https://doi.org/10.35662/unine-thesis-2631

Trace-zero subgroups of elliptic and twisted Edwards curves: a study for cryptographic applications

THÈSE DE DOCTORAT

Candidat:

Giulia Bianco

Directrice de thèse:

Prof. Elisa Gorla, Université de Neuchâtel

Rapporteurs:

Dr. Peter Birkner, Federal Office for Information Security (BSI)
Dr. Hugues Mercier, Université de Neuchâtel

Année Académique 2016-2017

Institut de Mathématiques, Université de Neuchâtel
Rue Emile-Argand 11, CH-2000 Neuchâtel (Suisse)

UNIVERSITÉ DE NEUCHÂTEL

Faculté des Sciences
Secrétariat-décanat de Faculté
Rue Emile-Argand 11
2000 Neuchâtel – Suisse
Tél : + 41 (0)32 718 21 00
E-mail : secretariat.sciences@unine.ch

IMPRIMATUR POUR THESE DE DOCTORAT

La Faculté des sciences de l'Université de Neuchâtel autorise l'impression de la présente thèse soutenue par

Madame Giulia BIANCO

Titre:

"Trace-Zero subgroups of elliptic and twisted Edwards curves: a study for cryptographic applications"

sur le rapport des membres du jury composé comme suit:

- Prof. Elisa Gorla, directrice de thèse, Université de Neuchâtel, Suisse
- Dr Hugues Mercier, Université de Neuchâtel, Suisse
- Dr Peter Birkner, Federal Office of information security, BSI, Bonn, Allemagne

Neuchâtel, le 12 décembre 2017

Le Doyen, Prof. R. Bshary

Al Nonno Sandro

Clever talk alarmed her, and withered her delicate imaginings;
it was the social counterpart of a motor-car, all jerks,
and she was a wisp of hay, a flower.

E. M. Forster, *Howards End*

C'era una volta una ragazzina che, quando doveva risolvere un problema,
diventava ansiosa, e allora chiedeva aiuto alla " Regina degli animali ".
Al suo richiamo lupi e balene accorrevano con il loro flusso magico.
Ma non era necessario il loro intervento perché la ragazzina era molto brava.

La ragazzina adesso è una giovane donna.
Un ultimo esame da superare per terminare il suo brillante dottorato.
" Regina degli animali, mi serve aiuto da lupi e balene "
Seduta sugli scogli la Regina lancia il suo richiamo:
" Balene, la mia principessa ha bisogno di voi ! "
Poi chiama il suo amico Eolo e si fa trasportare nella foresta,
lanciando il suo richiamo ai lupi.
Tutti rispondono alla sua richiesta di aiuto.
" Ma in tutti questi anni il nostro flusso magnetico non è servito a niente"
(dice il vecchio lupo e la balena acconsente)
" La principessa è bravissima. Questa è l'ultima volta che veniamo da lei!!"
Non possiamo perdere tempo con ci chiama solo perché è in ansia .
Addio principessa, puoi benissimo affrontare da sola la tua vita!.

Nonna Adriana

Summary

In 1999, Frey first suggested trace-zero subgroups of elliptic curves for applications to cryptography, as a valid alternative to the use of classical groups of points of elliptic curves in this sector. Take an elliptic curve E defined over a finite field $\mathbb{F}_q$, with the standard addition between points. Given a field extension $\mathbb{F}_q \subseteq \mathbb{F}_{q^n}$ of odd prime degree n, the trace-zero subgroup of the elliptic curve E of degree n is a subgroup of the group of points of E with coordinates in $\mathbb{F}_{q^n}$.

In 2007, Edwards curves were introduced by Edwards, and proposed for cryptographic applications by Bernstein and Lange. Right afterwards, they were generalized to twisted Edwards curves. Twisted Edwards curves can be seen as special elliptic curves, written in a new form. They have some cryptographic advantages over elliptic curves in the usual Weierstrass form. Trace-zero subgroups of twisted Edwards curves are defined in the same way as trace-zero subgroups of elliptic curves.

In this thesis, we study trace-zero subgroups of elliptic and twisted Edwards curves, from the point of view of their potential application to cryptography. In particular, we focus on three distinct aspects for trace-zero subgroups: the use of optimal representations for group elements, the construction of fast algorithms for scalar multiplication, and the study of possible cryptographic attacks based on the discrete logarithm problem. All these aspects are of great relevance for the efficiency and the security of a cryptosystem built on the given group.

Concerning optimal representations for group elements, we propose two optimal representations for trace-zero subgroups of twisted Edwards curves. We give efficient algorithms to deal with them, and we make comparisons with analogous representations for trace-zero subgroups of elliptic curves.

Concerning efficient arithmetic in trace-zero subgroups, our contribution consists of an algorithm to perform scalar multiplication in the trace-zero subgroup of degree three. This algorithm follows an original approach and makes use of optimal coordinates to represent the elements of the group.

Finally, we focus on the study of security of trace-zero subgroups against cryptographic attacks. We propose a new variant of the index calculus algorithm for the discrete logarithm problem in these subgroups. We compare the complexity of solving the polynomial systems obtained with our method with that of solving the systems computed with the only other specialization of the index calculus to trace-zero subgroups. We show that our systems are easier to solve in the case of trace-zero subgroups of small degree, that is the important case for cryptographic applications.

In both the algorithm for scalar multiplication in trace-zero subgroups of degree three, and in our new index calculus method for trace-zero subgroups, we make use of generalized summation polynomials of elliptic curves. Such polynomials are introduced in this thesis for the first time, and they can be seen as an original generalization of Semaev's summation polynomials of elliptic curves. Generalized summation polynomials allow to find relations between points on the elliptic curve. Thanks to their geometric property, they can have relevant applications to cryptography.

Keywords. Cryptography, elliptic curves, twisted Edwards curves, optimal representations for group elements, compression and decompression, Semaev's summation polynomials, generalized summation polynomials, efficient scalar product, discrete logarithm problem, index calculus.

Resumé

En 1999, Frey a proposé, pour la première fois, les sous-groupes de trace nulle des courbes elliptiques pour les applications à la cryptographie: il les a désignés comme une alternative valide, dans ce secteur, aux groupes classiques des points des courbes elliptiques. Considérons une courbe elliptique E définie sur un corps fini $\mathbb{F}_q$, avec l'addition standard entre ses points. Etant donnée une extension de corps $\mathbb{F}_q \subseteq \mathbb{F}_{q^n}$ de degré n premier et impair, le sous groupe de trace nulle de la courbe elliptique E, de degré n, est un sous-groupe du groupe des points de E avec coordonnées dans $\mathbb{F}_{q^n}$.

En 2007, les courbes d'Edwards ont été introduites par Edwards, et elles ont été proposées pour les applications cryptographiques par Bernstein et Lange. Après, elles ont été généralisées aux courbes d'Edwards tordues. Les courbes d'Edwards tordues peuvent être considérées comme des courbes elliptiques speciales, éctrites sous une nouvelle forme. Elles ont des advantages cryptographiques sur les courbes elliptiques dans la forme usuelle de Weierstrass. Les sous-groupes de trace nulle des courbes d'Edwards tordues sont definies de la même manière que les sous-groupes de trace nulle des courbes elliptiques.

Dans cette thèse, nous étudions les sous-groupes de trace nulle des courbes elliptiques et des courbes d'Edwards tordues, du point de vue de leur application potentielle à la cryptographie. En particulier, nous nous concentrons sur trois aspects distincts pour les sous-groupes de trace nulle: l'utilisation de représentations optimales des éléments du groupe, la construction d'algorithmes pour le produit scalaire, et l'étude de possibles attaques cryptographiques basés sur le problème du logarithme discret. Tous ces aspects sont tres importants pour l'efficacité et la sécurité d'un cryptosystème construit sur le groupe considéré.

À propos de représentations optimales de groupes, nous proposons deux représentations optimales pour les sous-groupes de trace nulle des courbes d'Edwards tordues. Nous donnons des algorithmes efficaces pour l'utilisasion de nos représentations, et nous faisons des comparaisons avec les représentations analogues pour les sous-groupes de trace nulle des curbes elliptiques.

En ce qui concerne l'arithmétique efficace dans les sous-groupes de trace nulle, notre contribution consiste en un algorithme pour effectuer le produit scalaire dans les sous-groupes de trace nulle de degré trois. Cet algorithme suit une approche originale et il utilise des coordonnées optimales pour représenter les éléments du groupe.

Enfin, nous nous concentrons sur la sécurité des sous-groupes de trace nulle contre les attaques cryptographiques. Nous présentons une nouvelle variante de l'algorithme d'index calculus pour le problème du logarithme discret dans ces sous-groupes. Nous comparons

la complexité de la résolution des systèmes polynômiaux obtenus avec notre méthode à celle de la résolution des systèmes construits avec l'unique autre spécialisation d'index calculus aux sous-groupes de trace nulle. Nous montrons que nos systèmes sont plus faciles à résoudre, dans les cas de sous-groupes de trace nulle de degré petit, qui sont les cas importantes pour les applications cryptographiques.

Dans l'algorithme pour le produit scalaire dans les sous-groupes de trace nulle de degré trois, et dans notre nouvelle méthode d'index calculus, nous utilisons les polynômes de sommation généralisés de courbes elliptiques. Ces polynômes sont présentés dans cette thèse pour la première fois, et ils peuvent être vus comme une généralisation originale des polynômes de sommation de courbes elliptiques de Semaev. Les polynômes de sommation généralisés permettent de trouver des relations entre points sur la courbe elliptique. Grâce à leur propriété géométrique, ils peuvent avoir des applications pertinentes en cryptographie.

Mots clés. Cryptographie, courbes elliptiques, courbes d'Edwards tordues, représentations optimales de groupes, compression et décompression, polynômes de sommation de Semaev, polynômes de sommation généralisés, produit scalaire efficace, problème du logarithme discret, index calculus.

List of Symbols and Notation

$\mathbb{Z}$	The integers.	p. 2
$\mathbb{Z}_{>n}$	The integers bigger than $n \in \mathbb{Z}$.	p. 2
$\mathbb{Z}_{\geq n}$	The integers bigger or equal than $n \in \mathbb{Z}$.	p. 2
$\mathbb{Z}_N$	The integers modulo N, $N \in \mathbb{Z}_{>1}$.	p. 4
$\mathbb{F}_q$	A finite field with q elements (q prime power).	p. 6
$\langle P \rangle$	Cyclic group generated by P.	p. 2
DLP	Discrete Logarithm Problem.	p. 3
$\log_P(Q)$	Discrete logarithm of $Q \in \langle P \rangle$ w.r.t. P.	p. 3
$O(), o()$	Big-O notation, small-o notation.	p. 3
$\Omega()$	Big-Ω notation.	p. 3
$L_N(\alpha, c)$	$\exp\left((c + o(1))(\log N)^{\alpha}(\log\log N)^{1-\alpha}\right)$, $0 \leq \alpha \leq 1$, $c > 0$.	p. 3
$\mathcal{R} = (\mathcal{R}_G)_{G \in \mathcal{G}}$	Optimal representation for $\mathcal{G}$, $\mathcal{G}$ family of finite groups.	p. 5
$\mathbb{K}$, $\overline{\mathbb{K}}$, $\mathbb{K} \subseteq \mathbb{L} \subseteq \overline{\mathbb{K}}$	A field, its algebraic closure, a field extension.	p. 7
$\mathbb{K}[x_1, \cdots, x_n]$	The ring of polynomials with coefficients in $\mathbb{K}$ and $n \in \mathbb{Z}_{\geq 1}$ variables x_i.	p. 30
$\mathbb{K}[x, y]$, $\mathbb{K}[x, y, z]$	Ring of polynomials with coefficients in $\mathbb{K}$ and variables x, y (resp. x, y, z).	pp. 7, 8
$\deg(f)$	Total degree of $f \in \mathbb{K}[x_1, \cdots, x_n]$.	p. 7

$\deg_{x_i} f$	Degree of $f \in \mathbb{K}[x_1, \cdots, x_n]$ in x_i.	p. 89
$f^h(x_1, \cdots, x_n, x_{n+1})$	Homogenization of $f(x_1, \cdots, x_n)$ w.r.t. x_{n+1}.	p. 9
$I = (f_1, \cdots, f_s)$	Ideal generated by $f_i \in \mathbb{K}[x_1, \cdots, x_n]$.	p. 124
I^h	Homogeneous ideal $I^h = (f_1^h, \cdots, f_s^h)$.	p. 125
$\mathbb{A}^2(\overline{\mathbb{K}})$, $\mathbb{A}^2$	Affine plane over $\overline{\mathbb{K}}$.	p. 7
$\mathbb{P}^2(\overline{\mathbb{K}})$, $\mathbb{P}^2$	Projective plane over $\overline{\mathbb{K}}$.	p. 8
$\mathbb{A}^n(\overline{\mathbb{K}})$, $\mathbb{A}^n$	Affine space of dimension n over $\overline{\mathbb{K}}$, $n \in \mathbb{Z}_{\geq 2}$	p. 30
$\mathbb{P}^n(\overline{\mathbb{K}})$, $\mathbb{P}^n$	Projective space of dimension n over $\overline{\mathbb{K}}$, $n \in \mathbb{Z}_{\geq 2}$	p. 30
$C_a : f(x, y) = 0$	Affine plane curve defined by the polynomial $f(x, y) \in \mathbb{K}[x, y]$.	p. 7
$C : F(x, y, z) = 0$	Projective plane curve defined by $F(x, y, z) \in \mathbb{K}[x, y, z]$.	p. 8
$C_a(\mathbb{L})$, $C(\mathbb{L})$	$\mathbb{L}$-rational points of C_a (resp. C).	pp. 7, 8
$\overline{C_a}$	Projective closure of C_a.	p. 9
C_x^*, C_y^*, C_z^*	Affine dehomogenization of C w.r.t. x, y or z.	p. 9
$\mathbb{L}[C_a]$	$\mathbb{L}$-coordinate ring of C_a.	p. 10
$\mathbb{L}(C_a)$, $\mathbb{L}(C)$	$\mathbb{L}$-rational functions of C_a, C.	pp. 10, 11
$\mathcal{O}_P(C)$	Local ring of $C : f(x, y) = 0$ at P, C absolutely irreducible affine curve.	p. 13
$\mathcal{M}_P(C)$	Maximal ideal of $\mathcal{O}_P(C)$.	p. 13
ord_P	Order function of $\mathcal{O}_P(C)$, P nonsingular point of C.	p. 13
$\mathrm{Div}(X)$	Group of divisors of X absolutely irreducible, nonsingular projective curve.	p. 15
$\mathrm{div}(r)$	Divisor of r rational function of X.	p. 15
$\mathrm{Div}^0(X)$	Group of divisors of X of degree 0.	p. 15
$\mathrm{Princ}(X)$	Group of principal divisors of X.	p. 15

$\mathrm{Pic}^0(X)$	Degree zero Picard group of X.	p. 15
$\mathrm{Pic}^0(X)(\mathbb{L})$	Group of $\mathbb{L}$-rational divisor classes of X.	p. 15
$\mathcal{L}(D)$	Vector space associated to $D \in \mathrm{Div}(X)$.	p. 16
$\ell(D)$	Dimension of $\mathcal{L}(D)$.	p. 16
g	Genus of an absolutely irreducible projective curve.	p. 16
E_W	Elliptic curve in Weierstrass form.	p. 19
E_M	Elliptic curve in Montgomery form.	p. 20
P_∞	Point at infinity of an elliptic curve.	p. 20
$E_{a,d}$	Twisted Edwards curve with parameters a, d.	p. 24
Ω_1, Ω_2	The two points at infinity of $E_{a,d}$.	p. 24
E	Elliptic curve, or $E = (E_{a,d})^*_z$.	p. 20, 25
$\oplus$	Point addition on elliptic curves or on twisted Edwards curves.	pp. 20, 27
$-P$	Inverse of a point P w.r.t. $\oplus$.	pp. 20, 25
$\mathcal{O}$	Neutral element of $\oplus$.	p. 27
φ	Frobenius endomorphism of E.	p. 22
χ_φ	Characteristic polynomial of φ.	p. 22
$f_n(x_1, \cdots, x_n)$	The n-th summation polynomial of E_W.	p. 27
$f_n(y_1, \cdots, y_n)$	The n-th summation polynomial of $E_{a,d}$.	p. 27
V_a	Affine algebraic set.	p. 30
V_p	Projective algebraic set.	p. 30
$\overline{V_a}$	Projective closure of V_a.	p. 30
$(V_p)^*_{x_{n+1}}$	Affine dehomogenization of V_p w.r.t. x_{n+1}.	p. 30

$\mathbb{L}(V_a)$	$\mathbb{L}$-rational functions of V_a affine variety.	p. 30
$\mathbb{L}(V_p)$	$\mathbb{L}$-rational functions of V_p projective variety.	p. 30
V	Abelian variety.	p. 31
$\dim(V_a)$, $\dim(V_p)$, $\dim(V)$	Dimension of a variety.	p. 31
W_E	Abelian variety obtained from $E(\mathbb{F}_{q^n})$ via Weil restriction of scalars.	p. 32
Tr	Trace endomorphism of $E(\mathbb{F}_{q^n})$.	p. 33
T_n	Trace-zero subgroup of $E(\mathbb{F}_{q^n})$.	p. 33
$\mathcal{T}_n$	Trace-zero variety.	p. 33
$S_{t,n_1,\cdots,n_t}$	A $(t, n_1, \cdots, n_t)$-generalized summation polynomial, $t, n_i \in \mathbb{Z}_{\geq 2}$.	p. 78
$\mathcal{F}$	A factor base for the index calculus.	p. 40
DRL ordering	The degree-reverse-lexicographic monomial ordering.	p. 124
$D = \text{solv.deg}(I)$	Solving degree of $I = (f_1, \cdots, f_s)$ w.r.t. the DRL ordering.	p. 124
$\mathcal{M}_D$	Macaulay matrix of the polynomial system associated to I.	p. 124

Acknowledgements

Thank you to my supervisor, Elisa Gorla.
Thank you Elisa for the enthusiasm you put into your job, and the energy that spreads around you. For your constructive criticism, that makes me learn so much. For the time you have always dedicated to me, even when you had a million of other things to do and to care about.

To little Marina, for making us all happy.

To Alberto, my historical office-mate.
Grazie di tutto Albertoso. Sei un fratello maggiore per me.

To Alessio and his super wife Elisa, i miei cari amici neuchatellosi barcellonesi genovesi.

To Relinde, for her useful suggestions about this thesis, and for all lunches together.

To Roberta, because she knows my favorite biscuits.

To Anurag, for the fundamental yellow paper where I wrote all the thesis.

Thank you to my family, with all my love. Grazie alla mia famiglia, con tutto il mio cuore.

Alla mia Mamma Silvia.

Al mio Papà Sergione.

Grazie Mamma e Papà.

A mio fratello Francesco, la nostra ancora di salvezza e il nostro piccolo sciamano.

Alle mie Nonne carissime. Alla Nonna Adriana e alla Nonna Rosa.

Al mio Nonno Sandro che guarda dal cielo la sua Julie.
Nonno, questa tesi è dedicata a te.

A Stefano e a Jasmin. To my uncle Sandro.

Thank you to all my dear friends.

Alle ragazze dell'Asse, che ci sono sempre. Anna, Chiara, Elisa, Sara.

A Sara e Nico.

Alla mia amica giramondo Shanti.

Thank you to Matteo.

Grazie Matte. Grazie di cuore.

Introduction

In this thesis we study trace-zero subgroups of elliptic and twisted Edwards curves, from the point of view of their potential application to cryptography. In fact, such groups can be a valid alternative to the standard use of elliptic curves in this area.

Cryptography is the study of techniques that guarantee secret communication between parties. Nowadays, it is of crucial importance for all computer security, from safe electronic commerce to trusted web authentication via a password.
In the basic cryptographic scenario, two people need to communicate, without being understood by malicious third parties. Hence, they encrypt and decrypt their messages with the use of secret keys. Messages and keys are, in important real cryptosystems, elements of a cyclic group G of prime order. This group has to ensure efficient and safe communication at the same time. From the point of view of efficiency, it is necessary to know fast algorithms to perform the arithmetic in G. Moreover, we need to represent group elements with the least possible number of bits, in order to save storage space. From the point of view of security, we need that the Discrete Logarithm Problem (DLP) in the group G is difficult to solve.

Among the groups that satisfy the efficiency and security conditions mentioned above, the groups of points of elliptic curves play an important role. Such groups are nowadays widely used in cryptographic applications. They are the first example of how algebraic geometry can be applied to cryptography.

Nevertheless, it is possible to go beyond this first example, as Frey suggested in [43]. In his paper, Frey proposed to get a deeper insight in the geometric world, in order to find valid alternatives to standard elliptic groups for the cryptographic setting. This was the first time that trace-zero subgroups of elliptic curves were proposed for applications to cryptography.

Let E be an elliptic curve defined over a prime field $\mathbb{F}_q$, the so-called base field of E. We denote by $\oplus$ the standard point addition on E, whose neutral element $\mathcal{O}$ is the point at infinity of the curve. For each field extension $\mathbb{F}_q \subseteq \mathbb{L}$, we denote by $(E(\mathbb{L}), \oplus)$ the abelian group of $\mathbb{L}$-rational points of E, which consists of all affine points of E with coordinates in $\mathbb{L}$ and the point at infinity. Moreover, we denote by φ the Frobenius endomorphism of the curve E, that raises each coordinate of a point of E to the q-th power.
For n an odd prime and the degree n field extension $\mathbb{F}_q \subseteq \mathbb{F}_{q^n}$, we define the trace-zero subgroup $T_n \subseteq E(\mathbb{F}_{q^n})$ as the group of points $P \in E(\mathbb{F}_{q^n})$ such that the sum of the n Frobenius conjugates of P is zero:

$$T_n = \{P \in E(\mathbb{F}_{q^n}) : P \oplus \varphi(P) \oplus \cdots \oplus \varphi^{n-1}(P) = \mathcal{O}\}.$$

It turns out that trace-zero subgroups satisfy the security and efficiency conditions to set up a safe cryptosystem. Regarding security, it can be shown that the security of

degree three trace-zero subgroups against DLP attacks is comparable to the security of groups of base field-rational points of elliptic curves, of the same size. These latter groups achieve the optimal security against such kind of attacks. This means that the best known algorithm to solve the DLP in them is the Pollard's rho method, which works in any group independently of its specific structure. On the other hand, in trace-zero subgroups T_n one can apply the index calculus algorithm for abelian varieties proposed by Gaudry in [46]. The complexity of this algorithm is the same as Pollard's rho complexity for $n = 3$. Therefore, the degree 3 trace-zero subgroup achieves the optimal security against DLP attacks. For $n \geq 5$, Gaudry's strategy lowers in theory the complexity of solving the DLP. However, the method requires solving huge polynomial systems, and this task is often infeasible in practice, even for small n.

Again from the point of view of security, it can be shown that solving a DLP in $E(\mathbb{F}_{q^n})$ is the same as solving a DLP in $E(\mathbb{F}_q)$ and a DLP in T_n. Hence, the increase in the security level from the group $E(\mathbb{F}_q)$ to the group $E(\mathbb{F}_{q^n})$ is the same as the increase from $E(\mathbb{F}_q)$ to the subgroup T_n. This means that, instead of working in the whole group $E(\mathbb{F}_{q^n})$, we can restrict ourselves to the subgroup T_n, without loosing security. This is an advantage if we use an optimal representation for trace-zero elements.

In this case, to use an optimal representation for group elements means to associate to each element of the group the shortest possible tuple of coordinates of $\mathbb{F}_q$. Each coordinate of $\mathbb{F}_q$ can be easily translated in a bit-string via its binary representation. Therefore, points of the group will be treated in a computer as bit-strings of the shortest possible length. As a consequence, they will be stored in the minimal space. It can be shown that the cardinality of $E(\mathbb{F}_{q^n})$ is about q^n, while the cardinality of T_n is about q^{n-1}. Therefore, elements of the whole group $E(\mathbb{F}_{q^n})$ are optimally represented via n coordinates of $\mathbb{F}_q$. On the other hand, only $n-1$ coordinates of $\mathbb{F}_q$ are required to optimally represent elements of the trace-zero subgroup. Hence, using an optimal representation for trace-zero elements, one can enjoy the benefit of optimal data storage for the level of security. In fact, various optimal representations are known for trace-zero subgroups of elliptic curves: see for example [62] for $n = 3$, [75] for $n = 5$, [27] and [69] for $n = 3, 5$, [47],[49].

Optimal representations for trace-zero elements can only be a practical advantage in the cryptographic setting if they are integrated with efficient algorithms to perform the arithmetic of the group. In this way, the benefit of saving storage space goes together with the efficiency of the computation in the group. One possible approach to this issue is to recover trace-zero elements from their optimal representations, then perform the efficient standard arithmetic in $E(\mathbb{F}_{q^n})$ and, in the end, compute the optimal representation of the result. This method requires fast algorithms for compression and decompression: compression is the process of computing the optimal representation of a point, decompression is the inverse procedure. In fact, all previously cited works about optimal representations of trace-zero elements give efficient compression and decompression algorithms. It follows that the combination of optimal data storage for security level and efficient arithmetic is made effective in the cryptographic scenario. Moreover, in the trace-zero subgroup, we can exploit the properties of the Frobenius endomorphism of the elliptic curve in this subgroup, in order to speed up the computation of the standard scalar multiplication of points: see [6, Section 15.3], [4], [55], [62], [75]. We remark that, for important cryptographic applications like the Diffie-Hellman key exchange, scalar multiplication is the primary operation to be taken into account. Using the mentioned technique based on the Frobenius endomorphism, one has that scalar multiplication in T_n is faster than in the whole group $E(\mathbb{F}_{q^n})$. Furthermore, it has been empirically verified that, thanks to the Frobenius strategy, scalar multiplication in T_3 and T_5 is faster than the same operation

in classical groups of base field-rational points of elliptic curves, of the same size (see [5]). We conclude that trace-zero subgroups provide a suitable combination of all security and efficiency aspects that are required for cryptographic applications, and that they can be a real, valid alternative to the standard use of elliptic curves.

Twisted Edwards curves are a quite recent development in cryptography. Edwards curves were first introduced by Edwards in 2007 ([34]). Right afterwards, they were proposed for cryptographic applications by Bernstein and Lange ([13]). They were then generalized to twisted Edwards curves ([12]). It can be shown that each twisted Edwards curve can be turned into an elliptic curve, by performing a rational change of variables. Therefore, we can see twisted Edwards curves as special elliptic curves, written in a new form. It follows that one can define addition of points of a twisted Edwards curve. Moreover, groups of points of twisted Edwards curves can be used in cryptography in the same way as groups of points of elliptic curves. Furthermore, groups of points of twisted Edwards curves have some security and efficiency advantages over the standard groups of points of elliptic curves. In fact, it can be shown that their arithmetic is faster (see [13], [14], [11][12], [19]), and that they are safer against certain types of cryptographic attacks (see [13], [12]).

Trace-zero subgroups of twisted Edwards curves are defined in the same way as trace-zero subgroups of elliptic curves. They have all the good cryptographic properties of trace-zero subgroups of elliptic curves mentioned before.

The thesis is organized as follows. Chapter 1 contains preliminary definitions and results, that are useful to understand the further work. In Chapter 2, we propose two optimal representations for trace-zero subgroups of twisted Edwards curves, with efficient compression and decompression algorithms. In Chapter 3, we introduce generalized summation polynomials, which will be used in the subsequent chapters. In Chapter 4, we give a new algorithm to perform scalar multiplication in the degree three trace-zero subgroup. Finally, Chapter 5 deals with a specialization of the index calculus algorithm proposed by Gaudry in [46] to trace-zero subgroups.

Chapter 1 is divided into five sections. Section 1.1 is a brief introduction to cryptography. It allows to understand how the work presented in the thesis can be of practical interest in this sector. In Section 1.2, we give basic notions about affine and projective curves. This notions will be used throughout the thesis to deal with the objects that we study, and to prove our results. In Section 1.3, we introduce elliptic curves and twisted Edwards curves, underlying their role in cryptography. Section 1.4 contains a detailed presentation of the main objects of the thesis, that are trace-zero subgroups of elliptic and twisted Edwards curves. In this survey, we point out and explain the main aspects that make trace-zero subgroups remarkable from the cryptographic point of view. In Section 1.5, we speak about the discrete logarithm problem in groups, whose hardness is fundamental for the security of cryptosystems. We focus on the strategy of index calculus for the solution of the DLP, and on the version of this algorithm proposed by Gaudry in [46], which can be applied to trace-zero subgroups.

In Chapter 2, we propose two optimal representations for trace-zero subgroups of twisted Edwards curves, giving efficient compression and decompression algorithms for both of them. The motivation of this study is the importance to combine the benefit of optimal data storage with an efficient performance of the arithmetic of the group. Moreover, as we pointed out before, twisted Edwards curves have been recently introduced in

cryptography, and they have some advantages over standard elliptic curves. Furthermore, all the previously cited works, about optimal representations for trace-zero subgroups, take trace-zero subgroups of elliptic curves. Therefore, to the extent of our knowledge, these are the first optimal representations to be proposed for trace-zero subgroups of twisted Edwards curves. Our representations, with annexed algorithms, are non-straightforward adaptations of the representations proposed in [47] and [49] for trace-zero subgroups of elliptic curves.

In Chapter 3, we introduce generalized summation polynomials of elliptic curves, and give a recursive algorithm to construct them. These polynomials allow to find relations between points of the curve. They can be seen as a generalization of summation polynomials of elliptic curves, introduced by Semaev in [67]. Thanks to their geometric property, generalized summation polynomials can have relevant applications to cryptography. In fact, we use them in both Chapter 4 and Chapter 5. Chapter 4 contains a constructive cryptographic application of generalized summation polynomials to cryptography, that provides an efficient algorithm to perform scalar multiplication in the degree three trace-zero subgroup. On the other hand, Chapter 5 describes a destructive cryptographic application of these polynomials, to the study of a new index calculus strategy for DLP attacks in trace-zero subgroups.

In Chapter 4, we give an algorithm to perform scalar multiplication in the degree three trace-zero subgroup of an elliptic curve, using the optimal coordinates proposed in [49] to represent the elements of the group. The motivation of this work is again to integrate the use of an optimal representation for trace-zero elements with efficient performances of the arithmetic in the group. We pointed out that a possible approach to this task is to decompress the trace-zero point that we want to multiply, to perform the standard scalar multiplication of points on elliptic curves, and then to compress the obtained result. On the other hand, the algorithm that we propose in this chapter follows the alternative approach. Namely, it performs scalar multiplication in the degree three trace-zero subgroup, using directly the optimal coordinates of the given representation, without decompression and compression of points. To the extent of our knowledge, it is the first algorithm that performs the scalar product in the trace-zero subgroup with the direct approach in optimal coordinates.
Furthermore, our algorithm adapts the technique that makes use of the Frobenius endomorphism of the elliptic curve, to speed up the standard scalar multiplication of points of the trace-zero subgroup. Therefore, we maintain the advantages of this strategy, even performing the operation directly in the optimal coordinates.

In Chapter 5, we propose a new variant of index calculus for trace-zero subgroups of elliptic curves. Our algorithm is a specialization, to trace-zero subgroups, of the index calculus algorithm proposed by Gaudry in [46]. To the extent of our knowledge, the only other specialization of this algorithm to trace-zero subgroups is given in [48]. As we mentioned before, the bottleneck of Gaudry's strategy is that one has to deal with huge polynomial systems, and the computation of their solutions is often not feasible in practice. Therefore, the goal of our work was to construct new polynomial systems, in alternative to those proposed in [48], that are easier to solve in the case of small n, that is the important case for cryptographic applications. We succeed in our aim, and prove that the complexity of solving our polynomial systems is in most cases lower than the complexity of solving the systems of [48], in the cryptographic relevant cases of $n = 3, 5, 7$.

In the Appendix, we list the explicit formulas and equations that we found, using some of the procedures and algorithms given in the thesis. We used these formulas and equations to make computations, examples and practical experiments.

We performed explicit computations and implemented algorithms with Magma ([22], [23]), version V2.22-1 of the software, running on a single 3 GHz core.

Articles

The new results in this thesis are contained in the following articles.

1. G. Bianco, E. Gorla, *Compression for trace zero points on twisted Edwards curves*, Journal of Mathematical Cryptology 10, no. 1 (2016), 15-34.

2. G. Bianco, E. Gorla, *Scalar multiplication in compressed coordinates in the trace-zero subgroup*, submitted (2017), available at https://arxiv.org/abs/1709.04178.

3. G. Bianco, E. Gorla, *Index calculus in trace-zero subgroups and generalized summation polynomials*, preprint (2017).

The results of Chapter 2 are contained in 1. The results of Chapter 3 and 5 are contained in 3. The results of Chapter 4 are contained in 2.

Contents

Chapter 1

Preliminaries

1.1 Introduction to cryptography

WhatsApp Messenger is one of the most popular smartphone application for text messages and phone calls. It enables the client to use his phone number as well as the internet to send text messages and make voice and video calls for free. It was created in 2009 by J. Koum. Since February 2016, it has more than one billion users. In 2012, the WhatsApp company started introducing encryption in its systems, to ensure secure communication between the users. The encryption process of all WhatsApp structures was officially completed in April 2016.

WhatsApp is a bright example of the massive presence of cryptography in our everyday life. This is the reason why we choose it to start our little excursus on this topic. We now give a general idea of how WhatsApp encryption works. We exploit such a real scenario to collect and recall fundamental definitions, problems and tools of modern cryptography. These notions establish the basis and the motivations of the thesis.

General informations about WhatsApp can be found at https://www.whatsapp.com and https://en.wikipedia.org/wiki/WhatsApp. As regards the technical aspects of WhatsApp encryption, we refer to [74], [57], [58], [21]. Moreover, the interested reader can consult [72] for a basic, ample overview of modern cryptography.

Suppose that the WhatsApp users Alice and Bob want to send messages to each other via WhatsApp. In order to do this, they have to start an encrypted messaging session. During the session, they have a common secret key to encrypt and decrypt their texts, so that a third malicious party cannot have informations about their private conversation. This is an example of symmetric encryption. In fact, Alice and Bob share the same key both to encrypt and decrypt their messages, and nobody but them knows their common secret key. The symmetric encryption protocol used by WhatsApp is AES-256 (see [72, Chapter 3.6]). Symmetric encryption raises the issue of exchanging the secret key among the two parties that want to communicate to each other. To exchange the common secret key at the beginning of the session, Alice and Bob use a key agreement protocol called Extended Triple Diffie-Hellman, or X3DH (see [57]). The protocol is based on the Diffie-Hellman key exchange. Diffie-Hellman key exchange was proposed by Diffie and Hellman in 1976 (see [32]). It is one of the most important tool of public-key (or asymmetric) cryptography. In contrast with the symmetric setting, public-key cryptography does not require the use of common secret keys. We explain below how Diffie-Hellman key exchange works, after giving some useful notation.

Notation 1. Let $N \in \mathbb{Z}_{>0}$. Let $(G, +)$ be a cyclic group of order N and let P be a generator of G. We write $G = \langle P \rangle$. For each integer k and $Q \in G$, we write $kQ =$

$$\underbrace{Q + Q + \cdots + Q}_{k \text{ times}}.$$

Definition 2. [Diffie-Hellman key exchange] Let $N \in \mathbb{Z}_{>0}$, and let $(G, +) = \langle P \rangle$ be a cyclic group of order N. Suppose that the generator P is a public information. Moreover, suppose that Alice has a secret key $1 \leq a \leq N-1$, and that Bob has a secret key $1 \leq b \leq N-1$. Notice that Alice does not know b and Bob does not know a. To share the same secret key $K \in G$, Alice and Bob perform Diffie-Hellman key exchange, that consists of the following steps :

1. Alice computes $P_A = aP$ and sends it to Bob.
2. Bob computes $P_B = bP$ and sends it to Alice.
3. Alice computes $K = (ab)P = aP_B$.
4. Bob computes $K = (ab)P = bP_A$.

Once performed the X3DH version of Diffie-Hellman key exchange, Alice and Bob can start their WhatsApp messaging session. They use the computed common key for encryption and decryption. The cryptosystem adopted by WhatsApp is a hybrid cryptosystem. It combines the public-key powerful strategy of the Diffie-Hellman key exchange with the efficiency of symmetric encryption. An alternative to this strategy is to perform public-key encryption. In this latter case, both Alice and Bob have a pair of keys, as in the Diffie-Hellman key exchange: a private key (only the owner knows it) and a public key. Alice uses Bob's public key to encrypt her messages for Bob, Bob uses his private key to decrypt them, and vice versa. Examples of public-key cryptosystems that are used nowadays are the RSA cryptosystem and the ElGamal cryptosystem. The scheme below sums up the different settings of cryptography we deal with.

Scheme 1.

Cryptography. Ensures secure communication between parties.

- Symmetric setting. The parties need a common secret key.
  - Symmetric encryption. Alice and Bob use a common secret key both to encrypt and decrypt messages.
- Public-key or asymmetric setting. The parties do not need a common secret key.
  - Public key (or asymmetric) encryption. Alice uses Bob's public key to encrypt messages, Bob uses his private key to decrypt them, and vice versa.
  - Diffie-Hellman key exchange.
- Hybrid setting. Combines symmetric and public-key tools.

1.1.1 Security and efficiency issues to set up a good cryptosystem

In the setting of a cryptosystem, two fundamental issues have to be taken into account: the security and the efficiency of all procedures. As we saw in Definition 2, the Diffie-Hellman key exchange performs operations in a cyclic group G. Hence, this group has to be chosen in such a way to satisfy both the security and the efficiency conditions.

To analyze these two aspects, we will deal with the complexity of algorithms in the group G. The complexity of an algorithm in the group G is the number of operations in G that the algorithm has to perform in order to return the output, in terms of the bit-length

of the input. We follow the definitions and notations of [6, Section 1.2], to speak about complexity. We use the big-O notation and the small-o notation of [6, Definition 1.4], as well as the big-Ω notation of [50, Section 2.21]. Moreover, we write

$$L_N(\alpha, c) = \exp\left((c + o(1))(\log N)^{\alpha}(\log\log N)^{1-\alpha}\right), \tag{1.1}$$

with $0 \leq \alpha \leq 1$ and $c > 0$, as in [6, Definition 1.7]. From (1.1), we say that an algorithm in G has polynomial (resp. exponential, resp. subexponential) complexity if it has complexity $L_N(\alpha, c)$ with $\alpha = 0$ (resp. $\alpha = 1$, resp. $\alpha < 1$). Notice that, according to this definition, an algorithm has polynomial (resp. exponential) complexity if it has logarithmic (resp. polynomial) complexity in the group order N. This is equivalent to saying that the algorithm has polynomial (resp. exponential) complexity in the bit-length $\log_2(N)$ of the group elements in input.

Security of a cryptosystem

Let us first focus on the aspect of security. We take as example the hybrid cryptosystem of WhatsApp. Obviously, the secrecy of communication is violated if a third party can somehow recover the key K. Since the communication channel during Diffie-Hellman key exchange is not supposed to be secure, an opposer (say Eve) can access the informations P_A and P_B, as well as the public information P. Hence, if Eve is able to compute K given P, P_A and P_B, the cryptosystem is broken. The problem of computing K given P, P_A and P_B is the so-called Computational Diffie-Hellman problem.

Definition 3 (Computational Diffie-Hellman problem)**.** Let $N \in \mathbb{Z}_{>0}$, $(G, +) = \langle P \rangle$ a cyclic group of order N, $P_A, P_B \in \langle P \rangle$ such that $P_A = aP$, $P_B = bP$, with $0 \leq a, b \leq N-1$. Solving the Computational Diffie-Hellman problem with respect to P, P_A and P_B means finding the element $K = (ab)P$, given P, P_A and P_B.

Notice that, if Eve is able to compute a from P_A and b from P_B, then she is able to solve the Computational Diffie-Hellman problem with respect to P, P_A and P_B.

Definition 4 (Discrete Logarithm problem)**.** Let $N \in \mathbb{Z}_{>0}$, $(G, +) = \langle P \rangle$ a cyclic group of order N, $Q \in G$. Solving the Discrete Logarithm Problem (DLP) with respect to P and Q means finding the unique $0 \leq \ell \leq N - 1$ such that $Q = \ell P$. The integer ℓ is called the discrete logarithm of Q with respect to the base P, and it is denoted by $\ell = \log_P(Q)$.

The solution of the Discrete Logarithm problem with respect to P and P_A, and with respect to P and P_B, implies the solution of the Computational Diffie-Hellman problem with respect to P, P_A and P_B.

The hardness of the Discrete Logarithm problem in a group G is of crucial importance for the security of a cryptosystem based on G. Namely, if the DLP in G is computationally difficult to solve, then G could be a good candidate to set a secure cryptosystem. Notice that this condition is necessary for security, but not sufficient. The computational difficulty of a problem corresponds to the estimated time to solve the problem, with the most efficient known algorithms and the most powerful available computers. We give an overview on the techniques for solving the DLP in Section 1.5. In the mentioned section, we deal with the complexity of such techniques in the different groups that are used in cryptography. We will especially focus on the strategy of index calculus.

Example 5. In the cyclic group $(\mathbb{Z}_N, +)$, one can easily solve the Discrete Logarithm problem with the extended euclidean algorithm. Hence, such group is not suitable for

secure encryption. For example, take $\mathbb{Z}_{10} = \langle 7 \rangle$, and $4 \in \mathbb{Z}_{10}$. One has that $\gcd(10, 7) = 1$. With the extended euclidean algorithm, we compute $1 = (-2)\cdot 10 + 3\cdot 7$, from which $7\cdot 3 = 1 \mod 10$. Then we compute $\ell = \log_7(4) = 2$, multiplying by 3 the relation $\ell \cdot 7 = 4 \mod 10$.

Remark 6. Each cyclic group $G = \langle P \rangle$ of order N is isomorphic to $\mathbb{Z}_N$, via the isomorphism $\Phi : G \longrightarrow \mathbb{Z}_N$, $P \mapsto 1$. So, for $Q = \ell P \in G$, we can solve the DLP with respect to P and Q if and only if one we can compute $\Phi(Q) = \ell$. Therefore, we know that the isomorphism Φ exists, but the explicit computation of it is in general a difficult task.

Efficiency of a cryptosystem

The issue of efficiency for the group G consists of three main aspects. First, we need a fast way to compute the order N of the group. Moreover, high performance algorithms for the arithmetic calculation in the group are required. Finally, it is of great importance to use an optimal, manageable representation for the group elements.

Fast algorithms for the order of the group. One has to know the order N of G for the security of the cryptosystem. In fact, security depends on the size of N as well as on its prime decomposition. If N is too small, it is easy to solve the DLP for $Q = \ell P$ in G, by simply trying all integers $0 \leq k \leq N - 1$ till we find the correct one. This kind of attack is called brute-force attack. Hence, if N is not big enough, the group G is not a safe group for cryptography. Moreover, one can apply the so-called Pohlig-Hellman method, and reduce the computation of a DLP in G to the computations of DLP's in the subgroups of G of prime order. Therefore, if N is the product of small primes, G cannot be used to set a secure cryptosystem. Thanks to the same method, one can always take cyclic groups of prime order for cryptographic applications. So it is necessary to know if N is prime, or if there is a large prime p that divides N. In this latter case, we restrict to the subgroup of G of order p to build the cryptosystem.

Fast arithmetic in the group. We will see in the sequel how to perform efficient arithmetic in some groups that are used in cryptography. More precisely, we will speak about efficient arithmetic in groups of points of elliptic and twisted Edwards curves, and trace-zero subgroups of elliptic and twisted Edwards curves. For practical applications as the Diffie-Hellman key exchange, one is especially interested in fast performance of scalar multiplication. Hence, in the following, we will mainly focus on this operation.

Optimal representations for group elements. The elements of a group G are represented in a computer as tuples of bits. In order to optimize the space storage, these tuples need to be as short as possible. Moreover, one needs efficient and fast compression and decompression algorithms for the group elements. The process of compression allows to pass from a group element to the corresponding bit-string. Decompression is the inverse procedure. From this intuitive explanation, it follows that an optimal representation for the group G can be given via an injective map $\mathcal{R} : G \longrightarrow \mathbb{F}_2^\ell$. The group G and $\mathbb{F}_2^\ell$ has to be about of the same size, that is, ℓ is about $\log_2 |G|$. Moreover, images $\mathcal{R}(g)$ and preimages $\mathcal{R}^{-1}(x)$ (for $g \in G$, $x \in \mathbb{F}_2^\ell$) should be easy to compute. Indeed, for practical applications, the injectivity condition can be relaxed. Namely, we allow the identification of a small amount of elements of G. This means that the number of elements that one can identify is negligible compared to the size of the group. We allow also some exceptions, that is, a small number of bit-strings for which the preimage under $\mathcal{R}$ could be larger.

We give below the rigorous definition of optimal representation that we follow in the sequel, together with some remarks and examples. We refer basically to [49, Definition 2.6, Definition 2.7].

Definition 7 (Optimal representation for a family of groups)**.** Let G be a finite group and $\ell \in \mathbb{Z}_{>0}$. A representation of G of size ℓ is a map

$$\mathcal{R} : G \longrightarrow \mathbb{F}_2^\ell.$$

Let $\mathcal{G}$ be a family of finite groups. An optimal representation for the family $\mathcal{G}$ is a family $\mathcal{R} = (\mathcal{R}_G)_{G\in\mathcal{G}}$ of representations

$$\mathcal{R}_G : G \longrightarrow \mathbb{Z}_2^{\ell_G} \times \mathbb{Z}_2^{k_G}$$

with the following properties. We have that $\ell_G = \lceil \log_2 |G| \rceil$ for all $G \in \mathcal{G}$. Moreover, there exist constants $c, d, e \in \mathbb{Z}_{\geq 0}$ for which the following facts hold. For all $G \in \mathcal{G}$, we have that $k_G \leq e$. Furthermore, there exists a set $\mathcal{S}_G \subseteq \mathbb{Z}_2^{\ell_G} \times \mathbb{Z}_2^{k_G}$ with $|\mathcal{S}_G| \leq c$ and $|\mathcal{R}_G^{-1}(x)| \leq d$ for all $x \in (\mathbb{Z}_2^{\ell_G} \times \mathbb{Z}_2^{k_G}) \setminus \mathcal{S}_G$.

For each $G \in \mathcal{G}$, we say that $\mathcal{R}_G$ is an optimal representation for G, or for the elements of G. In addition, for all $G \in \mathcal{G}$, $g \in G$ and $x \in \mathrm{Im}(\mathcal{R}_G)$, we call compression and decompression the process of computing $\mathcal{R}_G(g)$ and $\mathcal{R}_G^{-1}(x)$ respectively.

We point out that the previous definition does not require the group structure on G. One can define in the same way representations for sets and optimal representations for families of sets. We restricted the definition to groups since groups are the objects we work with. Moreover, notice that if $\mathcal{G} = \{G\}$, an optimal representation for $\mathcal{G}$ is given by each representation of G of size $\ell_G + k$, with $k \in \mathbb{Z}_{\geq 0}$. Hence, the previous definition makes sense if we take families of groups, for which the order grows if we increase some specific parameters. In this case, the constants c, d, e will be independent of these parameters. So we can fix the parameters in such a way that the constants are negligible compared to the size of the chosen group. Finally, observe that, given an optimal representation $\mathcal{R}$ for $\mathcal{G}$, one identifies, up to few exceptions, at most d elements of G, for each $G \in \mathcal{G}$. Hence, it is enough to add $\lceil \log_2 d \rceil$ bits to each representation map, in order to recover injectivity and represent the group elements without any ambiguity. The number of bits that we add is negligible compared to the size of the representation.

Example 8. We take the family of groups $(\mathbb{Z}_n)_{n\geq 2}$. We take its family of representations $\mathcal{R} = (\mathcal{R}_n)_{n\geq 2}$, defined as follows. For each $n \geq 2$, and for each class $\overline{x} \in \mathbb{Z}_n$, $\mathcal{R}_n(\overline{x}) \in \mathbb{Z}_2^{\lceil \log_2 n \rceil}$ is the binary representation of x. For example, for $n = 8$, we have

$$\mathcal{R}_8 : \mathbb{Z}_8 \longrightarrow \mathbb{Z}_2^3,$$

with $\mathcal{R}_8(\overline{0}) = (0,0,0)$, $\mathcal{R}_8(\overline{1}) = (0,0,1)$, $\mathcal{R}_8(\overline{2}) = (0,1,0)$, $\mathcal{R}_8(\overline{3}) = (0,1,1)$, $\mathcal{R}_8(\overline{4}) = (1,0,0)$, $\mathcal{R}_8(\overline{5}) = (1,0,1)$, $\mathcal{R}_8(\overline{6}) = (1,1,0)$, $\mathcal{R}_8(\overline{7}) = (1,1,1)$. Then $\mathcal{R}$ is an optimal representation for the family $(\mathbb{Z}_n)_{n\geq 2}$, with $c = e = 0$ and $d = 1$. Notice that all representations of the family are injective.

Example 9. Take the family of finite fields $(\mathbb{F}_q)_{q=p^m,\ \mathrm{p\ prime},\ m\in\mathbb{Z}_{\geq 1}}$. For this family, we can take the following optimal representation $\mathcal{R} = (\mathcal{R}_q)_{q=p^m}$. If $q = p$ is a prime, we take the representation $\mathcal{R}_p$ as in Example 8. If $q = p^m$,$m \in \mathbb{Z}_{>1}$, we choose a basis $\{\alpha_1, \cdots, \alpha_m\}$ of $\mathbb{F}_q$ over $\mathbb{F}_p$. So we can write each $x \in \mathbb{F}_q$ as $x = x_1\alpha_1 + \cdots + x_m\alpha_m$ for some $x_i \in \mathbb{Z}_p$. We take the representation $\mathcal{R}_q$ such that $\mathcal{R}_q(x) = (\mathcal{R}_p(x_1), \cdots, \mathcal{R}_p(x_m))$. Then $\mathcal{R}$ is an optimal representation for the family $(\mathbb{F}_q)_{q=p^m}$.

Take $\mathcal{G}$ a family of groups with $|G| \in O(q^m)$ for $G \in \mathcal{G}$. It follows from the previous example that an optimal representation for the family $\mathcal{G}$ can be given via a family of maps of the form

$$\mathcal{R}_G : G \longrightarrow \mathbb{F}_q^m \times \mathbb{Z}_2^{k_G}, \tag{1.2}$$

such that there exist constants c, d, e as in Definition 7. In fact, let $\mathcal{R}_q$ be as in Example 9, and let $\mathrm{id}_{\mathbb{Z}_2^{k_G}}$ be the identity map of $\mathbb{Z}_2^{k_G}$. Hence, the family $((\mathcal{R}_q^m \times \mathrm{id}_{\mathbb{Z}_2^{k_G}}) \circ \mathcal{R}_G)_{G \in \mathcal{G}}$ is an optimal representation for $\mathcal{G}$ as in Definition 7. In the sequel of the thesis, we give optimal representations in the form (1.2).

Remark 10. It is common in cryptographic applications to use optimal representations that are not defined in the neutral element of the group (see [49, Remark 2.8]). We introduce some of them in the sequel. In fact, the neutral element of G is a special element, that in practice is never used as a cryptographic key. Hence one can disregard it in the representation of the elements of the group.

Suitable groups for cryptography

Among the groups that satisfy the necessary conditions for secure and efficient encryption, we record the following families:

- The cyclic multiplicative groups $\mathbb{Z}_p \setminus \{0\}$, where p is a prime number.
- The cyclic multiplicative groups $\mathbb{F}_q \setminus \{0\}$, where q is a prime power and $\mathbb{F}_q$ is the finite field with q elements.
- The groups of points of elliptic curves, or of twisted Edwards curves.
- The trace-zero subgroups of elliptic curves, or of twisted Edwards curves.

WhatsApp uses the group of base field-rational points of the elliptic curve Curve25519 in its X3DH key exchange protocol. Such curve has been proposed for Diffie-Hellman key exchange schemes in 2005, by D. Bernstein (see [10]). It is considered to be a safe curve, in which point addition and scalar multiplication are particularly fast. We give a survey on elliptic curves and twisted Edwards curves in Section 1.3. We recall the basic notions about affine and projective curves in Section 1.2. Trace-zero subgroups of elliptic and twisted Edwards curves are the main object of this thesis. We introduce them in Section 1.4.

The scheme below sums up the necessary requirements to set up a good cryptosystem, that we discussed during this subsection.

Scheme 2.

Necessary requirements for groups in cryptography.

Security.

- DLP (Section 1.5). The DLP in the group has to be hard to solve.

Efficiency.

- Fast algorithms to compute the order of the group.
- Efficient and fast addition and scalar multiplication.
- Optimal representations for group elements, with fast compression and decompression algorithms.

Examples of suitable groups.

- Groups of points of elliptic and of twisted Edwards curves (Section 1.3).
- Trace-zero subgroups of elliptic and twisted Edwards curves (Section 1.4).

1.2 Affine and projective curves

In this section, we review basic notions of algebraic geometry, about affine and projective curves. These concepts will be used throughout the thesis, as we deal with elliptic curves and twisted Edwards curves. We integrate the theoretic results with examples that use such curves. The exposition refers mostly to [6, Chapter 4] , [44], [73] and [7].

Curves in the affine plane. Let $\mathbb{K}$ be a field, and denote with $\overline{\mathbb{K}}$ its algebraic closure. The affine plane over $\overline{\mathbb{K}}$ is

$$\mathbb{A}^2(\overline{\mathbb{K}}) = \overline{\mathbb{K}}^2.$$

We write $\mathbb{A}^2$ for $\mathbb{A}^2(\overline{\mathbb{K}})$ if there is no ambiguity about the field.

An affine plane curve defined over $\mathbb{K}$ is a subset

$$C_a = \{(x, y) \in \mathbb{A}^2 : f(x, y) = 0\} \subseteq \mathbb{A}^2,$$

where $f(x, y)$ is a polynomial of $\mathbb{K}[x, y]$. Throughout our work, we always deal with affine curves that are plane. Hence, we will write affine curve instead of affine plane curve. If the degree of f is $\deg(f) = 1$, then the affine curve is called affine line. We use the notation

$$C_a : f(x, y) = 0$$

to denote the affine curve associated to the polynomial $f(x, y)$. Moreover, for any field extension $\mathbb{K} \subseteq \mathbb{L} \subseteq \overline{\mathbb{K}}$, we define the set of $\mathbb{L}$-rational points of the curve C_a as follows:

$$C_a(\mathbb{L}) = \{(x, y) \in \mathbb{L}^2 : f(x, y) = 0\}.$$

Hence $C_a(\mathbb{L})$ is the set of all points of C_a with coordinates in $\mathbb{L}$.

Example 11. Let $\mathbb{K}$ be a field of characteristic different from 2. We take Ed the affine curve defined over $\mathbb{K}$ by the polynomial $-1 + 3x^2 + y^2 - x^2y^2 \in \mathbb{K}[x, y]$, and we write

$$Ed : -1 + 3x^2 + y^2 - x^2y^2 = 0.$$

Curves in the projective plane. Let $\sim$ be the equivalence relation over $\overline{\mathbb{K}}^3 \setminus \{(0, 0, 0)\}$ defined as follows:

$$x \sim y \text{ iff } \text{ there exists } \lambda \in \overline{\mathbb{K}} \setminus \{0\} \text{ such that } y = \lambda x.$$

The projective plane over $\overline{\mathbb{K}}$ is the quotient

$$\mathbb{P}^2(\overline{\mathbb{K}}) = \overline{\mathbb{K}}^3 \setminus \{(0, 0, 0)\}/ \sim .$$

We write $\mathbb{P}^2$ for $\mathbb{P}^2(\overline{\mathbb{K}})$, when there is no ambiguity about the field. We denote an element of $\mathbb{P}^2$ with $[x, y, z] \in \mathbb{P}^2$. The points $[x, y, 0] \in \mathbb{P}^2$ are called points at infinity.

A projective plane curve defined over $\mathbb{K}$ is a subset

$$C = \{[x, y, z] \in \mathbb{P}^2 : F(x, y, z) = 0\} \subseteq \mathbb{P}^2,$$

where $F(x, y, z)$ is a homogeneous polynomial of $\mathbb{K}[x, y, z]$. Notice that C is well- defined since F is homogeneous. In the sequel, we will always deal with projective curves that are plane. Hence, we will write projective curve, instead of projective plane curve. If the

degree of F is $\deg(F) = 1$, then the projective curve is called projective line. As in the affine case, we use the notation

$$C : F(x, y, z) = 0.$$

The points at infinity $[x, y, 0] \in C$ are called points at infinity of C. Moreover, for any field extension $\mathbb{K} \subseteq \mathbb{L} \subseteq \overline{\mathbb{K}}$, the set

$$C(\mathbb{L}) = \{[x, y, z] \in C : \exists\, t \in \{x, y, z\} \text{ such that } t \neq 0 \text{ and } x/t, y/t, z/t \in \mathbb{L}\}$$

is the set of $\mathbb{L}$-rational points of C.

Relation between the affine and the projective case. The affine and the projective plane over $\overline{\mathbb{K}}$ are strictly related. In fact, $\mathbb{P}^2$ can be seen as $\mathbb{A}^2$ together with the projective line $z = 0$ of points at infinity. More precisely, there is a natural one-to-one correspondence:

$$\Phi_z : U_z = \{[x, y, z] \in \mathbb{P}^2 : z \neq 0\} \longrightarrow \mathbb{A}^2,\ [x, y, z] \mapsto (x/z, y/z).$$

The inverse of Φ_z is defined by $\Phi_z^{-1}(x, y) = [x, y, 1]$ for each $(x, y) \in \mathbb{A}^2$. Similarly, we define U_x, U_y and the corresponding maps Φ_x, Φ_y.

Using the bijection Φ_z, we relate affine and projective curves. We call the projective closure of C_a, and denote it with $\overline{C_a}$, the projective curve

$$\overline{C_a} = \Phi_z^{-1}(C_a) \cup \{[x, y, 0] \in \mathbb{P}^2 : f^h(x, y, z) = 0\} = \{[x, y, z] \in \mathbb{P}^2 : f^h(x, y, z) = 0\},$$

where $f^h(x, y, z)$ is the homogenization of $f(x, y)$ with respect to the last variable z.

Vice versa, we associate to the projective curve C its affine dehomogenization with respect to the variable z:

$$C_z^* = \Phi_z(C \setminus \{[x, y, z] \in C : z = 0\}) = \{(x, y) \in \mathbb{A}^2 : F(x, y, 1) = 0\}.$$

Similarly, we define the affine dehomogenization of C with respect to the variable x or with respect to the variable y. We denote them with C_x^* and C_y^* respectively. Notice that $(\overline{C_a})_z^* = C_a$ and $\overline{(C_z^*)} = C$. Moreover, for any field extension $\mathbb{K} \subseteq \mathbb{L} \subseteq \overline{\mathbb{K}}$, we have that

$$C(\mathbb{L}) = \bigcup_{t \in \{x,y,z\}} \Phi_t^{-1}(C_t^*(\mathbb{L})).$$

This means that the set of $\mathbb{L}$-rational points of C can be identified with the union of the sets of $\mathbb{L}$-rational points of its affine dehomogenizations.

Example 12. Let Ed be the affine curve of Example 11. The projective closure of Ed is $\overline{Ed} : f^h(x, y, z) = -z^4 + 3x^2z^2 + y^2z^2 - x^2y^2$. The points at infinity of $\overline{Ed}$ are $\Omega_1 = [1, 0, 0]$ and $\Omega_2 = [0, 1, 0]$. In general, a projective curve of the form $ax^2z^2 + y^2z^2 = z^4 + dx^2y^2$, with $a, d \in \mathbb{K} \setminus \{0\}$ and $a \neq d$, is called twisted Edwards curve. Hence $\overline{Ed}$ is a twisted Edwards curve with parameters $a = 3$, $d = 1$. We will speak about twisted Edwards curves in the next section.

Absolutely irreducible curves. An affine curve $C_a : f(x, y) = 0$ is said to be absolutely irreducible if $f(x, y)$ is irreducible over $\overline{\mathbb{K}}[x, y]$. Similarly, a projective curve $C : F(x, y, z) = 0$ is said to be absolutely irreducible if $F(x, y, z)$ is irreducible over

$\overline{\mathbb{K}}[x, y, z]$. Notice that the affine curve C_a is absolutely irreducible if and only if its projective closure $\overline{C_a}$ is absolutely irreducible. Similarly, the projective curve C is absolutely irreducible if and only if its affine dehomogenization C_z^* is absolutely irreducible.

Any affine curve $C_a : f(x, y) = 0$ defined over $\mathbb{K}$ is the union of absolutely irreducible affine curves defined over $\overline{\mathbb{K}}$. More precisely, let $f = \prod_i f_i$ be the factorization of $f(x, y)$ into irreducible factors of $\overline{\mathbb{K}}[x, y]$. Then $C_a = \bigcup_i C_{a,i}$, where $C_{a,i} : f_i(x, y) = 0$ is the absolutely irreducible curve associated to the polynomial $f_i \in \overline{\mathbb{K}}[x, y]$, for all i. Notice that, in general, $C_{a,i}$ is not defined over $\mathbb{K}$. The analogous result holds for projective curves.

Absolutely irreducible curves whose defining polynomial is of degree 2 (resp. of degree 3, 4) are called conics (resp. cubics, quartics).

Example 13. Each affine or projective line defined over $\mathbb{K}$ is absolutely irreducible.

The affine curve $xy = 0$ defined over $\mathbb{K}$ is the union of the two affine lines $x = 0$ and $y = 0$. Notice that these two lines are still defined over $\mathbb{K}$.

Let $\mathbb{K} = \mathbb{R}$. The projective curve $x^2 + z^2 = 0$ is not absolutely irreducible, since its defining polynomial $x^2 + z^2$ splits into the two irreducible factors $x + iz$, $x - iz \in \mathbb{C}[x, y, z]$. Hence, the curve $x^2 + z^2 = 0$ is the union of the two lines $x + iz = 0$ and $x - iz = 0$. These lines are not defined over $\mathbb{R}$, but they are defined over $\mathbb{C}$.

One can show that the affine curve Ed of Example 11, as well as its projective closure $\overline{Ed}$ of Example 12, are absolutely irreducible. In general, any twisted Edwards curve, as defined in Example 12, is absolutely irreducible.

1.2.1 Rational functions and rational maps

From now on, we take absolutely irreducible affine and projective curves. We recall the notions of rational functions and rational maps of affine and projective absolutely irreducible curves. We explain the connection between the affine objects and the corresponding notions in the projective plane.

Regular and rational functions of an absolutely irreducible affine curve. Take an absolutely irreducible affine curve $C_a : f(x, y) = 0$ defined over $\mathbb{K}$, and a field extension $\mathbb{K} \subseteq \mathbb{L} \subseteq \overline{\mathbb{K}}$. The quotient ring

$$\mathbb{L}[C_a] = \mathbb{L}[x, y]/(f)$$

is called the $\mathbb{L}$-coordinate ring of C_a. An element $\overline{h}(x, y) \in \mathbb{L}[C_a]$ is called $\mathbb{L}$-regular function of the curve C_a and it can be identified with its evaluation function on points of C_a:

$$\overline{h} : C_a \longrightarrow \overline{\mathbb{K}},\ P \mapsto h(P).$$

For $\overline{h} \in \mathbb{L}[C_a]$, we denote $U_h = \{P \in C_a : h(P) \neq 0\}$. Since f is absolutely irreducible, the ring $\mathbb{L}[C_a]$ is a domain. So one can take the field of fractions of $\mathbb{L}[C_a]$, that is denoted by $\mathbb{L}(C_a)$. The elements of $\mathbb{L}(C_a)$ are called $\mathbb{L}$-rational functions of C_a. We can identify rational functions with equivalence classes of evaluation functions of the form

$$\varphi : U_h \longrightarrow \overline{\mathbb{K}},\ P \mapsto (g/h)(P),$$

where $\overline{g}$, $\overline{h} \in \mathbb{L}[C_a]$, $\overline{h} \neq \overline{0}$ in $\mathbb{L}[C_a]$, and $\varphi_1 = g_1/h_1$, $\varphi_2 = g_2/h_2$ are equivalent if and only if $\varphi_1 = \varphi_2$ on $U_{h_1} \cap U_{h_2}$. We define the domain of $r \in \mathbb{L}(C_a)$ as

$$\mathrm{Dom}(r) = \{P \in C_a : \text{there exists } U_{h_i} \ni P, \varphi_i \in r\}.$$

For each $P \in \mathrm{Dom}(r)$, one can take the evaluation of r at P, namely $r(P) = \varphi_i(P) = (g_i/h_i)(P)$. So we can identify each rational function r with its evaluation map:

$$r : \mathrm{Dom}(r) \longrightarrow \overline{\mathbb{K}},\ P \mapsto r(P).$$

Given $\varphi = g/h$ a representative of the rational function r, we will write $r = g/h \in \mathbb{L}(C_a)$. When we use this notation, we keep in mind that g/h is only one representative of r. Hence U_h can be smaller than $\mathrm{Dom}(r)$.

Example 14. Let Ed be the affine curve of Example 11. Let $r = g_1/h_1 = x^2/(1-y^2) \in \mathbb{K}(Ed)$, with $U_{h_1} = Ed \setminus \{(0, \pm 1)\}$. We have that $r = x^2/(1-y^2) = g_2/h_2 = 1/(3-y^2)$, and $U_{h_2} = Ed$. Hence $\mathrm{Dom}(r) = Ed$.

Rational functions of an absolutely irreducible projective curve. Let C be an absolutely irreducible projective curve $C : F(x, y, z) = 0$ defined over $\mathbb{K}$. As for the affine case, one can take the quotient $\mathbb{L}[x, y, z]/(F)$ and its field of fractions $Q_{\mathbb{L}}$. Nevertheless, in the projective case, we restrict our attention to a proper subfield of $Q_{\mathbb{L}}$. Namely, we define the field of $\mathbb{L}$-rational functions of C as

$$\mathbb{L}(C) = \{r = F/G \in Q_{\mathbb{L}} : F, G \in \mathbb{L}[x, y, z] \text{ homogeneous of the same degree } \}.$$

Notice that we write $r = F/G \in Q_{\mathbb{L}}$ to say that F/G is a representative of r, as in the affine case. We have that $\mathbb{L}(C)$ is isomorphic to the field of affine $\mathbb{L}$-rational functions $\mathbb{L}(C_z^*)$, via the map $\mathbb{L}(C) \ni r(x, y, z) \mapsto r(x, y, 1) \in \mathbb{L}(C_z^*)$.

Example 15. Let Ed be the affine curve of Example 11. The affine $\mathbb{K}$-rational function $x^2/(1-y^2) = 1/(3-y^2) \in \mathbb{K}(Ed)$ corresponds to the projective $\mathbb{K}$-rational function $x^2/(z^2-y^2) = z^2/(3z^2-y^2) \in \mathbb{K}(\overline{Ed})$, via the isomorphism mentioned above.

We obtain similar isomorphisms if we take the dehomogenization of C with respect to x and with respect to y.

Rational maps. Let C_1 and C_2 be two affine absolutely irreducible curves defined over $\mathbb{K}$. We define an affine $\mathbb{L}$-rational map r from C_1 to C_2, and we denote it by

$$r : C_1 \dashrightarrow C_2,$$

a pair of $\mathbb{L}$-rational functions of C_1, $r_1, r_2 \in \mathbb{L}(C_1)$, such that, for $P \in \mathrm{Dom}(r_1) \cap \mathrm{Dom}(r_2)$, one has that $(r_1(P), r_2(P)) \in C_2$. The domain of the rational map r is $\mathrm{Dom}(r) = \mathrm{Dom}(r_1) \cap \mathrm{Dom}(r_2)$, and r is identified with its evaluation function

$$r : \mathrm{Dom}(r) \longrightarrow C_2,\ P \mapsto (r_1(P), r_2(P)).$$

A $\mathbb{L}$-rational map is called $\mathbb{L}$-regular if the two defining $\mathbb{L}$-rational functions are $\mathbb{L}$-regular functions.

Example 16. Let $\mathbb{K}$ be a field of characteristic different from $2, 3$. Take the following absolutely irreducible affine curves defined over $\mathbb{K}$: $Ed : -1 + 3x^2 + y^2 - x^2y^2 = 0$, $Em : 2y^2 - x^3 - 4x^2 - x = 0$, $Ew : y^2 - x^3 + (13/12)x - (23/54) = 0$. The map

$$r_1 : Em \dashrightarrow Ed,$$

$$r_1(x, y) = (x/y, (x-1)/(x+1)) = ((2y(x+1)/(2y^2+x^2+4x+1), (2y^2-x^2-4x-1)/(2y^2+x^2+4x+1)),$$

is a $\mathbb{K}$-rational map from Em to Ed. It can be shown that its domain is

$$\mathrm{Dom}(r_1) = Em \setminus \{(-1, \pm 1), (x, 0) \in Em \text{ with } x \neq 0\}.$$

The map

$$r_2 : Em \longrightarrow Ew,\ r_2(x, y) = ((1/2)(x + \frac{4}{3}), (1/2)y)$$

is a $\mathbb{K}$-regular map from Em to Ew.

A projective $\mathbb{L}$-rational map $r : C_1 \dashrightarrow C_2$ between two absolutely irreducible projective curves C_1 and C_2 is an equivalence class of maps of the form

$$\varphi : U_\varphi \ni [x, y, z] \mapsto [F_1(x, y, z), F_2(x, y, z), F_3(x, y, z)] \in C_2,$$

where $U_\varphi \subseteq C_1$, $C_1 \setminus U_\varphi$ consists of a finite number of points, $F_1, F_2, F_3 \in \mathbb{L}[x, y, z]$ are homogeneous of the same degree, and $\varphi = [F_1, F_2, F_3]$, $\varphi' = [G_1, G_2, G_3]$ are equivalent if $F_i G_j = F_j G_i$ modulo the equation of the curve C_2, for all $i, j \in \{1, 2, 3\}$. The domain of r is the union of all U_φ. If the domain is the whole curve C_1, then r is called $\mathbb{L}$-regular map. As we did with rational functions, one can establish a correspondence between affine and projective rational maps, via (de)homogenization of the involved polynomials.

Example 17. Let $\overline{Ed}$, $\overline{Em}$, $\overline{Ew}$ be the projective closures of the affine curves of Example 16. The map

$$\overline{r_1} : \overline{Em} \dashrightarrow \overline{Ed},$$

$$r([x, y, z] = [x(x+z), y(x-z), y(x+z)] = [2y(x+z), 2y^2 - x^2 - 4xz - z^2, 2y^2 + x^2 + 4xz + z^2],$$

is the projective $\mathbb{K}$-rational map from $\overline{Em}$ to $\overline{Ed}$ which corresponds to the affine $\mathbb{K}$-rational map r_1. Notice that $\overline{r_1}([-1, -1, 1]) = \overline{r_1}([-1, 1, 1]) = [0, 1, 0] = \Omega_2$. Moreover, for $[x, 0, 1] \in \overline{Em}$, with $x \neq 0$, we have $\overline{r_1}([x, 0, 1]) = [1, 0, 0] = \Omega_1$. Hence $\mathrm{Dom}(\overline{r_1}) = Em$ and the map $\overline{r_1}$ is $\mathbb{K}$-regular. The map

$$\overline{r_1}' : \overline{Ed} \dashrightarrow \overline{Em},$$

$$\overline{r_1}'([x, y, z] = [x(z + y), z(z + y), x(z - y)] = [z(z + y)(z - y), x(3z^2 - y^2), z(z - y)^2]$$

is a projective $\mathbb{K}$-rational map from $\overline{Ed}$ to $\overline{Em}$. The domain of $\overline{r_1}'$ is $\mathrm{Dom}(\overline{r_1}') = \overline{Ed} \setminus \{\Omega_1, \Omega_2\}$. Notice that the maps $\overline{r_1}$ and $\overline{r_1}'$ are inverse to each other. Finally, take the map

$$\overline{r_2} : \overline{Em} \longrightarrow \overline{Ew},\ \overline{r_2}([x, y, z]) = [3x + 4z, 3y, 6z].$$

This map is the projective $\mathbb{K}$-regular map which corresponds to r_2.

Projective rational functions and maps are defined in such a way to correspond to the affine concepts on dehomogenizations of the given projective curve. In fact, rational functions and maps are used to study the local behavior of a curve. Given a point P of a projective curve C, this point belongs to $C \cap U_t$ for some $t \in \{x, y, z\}$. Hence one can restrict to the affine dehomogenization C_t^* to study the local behavior of the curve at the point P.

An affine (resp. projective) $\mathbb{L}$-rational map $r : C_1 \dashrightarrow C_2$ is called $\mathbb{L}$-birational if there is an affine (resp. projective) $\mathbb{L}$-rational map $r' : C_2 \dashrightarrow C_1$ such that $r \circ r' = \mathrm{id}_{C_2}$ and $r' \circ r = \mathrm{id}_{C_1}$. If both maps r and r' are $\mathbb{L}$-regular, r is said to be an isomorphism over $\mathbb{L}$. Two affine (resp. projective) curves defined over $\mathbb{K}$ are called birationally equivalent (resp. isomorphic) over $\mathbb{L}$ if there is an affine (resp. projective) $\mathbb{L}$-birational map (resp. an isomorphism over $\mathbb{L}$) between them.

Example 18. The two curves $\overline{Em}$ and $\overline{Ed}$ of Example 17 are birationally equivalent over $\mathbb{K}$, via the $\mathbb{K}$-birational map $\overline{r_1}$. The birational inverse of $\overline{r_1}$ is $\overline{r_1}'$.

The two curves $\overline{Em}$, $\overline{Ew}$ of Example 17 are isomorphic over $\mathbb{K}$, via the isomorphism $\overline{r_2}$. The inverse of $\overline{r_2}$ is given by $\overline{r_2}^{-1}([x,y,z]) = [6x-4z, 6y, 3z]$, for each $[x,y,z] \in E_w$.

Notice that an affine or projective $\mathbb{L}$-rational map $r : C_1 \dashrightarrow C_2$ defines the field homomorphism

$$r^* : \mathbb{L}(C_2) \longrightarrow \mathbb{L}(C_1),\ \alpha \mapsto \alpha \circ r.$$

One can show ([44, Proposition 12, Chapter 6]) that C_1 and C_2 are birationally equivalent over $\mathbb{L}$ if and only if the homomorphism r^* is a field isomorphism.

1.2.2 Local rings and singular points

We focus on local properties of a curve. These properties are related to single points of the curve. For a given point of a projective curve, one can restrict to the dehomogenization of the curve that contains the point. Hence, we now take an absolutely irreducible affine curve $C : f(x,y) = 0$ defined over $\mathbb{K}$. Let P be a point of C. The local ring

$$\mathcal{O}_P(C) = \{r \in \overline{\mathbb{K}}(C) :\ P \in \mathrm{Dom}(r)\}$$

is called the local ring of C at P. We denote by $\mathcal{M}_P(C)$ the maximal ideal of $\mathcal{O}_P(C)$. One has that

$$\mathcal{M}_P(C) = \{r \in \mathcal{O}_P(C) :\ r(P) = 0\}.$$

Let $r \in \overline{\mathbb{K}}(C)$, $P \in C$. We say that r has a pole in P if $r \notin \mathcal{O}_P(C)$. This is equivalent to saying that $P \notin \mathrm{Dom}(r)$. We say that r has a zero in P if $r \in \mathcal{M}_P(C)$. This is equivalent to saying that $P \in \mathrm{Dom}(r)$ and $r(P) = 0$.

Example 19. Let Ed be the affine curve of Example 11. The rational function $r(x,y) = (1-y)/(1+y)$ has a zero in $(0,1)$ and a pole in $(0,-1)$.

Example 20. Let $C : f(x,y) = 0$ be an absolutely irreducible affine curve defined over $\mathbb{K}$. Let $C_1 : f_1(x,y) = 0$ be an affine curve defined over $\overline{\mathbb{K}}$ (C_1 is not necessarily absolutely irreducible). We can view the polynomial f_1 associated to C_1 as a $\overline{\mathbb{K}}$-rational function of C: $f_1(x,y) \in \overline{\mathbb{K}}(C)$. Then $f_1 \in \overline{\mathbb{K}}(C)$ has no poles, while its zeroes are the points of intersection between C and C_1.

Singular and nonsingular points. We say that the point $P \in C$ is a nonsingular point of C if $\mathcal{O}_P(C)$ is a discrete valuation ring. The ring $\mathcal{O}_P(C)$ is a discrete valuation ring if and only if its maximal ideal $\mathcal{M}_P(C)$ is a principal ideal. If this is the case, we have the order function

$$\mathrm{ord}_P : \overline{\mathbb{K}}(C) \longrightarrow \mathbb{Z} \cup \{\infty\}$$

of the discrete valuation ring $\mathcal{O}_P(C)$. The order function ord_P is defined as follows. Let t be a generator of the principal ideal $\mathcal{M}_P(C)$, and $r \in \overline{\mathbb{K}}(C)$. If $r = 0$, then $\mathrm{ord}_P(r) = \infty$. If $r \in \mathcal{O}_P(C)$, then $\mathrm{ord}_P(r) = m$ is such that $r = t^m u$ for some $u \in \mathcal{O}_P(C) \setminus \mathcal{M}_P(C)$. If $r \notin \mathcal{O}_P(C)$, then $\mathrm{ord}_P(r) = -\mathrm{ord}_P(1/r)$.

If $\mathcal{O}_P(C)$ is not a discrete valuation ring, P is called a singular point of C. If all points of C are nonsingular, then C is called a nonsingular curve.

Let us analyze the case in which $P = (0,0)$. One always reduces to this case by performing a linear change of coordinates. For $P = (0,0)$, we have that $\mathcal{M}_P(C) = (x,y)$. Moreover, the polynomial $f(x,y)$ is of the form

$$f(x,y) = f_i(x,y) + f_{i+1}(x,y) + \cdots f_d(x,y),$$

where f_j is a homogeneous polynomial of degree j for $0 < i \leq j \leq d$, $f_i \neq 0$ and d is the degree of f. We say that P is a point of multiplicity i.
We have that P is nonsingular, that is $\mathcal{M}_P(C)$ is a discrete valuation ring, if and only if P has multiplicity $i = 1$. Let us prove that, if $i = 1$, then $\mathcal{M}_P(C)$ is a discrete valuation ring. If $i = 1$, then $f_1(x,y) = ax + by$, with $a \neq 0$ or $b \neq 0$. Suppose $a \neq 0$ (the case $b \neq 0$ is analogous). We can write $f(x,y) = x(a + \hat{g}(x,y)) + y(b + \hat{h}(y))$ for some $\hat{g}, \hat{h} \in \mathbb{K}[x,y]$. Let $H(y) = b + \hat{h}(y)$, $G(x,y) = a + \hat{g}(x,y)$. Since $a \neq 0$, we have that $G(P) \neq 0$ and the rational function $H(y)/G(x,y)$ belongs to $\mathcal{O}_P(C)$. Then we obtain the equality $x = -y(H(y)/G(x,y))$ in $\mathcal{O}_P(C)$, which implies $\mathcal{M}_P(C) = (y)$. So $\mathcal{O}_P(C)$ is a discrete valuation ring. Moreover, we have that all linear polynomials in $\overline{\mathbb{K}}[x,y]$ (that is, all polynomials that defines lines over $\overline{\mathbb{K}}$), except the polynomials defining the line $f_1(x,y) = 0$, are generators for $\mathcal{M}_P(C)$. The line $f_1(x,y) = 0$ is called the tangent line to C at the point P.
When P is singular, its multiplicity is $i > 1$. Then a singular point can be seen as a point in which the tangent to the curve is not uniquely determined. In fact, all lines defined by the linear polynomials of the factorization of f_i in $\overline{\mathbb{K}}[x,y]$ are tangents to C at this point.

Example 21. Let Ed be the affine curve of Example 11. Take $P = (0,1) \in Ed$. We perform the linear change of coordinates that maps the point P to the origin, namely $L(x,y) = (X,Y) = (x, y-1)$. We obtain the curve $f_1(X,Y) + \cdots + f_4(X,Y) = -2Y + (2X^2 + Y^2) + 2X^2Y - X^2Y^2 = 0$. Then the multiplicity of P is 1 and P is a nonsingular point of C.
Moreover, the tangent to Ed at P is $Y = 0$, that is $t : y - 1 = 0$ in the original system of coordinates. In the same way, we conclude that the point $(0,-1) \in Ed$ is nonsingular, and that the tangent to Ed at $(0,-1)$ is $y + 1 = 0$.
Since both $(0,1)$ and $(0,-1)$ are nonsingular, the two order functions $\mathrm{ord}_{(0,1)}$ and $\mathrm{ord}_{(0,-1)}$ are defined. Take $r(x,y) = (1-y)/(1+y) \in \mathbb{K}(Ed)$ the rational function of Example 19. We have that $\mathrm{ord}_{(0,1)}(r) = 2$ and $\mathrm{ord}_{(0,-1)}(r) = -2$.

Let us now study $\overline{Ed}$ at the point $\Omega_1 = [1,0,0] \in U_x$. In order to do this, we study the singularity of the point $(0,0)$ in the dehomogenization $(\overline{Ed})^*_x : F(1,y,z) = (3z^2 - y^2) + y^2z^2 - z^4 = 0$. We have that Ω_1 is a singular point of $\overline{Ed}$ of multiplicity 2, and the two tangents to $\overline{Ed}$ at Ω_1 are $\sqrt{3}z - y = 0$, $\sqrt{3}z + y = 0$ (with $\sqrt{3} \in \overline{\mathbb{K}}$). With the same procedure we obtain that Ω_2 is a singular point of multiplicity 2 of $\overline{Ed}$, and the two tangents to $\overline{Ed}$ at Ω_2 are $x - z = 0$ and $x + z = 0$.

Example 22 (Intersection multiplicity)**.** Let P be a nonsingular point of C. Hence the order function $\mathrm{ord}_P : \overline{\mathbb{K}}(C) \longrightarrow \mathbb{Z} \cup \{\infty\}$ is defined. Let $C_1 : f_1(x,y) = 0$ be an affine curve defined over $\overline{\mathbb{K}}$ (C_1 is not necessarily absolutely irreducible). We have seen in Example 20 that the $\overline{\mathbb{K}}$-rational function $f_1 \in \overline{\mathbb{K}}(C)$ has no poles. So $\mathrm{ord}_P(f_1) \geq 0$. Moreover, we saw in the mentioned example that the zeroes of $f_1 \in \overline{\mathbb{K}}(C)$ are the points of intersection between C and C_1. Then $P \in C \cap C_1$ if and only if $\mathrm{ord}_P(f_1) > 0$, and $\mathrm{ord}_P(f_1)$ can be seen as the number of times C and C_1 intersect at the point P. In fact, $\mathrm{ord}_P(f_1)$ is called the intersection multiplicity of C and C_1 at the point P.

Example 23. Let Ed be the affine curve of Example 11. We saw in Example 21 that $P = (0,1)$ is a nonsingular point of Ed. Take the affine curve $C_1 : f_1(x,y) = 1 - y + x^2 =$

0. Then the intersection multiplicity of C and C_1 at P, as defined on Example 22, is $\mathrm{ord}_P(f_1) = 4$. In fact, if we translate P into the origin $O = (0,0)$, we obtain the curves $\tilde{C} : -2y+2x^2+y^2+2x^2y-x^2y^2 = 0$ and $\tilde{C}_1 : \tilde{f}_1 = y-x^2 = 0$. We have that $\mathcal{M}_O(\tilde{C}) = (x)$ and $\tilde{f}_1 = x^4u$, with u invertible element of $\mathcal{O}_O(\tilde{C})$. Hence $\mathrm{ord}_O(\tilde{f}_1) = \mathrm{ord}_P(f_1) = 4$.

1.2.3 Divisors and genus

We deal with divisors of an absolutely irreducible, nonsingular projective curve X defined over a field $\mathbb{K}$. For an absolutely irreducible projective curve C defined over $\mathbb{K}$, which has some singular points, we take divisors on a nonsingular model X of C. A nonsingular model X of C is a nonsingular projective curve defined over $\mathbb{K}$, birationally equivalent to C over $\mathbb{K}$. One can show ([44, Theorem 3, Chapter 7]) that this model always exists, and it is unique up to isomorphism. Such model is not always plane. Nevertheless, it is plane for twisted Edwards curves, that is the case of interest for this thesis.

Example 24. Let $\overline{Ed}$ be the twisted Edwards curve of Example 12. We saw in Example 21 that the two points at infinity Ω_1 and Ω_2 are singular points of $\overline{Ed}$. Moreover, we saw in Example 18 that the curve $\overline{Em}$ is birationally equivalent to $\overline{Ed}$ over $\mathbb{K}$. Hence $\overline{Em}$ is the nonsingular model of $\overline{Ed}$. So, divisors on $\overline{Ed}$ are, by definition, divisors on $\overline{Em}$.

Divisors of a nonsingular projective curve. Let X be an absolutely irreducible, nonsingular projective curve defined over $\mathbb{K}$. A divisor on X is a formal sum

$$D = \sum_{P \in X} n_P P,$$

where $n_P \neq 0$ only for a finite number of points of X. The degree of D is $\deg(D) = \sum_{P \in X} n_P$. The set of divisors $\mathrm{Div}(X)$ is an abelian group with the formal sum. The subset of divisors of degree zero, denoted by $\mathrm{Div}^0(X)$, is a subgroup of $\mathrm{Div}(X)$. For a rational function $r \in \overline{\mathbb{K}}(X)$, we define the divisor of r as

$$\mathrm{div}(r) = \sum_{P \in X} \mathrm{ord}_P(r) P.$$

One can show that $\mathrm{div}(r) \in \mathrm{Div}^0(X)$, that is, r has only a finite number of zeroes that is equal to its number of poles ([44, Problem 4.17 and Proposition 1, Chapter 8]). A divisor $D \in \mathrm{Div}^0(X)$ is called a principal divisor if there exists $r \in \overline{\mathbb{K}}(X)$ such that $D = \mathrm{div}(r)$. The subset of principal divisors, denoted by $\mathrm{Princ}(X)$, is a subgroup of $\mathrm{Div}^0(X)$. We define the degree zero Picard group of the curve X as the quotient group

$$\mathrm{Pic}^0(X) = \mathrm{Div}^0(X)/\mathrm{Princ}(X).$$

Notice that, for each field isomorphism $\sigma : \overline{\mathbb{K}} \longrightarrow \overline{\mathbb{K}}$ such that $\sigma_{|\mathbb{K}} = \mathrm{id}_{\mathbb{K}}$, and each divisor $D = \sum_{P \in X} n_P P \in \mathrm{Div}(X)$, there is a natural way to apply σ to D. Namely, one defines

$$\sigma(D) = \sum_{P \in X} n_P \sigma(P) \in \mathrm{Div}(X),$$

where $\sigma(P) = \sigma([x_P, y_P, z_P]) = [\sigma(x_P), \sigma(y_P), \sigma(z_P)]$ for all $P \in X$. This definition projects on the quotient $\mathrm{Pic}^0(X)$. Hence, for $[D] \in \mathrm{Pic}^0(X)$, we define $\sigma([D]) = [\sigma(D)]$. Then, for each field extension $\mathbb{K} \subseteq \mathbb{L} \subseteq \overline{\mathbb{K}}$, we define the following subgroup of $\mathrm{Pic}^0(X)$:

$$\mathrm{Pic}^0(X)(\mathbb{L}) = \{[D] \in \mathrm{Pic}^0(X) : \sigma([D]) = [D] \text{ for all } \sigma \text{ such that } \sigma_{|\mathbb{L}} = \mathrm{id}_{\mathbb{L}}\}. \quad (1.3)$$

We call $\mathrm{Pic}^0(X)(\mathbb{L}) \subseteq \mathrm{Pic}^0(X)$ the group of $\mathbb{L}$-rational divisor classes of X.

Example 25. Let $\overline{Em}$ be the nonsingular projective curve of Example 17. We have that $D_1 = [-1,-1,1]+[-1,1,1]-2[0,0,1]$, $D_2 = [-1,-1,1]+[-1,1,1]-2[0,1,0] \in \mathrm{Div}^0(\overline{Em})$. Moreover, $\mathrm{div}(z/x) = 2[0,1,0] - 2[0,0,1]$. Hence $[D_1] = [D_2] \in \mathrm{Pic}^0(\overline{Em})$. We have also that $\mathrm{div}((x+z)/z) = [-1,-1,1] + [-1,1,1] - 2[0,1,0] = D_2$. Then $[D_1] = [D_2] = 0$ in $\mathrm{Pic}^0(\overline{Em})$.

Example 26 (Divisors of a singular projective curve)**.** We clarify the concept of divisors of a singular projective curve. We take the curve $\overline{Ed}$ of Example 21. By definition and by Example 24, we have that

$$\mathrm{Div}(\overline{Ed}) = \mathrm{Div}(\overline{Em}).$$

Let $\Phi = \overline{r_1} : \overline{Em} \longrightarrow \overline{Ed}$ be the birational map as in Example 17. To each divisor $D = \sum_{P\in\overline{Em}} n_P P$, one can associate the formal sum $\Phi(D)$ of points of $\overline{Ed}$, namely $\Phi(D) = \sum_{P\in\overline{Em}} n_P \Phi(P)$. Notice that Φ is not injective, more precisely $|\Phi^{-1}(P)| = 1$ for all $P \in \overline{Ed} \setminus \{\Omega_1, \Omega_2\}$, and $|\Phi^{-1}(\Omega_1)|=|\Phi^{-1}(\Omega_2)| = 2$. Hence there are different divisors that have the same image under Φ. Take now $r \in \overline{\mathbb{K}}(\overline{Ed})$: we define

$$\mathrm{div}(r) = \mathrm{div}(r \circ \Phi).$$

As we did before, we can take the image $\Phi(\mathrm{div}(r))$ of $\mathrm{div}(r)$, to obtain a formal sum of points on the singular curve $\overline{Ed}$.
For example, let $r = (y-z)/(y+z) \in \mathbb{K}(\overline{Ed})$. We have that $\mathrm{div}(r) = \mathrm{div}(r\circ\Phi) = \mathrm{div}(z/x) = 2[0,1,0] - 2[0,0,1]$. Moreover, we have $\Phi(\mathrm{div}(r)) = \Phi(2[0,1,0] - 2[0,0,1]) = 2[0,1,1] - 2[0,-1,1]$.

Genus of a projective curve. From the concept of divisors, one defines an important birational invariant for absolutely irreducible projective curves, namely the genus of the curve. We define the genus of an absolutely irreducible, nonsingular projective curve X defined over $\mathbb{K}$, as well as of an absolutely irreducible projective curve C defined over $\mathbb{K}$, whose birational model is X. For a divisor $D = \sum_{P\in X} n_P P \in \mathrm{Div}(X)$, we take the $\overline{\mathbb{K}}$-vector space

$$\mathcal{L}(D) = \{r \in \overline{\mathbb{K}}(X) : \mathrm{ord}_P(r) \geq -n_P\}.$$

We denote by $\ell(D)$ the dimension of $\mathcal{L}(D)$. One can show that $\ell(D)$ is finite ([44, Proposition 3, Chapter 8]). In addition, this dimension is related with the degree of the divisor. More precisely, Riemann's theorem ([44, Section 8.3]) states that there exists a natural number g such that, for all $D \in \mathrm{Div}(X)$, one has that

$$\ell(D) \geq \deg(D) + 1 - g. \tag{1.4}$$

We define the genus of X to be the minimum among these naturals.

Example 27 (Elliptic curves)**.** Take a projective curve of the form

$$E : y^2 z = F(x,z), \tag{1.5}$$

such that $F(x,1) \in \mathbb{K}[x]$ has degree 3 and no multiple roots. One can prove that this curve is absolutely irreducible and nonsingular. Moreover, it has only one point at infinity, namely the $\mathbb{K}$-rational point $P_\infty = [0,1,0]$.
We compute the genus g of E. It follows from Riemann's theorem ([44, Corollary 3, Section 8.3]), that

$$\ell(D) = \deg(D) + 1 - g, \text{ if the degree of } D \text{ is sufficiently large.} \tag{1.6}$$

This means that the inequality (1.4) of Riemann's theorem becomes an equality for divisors of large degree. Hence, to compute the genus of E, we search for a divisor of large degree, for which we are able to compute the dimension of its $\mathcal{L}$-space. We take the divisor $E_m = 3mP_\infty$ for $m \in \mathbb{Z}_{>0}$, of degree $\deg(E_m) = 3m$. One can show that $r \in \mathcal{L}(E_m)$ if and only if it is of the form $r = G/z^m$, where $G \in \overline{\mathbb{K}}[x, y, z]$ is a homogeneous polynomial of degree m. Denote V_m (resp. V_{m-3}) the $\overline{\mathbb{K}}$-vector space of the homogeneous polynomials of $\overline{\mathbb{K}}[x, y, z]$ of degree m (resp. $m-3$). For m large enough, we can define the short exact sequence

$$0 \longrightarrow V_{m-3} \xrightarrow{\varphi_1} V_m \xrightarrow{\varphi_2} \mathcal{L}(E_m) \longrightarrow 0,\ \varphi_1(H) = HF,\ \varphi_2(G) = G/z^m.$$

From this exact sequence, we compute

$$\ell(E_m) = \dim_{\overline{\mathbb{K}}} V_m - \dim_{\overline{\mathbb{K}}} V_{m-3} = (m+1)(m+2)/2 - (m-1)(m-2)/2 = 3m = \deg(E_m) + 1 - 1.$$

So, by (1.6) and the previous equality, we have that the genus of E is 1.

If we take an absolutely irreducible, nonsingular projective curve defined over $\mathbb{K}$, with at least one $\mathbb{K}$-rational point, one can show that also the vice versa holds true. More precisely, an absolutely irreducible, nonsingular projective curve of genus 1 defined over $\mathbb{K}$, with at least one $\mathbb{K}$-rational point, is isomorphic to a cubic plane curve defined over $\mathbb{K}$ ([6, Section 4.4.2.a]). Furthermore, if the characteristic of $\mathbb{K}$ is different from 2, then the curve has the form (1.5), up to isomorphism ([6, Section 4.4.2.a]). Absolutely irreducible, nonsingular projective curves of genus 1 defined over $\mathbb{K}$, with at least one $\mathbb{K}$-rational point, are called elliptic curves defined over $\mathbb{K}$. They are one of the main objects of this thesis. We speak in detail about them in the next section.

The genus of X allows to define a normal form for the elements of the Picard group $\mathrm{Pic}^0(X)$. This form is of particular interest for genus $g = 1$, that is, in the case of elliptic curves (see the previous example). In fact, in this case, such normal form establishes a natural one-to-one correspondence between divisor classes and points of the curve. Exploiting this bijection, an addition law between points can be derived. We conclude this section by describing the mentioned one-to-one correspondence.

Take E an absolutely irreducible, nonsingular projective curve of genus $g = 1$ defined over $\mathbb{K}$, with at least one $\mathbb{K}$-rational point. By definition, E is an elliptic curve defined over $\mathbb{K}$. We have the following result ([73, Theorem 11.2]).

Theorem 28. *Each divisor class $[D] \in Pic^0(E)$ has a representative of the form $[D] = [P - P_\infty]$, for some $P \in E$.*

We show such result for the case in which the characteristic of $\mathbb{K}$ is different from 2, and for $[D] = [P_1 - P_2]$, with $P_1, P_2 \neq P_\infty$. In Example 27, we recall that, if $\mathrm{char}(\mathbb{K}) \neq 2$, then E is a cubic of the form (1.5). To prove the thesis for $[D] = [P_1 - P_2]$, with $P_1, P_2 \neq P_\infty$, we have to prove that there exists $P_3 \in E$ such that $[P_1 - P_2] = [P_3 - P_\infty]$. This is equivalent to saying that there is a rational function $r \in \overline{\mathbb{K}}(E)$ such that $\mathrm{div}(r) = P_1 + P_\infty - P_2 - P_3$. Write $P_1 = [x_1, y_1, z_1]$, $P_2 = [x_2, y_2, z_2]$, with $z_1, z_2 \neq 0$. Let $\hat{P}_1 = [x_1, -y_1, z_1]$, $\hat{P}_2 = [x_2, -y_2, z_2] \in E$ be the symmetric points of P_1 and P_2 respectively, with respect to the x-axis. Let $\ell : \ell(x, y, z) = 0$ be the line through P_1 and $\hat{P}_2$. Since E has the form (1.5) (see Example 27), its defining polynomial has degree 3 and ℓ intersects E in a third point, say $\hat{P}_3 = [x_3, y_3, z_3]$. Let $P_3 = [x_3, -y_3, z_3]$. Let $v_2 : v_2(x, z) = x - x_2 z = 0$, $v_3 : v_3(x, z) = x - x_3 z = 0$ be the vertical lines through P_2 and P_3. We compute

the divisors $\mathrm{div}(\ell(x,y,z)/z) = P_1 + \hat{P}_2 + \hat{P}_3 - 3P_\infty$, $\mathrm{div}(v_2(x,z)/z) = P_2 + \hat{P}_2 - 2P_\infty$, $\mathrm{div}(v_3(x,z)/z) = P_3 + \hat{P}_3 - 2P_\infty$. Therefore, the rational function

$$r = ((\ell(x,y,z)/z) \cdot (z/v_2(x,z)) \cdot (z/v_3(x,z)))$$

is such that

$$\mathrm{div}(r) = (P_1 + \hat{P}_2 + \hat{P}_3 - 3P_\infty) - (P_2 + \hat{P}_2 - 2P_\infty) - (P_3 + \hat{P}_3 - 2P_\infty) = P_1 + P_\infty - P_2 - P_3,$$

as required. With analogous procedures one can show the thesis of Theorem 28 in the general case.

It follows from Theorem 28 that we can define the bijection

$$\Phi : \mathrm{Pic}^0(E) \longrightarrow E,\ [P - P_\infty] \mapsto P, \tag{1.7}$$

between divisor classes and points of the curve. As a consequence, the set of points of E is endowed with the structure of an abelian group, with the addition law derived from $\mathrm{Pic}^0(E)$:

$$P_1 \oplus P_2 = \Phi(\Phi^{-1}(P_1) + \Phi^{-1}(P_2)), \text{ for } P_1, P_2 \in E.$$

Moreover, notice that, for each field extension $\mathbb{K} \subseteq \mathbb{L} \subseteq \overline{\mathbb{K}}$, we have

$$\mathrm{Pic}^0(E)(\mathbb{L}) = \{[D] = [P - P_\infty] \in \mathrm{Pic}^0(E) :\ \sigma(P) = P \text{ for all } \sigma \text{ such that } \sigma_{|\mathbb{L}} = \mathrm{id}_{\mathbb{L}}\} =$$

$$= \{[D] = [P - P_\infty] \in \mathrm{Pic}^0(E) :\ P \in E(\mathbb{L})\},$$

by Definition (1.3) and Theorem 28. It follows that the map Φ establishes a one-to-one correspondence between $\mathrm{Pic}^0(E)(\mathbb{L})$ and $E(\mathbb{L})$. Therefore, the set $E(\mathbb{L})$ of $\mathbb{L}$-rational points of E is an abelian group endowed with the addition law derived from $\mathrm{Pic}^0(E)(\mathbb{L})$ via the map Φ. Moreover, for each double field extension $\mathbb{K} \subseteq \mathbb{L}_1 \subseteq \mathbb{L}_2 \subseteq \overline{\mathbb{K}}$, we have that $E(\mathbb{L}_1)$ is a subgroup of $E(\mathbb{L}_2)$. Such point addition turns out to be very efficient to compute. This is one of the aspects that make elliptic curves interesting for applications in cryptography. We recall this addition law, together with its geometric construction, in the next section.

1.3 Elliptic and twisted Edwards curves in cryptography

Among the groups that are suitable for cryptographic applications, the groups of points of elliptic curves and twisted Edwards curves play an important role. Elliptic curves were first proposed for cryptographic applications in 1985 (see [59]). Nowadays, they are used in numerous real cryptosystems. We introduce them in Subsection 1.3.1. Twisted Edwards curves have been proposed for cryptography in 2007. We speak about them in Subsection 1.3.2. In this subsection, we point out the security and efficiency advantages of twisted Edwards curves over the classical elliptic curves. Semaev's summation polynomials are a special family of polynomials associated to an elliptic curve or to a twisted Edwards curve. They allow to find relations between points on the given curve. Therefore, they have a big potential in cryptographic applications, both from the constructive point of view (efficiency improvements) and from the destructive one (DLP attacks). We introduce such polynomials in Subsection 1.3.3.

1.3.1 Elliptic curves

Elliptic curves in short Weierstrass form were first proposed for the construction of DLP-based cryptosystems by Miller in 1985 (see [59]), and independently by Koblitz in 1987 (see [52]). Until then, the groups that were mainly used in DLP cryptography were the multiplicative cyclic groups $\mathbb{Z}_p \setminus \{0\}$, where p is a prime number. The advantage of elliptic curve cryptography over this setting is that keys of shorter bit-length are required, in order to reach the same level of security.

Definition 29 (Elliptic curve)**.** Let $\mathbb{K}$ be a field. An elliptic curve defined over $\mathbb{K}$ is an absolutely irreducible, nonsingular projective curve of genus $g = 1$, with at least one $\mathbb{K}$-rational point.

By Theorem 28, the map (1.7) of Section 2 gives a one-to-one correspondence between the divisor classes of the Picard group of an elliptic curve and the points of the curve. Therefore, a group law on the points of the curve is defined from the group law on the Picard group of the curve, via the just mentioned one-to-one correspondence. In Example 27, we recall that elliptic curves are cubic curves. More precisely, each elliptic curve defined over $\mathbb{K}$ is isomorphic to a cubic plane curve defined over $\mathbb{K}$. The defining cubic polynomial of the elliptic curve can have a special form, such as the short Weierstrass form or the Montgomery form. Such special forms make the group operation between the points of the curve particularly easy. It turns out that efficient and fast formulas can be used to compute addition between points of elliptic curves. Moreover, this group operation has a simple geometric interpretation. Namely, addition between two points of the elliptic curve is defined by the intersection between the curve and a specific line.

Definition 30 (Elliptic curve in short Weierstrass form and in Montgomery form)**.** An elliptic curve defined over the field $\mathbb{K}$ is in short Weierstrass form if it is of the form :

$$E_W : y^2z = x^3 + Axz^2 + Bz^3, \text{ with } A, B \in \mathbb{K} \text{ such that } 4A^3 + 27B^2 \neq 0.$$

An elliptic curve defined over $\mathbb{K}$ is in Montgomery form if it is of the form :

$$E_M : By^2z = x^3 + Ax^2z + xz^2, \text{ with } A \in \mathbb{K} \setminus \{\pm 2\},\ B \in \mathbb{K} \setminus \{0\}.$$

We have seen in Example 27 that, if the characteristic of $\mathbb{K}$ is different from 2, then an elliptic curve defined over $\mathbb{K}$ has the form (1.5), up to isomorphism. Moreover, if the characteristic of $\mathbb{K}$ is not $2, 3$, then each elliptic curve is isomorphic over $\mathbb{K}$ to an elliptic curve in short Weierstrass form (see [6, Section 4.4.2.*a*]). On the other hand, not every elliptic curve is isomorphic, over $\mathbb{K}$, to an elliptic curve in Montgomery form. Anyway, such isomorphism always exists over the algebraic closure $\overline{\mathbb{K}}$ of the field of definition $\mathbb{K}$. Notice that we saw a special case of the following proposition in Example 16 and Example 18 of Section 1.2.

Proposition 31. ([63, Proposition 1]) *Let $\mathbb{K}$ be a field of characteristic different from $2, 3$.*

- *Let $E_M : By^2z = x^3 + Ax^2z + xz^2$ be an elliptic curve in Montgomery form defined over $\mathbb{K}$. Then E_M is isomorphic over $\mathbb{K}$ to the elliptic curve in short Weierstrass form $E_W : y^2z = x^3 + axz^2 + bz^3$, where $a = \frac{3-A^2}{3B^2}$, $b = \frac{2A^3-9A}{27B^3}$, via the isomorphism*

$$\phi : E_M \longrightarrow E_W,\ [x, y, z] \mapsto [x + (A/3)z, y, Bz].$$

- *Let $E_W : y^2 = x^3 + ax + b$ be an elliptic curve in short Weierstrass form defined over $\mathbb{K}$. Suppose that the polynomial $x^3 + ax + b$ has a root $\alpha \in \mathbb{K}$, and suppose that $(3\alpha^2 + a)$ is a quadratic residue in $\mathbb{K}$. Let $s \in \mathbb{K}$ be a square root of $(3\alpha^2 + a)^{-1}$. Then E_W is isomorphic over $\mathbb{K}$ to the elliptic curve in Montgomery form $E_M : By^2 = x^3 + Ax^2 + x$, where $A = 3\alpha s$, $B = s$, via the isomorphism*

$$\phi : E_W \longrightarrow E_M,\ [x, y, z] \mapsto [s(x - \alpha z), sy, z].$$

Elliptic curves in Montgomery form were introduced by Montgomery in 1987, in [61]. Elliptic curves of this form provide very efficient scalar multiplication (see [61]).

Point addition on elliptic curves. From now on, we take a field $\mathbb{K}$ of characteristic different from 2, 3. Let E be an elliptic curve defined over $\mathbb{K}$, written in short Weierstrass form or in Montgomery form. We denote the point at infinity of E by $P_\infty = [0, 1, 0]$. For each field extension $\mathbb{K} \subseteq \mathbb{L} \subseteq \overline{\mathbb{K}}$, let $E(\mathbb{L})$ be the set of $\mathbb{L}$-rational points of E. We have that this set is an abelian group, endowed with the addition between points which derives from the operation law on the Picard group of the curve. We denote such operation of point addition by $\oplus$. Moreover, we denote the corresponding group structure on $E(\mathbb{L})$ with $(E(\mathbb{L}), \oplus)$. We recall the geometric interpretation of the addition between two points P and $Q \in E(\mathbb{L})$. Take the line through P and Q. Notice that, if $P = Q$, the line is the tangent to the elliptic curve at the point P. Such line intersects the elliptic curve in a third point. The symmetric of this point with respect to the x-axis is the addition point $P \oplus Q$. The neutral element of point addition is the point at infinity P_∞. For each point P, the additive inverse $-P$ of P is the symmetric of P with respect to the x-axis.

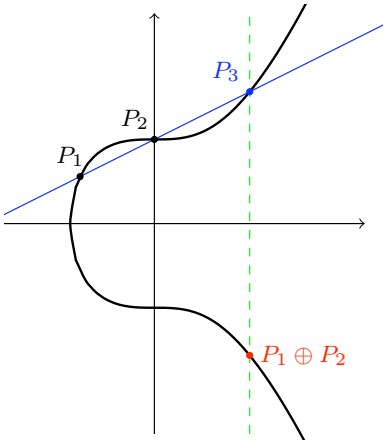

Figure 1. Geometric construction for addition between two points of an elliptic curve.

Elliptic curves for cryptography

We analyze elliptic curves from the cryptographic point of view. For groups of points of elliptic curves, we focus on the security and efficiency aspects introduced in Section 1.1. This discussion points out the crucial importance of these groups in cryptography.

For cryptographic applications, we take elliptic curves E defined over finite fields $\mathbb{F}_q$. Throughout our work, we make the assumption that the characteristic of $\mathbb{F}_q$ is different from 2, 3. We make this choice for ease of exposition and computation. Anyway, most of the given results can be applied, with some modification, also to the case in which $\mathrm{char}(\mathbb{F}_q) = 2, 3$. Given the elliptic curve E defined over $\mathbb{F}_q$, we take the group $E(\mathbb{F}_q)$ of

base field-rational points of E. We will see in Section 1.5 that, if q is a prime number, the group $E(\mathbb{F}_q)$ provides the optimal security level against DLP attacks. This means that, when q is a prime, the best known DLP attack in $E(\mathbb{F}_q)$ is the generic Pollard's rho method: we refer to Section 1.5.1 for more details on the concept of optimal DLP security. Moreover, we choose groups $E(\mathbb{F}_q)$ such that they are cyclic of prime order, or such that there is a large prime that divides the order of the group. The reason why we take cyclic subgroups of large prime order is that, as we mentioned in Section 1.1, we can always reduce a DLP in a cyclic group G to DLP's in the subgroups of G of prime order, thanks to the Pohlig-Hellman strategy.
Let us sum up the setting. For cryptographic applications, we take groups $E(\mathbb{F}_q)$, where q is a prime number. Moreover, we choose $E(\mathbb{F}_q)$ such that it is cyclic of prime order. An alternative is to choose a subgroup of $E(\mathbb{F}_q)$ of large prime order p. When we say that p is a large prime, we mean that $p \in O(|E(\mathbb{F}_q)|)$. By the Hasse-Weil theorem ([73, Theorem 4.10]), one has that $|E(\mathbb{F}_q)| \in O(q)$. Hence our groups have order $O(q)$. In such groups, the best known DLP attack is the Pollard's rho method. The complexity of the method is $O(q^{\frac{1}{2}})$ (the square root of the order of the group).

Security of elliptic curve cryptography. We mentioned before that the advantage of using elliptic curves in DLP cryptography, rather than cyclic groups of the type $\mathbb{Z}_p \setminus \{0\}$, is that keys of shorter bit-length are needed to obtain the same level of security. In the document [9], the National Institute of Technology (NIST, USA) recommends parameters for $\mathbb{Z}_p \setminus \{0\}$ and $E(\mathbb{F}_q)$, in order to perform secure Diffie-Hellman key exchange. The prescribed bit-length for p is 2048, while the recommended bit-length for q is only 224. In November 1997, the cryptography agency Certicom published a list of challenges to solve DLP's on given elliptic curves defined over prime fields $\mathbb{F}_q$ (the Certicom ECC Challenge, see [26]). The prime q is of bit-length 109, 131, 163, 191, 239, 259. The challenge for bit-length 109 has been solved in 7 years. The one for bit-length 131 has not been solved yet, but it is supposed to be feasible. The challenges for the other bit-lengths are supposed to be infeasible. The estimated time to solve them is $2.3 \cdot 10^{15}$, $4.8 \cdot 10^{19}$, $1.4 \cdot 10^{27}$, $3.7 \cdot 10^{45}$ machine days for bit-length 163, 191, 239, 259 respectively. We refer to the website https://safecurves.cr.yp.to/, developed by D. Bernstein and T. Lange, for a list of elliptic curves that are considered safe for cryptographic applications. Notice that the security of such curves does not only regards the hardness of solving the DLP. We refer to the introduction of the website https://safecurves.cr.yp.to/ for a discussion on the topic.

Efficiency of elliptic curve cryptography. We point out that there are fast algorithms to compute the order of $E(\mathbb{F}_q)$. By the Hasse-Weil theorem ([73, Theorem 4.10]), such computation reduces to the computation of the characteristic polynomial of the Frobenius endomorphism of E. We recall that the Frobenius endomorphism φ of E is the endomorphism

$$\varphi : E \longrightarrow E,\ [x, y, z] \mapsto [x^q, y^q, z^q]. \tag{1.8}$$

The characteristic polynomial of φ is the polynomial

$$\chi_\varphi(x) = x^2 - ax + q \in \mathbb{Z}[x], \tag{1.9}$$

where a is the unique integer k such that $\varphi^2 - k\varphi + q = 0$. The Hasse-Weil theorem states that

$$|E(\mathbb{F}_q)| = q + 1 - a.$$

Hence, computing the order of $E(\mathbb{F}_q)$ reduces to computing the coefficient a of χ_φ. This coefficient can be efficiently computed, with the so-called Schoof's algorithm ([73, Section 4.5]).

Concerning the performance of group operations in $E(\mathbb{F}_q)$, we already mentioned that efficient and fast formulas are known. We refer to [6, Chapter 13] for a detailed and complete survey on efficient arithmetic over elliptic curves. We refer also to the explicit formulas database for elliptic curves developed by D. Bernstein and T. Lange, at http://hyperelliptic.org/EFD.

We now focus on the aspect of optimal representations for elements of $E(\mathbb{F}_q)$. We refer to Definition 7 of Section 1.1 and to the subsequent notations and observations. The family of groups we are interested in is $\mathcal{G}_1 = (E(\mathbb{F}_q))_{q,E}$. By the Hasse-Weil theorem, we have $|E(\mathbb{F}_q)| \in O(q)$. A well known optimal representation for $\mathcal{G}_1$ is $\mathcal{R}_1 = (\mathcal{R}_{1,q,E})_{q,E}$, with

$$\mathcal{R}_{1,q,E} : E(\mathbb{F}_q) \setminus \{P_\infty\} \longrightarrow \mathbb{F}_q,\ [x, y, 1] \mapsto x. \tag{1.10}$$

For $\mathcal{R}_1$, we have $c = e = 0$ and $|\mathcal{R}_{1,q,E}^{-1}(x)| \leq 2 = d$, for all $x \in \mathbb{F}_q$ and for each group $E(\mathbb{F}_q)$. Notice that $\mathcal{R}_1$ is not defined in the neutral element of the group, as we alerted in Remark 10. This representation is defined over the affine points of $E(\mathbb{F}_q)$. Namely, one represents an affine $\mathbb{F}_q$-rational point $P = [x_0, y_0, 1]$ of E via its x-coordinate x_0. Then, one uses the short Weierstrass affine equation of E, that is $y^2 = x_0^3 + Ax_0 + B$, to recover the y-coordinate of P up to sign. Hence, as we pointed out in Section 1.1, it is enough to add one extra bit to the representation in order to recover the point P without any ambiguity. More precisely, we establish a convention to distinguish a priori between the two square roots $\sqrt{t}$ and $-\sqrt{t}$ of a quadratic residue $t \in \mathbb{F}_q$. Then, we take the injective maps:

$$\mathcal{R}'_{1,q,E} : E(\mathbb{F}_q) \setminus \{P_\infty\} \longrightarrow \mathbb{F}_q \times \mathbb{F}_2,\ [x_0, y_0, 1] \mapsto \begin{cases} (x_0, 0) \text{ if } y_0 = \sqrt{y^2} \\ (x_0, 1) \text{ otherwise} \end{cases}.$$

We point out that, for the optimal representation $\mathcal{R}_1$, compression and decompression of group elements are efficient in practice. The process of compression $\mathcal{R}_{1,q,E}([x, y, 1]) = x$ is costless. The process of decompression requires the computation of a square root in $\mathbb{F}_q$. Thanks to the fast compression and decompression algorithms, such optimal representation is widely used in cryptographic applications.

The use of the optimal representation $\mathcal{R}_1$ raises the issue of its efficient integration with the fast arithmetic of the group. We widely treat this aspect in Section 1.4, for the case of trace-zero subgroups. Notice that the operation of addition between P and $Q \in E(\mathbb{F}_q)$ is not defined if we use the optimal coordinates of $\mathcal{R}_1$. Denote by x_P the x-coordinate of a point $P \in E(\mathbb{F}_q)$. Given x_P and x_Q, it is not possible to compute $\mathcal{R}_{1,q,E}(P \oplus Q) = x_{P \oplus Q}$. In fact, we cannot distinguish between $x_{P \oplus Q}$ and $x_{P \oplus (-Q)}$. Nevertheless, scalar multiplication is still defined in the optimal coordinates of $\mathcal{R}_1$. Given $k \in \mathbb{Z}$ and x_P, one can compute $\mathcal{R}_{1,q,E}(kP) = x_{kP}$. In order to do it, one can decompress $\mathcal{R}_{1,q,E}(P)$ to recover P up to sign, then perform scalar multiplication kP (or $k(-P)$) in $E(\mathbb{F}_q)$, finally perform compression $\mathcal{R}_{1,q,E}(kP) = \mathcal{R}_{1,q,E}(k(-P)) = x_{kP}$. An alternative approach is computing scalar multiplication directly in the optimal coordinates of $\mathcal{R}_1$. In such a way, decompression and compression of points are not required. This is the approach of the so-called Montgomery's ladder algorithm for x-only scalar multiplication (see [6, Section 13.2.3.d]). The method turns out to be the best strategy to perform x-only scalar multiplication in $E(\mathbb{F}_q)$ (see [64]). We draw inspiration from this technique to construct our scalar multiplication algorithm for trace-zero elements, that we explain in Chapter 4. We now briefly describe a basic version of Montgomery's ladder algorithm for x-only scalar

multiplication. In the sequel, we will be able to get the interesting analogies between this method and our original technique.

Montgomery's ladder algorithm for x-only scalar multiplication is based on efficient formulas to compute $x_{P\oplus Q}$ given x_P, x_Q and $x_{P\oplus(-Q)}$. These formulas were first proposed by Montgomery in [61] for elliptic curves in Montgomery form. The algorithm uses a doubling formula $D(x_P)$ to compute x_{2P} given x_P, and an addition formula $A(x_P, x_Q, x_{P\oplus(-Q)})$ to compute $x_{P\oplus Q}$ given x_P, x_Q and $x_{P\oplus(-Q)}$. It performs as follows.

Algorithm 1 (Montgomery's ladder algorithm for x-only scalar multiplication)**.**

Input : x_P, for $P \in E(\mathbb{F}_q)$, and $k \in \mathbb{Z}$
Output: x_{kP}

---

1 : $k \leftarrow \sum_{i=0}^{\ell-1} k_i 2^i$ the binary representation of k, where $\ell = \lceil \log_2 k \rceil$ and $k_{\ell-1} = 1$
2 : $x_1 \leftarrow x_P$, $x_2 \leftarrow D(x_P)$
3 : **for** $i = \ell - 2, \cdots, 0$ **do**
4 : **if** $k_i = 0$ **then**
5 : $x_2 \leftarrow A(x_1, x_2, x_P)$, $x_1 \leftarrow D(x_1)$
6 : **else**
7 : $x_1 \leftarrow A(x_1, x_2, x_P)$, $x_2 \leftarrow D(x_2)$
8 : **end if**
9 : **end for**
10 : **return** x_1

For $i = \ell - 1, \cdots, 0$, let $H_i = \sum_{j=i}^{\ell-1} k_j 2^{j-i}$. Notice that, at step i, the algorithm computes $x_1 = x_{(H_i)P}$ and $x_2 = x_{(H_i+1)P}$. Hence, it correctly outputs $x_{(H_0)P} = x_{kP}$. Moreover, at step $i-1$ (for $i = \ell - 1, \cdots, 1$), the algorithm can compute x_1 or x_2 with the addition formula A. In fact, we have the information $x_{\overline{P}} = x_{(H_i)P}$, $x_{\overline{Q}} = x_{(H_i+1)P}$, that are x_1 and x_2 at the previous step i. Furthermore, we have the extra information $x_{\overline{P}\oplus(-\overline{Q})} = x_P$. So, at step $i-1$, it is possible to compute $x_{\overline{P}\oplus\overline{Q}} = A(x_{\overline{P}}, x_{\overline{Q}}, x_{\overline{P}\oplus(-\overline{Q})}) \in \{x_1, x_2\}$.

The following scheme summarize the important features of elliptic curve cryptography.

Scheme 3.

Elliptic curves in cryptography.
Elliptic groups for cryptosystems: cyclic subgroups of $E(\mathbb{F}_q)$ of large prime order, q prime number.

Security.

- Optimal DLP security: only Pollard's rho attack, of complexity $O(q^{\frac{1}{2}})$.
- Shorter keys with respect to groups $\mathbb{Z}_p \setminus \{0\}$, to reach the same security level.

Efficiency.

- Fast algorithm to compute the group order: Schoof's algorithm.
- Efficient formulas for point addition : http://hyperelliptic.org/EFD
- Optimal representation $\mathcal{R}_1$ for group elements (see (1.10)), with fast compression/decompression.

1.3.2 Twisted Edwards curves

Twisted Edwards curves have been introduced quite recently in cryptography. Edwards curves (a special case of twisted Edwards curves) were first defined by Edwards in 2007 (see

[34]). In the same year, they were proposed for cryptographic applications by Bernstein and Lange (see [13]). They were generalized to twisted Edwards curves by Bernstein et al. in 2008 (see [12]). The use of twisted Edwards curves in cryptography is an alternative to the standard use of elliptic curves. Twisted Edwards curves provide some cryptographic advantages over elliptic curves, both from the point of view of efficient arithmetic and of security.

Definition 32 (Twisted Edwards curve)**.** Let $\mathbb{K}$ be a field of characteristic different from $2, 3$. A twisted Edwards curve defined over $\mathbb{K}$ is an absolutely irreducible projective curve of the form

$$E_{a,d} : ax^2z^2 + y^2 = z^4 + dx^2y^2, \text{ with } a, d \in \mathbb{K} \setminus \{0\} \text{ and } a \neq d.$$

An Edwards curve is a twisted Edwards curve with $a = 1$.

A twisted Edwards curve has two points at infinity, namely $\Omega_1 = [1, 0, 0]$ and $\Omega_2 = [0, 1, 0]$. It is nonsingular in the affine plane. On the other hand, both points at infinity are singular and they have multiplicity 2 (see Example 21). The following theorem states that each twisted Edwards curve defined over $\mathbb{K}$ is birationally equivalent over $\mathbb{K}$ to an elliptic curve. More precisely, a twisted Edwards curve defined over $\mathbb{K}$ is birationally equivalent over $\mathbb{K}$ to an elliptic curve in Montgomery form. Furthermore, also the vice versa holds true. An elliptic curve in Montgomery form defined over $\mathbb{K}$ is birationally equivalent over $\mathbb{K}$ to a twisted Edwards curve.

Theorem 33. ([12, Theorem 3.2]) *Let $\mathbb{K}$ be a field of characteristic different from $2, 3$. Let $a, d \in \mathbb{K} \setminus \{0\}$, with $a \neq d$. Let $A = 2\frac{a+d}{a-d} \in \mathbb{K} \setminus \{\pm 2\}$, $B = \frac{4}{a-d} \in \mathbb{K} \setminus \{0\}$. Then the twisted Edwards curve $E_{a,d} : ax^2z^2 + y^2z^2 = z^4 + dx^2y^2$ is birationally equivalent over $\mathbb{K}$ to the elliptic curve in Montgomery form $E_M : By^2z = x^3 + Ax^2z + xz^2$, via the birational map*

$$\Phi : E_M \longrightarrow E_{a,d}$$

$$[x, y, z] \mapsto [x(x + z), y(x - z), y(x + z)], \text{ if } [x, y, z] \in E_M \setminus \{[0, 0, 1], [0, 1, 0]\},$$

$$[x, y, z] \mapsto [By(x + z), By^2 - (x^2 + Axz + z^2), By^2 + (x^2 + Axz + z^2)],$$

if $[x, y, z] \in E_M \setminus \{[x, 0, 1] \in E_M : x \neq 0\}$.

The birational inverse of Φ is

$$\Phi^{-1} : E_{a,d} \setminus \{\Omega_1, \Omega_2\} \longrightarrow E_M,$$

$$[x, y, z] \mapsto [x(z + y), z(z + y), x(z - y)] \text{ if } [x, y, z] \neq [0, -1, 1],$$

$$[x, y, z] \mapsto [z(z + y)(z - y), x(az^2 - dy^2), z(z - y)^2] \text{ if } [x, y, z] \neq [0, 1, 1].$$

It follows from the previous theorem that the nonsingular model of a twisted Edwards curve is an elliptic curve in Montgomery form. Hence, the genus of a twisted Edwards curve is $g = 1$ (see Section 1.2).

Point addition on twisted Edwards curves. We can define an operation of addition between two affine points of the twisted Edwards curve $E_{a,d}$. In order to do this, we use

the group law on the elliptic curve in Montgomery form E_M, and the birational map Φ of Theorem 33. We denote such operation by $\hat{\oplus}$. For P_1, $P_2 \in E_{a,d} \setminus \{\Omega_1, \Omega_2\}$, we define

$$P_1 \hat{\oplus} P_2 = \Phi(\Phi^{-1}(P_1) \oplus \Phi^{-1}(P_2)).$$

Notice that such operation is not defined for the points at infinity Ω_1 and $\Omega_2 \in E_{a,d}$. In fact, $\mathrm{Dom}(\Phi^{-1}) = E_{a,d} \setminus \{\Omega_1, \Omega_2\}$. Moreover, it can be shown that, for P_1, $P_2 \in E_{a,d} \setminus \{\Omega_1, \Omega_2\}$, we have $P_1 \hat{\oplus} P_2 \in E_{a,d} \setminus \{\Omega_1, \Omega_2\}$. Denote by E the affine dehomogenization of $E_{a,d}$ with respect to the variable z. By Theorem 33, the definition of point addition on elliptic curves and the previous discussion, one has that, for each field extension $\mathbb{K} \subseteq \mathbb{L} \subseteq \overline{\mathbb{K}}$, $(E(\mathbb{L}), \hat{\oplus})$ is an abelian group. The neutral element is the point $\mathcal{O} = (0, 1)$. The additive inverse of $P = (x, y) \in E(\mathbb{L})$ is $-P = (-x, y)$, the symmetric of P with respect to the y-axis. We recall below the addition formulas for point addition on a twisted Edwards curve.

Proposition 34. (Twisted Edwards addition law, [13, Section 3], [12, Section 6]) *The sum of two points $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ of $E(\mathbb{L})$ is defined as*

$$P_1 \hat{\oplus} P_2 = (x_1, y_1) + (x_2, y_2) = \left(\frac{x_1 y_2 + x_2 y_1}{1 + d x_1 x_2 y_1 y_2}, \frac{y_1 y_2 - a x_1 x_2}{1 - d x_1 x_2 y_1 y_2} \right).$$

The geometric construction for point addition on the Montgomery curve E_M can be mapped to the birationally equivalent twisted Edwards curve $E_{a,d}$, via the birational map Φ. It follows that the twisted Edwards addition law has a simple geometric interpretation.

Proposition 35. ([2, Section 4]) *Let P_1, P_2 be two affine points of $E_{a,d}$. Let C be the projective conic which intersects $E_{a,d}$ in the points P_1, P_2, $\mathcal{O}' = [0, -1, 1]$ with multiplicity 1, and in the points Ω_1, Ω_2 with multiplicity 2. Then the point $P_1 \hat{\oplus} P_2$ is the symmetric with respect to the y-axis of the eighth point of intersection between $E_{a,d}$ and C.*

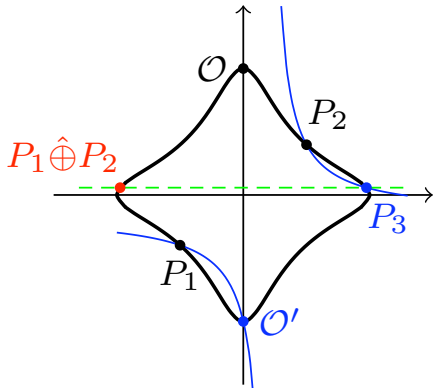

Figure 2. Geometric construction for addition between two affine points of a twisted Edwards curve.

Comparison between elliptic curves and twisted Edwards curves in cryptography

Cryptography on twisted Edwards curves is analogous to elliptic curve cryptography. Namely, we take groups of base field-rational affine points $E(\mathbb{F}_q)$, where q is a prime number. We take $E(\mathbb{F}_q)$ such that it is cyclic of prime order. Alternatively, we take a subgroup of $E(\mathbb{F}_q)$ of large prime order. The use of twisted Edwards curves in cryptography has some advantages over the traditional use of elliptic curves. The main advantage is that the arithmetic on twisted Edwards curves is faster. In fact, one can exploit the

double symmetry of the curve to speed up the computation. Notice that elliptic curves in short Weierstrass form and in Montgomery form are symmetric with respect to the x-axis only, while twisted Edwards curves are symmetric with respect both to the x-axis and the y-axis. Fast formulas for computation on Edwards curves and twisted Edwards curves are studied and given, for example, in [13], [14], [11][12], [19]. For point counting on twisted Edwards curves, we can apply the same fast algorithms that we know for elliptic curves, thanks to the birational map between $E_{a,d}$ and E_M. In addition, we have an analogous optimal representation for base field-rational points, with efficient compression and decompression algorithms. Namely, an optimal representation for the family of twisted Edwards groups $\mathcal{G}'_1 = (E(\mathbb{F}_q))_{q,E_{a,d}}$ is $\mathcal{R}'_1 = (\mathcal{R}'_{1,q,E_{a,d}})_{q,Ea,d}$, with

$$\mathcal{R}'_{1,q,E_{a,d}} : E(\mathbb{F}_q) \longrightarrow \mathbb{F}_q,\ (x,y) \mapsto y. \tag{1.11}$$

Also in this case, we have $e = c = 0$ and $d = 2$. Compression and decompression algorithms are analogous to those of $\mathcal{R}_1$ for the elliptic curve setting.

Concerning security, one has that the DLP on a twisted Edwards curve is as difficult as the corresponding DLP on the birationally equivalent elliptic curve in Montgomery form. This is a straightforward consequence of the presence of the birational map between the two curves. On the other hand, not all elliptic curves are isomorphic, over the field of definition, to elliptic curves in Montgomery form (see Proposition 31). Hence, one can conclude that the DLP on twisted Edwards curves is at most as difficult as the DLP on elliptic curves. However, it is not known if the two problems are equivalent. Anyway, there are elliptic curves in Montgomery form, with the corresponding birationally equivalent twisted Edwards form, that nowadays are considered secure for cryptography. An example is the curve Curve25519 used by WhatsApp encryption. We refer to [10] for a detailed analysis of the security of this curve. A security aspect in which twisted Edwards curves are better than elliptic curves is resistance against side-channel attacks (see [6, Chapter 28, 29]). Such type of cryptographic attacks consists of analyzing the physical behavior of the chip that performs the operations, in order to draw information about the exchanged data. The analysis can regard the timing of the computation process, the measure of power consumption, or the measure of the emitted electromagnetic radiation. As an example, we take a standard double-and-add algorithm to perform the scalar multiplication $kP = 7P$. The chip proceeds step by step, computing $2P$ (doubling), $2P + P = 3P$ (addition), $2(3P) = 6P$ (doubling), $6P + P = 7P$ (addition). Suppose that different formulas are used for doubling and addition: this is the case of elliptic curves in short Weierstrass form. In this case, a side channel attack lets the opposer distinguish between the two operations performed by the processor. Hence, the opposer can recover information about the scalar k. Twisted Edwards curves avoid this kind of attack. In fact, their addition formulas are strongly unified. This means that the same formula holds for both addition and doubling (see for example [13], [12]).

We sum up in the scheme below the most important aspects of twisted Edwards curve cryptography, as well as the main advantages of these curves over the standard elliptic curves.

Scheme 4.

Twisted Edwards curves VS elliptic curve cryptography.
Twisted Edwards groups for cryptosystems: cyclic subgroups $E(\mathbb{F}_q)$ of large prime order, q prime number.

Security.

- DLP as hard as in the birationally equivalent elliptic curve in Montgomery form.

- More security against side-channel attacks, thanks to the strongly unified formulas.

Efficiency.

- Fast algorithms to compute the group order as in the elliptic curve case, and analogous optimal representation $\mathcal{R}_1'$ (see (1.11)).
- Faster formulas for point addition, thanks to the double symmetry of the curve.

1.3.3 Summation polynomials of elliptic and twisted Edwards curves

Let $\mathbb{K}$ be a field of characteristic different from $2, 3$. Let E be an elliptic curve defined over $\mathbb{K}$. Let $E_{a,d}$ be a twisted Edwards curve defined over $\mathbb{K}$. We unify the notations of point addition on elliptic and twisted Edwards curves, of the previous subsections. We denote by $\oplus$ the point addition on E or on $E_{a,d}$. Moreover, we denote by $\mathcal{O}$ the neutral element of the operation. The summation polynomials of the elliptic curve, or of the twisted Edwards curve, are a family of polynomials that allow to study the geometry of the curve itself. They were introduced by Semaev in [67] for the case of elliptic curves. We give below the definition from the original paper ([67, Section 2]), together with its straightforward adaptation to the case of twisted Edwards curves ([40]).

Definition 36 (Summation polynomials of elliptic and twisted Edwards curves)**.** Let $t \geq 2$. A polynomial $f_t(x_1, \cdots, x_t) \in \mathbb{K}[x_1, \cdots, x_t]$ is called t-summation polynomial of the elliptic curve E if it satisfies the following property. For each $(\overline{x_1}, \cdots, \overline{x_t}) \in \overline{\mathbb{K}}^t$ one has that $f_t(\overline{x_1}, \cdots, \overline{x_t}) = 0$ if and only if there exist $P_i = [\overline{x}_i, y_i, 1] \in E$ for each i, such that $P_1 \oplus \cdots \oplus P_t = \mathcal{O}$.

A polynomial $f_t(y_1, \cdots, y_t) \in \mathbb{K}[y_1, \cdots, y_t]$ is called t-summation polynomial of the twisted Edwards curve $E_{a,d}$ if it satisfies the following property. For each $(\overline{y_1}, \cdots, \overline{y_t}) \in \overline{\mathbb{K}}^t$ one has that $f_t(\overline{y_1}, \cdots, \overline{y_t}) = 0$ if and only if there exist $P_i = [x_i, \overline{y}_i, 1] \in E_{a,d}$ for each i, such that $P_1 \oplus \cdots \oplus P_t = \mathcal{O}$.

In [67, Theorem 1], Semaev takes an elliptic curve E in short Weierstrass form. He gives a definition for f_2 and f_3, as well as a recursive construction for f_t when $t \geq 4$. The recursive construction is based on the Sylvester resultant between two polynomials. The definitions can be easily adapted to the case of twisted Edwards curves, as the authors did in [40]. In the following theorem, we give these constructions both for elliptic and twisted Edwards curves, as well as the basic properties of summation polynomials.

Theorem 37 (Recursive construction and properties of summation polynomials)**.** *Let $E : y^2z = x^3 + Axz^2 + Bz^3$ be an elliptic curve in short Weierstrass form defined over $\mathbb{K}$. The t-th summation polynomial f_t of the elliptic curve E can be defined by*

$$\begin{cases} f_2(x_1, x_2) = x_1 - x_2, \\ \\ f_3(x_1, x_2, x_3) = (x_1 - x_2)^2 x_3^2 - 2((x_1 + x_2)(x_1x_2 + A) + 2B)x_3 + ((x_1x_2 - A)^2 - 4B(x_1 + x_2)) \\ \\ f_t(x_1, \cdots, x_t) = \mathrm{res}_x(f_{t-k}(x_1, \cdots, x_{t-k-1}, x), f_{k+2}(x_{t-k}, \cdots, x_t, x)) \text{ for } t \geq 4 \text{ and } 1 \leq k \leq t-3 \end{cases},$$

where $res_x(f_i, f_j)$ denotes the resultant of the polynomials f_i and f_j with respect to the variable x.

Let $E_{a,d} : ax^2z^2 + y^2z^2 = z^4 + dx^2y^2$ be a twisted Edwards curve defined over $\mathbb{K}$. The t-th summation polynomial f_t of the twisted Edwards curve $E_{a,d}$ can be defined by

$$\begin{cases} f_2(y_1,y_2) = y_1 - y_2 \\ \\ f_3(y_1,y_2,y_3) = (y_1^2y_2^2 - y_1^2 - y_2^2 + ad^{-1})y_3^2 + 2(d-a)d^{-1}y_1y_2y_3 + ad^{-1}(y_1^2+y_2^2-1) - y_1^2y_2^2 \\ \\ f_t(y_1,\ldots,y_t) = \mathrm{res}_y(f_{t-k}(y_1,\ldots,y_{t-k-1},y), f_{k+2}(y_{t-k},\ldots,y_t,y)) \text{ for } t \geq 4 \text{ and } 1 \leq k \leq t-3 \end{cases},$$

The t-th summation polynomial f_t (of an elliptic curve E or of a twisted Edwards curve) is absolutely irreducible and it has degree 2^{t-2} in each variable. For $t > 2$, the t-th summation polynomial is symmetric.

Example 38. Let $\overline{Ed} : 3x^2z^2+y^2z^2 = z^4+x^2y^2$ be the twisted Edwards curve of Example 12, defined over the field $\mathbb{F}_{13}$. The 3-rd summation polynomial of $\overline{Ed}$ is $f_3(y_1,y_2,y_3) = (y_1^2y_2^2 - y_1^2 - y_2^2 + 3)y_3^2 + 9y_1y_2y_3 + 3(y_1^2+y_2^2-1) - y_1^2y_2^2$. We have that $f_3(0,0,-1) = 0$ and $P_1 = P_2 = [3,0,1]$, $P_3 = [0,-1,1] \in \overline{Ed}$ are such that $P_1 \oplus P_2 \oplus P_3 = \mathcal{O}$.

Constructive and destructive applications of summation polynomials

By definition, finding a zero of the t-th summation polynomial of an elliptic or twisted Edwards curve is equivalent to finding a vanishing relation $P_1 \oplus \cdots \oplus P_t = \mathcal{O}$ between t points $P_1, \cdots, P_t$ on the curve. Each point P_i is determined up to sign. Notice that, using coordinates $(x_1, \cdots, x_t, y_1, \cdots, y_t)$ and the definition of point addition of the curve E, one can define a polynomial $F_t(x_1, \cdots, x_t, y_1, \cdots, y_t) \in \mathbb{K}[x_1, \cdots, x_t, y_1, \cdots, y_t]$ such that the following property holds. We have that $F_t(\overline{x}_1, \cdots, \overline{x}_t, \overline{y}_1, \cdots, \overline{y}_t) = 0$ for $(\overline{x}_1, \cdots, \overline{x}_t, \overline{y}_1, \cdots, \overline{y}_t) \in \overline{\mathbb{K}}^{2t}$ if and only if $P_i = [\overline{x}_i, \overline{y}_i, 1] \in E$ for all i and $P_1 \oplus \cdots \oplus P_t = \mathcal{O}$. In this case, the points P_i are uniquely determined. However, F_t has twice the variables of f_t. The polynomial F_t will be much more difficult to compute and to manage. If we use f_t to find relations between points, we cannot distinguish any more between a point and its inverse. Nevertheless, in practice, this little ambiguity makes the difference between manageable and practically unmanageable polynomials, at least for little n. Actually, we point out that also f_t becomes huge even for little n. The complexity of computing the t-th Semaev polynomial is $\mathcal{O}(e^{t^2})$ ([30, Proposition 2.3]), and the computation becomes practically unmanageable for $t > 6$. In fact, the largest summation polynomial ever computed is the 6-th summation polynomial (see [41]).

From the discussion above, it follows that, for little t, summation polynomials are interesting cryptographic tools. They can play an important role in constructive aspects regarding the efficiency of cryptosystems, as well as in destructive aspects, regarding the study of DLP attacks. Concerning the first aspect, summation polynomials are used in [47] to give an optimal and efficient representation for trace-zero subgroups of elliptic curves. Moreover, we use summation polynomial in Chapter 2 for the same purpose, in the case of twisted Edwards curves. As regard the second aspect, in Section 1.5 we explain how summation polynomials are applied to Gaudry's index calculus method for DLP attacks, on elliptic groups $E(\mathbb{F}_{q^n})$ and trace-zero subgroups of elliptic curves (see [46],[30],[31],[48]).

Summation polynomials have been of fundamental importance throughout our work. In fact, we draw inspiration from them to define and compute generalized summation polynomials. We introduce these polynomials in Chapter 3. We use generalized summation polynomials to develop a scalar multiplication algorithm for trace-zero elements (see Chapter 4), as well as to build an index calculus variant for trace-zero subgroups (see Chapter 5).

The scheme below sums up the relevant features of summation polynomials we dealt with in this subsection.

Scheme 5.

Summation polynomials of elliptic and twisted Edwards curves.

Aim.

- Find vanishing relations $P_1 \oplus \cdots P_t = \mathcal{O}$ on the curve (up to sign).

Constructive cryptographic applications.

- Optimal representation for trace-zero elements ([47] and Chapter 2).

Destructive cryptographic applications.

- Index calculus for DLP attacks on $E(\mathbb{F}_{q^n})$ and on trace-zero subgroups (Section 1.5).

Generalization.

- Generalized summation polynomials (Chapter 3) and their applications (Chapter 4 and 5).

1.4 Trace-zero subgroups

Elliptic and twisted Edwards curves are the primary and most common application of algebraic geometry to cryptography. Nevertheless, a deeper insight in this world allows to find new and suitable alternatives to construct DLP cryptosystems. This was the aim of Frey in his paper "Applications of arithmetical geometry to cryptographic constructions " ([43], 1999). In the mentioned work, the author introduces some classes of admissible abelian varieties. Admissible abelian varieties are abelian varieties that satisfy some necessary conditions for a safe and efficient cryptographic setting. Among them, trace-zero subgroups of elliptic curves were first proposed for applications to cryptography. In Subsection 1.4.1, we briefly recall the basic notions about abelian varieties. We introduce trace-zero subgroups in Subsection 1.4.2.

1.4.1 Beyond elliptic curve cryptography: admissible abelian varieties

We shortly introduce abelian varieties starting from the notions of Section 1.2. Let $\mathbb{K}$ be field. For $n \in \mathbb{Z}_{\geq 2}$, the affine space of dimension n over $\mathbb{K}$, denoted by $\mathbb{A}^n(\overline{\mathbb{K}})$, is the natural generalization of $\mathbb{A}^2$ from 2 to n. Similarly, the projective space of dimension n over $\mathbb{K}$, denoted by $\mathbb{P}^n(\overline{\mathbb{K}})$, is the natural generalization of $\mathbb{P}^2$ from 2 to n. We write $\mathbb{A}^n$ and $\mathbb{P}^n$ for $\mathbb{A}^n(\mathbb{K})$ and $\mathbb{P}^n(\mathbb{K})$ respectively, when there is no ambiguity about the field of definition. We have that $\mathbb{A}^n$ and $\mathbb{P}^n$ are related via the map $\Phi_{x_{n+1}}$, which is the analogous of Φ_z in the case $n = 2$.
One defines affine/projective algebraic sets, and affine/projective varieties, as generalizations of affine/projective curves, and affine/projective absolutely irreducible curves respectively. More precisely, an affine algebraic set $V_a \subseteq \mathbb{A}^n$ defined over $\mathbb{K}$ is a subset of form:

$$V_a = \{P \in \mathbb{A}^n : f_1(P) = 0, \cdots, f_t(P) = 0\},$$

where $f_i(x_1, \cdots, x_n) \in \mathbb{K}[x_1, \cdots, x_n]$ for $i \in \{1, \cdots, t\}$. Moreover, for each field extension $\mathbb{K} \subseteq \mathbb{L} \subseteq \overline{\mathbb{K}}$, the set of $\mathbb{L}$-rational points of V_a is

$$V_a(\mathbb{L}) = \{(x_1, \cdots, x_n) \in V_a : x_i \in \mathbb{L} \text{ for } i \in \{1, \cdots, n\}\}.$$

When the affine algebraic set V_a defined over $\mathbb{K}$ is such that the corresponding ideal $(f_1, \cdots, f_t) \subseteq \overline{\mathbb{K}}[x_1, \cdots, x_n]$ is a prime ideal, then V_a is called affine variety defined over $\mathbb{K}$.

A projective algebraic set $V_p \subseteq \mathbb{P}^n$ defined over $\mathbb{K}$ is a subset of form:

$$V_p = \{P \in \mathbb{P}^n : f_1(P) = 0, \cdots, f_t(P) = 0\},$$

where $f_i(x_1, \cdots, x_{n+1}) \in \mathbb{K}[x_1, \cdots, x_{n+1}]$ are homogeneous polynomials. For each field extension $\mathbb{K} \subseteq \mathbb{L} \subseteq \overline{\mathbb{K}}$, the set of $\mathbb{L}$-rational points of V_p is

$$V_p(\mathbb{L}) = \{[x_1 \cdots, x_{n+1}] \in V_p : \exists\ i \in \{1, \cdots, n+1\} \text{ such that } x_k/x_i \in \mathbb{L} \text{ for all } k \in \{1, \cdots, n+1\}\}.$$

The projective algebraic set V_p defined over $\mathbb{K}$ is called projective variety defined over $\mathbb{K}$ if the corresponding ideal $(f_1, \cdots, f_t) \subseteq \overline{\mathbb{K}}[x_1, \cdots, x_{n+1}]$ is a prime ideal.

Hence affine/projective curves defined over $\mathbb{K}$ are affine/projective algebraic sets defined over $\mathbb{K}$, for $n = 2$ and $t = 1$. Moreover, affine/projective absolutely irreducible curves defined over $\mathbb{K}$ are affine/projective varieties defined over $\mathbb{K}$, for $n = 2$ and $t = 1$. As we did in the special case $n = 2$, $t = 1$, we define the projective closure $\overline{V_a}$ of V_a. Moreover, we define the dehomogenization $(V_p)^*_{x_{n+1}}$ of V_p with respect to the last variable. Let V_a be an affine variety defined over $\mathbb{K}$. Let V_p be a projective variety defined over $\mathbb{K}$. Take a field extension $\mathbb{K} \subseteq \mathbb{L} \subseteq \overline{\mathbb{K}}$. Again generalizing from the case $n = 2$, $t = 1$, we define the field of $\mathbb{L}$-rational functions $\mathbb{K}(V_a)$ of V_a and $\mathbb{K}(V_p) \cong \mathbb{K}((V_p)^*_{x_{n+1}})$ of V_p. In the same way, we define rational maps between affine or projective varieties.

Remark 39. Algebraic sets of $\mathbb{A}^n$ (resp. $\mathbb{P}^n$) are the closed sets of a topology on $\mathbb{A}^n$ (resp. $\mathbb{P}^n$), called the Zarisky topology. Sometimes, one is interested in geometric properties that hold true in a nonempty Zarisky open set of $\mathbb{A}^n$ or $\mathbb{P}^n$. Notice that a nonempty Zarisky open set of $\mathbb{A}^n$ (resp. $\mathbb{P}^n$) is dense in $\mathbb{A}^n$ (resp. $\mathbb{P}^n$). This means that its closure is the whole space $\mathbb{A}^n$ (resp. $\mathbb{P}^n$). Intuitively, if a property holds in a nonempty Zarisky open set, then the property is verified almost always. The given nonempty open set is the general case. The complement set is the set of exceptional cases.

The dimension of a projective variety V defined over $\mathbb{K}$ is denoted by $\dim(V)$. It is the minimal d such that $\mathbb{K}(V)$ is algebraic over $\mathbb{K}(x_1, \cdots, x_d)$.

Example 40. The dimension of an absolutely irreducible projective curve $C : F(x, y, z) = 0$ defined over $\mathbb{K}$ is $\dim(C) = 1$. In fact, $\mathbb{K}(C) \cong \mathbb{K}(C^*_z) = \mathbb{K}(\overline{x}, \overline{y})$, where $\overline{x}, \overline{y}$ are x, y mod $(f(x, y) = F(x, y, 1))$. We have that $\mathbb{K}(\overline{x}, \overline{y}) = \mathbb{K}(x)[y]/(f(x, y))$. Therefore, $\mathbb{K}(C)$ is algebraic over $\mathbb{K}(x)$, and the dimension of C is $\dim(C) = 1$.

An abelian variety defined over $\mathbb{K}$ is a variety V defined over $\mathbb{K}$, endowed with a commutative group operation

$$+ : V \times V \longrightarrow V,$$

that is a rational map between the varieties $V \times V$ and V. Let V be an abelian variety defined over $\mathbb{K}$. Take a field extension $\mathbb{K} \subseteq \mathbb{L} \subseteq \overline{\mathbb{K}}$. It follows from the definition of abelian variety that $(V(\mathbb{L}), +)$ is an abelian group.

Example 41 (Some abelian varieties)**.** An elliptic curve defined over $\mathbb{K}$ is a projective abelian variety of dimension 1 defined over $\mathbb{K}$, with the point addition law $\oplus$ given in Section 1.3.1.

Let $E_{a,d}$ be a twisted Edwards curve defined over $\mathbb{K}$. Its affine dehomogenization $(E_{a,d})^*_z$ is an affine abelian variety of dimension 1 defined over $\mathbb{K}$, with the point addition law $\hat{\oplus}$ given in Section 1.3.2.

Take an absolutely irreducible projective curve of the form

$$C : y^2 z^{2g-1} = f(x, z),$$

where $g \in \mathbb{Z}_{\geq 1}$, $f \in \mathbb{K}[x,z]$ is a homogeneous polynomial of degree $2g+1$, such that $f(x,1)$ has no multiple roots. One has that the genus of the curve C is g. A curve of this form is called hyperelliptic curve of genus g defined over $\mathbb{K}$. Notice that a hyperelliptic curve of genus 1 is an elliptic curve. One can show that the degree zero Picard group $\mathrm{Pic}^0(C)$ of the curve C has a structure of an abelian variety of dimension g defined over $\mathbb{K}$. Moreover, for the field extension $\mathbb{K} \subseteq \mathbb{L} \subseteq \overline{\mathbb{K}}$, the group $\mathrm{Pic}^0(C)(\mathbb{L})$ of $\mathbb{L}$-rational divisor classes that we defined in (1.3) corresponds to the group of $\mathbb{L}$-rational points of the variety $\mathrm{Pic}^0(C)$.

In the following example, we give a procedure to construct an abelian variety of dimension $n \in \mathbb{Z}_{>0}$ defined over a finite field $\mathbb{F}_q$, starting from an elliptic curve E defined over $\mathbb{F}_q$ and the group $E(\mathbb{F}_{q^n})$ of $\mathbb{F}_{q^n}$-rational points of E. Such method is called Weil restriction of scalars. We refer to [6, Chapter 7] and to [43, Section 3.2].

Example 42 (Weil restriction of scalars)**.** Let $E : F(x,y,z) = 0$ be an elliptic curve defined over a finite field $\mathbb{F}_q$. For $n \in \mathbb{Z}_{>0}$, take the field extension $\mathbb{F}_q \subseteq \mathbb{F}_{q^n}$. Choose a basis $\{\alpha_1, \cdots, \alpha_n\}$ of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$. For $\overline{x}, \overline{y} \in \mathbb{F}_{q^n}$, one can write $\overline{x} = \overline{x}_1\alpha_1 + \cdots + \overline{x}_n\alpha_n$, $\overline{y} = \overline{y}_1\alpha_1 + \cdots + \overline{y}_n\alpha_n$, for some $\overline{x}_1, \cdots, \overline{x}_n, \overline{y}_1, \cdots, \overline{y}_n \in \mathbb{F}_q$. So we have that an affine point $[\overline{x}, \overline{y}, 1]$ belongs to $E(\mathbb{F}_{q^n})$ if and only if $0 = F(\overline{x}, \overline{y}, 1) = f(\overline{x}, \overline{y}) = f(\overline{x}_1\alpha_1 + \cdots + \overline{x}_n\alpha_n, \overline{y}_1\alpha_1 + \cdots + \overline{y}_n\alpha_n) = f_1(\overline{x}_1, \cdots, \overline{x}_n, \overline{y}_1, \cdots, \overline{y}_n)\alpha_1 + \cdots + f_n(\overline{x}_1, \cdots, \overline{x}_n, \overline{y}_1, \cdots, \overline{y}_n)\alpha_n$, for some $f_1, \cdots, f_n \in \mathbb{F}_q[x_1, \cdots, x_n, y_1, \cdots, y_n]$. This is equivalent to saying $f_i(\overline{x}_1, \cdots, \overline{x}_n, \overline{y}_1, \cdots, \overline{y}_n) = 0$ for $i \in \{1, \cdots, n\}$. We take the following projective algebraic set defined over $\mathbb{F}_q$:

$$W_E = \{P \in \mathbb{P}^{2n} : f_i^h(P) = 0 \text{ for } i \in \{1, \cdots, n\}\}.$$

One can show that W_E is a n-dimensional abelian variety defined over $\mathbb{F}_q$. Moreover, we have that the group $E(\mathbb{F}_{q^n})$ of $\mathbb{F}_{q^n}$-rational points of E can be identified with the group $W_E(\mathbb{F}_q)$ of $\mathbb{F}_q$-rational points of W_E. We say that the n-dimensional abelian variety W_E is obtained from E via Weil restriction of scalars. Similarly, given a twisted Edwards curve $E_{a,d}$ defined over $\mathbb{F}_q$, we construct the n-dimensional affine abelian variety $W_{(E_{a,d})_z^*}$ defined over $\mathbb{F}_q$.

For cryptographic applications, one can take abelian varieties V defined over finite fields $\mathbb{F}_q$ and the groups $V(\mathbb{F}_q)$ of $\mathbb{F}_q$-rational points of V. An abelian variety V is called admissible abelian variety (see [43]), if it could be a good candidate to set a cryptosystem. This means that $V(\mathbb{F}_q)$ satisfies the conditions of security and efficiency that we recall in Section 1.1.

Elliptic and twisted Edwards curves are abelian varieties of dimension 1. If one takes abelian varieties of higher dimension, the formulas for the group law could be too complicated to be managed efficiently in a cryptosystem. In such situation, the abelian variety is called unmanageable. It is a non-admissible abelian variety. This is the case of the Picard group of hyperelliptic curves of high genus. On the other hand, a growth in the dimension and the more complex structure of the group under consideration could allow the use of new geometric tools, that one does not have in dimension 1. Such tools could have a positive effect from the point of view of efficiency. In fact, they could be used to speed up the computation algorithms. This is the case of groups $E(\mathbb{F}_{q^n})$ of $\mathbb{F}_{q^n}$-rational points of elliptic curves E defined over $\mathbb{F}_q$ (see Example 42). We will see in the sequel that, in such groups, one can use the Frobenius endomorphism of the curve to speed up the computation. Nevertheless, the additional structure on the variety could also represent a threat for the security of the cryptosystem. It might be exploited to perform specific DLP attacks, such as the index calculus attack that we describe in Section 1.5.

Among the families of admissible abelian varieties, Frey suggests, in [43], the use of Picard groups of hyperelliptic curves of small genus, as well as the use of trace-zero

subgroups of elliptic and hyperelliptic curves. The first kind of abelian varieties had been already proposed by Koblitz in [53]. On the other hand, Frey started the study of trace-zero subgroups in cryptography. Trace-zero subgroups of elliptic and twisted Edwards curves are the main objects of this thesis. We introduce them in the next subsection.

1.4.2 Trace-zero subgroups for cryptography

Let $\mathbb{F}_q$ be a finite field of characteristic different from 2, 3. Denote by $\overline{\mathbb{F}}_q$ the algebraic closure of $\mathbb{F}_q$. Let $E : f(x, y, z) = 0$ be an elliptic curve defined over $\mathbb{F}_q$. Let $E_{a,d} : f(x, y, z) = 0$ be a twisted Edwards curve defined over $\mathbb{F}_q$. We denote again by E the affine dehomogenization of $E_{a,d}$ with respect to the variable z. Denote by $\oplus$ the operation of point addition on E. Denote by $\mathcal{O}$ the neutral element of the operation. Let P be a point of E. Denote by $-P$ the inverse of P with respect to $\oplus$. Take a field extension $\mathbb{F}_q \subseteq \mathbb{L} \subseteq \overline{\mathbb{F}}_q$. We recall from Section 1.3 that $(E(\mathbb{L}), \oplus)$ is an abelian group. Let φ be the Frobenius endomorphism of the elliptic curve E:

$$\varphi : E \longrightarrow E,\ [x, y, z] \mapsto [x^q, y^q, z^q].$$

We denote again by φ the Frobenius endomorphism of the twisted Edwards curve $E_{a,d}$:

$$\varphi : E(\overline{\mathbb{F}}_q) \longrightarrow E(\overline{\mathbb{F}}_q),\ (x, y) \mapsto (x^q, y^q).$$

Notice that, in the case of twisted Edwards curves, the Frobenius endomorphism can be naturally extended to a map

$$\overline{\varphi} : E_{a,d} \longrightarrow E_{a,d},\ [x, y, z] \mapsto [x^q, y^q, z^q].$$

Nevertheless, since $E_{a,d}$ is not a group, we restrict to the group $E(\overline{\mathbb{F}}_q)$ of $\overline{\mathbb{F}}_q$-rational affine points of $E_{a,d}$, to define the group endomorphism φ.

Let $\mathbb{F}_q \subseteq \mathbb{F}_{q^n}$ be a field extension of odd prime degree n. We take the trace endomorphism of $E(\mathbb{F}_{q^n})$:

$$\mathrm{Tr} : E(\mathbb{F}_{q^n}) \longrightarrow E(\mathbb{F}_q),\ P \mapsto P \oplus \varphi(P) \oplus \cdots \oplus \varphi^{n-1}(P).$$

Notice that, for each $P \in E(\mathbb{F}_{q^n})$, one has that $\varphi(\mathrm{Tr}(P)) = \varphi(P) \oplus \cdots \oplus \varphi^{n-1}(P) \oplus \varphi^n(P) = P \oplus \cdots \oplus \varphi(P) \cdots \oplus \varphi^{n-1}(P) = \mathrm{Tr}(P)$. This means that $\mathrm{Tr}(P) \in E(\mathbb{F}_q)$ for all $P \in E(\mathbb{F}_{q^n})$. Hence, the map Tr is well defined.

Definition 43 (Trace-zero subgroups of elliptic/twisted Edwards curves)**.** The trace-zero subgroup T_n of $E(\mathbb{F}_{q^n})$ is the kernel of Tr, that is:

$$T_n = \{P \in E(\mathbb{F}_{q^n}) :\ P \oplus \varphi(P) \oplus \cdots \oplus \varphi^{n-1}(P) = \mathcal{O}\}.$$

In Example 42, we saw that we can identify $E(\mathbb{F}_{q^n})$ with the group of $\mathbb{F}_q$-rational points of the n-dimensional abelian variety W_E, via the process of Weil restriction of scalars. We have that T_n can be identified with the group of $\mathbb{F}_q$-rational points of a $(n-1)$-dimensional subvariety of W_E (see [6, Section 7.4.2] and [43, Section 3.2]). This subvariety is called trace-zero variety. We denote it by $\mathcal{T}_n$.

Remark 44. Trace-zero subgroups can be defined, in a more general setting, for the Picard group of hyperelliptic curves of genus g. For example, for genus $g = 2$, they have been studied by T. Lange in [55]. In this thesis, we only deal with trace-zero subgroups of elliptic and twisted Edwards curves. Therefore, we decided to give the definition of trace-zero subgroups in this specific case.

Trace-zero subgroups have been first proposed for cryptographic applications by Frey in [43]. They turn out to be interesting in this context, as they provide a suitable combination of good security, optimal data storage and efficient arithmetic. We now analyze these aspects for $T_n \subseteq E(\mathbb{F}_{q^n})$. We make comparisons with the groups of base field-rational points of elliptic curves and twisted Edwards curves. From such analysis, it follows that the use of trace-zero subgroups in cryptography can be a valid alternative to the standard use of elliptic curves.

Security and optimal data storage in trace-zero subgroups

Suppose that we have a DLP cryptosystem on $E(\mathbb{F}_q)$, where q is a prime number and $E(\mathbb{F}_q)$ is cyclic of prime order. We recall from Section 1.3 that, in such groups, the best algorithm to solve the DLP is the Pollard's rho algorithm. Its complexity is $O(q^{\frac{1}{2}})$. Suppose that we want to switch to a more secure cryptosystem. In order to do this, one can simply take $E'(\mathbb{F}_{q'})$, for some elliptic curve E' defined over a finite field $\mathbb{F}_{q'}$, where q' is a prime number and $q' > q$. As before, we choose $E'(\mathbb{F}_{q'})$ cyclic of prime order. Alternatively, we can take the group $E(\mathbb{F}_{q^n})$ of $\mathbb{F}_{q^n}$-rational points of E, where $n > 1$. Suppose that n is an odd prime. The following result shows that it is not necessary to take the whole group $E(\mathbb{F}_{q^n})$. As a matter of fact, we obtain the same increase in the level of security if we restrict to the subgroup $T_n \subseteq E(\mathbb{F}_{q^n})$.

Proposition 45. *We have the exact sequence:*

$$0 \longrightarrow T_n \hookrightarrow E(\mathbb{F}_{q^n}) \xrightarrow{Tr} E(\mathbb{F}_q) \longrightarrow 0.$$

Proof. We prove the result for the case in which $E(\mathbb{F}_q)$ is cyclic of prime order p, with p much larger than n. Notice that this is the case in cryptographic applications. In this case, one can take n^{-1} the inverse of n modulo p. Hence, for $Q \in E(\mathbb{F}_q)$, one obtains $\mathrm{Tr}(n^{-1}Q) = Q$. This implies the surjectivity of Tr and the exactness of the sequence. □

As a consequence of the previous proposition, one has that solving a DLP in $E(\mathbb{F}_{q^n})$ reduces to solving a DLP in T_n and a DLP in $E(\mathbb{F}_q)$. Therefore, we can take the smaller group $T_n \subseteq E(\mathbb{F}_{q^n})$, rather than the whole $E(\mathbb{F}_{q^n})$, without losing security. This advantage of the trace-zero subgroup is also captured by the security parameter for pairing based cryptography. In fact, in [65] and [66], it is proved that such parameter in T_n is higher than that of $E(\mathbb{F}_{q^n})$ by a factor $n/(n-1)$, when E is a supersingular elliptic curve.

Optimal representations for trace-zero elements. An advantage of working in T_n rather than in the entire group $E(\mathbb{F}_{q^n})$ is that we can represent trace-zero elements with shorter bit-strings. In this way, we obtain optimal data storage for the same level of security, as a consequence of Proposition 45 and the subsequent discussion. Hence, it is of great importance to know and use an optimal representation for trace-zero elements. We refer to Definition 7 of optimal representation. By the Hasse-Weil theorem, one has that $|E(\mathbb{F}_{q^n})| \in O(q^n)$. Moreover, by Proposition 45, we have $|T_n| = |E(\mathbb{F}_{q^n})|/|E(\mathbb{F}_q)|$. This implies that $|T_n| \in O(q^{n-1})$. Therefore, we need n coordinates of $\mathbb{F}_q$ for an optimal representation of elements of $E(\mathbb{F}_{q^n})$. On the other hand, we need only $n-1$ coordinates of $\mathbb{F}_q$ for an optimal representation of elements of T_n. In (1.10), we give the optimal representation $\mathcal{R}_1$ for the family of groups on elliptic curves $\mathcal{G}_1 = (E(\mathbb{F}_q))_{q,E}$. Furthermore, in (1.11), we give the analogous optimal representation $\mathcal{R}'_1$ for the family of groups on twisted Edwards curves $\mathcal{G}'_1 = (E(\mathbb{F}_q))_{q,E_{a,d}}$. If we combine the representation $\mathcal{R}_1$ with the

process of Weil restriction of scalars (see Example 42), we obtain the optimal representation $\mathcal{R}_n = (\mathcal{R}_{n,q,E})_{q,E}$ for the family of groups on elliptic curves $\mathcal{G}_n = (E(\mathbb{F}_{q^n}))_{q,E}$. This optimal representation is given by the maps

$$\mathcal{R}_{n,q,E} : E(\mathbb{F}_{q^n}) \setminus \{\mathcal{O}\} \longrightarrow \mathbb{F}_q^n,\ [x, y, 1] \mapsto (x_1, \cdots, x_n). \tag{1.12}$$

We refer to Example 42 for notation. Similarly, from $\mathcal{R}'_1$, and using Weil restriction of scalars, we have the optimal representation $\mathcal{R}'_n = (\mathcal{R}'_{n,q,E_{a,d}})_{q,E_{a,d}}$ for the family of groups on twisted Edwards curves $\mathcal{G}'_n = (E(\mathbb{F}_{q^n}))_{q,E_{a,d}}$. The optimal representation is given by the maps

$$\mathcal{R}'_{n,q,E_{a,d}} : E(\mathbb{F}_{q^n}) \longrightarrow \mathbb{F}_q^n,\ (x, y) \mapsto (y_1, \cdots, y_n). \tag{1.13}$$

Notice that, for each $T_n \subseteq E(\mathbb{F}_{q^n})$, one can restrict the map $\mathcal{R}_{n,q,E}$ (or the map $\mathcal{R}'_{n,q,E_{a,d}}$, in the case of twisted Edwards curves) to the subgroup T_n. In such a way, we obtain a family of representations for the family of groups $(T_n \subseteq E(\mathbb{F}_{q^n}))_{q,E}$. Nevertheless, this induced family of representations is not an optimal representation for the family of trace-zero subgroups. In fact, $|T_n| \in \mathcal{O}(q^{n-1})$. Hence, an optimal representation for the trace-zero family is given by maps of the type

$$\hat{R} : T_n \longrightarrow \mathbb{F}_q^{n-1} \times \mathbb{F}_2^{k_{T_n}},$$

with the properties of Definition 7. Moreover, efficient compression and decompression algorithms for $\hat{R}$ are needed. If we can efficiently perform compression and decompression, then the given representation is actually relevant for cryptographic applications. Optimal representations for trace-zero subgroups of elliptic curves in short Weierstrass form have been proposed by Naumann in [62] for $n = 3$, Weimerskirch in [75] for $n = 5$, Silverberg in [69] and Cesena in [27] for $n = 3, 5$, Gorla-Massierer in [47] and in [49] for any odd prime n. In all the mentioned works, efficient compression and decompression algorithms are given. In Chapter 2 of this thesis, we give two optimal representations for trace-zero subgroups of twisted-Edwards curves. We follow and adapt ideas from [47] and [49] for elliptic curves in short Weierstrass form. We refer to the mentioned chapter for the description and analysis of our work on the subject.

Optimal representations and efficient arithmetic

The use of an optimal representation for trace-zero elements has to be integrated with efficient performances of the arithmetic in T_n. For practical applications as the Diffie-Hellman key exchange, one is especially interested in fast performance of scalar multiplication. Therefore, the aim is combining fast performance of scalar multiplication in T_n with the use of an optimal representation for trace-zero points. There are two possible ways to achieve this goal. The first way is to perform the operation in the whole group $E(\mathbb{F}_{q^n})$ and to use compression/decompression algorithms to pass from the usual coordinates of $E(\mathbb{F}_{q^n})$ to the optimal coordinates of the trace-zero subgroup, and vice versa respectively. We call this approach non-compressed scalar multiplication. The second way is to perform the operation directly in the optimal compressed coordinates of the trace-zero subgroup. In Chapter 4 of this thesis, we give an algorithm that accomplishes this task for $n = 3$. The algorithm uses the compressed coordinates of the optimal representation proposed in [49]. Such algorithm is the first one to carry out scalar multiplication in trace-zero subgroups, performing the computation directly in optimal coordinates.

We now briefly describe the first approach to scalar multiplication in T_n, that is non-compressed scalar multiplication. We refer to Chapter 4 for the description and analysis of our original algorithm.

Non-compressed scalar multiplication in T_n and Frobenius reduction. Non-com-
pressed scalar multiplication in T_n is an efficient procedure. As a matter of fact, in all works dealing with optimal representations for T_n ([62], [75], [69], [27], [47], [49]), efficient compression/decompression algorithms are given. Moreover, fast algorithms to perform operations in $E(\mathbb{F}_{q^n})$ are well known. Furthermore, when we work in $E(\mathbb{F}_{q^n})$, we can exploit the Frobenius endomorphism φ of the curve to speed up scalar multiplication, as explained in [6, Section 15.1 and Section 15.2]. We call such strategy Frobenius reduction. Frobenius reduction can be applied, in general, to the group $\mathrm{Pic}^0(C)(\mathbb{F}_{q^r})$ of $\mathbb{F}_{q^r}$-rational divisor classes of a hyperelliptic curve C defined over $\mathbb{F}_q$, whenever $r > 1$ (see[6, Section 15.1]). It was first proposed by Koblitz in [54] for special elliptic curves. The idea is to take the characteristic polynomial χ_φ of the Frobenius endomorphism φ of E, as defined in (1.9). Notice that this definition is analogous in the case of twisted Edwards curves. For a root τ of χ_φ, one has that $\varphi(P) = \tau P$ for all $P \in E$. Suppose that we want to compute the scalar product mP of P. One can write the scalar m as

$$m = \sum m_i \tau^i, \tag{1.14}$$

in such a way that the m_i's are small compared to m. The sum (1.14) is called τ-expansion of m. Then, one can compute $mP = \sum m_i \varphi^i(P)$. Namely, we split the computation of the scalar product by m in the computations of scalar products of smaller size, and some few additions. The computations of the small scalar products can be done in parallel. Moreover, one can implement the arithmetic in the finite field extensions $\mathbb{F}_q \subseteq \mathbb{F}_{q^n}$, in such a way that applying the Frobenius endomorphism to a point is costless. In order to do this, we can for example take a normal basis of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$. Therefore, the result of Frobenius reduction is a speeding up of the whole multiplication process. For non-compressed scalar multiplication in trace-zero subgroups, the strategy enjoys the benefit of the extra property of the Frobenius endomorphism restricted to the trace-zero subgroup. The property is

$$1 + \varphi + \cdots \varphi^{n-1} = 0. \tag{1.15}$$

Using this property, the operation of scalar multiplication can be further sped up. Hence, computation in T_n is faster than in the whole group $E(\mathbb{F}_{q^n})$: see [6, Section 15.3], [4], [62], [75].

We now give more details on Frobenius reduction in T_n. We refer to [6, Section 15.3]. Suppose that T_n is cyclic of prime order p, that is the case for cryptographic applications. One has that there is an element $s \in \mathbb{Z}_p$ such that $\varphi(P) = sP$ for all $P \in T_n$. If we take the Frobenius endomorphism restricted to the trace-zero subgroup, we have that both equations (1.9) and (1.15) hold true. Therefore, one can easily compute s from the derived equalities $s^2 - as + q = 0 \mod p$ and $1 + s + \cdots s^{n-1} = 0 \mod p$. Now we write the scalar m as $m = \sum_{i=0}^{n-2} m_i s^i$, where the m_i's are in the order of s. Then, for $P \in T_n$, we compute the scalar product $mP = \sum m_i \varphi^i(P)$. The authors of [5] did some practical experiments for $n = 3, 5$. They took trace-zero subgroups T_3 and T_5 of elliptic curves defined over prime fields of odd characteristic, of cryptographic size 2^{160}, 2^{192} and 2^{256}. They performed scalar multiplication in such subgroups, using Frobenius reduction. Then, they performed the same operation in groups of base field-rational points of elliptic curves, of the same size. It turned out that scalar multiplication in the trace-zero subgroups was up to 30% faster than the same operation in the standard elliptic groups. In pairing-based cryptography, one can use analogous strategies involving the Frobenius endomorphism, to speed up the computation of the Miller function for the Tate pairing with trace-zero subgroups (see [27]). Moreover, Frobenius reduction can be adapted and applied to optimize our scalar

multiplication algorithm in compressed trace-zero coordinates, as we will see in Chapter 4.

Conclusion and comparison with standard elliptic groups

In the previous paragraphs, we described security and efficiency aspects for trace-zero subgroups $T_n \subseteq E(\mathbb{F}_{q^n})$. From the above discussion, it follows that such subgroups could be a valid alternative, for cryptographic applications, to classical elliptic groups of the type $E'(\mathbb{F}_{q'})$, where q' is a prime number. Nevertheless, we remark that, in T_n, one can apply other DLP attacks than the generic Pollard's rho. This is in contrast with the case of elliptic groups $E'(\mathbb{F}_{q'})$, where the best DLP attack is the generic Pollard's rho, of complexity $O((q')^{\frac{1}{2}})$. In fact, trace-zero subgroups T_n over $\mathbb{F}_q$ can be seen as groups of $\mathbb{F}_q$-rational points of abelian varieties of dimension $n-1$, via the process of Weil restriction of scalars. We have already mentioned that, when the dimension of the abelian variety is greater than one, we have more geometric structure on the groups under consideration. This could be an advantage for efficient arithmetic: a bright example is the application of Frobenius reduction to T_n. On the other hand, the additional geometric structure can be a threat for security. In fact, it can be exploited to perform DLP attacks. Indeed, we will see in the next section that the best DLP attack in trace-zero subgroups T_n is Gaudry's index calculus attack. The complexity of the attack is $O(q^{2\frac{n-2}{n-1}})$. This complexity is lower than Pollard's rho complexity for $n > 3$. Hence, to have cryptosystems with the same level of security, we have to compare $T_n \subseteq E(\mathbb{F}_{q^n})$ with $E'(\mathbb{F}_{q'})$, where $q' \in O(q^{4\frac{n-2}{n-1}})$. For $n = 3$, the groups T_3 and $E'(\mathbb{F}_{q'})$ are of the same cardinality $O(q^2)$. However, for $n > 3$, we need to take larger trace-zero subgroups to have the same level of security of elliptic groups $E'(\mathbb{F}_{q'})$. As a consequence, when n is large, trace-zero subgroups are in theory not convenient to build cryptosystems. On the other hand, we point out that Gaudry's strategy requires to solve huge polynomial systems. Up to now, this task is not feasible in practice. Even if, up to now, Gaudry's index calculus attack is not effective, it could be a potential threat for DLP security in trace-zero subgroups T_n, with n large. So, for cryptographic applications, one takes trace-zero subgroups of small degree n: namely $n = 3, 5, 7$. We remark that the degree 3 trace-zero subgroups T_3 guarantee optimal DLP security and the best arithmetic performances. When we say that T_3 has optimal DLP security, we mean that, in T_3, the complexity of Gaudry's index calculus algorithm is the same as the complexity of the generic Pollard's rho attack: we refer to Section 1.5.1 for more details on the concept of optimal DLP security. Hence, degree three trace-zero subgroups can be compared with standard elliptic groups of the same size, that is, with $E'(\mathbb{F}_{q'})$, where q' is a prime and $q' \in O(|T_3|) = O(q^2)$. Moreover, if we apply Frobenius reduction in T_3, we increase the efficiency of the scalar product algorithms. In this way, the computation is faster than in elliptic groups $E'(\mathbb{F}_{q'})$ of the same size. Furthermore, computing the cardinality of T_3 is more efficient than computing the cardinality of $E'(\mathbb{F}_{q'})$. The reason is that, by the Hasse-Weil theorem and by Proposition 45, we need to compute the characteristic polynomial of the Frobenius endomorphism of E and of E', in order to get the cardinality of T_3 and $E'(\mathbb{F}_{q'})$ respectively ([73, Theorem 4.10 and Theorem 4.12], [6, Section 15.3.1]). We have that E is defined over $\mathbb{F}_q$, while E' is defined over $\mathbb{F}_{q'}$, with $q' \in O(q^2)$. Then the characteristic polynomial of the Frobenius endomorphism of the first elliptic curve is easier to compute. To conclude, trace-zero subgroups of small degree, and especially the degree 3 trace-zero subgroups, are a valid, concrete alternative to standard elliptic cryptosystem. Their relevance for applications in cryptography is one of the main motivations that lead us to the work of this thesis.

The following scheme collects the important cryptographic aspects of trace-zero subgroups, that we discussed in this section.

Scheme 6.

Trace-zero subgroups for cryptography.

Security.

- We can restrict from $E(\mathbb{F}_{q^n})$ to T_n without losing security.
- T_3 has optimal DLP security (the complexity of DLP attacks is the Pollard's rho complexity).

Efficiency.

- Fast algorithms to compute the group order, since $|T_n| = |E(\mathbb{F}_{q^n})|/|E(\mathbb{F}_q)|$.
- Scalar multiplication sped up by Frobenius reduction.
- Optimal efficient representations are known ([62], [75], [69], [27], [47], [49], Chapter 2).

1.5 Index calculus for the discrete logarithm problem

In Section 1, we gave the definition of Discrete Logarithm Problem in cyclic groups (Definition 4). We pointed out that the security of important cryptosystems is nowadays based on the assumption that this problem on certain groups is hard to solve. Therefore, the study of strategies for solving the DLP is a fundamental issue in cryptography. In the following, we give a brief survey of the current techniques in this field. We deal with the complexity of these techniques in the different groups that are used in cryptographic applications. We refer to [6, Chapters 19, 20, 21, 22] and [70]. In Subsection 1.5.3 we focus on the so-called index calculus method for DLP attacks. In Chapter 5, we propose an original variant of index calculus for trace-zero subgroups.

1.5.1 Generic algorithms for the DLP

We start our survey with the most famous generic algorithms for solving the DLP in a cyclic group G of order N. Generic algorithms are, by definition, algorithms that can be applied to any group. These methods do not take into account the additional properties of the given group. Therefore, generic algorithms are supposed to carry out a limited number of tasks. Namely, they can only perform the group operation, compute inverses, and recognize if two group elements are the same. Generic algorithms for the DLP include the Pohlig-Hellman method, the baby-step giant-step method, and the Pollard's rho method with its variants (the Pollard's kangaroo method and the lambda method). Pohlig-Hellman method reduces the computation of the DLP in G to the computation of DLP's in the cyclic subgroups of G of prime order. Thanks to this strategy, one can always take cyclic groups G of prime order, for cryptographic applications. Hence, from now on, we suppose that N is a prime number. Due to a result of Shoup ([68]), a generic algorithm for the DLP in G must perform $\Omega(\sqrt{N})$ group operations. Nowadays, the best generic algorithm to solve the DLP is the Pollard's rho method, which has complexity $O(\sqrt{N})$: such complexity achieves the order of magnitude of Shoup's lower bound. Therefore, we say that a group G has optimal DLP security if the complexity of solving the DLP in G is the same as the complexity of the Pollard's rho method in the group. Pollard's rho is based on the well known probabilistic result called birthday paradox. According to this result, the expected number of draws of random elements of G before finding a collision is $O(\sqrt{N})$. The complexity of Pollard's rho method follows from this result.

1.5.2 Better than generic: exploit the structure to attack the cryptosystem

Non-generic algorithms for solving the DLP follow two main directions: the transfer of the DLP and the index calculus method. The transfer of the DLP consists of reducing the DLP in the given group to a DLP in another group, in which it is easier to solve. This can be done via pairings, as well as via Weil restriction of scalars. Pairings are non-degenerate bilinear maps $e : G \times G \longrightarrow H$, where G is the group in which one has to solve the DLP, and H is another group . If e is efficiently computable, one can take the injective group homomorphism $e_{P'} : G \longrightarrow H$, such that $e_{P'}(Q) = e(Q, P')$, for some fixed $P' \in G$. In such a way, one reduces the computation of the discrete logarithm for $Q = \ell P$ in G to the computation of the discrete logarithm for $e_{P'}(Q) = \ell e_{P'}(P)$ in H. This means that the complexity of the DLP in G is at most the complexity of the DLP in H. Hence, the DLP attack is successful if the DLP in H is easy to solve.

Example 46. Take a group $G = E(\mathbb{F}_q)$, of prime order N. One can exploit the so-called Tate pairing or the Weil pairing to transfer the DLP to the group $H = \mu_N = \{x \in \overline{\mathbb{F}}_q : x^N = 1\} \subseteq \mathbb{F}_{q^k}$ for some $k \geq 1$. Hence the attack is successful if k is a small integer. We remark that this is the case only for special elliptic curves (namely, the supersingular elliptic curves). These curves have to be discarded to avoid such kind of attack. On the other hand, we point out that the existence of an efficient bilinear structure between groups gives also good advantages for computation performances. For example, the use of pairings allows to speed up the computation of secret keys that have to be shared by three (or more) parties, in tripartite key exchange protocols (see [6, Section 1.6.4]). Therefore, pairings are widely studied and they find their application in some cryptographic settings. This is another example in which some algebraic/geometric features of the given group determine fallings in security and benefits in arithmetic at the same time.

Concerning Weil restriction of scalars, we refer to Example 42. In this example, we explained the procedure for elliptic groups $E(\mathbb{F}_{q^n})$. The strategy can be applied to groups of points of generic abelian varieties. The idea is to transfer the DLP to a group of points of a variety of higher dimension, in which the DLP could be easier to solve. As a matter of fact, we mentioned in the previous section that higher dimensions can determine a reduction in the level of security. Among the attacks that one can apply to the group of $\mathbb{F}_q$-rational points of an abelian variety V defined over $\mathbb{F}_q$, of dimension $d > 1$, we recall the cover attacks and Gaudry's index calculus strategy. The main difference between the two attacks is that the first one applies only to special varieties, while the second strategy works with any d-dimensional variety. The latter method combines Weil restriction of scalars with the index calculus method. We describe the index calculus method in the following subsection.

We give in the scheme below the list of the DLP algorithms that we have just mentioned.

Scheme 7.

Algorithms for the DLP.

Generic algorithms.
Algorithms that work in any group.

- Pohlig-Hellman: we can always take G of prime order.
- Baby-step, giant-step and Pollard's rho: optimal complexity $O(\sqrt{|G|})$ (Shoup).

Non generic algorithms.
Algorithms that exploit the algebraic/geometric structure of the given group.

- DLP transfer: pairings and Weil restriction of scalars.
- Index calculus.

1.5.3 Index calculus

Index calculus is a general strategy for DLP attacks. It is based on the possibility of decomposing the elements of the group into sums of group elements that belong to a chosen subset of the group, called the factor base. We give below the general outline of the algorithm. This is divided into four main step: (1) choice of the factor base, (2) search for relations, (3) linear algebra, (4) computation of the individual logarithm.

Algorithm 2 (Index calculus)**.**

Input : P and Q
Output: $\ell = log_P(Q)$

Step 1: **Choice of the factor base**
Choose $\mathcal{F} = \{P_1, \cdots, P_h\} \subseteq G$

Step 2: **Search for relations**
Find $k > h$ relations of the form:

$$\alpha_i P + \beta_i Q = \sum_{j=1}^{h} m_{j,i} P_j, \tag{1.16}$$

where α_i, β_i are randomly chosen elements of $\mathbb{Z}_N$ for all $i \in \{1, \cdots, k\}$, and $m_{j,i} \in \mathbb{Z}_N$ for $j \in \{1, \cdots, h\}$, $i \in \{1, \cdots, k\}$

Step 3: **Linear algebra**
Find a nonzero vector $(\gamma_1, \cdots, \gamma_k) \in \mathbb{Z}_N^k$ in the right kernel of $M = (m_{j,i})$

Step 4: **Individual logarithm**
if $\sum_{i=1}^{k} \beta_i \gamma_i$ is invertible modulo N **then**
return $\ell = -(\sum_{i=1}^{N} \alpha_i \gamma_i)(\sum_{i=1}^{N} \beta_i \gamma_i)^{-1} \mod N$
else come back to Step 2
end if

Proposition 47. *Algorithm* 2 *is correct.*

Proof. Notice that, since $k > h$, there exists a nonzero vector in the right kernel of M. This means that there exists $(\gamma_1, \cdots, \gamma_k) \in \mathbb{Z}_N^k \setminus \{(0, \cdots, 0)\}$, such that $\sum_{i=1}^{k} m_{j,i}\gamma_i = 0$ for all j. Hence, to prove the correctness of the algorithm, multiply relation (1.16) by γ_i and then sum over i. □

Remark 48. Notice that, if one enlarge the factor base, the second step of index calculus becomes easier. In fact, the probability of finding a relation increases. On the other side, however, one has to solve a larger linear system. Hence, the third step becomes more difficult. Vice versa, if one takes a smaller factor base, the linear algebra step is easier. In fact, we deal with a matrix of smaller dimension. However, finding relations is more difficult. Hence, for a suitable choice of the factor base, one has to take into account an optimal balance between the complexities of step 2 and step 3 of Algorithm 2.

The index calculus strategy has been specialized to the different families of groups that are suitable for cryptographic applications. Among such groups, we recall the multiplicative cyclic groups $\mathbb{Z}_p \setminus \{0\}$ of prime fields $\mathbb{Z}_p$, multiplicative cyclic groups $\mathbb{F}_q \setminus \{0\}$ of extension fields $\mathbb{F}_q$, Picard groups of elliptic and hyperelliptic curves. Such specialized index calculus algorithms often achieve better complexity than the Pollard's rho complexity. In some cases, their complexity is subexponential. Among these techniques, we recall the quadratic sieve and the number field sieve for the group $\mathbb{Z}_p \setminus \{0\}$. They have subexponential complexity $L(\alpha, c)$, with $\alpha = \frac{1}{2}$ and $\alpha = \frac{1}{3}$ respectively. In $\mathbb{F}_q \setminus \{0\}$, where q is a prime power, one has the function field sieve, which has again subexponential complexity with $\alpha = \frac{1}{3}$. For the group $\mathrm{Pic}^0(C)(\mathbb{F}_q)$ of $\mathbb{F}_q$-rational divisor classes of a hyperelliptic curve C of genus g defined over $\mathbb{F}_q$, both Adlemann, de Marrais, Huang ([1]), and Enge, Gaudry ([36], [37]) wrote algorithms that achieve subexponential complexity with $\alpha = \frac{1}{2}$, when g is much larger than q. Moreover, for the same family of groups, Gaudry gave an algorithm which has complexity $O(q^2)$ when $q \geq g!$ (see [45]). One has that $|\mathrm{Pic}^0(C)(\mathbb{F}_q)| \in O(q^g)$: this is a generalization of the Hasse-Weil theorem for elliptic curves, see [71, Theorem *V*.2.3]. Hence, Gaudry's method performs better than the Pollard's rho method for $g > 4$.

Gaudry's index calculus for abelian varieties

In [46], Gaudry proposed an index calculus strategy for groups $V(\mathbb{F}_q)$, where V is a general abelian variety of dimension $d > 1$ defined over $\mathbb{F}_q$. In his technique, the factor base consists of the $\mathbb{F}_q$-rational points of an absolutely irreducible curve contained in the variety itself. Moreover, for step 2 of Algorithm 2 (the relation search step), Gaudry uses the addition law on the variety: in this way, the problem of finding relations is reduced to the problem of solving polynomial systems of equations over $\mathbb{F}_q$. The complexity of this method is $O(q^{2-\frac{2}{d}})$ asymptotically in q, where d is regarded as a constant. Notice that Gaudry's strategy applies to any abelian variety of dimension $d > 1$ defined over $\mathbb{F}_q$. Let C be a hyperelliptic curve of genus g defined over $\mathbb{F}_q$. We recall from Section 4.1 that the Picard group $\mathrm{Pic}^0(C)$ can be given a structure of an abelian variety of dimension g defined over $\mathbb{F}_q$. Hence, if $g \geq 2$, Gaudry's strategy applies to the group $\mathrm{Pic}^0(C)(\mathbb{F}_q)$. Moreover, the complexity of this method for $\mathrm{Pic}^0(C)(\mathbb{F}_q)$ turns out to be lower than the Pollard's rho complexity whenever $g > 2$. Furthermore, we have seen in Example 42 that the group of $\mathbb{F}_{q^n}$-rational points of an elliptic curve E defined over $\mathbb{F}_q$ can be identified with the group $W_E(\mathbb{F}_q)$ of $\mathbb{F}_q$-rational points of the n-dimensional variety W_E defined over $\mathbb{F}_q$, via the process of Weil restriction of scalars. Then, Gaudry's algorithms applies also to groups $E(\mathbb{F}_{q^n})$, when $n > 1$. The complexity of the algorithm in these groups is lower than Pollard's rho complexity whenever $n > 2$. Finally, we pointed out in the previous section that, given n an odd prime number, the trace-zero subgroup $T_n \subseteq E(\mathbb{F}_{q^n})$ can be identified, via the process of Weil restriction of scalars, with the group of $\mathbb{F}_q$-rational points of the trace-zero variety $\mathcal{T}_n$. The trace-zero variety $\mathcal{T}_n$ is an abelian variety of dimension $n-1$ defined over $\mathbb{F}_q$. Therefore, one can apply Gaudry's algorithm to T_n. Gaudry's index calculus algorithm in T_n has lower complexity than the generic Pollard's rho method whenever $n \geq 5$. On the other hand, however, the heuristic complexity of this algorithm, analyzed in [46], hides a constant which depends on the dimension d of the variety. The complexity could grow very fast in d, so that in practice the computation required by the algorithm could not be manageable, except for small dimensions.

The general strategy of Gaudry has been applied to the group $E(\mathbb{F}_{q^n})$, with $n > 1$, by Gaudry himself ([46]), Diem ([30], [31]), and Joux, Vitse ([51]). Moreover, it has been applied to the trace-zero subgroup $T_n \subseteq E(\mathbb{F}_{q^n})$, with n odd prime, by Gorla-Massierer ([48]). In all these works, Semaev' summation polynomials (see Section 1.3.3) are used

for the relation search step. In the next paragraph we give more details about the index calculus variant proposed in [48], for index calculus in trace-zero subgroups. In Chapter 5 we compare this method with our variant of index calculus in T_n.

A variant of Gaudry's index calculus for trace-zero subgroups. In [48], the authors identify T_n with the group of $\mathbb{F}_q$-rational points of the trace-zero variety $\mathcal{T}_n$, via the process of Weil restriction of scalars. For the first step of the index calculus algorithm, they choose the factor base

$$\mathcal{F} = \{P = (x_P, y_P) = ((0, 0, \cdots, x_{n-1}, x_n), y_P) \in T_n\}.$$

This is the set of $\mathbb{F}_q$-rational points of a curve contained in T_n. The authors make the general assumption that such curve is absolutely irreducible, and that it is not contained in any proper subvariety of $\mathcal{T}_n$. For the second step of Algorithm 2, the authors look for relations of the form

$$R = \alpha_i P + \beta_i Q = P_1 \oplus \cdots \oplus P_{n-1}, \tag{1.17}$$

where R is a known trace-zero point, and $P_1, \cdots, P_{n-1} \in \mathcal{F}$ are unknown. Since the dimension of $\mathcal{T}_n$ is $d = n - 1$, then the probability of finding a relation of this type is $1/d! = 1/(n-1)!$ (see [46, Section 2.5]). Using the summation polynomials of the elliptic curve, each relation (1.17) is translated into a polynomial system of equations. The system consists of $2n-1$ polynomial equations and $2n-2$ unknowns. All equations of the system have coefficients in $\mathbb{F}_q$. More precisely, the unknowns are $x_{n-1,i}, x_{n,i}$, such that $x_{P_i} = (0, 0, \cdots, x_{n-1,i}, x_{n,i})$, for $i \in \{1, \cdots, n-1\}$. Let f_n be the n-th summation polynomial of the curve E. We have that $P_i \in \mathcal{F}$ if and only if $P_i \oplus \varphi(P_i) \oplus \cdots \oplus \varphi^{n-1}(P_i) = \mathcal{O}$, which implies $f_n(x_{P_i}, x_{\varphi(P_i)}, \cdots, x_{\varphi^{n-1}(P_i)}) = 0$. Applying Weil restriction of scalars to this equation, one obtains an equation of the form

$$F_i(x_{n-1,i}, x_{n,i}) = 0,$$

where F_i has coefficients in $\mathbb{F}_q$ and degree at most $(n-1)2^{n-2}$. Moreover, the relation (1.17) implies $f_n(x_{P_1}, \cdots, x_{P_{n-1}}, x_R) = 0$. Applying Weil restriction of scalars to this equation, one obtains n equations

$$G_j(x_{n-1,1}, x_{n,1}, \cdots, x_{n-1,n-1}, x_{n,n-1}) = 0 \text{ for } j \in \{1, \cdots, n\}.$$

Each equation G_j has coefficients in $\mathbb{F}_q$ and degree at most $(n-1)2^{n-2}$. Hence, for the relation search step, one has to solve the polynomial system

$$\begin{cases} F_i(x_{n-1,i} x_{n,i}) = 0 \text{ for } i \in \{1, \cdots, n-1\} \\ G_j(x_{n-1,1}, x_{n,1}, \cdots, x_{n-1,n-1}, x_{n,n-1}) = 0 \text{ for } j \in \{1, \cdots, n\} \end{cases}. \tag{1.18}$$

The system has $2n-2$ unknowns and $2n-1$ equations. Each equation has coefficients in $\mathbb{F}_q$ and degree at most $(n-1)2^{n-2}$.

We have seen that the complexity of Gaudry's index calculus strategy in $E(\mathbb{F}_{q^n})$ is lower than Pollard's rho complexity whenever $n > 2$. Moreover, if n is an odd prime, the complexity of this strategy in T_n is lower than Pollard's rho complexity whenever $n \geq 5$. We recall that we are speaking about the complexity in q, while n is regarded as a constant. On the other hand, all the mentioned index calculus variants for $E(\mathbb{F}_{q^n})$ ([46], [30], [31], [51]) and for T_n ([48]) have exponential complexity in n. This is due to the hardness of solving the polynomial systems that come from the relation search step. It follows that,

in order to make Gaudry's index calculus algorithm effective, it is of great importance to look for new techniques for the relation search step. The aim of such techniques would be to compute new polynomial systems, for which finding solutions is easier. In Chapter 5 we focus on this aim for index calculus in trace-zero subgroups. We refer to this chapter for the detailed presentation and analysis of our new variant of Gaudry's algorithm for trace-zero subgroups.

Chapter 2

Optimal representations for trace-zero subgroups of twisted Edwards curves

In this chapter, we propose two optimal representations for the family of trace-zero subgroups of twisted Edwards curves. For both representations, we provide efficient compression and decompression algorithms. The efficiency of these algorithms is compared with the efficiency of similar algorithms for elliptic curves in short Weierstrass form.

Our optimal representations for trace-zero subgroups of twisted Edwards curves draw inspiration from analogous ideas about optimal representations for trace-zero subgroups of elliptic curves in short Weierstrass form. More precisely, for the first representation, we follow ideas from [47]. We use Weil restriction of scalars (see Example 42 of Section 1.4) and Semaev's summation polynomials (see Subsection 1.3.3). For the second representation, we follow ideas from [49]. We make use of rational functions of the curve.

We refer to Section 1.3 and Section 1.4 for basic notions about twisted Edwards curves and trace-zero subgroups respectively. We use the notation of these sections. Let $\mathbb{F}_q$ be a finite field of characteristic different from $2, 3$. We take a twisted Edwards curve

$$E_{a,d} : ax^2z^2 + y^2z^2 = z^4 + dx^2y^2$$

defined over $\mathbb{F}_q$. We denote by $\oplus$ the operation of addition between the affine points of $E_{a,d}$. We denote by $\mathcal{O}$ the neutral element of the operation. Moreover, for each affine point P of the curve, we denote by $-P$ its inverse with resspect to $\oplus$. Let $E = (E_{a,d})^*_z$ be the affine dehomogenization of $E_{a,d}$ with respect to the variable z. We recall that, for each field extension $\mathbb{F}_q \subseteq \mathbb{L} \subseteq \overline{\mathbb{F}}_q$, the set $E(\mathbb{L})$ of affine $\mathbb{L}$-rational points of the curve $E_{a,d}$, is an abelian group with the operation $\oplus$. For a field extension of odd prime degree $\mathbb{F}_q \subseteq \mathbb{F}_{q^n}$, we denote by $T_n \subseteq E(\mathbb{F}_{q^n})$ the degree n trace-zero subgroup of the twisted Edwards curve. We refer to Definition 7 (Section 1.1) of optimal representation for a family of groups. Namely, the family of groups that we take into consideration in this chapter is

$$\mathcal{G}_n = (T_n \subseteq E(\mathbb{F}_{q^n}))_{q,E_{a,d}}.$$

The notation above means that, in the family $\mathcal{G}_n$, the odd prime n is fixed, while q and the twisted Edwards curve $E_{a,d}$ defined over $\mathbb{F}_q$ vary. We recall that, for this family of groups, we have $|T_n| \in O(q^{n-1})$ (see Section 1.4). Therefore, by Definition 7 and the subsequent remarks, an optimal representation

$$\mathcal{R}_n = (\mathcal{R}_{n,q,E_{a,d}})_{q,E_{a,d}}$$

for the family $\mathcal{G}_n$ can be given via maps of the type

$$\mathcal{R}_{n,q,E_{a,d}} : T_n \longrightarrow \mathbb{F}_q^{n-1} \times \mathbb{Z}_2^{k_{T_n}}. \tag{2.1}$$

These maps are such that there exist constants $c, d, e \in \mathbb{Z}_{\geq 0}$, for which the following facts hold. For all $T_n \in \mathcal{G}$, one has $k_{T_n} \leq e$. Moreover, there exists a subset $\mathcal{S}_{T_n} \subseteq \mathbb{F}_q^{n-1} \times \mathbb{Z}_2^{k_{T_n}}$ with $|\mathcal{S}_{T_n}| \leq c$ and $|\mathcal{R}^{-1}_{n,q,E_{a,d}}(x)| \leq d$, for all $x \in (\mathbb{F}_q^{n-1} \times \mathbb{Z}_2^{k_{T_n}}) \setminus \mathcal{S}_{T_n}$.
In the sequel, we give two optimal representations for the trace-zero family $\mathcal{G}_n$, with representation maps of the form (2.1). For the first representation of the chapter, we identify, in the general case, each trace-zero point P with its Frobenius conjugates ($\varphi^i(P)$ for $i \in \{0, \cdots, n-1\}$) and the inverses of its Frobenius conjugates ($-\varphi^i(P)$ for $i \in \{0, \cdots, n-1\}$). In rare occasions, we have some more ambiguity in the decompression. More precisely, we will see that this first representation is an optimal representation for a family $\mathcal{G}'_n \subseteq \mathcal{G}_n$, which disregards some exceptional cases. For the second representation, we have $d = n$. In fact, we identify each trace-zero point with its Frobenius conjugates.
We remark that an optimal representation for a family of groups is useful in cryptographic applications only if fast and efficient compression and decompression algorithms for the representation itself are given. In our setting, for each trace zero point $P \in T_n \subseteq E(\mathbb{F}_{q^n})$, one needs an efficient method to perform its compression, that is, to compute its representation $\mathcal{R}_{n,q,E_{a,d}}(P)$. Furthermore, for each $x \in \mathrm{Im}(\mathcal{R}_{n,q,E_{a,d}}) \subseteq (\mathbb{F}_q^{n-1} \times \mathbb{Z}_2^{k_{T_n}})$, an efficient method to compute its decompression $\mathcal{R}^{-1}_{n,q,E_{a,d}}(x)$ is required. For both representations that we propose in the chapter, we provide efficient algorithms for the compression and the decompression process.

We recall that each twisted Edwards curve $E_{a,d}$ defined over $\mathbb{F}_q$ is birationally equivalent over $\mathbb{F}_q$ to an elliptic curve in Montgomery form E_M, via the birational map Φ of Theorem 33. Moreover, the elliptic curve in Montgomery form E_M is isomorphic over $\mathbb{F}_q$ to an elliptic curve E_W in short Weierstrass form, via the isomorphism ϕ of Proposition 31. As a consequence, to optimally represent trace-zero points of the twisted Edwards curve $E_{a,d}$, one can use an optimal representation $\mathcal{R}'_n = (\mathcal{R}'_{n,q,E_W})_{q,E_W}$ for trace-zero subgroups of elliptic curves in short Weierstrass form. For example, we can use the optimal representation given in [47], or the one given in [49]. Such representations are given via maps of the type

$$\mathcal{R}'_{n,q,E_W} : T'_n \subseteq E_W(\mathbb{F}_{q^n}) \longrightarrow \mathbb{F}_q^{n-1} \times \mathbb{Z}_2^{k_{T'_n}}.$$

Hence, an optimal representation for $\mathcal{G}_n$ is

$$\mathcal{R}_{n,q,E_{a,d}} : T_n \longrightarrow \mathbb{F}_q^{n-1} \times \mathbb{Z}_2^{k_{T'_n}},\ P \mapsto \mathcal{R}'_{n,q,E_W}(\phi(\Phi^{-1}(P))).$$

Nevertheless, we expect that the direct construction of the optimal representation on the twisted Edwards curve gives faster and more efficient compression and decompression algorithms. Furthermore, we mentioned before that our optimal representations for trace-zero subgroups of twisted Edwards curves adapt ideas from [47] and [49], about optimal representations of trace-zero subgroups of elliptic curves in short Weierstrass form. Nevertheless, we point out that the adaptation is not straightforward, and that some obstacles have to be overcome, especially for adapting the method from [49].

The chapter is organized as follows. In Section 2.1 we propose our first optimal representation based on Weil restriction of scalars and summation polynomials, and we give compression and decompression algorithms for it. We then make explicit computations for the cases $n = 3$ and $n = 5$. We compare execution times of our Magma implementation with those of the corresponding algorithms for elliptic curves in short Weierstrass form,

proposed in [47]. In Section 2.2, we propose another representation based on rational functions, with its compression and decompression algorithms. We then make computations for $n = 3, 5$. We make efficiency comparisons with the corresponding algorithms for elliptic curves in short Weierstrass, given in [49]. When we count the number of operations in our computations, we denote respectively by M, S, and I multiplications, squarings, and inversions in the field. We do not take into account additions and multiplications by constants. The times for the implementation of our algorithms in Magma refer to version V2.22-1 of the software, running on a single 3 GHz core.

2.1 An optimal representation using summation polynomials

In this section, following ideas from [47], we use Weil restriction of scalars and Semaev's summation polynomials to give an optimal representation for the family of trace-zero subgroups

$$\mathcal{G}_n = (T_n \subseteq E(\mathbb{F}_{q^n}))_{q,E_{a,d}}.$$

We recall from Section 1.4.2 that an optimal representation for the family of groups $(E(\mathbb{F}_{q^n}))_{q,E_{a,d}}$ is $\rho_n = (\rho_{n,q,E_{a,d}})_{q,E_{a,d}}$, with

$$\rho_{n,q,E_{a,d}} : E(\mathbb{F}_{q^n}) \longrightarrow \mathbb{F}_q^n,\ (x, y) \mapsto (y_1, \cdots y_n).$$

The vector $(y_1, \cdots, y_n) \in \mathbb{F}_q^n$ corresponds to $y \in \mathbb{F}_{q^n}$ under the isomorphism $\mathbb{F}_{q^n} \cong \mathbb{F}_q^n$ induced by a chosen basis $\mathcal{B}$ of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$. Namely, a $\mathbb{F}_{q^n}$-rational affine point $P = (\overline{x}, \overline{y})$ of $E_{a,d}$ is represented via its y-coordinate $\overline{y}$. For the fixed basis $\mathcal{B} = \{\alpha_1, \cdots, \alpha_n\}$, one can write uniquely $\overline{y} = \overline{y}_1\alpha_1 + \cdots + \overline{y}_n\alpha_n$ for some $\overline{y}_1, \cdots, \overline{y}_n \in \mathbb{F}_q$. So $\overline{y}$ is uniquely determined by the vector $(\overline{y}_1, \cdots, \overline{y}_n)$ of its $\mathbb{F}_q$-coordinates. Hence, from the vector $(\overline{y}_1, \cdots, \overline{y}_n)$, one can recover the y-coordinate $\overline{y}$ of P. Then, from $\overline{y}$, one computes the x-coordinate $\overline{x}$ of P up to sign, using the equation of the curve. This means that we solve the equation $ax^2 + \overline{y}^2 - 1 - dx^2\overline{y}^2 = 0$ for the variable x. We recall that, for the optimal representation ρ_n, we have $c = e = 0$ and $d = 2$.
Let e_i be the i-th elementary symmetric polynomial in n variables, for $i \in \{1, \cdots, n\}$. Notice that, for $\overline{y} \in \mathbb{F}_{q^n}$, we have $e_i(\overline{y}, \overline{y}^q, \cdots, \overline{y}^{q^{n-1}}) \in \mathbb{F}_q$ for all $i \in \{1, \cdots, n\}$. Therefore, another optimal representation for the family $(E(\mathbb{F}_{q^n}))_{q,E_{a,d}}$ is $\hat{\rho}_n = (\hat{\rho}_{n,q,E_{a,d}})_{q,E_{a,d}}$, with

$$\hat{\rho}_{n,q,E_{a,d}} : E(\mathbb{F}_{q^n}) \longrightarrow \mathbb{F}_q^n,\ (x, y) \mapsto (e_1(y, y^q, \cdots, y^{q^{n-1}}), \cdots, e_n(y, y^q, \cdots, y^{q^{n-1}})).$$

For the optimal representation $\hat{\rho}_n$, we have again $c = e = 0$, while $d = 2n$. In fact, each $\mathbb{F}_{q^n}$-rational affine point of the curve is identified with its Frobenius conjugates, and with the inverses of its Frobenius conjugates. Namely, we have

$$\hat{\rho}_{n,q,E_{a,d}}(P) = \hat{\rho}_{n,q,E_{a,d}}(\pm\varphi^i(P))$$

for all $P \in E(\mathbb{F}_{q^n})$ and for all $i \in \{0, \cdots, n-1\}$.

As we pointed out in Section 1.4, one can restrict each map $\rho_{n,q,E_{a,d}}$ (or $\hat{\rho}_{n,q,E_{a,d}}$) from $E(\mathbb{F}_{q^n})$ to the subgroup T_n. In this way, we obtain two families of representations $(\rho_n)_{|T_n}$ and $(\hat{\rho}_n)_{|T_n}$ for the trace-zero family $\mathcal{G}_n$. Nevertheless, these two families are not optimal representations for $\mathcal{G}_n$. Our idea to recover the optimality of the representations is to drop one coordinate from each representation map. Namely, we can represent $P = (\overline{x}, \overline{y}) \in T_n$ via $(\overline{y}_1, \cdots, \overline{y}_{n-1}) \in \mathbb{F}_{q^{n-1}}$, or via $(\overline{e}_i)_{i=1,\cdots,n-1} = (e_i(\overline{y}, \overline{y}^q, \cdots, \overline{y}^{q^{n-1}}))_{i=1,\cdots,n-1} \in \mathbb{F}_q^{n-1}$.

Then, we can use an equation for the trace-zero subgroup T_n to compute the missing coordinate ($\overline{y}_n$ or $\overline{e}_n$). We use Semaev's summation polynomials to obtain such equation for T_n.

Let $f_n(y_1, \cdots, y_n) \in \mathbb{F}_q[y_1, \cdots, y_n]$ be the n-th summation polynomial of $E_{a,d}$, as in Theorem 37. By Theorem 37, we have that f_n is symmetric in $y_1, \cdots, y_n$. So one can write $f_n(y_1, \cdots, y_n)$ as a function of the elementary symmetric polynomials $e_1(y_1, \cdots, y_n)$, $\cdots$, $e_n(y_1, \cdots, y_n)$. Namely, there is a unique polynomial $g_n(e_1, \cdots, e_n) \in \mathbb{F}_q[e_1, \cdots, e_n]$ such that

$$g_n(e_1(y_1, \cdots, y_n), \cdots, e_n(y_1, \cdots, y_n)) = f_n(y_1, \cdots, y_n).$$

If $P = (\overline{x}, \overline{y}) \in T_n$, one has that $P \oplus \varphi(P) \oplus \cdots \oplus \varphi^{n-1}(P) = \mathcal{O}$. By definition of summation polynomial (Definition 36), this implies

$$\text{(a). } f_n(\overline{y}, \overline{y}^q, \cdots, \overline{y}^{q^{n-1}}) = 0, \quad \text{and} \quad \text{(b). } g_n(e_i(\overline{y}, \overline{y}^q, \cdots, \overline{y}^{q^{n-1}})_{i=1,\cdots,n}) = 0. \tag{2.2}$$

A partial converse and exceptions to the opposite implication are given in the next proposition.

Proposition 49. ([47, Lemma 1 and Proposition 4]) *Let $m \in \mathbb{Z}_{\geq 2}$. Take a field extension $\mathbb{F}_q \subseteq \mathbb{L} \subseteq \overline{\mathbb{F}}_q$. Denote by $E(\mathbb{L})[m]$ the subgroup of the m-torsion points of the group $E(\mathbb{L})$, that is $E(\mathbb{L})[m] = \{P \in E(\mathbb{L}) : mP = \mathcal{O}\}$. We have the following facts.*

1. $T_3 = \{(x, y) \in E(\mathbb{F}_{q^3}) : f_3(y, y^q, y^{q^2}) = 0\}$,
2. $T_5 \cup E(\mathbb{F}_q)[3] = \{(x, y) \in E(\mathbb{F}_{q^5}) : f_5(y, y^q, \ldots, y^{q^4}) = 0\}$,
3. $T_n \cup \bigcup_{k=0}^{\lfloor \frac{n}{2} \rfloor} E(\mathbb{F}_q)[n-2k] \subseteq \{(x, y) \in E(\mathbb{F}_{q^n}) : f_n(y, y^q, \ldots, y^{q^{n-1}}) = 0\}$ *for* $n \geq 7$.

Proof. The proof proceeds as in [47, Lemma 1 and Proposition 4], after observing that, for any odd prime n, one has $E(\overline{\mathbb{F}}_q)[2] \cap T_n = \{\mathcal{O}\}$. □

Remark 50. Our aim is to use (2.2.a) or (2.2.b) as an equation for T_n. In fact, we saw above that $P = (\overline{x}, \overline{y}) \in T_n$ implies both (2.2.a) and (2.2.b). Nevertheless, it follows from the previous proposition that the vice versa is not always true, since there are some few exceptions. Hence, Proposition 49 raises the question of efficiently deciding, for each root $y \in \mathbb{F}_{q^n}$ of equations (2.2), whether the corresponding points $(\pm x, y) \in E(\overline{\mathbb{F}}_q)$ are elements of T_n. However, this issue is easily solved in the two cases of major interest $n = 3$ and $n = 5$. In fact:

- By Proposition 49 (1), $(\pm x, y) \in T_3$ if and only if $x \in \mathbb{F}_{q^3}$.
- By Proposition 49 (2), $(\pm x, y) \in T_5$ if and only if $x \in \mathbb{F}_{q^5}$ and $(\pm x, y) \notin E(\mathbb{F}_q)[3] \setminus \{\mathcal{O}\}$. By storing the list $\mathcal{L}$ of the y-coordinates of the elements of $E(\mathbb{F}_q)[3] \setminus \{\mathcal{O}\}$, one can easily decide whether a point of $E(\mathbb{F}_{q^5})$ of coordinates (x, y) belongs to T_5 by checking that $y \notin \mathcal{L}$. Notice that $\mathcal{L}$ consists of at most 4 elements of $\mathbb{F}_q$.

Using the above considerations as a starting point, we can give an optimal representation for the trace-zero family $\mathcal{G}_n$, with efficient compression and decompression algorithms.

1. The representation. We mentioned before that our idea is to drop the coordinate y_n from the representation $(\rho_n)_{|T_n}$, or the coordinate e_n from the representation $(\hat{\rho}_n)_{|T_n}$, in order to obtain an optimal representation for $\mathcal{G}_n$. Then, we use the equation (2.2.a) or (2.2.b) to recover the missing $\mathbb{F}_q$-coordinate, paying attention to the special

cases of Proposition 49. We point out that the polynomial g_n is much more sparse than the polynomial f_n. This implies that, in general, the equation derived from (2.2.b) will be easier to solve. Hence we choose to represent trace-zero points with the elementary symmetric polynomials in the Frobenius powers of their y-coordinates. So we propose, for the trace-zero family $\mathcal{G}_n$, the family of representations $\mathcal{R}_n = (\mathcal{R}_{n,q,E_{a,d}})_{q,E_{a,d}}$, with

$$\mathcal{R}_{n,q,E_{a,d}} : T_n \longrightarrow \mathbb{F}_q^{n-1},\ (x,y) \mapsto (e_i(y, y^q, \cdots, y^{q^{n-1}}))_{i=1,\cdots,n-1}.$$

We will see in the following that $\mathcal{R}_n$ is an optimal representation for a family $\mathcal{G}'_n \subseteq \mathcal{G}_n$. We obtain $\mathcal{G}'_n$ by removing from $\mathcal{G}_n$ some exceptional cases.

2. Fast compression. In order to efficiently compute $e_i(\overline{y}, \overline{y}^q, \cdots, \overline{y}^{q^{n-1}})$, given $P = (\overline{x}, \overline{y}) \in T_n$, we make use of Weil restriction of scalars (see Example 42). We choose a basis $\mathcal{B}$ of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$, in such a way that the computation is particularly fast.

2.i. Choice of the basis. Since we have to compute Frobenius powers of elements of $\mathbb{F}_{q^n}$, it is convenient to choose a normal basis of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$. This is a basis of the form

$$\mathcal{B} = \{\alpha, \alpha^q, \cdots, \alpha^{q^{n-1}}\},$$

for some $\alpha \in \mathbb{F}_{q^n}$. One has that normal basis of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$ always exist, for each q and each n ([56, Theorem 2.35]). Observe that, for $\overline{y} = \overline{y}_1\alpha + \overline{y}_2\alpha^q + \cdots + \overline{y}_n\alpha^{q^{n-1}} \in \mathbb{F}_{q^n}$, we have

$$\overline{y}^q = \overline{y}_n\alpha + \overline{y}_1\alpha^q + \cdots + \overline{y}_{n-1}\alpha^{q^{n-1}}.$$

Therefore, if we choose a normal basis of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$, the rising of $\overline{y}$ to the Frobenius power q is costless. In fact, the operation reduces to a shift of the $\mathbb{F}_q$-coefficients $\overline{y}_1, \cdots, \overline{y}_n$ of $\overline{y}$. Notice moreover that

$$\overline{y} = \overline{y}_1\alpha + \overline{y}_2\alpha^q + \cdots + \overline{y}_n\alpha^{q^{n-1}} \in \mathbb{F}_q \text{ if and only if } \overline{y}_i = \overline{y}_j \text{ for all } i,j \in \{1,\cdots,n\}. \quad (2.3)$$

When $n|(q-1)$, an alternative to the choice of a normal basis is the choice of a Kummer basis of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$. Namely, if $n|(q-1)$, we can see $\mathbb{F}_{q^n}$ as a Kummer extension of $\mathbb{F}_q$, that is

$$\mathbb{F}_{q^n} \cong \mathbb{F}_q[\zeta]/(\zeta^n - \mu),$$

with $\mu \in \mathbb{F}_q$ and $\zeta^n - \mu$ irreducible polynomial of $\mathbb{F}_q[\zeta]$ (see [3, Chapter 6, Section 7]). So we can choose the Kummer basis

$$\mathcal{B} = \{1, \zeta, \cdots, \zeta^{n-1}\}.$$

Observe that $\zeta^q = \zeta^{q-1}\zeta = \mu^{\frac{q-1}{n}}\zeta$, since $n|(q-1)$ and $\zeta^n = \mu$. Let $h = \frac{q-1}{n}$. For $\overline{y} = \overline{y}_1 + \overline{y}_2\zeta + \cdots + \overline{y}_n\zeta^{n-1} \in \mathbb{F}_{q^n}$, we have that

$$\overline{y}^q = \overline{y}_1 + \overline{y}_2\mu^h\zeta + \cdots + \overline{y}_n\mu^{(n-1)h}\zeta^{n-1}.$$

Hence, even with the choice of a Kummer basis, the computation of the Frobenius powers in $\mathbb{F}_{q^n}$ is particularly easy. Notice moreover that

$$\overline{y} = \overline{y}_1 + \overline{y}_2\zeta + \cdots + \overline{y}_n\zeta^{n-1} \in \mathbb{F}_q \text{ if and only if } \overline{y}_i = 0 \text{ for all } i \in \{2,\cdots,n\}. \quad (2.4)$$

2.ii. Weil restriction of scalars. Let $\mathcal{B} = \{\alpha_1, \cdots, \alpha_n\}$ be a normal basis of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$, or a Kummer basis of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$, if $n|(q-1)$. We apply Weil restriction of scalars

with respect to $\mathcal{B}$, to each polynomial $e_i(y, y^q, \cdots, y^{q^{n-1}})$, $i \in \{1, \cdots, n\}$. Namely, in each polynomial $e_i(y, y^q, \cdots, y^{q^{n-1}})$, we replace y with $y_1\alpha_1 + \cdots + y_n\alpha_n$, where $y_1, \cdots, y_n$ are new variables. Then we obtain

$$e_i(y, y^q, \cdots, y^{q^{n-1}}) = \sum_{h=1}^{n} \overline{e}_{h,i}(y_1, \cdots, y_n)\alpha_h,$$

where $\overline{e}_{h,i}(y_1, \cdots, y_n) \in \mathbb{F}_q[y_1, \cdots, y_n]$ for each $h \in \{1, \cdots, n\}$, $i \in \{1, \cdots, n\}$.
For all $\overline{y} = \overline{y}_1\alpha_1 + \cdots + \overline{y}_n\alpha_n \in \mathbb{F}_{q^n}$, we have that $\overline{y}_j \in \mathbb{F}_q$ for all $j \in \{1, \cdots, n\}$. Hence the equality $\overline{y}_j^q = \overline{y}_j$ holds for all j. Therefore, we can reduce the polynomial $\overline{e}_{h,i} \in \mathbb{F}_q[y_1, \cdots, y_n]$ modulo $y_j^q - y_j$, for $i, h, j \in \{1, \cdots, n\}$.
If $\mathcal{B} = \{\alpha_1, \cdots, \alpha_n\} = \{\alpha, \alpha^q, \cdots, \alpha^{q^{n-1}}\}$ is a normal basis of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$, we obtain

$$\sum_{h=1}^{n} \overline{e}_{h,i}(y_1, \cdots, y_n)\alpha^{q^{h-1}} = \tilde{e}_i(y_1, \cdots, y_n)(\alpha + \alpha^q + \cdots \alpha^{q^{n-1}}) \mod (y_j^q - y_j),\ j \in \{1, \cdots, n\},$$

for some $\tilde{e}_i(y_1, \cdots, y_n) \in \mathbb{F}_q[y_1, \cdots, y_n]$, and for all $i \in \{1, \cdots, n\}$. In fact, $e_i(\overline{y}, \overline{y}^q, \cdots, \overline{y}^{q^{n-1}}) \in \mathbb{F}_q$ for $\overline{y} \in \mathbb{F}_{q^n}$. Hence the obtained equality is a consequence of [47, Proposition 3] and of the observation (2.3).
For the same reason, if $\mathcal{B} = \{1, \zeta, \cdots, \zeta^{n-1}\}$ is a Kummer basis of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$, we obtain

$$\sum_{h=1}^{n} \overline{e}_{h,i}(y_1, \cdots, y_n)\zeta^{h-1} = \tilde{e}_i(y_1, \cdots, y_n) \mod (y_j^q - y_j),\ j \in \{1, \cdots, n\},$$

for some $\tilde{e}_i(y_1, \cdots, y_n) \in \mathbb{F}_q[y_1, \cdots, y_n]$, and for all $i \in \{1, \cdots, n\}$. This is a consequence of [47, Proposition 3] and of the observation (2.4). In this way, we have computed n polynomials

$$\tilde{e}_i(y_1, \cdots, y_n) \in \mathbb{F}_q[y_1, \cdots, y_n],\ i \in \{1, \cdots, n\}. \tag{2.5}$$

These polynomials are such that, for each $(\overline{x}, \overline{y}) \in T_n$ with $\overline{y} = \overline{y}_1\alpha_1 + \cdots + \overline{y}_n\alpha_n$, we have

$$e_i(\overline{y}, \cdots, \overline{y}^{q^{n-1}}) = \lambda\tilde{e}_i(\overline{y}_1, \cdots, \overline{y}_n) \text{ for } i \in \{1, \cdots, n\},$$

where

$$\lambda = (\alpha + \alpha^q + \cdots + \alpha^{q^{n-1}}) \in \mathbb{F}_q \text{ if } \mathcal{B} = \{\alpha_1, \cdots, \alpha_n\} = \{\alpha, \alpha^q, \cdots, \alpha^{q^{n-1}}\} \text{ normal basis}, \tag{2.6}$$

and

$$\lambda = 1 \text{ if } \mathcal{B} = \{\alpha_1, \cdots, \alpha_n\} = \{1, \zeta, \cdots, \zeta^{n-1}\} \text{ Kummer basis}. \tag{2.7}$$

Hence, to compress $P = (\overline{x}, \overline{y}) \in T_n$, we compute

$$\mathcal{R}_{n,q,E_{a,d}}(P) = \lambda(\tilde{e}_i(\overline{y}_1, \cdots, \overline{y}_n))_{i=1,\cdots,n-1} \in \mathbb{F}_q^{n-1},$$

where $\lambda \in \mathbb{F}_q$ is the constant (2.6) or (2.7), according to the chosen basis $\mathcal{B}$.

To sum up, using Weil restriction of scalars, we replace computation in $\mathbb{F}_{q^n}$ with computation in $\mathbb{F}_q$. Such computation is particularly efficient since we choose appropriate basis $\mathcal{B}$ of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$, namely normal basis or Kummer basis of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$.

3. Fast decompression. For $(\overline{e}_1, \cdots, \overline{e}_{n-1}) \in \mathrm{Im}(\mathcal{R}_{n,q,E_{a,d}})$, we first solve

$$g_n(\overline{e}_1, \cdots, \overline{e}_{n-1}, t) = 0 \tag{2.8}$$

for the variable t.
Now, for any solution $\overline{e}_n \in \mathbb{F}_q$ of (2.8), we have to find the corresponding $\overline{y} \in \mathbb{F}_{q^n}$. In order to do this, we can proceed in two ways. One method is to compute a $\mathbb{F}_{q^n}$-root of the polynomial

$$Q(y) = (\sum_{i=1}^{n}(-1)^i \overline{e}_i y^{n-i}) + y^n.$$

This process requires a polynomial factorization in $\mathbb{F}_{q^n}$, for a polynomial with coefficients in $\mathbb{F}_q$ and of degree n. Alternatively, we can solve the system

$$\overline{e}_i = \lambda \tilde{e}_i(y_1, \cdots, y_n) \text{ for } i \in \{1, \cdots, n\}, \tag{2.9}$$

where $\lambda \in \mathbb{F}_q$ is the constant of (2.6) or (2.7), according to the basis $\mathcal{B}$ that we chose in the process of compression. We solve the system respect to the variables $y_1, \cdots, y_n$, to find $(\overline{y}_1, \cdots, \overline{y}_n) \in (\mathbb{F}_q)^n$ corresponding to $\overline{y} \in \mathbb{F}_{q^n}$. This latter strategy requires, in general, the computation of a lexicographical Gröbner basis in $\mathbb{F}_q[y_1, \cdots, y_n]$. In the next subsection, we will see that, for $n = 3$ and choosing a Kummer basis of $\mathbb{F}_{q^3}$ over $\mathbb{F}_q$, the second method is more efficient than the first one. In fact, in this case, the system (2.9) is particularly easy to solve and the computation of the Gröbner basis is not required. However, the computational experiments of [47] shows that, for $n = 5$, the polynomial factorization is faster than the resolution of system (2.9) via Gröbner basis techniques. So we expect that, for $n \geq 5$, the factorization method is more efficient than the Gröbner basis method.
Once we recover $\overline{y} \in \mathbb{F}_{q^n}$, we can compute $\overline{x} \in \mathbb{F}_{q^n}$ up to sign, using the equation of the twisted Edwards curve.

4. Optimality of the representation. We show that $\mathcal{R}_n$ is an optimal representation for $\mathcal{G}'_n \subseteq \mathcal{G}_n$, where $\mathcal{G}'_n$ is $\mathcal{G}_n$ without some exceptions. We refer to Definition 7 of optimal representation for notation. Notice that, for $\mathcal{R}_n$, we can take $e = 0$. Now we have to show that, for $\mathcal{G}'_n$, there exist constants $c, d \in \mathbb{Z}_{\geq 0}$ such that, for all $T_n \in \mathcal{G}'_n$, there exists a subset $\mathcal{S}_{T_n} \subseteq \mathbb{F}_q^{n-1}$ with $|\mathcal{S}_{T_n}| \leq c$ and $|\mathcal{R}^{-1}_{n,q,E_{a,d}}(\overline{e}_1, \cdots, \overline{e}_{n-1})| \leq d$, for all $(\overline{e}_1, \cdots, \overline{e}_{n-1}) \in \mathbb{F}_q^{n-1} \setminus \mathcal{S}_{T_n}$. For $(\overline{e}_1, \cdots, \overline{e}_{n-1}) \in \mathbb{F}_q^{n-1}$, we take the polynomial $g_n(t) = g_n(\overline{e}_1, \cdots, \overline{e}_{n-1}, t)$ of equation (2.8).
Suppose first that $g_n(t) \neq 0$. In this case, we have that $\deg(g_n(t)) \leq 2^{n-2}$ by Theorem 37. So, by solving (2.8), we can find at most 2^{n-2} values for $\overline{e}_n \in \mathbb{F}_q$. Then, in the second step of the decompression procedure, we recover $\overline{y}$ from $\overline{e}_1, \cdots, \overline{e}_n$ up to Frobenius conjugates, and in the last step we recover $\overline{x}$ from $\overline{y}$, up to sign. Hence, when $g_n(t) \neq 0$, we have that $|\mathcal{R}^{-1}_{n,q,E_{a,d}}(\overline{e}_1, \cdots, \overline{e}_{n-1})| \leq (2^{n-2})2n = 2^{n-1}n = d$. We remark that, even if in theory we can have up to 2^{n-2} values for $\overline{e}_n$, it is a rare phenomenon in practice to obtain more than one value. This means that, for a generic point P, the representation $\mathcal{R}_n$ identifies only $\pm P$ and their Frobenius conjugates. We come back to this discussion in Subsection 2.1.2, where we analyze the issue for $n = 5$.
Now suppose that, for $(\overline{e}_1, \cdots, \overline{e}_{n-1}) \in \mathbb{F}_q^{n-1}$, we have $g_n(t) = 0$. This means that

$$g_n(\overline{e}_1, \cdots, \overline{e}_{n-1}, e_n) = 0$$

for all $e_n \in \overline{\mathbb{F}}_q$. In this case, our decompression algorithm is not efficient in practice. In fact, q is large in real applications, and one should apply the second and the third step of decompression for each element $\overline{e}_n \in \mathbb{F}_q$. Moreover, for such $(\overline{e}_1, \cdots, \overline{e}_{n-1}) \in \mathbb{F}_q^{n-1}$, we cannot prove any more that $|\mathcal{R}^{-1}_{n,q,E_{a,d}}(\overline{e}_1, \cdots, \overline{e}_{n-1})|$ is upper bounded by the constant d.

Hence, for all T_n, we define the set of exceptional cases

$$\mathcal{S}_{T_n} = \{(\overline{e}_1, \cdots, \overline{e}_{n-1}) \in \mathbb{F}_q^{n-1} : g_n(\overline{e}_1, \cdots, \overline{e}_{n-1}, t) = 0\}.$$

Take $g_n(e_1, \cdots, e_{n-1}, t) \in \mathbb{F}_q[e_1, \cdots, e_{n-1}][t]$. Denote by $C_j(e_1, \cdots, e_{n-1}) \in \mathbb{F}_q[e_1, \cdots, e_{n-1}]$ the coefficients of g_n with respect to the variable t, for $j \in \{0, \cdots, 2^{n-2}\}$. By Theorem 37, we have that $\deg(C_j) \leq (n-1)2^{n-2}$ for all $j \in \{0, \cdots, 2^{n-2}\}$. Then we have

$$\mathcal{S}_{T_n} = \{(\overline{e}_1, \cdots, \overline{e}_{n-1}) \in \mathbb{F}_q^{n-1} : C_j(\overline{e}_1, \cdots, \overline{e}_{n-1}) = 0 \text{ for } j \in \{0, \cdots, 2^{n-2}\}\}.$$

Hence the set $\mathcal{S}_{T_n}$ is the set of $\mathbb{F}_q$-solutions of the system

$$C_j(e_1, \cdots, e_{n-1}) = 0 \text{ for } j \in \{0, \cdots, 2^{n-2}\}. \tag{2.10}$$

This system has up to $m_{\mathcal{S}} = 2^{n-2} + 1$ polynomial equations with coefficients in $\mathbb{F}_q$ and $n_{\mathcal{S}} = n-1$ variables. Each equation is of degree $d_{\mathcal{S}} \leq (n-1)2^{n-2}$. Since $m_{\mathcal{S}} > n_{\mathcal{S}}$, we can make the general assumption that the system (2.10) is zero dimensional. This means that the system has a finite number of solutions on the algebraic closure $\overline{\mathbb{F}}_q$. In this case, by Bezout's theorem, one has that the number of solutions of the system over the algebraic closure is bounded by $(d_{\mathcal{S}})^{n_{\mathcal{S}}}$. Hence we obtain

$$|\mathcal{S}_{T_n}| \leq (n-1)^{(n-1)} 2^{(n-2)(n-1)} = c,$$

for the trace-zero family

$$\mathcal{G}'_n = \{T_n \subseteq E(\mathbb{F}_{q^n}) : \text{the system (2.10) is zero-dimensional } \} \subseteq \mathcal{G}_n.$$

We now give the pseudocode of a compression and decompression algorithm for the elements of T_n. The correctness of Algorithm 3 and Algorithm 4 below is a straightforward consequence of the previous discussion.

Algorithm 3 (Compression).

Input : $P = (\overline{x}, \overline{y}) \in T_n \subseteq E(\mathbb{F}_{q^n})$
Output : $\mathcal{R}_{n,q,E_{a,d}}(P) \in \mathbb{F}_q^{n-1}$

$\mathcal{B} = \{\alpha_1, \cdots, \alpha_n\} \leftarrow$ chosen basis of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$ (normal basis or Kummer basis)
if $\mathcal{B} = \{\alpha, \alpha^q, \cdots, \alpha^{q^{n-1}}\}$ normal basis **then**
  $\lambda \leftarrow (\alpha + \cdots + \alpha^{q^{n-1}})$
else if $\mathcal{B}$ Kummer basis **then**
  $\lambda \leftarrow 1$
end if
Write $\overline{y} = \overline{y}_1\alpha_1 + \ldots + \overline{y}_n\alpha_n$
$\tilde{e}_i(y_1, \ldots, y_n) \in \mathbb{F}_q[y_1, \cdots, y_n] \leftarrow$ polynomials of (2.5) for $i = 1, \ldots, n-1$
Compute $\overline{e}_i = \lambda\tilde{e}_i(\overline{y}_1, \ldots, \overline{y}_n)$ for $i = 1, \ldots, n-1$
return $(\overline{e}_1, \ldots, \overline{e}_{n-1})$

Algorithm 4 (Decompression).

Input : $(\overline{e}_1, \ldots, \overline{e}_{n-1}) \in \mathbb{F}_q^{n-1}$
Output : $\mathcal{R}^{-1}_{n,q,E_{a,d}}(\overline{e}_1, \ldots, \overline{e}_{n-1}) \subseteq T_n$

1: $\mathcal{B}, \lambda \leftarrow$ basis and constant of Algorithm 3
2: Solve $g_n(\overline{e}_1, \ldots, \overline{e}_{n-1}, t) = 0$ for t in $\mathbb{F}_q$
3: $T \leftarrow$ list of solutions of $g_n(e_1, \ldots, e_{n-1}, t) = 0$ in $\mathbb{F}_q$
4: **for** $\overline{e}_n \in T$:

Method 1: $\tilde{e}_i(y_1, \ldots, y_n) \in \mathbb{F}_q[y_1, \cdots, y_n] \leftarrow$ polynomials of (2.5) for $i = 1, \ldots, n$
Find a solution $(\overline{y}_1, \cdots, \overline{y}_n) \in \mathbb{F}_q^n$ of the system

$$\begin{cases} \overline{e}_1 &= \lambda\tilde{e}_1(y_0, \ldots, y_{n-1}) \\ &\vdots \\ \overline{e}_n &= \lambda\tilde{e}_n(y_0, \ldots, y_{n-1}) \end{cases}$$

if it exists, and compute $\overline{y} = \overline{y}_1\alpha_1 + \cdots + \overline{y}_n\alpha_n \in \mathbb{F}_{q^n}$

Method 2: Compute $\overline{y}$ a $\mathbb{F}_{q^n}$-root of the polynomial

$$Q(y) = (\sum_{i=1}^{n}(-1)^i\overline{e}_i y^{i-1}) + y^n$$

5: For any found $\overline{y}$, recover one of the corresponding $\overline{x}$ using the curve equation
6: **end for**
7: **if** $(\overline{x}, \overline{y}) \in T_n$ **then**
8: Add $P = (\pm\overline{x}, \overline{y})$ and all their Frobenius conjugates to the list L of output points
9: **end if**
10: **return** L

2.1.1 Explicit equations, complexity, and timings for $n = 3$

In this subsection we give explicit equations for trace-zero point compression and decompression on twisted Edwards curves, for $n = 3$. We also estimate the number of operations needed for the computations. Moreover, we give some simulation times obtained with Magma, and compare with the results from [47] for elliptic curves in short Weierstrass form.

If we write the third summation polynomial of $E_{a,d}$ as a function of the elementary symmetric polynomials e_1, e_2, e_3, we obtain the polynomial

$$g_3(e_1, e_2, e_3) = e_1^2 - 1 + (d/a)(e_3^2 - e_2^2) + (2d/a)e_1e_3 - 2e_2 + ((-2a + 2d)/a)e_3. \quad (2.11)$$

Let $E : y^2z = x^3 + Axz^2 + Bz^3$ be an elliptic curve in short Weierstrass form. We remark that, if we write the third summation polynomial of E as a function of the elementary symmetric polynomials, we obtain the polynomial

$$G_3(e_1, e_2, e_3) = e_2^2 - 4e_1e_3 - 4Be_1 - 2Ae_2 + A^2. \quad (2.12)$$

Notice that, while G_3 is linear in e_1 and e_3, g_3 is of degree 2 in each variable. Therefore, none of e_1, e_2, e_3 is determined uniquely by the other two, as is the case of elliptic curves in Weierstrass form. However, applying the change of coordinates

$$\left\{\begin{array}{lcl} t_1 & = & e_1 \\ t_2 & = & e_3 + e_2 \\ t_3 & = & e_3 - e_2 \end{array}\right. \tag{2.13}$$

to g_3, we obtain the polynomial

$$\tilde{g}_3(t_1, t_2, t_3) = t_1^2 + (d/a)(t_2t_3 + t_1t_2 + t_1t_3) + ((d/a) - 2)t_2 + dt_3 - 1. \tag{2.14}$$

This polynomial is linear in both t_2 and t_3.
So, for $n = 3$, we take a variant of the optimal representation $\mathcal{R}_n$ that we proposed for the general case. Namely, we represent a trace-zero point $P = (\overline{x}, \overline{y}) \in T_3$ via the two $\mathbb{F}_q$-coordinates

$$(\bar{t}_1, \bar{t}_2) = (t_1(\overline{y}, \overline{y}^q, \overline{y}^{q^2}), t_2(\overline{y}, \overline{y}^q, \overline{y}^{q^2})) = (e_1(\overline{y}, \overline{y}^q, \overline{y}^{q^2}), (e_2 + e_3)(\overline{y}, \overline{y}^q, \overline{y}^{q^2})).$$

This variant allows us to recover the dropped coordinate

$$\bar{t}_3 = t_3(\overline{y}, \overline{y}^q, \overline{y}^{q^2}) = (e_3 - e_2)(\overline{y}, \overline{y}^q, \overline{y}^{q^2})$$

without any ambiguity, by solving the linear equation

$$\tilde{g}_3(t_1, t_2, t) = 0$$

for the variable t. Hence we take the optimal representation $\mathcal{R}'_3 = (\mathcal{R}'_{3,q,E_{a,d}})_{q,E_{a,d}}$, with

$$\mathcal{R}'_{3,q,E_{a,d}} : T_3 \longrightarrow \mathbb{F}_q^2,\ (x, y) \mapsto (t_i(y, y^q, y^{q^2}))_{i=1,2}.$$

As in the general case, we choose a normal basis or a Kummer basis $\mathcal{B} = \{\alpha_1, \alpha_2, \alpha_3\}$ of $\mathbb{F}_{q^3}$ over $\mathbb{F}_q$. Then we apply Weil restriction of scalars and reduction modulo $y_j^q - y_j$, for $j \in \{1, 2, 3\}$, to $t_i(y, y^q, y^q)$, for $i \in \{1, 2, 3\}$. We obtain the polynomials

$$\tilde{t}_i(y_1, y_2, y_3) \in \mathbb{F}_q[y_1, y_2, y_3] \text{ for } i \in \{1, 2, 3\}.$$

These polynomials are such that, for each $P = (\overline{x}, \overline{y}) \in T_3$, $\overline{y} = \overline{y}_1\alpha_1 + \overline{y}_2\alpha_2 + \overline{y}_3\alpha_3$, we have

$$t_i(\overline{y}, \overline{y}^q, \overline{y}^{q^2}) = \lambda\tilde{t}_i(\overline{y}_1, \overline{y}_2, \overline{y}_3) \text{ for } i \in \{1, 2, 3\},$$

where $\lambda = \alpha + \alpha^q + \alpha^{q^2}$ if we have chosen the normal basis $\mathcal{B} = \{\alpha, \alpha^q, \alpha^{q^2}\}$, while $\lambda = 1$ if the chosen basis $\mathcal{B}$ is a Kummer basis.

In the sequel of this subsection, we suppose that $3|(q - 1)$, and we regard $\mathbb{F}_{q^3} \cong \mathbb{F}_q[\zeta]/(\zeta^3 - \mu)$ as a Kummer extension of $\mathbb{F}_q$. We take the Kummer basis $\mathcal{B} = \{1, \zeta, \zeta^2\}$. In this way, we obtain

$$\left\{\begin{array}{l} t_1 = \tilde{t}_1(y_1, y_2, y_3) = 3y_1 \\ t_2 = \tilde{t}_2(y_1, y_2, y_3) = y_1^3 - 3\mu y_1y_2y_3 + \mu y_2^3 + \mu^2 y_3^3 + 3y_1^2 - 3\mu y_2y_3 \\ t_3 = \tilde{t}_3(y_1, y_2, y_3) = y_1^3 - 3\mu y_1y_2y_3 + \mu y_2^3 + \mu^2 y_3^3 - 3y_1^2 + 3\mu y_2y_3 \end{array}\right. \tag{2.15}$$

which express t_1, t_2, t_3 as polynomials in y_1, y_2, y_3.

Point compression. For a point $P = (\overline{x}, \overline{y}) \in T_3 \subseteq E(\mathbb{F}_{q^3})$, with $\overline{y} = \overline{y}_1 + \overline{y}_2\zeta + \overline{y}_3\zeta^2$, we perform the compression

$$\mathcal{R}'_{3,q,E_{a,d}}(P) = (\bar{t}_1, \bar{t}_2) = (3\overline{y}_1, \overline{y}_1^3 - 3\mu\overline{y}_1\overline{y}_2\overline{y}_3 + \mu\overline{y}_2^3 + \mu^2\overline{y}_3^3 + 3\overline{y}_1^2 - 3\mu\overline{y}_2\overline{y}_3).$$

If we compute $\bar{t}_2$ as $(\overline{y}_1 + 1)(\overline{y}_1^2 - 3\mu\overline{y}_2\overline{y}_3) + \mu\overline{y}_2^3 + \mu^2\overline{y}_3^3 + 2\overline{y}_1^2$, the cost of computing $\mathcal{R}'_{3,q,E_{a,d}}(P)$ is 3S+4M in $\mathbb{F}_q$. In the case of elliptic curves in short Weierstrass form, computing the representation of a point is less expensive, as it takes 1S+1M in $\mathbb{F}_q$ or 1M in $\mathbb{F}_q$ with the two methods of [47, Section 5].

Point decompression. We analyze the decompression algorithm for $n = 3$ and the optimal representation $\mathcal{R}'_3$.
As in the general case, we disregard $(t_1, t_2) \in \mathbb{F}_q^2$ such that $\tilde{g}_3(t) = \tilde{g}_3(t_1, t_2, t)$ is the zero polynomial. Therefore, we define

$$\mathcal{S}_{T_3} =$$

$$\{(t_1, t_2) \in \mathbb{F}_q^2 : \tilde{g}_3(t_1, t_2, t) = 0\} = \{(t_1, -t_1 - a) \in \mathbb{F}_q^2 : (a-d)t_1^2 + (2a - da - d)t_1 + a(2a - d - 1) = 0\}.$$

Notice that, for each $T_3 \subseteq E(\mathbb{F}_{q^3})$, we have $|\mathcal{S}_{T_3}| \leq 2 = c$, so $\mathcal{R}'_3$ is an optimal representation for $\mathcal{G}_3$ (we have $\mathcal{G}'_3 = \mathcal{G}_3$).

Now, in order to decompress $(\bar{t}_1, \bar{t}_2) \in \mathrm{Im}(\mathcal{R}'_{3,q,E_{a,d}}) \setminus \mathcal{S}_{T_3}$, we proceed as follows.

1. We solve $\tilde{g}_3(\bar{t}_1, \bar{t}_2, t) = 0$ for the variable t, that is, we compute $\bar{t}_3$ as

$$\bar{t}_3 = -\frac{((d/a) - 2)\bar{t}_2 + (d/a)\bar{t}_1\bar{t}_2 + (\bar{t}_1 + 1)(\bar{t}_1 - 1)}{(d/a)(\bar{t}_1 + \bar{t}_2 + a)}.$$

Notice that, if $(\bar{t}_1 + \bar{t}_2 + a) = 0$, then $\tilde{g}_3(\bar{t}_1, \bar{t}_2, \bar{t}_3) \neq 0$ for all $\bar{t}_3 \in \mathbb{F}_q$. In fact, we are supposing that $(\bar{t}_1, \bar{t}_2) \notin \mathcal{S}_{T_3}$, that is, $\tilde{g}_3(\bar{t}_1, \bar{t}_2, t) \neq 0$. On the other hand, the case $\tilde{g}_3(\bar{t}_1, \bar{t}_2, \bar{t}_3) \neq 0$ for all $\bar{t}_3 \in \mathbb{F}_q$ is not possible for $(\bar{t}_1, \bar{t}_2) \in \mathrm{Im}(\mathcal{R}'_{3,q,E_{a,d}})$, by definition of summation polynomials. So $(\bar{t}_1 + \bar{t}_2 + a) \neq 0$ for all $(\bar{t}_1, \bar{t}_2) \in \mathrm{Im}(\mathcal{R}'_{3,q,E_{a,d}}) \setminus \mathcal{S}_{T_3}$. Hence $\bar{t}_3$ can be computed with 3M+1I in $\mathbb{F}_q$.

2. Given $(\bar{t}_1, \bar{t}_2, \bar{t}_3)$, we solve system (2.15) with $t_i = \bar{t}_i$ for $i \in \{1, 2, 3\}$, for the variables y_1, y_2, y_3. Notice that, since the t_i's are obtained from the e_i's by a linear change of coordinates, all considerations from [47] apply to our situation. It follows that one can compute $\overline{y}$ from $(\bar{t}_1, \bar{t}_2, \bar{t}_3)$ with at most 3S+3M+1I, 1 square root and 2 cube roots in $\mathbb{F}_q$.

To sum up, the complete decompression algorithm takes at most 3S+6M+2I, 1 square root, and 2 cube roots in $\mathbb{F}_q$. For elliptic curves in short Weierstrass form, decompression takes at most 3S+5M+2I, 1 square root, and 2 cube roots in $\mathbb{F}_q$ or 4S+4M+2I, 1 square roots and 2 cube roots in $\mathbb{F}_q$, depending on the method used. We refer the interested reader to [47, Section 5], for details on the complexity of the computation for curves in short Weierstrass form.

Remark 51. Notice that one can also use (t_1, t_3) as an optimal representation of $(x, y) \in T_3$, and then solve $\tilde{g}_3$ for t_2 in order to recover y. This choice is analogous to the one we have made, and the computational cost of compression and decompression does not change.

Remark 52. The symmetry of twisted Edwards curves makes the computation of point addition on these curves more efficient than on elliptic curves in short Weierstrass form, as

we saw in Section 1.3.2. However, the same symmetry results in summation polynomials of higher degree and with a denser support. This explains our empirical observation that the summation polynomials in the elementary symmetric functions for elliptic curves in short Weierstrass form are sparser than those for twisted Edwards curves for $n = 3, 5$, even though for both curves they have the same degree 2^{n-2}. For $n = 3$, this behavior is apparent if one compares the polynomials (2.11) and (2.12). Therefore, one should expect that compression and decompression for a representation based on summation polynomials for twisted Edwards curves are less efficient than for elliptic curves in short Weierstrass form. This is confirmed by our findings.

The following examples and statistics have been implemented in Magma ([22],[23]), version V2.22-1 of the software, running on a single 3 GHz core.

Example 53. Let $q = 2^{79} - 67$ and $\mu = 3$. We choose randomly the twisted Edwards curve

$$E_{a,d} : 31468753957068040687814x^2z^2 + y^2z^2 = z^4 + 19269782127663896649899 7x^2y^2$$

defined over $\mathbb{F}_q$ and birationally equivalent over $\mathbb{F}_q$ to the elliptic curve in short Weierstrass form

$$E_W : y^2z = x^3 + 2924678484276594994785 03xz^2 + 3613610267364040043454 21z^3.$$

We choose a random affine point of trace-zero $P' \in E_W(\mathbb{F}_{q^3})$, and let P be the corresponding point on $E_{a,d}$. For brevity, here we only write the x-coordinates $x_{P'}$ of points of E_W and the y-coordinates y_P of points of $E_{a,d}$:

$$x_{P'} = 3465609281460769593147 53\xi^2+4568265396285359810342 12\xi+3441674704030266528266 72,$$

$$y_P = 2085207138975182362159 66\xi^2 + 4511219445502199473688 11\xi+$$
$$+6804108986042990130625 2.$$

We represent the points of E using the compression coordinates (t_1, t_2) from [47, Section 5]. We denote by $\mathcal{R}$ and $\mathcal{R}'$ the representation maps on $E_{a,d}$ and E, respectively. We compute

$$\mathcal{R}'(P') = (344167470403026652826672, 334324534997495805088214),$$

$$\mathcal{R}(P) = (204123269581289703918756, 9878878293607652441 3527).$$

We now apply the corresponding decompression algorithms to $\mathcal{R}'(P')$ and $\mathcal{R}(P)$. We obtain

$$\mathcal{R}'^{-1}(344167470403026652826672, 334324534997495805088214) =$$

$\{3465609281460769593147 53\xi^2+4568265396285359810342 12\xi+3441674704030266528266 72,$

$1647594986145075031874 93\xi^2+3615206909881977515343 81\xi+3441674704030266528266 72,$

$9314248304673012485077 5\xi^2+3905785889978954421374 49\xi+3441674704030266528266 72\}$,

which are exactly the x-coordinate of P' and its Frobenius conjugates. Similarly

$$\mathcal{R}^{-1}(204123269581289703918756, 9878878293607652441 3527) =$$

$\{2085207138975182362159 66\xi^2+4511219445502199473688 11\xi+6804108986042990130625 2,$

$5393215369610668550111 67\xi^2+2374313910976429683867 19\xi+6804108986042990130625 2,$

$4610835687560440834789 09\xi^2+5203724839667662589505 12\xi+6804108986042990130625 2\}$,

which are exactly the y-coordinate of P and its Frobenius conjugates.

We now give an estimate of the average time of compression and decompression for groups of different bit-size. We take primes q_1, q_2, and q_3 such that $3|(q_i - 1)$ for all i, of bit-length 96, 112, and 128, respectively. For each q_i, we take five pairs of birationally equivalent curves $(E, E_{a,d})$ defined over $\mathbb{F}_{q_i}$, such that the order of T_3 is prime of bit-length respectively 192, 224 and 256. On each pair of curves, we randomly choose $20'000$ pairs of points (P', P) of trace-zero, as in Example 53. For each pair of points, we compute $\mathcal{R}'(P'), \mathcal{R}(P), \mathcal{R}'^{-1}(\mathcal{R}'(P')), \mathcal{R}^{-1}(\mathcal{R}(P))$. For each computation, we take the average time in milliseconds for each curve, and then the averages over the five curves. The average computation times are given in the table below.

Table 1. Average computation times in milliseconds, for point compression and decompression in T_3, for elliptic curves in short Weierstrass form and for twisted Edwards curves.

Bit-length of $\lvert T_3\rvert$	192	224	256
Compression on E	0.006	0.005	0.006
Compression on $E_{a,d}$	0.016	0.017	0.015
Decompression on E	0.81	2.40	1.20
Decompression on $E_{a,d}$	0.88	2.44	1.17

The following table contains the ratios between the average times for point compression and decompression on elliptic curves in short Weierstrass form and twisted Edwards curves.

Table 2. Ratios between the average times for point compression and decompression on elliptic curves in short Weierstrass form and on twisted Edwards curves.

Bit-length of $\lvert T_3\rvert$	192	224	256
Comp on E / Comp on $E_{a,d}$	0.375	0.294	0.400
Dec on E / Dec on $E_{a,d}$	0.920	0.984	1.026

2.1.2 Explicit equations, complexity, and timings for $n = 5$

In this subsection, we treat in detail the case $n = 5$. We compute explicit equations for compression and decompression, we give an estimate of the complexity of the computations in terms of the number of operations, and give some timings computed in Magma. We also compare with the results obtained in [47] for elliptic curves in short Weierstrass form.

The fifth Semaev polynomial f_5 of a twisted Edwards curve has degree 40, while for curves in short Weierstrass form it has degree 32. The first polynomial also contains many more terms than the second. This agrees with what we observed in Remark 52 for the case $n = 3$. The fifth summation polynomial $g_5(e_1, \cdots, e_5)$ written as a function of the elementary symmetric polynomials $e_1, \cdots, e_5$ has degree 8 for both Weierstrass and twisted Edwards curves. However, for twisted Edwards curves, g_5 has degree 8 in each variable, while for elliptic curves in short Weierstrass form it has degree 6 in some of the variables. Because of these reasons, we expect that compression and decompression for a trace-zero subgroup coming from a twisted Edwards curve are less efficient than for one coming from a curve in short Weierstrass form.

For fields such that $16|(q - 1)$, we perform a linear change of coordinates on the e_i's in order to obtain a polynomial $\tilde{g}_5$, of degree strictly less than 8 in some variable. The polynomial g_5 is too big to be printed here. However, denoting by $(g_5)_8$ the part of g_5 which is homogeneous of degree 8, we have:

$$(g_5)_8(e_1, \ldots, e_5) = e_1^8 + (d/a)^4(e_2^8 + e_3^8) + (d/a)^8(e_4^8 + e_5^8). \tag{2.16}$$

Let $\mu_1 \in \overline{\mathbb{F}}_q$ be a primitive 16-th roots of unity. Then we can factor $t^8 + s^8$ over $\mathbb{F}_q$ as

$$t^8 + s^8 = (t - \mu_1 s)(t + \mu_1 s) r_6(t, s).$$

Therefore, (2.16) can be written in the form

$$(g_5)_8 = e_1^8 + (d/a)^4 (e_2 - \mu_1 e_3)(e_2 + \mu_1 e_3) p_6(e_2, e_3) + (d/a)^8 (e_4^8 + e_5^8).$$

Hence, after performing the change of coordinates

$$\begin{cases} t_2 &= e_2 - \mu_1 e_3 \\ t_3 &= e_2 + \mu_1 e_3 \\ t_i &= e_i \text{ for } i = 1, 4, 5 \end{cases}$$

we obtain a polynomial $\tilde{g}_5(t_1, \ldots, t_5)$ of degree 8 in t_1, t_4, t_5, and degree 7 in t_2, t_3. Then, in the case $16|(q-1)$ we can take the optimal representation $\mathcal{R}'_5 = (\mathcal{R}'_{5,q,E_{a,d}})_{16|(q-1),E_{a,d}}$, with

$$\mathcal{R}'_{5,q,E_{a,d}} : T_5 \longrightarrow \mathbb{F}_q^4,\ (x, y) \mapsto (t_1(y, \cdots, y^{q^4}), t_3(y, \cdots, y^{q^4}), t_4(y, \cdots, y^{q^4}), t_5(y, \cdots, y^{q^4})).$$

In such a way, we lower the degree of the equation in the first step of the decompression algorithm. In fact, $g_5(\overline{e}_1 \cdots, \overline{e}_4, t)$ has degree 8 in t, while $\tilde{g}_5(\overline{t}_1, t, \overline{t}_3, \overline{t}_4, \overline{t}_5)$ has degree 7 in t.

Example 54. Let $q = 2^{10} - 3$, $\mu = 2$. Let $\mathbb{F}_{q^5}$ be the Kummer extension $\mathbb{F}_{q^5} \cong \mathbb{F}_q[\zeta]/(\zeta^5 - \mu)$. Take the corresponding Kummer basis $\mathcal{B} = \{1, \zeta, \zeta^2\}$ of $\mathbb{F}_{q^5}$ over $\mathbb{F}_q$.
Let $E_{1,486}$ be the Edwards curve of equation

$$E_{1,486} : x^2 z^2 + y^2 z^2 = z^4 + 6x^2 y^2.$$

Let $P \in T_5$ be the point

$$P = (951\xi^4 + 338\xi^3 + 246\xi^2 + 934\xi + 133, 650\xi^4 + 927\xi^3 + 301\xi^2 + 171\xi + 973).$$

We denote by $\mathcal{R}$ the representation map on T_5. The compression of P is

$$\mathcal{R}(P) = (e_1, e_2, e_3, e_4) = (686, 289, 865, 418).$$

In order to decompress, we solve

$$g_5(e_1, e_2, e_3, e_4, t) = g_5(686, 289, 865, 418, t) =$$

$$71t^8 + 705t^7 + 1007t^6 + 970t^5 + 233t^4 + 1014t^3 + 356t^2 + 198t + 575 = 0,$$

which has a unique solution $e_5 = 790 \in \mathbb{F}_q$. In order to recover the value of the y-coordinate of P up to Frobenius conjugates, we find a root $\overline{y} \in \mathbb{F}_{q^5}$ of

$$y^5 - e_1 y^4 + e_2 y^3 - e_3 y^2 + e_4 y - e_5 = y^5 + 335y^4 + 289y^3 + 156y^2 + 418y + 231.$$

Hence, we use method 2 in line 4 of the decompression Algorithm 4. Notice that the five roots are Frobenius conjugates of each other. From $\overline{y} \in \mathbb{F}_{q^5}$ we can recompute the x-coordinate up to sign, via the curve equation. So the decompression algorithm returns $\mathcal{R}^{-1}(\mathcal{R}(P)) = \{\pm P, \pm\varphi(P), \pm\varphi^2(P), \pm\varphi^3(P), \pm\varphi^4(P)\}$.

We now give an example that has some indeterminacy in the decompression algorithm.

Example 55. Let $q = 2^{10} - 3$, $\mu = 2$, $\mathbb{F}_{q^5} \cong \mathbb{F}_q[\zeta]/(\zeta - \mu)$, and $\mathcal{B} = \{1, \zeta, \zeta^2\}$ Kummer basis of $\mathbb{F}_{q^5}$ over $\mathbb{F}_q$. Take the twisted Edwards curve

$$E_{210,924} : 210x^2z^2 + y^2z^2 = \zeta^4 + 924x^2y^2$$

and the point

$$P = (1020\xi^4 + 713\xi^3 + 158\xi^2 + 745\xi + 515, 891\xi^4 + 557\xi^3 + 135\xi^2 + 976\xi + 62) \in T_5.$$

As in the previous example, we denote by $\mathcal{R}$ the representation map on T_5. The compressed representation of P is

$$\mathcal{R}(P) = (e_1, e_2, e_3, e_4) = (310, 887, 19, 660).$$

The decompressing equation is

$$g_5(e_1, e_2, e_3, e_4, t) = 62t^8 + 502t^7 + 388t^6 + 294t^5 + 2t^4 + 466t^3 + 723t^2 + 55t + 388 = 0,$$

which has solutions $e_5 = 428, e_5' = 835, e_5'' = 550 \in \mathbb{F}_q$. By solving the equation

$$y^5 - e_1y^4 + e_2y^3 - e_3y^2 + e_4y - e_5 = y^5 + 310y^4 + 887y^3 + 19y^2 + 660y + 593 = 0$$

we recover the y-coordinate of P and all its Frobenius conjugates. By solving the equation

$$y^5 - e_1y^4 + e_2y^3 - e_3y^2 + e_4y - e_5' = y^5 + 310y^4 + 887y^3 + 19y^2 + 660y + 186 = 0$$

we find roots in $\mathbb{F}_{q^5}$, which do not correspond to points of trace-zero. By solving the equation

$$y^5 - e_1y^4 + e_2y^3 - e_3y^2 + e_4y - e_5'' = y^5 + 310y^4 + 887y^3 + 19y^2 + 660y + 471 = 0$$

we find $Q \in T_5$ which is not a Frobenius conjugate of P. Hence in this case

$$\mathcal{R}^{-1}(\mathcal{R}(P)) = \{\pm P, \ldots, \pm\varphi^4(P), \pm Q, \ldots, \pm\varphi^4(Q)\}.$$

As we pointed out in the general case, we remark that there could be $(\overline{e}_1, \cdots, \overline{e}_4) \in \mathrm{Im}(\mathcal{R}_{5,q,E_{a,d}})$ such that $g_5(t) = g_5(\overline{e}_1, \cdots, \overline{e}_4, t)$ is the zero polynomial. Moreover, when $g_5(t) \neq 0$, the equation $g_5(t) = g_5(\overline{e}_1, \cdots, \overline{e}_4, t) = 0$ can have up to eight $\mathbb{F}_q$-solutions. So it can happen to identify more than the Frobenius conjugates of a trace-zero point and their inverses. This is the case of Example 55. Nevertheless, an heuristic argument shows that a generic $(\overline{e}_1, \cdots, \overline{e}_4) \in \mathrm{Im}(\mathcal{R}_{5,q,E_{a,d}})$ has exactly ten inverse images (the Frobenius conjugates of the point and their inverses). In order to support the heuristics, we tested $15'000$ random points in the trace-zero subgroup T_5 of 15 twisted Edwards curves $E_{a,d}$. The groups had prime cardinality and bit-length $192, 224$, and 256. For any random point P, we computed the cardinality of $\mathcal{R}^{-1}_{5,q,E_{a,d}}(\mathcal{R}_{5,q,E_{a,d}}(P))$. We found that this cardinality is 10 for about 91% of the points, 20 for about 8.5% of the points, and 30 for about 0.5% of the points. We also found a few points for which $|\mathcal{R}^{-1}_{5,q,E_{a,d}}(\mathcal{R}_{5,q,E_{a,d}}(P))| = 40$, but the percentage was less than 0.02%. Finally, we did not find any point for which $40 < |\mathcal{R}^{-1}_{5,q,E_{a,d}}(\mathcal{R}_{5,q,E_{a,d}}(P))| \leq 80$, or for which $g_5(t)$ is the zero polynomial.

In order to test the efficiency of the compression and decompression algorithms for $n = 5$, we have implemented them in Magma. We take primes q_1, q_2, and q_3 of bit-length 48, 56, and 64, respectively. We choose primes such that $5|(q_i - 1)$ for all i. For each q_i we take five pairs of birationally equivalent curves $(E, E_{a,d})$ defined over $\mathbb{F}_{q_i}$, such

that the order of T_5 is prime of bit-length 192, 224, and 256, respectively. The following table contains the average times for compression and decompression in milliseconds. Each average is computed on a set of 20'000 randomly chosen points on each of the five curves.

Table 3. Average computation times in milliseconds, for point compression and decompression in T_5, for elliptic curves in short Weierstrass form and for twisted Edwards curves.

Bit-length of $\lvert T_5\rvert$	192	224	256
Compression on E	0.057	0.055	0.060
Compression on $E_{a,d}$	0.049	0.058	0.053
Decompression on E	64.17	104.31	121.51
Decompression on $E_{a,d}$	63.66	104.45	121.42

The following table contains the ratios between the average times for point compression and decompression on elliptic curves in short Weierstrass form and twisted Edwards curves.

Table 4. Ratios between the average times for point compression and decompression on elliptic curves in short Weierstrass form and on twisted Edwards curves.

Bit-length of $\lvert T_5\rvert$	192	224	256
Comp on E / Comp on $E_{a,d}$	1.163	0.948	1.132
Dec on E / Dec on $E_{a,d}$	1.008	0.999	1.001

2.2 An optimal representation using rational functions

Let $E_{a,d}$ be a twisted Edwards curve defined over $\mathbb{F}_q$. In this section, we propose another optimal representation for the trace-zero family $\mathcal{G}_n = (T_n \subseteq E(\mathbb{F}_{q^n}))_{q,E_{a,d}}$. This representation makes use of rational functions of the curve. We refer to Section 1.2 for basic notions about rational functions and divisors of a curve and the annexed notation. In [49], the authors propose to represent trace-zero points $P \in T_n$ of an elliptic curve in short Weierstrass form via the coefficients of the rational function which corresponds to the principal divisor $P + \varphi(P) + \ldots + \varphi^{n-1}(P) - n\mathcal{O}$ on the curve. Optimality of the representation depends on the fact that the rational function associated to this divisor has a special form, and it can be given via $n-1$ coefficients of $\mathbb{F}_q$. In order to adapt this idea to the case of twisted Edwards curves, there are several questions that need to be answered, and one has to overcome some difficulties in order to successfully carry out the same strategy. For example, we saw in Section 1.3.2 that a twisted Edwards $E_{a,d}$ has two singular points, namely the two points at infinity Ω_1 and Ω_2. On the other hand, an elliptic curve is a nonsingular projective curve. Since a twisted Edwards curve has two singular points, we have to care about the definitions of divisor, principal divisor and divisor of a rational function for the curve $E_{a,d}$: we explained this issue in Example 26.

We start with recalling some preliminaries and notations about rational functions and divisors on a twisted Edwards curve.
For a field extension $\mathbb{F}_q \subseteq \mathbb{L} \subseteq \overline{\mathbb{F}}_q$, let $h = h(x,y,z) \in \mathbb{L}(E_{a,d})$ be a $\mathbb{L}$-rational function of $E_{a,d}$. We have seen in Section 1.2.1 that the field $\mathbb{L}(E_{a,d})$ is isomorphic to the field of affine $\mathbb{L}$-rational function $\mathbb{L}((E_{a,d})^*_z)$, via the isomorphism $\mathbb{L}(E_{a,d}) \ni h(x,y,z) \mapsto h(x,y,1) \in \mathbb{L}((E_{a,d})^*_z)$. So here we will write the rational function h in the affine form $h(x,y) = h(x,y,1) = r(x,y)/s(x,y)$, following the notation of Section 1.2.1.

By Theorem 33, $E_{a,d}$ is birationally equivalent over $\mathbb{F}_q$ to an elliptic curve E_M in Montgomery form, via the birational map $\Phi : E_M \longrightarrow E_{a,d}$. Notice that, for each $P \in E_{a,d} \setminus \{\Omega_1, \Omega_2\}$, we have $|\Phi^{-1}(P)| = 1$, while $|\Phi^{-1}(\Omega_1)| = |\Phi^{-1}(\Omega_2)| = 2$. We denote

$$\Phi([0,0,1]) = [0,-1,1] = \mathcal{O}',$$

$$\Phi^{-1}(\Omega_1) = \{[\alpha_1,0,1],[\alpha_2,0,1] \in E_M \text{ with } \alpha_i \neq 0\} = \{Q_1, Q_2\},$$

and

$$\Phi^{-1}(\Omega_2) = \{[-1,\sqrt{d},1],[-1,-\sqrt{d},1] \in E_M\} = \{Q_3, Q_4\}.$$

We recall that, by definition, divisors on an absolutely irreducible projective curve are divisors on its nonsingular model. Hence, in the case of $E_{a,d}$, divisors on $E_{a,d}$ are divisors $D = \sum_{P \in E_M} n_P P$ on E_M. Given the divisor D, we can take its image with respect to Φ, that is $\Phi(D) = \sum_{P \in E_M} n_P \Phi(P)$, to obtain a formal sum of points on the given curve $E_{a,d}$ (see Example 26). Moreover, for $h \in \overline{\mathbb{F}}_q(E_{a,d})$, its divisor $\mathrm{div}_{E_{a,d}}(h)$ is

$$\mathrm{div}_{E_{a,d}}(h) = \mathrm{div}_{E_M}(h \circ \Phi).$$

As before, we can take the image $\Phi(\mathrm{div}_{E_{a,d}}(h))$ to obtain a formal sum of points on $E_{a,d}$.

As we have already mentioned, the authors of [49] take elliptic curves E_W in short Weierstrass form defined over $\mathbb{F}_q$, and trace-zero points $P \in T_n \subseteq E_W(\mathbb{F}_{q^n})$. For such points, by definition of trace-zero, we have $P \oplus \varphi(P) \oplus \cdots \oplus \varphi^{n-1}(P) = \mathcal{O}$. By Theorem 28, this equality is equivalent to saying that there is a rational function $h \in \overline{\mathbb{F}}_q(E_W)$ such that $\mathrm{div}(h) = P + \varphi(P) + \cdots + \varphi^{n-1}(P) - n\mathcal{O}$. The authors of [49] observe that h is a polynomial of a particular form, that can be given via $n-1$ coefficients of $\mathbb{F}_q$. So they optimally represent the trace-zero point P via these $n-1$ coefficients.
Let now $P \in T_n$ be a trace-zero point of our twisted Edwards curve $E_{a,d}$. Let $h \in \overline{\mathbb{F}}_q(E_{a,d})$ such that $\Phi(\mathrm{div}_{E_{a,d}}(h)) = P+\varphi(P)+\cdots\varphi^{n-1}(P)-n\mathcal{O}$. One has that h is not a polynomial any more. So the straightforward adaptation of the method in [49] cannot be applied. Our idea is then to take another rational function associated to $\mathrm{Tr}(P)$, rather than the rational function h. This rational function have the polynomial form that is required for the optimal representation. We describe and construct such rational function in the following theorem.

Theorem 56. *Let $E_{a,d}$ be a twisted Edwards curve defined over $\mathbb{F}_q$. Let $P \in T_n \subseteq E(\mathbb{F}_{q^n})$. Then there exists a polynomial $q_P(x,y) = q_1(y) + xq_2(y) \in \mathbb{F}_q[x,y]$, with $q_1(y), q_2(y) \in \mathbb{F}_q[y]$, such that*

1. $\Phi(\mathrm{div}_{E_{a,d}}(q_P)) = P + \varphi(P) + \ldots + \varphi^{n-1}(P) + \mathcal{O}' - 2\Omega_1 - (n-1)\Omega_2$.
2. $\max\{\deg(q_1), \deg(q_2)\} = \frac{n-1}{2}$.
3. $q_1(y) = (1+y)\hat{q_1}(y)$, *where* $\hat{q_1} \in \mathbb{F}_q[y]$ *and* $\deg(\hat{q_1}) \leq \frac{n-3}{2}$.
4. q_2 *is not the zero polynomial.*

Proof. If $P = \mathcal{O}$, we define $q_P = x(1-y)^{\frac{n-1}{2}}$, for which points $1, 2, 3, 4$ of the theorem are easily verified.
Let now $P = (\overline{x}, \overline{y}) \neq \mathcal{O}$. For each $1 \leq i \leq n$, let $P_i = \varphi^{i-1}(P)$. For each $1 \leq i \leq n-2$, let $\phi_i(x,y) = 0$ be the conic passing through the points $(P_1 \oplus \cdots \oplus P_{i-1} \oplus P_i)$, P_{i+1}, $(-(P_1 \oplus \cdots \oplus P_i \oplus P_{i+1}))$, $\mathcal{O}' = [0,-1,1]$, Ω_1 and Ω_2. Notice that ϕ_i exists by [2, Theorem

1 and Theorem 2]. Moreover, this conic is unique up to multiplication by a constant, and it is of the form

$$\phi_i(x,y) = B_i(y)x + A_i(y),$$

where $A_i(y)$ and $B_i(y)$ are polynomials of $\overline{\mathbb{F}}_q[y]$, of degree at most one. Furthermore, for $\phi_i \in \overline{\mathbb{F}}_q(E_{a,d})$, we have that

$$\mathrm{div}_{E_{a,d}}(\phi_i) = \mathrm{div}_{E_M}(\phi_i \circ \Phi) = \Phi^{-1}(P_1 \oplus + \cdots + P_{i-1} \oplus P_i) + \Phi^{-1}(P_{i+1}) + \tag{2.17}$$

$$+\Phi^{-1}(-(P_1 \oplus \cdots \oplus P_i \oplus P_{i+1})) + [0,0,1] - Q_1 - Q_2 - Q_3 - Q_4.$$

For each $1 \leq i \leq n-3$, let $h_i(y) = 0$ be the horizontal line through the point $P_1 \oplus \cdots \oplus P_{i+1} \in E_{a,d}$. Then

$$\mathrm{div}_{E_{a,d}}(h_i) = \mathrm{div}_{E_M}(h_i \circ \Phi) = \Phi^{-1}((P_1 \oplus \cdots \oplus P_i \oplus P_{i+1})) + \tag{2.18}$$

$$+\Phi^{-1}(-(P_1 \oplus \cdots \oplus P_i \oplus P_{i+1})) - Q_3 - Q_4.$$

Moreover, we have that

$$h(y) = \prod_{i=1}^{n-3} h_i \in \overline{\mathbb{F}}_q(E_{a,d})$$

is a polynomial of degree $n-3$. Notice that we have

$$\mathrm{div}_{E_{a,d}}(a - dy^2) = \mathrm{div}_{E_{a,d}}(((1-y^2)/x^2) \circ \Phi) = \mathrm{div}_{E_M}(y^2/x(x+1)^2) = 2Q_1 + 2Q_2 - 2Q_3 - 2Q_4, \tag{2.19}$$

and

$$\mathrm{div}_{E_{a,d}}(1+y) = \mathrm{div}_{E_M}((1-y^2) \circ \Phi) = \mathrm{div}_{E_M}(x/(x+1)) = 2[0,0,1] - Q_3 - Q_4.$$

Let now define

$$\tilde{q}_P(x,y) = \frac{(a-dy^2)^{\frac{n-3}{2}}}{h(y)(1+y)^{\frac{n-3}{2}}} \prod_{i=1}^{n-2} \phi_i \in \overline{\mathbb{F}}_q(E_{a,d}). \tag{2.20}$$

By (2.17), (2.18), and the previous observations, we have that

$$\mathrm{div}_{E_{a,d}}(\tilde{q}_P) = \mathrm{div}_{E_M}(q_P \circ \Phi) = (\sum_{i=1}^{n} \Phi^{-1}(P_i)) + [0,0,1] - Q_1 - Q_2 - (\frac{n-1}{2})Q_3 - (\frac{n-1}{2})Q_4. \tag{2.21}$$

Now let $(\tilde{q}_P \circ \Phi)(x,y) = p_1(x,y)/q_1(x,y)$. Denote by $p_1^{\varphi}(x,y)$, $q_1^{\varphi}(x,y)$ the polynomials that we obtain from p_1 and q_1 respectively, by rising each coefficient to the q-th power. We have that

$$\mathrm{div}_{E_M}(p_1^{\varphi}/q_1^{\varphi}) = \varphi(\mathrm{div}_{E_M}(p_1/q_1)) =$$

$$(\textstyle\sum_{i=1}^{n} \Phi^{-1}(\varphi(P_i))) + \varphi([0,0,1]) - \varphi(Q_1) - \varphi(Q_2) - (\frac{n-1}{2})\varphi(Q_3) - (\frac{n-1}{2})\varphi(Q_4) =$$

$$(\textstyle\sum_{i=1}^{n} \Phi^{-1}(P_i)) + [0,0,1] - Q_1 - Q_2 - (\frac{n-1}{2})Q_3 - (\frac{n-1}{2})Q_4 = \mathrm{div}_{E_M}(p_1/q_1) = \mathrm{div}_{E_M}(\tilde{q}_P \circ \Phi).$$

Notice that $Q_1, Q_2, Q_3, Q_4 \in E_M(\mathbb{F}_{q^2})$. Hence for Q_1, Q_2, we have that $\varphi(Q_1) = Q_1$, $\varphi(Q_2) = Q_2$ if $Q_1, Q_2 \in E_M(\mathbb{F}_q)$, $\varphi(Q_1) = Q_2$, $\varphi(Q_2) = Q_1$ otherwise. We have the analogous result for Q_3, Q_4. So we have obtained that $\mathrm{div}_{E_M}(p_1^{\varphi}/q_1^{\varphi}) = \mathrm{div}_{E_M}(p_1/q_1) = \mathrm{div}_{E_M}(\tilde{q}_P \circ \Phi)$. By [44, Proposition 2, Chapter 8], this implies that $p_1^{\varphi}/q_1^{\varphi} = p_1/q_1$ up to

multiplication by a nonzero constant. Hence (up to multiplication by a nonzero constant) we have that $p_1/q_1 = \tilde{q}_P \circ \Phi$ has coefficients in $\mathbb{F}_q$, that is, $\tilde{q}_P \circ \Phi \in \mathbb{F}_q(E_M)$. Since Φ is a $\mathbb{F}_q$-birational map, this implies that also $\tilde{q}_P \in \mathbb{F}_q(E_{a,d})$.

Take now the product $\prod_{i=1}^{n-2} \phi_i$ in the numerator of $\tilde{q}_P$. This product has the form

$$\prod_{i=1}^{n-2} \phi_i = H_{n-2}(y)x^{n-2} + H_{n-3}(y)x^{n-3} + \ldots + H_1(y)x + H_0(y),$$

where each $H_i(y)$ is a polynomial in y of degree at most $n-2$. Then, if we take the equality $x^2 = (1-y^2)/(a-dy^2)$ in $\overline{\mathbb{F}}_q(E_{a,d})$, we obtain the following equality of rational functions:

$$(a-dy^2)^{\frac{n-3}{2}} \prod_{i=1}^{n-2} \phi_i(x,y) = R_1(y) + xR_2(y), \tag{2.22}$$

where each $R_i(y)$ is a polynomial of degree

$$\deg(R_i) \leq \max\{\deg(H_j)\} + n - 3 \leq 2n - 5. \tag{2.23}$$

By definition of ϕ_i and h_j for each i, j, we have that, for each $Q = (x_0, y_0)$ such that $h(y_0) = 0$, the following equalities hold:

$$R_1(y_0) + x_0 R_2(y_0) = 0 \text{ and } R_1(y_0) - x_0 R_2(y_0) = 0. \tag{2.24}$$

Moreover, we claim that $x_0 \neq 0$. Let us prove the claim.

Proof of the claim. By definition of $h(y)$, we have that $Q = (x_0, y_0) = \pm(P \oplus \cdots \oplus \varphi^t(P))$ for some $0 \leq t \leq n-3$. So $x_0 = 0$ is equivalent to saying that $P \oplus \cdots \oplus \varphi^t(P) = \mathcal{O}'$ or $P \oplus \cdots \oplus \varphi^t(P) = \mathcal{O}$ for some $0 \leq t \leq n-3$.
We have that $\sum_{i=0}^{t} \varphi^i(P) \neq \mathcal{O}'$ for all $0 \leq t \leq n-3$, since $\mathcal{O}' = (0,-1) \notin T_n$.
Now we prove that we have also $\sum_{i=0}^{t} \varphi^i(P) \neq \mathcal{O}$ for all $0 \leq t \leq n-3$.
Suppose that it is not the case, that is, suppose that there exists $0 \leq t \leq n-3$ such that $\sum_{i=0}^{t} \varphi^i(P) = \mathcal{O}$. If $t = 0$, then $P = \mathcal{O}$. This is not possible since we are supposing $P \neq \mathcal{O}$.
Let now $t > 0$. We want to prove that, in this case, there exists $0 \leq j < t$ such that $\sum_{i=0}^{j} \varphi^i(P) = \mathcal{O}$.
If t is odd, that is $t = 2j+1$ with $0 \leq j < t$, we have that $Q = P \oplus \cdots \oplus \varphi^j(P) = \mathcal{O}$. In fact, $\sum_{i=0}^{t} \varphi^i(P) = \mathcal{O}$ implies $Q = -\varphi^{j+1}(Q)$, with $j+1 < n$ and $Q \in T_n \subseteq E(\mathbb{F}_{q^n})$. The equality $Q = -\varphi^{j+1}(Q)$ implies $Q \in E(\mathbb{F}_{q^{2(j+1)}})$, hence $Q \in E(\mathbb{F}_{q^{2(j+1)}}) \cap E(\mathbb{F}_{q^n}) = E(\mathbb{F}_q)$, since n is an odd prime and $j+1 < n$. Therefore, $Q = -\varphi^{j+1}(Q) = -Q$, that is $2Q = \mathcal{O}$. On the other hand $Q \in T_n$ with n odd prime, hence we have that $Q = \mathcal{O}$.
Let now assume that t is an even positive number. We have that $P \in T_n$ together with $\sum_{i=0}^{t} \varphi^i(P) = \mathcal{O}$ imply $\varphi^{t+1} \oplus \cdots \oplus \varphi^{n-1}(P) = \mathcal{O}$. This implies $\sum_{i=0}^{h} \varphi^i(P) = \mathcal{O}$, where $h = n-t-2$ is odd since n is odd and t even, and $0 < h \leq n-3$. Hence $h = 2j'+1$ with $0 \leq j' < h$, and $\sum_{i=0}^{j'} \varphi^i(P) = \mathcal{O}$ from the previous argument.
Now, if $j' < t$, we take $j = j'$. Otherwise, we have $j' \geq t$, and we have the equality $j'+1 = (t+1)k + r$ for some $k \in \mathbb{Z}_{>0}$ and some $0 \leq r < (t+1)$.
Suppose $r = 0$. Then $n-t-2 = h = 2j'+1$ implies $n = (t+1) + (j'+1) + (j'+1) = (t+1)(1+2k)$, with $t > 0$. This is not possible since n is prime.
Then, $0 < r < t+1$, from which $0 \leq r-1 < t$. Moreover, combining the equalities $\sum_{i=0}^{t} \varphi^i(P) = \mathcal{O}$ and $\sum_{i=0}^{j'} \varphi^i(P) = \mathcal{O}$ we have $\sum_{i=0}^{r-1} \varphi^i(P) = \mathcal{O}$. Hence, if $j' \geq t$, we take $j = r-1$.

In this way we have proved that, if $t > 0$, then there exists $0 \leq j < t$ such that $\sum_{i=0}^{j} \varphi^i(P) = \mathcal{O}$.
By iterating the same argument, we obtain a strictly decreasing chain $0 \leq \cdots < j_2 < j_1 < t$ such that $\sum_{i=0}^{j_s} \varphi^i(P) = \mathcal{O}$ for all j_s. This implies that there exists some $j_{s_0} = 0$. Then $P = \mathcal{O}$ and we have finished the proof of the claim.

From the precedent claim and (2.24) we have

$$h(y)|R_1(y) \quad \text{and} \quad h(y)|R_2(y). \tag{2.25}$$

Now, by definition of ϕ_i, we have that $\mathrm{ord}_{\mathcal{O}'}(R_1 + xR_2) = n - 2$. This implies that

$$(1+y)^{\frac{n-1}{2}}|R_1(y) \quad \text{and} \quad (1+y)^{\frac{n-3}{2}}|R_2(y), \tag{2.26}$$

and

$$(1+y)^{\frac{n-1}{2}} \not| R_2(y) \tag{2.27}$$

From (2.25) and (2.26) we obtain that $R_1(y) + xR_2(y)$ has the form

$$R_1(y) + xR_2(y) = (1+y)^{\frac{n-3}{2}} h(y)(1+y)\hat{q}_1(y) + (1+y)^{\frac{n-3}{2}} h(y) q_2(y),$$

where $\hat{q}_1(y)$, $q_2(y) \in \mathbb{F}_q[y]$ and $q_2(-1) \neq 0$. We denote $q_1(y) = (1+y)\hat{q}_1(y)$ and $q_P(x,y) = q_1(y) + xq_2(y)$. Therefore, from the definition (2.20) of $\tilde{q}_P$ and the equality of rational functions (2.22), and simplifying the denominator, we obtain the equality of rational functions

$$\tilde{q}_P(x,y) = q_P(x,y) \in \mathbb{F}_q(E_{a,d}).$$

The equality implies $\mathrm{div}_{E_{a,d}}(q_P) = \mathrm{div}_{E_{a,d}}(\tilde{q}_P)$, and, by (2.21),

$$\Phi(\mathrm{div}_{E_{a,d}}(q_P)) = P \oplus \cdots \oplus \varphi^{n-1}P + \mathcal{O}' - 2\Omega_1 - (n-1)\Omega_2.$$

Hence we have proved that $q_P(x,y) = q_1(y) + xq_2(y) \in \mathbb{F}_q[x,y]$ satisfies points 1, 3, 4 of the theorem. Notice that point 4 is a straightforward consequence of the fact that $q_2(-1) \neq 0$ by (2.27).

Now we show point 2 to finish the proof. From the previous discussion and (2.23), we have

$$\deg(q_i) = \deg(R_i) - \deg(1+y)^{\frac{n-3}{2}} - \deg(h) \leq 2n - 5 - \frac{(n-3)}{2} - (n-3) = \frac{n-1}{2} \text{ for } i \in \{1,2\}. \tag{2.28}$$

Observe that we have the following equality of rational functions of $\mathbb{F}_q(E_{a,d})$:

$$q_P(x,y)q_P(-x,y) = q_1^2(y) - \frac{1-y^2}{a-dy^2}\, q_2^2(y). \tag{2.29}$$

Moreover, we have that $\mathrm{div}_{E_{a,d}}(q_P(-x,y)) = \mathrm{div}_{E_M}(q_P(-x,y) \circ \Phi) = \mathrm{div}_{E_M}(q_P \circ \Phi(x,-y)) = -\mathrm{div}_{E_M}(q_P \circ \Phi) = -\mathrm{div}_{E_{a,d}}(q_P)$, from which

$$q_{-P}(x,y) = q_P(-x,y), \tag{2.30}$$

up to multiplication by a nonzero constant.
We take the polynomial

$$R_P(y) = (a - dy^2)q_1^2(y) - (1-y^2)q_2^2(y) \in \mathbb{F}_q[y].$$

By (2.19), (2.29) and (2.30), we have that

$$\Phi(\mathrm{div}_{E_{a,d}}(R_P)) = (\pm P) + (\pm\varphi(P)) + \cdots + (\pm\varphi^{n-1}(P)) + 2\mathcal{O}' - 2(n+1)\Omega_2.$$

Hence $(1+y)\prod_{i=0}^{n-1}(y-\overline{y}^{q^i})|R_P(y)$ (recall that $\overline{y}$ is the y-coordinate of P). Therefore, we have

$$n+1 \leq \deg(R_P(y)) \leq 2 + 2\max\{\deg(q_1), \deg(q_2)\}. \tag{2.31}$$

So part 2 follows directly from (2.28) and (2.31).

Notice that we have also obtained that R_P is a polynomial of degree exactly $n+1$ with coefficients in $\mathbb{F}_q$ and roots -1, $\overline{y}^{q^i}$, for $0 \leq i \leq n-1$. We need this result for the decompression algorithm: see Proposition 59. □

Computation of q_P. In the proof of the previous theorem, we have seen that one can compute the polynomial q_P in the following way.

1. Compute the fraction $\tilde{q}_P$ as

$$\tilde{q}_P(x,y) = \frac{(a-dy^2)^{\frac{n-3}{2}}}{h(y)(1+y)^{\frac{n-3}{2}}} \prod_{i=1}^{n-2} \phi_i, \tag{2.32}$$

where:

- For each $1 \leq i \leq n$, $P_i = \varphi^{i-1}(P)$.
- For each $1 \leq i \leq n-2$, $\phi_i(x,y) = 0$ is the conic through $(P_1 \oplus \ldots \oplus P_{i-1} \oplus P_i)$, P_{i+1}, $\mathcal{O}'$, Ω_1 and Ω_2.
- For each $1 \leq i \leq n-3$, $h_i(y) = 0$ is the horizontal line through $P_1 \oplus \cdots \oplus P_{i+1} \in E_{a,d}$.

Notice that we can easily calculate ϕ_i for each i, with the formulas given in [2, Theorem 1 and Theorem 2].

2. Reduce $\tilde{q}_P$ modulo the curve equation $x^2 = (1-dy^2)/(a-dy^2)$: replace each x^{2k} with $((1-y^2)/(a-dy^2))^k$, to obtain q_P up to multiplication by a nonzero constant.

We now discuss how to use the polynomial q_P to represent P via $(n-1)$ elements of $\mathbb{F}_q$ plus a bit. As a consequence of Theorem 56, q_P has the form

$$q_P(x,y) = (1+y)\left(a_{\frac{n-3}{2}} y^{\frac{n-3}{2}} + \cdots + a_1 y + a_0\right) + x\left(b_{\frac{n-1}{2}} y^{\frac{n-1}{2}} + \cdots + b_1 y + b_0\right), \tag{2.33}$$

where $a_i, b_j \in \mathbb{F}_q$ for all i, j, and $b_{\frac{n-1}{2}} \in \{0,1\}$. We have therefore obtained an optimal representation $\mathcal{R}_n = (\mathcal{R}_{n,q,E_{a,d}})$ for the trace-zero family $\mathcal{G}_n$, where

$$\mathcal{R}_{n,q,E_{a,d}} : T_n \longrightarrow \mathbb{F}_q^{n-1} \times \mathbb{F}_2,\ P \mapsto \left(a_0, \ldots, a_{\frac{n-3}{2}}, b_0, \ldots, b_{\frac{n-1}{2}}\right). \tag{2.34}$$

We now give the complete algorithm for point compression.

Algorithm 5 (Compression).

Input : $P \in T_n \subseteq E(\mathbb{F}_{q^n})$
Output : $\mathcal{R}_{n,q,E_{a,d}}(P) \in \mathbb{F}_q^{n-1} \times \mathbb{F}_2$

1: Compute $q_P(x,y) = q_1(y) + xq_2(y)$:
Use (2.32) and reduce modulo the curve equation, as explained above.
2: Compute $\hat{q_1}(y) = q_1(y)/(1+y) = a_{\frac{n-3}{2}} y^{\frac{n-1}{2}} + \cdots + a_1 y + a_0$
3: $q_2(y) = b_{\frac{n-1}{2}} y^{\frac{n-1}{2}} + \cdots + b_1 y + b_0$
4: $\mathcal{R}_{n,q,E_{a,d}}(P) \leftarrow (a_0, \ldots, a_{\frac{n-3}{2}}, b_0, \ldots, b_{\frac{n-1}{2}})$
5: **return** $\mathcal{R}_{n,q,E_{a,d}}(P)$

Theorem 57. *Algorithm* 5 *is correct.*

Proof. The thesis is a direct consequence of the previous results. □

Given a n-tuple $(\alpha_1, \ldots, \alpha_{n-1}, b) \in \mathbb{F}_q^{n-1} \times \mathbb{F}_2$ such that $(\alpha_1, \ldots, \alpha_{n-1}, b) = \mathcal{R}_{n,q,E_{a,d}}(P)$ for some $P \in T_n$, we want to compute the decompression $\mathcal{R}^{-1}_{n,q,E_{a,d}}(\alpha_1, \ldots, \alpha_{n-1}, b)$. We start with some preliminary results. The next lemma guarantees that the x-coordinate of P can be computed from its y-coordinate and the polynomial q_P.

Lemma 58. *Let $P = (\overline{x}, \overline{y}) \in T_n$. Let $q_P(x,y) = q_1(y) + xq_2(y) \in \mathbb{F}_q[x,y]$ be the polynomial of Theorem 56, with $\Phi(\mathrm{div}_{E_{a,d}}(q_P)) = P + \varphi(P) + \ldots + \varphi^{n-1}(P) + \mathcal{O}' - 2\Omega_1 - (n-1)\Omega_2$. Then $q_2(\overline{y}) = 0$ if and only if $P = \mathcal{O}$.*

Proof. If $q_2(\overline{y}) = 0$, then $q_1(\overline{y}) = 0$. Hence $q_P(-\overline{x}, \overline{y}) = 0$. Since the affine points of the curve on which q_P vanishes are exactly $\mathcal{O}'$ and $\varphi^i(P)$ for $0 \le i \le n-1$ by Theorem 56, and $\mathcal{O}' \notin T_n$, then $-P = \varphi^i(P)$ for some i. If $i = 0$, we have $-P = P$, hence $P = \mathcal{O}$. If $i \neq 0$, then $(-\overline{x}, \overline{y}) = (\overline{x}^{q^i}, \overline{y}^{q^i})$ for some $i \in \{1, \ldots, n-1\}$. Then $\overline{y} \in \mathbb{F}_{q^i} \cap \mathbb{F}_{q^n} = \mathbb{F}_q$ and $\overline{x}^{q^{2i}} = \overline{x} \in \mathbb{F}_{q^{2i}} \cap \mathbb{F}_{q^n} = \mathbb{F}_q$. Hence $P \in E(\mathbb{F}_q)$ and $-P = \varphi^i(P) = P$, from which $P = \mathcal{O}$.

Conversely, if $P = \mathcal{O}$ then $q_P(x,y) = x(1-y)^{\frac{n-1}{2}}$ and $q_2(1) = 0$. □

Given $q_P(x,y)$, we can compute a polynomial $Q_P(y)$ whose roots are exactly the Frobenius conjugates of the y-coordinate of P. This polynomial is used in our decompression algorithm.

Proposition 59. *Let $P = (\overline{x}, \overline{y}) \in T_n$. Let $q_P(x,y) = q_1(y) + xq_2(y) = (1+y)\hat{q_1}(y) + xq_2(y) \in \mathbb{F}_q[x,y]$ be the polynomial of Theorem 56, with $\Phi(\mathrm{div}_{E_{a,d}}(q_P)) = P + \varphi(P) + \ldots + \varphi^{n-1}(P) + \mathcal{O}' - 2\Omega_1 - (n-1)\Omega_2$.*

1. *Let*
$$R_P = (a - dy^2)q_1^2(y) - (1-y^2)q_2^2(y).$$
Then $R_P \in \mathbb{F}_q[y]$, $\deg(R_P) = n+1$, and its roots are $-1, \overline{y}, \overline{y}^q, \ldots, \overline{y}^{q^{n-1}}$.

2. *Let*
$$Q_P(y) = (a - dy^2)(1+y)\hat{q_1}^2(y) + (y-1)q_2^2(y).$$
Then $Q_P(y) \in \mathbb{F}_q[y]$, $\deg(Q_P) = n$, and its roots are $\overline{y}, \overline{y}^q, \ldots, \overline{y}^{q^{n-1}}$.

Proof. We have proved point 1 in the last part of the proof of Theorem 56. We have that $R_P = (1+y)Q_P$. Then point 2 follows from point 1. □

We give below the decompression algorithm.

Algorithm 6 (Decompression).

Input : $(\alpha_1, \ldots, \alpha_{n-1}, b) \in \mathbb{F}_q^{n-1} \times \mathbb{F}_2$
Output : $P = (\overline{x}, \overline{y}) \in T_n$ with $\mathcal{R}(P) = (\alpha_1, \ldots, \alpha_{n-1}, b)$

1: $\hat{q}_1(y) \leftarrow \alpha_{\frac{n-1}{2}} y^{\frac{n-3}{2}} + \cdots + \alpha_2 y + \alpha_1$.
2: $q_2(y) \leftarrow b y^{\frac{n-1}{2}} + \alpha_{n-1} y^{\frac{n-3}{2}} + \cdots + \alpha_{\frac{n+3}{2}} y + \alpha_{\frac{n+1}{2}}$.
3: $Q_P(y) \leftarrow (a - dy^2) \cdot (1+y) \cdot \hat{q}_1^2(y) + (y-1) \cdot q_2^2(y)$.
4: $\overline{y} \leftarrow$ one root of $Q_P(y)$.
5: **if** $\overline{y} = 1$ **then** $\overline{x} \leftarrow 0$ **else** $\overline{x} \leftarrow -\frac{\hat{q}_1(\overline{y})(\overline{y}+1)}{q_2(\overline{y})}$ **end if**
6: **return** $(\overline{x}, \overline{y})$.

Theorem 60. *Algorithm* 6 *is correct.*

Proof. Let $P \in T_n$ be a point with $\mathcal{R}(P) = (\alpha_1, \ldots, \alpha_{n-1}, b)$. By Theorem 56, the Frobenius conjugates of P are the only other points of T_n with the same representation. Correctness of the first four lines of the algorithm follows from Proposition 59. Correctness of line 5 follows from Lemma 58. Hence the given algorithm correctly recovers the point P, up to Frobenius conjugates. □

2.2.1 Explicit equations, complexity, and timings for $n = 3$

In this subsection, we give explicit equations and perform some computations for $n = 3$. We estimate the number of operations needed for compression and decompression. We give some timings obtained with Magma. We also make comparisons with the analogous compression and decompression algorithms for degree three trace-zero subgroups of elliptic curves in short Weierstrass form, treated in [49].

Point Compression. Let $P = (\overline{x}, \overline{y}) \in T_3 \subseteq E(\mathbb{F}_{q^3})$. For brevity, we denote the representation map $\mathcal{R}_{3,q,E_{a,d}}$ of T_3 by $\mathcal{R}$. By Theorem 56, we may write

$$q_P(x,y) = \hat{q}_1(y)(1+y) + x q_2(y) = a_0(1+y) + x(b_1 y + b_0),$$

where a_0, $b_0 \in \mathbb{F}_q$, $b_1 \in \{0,1\}$.

If $P \notin E(\mathbb{F}_q)$, let $t = \frac{\overline{y}+1}{\overline{x}}$. Notice that $\overline{x} \neq 0$, since $\overline{x} = 0$ implies $P = \mathcal{O}$, hence $P \in E(\mathbb{F}_q)$.

1. If $t^q - t \neq 0$, by [2, Theorem 1], we have

$$\mathcal{R}(P) = (a_0, b_0, b_1) = \left(-\frac{\overline{y}^q - \overline{y}}{t^q - t}, -a_0 t - \overline{y}, 1 \right).$$

Computing t from $\overline{x}$ and $\overline{y}$ takes 1M+1I in $\mathbb{F}_{q^3}$. Once we have t, the situation is analogous to the case of elliptic curves in short Weierstrass form. Hence we refer to [49, Section

5.1] for a detailed discussion of how to efficiently compute $\mathcal{R}(P)$. It is shown that one can compute a_0 and b_0 with 2S+6M +1I in $\mathbb{F}_q$. Summarizing, point compression in this case takes 1M+1I in $\mathbb{F}_{q^3}$ and 2S+6M +1I in $\mathbb{F}_q$. Due to the calculation of t, it is more expensive than that for elliptic curves in short Weierstrass form.

2. If $t^q - t = 0$, then $q_P(x, y) = 0$ is the line passing through P and $\mathcal{O}'$, by [2, Theorem 1]. Hence

$$\mathcal{R}(P) = \left(-t^{-1}, 1, 0\right). \tag{2.35}$$

Since $\mathcal{O}' \notin T_3$, then $t \neq 0$. In this case, point compression requires only 1M + 1I in $\mathbb{F}_{q^3}$.

If $P \in E(\mathbb{F}_q)$, then the computation takes place in $\mathbb{F}_q$ instead of $\mathbb{F}_{q^3}$. Hence we expect the complexity to be lower. We carry on a precise operation count, as in the previous case.

3. If $d\overline{x}^2\overline{y} - 1 \neq 0$, by [2, by Theorem 1], we have

$$\mathcal{R}(P) = \left(\frac{\overline{x}(1-\overline{y})}{d\overline{x}^2\overline{y}-1}, \frac{\overline{y} - a\overline{x}^2}{d\overline{x}^2\overline{y}-1}, 1\right).$$

Therefore, point compression takes 1S+4M+1I in $\mathbb{F}_q$.

4. If $d\overline{x}^2\overline{y} - 1 = 0$, then the situation is analogous to **2.** and $\mathcal{R}(P)$ is given by (2.35). Hence point compression requires 1M + 1I in $\mathbb{F}_q$.

Since **1.** is the generic case, the expected complexity of point compression is 1M+1I in $\mathbb{F}_{q^3}$ and 2S+6M +1I in $\mathbb{F}_q$.

Point Decompression. Let $(\alpha_1, \alpha_2, b) \in \mathbb{F}_q^2 \times \mathbb{F}_2$ and $P = (\overline{x}, \overline{y}) \in T_3$ such that $\mathcal{R}(P) = (\alpha_1, \alpha_2, b)$. In order to recover P from $\mathcal{R}(P)$, we want to find the roots of

$$Q_P(y) = (b - d\alpha_1^2)y^3 + (-d\alpha_1^2 + 2\alpha_2 b - b)y^2 + (a\alpha_1^2 - 2\alpha_2 b + \alpha_2^2)y + (a\alpha_1^2 - \alpha_2^2).$$

They are the solutions system

$$\begin{cases} y + y^q + y^{q^2} & = & c(d\alpha_1^2 - 2\alpha_2 b + b) \\ y^{q+1} + y^{q^2+1} + y^{q^2+q} & = & c(a\alpha_1^2 - 2\alpha_2 b + \alpha_2^2) \\ y^{1+q+q^2} & = & c(-a\alpha_1^2 + \alpha_2^2) \end{cases} \tag{2.36}$$

where $c = (b - d\alpha_1^2)^{-1}$. Notice that $(b - d\alpha_1^2) \neq 0$, since Q_P has degree 3 by Proposition 59.

Computing the constant terms of (2.36) takes 2S+3M+1I in $\mathbb{F}_q$. Computing a solution of the system takes at most 3S+3M+1I, one square root and two cube roots in $\mathbb{F}_q$, as shown in [49]. Finally, computing $\overline{x}$ from $\overline{y}$ requires 2M+1I in $\mathbb{F}_{q^3}$. Summarizing, for $n = 3$ point decompression takes at most 2M+1I in $\mathbb{F}_{q^3}$ and 5S+6M+2I, one square root and two cube roots in $\mathbb{F}_q$. It is more expensive than that for elliptic curves in short Weierstrass form, which takes at most 1M in $\mathbb{F}_{q^3}$ and 5S+4M+1I, one square root and two cube roots in $\mathbb{F}_q$.

We now give an example and some statistics implemented in Magma, version V2.22-1 of the software, running on a single 3 GHz core. We follow the same setup as in Example 53. We compare with the method for elliptic curves in short Weierstrass form proposed in [49].

Example 61. Let $q = 2^{79} - 67$ and $\mu = 3$. We choose random, birationally equivalent curves defined over $\mathbb{F}_q$:

$$E_{a,d} : 31468753957068040687814x^2 + y^2 = 1 + 1926978212766389664989 97x^2y^2$$

and

$$E_W : y^2 = x^3 + 292467848427659499478503x + 36136102673640400434542 1.$$

We choose a random point $P' \in E_W(\mathbb{F}_{q^3})$ of trace-zero, and let P be the corresponding point on $E_{a,d}$. For brevity, we only write the x-coordinates $x_{P'}$ of points of E and the y-coordinates y_P of points of $E_{a,d}$:

$x_{P'} = 34656092814607695931475 3\xi^2+4568265396285359810342 12\xi+3441674704030266528266 72,$

$y_P = 20852071389751823621596 6\xi^2+4511219445502199473688 11\xi+6804108986042990130625 2.$

We denote by $\mathcal{R}$ and $\mathcal{R}'$ the representation maps on $E_{a,d}$ and E, respectively. We compute:

$$\mathcal{R}'(P') = (\gamma_0, \gamma_1) = (48823870679406912678832, 28345175156076495772030 2),$$

$$\mathcal{R}(P) = (a_1, b_0, b_1) = (31308434255223282002781 6, 53581470317932429707416 1, 1).$$

Applying the decompression algorithms to $\mathcal{R}'(P')$ and $\mathcal{R}(P)$, we obtain

$$\mathcal{R}'^{-1}(48823870679406912678832, 28345175156076495772030 2) =$$

$\{34656092814607695931475 3\xi^2+4568265396285359810342 12\xi+3441674704030266528266 72,$

$16475949861450750318749 3\xi^2+3615206909881977515343 81\xi+3441674704030266528266 72,$

$93142483046730124850775\xi^2+3905785889978954421374 49\xi+3441674704030266528266 72\},$

which are the x-coordinates of P' and its Frobenius conjugates. Similarly

$$\mathcal{R}^{-1}(31308434255223282002781 6, 53581470317932429707416 1, 1) =$$

$\{20852071389751823621596 6\xi^2+4511219445502199473688 11\xi+6804108986042990130625 2,$

$53932153696106685501116 7\xi^2+2374313910976429683867 19\xi+6804108986042990130625 2,$

$46108356875604408347890 9\xi^2+5203724839667662589505 12\xi+6804108986042990130625 2\},$

which are the y-coordinates of P and its Frobenius conjugates.

We now give an estimate of the average time of compression and decompression for groups of different bit-length. We take primes q_1, q_2, and q_3 such that $3|(q_i - 1)$ for all i, of bit-length 96, 112, and 128, respectively. For each q_i, we take five pairs of birationally equivalent curves $(E, E_{a,d})$ defined over $\mathbb{F}_{q_i}$, such that the order of T_3 is prime of bit-length respectively 192, 224 and 256. On each pair of curves we randomly choose $20'000$ pairs of points (P', P) of trace-zero which correspond to each other via the birational map between the curves. For each pair of points, we compute $\mathcal{R}'(P'), \mathcal{R}(P), \mathcal{R}'^{-1}(\mathcal{R}'(P')), \mathcal{R}^{-1}(\mathcal{R}(P))$. For each computation, we take the average time in milliseconds for each curve, and then the averages over the five curves. The average computation times are given in the table below.

Table 5. Average computation times in milliseconds, for point compression and decompression in T_3, for elliptic curves in short Weierstrass form and for twisted Edwards curves.

Bit-length of $\lvert T_3\rvert$	192	224	256
Compression on E	0.015	0.013	0.011
Compression on $E_{a,d}$	0.034	0.037	0.035
Decompression on E	0.09	0.13	0.15
Decompression on $E_{a,d}$	0.14	0.19	0.20

The next table contains the ratios of the average times for point compression and decompression on elliptic curves in short Weierstrass form and twisted Edwards curves.

Table 6. Ratios between the average times for point compression and decompression on elliptic curves in short Weierstrass form and on twisted Edwards curves.

Bit-length of $\lvert T_3\rvert$	192	224	256
Comp on E / Comp on $E_{a,d}$	0.441	0.351	0.314
Dec on E / Dec on $E_{a,d}$	0.643	0.684	0.750

2.2.2 Explicit equations, complexity, and timings for $n = 5$

In this subsection we give explicit equations and perform computations for $n = 5$. We estimate the number of operations needed for the computations and give some timings obtained with Magma. We also make comparisons with the method proposed in [49] for elliptic curves in short Weierstrass form.

Point Compression. Let $P \in T_5 \subseteq E(\mathbb{F}_{q^5})$. For brevity, we denote the representation map $\mathcal{R}_{5,q,E_{a,d}}$ of T_5 by $\mathcal{R}$. By Theorem 56, q_P is of the form

$$q_P(x,y) = (1+y)\hat{q_1}(y) + xq_2(y) = (1+y)(a_1y + a_0) + x(b_2y^2 + b_1y + b_0),$$

where a_0, a_1, b_0, $b_1 \in \mathbb{F}_q$, and $b_2 \in \mathbb{F}_2$. By (2.32), we have that

$$(1+y)h_1h_2q_P = \phi_1\phi_2\phi_3(a - dy^2)$$

modulo the equation of $E_{a,d}$, and up to a nonzero constant factor. We focus on the generic case, where $b_2 = 1$ and ϕ_i is of the form

$$\phi_i(x,y) = p_i(y+1) + x(y+q_i)$$

with p_i, $q_i \in \mathbb{F}_{q^5}$, and $i \in \{1,2,3\}$. Denote by k_1 and k_2 the y-coordinates of $P_1 \oplus P_2$ and $P_1 \oplus P_2 \oplus P_3$, respectively. We have

$$\mathcal{R}(P) = (a_0, a_1, b_0, b_1, 1),$$

where

$$\begin{aligned}
a_1 &= k \cdot (d(p_1p_2p_3) + (p_1 + p_2 + p_3)),\\
a_0 &= k \cdot (3d(p_1p_2p_3) + (p_1q_2 + p_1q_3 + q_1p_2 + q_1p_3 + p_2q_3 + q_2p_3) + (p_1 + p_2 + p_3)) +\\
&\quad a_1 \cdot (k_1 + k_2 - 2),\\
b_1 &= k \cdot (d(p_1p_2q_3 + p_1p_3q_2 + p_2p_3q_1) + 2d(p_1p_2 + p_1p_3 + p_2p_3) + (q_1 + q_2 + q_3)) +\\
&\quad (k_1 + k_2 - 1),\\
b_0 &= k \cdot (2d(p_1p_2q_3 + p_1p_3q_2 + p_2p_3q_1) + (d - a)(p_1p_2 + p_1p_3 + p_2p_3) +\\
&\quad (q_1q_2 + q_1q_3 + q_2q_3)) - 1) + b_1(k_1 + k_2 - 1) + (k_1 + k_2 - k_1k_2),\\
k &= (d(p_1p_2 + p_1p_3 + p_2p_3) + 1)^{-1}.
\end{aligned}$$

Computing ϕ_1, ϕ_2, and ϕ_3 takes 2S+34M+2I in $\mathbb{F}_{q^5}$. Computing a_1, a_2, b_1, b_0 with the formulas above requires 45M+1I in $\mathbb{F}_{q^5}$. So point compression for $n = 5$ takes a total of 2S+79M+3I in $\mathbb{F}_{q^5}$. The method of [49] for elliptic curves in short Weierstrass form is less expensive, as it takes 3S+18M+3I in $\mathbb{F}_{q^5}$.

Point Decompression. Let $(\alpha_1, \alpha_2, \alpha_3, \alpha_4, b) \in \mathbb{F}_q^4 \times \mathbb{F}_2$ and let $P = (\overline{x}, \overline{y}) \in T_5$ such that $\mathcal{R}(P) = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, b)$. In order to decompress $\mathcal{R}(P)$, we look for the roots of

$$Q_P(y) = Q_5 y^5 + Q_4 y^4 + Q_3 y^3 + Q_2 y^2 + Q_1 y + Q_0,$$

where

$$\begin{array}{lcl} Q_0 & = & a\alpha_1^2 - \alpha_3^2, \\ Q_1 & = & a\alpha_1^2 + 2a\alpha_1\alpha_2 + \alpha_3^2 - 2\alpha_3\alpha_4, \\ Q_2 & = & -d\alpha_1^2 + 2a\alpha_1\alpha_2 + a\alpha_2^2 + 2\alpha_3\alpha_4 - 2\alpha_3 b - \alpha_4^2, \\ Q_3 & = & -d\alpha_1^2 - 2d\alpha_1\alpha_2 + a\alpha_2^2 + 2\alpha_3 b + \alpha_4^2 - 2\alpha_4 b, \\ Q_4 & = & -2d\alpha_1\alpha_2 - d\alpha_2^2 + 2\alpha_4 b - b, \\ Q_5 & = & -d\alpha_2^2 + b. \end{array}$$

This amounts to solving the system

$$\left\{ \begin{array}{lcl} e_1(y, y^q, \ldots, y^{q^4}) & = & -Q_5^{-1} Q_4 \\ e_2(y, y^q, \ldots, y^{q^4}) & = & Q_5^{-1} Q_3 \\ e_3(y, y^q, \ldots, y^{q^4}) & = & -Q_5^{-1} Q_2 \\ e_4(y, y^q, \ldots, y^{q^4}) & = & Q_5^{-1} Q_1 \\ e_5(y, y^q, \ldots, y^{q^4}) & = & -Q_5^{-1} Q_0 \end{array} \right.$$

where $e_i(y, y^q, \ldots, y^{q^4})$ is the i-th elementary symmetric polynomial in $y, y^q, \ldots, y^{q^4}$. Computing the constants in the system takes 4S+7M+1I in $\mathbb{F}_q$, while solving the system requires $O(\log_2 q)$ operations in $\mathbb{F}_q$ following the approach from [49]. Finally, recovering $\overline{x}$ from $\overline{y}$ takes 1S+5M+1I in $\mathbb{F}_{q^5}$. The computational cost of point decompression is comparable to that of the decompression algorithm from [49] for elliptic curves in short Weierstrass form.

In order to estimate the average time of compression and decompression for groups of different bit-length, we take primes q_1, q_2, and q_3 such that $5|(q_i - 1)$ for all i, of bit-length 48, 56, and 64, respectively. For each q_i, we take five pairs of birationally equivalent curves $(E, E_{a,d})$ defined over $\mathbb{F}_{q_i}$, such that the order of T_5 is prime of bit-length respectively 192, 224 and 256. On each pair of curves we randomly choose 20′000 pairs of points (P', P) of trace-zero which correspond to each other via the birational map between the curves. For each pair of points, we compute $\mathcal{R}'(P'), \mathcal{R}(P), \mathcal{R}'^{-1}(\mathcal{R}'(P')), \mathcal{R}^{-1}(\mathcal{R}(P))$. For each computation, we take the average time in milliseconds for each curve, and then the averages over the five curves. The average computation times are given in the table below.

Table 7. Average computation times in milliseconds, for point compression and decompression in T_5, for elliptic curves in short Weierstrass form and for twisted Edwards curves.

Bit-length of $\lvert T_5 \rvert$	192	224	256
Compression on E	1.566	1.725	1.894
Compression on $E_{a,d}$	1.704	1.868	2.052
Decompression on E	6.10	31.69	36.99
Decompression on $E_{a,d}$	6.15	31.37	36.59

The next table contains the ratios of the average times for point compression and decompression on elliptic curves in short Weierstrass form and twisted Edwards curves.

Table 8. Ratios between the average times for point compression and decompression on elliptic curves in short Weierstrass form and on twisted Edwards curves.

Bit-length of $\lvert T_5 \rvert$	192	224	256
Comp on E / Comp on $E_{a,d}$	0.919	0.923	0.923
Dec on E / Dec on $E_{a,d}$	0.992	1.010	1.011

Finally, Table 9 summarizes the number of operations for point compression and decompression. We compare the operation count from this paper with the one for elliptic curves in short Weierstrass form from [49].

Table 9. Number of operations for point compression and decompression.

Compression, $n = 3$, elliptic	2S+6M+1I in $\mathbb{F}_q$
Compression, $n = 3$, Edwards	1M+1I in $\mathbb{F}_{q^3}$ and 2S+6M+1I in $\mathbb{F}_q$
Decompression, $n = 3$, elliptic	1M in $\mathbb{F}_{q^3}$, 5S+4M+1I, 1 sq. root, 2 cube roots in $\mathbb{F}_q$
Decompression, $n = 3$, Edwards	2M + 1I in $\mathbb{F}_{q^3}$, 5S+6M+2I, 1 sq. root, 2 cube roots in $\mathbb{F}_q$
Compression, $n = 5$, elliptic	3S+18M+3I in $\mathbb{F}_{q^5}$
Compression, $n = 5$, Edwards	2S+79M+3I in $\mathbb{F}_{q^5}$
Decompression, $n = 5$, elliptic	$O(\log_2 q)$ operations in $\mathbb{F}_q$, 1S+3M+1I in $\mathbb{F}_{q^5}$
Decompression, $n = 5$, Edwards	$O(\log_2 q)$ operations in $\mathbb{F}_q$, 1S+5M+1I in $\mathbb{F}_{q^5}$

Chapter 3

Generalized summation polynomials

In this chapter, we deal with generalized summation polynomials of elliptic curves. First, we explain the idea on which generalized summation polynomials are based. Next, we give the definition, and we show that such polynomials exist, by giving algorithms to construct them for each choice of the parameters. We then provide a result on the degree of these polynomials, that will be used in Chapter 5. Finally, we briefly deal with some applications of generalized summation polynomials to cryptography. Such applications will be treated in details in the subsequent chapters.

Throughout the chapter, we take a finite field $\mathbb{K}$ of characteristic different from 2, 3, and we denote by $\overline{\mathbb{K}}$ its algebraic closure. We take an elliptic curve E defined over $\mathbb{K}$, written in short Weierstrass form

$$E : y^2z = f(x, z) = x^3 + Axz^2 + Bz^3.$$

We denote by $\oplus$ the operation of point addition on E. Moreover, we denote by $\mathcal{O}$ the neutral element of the operation. In Section 1.3.3, we mention that we drew inspiration from Semaev's summation polynomials to define generalized summation polynomials. We refer to this section for the definition of summation polynomials of elliptic curves, with annexed constructions, basic properties and notations. Our definition and constructions can be adapted to twisted Edwards curves, as in the case of classical summation polynomials.

3.1 A generalization of Semaev's summation polynomials

Let $f_{t+1}(x_1, \cdots, x_t, x_{t+1}) \in \mathbb{K}[x_1, \cdots, x_t, x_{t+1}]$ be the $(t+1)$-th summation polynomial of E, as in Theorem 37. For each $(\overline{x}_1, \cdots, \overline{x}_t) \in \overline{\mathbb{K}}^t$, take $f_{t+1}(\overline{x}_1, \cdots, \overline{x}_t, x) \in \overline{\mathbb{K}}[x, y]$, and the associated affine plane curve defined over $\overline{\mathbb{K}}$:

$$C : f_{t+1}(\overline{x}_1, \cdots, \overline{x}_t, x) = 0.$$

The polynomial $f_{t+1}(x_1, \cdots, x_{t+1})$ has degree 2^{t-1} in each variable by Theorem 37. Then the curve C is the union of at most 2^{t-1} vertical lines defined over $\overline{\mathbb{K}}$. Moreover, by definition of summation polynomials, we have that the affine points of intersection between this curve and the elliptic curve E are exactly the affine addition points $P_1 \oplus \cdots \oplus P_t$, for each $P_i = (\overline{x}_i, y_i) \in E$ on the vertical line

$$v_{-\overline{x}_i} : x - \overline{x}_i = 0,$$

for $i \in \{1, \cdots, t\}$. Our aim in defining generalized summation polynomials is to extend this property from vertical lines $v_{-\overline{x}_i}$ to general affine zero-locus of the elliptic curve.

For $n \in \mathbb{Z}_{\geq 2}$, we will define a polynomial

$$h(a_1, \cdots, a_{n-1})(x, y) \in \mathbb{K}[a_1, \cdots, a_{n-1}][x, y],$$

of a special form. This polynomial has the following property. For each $(\overline{a}_1, \cdots, \overline{a}_{n-1}) \in \overline{\mathbb{K}}^{n-1}$, we take the curve defined over $\overline{\mathbb{K}}$:

$$C_{\overline{a}_1, \cdots, \overline{a}_{n-1}} : h(\overline{a}_1, \cdots, \overline{a}_{n-1})(x, y) = 0. \tag{3.1}$$

Then this curve has exactly n affine points of intersection $P_1, \cdots, P_n$ with E.
Notice that $P_1, \cdots, P_n$ are not necessarily all distinct. This means that the intersection points are counted with multiplicity, according to the definition of intersection multiplicity that we give in Example 22.
For example, for $n = 2$, we define

$$h(a_1)(x, y) = x + a_1 \in \mathbb{K}[a_1][x, y].$$

Then, for each $\overline{a}_1 \in \overline{\mathbb{K}}$, we have that

$$C_{\overline{a}_1} = v_{\overline{a}_1} : x + \overline{a}_1 = 0$$

is the vertical line through the two affine points of E whose x-coordinate is $-\overline{a}_1$.
In the next subsection, we see how to define $h(a_1, \cdots, a_{n-1})(x, y) \in \mathbb{K}[a_1, \cdots, a_{n-1}][x, y]$ for each $n \in \mathbb{Z}_{\geq 2}$.

Now we fix parameters $t, n_1, \cdots, n_t \in \mathbb{Z}_{\geq 2}$. We want to define a polynomial

$$S = S_{t,n_1,\cdots,n_t}(a_{1,1}, \cdots, a_{1,n_1-1}, \cdots, a_{t,1}, \cdots, a_{t,n_t-1})(x, y) \in$$

$$\mathbb{K}[a_{1,1}, \cdots, a_{1,n_1-1}, \cdots, a_{t,1}, \cdots, a_{t,n_t-1}][x, y],$$

that satisfies the following property. We fix parameters

$$(\overline{a}_{1,1}, \cdots, \overline{a}_{1,n_1-1}, \cdots, \overline{a}_{t,1}, \cdots, \overline{a}_{t,n_t-1}) \in \overline{\mathbb{K}}^{\sum_{i=1}^{t}(n_i-1)}.$$

We take the curve defined over $\overline{\mathbb{K}}$:

$$S_{t,n_1,\cdots,n_t}(\overline{a}_{1,1}, \cdots, \overline{a}_{1,n_1-1}, \cdots, \overline{a}_{t,1}, \cdots, \overline{a}_{t,n_t-1})(x, y) = 0.$$

We have that, generically, the affine points of intersection between this curve and the elliptic curve E are exactly the affine addition points

$$P_1 \oplus \cdots \oplus P_t,$$

for each P_i affine point of intersection between E and the affine curve

$$C_{\overline{a}_{i,1},\cdots,\overline{a}_{i,n_i-1}} : h(\overline{a}_{i,1}, \cdots, \overline{a}_{i,n_i-1})(x, y) = 0 \text{ as in (3.1)}.$$

When we say that the mentioned property holds generically, we mean that it holds in a nonempty Zarisky open set

$$\mathcal{U} \subseteq \overline{\mathbb{K}}^{\sum_{i=1}^{t}(n_i-1)}.$$

So the property is verified for all $(\overline{a}_{1,1}, \cdots, \overline{a}_{1,n_1-1}, \cdots, \overline{a}_{t,1}, \cdots, \overline{a}_{t,n_t-1}) \in \mathcal{U}$. We refer to Remark 39 for the definition of Zarisky topology.

For example, let $n_i = 2$ for each $i = \{1, \cdots, t\}$. We have defined $h(a_{i,1})(x, y) = x + a_{i,1} \in \mathbb{K}[a_{i,1}][x, y]$. Then, for all $i \in \{1, \cdots, t\}$, $\overline{a}_{i,1} \in \overline{\mathbb{K}}$, we have the vertical line

$$C_{\overline{a}_{i,1}} = v_{\overline{a}_{i,1}} : x + \overline{a}_{i,1} = 0.$$

Now we take $f_{t+1}(x_1, \cdots, x_{t+1})$ the $(t+1)$-th Semaev's summation polynomial of E. We denote $a_{i,1} = -x_i$ for $i \in \{1, \cdots, t\}$, and $x = x_{t+1}$. Hence, by the previous discussion, we have that the polynomial

$$S_{t,2,\cdots,2}(a_{1,1}, \cdots, a_{t,1})(x, y) = f_{t+1}(-a_{1,1}, \cdots, -a_{t,1})(x) \in \mathbb{K}[a_{1,1}, \cdots, a_{t,1}](x, y)$$

satisfies the required property in $\mathcal{U} = \overline{\mathbb{K}}^t$.

From this point of view, the polynomials S described above can be seen as a generalization of Semaev's summation polynomials. This is the reason why we will call such polynomials generalized summation polynomials.

3.2 Definition of generalized summation polynomials

In the following, we use the definition of divisor of a rational function of a projective curve, that we give in Section 1.2. We follow the notation of this section. More precisely, we identify the field $\overline{\mathbb{K}}(E)$ of $\overline{\mathbb{K}}$-rational functions of E with the field $\overline{\mathbb{K}}(E_z^*)$ of $\overline{\mathbb{K}}$-rational functions of the affine dehomogenization E_z^* of E, with respect to the variable z. In fact, we saw in Section 1.2.1 that this fields are isomorphic, via the isomorphism $\overline{\mathbb{K}}(E) \ni h(x, y, z) \mapsto h(x, y, 1) \in \overline{\mathbb{K}}(E_z^*)$. Hence, we will write a $\overline{\mathbb{K}}$-rational function $h(x, y, z)$ of E in the affine form $h(x, y, 1)$, as we already did in Section 2 of Chapter 2.

We start with a proposition on zero-locus of the elliptic curve E. The proposition states that each curve which intersects the elliptic curve E in exactly n affine points can be given via $n-1$ coefficients of $\overline{\mathbb{K}}$. Vice versa, $n-1$ coefficients of $\overline{\mathbb{K}}$ define a curve that intersects E in exactly n affine points. This result tells us the form of the polynomial $h(a_1, \cdots, a_{n-1})(x, y) \in \mathbb{K}[a_1, \cdots, a_{n-1}][x, y]$ of the previous subsection, for each $n \in \mathbb{Z}_{\geq 2}$. Notice that we use a special case of the proposition below in Section 2 of Chapter 2, in order to represent a trace-zero point $P \in T_n \subseteq E(\mathbb{F}_{q^n})$ via $n-1$ coordinates of $\mathbb{F}_q$.

Proposition 62. *Let $n \in \mathbb{Z}_{\geq 2}$, $d_1 = \lfloor \frac{n}{2} \rfloor$, $d_2 = \lfloor \frac{n-3}{2} \rfloor$. Let $a = (a_1, \cdots, a_{n-1})$ be a vector of parameters. If $n = 2k$, let*

$$h_1(a)(x) = a_1 + a_2 x + \cdots + a_k x^{k-1} + x^k \in \mathbb{K}[a][x],$$

$$h_2(a)(x) = a_{k+1} + a_{k+2} x + \cdots + a_{n-1} x^{k-2} \in \mathbb{K}[a][x].$$

If $n = 2k+1$, let

$$h_1(a)(x) = a_1 + a_2 x + \cdots + a_{k+1} x^k \in \mathbb{K}[a][x],$$

$$h_2(a)(x) = a_{k+2} + \cdots + a_{n-1} x^{k-2} + x^{k-1} \in \mathbb{K}[a][x].$$

Let

$$h(a)(x, y) = h_1 + y h_2 \in \mathbb{K}[a][x, y]. \tag{3.2}$$

Let $P_i = (x_i, y_i) \in E$ for $i \in \{1, \cdots, n\}$, such that $P_1 \oplus \cdots \oplus P_n = \mathcal{O}$. Then there exists a unique $\overline{a} \in \overline{\mathbb{K}}^{n-1}$ such that

1. $\mathrm{div}(h(\overline{a})(x, y)) = \sum_{i=1}^{n} P_i - n\mathcal{O}$.

Moreover, we have that

2. $h_1(\overline{a})(x)^2 - f(x, 1)h_2(\overline{a})(x)^2 = \prod_{i=1}^{n}(x - x_i)$ *up to sign.*

Vice versa, for any $\overline{a} \in \overline{\mathbb{K}}^{n-1}$, there exist $P_i = (x_i, y_i) \in E$ such that 1. *and* 2. *are verified. Moreover, the P_i's are unique up to permutation of the indices.*

Proof. The result is a straightforward generalization of [49, Corollary 4.2]. □

Notice that, by Theorem 28, if 1. of Proposition 62 is verified, then $P_1 \oplus \cdots \oplus P_n = \mathcal{O}$. The previous proposition tells us that the polynomial

$$h(a)(x, y) \in \mathbb{K}[a][x, y]$$

of the form (3.2) is such that, for any choice of $\overline{a} \in \overline{\mathbb{K}}^{n-1}$, the curve

$$C_{\overline{a}} : h(\overline{a})(x, y) = 0$$

intersects E in exactly n affine points $P_1, \cdots, P_n$, as we required in the previous subsection. Furthermore, Proposition 62 states that for each n affine points $P_1, \cdots, P_n$ on E that sum to $\mathcal{O}$, there exists a unique $\overline{a} \in \overline{\mathbb{K}}^{n-1}$ such that $C_{\overline{a}}$ and E intersect exactly in the affine points $P_1, \cdots, P_n$. Hence, for $n \in \mathbb{Z}_{\geq 2}$, there is a one-to-one correspondence between $\overline{\mathbb{K}}^{n-1}$ and multisets of n affine points of E that sum to $\mathcal{O}$.

Notation 63. Let $n \in \mathbb{Z}_{\geq 2}$. Let

$$h(a)(x, y) \in \mathbb{K}[a][x, y]$$

as in (3.2). For any $\overline{a} \in \overline{\mathbb{K}}^{n-1}$, denote by $P_1(\overline{a}), \cdots, P_n(\overline{a})$ the n affine points of E such that

$$\mathrm{div}(h(\overline{a})(x, y)) = \sum_{i=1}^{n} P_i(\overline{a}) - n\mathcal{O}.$$

Let $t, n_1, \cdots, n_t \in \mathbb{Z}_{\geq 2}$. For $i \in \{1, \cdots, t\}$, let $a_i = (a_{i,1}, \cdots, a_{i,n_i-1})$ be a vector of parameters. Let

$$S_{t,n_1,\cdots,n_t} \in \mathbb{K}[a_1, \cdots, a_t][x, y]$$

such that: for a generic choice of parameters $\overline{a}_1 \in (\overline{\mathbb{K}})^{n_1-1}, \cdots, \overline{a}_t \in (\overline{\mathbb{K}})^{n_t-1}$, the affine points of intersection between the elliptic curve E and the curve of equation

$$S_{t,n_1,\cdots,n_t}(\overline{a}_1 \cdots, \overline{a}_t)(x, y) = 0$$

are exactly all the possible affine sums of t points

$$P_{j_1}(\overline{a}_1) \oplus \cdots \oplus P_{j_t}(\overline{a}_t),$$

for $j_i \in \{1, \cdots, n_i\}$ and $i \in \{1, \cdots, t\}$. We will call such polynomial $(t, n_1, \cdots, n_t)$-generalized summation polynomial of E.

Definition 64 (Generalized summation polynomials of an elliptic curve)**.** Let $t, n_1, \cdots, n_t \in \mathbb{Z}_{\geq 2}$. We follow the notation above. A polynomial

$$S_{t,n_1,\cdots,n_t} \in \mathbb{K}[a_1, \cdots, a_t][x, y]$$

is called a $(t, n_1, \cdots, n_t)$-generalized summation polynomial of E if there exists a nonempty Zarisky open set

$$\mathcal{U} \subseteq \overline{\mathbb{K}}^{\sum_{i=1}^{t}(n_i-1)}$$

such that, for each $\overline{a}_i \in \overline{\mathbb{K}}^{n_i-1}$, $i \in \{1, \cdots, t\}$, with $(\overline{a_1}, \cdots, \overline{a}_t) \in \mathcal{U}$, we have the equality

$$\operatorname{div}\left(S_{t,n_1,\cdots,n_t}(\overline{a}_1, \cdots, \overline{a}_t)(x, y)\right) = \sum_{j_1=1,\cdots n_1,\cdots,j_t=1,\cdots,n_t} (P_{j_1}(\overline{a}_1) \oplus \cdots \oplus P_{j_t}(\overline{a}_t)) - \prod_{i=1}^{t} n_i \mathcal{O}. \tag{3.3}$$

Remark 65. A $(t, n_1, \cdots, n_t)$-generalized summation polynomial is a polynomial modulo the equation of the curve E. Moreover, for $\overline{a}_i \in \overline{\mathbb{K}}^{n_i-1}$, $i \in \{1, \cdots, t\}$, there are at most $\prod_{i=1}^{t} n_i$ affine points among the addition points $P_{j_1}(\overline{a}_1) \oplus \cdots \oplus P_{j_t}(\overline{a}_t)$. Therefore, by Proposition 62, a $(t, n_1, \cdots, n_t)$-generalized summation polynomial can be given in the form

$$S = S_1(x) + yS_2(x), \tag{3.4}$$

with $S_1, S_2 \in \mathbb{K}[a_1, \cdots, a_t][x]$, of degree $\deg(S_1(x)) = \left\lfloor \frac{\prod_{i=1}^{t} n_i}{2} \right\rfloor$, $\deg(S_2(x)) = \left\lfloor \frac{(\prod_{i=1}^{t} n_i)-3}{2} \right\rfloor$.

We recall from the previous subsection that one can interpret Semaev's summation polynomials as generalized summation polynomials for special parameters, in the following way.

Remark 66. Let $t \geq 2$, $n_1 = \cdots = n_t = 2$, $f_{t+1}(x_1, \cdots, x_t, x_{t+1}) \in \mathbb{K}[x_1, \cdots x_{t+1}]$ the $(t+1)$-th summation polynomial of E. For $i \in \{1, \cdots, t\}$, let $a_i = -x_i$ be parameters, and let $x_{t+1} = x$ be a variable. Then the polynomial $S(a_1, \cdots a_t)(x, y) = f_{t+1}(-a_1, \cdots, -a_t, x)$ is a $(t, n_1, \cdots, n_t)$-generalized summation polynomial of E.

3.3 Computation of generalized summation polynomials

We give here a recursive procedure to compute generalized summation polynomials of the elliptic curve E. First, we give an algorithm to compute a $(2, n_1, n_2)$-generalized summation polynomial, for any $n_1, n_2 \in \mathbb{Z}_{\geq 2}$. Then we use this algorithm to compute recursively $(t, n_1, \cdots, n_t)$-generalized summation polynomials, for $t \geq 2$. The correctness of our algorithms implies the existence of $(t, n_1, \cdots, n_t)$-generalized summation polynomials of E for any choice of parameters $t, n_1, \cdots, n_t \in \mathbb{Z}_{\geq 2}$.

We start with giving some notations and preliminary results.

Notation 67. Let $n_1, n_2 \in \mathbb{Z}_{\geq 2}$. Let $a_1 = (a_{1,1}, \cdots, a_{1,n_1-1})$ and $a_2 = (a_{2,1}, \cdots, a_{2,n_2-1})$ be two vectors of parameters. Let

$$h_1 = h_{1,1} + yh_{1,2} = h(a_1)(x, y) \in \mathbb{K}[a_1])[x, y] \text{ and } h_2 = h_{2,1} + yh_{2,2} = h(a_2)(x, y) \in \mathbb{K}[a_2][x, y]$$

as in (3.2). Moreover, let

$$H_1(a_1)(x) = h_{1,1}(a_1)(x)^2 - f(x, 1)h_{1,2}(a_2)(x)^2 \in \mathbb{K}[a_1][x]$$

and

$$H_2(a_2)(x) = h_{2,1}(a_2)(x)^2 - f(x,1)h_{2,2}(a_2)(x)^2 \in \mathbb{K}[a_2][x].$$

For $i \in \{1, \cdots, n_1\}$, $j \in \{1, \cdots, n_2\}$, let

$$P_i = P_i(a_1) = (x_{P_i}(a_1), y_{P_i}(a_i)) \text{ and } Q_j = Q_j(a_2) = (x_{Q_j}(a_2), y_{Q_j}(a_2)),$$

such that, for each $\overline{a}_1 \in \overline{\mathbb{K}}^{n_1-1}$, $\overline{a}_2 \in \overline{\mathbb{K}}^{n_2-1}$, $i \in \{1, \cdots, n_1\}$, $j \in \{1, \cdots, n_2\}$, we have $P_i(\overline{a}_1)$ and $Q_j(\overline{a}_2)$ as in Notation 63.
For $i \in \{1, \cdots, n_1\}$, let $e_i(x_{P_1}, \cdots, x_{P_{n_1}})$ be the i-th elementary symmetric polynomial in the variables $x_{P_1}, \cdots, x_{P_{n_1}}$. Point 2. of Proposition 62 implies that one can express the elementary symmetric polynomials $e_i(x_{P_1}, \cdots, x_{P_{n_1}})$ in terms of $a_1 = (a_{1,1} \cdots, a_{1,n-1})$. More precisely, they will be quadratic polynomials in $a_{1,1} \cdots, a_{1,n_1-1}$. Therefore, for $1 \in \{1, \cdots, n_1\}$ let $E_i(a_1) \in \mathbb{K}[a_1]$ be the quadratic polynomial such that

$$e_i(x_{P_1}, \cdots, x_{P_{n_1}}) = E_i(a_1).$$

Similarly, for $j \in \{1, \cdots, n_2\}$, let $s_j(x_{Q_1}, \cdots, x_{Q_{n_2}})$ be the j-th elementary symmetric polynomial in the variables $x_{Q_1}, \cdots, x_{Q_{n_2}}$ and $S_j(a_2) \in \mathbb{K}[a_2]$ be the quadratic polynomial such that

$$s_j(x_{Q_1}, \cdots, x_{Q_{n_2}}) = S_j(a_2).$$

Lemma 68. *We refer to Notation 67. Let $\overline{a}_1 \in \overline{\mathbb{K}}^{n_1-1}$, $\overline{a}_2 \in \overline{\mathbb{K}}^{n_2-1}$. We have the following facts.*

1. *If $n_1 > 2$, then $h_{1,2}(\overline{a}_1)(x_{P_i}(\overline{a}_1)) = 0$ for some $i \in \{1, \cdots, n_1\}$ if and only if*

$$\gcd(h_{1,1}(\overline{a}_1)(x), h_{1,2}(\overline{a}_1)(x)) \neq 1.$$

Moreover, if $n_1 > 2$, then $h_{1,2}(\overline{a}_1)(x_{P_i}(\overline{a}_1)) = 0$ implies that there exist $j \in \{1, \cdots, n_1\}$ such that $P_i(\overline{a}_1) = -P_j(\overline{a}_1)$.

2. *$P_i(\overline{a}_1) = Q_j(\overline{a}_2)$ for some $i \in \{1, \cdots, n_1\}$, $j \in \{1, \cdots, n_2\}$ if and only if*

$$\gcd(H_1(\overline{a}_1)(x), H_2(\overline{a}_2)(x), (h_{1,1}(\overline{a}_1)h_{2,2}(\overline{a}_2) - h_{1,2}(\overline{a}_1)h_{2,1}(\overline{a}_2))(x)) \neq 1. \quad (3.5)$$

3. *$P_i(\overline{a}_1) = -Q_j(\overline{a}_2)$ for some $i \in \{1, \cdots, n_1\}$, $j \in \{1, \cdots, n_2\}$ if and only if*

$$\gcd(H_1(\overline{a}_1)(x), H_2(\overline{a}_2)(x), (h_{1,1}(\overline{a}_1)h_{2,2}(\overline{a}_2) + h_{1,2}(\overline{a}_1)h_{2,1}(\overline{a}_2))(x)) \neq 1.$$

Proof. 1. We have that $h_1(\overline{a}_1)(P_i(\overline{a}_1)) = 0$ by definition of h_1. Then $h_{1,2}(\overline{a}_1)(x_{P_i}(\overline{a}_1)) = 0$ implies $h_{1,1}(\overline{a}_1)(x_{P_i}(\overline{a}_1)) = 0$, from which $\gcd(h_{1,1}(\overline{a}_1)(x), h_{1,2}(\overline{a}_1)(x)) \neq 1$, since the two polynomials share the common root $x_{P_i}(\overline{a}_1)$. Vice versa, if $\gcd(h_{1,1}(\overline{a}_1)(x), h_{1,2}(\overline{a}_1)(x)) \neq 1$, then there exists $x_0 \in \overline{\mathbb{K}}$ such that $h_{1,1}(\overline{a}_1)(x_0) = h_{1,2}(\overline{a}_1)(x_0) = 0$. By definition of h_1 there exists $i \in \{1, \cdots, n_1\}$ such that $x_0 = x_{P_i}(\overline{a}_1)$, from which $h_{1,2}(\overline{a}_1)(x_{P_i}(\overline{a}_1)) = 0$. Moreover, since $h_{1,2}(\overline{a}_1)(x_{P_i}(\overline{a}_1)) = 0$ implies $h_{1,1}(\overline{a}_1)(x_{P_i}(\overline{a}_1)) = 0$, we have $h_1(\overline{a}_1)(\pm P_i(\overline{a}_1)) = 0$. Hence, there exists $j \in \{1, \cdots, n_1\}$ such that $P_i(\overline{a}_1) = -P_j(\overline{a}_1)$.

2. If $P_i(\overline{a}_1) = Q_j(\overline{a}_2) = (x_0, y_0)$ for some $i \in \{1, \cdots, n_1\}$, $j \in \{1, \cdots, n_2\}$, then, by Proposition 62, we have $H_1(\overline{a}_1)(x_0) = H_2(\overline{a}_2)(x_0) = (h_{1,1}(\overline{a}_1)h_{2,2}(\overline{a}_2) - h_{1,2}(\overline{a}_1)h_{2,1}(\overline{a}_2))(x_0) = 0$. Then the three polynomials of (3.5) share the common root x_0, and their greatest common divisor is not 1. Vice versa, suppose that (3.5) holds true. Then there exists $x_0 \in \overline{\mathbb{K}}$ such

that $H_1(\overline{a}_1)(x_0) = H_2(\overline{a}_2)(x_0) = (h_{1,1}(\overline{a}_1)h_{2,2}(\overline{a}_2) - h_{1,2}(\overline{a}_1)h_{2,1}(\overline{a}_2))(x_0) = 0$. The equality $H_1(\overline{a}_1)(x_0) = H_2(\overline{a}_2)(x_0) = 0$ implies that there exist $i \in \{1, \cdots, n_1\}$, $j \in \{1, \cdots, n_2\}$, such that

$$P_i(\overline{a}_1) = (x_0, y_0) = \pm Q_j(\overline{a}_2). \tag{3.6}$$

If $h_{1,2}(\overline{a}_1)(x_0) \neq 0$, then the equality $(h_{1,1}(\overline{a}_1)h_{2,2}(\overline{a}_2) - h_{1,2}(\overline{a}_1)h_{2,1}(\overline{a}_2))(x_0) = 0$ implies $h_2(\overline{a}_2)(x_0, y_0) = 0$, dividing by $h_{1,2}(\overline{a}_1)(x_0)$ and replacing $(-h_{1,1}(\overline{a}_1)/h_{1,2}(\overline{a}_1))(x_0)$ with y_0. Hence, in this case, $P_i(\overline{a}_1) = Q_j(\overline{a}_2)$ for some $j \in \{1, \cdots, n_2\}$. If $h_{1,2}(\overline{a}_1)(x_0) = 0$, then there exists $i' \in \{1, \cdots, n_1\}$ such that $P_{i'}(\overline{a}_1) = -P_i(\overline{a}_1)$. Since we have (3.6), the thesis follows again.

3. The proof of point 3 is analogous to that of point 2. □

Remark 69. Let $s(x), t(x) \in \overline{\mathbb{K}}[x]$. We recall that $\gcd(s, t) = 1$ if and only if

$$\mathrm{res}_x(s, t) = 0, \tag{3.7}$$

where $\mathrm{res}_x(s, t)$ is the Sylvester resultant of the polynomials s and t with respect to x. Moreover, the resultant is a polynomial in the coefficients of s and t.

The following result allows us to define the nonempty Zarisky open set in which the $(2, n_1, n_2)$-generalized summation polynomial that we compute in Algorithm 7 satisfies the property (3.3).

Lemma 70. *We refer to Notation 67. Let $n_1, n_2 \in \mathbb{Z}_{\geq 2}$. We define the following sets $\mathcal{U}_1$, $\mathcal{U}_2$, $\mathcal{U}_3$, $\mathcal{U}_4$.*

1. $\mathcal{U}_1 = \{\overline{a}_1 \in \overline{\mathbb{K}}^{n_1-1} : \mathrm{res}_x(h_{1,1}(\overline{a}_1)(x), h_{1,2}(\overline{a}_1)(x)) \neq 0 \text{ if } n_1 > 2\} \subseteq \overline{\mathbb{K}}^{n_1-1}$.
2. $\mathcal{U}_2 = \{\overline{a}_2 \in \overline{\mathbb{K}}^{n_2-1} : \mathrm{res}_x(h_{2,1}(\overline{a}_2)(x), h_{2,2}(\overline{a}_2)(x)) \neq 0 \text{ if } n_2 > 2\} \subseteq \overline{\mathbb{K}}^{n_2-1}$.
3. $\mathcal{U}_3 = \{(\overline{a}_1, \overline{a}_2) \in \overline{\mathbb{K}}^{n_1-1} \times \overline{\mathbb{K}}^{n_2-1} :$

   $\mathrm{res}_x(\gcd(H_1(\overline{a}_1)(x), H_2(\overline{a}_2)(x)), (h_{1,1}(\overline{a}_1)h_{2,2}(\overline{a}_2) - h_{1,2}(\overline{a}_1)h_{2,1}(\overline{a}_2))(x)) \neq 0\} \subseteq \overline{\mathbb{K}}^{n_1+n_2-2}$.
4. $\mathcal{U}_4 = \{(\overline{a}_1, \overline{a}_2) \in \overline{\mathbb{K}}^{n_1-1} \times \overline{\mathbb{K}}^{n_2-1} :$

   $\mathrm{res}_x(\gcd(H_1(\overline{a}_1)(x), H_2(\overline{a}_2)(x)), (h_{1,1}(\overline{a}_1)h_{2,2}(\overline{a}_2) + h_{1,2}(\overline{a}_1)h_{2,1}(\overline{a}_2))(x)) \neq 0\} \subseteq \overline{\mathbb{K}}^{n_1+n_2-2}$.

Then we have the following facts.

1. *The set $\mathcal{U}_1$ is a nonempty Zarisky open set of $\overline{\mathbb{K}}^{n_1-1}$. Moreover, for each $\overline{a}_1 \in \mathcal{U}_1$, one has $h_{1,2}(\overline{a}_1)(x_{P_i}(\overline{a}_1)) \neq 0$ for all $i \in \{1, \cdots, n_1\}$.*
2. *The set $\mathcal{U}_2$ is a nonempty Zarisky open set of $\overline{\mathbb{K}}^{n_2-1}$. Moreover, for each $\overline{a}_2 \in \mathcal{U}_2$, one has $h_{2,2}(\overline{a}_2)(x_{Q_i}(\overline{a}_1)) \neq 0$ for all $i \in \{1, \cdots, n_2\}$.*
3. *The set $\mathcal{U}_3$ is a nonempty Zarisky open set of $\overline{\mathbb{K}}^{n_1+n_2-2}$. Moreover, for each $\overline{a}_1 \in \overline{\mathbb{K}}^{n_1-1}$, $\overline{a}_2 \in \overline{\mathbb{K}}^{n_2-1}$, with $(\overline{a}_1, \overline{a}_2) \in \mathcal{U}_3$, one has $P_i(\overline{a}_1) \neq Q_j(\overline{a}_2)$ for each $i \in \{1, \cdots, n_1\}$ and $j \in \{1, \cdots, n_2\}$.*
4. *The set $\mathcal{U}_4$ is a nonempty Zarisky open set of $\overline{\mathbb{K}}^{n_1+n_2-2}$. Moreover, for each $\overline{a}_1 \in \overline{\mathbb{K}}^{n_1-1}$, $\overline{a}_2 \in \overline{\mathbb{K}}^{n_2-1}$, with $(\overline{a}_1, \overline{a}_2) \in \mathcal{U}_4$, one has $P_i(\overline{a}_1) \neq -Q_j(\overline{a}_2)$ for each $i \in \{1, \cdots, n_1\}$ and $j \in \{1, \cdots, n_2\}$.*

5. For each $\overline{a}_1 \in \overline{\mathbb{K}}^{n_1-1}$, there exists $\overline{a}_2 \in \mathcal{U}_2$, such that $(\overline{a}_1, \overline{a}_2) \in \mathcal{U}_3 \cap \mathcal{U}_4$.

Proof. The sets $\mathcal{U}_1, \mathcal{U}_2, \mathcal{U}_3, \mathcal{U}_4$ are Zarisky open sets by Remark 69. Moreover, they satisfy the required properties by Lemma 68 and Remark 69. We now prove that the defined open sets are nonempty.
We show that $\mathcal{U}_1$ is nonempty. With the same arguments, one shows that $\mathcal{U}_2$ is nonempty. Take $P_1, \cdots, P_{n_1} \in E \setminus \{\mathcal{O}\}$, such that $P_1 \oplus \cdots \oplus P_{n_1} = \mathcal{O}$ and $P_i \neq -P_j$ for all $i, j \in \{1, \cdots, n_1\}$. Notice that we can always take $P_1, \cdots, P_{n_1}$ which satisfy the required condition, since $\overline{\mathbb{K}}$ is an infinite field (because it is algebraically closed). Take the tuple $\overline{a}_1 \in \overline{\mathbb{K}}^{n_1-1}$ associated to $P_1, \cdots, P_{n_1}$, as in Proposition 62. Then, by point 1 of Lemma 68, we have that $\overline{a}_1 \in \mathcal{U}_1$.
We now prove point 5 of the lemma, which implies that both $\mathcal{U}_3$ and $\mathcal{U}_4$ are nonempty, since $\mathcal{U}_2$ is nonempty. Let $\overline{a}_1 \in \overline{\mathbb{K}}^{n_1-1}$. Take $Q_1, \cdots, Q_{n_2} \in E \setminus \{\mathcal{O}\}$ such that : $Q_1 \oplus \cdots \oplus Q_{n_2} = \mathcal{O}$, $Q_i \neq -Q_j$ for all $i, j \in \{1, \cdots, n_2\}$, $Q_i \neq \pm P_j(\overline{a}_1)$ for each $i \in \{1, \cdots, n_2\}$, $j \in \{1, \cdots, n_1\}$. Notice that we can always take $Q_1, \cdots, Q_{n_2}$ which satisfy the required conditions, since $\overline{\mathbb{K}}$ is infinite. Take the tuple $\overline{a}_2 \in \overline{\mathbb{K}}^{n_2-1}$ associated to $Q_1, \cdots, Q_{n_2}$, as in Proposition 62. Hence $\overline{a}_2 \in \mathcal{U}_2$, and $(\overline{a}_1, \overline{a}_2) \in \mathcal{U}_3 \cap \mathcal{U}_4$ by Lemma 68. □

Algorithm 7 (Computation of a $(2, n_1, n_2)$-generalized summation polynomial)**.**

Input : $n_1, n_2 \in \mathbb{Z}_{\geq 2}$.
Output: $S = S_1 + yS_2$ a $(2, n_1, n_2)$-generalized summation polynomial, in the form (3.4)

---

for $i \in \{1, \cdots, n_1\}$
  Compute $E_i \leftarrow E_i(a_{1,1}, \cdots, a_{1,n_1-1})$ using the equality of point 2 of Proposition 62
end for
for $i \in \{1, \cdots, n_2\}$
  Compute $S_i \leftarrow S_i(a_{2,1}, \cdots, a_{2,n_2-1})$ using the equality of point 2 of Proposition 62
end for
for $i \in \{1 \cdots, n_1\}$ ▷ $r_{i,1}$ line through P_i and Q_1 as a polynomial in the variables $x_{P_i}, y_{P_i}, x_{Q_1}, y_{Q_1}, x, y$
  $r_{i,1}(x_{P_i}, x_{Q_1}, y_{P_i}, y_{Q_1}, x, y) \leftarrow (y_{Q_1} - y_{P_i})x + (x_{P_i} - x_{Q_1})y + ((x_{Q_1} - x_{P_i})y_{P_i} + (y_{P_i} - y_{Q_1})x_{P_i})$
end for
  $K(x_{P_1}, \cdots, x_{P_{n_1}}, y_{P_1}, \cdots, y_{P_{n_1}}, x_{Q_1}, y_{Q_1}, x, y) \leftarrow \prod_{i=1}^{n_1} r_{i,1}$
if $n_1 = 2$ **then**
  Replace x_{P_1} and x_{P_2} by $-a_{1,1}$ in K
  Replace y_{P_2} by $-y_{P_1}$ in K ▷ $K(a_{1,1}, x_{Q_1}, y_{Q_1}, x, y)(y_{P_1}) = (K_1^2 - K_2^2 y_{P_1}^2)$
  Replace $(y_{P_1})^2$ by $f(a_{1,1}, 1)$ in K
else
  for $i \in \{1, \cdots, n_1\}$
    Replace y_{P_i} by $-h_{1,1}(x_{P_i})/h_{1,2}(x_{P_i})$ in K
  end for
  Write $K(x_{P_1}, \cdots, x_{P_{n_1}})$ as a function of $e_1, \cdots, e_{n_1}$
  for $i \in \{1, \cdots, n_1\}$
    Replace e_i by E_i in K
  end for
end if ▷ Now $K = K(a_1)(x_{Q_1}, y_{Q_1}, x, y)$
for $i \in \{1, \cdots, n_2\}$
  $K_i \leftarrow K(a_1, x_{Q_i}, y_{Q_i}, x, y)$

26: **end for**
27: $K(a_1, x_{Q_1}, \cdots, x_{Q_{n_2}}, y_{Q_1}, \cdots, y_{Q_{n_2}}, x, y) \leftarrow \prod_{i=1}^{n_2} K_i$
28: **if** $n_2 = 2$ **then**
29: Replace x_{Q_1} and x_{Q_2} by $-a_{2,1}$ in K
30: Replace y_{Q_2} by $-y_{Q_1}$ in K
31: Replace $(y_{Q_1})^{2k}$ by $f(a_{2,1}, 1)^k$ in K for all k.
32: **else**
33: **for** $i \in \{1, \cdots, n_2\}$
34: Replace y_{Q_i} by $-h_{2,1}(x_{Q_i})/h_{2,2}(x_{Q_i})$ in K
35: **end for**
36: Write $K(x_{Q_1}, \cdots, x_{Q_{n_2}})$ as a function of $s_1, \cdots, s_{n_2}$
37: **for** $i \in \{1, \cdots, n_2\}$
38: Replace s_i by S_i in K
39: **end for**
40: **end if** ▷ Now $K = K(a_1, a_2)(x, y)$
41: Use equality $S_1(x) - yS_2(x) = K(x, y)/(h_1^{n_2} h_2^{n_1}) \mod y^2 - f(x, 1)$, removing denominators
42: **return** S

Theorem 71. *Algorithm* 7 *is correct.*

Proof. We first prove that Algorithm 7 outputs a polynomial

$$S(a_1, a_2)(x, y) \in \mathbb{K}[a_1, a_2][x, y].$$

For $i \in \{1, \cdots, n_1\}$, we take the polynomial

$$r_{i,1}(x_{P_i}, y_{P_i}, x_{Q_1}, y_{Q_1}, x, y) \in \mathbb{K}[x_{P_i}, y_{P_i}, x_{Q_1}, y_{Q_1}, x, y]$$

computed in line 8. Moreover, we take the polynomial

$$K(x_{P_1}, \cdots, x_{P_n}, y_{P_1}, \cdots, y_{P_n}, x_{Q_1}, y_{Q_1}, x, y) \in \mathbb{K}[x_{P_1}, \cdots, x_{P_n}, y_{P_1}, \cdots, y_{P_n}, x_{Q_1}, y_{Q_1}, x, y]$$

computed in line 10. The polynomial K is the product of all the $r_{i,1}$'s.
Suppose first that $n_1 = 2$, as in lines $12 - 14$ of the algorithm. Observe that $r_{i,1}$ is linear as polynomial in y_{P_i}. Hence, after line 13, we have that K is a polynomial of the form $K(a_{1,1}, x_{Q_1}, y_{Q_1}, x, y)(y_{P_1}) = (K_1 + K_2 y_{P_1})(K_1 - K_2 y_{P_1}) = K_1^2 - K_2^2 y_{P_1}^2$, for some $K_1, K_2 \in \mathbb{K}[a_{1,1}, x_{Q_1}, y_{Q_1}, x, y]$. Then, after line 14, we have $K = K(a_{1,1}, x_{Q_1}, y_{Q_1}, x, y) \in \mathbb{K}[a_{1,1}, x_{Q_1}, y_{Q_1}, x, y]$.
Now suppose $n_1 > 2$ as in lines $15 - 23$. After lines $16 - 18$, we have

$$K = K(a_1, x_{P_1}, \cdots, x_{P_n}, x_{Q_1}, y_{Q_1}, x, y) \in \mathbb{K}[x_{Q_1}, y_{Q_1}, x, y](a_1)(x_{P_1}, \cdots, x_{P_n}).$$

Moreover, from line 10, both the numerator and the denominator of K are symmetric as polynomials in the variables $x_{P_1}, \cdots, x_{P_n}$. Then we can write them as a function of the elementary symmetric polynomials $e_i(x_{P_1}, \cdots, x_{P_n})$, for $i \in \{1, \cdots, n_1\}$, as in line 19. Then, in lines $20 - 22$, we replace all $e_i(x_{P_1}, \cdots, x_{P_{n_1}})$ with the polynomials $E_i(a_1)$ computed in lines $1 - 3$. In this way, after line 23, we obtain $K = K(a_1, x_{Q_1}, y_{Q_1}, x, y) \in \mathbb{K}(a_1)[x_{Q_1}, y_{Q_1}, x, y]$.

With the same arguments, one shows that, after line 40, we have $K \in \mathbb{K}(a_1, a_2)[x, y]$. Hence, after performing line 41, we obtain

$$S(a_1, a_2)(x, y) \in \mathbb{K}[a_1, a_2](x, y).$$

We now define a nonempty Zarisky open set $\mathcal{U}_S \subseteq \overline{\mathbb{K}}^{n_1+n_2-2}$ and we show that, for each $\overline{a}_1 \in \overline{\mathbb{K}}^{n_1-1}$, $\overline{a}_2 \in \overline{\mathbb{K}}^{n_2-1}$ with $(\overline{a}_1, \overline{a}_2) \in \mathcal{U}_S$, we have that the polynomial $S(\overline{a}_1, \overline{a}_2)(x, y)$ satisfies the property (3.3).
We define $\mathcal{U}_S$ in the following way.

$$\mathcal{U}_S = (\mathcal{U}_1 \times \overline{\mathbb{K}}^{n_2-1}) \cap (\overline{\mathbb{K}}^{n_1-1} \times \mathcal{U}_2) \cap \mathcal{U}_3, \tag{3.8}$$

where $\mathcal{U}_1$, $\mathcal{U}_2$, $\mathcal{U}_3$ are the nonempty Zarisky open sets defined in Lemma 70. We have that $\mathcal{U}_S$ is a nonempty Zarisky open set of $\overline{\mathbb{K}}^{n_1+n_2-2}$, since it is the intersection of three nonempty Zarisky open sets, by Lemma 70.
Now we prove that, for each $\overline{a}_1 \in \overline{\mathbb{K}}^{n_1-1}$, $\overline{a}_2 \in \overline{\mathbb{K}}^{n_2-1}$ with $(\overline{a}_1, \overline{a}_2) \in \mathcal{U}_S$, the polynomial $S(\overline{a}_1, \overline{a}_2)(x, y)$ satisfies the property (3.3). This means that we have

$$\operatorname{div}(S(\overline{a}_1, \overline{a}_2)(x, y)) = \sum_{i=1,\cdots n_1, j=1,\cdots,n_2} (P_i(\overline{a}_1) \oplus Q_j(\overline{a}_2)) - n_1 n_2 \mathcal{O}.$$

Take $(\overline{a}_1, \overline{a}_2) \in \mathcal{U}_S$, and let $r_{i,j}(x_{P_i}, x_{Q_j}, y_{P_i}, y_{Q_j}, x, y)$ be the polynomial of line 8, where 1 is replaced with j, for $j \in \{1, \cdots, n_2\}$. Since $(\overline{a}_1, \overline{a}_2) \in \mathcal{U}_S$, we have that $P_i(\overline{a}_1) \neq Q_j(\overline{a}_2)$ for all $i \in \{1, \cdots, n_1\}$, $j \in \{1, \cdots, n_2\}$, by point 3 of Lemma 68. Then

$$r_{i,j}(\overline{a}_1, \overline{a}_2)(x, y) = r_{i,j}(x_{P_i}(\overline{a}_1), x_{Q_j}(\overline{a}_2), y_{P_i}(\overline{a}_1), y_{Q_j}(\overline{a}_2), x, y) = 0$$

is the equation of the line through $P_i(\overline{a}_1)$ and $Q_j(\overline{a}_2)$, and we have

$$\operatorname{div}(r_{i,j}(\overline{a}_1, \overline{a}_1)(x, y)) = P_i(\overline{a}_1) + Q_j(\overline{a}_2) + (-(P_i(\overline{a}_1) \oplus Q_j(\overline{a}_2)) - 3\mathcal{O}.$$

So the polynomial $K(x_{P_1}, \cdots, x_{P_{n_1}}, y_{P_1}, \cdots, y_{P_{n_1}}, x_{Q_1}, y_{Q_1}, x, y)$ computed in line 10 is such that

$$K(x_{P_1}(\overline{a}_1), \cdots, x_{P_{n_1}}(\overline{a}_1), y_{P_1}(\overline{a}_1), \cdots, y_{P_{n_1}}(\overline{a}_1), x_{Q_1}(\overline{a}_2), y_{Q_1}(\overline{a}_2))(x, y) = 0$$

is the union of the n_1 lines $r_{i,1}(\overline{a}_1, \overline{a}_2)(x, y) = 0$ through $P_i(\overline{a}_1)$ and $Q_1(\overline{a}_2)$, for $i \in \{1, \cdots, n_1\}$.
Suppose first that $n_1 = 2$, as in lines $11 - 14$ of the algorithm. Then h_1 has the form $h_1 = a_{1,1} + x$, and

$$h_1(\overline{a}_1)(x, y) = \overline{a}_{1,1} + x = 0$$

defines the vertical line through $P_1(\overline{a}_1) = (-\overline{a}_{1,1}, y_{P_1}(\overline{a}_1))$ and $P_2(\overline{a}_1) = -P_1(\overline{a}_1) = (-\overline{a}_{1,1}, -y_{P_1}(\overline{a}_1))$. Hence $x_{P_1}(\overline{a}_1) = x_{P_2}(\overline{a}_1) = -\overline{a}_{1,1}$ and $y_{P_2}(\overline{a}_1) = -y_{P_1}(\overline{a}_1)$, moreover $P_1(\overline{a}_1) \in E$. So, after line 14 of the algorithm,

$$K(\overline{a}_1, x_{Q_1}(\overline{a}_2), y_{Q_1}(\overline{a}_2))(x, y) = 0$$

is still the union of the two lines $r_{1,1}(\overline{a}_1, \overline{a}_2)(x, y) = 0$ and $r_{2,1}(\overline{a}_1, \overline{a}_2)(x, y) = 0$.
Now suppose that $n_1 > 2$ as in lines $15 - 23$ of the algorithm. Hence, for each $i \in \{1, \cdots, n_1\}$, we have that $h_{1,2}(\overline{a}_1)(x_{P_i}(\overline{a}_1)) \neq 0$ by point 1 of Lemma 68, since $(\overline{a}_1, \overline{a}_2) \in \mathcal{U}_S$. Furthermore, by definition of h_1, we have that $y_{P_i}(\overline{a}_1) = -h_{1,1}(\overline{a}_1)(x_{P_i}(\overline{a}_1))/h_{1,2}(\overline{a}_1)(x_{P_i}(\overline{a}_1))$.

Moreover, by point 2 of Proposition 62, we have that $e_i(x_{P_1}(\overline{a}_1), \cdots, x_{P_{n_1}}(\overline{a}_1)) = E_i(\overline{a}_1)$ for each $i \in \{1, \cdots, n_1\}$. Hence, after line 23,

$$K(\overline{a}_1, x_{Q_1}(\overline{a}_2), y_{Q_1}(\overline{a}_2))(x, y) = 0$$

is still the union of the n_1 lines $r_{i,1}(\overline{a}_1, \overline{a}_2)(x, y) = 0$, for $i \in \{1, \cdots, n_1\}$. With the same arguments we show that, after line 40, we have that

$$K(\overline{a}_1, \overline{a}_2)(x, y) = 0$$

is the union of the $n_1 n_2$ lines $r_{i,j}(\overline{a}_1, \overline{a}_2)(x, y) = 0$, for $i \in \{1, \cdots, n_1\}$, $j \in \{1, \cdots, n_2\}$. Therefore, we have

$$\operatorname{div}(K(\overline{a}_1, \overline{a}_2)(x, y))) = n_2 \sum_{i=1}^{n_1} P_i(\overline{a}_1) + n_1 \sum_{j=1}^{n_2} Q_j(\overline{a}_2) + \sum_{i=1,\cdots,n_1, j=1,\cdots,n_2} (-(P_i(\overline{a}_1) \oplus Q_j(\overline{a}_2))) - 3n_1 n_2 \mathcal{O}.$$

Moreover, we have

$$\operatorname{div}(h_1(\overline{a}_1)^{n_2} h_2(\overline{a}_2)^{n_1}(x, y)) = n_2 \sum_{i=1}^{n_1} P_i(\overline{a}_1) + n_1 \sum_{i=1}^{n_2} Q_i(\overline{a}_2) - 2n_1 n_2 \mathcal{O}.$$

Hence, the polynomial $S(a_1, a_2)(x, y) \in \mathbb{K}[a_1, a_2][x, y]$ computed in line 41 is such that

$$\operatorname{div}(S(\overline{a}_1, \overline{a}_2)(x, y)) = \sum_{i=1,\cdots,n_1, j=1,\cdots,n_2} (P_i(\overline{a}_1) \oplus Q_j(\overline{a}_2)) - n_1 n_2 \mathcal{O},$$

as required. Then Algorithm 7 correctly computes and outputs a $(2, n_1, n_2)$-generalized summation polynomial, for $n_1, n_2 \in \mathbb{Z}_{\geq 2}$. □

Remark 72. We have showed that Algorithm 7 outputs a $(2, n_1, n_2)$-generalized summation polynomial that verifies property (3.3) in the nonempty Zarisky open set $\mathcal{U}_S$ defined in (3.8).
Now we want to take nonempty Zarisky open subsets of $\mathcal{U}_S$, where the affine points of intersection between $S(\overline{a}_1, \overline{a}_2)(x, y) = 0$ and E have additional properties.

1. Take the nonempty Zarisky open set $\mathcal{U}_4$ defined in point 4 of Lemma 70. Let $\mathcal{U}'_S$ be the Zarisky open set
$$\mathcal{U}'_S = \mathcal{U}_S \cap \mathcal{U}_4. \tag{3.9}$$
We have that $\mathcal{U}'_S$ is nonempty since $\mathcal{U}_S$ and $\mathcal{U}_4$ are nonempty by Lemma 70. Moreover, for all $(\overline{a}_1, \overline{a}_2) \in \mathcal{U}'_S$, we have that $S(\overline{a}_1, \overline{a}_2)(x, y)$ satisfies property (3.3), and $P_i(\overline{a}_1) \oplus Q_j(\overline{a}_2) \neq \mathcal{O}$ for all $i \in \{1, \cdots, n_1\}$, $j \in \{1, \cdots, n_2\}$, by point 4 of Lemma 70. In this case, we have that $S(\overline{a}_1, \overline{a}_2)(x, y) = 0$ has the maximal number $n_1 n_2$ of affine points of intersection with the elliptic curve E. Then, if $(\overline{a}_1, \overline{a}_2) \in \mathcal{U}'_S$, by Proposition 62, we have $\deg(S_1(\overline{a}_1, \overline{a}_2)(x)) = \frac{n_1 n_2}{2}$ if $n_1 n_2$ is even, and $\deg(S_2(\overline{a}_1, \overline{a}_2)(x)) = \frac{n_1 n_2 - 3}{2}$ if $n_1 n_2$ is odd.

2. Let $\mathcal{U}''_S$ be the set
$$\mathcal{U}''_S = \{(\overline{a}_1, \overline{a}_2) \in \mathcal{U}'_S : \operatorname{res}_x(S_1(\overline{a}_1, \overline{a}_2)(x), S_2(\overline{a}_1, \overline{a}_2)(x)) \neq 0\}. \tag{3.10}$$
This is a Zarisky open subset of $\mathcal{U}'_S$. Moreover, and for all $(\overline{a}_1, \overline{a}_2) \in \mathcal{U}''_S$, we have
$$S_2(\overline{a}_1, \overline{a}_2)(x_{P_i(\overline{a}_1) \oplus Q_j(\overline{a}_2)}) \neq 0$$

for all $i \in \{1, \cdots, n_1\}$, $j \in \{1, \cdots, n_2\}$, by point 1 of Lemma 68. In addition, $\mathcal{U}_S''$ is nonempty. More precisely, we have that for all $\overline{a}_1 \in \mathcal{U}_1$, there exists $\overline{a}_2 \in \mathcal{U}_2$ such that $(\overline{a}_1, \overline{a}_2) \in \mathcal{U}_S''$. This can be shown with the same arguments used in the proof of point 5 of Lemma 70.

Example 73 (Computation of a $(2,3,3)$-generalized summation polynomial)**.** We computed a $(2,3,3)$-generalized summation polynomial following Algorithm 7. For $t = 2$, $n_1 = n_2 = 3$, we let $a_1 = (-\alpha_0, -\alpha_1)$, $a_2 = (-\beta_0, -\beta_1)$, $h_1(a_1)(x,y) = (-\alpha_0 - \alpha_1 x) + y \in \mathbb{K}[\alpha_0, \alpha_1](x,y)$, $h_2(a_2)(x,y) = (-\beta_0 - \beta_1 x) + y \in \mathbb{K}[\beta_0, \beta_1](x,y)$. By Remark 65, we have that the $(2,3,3)$-generalized summation polynomial computed with Algorithm 7 is a polynomial

$$S(\alpha_0, \alpha_1, \beta_0, \beta_1)(x,y) \in \mathbb{K}[\alpha_0, \alpha_1, \beta_0, \beta_1][x,y]$$

of the form

$$S_1(x) + yS_2(x) = (a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0) + y(b_3x^3 + b_2x^2 + b_1x + b_0),$$

where $a_i, b_j \in \mathbb{K}[\alpha_0, \alpha_1, \beta_0, \beta_1]$ for each $i \in \{0, \cdots, 4\}$, $j \in \{0, \cdots, 3\}$. The formulas for the coefficients $a_0, \cdots, a_4, b_0, \cdots, b_3$ that we have computed can be found in Section 1 of the appendix.

Now we give a recursive procedure which uses Algorithm 7 to compute $(t, n_1, \cdots, n_t)$-generalized summation polynomials for all $t, n_1, \cdots, n_t \in \mathbb{Z}_{\geq 2}$.

Notation 74. For $t, n_1, \cdots, n_t \in \mathbb{Z}_{\geq 2}$, we denote by $a_1, \cdots, a_t$ the t vectors of parameters $a_i = (a_{i,1}, \cdots, a_{i,n_i-1})$, for each $i \in \{1, \cdots, t\}$.

Algorithm 8 (Computation of a $(t, n_1, \cdots, n_t)$-generalized summation polynomial)**.**

Input : $t, n_1, \cdots, n_t \in \mathbb{Z}_{\geq 2}$.
Output: $S = S_1 + yS_2$ a $(t, n_1, \cdots, n_t)$-generalized summation polynomial, of the form (3.4)

---

if $t = 2$ **then**
  $S \leftarrow \text{Algorithm}_7(n_1, n_2)$
else
  $T \leftarrow \text{Algorithm}_8(t-1, n_1, \cdots, n_{t-1})$
  $N \leftarrow \prod_{i=1}^{t-1} n_i$
  if N is even **then** ▷ $\hat{\alpha}_1 \cdots, \hat{\alpha}_N, d \in \mathbb{K}[a_1, \cdots, a_{t-1}]$
    $T = T_1(x) + yT_2(x) = (\hat{\alpha}_1 + \hat{\alpha_2}x + \cdots + \hat{\alpha}_{\frac{N}{2}} x^{\frac{N-2}{2}} + dx^{\frac{N}{2}}) + y(\hat{\alpha}_{\frac{N+2}{2}} + \cdots + \hat{\alpha}_{N-1} x^{\frac{N-4}{2}})$
  else
    $T = T_1(x) + yT_2(x) = (\hat{\alpha}_1 + \hat{\alpha_2}x + \cdots + \hat{\alpha}_{\frac{N+1}{2}} x^{\frac{N-1}{2}}) + y(\hat{\alpha}_{\frac{N+3}{2}} + \cdots + \hat{\alpha}_{N-1} x^{\frac{N-5}{2}} + dx^{\frac{N-3}{2}})$
  end if
  $\hat{S} \leftarrow \text{Algorithm}_7(N, n_t)$
  $\hat{S} = \hat{S}(\alpha_1, \cdots, \alpha_{N-1}, a_t)(x,y)$
  $S \leftarrow \hat{S}((\hat{\alpha}_1/d)(a_1, \cdots, a_{t-1}), \cdots, (\hat{\alpha}_{N-1}/d)(a_1, \cdots, , a_{t-1}), a_t)(x,y)$
  Remove denominators from S
end if
return S

Theorem 75. *Given the input parameters $t, n_1, \cdots, n_t \in \mathbb{Z}_{\geq 2}$, we have that the polynomial S computed in Algorithm 8 is a $(t, n_1, \cdots, n_t)$-generalized summation polynomial, which verifies property (3.3) in a nonempty Zarisky open set $\mathcal{U}_S$.*
Moreover, there exists a nonempty Zarisky open set $\mathcal{U}'_S \subseteq \mathcal{U}_S \subseteq \overline{\mathbb{K}}^{(\sum_{i=1}^{t} n_i)-t}$ such that, for all $\overline{a}_i \in \overline{\mathbb{K}}^{n_i-1}$, with $i \in \{1, \cdots, t\}$, and $(\overline{a}_1, \cdots, \overline{a}_t) \in \mathcal{U}'_S$, the curve

$$S(\overline{a}_1, \cdots, \overline{a}_t)(x, y) = 0$$

has exactly $\prod_{i=1}^{t} n_i$ affine points of intersection with E. This means that we have

$$P_{i_1}(\overline{a}_1) \oplus \cdots \oplus P_{i_t}(\overline{a}_t) \neq \mathcal{O}$$

for each $i_j \in \{1, \cdots, n_j\}$, $j \in \{1, \cdots, t\}$.
Finally, there is a nonempty Zarisky open set $\mathcal{U}''_S \subseteq \mathcal{U}'_S$ such that, for all $\overline{a}_i \in \overline{\mathbb{K}}^{n_i-1}$, with $i \in \{1, \cdots, t\}$, and $(\overline{a}_1, \cdots, \overline{a}_t) \in \mathcal{U''}_S$, we have

$$S_2(\overline{a}_1, \cdots, \overline{a}_t)(x_{P_{i_1}(\overline{a}_1) \oplus \cdots \oplus P_{i_t}(\overline{a}_t)}) \neq 0$$

for each $i_j \in \{1, \cdots, n_j\}$, $j \in \{1, \cdots, t\}$.

Proof. We prove the theorem by induction on t. For $t = 2$, the thesis follows from Theorem 71 and Remark 72.
Let now $t > 2$. By the induction step, the polynomial T computed in line 4 is a $(t-1, n_1, \cdots, n_{t-1})$-generalized summation polynomial, which verifies property (3.3) in a nonempty Zarisky open set $\mathcal{U}_T \subseteq \overline{\mathbb{K}}^{(\sum_{i=1}^{t-1} n_i)-t+1}$.
Moreover, there exists a nonempty Zarisky open set $\mathcal{U}'_T \subseteq \mathcal{U}_T$ such that

$$P_{i_1}(\overline{a}_1) \oplus \cdots \oplus P_{i_{t-1}}(\overline{a}_{t-1}) \neq \mathcal{O}$$

for each $(\overline{a}_1, \cdots, \overline{a}_{t-1}) \in \mathcal{U}'_T$, $i_j \in \{1, \cdots, n_j\}$, $j \in \{1, \cdots, t-1\}$.
Finally, there is a nonempty Zarisky open set $\mathcal{U}''_T \subseteq \mathcal{U}'_T$ such that, for all $(\overline{a}_1, \cdots, \overline{a}_{t-1}) \in \mathcal{U}''_T$, we have

$$T_2(\overline{a}_1, \cdots, \overline{a}_{t-1})(x_{P_{i_1}(\overline{a}_1) \oplus \cdots \oplus P_{i_{t-1}}(\overline{a}_{t-1})}) \neq 0$$

for each $i_j \in \{1, \cdots, n_j\}$, $j \in \{1, \cdots, t-1\}$.
Take $N = \prod_{i=1}^{t-1} n_i$ as in line 5 of the algorithm. Then, by Proposition 62 and Remark 65, T is a polynomial of the form of line 7, if N is even, and of the form of line 9 if N is odd, where $\hat{\alpha}_1, \cdots, \hat{\alpha}_{N-1}, d$ are polynomials of $\mathbb{K}[a_1, \cdots, a_{t-1}]$.
Now take $\hat{S}$ as in line 11. It is a $(2, N, n_t)$-generalized summation polynomial

$$\hat{S} \in \mathbb{K}[\alpha_1, \cdots, \alpha_{N-1}, a_t](x, y),$$

that verifies property (3.3) in $\mathcal{U}_{\hat{S}}$, where

$$\mathcal{U}_{\hat{S}} = (\mathcal{U}_{1,\hat{S}} \times \overline{\mathbb{K}}^{n_t-1}) \cap (\overline{\mathbb{K}}^{N-1} \times \mathcal{U}_{2,\hat{S}}) \cap \mathcal{U}_{3,\hat{S}}$$

is defined as in (3.8). More precisely, $\mathcal{U}_{1,\hat{S}}$, $\mathcal{U}_{2,\hat{S}}$ and $\mathcal{U}_{3,\hat{S}}$ are the Zarisky open sets $\mathcal{U}_1$, $\mathcal{U}_2$ and $\mathcal{U}_3$ defined in Lemma 70, where n_1 is replaced by N and n_2 is replaced by n_t. We define the set $\mathcal{U}_S$ as follows.

$$\mathcal{U}_S = \{(\overline{a}_1, \cdots, \overline{a}_{t-1}, \overline{a}_t) \in \mathcal{U}''_T \times \overline{\mathbb{K}}^{n_t-1} : ((\hat{\alpha}_1/d)(\overline{a}_1, \cdots, \overline{a}_{t-1}), \cdots, (\hat{\alpha}_{N-1}/d)(\overline{a}_1, \cdots, \overline{a}_{t-1}), \overline{a}_t) \in \mathcal{U}_{\hat{S}}\}.$$

Notice that, for each $(\overline{a}_1, \cdots, \overline{a}_{t-1}) \in \mathcal{U}''_T$, we have that $d(\overline{a}_1, \cdots, \overline{a}_{t-1}) \neq 0$, by Proposition 62 and by definition of $\mathcal{U}''_T$. Then, removing denominators, the set $\mathcal{U}_S$ is defined by the non-vanishing of polynomial equations. Therefore, $\mathcal{U}_S$ is a Zarisky open set of $\overline{\mathbb{K}}^{(\sum_{i=1}^{t} n_i)-t}$. Moreover, for each $(\overline{a}_1, \cdots, \overline{a}_{t-1}) \in \mathcal{U}''_T$, we have that

$$((\hat{\alpha}_1/d)(\overline{a}_1, \cdots, \overline{a}_{t-1}), \cdots, (\hat{\alpha}_{N-1}/d)(\overline{a}_1, \cdots, \overline{a}_{t-1})) \in \mathcal{U}_{1,\hat{S}},$$

by definition of $\mathcal{U}''_T$ and point 1 of Lemma 68.
In addition, for each $(\overline{a}_1, \cdots, \overline{a}_{t-1}) \in \mathcal{U}''_T$, there exists $\overline{a}_t \in \mathcal{U}_{2,\hat{S}}$ such that

$$((\hat{\alpha}_1/d)(\overline{a}_1, \cdots, \overline{a}_{t-1}), \cdots, (\hat{\alpha}_{N-1}/d)(\overline{a}_1, \cdots, \overline{a}_{t-1}), \overline{a}_t) \in \mathcal{U}_{3,\hat{S}},$$

by point 5 of Lemma 68. Hence, since $\mathcal{U}''_T$ is nonempty, also $\mathcal{U}_S$ is nonempty.
Now take $\mathcal{U}'_{\hat{S}}$ and $\mathcal{U}''_{\hat{S}}$ as in (3.9) and (3.10) respectively. Let

$$\mathcal{U}'_S = \{(\overline{a}_1, \cdots, \overline{a}_{t-1}, \overline{a}_t) \in \mathcal{U}_S : ((\hat{\alpha}_1/d)(\overline{a}_1, \cdots, \overline{a}_{t-1}), \cdots, (\hat{\alpha}_{N-1}/d)(\overline{a}_1, \cdots, \overline{a}_{t-1}), \overline{a}_t) \in \mathcal{U}'_{\hat{S}}\},$$

$$\mathcal{U}''_S = \{(\overline{a}_1, \cdots, \overline{a}_{t-1}, \overline{a}_t) \in \mathcal{U}'_S : ((\hat{\alpha}_1/d)(\overline{a}_1, \cdots, \overline{a}_{t-1}), \cdots, (\hat{\alpha}_{N-1}/d)(\overline{a}_1, \cdots, \overline{a}_{t-1}), \overline{a}_t) \in \mathcal{U}''_{\hat{S}}\}.$$

With the same arguments that we used to prove that $\mathcal{U}_S$ is a nonempty Zarisky open set, we can prove that both $\mathcal{U}'_S$ and $\mathcal{U}''_S$ are nonempty Zarisky open sets. It is now straightforward to verify, by induction on t, that the polynomial computed in Algorithm 8 is a $(t, n_1, \cdots, n_t)$-generalized summation polynomial, which verifies property (3.3) in $\mathcal{U}_S$, and that $\mathcal{U}'_S, \mathcal{U}''_S$ are the nonempty Zarisky open subsets of $\mathcal{U}_S$ required by the theorem. □

With Algorithm 7 and Algorithm 8, we give a procedure to compute $(t, n_1, \cdots, n_t)$-generalized summation polynomials for any choice of parameters $t, n_1, \cdots, n_t \in \mathbb{Z}_{\geq 2}$. So the following corollary is a straightforward consequence of Theorem 71 and Theorem 75.

Corollary 76. *For any $t, n_1, \cdots, n_t \in \mathbb{Z}_{\geq 2}$, there exist $(t, n_1, \cdots, n_t)$-generalized summation polynomials.*

3.4 Degree of generalized summation polynomials in the parameters

We give now a result on the degree of generalized summation polynomials in each parameter. We will use this result in Chapter 5.

Theorem 77. *Let $t, n_1, \cdots, n_t \in \mathbb{Z}_{\geq 2}$. Let $S \in \mathbb{K}[a_1, \cdots, a_t][x, y]$ be a $(t, n_1, \cdots, n_t)$-generalized summation polynomial as in Definition 64. Then, for each $i \in \{1, \cdots, t\}$ and for each $j \in \{1, \cdots, n_i - 1\}$, one has that the degree of S in the parameter $a_{i,j}$ is*

$$\deg_{a_{i,j}} S = \prod_{h \in \{1, \cdots t\} \setminus \{i\}} n_h.$$

Proof. We prove first that $\deg_{a_{1,1}} S = \prod_{h=2}^{t} n_h$. We follow Notation 63 and Notation 67. Let $\mathcal{U}_S$ be the nonempty Zarisky open set in which S verifies the property (3.3). Write $h_1 \in \mathbb{K}[a_1][x, y]$ in the form

$$h_1 = a_{1,1} + (h_{1,1} - a_{1,1} + y h_{1,2}) = a_{1,1} + k(a_{1,2}, \cdots, a_{1,n_1-1})(x, y),$$

for some $k \in \mathbb{K}[a_{1,2}, \cdots, a_{1,n_1-1}](x, y)$. Then, if we fix parameters $\overline{a}_{1,2}, \cdots, \overline{a}_{1,n_1-1} \in \overline{\mathbb{K}}$, and a zero $T = (x_T, y_T)$ on h_1, we have that $a_{1,1}$ is uniquely determined as

$$a_{1,1,T} = -k(\overline{a}_{1,2}, \cdots, \overline{a}_{1,n_1-1})(x_T, y_T).$$

For $(\overline{a}_{1,2}, \cdots, \overline{a}_{1,n_1-1}) \in \overline{\mathbb{K}}^{n_1-2}$, $\overline{a}_i \in \overline{\mathbb{K}}^{n_i-1}$ for $i \in \{2, \cdots, t\}$, and $R = (x_R, y_R) \in E$, we follow the notation below.

- Let $S_{1,1} = S(\overline{a}_{1,2}, \cdots, \overline{a}_{1,n_1-1}, \overline{a}_2, \cdots, \overline{a}_t, x_R, y_R)(a_{1,1}) \in \overline{\mathbb{K}}[a_{1,1}]$.
- For $i_j \in \{1, \cdots n_j\}$, $j \in \{2, \cdots, t\}$, let

$$T_{i_2,\cdots,i_t} = R - (P_{i_2}(\overline{a}_2) \oplus \cdots \oplus P_{i_t}(\overline{a}_t)).$$

We choose now $(\overline{a}_{1,2}, \cdots, \overline{a}_{1,n_1-1}) \in \overline{\mathbb{K}}^{n_1-2}$, $\overline{a}_i \in \overline{\mathbb{K}}^{n_i-1}$ for $i \in \{2, \cdots, t\}$, and $R = (x_R, y_R) \in E$ such that the following conditions are verified.

1. $\deg_{a_{1,1}} S = \deg(S_{1,1})$ (that is, remove the exceptional cases in which the second degree is strictly smaller).
2. For all $i_2, \cdots, i_t$, with $i_j \in \{1, \cdots, n_j\}$, $j \in \{2, \cdots, t\}$, we have that $T_{i_2,\cdots,i_t} \neq \mathcal{O}$, that is, $R \neq P_{i_2}(\overline{a}_2) \oplus \cdots \oplus P_{i_t}(\overline{a}_t)$.
3. For all $(i_2, \cdots, i_t) \neq (\hat{i}_2, \cdots, \hat{i}_t)$, with $i_j, \hat{i}_j \in \{1, \cdots, n_j\}$, $j \in \{2, \cdots, t\}$, one has that $a_{1,1,T_{i_2,\cdots,i_t}} \neq a_{1,1,T_{\hat{i}_2,\cdots,\hat{i}_t}}$.
4. For all $\hat{a}_{1,1} \in \overline{\mathbb{K}}$ such that $S_{1,1}(\hat{a}_{1,1}) = 0$, we have $(\hat{a}_{1,1}, \overline{a}_{1,2}, \cdots, \overline{a}_{1,n_1-1}, \overline{a}_2, \cdots, \overline{a}_t) \in \mathcal{U}_S$.

Notice that we can always choose $(\overline{a}_{1,2}, \cdots, \overline{a}_{1,n_1-1}) \in \overline{\mathbb{K}}^{n_1-2}$, $\overline{a}_i \in \overline{\mathbb{K}}^{n_i-1}$ for $i \in \{2, \cdots, t\}$, and $R = (x_R, y_R) \in E$ such that the conditions 1, 2, 3 and 4 written above are verified, since $\overline{\mathbb{K}}$ is infinite, as it is algebraically closed.
We have that $\hat{a}_{1,1} \in \overline{\mathbb{K}}$ is a root of $S_{1,1}$, that is $S_{1,1}(\hat{a}_{1,1}) = 0$, if and only if R is a zero of $S(\hat{a}_{1,1}, \overline{a}_{1,2}, \cdots, \overline{a}_{1,n_1-1}, \overline{a}_2, \cdots, \overline{a}_t)(x, y)$. By definition of $(t, n_1, \cdots, n_t)$-generalized summation polynomial, since condition 4 holds true, this is equivalent to say that $T_{i_2,\cdots,i_t}$ is a zero of $h_1(\hat{a}_{1,1}, \overline{a}_{1,2}, \cdots, \overline{a}_{1,n_1-1})(x, y)$ for some $i_2, \cdots, i_t$ with $i_j \in \{1, \cdots, n_j\}$, $j \in \{2, \cdots, t\}$. Hence

$$\hat{a}_{1,1} \in \mathcal{A} = \{a_{1,1,T_{i_2,\cdots,i_t}} : i_j \in \{1, \cdots, n_j\}, j \in \{2, \cdots, t\}\}.$$

By condition 2 and condition 3, the cardinality of $\mathcal{A}$ is equal to $\prod_{h=2}^{t} n_h$.
So $\deg(S_{1,1}(a_{1,1})) = \prod_{h=2}^{t} n_h$, and by condition 1 also $\deg_{a_{1,1}} S = \prod_{h=2}^{t} n_h$.
For all other parameter-variables, choose each $\overline{a}_{i,j} \in \overline{\mathbb{K}}$ and the point R such that all the x_T, y_T are not zero. For the rest, the proof is analogous. □

Example 78. By Remark 66, the $(t+1)$-th Semaev's summation polynomial $f_{t+1}(a_1, \cdots, a_t, x)$ is a $(t, n_1, \cdots, n_t)$-generalized summation polynomial with $n_i = 2$ for all $i \in \{1, \cdots, t\}$.
One has that f_{t+1} has degree $2^{(t+1)-2} = 2^{t-1}$ in each parameter a_i (see Theorem 37).
The $(2, 3, 3)$-generalized summation polynomial that we have computed in Example 73 following Algorithm 7 has degree 3 in each parameter.

3.5 Generalized summation polynomials for cryptography

We saw in the precedent chapters that Semaev's summation polynomials have interesting applications to cryptography. More precisely, in Section 1 of Chapter 2, we use these polynomials to give an optimal representation for trace-zero subgroups of twisted Edwards curves. A similar strategy is applied in [47] in the case of elliptic curves in short Weierstrass form. Furthermore, in Section 5 of Chapter 1, we explain how summation polynomials can be used to perform an index calculus attack, on elliptic groups $E(\mathbb{F}_{q^n})$ as well as on trace-zero subgroups T_n, according to [46], [30], [31], [49]. The first example is an example of constructive application of summation polynomials to cryptography, that is, an application that could improve the efficiency of the given cryptosystem. The second example is a destructive cryptographic application, since it gives a strategy to attack the DLP, on which is based the security of the cryptosystem itself.

As in the case of classical summation polynomials, we will see in the next two chapters that generalized summation polynomials can be interesting cryptographic tools both from the constructive and the destructive point of view. More precisely, in Chapter 4, we use the $(2,3,3)$-generalized summation polynomial that we computed in Example 73 to perform an original scalar multiplication algorithm in the trace-zero subgroup T_3. This is an example of constructive cryptographic application of generalized summation polynomials. On the other hand, in Chapter 5, we give a destructive application of generalized summation polynomials to cryptosystems. In fact, we use such polynomials in the relation search step of a new variant of index calculus, to attack the DLP in trace-zero subgroups of elliptic curves.

The scheme below summarizes the basic ideas and the cryptographic applications of generalized summation polynomials. It is analogous to Scheme 5 dealing with Semaev's summation polynomials.

Scheme 8.

Generalized summation polynomials of elliptic curves.

Idea.

- Generalize summation polynomials from vertical lines to generic zero-locus of elliptic curves.

Constructive cryptographic applications.

- Scalar multiplication in T_3 with optimal coordinates (Chapter 4).

Destructive cryptographic applications.

- Index calculus in T_n (Chapter 5).

Chapter 4

Scalar product in the degree 3 trace-zero subgroup

In this chapter, we give an algorithm to perform scalar multiplication in the degree three trace-zero subgroup of an elliptic curve, using optimal coordinates to represent the elements of the subgroup. We make use of the coordinates of the optimal representation for trace-zero subgroups proposed in [49]. Let T_3 be the degree three trace-zero subgroup of an elliptic curve E defined over a finite field $\mathbb{F}_q$. We represent each point of this subgroup via the two coefficients, in $\mathbb{F}_q$, of the equation of the line that passes through the point and its Frobenius conjugates. Our algorithm computes scalar multiplication in T_3 using these coordinates. It takes as input an integer m and the line through $P \in T_3$ and its Frobenius conjugates. It returns as output the line through the point mP and its Frobenius conjugates.

We saw in Section 1.4.2 that, if we want to increase the security of a cryptosystem on the group $E(\mathbb{F}_q)$ of $\mathbb{F}_q$-rational points of E, we can enlarge the group under consideration by taking the group $E(\mathbb{F}_{q^n})$ of $\mathbb{F}_{q^n}$-rational points of E, with $n > 1$. However, if n is an odd prime, we obtain the same increase in the level of security if we take the trace-zero subgroup $T_n \subseteq E(\mathbb{F}_{q^n})$ (see Proposition 45). This means that we can work in the smaller group T_n, rather than in $E(\mathbb{F}_{q^n})$, without losing security. Moreover, we saw in Section 1.4.2 that one needs n coordinates of $\mathbb{F}_q$ to optimally represent a $\mathbb{F}_{q^n}$-rational point of E. On the other hand, only $n-1$ coordinates of $\mathbb{F}_q$ are required to give an optimal representation of trace-zero elements of T_n. Therefore, the advantage of working in T_n rather than in the whole group $E(\mathbb{F}_{q^n})$ is that, if we use an optimal representation for trace-zero elements, we obtain optimal data storage for level of security. Nevertheless, we pointed out that the use of an optimal representation for trace-zero elements is a real advantage in cryptographic applications only if it is integrated with an efficient performance of the arithmetic in the group. For practical cryptographic applications, such as the Diffie-Hellman key exchange, one is especially interested in the operation of scalar multiplication. Therefore, the aim is to combine the use of an optimal representation for trace-zero elements with efficient algorithms for scalar multiplication in the group.
We recall that there are two ways to approach this task. One can compute scalar multiplication in $E(\mathbb{F}_{q^n})$ and use compression and decompression algorithms to go back and forth between the usual coordinates in $E(\mathbb{F}_{q^n})$ and the compressed coordinates in T_n. The alternative is to compute scalar multiplication directly in compressed coordinates in T_n. In Section 1.4.2, we refer to the first approach as non-compressed scalar multiplication in T_n. In the mentioned section, we explain how the Frobenius endomorphism of the elliptic curve is used to speed up non-compressed scalar multiplication, via the strategy of

Frobenius reduction. On the other hand, the algorithm that we propose in this chapter computes scalar multiplication in T_3 following the second approach. This means that it makes direct use of optimal coordinates in the trace-zero, without performing compression and decompression of points. To the extent of our knowledge, it is the first algorithm to compute scalar multiplication in T_3 with this direct approach in optimal coordinates.

Such approach for scalar multiplication is used by the Montgomery's ladder algorithm for x-only scalar multiplication in $E(\mathbb{F}_{q^n})$, that we recall in Section 1.4.2 (Algorithm 1). This method uses the optimal representation (1.12) for $\mathbb{F}_{q^n}$-rational points of E. Each point is given via its x-coordinate. The x-coordinate is in turn represented via a n-tuple of coefficients of $\mathbb{F}_q$, after choosing a base of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$. Given the optimal representation x_P of an element $P = (x_P, y_P) \in E(\mathbb{F}_{q^n})$, and an integer m, the algorithm computes the optimal representation x_{mP} of the scalar product $mP = (x_{mP}, y_{mP})$. The computation does not require compression and decompression of the involved points. In other words, this algorithm takes as input the vertical line $x = x_P$ through P, and an integer m, and it gives as output the vertical line $x = x_{mP}$ through mP. Notice the analogy with the algorithm that we give here, which works with (non vertical) lines through the Frobenius conjugates of points of T_3.

To perform the computation, our method makes use of the $(2, 3, 3)$-generalized summation polynomial that we computed in Example 73 of Chapter 3. So it is the first example of cryptographic application of generalized summation polynomials. Moreover, the algorithm adapts the strategy of Frobenius reduction that speeds up non-compressed scalar multiplication in trace-zero subgroups. Hence, we can maintain the advantages of such a strategy, even performing the operation directly in compressed coordinates.

The chapter is organized as follows. In Section 4.1 we establish the notation, and give some preliminaries on the degree three trace-zero subgroup of an elliptic curve. We also present some procedures for computation, that will be used in the subsequent algorithms. In Section 4.2 we give our algorithm for scalar multiplication. Subsection 4.2.1 contains a subalgorithm that will be called in the main algorithms, and a lemma which allows us to deal with special cases. In Subsection 4.2.2 we propose a Montgomery-ladder-style algorithm which computes scalar multiplication in T_3. The algorithm makes use of the subalgorithm of Subsection 4.2.1. Subsection 4.2.3 contains the algorithm that computes scalar multiplication in T_3. The algorithm exploits the properties of the Frobenius endomorphism to optimize the Montgomery-ladder-style algorithm of Subsection 4.2.2.

4.1 Preliminaries, notations and formulas

Let $\mathbb{F}_q$ be a finite field of characteristic different from 2 and 3. Let E be an elliptic curve defined over $\mathbb{F}_q$, written in short Weierstrass form

$$E : y^2 z = f(x, z) = x^3 + Axz^2 + Bz^3.$$

We refer to Section 1.3 and Section 1.4 for basic notions about elliptic curves and trace-zero subgroups. We use the notation of these sections. We denote by $\oplus$ the operation of addition between points of E. We denote by $\mathcal{O} = [0, 1, 0]$ the neutral element of the operation. Moreover, for each point $P \in E$, we denote by $-P$ the inverse of P with respect to $\oplus$. Let $E(\mathbb{F}_{q^3})$ be the group of $\mathbb{F}_{q^3}$-rational points of E. Let $T_3 \subseteq E(\mathbb{F}_{q^3})$ be the degree three trace-zero subgroup of E. We denote by φ the Frobenius endomorphism of the curve E.

Let h be the polynomial

$$h = h(\alpha_0, \alpha_1)(x, y) = y - (\alpha_0 + \alpha_1 x) \in \mathbb{F}_q[\alpha_0, \alpha_1][x, y].$$

By Proposition 62 and [49, Corollary 4.2], for each $P = (x_P, y_P) \in T_3$, there exists a unique $(\overline{\alpha}_0, \overline{\alpha}_1) \in \mathbb{F}_q^2$ such that

$$h_P(x, y) = h(\overline{\alpha}_0, \overline{\alpha}_1)(x, y) = y - (\overline{\alpha}_0 + \overline{\alpha}_1 x) = 0$$

is the line through P, $\varphi(P)$, $\varphi^2(P)$. This is equivalent to saying that

$$\mathrm{div}(h_P(x, y)) = P + \varphi(P) + \varphi^2(P) - 3\mathcal{O}.$$

Moreover, notice that

$$h_{-P} = -h_P(x, -y) = h(-\overline{\alpha}_0, -\overline{\alpha}_1) = y + (\overline{\alpha}_0 + \overline{\alpha}_1 x).$$

Following [49], we take the optimal representation $\mathcal{R}_3 = (\mathcal{R}_{3,q,E})_{q,E}$ for the family of degree three trace-zero subgroups $\mathcal{G}_3 = (T_3 \subseteq E(\mathbb{F}_{q^3}))_{q,E}$, with

$$R_{3,q,E} : T_3 \setminus \{\mathcal{O}\} \longrightarrow \mathbb{F}_{q^2},\ P \mapsto (-\overline{\alpha}_0, -\overline{\alpha}_1).$$

This means that we represent an element $P \in T_3 \setminus \{\mathcal{O}\}$ via the coefficients $(-\overline{\alpha}_0, -\overline{\alpha}_1)$ of h_P.

Notice that the representation $\mathcal{R}_3$ identifies each point with its Frobenius conjugates. As a consequence, addition in compressed coordinates is not well-defined. The polynomials h_P and h_Q do not determine $h_{P \oplus Q}$. However, scalar multiplication is well-defined. Namely, given the line $h_P = 0$ and an integer m, the line $h_{mP} = 0$ through mP and its Frobenius conjugates is uniquely determined. Observe the analogy with the representation of points of E via their x-coordinates. The integer m and the x-coordinate of a point $P \in E$ determine the x-coordinate of mP. However, the x-coordinates of P and Q do not determine the x-coordinate of the point $P \oplus Q$.

We give some procedures for computation in the optimal coordinates of T_3 mentioned just above, that we use in the subsequent algorithms. We use the notation below.

Notation 79. Let α_0, α_1 be two parameters. For $i \in \{1, 2, 3\}$, let

$$P_i = P_i(\alpha_0, \alpha_1) = (x_{P_i}(\alpha_0, \alpha_1), y_{P_i}(\alpha_0, \alpha_1))$$

such that the following fact holds. For each $P = (x_P, y_P) \in T_3$, with

$$h_P(x, y) = h(\overline{\alpha}_0, \overline{\alpha}_1)(x, y) = y - (\overline{\alpha}_0 + \overline{\alpha}_1 x),\ \overline{\alpha}_0, \overline{\alpha}_1 \in \mathbb{F}_q,$$

we have

$$P_i(\overline{\alpha}_0, \overline{\alpha}_1) = \varphi^{i-1}(P).$$

Notice that the pair $(\overline{\alpha}_0, \overline{\alpha}_1) \in \mathbb{F}_q^2$ depends on trace-zero point P that we take. We write $(\overline{\alpha}_0, \overline{\alpha}_1)$ instead of $((\overline{\alpha}_0)_P, (\overline{\alpha}_1)_P)$ for ease of notation.

Doubling and tripling formulas in T_3. *Doubling.* Following Procedure 1, we were able to write explicit formulas for the coefficients of h_{2P} in terms of the coefficients of h_P. More precisely, we performed Procedure 1 to compute a polynomial

$$h_2(\alpha_0, \alpha_1)(x, y) = c(\alpha_0, \alpha_1)y - (u_0(\alpha_0, \alpha_1) + u_1(\alpha_0, \alpha_1)x) \in \mathbb{F}_q[\alpha_0, \alpha_1][x, y], \tag{4.1}$$

where $c(\alpha_0, \alpha_1), u_0(\alpha_0, \alpha_1), u_1(\alpha_0, \alpha_1) \in \mathbb{F}_q[\alpha_0, \alpha_1]$. This polynomial is such that, for each $P \in T_3 \setminus \{\mathcal{O}\}$, with $h_P(x, y) = h(\overline{\alpha}_0, \overline{\alpha}_1)(x, y) = y - (\overline{\alpha}_0 + \overline{\alpha}_1 x)$, $\overline{\alpha}_0, \overline{\alpha}_1 \in \mathbb{F}_q$, we have

$$\mathrm{div}(h_2(\overline{\alpha}_0, \overline{\alpha}_1)(x, y)) = 2P + \varphi(2P) + \varphi^2(2P) - 3\mathcal{O}. \tag{4.2}$$

This is equivalent to saying that $h_2(\overline{\alpha}_0, \overline{\alpha}_1)(x, y) = h_{2P}$ up to multiplication by a nonzero constant, if $2P \neq \mathcal{O}$, and $h_2(\overline{\alpha}_0, \overline{\alpha}_1)(x, y) \in \mathbb{F}_q \setminus \{0\}$ otherwise. Notice that $2P = \mathcal{O}$ if and only if $h_P = y$.
The computed coefficients $c(\alpha_0, \alpha_1), u_0(\alpha_0, \alpha_1), u_1(\alpha_0, \alpha_1) \in \mathbb{F}_q[\alpha_0, \alpha_1]$ can be found in Section 2.1 of the appendix.

Procedure 1. Procedure to compute the polynomial $h_2(\alpha_0, \alpha_1)(x, y)$.

1: **for** $i \in \{1, 2, 3\}$ ▷ $t_i = 0$ tangent to E in P_i, t_i polynomial in the variables x_{P_i}, y_{P_i}, x, y
2: $t_i(x_{P_i}, y_{P_i}, x, y) \leftarrow (3x_{P_i}^2 + A)x - 2y_{P_i}y + (2y_{P_i}^2 - (3x_{P_i}^2 + A)x_{P_i})$
3: **end for**
4: $T(x_{P_1}, x_{P_2}, x_{P_3}, y_{P_1}, y_{P_2}, y_{P_3}, x, y) \leftarrow \prod_{i=1}^{3} t_i$
5: **for** $i \in \{1, 2, 3\}$
6: Replace y_{P_i} by $(\alpha_1 x_{P_i} + \alpha_0)$ in T
7: **end for**
8: Write $T(x_{P_1}, x_{P_2}, x_{P_3})$ as a function of the elementary symmetric polynomials e_1, e_2, e_3
9: $E_1 \leftarrow \alpha_1^2$, $E_2 \leftarrow A - 2\alpha_0\alpha_1$, $E_3 \leftarrow \alpha_0^2 - B$
10: **for** $i \in \{1, 2, 3\}$
11: Replace e_i by E_i in T
12: **end for** ▷ Now $T = T(\alpha_0, \alpha_1)(x, y) \in \mathbb{F}_q[\alpha_0, \alpha_1][x, y]$
13: Use equality $h_2(x, y) = T(x, -y)/((-h(x, -y))^2) \mod y^2 - f(x, 1)$, removing denominators
14: **return** h_2

Theorem 80. *Procedure* 1 *is correct.*

Proof. We follow the same arguments as in the proof of Theorem 71.
It is easy to verify that Procedure 1 outputs a polynomial $h_2(\alpha_0, \alpha_1)(x, y) \in \mathbb{F}_q[\alpha_0, \alpha_1][x, y]$ of the form (4.1).
Now we prove that, for each $P \in T_3 \setminus \{\mathcal{O}\}$, with $h_P(x, y) = h(\overline{\alpha}_0, \overline{\alpha}_1)(x, y)$, we have that $h_2(\overline{\alpha}_0, \overline{\alpha}_1)(x, y)$ satisfies (4.2).
Suppose first that $2P \neq \mathcal{O}$, that is $h_P \neq y$ and $(\overline{\alpha}_0, \overline{\alpha}_1) \neq (0, 0)$. Let $t_i(x_{P_i}, y_{P_i}, x, y)$ be the polynomial computed in line 2. Since $2P \neq \mathcal{O}$, we have that

$$t_i(\overline{\alpha}_0, \overline{\alpha}_1)(x, y) = t_i(x_{P_i}(\overline{\alpha}_0, \overline{\alpha}_1), y_{P_i}(\overline{\alpha}_0, \overline{\alpha}_1), x, y) = 0$$

is the tangent to E at $P_i(\overline{\alpha}_0, \overline{\alpha}_1) = \varphi^{i-1}(P)$. Therefore, we have

$$\operatorname{div}(t_i(\overline{\alpha}_0, \overline{\alpha}_1)(x, y)) = \varphi^{i-1}(P) + \varphi^{i-1}(P) + (-(2\varphi^{i-1}(P)) - 3\mathcal{O}.$$

So the polynomial T computed in line 4 is such that

$$T(\overline{\alpha}_0, \overline{\alpha}_1) = T(x_{P_1}(\overline{\alpha}_0, \overline{\alpha}_1), x_{P_2}(\overline{\alpha}_0, \overline{\alpha}_1), x_{P_3}(\overline{\alpha}_0, \overline{\alpha}_1), y_{P_1}(\overline{\alpha}_0, \overline{\alpha}_1), y_{P_2}(\overline{\alpha}_0, \overline{\alpha}_1), y_{P_3}(\overline{\alpha}_0, \overline{\alpha}_1), x, y) = 0$$

is the union of the three tangents to E at $\varphi^{i-1}(P)$, for $i \in \{1, 2, 3\}$.
Furthermore, each $P_i(\overline{\alpha}_0, \overline{\alpha}_1)$ is on the line $h_P = 0$. Hence, we have the equality

$$y_{P_i}(\overline{\alpha}_0, \overline{\alpha}_1) = \overline{\alpha}_0 + \overline{\alpha}_1 x_{P_i}(\overline{\alpha}_0, \overline{\alpha}_1),$$

for $i \in \{1, 2, 3\}$. For $i \in \{1, 2, 3\}$, let $E_i(\alpha_0, \alpha_1) \in \mathbb{F}_q[\alpha_0, \alpha_1]$ be the polynomial defined in line 9. By point 2 of Proposition 62, we have that

$$e_i(x_{P_1}(\overline{\alpha}_0, \overline{\alpha}_1), x_{P_2}(\overline{\alpha}_0, \overline{\alpha}_1), x_{P_3}(\overline{\alpha}_0, \overline{\alpha}_1)) = E_i(\overline{\alpha}_0, \overline{\alpha}_1).$$

So, after line 12, the polynomial $T = T(\alpha_0, \alpha_1)(x, y) \in \mathbb{F}_q[\alpha_0, \alpha_1][x, y]$ is such that $T(\overline{\alpha}_0, \overline{\alpha}_1)(x, y) = 0$ is still the union of the three tangents to E at the points $\varphi^{i-1}(P)$, for $i \in \{1, 2, 3\}$. Therefore, we have

$$\mathrm{div}(T(\overline{\alpha}_0, \overline{\alpha}_1)(x, y)) = \sum_{i=1}^{3} \varphi^{i-1}(P) + \sum_{i=1}^{3} \varphi^{i-1}(P) + \sum_{i=1}^{3} (-2\varphi^{i-1}(P)) - 9P_{\mathcal{O}}.$$

We have also that

$$\mathrm{div}(h_P^2) = \sum_{i=1}^{3} \varphi^{i-1}(P) + \sum_{i=1}^{3} \varphi^{i-1}(P) - 6\mathcal{O}.$$

Hence, we obtain that the polynomial $h_2(\alpha_0, \alpha_1)(x, y)$ computed in line 13 is such that

$$\mathrm{div}(h_2(\overline{\alpha}_0, \overline{\alpha}_1)(x, y)) = 2P + \varphi(2P) + \varphi^2(2P) - 3\mathcal{O}.$$

We have then proved that $h_2(\alpha_0, \alpha_1)(x, y)$ satisfies property (4.2), when $(\overline{\alpha}_0, \overline{\alpha}_1) \neq (0, 0)$. Suppose now that $(\overline{\alpha}_0, \overline{\alpha}_1) = (0, 0)$. This is equivalent to saying that $2P = \mathcal{O}$. One can directly check that $h_2(0, 0)(x, y) \in \mathbb{F}_q \setminus \{0\}$. Hence

$$\mathrm{div}(h_2(0, 0)(x, y) = 0 = \mathcal{O} + \mathcal{O} + \mathcal{O} - 3\mathcal{O} = 2P + \varphi(2P) + \varphi^2(2P) - 3\mathcal{O}.$$

This conclude the proof of the theorem. □

Tripling. Following Procedure 3 below, we wrote explicit formulas for the coefficients of h_{3P} in terms of the coefficients of h_P. More precisely, we performed Procedure 2 to compute a polynomial

$$h_3(\alpha_0, \alpha_1)(x, y) = d(\alpha_0, \alpha_1)y - (v_0(\alpha_0, \alpha_1) + v_1(\alpha_0, \alpha_1)x) \in \mathbb{F}_q[\alpha_0, \alpha_1][x, y], \tag{4.3}$$

where $d(\alpha_0, \alpha_1), v_0(\alpha_0, \alpha_1), v_1(\alpha_0, \alpha_1) \in \mathbb{F}_q[\alpha_0, \alpha_1]$. This polynomial is such that, for each $P \in T_3 \setminus \{\mathcal{O}\}$, with $h_P(x, y) = h(\overline{\alpha}_0, \overline{\alpha}_1)(x, y) = y - (\overline{\alpha}_0 + \overline{\alpha}_1 x)$, $\overline{\alpha}_0, \overline{\alpha}_1 \in \mathbb{F}_q$, we have

$$\mathrm{div}(h_3(\overline{\alpha}_0, \overline{\alpha}_1)(x, y)) = 3P + \varphi(3P) + \varphi^2(3P) - 3\mathcal{O}.$$

This is equivalent to saying that $h_3(\overline{\alpha}_0, \overline{\alpha}_1)(x, y) = h_{3P}$ up to multiplication by a nonzero constant, if $3P \neq \mathcal{O}$, and $h_3(\overline{\alpha}_0, \overline{\alpha}_1)(x, y) \in \mathbb{F}_q \setminus \{0\}$ otherwise.
The computed coefficients $d(\alpha_0, \alpha_1), v_0(\alpha_0, \alpha_1), v_1(\alpha_0, \alpha_1) \in \mathbb{F}_q[\alpha_0, \alpha_1]$ can be found in Section 2.2 of the appendix.

Procedure 2. Procedure to compute the polynomial $h_3(\alpha_0, \alpha_1)(x, y)$

1: **for** $i \in \{1, 2, 3\}$ ▷ doubling formulas for P_i and $\ell_i = 0$ line through P_i, $2P_i$
▷ x_{2P_i} written as a rational function in the variables x_{P_i}, y_{P_i}
2: $x_{2P_i}(x_{P_i}, y_{P_i}) \leftarrow ((3x_{P_i}^2 + A)/2y_{P_i})^2 - 2x_{P_i}$
▷ y_{2P_i} written as a rational function in the variables x_{P_i}, y_{P_i}
3: $y_{2P_i}(x_{P_i}, y_{P_i}) \leftarrow ((3x_{P_i}^2 + A)/2y_{P_i})(x_{P_i} - x_{2P_i}) - y_{P_i}$
▷ ℓ_i written as a rational function in the variables x_{P_i}, y_{P_i}, x, y
4: $\ell_i(x_{P_i}, y_{P_i}, x, y) \leftarrow (y_{2P_i} - y_{P_i})x + (x_{P_i} - x_{2P_i})y + ((x_{2P_i} - x_{P_i})y_{P_i} + (y_{P_i} - y_{2P_i})x_{P_i})$
5: **end for**
6: $L(x_{P_1}, x_{P_2}, x_{P_3}, y_{P_1}, y_{P_2}, y_{P_3}, x, y) \leftarrow \prod_{i=1}^{3} \ell_i$
7: **for** $i \in \{1, 2, 3\}$

8: Replace y_{P_i} by $(\alpha_1 x_{P_i} + \alpha_0)$ in L
9: **end for**
10: Write $L(x_{P_1}, x_{P_2}, x_{P_3})$ as a function of the elementary symmetric polynomials e_1, e_2, e_3
11: $E_1 \leftarrow \alpha_1^2$, $E_2 \leftarrow A - 2\alpha_0\alpha_1$, $E_3 \leftarrow \alpha_0^2 - B$
12: **for** $i \in \{1, 2, 3\}$
13: Replace e_i by E_i in L
14: **end for**
15: Use $h_2(\alpha_0, \alpha_1)(x, y)$ found with Procedure 1, and equality
$h_3(x, y) = L(x, -y)/h(x, -y)h_2(x, -y))) \mod y^2 - f(x, 1)$, removing denominators
16: **return** h_3

Theorem 81. *Procedure* 2 *is correct.*

We omit the proof of Theorem 81, since it is analogous to those of Theorem 80 and of Theorem 71.

The $(2, 3, 3)$-generalized summation polynomial for T_3. As we mentioned before, one cannot compute the polynomial $h_{P\oplus Q}$ from the polynomials h_P and h_Q. Nevertheless, given h_P and h_q, one can compute a polynomial $S_{P,Q}(x, y) \in \mathbb{F}_q[x, y]$, such that

$$\operatorname{div}(S_{P,Q}(x, y)) = \sum_{0\le i,j\le 2} (\varphi^i(P) \oplus \varphi^j(Q)) - 9\mathcal{O}. \tag{4.4}$$

By Proposition 62 and [49, Corollary 4.2], the polynomial $S_{P,Q}$ is unique modulo the equation of the elliptic curve, and up to multiplication by a nonzero constant. Hence, we take the unique $S_{P,Q}(x, y)$ of the form

$$S_{P,Q} = (S_{P,Q})_1 + y(S_{P,Q})_2 = (a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0) + y(b_3x^3 + b_2x^2 + b_1x + b_0), \tag{4.5}$$

where $a_0, \cdots, a_4, b_0, \cdots, b_3 \in \mathbb{F}_q$, and such that $S_{P,Q}(x, y)$ verifies property (4.4).
One can compute $S_{P,Q}$ from h_P and h_Q, using the $(2, 3, 3)$-generalized summation polynomial that we computed in Example 73 of Chapter 3. In fact, we have the following result.

Proposition 82. *Let $S(\alpha_0, \alpha_1, \beta_0, \beta_1)(x, y) \in \mathbb{F}_q[\alpha_0, \alpha_1, \beta_0, \beta_1][x, y]$ be the $(2, 3, 3)$-generalized summation polynomial that we computed in Example 73 of Chapter* 3*, using Algorithm* 7*. Let P, $Q \in T_3 \setminus \{\mathcal{O}\}$, with $h_P = y - (\overline{\alpha}_0 + \overline{\alpha}_1 x)$ and $h_Q = y - (\overline{\beta}_0 + \overline{\beta}_1 x)$. Then we have*

$$S_{P,Q}(x, y) = S(\overline{\alpha}_0, \overline{\alpha}_1, \overline{\beta}_0, \overline{\beta}_1)(x, y).$$

Proof. By construction, the polynomial $S(\overline{\alpha}_0, \overline{\alpha}_1, \overline{\beta}_0, \overline{\beta}_1)(x, y)$ is of the form (4.5). Moreover, since $S(\alpha_0, \alpha_1, \beta_0, \beta_1)(x, y)$ has coefficients in $\mathbb{F}_q$ and $\overline{\alpha}_0, \overline{\alpha}_1, \overline{\beta}_0, \overline{\beta}_1 \in \mathbb{F}_q$, we have that $S(\overline{\alpha}_0, \overline{\alpha}_1, \overline{\beta}_0, \overline{\beta}_1)(x, y) \in \mathbb{F}_q[x, y]$.
By definition of generalized summation polynomial and Theorem 71 of Chapter 3, we have that

$$\operatorname{div}(S(\overline{\alpha}_0, \overline{\alpha}_1, \overline{\beta}_0, \overline{\beta}_1)(x, y)) = \sum_{0\le i,j\le 2} (\varphi^i(P) \oplus \varphi^j(Q)) - 9\mathcal{O},$$

if $(\overline{\alpha}_0, \overline{\alpha}_1, \overline{\beta}_0, \overline{\beta}_1) \in \mathcal{U}_S \subseteq \overline{\mathbb{F}}_q^4$, where $\mathcal{U}_S = (\mathcal{U}_1 \times \overline{\mathbb{F}}_q^2) \cap (\overline{\mathbb{F}}_q^2 \times \mathcal{U}_2) \cap \mathcal{U}_3$ is the nonempty Zarisky open set defined in (3.8), for $\mathbb{K} = \mathbb{F}_q$ and $n_1 = n_2 = 3$.

We claim that $(\overline{\alpha}_0, \overline{\alpha}_1, \overline{\beta}_0, \overline{\beta}_1) \in \mathcal{U}_S$ if and only if $(\overline{\alpha}_0, \overline{\alpha}_1) \neq (\overline{\beta}_0, \overline{\beta}_1)$. In fact, for

$n_1 = n_2 = 3$, we have that $\mathcal{U}_1 = \mathcal{U}_2 = \overline{\mathbb{F}}_q^2$, by point 1 and point 2 of Lemma 70. Moreover, by point 3 of Lemma 70, $(\overline{\alpha}_0, \overline{\alpha}_1, \overline{\beta}_0, \overline{\beta}_1) \in \mathcal{U}_3$ if and only if $P = \varphi^i(Q)$ for some $i \in \{0, 1, 2\}$. This is equivalent to saying $h_P = h_Q$, or $(\overline{\alpha}_0, \overline{\alpha}_1) = (\overline{\beta}_0, \overline{\beta}_1)$. Hence we have proved the claim.

The claim and the discussion above imply the thesis of the theorem for $(\overline{\alpha}_0, \overline{\alpha}_1) \neq (\overline{\beta}_0, \overline{\beta}_1)$. We now prove the thesis in the case $(\overline{\alpha}_0, \overline{\alpha}_1) = (\overline{\beta}_0, \overline{\beta}_1)$. In this case, by Theorem 80, we have that

$$S_{P,Q}(x, y) = h(-\overline{\alpha}_0, -\overline{\alpha}_1)(x, y)^2 h_2(\overline{\alpha}_0, \overline{\alpha}_1)(x, y) \mod y^2 - f(x, 1), \tag{4.6}$$

where $h_2(\alpha_0, \alpha_1)(x, y) \in \mathbb{F}_q[\alpha_0, \alpha_1][x, y]$ is the polynomial computed with Procedure 1. Moreover, one can directly check that

$$S(\alpha_0, \alpha_1, \alpha_0, \alpha_1)(x, y) = h(-\alpha_0, -\alpha_1)(x, y) h_2(\alpha_0, \alpha_1)(x, y) \mod y^2 - f(x, 1). \tag{4.7}$$

Hence, for $(\overline{\alpha}_0, \overline{\alpha}_1) = (\overline{\beta}_0, \overline{\beta}_1)$, we obtain the thesis of the theorem from equalities (4.6) and (4.7). □

We recall that, in Chapter 3, we computed the $(2, 3, 3)$-generalized summation polynomial $S(\alpha_0, \alpha_1, \beta_0, \beta_1)(x, y)$ of Example 73 using Algorithm 7 with input $n_1 = n_2 = 3$. The coefficients of S, namely $a_i(\alpha_0, \alpha_1, \beta_0, \beta_1), b_j(\alpha_0, \alpha_1, \beta_0, \beta_1) \in \mathbb{K}[\alpha_0, \alpha_1, \beta_0, \beta_1]$, for $i \in \{0, \cdots, 4\}$, $j \in \{0, \cdots, 3\}$, can be found in Section 1 of the appendix. In order to be self-contained inside the chapter, we give below the procedure that we followed to compute the $a_i(\alpha_0, \alpha_1, \beta_0, \beta_1), b_j(\alpha_0, \alpha_1, \beta_0, \beta_1)$. The procedure performs the steps of Algorithm 7 for input $n_1 = n_2 = 3$, parameters $a_1 = (-\alpha_0, -\alpha_1)$, $a_2 = (-\beta_0, -\beta_1)$, and $h_1 = h(\alpha_0, \alpha_1)(x, y) = y - (\alpha_0 + \alpha_1 x)$, $h_2 = h(\beta_0, \beta_1)(x, y) = y - (\beta_0 + \beta_1 x)$.

Procedure 3. Procedure to compute the $(2, 3, 3)$-generalized summation polynomial S of Example 73.

---

for $i \in \{1, 2, 3\}$
  for $j \in \{1, 2, 3\}$
    $r_{ij}(x_{P_i}, x_{Q_j}, y_{P_i}, y_{Q_j}, x, y) \leftarrow (y_{Q_j} - y_{P_i})x + (x_{P_i} - x_{Q_j})y + ((x_{Q_j} - x_{P_i})y_{P_i} + (y_{P_i} - y_{Q_j})x_{P_i})$
  end for
end for
$K(x_{P_1}, x_{P_2}, x_{P_3}, y_{P_1}, y_{P_2}, y_{P_3}, x_{Q_1}, x_{Q_2}, x_{Q_3}, y_{Q_1}, y_{Q_2}, y_{Q_3}, x, y) \leftarrow \prod_{1 \leq i,j \leq 3} r_{i,j}$
for $i \in \{1, 2, 3\}$
  Replace y_{P_i} by $(\alpha_1 x_{P_i} + \alpha_0)$ in K
  Replace y_{Q_i} by $(\beta_1 x_{Q_i} + \beta_0)$ in K
end for
Write $K(x_{P_1}, x_{P_2}, x_{P_3})$ as a function of the elementary symmetric polynomials e_1, e_2, e_3
Write $K(x_{Q_1}, x_{Q_2}, x_{Q_3})$ as a function of the elementary symmetric polynomials in s_1, s_2, s_3
$E_1 \leftarrow \alpha_1^2$, $E_2 \leftarrow A - 2\alpha_0\alpha_1$, $E_3 \leftarrow \alpha_0^2 - B$
$S_1 \leftarrow \beta_1^2$, $S_2 \leftarrow A - 2\beta_0\beta_1$, $S_3 \leftarrow \beta_0^2 - B$
for $i \in \{1, 2, 3\}$
  Replace e_i by E_i in R

17: Replace s_i by S_i in R
18: **end for**
19: Use equality $S_1(x) - yS_2(x) = K(x,y)/(h(\alpha_0,\alpha_1)(x,y)^3 h(\beta_0,\beta_1)(x,y)^3) \mod y^2 - f(x,1)$
20: Remove denominators from S
21: **return** S

Notice that, if $P \oplus Q, P \oplus \varphi(Q), P \oplus \varphi^2(Q) \neq \mathcal{O}$, then

$$S_{P,Q} = h_{P\oplus Q} h_{P\oplus\varphi(Q)} h_{P\oplus\varphi^2(Q)} \mod y^2 - f(x,1). \tag{4.8}$$

Moreover, from h_P and $S_{P,Q}$, one can compute the polynomials

$$H_P = f(x,1) - (\overline{\alpha}_1 x + \overline{\alpha}_0)^2,\ \Sigma_{P,Q} = f(x,1)(S_{P,Q})_2^2 - (S_{P,Q})_1^2 \in \mathbb{F}_q[x].$$

In the next lemma we collect a few facts on the polynomial $S_{P,Q}$, that will be useful in the sequel.

Lemma 83. *Let $P, Q \in T_3 \setminus \{\mathcal{O}\}$. Let h_P, H_P, $S_{P,Q}$ and $\Sigma_{P,Q}$ be the polynomials as above. The following equalities hold, up to a nonzero constant:*

1. $H_P = h_P h_{-P} \mod y^2 - f(x,1)$,
2. $H_P = (x - x_P)(x - x_P^q)(x - x_P^{q^2})$,
3. $S_{-P,-Q}(x,y) = S_{P,Q}(x,-y)$,
4. $\Sigma_{P,Q} = S_{P,Q} S_{-P,-Q} \mod y^2 - f(x,1)$,
5. $\Sigma_{P,Q} = \prod_{0\le i,j\le 2}(x - x_{\varphi^i(P)\oplus\varphi^j(Q)})$.

Moreover, the following are equivalent:

6. $(S_{P,Q})_2 = 0$,
7. $b_3 = 0$,
8. $\varphi^i(P) \oplus \varphi^j(Q) = \mathcal{O}$ *for some* i, j,
9. $\mathrm{div}(S_{P,Q}) = (P \oplus (-\varphi(P))) + (\varphi(P) \oplus (-P)) + (P \oplus (-\varphi^2(P))) + (\varphi^2(P) \oplus (-P)) + (\varphi(P) \oplus (-\varphi^2(P))) + (\varphi^2(P) \oplus (-\varphi(P))) - 6\mathcal{O}$.

Proof. *1.* and *2.* follow from point 2 of Proposition 62.
3. Observe that $\mathrm{div}(S_{-P,-Q}) = \sum_{0\le i,j\le 2}(-\varphi^i(P) \oplus (-\varphi^j(Q))) - 9\mathcal{O}$, hence

$$S_{-P,-Q}(x,y) = (S_{P,Q})_1(x) - y(S_{P,Q})_2(x) = S_{P,Q}(x,-y)$$

up to a nonzero constant.
4. and *5.* follow from point 3 of this lemma, and point 2 of Proposition 62.
7. $\Rightarrow$ *8.* If $b_3 = 0$, then $\deg(\Sigma_{P,Q}) \le 8$, hence one of the sums $\varphi^i(P) \oplus \varphi^j(Q)$ must be $\mathcal{O}$.
8. $\Rightarrow$ *9.* If $\varphi^i(P) \oplus \varphi^j(Q) = \mathcal{O}$ for some i and j, then $S_{P,Q} = S_{\varphi^i(P),\varphi^j(Q)} = S_{\varphi^i(P),-\varphi^i(P)} = S_{P,-P}$. Hence, by definition of $S_{P,Q}$, we have that $\mathrm{div}(S_{P,Q}) = (P \oplus (-\varphi(P))) + (\varphi(P) \oplus (-P)) + (P \oplus (-\varphi^2(P))) + (\varphi^2(P) \oplus (-P)) + (\varphi(P) \oplus (-\varphi^2(P))) + (\varphi^2(P) \oplus (-\varphi(P))) + 3\mathcal{O} - 9\mathcal{O}$, from which point 9. follows.
9. $\Rightarrow$ *6.* Since $\mathrm{div}(S_{P,Q}) = (P \oplus (-\varphi(P))) + (\varphi(P) \oplus (-P)) + (P \oplus (-\varphi^2(P))) + (\varphi^2(P) \oplus (-P)) + (\varphi(P) \oplus (-\varphi^2(P))) + (\varphi^2(P) \oplus (-\varphi(P))) - 6\mathcal{O}$, then $S_{P,Q} = (x - x_{P\oplus(-\varphi(P))})(x - x_{P\oplus(-\varphi^2(P))})(x - x_{\varphi(P)\oplus(-\varphi^2(P))}) \in \mathbb{F}_q[x]$. Hence $(S_{P,Q})_2 = 0$. □

How to recover $h_{P\oplus Q}$ from $H_{P\oplus Q}$ and $S_{P,Q}$. Let $P, Q \in T_3 \setminus \{\mathcal{O}\}$, such that $P \oplus Q \neq \mathcal{O}$. We give a procedure to compute the coefficients of $h_{P\oplus Q}(x, y) = y - (\gamma_0 + \gamma_1 x)$ in terms of the coefficients of $H_{P\oplus Q}(x) = x^3 + w_2x^2 + w_1x + w_0$ and the coefficients of $S_{P,Q}$. A straightforward way to do this is computing the coefficients of $h_{P\oplus Q}$ from those of $H_{P\oplus Q}$ up to sign, via the relations $w_2 = -\gamma_1^2$, $w_1 = A - 2\gamma_0\gamma_1$, $w_0 = B - \gamma_0^2$. One can then distinguish $h_{P\oplus Q} = y - (\gamma_0 + \gamma_1 x)$ and $h_{-P\oplus(-Q)} = y + (\gamma_0 + \gamma_1 x)$, since $H_{P\oplus Q} \mid (S_{P,Q})_1 + (\gamma_0 + \gamma_1 x)(S_{P,Q})_2$. This however requires extracting a square root. The next proposition allows us to compute $h_{P\oplus Q}$ from $H_{P\oplus Q}$ and $S_{P,Q}$ more efficiently, by solving a simple linear system. We assume that $(S_{P,Q})_2 \neq 0, H_{P\oplus Q}$ and that $H_{P\oplus Q}$ is irreducible over $\mathbb{F}_q[x]$. Notice that $H_{P\oplus Q}$ is irreducible over $\mathbb{F}_q[x]$ if and only if $P \oplus Q$ does not belong to the group

$$E(\mathbb{F}_q)[3] = \{P \in E(\mathbb{F}_q) : 3P = \mathcal{O}\}.$$

Proposition 84. *Let P, $Q \in T_3 \setminus \{\mathcal{O}\}$. Suppose that $P \oplus Q \notin E(\mathbb{F}_q)[3]$, that Q is not a Frobenius conjugate of $-P$ or $-2P$, and that P is not a Frobenius conjugate of $-2Q$. Let $H_{P\oplus Q} = x^3 + w_2x^2 + w_1x + w_0$ and $h_{P\oplus Q} = y - (\gamma_1 x + \gamma_0)$ with $\gamma_1, \gamma_0, w_2, w_1, w_0 \in \mathbb{F}_q$. Then (γ_1, γ_0) is the unique solution of the linear system whose augmented matrix is*

$$L(H_{P\oplus Q}, S_{P,Q}) = \begin{pmatrix} w_0(w_2 - b_2) & (b_0 - w_0) & w_0a_3 - a_4w_2w_0 - a_0 \\ w_0(w_1 - b_1) & (b_0w_2 - w_0b_2) & w_0a_2 - a_4w_1w_0 - a_0w_2 \\ w_0(w_0 - b_0) & (b_0w_1 - b_1w_0) & w_0a_1 - a_4w_0^2 - a_0w_1 \end{pmatrix}.$$

Proof. Using the fact that $H_{P\oplus Q}|(S_{P,Q})_1 + (\gamma_1 x + \gamma_0)(S_{P,Q})_2$, a simple calculation shows that (γ_1, γ_0) is a solution of the linear system with augmented matrix $L(H_{P\oplus Q}, S_{P,Q})$. Let us prove that the solution is unique. Let (t_1, t_0) be a solution of the linear system with augmented matrix $L(H_{P\oplus Q}, S_{P,Q})$. Let $(x_0, y_0) \in T_3$ be one of the Frobenius conjugates of $P \oplus Q$. Notice that, since $P \oplus Q \notin E(\mathbb{F}_q)[3]$, the three Frobenius conjugates are distinct. By construction, $(S_{P,Q})_1(x_0) + (t_1x_0 + t_0)(S_{P,Q})_2(x_0) = 0$.

We claim that $(S_{P,Q})_2(x_0) \neq 0$. In fact, if $(S_{P,Q})_2(x_0) = 0$, then $(S_{P,Q})_2 = H_{P\oplus Q}$ and $H_{P\oplus Q} \mid (S_{P,Q})_1$. So $S_{P,Q}$ is of the form $S_{P,Q}(x, y) = H_{P\oplus Q}(x)(y + d_0 + d_1x)$, with $d_0, d_1 \in \mathbb{F}_q$. Hence, all the zeroes of $H_{P\oplus Q}(x, y) = H_{P\oplus Q}(x)$ on E are also zeroes of $S_{P,Q}(x, y)$ on E. This implies that $-P \oplus (-Q) = \varphi^i(P) \oplus \varphi^j(Q)$ for some i, j distinct. If $i, j \neq 0$, then $-\varphi^k(P) = P \oplus \varphi^i(P) = -Q \oplus (-\varphi^j(Q)) = \varphi^h(Q)$ for some h, k. Hence P and $-Q$ are Frobenius conjugates. Similarly, Q and $-2P$ are Frobenius conjugates if $i = 0$ and $j \neq 0$, and P and $-2Q$ are Frobenius conjugates if $i = 0$ and $j \neq 0$. This concludes the proof of the claim.

Since $(S_{P,Q})_2(x_0) \neq 0$ by the claim above, then $y_0 = t_1x_0 + t_0$. Hence the line of equation $y - (t_1x + t_0)$ has three points in common with the line of equation $h_{P\oplus Q} = 0$. This implies that $t_1 = \gamma_1$ and $t_0 = \gamma_0$. □

Example 85. Let $q = 1021$ and $\mathbb{F}_{q^3} = \mathbb{F}_q[\zeta]/(\zeta^3 - 5)$. Let E be the elliptic curve over $\mathbb{F}_q$ of equation $y^2z = x^3 + 230xz^2 + 191z^3$. Let $P = (782\zeta^2 + 802\zeta + 45, 979\zeta^2 + 299\zeta + 133)$, $Q = (466\zeta^2 + 528\zeta + 514, 742\zeta^2 + 1016\zeta + 704) \in T_3$, with $h_P = y - (987x + 642)$, $h_Q = y - (729x + 705)$. Using the formulas in the appendix, we can compute:

$$h_{2P} = y - (1000x + 280), \quad h_{3P} = y - (646x + 693)$$

$$S_{P,Q} = (823x^4 + 948x^3 + 709x^2 + 530x + 741) + y(x^3 + 782x^2 + 636x + 100).$$

The matrix from Proposition 84 is:

$$L(H_{P\oplus Q}, S_{P,Q}) = \begin{pmatrix} 809 & 123 & 843 \\ 568 & 823 & 755 \\ 787 & 382 & 388 \end{pmatrix}.$$

Before we compute L, we compute $H_{P\oplus Q} = x^3 + 880x^2 + 123x + 998$. In the next section we discuss how to compute $H_{P\oplus Q}$. Solving the system associated to L we find $h_{P\oplus Q} = y - (65x + 260)$.

4.2 Scalar multiplication in T_3 using compressed coordinates

Throughout this section, we assume that $T_3 = \langle P \rangle$ is cyclic of order p, where p is a prime of cryptographic size. This is the setting of real cryptographic applications. Hence $\varphi(P) = sP$, with $s = (q-1)/(2+q-|E(\mathbb{F}_q)|) \mod p$ (see [6, Section 15.3.1]).
Let m be an integer modulo p. In this section, we develop an efficient algorithm to compute h_{mP} given m and h_P. In order to do this, in Subsection 4.2.1, we give a subalgorithm that we use within the main algorithm, as well as a lemma which helps us deal with special cases. In Subsection 4.2.2, we give a Montgomery-ladder-style algorithm that computes h_{mP} from m and h_P. Finally, in Subsection 4.2.3, we apply the usual Frobenius endomorphism strategy to speed up our algorithm from Section 4.2.2. This gives our main algorithm to compute scalar multiplication in T_3 using compressed coordinates.

4.2.1 Subalgorithm and special cases

Throughout this subsection, m is an integer such that $0 < m < p$. Because of the doubling formulas in the appendix, we may assume that m is odd.

Notation 86. Let m_1, m_2, n_1, n_2 be integers such that $m_1 + m_2 = n_1 + n_2 = m$. For $i \in \{0,1,2\}$, let $h_i = h_{m_1P\oplus\varphi^i(m_2P)}$, $H_i = H_{m_1P\oplus\varphi^i(m_2P)}$, $k_i = h_{n_1P\oplus\varphi^i(n_2P)}$, $K_i = K_{n_1P\oplus\varphi^i(n_2P)}$.

Let m_1, m_2, n_1, n_2 be positive integers such that $m_1 + m_2 = n_1 + n_2 = m$. Suppose that we are given $h_{m_1P}, h_{m_2P}, h_{n_1P}, h_{n_2P}$. The subalgorithm computes h_{mP} by applying the following strategy. By Proposition 82, one can use the $(2,3,3)$-generalized summation polynomial computed with Procedure 3, to compute

$$S_1 = S_{m_1P,m_2P} = S_{1,1} + yS_{1,2}$$

from h_{m_1P}, h_{m_2P}, and

$$S_2 = S_{n_1P,n_2P} = S_{2,1} + yS_{2,2}$$

from h_{n_1P}, h_{n_2P}. Up to multiplying by a nonzero constant, $S_1 = \prod_{i=0}^{2} h_i \mod y^2 - f(x,1)$ and $S_2 = \prod_{i=0}^{2} k_i \mod y^2 - f(x,1)$. Hence S_1, S_2 share the factor $h_0 = k_0 = h_{mP}$. By Lemma 83, one has that

$$H_{mP} | G, \text{ where } G = \gcd\left(f(x,1)S_{1,2}^2 - S_{1,1}^2, f(x,1)S_{2,2}^2 - S_{1,2}^2\right).$$

Moreover, if $m_1P\oplus\varphi(m_2P)$ and $m_1P\oplus\varphi^2(m_2P)$ are not Frobenius conjugates of $\pm(n_1P\oplus\varphi(n_2P))$ or $\pm(n_1P\oplus\varphi^2(n_2P))$, that is if $h_1, h_2 \notin \{k_1(x,y), k_2(x,y), -k_1(x,-y), -k_2(x,-y)\}$, then $G = H_{mP}$. In this case, one can compute h_{mP} from G and S_1 (or from G and S_2) by solving the linear system of Proposition 84, provided that the assumptions of the proposition are satisfied.

We now give the subalgorithm and we prove its correctness.

Subalgorithm 1.

Input: The polynomials $h_{m_1P}, h_{m_2P}, h_{n_1P}, h_{n_2P}$, such that $h_1, h_2 \notin \{k_1, k_2\}$
Output : $h_{mP} = y - (\gamma_1 x + \gamma_0)$

1: **if** $h_{m_1P} = h_{m_2P}$ **then return** h_{-m_1P} **endif**
2: **if** $h_{n_1P} = h_{n_2P}$ **then return** h_{-n_1P} **endif**
3: Compute $S_1 = S_{m_1P,m_2P}$ from h_{m_1P}, h_{m_2P} ▷ formulas in Section 1 of the appendix
4: Compute $S_2 = S_{n_1P,n_2P}$ from h_{n_1P}, h_{n_2P}
5: **if** $h_{m_1P}(x, y) = -h_{m_2P}(x, -y)$ **then**
6: $W \leftarrow \text{monic}(S_1)$
7: $L \leftarrow L(W, S_2)$ ▷ see Proposition 84
8: Compute $h = y - (\gamma_1 x + \gamma_0)$ by solving the linear system associated to L
9: **return** h
10: **end if**
11: **if** $h_{n_1P}(x, y) = -h_{n_2P}(x, -y)$ **then**
12: $W \leftarrow \text{monic}(S_2)$
13: $L \leftarrow L(W, S_1)$ ▷ see Proposition 84
14: Compute $h = y - (\gamma_1 x + \gamma_0)$ by solving the linear system associated to L
15: **return** h
16: **end if**
17: $G \leftarrow \gcd(fS_{1,2}^2 - S_{1,1}^2, fS_{2,2}^2 - S_{2,1}^2)$
18: Decompose G in irreducible factors in $\mathbb{F}_q[x]$
19: $W_1, \cdots W_s \leftarrow$ monic distinct irreducible factors of G of degree 3
20: **for** $j \in \{1, \cdots s\}$ **do**
21: $W \leftarrow W_j$
22: **if** $W \neq S_{1,2}$ **then**
23: $L \leftarrow L(W, S_1)$ ▷ see Proposition 84
24: Compute $h = y - (\gamma_1 x + \gamma_0)$ by solving the linear system associated to L
25: **if** $W | (\gamma_1 x + \gamma_0) S_{2,2} + S_{2,1}$ **then** return h **end if**
26: **else** ▷ $W = S_{1,2}$
27: $L \leftarrow L(W, S_2)$ ▷ see Proposition 84
28: Compute $h = y - (\gamma_1 x + \gamma_0)$ by solving the linear system associated to L
29: **return** h
30: **end if**
31: **end for**

Theorem 87. *Subalgorithm* 1 *is correct.*

To prove the theorem we use the following.

Remark 88. Since T_3 has prime order $p > 3$, then $T_3 \cap E(\mathbb{F}_q)[3] = \{\mathcal{O}\}$. Hence H_Q is irreducible over $\mathbb{F}_q$ for every $Q \in T_3 \setminus \{\mathcal{O}\}$. So H_{mP} is irreducible over $\mathbb{F}_q[x]$ for every $0 < m < p$. Moreover, $h_{mP} \neq h_{-mP}$, since, if this were the case, then $mP \oplus \varphi^i(mP) = \mathcal{O}$.

Proof of Theorem 87. If $h_{m_1P} = h_{m_2P}$ as in line 1 of the subalgorithm, then $m_2P = \varphi^i(m_1P)$ for some $i \in \{0, 1, 2\}$. Since we assume that m is odd, then $m_1 \neq m_2$ and $m_1 + m_2 = m < p$, hence $i \neq 0$. Therefore $mP = (m_1 + m_2)P = m_1(P \oplus \varphi^i(P)) = -m_1\varphi^j(P)$ where $\{i, j\} = \{1, 2\}$, and $i \neq j$. It follows that $h_{mP} = h_{-m_1P}$ and line 1 is

correct. The same argument shows that, if $h_{n_1P} = h_{n_2P}$ as in line 2 of the subalgorithm, then $h_{mP} = h_{-n_1P}$, and line 2 is correct. Correctness of lines 3, 4 follows from Proposition 82.

Observe that, up to multiplication by a nonzero constant, $S_1 = h_{mP}h_1h_2$ and $S_2 = h_{mP}k_1k_2 \mod y^2 - f(x,1)$. Moreover, by Lemma 83, $f(x,1)S_{1,2}^2 - S_{1,1}^2 = H_0H_1H_2$ and $f(x,1)S_{2,2}^2 - S_{2,1}^2 = H_0K_1K_2$ up to multiplication by a nonzero constant.

Suppose first that $h_{m_1P} = h_{-m_2P}$ as in line 5. Then $m_2P = -m_1\varphi^i(P)$ for some $i \in \{0,1,2\}$. Since $0 < m < p$, we have that $i \neq 0$. Then $m_2P = -m_1\varphi(P)$ or $m_2P = -m_1\varphi^2(P)$. The case $m_2P = -m_1\varphi(P)$ implies $mP = (m_1+m_2)P = m_1(P \oplus (-\varphi(P)))$, $m_1P \oplus \varphi(m_2P) = m_1(P \oplus (-\varphi^2(P))) = -\varphi^2(mP)$ and $m_1P \oplus \varphi^2(m_2P) = \mathcal{O}$. From these equalities, we have $S_1 = h_0h_1 = h_{mP}h_{-mP} = H_{mP} \mod y^2 - f(x,1)$ (up to multiplication by a nonzero constant). The case $m_2P = -m_1\varphi^2(P)$ implies $mP = (m_1+m_2)P = m_1(P \oplus (-\varphi^2(P)))$, $m_1P \oplus \varphi(m_2P) = \mathcal{O}$ and $m_1P \oplus \varphi^2(m_2P) = m_1(P \oplus (-\varphi(P))) = -\varphi(mP)$. From these equalities, we have $S_1 = h_0h_2 = h_{mP}h_{-mP} = H_{mP} \mod y^2 - f(x,1)$ (up to multiplication by a nonzero constant). Moreover, if $h_{m_1P} = h_{-m_2P}$, we have that $h_{n_1P} \neq h_{-n_2P}$. In fact, if $h_{n_1P} = -h_{n_2P}$, using the same arguments as above, we have that $S_2 = k_0k_1 = h_{mP}h_{-mP} = H_{mP}$ or $S_2 = k_0k_2 = h_{mP}h_{-mP} = H_{mP} \mod y^2 - f(x,1)$ (up to multiplication by a nonzero constant). So $h_{-mP} \in \{k_1, k_2\}$. This is not possible since $h_{-mP} \in \{h_1, h_2\}$ from the previous discussion, and we are supposing $h_1, h_2 \notin \{k_1, k_2\}$. Hence we have that $h_{n_1P} \neq h_{-n_2P}$. The latter inequality implies $S_{2,2} \neq 0$ by Lemma 83. Moreover, by Remark 88, H_{mP} is irreducible over $\mathbb{F}_q[x]$. So, in order to apply Proposition 84 with $W = monic(S_1)$ and S_2, it remains to prove that, if $h_{m_1P} = h_{-m_2P}$, then $H_{mP} \neq S_{2,2}$. Suppose this is not the case. Let $mP = (\overline{x}, \overline{y})$. We have that $H_{mP} = S_{2,2}$ implies $S_{2,2}(\overline{x}^{q^j}) = 0$ for all $j \in \{0,1,2\}$. Moreover, $S_2(\overline{x}^{q^j}, \overline{y}^{q^j}) = 0$ for all $j \in \{0,1,2\}$, by construction of S_2. Hence, for $j \in \{0,1,2\}$, we have $S_2(-\varphi^j(P)) = S_2(\overline{x}^{q^j}, -\overline{y}^{q^j}) = S_{2,1}(\overline{x}^{q^j}) - \overline{y}^{q^j}S_{2,2}(\overline{x}^{q^j}) = S_{2,1}(\overline{x}^{q^j}) = S_{2,1}(\overline{x}^{q^j}) + \overline{y}^{q^j}S_{2,2}(\overline{x}^{q^j}) = S_2(\overline{x}^{q^j}, \overline{y}^{q^j}) = 0$. This implies that $k_i = h_{-mP}$ for some $i \in \{0,1,2\}$. Since $h_{mP} \neq h_{-mP}$ by Remark 88, we have that $\exists\ i \in \{1,2\}$ such that $k_i = h_{-mP} \in \{h_1, h_2\}$. This is not possible because $h_1, h_2 \notin \{k_1, k_2\}$ by hypothesis. Hence all hypothesis of Proposition 84 hold for $W = H_{mP} = monic(S_1)$ and S_2, and correctness of lines $5-10$ follows from the proposition.

The proof of correctness of lines $11-16$ is analogous to that for lines $5-10$.

From now on, we have $h_{m_1P} \neq h_{-m_2P}$ and $h_{n_1P} \neq h_{-n_2P}$, which imply $S_{1,2}, S_{2,2} \neq 0$ by Lemma 83.

Let $1 \leq s \leq 3$. Let $W_1, \cdots, W_s$ be the monic distinct irreducible factors of degree 3 over $\mathbb{F}_q[x]$ of $G = \gcd(fS_{1,2}^2 - S_{1,1}^2, fS_{2,2}^2 - S_{2,1}^2)$, as in lines 17, 18, 19 of the subalgorithm. Hence $H_0 \in \{W_1, \cdots, W_s\}$ by Remark 88. Moreover, for $W \in \{W_1, \cdots, W_s\}$, one has that $W = H_j$ for some $j \in \{0,1,2\}$. Then, if $W \neq S_{1,2}$, one recovers $h = h_j$ from W and S_1 solving the linear system of Proposition 84 (lines 22-24 of the subalgorithm).

Now we focus on line 25. If $h = h_0 = h_{mP}$, one has that $W|(\gamma_1x+\gamma_0)S_{2,2}+S_{2,1}$. Otherwise, $h \neq k_s$ for all $s \in \{0,1,2\}$, as $h_1, h_2 \notin \{k_1, k_2\}$ by hypothesis. So $W \not|(\gamma_1x+\gamma_0)S_{2,2}+S_{2,1}$ by Proposition 84, and line 25 is correct.

Finally, suppose $W = S_{1,2}$ as in line 26. If $W \neq H_0$, one has that there exists $r \in \{1,2\}$ such that $h_j = -(h_r(x,-y))$. Moreover, there exists $s \in \{1,2\}$ such that $h_j = -(k_s(x,-y))$, since $W|G$ and $h_1, h_2 \notin \{k_1, k_2\}$. Then $h_r = k_s$ with $r, s \in \{1,2\}$. This is not possible as $h_1, h_2 \notin \{k_1, k_2\}$. Hence $W = H_0$ and there exists $r \in \{1,2\}$ such that $h_{mP} \neq h_{rP} = h_{-mP}$. This implies that $k_s \neq h_{-mP}$ for all $s \in \{0,1,2\}$, since $h_1, h_2 \notin \{k_1, k_2\}$. So $W \neq S_{2,2}$, and one recovers $h = h_{mP}$ from W and S_2 solving the linear system of Proposition 84. Hence, lines 26-30 are correct. □

We use the subalgorithm at each step of our Montgomery-ladder-style algorithm. We have two different types of input lines. The first type is used in the general case, and the second for exceptional cases.

(a) *Input lines of type* (a): The subalgorithm computes h_{mP} from h_P, $h_{(m-1)P}$, $h_{\frac{m-1}{2}P}$ and $h_{\frac{m+1}{2}P}$. The subalgorithm does not apply to a set M of exceptional values for m.

(b) *Input lines of type* (b): Let $R = \{(-3,-7),(-3,5),(3,-5),(3,7)\}$, $(r_1,r_2) \in R$. The subalgorithm computes h_{mP} from h_{r_iP}, $h_{(m-r_i)P}$ for $i \in \{1,2\}$. The subalgorithm does not apply to a set $M_{(r_1,r_2)}$ of exceptional values for m.

In the next lemma we describe the sets M and $M_{(r_1,r_2)}$. Moreover, we show that $M \cap (\bigcup_{(r_1,r_2)\in R} M_{(r_1,r_2)}) = \emptyset$. Therefore, one can compute h_{mP} using the subalgorithm with input of type (a) if $m \notin M$ and with input of type (b) if $m \in M$.

Lemma 89. *In the setting established above, one has the following:*

1. $h_{P\oplus(m-1)\varphi^i(P)} = h_{\frac{m-1}{2}P\oplus\frac{m+1}{2}\varphi^j(P)}$ *for some* $i, j \in \{1,2\}$ *if and only if* $m \in M$, *where*

$$M = \left\{ \frac{\pm 3}{2s+1}, \frac{s-4}{3s}, \frac{4s-1}{2s+1}, \frac{s+5}{3(s+1)}, \frac{4s+5}{2s+1} \mod p \right\}.$$

   Hence Subalgorithm 1 *correctly computes* h_{mP} *from* h_P, $h_{(m-1)P}$, $h_{\frac{m-1}{2}P}$ *and* $h_{\frac{m+1}{2}P}$ *if* $m \notin M$.

2. *Let* $R = \{(-3,-7),(3,7),(-3,5),(3,-5)\}$, $(r_1,r_2) \in R$. *Then* $h_{r_1P\oplus(m-r_1)\varphi^i(P)} = h_{r_2P\oplus(m-r_2)\varphi^j(P)}$ *for some* $i,j \in \{1,2\}$ *if and only if* $m \in M_{(r_1,r_2)}$, *where*

   - $M_{(3,7)} = \{\frac{17s+4}{2s+1}, \frac{-4s-17}{s-1}, \frac{10s+11}{2s+1}, \frac{10s-1}{2s+1}, \frac{4s-13}{-s-2}, \frac{17s+13}{2s+1} \mod p\}$,
   - $M_{(-3,-7)} = \{-m \mod p : m \in M_{(3,7)}\}$,
   - $M_{(-3,5)} = \{\frac{7s+8}{2s+1}, \frac{-8s-7}{s-1}, \frac{2s+13}{2s+1}, \frac{2s-11}{2s+1}, \frac{8s+1}{-s-2}, \frac{7s-1}{2s+1} \mod p\}$,
   - $M_{(3,-5)} = \{-m \mod p : m \in M_{(-3,5)}\}$.

   Fix $(r_1,r_2) \in R$. *Subalgorithm* 1 *correctly computes* h_{mP} *from* $h_{r_1P}, h_{r_2P}, h_{(m-r_1)P}, h_{(m-r_2)P}$ *if* $m \notin M_{(r_1,r_2)}$.

3. *One has that* $M \cap (\bigcup_{(r_1,r_2)\in R} M_{(r_1,r_2)}) = \emptyset$. *Hence, if Subalgorithm* 1 *cannot compute* h_{mP} *with input of type (a), it can compute it with input of type (b).*

Proof. By Theorem 87, and following Notation 86, we have that Subalgorithm 1 correctly computes h_{mP} from the input lines $h_{m_1P} = h_P$, $h_{m_2P} = h_{(m-1)P}$, $h_{n_1P} = h_{\frac{m-1}{2}P}$ and $h_{n_2P} = h_{\frac{m+1}{2}P}$ if $h_1, h_2 \notin \{k_1, k_2\}$, that is, if $h_{P\oplus(m-1)\varphi^i(P)} \neq h_{\frac{m-1}{2}P\oplus\frac{m+1}{2}\varphi^j(P)}$ for all $i,j \in \{1,2\}$. We have that

$$h_{P\oplus(m-1)\varphi^i(P)} = h_{\frac{m-1}{2}P\oplus\frac{m+1}{2}\varphi^j(P)} \text{ for some } i,j \in \{1,2\}$$

if and only if

$$P \oplus (m-1)\varphi^i(P) = \varphi^h(\frac{m-1}{2}P \oplus \frac{m+1}{2}\varphi^j(P)) \text{ for some } i,j \in \{1,2\}, h \in \{0,1,2\}.$$

Since $\varphi(P) = sP$ and P is of period p, the latter equality is equivalent to saying

$$1 + (m-1)s^i = s^h(\frac{m-1}{2} + \frac{m+1}{2}s^j) \mod p \text{ for some } i,j \in \{1,2\}, h \in \{0,1,2\}. \quad (4.9)$$

Moreover, $P \in T_3$, so $P \oplus \varphi(P) \oplus \varphi^2(P) = \mathcal{O}$. This implies the equality

$$1 + s + s^2 = 0 \mod p, \tag{4.10}$$

since $\varphi(P) = sP$ and P has period p. By applying the equality (4.10) to the equality (4.9) for all $i, j \in \{1, 2\}$ and all $h \in \{0, 1, 2\}$, one directly computes that (4.9) is equivalent to saying that $m \in M$. Notice that all denominators in M are nonzero modulo p, since equality (4.10) holds and $p \neq 2, 3$. We have then proved point 1 of the lemma.
The proof for part 2 is analogous to that of part 1.
We now prove part 3. Suppose that $M \cap (\bigcup_{(r_1,r_2)\in R} M_{(r_1,r_2)}) \neq \emptyset$. One can check by direct computation that this implies $as = b \mod p$ or $as = -b \mod p$ for some a and b such that $0 < a, b \leq 60$ and $a \neq b$. If $as = b \mod p$, then from (4.10) one obtains that $a^2 + ab + b^2 = 0 \mod p$. This is not possible since $0 < a^2 + ab + b^2 \ll p$. The case $as = -b \mod p$ can be treated similarly. □

Remark 90. Lemma 89 is no longer true for small values of p. Consider e.g. the elliptic curve $y^2z = x^3 + 5xz^2 + 4z^3$ over $\mathbb{F}_7$, with $p = 31$ and $s = 25$. We have $M \cap M_{(-3,-7)} = \{7, 11, 13\} \cap \{13, 15\} = \{13\} \neq \emptyset$.

Example 91. Let $q = 1021$ and $\mathbb{F}_{q^3} = \mathbb{F}_q[\zeta]/(\zeta^3 - 5)$. Let E and P as in Example 85, i.e., let E be the elliptic curve over $\mathbb{F}_q$ of equation $y^2z = x^3 + 230xz^2 + 191z^3$, and let $P = (782\zeta^2 + 802\zeta + 45, 979\zeta^2 + 299\zeta + 133)$. Then $p = 1021381$, $s = 161217$, $M = \{161219, 322435, 322437, 465965\}$.

We show how to compute h_{5P} using Subalgorithm 1 with input of type (a). In Example 85 we computed h_{2P} and h_{3P}. Using the formulas of Section 1 and Section 2 of the appendix, we compute $h_{4P} = y - (698x + 155)$ from h_{2P}, $S_1 = (524x^4 + 131x^3 + 826x^2 + 631x + 160) + y(x^3 + 243x^2 + 651x + 776)$ from h_P and h_{4P}, $S_2 = (331x^4 + 653x^3 + 169x^2 + 259x + 536) + y(x^3 + 570x^2 + 680x + 578)$ from h_{2P} and h_{3P}. Then we compute $G = \gcd(fS_{1,2}^2 - S_{1,1}^2, fS_{2,2}^2 - S_{2,1}^2) = x^3 + 455x^2 + 81x + 68$. Hence $G = H_{5P}$, and $H_{5P} \neq S_{1,2}$. So we obtain $h_{5P} = y - (736x + 804)$ from G and S_1 as in line 24 of Subalgorithm 1.

Similarly one can compute $h_{7P} = y - (112x + 43)$ from h_P, h_{6P}, h_{3P}, h_{4P}.

The next two examples illustrate some special cases of Subalgorithm 1.

Example 92. Let E and P be as in the previous example and let $m = 337887$. One can check that

$$P \oplus (m-1)\varphi^2(P) = -\frac{m-1}{2}\varphi^2(P) \oplus (-\frac{m+1}{2}\varphi(P)).$$

If we try to compute h_{mP} using Subalgorithm 1 with input of type (a), we first compute $G = x^6 + 778x^5 + 86x^4 + 778x^3 + 599x^2 + 494x + 658$, which splits over $\mathbb{F}_q$ into two irreducible factors of degree 3, namely $W_1 = x^3 + 11x^2 + 843x + 540$ and $W_2 = x^3 + 767x^2 + 1016x + 5$. From W_1 we recover $h_1 = y - (166x + 727) = 0$ which is the line through $P + (m-1)\varphi^2(P)$. From W_2 we recover $h_2 = y - (423x + 57) = 0$ which is the line through mP. By checking the condition at line 25 of the subalgorithm, we are able to decide that $h_{mP} = h_2$.

Example 93. Let $q = 1021$ and $\mathbb{F}_{q^3} = \mathbb{F}_q[\zeta]/(\zeta^3 - 5)$. Let E be the elliptic curve of equation $y^2z = x^3 + 71xz^2 + 529z^3$ defined over $\mathbb{F}_q$. Then T_3 is generated by $P = (853\zeta^2 + 995\zeta + 244, 178\zeta^2 + 927\zeta + 959)$, which has prime order $p = 1009741$. Moreover $s = 325960$ and $M_{(3,-5)} = \{32671, 391027\}$. Let $m = 65339$. One can check that

$$mP = -3P \oplus (-(m-3)\varphi^2(P)).$$

We compute h_{mP} using Subalgorithm 1 with input of type (b), with $(r_1, r_2) = (3, -5)$ and obtain $G = S_{1,2}$. Then we can compute $h_{mP} = y - (566x + 37)$ from G and S_2.

4.2.2 A first algorithm for scalar multiplication

We give our Montgomery-ladder style algorithm for scalar multiplication in T_3, in its basic form.

Notation 94. Let m be an integer with $0 < m < p$. Let $m = \sum_{i=0}^{\ell-1} m_i 2^i$ be the binary representation of m, with $m_i \in \{0, 1\}$ for all i, $\ell = \lceil \log_2 m \rceil$ and $m_{\ell-1} = 1$. Let

$$k_i = \sum_{j=i}^{\ell-1} m_j 2^{j-i}$$

for $i \in \{0, \cdots, \ell - 1\}$. Notice that $k_0 = m$. Let

$$M = \left\{ \frac{\pm 3}{2s+1}, \frac{s-4}{3s}, \frac{4s-1}{2s+1}, \frac{s+5}{3(s+1)}, \frac{4s+5}{2s+1} \mod p \right\},$$

and let $\mathcal{M} = M \cap (2\mathbb{Z} + 1)$.

General strategy of the algorithm. Our algorithm takes h_P and m as input, and it returns h_{mP} as output. It adopts the classical double-and-add strategy for scalar multiplication. For each step i, from $i = \ell - 1$ down to $i = 0$, it computes

$$u_i = h_{k_i P} \text{ and } v_i = h_{(k_i+1)P}.$$

At the end of the cycle, it outputs $u_0 = h_{mP}$. In order to compute the polynomials u_i and v_i of each step, the algorithm uses the doubling formulas of Section 2.1 of the appendix and Subalgorithm 1 with input the polynomials that it has computed in the previous steps.

The proposition below gives recursive definitions for u_i and v_i. Our algorithm applies this proposition to compute the polynomials u_i and v_i at each step i.

Notation 95. Write $\text{Subalg}(h_1, h_2, h_3, h_4)$, for the output of Subalgorithm 1 with input h_1, h_2, h_3, h_4. For any $h = h_Q$ with $Q \in T_3$, let $D(h) = h_{2Q}$, where h_{2Q} is computed from the coefficients of h by means of the doubling formulas of the appendix. Then $D^k(h) = h_{2^k Q}$, where $h_{2^k Q}$ is computed from h via iterated application of the doubling formulas of the appendix.

Proposition 96. *For i from $i = \ell - 1$ down to $i = 0$, recursively define u_i and v_i as follows.*

- $u_{\ell-1} = h_P$, $v_{\ell-1} = h_{2P}$.
- $u_{\ell-2} = h_{2P}$ *and* $v_{\ell-2} = h_{3P}$ *if* $m_{\ell-2} = 0$,
  $u_{\ell-2} = h_{3P}$ *and* $v_{\ell-2} = h_{4P}$ *if* $m_{\ell-2} = 1$.
- *For* $0 \leq i \leq \ell - 3$*:*
  - *(General case) if* k_i *and* $k_i + 1 \notin \mathcal{M}$*, let*
    $u_i = D(u_{i+1})$ *and* $v_i = \text{Subalg}(h_P, D(u_{i+1}), u_{i+1}, v_{i+1})$ *if* $m_i = 0$
    $u_i = \text{Subalg}(h_P, D(u_{i+1}), u_{i+1}, v_{i+1})$ *and* $v_i = D(v_{i+1})$ *if* $m_i = 1$.

– *(Special cases) if k_i or $k_i+1 \in \mathcal{M}$*

* *If $m_i = 0$, let*

$$u_i = D(u_{i+1}),$$
$$v_i = \begin{cases} \text{Subalg}(h_{3P}, D^2(u_{i+2}), h_{7P}, D^3(u_{i+3})) & \text{if} \quad m_{i+1} = m_{i+2} = 1 \\ \text{Subalg}(h_{3P}, D^3(u_{i+3}), h_{-5P}, D^3(v_{i+3})) & \text{if} \quad m_{i+1} = 1, m_{i+2} = 0 \\ \text{Subalg}(h_{-3P}, D^3(v_{i+3}), h_{5P}, D^3(u_{i+3})) & \text{if} \quad m_{i+1} = 0, m_{i+2} = 1 \\ \text{Subalg}(h_{-3P}, D^2(v_{i+2}), h_{-7P}, D^3(v_{i+3})) & \text{if} \quad m_{i+1} = m_{i+2} = 0 \end{cases}.$$

* *If $m_i = 1$, let*

$$v_i = D(v_{i+1}),$$
$$u_i = \begin{cases} \text{Subalg}(h_{3P}, D^2(u_{i+2}), h_{7P}, D^3(u_{i+3})) & \text{if} \quad m_{i+1} = m_{i+2} = 1 \\ \text{Subalg}(h_{3P}, D^3(u_{i+3}), h_{-5P}, D^3(v_{i+3})) & \text{if} \quad m_{i+1} = 1, m_{i+2} = 0 \\ \text{Subalg}(h_{-3P}, D^3(v_{i+3}), h_{5P}, D^3(u_{i+3})) & \text{if} \quad m_{i+1} = 0, m_{i+2} = 1 \\ \text{Subalg}(h_{-3P}, D^2(v_{i+2}), h_{-7P}, D^3(v_{i+3})) & \text{if} \quad m_{i+1} = m_{i+2} = 0 \end{cases}.$$

Then $u_i = h_{k_iP}$ and $v_i = h_{(k_i+1)P}$, for all $i \in \{0, \cdots, \ell-1\}$.

Proof. We proceed by induction on i. The thesis is easily verified for $i = \ell-1$ and $i = \ell-2$. Hence let $0 \leq i \leq \ell-3$ and assume that the thesis holds for $j \in \{i+1, \cdots, \ell-1\}$. Suppose first that $k_i, k_i+1 \notin \mathcal{M}$ and that $m_i = 0$ (the proof for the case $m_i = 1$ is analogous). Then $k_i = 2(k_{i+1})$ and $u_i = D(u_{i+1}) = h_{2k_{i+1}P} = h_{k_iP}$ by induction. Moreover, by induction we get

$$\text{Subalg}(h_P, D(u_{i+1}), u_{i+1}, v_{i+1}) = \text{Subalg}(h_P, h_{2k_{i+1}P}, h_{k_{i+1}P}, h_{(k_{i+1}+1)P}) =$$
$$\text{Subalg}\left(h_P, h_{k_iP}, h_{\frac{k_i}{2}P}, h_{\left(\frac{k_i}{2}+1\right)P}\right).$$

Since $k_i + 1 \notin \mathcal{M}$, Subalgorithm 1 with input of type (a) correctly outputs $v_i = h_{(k_i+1)P}$. Now suppose that k_i or $k_i + 1 \in \mathcal{M}$ and assume that $m_i = 0$, $m_{i+1} = m_{i+2} = 1$ (the proof for the other cases is analogous). If k_i or $k_i + 1 \in \mathcal{M}$, then $i < \ell - 3$, since $5, 7 \notin \mathcal{M}$. Hence we already have computed the polynomials of the three previous steps $i + 1$, $i + 2$, $i + 3$. Since $m_i = 0$, we prove the thesis for u_i as in the general case. On the other hand, $k_i + 1 \in \mathcal{M}$ so we cannot define v_i using Subalgorithm 1 with input of type (a), as we did before. However $k_i + 1 = 3 + 4k_{i+2} = 7 + 8k_{i+3}$, so by induction we get

$$\text{Subalg}(h_{3P}, D^2(u_{i+2}), h_{7P}, D^3(u_{i+3})) = \text{Subalg}(h_{3P}, h_{4(k_{i+2})P}, h_{7P}, h_{8(k_{i+3})P}).$$

Moreover, since $k_i + 1 \in \mathcal{M}$, then $k_i + 1 \notin M_{(3,7)}$ by Lemma 89, hence Subalgorithm 1 with input of type (b) correctly outputs $v_i = h_{(k_i+1)P}$. □

Remark 97. If $k_i, k_i + 1 \notin \mathcal{M}$, at step i one needs only the polynomials computed in the previous step in order to compute the polynomials u_i, v_i. If k_i or $k_i+1 \in \mathcal{M}$ one needs the polynomials computed in the steps $i + 2$ and $i + 3$ in order to compute them. Therefore:

- In our algorithm, the last three pairs of polynomials that have been computed are stored in a vector L, which is updated at each step of the cycle.
- The algorithm looks the i's for which k_i or $k_i + 1 \in \mathcal{M}$ at the start. Namely, for each $i \in \{0, \cdots, \ell - 2\}$, it computes k_i and $k_i + 1$, and it adds i to the list S if k_i or $k_i + 1 \in \mathcal{M}$. Hence, at each step i, we know whether we have to call Subalgorithm 1 with input of type (a) or of type (b), by simply checking if $i \in S$.

Algorithm 9 (Scalar multiplication in T_3, basic form)**.**

Input : h_P, m an integer modulo p
Output : h_{mP}

1 : $m \leftarrow \sum_{i=0}^{\ell-1} m_i 2^i$ binary expansion of m
▷ collection of the exceptional steps
2 : $S \leftarrow \{i \in \{0, \cdots, \ell-2\} : k_i \leftarrow \sum_{j=i}^{\ell-1} m_j 2^{j-i} \in \mathcal{M} \text{ or } k_i + 1 \in \mathcal{M}\}$
▷ step $i = \ell - 1$
3 : $u \leftarrow h_P$, $v \leftarrow h_{2P}$, $L \leftarrow [(u, v)]$ ▷ $L = [(u_{\ell-1}, v_{\ell-1})]$
4 : **if** $\ell - 1 = 0$ **then return** u **end if**
▷ step $i = \ell - 2$
5 : **if** $m_{\ell-2} = 0$ **then** $u \leftarrow h_{2P}$, $v \leftarrow h_{3P}$ **else** $u \leftarrow h_{3P}$, $v \leftarrow h_{4P}$ **end if**
6 : Append (u, v) to L ▷ $L = [(u_{\ell-1}, v_{\ell-1}), (u_{\ell-2}, v_{\ell-2})]$
7 : **if** $\ell - 2 = 0$ **then return** u **end if**
▷ cycle for: steps from $i = \ell - 3$ to $i = 0$
8: **for** i from $\ell - 3$ down to 0 **do**
▷ special cases
9: **if** $i \in S$ **then**
10: **if** $m_{i+1} = 1$ **then**
11: **if** $m_{i+2} = 1$ **then**
12: $h_{exc} \leftarrow \text{Subalg}(h_{3P}, D^2(L[2][1]), h_{7P}, D^3(L[1][1]))$
13: **else** ▷ $m_{i+1} = 1$, $m_{i+2} = 0$
14: $h_{exc} \leftarrow \text{Subalg}(h_{3P}, D^3(L[1][1]), h_{-5P}, D^3(L[1][2]))$
15: **end if**
16: **else** ▷ $m_{i+1} = 0$
17: **if** $m_{i+2} = 1$ **then**
18: $h_{exc} \leftarrow \text{Subalg}(h_{-3P}, D^3(L[1][2]), h_{5P}, D^3(L[1][1]))$
19: **else** ▷ $m_{i+1} = 0$, $m_{i+2} = 0$
20: $h_{exc} \leftarrow \text{Subalg}(h_{-3P}, D^2(L[2][2]), h_{-7P}, D^3(L[1][2]))$
21: **end if**
22: **end if**
23: **if** $|L| = 3$ **then** remove $L[1]$ from L **end if** ▷ $L = [(u_{i+2}, v_{i+2}), (u_{i+1}, v_{i+1})]$
▷ computation of u, v at step i
24: **if** $m_i = 0$ **then**
25: $u \leftarrow D(L[2][1])$
26: **if** $i \in S$ **then**
27: $v \leftarrow h_{exc}$
28: **else**
29: $v \leftarrow \text{Subalg}(h_P, D(L[2][1]), L[2][1], L[2][2])$
30: **else** ▷ $m_i = 1$
31: **if** $i \in S$ **then**
32: $u \leftarrow h_{exc}$
33: **else**
34: $u \leftarrow \text{Subalg}(h_P, D(L[2][1]), L[2][1], L[2][2])$
35: **end if**
36: $v \leftarrow D(L[2][2])$
37: **end if**

38: Append (u, v) to L ▷ $L = [(u_{i+2}, v_{i+2}), (u_{i+1}, v_{i+1}), (u_i, v_i)]$
39: **end for**
40: **return** $L[3][1]$

Theorem 98. *Algorithm* 9 *is correct.*

Proof. Correctness of lines $3-7$ is easy to check. Notice that, at the beginning of the cycle at line 8, the list L is $L = [(u_{\ell-1}, v_{\ell-1}), (u_{\ell-2}, v_{\ell-2})]$. Moreover, one has that $\ell - 3 \notin S$, since $5, 7 \notin \mathcal{M}$. So we do not check whether $\ell - 3 \in \mathcal{S}$. Observe now that for each i from $i = \ell - 3$ down to $i = 0$, the list L at line 23 is $L = [(u_{i+2}, v_{i+2}), (u_{i+1}, v_{i+1})]$, while at line 38 the list is $L = [(u_{i+2}, v_{i+2}), (u_{i+1}, v_{i+1}), (u_i, v_i)]$. Hence correctness follows from Proposition 96. □

We now give an example of computation of a multiplication by m for which the algorithm runs into the special cases.

Example 99. Let $q = 1021$ and $\mathbb{F}_{q^3} = \mathbb{F}_q[\zeta]/(\zeta^3 - 5)$. Let E and P be as in Example 85 and Example 91, i.e., let E be the elliptic curve over $\mathbb{F}_q$ of equation $y^2z = x^3 + 230xz^2 + 191z^3$ and let $P = (782\zeta^2 + 802\zeta + 45, 979\zeta^2 + 299\zeta + 133)$. Let $m = 644875$, with binary representation

$$m = 2^{19} + 2^{16} + 2^{15} + 2^{14} + 2^{12} + 2^{10} + 2^9 + 2^8 + 2^3 + 2 + 1.$$

For i from 19 to 0 the pairs $(k_i, k_i + 1)$ are

$$(1,2), (2,3), (4,5), (9,10), (19,20), (39,40), (78,79), (157,158), (314,315), (629,630),$$

$$(1259,1260), (2519,2520), (5038,5039), (10076,10077), (20152,20153), (40304,40305),$$

$$(80609,80610), (161218,161219), (322437,322438), (644875,644876).$$

Hence the set of the special cases is $S = \{2, 1\}$ since $k_2 + 1 = 161219$, $k_1 = 322437 \in \mathcal{M}$. We compute $h_{mP} = y - (105x + 587)$ using Algorithm 1. At step $i = 2$ we compute $v = h_{exc}$ with $m_3 = 1$ and $m_4 = 0$ (line 14 of the algorithm). At step $i = 1$ we compute $u = h_{exc}$ with $m_2 = 0$ and $m_3 = 1$ (line 18 of the algorithm).

4.2.3 The optimized algorithm for scalar multiplication

In this subsection, we optimize the Montgomery-ladder style algorithm given in the previous subsection and give the conclusive algorithm to perform scalar multiplication in T_3 in optimal coordinates.

Remark 100. Let m be an integer modulo p. If $m > \frac{p-1}{2}$, one can reduce the computation of multiplication by m to the computation of multiplication by $m' = -m \mod p$, with $m' \leq \frac{p-1}{2}$. One does so by using the equality $h_{-P}(x, y) = -h_P(x, -y)$.

Frobenius reduction. In Section 1.4.2, we explained how the Frobenius endomorphism of the elliptic curve E can be used to speed up non-compressed scalar multiplication in T_n. We called such strategy Frobenius reduction. We now adapt this method and apply Frobenius reduction to our scalar multiplication algorithm, in order to increase the efficiency of the computation process.

Let m be an integer modulo p. One can write $m = m_0 + sm_1$, with $m_0, m_1 \in O(q) = O(\sqrt{p})$ (see [6, Section 15.3.2]). In order to compute h_{mP} given m and h_P, we call Algorithm 9 three times with input m_0, m_1 and $m_0 + m_1$ respectively, instead of calling

Algorithm 9 once with input m. Notice that $m_0, m_1, m_0 + m_1 \in O(\sqrt{p})$, while $m \in O(p)$. Hence one reduces the computation of the multiplication by m to the computation of at most three multiplications by integers of smaller size. Similarly to what we did in Algorithm 9, one needs to pay attention to the special cases where one cannot apply Subalgorithm 1.

Lemma 101. *Let m, m_0, m_1 be integers modulo p, with $m_0, m_1 \neq 0$. One has the following facts.*

1. *Subalgorithm 1 with input h_P, h_{mP}, $h_{(m+1)P}$, $h_{(s-1)P}$ correctly outputs $h_{(m+s)P}$ if $m \notin \mathcal{A}_1$, where*

$$\mathcal{A}_1 = \left\{ -2, s, \frac{-3(1+s)}{2+s}, \frac{-3}{2+s}, \frac{s+2}{s-1}, \frac{-3}{2s+1} \mod p \right\}.$$

2. *Subalgorithm 1 with input h_{mP}, h_{-mP}, $h_{(m+s)P}$, $h_{-(m+1)P}$ correctly outputs $h_{m(1-s)P}$ if $m \notin \mathcal{A}_2$, where*

$$\mathcal{A}_2 = \left\{ 1, s, \frac{s+2}{s-1}, \frac{2s+1}{-3}, \frac{1-s}{3s} \mod p \right\}.$$

3. *Subalgorithm 1 with input h_{m_0P}, h_{m_1P}, $h_{(m_0+m_1)P}$, $h_{m_0(1-s)P}$ correctly outputs $h_{(m_0+sm_1)P}$ if $2m_0 + m_1 \neq 0 \mod p$ and $s \notin \mathcal{B}_1$, where*

$$\mathcal{B}_1 = \Big\{ \left(\tfrac{3m_0+m_1}{m_1}\right)^{\pm 1}, \left(\tfrac{m_1-m_0}{2m_0+m_1}\right)^{\pm 1}, \tfrac{m_0+2m_1}{-(2m_0+m_1)}, \tfrac{3m_0+2m_1}{-(3m_0+m_1)}, \tfrac{2m_1}{-(3m_0+m_1)} \mod p,$$

*with $3m_0 + m_1, 2m_0 + m_1, m_1 - m_0 \neq 0 \mod p$}.

4. *Subalgorithm 1 with input h_{m_0P}, h_{m_1P}, $h_{(m_0+m_1)P}$, $h_{m_1(s-1)P}$ correctly outputs $h_{(m_0+sm_1)P}$ if $m_0 + 2m_1 \neq 0 \mod p$ and $s \notin \mathcal{B}_2$, where*

$$\mathcal{B}_2 = \Big\{ \left(\tfrac{m_0+3m_1}{-(2m_0+3m_1)}\right)^{\pm 1}, \left(\tfrac{m_0-m_1}{m_0+2m_1}\right)^{\pm 1}, \tfrac{2m_0+3m_1}{-m_0}, \tfrac{m_0+3m_1}{-2m_0} \mod p,$$

*with $m_0 + 3m_1, 2m_0 + 3m_1, m_0 + 2m_1, m_1 - m_0 \neq 0 \mod p$}.

5. *We take the set*

$$\text{Poly} = \{t+1, t-1, t+2, t+3, 3t+1, t^2+1, t^2+t+1, t^2+4t+2, 2t^2+t+1,$$

$$t^2-t-1, 2t^2+4t+1, t^2+4t+1, t^2+2t+2, t^2+3t+1, t^2+t-1, 2t^2+2t+1,$$

$$t^2+3t+1, t^2-2t-1, t^2+2t-1, 2t^2+3t-1, 2t^2+3t+1\} \subseteq \mathbb{F}_p[t]$$

and the corresponding set of roots in $\mathbb{F}_p$

$$\mathcal{R} = \{\alpha \in \mathbb{F}_p \mid f(\alpha) = 0 \text{ for some } f \in \text{Poly}\}.$$

Then $s \in \mathcal{B}_1 \cap \mathcal{B}_2$ if and only if $m_0 = \alpha m_1$ for some $\alpha \in \mathcal{R}$.

Proof. Recall that Subalgorithm 1 requires the condition $h_1, h_2 \notin \{k_1, k_2\}$ for the input lines, where we follow Notation 86. The lemma then follows from Theorem 87 by direct computation. □

Precomputation. In order to apply Frobenius reduction to scalar multiplication, we need to be able to deal with the exceptional cases of Lemma 101. We chose to solve this problem by using Algorithm 9 to precompute the polynomials of the set

$$\mathcal{L} = \{h_{m(1-s)P} : m \in \mathcal{A}_1 \cup \mathcal{A}_2\} \cup \{h_{(s+\alpha)P} : \alpha \in \mathcal{R}\}. \tag{4.11}$$

In order to compute the polynomials $h_{m(1-s)P}$, we first compute $h_{(s-1)P} \in \mathcal{L}$. Then, we call Algorithm 9 with input $h_{(1-s)P}$ and m.

We are now ready to give our final algorithm for scalar multiplication in T_3.

Notation 102. Recall that at the end of the cycle for in Algorithm 9, one has the pair $L[3] = (h_{mP}, h_{(m+1)P})$. Write $Al_9(h_P, m)$ for the pair $(h_{mP}, h_{(m+1)P})$, computed with a modified version of Algorithm 9 that outputs the whole pair $L[3]$.

Algorithm 10 (Scalar multiplication in T_3, optimized form)**.**

Input : h_P, m an integer modulo p
Output : h_{mP}

$\mathcal{L} \leftarrow$ set (4.11) of precomputed polynomials
if $m > \frac{p-1}{2}$ **then** $\overline{m} \leftarrow -m \mod p$ **else** $\overline{m} \leftarrow m$ **end if**
$\overline{m} \leftarrow m_0 + sm_1$
if $m_0 = 0$ **then** $h \leftarrow Al_9(h_P, m_1)[1]$
else if $m_1 = 0$ **then** $h \leftarrow Al_9(h_P, m_0)[1]$
else ▷ $m_0, m_1 \neq 0$
  if $s \in \mathcal{B}_1 \cap \mathcal{B}_2$ **then** ▷ $\overline{m} = m_1(s+\alpha)$ for some $\alpha \in \mathcal{R}$
    $h \leftarrow Al_9(h_{(s+\alpha)P}, m_1)[1]$ ▷ $h_{(s+\alpha)P} \in \mathcal{L}$
  else ▷ $s \notin \mathcal{B}_1 \cap \mathcal{B}_2$
    $h_{m_0P} \leftarrow Al_9(h_P, m_0)[1]$
    $h_{m_1P} \leftarrow Al_9(h_P, m_1)[1]$
    $h_{(m_0+m_1)P} \leftarrow Al_9(h_P, m_0 + m_1)[1]$
    if $s \notin \mathcal{B}_1$ and $2m_0 + m_1 \neq 0 \mod p$ **then** ▷ Compute $h_{(m_0+sm_1)P}$ from $h_{m_0P}, h_{m_1P}, h_{(m_0+m_1)P}, h_{m_0(1-s)P}$
      if $m_0 \notin \mathcal{A}_1 \cup \mathcal{A}_2$ **then**
        $h_{(m_0+1)P} \leftarrow Al_9(h_P, m_0)[2]$
        $h_{(m_0+s)P} \leftarrow \text{Subalg}(h_P, h_{m_0P}, h_{(m_0+1)P}, h_{(s-1)P})$
        $h_{m_0(1-s)P} \leftarrow \text{Subalg}(h_{m_0P}, h_{-m_0P}, h_{-(m_0+1)P}, h_{(m_0+s)P})$
      end if
      $h \leftarrow \text{Subalg}(h_{m_0P}, h_{m_1P}, h_{(m_0+m_1)P}, h_{m_0(1-s)P})$
    else ▷ $s \notin \mathcal{B}_2$ and $m_0+2m_1 \neq 0 \mod p$: Compute $h_{(m_0+sm_1)P}$ from $h_{m_0P}, h_{m_1P}, h_{(m_0+m_1)P}, h_{m_1(s-1)P}$
      if $m_1 \notin \mathcal{A}_1 \cup \mathcal{A}_2$ **then**
        $h_{(m_1+1)P} \leftarrow Al_9(h_P, m_1)[2]$
        $h_{(m_1+s)P} \leftarrow \text{Subalg}(h_P, h_{m_1P}, h_{(m_1+1)P}, h_{(s-1)P})$
        $h_{m_1(1-s)P} \leftarrow \text{Subalg}(h_{m_1P}, h_{-m_1P}, h_{-(m_1+1)P}, h_{(m_1+s)P})$
      end if
      $h \leftarrow \text{Subalg}(h_{m_0P}, h_{m_1P}, h_{(m_0+m_1)P}, h_{m_1(s-1)P})$
    end if
  if $m > \frac{p-1}{2}$ **then return** $-h(x, -y)$ **else return** h **end if**

Theorem 103. *Algorithm* 10 *is correct.*

Proof. Let $\overline{m}$ as in line 3 of the algorithm.
If $m_0 = 0$ as in line 4, or $m_1 = 0$ as in line 5, then $h = h_{\overline{m}P}$ by Theorem 98.
Suppose now $m_0, m_1 \neq 0$, as in line 6.
If $s \in \mathcal{B}_1 \cap \mathcal{B}_2$ as in line 7, then by Lemma 101, 5, we have that $\overline{m} = m_1(s + \alpha)$ for some $\alpha \in \mathcal{R}$. Furthermore, $h_{(s+\alpha)P} \in \mathcal{L}$, where $\mathcal{L}$ is the set of precomputed polynomials of line 1, defined in (4.11). Hence, by Theorem 98, one can compute $h = h_{\overline{m}P}$ as in line 8 of the algorithm.
Now suppose that $s \notin \mathcal{B}_1 \cap \mathcal{B}_2$, as in line 9 of the algorithm.
Correctness of lines 10, 11 and 12 follows from Theorem 98.
Suppose that $s \notin \mathcal{B}_1$ and $2m_0 + m_1 \neq 0 \mod p$, as in line 13 of the algorithm. Then, by Lemma 101, 3, one can compute $h_{\overline{m}P} = h_{(m_0+sm_1)P}$ using Subalgorithm 1 with input lines h_{m_0P}, h_{m_1P}, $h_{(m_0+m_1)P}$ and $h_{m_0(1-s)P}$. We have already computed h_{m_0P}, h_{m_1P}, $h_{(m_0+m_1)P}$ in lines $10-12$. Hence, we have now to compute $h_{m_0(1-s)P}$, in order to be able to compute $h_{\overline{m}P}$ with Subalgorithm 1, in the case of Lemma 101, 3.
If $m_0 \notin \mathcal{A}_1 \cup \mathcal{A}_2$ as in line 14, then one correctly computes $h_{m_0(1-s)P}$ as in lines $15 - 17$ of the Algorithm, by Theorem 87, Theorem 98 and Lemma 101, 1, 2.
If $m_0 \in \mathcal{A}_1 \cup \mathcal{A}_2$, then by Lemma 101 we cannot compute the polynomial $h_{m_0(1-s)P}$ as we do in lines $15-17$ of the algorithm. Nevertheless, in this case, $h_{m_0(1-s)P}$ belongs to the set $\mathcal{L}$ of precomputed polynomials, by definition of $\mathcal{L}$. Therefore, in both cases $m_0 \notin \mathcal{A}_1 \cup \mathcal{A}_2$ and $m_0 \in \mathcal{A}_1 \cup \mathcal{A}_2$, Subalgorithm 1 in line 19 correctly computes $h = h_{\overline{m}P}$, by Theorem 87.
Now focus on lines $20 - 26$ of the algorithm. We are now in the case in which $s \in \mathcal{B}_1$ or $2m_0 + m_1 = 0 \mod p$. Suppose first that $s \in \mathcal{B}_1$. Then $s \notin \mathcal{B}_2$, since $s \notin \mathcal{B}_1 \cap \mathcal{B}_2$ (line 9 of the algorithm). Moreover, if $s \in \mathcal{B}_1$, then $m_0 + 2m_1 \neq 0 \mod p$. In fact, one can check by direct computation that $s \in \mathcal{B}_1$ and $m_0 + 2m_1 = 0 \mod p$ implies $s \in \{0, -5^{\pm 1}, -3, -1, -4/5, 2/5 \mod p\}$, since $m_0, m_1 \neq 0 \mod p$ (line 6 of the algorithm). This contradicts the equality $s^2 + s + 1 = 0 \mod p$. Now suppose that $2m_0 + m_1 = 0 \mod p$. Using the same arguments as above, one has that $2m_0 + m_1 = 0 \mod p$ implies $s \notin \mathcal{B}_2$ and $m_0 + 2m_1 \neq 0 \mod p$. Hence, in both cases $s \in \mathcal{B}_1$ or $2m_0 + m_1 = 0 \mod p$, of line 20, we have that $s \notin \mathcal{B}_2$ and $m_0 + 2m_1 \neq 0 \mod p$, and one can compute $h_{(m_0+sm_1)P}$ as in line 26 by Lemma 101, 4.
With the same arguments as above, one shows that lines $21 - 25$ of the algorithm are correct, so Subalgorithm 1 at line 26 correctly computes $h = h_{\overline{m}P}$.
From line 2, we have that $h = h_{\overline{m}P} = h_{-mP}$ if $m > \frac{p-1}{2}$, and $h = h_{\overline{m}P} = h_{mP}$ otherwise. Hence, by Remark 100, the algorithm correctly outputs h_{mP} in line 28. □

Remark 104. The aim of Algorithm 10 is to show how to apply Frobenius reduction in order to speed up our scalar multiplication algorithm. Nevertheless, further optimizations are possible. For example, one can introduce variations of Subalgorithm 1 in order to reduce the number of precomputed lines.

In conclusion, we give an example of optimized computation obtained with Algorithm 10.

Example 105. Let $q = 1021$ and $\mathbb{F}_{q^3} = \mathbb{F}_q[\zeta]/(\zeta^3 - 5)$. Let E and P be as in Example 85, Example 91, and Example 99, i.e., let E be the elliptic curve over $\mathbb{F}_q$ of equation $y^2z = x^3 + 230xz^2 + 191z^3$ and let $P = (782\zeta^2 + 802\zeta + 45, 979\zeta^2 + 299\zeta + 133)$. Let $m = 483925 = m_0 + sm_1$, where $m_0 = 274$ and $m_1 = 3$.

Algorithm 9 computes h_{mP} by calling Subalgorithm 1 17 times with input h_{m_1P}, h_{m_2P}, h_{n_1P}, h_{n_2P}, for the following values of (m_1, m_2, n_1, n_2):

$$(1,6,3,4),(1,14,7,8),(1,28,14,15),(1,58,29,30),(1,118,59,60),(1,236,118,119),$$

$$(1,472,236,237),(1,944,472,473),(1,1890,945,946),(1,3780,1890,1891),$$

$$(1,7560,3780,3781),(1,15122,7561,7562),(1,30244,15122,15123),(1,60490,30245,30246),$$

$$(1,120980,60490,60491),(1,241962,120981,120982),(1,483924,241962,241963).$$

Performing the same computation with Algorithm 10, one has that

$$s \notin \mathcal{B}_1 = \{275, 757679, 717376, 508804, 304004, 263701, 527404\},\ 2m_0 + m_1 \neq 0 \mod p$$

and

$$m_0 \notin \mathcal{A}_1 \cup \mathcal{A}_2 = \{1021379, 860162, 161216, 860163, 322435, 161217, 232982, 627181\}.$$

Hence, after computing h_{m_0P}, $h_{(m_0+1)P}$, h_{m_1P}, $h_{(m_0+m_1)P}$, Algorithm 10 calls Subalgorithm 1 three times (in lines 16, 17 and 19) in order to compute h_{mP}. To compute h_{m_0P} and $h_{(m_0+1)P}$, Algorithm 9 calls Subalgorithm 1 with input $h_{m_1P}, h_{m_2P}, h_{n_1P}, h_{n_2P}$ for the following values of (m_1, m_2, n_1, n_2):

$$(1,4,2,3),(1,8,4,5),(1,16,8,9),(1,34,17,18),(1,68,34,35),(1,136,68,69),(1,274,137,138).$$

To compute $h_{(m_0+m_1)P}$, Algorithm 9 calls Subalgorithm 1 with input $h_{m_1P}, h_{m_2P}, h_{n_1P}, h_{n_2P}$ for the following values of (m_1, m_2, n_1, n_2):

$$(1,4,2,3),(1,8,4,5),(1,16,8,9),(1,34,17,18),(1,68,34,35),(1,138,69,70),(1,276,138,139).$$

Hence in total, taking into account overlapping in the computation of h_{m_0P} and $h_{(m_0+m_1)P}$, Algorithm 10 calls Subalgorithm 1 only 12 times.

Chapter 5

Index calculus in trace-zero subgroups

In this chapter, we give an index calculus algorithm for the DLP in trace-zero subgroups of elliptic curves. Our algorithm is a specialization, to trace-zero subgroups, of the index calculus algorithm for general abelian varieties proposed by Gaudry in [46].
We refer to Section 1.4 for basic notions about trace-zero subgroups. Moreover, we refer to Section 1.5 for basic notions about the index calculus algorithm for the DLP.

Gaudry's index calculus algorithm, proposed in [46], is a variant of the index calculus method (see Algorithm 2, Section 1.5.3), that can be applied to groups $V(\mathbb{F}_q)$, of $\mathbb{F}_q$-rational points of an abelian variety V of dimension $d > 1$, defined over a finite field $\mathbb{F}_q$. We recall that such strategy has complexity $O(q^{2-\frac{2}{d}})$, asymptotically in q, where d is regarded as a constant. However, this complexity can grow very fast in the constant d. As a result, the serious drawback of the method is that, in practice, the computation required by the algorithm is not manageable, except for small dimensions.
We have seen in Section 1.4.2 that the trace-zero subgroup T_n can be identified with the group of $\mathbb{F}_q$-rational points of the trace-zero variety $\mathcal{T}_n$, via the process of Weil restriction of scalars. The trace-zero variety $\mathcal{T}_n$ is an abelian variety of dimension $n-1$, defined over $\mathbb{F}_q$. Hence, Gaudry's index calculus algorithm of [46] can be applied to T_n, and it has complexity $O(q^{2\frac{n-2}{n-1}})$ asymptotically in q, regarding n as a constant. Since $|T_n| \in O(q^{n-1})$ by the Hasse-Weil Theorem and by Proposition 45, one has that, in the trace-zero subgroup, the complexity of Gaudry's method is lower than the complexity of the Pollard's rho method for $n \geq 5$. As a consequence, the security against DLP attacks in T_n is comparable to that of classical groups of base field-rational points of elliptic curves, of the same size, only for small values of n. In particular, we have that Gaudry's algorithm in T_3 has the same complexity as Pollard's rho. Hence, the degree three trace-zero subgroup achieves the optimal security against DLP attacks.

Gaudry's method has been previously applied to trace-zero subgroups by Gorla, Massierer in [48]. We describe their variant in Section 1.5.3. To the extent of our knowledge, it is the only other specialization of the algorithm to trace-zero subgroups. Such specialization of Gaudry's algorithm has the drawback of the general method, that we mentioned just above. In fact, it has exponential complexity in n, due to the hardness of solving the polynomial systems that comes from the relation search step of the algorithm (step 2 of Algorithm 2). So our aim in thinking about a new variant of index calculus in T_n was to give systems of polynomial equations for the relation search step, which are easier to solve than those proposed in [48].
In [48], the authors represent each trace-zero point via its x-coordinate. Moreover, they

use Semaev's summation polynomials to get the polynomial systems for the relation search step. On the other hand, we make use of the optimal representation for trace-zero elements given in [49], in order to deal with the least possible number of variables. Moreover, we use the generalized summation polynomials that we have defined in Chapter 3, to get the polynomial systems for the relation search step. We show that our systems are easier to solve than the systems obtained in [48], when $n = 3, 5, 7$. These are the important cases for cryptographic applications.

The chapter is organized as follows. In Section 5.1 we describe the algorithm, and compute explicit equations for $n = 3$. In Section 5.2 we focus on the hardness of solving the polynomial systems that we obtain with our method. We make comparisons with the polynomial systems obtained in [48], and we give some experimental results for $n = 3$. Finally, in Section 5.3, we propose a hybrid variant of our algorithm for the case $n = 5$. Such variant turns out to be faster than the corresponding hybrid version proposed in [48].

Throughout the chapter, we take a finite field $\mathbb{F}_q$ of characteristic different from $2, 3$. Let E be an elliptic curve defined over $\mathbb{F}_q$, written in short Weierstrass form

$$E : y^2z = x^3 + Axz^2 + Bz^3.$$

We denote by $\oplus$ the operation of point addition on E. We denote by $\mathcal{O}$ the neutral element of the operation. Moreover, for each point $P \in E$, we denote by $-P$ the inverse of P with respect to $\oplus$. Let φ the Frobenius endomorphism of E. Moreover, for n odd prime, let T_n be the degree n trace-zero subgroup of E. We suppose that T_n is cyclic of cryptographic prime order p, and that p is much bigger than n. This is the setting of practical applications.

5.1 A new variant of index calculus for trace-zero subgroups

In this section, we describe our specialization of Gaudry's index calculus algorithm to the trace-zero subgroup T_n. We use the coordinates of the optimal representation for trace-zero elements given in [49], in order to deal with the minimal number of variables. Moreover, we make use of the generalized summation polynomials that we have defined in Chapter 3.

We start with recalling the representation for trace-zero elements proposed in [49], and we give some notations to deal with it. For $P \in T_n \setminus \{\mathcal{O}\}$, let $h_P(x, y)$ be the polynomial as in Proposition 62, such that

$$\mathrm{div}(h_P) = \sum_{i=0}^{n-1} \varphi^i(P) - n\mathcal{O}.$$

The polynomial h_P is of the form $h_P(x, y) = h_{P,1}(x) + yh_{P,2}(x) =$

$$(a_1 + a_2x + \cdots + a_{\frac{n-1}{2}}x^{\frac{n-3}{2}} + a_{\frac{n+1}{2}}x^{\frac{n-1}{2}}) + y(a_{\frac{n+3}{2}} + \cdots + a_{n-1}x^{\frac{n-5}{2}} + x^{\frac{n-3}{2}}).$$

By [49, Corollary 4.2], the polynomial h_P has coefficients in $\mathbb{F}_q$, that is $a_i \in \mathbb{F}_q$ for all $i \in \{1, \cdots, n-1\}$.

Notation 106. Denote by $\mathcal{R}(P) = (a_1, \cdots, a_{n-1}) \in \mathbb{F}_q^{n-1}$ the tuple of the coefficients of h_P, as in Proposition 62. By [49, Corollary 4.3], $\mathcal{R}(P)$ is an optimal representation for the trace-zero point P, according to Definition 7 of optimal representation.

Recall that a general index calculus algorithm (Algorithm 2, Section 1.5.3) performs four main steps: (1) choice of the factor base, (2) search for relations, (3) linear algebra, (4) computation of the individual logarithm. In order to apply the index calculus procedure of [46], we regard T_n as the group of $\mathbb{F}_q$-rational points of the trace-zero variety $\mathcal{T}_n$. The factor base consists of the $\mathbb{F}_q$-rational points of a chosen irreducible curve contained in $\mathcal{T}_n$. We look for relations of the form

$$R = P_1 \oplus \cdots \oplus P_{n-1}, \tag{5.1}$$

where $R = (x_R, y_R)$ is a known point of T_n, and the P_i's are unknown points of the factor base. After the collection of relations, one proceeds by performing steps (3) and (4) as in Algorithm 2. We now focus on the first two steps.

(1) Choice of the factor base. We make use of the optimal coordinates for trace-zero elements given in [49] and we follow Notation 106. We choose the following factor base:

$$\mathcal{F} = \{P \in T_n : \mathcal{R}(P) = (a, \underbrace{0, \cdots, 0}_{n-2 \text{ times}})\}.$$

This means that, for $P \in \mathcal{F}$, h_P is of the form $h_P(x, y) = a + yx^{\frac{n-3}{2}}$. We assume that the curve associated to $\mathcal{F}$ is absolutely irreducible, and not contained in any proper subvariety of $\mathcal{T}_n$. Moreover, we assume that $\mathcal{F}$ has about q points. All these assumption are true up to a generic change of coordinates (see [46, Section 2.2]). For $i \in \{1, \cdots, n-1\}$ and P_i unknown factor base point of relation (5.1), we denote by a_i the unknown such that $\mathcal{R}(P_i) = (a_i, 0, \cdots, 0)$.

Remark 107. For $n > 3$ and $i > 1$, there is no $P \in T_n \setminus \{\mathcal{O}\}$ such that $\mathcal{R}(P) = (a_1, \cdots, a_{n-1})$ with $a_j = 0$ for all $j \neq i$. In fact, if this is the case, one has that h_P is of the form

$$h_P(x, y) = a_i x^d + yx^{\frac{n-3}{2}},$$

with $d, \frac{n-3}{2} > 0$ (and so $x | h_P(x, y)$), or

$$h_P(x, y) = y(x^{\frac{n-3}{2}} + a_i x^e),$$

with $e \geq 0$ (and so $y | h_P(x, y)$). In both cases, T_n would contain $\mathbb{F}_q$-rational points of order 2. This is not possible since n is odd.

On the other hand, for $n = 3$, we can choose either the factor base

$$\mathcal{F}_1 = \{P \in T_3 : \mathcal{R}(P) = (-a, 0)\}, \tag{5.2}$$

or the factor base

$$\mathcal{F}_2 = \{P \in T_3 : \mathcal{R}(P) = (0, -b)\}. \tag{5.3}$$

The points of $\mathcal{F}_1$ are the points $P \in T_3$ such that the line through P and its Frobenius conjugates is horizontal (that is, the line is of the form $y = a$). The points of $\mathcal{F}_2$ are the points $P \in T_3$ such that the line through P and its Frobenius conjugates passes through the origin (that is, the line is of the form $y = bx$). We use this notation later, when we deal with the case $n = 3$.

(2) Search for relations. This step consists of three parts. We sum them up just below, then we explain in more details the first and the second part.

(a) *Computation of the polynomial system.* For a given trace-zero point R as in (5.1), we compute a polynomial system in which the unknowns are the optimal coordinates a_i of P_i, for each i. In order to do this, we use generalized summation polynomials, as well as Weil restriction of scalars from $\mathbb{F}_{q^n}$ to $\mathbb{F}_q$. We obtain a polynomial system with n equations with coefficients in $\mathbb{F}_q$, and $n-1$ unknowns $a_1, \cdots, a_{n-1}$. Moreover, each equation of the system is of total degree at most $(n-1)n^{n-2}$.

(b) *Computation of the polynomial system: symmetrization.* We apply a symmetrization process to reduce the total degree of the previous system of a factor n. We obtain a polynomial system with n equations with coefficients in $\mathbb{F}_q$, and $n-1$ unknowns $s_1, \cdots, s_{n-1}$. Moreover, each equation of the system is of total degree at most n^{n-2}.

(c) *Decompression of the solutions and final search.* For each system of point (b), we search solutions in $\mathbb{F}_q$. For each $(\overline{s}_1, \cdots, \overline{s}_{n-1}) \in \mathbb{F}_q^{n-1}$, $\mathbb{F}_q$-solution of the system (b), we recover a $\mathbb{F}_q$-solution $(\overline{a}_1, \cdots, \overline{a}_{n-1}) \in \mathbb{F}_q^{n-1}$ of the system (a), via a process of desymmetrization. Notice that the optimal representation $\mathcal{R}$, as in Notation 106, identifies each trace-zero point with its Frobenius conjugates. This means that $\mathcal{R}^{-1}(\mathcal{R}(P)) = \{P, \varphi(P), \cdots, \varphi^{n-1}(P)\}$ for each $P \in T_n$. Hence, in order to recover the unknown factor base points of relation (5.1) (if they exist), we search for $\overline{P}_i \in \mathcal{R}^{-1}(\overline{a}_i, 0, \cdots, 0)$, $i \in \{1, \cdots, n-1\}$, such that $R = \overline{P}_1 \oplus \cdots \oplus \overline{P}_{n-1}$.

In the next subsection, we focus on parts (a) and (b): the computation of the polynomial system.

5.1.1 Computation of the polynomial system

(a) *Computation of the system.* Let

$$S \in \mathbb{F}_q[a_{1,1}, \cdots, a_{1,n-1}, \cdots, a_{n-1,1}, \cdots, a_{n-1,n-1}][x, y]$$

be a $(n-1, n, \cdots, n)$-generalized summation polynomial. Let $\mathcal{U} \subseteq \overline{\mathbb{F}}_q^{(n-1)^2}$ be the nonempty Zarisky open set in which S satisfies property (3.3), as in Definition 64.

Let $R = (x_R, y_R) \in T_n$. Let $\overline{a}_1, \cdots, \overline{a}_{n-1} \in \mathbb{F}_q$ such that

$$(\overline{a}_1, 0, \cdots, 0, \overline{a}_2, 0, \cdots, 0, \cdots, \overline{a}_{n-1}, 0 \cdots, 0) \in \mathcal{U}. \tag{5.4}$$

Suppose that there exist factor base points $\overline{P}_1, \cdots, \overline{P}_{n-1} \in \mathcal{F}$ that satisfy

$$R = \overline{P}_1 \oplus \cdots \oplus \overline{P}_{n-1}, \text{ with } \mathcal{R}(\overline{P}_i) = (\overline{a}_i, 0, \cdots, 0) \text{ for } i \in \{1, \cdots, n\}. \tag{5.5}$$

Then, by definition of generalized summation polynomials, we have that

$$S(\overline{a}_1, 0, \cdots, 0, \overline{a}_2, 0, \cdots, 0, \cdots, \overline{a}_{n-1}, 0, \cdots, 0)(x_R, y_R) = 0.$$

We take the polynomial

$$S(a_1, 0, \cdots, 0, a_2, 0, \cdots, 0, \cdots, a_{n-1}, 0, \cdots, 0)(x_R, y_R) = S(x_R, y_R)(a_1, \cdots, a_{n-1}) \in \mathbb{F}_{q^n}[a_1, \cdots, a_{n-1}].$$

This polynomial has coefficients in $\mathbb{F}_{q^n}$, since $R \in T_n$ implies that $x_R, y_R \in \mathbb{F}_{q^n}$. Moreover, it is of degree at most n^{n-2} in each variable a_i by Theorem 77.
Our strategy consists of searching $\mathbb{F}_q$-solutions $(\overline{a}_1, \cdots, \overline{a}_{n-1}) \in \mathbb{F}_q^{n-1}$ of the equation

$$S(x_R, y_R)(a_1, \cdots, a_{n-1}) = 0, \tag{5.6}$$

in order to find relations of the type (5.5) for the relation search step of the index calculus algorithm. In fact, relation (5.1) for the relation search step implies the polynomial equation (5.6), except for the special cases of the following remark.

Remark 108. We remark that each relation of the type (5.5) determines a $\mathbb{F}_q$-solution $(\overline{a}_1, \cdots, \overline{a}_{n-1})$ of the equation (5.6) only if condition (5.4) holds true. Hence, there can be a relation of type (5.5), such that $(\overline{a}_1, \cdots, \overline{a}_{n-1})$ does not verify condition (5.4) and $(\overline{a}_1, \cdots, \overline{a}_{n-1})$ is not a solution of (5.6). In this case, the relation (5.5) is not detected by our strategy. Nevertheless, since $\mathcal{U}$ is a nonempty Zarisky open set, we expect our method to detect most of relations (5.5), except for a few special cases.

We are interested in $\mathbb{F}_q$-solutions of equation (5.6). Hence, we apply Weil restriction of scalars, from $\mathbb{F}_{q^n}$ to $\mathbb{F}_q$, to this equation. We refer to Example 42 of Section 1.4.1, as well as to Section 2.1, for details on the technique of Weil restriction of scalars. We obtain a polynomial system of the form

$$\begin{cases} S_1(a_1, \cdots, a_{n-1}) = 0 \\ \quad \vdots \\ S_n(a_1, \cdots, a_{n-1}) = 0 \end{cases}. \tag{5.7}$$

For $i \in \{1, \cdots, n\}$, $S_i \in \mathbb{F}_q[a_1, \cdots, a_{n-1}]$. Moreover, the degree of S_i in each variable is at most n^{n-2}. So the total degree of S_i is at most $(n-1)n^{n-2}$.

(b) *Symmetrization of the polynomial system (5.7).* One has that

$$R = \overline{P}_1 \oplus \cdots \oplus \overline{P}_{n-1}$$

for some $\overline{P}_1, \cdots, \overline{P}_n \in \mathcal{F}$ if and only if

$$R = \overline{P}_{\sigma(1)} \oplus \cdots \oplus \overline{P}_{\sigma(n-1)}$$

for all σ permutation of $\{1, \cdots, n-1\}$. So we can replace system (5.7) with a new polynomial system of the form

$$\begin{cases} E_1(s_1, \cdots, s_{n-1}) = 0 \\ \quad \vdots \\ E_n(s_1, \cdots, s_{n-1}) = 0 \end{cases}. \tag{5.8}$$

For $j \in \{1, \cdots, n-1\}$, $s_j = s_j(a_1, \cdots, a_{n-1})$ is the j-th elementary symmetric polynomial in the variables $a_1, \cdots, a_{n-1}$. For each $i \in \{1, \cdots, n\}$,

$$E_i(s_1, \cdots, s_n) = \sum_{\sigma \text{ permutation of } \{1, \cdots, n-1\}} S_i(a_{\sigma(1)}, \cdots, a_{\sigma(n-1)}).$$

Hence, for each $i \in \{1, \cdots, n\}$, $E_i \in \mathbb{F}_q[s_1, \cdots, s_{n-1}]$. Moreover, the polynomial E_i is of total degree at most n^{n-2}.

To sum up, following the described procedure, one obtains the polynomial system (5.8) for the relation search step of index calculus in T_n. The system consists of n equations with coefficients in $\mathbb{F}_q$, and $n-1$ unknowns $s_1, \cdots, s_{n-1}$. Moreover, each equation of the system is of total degree at most n^{n-2}.

5.1.2 Polynomial systems for $n = 3$

In this subsection, we compute explicit equations and we make some examples of search of relations for the index calculus algorithm, for the degree three trace-zero subgroup T_3. We follow the general strategy given above.
Notice that, for $n = 3$, relation (5.1) is of the form

$$R = P_1 \oplus P_2, \tag{5.9}$$

with $R = (x_R, y_R)$ a known point of T_3 and P_1, P_2 unknown points of the factor base $\mathcal{F}_1$, or unknown points of the factor base $\mathcal{F}_2$. The factor basis $\mathcal{F}_1$ and $\mathcal{F}_2$ are respectively defined in (5.2) and (5.3) of Remark 107. Let $S \in \mathbb{F}_q[\alpha_0, \alpha_1, \beta_0, \beta_1](x, y)$ be the $(2, 3, 3)$-generalized summation polynomial that we have computed in Example 73 of Chapter 3. We follow the procedure described in the previous subsection, to explicitly compute a polynomial system of the form

$$\begin{cases} E_1(\mu, A, B, x_0, x_1, x_2, y_0, y_1, y_2)(s_1, s_2) = 0 \\ E_2(\mu, A, B, x_0, x_1, x_2, y_0, y_1, y_2)(s_1, s_2) = 0 \\ E_3(\mu, A, B, x_0, x_1, x_2, y_0, y_1, y_2)(s_1, s_2) = 0 \end{cases} . \tag{5.10}$$

For each $i \in \{1, 2, 3\}$,

$$E_i(\mu, A, B, x_0, x_1, x_2, y_0, y_1, y_2)(s_1, s_2) \in \mathbb{F}_q[\mu, A, B, x_0, x_1, x_2, y_0, y_1, y_2][s_1, s_2].$$

This polynomial system has the following property. Take q such that $3|(q-1)$ and $\mathbb{F}_q$ finite field of characteristic different from 2, 3. Let E be an elliptic curve defined over $\mathbb{F}_q$, written in short Weierstrass form

$$E : y^2z = x^3 + \overline{A}xz^2 + \overline{B}z^3.$$

Let $\mathbb{F}_{q^3} \cong \mathbb{F}_q[\zeta]/(\zeta^3 - \overline{\mu})$ be a Kummer extension of $\mathbb{F}_q$ of degree three. Finally, let $R = (\overline{x}_0 + \overline{x}_1\zeta + \overline{x}_2\zeta^2, \overline{y}_0 + \overline{y}_1\zeta + \overline{y}_2\zeta^2) \in T_3$. We have that the system (5.10), evaluated in $\overline{\mu}, \overline{A}, \overline{B}, \overline{x}_0, \overline{x}_1, \overline{x}_2, \overline{y}_0, \overline{y}_1, \overline{y}_2$, is the system (5.8) for the point $R \in T_3 \subseteq E(\mathbb{F}_{q^3})$ and the factor base $\mathcal{F}_1$.
Notice that, once we have the polynomial system of the form (5.10), the process of computation of the polynomial system (5.8) for a point $R \in T_3$ is strongly sped up. In fact, one does not have to compute the system (5.8) for each R, following the procedure described in the previous subsection. It is enough to evaluate the equations of (5.10) in the given point R, to obtain the required system (5.8).
We then compute the analogous polynomial system, of the form (5.10), for the factor base $\mathcal{F}_2$. The two polynomial systems that we have obtained are given in Section 3 of the appendix, as polynomial system (PS1) and polynomial system (PS2) respectively.

Example 109. Let $q = 1021$, $\mathbb{F}_{q^3} \cong \mathbb{F}_q[\zeta]/(\zeta^3 - 5)$ and $E : y^2z = x^3 + 978xz^2 + 401z^3$. The trace-zero subgroup T_3 of $E(\mathbb{F}_{q^3})$ is cyclic of prime order $p = 1079509$.

1. We perform the relation search step of the index calculus in T_3 as described in this section, for the point $R = (354\zeta^2+189\zeta+422, 750\zeta^2+660\zeta+584)$ and the factor base $\mathcal{F}_1$. We evaluate the polynomial system (PS1) of the appendix in $\mu = 5, A = 978$, $B = 401$ and in the point R. We obtain the system

$$\begin{cases} 439s_1^3 + 319s_1^2s_2 + 225s_1^2 + 731s_1s_2^2 + 888s_1s_2 + 345s_1 + 1020s_2^3 + 400s_2^2 + 793s_2 + \\ +890 = 0 \\ 462s_1^3 + 990s_1^2s_2 + 422s_1^2 + 959s_1s_2^2 + 118s_1s_2 + 285s_1 + 808s_2^2 + 482s_2 + 368 = 0 \\ 362s_1^3 + 134s_1^2s_2 + 441s_1^2 + 208s_1s_2^2 + 420s_1s_2 + 1020s_1 + 741s_2^2 + 89s_2 + 89 = 0 \end{cases} .$$

Now we perform part (c) of our procedure of relation search for the index calculus. We have that the computed system has a unique $\mathbb{F}_q$-solution $(\overline{s}_1, \overline{s}_2) = (848, 274)$. From the relations $a_1 + a_2 = \overline{s}_1$ and $a_1 a_2 = \overline{s}_2$, we get $\overline{a}_1 = 53$ and $\overline{a}_2 = 795$. Finally, we find $\overline{P}_1 = (300\zeta^2 + 140\zeta, 53) \in \mathcal{R}^{-1}(-\overline{a}_1, 0)$ and $\overline{P}_2 = (605\zeta^2 + 331\zeta, 795) \in \mathcal{R}^{-1}(-\overline{a}_2, 0)$, such that $R = \overline{P}_1 \oplus \overline{P}_2$.

2. We now perform the relation search step for the point $R = (261\zeta^2 + 480\zeta + 340, 402\zeta^2 + 176\zeta + 427)$, choosing the factor base $\mathcal{F}_2$. From the polynomial system (PS2) of the appendix, we obtain the system

$$\begin{cases} 649s_1^3 + 208s_1^2 s_2 + 795s_1^2 + 424s_1 s_2^2 + 372s_1 s_2 + 882s_1 + 965s_2^3 + 16s_2^2 + 298s_2 + \\ +473 = 0 \\ 80s_1^3 + 301s_1^2 s_2 + 213s_1^2 + 468s_1 s_2^2 + 441s_1 s_2 + 935s_1 + 276s_2^3 + 854s_2^2 + 138s_2 + \\ +688 = 0 \\ 525s_1^3 + 1000s_1^2 s_2 + 759s_1^2 + 839s_1 s_2^2 + 951s_1 s_2 + 941s_1 + 612s_2^3 + 182s_2^2 + 154s_2 + \\ +119 = 0 \end{cases}.$$

The system has again one $\mathbb{F}_q$-solution $(\overline{s}_1, \overline{s}_2) = (320, 364)$. Proceeding as before, we get $\overline{b}_1 = 392$ and $\overline{b}_2 = 949$, and we find $\overline{P}_1 = (1011\zeta^2 + 150\zeta + 852, 164\zeta^2 + 603\zeta + 117) \in \mathcal{R}^{-1}(0, -\overline{b}_1)$, $\overline{P}_2 = (63\zeta^2 + 393\zeta + 707, 569\zeta^2 + 292\zeta + 146) \in \mathcal{R}^{-1}(0, -\overline{b}_2)$, such that $R = \overline{P}_1 \oplus \overline{P}_2$.

3. The polynomial system for the factor base $\mathcal{F}_1$ and the point $R = (857\zeta^2 + 982\zeta + 1004, 309\zeta^2 + 5\zeta + 1003)$ has two $\mathbb{F}_q$-solutions, namely $(\overline{s}_1, \overline{s}_2) = (25, 316)$ and $(\overline{s}_3, \overline{s}_4) = (908, 38)$. From these solutions, we get $\overline{a}_1 = 419$, $\overline{a}_2 = 627$, $\overline{a}_3 = 359$, $\overline{a}_4 = 549$. We find the points $\overline{P}_1 = (1011\zeta^2 + 905\zeta, 419) \in \mathcal{R}^{-1}(-\overline{a}_1, 0)$, $\overline{P}_2 = (58\zeta^2 + 20\zeta, 627) \in \mathcal{R}^{-1}(-\overline{a}_2, 0)$, $\overline{P}_3 = (299\zeta^2 + 960\zeta, 359) \in \mathcal{R}^{-1}(-\overline{a}_3, 0)$, $\overline{P}_4 = (736\zeta^2 + 390\zeta, 549) \in \mathcal{R}^{-1}(-\overline{a}_4, 0)$, such that $R = \overline{P}_1 \oplus \overline{P}_2$, $R = \overline{P}_3 \oplus \overline{P}_4$. Hence the point R give us two relations of type (5.9).

5.2 Complexity of the polynomial systems

In this section, we analyze the complexity of solving the polynomial systems of type (5.8) of Section 5.1.1. We make comparisons with the hardness of solving the polynomial systems proposed in [48]. To the extent of our knowledge, the algorithm given in the latter paper is the only other specialization of Gaudry's strategy of [46] to trace-zero subgroups.

We recall that Gaudry's index calculus algorithm in T_n has complexity $O(q^{2\frac{n-2}{n-1}})$ asymptotically in q, where n is regarded as a constant. Hence, from this perspective, it lower the complexity of Pollard's rho for $n \geq 5$. However, the method has exponential complexity in n, due to the hardness of solving the polynomial systems that come from the relation search step. The goal of our work was then to find an alternative to the polynomial systems proposed in [48], for which finding solutions is easier, at least in the cryptographic relevant cases in which n is small. In fact, we will show in the following that our polynomial systems are in general easier to solve than the polynomial systems proposed in [48], for $n = 3, 5, 7$. These are the important cases for cryptographic applications.

We refer to [25] for some background and a survey on the techniques for solving polynomial systems. The complexity of the procedure is in general dominated by the computation of a degree-reverse-lexicographic (DRL) Gröbner basis of the ideal generated by the polynomials that define the equations. Hence, we focus on the complexity of this step.

Nowadays, the fastest known algorithms to compute DRL Gröbner basis are Faugère's F_4 and F_5 algorithm and their variants (see [38], [39], [33]). The matrix versions of these algorithms (see [8], [33], [42]) perform at each step gaussian row-reduction of a particular matrix of increasing size. We give a general idea of how such algorithms work.

We use a matrix version of Faugère's F_4 or F_5 algorithm. The algorithm takes as input polynomials $f_1, \cdots, f_s \in \mathbb{K}[x_1, \cdots, x_t]$, where $\mathbb{K}$ is a field. It returns a DRL Gröbner basis of the ideal $I = (f_1, \cdots, f_s)$ generated by the input polynomials. Let d_{min} be the minimum among the degrees of the input polynomials $f_1, \cdots, f_s$. Moreover, let D be the solving degree of their defining ideal $I = (f_1, \cdots, f_s)$, with respect to the DRL ordering, as in Definition 3.1 of [25]. According to this definition, D is the maximal degree of the polynomials involved in the computation of a DRL Gröbner basis of I. For $d_{min} \leq d \leq D$, the algorithm row-reduces a matrix that derives from the Macaulay matrix $\mathcal{M}_d$ of $f_1, \cdots, f_s$, after removing some eventual useless rows. The (inhomogeneous) Macaulay matrix $\mathcal{M}_d$ of the polynomials $f_1, \cdots, f_s$, with respect to the DRL ordering, is defined for example in [25], [39] and [42]. The columns of this matrix are indexed by the monomials of $\mathbb{K}[x_1, \cdots, x_t]$ of degree less than or equal to d, sorted, from left to right, by decreasing order DRL. Its rows are indexed by the polynomials mf_j, for all $j \in \{1, \cdots, s\}$ and for all monomials $m \in \mathbb{K}[x_1, \cdots, x_t]$ such that the degree of mf_j is less than or equal to d. The entry (i, j) of the matrix $\mathcal{M}_d$ is the coefficient of the monomial corresponding to the column j in the polynomial corresponding to the row i.
One has that the complexity of this algorithm is dominated by the gaussian row-reduction of the computed Macaulay matrices.

Let $\mathcal{S}$ be the polynomial system associated to the polynomials $f_1, \cdots, f_s$. It follows from the previous discussion that the size of the Macaulay matrix $\mathcal{M}_D$ of the polynomials $f_1, \cdots, f_s$, with respect to the DRL ordering, gives a good estimate about the hardness of solving the system. We call the Macaulay matrix $\mathcal{M}_D$ the Macaulay matrix of the system $\mathcal{S}$. Therefore, we are interested in giving bounds for the size of the Macaulay matrix of the polynomial systems (5.8) of Section 5.1.1, as well as of the polynomial systems proposed in [48].

In Section 1.5.3, we explain how the authors of [48] compute their polynomial systems for the relation search step of index calculus in T_n. We give the form of their systems in (1.18). The system (1.18) is analogous to [48, system (7)].

5.2.1 Solving degree and maximal Macaulay matrix for the system

Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be the polynomial systems (5.8) of Section 5.1.1 and (1.18) of Section 1.5.3 respectively. We recall that $\mathcal{S}_1$ consists of n equations of the form $f_i(x_1, \cdots, x_{n-1}) = 0$, where $f_i \in \mathbb{F}_q[x_1, \cdots, x_{n-1}]$ is of degree at most n^{n-2}, for all $i \in \{1, \cdots, n\}$. On the other hand, $\mathcal{S}_2$ consist of $2n-1$ equations of the form $g_i(x_1, \cdots, x_{2n-2}) = 0$, where $g_i(x_1, \cdots, x_{2n-2}) \in \mathbb{F}_q[x_1, \cdots, x_{2n-2}]$ is of degree at most $(n-1)2^{(n-2)}$, for all $i \in \{1, \cdots, 2n-1\}$.

We follow the notation of [25]. Let $I_1 = (f_1, \cdots, f_n)$ and $I_2 = (g_1, \cdots, g_{2n-1})$ be the ideals corresponding to $\mathcal{S}_1$ and $\mathcal{S}_2$ respectively. Let $\tilde{I}_1 = (f_1^h, \cdots, f_n^h)$ and $\tilde{I}_2 = (g_1^h, \cdots, g_{2n-1}^h)$ be the homogeneous ideals associated to I_1 and I_2 respectively. For each i, $f_i^h \in \mathbb{F}_q[x_1, \cdots, x_n]$ is the homogenization of f_i with respect to the last variable. Similarly, for each i, $g_i^h \in \mathbb{F}_q[x_1, \cdots, x_{2n-1}]$ is the homogenization of g_i with respect to the last variable. Let $D_1 = \text{solv.deg}(I_1)$ be the solving degree of $I_1 = (f_1, \cdots, f_n)$ with respect to the DRL ordering. Let $D_2 = \text{solv.deg}(I_2)$ be the solving degree of $I_2 = (g_1, \cdots, g_{2n-1})$

with respect to the DRL ordering. Moreover, let $\mathcal{M}_1$ and $\mathcal{M}_2$ be the Macaulay matrices of the systems $\mathcal{S}_1$ and $\mathcal{S}_2$ respectively. Denote by $m_1 \times n_1$ the size of the matrix $\mathcal{M}_1$. Similarly, denote by $m_2 \times n_2$ the size of the matrix $\mathcal{M}_2$.
We make the following assumptions, which are true generically.

• *Assumption* 1. The ideals $\tilde{I}_1$ and $\tilde{I}_2$ are in generic coordinates (see [25, Definition 1.9]).
• *Assumption* 2. The projective algebraic sets

$$\{P \in \mathbb{P}^{n-1}(\overline{\mathbb{F}}_q) : f_i^h(P) = 0, i \in \{1, \cdots, n\}\}$$

and

$$\{P \in \mathbb{P}^{2n-2}(\overline{\mathbb{F}}_q) : g_i^h(P) = 0, i \in \{1, \cdots, 2n-1\}\}$$

are finite sets.

We have the following result.

Theorem 110. *Suppose to be in the setting established above.*

1. *We have the following bound for the solving degree D_1 of I_1:*

$$D_1 \leq n(n^{(n-2)} - 1) + 1.$$

As regard the size $m_1 \times n_1$ of the Macaulay matrix $\mathcal{M}_1$, one has that

$$m_1 \leq n\binom{n^{(n-1)}}{n-1} \text{ and } n_1 \leq \binom{n^{(n-1)}}{n-1}.$$

2. *We have the following bound for the solving degree D_2 of I_2:*

$$D_2 \leq (2n-1)((n-1)2^{(n-2)} - 1) + 1.$$

As regard the size $m_2 \times n_2$ of the Macaulay matrix $\mathcal{M}_2$, one has that

$$m_2 \leq (2n-1)\binom{(2n-1)(n-1)2^{(n-2)}}{2n-2} \text{ and } n_2 \leq \binom{(2n-1)(n-1)2^{(n-2)}}{2n-2}.$$

Proof. Since we are supposing that Assumption 1 and Assumption 2 hold true, we can apply [25, Corollary 3.23] to obtain the required bound on D_1.
Now we focus on the size of $\mathcal{M}_1$. By definition, the rows of $\mathcal{M}_1$ are indexed by all mf_i, for $i \in \{1, \cdots, n\}$ and m monomial in $x_1, \cdots, x_{n-1}$ of degree $\deg(m) \leq D_1 - \deg(f_i)$. Moreover, the columns of $\mathcal{M}_1$ are indexed by the monomials in $x_1, \cdots, x_{n-1}$ of degree at most D_1. Recall that the number of monomials in m variables of degree at most d is $\binom{d+m}{m}$. Hence, from the bound for D_1 , we obtain the desired bounds for m_1 and n_1.
Proceeding in the same way, we obtain the bounds for D_2, m_2 and n_2. □

One has that, for $n \in \{3, 5, 7\}$:

$$n\binom{n^{(n-1)}}{n-1} < (2n-1)\binom{(2n-1)(n-1)2^{(n-2)}}{2n-2} \text{ and } \binom{n^{(n-1)}}{n-1} < \binom{(2n-1)(n-1)2^{(n-2)}}{2n-2}.$$

Hence the following corollary is a straightforward consequence of Theorem 110.

Corollary 111. *One has that, for $n \in \{3, 5, 7\}$, the size of the Macaulay matrix of the polynomial system $\mathcal{S}_1$ is in general strictly smaller than the size of the Macaulay matrix of the system $\mathcal{S}_2$.*

From Corollary 111 and all previous observations, we conclude that, for the cryptographic important cases of $n = 3, 5, 7$, the polynomial systems the we propose for the relation search step in T_n are in general easier to solve than the systems proposed in [48].

5.2.2 Experiments and timings for n=3

We give some experimental results for the relation search step of the index calculus in T_3. We perform this step following both the method of Section 5.1 and the one of [48].

We take three primes q of size 2^{96}, 2^{112} and 2^{128} respectively, such that $3|(q-1)$. For each of these primes q, we take the finite field $\mathbb{F}_q$, the degree 3 Kummer extension $\mathbb{F}_{q^3} \cong \mathbb{F}_q[\zeta]/(\zeta^3 - \mu)$ with the minimal μ, and five elliptic curves defined over $\mathbb{F}_q$ and written in short Weierstrass form, with T_3 cyclic of prime order. So we have that T_3 is of cryptographic size q^2, that is 2^{192}, 2^{224} or 2^{256}, depending on the chosen q. For each chosen elliptic curve, we take $10'000$ random points $R \in T_3 \setminus \{\mathcal{O}\}$. We perform the relation search step of index calculus for R, in the following ways.

- *Method* 1. We follow the method described in Section 5.1, and use the explicit equations of the system (PS1) of the appendix, as we did in Example 109.

- *Method* 2. We follow the method of [48, Section 5.1], and use the explicit equations of [48, system (10)].

For both methods, we focus on three aspects: the number of relations that we get, the form of the polynomial systems that we obtain and the timings of the procedure.

As regards the number of relations, for both methods we obtain on average 50% relations over the tested random points R. More precisely, about 40% of points gives at least one relation, and 8% gives more than one relation: 6% gives two relations and 2% gives three relations. None of the tested points gives more than three relations. We checked that all multiple relations that we obtained were independent. In the heuristic, one expects that one half of the tested points gives one relation (see [46, Section 2.5]). On the other hand, our examples show that only 40% of points gives relations, but 8% of them gives more than one relation. Therefore, in the end, one has anyway 50% relations over the tested points.

Let now focus on the form of the obtained polynomial systems. Each polynomial system that we compute with Method 1 consists of three equations in two indeterminates. All equations are of degree 3. The solving degree of the systems is 5. The polynomial systems computed with Method 2 consists of three equations in two indeterminates, of degree 7, 8 and 7. The solving degree of the systems is 15. For all points we tested, we did not obtain any exception to these values (nevertheless, for smaller q - of size 2^{10}, 2^{12} and 2^{14} - we obtained some few exceptions: polynomial systems constructed with Method 1, in which one equation was of degree 3 and the other two of degree 2, and whose solving degree was 4 instead of 5; polynomial systems obtained with Method 2, in which the degree of the first or the third equation was 6 instead of 7, and/or the solving degree was 13 or 14 instead of 15). The smaller values of the degrees and of the solving degree tell us that the polynomial systems we computed with our method are easier to solve than the polynomial systems we computed with the method of [48]. This agrees with what we stated in the previous subsection, as well as with the timings that we obtained for solving the systems (see the table in the following).

We give below the timings of the performed procedures. For both methods, we take the average time of the steps we perform. More precisely, for Method 1, we take the average time of the following steps.

(a) Evaluate the system (PS1) of the appendix in the point R.

(b) Solve the obtained polynomial system.

(c) Recover $(\overline{a}_1, \overline{a}_2)$ from a $\mathbb{F}_q$-solution $(\overline{s}_1, \overline{s}_2)$ of the system in step (b) (we call this step de-symmetrization).

(d) Find $\overline{P}_1 \in \mathcal{R}^{-1}(-\overline{a}_1, 0)$, $\overline{P}_2 \in \mathcal{R}^{-1}(-\overline{a}_2, 0)$ such that $R = \overline{P}_1 \oplus \overline{P}_2$ (we call this step decompression).

For Method 2, we take the average time of the following steps.

(a) Evaluate [48, system (10)] in R.

(b) Solve the obtained polynomial system.

(c) Recover the x-coordinates of $\overline{P}_1$ and $\overline{P}_2$.

(d) Find $\overline{P}_1$ and $\overline{P}_2$ such that $R = \overline{P}_1 \oplus \overline{P}_2$.

We refer to Section 5.1 and to [48, Section 5.1] for more details on the general strategies. We take the average time over the $10'000$ random points on each curve, then we take the average over the curves for each $\mathbb{F}_q$. Our results are given in the table below.

Table 10. Timings in seconds of the various phases of the relation search step of index calculus in T_3, following our method (Method 1), and the method of [48] (Method 2).

Bit-length of $\lvert T_3 \rvert$	192	224	256
Method 1			
(a) Evaluate equations	$4.5 \cdot 10^{-4}$	$4.7 \cdot 10^{-4}$	$4.4 \cdot 10^{-4}$
(b) Solve poly system	$7.6 \cdot 10^{-4}$	$8.1 \cdot 10^{-4}$	$8.5 \cdot 10^{-4}$
(c) De-symmetrization	$0.25 \cdot 10^{-4}$	$0.28 \cdot 10^{-4}$	$0.27 \cdot 10^{-4}$
(d) Decompression	$0.77 \cdot 10^{-4}$	$0.76 \cdot 10^{-4}$	$0.7 \cdot 10^{-4}$
Method 2			
(a) Evaluate equations	$6.9 \cdot 10^{-4}$	$7 \cdot 10^{-4}$	$6.5 \cdot 10^{-4}$
(b) Solve poly system	$21.6 \cdot 10^{-4}$	$22 \cdot 10^{-4}$	$23.2 \cdot 10^{-4}$
(c) Recover x's	$0.13 \cdot 10^{-4}$	$0.18 \cdot 10^{-4}$	$0.15 \cdot 10^{-4}$
(d) Recover points	$0.5 \cdot 10^{-4}$	$0.57 \cdot 10^{-4}$	$0.5 \cdot 10^{-4}$

5.3 A hybrid approach for $n = 5$

In Section 5.2.1, we saw that the polynomial systems (5.8), that we propose in Section 5.1 for the relation search step of index calculus in T_n, are in general easier to solve than those proposed in [48], for $n = 3, 5, 7$. Our theoretic statement is confirmed by the computational experiments that we made for the degree three trace-zero subgroup T_3, whose results are given in Section 5.2.2. Nevertheless, already for $n = 5$, the equations for the system (5.8) are not manageable in practice. In this section, we focus on the degree five trace-zero subgroup. We give a variant of the index calculus strategy proposed in Section 5.1, in order to deal with easier polynomial systems. We follow the same approach as in [48, Section 5.2].

1. We choose the factor base

$$\mathcal{F} = \{P \in T_5 : \mathcal{R}(P) = (a, 0, 0, 0)\},$$

as in the general index calculus algorithm of Section 5.1. We look for relations of the form

$$R = P_1 \oplus P_2 \oplus P_3, \tag{5.11}$$

with $R = (x_R, y_R)$ known point of T_5, and the P_i's unknown points of $\mathcal{F}$. In practice, we drop one factor base point from relation (5.1). The device was first proposed by Joux and Vitse in [51]. It allows to compute systems of polynomial equations of much lower degree and with less unknowns. The drawback is that the probability of finding a relation is reduced by a factor q. In fact, the probability of finding a relation of type (5.1) is $\frac{1}{4!}$ for $n = 5$, while the probability of finding a relation of type (5.11) is $\frac{1}{6q}$ (see [46] and [51]). On the other hand, this strategy can determine the difference between getting polynomial systems that are computationally impossible to solve, and getting polynomial systems that are solvable in practice.

2. Since equations for relation (5.11) are still not manageable, we apply a hybrid method as proposed in [15] and [29, Section 7]. Such method combines exhaustive search with solving polynomial systems with algebraic techniques. Namely, for all $\overline{P}_3 \in \mathcal{F}$, with $\mathcal{R}(\overline{P}_3) = (\overline{a}_3, 0, 0, 0)$, $\overline{a}_3 \in \mathbb{F}_q$, we search for relations of the form

$$(R \oplus (-\overline{P}_3)) \oplus (-P_1) = P_2, \tag{5.12}$$

with $\hat{R} = (R \oplus (-\overline{P}_3))$ known point of $T_5 \setminus \{\mathcal{O}\}$ and P_1, P_2 unknown points of $\mathcal{F}$, that is $\mathcal{R}(P_1) = (a_1, 0, 0, 0)$, $\mathcal{R}(P_2) = (a_2, 0, 0, 0)$, a_1 and a_2 unknowns. In this way, one has to solve about q small polynomial systems instead of a big one. As for the idea of Joux and Vitse, this device can determine the difference between obtaining computationally unsolvable systems or practically solvable systems. In the next subsection, we explain in detail how to compute a polynomial system for a relation of type (5.12).

5.3.1 Construction of the polynomial system

We are interested in relations of type (5.12). Let $\hat{R} = R - \overline{P}_3 = (\hat{x}, \hat{y}) \in T_5$, $\mathcal{R}(\hat{R}) = (\hat{a}_1, \hat{a}_2, \hat{a}_3, \hat{a}_4) \in \mathbb{F}_q^4$, $h_{\hat{R}}(x, y) = (\hat{a}_1 + \hat{a}_2 x + \hat{a}_3 x^2) + y(\hat{a}_4 + x)$. Let

$$S(a_{1,1}, \cdots, a_{1,4}, a_{2,1}, \cdots, a_{2,4})(x, y) \in \mathbb{F}_q[a_{1,1}, \cdots, a_{1,4}, a_{2,1}, \cdots, a_{2,4}][x, y]$$

be a $(2, 5, 5)$-generalized summation polynomial. Let $\mathcal{U} \subseteq \overline{\mathbb{F}}_q^8$ be the Zarisky open set in which S satisfies property (3.3), as in Definition 64. Take the polynomial

$$S_{\hat{R}}(a_1)(x, y) = S(\hat{a}_1, \hat{a}_2, \hat{a}_3, \hat{a}_4, -a_1, 0, 0, 0)(x, y) \in \mathbb{F}_q[a_1][x, y]. \tag{5.13}$$

Notice that, by definition of generalized summation polynomial, for each $\overline{P}_1 \in \mathcal{F}$ such that $\mathcal{R}(\overline{P}_1) = (\overline{a}_1, 0, 0, 0)$ and $(\hat{a}_1, \hat{a}_2, \hat{a}_3, \hat{a}_4, -\overline{a}_1, 0, 0, 0) \in \mathcal{U}$, we have that

$$\operatorname{div}(S_{\hat{R}}(\overline{a}_1)(x, y)) = \sum_{0 \leq i,j \leq 4} (\varphi^i(\hat{R}) \oplus \varphi^j(-\overline{P}_1)) - 25\mathcal{O}.$$

In order to compute $S_{\hat{R}}$, one can evaluate a $(2, 5, 5)$-generalized summation polynomial $S \in \mathbb{F}_q[a_{1,1}, \cdots, a_{1,4}, a_{2,1}, \cdots, a_{2,4}][x, y]$ in $(a_{1,1}, \cdots, a_{1,4}, a_{2,1}, \cdots, a_{2,4}) = (\hat{a}_1, \hat{a}_2, \hat{a}_3, \hat{a}_4, -a_1, 0, 0, 0)$, as in the definition (5.13). Nevertheless, since a $(2, 5, 5)$-generalized summation polynomial is not manageable in practice, we give below an algorithm to compute $S_{\hat{R}}(a_1)(x, y)$ directly, for each given $\hat{R}$. This algorithm is a specialization of Algorithm 7 of Chapter 3, Section 3.3. Hence we omit the proof of its correctness, since it is analogous to the proof of Theorem 71.

Notation 112. For $i \in \{1, \cdots, 5\}$, let $P_i = P_i(a_1)$ and $x_i = x_i(a_1)$, such that, for each $\overline{P}_1 \in \mathcal{F}$, with $\mathcal{R}(\overline{P}_1) = (\overline{a}_1, 0, 0, 0)$, we have $\varphi^{i-1}(-\overline{P}_1) = P_i(\overline{a}_1) = (x_i(\overline{a}_1), \frac{\overline{a}_1}{x_i(\overline{a}_1)})$.

Algorithm 11 (Computation of $S_{\hat{R}}(a_1)(x, y)$)**.**

Input : A point $\hat{R} \in T_5 \setminus \{\mathcal{O}\}$
Output: The polynomial $S_{\hat{R}}(a_1)(x, y) = S_1 + yS_2 \in \mathbb{F}_q[a_1][x, y]$

for $i \in \{1 \cdots, 5\}$ ▷ r_i line through P_i and $\hat{R}$ as a rational function in the variables x_i, a_1, x, y
  $r_i(x_i, a_1, x, y) \leftarrow (\hat{y} - \frac{a_1}{x_i})x + (x_i - \hat{x})y + ((\hat{x} - x_i)\frac{a_1}{x_i} + (\frac{a_1}{x_i} - \hat{y})x_i) \in (\mathbb{F}_{q^5}(x_i))[a_1, x, y]$
end for
$K(x_1, \cdots, x_5, a_1, x, y) \leftarrow \prod_{i=1}^{5} r_i$
Write $K(x_1, \cdots, x_5)$ as a function of the elementary symmetric polynomials $e_1, \cdots, e_5$
  $E_1 \leftarrow 0$, $E_2 \leftarrow A$, $E_3 \leftarrow -B$, $E_4 \leftarrow 0$, $E_5 \leftarrow a_1^2$
for $i \in \{1, \cdots, 5\}$
  Replace e_i by E_i in K
end for ▷ Now $K = K(a_1)(x, y) \in \mathbb{F}_{q^5}(a_1)[x, y]$
  Notation: $K = K(a_1)(x, y) = \sum_{0 \le i+j \le 5} c_{i,j}(a_1)x^i y^j$, with $c_{i,j}(a_1) \in \mathbb{F}_{q^5}(a_1)$ for all i, j
for $h \in \{1, \cdots, 4\}$
  for $0 \le i + j \le 5$
    $c_{i,j,h}(a_1) \leftarrow$ obtained from $c_{i,j}(a_1) \in \mathbb{F}_{q^5}(a_1)$, raising all its coefficients to the q^h
  end for
end for
$K_0 \leftarrow K$
for $h \in \{1, \cdots, 4\}$
  $K_h \leftarrow \sum_{0 \le i+j \le 5} c_{i,j,h}(a_1)x^i y^j$
end for ▷ $K_h(a_1)(x, y)$ is the product of the five lines through $\varphi^h(\hat{R})$ and P_i, for $i \in \{1, \cdots, 5\}$
$K(a_1)(x, y) = \prod_{h=0}^{4} K_h$
Use equality $S_1(x) - yS_2(x) = K(x, y)/((h_{\overline{R}})^5(-a_1 + yx)^5) \mod y^2 - f(x, 1)$, removing denominators
return S

Once computed $S_{\hat{R}}(a_1)(x, y) = S_1(a_1)(x) + yS_2(a_1)(x) \in \mathbb{F}_q[a_1][x, y]$, we make the following observation. For $\overline{P}_1, \overline{P}_2 \in \mathcal{F}$ with $\mathcal{R}(\overline{P}_1) = (\overline{a}_1, 0, 0, 0)$, $\mathcal{R}(\overline{P}_2) = (\overline{a}_2, 0, 0, 0)$, such that $(\hat{a}_1, \hat{a}_2, \hat{a}_3, \hat{a}_4, -\overline{a}_1, 0, 0, 0) \in \mathcal{U}$ and $\hat{R} \oplus (-\overline{P}_1) = \overline{P}_2$, we have that

$$(f(x, 1)x^2 - \overline{a}_2^2) | xS_1(\overline{a}_1)(x) - \overline{a}_2 S_2(\overline{a}_1)(x). \tag{5.14}$$

From the linear system associated to this divisibility condition, we derive a system of five polynomial equations $f_i(a_1, a_2) = 0$, with coefficients in $\mathbb{F}_q$ and a_1, a_2 unknowns. In general, we obtain that f_1 is of total degree 8, of degree 5 in a_1 and of degree 3 in a_2, f_2 and f_3 are of total degree 9, of degree 5 in a_1 and of degree 4 in a_2, f_4 and f_5 are of total degree 10, of degree 5 in both a_1 and a_2.
Notice that we have $\hat{R} \oplus (-\overline{P}_1) = \overline{P}_2$ for some $\overline{P}_1, \overline{P}_2 \in \mathcal{F}$ if and only if $\hat{R} \oplus (-\overline{P}_2) = \overline{P}_1$. We can then apply a symmetrization process to the computed polynomial system in order to reduce the total degree of the equations. This part is analogous to step (b) of the algorithm of Section 5.1. After performing this step, we obtain a system of five polynomial equations $g_i(s_1, s_2) = 0$, with coefficients in $\mathbb{F}_q$ and s_1, s_2 unknowns. In general, we have that all g_i's are of degree 5, g_1 is of degree 4 in both in s_1 and s_2, g_2 and g_3 are of degree 5 in s_1 and of degree 4 in s_2, g_4 and g_5 have degree 5 in both s_1 and s_2.
In such a way we compute the polynomial system for the relation search step, for the random point $R \in T_5 \setminus \{\mathcal{O}\}$ and the point $\overline{P}_3 \in \mathcal{F}$, such that $\hat{R} = R - \overline{P}_3 \in T_5 \setminus \{\mathcal{O}\}$.

As in the algorithm of Section 5.1, we now search for $\mathbb{F}_q$-solutions $(\overline{s}_1, \overline{s}_2)$ of this system. Then, we apply relations $\overline{s}_1 = a_1 + a_2$, $\overline{s}_2 = a_1 a_2$, in order to find $(\overline{a}_1, \overline{a}_2)$ and $\overline{P}_1 \in \mathcal{R}^{-1}(\overline{a}_1, 0, 0, 0)$, $\overline{P}_2 \in \mathcal{R}^{-1}(\overline{a}_2, 0, 0, 0)$, such that $\hat{R} \oplus (-\overline{P}_1) = \overline{P}_2$.

Example 113. Let $q = 31$, $\mathbb{F}_{q^5} \cong \mathbb{F}_q[\zeta]/(\zeta^5 - 2)$ and $E : y^2 z = x^3 + 15xz^2 + 20z^3$. The trace-zero subgroup T_5 of $E(\mathbb{F}_{q^5})$ is cyclic of prime order $p = 1060051$, and $|\mathcal{F}| = 40$. We perform the relation search step of the index calculus in T_5 as described in this subsection, for the point $R = (22\zeta^4 + 24\zeta^3 + 18\zeta^2 + 25\zeta + 23, 6\zeta^4 + \zeta^3 + 13\zeta^2 + 13\zeta + 9) \in T_5$. For all $\overline{P}_3 \in \mathcal{F}$, with $\mathcal{R}(\overline{P}_3) = (\overline{a}_3, 0, 0, 0)$, $\overline{a}_3 < 16$, we do not find any relation of type (5.11). Then we try $\overline{P}_3 = (\zeta^4 + 5\zeta^3 + 13\zeta^2 + 11\zeta, 19\zeta^4 + 2\zeta^3 + 22\zeta^2 + 19\zeta) \in \mathcal{F}$, with $\mathcal{R}(P_3) = (16, 0, 0, 0)$. Let $\hat{R} = R - \overline{P}_3 = (17\zeta^4 + 24\zeta^3 + 2\zeta^2 + 13\zeta + 26, 6\zeta^4 + 26\zeta^3 + 18\zeta^2 + 29\zeta)$. We use Algorithm 11 to compute $S_{\hat{R}}(a_1)(x, y)$. Next, we use relation (5.14) and apply the symmetrization process to obtain the polynomial system:

$$\begin{cases} 24s_1^4 s_2 + 18s_1^3 s_2^2 + 26s_1^3 s_2 + 22s_1^3 + 22s_1^2 s_2^3 + 27s_1^2 s_2^2 + 7s_1^2 s_2 + 29s_1^2 + 29s_1 s_2^3 + 17s_1 s_2^2 \\ +13s_1 s_2 + 23s_1 + 18s_2^4 + 12s_2^3 + 18s_2^2 + 16s_2 + 20 = 0 \\ 2s_1^5 + 30s_1^4 s_2 + 4s_1^4 + 6s_1^3 s_2^2 + 7s_1^3 s_2 + 24s_1^3 + 27s_1^2 s_2^3 + 5s_1^2 s_2^2 + 8s_1^2 s_2 + 25s_1^2 + 22s_1 s_2^4 \\ +15s_1 s_2^3 + 4s_1 s_2^2 + 11s_1 s_2 + 27s_1 + 13s_2^4 + 11s_2^3 + 23s_2^2 + 29s_2 + 7 = 0 \\ 17s_1^5 + 7s_1^4 s_2 + 15s_1^4 + 24s_1^3 s_2^2 + 22s_1^3 s_2 + 7s_1^3 + 22s_1^2 s_2^3 + 29s_1^2 s_2^2 + 29s_1^2 s_2 + 16s_1^2 + 29s_1 s_2^4 \\ +27s_1 s_2^3 + 13s_1 s_2^2 + 11s_1 s_2 + 29s_1 + 11s_2^4 + 25s_2^3 + 4s_2^2 + s_2 + 30 = 0 \\ 9s_1^5 + 22s_1^4 s_2 + 21s_1^4 + 15s_1^3 s_2^2 + 26s_1^3 s_2 + 14s_1^3 + 16s_1^2 s_2^3 + 19s_1^2 s_2^2 + 29s_1^2 s_2 + s_1^2 + 19s_1 s_2^4 \\ +30s_1 s_2^3 + 7s_1 s_2^2 + 7s_1 s_2 + 10s_1 + 29s_2^5 + 9s_2^4 + 2s_2^3 + 23s_2^2 + 18s_2 + 18 = 0 \\ 15s_1^5 + 30s_1^4 s_2 + 7s_1^4 + 11s_1^3 s_2^2 + 28s_1^3 s_2 + 2s_1^3 + 12s_1^2 s_2^3 + 2s_1^2 s_2^2 + 17s_1^2 s_2 + 29s_1^2 + 18s_1 s_2^4 \\ +14s_1 s_2^3 + 13s_1 s_2^2 + 6s_1 s_2 + 29s_1 + 11s_1^5 + 7s_2^4 + 21s_2^3 + 7s_2^2 + 20s_2 = 0 \end{cases}.$$

This system has one $\mathbb{F}_q$-solution, namely $(\overline{s}_1, \overline{s}_2) = (6, 20)$. Hence $(\overline{a}_1, \overline{a}_2) = (15, 22)$, from which we find $\overline{P}_1 = (4\zeta^4 + 18\zeta^3 + 26\zeta^2 + 26\zeta, 17\zeta^4 + 30\zeta^3 + 18\zeta^2 + 3\zeta) \in \mathcal{R}^{-1}(15, 0, 0, 0)$, $\overline{P}_2 = (\zeta^4 + 2\zeta^3 + 4\zeta^2 + 6\zeta, 22\zeta^4 + 12\zeta^3 + 12\zeta^2 + 2\zeta) \in \mathcal{R}^{-1}(22, 0, 0, 0)$, such that $\hat{R} \oplus (-\overline{P}_1) = \overline{P}_2$. This is equivalent to saying $R = \overline{P}_1 \oplus \overline{P}_2 \oplus \overline{P}_3$.

5.3.2 Experiments, timings and comparisons

We compare the method proposed in this section with the analogous method given in [48, Section 5.2]. We perform the relation search step for index calculus in T_5 following the hybrid procedure described above. We take the average time to compute and to solve the involved polynomial systems. Moreover, we take the average of the number of points R that we have to test, and of the number of polynomial systems that we have to solve, before finding a relation. We compare our results with the analogous results of [48].

In [48], the authors also apply Joux and Vitse's strategy together with a hybrid approach to treat the relation search step in T_5. They use Semaev's summation polynomials rather than generalized summation polynomials to compute the systems, as in the general case. The polynomial systems they obtain consist of 7 equations $f_i(x_1, x_2, x_3, x_4)$ with coefficients in $\mathbb{F}_q$ and 4 unknowns. The first five equations have total degree 8, and degree 4 in each variable. The last two equations are of the form $f_6(x_1, x_2) = 0$ and $f_7(x_3, x_4) = 0$. They have total degree 32, and degree 30 in both variables. Due to the lower number of equations and unknowns and to the smaller degree, our polynomial systems will be in general easier to solve.

We implemented our procedure for small q. Namely, for $q = 2^5 - 1$, $2^6 - 23$, $2^7 - 27$, $2^8 - 15$, we take $\mathbb{F}_q$, the degree 5 Kummer extension $\mathbb{F}_{q^5} \cong \mathbb{F}_q[\zeta]/(\zeta^5 - \mu)$ with the minimal μ, and one elliptic curve defined over $\mathbb{F}_q$ written in short Weierstrass form, with T_5 cyclic of prime order, and $|\mathcal{F}| > q$. For each curve we take random trace-zero points R and

perform our procedure till we find a relation. For each R and for each $\overline{P}_3$ we tested, we take the time to compute the polynomial system for relation (5.12), as well as the time to solve it, then we make the average over all $\overline{P}_3$ and R. We obtain that, for all q, it takes about 0.2 seconds to compute a system and $5 \cdot 10^{-4}$ seconds to solve it. We compare with the results in [48, Table 2]. We observe that the whole procedure of computing and solving our systems is much faster. In fact, solving the systems computed there requires about 1.3 seconds for the same values of q. Indeed, the bottleneck of the strategy is not any more the hardness of such polynomial systems, but the big amount of them, that one has to solve before getting a relation: we expect that it is about $6q^2$. For each q, we perform our routine for 10 times (or 5 times in the case $q = 2^8 - 15$). Then we take the average of the number of points R that we have to test before finding a relation, as well as of the number of systems that we have to solve for this. We give our results in the table below. Notice that we are actually able to find relations for $q = 2^8 - 15$, while the values given in [48, Table 2] for the same q are only theoretical estimates.

Table 11. Number of random points that we tested and number of system that we solved before finding a relation of type (5.11) (average over 10 trials for the first three q, over 5 trials for the last q).

q	$2^5 - 1$	$2^6 - 23$	$2^7 - 40$	$2^8 - 15$
Number of R tested	78	144	205	715
Number of systems solved	2580	7003	26345	189000

Conclusions

We studied trace-zero subgroups of elliptic curves and twisted Edwards curves from the cryptographic point of view. We remark that, up to now, the theoretical research on the topic is not supported by real applications. However, our analysis confirmed that trace-zero subgroups are suitable for the construction of DLP-based cryptosystems. More specifically, trace-zero subgroups of degree three have the same DLP security as groups of prime field-rational points of elliptic curves of the same size. Furthermore, the computation of scalar multiplication, as well as the computation of the cardinality of the group, is faster in the first family of groups. As a consequence, DLP cryptosystems based on degree three trace-zero subgroups will have the same security and better performance than those based on standard groups of points of elliptic curves. Therefore, we strongly recommend the implementation of such cryptosystems in cryptographic libraries.

We gave some new interesting results in trace-zero cryptography. First, we proposed two optimal representations for trace-zero subgroups of twisted Edwards curves, with efficient compression and decompression algorithms. It turned out that the correspondent algorithms on elliptic curves in Weierstrass form are faster, due to the fact that the involved formulas and equations are sparser and of smaller degree. Nevertheless, our representations are useful in the case of direct use of twisted Edwards curves. In fact, they allow us to perform computation directly on the given curve, without passing to the Weierstrass form. The relevance of our results is strengthened by the cryptographic importance of twisted Edwards curves, which are secure against side-channel attacks, in contrast to elliptic curves in short Weierstrass form. Therefore, our compression and decompression algorithms should be implemented in a possible future library for trace-zero cryptography, together with the correspondent algorithms on elliptic curves in short Weierstass form. We underline that the implementation we performed during our work was for example purposes only, and that careful programming work is required in order to optimize the algorithms for practical use.

We then proposed a new algorithm to perform scalar multiplication in the degree three trace-zero subgroup, using optimal coordinates to represent the group elements. The novelty of our algorithm consists of performing computation directly in the optimal coordinates, without compression and decompression of points. This is the first algorithm for scalar product in the trace-zero subgroups that follows such approach and in particular it proves that computing in compressed coordinates is possible. Also in this case, our implementation of the algorithm had only illustrative purposes, and some work has to be done in order to optimize it. On the other hand, we point out that the standard non-direct approach for scalar multiplication in the group is more efficient. This is due to the extreme speed of the scalar product operation for elliptic curves, to the further speeding up of the operation for trace zero points via the strategy of Frobenius reduction, and to the efficiency of the compression and decompression algoritms. For all these reasons, we intuitively expected that the non-direct approach to the operation was to prefer. Hence our

result gives an effective confirmation to the heuristic expectations on the topic. Moreover, our strategy has the intrinsic value to give a new perspective on the arithmetic in the group, having interesting, non-trivial analogies with the Montgomery ladder algorithm for x-only scalar multiplication for elliptic curves.

Finally, we gave a new variant of the index calculus algorithm for the DLP in trace-zero subgroups of elliptic curves. Our variant is a specialization to trace-zero subgroups of Gaudry's index calculus algorithm for general abelian varieties. The aim was not to improve the asymptotical complexity of Gaudry's method, but to face its most serious drawback: the effective unsolvability of the polynomial systems computed in the relation search step. We made use of a non-trivial generalization of Semaev's summation polynomials, that is given in this thesis for the first time. Such tool allows to write polynomial systems in the optimal number of variables. In this way, we could lower the complexity of the polynomial systems in the relation search step of the algorithm, for extension degree 3, 5, 7, that are the most relevant cases in cryptographic applications.

Bibliography

[1] L. M. Adleman, J. DeMarrais, M.D. Huang, *A subexponential algorithm for discrete logarithms over hyperelliptic curves of large genus over $GF(q)$*, Theoret. Comput. Sci., vol. 226 (1999), 7-18.

[2] C. Arène, T. Lange, M. Naehrig, C. Ritzenthaler, *Faster computation of the Tate pairing*, Journal of Number Theory, vol. 131 no. 5 (2011), 842-857.

[3] R. B. Ash, *Abstract Algebra: The Basic Graduate Year*, October 2000, available at https://faculty.math.illinois.edu/ r-ash/Algebra.html.

[4] R. M. Avanzi, E. Cesena, *Trace zero varieties over fields of characteristic 2 for cryptographic applications*, Proceedings of the First Symposium on Algebraic Geometry and Its Applications – SAGA '07 (2007), 188-215.

[5] R. M. Avanzi, E. Cesena, T. Lange, *Trace Zero Varieties for Cryptographic Applications*, slides for SPEED-CC, Berlin, October 13th 2009, HGI and Faculty of Mathematics, Ruhr University Bochum.

[6] R. M. Avanzi, H. Cohen, C. Doche, G. Frey, T. Lange, K. Nguyen, F. Vercauteren, *Handbook of Elliptic and Hyperelliptic Curve Cryptography*, Discrete Mathematics and Its Applications, Series Editor K. H. Rosen, Chapman & Hall/CRC (2006).

[7] L. Badescu, *Istituzioni di Geometria Superiore 2*, available at http://www.dima.unige.it/ badescu/attivita%20didattica/Laurea%20Specialistica /igs2/igs2(book).pdf

[8] M. Bardet, J. C. Faugère, B. Salvy, *On the complexity of the F_5 Gröbner basis algorithm*, J. Symbolic Comput., vol 70 (2015), 49-70.

[9] E. Barker, A. Roginsky, *Transitions: Reccomendation for Transitioning the Use of Cryptographic Algorithms and Key Lenghts*, NIST Special Publication 800-131A, Rev. 1, November 2015.

[10] D. J. Bernstein, *Curve25519: New Diffie-Hellman Speed Records*, Public Key Cryptography, LNCS vol. 3958, Springer-Verlag (2006), 207-228.

[11] D. J. Bernstein, P. Birkner, T. Lange, C. Peters, *Optimizing Double-Base Elliptic Curve Single-Scalar Multiplication*, Progress in Cryptology - INDOCRYPT 2007, LNCS vol. 4859, Springer-Verlag (2007), 167-182.

[12] D. J. Bernstein, P. Birkner, M. Joye, T. Lange, C. Peters, *Twisted Edwards Curves*, Progress in Cryptology - AFRICACRYPT 2008, LNCS vol. 5023, Springer-Verlag (2008), 389-405.

[13] D. J. Bernstein, T. Lange, *Faster addition and doubling on elliptic curves*, Advances in Cryptology - ASIACRYPT 2007, LNCS vol. 4833, Springer-Verlag (2007), 29-50.

[14] D. J. Bernstein, T. Lange, *Inverted Edwards Coordinates*, Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, LNCS vol .4851, Springer-Verlag (2007), 20-27.

[15] L. Bettale, J.C. Faugère, L. Perret, *Hybrid approach for solving multivariate systems over finite fields*, J. Math. Cryptol., vol. 3 (2009), 177-197.

[16] G. Bianco, E. Gorla, *Compression for trace zero points on twisted Edwards curves*, Journal of Mathematical Cryptology 10, no. 1 (2016), 15-34.

[17] G. Bianco, E. Gorla, *Scalar multiplication in compressed coordinates in the trace-zero subgroup*, submitted (2017), available at https://arxiv.org/abs/1709.04178.

[18] G. Bianco, E. Gorla, *Index calculus in trace-zero subgroups and generalized summation polynomials*, preprint (2017).

[19] P. Birkner, *Efficient Arithmetic on Low-Genus Curves*, Ph.D. thesis (2009), available at http://alexandria.tue.nl/extra2/200910363.pdf.

[20] G. Blady, *Die Weil-Restriktion elliptischer Kurven in der Kryptographie*, Master's thesis, University GHS Essen, 2002.

[21] M. Bolli, P. Kofmel, *WhatsApp End-to-End Encryption*, seminar paper for BTI 7311 Computer Science Seminar, Bern University of Applied Science, January 2017, available at tmp.bolli.us/signal.pdf

[22] W. Bosma, J. Cannon, C. Fieker, A. Steel (eds.), *Handbook of MAGMA functions*, Version 2.19, Sydney, April 2013.

[23] W. Bosma, J. Cannon, C. E. Playoust, *The Magma algebra system. I. The user language*, J. Symbolic Comput., vol. 24 (1997), 235-265.

[24] E. Brier, M. Joye, *Weierstrass Elliptic Curves and Side Channels Attacks*, Public Key Cryptography, LNCS vol. 2274, Springer-Verlag (2002), 335-345.

[25] A. Caminata, E. Gorla, *Solving Multivariate Polynomial Systems and an Invariant from Commutative Algebra*, June 2017, available at https://arxiv.org/abs/1706.06319

[26] Certicom Research, *Certicom ECC Challenge*, November 2009, available at https://www.certicom.com/content/dam/certicom/images/pdfs/challenge-2009.pdf

[27] E. Cesena, *Pairing with Supersingular Trace Zero Varieties Revisited*, EUROCRYPT 2009, Cologne, Germany, April 2009, available at http://porto.polito.it/2373213/

[28] E. Cesena, *Trace zero varieties in pairing-based cryptography*, Ph.D. Thesis (2010), available at https://ricerca.mat.uniroma3.it/dottorato/Tesi/tesicesena.pdf.

[29] N. Curtois, A. Klimov, J. Patarin, A. Shamir, *Efficient Algorithms for solving overdefined systems of multivariate polynomial equations*, Advances in Cryptology - EUROCRYPT 2000, LNCS vol. 1807, Springer-Verlag (2000), 392-407.

[30] C. Diem, *On the discrete logarithm problem in elliptic curves*, Compos. Math., vol. 147 (2011), 75-104.

[31] C. Diem, *On the discrete logarithm problem in elliptic curves II*, Algebra and Number Theory, vol. 7 (2013), 1281-1323.

[32] W. Diffie, M. Hellman. *New directions in cryptography*, IEEE Trans. Inform. Theory, vol. 22 no. 6 (1976), 644-654.

[33] C. Eder, J. C. Faugère, *A survey on signature-based Gröbner basis computations*, Proceedings of the 39th international symposium on symbolic and algebraic computation - ISSAC 2014, 178-185.

[34] H. M. Edwards, *A normal form for elliptic curves*, Bulletin of the American Mathematical Society, vol. 44 (2007), 393-422.

[35] D. Eisebund, *Commutative Algebra with a View Toward Algebraic Geometry*, Graduate Texts in Mathematics, vol. 150, Springer-Verlag (1995).

[36] A. Enge, *Computing discrete logarithms in high-genus hyperelliptic Jacobians in provably subexponential time*, Math. Comp., vol. 71, no. 238 (2002) 729-742.

[37] A. Enge, P. Gaudry, *A general framework for subexponential discrete logarithms algorithms*, Acta Arith. vol. 102 (2002), 83-103.

[38] J. C. Faugère, *A new efficient algorithm for computing Gröbner bases (F_4)*, Journal of Pure and Applied Algebra, vol. 139 (1999), 61-88.

[39] J. C. Faugère, *A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5)*, Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation, ISSAC '02, New York 2002, 75-83.

[40] J. C. Faugère, P. Gaudry, L. Huot, G. Renault, *Using Symmetries in the Index Calculus for Elliptic Curves Discrete Logarithm*, Journal of Cryptology, vol. 27, no. 4 (2014), 595-635.

[41] J. C. Faugère, L. Hout, A. Joux, G. Renault, V. Vitse, *Symmetrized summation polynomials: using small order torsion points to speed up elliptic curve index calculus*, EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, LCNS vol. 8441, Springer (2014), 40-57.

[42] J. C. Faugère, L. Huot, G. Renault, *Solving efficiently structured polynomial systems and Applications in Cryptology*, talk at ECC 2011, The 15th workshop on Elliptic Curve Cryptography, INRIA, Nancy, France.

[43] G. Frey, *Applications of arithmetical geometry to cryptographic constructions*, Proceedings of the 5th International Conference on Finite Fields and Applications, Springer (1999), 128-161.

[44] W. Fulton, *Algebraic Curves, An Introduction to Algebraic Geometry*, January 2008, available at http://www.math.lsa.umich.edu/wfulton/CurveBook.pdf

[45] P. Gaudry, *An algorithm for solving the discrete log problem on hyperelliptic curves*, Advances in Cryptology - EUROCRYPT 2000, LNCS vol. 1807, Springer-Verlag (2000), 19-34.

[46] P. Gaudry, *Index calculus for abelian varieties of small dimension and the elliptic curve discrete logarithm problem*, J. Symbolic Comput., vol. 44 (2009), 1690-1702.

[47] E. Gorla, M. Massierer, *Point compression for the trace zero subgroup over a small degree extension field*, Designs, Codes and Cryptography, vol. 75 no. 2 (2015), 335-357.

[48] E. Gorla, M. Massierer, *Index calculus in the trace zero variety*, Advances in Mathematics of Communications, vol. 9 no. 4 (2015), 515-539.

[49] E. Gorla, M. Massierer, *An optimal representation for the trace zero variety*, Designs, Codes and Cryptography, vol. 83 no. 3 (2017), 519-548.

[50] G. H. Hardy, J. E. Littlewood, *Some problems of diophantine approximation: Part II. The trigonometrical series associated with the elliptic θ-functions*, Acta Mathematica (1914), 37-225.

[51] A. Joux, V. Vitse, *Elliptic curve discrete logarithm problem over small degree extension fields. Application to the static Diffie-Hellman problem on $E(\mathbb{F}_{q^5})$*, J. Cryptology, vol. 26 (2013), 119-143.

[52] N. Koblitz, *Elliptic curve cryptosystems*, Math. Comp., vol. 48 (1987), 203-209.

[53] N. Koblitz, *Hyperelliptic cryptosystems*, J. Cryptology, vol. 1 no. 3 (1989), 139-150.

[54] N. Koblitz, *CM-curves with good cryptographic properties*, Advances in Cryptology - CRYPTO 1991, LNCS vol. 576, Springer-Verlag (1992), 279-287.

[55] T. Lange, *Trace zero subvarieties of genus 2 curves for cryptosystem*, Ramanujan Math. Soc., vol. 19 no. 1 (2004) 15-33.

[56] R. Lidl, H. Niederreiter, *Introduction to finite fields and their applications*, Revised edition, Cambridge University Press (1994).

[57] M. Marlinspike, T. Perrin, *The Double Ratchet Algorithm*, November 2016, available at https://whispersystems.org/docs/specifications/doubleratchet/doubleratchet.pdf

[58] M. Marlinspike, T. Perrin, *The X3DH Key Agreement Protocol*, November 2016, available at https://whispersystems.org/docs/specifications/x3dh/x3dh.pdf

[59] V. S. Miller, *Use of elliptic curves in cryptography*, Advances in Cryptology - CRYPTO 1985, LNCS vol. 218, Springer-Verlag (1985), 417-426.

[60] V. S. Miller, *The Weil pairing, and its efficient calculation*, J. Cryptology, vol. 17 no. 4 (1994), 235-261.

[61] P. L. Montgomery, *Speeding the Pollard and elliptic curve methods of factorization*, Mathematics of Computation, vol. 48 no. 177 (1987), 243-264.

[62] N. Naumann, *Weil-Restriktion abelscher Varietäten*, Master's thesis (1999), University GHS Essen.

[63] K. Okeya, H. Kurumatani, K. Sakurai, *Elliptic Curves with the Montgomery-Form and Their Cryptographic Applications*, Public Key Cryptography, LNCS vol. 1751, Springer-Verlag (2000), 238-257.

[64] M. Rivain, *Fast and Regular Algorithms for Scalar Multiplication over Elliptic Curves*, IACR Cryptology ePrint Archive (2011).

[65] K. Rubin, A. Silverberg, *Supersingular abelian varieties in cryptology*, Advances in Cryptology: Proceedings of CRYPTO '02, LNCS vol. 2442, Springer-Verlag (2002), 336-353.

[66] K. Rubin, A. Silverberg, *Using abelian varieties to improve pairing-based cryptography*, Journal of Cryptology, vol. 22 no. 3 (2009), 330-364.

[67] I. Semaev, *Summation polynomials and the discrete logarithm problem on elliptic curves*, 2004 available at http://eprint.iacr.org/2004/031.

[68] V. Shoup, *Lower bounds for discrete logarithms and related problems*, Advances in Cryptology - EUROCRYPT 1997, LNCS vol 1233, Springer-Verlag (1997), 256-266.

[69] A. Silverberg, *Compression for trace zero subgroups of elliptic curves*, Trends in Mathematics, vol. 8 (2005), 93-100.

[70] S. Singh, *Studies on Index Calculus Techniques for the Discrete Log Problem*, PhD thesis (2016), available at https://www.iacr.org/phds/download.php?id=161

[71] H. Stichtenoth, *Algebraic Function Fields and Codes*, Graduate Texts in Mathematics, Springer-Verlag (1993).

[72] D. R. Stinson, *Cryptography. Theory and Practice*, Discrete Mathematics and its Applications, Third edition, Chapman & Hall/CRC (2006).

[73] L. C. Washington, *Elliptic curves: Number theory and cryptography*, Second edition, Discrete Mathematics and its Applications, Chapman & Hall/CRC (2008).

[74] WhatsApp Inc, *WhatsApp Encryption Overview*, Technical white paper, December, July 2017, available at https://www.whatsapp.com/security/WhatsApp-Security-Whitepaper.pdf

[75] A. Weimerskirch, *The application of the Mordell-Weil group to cryptographic systems*, Master's thesis (2011), Worcester Polytechnic Institute, available at http :// www.emsec.rub.de/media/crypto/attachments/files/2010/04/ms-weika.pdf.

Appendix A

Explicit formulas

A.1 Computation of a $(2,3,3)$-generalized summation polynomial

Coefficients of the $(2,3,3)$-generalized summation polynomial

$$S(\alpha_0,\alpha_1,\beta_0,\beta_1)(x,y) \in \mathbb{K}[\alpha_0,\alpha_1,\beta_0,\beta_1][x,y]$$

of the form

$$S_1(x) + yS_2(x) = (a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0) + y(b_3x^3 + b_2x^2 + b_1x + b_0),$$

computed with Algorithm 7 of Chapter 3.

$a_4 = -\alpha_1^3\beta_1\beta_0^2 - 3B\alpha_1^3\beta_1 + 2A\alpha_1^3\beta_0 + 2\alpha_1^2\alpha_0\beta_1^2\beta_0 + A\alpha_1^2\alpha_0\beta_1 - 6B\alpha_1^2\beta_1^2 + 3A\alpha_1^2\beta_1\beta_0 + A^2\alpha_1^2 - \alpha_1\alpha_0^2\beta_1^3 + 6\alpha_1\alpha_0^2\beta_0 + 3A\alpha_1\alpha_0\beta_1^2 + 3\alpha_1\alpha_0\beta_0^2 + 9B\alpha_1\alpha_0 - 3B\alpha_1\beta_1^3 + A\alpha_1\beta_1^2\beta_0 + 2A^2\alpha_1\beta_1 - 3\alpha_1\beta_0^3 + 9B\alpha_1\beta_0 - 3\alpha_0^3\beta_1 + 3\alpha_0^2\beta_1\beta_0 - 3A\alpha_0^2 + 2A\alpha_0\beta_1^3 + 6\alpha_0\beta_1\beta_0^2 + 9B\alpha_0\beta_1 - 6A\alpha_0\beta_0 + A^2\beta_1^2 + 9B\beta_1\beta_0 - 3A\beta_0^2$

$a_3 = 4B\alpha_1^3\beta_1^3 - 2A\alpha_1^3\beta_1^2\beta_0 + A^2\alpha_1^3\beta_1 - \alpha_1^3\beta_0^3 + 9B\alpha_1^3\beta_0 - 2A\alpha_1^2\alpha_0\beta_1^3 - \alpha_1^2\alpha_0\beta_1\beta_0^2 + 3B\alpha_1^2\alpha_0\beta_1 - 7A\alpha_1^2\alpha_0\beta_0 + A^2\alpha_1^2\beta_1^2 - 6B\alpha_1^2\beta_1\beta_0 + 3A\alpha_1^2\beta_0^2 + 6AB\alpha_1^2 - \alpha_1\alpha_0^2\beta_1^2\beta_0 + A\alpha_1\alpha_0^2\beta_1 - 6B\alpha_1\alpha_0\beta_1^2 + 12A\alpha_1\alpha_0\beta_1\beta_0 - 8A^2\alpha_1\alpha_0 + A^2\alpha_1\beta_1^3 + 3B\alpha_1\beta_1^2\beta_0 + A\alpha_1\beta_1\beta_0^2 - 6AB\alpha_1\beta_1 + 4A^2\alpha_1\beta_0 - \alpha_0^3\beta_1^3 - 3\alpha_0^3\beta_0 + 3A\alpha_0^2\beta_1^2 + 21\alpha_0^2\beta_0^2 - 18B\alpha_0^2 + 9B\alpha_0\beta_1^3 - 7A\alpha_0\beta_1^2\beta_0 + 4A^2\alpha_0\beta_1 - 3\alpha_0\beta_0^3 + 18B\alpha_0\beta_0 + 6AB\beta_1^2 - 8A^2\beta_1\beta_0 - 18B\beta_0^2 + 4A^3 + 27B^2$

$a_2 = -A^2\alpha_1^3\beta_1^3 - 2A\alpha_1^3\beta_1\beta_0^2 - 6AB\alpha_1^3\beta_1 + A^2\alpha_1^3\beta_0 - 2A\alpha_1^2\alpha_0\beta_1^2\beta_0 + 5A^2\alpha_1^2\alpha_0\beta_1 - 3\alpha_1^2\alpha_0\beta_0^3 - 9B\alpha_1^2\alpha_0\beta_0 + 6AB\alpha_1^2\beta_1^2 + 9B\alpha_1^2\beta_0^2 + (2A^3 + 27B^2)\alpha_1^2 - 2A\alpha_1\alpha_0^2\beta_1^3 - 3\alpha_1\alpha_0^2\beta_1\beta_0^2 + 9B\alpha_1\alpha_0^2\beta_1 + 9A\alpha_1\alpha_0^2\beta_0 + 36B\alpha_1\alpha_0\beta_1\beta_0 - 12A\alpha_1\alpha_0\beta_0^2 - 18AB\alpha_1\alpha_0 - 6AB\alpha_1\beta_1^3 + 5A^2\alpha_1\beta_1^2\beta_0 + 9B\alpha_1\beta_1\beta_0^2 + (4A^3 - 27B^2)\alpha_1\beta_1 - 3A\alpha_1\beta_0^3 + 36AB\alpha_1\beta_0 - 3\alpha_0^3\beta_1^2\beta_0 - 3A\alpha_0^3\beta_1 + 9B\alpha_0^2\beta_1^2 - 12A\alpha_0^2\beta_1\beta_0 + 6A^2\alpha_0^2 + A^2\alpha_0\beta_1^3 - 9B\alpha_0\beta_1^2\beta_0 + 9A\alpha_0\beta_1\beta_0^2 + 36AB\alpha_0\beta_1 - 24A^2\alpha_0\beta_0 + (2A^3 + 27B^2)\beta_1^2 - 18AB\beta_1\beta_0 + 6A^2\beta_0^2$

$a_1 = -A^2\alpha_1^3\beta_1^2\beta_0 - 4B\alpha_1^3\beta_1\beta_0^2 + (A^3 - 12B^2)\alpha_1^3\beta_1 + 8AB\alpha_1^3\beta_0 - A^2\alpha_1^2\alpha_0\beta_1^3 - 4B\alpha_1^2\alpha_0\beta_1^2\beta_0 + 16AB\alpha_1^2\alpha_0\beta_1 - 3A^2\alpha_1^2\alpha_0\beta_0 + (-3A^3 + 12B^2)\alpha_1^2\beta_1^2 - 24AB\alpha_1^2\beta_1\beta_0 + 3A^2\alpha_1^2\beta_0^2 - 2A^2B\alpha_1^2 - 4B\alpha_1\alpha_0^2\beta_1^3 - 3A^2\alpha_1\alpha_0^2\beta_1 - 3\alpha_1\alpha_0^2\beta_0^3 + 15B\alpha_1\alpha_0^2\beta_0 - 24AB\alpha_1\alpha_0\beta_1^2 + 12A^2\alpha_1\alpha_0\beta_1\beta_0 - 6B\alpha_1\alpha_0\beta_0^2 - 18B^2\alpha_1\alpha_0 + (A^3 - 12B^2)\alpha_1\beta_1^3 + 16AB\alpha_1\beta_1^2\beta_0 - 3A^2\alpha_1\beta_1\beta_0^2 + 14A^2B\alpha_1\beta_1 - 3B\alpha_1\beta_0^3 + 63B^2\alpha_1\beta_0 - 3\alpha_0^3\beta_1\beta_0^2 - 3B\alpha_0^3\beta_1 - 3A\alpha_0^3\beta_0 + 3A^2\alpha_0^2\beta_1^2 - 6B\alpha_0^2\beta_1\beta_0 + 9A\alpha_0^2\beta_0^2 + 6AB\alpha_0^2 + 8AB\alpha_0\beta_1^3 - 3A^2\alpha_0\beta_1^2\beta_0 + 15B\alpha_0\beta_1\beta_0^2 + 63B^2\alpha_0\beta_1 - 3A\alpha_0\beta_0^3 - 42AB\alpha_0\beta_0 - 2A^2B\beta_1^2 -$

$18B^2\beta_1\beta_0 + 6AB\beta_0^2 + 4A^4 + 27AB^2$

$a_0 = 2A^2B\alpha_1^3\beta_1 - A^3\alpha_1^3\beta_0 - A^2\alpha_1^2\alpha_0\beta_1^2\beta_0 - 4B\alpha_1^2\alpha_0\beta_1\beta_0^2 + 12B^2\alpha_1^2\alpha_0\beta_1 - 4AB\alpha_1^2\alpha_0\beta_0 - 5A^2B\alpha_1^2\beta_1^2 + (A^3 - 24B^2)\alpha_1^2\beta_1\beta_0 + 6AB\alpha_1^2\beta_0^2 + (A^4 + 6AB^2)\alpha_1^2 - 4B\alpha_1\alpha_0^2\beta_1^2\beta_0 + 2A\alpha_1\alpha_0^2\beta_1\beta_0^2 - 2AB\alpha_1\alpha_0^2\beta_1 - A^2\alpha_1\alpha_0^2\beta_0 + (A^3 - 24B^2)\alpha_1\alpha_0\beta_1^2 + 24AB\alpha_1\alpha_0\beta_1\beta_0 - 3A^2\alpha_1\alpha_0\beta_0^2 + A^2B\alpha_1\alpha_0 + 2A^2B\alpha_1\beta_1^3 + 12B^2\alpha_1\beta_1^2\beta_0 - 2AB\alpha_1\beta_1\beta_0^2 + (-2A^4 - 6AB^2)\alpha_1\beta_1 - 5A^2B\alpha_1\beta_0 - \alpha_0^3\beta_0^3 - 3B\alpha_0^3\beta_0 + 6AB\alpha_0^2\beta_1^2 - 3A^2\alpha_0^2\beta_1\beta_0 + 3B\alpha_0^2\beta_0^2 + (A^3 + 9B^2)\alpha_0^2 - A^3\alpha_0\beta_1^3 - 4AB\alpha_0\beta_1^2\beta_0 - A^2\alpha_0\beta_1\beta_0^2 - 5A^2B\alpha_0\beta_1 - 3B\alpha_0\beta_0^3 + (2A^3 - 9B^2)\alpha_0\beta_0 + (A^4 + 6AB^2)\beta_1^2 + A^2B\beta_1\beta_0 + (A^3 + 9B^2)\beta_0^2 + 4A^3B + 27B^3$

$b_3 = \alpha_1^3\beta_0^2 - B\alpha_1^3 - 2\alpha_1^2\alpha_0\beta_1\beta_0 + A\alpha_1^2\alpha_0 + \alpha_1^2\beta_1\beta_0^2 - 3B\alpha_1^2\beta_1 + A\alpha_1^2\beta_0 + \alpha_1\alpha_0^2\beta_1^2 - 2\alpha_1\alpha_0\beta_1^2\beta_0 + 2A\alpha_1\alpha_0\beta_1 - 3B\alpha_1\beta_1^2 + 2A\alpha_1\beta_1\beta_0 + \alpha_0^3 + \alpha_0^2\beta_1^3 + 3\alpha_0^2\beta_0 + A\alpha_0\beta_1^2 + 3\alpha_0\beta_0^2 - B\beta_1^3 + A\beta_1^2\beta_0 + \beta_0^3$

$b_2 = A^2\alpha_1^3 + 3\alpha_1^2\alpha_0\beta_0^2 + 9B\alpha_1^2\alpha_0 + 3A^2\alpha_1^2\beta_1 + 3\alpha_1^2\beta_0^3 + 9B\alpha_1^2\beta_0 - 6\alpha_1\alpha_0^2\beta_1\beta_0 - 3A\alpha_1\alpha_0^2 - 6\alpha_1\alpha_0\beta_1\beta_0^2 + 18B\alpha_1\alpha_0\beta_1 - 6A\alpha_1\alpha_0\beta_0 + 3A^2\alpha_1\beta_1^2 + 18B\alpha_1\beta_1\beta_0 - 3A\alpha_1\beta_0^2 + 3\alpha_0^3\beta_1^2 + 3\alpha_0^2\beta_1^2\beta_0 - 3A\alpha_0^2\beta_1 + 9B\alpha_0\beta_1^2 - 6A\alpha_0\beta_1\beta_0 + A^2\beta_1^3 + 9B\beta_1^2\beta_0 - 3A\beta_1\beta_0^2$

$b_1 = -A^2\alpha_1^3\beta_1^2 - 2A\alpha_1^3\beta_0^2 + 2AB\alpha_1^3 - 12B\alpha_1^2\alpha_0\beta_1^2 + 4A\alpha_1^2\alpha_0\beta_1\beta_0 - 3A^2\alpha_1^2\alpha_0 - A^2\alpha_1^2\beta_1^3 - 12B\alpha_1^2\beta_1^2\beta_0 + 4A\alpha_1^2\beta_1\beta_0^2 + A^2\alpha_1^2\beta_0 + 4A\alpha_1\alpha_0^2\beta_1^2 - 3\alpha_1\alpha_0^2\beta_0^2 - 9B\alpha_1\alpha_0^2 + 4A\alpha_1\alpha_0\beta_1^2\beta_0 + 2A^2\alpha_1\alpha_0\beta_1 + 6\alpha_1\alpha_0\beta_0^3 + 18B\alpha_1\alpha_0\beta_0 + 2A^2\alpha_1\beta_1\beta_0 + 9B\alpha_1\beta_0^2 + (4A^3 + 27B^2)\alpha_1 + 6\alpha_0^3\beta_1\beta_0 + A\alpha_0^3 - 2A\alpha_0^2\beta_1^3 - 3\alpha_0^2\beta_1\beta_0^2 + 9B\alpha_0^2\beta_1 - 9A\alpha_0^2\beta_0 + A^2\alpha_0\beta_1^2 + 18B\alpha_0\beta_1\beta_0 - 9A\alpha_0\beta_0^2 + 2AB\beta_1^3 - 3A^2\beta_1^2\beta_0 - 9B\beta_1\beta_0^2 + (4A^3 + 27B^2)\beta_1 + A\beta_0^3$

$b_0 = -2A^2\alpha_1^3\beta_1\beta_0 - 8B\alpha_1^3\beta_0^2 + (A^3 + 8B^2)\alpha_1^3 + A^2\alpha_1^2\alpha_0\beta_1^2 - 8B\alpha_1^2\alpha_0\beta_1\beta_0 + 6A\alpha_1^2\alpha_0\beta_0^2 - 2AB\alpha_1^2\alpha_0 + A^2\alpha_1^2\beta_1^2\beta_0 + 4B\alpha_1^2\beta_1\beta_0^2 + (-A^3 - 12B^2)\alpha_1^2\beta_1 + 4AB\alpha_1^2\beta_0 + 4B\alpha_1\alpha_0^2\beta_1^2 + A^2\alpha_1\alpha_0^2 - 2A^2\alpha_1\alpha_0\beta_1^3 - 8B\alpha_1\alpha_0\beta_1^2\beta_0 + 8AB\alpha_1\alpha_0\beta_1 - 6A^2\alpha_1\alpha_0\beta_0 + (-A^3 - 12B^2)\alpha_1\beta_1^2 + 8AB\alpha_1\beta_1\beta_0 - 3A^2\alpha_1\beta_0^2 + 3\alpha_0^3\beta_0^2 + B\alpha_0^3 - 8B\alpha_0^2\beta_1^3 + 6A\alpha_0^2\beta_1^2\beta_0 - 3A^2\alpha_0^2\beta_1 + 3\alpha_0^2\beta_0^3 - 15B\alpha_0^2\beta_0 + 4AB\alpha_0\beta_1^2 - 6A^2\alpha_0\beta_1\beta_0 - 15B\alpha_0\beta_0^2 + (4A^3 + 27B^2)\alpha_0 + (A^3 + 8B^2)\beta_1^3 - 2AB\beta_1^2\beta_0 + A^2\beta_1\beta_0^2 + B\beta_0^3 + (4A^3 + 27B^2)\beta_0$

A.2 Doubling and tripling formulas in T_3

A.2.1 Doubling formulas

Coefficients of the doubling polynomial

$$h_2(\alpha_0, \alpha_1)(x, y) \in \mathbb{F}_q[\alpha_0, \alpha_1][x, y]$$

of the form

$$h_2(\alpha_0, \alpha_1)(x, y) = c(\alpha_0, \alpha_1)y - (u_0(\alpha_0, \alpha_1) + u_1(\alpha_0, \alpha_1)x),$$

computed with Procedure 1 of Chapter 4.

$c = 8B\alpha_1^3 - 8A\alpha_1^2\alpha_0 - 8\alpha_0^3$

$u_0 = -A^2\alpha_1^4 - 8B\alpha_1^3\alpha_0 + 2A\alpha_1^2\alpha_0^2 + 6AB\alpha_1^2 - 8A^2\alpha_1\alpha_0 - \alpha_0^4 - 18B\alpha_0^2 + 4A^3 + 27B^2$

$u_1 = 4B\alpha_1^4 - 4A\alpha_1^3\alpha_0 + 4A^2\alpha_1^2 - 4\alpha_1\alpha_0^3 + 36B\alpha_1\alpha_0 - 12A\alpha_0^2$

A.2.2 Tripling formulas

Coefficients of the tripling polynomial

$$h_3(\alpha_0, \alpha_1)(x, y) \in \mathbb{F}_q[\alpha_0, \alpha_1][x, y]$$

of the form

$$h_3(\alpha_0, \alpha_1)(x, y) = d(\alpha_0, \alpha_1)y - (v_0(\alpha_0, \alpha_1) + v_1(\alpha_0, \alpha_1)x),$$

computed with Procedure 2 of Chapter 4.

$d = A^4\alpha_1^8 + 24A^2B\alpha_1^7\alpha_0 + (-12A^3 + 144B^2)\alpha_1^6\alpha_0^2 + (-24A^3B - 144B^3)\alpha_1^6 - 144AB\alpha_1^5\alpha_0^3 + (32A^4 + 144AB^2)\alpha_1^5\alpha_0 + 30A^2\alpha_1^4\alpha_0^4 + 120A^2B\alpha_1^4\alpha_0^2 + (-8A^5 - 54A^2B^2)\alpha_1^4 - 72B\alpha_1^3\alpha_0^5 + 720B^2\alpha_1^3\alpha_0^3 + (-96A^3B - 648B^3)\alpha_1^3\alpha_0 + 36A\alpha_1^2\alpha_0^6 - 360AB\alpha_1^2\alpha_0^4 + (48A^4 + 324AB^2)\alpha_1^2\alpha_0^2 + 96A^2\alpha_1\alpha_0^5 + 9\alpha_0^8 + 72B\alpha_0^6 + (24A^3 + 162B^2)\alpha_0^4 - 16/3A^6 - 72A^3B^2 - 243B^4$

$v_0 = (-8/3A^3B - 64/3B^3)\alpha_1^9 + (3A^4 + 32AB^2)\alpha_1^8\alpha_0 - 16A^2B\alpha_1^7\alpha_0^2 - 8A^2B^2\alpha_1^7 + (12A^3 + 16B^2)\alpha_1^6\alpha_0^3 + (8A^3B - 144B^3)\alpha_1^6\alpha_0 + 8AB\alpha_1^5\alpha_0^4 + 288AB^2\alpha_1^5\alpha_0^2 + (32A^4B + 216AB^3)\alpha_1^5 + 10A^2\alpha_1^4\alpha_0^5 - 200A^2B\alpha_1^4\alpha_0^3 + (-24A^5 - 162A^2B^2)\alpha_1^4\alpha_0 + 32B\alpha_1^3\alpha_0^6 + (64A^3 + 72B^2)\alpha_1^3\alpha_0^4 + (192A^3B + 1296B^3)\alpha_1^3\alpha_0^2 + (96A^3B^2 + 648B^4)\alpha_1^3 - 4A\alpha_1^2\alpha_0^7 - 72AB\alpha_1^2\alpha_0^5 + (-176A^4 - 1188AB^2)\alpha_1^2\alpha_0^3 + (-192A^4B - 1296AB^3)\alpha_1^2\alpha_0 + 64A^2\alpha_1\alpha_0^6 + (128A^5 + 864A^2B^2)\alpha_1\alpha_0^2 + 1/3\alpha_0^9 + 72B\alpha_0^7 + (-120A^3 - 810B^2)\alpha_0^5 + (192A^3B + 1296B^3)\alpha_0^3 + (-16A^6 - 216A^3B^2 - 729B^4)\alpha_0$

$v_1 = 1/3A^4\alpha_1^9 + 8A^2B\alpha_1^8\alpha_0 + (-4A^3 + 48B^2)\alpha_1^7\alpha_0^2 + (16A^3B + 144B^3)\alpha_1^7 - 48AB\alpha_1^6\alpha_0^3 + (-16A^4 - 240AB^2)\alpha_1^6\alpha_0 + 10A^2\alpha_1^5\alpha_0^4 + 192A^2B\alpha_1^5\alpha_0^2 + (8A^5 + 54A^2B^2)\alpha_1^5 - 24B\alpha_1^4\alpha_0^5 + (-112A^3 + 144B^2)\alpha_1^4\alpha_0^3 + (96A^3B + 648B^3)\alpha_1^4\alpha_0 + 12A\alpha_1^3\alpha_0^6 - 240AB\alpha_1^3\alpha_0^4 + (-48A^4 - 324AB^2)\alpha_1^3\alpha_0^2 + (-32A^4B - 216AB^3)\alpha_1^3 - 48A^2\alpha_1^2\alpha_0^5 + (64A^5 + 432A^2B^2)\alpha_1^2\alpha_0 + 3\alpha_1\alpha_0^8 - 288B\alpha_1\alpha_0^6 + (-24A^3 - 162B^2)\alpha_1\alpha_0^4 + (288A^3B + 1944B^3)\alpha_1\alpha_0^2 + (-16A^6 - 216A^3B^2 - 729B^4)\alpha_1 + 48A\alpha_0^7 + (-64A^4 - 432AB^2)\alpha_0^3$

A.3 Polynomial systems for the relation search step of index calculus in T_3

(PS1). Polynomial system for the relation search step of index calculus in T_3, for the factor base $\mathcal{F}_1$, of the form (5.10) of Section 5.1.3.

$(A\mu x_1y_2 + A\mu x_2y_1 + Ax_0y_0 + By_0 + 3\mu^2x_0x_2^2y_2 + 3\mu^2x_1^2x_2y_2 + 3\mu^2x_1x_2^2y_1 + \mu^2x_2^3y_0 + 3\mu x_0^2x_1y_2 + 3\mu x_0^2x_2y_1 + 3\mu x_0x_1^2y_1 + 6\mu x_0x_1x_2y_0 + \mu x_1^3y_0 + x_0^3y_0)s_1^3 + (-3Ax_0 - 3B - 3\mu^2x_2^3 - 18\mu x_0x_1x_2 - 3\mu x_1^3 - 3x_0^3)s_1^2s_2 + (A^3 + 12A^2\mu x_1x_2 + 6A^2x_0^2 + 6ABx_0 - 12A\mu^2x_0x_2^3 - 18A\mu^2x_1^2x_2^2 - 36A\mu x_0^2x_1x_2 - 12A\mu x_0x_1^3 - 3Ax_0^4 + 9B^2 - 18B\mu^2x_2^3 - 108B\mu x_0x_1x_2 - 18B\mu x_1^3 - 18Bx_0^3)s_1^2 + 3y_0s_1s_2^2 + (-12A\mu x_1y_2 - 12A\mu x_2y_1 - 12Ax_0y_0 - 18By_0)s_1s_2 + (4A^3y_0 + 27B^2y_0)s_1 - s_2^3 + (15Ax_0 + 9B + 27\mu^2x_2^3 + 162\mu x_0x_1x_2 + 27\mu x_1^3 + 27x_0^3)s_2^2 + (-72A^2\mu x_1x_2 - 36A^2x_0^2 - 54ABx_0 - 27B^2 + 54B\mu^2x_2^3 + 324B\mu x_0x_1x_2 + 54B\mu x_1^3 + 54Bx_0^3)s_2 + 4A^4x_0 + 4A^3B + 4A^3\mu^2x_2^3 + 24A^3\mu x_0x_1x_2 + 4A^3\mu x_1^3 + 4A^3x_0^3 + 27AB^2x_0 + 27B^3 + 27B^2\mu^2x_2^3 + 162B^2\mu x_0x_1x_2 + 27B^2\mu x_1^3 + 27B^2x_0^3 = 0$

$(A\mu x_2y_2 + Ax_0y_1 + Ax_1y_0 + By_1 + 3\mu^2x_1x_2^2y_2 + \mu^2x_2^3y_1 + 3\mu x_0^2x_2y_2 + 3\mu x_0x_1^2y_2 + 6\mu x_0x_1x_2y_1 + 3\mu x_0x_2^2y_0 + \mu x_1^3y_1 + 3\mu x_1^2x_2y_0 + x_0^3y_1 + 3x_0^2x_1y_0)s_1^3 + (-3Ax_1 - 9\mu x_0x_2^2 - 9\mu x_1^2x_2 - 9x_0^2x_1)s_1^2s_2 +$

$(6A^2\mu x_2^2+12A^2x_0x_1+6ABx_1-12A\mu^2x_1x_2^3-18A\mu x_0^2x_2^2-36A\mu x_0x_1^2x_2-3A\mu x_1^4-12Ax_0^3x_1-54B\mu x_0x_2^2-54B\mu x_1^2x_2-54Bx_0^2x_1)s_1^2+3y_1s_1s_2^2+(-12A\mu x_2y_2-12Ax_0y_1-12Ax_1y_0-18By_1)s_1s_2+(4A^3y_1+27B^2y_1)s_1+(15Ax_1+81\mu x_0x_2^2+81\mu x_1^2x_2+81x_0^2x_1)s_2^2+(-36A^2\mu x_2^2-72A^2x_0x_1-54ABx_1+162B\mu x_0x_2^2+162B\mu x_1^2x_2+162Bx_0^2x_1)s_2+4A^4x_1+12A^3\mu x_0x_2^2+12A^3\mu x_1^2x_2+12A^3x_0^2x_1+27AB^2x_1+81B^2\mu x_0x_2^2+81B^2\mu x_1^2x_2+81B^2x_0^2x_1=0$

$(Ax_0y_2+Ax_1y_1+Ax_2y_0+By_2+\mu^2x_2^3y_2+6\mu x_0x_1x_2y_2+3\mu x_0x_2^2y_1+\mu x_1^3y_2+3\mu x_1^2x_2y_1+3\mu x_1x_2^2y_0+x_0^3y_2+3x_0^2x_1y_1+3x_0^2x_2y_0+3x_0x_1^2y_0)s_1^3+(-3Ax_2-9\mu x_1x_2^2-9x_0^2x_2-9x_0x_1^2)s_1^2s_2+(12A^2x_0x_2+6A^2x_1^2+6ABx_2-3A\mu^2x_2^4-36A\mu x_0x_1x_2^2-12A\mu x_1^3x_2-12Ax_0^3x_2-18Ax_0^2x_1^2-54B\mu x_1x_2^2-54Bx_0^2x_2-54Bx_0x_1^2)s_1^2+3y_2s_1s_2^2+(-12Ax_0y_2-12Ax_1y_1-12Ax_2y_0-18By_2)s_1s_2+(4A^3y_2+27B^2y_2)s_1+(15Ax_2+81\mu x_1x_2^2+81x_0^2x_2+81x_0x_1^2)s_2^2+(-72A^2x_0x_2-36A^2x_1^2-54ABx_2+162B\mu x_1x_2^2+162Bx_0^2x_2+162Bx_0x_1^2)s_2+4A^4x_2+12A^3\mu x_1x_2^2+12A^3x_0^2x_2+12A^3x_0x_1^2+27AB^2x_2+81B^2\mu x_1x_2^2+81B^2x_0^2x_2+81B^2x_0x_1^2=0$

(PS2). Polynomial system for the relation search step of index calculus in T_3, for the factor base $\mathcal{F}_2$, of the form (5.10) of Section 5.1.3.

$(A^3y_0+A^2\mu^2x_2^2y_2+2A^2\mu x_0x_1y_2+2A^2\mu x_0x_2y_1+A^2\mu x_1^2y_1+2A^2\mu x_1x_2y_0+A^2x_0^2y_0+2AB\mu x_1y_2+2AB\mu x_2y_1+2ABx_0y_0+8B^2y_0-3B\mu^2x_0x_2^2y_2-3B\mu^2x_1^2x_2y_2-3B\mu^2x_1x_2^2y_1-B\mu^2x_2^3y_0-3B\mu x_0^2x_1y_2-3B\mu x_0^2x_2y_1-3B\mu x_0x_1^2y_1-6B\mu x_0x_1x_2y_0-B\mu x_1^3y_0-Bx_0^3y_0)s_1^3+(A^3x_0+2A^2B+A^2\mu^2x_2^3+6A^2\mu x_0x_1x_2+A^2\mu x_1^3+A^2x_0^3-12AB\mu x_1x_2-6ABx_0^2-12B^2x_0-12B\mu^2x_0x_2^3-18B\mu^2x_1^2x_2^2-36B\mu x_0^2x_1x_2-12B\mu x_0x_1^3-3Bx_0^4)s_1^2s_2+(A^4+4A^3\mu x_1x_2+2A^3x_0^2-2A^2Bx_0+4A^2\mu^2x_0x_2^3+6A^2\mu^2x_1^2x_2^2+12A^2\mu x_0^2x_1x_2+4A^2\mu x_0x_1^3+A^2x_0^4+6AB^2+6AB\mu^2x_2^3+36AB\mu x_0x_1x_2+6AB\mu x_1^3+6ABx_0^3+54B^2\mu x_1x_2+27B^2x_0^2)s_1^2+(-A^2\mu x_1y_2-A^2\mu x_2y_1-A^2x_0y_0)s_1s_2^2+(-4A^3y_0-6AB\mu x_1y_2-6AB\mu x_2y_1-6ABx_0y_0-36B^2y_0)s_1s_2+(4A^3\mu x_1y_2+4A^3\mu x_2y_1+4A^3x_0y_0+27B^2\mu x_1y_2+27B^2\mu x_2y_1+27B^2x_0y_0)s_1+(-2A^2\mu x_1x_2-A^2x_0^2+4B\mu^2x_2^3+24B\mu x_0x_1x_2+4B\mu x_1^3+4Bx_0^3)s_2^3+(-5A^3x_0-9A^2B-A^2\mu^2x_2^3-6A^2\mu x_0x_1x_2-A^2\mu x_1^3-A^2x_0^3+36AB\mu x_1x_2+18ABx_0^2+36B^2x_0)s_2^2+(-4A^4+18A^2Bx_0-18AB^2-18AB\mu^2x_2^3-108AB\mu x_0x_1x_2-18AB\mu x_1^3-18ABx_0^3-162B^2\mu x_1x_2-81B^2x_0^2)s_2+4A^4x_0+4A^3B+4A^3\mu^2x_2^3+24A^3\mu x_0x_1x_2+4A^3\mu x_1^3+4A^3x_0^3+27AB^2x_0+27B^3+27B^2\mu^2x_2^3+162B^2\mu x_0x_1x_2+27B^2\mu x_1^3+27B^2x_0^3=0$

$(A^3y_1+2A^2\mu x_0x_2y_2+A^2\mu x_1^2y_2+2A^2\mu x_1x_2y_1+A^2\mu x_2^2y_0+A^2x_0^2y_1+2A^2x_0x_1y_0+2AB\mu x_2y_2+2ABx_0y_1+2ABx_1y_0+8B^2y_1-3B\mu^2x_1x_2^2y_2-B\mu^2x_2^3y_1-3B\mu x_0^2x_2y_2-3B\mu x_0x_1^2y_2-6B\mu x_0x_1x_2y_1-3B\mu x_0x_2^2y_0-B\mu x_1^3y_1-3B\mu x_1^2x_2y_0-Bx_0^3y_1-3Bx_0^2x_1y_0)s_1^3+(A^3x_1+3A^2\mu x_0x_2^2+3A^2\mu x_1^2x_2+3A^2x_0^2x_1-6AB\mu x_2^2-12ABx_0x_1-12B^2x_1-12B\mu^2x_1x_2^3-18B\mu x_0^2x_2^2-36B\mu x_0x_1^2x_2-3B\mu x_1^4-12Bx_0^3x_1)s_1^2s_2+(2A^3\mu x_2^2+4A^3x_0x_1-2A^2Bx_1+4A^2\mu^2x_1x_2^3+6A^2\mu x_0^2x_2^2+12A^2\mu x_0x_1^2x_2+A^2\mu x_1^4+4A^2x_0^3x_1+18AB\mu x_0x_2^2+18AB\mu x_1^2x_2+18ABx_0^2x_1+27B^2\mu x_2^2+54B^2x_0x_1)s_1^2+(-A^2\mu x_2y_2-A^2x_0y_1-A^2x_1y_0)s_1s_2^2+(-4A^3y_1-6AB\mu x_2y_2-6ABx_0y_1-6ABx_1y_0-36B^2y_1)s_1s_2+(4A^3\mu x_2y_2+4A^3x_0y_1+4A^3x_1y_0+27B^2\mu x_2y_2+27B^2x_0y_1+27B^2x_1y_0)s_1+(-A^2\mu x_2^2-2A^2x_0x_1+12B\mu x_0x_2^2+12B\mu x_1^2x_2+12Bx_0^2x_1)s_2^3+(-5A^3x_1-3A^2\mu x_0x2^2-3A^2\mu x_1^2x_2-3A^2x_0^2x_1+18AB\mu x_2^2+36ABx_0x_1+36B^2x_1)s_2^2+(18A^2Bx_1-54AB\mu x_0x_2^2-54AB\mu x_1^2x_2-54ABx_0^2x_1-81B^2\mu x_2^2-162B^2x_0x_1)s_2+4A^4x_1+12A^3\mu x_0x_2^2+12A^3\mu x_1^2x_2+12A^3x_0^2x_1+27AB^2x_1+81B^2\mu x_0x_2^2+81B^2\mu x_1^2x_2+81B^2x_0^2x_1=0$

$(A^3y_2+2A^2\mu x_1x_2y_2+A^2\mu x_2^2y_1+A^2x_0^2y_2+2A^2x_0x_1y_1+2A^2x_0x_2y_0+A^2x_1^2y_0+2ABx_0y_2+2ABx_1y_1+2ABx_2y_0+8B^2y_2-B\mu^2x_2^3y_2-6B\mu x_0x_1x_2y_2-3B\mu x_0x_2^2y_1-B\mu x_1^3y_2-3B\mu x_1^2x_2y_1-3B\mu x_1x_2^2y_0-Bx_0^3y_2-3Bx_0^2x_1y_1-3Bx_0^2x_2y_0-3Bx_0x_1^2y_0)s_1^3+(A^3x_2+3A^2\mu x_1x_2^2+3A^2x_0^2x_2+$

$3A^2x_0x_1^2-12ABx_0x_2-6ABx_1^2-12B^2x_2-3B\mu^2x_2^4-36B\mu x_0x_1x_2^2-12B\mu x1^3x_2-12Bx_0^3x_2-18Bx_0^2x_1^2)s_1^2s_2+(4A^3x_0x_2+2A^3x_1^2-2A^2Bx_2+A^2\mu^2x_2^4+12A^2\mu x_0x_1x_2^2+4A^2\mu x_1^3x_2+4A^2x_0^3x_2+6A^2x_0^2x_1^2+18AB\mu x_1x_2^2+18ABx_0^2x_2+18ABx_0x_1^2+54B^2x_0x_2+27B^2x_1^2)s_1^2+(-A^2x_0y_2-A^2x_1y_1-A^2x_2y_0)s_1s_2^2+(-4A^3y_2-6ABx_0y_2-6ABx_1y_1-6ABx_2y_0-36B^2y_2)s_1s_2+(4A^3x_0y_2+4A^3x_1y_1+4A^3x_2y_0+27B^2x_0y_2+27B^2x_1y_1+27B^2x_2y_0)s_1+(-2A^2x_0x_2-A^2x_1^2+12B\mu x_1x_2^2+12Bx_0^2x_2+12Bx_0x_1^2)s_2^3+(-5A^3x_2-3A^2\mu x_1x_2^2-3A^2x_0^2x_2-3A^2x_0x_1^2+36ABx_0x_2+18ABx_1^2+36B^2x_2)s_2^2+(18A^2Bx_2-54AB\mu x_1x_2^2-54ABx_0^2x_2-54ABx_0x_1^2-162B^2x_0x_2-81B^2x_1^2)s_2+4A^4x_2+12A^3\mu x_1x_2^2+12A^3x_0^2x_2+12A^3x_0x_1^2+27AB^2x_2+81B^2\mu x_1x_2^2+81B^2x_0^2x_2+81B^2x_0x_1^2=0$