternant codes, the following description of Goppa codes, which is due to Proposition 2 will be extremely useful in this article.

Lemma 2.

$$\begin{aligned}\mathscr{G}(\boldsymbol{x}, \Gamma) &= \left\{ \left(\frac{\Gamma(x_i) f(x_i)}{\pi'_{\boldsymbol{x}}(x_i)} \right)_{0 \leqslant i < n} \;\middle|\; f \in \mathbb{F}_{q^2}[z]_{<n-\deg(\Gamma)} \right\} \cap \mathbb{F}_q^n \\ &= \left\{ \Gamma(\boldsymbol{x}) \star f(\boldsymbol{x}) \star (\pi'_{\boldsymbol{x}}(\boldsymbol{x}))^{-1} \;\middle|\; f \in \mathbb{F}_{q^2}[z]_{<n-\deg(\Gamma)} \right\} \cap \mathbb{F}_q^n\end{aligned}$$

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

Proposition 3 ([46]). *Let γ be a monic and square free polynomial of degree r. Let $\boldsymbol{x}$ be an n-tuple of distinct elements of $\mathbb{F}_{q^m}$ satisfying $\gamma(x_i) \neq 0$ for all i in $\{0, \ldots, n-1\}$, then:*

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

From Fact 1, these 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 seen as $\mathscr{A}_{(q-1)r}(\boldsymbol{x}, \gamma^{-(q-1)}(\boldsymbol{x}))$. 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} (called *wild Goppa codes*) are proposed in [6, 7].

McEliece encryption scheme. We recall here the general principle of McEliece's public-key scheme [35]. 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 corrected efficiently. 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 t and computes the ciphertext $\boldsymbol{c} \stackrel{\text{def}}{=} \boldsymbol{u}\boldsymbol{G} + \boldsymbol{e}$. The receiver then recovers the plaintext by applying $\mathcal{D}$ on $\boldsymbol{c}$.

This describes the general scheme suggested by McEliece. From now on, we will say that $\boldsymbol{G}$ is the *public generator matrix* and that the vector space $\mathscr{C}$ spanned by its rows is the *public code i.e.* $\mathscr{C} \stackrel{\text{def}}{=} \{\boldsymbol{u}\boldsymbol{G} \mid \boldsymbol{u} \in \mathbb{F}_q^k\}$. McEliece based his scheme solely on binary Goppa codes. In [6, 7], it is advocated to use q-ary Goppa codes with Goppa polynomials of the form γ^{q-1} because of their better error correction capability (see Proposition 3). Such codes are then named wild Goppa codes. 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.

3 A Distinguisher Based on Square Codes

From now on, and until the end of the article, $\mathscr{C}$ denotes the public code of the wild McEliece scheme we want to attack, that is $\mathscr{C} \stackrel{\text{def}}{=} \mathscr{G}\left(\boldsymbol{x}, \gamma^{q-1}\right)$ and we want to recover the *secret* support vector $\boldsymbol{x} \in \mathbb{F}_{q^2}^n$ and the *secret* irreducible polynomial $\gamma \in \mathbb{F}_{q^2}[z]$ that is assumed to be of degree $r > 1$. Such a Goppa code has extension degree 2 and we will first show in this section that it displays some peculiarities which allows to distinguish such codes from random ones. As in [16, 33], the main tool for achieving this purpose is given by square product considerations. It will turn out later on in Section 4 that the very reason which allows to distinguish these wild Goppa codes is also the fundamental reason which enables to compute a nested family of codes and hence used to reveal their algebraic structure.

3.1 Square code

One of the keys for the distinguisher presented here and the attack oulined in Section 4 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* star 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 $\mathscr{B} = \mathscr{A}$ then $\mathscr{A} \star \mathscr{A}$ is called the* square code *of $\mathscr{A}$ and is denoted by $\mathscr{A}^{\star 2}$.*

The dimension of the star product is easily bounded by:

Proposition 4. *Let $\mathscr{A}$ and $\mathscr{B}$ be two linear codes $\subseteq \mathbb{F}_q^n$, then*

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

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

Proof. Let $(e_1, \ldots, e_s)$ be a basis of $\mathscr{A} \cap \mathscr{B}$. Complete it as two bases $B_{\mathscr{A}} = (e_1, \ldots, e_s, a_{s+1}, \ldots, a_k)$ and $B_{\mathscr{B}} = (e_1, \ldots, e_s, b_{s+1}, \ldots, b_\ell)$ of $\mathscr{A}$ and $\mathscr{B}$ respectively. The star 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 $i \neq j$. This proves (1). The inequality given in (2) is a consequence of (1). □

Most codes of a given length and dimension reach these bounds while GRS codes behave completely differently when they have the same support.

Proposition 5. *Let $\boldsymbol{x}$ be an n–tuple of pairwise distinct elements of $\mathbb{F}_q$ and $\boldsymbol{y}, \boldsymbol{y}'$ be two n–tuples of nonzero elements of $\mathbb{F}_q$. 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) $\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})^{\star 2} = \mathbf{GRS}_{2k-1}(\boldsymbol{x}, \boldsymbol{y} \star \boldsymbol{y})$.

Remark 1. 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 [48] and appears for instance explicitly as Proposition 10 in [31]. We provide the proof here because it is crucial for understanding why the star products of GRS codes and some alternant codes behave in a non generic way.

Proof. Let $\boldsymbol{c} = (y_0 f(x_0), \ldots, y_{n-1} f(x_{n-1})) \in \mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$ and $\boldsymbol{c}' = (y'_0 g(x_0), \ldots, y'_{n-1} g(x_{n-1})) \in \mathbf{GRS}_{k'}(\boldsymbol{x}, \boldsymbol{y}')$ where $\deg(f) \leqslant k-1$ and $\deg(g) \leqslant k'-1$. Then, $\boldsymbol{c} \star \boldsymbol{c}'$ is of the form:

$$\boldsymbol{c} \star \boldsymbol{c}' = (y_0 y'_0 f(x_0) g(x_0), \ldots, y_{n-1} y'_{n-1} f(x_{n-1}) g(x_{n-1})) = (y_0 y'_0 r(x_0), \ldots, y_{n-1} y'_{n-1} r(x_{n-1}))$$

where $\deg(r) \leqslant k + k' - 2$. Conversely, any element $(y_0 y'_0 r(x_0), \ldots, y_{n-1} y'_{n-1} r(x_{n-1}))$ where $\deg(r) \leqslant k + k' - 2$, is a linear combination of star 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 an abnormal low dimension. This is is true but in a very attenuated form as shown by:

Theorem 2. *Let $\boldsymbol{x}$ be an n–tuple of distinct elements of $\mathbb{F}_{q^m}$, with $m \geqslant 1$. Let $\boldsymbol{y}, \boldsymbol{y}'$ be two n–tuples of nonzero elements of $\mathbb{F}_{q^m}$. There exists then $\boldsymbol{y}'' \in \mathbb{F}_{q^m}^n$ such that:*

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

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 Proposition 2,

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

for some polynomials f, g of respective degrees $< n - s$ and $< n - s'$. 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 $\boldsymbol{c}, \boldsymbol{c}'$ have their entries in $\mathbb{F}_q$, then, so has $\boldsymbol{c} \star \boldsymbol{c}'$. Consequently,

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

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

Remark 2. This theorem generalizes Proposition 5: it corresponds to the particular case $m = 1$. 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 dimension of $\mathscr{A}_s(\boldsymbol{x}, \boldsymbol{y})$ is $n - sm$ whereas the dimension of $\mathscr{A}_{s'}(\boldsymbol{x}, \boldsymbol{y}')$ is equal to $n - s'm$. If we assume that both codes have non trivial dimension then we should have $n - sm > 0$ and $n - s'm > 0$ which implies that $s < \frac{n}{m} \leqslant n/2$. Therefore we have $s \leqslant n/2 - 1$ and $s' \leqslant n/2 - 1$. This implies that $(s + s') - n + 2 \leqslant 0$, 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) at least one of the codes $\mathscr{A}_s(\boldsymbol{x}, \boldsymbol{y})$ and $\mathscr{A}_{s'}(\boldsymbol{x}, \boldsymbol{y}')$ has dimension greater than the designed dimension, or (ii) 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 $s'' > s$, then the right-hand term of (3) can be smaller than the full space (at least for small dimensions). This is precisely what happens for our wild Goppa codes of extension degree 2 as shown by:

Proposition 6 ([15]). *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 assumed to be 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) $\mathscr{G}\left(\boldsymbol{x}, \gamma^{q+1}\right) \sim_{\mathbb{F}_q} \mathscr{A}_{r(q+1)}(\boldsymbol{x}, \mathbf{1})$.

The results (i) and (ii) are respective straighforward consequences of Theorems 1 and 24 of [15]. Only (iii) which is used later on, requires further details, see the forthcoming long version of this article.

3.2 A distinguisher obtained by shortening

As explained in Remark 2 the square code of an alternant code of extension degree 2 may have an unusually low dimension when its dimension is larger than its designed rate. This is precisely what happens for wild Goppa codes as explained by Proposition 6.

Taking directly the square of the Goppa code does not work unless the rate of the code is close to 0. However, one can reduce to this case by the shortening operation:

Proposition 7. *Let $\boldsymbol{x}$ be an n–tuple of pairwise distinct elements in $\mathbb{F}_{q^m}$ and let $\boldsymbol{y}$ be an n–tuple of nonzero elements of $\mathbb{F}_{q^m}$ 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 $\mathscr{A}_r(\boldsymbol{x}_{\mathcal{I}}, \boldsymbol{y}_{\mathcal{I}})$. □

This shortening trick, together with Proposition 6 (ii) explain that the square of a shortened wild Goppa code of extension degree 2 is contained in an alternant code of non trivial dimension.

Proposition 8. *Let $\mathcal{I} \subseteq \{0, \dots n-1\}$ and $r' \stackrel{def}{=} 2r(q+1) - (n - |\mathcal{I}|) + 1$. Then there exists some $\boldsymbol{y} \in \left(\mathbb{F}_{q^2}^{\times}\right)^{n-|\mathcal{I}|}$ such that:*

$$\mathcal{S}_{\mathcal{I}}\left(\mathscr{C}\right) \star \mathcal{S}_{\mathcal{I}}\left(\mathscr{C}\right) \subseteq \mathscr{A}_{r'}(\boldsymbol{x}_{\mathcal{I}}, \boldsymbol{y}) \tag{4}$$

Proof. By Proposition 6, we know that $\mathscr{C} = \mathscr{G}\left(\boldsymbol{x}, \gamma^{q-1}\right) = \mathscr{G}\left(\boldsymbol{x}, \gamma^{q+1}\right)$ which can therefore be viewed as an alternant code $\mathscr{A}_{r(q+1)}(\boldsymbol{x}, \boldsymbol{y})$ for $\boldsymbol{y} = \gamma(\boldsymbol{x})^{-(q+1)}$. By applying Proposition 7 to it, we know that $\mathcal{S}_{\mathcal{I}}\left(\mathscr{C}\right)$ is an alternant code of degree $r(q+1)$ and length $n - |\mathcal{I}|$. We then finish the proof by applying Theorem 2 to it. □

Let us bring in now the quantities:

$$\begin{aligned}
k(a) &\stackrel{\text{def}}{=} n - a - 2r(q-1) + r(r-2) \\
k_{\text{Alt}}(a) &\stackrel{\text{def}}{=} 3(n-a) - 4r(q+1) - 2 \\
k_{\text{Rand}}(a) &\stackrel{\text{def}}{=} \min\left\{n - a, \binom{k(a)+1}{2}\right\} \\
a^- &\stackrel{\text{def}}{=} n - 2r(q+1) - 1. \\
a^+ &\stackrel{\text{def}}{=} \sup\left\{a \in \{0, k(0) - 1\} \mid k_{\text{Alt}}(a) \leqslant k_{\text{Rand}}(a)\right\}.
\end{aligned}$$

Let $\mathscr{R}$ be a random code of the same length and dimension as $\mathcal{S}_{\mathcal{I}}(\mathscr{C})$. Then for $a = |\mathcal{I}|$, $k(a)$ and $k_{\mathrm{Rand}}(a)$ would be the dimensions we expect for $\mathcal{S}_{\mathcal{I}}(\mathscr{C})$ and $\mathscr{R}^{\star 2}$. The quantity $k(0)$ is the dimension we expect for $\mathscr{C}$. In our experiments we never found a case where the dimensions of $\mathcal{S}_{\mathcal{I}}(\mathscr{C})$ and $\mathscr{A}^{\star 2}$ differ from $k(a)$ and $k_{\mathrm{Rand}}(a)$ respectively. On the other hand, notice that from Proposition 8, $k_{\mathrm{Alt}}(a)$ can be viewed as an upper bound on the dimension of $(\mathcal{S}_{\mathcal{I}}(\mathscr{C}))^{\star 2}$. In other words, as soon as $k_{\mathrm{Alt}}(a) < k_{\mathrm{Rand}}(a)$, we expect to distinguish $\mathcal{S}_{\mathcal{I}}(\mathscr{C})$ from $\mathscr{R}$. It also turns out in our experiments that the observed dimension of $(\mathcal{S}_{\mathcal{I}}(\mathscr{C}))^{\star 2}$ is equal to $k_{\mathrm{Alt}(a)} - 1$ when $k_{\mathrm{Alt}}(a) \leqslant k_{\mathrm{Rand}}(a)$. We can therefore include the a's for which $k_{\mathrm{Alt}}(a) = k_{\mathrm{Rand}}(a)$ in the choices for a for which we distinguish $\mathcal{S}_{\mathcal{I}}(\mathscr{C})$ from $\mathscr{R}$. This motivates to define the *distinguisher interval* as the set of $a \in \{0, \ldots, k(0)-1\}$ such that $k_{\mathrm{Alt}}(a) \leqslant k_{\mathrm{Rand}}(a)$. Finally a^- corresponds to the critical value of a for which $k_{\mathrm{Alt}}(a) = n - a$. It turns out that there is a simple characterization of the distinguisher interval, namely

Proposition 9. *The distinguisher interval is empty if* $\binom{r(r+2)+2}{2} < 2r(q+1)+1$. *On the other hand if* $\binom{r(r+2)+2}{2} \geqslant 2r(q+1)+1$ *and* $a^- \geqslant 0$, *then it is non empty and is an interval of the form* $[a^-, a^+]$.

We checked that this allows to distinguish all the wild Goppa codes of extension degree 2 suggested in [6] from random codes when $r > 3$. For instance, consider the first entry in Table 7.1 in [6] which is a code of this kind. It has length 794, dimension 529, is defined over $\mathbb{F}_{29}$ and is associated to a Goppa polynomial $\gamma(x)^{29}$ where γ has degree 5. Table 1 shows that for a in the range $\{493, \ldots, 506\}$ the dimensions of $(\mathcal{S}_{\mathcal{I}}(\mathscr{C}))^{\star 2}$ differ when $\mathscr{C}$ is the aforementioned wild Goppa code or is a random code with the same parameters. Note that for this example $a^- = 493$. This is a typical behavior and it is only when the degree of γ is very small and the field size is large that we cannot distinguish the Goppa code in this way. More precisely, we have gathered in Table 2 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 r, the degree of γ.

Table 1. Dimension of $(\mathcal{S}_{\mathcal{I}}(\mathscr{C}))^{\star 2}$ when $\mathscr{C}$ is either the aforementioned wild Goppa code or a random code of the same length and dimension for various values of the size of $\mathcal{I}$. We can notice that for all $|\mathcal{I}| \in \{493, \ldots, 506\}$ the dimension of the square of the random code and that of the square of the Goppa code differ.

$\lvert\mathcal{I}\rvert$	493	494	495	496	497	498	499	500	501	502	503	504
Goppa	300	297	294	291	288	285	282	279	276	273	270	267
random	301	300	299	298	297	296	295	294	293	292	291	290

$\lvert\mathcal{I}\rvert$	505	506	507	508	509	510	511	512	513	514
Goppa	264	261	253	231	210	190	171	153	136	120
random	289	276	253	231	210	190	171	153	136	120

Table 2. 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

4 The Code Filtration

4.1 Main tool

We bring in here the crucial ingredient of our attack which is the following family of nested codes defined for any a in $\{0,\dots,n-1\}$:

$$\mathscr{C}^a(0) \supseteq \mathscr{C}^a(1) \supseteq \cdots \mathscr{C}^a(i) \supseteq \mathscr{C}^a(i+1) \supseteq \cdots \supseteq \mathscr{C}^a(q+1).$$

Roughly speaking, $\mathscr{C}^a(j)$ (see Definition 5 below) consists in the codewords of $\mathscr{C}$ which correspond to polynomials which have a zero of order j at position a. Using a common terminology in algebra, we will call this family of nested codes a *filtration*. It turns out that the first two elements of this filtration are just punctured and shortened versions of $\mathscr{C}$ and the rest of them can be computed from $\mathscr{C}$ only by computing star products and solving linear systems. The key point is that this nested family of codes reveals a lot about the algebraic structure of $\mathscr{C}$. In particular, we will be able to recover the support from it. This is a consequence of the following proposition:

Proposition 10. *For all $a \in \{0,\dots,n-1\}$, we have*[4]*:*

$$(\boldsymbol{x}_a - x_a)^{-(q+1)} \star \mathscr{C}^a(q+1) \subseteq \mathcal{P}_a(\mathscr{C}).$$

Without loss of generality, one can assume that the first two entries of $\boldsymbol{x}$ are $x_0 = 0$ and $x_1 = 1$. As explained further, this will in particular make possible the computation of the vectors $\boldsymbol{x}_0^{-(q+1)}$ and $(\boldsymbol{x}_1 - \mathbf{1})^{q+1}$ and we prove further that the knowledge of these two vectors provides that of $\boldsymbol{x}$ up to some Galois action. Let us now define precisely these codes $\mathscr{C}^a(j)$. They are defined for any $a \in \{0,\dots,n-1\}$ and for any integer j as follows:

Definition 5. *For all $a \in \{0,\dots,n-1\}$ and for all $j \in \mathbb{Z}$, we define the code $\mathscr{C}^a(j)$ as:*

$$\mathscr{C}^a(j) \stackrel{def}{=} \left\{ \left(\frac{\gamma^{q+1}(x_i)}{\pi'_{\boldsymbol{x}}(x_i)} (x_i - x_a)^j f(x_i) \right)_{i \in \{0,\dots,n-1\}\setminus\{a\}} \;\middle|\; f \in \mathbb{F}_{q^2}[z]_{<n-r(q+1)-j} \right\} \cap \mathbb{F}_q^{n-1}.$$

The link with $\mathscr{C}$ becomes clearer if we use Proposition 6, which gives that $\mathscr{C} = \mathscr{G}\left(\boldsymbol{x}, \gamma^{q-1}\right) = \mathscr{G}\left(\boldsymbol{x}, \gamma^{q+1}\right)$. Viewing now $\mathscr{C}$ as a subfield subcode of a GRS code, and thanks to Lemma 2, we have:

$$\mathscr{C} = \left\{ \left(\frac{\gamma^{q+1}(x_i) f(x_i)}{\pi'_{\boldsymbol{x}}(x_i)} \right)_{0 \leqslant i < n} \;\middle|\; f \in \mathbb{F}_{q^2}[z]_{<n-r(q+1)} \right\} \cap \mathbb{F}_q^n. \tag{5}$$

From this definition, it is clear that $\mathscr{C}^a(1)$ is $\mathscr{C}$ shortened in a.

4.2 The computation of the filtration

This filtration is strongly related to $\mathscr{C}$ since, as explained in the following statement, its two first elements are respectively obtained by puncturing and shortening $\mathscr{C}$ at a.

Theorem 3. *For all $a \in \{0,\dots,n-1\}$, we have:*

(i) $\mathscr{C}^a(0) = \mathcal{P}_a(\mathscr{C})$;
(ii) $\mathscr{C}^a(1) = \mathcal{S}_a(\mathscr{C})$;
(iii) $\mathscr{C}^a(q-r) = \mathscr{C}^a(q+1)$;
(iv) $\mathscr{C}^a(-r) = \mathscr{C}^a(0)$.

[4] Recall that by $(\boldsymbol{x}_a - x_a)^{-(q+1)}$ we mean the vector $\left((x_i - x_a)^{-(q+1)}\right)_{i \in \{0,\dots,n-1\}\setminus\{a\}}$.

After the computation of the two first elements and for the same reason we need to take shortened versions of the public code to distinguish it from a random code, the rest of the filtration relies in a crucial way on taking star products of shortened versions of the codes $\mathscr{C}^0(s)$ that we denote by $\mathscr{C}^{0,\mathcal{I}}(s)$ which stands for the code $\mathscr{C}^0(s)$ shortened in the positions belonging to $\mathcal{I} \subset \{1, \ldots, n-1\}$. It is readily checked that such a code can be written as:

$$\mathscr{C}^{0,\mathcal{I}}(s) = \left\{ \left(\frac{x_i^s \gamma^{q+1}(x_i) f(x_i)}{\pi'_{\boldsymbol{x}_\mathcal{I}}(x_i)} \right)_{i \in \{1,\ldots,n-1\} \setminus \mathcal{I}} \;\middle|\; f \in \mathbb{F}_{q^2}[z]_{<n-r(q+1)-s-|\mathcal{I}|} \right\} \cap \mathbb{F}_q^{n-1-|\mathcal{I}|} \tag{6}$$

where we recall that $\boldsymbol{x}_\mathcal{I}$ denotes the vector $\boldsymbol{x}$ punctured at $\mathcal{I}$. From this form, it is clear that by applying the equivalent definition of an alternant code given in Definition 2, that we have:

Lemma 3. *For some $\boldsymbol{y} \in (\mathbb{F}_{q^2}^\times)^{n-|\mathcal{I}|-1}$ we have $\mathscr{C}^{0,\mathcal{I}}(s) = \mathscr{A}_{r(q+1)+s-1}(\boldsymbol{x}_{\mathcal{I}\cup\{0\}}, \boldsymbol{y})$.*

Since such codes are alternant codes, a simple consequence of Lemma 3 is:

Proposition 11. *Let $\mathcal{I}$ be a subset of $\{1, \ldots, n-1\}$ and let us define $r(s,t) \stackrel{def}{=} 2r(q+1) + s + t - n + |\mathcal{I}|$ then there exists some $\boldsymbol{y} \in \mathbb{F}_{q^2}^{n-1-|\mathcal{I}|}$ such that:*

$$\mathscr{C}^{0,\mathcal{I}}(s) \star \mathscr{C}^{0,\mathcal{I}}(t) \subseteq \mathscr{A}_{r(s,t)}(\boldsymbol{x}_{\mathcal{I}\cup\{0\}}, \boldsymbol{y}) \tag{7}$$

Proof. This follows at once from Lemma 3 which says that $\mathscr{C}^{0,\mathcal{I}}(s)$ and $\mathscr{C}^{0,\mathcal{I}}(t)$ are alternant codes of respective degrees $r(q+1)+s-1$ and $r(q+1)+t-1$. From Theorem 2, we know that their star product is included in an alternant code with support $\boldsymbol{x}_{\mathcal{I}\cup\{0\}}$ and of degree r' with:

$$r' = r(q+1) + s - 1 + r(q+1) + t - 1 - (n - |\mathcal{I}| - 1) + 1 \;\; = r(s,t)$$

□

This suggests that the product $\mathscr{C}^{0,\mathcal{I}}(s) \star \mathscr{C}^{0,\mathcal{I}}(t)$ might only depend on $s+t$. In order to find $\mathscr{C}^{0,\mathcal{I}}(t)$ once $\mathscr{C}^{0,\mathcal{I}}(0), \mathscr{C}^{0,\mathcal{I}}(1), \ldots, \mathscr{C}^{0,\mathcal{I}}(t-1)$ have been found, we might be tempted to use the "Equation":

$$\mathscr{C}^{0,\mathcal{I}}(0) \star \mathscr{C}^{0,\mathcal{I}}(t) = \mathscr{C}^{0,\mathcal{I}}(\lfloor t/2 \rfloor) \star \mathscr{C}^{0,\mathcal{I}}(\lceil t/2 \rceil).$$

Unfortunately, this equality does not hold in general. However, we have the following related statement.

Lemma 4. *Let $\mathcal{I}$ be a subset of $\{1, \ldots, n-1\}$ such that $r' \stackrel{def}{=} r(\lfloor t/2 \rfloor, \lceil t/2 \rceil) = -n + |\mathcal{I}| + 2r(q+1) + t > 0$. We have:*

(i) Any codeword $\boldsymbol{s}$ in $\mathscr{C}^{0,\mathcal{I}}(t-1)$ such that $\boldsymbol{s} \star \mathscr{C}^{0,\mathcal{I}}(0) \subseteq \mathscr{C}^{0,\mathcal{I}}(\lfloor t/2 \rfloor) \star \mathscr{C}^{0,\mathcal{I}}(\lceil t/2 \rceil)$, necessarily belongs to $\mathscr{C}^{0,\mathcal{I}}(t)$.

(ii) Conversely, $\mathscr{C}^{0,\mathcal{I}}(t)$ is equal to the set of codewords $\boldsymbol{s}$ in $\mathscr{C}^{0,\mathcal{I}}(t-1)$ such that

$$\boldsymbol{s} \star \mathscr{C}^{0,\mathcal{I}}(0) \subseteq \mathscr{A}_{r'}(\boldsymbol{x}_{\mathcal{I}\cup\{0\}}, \boldsymbol{y}).$$

for some $\boldsymbol{y} \in \mathbb{F}_q^{n-1-|\mathcal{I}|}$.

Thus, one expects to find $\mathscr{C}^{0,\mathcal{I}}(t)$ by solving the following problem, which has already been considered in [13].

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

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

Proposition 12. *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 $\boldsymbol{s} \in \mathscr{S}$, $\boldsymbol{a} \in \mathscr{A}$ and $\boldsymbol{b}^\perp \in \mathscr{B}^\perp$. Then $\boldsymbol{s} \in \mathscr{D}$ and $\langle \boldsymbol{s}, \boldsymbol{a} \star \boldsymbol{b}^\perp \rangle = \sum_{i=0}^{n-1} s_i a_i b_i^\perp = \langle \boldsymbol{s} \star \boldsymbol{a}, \boldsymbol{b}^\perp \rangle$ 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 same way. □

This allows to find several of these $\mathscr{C}^{0,\mathcal{I}}(t)$'s associated to different subsets of $\mathcal{I}$. It is straightforward to use such sets in order to recover $\mathscr{C}^0(t)$. Indeed, from the characterization of $\mathscr{C}^{0,\mathcal{I}}(t)$ given in (6) we clearly expect that:

$$\mathscr{C}^{0,\mathcal{I}\cap\mathcal{J}}(t) = \mathscr{C}^{0,\mathcal{J}}(t) + \mathscr{C}^{0,\mathcal{I}}(t) \tag{8}$$

where with an abuse of notation we mean by $\mathscr{C}^{0,\mathcal{J}}(t)$ and $\mathscr{C}^{0,\mathcal{I}}(t)$ the code $\mathscr{C}^{0,\mathcal{J}}(t)$ and $\mathscr{C}^{0,\mathcal{J}}(t)$ whose set of positions has 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 $\dim\left(\mathscr{C}^{0,\mathcal{I}}(t)\right) + \dim\left(\mathscr{C}^{0,\mathcal{J}}(t)\right) \geqslant \dim\left(\mathscr{C}^{0,\mathcal{I}\cap\mathcal{J}}(t)\right)$. These considerations suggest the following Algorithm 1 for computing the $\mathscr{C}^0(t)$'s.

Algorithm 1 Algorithm for computing $\mathscr{C}^0(q+1)$.

for $t = 2$ to $q+1$ **do**
  $\mathscr{C}^0(t) \leftarrow \{0\}$
  while $\dim \mathscr{C}^0(t) \neq k(t)$ **do**
    $\{k(t)$ is obtained "offline" by computing the true dimension of a $\mathscr{C}^0(t)$ for an arbitrary choice of γ and $\boldsymbol{x}$.$\}$
    $\mathcal{I} \leftarrow$ rand. subset of $\{1, \ldots, n-1\}$ of size $a(t)$ {We explain in (11) how $a(t)$ is obtained.}
    $\mathscr{A} \leftarrow \mathscr{C}^{0,\mathcal{I}}(0)$
    $\mathscr{B} \leftarrow \mathscr{C}^{0,\mathcal{I}}(\lfloor \frac{t}{2} \rfloor) \star \mathscr{C}^{0,\mathcal{I}}(\lceil \frac{t}{2} \rceil)$
    $\mathscr{D} \leftarrow \mathscr{C}^{0,\mathcal{I}}(t-1)$
    $\mathscr{C}^{0,\mathcal{I}}(t) \leftarrow \mathscr{D} \cap \left(\mathscr{A} \star \mathscr{B}^\perp\right)^\perp$ {Problem 1.}
    $\mathscr{C}^0(t) \leftarrow \mathscr{C}^0(t) + \mathscr{C}^{0,\mathcal{I}}(t)$
  end while
end for
return $\mathscr{C}^0(q+1)$

In Algorithm 1 it is essential to choose the sizes $a(t)$ of the set of indices $\mathcal{I}$ used to compute $\mathscr{C}^{0,\mathcal{I}}(t)$ appropriately. Let us denote by k the dimension of $\mathscr{C}$ and bring in the quantity:

$$k_{\mathrm{Alt}}(s,t,a) \stackrel{\text{def}}{=} 3(n-a) - 4r(q+1) - 2(s+t) - 1 \tag{9}$$

$$k_{\mathrm{Rand}}(s,t,a) \stackrel{\text{def}}{=} \frac{1}{2}\left(k - 2s - a + 1\right)\left(k - 4t + 2s - a + 2\right) \tag{10}$$

then we choose we choose $a(t)$ such that:

$$a(t) > n - 2r(q+1) - t \tag{11}$$

$$k_{\mathrm{Alt}}(\lceil t/2 \rceil, \lfloor t/2 \rfloor, a(t)) < k_{\mathrm{Rand}}(\lceil t/2 \rceil, \lfloor t/2 \rfloor, a(t)) \tag{12}$$

The reasons for this choice are explained in a forthcoming long version of this paper.

5 An Efficient Attack Using the Distinguisher

The attack consists in 5 steps which are outlined below.

Step 1. Compute $\mathscr{C}^0(q+1)$ and $\mathscr{C}^1(q+1)$ using the distinguisher–based methods developed in Section 4. Thanks to Theorem 3(iii), it is sufficient to compute $\mathscr{C}^0(q-r)$ and $\mathscr{C}^1(q-r)$.

Step 2. From $\mathscr{C}^0(q+1)$ and $\mathscr{C}^1(q+1)$ respectively, we compute two sets of vectors in $\mathbb{F}_q^{n-1}$ which are the respective solution sets of the systems:

$$(S_0): \begin{cases} \boldsymbol{z} \star \mathscr{C}^0(q+1) \subseteq \mathcal{P}_0(\mathscr{C}) \\ \forall i \in \{0 \dots n-2\},\ z_i \neq 0 \\ z_0 = 1 \end{cases} \quad \text{and} \quad (S_1): \begin{cases} \boldsymbol{z} \star \mathscr{C}^1(q+1) \subseteq \mathcal{P}_1(\mathscr{C}) \\ \forall i \in \{0 \dots n-2\},\ z_i \neq 0 \\ z_0 = 1 \end{cases} \tag{13}$$

From Proposition 10, $\boldsymbol{x}_0^{-(q+1)}$ is a solution of (S_0) and $(\boldsymbol{x}_1 - 1)^{-(q+1)}$ is a solution of (S_1). In addition, from Proposition 12, the sets of solutions of the above systems are the respective full-weight codewords whose first entry is 1 of the following codes:

$$\mathscr{D} \stackrel{\text{def}}{=} \left(\mathscr{C}^0(q+1) \star (\mathcal{P}_0(\mathscr{C}))^{\perp}\right)^{\perp} \quad \text{and} \quad \mathscr{D}' \stackrel{\text{def}}{=} \left(\mathscr{C}^1(q+1) \star (\mathcal{P}_1(\mathscr{C}))^{\perp}\right)^{\perp}. \tag{14}$$

Experimentally we found out that $\mathscr{D}$ $\mathscr{D}'$ have dimension 4. A heuristic explaining this observation is given in the forthcoming full version of this paper. Therefore an exhaustive search can be performed to find the full-weight codewords. In addition, we have a complete description of these sets.

Proposition 13. *There are at least $q^2 - n + 2$ solutions for (S_0) which are $\mathbf{1}$, $\boldsymbol{x}_0^{-(q+1)}$ and the following vectors $(1-a)^{-(q+1)}\left((\boldsymbol{x}_0 - a)^{q+1} \star \boldsymbol{x}_0^{-(q+1)}\right)$ obtained with $a \in \mathbb{F}_{q^2} \setminus \mathcal{L}_{\boldsymbol{x}}$. Similarly, there at least $q^2 - n + 2$ solutions for (S_1) which are $\mathbf{1}$, $(\boldsymbol{x}_1 - 1)^{-(q+1)}$ and the vectors $a^{-(q+1)}\left((\boldsymbol{x}_1 - a)^{q+1} \star (\boldsymbol{x}_1 - 1)^{-(q+1)}\right)$ also obtained with $a \in \mathbb{F}_{q^2} \setminus \mathcal{L}_{\boldsymbol{x}}$:*

Remark 3. It is possible to give a lower-bound for the probability P that (S_0) (and (S_1)) has no other solution:

$$P \geqslant 1 - (q^3+q)\frac{(q^2-n)!}{(q^2-n-q)!} \cdot \frac{(q^2-q)!}{q^2!}.$$

In [6, Table 7.1], the authors propose a code over $\mathbb{F}_{32}$, with $m = 2, t = 4$ of length 841. For such parameters, the above probability is lower than $3.52\ 10^{-21}$. Table 3 summarizes this probability for other parameters proposed in [6] for $m = 2$ and $t > 3$.

$q = 29$, $n = 791$	$q = 31$, $n = 892$	$q = 31$, $n = 851$	$q = 31$, $n = 813$	$q = 31$, $n = 795$
$3.6\ 10^{-36}$	$5.5\ 10^{-35}$	$3\ 10^{-27}$	$1.08\ 10^{-22}$	$5.6\ 10^{-21}$

Table 3. Estimates of $1 - P$, where P denotes the probability of Remark 3, for some explicit parameters.

Step 3. First, notice that the vectors $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$ punctured at the first position are both equal to the vector $\boldsymbol{x}_{01} \stackrel{\text{def}}{=} (x_2, \dots, x_{n-1}) \in \mathbb{F}_{q^2}^{n-2}$. From the previous step, one can obtain the two following sets of vectors:

$$\begin{aligned} L_0 &\stackrel{\text{def}}{=} \left\{\boldsymbol{x}_{01}^{q+1}\right\} \cup \left\{ (1-a)^{q+1}\left(\boldsymbol{x}_{01}^{q+1} \star (\boldsymbol{x}_{01} - a)^{-(q+1)}\right) \;\middle|\; a \in \mathbb{F}_{q^2} \setminus \mathcal{L}_{\boldsymbol{x}} \right\} \\ L_1 &\stackrel{\text{def}}{=} \left\{(\boldsymbol{x}_{01} - 1)^{q+1}\right\} \cup \left\{ a^{q+1}\left((\boldsymbol{x}_{01} - 1)^{q+1} \star (\boldsymbol{x}_{01} - a)^{-(q+1)}\right) \;\middle|\; a \in \mathbb{F}_{q^2} \setminus \mathcal{L}_{\boldsymbol{x}} \right\}. \end{aligned} \tag{15}$$

They are computed by puncturing the first entry of each solution vector and taking the inverse for the star product. Note that the trivial solution $\mathbf{1}$ is always removed. The problem now is to identify $\boldsymbol{x}_0^{q+1}$ and $(\boldsymbol{x}_1 - 1)^{q+1}$ among them.

Proposition 14. *If $n > 2q + 4$, then there exists a one-to-one map $\phi : L_0 \to L_1$ such that $\phi(\boldsymbol{x}_{01}^{q+1}) = (\boldsymbol{x}_{01} - 1)^{q+1}$ and for all $\boldsymbol{s} \in L_0$, the vector $\phi(\boldsymbol{s})$ is the unique element of L_1 such that every element of $\boldsymbol{s} \star L_1$ is collinear to a unique element of $\phi(\boldsymbol{s}) \star L_0$.*

The end of the attack works as follows: for $\boldsymbol{s}_0 \in L_0$, compute $\boldsymbol{s}_1 = \phi(\boldsymbol{s}_0)$, then apply Steps 4 of the attack. If $\boldsymbol{s}_0 \neq \boldsymbol{x}_0^{q+1}$, then the Final Step will fail to find a nontrivial solution. In such situation, choose another $\boldsymbol{s}_0 \in L_0$. Therefore, in the worst case, Step 4 and Final Step will be iterated $|L_0| = q^2 - n + 1$ times.

Step 4. This step is better explained when we have a valid $(\boldsymbol{s}_0, \boldsymbol{s}_1) = \left(\boldsymbol{x}_0^{q+1}, (\boldsymbol{x}_1 - 1)^{q+1}\right)$. Recall that $\mathrm{N}(t) = t^{q+1}$ is the norm of t over $\mathbb{F}_q$ for all $t \in \mathbb{F}_{q^2}$. The following lemma shows that the minimal polynomial $P_{x_i} \in \mathbb{F}_q[z]$ of x_i can be computed using: if $\mathrm{N}(x_i)$ and $\mathrm{N}(x_i - 1)$ are known:

Lemma 5. *Let t be an element of $\mathbb{F}_{q^2}$ and $P_t(z) \stackrel{def}{=} z^2 - (N(t) - N(t-1) - 1)z + N(t)$. Then, either P_t is irreducible and is the minimal polynomial of t over $\mathbb{F}_q$, or P_t is reducible and in this case $P_t(z) = (z - t)^2$.*

Proof. First, notice that $\mathrm{N}(t-1) = (t-1)(t^q - 1) = t^{q+1} - t^q - t + 1 = \mathrm{N}_{\mathbb{F}_{q^2}/\mathbb{F}_q}(t) - \mathrm{Tr}_{\mathbb{F}_{q^2}/\mathbb{F}_q}(t) + 1$. Therefore, $P_t(z) = z^2 - \mathrm{Tr}(z) + \mathrm{N}(z)$, 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 $(z - t)^2$.

Final Step. For the sake of simplicity we will assume in what follows that $\boldsymbol{x}$ is full, that is to say that $n = q^2$. Since the support $\boldsymbol{x}$ is known up to Galois action, after applying some permutation to $\mathscr{C}$, one can assume that

- the q first entries of $\boldsymbol{x}$ are the elements of $\mathbb{F}_q$;
- in the $q^2 - q$ remaining entries, two conjugated elements a, a^q of $\mathbb{F}_{q^2} \setminus \mathbb{F}_q$ are consecutive entries in $\boldsymbol{x}$.

Next, we compute a vector $\boldsymbol{x}' \in \mathbb{F}_{q^2}^{q^2}$ such that for all $0 \leqslant i < q^2$, the minimal polynomial of x'_i equals that of x_i. Thus, $\boldsymbol{x}'$ is the image of $\boldsymbol{x}$ by a product of transpositions with pairwise disjoint supports. Moreover, the possible supports for these transpositions are pairs $(i, i+1)$ such that $x_i^q = x_{i+1}$. We denote by τ this permutation. Its matrix is of the form

$$\mathbf{R}_\tau \stackrel{\text{def}}{=} \begin{pmatrix} \mathbf{I}_q & (0) \\ (0) & \mathbf{B} \end{pmatrix}, \tag{16}$$

where $\mathbf{B} \in \mathfrak{M}_{q^2-q}(\mathbb{F}_q)$ is 2×2–block diagonal with blocks of the form $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ or $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$.

From Proposition 6(iii), $\mathscr{G}\left(\boldsymbol{x}', \gamma^{q+1}\right) = \boldsymbol{u} \star \mathscr{A}_{r(q+1)}(\boldsymbol{x}', \mathbf{1})$ for some vector $\boldsymbol{u}$ with no zero entry. Therefore, if we denote by $\mathbf{D}_{\boldsymbol{u}}$ the diagonal matrix whose diagonal entries are those of $\boldsymbol{u}$, we see that

$$\mathscr{C}\mathbf{R}_\tau\mathbf{D}_{\boldsymbol{u}} = \mathscr{A}_{r(q+1)}(\boldsymbol{x}', \mathbf{1}). \tag{17}$$

Thus, since $\boldsymbol{x}'$ is known, we can recover τ and $\boldsymbol{u}$ by solving

Problem 2. *Compute the space of matrices $\mathbf{M}$ of the form*

$$\mathbf{M} = \begin{pmatrix} \mathbf{E} & (0) \\ (0) & \mathbf{F} \end{pmatrix}$$

such that $\mathbf{E}$ is diagonal, $\mathbf{F}$ si 2×2 blockdiagonal and which satisfy

$$\mathscr{C}\mathbf{M} \subseteq \mathscr{A}_{r(q+1)}(\boldsymbol{x}', \mathbf{1}). \tag{18}$$

The solution space is computed by solving a linear system whose unknowns are the entries of $\mathbf{M}$. Since $\mathbf{M}$ is block diagonal, the number of unknowns is linear in n while the number of equations is $\dim(\mathscr{C}) \times \dim(\mathscr{A}_{r(q+1)}(\boldsymbol{x}', \mathbf{1})^\perp) = k(n-k)$ and hence is quadratic in n. Therefore, the solution space will have a very low dimension. Experimentally, this dimension is observed to be 2. Therefore, by exhaustive search in this low-dimensional solution space one finds easily a matrix $\mathbf{M}$ of the form $\mathbf{RD}$, where $\mathbf{R}$ is a permutation matrix and $\mathbf{D}$ is invertible and diagonal. This yields $\boldsymbol{u}$ and τ and hence $\boldsymbol{x}$ and the description of $\mathscr{C}$ as an alternant code.

Algorithm 2 Algorithm of the attack.

Compute $\mathscr{C}^0(q+1)$, $\mathscr{C}^1(q+1)$ using Algorithm 1.
$L_0 \leftarrow$ List of candidates for $\boldsymbol{x}_0^{q+1}$ (Obtained by solving System (S_0) in (13))
$L_1 \leftarrow$ List of candidates for $(\boldsymbol{x}_1 - 1)^{q+1}$ (Obtained by solving System (S_1) in (13))
$\boldsymbol{M}_0 \leftarrow 0$
while $\boldsymbol{M}_0 = 0$ and $L_0 \neq \emptyset$ **do**
  Pick a random element $\boldsymbol{a}_0$ in L_0.
  $\boldsymbol{a}_1 \leftarrow \phi(\boldsymbol{a}_0)$ {where ϕ is the map obtained thanks to Proposition 14}
  $L_0 \leftarrow L_0 \setminus \{\boldsymbol{a}_0\}$
  $L_1 \leftarrow L_1 \setminus \{\boldsymbol{a}_1\}$
  Compute the minimal polynomials P_{x_i} of the positions using Lemma 5.
  Compute, an arbitrary vector $\boldsymbol{x}'$ as explained in Final Step.
  $V \leftarrow$ Space of solutions of Problem 2.
  if $\dim V > 0$ and $\exists \boldsymbol{M} \in V$ of the form **RD** as in Final Step **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 described in Final Step.
  return $\boldsymbol{x}, \boldsymbol{u}$
end if

6 Improvement of the Attack

For some parameters, the computation of the filtration up to $\mathscr{C}^a(q+1)$ or actually up to $\mathscr{C}^a(q-r)$ (thanks to Theorem 3 (iii)) is not possible, while it is still possible to compute the filtration up to $\mathscr{C}^a(q+1-s)$ for some s satisfying $r+1 < s \leqslant (q+1)/2$. This for instance what happens for codes over $\mathbb{F}_{32}$, with $t = 4$. In such situation, $\mathscr{C}^a(s)$ is known since by assumption $s \leqslant q+1-s$ and then, we can compute $\mathscr{C}^a(-s)$ from the knowledge of $\mathcal{P}_a(\mathscr{C})$ and $\mathscr{C}^a(s)$. This computation consists in solving a problem very similar to Problem 1. Then, as a generalization of Proposition 10, we have $(\boldsymbol{x}_a - x_a)^{-(q+1)} \star \mathscr{C}^a(q+1-s) \subseteq \mathscr{C}^a(-s)$ and the rest of the attack runs in a very same manner.

7 Complexity and Implementation

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

7.1 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 $\mathcal{O}(nab)$ operations. Then, the Gaussian elimination costs $\mathcal{O}(nab \min(n, ab))$. 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 $\mathcal{O}(n^2k^2)$. Thanks to Proposition 12, one shows that the dominant part of the resolution of Problem 1, consists in computing $\mathscr{A} \star \mathscr{B}^{\perp}$ and hence costs $\mathcal{O}(na(n-b)\min(n, a(n-b)))$

7.2 Computation of the filtration

Let us first evaluate the cost of computing $\mathscr{C}^{a,\mathcal{I}}(s+1)$ from $\mathscr{C}^{a,\mathcal{I}}(s)$. Equations (9) to (12) suggest that the dimension of $\mathscr{C}^{a,\mathcal{I}}(s)$ used to compute the filtration is in $\mathcal{O}(\sqrt{n})$. From §7.1, the computation of the square of $\mathscr{C}^{a,\mathcal{I}}(s)$ costs $\mathcal{O}(n^3)$ operations in $\mathbb{F}_q$. Then, the resolution of Problem 12

in the context of Lemma 4, costs $\mathcal{O}(na(n-b)\min(n, a(n-b)))$, where $a = \dim \mathscr{C}^{a,\mathcal{I}}(s) = \mathcal{O}(\sqrt{n})$ and $b = \dim \mathscr{A}_{r'}(\boldsymbol{x}_{\mathcal{I}\cup\{0\}}, \boldsymbol{y})$. We have $n - b = \mathcal{O}(n)$, hence we get a cost of $\mathcal{O}(n^3\sqrt{n})$.

The heuristic below Proposition 12, suggests that we need to perform this computation for $\mathcal{O}(\sqrt{n})$ choices of $\mathcal{I}$. Since addition of codes is negligible compared to $\mathcal{O}(n^3\sqrt{n})$ this leads to a total cost of $\mathcal{O}(n^4)$ for the computation of $\mathscr{C}^a(s+1)$. This computation should be done $q+1$ times (actually $q-r$ times from Theorem 3 (iii)) and, we assumed that $q = \mathcal{O}(\sqrt{n})$. Thus, the computation of $\mathscr{C}^a(q+1)$ costs $\mathcal{O}(n^4\sqrt{n})$.

7.3 Other computations

The resolution of Problems (13) in Step 2, costs $\mathcal{O}(n^4)$ (see (14)). Since the solution spaces $\mathscr{D}$ and $\mathscr{D}'$ in (14) have $\mathbb{F}_q$–dimension 4, the exhaustive search in them costs $\mathcal{O}(q^4) = \mathcal{O}(n^2)$ which is negligible. The computation of the map ϕ and that of minimal polynomials is also negligible. Finally, the resolution of Problem 2 costs $\mathcal{O}(n^4)$ since it is very similar to Problem 1. Since 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 $\mathcal{O}(n^5)$. Thus, the global complexity of the attack is in $\mathcal{O}(n^5)$ operations in $\mathbb{F}_q$.

7.4 Implementation

This attack has been implemented with Magma [9] and run over random examples of codes corresponding to the seven entries [6, 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)	(31, 795, 563, 4)
Average time	16min	19.5min	15.5min	31.5min

(q, n, k, r)	(31,813, 581,4)	(31, 851, 619, 4)	(32,841,601,4)
Average time	31.5min	27.2min	49.5min

Remark 4. In the above table the code dimensions are not the ones mentioned in [6]. What happens here is that the formula for the dimension given [6, 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 Proposition 6 (ii).

All these parameters are given in [6] with a 128-bit security that is measured against information set decoding attack which is described in [6, 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 [6] by the biohazard symbol ☣ (together with about two dozens other parameters). This corresponds, as explained in [6], 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 [6] 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 [42], when applied to permutation equivalent codes, generally finds in polynomial time a permutation that sends one code onto the other. The authors of [6] call this requirement on the length the *second defense* and write [6, 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* $\{a_1, \ldots, a_n\}$ [5] *simultaneously, and we would not be surprised if the answer turns out to be yes.*" The authors also add in [6, p.154,§1] that

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

"the security of these cases [6] *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 [6] 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 Proposition 6, 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 [41] is also high (about 2^{125} for such parameters).

8 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 [38] was successfully attacked in [44]. 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 [29] by this operation whereas GRS codes behave differently from random codes with respect to this criterion. We provide 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 $\mathbb{F}_{q^2}$.

It could be argued that this 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 Proposition 6 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 nested family 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 nested family and from here the algebraic description of the Goppa code could be recovered. The crucial point here is really the existence of such a nested family 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 [17] 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.

References

1. 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].
2. M. Baldi, M. Bodrato, and F. Chiaraluce. A new analysis of the McEliece cryptosystem based on QC-LDPC codes. In *Proceedings of the 6th international conference on Security and Cryptography for Networks*, SCN '08, pages 246–262, Berlin, Heidelberg, 2008. Springer-Verlag.
3. A. Becker, A. Joux, A. May, and A. Meurer. Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 = 0$ improves information set decoding. In *EUROCRYPT*, pages 520–536, 2012.

[6] meaning here the cases marked with ☣.

4. T. P. Berger and P. Loidreau. How to mask the structure of codes for a cryptographic use. *Des. Codes Cryptogr.*, 35(1):63–79, 2005.
5. D. J. Bernstein, T. Lange, and C. Peters. Attacking and defending the McEliece cryptosystem. In *PQCrypto*, volume 5299 of *Lecture Notes in Comput. Sci.*, pages 31–46, 2008.
6. D. J. Bernstein, T. Lange, and C. Peters. Wild McEliece. In *Selected Areas in Cryptography*, pages 143–158, 2010.
7. D. J. Bernstein, T. Lange, and C. Peters. Wild McEliece incognito. In *PQCrypto*, pages 244–254, 2011.
8. A. Bogdanov and C.H. Lee. Homorphic encryption from codes. In *Proceedings of the 44th ACM Symposium on Theory of Computing (STOC) (to appear)*, 2012.
9. W. Bosma, J. J. Cannon, and Catherine Playoust. The Magma algebra system I: The user language. *J. Symbolic Comput.*, 24(3/4):235–265, 1997.
10. A. Canteaut and F. Chabaud. A new algorithm for finding minimum-weight words in a linear code: Application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511. *IEEE Trans. Inform. Theory*, 44(1):367–378, 1998.
11. I. Cascudo, H. Chen, R. Cramer, and C. Xing. Asymptotically Good Ideal Linear Secret Sharing with Strong Multiplication over Any Fixed Finite Field. In S. Halevi, editor, *Advances in Cryptology - CRYPTO 2009*, volume 5677 of *Lecture Notes in Comput. Sci.*, pages 466–486. Springer Berlin / Heidelberg, 2009.
12. I. Cascudo, R. Cramer, and C. Xing. The Torsion-Limit for Algebraic Function Fields and Its Application to Arithmetic Secret Sharing. In P. Rogaway, editor, *Advances in Cryptology - CRYPTO 2011*, volume 6841 of *Lecture Notes in Comput. Sci.*, pages 685–705. Springer Berlin / Heidelberg, 2011.
13. A. Couvreur, P. Gaborit, V. Gauthier-Umaña, A. Otmani, and J.-P. Tillich. Distinguisher-based attacks on public-key cryptosystems using Reed-Solomon codes, 2014. ArXiv:1307.6458. To appear in Des. Codes Cryptogr.
14. A. Couvreur, I. Márquez-Corbella, and R. Pellikaan. A polynomial time attack against algebraic geometry code based public key cryptosystems. ArXiv:1401.6025, January 2014.
15. A. Couvreur, A. Otmani, and J.-P. Tillich. New identities relating Wild Goppa codes. ArXiv:1310.3202v2, 2013.
16. J.-C. Faugère, V. Gauthier-Umaña, A. Otmani, L. Perret, and J.-P. Tillich. A distinguisher for high rate McEliece cryptosystems. In *Proceedings of the Information Theory Workshop 2011, ITW 2011*, pages 282–286, Paraty, Brasil, 2011.
17. J.-C. Faugère, V. Gauthier-Umaña, A. Otmani, L. Perret, and J.-P. Tillich. A distinguisher for high rate McEliece cryptosystems. *IEEE Trans. Inform. Theory*, 59(10):6830–6844, 2013.
18. J.-C. Faugère, A. Otmani, L. Perret, and J.-P. Tillich. Algebraic cryptanalysis of McEliece variants with compact keys. In *EUROCRYPT*, pages 279–298, 2010.
19. 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*, pages 99–107, Pamporovo, Bulgaria, June 2008.
20. V. Gauthier-Umaña and G. Leander. Practical key recovery attacks on two McEliece variants, 2009. IACR Cryptology ePrint Archive 509.
21. J. Gibson. Equivalent Goppa codes and trapdoors to McEliece's public key cryptosystem. In Donald Davies, editor, *Advances in Cryptology – EUROCRYPT'91*, volume 547 of *Lecture Notes in Comput. Sci.*, pages 517–521. Springer Berlin / Heidelberg, 1991.
22. H. Janwa and O. Moreno. McEliece public key cryptosystems using algebraic-geometric codes. *Des. Codes Cryptogr.*, 8(3):293–307, 1996.
23. R. Kötter. A unified description of an error locating procedure for linear codes. In *Proc. Algebraic and Combinatorial Coding Theory*, pages 113–117, Voneshta Voda, 1992.
24. G. Landais and J.-P. Tillich. An efficient attack of a McEliece cryptosystem variant based on convolutional codes. In P. Gaborit, editor, *PQCrypto 2013*, volume 7932 of *Lecture Notes in Comput. Sci.*, pages 102–117. Springer, June 2013.
25. P. J. Lee and E. F. Brickell. An observation on the security of McEliece's public-key cryptosystem. In *Advances in Cryptology - EUROCRYPT'88*, volume 330/1988 of *Lecture Notes in Comput. Sci.*, pages 275–280. Springer, 1988.
26. J. S. Leon. A probabilistic algorithm for computing minimum weights of large error-correcting codes. *IEEE Trans. Inform. Theory*, 34(5):1354–1359, 1988.
27. P. Loidreau and N. Sendrier. Weak keys in the McEliece public-key cryptosystem. *IEEE Trans. Inform. Theory*, 47(3):1207–1211, 2001.

28. C. Löndahl and T. Johansson. A new version of McEliece PKC based on convolutional codes. In *ICICS*, pages 461–470, 2012.
29. F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*. North–Holland, Amsterdam, fifth edition, 1986.
30. I. Márquez-Corbella, E. Martínez-Moro, and R. Pellikaan. Evaluation of public-key cryptosystems based on algebraic geometry codes. In J. Borges and M. Villanueva, editors, *Proceedings of the Third International Castle Meeting on Coding Theory and Applications*, pages 199–204, Barcelona, Spain, September 11-15 2011.
31. 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.*, pages 1–17, 2012.
32. I. Márquez-Corbella, E. Martínez-Moro, and R. Pellikaan. On the unique representation of very strong algebraic geometry codes. *Des. Codes Cryptogr.*, pages 1–16, 2012. In Press.
33. I. Márquez-Corbella and R. Pellikaan. Error-correcting pairs for a public-key cryptosystem. Preprint, 2012.
34. A. May, A. Meurer, and E. Thomae. Decoding random linear codes in $\tilde{O}(2^{0.054n})$. In *ASIACRYPT*, pages 107–124, 2011.
35. R. J. McEliece. *A Public-Key System Based on Algebraic Coding Theory*, pages 114–116. Jet Propulsion Lab, 1978. DSN Progress Report 44.
36. L. Minder and A. Shokrollahi. Cryptanalysis of the Sidelnikov cryptosystem. In *EUROCRYPT 2007*, volume 4515 of *Lecture Notes in Comput. Sci.*, pages 347–360, Barcelona, Spain, 2007.
37. R. Misoczki, J.-P. Tillich, N. Sendrier, and P. S. L. M. Barreto. MDPC-McEliece: New McEliece variants from moderate density parity-check codes. *IACR Cryptology ePrint Archive*, 2012:409, 2012.
38. H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Problems of Control and Information Theory*, 15(2):159–166, 1986.
39. 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*, 3(2):129–140, January 2010.
40. R. Pellikaan. On decoding by error location and dependent sets of error positions. *Discrete Math.*, 106-107:368–381, 1992.
41. C. Peters. Information-set decoding for linear codes over $\mathbf{F}_q$. In *PQCrypto*, volume 6061 of *Lecture Notes in Comput. Sci.*, pages 81–94. Springer, 2010.
42. N. Sendrier. Finding the permutation between equivalent linear codes: The support splitting algorithm. *IEEE Trans. Inform. Theory*, 46(4):1193–1203, 2000.
43. V.M. Sidelnikov. A public-key cryptosytem based on Reed-Muller codes. *Discrete Math. Appl.*, 4(3):191–207, 1994.
44. V.M. Sidelnikov and S.O. Shestakov. On the insecurity of cryptosystems based on generalized Reed-Solomon codes. *Discrete Math. Appl.*, 1(4):439–444, 1992.
45. J. Stern. A method for finding codewords of small weight. In G. D. Cohen and J. Wolfmann, editors, *Coding Theory and Applications*, volume 388 of *Lecture Notes in Comput. Sci.*, pages 106–113. Springer, 1988.
46. 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*, 22:518–526, 1976.
47. C. Wieschebrink. Two NP-complete problems in coding theory with an application in code based cryptography. In *Information Theory, 2006 IEEE International Symposium on*, pages 1733–1737, 2006.
48. C. Wieschebrink. Cryptanalysis of the Niederreiter public key scheme based on GRS subcodes. In *PQCrypto*, pages 61–72, 2010.