# On the security of Some Compact Keys for McEliece Scheme

Élise Barelli[*]

INRIA Saclay and LIX, CNRS UMR 7161 École Polytechnique,
91120 Palaiseau Cedex
email: elise.barelli@inria.fr

July 31, 2021

## Abstract

In this paper we study the security of the key of compact McEliece schemes based on alternant/Goppa codes with a non-trivial permutation group, in particular quasi-cyclic alternant codes. We show that it is possible to reduce the key-recovery problem on the original quasi-cyclic code to the same problem on a smaller code derived from the public key. This result is obtained thanks to the *invariant* code operation which gives the subcode whose elements are fixed by a permutation $\sigma \in \mathrm{Perm}(\mathcal{C})$. The fundamental advantage is that the invariant subcode of an alternant code is an alternant code. This approach improves the technique of Faugère, Otmani, Tillich, Perret and Portzamparc which uses *folded* codes of alternant codes obtained by using supports globally stable by an affine map. We use a simpler approach with a unified view on quasi-cyclic alternant codes and we treat the case of automorphisms arising from a non affine homography. In addition, we provide an efficient algorithm to recover the full structure of the alternant code from the structure of the invariant code.

## 1 Introduction

In 1978, McEliece [16] introduced a public key encryption scheme based on linear codes and suggested to use classical Goppa codes which belong to the family of alternant codes. This proposition still remains secure but leads to very large public keys compared to other public-key cryptosystems. That is why, in despite of its fast encryption and decryption, McEliece scheme is limited for practical applications. To overcome this limitation, lot of activity devote to decrease the key size by choosing codes which admit a very compact public matrix. For instance, quasi-cyclic (QC) codes enable to build public key encryption schemes with short keys [11, 3]. These first papers were followed by proposals using alternant and Goppa codes with different automorphism groups like quasi-dyadic (QD) Goppa codes [17].

The cryptanalisis of code-based schemes can be split in two categories: the message recovery attacks and the key-recovery attacks. The first kind of attacks consists in generic methods for decoding a random linear code. These methods are known as Information Set Decoding (ISD) methods. The second kind of attacks consists in recovering the secret elements of the code used in the scheme. In this case, the methods are specific to the code family. Recently, in the category of key-recovery attacks, new methods appeared, known as *algebraic attacks*. This method consists in recovering the secret element of an alternant code by solving a system of

---

polynomial equations. In [9], the authors improved this new method to attack QC and QD alternant codes. Such attacks use the specific structure of QC/QD codes in order to build an algebraic system with much fewer unknowns compared to the generic case. A new approach has been used in [8, 7] to explain that the reduction of the number of unknowns in the algebraic system comes from a smaller code easily computable from the public generator matrix. This smaller code can be obtained by summing up the codewords which belong to the same orbit under the action of the permutation group and is referred to as the *folded* code. (We advertise the reader that the folded codes referenced in [13] are not the same codes as in this paper.) A relation between the support and multiplier defining the alternant code and those of the folded code exists and is sufficient to find the original alternant code. This relation comes from the structure of the folded code: [7] shows that the folding operation preserves the structure of the dual code. That is, the folding of the dual of an alternant code (resp. a Goppa code) is the dual of an alternant code (resp. a Goppa code).

In [7], the authors attack only codes with an automorphism induced by an affine transformation acting on the support and the multiplier, we call them *affine induced* automorphims. Another kind of quasi-cyclic alternant codes can be built from the action of the projective linear group on the support and multiplier. We use in this paper another alternant code built from the public generator matrix of a quasi-cyclic alternant code induced by a projective linear transformation, called the *invariant* code and introduced by Loidreau in [14]. This invariant code can be built easily from the public generator matrix of the alternant code $\mathcal{C}$ since it is the kernel of the linear map: $c \in \mathcal{C} \mapsto c - \sigma(c)$, where $\sigma$ is a permutation of $\mathcal{C}$. We remark also that the folded code used by [7] is included in the invariant code with equality when the characteristic of the field not divide the order of the permutation.

Our main contribution is to consider more general tools coming from algebraic geometry and use the invariant code instead of folded code. This approach has two advantages. First the geometric point of view simplifies the attack by giving a unified view of quasi-cyclic alternant codes. It also simplifies some proofs and enables to consider alternant codes as algebraic geometric codes on the projective line. This method allows us to treat the general case of projective linear transformations. The second advantage is that the invariant code operation is applied directly on the alternant code and not on the dual code. More precisely, we prove the following results.

**Theorem 3.5.** *Let* $\mathrm{GRS}_k(x,y) \subset \mathbb{F}_{q^m}^n$ *be a quasi-cyclic GRS code, and* $\sigma \in \mathfrak{S}_n$ *of order* $\ell$, *such that* $\ell|n$, *the permutation acting on the code* $\mathrm{GRS}_k(x,y)$. *Then the invariant code* $\mathrm{GRS}_k(x,y)^\sigma$ *is a GRS code of length* $n/\ell$ *and dimension* $\lfloor k/\ell \rfloor$.

**Corollary 3.6.** *Let* $\mathcal{A}_r(x,y) \cap \mathbb{F}_q^n$ *be a quasi-cyclic alternant code, and* $\sigma \in \mathfrak{S}_n$ *of order* $\ell$, *such that* $\ell|n$, *the permutation acting on the code* $\mathcal{A}_r(x,y)$. *Then the invariant code* $\mathcal{A}_r(x,y)^\sigma$ *is an alternant code of length* $n/\ell$ *and order* $r/\ell$.

In the last section we show that this means that **the key security of compact McEliece scheme based on alternant codes with some induced permutation reduces to the key security of the invariant code**, which has smaller parameters. We provide an algorithm to recover the secret elements of the alternant code from the knowledge of the its invariant code. This algorithm uses only linear algebra.

However we can notice that key-recovery is generally more expensive than message recovery. With a good choice of parameters it is still possible to construct quasi-cyclic codes with high complexity of key recovery attack on the invariant code.

## 2 Quasi-cyclic Alternant Codes

In this section, we introduce some notation about alternant codes. We denote by $\mathbb{F}_q$ the finite field with $q$ elements, where $q$ is a power of a prime $p$.

Let $x = (x_1, \ldots, x_n)$ be an $n$-tuple of distinct elements of $\mathbb{F}_q$, and $y = (y_1, \ldots, y_n)$ be an $n$-tuple of nonzero elements of $\mathbb{F}_q$. The generalized Reed-Solomon (GRS) code of dimension $k$, denoted $\mathrm{GRS}_k(x, y)$, consists of vectors $(y_1 f(x_1), \ldots, y_n f(x_n))$ where $f$ ranges over all polynomials of degree $< k$, with coefficients in $\mathbb{F}_q$. The vector $x$ is called the *support* and $y$ a *multiplier* of the code $\mathrm{GRS}_k(x, y)$. In order to define alternant codes, we use the following property whose proof can be found in [15, Chap. 12].

**Proposition 2.1.** *The dual of $GRS_k(x, y)$ is $GRS_{n-k}(x, y^{\perp})$ for some $y^{\perp} \in (\mathbb{F}_q \setminus \{0\})^n$.*

**Definition 1.** Let $m$ be a positive integer, $x$ be an $n$-tuple of distinct elements of $\mathbb{F}_{q^m}$ and $y$ be an $n$-tuple of nonzero elements of $\mathbb{F}_{q^m}$. The alternant code $\mathcal{A}_k(x, y)$ over $\mathbb{F}_q$ is the subfield subcode of $\mathrm{GRS}_k(x, y)^{\perp}$, i.e. $\mathcal{A}_k(x, y) := \mathrm{GRS}_k(x, y)^{\perp} \cap \mathbb{F}_q^n$.

### 2.1 Representation of $\mathcal{A}_k(x, y)$ as a subfield subcode of an AG code

For the rest of our work, it is convenient to use a projective representation of alternant codes. This is possible thanks to algebraic geometric codes introduced by Goppa in [12]. To avoid the confusion with classical Goppa codes which are specific alternant codes, we refer to algebraic geometric codes as AG codes. Any GRS code is an AG code on $\mathbb{P}^1$, the projective line over $\mathbb{F}_{q^m}$. Recall some definitions in this case (cf [10, 19] for further details).

For brevity we denote by $\mathbb{P}^1$ the projective line over $\mathbb{F}_{q^m}$. We can consider $\mathbb{F}_{q^m}(\mathbb{P}^1)$, the function field over $\mathbb{F}_{q^m}$ associated to the curve $\mathbb{P}^1$. We recall that a *closed point* of $\mathbb{P}^1$ is an orbit of a point, with coordinates in a finite extension of $\mathbb{F}_{q^m}$, under the Frobenius transformation: $(x : y) \mapsto (x^{q^m} : y^{q^m})$. The *degree* of a closed point is the cardinality of the orbit. A *rational point* is a closed point of degree 1, or equivalently a point whose coordinates are in $\mathbb{F}_{q^m}$. The set of rational points is denoted $\mathbb{P}^1(\mathbb{F}_{q^m})$. A divisor of $\mathbb{P}^1$ is a formal sum, with integers coefficients, of closed points of $\mathbb{P}^1$ and for $f \in \mathbb{F}_{q^m}(\mathbb{P}^1)$, the principal divisor of $f$, denoted by $(f)$, is defined as the formal sum of zeros and poles of $f$, counted with multiplicity. For a divisor $G$, we denote by $\deg(G)$ the degree of $G$ and by $L(G) := \{f \in \mathbb{F}_{q^m}(\mathbb{P}^1) \mid (f) \geq -G\} \cup \{0\}$, the Riemann-Roch space associated to $G$. Let $\mathcal{P} = \{P_1, \ldots, P_n\}$ be a set of $n$ distinct points of $\mathbb{P}^1$ with coordinates in $\mathbb{F}_{q^m}$ and $G$ be a divisor such that $\deg(G) < n$ and $G$ does not contain any point of $\mathcal{P}$. We consider the subspace $V \subset \mathbb{F}_{q^m}(\mathbb{P}^1)$ of functions without poles in $\mathcal{P}$ and the following map:

$$\begin{array}{rccc} \mathrm{Ev}_{\mathcal{P}} : & V & \longrightarrow & \mathbb{F}_{q^m}^n \\ & f & \longmapsto & (f(P_1), \ldots, f(P_n)). \end{array}$$

The AG code $C_L(\mathbb{P}^1, \mathcal{P}, G)$ is defined by $C_L(\mathbb{P}^1, \mathcal{P}, G) := \{\mathrm{Ev}_{\mathcal{P}}(f) \mid f \in L(G)\}$.

Let $x = (x_1, \ldots, x_n)$ be an $n$-tuple of distinct elements of $\mathbb{F}_{q^m}$, and $y = (y_1, \ldots, y_n)$ be an $n$-tuple of nonzero elements of $\mathbb{F}_{q^m}$. Then $\mathrm{GRS}_k(x, y)$ is the AG code $C_L(\mathbb{P}^1, \mathcal{P}, G)$ where $\mathcal{P} := \{(x_i : 1) \mid i \in \{1, \ldots, n\}\}$ and $G := (k-1)P_{\infty} - (f)$, with $f \in \mathbb{F}_{q^m}(\mathbb{P}^1)$ a function with pole order $n - 1$ at $P_{\infty}$, which is the interpolation polynomial of degree $n - 1$ of $y_1, \ldots, y_n$ through the points $x_1, \ldots, x_n$. With the same notation, we have $\mathcal{A}_k(x, y) := C_L(\mathbb{P}^1, \mathcal{P}, G)^{\perp} \cap \mathbb{F}_q^n$.

### 2.2 Induced permutations of Alternant Codes

We explain how we can construct an alternant code invariant under a prescribed permutation of the support $\{1, \ldots, n\}$, with $n$ the length of the code. In [6], Dür determines the automorphism

group of GRS codes and in [1, 2], Berger uses this to construct families of alternant codes invariant under a permutation. In particular, Berger deals with some alternant codes invariant under a permutation induced by the action of an element of the projective semi-linear group $\mathrm{P\Gamma L}_2(\mathbb{F}_{q^m})$ on the support and the multiplier. Here we will only be interested in projective linear transformations. First of all, we recall the definition of the projective linear group $\mathrm{PGL}_2(\mathbb{F}_{q^m})$. It is the automorphism group of the projective line $\mathbb{P}^1$ defined by:

$$
\mathrm{PGL}_2(\mathbb{F}_{q^m}) := \left\{ \begin{array}{ccc} \mathbb{P}^1 & \longrightarrow & \mathbb{P}^1 \\ (x:y) & \longmapsto & (ax+by : cx+dy) \end{array} \ \middle| \ \begin{cases} a,b,c,d \in \mathbb{F}_{q^m}, \\ ad-bc \neq 0 \end{cases} \right\}.
$$

The elements of $\mathrm{PGL}_2(\mathbb{F}_{q^m})$ have also a matrix representation, i.e.

$$
\forall \sigma \in \mathrm{PGL}_2(\mathbb{F}_{q^m}), \text{ we write } \sigma := \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \text{ with } ad-bc \neq 0. \tag{1}
$$

Where the elements $a, b, c$ and $d$ are defined up to a multiplication by a nonzero scalar. That is to say:

$$
\mathrm{PGL}_2(\mathbb{F}_{q^m}) \simeq \mathrm{GL}_2(\mathbb{F}_{q^m}) / \left\{ \begin{pmatrix} \alpha & 0 \\ 0 & \alpha \end{pmatrix}, \alpha \in \mathbb{F}_{q^m}^* \right\}
$$

Now, we deal with permutations of an alternant code. We recall the following definition.

**Definition 2.** Let $\mathcal{C}$ be a linear code of length $n$ over $\mathbb{F}_{q^m}$. Let $\sigma \in \mathfrak{S}_n$ be a permutation, acting on $\mathcal{C}$ via $\sigma(c_1, \ldots, c_n) = (c_{\sigma(1)}, \ldots, c_{\sigma(n)})$. Then the permutation group of a code $\mathcal{C} \subset \mathbb{F}_{q^m}^n$, is $Perm(\mathcal{C}) := \{\sigma \in \mathfrak{S}_n \mid \sigma(\mathcal{C}) = \mathcal{C}\}$.

In the case of GRS codes, for appropriate dimension, Dür [6] shows that the whole permutation group is induced by the action of the projective linear group on the support of the code. The same property has been shown by Stichtenoth [18], with the representation of GRS codes as AG rational codes. More precisely, for appropriate parameters, every permutation of $\mathcal{C}_L(\mathbb{P}^1, \mathcal{P}, G)$ is induced by a projective linear transformation. We give the main definitions and theorems of [18].

We keep the notation of the previous section. Let $G$ and $G'$ be divisors of $\mathbb{P}^1$, we note $G \approx_{\mathcal{P}} G'$ if there exists $f \in \mathbb{F}_{q^m}(\mathbb{P}^1)$, $f \neq 0$, such that $G - G' = (f)$ and $f(P) = 1$, for all $P \in \mathcal{P}$. With this definition we have the following lemma:

**Lemma 2.2** ([18]). *If $G \approx_{\mathcal{P}} G'$ then $\mathcal{C}_L(\mathbb{P}^1, \mathcal{P}, G) = \mathcal{C}_L(\mathbb{P}^1, \mathcal{P}, G')$.*

Before giving the theorem which allows us to construct any GRS code invariant under a permutation, we define:

**Definition 3.** $\mathrm{Aut}_{\mathcal{P},G}(\mathbb{P}^1) := \{\sigma \in \mathrm{Aut}(\mathbb{P}^1) \mid \sigma(\mathcal{P}) = \mathcal{P} \text{ and } \sigma(G) \approx_{\mathcal{P}} G\}$.

**Theorem 2.3** ([18]). *Let $\mathcal{C} = \mathcal{C}_L(\mathbb{P}^1, \mathcal{P}, G)$ be an AG code with $1 \leq deg(G) \leq n - 3$. Then $Perm(\mathcal{C}) = \mathrm{Aut}_{\mathcal{P},G}(\mathbb{P}^1)$.*

Now we have all the properties required to construct some alternant codes invariant under a permutation. We consider $\sigma \in \mathrm{PGL}_2(\mathbb{F}_{q^m})$ and $\ell = ord(\sigma)$. We define the support:

$$
\mathcal{P} := \coprod_{i=1}^{n/\ell} Orb_\sigma(P_i), \tag{2}
$$

where the points $P_i \in \mathbb{P}^1(\mathbb{F}_{q^m})$ are pairwise distinct with trivial stabilizer subgroup and $Orb_\sigma(P_i) := \{\sigma^j(P_i) \mid j \in \{1..\ell\}\}$. We define the divisor:

$$G := \sum_{i=1}^s t_i \sum_{Q \in Orb_\sigma(Q_i)} Q, \tag{3}$$

with $Q_i$ closed points of $\mathbb{P}^1$, $s \in \mathbb{N}$, $t_i \in \mathbb{Z}$ for $i \in \{1, \ldots, s\}$ and $\deg(G) = \sum_{i=1}^s t_i \ell$.

The automorphism $\sigma$ of $\mathbb{P}^1$ induces a permutation $\tilde{\sigma}$ of $\mathcal{C} = \mathcal{C}_L(\mathbb{P}^1, \mathcal{P}, G)$ defined by:

$$\tilde{\sigma} \colon \quad \begin{array}{ccc} \mathcal{C} & \longrightarrow & \mathcal{C} \\ (f(P_1), \ldots, f(P_n)) & \longmapsto & (f(\sigma(P_1)), \ldots, f(\sigma(P_n))) \end{array}\cdot$$

Then $\tilde{\sigma}$ is also a permutation of $\mathcal{A} := \mathcal{C}^\perp \cap \mathbb{F}_q^n$. For short, we denote by $\sigma \in \mathrm{PGL}_2(\mathbb{F}_{q^m})$ both the homography and the induced permutation on the code $\mathcal{C}$.

# 3 Subcodes of Alternant Codes

We can construct subcodes of $\mathcal{A}_r(x, y)$ with smaller parameters, by simple operations, which can be used to recover the alternant code $\mathcal{A}_r(x, y)$. We describe in the next section two subcodes: the *folded* code and the *invariant* code. Their interactions are also discussed. In the papers [8, 7], the folding operation was used to recover dual of the considered alternant code. Here we do not need to consider the dual code. More precisely, we show that for alternant codes with a non trivial permutation group, the invariant code is an alternant code.

## 3.1 Invariant and Folded Codes

This section deals with subcodes called the *invariant* code and the *folded* code whose definitions are the following.

**Definition 4.** Let $\mathcal{C}$ be a linear code and $\sigma \in Perm(\mathcal{C})$ of order $\ell$, we consider the following map:

$$\psi \colon \mathcal{C} \to \mathcal{C}$$
$$c \mapsto \sum_{i=0}^{\ell-1} \sigma^i(c).$$

The folded code of $\mathcal{C}$ is defined by $\mathrm{Fold}_\sigma(\mathcal{C}) := \mathrm{Im}(\psi) = \mathrm{Im}(Id + \sigma + \cdots + \sigma^{\ell-1})$. The invariant code of $\mathcal{C}$ is defined by $\mathcal{C}^\sigma := \ker(\sigma - \mathrm{Id})$.

The folded code was used in [8, 7], in order to construct a structured subcode invariant by a given permutation $\sigma$. Indeed, by the previous definition, we remark that $\mathrm{Fold}_\sigma(\mathcal{C})$ is $\sigma$-invariant.

**Proposition 3.1.** *The codes* $\mathrm{Fold}_\sigma(\mathcal{C})$ *and* $\mathcal{C}^\sigma$ *are subcodes of* $\mathcal{C}$ *and we have:*

$$\mathrm{Fold}_\sigma(\mathcal{C}) \subseteq \mathcal{C}^\sigma.$$

These two codes are not equal in the general case but we have the following lemma. We recall that $p$ is the characteristic of $\mathbb{F}_q$ and $\ell = ord(\sigma)$.

**Lemma 3.2.** *If $p \nmid \ell$ then $\mathrm{Fold}_\sigma(\mathcal{C}) = \mathcal{C}^\sigma$.*

*Proof.* Let $\psi$ be the map of Definition 4, by the previous proposition we know that:

$$\mathrm{Im}(\psi) \subseteq \ker(\sigma - \mathrm{Id}).$$

Now, we will show that $\dim(\mathrm{Im}(\psi)) = \dim(\ker(\sigma - \mathrm{Id}))$.
By the rank–nullity theorem we know that:

$$\dim(\mathrm{Im}(\psi)) = \dim(\mathcal{C}) - \dim(\ker(\psi)).$$

Moreover, $\sigma^\ell - \mathrm{Id} = (\mathrm{Id} + \sigma + \cdots + \sigma^{\ell-1})(\sigma - \mathrm{Id})$. Since $p \nmid \ell$, we have:

$$\gcd(\sum_{i=0}^{\ell-1} X^i, X - 1) = 1.$$

Hence, $\mathcal{C} = \ker(\psi) \oplus \ker(\sigma - \mathrm{Id})$ and

$$\dim(\ker(\sigma - \mathrm{Id})) = \dim(\mathcal{C}) - \dim(\ker(\psi)).$$

Therefore $\ker(\sigma - \mathrm{Id}) = \mathrm{Im}(\psi)$ and $Fold(\mathcal{C}) = \mathcal{C}^\sigma$. $\qquad\square$

*Remark* 1. In [7, Example 1] an example of the folded and the invariant codes of a $\sigma$-invariant alternant code $\mathcal{A}$ is given. In this example, the authors wrote that $\mathrm{Fold}_\sigma(\mathcal{A}) \subsetneq \mathcal{A}^\sigma$ but in this case these two codes must be equal. Indeed, for this example, $p = 3 \nmid 2 = ord(\sigma)$ and by the previous lemma we have $\mathrm{Fold}_\sigma(\mathcal{A}) = \mathcal{A}^\sigma$. Actually, the computation of the subfield subcode of the folded code in [7, Example 1] contains a mistake.

*Remark* 2. If $c \in \mathrm{Fold}_\sigma(\mathcal{C})$ or $c \in \mathcal{C}^\sigma$, then $c$ takes constant value on the orbits under the action of $\sigma$: $\{i, \sigma(i), \ldots, \sigma^{\ell-1}(i)\}$. In order to work with codes without repeated coordinates, we choose $I \subset \{1, \ldots, n\}$ a set of representatives of orbits $\{\sigma^j(i) | j \in \{0, \ldots, \ell-1\}\}$ and we consider the codes punctured on this set: $\mathrm{Fold}_\sigma(\mathcal{C})|_I$ and $\mathcal{C}^\sigma|_I$. For short, we keep the notations $\mathrm{Fold}_\sigma(\mathcal{C})$ and $\mathcal{C}^\sigma$, for the restricted codes.

*Remark* 3. The folding operation is $\mathbb{F}_q$-linear so to apply this operation on a linear code $\mathcal{C}$ it suffices to apply folding operation on a basis of $\mathcal{C}$. This property will be useful in §3.2.3.

## 3.2 The Invariant Code of $\mathcal{A}_r(x, y)$

In order to study the invariant code of $\mathcal{A}_r(x, y)$, we first notice that the invariant operation commutes with the subfield subcode operation. Indeed, if $\mathcal{C}$ is a linear code over $\mathbb{F}_{q^m}$, $\sigma$-invariant then:

$$(\mathcal{C} \cap \mathbb{F}_q^n)^\sigma = \{c \in \mathcal{C} \mid c \in \mathbb{F}_q^n \text{ and } \sigma(c) = c\} = \mathcal{C}^\sigma \cap \mathbb{F}_q^n.$$

In order to prove that the invariant code of $\mathcal{A}_r(x, y)$ is also an alternant code we have to prove that the invariant code of a GRS code is a GRS code. Later on, the GRS codes will be described by $\mathcal{C}_L(\mathbb{P}^1, \mathcal{P}, G)$, as in Section 2.2. The two lemmata to follow describe the action of an element $\sigma \in \mathrm{PGL}_2$ on the codes $\mathcal{C}_L(\mathbb{P}^1, \mathcal{P}, G)$ and provide a description of $\mathcal{C}_L(\mathbb{P}^1, \mathcal{P}, G)^\sigma$.

**Lemma 3.3.** *Let $c = Ev_\mathcal{P}(f) \in \mathcal{C}_L(\mathbb{P}^1, \mathcal{P}, G)^\sigma$ such that $\sigma(c) = c$, then $f$ is $\sigma$-invariant, i.e. $f \circ \sigma = f$.*

*Proof.* Let $c = (f(P_1), \ldots, f(P_n)) \in \mathcal{C}$ such that $\sigma(c) = c$, then:

$$\forall i \in \{1, \ldots, n\}, \ f(P_{\sigma(i)}) = f(P_i) \Leftrightarrow \forall i \in \{1, \ldots, n\}, \ f \circ \sigma(P_i) = f(P_i)$$
$$\Leftrightarrow \forall i \in \{1, \ldots, n\}, \ (f \circ \sigma - f)(P_i) = 0.$$

Since $\sigma(G) = G$, $f \circ \sigma \in L(G)$, and then $(f \circ \sigma - f) \in L(G)$. Hence if $(f \circ \sigma - f)$ was nonzero, it should have at most $\deg(G) < n$ zeros on $\mathbb{P}^1$, which is a contradiction. Therefore $(f \circ \sigma - f) \equiv 0$ and $f$ is $\sigma$-invariant. $\qquad\square$

**Lemma 3.4.** *Let $\mathcal{C} := \mathcal{C}_L(\mathbb{P}^1, \mathcal{P}, G)$ be an AG code such that $\sigma(\mathcal{C}) = \mathcal{C}$ and $\rho \in \mathrm{PGL}_2(\mathbb{F}_{q^m})$. Then $\sigma' := \rho \circ \sigma \circ \rho^{-1}$ induces the same permutation on $\mathcal{C}$ as $\sigma$.*

*Proof.* We first prove that:

$$\mathcal{C}_L(\mathbb{P}^1, \rho^{-1}(\mathcal{P}), \rho^{-1}(G)) = \mathcal{C}_L(\mathbb{P}^1, \mathcal{P}, G).$$

Let $c = (f(P_1), \ldots, f(P_n))$ be a codeword of $\mathcal{C}_L(\mathbb{P}^1, \mathcal{P}, G)$. Then, we have $c = (f \circ \rho \circ \rho^{-1}(P_1), \ldots, f \circ \rho \circ \rho^{-1}(P_n))$. As $f \in L(G)$, the function $h = f \circ \rho \in L(\rho^{-1}(G))$. Hence, $c \in \{\mathrm{Ev}_{\rho^{-1}(\mathcal{P})}(h) \mid h \in L(\rho^{-1}(G))\} = \mathcal{C}_L(\mathbb{P}^1, \rho^{-1}(\mathcal{P}), \rho^{-1}(G))$.
Now, for all $c = (f(P_1), \ldots, f(P_n)) \in \mathcal{C}$, we have:

$$\sigma'(c) = (f \circ \rho \circ \sigma \circ \rho^{-1}(P_1), \ldots, f \circ \rho \circ \sigma \circ \rho^{-1}(P_n))$$
$$= (h \circ \sigma(\rho^{-1}(P_1)), \ldots, h \circ \sigma(\rho^{-1}(P_n)))$$

with $h = f \circ \rho \in L(\rho^{-1}(G))$. Since $\mathcal{C}_L(\mathbb{P}^1, \rho^{-1}(\mathcal{P}), \rho^{-1}(G)) = \mathcal{C}_L(\mathbb{P}^1, \mathcal{P}, G)$, $\sigma'$ induces the same permutation of the code $\mathcal{C}$ as $\sigma$. $\qquad\square$

**Theorem 3.5.** *Let $\mathcal{C}_L(\mathbb{P}^1, \mathcal{P}, G) \subseteq \mathbb{F}_{q^m}^n$ be an AG code of length $n$ and dimension $k$, and $\sigma \in \mathrm{PGL}_2(\mathbb{F}_{q^m})$ of order $\ell$ acting on it, with $\ell | n$. Let $\mathcal{P}$ and $G$ as in (2) and (3). Then the invariant code $\mathcal{C}_L(\mathbb{P}^1, \mathcal{P}, G)^\sigma$ is an AG code of length $n/\ell$ and dimension $\lfloor k/\ell \rfloor$.*

**Corollary 3.6.** *Let $\mathcal{A}(\mathbb{P}^1, \mathcal{P}, G) := \mathcal{C}_L(\mathbb{P}^1, \mathcal{P}, G) \cap \mathbb{F}_q^n$ be an alternant AG code of length $n$ and order $r$, and $\sigma \in \mathrm{PGL}_2(\mathbb{F}_{q^m})$ of order $\ell$ acting on it, with $\ell | n$. Let $\mathcal{P}$ and $G$ as in (2) and (3). Then the invariant code $\mathcal{A}(\mathbb{P}^1, \mathcal{P}, G)^\sigma$ is an alternant AG code of length $n/\ell$ and order $\lfloor r/\ell \rfloor$.*

In order to prove Theorem 3.5, we consider $\sigma \in \mathrm{PGL}_2(\mathbb{F}_{q^m})$ with $\ell = ord(\sigma)$ and we define the support $\mathcal{P}$ and the divisor $G$ as in (2) and (3). Later on, to simplify the proofs we assume that $G$ is constructed from single rational point $Q$, but the result remains true in the general case.

We denote:

$$\begin{array}{ll} \sigma^j(P_i) & := (\alpha_{i\ell+j} : \beta_{i\ell+j}), \text{ for } i \in \{0, \ldots, \frac{n}{\ell} - 1\}, j \in \{0, \ldots, \ell - 1\} \\ \sigma^j(Q) & := (\gamma_j : \delta_j), \text{ for } j \in \{0, \ldots, \ell - 1\}. \end{array} \qquad (4)$$

**Lemma 3.7.** *With the previous notation, we have:*

$$L(G) = \left\{ \frac{F(X, Y)}{\displaystyle\prod_{j=0}^{\ell-1} (\delta_j X - \gamma_j Y)^t} \ \middle| \ F \in \mathbb{F}_{q^m}[X, Y] \ \text{homogeneous polynomial of degree } t\ell. \right\}$$

To the automorphism $\sigma \in \mathrm{PGL}_2(\mathbb{F}_{q^m})$, we associate a matrix $M := \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ as in (1). Three cases are possible, depending on the eigenvalues of the matrix $M$:

7

1. $M \sim \begin{pmatrix} a & 0 \\ 0 & 1 \end{pmatrix}$, with $a \in \mathbb{F}_{q^m}$ (case diagonalizable in $\mathbb{F}_{q^m}$),

2. $M \sim \begin{pmatrix} 1 & b \\ 0 & 1 \end{pmatrix}$, with $b \in \mathbb{F}_{q^m}$ (case trigonalizable in $\mathbb{F}_{q^m}$),

3. $M \sim \begin{pmatrix} \alpha & 0 \\ 0 & \alpha^q \end{pmatrix}$, with $\alpha \in \mathbb{F}_{q^{2m}}$ (case diagonalizable in $\mathbb{F}_{q^{2m}}$),

where $M \sim N$, with $M, N \in \mathrm{PGL}_2(\mathbb{F}_{q^m})$, means there exist $P \in \mathrm{PGL}_2(\mathbb{F}_{q^m})$ such that $M = PNP^{-1}$. In the following, we study these three cases.

### 3.2.1 Case $\sigma$ diagonalizable over $\mathbb{F}_{q^m}$

We suppose $\sigma = \rho \circ \sigma_d \circ \rho^{-1}$ with $\sigma_d$ diagonal and $\rho \in \mathrm{PGL}_2(\mathbb{F}_{q^m})$ an automorphism of $\mathbb{P}^1_{\mathbb{F}_{q^m}}$. W.l.o.g and by Lemma 3.4, one can assume that:

$$\sigma: \quad \begin{array}{ccc} \mathbb{P}^1 & \to & \mathbb{P}^1 \\ (x:y) & \mapsto & (ax:y), \end{array} \tag{5}$$

with $a \in \mathbb{F}_{q^m}^*$.

**Proposition 3.8.** *Let $F \in \mathbb{F}_q[X, Y]$ be a homogeneous polynomial of degree $t\ell$, and $a \in \mathbb{F}_{q^m}$ of order $\ell$. If $F(aX, Y) = F(X, Y)$, then $F(X, Y) = R(X^\ell, Y^\ell)$, with $R \in \mathbb{F}_{q^m}[X, Y]$ a homogeneous polynomial of degree $t$.*

A proof is given in [7, Prop 4]. Here, we present a simpler proof of Proposition 3.8.

*Proof.* The homogeneous polynomial $F$ can be written as:

$$F(X, Y) = \sum_{i+j=t\ell} f_{ij} X^i Y^j,$$

with $f_{ij} \in \mathbb{F}_{q^m}$. Since $F(aX, Y) = F(X, Y)$, we have:

$$\sum_{i+j=t\ell} f_{ij} X^i Y^j = \sum_{i+j=t\ell} f_{ij} a^i X^i Y^j.$$

Hence $f_{ij} = a^i f_{ij}, \forall i, j \in \mathbb{N}$ such that $i + j = t\ell$. As the order of $a$ is $\ell$, we have $a^i \neq 1, \forall i \in \mathbb{N}$ such that $\ell \nmid i$. Therefore $f_{ij} = 0, \forall i \in \mathbb{N}$ such that $\ell \nmid i$. So $F(X, Y) = R(X^\ell, Y^\ell)$, with $R \in \mathbb{F}_{q^m}[X, Y]$ an homogeneous polynomial of degree $t$. $\square$

**Proposition 3.9.** *Let $\mathcal{C} := \mathcal{C}_L(\mathbb{P}^1, \mathcal{P}, G)$ be an AG code as in Theorem 3.5, with $\sigma$ as in (5). Let $\tilde{P}_i = (\alpha_i^\ell : \beta_i^\ell)$ and $\tilde{G} = t\tilde{Q}$, where either $\tilde{Q} = ((-1)^{\ell-1} a^{\frac{\ell(\ell-1)}{2}} (\frac{\gamma_0}{\delta_0})^\ell : 1)$ or $\tilde{Q} = P_\infty$. Then $\mathcal{C}^\sigma = \mathcal{C}_L(\mathbb{P}^1, \tilde{\mathcal{P}}, \tilde{G})$, which is a GRS code.*

*Proof.* Let $c = \big(f(P_1), f(\sigma(P_1)), \dots, f(\sigma^{\ell-1}(P_{\frac{n}{\ell}}))\big) \in \mathcal{C}$ such that $\sigma(c) = c$, by Lemma 3.3, $f \in L(G)$ is $\sigma$-invariant, so $f(aX, Y) = f(X, Y)$. By Lemma 3.7, we have:

$$\frac{F(aX, Y)}{\left(\prod_{j=0}^{\ell-1} (a\delta_j X - \gamma_j Y)\right)^t} = \frac{F(X, Y)}{\left(\prod_{j=0}^{\ell-1} (\delta_j X - \gamma_j Y)\right)^t} \tag{6}$$

8

with $F \in \mathbb{F}_{q^m}[X, Y]$ a homogeneous polynomial of degree $t\ell$. Moreover the support of $G$ is $\sigma$-invariant, so:

$$\prod_{j=0}^{\ell-1} (a\delta_j X - \gamma_j Y) = \prod_{j=0}^{\ell-1} (a\delta_j X - a\gamma_j Y) = a^\ell \prod_{j=0}^{\ell-1} (\delta_j X - \gamma_j Y) = \prod_{j=0}^{\ell-1} (\delta_j X - \gamma_j Y).$$

Hence, (6) becomes $F(aX, Y) = F(X, Y)$, because $\ell = \mathrm{ord}(a)$. By Proposition 3.8:

$$F(X, Y) = R(X^\ell, Y^\ell),$$

with $R \in \mathbb{F}_{q^m}[X, Y]$ an homogeneous polynomial of degree $t$.

The product $\prod_{j=0}^{\ell-1} (\delta_j X - \gamma_j Y)$ is also $\sigma$-invariant and, by Proposition 3.8, we have:

$$\prod_{j=0}^{\ell-1} (\delta_j X - \gamma_j Y) = \left( \prod_{j=0}^{\ell-1} \delta_j \right) X^\ell + (-1)^\ell \left( \prod_{j=0}^{\ell-1} \gamma_j \right) Y^\ell.$$

Therefore:

$$f(X, Y) = \frac{R(X^\ell, Y^\ell)}{\left( \left( \prod_{j=0}^{\ell-1} \delta_j \right) X^\ell - (-1)^{\ell-1} \left( \prod_{j=0}^{\ell-1} \gamma_j \right) Y^\ell \right)^t}. \tag{7}$$

Forall $i \in \{1, \ldots, \frac{n}{\ell}\}$, we have $f(P_i) = \tilde{f}(\tilde{P}_i)$, with:

$$\tilde{f}(X, Y) = \frac{R(X, Y)}{\left( \left( \prod_{j=0}^{\ell-1} \delta_j \right) X - (-1)^{\ell-1} \left( \prod_{j=0}^{\ell-1} \gamma_j \right) Y \right)^t}.$$

and $\tilde{P}_i := (\alpha_i^\ell : \beta_i^\ell)$. We denote $\tilde{\delta} = \left( \prod_{j=0}^{\ell-1} \delta_j \right)$ and $\tilde{\gamma} = (-1)^{\ell-1} \left( \prod_{j=0}^{\ell-1} \gamma_j \right)$. By Lemma 3.7, we have:

$$\left\{ \frac{R(X, Y)}{\left( \tilde{\gamma} X - \tilde{\delta} Y \right)^t} \mid R \in \mathbb{F}_{q^m}[X, Y] \text{ homogeneous polynomial of degree } t. \right\} \cup \{0\} = L(\tilde{G}),$$

with $\tilde{G} = t(\tilde{\gamma} : \tilde{\delta}) = t\tilde{Q}$. Hence the codeword $c \in \mathcal{C}_L(\mathbb{P}^1, \tilde{\mathcal{P}}, \tilde{G})$.
If $\tilde{Q} \neq P_\infty$, then $\forall j, \delta_j \neq 0$ and we can write:

$$\tilde{Q} = \left( (-1)^{\ell-1} \prod_{j=0}^{\ell-1} \frac{\gamma_j}{\delta_j} : 1 \right).$$

Moreover we have:

$$\prod_{j=0}^{\ell-1} \frac{\gamma_j}{\delta_j} = \prod_{j=0}^{\ell-1} a^j \frac{\gamma_0}{\delta_0} = \left( \prod_{j=0}^{\ell-1} a^j \right) \left( \frac{\gamma_0}{\delta_0} \right)^\ell = a^{\frac{\ell(\ell-1)}{2}} \left( \frac{\gamma_0}{\delta_0} \right)^\ell.$$

Reciprocally, for $c \in \mathcal{C}_L(\mathbb{P}^1, \tilde{\mathcal{P}}, \tilde{G})$ it is easy to see that we have $c = (f(P_1), \ldots, f(\sigma^{\ell-1}(P_{\frac{n}{\ell}})))$, with $f$ as in (7). Then $c \in \mathcal{C}^\sigma$. $\qquad \square$

### 3.2.2 Case $\sigma$ trigonalizable over $\mathbb{F}_{q^m}$

Here we consider the case where $\sigma$ is trigonalizable in $\mathbb{F}_{q^m}$. As in the previous section we only have to treat the case where $\sigma$ is upper triangular. So w.l.o.g one can assume that:

$$\sigma: \quad \begin{array}{ccc} \mathbb{P}^1 & \to & \mathbb{P}^1 \\ (x:y) & \mapsto & (x+by:y) \end{array} \tag{8}$$

with $b \in \mathbb{F}_{q^m}^*$. In this case, we have $\ell = \mathrm{ord}(\sigma) = p$.

**Proposition 3.10** ([7, Prop 4]). *Let $F \in \mathbb{F}_q[z]$ be a polynomial of degree $\deg(F) \le tp$ and $b \in \mathbb{F}_q^*$. If $F(z+b) = F(z)$, then $F(z) = R(z^p - b^{p-1}z)$, with $R \in \mathbb{F}_q[z]$ a polynomial of degree $\deg(R) \le t$.*

**Proposition 3.11.** *Let $\mathcal{C} := \mathcal{C}_L(\mathbb{P}^1, \mathcal{P}, G)$ be a $\sigma$-invariant AG code as in Theorem 3.5, with $\sigma$ as in (8). Let $\tilde{P}_i = (\alpha_i^p - b^{p-1}\alpha_i\beta_i^{p-1} : \beta_i^p)$ and $\tilde{G} = t(\tilde{Q})$, where either $\tilde{Q} = ((\frac{\gamma_0}{\delta_0})^p - b^{p-1}\frac{\gamma_0}{\delta_0} : 1)$ or $\tilde{Q} = P_\infty$. Then $\mathcal{C}^\sigma = \mathcal{C}_L(\mathbb{P}^1, \tilde{\mathcal{P}}, \tilde{G})$, which is a GRS code.*

*Proof.* Let $c = \big(f(P_1), f(\sigma(P_1)), \dots, f(\sigma^{\ell-1}(P_{\frac{n}{\ell}}))\big) \in \mathcal{C}$ such that $\sigma(c) = c$. By Lemma 3.3, $f$ is $\sigma$-invariant so: $f(X+bY,Y) = f(X,Y)$. By Lemma 3.7, we have:

$$\frac{F(X+bY,Y)}{\Big(\prod\limits_{j=0}^{p-1}\big(\delta_j(X+bY)-\gamma_jY\big)\Big)^t} = \frac{F(X,Y)}{\Big(\prod\limits_{j=0}^{p-1}\big(\delta_jX-\gamma_jY\big)\Big)^t}, \tag{9}$$

with $F \in \mathbb{F}_q[X,Y]$ an homogeneous polynomial of degree $tp$. Moreover the support of $G$ is $\sigma$-invariant, so:

$$\prod_{j=0}^{p-1}(\delta_j(X+bY)-\gamma_jY) = \prod_{j=0}^{p-1}(\delta_jX-(\gamma_j-b\delta_j)Y) = \prod_{j=0}^{p-1}(\delta_jX-\gamma_jY).$$

Hence, (9) becomes $F(X+bY,Y)) = F(X,Y)$. If we write $z = \frac{X}{Y}$, then we have $F(z+b,1) = F(z,1)$. By Proposition 3.10, we have $F(z) = R(z^p - b^{p-1}z)$, with $R \in \mathbb{F}_q[z]$ a polynomial of degree $\deg(R) \le t$.

The product $\prod\limits_{j=0}^{p-1}(\delta_jz-\gamma_j)$ is also $\sigma$-invariant and, by Proposition 3.10, we have:

$$\prod_{j=0}^{p-1}(\delta_jz-\gamma_j) = \Big(\prod_{j=0}^{p-1}\delta_j\Big)\big(z^p - b^{p-1}z\big) + (-1)^p\prod_{j=0}^{p-1}\gamma_j.$$

Hence:

$$f(X,Y) = \frac{R(X^p - b^{p-1}XY^{p-1},Y^p)}{\Big(\big(\prod\limits_{j=0}^{p-1}\delta_j\big)\big(X^p - b^{p-1}XY^{p-1}\big) - \big((-1)^{p-1}\prod\limits_{j=0}^{p-1}\gamma_j\big)Y^p\Big)^t}.$$

The arguments to conclude this proof are the same that in Proposition 3.9.

Moreover, if $\tilde{Q} \ne P_\infty$, $\forall j, \delta_j \ne 0$ and we can write:

$$\tilde{Q} = \Big(\prod_{j=0}^{p-1}\frac{\gamma_j}{\delta_j} : 1\Big).$$

We have:

$$\prod_{j=0}^{p-1}\frac{\gamma_j}{\delta_j} = \prod_{j=0}^{p-1}\big(\frac{\gamma_0}{\delta_0}+jb\big) = \big(\frac{\gamma_0}{\delta_0}\big)^p - b^{p-1}\frac{\gamma_0}{\delta_0}.$$

$\square$

### 3.2.3 Case $\sigma$ diagonalizable in $\mathbb{F}_{q^{2m}} \backslash \mathbb{F}_{q^m}$

We suppose that $\sigma = \rho \circ \sigma_d \circ \rho^{-1}$ with $\sigma_d$ diagonal in $\mathrm{GL}_2(\mathbb{F}_q^{2m})$ and $\rho \in \mathrm{PGL}_2(F_{q^{2m}})$. We want to extend the code $\mathcal{C}$ defined on $\mathbb{F}_{q^m}$ to the field $\mathbb{F}_{q^{2m}}$. So we consider the set $\mathrm{Span}_{\mathbb{F}_{q^{2m}}} < \mathcal{C} >$, i.e. $\mathcal{C} \otimes \mathbb{F}_{q^{2m}}$.

The order $\ell$ of $\sigma_d := \begin{pmatrix} \alpha & 0 \\ 0 & \alpha^q \end{pmatrix}$ is the order of $\alpha \in \mathbb{F}_{q^{2m}}$, so $\ell \mid (q^{2m} - 1)$. Since $q := p^s$, where $s \in \mathbb{N}^*$, we have $\ell \mid (p^{s2m} - 1)$ and so $p \nmid \ell$. By Lemma 3.2, $\mathrm{Fold}_\sigma = \mathrm{Inv}_\sigma$, and so we have the following diagram:

$$\mathcal{C} \otimes \mathbb{F}_{q^{2m}} = \{\mathrm{Ev}_\mathcal{P}(f) | f \in L_{\mathbb{F}_{q^{2m}}}(G)\} \xrightarrow{\mathrm{Fold}_{\sigma_d}} \mathrm{Inv}_{\sigma_d}(\mathcal{C} \otimes \mathbb{F}_{q^{2m}})$$

$$\big\uparrow \text{Subfield Subcode} \qquad\qquad\qquad \big\uparrow \text{Subfield Subcode}$$

$$\mathcal{C} = \{\mathrm{Ev}_\mathcal{P}(f) | f \in L_{\mathbb{F}_{q^m}}(G)\} \xrightarrow{\mathrm{Fold}_\sigma} \mathrm{Inv}_\sigma(\mathcal{C})$$

By Section 3.2.1, the code $\mathrm{Inv}_\sigma(\mathcal{C} \otimes \mathbb{F}_{q^{2m}})$ is a GRS code. Since the application $\mathrm{Fold}_\sigma$ is $\mathbb{F}_q$-linear and $\mathcal{C} \otimes \mathbb{F}_{q^{2m}}$ has, by definition, a basis in $\mathbb{F}_{q^m}^n$, the code $\mathrm{Inv}_\sigma(\mathcal{C} \otimes \mathbb{F}_{q^{2m}})$ also has a basis in $\mathbb{F}_{q^m}^n$. Therefore, the subfield subcode on $\mathbb{F}_{q^m}$ of the GRS code $\mathrm{Inv}_\sigma(\mathcal{C} \otimes \mathbb{F}_{q^{2m}})$ is a GRS code.

## 4 Security of $\sigma$-invariant Alternant Codes

In this section, we study the security of keys of the McEliece scheme based alternant codes, with $\sigma$ an automorphism of the projective line acting on it. We consider an automorphism $\sigma \in \mathrm{PGL}_2(\mathbb{F}_{q^m})$, and a $\sigma$-invariant alternant code $\mathcal{A}(\mathbb{P}^1, \mathcal{P}, G) = \mathcal{C}_L(\mathbb{P}^1, \mathcal{P}, G)^\perp \cap \mathbb{F}_q^n$, with $\mathcal{P}$ a $\sigma$-invariant set of distinct points of $\mathbb{P}^1$ defined as (2) and $G$ a $\sigma$-invariant divisor of $\mathbb{P}^1$ defined as (3). By Section 3.2 and Corollary 3.6, the invariant code $\mathcal{A}(\mathbb{P}^1, \mathcal{P}, G)^\sigma$ is an alternant code $\mathcal{A}(\mathbb{P}^1, \tilde{\mathcal{P}}, \tilde{G})$, with smaller parameters.

Here we assume that it is possible to recover $\tilde{\mathcal{P}}$ and $\tilde{G}$ from a generator matrix of the code $\mathcal{A}(\mathbb{P}^1, \tilde{\mathcal{P}}, \tilde{G})$. This can be done by a brute force attack, if the parameters of $\mathcal{A}(\mathbb{P}^1, \tilde{\mathcal{P}}, \tilde{G})$ are smaller enough. Otherwise, we assume that an algebraic attack, proposed in [9], can recover $\tilde{\mathcal{P}}$ and $\tilde{G}$.

We will show that thanks to the knowledge of $\tilde{\mathcal{P}}$ and $\tilde{G}$ we are able to recover $\mathcal{P}$ and $G$. We already know that there is a link between the form of $\tilde{\mathcal{P}}$ and $\tilde{G}$ of the invariant code and the form of $\mathcal{P}$ and $G$ of the original alternant code. The link is described by Propositions 3.9 and 3.11 of the previous section.

Later on, we denote $\tilde{\mathcal{P}} := \{(\tilde{\alpha}_i : 1) \mid i \in \{1, \ldots, \frac{n}{\ell}\}\}$ the support of the invariant code. All the $\tilde{\alpha}_i$ are known. We denote by $(\alpha_{i,j} : 1)$ for $i \in \{1, \ldots, \frac{n}{\ell}\}$ and $j \in \{0, \ldots, \ell - 1\}$ the elements of $\mathcal{P}$. We assume that $G$ is constructed from one rational point $Q$, ie: $G = t \sum_{j=0}^{\ell-1} \sigma^j(Q)$, with $\sigma^j(Q) := (\gamma_j : \delta_j)$, for all $j \in \{0, \ldots, \ell - 1\}$. The result remains true for the general case.

### 4.1 Recover the divisor and guess the support

As previously, we know three cases are possible: $\sigma$ is diagonalizable over $\mathbb{F}_{q^m}$, $\sigma$ is trigonalizable, or $\sigma$ is diagonalizable over $\mathbb{F}_{q^{2m}}$. In this section we treat the two first cases, the third case will be treated at the end of Section 4. The order $\ell$ of $\sigma$ is known, hence we know the form of $\sigma$.

### 4.1.1 Case $\sigma$ diagonalizable over $\mathbb{F}_{q^m}$

In this case, we recall that the form is:

$$\sigma : \quad \begin{matrix} \mathbb{P}^1 & \to & \mathbb{P}^1 \\ (x:y) & \mapsto & (ax:y), \end{matrix} \tag{10}$$

with $a \in \mathbb{F}_{q^m}^\times$, an $\ell$-th root of unity. There exist only $\varphi(\ell) < n$ possibilities for $a$, where $\varphi$ the Euler's phi function, hence we are able to test all the possibilities. W.l.o.g we assume for now that we know the element $a$. The first step is to recover $G$ from $\tilde{G}$. By Proposition 3.9, we know that $\tilde{G} = t\tilde{Q}$, where either $\tilde{Q} = ((-1)^{\ell-1} a^{\frac{\ell(\ell-1)}{2}} (\frac{\gamma_0}{\delta_0})^\ell : 1)$ or $\tilde{Q} = P_\infty$. Since we know $a$, we can recover the support of $G$ thanks to the support $\tilde{Q}$ of $\tilde{G}$.

*Remark* 4. For all $i \in \{0, \dots, \ell-1\}$, we have $\tilde{Q} = ((-1)^{\ell-1} \sigma^i(a)^{\frac{\ell(\ell-1)}{2}} (\frac{\gamma_i}{\delta_i})^\ell : 1)$. We denote $A := \{\sigma^i(a) \mid i \in \{0, \dots, \ell-1\}\}$ and then from every $a \in A$ we are able to recover the support of $G$. The set $A$ is exactly the set of roots of the unity.

---

**Algorithm 1:** Recover the divisor in the case $\sigma$ diagonalizable in $\mathbb{F}_{q^m}$

**Input** : The divisor $\tilde{G}$ of the invariant code $\mathcal{A}(\mathcal{P}, G)^\sigma$.
**Output:** Return the support $G$

1  $a \leftarrow$ a primitive $\ell$-th root of $\mathbb{F}_{q^m}$
2  **if** $\tilde{Q} \neq P_\infty$ **then**              `// ` $\tilde{G} = t * \tilde{Q}$`, with ` $\tilde{Q} = (\tilde{\gamma} : 1)$
3  $\quad \Gamma \leftarrow \mathrm{roots}(a^{\frac{\ell(\ell-1)}{2}} X^\ell - \tilde{\gamma})$      `// ` $a \in A$
4  $\quad G' \leftarrow t \sum\limits_{\gamma \in \Gamma} (\gamma : 1)$
5  **else**
6  $\quad G' = tP_\infty$
7  **return** $G$.

---

The second step is to recover a support $\mathcal{P}'$ such as $\mathcal{A}(\mathcal{P}', G) = \mathcal{A}(\mathcal{P}, G)$. By Proposition 3.9, we know that a point $P = (x : y)$ in $\mathcal{P}$ satisfies:

$$\begin{cases} x^\ell - \tilde{\alpha}_i = 0 \\ y^\ell - \tilde{\beta}_i = 0, \end{cases} \tag{11}$$

for some $i \in \{1, \dots, \frac{n}{\ell}\}$ such that $(\tilde{\alpha}_i : \tilde{\beta}_i) = \tilde{P}_i$. Since we know $\tilde{\mathcal{P}}$, we are able to recover all elements of $\mathcal{P}$ but as an unordered set. We choose one solution $(\alpha_i, \beta_i)$ of (11) for each $i \in \{1, \dots, \frac{n}{\ell}\}$ and we choose $a \in A$, then the set:

$$\mathcal{P}' := \left\{ \left( a^j \frac{\alpha_i}{\beta_i} : 1 \right) \mid j \in \{0, \dots, \ell-1\}, i \in \{1, \dots, \frac{n}{\ell}\} \right\} \tag{12}$$

is a support such as $\mathcal{A}(\mathcal{P}', G)$ is a permutation of the code $\mathcal{A}(\mathcal{P}, G)$. For each choice of set of solutions $S := \{(\alpha_i, \beta_i) \mid i \in \{1, \dots, \frac{n}{\ell}\}\}$ and each choice of $a \in A$, we have a different support $\mathcal{P}'$. In Section 4.2 we give an algorithm to find a good choice for $S$ and $a$ and hence the permutation between $\mathcal{A}(\mathcal{P}', G)$ and $\mathcal{A}(\mathcal{P}, G)$.

### 4.1.2 Case $\sigma$ trigonalisable over $\mathbb{F}_{q^m}$

In the case where $\sigma$ is trigonalisable, it is more complicated to know exactly $\sigma$. In this case we know that $\sigma$ has the following form:

$$
\sigma: \quad \begin{array}{ccc} \mathbb{P}^1 & \to & \mathbb{P}^1 \\ (x:y) & \mapsto & (x+by:y), \end{array} \tag{13}
$$

with $b \in \mathbb{F}_{q^m}^{\times}$. Here the order of $\sigma$ is $\ell = p := \mathrm{Char}(\mathbb{F}_{q^m})$ and the first step is to recover $b$.

**Lemma 4.1.** $b$ is a root of the polynomial:

$$
P_b := \gcd\Big( \big\{ \mathrm{Res}_X(X^p - Y^{p-1}X - \tilde{\alpha}_i, \ X^{q^m} - X) \mid i \in \{1, \ldots, \tfrac{n}{\ell}\}\big\}, \ Y^{q^m} - Y\Big),
$$

where $\mathrm{Res}_X(P, Q)$ denotes the resultant of the two polynomials $P$ and $Q$ with respect to $X$.

*Proof.* By Proposition 3.11, $b$ is a root of the polynomial $\alpha_{i,j}^p - Y^{p-1}\alpha_{i,j} - \tilde{\alpha}_i \in \mathbb{F}_{q^m}[Y]$ for all $i \in \{1, \ldots, \tfrac{n}{p}\}$ and $j \in \{0, \ldots, p-1\}$. As $\alpha_{i,j} \in \mathbb{F}_{q^m}^{\times}$ for all $i, j$, we can also write that $b$ is a root of the polynomial $\mathrm{Res}_X(X^p - Y^{p-1}X - \tilde{\alpha}_i, \ X^{q^m} - X) \in \mathbb{F}_{q^m}[Y]$ for all $i \in \{1, \ldots, \tfrac{n}{p}\}$. $\qquad\square$

All the elements of the orbit of $b$ under the action of $\sigma$ are roots of the polynomial $P_b$ defined previously, i.e: the elements of the set $B := \{b, 2b, \ldots, (\ell - 1)b\}$. In practice, and according to computer aided experiment, the degree of the polynomial $P_b$ is $\ell$ and the set $B$ is exactly the set of its roots. Then there exist only $\ell < n$ possibilities for $b$, so we assume for now that we know the element $b$.

The second step is to recover the divisor $G$ from $\tilde{G}$. By Proposition 3.11, we know that $\tilde{G} = t\tilde{Q}$, where either $\tilde{Q} = ((\tfrac{\gamma_0}{\delta_0})^p - b^{p-1}\tfrac{\gamma_0}{\delta_0} : 1)$ or $\tilde{Q} = P_{\infty}$. Since we know $a$, we can recover the support of $G$ thanks to the support $\tilde{Q}$ of $\tilde{G}$.

Since we know $b$, we can recover the support of $G$ thanks to the support $\tilde{Q}$ of $\tilde{G}$.

---

**Algorithm 2:** Recover the divisor in the case $\sigma$ trigonalizable over $\mathbb{F}_{q^m}$

**Input** : The divisor $\tilde{G}$ of the invariant code $\mathcal{A}(\mathcal{P}, G)^{\sigma}$.
**Output:** Return $G$ and $B$ the set of possible values of $b$ in (13)

1 /* Recover $B := \{b, 2b, \ldots, (p-1)b\}$ */          // $\tilde{\mathcal{P}} = \{\tilde{\alpha}_i \mid i \in \{1, \ldots, \tfrac{n}{\ell}\}\}$
2 $R \leftarrow \gcd\big(\{\mathrm{Res}_X(X^p - Y^{p-1}X - \tilde{\alpha}_i, \ X^{q^m} - X) \mid i \in \{1, \ldots, \tfrac{n}{p}\}\}, \ Y^{q^m} - Y\big)$
3 $B \leftarrow \mathrm{roots}(R)$
4 /* Recover $G'$ from $\tilde{G}$ */
5 **if** $\tilde{Q} \neq P_{\infty}$ **then**          // $\tilde{G} = t * \tilde{Q}$, with $\tilde{Q} = (\tilde{\gamma} : 1)$
6     $\Gamma \leftarrow \mathrm{roots}(X^p - b^{p-1}X - \tilde{\gamma})$          // $b \in B$
7     $G' \leftarrow t \sum\limits_{\gamma \in \Gamma} (\gamma : 1)$
8 **else**
9     $G' = tP_{\infty}$
10 **return** $G', B$

---

**Proposition 4.2.** *Algorithm 2 finds the set $B$ and the divisor $G$ in $\mathcal{O}(n(q^m + p)^{\omega+1})$ operations in $\mathbb{F}_{q^m}$, where $\omega$ is the exponent of the linear algebra.*

*Proof.* We only prove the cost of the algorithm, its correctness is a consequence of Lemma 4.1 and Lemma 3.7.

The resultant of two polynomials can be computed with an "evaluation-interpolation" method, which reduces to compute determinants of scalar matrices and one interpolation. Here the degree of the resultant that we must compute is at most $(q^m + p)(p-1)$, so we must compute $(q^m + p)(p-1) + 1$ determinants with scalar coefficients. The only polynomial to evaluate here is $Y^{(p-1)}$, so the cost of the evaluation is $\mathcal{O}((q^m + p)(p-1)\log_2(p-1))$. Computing determinants costs $\mathcal{O}((q^m + p)^{\omega}((q^m + p)(p-1)))$ operations, where $\omega$ is the exponent of the cost of linear algebra. Then the interpolation cost $(q^m + p)^2(p-1)^2$ operations, using Lagrange interpolation, but this is negligible behind the cost of the previous determinants.

With Euclid Algorithm we can compute the gcd of two polynomials in $\mathbb{F}_{q^m}[Y]$ of degree at most $(q^m + p)(p-1)$ in $\mathcal{O}((q^m + p)p)^2)$ operations in $\mathbb{F}_{q^m}$. We compute at most $\frac{n}{p}$ resultants and gcd, so the cost of the first step is $\mathcal{O}(n(q^m + p)^{\omega+1})$ operations in $\mathbb{F}_{q^m}$.

The second step is negligible behind the first step event if we use Berlekamp algorithm. $\square$

The third step is to recover a support $\mathcal{P}'$ such as $\mathcal{A}(\mathcal{P}', G) = \mathcal{A}(\mathcal{P}, G)$ . By Proposition 3.11, we know that a point $P = (x : y)$ in $\mathcal{P}$ satisfies:

$$\begin{cases} x^p - b^{p-1}x - \tilde{\alpha}_i = 0 \\ y^p - \tilde{\beta}_i = 0 \end{cases}$$

for $i \in \{1, \ldots, \frac{n}{\ell}\}$, such that $(\tilde{\alpha}_i : \tilde{\beta}_i) = \tilde{P}_i$. Since we know $\tilde{\mathcal{P}}$, we are able to recover all elements of $\mathcal{P}$ but as an unordered set. Hence we know a support $\mathcal{P}'$ such as $\mathcal{A}(\mathcal{P}', G)$ is a permutation of the code $\mathcal{A}(\mathcal{P}, G)$.

## 4.2   Recover the permutation

At this point the problem is to recover the permutation between $\mathcal{A}(\mathcal{P}', G)$ and $\mathcal{A}(\mathcal{P}, G)$. Let $Gen_{\mathcal{A}(\mathcal{P}, G)}$ be a generator matrix of the code $\mathcal{A}(\mathcal{P}, G)$, and $H_{\mathcal{A}(\mathcal{P}', G)}$ be a parity check matrix of the code $\mathcal{A}(\mathcal{P}', G)$, the permutation between $\mathcal{A}(\mathcal{P}, G)$ and $\mathcal{A}(\mathcal{P}', G)$ is represented by matrix $\Pi$ such that

$$Gen_{\mathcal{A}(\mathcal{P}, G)}\Pi H_{\mathcal{A}(\mathcal{P}', G)}^T = 0. \tag{14}$$

If we have no assumption on the permutation between $\mathcal{P}'$ and $\mathcal{P}$, to find the permutation $\Pi$ we must resolve a linear system with $n^2$ unknowns, while (14) is a system of $k(n-k)$ linear equations which is not enough to find an unique solution.

Now we assume that we made the good choice for $a \in A$ (or $b \in B$). Then the permutation matrix $\Pi$ has the following form:

$$\Pi = \begin{pmatrix} \sum\limits_{i=1}^{\ell} x_{1,i}J^i & \left(0\right) \\ & \ddots & \\ \left(0\right) & \sum\limits_{i=1}^{\ell} x_{\frac{n}{\ell},i}J^i \end{pmatrix} \quad \text{where } J := \begin{pmatrix} 0 & \cdots & 0 & 1 \\ 1 & & & 0 \\ 0 & \ddots & & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix}. \tag{15}$$

$J$ is an $\ell \times \ell$ matrix, and $x_{j,i} \in \{0,1\}$ are unknowns, for $j \in \{1, \ldots, \frac{n}{\ell}\}$ and $i \in \{1, \ldots, \ell\}$. With this form we have $n$ unknowns. Assume that $n - k \leq \frac{n}{2}$, then $n \leq (n-k)k$. In this case, we can hope to find a unique solution for $\Pi$. In all our computer aided experiments we got a unique solution for $\Pi$. If the choice for $a \in A$ (or $b \in B$) is wrong, then there was no solution for the system in all our experiments.

We present an algorithm to recover the permutation matrix $\Pi$ and so the good choice for $a \in A$ (or $b \in B$). The algorithm is only written for the first case, where $\sigma$ is diagonalizable, but the other case is similar.

---

**Algorithm 3:** Recover the support

---

   **Input**   : A generator matrix of a quasi-cyclic alternant code: $Gen_{\mathcal{A}(\mathcal{P},G)}$, the divisor $G$,
               and the support $\tilde{\mathcal{P}}$ of the invariant code.

   **Output:** Returns FALSE if no solution is found. Else, returns TRUE and $\mathcal{P}'$ such that
         $\mathcal{A}(\mathcal{P}', G) = \mathcal{A}(\mathcal{P}, G)$

**1** **for** $i \in \{1, \ldots, \frac{n}{\ell}\}$ **do**

**2**    $\alpha_i \leftarrow \text{roots}(x^\ell - \tilde{\alpha}_i)[1]$                                            // cf (12)

**3**    $\beta_i \leftarrow \text{roots}(y^\ell - \tilde{\beta}_i)[1]$

**4** **forall** $a \in A$ **do**

**5**    /* Guess $\mathcal{P}'$ */

**6**    $\mathcal{P}' \leftarrow \{(a^j \frac{\alpha_i}{\beta_i} : 1) \mid j \in \{0 \ldots \ell - 1\}, i \in \{1 \ldots \frac{n}{\ell}\}\}$

**7**    $\mathcal{C} \leftarrow \mathcal{A}(P', G)$

**8**    **if** $\mathcal{C} = \mathcal{A}(\mathcal{P}, G)$ **then**

**9**       **return** TRUE, $\mathcal{P}'$

**10**    **else**

**11**       $H \leftarrow \text{ParityCheckMatrix}(\mathcal{C})$

**12**       $S \leftarrow \textbf{solve}(Gen_{\mathcal{A}(\mathcal{P},G)} \Pi H^T = 0$, with $\Pi$ a permutation matrix of the form (15))

**13**       **if** $|S| = 1$ **then**

**14**          **return** TRUE, $(\mathcal{P}' * \Pi)$

**15** **return** FALSE

---

**Proposition 4.3.** *Algorithm 3 finds a support $\mathcal{P}'$ such that $\mathcal{A}(\mathcal{P}', G) = \mathcal{A}(\mathcal{P}, G)$ in $\mathcal{O}(\ell n^2(n - k)k)$ operations in $\mathbb{F}_{q^m}$, where $n$ is the length of $\mathcal{A}(\mathcal{P}, G)$, $k$ is the dimension of $\mathcal{A}(\mathcal{P}, G)$, $\ell$ is the order of $\sigma$.*

*Proof.* We only prove the cost of the algorithm. We must to resolve a linear system of $(n - k)k$ equations with $n$ unknowns, this is possible in $\mathcal{O}(n^2(n - k)k)$ operations in $\mathbb{F}_{q^m}$. This step is repeated at most $\ell$ times so the cost is in $\mathcal{O}(\ell n^2(n - k)k)$ operations in $\mathbb{F}_{q^m}$. $\qquad\square$

In order to give practical running times for this part of the attack, we implemented Algorithm 3 in MAGMA [4]. The platform used in the experiments is a 2.27GHz Intel® Xeon® Processor E5520. For each set of parameters, we give the average time obtained after 10 tests. In the following table we use notation:

- $m$ : extension degree of the field of definition of the support and divisor over $\mathbb{F}_q$

- $n$ length of the quasi–cyclic code

- $k$ dimension of the quasi–cyclic code

- $\ell$ denotes the order of quasi–cyclicity of the code

- $w_{ISD}$ denotes the logarithm of the work factor for message recovery attacks. It is computed using **CaWoF** library [5].

| $q$ | $m$ | $n$ | $k$ | $\ell$ | $w_{ISD}$ | Algorithm 3 |
|---|---|---|---|---|---|---|
| 2 | 12 | 3600 | 2825 | 3 | 129 | 1659 s ($\approx$ 27 min) |
| 2 | 12 | 3500 | 2665 | 5 | 130 | 2572 s ($\approx$ 42 min) |
| 2 | 12 | 3510 | 2579 | 13 | 132 | 8848 s ($\approx$ 2h27) |

## 4.3 Case $\sigma$ diagonalizable in $\mathbb{F}_{q^{2m}} \backslash \mathbb{F}_{q^m}$

In this case, we recall that $\sigma = \rho \circ \sigma_d \circ \rho^{-1}$ with $\rho \in \mathrm{GL}_2(\mathbb{F}_{q^{2m}})$ and:

$$
\sigma_d \colon \quad \begin{array}{ccc} \mathbb{P}^1 & \to & \mathbb{P}^1 \\ (x : y) & \mapsto & (\alpha x : \alpha^q y), \end{array}
$$

where $\alpha \in \mathbb{F}_{q^{2m}}$ is an $\ell$-th root of unity. As $\sigma_d$ is diagonal in $\mathbb{F}_{q^{2m}}$, we can recover a support $\mathcal{P}'$ and a divisor $G'$ in $\mathbb{F}_{q^{2m}}$, using the same method as in Sections 4.1 and 4.2.

Now we want to recover a support $\mathcal{P}$ and a divisor $G$ in $\mathbb{F}_{q^m}$. We consider $\pi_\alpha := X + aX + b$ the minimal polynomial of $\alpha$, with $a, b \in \mathbb{F}_{q^m}$. Then:

$$
M_{\sigma_d} = \begin{pmatrix} \alpha & 0 \\ 0 & \alpha^q \end{pmatrix} \sim \begin{pmatrix} 0 & -b \\ 1 & -a \end{pmatrix} = M_{\sigma'}
$$

and there exist $\rho' \in \mathrm{GL}_2(\mathbb{F}_{q^{2m}})$ such that $\sigma_d = \rho' \circ \sigma' \circ \rho'^{-1}$, where $\sigma'$ is the element of $\mathrm{PGL}_2(\mathbb{F}_{q^m})$ associated to $M_{\sigma'}$. We can assume that $\sigma = \sigma'$, then we want to recover $\rho'$. Thanks to Section 4.2, we know $\alpha$ and it is easy to compute $a$ and $b$. To recover $\rho'$ it suffices to diagonalize the matrix $M_{\sigma'}$. From $\rho'$ and a support $\mathcal{P}'$ and a divisor $G'$ in $\mathbb{F}_{q^{2m}}$, we can recover a support $\mathcal{P} = \rho'^{-1}(\mathcal{P}')$ and a divisor $G = \rho'^{-1}(G')$ in $\mathbb{F}_{q^m}$.

## 5 Conclusion

To summarise, we showed that the key security of compact McEliece schemes based on alternant codes with some induced permutation is not better than the key security of the short code obtained from the invariant operation. A similar result was showed for permutations induced by the affine group with the folded code. Our new approach simplifies the reduction and extend the result to the projective linear group. Moreover, we present a simpler lifting from the invariant code to recover the original code, with method from linear algebra.

Another kind of quasi-cyclic alternant codes could be obtained from the action of the *semi-linear projective group* on the support. By semilinear projective group, we mean transformation of the form: $x \mapsto \frac{ax^{q^i} + b}{cx^{q^i} + d}$, with $a, b, c, d \in \mathbb{F}_{q^m}^n$. These transformations induce a permutation on the alternant code $\mathcal{C} \cap \mathbb{F}_q^n$ but not on the GRS code $\mathcal{C}$. So we cannot use the same property of the invariant of a GRS code to study this kind of quasi-cyclic alternant code.

We can notice that key-recovery is generally more expensive than message recovery. With a good choice of parameters it is still possible to construct quasi-cyclic codes with high complexity of key recovery attack on the invariant code.

# References

[1] Thierry P. Berger, *Goppa and related codes invariant under a prescribed permutation*, IEEE Trans. Inform. Theory **46** (2000), no. 7, 2628–2633.

[2] _____, *On the cyclicity of Goppa codes, parity-check subcodes of Goppa codes and extended Goppa codes*, Finite Fields Appl. **6** (2000), no. 3, 255–281.

[3] Thierry P. Berger, Pierre-Louis Cayrel, Philippe Gaborit, and Ayoub Otmani, *Reducing key length of the McEliece cryptosystem*, International Conference on Cryptology in Africa, Springer, 2009, pp. 77–97.

[4] Wieb Bosma, John Cannon, and Catherine Playoust, *The Magma algebra system I: The user language*, J. Symbolic Comput. **24** (1997), no. 3/4, 235–265, http://dx.doi.org/10.1006/jsco.1996.0125.

[5] Rodolfo Canto Torres, *CaWoFa, C library for computing asymptotic exponents of generic decoding work factors*, 2016, `https://gforge.inria.fr/projects/cawof/`.

[6] Arne Dür, *The automorphism groups of Reed-Solomon codes*, J. Combin. Theory Ser. A **44** (1987), 69–82.

[7] Jean-Charles Faugère, Ayoub Otmani, Ludovic Perret, Frédéric de Portzamparc, and Jean-Pierre Tillich, *Folding alternant and Goppa codes with non-trivial automorphism groups*, IEEE Trans. Inform. Theory **62** (2016), no. 1, 184–198.

[8] _____, *Structural cryptanalysis of McEliece schemes with compact keys*, Des. Codes Cryptogr. **79** (2016), no. 1, 87–112.

[9] Jean-Charles Faugere, Ayoub Otmani, Ludovic Perret, and Jean-Pierre Tillich, *Algebraic cryptanalysis of McEliece variants with compact keys*, Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2010, pp. 279–298.

[10] William Fulton, *Algebraic curves: an introduction to algebraic geometry*, Addison-Wesley Redwood City California, 1989.

[11] Philippe Gaborit, *Shorter keys for code based cryptography*, Proceedings of the 2005 International Workshop on Coding and Cryptography (WCC 2005), 2005, pp. 81–91.

[12] V. D. Goppa, *Codes on algebraic curves*, Dokl. Akad. Nauk SSSR **259** (1981), no. 6, 1289–1290.

[13] Venkatesan Guruswami, *Linear-algebraic list decoding of folded Reed-Solomon codes*, Computational Complexity (CCC), 2011 IEEE 26th Annual Conference on, IEEE, 2011, pp. 77–85.

[14] Pierre Loidreau, *Codes derived from binary Goppa codes*, Probl. Inf. Transm. **37** (2001), no. 2, 91–99.

[15] Florence J. MacWilliams and Neil J. A. Sloane, *The theory of error-correcting codes*, fifth ed., North–Holland, Amsterdam, 1986.

[16] Robert J. McEliece, *A public-key system based on algebraic coding theory*, pp. 114–116, Jet Propulsion Lab, 1978, DSN Progress Report 44.

[17] Rafael Misoczki and Paulo S.L.M. Barreto, *Compact McEliece keys from Goppa codes*, International Workshop on Selected Areas in Cryptography, Springer, 2009, pp. 376–392.

[18] Henning Stichtenoth, *On automorphisms of geometric Goppa codes*, Journal of Algebra **130** (1990), no. 1, 113–121.

[19] ———, *Algebraic function fields and codes*, Universitext, Springer, 1993.