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†, PIERRICK GAUDRY‡, LOUISE HUOT†, AND GUÉNAËL RENAULT†

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 $2^{\omega(n-1)}$ to solve the corresponding PDP where ω is the exponent in the complexity of multiplying two dense matrices. 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 $\sim 2^{160}$ operations in $E(\mathbb{F}_{q^5})$ using generic algorithms compared to 2^{130} operations (multiplications of two 32-bits words) with our method. For these parameters the PDP is intractable with the original algorithm.

The main tool to achieve these results relies on the use of the symmetries and the quasi-homogeneous structure induced by these symmetries during the polynomial system solving step. Also, we use a recent work on a new algorithm 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

$$h = \underbrace{g * \cdots * g}_{x \text{ times}}.$$

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

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

† PolSys project INRIA Paris-Rocquencourt; UPMC Paris 06; CNRS, UMR 7606; LIP6 .

‡ CARAMEL project INRIA Grand-Est; Université de Lorraine; CNRS, UMR 7503; LORIA .

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

This work was partly supported by the HPAC grant of the French National Research Agency (HPAC ANR-11-BS02-013).

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.

A major application of the DLP is to design cryptographic protocols whose 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 Miller [39] and Koblitz [36] independently 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 [5, 4, 18] and twisted Jacobi intersections [9, 29] 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, 19, 20, 33] have been developed for various classes of high genus curves. Using the double large prime variation of Gaudry, Thomé, Thériault and Diem [32], 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 [14, 17]. 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 of larger dimension over the smaller field. In [31], an index calculus attack suited to this context was proposed. Later on, Diem [16, 15] 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 [31] 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 of prime order and Q, two points of $E(\mathbb{F}_{q^n})$ in Weierstrass representation, we look for an integer X, if it exists, such that $Q = [X]P$ (where the notation $[m]P$ denotes, as usual, the scalar multiplication of P by m).

Step 1: First we compute the factor base $\mathscr{F} = \{(x,y) \in E(\mathbb{F}_{q^n}) \mid x \in \mathbb{F}_q\}$.

Step 2: Then we look for $\#\mathscr{F}+1$ relations ($\#\mathscr{F}$ independent relations and any other) of the form

$$[a_j]P \oplus [b_j]Q = P_1 \oplus \cdots \oplus P_n\,, \tag{1}$$

where $P_1, \cdots, P_n \in \mathscr{F}$ and a_j and b_j are randomly picked up in $\mathbb{Z}$.

Step 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 $B = \sum_j \lambda_j \cdot b_j$.

Our study starts from this algorithm. Thus, we assume the same two hypotheses as in [31].

Hypothesis 1. *There exist approximately $\frac{q^n}{n!}$ points of $E(\mathbb{F}_{q^n})$ which can be decomposed as the sum of n points in $\mathscr{F}$. Thus each relation of Step 2 can be found with probability $\frac{1}{n!}$.*

Hypothesis 2. *Polynomial systems coming from the resolution of Equation (1) in Step 2 are of dimension zero (they thus have a finite number of solutions over an algebraic closure of $\mathbb{F}_{q^n}$).*

Using the double large prime variation and for a fixed degree extension n, the complexity of this index calculus attack is $\widetilde{O}(q^{2-\frac{2}{n}})$ where the notation $\widetilde{O}$ means that we omit the logarithmic factors in q. It is thus faster than Pollard rho method in $\widetilde{O}(q^{\frac{n}{2}})$ for $n \geq 3$ and sufficiently large q. However, this complexity hides an exponential dependence in n in step 2, which is the main topic of this work. Thus, the main focus of this paper is the resolution of the following problem.

Point Decomposition Problem (PDP). *Given a point R in an elliptic curve $E(\mathbb{F}_{q^n})$ and a factor base $\mathscr{F} \subset E(\mathbb{F}_{q^n})$, 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 solving a polynomial system. This can be done by first computing a Gröbner basis of the system for a degree ordering with F_4 [21] or F_5 [22]. Then computing the lexicographical Gröbner basis by using a change of ordering algorithm [25, 26, 24].

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, the Nagao variant needs to solve polynomial systems with a number of variables quadratic in n instead of n variables with the summation polynomials of Semaev. Therefore, in the elliptic case, it seems to be always better to use Semaev's polynomials, 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 if there is a small rational subgroup in $\mathbb{G}$, the Pohlig-Hellman reduction allows to 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 emphasize some elliptic curves models 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 from a cryptographic point of view, we decrease the bound on the complexity by a factor of $2^{\omega(n-1)}$. More precisely, under the hypothesis that the systems are regular, 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)* $\widetilde{O}\left(n \cdot 2^{3(n-1)^2}\right)$
- *(heuristic complexity)* $\widetilde{O}\left(n^2 \cdot 2^{\omega(n-1)^2}\right)$

where $2 \leq \omega < 3$ is the linear algebra constant that is the exponent in the complexity of multiplying two dense matrices.

The proven complexity of Theorem 1.1 is obtained by using the classical complexity of change of ordering algorithm, FGLM in $O(nD^3)$ [25] where D is the number of solutions counted with multiplicities in the algebraic closure of the coefficient field. The heuristic complexity is obtained by using a change of ordering algorithm recently proposed in [24]. This algorithm follows the approach of [26]. 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 in $\overline{\mathbb{F}_q}$ of the polynomial systems to be solved and to speed up 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 with an interesting action. 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 $2^{3(n-1)}$ (or $2^{\omega(n-1)}$ for the heuristic case) in the change of ordering step, compared to the general case.

In the first step of the general method for solving polynomial systems, one has to compute a degree reverse lexicographical ordering Gröbner basis. The complexity of computing such a Gröbner basis with F_4 or F_5 is related to the maximal degree reached by the polynomials during the computation. Without some assumptions on the system, such a bound is very hard to handle. We will show that by using the 2-torsion of twisted Edwards or Jacobi intersections curves the bound on the complexity of computing a Gröbner basis for a degree monomial ordering is divided by $2^{\omega(n-1)}$ when the systems are assumed to be regular (note that in [34], a similar hypothesis for overdetermined systems has been supposed). Indeed, a quasi-homogeneous structure (see [28]) appears when we apply the change of coordinates associated to the action of the dihedral Coxeter group. Such a structure amounts to consider a weighted degree instead of the usual degree.

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 [23].

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 approximately 2^{160} operations in $E(\mathbb{F}_{q^5})$ and only 2^{130} 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 prime finite fields of high characteristic or binary fields of prime degree extension.

1.4. **Related work.** In [34], Joux and Vitse improve the complexity of the index calculus algorithm for medium 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 dependence 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. The end of this section is devoted to the complexity of computing a Gröbner basis for a degree ordering of an invariant polynomial system. 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 that confirm the theoretical results and Section 6 concludes the paper by giving some possible perspectives.

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 the summation polynomials introduced by Semaev to improve the efficiency of this 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 in Weierstrass representation 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 $P_1, \ldots, P_n \in \mathscr{F}$ such that

$$R = P_1 \oplus \cdots \oplus P_n .$$

Writing $\mathbb{F}_{q^n} = \mathbb{F}_q[X]/\mu(X) = \mathbb{F}_q[\alpha]$ where $\mu(X)$ is an irreducible polynomial over $\mathbb{F}_q$ of degree n and α 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 $\{1, \alpha, \ldots, \alpha^{n-1}\}$ is a basis. Frey [30] showed that any instance of the ECDLP can be mapped to an instance of the DLP in the Weil restriction of $E(\mathbb{F}_{q^n})$ from $\mathbb{F}_{q^n}$ to $\mathbb{F}_q$. In the same way, the PDP over any elliptic curve defined over a non prime finite field can be mapped 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 $(x_0, \ldots, x_{n-1}, y_0, \ldots, y_{n-1}) \in (\mathbb{F}_q)^{2n}$ such that $\left(\sum_{i=0}^{n-1} x_i \cdot \alpha^i, \sum_{i=0}^{n-1} y_i \cdot \alpha^i\right)$ is a point of $E(\mathbb{F}_{q^n})$. The group law of E gives 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 $P_i \in E$ for $i = 1, \ldots, n-1$, we obtain a polynomial system with $n(n+1)$ variables and $n(n+1)$ equations in $\mathbb{F}_q$. It is not necessary to add the equation for $P_n \in E$ because this information is already in the system. Indeed, we have $P_1, \ldots, P_{n-1} \in E$ and $P_n = R \ominus (P_1 \oplus \ldots \oplus P_{n-1})$ with $R \in E$ and by consequence P_n too. The

system has as many unknowns as equations then under regularity assumptions, it is of dimension 0. The hypothesis of dimension 0 has been checked in practice so we follow Hypothesis 2. In order to solve this system, we use Gröbner bases. 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 1. *Let E be an elliptic curve defined by a planar equation over a field $\mathbb{F}_{q^n}$ and let $\overline{\mathbb{F}_{q^n}}$ be an algebraic closure of this field. 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 $(x_1,y_1)\oplus\ldots\oplus(x_m,y_m)$ 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 $\mathbb{F}_{q^n}$. The probability of decomposing a point w.r.t. $\mathscr{F}$ still follows the Hypothesis 1. In the context of Definition 1 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 of characteristic > 3 by a Weierstrass equation*

$$E : y^2 = x^3 + a_4x + a_6 \tag{2}$$

the summation polynomials of E are given by

$$\begin{cases} f_2(x_1,x_2) &= x_1 - x_2 \\ f_3(x_1,x_2,x_3) &= (x_1-x_2)^2x_3^2 - 2((x_1x_2+a_4)(x_1+x_2)+2a_6)x_3 + \\ & \quad (x_1x_2-a_4)^2 - 4a_6(x_1+x_2) \\ f_m(x_1,\ldots,x_n) &= Res_X(f_{m-k}(x_1,\ldots,x_{m-k-1},X), f_{k+2}(x_{m-k},\ldots,x_m,X)) \\ & \quad \text{for all } m \geq 4 \text{ and for all } m-3 \geq k \geq 1 \end{cases}$$

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. Summation polynomials are irreducible.

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

$$f_{n+1}(x_{P_1},\ldots,x_{P_n},x_R) = 0_{\mathbb{F}_{q^n}} \Longleftrightarrow \sum_{k=0}^{n-1} \varphi_{R,k}(x_{P_1},\ldots,x_{P_n})\cdot\alpha^k = 0_{\mathbb{F}_{q^n}}$$

where the $\varphi_{R,k}(x_{P_1},\dots,x_{P_n})$ are polynomials in $\mathbb{F}_q[x_{P_1},\dots,x_{P_n}]$. Thus, solving equation 3 is equivalent to solving the polynomial system $\mathscr{S} = \{\varphi_{R,k}(x_{P_1},\dots,x_{P_n}),\ k = 0,\dots,n-1\}$ 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. *Let ι be the automorphism of degree 2 of E which associates to a point its negation:*

$$\begin{array}{rccc} \iota: & E(\mathbb{F}_{q^n}) & \longrightarrow & E(\mathbb{F}_{q^n}) \\ & (x,y) & \longmapsto & \ominus(x,y) = (x,-y)\,. \end{array}$$

Let π_x and π_y be respectively, the projection on x and y. We can note that $\pi_x(x,y) = \pi_x(\iota(x,y))$ and $\pi_y(x,y) \neq \pi_y(\iota(x,y))$. Clearly, $\pi_x(E) \simeq E/\iota$ and the PDP in m points have more solutions in E^m than in $(E/\iota)^m$. This is not true for π_y. By consequence, by projecting on x, we obtain summation polynomials with smaller degree. In the following, we then choose to project on the coordinate c, if it exists, such that there exists an automorphism ψ of E such that $\pi_c(E) \simeq E/\psi$ and for all P, $\pi_c(P) = \pi_c(\psi(P))$. For both studied representations, this automorphism exists and will be ι.

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 [4]. This is a generalization of the representation proposed by Edwards in [18]. These curves were deeply studied by the cryptology community, especially by Bernstein and Lange [5], for their efficient arithmetic. In [4] 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

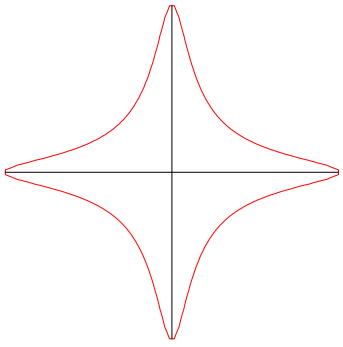

FIGURE 1. Edwards curve over the real numbers.

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

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 $\left(-a^{-\frac{1}{2}},0\right)$.

To solve the PDP in twisted Edwards representation, we have to construct the summation polynomial of such a curve. As said in 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. Moreover, the proof of irreducibility of summation polynomials by Semaev does not depend on the representation of the curve or the coordinate we project to. Hence, it can be applied *mutatis mutandis* for twisted Edwards or Jacobi intersections summation polynomials.

2.3.2. *Twisted Jacobi intersections curves.* This form of elliptic curves was introduced in 2010 in [29]. As for twisted Edwards curves, it is a generalization of Jacobi intersections curves (which are the intersections of two quadratic surfaces defined in a 3-dimensional space) proposed by D.V. and G.V. Chudnovsky in [9]. The twisted Jacobi intersections curves are 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 intersection 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

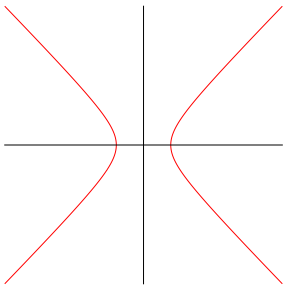

FIGURE 2. Projection of a Jacobi intersection curve over the real numbers.

P gives respectively the points $(-x,y,-z),(-x,-y,z)$
and $(x,-y,-z)$. The group law is given by

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

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 squares 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 squares 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 squares 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 and twisted Edwards representations, these summation polynomials are irreducible and 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 two sections.

3. Solving polynomial systems and symmetries

In this section, we first recall some results about the complexity of computing Gröbner bases. All these complexities are given in numbers of arithmetic operations. Then, we give some background on invariant theory. Finally, we recall a classical strategy to solve invariant polynomial systems and we discuss its impact on Gröbner basis computation complexity. For a more thorough reading on the subject, see [13] for an introduction on computational commutative algebra and [47] for a general exposition on computational invariant theory. In all this section, we consider ideals generated by polynomial systems and their corresponding

algebraic variety. It is worth noticing that even if some considered ideals are generated by homogeneous polynomials, we always consider their affine variety only. In particular, the dimension of such an ideal is the one corresponding to its affine variety.

3.1. **Gröbner basis.** A reduced Gröbner basis of a given ideal $\mathscr{I} \subset \mathbb{K}[x_1,\dots,x_n]$ is a set of polynomials generating this ideal. It is not the unique basis of an ideal but once the monomial ordering is fixed in the polynomial ring, it is a canonical basis after normalization. This canonical basis can have a lot of useful properties. In particular, by setting $\overline{\mathbb{K}}$ an algebraic closure of $\mathbb{K}$, from the lexicographical reduced Gröbner basis of $\mathscr{I}$, one can read off the set of elements in the affine space $\mathbb{A}^n = \overline{\mathbb{K}}^n$ canceling all the polynomials in $\mathscr{I}$. This set is called the algebraic *variety* or the *solutions* of the ideal $\mathscr{I}$. In the sequel, we consider ideals with corresponding varieties of finite cardinality only, such ideals are said to be of *dimension zero*. In this particular case, the reduced lexicographical Gröbner basis has the following triangular form

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

From such a triangular form, one can deduce the solutions of $\mathscr{I}$ by factoring univariate polynomials using Berlekamp or Cantor-Zassenhaus algorithm (see [49]). As here the ideal is assumed to be zero-dimensional, one can count its number of solutions in $\mathbb{A}^n$ with multiplicities, this number is denoted by D and it is also called the *degree* of the ideal in this situation. The expected shape of a lexicographical Gröbner basis is named *shape position* and has the following form:

$$\begin{cases} x_1 - h_1(x_n) \\ \vdots \\ x_{n-1} - h_{n-1}(x_n) \\ h_n(x_n) \end{cases}$$

where, $h_1,\dots,h_{n-1}$ are univariate polynomials of degree less than D and h_n is a univariate polynomial of degree exactly D.

Usually, to compute such a Gröbner basis we proceed in two steps. First we compute a Gröbner basis for the degree reverse lexicographical ordering. Then, from this basis, we compute the lexicographical Gröbner basis by using a change of ordering algorithm [26, 25, 24]. For the the first step, we consider the algoritms F_4 or F_5 [21, 22], we now present some results about their complexity.

3.1.1. *Complexity of F_4 and F_5 algorithms.* For these algorithms, we investigate their complexity in the case of *graded monomial ordering*, that is to say, the monomials are ordered with respect to a given graduation and in case of equality, another ordering (e.g. reverse lexicographical) is applied in order to make it total. Such a

usual graded monomial ordering is the degree reverse lexicographical (see [13]). We recall that a graduation $\deg_w$ on the monomials of $\mathbb{K}[x_1,\ldots,x_n]$ is defined from a given sequence of weights $w=(w_1,\ldots,w_n)$ in the following way:

$$\deg_w(x_1^{\alpha_1}\cdots x_n^{\alpha_n})=\sum_{i=1}^{n} w_i\alpha_i.$$

It is worth noticing that the usual degree corresponds to $\deg_w$ with weights $(1,\ldots,1)$. In order to keep the standard notation, we use deg in this case and call weighted degree for any other graduation (i.e when $w\neq(1,\ldots,1)$). In this general context, we say that a polynomial is *homogeneous* if all its monomials have the same graduation (in the literature, a polynomial which is homogeneous for a weighted degree is usually said quasi-homogeneous but we do not use this terminology here). It is important to note that the homogeneity of a polynomial depends on the graduation.

Among polynomial systems, the homogeneous regular systems form a family of polynomial systems for which the complexity of F_4 and F_5 is well handled.

Definition 2 (Regular systems)**.** *Let $F=(f_1,\ldots,f_s)\in(\mathbb{K}[x_1,\ldots,x_n])^s$ be a sequence of $s\leq n$ non-zero homogeneous polynomials for a fixed graduation $\deg_w$. The sequence F is said to be* regular *if for all $i\in\{1,\ldots,s-1\}$, the polynomial f_{i+1} is not a zero divisor in the quotient ring $\mathbb{K}[x_1,\ldots,x_n]/\langle f_1,\ldots,f_i\rangle$. A homogeneous polynomial system $\{f_1,\ldots,f_s\}$ is said to be regular if the sequence $(f_1,\ldots,f_s)$ is regular.*

Here we consider only zero-dimensional ideals generated by a regular sequence of polynomials. Moreover, if a regular sequence is of length the number of variables ($s=n$) then the ideal that it generates is zero-dimensional. In order to simplify the notations we then consider that the number of polynomials in the system is always the number of variables. For homogeneous regular systems, the complexity of computing a graded reverse lexicographical Gröbner basis can be bounded by the complexity of computing the reduced row echelon form of a particular matrix (the Macaulay matrix, see Definition 4 below) which its size depends on a certain graduation $d=d_{\mathrm{reg}}$ (see [3]) called the *degree of regularity* of the system. This quantity is defined as follows.

Definition 3 (Degree of regularity)**.** *Let $\mathscr{I}$ be a zero dimensional ideal in the polynomial ring $\mathbb{K}[x_1,\ldots,x_n]$ equipped with a graded monomial ordering for a fixed graduation $\deg_w$. We assume that the ideal $\mathscr{I}$ is generated by a sequence of homogeneous polynomials $(f_1,\ldots,f_n)$. Let $LT(\mathscr{I})$ be the leading term ideal of $\mathscr{I}$, also called initial ideal, which is the ideal of $\mathbb{K}[x_1,\ldots,x_n]$ generated by the leading terms $LT(f)$ of the elements f in $\mathscr{I}$. The degree of regularity of $\mathscr{I}$, denoted d_{reg}, is defined as the minimal graduation d such that the set $M(d)$ of monomials $m\in\mathbb{K}[x_1,\ldots,x_n]$ of graduation $\deg_w(m)$ greater or equal to d verifies*

$$M(d)\subset LT(\mathscr{I}).$$

For regular systems, the Macaulay bound gives a bound on d_{reg} when the graduation is the usual degree (see [38]). For a weighted degree, such a bound is given in [28]. These results can be summarized in the following theorem.

Theorem 3.1 ([38][28]). *Let $F = (f_1, \ldots, f_n)$ be a regular sequence of non-zero homogeneous polynomials of $\mathbb{K}[x_1, \ldots, x_n]$ equipped with a graded monomial ordering for a fixed graduation $\deg_w$. By denoting d_i the graduation $\deg_w(f_i)$ we have the following bound*

$$d_{reg} \leq \max_{i=1,\ldots,n} \{w_i\} + \sum_{i=1}^{n} (d_i - w_i).$$

One can notice that if $w = (1, \ldots, 1)$, this bound is consistent with the usual one given by the Macaulay bound. Finally, in order to estimate the complexity of F_4 or F_5 algorithms, we need the size of the Macaulay matrix in graduation d_{reg}.

Definition 4 (Macaulay matrix). *Let $\{f_1, \ldots, f_n\}$ be a set of homogeneous polynomials of $\mathbb{K}[x_1, \ldots, x_n]$ and $>$ be a graded monomial ordering for a fixed graduation $\deg_w$. The Macaulay matrix in graduation d, denoted $Mac(d)$, is the matrix whose rows contain the coefficients of the polynomials tf_j for $j = 1, \ldots, n$ and all monomials t of $\mathbb{K}[x_1, \ldots, x_n]$ such that $\deg_w(tf_j) = d$. Each column of the matrix corresponds to a monomial of $\mathbb{K}[x_1, \ldots, x_n]$ of graduation d. The columns are arranged in descending order w.r.t. the monomial ordering $>$.*

The size of the Macaulay matrix in graduation d, is then deduce from the number of monomials in n variables of graduation d. Hence, for homogeneous regular systems, the arithmetic complexity of F_4 or F_5 algorithms can be bounded by:

$$O\left(\binom{n + d_{\text{reg}} - 1}{d_{\text{reg}}}^{\omega}\right) \text{ for the usual degree,} \tag{5}$$

$$O\left(\left(\frac{\text{Gcd}_{i=1,\ldots,n}\{w_i\}}{\prod_{i=1}^{n} w_i} \binom{d_{\text{reg}} + S_n}{d_{\text{reg}} + S_n - n + 1}\right)^{\omega}\right) \text{ for a weighted degree,} \tag{6}$$

where S_n is defined by $S_1 = 0$ and $S_i = S_{i-1} + w_i \frac{\text{Gcd}_{j=1,\ldots,i-1}\{w_j\}}{\text{Gcd}_{j=1,\ldots,i}\{w_j\}}$ for $i \geq 2$ and $2 \leq \omega < 3$ is the linear algebra constant. See [28] for more details about the size of Macaulay matrices with weighted degree.

In most applications as in this work, polynomial systems are not homogeneous. By consequence one needs to relate the complexity of solving an affine polynomial system to the complexity of solving a particular homogeneous system. For this purpose, we use the *homogeneous component of highest graduation* as specified in the next definition.

Definition 5 (Affine regular systems). *Let $F = (f_1, \ldots, f_n)$ be a sequence of non-zero affine polynomials of $\mathbb{K}[x_1, \ldots, x_n]$. We denote by $f_i^{(h)}$ the homogeneous component of highest graduation of f_i. The sequence F is said to be regular if the sequence of homogeneous polynomials $F^{(h)} = (f_1^{(h)}, \ldots, f_n^{(h)})$ is regular. An affine polynomial system is said to be regular if it is defined by an affine regular sequence.*

Let $F = \{f_1, \ldots, f_n\} \subset \mathbb{K}[x_1, \ldots, x_n]$ equipped with a fixed graduation $\deg_w$. Assume that F is an affine regular system as specified in the preceding definition. Let $G = \{g_1, \ldots, g_n\} \subset \mathbb{K}[x_1, \ldots, x_n, h]$ be the set of the homogenization of the elements in F. By equipping the polynomial ring $\mathbb{K}[x_1, \ldots, x_n, h]$ with the graduation $\deg_{w'}$ where $w'_{n+1} = 1$ and $w'_i = w_i$ for $i = 1, \ldots, n$, the complexity of computing the graded reverse lexicographical Gröbner basis of $\langle F \rangle$ can be bounded by the complexity of computing the graded reverse lexicographical Gröbner basis of $\langle G \rangle$. By consequence, for affine regular systems in $\mathbb{K}[x_1, \ldots, x_n]$, the complexity of computing a graded reverse lexicographical Gröbner basis can be bounded by the formula in equation (5) or (6) after replacing n by $n+1$ and setting $w_{n+1} = 1$.

When the system is not regular, the complexity of algorithms F_4 and F_5 is much more difficult to handle. Indeed, for affine non regular systems, some polynomials of graduation d in the ideal can be obtained by combination of polynomials of higher graduation *i.e.*:

$$f = \sum_{i=1}^{n} h_i f_i \text{ and } \exists i \in \{1, \ldots, n\} \text{ such that } \deg_w(h_i f_i) > \deg_w(f)\,. \tag{7}$$

As this phenomenon is difficult to anticipate, the complexity of F_4 or F_5 is very hard to estimate and there is no general tight bound on the complexity of F_4 and F_5 in this case.

In contrary to the computation of a Gröbner basis, for any class of polynomial systems, the complexity of the second step in the resolution of polynomial systems is well understood. This is what we present in the next section.

3.1.2. *Complexity of change of ordering.* The classical algorithm of change of ordering for Gröbner basis is FGLM [25]. Its complexity is in $O(nD^3)$ arithmetic operations. For generic systems, this complexity can be reduced to $O(n\log^2(D)D + \log(D)D^\omega)$ (see [24]).

Nevertheless, polynomial systems arising in this work are not generic in the sense of [24]. However, the authors proposed also an algorithm for non generic polynomial systems for which the complexity of the change of ordering can heuristically be bounded by $O(n\log^2(D)D + \log(D)D^\omega)$. This heuristic complexity has been checked on various examples. In particular, it seems to be valid for polynomial systems considered here.

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 the sequel, we consider the action of a finite linear group $\mathbb{G}$. We assume that the field $\mathbb{K}$ has a positive "large enough characteristic", that is to say not dividing the cardinality of $\mathbb{G}$. All notions of invariant theory recalled 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 $\mathbb{A}^n$ or any $\mathbb{K}$-vector space of dimension n by the matrix vector multiplication. This action can be translated to polynomial rings. More precisely we have the following definition.

Definition 6 (Invariant rings)**.** *Let $\mathbb{K}[x_1,\ldots,x_n]$ be a polynomial ring in n variables with coefficients in $\mathbb{K}$. The action of a group $\mathbb{G} \subset GL(\mathbb{K},n)$ on $\mathbb{K}[x_1,\ldots,x_n]$ is defined by*

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

where $g \cdot f$ is defined by $(g \cdot f)(v) = f(g^{-1} \cdot v)$ where v is the vector $(x_1,\ldots,x_n)$. This definition uses the inverse of g in order to get a left action. The invariant ring of $\mathbb{G}$ is the set of all invariant polynomials in $\mathbb{K}[x_1,\ldots,x_n]$:

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

One of the fundamental results in invariant theory was proven by Hilbert in the last decade of the nineteenth century and is summarized in the following theorem.

Theorem 3.2 (Hilbert's finiteness theorem)**.** *The invariant ring of $\mathbb{G}$ is finitely generated.*

Following this theorem, many results were provided for the decomposition of invariant rings. In particular, it is proven that $\mathbb{K}[x_1,\ldots,x_n]^{\mathbb{G}}$ is a finitely generated free module over $\mathbb{K}[\theta_1,\ldots,\theta_n]$ where $\theta_1,\ldots,\theta_n$ are algebraically independent. Consequently there exist $\eta_1,\ldots,\eta_t \in \mathbb{K}[x_1,\ldots,x_n]^{\mathbb{G}}$ such that

$$\mathbb{K}[x_1,\ldots,x_n]^{\mathbb{G}} = \bigoplus_{i=1}^{t} \eta_i \mathbb{K}[\theta_1,\ldots,\theta_n]. \tag{8}$$

The decomposition (8) is called a Hironaka decomposition of $\mathbb{K}[x_1,\ldots,x_n]^{\mathbb{G}}$. The polynomials $\theta_1,\ldots,\theta_n$ (resp. $\eta_1,\ldots,\eta_t$) are the *primary invariants* (resp. *secondary invariants*) of $\mathbb{K}[x_1,\ldots,x_n]^{\mathbb{G}}$.

To solve pointwise invariant polynomial systems (*i.e.* each polynomial in the system is in the invariant ring of the corresponding group) by using the symmetries, one has to rewrite the systems in terms of the primary and secondary invariants. If the invariant ring of $\mathbb{G}$ is not a polynomial algebra – *i.e.* the secondary invariants are not reduced to $\{1\}$ – considering the symmetries can complicate the resolution of the system. Actually, since secondary invariants are not independent, then considering the symmetries when these invariants are not trivial increases the number of equations and variables to consider. Consequently, the polynomial systems could be more 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 bases, see for instance [27]; we will not need this strategy in this work.

By consequence an elementary question is to know under which conditions on $\mathbb{G}$, its invariant ring is a graded polynomial algebra (and thus when the set of secondary invariants is trivial). The answer is given in the following theorem.

Theorem 3.3 (Shephard, Todd, Chevalley[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 groups can be represented thanks to a pseudo reflection group. In particular, the dihedral Coxeter group $D_n = (\mathbb{Z}/2\mathbb{Z})^{n-1} \rtimes \mathfrak{S}_n$ can be represented by the action on $\mathbb{A}^n$ defined by the rule that $\mathfrak{S}_n$ permutes the coordinates of the vectors, whereas $(\mathbb{Z}/2\mathbb{Z})^{n-1}$ changes the sign on an even number of its coordinates. From Theorem 3.3 the invariant ring of D_n is then a polynomial algebra. In the sequel, the dihedral Coxeter group D_n will always correspond to this representation. It is a well known group and its invariant ring too. Actually,*

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

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

In the case where $\mathbb{G}$ is a pseudo-reflection group, Theorem 3.3 allows to construct an isomorphism $\Omega_{\mathbb{G}}$ between $\mathbb{K}[x_1, \ldots, x_n]^{\mathbb{G}}$ and $\mathbb{K}[y_1, \ldots, y_n]$ where $y_1, \ldots, y_n$ are new indeterminates.

Definition 7. *Let $\mathbb{G}$ be a pseudo-reflective group and $\theta_1, \ldots, \theta_n \in \mathbb{K}[x_1, \ldots, x_n]^{\mathbb{G}}$ be the primary invariants of $\mathbb{G}$. We denote by $\Omega_{\mathbb{G}}$ the ring isomorphism from $\mathbb{K}[x_1, \ldots, x_n]^{\mathbb{G}}$ to $\mathbb{K}[y_1, \ldots, y_n]$ corresponding to the change of coordinates by the θ_i's and defined by*

$$\begin{array}{rcl} \Omega_{\mathbb{G}}^{-1} : \mathbb{K}[y_1, \ldots, y_n] & \longrightarrow & \mathbb{K}[x_1, \ldots, x_n]^{\mathbb{G}} \\ f & \longmapsto & f(\theta_1, \ldots, \theta_n)\,. \end{array}$$

In the following, we denote by $\mathbb{K}[\theta_1, \ldots, \theta_n]$ the polynomial ring given by the image of $\Omega_{\mathbb{G}}$.

We now see how to simplify the resolution of polynomial systems that are pointwise invariant under a pseudo-reflection group.

3.3. **Solving pointwise invariant system.** Let $\mathbb{G} \subset \mathrm{GL}(\mathbb{K}, n)$ be a pseudo reflection group. Let $\mathscr{I} = \langle f_1(x_1, \ldots, x_n), \ldots, f_n(x_1, \ldots, x_n) \rangle$ be an ideal of $\mathbb{K}[x_1, \ldots, x_n]$ such that for $i = 1, \ldots, n$, the polynomial f_i is in $\mathbb{K}[x_1, \ldots, x_n]^{\mathbb{G}}$. 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 $\mathbb{G}$ admits a Hironaka decomposition, we will see in the sequel that from $V(\mathscr{I})/\mathbb{G}$ one can compute all elements in $V(\mathscr{I})$. Thus, to compute Gröbner bases keeping symmetries, one can compute a Gröbner basis of an ideal having for variety 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 $\{\theta_1(x_1,\ldots,x_n),\ldots,\theta_n(x_1,\ldots,x_n)\}$ be a set of generators – primary invariants – of $\mathbb{K}[x_1,\ldots,x_n]^{\mathbb{G}}$. Since, the primary invariants are algebraically independent, the $\mathbb{G}$-orbit space $\mathbb{A}^n/\mathbb{G}$ is the variety $\mathbb{A}^n$ see [47]. Let $\mathscr{G}_{inv}$ be the lexicographical Gröbner Basis of

$$\langle \theta_1(x_1,\ldots,x_n)-y_1,\ldots,\theta_n(x_1,\ldots,x_n)-y_n\rangle \subset \mathbb{K}[x_1,\ldots,x_n,y_1,\ldots,y_n]$$

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

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

$$\mathscr{G}_{inv}\cup\{f_1(x_1,\ldots,x_n),\ldots,f_n(x_1,\ldots,x_n)\}$$

with respect to an ordering eliminating the x_i's. Actually, $\mathscr{G}=\mathscr{G}_{orb}\cap\mathbb{K}[y_1,\ldots,y_n]$ is a Gröbner basis of an ideal of variety $V(\mathscr{I})/\mathbb{G}$.

Example 2. *Let $n=2$ and $\mathbb{K}=\mathbb{F}_{65521}$. Let us consider the ideal $\mathscr{I}=\langle f_1,f_2\rangle$ where*

$$\begin{aligned} f_1(x_1,x_2) &= x_1^2x_2^2-x_1^2-x_2^2-1\\ f_2(x_1,x_2) &= x_1^4+x_1^3x_2+x_1x_2^3+x_2^4. \end{aligned}$$

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

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

The corresponding $\mathscr{G}_{inv}$ and $\mathscr{G}_{orb}$ Gröbner basis are respectively

$$\left\{\begin{array}{l} x_1^2+x_2^2-y_1\\ x_1x_2-y_2\\ x_1y_2+x_2^3-x_2y_1\\ x_2^4-x_2^2y_1+y_2^2 \end{array}\right. \quad \left\{\begin{array}{l} x_1-x_2^3y_2^3-x_2^3y_2^2+4x_2^3y_2+x_2^3-x_2y_2^3-x_2y_2^2+3x_2y_2+x_2\\ x_2^4-x_2^2y_2^2+x_2^2+y_2^2\\ y_1-y_2^2+1\\ y_2^4+y_2^3-4y_2^2-y_2+1 \end{array}\right.$$

The corresponding $\mathscr{G}$ basis in terms of y_1 and y_2 only is then

$$\left\{\begin{array}{l} y_1-y_2^2+1\\ y_2^4+y_2^3-4y_2^2-y_2+1 \end{array}\right.$$

which preserves the symmetries. One can notice that the degree of the ideal $\mathscr{I}$ is 16 *whereas considering the symmetries yields an ideal of degree divided by* 4.

In our case, we consider groups that are pseudo reflective, the impact on the complexity comes from the fact that we reduce the degree of the polynomials we consider by the change of coordinates $\Omega_{\mathbb{G}}$ and that all solutions in the same orbit will correspond to only one solution of the new system. So that the total number of solutions decreases. Hence, the complexity of the F_4 and FGLM steps are reduced accordingly.

The end of this section is devoted to the impact of such a change of coordinates on the complexity of computing a graded reverse lexicographical or lexicographical Gröbner basis.

3.3.1. *Complexity of F_4 and F_5 algorithms for a given pointwise invariant system.* For the resolution of the *Point Decomposition Problem*, we will see in the next section that we can construct polynomial systems invariant under the action of the dihedral Coxeter group. Moreover, we have observed in practice that using the action of the symmetric group only, yields a regular system in this case. By consequence, we now consider the complexity of computing a weighted degree reverse lexicographical, denoted WDRL, Gröbner basis of $\mathscr{S}_{D_n}$ when it is assumed that $\mathscr{S}_{\mathfrak{S}_n}$ is regular.

Let $s_1,\dots,s_{n-1},e_n \in \mathbb{K}[x_1,\dots,x_n]$ be the primary invariants of the dihedral Coxeter group D_n. As the symmetric group is a subgroup of D_n each of the primary invariants of D_n can be written in terms of the elementary symmetric polynomials. Let ρ_i denotes an expression of s_i in $\mathbb{K}[e_1,\dots,e_n]$ one can easily deduce that,

$$\begin{cases} \rho_i = e_i^2 + 2\sum_{j=1}^{i-1}(-1)^j e_{i-j}e_{i+j} + 2(-1)^i e_{2i} & \text{if} \quad i \leq \lfloor n/2 \rfloor \\ \rho_i = e_i^2 + 2\sum_{j=1}^{n-i}(-1)^j e_{i-j}e_{i+j} & \text{if} \quad \lfloor n/2 \rfloor < i < n \\ \rho_n = e_n \end{cases} .$$

This representation of the primary invariants of D_n in $\mathbb{K}[e_1,\dots,e_n]$ allows to construct a weighted degree which preserves the grading between the two rings $\mathbb{K}[e_1,\dots,e_n]$ and $\mathbb{K}[s_1,\dots,s_{n-1},e_n]$.

Lemma 1. *For all $f \in \mathbb{K}[x_1,\dots,x_n]^{D_n} \subset \mathbb{K}[x_1,\dots,x_n]^{\mathfrak{S}_n}$, if $\mathbb{K}[s_1,\dots,s_{n-1},e_n]$ is equipped with the graduation $\deg_w$ with weights $(2,\dots,2,1)$ then $\deg_w(\Omega_{D_n}(f)) = \deg(\Omega_{\mathfrak{S}_n}(f))$.*

Proof. Let $\Omega_{D_n}(f) = \sum_{\alpha=(\alpha_1,\dots,\alpha_n)} c_\alpha s_1^{\alpha_1}\cdots s_{n-1}^{\alpha_{n-1}} e_n^{\alpha_n}$ with $c_\alpha \in \mathbb{K}$ and

$$\deg_w(\Omega_{D_n}(f)) = \max\left\{ \alpha_n + 2\sum_{i=1}^{n-1}\alpha_i \mid c_\alpha \neq 0 \right\} .$$

Then $\Omega_{\mathfrak{S}_n}(f) = \sum_{\alpha=(\alpha_1,\dots,\alpha_n)} c_\alpha \rho_1^{\alpha_1}\cdots\rho_{n-1}^{\alpha_{n-1}}\rho_n^{\alpha_n}$ with

$$\deg(\Omega_{\mathfrak{S}_n}) = \max\left\{ \sum_{i=1}^{n}\deg(\rho_i)\alpha_i \mid c_\alpha \neq 0 \right\} = \deg_w(\Omega_{D_n}(f)) .$$

□

Let F be a sequence of invariant polynomials under the action of the dihedral Coxeter group. If the image of F by $\Omega_{\mathfrak{S}_n}$ is a regular sequence, we now show that Ω_{D_n} also allows to construct a regular sequence.

Proposition 1. *Let $(f_1,\dots,f_n) \in \left(\mathbb{K}[x_1,\dots,x_n]^{D_n}\right)^n \subset \left(\mathbb{K}[x_1,\dots,x_n]^{\mathfrak{S}_n}\right)^n$ be a sequence of polynomials such that $(\Omega_{\mathfrak{S}_n}(f_1),\dots,\Omega_{\mathfrak{S}_n}(f_n)) \in (\mathbb{K}[e_1,\dots,e_n])^n$ is a regular sequence for the usual graduation $\deg = \deg_w$ with $w = (1,\dots,1)$.*

If $\mathbb{K}[s_1,\dots,s_{n-1},e_n]$ is equipped with a weighted degree $\deg_w$ of weights $w = (2,\dots,2,1)$ then $(\Omega_{D_n}(f_1),\dots,\Omega_{D_n}(f_n)) \in (\mathbb{K}[s_1,\dots,s_{n-1},e_n])^n$ is a regular sequence.

Proof. In order to simplify the notations, for all $f \in \mathbb{K}[x_1,\dots,x_n]^{D_n}$ we denote by $f^{(s)}$ (resp. $f^{(d)}$) the polynomial $\Omega_{\mathfrak{S}_n}(f)$ (resp. $\Omega_{D_n}(f)$) and by $f^{(s,h)}$ (resp. $f^{(d,h)}$) its homogeneous component of highest degree (resp. weighted degree).

Let $\alpha = (\alpha_1,\dots,\alpha_n) \in \mathbb{N}^n$, we denote $|\alpha| = \sum_{i=1}^n \alpha_i$ and $|\alpha|_w = \sum_{i=1}^{n-1} 2\alpha_i + \alpha_n$. For all $f \in \mathbb{K}[x_1,\dots,x_n]^{D_n}$ we have

$$f^{(d)}(s_1,\dots,s_{n-1},e_n) = \sum_{|\alpha|_w=\delta} c_\alpha s_1^{\alpha_1}\cdots e_n^{\alpha_n} + R_1(s_1,\dots,s_{n-1},e_n)$$

where δ is the weighted degree of $f^{(d)}$, $c_\alpha \in \mathbb{K}$ and R_1 is a polynomial of weighted degree less than δ. Let denote $\rho_i - \rho_i^{(h)}$ by r_i we have:

$$\begin{aligned} f^{(s)}(e_1,\dots,e_n) &= f^{(d)}(\rho_1,\dots,\rho_n) \\ &= \sum_{|\alpha|_w=d} c_\alpha(\rho_1^{(h)}+r_1)^{\alpha_1}\cdots(\rho_n^{(h)}+r_n)^{\alpha_n} + R_1(\rho_1,\dots,\rho_n) \\ &= \sum_{|\alpha|_w=d} c_\alpha(\rho_1^{(h)})^{\alpha_1}\cdots(\rho_n^{(h)})^{\alpha_n} + R_2(e_1,\dots,e_n) \end{aligned}$$

where R_2 is a polynomial of degree less than δ which contains $R_1(\rho_1,\dots,\rho_n)$ by Lemma 1. This implies that

$$\begin{aligned} f^{(s,h)} &= \sum_{|\alpha|_w=d} c_\alpha(\rho_1^{(h)})^{\alpha_1}\cdots(\rho_n^{(h)})^{\alpha_n} \\ &= f^{(d,h)}(\rho_1^{(h)},\dots,\rho_n^{(h)}). \end{aligned} \tag{9}$$

Assume that the sequence $(f_1^{(d,h)},\dots,f_n^{(d,h)})$ is not regular *i.e.* there exists $i \in \{2,\dots,n\}$ and $0 \neq g, g_1,\dots,g_{i-1} \in \mathbb{K}[s_1,\dots,s_{n-1},e_n]$ such that

$$g_1 f_1^{(d,h)} + \cdots + g_{i-1} f_{i-1}^{(d,h)} - g f_i^{(d,h)} = 0.$$

From equation (9) this implies that

$$g^{(h)}(\rho_1^{(h)},\dots,\rho_n^{(h)}) f_i^{(s,h)} - \sum_{j=1}^{i-1} g_j^{(h)}(\rho_1^{(h)},\dots,\rho_n^{(h)}) f_j^{(s,h)} = 0.$$

Since, $\rho_1^{(h)},\dots,\rho_n^{(h)}$ are algebraically independent we have $g^{(h)}(\rho_1^{(h)},\dots,\rho_n^{(h)}) \neq 0$. Hence, $f_i^{(s,h)}$ is a zero divisor in the quotient ring $\mathbb{K}[e_1,\dots,e_n]/\langle f_1^{(s,h)},\dots,f_{i-1}^{(s,h)}\rangle$. This yields a contradiction hence the sequence $(f_1^{(d,h)},\dots,f_n^{(d,h)})$ is regular. □

Finally, we study the complexity of computing a (W)DRL Gröbner basis with F_4 or F_5 for some regular sequences.

Theorem 3.4. *Let $f_1,\dots,f_n \in \mathbb{K}[x_1,\dots,x_n]^{D_n}$ be such that* $\deg(\Omega_{\mathfrak{S}_n}(f_i)) = 2^{n-1}$ *and such that the sequence $F^{(s)} = (\Omega_{\mathfrak{S}_n}(f_1),\dots,\Omega_{\mathfrak{S}_n}(f_n))$ is regular for the usual graduation deg. The arithmetic complexity of computing a DRL Gröbner basis of the system generated by $F^{(s)}$ is bounded by*

$$O\left(\binom{n2^{n-1}+1}{n}^{\omega}\right) = O\left(2^{\omega n(n-1)}\right).$$

Let $F^{(d)} = (\Omega_{D_n}(f_1), \ldots, \Omega_{D_n}(f_n))$. The arithmetic complexity of computing a WDRL Gröbner basis with weights $(2, \ldots, 2, 1)$ of the system generated by $F^{(d)}$ is bounded by

$$O\left(2^{-\omega(n-1)}\binom{n2^{n-1}+2}{n}^{\omega}\right) = O\left(2^{\omega(n-1)^2}\right).$$

Proof. As $F^{(s)}$ is a regular sequence, from Theorem 3.1 we can bound $d_{\mathrm{reg}}(F^{(s)})$ by the Macaulay bound *i.e.*

$$d_{\mathrm{reg}}(F^{(s)}) \leq 1 + \sum_{i=1}^{n}(2^{n-1}-1) = n2^{n-1} - n + 1\,.$$

Hence, from equation (5) we obtain the expected result. From Lemma 1 and Proposition 1, $F^{(d)}$ is a regular sequence such that $\deg_w(\Omega_{D_n}(f_i)) = 2^{n-1}$. Thus, again from Theorem 3.1, we obtain

$$d_{\mathrm{reg}}(F^{(d)}) \leq \sum_{i=1}^{n-1}(2^{n-1}-2) + 2^{n-1} - 1 + 2 = n2^{n-1} - 2(n-1) + 1\,.$$

Hence, from equation (6) we obtain the second expected result. □

Remark 2. *One can notice that considering the sequence $F^{(d)}$ (i.e. the system $\mathscr{S}_{D_n}$) instead of $F^{(s)}$ (i.e. $\mathscr{S}_{\mathfrak{S}_n}$) divides by $2^{\omega(n-1)}$ the complexity of F_4 or F_5 in the step of Gröbner basis computation. This factor on the complexity is consistent with the results that we obtain in practice (see Section 5).*

We now present the impact on the complexity of the change of ordering algorithm.

3.3.2. *Complexity of change of ordering for invariant ideals.* Let $\mathscr{I}$ be a zero dimensional ideal of $\mathbb{K}[x_1, \ldots, x_n]$ which is invariant under the action of a finite pseudo reflection group $\mathbb{G} \subset \mathrm{GL}(\mathbb{K}, n)$. 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 $\mathbb{G}$ denoted $\mathscr{I}_{\mathbb{G}}$. Let $\mathrm{Orb}(\mathbb{G}, v)$ be the orbit of $v \in \mathbb{A}^n$ under the action of $\mathbb{G}$ and $\mathrm{Stab}(\mathbb{G}, v)$ be the stabilizer of v. From the orbit-stabilizer theorem, for all $v \in \mathbb{A}^n$ we have

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

The degree $\deg(\mathscr{I})$ of the ideal $\mathscr{I}$ is the number of its solutions counted with multiplicities. Let $v \in V(\mathscr{I})$ such a solution, its orbit $\mathrm{Orb}\,(\mathbb{G}, v)$ under the action of $\mathbb{G}$ is a solution of $\mathscr{I}_{\mathbb{G}}$. The multiplicity of v is then given by the multiplicity of $\mathrm{Orb}\,(\mathbb{G}, v)$, seen as a solution of $\mathscr{I}_{\mathbb{G}}$, times the number of elements in the stabilizer $\mathrm{Stab}\,(\mathbb{G}, v)$ of v. 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 a representative of the orbit $\widetilde{v}$ and N is 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 work in the orbit space. Hence the number of solutions counted with multiplicities of $\mathscr{I}_{\mathbb{G}}$ is the number of $\mathbb{G}$-orbits 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. Since the complexities of change of ordering algorithms are polynomial in the degree of the ideal, their complexities are then reduced accordingly. This is summarized in the following Proposition.

Proposition 2. *Let $\mathbb{G}$ be a pseudo reflection group. Let $\mathscr{I}$ be an ideal generated by pointwise invariant polynomials under $\mathbb{G}$. Applying the change of coordinates associated to $\mathbb{G}$ divides the complexity of the change of ordering algorithm by $(\#\mathbb{G})^3$ and by $(\#\mathbb{G})^\omega$ in the heuristic case.*

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 $\langle\mathscr{G}\rangle$ is* $4=\dfrac{16}{\#D_2}$ *and*

- $O_1=(64799,361)$ *is a representative of* $\{(2996,62525),(62525,2996)\}$
- $O_2=(726,65158)$ *is a representative of* $\{(6897,58624),(58624,6897)\}$
- $O_3=(6009,6009)$ *is a representative of* $\{(7493,55256),(10265,58028),(55256,7493),(58028,10265)\}$
- $O_4=(59513,59513)$ *is a representative of* $\{(14169,28989),(28989,14169),(36532,51352),(51352,36532)\}$

Remark 3. *Note that in general, a $\mathbb{K}$-rational orbit can be formed by non $\mathbb{K}$-rational elements. That is to say, some $\mathbb{K}$-rational solutions of the system after a non-linear change of variables can correspond to solutions of the initial system which have coordinates not in $\mathbb{K}$.*

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.

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. The notation $c(P)$ denotes the c-coordinate of the point P. Let $\mathscr{F}_i=\left\{P\in E(\mathbb{F}_{q^n})\mid \frac{c(P)}{\alpha^i}\in\mathbb{F}_q\right\}$ for any $i=0,\ldots,n-1$ where α is a generator of $\mathbb{F}_{q^n}$. For Weierstrass or twisted Edwards representations, we take as factor base $\mathscr{F}=\mathscr{F}_0$. For Jacobi intersections curves, if $\mathbb{F}_q$ is a prime field then $\mathscr{F}_0$ contains only the 2-torsion of the curves; hence it does not contain enough points to be used as factor base. Therefore, for this representation we take as factor base $\mathscr{F}=\mathscr{F}_1$.

4.1. **Group action on the point decomposition problem.**

4.1.1. *The symmetric group* $\mathfrak{S}_n$. As we have seen in Section 2, the summation polynomials are symmetric and it is natural [31] 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 $\{e_1,\dots,e_n\}$ where e_i is the i^{th} elementary symmetric polynomial in terms of $c_1,\dots,c_n$. There exists a unique polynomial $g_n^R \in \mathbb{F}_{q^n}[e_1,\dots,e_n]$ 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 $\mathscr{S}_{\mathfrak{S}_n}{}^1 \subset \mathbb{F}_q[e_1,\dots,e_n]$ 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 $\mathscr{S}$ and we observe that the system becomes regular. 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 [31]	4	531	535
	Edwards	0	201	201
	Jacobi	0	209	209
64	Weierstrass [31]	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 and secondary invariants of $\mathbb{F}_q[x_1,\dots,x_n]^{\mathbb{G}}$.

Lexicographical Gröbner basis of $\langle \mathscr{S}_{\mathfrak{S}_n} \rangle$ 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 $\langle \mathscr{S}_{\mathfrak{S}_n} \rangle$ 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 $\#V_{\overline{\mathbb{F}_q}}(\langle \mathscr{S}_{\mathfrak{S}_n} \rangle) = 2^{n(n-1)}$.

Remark 4. *The form of the lexicographical Gröbner basis is given here in order to explain some intuition of our approach. In particular, such a form does not represent any assumption in the proof of our main result Theorem 4.1, below. Actually, one needs only a bound on the degree of the ideal considered in this proof. This bound is obtained thanks to the Bezout's theorem and results from invariant theory.*

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 bases for Weierstrass representation is in shape position. 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 bases for twisted Edwards and Jacobi intersections curves are not in shape position. The variable e_n is not separating for these two representations. In fact, for each solution of the system, there are $2^{n-1}-1$ others solutions with same value in e_n. By consequence, one would like to find a larger group than $\mathfrak{S}_n$ acting on the system (and thus on the variety of solutions) such that each orbit gathers all such solutions with the same value in e_n. In the next section, we show how to use such a larger group related to 2-torsion points in order to increase the efficiency of the computation.

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 $\mathscr{F}$. If the action of the 2-torsion point leaves invariant the factor base $\mathscr{F}$ *i.e.* $P \in \mathscr{F}$ implies that $P \oplus T_2 \in \mathscr{F}$ then the 2-torsion point can be used to reduce the size of the factor base (see Remark 5). By consequence, if we know a decomposition of R w.r.t. the factor base $\mathscr{F}$ (respectively a solution of the polynomial system to solve for solving the PDP) we can construct 2^{n-1} decompositions of R w.r.t. $\mathscr{F}$ (respectively 2^{n-1} solutions of the polynomial system).

Let c and c_2 be respectively the c-coordinate of P and $P \oplus T_2$. The action of the 2-torsion point leaves the factor base invariant if

$$\begin{cases} c_2 = \frac{p_1(c)}{p_2(c)} \text{ with } p_1, p_2 \in \mathbb{F}_q[c] & \text{if } \mathscr{F} = \mathscr{F}_0 \\ c_2 = \beta c + \gamma \text{ with } \beta \in \mathbb{F}_q \text{ and } \frac{\gamma}{\alpha^i} \in \mathbb{F}_q & \text{if } \mathscr{F} = \mathscr{F}_i, 1 \leq i < n \end{cases} \tag{10}$$

where α is a generator of $\mathbb{F}_{q^n}$. The difference between the two cases is due to when $\mathscr{F} = \mathscr{F}_0$ the c-coordinates of the points in the factor base are in a field whereas when $\mathscr{F} = \mathscr{F}_i$ with $i > 0$ the c-coordinates of the points in the factor base are in a vector space.

By consequence, if condition (10) is satisfied then the size of the factor base can be reduced. Moreover, we can *a priori* use the action of the 2-torsion to speed up the polynomial systems solving step in the PDP solving. Nevertheless, in order to use the action of the 2-torsion point in the polynomial system solving process, we need that c_2 depends only on c and that the action of T_2 on the coordinates is not too much complicated. The simplest being a linear action.

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

$$P \oplus T_2 = \left(\frac{x^3 + a_4x + a_6}{(X-x)^2} - x - X, \frac{(2x+X)y}{(x-X)} - \frac{y^3}{(x-X)^3} - y \right).$$

In this representation, we project the PDP on x-coordinate. As the x-coordinate of the point $P \oplus T_2$ does not verify any of the equalities in (10), the 2-torsion points cannot be used to decrease the factor base. Moreover, the action of the 2-torsion points is not easy to handle in the polynomial systems solving process.

In the 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 action of the 2-torsion point leaves invariant the factor base and the 2^{n-1} decompositions of the point R translate into as many solutions of the PDP. Furthermore, the action of the 2-torsion point being very simple we can use it to decrease the number of solutions in the polynomial systems solving process.

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 $P \oplus T_2 = (-x,y,-z),(-x,-y,z),(x,-y,-z)$ and similarly to the twisted Edwards curves, the decompositions mentionned above should correspond to solutions of the system associated to the decomposition of the point R.

Obviously, as Jacobi intersections curves have three 2-torsion points, the factor base can be further decreased and from one decomposition of R one can construct more than 2^{n-1} decompositions of R. However, since after projection on the c-coordinate (y or z) for any 2-torsion points, $c_2 = \pm c$ these decompositions will match with only 2^{n-1} solutions of the system we want to solve.

As a consequence, for twisted Edwards or Jacobi intersections curve from one solution of the polynomial system $(c_1,\dots,c_n)$ corresponding to the decomposition $R = P_1 \oplus \cdots \oplus P_n$, we can construct 2^{n-1} solutions of the system by applying an even number of sign changes. Obviously, each of these solutions can be the projection of many decompositions. Hence, from one solution $(c_1,\dots,c_n)$ 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 $(c_1,\dots,c_n)$ to which we apply an even number of sign changes and a permutation of $\mathfrak{S}_n$, that is the orbit of $(c_1,\dots,c_n)$ 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 obtained from f_{n+1}^R by a Weil restriction is invariant under the action of D_n and we have the following result.

Proposition 3. $f_{n+1}^R(c_1,\dots,c_n) \in \mathbb{F}_{q^n}[c_1,\dots,c_n]^{D_n}$.

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 on 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 the 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 $g \in D_n$,

$g \cdot f_{n+1}^R \in \langle f_{n+1}^R \rangle$ and so $g \cdot f_{n+1}^R = h_g \cdot f_{n+1}^R$ where $h_g \in \mathbb{F}_{q^n}[c_1, \ldots, c_n]$. 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 $h_g \in \mathbb{F}_{q^n}^{\times}$.

Let $\phi : D_n \to \mathbb{F}_{q^n}^{\times}$ be the application which maps g to h_g as defined above. Clearly, this application is a group morphism and thus $\phi(g)^m = h_g^m = 1$ where m is the order of g.

We note $\tau_{i,j}$ the transposition which swaps the elements in position i and j. Let $\mathscr{B} = \{\tau_{i,i+1} \mid i = 1, \ldots, n-1\}$ be a basis of $\mathfrak{S}_n$. A transposition is of order two and all the transpositions are conjugated, hence $\phi(\tau_{i,j}) = \phi(\tau_{k,l}) \in \{-1, 1\}$ for all $i, j, k, l \in \{1, \ldots, n\}$.

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

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

where $\mathrm{Syl}_X(p_1, p_2)$ is the Sylvester matrix of p_1 and p_2 w.r.t. the variable X. The Sylvester matrix of $f_k(c_1, \ldots, c_{k-1}, X)$ 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 $h_\tau = 1$ for all $\tau \in \mathscr{B}$. 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 $\{s_1, \ldots, s_{n-1}, e_n\}$ (or $\{p_2, \ldots, p_{2(n-1)}, p_n\}$). Hence, there exists a unique polynomial $g_n^R \in \mathbb{F}_{q^n}[s_1, \ldots, s_{n-1}, e_n]$ (respectively $\mathbb{F}_{q^n}[p_2, \ldots, p_{2(n-1)}, p_n]$) such that g_n^R is the expression of f_{n+1}^R in terms of the primary invariants $\{s_1, \ldots, s_{n-1}, e_n\}$ (respectively $\{p_2, \ldots, p_{2(n-1)}, p_n\}$). 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 $\mathbb{F}_q[p_2, \ldots, p_{2(n-1)}, p_n]$) with n variables and n equations. The degree of $\langle \mathscr{S}_{D_n} \rangle$ can be bounded by

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

To estimate an explicit complexity bound on the resolution of the *Point Decomposition Problem* we need to assume that the system $\mathscr{S}_{\mathfrak{S}_n}$ is regular. This property for $\mathscr{S}_{\mathfrak{S}_n}$ has been verified on all experiments we did (see Table 1). Moreover, a similar hypothesis was already done for the same kind of systems in [34]. Hence, it is reasonable to assume it.

Hypothesis 3. *Polynomial systems arising from a Weil descent on summation polynomial on which we apply the change of coordinates corresponding to the action of the symmetric group are regular.*

We can note that Hypothesis 3 implies Hypothesis 2. We have therefore obtained our main theorem.

Theorem 4.1. *In twisted Edwards (respectively twisted Jacobi intersections) representation, under the Hypothesis 3, the Point Decomposition Problem can be solved in time*

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

where $2 \leq \omega < 3$ *is the linear algebra constant.*

Proof. From Theorem 3.4, computing a Gröbner basis for a degree order of $\mathscr{S}_{D_n}$ can be done in time $\widetilde{O}\left(2^{\omega(n-1)^2}\right)$.

Given this previous Gröbner basis, computing the lexicographical Gröbner basis can be done in time $\widetilde{O}\left(n \cdot 2^{3(n-1)^2}\right)$ (resp. $\widetilde{O}\left(n^2 \cdot 2^{\omega(n-1)^2}\right)$ in the heuristic case).

Finally, it is straightforward that the change of ordering step dominates which concludes the proof. □

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 $\langle \mathscr{S}_{\mathfrak{S}_n} \rangle$:

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

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

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

where

- $\deg\left(\langle \mathscr{S}_{\mathfrak{S}_n} \rangle\right) = 2^{n(n-1)}$ and $\deg\left(\langle \mathscr{S}_{D_n} \rangle\right) = 2^{(n-1)^2}$
- $\deg_{e_{n-1}}(\mathfrak{p}_{n-1}) = 2^{n-1}$, $\deg(\mathfrak{p}_n) = 2^{(n-1)^2}$ 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.

Remark 5. *In* [31]*, the author uses the action of the automorphism* ι *to decrease the size of the factor base. Let* $S_1, S_2 \subset E$ *be such that* $\mathscr{F} = S_1 \cup S_2$, $S_1 \cap S_2 = \{P \in \mathscr{F} | [2]P = P_\infty\}$ *and* $S_i = Img(\iota(S_j))$ *with* $i \neq j$*. Instead of taking* $\mathscr{F}$ *as factor base, he takes* S_1 *of size* $\sim \frac{q}{2}$ *without decreasing the probability of decomposition.*

In addition to speed up the resolution of the polynomial systems, the use of the 2-torsion points of twisted Edwards or Jacobi intersections curves allows to

further decrease the size of the factor base by keeping the same probability of decomposition. Following the previous idea we can write $\mathscr{F} = S_1 \cup S_2$ such that for all $P \in \mathscr{F}$, S_1 contains a representative of the orbit of P under the action of ι and T_2 and S_2 contains all the others points in the orbit of P. Finally, we take as factor base S_1 of size $\sim \frac{q}{4}$ for twisted Edwards curves and $\sim \frac{q}{8}$ for twisted Jacobi intersections curves.

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 the 4-torsion points be used in the same way?** 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. Hence, the action of T_4 does not leave invariant the factor base. Moreover, in this representation the x-coordinate cannot be expressed in terms of the y-coordinate only so we cannot use this action to decrease the number of solutions of polynomial systems to solve.

4.2.2. *Action of the 4-torsion points of a twisted Jacobi intersections curve.* In this section, we present a similar method, as for 2-torsion, to use the 4-torsion of twisted Jacobi intersections curves. Although we will see in Section 5 that this method does not allow to simplify the polynomial system solving step in the PDP solving, we present it for completeness and in order to report the experiments we did. Moreover, we will see that this approach is not useless, since it allows to further decrease the size of the factor base and consequently to speed up the complete solving of the ECDLP by index calculus attack.

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

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

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.

One can notice that the z-coordinate of $P \oplus T_4$ depends only on the z-coordinate of P. However, due to the factor $\pm\sqrt{\frac{a-b}{a}}$ and also that for this representation the factor base cannot be $\mathscr{F}_0$ the action of T_4 does not leave the factor base invariant.

By consequence, in order to normalize a bit more the action of T_4 and to use the action of the 4-torsion, 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 changes the Z-coordinate to $\pm 1/Z$. Moreover, in this case the factor base $\mathscr{F} = \mathscr{F}_0$ seems to be large enough. Hence, the action of T_4 leaves the factor base invariant and can be used to further decrease the size of the factor base $\sim \frac{q}{16}$. 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 developed. 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 a well-chosen variable to make the symmetry more visible, we constrain the LEX Gröbner basis to be in non shape position that had shown 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 6. *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.*

We can note that adding two times T_4 (*i.e.* adding a 2-torsion point) does not change the Z-coordinate. By consequence, we can change only an even number

of variables by their inverse. Instead of $v_4 = e_4 + \frac{1}{e_4}$ we could use $v_4' = \frac{s_2+1+e_4^2}{e_4}$ to further decrease the degree of the univariate polynomial in the lexicographical Gröbner basis.

The construction that we have just shown works *mutatis mutandis* with the other 4-torsion point 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 clear that adding a variable to reduce the degree of the polynomials in the resulting Gröbner basis is worthwhile. Nevertheless, whether we add the variable v_4 or not, the action of this 4-torsion point allows to further decrease the size of the factor base by a factor 2. Indeed, we mention in the beginning of Section 4 that for twisted Jacobi intersections curves we cannot use the factor base $\mathscr{F}_0$ since it does not contain enough points. Hence, in this case the 4-torsion does not leave invariant the factor base and then cannot be used to decrease to size of the factor base. However, by changing the representation of the curve to normalize the action of the 4-torsion, the corresponding factor base $\mathscr{F}_0$ seems to contain the expected number of points and then can be choose for index calculus attack. Moreover, in this case the action of the 4-torsion leaves invariant the factor base and in consequence can be used to further decrease the size of the factor base by a factor 2.

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 intersection 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 on the website of the third author 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. The FGb library also provides a precise count of the number of basic operations (a multiplication of two 32-bit integers 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 $n = 4$, as mentioned in [34] 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 [31] based on Weierstrass representation (denoted W. [31], 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. As previously announced, we observe that $\mathscr{S}_{\mathfrak{S}_n} \in \mathbb{K}[e_1,\ldots,e_n]$ is a regular sequence. Which is not the case of $\mathscr{S}_{\mathfrak{S}_n} \in \mathbb{K}[p_1,\ldots,p_n]$. Hence, following results in Section 3, we equipped the ring $\mathbb{K}[s_1,\ldots,s_{n-1},e_n]$ with the weighted degree with weights $(2,\ldots,2,1)$. While the ring $\mathbb{K}[p_2,\ldots,p_{2(n-1)},p_n]$ is equipped with the usual degree. The results are given in Table 1, where one finds for various sizes of the base field the runtimes and the maximal (weighted) degree reached by polynomials during the computation of a (W)DRL Gröbner basis with F_4. In column d_{max}/d_{theo} one can find the maximal (weighted) degree reached by the polynomials and when the system is regular the bound on this maximal degree given by Theorem 3.1. The two last columns of Table 1 give the number of multiplications of two 32-bits words required to solve the corresponding polynomial system. The penultimate column gives an interpolated number of multiplications of two 32-bits words required by the MAGMA software. Since we observe that the most consuming step is the change of ordering we interpolate this number thanks to the complexity of the FGLM algorithm in $O(nD^3)$ arithmetic operations. The last column gives the exact number of multiplications of two 32-bits words required by the FGb implementation. Since, FGb library uses the recent sparse change of ordering algorithm in [26] its practical arithmetic complexity is closer to be quadratic in the number of solutions than cubic.

$\log_2(q)$	weights	F_4		d_{max}/d_{theo}		Change Order		Total		#ops	# ops FGb	
		s_i $(2,\ldots,2,1)$	p_i $(1,\ldots,1)$	s_i $(2,\ldots,2,1)$	p_i $(1,\ldots,1)$	s_i $(2,\ldots,2,1)$	p_i $(1,\ldots,1)$	s_i $(2,\ldots,2,1)$	p_i $(1,\ldots,1)$	MAGMA	s_i $(2,\ldots,2,1)$	p_i $(1,\ldots,1)$
16	W. [31]	5s		29/29		423s		428s		2^{36}	2^{29}	
	T_2	<1s	<1s	26/27	14	1s	3s	$<$ **2**s	<4s	2^{27}	2^{24}	2^{26}
	$T_{2,4}$	<1s	1s	21	15	2s	3s	<3s	4s		2^{24}	2^{27}
64	W. [31]	331s		29/29		5994s		6325s		2^{40}	2^{33}	
	T_2	2s	32s	26/27	14	13s	24s	**15**s	56s	2^{31}	2^{28}	2^{30}
	$T_{2,4}$	8s	61s	21	15	12s	25s	20s	86s		2^{28}	2^{31}
128	W. [31]	480s		29/29		7179s		7559s		2^{42}	2^{35}	
	T_2	2s	40s	26/27	14	14s	32s	**16**s	72s	2^{33}	2^{30}	2^{32}
	$T_{2,4}$	9s	80s	21	15	16s	32s	25s	112s		2^{30}	2^{33}

TABLE 1. Computing time of Gröbner basis with MAGMA (V2-19.1) on one core of a 2.00 GHz Intel® E7540 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 400. This is consistent with the theoretical expected gain, as shown by the interpolated number

of multiplications of two 32-bits words required by MAGMA which is divided by $2^9 = 2^{3(n-1)}$; and also shown by the exact number of multiplications of two 32-bits words required by FGb which is divided by 2^5 of the order of $2^{2(n-1)}$ corresponding to a quadratic complexity for the change of ordering.

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. Moreover, one of the sequence is regular while the other is not. As a consequence, the DRL part of the computation is more costly for the p_i than for the s_i. One can notice that for the systems expressed in terms of the primary invariant of $\mathfrak{S}_n$ and the systems expressed in terms of the primary invariants of D_n, $s_1, \ldots, s_{n-1}, e_n$, the maximal (weighted) degree reached by the polynomials during the computation of a degree monomial ordering Gröbner basis is tightly bounded by the bound of Theorem 3.1. We observe that the system $\mathscr{S}_{\mathfrak{S}_n}$ (resp. $\mathscr{S}_{D_n}$) is regular when we consider the usual degree (resp. the weighted degree with weights $(2, \ldots, 2, 1)$).

Moreover, we notice that the change of ordering step is the most time consuming step which is consistent with the complexity analysis of Theorem 4.1. This shows that it is important to have precise complexity bound for the change of ordering. Moreover, the complexity of change of ordering depends on the number of solutions of the system so this emphasizes the impact of the action of a pseudo reflective group.

One can notice that adding a variable to decrease the degree of polynomials in the computation of Gröbner basis (to use the 4-torsion) does not speed up the computation in this case. Indeed, adding the variable v_4 breaks the quasi-homogeneous structure since we do not find an appropriate weight for this variable. Hence, in the following the 4-torsion point is used only to further decrease the size of the factor base. That is to say, we change the representation as presented in the previous section but we do not add the variable v_4. In this context the 4-torsion can be used for any n.

It can be observed that 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, we consider only the s_i basis.

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 [34]. This approach can be seen as an hybrid approach where one mixes an exhaustive search and an algebraic resolution (*e.g.* see [6] for application of such a strategy in another context). 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 computations are not feasible with MAGMA and we use the FGb library. Actually, the graded reverse lexicographical Gröbner basis can be computed with MAGMA but the change of ordering cannot. The timings are given in table 2.

$\log_2(q)$		F_5	d_{max}/d_{theo}	Change-Order	Total	# ops
16	W. [31]	>2 days	??/76			
	T_2	567s	72/73	2165s	2732s	2^{44}

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

For $n = 5$ Theorem 3.1 gives also a precise bound on the maximal degree reached by the polynomials. The regular hypothesis has been checked also on these systems.

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

$\log_2(q)$		F_4	# ops
16	W. [34]	13.400s	2^{32}
	T_2	**0.090**s	2^{22}
	$T_{2,4}$	0.130s	2^{24}
32	W. [34]	1278s	2^{34}
	T_2	**1.100**s	2^{24}
	$T_{2,4}$	1.760s	2^{26}

TABLE 3. Computing time of Gröbner basis with MAGMA (V2-19.1) on one core of a 2.00 GHz Intel® E7540 CPU for $n = 5$ and decomposition in $n-1$ points. Operation counts are obtained using FGb.

Using symmetries decreases the running time also for decompositions in $n-1$ points. For $n = 5$, the speed-up is by a factor about 150 for a 16-bit base field and by 1000 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 40 minutes.

$\log_2(q)$		F_5	# ops
		s_i	s_i
16	W. [34]	> 2 days	
	T_2	2448s	2^{39}

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.

In Table 3, we can observe that considering the action of 4-torsion points of Jacobi intersections curves is more time consuming. Indeed, if the system admits a solution then it also admits all the solutions associated to the action of the 4-torsion points. By consequence, the overdetermined systems have not the same DRL and LEX Gröbner basis and their computation are slower. By consequence, for the "$n-1$" variant, the trade-off between the size of the factor base and the difficulty of decomposing a point is better when using only the 2-torsion.

Indeed, when we consider only the action of T_2, we use the factor base $\mathscr{F} = \mathscr{F}_1$ ($\mathscr{F}_0$ is too small). Hence, the action of T_4 does not leave the factor base invariant. Moreover, the decompositions related to the action of the 4-torsion do not necessarily correspond to solutions of the system obtained after the Weil restriction on summation polynomials. In fact, we observe that the corresponding system has the expected number of solutions that is 0 or 1.

Remark 7. *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 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 does 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 $E(\mathbb{F}_{q^n})$. The cost of an elliptic curve operation can be approximated by $\log_2(q^n)^2$. 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

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

According to Remark 5 and the end of Section 4, for index calculus using the point decomposition in n points we look for N relations where N is:

- $\frac{q}{2}$ for Weierstrass representation,
- $\frac{q}{4}$ for twisted Edwards curves,

- $\frac{q}{8}$ for twisted Jacobi intersections curves and by using only the 2-torsion,
- $\frac{q}{16}$ for twisted Jacobi intersections curves and by using the 2-torsion and the 4-torsion.

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 $\mathbb{F}_{q^n}$:

$$N \cdot n! \cdot c(n,q,n) + n^3 \log_{2^{32}}(q)^2 N^2 .$$

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

$$q(n-1)! \cdot N \cdot c(n,q,n-1) + n^2(n-1)\log_{2^{32}}(q)^2 \cdot N^2 .$$

When the linear algebra step is more time consuming than the relation search, by using the double large prime variation [32] we can rebalance the costs of these two steps (see [48, 32]). 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:

$$\log_2(q)\left(1 + r\frac{n-1}{n}\right)(n-2)!q^{1+(n-2)(1-r)}c(n,q,n) + n^3\log_{2^{32}}(q)^2 N^{2r}$$

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

The results are summarized in Table 5. The notations 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 parameter is given by index calculus using symmetries induced by the 2-torsion points (and 4-torsion point when decomposing in n points is possible) or generic algorithms. We note that for $n \leq 5$ our version of the index calculus 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^{116} by using the 2-torsion points and 4-torsion point. In this case, our approach is more efficient than the basic index calculus, solving this instance of ECDLP in 2^{121} 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^{39} over generic algorithms and a gain of 2^{40} 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

Curve parameters		Curve representation and torsion used	Generic algorithm	Linear algebra	Relations search decomposition in		Double large prime variation	Total DLP
					n points	$n-1$ points		
n	$\log_2(q)$							
4	32	Weierstrass	2^{68}	2^{68}	2^{67} [31]			2^{68}
		T_2 Edwards		2^{66}	2^{61}		2^{66}	2^{66}
		T_2 Jacobi		2^{64}	2^{60}		2^{64}	2^{64}
		$T_{2,4}$ Jacobi		2^{62}	2^{59}			$\mathbf{2^{62}}$
	64	Weierstrass	2^{134}	2^{134}	2^{101} [31]		2^{121}	2^{121}
		T_2 Edwards		2^{132}	2^{95}		2^{118}	2^{118}
		T_2 Jacobi		2^{130}	2^{94}		2^{117}	2^{117}
		$T_{2,4}$ Jacobi		2^{128}	2^{93}		2^{116}	$\mathbf{2^{116}}$
	128	Weierstrass	2^{264}	2^{264}	2^{167} [31]		2^{220}	2^{220}
		T_2 Edwards		2^{262}	2^{161}		2^{216}	2^{216}
		T_2 Jacobi		2^{260}	2^{160}		2^{215}	$\mathbf{2^{215}}$
		$T_{2,4}$ Jacobi		2^{258}	2^{159}		2^{215}	$\mathbf{2^{215}}$
5	32	Weierstrass	2^{85}	2^{69}	∞	2^{102} [34]		2^{85}
		T_2 Edwards		2^{67}	2^{83}	2^{91}		2^{83}
		T_2 Jacobi		2^{65}	2^{82}	2^{90}		2^{82}
		$T_{2,4}$ Jacobi		2^{63}	2^{81}	2^{92}		$\mathbf{2^{81}}$
	64	Weierstrass	2^{167}	2^{135}	∞	2^{168} [34]		2^{167}
		T_2 Edwards		2^{133}	2^{117}	2^{157}	2^{130}	2^{130}
		T_2 Jacobi		2^{131}	2^{116}	2^{156}	2^{129}	2^{129}
		$T_{2,4}$ Jacobi		2^{129}	2^{115}	2^{158}	2^{128}	$\mathbf{2^{128}}$
	128	Weierstrass	2^{329}	2^{265}	∞	2^{298} [34]		2^{298}
		T_2 Edwards		2^{263}	2^{183}	2^{287}	2^{235}	2^{235}
		T_2 Jacobi		2^{261}	2^{182}	2^{286}	2^{234}	2^{234}
		$T_{2,4}$ Jacobi		2^{259}	2^{181}	2^{288}	2^{233}	$\mathbf{2^{233}}$
6	32	Weierstrass	2^{102}	2^{70}	∞	∞		$\mathbf{2^{102}}$
		T_2 Edwards		2^{68}	∞	2^{110}		$\mathbf{2^{102}}$
		T_2 Jacobi		2^{66}	∞	2^{109}		$\mathbf{2^{102}}$
	64	Weierstrass	2^{200}	2^{136}	∞	∞		2^{200}
		T_2 Edwards		2^{134}	∞	2^{176}		2^{176}
		T_2 Jacobi		2^{132}	∞	2^{175}		$\mathbf{2^{175}}$
	128	Weierstrass	2^{394}	2^{266}	∞	∞		2^{394}
		T_2 Edwards		2^{264}	∞	2^{306}		2^{306}
		T_2 Jacobi		2^{262}	∞	2^{305}		$\mathbf{2^{305}}$

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

have a complexity in $O\left(q^3\right)$, while the index calculus attack using the decomposition in $n-1$ points has a complexity in $O\left(\mathrm{C}(n)\cdot q^2\right)$ where $\mathrm{C}(n)$ is exponential in n. Hence to be better than generic algorithms, we have to consider high values of q and consequently high security levels. For instance if $\log_2(q)=64$, the index calculus attack using symmetries of twisted Edwards or twisted Jacobi intersections curves and decomposition in $n-1$ points needs less operations (2^{176}) than the 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.

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

Security level		2^{80}			2^{112}		
n		4	5	6	4	5	6
Generic Algorithm	$\log_2(q)$	38	31	**26**	54	43	**36**
Index Calculus		**42**	**32**	19	**62**	**56**	34

Security level		2^{128}			2^{192}		
n		4	5	6	4	5	6
Generic Algorithm	$\log_2(q)$	62	49	41	93	74	62
Index Calculus		**72**	**64**	**42**	**113**	**103**	**73**

TABLE 6. Domain 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 the attack based on generic algorithms with the attack based on the best version of index calculus. For the index calculus attack, except for $n=6$, the size of q is obtained by considering decomposition in n points using the symmetries (2-torsion and 4-torsion) of twisted Jacobi intersections curves. This table confirms the previous observations. For $n=4,5$, the size of q is increased because of the new version of index calculus proposed in this work. For $n=6$ this is true only for very high security level.

6. PERSPECTIVES

We have highlighted some geometrical properties of twisted Edwards and Jacobi intersections curves implying new symmetries simplifying the resolution of the *Point Decomposition Problem*. 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 here.

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 the $n-1$ variant, one can use the index calculus attack only for elliptic curves defined over $\mathbb{F}_{q^n}$ with $n < 6$. Thus to further 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 2.)

[2] L. 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. On the complexity of gröbner basis computation of semi-regular overdetermined algebraic equations. In *International Conference on Polynomial System Solving - ICPSS*, pages 71 –75, November 2004. (Cited on page 12.)

[4] D. 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 8.)

[5] 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 8.)

[6] L. Bettale, J.-C. Faugère, and L. Perret. Hybrid approach for solving multivariate systems over finite fields. *Journal of Mathematical Cryptology*, volume 3(issue 3):177–197, 2009. (Cited on page 32.)

[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 5.)

[8] C. Chevalley. Invariants of finite groups generated by reflections. *American Journal of Mathematics*, 77(4):pp. 778–782, 1955. (Cited on page 16.)

[9] D. Chudnovsky and G. 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 9.)

[10] A. Cohen, H. Cuypers, and H. Sterk. *Some Tapas of Computer Algebra*. Algorithms and Computation in Mathematics Series. Springer, 2011. (Cited on page 23.)

[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 4.)

[13] D. A. Cox, J. Little, and D. O'Shea. *Ideals, varieties, and algorithms - an introduction to computational algebraic geometry and commutative algebra (2. ed.)*. Undergraduate texts in mathematics. Springer, 1997. (Cited on pages 10 and 12.)

[14] 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.)

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

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

[17] 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.)

[18] H. 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 8.)

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

[20] 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.)

[21] 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 11.)

[22] 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 11.)

[23] J.-C. Faugère. FGb: A library for computing Gröbner bases. In K. Fukuda, J. Hoeven, M. Joswig, and N. 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 5.)

[24] J.-C. Faugère, P. Gaudry, L. Huot, and G. Renault. Fast change of ordering with exponent ω. *ACM Commun. Comput. Algebra*, 46:92–93, September 2012. (Cited on pages 3, 4, 11, and 14.)

[25] 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, 11, and 14.)

[26] 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, 11, and 31.)

[27] 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 15.)

[28] J.-C. Faugère, M. Safey El Din, and T. Verron. Computing Gröbner bases for quasi-homogeneous systems, Jan. 2013. `http://arxiv.org/abs/1301.5612`. (Cited on pages 5 and 13.)

[29] 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 9.)

[30] G. Frey. Applications of arithmetical geometry to cryptographic constructions. In *International Conference on Finite Fields and Applications*, pages 128–161, 2001. (Cited on page 6.)

[31] 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, 3, 22, 27, 30, 31, 33, and 36.)

[32] 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 35.)

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

[34] A. Joux and V. Vitse. Elliptic curve discrete logarithm problem over small degree extension fields. *Journal of Cryptology*, 26(1):119–143, 2013. (Cited on pages 5, 26, 30, 32, 33, 34, and 36.)

[35] R. Kane. *Reflection Groups and Invariant Theory*. Springer, 2001. (Cited on page 5.)

[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. Gröbner bases, gaussian elimination and resolution of systems of algebraic equations. In J. van Hulzen, editor, *Computer Algebra*, volume 162 of *Lecture Notes in Computer Science*, pages 146–156. Springer Berlin / Heidelberg, 1983. (Cited on page 13.)

[39] V. Miller. Use of elliptic curves in cryptography. In *Lecture notes in computer sciences; 218 on Advances in cryptology—CRYPTO 85*, pages 417–426, New York, NY, USA, 1986. Springer-Verlag New York, Inc. (Cited on page 2.)

[40] P. Montgomery. Speeding the Pollard and elliptic curve methods of factorization. *Mathematics of computation*, 48(177):243–264, 1987. (Cited on page 8.)

[41] K. Nagao. Decomposed attack for the jacobian of a hyperelliptic curve over an extension field. In G. Hanrot, F. Morain, and E. 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] N. I. 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. Pollard. Monte carlo methods for index computation mod p. *Math. Comp.*, 32(143):918–924, July 1978. (Cited on page 2.)

[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 7.)

[45] G. C. Shephard and J. A. Todd. Finite unitary reflection groups. *Canadian J. Math.*, 6:274–304, 1954. (Cited on page 16.)

[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 2.)

[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 5, 10, and 17.)

[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 35.)

[49] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, 2002. (Cited on page 11.)

[50] D. Wiedemann. Solving sparse linear equations over finite fields. *IEEE Trans. Inf. Theor.*, 32(1):54–62, 1986. (Cited on page 35.)