Structural attack on structured-SSAG-code-based McEliece cryptosystems

Elise Barelli
Philippe Lebacque*
Mathieu Lhotel†
Hugues Randriam‡

September 16, 2022

Abstract. *In this paper, we study the security of variants of McEliece's cryptosystem, which are based on structured (e.g. quasi-cyclic) subfield-subcodes of algebraic geometry codes (SSAG codes). More precisely, we show that the underlying secret structure of the public SSAG code can be recovered from that of its invariant subcode, which is also an SSAG code but with smaller parameters.*

After reviewing known attacks against this class of codes, we then finally propose a McEliece-like scheme based on quasi-cyclic SSAG Hermitian codes, together with a set of parameters that seem to resist these attacks.

*Laboratoire de Mathématiques de Besançon, UMR 6623 CNRS, Université de Bourgogne Franche-Comté, France
†Laboratoire de Mathématiques de Besançon, UMR 6623 CNRS, Université de Bourgogne Franche-Comté, France
‡ANSSI, Laboratoire de Cryptographie & LTCI équipe C2, Télécom Paris

1 Introduction

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 key, *i.e.* a generator matrix of the code, is too large for most practical use cases. Many propositions were made in order to correct this, mostly by considering codes with additional structure, *e.g.* quasi-cyclic codes [BBB+17]. Indeed this makes the public key more compact, since in order to specify a generating matrix of the code, it then suffices to give its first row only.

Moreover, replacing classical Goppa codes, defined over the projective line, by their natural generalization, defined on curves of higher genus, should 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 schemes based on structured SSAG codes. The security of such a scheme relies on the fact that, given a random generator matrix of an SSAG code, one should not be able to decode it more efficiently than a generic linear code. In particular, it should be difficult to recover the geometric data from which the code was defined, or more generally, to find any geometric data giving an equivalent definition (for this would provide an efficient decoding algorithm). However, in our setting where we have an additional structure (*e.g.* quasi-cyclicity), we show that the structural recovery of the public SSAG code reduces to that of its invariant subcode. This invariant subcode is also an SSAG code, of which a generator matrix is easily derived from the public data, and it has smaller parameters. As a consequence, the parameters of the scheme must be chosen carefully in order to keep this invariant SSAG subcode immune against structural recovery attacks. This leads us to review known attacks of this sort, after which we can finally propose a scheme based on quasi-cyclic SSAG Hermitian codes, and provide a set of parameters that seem to secure the corresponding invariant code.

Beside structural recovery attacks, it is also interesting to discuss message recovery attacks. Essentially the only approach for message recovery is via decoding algorithms for generic linear codes. All these algorithms, most of which are variants of Prange's algorithm [Pra62], have exponential complexity. Moreover they do not seem to benefit from the additional quasi-cyclic structure. More precisely, given a received word, the additional structure provides other words with a same weight error, on which one could apply Sendrier's Decoding-One-Out-of-Many Algorithm [Sen11]. But ultimately this only gives a negligible advantage. As a consequence, we see that although the structured SSAG code is equivalent to its invariant subcode from the point of view of structural recovery, then from the point of view of message recovery, it remains more secure.

Organization of the paper. In section 2, we give some classical notations and definitions that will be useful, both in algebraic geometry and coding theory. Section 3 is dedicated to the invariant code, with a focus on its geometric structure. In section 4, we present our attack in a general framework, while specific instances such as the Kummer and Artin-Schreier cases are given in section 5. General attacks against SSAG codes are discussed in section 6, such as the brute force search. We conclude with the proposition of a scheme based on SSAG Hermitian codes in section 7.

Genesis of this work. This work can be seen as an extension of Chapter 5 of the first author's PhD memoir [Bar18a] and of her unpublished manuscript [Bar18b]. The general method introduced in [Bar18a] was slightly simplified, avoiding an unnecessary duality argument in the security reduction, and it was also extended in order to apply to a more general geometrical situation. On the other hand, the proposed scheme based on quasi-cyclic SSAG Hermitian code was left essentially unchanged.

Acknowledgements. The authors benefited from the support of ANR-21-CE39-0009 Barracuda. Also, through the LMB, the second and third authors received support from the EIPHI Graduate School (contract ANR-17-EURE-0002).

2 Notations and properties

2.1 Algebraic function fields and curves

Let $\mathbb{F}$ be a finite field. A function field (of one variable) over $\mathbb{F}$ is a field K that is finitely generated and of transcendance degree one over $\mathbb{F}$. Our main reference for function fields will be [Sti09]. It is well known that the theory of function fields over $\mathbb{F}$ is equivalent to that of algebraic curves over $\mathbb{F}$. More precisely:

Theorem 1. *There is an anti-equivalence between the following two categories:*

- *smooth, irreducible, projective curves over $\mathbb{F}$, with non-constant morphisms of curves over $\mathbb{F}$*
- *function fields over $\mathbb{F}$, with field morphisms over $\mathbb{F}$,*

which to a curve X associates its function field $K = \mathbb{F}(X)$.

(A proof can be found in [Liu02, Prop. 7.3.13 and Rem. 7.3.14]; observe that over a finite field, or more generally over a perfect field, an irreducible projective curve is normal iff it is smooth. One can also look at [Ful69, §7.5] or [Har77, Cor. I.6.12], which give a proof over an algebraically closed field, and conclude using a base change argument.)

In accordance with [Sti09, sec. 1.4 ff.], when speaking about a function field K over $\mathbb{F}$, we will always implicitly assume that the field of constants of K is reduced to $\mathbb{F}$. Alternatively, that means that we will always implcitly assume that our curves are *geometrically irreducible*.

Thus, throughout this paper, the word "curve" will be a shorthand for "smooth, geometrically irreducible, projective curve".

A function field K over $\mathbb{F}$ can always be described as $K = \mathbb{F}(x)[y]$ where x is transcendental over $\mathbb{F}$ and y is a primitive element for K over $\mathbb{F}(x)$. There is then an (absolutely) irreducible polynomial f in two variables over $\mathbb{F}$, unique up to multiplication by a constant, such that $f(x, y) = 0$. This description might be convenient for computations. Observe that the zero locus $V(f)$ of f in the affine plane, and also its closure in the projective plane, might be singular. In order to recover the (smooth) curve X associated to K, we have to consider the normalization of the projective closure of $V(f)$. Occasionally we will say that $V(f)$ is a (possibly singular) *plane model* of X.

Let $\mathbb{P}_K$ be 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 $\deg(P) := [O_P/P : \mathbb{F}] < \infty$. The divisor group of K, denoted $\mathrm{Div}(K)$, is the free abelian group on $\mathbb{P}_K$ generated by the sums

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

where $\mathrm{Supp}(A)$ is a finite subset of $\mathbb{P}_K$, called the support of A, whose elements are places such that $\nu_P(A) \in \mathbb{Z}\backslash\{0\}$.
An element $P \in \mathrm{Supp}(A)$ is called a zero of A if $\nu_P(A) > 0$ (resp. a pole of A if $\nu_P(A) < 0$). Given $z \in K$, we denote by $(z)^K$, $(z)_0^K$ and $(z)_\infty^K$ its associated divisor, divisor of its zeros and divisor of its poles respectively, meaning we have $(z)^K = (z)_0^K - (z)_\infty^K$, 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 denote by $\mathrm{Princ}(K)$ 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 $h(K)$. Finally, the Weil bound implies

Theorem 2 (see [TVN07], Proposition 3.1.23)**.** *Let K be an algebraic function field over $\mathbb{F}$ with genus $g(K)$. Then the class number $h(K)$ satisfies*

$$(\sqrt{|\mathbb{F}|}-1)^{2g(K)} \leq h(K) \leq (\sqrt{|\mathbb{F}|}+1)^{2g(K)}.$$

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

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

of dimension $\ell(A) := \dim_{\mathbb{F}}(\mathcal{L}_K(A))$. If there is no ambiguity with the function field, we just write $\mathcal{L}(A)$.

Let $\pi : \mathcal{Y} \to \mathcal{X}$ be a morphism between two curves $\mathcal{X}$ and $\mathcal{Y}$ over $\mathbb{F}$, whose function fields are given by $L := \mathbb{F}(\mathcal{Y})$ and $K := \mathbb{F}(\mathcal{X})$. We define the pullback of $P \in \mathbb{P}_K$ as the divisor

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

where $e(Q|P)$ denotes the ramification index as usual. 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 3. *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 codes, meaning that we start by defining AG codes over an extension $\mathbb{F}_{q^m}$ of $\mathbb{F}_q$, where $q = p^s$ be a prime power and $m \geq 1$ is an integer, and then we get codes over $\mathbb{F}_q$ using the subfield-subcode construction.

Definition 4. Let $\mathcal{Y}$ be a smooth irreducible projective curve over $\mathbb{F}_{q^m}$ with function field $L = \mathbb{F}_{q^m}(\mathcal{Y})$, $\mathcal{Q} = \{Q_1, ..., Q_n\}$ an ordered set of n distinct places of degree one in L and $G \in \mathrm{Div}(L)$ 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 by

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

Recall that the dual code $\mathcal{C}^\perp$ of a linear code $\mathcal{C}$ over $\mathbb{F}_{q^m}$ is

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

Equivalently, $\mathcal{C}^\perp$ is the linear code over $\mathbb{F}_{q^m}$ generated by any parity check matrix of $\mathcal{C}$.

We will often use the fact that the dual of an AG code is an AG code:

Proposition 5. *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*

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

See *e.g.* [Sti09], Proposition 2.2.10 for the proof.

In this work we will be more specifically interested in "structured" codes, which means codes with a non trivial permutation group, coming from the underlying geometry.

In order to fix notations, let $\mathrm{Aut}(\mathcal{Y}/\mathbb{F}_{q^m})$ be the automorphism group of $\mathcal{Y}$ over $\mathbb{F}_{q^m}$, acting on the left. Thanks to the anti-equivalence of categories in Theorem 1, we can identify $\mathrm{Aut}(L/\mathbb{F}_{q^m})$ with $\mathrm{Aut}(\mathcal{Y}/\mathbb{F}_{q^m})$, acting on L on the right: for $\sigma \in \mathrm{Aut}(L/\mathbb{F}_{q^m}) = \mathrm{Aut}(\mathcal{Y}/\mathbb{F}_{q^m})$ and $f \in L$ we set

$$f^\sigma := \sigma^* f = f \circ \sigma.$$

Now consider a finite subgroup $\Sigma \subseteq \mathrm{Aut}(\mathcal{Y}/\mathbb{F}_{q^m})$, and assume that $\mathcal{Q}$ and G are invariant under Σ; observe that this means precisely that $\mathcal{Q}$ is a union, and G a sum, of orbits under Σ. By invariance of $\mathcal{Q}$, each $\sigma \in \Sigma$ induces a permutation $\tilde{\sigma}$ of the index set $\{1, \ldots, n\}$, such that

$$\sigma(Q_i) = Q_{\tilde{\sigma}(i)}.$$

Then $\tilde{\sigma}$ actually defines a permutation automorphism of $C_{\mathcal{L}}(\mathcal{Y}, \mathcal{Q}, G)$: indeed, if

$$c = \mathrm{Ev}_{\mathcal{Q}}(f) \in C_{\mathcal{L}}(\mathcal{Y}, \mathcal{Q}, G)$$

for $f \in \mathcal{L}(G)$, then $f^{\sigma} \in \mathcal{L}(G)$ by invariance of G, and so

$$c^{\tilde{\sigma}} = \mathrm{Ev}_{\mathcal{Q}}(f^{\sigma}) \in C_{\mathcal{L}}(\mathcal{Y}, \mathcal{Q}, G).$$

As σ ranges in Σ, these $\tilde{\sigma}$ form a subgroup $\tilde{\Sigma}$ of the permutation group of $C_{\mathcal{L}}(\mathcal{Y}, \mathcal{Q}, G)$, acting on the right.

Now we come to SSAG codes.

Definition 6. With notations as in Definition 4, $\mathrm{SSAG}_q(\mathcal{Y}, \mathcal{Q}, G)$ is the subfield subcode of $C_{\mathcal{L}}(\mathcal{Y}, \mathcal{Q}, G)$ over $\mathbb{F}_q$:

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

It is clear that any permutation automorphism of a linear code stabilizes its subfield subcodes. As a consequence, given $\Sigma \subseteq \mathrm{Aut}(\mathcal{Y}/\mathbb{F}_{q^m})$, and assuming $\mathcal{Q}$ and G invariant under Σ as above, we also get a subgroup $\tilde{\Sigma}$ of the permutation group of $\mathrm{SSAG}_q(\mathcal{Y}, \mathcal{Q}, G)$, acting on the right.

Later on we will occasionally abuse notations and write σ for $\tilde{\sigma}$, and Σ for $\tilde{\Sigma}$.

3 Invariant code

Definition 7. Given a linear code $\mathcal{C}$ together with a subgroup Σ of its group of permutation automorphisms, we define the invariant code of $\mathcal{C}$ under Σ as the subcode

$$\mathcal{C}^{\Sigma} := \{c \in \mathcal{C} \mid c^{\sigma} = c, \ \forall \sigma \in \Sigma\}.$$

Note that this code has repeated entries (*i.e.* all its words are constant on each orbit under Σ), so we usually use another one: the *punctured invariant code*. It is denoted $\overline{\mathcal{C}}^{\Sigma}$, and is obtained from $\mathcal{C}^{\Sigma}$ by keeping only one entry in each orbit.

Example 8. Suppose $\ell | n = \mathrm{Length}(\mathcal{C})$, and $\Sigma = \langle \sigma \rangle$ is cyclic of order ℓ, generated by the ℓ-quasi-cyclic shift σ. Let also $I_\ell := \{1, ..., n\} \backslash \{1, \ell + 1, ..., n - \ell + 1\}$. In this situation, the punctured invariant code can also be defined as

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

where for any linear code $\mathcal{C}$ and any set of indices I, $\mathrm{Punct}_I(\mathcal{C}) := \{(c_i)_{i \in \{1,\ldots,n\} \backslash I} \mid \mathbf{c} \in \mathcal{C}\}$. 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.

Later on, we will always work only with the punctered invariant code. We will then occasionally abuse notations and write $\mathcal{C}^{\Sigma}$ for $\overline{\mathcal{C}}^{\Sigma}$.

In what follows, we are interested in the case where $\mathcal{C}$ is an AG code (resp. a SSAG code), and Σ comes from a group of automorphisms of $\mathcal{Y}$, as in the previous section. We study the structure of the invariant code, which will be an important result for the next parts. In particular, we show that this subcode is also an AG code (resp. a SSAG code), defined on the quotient curve $\mathcal{Y}/\Sigma$, whose function field is precisely 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 4, and suppose that they are invariant under an automorphism $\sigma \in \mathrm{Aut}(\mathcal{Y}/\mathbb{F}_{q^m})$. If a codeword $c = \mathrm{Ev}_{\mathcal{Q}}(f) \in C_{\mathcal{L}}(\mathcal{Y}, \mathcal{Q}, G)$ is σ-invariant,* i.e *if $c^{\sigma} = c$, then $f^{\sigma} = f$ also.*

Proof. Let us write $\mathcal{Q} = \{Q_1, ..., Q_n\}$. We have

$$0 = c^\sigma - c = \mathrm{Ev}_{\mathcal{Q}}(f^\sigma - f)$$

so $f^\sigma - f$ admits at least n zeroes $Q_1, ..., Q_n$. Now, since G is σ-invariant, we have $f^\sigma \in \mathcal{L}(G)$, hence $f^\sigma - f \in \mathcal{L}(G)$. However we just saw that this function admits n zeroes, with $n > \deg(G)$ by hypothesis, which is possible only if $f^\sigma - f = 0$. □

Definition 10. Consider an extension L/K of function fields, corresponding to a morphism of curves $\pi : \mathcal{Y} \to \mathcal{X}$. Given a divisor $G \in \mathrm{Div}(L)$, we denote by $\tilde{G} \in \mathrm{Div}(K)$ the largest divisor such that

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

Lemma 11. *With notations as above:*

(i) $\mathcal{L}_L(G) \cap K = \mathcal{L}_K(\tilde{G})$

(ii) if we write

$$G = \sum_{S \in \mathbb{P}_L} t_S S$$

for integers t_S almost all of which are 0, then

$$\tilde{G} = \sum_{R \in \mathbb{P}_K} \left(\min_{S|R} \left\lfloor \frac{t_S}{e(S|R)} \right\rfloor \right) R$$

(iii) if $A \in \mathrm{Div}(K)$, we have

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

Proof. *(i)* Given $h \in K$ we have $(h)_L = \pi^*((h)_K)$, and thus we have $h \in \mathcal{L}_L(G)$ iff $\pi^*((h)_K) \geq -G$. But by Definition 10 this means precisely $(h)_K \geq -\tilde{G}$, *i.e.* $h \in \mathcal{L}_K(\tilde{G})$.

(ii) If $\tilde{G} = \sum_{R \in \mathbb{P}_K} x_R R$, then

$$\pi^* \tilde{G} = \sum_{R \in \mathbb{P}_K} \sum_{S|R} x_R e(S|R) S,$$

and so $\pi^* \tilde{G} \leq G$ iff

$$x_R \leq \frac{t_S}{e(S|R)}$$

for all $R \in \mathbb{P}_K$ and all $S|R$. Then $\tilde{G}$ is maximal when all x_R are maximal under this condition, which means precisely

$$x_R = \min_{S|R} \left\lfloor \frac{t_S}{e(S|R)} \right\rfloor.$$

(iii) Obvious (or direct consequence of *(ii)*).

□

Proposition 12. *Let L/K be a Galois extension of function fields over $\mathbb{F}_{q^m}$, with Galois group Σ, and let $G \in \mathrm{Div}(L)$ be a Σ-invariant divisor. Then:*

(i) $\mathcal{L}_L(G)^\Sigma = \mathcal{L}_K(\tilde{G})$

(ii) if we write

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

for places $S_1, ..., S_s \in \mathbb{P}_L$ nonconjugate under Σ and integers $t_i \in \mathbb{Z} \backslash \{0\}$, then

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

where $R_i \in \mathbb{P}_K$ is such that $S_i | R_i$.

Proof. (i) Since $\mathcal{L}_L(G)^\Sigma = \mathcal{L}_L(G) \cap L^\Sigma = \mathcal{L}_L(G) \cap K$, this is a special case of (i) in the previous Lemma.

(ii) Observing that G is Σ-invariant and that $e(Q|R_i) = e(S_i|R_i)$ for all $Q \in \mathrm{Orb}_\Sigma(S_i)$, this is again a special case of (ii) in the previous Lemma.

□

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 $\mathcal{Q}$ and G invariant under the action of $\Sigma \subseteq \mathrm{Aut}(\mathcal{Y}/\mathbb{F}_{q^m})$. Then its invariant code under Σ 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 made more precise, since the proof given make $\mathcal{P}$ and $\tilde{G}$ explicit. Indeed, $\tilde{G}$ is given by the formula in Proposition 12 (ii), and

$$\mathcal{P} = \{Q \cap L^\Sigma \ , Q \in \mathcal{Q}\}.$$

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

Remark 15. It is readily seen that the invariant and subfield subcode operations commute. More generally, if $\mathcal{C} \subseteq \mathbb{F}_{q^m}^n$ is a linear code and Σ is a group of permutation automorphisms of $\mathcal{C}$, then

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

Corollary 16. *With the notations of Theorem 13, we have*

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

where $\mathcal{P}$ and $\tilde{G}$ are given 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 smooth, irreducible and projective 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

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

In order to recover an equation of $\mathcal{Y}$, we need to find the minimal polynomial H of y over K. Denote by $\ell = [L : K]$ the degree of the extension L/K (*i.e.* the order of its Galois group) and 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

$$\pi^* P_i = Q_{i,1} + ... + Q_{i,\ell},$$

where $Q_{i,j}|P_i$. In particular, it means that for all $1 \leq i \leq r$, we have

$$\mathrm{Orb}_{\mathrm{Gal}(L/K)}(Q_{i,1}) = \{Q_{i,1} + ... + Q_{i,\ell}\}.$$

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. Note that the set $\mathcal{Q}$ is an ordered set, which in general is not ordered by orbits of length ℓ (we will sometimes make this assumption later to simplify the computations).
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 satisfies $\mathrm{Supp}(\tilde{G}) \cap \mathcal{P} = \emptyset$.

Remark 17. The situation above can be described in terms of AG codes. In fact, it is clear from the construction that both the support $\mathcal{Q}$ and the divisor G are invariant under the action of Σ, which then induces a permutation on the AG code $C_{\mathcal{L}}(\mathcal{Y}, \mathcal{Q}, G)$, whose invariant code is then given by $C_{\mathcal{L}}(\mathcal{X}, \mathcal{P}, \tilde{G})$.

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

1. We know a parity check matrix $\mathbf{H}$ of the SSAG code
$$\mathcal{C} := \mathrm{SSAG}_q(\mathcal{Y}, \mathcal{Q}, G);$$
2. We have full knowledge of the invariant code $\mathcal{C}^{\Sigma}$: more precisely, we know the quotient curve $\mathcal{X}$ (and thus its function field K), the finite set of places $\mathcal{P}$ of K and the divisor $\tilde{G} \in \mathrm{Div}(K)$;
3. We have some information about the action of $\mathrm{Gal}(L/K)$ on the unkown support $\mathcal{Q}$. In particular, for any known place $P \in \mathcal{P}$, we know the positions of its ℓ extensions in the ordered set $\mathcal{Q}$;
4. We have *enough information* on the pole divisor of y, where $y \in L$ is such that $L = K(y)$. This assumption will be discussed later, since the key point will be to control the divisor
$$\widetilde{(y)_{\infty}^{L}} \in \mathrm{Div}(K).$$

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 recover the minimal polynomial of y using interpolation. Thanks to hypothesis 2, we know the coordinates of the rational points corresponding to the places P_i's in K. Since places in $\mathcal{Q}$ are extensions of places in $\mathcal{P}$ (*i.e.* for all $Q \in \mathcal{Q}$, there exist $P \in \mathcal{P}$ such that $P = Q \cap K$), we only need to recover the y-evaluation of points in $\mathcal{Q}$. Thus, we are led to find 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 $1 \leq j \leq \ell$. For this purpose, we construst a linear system of which it is solution: recall that by definition, the parity check matrix of the code $\mathcal{C} = \mathrm{SSAG}_q(\mathcal{Y}, \mathcal{Q}, G)$ satisfies

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

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

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

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

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

In particular, since $g \in K$, the numbers $g(Q_{i,j})$ are known for all i, j (g does not depend on y), the right-hand side of (2) gives a system of linear equations where everything is known exept for the $y(Q_{i,j})$.
Since we know the quotient curve (*i.e.* its function field K) as well as the morphism $\pi : \mathcal{Y} \to \mathcal{X}$, we construct the space $\mathcal{L}$ as a set of pullbacks 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* $D := \widetilde{G} - \widetilde{(y)_{\infty}^{L}} \in \mathrm{Div}(K)$. *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 $\mathcal{F} \subseteq \mathcal{L}_K(\tilde{G})$ easyly follows from the fact that $\widetilde{(y)^L_\infty}$ is a positive divisor, and thus $\tilde{G} - \widetilde{(y)^L_\infty} \leq \tilde{G}$. Next we show that (3) holds. Fix $f \in \mathcal{F} := \mathcal{L}_K\left(\tilde{G} - \widetilde{(y)^L_\infty}\right)$. By definition, we have

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

and then

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

Now, we get

$$((\pi^* f) \cdot y)^L = (\pi^* f)^L + (y)^L \geq ((y)^L_\infty - G) + (y)^L = (y)^L_0 - G \geq -G,$$

since $(y)^L_0$ is an effective divisor. In particular, we just proved that $(\pi^* f) \cdot y \in \mathcal{L}_L(G)$ for every $f \in \mathcal{F}$, which is exactly (3). □

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 information 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 optimal.

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

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

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 $1 \leq k \leq s$:

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

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

1. We first compute the vectors $\mathbf{a}_k := (f_k(P_i))_i$, $1 \leq i \leq r$, which can be easily done since both the f_k's and P_i's are known by this point;
2. Next we use hypothesis 3 to recover the $\mathbf{u}_k$'s: by definition of pullbacks, 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.

Using (2) and (3), 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 is 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. In the discussion below, we study how the parameters impact the structure of $\mathrm{Ker}(\mathbf{A})$. If S denotes the number of equations in the linear system (5), we have

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

where $s := \ell(D)$. By definition, the number of rows of $\mathbf{H}$ equals $n - \dim_{\mathbb{F}_q}(\mathcal{C})$. Using Delsarte's theorem, we have

$$\dim_{\mathbb{F}_q}(\mathcal{C}) \geq n - m \dim_{\mathbb{F}_q}(C_{\mathcal{L}}(\mathcal{Y}, \mathcal{Q}, G)),$$

so applying twice the Riemann-Roch theorem show that the number of equations in (5) is closely related to the degree of G (since $\deg(\tilde{G}) \leq \lfloor \frac{\deg(G)}{\ell} \rfloor$) and both genera of L and K, provided that the code length is fixed.
Nevertheless, in all our computing experiments, we noticed that thes do not impact the structure of $\mathrm{Ker}(\mathbf{A})$ 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*

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

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 $\mathbf{A}$.

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

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

Thus,

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

which proves $(\pi^* \mathcal{F}) \cdot h \subseteq \mathcal{L}_L(G)$ and conclude the proof. □

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 Σ, we can add other equations to (5) 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 a defining equation of a smooth irreducible projective curve $\mathcal{Y}$, provided that we are given enough information about one of its quotient curves $\mathcal{X}$. We can wonder for which kind of curve $\mathcal{X}$ we can apply the attack described in section 4: a first idea could be to take $\mathcal{X}$ equals 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 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 authors proposed a new method to break cryptographic schemes based on alternant

codes, called *algebraic attacks*. In particular, they show that the secret structure of these codes satisfy a system of polynomial equations (we give more explanations in section 6.2). Despite the fact that the cost of solving such systems might be hard to estimate, it seems reasonable to consider a more general class of curves for the choice of $\mathcal{X}$ in order to protect future schemes from this attack.

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 ∞ in the projective line.

Definition 21. Let K be a function field of one variable over $\mathbb{F}_{q^m}$. We say it has separated variables if there exist two functions x, α such that $K = \mathbb{F}_{q^m}(x, \alpha)$, with

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

where $F_1, F_2 \in \mathbb{F}_{q^m}[T]$ and $[K : \mathbb{F}_{q^m}(x)] = \deg(F_1)$. Naturally, we say that a smooth irreducible projective curve $\mathcal{X}$ over $\mathbb{F}_{q^m}$ has separated variables if its function field $\mathbb{F}_{q^m}(\mathcal{X})$ has separated variables.

Notice that the above definition hightly rely on the choice of x and α. 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. If ∞ is the pole of x in $\mathbb{F}_{q^m}(x)$, then it totally ramifies in $K/\mathbb{F}_{q^m}(x)$, and its 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_∞ be some extension of ∞ 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 $\deg(F_1)|e(P_\infty|\infty)$, and thus $e(P_\infty|\infty) = \deg(F_1)$. 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 ∞) in the corresponding extension of function fields. This will later allows us to look at the ramification of ∞ in the tower $\mathbb{F}_{q^m}(x) \subseteq K \subseteq L$, 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

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

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

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

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

Set $d := \deg\left((f)_\infty^L\right)$, and 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 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 given in section 4) in this famework. Denote by $\pi : \mathcal{Y} \longrightarrow \mathcal{X}$ a morphism of algebraic curves that corresponds to a Kummer extension of function fields L/K. Suppose we are given an SSAG code $\mathcal{C}$ on $\mathcal{Y}$, that is invariant under the action of $\mathrm{Gal}(L/K)$. This group being well-known, the corresponding action is completely determined by the choice of an ℓ^{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 (and thus both polynomials F_1 and F_2), as well as the structure of the invariant code of $\mathcal{C}$, *i.e.* $\mathcal{P}$ and $\tilde{G}$ such that $\mathcal{C}^\sigma = \mathrm{SSAG}_q(\mathcal{X}, \mathcal{P}, \tilde{G})$;
3. We know how the automorphism $\sigma \in \mathrm{Gal}(L/K)$ permutes the support $\mathcal{Q}$, but we do not know the primitive root of unity ξ that defines it.

As for the last hypothese, we need to control the divisor $\widetilde{(y)_\infty^L}$, which is the point of the upcoming results:

Lemma 24. *Keep notations as in Lemma 22 and denote by P_∞ the unique pole of α in K. Then P_∞ is totally ramified in L/K and its 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 ∞ 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 22), so $(x)^K = \deg(F_1) \cdot P_\infty$. We also know the divisor of poles of α in K, so using Lemma 3 yields

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

and

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

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

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

□

Remark 26. The pole divisor of y we are interested in is particularly simple here because it is only supported by one place, which is the unique extension of the point at infinity ∞ 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 defined 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 $(y)^L_\infty$ given in Proposition 25, it is clear that

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

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

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

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 we can construct the corresponding linear system (5).

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

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

In the context of a Kummer covering, we 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 α, and denote by $\mathbf{h} := \mathbf{x}^i \boldsymbol{\alpha}^j$ its corresponding evaluation vector (at the places in $\mathcal{Q}$). Using Proposition 25, and in particular the description of the pole divisors of x and α, we see that

$$(h)^L_\infty = \ell \left(i \deg(F_1) + j \deg(F_2) \right) \cdot Q_\infty.$$

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

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

Since we have found other potential 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 $\mathbf{y}$. 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 the set $\mathcal{Q}$ is made of r orbits of length ℓ, 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 ℓ 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 ξ is the root of unity that defines σ, and $\mathbf{A}(\xi) \in M_n(\mathbb{F}_{q^m})$. We have

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

and thus

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

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 ℓ, since σ only acts on y-coordinates.

Remark 28. Since, σ (and so ξ) is supposed to be unknown at the beginning of the attack, one may have to test all the possibilities for ξ 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, whose function field $K = \mathbb{F}_q(x, \alpha)$ satisfies

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

with $F_1, F_2 \in \mathbb{F}_{q^m}[T]$ and $\gcd(\deg(F_1), \deg(F_2)) = 1$.
Let $p := \mathrm{char}(\mathbb{F}_{q^m})$ and consider the extension $L = K(y)$ of K defined by

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

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

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

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 is to recover $f \in \mathbb{F}_{q^m}[X, T]$. Using the defining equation of the function field L and the fact that d is prime to p, we 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_∞ its unique extension in L. With our choice of parameters and hypotheses, we can prove:

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, and that the pole divisor of y in L is only supported by P_∞ also. 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, a monomial $x^i \alpha^j \in K$ gives a solution vector if and only if

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

To find the good solution, we again add other equations that are only satisfied by it. Here, the action of σ on $\mathcal{Q}$ 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. We 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

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

where β is the element in $\mathbb{F}_p$ that defines the automorphism σ (note that β is supposed to be unknown here, but as in Kummer case, we can seek for it in a reasonable time). Adding 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. Finally, we can finish the attack by retrieving the polynomial f using an interpolation method.

5.4 Extension to solvable Galois covering

In Sections 5.2 and 5.3, we applied our attack in the special case of Kummer and Artin-Schreier covers. These cases are especially interesting since they are the elementary blocks of cyclic covers (see [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 definition of Kummer extensions. In fact, let L/K be an order ℓ 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 primitive ℓ-th roots of unity are in $\mathbb{F}_{q^m}$, which is given by the condition $\ell \mid q^m - 1$. As we aim to apply several times the attack described in section 5.2 in our generalization, this condition needs to be satisfied at each iteration.

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, meaning that 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 $n_i \in \mathbb{N}$. More precisely, let $p := \mathrm{char}(\mathbb{F}_{q^m})$ and write $n_i = p^{r_i} m_i$, with $(m_i, \mathrm{char}(\mathbb{F}_{q^m})) = 1$. According to Remark 30, we also suppose that $m_i \mid q^m - 1$ for all $0 \leq i \leq s-1$, meaning that $\mathbb{F}_{q^m}$ contains all the $m_i's$ primitive roots of unity.

Remark 31. The above conditions on roots of unity is not mandatory since we could extend the base field. Nevertheless, considering bigger base field would increase the complexity of our algorithms, as we will see later in Annex B. For simplicity, we will stick with these assumptions for the remaining of this section, even it weakens our generalization.

Let us denote $L_i := L^{\mathcal{G}_i}$ the fixed field by $\mathcal{G}_i$, for all $0 \leq i \leq s$. 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 $\mathcal{G}_{i+1}/\mathcal{G}_i$. In particular, the tower (10) corresponds to a sequence of cyclic covers of $\mathbb{F}_{q^m}$-curves:

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

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 $\mathcal{Y} = \mathcal{X}_0$, say

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

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 $\mathcal{X}_s = \mathcal{X}$. It is then 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 a subcode $\mathcal{C}_i = \mathrm{SSAG}_q(\mathcal{X}_i, \mathcal{Q}_i, G_i)$ of $\mathcal{C}_0$ defined over $\mathcal{X}_i$. Depending on the order $\mathcal{G}_{i+1}/\mathcal{G}_i$, we are led to use either section 5.2 or section 5.3.

More precisely, fix $0 \leq i \leq s-1$ and recall $n_i = p^{r_i} m_i$, with $(m_i, p) = 1$ and $m_i \mid q^m - 1$. 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 and one time the Kummer case 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 $\mathcal{C}_0$.

Remark 32. For every $0 \leq i \leq s-1$, $\mathcal{C}_i := \mathcal{C}_{i+1}^{\mathcal{G}_{i+1}/\mathcal{G}_i}$.

Remark 33. The hypothesis 3 in section 4 needs to be explained a bit in this framework. It says that for any place in the invariant support $\mathcal{Q}_s$, we know which elements in $\mathcal{Q}_0$ are its extensions. From this knowledge, we can extract 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.

Example 34. With previous notations, let $\mathcal{G} = \mathrm{Gal}(L/K)$ be a solvable Galois group of order $\#\mathcal{G} = ps$, with $(p, s) = 1$ and $s \mid q^m - 1$. Suppose we are given a generator matrix of the code $\mathcal{C}_0 = \mathrm{SSAG}_q(\mathcal{Y}, \mathcal{Q}, G)$, as well as the knowledge of its invariant code $\mathcal{C}_2 := \mathrm{SSAG}_q(\mathcal{X}, \mathcal{P}, \tilde{G})$. We 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 $\mathcal{C}_0$ defined over an intermediary curve $\mathcal{X}'$ with function field F, such that both extentions L/F and F/K are cyclic, with respective order s and p (the first one being a Kummer extension, the second an Artin-Schreier one).
2. From the knowledge of the code $\mathcal{C}_1$, we can now apply section 5.2 (possible since L/F as order $s \mid q^m - 1$) in order to recover the secret structure of $\mathcal{C}_0$.

In the next section, we will describe some possible attacks against the invariant code, since we have proved in section 4 that under the correct hypotheses, its knowledge allows us to recover the associated SSAG code.

6 Known attacks against the invariant code

In this section, we focus more on the coding theory point of view. In particular, we will be dealing with McEliece's enryption schemes (see [McE78] for more details), which is a public key cryptosystem whose security rely on a linear code. In our settings, the secret key of the scheme is given by the structure of a quasi-cyclic SSAG code. From the attack of section 4, we know that this security reduces (under some assumptions) to the security of the invariant code.

As a consequence, we could decide to choose the invariant code as the public key while building the scheme, which actually would not affect key sizes. Nevertheless, this would make *message recovery attacks* easier, because solving the *Syndrome Decoding Problem* is faster with smaller parameters. We do not go further into details here, but some solving methods are explaned in the paper of Peters [Pet10], and come from improvments of the attack by *Information-set Decoding*, introduced by Prange in [Pra62]. For the remaining of this section, we will study the security of such schemes.

Keeping usual notations, consider a Galois cover $\mathcal{Y} \longrightarrow X$ of smooth irreductible projective curves over $\mathbb{F}_{q^m}$, and let us denote by

$$\mathcal{C}_{\text{pub}} = \text{SSAG}_q(\mathcal{Y}, \mathcal{Q}, G)$$

the public ℓ-quasi-cyclic SSAG code of a McEliece's scheme. As shown by Corollary 16, the invariant subcode $\mathcal{C}_{\text{inv}}$ is an SSAG code on the quotient curve $\mathcal{X}$, say

$$C_{\text{inv}} := \text{SSAG}_q(\mathcal{X}, \mathcal{P}, \tilde{G}).$$

Note that Remark 8 implies that this code can be constructed in polynomial time from a generator matrix of $\mathcal{C}_{\text{pub}}$ and the action of the induced permutation on it (*i.e.* the public key). This means that a generator matrix $\mathbf{G}_{\text{inv}}$ of $\mathcal{C}_{\text{inv}}$ must also be considered as public data. Below, we describe two possible attacks against SSAG codes.

6.1 Invariant code on the projective line

In the particular case where the quotient curve $\mathcal{X}$ is the projective line $\mathbb{P}^1$, it is possible to construct an algebraic system to recover the secret elements of $\mathcal{C}_{\text{inv}}$ (it is a special case of *algebraic attacks* already mentionned). Combining it with the attack of section 4, we can recover the secret key of the scheme. The main ingredient that allows us to build such a system is the fact that $\mathbb{P}^1$ has genus zero and thus a trivial divisor class group. From Corollary 16, the invariant code is given by

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

To construct an algebric system, we start from the inclusion

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

that is a direct consequence of definitions (in the right-hand side, we have extended the scalars to $\mathbb{F}_{q^m}$ since it is the field definition of the AG code). This means that for every codeword $\mathbf{c} \in C_{\mathcal{L}}(\mathbb{P}^1, \mathcal{P}, \tilde{G})^{\perp}$, we have

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

Moreover, we know from Proposition 5 that there exists a divisor $G' \in \text{Div}(\mathbb{P}^1)$ such that

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

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 support $\mathcal{P}$. Since $\mathbb{P}^1$ has a trivial divisor class group, there exists a function h in the rational function field $\mathbb{F}_{q^m}(x) = \mathbb{F}_{q^m}(\mathbb{P}^1)$ such that

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

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

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

which has dimension $k := \deg(G') + 1$ (consequence of the Riemann-Roch theorem and the fact that the projective line has genus 0).

Let us write the unknown support $\mathcal{P} = (P_1, \cdots, P_r)$, with $P_i := (x_i : 1)$ (here we made the analogy between projective points and rational places). The goal is to recover the $x_i's$, which are the evaluations of the rational function x at the places $P_i's$. For that, we denote by $\mathbf{X} = (X_1, \cdots, X_r)$ and $\mathbf{Z} = (Z_1, \cdots, Z_r)$ two sets of formal variables, respectively corresponding to $x_i's$ and $h(x_i)'s$. From (12), we have

$$\begin{pmatrix} Z_1 & \cdots & Z_r \\ Z_1X_1 & \cdots & Z_rX_r \\ \vdots & \ddots & \vdots \\ Z_rX_r^k & \cdots & Z_rX_r^k \end{pmatrix} \cdot \mathbf{G}_{\text{inv}}^T = 0. \tag{13}$$

The first row provides $k = \dim_{\mathbb{F}_q}(\mathcal{C}_{\text{inv}})$ linear equations in the variables $\mathbf{Z}$, and since $k < r$, we 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 $r-2$ variables $\mathbf{X}$ and $r-k$ variables $\mathbf{Z}$.
If we are able to solve it, we recover the function h (and thus the invariant divisor $\tilde{G}$) as well as the support $\mathcal{P}$; *i.e.* the full structure of the invariant code. Since the security of the whole system rely on it, we have broke the scheme.

Remark 35. The cost of solving (13) is hard to estimate. We can have an upper bound on it if the system has a specific form, which can be useful to study the security of schemes using SSAG codes over the line. Theses results can be found in [FOP+16a], [FOP+16b] and [FOPT10].

The above discussion shows that 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 is no advantages to use it. In particular, this means that the automorphism should be chosen such that the quotient curve is not rational. In the latter case, the fixed field has a more complex divisor class group and the above attack does not work. In the next section, we will describe the cost of an exhaustive search on the invariant code, depending on different parameters of the fixed field.

6.2 Brute force on the invariant code

Here, we keep the previous notations and we consider the invariant code over the quotient curve $\mathcal{X}$ given by

$$C_{\text{inv}} := \text{SSAG}_q(\mathcal{X}, \mathcal{P}, \tilde{G}).$$

A brute force attack on it consists in recovering its secret structure from one of its generator matrix. It is devided inti three steps:

1. Enumerate all the possible divisor classes of a given degree on the quotient curve $\mathcal{X}$;
2. Guess the good divisor $\tilde{G}$ in the class;
3. Guess the support $\mathcal{P}$ of length $r := n/\ell$, where ℓ is the quasi-cyclicity order and n the length of the public code.

We start by discussing the third step, which is the easiest to formalize.

Recovering the invariant support. Here we assume that the first two 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 $S \subseteq \mathcal{X}(\mathbb{F}_{q^m})$ of length r, then we recover the good permutation using the SSA algorithm (see [Sen00] for more details).

The second is to solve a linear system as in section 6.1. In order to build it, we start by recalling (Proposition 5) that there exists a divisor $G' \in \text{Div}(\mathcal{X})$ such that

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

The main difference with the attack of section 6.1 is that we do not know the curve $\mathcal{X}$ from our hypotheses. Nevertheless, let us suppose that the attacker found a way to recover it, meaning that it is possible to find a basis of the Riemann-Roch space $\mathcal{L}_K(G') \subseteq K := \mathbb{F}_{q^m}(\mathcal{X})$, *i.e.* we have $\mathcal{L}_K(G') = \langle f_1, ..., f_r \rangle_{\mathbb{F}_{q^m}}$, where $r = \dim(G')$. If $\mathcal{P} := \{P_1, \cdots, P_r\}$, we get

$$\forall\ 1 \leq i \leq r\ ,\ (f_i(P_1), \cdots, f_i(P_r)) \cdot \mathbf{G}_{\text{inv}}^T = 0.$$

From this point, we can build a linear system whose unknows are the elements of $\mathcal{P}$. Note however that this system is even harder to solve than (13), since we can not estimate the form of the rational functions $f_i's$. Even if the system is polynomial, the complexity of solving it using Gröbner bases depends on the form and degree of the polynomials and is thus difficult to forecast.

Enumeration of divisor classes. We start by explaining why it is not necessary to enumerate all divisors in 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. To precise this, we introduce the notion of diagonal equivalent codes.

Definition 36. 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 $\lambda_1, ..., \lambda_n$ in $\mathbb{F}_{q^m}$ 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 checked 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

$$\mathbf{G}_{\mathcal{C}_1} \cdot \begin{pmatrix} W_1 & \cdots & 0 \\ & \ddots & \\ 0 & \cdots & W_n \end{pmatrix} \cdot \mathbf{H}_{\mathcal{C}_2}^T = 0. \tag{14}$$

By Definition 36, this system has at least one solution if and only if $\mathcal{C}_1 \sim_{\text{diag}} \mathcal{C}_2$, which provides an easy way to check whever two codes are diagonal equivalent or not.
This property leads to a smarter brute force search on the divisor in the case of SSAG codes. We first deal with the case of AG codes as they are easier to treat.

Theorem 37 ([MP93], Corollary 4.15.)**.** *Let $\mathcal{X}$ be a smooth irreducible projective curve of genus $g(\mathcal{X})$ and $\mathcal{P}$ be a set of $n > 2g(\mathcal{X}) - 2$ rational places on $\mathcal{X}$. If $G, H \in \text{Div}(\mathcal{X})$ are two divisors of same degree r such that $2g(\mathcal{X}) - 1 < r < n - 1$, then*

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

Let us denote by $\text{AG}_r(\mathcal{X}, \mathcal{P})$ the set of AG codes on $\mathcal{X}$, defined by a fixed support $\mathcal{P}$ and any divisor of degree r. The above theorem implies

Corollary 38. *Let $\mathcal{P} \subseteq \mathcal{X}(\mathbb{F}_{q^m})$ be a support of lenght $n > 2g(\mathcal{X}) + 2$ and $r \in \mathbb{N}$ be such that $2g(\mathcal{X}) - 1 < r < n - 1$. 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.

The above estimation is sufficient if we want to perform a brute force search on an AG code. We could proceed as follows:

1. Perform a brute force search among divisor classes of degree r, *i.e.* choose a representative divisor G' in the class.
2. Guess a support $\mathcal{P}'$ (see above) and attempt to solve the system (14) to check whether $C_{\mathcal{L}}(\mathcal{X}, \mathcal{P}', G')$ is diagonal equivalent to the public code or not.
3. If (14) has a non zero solution $(\lambda_1, ..., \lambda_n) \in (\mathbb{F}_{q^m}^*)^n$, then we have recovered the public code $(\lambda_1, ..., \lambda_n) \star C_{\mathcal{L}}(\mathcal{X}, \mathcal{P}', G')$.

Both Theorem 37 and Corollary 38 provide an estimation of the cost of a brute force search amoung divisor classes in the case of AG codes. Unfortunately, theses results do not hold with SSAG codes. In the latter case, the situation is more complicated but we can have a first estimation by using the following proposition.

Proposition 39 ([MP93], Corollary 7.4)**.** *With notations of Corollary 38, if $n > 2g(\mathcal{X}) + 2$ and $2g(\mathcal{X}) - 1 < r < n$, then*

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

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

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

We can decrease the previous bound thanks to the following remark.

Remark 40. 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 over $\mathbb{F}_q$ are equal, i.e.

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

This leads to the final upper bound

$$\#\,\mathrm{SSAG}_{q,r}(\mathcal{X},\mathcal{P}) \leq \frac{(q^m-1)^{n-1}}{(q-1)^{n-1}}h(\mathcal{X}). \tag{15}$$

In Annex B, we will formally describe a brute search algorithm on the invariant SSAG code and study its complexity, which is clearly bounded by $h(\mathcal{X})$. In particular, this will lead to choose optimally the automorphism $\sigma \in \mathrm{Aut}(\mathcal{Y})$ in order this number to be high enough.

In the next section, we propose a McEliece scheme based on a quasi-cyclic SSAG code defined over the *Hermitian curve*, as well as a set of parameters to secure the corresponding invariant code against the attacks discussed above.

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

7.1 The proposed scheme

We recall that q is the power of a prime p, and that we work with the finite field $\mathbb{F}_{q^m}$. Since the Hermitian curve is defined over a field with square cardinality, we assume that $q^m = p^{2s}$ is a square. To avoid missunderstanding with the field $\mathbb{F}_q$ where the subfield subcode is defined, we introduce an intermediary field $\mathbb{F}_{q_0}$ with $q_0 = p^s$ elements. We define the Hermitian curve $\mathcal{H}$ over the $\mathbb{F}_{q^m} = \mathbb{F}_{q_0^2}$ by $\mathbb{F}_{q^m}(\mathcal{H}) := \mathbb{F}_{q_0^2}(x,y)$,with

$$y^{q_0} + y = x^{q_0+1}. \tag{16}$$

The idea is to construct a McEliece scheme based on an SSAG code defined over $\mathcal{H}$, stable under the action of some cyclic automorphism. There are two motivations to use this curve: first, it is a maximal curve, meaning that it atteins the Hasse-Weil bound. More precisely, this curve has genus $g(\mathcal{H}) = \frac{q_0(q_0-1)}{2}$ and $N(\mathcal{H}) = q_0^3+1$ rational points. It allows us to consider longs codes, and thus more flexibility. Moreover, the automorphism group $\mathrm{Aut}(\mathcal{H})$ is very large and has been well-studied (see [Sti09] or [GSX00]); which gives us a large choice for our automorphism σ (it will be important for the security of the scheme, see section 7.2.2).

From now on, we fix the following notations:

- $\sigma \in \mathrm{Aut}(\mathcal{H})$ is an automorphism of order $\ell \in \mathbb{N}^*$ (which will be chosen later);
- $\mathcal{Q} := \bigsqcup_{i=1}^{r} \mathrm{Orb}_\sigma(Q_i)$ is a support made of r distincts orbits under the action of σ, and $n := \ell r$ refers to the lenght of the public code;
- $G = \sum_{i=1}^{s} t_i \sum_{R\in\mathrm{Orb}_\sigma(R_i)} R$ is a σ-invariant divisor, with $R_i \in \mathbb{P}_\mathcal{H}$ and $t_i \in \mathbb{Z}$.

Finally, we suppose as usual that $\mathrm{Supp}(G)\cap\mathcal{Q} = \emptyset$. We can now describe our scheme:

Key generation. We consider the quasi-cyclic code

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

constructed on the Hermitian curve $\mathcal{H}$, with length $n \leq N(\mathcal{H})$ 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 ℓ-blocks-circulant matrix, *i.e.* 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

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

where M_i is 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 $\mathcal{Q}$ 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 $\omega(\mathbf{e}) \leq t$ (ω being the Hamming weight).

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

The security of the scheme is based on the support $\mathcal{Q}$ and the divisor G which are unknown. However, we know from Corollary 16 that the invariant code is an SSAG code on the quotient curve $\mathcal{H}/\langle\sigma\rangle$. Recall that the attack proposed in section 4 shows that the security reduces to the security of the invariant code. To secure our scheme, we need to ensure that it can not be recovered easily. In particular, we propose in the next section a set of parameters to be out of range of the attacks described in section 6.

7.2 Suggested parameters

7.2.1 Choice of the quasi-cyclicity order ℓ

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

1. ℓ **shoud be prime**. In fact, if there exists a prime $s \mid \ell$, then the power permutation σ^s also acts on the public code. In particular, $\mathcal{C}_{\text{pub}}$ is also s-quasi-cyclic, and thus we can construct another invariant code, *i.e.* $\mathcal{C}_{\text{pub}}^{\sigma^s}$. Since this is possible for every divisor of ℓ, we are able to construct several SSAG-subcodes of the public code. Actually, we don't know if it makes the attack easier, but it is clear that every intermediary code could provides informations about the secret support and divisor. As we want to give the least information possible, ℓ will be chosen as a prime.
2. ℓ **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 exists another intermediary code that can be constructed from the knowledge of a generator matrix and the induced permutation, namely the *folded code* (see [FOP+16a]). It is defined as the image of $\mathcal{C}_{\text{pub}}$ by the map $\text{id} + \sigma + ... + \sigma^{\ell-1}$. 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 by computing the image by the map $P(\sigma)$, where P is a divisor of $1 + z + z^2 + ... + z^{\ell-1}$. Again, we get another subcode that has a special structure related to the secrets of the scheme. We don't know if the knowledge of several folded codes simplify an attack, but it is a safer choice to make this assumption.

Remark 41. From its definition, the folded code is σ-invariant, meaning that is is a subcode of $\mathcal{C}_{\text{pub}}^{\sigma}$. If $p \nmid \ell$, then we can show that these two codes are equal (see [Bar17] , Lemma 3.2). Hence, if ℓ satisfies both the above conditions and if $p \nmid \ell$, then the only subcode of $\mathcal{C}_{\text{pub}}$ that an attacker can construct is the invariant code. In particular, we can focus on its security without having to worry about possible other subcodes.

7.2.2 Choice of the automorphism σ

The complexity of the brute force attack on $\mathcal{C}_{\text{inv}}$, described in 6.2, depends on the class number of the quotient curve $h(\mathcal{H}/\langle\sigma\rangle)$. In particular, from Theorem 2, this number can be estimated using the genus $g(\mathcal{H}/\langle\sigma\rangle)$. In this section, we detail how to construct an automorphism $\sigma \in \text{Aut}(\mathcal{H})$ of order ℓ, with ℓ satisfying both conditions of section 7.2.1. 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 curves.

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

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

We then introduce the subgroup

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

consisting in all automorphisms fixing the unique point at infinity Q_∞ in $\mathcal{H}(\mathbb{F}_{q_0^2})$. If $\mathbb{F}_{q_0^2}(x,y)$ is the function field of $\mathcal{H}$, defined in (16), then any automorphism in $\mathcal{A}(Q_\infty)$ acts as follows:

$$\begin{cases} \sigma(x) = ax + b \\ \sigma(y) = a^{q_0+1}y + ab^{q_0}x + c, \end{cases} \tag{17}$$

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 also have

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

According to (17), any element $\sigma \in \mathcal{A}(Q_\infty)$ can be identified to a triple (a,b,c). For convenience, it will be denoted by $\sigma = [a,b,c]$. Next we show that the order of such an automorphism depends only on the order of a and the choice of c.

Lemma 42. *(see [GSX00], Lemma 4.1.) Let $\sigma = [a,b,c] \in \mathcal{A}(Q_\infty)$, with $a \neq 1$. 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

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

Now, let ℓ be an integer satisfying both conditions in 7.2.1, and such that $\ell \mid q_0^2-1$. We randomly pick $a \in \mathbb{F}_{q_0^2}^*$ of order ℓ and $b \in \mathbb{F}_{q_0^2}$. If $\ell \mid q_0+1$, we set $c := \frac{ab^{q_0+1}}{a-1}$, else we choose any c amoung the roots of $X^{q_0} + X - b^{q_0+1}$. From Lemma 42, the automorphism $\sigma = [a,b,c]$ has order ℓ.

The genus of $\mathcal{H}/\langle\sigma\rangle$. The authors in [GSX00] provide a genus formula to compute $g(\mathcal{H}/\langle\sigma\rangle)$ when $\sigma \in \mathcal{A}(Q_\infty)$.

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

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

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

Proof. see [GSX00], Theorem 4.4. □

Notice that since we want the quotient curve to have positive genus, ℓ should be strictly less than q_0+1. Using this proposition and Theorem 2, we can estimate the class number $h(\mathcal{H}/\langle\sigma\rangle)$:

Corollary 44. *Let $\sigma = [a, b, c] \in \mathcal{A}(Q_\infty)$ be an automorphism of prime order $\ell > 2$.*

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

As we have seen in section 6.2, the complexity of the brute force attack on the invariant code is upper bounded by the class number $h(\mathcal{H}/\langle\sigma\rangle)$ (see (15)). In 7.2.4, we propose some parameters for our scheme such that this class number is large enough to reach the desired security.

7.2.3 Choice of the base field

To provide SSAG codes over the base field $\mathbb{F}_q$, we have to choose an extension $\mathbb{F}_{q^m}$ of $\mathbb{F}_q$ such that q^m is a square. Let us discuss the choice of q,m and 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 $n - m(n - \dim(G))$ (consequence of Delsarte's theorem). As a result, if m is too big, the rate (*i.e.* k/n) of the code might be too low, which is not desired in practice.
- The same remark holds for the choice of q_0: since $g(\mathcal{H}) = \frac{q_0(q_0-1)}{2}$, the dimension k of the SSAG code satysfies
$$k \geq n - m(n - \dim(G)) = n - m(n - \deg(G) + g(\mathcal{H}) - 1),$$
so q_0 should not be too large as well.
- On the other hand, the choice $q = q_0$ and $m = 2$, which satisfies both the previous points, is not encouraged. The formal argument in this direction is that the smaller the extension degree m is, the closer the structure of the SSAG code is from the AG one. Since the latter have been broken in polynomial time (see [CMCP17]), it might be possible to addapt this in the subfield subcode scenario. In the upcoming section, we still provide some parameters with $m = 2$ because it results in the best key sizes, but we warn the reader that it could be the weakest keys.

7.2.4 Parameters

Let us fix the notations used 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 $q_0^2 = q^m$.
- n and k are respectively the length and the dimension of the public code.
- ℓ is the quasi-cyclicity order (*i.e.* the order of the automorphism σ).
- $g(\mathcal{H}/\langle\sigma\rangle)$ and $h(\mathcal{H}/\langle\sigma\rangle)$ are respectively the genus and the class number of the quotient curve.
- Key sizes are given in bytes, and computed with the formula $\left\lceil \frac{\log_2(q) \cdot \frac{k}{l} \cdot (n-k)}{8} \right\rceil$, since public keys are systematic and quasi-cyclic generator matrices of $[n, k]$-codes.
- ω_{ISD} is the $\log_2$ of the work factor for the *Information-set Decoding* (ISD) attack, computed using **CaWoF** library (see. [CT16]).

The desired security is 128 bits, and we keep in mind that the class number $h(\mathcal{H}/\langle\sigma\rangle)$ is an upper bound for the brute force attack.

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

m	q	q_0	n	k	ℓ	Key sizes (bytes)	ω_{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, $m > 2$

m	q	n	k	ℓ	Key sizes (bytes)	ω_{ISD}	$g(\mathcal{H}/\langle\sigma\rangle)$	$h(\mathcal{H}/\langle\sigma\rangle)$	$\#\,\text{SSAG}_{q,r}(\mathcal{H},\mathcal{P})$
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 section, we present a formal algorithm that describes the attack proposed in section 5.2 in the special case of a Kummer covering of the projective line $\mathbb{P}^1$. Recall that we are led to solve linear systems of the form :

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

which consist in $s(n-k)+n$ equations and n unknows, where $s = \dim(D)$ (see section 4 for the definition of the divisor D), n and k are respectively the lenght and dimension of the public SSAG code and $\xi \in \mu_\ell^*(\mathbb{F}_{q^m})$. Suppose the function field $\mathbb{F}_{q^m}(x,y)$ of our Kummer curve is defined by

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

with $f \in \mathbb{F}_{q^m}[T]$ and $d := \deg(f)$ (note that d is both the degree of the pole divisor of f and its degree as a polynomial). In this framework, we describe in Algorithm 1 the attack of section 5.2 to recover the polynomial f, as well as the secret structure of the public SSAG code.

Proposition 45. *Let n, k be the lenght and the dimension of the public* SSAG*-code respectively. 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 details about complexity analysis.
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 ω 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 $\mathbb{F}_{q^m}$ (operations need to be done in $\mathbb{F}_{q^m}$ and not $\mathbb{F}_q$ since the roots of unity might be in $\mathbb{F}_{q^m}\backslash\mathbb{F}_q$). Since we have to seek for the correct root of unity ξ, this step might be repeated at most $\varphi(\ell)$-times, where φ is the Euler totient function. Next, we have to realize one Lagrange's interpolation at line 13 in order to recover a defining equation of the Kummer cover. Let $d := \deg(f)$ and note that 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 15. 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 $\text{Supp}(G)$. 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

Algorithm 1: Security reduction in Kummer case

Input : A parity check matrix $\mathbf{H}_{\text{pub}}$ of the public code, the invariant support $\tilde{P} = \{P_1, \cdots, P_r\}$ and divisor $\tilde{G}$. Also we are given the quasi-cyclicity order $\ell > 0$ and the integer $d := \deg(f)$.

Output: The polynomial f and the secret structure $(\mathcal{P}, G)$.

$x \longleftarrow (\underbrace{x(P_1), ..., x(P_1)}_{\ell-\text{fois}}, ..., x(P_r), ..., x(P_r))$

$D \longleftarrow \tilde{G} - \lceil d/\ell \rceil \cdot P_\infty \in \text{Div}(\mathbb{P}^1)$

$s \longleftarrow \dim(D)$

$M \longleftarrow$ set of primitive ℓ-th roots of unity

cpt := 0

while cpt = 0 **do**

  $\xi \xleftarrow{\$} M$

  Exclude(M,ξ)

  $S \longleftarrow \text{Solve}(\Delta(\xi))$

  if $\dim(S) = 1$ **then**

    cpt := 1

    $y^* \xleftarrow{\$} S \backslash \{0\}$

    $f \longleftarrow \text{Interpolate}(x, y^*)$

    $\mathcal{P} \longleftarrow \{P_{ij} = (x_{ij} : y_{ij} : 1)\}$

$G \longleftarrow \pi^*(\tilde{G})$

return f, $\mathcal{P}$ and G

point $P \in \text{Supp}(\tilde{G})$, then the y-coordinates of the extensions of P in $\text{Supp}(G)$ are exactly the roots of the polynomial $T^\ell - f(x(Q)) \in \mathbb{F}_{q^m}[T]$. This step can be done by factorizing each polynomial 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 (*i.e.* $n > q^m$) 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 Algorithm 1 can also be used in the case of an Artin-Schreier cover of $\mathbb{P}^1$, by only changing a few lines. In fact, in the latter setup, we have to solve at most $p = \#(\mathbb{F}_p)$ linear system of the form (8), whose total cost is in $\mathcal{O}(p(n^\omega + n^{\omega-1}s(n-k))$ operations over $\mathbb{F}_{q^m}$.

Remark 46. In Algorithm 1, we describe our attack in the special case were the quotient curve is the projective line $\mathbb{P}^1$, as it is easier to produce a complexity analysis. However, it can be generalized to any Kummer or Artin-Schreier cover, the only difference being the interpolation step at the end. In fact, in the more general case, the polynomial we have to recover is bivariate, thus the interpolation step might be harder. More precisely, we can use a bivariate Lagrange's interpolation that requires more evaluation points, *i.e.* 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 polynomials instead of rational functions. We could actually use interpolation techniques in the function field of a curve insted of using it on polynomials, but this theory is not well-developed yet, which is the reason why we only work with polynomials.

Remark 47. In section 5.4, we generalized our attack in the case of a solvable Galois covering, by applying several times the Kummer or Artin-Schreier case. The total cost of the attack in this context corresponds to the number of iterations we have to use Algorithm 1. In particular, if our solvable Galois group has order $p^s\lambda$, with $p = \text{char}(\mathbb{F}_{q^m})$ and $(p, \lambda) = 1$, the attack consists in $s+1$-iterations of Algorithm 1. Do not forget that this is possible if and only if the conditions on primitives roots of unity is satisfied. If not, it is possible to extend the base field, say $\mathbb{F}_{q^m} \supseteq \mathbb{F}_{q^m}$. Doing so, so complexity of the algorithm is increased since we now have to compte in a bigger field.

B Brute force Algorithm on the invariant code

In section 6.2, we described a brute force attack against SSAG codes, which is detailled in Algorithm 2 below. As for the notations, we consider an SSAG code of length n defined over a curve $\mathcal{X}$, associated to a degree $r \geq 0$ divisor. We denote by $\mathrm{Cl}^r(\mathcal{X})$ the group of divisor class of degree r in $\mathcal{X}$, and for $G \in \mathrm{Div}(\mathcal{X})$, $[G]$ represents its class in $\mathrm{Cl}^r(\mathcal{X})$. The idea is to recover the good permutation amoung all subsets of lenght n, and then use the SSA algorithm (see [Sen00]). The divisor is searched amoung all classes of divisor of the given degree, using (15).

Algorithm 2: Brute force on SSAG

Input : A generator matrix $\mathbf{G}$ of the invariant SSAG code $\mathcal{C}$ and an integer r which refers to the degree of the divisor.

Output: The support $\mathcal{P}$ and the divisor G such that $\mathcal{C} = \mathrm{SSAG}_q(\mathcal{X}, \mathcal{P}, G)$.

$S \leftarrow \{P \in \mathcal{X} \mid \deg(P) = 1\}$ // The order of S is fixed
for $[G] \in \mathrm{Cl}^r(\mathcal{X})$ **do**
  $\mathcal{C}' \leftarrow \mathrm{SSAG}_q(\mathcal{X}, S, G)$ // G will be a representative of $[G]$
  for $[\mathbf{w}] \in \mathbb{F}_{q^m}^n / \mathbb{F}_q^n$ **do**
    /* $\mathbf{w}$ will be a representative of $[\mathbf{w}]$ */
    for $\mathcal{I} \subseteq \{1, \ldots, N(\mathcal{X}) - 1\}$ with $|\mathcal{I}| = n$ **do**
      $\mathcal{C}'_{\mathcal{I}} \leftarrow \mathrm{Punct}_{\mathcal{I}}(\mathcal{C}')$
      $\pi \leftarrow \mathrm{SSA}(\mathbf{w} \star \mathcal{C}'_{\mathcal{I}}, \mathcal{C})$ // SSA return a permutation π or '?'
      if $\pi \in \mathfrak{S}_n$ **then**
        $S_I \leftarrow \{P_i \in S \mid i \in \mathcal{I}\}$
        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 $\mathbf{w}$. For our proposal on the Hermitian curve and using (15), the algorithm requires at least

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

By Theorem 2, we know that

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

which leads to $h(\mathcal{H}/\langle\sigma\rangle) \in \mathcal{O}(q^{mg(\mathcal{H}/\langle\sigma\rangle)})$. As a result, the cost of the enumeration of divisors G in $\mathrm{Cl}^r(\mathcal{H}/\langle\sigma\rangle)$ and vectors in $\mathbb{F}_{q^m}/\mathbb{F}_q$ is in $\mathcal{O}(q^{m(n-1)+mg(\mathcal{H}/\langle\sigma\rangle)})$ operations over $\mathbb{F}_q$.
Finally, Proposition 43 and Corollary 44 allow us to choose the paramerters (see section 7.2.4) 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.

[Bar18a] Elise Barelli. *On the security of short McEliece keys from algebraic and algebraic geometry codes with automorphisms*. PhD thesis, universite Paris-Saclay, 2018.

[Bar18b] Elise Barelli. Short McEliece keys from Hermitian curves. *Manuscript*, 2018.

[BBB+17] Magali Bardet, Elise Barelli, Olivier Blazy, Rodolfo Canto-Torres, Alain Couvreur, Philippe Gaborit, Ayoub Otmani, Nicolas Sendrier, and Jean-Pierre Tillich. Big quake. *NIST Round 1 submission for Post-Quantum Cryptography*, 2017.

[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**, no.8:5404–5418, 2017.

[CT16] Rodolfo Cantos-Torres. CaWoF, C library for computing asymptotic exponents of generic decoding work factors. no.8:5404–5418, 2016.

[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(1):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(1):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*, vol. 6110:279–298, • 2010. •.

[Ful69] William Fulton. *Algebraic Curves*. Benjamin, 1969.

[GSX00] Arnaldo Garci, Henning Stichtenoch, and Chao-Ping Xing. On subfields of the Hermitian function field. *Compositio Mathematica* **120**, no.2:137–170, 2000.

[Har77] Robin Hartshorne. *Algebraic geometry*, volume 52 of *Graduate Texts in Mathematics*. Springer, 1977.

[HP95] Tom Høholft and Ruud Pellikaan. On the decoding of algebraic-geometric codes. *IEEE Transactions on Information Theory* **41**, no.6:1589–1614, 1995.

[Liu02] Qing Liu. *Algebraic geometry and arithmetic curves*. Oxford, 2002.

[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**, no.3:229–252, 1993.

[Pet10] Christiane Peters. Information-set decoding for linear codes over $\mathbf{F}_q$. In *Post-Quantum Cryptography 2010*, volume 6061 of *LNCS*, pages 81–94. Springer, 2010.

[Pra62] Eugene Prange. The use of information sets in decoding cyclic codes. *IRE Transactions on Information Theory* ***8***, no.5:5–9, 1962.

[Sen00] Nicolas Sendrier. Finding the permutation between equivalent linear codes: The support splitting algorithm. *IEEE Transactions on Information Theory* **46**, no.4:1193–1203, 2000.

[Sen11] Nicolas Sendrier. Decoding one out of many. In *Post-Quantum Cryptography 2011*, volume 7071 of *LNCS*, pages 51–67. Springer, 2011.

[Sti09] Henning Stichtenoch. *Algebraic Function Fields and Codes*, volume 254 of *Graduate Texts in Mathematics*. Springer, 2nd edition, 2009.

[TVN07] Michael A. Tsfasman, Serge G. Vladut, and Dmitry Nogin. *Algebraic geometric codes: basic notions*. no.139, American Mathematical Soc., 2007.