Summation polynomial algorithms for elliptic curves in characteristic two

Steven D. Galbraith and Shishay W. Gebregiyorgis

Mathematics Department,
University of Auckland,
New Zealand.
S.Galbraith@math.auckland.ac.nz,sgeb522@aucklanduni.ac.nz

Abstract. The paper is about the discrete logarithm problem for elliptic curves over characteristic 2 finite fields $\mathbb{F}_{2^n}$ of prime degree n. We consider practical issues about index calculus attacks using summation polynomials in this setting. The contributions of the paper include: a choice of variables for binary Edwards curves (invariant under the action of a relatively large group) to lower the degree of the summation polynomials; a choice of factor base that "breaks symmetry" and increases the probability of finding a relation; an experimental investigation of the use of SAT solvers rather than Gröbner basis methods for solving multivariate polynomial equations over $\mathbb{F}_2$.

We show that our choice of variables gives a significant improvement to previous work in this case. The symmetry-breaking factor base and use of SAT solvers seem to give some benefits in practice, but our experimental results are not conclusive. Our work indicates that Pollard rho is still much faster than index calculus algorithms for the ECDLP (and even for variants such as the oracle-assisted static Diffie-Hellman problem of Granger and Joux-Vitse) over prime extension fields $\mathbb{F}_{2^n}$ of reasonable size.

Keywords: ECDLP, summation polynomials, index calculus.

1 Introduction

Let E be an elliptic curve over a finite field $\mathbb{F}_{2^n}$ where n is prime. The elliptic curve discrete logarithm problem (ECDLP) is: Given $P, Q \in E(\mathbb{F}_{2^n})$ to compute an integer a, if it exists, such that $Q = aP$. As is standard, we restrict attention to points P of prime order r. The Diffie-Hellman problem (CDH) is: Given $P \in E(\mathbb{F}_{2^n})$ and points $P_1 = aP$ and $P_2 = bP$, for some integers a and b, to compute abP. These two computational problems are fundamental to elliptic curve cryptography. There is a wide variety of "interactive" Diffie-Hellman assumptions, meaning that the attacker/solver is given access to an oracle that will perform various computations for them. These problems also arise in some cryptographic settings, and it is interesting to study them (for scenarios where they arise in practice for static-CDH see Brown and Gallant [3]). These problems are surveyed by Koblitz and Menezes [23] and we recall some of them now.

- The "Delayed Target One-More Discrete Logarithm Problem" in the sense of Joux-Naccache-Thomé is the following. The solver is supplied with a discrete logarithm oracle and must find the discrete logarithm of a random group element Y that is given to the solver only after all the queries to the oracle have been made.
- The "oracle-assisted static Diffie-Hellman problem" (also called the "delayed target One-More Diffie-Hellman problem") is the following. The solver is given $(P, X = aP)$ and a static (also called "one-sided") Diffie-Hellman oracle (i.e., $O(Y) = aY$), and must solve the DHP with input (P, X, Y), where Y is a random group element that is given to the solver only after all the queries to O have been made. In other words, the solver must compute $Z = aY$.
- The "static One-More Diffie-Hellman Problem" is as follows. The solver is again given $(P, X = aP)$ and access to an oracle $O(Y) = aY$, and also a challenge oracle that produces random group elements Y_i . After t queries to the challenge oracle (where t is chosen by the solver) and at most $t - 1$ queries to the DHP oracle O, the solver must find $Z_i = aY_i$ for all $i = 1, \dots, t$.

Early papers on attacking these sorts of interactive assumptions (e.g., [20]) used index calculus algorithms for finite fields. Granger [18] and Joux-Vitse [22] were the first to consider the case of elliptic curve groups $E(\mathbb{F}_{q^n})$ (both papers mainly focus on the case where q is a large prime, and briefly mention small characteristic but not prime degree extension fields $\mathbb{F}_{2^n}$).

One approach to solving the ECDLP (or these interactive assumptions) is to use Semaev's summation polynomials [27] and index calculus ideas of Gaudry, Diem, and others [5–8, 11–13, 16, 21, 22, 25]. The main idea is to specify a factor base and then to try and "decompose" random points $R = uP + wQ$ as a sum $P_1 + \cdots + P_m$ of points in the factor base. Semaev's summation polynomials allow to express the sum $P_1 + \cdots + P_m - R = \infty$, where ∞ is the identity element, as a polynomial equation over $\mathbb{F}_{2^n}$, and then Weil descent reduces this problem to a system of polynomial equations over $\mathbb{F}_2$

There is a growing literature on these algorithms. Much of the previous research has been focussed on elliptic curves over $\mathbb{F}_{q^n}$ where q is prime or a prime power, and n is small.

Our Work This paper is about the case $\mathbb{F}_{2^n}$ where n is prime. Other work (for example [6–8, 25]) has focused on asymptotic results and theoretical considerations. Instead, we focus on very practical issues and ask about what can actually be computed in practice today. In other words, we follow the same approach as Huang, Petit, Shinohara and Takagi [19] and Shantz and Teske [26].

We assume throughout that the ECDLP instance cannot be efficiently solved using the Gaudry-Hess-Smart approach [15] or its extensions, and that the point decomposition step of the algorithm is the bottleneck (so we ignore the cost of the linear algebra). This will be the case in our examples.

The goal of our paper is to report on our experiments with three ideas:

(1) We describe a choice of variables for binary Edwards curves that is invariant under the action of a relatively large group (generated by the action of the symmetric group and addition by a point of order 4). This allows the summation polynomials to be re-written with lower degree, which in turn speeds up the computation of relations.

(2) We consider a factor base that "breaks symmetry" and hence significantly increases the probability that relations exist. It may seem counterintuitive that one can use symmetric variables to reduce the degree and also a non-symmetric factor base, but if one designs the factor base correctly then this is seen to be possible.

The basic idea is as follows. The traditional approach has relations $R = P_1 + \cdots + P_m$ where $P_i \in \mathcal{F} = \{P \in E(\mathbb{F}_{2^n}) : x(P) \in V\}$ where $V \subseteq \mathbb{F}_{2^n}$ is some $\mathbb{F}_2$-vector subspace of dimension l. Instead, we demand $P_i \in \mathcal{F}_i$ over $1 \leq i \leq m$ for m different factor bases $\mathcal{F}_i = \{P \in E(\mathbb{F}_{2^n}) : x(P) \in V + v_i\}$ where $v_i \in \mathbb{F}_{2^n}$ are elements of a certain form so that the sets $V + v_i$ are all distinct. (Diem [8] has also used different factor bases $\mathcal{F}_i$, but in a different way.) The probability of finding a relation is increased by a factor approximately $m!$, but we need m times as many relations, so the total speedup is approximately by a factor of $(m - 1)!$.

(3) We experiment with SAT solvers rather than Gröbner basis methods for solving the polynomial systems. This is possible since we obtain a system of multivariate polynomial equations over $\mathbb{F}_2$, rather than over larger fields. (SAT solvers have been considered in cryptanalysis before, e.g. [4, 24].)

Our conclusions are: The suggested coordinates for binary Edwards curves give a significant improvement over previous work on elliptic curves in characteristic 2. The use of SAT solvers may potentially enable larger factor bases to be considered (however, it seems an "early abort" strategy should be taken, as we will explain). Symmetry breaking seems to give a moderate benefit when n is large compared with lm.

Finally, our overall conclusion is that, for parameters of interest for actual computation, it is slower to use summation polynomials to solve an ECDLP instance (or even the interactive assumptions mentioned earlier) in characteristic 2 elliptic curves than to use Pollard rho. Hence, summation polynomial algorithms do not seem to be a useful tool for attacking current ECDLP challenge curves for curves defined over $\mathbb{F}_{2^n}$ where n is prime.

The paper is organised as follows. Section 2 recalls previous work. Section 3 recalls binary Edwards curves and introduces our new variables. Section 4 shows how to do the index calculus attack in this setting and discusses the symmetry-breaking idea. Section 5 discusses the use of SAT solvers, while Section 6 reports on our experimental results.

2 Index Calculus Algorithms and Summation Polynomials

We briefly recall the basic ideas of these methods and introduce our notation. Let $P \in E(\mathbb{F}_{2^n})$ have prime order r and suppose $Q = aP$. One chooses an appropriate factor base $\mathcal{F} \subseteq E(\mathbb{F}_{2^n})$, computes random points $R = uP + wQ$ and then tries to write $R = P_1 + \cdots + P_m$ for $P_i \in \mathcal{F}$. Each successful decomposition of the point R is called a "relation". Let $\ell = \#\mathcal{F}$. Writing $\mathcal{F} = \{F_1, \ldots, F_\ell\}$ we can write the j-th relation as $u_jP+w_jQ = \sum_{i=1}^{\ell} z_{j,i}F_i$ and store the relation by storing the values (u_j, w_j) and the vector $(z_{j,1}, \ldots, z_{j,\ell})$. When enough relations (more than ℓ) are found then one can apply (sparse) linear algebra to find a kernel vector of the matrix $M = (z_{j,i})$ and hence obtain a pair of integers u and w such that $uP + wQ = 0$ from which we can solve for $a \equiv -uw^{-1} \pmod{r}$ as long as $w \not\equiv 0 \pmod{r}$. The details are standard.

One can use this approach to solve interactive Diffie-Hellman assumptions. We give the details in the case of the oracle-assisted static Diffie-Hellman. Choose a factor base $\mathcal{F} = \{F_1, \ldots, F_\ell\}$ then call the oracle for each element F_i to get the points aF_i for $1 \le i \le \ell$. When provided with the challenge point Y one tries to decompose $Y = P_1 + \cdots + P_m$ for points $P_i \in \mathcal{F}$. If such a relation is found then we can compute the required point aY as $aP_1 + \cdots + aP_m$. (If we fail to find a relation then we can randomise by taking $Y + uP$ and recalling that $uX = uaP$.)

One sees that all applications require decomposing random points over the factor base. This is the difficult part of the algorithm and is the main focus of our paper. Note however that the ECDLP application requires a very large number of relations and hence a very large number of point decompositions, whereas the oracle-assisted static-DH application only requires a single relation.

We will ignore the linear algebra step as, for the parameters considered in the paper, its cost will always be insignificant.

2.1 Summation Polynomials

Let E be an elliptic curve in Weierstrass form over a field $\mathbb{K}$ of odd characteristic. The m^{th} summation polynomial $f_m(x_1, x_2, \ldots, x_m) \in \mathbb{K}[x_1, x_2, \ldots, x_m]$ for E, defined by Semaev [27], has the following defining property. Let $X_1, X_2, \ldots, X_m \in \mathbb{K}$. Then $f_m(X_1, X_2, \ldots, X_m) = 0$ if and only if there exist $Y_1, Y_2, \ldots, Y_m \in \overline{\mathbb{K}}$ such that $(X_i, Y_i) \in E(\overline{\mathbb{K}})$ for all $1 \le i \le m$ and $(X_1, Y_1) + (X_2, Y_2) + \cdots + (X_m, Y_m) = \infty$, where ∞ is the identity element.

Lemma 1. *(Semaev [27]) Let $E : y^2 = x^3 + a_4x + a_6$ be an elliptic curve over a field $\mathbb{K}$ of characteristic $\neq 2, 3$ and $\{a_4, a_6\} \in \mathbb{K}$. The summation polynomials for E are given as follows.*

$$\begin{aligned} f_2(X_1, X_2) &= X_1 - X_2 \\ f_3(X_1, X_2, X_3) &= (X_1 - X_2)^2 X_3^2 - 2((X_1 + X_2)(X_1X_2 + a_4) + 2a_6)X_3 \\ &\quad +((X_1X_2 - a_4)^2 - 4a_6(X_1X_2)). \end{aligned}$$

For $m \geq 4$ and a constant j such that $1 \leq j \leq m-3$, then

$$f_m(X_1, \ldots, X_m) = \textit{Resultant}_X(f_{m-j}(X_1, \ldots, X_{m-j-1}, X), f_{j+2}(X_{m-j}, X_{m-j+1}, \ldots, X_m, X)).$$

For $m \geq 2$, the m^{th} summation polynomial f_m is an irreducible symmetric polynomial that has degree 2^{m-2} in each of the variables.

Gaudry and Diem noted that, for elliptic curves $E(\mathbb{F}_{q^n})$ over extension fields, there are choices of factor base for which the problem of finding solutions to summation polynomials can be approached using Weil descent with respect to $\mathbb{F}_{q^n}/\mathbb{F}_q$. In other words, the problem of solving $f_{m+1}(x_1, \ldots, x_m, x(R))$ for $x_i \in \mathbb{F}_q$ can be reduced to a system of multivariate polynomial equations over $\mathbb{F}_q$. The details are standard.

To solve the system of multivariate polynomial equations, the current most effective approach (see [11, 19]) is to perform the F_4 or F_5 algorithm for the graded reverse lex order, followed by the FGLM algorithm [14].

2.2 Degree Reduction Via Symmetries

The summation polynomials have high degree, which makes solving them difficult. Since the summation polynomial is invariant under the action of the symmetric group S_m, Gaudry [16] observed that re-writing the polynomial in terms of invariant variables reduces the degree and speeds up the resolution of the system of equations. As well as lowering the degree of the polynomials, this idea also makes the solution set smaller and hence faster to compute using the FGLM algorithm.

Faugère et al [12, 13] have considered action by larger groups (by using points of small order) for elliptic curves over $\mathbb{F}_{q^n}$ where n is small (e.g., $n = 4$ or $n = 5$) and the characteristic is $\neq 2, 3$. Their work gives further reduction in the cost of solving the system. We sketch (for all the details see [12, 13]) the case of points of order 2 on twisted Edwards curves.

For a point $P = (x, y)$ on a twisted Edwards curve we have $-P = (-x, y)$ and so it is natural to construct summation polynomials in terms of the y-coordinate (invariant under $P \mapsto -P$). Accordingly Faugère et al [12] define their factor base as

$$\mathcal{F} = \{P = (x, y) \in \mathbb{F}_{q^n} : y \in \mathbb{F}_q\}.$$

Further, the addition of P with the point $T_2 = (0, -1)$ (which has order 2) satisfies $P + T_2 = (-x, -y)$. Note that $P \in \mathcal{F}$ if and only if $P + T_2 \in \mathcal{F}$. Hence, for each decomposition $R = P_1 + P_2 + \cdots + P_n$, there exist 2^{n-1} further decompositions, such as

$$R = (P_1 + T_2) + (P_2 + T_2) + P_3 + \cdots + P_n.$$

It follows that the dihedral coxeter group $D_n = (\mathbb{Z}/2\mathbb{Z})^{n-1} \rtimes S_n$ of order $2^{n-1}n!$ acts on the set of relations $R = P_1 + \cdots + P_n$ for any given point R (and all these relations correspond to solutions of the summation polynomial). It is therefore natural to try to write the summation polynomial $f_{n+1}(y_1, y_2, \ldots, y_n, y(R))$ in terms of new variables that are invariant under the group action. For further details see [12].

A recent idea (originating in the work of Joux-Vitse [21] for $E(\mathbb{F}_{q^n})$) is to consider relations with fewer summands $R = P_1 + \cdots + P_m$ with $m < n$. Joux and Vitse take $m = n - 1$ so the probability of a relation is reduced from $1/n!$ to $1/(q(n-1)!)$. The cost of solving the polynomial system is significantly reduced, but the running time increases by the factor q. Shantz and Teske [26] call this the "delta method".

2.3 The Case of $\mathbb{F}_{2^n}$ where n is Prime

Following Diem [8] we define the factor base in terms of an $\mathbb{F}_2$-vector space $V \subset \mathbb{F}_{2^n}$ of dimension l. A typical choice for the factor base in the case of Weierstrass curves is $\mathcal{F} = \{P \in E(\mathbb{F}_{2^n}) : x(P) \in V\}$, and one wants to decompose random points as $R = P_1 + \cdots + P_m$ for $P_i \in \mathcal{F}$.

As above, the symmetric group S_m of order $m!$ acts on the set of relations $R = P_1 + \cdots + P_m$ for any given point R (and all these relations correspond to solutions of the summation polynomial). It is therefore natural to try to write the summation polynomial $f_{m+1}(x_1, x_2, \ldots, x_m, x(R))$ in terms of new variables that are invariant under the group action. In this example, such variables are the elementary symmetric polynomials in the x_i. This approach gives polynomials of lower degree.

Huang et al [19] observe that it is hard to combine re-writing the summation polynomial in terms of symmetric variables and also using a factor base defined with respect to an arbitrary vector subspace of $\mathbb{F}_{2^n}$. The point is that if $x_1, \ldots, x_m \in V$ then it is not necessarily the case that the value of the symmetric polynomial $e_2 = x_1x_2 + x_1x_3 + \cdots + x_{m-1}x_m$ (or higher ones) lies in V. Hence, one might think that one cannot use symmetries in this setting.

Section 3 of [19] considers prime n and the new idea of "both symmetric and non-symmetric variables". It is suggested to use a "special subspace" V that behaves relatively well under multiplication: $x_i, x_j \in V$ implies $x_ix_j \in V'$ for a somewhat larger space V'. The experiments in [19], for n prime in the range $17 \leq n \leq 53$, $m = 3$, and $l \in \{3, 4, 5, 6\}$, show a significant decrease of the degree of regularity (the highest degree reached) during Gröbner basis computations. However, the decrease in the degree of regularity is at the expense of an increased number of variables, which in turn increases the complexity of the Gröbner basis computations (which roughly take time N^{3D} and require N^{2D} memory, where N is the number of variables and D is the degree of regularity).

Huang et al [19] exploit the action of S_m on the summation polynomials but do not exploit points of order 2 or 4. One of our contributions is to give coordinates that allow to exploit larger symmetry groups in the case of elliptic curves over binary fields. We are able to solve larger experiments in this case (e.g., taking decompositions into $m = 4$ points, while [19] could only handle $m = 3$). For more details of our experiments see Section 6.

3 Edwards Elliptic Curves in Characteristic Two

We study binary Edwards curves [1] since the addition by points of order 2 and 4 is nicer than when using the Weierstrass model as was done in [12, 13]. Hence we feel this model of curves is ideally suited for the index calculus application.

Definition 1. *Let $d_1, d_2 \in \mathbb{F}_{2^n}$ be such that $d_1 \neq 0$ and $d_2 \neq d_1^2 + d_1$. The binary Edwards curve with coefficients d_1 and d_2 is the elliptic curve given by the affine model*

$$E_{d_1,d_2} \,:\, d_1(x + y) + d_2(x^2 + y^2) = xy + xy(x + y) + x^2y^2.$$

The binary Edwards curve is symmetric in the variables x and y with the following group law [1].

1. The identity element is the point $P_0 = (0, 0)$.
2. For a point $P = (x, y) \in E_{d_1,d_2}$, its negation is given by $-P = (y, x)$. We have $P + -P = P_0 = (0, 0)$.
3. Let $P_1 = (x_1, y_1), P_2 = (x_2, y_2) \in E_{d_1,d_2}$, then $P_3 = (x_3, y_3) = P_1 + P_2$ is given by

$$x_3 = \frac{d_1(x_1 + x_2) + d_2(x_1 + y_1)(x_2 + y_2) + (x_1 + x_1^2)(x_2(y_1 + y_2 + 1) + y_1y_2)}{d_1 + (x_1 + x_1^2)(x_2 + y_2)}$$

$$y_3 = \frac{d_1(y_1 + y_2) + d_2(x_1 + y_1)(x_2 + y_2) + (y_1 + y_1^2)(y_2(x_1 + x_2 + 1) + x_1x_2)}{d_1 + (y_1 + y_1^2)(x_2 + y_2)}.$$

4. The point $T_2 = (1,1) \in E_{d_1,d_2}$ is invariant under negation so it has order 2. For any point $P = (x, y) \in E_{d_1,d_2}$ we have $P + T_2 = (x+1, y+1)$.

If $d_1 \neq 0$ and $\mathrm{Tr}_{\mathbb{F}_{2^n}/\mathbb{F}_2}(d_2) = 1$, i.e., there is no element $u \in \mathbb{F}_{2^n}$ such that u satisfies $u^2 + u + d_2 = 0$, then the addition law on the binary Edwards curve is complete [1]. That is, the denominators in the addition law $d_1 + (y_1 + y_1^2)(x_2 + y_2)$ and $d_1 + (x_1 + x_1^2)(x_2 + y_2)$ never vanish.

For summation polynomials with these curves, the best choice of variable is $t = x + y$. This is the natural choice, consistent with previous work [16, 12], as this function is invariant under the action of $[-1] : P \mapsto -P$. The coordinate t was used in [1] for differential addition, but it was called ω.

The function $t : E_{d_1,d_2} \to \mathbb{P}^1$ has degree 4. Given a value $t \in \mathbb{F}_{2^n}$ there are generically four points $P = (x, y) \in E(\overline{\mathbb{F}}_2)$ having the same value for $t(P)$, namely $(x, y), (y, x), (x + 1, y + 1), (y + 1, x + 1)$.

When we come to define the factor base, we will choose a vector subspace V of $\mathbb{F}_{2^n}/\mathbb{F}_2$ of dimension l and will define the factor base to be the set of points corresponding to $t(P) = x(P) + y(P) \in V$.

Theorem 1. *Let E_{d_1,d_2} be a binary Edwards curve over $\mathbb{F}_{2^n}$ and define the function $t(P) = x(P) + y(P)$. Let the m^{th} summation polynomials for binary Edwards curves be defined as follows:*

$$
\begin{aligned}
f_2(t_1, t_2) =& t_1 - t_2 \\
f_3(t_1, t_2, t_3) =& (d_2 t_1^2 t_2^2 + d_1(t_1^2 t_2 + t_1 t_2^2 + t_1 t_2 + d_1))t_3^2 + d_1(t_1^2 t_2^2 + t_1^2 t_2 + t_1 t_2^2 + t_1 t_2)t_3 \\
& + d_1^2(t_1^2 + t_2^2) \\
f_m(t_1, \ldots, t_m) =& \textit{Resultant}_t(f_{m-k}(t_1, t_2, \ldots, t_{m-k-1}, t), f_{k+2}(t_{m-k}, t_{m-k+1}, \ldots, t_m, t)), \\
& \textit{for } m \geq 4 \textit{ and } 1 \leq k \leq m - 3.
\end{aligned}
$$

For any points $P_1, \ldots, P_m \in E_{d_1,d_2}(\overline{\mathbb{F}}_2)$ such that $P_1 + \cdots + P_m = P_0$, then $f_m(t(P_1), \ldots, t(P_m)) = 0$. Conversely, given any $t_1, \ldots, t_m \in \overline{\mathbb{F}}_2$ such that $f_m(t_1, \ldots, t_m) = 0$, then there exist points $P_1, \ldots, P_m \in E_{d_1,d_2}(\overline{\mathbb{F}}_2)$ such that $t(P_i) = t_i$ for all $1 \leq i \leq m$ and $P_1 + \cdots + P_m = P_0$. For $m \geq 2$, the polynomials have degree 2^{m-2} in each variable.

Proof. Let $P_i = (x_i, y_i) \in E_{d_1,d_2}$ and $t_i = x_i + y_i$, where $1 \leq i \leq m$. For $m = 2$, we have $P_1 + P_2 = P_0$ that is $P_1 = -P_2 = (y_2, x_2)$ and this in turn implies $t_1 = t_2$. So, it is clear to see that $f_2(t_1, t_2) = t_1 - t_2 = 0$.

For $m = 3$, we have to construct the 3^{rd} summation polynomial $f_3(t_1, t_2, t_3)$ corresponding to $P_1 + P_2 + P_3 = P_0$. Let $(x_3, y_3) = (x_1, y_1) + (x_2, y_2)$ and $(x_4, y_4) = (x_1, y_1) - (x_2, y_2)$. Applying the group law, we have

$$
\begin{aligned}
x_3 =& \frac{d_1(x_1 + x_2) + d_2(x_1 + y_1)(x_2 + y_2) + (x_1 + x_1^2)(x_2(y_1 + y_2 + 1) + y_1 y_2)}{d_1 + (x_1 + x_1^2)(x_2 + y_2)} \\
y_3 =& \frac{d_1(y_1 + y_2) + d_2(x_1 + y_1)(x_2 + y_2) + (y_1 + y_1^2)(y_2(x_1 + x_2 + 1) + x_1 x_2)}{d_1 + (y_1 + y_1^2)(x_2 + y_2)}
\end{aligned}
$$

and

$$
\begin{aligned}
t_3 =& \frac{d_1(x_1 + x_2) + d_2(x_1 + y_1)(x_2 + y_2) + (x_1 + x_1^2)(x_2(y_1 + y_2 + 1) + y_1 y_2)}{d_1 + (x_1 + x_1^2)(x_2 + y_2)} \\
& + \frac{d_1(y_1 + y_2) + d_2(x_1 + y_1)(x_2 + y_2) + (y_1 + y_1^2)(y_2(x_1 + x_2 + 1) + x_1 x_2)}{d_1 + (y_1 + y_1^2)(x_2 + y_2)}.
\end{aligned}
$$

Then,

$$t_3 = \frac{\big(d_1 + (y_1 + y_1^2)(x_2 + y_2)\big)\,\big(d_1(x_1 + x_2) + d_2(x_1 + y_1)(x_2 + y_2) + (x_1 + x_1^2)(x_2(y_1 + y_2 + 1) + y_1y_2\big)}{\big(d_1 + (x_1 + x_1^2)(x_2 + y_2)\big)\,\big(d_1 + (y_1 + y_1^2)(x_2 + y_2)\big)}$$
$$+ \frac{\big(d_1 + (x_1 + x_1^2)(x_2 + y_2)\big)\,\big(d_1(y_1 + y_2) + d_2(x_1 + y_1)(x_2 + y_2) + (y_1 + y_1^2)(y_2(x_1 + x_2 + 1) + x_1x_2)\big)}{\big(d_1 + (x_1 + x_1^2)(x_2 + y_2)\big)\,\big(d_1 + (y_1 + y_1^2)(x_2 + y_2)\big)}.$$

Now (x_4, y_4) and t_4 are computed in a similar way and are given,

$$x_4 = \frac{d_1(x_1 + y_2) + d_2(x_1 + y_1)(x_2 + y_2) + (x_1 + x_1^2)(y_2(y_1 + x_2 + 1) + y_1x_2)}{d_1 + (x_1 + x_1^2)(x_2 + y_2)}$$
$$y_4 = \frac{d_1(y_1 + x_2) + d_2(x_1 + y_1)(x_2 + y_2) + (y_1 + y_1^2)(x_2(x_1 + y_2 + 1) + x_1y_2)}{d_1 + (y_1 + y_1^2)(x_2 + y_2)}$$

and $t_4 = x_4 + y_4$.

We now require to construct a quadratic polynomial in the indeterminate variable t whose roots are t_3 and t_4, that is $t^2 + (t_3 + t_4)t + t_3t_4$. We can use the `EliminationIdeal()` function of Magma [32] and the curve equation to express $t_3 + t_4$ and t_3t_4 in terms of the variables t_1 and t_2. So, we have finally

$$t_3 + t_4 = \frac{d_1t_1t_2(t_1t_2 + t_1 + t_2 + 1)}{d_1^2 + d_1\left(t_1 + t_1^2\right)t_2 + \left(d_1t_1 + d_2t_1^2\right)t_2^2} \quad \text{and} \quad t_3t_4 = \frac{d_1^2(t_1 + t_2)^2}{d_1^2 + d_1\left(t_1 + t_1^2\right)t_2 + \left(d_1t_1 + d_2t_1^2\right)t_2^2}.$$

Hence,

$$t^2 + (t_3 + t_4)t + t_3t_4 = \left(d_1^2 + d_1\left(t_1 + t_1^2\right)t_2 + \left(d_1t_1 + d_2t_1^2\right)t_2^2\right)t^2$$
$$+ \left(d_1t_1t_2(t_1t_2 + t_1 + t_2 + 1)\right)t + d_1^2(t_1 + t_2)^2.$$

Rearranging terms, we have

$$f_3(t_1, t_2, t_3) = (d_2t_1^2t_2^2 + d_1(t_1^2t_2 + t_1t_2^2 + t_1t_2 + d_1))t_3^2 + d_1(t_1^2t_2^2 + t_1^2t_2 + t_1t_2^2 + t_1t_2)t_3 + d_1^2(t_1 + t_2)^2.$$

For $m \geq 4$ we use the fact that $P_1 + \cdots + P_m = P_0$ if and only if there exists a point R on the curve such that $P_1 + \cdots + P_{m-k-1} + R = P_0$ and $-R + P_{m-k} + \cdots + P_m = P_0$. It follows that

$$f_m(t_1, \ldots, t_m) = \text{Resultant}_t(f_{m-k}(t_1, t_2, \ldots, t_{m-k-1}, t), f_{k+2}(t_{m-k}, t_{m-k+1}, \ldots, t_m, t)),$$
$$\text{(for all } m \geq 4 \text{ and } m - 3 \geq k \geq 1).$$

We can observe that the 3^{rd} summation polynomial has degree 2 in each variable t_i. The 4^{th} summation polynomial $f_4(t_1, t_2, t_3, t_4) = \text{Resultant}_t(f_3(t_1, t_2, t), f_3(t_3, t_4, t))$, which is the resultant of two third summation polynomials, has degree $2 \cdot 2 = 4$ in each variable t_i. Computing recursively using resultants, the m^{th} summation polynomial has degree 2^{m-2} in each variable. Irreducibility and symmetry follow by the same arguments as used by Semaev [27]. This completes the proof. □

Note that the degree bound 2^{m-2} is consistent with the arguments on page 44 (Sections 2 and 3.1) of [13]: Since $\deg(t) = 4$ we would expect polynomials of degree 4^{m-1}, but t is invariant and so factors through a 2-isogeny, so we get degree 2^{m-1}. The further saving of a factor 2 follows since $t(-P) = t(P)$.

We now specialise to the case $d_1 = d_2$, which will be the case considered in Section 3.3.

Lemma 2. *Let E_{d_1,d_2} be a binary Edwards curve over $\mathbb{F}_{2^n}$ such that $d_1 = d_2$ and define the function $t(P) = x(P) + y(P)$. Let the m^{th} summation polynomials for binary Edwards curves be defined as follows:*

$$\begin{aligned} f_2(t_1, t_2) =& t_1 + t_2 \\ f_3(t_1, t_2, t_3) =& (d_1 + t_1t_2(t_1 + 1)(t_2 + 1))t_3^2 + (t_1t_2 + (t_1 + 1)(t_2 + 1))t_3 + d_1(t_1 + t_2)^2 \\ f_m(t_1, \ldots, t_m) =& \textit{Resultant}_t(f_{m-j}(t_1, t_2, \ldots, t_{m-j-1}, t), f_{j+2}(t_{m-j}, t_{m-j+1}, \ldots, t_m, t)) \\ & (\textit{for all } m \geq 4 \textit{ and } 1 \leq j \leq m - 3). \end{aligned}$$

For any points $P_1, \ldots, P_m \in E_{d_1,d_1}(\overline{\mathbb{F}_2})$ such that $P_1 + \cdots + P_m = P_0$, then $f_m(t(P_1), \ldots, t(P_m)) = 0$. Conversely, given any $t_1, \ldots, t_m \in \overline{\mathbb{F}_2}$ such that $f_m(t_1, \ldots, t_m) = 0$, then there exist points $P_1, \ldots, P_m \in E_{d_1,d_1}(\overline{\mathbb{F}_2})$ such that $t(P_i) = t_i$ for all $1 \leq i \leq m$ and $P_1 + \cdots + P_m = P_0$. For $m \geq 2$, the polynomials have degree 2^{m-2} in each variable.

3.1 Action of Symmetric Group

Since the equation $P_1 + \cdots + P_m$ is symmetric it follows that the summation polynomials for binary Edwards curves are symmetric. Hence

$$f_{m+1}(t_1, t_2, \ldots, t_m, t(R)) \in \mathbb{F}_{2^n}[t_1, t_2, \ldots, t_m]^{S_m}$$

where S_m is the symmetric group and the right hand side denotes the ring of polynomials invariant under the group S_m. Hence, it is possible to express the summation polynomials in terms of the elementary symmetric polynomials $(e_1, e_2, \ldots, e_m)$ in the variables t_i, as they are generators of the ring $\mathbb{F}_{2^n}[t_1, \ldots, t_m]^{S_m}$.

Since the elementary symmetric polynomial e_i has degree i, it is natural to expect the polynomial to have lower degree after this change of variables. Another way to explain this degree reduction is to note that each relation $R = P_1 + \cdots + P_m$ comes in an orbit of size (at least, when the points P_i are all distinct) $m!$. This implies that the number of solutions to the polynomial when expressed in terms of the e_i is smaller than the original polynomial, and this is compatible with a lowering of the degree.

3.2 Action of Points of Order 2

It was proposed in [12, 13] to consider the action of small torsion points to further lower the degree of the summation polynomials. This idea also allows to effectively reduce the size of the factor base when performing the linear algebra. Hence, it is important to exploit torsion points as much as possible. Of the previous papers, [12] only considers odd characteristic, while [13] considers even characteristic (and even goes as far as summation polynomials of 8 points!) but only for curves in Weierstrass form and using a point of order 2. In this section we consider these ideas for binary Edwards curves, and in the next section extend to using a point of order 4.

Fix a vector space $V \subset \mathbb{F}_{2^n}$ of dimension l. The factor base will be

$$\mathcal{F} = \{P \in E_{d_1,d_2}(\mathbb{F}_{2^n}) : t(P) \in V\}.$$

We expect $\#\mathcal{F} \approx \#V$, and our experiments confirm this.

As mentioned in Section 3, if $P = (x, y) \in E_{d_1,d_2}$ then $P + T_2 = (x + 1, y + 1)$. Note that $t(P + T_2) = (x + 1) + (y + 1) = x + y = t(P)$ and so the function t is already invariant under addition by T_2. Since the factor base is defined in terms of $t(P)$ we have that $P \in \mathcal{F}$ implies $P + T_2 \in \mathcal{F}$. In other words, our choice of variables is already invariant under the action of adding a 2-torsion point.

Let $R = P_1 + \cdots + P_m$ and let $u = (u_1, \ldots, u_{m-1}) \in \{0,1\}^{m-1}$. Then

$$R = (P_1 + u_1 T_2) + (P_2 + u_2 T_2) + \cdots + (P_{m-1} + u_{m-1} T_2) + \left(P_m + \left(\sum_{i=1}^{m-1} u_i \right) T_2 \right).$$

This gives an action of the group $(\mathbb{Z}/2\mathbb{Z})^{m-1}$ on the set of relations $R = P_1 + \cdots + P_m$. Combining with the action of the symmetric group, we have that the Dihedral Coxeter group $D_m = (\mathbb{Z}/2\mathbb{Z})^{m-1} \rtimes S_m$ acts on the set of relations, and hence on the summation polynomial. Analogous to the discussion in the previous section, each relation $R = P_1 + \cdots + P_m$ generically comes in an orbit of size $2^{m-1} m!$.

Since the variables t_i are already invariant under addition by T_2, it follows that

$$f_{m+1}(t_1, t_2, \ldots, t_m, t(R)) \in \mathbb{F}_{2^n}[t_1, t_2, \ldots, t_m]^{D_m}.$$

Hence it can be written in terms of the elementary symmetric polynomials e_i, as they are the generators of the ring $\mathbb{F}_{2^n}[t_1, t_2, \ldots, t_m]^{D_m}$. This reduces its degree and we experience a speed-up in the FGLM algorithm due to the reduction in the size of the set of solutions.

To speed-up the linear algebra, the factor base can be reduced in size. Recall that each solution $(t_1, \ldots, t_m)$ corresponds to many relations. Let us fix, for each t, one of the four points $\{P, -P, P + T_2, -P + T_2\}$, and put only that point into our factor base. Hence the size of $\mathcal{F}$ is exactly the same as the number of $t \in V$ that correspond to elliptic curve points, which is roughly $\frac{1}{4}\#V$.

Then, for a point R, given a solution $f_{m+1}(t_1, \ldots, t_m, t(R)) = 0$ there is a unique value $z_0 \in \{0,1\}$, unique points $P_1, \ldots, P_m \in \mathcal{F}$, and unique choices of sign $z_1, \ldots, z_m \in \{-1, 1\}$ such that

$$R + z_0 T_2 = \sum_{i=1}^{m} z_i P_i.$$

It follows that the matrix size is reduced by a factor of $1/4$ (with one extra column added to store the coefficient of T_2) which means we need to find fewer relations and the complexity of the linear algebra, which has a complexity of $\tilde{O}(m\#\mathcal{F}^2)$ using the Lanczos or Wiedemann algorithm, is reduced by a factor of $(1/4)^2$.

3.3 Action of Points of Order 4

We now consider binary Edwards curves in the case $d_1 = d_2$. Then $T_4 = (1, 0) \in E_{d_1,d_1}$ and one can verify that $T_4 + T_4 = (1,1) = T_2$ and so T_4 has order four. The group generated by T_4 is therefore $\{P_0, T_4, T_2, -T_4 = (0,1)\}$.

For a point $P = (x, y) \in E_{d_1,d_1}$ we have $P + T_4 = (y, x+1)$. Hence $t(P + T_4) = t(P) + 1$. We construct our factor base $\mathcal{F}$ such that $(x, y) \in \mathcal{F}$ implies $(y, x+1) \in \mathcal{F}$. For example, we can choose a vector subspace $V \subseteq \mathbb{F}_{2^n}$ such that $v \in V$ if and only if $v + 1 \in V$, and set $\mathcal{F} = \{P \in E_{d_1,d_1}(\mathbb{F}_{2^n}) : t(P) \in V\}$.

If $R = P_1 + P_2 + \cdots + P_m$ is a relation and $(u_1, \ldots, u_{m-1}) \in \{0,1,2,3\}^{m-1}$ then we also have

$$R = (P_1 + [u_1]T_4) + (P_2 + [u_2]T_4) + \cdots + (P_{m-1} + [u_{m-1}]T_4) + (P_m + [u_m]T_4) \tag{1}$$

for $u_m = -\sum_{i=1}^{m-1} u_i$. Hence, one can consider the group $G_m = (\mathbb{Z}/4\mathbb{Z})^{m-1} \rtimes S_m$ acting on the summation polynomial. To express the summation polynomial in terms of invariant variables it suffices to note that the invariants under the action $t \mapsto t+1$ in characteristic 2 are $t(t+1) = t^2 + t$ (this is mentioned in Section 4.3 of [13]). Hence,

$$\begin{aligned} s_2 =& (t_1^2 + t_1)(t_2^2 + t_2) + \cdots + (t_{m-1}^2 + t_{m-1})(t_m^2 + t_m), \\ \vdots& \\ s_m =& (t_1^2 + t_1)(t_2^2 + t_2) \cdots (t_m^2 + t_m). \end{aligned} \tag{2}$$

are invariant variables. One might also expect to use

$$e_1 + e_1^2 = t_1 + t_1^2 + \cdots + t_m + t_m^2$$

but since the addition by T_4 cancels out in equation (1) we actually have that $e_1 = t_1 + \cdots + t_m$ remains invariant. Hence, we can use the invariant variables $e_1, s_2, \ldots, s_m$, which are the generators of the ring $\mathbb{F}_{2^n}[t_1, t_2, \ldots, t_m]^{G_m}$.

It is clear that we further halve the size of the factor base by choosing a unique representative of the orbit under the action. Overall, the factor base is reduced in total by a factor of $1/8$ over the basic method. Hence the complexity of the linear algebra is reduced by a factor of $(1/8)^2$.

4 The Index Calculus Algorithm

We now present the full index calculus algorithm combined with the new variables introduced in Section 3.1. We work in $E(\mathbb{F}_{2^n}) := E_{d_1,d_1}(\mathbb{F}_{2^n})$ where n is prime and E_{d_1,d_1} is a binary Edwards curve with parameters $d_2 = d_1$. We choose an integer m (for the number of points in a relation) and an integer l. Considering $\mathbb{F}_{2^n}$ as a vector space over $\mathbb{F}_2$ we let V be a vector subspace of dimension l. More precisely, we will suppose $\mathbb{F}_{2^n}$ is represented using a polynomial basis $\{1, \theta, \ldots, \theta^{n-1}\}$ where $F(\theta) = 0$ for some irreducible polynomial $F(x) \in \mathbb{F}_2[x]$ of degree n. We will take V to be the vector subspace of $\mathbb{F}_{2^n}$ over $\mathbb{F}_2$ with basis $\{1, \theta, \ldots, \theta^{l-1}\}$.

We start with the standard approach, leaving the symmetry-breaking to Section 4.2. We define a factor base $\mathcal{F} = \{P \in E(\mathbb{F}_{2^n}) : t(P) \in V\}$, where $t(x,y) = x + y$. Relations will be sums of the form $R = P_1 + P_2 + \cdots + P_m$ where $P_i \in \mathcal{F}$. We heuristically assume that $\#\mathcal{F} \approx 2^l$. Under this heuristic assumption we expect the number of points in $\{P_1 + \cdots + P_m : P_i \in \mathcal{F}\}$ to be roughly $2^{lm}/m!$. Hence, the probability that a uniformly chosen point $R \in E(\mathbb{F}_{2^n})$ can be decomposed in this way is heuristically $(2^{lm}/m!)/2^n = 1/(m!2^{n-lm})$. Hence we would like to choose m and l so that lm is not too much smaller than n.

To compute relations we evaluate the summation polynomial at the point R to get

$$f_{m+1}(t_1, t_2, \ldots, t_m, t(R)) \in \mathbb{F}_{2^n}[t_1, t_2, \ldots, t_m].$$

If we can find a solution $(t_1, t_2, \ldots, t_m) \in V^m$ satisfying $f_{m+1}(t_1, t_2, \ldots, t_m, t(R)) = 0$ then we need to determine the corresponding points, if they exist, $(x_i, y_i) \in E(\mathbb{F}_{2^n})$ such that $t_i = x_i + y_i$ and $(x_1, y_1) + \cdots + (x_m, y_m) = R$. Finding (x_i, y_i) given t_i is just taking roots of a univariate quartic polynomial. Once we have m points in $E(\mathbb{F}_{2^n})$, we may need to check up to 2^{m-1} choices of sign (and also determine an additive term $z_{j,0}T_4$, since our factor base only includes one of the eight points for each value of $t_i(t_i+1)$) to be able to record the relation as a vector. The cost of computing the points (x_i, y_i) is almost negligible, but checking the signs may incur some cost for large m.

When a relation exists (i.e., the random point R can be written as a sum of m points in the factor base) then there exists a solution $(t_1, \ldots, t_m) \in V^m$ to the polynomial system that can be lifted to points in

$E(\mathbb{F}_{2^n})$. When no relation exists there are two possible scenarios: Either there is no solution $(t_1, \ldots, t_m) \in V^m$ to the polynomial system, or there are solutions but they don't lift to points in $E(\mathbb{F}_{2^n})$. In both cases, the running time of detecting that a relation does not exist is dominated by the Gröbner basis computation and so is roughly the same.

In total we will need $\#\mathcal{F} + 1 \approx \#V = 2^l$ relations. Finally, these relations are represented as the system of equations

$$u_j P + w_j Q = z_{j,0} T_4 + \sum_{P_i \in \mathcal{F}} z_{j,i} P_i$$

where $M = (z_{j,i})$ is a sparse matrix with at most m non-zero entries per row. Let r be the order of P (assumed to be odd). If S is any vector in the kernel of the matrix (meaning $SM \equiv 0 \pmod{r}$), then writing $u = S(u_1, \ldots, u_{\ell+1})^T$ (where $\ell = \#\mathcal{F}$) and $w = S(w_1, \ldots, w_{\ell+1})^T$. We have $uP + wQ = 0$ (the T_4 term must disappear if r is odd) and so $u + wa \equiv 0 \pmod{r}$ and we can solve for the discrete logarithm a.

The details are given in Algorithm 1.

4.1 The Choice of Variables

Recall that our summation polynomials $f_{m+1}(t_1, t_2, \ldots, t_m, t(R))$ can be written in terms of the invariant variables $(e_1, s_2, \ldots, s_m)$. Here we are exploiting the full group $(\mathbb{Z}/4\mathbb{Z})^{m-1} \rtimes S_m$. Note that $t(R) \in \mathbb{F}_{2^n}$ is a known value and can be written as $t(R) = r_0 + r_1\theta + r_2\theta^2 + \cdots + r_{n-1}\theta^{n-1}$ with $r_i \in \mathbb{F}_2$.

As noted by Huang et al [19], and using their notation, let us write t_j, e_1, and s_j in terms of binary variables with respect to the basis for $\mathbb{F}_{2^n}$. We have

$$t_j = \sum_{i=0}^{l-1} c_{j,i} \theta^i \tag{3}$$

for $1 \leq j \leq m$, which is a total of lm binary variables $c_{j,i}$. Set $k = \min(\lfloor n/(2(l-1)) \rfloor, m)$. The invariant variables $e_1, s_2, \ldots, s_m$ can be written as,

$$\begin{aligned}
e_1 &= d_{1,0} + d_{1,1}\theta + d_{1,2}\theta^2 + \cdots + d_{1,l-1}\theta^{l-1} \\
s_2 &= d_{2,0} + d_{2,1}\theta + d_{2,2}\theta^2 + \cdots + d_{2,4(l-1)}\theta^{4(l-1)} \\
&\vdots \\
s_j &= d_{j,0} + d_{j,1}\theta + d_{j,2}\theta^2 + \cdots + d_{j,2j(l-1)}\theta^{2j(l-1)} \\
&\quad \text{where } 1 \leq j \leq k = \min(\lfloor n/(2(l-1)) \rfloor, m) \\
s_{j+1} &= d_{j+1,0} + d_{j+1,1}\theta + d_{j+1,2}\theta^2 + \cdots + d_{j+1,(n-1)}\theta^{n-1} \\
&\vdots \\
s_m &= d_{m,0} + d_{m,1}\theta + d_{m,2}\theta^2 + \cdots + d_{m,n-1}\theta^{n-1}.
\end{aligned}$$

Suppose that $n \approx lm$. Then $k = n/(2(l-1)) \approx m/2$ and so we suppose it takes the value $\acute{m} = \lceil \frac{m}{2} \rceil$. Then the number of binary variables $d_{i,j}$ is

$$N = l + (4(l-1) + 1) + (6(l-1) + 1) + \cdots + (2\acute{m}(l-1) + 1) + \acute{m}n \approx (m^2 l + mn)/2.$$

Writing the evaluated summation polynomial as $G(e_1, s_2, \ldots, s_m)$ we now substitute the above formulae to obtain a polynomial in the variables $d_{j,i}$. Apply Weil descent to the polynomial to get

$$\phi_1 + \phi_2\theta + \cdots + \phi_n\theta^{n-1} = 0.$$

where the ϕ_i are polynomials over $\mathbb{F}_2$ in the $d_{j,i}$. This forms a system of n equations in the N binary variables $d_{j,i}$. We add the field equations $d_{j,i}^2 - d_{j,i}$ and then denote this system of equations by sys_1.

One could attempt to solve this system using Gröbner basis methods. For each candidate solution $(d_{j,i})$ one would compute the corresponding solution $(e_1, s_2, \ldots, s_m)$ and then solve a univariate polynomial equation (i.e., take roots) to determine the corresponding solution $(t_1, \ldots, t_m)$. From this one determines whether each value t_j corresponds to an elliptic curve point $(x_j, y_j) \in E(\mathbb{F}_{2^n})$ such that $x_j + y_j = t_j$. If everything works ok then one forms the relation.

However, the approach just mentioned is not practical as the number N of binary variables is too large compared with the number of equations. Hence, we include the $lm < n$ variables $c_{j,\tilde{i}}$ (for $1 \le j \le m$, $0 \le \tilde{i} \le l - 1$) to the problem, and add a large number of new equations relating the $c_{j,\tilde{i}}$ to the $d_{j,i}$ via the t_j and equations (2) and (3).

This gives N additional equations in the $N + lm$ binary variables. After adding the field equations $c_{j,\tilde{i}}^2 - c_{j,\tilde{i}}$ we denote this system of equations by sys_2. Finally we solve $sys_1 \cup sys_2$ using Gröbner basis algorithms F4 or F5 using the degree reverse lexicographic ordering. From a solution, the corresponding points P_j are easily computed.

Algorithm 1 Index Calculus Algorithm on Binary Edwards Curve

```
Set $N_r \leftarrow 0$
while $N_r \le \#\mathcal{F}$ do
    Compute $R \leftarrow uP + wQ$ for random integer values $u$ and $w$
    Compute summation polynomial $G(e_1, s_2, \ldots, s_m) := f_{m+1}(e_1, s_2, \ldots, s_m, t(R))$ in the variables $(e_1, s_2, \ldots, s_m)$
    Use Weil descent to write $G(e_1, s_2, \ldots, s_m)$ as $n$ polynomials in binary variables $d_{j,i}$
    Add field equations $d_{j,i}^2 - d_{j,i}$ to get system of equations $sys_1$
    Buld new polynomial equations relating the variables $d_{j,i}$ and $c_{j,\tilde{i}}$
    Add field equations $c_{j,\tilde{i}}^2 - c_{j,\tilde{i}}$ to get system of equations $sys_2$
    Solve system of equations $sys_1 \cup sys_2$ to get $(c_{j,\tilde{i}}, d_{j,i})$
    Compute corresponding solution(s) $(t_1, \ldots, t_m)$
    For each $t_j$ compute, if it exists, a corresponding point $P_j = (x_j, y_j) \in \mathcal{F}$
    if $z_1P_1 + z_2P_2 + \cdots + z_mP_m + z_0T_4 = R$ for suitable $z_0 \in \{0, 1, 2, 3\}, z_i \in \{1, -1\}$ then
        $N_r \leftarrow N_r + 1$
        Record $z_i, u, w$ in a matrix $M$ for the linear algebra
Use linear algebra to find non-trivial kernel element and hence solve ECDLP
```

4.2 Breaking Symmetry

We now explain how to break symmetry in the factor base while using the new variables as above.

Again, suppose $\mathbb{F}_{2^n}$ is represented using a polynomial basis and take V to be the subspace with basis $\{1, \theta, \ldots, \theta^{l-1}\}$. We choose m elements $v_i \in \mathbb{F}_{2^n}$ (which can be interpreted as vectors in the n-dimensional $\mathbb{F}_2$-vector space corresponding to $\mathbb{F}_{2^n}$) as follows: $v_1 = 0$, $v_2 = \theta^l = (0, 0, \ldots, 0, 1, 0, \ldots, 0)$ where the 1 is in position l. Similarly, $v_3 = \theta^{l+1}$, $v_4 = \theta^{l+1} + \theta^l$, $v_5 = \theta^{l+2}$ etc. In other words, v_i is represented as a vector of the form $(0,\dots,0,w_0,w_1,w_2,\dots)$ where $\cdots w_2w_1w_0$ is the binary expansion of $i-1$. Note that the subsets $V+v_i$ in $\mathbb{F}_{2^n}$ are pair-wise disjoint.

Accordingly, we define the factor bases to be $\mathcal{F}_i = \{P \in E(\mathbb{F}_{2^n}) : t(P) \in V + v_i\}$ for $1 \le i \le m$, where $t(x,y) = x + y$. The decomposition over the factor base of a point R will be a sum of the form $R = P_1 + P_2 + \cdots + P_m$ where $P_i \in \mathcal{F}_i$ for $1 \le i \le m$. Since we heuristically assume that $\#\mathcal{F}_i \approx 2^l$, we expect the number of points in $\{P_1 + \cdots + P_m : P_i \in \mathcal{F}_i\}$ to be roughly 2^{lm}. Note that there is no $1/m!$ term here. The entire purpose of this definition is to break the symmetry and hence increase the probability of relations. Hence, the probability that a uniformly chosen point $R \in E(\mathbb{F}_{2^n})$ can be decomposed in this way is heuristically $2^{lm}/2^n = 1/2^{n-lm}$.

There is almost a paradox here: Of course if $R = P_1 + \cdots + P_m$ then the points on the right hand side can be permuted and the point T_2 can be added an even number of times, and hence the summation polynomial evaluated at $t(R)$ is invariant under D_m. On the other hand, if the points P_i are chosen from distinct factor bases $\mathcal{F}_i$ then one does not have the action by S_m, so why can one still work with the invariant variables $(e_1, s_2, \dots, s_m)$?

To resolve this "paradox" we must distinguish the computation of the polynomial from the construction of the system of equations via Weil descent. The summation polynomial does have an action by D_m (and G_m), and so that action should be exploited. When we do the Weil descent and include the definitions of the factor bases $\mathcal{F}_i$, we then introduce some new variables. As noted by Huang et al [19], expressing the invariant variables with respect to the variables from the construction of the factor bases is non-trivial. But it is this stage where we introduce symmetry-breaking.

When re-writing the system in terms of new variables, there is a penalty from the additional factors $+v_i$. For example, previously we had $t_2 = c_{2,0} + c_{2,1}\theta + c_{2,2}\theta^2 + \cdots + c_{2,l-1}\theta^{l-1}$ but now we have (for the case $m = 4$)

$$\begin{aligned}
t_1 &= c_{1,0} + c_{1,1}\theta + c_{1,2}\theta^2 + \cdots + c_{1,l-1}\theta^{l-1}\\
t_2 &= c_{2,0} + c_{2,1}\theta + c_{2,2}\theta^2 + \cdots + c_{2,l-1}\theta^{l-1} + \theta^l\\
t_3 &= c_{3,0} + c_{3,1}\theta + c_{3,2}\theta^2 + \cdots + c_{3,l-1}\theta^{l-1} + \theta^{l+1}\\
t_4 &= c_{4,0} + c_{4,1}\theta + c_{4,2}\theta^2 + \cdots + c_{4,l-1}\theta^{l-1} + \theta^l + \theta^{l+1}.
\end{aligned}$$

It follows that

$$e_1 = t_1 + t_2 + t_3 + t_4 = d_{1,0} + d_{1,1}\theta + \cdots + d_{1,l-1}\theta^{l-1}$$

can be represented exactly as before. But the other polynomials are less simple. For example,

$$s_2 = (t_1^2 + t_1)(t_2^2 + t_2) + \cdots + (t_3^2 + t_3)(t_4^2 + t_4)$$

previously had highest term $d_{2,4l-4}\theta^{4l-4}$ but now has highest terms $d_{2,4l-4}\theta^{4l-4} + d_{2,4l-2}\theta^{4l-2} + \theta^{4l+2}$. Hence, we require one more variable than the previous case, and things get worse for higher degree terms. So the symmetry breaking increases the probability of a relation but produces a harder system of polynomial equations to solve.

An additional consequence of this idea is that the factor base is now roughly m times larger than in the symmetric case. So the number of relations required is increased by a factor m, and so the speedup over previous methods is actually by a factor approximately $m!/m = (m-1)!$. Also, the cost of the linear algebra is increased by a factor m^2 (though the system of linear equations is structured in blocks and so some optimisations may be possible). When using a point of order 4 with binary Edwards curves, the linear algebra cost is reduced (in comparison with the naive method) by a factor $(m/8)^2$.

For large q and small n, it seems that symmetry-breaking is not a useful idea, as the increase in number of variables becomes a huge problem that is not compensated by the $(m-1)!$ factor. However, for small q and large n the situation is less clear. To determine whether the idea is a good one, it is necessary to perform some experiments (see Section 6).

5 SAT Solvers

Shantz and Teske [26] discuss a standard idea [30, 31, 2] they call the "hybrid method", which is to partially evaluate the system at some random points before applying Gröbner basis algorithms. They argue (Section 5.2) that it is better to just use the "delta method" ($n - ml > 0$), where m is the number points in a relation and 2^l is the size of the factor base. The main observation of Shantz and Teske [26] is that using smaller l speeds-up the Gröbner basis computation at the cost of decreasing the probability of getting a relation. So, they try to find such an optimal l value.

Our choice of coordinates for binary Edwards curves helps us lower the degree of our systems. As a result we were able to make successful experiments for $m = 4$ and $l \in \{3, 4\}$ using Gröbner basis algorithms, as reported in Table 3. For $l > 4$, values such that $n - ml > 0$ suffered high running times as the result of increased number of variables coming from our invariant variables.

To increase the range for these methods, we investigated other approaches to solving systems of multivariate polynomial equations over a binary field. In particular, we experimented with SAT solvers. We used Minisat 2.1 [29], a version of Minisat [10, 28, 29], coupled with the Magma system for converting the polynomial system into conjunctive normal form (CNF).

On the positive side, our experiments show that SAT solvers can be faster and, more importantly, handle larger range of values for l. As is shown in Table 1, we can work with l up to 7 for some n, whereas Gröbner basis methods are limited to $l \in \{3, 4\}$ in our experiments.

However, on the negative side, the running time of SAT solvers varies a lot depending on many factors. They are randomised algorithms, but more significantly they seem to be faster when there is a solution of low hamming weight. They are even better when there is a solution of low Hamming weight and it is the lower bits that are non-zero. The value of the curve parameter d_1 also seems to effect the running time. Finally, SAT solvers are usually slow when no solution exists. This behaviour is very different to the case of Gröbner basis methods, which perform rather reliably and are slightly better when the system of equations has no solution.

Hence, we suggest using SAT solvers with an "early abort" strategy: One can generate a lot of instances and run SAT solvers in parallel and then kill all instances that are still running after some time threshold has been passed (a similar idea is mentioned in Section 7.1 of [24]). This could allow the index calculus algorithm to be run for a larger set of parameters. The probability of finding a relation is now decreased. The probability that a relation exists must be multiplied by the probability that the SAT solver terminates in less than the time threshold (we took an upper bound of 200 seconds for the execution time), in the case when a solution exists. It is this latter probability that we estimate in the P_{succ} column of Table 1.

Note that all modern fast SAT solvers periodically restart the search for satisfiability or unsatisfiability with "restarting strategies" [17]: a cutoff value in the number of backtracks. Minisat has a small first restart (100), second restart (250), and the size of consecutive restarts grows geometrically. So an "early-abort" mechanism is related to rejecting an instance when the number of backtracks becomes too large.

SAT solvers take an input in Conjunctive Normal Form (CNF): a conjunction of clauses where a clause is a disjunction of literals, and a literal is a variable or its negation. The Magma interface with Minisat performs the conversion from polynomial equations to CNF. The number of variables, the number of clauses, and the total length of all the clauses (that is, the total number of literals) determines the size of the CNF expression. We list these numbers in Table 1. Although the running time of SAT solvers in the worst case is exponential in the number of variables in the problem, practical running times may be shorter. It is beyond the scope of this paper to discuss the relations between problem size and hardness for SAT solvers.

6 Experimental Results

We conducted several experiments using binary Edwards elliptic curves E over $\mathbb{F}_{2^n}$. We always use the $m+1$-summation polynomial to find relations as a sum of m points in the factor base. The factor base is defined using a vector space of dimension l. In our experiments we follow the approach of Huang et al [19] and examine the effect of different choices of variables on the computation of intermediate results and degree of regularity D_{reg} (as it is the main complexity indicator of F4 or F5 Gröbner basis algorithms: the time and memory complexities are roughly estimated to be $N^{3D_{\text{reg}}}$ and $N^{2D_{\text{reg}}}$ respectively where N is the number of variables). Our hope is to get better experimental results resulting from exploiting the symmetries of binary Edwards curves.

Experiment 1: For the summation polynomials we use the variables $e_1, e_2, \dots, e_m$, which are invariants under the group $D_m = (\mathbb{Z}/2\mathbb{Z})^{m-1} \rtimes S_m$. The factor base is defined with respect to a fixed vector space of dimension l.

Experiment 2: For the summation polynomials we use the variables $e_1, s_2, \dots, s_m$ from equation (2), which are invariants under the group $G_m = (\mathbb{Z}/4\mathbb{Z})^{m-1} \rtimes S_m$. The factor base is defined with respect to a fixed vector space V of dimension l such that $v \in V$ if and only if $v + 1 \in V$.

Experiment 3: For the summation polynomials we use the variables $e_1, s_2, \dots, s_m$, which are invariants under the group $(\mathbb{Z}/4\mathbb{Z})^{m-1} \times S_m$. We use symmetry-breaking to define the factor base by taking affine spaces (translations of a vector space of dimension l).

We denoted the set-up operations (lines 4 to 8 of Algorithm 1) by T_{Inter}, while T_{SAT} and T_{GB} denote the time for line 9. Other notation includes Mem (the average memory used in megabytes by the Minisat SAT solver or Gröbner basis), D_{reg} (the degree of regularity), Var (the total number of variables in the system) and P_{equ} (the total number of equations). In Table 1 we also give a success probability P_{succ} the percentage of times our SAT program terminated with solution within 200 seconds, T_{SAT} the average of the running times in seconds to compute step 9 using a SAT solver, and $\#\text{Clauses}$ and $\#\text{Literals}$ are the average number of clauses and total number of literals (i.e., total length) of the CNF input to the SAT solver. All experiments are carried out using a computational server (3.0GHz CPU x8, 28G RAM). In all our experiments, timings are averages of 100 trials except for values of $T_{\text{GB}} + T_{\text{Inter}} > 200$ seconds (our patience threshold), in this case they are single instances. We stress that all the tables do not report experiments for the case when the system of equations has no solution. As indicated in [19], the computational complexity is lower when the system of equations has no solution.

Table 1 compares Minisat with Gröbner basis methods (experiment 2) for $m = 4$. The main observation of this experiment is we can handle larger values of l with Minisat in reasonable amount of time than Gröbner basis methods. But the process has to be repeated 1/P_{succ} times on average, as the probability of finding a relation is decreased by P_{succ}. We also observe that the memory used by Minisat is much lower than that of the Gröbner basis algorithm. We do not report experiments using Gröbner basis method for values of $l > 4$ as they are too slow and have huge memory requirements.

Table 1. Comparison of solving polynomial systems, when there exists a solution to the system, in experiment 2 using SAT solver (Minisat) versus Gröbner basis methods for $m = 4$. $\#\mathrm{Var}$ and $\#P_{\mathrm{equ}}$ are the number of variables and the number of polynomial equations respectively. Mem is average memory used in megabytes by the SAT solver or Gröbner basis algorithm. $\#\mathrm{Clauses}$, $\#\mathrm{Literals}$, and P_{succ} represent the average number of clauses, total number of literals, and the percentage of times Minisat halts with solutions within 200 seconds respectively.

Experiment 2 with SAT solver Minisat									
n	l	$\#\mathrm{Var}$	$\#P_{\mathrm{equ}}$	$\#\mathrm{Clauses}$	$\#\mathrm{Literals}$	T_{Inter}	T_{SAT}	Mem	P_{succ}
17	3	54	59	46678	181077	0.35	7.90	5.98	94%
	4	67	68	125793	485214	0.91	27.78	9.38	90%
19	3	54	61	55262	215371	0.37	3.95	6.07	93%
	4	71	74	140894	543422	1.29	18.38	18.05	86%
23	3	54	65	61572	240611	0.39	1.53	7.60	87%
	4	75	82	194929	760555	2.15	5.59	14.48	83%
	5	88	91	394759	1538560	4.57	55.69	20.28	64%
29	4	77	90	221828	868619	3.01	7.23	19.05	87%
	5	96	105	572371	2242363	9.95	39.41	32.87	67%
	6	109	114	855653	3345987	21.23	15.87	43.07	23%
	7	118	119	1063496	4148642	36.97	26.34	133.13	14%
31	4	77	92	284748	1120243	3.14	17.12	20.52	62%
	5	98	109	597946	2345641	11.80	33.48	45.71	57%
	6	113	120	892727	3489075	26.23	16.45	118.95	12%
	7	122	125	1307319	5117181	44.77	21.98	148.95	8%
37	4	77	98	329906	1300801	3.41	26.12	29.97	59%
	5	100	117	755621	2977220	13.58	48.19	50.97	40%
	6	119	132	1269801	4986682	41.81	42.85	108.41	11%
	7	134	143	1871867	7350251	94.28	40.15	169.54	6%
41	4	77	102	317272	1250206	3.08	19.28	27.59	68%
	5	100	121	797898	3146261	15.71	27.14	49.34	65%
	6	123	140	1353046	5326370	65.25	31.69	89.71	13%
43	4	77	104	374011	1477192	2.97	17.77	28.52	68%
	5	100	123	825834	3258080	13.85	29.60	54.83	52%
47	4	77	108	350077	1381458	3.18	11.40	29.93	59%
	5	100	127	836711	3301478	14.25	27.56	61.55	43%
53	4	77	114	439265	1738168	11.02	27.88	32.35	75%
	5	100	133	948366	3748119	14.68	34.22	64.09	62%
	6	123	152	1821557	7200341	49.59	41.55	123.38	11%
	7	146	171	2930296	11570343	192.20	67.27	181.20	4%

Experiment 2 with Gröbner basis: F_4						
n	l	$\#\mathrm{Var}$	$\#P_{\mathrm{equ}}$	T_{Inter}	T_{GB}	Mem
17	3	54	59	0.29	0.29	67.24
	4	67	68	0.92	51.79	335.94
19	3	54	61	0.33	0.39	67.24
	4	71	74	1.53	33.96	400.17
23	3	54	65	0.26	0.31	67.24
	4	75	82	2.52	27.97	403.11
29	3	54	71	0.44	0.50	67.24
	4	77	90	3.19	35.04	503.87
31	3	54	73	0.44	0.58	67.24
	4	77	92	3.24	9.03	302.35
37	3	54	79	0.36	0.43	67.24
	4	77	98	3.34	9.07	335.94
41	3	54	83	0.40	0.54	67.24
	4	77	102	3.39	17.19	382.33
43	3	54	85	0.43	0.53	67.24
	4	77	104	3.44	9.09	383.65
47	3	54	89	0.50	0.65	67.24
	4	77	108	3.47	9.59	431.35
53	3	54	95	0.33	0.40	67.24
	4	77	114	11.43	11.64	453.77

Table 2 compares experiment 1 and experiment 2 in the case $m = 3$. Gröbner basis methods are used in both cases. Timings are averages from 100 trials except for values of $T_{GB} + T_{Inter} > 200$ seconds, in this case they are single instances.

Experiments in [19] are limited to the case $m = 3$ and $l \in \{3, 4, 5, 6\}$ for prime degree extensions

$$n \in \{17, 19, 23, 29, 31, 37, 41, 43, 47, 53\}.$$

This is due to high running times and large memory requirements, even for small parameter sizes. As shown in Table 2, we repeated these experiments. Exploiting greater symmetry (in this case experiment 2) is seen to reduce the computational costs. Indeed, we can go up to $l = 8$ with reasonable running time for some n, which is further than [19]. The degree of regularity stays ≤ 4 in both cases.

Table 2. Comparison of solving our systems of equations, having a solution, using Gröbner basis methods in experiment 1 and experiment 2 for $m = 3$. Notation is as above. '*' indicates that the time to complete the experiment exceeded our patience threshold.

Experiment 1						
n	l	D_{reg}	#Var	$\#P_{\text{equ}}$	T_{Inter}	T_{GB}
17	5	4	42	44	0.08	13.86
19	5	4	42	46	0.08	18.18
	6	4	51	52	0.18	788.91
23	5	4	42	50	0.10	35.35
	6	4	51	56	0.21	461.11
	7	*	*	*	*	*
29	5	4	42	56	0.11	31.64
	6	4	51	62	0.25	229.51
	7	4	60	68	0.60	5196.18
	8	*	*	*	*	*
31	5	4	42	58	0.12	5.10
	6	5	51	64	0.27	167.29
	7	5	60	70	0.48	3259.80
	8	*	*	*	*	*
37	5	4	42	64	0.18	0.36
	6	4	51	70	0.34	155.84
	7	4	60	76	0.75	1164.25
	8	*	*	*	*	*
41	5	4	42	68	0.16	0.24
	6	4	51	74	0.36	251.37
	7	4	60	80	0.77	1401.18
	8	*	*	*	*	*
43	5	4	42	70	0.19	0.13
	6	4	51	76	0.38	176.67
	7	3	60	82	0.78	1311.23
	8	*	*	*	*	*
47	5	4	42	74	0.19	0.14
	6	4	51	80	0.54	78.43
	7	*	*	*	*	*
	8	*	*	*	*	*
53	5	4	51	80	0.22	0.19
	6	5	51	86	0.45	1.11
	7	4	60	92	1.20	880.59
	8	*	*	*	*	*

Experiment 2						
n	l	D_{reg}	#Var	$\#\text{P}_{\text{eq}}$	T_{Inter}	T_{GB}
17	5	4	54	56	0.02	0.41
19	5	3	56	60	0.02	0.48
	6	4	62	63	0.03	5.58
23	5	4	60	68	0.02	0.58
	6	4	68	73	0.04	2.25
	7	*	*	*	*	*
29	5	4	62	76	0.03	0.12
	6	4	74	85	0.04	2.46
	7	4	82	90	0.07	3511.14
	8	*	*	*	*	*
31	5	4	62	78	0.03	0.36
	6	4	76	89	0.05	2.94
	7	4	84	94	0.07	2976.97
	8	*	*	*	*	*
37	5	4	62	84	0.04	0.04
	6	4	76	95	0.06	4.23
	7	4	90	106	0.09	27.87
	8	*	*	*	*	*
41	5	4	62	88	0.03	0.04
	6	4	76	99	0.06	0.49
	7	4	90	110	0.09	11.45
	8	*	*	*	*	*
43	5	3	62	90	0.04	0.05
	6	4	76	101	0.06	5.35
	7	4	90	112	0.10	15.360
	8	*	*	*	*	*
47	5	4	62	94	0.04	0.06
	6	4	76	105	0.06	1.28
	7	4	90	116	0.13	8.04
	8	4	104	127	0.16	152.90
53	5	3	62	100	0.04	0.02
	6	4	76	111	0.06	0.19
	7	4	90	122	0.14	68.23
	8	4	104	133	0.19	51.62

Table 3 considers $m = 4$, which was not done in [19]. For the sake of comparison, we gather some data for experiment 1 and experiment 2. Again, exploiting greater symmetry (experiment 2) gives a significant decrease in the running times, and the degree of regularity D_{reg} is slightly decreased. The expected degree of regularity for $m = 4$, stated in [25], is $m^2 + 1 = 17$. The table shows that our choice of coordinates makes the case $m = 4$ much more feasible.

Table 3. Comparison of solving our systems of equations, having a solution, using Gröbner basis methods in experiment 1 and experiment 2 for $m = 4$. Notation is as above. The second tabular column already appeared in Table 1.

Experiment 1						
n	l	D_{reg}	$\#\text{Var}$	$\#P_{\text{equ}}$	T_{Inter}	T_{GB}
17	3	5	36	41	590.11	216.07
	4	*	*	*	*	*
19	3	5	36	43	564.92	211.58
	4	*	*	*	*	*
23	3	5	36	47	1080.14	146.65
	4	*	*	*	*	*
29	3	5	36	53	1069.49	232.49
	4	*	*	*	*	*
31	3	5	36	55	837.77	118.11
	4	*	*	*	*	*
37	3	5	36	61	929.82	178.04
	4	*	*	*	*	*
41	3	4	36	65	1261.72	217.22
	4	*	*	*	*	*
43	3	4	36	67	1193.13	220.25
	4	*	*	*	*	*
47	3	4	36	71	1163.94	247.78
	4	*	*	*	*	*
53	3	4	36	77	1031.93	232.110
	4	*	*	*	*	*

Experiment 2						
n	l	D_{reg}	$\#\text{Var}$	$\#P_{\text{equ}}$	T_{Inter}	T_{GB}
17	3	4	54	59	0.29	0.29
	4	4	67	68	0.92	51.79
19	3	4	54	61	0.33	0.39
	4	4	71	74	1.53	33.96
23	3	4	54	65	0.26	0.31
	4	4	75	82	2.52	27.97
29	3	4	54	71	0.44	0.50
	4	4	77	90	3.19	35.04
31	3	4	54	73	0.44	0.58
	4	4	77	92	3.24	9.03
37	3	4	54	79	0.36	0.43
	4	4	77	98	3.34	9.07
41	3	4	54	83	0.40	0.54
	4	4	77	102	3.39	17.19
43	3	4	54	85	0.43	0.53
	4	4	77	104	3.44	9.09
47	3	4	54	89	0.50	0.65
	4	4	77	108	3.47	9.59
53	3	4	54	95	0.33	0.40
	4	4	77	114	11.43	11.64

Our idea of symmetry breaking (experiment 3) is investigated in Table 4 for the case $m = 3$. Some of the numbers in the second tabular column already appeared in Table 2. Recall that the relation probability is increased by a factor $3! = 6$ in this case, so one should multiply the timings in the right hand column by $(m - 1)! = 2$ to compare overall algorithm speeds. The experiments are not fully conclusive (and there are a few "outlier" values that should be ignored), but they suggest that symmetry-breaking can give a speedup in many cases when n is large.

For larger values of n, the degree of regularity D_{reg} is often 3 when using symmetry-breaking while it is 4 for most values in experiment 2. The reason for this is unclear, but we believe that the performance we observe is partially explained by the fact that the degree of regularity stayed at 3 as n grows.

7 Conclusions

We have suggested that binary Edwards curves are most suitable for obtaining coordinates invariant under the action of a relatively large group. Faugère et al [12] studied Edwards curves in the non-binary case and showed how the symmetries can be used to speed-up point decomposition. We show that these ideas are equally applicable in the binary case. For large q and small n one would get the same result as in [12]: that the FGLM complexity is reduced by a factor of 2^{m-1}. We have studied small q and large (prime) n and shown that one can get overdetermined systems and that the use of symmetries reduces the degree of regularity.

The idea of a factor base that breaks symmetry allows to maximize the probability of finding a relation. For large enough n (keeping m and l fixed) this choice can give a small speed-up compared with previous methods.

Table 4. Comparison of solving our systems of equations using Gröbner basis methods having a solution in experiment 3 and experiment 2 when $m = 3$. Notation is as in Table 1. For a fair comparison, the timings in the right hand column should be doubled.

Experiment 3						
n	l	D_{reg}	#Var	$\#P_{\mathrm{equ}}$	T_{Inter}	T_{GB}
37	5	3	68	90	0.04	0.25
	6	4	80	99	0.07	5.67
	7	*	*	*	*	*
41	5	4	68	94	0.05	0.39
	6	3	80	103	0.07	4.55
	7	4	93	113	0.11	1905.21
43	5	4	68	96	0.05	0.21
	6	4	80	105	0.08	4.83
	7	3	94	116	0.12	100.75
47	5	4	68	100	0.05	0.17
	6	3	80	109	0.08	3.88
	7	3	94	120	0.11	57.61
53	5	3	68	106	0.06	0.08
	6	4	80	115	0.09	12.75
	7	3	94	126	0.14	11.38
59	5	4	68	112	0.06	0.05
	6	4	80	121	0.10	0.59
	7	4	94	132	0.16	13.60
61	5	4	68	114	0.06	0.04
	6	4	80	123	0.11	0.46
	7	4	94	134	0.16	8.61
67	5	3	68	120	0.07	0.02
	6	3	80	129	0.11	0.17
	7	4	94	140	0.16	121.33
71	5	3	68	124	0.07	0.02
	6	3	80	133	0.12	0.12
	7	4	94	144	0.18	2.06
73	5	3	68	126	0.08	0.02
	6	3	80	135	0.12	0.11
	7	4	94	146	0.18	1.47
79	5	3	68	132	0.08	0.02
	6	4	80	141	0.12	0.07
	7	4	94	152	0.19	0.62
83	5	3	68	136	0.08	0.02
	6	4	80	145	0.13	0.04
	7	3	94	156	0.21	0.29
89	5	3	68	142	0.09	0.02
	6	3	80	151	0.14	0.03
	7	3	94	162	0.21	0.17
97	5	3	68	150	0.09	0.02
	6	3	80	159	0.14	0.03
	7	4	94	170	0.22	0.10

Experiment 2						
n	l	D_{reg}	#Var	$\#P_{\mathrm{equ}}$	T_{Inter}	T_{GB}
37	5	4	62	84	0.04	0.04
	6	4	76	95	0.06	4.23
	7	4	90	106	0.09	27.87
41	5	4	62	88	0.03	0.04
	6	4	76	99	0.06	0.49
	7	4	90	110	0.09	11.45
43	5	3	62	90	0.04	0.05
	6	4	76	101	0.06	5.35
	7	4	90	112	0.10	15.360
47	5	4	62	94	0.04	0.06
	6	4	76	105	0.06	1.28
	7	4	90	116	0.13	8.04
53	5	3	62	100	0.04	0.02
	6	4	76	111	0.06	0.19
	7	4	90	122	0.14	68.23
59	5	4	62	106	0.04	0.02
	6	3	76	117	0.07	0.11
	7	4	90	128	0.11	4.34
61	5	4	62	108	0.04	0.02
	6	3	76	119	0.07	0.09
	7	4	90	130	0.11	5.58
67	5	4	62	114	0.04	0.02
	6	4	76	125	0.07	0.07
	7	4	90	136	0.11	0.94
71	5	4	62	118	0.04	0.02
	6	4	76	129	0.07	0.04
	7	3	90	140	0.12	0.25
73	5	4	62	120	0.05	0.02
	6	4	76	131	0.07	0.03
	7	3	90	142	0.13	0.22
79	5	4	62	126	0.05	0.02
	6	4	76	137	0.08	0.03
	7	4	90	148	0.12	0.33
83	5	4	62	130	0.05	0.02
	6	4	76	141	0.09	0.03
	7	4	90	152	0.13	0.13
89	5	4	62	136	0.05	0.02
	6	4	76	147	0.09	0.03
	7	4	90	158	0.13	0.05
97	5	4	62	144	0.05	0.02
	6	4	76	155	0.09	0.03
	7	4	90	166	0.13	0.04

SAT solvers often work better than Gröbner methods, especially in the case when the system of equations has a solution with low hamming weight supported mainly on the lower bits. They are non-deterministic and the running time varies widely depending on the inputs, including the curve parameter. Unfortunately, most of the time SAT solvers are slow (for example, because the system of equations does not have any solutions). We suggest an early abort strategy that may still make SAT solvers a useful approach.

We conclude by analysing whether these algorithms are likely to be effective for ECDLP instances in $E(\mathbb{F}_{2^n})$ when $n > 100$. The best we can seem to hope for in practice is $m = 4$ and $l \leq 10$. Note that the linear algebra cost is negligible for such parameters. Since the probability of a relation is roughly $2^{lm}/2^n$, so the number of trials (i.e., executions of polynomial system solving) needed to find a relation is at least $2^n/2^{ml} \geq 2^{n-40} \geq \sqrt{2^n}$. Since solving a system of equations is much slower than a group operation, we conclude that our methods are worse than Pollard rho. This is true even in the case of static-Diffie-Hellman, when only one relation is required to be found. Hence, we conclude that elliptic curves in characteristic 2 are safe against these sorts of attacks for the moment, though one of course has to be careful of other "Weil descent" attacks in this case, such as the Gaudry-Hess-Smart approach [15].

Acknowledgements

We thank Claus Diem, Christophe Petit and the anonymous referees for their helpful comments.

References

1. Daniel J. Bernstein, Tanja Lange and Reza Rezaeian Farashahi, Binary Edwards Curves, in E. Oswald and P. Rohatgi (eds.), CHES 2008, Springer LNCS 5154 (2008) 244–265.
2. Luk Bettale, Jean-Charles Faugère and Ludovic Perret, Hybrid approach for solving multivariate systems over finite fields, J. Math. Crypt. **3** (2009) 177–197.
3. Daniel R. L. Brown and Robert P. Gallant, The Static Diffie-Hellman Problem, IACR Cryptology ePrint Archive 2004/306 (2004)
4. Nicolas T. Courtois and Gregory V. Bard, Algebraic Cryptanalysis of the Data Encryption Standard, in S. D. Galbraith (ed.), IMA Int. Conf. Cryptography and Coding, Springer LNCS 4887 (2007) 152–169.
5. Claus Diem, On the discrete logarithm problem in elliptic curves over non-prime finite fields, Lecture at ECC 2004, 2004.
6. Claus Diem, On the discrete logarithm problem in class groups of curves, Mathematics of Computation, **80** (2011) 443-475
7. Claus Diem, On the discrete logarithm problem in elliptic curves, Compositio Math. **147**, No. 1 (2011) 75–104.
8. Claus Diem, On the discrete logarithm problem in elliptic curves II, Algebra and Number Theory, **7**, No. 6 (2013) 1281–1323.
9. Iwan M. Duursma, Pierrick Gaudry and Francois Morain, Speeding up the discrete logarithm computation on curves with automorphisms, in K.Y. Lam, E. Okamoto and C. Xing (eds.), ASIACRYPT 1999, Springer LNCS 1716 (1999) 103–121.
10. Niklas Eén and Niklas Sörensson, The Minisat Page, `http://www.minisat.se/`
11. Jean-Charles Faugère, Ludovic Perret, Christophe Petit and Guénaël Renault, Improving the Complexity of Index Calculus Algorithms in Elliptic Curves over Binary Fields, in D. Pointcheval and T. Johansson (eds.), EUROCRYPT 2012, Springer LNCS 7237 (2012) 27–44.
12. Jean-Charles Faugère, Pierrick Gaudry, Louise Huot and Guénaël Renault, Using Symmetries in the Index Calculus for Elliptic Curves Discrete Logarithm, to appear in Journal of Cryptology (2014). doi: `10.1007/s00145-013-9158-5`
13. Jean-Charles Faugère, Louise Huot, Antoine Joux, Guénaël Renault and Vanessa Vitse, Symmetrized summation polynomials: Using small order torsion points to speed up elliptic curve index calculus, in P. Q. Nguyen and E. Oswald (eds.), EUROCRYPT 2014, Springer LNCS 8441 (2014) 40-57.
14. Jean-Charles Faugère, P. Gianni, D. Lazard, and T. Mora. Efficient Computation of zero-dimensional Gröbner bases by change of ordering, Journal of Symbolic Computation, **16**, No. 4 (1993) 329–344.
15. Pierrick Gaudry, Florian Hess, and Nigel P. Smart, Constructive and destructive facets of Weil descent on elliptic curves, J. Crypt., **15**, no. 1 (2002) 19–46.
16. Pierrick Gaudry, Index calculus for abelian varieties of small dimension and the elliptic curve discrete logarithm problem, Journal of Symbolic Computation, **44**, no. 12 (2009) 1690–1702.
17. Carla P. Gomes, Bart Selman and Henry Kautz, Boosting combinatorial search through randomization, in Mostow J. and Rich C. (eds.), Proceedings AAAI-98 , AAAI (1998) 431–437.

18. Robert Granger, On the Static Diffie-Hellman Problem on Elliptic Curves over Extension Fields, in M. Abe (ed.), ASIACRYPT 2010, Springer LNCS 6477 (2010) 283–302.
19. Yun-Ju Huang, Christophe Petit, Naoyuki Shinohara and Tsuyoshi Takagi, Improvement of Faugère et al.'s Method to Solve ECDLP, in K. Sakiyama and M. Terada (eds.), IWSEC 2013, Springer LNCS 8231 (2013) 115–132.
20. Antoine Joux, Reynald Lercier, David Naccache and Emmanuel Thomé, Oracle-Assisted Static Diffie-Hellman Is Easier Than Discrete Logarithms, in M. G. Parker (ed.), IMA Int. Conf. Cryptography and Coding, Springer LNCS 5921 (2009) 351–367.
21. Antoine Joux and Vanessa Vitse, Cover and Decomposition Index Calculus on Elliptic Curves Made Practical - Application to a Previously Unreachable Curve over $\mathbb{F}_{p^6}$, in D. Pointcheval and T. Johansson (eds.), EUROCRYPT 2012, Springer LNCS 7237 (2012) 9–26.
22. Antoine Joux and Vanessa Vitse, Elliptic Curve Discrete Logarithm Problem over Small Degree Extension Fields - Application to the Static Diffie-Hellman Problem on $E(\mathbb{F}_{q^5})$, J. Cryptology, **26**, no. 1 (2013) 119–143.
23. Neal Koblitz and Alfred Menezes, Another look at non-standard discrete logarithm and Diffie-Hellman problems, J. Mathematical Cryptology, **2**, No. 4 (2008) 311–326.
24. Cameron McDonald, Chris Charnes and Josef Pieprzyk, Attacking Bivium with MiniSat, ECRYPT Stream Cipher Project, Report 2007/040 (2007).
25. Christophe Petit and Jean-Jacques Quisquater, On Polynomial Systems Arising from a Weil Descent, in X. Wang and K. Sako (eds.), ASIACRYPT 2012, Springer LNCS 7658 (2012) 451–466.
26. Michael Shantz and Edlyn Teske, Solving the Elliptic Curve Discrete Logarithm Problem Using Semaev Polynomials, Weil Descent and Gröbner Basis Methods - An Experimental Study, in M. Fischlin and S. Katzenbeisser (eds.), Number Theory and Cryptography - Papers in Honor of Johannes Buchmann on the Occasion of His 60th Birthday, Springer LNCS 8260 (2013) 94–107.
27. Igor A. Semaev, Summation polynomials and the discrete logarithm problem on elliptic curves, Cryptology ePrint Archive, Report 2004/031, 2004.
28. Niklas Sörensson and Niklas Eén, MiniSat - A SAT Solver with Conflict-Clause Minimization. Proc. Theory and Applications of Satisfiability Testing (SAT 05) 2005.
29. Niklas Sörensson and Niklas Eén, Minisat 2.1 and minisat++ 1.0, SAT race 2008 editions, SAT (2008) 31–32.
30. Bo-Yin and Jiun-Ming Chen, Theoretical analysis of XL over small fields, in H. Wang, J. Pieprzyk and V. Varadharajan (eds.), ACISP, Springer LNCS 3108 (2004) 277–288.
31. Bo-Yin Yang, Jiun-Ming Chen and Nicolas Courtois, On asymptotic security estimates in XL and gröbner bases-related algebraic cryptanalysis, in J. Lopez, S. Qing and E. Okamoto (eds.), ICICS, Springer LNCS 3269 (2004) 401–413.
32. Wieb Bosma, John Cannon and Catherine Playoust, The Magma algebra system. I. The user language, J. Symbolic Comput., 24 (1997) 235–265.