# Structural attack on quasi-cyclic SSAG-code-based McEliece cryptosystems

Elise Barelli*
(Philippe Lebacque?)
Mathieu Lhotel[†](✉)
Hugues Randriam[‡]

May 3, 2022

**Abstract.** *In this paper, we present a structural attack on quasi-cyclic SSAG-code-based McEliece cryptosystems, by showing that the knowledge of the invariant code allows us to recover the secret data. In particular, this shows that the security of such systems must rely on the security of the invariant code. We then propose a McEliece-like scheme based on quasi-cyclic Hermitian codes, and describe known attacks against the invariant code. We conclude by providing some parameters that resist those attacks.*

*Si la sécurité du système repose sur celle du sous-code invariant, pourquoi ne pas définir directement un McEliece avec le code invariant ? En termes de taille de clé publique j'imagine que ça revient au même (à vérifier) ? Cela étant on parle ici de sécurité contre une attaque en reconstruction de clé. Peut-être que pour l'attaque en reconstruction de message clair, il y a une différence ? Dans tous les cas ça mériterait d'être précisé et discuté (pas dans l'abstract évidemment).*

*

[†]Laboratoire de Mathématiques de Besançon, UMR 6623 CNRS, Université de Bourgogne Franche-Comté, France
[‡]ANSSI, Laboratoire de Cryptographie & LTCI, Télécom Paris

# 1 Introduction

Expliquer pourquoi on ne considère pas l'invariant comme code publique directement.

In the area of post-quantum cryptography, public-key cryptosystems using linear codes look promising. The first such system, based on binary classical Goppa codes, was introduced by McEliece in 1978 [McE78]. Its main drawback is that the size of the corresponding public keys (that are generating matrices of codes) is too large for practical use cases. Many propositions were made in order to correct this, mostly by considering codes with additional structure, e.g. quasi-cyclic codes. Moreover, replacing classical Goppa codes (defined over the projective line) by their natural generalization, AG-codes on curves, can also help in providing more flexibility.

However, recent works from Couvreur, Marquez-Corbella and Pellikaan [CMCP14] broke McEliece cryptosystems based on raw AG-codes on arbitrary genus curves. So, in the same way that classical Goppa codes can be seen as subfield subcodes of GRS codes, this leads to more specifically consider subfield subcodes of AG-codes (SSAG in short), for which there are only few propositions to date.

The present work describes a structural attack on McEliece-like scheme based on structured SSAG-codes. In fact, we show that the security of these systems can be reduced to the security of the so-called invariant code, which can be constructed from the public key. Depending on the assumption of the scheme, this subcode, which was first introduced by [Loi01], leaks too much information on the secret data about the public codes, which can be recovered from it. As a consequence, the parameters of the underlying scheme must be chosen carefully in order to keep a good security level.

As a countermeasure to this attack, we also propose a scheme based on SSAG-codes from the Hermitian curve, and provide a set of parameters to secure the corresponding invariant code.

In Section 2, we will give some classical notations and definitions what will be useful, both in algebraic and coding theory. Section 3 will be devoted to the invariant code. In Section 4, we detail our attack in a general framework. Section 5 gives some applications, such that Kummer or Artin-Schreier cases. Section 6 will discuss the generalization of the attack to any cover with solvable Galois group. Finally, we will propose in Section 7 a scheme using QC-SSAG codes on the Hermitian curve which resists our attack, and describe known attacks against the invariant code and countermeasures.

# 2 Notations and properties

## 2.1 Algebraic function fields

Let $\mathbb{F}_{q^m}$ be the finite field with $q^m$ elements, were $q = p^s$ is a power of a prime $p$ and $m \geq 1$. A function field of one variable over $\mathbb{F}_{q^m}$ is a field $K$ with the property that there exists an element $x \in K$ such that $K/\mathbb{F}_{q^m}(x)$ is a finite separable extension.

In our context, an algebraic function field $K$ over $\mathbb{F}_{q^m}$ will always be seen as the field of rational functions of an algebraic curve $\mathcal{Y}$ defined over $\mathbb{F}_{q^m}$, that is, with classical notations, $K = \mathbb{F}_{q^m}(\mathcal{Y})$.

Denote by $\mathbb{P}_K$ the set of places of $K$. Any place $P \in \mathbb{P}_K$ comes with its valuation ring $O_P$ and its discrete valuation $\nu_P : K \to \mathbb{Z} \cup \{\infty\}$ . The degree of the place $P$, denoted by $\deg(P)$ is defined as the integer $\deg(P) := [O_P/P : \mathbb{F}_{q^m}] < \infty$. The divisor group of $K$, denoted $\mathrm{Div}(K)$, is the free abelian group on $\mathbb{P}_K$ generated by

$$A = \sum_{P \in \mathrm{Supp}(A)} \nu_P(A) \cdot P,$$

where $\mathrm{Supp}(A)$ is a finite subset of $\mathbb{P}_K$, called the support of $A$, whose elements are places such that $\nu_P(A) \in \mathbb{Z}\backslash\{0\}$.
A place $P \in \mathrm{Supp}(A)$ is called a zero of $A$ if $\nu_P(A) > 0$ (resp. a pole of $A$ if $\nu_P(A) < 0$). For a function $z \in K$, we denote by $(z)^K$, $(z)_0^K$ and $(z)_\infty^K$ its assiciated divisor, divisor of its zeros and divisor of its poles respectively, that is $(z)^K = (z)_0^K - (z)_\infty^K$, where

$$(z)_0^K = \sum_{\nu_P(A)>0} \nu_P(A) \cdot P \quad \text{and} \quad (z)_\infty^K = \sum_{\nu_P(A)<0} -\nu_P(A) \cdot P.$$

The degree of a divisor is naturally defined by the formula

$$\deg(A) = \sum_{P \in \mathrm{Supp}(A)} \nu_P(A) \cdot \deg(P).$$

We will denote $\mathrm{Princ}(K)$ the subgroup of $\mathrm{Div}(K)$ made of principal divisors and $\mathrm{Div}^0(K)$ the subgroup of degree zero divisors. The divisor class group of $K$ is defined by $\mathrm{Cl}(K) := \mathrm{Div}(K)/\mathrm{Princ}(K)$ and the group of divisor classes of degree zero by $\mathrm{Cl}^0(K) := \mathrm{Div}^0(K)/\mathrm{Princ}(K)$. Let $h(K) := \#\,\mathrm{Cl}^0(K)$ be the *class number* of $K$. Then for any $r \geq 1$, the number of divisor classes in $\mathrm{Cl}(K)$ of degree $r$ does not depend on $r$, and is equal to $h(K)$. The degree map is a surjective map from $\mathrm{Cl(K)} \to \mathbb{Z} \to \{0\}$. Finally, the Weil bound implies

**Theorem 1** (see [TaDN07], Proposition 3.1.23)**.** *Let $K$ be an algebraic function field over $\mathbb{F}_{q^m}$ with genus $g(K)$. Then the class number $h(K)$ satisfies*

$$(\sqrt{q^m} - 1)^{2g(K)} \leq h(K) \leq (\sqrt{q^m} + 1)^{2g(K)}.$$

Given a divisor $A \in \mathrm{Div}(K)$, its Riemann-Roch space is defined as the $\mathbb{F}_q$-vector space

$$\mathcal{L}_K(A) = \{z \in K \mid (z)^L \geq -A\} \cup \{0\},$$

of dimension $\ell(A) := \dim_{\mathbb{F}_q}(\mathcal{L}_K(A))$. If there is nos ambiguity with the function field, we will just denote $\mathcal{L}(A)$.

Given a tower of $\mathbb{F}_{q^m}(x) \subseteq K \subseteq L$ of function fields over $\mathbb{F}_{q^m}$, let us consider a place $P \in \mathbb{P}_K$ and one of its extension $Q \in \mathbb{P}_L$, denoted by $Q|P$. The ramification index will as usual be denoted by $e(Q|P)$.
Let $\pi : \mathcal{Y} \to \mathcal{X}$ be a separable morphism between two curves $\mathcal{X}$ and $\mathcal{Y}$ over $\mathbb{F}_{q^m}$, whose function fields are given by $L := \mathbb{F}_{q^m}(\mathcal{Y})$ and $K := \mathbb{F}_{q^m}(\mathcal{X})$. We define the pullback of a place $P \in \mathbb{P}_K$ as the divisor

$$\pi^* P = \sum_{Q|P} e(Q|P) \cdot Q \in \mathrm{Div}(L).$$

Note that this definition extends to pullback of divisors by linearity.
Throughout the paper, we will make great use of the following lemma, which shows that pullbacks preserve the notion of principal divisors.

**Lemma 2.** *Let $z \in L$ be a function. Then*

$$(z)^L = \pi^*(z)^K, \ (z)_0^L = \pi^*(z)_0^K \text{ and } (z)_\infty^L = \pi^*(z)_\infty^K.$$

*Proof.* see [Sti09], Proposition 3.1.9. □

## 2.2 Coding theory

As explained in the introduction, we will be dealing with SSAG-code, that is our AG-codes will be defined over some extension $\mathbb{F}_{q^m}$ of $\mathbb{F}_q$. Let $q = p^s$ be a prime power, and $m \geq 1$ be an integer. Let us recall the definition of an AG-code.

**Definition 3.** Let $\mathcal{Y}$ be a smooth irreducible projective curve over $\mathbb{F}_{q^m}$ with function field $L = \mathbb{F}_{q^m}(\mathcal{Y})$, $\mathcal{Q} = \{Q_1, ..., Q_n\}$ be a set of $n$ distinct places of degree one in $L$ and $G \in \mathrm{Div}(L)$ be a divisor such that $\mathrm{Supp}(G) \cap \mathcal{Q} = \emptyset$. Suppose that $\deg(G) < n$. Then the AG-code associated to the triple $(\mathcal{Y}, \mathcal{Q}, G)$ is defined as the $\mathbb{F}_{q^m}$-vector space

$$C_{\mathcal{L}}(\mathcal{Y}, \mathcal{Q}, G) = \{(f(Q_1), ..., f(Q_n)) \mid f \in \mathcal{L}(G)\} \subseteq \mathbb{F}_{q^m}^n.$$

**Definition 4.** With notation as above, we define the subfield subcode of $C_{\mathcal{L}}(\mathcal{Y}, \mathcal{Q}, G)$ over $\mathbb{F}_q$, denoted $\mathrm{SSAG}_q(\mathcal{Y}, \mathcal{Q}, G)$, as follows

$$\mathrm{SSAG}_q(\mathcal{Y}, \mathcal{Q}, G) = C_{\mathcal{L}}(\mathcal{Y}, \mathcal{Q}, G) \cap \mathbb{F}_q^n.$$

In particular, we will be interested in "structured" codes, which are codes with non trivial permutation group. To be concrete, consider an automorphism subgroup $\Sigma \subseteq \mathrm{Aut}(L)$, and denote by $\mathrm{Orb}_\Sigma(Q)$ the orbit of a place $Q \in \mathbb{P}_L$ under the action of this automorphism group. If $\Sigma(\mathcal{Q}) = \mathcal{Q}$ and $\Sigma(G) = G$, then $\Sigma$ induces a permutation $\tilde{\Sigma}$ on the code $C_{\mathcal{L}}(\mathcal{Y}, \mathcal{Q}, G)$. In order to satisfy this property, the support $\mathcal{Q}$ and the divisor $G$ will be constructed in the follwing way: $\mathcal{Q}$ will be a union, and $G$ a sum, of orbits under the action of $\Sigma$. With this satisfied, it is clear that they both are $\Sigma$-invariant since each orbit of places in $L$ is. Note that as a subfield subcode, the permutation $\tilde{\Sigma}$ also acts on the subfield subcode $\mathrm{SSAG}_q(\mathcal{Y}, \mathcal{Q}, G) := C_{\mathcal{L}}(\mathcal{Y}, \mathcal{Q}, G) \cap \mathbb{F}_q^n$.
Throughout the rest of the paper, we will only consider the case where $\Sigma$ is the Galois group of some extension of function fields $L/K$. In this case, $K$ will be the function field of the quotient curve $\mathcal{Y}/\Sigma$. This settings will be detailed in Section 4 .

Let us also recall the definition of the dual of a linear code:

**Definition 5.** Let $\mathcal{C}$ be a $[n, k]$-linear code over a finite field $\mathbb{F}_{q^m}$. Its dual code is defined by

$$\mathcal{C}^\perp := \{y \in \mathbb{F}_{q^m}^n \mid xy^T = 0 \ , \ \forall\, x \in \mathcal{C}\}.$$

It is easy to see that any generator matrix of $\mathcal{C}^\perp$ is a parity check matrix of $\mathcal{C}$. In particular, $\mathcal{C}^\perp$ is a $[n, n-k]$-code over $\mathbb{F}_{q^m}$.

Finally, we will also need a result using duality of AG-codes.

**Proposition 6.** *Let $C_{\mathcal{L}}(\mathcal{Y}, \mathcal{Q}, G)$ be an* AG*-code defined on a curve $\mathcal{Y}$. Then there exists a divisor $G' \in \mathrm{Div}(\mathcal{Y})$ such that*

$$C_{\mathcal{L}}(\mathcal{Y}, \mathcal{Q}, G)^\perp = C_{\mathcal{L}}(\mathcal{Y}, \mathcal{Q}, G').$$

*Moreover, we have $G' = \sum_{Q \in \mathbb{Q}} Q - G + (\eta)$, where $\eta$ is some Weil differential. (utile ?)*

*Proof.* See [Sti09], Proposition 2.2.10. □

# 3 Invariant code

**Definition 7.** Given an AG-code $\mathcal{C} = C_{\mathcal{L}}(\mathcal{Y}, \mathcal{Q}, G)$ on a curve $\mathcal{Y}$ with function field $L$, such that $\mathcal{Q}$ and $G$ are invariant under the action of $\Sigma \subseteq \mathrm{Aut}(L)$, we define its invariant code as the subcode

$$\mathcal{C}^\Sigma := \{c \in \mathcal{C} \mid \tilde{\Sigma}(c) = c\},$$

where $\tilde{\Sigma}$ is the permutation induced by $\Sigma$ on the code.

Note that this code has repeated entries (ie. all its words are equal on each orbit under $\Sigma$), so we usually use another one: the *punctured invariant code*. It is denoted $\overline{\mathcal{C}}^\Sigma$, and is optained from $\mathcal{C}^\Sigma$ by keeping only one entry on each orbit of length $\#\Sigma$. Later on, we will use the punctered invariant code, but we will keep the notation $\mathcal{C}^\Sigma$.

**Remark 8.** Suppose $\Sigma = \langle\sigma\rangle$ is cyclic of order $\ell$, and denote by $\sigma_{\mathcal{C}}$ the $\ell$-quasi-cyclic shift restricted to $\mathcal{C}$. Let also $I_\ell := \{1, ..., n\} \backslash \{1, \ell+1, ..., n-\ell+1\}$, where $n = \mathrm{Length}(\mathcal{C})$. Then the punctered invariant code can also be defined as

$$\overline{\mathcal{C}}^\Sigma := \mathrm{Punct}_{I_\ell}(\mathcal{C}^\Sigma) = \mathrm{Punct}_{I_\ell}(\ker(\sigma_{\mathcal{C}} - id)).$$

As a consequence, it is possible to build in polynomial time in the parameters of $\mathcal{C}$, a generator matrix of the invariant code from the knowledge of a generator matrix of $\mathcal{C}$ and the induced permutation.

In what follows, we will study the structure of the invariant code, which will be a important result for the next parts. In particular, we will show that this subcode is nothing but an AG-code itself, defined on the quotient curve $\mathcal{Y}/\Sigma$, whose function field is precisly the fixed field $K := L^{\Sigma}$ of $L$. Let us start with the following lemma.

**Lemma 9.** *Let $\mathcal{Q}$ and $G$ be as in Definition 3, and suppose that they are invariant under an automorphism $\sigma \in \Sigma \subseteq \mathrm{Aut}(L)$. If $c = \mathrm{Ev}_{\mathcal{Q}}(g) \in C_{\mathcal{L}}(\mathcal{Y}, \mathcal{Q}, G)$ is such that $\sigma(c) = c$, then $g$ is $\sigma$-invariant, ie. $g \circ \sigma = g$.*

*Proof.* Let us write $\mathcal{Q} = \{Q_1, ..., Q_n\}$, and let $c = (g(Q_1), ..., g(Q_n))$ be such that $\sigma(c) = c$. Then we have

$$\begin{aligned} \forall\ i \in \{1, ..., n\}\ ,\ g(Q_{\sigma(i)}) = g(Q_i) &\iff \forall\ i \in \{1, ..., n\}\ ,\ (g \circ \sigma)(Q_i) = g(Q_i) \\ &\iff \forall\ i \in \{1, ..., n\}\ ,\ (g \circ \sigma - g)(Q_i) = 0. \end{aligned}$$

Since the divisor $G$ is $\sigma$-invariant, we have $g \circ \sigma \in \mathcal{L}(G)$, and thus $g \circ \sigma - g \in \mathcal{L}(G)$ also. However the equivalence above gives $n > \deg(G)$ zeroes to this function. This implies $(g \circ \sigma - g) \equiv 0$, and the result follows. □

**Definition 10.** Consider an extension $L/K$ of function field such that $K = L^{\Sigma}$ For a divisor $G \in \mathrm{Div}(L)$, let us denote by $\tilde{G} \in \mathrm{Div}(K)$ the largest divisor (according to the degree) such that

$$\pi^* \tilde{G} \leq G.$$

**Remark 11.** If $A \in \mathrm{Div}(K)$, we have

$$\widetilde{\pi^* A} = A.$$

**Proposition 12.** *Let $L/K$ be as above and $\sigma \in \mathrm{Aut}(L)$ an automorphism. Let also $G \in \mathrm{Div}(L)$ be a $\sigma$-invariant divisor, that is there exists $s \in \mathbb{N}^*$ and places $S_1, ..., S_s \in \mathbb{P}_L$ such that*

$$G = \sum_{i=1}^{s} t_i \sum_{S \in \mathrm{Orb}_{\sigma}(S_i)} S,$$

*for some $t_i \in \mathbb{Z} \backslash \{0\}$. Then*

*(i) The divisor $\tilde{G}$ associated to $G$ (see Definition 10) is given by*

$$\tilde{G} := \sum_{i=1}^{s} \left\lfloor \frac{t_i}{e(S_i|R_i)} \right\rfloor R_i \in \mathrm{Div}(K),$$

*where $S_i|R_i$ for all $1 \leq i \leq s$;*

*(ii) $\mathcal{L}_L(G)^{\sigma} = \mathcal{L}_K(\tilde{G})$.*

*Proof.* *(i)* By definition of the pullback of a divisor, we have

$$\pi^* \tilde{G} = \sum_{i=1}^{s} e(S_i|R_i) \cdot \left\lfloor \frac{t_i}{e(S_i|R_i)} \right\rfloor \sum_{S \in \mathrm{Orb}_{\sigma}(S_i)} S \leq G,$$

which gives the result.

(*ii*) Let $g \in \mathcal{L}_L(G) \subseteq L$ be such that $g \circ \sigma = g$ (that is $g \in \mathcal{L}_L(G)^\sigma \subseteq L^\sigma = K$). For each $i \in \{1, ..., s\}$, consider a place $R_i \in \mathbb{P}_{L^\sigma}$ be such that $S_i | R_i$. It is well known that for every $S \in \mathrm{Orb}_\sigma(S_i)$, one also have $S|R_i$ and $e(S|R_i) = e(S_i|R_i)$. Since $g \in \mathcal{L}_L(G)$, we know by definition that

$$(g)_L \geq -G = -\sum_{i=1}^{s} t_i \sum_{S \in \mathrm{Orb}_\sigma(S_i)} S$$

Note that for every $i \in \{1, ..., s\}$, we have $e(S_i|R_i) \cdot \nu_{R_i}(g) = \nu_{S_i}(g)$, so we have

$$(g)_{L^\sigma} \geq -\sum_{i=1}^{s} \frac{t_i}{e(S_i|R_i)} R_i \geq -\tilde{G},$$

which implies $g \in \mathcal{L}_K(\tilde{G}) \subseteq L^\sigma$. Hence $\mathcal{L}_L(G)^\sigma \subseteq \mathcal{L}_K(\tilde{G})$.
Conversly, let $g \in L^\sigma$ such that $g \in \mathcal{L}_K(\tilde{G})$, with $\tilde{G}$ defined as above. We have

$$(g)_L = \pi^*(g)_{L^\sigma} \geq -\sum_{i=1}^{s} e(S_i|R_i) \cdot \left\lfloor \frac{t_i}{e(S_i|R_i)} \right\rfloor \sum_{S \in \mathrm{Orb}_\sigma(S_i)} S \geq -G,$$

and thus $g \in \mathcal{L}_L(G) \cap L^\sigma = \mathcal{L}_L(G)^\sigma$.

□

**Theorem 13** (Structure of the invariant code)**.** *Let $\mathcal{C} := C_{\mathcal{L}}(\mathcal{Y}, \mathcal{Q}, G)$ be an* AG*-code defined on a curve $\mathcal{Y}$ with function field $L$, invariant under the action of $\Sigma \subseteq \mathrm{Aut}(L)$. Then its invariant code is also an* AG*-code, defined on the quotient curve $\mathcal{Y}/\Sigma$. In particular, there exist a support $\mathcal{P}$ and a divisor $\tilde{G}$ on $\mathcal{Y}/\Sigma$ such that*

$$\mathcal{C}^\Sigma = C_{\mathcal{L}}(\mathcal{Y}/\Sigma, \mathcal{P}, \tilde{G}).$$

*Proof.* Straightforward consequence of Lemma 9 and Proposition 12. □

**Remark 14.** The above theorem can be made a bit more precise, since we can make $\mathcal{P}$ and $\tilde{G}$ explicit. Indeed, the divisor $\tilde{G}$ is nothing but the one introduced Proposition 12 (*ii*), and

$$\mathcal{P} := \{P \in \mathbb{P}_{L^\Sigma} \ ; \ Q|P\} = \{Q \cap L^\Sigma \ , Q \in \mathcal{Q}\}$$

In particular, they can be described by using the ramification in the cover of curves $\mathcal{Y} \longrightarrow \mathcal{Y}/\Sigma$.

**Remark 15.** Let $\mathcal{C}$ be an AG-code on a $\mathcal{Y}$ with function field $L$, invariant under $\Sigma \subseteq \mathrm{Aut}(L)$, then

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

that is invariant and subfield subcode operations commute.

**Corollary 16.** *With the notations of Theorem 13, let $\mathrm{SSAG}_q(\mathcal{Y}, \mathcal{Q}, G)$ be a subfield subcode of an AG-code with an action of $\Sigma$. Then*

$$\mathrm{SSAG}_q(\mathcal{Y}, \mathcal{Q}, G)^\Sigma = \mathrm{SSAG}_q(\mathcal{Y}/\Sigma, \mathcal{P}, \tilde{G}),$$

*where $\mathcal{P}$ and $\tilde{G}$ are defined as in Remark* 14.

*Proof.* Immediate consequence of Theorem 13 and Remark 15. □

# 4 Recovering the equation of a Galois cover

Throughout all this section, we work in the finite field $\mathbb{F}_{q^m}$, with $m \geq 1$ and $q = p^s$ a prime power. Let us consider a Galois cover

$$\pi : \mathcal{Y} \longrightarrow \mathcal{X}$$

between curves defined over $\mathbb{F}_{q^m}$. It corresponds to a Galois extension of function fields $K \subseteq L$, where $K = \mathbb{F}_{q^m}(\mathcal{X})$ and $L = \mathbb{F}_{q^m}(\mathcal{Y})$. In particular, there exists a primitive element $y \in L$ such that $L = K(y)$ and

$$H(y) = 0 \ , \ H \in K[T] \text{ irreducible polynomial.}$$

The key part of our attack will be to recover a defining equation of the curve $\mathcal{Y}$, that is the minimal polynomial $H$ of $y$ over $K$. To this end, let us introduce the following concept.

Let us give the setting of our attack. Denote by $\ell = [L : K]$ the degree of the extension $L/K$ (*ie.* the order of its Galois group). Suppose that we are given a set of $r$ places of degree one in $K$, say $\mathcal{P} = \{P_1, ..., P_r\}$, that totally split in $L/K$. For any $1 \leq i \leq r$, we then have

$$\pi^* P_i = Q_{i,1} + ... + Q_{i,\ell} \text{ with } Q_{i,j} \in \mathbb{P}_L.$$

In particular, it meeans that for all $1 \leq i \leq r$, we have $\{Q_{i,1} + ... + Q_{i,\ell}\} = \mathrm{Orb}_{\mathrm{Gal}(L/K)}(Q_{i,1})$. Denote by $\mathcal{Q} = \{Q_{i,j} \mid 1 \leq i \leq r \text{ and } 1 \leq j \leq \ell\}$ the set of all extensions of the $P_i$'s in $L$. Let $G \in \mathrm{Div}(L)$ be an invariant divisor as in Proposition 12, such that $\mathrm{Supp}(G) \cap \mathcal{Q} = \emptyset$ and $\deg(G) < n := \ell r$. Also, we denote by $\tilde{G} \in \mathrm{Div}(K)$ the corresponding divisor given by Proposition 12, $(i)$, which implies that $\mathrm{Supp}(\tilde{G}) \cap \mathcal{P} = \emptyset$.

**Remark 17** (À garder ?)**.** The situation above can be described in terms of AG-codes. In fact, the quotient curve $\mathcal{X}$ is nothing but the curve $\mathcal{Y}/\Sigma$, where $\Sigma := \mathrm{Gal}(L/K)$. Moreover, it is clear that both the support $\mathcal{Q}$ and $G$ are invariant under the action of $\Sigma$, and thus $\Sigma$ induces a permutation on the AG-code $C_{\mathcal{L}}(\mathcal{Y}, \mathcal{Q}, G)$, and its invariant code is given by $C_{\mathcal{L}}(\mathcal{Y}/\Sigma, \mathcal{P}, \tilde{G})$. As explained in the introduction, we will later be dealing with the subfield subcode version of those codes.

Before describing the procedure to recover the equation of $\mathcal{Y}$, let us put together our assumptions :

1. We know a parity check matrix $\mathbf{H}$ of the SSAG-code

   $$\mathcal{C} = \mathrm{SSAG}_q(\mathcal{Y}, \mathcal{Q}, G),$$

   that can be computed from the public key;

2. We know a plane model of the quotient curve $\mathcal{X}$ (ie. a defining equation of the function field $K$), the finte set of places $\mathcal{P}$ of $K$ and the divisor $\tilde{G} \in \mathrm{Div}(K)$ (that is exactly the invariant code);

3. We know the morphism $\pi : \mathcal{Y} \longrightarrow \mathcal{X}$ at the level of the set of places $\mathcal{Q}$, that is for every (unknown) $Q \in \mathcal{Q}$, we know the corresponding place $P \in \mathcal{P}$ such that $Q|P$;

4. We have *"enough information"* on the pole divisor of $y$ in $K$, where $y \in L$ is such that $L = K(y)$. This assumption will be discussed later, since the key point of the attack will be to control the divisor

   $$\widetilde{(y)_\infty^L} \in \mathrm{Div}(K).$$

   In fact, we will need to understand its support in $K$, and how it ramifies in $L/K$.

Let us now explain what we plan to do. The main idea is first to recover the finite set of points $\mathcal{Q}$ in $\mathcal{Y}$, in order to be able to recover one of its defining equation using interpolation. Thanks to hypothesis 2, we know the coordinates of the rational points corresponding to $P_i$'s in a plane model of the curve $\mathcal{X}$. In fact, let us denote by $\alpha$ a primitive element of $K$ over $\mathbb{F}_{q^m}(x)$, that is $K = \mathbb{F}_{q^m}(x, \alpha)$ (possible since $K/\mathbb{F}_{q^m}(x)$ is separable and algebraic). Then one can denote by $(x(P_i) : \alpha(P_i) : 1)$ the coordinate of the rational point in $\mathcal{X}(\mathbb{F}_{q^m})$ corresponding to the place $P_i \in \mathcal{P}$.

As the curve $\mathcal{Y}$ covers the plane model of $\mathcal{X}$, it admits a model in $\mathbb{P}^3$; that is any $Q \in \mathcal{Q}$ corresponds to a point with coordinates $(x(Q) : \alpha(Q) : y(Q) : 1) \in \mathbb{P}^3(\mathbb{F}_{q^m})$. Since places in $\mathcal{Q}$ are extensions of

places in $\mathcal{P}$, they correspond to points that have the same $x$ and $\alpha$ coordinates, equal to those of their restriction in $K$. In other words, for all $1 \leq i \leq r$ and $1 \leq j \leq \ell$, the place $Q_{i,j} \in \mathcal{Q}$ corresponds to the rational point

$$(x(P_i) : \alpha(P_i) : y(Q_{i,j}) : 1) \in \mathcal{Y}(\mathbb{F}_{q^m}).$$

As a result, from hypothesis 2, one only needs to recover the $y$-evaluation of points in $\mathcal{Q}$ in order to conclude. So the key part will be to recover the row vector

$$\mathbf{y} = (y_{i,j})_{i,j}, \tag{1}$$

where $y_{i,j} := y(Q_{i,j})$, for every $1 \leq i \leq r$ and $1 \leq j \leq \ell$.

In order to recover $\mathbf{y}$, we will construst a linear system of which it is solution. For that, recall that by definition, the parity check matrix of the code $\mathcal{C} = \mathrm{SSAG}_q(\mathcal{Y}, \mathcal{Q}, G)$ satisfies

$$c \in \mathcal{C} \iff \mathbf{H}c^T = 0. \tag{2}$$

Moreover, we know that a codeword $c \in \mathcal{C}$ comes from evaluation at $Q_{i,j} \in \mathcal{Q}$ of functions in the Riemann-Roch space of $G$. In particular, there exists a function $f \in \mathcal{L}_L(G)$ such that

$$c = (f(Q_{i,j}))_{i,j}$$

Of course, $\mathcal{L}_L(G)$ is unknown since the divisor $G$ is as well. But we actually do not need the whole space $\mathcal{L}_L(G)$ to recover $\mathbf{y}$. In fact, we are searching for a subspace $\mathcal{L} \subseteq \mathcal{L}_L(G)$, big enough (we will explain it later), and made of functions that specifically have the form $g \cdot y$, where $g \in K$ and $y$ is such that $L = K(y)$. Once such a space is found, we have

$$\{c = (g(Q_{i,j}) \cdot y(Q_{i,j})) \ , \ 1 \leq i \leq r \ , \ 1 \leq j \leq \ell \text{ and } f \cdot y \in \mathcal{L}\} \cap \mathbb{F}_q^n \subseteq \mathcal{C}.$$

In particular, since $g \in K$, the numbers $g(Q_{i,j})$ are known for all $i, j$ (recall the discussion above), and thus the right-hand side of (2) gives us a system where everything is known exept for the $y(Q_{i,j})$.

Let us now explain how to get a space of functions $\mathcal{L} \subseteq \mathcal{L}_L(G)$ as above. In particular, since we know the quotient curve (that is we know its function field $K$) as well as the morphism $\pi : \mathcal{Y} \to \mathcal{X}$, we will construct $\mathcal{L}$ as a set of pull-backs of functions in $K$. To be concrete, we are searching for a space of functions $\mathcal{F} \subseteq K$, as big as possible, such that for all $f \in \mathcal{F}$, we have

$$(\pi^* f) \cdot y \in \mathcal{L}_L(G). \tag{3}$$

The following lemma shows that a good choice is to consider a Riemann-Roch space of a specific divisor $D \in \mathrm{Div}(K)$.

**Lemma 18.** *Let $D := \tilde{G} - \widetilde{(y)_\infty^L} \in \mathrm{Div}(K)$. Then we have*

$$\mathcal{F} := \mathcal{L}_K(D) \subseteq \mathcal{L}_K(\tilde{G}),$$

*and $\mathcal{F}$ satisfies condition (3) above.*

*Proof.* The inclusion $\mathcal{F} \subseteq \mathcal{L}_K(\tilde{G})$ easy follows from the fact that $\widetilde{(y)_\infty^L}$ is a positive divisor, and thus $\tilde{G} - \widetilde{(y)_\infty^L} \leq \tilde{G}$. Let us show that (3) holds. Let $f \in \mathcal{F} = \mathcal{L}_K\left(\tilde{G} - \widetilde{(y)_\infty^L}\right)$. By definition, we have

$$(f)^K \geq -\left(\tilde{G} - \widetilde{(y)_\infty^L}\right),$$

and then

$$(\pi^* f)^L = \pi^*((f)^K) \geq -\pi^*\left(\tilde{G} - \widetilde{(y)_\infty^L}\right) \geq (y)_\infty^L - G \ ,$$

Now, we get

$$((\pi^* f) \cdot y)^L = (\pi^* f)^L + (y)^L \geq ((y)_\infty^L - G) + (y)^L = (y)_0^L - G \geq -G,$$

since $(y)_0^L$ is an effective divisor. In particular, we just proved that $(\pi^* f) \cdot y \in \mathcal{L}_L(G)$ for every $f \in \mathcal{F}$, that is (3) holds. □

**Remark 19.** From our assumptions, the space of functions $\mathcal{F}$ given in the above lemma is the best we can construct since we do not have any informations on the structure of $\mathcal{L}_L(G)$. Note however that from Definition 10 and the proof of Lemma 18, our choice is not far from being the best possible.

In our situation, the space $\mathcal{F}$ in Lemma 18 can be explicitely determined, since it is a subspace of $K$ which is supposed to be known (see hypothesis 2 and 4 above). Thus, suppose the divisor

$$D := \tilde{G} - \widetilde{(y)_\infty^L} \in \mathrm{Div}(K) \tag{4}$$

is known from now on. In particular, we can find a basis $f_1, ..., f_s \in K$ of its Riemann-Roch space, meaning that $\mathcal{F}$ is given by

$$\mathcal{F} := \mathcal{L}_K(D) = \langle f_1, ..., f_s \rangle_{\mathbb{F}_{q^m}},$$

where $s := \dim(D)$.
Now let us consider the row vectors, for every $1 \leq k \leq s$ :

$$\mathbf{u}_k := (\pi^* f_k(Q_{i,j}))_{i,j} \text{ , with } 1 \leq i \leq r \text{ , } 1 \leq j \leq \ell.$$

At this point, we are able to compute the $\mathbf{u}_k$'s in the following way :

1. We first compute the vectors $\mathbf{a}_k := (f_k(P_i))_i$ , $1 \leq i \leq r$. This can be easily done since both the $f_k$'s and $P_i$'s are known at this point;
2. Next we use hypothesis 3 to recover the $\mathbf{u}_k$'s: by definition of pull-backs, for any fixed $1 \leq i \leq r$, we have
$$f_k(P_i) = \pi^* f_k(Q_{i,j}) \text{ , } 1 \leq j \leq \ell.$$
Since we know the indices (in the ordered set $\mathcal{Q}$) corresponding to the extensions in $\mathbb{P}_L$ of any place $P \in \mathcal{P}$, we can re-build the $\mathbf{u}_k$'s by duplicating the value of $f_k(P_i)$ in the corresponding coordinates.

Now, using equations (2) and (3) above, one gets

$$\mathbf{u}_k \star \mathbf{y} \in \mathcal{C}, \text{ for every } 1 \leq k \leq s,$$

where $\star$ is the componentwise product of row vectors. If we denote by $\mathbf{D}_k = \mathrm{Diag}(\mathbf{u}_k)$, the right hand-side of (2) leads to the linear system

$$\begin{pmatrix} \mathbf{H} \cdot \mathbf{D}_1 \\ \vdots \\ \mathbf{H} \cdot \mathbf{D}_s \end{pmatrix} \cdot \mathbf{y}^T = 0, \tag{5}$$

from which $\mathbf{y}$ is a particular solution. Then if we have enough equations, we can hope to recover $\mathbf{y}$ by solving (5).

Let us denote by

$$\mathbf{A} := \begin{pmatrix} \mathbf{H} \cdot \mathbf{D}_1 \\ \vdots \\ \mathbf{H} \cdot \mathbf{D}_s \end{pmatrix}$$

the above matrix. By the above discussion, $\mathbf{y}$ is in the kernel of $\mathbf{A}$, but since it's the only vector we are searching for, we have to study a bit more the vector space $\mathrm{Ker}(\mathbf{A})$. In order this space to have dimension one (meaning that $\mathbf{y}$ is the unique solution up to scalar multiplication), we would like to have as many equations as possible. Let us now study how the parameters impact the structure of $\mathrm{Ker}(\mathbf{A})$.

If $S$ denotes the number of equations in the linear system (5), one have

$$S = \#\,\mathrm{Rows}(\mathbf{H}) \times s,$$

where $s := \ell(D)$. By definition, the number of rows of $H$ equals $n - \dim_{\mathbb{F}_q}(\mathcal{C}) = \ell r - \dim_{\mathbb{F}_q}(\mathcal{C})$. Using the Riemann-Roch theorem twice gives

$$s = \ell(D) \leq \deg\left(\tilde{G} - \widetilde{(y)_\infty^L}\right) + 1 - g(K),$$

and

$$\dim_{\mathbb{F}_{q^m}}(\mathcal{C}) \leq \deg(G) + 1 - g(L).$$

From the definition, we have $\deg(\tilde{G}) \leq \lfloor \frac{\deg(G)}{\ell} \rfloor$, so it is clear from the above estimation that the number $S$ of equations depends on both genera of $L$ and $K$, as well as the degree of the divisor $G$ (if the length of the code $n = \ell r$ is fixed). In particular, if $g(K)$ and $g(L)$ are too high, we will probably have less equations (keep in mind that $g(L) \geq g(K)$ and that $g(L)$ can be computed from $g(K)$ (see. Hurwitz' formula, [Sti09], theorem 3.4.13)). Nevertheless, in all our computing experiments, we noticed that theses parameters (ie. $\deg(G)$ and both genera in particular) do not impact the structure of $\mathrm{Ker}(A)$ and thus the solutions of (5). The upcoming Proposition tells us that it is not really surprising: In fact, it is possible to describe other solutions independently of theses parameters.

**Proposition 20.** *Let $h \in L$ be a function such that*

$$(h)_\infty^L \leq (y)_\infty^L$$

*holds. Then the evaluation vector $\mathbf{h} := (h_{i,j})_{i,j}$ ,where $h_{i,j} := h(Q_{i,j})$, for every $1 \leq i \leq r$ and $1 \leq j \leq \ell$ is also in the kernel of $A$.*

*Proof.* Using notations of Lemma 18, take $h \in L$ as above, and a function $g \in \mathcal{F}$. By definition of $\mathcal{F}$, we have

$$(\pi^* g)^L \geq -\pi^* \left( \tilde{G} - \widetilde{(y)_\infty^L} \right) \geq -G + (y)_\infty^L.$$

Thus we have

$$((\pi^* g) \cdot h)^L = (\pi^* g)^L + (h)^L \geq -G + (y)_\infty^L + (h)^L = -G + (h)_0^L + \underbrace{((y)_\infty^L - (h)_\infty^L)}_{\geq 0} \geq -G,$$

that is $(\pi^* \mathcal{F}) \cdot h \subseteq \mathcal{L}_L(G)$, and thus the evaluation vector $\mathbf{h}$ also leads to a solution.

□

The above proposition proves that the space of solutions doesn't depend on the number of equations. We will see in some examples later that we can explicitely decide whenever a function gives a solution or not, depending on the divisor $(y)_\infty^L$. This leads to a problem: how to find $\mathbf{y}$ among all possible solutions? We will see in practical examples later that depending on the cover, and especially on the action of the automorphism group $\Sigma$, we can add to the system (5) others equations, that are only satisfied by $\mathbf{y}$, allowing us to separate it from other solutions.

# 5 Applications

## 5.1 About the quotient curve

As we saw in Section 4, our method allows us to recover the defining equation of a curve $\mathcal{Y}$, provided that we are given enough information about one of its quotient curves $\mathcal{X}$. We can also see the situation as follows: Given a plane curve $\mathcal{X}$, we can recover the defining equation of one of its covers $\mathcal{Y}$. A natural question is then for which type of curve $\mathcal{X}$ this is possible.

[À FAIRE LIRE/RELIRE]
A first idea could be to take $\mathcal{X}$ equal to the projective line $\mathbb{P}^1$ over $\mathbb{F}_{q^m}$. In this situation, the AG-code we are looking for is a Generalized Reed-Solomon code (GRS in short). As the dual of a GRS code is also a GRS-code, the corresponding invariant SSAG-code turns out to be the subfield subcode of a GRS-code, that is an alternant code. Note that is easy to pass from the description of an SSAG-code over $\mathbb{P}^1$ to a description of an alternant code (see [Sti09], Chapter 2). In [FOPT10], the author proposed a new method to study the key security of such scheme. In particular, they show that the secret elements of the publique alterant code satisfy a system of polynomial equations. As a result, it is possible to attack such schemes, even if the cost of solving the correspond system using Gröbner

basis may be hard to estimate. Nevertheless, it seems reasonable to consider a more general classe of curves for the choice of $\mathcal{X}$, which will be the discussed below.

From hypothesis 4 (see Section 4), we would like to control the divisor $\widetilde{(y)}_\infty^L$, where $y$ is a primitive element of $L/K$. It is clear that if $K = \mathbb{P}^1$, this divisor is only support by the extension of the point at infinity $\infty$ in the projective line. The following definition will provide the class of curve where $\mathcal{X}$ will be chosen.

**Definition 21.** Let $\mathcal{X}$ be a curve over $\mathbb{F}_{q^m}$ and $K$ its function field. Let us assume that $K = \mathbb{F}_{q^m}(x, \alpha)$ is an separable and algebraic extension of the rational function field $\mathbb{F}_{q^m}(x)$. We say that $\mathcal{X}$ has separated variables if the primitive element $\alpha$ satisfies

$$F_1(\alpha) = F_2(x),$$

where $F_1, F_2 \in \mathbb{F}_{q^m}[T]$ and $[K : \mathbb{F}_{q^m}(x)] = \deg(F_1)$.

The following lemma explains why these curves are interesting in our case :

**Lemma 22.** *Let $\mathcal{X}$ be a curve with separated variables, whose function field $K = \mathbb{F}_{q^m}(x, \alpha)$ is given by the equation*

$$F_1(\alpha) = F_2(x),$$

*where $F_1, F_2 \in \mathbb{F}_{q^m}[T]$ are two univariate polynomials with coprime degrees, and denote by $\pi : \mathcal{X} \longrightarrow \mathbb{P}^1$ the corresponding morphism of curves. Let $\infty$ be the pole of $x$ in $\mathbb{F}_{q^m}(x)$. Then $\infty$ is totally ramified in $K/\mathbb{F}_{q^m}(x)$, and its unique extension $P_\infty \in K$ is the unique pole of $\alpha \in \mathbb{P}_K$. Moreover, we have*

$$(\alpha)_\infty^K = \deg(F_2) \cdot P_\infty.$$

*Proof.* Let $P_\infty$ be some extension of $\infty$ in $K$. Obviously, we have

$$e(P_\infty|\infty) \leq \deg(F_1) = [K : \mathbb{F}_{q^m}(x)].$$

On the other side, from the defining equation of $K$; we get

$$F_1(\alpha) = F_2(x) \Rightarrow \deg(F_1) \cdot \nu_{P_\infty}(\alpha) = e(P_\infty|\infty) \cdot \deg(F_2) \cdot \underbrace{\nu_{R_\infty}(x)}_{=-1}.$$

Since $\gcd(\deg(F_1), \deg(F_2)) = 1$, we get $\deg(F_1) \mid e(P_\infty|\infty)$, that is $\infty$ is totally ramified in $K/\mathbb{F}_{q^m}(x)$ and $e(P_\infty|\infty) = \deg(F_1)$. Moreover, we have

$$\begin{aligned} \deg(F_1) \cdot (\alpha)_\infty^K &= \deg(F_2) \cdot \pi^*(x)_\infty^{\mathbb{F}_q(x)} \\ &= \deg(F_2) \cdot \pi^* \infty \\ &= \deg(F_2) \cdot e(P_\infty|\infty) \cdot P_\infty, \end{aligned}$$

which gives the result. □

The main point in this type of curves is that we keep track of the place at infinity (namely $\infty$) in the corresponding extension of function fields. That will later allow us to look at the ramification of $\infty$ in the tower $\mathcal{Y} \longrightarrow \mathcal{X} \longrightarrow \mathbb{P}^1$, giving us a good way to describe the divisor $\widetilde{(y)}_\infty^L \in \mathrm{Div}(K)$.

## 5.2 Kummer covering

Let $\mathcal{X}$ be a curve over $\mathbb{F}_{q^m}$ with separated variables (see. Definition 21), whose function field is given by $K = \mathbb{F}_{q^m}(x, \alpha)$, with

$$F_1(\alpha) = F_2(x),$$

$F_1, F_2 \in \mathbb{F}_{q^m}[T]$ and $\gcd(\deg(F_1), \deg(F_2)) = 1$.

Let $\ell \mid q^m - 1$ be an integer (not necessarily a prime). Consider the extension $L = K(y)$, with

$$y^\ell = f(x, \alpha) \ , \ f \in \mathbb{F}_{q^m}[X, T] \tag{6}$$

and denote by $d := \deg\left((f)_\infty^L\right)$ the pole degree of $f$. Suppose also that $\gcd(d, \ell) = 1$. Then $L/K$ is a Kummer extension (see [Sti09] Annex A.13), cyclic of order $\ell = [L : K]$ and

$$\mathrm{Gal}(L/K) = \{\sigma : y \mapsto \xi \cdot y \mid \xi \in \mu_\ell^*(\mathbb{F}_{q^m})\}.$$

**Remark 23.** This special case of cyclic extensions have been extensively studied and are well-known. With Artin-Schreier extensions, they caracterize in a sense all cyclic extensions (see [Sti09] Annex A.13). In particular, the element $f$ as in (6) should be an element in $K$, that is potentially a rational function. Above, we supposed that $f$ is a polynomial, since we can always reduce to this case using a change of variables.

Let us explain the hypotheses before describing our attack in this context (this is a special case of the one given in Section 4). Denote by $\pi\mathcal{Y} \longrightarrow \mathcal{X}$ a morphism of algebraic curves that corresponds to the extension of function fields $L/K$. Suppose we are given an SSAG code $\mathcal{C}$ on the curve $\mathcal{Y}$, that is invariant under the action of the whole Galois group $\mathrm{Gal}(L/K)$. Since this group is well-known, the corresponding action is completely determined by the choice of an $\ell^{th}$-root of unity $\xi \in \mu_\ell^*(\mathbb{F}_{q^m})$. Our hypotheses are then the following :

1. We know a parity check matrix $\mathbf{H}$ of the code $\mathcal{C} = \mathrm{SSAG}_q(\mathcal{Y}, \mathcal{Q}, G)$;
2. The quotient curve $\mathcal{X}$ is known (that is polynomials $F_1$ and $F_2$), as well as the structure of the invariant code of $\mathcal{C}$, ie. $\mathcal{P}$ and $\tilde{G}$ such that $\mathcal{C}^\sigma = \mathrm{SSAG}_q(\mathcal{X}, \mathcal{P}, \tilde{G})$;
3. The automorphism $\sigma \in \mathrm{Gal}(L/K)$ acting on $\mathcal{C}$ is unknown. In particular, we don't know the corresponding root of unity $\xi$.

According to Section 4, we need to control the divisor $\widetilde{(y)_\infty^L}$. Let us start with the following lemma.

**Lemma 24.** *Keep notations as in Lemma 22. Then in the above situation, the place $P_\infty$ (the unique pole of $\alpha$) is fully ramified in $L/K$, and its unique extension $Q_\infty \in L$ is the unique pole of $y$ in $L$.*

*Proof.* See [Sti09], Proposition 3.7.3. □

**Proposition 25.** *We have*

$$(x)_\infty^L = \ell \cdot \deg(F_1) \cdot Q_\infty,$$
$$(\alpha)_\infty^L = \ell \cdot \deg(F_2) \cdot Q_\infty,$$

*and*

$$(y)_\infty^L = d \cdot Q_\infty.$$

*Proof.* Let $R_\infty$ be the simple pole of $x$ in $\mathbb{F}_{q^m}(x)$. It is totally ramified in $K/\mathbb{F}_{q^m}(x)$ (see Lemma 24), so $(x)^K = \deg(F_1) \cdot P_\infty$. We also know the divisor of poles of $\alpha$ in $K$, so using Lemma 2 yields

$$(x)_\infty^L = \pi^*(x)_\infty^K = \deg(F_1) \cdot \pi^* P_\infty = \ell \cdot \deg(F_1) \cdot Q_\infty,$$

and

$$(\alpha)_\infty^L = \deg(F_2) \cdot \pi^* Q_\infty = \ell \cdot \deg(F_2) \cdot Q_\infty.$$

Next, by hypothesis we have $(f)_\infty^K = d \cdot P_\infty$ (recall that $P_\infty$ is the unique pole of $x$ and $\alpha$ and $K$), so the equation $y^n = f$ gives

$$\begin{aligned} \ell \cdot (y)_\infty^L &= \pi^*(f)_\infty^K \\ &= d \cdot e(Q_\infty | P_\infty) \cdot Q_\infty \ , \end{aligned}$$

that is $(y)_\infty^L = d \cdot Q_\infty$. □

**Remark 26.** Considering these extensions, the divisor of poles of $y$ we are interested in is particularly simple because it is only supported by one place, which is the unique extension of the point at infinity $\infty$ in the tower $\mathbb{F}_{q^m}(x) \subseteq K \subseteq L$.

Proposition 25 above allows us to give the precise structure of the divisor $D$ (recall its definition in (4), Section 4).

**Corollary 27.** *We have*

$$D = \tilde{G} - \left\lceil \frac{d}{\ell} \right\rceil \cdot P_\infty \in \mathrm{Div}(K).$$

*Proof.* From the structure of $(y)_\infty^L$ given in Proposition 25, it is clear that

$$\mathrm{Supp}\left(\widetilde{(y)_\infty^L}\right) \subseteq \{P_\infty\},$$

since $P_\infty$ is fully ramified. It remains to show that if $D$ is defined as above, then $D = \tilde{G} - \widetilde{(y)_\infty^L}$. In fact, we have

$$\begin{aligned}
\pi^* D &= \pi^* \left( \tilde{G} - \left\lceil \frac{d}{\ell} \right\rceil \cdot P_\infty \right) \\
&\leq G - \ell \cdot \left\lceil \frac{d}{\ell} \right\rceil \cdot Q_\infty \quad \text{(since } \pi^* P_\infty = \ell \cdot Q_\infty, \text{see Remark 11)} \\
&\leq G - d \cdot Q_\infty \\
&= G - (y)_\infty^L,
\end{aligned}$$

the last equality coming from Proposition 25. □

Note that the divisor $D$ in the above corollary is known in this context from our hypotheses, and thus one can construct the corresponding linear system (see (5)).

As we already mentionned earlier, the linear system (5) doesn't only have the vector $\mathbf{y}$ as solution, but also any evaluation vector that comes from a function $h \in L$ such that

$$(h)_\infty^L \leq (y)_\infty^L = d \cdot Q_\infty.$$

In the context of a Kummer covering, one can easily find other solutions. In fact, let $h := x^i \alpha^j \in K$ be a function that only depend on variables $x$ and $\alpha$, and $\mathbf{h} = \mathbf{x}^i \boldsymbol{\alpha}^j$ its corresponding evaluation vector, with usual notations. Using Proposition 25, and in particular the description of the pole divisors of $x$ and $\alpha$, we see that

$$(h)_\infty^L = \ell \cdot (i \cdot \deg(F_1) + j \cdot \deg(F_2)) \cdot Q_\infty.$$

As a result, $\mathbf{h}$ is also a solution of the system (5), provided that

$$\ell \cdot (i \cdot \deg(F_1) + j \cdot \deg(F_2)) \leq d.$$

Since we have found other solutions, we need to choose the vector $\mathbf{y}$ among them. This can be done by adding other equations to the system, that are only satisfied by the vector $\mathbf{y}$. Indeed, since the action of the automorphism group $\Sigma = \langle \sigma \rangle$ that acts on the support $\mathcal{Q}$ of the code $\mathcal{C}$ is given by

$$\sigma : y \longrightarrow \xi \cdot y,$$

with $\xi \in \mu_\ell^*(\mathbb{F}_{q^m})$, the components of $\mathbf{y}$ satisfy a geometric progression by orbit (recall that $\mathcal{Q}$ is made of distincts orbit under the action of $\sigma$).

Recall that the set $\mathcal{Q}$ is made of $r$ orbits of length $\ell$, and to simplify the notations, suppose its elements are ordered orbit by orbit. To be concrete, if $\mathcal{P} = \{P_1, ..., P_r\} \subseteq \mathbb{P}_K$, then elements of $\mathcal{Q}$ at indices $(i-1)\ell + 1, ..., i\ell$ correspond to the $\ell$ extensions of $P_i$ in $L$ (for every $1 \leq i \leq r$). Let us now consider the following block matrices

$$\mathbf{A}(\xi) := \begin{pmatrix} \mathbf{B}(\xi) & 0 & \cdots & 0 \\ 0 & \mathbf{B}(\xi) & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \mathbf{B}(\xi) \end{pmatrix}, \text{ where } \mathbf{B}(\xi) = \begin{pmatrix} \xi & -1 & 0 & \cdots & 0 \\ 0 & \xi & -1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & -1 \\ -1 & 0 & \cdots & \cdots & \xi \end{pmatrix}$$

and $\xi$ is the root of unity that defines $\sigma$, and $\mathbf{A}(\xi) \in M_n(\mathbb{F}_{q^m})$. Then we have

$$\mathbf{A}(\xi) \cdot \mathbf{y}^T = 0.$$

In particular, one gets

$$\begin{pmatrix} \mathbf{A}(\xi) \\ \mathbf{H} \cdot \mathbf{D}_1 \\ \vdots \\ \mathbf{H} \cdot \mathbf{D}_s \end{pmatrix} \cdot \mathbf{y}^T = 0. \tag{7}$$

The system (7) is enough to recover $\mathbf{y}$ since the other solutions of (5) do not satisfy this geometric progression structure, because it is clear by construction that the evaluation vectors $\mathbf{x}$ and $\boldsymbol{\alpha}$ are equal on each orbit of length $\ell$, since $\sigma$ only acts on $y$-coordinate of points on the curve $\mathcal{Y}$.

**Remark 28.** Since, $\sigma$ (and so $\xi$) is supposed to be unknown at the beginning of the attack, one may have to test all the possibilities for $\xi$ in order to find the correct one. This leads to solve at most $\#\mu_\ell^*(\mathbb{F}_q) = \varphi(\ell)$ linear systems like (7), which remains feasable since $\varphi(\ell)$ is rather small.

In all our computing experiences, system (7) allows us to recover the desired vector $\mathbf{y}$. To finish the attack, we only have to recover the polynomial $f$ that defines the extension $L/K$ by using a bivariate polynomial interpolation method. A detailed algorithm for this attack can be found in Annex A, as well as a complexity analysis.

### 5.3 Artin-Schreier covering

As in section 5.2, the quotient curve $\mathcal{X}$ is taken as a curve over $\mathbb{F}_{q^m}$ with separated variables, those function field $K = \mathbb{F}_q(x, \alpha)$ is given by

$$F_1(\alpha) = F_2(x),$$

$F_1, F_2 \in \mathbb{F}_{q^m}[T]$ and $\gcd(\deg(F_1), \deg(F_2)) = 1$.
Here, we will consider an Artin-Schreier cover of the curve $\mathcal{X}$. Let $p := \mathrm{char}(\mathbb{F}_{q^m})$ denote the characteristic of the base field $\mathbb{F}_{q^m}$. Consider the extension $L = K(y)$ such that

$$y^p - y = f(x, \alpha),$$

with $f \in \mathbb{F}_{q^m}[X, T]$ and denote by $d := \deg\left((f)_\infty^L\right)$ the pole degree of $f$.Suppose also that $\gcd(d, p) = 1$. Then the extension $L/K$ is an Artin-Schreier extension, cyclic of order $p$ and

$$\mathrm{Gal}(L/K) = \{\sigma : y \mapsto y + \beta \ , \ \beta \in \{0, ..., p-1\}\}.$$

Note that once again, we took $f$ as a polynomial instead of a potential rational function for the same reasons as in section 5.2 (see. Remark 23).

In this case, the hypotheses of our procedure are the same as in section 5.2, knowing that this time the automorphism is completely determined by the choice of an element $\beta \in \mathbb{F}_p$. Here again, our goal will be to recover the minimal polynomial of $y$ over $K$, that is $f \in \mathbb{F}_{q^m}[X, T]$. Using the defining equation of the function field $L$ and the fact that $d$ is prime to $p$, one can show that the place $Q_\infty \in \mathbb{P}_K$ (same notations as in Lemma 24) is totally ramified in $L/K$. As usual, we denote by $P_\infty$ its unique extension in $L$. With our choice of parameters and hypotheses, we can prove that

**Proposition 29.** *We have*

$$(x)_\infty^L = p \cdot \deg(F_1) \cdot Q_\infty,$$
$$(\alpha)_\infty^L = p \cdot \deg(F_2) \cdot Q_\infty,$$

*and*

$$(y)_\infty^L = d \cdot Q_\infty.$$

*Proof.* Similar to the proof of Proposition 25 above. □

Note that this is almost the same result as in the Kummer case. In particular, the divisor of poles of $y$ in $L$ is only supported by the place $P_\infty$. As a result, the divisor in $K$ that we will use to construct our linear system is here given by

$$D = \tilde{G} - \left\lceil \frac{d}{p} \right\rceil \cdot P_\infty \in \mathrm{Div}(K).$$

This allows us to construct the linear system (5), since $D$ can be constructed from our hypotheses.

In the Artin-Schreier case, one can proceed the same way to find other solutions of (5). In particular, a monomial $x^i\alpha^j \in K$ gives a solution vector if and only if

$$p \cdot (i \cdot \deg(F_1) + j \cdot \deg(F_2)) \leq d.$$

Note that this is pretty much the same condition as in the Kummer case. Thus, we need a way to select the correct solution. For that, we add again other equations that are only satisfied by the vector $\mathbf{y}$, recalling that here, the action of the automorphism group $\langle\sigma\rangle$ on the set $\mathcal{Q}$ is given by

$$\sigma : y \longmapsto y + \beta,$$

where $\beta \in \mathbb{F}_p$. Thus the vector $\mathbf{y}$ we are searching for satisfies an arithmetic progression by orbit. In order to see it fluently, let us assume again that the support $\mathcal{Q}$ is ordered by orbit. Then let us consider the following block matrices :

$$\mathbf{C} := \begin{pmatrix} \mathbf{B} & 0 & \cdots & 0 \\ 0 & \mathbf{B} & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \mathbf{B} \end{pmatrix}, \text{where } \mathbf{B} = \begin{pmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & 1 \\ 1 & 0 & \cdots & \cdots & -1 \end{pmatrix}.$$

Then we have

$$\mathbf{C} \cdot \mathbf{y}^T = \begin{pmatrix} \beta \\ \vdots \\ \beta \end{pmatrix},$$

where $\beta$ is the element in $\mathbb{F}_p$ that defines the automorphism $\sigma$ (note that $\beta$ is supposed to be unknown here, but as in Kummer case, we can search for it in reasonable time) . Thus, if we add this to (5) we get

$$\begin{pmatrix} \mathbf{C} \\ \mathbf{H} \cdot \mathbf{D}_1 \\ \vdots \\ \mathbf{H} \cdot \mathbf{D}_s \end{pmatrix} \cdot \mathbf{y}^T = \begin{pmatrix} \beta \\ \vdots \\ \beta \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{8}$$

The new system (8) allows us to isolate $\mathbf{y}$, since the other solutions do not satisfy this arithmetic progression, for the same reason as in the Kummer case. Thus one can finish the attack by retrieving the polynomial $f$ using an interpolation method.

### 5.4 Extension to solvable Galois covering

In section 5.2 and 5.3, we applied our attack in the special case of Kummer and Artin-Schreier covers. Thoses cases are especially interesting since they are the elementary blocks of cyclic covers (see for example [Sti09], Annex A.13 for a characterization of cyclic extensions of function fields). Our aim in this section is to generalize the attack in the case of a solvable Galois covering.

**Remark 30.** We warn the reader about the characterization of cyclic extensions. Let $L/K$ be on order $\ell$ cyclic extension of function fields over $\mathbb{F}_{q^m}$. If $\gcd(\ell, \mathrm{char}(\mathbb{F}_{q^m})) = 1$, then $L/K$ is a Kummer extension if and only if the primitve $\ell$-th roots of unity are in $\mathbb{F}_{q^m}$, that is given by the condition $\ell \mid q^m - 1$. As a consequence, we need to keep

Throughout all this section, we keep the same notations as in section 4, and we add the assumption that the Galois group $\mathcal{G} := \mathrm{Gal}(L/K)$ is solvable, that is there exist a sequence of normal subgroups

$$\{Id\} := \mathcal{G}_0 \triangleright \mathcal{G}_1 \triangleright \cdots \triangleright \mathcal{G}_s := \mathcal{G}, \tag{9}$$

such that any quotient $\mathcal{G}_{i+1}/\mathcal{G}_i$ in (9) is cyclic of degree $n_i \in \mathbb{N}$. If $n_i = p^{r_i} m_i$, with $(m_i, \mathrm{char}(\mathbb{F}_{q^m})) = 1$; we will later need that $m_i \mid q^m - 1$ for all $0 \leq i \leq s-1$ (this assumption is required in order to have $\mu^*_{m_i}(\overline{\mathbb{F}_{q^m}}) \subseteq \mathbb{F}_{q^m}$).
Let us denote $L_i := L^{\mathcal{G}_i}$ the fixed field by $\mathcal{G}_i$, for $0 \leq i \leq s$. Note that $L_0 = L$ and $L_s = K$. From Galois theory, the sequence (9) leads to a tower of function fields

$$K := L_s \subseteq L_{s-1} \subseteq \cdots \subseteq L_0 := L, \tag{10}$$

such that for every $0 \leq i \leq s-1$, the extension $L_i/L_{i+1}$ is cyclic, with Galois group equals $\mathcal{G}_{i+1}/\mathcal{G}_i$. In particular, the tower (10) corresponds to a sequence of cyclic covers of $\mathbb{F}_{q^m}$-curves.

$$\mathcal{Y} := \mathcal{X}_0 \longrightarrow \mathcal{X}_1 \longrightarrow \cdots \longrightarrow \mathcal{X}_s := \mathcal{X}. \tag{11}$$

Note that for every $0 \leq i \leq s-1$, the curve $\mathcal{X}_i$ is equipped with the action of the group $\mathcal{G}_{i+1}/\mathcal{G}_i$, and $\mathcal{X}_{i+1}$ is the corresponding quotient curve.

Now let us assume that we want to attack the secret structure of a public SSAG-code on the curve $\mathcal{Y} = \mathcal{X}_0$, that is the code

$$\mathcal{C}_0 := \mathrm{SSAG}_q(\mathcal{X}_0, \mathcal{Q}_0, G_0),$$

for some support $\mathcal{Q}_0$ and divisor $G_0$ on $\mathcal{X}_0$. In the context of Section 4, let us suppose that we know the secret structure of the invariant code

$$\mathcal{C}_s := C_0^{\mathcal{G}} = \mathrm{SSAG}_q(\mathcal{X}_s, \mathcal{Q}_s, G_s)$$

on the smallest curve $\mathcal{X}_s = \mathcal{X}$. Then it is possible to attack the secret structure of $\mathcal{C}_0$ from the knowledge of $\mathcal{C}_s$, by applying the attack of Section 4 multiple times, each step allowing us to recover the secret structure of a subcode $\mathcal{C}_i = \mathrm{SSAG}_q(\mathcal{X}_i, \mathcal{Q}_i, G_i)$ of $\mathcal{C}_0$ defined over one of the curve $\mathcal{X}_i$. In fact, depending on the order of the Galois group $\mathcal{G}_{i+1}/\mathcal{G}_i$, we are led to use section 5.2 or section 5.3. Let us ewplain how we plan to act: Recall that for every $0 \leq i \leq s-1$, the integer $n_i := \#(\mathcal{G}_{i+1}/\mathcal{G}_i)$ can be written $n_i = p^{r_i} m_i$, with $(m_i, \mathrm{char}(\mathbb{F}_{q^m})) = 1$. More importantly, we supposed that $m_i \mid q^m - 1$.

Now fix $0 \leq i \leq s-1$. If we know the structure of the code $\mathcal{C}_i = \mathrm{SSAG}_q(\mathcal{X}_i, \mathcal{Q}_i, G_i)$ as well as a generator matrix of $\mathcal{C}_0$ and the corresponding automorphism acting on it, we can apply successively $r_i$-times the Artin-Schreier case (section 5.3) and one time the Kummer case (section 5.2) with degree $m_i$ (possible since $m_i \mid q^m - 1$) in order to build an SSAG-code $\mathcal{C}_{i+1}$ defined over $\mathcal{X}_{i+1}$.

This way, we can ride up the sequence of curves (11) until we have rebuilt the public code $\mathcal{C}_0$. In fact, from the knowledge of $\mathcal{C}_s$ and the public key, we can reconstruct a sequence of codes $(\mathcal{C}_i)_i$ until we get the public one.

**Remark 31.** Notice that for every $0 \leq i \leq s-1$, $\mathcal{C}_i = \mathcal{C}_{i+1}^{\mathcal{G}_{i+1}/\mathcal{G}_i}$.

**Remark 32.** By hypotheses in Section 4, we know the action of $\mathcal{G}$ on the invariant support $\mathcal{Q}_s$ on $\mathcal{X}$. In particular, we are able to find the corresponding induced permutation at each step of the procedure,

*ie.* for every $0 \leq i \leq s-1$, we know the permutation acting on the code $\mathcal{C}_{i+1}$ leading to the code $\mathcal{C}_i$ on the quotient curve. This allows us to get a parity check matrix of any code in the sequence $(\mathcal{C}_i)_i$ from the public generator matrix, which is necessary to build the linear systems we have to solve at each step. As a result, this shows that $\mathcal{C}_0$ admits several invariant codes, and one can build a generator matrix of anyone of them from the public data.

Let us conclude this section by presenting a toy example.

**Example 33.** With previous notations, let us assume that $\mathcal{G} = \mathrm{Gal}(L/K)$ is solvable of order $\#\mathcal{G} = pm$, with $(p, m) = 1$. Let us suppose that we are given the public key corresponding to the code $\mathcal{C}_0 = \mathrm{SSAG}_q(\mathcal{Y}, \mathcal{Q}, G)$, as well as a full access to the invariant code $\mathcal{C}_2 := \mathrm{SSAG}_q(\mathcal{X}, \mathcal{P}, \tilde{G})$. We then proceed as follow :

1. We start by applying section 5.3 on $\mathcal{C}_2$ in order to build a subcode $\mathcal{C}_1 = \mathrm{SSAG}_q(\mathcal{X}', \mathcal{P}', G')$ of $\mathcal{C}_0$, obtaining during the process a model of a curve $\mathcal{X}'$ with function field $F$ such that both extentions $L/F$ and $F/K$ are cyclic, with respective order $m$ and $p$ (that is the first one is a Kummer extension, the second one is an Artin-Schreier one).
2. From the knowledge of the code $\mathcal{C}_1$, we can now apply section 5.2 in order to recover the secret structure of $\mathcal{C}_0$.

In the next section, we will propose a scheme based on quasi-cyclic SSAG code constructed from the Hermitian curve, and propose parameters to protect the corresponding invariant code from our attack.

# 6 A McEliece scheme using quasi-cyclic SSAG-codes over the Hermitian curve

## 6.1 The proposed scheme

Here again, we work with on the finite field $\mathbb{F}_{q^m}$. Since the Hermitian curve is defined over a field with square cardinality, we suppose that $q^m = p^{2s}$ is a square. To avoid misunderstanding with the field $\mathbb{F}_q$ , where the SSAG code is defined, we introduced another field $\mathbb{F}_{q_0}$ with $q_0 = p^s$ elements. The Hermitian function field will be defined on $\mathbb{F}_{q^m} = \mathbb{F}_{q_0^2}$ by $\mathcal{H} := \mathbb{F}_{q_0^2}(x, y)$ with

$$y^{q_0} + y = x^{q_0+1}.$$

The idea is to construct a McEliece scheme based on a SSAG-code defined over the Hermitian curve, stable under the action of some automorphism of $\mathcal{H}$. There are two motivations to use this curve : first, it is a maximal curve, which means it atteins the Hasse-Weil bound (+ donner le genre et le nombre de points) . It allows us to consider longs codes, and thus more flexibility. Moreover, the automorphism group of $\mathcal{H}$ is very large and has been well-studied (see for example [Sti09] or [GSX00]); which permits us to chose a good automorphism $\sigma$ acting on our code (it will be really important later for the security of the scheme, see section 6.4.2).

We use the following notations :

- let $\sigma \in \mathcal{H}$ be an automorphism of order $\ell \in \mathbb{N}^*$ (we will describe later how to chose it);
- let $n_0 \in \mathbb{N}^*$ and $\mathcal{P} := \bigsqcup_{i=1}^{n_0} \mathrm{Orb}_\sigma(P_i)$ be a support made of $n_0$ distincts orbits under the action of $\sigma$;
- let $s \in \mathbb{N}^*$ and $G = \sum_{i=1}^{s} t_i \sum_{Q \in \mathrm{Orb}_\sigma(Q_i)} Q$ be an invariant divisor, with $Q_i \in \mathbb{P}_\mathcal{H}$ and $t_i \in \mathbb{Z}$. We also suppose that $\mathrm{supp}(G) \cap \mathcal{P} = \emptyset$.

We now can describe the scheme :

**Key generation:** We consider the quasi-cyclic code

$$\mathcal{C}_{\text{pub}} := \text{SSAG}_q(\mathcal{H}, \mathcal{P}, G)$$

constructed on the Hermitian curve $\mathcal{H}$, with length $n = n_0 \cdot \ell \leq N(\mathcal{H})$ and dimension $k$. Let $t$ be the correction capability of the code and $\mathbf{G}_{\text{pub}} = (I_k|\mathbf{M})$ be a systematic generator matrix of $\mathcal{C}_{\text{pub}}$, where $\mathbf{M}$ is an $\ell$-blocks-circulant matrix, that is of the form

$$\mathbf{M} = \begin{pmatrix} \cdots & \cdots & \cdots \\ \cdots & \mathbf{M}_i & \cdots \\ \cdots & \cdots & \cdots \end{pmatrix} \text{ with } \mathbf{M}_i := \begin{pmatrix} a_0 & a_1 & \cdots & a_{\ell-1} \\ a_{\ell-1} & a_0 & \cdots & a_{\ell-2} \\ \vdots & \ddots & \ddots & \vdots \\ a_1 & \cdots & a_{\ell-1} & a_0 \end{pmatrix}$$

Thus, $\mathbf{G}_{\text{pub}}$ can entirely be described by the set of rows

$$\rho(\mathbf{G}_{\text{pub}}) := \{M_i \mid i \in \{1, \ell+1, 2\ell+1, ..., (n-k)-\ell+1\}\},$$

$M_i$ representing the $i$-th row of $\mathbf{M}$.

- **Public key:** the set of rows $\rho(\mathbf{G}_{\text{pub}})$ and the integer $t$.
- **Secret key:** the support $\mathcal{P}$ and the divisor $G$.

**Encryption:** A plain text $\mathbf{m} \in \mathbb{F}_q^k$ is encrypted by

$$\mathbf{y} = \mathbf{m}\mathbf{G}_{\text{pub}} + \mathbf{e} \ ,$$

where $\mathbf{e} \in \mathbb{F}_q^n$ is an error vector such that $\omega(\mathbf{e}) \leq t$ ($\omega$ being the Hamming weight).

**Decryption:** Using a general decoding algorithm for algebraic geometry codes (see for example [HP95]), we can find a codeword $\mathbf{c} = \mathbf{y} - \mathbf{e} \in \mathcal{C}_{\text{pub}}$. From $\mathbf{c}$ and the knowledge of $\mathbf{G}_{\text{pub}}$, we can recover the message $\mathbf{m}$.

Note that in this scheme, the automorphism itself and the curve $\mathcal{H}$ is considered as secret data, and only the order $\ell$ of $\sigma$ is known from the public key. In previous sections, we have shown that the knowledge of the full invariant code allows us to attack the system, recovering the secret elements. It means that in order to secure our scheme, we need to ensure that the invariant code can not be recovered easily.

In what follow, we will describe somes known attacks against the invariant SSAG-code, and then we will propose a set of parameters to block those attacks.

## 6.2 Invariant code on the projective line

In the proposed scheme, the public code is a QC-SSAG-code constructed on the Hermitian curve, ie.

$$\mathcal{C}_{\text{pub}} = \text{SSAG}_q(\mathcal{H}, \mathcal{P}, G);$$

that is invariant under some order $\ell$ automorphism $\sigma \in \text{Aut}(\mathcal{H})$. As shown by Corollary 16, the invariant subcode (say $\mathcal{C}_{\text{inv}}$) is an SSAG-code on the quotient curve. Moreover, this code can be constructed in polynomial time (see Remark 8) from the generator matrix of $\mathcal{C}_{\text{pub}}$ and the action of the induced permutation on it (that is the public key). This means that a generator matrix of $\mathcal{C}_{\text{inv}}$ must also be considered as public data. From now on, let us denote by $\mathbf{G}_{\text{inv}}$ a generator matrix of $\mathcal{C}_{\text{inv}}$.

In the particular case where the quotient curve $\mathcal{H}/\langle\sigma\rangle$ is the projective line $\mathbb{P}^1(\mathbb{F}_{q^m})$, it is possible to construct an algebraic system to recover the secret elements of $\mathcal{C}_{\text{inv}}$. Using the attack of Section 4., we can then recover the public code, breaking the system. The key ingredient that allows us to build such a system is the fact that $\mathbb{P}^1(\mathbb{F}_{q^m})$ has genus zero and thus has a trivial divisor class group.

From Corollary 16, we know that the invariant code is given by

$$C_{\text{inv}} := \text{SSAG}_q\left(\mathbb{P}^1, \tilde{\mathcal{P}}, \tilde{G}\right),$$

To construct an algebric system, we start from the inclusion

$$C_{\mathcal{L}}(\mathbb{P}^1, \tilde{\mathcal{P}}, \tilde{G})^{\perp} \subseteq \text{SSAG}_q(\mathbb{P}, \tilde{\mathcal{P}}, \tilde{G})^{\perp} \otimes \mathbb{F}_{q^m},$$

that is a direct consequence of definitions (in the right-hand side, we have extended the scalars to $\mathbb{F}_{q^m}$ since its the field definition of the AG-code). This means that for every codeword $c \in C_{\mathcal{L}}(\mathbb{P}, \tilde{\mathcal{P}}, \tilde{G})^{\perp}$, we have

$$c \cdot \mathbf{G}_{\text{inv}}^T = 0, \tag{12}$$

where $\mathbf{G}_{\text{inv}}$ is a generator matrix of the invariant code $\mathcal{C}_{\text{pub}}^{\sigma}$ (that can be computed in polynomial time from the public data). Moreover, we know from Proposition 6 that there exist a divisor $G' \in \text{Div}(\mathbb{P}^1)$ such that

$$C_{\mathcal{L}}(\mathbb{P}^1, \tilde{\mathcal{P}}, \tilde{G})^{\perp} = C_{\mathcal{L}}(\mathbb{P}^1, \tilde{\mathcal{P}}, G').$$

As a consequence, the knowledge of a basis of the Riemann-Roch space $\mathcal{L}_{\mathbb{P}^1}(G')$ allows us to write formally codewords in $C_{\mathcal{L}}(\mathbb{P}^1, \tilde{\mathcal{P}}, G')$, without knowing the set $\tilde{\mathcal{P}}$. The key part is that the projective line $\mathbb{P}^1$ has a trivial divisor class group. In particular, there exist a function $h$ in the rational function field $\mathbb{F}_{q^m}(x)$ such that

$$G' = (h)^{\mathbb{F}_{q^m}(x)} + \deg(G') \cdot P_{\infty},$$

where $P_{\infty}$ is the unique pole of $x$ in $\mathbb{F}_{q^m}(x)$. From the last equality, we get

$$\mathcal{L}_{\mathbb{P}^1}(G') = \langle h(x)x^i \mid 0 \leq i \leq \deg(G') - 1 \rangle_{\mathbb{F}_{q^m}},$$

which is a dimension $r = \deg(G') - 1$ vector space over $\mathbb{F}_{q^m}$.
Let us write the unknown support $\tilde{\mathcal{P}} = (\tilde{P}_i)_{i=1}^{n_0}$, with $\tilde{P}_i = (x_i : 1)$ (here we made the analogy between rational points on a curve and degree one place on the function field). The goal is to recover the $x_i's$. For that, we denote by $\mathbf{X} = (X_1, ..., X_{n_0})$ and $\mathbf{Z} = (Z_1, ..., Z_{n_0})$ two sets of formal variables, respectively corresponding to $x_i's$ and $h(x_i)'s$. From equation (12) above, we have the following system

$$\begin{pmatrix} Z_1 & \cdots & Z_{n_0} \\ Z_1X_1 & \cdots & Z_{n_0}X_{n_0} \\ \vdots & \ddots & \vdots \\ Z_1X_1^r & \cdots & Z_{n_0}X_{n_0}^r \end{pmatrix} \cdot \mathbf{G}_{\text{inv}}^T = 0.$$

The first row provides $k = \dim(\mathcal{C}_{\text{inv}})$ linear equations in the variables $\mathbf{Z}$, and since $k < n_0$, one can eliminate some variables in the set $\mathbf{Z}$. On the other hand, the 2-transitivity of the affine group on $\mathbb{F}_{q^m}$ allows us to fix arbitrarily 2 unknows, say $x_1$ and $x_2$. Therefore, the above system consist in $kr$ equations in $n_0 - 2$ variables $\mathbf{X}$ and $n_0 - k$ variables $\mathbf{Z}$.
If we are able to solve this system, we recover the function $h$ (and thus $\tilde{G}$) as well as the support $\tilde{\mathcal{P}}$; that is we have reconstructed the invariant code $\mathcal{C}_{\text{inv}}$. Since the security of the whole system rely on it, we have broke the scheme.

**Remark 34.** The cost of solving the above system is hard to estimate. It is possible to have an upper bound on it in the case where the system has a specific form; which can be useful to estimate the security of schemes using SSAG-codes over the line. Thoses results can be found in [FOP+16a], [FOP+16b] and [FOPT10].

Actually, if the quotient curve is $\mathbb{P}^1$, the security is the same as the scheme using quasi-cyclic classical Goppa codes, and thus there no advantages to use it. In particular, this means that the automorphism $\sigma$ should be chosen such that $\mathcal{H}^{\sigma}$ is not rational. In the latter case, the fixed field has a more complex divisor class group and the above attack doesn't work. In the following section, we will describe the cost of an exhaustive search on the invariant code, depending on different parameters of the fixed field.

## 6.3 Brute force on the invariant code

Let us recall that the security of the private key of the scheme proposed in Section 6.1 rely on the security of the invariant code. This section will be dedicated to the cost of an exhaustive search on it, allowing us to understand how to choose the automorphism $\sigma$ in order to maximize the complexity of the exhaustive search. Let us rewrite

$$C_{\text{inv}} := \text{SSAG}_q\left(\mathcal{H}/\langle\sigma\rangle, \tilde{\mathcal{P}}, \tilde{G}\right)$$

the invariant code. A brute force attack on it will consists in the three following steps :

1. Enumerating all the possible divisor classes of a given degree on the quotient curve $\mathcal{H}/\langle\sigma\rangle$;
2. Guess the good divisor $\tilde{G}$ in the class;
3. Then guess the support $\tilde{\mathcal{P}}$ of length $n_0 := n/\ell$.

Let us first discuss the third step, which is the easiest to formalize.

**Recovering the invariant support.**

Here we assume that the two first steps were done and that a divisor $\tilde{G}$ was found. To recover the invariant support there are two ways to proceed :

The first consists in an exhaustive search on all subset $S \subseteq \mathcal{H}/\langle\sigma\rangle\,(\mathbb{F}_{q^m})$ of length $n_0 = n/\ell$, then we get the good permutation using the SSA algorithm (see [Sen00] for more details).

The second way is to solve a linear system as in Section 6.2. In order to build it, we start be recalling that there exists (from Proposition 6) a divisor $G' \in \text{Div}(\mathcal{H}/\langle\sigma\rangle)$ such that

$$C_{\mathcal{L}}(\mathcal{H}/\langle\sigma\rangle, \tilde{\mathcal{P}}, \tilde{G})^{\perp} = C_{\mathcal{L}}(\mathcal{H}/\langle\sigma\rangle, \tilde{\mathcal{P}}, G').$$

Here we don't know the quotient curve from our hypothesis. Nevertheless, let us suppose that the attacker found a way to recover it. It then becomes possible to find a basis of the Riemann-Roch space $\mathcal{L}_K(G') \subseteq K := \mathbb{F}_{q^m}(\mathcal{H}/\langle\sigma\rangle)$, *ie.* we have $\mathcal{L}_K(G') = \langle f_1, ..., f_s\rangle_{\mathbb{F}_{q^m}}$. As in section 6.2, we get

$$\forall\ 1 \leq i \leq s\ ,\ (f_i(\tilde{P}_1), ..., f_i(\tilde{P}_{n_0})) \cdot \mathbf{G}_{\text{inv}}^T = 0,$$

where $\tilde{\mathcal{P}} = \{\tilde{P}_1, ..., \tilde{P}_s)$. In particular, les us introduce $2n_0$ formal variables $X_1, ..., X_{n_0}$ and $Y_1, ..., Y_{n_0}$ corresponding to the evaluation in $x$ and $y$ on the places of $\tilde{\mathcal{P}}$. Here, we supposed that the quotient curve is seen in a plane model. This leads to the following system

$$\begin{pmatrix} f_1(X_1, Y_1) & \cdots & f_1(X_{n_0}, Y_{n_0}) \\ f_2(X_1, Y_1) & \cdots & f_2(X_{n_0}, Y_{n_0}) \\ \vdots & \ddots & \vdots \\ f_s(X_1, Y_1) & \cdots & f_s(X_{n_0}, Y_{n_0}) \end{pmatrix} \cdot \mathbf{G}_{\text{inv}}^T = 0.$$

Note that this method is nothing but a generalization of the method describe in 6.2, but this system is actually harder to solve since we cannot estimate the form of the rational functions $f_i's$. On the other hand, even if the system is polynomial, the complexity of solving it using Gröbner bases depend on the form and degree of the polynomials ans is thus difficult to forecast.

**Enumeration of divisor classes.**

Here, we will first explain why it is not necessary to enumerate all divisors in the quotient curve $\mathcal{H}/\langle\sigma\rangle$ in order to find the correct one. In fact, if the support is fixed, two different divisors can produce the same code. This fact comes from the specific structure of SSAG-codes, inherited from AG ones. In order to precise this, let us introduce the notion of diagonal equivalent codes.

**Definition 35.** Let $\mathcal{C}_1$,$\mathcal{C}_2 \subseteq \mathbb{F}_{q^m}^n$ be two linear codes. We say they are diagonal equivalent, denoted $\mathcal{C}_1 \sim_{\text{diag}} \mathcal{C}_2$, if there exist $n$ non zero scalars in $\mathbb{F}_{q^m}$, say $\lambda_1, ..., \lambda_n$, such that

$$\mathcal{C}_1 = (\lambda_1, ..., \lambda_n) \star \mathcal{C}_2 := \{(\lambda_1 c_1, ..., \lambda_n c_n) \mid (c_1, ..., c_n) \in \mathcal{C}_2\}.$$

Note that the diagonal equivalence between two codes $\mathcal{C}_1$ and $\mathcal{C}_2$ can be check by solving yet another linear system. In fact, let $\mathbf{G}_{\mathcal{C}_1}$ and $\mathbf{H}_{\mathcal{C}_2}$ be respectively a generator matrix of $\mathcal{C}_1$ and a parity check matrix of $C_2$. Let also $W_1, ..., W_n$ be $n$ formal variables, and consider the following system

$$\mathbf{G}_{\mathcal{C}_1} \cdot \begin{pmatrix} W_1 & \cdots & 0 \\ & \ddots & \\ 0 & \cdots & W_n \end{pmatrix} \cdot \mathbf{H}_{\mathcal{C}_2}^T = 0. \tag{13}$$

By Definition 35, this system has at least one solution if and only if $\mathcal{C}_1 \sim_{\text{diag}} \mathcal{C}_2$, that is we have an easy way to check whever two codes are diagonal equivalent or not.
Let us now explain why this property leads to a smarter brute force search on the divisor in the case of SSAG-codes. First, we deal with AG codes which are easier to treat. The following result shows that the equivalence class of AG-codes depends only on the equivalence class of its divisor.

**Theorem 36** ([MP93], Corollary 4.15.)**.** *Let $\mathcal{X}$ be an algebraic curve of genus $g$ and $\mathcal{P}$ be a set of $n > 2g-2$ rationnal places on $\mathcal{X}$. If $G$ and $H$ are two divisors on $\mathcal{X}$ of same degree $r$ such that $2g-1 < r < n-1$, then we have*

$$C_{\mathcal{L}}(\mathcal{X}, \mathcal{P}, G) \sim_{\text{diag}} C_{\mathcal{L}}(\mathcal{X}, \mathcal{P}, H) \iff G \sim H.$$

In what follow, we denote by $\text{AG}_r(\mathcal{X}, \mathcal{P})$ the set of AG-code on the curve $\mathcal{X}$ over $\mathbb{F}_{q^m}$, defined by a fixed support $\mathcal{P}$ and any divisor of degree $r$. Then we have

**Corollary 37.** *Let $\mathcal{P} \subseteq \mathcal{X}(\mathbb{F}_{q^m})$ be a support of lenght $n > 2g+2$, where $g$ is the genus of $\mathcal{X}$. Let also $r \in \mathbb{N}$ be such that $2g-1 < r < n-1$. Then*

$$\#\left(\text{AG}_r(\mathcal{X}, \mathcal{P}) / \sim_{\text{diag}}\right) = h(\mathcal{X}),$$

*where $h(\mathcal{X})$ is the number of divisor classes.*

*Proof.* Immediate consequence of Theorem 36 and the definition of the class number. □

The above estimation is sufficient if we want to perform a brute force search on an AG-code defined over a curve $\mathcal{X}$. In fact, we could proceed as follows.

1. Perform a brute force search among divisor classes of degree $r$ (corresponding to the public AG-code), that is choose a representative divisor $G'$ in the class.
2. Guess the support $\mathcal{P}'$ (see above) and attempt to solve the system (13) to check whether the code $C_{\mathcal{L}}(\mathcal{X}, \mathcal{P}', G')$ is diagonal equivalent to the public code.
3. If (13) has a solution $(\lambda_1, ..., \lambda_n) \in (\mathbb{F}_{q^m}^*)^n$, then we have recovered the public code $(\lambda_1, ..., \lambda_n) \star C_{\mathcal{L}}(\mathcal{X}, \mathcal{P}', G')$.

By Theorem 36 and Corollary 37, we could have an estimation of the cost of a brute force search amoung divisor classes. Unfortunately, both theses results cannot be applied on subfield subcodes directly.

In the case of SSAG-codes, the situation is more complicated, but one can use the following result to have a first estimation on the number of SSAG-codes.

**Proposition 38** ([MP93], Corollary 7.4)**.** *With above notations, if $n > 2g+2$ and $2g-1 < r < n$, then*

$$\#\text{AG}_r(\mathcal{X}, \mathcal{P}) = (q^m - 1)^{n-1} h(\mathcal{X}).$$

Now let us denote by $\text{SSAG}_{q,r}(\mathcal{X}, \mathcal{P})$ the set of subfield subcodes on $\mathbb{F}_q$ of AG codes on the curve $\mathcal{X}$, defined by the support $\mathcal{P}$ and a divisor of degree $r$. Since we are taking subcodes, it is clear from the previous proposition that

$$\#\,\text{SSAG}_{q,r}(\mathcal{X}, \mathcal{P}) \leq \#AG_r(\mathcal{X}, \mathcal{P}) = (q^m - 1)^{n-1} h(\mathcal{X}).$$

Actually, we can decrease a little the previous bound with the following remark.

**Remark 39.** Let $\mathcal{C}_1$,$\mathcal{C}_2$ be two linear codes of lenght $n$ over $\mathbb{F}_{q^m}$, and suppose that there exist $\lambda_1, ..., \lambda_n \in (\mathbb{F}_q^*)^n$ such that $\mathcal{C}_1 = (\lambda_1, ..., \lambda_n) \star \mathcal{C}_2$. Then their subfield subcodes on $\mathbb{F}_q$ are equal, ie.

$$\mathcal{C}_1 \cap \mathbb{F}_q^n = \mathcal{C}_2 \cap \mathbb{F}_q^n.$$

This leads to the final upper bound

$$\#\,\mathrm{SSAG}_{q,r}(\mathcal{X}, \mathcal{P}) \leq \frac{(q^m - 1)^{n-1}}{(q-1)^{n-1}} h(\mathcal{X}). \tag{14}$$

In annex B, we will formaly describe a brute search algorithm on the invariant SSAG-code and study its complexity.
Note that from the above approximation, it is clear that the complexity of the exhaustive search is lower bounded by the number of divisor classes on the quotient curve, that is $h(\mathcal{H}/\langle\sigma\rangle)$. In particular, this will lead to choose optimally the automorphism $\sigma \in \mathrm{Aut}(\mathcal{H})$ in order this number to be high enough. In the next section, we will discuss the choice of parameters for our system.

## 6.4 Suggested parameters

### 6.4.1 Choice of the quasi-cyclicity order $\ell$

The main interest in using quasi-cyclic codes is that it allows us to reduce public key sizes. In particular, the larger $\ell$ is, the smaller the public key is. This choice also influence the security of the scheme, as the security reduces to the security of the invariant code. In particular, if $\ell$ is too large, the invariant code will be rather small and probably less secured. Recall that it is always possible to build the invariant code from a public generator matrix and the permutation $\sigma$. Here we would like to choose $\ell$ such that it is not possible to construct any other intermediary code. We have two ways to avoid this.

1. $\ell$ **shoud be prime**. If there exist a prime $s \mid \ell$, then the power permutation $\sigma^s$ also act on the public code. In particular, $\mathcal{C}_{\mathrm{pub}}$ is also $s$-quasi-cyclic, that is one can construct another invariant code, ie. $\mathcal{C}_{\mathrm{pub}}^{\sigma^s}$. Since this is possible for every divisor of $\ell$, we are able to construct several SSAG codes smaller than the public code (see Corollary 16 for the structure of the invariant codes). Actually, we don't know if it makes the attack easier, but it is clear that every intermediary code could provides informations about the public support and divisor. As we want to give the least information possible, $\ell$ will be chosen as a prime.

2. $\ell$ **should be such that** $q$ **is in** $\mu_{\ell-1}^*(\mathbb{F}_{q^m})$. If this is satysfied, the polynomial $1 + z + z^2 + ... + z^{\ell-1} \in \mathbb{F}_q[z]$ is irreducible. This is motivated by the fact that there exist another intermediary code that can be constructed from the knowledge of a generator matrix and the automorphism $\sigma$, that is the *folded code* (see [FOP+16a]). This code is the image of the public code $\mathcal{C}_{\mathrm{pub}}$ by the map $\mathrm{id} + \sigma + ... + \sigma^{\ell-1}$. Now, if the polynomial $1 + z + z^2 + ... + z^{\ell-1}$ is reducible over $\mathbb{F}_q$, then it is possible to construct another intermediary subcode of $\mathcal{C}_{\mathrm{pub}}$ by computing its image by the map $P(\sigma)$, where $P$ is a divisor of $1 + z + z^2 + ... + z^{\ell-1}$. As in 1., we get another subcode of the public one that has a special structure related to the secret support and divisor. Actually we don't know if the knowledge of several folded code (note that we can always construct the full one) simplify an attack, but it could anyway be helpful for an attacker.

**Remark 40.** From the definition, it is clear that the folded code is $\sigma$-invariant, that is it is a subcode of $\mathcal{C}_{\mathrm{pub}}^{\sigma}$. Now let $p$ be the characteristic of $\mathbb{F}_{q^m}$. If $p \nmid \ell$, then those two codes are equal (see [Bar17] , Lemma 3.2). In what follow, we will always have $p \nmid \ell$. Hence, if $\ell$ satisfies 1. and 2. above, then the only subcode of $\mathcal{C}_{\mathrm{pub}}$ that an attacker can construct is the invariant code. In particular, we can focus on the security of $\mathcal{C}_{\mathrm{pub}}^{\sigma}$ without having to worry about possible other subcodes.

### 6.4.2 Choice of the automorphism $\sigma$

The complexity of the brute force attack on $\mathcal{C}_{\text{inv}}$, described in 6.3, depends on the class number of the quotient curve $h(\mathcal{H}/\langle\sigma\rangle)$. In particular, from Theorem 1, this number can be estimated using the genus $g(\mathcal{H}/\langle\sigma\rangle)$. In this section, we details how to construct an automorphism $\sigma \in Au(\mathcal{H})$ of order $\ell$, with $\ell$ satisfying conditions 1. and 2. above. After, we will see the influence of this choice on the genus of the quotient curve. A complete study on the automorphism group of the Hermitian curve can be found in [GSX00]. In particular, they compute the class number as well as the genus of several quotient curve of the Hermitian curve, which will help us in choosing a good automorphism.

**Seeking for an order $\ell$ automorphism.**

Let us denote by $\mathcal{A} := \text{Aut}(\mathcal{H})$ the automorphism group of the Hermitian function field. It is isomorphic to the projective unitary group $\text{PGU}_3(\mathbb{F}_{q_0^2})$, and has order (see for example [Sti09])

$$\text{ord}(\mathcal{A}) = q_0^3(q_0^2-1)(q_0^3+1).$$

As we aim to find an automorphism $\sigma$ of order $\ell$ satysfying the conditions in 6.4.1, we introduice the subgroup

$$\mathcal{A}(P_\infty) := \{\sigma \in \mathcal{A} \mid \sigma(P_\infty) = P_\infty\} \subseteq \mathcal{A},$$

consisting in all automorphisms fixing the point at infinity $P_\infty$ in $\mathcal{H}(\mathbb{F}_{q_0^2})$. Let $F = \mathbb{F}_{q_0^2}(x,y) = \mathbb{F}_{q_0^2}(\mathcal{H})$ be the function field of $\mathcal{H}$. It is proven in [GSX00] that such an automorphism acts as follows :

$$\begin{cases} \sigma(x) = ax + b \\ \sigma(y) = a^{q_0+1}y + ab^{q_0}x + c, \end{cases} \tag{15}$$

with $a \in \mathbb{F}_{q_0^2}^*$, $b \in \mathbb{F}_{q_0^2}$ and $b^{q_0+1} = c^{q_0} + c$ (see (2.2) in [GSX00]). We have

$$\text{ord}(\mathcal{A}(P_\infty)) = q^3(q^2-1).$$

From (15), any automorphism $\sigma \in \mathcal{A}(P_\infty)$ can be identified to a triple $(a,b,c)$, with $a \in \mathbb{F}_{q_0^2}^*$, $b \in \mathbb{F}_{q_0^2}$ and $b^{q_0+1} = c^{q_0} + c$. For convenience, this automorphism will be denoted $\sigma = [a,b,c]$. The order of such a $\sigma$ depends only on the order of $a$ and the choice of $c$.

**Lemma 41.** *(see [GSX00], Lemma 4.1.) Let $\sigma = [a,b,c] \in \mathcal{A}(P_\infty)$, with $a \neq 1$. Then we have*

*(i) If $\text{ord}(a) \nmid q_0+1$, then $\text{ord}(\sigma) = \text{ord}(a)$;*

*(ii) If $\text{ord}(a) \mid q_0+1$ then*

$$\text{ord}(\sigma) = \begin{cases} \text{ord}(a), \text{ if } c = \dfrac{ab^{q_0+1}}{a-1} \\ p \cdot \text{ord}(a), \text{ otherwise,} \end{cases}$$

*where $p = \text{char}(\mathbb{F}_{q_0^2})$.*

Now, let $\ell$ be an integer satisfying conditions 1. and 2. in 6.4.1, which also devides $q_0^2-1$. We chose randomly an element $a \in \mathbb{F}_{q_0^2}^*$ of order $\ell$ and $b \in \mathbb{F}_{q_0^2}$. If $\ell \mid q_0+1$, we choose $c = \dfrac{ab^{q_0+1}}{a-1}$, else we chose any $c$ amoung the roots of $X^{q_0} + X - b^{q_0+1}$. From Lemma 41, we get an automorphism $\sigma = [a,b,c]$ of order $\ell$.

**The genus of $\mathcal{H}/\langle\sigma\rangle$.**

The authors in [GSX00] provide a formula to compute the genus of the curve $\mathcal{H}/\langle\sigma\rangle$ in our context.

**Proposition 42.** *Let $\sigma = [a,b,c] \in \mathcal{A}(P_\infty)$ be an automorphism of prime order $\ell > 2$. Then*

*(i) If $\ell \mid (q_0-1)$, then $g(\mathcal{H}/\langle\sigma\rangle) = \dfrac{(q_0-1)q_0}{2\ell}$.*

*(ii) If $\ell \mid (q_0+1)$ and $c = \dfrac{ab^{q_0+1}}{a-1}$, then $g(\mathcal{H}/\langle\sigma\rangle) = \dfrac{(q_0-1)(q_0-(\ell-1))}{2\ell}$.*

*Proof.* It is a particular case of [GSX00], Theorem 4.4 □

Notice that since we want the quotient curve to have positive genus, $\ell$ should be strictly less than $q_0 + 1$. Using this Proposition and Theorem 1, we can estimate the class number $h(\mathcal{H}/\langle\sigma\rangle)$.

**Corollary 43.** *Let $\sigma = [a, b, c] \in \mathcal{A}(P_\infty)$ be an automorphism of prime order $\ell > 2$. Then we have*

*(i) If $\ell \mid (q_0 - 1)$, then $h(\mathcal{H}/\langle\sigma\rangle) = \mathcal{O}\left(q_0^{\frac{q_0^2}{\ell}}\right)$.*

*(ii) If $\ell \mid (q_0 + 1)$ and $c = \dfrac{ab^{q_0+1}}{a-1}$, then $h(\mathcal{H}/\langle\sigma\rangle) = \mathcal{O}\left(q_0^{\frac{q_0(q_0-1)}{\ell}}\right)$.*

We will see in Annex B that the divisor class $h(\mathcal{H}/\langle\sigma\rangle)$ is an upper bound for the cost of the brute force algorithm. In our suggested parameters for the scheme (see section 6.4.4), this number will be large enough to reach a complexity bigger than $2^{128}$ operations over the base field.

### 6.4.3 Choice of the base field

In order to provide SSAG-codes over $\mathbb{F}_q$, defined on the Hermitian function field, we have to choose and extension $\mathbb{F}_{q^m}$ of $\mathbb{F}_q$ such that $q^m$ is a square. Let us discuss the choice of $q$ and $m$, and the related $q_0$ such that $q^m = q_0^2$.

- $m$ should not be too large since it has a negative influence on the dimension of the code. In fact, for a fixed lenght $n$ and divisor $G$, the dimension of the SSAG-code is lower bounded by $n - m(n - \dim(G))$. As a result, if $m$ is too big, the rate (ie. $k/n$) of the SSAG might be too low.
- The same remark holds for the choice of $q_0$. In fact, recall that the genus of the Hermitian function fields is $g = \frac{q_0(q_0-1)}{2}$. Then the same estimation on the dimension $k$ of the SSAG-code leads to
$$k \geq n - m(n - \dim(G)) = n - m(n - \deg(G) + g - 1),$$
that is $q_0$ should not be too large as well.
- On the other hand, the choice $q = q_0$ and $m = 2$, which could be a good choice about the two previous points, is not encouraged. The formal argument in this direction is that the smaller the dregree extension $m$ is, the closer the structure of the SSAG-code is from the AG one. Since AG codes have been broken in polynomial time (see [CMCP17]), it might be possible to addapt this attack to SSAG-codes if they are too close to AG ones. In section 6.4.4, we will still give some parameters with $m = 2$ because it provides the best key sizes. However, we warn the reader that it could be the weakest keys.

### 6.4.4 Parameters

Let us recall the notation we will use in the following tables.

- $q$ is the power of a prime such that the SSAG-code is defined over $\mathbb{F}_q$.
- $m$ is the extension degree such that the underlying AG-code is defined over $\mathbb{F}_{q^m}$.
- $q_0$ is a prime power such that $q_0^2 = q^m$, such that the Hermitian curve is defined over the field $\mathbb{F}_{q_0^2}$ with square cardinality.
- $n$ and $k$ are respectively the length and the dimension of the SSAG-code.
- $\ell$ is the order of quasi-cyclicity.
- $g(\mathcal{H}/\langle\sigma\rangle)$ and $h(\mathcal{H}/\langle\sigma\rangle)$ are respectively the genus and the class number of the quotient curve.

- The Key sizes are given in bytes, using the formula $\left\lceil \frac{\log_2(q) \cdot \frac{k}{l} \cdot (n-k)}{8} \right\rceil$, since the public key is a systematic and quasi-cyclic generator matrix of the SSAG-code.
- $\omega_{ISD}$ is the $\log_2$ of the work factor for the ISD attack, computed using **CaWoF** library (see. [Tor16]).

We will suggest paramaters for 128 bits of security, keeping in mind that the class number $h(\mathcal{H}/\langle\sigma\rangle)$ is an upper bound for the brute force attack.

| $m$ | $q$ | $q_0$ | $n$ | $k$ | $\ell$ | *Key sizes (bytes)* | $\omega_{ISD}$ | $g(\mathcal{H}/\langle\sigma\rangle)$ | $h(\mathcal{H}/\langle\sigma\rangle)$ |
|---|---|---|---|---|---|---|---|---|---|
| 8 | 2 | $2^4$ | 4083 | 2307 | 3 | 170718 | 128 | 40 | $\simeq 2^{326}$ |
| 8 | 2 | $2^4$ | 4085 | 2315 | 5 | 102438 | 129 | 24 | $\simeq 2^{196}$ |
| 4 | $2^2$ | $2^4$ | 3000 | 1246 | 3 | 182123 | 128 | 40 | $\simeq 2^{326}$ |
| 4 | $2^2$ | $2^4$ | 3000 | 1252 | 5 | 109424 | 129 | 24 | $\simeq 2^{196}$ |
| 3 | $3^2$ | $3^3$ | 2996 | 1277 | 7 | 124258 | 130 | 39 | $\simeq 2^{374}$ |

Table 1: Suggested parameters for security 128, $m > 2$

For some parameters with $m = 2$ in the table below, one have $h(\mathcal{H}/\langle\sigma\rangle) < 2^{128}$. In this case, we precise the number $\#\,\mathrm{SSAG}_{q,r}(\mathcal{H}, \mathcal{P})$ of SSAG-codes over $\mathbb{F}_q$ with same support $\mathcal{P}$ and a degree divisor equals to $r$.

| $m$ | $q$ | $n$ | $k$ | $\ell$ | *Key sizes (bytes)* | $\omega_{ISD}$ | $g(\mathcal{H}/\langle\sigma\rangle)$ | $h(\mathcal{H}/\langle\sigma\rangle)$ | $\#\,\mathrm{SSAG}_{q,r}(\mathcal{H}, \mathcal{P})$ |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 11 | 900 | 613 | 3 | 25359 | 128 | 15 | $\simeq 2^{107}$ | $\simeq 2^{6316}$ |
| 2 | 11 | 1200 | 740 | 5 | 29439 | 128 | 11 | $\simeq 2^{78}$ | $\simeq 2^{8360}$ |
| 2 | 13 | 1299 | 775 | 3 | 62614 | 128 | 26 | $\simeq 2^{197}$ | $\simeq 2^{9793}$ |
| 2 | 13 | 994 | 659 | 7 | 14587 | 128 | 6 | $\simeq 2^{45}$ | $\simeq 2^{7386}$ |

Table 2: Suggested parameters for security 128, $m = 2$

# A Retrieving the equation of a cover : complexity analysis

In this first Annex, we will present a formal algorithm that describes the attack proposed in Section 5.2 in the special case of a Kummer covering of $\mathbb{P}^1(\mathbb{F}_{q^m})$. Recall that in this special case, we are led to solve linear systems of the form :

$$\begin{pmatrix} \mathbf{A}(\xi) \\ \mathbf{H} \cdot \mathbf{D}_1 \\ \vdots \\ \mathbf{H} \cdot \mathbf{D}_s \end{pmatrix} \cdot \mathbf{y}^T = 0, \qquad (\Delta(\xi))$$

that is a system with $s(n-k)+n$ equations for $n$ unknows; where $s = \dim(D)$ (see Section 4) and $n$,$k$ are respectively the lenght and dimension of the public SSAG-code. Let us suppose that a plane model of the Kummer covering is given by

$$y^\ell = f(x),$$

with $f \in \mathbb{F}_{q^m}[T]$ and $d := \deg(f)$ (note that $d$ is both the pole divisor degree of $f$ and its degree as a polynomial). The following proposition gives the complexity of Algorithm 1 below

**Proposition 44.** *Let $n, k$ be the lenght and the dimension of the public* SSAG*-code. Let $r := n/\ell$ be the number of orbits in $\mathcal{P}$. If $r \geq d+1$, then Algorithm 1 finds an equation of the cover, as well as the secret structure of the public* SSAG*-code in $\mathcal{O}(\varphi(\ell)(n^\omega + n^{\omega-1}s(n-k))$ operations over $\mathbb{F}_{q^m}$.*

*Proof.* see [BCG+17] for complexity analysis.
Recall that the complexity of solving a linear system with $k$ equations and $n$ unknowns is in $\mathcal{O}(n^{\omega-1}k)$ operations over the base field, where $\omega$ is the exponent of linear algebra. As a result, the cost of line 9. is $\mathcal{O}(n^\omega + n^{\omega-1}s(n-k))$ operations over $\mathbb{F}_{q^m}$. Since we have to seek for the correct root of unity $\xi$, this step might be repeated at most $\varphi(\ell)$-times, where $\varphi$ is the Euler totient function. Next, one have to realise one Lagrange's interpolation at line 13., in order to recover a defining equation of the Kummer cover. In particular, let $d$ denote the degree of the polynomial $f$ we have to build. At this step of the algorithm, we have recovered all the points in $\mathcal{P}$, and thus if the number $r$ of orbits in $\mathcal{P}$ is larger that $d+1$, Lagrange's interpolation finds a unique polynomial $f$ of degree $d$ such that a plane model of the cover is given by $y^\ell = f(x)$ in $\mathcal{O}(d^2)$ operations over $\mathbb{F}_{q^m}$. Note that this step is negligible compared to the cost of line 9. Finally, the last step we have to care about is at line 17. In fact, at this stage of the algorithm, the whole cover is known and it remains to compute the pullback of the invariant divisor. As for the support, we need to recover the $y$-coordinates of points in $\mathrm{Supp}(G)$. This can be done by finding roots of several polynomials. Indeed, from Kummer's theorem (see [Sti09], Theore 3.3.7), if $x(Q)$ denotes the $x$-coordinate of a point $Q \in \mathrm{Supp}(\tilde{G})$, then the $y$-coordinates of the extensions of $Q$ in $\mathrm{Supp}(G)$ are exactly the roots of the polynomial $P_Q(T) = T^\ell - f(x(Q)) \in \mathbb{F}_{q^m}[T]$. This step can be done by factorizing each polynomial $P_Q$ using Berlekamp algorithm, whose cost is $\mathcal{O}(\ell^\omega + q^m\ell^2)$ operations over $\mathbb{F}_{q^m}$. In any practical cases, the lenght of the public code is larger that the cardinality of the base field, that is $n > q^m$ and thus this step is also negligible. As a result, the total cost of Algorithm 1 is in $\mathcal{O}(\varphi(\ell)(n^\omega + n^{\omega-1}s(n-k))$ over $\mathbb{F}_{q^m}$. □

Note that this algorithm can be used in the case of Artin-Schreier covers of the projective line, by only changing a few lines. In fact, in the Artin-Schreier setup, we have to solve at most $p = \#(\mathbb{F}_p)$ linear system of the form (8), whose total cost is in $\mathcal{O}(p(n^\omega + n^{\omega-1}s(n-k))$ over $\mathbb{F}_{q^m}$. As in Algorithm 1, this is the total cost of the correponding algorithm since others steps are the same.

**Remark 45.** In Algorithm 1, we describe our attack in the special case were the quotient curve is $\mathbb{P}$, as it is easier to produce a complexity analysis. However, note that this algorithm can be generalized to general Kummer or Artin-Schreier cover, the only difference being the interpolation step at the end. In fact, in the general case, the polynomial we have to recover is bivariate, thus the interpolation step might be harder. To be a bit more precise, it is possible to use a bivariate Lagrange's interpolation, and the complexity is the same as the classical case, exepted that we need more evaluation points, ie. at least $\binom{d^*+2}{d^*}$ of them, where $d^*$ is the degree of $f$ as a bivariate polynomial.Another point that could

be problematic is that the corresponding system might have more than one solution.
This is the main counterpart of considering polynomial instead of rational functions in our settings. In fact, it could be possible to use Lagrange's interpolation in a function field of a curve insted of using it on polynomials, but this theory is not well-developed yet.

**Algorithm 1 : Security reduction in Kummer case**

**Inputs:**

- A parity check matrix $\mathbf{H}_{\text{pub}}$ of the public code;
- The full invariant structure, that is $\tilde{P} = (P_1, ..., P_r)$ and $\tilde{G}$;
- The quasi-cyclicity order $\ell$ and the degree $m = \deg(f)$.

**Outputs:**

- The polynomial $f$;
- The secret structure $(\mathcal{P}, G)$.

1. $x \longleftarrow (\underbrace{x(P_1), ..., x(P_1)}_{\ell-\text{fois}}, ..., x(P_r), ..., x(P_r))$
2. $D \longleftarrow \tilde{G} - \lceil d/\ell \rceil \cdot P_\infty \in \text{Div}(\mathbb{P}^1)$
3. $s \longleftarrow \dim(D)$
4. $M \longleftarrow$ set of primitive $\ell$-th roots of unity
5. cpt := 0
6. while cpt := 0 do
7. $\quad \xi \xleftarrow{\$} M$
8. $\quad \text{Exclude}(M, \xi)$
9. $\quad S \longleftarrow \text{Solve}(\Delta(\xi))$
10. $\quad$ if $\dim(S) = 1$ then
11. $\quad\quad$ cpt := 1
12. $\quad\quad y^* \xleftarrow{\$} S \backslash \{0\}$
13. $\quad\quad f \longleftarrow \text{Interpolate}(x, y^*)$
14. $\quad\quad \mathcal{P} \longleftarrow \{P_{ij} = (x_{ij} : y_{ij} : 1)\}$
15. $\quad$ end if
16. end while
17. $G \longleftarrow \pi^*(\tilde{G})$
18. Return $\mathcal{X}$, $\mathcal{P}$ and $G$

**Remark 46.** In section 5.4, we generalized our attack in the case of a solvable Galois covering, instead of just a cyclic one. We explained that in order to attack the corresponding public code, we could apply several times the Kummer or Artin-Schreier case. The total cost of the attack in this context is just the number of iterations we have to use Algorithm 1. In particular, if the corresponding solvable

Galois group have order $p^s\lambda$, with $p = \mathrm{char}(\mathbb{F}_{q^m})$ and $(p, \lambda) = 1$. Then we can attack the public code by using $s + 1$-times Algorithm 1.

# B Brute force Algorithm on the invariant code

In section 6.3, we described a brute force attack against the invariant SSAG-code in our settings on the Hermitian curve. This section will be dedicated to a formal algorithm for this attack. Let $r \geq 0$, and denote by $\mathrm{Cl}^r(\mathcal{H}/\langle\sigma\rangle)$ the group of divisor class of degree $r$ in $\mathcal{H}/\langle\sigma\rangle$. For a divisor $G \in \mathrm{Div}(\mathcal{H}/\langle\sigma\rangle)$, we denote $[G]$ its class in $\mathrm{Cl}^r(\mathcal{H}/\langle\sigma\rangle)$. In Algorithm 2 below, we plan to recover the good permutation amoung all subsets of lenght $n$, and then use the SSA algorithm (see [Sen00]). In order to find the good divisor amoung all classes of a given degree, we essentially use the estimation (14).

**Algorithm 2 : Brute force on** SSAG

**Inputs:**

- A generator matrix $\mathbf{G}$ of the invariant SSAG-code $\mathcal{C}$;
- The degree $r$ of the divisor associated to $\mathcal{C}$.

**Outputs:**

- A couple $(\mathcal{P}, G)$ such that $\mathcal{C} = \mathrm{SSAG}_q(\mathcal{H}/\langle\sigma\rangle, \mathcal{P}, G)$.

1. $S \longleftarrow \{P \in \mathcal{H} \mid \deg(P) = 1\}$ // the lenght is fixed
2. for $[G] \in \mathrm{Cl}^r(\mathcal{H})$ do
3. $\quad \mathcal{C}' \longleftarrow \mathrm{SSAG}_q(\mathcal{H}, S, G)$
4. $\quad$ for $[\mathbf{w}] \in \mathbb{F}_{q^m}^n/\mathbb{F}_q^n$ do // $\mathbf{w}$ representative of $[\mathbf{w}]$
5. $\quad\quad$ for $\mathcal{I} \subseteq \{1, ..., N(\mathcal{H}) - 1\}$ with $|\mathcal{I}| = n$ do
6. $\quad\quad\quad \mathcal{C}'_{\mathcal{I}} \longleftarrow \mathrm{Punct}_{\mathcal{I}}(\mathcal{C}')$
7. $\quad\quad\quad \pi \longleftarrow \mathrm{SSA}(\mathbf{w} \star \mathcal{C}'_{\mathcal{I}}, \mathcal{C})$ // SSA return a permutation or '?'
8. $\quad\quad\quad$ If $\pi \in S_n$ then
9. $\quad\quad\quad\quad S_{\mathcal{I}} \longleftarrow \{P_i \in S \mid i \in \mathcal{I}\}$
10. $\quad\quad\quad\quad$ return $\pi(S_{\mathcal{I}}), G$ and $\mathbf{w}$

As it is complicated to provide a complexity analysis for the support recovering, we can at least say that the complexity of Algorithm 2 is at least the cost of the exhaustive search on $G$ and $\mathbf{w}$, that is at least (see (14))

$$((q^m - 1)^{n-1} - (q - 1)^{n-1})h(\mathcal{H}/\langle\sigma\rangle) \text{ operations over } \mathbb{F}_q.$$

By Theorem 1, we know that

$$(\sqrt{q^m} - 1)^{2g(\mathcal{H}/\langle\sigma\rangle)} \leq h(\mathcal{H}/\langle\sigma\rangle) \leq (\sqrt{q^m} - 1)^{2g(\mathcal{H}/\langle\sigma\rangle)}.$$

Then we can write $h(\mathcal{H}/\langle\sigma\rangle) \in \mathcal{O}(q^{mg(\mathcal{H}/\langle\sigma\rangle)})$ and the cost of the enumeration of divisors $G$ in $\mathrm{Cl}^r(\mathcal{H}/\langle\sigma\rangle)$ ans vectors in $\mathbb{F}_{q^m}/\mathbb{F}_q$ is in $\mathcal{O}(q^{m(n-1)+mg(\mathcal{H}/\langle\sigma\rangle)})$. To conclude, note that this complixity can be precised in our context for our choice of $\sigma$ (see Proposition 42 and Corollary 43).

Our suggested parameters for this scheme (see section 6.4.4) are chosen specifically in order $h(\mathcal{H}/\langle\sigma\rangle)$ to be high enough to block the brute force attack.

# References


[Bar17] Elise Barelli. On the security of some compact keys for mceliece scheme. *WCC Workshop on Coding and Cryptography*, September 2017.

[Bar18] Elise Barelli. *On the security of short McEliece keys from algebraic and algebraic geometry codes with automorphisms*. PhD thesis, universite Paris-Saclay, 2018.

[BCG+17] Alin Bostan, Frédéric Chyzak, Marc Giusti, Romain Lebreton, Grégoire Lecerf, Bruno Salvy, and Éric Schost. *Algorithmes Efficaces en Calcul Formel*. Frédéric Chyzak (auto-édit.), Palaiseau, September 2017.

[CMCP14] Alain Couvreur, Irene Marquez-Corbella, and Ruud Pellikaan. A polynomial time attack against algebraic geometry codes based public key cryptosystems. *Proc. IEEE Int. Symposium Inf. Theory- ISIT*, pages 1446–1450, 2014.

[CMCP17] Alain Couvreur, Irene Marquez-Corbella, and Ruud Pellikaan. Cryptanalysis of mceliece cryptosystem based on algebraic geometry codes ans their subcodes. *IEEE Trans. Inform. Theory* **63**, pages 5404–5418, 2017.

[FOP+16a] Jean-Charles Faugere, Ayoub Otmani, Ludovic Perret, Frederic de Portzamparc, and Jean-Pierre Tillich. Folding alternant and goppa codes with non-trivial automorphism groups. *IEEE. Trans. Inform. Theory* **62**, pages 184–198, 2016.

[FOP+16b] Jean-Charles Faugere, Ayoub Otmani, Ludovic Perret, Frederic de Portzamparc, and Jean-Pierre Tillich. Structural cryptanalysis of mceliece schemes with compact keys. *Des. Codes Cryptogr.* **79**, pages 87–112, 2016.

[FOPT10] Jean-Charles Faugere, Ayoub Otmani, Ludovic Perret, and Jean-Pierre Tillich. Algebraic cryptanalysis of mceliece variants with compact keys. *Advances in Cryptology - EUROCRYPT 2010, LNCS*, pages 279–298, 2010.

[GSX00] Arnaldo Garci, Henning Stichtenoch, and Chao-Ping Xing. On subfields of the hermitian function field. *Compositio Mathematica* **120**, pages 137–170, 2000.

[HP95] Tom Høholft and Ruud Pellikaan. On the decoding of algebraic-geometric codes. *IEEE Transactions on Information Theory* **41**, pages 1589–1614, 1995.

[Loi01] Pierre Loidreau. Codes derived from binary goppa codes. *Probl. Inf. Transm.* **37**, pages 91–99, 2001.

[McE78] Robert J. McEliece. A public-key cryptosystem based on algebraic coding theory. *DSN Progress Report 44*, pages 114–116, 1978.

[MP93] Carlos Munuera and Ruud Pellikaan. Equality of geometric goppa codes and equivalence of divisors. *Journal of Pure and Applied Algebra* **90**, pages 229–252, 1993.

[Sen00] Nicolas Sendrier. Finding the permutation between equivalent linear codes: The support splitting algorithm. *IEEE Transactions on Information Theory* **46**, pages 1193–1203, 2000.

[Sti09] Henning Stichtenoch. *Algebraic Function Fields and Codes*. Springer, 2nd edition edition, 2009.

[TaDN07] Michael A. Tsfasman and Serge G. Vladut annd Dmitry Nogin. *Algebraic geometric codes : basic notions*. no.139, American Mathematical Soc., 2007.

[Tor16] Rodolfo Cantos Torres. Cawof, c library for computing asymptotic exponents of generic decoding work factors. pages 5404–5418, 2016.