Using Symmetries in the Index Calculus for Elliptic Curves Discrete Logarithm

Jean-Charles Faugère, Pierrick Gaudry, Louise Huot, Guénaël Renault

USING SYMMETRIES IN THE INDEX CALCULUS FOR ELLIPTIC CURVES DISCRETE LOGARITHM

JEAN-CHARLES FAUGÈRE[1], PIERRICK GAUDRY[2], LOUISE HUOT[1], AND GUÉNAËL RENAULT[1]

ABSTRACT. In 2004, an algorithm is introduced to solve the DLP for elliptic curves defined over a non prime finite field $\mathbb{F}_{q^n}$. One of the main steps of this algorithm requires decomposing points of the curve $E(\mathbb{F}_{q^n})$ with respect to a factor base, this problem is denoted PDP. In this paper, we will apply this algorithm to the case of Edwards curves, the well known family of elliptic curves that allow faster arithmetic as shown by Bernstein and Lange. More precisely, we show how to take advantage of some symmetries of twisted Edwards and twisted Jacobi intersections curves to gain an exponential factor $2^{3(n-1)}$ to solve the corresponding PDP. Practical experiments supporting the theoretical result are also given. For instance, the complexity of solving the ECDLP for twisted Edwards curves defined over $\mathbb{F}_{q^5}$, with $q \approx 2^{64}$, is supposed to be 2^{160} operations in $E(\mathbb{F}_{q^5})$ using generic algorithms compared to 2^{127} operations (multiplication of two 32 bits words) with our method. For these parameters the PDP is untractable with the original algorithm.

The main tool to achieve these results relies on the use of the symmetries during the polynomial system solving step. Also, we use a recent work on a new strategy for the change of ordering of Gröbner basis which provides a better heuristic complexity of the total solving process.

1. INTRODUCTION

1.1. **Context.** One of the main number theoretic problems is, given a cyclic group $\mathbb{G}$ of generator g and an element h of this group, to find an integer x such that $h = g^x$. This problem is called the discrete logarithm problem and it is denoted DLP. To solve the DLP, there exist algorithms which do not consider the structure and the representation of the group where the DLP is defined. They are called generic algorithms and Shoup shows in [46] that they are exponential in general. The Pollard rho method [43] is optimal among generic algorithms, up to a constant factor, with a running time in $O(\sqrt{\#\mathbb{G}})$ group operations. Nevertheless for some groups, the DLP is easier to solve. For instance if $\mathbb{G}$ is a multiplicative group formed by the invertible elements of a finite field, the index calculus method [1] solves the DLP in sub-exponential time.

Key words and phrases. ECDLP, Edwards curves, elliptic curves, decomposition attack, Gröbner Basis with symmetries, index calculus, Jacobi intersections curves.

1 : POLSYS Project INRIA Paris-Rocquencourt ; UPMC, Univ Paris 06, LIP6 ; CNRS, UMR 7606, LIP6 ; UFR Ingénierie 919, LIP6 .

2 : CARAMEL Project INRIA Grand-Est ; LORIA Campus Scientifique .

A major application of the DLP is to design cryptographic protocols which security depends on the difficulty of solving the DLP. A cryptosystem has to be secure and fast. Hence we have to consider groups with an efficient arithmetic, a compact representation of their elements and where the DLP is intractable. To this end, in 1985 V. Miller [39] and N. Koblitz [36] introduced elliptic curve cryptography based on the DLP in the group formed by rational points of an elliptic curve defined over a finite field. This particular problem is denoted ECDLP. More recently, some curve representations such as twisted Edwards [4, 5, 17] and twisted Jacobi intersections [9, 27] have been widely studied by the cryptology community for their efficient arithmetic. A few years after the introduction of elliptic curve cryptography, it has been proposed to use the divisor class group of a hyperelliptic curve over a finite field [37], in this case we note the discrete logarithm problem HCDLP.

To estimate the security of cryptosystems based on the HCDLP, the resolution of this problem has been extensively studied in recent years and index calculus methods [2, 11, 18, 19, 32] have been developed for various classes of high genus curves. Using the double large prime variation of Gaudry, Thomé, Thériault and Diem [31], if the size of the finite field is sufficiently large and for curves having genus greater than three, index calculus method is then faster than Pollard rho method. In the particular case of non-hyperelliptic curves of genus 3, Diem and Thomé got a further improvement of the index calculus [13, 16]. These methods do not apply to curves having genus 1 or 2.

If the curve is defined over a non prime finite field, by applying a Weil restriction, the discrete logarithm problem can be seen in an abelian variety over the smaller field. In [29], Gaudry proposed an index calculus attack suited to this context. Later on, Diem [15, 14] obtained rigorous proofs that for some particular families of curves the discrete logarithm problem can be solved in subexponential time.

Let us recall the principle of the algorithm in the case of interest in this paper, namely the ECDLP in an elliptic curve E defined over a non prime finite field $\mathbb{F}_{q^n}$ with $n > 1$. Given P and Q, two points of $E(\mathbb{F}_{q^n})$, we look for x, if it exists, such that $Q = [x]P$ (where the notation $[m]P$ denotes, as usual, the multiplication of P by m).

(1) First we compute the factor base $\mathscr{F} = \{(x,y) \in E(\mathbb{F}_{q^n}) \mid x \in \mathbb{F}_q\}$.
(2) Then we look for at least $\#\mathscr{F} + 1$ relations of the form

$$[a_j]P \oplus [b_j]Q = P_1 \oplus \cdots \oplus P_n,$$

where $P_1, \cdots, P_n \in \mathscr{F}$ and a_j and b_j are randomly picked up in $\mathbb{Z}$.
(3) Finally, using linear algebra, find $\lambda_1, \ldots, \lambda_{\#\mathscr{F}+1}$ such that the neutral element of $E(\mathbb{F}_{q^n})$ is equal to $\sum_j [\lambda_j \cdot a_j]P \oplus [\lambda_j \cdot b_j]Q$ and return $x = -\frac{A}{B}$ modulo the order of P, where $A = \sum_j \lambda_j \cdot a_j$ and $B = \sum_j \lambda_j \cdot b_j$.

It is important to note that there exists approximately $\frac{q^n}{n!}$ points of $E(\mathbb{F}_{q^n})$ which can be decomposed w.r.t. $\mathscr{F}$, thus each relation of step 2 can be found with probability

$\frac{1}{n!}$. Using the double large prime variation and for a fixed degree extension n, the complexity of this index calculus attack is $O(q^{2-\frac{2}{n}})$. It is thus faster than Pollard rho method in $O(q^{\frac{n}{2}})$ for $n \geq 3$. However, this complexity hides an exponential dependance in n in step 2, which is the main topic of this article.

Definition 1. *The* point decomposition problem*, denoted* **PDP** *in this article, is: Given a point R in an elliptic curve $E(\mathbb{F}_{q^n})$ with a factor base $\mathscr{F}$ formed of the points with an $\mathbb{F}_q$-rational abscissa, find, if they exist, $P_1, \ldots, P_n$ in $\mathscr{F}$, such that $R = P_1 \oplus \cdots \oplus P_n$.*

To solve the PDP, one can use the summation polynomials introduced by Semaev [44] and the resolution of the PDP is equivalent to solve a polynomial system. In this context, the PDP has a complexity in $O\left(\log(q)\left(d^{\omega n} + n \cdot 2^{3n(n-1)}\right)\right)$ where ω is the linear algebra constant and d is the degree of regularity, that is, a bound on the maximal degree reached during the computation of Gröbner basis with F_4 [20] or F_5 [21]. The second part of the PDP complexity is due to the complexity of the FGLM [25, 24] algorithm which is polynomial in the number of solutions of the polynomial system.

We note that Nagao [41] introduced a variant of the index calculus algorithm, well suited to hyperelliptic curves, in which the PDP step is replaced by another approach that creates relations from Riemann-Roch spaces. It also relies, in the end, on polynomial system solving. If the curve is elliptic, it seems to be always better to use Semaev's polynomials and the PDP, so we stick to that case in our study.

1.2. **Contributions.** In the case of the Pollard rho and sibling methods, it is well known that having a small rational subgroup in $\mathbb{G}$ speeds-up the computation by a factor of roughly the square root of the order of this subgroup. It is also the case if there is an explicit automorphism of small order. For index calculus in general, it is far less easy to make use of such an additional structure. For instance, in the multiplicative group of a prime finite field, the number field sieve algorithm must work in the full group, even if one is interested only in the discrete logarithm in a subgroup. A key element is the action of the rational subgroup that must be somewhat compatible with the factor base. See for instance the article by Couveignes and Lercier [12], where a factor base is chosen especially to fit this need, again in the context of multiplicative groups of finite fields.

The aim of this paper is to reveal some elliptic curves where one can indeed make use of the presence of a small rational subgroup to speed-up the index calculus algorithm, and especially the PDP step. In particular, for curve representations having an important interest in cryptographic point of view, we decrease the complexity of the FGLM step. More precisely, we have the following result.

Theorem 1.1. *Let E be an elliptic curve defined over a non binary field $\mathbb{F}_{q^n}$ where $n > 1$. If E can be put in twisted Edwards or twisted Jacobi intersections representation then the complexity of solving the PDP is*

- *(proven complexity)* $O\left(\log(q)\left(d^{\omega n} + n \cdot 2^{3(n-1)^2}\right)\right)$

- *(heuristic complexity)* $O\left(\log(q)\left(d^{\omega n}+n^2\cdot 2^{\omega(n-1)^2}\right)\right)$

where $2\leq\omega<3$ *is the linear algebra constant, and* d *is the degree of regularity which is a bound on the maximal degree reached during the computation of Gröbner basis with* F_5.

The proven complexity of theorem 1.1 is obtained by using the classical complexity of FGLM in $O(nD^3)$ [24]. The heuristic complexity is obtained by using a change of ordering algorithm recently proposed in [23]. This algorithm follows the approach of [25]. In the case of generic polynomial systems this algorithm has a proven complexity of $O(n\log(D)D+\log(D)D^{\omega})$. In the case where the given polynomial system is not generic, a randomization technique allows to obtain the same, but heuristic, complexity.

The main ingredient of the proof of theorem 1.1 is to use the symmetries of the curves corresponding to the group action: they allow to reduce the number of solutions of the polynomial systems to be solved and to speedup intermediate Gröbner bases computations.

The first symmetries to be used are inherent in the very definition of the PDP: the ordering of the P_i's does not change their sum, so that the full symmetric group acts naturally on the polynomial system corresponding to the PDP. It is a classical way to reduce the number of solutions by a factor $n!$, and speed up accordingly the resolution.

Twisted Edwards and twisted Jacobi intersections curves have more symmetries than ordinary elliptic curves, due to the presence of a rational 2-torsion point. It is remarkable that, for the natural choice of the factor base, this action translates into the polynomial systems constructed using summation polynomials in a very simple manner: any sign change on an even number of variables is allowed. This action combined with the full symmetric group gives the so-called Dihedral Coxeter group, see for instance [35]. Using invariant theory techniques [47], we can thus express the system in terms of adapted coordinates, and therefore the number of solutions is reduced by a factor $2^{n-1}\cdot n!$ (the cardinality of the Dihedral Coxeter group). This yields a speed-up by a factor $2^{3(n-1)}$ in the FGLM step, compared to the general case.

In this paper, we present also several practical experiments which confirm the exponential decrease of the complexity. All experiments were carried out using the computer algebra system MAGMA [7] and the FGb library [22].

1.3. **Consequences and limitations.** Our experiments show that for some parameters, the new version of the algorithm is significantly faster than generic algorithms. For instance for a twisted Edwards or twisted Jacobi intersections curve defined over $\mathbb{F}_{q^5}$ where $\log_2(q)=64$, solving the ECDLP with generic algorithms requires 2^{160} operations in $E(\mathbb{F}_{q^n})$ and only 2^{127} basic arithmetic operations (multiplications of two 32 bits words) with our approach.

We do not change the very nature of the attack; therefore it applies only to curves defined over small extension fields. This work has no implication on the ECDLP

instances recommended by the NIST [42], since they are defined over a prime finite field of high characteristic or binary fields of prime degree extension.

1.4. **Related Work.** In [33], Joux and Vitse improve the complexity of the index calculus algorithm for small q. Indeed, to decrease the cost of polynomial systems involved in the attack they look for decompositions of points of the curve in $n-1$ points instead of n. At a high level, it can be seen as looking for a decomposition in n points, where one of the point has been fixed to be the point at infinity. As a consequence, the probability of finding a decomposition is reduced by a factor of q, so that the complexity grows accordingly, and the range of application is for moderate values of q. Conversely, in our work, the dependance in q is not affected, but it is only limited to twisted Edwards and twisted Jacobi intersections curves.

1.5. **Organization of the paper.** The paper is organized as follows. In Section 2, we recall how to use the summation polynomials to solve the PDP. We also present some properties of twisted Edwards and Jacobi intersections curves. In Section 3 we give some results from invariant theory and present a general algorithm for computing a Gröbner basis of an invariant ideal. Section 4 is devoted to the main contribution of this article. We show how 2-torsion and 4-torsion points can be used to efficiently solve the PDP. Finally, we present in Section 5 some experiments which confirm the theoretical results.

2. Point decomposition problem

In this section we first present the point decomposition problem (denoted PDP) in the context of ECDLP and a general method to solve it. Then, we recall summation polynomials introduced by Semaev to improve the efficiency of general method. Finally, we show how to compute summation polynomials corresponding to the PDP over twisted Edwards and Jacobi intersections curves and recall some properties of these curves.

2.1. **General method for solving the PDP.** Let E be an elliptic curve defined over $\mathbb{F}_{q^n}$ with $n>1$. Recall the PDP: given a point $R\in E(\mathbb{F}_{q^n})$ and the factor base $\mathscr{F}=\{(x,y)\in E(\mathbb{F}_{q^n}) \mid x\in\mathbb{F}_q\}\subset E$ find $P_1,\dots,P_n\in\mathscr{F}$ such that

$$R=P_1\oplus\cdots\oplus P_n\,. \tag{1}$$

Writing $\mathbb{F}_{q^n}=\mathbb{F}_q[X]/\mu(X)=\mathbb{F}_q[\omega]$ where $\mu(x)$ is an irreducible polynomial over $\mathbb{F}_q$ of degree n and ω is a root of $\mu(x)$ in $\mathbb{F}_{q^n}$, we can see $\mathbb{F}_{q^n}$ as a vector space over $\mathbb{F}_q$ for which $\{1,\omega,\dots,\omega^{n-1}\}$ is a basis. Frey [28] shows that any instances of the ECDLP can be mapped to an instance of the DLP in the Weil restriction of $E(\mathbb{F}_{q^n})$. In the same way, the PDP over any elliptic curve defined over a non prime finite field can be map to the PDP over the Weil restriction of this curve. Indeed the Weil restriction A of $E(\mathbb{F}_{q^n})$ is the abelian variety of dimension n for which an affine patch can be described by the set of $2n$-tuples $(x_0,\dots,x_{n-1},y_0,\dots,y_{n-1})\in(\mathbb{F}_q)^{2n}$ such that $\left(\sum_{i=0}^{n-1}x_i\cdot\omega^i,\sum_{i=0}^{n-1}y_i\cdot\omega^i\right)$ is a point of $E(\mathbb{F}_{q^n})$. The group law of E infers a group law on A which is given by rational fractions depending on the coordinates

of the summed points. Consequently we can construct $2n$ rational fractions λ_j in terms of the $n(n+1)$ variables $x_{i,0}, y_{i,0}, \ldots, y_{i,n-1}$ for $i = 1, \ldots, n$ such that

$$P_1 \oplus \ldots \oplus P_n = (\lambda_1, \ldots, \lambda_{2n})$$

where $P_i = (x_{i,0}, 0, \ldots, 0, y_{i,0}, \ldots, y_{i,n-1}) \in \mathscr{F}$. To solve the PDP, we write $P_1 \oplus \ldots \oplus P_n = R$ which gives $2n$ equations in $\mathbb{F}_q$. Adding the equations describing $P_i \in \mathscr{F}$ for $i = 1, \ldots, n-1$, we obtain a polynomial system with $n(n+1)$ variables and $n(n+1)$ equations in $\mathbb{F}_q$. The system has as many unknowns as equations then it is in general of dimension 0. In order to solve this system, we use Gröbner basis. The complexity of Gröbner basis computation depends on the number of variables which is quadratic in n. To speed up the resolution, one can reduce the number of variables by using the summation polynomials introduced by Semaev in [44].

2.2. **Solving the PDP using summation polynomials.** The summation polynomials are introduced by Semaev as a projection of the PDP over the set of x-coordinate of each point.

Definition 2. *Let E be an elliptic curve defined over a field $\mathbb{F}_{q^n}$ whose algebraic closure is denoted by $\overline{\mathbb{F}_{q^n}}$. For all $m \geq 2$, the m^{th} summation polynomial of E is defined by $f_m(x_1, \ldots, x_m)$ such that for all $x_1, \ldots, x_m$ in $\overline{\mathbb{F}_{q^n}}$, its evaluation $f_m(x_1, \ldots, x_m)$ is zero if and only if there exist $y_1, \ldots, y_m \in \overline{\mathbb{F}_{q^n}}$ such that (x_i, y_i) is in $E(\overline{\mathbb{F}_{q^n}})$ and $(x_1, y_1) \oplus \ldots \oplus (x_n, y_n)$ is the neutral element of E.*

More generally the summation polynomials can be defined as a projection over the set of any coordinate. Depending on the coordinate we project to, we need to adjust the factor base : let c be the chosen coordinate, $\mathscr{F}$ has to be the set of all points of the curve with c in $\mathbb{F}_q$ instead of $\mathbb{F}_{q^n}$. In the context of definition 2 and if E is in Weierstrass representation we have the following result.

Theorem 2.1 (Semaev [44])**.** *Let E be an elliptic curve defined over a field $\mathbb{K}$ of characteristic > 3 by a Weierstrass equation*

$$E : y^2 = x^3 + a_4 x + a_6 \tag{2}$$

the m^{th} summation polynomials of E are given by

$$\begin{cases} f_2(x_1, x_2) &= x_1 - x_2 \\ f_3(x_1, x_2, x_3) &= (x_1 - x_2)^2 x_3^2 - 2((x_1 x_2 + a_4)(x_1 + x_2) + 2a_6)x_3 + \\ & \quad (x_1 x_2 - a_4)^2 - 4a_6(x_1 + x_2) \\ f_m(x_1, \ldots, x_n) &= Res_X(f_{m-k}(x_1, \ldots, x_{m-k-1}, X), f_{k+2}(x_{m-k}, \ldots, x_m, X)) \\ & \quad \text{for all } m \geq 4 \text{ and for all } m - 3 \geq k \geq 1 \end{cases}$$

where $Res_X(f_1, f_2)$ is the resultant of f_1 and f_2 with respect to X. Moreover, for all $m \geq 3$ the m^{th} summation polynomial is symmetric and of degree 2^{m-2} in each variable.

We now detail how to use the summation polynomials to solve the PDP. Assume that E is given by a Weierstrass equation. By definition, if the points P_1, ..., P_n verify

$$f_{n+1}(x_{P_1}, \ldots, x_{P_n}, x_R) = 0_{\mathbb{F}_{q^n}} \tag{3}$$

then, up to signs, they give a solution of the PDP for R. By applying a Weil restriction, we obtain

$$f_{n+1}(x_{P_1},\dots,x_{P_n},x_R)=0_{\mathbb{F}_{q^n}} \Longleftrightarrow \sum_{k=0}^{n-1}\varphi_k(x_{P_1},\dots,x_{P_n})\cdot\omega^k=0_{\mathbb{F}_{q^n}}$$

where the $\varphi_k(x_{P_1},\dots,x_{P_n})$ are polynomials in $\mathbb{F}_q[x_{P_1},\dots,x_{P_n}]$. Thus, solving equation 3 is equivalent to solve the polynomial system $\mathscr{S}=\{\varphi_k(x_{P_1},\dots,x_{P_n}),\ k=0,\dots,n-1\}$ in $\mathbb{F}_q$.

We will detail in the next section how to solve such a system, taking advantage from the fact that it is symmetric. An important parameter is the degree in each variable which is 2^{n-1}.

Remark 1. *If we choose to define the summation polynomials as a projection to the y-coordinate, then the system $\mathscr{S}$ will have more solutions and consequently the ideal $\langle\mathscr{S}\rangle$ will have higher degree. Indeed if we have several decompositions of a point with the same set of points up to sign change these decompositions do not match with the same solution of the summation polynomials. Choosing the x-coordinate for the projection enables to associate these different decompositions to the same solution of the summation polynomial. Consequently, depending on the curve representation, it is important to define the summation polynomials as a projection on the invariant coordinate by the action of $\ominus$.*

We now study two curve representations having more symmetries than Weierstrass representation. Following the same idea, we will show in the sequel, that these additional symmetries allow to further reduce the difficulty of the resolution of the PDP.

2.3. **Curve representations adding symmetries in the PDP.** Any elliptic curve can be represented by a Weierstrass equation. Among these curves, some share common properties that allow to choose another form of equation. In particular, we study two families of elliptic curves, the twisted Edwards and Jacobi intersections curves.

2.3.1. *Twisted Edwards curves.* This family of elliptic curve was introduced in 2008 in cryptography [5]. This is a generalization of the representation proposed by Edwards in [17]. These curves were deeply studied by the cryptology community, especially by Bernstein and Lange [4], for their efficient arithmetic. In [5] the authors show that the family of twisted Edwards curves is isomorphic to the family of Montgomery curves [40]. In particular these curves always have a rational 2-torsion point $T_2=(0,-1)$ (and a rational 4-torsion point for Edwards curves). A twisted Edwards curve is defined over a field $\mathbb{K}$ of characteristic >2 by

$$E_{a,d}: ax^2+y^2=1+dx^2y^2 \tag{4}$$

where $a,d\neq 0$ and $a\neq d$. If $a=1$, $E_{1,d}$ is an Edwards curve. The group law of a twisted Edwards curve is given by

$$(x_1,y_1)\oplus(x_2,y_2)=\left(\frac{x_1y_2+y_1x_2}{1+dx_1x_2y_1y_2},\frac{y_1y_2-ax_1x_2}{1-dx_1x_2y_1y_2}\right)$$

with neutral element $P_\infty = (0,1)$. The opposite of a point $P = (x,y) \in E_{a,d}(\mathbb{K})$ is $\ominus P = (-x,y)$, and adding T_2 to P gives $P + T_2 = (-x,-y)$. Therefore the symmetries can be interpreted in terms of the group law. If a is a square in $\mathbb{K}$ then a twisted Edwards curve has two 4-torsion points $T_4 = \left(a^{-\frac{1}{2}},0\right)$ or $\left(-a^{-\frac{1}{2}},0\right)$.

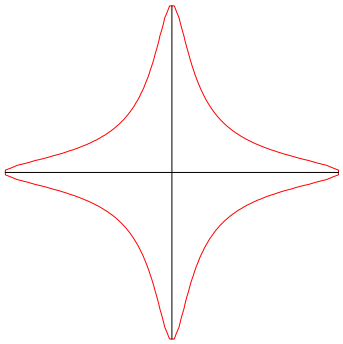

FIGURE 1. Twisted Edwards curve over $\mathbb{R}$.

To solve the PDP in twisted Edwards representation, we have to construct the summation polynomial of a such curve. As said in the remark 1, we compute the summation polynomials as a projection of the PDP to the coordinate which is invariant under the $\ominus$ action. That is to say the y-coordinate for twisted Edwards curves. The n^{th} summation polynomial for twisted Edwards curves is then given by

$$\begin{cases} f_2(y_1,y_2) &= y_1 - y_2 \\ f_3(y_1,y_2,y_3) &= (y_1^2y_2^2 - y_1^2 - y_2^2 + \frac{a}{d})y_3^2 + 2\frac{d-a}{d}y_1y_2y_3 + \\ & \quad \frac{a}{d}\left(y_1^2 + y_2^2 - 1\right) - y_1^2y_2^2 \\ f_n(y_1,\ldots,y_n) &= \text{Res}_Y(f_{n-k}(y_1,\ldots,y_{n-k-1},Y), f_{k+2}(y_{n-k},\ldots,y_n,Y)) \\ & \quad \text{for all } n \geq 4 \text{ and for all } n-3 \geq k \geq 1 \end{cases}$$

As in the case of Weierstrass representation, for all $n \geq 3$ the n^{th} summation polynomial is symmetric (see proof in Section 4.1.2) and of degree 2^{n-2} in each variable.

2.3.2. *Twisted Jacobi intersections curves.* This form of elliptic curves was introduced in 2010 in [27]. As for twisted Edwards curves, it is a generalization of Jacobi intersections curves (which are the intersection of two quadratic surfaces defined in a 3-dimensional space) proposed by D.V. and G.V. Chudnovsky in [9]. The twisted Jacobi intersections is defined over a non binary field $\mathbb{K}$ by

$$E_{a,b} : \begin{cases} ax^2 + y^2 = 1 \\ bx^2 + z^2 = 1 \end{cases}$$

where $a,b \in \mathbb{K}$, $a,b \neq 0$ and $a \neq b$. If $a = 1$, $E_{1,b}$ is a Jacobi intersections curve. The family of twisted Jacobi intersections curves contains all curves having three rational 2-torsion points. These three 2-torsion points are $T_2 = (0,1,-1),(0,-1,1)$ and $(0,-1,-1)$. The neutral element is $P_\infty = (0,1,1)$ and the negative of a point $P = (x,y,z) \in E_{a,b}(\mathbb{K})$ is given by $\ominus P = (-x,y,z)$. Adding one of the 2-torsion point to P gives the point $(\pm x, \pm y, \pm z)$. The group law is given by

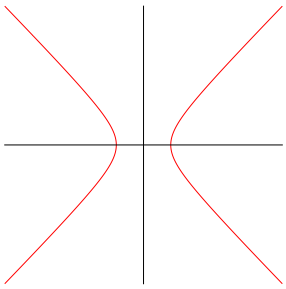

FIGURE 2. Twisted Jacobi intersections curve over $\mathbb{R}$.

$$(x_1,y_1,z_1) \oplus (x_2,y_2,z_2) = \left(\frac{x_1y_2z_2 + x_2y_1z_1}{y_2^2 + az_1^2x_2^2}, \frac{y_1y_2 - ax_1z_1x_2z_2}{y_2^2 + az_1^2x_2^2}, \frac{z_1z_2 - bx_1y_1x_2y_2}{y_2^2 + az_1^2x_2^2}\right).$$

Jacobi intersections curves can have zero, four or eight 4-torsion points :

- $\left(\pm\frac{1}{\sqrt{b}},\pm\sqrt{\frac{b-a}{b}},0\right)$, if $a\neq 1$ non square or $a=1$ and -1 non square and b and $b-a$ are square in $\mathbb{K}$.
- $\left(\pm\frac{1}{\sqrt{a}},0,\pm\sqrt{\frac{a-b}{a}}\right)$, if $b\neq 1$ non square or $b=1$ and -1 non square and a and $a-b$ are square in $\mathbb{K}$.
- $\left(\pm\frac{1}{\sqrt{b}},\pm\sqrt{\frac{b-a}{b}},0\right)$, $\left(\pm\frac{1}{\sqrt{a}},0,\pm\sqrt{\frac{a-b}{a}}\right)$, if $a,b,-1$ and $a-b$ are square in $\mathbb{K}$.

For these curves the y and z coordinates are invariant under the action of $\ominus$. Hence we can compute the summation polynomials for these curves as a projection of the PDP to the y or z coordinate. In fact the two summation polynomials for n fixed are the same up to permutation of a and b, so we give only the polynomials obtained by projection to y:

$$\begin{cases} f_2(y_1,y_2) &= y_1-y_2\\ f_3(y_1,y_2,y_3) &= \left(y_1^2y_2^2-y_1^2-y_2^2+\frac{b-a}{b}\right)y_3^2+2\frac{a}{b}y_1y_2y_3+\\ & \quad \frac{b-a}{b}\left(y_1^2+y_2^2-1\right)-y_1^2y_2^2\\ f_n(y_1,\ldots,y_n) &= \mathrm{Res}_Y(f_{n-k}(y_1,\ldots,y_{n-k-1},Y),f_{k+2}(y_{n-k},\ldots,y_n,Y))\\ & \quad \text{for all } n\geq 4 \text{ and for all } n-3\geq k\geq 1 \end{cases}$$

As for Weierstrass or twisted Edwards representation, for all $n\geq 3$ the n^{th} summation polynomial is symmetric and of degree 2^{n-2} in each variable.

To take advantage of the symmetries introduced by twisted Edwards and Jacobi intersections curves, we have to know how to use the symmetries of a polynomial ideal to simplify the computation of its Gröbner basis; this is the topic of the next section.

3. Solving polynomial systems and symmetries

In this section we first recall a recent strategy to solve polynomial systems. Then we briefly give some backgrounds of invariant theory needful to the last part of this section devoted to the resolution of polynomial systems invariant under the action of a linear group.

3.1. **Gröbner basis.**

A Gröbner basis of the ideal $\mathscr{I}\subset\mathbb{K}[x_1,\ldots,x_n]$ is defined for a fixed monomial ordering. It is a set of polynomials generating $\mathscr{I}$ which has good properties. In particular from lexicographical Gröbner basis of $\mathscr{I}$ one can read off the solutions – the variety – of the ideal. Indeed, the reduced lexicographical Gröbner basis of an ideal having a finite set of solutions over $\overline{\mathbb{K}}$ has the following

triangular form

$$\begin{cases} h_{1,1}(x_1,\ldots,x_n),\ldots,h_{1,k_1}(x_1,\ldots,x_n) \\ h_{2,1}(x_2,\ldots,x_n),\ldots,h_{2,k_2}(x_2,\ldots,x_n) \\ \vdots \\ h_{n-1,1}(x_{n-1},x_n),\ldots,h_{n-1,k_{n-1}}(x_{n-1},x_n) \\ h_n(x_n) \end{cases}$$

From such a triangular system, one can find the solutions of $\mathscr{I}$. Indeed, using the LEXtriangular algorithm [38] one can decompose this triangular system in a family of triangular systems where each of them verify $k_i = 1$ for all $i = 1,\ldots,n-1$. Thus, by factoring univariate polynomials using Berlekamp or Cantor-Zassenhaus algorithms (see [49]), one can find the solutions of each triangular system. Finally, the solutions of $\mathscr{I}$ is the union of the solutions of each triangular system.

Usually, to compute such a Gröbner basis we proceed in two steps. First we compute a degree reverse lexicographical Gröbner basis with F_4 or F_5 [20, 21] which complexities can be bounded by $O\left(\binom{n+d}{d}^{\omega}\right)$, where n is the number of variables of the system, d is a bound on the maximal degree reached by the polynomials during the computation of the Gröbner basis (namely the degree of regularity of the system) and ω is the linear algebra constant (see [3]). Then we compute the lexicographical Gröbner basis by using a change of ordering algorithm, FGLM [25, 24]. The classical complexity for this step is $O(nD^3)$ where D is the degree of the ideal (the number of solutions counted with multiplicities in the algebraic closure of $\mathbb{K}$). For generic systems, this complexity can be reduced to $O(n\log(D)D + \log(D)D^{\omega})$ (see [23]).

However, in this paper to solve polynomial systems by using Gröbner basis, we use a recent strategy proposed in [23]. This strategy has been checked on various examples. In particular, it is valid and more efficient for examples studied in this paper (see Section 5). Change of ordering algorithms require to compute multiplication matrices T_i describing the multiplication by x_i in the quotient ring $\mathbb{K}[x_1,\ldots,x_n]/\langle G_{\mathrm{DRL}}\rangle$ (for ideals in shape position, only the matrix T_n is required). That is to say, let $B = \{\varepsilon_1,\ldots,\varepsilon_D\}$ be the canonical basis of $\mathbb{K}[x_1,\ldots,x_n]/\langle G_{\mathrm{DRL}}\rangle$ seen as a D-dimensional vector space. The i^{th} column of the matrix T_n is constructed as the normal form of $\varepsilon_i x_n$. From [24], the terms $\varepsilon_i x_n$ can be of three types:

I. $\varepsilon_i x_n$ is in B, then $\varepsilon_i x_n = \varepsilon_j$ for some j in $[i+1,\ldots,D]$ and $\mathrm{NF}(\varepsilon_i x_n) = \varepsilon_j$.
II. $\varepsilon_i x_n$ is a leading term of a polynomial of G_{DRL}. Hence, let $g \in G_{\mathrm{DRL}}$ such that $\mathrm{LT}(g) = \varepsilon_i x_n$ then $\mathrm{NF}(\varepsilon_i x_n) = LT(g) - g$.
III. otherwise, the normal form of $\varepsilon_i x_n$ has to be computed.

When the ideal $\mathscr{I}$ is in shape position, we propose the following strategy to solve polynomial systems. First we compute the DRL Gröbner basis of $\mathscr{I}$, then we try to compute the multiplication T_n. If all terms $\varepsilon_i x_n$ are of type I or II then we compute the LEX Gröbner basis by using a change of ordering algorithm. Else, if a term $\varepsilon_i x_n$ is of type III we stop the computation of the matrix T_n and we consider the

new ideal $\mathscr{I}^{(t)} \subset \mathbb{K}[x_1,\dots,x_n,t]$, with $x_1 > \cdots > x_n > t$, generated by the DRL Gröbner basis of $\mathscr{I}$ and the equation $t - \lambda_1 x_1 - \lambda_2 x_2 - \cdots - \lambda_n x_n$ where the λ_i's are randomly chosen in $\mathbb{K}$. Under the heuristic proposed in [23] all terms needed to construct the multiplication matrix w.r.t. the variable t are of type I or II. This strategy has the same complexity that the usual one.

In conclusion, solving a polynomial system using Gröbner basis has a complexity given by:

- generic systems: $O\left(\binom{n+d}{d}^{\omega} + n\log(D)D + \log(D)D^{\omega}\right)$
- non-generic systems:
  - proven: $O\left(\binom{n+d}{d}^{\omega} + nD^3\right)$
  - heuristic: $O\left(\binom{n+d}{d}^{\omega} + n\log(D)D + \log(D)D^{\omega}\right)$

For systems having symmetries *i.e.* invariant under the action of a linear group, computing directly a Gröbner basis breaks symmetries, which is not satisfactory. The two next sections are devoted to handle symmetries in the polynomial systems solving process.

3.2. **Invariant ring and reflection groups.** In this paper, we assume that the field $\mathbb{K}$ has a "large enough characteristic", that is to say not dividing the cardinality of the linear group $\mathrm{GL}(\mathbb{K},n)$. All notions of invariant theory recall in the following section, can be generalized to an affine variety instead of the affine space.

A linear group $\mathbb{G} \subset \mathrm{GL}(\mathbb{K},n)$ naturally acts on the affine space $\mathbb{A}^n$ by the matrix vector multiplication. This action can be translated to polynomial ring, more precisely we have the following definition.

Definition 3 (Invariant ring)**.** *Let $\mathbb{K}[x_1,\dots,x_n]$ be a polynomial ring in n variables with coefficients in $\mathbb{K}$. The action of $\mathbb{G}$ on $\mathbb{A}^n$ defines an action of $\mathbb{G}$ on $\mathbb{K}[x_1,\dots,x_n]$ by*

$$\begin{array}{ccc} \mathbb{G}\times\mathbb{K}[x_1,\dots,x_n] & \longrightarrow & \mathbb{K}[x_1,\dots,x_n] \\ g, f & \longmapsto & g\cdot f \end{array}$$

where $g\cdot f$ is defined by $(g\cdot f)(v) = f(g^{-1}\cdot v)$ for all $v \in \mathbb{A}^n$. The invariant ring of $\mathbb{G}$ is the set of all invariant polynomials in $\mathbb{K}[x_1,\dots,x_n]$:

$$\mathbb{K}[x_1,\dots,x_n]^{\mathbb{G}} = \{f \in \mathbb{K}[x_1,\dots,x_n] \mid g\cdot f = f \text{ for all } g \in \mathbb{G}\}.$$

The invariant theory fundamental problem was to decide if invariant ring have a finite system of generators. The answer is given by Hilbert in the last decade of the nineteenth century and it is summarized in the following theorem.

Theorem 3.1 (Hilbert's finiteness theorem)**.** *The invariant ring of $\mathbb{G}$ is finitely generated.*

Precisely $\mathbb{K}[x_1,\dots,x_n]^{\mathbb{G}}$ is Cohen-Macaulay, that is to say it is a finitely generated free module over $\mathbb{K}[\theta_1,\dots,\theta_n]$ where $\theta_1,\dots,\theta_n$ are algebraically independent.

Consequently there exist $\eta_1,\dots,\eta_t \in \mathbb{K}[x_1,\dots,x_n]^{\mathbb{G}}$ such that

$$\mathbb{K}[x_1,\dots,x_n]^{\mathbb{G}} = \bigoplus_{i=1}^{t} \eta_i \mathbb{K}[\theta_1,\dots,\theta_n]. \tag{5}$$

The decomposition 5 is called a Hironaka decomposition of the Cohen-Macaulay algebra $\mathbb{K}[x_1,\dots,x_n]^{\mathbb{G}}$. The polynomials $\theta_1,\dots,\theta_n$ are called the primary invariants of $\mathbb{K}[x_1,\dots,x_n]^{\mathbb{G}}$ and $\eta_1,\dots,\eta_t$ are the secondary invariants of $\mathbb{K}[x_1,\dots,x_n]^{\mathbb{G}}$.

A natural question of invariant theory is to know under which conditions on $\mathbb{G}$, its invariant ring is a graded polynomial algebra. The answer is given in the following theorem.

Theorem 3.2 (Chevalley-Shepard-Todd [8, 45])**.** *The invariant ring of $\mathbb{G}$ is a polynomial algebra if and only if $\mathbb{G}$ is a pseudo-reflection group.*

A group $\mathbb{G} \subset \mathrm{GL}(\mathbb{K},n)$ is said to be a pseudo-reflection group if it is generated by its pseudo-reflections. A pseudo-reflection is a linear automorphism of $\mathbb{A}^n$ that is not the identity map, but leaves a hyperplane $H \subset \mathbb{A}^n$ pointwise invariant.

Example 1. *Coxeter group are reflection groups. Particularly the dihedral Coxeter group $D_n = (\mathbb{Z}/2\mathbb{Z})^{n-1} \rtimes \mathfrak{S}_n$ is a reflection group. D_n acts on $\mathbb{A}^n$ by the rule that $\mathfrak{S}_n$ permutes a chosen basis, whereas $(\mathbb{Z}/2\mathbb{Z})^{n-1}$ changes the sign on an even number of basis elements. From theorem 3.2 the invariant ring of D_n is a polynomial algebra. The dihedral Coxeter group is a well known group and its invariant ring too. Actually,*

$$\mathbb{K}[x_1,\dots,x_n]^{D_n} = \mathbb{K}[p_2,\dots,p_{2(n-1)},e_n] = \mathbb{K}[s_1,\dots,s_{n-1},e_n]$$

where $p_i = \sum_{k=1}^{n} x_k^i$ is the i^{th} power sum, $s_i = \sum_{1\le k_1<\dots<k_i\le n} \prod_{j=1}^{i} x_{k_j}^2$ is the i^{th} elementary symmetric polynomial in terms of $x_1^2,\dots,x_n^2$ and $e_n = \prod_{k=1}^{n} x_k$ is the n^{th} elementary symmetric polynomial in terms of $x_1,\dots,x_n$.

We now see how to use these properties to simplify the resolution of polynomial systems invariant under a linear group.

3.3. **Solving invariant ideals.** Let $\mathscr{I} = \langle \rho_1(x_1,\dots,x_n),\dots,\rho_s(x_1,\dots,x_n)\rangle$ be an ideal of $\mathbb{K}[x_1,\dots,x_n]$ such that for $i = 1,\dots,s$, $\rho_i \in \mathbb{K}[x_1,\dots,x_n]^{\mathbb{G}}$. Clearly the variety $V(\mathscr{I})$ is $\mathbb{G}$-invariant. Let $V(\mathscr{I})/\mathbb{G}$ be the set of $\mathbb{G}$-orbits of $V(\mathscr{I})$, we call it the orbit variety of $\mathscr{I}$. As the invariant ring of $\mathbb{G}$ is Cohen-Macaulay, we will see in the sequel that from $V(\mathscr{I})/\mathbb{G}$ one can compute all elements in $V(\mathscr{I})$. Consequently to compute Gröbner basis keeping symmetries, one can compute a Gröbner basis for the orbit variety $V(\mathscr{I})/\mathbb{G}$ instead of $V(\mathscr{I})$ and then find all elements in all orbits $\widetilde{v} \in V(\mathscr{I})/\mathbb{G}$. Let $\{I_1(x_1,\dots,x_n),\dots,I_r(x_1,\dots,x_n)\}$ be a set of generators – primary and secondary invariants – of $\mathbb{K}[x_1,\dots,x_n]^{\mathbb{G}}$. These invariants define the $\mathbb{G}$-orbits space $\mathbb{A}^n/\mathbb{G}$ as an algebraic subvariety of the affine space $\mathbb{A}^r$ *i.e.* $V(\mathscr{I})/\mathbb{G} \subset \mathbb{A}^r$. Let $\mathscr{G}_i$ be the lexicographical Gröbner Basis of

$$\langle I_1(x_1,\dots,x_n)-y_1,\dots,I_r(x_1,\dots,x_n)-y_r\rangle \subset \mathbb{K}[x_1,\dots,x_n,y_1,\dots,y_r]$$

where $x_1 > \ldots > x_n > y_1 > \ldots > y_r$. Let $\widetilde{v} = (\widetilde{v}_1, \ldots, \widetilde{v}_r) \in V(\mathscr{I})/\mathbb{G}$. All elements in the $\mathbb{G}$-orbit represented by $\widetilde{v}$ can be found by substituting the variables $y_1, \ldots, y_n$ by $\widetilde{v}_1, \ldots, \widetilde{v}_r$ in the lexicographical Gröbner basis $\mathscr{G}_i$.

To compute $V(\mathscr{I})/\mathbb{G}$ we have to compute a Gröbner basis $\mathscr{G}_0$ of

$$\mathscr{G}_i \cup \{\rho_1(x_1, \ldots, x_n), \ldots, \rho_s(x_1, \ldots, x_n)\}$$

with respect to an elimination order. Finally $\mathscr{G} = \mathscr{G}_0 \cap \mathbb{K}[y_1, \ldots, y_n]$ is a Gröbner basis of the ideal of $V(\mathscr{I})/\mathbb{G}$.

Example 2. *Let $n = 2$ and $\mathbb{K} = \mathbb{F}_{65521}$. Let us consider the ideal $\mathscr{I} = \langle \rho_1, \rho_2 \rangle$ where*

$$\begin{aligned} \rho_1(x_1, x_2) &= x_1^2 x_2^2 - x_1^2 - x_2^2 - 1 \\ \rho_2(x_1, x_2) &= x_1^4 + x_1^3 x_2 + x_1 x_2^3 + x_2^4 . \end{aligned}$$

The action of D_2 leaves invariant both $\mathscr{I}$ and its variety, but not its lexicographical Gröbner basis, which is:

$$\begin{cases} 4x_1 + 3x_2^{15} - 16x_2^{13} + 29x_2^{11} - 23x_2^9 - 2x_2^7 + 21x_2^5 + 16x_2^3 + 8x_2 \\ x_2^{16} - 5x_2^{14} + 8x_2^{12} - 5x_2^{10} - 2x_2^8 + 5x_2^6 + 8x_2^4 + 5x_2^2 + 1 \end{cases}$$

The corresponding $\mathscr{G}$ basis in terms of $y_1 = x_1^2 + x_2^2$ and $y_2 = x_1 x_2$ is

$$\begin{cases} y_1 - y_2^2 + 1 \\ y_2^4 + y_2^3 - 4y_2^2 - y_2 + 1 \end{cases}$$

which preserves the symmetries.

In the above example, the polynomial systems are simpler when we use the symmetries. This is due to the good properties of the Coxeter dihedral group. Indeed if the invariant ring of $\mathbb{G}$ is not a polynomial algebra – the secondary invariants are not reduced to 1 – considering the symmetries can complicate the resolution of the system. Indeed, the secondary invariant are not independent then considering the symmetries when these invariants are not reduced to 1 increases the number of equations and variables. Consequently, the polynomial systems will be most difficult to solve. Moreover, computing a Hironaka decomposition can be a difficult task. In the case where the invariant ring is not a polynomial algebra one can use also SAGBI Gröbner basis, see for instance [26]; we will not need this strategy in this work.

In our case, where the groups are reflection groups, the impact on the complexity comes from the fact that we keep just one representative instead of the complete orbit of solutions. Hence the runtime of the F_4 and FGLM steps are reduced accordingly. We now see more precisely the relation between the number of solutions of $\mathscr{I}$ and the number of solutions of the ideal corresponding to $\mathscr{I}$ after the change of variables associated to $\mathbb{G}$.

From the class formulae, for all $v \in V(\mathscr{I})$ we have

$$\#\mathrm{Orb}\,(\mathbb{G}, v) = \frac{\#\mathbb{G}}{\#\mathrm{Stab}\,(\mathbb{G}, v)} .$$

In fact, the stabilizer of an element v of the variety counts its multiplicities. The degree of the ideal $\mathscr{I}$ is the number of solutions counted with multiplicities. It is denoted $\deg(\mathscr{I})$. Moreover, $V(\mathscr{I}) = \bigcup_{v \in V(\mathscr{I})} \mathrm{Orb}(\mathbb{G}, v)$ thus

$$\deg(\mathscr{I}) = \sum_{\widetilde{v} \in V(\mathscr{I})/\mathbb{G}} m_{\widetilde{v}} \cdot \#\mathrm{Stab}(\mathbb{G}, v) \cdot \#\mathrm{Orb}(\mathbb{G}, v) = N \cdot \#\mathbb{G}$$

where $m_{\widetilde{v}}$ is the multiplicities of $\widetilde{v}$ in $V(\mathscr{I})/\mathbb{G}$, v is an element of the orbit represented by $\widetilde{v}$ and N the number of $\mathbb{G}$-orbits counted with multiplicities in $V(\mathscr{I})/\mathbb{G}$. By applying the change of variables associated to $\mathbb{G}$ we keep only one element in each orbit of $\mathbb{G}$ in $V(\mathscr{I})$. Hence the number of solutions counted with multiplicities of the ideal for $V(\mathscr{I})/\mathbb{G}$ is the number of orbits of $\mathbb{G}$ counted with multiplicities in $V(\mathscr{I})$ that is to say N. In conclusion, considering the action of a linear group divides the degree of the ideal by the group cardinality. Thus the heuristic complexity of FGLM is divided by $(\#\mathbb{G})^{\omega}$ and by $(\#\mathbb{G})^3$ if we stick to proven results.

Example 3. *Continuing the example 2, the degree of $\mathscr{I}$ is* 16 *where the solutions* $(2996, 62525), (6897, 58624), (58624, 6897)$ *and* $(62525, 2996)$ *are of multiplicity two. The degree of $\mathscr{G}$ is* $4 = \dfrac{16}{\#D_2}$ *and*

- $P_1 = (64799, 361)$ *is a representative of* $\{P_1, (2996, 62525), (62525, 2996)\}$
- $P_2 = (726, 65158)$ *is a representative of* $\{P_2, (6897, 58624), (58624, 6897)\}$
- $P_3 = (6009, 6009)$ *is a representative of* $\{P_3, (7493, 55256), (10265, 58028), (55256, 7493), (58028, 10265)\}$
- $P_4 = (59513, 59513)$ *is a representative of* $\{P_4, (14169, 28989), (28989, 14169), (36532, 51352), (51352, 36532)\}$

Remark 2. *Let $\mathscr{I}$ be an ideal of $\mathbb{K}[x_1, \ldots, x_n]$. Assume that $\mathscr{I}$ is an invariant ideal w.r.t. $\mathbb{G}$, that is to say for all $f \in \mathscr{I}$ and for all $g \in \mathbb{G}$, $g \cdot f \in \mathscr{I}$. Then $\mathscr{I}$ is not necessarily an ideal of $\mathbb{K}[x_1, \ldots, x_n]^{\mathbb{G}}$. For such ideal one can apply the above method to the sub-ideal containing only the invariant polynomials of $\mathscr{I}$. This sub-ideal has the same radical that $\mathscr{I}$ hence the same variety. For more details see for instance* [47].

4. Use of symmetries to improve the ECDLP solving

We now come back to the PDP problem, which is the heart of the index calculus attack on elliptic curves. We will start by recalling the well-known strategy of using the symmetric group to reduce the size of the systems, and then we will consider the case of twisted Edwards and Jacobi intersections that provide further symmetries.

4.1. Group action on the point decomposition problem.

4.1.1. *The symmetric group $\mathfrak{S}_n$.* Depending on the curve representation, the coordinate chosen for the projection can be x, y or z. For more generality, here we note the chosen coordinate c and the $(n+1)^{\text{th}}$ summation polynomial evaluated in one variable in the c-coordinate of R is denoted f_{n+1}^R. As we have seen in Section 2, the summation polynomials are symmetric and it is natural [29] to use this to decrease the cost of the Gröbner basis computation. It is well known that the invariant ring of $\mathfrak{S}_n$ is a polynomial algebra with basis $\{e_1, \ldots, e_n\}$ where e_i is the i^{th} elementary symmetric polynomial in terms of $c_1, \ldots, c_n$. There exists a unique polynomial $g_n^R \in \mathbb{F}_{q^n}[e_1, \ldots, e_n]$ such that g_n^R is the expression of f_{n+1}^R in terms of the e_i. We have seen in Section 2 that f_{n+1} is of degree 2^{n-1} in each variable thus f_{n+1}^R too. Consequently, by construction g_n^R is of total degree 2^{n-1}. Hence after the Weil restriction on g_n^R we obtain a new system $\mathscr{S}_{\mathfrak{S}_n}{}^1 \subset \mathbb{F}_q[e_1, \ldots, e_n]$ with n polynomials of total degree 2^{n-1}. The Bezout's bound allows to bound the degree of the ideal generated by $\mathscr{S}_{\mathfrak{S}_n}$ by $2^{n(n-1)}$. In practice, we observe in this context that this bound is reached. Without taking into account the symmetric group, the bound would have been $n!$ times larger, therefore, the complexity of FGLM is reduced by $(n!)^\omega$ (or by $(n!)^3$ in the non-heuristic case). Moreover the degree of the equations of $\mathscr{S}_{\mathfrak{S}_n}$ are smaller than those of the equations of $\mathscr{S}$. Even if the gain of the F_4, F_5 algorithms is not quantifiable in theory, it is significant in practice.

We are able to solve these systems for $n = 2, 3, 4$. For $n = 2$ or 3 the resolution is instantaneous for all curve representations. In the following, we present some practical results for $n = 4$ obtained by using the computer algebra system MAGMA (V2.17-1) on a 2.93 GHz Intel® E7220 CPU.

$\log_2(q)$		F_4 (s)	Change-Order (s)	Total time (s)
16	Weierstrass [29]	4	531	535
	Edwards	0	201	201
	Jacobi	0	209	209
64	Weierstrass [29]	354	4363	4717
	Edwards	3	1100	1103
	Jacobi	4	1448	1452

We note that for twisted Edwards or Jacobi intersections curves the running time of the system resolution is equivalent and significantly smaller than for Weierstrass representation. This can be explained by the particular shapes of the lexicographical Gröbner basis :

[1]The notation $\mathscr{S}_{\mathbb{G}}$ means that the system is expressed w.r.t. the change of variables associated to $\mathbb{G}$ *i.e.* the change of variables formed by the primary invariant of $\mathbb{F}_q[x_1, \ldots, x_n]^{\mathbb{G}}$.

Lexicographical Gröbner basis of $\mathscr{S}_{\mathfrak{S}_n}$ for Weierstrass representation :

$$\begin{cases} e_1 + h_1(e_n) \\ e_2 + h_2(e_n) \\ \vdots \\ e_{n-2} + h_{n-2}(e_n) \\ e_{n-1} + h_{n-1}(e_n) \\ h_n(e_n) \end{cases}$$

Lexicographical Gröbner basis of $\mathscr{S}_{\mathfrak{S}_n}$ for twisted Edwards and Jacobi intersections representations :

$$\begin{cases} e_1 + \mathfrak{p}_1(e_{n-1}, e_n) \\ e_2 + \mathfrak{p}_2(e_{n-1}, e_n) \\ \vdots \\ e_{n-2} + \mathfrak{p}_{n-2}(e_{n-1}, e_n) \\ \mathfrak{p}_{n-1}(e_{n-1}, e_n) \\ \mathfrak{p}_n(e_n) \end{cases}$$

where $\deg(h_n) = 2^{n(n-1)}$, $\deg(\mathfrak{p}_n) = 2^{(n-1)^2}$, $\deg_{e_{n-1}}(\mathfrak{p}_{n-1}) = 2^{n-1}$ and for all curve representations $\#V_{\overline{\mathbb{F}_q}}(\langle \mathscr{S}_{\mathfrak{S}_n} \rangle) = 2^{n(n-1)}$. Actually, the gain of efficiency observed in the case of twisted Edwards and Jacobi intersections curves is due to the smaller degree appearing in the computation of Gröbner basis of $\mathscr{S}_{D_n}$ in comparison with the Weierstrass case. Note that the lexicographical Gröbner basis for Weierstrass representation is in shape lemma. That is to say, to find the solutions of the system from the lexicographical Gröbner basis, we need to factor only one univariate polynomial in the smallest variable. The value of the others variables is obtained when the value of the smallest variable is fixed. In this case, the smallest variable, here e_n, is said to be separating (see for instance [10]). This means that any element in the variety of the ideal generated by $\mathscr{S}_{\mathfrak{S}_n}$ is distinguishable by e_n. Contrary to Weierstrass representation, the lexicographical Gröbner basis for twisted Edwards and Jacobi intersections curves are not in shape lemma. The variable e_n is not separating for these two representations. This implies that there is an additional group acting on the variety and for which the n^{th} elementary symmetric polynomial is invariant. In the next section, we will see that this group action is linked to the 2-torsion point and we will use this action to explicitly improve the resolution to the system instead of blindly trusting a generic approach to take advantage of symmetries.

4.1.2. *Consequence of the existence of* 2*-torsion points for twisted Edwards and Jacobi intersections curves.* Suppose that we have a solution $(P_1, P_2, \ldots, P_n)$ to the PDP, and denote by T_2 a 2-torsion point. Thus for all $k = 1, \ldots, \left\lfloor \frac{n}{2} \right\rfloor$ we have $P_1 \oplus \ldots \oplus P_n \oplus [2k]T_2 = R$. Therefore from one decomposition of R (modulo the order) we have in fact $\sum_{k=0}^{\left\lfloor \frac{n}{2} \right\rfloor} \binom{n}{2k} = 2^{n-1}$ decompositions of R obtained by adding

an even number of times a 2-torsion point :

$$\begin{aligned} R &= P_1 \oplus \cdots \oplus P_n \\ &= (P_1 \oplus T_2) \oplus (P_2 \oplus T_2) \oplus P_3 \oplus \cdots \oplus P_n \\ &= (P_1 \oplus T_2) \oplus P_2 \oplus (P_3 \oplus T_2) \oplus P_4 \oplus \cdots \oplus P_n \\ &\vdots \\ &= P_1 \oplus \cdots \oplus P_{n-2} \oplus (P_{n-1} \oplus T_2) \oplus (P_n \oplus T_2) \\ &= (P_1 \oplus T_2) \oplus (P_2 \oplus T_2) \oplus (P_3 \oplus T_2) \oplus (P_4 \oplus T_2) \oplus P_5 \oplus \cdots \oplus P_n \\ &\vdots \end{aligned}$$

In general, these decompositions do not correspond to solutions of the PDP, since $(P_i + T_2)$ is not always in the factor base $\mathscr{F}$. Let us now consider the consequence on the system obtain from summation polynomials with respect to a coordinate c. The complexity of the computation of Gröbner basis (FGLM) depends on the number of solutions in the algebraic closure of the coefficient field, counted with multiplicities. Hence if the c-coordinate of $P_i \oplus T_2$ depends only on the c-coordinate of P_i, then the $2^{n-1}-1$ aforementioned decompositions of R correspond to solutions (in the algebraic closure of the coefficient field) of the polynomial system. Consequently, we now study for which curve representation, this property is verified. Let $P = (x,y)$ (respectively (x,y) or (x,y,z)) be a point of $E(\mathbb{F}_{q^n})$ (respectively $E_{a,d}(\mathbb{F}_{q^n})$ or $E_{a,b}(\mathbb{F}_{q^n})$).

For Weierstrass representation, the 2-torsion points of $E(\mathbb{F}_{q^n})$ are $T_2 = (X,0)$ where X is a root of $X^3 + a_4X + a_6 = 0$ and we have

$$P \oplus T_2 = \left(\frac{y^2}{(X-x)^2} - x - X, \frac{(2x+X)y}{(x-X)} - \frac{y^3}{(x-X)^3} - y \right).$$

In this representation, we project the PDP on x-coordinate. As the x-coordinate of the point $P \oplus T_2$ does not depend only of x, the $2^{n-1}-1$ decompositions of the point R are not taken into account during the resolution of the polynomial system associated to the decomposition of the point R.

In case of twisted Edwards representation, the 2-torsion point of a twisted Edwards curve is $T_2 = (0,-1)$ and $P \oplus T_2 = (-x,-y)$. Thus the 2^{n-1} decompositions of the point R translate into as many solutions of the PDP.

Finally for twisted Jacobi intersections representation, the three 2-torsion points of a twisted Jacobi intersections curve are $T_2 = (0,1,-1),(0,-1,1),(0,-1,-1)$. Thus we have $P \oplus T_2 = (\pm x, \pm y, \pm z)$ and similarly to the twisted Edwards curves, the 2^{n-1} decompositions should be solutions of the system associated to the decomposition of the point R. The action of the three 2-torsion points of a twisted Jacobi intersections curve, is not distinguishable after projection. Consequently all decompositions obtained from $(P_1,\ldots,P_n)$ and the three 2-torsion points match with only 2^{n-1} solutions of the system associated to the projection of the decomposition of R.

In conclusion, for twisted Edwards and Jacobi intersections curves, the 2-torsion points give from one decomposition of a given point R, 2^{n-1} decompositions of

this point in n elements of the factor base $\mathscr{F}$. When we project the PDP to the appropriate coordinate, the sum of a point P with T_2 is equivalent to a sign change over the coordinate of P. Hence from one solution $(v_1,\dots,v_n) \in (\overline{\mathbb{F}_{q^n}})^n$ of f_{n+1}^R, we have not only $n!$ solutions coming from $\mathfrak{S}_n$ (see Section 4.1.1) but $n! \cdot 2^{n-1}$: all n-tuples formed by $(v_1,\dots,v_n)$ to which we apply an even number of sign changes and a permutation of $\mathfrak{S}_n$, that is the orbit of $(v_1,\dots,v_n)$ under the action of the Coxeter group D_n introduced in Section 3.

If a linear group acts on the variety of a polynomial system, there is no guarantee that the system is in the invariant ring of the linear group. In our case, the system obtain from f_{n+1}^R by a Weil restriction is invariant under the action of D_n and we have the following result.

Proposition 1. $f_{n+1}^R(c_1,\dots,c_n) \in \mathbb{F}_{q^n}[c_1,\dots,c_n]^{D_n}$.

The idea of the proof is to use the relations between generators of the Dihedral Coxeter group to show that these generators leave f_{n+1}^R invariant. First we use the action of the linear group D_n under the solutions of f_{n+1}^R to underline that for any g in D_n, the action of g on f_{n+1}^R leaves it invariant, up to a multiplicative factor $h_g \in \mathbb{F}_{q^n}$. Then we use that D_n is generated by elements of order 2, relations between generators of D_n and that D_n contains $\mathfrak{S}_n$ to show that $h_g = \pm 1$ and $h_g = h_{g'}$ for all elements g and g' in D_n. Finally we use the recursive construction of summation polynomials to show that one generator of D_n leaves f_{n+1}^R invariant and consequently that D_n leaves f_{n+1}^R invariant.

Proof. The summation polynomials are irreducible hence f_{n+1}^R too and $\left\langle f_{n+1}^R \right\rangle = \sqrt{\left\langle f_{n+1}^R \right\rangle}$. The solutions of f_{n+1}^R are invariant by action of D_n thus for all $g \in D_n$, $g \cdot f_{n+1}^R$ vanishes in all solutions of f_{n+1}^R. Consequently for all $g \in D_n$, $g \cdot f_{n+1}^R \in \left\langle f_{n+1}^R \right\rangle$ and so $g \cdot f_{n+1}^R = h_g \cdot f_{n+1}^R$ where $h_g \in \mathbb{F}_{q^n}[c_1,\dots,c_n]$. The group D_n is a linear group hence for all $g \in D_n$, $\deg\left(g \cdot f_{n+1}^R\right) = \deg\left(f_{n+1}^R\right)$ thus $h_g \in \mathbb{F}_{q^n}$.

Let g be an element of D_n and m its order, we have $f_{n+1}^R = g^m \cdot f_{n+1}^R = h_g^m \cdot f_{n+1}^R$. hence $h_g^m = 1$.

We note $\tau_{i,j}$ the transposition witch swaps the elements in position i and j. Let $\mathscr{B} = \{\tau_{i,i+1} \mid i = 1,\dots,n-1\}$ be a basis of $\mathfrak{S}_n$. Let $\mathscr{C} = \{\tau_{i,i+1} \cdot \tau_{i-1,i} = (i-1,i,i+1) \mid i = 2,\dots,n-1\}$ be a set of cycles of order 3. A transposition is of order two, hence $h_{\tau_{i,j}} = \pm 1$ for any i, j. Moreover each left composition of two successive transpositions (elements of $\mathscr{C}$) is of order 3 thus $h_\alpha = h_\beta$ for all $\alpha, \beta \in \mathscr{B}$.

We now show, by induction, that f_m is invariant under the permutation $\tau_{1,2}$. Clearly (see Section 2.3), f_3 is invariant under $\tau_{1,2}$. Let $k > 2$, assume that f_k is invariant under $\tau_{1,2}$. We have

$$\begin{aligned} f_{k+1} &= \mathrm{Res}_X\Big(f_k(c_1,\dots,c_{k-1},X), f_3(c_k,c_{k+1},X)\Big) \\ &= \mathrm{Det}\Big(\mathrm{Syl}_X\big(f_k(c_1,\dots,c_{k-1},X), f_3(c_k,c_{k+1},X)\big)\Big) \end{aligned}$$

where $\mathrm{Syl}_X(p_1,p_2)$ is the Sylvester matrix of p_1 and p_2 w.r.t. the variable X. Clearly the Sylvester matrix of $f_k(c_1,\dots,c_{k-1},X)$ and $f_3(c_k,c_{k+1},X)$ w.r.t. X is

stable by permutation of c_1 and c_2 (induction hypothesis). Hence its determinant too and f_{k+1} also. Consequently, f_m is invariant under $\tau_{1,2}$ for all $m \geq 3$. Thus f_{n+1}^R is invariant under $\tau_{1,2}$ and $h_\alpha = 1$ for all $\alpha \in \mathscr{B}$. This confirms that the summation polynomials are symmetric.

A basis of D_n is given by $\mathscr{A} = \mathscr{B} \cup (-1,-2)$ where $(-1,-2)$ denotes the sign changes of the first two elements. The element $(-1,-2)$ is of order 2 hence $h_{(-1,-2)} = \pm 1$. Let $g = (-1,-2) \cdot \tau_{2,3} \cdot \tau_{1,2}$, g is of order 3 thus $h_g^3 = 1 = (h_{\tau_{1,2}} \cdot h_{\tau_{2,3}} \cdot h_{(-1,-2)})^3 = h_{(-1,-2)}^3$. Consequently for all elements g in $\mathscr{A}$, $h_g = 1$ and so f_{n+1}^R is invariant under D_n. □

As previously announced in Section 3, $\mathbb{F}_{q^n}[c_1,\ldots,c_n]^{D_n}$ is a polynomial algebra of basis $\{s_1,\ldots,s_{n-1},e_n\}$ (respectively $\{p_2,\ldots,p_{2(n-1)},e_n\}$). Hence there exists a unique polynomial $g_n^R \in \mathbb{F}_{q^n}[s_1,\ldots,s_{n-1},e_n]$ (respectively $\mathbb{F}_{q^n}[p_2,\ldots,p_{2(n-1)},e_n]$) such that g_n^R is the expression of f_{n+1}^R in terms of $\{s_1,\ldots,s_{n-1},e_n\}$ (respectively $\{p_2,\ldots,p_{2(n-1)},e_n\}$). By applying a Weil restriction on g_n^R we obtain a new system $\mathscr{S}_{D_n} \subset \mathbb{F}_q[s_1,\ldots,s_{n-1},e_n]$ (respectively $\mathbb{F}_q[p_2,\ldots,p_{2(n-1)},e_n]$) with n variables and n equations. The degree of $\langle \mathscr{S}_{D_n} \rangle$ is

$$\frac{\deg(\langle \mathscr{S} \rangle)}{\#D_n} = \frac{\deg(\langle \mathscr{S} \rangle)}{n! \cdot 2^{n-1}} = \frac{\deg(\langle \mathscr{S}_{\mathfrak{S}_n} \rangle)}{2^{n-1}} = \frac{2^{n(n-1)}}{2^{n-1}} = 2^{(n-1)^2}.$$

We have therefore obtained our main theorem.

Theorem 4.1. *In twisted Edwards (respectively twisted Jacobi intersections) representation, the point decomposition problem can be solved in time*

- *(proven complexity)* $O\left(\log(q)\left(d^{\omega n} + n \cdot 2^{3(n-1)^2}\right)\right)$
- *(heuristic complexity)* $O\left(\log(q)\left(d^{\omega n} + n^2 \cdot 2^{\omega(n-1)^2}\right)\right)$

where $2 \leq \omega < 3$ *is the linear algebra constant and* d *is the degree of regularity which is a bound on the maximal degree reached during the computation of Gröbner basis with* F_4, F_5.

Considering the action of the dihedral Coxeter group reduces the lexicographical Gröbner basis – for twisted Edwards and Jacobi intersections curves – which is now in shape lemma.

Lexicographical Gröbner basis of $\mathscr{S}_{\mathfrak{S}_n}$:

$$\begin{cases} e_1 + \mathfrak{p}_1(e_{n-1},e_n) \\ e_2 + \mathfrak{p}_2(e_{n-1},e_n) \\ \vdots \\ e_{n-2} + \mathfrak{p}_{n-2}(e_{n-1},e_n) \\ \mathfrak{p}_{n-1}(e_{n-1},e_n) \\ \mathfrak{p}_n(e_n) \end{cases}$$

Lexicographical Gröbner basis of $\mathscr{S}_{D_n}$:

$$\begin{cases} s_1 + h_1(e_n) \\ s_2 + h_2(e_n) \\ \vdots \\ s_{n-2} + h_{n-2}(e_n) \\ s_{n-1} + h_{n-1}(e_n) \\ h_n(e_n) \end{cases}$$

where

- $\deg(\langle \mathscr{S}_{\mathfrak{S}_n} \rangle) = 2^{n(n-1)}$ and $\deg(\langle \mathscr{S}_{D_n} \rangle) = 2^{(n-1)^2}$

- $\deg_{e_{n-1}}(\mathfrak{p}_{n-1}) = 2^{n-1}$, $\deg(\mathfrak{p}_n) = 2^{(n-1)^2}$ and $\deg(h_n) = 2^{(n-1)^2}$.

As expected the degree of the ideal is divided by the cardinality of D_n, $2^{n-1} \cdot n!$ instead of $n!$ when taking into account only the symmetric group.

In Section 5 we will show some experimental results which confirm that considering the action of the 2-torsion points significantly simplifies the resolution of the PDP.

4.2. **Can we use in the same way the 4-torsion points?** As we saw in Section 2.3 the twisted Edwards and Jacobi intersections curves can also have rational 4-torsion points. The natural question follows, whether 4-torsion points are as useful as 2-torsion points for PDP resolution?

4.2.1. *Action of the 4-torsion points of a twisted Edwards curve.* The two 4-torsion points of a twisted Edwards curve are $T_4 = \left(\pm a^{-\frac{1}{2}}, 0\right)$. Thus if $P = (x,y) \in E_{a,d}(\mathbb{F}_{q^n})$ then we have

$$P \oplus T_4 = \left(\pm a^{-\frac{1}{2}} \cdot y, \pm a^{\frac{1}{2}} \cdot x\right)$$

The sum of P with a 4-torsion point swaps – up to multiplication by $\pm a^{\frac{1}{2}}$ or $\pm a^{-\frac{1}{2}}$ – the coordinates of the point P. As previously showed, to use the 4-torsion points of a twisted Edwards curve, the y-coordinate of $P \oplus T_4$ should depend only of the y-coordinate of P. Consequently the 4-torsion points of twisted Edwards curves can not be used in the same way as we did with 2-torsion points.

4.2.2. *Action of the 4-torsion points of a twisted Jacobi intersections curve.* We concentrate first on the case of the following 4-torsion point:

$$T_4 = \left(\pm\frac{1}{\sqrt{a}}, 0, \pm\sqrt{\frac{a-b}{a}}\right).$$

After a few simplifications, adding T_4 to a generic point $P = (x,y,z)$ of $E_{a,b}(\mathbb{F}_{q^n})$ gives the formula

$$P \oplus T_4 = \left(\pm\frac{1}{\sqrt{a}} \cdot \frac{y}{z}, \pm\sqrt{a-b} \cdot \frac{x}{z}, \pm\sqrt{\frac{a-b}{a}} \cdot \frac{1}{z}\right),$$

where the resulting z-coordinate depends only on z. As seen in Section 2.3, for twisted Jacobi intersections curves, it is possible to use either y or z for projecting the PDP and obtain interesting summation polynomials. To take advantage of the action of T_4, we project on z and work with the summation polynomial f_z.

In order to normalize a bit more the action of T_4, we assume that $\frac{a-b}{a}$ is a fourth power and do the change of coordinate

$$Z = \sqrt[4]{\frac{a}{a-b}}\, z,$$

so that adding T_4 change the Z-coordinate to $\pm 1/Z$. This change of coordinate preserves the property that adding T_2 changes the sign of the Z-coordinate, so that

we still have the action of D_n on f_Z. This explicit action of T_4 transforms a decomposition into another one, but unfortunately, this action is not linear and therefore does not fit easily in the framework that we have developped. As a consequence, we will not be able to reduce the degree of the ideal as much as we could hope for. Still, by adding an additional variable to make the symmetry more visible, we will force a non-shape lemma for the LEX Gröbner basis that had proved to be useful for T_2, before reducing the degree of the ideal.

We explain this strategy in the case of $n = 4$. Adding T_4 to the 4 points of a decomposition gives another decomposition, where all the Z_i have been inverted. We defined a new coordinate v_4 that is invariant by this involution:

$$v_4 = Z_1Z_2Z_3Z_4 + \frac{1}{Z_1Z_2Z_3Z_4} = e_4(Z_1,Z_2,Z_3,Z_4) + \frac{1}{e_4(Z_1,Z_2,Z_3,Z_4)}.$$

Therefore, we add the equation $e_4v_4 - e_4^2 - 1 = 0$ to the system obtained by applying a Weil restriction on g_4 (the expression of $f^R_{Z,5}$ in terms of s_1,s_2,s_3,e_4). The corresponding LEX Gröbner basis has the following form:

$$\begin{cases} s_1 + \ell_1(e_4,v_4) \\ s_2 + \ell_2(e_4,v_4) \\ s_3 + \ell_3(e_4,v_4) \\ e_4v_4 - e_4^2 - 1 \\ \ell_4(v_4) \end{cases}$$

where $\deg(\ell_i) = 2^{n(n-2)}$ for all $i = 1,\ldots,4$ and the degree of the ideal remains $2^{(n-1)^2}$ as when using only T_2.

Remark 3. *For $n > 4$, the variable v_4 must be replaced by a variable that is invariant by any change of a multiple of four number of variables by their inverses. For instances, one can use $v_n = \sum_{i=1}^{n}\left(Z_i^2 + \frac{1}{Z_i^2}\right) = s_1 + \frac{s_{n-1}}{e_n^2}$. For $n = 4$, the aforementioned variable $v_4 = e_4 + \frac{1}{e_4}$ gives better results in practice.*

The construction that we have just shown works mutatis mutandis with the other 4-torsion points of the form

$$T_4 = \left(\pm\frac{1}{\sqrt{b}}, \pm\sqrt{\frac{b-a}{b}}, 0\right),$$

but in that case, we have to work with the y coordinate instead of the z coordinate.

From the parameters of the system, it is not obvious that adding a variable to reduce the degree of the polynomials in the resulting Gröbner basis is worthwhile. We will see in the next section that for some parameters n and q, it is indeed the best choice to use the action of T_4.

5. Experimental results and security estimates

All experiments or comparisons in this section assume that the elliptic curve is a twisted Edwards or twisted Jacobi intersections curve. We recall that only curves

with a particular torsion structure can be put into these forms and are subject to our improved attack.

The PDP problem for $n = 2$ is not interesting, since it does not yield an attack that is faster than the generic ones. For $n = 3$, the PDP problem can be solved very quickly, so that our improvements using symmetries are difficult to measure. Therefore, we will concentrate on the $n = 4$ and higher cases.

Most of our experiments are done with MAGMA, which provides an easy-to-reproduce environment (the MAGMA codes to solve the PDP are available at `http://www-polsys.lip6.fr/~huot/CodesPDP`). For the largest computations, we used the FGb library which is more efficient for systems of the type encountered in the context of this paper. FGb also provides a precise count of the number of basic operations (a multiplication of two 32-bit numbers is taken as unit) that are required in a system resolution. We will use this information to interpolate security levels for large inputs.

5.1. **Experiments with $n = 4$.** In the case of $n = 4$, the resolution is still fast enough so that the "$n-1$" approach by Joux-Vitse does not pay. So we compare the three following approaches: the classical index-calculus of [29] based on Weierstrass representation (denoted W. [29], in the following) and our approaches using the 2-torsion point (denoted T_2) and using additionally the 4-torsion point (denoted $T_{2,4}$). For T_2 and $T_{2,4}$, we have implemented the two choices for the basis of the invariant ring for the dihedral Coxeter group given in Section 3.2, that we denote by s_i and p_i. The results are given in table 1, where one finds the runtimes for various sizes of the base field.

$\log_2(q)$		F_4 (s)		Change-Order (s)		Total (s)		# ops	
		s_i	p_i	s_i	p_i	s_i	p_i	s_i	p_i
16	W. [29]	4		531		535		2^{29}	
	T_2	0	0	22	3	22	**3**	2^{23}	2^{26}
	$T_{2,4}$	0	1	5	3	5	4	2^{25}	2^{27}
64	W. [29]	354		4363		4717		2^{31}	
	T_2	4	23	80	18	84	41	2^{25}	2^{28}
	$T_{2,4}$	13	42	25	17	**38**	59	2^{27}	2^{29}
128	W. [29]	532		5305		5837		2^{32}	
	T_2	4	31	98	23	102	54	2^{26}	2^{29}
	$T_{2,4}$	17	62	29	23	**46**	85	2^{28}	2^{30}

TABLE 1. Computing time of Gröbner basis with MAGMA (V2-17.1) on one core of a 2.93 GHz Intel® E7220 CPU for $n = 4$. The last column (number of operations) is based on FGb.

We can observe that taking into account the symmetries, dramatically decreases the computing time of the PDP resolution, by a factor of about 100. These experiments also show that the choice of the invariant ring basis s_i or p_i for the dihedral

Coxeter group is not computationally equivalent. Indeed, the degrees of the polynomials depend on it: it is 8 for the s_i basis and 12 with the p_i. As a consequence, the DRL part of the computation is more costly for the p_i than for the s_i. On the other hand, the FGLM step if more efficient for the p_i than for the s_i.

All in all, the best trade-off for $n = 4$ with MAGMA is to use the $T_{2,4}$ approach with the s_i basis. Adding the v_4 variable increases the cost of the DRL computation, but this is compensated by the fact the polynomial degrees in the LEX Gröbner basis decrease, and thus the cost of the change of ordering decreases too.

5.2. **Experiments for $n = 5$ and $n = 6$.** Until now, the only viable approach for handling the cases where n is at least 5 was the approach by Joux and Vitse [33] which is reminiscent of the hybrid approach of Bettale, Faugère and Perret in [6] where one mixes an exhaustive search and an algebraic resolution. If one looks for a decomposition of a given point R, instead of searching for n points of the factor base whose sum is equal to R, one can search for only $n-1$ points of the factor base whose sum is equal to R. Using this technique simplifies the resolution of the polynomial systems, since we manipulate the summation polynomial of degree n instead of $n+1$ so that the degree and the number of variables are reduced. Furthermore the systems become overdetermined and if they have a solution, then in general it is unique. Hence the DRL Gröbner basis is also the LEX Gröbner basis and we do not need the FGLM step in the general solving strategy. On the other hand, it decreases the probability of finding a decomposition by a factor q/n.

One of the main improvement brought by this work, is that we are now able to solve the polynomial systems coming from the summation polynomials for $n = 5$ when the symmetries are used. Still, these computation are not feasible with MAGMA and we use the FGb library. The timings are given in table 2.

$\log_2(q)$		F_5 (s)		Change-Order (s)		Total (s)		# ops
		s_i	p_i	s_i	p_i	s_i	p_i	s_i
16	W. [29]	> 2 days						
	T_2	12297	30406	7866	14465	**20163**	44871	2^{45}

TABLE 2. Computing time of Gröbner basis with FGb on a 3.47 GHz Intel® X5677 CPU for $n = 5$.

It can be observed that with FGb, the two steps of the resolution are faster with the s_i basis. This is a general practical fact observed during our experiments. Thus, in the sequel, for computation with FGb, we consider only the s_i basis.

Our improved algorithm using symmetries can be combined with the "$n-1$" approach of Joux and Vitse. This allows us to compare the running times with the approach taken in [33] in the case of $n = 5$, and to handle, for the first time, the case of $n = 6$. The results are summarized in tables 3 and 4. For $n = 6$, MAGMA was not able to solve the system, so we used again FGb. Because of the low success

$\log_2(q)$		F_4 (s)		# ops
		s_i	p_i	s_i
16	W. [33]	8.890		2^{32}
	T_2	**0.070**	0.160	2^{25}
	$T_{2,4}$	0.110	0.890	2^{23}
32	W. [33]	1501.680		2^{33}
	T_2	**3.020**	5.200	2^{26}
	$T_{2,4}$	4.940	48.060	2^{24}

TABLE 3. Computing time of DRL Gröbner basis with MAGMA (V2-17.1) on a 2.93 GHz Intel® E7220 CPU for $n = 5$ and decomposition in $n-1$ points. Operation counts are obtained using FGb.

$\log_2(q)$		F_5 (s)	# ops
		s_i	s_i
16	W. [33]	> 2 days	
	T_2	11188	2^{44}

TABLE 4. Computing time of DRL Gröbner basis with FGb on a 3.47 GHz Intel® X5677 CPU for $n = 6$ and decomposition in $n-1$ points.

probability, this technique is interesting only for small q. Hence, we limit the size of q to 32 bits, and even to 16 bits for $n = 6$.

Using symmetries decreases the running time also for decompositions in $n-1$ points. For $n = 5$, the speed-up is by a factor about 100 for a 16-bit base field and by 500 for a 32-bit base field. For $n = 6$, without using the symmetries of twisted Edwards or twisted Jacobi intersections curves, we can not compute decompositions in $n-1$ points while this work allows to compute them in approximately three hours.

Remark 4. *The computing time for twisted Jacobi intersections curves when using the symmetries induced by the* 4*-torsion point is higher because we do not have only one solution. Actually, the curve has more symmetries than those considered but we have not found a way to use them all.*

Remark 5. *For $n \geq 6$, the first difficulty to solve the PDP is the construction of the summation polynomials. Actually, the seventh summation polynomial or the seventh summation polynomial evaluated in the c-coordinate of a point R have never been computed.*

5.3. **Security level estimates.** To conclude these experimental results, we use our new operation counts for the PDP to estimate the cost of a complete resolution of the ECDLP for twisted Edwards or twisted Jacobi intersections curves. In this section, we count only arithmetic operations and we neglect communications and memory occupation. Hence, this do not give an approximation of the computation

time but this gives a first approximation of the cost to solve some instances of the ECDLP.

We compare the result with all previously known attacks, including the generic algorithms, whose complexity is about $q^{\frac{n}{2}}$ operations in $E(\mathbb{F}_{q^n})$. Since our cost unit for boolean operations is a 32-bit integer multiplication, we roughly approximate the cost of an elliptic curve operation by $n^2\log_{2^{32}}(q)^2$ and the total boolean cost of a generic attack by

$$(GA) \qquad n^2 q^{\frac{n}{2}} \log_{2^{32}}(q)^2.$$

For index calculus using the point decomposition in n points we look for $\frac{q}{2}$ relations. Indeed, as suggested in [29] we can divide the size of the factor base by a factor 2 by keeping either a point P or its negative $\ominus P$ instead of both. The probability to decompose a point is $\frac{1}{n!}$. Let $c(n,q,m)$ be the number of boolean operations needed to solve one polynomial system obtained from a Weil restriction of the $(m+1)^{\text{th}}$ summation polynomial defined over $\mathbb{F}_{q^n}$, evaluated in one variable. This number of operations is obtained by experiments with FGb as demonstrated in the previous subsections. From the function $c(n,q,m)$ one can deduce the total number of operations needed to solve the ECDLP over $\mathbb{F}_{q^n}$:

$$(RS(n)+LA) \qquad \frac{q\cdot n!}{2}\cdot c(n,q,n)+\log_2(q)\cdot\frac{q^2}{4}.$$

The second term in the sum is the cost of sparse linear algebra by using for instance Wiedemann algorithm [50].

If we use the point decomposition in $n-1$ points, due to exhaustive search, the probability to find a decomposition is now $\frac{1}{q\cdot(n-1)!}$. Hence the total number of operations is, in this case, given by

$$(RS(n-1)+LA) \qquad \frac{q^2\cdot(n-1)!}{2}\cdot c(n,q,n-1)+\log_2(q)\cdot\frac{q^2}{4}.$$

When the linear algebra step is more time consuming than the relation search, by using the double large prime variation [31] (denoted hereafter DLPV) we can rebalance the costs of these two steps (see [48, 31]). The total number of operations needed to solve the ECDLP over $\mathbb{F}_{q^n}$ by using the double large prime variation is given by:

$$(DLPV) \quad \log_2(q)\left(1+r\frac{n-1}{n}\right)(n-2)!q^{1+(n-2)(1-r)}c(n,q,n)+\log_2(q)\frac{q^{2r}}{4}$$

where we look for r such that the two parts of this complexity are equal.

The results are summarized in table 5, where, as in Section 5.1, W denotes Weierstrass representation which corresponds for $\mathrm{RS}(n)$ to the initial index calculus of Gaudry and for $\mathrm{RS}(n-1)$ that corresponds to Joux and Vitse work. The notation T_2 and $T_{2,4}$ still denote the use of the 2-torsion points of twisted Edwards and twisted Jacobi intersections curves and the use of the 2-torsion and 4-torsion points of twisted Jacobi intersections curves respectively.

We observe that the smallest number of operations obtained for each parameters is given by index calculus using symmetries induced by the 2-torsion points or generic algorithms. We note that for $n \leq 5$ our version of the index calculus

n	$\log_2(q)$	$\#E(\mathbb{F}_{q^n})$		GA	LA	RS(n)	RS($n-1$)	DLPV
4	32	2^{128}	W. T_2 $T_{2,4}$	2^{68}	2^{67}	2^{66} [29] 2^{60} 2^{62}		**2^{65}** 2^{66}
	64	2^{256}	W. T_2 $T_{2,4}$	2^{134}	2^{132}	2^{99} [29] 2^{93} 2^{95}		2^{117} **2^{114}** 2^{115}
	128	2^{512}	W. T_2 $T_{2,4}$	2^{264}	2^{261}	2^{164} [29] 2^{158} 2^{160}		2^{214} **2^{211}** 2^{212}
5	32	2^{160}	W. T_2 $T_{2,4}$	2^{85}	2^{67}	**2^{84}**	2^{100} [33] 2^{94} 2^{91}	
	64	2^{320}	W. T_2 $T_{2,4}$	2^{167}	2^{132}	2^{117}	2^{165} [33] 2^{159} 2^{156}	**2^{127}**
	128	2^{640}	W. T_2 $T_{2,4}$	2^{329}	2^{261}	2^{182}	2^{294} [33] 2^{288} 2^{285}	**2^{231}**
6	32	2^{192}	T_2	**2^{102}**	2^{67}		2^{115}	
	64	2^{384}	T_2	2^{200}	2^{132}		**2^{180}**	
	128	2^{768}	T_2	2^{394}	2^{261}		**2^{309}**	

TABLE 5. Number of operations needed to solve the ECDLP defined over $\mathbb{F}_{q^n}$ for $n = 4, 5, 6$ and $32 \leq \log_2(q) \leq 128$.

attack is better than generic algorithms. For example, if $\log_2(q) = 64$ and $n = 4$ generic algorithms need 2^{134} operations to attack the ECDLP and we obtain 2^{114} by using the 2-torsion points. In this case, our approach is more efficient the basic index calculus, solving this instance of ECDLP in 2^{117} operations. For $n = 5$, the resolution of the PDP was intractable but with our method, we can now solve these instances of PDP and we attack the corresponding instances of ECDLP with a gain of 2^{40} over generic algorithms and a gain of 2^{38} over Joux and Vitse approach.

We remark that for parameters for which it is possible to choose between the decomposition in n or $n-1$ points, the best solution is the first. For $n = 6$ we are not able to decompose a point in n points of the factor base. Consequently it is necessary to use the decomposition in $n-1$ points. For $n = 6$ generic algorithms have a complexity in $O\left(q^3\right)$, while index calculus attack using the decomposition in $n-1$ points has a complexity in $O\left(\mathrm{C}(n) \cdot \log_2(q) \cdot q^2\right)$ where $\mathrm{C}(n)$ is exponentially in n. Hence to be better than generic algorithms, we have to consider high value of q and consequently high security level. For instance if $\log_2(q) = 64$, index calculus attack using symmetries of twisted Edwards or twisted Jacobi intersections curves

and decomposition in $n-1$ points needs less operations (2^{180}) than generic algorithms, (2^{200}). In our point of view the only hope to have a better gain in general (for lower security level) compared to generic algorithms, would be to remove the bad dependence in q in the complexity that seems intrinsic to the "$n-1$" approach

Remark 6. *If $n=6$, it is possible to decrease the number of operations by using a recent result from Joux and Vitse* [34]. *The idea is to use a GHS attack* [30] *to transfer the ECDLP over $\mathbb{F}_{p^6}$ to the HCDLP over $\mathbb{F}_{p^2}$ with a curve of genus three and then using an index calculus attack for hyperelliptic curves. Contrary to our attack valid for most curves, this attack is possible only for few particular curves.*

In cryptology, one looks for parameters giving some user-prescribed security level. Thereafter we give the domains parameters for different security level expressed in number of boolean operations.

Security level	2^{80}						2^{96}					
n	4		5		6		4		5		6	
$\log_2(q)$	38	**43**	**31**	29	**26**	16	46	**53**	37	**44**	**31**	23
Security level	2^{112}						2^{128}					
n	4		5		6		4		5		6	
$\log_2(q)$	54	**63**	43	**56**	**36**	31	62	**74**	49	**65**	**41**	39
Security level	2^{192}						2^{256}					
n	4		5		6		4		5		6	
$\log_2(q)$	93	**116**	74	**105**	62	**70**	125	**158**	100	**144**	83	**102**

TABLE 6. Domains parameters according to the security level given in number of boolean operations needed to solve the ECDLP.

In table 6, we compare for a fixed security level the size of q that we have to choose for $n=4,5,6$ by considering attack based on generic algorithms (left column) with attack based on the best version of index calculus attack (right column). For the index calculus attack, except for $n=6$, the size of q is obtained by considering decomposition in n points using the symmetries of twisted Edwards and Jacobi intersections curves. This table confirms the previous observations. For $n=4,5$, the size of q is increased because the new version of index calculus proposed in this work. For $n=6$ this is true only for very high security level.

6. CONCLUSION

We have enlightened some geometrical properties of twisted Edwards and Jacobi intersections curves implying new symmetries simplifying the resolution of the *Point Decomposition Problem*. In the same way that one can reduce the size of the factor base by 2 by using the action of $\ominus$, the adding symmetries of twisted

Edwards or Jacobi intersections curves allow to reduce the size of the factor base again by 2 (we keep either P or $P+T_2$). This decreases the cost of the linear algebra in the last part of the index calculus attack by a factor 4 and the relations search step by a factor 2. However, this improvement applies to only particular instances of ECDLP defined over a finite field of characteristic different from two. Using symmetries to improve some instances of ECDLP in characteristic two is more difficult. Actually, when the characteristic of the based field divides the order of the linear group acting on the polynomial system to solve, the invariant theory cannot be applied in the same way as done here. This is in general the case when the characteristic is two. Thus, even if we note some symmetries in characteristic two, it is still an open issue to prove same results in this case as the ones we provide in this paper.

In order to solve the PDP, we construct the $(n+1)^{\text{th}}$ summation polynomials. However, in practice, one can effectively compute the m^{th} summation polynomials up to $m=6$ only. Hence, without exhaustive search, one can use the index calculus attack only for elliptic curves defined over $\mathbb{F}_{q^n}$ with $n<6$. Thus to more improve the PDP resolution, a question remains: how *good* polynomial systems modeling the PDP for $n\geq 6$ can be constructed efficiently? Where *good* means here a polynomial system with a comparable resolution complexity as the one given in Theorem 4.1.

Finally, as we study only instances of ECDLP, a natural question follows: in the same way, by using symmetries, is it possible to increase the efficiency of the resolution of some instances of HCDLP for genus two curves?

REFERENCES

[1] L. Adleman and J. DeMarrais. A subexponential algorithm for discrete logarithms over all finite fields. In *Advances in Cryptology—CRYPTO'93*, pages 147–158. Springer, 1994. (Cited on page 1.)

[2] L.M. Adleman, J. DeMarrais, and M.-D. Huang. A subexponential algorithm for discrete logarithms over the rational subgroup of the jacobians of large genus hyper- elliptic curves over finite fields. In *Algorithmic Number Theory*, volume 877 of *Lecture Notes in Comput. Sci.* Springer–Verlag, 1994. 6th International Symposium. (Cited on page 2.)

[3] M. Bardet, J.C Faugère, and B. Salvy. Asymptotic expansion of the degree of regularity for semi-regular systems of equations. In P. Gianni, editor, *The Effective Methods in Algebraic Geometry Conference, Mega 2005*, pages 1 –14, May 2005. (Cited on page 10.)

[4] D. J. Bernstein and T. Lange. Faster addition and doubling on elliptic curves. In *Advances in Cryptology : ASIACRYPT 2007*, volume 4833 of *Lecture Notes in Computer Science*, pages 29–50. Springer, 2007. (Cited on pages 2 and 7.)

[5] D.J. Bernstein, P. Birkner, M. Joye, T. Lange, and C. Peters. Twisted edwards curves. In *Proceedings of the Cryptology in Africa 1st international conference on Progress in cryptology*, AFRICACRYPT'08, pages 389–405, Berlin, Heidelberg, 2008. Springer-Verlag. (Cited on pages 2 and 7.)

[6] L. Bettale, J.-C. Faugère, and L. Perret. Hybrid approach: a tool for multivariate cryptography. In *Proceedings of the ECRYPT Workshop on Tools for Cryptanalysis 2010*, pages 15–23. ECRYPT II, 2010. (Cited on page 23.)

[7] W. Bosma, J. Cannon, and C. Playoust. The Magma algebra system. I. The user language. *J-SYMBOLIC-COMP*, 24(3–4):235–265, 1997. (Cited on page 4.)

[8] C. Chevalley. Invariants of finite groups generated by reflections. *American Journal of Mathematics*, 77(4):pp. 778–782, 1955. (Cited on page 12.)

[9] D.V. Chudnovsky and G.V. Chudnovsky. Sequences of numbers generated by addition in formal groups and new primality and factorization tests. *Advances in Applied Mathematics*, 7(4):385–434, 1986. (Cited on pages 2 and 8.)

[10] A.M. Cohen, H. Cuypers, and H. Sterk. *Some Tapas of Computer Algebra*. Algorithms and Computation in Mathematics Series. Springer, 2011. (Cited on page 16.)

[11] J.-M. Couveignes. Algebraic groups and discrete logarithm. In *Public-key cryptography and computational number theory*, pages 17–27, 2001. (Cited on page 2.)

[12] J.-M. Couveignes and R. Lercier. Galois invariant smoothness basis. *Series on Number Theory and Its Applications*, 5:142–167, May 2008. World Scientific. (Cited on page 3.)

[13] C. Diem. An index calculus algorithm for plane curves of small degree. In *Algorithmic number theory ANTS-VII*, volume 4076 of *Lecture Notes in Computer Science*, pages 543–557. Springer, 2006. (Cited on page 2.)

[14] C. Diem. On the discrete logarithm problem in class groups of curves. *Math. Comp*, 80:443–475, 2011. (Cited on page 2.)

[15] C. Diem. On the discrete logarithm problem in elliptic curves. *Compositio Mathematica*, 147:75–104, 2011. (Cited on page 2.)

[16] C. Diem and E. Thomé. Index calculus in class groups of non-hyperelliptic curves of genus three. *Journal of Cryptology*, 21(4):593–611, 2008. (Cited on page 2.)

[17] H.M. Edwards. A normal form for elliptic curves. In *Bulletin of the American Mathematical Society*, volume 44, pages 393–422, July 2007. (Cited on pages 2 and 7.)

[18] A. Enge and P. Gaudry. A general framework for subexponential discrete logarithm algorithms. *Acta Arith*, 102(1):83–103, 2002. (Cited on page 2.)

[19] A. Enge and P. Gaudry. An L (1/3+ ε) algorithm for the discrete logarithm problem for low degree curves. In *Advances in Cryptology-EUROCRYPT 2007*, pages 379–393. Springer, 2007. (Cited on page 2.)

[20] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases (F4). *Journal of Pure and Applied Algebra*, 139(1–3):61–88, June 1999. (Cited on pages 3 and 10.)

[21] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). In *Proceedings of the 2002 international symposium on Symbolic and algebraic computation*, ISSAC '02, pages 75–83, New York, NY, USA, 2002. ACM. (Cited on pages 3 and 10.)

[22] J.-C. Faugère. FGb: A Library for Computing Gröbner Bases. In Komei Fukuda, Joris Hoeven, Michael Joswig, and Nobuki Takayama, editors, *Mathematical Software - ICMS 2010*, volume 6327 of *Lecture Notes in Computer Science*, pages 84–87, Berlin, Heidelberg, September 2010. Springer Berlin / Heidelberg. (Cited on page 4.)

[23] J.-C. Faugère, P. Gaudry, L. Huot, and G. Renault. Fast change of ordering with exponent ω, 2012. Available at `http://www-polsys.lip6.fr/~huot/unpublished/orderChange.pdf`. (Cited on pages 4, 10, and 11.)

[24] J.-C. Faugère, P. Gianni, D. Lazard, and T. Mora. Efficient Computation of Zero-dimensional Gröbner Bases by Change of Ordering. *Journal of Symbolic Computation*, 16(4):329–344, 1993. (Cited on pages 3, 4, and 10.)

[25] J.-C. Faugère and C. Mou. Fast Algorithm for Change of Ordering of Zero-dimensional Gröbner Bases with Sparse Multiplication Matrices. In *ISSAC '11: Proceedings of the 2011 international symposium on Symbolic and algebraic computation*, ISSAC '11, pages 1–8, New York, NY, USA, 2011. ACM. (Cited on pages 3, 4, and 10.)

[26] J.-C. Faugère and S. Rahmany. Solving systems of polynomial equations with symmetries using SAGBI-Gröbner bases. In *ISSAC '09: Proceedings of the 2009 international symposium on Symbolic and algebraic computation*, ISSAC '09, pages 151–158, New York, NY, USA, 2009. ACM. (Cited on page 13.)

[27] R. Feng, M. Nie, and H. Wu. Twisted Jacobi intersections curves. *Theory and Applications of Models of Computation*, pages 199–210, 2010. (Cited on pages 2 and 8.)

[28] G. Frey. Applications of arithmetical geometry to cryptographic constructions. In *International Conference on Finite Fields and Applications*, pages 128–161, 2001. (Cited on page 5.)

[29] P. Gaudry. Index calculus for abelian varieties of small dimension and the elliptic curve discrete logarithm problem. *Journal of Symbolic Computation*, 44(12):1690–1702, 2009. (Cited on pages 2, 15, 22, 23, 25, and 26.)

[30] P. Gaudry, F. Hess, and N. Smart. Constructive and destructive facets of Weil descent on elliptic curves. *Journal of Cryptology*, 15:19–46, 2002. (Cited on page 27.)

[31] P. Gaudry, E. Thomé, N. Thériault, and C. Diem. A double large prime variation for small genus hyperelliptic index calculus. *Mathematics of Computation*, 76:475–492, 2007. (Cited on pages 2 and 25.)

[32] F. Hess. Computing relations in divisor class groups of algebraic curves over finite fields. Preprint, 2004. (Cited on page 2.)

[33] A. Joux and V. Vitse. Elliptic Curve Discrete Logarithm Problem over Small Degree Extension Fields. Application to the static Diffie-Hellman problem on $E(\mathbb{F}_{q^5})$. Cryptology ePrint Archive, Report 2010/157, 2010. `http://eprint.iacr.org/` To appear in Journal of Cryptology, Springer, DOI: 10.1007/s00145-011-9116-z. (Cited on pages 5, 23, 24, and 26.)

[34] A. Joux and V. Vitse. Cover and Decomposition Index Calculus on Elliptic Curves made practical. Application to a seemingly secure curve over $\mathbb{F}_{p^6}$. Cryptology ePrint Archive, Report 2011/020, 2011. `http://eprint.iacr.org/` Accepted at Eurocrypt 2012. (Cited on page 27.)

[35] R. Kane. *Reflection Groups and Invariant Theory*. Springer, 2001. (Cited on page 4.)

[36] N. Koblitz. Elliptic curve cryptosystems. *Math. Comp.*, 48(177):203–209, 1987. (Cited on page 2.)

[37] N. Koblitz. Hyperelliptic cryptosystems. *Journal of Cryptology*, 1:139–150, 1989. (Cited on page 2.)

[38] D. Lazard. Solving zero-dimensional algebraic systems. *Journal of Symbolic Computation*, 13(2):117 – 131, 1992. (Cited on page 10.)

[39] V.S. Miller. The weil pairing, and its efficient calculation. *J. Cryptology*, 17(4):235–261, 2004. (Cited on page 2.)

[40] P.L. Montgomery. Speeding the Pollard and elliptic curve methods of factorization. *Mathematics of computation*, 48(177):243–264, 1987. (Cited on page 7.)

[41] K. Nagao. Decomposed attack for the jacobian of a hyperelliptic curve over an extension field. In Guillaume Hanrot, François Morain, and Emmanuel Thomé, editors, *Algorithmic Number Theory*, volume 6197 of *Lecture Notes in Comput. Sci.* Springer–Verlag, 2010. 9th International Symposium, Nancy, France, ANTS-IX, July 19-23, 2010, Proceedings. (Cited on page 3.)

[42] National Institute of Standards and Technology. Digital signature standard (dss). Technical Report FIPS PUB 186-3, U.S. Department of Commerce, June 2009. (Cited on page 5.)

[43] J.M. Pollard. Monte carlo methods for index computation mod p. *Math. Comp.*, 32(143):918–924, July 1978. (Cited on page 1.)

[44] I. Semaev. Summation polynomials and the discrete logarithm problem on elliptic curves. Cryptology ePrint Archive, Report 2004/031, 2004. `http://eprint.iacr.org/`. (Cited on pages 3 and 6.)

[45] G. C. Shephard and J. A. Todd. Finite unitary reflection groups. *Canadian J. Math.*, 6:274–304, 1954. (Cited on page 12.)

[46] V. Shoup. Lower bounds for discrete logarithms and related problems. In *Proceedings of the 16th annual international conference on Theory and application of cryptographic techniques*, pages 256–266. Springer-Verlag, 1997. (Cited on page 1.)

[47] B. Sturmfels. *Algorithms in Invariant Theory (Texts and Monographs in Symbolic Computation)*. Springer Publishing Company, Incorporated, 2nd ed.; vii, 197 pp.; 5 figs. edition, 2008. (Cited on pages 4 and 14.)

[48] N. Thériault. Index calculus attack for hyperelliptic curves of small genus. In *Advances in Cryptology : ASIACRYPT 2003*, volume 2894 of *Lecture Notes in Computer Science*, pages 75–92, 2003. (Cited on page 25.)

[49] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, 2002. (Cited on page 10.)

[50] D.H. Wiedemann. Solving sparse linear equations over finite fields. *IEEE Trans. Inf. Theor.*, 32(1):54–62, 1986. (Cited on page 25.)

E-mail address: {Jean-Charles.Faugere, Louise.Huot, Guenael.Renault}@lip6.fr, Pierrick.Gaudry@loria.fr

1 : UPMC-LIP6, BP 169, 4 PLACE JUSSIEU 75252 PARIS CEDEX 05, FRANCE

2 : LORIA-CARAMEL, BP 239, 54506 VANDOEUVRE-LÈS-NANCY, FRANCE