Attending Sudan's decoding radius with no genus penalty for algebraic geometry codes

Isabella Panaccione

Attending Sudan's decoding radius with no genus penalty for algebraic geometry codes

Isabella Panaccione*[1,2]

[1]INRIA
[2] LIX, CNRS UMR 7161
École Polytechnique,
91128 Palaiseau Cedex, France

March 20, 2021

Abstract

In this paper we present a decoding algorithm for algebraic geometry codes with error–correcting capacity beyond half the designed distance of the code. This algorithm comes as a fusion of the Power Error Locating Pairs algorithm for algebraic geometry codes and the technique used by Ehrhard in order to correct these codes up to half the designed distance. The decoding radius of this algorithm reaches that of Sudan algorithm, without any penalty given by the genus of the curve.

Key words : Error correcting codes; algebraic geometry codes; decoding algorithms; error correcting pairs; Sudan algorithm; genus.

Introduction

Algebraic geometry codes were first introduced by Goppa in 1981 [Gop81] and gave a breakthrough in coding theory when Tsafsman, Vlădut and Zink proved that Gilbert Varshamov bound could be exceeded when some specific curves where considered [TVZ82]. Furthermore, these codes have interested the Cryptography scene too, in particular for McEliece scheme [JM96].

Unique decoding of algebraic geometry codes

Thanks to their strong algebraic structure, it has been possible to design several decoding algorithms for algebraic geometry codes. In 1989 Justesen, Larsen, Jensen, Havemose and Høholdt proposed one of the first decoding algorithms for a specific class of algebraic geometry codes [JLJ$^+$89] achieving the correction capacity of $\frac{d^*-1-g}{2}$, where d^* is the designed distance of the code and g the genus of the curve. This algorithm, also called *basic algorithm* in the literature, is the starting point for several years of research aimed at improving this decoding radius by erasing the penalty in the genus of the curve. One of the first attempts in this sense, came from Skorobatov and Vlădut [SV90] who generalised the basic algorithm to arbitrary curves and improved the decoding radius in some cases. Their result was in turn improved by Duursma [Duu93] who generalised it to all algebraic geometry codes and reached the decoding radius $\frac{d^*-1}{2} - \sigma$, where σ is the *Clifford defect*. The problem though was not complitely solved, as for instance for plane curves we have in average $\sigma = \frac{g}{4}$. This last algorithm is also referred to as the *modified algorithm*.

In parallel to the basic algorithm and with the same correction capacity, we have the algorithm proposed by Porter in his thesis [Por88]. Porter's idea mainly consists in solving a *key equation*, using a generalisation of Euclide's algorithm for functions on curves. Though, the price of this generalisation lies in strong

*isabella.panaccione@inria.fr

restrictions on the codes and the curves, which entail the correctness of the algorithm only for a small class of codes. In [Ehr92], Ehrhard generalised this algorithm to all curves by solving the key equation of Porter's algorithm with simple linear algebra operations and proved this algorithm to be equivalent to the basic algorithm for a divisor F with no evaluation points in its support. The correctness of the algorithm was proved independently as well by Porter, Shen and Pellikaan in [PSP92], where in addition they succeded in pushing the decoding radius up to the one of the modified algorithm.

The first algorithm able to correct up to $\frac{d^*-1}{2}$ has been proposed by Pellikaan in [Pel89]. This algorithm, whose correctness is ensured for maximal curves and some other cases, consists in running the basic algorithm in parallel on several divisors $F_1, \ldots, F_s$, as for counting reasons one among them has to work. Though, this result only ensures the existence of these divisors and even if it was extended to almost all curves (Vlădut [Vlă90]), no practical precedure to find the F_i's has been found yet. A constructive algorithm to achieve the correction capacity $\frac{d^*-1}{2}$ was finally found by Ehrhard in [Ehr93], whose idea consists in providing a more suitable divisor F, obtained from a gradual adaptation process, and running [Ehr92] on this F. In the same year Feng and Rao proposed the so called *majority vote* for *unknown syndromes* [FR93], which also corrects $\frac{d^*-1}{2}$ errors.

In 1992 Pellikaan [Pel92] and, independently, Köetter [Köt92] introduced the so called *error correcting pairs* algorithm, which generalises the basic algorithm to all codes which dispose from a particular structure called *error correcting pair*. This algorithm cannot correct more than $\frac{d-1}{2}$ errors and in particular, since for algebraic geometry codes it reduces to the basic algorithm, for this class of codes it is equivalent to Ehrhard algorithm [Ehr92] for a divisor F with no evaluation points in its support and corrects up to $\frac{d^*-1-g}{2}$ errors.

Beyond half the designed distance

It is known that several decoding algorithms have been extended from Reed–Solomon codes to algebraic geometry codes to correct amounts of errors superior than half the designed distance. Though, as for the basic algorithm, anytime such a generalisation is made, a penalty in the genus of the curve appears in the decoding radius. Sudan algorithm [Sud97] has been extended to algebraic geometry codes by Shokrollahi Wasserman [SW99] with a penalty of $\frac{\ell g}{\ell+1}$, where ℓ is the degree of Sudan's polynomial. It is known that Sudan algorithm gives in return the list of all possible solutions to the decoding problem. Though it is possible to generalise to algebraic geometry codes also algorithms like the power decoding ([SSB10], revisited in [RnN15]), which gives back the closer solution (if it exists) or fails. Also the version of the error correcting pairs algorithm to correct more errors, which is the *power error locating pairs* algorithm [CP20] can be run on algebraic geometry codes with a penalty in the genus $\frac{\ell g}{\ell+1}$, where the paramter ℓ is the *power* used in the algorithm. These three algorithms have in practice the same decoding radius (at most they differ by 1) and in particular they share the penalty in the genus. Now, it is known it is possible to get an improved decoding radius with no genus penalty using Guruswami–Sudan algorithm for an appropriate multiplicity. However, since the size of the linear system of the algorithm depends on this multiplicity, the complexity of the algorithm can become quite large very fast.

Our contribution

In this paper we propose a decoding algorithm for algebraic geometry codes, whose decoding radius turns to equal the one of Sudan algorithm, but with no factor in g. To do so, first we adapt the language of power error locating pairs algorithm [CP20] to the one of Ehrhard's result [Ehr93]. In this way we get a decoding algorithm with a correction capacity equivalent to Sudan's. Therefore, to erase the penalty, we apply our algorithm to a suitable divisor F provided by using Ehrhard's adaptation process ([Ehr93]).

Outline of the article

In §1 we will give some notations and results about Riemann–Roch spaces, algebraic geometry codes and star product. In §2, also introductive, we will give an overview of Ehrhard's algorithm [Ehr93] and show it as an extension of [Ehr92] (see Remark 2.7). In §3 and §4 the new algorithm is presented, while some experimental observations are given in §5. The nature of these results being quite technical, we decided to report some proofs in appendix and suggest the reader to postpone their reading to a second moment.

1 Notation and preliminaries

1.1 Codes and decoding problems

Given a finite field $\mathbb{F}_q$, a code over $\mathbb{F}_q$ of length n is simply a subset of $\mathbb{F}_q^n$. The code is said to be *linear* if it is a vector subspace of $\mathbb{F}_q^n$. Its elements are called *codewords*.

Definition 1.1. Given two vectors $\boldsymbol{a} = (a_1, \ldots, a_n), \boldsymbol{b} = (b_1, \ldots, b_n) \in \mathbb{F}_q^n$, their *Hamming distance* is

$$\mathrm{d}(\boldsymbol{a}, \boldsymbol{b}) \stackrel{\text{def}}{=} \#\{i \in \{1, \ldots, n\} \mid a_i \neq b_i\}.$$

The weight of $\boldsymbol{a}$ is defined as $\mathrm{w}(\boldsymbol{a}) \stackrel{\text{def}}{=} \mathrm{d}(\boldsymbol{a}, \boldsymbol{0})$.

Definition 1.2. Given a code $\mathcal{C} \subseteq \mathbb{F}_q^n$, the *minimum distance* of $\mathcal{C}$ is

$$\mathrm{d}(\mathcal{C}) \stackrel{\text{def}}{=} \min_{\substack{\boldsymbol{a}, \boldsymbol{b} \in \mathcal{C} \\ \boldsymbol{a} \neq \boldsymbol{b}}} \mathrm{d}(\boldsymbol{a}, \boldsymbol{b}).$$

In the rest of the paper we will write sometimes d instead of $\mathrm{d}(\mathcal{C})$ when there is no ambiguity on the code. We recall that if the code $\mathcal{C}$ is linear, which is the case for the codes considered in this paper, we have $\mathrm{d}(\mathcal{C}) = \min_{\boldsymbol{a} \in \mathcal{C} \setminus \{\boldsymbol{0}\}} \mathrm{w}(\boldsymbol{a})$.

Definition 1.3. Given a vector $\boldsymbol{a} = (a_1, \ldots, a_n) \in \mathbb{F}_q^n$ we define its *support* as

$$\mathrm{supp}(\boldsymbol{a}) \stackrel{\text{def}}{=} \{i \in \{1, \ldots, n\} \mid a_i \neq 0\}.$$

We can now present the decoding problem we want to solve.

Problem 1. Let $\mathcal{C} \subseteq \mathbb{F}_q^n$ be a code, $\boldsymbol{y} \in \mathbb{F}_q^n$ and $t \in \{1, \ldots, n\}$. Return (if it exists) $\boldsymbol{c} \in \mathcal{C}$ such that

$$\mathrm{d}(\boldsymbol{y}, \boldsymbol{c}) \leq t.$$

This problem takes the name of *bounded decoding problem*. A *decoding algorithm* for a code $\mathcal{C}$, is an algorithm which solves Problem 1 for $\mathcal{C}$ for some t. The maximum value of t for which the algorithm can solve this problem, is called *decoding radius* of the algorithm. Depending on the value of t, it is possible to estimate the number of possible solutions to Problem 1. In particular, it is known that if $t \leq \frac{\mathrm{d}(\mathcal{C})-1}{2}$, then there exists at most one solution. Once t exceeds this amount, called *unique decoding bound*, the uniqueness of the solution is no longer entailed. This last case is the one we want to treat for algebraic geometry codes.

Remark 1.4. Mind that it is not always easy to compute the exact minimum distance of an algebraic geometry code. Only lower bounds are known and the main one is referred to as the *designed distance*, which is why for this codes the unique decoding bound is considered to be half the designed distance of the code. We will present the notion of algebraic geometry codes and designed distance in the next section.

There exist several decoding algorithms which solve Problem 1 for algebraic geometry codes for t up to half the designed distance and even beyond. We know that in the second case, there could be more than one solution, and it is not always easy to chose the "best" one. Indeed there could be cases, called *worst cases*, where two solutions $\boldsymbol{c}_1, \boldsymbol{c}_2$ exist and satisfy $\mathrm{d}(\boldsymbol{c}_1, \boldsymbol{y}) = \mathrm{d}(\boldsymbol{c}_2, \boldsymbol{y})$. Some decoding algorithms treat these cases by giving back the list of all possible solutions and then take the name of *list decoding algorithm*. Other algorithms, like the one we are going to propose in this paper, are called *probabilistic* and give back one solution or fail. For sake of simplicity, all along this paper we work on Problem 1 for a generic t, by making the following assumption.

Assumption 1. There exists $\boldsymbol{c} \in \mathcal{C}$ and $\boldsymbol{e} \in \mathbb{F}_q^n$ with $\mathrm{w}(\boldsymbol{e}) = t$, such that

$$\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}.$$

That is, we assume Problem 1 to have a solution $\boldsymbol{c}$ and that $\mathrm{d}(\boldsymbol{c}, \boldsymbol{y}) = t$.

1.2 Algebraic geometry codes

In what follows, we will only consider smooth projective geometrically connected curves over $\mathbb{F}_q$. For the sake of semplicity, no proof will be included in this first section, but we direct the reader to [Sti09] and [TVN07] for further details. The Riemann Roch space associated to a curve $\mathcal{X}$ and a divisor G is defined as

$$L(G) \stackrel{\text{def}}{=} \{f \in \mathbb{F}_q(\mathcal{X})^* \mid (f) \geq -G\} \cup \{0\}.$$

It is in particular a vector space over $\mathbb{F}_q$ and its dimension is denoted by $\ell(G)$. Depending on the degree of G it is possible to deduce some informations about the dimension of the space $L(G)$.

Proposition 1.5. *The following properties hold*

- *If* $\deg G < 0$*, then* $L(G) = \{0\}$*;*
- $\ell(G) \geq \deg G - g + 1$*;*
- *if* $\deg G > 2g - 2$*, then* $\ell(G) = \deg G - g + 1$.

Theorem 1.6 (Clifford's Theorem)**.** *For all divisors* A *with* $0 \leq \deg A \leq 2g - 2$ *holds*

$$\ell(A) \leq 1 + \frac{1}{2} \deg A.$$

The reader can find the proof of this result in [Sti09, §1.6]. In the following sections, we will work consistently with Riemann Roch spaces and their dimension. First, we recall that the *support* of a divisor $A = \sum_P v_P(A)P$ is the set $\text{supp}(A) \stackrel{\text{def}}{=} \{P \mid v_P(A) \neq 0\}$. Given two divisors A, B it is possible to define the minimum (and symmetrically the maximum) of A and B

$$\min\{A, B\} \stackrel{\text{def}}{=} \sum_P \min\{v_P(A), v_P(B)\}P.$$

We recall the following properties:

- $L(A) \cap L(B) = L(\min\{A, B\})$,
- $L(A) + L(B) \subseteq L(\max\{A, B\})$.

Finally, we will need the following results which bound the dimension of the Riemann Roch space of a sum of divisors.

Proposition 1.7. *Let* A, B *be two divisors with* $B \geq 0$*. Then*

(i) $\ell(A - B) \geq \ell(A) - \deg B$

(ii) $\ell(A - B) \leq \max\{0, \ell(A) - \ell(B) + 1\}$

For the proof of (i) see [Sti09, Lemma 1.4.8]. For (ii), one can observe that if $\ell(A - B) = 0$ or $\ell(B) = 0$, then the inequality is clearly true, while the proof for $\ell(A - B), \ell(B) > 0$ can be found in [Sti09, Lemma 1.6.14].

It is now possible to introduce algebraic geometry codes, whose notion relies indeed on the one of Riemann Roch space. Given a sequence of rational points $\mathcal{P} = (P_1, \ldots, P_n)$ of a curve $\mathcal{X}$, the algebraic geometry code associated to $\mathcal{X}$, the divisor G and $\mathcal{P}$ is defined as

$$\mathcal{C}_L(\mathcal{X}, \mathcal{P}, G) = \{(f(P_1), \ldots, f(P_n)) \mid f \in L(G)\}.$$

In the rest of the paper we will often use the notation $\text{ev}_{\mathcal{P}}(f)$ to indicate the vector $(f(P_1), \ldots, f(P_n))$. Given an algebraic geometry code, we will call the P_i's, the *evaluation points* of the code and we will denote by D the divisor $P_1 + \cdots + P_n$. Furthermore, given the evaluation points $\mathcal{P}$ of an algebraic geometry code, we denote by ω a differential form such that $v_P(\omega) = -1$ and $\text{Res}_P(\omega) = 1$ for all $P \in \mathcal{P}$

and by W the canonical divisor associated to ω, that is $W = (\omega)$. It is known that, given such a W for an algebraic geometry code $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, G)$

$$\mathcal{C}_L(\mathcal{X}, \mathcal{P}, G)^\perp = C_\Omega(\mathcal{X}, \mathcal{P}, G) = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, W + D - G). \tag{1}$$

A proof of this result is given in [Sti09, Proposition 2.2.10]. As it is known, it is possible to estimate the dimension and the minimum distance of an algebraic geometry code. We recall here the properties we will need in the following sections.

Proposition 1.8. *Let $\mathcal{C} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, G)$, with $\deg G < n$. Then we have $\dim \mathcal{C} = \ell(G)$. In particular,*

- $\dim \mathcal{C} \geq \deg G - g + 1$ *and* $d(\mathcal{C}) \geq n - \deg G$*;*
- *if $\deg G > 2g - 2$, then $\dim \mathcal{C} = \deg G - g + 1$.*

The quantity $n - \deg G$ takes the name of *designed distance* of the code $\mathcal{C}$ and we denote it by d^*.

1.3 Star product

Given two vectors $\boldsymbol{a} = (a_1, \ldots, a_n)$, $\boldsymbol{b} = (b_1, \ldots, b_n) \in \mathbb{F}_q^n$, the *star product* (also called Schur product) of $\boldsymbol{a}$ and $\boldsymbol{b}$ is defined as

$$\boldsymbol{a} * \boldsymbol{b} \stackrel{\text{def}}{=} (a_1 b_1, \ldots, a_n b_n).$$

We denote by $\boldsymbol{a}^i$ the power with respect to the star product of the vector $\boldsymbol{a}$, that is $\boldsymbol{a}^i \stackrel{\text{def}}{=} (a_1^i, \ldots, a_n^i)$. One should be careful not to mix the notions of star product $\boldsymbol{a} * \boldsymbol{b}$ and canonical inner product in $\mathbb{F}_q^n$, $\langle \boldsymbol{a}, \boldsymbol{b} \rangle = \sum_{i=1}^n a_i b_i$. It is easy to prove that the following relation between the two operations holds:

$$\langle \boldsymbol{a} * \boldsymbol{b}, \boldsymbol{c} \rangle = \langle \boldsymbol{a}, \boldsymbol{b} * \boldsymbol{c} \rangle.$$

It is possible to generalise the notion of star product to subsets of $\mathbb{F}_q^n$ as well.

Definition 1.9. Given $A, B \subseteq \mathbb{F}_q^n$, the star product of A and B is defined as

$$A * B \stackrel{\text{def}}{=} \operatorname{span}_{\mathbb{F}_q}\{\boldsymbol{a} * \boldsymbol{b} \mid \boldsymbol{a} \in A, \boldsymbol{b} \in B\}.$$

Finally, given $i \in \mathbb{N}$, the power A^i is defined by induction as $A^i \stackrel{\text{def}}{=} A * A^{i-1}$, where $A^1 \stackrel{\text{def}}{=} A$.

1.4 Star product and algebraic geometry codes

Riemann Roch spaces, and so algebraic geometry codes, behave well with respect to Schur product. In particular given two divisors F and G, it is easy to see that $L(F)L(G) \subseteq L(F + G)$, hence

$$\mathcal{C}_L(\mathcal{X}, \mathcal{P}, G) * \mathcal{C}_L(\mathcal{X}, \mathcal{P}, G') \subseteq \mathcal{C}_L(\mathcal{X}, \mathcal{P}, G + G').$$

Under some further condition we can have the equality.

Proposition 1.10 (Star product of AG codes)**.** *Let $\mathcal{X}$ be a curve of genus g, $\mathcal{P} = (P_1, \ldots, P_n)$ be a sequence of rational points of $\mathcal{X}$ and G, G' be two divisors of $\mathcal{X}$ such that $\deg G \geq 2g$ and $\deg G' \geq 2g+1$. Then,*

$$\mathcal{C}_L(\mathcal{X}, \mathcal{P}, G) * \mathcal{C}_L(\mathcal{X}, \mathcal{P}, G') = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, G + G').$$

Proof. This is a consequence of [Mum70, Theorem 6]. For instance, see [CMCP17, Corollary 9]. □

2 Ehrhard's algorithm

In this section, we recall the version of Porter's algorithm for unique decoding proposed by Ehrhard [Ehr93]. In particular we report the adaptation process which premits to push the decoding radius up to half the designed distance of the code with no genus penalty, which is

$$\frac{d^* - 1}{2}.$$

Let us consider a code $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, G) \subseteq \mathbb{F}_q^n$, where $g - 1 \leq \deg(G) \leq n$ and $\mathrm{supp}(G) \cap \mathcal{P} = \emptyset$. We recall that by Assumption 1 the received vector is of the form $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$, where $\boldsymbol{c} = \mathrm{ev}_D(f_{\boldsymbol{c}})$ with $f_{\boldsymbol{c}} \in L(G)$ and $t = \mathrm{w}(e)$. In particular, since this is an algorithm for unique decoding, we suppose $t \leq \frac{d^*-1}{2}$. Note that in Ehrhard's paper the used language is the one of C_Ω codes, while we translated everything into C_L codes.

2.1 Foundations and purpose of the algorithm

First, we would like to express all vectors in $\mathbb{F}_q^n$ as vectors of evaluations of certain functions. In order to do that, we introduce a divisor G' such that $\mathrm{supp}(G') \cap \mathcal{P} = \emptyset$, $G' \geq G$ and $\ell(W + D - G') = 0$, where W has been defined in §1. We get then the inclusion

$$L(G) \subset L(G').$$

By using the hypothesis $\ell(W + D - G') = 0$, one can prove that there exists a vector space V such that $L(G) \subset V \subset L(G')$ and $\mathrm{ev}_{D|V} : V \to \mathbb{F}_q^n$ is an isomorphism (see [Ehr92]). Hence, we get the following diagram:

$$\begin{array}{ccccc} L(G) & \hookrightarrow & V & \hookrightarrow & L(G') \\ \downarrow{\scriptstyle \mathrm{ev}_D} & & \wr\downarrow{\scriptstyle \mathrm{ev}_D} & & \\ C_L(D, G) & \hookrightarrow & \mathbb{F}_q^n & & \end{array}$$

We can now see the received vector $\boldsymbol{y}$ and the error vector $\boldsymbol{e}$ as vectors of the evaluation of two functions $f_{\boldsymbol{e}}, f_{\boldsymbol{y}} \in L(G')$. We denote by $D_{\boldsymbol{e}}$ the divisor such that $0 \leq D_{\boldsymbol{e}} \leq D$ and $P_i \in \mathrm{supp}(D_{\boldsymbol{e}})$ if and only if $i \in \mathrm{supp}(\boldsymbol{e})$ (i.e. $e_i \neq 0$). The aim of Ehrhard's and many other decoding algorithms for algebraic geometry codes is to introduce an additional divisor F with $t + 2g \leq \deg(F) < n$ and try to compute the space

$$L(F - D_{\boldsymbol{e}}). \tag{2}$$

Indeed the space $L(F - D_{\boldsymbol{e}})$ is composed by all functions in $L(F)$ locating in some way the error positions. In particular, if $\mathrm{supp}(F) \cap \mathrm{supp}(D_{\boldsymbol{e}}) = \emptyset$, then $L(F - D_{\boldsymbol{e}})$ is composed by all functions in $L(F)$ which vanish at $\{P_i \mid i \in \mathrm{supp}(\boldsymbol{e})\}$. However, we will see soon that this last hypothesis on the support of F is not necessary to the algorithm and that, by adding a simple assumption on the degree of F, the very knowledge of an arbitrary nonzero $f \in L(F - D_{\boldsymbol{e}})$ makes possible to recover $f_{\boldsymbol{e}}$. First let us consider $\Lambda \in L(F - D_{\boldsymbol{e}})$. We have

$$\Lambda f_{\boldsymbol{y}} = \Lambda f_{\boldsymbol{c}} + \Lambda f_{\boldsymbol{e}},$$

where $\Lambda f_{\boldsymbol{c}} \in L(F + G)$ and $\Lambda f_{\boldsymbol{e}} \in L(F + G' - D)$. One can note that, if it is possible to isolate the second part, that is $\Lambda f_{\boldsymbol{e}}$, then by dividing by Λ we can recover $f_{\boldsymbol{e}}$. To do so, we want to add an assumption in order to have uniqueness of the decomposition of $\Lambda f_{\boldsymbol{y}}$. Hence, we now introduce the following map

$$\delta_{\boldsymbol{y}} : \left\{ \begin{array}{ccc} L(F) & \longrightarrow & L(F + G') \\ \Lambda & \longmapsto & \Lambda f_{\boldsymbol{y}}. \end{array} \right.$$

Let us analyse the set $L(F + G')$. One can note that $L(F + G)$, $L(F + G' - D) \subseteq L(F + G')$. Moreover, since $G' \geq G$ and $\mathrm{supp}(G') \cap \mathcal{P} = \emptyset$, we have $L(F + G) \cap L(F + G' - D) = L(F + G - D)$. The assumption we need is then the following one.

Assumption 2. We assume $\deg(G+F) < n$.

Now, thanks to Assumption 2, we get $L(F+G-D) = \{0\}$ and, for a certain vector space $Z_1 \subset L(F+G')$,

$$L(F+G') = L(F+G) \oplus L(F+G'-D) \oplus Z_1. \tag{3}$$

Theorem 2.1. *The very knowledge of an arbitrary $\Lambda \in L(F-D_{\boldsymbol{e}}) \setminus \{0\}$ makes possible to recover $f_{\boldsymbol{e}}$.*

Proof. Let us denote by π the projection $L(F+G') \to L(F+G'-D)$ with respect to the decomposition in (3). As said before, for any $\Lambda \in L(F-D_{\boldsymbol{e}})$ we have $\Lambda f_{\boldsymbol{c}} \in L(F+G)$ and $\Lambda f_{\boldsymbol{e}} \in L(F+G'-D)$, that is

$$\Lambda f_{\boldsymbol{y}} \in L(F+G) \oplus L(F+G'-D). \tag{4}$$

In particular, since the decomposition is unique, the equality $\Lambda f_{\boldsymbol{e}} = \pi(\Lambda f_{\boldsymbol{y}})$ holds. Therefore, if $\Lambda \neq 0$, we can easily recover $f_{\boldsymbol{e}} = \frac{\pi(\Lambda f_{\boldsymbol{y}})}{\Lambda}$. □

Remark 2.2. One can note that Theorem 2.1 holds whenever Λ belongs to a space different from $\{0\}$ of the form

$$L(F-D_{\boldsymbol{e}'}),$$

where $\mathrm{supp}(\boldsymbol{e}') \supseteq \mathrm{supp}(\boldsymbol{e})$. However, in order to have $L(F-D_{\boldsymbol{e}'}) \neq \{0\}$, we need the support of $\boldsymbol{e}'$ to be not too large. Indeed, let us consider the designed distance $d^* = n - \deg G$ of the code and suppose $\mathrm{w}(\boldsymbol{e}') \geq d^*$. By Assumption 2, we have

$$\deg(F-D_{\boldsymbol{e}'}) = \deg F - \mathrm{w}(\boldsymbol{e}') \leq \deg F - n + \deg G < 0.$$

Hence, by Proposition 1.5, $L(F-D_{\boldsymbol{e}'}) = \{0\}$.

2.2 The algorithm

The problem now is to find a way to compute $L(F-D_{\boldsymbol{e}})$, without knowing the support of the error vector. In Ehrhard's paper for decoding up to half the designed distance [Ehr93], the idea is to compute the space

$$S(F) \stackrel{\text{def}}{=} \{f \in L(F) \mid \delta_{\boldsymbol{y}}(f) \in L(F+G) \oplus L(F+G'-D)\}, \tag{5}$$

which fulfills the inclusion $L(F-D_{\boldsymbol{e}}) \subseteq S(F)$ (see (4)), and adapt the divisor F to have the equality. The main result which makes that possible is the following:

Proposition 2.3. *Assume $L(F-D_{\boldsymbol{e}}) \neq \{0\}$ and $\deg(F) \leq d^* - g - 1$. Then one and only one of the following statements holds:*

- *$S(F) = L(F-D_{\boldsymbol{e}})$;*
- *There exists a rational point $P \in \mathrm{supp}(D)$ with $\dim S(F-P) \leq \dim S(F) - 2$.*

For the proof, see [Ehr93]. We emphasise that there are no hypotheses on the support of F and that in particular the result remains true if $\mathrm{supp}(F) \cap \mathcal{P} \neq \emptyset$. Proposition 2.3 tells us that either we already have $S(F) = L(F-D_{\boldsymbol{e}})$, or we can construct a new divisor $F-P$ for some $P \in \mathrm{supp}(D)$, such that $\dim S(F-P)$ decreases quite fast with respect to $\dim S(F)$. Furthermore, we have $L(F-P-D_{\boldsymbol{e}}) \subseteq L(F-D_{\boldsymbol{e}})$, where in particular

$$\ell(F-D_{\boldsymbol{e}}) - 1 \leq \ell(F-P-D_{\boldsymbol{e}}) \leq \ell(F-D_{\boldsymbol{e}}). \tag{6}$$

Let us denote by $\{P_{i_m}\}_{m \geq 1}$ the sequence of found points in the support of D such that for any $m \geq 0$

$$\dim S(F - \sum_{j=1}^{m} P_{i_j} - P_{i_{m+1}}) \leq \dim S(F - \sum_{j=1}^{m} P_{i_j}) - 2,$$

and by F_{m+1} the divisor $F_m - P_{i_{m+1}}$, where $F_0 \stackrel{\text{def}}{=} F$. Hence, since the sequence

$$\dim S(F) \geq \dim S(F_1) \geq \dim S(F_2) \geq \cdots$$

decreases faster than the sequence

$$\ell(F - D_{\boldsymbol{e}}) \geq \ell(F_1 - D_{\boldsymbol{e}}) \geq \ell(F_2 - D_{\boldsymbol{e}}) \geq \cdots$$

and $L(F_m - D_{\boldsymbol{e}}) \subseteq S(F_m)$ for any m, there will be an equality for some m. However, we need to have enough elements F_m in the sequence, in order to have the equality $L(F_m - D_{\boldsymbol{e}}) = S(F_m)$ for one of them. Therefore, the two hypotheses of Proposition 2.3 $L(F_m - D_{\boldsymbol{e}}) \neq \{0\}$ and $\deg(F_m) \leq d^* - g - 1$ need to be fulfilled for several F_m's, in order to build a long enough sequence. The result in Theorem 2.6 will emphasise the role of the hypothesis $t \leq \frac{d^*-1}{2}$ in this problem. In order to prove this theorem, we will need the following proposition and corollary.

Proposition 2.4. *Let π be the projection $L(F+G') \to L(F+G'-D)$ with respect to the decomposition in* (3)*. There is an exact sequence of vector spaces*

$$0 \longrightarrow L(F - D_{\boldsymbol{e}}) \xrightarrow{\ i\ } S(F) \xrightarrow{\ \Phi\ } L(G + F - D + D_{\boldsymbol{e}})$$

where for any $\Gamma \in S(F)$, $\Phi(\Gamma) = \Gamma f_{\boldsymbol{e}} - \pi(\Gamma f_{\boldsymbol{y}})$.

Proof. See [Ehr93]. □

Corollary 2.5. *We have $\ell(F - D_{\boldsymbol{e}}) \leq \dim S(F) \leq \ell(F - D_{\boldsymbol{e}}) + \ell(G + F - D + D_{\boldsymbol{e}})$.*

Theorem 2.6. *Let $\mathcal{X}$ be a curve of genus g and $C = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, G)$ an algebraic geometry code on $\mathcal{X}$ with designed distance $d^* \geq 6g$. Let F be any divisor of degree $\deg F = t + 2g$. Then Algorithm 1 corrects every vector $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$ with $t = \mathrm{w}(\boldsymbol{e}) \leq \frac{d^*-1}{2}$.*

Proof. The proof can be found in [Ehr93], but we report it here, since a generalisation of this result will be presented in the next section. Let F be a divisor with $\deg F = t + 2g$, where $t = \mathrm{w}(\boldsymbol{e}) \leq \frac{d^*-1}{2}$. We denote by $F_1 = F, F_2, F_3 \ldots$ the sequence of divisors constructed by applying Proposition 2.3. First, let us prove that this sequence exists, that is, for any i smaller than a certain bound, the hypotheses of Proposition 2.3 hold for F_i. One can observe that since by hypothesis $t \leq \frac{d^*-1}{2}$ and $d^* \geq 6g$, then

$$d^* - t \geq d^* - \frac{d^* - 1}{2} \geq \frac{6g + 1}{2} = 3g + \frac{1}{2}.$$

In particular $t \leq d^* - 3g - 1$. Therefore, for any m we have

$$\deg F_m = 2g + t - m \leq 2g + t \leq d^* - g - 1.$$

We now prove that $L(F_m - D_{\boldsymbol{e}}) \neq \{0\}$ for any $m \leq g$. By Riemann-Roch theorem we get

$$\ell(F_m - D_{\boldsymbol{e}}) \geq t + 2g - m - t - g + 1 \geq 1. \tag{7}$$

Therefore the hypotheses of Proposition 2.3 are fulfilled for at least $F_0, \ldots, F_g$, which means that we can actually construct this sequence of divisors. We define

$$\Delta_m \stackrel{\text{def}}{=} \dim S(F_m) - \ell(F_m - D_{\boldsymbol{e}}). \tag{8}$$

We now show that $\Delta_0 \leq g$ and that the sequence $\{\Delta_m\}_m$ is strictly decreasing. By Corollary 2.5 we get

$$\Delta_0 = \dim S(F) - \ell(F - D_{\boldsymbol{e}}) \leq \ell(G + F - D + D_{\boldsymbol{e}}). \tag{9}$$

Since $t \leq \frac{d^*-1}{2}$ and $d^* = n - \deg G$, we have in particular

$$\deg(G + F - D + D_{\boldsymbol{e}}) = n - d^* + t + 2g - n + t = 2t - d^* + 2g \leq 2g - 1. \tag{10}$$

Now we claim that $\ell(G + F - D + D_{\boldsymbol{e}}) \leq g$. If $\deg(G + F - D + D_{\boldsymbol{e}}) = 2g - 1 > 2g - 2$, we have by Riemann Roch theorem

$$\ell(G + F - D + D_{\boldsymbol{e}}) = \deg(G + F - D + D_{\boldsymbol{e}}) - g + 1 = 2g - 1 - g + 1 = g.$$

Otherwise, if $\deg(G+F-D+D_{\boldsymbol{e}}) \leq 2g-2$, by Clifford's Theorem (see Theorem 1.6), we get

$$\ell(G+F-D+D_{\boldsymbol{e}}) \leq 1+\frac{1}{2}\deg(G+F-D+D_{\boldsymbol{e}}) \leq g.$$

Finally, we prove that the sequence of the Δ_m's is stricly decreasing. By using (6) and the definition of the P_{i_m}'s, we have

$$\begin{aligned}\Delta_{m+1} &= \dim S(F_m - P_{i_{m+1}}) - \ell(F_m - P_{i_{m+1}} - D_{\boldsymbol{e}}) \\ &\leq \dim S(F_m) - 2 - \ell(F_m - D_{\boldsymbol{e}}) + 1 \\ &= \Delta_m - 1.\end{aligned}$$

For any $m \leq g$, if $\Delta_m = 0$ then the algorithm stops. Otherwise the algorithm constructs F_{m+1} and gets $\Delta_{m+1} \leq \Delta_m - 1$. In this way, we can contruct for sure at least $g+1$ divisors $F_0, \ldots, F_g$ and it is enough. Indeed, since $\Delta_0 \leq g$ and the sequence is strictly decreasing, we will get $\Delta_m = 0$ for some $m \leq g$. □

Remark 2.7. In this remark we want to point out the reason why the process of adaptation of the divisor F becomes necessary to reach the decoding radius $\frac{d^*-1}{2}$. In [Ehr92], the only space $S(F)$ is computed and there is no adaptation process to have $S(F) = L(F - D_{\boldsymbol{e}})$. Indeed the strategy is rather to find a condition for this equality to hold from the start. This condition turns out to be a bound on the degree of F, $\deg F < d^* - t$ (see Proposition 1 at the end of the remark). The price of this bound though, is a limitation on the decoding radius. Indeed, in [Ehr92], the lower bound for $\deg F$ is $t+g$, which together with the hypothesis in Proposition 1 of [Ehr92], gives

$$t + g \leq \deg F < d^* - t,$$

that is, we have the decoding radius $t \leq \frac{d^*-1-g}{2}$. Once this decoding radius is exceeded, we have $\deg F \geq d^* - t$, the equality between $S(F)$ and $L(F - D_{\boldsymbol{e}})$ could no longer hold and the process of adaptation of the divisor F becomes necessary to ensure it. Thanks to this process, we have then the improvement of the decoding radius from $\frac{d^*-1-g}{2}$ ([Ehr92]) up to $\frac{d^*-1}{2}$ ([Ehr93]).

Proposition 1 [Ehr92] (Function version). *If* $\deg F + t < d^*$, *then* $L(F - D_{\boldsymbol{e}}) = S(F)$.

The reader can find the proof in Appendix B.

Remark 2.8. Observe that once $t > \frac{d^*-1}{2}$, we may have $\Delta_0 > g$ (see (10)). In particular, we could need more than g steps to have the equality $S(F_j) = L(F_j - D_{\boldsymbol{e}})$ for some j. Though, if $j > g$, we may have $L(F_j - D_{\boldsymbol{e}}) = \{0\}$ (see (7)), while for the algorithm to work, we need $L(F_j - D_{\boldsymbol{e}}) \neq \{0\}$ (see Theorem 2.1).

Algorithm 1 Ehrhard algorithm - unique decoding

Inputs: $f_{\boldsymbol{y}} = f_{\boldsymbol{c}} + f_{\boldsymbol{e}} \in \mathbb{F}_q^n$ where $\boldsymbol{c} \in C$ and $\mathrm{w}(\boldsymbol{e}) \leq \frac{d^*-1}{2}$, $t = \mathrm{w}(\boldsymbol{e})$.
Output: $f_{\boldsymbol{e}} \in L(G')$ such that $\mathrm{ev}_{\mathcal{P}}(f_{\boldsymbol{e}}) = \boldsymbol{e}$.

1: Choose F with $\mathrm{supp}(F) \cap \mathcal{P} = \emptyset$ and $\deg F = t + 2g$;
2: $j \leftarrow 0$ and $F_0 \leftarrow F$;
3: Look for a point $P \in \{P_1, \ldots, P_n\}$ such that $\dim(S(F_j - P)) \leq \dim(S(F_j)) - 2$;
4: **if** such a point P exists **then**
5: $\quad F_{j+1} \leftarrow F_j - P$;
6: $\quad j \leftarrow j + 1$;
7: $\quad$ go to Step 3;
8: **else** compute $f_{\boldsymbol{e}} = \frac{\pi(\Lambda f_{\boldsymbol{y}})}{\Lambda}$ for some $\Lambda \in S(F_j)$;
9: **return** $f_{\boldsymbol{e}}$;

Remark 2.9. In the process of adaptation of the divisor F, the points $P_{m+1} \in \mathrm{supp}(D)$ such that

$$\dim S(F_m - P_{m+1}) \leq \dim S(F_m) - 2,$$

does not belong necessarely to $\mathrm{supp}(D_{\boldsymbol{e}})$. The tests we made on the generalised algorithm give some evidences of this fact in §5.

3 Generalisation of the algorithm to correct more errors ($\ell = 2$)

We want now to solve the decoding problem for $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$ with $\mathrm{w}(\boldsymbol{e}) > \frac{d^*-1}{2}$. In particular we know that any decoding algorithm for Reed-Solomon codes, decoding beyond half the minimum distance of the code, once generalised to algebraic geometry codes, presents a penalty given by the genus of the curve. For instance, the decoding radius of Sudan algorithm ($\ell = 2$) for algebraic geometry codes is

$$t_{Sud} = \frac{2n - 3\deg G - 2}{3} - \frac{2}{3}g. \tag{11}$$

(see Appendix A). We would like to generalise Ehrhard algorithm in order to correct the same amount of errors, without the term in g.

3.1 Foundation of the algorithm

The purpose of the algorithm stays the same, that is to introduce a certain divisor F and compute the space $L(F - D_{\boldsymbol{e}})$. What changes is that we want to be able to find this space even for an amount of errors which is larger than half the designed distance. From Remark 2.8, we know that in this situation, given F with $\deg F = t + 2g$, the gap between $S(F)$ and $L(F - D_{\boldsymbol{e}})$ could be too large with respect to g. This gap decreases by one at each step, but we would like to fill it in g steps. The idea then, is to work with a different space $S(F)$ such that the gap decreases by ℓ at each step instead, for a certain parameter ℓ. In order to do so, we now generalise the foundations of the algorithm. Let us consider $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, G)$ with G as in the previous section and the codes

$$A = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, F) \quad B = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, W + D - G - F). \tag{12}$$

The idea of the generalisation, is based on the following remark.

Remark 3.1. One can prove that $\mathrm{ev}_{\mathcal{P}}(S(F)) \subseteq K_{\boldsymbol{y}}$, where $K_{\boldsymbol{y}}$ is the set computed in the *error correcting pairs algorithm* (see [Pel92]), that is

$$K_{\boldsymbol{y}} \stackrel{\text{def}}{=} \{\boldsymbol{a} \in A \mid \langle \boldsymbol{a} * \boldsymbol{y}, \boldsymbol{b} \rangle = 0 \ \ \forall \boldsymbol{b} \in B\}.$$

In [Pel92], this set is $\mathrm{Ker}(E_{\boldsymbol{w}})$, for a specific linear application $E_{\boldsymbol{w}}$ and in [Ehr92, §3] it is proved that the equality $\mathrm{ev}_D(S(F)) = K_{\boldsymbol{y}}$ holds whenever $\mathrm{supp}(F) \cap \mathcal{P} = \emptyset$.

Let us rename the spaces $S(F)$ and $K_{\boldsymbol{y}}$ respectively by $S_1(F)$ and $K_{\boldsymbol{y}}^{(1)}$. We know that in the generalisation of the error correcting pairs algorithm to correct more errors, that is the *power error locating pairs* (see [CP20]), the intersection of several spaces $K_{\boldsymbol{y}} = \cap_{i=1}^{\ell} K_{\boldsymbol{y}}^{(i)}$ is computed rather than the only $K_{\boldsymbol{y}}^{(1)}$. Let us consider $\ell = 2$ for the moment (we present the general case $\ell \geq 2$ in §4). We have

$$K_{\boldsymbol{y}}^{(1)} \stackrel{\text{def}}{=} \{\boldsymbol{a} \in A \mid \langle \boldsymbol{a} * \boldsymbol{y}, \boldsymbol{b} \rangle = 0 \ \ \forall \boldsymbol{b} \in B\}, \tag{13}$$

$$K_{\boldsymbol{y}}^{(2)} \stackrel{\text{def}}{=} \{\boldsymbol{a} \in A \mid \langle \boldsymbol{a} * \boldsymbol{y}^2, \boldsymbol{b} \rangle = 0 \ \ \forall \boldsymbol{b} \in (B^{\perp} * C)^{\perp}\}. \tag{14}$$

As said in Remark 3.1, we know that, whenever $\mathrm{supp}(F) \cap \mathcal{P} = \emptyset$, $K_{\boldsymbol{y}}^{(1)}$ corresponds to $S_1(F)$ by evaluating in the points $P_1, \ldots, P_n$. Let us find a space $S_2(F)$ which reformulates in the same spirit $K_{\boldsymbol{y}}^{(2)}$. First, let us consider the vector $\boldsymbol{y}^2$. We have seen in §2 that there exists $f_{\boldsymbol{y}} \in L(G')$ such that $\mathrm{ev}_{\mathcal{P}}(f_{\boldsymbol{y}}) = \boldsymbol{y}$. In particular, we get $f_{\boldsymbol{y}}^2 \in L(2G')$ and $\mathrm{ev}_{\mathcal{P}}(f_{\boldsymbol{y}}^2) = \boldsymbol{y}^2$. We now denote $f_{\boldsymbol{y}}^2$ by $f_{\boldsymbol{y}^2}$ and define the following map

$$\delta_{\boldsymbol{y}^2} : \left\{ \begin{array}{ccc} L(F) & \longrightarrow & L(F + 2G') \\ \Lambda & \longmapsto & \Lambda f_{\boldsymbol{y}^2}. \end{array} \right.$$

One can easily prove that this map is well-defined. We would like to see the space $L(F + 2G')$ as the direct sum of some particular subspaces (as for $L(F + G')$ in §2.1). As in §2.1, we have that both the spaces $L(F + 2G)$ and $L(F + 2G' - D)$ are included in $L(F + 2G')$ and it holds

$$L(F + 2G) \cap L(F + 2G' - D) = L(F + 2G - D).$$

Assumption 3. We assume that $\deg(F + 2G) < n$.

Under Assumption 3, we get $\deg(F+2G-D)<0$, hence $L(F+2G-D)=\{0\}$ and there exists a subspace Z_2 of $L(F+2G')$ such that

$$L(F+2G') = L(F+2G) \oplus L(F+2G'-D) \oplus Z_2. \tag{15}$$

Remark 3.2. The idea of decoding at the same time several powers of the same vector $\boldsymbol{y}$, follows the one of Sidorenko, Schmidt and Bossert in the so called *power decoding algorithm* for Reed–Solomon codes [SSB10], inspired in turn by a decoding algorithm of interleaved Reed–Solomon codes. Observe that we are applying here the same procedure. Indeed we are now considering two decoding problems, that is, one with received vector $\boldsymbol{y}$ and code $\mathcal{C}_L(\mathcal{X},\mathcal{P},G)$ and one with received vector $\boldsymbol{y}^2$ and code $\mathcal{C}_L(\mathcal{X},\mathcal{P},2G)$. That is why the construction we have just made for $\boldsymbol{y}^2$ is equivalent to that for $\boldsymbol{y}$ (§2.1) but with the divisor $2G'$, the code $\mathcal{C}_L(\mathcal{X},\mathcal{P},2G)$ and the received vector $\boldsymbol{y}^2$, instead of respectively G', $\mathcal{C}_L(\mathcal{X},\mathcal{P},G)$ and $\boldsymbol{y}$. From the point of view of applying the algorithm to two received vectors, Assumption 3 comes as a natural request to correct the received vector $\boldsymbol{y}^2$ as it plays the role of Assumption 2 with $\boldsymbol{y}$. Furthermore this assumption makes easier to compute the decoding radius of our algorithm (see Lemma 3.4 and Theorem 3.11). Though, since the two decoding problems are related, $\boldsymbol{y}^2$ being the square of $\boldsymbol{y}$, Assumption 3 is not as important as it seems. Indeed, given $\Lambda \in L(F-D_{\boldsymbol{e}})$, we do not really need the space $L(F+2G')$ to split as in (15) to recover $f_{\boldsymbol{e}}$, since we already know how to do that by Assumption 2 together with Theorem 2.1. We want to point out then that Assumption 3 is not a necessary condition for the algorithm to work, as shown by our tests in §5.

Remark 3.3. Note that, since $\deg(G)>0$, Assumption 3 implies Assumption 2.

It is actually possible to compute the dimension of the spaces Z_1 and Z_2.

Lemma 3.4. *Given Z_1 as in* (3) *and Z_2 as in* (15)*, then*

$$\begin{aligned}\dim Z_1 &= \deg(D-F-G)+g-1\\ \dim Z_2 &= \deg(D-F-2G)+g-1\end{aligned}$$

Proof. First, we show that $\Omega(F+iG)=\Omega(F+iG'-D)=\{0\}$ for $i=1,2$. Since we took G' such that $\ell(W+D-G')=0$, by Riemann-Roch theorem we have

$$0=\ell(W+D-G') \geq 2g+n-\deg G'-g+1,$$

that is $\deg G' \geq n+g-1$. Thus, since $\deg F \geq t+g$, we get

$$\deg(F+2G'-D) > \deg(F+G'-D) > 2g-2 \tag{16}$$

and in particular $\Omega(F+G'-D)=\Omega(F+2G'-D)=\{0\}$. Furthermore,

$$\deg(F+2G) > \deg(F+G) \geq t+g+g-1 > 2g-2,$$

hence $\Omega(F+2G)=\Omega(F+G)=\{0\}$. Therefore we can compute from (3)

$$\begin{aligned}\dim Z_1 &= \ell(F+G')-\ell(F+G)-\ell(F+G'-D)\\ &= \deg F+\deg G'-g+1-\deg F-\deg G+g-1-\deg F-\deg G'+n+g-1\\ &= \deg(D-F-G)+g-1.\end{aligned}$$

In the same way from (15) we get $\dim Z_2=\deg(D-F-2G)+g-1$. □

3.2 The algorithm

We can now define the space

$$S_2(F) \stackrel{\text{def}}{=} \{f \in L(F) \mid \delta_{\boldsymbol{y}^2}(f) \in L(F+2G) \oplus L(F+2G'-D)\}. \tag{17}$$

As for $S_1(F)$, we have that $\mathrm{ev}_{\mathcal{P}}(S(F)) \subseteq K_{\boldsymbol{y}}^{(2)}$ and under further conditions on the support of the divisor F and $\deg G$, we have the equality.

Theorem 3.5. *Given $S_2(F)$ as in (17) and $K_{\boldsymbol{y}}^{(2)}$ as in (14), if $\mathrm{supp}(F) \cap \mathcal{P} = \emptyset$ and $\deg G \geq 2g$, we have*

$$\mathrm{ev}_{\mathcal{P}}(S_2(F)) = K_{\boldsymbol{y}}^{(2)}.$$

Proof. See Appendix B. □

Proposition 3.6. *Let $D_{\boldsymbol{e}}$ be as in the previous section. Then $L(F - D_{\boldsymbol{e}}) \subset S_2(F)$.*

Proof. Let us consider $f_{\boldsymbol{y}^2}$. We recall that we defined this function to be equal to $f_{\boldsymbol{y}}^2$, hence it belongs to $L(2G')$ and fulfills $\mathrm{ev}_{\mathcal{P}}(f_{\boldsymbol{y}^2}) = \boldsymbol{y}^2$. In particular

$$f_{\boldsymbol{y}}^2 = f_{\boldsymbol{c}}^2 + 2 f_{\boldsymbol{c}} f_{\boldsymbol{e}} + f_{\boldsymbol{e}}^2,$$

where $f_{\boldsymbol{c}} \in L(G)$ and $f_{\boldsymbol{e}} \in L(G')$. If $\Lambda \in L(F - D_{\boldsymbol{e}})$, then we get

$$\begin{aligned} (\Lambda f_{\boldsymbol{c}}^2) &\geq -F - 2G, \\ (\Lambda f_{\boldsymbol{e}}^2) &\geq -F + D - 2G', \\ (\Lambda f_{\boldsymbol{c}} f_{\boldsymbol{e}}) &\geq -F + D - G - G' \geq -F + D - 2G', \end{aligned}$$

since $G' > G$ and in particular $f_{\boldsymbol{e}} \in L(G' - D + D_{\boldsymbol{e}})$. □

Let us finally introduce the space

$$S(F) \stackrel{\mathrm{def}}{=} S_1(F) \cap S_2(F). \tag{18}$$

Thanks to Proposition 3.6, we have $L(F - D_{\boldsymbol{e}}) \subseteq S(F)$. As in Ehrhard's paper, the idea is now to close the gap between these two spaces. Hence, the next task is to adapt Proposition 2.3 to the $S(F)$ we have just constructed. To do so, we have to adapt two lemmas. The proofs of these two lemmas and of the adaptation of Proposition 2.3 come directly from the proofs of Lemma 1, Lemma 2 and Proposition 8 in [Ehr93], though we write them here anyway for sake of completeness.

Lemma 3.7. *If $L(F - D_{\boldsymbol{e}}) \neq S(F)$, then there exist at most $\deg(F + D_{\boldsymbol{e}}) - d^*$ rational points $P \in \mathrm{supp}(D_{\boldsymbol{e}})$ such that $S(F) \subseteq L(F - P)$.*

Proof. Without loss of generality, after reindexing, one can suppose $P_1, \ldots, P_m$ to be the points in $\mathrm{supp}(D_{\boldsymbol{e}})$ such that

$$S(F) \subseteq L(F - P_i) \quad \forall i = 1, \ldots, m.$$

In particular we have $S(F) \subseteq \bigcap_{i=1}^m L(F - P_i) = L(F - \tilde{D})$, where $\tilde{D} = \sum_{i=1}^m P_i$. Let us consider

$$\Gamma \in S(F) \setminus L(F - D_{\boldsymbol{e}}) \subseteq S_1(F) \setminus L(F - D_{\boldsymbol{e}}).$$

By Proposition 2.4, we have $\Phi(\Gamma) = \Gamma f_{\boldsymbol{e}} - \pi(\Gamma f_{\boldsymbol{y}}) \neq 0$ where $\pi(\Gamma f_{\boldsymbol{y}}) \in L(F + G' - D)$. We get then

$$(\Gamma f_{\boldsymbol{e}} - \pi(\Gamma f_{\boldsymbol{y}})) \geq \min(-F + \tilde{D} - G' + D - D_{\boldsymbol{e}}, -F - G' + D) = -G' - F + D - (D_{\boldsymbol{e}} - \tilde{D}).$$

By definition of Φ in Proposition 2.4 we get in particular

$$0 \neq \Gamma f_{\boldsymbol{e}} - \pi(\Gamma f_{\boldsymbol{y}}) \in L(G' + F - D + (D_{\boldsymbol{e}} - \tilde{D})) \cap L(G + F - D + D_{\boldsymbol{e}}) = L(G + F - D + (D_{\boldsymbol{e}} - \tilde{D})).$$

Hence, $\deg(G + F - D + (D_{\boldsymbol{e}} - \tilde{D})) \geq 0$, that is

$$m \leq \deg(F + D_{\boldsymbol{e}}) - d^*.$$

□

Lemma 3.8. *If $L(F - D_{\boldsymbol{e}}) \neq \{0\}$, then there are at most g rational points $P \in \mathrm{supp}(D_{\boldsymbol{e}})$ such that $S_1(F - P) \cap S_2(F) = S(F) \cap L(F - P)$.*

Proof. As a consequence of Riemann Roch theorem, there are at most g points P in $\mathrm{supp}(D_{\boldsymbol{e}})$ such that $L(F - D_{\boldsymbol{e}} - P) = L(F - D_{\boldsymbol{e}})$. Indeed again without loss of generality we can suppose that $P_1, \dots, P_m$ are the points in $\mathrm{supp}(D_{\boldsymbol{e}})$ such that $L(F - D_{\boldsymbol{e}}) = L(F - D_{\boldsymbol{e}} - P_i)$ for all $i = 1, \dots, m$. In particular we have

$$L(F - D_{\boldsymbol{e}}) = L\left(F - D_{\boldsymbol{e}} - \sum_{i=1}^{m} P_i\right).$$

Furthermore, by Proposition 1.7,

$$\ell(F - D_{\boldsymbol{e}}) \leq \ell(F - D_{\boldsymbol{e}}) - \ell\left(\sum_{i=1}^{m} P_i\right) + 1 \leq \ell(F - D_{\boldsymbol{e}}) - m + g - 1 + 1.$$

Hence, $m \leq g$. Now, it suffices to prove that given a point $P \in \mathrm{supp}(D_{\boldsymbol{e}})$

$$L(F - D_{\boldsymbol{e}} - P) \neq L(F - D_{\boldsymbol{e}}) \implies S_1(F - P) \cap S_2(F) \neq S(F) \cap L(F - P).$$

Let us consider $\Gamma \in L(F - D_{\boldsymbol{e}}) \setminus L(F - D_{\boldsymbol{e}} - P)$. In particular we have $\Gamma \in S(F) \cap L(F - P)$. We now show that $\Gamma \notin S_1(F - P)$. If it were, we would have $\Gamma f_{\boldsymbol{y}} = g + h$ with $g \in L(F + G - P)$ and $h \in L(F + G' - D - P)$. On the other hand, $\Gamma f_{\boldsymbol{y}} = \Gamma f_{\boldsymbol{c}} + \Gamma f_{\boldsymbol{e}} \in L(F + G) \oplus \text{Ł}(F + G' - D)$. Both decompositions are in $L(F + G) \oplus L(F + G' - D)$, hence the uniqueness gives

$$h = \Gamma f_{\boldsymbol{e}} \in L(F + G' - D - P). \tag{19}$$

Though note that $v_P(\Gamma f_{\boldsymbol{e}}) = v_P(\Gamma) + v_P(f_{\boldsymbol{e}}) = -v_P(F)$, against (19). □

Proposition 3.9. *Assume* $L(F - D_{\boldsymbol{e}}) \neq \{0\}$ *and* $\deg F \leq d^* - g - 1$. *Let* $S(F)$ *be as in* (18). *Then one and only one of the following statement holds:*

- $S(F) = L(F - D_{\boldsymbol{e}})$
- *There exists a rational point* $P \in \mathrm{supp}(D)$ *with* $\dim(S(F - P)) \leq \dim(S(F)) - 2$.

Proof. If $S(F) = L(F - D_{\boldsymbol{e}})$, then for any point $P \in \mathrm{supp}(D)$, we get

$$\dim S(F - P) \geq \ell(F - P - D_{\boldsymbol{e}}) \geq \ell(F - D_{\boldsymbol{e}}) - 1 = \dim S(F) - 1.$$

Now, if $S(F) \neq L(F - D_{\boldsymbol{e}})$, by Proposition 2.4 we have $L(G + F - D + D_{\boldsymbol{e}}) \neq \{0\}$. Hence we get $\deg(G + F - D + D_{\boldsymbol{e}}) \geq 0$ that is $\deg(F + D_{\boldsymbol{e}}) - d^* \geq 0$. Hence by applying Lemma 3.7 and Lemma 3.8 and using the hypothesis $\deg(F) \leq d^* - g - 1$, we get that there exist at least

$$\deg D_{\boldsymbol{e}} - g - \deg(F + D_{\boldsymbol{e}}) + d^* = d^* - \deg F - g \geq 1$$

points P in $\mathrm{supp}(D_{\boldsymbol{e}})$ such that $S(F) \nsubseteq L(F - P)$ and $S_1(F - P) \cap S_2(F) \neq S(F) \cap L(F - P)$. Hence for such a point we get

$$S(F - P) \subseteq S_1(F - P) \cap S_2(F) \nsubseteq S(F) \cap L(F - P) \nsubseteq S(F).$$

In particular $\dim S(F - P) \leq \dim S(F) - 2$. □

Thanks to Proposition 3.9, the sequence $\{\Delta_i\}_{i \geq 0}$ defined in (8) verifies $\Delta_{i+1} \leq \Delta_i - 1$. As said in the beginning of the section, we would like this sequence to decrease faster and it is clear that the faster decreases the sequence $\{\dim S(F_i)\}_{i \geq 0}$, the faster decreases the sequence $\{\Delta_i\}_{i \geq 0}$.

Remark 3.10. Let us consider F a generic divisor in $\{F_i\}_{i \geq 0}$, $P \in \mathrm{supp}(D_{\boldsymbol{e}})$ and $\Lambda \in S_1(F - P) \cap S_2(F)$. We want to understand whether $\Lambda \in S_2(F - P)$. We get by definition of $S_1(F - P)$ and $S_2(F)$:

$$\begin{aligned} &\Lambda f_{\boldsymbol{y}} \in L(F - P + G) \oplus L(F - P + G' - D) \\ &\Lambda f_{\boldsymbol{y}^2} \in L(F - 2G) \oplus L(F + 2G' - D). \end{aligned} \tag{20}$$

In particular, $\Lambda f_{\boldsymbol{y}} = g + h$ with $g \in L(F - P + G)$ and $h \in L(F - P + G' - D)$. Let us analyse $\Lambda f_{\boldsymbol{y}^2} = (g + h)(f_{\boldsymbol{c}} + f_{\boldsymbol{e}})$:

$$\begin{aligned}
(gf_{\boldsymbol{c}}) &\geq -F + P - 2G, \\
(gf_{\boldsymbol{e}}) &\geq -F + P - G - G' + D - D_{\boldsymbol{e}} \geq -F + P - 2G' + D - D_{\boldsymbol{e}}, \\
(hf_{\boldsymbol{c}}) &\geq -F + P - G' + D - G \geq -F + P - 2G' + D, \\
(hf_{\boldsymbol{e}}) &\geq -F + P - G' + D - G' + D - D_{\boldsymbol{e}} \geq -F + P - 2G' + D.
\end{aligned}$$

In particular, we have $gf_{\boldsymbol{c}}, hf_{\boldsymbol{c}}, hf_{\boldsymbol{e}} \in L(F - P + 2G) \oplus L(F - P + 2G' - D)$, while

$$gf_{\boldsymbol{e}} \in L(F - P - 2G' - D + D_{\boldsymbol{e}}).$$

Therefore, by (20), given $\Lambda \in S_1(F - P) \cap S_2(F)$ we have

$$\Lambda \in S_2(F - P) \iff gf_{\boldsymbol{e}} \in L(F - P + 2G) \oplus L(F - P + 2G' - D).$$

In particular, if Λ verifies this property for any $\Lambda \in S_1(F - P) \cap S_2(F)$, then

$$S_1(F - P) \cap S_2(F - P) = S_1(F - P) \cap S_2(F).$$

Empirical Behavior: we observed that for a random error vector, we have

$$\dim S(F_m - P_{m+1}) \leq \dim S(F_m) - 3.$$

In particular, it seems the three strict inclusions to be the following

$$S(F_m - P_{m+1}) = S_1(F_m - P_{m+1}) \cap S_2(F_m - P_{m+1}) \subsetneq S_1(F_m - P_{m+1}) \cap S_2(F_m) \subsetneq S(F_m) \cap L(F_m - P_{m+1}) \subsetneq S(F_m).$$

The second and third inclusions correspond to the two inclusions of respectively Lemma 3.7 and Lemma 3.8, while the first one seems to depend strictly on the chosen error vector. For random vectors, it is easy to find points such that the three inclusions are strict, while for a "worst case", that is, when we have two codewords at the same distance from $\boldsymbol{y}$, we could not find such a point and we got

$$\dim S(F_m - P_{m+1}) = \dim S(F_m) - 2.$$

Now that we have all the ingredients, we can describe the algorithm (actually the only thing that changes with respect to Algorithm 1 is that we use the new notion of $S(F)$) and try to compute its decoding radius by adapting the proof of Theorem 1 of [Ehr93]. Since we want to correct more than half the designed distance of the code and the algorithm gives back only one solution, we do not look for a sufficient condition for the algorithm to work, but rather for a necessary one.

Theorem 3.11. *Assume* $\deg F = t + 2g$ *with* $t \leq d^* - 3g - 1$. *If the error vector is such that* $S(F)$ *verifies the empirical behavior and* $\deg(2G + F) < n$, *then a necessary condition for the algorithm to work is*

$$t \leq \frac{2n - 3\deg G - 2}{3}. \tag{21}$$

Proof. Let us start by observing that, since $t \leq d^* - 3g - 1$, we have

$$\deg F_j = \deg F - j \leq \deg F = t + 2g \leq d^* - g - 1.$$

Note that as for Theorem 2.6, for any $j \leq g$, it holds

$$\ell(F_j - D_{\boldsymbol{e}}) \geq t + 2g - j - t - g + 1 \geq 1.$$

Hence, if necessary[1] we can apply Proposition 2.3 to F_j for any $1 \leq j \leq g$ and construct a sequence of divisors of length at least $g + 1$. Let us consider again the quantity

$$\Delta_j = \dim S(F_j) - \ell(F_j - D_{\boldsymbol{e}}),$$

[1] If we do not find $S(F_j) = L(F_j - D_{\boldsymbol{e}})$ for some $j < g$

where $\{F_j\}$ is the sequence of constructed divisor, that is, $F_{j+1} = F_j - P_{i_{j+1}}$. We claim that, if $\Delta_0 \leq 2g$, then $\Delta_j = 0$ for some $j \leq g$. Indeed as said before, we have $\ell(F_{j+1} - D_{\boldsymbol{e}}) \geq \ell(F_j - D_{\boldsymbol{e}}) - 1$ and by hypothesis $\dim S(F_{j+1}) \leq \dim S(F_j) - 3$, hence

$$\dim S(F_{j+1}) - \ell(F_{j+1} - D_{\boldsymbol{e}}) \leq \dim S(F_j) - \ell(F_j - D_{\boldsymbol{e}}) - 2.$$

Therefore the sequence of the Δ_j is strictly decreasing and $\Delta_j = 0$ for some $j \leq g$. Now we want to find a necessary condition to have $\Delta_0 \leq 2g$, that is to have

$$\ell(F - D_{\boldsymbol{e}}) + 2g \geq \dim S(F). \tag{22}$$

In order to do so, we want to bound $\dim S(F)$. It is possible to write $S(F)$ in the following way

$$S(F) = \{f \in L(F) \mid \pi_{Z_1}(\delta_{\boldsymbol{y}}(f)) = \mathbf{0} \ \wedge \ \pi_{Z_2}(\delta_{\boldsymbol{y}^2}(f)) = \mathbf{0}\}, \tag{23}$$

where π_{Z_1} and π_{Z_2} are respectively the projections $L(F + G') \to Z_1$ and $L(F + 2G') \to Z_2$ with respect to the decompositions (3) and (15). In particular, $S(F)$ is composed by the elements of $L(F)$ which fulfill certains conditions, therefore we can bound

$$\dim S(F) = \ell(F) - \#\text{conditions} \geq \ell(F) - \dim Z_1 - \dim Z_2. \tag{24}$$

Therefore, putting together the condition on Δ_0 (22) and (24), we get

$$\ell(F - D_{\boldsymbol{e}}) + 2g \geq \dim S(F) \geq \ell(F) - \dim Z_1 - \dim Z_2. \tag{25}$$

Now, by Lemma 3.4, we have

$$\dim Z_1 = n - t - g - \deg G - 1 \quad \dim Z_2 = n - t - g - 2\deg G - 1$$

By substituting the values of Z_1 and Z_2 in (25) and applying Riemann-Roch theorem we get

$$t \leq \frac{2n - 3\deg G - 2}{3}.$$

□

4 Generalisation to $\ell \geq 2$

In this section we will show how to generalise the strategy we have seen in §3 to build an algorithm with parameter $\ell \geq 2$. Experimentally, the decoding radius of this algorithm reaches the amount

$$\frac{2n\ell - \ell(\ell+1)\deg G - 2\ell}{2(\ell+1)}$$

which is the decoding radius of Sudan algorithm without any penalty in the genus of the curve (see Appendix A).

4.1 Foundation of the algorithm

As in the cases $\ell = 1, 2$, a divisor F with certain properties is introduced and the aim of the algorithm is to find the space $L(F - D_{\boldsymbol{e}})$. For that, once we fix ℓ, we need to define a space $S(F)$ wich contains $L(F - D_{\boldsymbol{e}})$ and such that, given a specific sequence of divisors $\{F_j\}_j$, the gap

$$\Delta_j = \dim S(F_j) - \ell(F_j - D_{\boldsymbol{e}})$$

decreases fast enough with respect to g. Again, the following assumption comes naturally if we think we are applying the basic algorithm to the first ℓ powers of $\boldsymbol{y}$ and will help to estimate the decoding radius of the algorithm. We recall though that it is not a necessary condition for the algorithm to work (see $(*)$ test for $\ell = 3$ in §5).

Assumption 4. We assume that $\deg(F + \ell G) < n$;

Observe that, since $\deg G > 0$, by Assumption 4 we have $\deg(F+iG) < n$ for any $i = 1, \ldots, \ell$. Therefore, for any $i = 1, \ldots, \ell$ there exists $Z_i \subseteq L(F + iG)$ such that the following equalities hold

$$\begin{aligned} L(F + G') &= L(F + G) \oplus L(F + G' - D) \oplus Z_1 \\ L(F + 2G') &= L(F + 2G) \oplus L(F + 2G' - D) \oplus Z_2 \\ \ldots &= \ldots \\ L(F + \ell G') &= L(F + \ell G) \oplus L(F + \ell G' - D) \oplus Z_\ell. \end{aligned}$$

Lemma 4.1. *Given $Z_i \subset L(F + iG')$ such that $L(F + iG') = L(F + iG) \oplus L(F + iG' - D) \oplus Z_i$, then*

$$\dim Z_i = \deg(D - F - G) + g - 1.$$

Proof. The proof is an easy generalisation of the one for $\ell = 2$ (see Lemma 3.4). □

For any $i = 1, \ldots, \ell$, we define $f_{\boldsymbol{y}^i} \stackrel{\text{def}}{=} f^i_{\boldsymbol{y}} \in L(iG')$ and the map

$$\delta_{\boldsymbol{y}^i} : \left\{ \begin{array}{ccc} L(F) & \longrightarrow & L(F + iG') \\ \Lambda & \longmapsto & \Lambda f_{\boldsymbol{y}^i}. \end{array} \right.$$

4.2 The algorithm

It is possible now to define for any $i = 1, \ldots, \ell$ the space

$$S_i(F) \stackrel{\text{def}}{=} \{f \in L(F) \mid \delta_{\boldsymbol{y}^2}(f) \in L(F + iG) \oplus L(F + iG' - D)\}. \tag{26}$$

Remark 4.2. Again, we have $\mathrm{ev}_{\mathcal{P}}(S_i(F)) \subseteq K^{(i)}_{\boldsymbol{y}}$, where, given A, B as in (12),

$$K^{(i)}_{\boldsymbol{y}} = \{\boldsymbol{a} \in A \mid \langle \boldsymbol{a} * \boldsymbol{y}, \boldsymbol{b} \rangle = 0 \ \forall \boldsymbol{b} \in (B^{\perp} * C^{i-1})^{\perp}\}. \tag{27}$$

Furthermore, it is possible to generalise Theorem 3.5 to every $i \leq \ell$.

Theorem 4.3. *Given $S_i(F)$ as in (26) and $K^{(i)}_{\boldsymbol{y}}$ as in (27), if $\mathrm{supp}(F) \cap \mathcal{P} = \emptyset$ and $\deg G \geq 2g$, we have*

$$\mathrm{ev}_{\mathcal{P}}(S_i(F)) = K^{(i)}_{\boldsymbol{y}}.$$

Proof. The proof is exactly the same as for $\ell = 2$. See Appendix B. □

Proposition 4.4. *For any $i \leq \ell$, we have $L(F - D_{\boldsymbol{e}}) \subseteq S_i(F)$.*

Proof. The proof is an easy adaptation of the proof of Proposition 3.6. Let $\Lambda \in L(F - D_{\boldsymbol{e}})$ and $i \leq \ell$. We know that $\delta_{\boldsymbol{y}^i}(\Lambda) = \Lambda f^i_{\boldsymbol{y}} = \sum_{j=0}^{i} \binom{i}{j} \Lambda f^j_{\boldsymbol{c}} f^{i-j}_{\boldsymbol{e}}$. Now, we treat the term with $j = i$ and separately the others with $j < i$:

$j = i$: $(\Lambda f^i_{\boldsymbol{c}}) \geq -F - iG$

$j < i$: $(\Lambda f^j_{\boldsymbol{c}} f^{i-j}_{\boldsymbol{e}}) \geq -F + D_{\boldsymbol{e}} - jG - (i-j)G' + (i-j)(D - D_{\boldsymbol{e}}) \geq -F - iG' + D$

that is $\delta_{\boldsymbol{y}^i}(\Lambda) \in L(F + iG) \oplus L(F + iG' - D)$ and $\Lambda \in S_i(F)$. □

It is now possible to define the space $S(F)$ for this choice of ℓ:

$$S(F) \stackrel{\text{def}}{=} \bigcap_{i=1}^{\ell} S_i(F). \tag{28}$$

Thanks to Proposition 4.4, we have $L(F - D_{\boldsymbol{e}}) \subseteq S(F)$. We want now to close the gap between the two spaces. One can observe that Lemma 3.7, Lemma 3.8 and Proposition 3.9 can be generalised straightforwardly to the $S(F)$ defined in (28). In particular Lemma 3.8 changes in the following way.

Lemma 4.5. *If $L(F - D_{\boldsymbol{e}}) \neq \{0\}$, then there are at most g rational points $P \in \mathrm{supp}(D_{\boldsymbol{e}})$ such that*

$$S_1(F-P) \cap \bigcap_{i=2}^{\ell} S_i(F) = S(F) \cap L(F-P).$$

By Proposition 3.9, it is possible then to build a sequence $F_0 = F, F_1, F_2, \ldots$ such that for any $j \geq 0$

$$\dim S(F_{j+1}) \leq \dim S(F_j) - 2. \tag{29}$$

As for the case $\ell = 2$, among the following inclusions,

$$S(F_j - P) = \bigcap_{i=1}^{\ell} S_i(F_j - P) \subseteq S_1(F_j - P) \cap \bigcap_{i=1}^{\ell} S_i(F_j) \subsetneq S(F) \cap L(F_j - P) \subsetneq S(F),$$

the last two are the ones that give the gap in (29) and entail then $\Delta_{j+1} \leq \Delta_j - 1$. In order for the sequence $\{\Delta_j\}_j$ to decrease faster, we need more strict inclusions between $S(F_j - P)$ and $S(F_j)$.

Empirical Behavior: as in the case $\ell = 2$, we observed that the dimension of $S(F_j)$ decreases faster than expected when a random error vector is considered. In particular we got

$$\dim S(F_j - P) \leq \dim S(F_j) - (\ell + 1) \tag{30}$$

which implies $\Delta_{j+1} \leq \Delta_j - \ell$. The further $\ell - 1$ strict inclusions which cause this drop in dimension are the following

$$S(F_j-P) \subsetneq \bigcap_{i=1}^{\ell-1} S_i(F_j-P) \cap S_\ell(F_j) \subsetneq \bigcap_{i=1}^{\ell-2} S_i(F_j-P) \cap S_{\ell-1}(F_j) \cap S_\ell(F_j) \subsetneq \cdots \subsetneq S_1(F_j-P) \cap \bigcap_{i=2}^{\ell} S_i(F_j).$$

It is now possible to compute the decoding radius of the algorithm for $\ell \geq 2$. To do so, we generalise Theorem 3.11.

Theorem 4.6. *Assume $\deg F = t + 2g$ with $t \leq d^* - 3g - 1$. If the error vector is such that $S(F)$ verifies the empirical behavior and $\deg(F + \ell G) < n$, then a necessary condition for the algorithm to work is*

$$t \leq \frac{2\ell n - \ell(\ell+1)\deg G - 2\ell}{2(\ell+1)}. \tag{31}$$

Proof. The proof is almost the same as in the case with $\ell = 2$. The only difference consists in the necessary condition to fill the gap between $S(F)$ and $L(F - D_{\boldsymbol{e}})$ in g steps. Indeed we impose here the condition

$$\Delta_0 \leq \ell g$$

instead of $\Delta_0 \leq 2g$. An estimate for the dimension of $S(F)$ can be deduced by (26) and is

$$\dim S(F) \geq \ell(F) - \sum_{i=1}^{\ell} \dim Z_i,$$

where $\dim Z_i$ is given by Lemma 4.1. □

5 Some experimentations

In this section we first propose some guidelines on the parameters of the algorithm for $\ell = 2, 3$ and then we give some experimental observations from the tests we made.

5.1 The parameters

In order to test the algorithm with the right parameters, we need the genus g of the curve, the number of evaluation points n and the degree of the divisor G to fulfill several conditions:

(*i*) $t \leq \frac{2\ell n - \ell(\ell+1)\deg G - 2\ell}{2(\ell+1)}$ (decoding radius (31))

(*ii*) $\deg(F + \ell G) < n$ (Assumption 3)

(*iii*) $\deg F \leq d^* - g - 1$ (hypothesis in Theorem 3.11)

(*iv*) $\frac{2(\ell-1)n - \ell(\ell-1)\deg G - 2(\ell-1)}{2\ell} < \frac{2\ell n - \ell(\ell+1)\deg G - 2\ell}{2(\ell+1)}$ (decoding radius$(\ell-1)$ < decoding radius(ℓ))

(*v*) $\deg G \geq g - 1$

First, notice that Theorem 3.11 holds for all F with $t + 2g \leq \deg F \leq d^* - g - 1$. Here, we will just study the parameters for $\deg F = t + 2g$. Moreover, we want to be able to run the algorithm up to its decoding radius, hence we set

$$t = \frac{2\ell n - \ell(\ell+1)\deg G - 2\ell}{2(\ell+1)}.$$

By imposing these conditions on $\deg F$ and t, and developing (*iv*), (*i-iv*) become:

(*i*) $t = \frac{2\ell n - \ell(\ell+1)\deg G - 2\ell}{2(\ell+1)}$

(*ii*) $2n - \ell(\ell+1)\deg G - 4g(\ell+1) \geq 0$

(*iii*) $2n + (\ell-2)(\ell+1)\deg G - 6g\ell - 6g - 2 \geq 0$

(*iv*) $2n - \ell(\ell+1)\deg G - 2(\ell^2+\ell+1) \geq 0.$

In particular, notice that (*ii*) implies (*iii*) and (*iv*) when $g > \frac{l}{2}$ and $\deg G \geq 0$. That means that for $\ell = 2$ we can run the algorithm on codes which fulfill:

$$g - 1 \leq \deg G \leq \frac{n - 6g}{3},$$

while for $\ell = 3$ we need our code to satisfy

$$g - 1 \leq \deg G \leq \frac{n - 8g}{6}.$$

5.2 Some tests

In Table 1 and Table 2 there are listed some results about the algorithm's behavior with $\ell = 2, 3$. We worked with the following three curves:

1. $X^6 + Y^6 + XZ^5 = 0$ on $\mathbb{F}_{7^3}$;
2. $X^8 - YZ^7 - ZY^7 = 0$ on $\mathbb{F}_{7^2}$;
3. $ZY^5 - X^6 - XZ^5 - Z^6 = 0$ on $\mathbb{F}_{11^3}$;

Looking at Table 1 and Table 2, if $C = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, G)$ is the code we are running the algorithm on, we indicate by q the cardinality of the field, $\mathcal{X}$ the curve, g the genus of $\mathcal{X}$, n the length of the code, that is $n = |\mathcal{P}|$, $\deg G$ the degree of the divisor G. Moreover we list the values of half the designed distance of the code (column $\frac{d^*-1}{2}$) and of the decoding radius of Sudan algorithm (column "Sudan"), in order to compare them with the decoding radius of the new algorithm ("dec.radius"). For each test, we pick randomly an error vector $\boldsymbol{e}$ with $\mathrm{w}(\boldsymbol{e}) = t$ for a specific t and run the algorithm on $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$ with a random $\boldsymbol{c} \in C$ and with a power parameter ℓ. The value of t will be underlined when the decoding radius of the new algorithm is exceeded. We denote by "pts" the set of points $\{P_{i_j}\} \subseteq \mathrm{supp}(D)$, such that

$$\dim S(F_{j-1} - P_{i_j}) \leq \dim S(F_{j-1}) - 2.$$

In particular, for any test, we check if the points which guarantee this gap in the dimension belong to the support of $D_{\boldsymbol{e}}$. We recall that $\Delta_0 = \dim S(F) - \ell(F - D_{\boldsymbol{e}})$, where F is the initial divisor with $\deg F = t + 2g$ and that, if the algorithm verifies the empirical behavior (30), then $\Delta_{i+1} - \Delta_i \leq \ell$. Finally in Table 1 we list the tests where the algorithm succeeds, while in Table 2 there are some cases where the algorithm fails.

ℓ	q	$\mathcal{X}$	g	n	$\deg G$	$\frac{d^*-1}{2}$	Sudan	dec. radius	t	pts$\subseteq D_{\boldsymbol{e}}$	Δ_0	$\Delta_{i+1} - \Delta_i$
2	7^3	1	10	200	$2g-1$	90	107	113	113	true	18	2
2	7^3	1	10	200	$\frac{n-6g-1}{3}$	76	80	86	86	true	18	2
2	7^2	2	21	230	$2g-1$	94	98	111	111	false	40	2
3	7^3	1	10	200	$2g-2$	90	113	120	120	true	27	3
3	7^3	1	10	200	$\frac{n-8g-1}{6} - 1$	90	115	122	122	false	29	3
2	7^3	1	10	200	50 $(*)$	72	76	82	82	false	18	2
3	7^3	1	10	200	$\frac{n-1}{6} - 3$ $(*)$	84	97	104	104	true	24	$2, 3 \times 4$
2	11^3	3	10	200	$\frac{n-6g-1}{3} - 10$	81	90	96	96	false	18	2

Table 1: Tests on the algorithm for $\ell = 2$, $\ell = 3$.

Comments: First we want to point out that, whenever the parameter are chosen to satisfy (*i-iv*), then $\Delta_0 \leq \ell g$. That was not free, as we recall the decoding radius bound was a necessary condition to have $\Delta_0 \leq \ell g$ and not a sufficient one. Furthermore, we can see that it is actually possible to correct up to the decoding radius, which is then larger than Sudan decoding radius. In particular the gap Δ_0 reduces by expected at every step by ℓ for both $\ell = 2, 3$, that is the algorithm satisfies the hypothesis of empirical behavior (30) and we get to have $\Delta_j = 0$ and $L(F_j - D_{\boldsymbol{e}}) \neq \{0\}$ for some $j \leq g$. One can observe that pts is not always contained in the support of $D_{\boldsymbol{e}}$. Moreover it is really difficult to find a point which does not fulfill

$$\dim S(F_i - P) \leq \dim S(F_i) - 2, \tag{32}$$

and that once a point P which does fulfill (32) is found for the first step, it will satisfy it also for the next steps, that is, in our sequence $\{F_j\}_j$ we have $F_j = F - jP$ for every $j \leq g$. Finally, observe that the cases with the symbol $(*)$ are the only cases where not all the bounds (*i-iv*) hold. In particular we have $\deg(F + \ell G) \geq n$, that is, it is no longer sure that the spaces

$$L(F + \ell G), \qquad L(F + \ell G' - D)$$

are in direct sum. In this situation it is possible to use the following modified notion of $S_\ell(F)$

$$S_\ell(F) = \{f \in L(F) \mid \delta_{\boldsymbol{y}^\ell}(f) \in L(F + \ell G) + L(F + \ell G' - D)\}.$$

In the $(*)$ case with $\ell = 2$, we observed that actually the two spaces are still in direct sum and the gap Δ_i decreases by ℓ. Hence the algorithm works up to the decoding radius. In the $(*)$ case with $\ell = 3$, the algorithm works anyway even if Δ_i does not decrease by $\ell = 3$ at every step, but most of the times by 2. That is, almost at every step we have $\dim S(F - P) \leq \dim S(F) - 3$, where the inclusion which is not strict is the first from the left in the following sequence

$$S(F - P) \subseteq S_1(F - P) \cap S_2(F - P) \cap S_3(F) \subsetneq S_1(F - P) \cap S_2(F) \cap S_3(F) \subsetneq S(F) \cap L(F - P) \subsetneq S(F).$$

5.3 Failure cases

ℓ	q	$\mathcal{X}$	g	n	$\deg G$	$\frac{d^*-1}{2}$	Sudan	dec. radius	t	pts$\subseteq D_{\boldsymbol{e}}$	Δ_0	$\Delta_{i+1} - \Delta_i$
2	7^3	1	10	200	$2g-1$	90	107	113	<u>114</u>	true	21	2
2	7^2	2	21	230	$2g-1$	94	98	111	<u>112</u>	false	43	2
2	11^3	3	10	200	$\frac{n-6g-1}{3}$	76	80	86	86	false	14	1
3	7^3	1	10	200	25	67	104	111	96	false	25	1

Table 2: Failure cases.

Comments: We report here four cases where the algorithm does not work. Actually one should not consider all of them as failure cases, as the amount of error exceeds the decoding radius of the algorithm in the first two of them. One can see that in these situations, $\Delta_0 > \ell g$. Hence, although the gaps $\Delta_{i+1} - \Delta_i$ are the good ones, by the time $\Delta_i = 0$ we have $L(F_i - D_{\boldsymbol{e}}) = \{0\}$ as well and the algorithm fails as expected. In the two last cases all parameters are bounded as requested for the algorithm to work, but unlike the other tests, here the choice of the error vector is not random. Indeed it has been chosen in order to have two solutions $\boldsymbol{c}_1, \boldsymbol{c}_2 \in C$ such that

$$\mathrm{d}(\boldsymbol{y}, \boldsymbol{c}_1) = \mathrm{d}(\boldsymbol{y}, \boldsymbol{c}_2) = t.$$

In these cases, the empirical behavior (30) is not fulfilled, indeed Δ_i decreases only by 1 and no point in $\mathcal{P}$ can make Δ_i decrease faster. In particular here we only have two strict inclusions given by Proposition 3.9 and the following chain of equalities

$$S(F_j - P) = \bigcap_{i=1}^{\ell-1} S_i(F_j - P) \cap S_\ell(F_j) = \bigcap_{i=1}^{\ell-2} S_i(F_j - P) \cap S_{\ell-1}(F_j) \cap S_\ell(F_j) = \cdots = S_1(F_j - 1) \cap \bigcap_{i=2}^{\ell} S_i(F_j).$$

Hence, even if $\Delta_0 \leq \ell g$, g steps are not enough to find $S(F_i) = L(F_i - D_{\boldsymbol{e}})$.

References

[BH08] Peter Beelen and Tom Høholdt. The decoding of algebraic geometry codes. In *Advances in algebraic geometry codes*, volume 5 of *Ser. Coding Theory Cryptol.*, pages 49–98. World Sci. Publ., Hackensack, NJ, 2008.

[CMCP17] Alain Couvreur, Irene Márquez-Corbella, and Ruud Pellikaan. Cryptanalysis of McEliece Cryptosystem Based on Algebraic Geometry Codes and Their Subcodes. *IEEE Trans. Inform. Theory*, 63(8):5404–5418, August 2017.

[CP20] Alain Couvreur and Isabella Panaccione. Power Error Locating Pairs. *Designs, Codes and Cryptography*, 88(8):1561–1593, August 2020.

[Duu93] Iwan M. Duursma. Algebraic decoding using special divisors. *IEEE Transactions on Information Theory*, 39(2):694–698, 1993.

[Ehr92] Dirk Ehrhard. Decoding algebraic-geometric codes by solving a key equation. In Henning Stichtenoth and Michael A. Tsfasman, editors, *Coding Theory and Algebraic Geometry*, pages 18–25, Berlin, Heidelberg, 1992.

[Ehr93] Dirk Ehrhard. Achieving the designed error capacity in decoding algebraic-geometric codes. *IEEE Transactions on Information Theory*, 39(3):743–751, 1993.

[FR93] G. L. Feng and T. R. N. Rao. Decoding algebraic-geometric codes up to the designed minimum distance. *IEEE Transactions on Information Theory*, 39(1):37–45, 1993.

[Gop81] V. D. Goppa. Codes on algebraic curves. *Dokl. Akad. Nauk SSSR*, 259:1289–1290, 1981.

[JLJ+89] J. Justesen, K. J. Larsen, H. E. Jensen, A. Havemose, and T. Høholdt. Construction and decoding of a class of algebraic geometry codes. *IEEE Transactions on Information Theory*, 35(4):811–821, 1989.

[JM96] Heeralal Janwa and Oscar Moreno. McEliece public key cryptosystem using algebraic-geometry codes. *Designs, Codes and Cryptography*, 8:293–307, 1996.

[Köt92] Ralf Kötter. A unified description of an error locating procedure for linear codes. In *Proceedings Algebraic and Combinatorial Coding Theory III*, pages 113–117. Hermes, 1992.

[Mum70] David Mumford. Varieties defined by quadratic equations. In *Questions on algebraic varieties, C.I.M.E., III Ciclo, Varenna, 1969*, pages 29–100. Edizioni Cremonese, Rome, 1970.

[Pel89] Ruud Pellikaan. On a decoding algorithm for codes on maximal curves. *Information Theory, IEEE Transactions on*, 35:1228 – 1232, December 1989.

[Pel92] Ruud Pellikaan. On decoding by error location and dependent sets of error positions. *Discrete Math.*, 106–107:369–381, 1992.

[Por88] Sidney C. Porter. *Decoding codes arising from Goppa's construction on algebraic curves.* PhD thesis, Yale Univ., December 1988.

[PSP92] Sidney C. Porter, B.Z. Shen, and Ruud Pellikaan. Decoding geometric goppa codes using an extra place. *IEEE Transactions on Information Theory*, 38(6):1663–1676, 1992.

[RnN15] Johan Rosenkilde (né Nielsen). Power Decoding of Reed–Solomon Codes Revisited. In *Coding Theory and Applications*, pages 297–305, Cham, 2015. Springer International Publishing.

[SSB10] Georg Schmidt, Vladimir R. Sidorenko, and Martin Bossert. Syndrome Decoding of Reed–Solomon Codes Beyond Half the Minimum Distance Based on Shift-Register Synthesis. *IEEE Trans. Inform. Theory*, 56(10):5245–5252, October 2010.

[Sti09] Henning Stichtenoth. *Algebraic Function Fields and Codes.* Springer Publishing Company, Incorporated, 2nd edition, 2009.

[Sud97] Madhu Sudan. Decoding of Reed–Solomon Codes beyond the Error-Correction Bound. *J. Complexity*, 13(1):180–193, 1997.

[SV90] Alexei Skorobogatov and Serge Vlăduţ. On the decoding of algebraic-geometric codes. *Information Theory, IEEE Transactions on*, 36:1051 – 1060, October 1990.

[SW99] M. Amin Shokrollahi and Hal Wasserman. List decoding of algebraic-geometric codes. *IEEE Trans. Inform. Theory*, 45(2):432–437, March 1999.

[TVN07] Michael Tsfasman, Serge Vlăduţ, and Dmitrii Nogin. *Algebraic Geometric Codes: Basic Notions.* January 2007.

[TVZ82] M. Tsfasman, S. G. Vlădut, and T. Zink. Modular curves, Shimura curves, and Goppa codes, better than Varshamov-Gilbert bound. *Mathematische Nachrichten*, 109:21–28, 1982.

[Vlă90] Serge Vlădut. On the decoding of algebraic-geometric codes over $\mathbb{F}_q$ for $q \geq 16$. *IEEE Transactions on Information Theory*, 36(6):1461–1463, 1990.

A On the decoding radius of Sudan algorithm

This section mainly comes from the ideas of Peter Beelen and shows how to get an improved decoding radius for Sudan algorithm with respect to [BH08, §2.6]. This improvement mainly consists in analysing the parameters of the linear system to get Sudan polynomial Q. Given a curve of genus g, we consider a code $C = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, G)$, where G is a divisor with $2g - 2 < \deg G < n$ and $\mathcal{P} = \{P_1, \ldots, P_n\}$. Let us suppose a vector $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$ is given, where $\mathrm{w}(\boldsymbol{e}) = t$ and there exists $f \in L(G)$ such that

$$\boldsymbol{c} = (f(P_1), \ldots, f(P_n)). \tag{33}$$

We denote by I the support of the error vector $I = \mathrm{supp}(\boldsymbol{e})$ (in particular $|I| = t$). Let F be a divisor with $\deg F = n - t - 1$.

Original problem (Sudan): given $\ell \geq 1$, find a polynomial $Q(\boldsymbol{x}, y) = Q_0(\boldsymbol{x}) + Q_1(\boldsymbol{x})y + \cdots + Q_\ell(\boldsymbol{x})y^\ell$ such that

(*i*) $Q_i(\boldsymbol{x}) \in L(F - iG)$ for all $i = 0, \ldots, \ell$

(*ii*) $Q(P_j, y_j) = 0$ for all $j = 1, \ldots, n$.

This problem can be solved with a linear system of n equations in $\sum_{i=0}^{\ell} \ell(F - iG)$ unknowns. Hence the system has nonzero solutions if

$$t \leq \frac{2n\ell - \ell(\ell+1)\deg(G) - 2}{2(\ell+1)} - g. \tag{34}$$

Now we want to show that this decoding radius can be actually optimised. To do so, we consider the following problem.

Modified problem (Sudan): Given f as in (33), find a polynomial

$$Q(\boldsymbol{x}, y) = (y - f(\boldsymbol{x}))(\tilde{Q}_0(\boldsymbol{x}) + \tilde{Q}_1(\boldsymbol{x})y + \cdots + \tilde{Q}_{\ell-1}(\boldsymbol{x})y^{\ell-1}), \tag{35}$$

such that, if we denote by $\tilde{Q}(\boldsymbol{x}, y)$ the factor $\tilde{Q}_0(\boldsymbol{x}) + \tilde{Q}_1(\boldsymbol{x})y + \cdots + \tilde{Q}_{\ell-1}(\boldsymbol{x})y^{\ell-1}$,

(i') $\tilde{Q}_i(\boldsymbol{x}) \in L(F - (i+1)G)$ for all $i = 0, \ldots, \ell - 1$

(i") $\tilde{Q}(P_j, y_j) = 0$ for all $j \in I$.

It is clear that if the modified problem has a solution, then the original problem has one too. This problem can be solved, as the previous one, by a linear system. This time, we have a system of t equations in $\sum_{i=1}^{\ell} \ell(F - iG)$ unknowns. Therefore it admits nonzero solutions if

$$t \leq \frac{2n\ell - \ell(\ell+1)\deg(G) - 2}{2(\ell+1)} - \frac{\ell g}{\ell+1}. \tag{36}$$

B Some technical results

Most of the proofs presented in this appendix, are straightforward adaptations of proofs of [Ehr92] and [Ehr93] to $S_2(F)$ or to the language of functions rather than differentials, but we decided to report them here for sake of completeness.

Proposition B.1. *[Ehr92, Proposition 1](Function version) If* $\deg F + t < d^*$*, then* $L(F - D_{\boldsymbol{e}}) = S(F)$.

Proof. We recall that we consider here the $S(F)$ defined in (5). We already know that $L(F - D_{\boldsymbol{e}}) \subseteq S(F)$. Hence we consider now $\Lambda \in S(F)$ and we want to show that $\Lambda \in L(F - D_{\boldsymbol{e}})$. In order to do so, we first prove that $\Lambda f_{\boldsymbol{e}} \in L(F + G' - D)$. Since $\Lambda \in S(F)$, there exist $g \in L(F + G)$ and $h \in L(F + G' - D)$ such that $\Lambda f_{\boldsymbol{y}} = g + h$. Furthermore $f_{\boldsymbol{y}} = f_{\boldsymbol{c}} + f_{\boldsymbol{e}}$, hence

$$\Lambda f_{\boldsymbol{c}} - g = h - \Lambda f_{\boldsymbol{e}}.$$

By way of contradiction let us suppose that $h \neq \Lambda f_{\boldsymbol{e}}$. Since $f_{\boldsymbol{e}} \in L(G' - D + D_{\boldsymbol{e}})$, we have

$$\begin{aligned}(h - \Lambda f_{\boldsymbol{e}}) &\geq \min(-F - G' + D, -F - G' + D - D_{\boldsymbol{e}}) = -F - G' + D - D_{\boldsymbol{e}} \\ (\Lambda f_{\boldsymbol{c}} - g) &\geq \min(-F - G, -F - G) = -F - G.\end{aligned}$$

Hence, in particular

$$(h - \Lambda f_{\boldsymbol{e}}) \geq \max(-F - G' + D - D_{\boldsymbol{e}}, -F - G) = -F - G + D - D_{\boldsymbol{e}},$$

that is $h - \Lambda f_{\boldsymbol{e}} \in L(F + G - D + D_{\boldsymbol{e}})$. Though, by hypothesis we have

$$\deg(F + G - D + D_{\boldsymbol{e}}) = \deg F + \deg G - n + t < 0,$$

that is $L(F + G - D + D_{\boldsymbol{e}}) = \{0\}$, which is a contradiction since we supposed $h \neq \Lambda f_{\boldsymbol{e}}$. Now we know that $\Lambda f_{\boldsymbol{e}} \in L(F + G' - D)$, we can conclude the proof. First, since $\mathrm{ev}_{\mathcal{P}}(f_{\boldsymbol{e}}) = \boldsymbol{e}$, we observe that if $P \in \mathrm{supp}(D_{\boldsymbol{e}})$, then

$$f_{\boldsymbol{e}} \in L(G' - D + D_{\boldsymbol{e}}) \setminus L(G' - D + D_{\boldsymbol{e}} - P).$$

In particular, for any $P \in \mathrm{supp}(D_{\boldsymbol{e}})$, since $\Lambda f_{\boldsymbol{e}} \in L(F + G' - D)$ and $v_P(f_{\boldsymbol{e}}) = 0$, we get

$$v_P(\Lambda) = v_P(\Lambda) + v_P(f_{\boldsymbol{e}}) = v_P(\Lambda f_{\boldsymbol{e}}) \geq -v_P(F) + 1,$$

that is $\Lambda \in L(F - D_{\boldsymbol{e}})$. □

Theorem B.2. *Given $S_2(F)$ as in (17) and $K_{\boldsymbol{y}}^{(2)}$ as in (14), if* $\operatorname{supp}(F) \cap \mathcal{P} = \emptyset$ *and* $\deg G \geq 2g$*, we have*

$$\operatorname{ev}_{\mathcal{P}}(S_2(F)) = K_{\boldsymbol{y}}^{(2)}.$$

In order to prove this theorem we need the following result.

Proposition B.3. *Let us consider G' as in §2.1, that is $G' \geq G$ and $L(W + D - G') = \{0\}$ and let φ be the map $\varphi : L(F + 2G') \longrightarrow \Omega(F + 2G - D)^{\vee}$ where, for any $f \in L(F + 2G')$, given $\omega \in \Omega(F + 2G - D)$,*

$$\varphi(f)(\omega) = \sum_{i=1}^{n} \operatorname{Res}_{P_i}(f\omega).$$

If $\operatorname{supp}(F) \cap \mathcal{P} = \emptyset$*, then $\varphi_{|Z_2} : Z_2 \to \Omega(F + 2G - D)^{\vee}$ is an isomorphism.*

Proof. This proof is an adaptation of the proof of Remark 3.1 given in [Ehr93]. It is composed by the following steps:

(1) φ is surjective;

(2) $L(F + 2G' - D) \oplus L(F + 2G) \subseteq \operatorname{Ker}(\varphi)$;

(3) $\dim(Z_2) = \dim L(W + D - 2G - F)$.

In order to prove that φ is surjective, we first show that it suffices to prove the surjectivity of the map $\tilde{\varphi} : L(F + 2G') \to (\mathbb{F}_q^n)^{\vee}$, where for any $h \in L(F + 2G')$, given $\boldsymbol{a} \in \mathbb{F}_q^n$,

$$\tilde{\varphi}(f)(\boldsymbol{a}) = \sum_{i=1}^{n} a_i \operatorname{ev}_{P_i}(f).$$

Let us suppose then that $\tilde{\varphi}$ is surjective. One can easily see that the map

$$\Psi : \left\{ \begin{array}{ccc} \Omega(F + 2G - D) & \longrightarrow & \mathbb{F}_q^n \\ \omega & \longmapsto & (\operatorname{Res}_{P_1}(\omega), \ldots, \operatorname{Res}_{P_n}(\omega)). \end{array} \right.$$

is injective, its kernel being the space $\Omega(F + 2G)$ which is equal to $\{0\}$ as

$$\deg(F + 2G) > 2g - 2.$$

Hence, Ψ being injective, its transpose $\Psi^T : (\mathbb{F}_q^n)^{\vee} \to \Omega(F + 2G - D)^{\vee}$ is surjective. By the hypothesis on the surjectivity of $\tilde{\varphi}$, the composition of $\tilde{\varphi}$ and Ψ^T, gives a surjective map. We claim that this map

$$L(F + 2G') \xrightarrow{\tilde{\varphi}} \!\!\!\rightarrow (\mathbb{F}_q^n)^{\vee} \xrightarrow{\Psi^T} \!\!\!\rightarrow \Omega(F + 2G - D)^{\vee}.$$

is equal to φ: for any $f \in L(F + 2G)$ and $\omega \in \Omega(F + 2G - D)$ the following equalities hold

$$\begin{aligned} \Psi^T(\tilde{\varphi}(f))(\omega) &= (\tilde{\varphi}(f) \circ \Psi)(\omega) \\ &= \tilde{\varphi}(f)(\operatorname{Res}_{\mathcal{P}}(\omega)) \\ &= \sum_{i=1}^{n} \operatorname{Res}_{P_i}(f\omega) = \varphi(f)(w). \end{aligned}$$

Hence, we now prove that $\tilde{\varphi}$ is surjective. Let us consider $(\boldsymbol{e}_i^{\vee})_i$ the canonical basis of $(\mathbb{F}_q^n)^{\vee}$. First we claim that for any $i = 1, \ldots, n$, the set

$$L(2G' + F - D + P_i) \setminus L(2G' + F - D) \neq \emptyset.$$

To see that, notice that as we proved in (16), we have $\Omega(F + 2G' - D + P_i) = \Omega(F + 2G' - D) = \{0\}$, and

$$\ell(2G' + F - D + P_i) = \deg(2G' - D + F + P_i) - g + 1 \tag{37}$$

$$\ell(2G' + F - D) = \deg(2G' + F - D) - g + 1, \tag{38}$$

hence $L(F+2G'-D+P_i)\setminus L(F+2G'-D)\neq\emptyset$. Now it suffices to note that for any h in this set, there is $\lambda\in\mathbb{F}_q^*$ such that $\tilde{\varphi}(h)=\lambda \boldsymbol{e}_i^\vee$, hence (1) is proved. Let us now consider $h\in L(F+2G'-D)$. Given $\omega\in\Omega(F+2G-D)$ we have $v_{P_i}(h\omega)\geq 0$ for any $i=1,\ldots,n$, hence $\mathrm{Res}_{\mathcal{P}}(h\omega)=0$ for any $\omega\in\Omega(F+2G-D)$, that is $\varphi(h)=0$. We consider now $h\in L(F+2G)$. For any $\omega\in\Omega(F+2G-D)$,

$$(h\omega)\geq -D,$$

thus we get $\varphi(h)(\omega)=\sum_{i=1}^n \mathrm{Res}_{P_i}(h\omega)=\sum_{P\in\mathcal{X}}\mathrm{Res}_P(h\omega)=0$ from the residue Theorem. Hence we proved (2). We now finally prove (3). We have

$$\begin{aligned}\dim Z_2 &= g-1-\deg(F+2G-D)\\ &= \dim\Omega(F+2G-D),\end{aligned}$$

where in the first equality we used Lemma 3.4, while in the second one, we use Assumption 3. □

Remark B.4. Observe that by (1) and Proposition 1.10, if $\deg G\geq 2g$,

$$\mathrm{Res}_{\mathcal{P}}(\Omega(F+2G-D))=C_\Omega(F+2G)=C_L(W+D-F-2G)=(B^\perp * C)^\perp,$$

where B is defined in (12).

We can now prove Theorem 3.5.

Proof. Let π_{Z_2} be the projection $L(F+2G')\to Z_2$ with respect to the decomposition of the space $L(F+2G')=L(F+2G)\oplus L(G+2G'-D)\oplus Z_2$. We then have

$$S_2=\{\Gamma\in L(F)\mid \pi_{Z_2}\circ\delta_{\boldsymbol{y}^2}(\Gamma)=0\}.$$

In particular, by Proposition B.3, for any $\Gamma\in L(F)$ we have

$$\Gamma\in S_2(F)\iff \pi_{Z_2}\circ\delta_{\boldsymbol{y}^2}(\Gamma)=0\iff \varphi_{|Z_2}\circ\pi_{Z_2}\circ\delta_{\boldsymbol{y}^2}(\Gamma)=0,$$

that is if and only if, for any $\omega\in\Omega(F+2G-D)$

$$(\varphi_{|Z_2}\circ\pi_{Z_2}\circ\delta_{\boldsymbol{y}^2}(\Gamma))(\omega)=0 \tag{39}$$

Note that, by Proposition B.3, $\varphi(L(F+2G)\oplus L(F+2G'-D))=0$, therefore for any $f\in L(F+2G')$ the following equality holds

$$\varphi(f)=(\varphi_{|Z_2}\circ\pi_{Z_2})(f).$$

Hence, the left hand side of the equation in (39) becomes

$$\begin{aligned}(\varphi_{|Z_2}\circ\pi_{Z_2}\circ\delta_{\boldsymbol{y}^2}(\Gamma))(\omega) &= \varphi(\delta_{\boldsymbol{y}^2}(\Gamma))(\omega)=\sum_{i=1}^n \mathrm{Res}_{P_i}(\omega\Gamma f_{\boldsymbol{y}^2})\\ &= \sum_{i=1}^n \Gamma(P_i)\,\mathrm{Res}_{P_i}(\omega)y_i^2\end{aligned}$$

By Remark B.4, we have

$$\{(\mathrm{Res}_{P_1}(\omega),\ldots,\mathrm{Res}_{P_n}(\omega))\mid\omega\in\Omega(F+2G-D)\}=(B^\perp * C)^\perp$$

where B is defined in (12). Hence, for any $\Gamma\in L(F)$, $\Gamma\in S_2(F)$ if and only if $\mathrm{ev}_D(\Gamma)\in K_{\boldsymbol{y}}^{(2)}$. □