CONSTRUCTION OF ASYMMETRIC CHUDNOVSKY ALGORITHMS WITHOUT DERIVATED EVALUATION FOR MULTIPLICATION IN FINITE FIELDS

Stéphane Ballet, Nicolas Baudru, Alexis Bonnecaze, Mila Tukumuli

CONSTRUCTION OF ASYMMETRIC CHUDNOVSKY ALGORITHMS WITHOUT DERIVATED EVALUATION FOR MULTIPLICATION IN FINITE FIELDS

STÉPHANE BALLET, NICOLAS BAUDRU, ALEXIS BONNECAZE, AND MILA TUKUMULI

Abstract.

The Chudnovsky and Chudnovsky algorithm for the multiplication in extensions of finite fields provides a bilinear complexity which is uniformly linear with respect to the degree of the extension. Recently, Randriambololona has generalized the method, allowing asymmetry in the interpolation procedure and leading to new upper bounds on the bilinear complexity. In this article, we first translate this generalization into the language of algebraic function fields. Then, we propose a strategy to effectively construct asymmetric algorithms using places of higher degrees and without derivated evaluation. Finally, we provide examples of three multiplication algorithms along with their Magma implementation: in $\mathbb{F}_{16^{13}}$ using only rational places, in $\mathbb{F}_{4^5}$ using also places of degree two, and in $\mathbb{F}_{2^5}$ using also places of degree four.

Keywords: Multiplication algorithm, bilinear complexity, interpolation on algebraic curve, finite field.

1. Introduction

Let q be a prime power, $\mathbb{F}_q$ the finite field with q elements and $\mathbb{F}_{q^n}$ the degree n extension of $\mathbb{F}_q$. Among all algorithms of multiplications in $\mathbb{F}_{q^n}$, those based on Chudnovsky-Chudnovsky method [18] are known to provide the lowest bilinear complexity. This method is based on interpolation on algebraic curves defined over a finite field and provides a bilinear complexity which is linear in n. The original algorithm (called here CCMA) uses only points of degree 1, with multiplicity 1. Ballet and Rolland [8, 10] and Arnaud [1] improved the algorithm introducing interpolation at points of higher degree or higher multiplicity. The symmetry of the CCMA construction involves 2-torsion points that represent an obstacle to the improvement of upper bilinear complexity bounds. To eliminate this difficulty, Randriambololona [23] allowed asymmetry in the interpolation procedure, and then Pieltant and Randriambololona [22] derived new bounds, uniform in q, of the bilinear complexity. Unlike symmetric constructions, no effective implementation of this asymmetric construction has been done yet. In this article, we construct explicitly such multiplication algorithms for finite extensions of finite fields.

1.1. **Multiplication algorithm and tensor rank.** Let q be a prime power, $\mathbb{F}_q$ the finite field with q elements and $\mathbb{F}_{q^n}$ the degree n extension of $\mathbb{F}_q$. The multiplication of two elements of $\mathbb{F}_{q^n}$ is an $\mathbb{F}_q$-bilinear application from $\mathbb{F}_{q^n} \times \mathbb{F}_{q^n}$ onto $\mathbb{F}_{q^n}$. Then it can be considered as an $\mathbb{F}_q$-linear application from the tensor product $\mathbb{F}_{q^n} \otimes_{\mathbb{F}_q} \mathbb{F}_{q^n}$ onto $\mathbb{F}_{q^n}$. Consequently, it can also be considered as a tensor T_m of ${\mathbb{F}_{q^n}}^\star \otimes_{\mathbb{F}_q} {\mathbb{F}_{q^n}}^\star \otimes_{\mathbb{F}_q} \mathbb{F}_{q^n}$ where $\star$ denotes the dual. When T_m is written

$$T_m = \sum_{i=1}^{r} x_i^\star \otimes y_i^\star \otimes c_i, \tag{1}$$

where the r elements $x_i^\star$ as well as the r elements $y_i^\star$ are in the dual ${\mathbb{F}_{q^n}}^\star$ of $\mathbb{F}_{q^n}$ while the r elements c_i are in $\mathbb{F}_{q^n}$, the following holds for any $x, y \in \mathbb{F}_{q^n}$:

$$x \cdot y = \sum_{i=1}^{r} x_i^\star(x) y_i^\star(y) c_i.$$

The decomposition (1) is not unique.

Definition 1.1. Every expression

$$x \cdot y = \sum_{i=1}^{r} x_i^\star(x) y_i^\star(y) c_i$$

defines a bilinear multiplication algorithm $\mathcal{U}$ of bilinear complexity $\mu(\mathcal{U}) = r$. Such an algorithm is said symmetric if $x_i = y_i$ for all i.

Definition 1.2. The minimal number of summands in a decomposition of the tensor T_m of the multiplication is called the bilinear complexity (resp. symmetric bilinear complexity) of the multiplication and is denoted by $\mu_q(n)$ (resp. $\mu_q^{sym}(n)$):

$$\mu_q(n) = \min_{\mathcal{U}} \mu(\mathcal{U})$$

where $\mathcal{U}$ is running over all bilinear multiplication algorithms (resp. all bilinear symmmetric multiplication algorithms) in $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$.

1.2. **Known results.** In 1979, Winograd proved [28] that optimal multiplication algorithms realizing the lowest bilinear bound belong to the class of interpolation algorithms. Then, in 1988, Chudnovsky and Chudnovsky introduced a method [18] to prove the linearity [4] of the bilinear complexity of the multiplication in finite extensions of a finite field. In doing so, they proposed the first known multiplication algorithm using interpolation on non trivial algebraic curves over $\mathbb{F}_q$. This original algorithm, that we call in this article CCMA, only uses points of degree 1, with multiplicity 1. Later, several studies focused on the qualitative improvement of this algorithm (for example [8, 1, 17]) allowing interpolation at points of higher degree, or with higher multiplicity. In parallel, improvements of upper bounds (for example [7, 12]) and asymptotic upper bounds (for example [26, 11]) of the bilinear complexity were obtained.

The first known effective finite field multiplication through interpolation on algebraic curves was proposed by Shokrollahi and Baum [14]. They used the Fermat curve $x^3+y^3=1$ to construct a multiplication algorithm over $\mathbb{F}_{4^4}$ with 8 bilinear multiplications. In [5], Ballet proposed one over $\mathbb{F}_{16^n}$ where $n \in [13, 14, 15]$, using the hyperelliptic curve $y^2 + y = x^5$ of genus 2, with $2n + 1$ bilinear multiplications. Notice that these aforementioned two algorithms only used rational points, with multiplicity 1. In 2009, Cenk and Özbudak proposed in [17] an explicit elliptic multiplication algorithm in $\mathbb{F}_{3^9}$ with 26 bilinear multiplications. To this end, they used the elliptic curve $y^2 = x^3 + x + 2$, combining the ideas of using points of higher degree and higher multiplicity. In fact, few studies have been devoted to the effective construction of Chudnovsky type algorithms, and in particular when the degree of extensions reaches cryptographic size. In 2013, Ballet et al. [3] detailed a multiplication algorithm in $\mathbb{F}_{3^{57}}$ with 234 bilinear multiplications using the elliptic curve $y^2+2x^3+2x^2+1=0$ with points of degree at most 4 and multiplicity at most 3. In 2015, Atighehchi et al. [2] proposed a model allowing parallel computation, in which an ingenious use of normal bases provides efficient algorithms for both multiplication and exponentiation. They detailed an implementation in the finite field $\mathbb{F}_{16^{13}}$ using the hyperelliptic curve $y^2 + y = x^5$ of genus 2. In 2012, Riandriambololona [23] introduced an asymmetric algorithm which generalizes the Chudnovsky algorithm and leads to better bounds, uniform in q, of the bilinearity. When the genus g of the curve is equal to 1, it is known [3] that an asymmetric algorithm can always be symmetrized (i.e. there always exists a symmetric version of an asymmetric algorithm). However, for greater values of g, it may not be the case. Thus, it is of interest to know an effective construction of this asymmetric algorithm. So far, no effective implementation has been proposed for such an algorithm. In this article, we detail a strategy to effectively construct asymmetric algorithms with higher degree but without derivated evaluation.

1.3. **Organization of the paper and new results.** In Section 1, we define the bilinear complexity of the multiplication in any extension of finite field and we recall the main known related results. In Section 2, we give an explicit translation of the generalization of CCMA given by Randriambololona [23, Theorem 3.5]. Then in Section 3, by defining a new design of this algorithm, we give a strategy of construction and implementation. In particular, thanks to a suitable representation of the Riemann-Roch spaces, we present the first construction of asymmetric effective algorithms of multiplication in finite fields. These algorithms are tailored to hardware implementation and they allow computations to be parallelized while maintaining a low number of bilinear multiplications. In Section 4, we give an analysis of the not asymptotical complexity of this algorithm. Finally, in Sections 5, 6, and 7, we give examples with the finite field $\mathbb{F}_{16^{13}}$ using only rational places, $\mathbb{F}_{4^5}$ using also places of degree two and $\mathbb{F}_{2^5}$ using also places of degree four.

2. Multiplication algorithms of type Chudnovsky : Generalization of Randriambololona

In this section we present a generalization of Chudnovsky type algorithms, introduced in [23, Theorem 3.5] by Randriambololona, which is possibly asymmetric. Since our aim is to describe explicitly the effective construction of this asymmetric algorithm, we transform the representation of this algorithm, initially made in the abstract geometrical language, in the more explicit language of algebraic function fields. A comprehensive course on algebraic function fields can be found in [27]. Only the elementary terminology used along this paper is introduced.

Let $F/\mathbb{F}_q$ be an algebraic function field over the finite field $\mathbb{F}_q$ of genus $g(F)$. We denote by $\mathbb{P}_1(F/\mathbb{F}_q)$ the set of places of degree one of the algebraic function field $F/\mathbb{F}_q$ and by $N_1(F/\mathbb{F}_q)$ its cardinality. If D is a divisor, $\mathcal{L}(D)$ denotes the Riemann-Roch space associated to D. We denote by $\mathcal{O}_Q$ the valuation ring of the place Q and by F_Q its residue class field $\mathcal{O}_Q/Q$ which is isomorphic to $\mathbb{F}_{q^{\deg Q}}$ where $\deg Q$ is the degree of the place Q.

In the framework of algebraic function fields, the result [23, Theorem 3.5] of Randriambololona can be stated as in Theorem 2.1. Note that we do not take into account derivated evaluations, since we are not

interested in asymptotic results. It means that we describe this asymmetric algorithm with the divisor $G = P_1 + \cdots + P_N$ where the P_i are pairwise disctinct closed points of degree $\deg P_i = d_i$.

Let us define the following Hadamard product in $\mathbb{F}_{q^{l_1}} \times \mathbb{F}_{q^{l_2}} \times \cdots \times \mathbb{F}_{q^{l_N}}$, where the l_i's denote positive integers, by $(u_1, \ldots, u_N) \odot (v_1, \ldots, v_N) = (u_1 v_1, \ldots, u_N v_N)$.

Theorem 2.1. *Let $F/\mathbb{F}_q$ be an algebraic function field of genus g over $\mathbb{F}_q$. Suppose there exists a place Q of degree n. Let $\mathcal{P} = \{P_1, \ldots, P_N\}$ be a set of N places of arbitrary degree not containing the place Q. Suppose there exist two effective divisors D_1, D_2 of $F/\mathbb{F}_q$ such that:*

(i) The place Q and the places of $\mathcal{P}$ are not in the support of the divisors D_1 and D_2.

(ii) The natural evaluation maps E_i for $i = 1, 2$ defined as

$$E_i \ : \ \left\{ \begin{array}{ccc} \mathcal{L}(D_i) & \longrightarrow & \mathbb{F}_{q^n} \simeq F_Q \\ f & \longmapsto & f(Q) \end{array} \right.$$

are surjective.

(iii) The natural evaluation map

$$T \ : \ \left\{ \begin{array}{ccc} \mathcal{L}(D_1 + D_2) & \longrightarrow & \mathbb{F}_{q^{\deg P_1}} \times \mathbb{F}_{q^{\deg P_2}} \times \cdots \times \mathbb{F}_{q^{\deg P_N}} \\ f & \longmapsto & (f(P_1), f(P_2), \ldots, f(P_N)) \end{array} \right.$$

is injective.

Then for any two elements x, y in $\mathbb{F}_{q^n}$, we have:

$$xy = E_Q \circ T^{-1}_{|Im\ T} \left(T \circ E_1^{-1}(x) \odot T \circ E_2^{-1}(y) \right),$$

where E_Q denotes the canonical projection from the valuation ring $\mathcal{O}_Q$ of the place Q in its residue class field F_Q, $\circ$ the standard composition map, $T^{-1}_{|Im\ T}$ the restriction of the inverse map of T on the image of T, E_i^{-1} the inverse map of the restriction of the map E_i on the quotient group $\mathcal{L}(D_i)/\ker E_i$ and $\odot$ the Hadamard product in $\mathbb{F}_{q^{\deg P_1}} \times \mathbb{F}_{q^{\deg P_2}} \times \cdots \times \mathbb{F}_{q^{\deg P_N}}$; and

$$\mu_q(n) \leq \sum_{i=1}^{N} \mu_q(\deg P_i).$$

Remark 2.2. The condition $D_1 = D_2$ (modulo the group of principal divisors) is not a sufficient condition to have a symmetric algorithm. Indeed, note that even if $D_1 = D_2$ then this algorithm can be not symmetric in the sens of Definition (1.1) if it uses places of degree strictly greater than one and if the bilinear multiplications in the residue class fields of these places are not computed (via the operation $\odot$) with a symmetric algorithm. However, for simplicity, we will say that an algorithm of type Chudnovsky is symmetric when $D_1 = D_2$ (modulo the group of principal divisors). Indeed, the interest of such an asymmetric algorithm is to avoid the problem of 2-torsion elements in the divisor class group of the algebraic function field $F/\mathbb{F}_q$ in order to have more flexibility in the choice of the algebraic function field $F/\mathbb{F}_q$ since the conditions (ii) and (iii) become easier to satisfy (cf. [23, Remark 3.7]).

Note also that in this presentation, we require that the divisors D_1 and D_2 are positive divisors in contrast to Theorem 3.5 in [23]. Indeed, in our context of effective construction, it is important to have $\mathcal{L}(D_i) \subseteq \mathcal{L}(D_1 + D_2)$, which is the case if the divisors D_1 and D_2 are positive divisors (cf. Remark 3.1).

Finally, for simplicity, we will use the same representation for $\mathbb{F}_{q^{\deg P_i}}$ and $\mathbb{F}_{q^{\deg P_j}}$ when $\deg P_i = \deg P_j$. Actually, we could only suppose that the products in $\mathbb{F}_{q^{\deg P_i}}$ and $\mathbb{F}_{q^{\deg P_j}}$ use bilinear multiplication algorithms having the same bilinear complexity.

3. Effective algorithm

3.1. **Method and strategy of implementation.** The construction of the algorithm is based on the choice of the place Q of degree n, the effective divisors D_1 and D_2 of degree $n + g - 1$, the bases of spaces $\mathcal{L}(D_1)$, $\mathcal{L}(D_2)$ and $\mathcal{L}(D_1 + D_2)$ and the basis of the residue class field F_Q of the place Q.

In practice, following the ideas of [4], we take as a divisor D_1 one place of degree $n + g - 1$. This has the advantage to solve both the problem of the support of divisor D_1 and the problem of the effectivity of the divisor D_1. For the same reasons, the divisor D_2 is also chosen as a place of degree $n + g - 1$. Furthermore, we require additional properties described below.

3.2. **Finding good places D_1, D_2 and Q.** In order to obtain the good places, we draw them at random and check that they satisfy the required conditions. We proceed as follows:

(1) We draw at random an irreducible polynomial $\mathcal{Q}(x)$ of degree n in $\mathbb{F}_q[X]$ and check that this polynomial is:
   (a) Primitive.

(b) Totally decomposed in the algebraic function field $F/\mathbb{F}_q$ (which implies that there exists a place Q of degree n above the polynomial $\mathcal{Q}(x)$).

(2) We choose a place Q of degree n among the places of $F/\mathbb{F}_q$ lying above the polynomial $\mathcal{Q}(x)$.

(3) We draw at random a place D_1 of degree $n+g-1$ and check that $D_1 - Q$ is a non-special divisor of degree $g-1$ i.e. $\dim \mathcal{L}(D_1 - Q) = 0$.

(4) We draw at random a place D_2 of degree $n+g-1$ and check that $D_2 - Q$ is a non-special divisor of degree $g-1$ i.e. $\dim(D_2 - Q) = 0$.

Remark 3.1. Clearly, our method relies on the existence of the places Q, D_1 and D_2 as defined above and such that $D_1 - Q$ and $D_2 - Q$ are non-special divisors of degree $g-1$.

A sufficient condition of existence of Q relies on the existence of at least one place of degree n given by the following inequality [27, Corollary V.2.10 (c)]:

$$2g+1 \leq q^{\frac{n-1}{2}}\left(q^{\frac{1}{2}}-1\right).$$

Then, we are sure of the existence of a non-special divisor of degree $g-1$ when $q \geq 4$ [6]. The larger q is, the larger the probability to draw a non-special divisor of degree $g-1$ becomes (Proposition 5.1 [13]), but not necessarily as a difference of two places: this is an open problem. However, looking for non-special divisors of degree $g-1$ as a difference of two places has many advantages.

Most of all, this solves easily the problem of the support of divisors D_1 and D_2 (condition (i) of Theorem 2.1) as well as the problem of the effectivity of these divisors. Indeed, in our context of construction, it is important to have $\mathcal{L}(D_i) \subseteq \mathcal{L}(D_1 + D_2)$, which is the case if the divisors D_1 and D_2 are effective divisors. However, the property to have simultaneously an effective divisor (with the required properties) without having given places in its support is difficult to obtain theoretically because the method of the support moving (cf. [21]), which is a direct consequence of Strong Approximation Theorem (cf. [27, Proof of Theorem I.6.4]), has the drawback to imply the loss of effectivity.

Furthermore, in practice, it is easy to find Q and the divisors D_i satisfying the required properties since there exist many such places in our context. However, it is not true in the general case. For instance, this fails when we consider an elliptic curve with only one rational point since for any elliptic curve, there exists a non-special divisor of degree $g-1=0$ if and only if the divisor class number h is >1, i.e. $N_1 \geq 2$ (cf. [6, Section 3.2]).

3.3. **Choosing good bases of the spaces.**

3.3.1. *The residue field F_Q.* In Subsection 3.2, we chose a place Q of degree n lying above a primitive polynomial $\mathcal{Q}(x)$ in $\mathbb{F}_q[X]$. Then we identify the residue class field F_Q with $\mathbb{F}_{q^n}$ and set as representation basis of $\mathbb{F}_{q^n}$ the canonical basis $\mathcal{B}_Q = (1, \alpha, \alpha^2, ..., \alpha^{n-1})$ generated by a root α of $\mathcal{Q}(x)$. Note that if we wish to use a normal basis as a representation basis, it is convenient to find a place Q above a normal polynomial $\mathcal{Q}(x)$ and to make a change of basis between the canonical basis and the normal basis $(\alpha, \alpha^q, \alpha^{q^2}..., \alpha^{q^{n-1}})$. However, even in this case, it is necessary in our algorithm to preserve the canonical basis as basis of the residue field F_Q because we need to have the constant component in the bases of the Riemann-Roch spaces $\mathcal{L}(D_i)$ for $i \in \{1,2\}$ (cf. Section 3.3.2). From now on, we identify $\mathbb{F}_{q^n}$ to F_Q, as the residue class field F_Q of the place Q is isomorphic to the finite field $\mathbb{F}_{q^n}$.

3.3.2. *The Riemann-Roch spaces $\mathcal{L}(D_1)$ and $\mathcal{L}(D_2)$.* Clearly, the choice of D_i, $i \in \{1,2\}$ and Q of Section 3.2 implies that the maps E_i of Theorem 2.1 are isomorphisms, since $\deg(D_i) = n+g-1$, $\dim \mathcal{L}(D_i - Q) = 0$ and $\mathcal{L}(D_i - Q) = Ker(E_i)$. Thereby, we choose as basis of $\mathcal{L}(D_i)$ the reciprocal image $\mathcal{B}_{D_i}$ of the basis $\mathcal{B}_Q = (\phi_1, \ldots, \phi_n)$ of F_Q by the evaluation map E_i, namely $\mathcal{B}_{D_i} = (E_i^{-1}(\phi_i), \ldots, E_i^{-1}(\phi_n))$. Note that by Section 3.3.1, the choice of the basis of the residue field F_Q implies that $\phi_1 = 1$ and so $E_1^{-1}(1) = E_2^{-1}(1) = 1$. Let us denote $\mathcal{B}_{D_i} = (f_{i,1}, ..., f_{i,n})$ with $f_{i,1} = 1$ for $i = 1, 2$.

3.3.3. *The Riemann-Roch space $\mathcal{L}(D_1 + D_2)$.* Note that since D_1 and D_2 are effective divisors, we have $\mathcal{L}(D_1) \subset \mathcal{L}(D_1 + D_2)$ and $\mathcal{L}(D_2) \subset \mathcal{L}(D_1 + D_2)$.

Lemma 3.2. *Let D_1 and D_2 be two effective divisors with disjoint supports. Then*

$$\mathcal{L}(D_1) \cap \mathcal{L}(D_2) = \mathbb{F}_q.$$

Proof. It is clear that $\mathbb{F}_q \subset \mathcal{L}(D_1) \cap \mathcal{L}(D_2)$ because the divisors are effective. Suppose that the function $f \in \mathcal{L}(D_1) \cap \mathcal{L}(D_2)$ is such that $f \notin \mathbb{F}_q$. Then there exist P_1 in the support of D_1 and P_2 in the support of D_2 such that $v_{P_1}(f) \leq -1$ and $v_{P_2}(f) \leq -1$. But then the function f admits a pole of order at least 1 at P_1 and a pole of order at least 1 at P_2, which is impossible because the supports are disjoint. So, $f \in \mathbb{F}_q$ and the proof is complete. □

Proposition 3.3. *Let D_1, D_2 and Q be places having the properties described in (3.2). Consider the map $\Lambda : \mathcal{L}(D_1 + D_2) \to F_Q$ such that $\Lambda(f) = f(Q)$ for $f \in \mathcal{L}(D_1 + D_2)$. There exists a vector space $\mathcal{M} \subseteq \ker \Lambda$ of dimension g such that*

$$\mathcal{L}(D_1 + D_2) = \mathcal{L}(D_1) \oplus \mathcal{L}_r(D_2) \oplus \mathcal{M},$$

where $\mathcal{L}_r(D_2)$ is such that

$$\mathcal{L}(D_2) = \mathbb{F}_q \oplus \mathcal{L}_r(D_2)$$

and $\oplus$ denotes the direct sum. In particular, if $g = 0$, then $\mathcal{M} = Ker\Lambda$ is equal to $\{0\}$.

Proof. The divisors D_1 and D_2 which are effective divisors of degree $n+g-1$ have the same dimension n by Section 3.3.2. Moreover, as $\deg(D_1 + D_2) > 2g-2$, the divisor $D_1 + D_2$ is non-special and $\dim \mathcal{L}(D_1 + D_2) = 2n+g-1$. Hence, $\dim Ker\Lambda = n+g-1$ and by Lemma 3.2, the spaces $\mathcal{L}(D_1)$ and $\mathcal{L}_r(D_2)$ consist on a direct sum of dimension $\dim\Big(\mathcal{L}(D_1) \oplus \mathcal{L}_r(D_2)\Big) = 2n-1$. In Section 3.3.2, we have showed that $Ker(E_i) = \{0\}$. Thus, the set $\mathcal{L}(D_i) \cap Ker\Lambda = \{0\}$ as well. Hence, there exists a vector space $\mathcal{M} \subseteq Ker\Lambda$ of dimension g which gives the result. □

Hence, we choose as basis of $\mathcal{L}(D_1 + D_2)$ the basis $\mathcal{B}_{D_1+D_2}$ defined by:

$$\mathcal{B}_{D_1+D_2} = (f_1, \ldots, f_n, f_{n+1}, \ldots, f_{2n+g-1})$$

where $\mathcal{B}_{D_1} = (f_1, \ldots, f_n)$ is the basis of $\mathcal{L}(D_1)$, $(f_{n+1}, \ldots, f_{2n-1})$ is a basis of $\mathcal{L}_r(D_2)$ such that $f_{n+j} = f_{2,j+1} \in \mathcal{B}_{D_2}$ with $\mathcal{B}_{D_1}$ and $\mathcal{B}_{D_2}$ defined in Section 3.3.2 and $\mathcal{B}_{\mathcal{M}} = (f_{2n}, \ldots, f_{2n+g-1})$ is a basis of $\mathcal{M}$.

3.4. **Product of two elements in $\mathbb{F}_{q^n}$.** In this section, we use as representation bases of spaces F_Q, $\mathcal{L}(D_i)$ $(i \in \{1, 2\})$, $\mathcal{L}(D_1+D_2)$, the bases defined in Section 3.3. The product of two elements in $\mathbb{F}_{q^n}$ is computed by the algorithm of Chudnovsky and Chudnovsky. Let $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ be two elements of $\mathbb{F}_{q^n}$ given by their components over $\mathbb{F}_q$ relative to the chosen basis $\mathcal{B}_Q$. According to the previous notation, we can consider that x and y are identified to the following elements:

$$f_x = \sum_{i=1}^{n} x_i f_{1,i} \in \mathcal{L}(D_1) \quad \text{and} \quad f_y = \sum_{i=1}^{n} y_i f_{2,i} \in \mathcal{L}(D_2).$$

The product $f_x f_y$ of the two elements f_x and f_y is their product in the valuation ring $\mathcal{O}_Q$. This product lies in $\mathcal{L}(D_1 + D_2)$ since D_1 and D_2 are effective divisors. We consider that x and y are respectively the elements f_x and f_y embedded in the Rieman-Roch space $\mathcal{L}(D_1 + D_2)$, via respectively the embeddings $I_i : \mathcal{L}(D_i) \longrightarrow \mathcal{L}(D_1+D_2)$ defined by $I_1(f_x)$ and $I_2(f_y)$ as follows. If, f_x and f_y have respectively coordinates f_{x_i} and f_{y_i} in $\mathcal{B}_{D_1+D_2}$ where $i \in \{1, \ldots, 2n+g-1\}$, we have: $I_1(f_x) = (f_{x_1} := x_1, \ldots, f_{x_n} := x_n, 0, \ldots, 0)$ and $I_2(f_y) = (f_{x_1} := y_1, 0, \ldots, 0, f_{y_{n+1}} := y_2, \ldots, f_{y_{2n-1}} := y_n, 0, \ldots 0)$. Now it is clear that knowing x (resp. y) or f_x (resp. f_y) by their coordinates is the same thing.

Theorem 3.4. *Let $P_{\mathcal{M}^s}$ be the projection of $\mathcal{L}(D_1 + D_2)$ onto $\mathcal{M}^s = \mathcal{L}(D_1) \oplus \mathcal{L}_r(D_2)$ and let Λ be the map defined as in Proposition (3.3). Then, for any elements $x, y \in \mathbb{F}_{q^n}$, the product of x by y is such that*

$$xy = \Lambda \circ P_{\mathcal{M}^s}\left(T^{-1}_{|Im\ T}\left(T \circ I_1 \circ E_1^{-1}(x) \odot T \circ I_2 \circ E_2^{-1}(y)\right)\right),$$

where $\circ$ denotes the standard composition map, $T^{-1}_{|Im\ T}$ the restriction of the inverse map of T on the image of T, and $\odot$ the Hadamard product as in Theorem 2.1.

Proof. Let $\mathbb{F}_{q^n}$ be a finite extension of $\mathbb{F}_q$ of degree n with the representation defined in Section 3.3.1. For any two elements $x, y \in \mathbb{F}_{q^n}$, there exist two elements f_x and f_y respectively in $\mathcal{L}(D_1)$ and $\mathcal{L}(D_2)$ defined as in Section 3.3.2 such that $E_1(f_x) = f_x(Q) = x$ and $E_2(f_y) = f_y(Q) = y$ where $f(Q)$ denotes the class of f in the residue class field F_Q of the place Q. Thus,

$$xy = f_x(Q) f_y(Q) = (f_x f_y)(Q)$$

and so computing the product xy is equivalent to computing the product $f_x f_y$. Moreover, since the divisors D_1 and D_2 are effective by the assumptions of Theorem 2.1 and Section 3.1, we have $< \mathcal{L}(D_1)\mathcal{L}(D_2) > \subset \mathcal{L}(D_1+D_2)$ where $< \mathcal{L}(D_1)\mathcal{L}(D_2) >$ denotes the vector space generated by the products $f_x f_y$ with $f_x \in \mathcal{L}(D_1)$, $f_y \in \mathcal{L}(D_2)$ and so $f_x f_y \in \mathcal{L}(D_1 + D_2)$. Now, the principle of the algorithm is to compute $f_x f_y$ via the evaluation map T. In this aim, we represent the elements f_x and f_y in $\mathcal{L}(D_1 + D_2)$ respectively by $I_1(f_x)$ and $I_2(f_y)$ defined in this section. Then by Theorem 2.1,

$$h = f_x f_y = T^{-1}_{|Im\ T}\left(T \circ I_1(f_x) \odot T \circ I_2(f_y)\right) \text{ and } h(Q) = xy.$$

Then, the proof is complete since $h(Q) = \Lambda \circ P_{\mathcal{M}^s}(h)$. □

We can now present the setup algorithm and the multiplication algorithm. Note that the setup algorithm is only done once.

Algorithm 1 Setup algorithm

INPUT: $F/\mathbb{F}_q,\ Q, D_1, D_2, P_1, \ldots, P_N$.
OUTPUT: T and T^{-1}.

(1) The representation of the finite field $\mathbb{F}_q =< a >$, where a is a primitive element i.e. a generator of the associated cyclic group, is fixed.
(2) The function field $F/\mathbb{F}_q$, the place Q, the divisors D_1 and D_2 and the points $P_1, \ldots, P_N$ are such that Conditions (ii) and (iii) in Theorem 2.1 are satisfied. In addition, we require that $\sum_{1\leq i\leq N} \deg P_i = 2n + g - 1$.
(3) Represent $\mathbb{F}_{q^n}$ in the canonical basis $\mathcal{B}_Q = \{1, \alpha, \alpha^2, ..., \alpha^{n-1}\}$, where $\mathbb{F}_{q^n} =< \alpha >$ with α a primitive element as in Section 3.3.1.
(4) Construct a basis $(f_1, \ldots, f_n, f_{n+1}, \ldots, f_{2n+g-1})$ of $\mathcal{L}(D_1 + D_2)$ where $(f_1, \ldots, f_n)$ is the basis of $\mathcal{L}(D_1)$, $(f_1, f_{n+1}, \ldots, f_{2n-1})$ the basis of $\mathcal{L}(D_2)$ and $(f_{2n}, \ldots, f_{2n+g-1})$ the basis of $\mathcal{M}$, defined in Section 3.3.2.
(5) Compute the matrices T and T^{-1}.
(6) Compute the matrice Λ.

Note that any element z of the field $\mathbb{F}_{q^n}$ is known by its components relatively to the canonical basis $\mathcal{B}_Q$: $z = (z_1, \ldots, z_n) \in \mathbb{F}_{q^n}$ (where $z_i \in \mathbb{F}_q$). Then, we have two ways to represent z in our algorithm: embedding z in $\mathcal{L}(D_1 + D_2)$ via $I_1 \circ E_1^{-1}$ or via $I_2 \circ E_2^{-1}$. When we want to multiply two elements x and y, we choose conventionally to represent x by $I_1 \circ E_1^{-1}(x)$ and y by $I_2 \circ E_2^{-1}(y)$.

Algorithm 2 Multiplication algorithm

INPUT: $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$.
OUTPUT: xy.

(1) Compute

$$\begin{pmatrix} z_{1,d_1} \\ \vdots \\ z_{N,d_N} \end{pmatrix} = \begin{pmatrix} z_1 \\ \vdots \\ z_n \\ z_{n+1} \\ \vdots \\ z_{2n+g-1} \end{pmatrix} = T \begin{pmatrix} x_1 \\ \vdots \\ x_n \\ 0 \\ \vdots \\ 0 \end{pmatrix} \text{ and } \begin{pmatrix} t_{1,d_1} \\ \vdots \\ t_{N,d_N} \end{pmatrix} = \begin{pmatrix} t_1 \\ \vdots \\ t_n \\ t_{n+1} \\ \vdots \\ t_{2n+g-1} \end{pmatrix} = T \begin{pmatrix} y_1 \\ 0 \\ \vdots \\ 0 \\ y_2 \\ \vdots \\ y_n \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

where $\sum_{i=1}^{N} d_i = 2n + g - 1$ and $(z_{i,j}, t_{i,j}) \in (\mathbb{F}_{q^{d_j}})^2$ and $(z_i, t_i) \in (\mathbb{F}_q)^2$.
(2) Compute the Hadamard product $u = (u_{1,d_1}, \ldots, u_{N,d_N}) = (u_1, \ldots, u_{2n+g-1})$, where $u_{i,d_i} = z_{i,d_i} t_{i,d_i}$, in $\mathbb{F}_{q^{d_1}} \times \mathbb{F}_{q^{d_2}} \times \cdots \times \mathbb{F}_{q^{d_N}}$ as in Theorem 2.1.
(3) Compute $w = (w_1, \ldots, w_{2n+g-1}) = T^{-1}(u)$.
(4) Extract $w' = (w_1, \ldots, w_{2n-1})$ (remark that in the previous step we just have to compute the $2n - 1$ first components of w).
(5) Return xy=$\Lambda(w')$.

4. Complexity analysis

By Theorem 2.1, the condition (ii) implies $\dim \mathcal{L}(D_i) \geq n$. But by Riemann-Roch Theorem, $\dim \mathcal{L}(D_i) \geq -g + 1 + \deg D_i$ which gives in the least case: $\deg D_i \geq n + g - 1$ and so, $\deg(D_1 + D_2) \geq 2n + 2g - 2$. Without loss of generality, we suppose that $\deg D_i = n + g - 1$ and so $\deg(D_1 + D_2) = 2n + 2g - 2$. In this case, $\deg(D_1 + D_2) \geq 2g - 1$ and then we obtain $\dim \mathcal{L}(D_1 + D_2) = 2n + g - 1$. According to Theorem 2, Algorithm 2 requires that the natural evaluation map T is injective. A sufficient condition to get injectivity

is given by the following condition:

$$\sum_{i|r}^{r} iN_i > 2n + 2g - 2 \tag{2}$$

where N_i denotes the number of places of degree i in $\mathcal{P}$ and r an integer > 1. Indeed, the kernel of T is $\mathcal{L}(D_1 + D_2 - \sum_{P\in\mathcal{P}} P)$ and, under Condition 2, this kernel is trivial since the divisor $D_1 + D_2 - \sum_{P\in\mathcal{P}} P$ has negative degree.

In terms of number of multiplications in $\mathbb{F}_q$, the complexity of this multiplication algorithm is as follows: calculation of z and t needs $2(2n^2+ng-n)$ multiplications, calculation of u needs $(2n+2g-2+r)\sup_{1\leq i\leq r}\frac{\mu_q(i)}{i}$ bilinear multiplications and calculation of the $2n-1$ first components of w needs $(2n+g-1)(2n-1)$ multiplications (remark that in Algorithm 2, we just have to compute the $2n-1$ first components of w). The calculation of xy needs $n+g$ multiplications. The total complexity with respect to the multiplications is bounded by $8n^2 + n(4g-5) + (2n+2g-2+r)\sup_{1\leq i\leq r}\frac{\mu_q(i)}{i}$.

The asymptotic analysis of our method needs to consider infinite families of algebraic function fields defined over $\mathbb{F}_q$ with increasing genus (or equivalently of algebraic curves) having the required properties. The existence of such families follows from that of families of algebraic function fields reaching the Generalized Drinfeld-Vladut bound of order r (cf. [9]). For example, it is proved in [4] (with rational places i.e. $r = 1$), in [8, 10] (with places of degree two i.e. $r = 2$) and in [7] (with places of degree four i.e. $r = 4$) from a specialization of the Chudnovsky type symmetric algorithms on recursive towers of algebraic function fields of type Garcia-Stichtenoth that the bilinear complexity of the multiplication in any degree n extension of $\mathbb{F}_q$ is uniformly linear in q with respect to n. Good asymptotic bounds are also obtained by using families of modular Shimura curves [11]. Similarly, Randriambololona improved the uniform (resp. asymptotic) bounds with an asymmetric algorithm of type Theorem 2.1. Hence, the number of bilinear multiplications of the algorithm 2.1 is in $O(n)$ when the places used in the algorithm 2.1 have a degree one or two. Moreover, the genus g of the required curves also necessarily increases in $O(n)$. Consequently, the total number of multiplications/additions/subtractions of the algorithm 2.1 is in $\mathrm{O}\big(n^2\big)$ and the total number of bilinear multiplications is in $\mathrm{O}\big(n\big)$.

Remark 4.1. The general construction of the set-up algorithm involves some random choice of divisors having prescribed properties over an exponentially large set of divisors. To get a polynomially constructible algorithm with linear complexity, one needs to construct explicitly (i.e. polynomially) points of corresponding degrees n on curves of arbitrary genus with many rational points. Unfortunately, so far it is unknown how to produce such points (cf. [26, Section 4, Remark 5] and [23, Remark 6.6]). Hence, the asymptotic complexity of such a construction is an open problem. Note that there exists a polynomial time construction of a bilinear algorithm with linear bilinear complexity for the multiplication of two elements in any extension finite field [16], but this type of bilinear algorithm is not as performant as algorithms of Chudnovsky type with respect to the bilinear complexity [16, Section 8], *idem* for elliptic Chudnovsky type algorithms [3] which are symmetric with an only quasi-linear bilinear complexity.

5. Multiplication in $\mathbb{F}_{16^n/\mathbb{F}_{16}}$

Set $q = 16$ and $n = 13, 14, 15$. Note that the multiplication algorithms in the extensions of degree $n < 13$ are symmetric because they are obtained with rational and elliptic function fields (with the best possible bilinear complexities of multiplication). Hence, it is only pertinent to consider the multiplication in extensions of degree ≥ 13 namely with function fields of genus $g \geq 2$. From now on, $F/\mathbb{F}_q$ denotes the algebraic function field associated to the hyperelliptic curve X with plane model $y^2 + y = x^5$, of genus two. This curve has 33 rational points, which is maximal over $\mathbb{F}_q$ according to the Hasse-Weil bound. We represent $\mathbb{F}_{16}$ as the field $\mathbb{F}_2(a) = \mathbb{F}_2[X]/(P(X))$ where $P(X)$ is the irreducible polynomial $P(X) = X^4 + X + 1$ and a denotes a primitive root of $P(X) = X^4 + X + 1$. Let us give the projective coordinates $(x : y : z)$ of rational points of the curve X:

$$
\begin{array}{lll}
P_\infty = (0:1:0) & P_2 = (0:0:1) & P_3 = (0:1:1) \\
P_4 = (a:a:1) & P_5 = (a:a^4:1) & P_6 = (a^2:a^2:1) \\
P_7 = (a^2:a^8:1) & P_8 = (a^3:a^5:1) & P_9 = (a^3:a^{10}:1) \\
P_{10} = (a^4:a:1) & P_{11} = (a^4:a^4:1) & P_{12} = (a^5:a^2:1) \\
P_{13} = (a^5:a^8:1) & P_{14} = (a^6:a^5:1) & P_{15} = (a^6:a^{10}:1) \\
P_{16} = (a^7:a:1) & P_{17} = (a^7:a^4:1) & P_{18} = (a^8:a^2:1) \\
P_{19} = (a^8:a^8:1) & P_{20} = (a^9:a^5:1) & P_{21} = (a^9:a^{10}:1) \\
P_{22} = (a^{10}:a:1) & P_{23} = (a^{10}:a^4:1) & P_{24} = (a^{11}:a^2:1) \\
P_{25} = (a^{11}:a^8:1) & P_{26} = (a^{12}:a^5:1) & P_{27} = (a^{12}:a^{10}:1) \\
P_{28} = (a^{13}:a:1) & P_{29} = (a^{13}:a^4:1) & P_{30} = (a^{14}:a^2:1) \\
P_{31} = (a^{14}:a^8:1) & P_{32} = (1:a^5:1) & P_{33} = (1:a^{10}:1)
\end{array}
$$

5.1. **Construction of the required divisors.**

5.1.1. *A place Q of degree n.* It is sufficient to take a place Q of degree n in the rational function field $\mathbb{F}_q(x)/\mathbb{F}_q$, which totally splits in $F/\mathbb{F}_q$. It is equivalent to choose a monic irreducible polynomial $\mathcal{Q}(x) \in \mathbb{F}_q[x]$ of degree n such that its roots α_i in $\mathbb{F}_{q^n}$ satisfy $Tr_{\mathbb{F}_2}(\alpha_i^5) = 0$ for $i = 1, ..., n$ where the map $Tr_{\mathbb{F}_2}$ denotes the classical function Trace over $\mathbb{F}_2$ by [20, Theorem 2.25]. In fact, it is sufficient to verify that this property is satisfied for only one root since a finite field is Galois.

For example, for the extension $n = 13$, we choose the irreducible polynomial

$$
\begin{aligned}
\mathcal{Q}(x) = x^{13} + a^6x^{12} + a^5x^{11} + a^{11}x^{10} + x^9 + a^{12}x^8 + \\
a^7x^7 + a^7x^5 + a^2x^4 + a^{11}x^3 + a^8x^2 + a^6x + a^{14}.
\end{aligned} \tag{3}
$$

Let b be a root of $\mathcal{Q}(x)$. It is easy to check that $Tr_{\mathbb{F}_2}(b^5) = 0$, hence the place $(\mathcal{Q}(x))$ of $\mathbb{F}_{16}(x)/\mathbb{F}_{16}$ is totally splitted in the algebraic function field $F/\mathbb{F}_q$, which means that there exist two places of degree n in $F/\mathbb{F}_q$ lying over the place $(\mathcal{Q}(x))$ of $\mathbb{F}_{16}(x)/\mathbb{F}_{16}$. For the place Q of degree n in the algebraic function field $F/\mathbb{F}_q$, we consider one of the two places in $F/\mathbb{F}_q$ lying over the place $(\mathcal{Q}(x))$ of $\mathbb{F}_{16}(x)/\mathbb{F}_{16}$, namely the orbit of the $\mathbb{F}_{16^{13}}$-rational point $\mathcal{P}_{1i} = (\alpha_i : \beta_i : 1)$ where α_i is a root of $\mathcal{Q}(x)$ and $\beta_i = a^6\alpha_i^{12} + a^{13}\alpha_i^{11} + a\alpha_i^{10} + a^{13}\alpha_i^9 + a^8\alpha_i^8 + a\alpha_i^7 + a^8\alpha_i^6 + a^9\alpha_i^5 + a^5\alpha_i^4 + a^2\alpha_i^2 + a^8\alpha_i + a^{13}$ for $i = 1, ..., 13$. Notice that the second place is given by the conjugated points $\mathcal{P}_{2i} = (\alpha_i : \beta_i + 1 : 1)$ for $i = 1, ..., 13$.

5.1.2. *The two divisors D_1 and D_2 of degree n+g-1.* For the divisor D_1 of degree $n + g - 1$, we choose a place D_1 of degree 14 according to the method used for the place Q. We consider the orbit of the $\mathbb{F}_{16^{14}}$-rational point $\mathcal{P}_{1i}^1 = (\gamma_i : \delta_i : 1)$ where γ_i is a root of $\mathcal{D}_1(x) = x^{14}+a^9x^{13}+a^6x^{12}+a^7x^{11}+a^{11}x^{10}+a^{12}x^9+a^{10}x^8+a^6x^7+a^7x^6+a^{10}x^5+a^{14}x^4+x^3+x^2+a^3x+a$ and $\delta_i = a^4\gamma_i^{12}+a^8\gamma_i^{11}+a^7\gamma_i^9+a^2\gamma_i^8+a^3\gamma_i^7+a^8\gamma_i^6+a^4\gamma_i^5+a^{14}\gamma_i^4+\gamma_i^2+a^6\gamma_i+a^3$ for $i = 1, ..., 14$. Notice that the second place is given by the conjugated points $\mathcal{T}_{2i}^1 = (\gamma_i : \delta_i + 1 : 1)$ for $i = 1, ..., 14$.

For the divisor D_2 of degree $n + g - 1$, we choose a place D_2 of degree 14 according to the method used for the place Q. We consider the orbit of the $\mathbb{F}_{16^{14}}$-rational point $\mathcal{P}_{1i}^2 = (\gamma_i : \delta_i : 1)$ where γ_i is a root of $\mathcal{D}_2(x) = x^{14}+x^2+ax+1$ and $\delta_i = a^5\gamma_i^{12}+a^{11}\gamma_i^{11}+a^{11}\gamma_i^{10}+a^8\gamma_i^9+a^4\gamma_i^8+a^8\gamma_i^7+\gamma_i^6+a^8\gamma_i^5+a^2\gamma_i^4+a^9\gamma_i^3+a^2\gamma_i^2+a^2\gamma_i+a^7$ for $i = 1, ..., 14$. Notice that the second place is given by the conjugated points $\mathcal{T}_{2i}^2 = (\gamma_i : \delta_i + 1 : 1)$ for $i = 1, ..., 14$.

The place Q and the divisors D_1 and D_2 satisfy the good properties since the dimensions of the divisor $D_1 - Q$ and $D_2 - Q$ are zero which means that the divisors $D_1 - Q$ and $D_2 - Q$ are non-special of degree $g - 1$.

5.2. **Construction of required bases.**

5.2.1. *The basis of the residue class field F_Q.* We choose as basis of the residue class field F_Q the basis $\mathcal{B}_Q$ associated to the place Q obtained in Section 5.1.1.

5.2.2. *The basis of $\mathcal{L}(D_i)$ for i=1,2.* We choose as basis of the Riemann-Roch space $\mathcal{L}(D_i)$ the basis $\mathcal{B}_{D_i}$ such that $E_i(\mathcal{B}_{D_i}) = \mathcal{B}_Q$ is a basis of F_Q as in Section 3.3.2, $\mathcal{B}_{D_1} = (f_1, ..., f_n)$ and $\mathcal{B}_{D_2} = (f_1, f_{n+1}..., f_{2n-1})$. For $j \in \{2, \ldots, n\}$, any element f_j of $\mathcal{B}_{D_1}$ is such that

$$
f_j(x, y) = \frac{f_{j1}(x)y + f_{j2}(x)}{\mathcal{D}_1(x)},
$$

and for $j \in \{n+1, \ldots, 2n-1\}$, any element f_j of $\mathcal{B}_{D_2}$ is such that

$$f_j(x,y) = \frac{f_{j1}(x)y + f_{j2}(x)}{\mathcal{D}_2(x)},$$

where $f_{j1}, f_{j2} \in \mathbb{F}_{16}[x]$. To simplify, we set $f_j(x,y) = (f_{j1}(x), f_{j2}(x))$. We have:

$f_1(x,y) = 1$,

$f_2(x,y) = (a^{13}x^{11} + a^{10}x^{10} + a^3x^9 + a^{10}x^8 + a^{14}x^7 + a^{11}x^6 + a^8x^5 + a^{11}x^4 + x^3 + ax^2 + a^{11}x + a^{11},\ a^{12}x^{14} + a^{12}x^{13} + a^9x^{12} + x^{11} + a^8x^{10} + a^{13}x^9 + a^{12}x^8 + ax^7 + a^5x^6 + x^5 + a^{13}x^4 + a^5x^3 + a^{12}x^2 + a^4x)$,

$f_3(x,y) = (a^2x^{11} + a^{11}x^{10} + a^{13}x^9 + a^3x^8 + a^{10}x^7 + ax^6 + a^9x^5 + a^6x^4 + a^5x^3 + a^3x^2 + a^7x + a^{14},\ a^{12}x^{14} + a^2x^{13} + a^{11}x^{12} + a^3x^{11} + a^{13}x^{10} + a^7x^9 + a^9x^8 + a^9x^7 + a^{13}x^6 + ax^5 + a^5x^4 + a^{12}x^3 + a^{13}x^2 + a^8x + a^2)$,

$f_4(x,y) = (a^8x^{11} + ax^{10} + a^{10}x^9 + x^8 + a^8x^7 + a^{14}x^6 + a^6x^5 + a^3x^4 + a^{14}x^3 + a^3x^2 + a^6x + a,\ a^3x^{14} + a^7x^{13} + a^{10}x^{12} + a^{11}x^{11} + a^{13}x^9 + a^8x^8 + a^8x^7 + a^3x^6 + a^5x^5 + a^6x^4 + a^3x^3 + x^2 + a^5x + a^6)$,

$f_5(x,y) = (a^{12}x^{11} + a^{12}x^{10} + a^{14}x^9 + a^7x^8 + a^7x^7 + a^7x^6 + a^3x^5 + a^{13}x^4 + a^2x^3 + a^7x^2 + a^7x + a^7,\ a^2x^{14} + a^{11}x^{13} + a^5x^{12} + a^{10}x^{11} + a^{10}x^9 + a^{13}x^8 + ax^7 + a^{10}x^6 + a^6x^5 + a^{12}x^4 + a^3x^3 + a^4x^2 + a^{10}x + a^{12})$,

$f_6(x,y) = (a^6x^{11} + a^{14}x^{10} + x^9 + x^8 + a^4x^7 + a^2x^6 + a^7x^5 + a^{13}x^4 + a^4x^3 + a^{12}x^2 + a^5x + a^7,\ a^9x^{14} + a^6x^{13} + a^4x^{12} + a^4x^{11} + a^{12}x^{10} + a^{10}x^9 + a^7x^8 + a^2x^7 + a^{11}x^5 + a^{14}x^4 + a^4x^2 + a^{11}x + a)$,

$f_7(x,y) = (a^8x^{11} + a^{11}x^{10} + a^{12}x^9 + a^2x^8 + a^{14}x^7 + a^{10}x^6 + a^4x^5 + a^7x^4 + a^2x^3 + a^{13}x + a^{12},\ a^8x^{14} + a^{10}x^{13} + a^{14}x^{12} + a^7x^{11} + a^5x^{10} + a^{13}x^9 + a^{13}x^8 + a^{12}x^7 + a^7x^6 + a^8x^5 + a^{12}x^4 + x^3 + a^{12}x^2 + a^6x + a^{10})$,

$f_8(x,y) = (a^{13}x^{11} + ax^{10} + a^{11}x^9 + ax^8 + a^9x^7 + a^{11}x^6 + a^{10}x^5 + a^9x^4 + x^3 + a^4x^2 + a^6x + 1,\ a^2x^{14} + a^2x^{13} + a^{11}x^{12} + a^5x^{11} + a^7x^{10} + a^2x^9 + a^4x^8 + a^{11}x^7 + a^{14}x^6 + a^{13}x^5 + a^8x^4 + a^4x^3 + a^6x^2 + x + a^{12})$,

$f_9(x,y) = (a^{10}x^{11} + a^6x^{10} + a^{12}x^9 + a^{12}x^8 + a^7x^7 + a^3x^6 + a^{12}x^5 + a^2x^4 + a^6x^3 + a^{12}x^2 + a^6x + 1,\ a^8x^{14} + a^{11}x^{13} + ax^{12} + a^8x^{11} + a^5x^{10} + a^8x^9 + a^3x^8 + a^{14}x^7 + a^9x^6 + a^{13}x^5 + a^{11}x^4 + a^3x^3 + a^7x^2 + x + a^2)$,

$f_{10}(x,y) = (a^{14}x^{11} + ax^9 + a^{12}x^8 + a^3x^7 + x^6 + a^7x^5 + a^{11}x^4 + a^{14}x^3 + a^8x^2 + ax + a^7,\ a^3x^{14} + a^8x^{13} + a^7x^{11} + a^{14}x^{10} + a^{13}x^9 + a^9x^8 + a^6x^7 + a^9x^6 + a^8x^5 + a^{12}x^2 + a^{13}x + a^{14})$,

$f_{11}(x,y) = (a^9x^{11} + a^5x^{10} + a^5x^9 + ax^8 + a^7x^7 + a^5x^6 + a^2x^5 + a^4x^4 + a^3x^3 + a^2x^2 + ax + a,\ a^{10}x^{14} + a^4x^{12} + a^5x^{11} + a^{14}x^{10} + a^5x^9 + a^9x^7 + a^7x^6 + a^4x^5 + a^{14}x^4 + a^{10}x^3 + a^9x^2 + a^5x + a^3)$,

$f_{12}(x,y) = (a^{11}x^{11} + a^9x^{10} + a^{10}x^9 + a^7x^8 + a^{10}x^6 + a^2x^5 + a^{13}x^4 + a^7x^3 + a^2x^2 + a^{11}x + a^3,\ a^9x^{14} + a^{11}x^{13} + x^{12} + a^{11}x^{11} + x^{10} + a^9x^9 + a^8x^8 + a^{11}x^7 + a^8x^6 + a^{12}x^5 + a^6x^4 + a^2x^3 + a^4x^2 + a^{14}x + a^{13})$,

$f_{13}(x,y) = (a^9x^{10} + ax^9 + a^3x^8 + a^{10}x^7 + a^7x^6 + a^{12}x^4 + a^3x^3 + a^{12}x^2 + a^{11}x + a^8,\ ax^{14} + a^{12}x^{13} + a^2x^{12} + a^3x^{11} + x^{10} + a^{12}x^9 + a^3x^8 + a^{14}x^7 + x^6 + a^{14}x^5 + a^{10}x^4 + a^5x^3 + a^5x^2 + a^6x + 1)$,

$f_{14}(x,y) = (a^{14}x^{11} + a^4x^{10} + a^{10}x^9 + a^6x^8 + a^7x^7 + a^{14}x^6 + a^{12}x^5 + ax^4 + a^5x^3 + a^4x + a^{12},\ a^{14}x^{14} + a^{14}x^{13} + a^3x^{12} + a^{11}x^{11} + a^{12}x^{10} + ax^9 + a^5x^8 + a^2x^7 + a^6x^6 + a^{14}x^5 + a^6x^3 + a^{11}x^2 + a^4x + a^9)$,

$f_{15}(x,y) = (a^8x^{11} + a^8x^{10} + a^2x^9 + a^{13}x^8 + a^4x^7 + ax^6 + a^7x^4 + a^3x^3 + a^9x^2 + a^5x + a^5,\ a^{10}x^{14} + a^6x^{13} + x^{12} + a^{11}x^{11} + a^4x^{10} + a^{10}x^9 + a^{14}x^8 + a^{13}x^7 + a^{11}x^6 + ax^5 + a^{12}x^4 + x^2 + a^5)$,

$f_{16}(x,y) = (a^{13}x^{11} + a^8x^{10} + a^6x^9 + x^8 + a^{12}x^7 + a^9x^6 + x^5 + a^{10}x^4 + a^{14}x^3 + a^4x^2 + a^8x + a^2,\ a^7x^{14} + a^{11}x^{13} + x^{12} + a^5x^{11} + a^{10}x^9 + a^9x^8 + a^6x^7 + a^9x^6 + a^{14}x^5 + a^{13}x^4 + a^{11}x^3 + a^6x^2 + a^9x + a^{11})$,

$f_{17}(x,y) = (a^8x^{11} + a^{10}x^{10} + a^2x^9 + ax^8 + a^{13}x^6 + ax^5 + a^9x^4 + a^3x^2 + a^4x + a^{10},\ a^9x^{14} + a^{13}x^{12} + a^7x^{11} + a^9x^{10} + a^{12}x^9 + a^2x^8 + a^3x^7 + a^{13}x^6 + a^{12}x^5 + a^5x^4 + a^4x^3 + a^8x^2 + a^{10}x + a^9)$,

$f_{18}(x,y) = (x^{10} + a^5x^9 + x^8 + a^5x^7 + a^{10}x^6 + a^{10}x^5 + a^5x^4 + a^4x^3 + a^5x^2 + a^4,\ a^3x^{14} + a^{13}x^{13} + a^5x^{12} + a^8x^{11} + a^{12}x^{10} + a^4x^9 + a^{10}x^8 + a^{11}x^7 + a^2x^6 + a^{12}x^5 + a^{11}x^4 + a^2x^3 + a^3x^2 + a^{12}x + a^8)$,

$f_{19}(x,y) = (a^8x^{11} + a^{13}x^{10} + a^6x^9 + a^6x^8 + a^4x^6 + a^{10}x^5 + a^4x^3 + a^3x + 1,\ a^{14}x^{14} + ax^{13} + a^{14}x^{12} + a^5x^{11} + ax^{10} + a^2x^9 + a^7x^8 + ax^7 + a^8x^6 + a^9x^5 + a^2x^4 + a^9x^3 + a^8x^2 + a^5x + a^{12})$,

$f_{20}(x,y) = (ax^{11} + a^6x^{10} + x^9 + a^5x^8 + a^5x^7 + a^9x^6 + a^{12}x^5 + a^4x^4 + x^3 + a^{13}x^2 + a^4x + 1,\ ax^{14} + x^{13} + a^6x^{11} + a^5x^{10} + a^9x^9 + a^{10}x^8 + a^9x^7 + a^5x^6 + a^8x^5 + a^{10}x^4 + a^{11}x^2 + a^6x + a^3)$,

$f_{21}(x,y) = (a^{10}x^{11} + a^{14}x^{10} + a^{13}x^8 + a^2x^7 + a^{11}x^6 + a^7x^5 + a^7x^3 + x^2 + a^4x + a^7,\ a^6x^{14} + a^4x^{13} + a^8x^{12} + ax^{11} + a^{11}x^{10} + a^4x^9 + a^7x^8 + ax^7 + a^9x^6 + a^{12}x^5 + a^6x^4 + a^4x^3 + a^2x^2 + a^5x + a^7)$,

$f_{22}(x,y) = (a^{10}x^{11} + x^{10} + a^7x^8 + x^7 + a^{13}x^6 + a^6x^5 + a^2x^4 + a^{10}x^3 + a^2x^2 + ax + 1,\ a^2x^{14} + a^2x^{13} + a^3x^{11} + a^{11}x^{10} + a^{10}x^8 + a^{11}x^7 + a^{11}x^6 + a^{12}x^5 + ax^4 + a^{14}x^3 + a^2x^2 + a^3x)$,

$f_{23}(x,y) = (a^7x^{11} + a^9x^{10} + a^7x^9 + a^2x^8 + a^8x^7 + a^{11}x^6 + a^{14}x^5 + a^7x^4 + a^{10}x^3 + a^8x^2 + a^4x,\ a^{10}x^{14} + a^8x^{13} + a^5x^{12} + x^{11} + a^6x^{10} + a^{12}x^9 + a^{11}x^8 + a^6x^7 + a^2x^6 + a^8x^5 + a^3x^4 + a^7x^3 + a^{12})$,

$f_{24}(x,y) = (x^{11} + a^5x^{10} + a^3x^9 + a^{10}x^8 + a^{10}x^7 + a^4x^6 + a^{14}x^5 + a^8x^3 + a^3x^2 + a^6x + a^8,\ a^{10}x^{13} + a^{13}x^{12} + a^{13}x^{11} + ax^{10} + a^{11}x^8 + a^2x^7 + ax^6 + a^2x^5 + a^2x^4 + a^8x^2 + a^6)$,

$f_{25}(x,y) = (a^4x^{11} + a^{12}x^{10} + a^4x^9 + a^{10}x^8 + a^{12}x^7 + a^{14}x^6 + a^7x^5 + a^{11}x^4 + a^{13}x^3 + a^{13}x^2 + a^{10}x + a^4,\ x^{14} + a^6x^{13} + a^3x^{12} + a^{11}x^{11} + a^{13}x^{10} + a^{11}x^9 + a^7x^8 + a^2x^7 + a^5x^6 + a^{14}x^5 + a^5x^4 + a^{10}x^3 + a^3x^2 + a^{14}x + a^5)$.

5.2.3. *The basis of* $\mathcal{L}(D_1 + D_2)$. As seen in Section 3.3.3, $\mathcal{B}_{D_1+D_2} = (f_1, \ldots, f_n, f_{n+1}, \ldots, f_{2n+g-1})$ where, for $j \in \{1, \ldots, 2n-1\}$, the f_i are defined above. The basis is completed with

$$f_{26}(x,y) = \frac{g_{26}(x)y + h_{26}(x)}{r(x)} \text{ and } f_{27}(x,y) = \frac{g_{27}(x)y + h_{27}(x)}{r(x)}$$

where:

$g_{26}(x) = a^{13}x^{25} + a^{14}x^{24} + a^2x^{23} + a^8x^{22} + a^7x^{21} + ax^{19} + a^9x^{18} + a^7x^{17} + a^6x^{16} + a^{10}x^{15} + a^{10}x^{14} + a^2x^{13} + a^5x^{12} + a^4x^{11} + a^{11}x^9 + a^{12}x^8 + a^7x^7 + a^{14}x^6 + x^5 + a^8x^4 + a^9x^3 + a^7x^2$,

$h_{26}(x) = a^{13}x^{28} + a^{11}x^{26} + x^{25} + a^4x^{24} + a^5x^{23} + a^3x^{22} + a^3x^{21} + x^{20} + a^{12}x^{19} + a^4x^{18} + a^{13}x^{16} + a^5x^{15} + a^{11}x^{14} + a^6x^{13} + ax^{12} + x^{11} + a^8x^{10} + a^8x^9 + a^{12}x^7 + a^2x^6 + a^2x^5 + x^4 + a^8x^3 + a^{14}x^2 + a^{10}x + a^{10}$,

$g_{27}(x) = a^4x^{25} + a^6x^{24} + a^6x^{23} + a^2x^{22} + a^3x^{21} + a^9x^{20} + a^2x^{19} + a^{14}x^{18} + a^9x^{17} + a^8x^{16} + a^{13}x^{15} + a^{12}x^{14} + a^8x^{13} + x^{12} + a^3x^{10} + a^4x^9 + a^{12}x^8 + a^6x^7 + a^2x^6 + a^{14}x^5 + a^5x^4 + a^4x^3 + a^4x^2 + a^5x + a^5$,

$h_{27}(x) = a^9x^{28} + a^5x^{27} + a^6x^{26} + a^4x^{25} + x^{24} + a^7x^{23} + a^5x^{22} + a^6x^{21} + x^{20} + a^{12}x^{19} + a^7x^{18} + a^{11}x^{17} + a^8x^{16} + a^7x^{15} + a^6x^{14} + a^4x^{13} + a^7x^{12} + a^3x^{11} + a^9x^{10} + x^9 + a^5x^8 + a^9x^7 + a^9x^6 + a^5x^5 + x^4 + a^{13}x^2 + a^3x + a^2$,

$r(x) = x^{28} + a^9x^{27} + a^6x^{26} + a^7x^{25} + a^{11}x^{24} + a^{12}x^{23} + a^{10}x^{22} + a^6x^{21} + a^7x^{20} + a^{10}x^{19} + a^{14}x^{18} + x^{17} + a^3x^{14} + a^9x^{13} + a^{10}x^{12} + a^7x^{11} + a^2x^{10} + a^{13}x^9 + a^{10}x^8 + a^{11}x^7 + a^6x^6 + a^{10}x^5 + a^9x^4 + a^7x^3 + a^6x + a$.

6. Multiplication in $\mathbb{F}_{4^n/\mathbb{F}_4}$

Set $q = 4$ and $n = 4$. With the algebraic function field $F/\mathbb{F}_q$ defined over $\mathbb{F}_4$ associated to the hyperelliptic curve X with plane model $y^2 + y = x^5$, of genus two, we cannot multiply only with places of degree one. In fact, if we use only places of degree one we get that $\dim \mathrm{Im}(T) \leq N_1(F/\mathbb{F}_4)$ whereas $\dim \mathcal{L}(D_1 + D_2) = 2n + g - 1 = 9$. As a consequence, the evaluation map T is not injective as required in Theorem 2.1 since $N_1(F/\mathbb{F}_4) = 5$. We can ask whether there exists another hyperelliptic curve of genus two that allows the multiplication with only places of degree one in $\mathbb{F}_{4^4}$. By definition, an hyperelliptic curve is a covering of degree two of the projective line $\mathbb{P}_1(\mathbb{F}_q)$. It is clear that the maximal number $N_q(g)$ of rational points of a projective smooth absolutely irreducible curve of genus $g = 2$ over $\mathbb{F}_q$ is such that $N_q(2) \leq 2q+2$. In the case of $q = 4$ and $g = 2$, this bound is least that the bound of Serre-Weil $N_q(2) \leq q + 1 + 2m$ where $m = \lfloor 2\sqrt{q} \rfloor$. Hence, there does not exist a maximal Serre-Weil curve (i.e. attaining the Serre-Weil bound) over $\mathbb{F}_4$ of genus two and $N_4(2) \leq 2q + 2 = q + 1 + 2m - 3 = 10$. In fact, Serre in [24] proves that $N_4(2) = 10$ and Shabat in [25] exhibits an optimal (i.e. attaining $N_q(g)$) curve of genus two over $\mathbb{F}_4$, namely the curve of equation $y^2 + y = \frac{x}{x^3+x+1}$. Clearly, the sufficient Condition (2) fails since $N_1(F/\mathbb{F}_4) > 2n + 2g - 2$ implies that $n < 4$. Nevertheless, it is still possible to multiply according to Theorem 2.1 with the ten places of degree one of the algebraic function field $F/\mathbb{F}_4$ associated to the curve $y^2 + y = \frac{x}{x^3+x+1}$, by choosing suitable degree n place

Q and divisors D_1 and D_2 of $F/\mathbb{F}_4$. For instance, using the description given by Magma, we can choose the n-degree place

$$Q = (x^4 + a^2x^2 + a^2x + a, (x^3 + x + 1)y + x^2 + a^2x + a)$$

that lies over the place $(\mathcal{Q}(x))$ of the rational field $\mathbb{F}_4(x)$ defined by $\mathcal{Q}(x) = x^4 + a^2x^2 + a^2x + a$, and the divisors

$$D_1 = (x^5 + x^4 + x^3 + x^2 + x + a^2, (x^3 + x + 1)y + a^2x^3 + a^2x^2 + a^2x + a^2)$$

and

$$D_2 = (x^5 + x + a, (x^3 + x + 1)y + a^2x^4 + a^2x^3 + a^2x + a^2)$$

of degree $n + g - 1 = 5$ that respectively lie over the places $(\mathcal{D}_1(x))$ and $(\mathcal{D}_2(x))$ of the rational field $\mathbb{F}_4(x)$ defined by $\mathcal{D}_1(x) = x^5 + x^4 + x^3 + x^2 + x + a^2$ and $\mathcal{D}_2(x) = x^5 + x + a$. Indeed, in this case, the divisors D_1 and D_2 as well as the place Q satisfy the conditions required by Theorem 2.1, namely: $\dim \mathcal{L}(D_1 - Q) = 0$, $\dim \mathcal{L}(D_2 - Q) = 0$ and $\dim \mathcal{L}(D_1 + D_2 - \sum_{P_i \in \mathbb{P}_1(F/\mathbb{F}_q)} P_i) = 0$.

Now, suppose that $n = 5$. In this case, we have $\dim \mathcal{L}(D_1 + D_2) = 2n + g - 1 = 11$ and still $\dim \mathrm{Im}(T) \leq N_1(F/\mathbb{F}_4) = 10$. This means that T cannot be injective. Since the curve of equation $y^2 + y = \frac{x}{x^3+x+1}$ is optimal, it is impossible to multiply with only places of degree one in extensions of $\mathbb{F}_4$ of degree $n \geq 5$. Consequently, the algorithm has to be modified in order to use places of degree two. It is clear that we need to use as much as possible places of degree one to minimize bilinear complexity. Therefore, our algorithm will still use the curve of equation $y^2 + y = \frac{x}{x^3+x+1}$. This curve has $N_1(F/\mathbb{F}_4) = 10$ rational places and $N_2(F/\mathbb{F}_4) = 4$ places of degree two. We represent $\mathbb{F}_4$ as the field $\mathbb{F}_2(a) = \mathbb{F}_2[X]/(P_1(X))$ where $P_1(X)$ is the primitive irreducible polynomial $P_1(X) = X^2 + X + 1$ and a denotes a primitive root of $P_1(X) = X^2 + X + 1$. As we also use evaluations over places of degree two, we need to define the finite field $\mathbb{F}_{16} = \mathbb{F}_{4^2}$ as $\mathbb{F}_4$-vector space. So, we represent $\mathbb{F}_{4^2}$ as the field $\mathbb{F}_4(b) = \mathbb{F}_4[X]/(P_2(X))$ where $P_2(X)$ is the primitive irreducible polynomial $P_2(X) = X^2 + X + a$ and b denotes a primitive root of $P_2(X) = X^2 + X + a$.

For the description of the places of degree one, we use the description given by Magma:

$$\begin{array}{ll}
P_{\infty,1} = (\frac{1}{x}, \frac{x^3+x+1}{x^3}y + \frac{1}{x}) & P_{\infty,2} = (\frac{1}{x}, \frac{x^3+x+1}{x^3}y + \frac{x+1}{x}) \\
P_3 = (x, (x^3 + x + 1)y) & P_4 = (x, (x^3 + x + 1)y + x + 1) \\
P_5 = (x + a, (x^3 + x + 1)y + 1) & P_6 = (x + a, (x^3 + x + 1)y + x + 1) \\
P_7 = (x + a^2, (x^3 + x + 1)y + 1) & P_8 = (x + a^2, (x^3 + x + 1)y + x + 1) \\
P_9 = (x + 1, (x^3 + x + 1)y + a) & P_{10} = (x + 1, (x^3 + x + 1)y + a^2).
\end{array}$$

Note that the first two infinite places lie above the infinite place $1/x$ of the rational function field $\mathbb{F}_q(x)$. Moreover, if $(x : y : z)$ denotes the projective coordinates of rational points of the curve X, then these infinite places correspond to the two points to the infinity $P_{\infty,1} = (1 : 0 : 0)$ and $P_{\infty,2} = (0 : 1 : 0)$ of X.

For the description of the places of degree two, we use the description given by Magma:

$$\begin{array}{ll}
Q_1 = (x^2 + ax + a, (x^3 + x + 1)y + a^2x + a^2) & Q_2 = (x^2 + ax + a, (x^3 + x + 1)y + a^2x + 1) \\
Q_3 = (x^2 + a^2x + a^2, (x^3 + x + 1)y + ax + a) & Q_4 = (x^2 + a^2x + a^2, (x^3 + x + 1)y + ax + 1).
\end{array}$$

Again, we proceed as in Subsection 5.1. We choose an irreducible polynomial $\mathcal{Q}(x)$ of degree $n = 5$ and two irreducible polynomials $\mathcal{D}_1(x)$ and $\mathcal{D}_2(x)$ of degree $n + g - 1 = 6$. For instance,

$$\begin{array}{l}
\mathcal{Q}(x) = x^5 + ax^4 + x^3 + a^2x^2 + ax + 1, \\
\mathcal{D}_1(x) = x^6 + ax^4 + ax^2 + x + a^2, \\
\mathcal{D}_2(x) = x^6 + ax^3 + a^2x^2 + a^2.
\end{array}$$

The degree n place $(\mathcal{Q}(x))$ and the two degree $n + g - 1$ places $(\mathcal{D}_1(x))$ and $(\mathcal{D}_2(x))$ of $\mathbb{F}_q(x)/\mathbb{F}_q$ totally split in $F/\mathbb{F}_q$. Then we choose suitable places Q, D_1 and D_2 of $F/\mathbb{F}_q$ lying over the places $(\mathcal{Q}(x))$, $(\mathcal{D}_1(x))$ and $(\mathcal{D}_2(x))$ respectively, that is, such that $D_1 - Q$ and $D_2 - Q$ are non-special divisors of degree $g - 1$. For instance, using the description of Magma,

$$\begin{array}{l}
Q = (x^5 + ax^4 + x^3 + a^2x^2 + ax + 1, (x^3 + x + 1)y + ax^4 + a^2x^3 + ax^2 + a^2x + 1), \\
D_1 = (x^6 + ax^4 + ax^2 + x + a^2, (x^3 + x + 1)y + x^5 + a^2), \\
D_2 = (x^6 + ax^3 + a^2x^2 + a^2, (x^3 + x + 1)y + a^2x^5 + ax^2 + a^2).
\end{array}$$

As in Section 3.3.2, we choose as basis of the Riemann-Roch space $\mathcal{L}(D_i)$ the basis $\mathcal{B}_{D_i}$ such that $E_i(\mathcal{B}_{D_i}) = \mathcal{B}_Q$ is a basis of F_Q. We write $\mathcal{B}_{D_1} = (f_1, ..., f_n)$ and $\mathcal{B}_{D_2} = (f_1, f_{n+1}..., f_{2n-1})$. For $j \in \{2, \ldots, n\}$ and $k \in \{n + 1, \ldots, 2n - 1\}$, any element f_j of $\mathcal{B}_{D_1}$ and f_k of $\mathcal{B}_{D_2}$ are respectively of the form:

$$f_j(x, y) = \frac{f_{j1}(x)y + f_{j2}(x)}{\mathcal{D}_1(x)} \text{ and } f_k(x, y) = \frac{f_{k1}(x)y + f_{k2}(x)}{\mathcal{D}_2(x)},$$

where $f_{j1}, f_{j2}, f_{k1}, f_{k2} \in \mathbb{F}_4[x]$. To simplify, we set $f_j(x,y) = (f_{j1}(x), f_{j2}(x))$ for all $j \in \{2, 2n-1\}$. We have:

$$f_1(x,y) = 1,$$

$$f_2(x,y) = (a^2x^5 + a^2x^4 + ax^3 + ax + 1, a^2x^6 + a^2x^5 + a^2x^4 + a^2x^3 + x^2 + ax + a),$$

$$f_3(x,y) = (a^2x^6 + x^5 + x^3 + ax^2 + a^2, x^6 + x^4 + ax^3 + ax^2 + x + 1),$$

$$f_4(x,y) = (ax^6 + a^2x^5 + ax^4 + x^3 + a^2x^2, ax^6 + ax^5 + a^2x^3 + ax^2 + x + a),$$

$$f_5(x,y) = (x^6 + a^2x^5 + a^2x^3 + ax^2 + 1, a^2x^5 + x^4 + a^2x^2 + a^2x + a^2),$$

$$f_6(x,y) = (a^2x^6 + ax^4 + x^2 + ax + a^2, ax^6 + a^2x^4 + a^2x^3 + x),$$

$$f_7(x,y) = (x^6 + x^2 + 1, ax^4 + ax^3 + a^2x + a^2),$$

$$f_8(x,y) = (a^2x^5 + x^4 + a^2x^3 + ax^2 + x, x^6 + a^2x^5 + a^2x^4 + a^2x^3 + x^2 + a^2x + 1),$$

$$f_9(x,y) = (x^5 + x^4 + ax^3 + ax + a^2, x^6 + a^2x^5 + ax^4 + x^3 + a^2x^2 + a^2x),$$

The basis $\mathcal{B}_{D_1+D_2} = (f_1, \ldots, f_{2n+g-1})$ consists of the f_i's that are defined above for $j \in \{1, \ldots, 2n-1\}$ and the $g = 2$ components

$$f_{10}(x,y) = \frac{g_{10}(x)y + h_{10}(x)}{r(x)} \text{ and } f_{11}(x,y) = \frac{g_{11}(x)y + h_{11}(x)}{r(x)}$$

where:

$$g_{10}(x) = ax^{12} + x^{11} + a^2x^9 + a^2x^8 + ax^7 + ax^6 + a^2x^5 + x^3 + x^2 + ax,$$

$$h_{10}(x) = x^{12} + a^2x^9 + ax^8 + a^2x^7 + x^6 + ax^4 + ax^3 + x^2 + a^2x + 1),$$

$$f_{11}(x) = a^2x^{12} + a^2x^{11} + a^2x^{10} + x^7 + ax^5 + ax^3 + ax^2 + a^2x + 1,$$

$$h_{11}(x) = x^{10} + x^8 + ax^5 + a^2x^2 + 1,$$

$$r(x) = x^{12} + ax^{10} + ax^9 + x^8 + ax^7 + x^6 + a^2x^5 + ax^4 + ax^3 + a^2x^2 + a^2x + a.$$

7. Multiplication in $\mathbb{F}_{2^n/\mathbb{F}_2}$

Set $q = 2$ and $g = 2$. In this case, it is known that the number of rational places $N_q(g) = 6$ by the upper bound of Ihara [19] and the lower bound of Serre [24] and it is not sufficient to multiply in the extensions of $\mathbb{F}_2$ of degree $n \geq 3$. Hence, we need to use places of higher degree. Note that as by the above section, $N_4(g) = 10 = N_1(F/\mathbb{F}_2) + 2N_2(F/\mathbb{F}_2)$, we can only multiply in extensions of degree $n \leq 4$ if we only use places of degree one and two (for any curve of genus two) *a priori* (up to the remark of previous subsection that the condition of Theorem 2.1 is only a sufficient condition). By consequence, we set $n = 5$ and we consider the curve used in Section 6 namely the algebraic function field $F/\mathbb{F}_2$ associated to the hyperelliptic curve X with plane model $y^2 + y = \frac{x}{x^3+x+1}$, of genus two. This curve has $N_1(F/\mathbb{F}_2) = 4$ rational places, $N_2(F/\mathbb{F}_2) = 3$ places of degree two and $N_4(F/\mathbb{F}_2) = 2$ places of degree four. For the description of the places, we use the description given by Magma. The places of degree one are :

$$\begin{array}{ll} P_{\infty,1} = (\frac{1}{x}, \frac{x^3+x+1}{x^3}y + \frac{1}{x}) & P_{\infty,2} = (\frac{1}{x}, \frac{x^3+x+1}{x^3}y + \frac{x+1}{x}) \\ P_3 = (x, (x^3 + x + 1)y) & P_4 = (x, (x^3 + x + 1)y + x + 1). \end{array}$$

The places of degree two are:

$$Q_1 = (x+1) \quad Q_2 = (x^2 + x + 1, (x^3 + x + 1)y + 1) \quad Q_3 = (x^2 + x + 1, (x^3 + x + 1)y + x + 1).$$

And the places of degree four are:

$$R_1 = (x^4 + x^3 + 1, (x^3 + x + 1)y + x^2 + x + 1) \quad R_2 = (x^4 + x^3 + 1, (x^3 + x + 1)y + x^3 + x^2).$$

According to our method, we choose the irreducible polynomial $\mathcal{Q}(x) = x^5 + x^3 + 1$ of degree $n = 5$ and the two irreducible polynomials $\mathcal{D}_1(x) = x^6 + x^5 + x^4 + x + 1$ and $\mathcal{D}_2(x) = x^6 + x^5 + x^2 + x + 1$ of degree $n+g-1 = 6$. The degree n place $(\mathcal{Q}(x))$ and the two degree $n+g-1$ places $(\mathcal{D}_1(x))$ and $(\mathcal{D}_2(x))$ of $\mathbb{F}_q(x)/\mathbb{F}_q$

totally split in $F/\mathbb{F}_q$. Then we choose places Q, D_1 and D_2 of $F/\mathbb{F}_q$ lying over these three places and such that $D_1 - Q$ and $D_2 - Q$ are non-special divisors of degree $g-1$. Using the description of Magma,

$$\begin{aligned} Q &= (x^5 + x^3 + 1, (x^3 + x + 1)y + x^4 + x + 1) \\ D_1 &= (x^6 + x^5 + x^4 + x + 1, (x^3 + x + 1)y + x^5 + x^3 + 1) \\ D_2 &= (x^6 + x^5 + x^2 + x + 1, (x^3 + x + 1)y + x^5 + x^4 + x). \end{aligned}$$

As in Section 3.3.2, we choose as basis of the Riemann-Roch space $\mathcal{L}(D_i)$ the basis $\mathcal{B}_{D_i}$ such that $E_i(\mathcal{B}_{D_i}) = \mathcal{B}_Q$ is a basis of F_Q. We write $\mathcal{B}_{D_1} = (f_1, ..., f_n)$ and $\mathcal{B}_{D_2} = (f_1, f_{n+1}..., f_{2n-1})$. For $j \in \{2, \ldots, n\}$ and $k \in \{n+1, \ldots, 2n-1\}$, any element f_j of $\mathcal{B}_{D_1}$ and f_k of $\mathcal{B}_{D_2}$ are respectively of the form:

$$f_j(x,y) = \frac{f_{j1}(x)y + f_{j2}(x)}{\mathcal{D}_1(x)} \text{ and } f_k(x,y) = \frac{f_{k1}(x)y + f_{k2}(x)}{\mathcal{D}_2(x)},$$

where $f_{j1}, f_{j2}, f_{k1}, f_{k2} \in \mathbb{F}_4[x]$. To simplify, we set $f_j(x,y) = (f_{j1}(x), f_{j2}(x))$ for all $j \in \{2, 2n-1\}$. We have:

$f_1(x,y) = 1,$

$f_2(x,y) = (x^3 + x + 1, x^6 + x^4 + 1),$

$f_3(x,y) = (x^4 + x^3 + x^2 + 1, x^4 + x^2 + 1),$

$f_4(x,y) = (x^5 + x^4 + x^3 + x, x^5 + x^3 + x),$

$f_5(x,y) = (x^6 + x^5 + x^4 + x^2, x^6 + x^4 + x^2),$

$f_6(x,y) = (x^6 + x^5 + x^4 + x^3 + x^2 + x + 1, x^5),$

$f_7(x,y) = (x^6 + x^5 + x^3 + 1, x^5 + x^4 + x^2 + 1),$

$f_8(x,y) = (x^6 + x^2 + 1, x^4 + x^3 + x^2 + x + 1),$

$f_9(x,y) = (x^4 + x^3 + x^2 + 1, x^6 + x^3 + x + 1),$

The basis is completed with

$$f_{10}(x,y) = \frac{(x^{12} + x^8 + x^6 + x^5 + x^2 + x + 1)y + (x^{11} + x^8 + x^5 + x^4 + x^3 + x^2 + x)}{r(x)}$$

$$f_{11}(x,y) = \frac{(x^{12} + x^{11} + x^9 + x^8 + x^6 + x)y + (x^{12} + x^{10} + x^8 + x^7 + x^6 + x^4 + x^3 + x^2)}{r(x)}$$

with $r(x) = x^{12} + x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + 1$.

Appendix A. Magma implementation of the multiplication algorithms in the finite fields

A.1. $\mathbb{F}_{16^{13}}$ over $\mathbb{F}_{16}$.

```
// Asymmetric version of Chudnovsky multiplication algorithm in GF16^13
// using only places of degree 1

n:=13; g:=2; q:=16;
F16<a>:=GF(16);

// %%%%%%%%%%%%%%%%%%%%%%%%%%
// ALGEBRAIC FUNCTION FIELD WITH CURVE y^2 + y + x^5 OF GENUS 2
// %%%%%%%%%%%%%%%%%%%%%%%%%%

Kx<x> := FunctionField(F16);
Kxy<y> := PolynomialRing(Kx);
f:=y^2 + y + x^5;
F<c> := FunctionField(f);

//------ find places of higher degree
LP1:=Places(F,1);    // 33 degree 1 places
```

```
// %%%%%%%%%%%%%%%%%%%%%%%%%
// CHOOSE GOOD PLACE Q AND GOOD DIVISORS D1, D2
// %%%%%%%%%%%%%%%%%%%%%%%%%

//------ q<x>:=RandomIrreduciblePolynomial(F16,n);
q := x^13 + a^6 *x^12 + a^5*x^11 + a^11*x^10 + x^9 + a^12 *x^8
 + a^7*x^7 + a^7*x^5 + a^2*x^4 + a^11*x^3 + a^8*x^2 + a^6*x+a^14;
Q := Decomposition(F,Zeros(Kx!q)[1])[1];
K<b>:=ResidueClassField(Q);
"degree of Q is ", Degree(Q); // n

//------ D1 := RandomIrreduciblePolynomial(F16,n+g-1);
D1 := x^14 + a^9*x^13 + a^6*x^12 + a^7*x^11 + a^11*x^10 + a^12*x^9 + a^10*x^8 + a^6*x^7 + a^7*x^6
+ a^10*x^5 + a^14*x^4 + x^3 + x^2 + a^3*x + a;
D1:=Decomposition(F,Zeros(Kx!D1)[1])[1];
D1:=1*D1;

//------ D2:=RandomIrreduciblePolynomial(F16,n+g-1);
D2 := x^14 + x^2 + a*x + 1;
D2:=Decomposition(F,Zeros(Kx!D2)[1])[1];
D2:=1*D2;

//------ Check D1 and D2 are suitable
"D1-Q is special ? ",IsSpecial(D1-Q); // false
"dim L(D1) is ", Dimension(D1); // n
"D2-Q is special ? ", IsSpecial(D2-Q);  // false
"dim L(D2) is ", Dimension(D2); // n
"Is D1 equivalent to D2 ? ", D1 eq D2; // false
"dim L(D1+D2) is ", Dimension(D1+D2); // 27

// %%%%%%%%%%%%%%%%%%%%%%%%%
// CONSTRUCTION OF THE RIEMANN-ROCH SPACES
// %%%%%%%%%%%%%%%%%%%%%%%%%

LD1, h1 :=RiemannRochSpace(D1);
BD1 := h1(Basis(LD1));
LD2, h2 :=RiemannRochSpace(D2);
BD2 := h2(Basis(LD2));
LD1D2, h := RiemannRochSpace(D1+D2);

// %%%%%%%%%%%%%%%%%%%%%%%%%
// SET GOOD BASES
// %%%%%%%%%%%%%%%%%%%%%%%%%

//------ Construction of E1=Evalf(Q) and set a good basis for L(D1)
L:=[]; for i in [1..n] do L:=Append(L,ElementToSequence(Evaluate(BD1[i],Q))); end for;
E1:=Transpose(Matrix(L));
BasisLD1 := Matrix(F,1,n,BD1)*Matrix(F,E1^-1);
//BasisLD1;

//------ Construction of E2=Evalf(Q) and set a good basis for L(D2)
L:=[]; for i in [1..n] do L:=Append(L,ElementToSequence(Evaluate(BD2[i],Q))); end for;
E2:=Transpose(Matrix(L));
BasisLD2 := Matrix(F,1,n,BD2)*Matrix(F,E2^-1);
//BasisLD2;

//------ Merge the two previous bases to a basis of L(D1+D2)
L1 := ElementToSequence(BasisLD1);
L2 := ElementToSequence(BasisLD2);
// Concatenate L1 to L2 except the first component of L2
LL := [LD1D2!L1[i] : i in [1..n]] cat [LD1D2!L2[i] : i in [2..n]];
BasisLD1D2 := h(ExtendBasis(LL,LD1D2));

//------ In addition, we require that the two last elements of the basis are evaluated to 0
```

```
X := Transpose(Matrix(F,[ElementToSequence(Evaluate(BasisLD1D2[i],Q)): i in [2*n+g-2..2*n+g-1]]));
TT := Matrix(F,1,n,L1)*X;
BasisLD1D2[2*n+g-2] := BasisLD1D2[2*n+g-2] - ElementToSequence(TT)[1];
BasisLD1D2[2*n+g-1] := BasisLD1D2[2*n+g-1] - ElementToSequence(TT)[2];
BLD1D2 := ExtendBasis([LD1D2!BasisLD1D2[i] : i in [1..n]], LD1D2);
"Basis of BLD1D2 : ";
for i in [1..2*n+g-1] do  printf "f%o=",i; BasisLD1D2[i]; end for;
//for i in [1..2*n+g-1] do  Evaluate(BasisLD1D2[i],Q); end for;

// %%%%%%%%%%%%%%%%%%%%%%%%%%
// CONSTRUCTION OF T AND T^-1 USING PLACES OF DEGREE ONE
// %%%%%%%%%%%%%%%%%%%%%%%%%%

// the rows of T are the evaluation on the degree 1 places over F16
ST:=[];
for j:=1 to 2*n+g-1 do
    for i:=1 to 2*n+g-1 do
        ST:=Append(ST, Evaluate(BasisLD1D2[i], LP1[j]));
    end for;
end for;

T := Matrix(2*n+g-1,2*n+g-1, ST);
"Rank of T : ", Rank(T);
TI := T^-1;

// %%%%%%%%%%%%%%%%%%%%%%%%%%
// ALGORITHM FOR THE MULTIPLICATION
// %%%%%%%%%%%%%%%%%%%%%%%%%%

// ============= FUNCTION MULT =============
// @parameter : VarX, VarY are the coordinates  in a canonical basis
// of the elements of (F16)^n to multipliate
// @return : the result of VarX * VarY in the canonical basis of (F16)^n

mult := function(varX, varY)

//------ injection of the coordinates of X into L(D1+D2) using basis BLD1D2
fx := VerticalJoin(varX,ZeroMatrix(F16,n+g-1,1));

//------ injection of the coordinates of Y into L(D1+D2) using basis BLD1D2
Y1 := [varY[1,1]] cat [0 : i in [2..n]] cat [varY[i,1] : i in [2..n]] cat [0,0];
fy := Matrix(F16,2*n+g-1,1,Y1);

//------ Hadamard product  u = T(fx)*T(fy)
u:=ZeroMatrix(F16,2*n+g-1,1);
// the products are done in F16 for all coordinates
TFX:=T*fx;
TFY:=T*fy;
for i:=1 to 2*n+g-1 do u[i,1]:=TFX[i][1]*TFY[i][1]; end for;

//------ E_Q(TI(u)) : T^-1 then evaluation in Q
uu:=Matrix(F,TI*u);
result := Evaluate(Matrix(1,2*n+g-1,BasisLD1D2)*uu,Q);
return result;
end function;
// =============  END FUNCTION MULT =============

// %%%%%%%%%%%%%%%%%%%%%%%%%%
// EXAMPLES
//%%%%%%%%%%%%%%%%%%%%%%%%%%

// Example 1 : (a+b)*(1+a*b+a*b^2) = a*b^3 + a^5*b^2 + a^8*b + a
X1 := Matrix(F16,n,1,ElementToSequence(a+b));
Y1 := Matrix(F16,n,1,ElementToSequence(1+a*b+a*b^2));
```

```
mult(X1,Y1);

// Example 2 : b^5 * (1 + a^2*b^3 + b^4) = b^9 + a^2*b^8 + b^5
X2 := Matrix(F16,n,1,ElementToSequence(b^5));
Y2 := Matrix(F16,n,1,ElementToSequence(1 + a^2*b^3 + b^4));
mult(X2,Y2);

// Example 3 : (a*b + b^2 + a*b^4) * (a*b + b^2 + a*b^4) = a^2*b^8 + b^4 + a^2*b^2
X3 := Matrix(F16,n,1,ElementToSequence(a*b + b^2 + a*b^4));
Y3 := Matrix(F16,n,1,ElementToSequence(a*b + b^2 + a*b^4));
mult(X3,Y3);

// Example 4 : (a*b + b^2 + a*b^4 + b^7 + a*b^12) * (a*b + b^2 + a*b^4) =
// a*b^12 + a^8*b^11 + b^10 + a^2*b^9 + a^8*b^8 + a^3*b^7 + a^9*b^6 + a^11*b^5 +
//  a^8*b^4 + a^5*b^3 + a^3*b^2 + a^8*b + a
X4 := Matrix(F16,n,1,ElementToSequence(a*b + b^2 + a*b^4 + b^7 + a*b^12));
Y4 := Matrix(F16,n,1,ElementToSequence(a*b + b^2 + a*b^4));
mult(X4,Y4);
```

A.2. $\mathbb{F}_{4^5}$ over $\mathbb{F}_4$.

```
// Asymmetric version of Chudnovsky multiplication algorithm in GF4^5
// In this case, the curve is defined over F4 instead of F16
// and, for n:=5, we must use places of degree 2.
// Note that in GF4^4 (i.e. when n:= 4) it suffices to use degree 1 places.

n:=5; g:=2; q:=4;
F4<a>:=GF(4);

// %%%%%%%%%%%%%%%%%%%%%%%%%
// ALGEBRAIC FUNCTION FIELD WITH CURVE y^2 + y + x/(x^3 + x + 1) OF GENUS 2
// %%%%%%%%%%%%%%%%%%%%%%%%%

Kx<x> := FunctionField(F4);
Kxy<y> := PolynomialRing(Kx);
f:=y^2 + y + x/(x^3 + x + 1);
F<c> := FunctionField(f);

//------ find places of higher degree
LP1:=Places(F,1);    // 10 degree 1 places
LP2:=Places(F,2);   // 4 degree 2 places

// %%%%%%%%%%%%%%%%%%%%%%%%%
// CHOOSE GOOD PLACE Q AND GOOD DIVISORS D1, D2
// %%%%%%%%%%%%%%%%%%%%%%%%%

//------ q<x>:=RandomIrreduciblePolynomial(F4,n);
q := x^5 + a*x^4 + x^3 + a^2*x^2 + a*x + 1;
Q := Decomposition(F,Zeros(Kx!q)[1])[1];
K<b>:=ResidueClassField(Q);
"degree of Q is ", Degree(Q); // n

//------ D1 := RandomIrreduciblePolynomial(F4,n+g-1);
D1 := x^6 + a*x^4 + a*x^2 + x + a^2;
D1:=Decomposition(F,Zeros(Kx!D1)[1])[1];
D1:=1*D1;

//------ D2:=RandomIrreduciblePolynomial(F4,n+g-1);
D2 := x^6 + a*x^3 + a^2*x^2 + a^2;
D2:=Decomposition(F,Zeros(Kx!D2)[1])[1];
D2:=1*D2;

//------ Check D1 and D2 are suitable
"D1-Q is special ? ",IsSpecial(D1-Q); // false
"dim L(D1) is ", Dimension(D1); // n
```

```
"D2-Q is special ? ", IsSpecial(D2-Q);  // false
"dim L(D2) is ", Dimension(D2); // n
"Is D1 equivalent to D2 ? ", D1 eq D2; // false
"dim L(D1+D2) is ", Dimension(D1+D2); // 11

// %%%%%%%%%%%%%%%%%%%%%%%%%%
// CONSTRUCTION OF THE RIEMANN-ROCH SPACES
// %%%%%%%%%%%%%%%%%%%%%%%%%%

LD1, h1 :=RiemannRochSpace(D1);
BD1 := h1(Basis(LD1));
LD2, h2 :=RiemannRochSpace(D2);
BD2 := h2(Basis(LD2));
LD1D2, h := RiemannRochSpace(D1+D2);

// %%%%%%%%%%%%%%%%%%%%%%%%%%
// SET GOOD BASES
// %%%%%%%%%%%%%%%%%%%%%%%%%%

//------ Construction of E1=Evalf(Q) and set a good basis for L(D1)
L:=[]; for i in [1..n] do L:=Append(L,ElementToSequence(Evaluate(BD1[i],Q))); end for;
E1:=Transpose(Matrix(L));
BasisLD1 := Matrix(F,1,n,BD1)*Matrix(F,E1^-1);
//BasisLD1;

//------ Construction of E2=Evalf(Q) and set a good basis for L(D2)
L:=[]; for i in [1..n] do L:=Append(L,ElementToSequence(Evaluate(BD2[i],Q))); end for;
E2:=Transpose(Matrix(L));
BasisLD2 := Matrix(F,1,n,BD2)*Matrix(F,E2^-1);
//BasisLD2;

//------ Merge the two previous bases to a basis of L(D1+D2)
L1 := ElementToSequence(BasisLD1);
L2 := ElementToSequence(BasisLD2);
// Concatenate L1 to L2 except the first component of L2
LL := [LD1D2!L1[i] : i in [1..n]] cat [LD1D2!L2[i] : i in [2..n]];
BasisLD1D2 := h(ExtendBasis(LL,LD1D2));

//------ In addition, we require that the two last elements of the basis are evaluated to 0
X := Transpose(Matrix(F,[ElementToSequence(Evaluate(BasisLD1D2[i],Q)): i in [2*n+g-2..2*n+g-1]]));
TT := Matrix(F,1,n,L1)*X;
BasisLD1D2[2*n+g-2] := BasisLD1D2[2*n+g-2] - ElementToSequence(TT)[1];
BasisLD1D2[2*n+g-1] := BasisLD1D2[2*n+g-1] - ElementToSequence(TT)[2];
BLD1D2 := ExtendBasis([LD1D2!BasisLD1D2[i] : i in [1..n]], LD1D2);
"Vectors of BLD1D2 are independent ? ", IsIndependent(BLD1D2);
"Basis of BLD1D2 : ";
for i in [1..2*n+g-1] do  printf "f%o=",i; BasisLD1D2[i]; end for;
//for i in [1..2*n+g-1] do  Evaluate(BasisLD1D2[i],Q); end for;

// %%%%%%%%%%%%%%%%%%%%%%%%%%
// CONSTRUCTION OF T AND T^-1 USING PLACES OF DEGREE 1 AND 2
// %%%%%%%%%%%%%%%%%%%%%%%%%%

// The 9 first rows of T are the evaluation on 9 places of degree 1.
// The two last rows are the evaluation on one place of degree 2,
// since it has 2 coordinates over F4

ST:=[];
for j:=1 to 9 do
    for i:=1 to 2*n+g-1 do
        ST:=Append(ST, Evaluate(BasisLD1D2[i], LP1[j]));
    end for;
end for;
STemp:= ST;
```

```
// Choose a degree 2 place such that the 11x11-matrix T has rank 11
numPlace:=0;
for j:=1 to #LP2 do
    ST := STemp;
    ST1:=[];
    for i:=1 to 2*n+g-1 do
        eva:= ElementToSequence(Evaluate(BasisLD1D2[i], LP2[j]),F4);
        ST1:=Append(ST1, eva[1]);
        ST1:=Append(ST1, eva[2]);
    end for;
    for i:=1 to 4*n+2*g-2 by 2 do ST:=Append(ST, ST1[i]); end for;
    for i:=2 to 4*n+2*g-2 by 2 do ST:=Append(ST, ST1[i]); end for;
    T := Matrix(2*n+g-1,2*n+g-1, ST);
    if Rank(T) eq 11 then numPlace:=j; break; end if;
end for;
"num of the chosen 2 degree place : ", numPlace;
"Rank of T : ", Rank(T);
TI := T^-1;

// %%%%%%%%%%%%%%%%%%%%%%%%%%
// ALGORITHM FOR THE MULTIPLICATION
// %%%%%%%%%%%%%%%%%%%%%%%%%%

// ============= FUNCTION MULT =============
// @parameter : VarX, VarY are the coordinates  in a canonical basis
// of the elements of (F4)^n to multipliate
// @return : the result of VarX * VarY in the canonical basis of (F4)^n

mult := function(varX, varY)

//------ injection of the coordinates of X into L(D1+D2) using basis BLD1D2
fx := VerticalJoin(varX,ZeroMatrix(F4,n+g-1,1));

//------ injection of the coordinates of Y into L(D1+D2) using basis BLD1D2
Y1 := [varY[1,1]] cat [0 : i in [2..n]] cat [varY[i,1] : i in [2..n]] cat [0,0];
fy := Matrix(F4,2*n+g-1,1,Y1);

//------ Hadamard product  u = T(fx)*T(fy)
u:=ZeroMatrix(F4,2*n+g-1,1);
TFX:=T*fx;
TFY:=T*fy;

// the products are done in F4 for the 9 first coordinates
for i:=1 to 9 do u[i,1]:=TFX[i][1]*TFY[i][1]; end for;

// and over F16 for the other coordinates taken 2 by 2
KK<hh>:=Parent(Evaluate(BasisLD1D2[2], LP2[numPlace]));
mb:=KK![TFX[10][1], TFX[11][1]]* KK![TFY[10][1], TFY[11][1]];
mp:=ElementToSequence(mb,F4);
u[10,1]:=mp[1];
u[11,1]:=mp[2];

//------ E_Q(TI(u)) : T^-1 then evaluation in Q
uu:=Matrix(F,TI*u);
result := Evaluate(Matrix(1,2*n+g-1,BasisLD1D2)*uu,Q);
return result;
end function;
// =============  END FUNCTION MULT =============

// %%%%%%%%%%%%%%%%%%%%%%%%%%
// EXAMPLES
// %%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
// Example 1 : (a+b)*(1+a*b+a*b^2) = a*b^3 + b^2 + a*b + a
X1 := Matrix(F4,n,1,ElementToSequence(a+b));
Y1 := Matrix(F4,n,1,ElementToSequence(1+a*b+a*b^2));
mult(X1,Y1);

// Example 2 : b^5 * (1 + a^2*b^3 + b^4) = b^4 + a^2*b^2 + a
X2 := Matrix(F4,n,1,ElementToSequence(b^5));
Y2 := Matrix(F4,n,1,ElementToSequence(1 + a^2*b^3 + b^4));
mult(X2,Y2);

// Example 3 : (a*b + b^2 + a*b^4) * (a*b + b^2 + a*b^4) = a^2*b^3 + a^2*b^2 + a^2*b + 1
X3 := Matrix(F4,n,1,ElementToSequence(a*b + b^2 + a*b^4));
Y3 := Matrix(F4,n,1,ElementToSequence(a*b + b^2 + a*b^4));
mult(X3,Y3);
```

A.3. $\mathbb{F}_{2^5}$ over $\mathbb{F}_2$.

```
// Same curve but defined over GF2.
// We use 3 degree 1 places, 2 degree 2 places and 1 degree 4 place

n:=5; g:=2; q:=2;
F2:=GF(2);

// %%%%%%%%%%%%%%%%%%%%%%%%%%
// ALGEBRAIC FUNCTION FIELD WITH CURVE y^2 + y + x/(x^3 + x + 1) OF GENUS 2
// %%%%%%%%%%%%%%%%%%%%%%%%%%

Kx<x> := FunctionField(F2);
Kxy<y> := PolynomialRing(Kx);
f:=y^2 + y + x/(x^3+x+1);
F<c> := FunctionField(f);

//------ find places of higher degree
LP1:=Places(F,1);    // 3 degree 1 places
LP2:=Places(F,2);   // 1 degree 2 places
LP4:=Places(F,4);   // 7 degree 4 places
#LP1;#LP2;#LP4;

// %%%%%%%%%%%%%%%%%%%%%%%%%%
// CHOOSE GOOD PLACE Q AND GOOD DIVISORS D1, D2
// %%%%%%%%%%%%%%%%%%%%%%%%%%

//------ q<x>:= RandomIrreduciblePolynomial(F2,n);
q := x^5 + x^3 + 1;
Q := Decomposition(F,Zeros(Kx!q)[1])[1];
K<b>:=ResidueClassField(Q);

//------ D1 := RandomIrreduciblePolynomial(F2,n+g-1); D1;
D1 := x^6 + x^5 + x^4 + x + 1;
D1:=Decomposition(F,Zeros(Kx!D1)[1])[1];
D1:=1*D1;

//------ D2:=RandomIrreduciblePolynomial(F2,n+g-1);
D2 := x^6 + x^5 + x^2 + x + 1;
D2:=Decomposition(F,Zeros(Kx!D2)[1])[1];
D2:=1*D2;

//------ Check D1 and D2 are suitable
"D1-Q is special ? ",IsSpecial(D1-Q); // false
"dim L(D1) is ", Dimension(D1); // n
"D2-Q is special ? ", IsSpecial(D2-Q);  // false
"dim L(D2) is ", Dimension(D2); // n
"Is D1 equivalent to D2 ? ", D1 eq D2; // false
"dim L(D1+D2) is ", Dimension(D1+D2); // 11
```

```
// %%%%%%%%%%%%%%%%%%%%%%%%%%
// CONSTRUCTION OF THE RIEMANN-ROCH SPACES
// %%%%%%%%%%%%%%%%%%%%%%%%%%

LD1, h1 :=RiemannRochSpace(D1);
BD1 := h1(Basis(LD1));
LD2, h2 :=RiemannRochSpace(D2);
BD2 := h2(Basis(LD2));
LD1D2, h := RiemannRochSpace(D1+D2);

// %%%%%%%%%%%%%%%%%%%%%%%%%%
// SET GOOD BASES
// %%%%%%%%%%%%%%%%%%%%%%%%%%

//------ Construction of E1 : E1=Evalf(Q) and set a good basis for L(D1)
L:=[]; for i in [1..n] do L:=Append(L,ElementToSequence(Evaluate(BD1[i],Q))); end for;
E1:=Transpose(Matrix(L));
BasisLD1 := Matrix(F,1,n,BD1)*Matrix(F,E1^-1);
//BasisLD1;

//------ Construction of E2 : E2=Evalf(Q) and set a good basis for L(D2)
L:=[]; for i in [1..n] do L:=Append(L,ElementToSequence(Evaluate(BD2[i],Q))); end for;
E2:=Transpose(Matrix(L));
BasisLD2 := Matrix(F,1,n,BD2)*Matrix(F,E2^-1);
//BasisLD2;

//------ Merge the two previous bases to a basis of L(D1+D2)
L1 := ElementToSequence(BasisLD1);
L2 := ElementToSequence(BasisLD2);
// Concatenate L1 to L2 except the first component of L2
LL := [LD1D2!L1[i] : i in [1..n]] cat [LD1D2!L2[i] : i in [2..n]];
BasisLD1D2 := h(ExtendBasis(LL,LD1D2));

//------ In addition, we require that the two last elements of the basis are evaluated to 0
X := Transpose(Matrix(F,[ElementToSequence(Evaluate(BasisLD1D2[i],Q)): i in [2*n+g-2..2*n+g-1]]));
TT := Matrix(F,1,n,L1)*X;
BasisLD1D2[2*n+g-2] := BasisLD1D2[2*n+g-2] - ElementToSequence(TT)[1];
BasisLD1D2[2*n+g-1] := BasisLD1D2[2*n+g-1] - ElementToSequence(TT)[2];
BLD1D2 := ExtendBasis([LD1D2!BasisLD1D2[i] : i in [1..n]], LD1D2);
"Vectors of BLD1D2 are independent ? ", IsIndependent(BLD1D2);
"Basis of BLD1D2 : ";
for i in [1..2*n+g-1] do  printf "f%o=",i; BasisLD1D2[i]; end for;
//for i in [1..2*n+g-1] do  Evaluate(BasisLD1D2[i],Q); end for;

// %%%%%%%%%%%%%%%%%%%%%%%%%%
// CONSTRUCTION OF T AND T^-1 USING PLACES OF DEGREE 1, 2 and 4
// %%%%%%%%%%%%%%%%%%%%%%%%%%

// The 3 first rows of T are the evaluation on the degree 1 places.
// The 4 next rows are the evaluation on 2 places of degree;
// each evaluation is on 2 rows since it has 2 coordinates over F2
// Finally, the 4 last rows are the evaluation on 1 place of degree 4;
// this evaluation is on 4 rows since it has 4 coordinates over F2

ST:=[];
for j:=1 to 3 do
    for i:=1 to 2*n+g-1 do
        ST:=Append(ST, Evaluate(BasisLD1D2[i], LP1[j]));
    end for;
end for;

ST2:=[];
for j:=1 to 2 do
```

```
    ST2:=[];
    for i:=1 to 2*n+g-1 do
        eva:= ElementToSequence(Evaluate(BasisLD1D2[i], LP2[j]),F2);
        ST2:=Append(ST2, eva[1]);
        ST2:=Append(ST2, eva[2]);
    end for;
    for i:=1 to 4*n+2*g-2 by 2 do ST:=Append(ST, ST2[i]); end for;
    for i:=2 to 4*n+2*g-2 by 2 do ST:=Append(ST, ST2[i]); end for;
end for;

ST4:=[];
for i:=1 to 2*n+g-1 do
    eva:= ElementToSequence(Evaluate(BasisLD1D2[i], LP4[2]),F2);
    ST4:=Append(ST4, eva[1]);
    ST4:=Append(ST4, eva[2]);
    ST4:=Append(ST4, eva[3]);
    ST4:=Append(ST4, eva[4]);
end for;

for i:=1 to 8*n+4*g-4 by 4 do ST:=Append(ST, ST4[i]); end for;
for i:=2 to 8*n+4*g-4 by 4 do ST:=Append(ST, ST4[i]); end for;
for i:=3 to 8*n+4*g-4 by 4 do ST:=Append(ST, ST4[i]); end for;
for i:=4 to 8*n+4*g-4 by 4 do ST:=Append(ST, ST4[i]); end for;

T := Matrix(2*n+g-1,2*n+g-1, ST);
"Rank of T : ", Rank(T);
TI := T^-1;

// %%%%%%%%%%%%%%%%%%%%%%%%%%
// ALGORITHM FOR THE MULTIPLICATION
// %%%%%%%%%%%%%%%%%%%%%%%%%%

// ============= FUNCTION MULT =============
// @parameter : VarX, VarY are the coordinates  in a canonic basis
// of the elements of (F4)^n to multiply
// @return : the result of VarX * VarY in the canonic basis of (F2)^n

mult := function(varX, varY)

//------ injection of the coordinates of X into L(D1+D2) using basis BLD1D2
fx := VerticalJoin(varX,ZeroMatrix(F2,n+g-1,1));

//------ injection of the coordinates of Y into L(D1+D2) using basis BLD1D2
Y1 := [varY[1,1]] cat [0 : i in [2..n]] cat [varY[i,1] : i in [2..n]] cat [0,0];
fy := Matrix(F2,2*n+g-1,1,Y1);

//------ Hadamard product  u = T(fx)*T(fy)
u:=ZeroMatrix(F2,2*n+g-1,1);
TFX:=T*fx;
TFY:=T*fy;

// the products are done in F2 for the 3 first coordinates
for i:=1 to 3 do u[i,1]:=TFX[i][1]*TFY[i][1]; end for; // degree 1 places

// over F4 for the next 4 coordinates taken 2 by 2
for i:=4 to 7 by 2 do  // degree 2 places
KK<hh>:=Parent(Evaluate(BasisLD1D2[2], LP2[(i-2) div 2]));
mb:=KK![TFX[i][1], TFX[i+1][1]]* KK![TFY[i][1], TFY[i+1][1]];
mp:=ElementToSequence(mb,F2);
u[i,1]:=mp[1];
u[i+1,1]:=mp[2];
end for;

// and over F16 for the other coordinates taken 4 by 4
```

```
KK4<hh4>:=Parent(Evaluate(BasisLD1D2[2], LP4[ 2] ));  // degree 4 place
mb:=KK4![TFX[8][1], TFX[9][1], TFX[10][1], TFX[11][1]]
    * KK4![TFY[8][1], TFY[9][1], TFY[10][1], TFY[11][1]];
mp:=ElementToSequence(mb,F2);
u[8,1]:= mp[1];
u[9,1]:= mp[2];
u[10,1]:= mp[3];
u[11,1]:= mp[4];

//------ E_Q(TI(u)) : T^-1 then evaluation in Q
uu:=Matrix(F,TI*u);
result := Evaluate(Matrix(1,2*n+g-1,BasisLD1D2)*uu,Q);
return result;
end function;
// =========  END FUNCTION MULT =========

// %%%%%%%%%%%%%%%%%%%%%%%%%
// EXAMPLES
//%%%%%%%%%%%%%%%%%%%%%%%%%

// Example 1 : (1+b)*(1+b+b^2) = b^5
X1 := Matrix(F2,n,1,ElementToSequence(1+b));
Y1 := Matrix(F2,n,1,ElementToSequence(1+b+b^2));
mult(X1,Y1);

// Example 2 : b^4*(b^2+b^3) = b^20
X2 := Matrix(F2,n,1,ElementToSequence(b^4));
Y2 := Matrix(F2,n,1,ElementToSequence(b^2+b^3));
mult(X2,Y2);

// Example 3 : (1+b+b^3)*(1+b+b^3) = b^21
X3 := Matrix(F2,n,1,ElementToSequence(1+b+b^3));
Y3 := Matrix(F2,n,1,ElementToSequence(1+b+b^3));
mult(X3,Y3);
```

References

[1] N. Arnaud. Évaluation Dérivées, Multiplication dans les Corps Finis et Codes Correcteurs. *PhD Thesis*, 2006. Université de la Méditerranée, Institut de Mathématiques de Luminy.

[2] K. Atighehchi, S. Ballet, A. Bonnecaze, R. Rolland. On Chudnovsky-Based Arithmetic Algorithms in Finite Fields. Mathematics of Computation, to appear.

[3] S. Ballet, A. Bonnecaze, M. Tukumuli. On the construction of elliptic Chudnovsky-type algorithms for multiplication in large extensions of finite fields. *Journal of Algebra and Its Applications*, Vol. 15, No. 1 (2016) 1650005.

[4] S. Ballet. Curves with many points and multiplication complexity in any extension of $\mathbb{F}_q$. *Finite Fields and their Applications*, 5(4), 364-377, 1999.

[5] S. Ballet. Quasi-optimal algorithms for multiplication in the extensions of $\mathbb{F}_{16}$ of degree $13, 14$ and 15. *Journal of Pure and Applied Algebra*, 171(2-3), 149-164, 2002.

[6] S. Ballet and D. Le Brigand. On the existence of non special divisor of degree g and $g-1$ in algebraic function fields over $\mathbb{F}_q$. *Journal of Number Theory*, 116, 293-310, 2006.

[7] S. Ballet and J. Pieltant. On the tensor rank of multiplication in any extension of $\mathbb{F}_2$. *Journal of Complexity*, 27, 230-245, 2011.

[8] S. Ballet and R. Rolland. Multiplication algorithm in a finite field and tensor rank of the multiplication. *Journal of Algebra*, 272/1, 173-185, 2004.

[9] S. Ballet and R. Rolland. Families of curves over any finite field attaining the generalized Drinfeld-Vladut bound. In *Actes de la Conférence "Théorie des Nombres et Applications", 5–18, Publ. Math. Besançon Algèbre Théorie Nr., Presses Univ. Franche-Comté, Besançon*, 2011.

[10] S. Ballet, D. Le Brigand and R. Rolland On an application of the definition field descent of a tower of function fields. Arithmetics, geometry, and coding theory (AGCT 2005), 187–203, Sémin. Congr., 21, Soc. Math. France, Paris, 2010.

[11] S. Ballet, J. Chaumine and J. Pieltant Shimura modular curves and asymptotic symmetric tensor rank of multiplication in any finite field. In *Proceedings of the Conference Algebraic informatics, Lecture Notes in Comput. Sci., 8080, Springer, Heidelberg, 160–172*, 2013.

[12] S. Ballet, J. Pieltant, M. Rambaud and J. Sisjling On some bounds for symmetric tensor rank of multiplication in finite fields. In *Arithmetic, geometry, cryptography and coding theory, Contemporary Mathematics, Amer. Math. Soc., Providence, RI*, 2017.

[13] S. Ballet, C. Ritzenthaler and R. Rolland. On the existence of dimension zero divisors in algebraic function fields defined over $\mathbb{F}_q$ *Acta Arithmetica*, 143 (4), 377-392, 2010.

[14] U. Baum and M. A. Shokrollahi. An optimal algorithm for multiplication in $\mathbb{F}_{256}/\mathbb{F}_4$. *Applicable Algebra in Engineering, Communication and Computing*, 2:15–20, 1991.
[15] W. Bosma, J. Cannon and C. Playoust. The Magma Algebra System I. The user language. *Journal of Symbolic Computation 24*, 3-4, 235-265, 1957.
[16] N. H. Bshouty. Multinlinear complexity is equivalent to optimal tester size. *Electronic Colloquium on Computational Complexity*, Report No11, 2013.
[17] M. Cenk and F. Özbudak. On multiplication in finite fields. *Journal of Complexity*, 26, 172-186, 2010.
[18] D. V. and G. V. Chudnovsky. Algebraic complexities and algebraic curves over finite fields. *Journal of Complexity*, 4, 285-316, 1988.
[19] Y. Ihara. Some remarks on the number of rational points of algebraic curves. *J. Fac. Sci. Univ. Tokyo Sect. IA Math.*, 28 (1981) 721–724 (1982).
[20] R. Lidl and H. Niederreiter Finite fields. *Encyclopedia of Mathematics and Its applications,* Cambridge University Press, volume 20, 2000.
[21] Pieltant, Julia. Tours de corps de fonctions algébriques et rang de tenseur de la multiplication dans les corps finis, *PhD of Université d'Aix-Marseille*, Institut de Mathématiques de Luminy, 2012.
[22] J. Pieltant and H. Randriambololona. New uniform and asymptotic upper bounds on the tensor rank of multiplication in extensions of finite fields. *Mathematics of Computation*, 84, 2023–2045, 2015.
[23] H. Randriambololona. Bilinear complexity of algebras and the Chudnovsky-Chudnovsky interpolation method. *Journal of Complexity*, 28, 489-517, 2012.
[24] J-P. Serre. Sur le nombre de points rationnels d'une courbe algébrique sur un corps fini. *C. R. Acad. Sci. Paris*, Sér. I Math. 296.9, 397-402, 1983.
[25] G. V. Shabat. Curves with many points. *PhD Thesis*, Amsterdam, 2001.
[26] I. Shparlinski, M. Tsfasman, and S. Vladut. Curves with many points and multiplication in finite fields. In H. Stichtenoth and M.A. Tsfasman, editors, *Coding Theory and Algebraic Geometry*, number 1518 in Lectures Notes in Mathematics, pages 145–169, Berlin, 1992. Springer-Verlag. Proceedings of AGCT-3 conference, June 17-21, 1991, Luminy.
[27] H. Stichtenoth. *Algebraic Function Fields and Codes*. Berlin. Springer, 1993.
[28] S. Winograd. On Multiplication in Algebraic Extension Fields. *Theoretical Computer Science*, 8:359–377, 1979.

Aix Marseille Univ, CNRS, Centrale Marseille, I2M, Marseille, France

Aix Marseille Univ, CNRS, Centrale Marseille, LIF, Marseille, France
E-mail address: `First Name.Last Name@univ-amu.fr`