Index calculus in the trace zero variety

Elisa Gorla, Maike Massierer

INDEX CALCULUS IN THE TRACE ZERO VARIETY

ELISA GORLA AND MAIKE MASSIERER

Abstract. We discuss how to apply Gaudry's index calculus algorithm for abelian varieties to solve the discrete logarithm problem in the trace zero variety of an elliptic curve. We treat in particular the practically relevant cases of field extensions of degree 3 or 5. Our theoretical analysis is compared to other algorithms present in the literature, and is complemented by results from a prototype implementation.

1. Introduction

Given an elliptic curve E defined over a finite field $\mathbb{F}_q$, consider the group $E(\mathbb{F}_{q^n})$ of rational points over a field extension of prime degree n. Since E is defined over $\mathbb{F}_q$, the group $E(\mathbb{F}_{q^n})$ contains the subgroup $E(\mathbb{F}_q)$ of $\mathbb{F}_q$-rational points of E. Moreover, it contains the subgroup T_n of points $P \in E(\mathbb{F}_{q^n})$ whose trace $P + \varphi(P) + \ldots + \varphi^{n-1}(P)$ is zero, where φ denotes the Frobenius homomorphism on E. The group T_n is called the trace zero subgroup of $E(\mathbb{F}_{q^n})$, and it is the group of $\mathbb{F}_q$-rational points of the trace zero variety relative to the field extension $\mathbb{F}_{q^n}|\mathbb{F}_q$.

In this paper, we study the hardness of the DLP in the trace zero variety. Our interest in this question has several motivations. First of all, supersingular trace zero varieties can achieve higher security per bit than supersingular elliptic curves, as shown by Rubin and Silverberg in [RS02, RS09] and by Avanzi and Cesena in [AC07, Ces10]. Ideally, in pairing-based protocols the embedding degree k is such that the DLP in T_n and in $\mathbb{F}^*_{q^{kn}}$ have the same complexity. In order to achieve this, an accurate assessment of the complexity of the DLP in T_n is necessary. Moreover, since T_n is isomorphic to $E(\mathbb{F}_{q^n})/E(\mathbb{F}_q)$, the DLP in $E(\mathbb{F}_{q^n})$ has the same complexity as the DLP in T_n. This provides another motivation to study the hardness of the DLP in T_n. A further motivation comes from the fact that the trace zero subgroup itself can be used within asymmetric cryptographic protocols using the DLP as a primitive, as proposed by Frey in [Fre98].

Using trace zero varieties in cryptographic protocols presents some advantages with respect to elliptic curves. In fact, a clever use of the Frobenius endomorphism allows us to compute the group operation more efficiently than for an elliptic curve of about the same cardinality, leading to more efficient scalar multiplication in the group (see [Fre99, Lan01, Lan04, AC07] or [ACD+06, Section 15.3.2]). This technique is analogous to the one for Koblitz curves [Kob91] and was later applied to GLV–GLS curves [GLV01, GLS11]. Another advantage is that for groups of cryptographically relevant size, the order of the group can simply be calculated using the characteristic polynomial of the Frobenius endomorphism. This allows for more efficient computation of the group order in comparison to the group of rational points of an elliptic curve over a prime field of comparable size (see [ACD+06, Section 15.3.1]). Finally, in the recent papers [GM14] and [GM13] we proposed new efficient representations for the elements of T_n, for any n. More precisely, we can represent the elements of the group with $(n-1)\log_2 q + 1$ bits, which is optimal since $|T_n| \sim q^{n-1}$. We refer the interested reader to [Gor11] for a discussion of the relevance of efficient representations.

In this paper, we discuss how to apply Gaudry's index calculus algorithm for abelian varieties to solve the discrete logarithm problem in T_n. Gaudry's algorithm first appeared in [Gau09], and proposes a general framework to do index calculus on a general abelian variety. A main difficulty of running an index calculus attack on an abelian variety is producing the relations. When the abelian variety is an elliptic curve, Gaudry proposes to use Semaev polynomials ([Sem04]) to build a system of polynomial equations, such that a solution to the system corresponds to a relation.

2010 *Mathematics Subject Classification.* primary: 14G50, 11G25, 11Y40, secondary: 11T71, 14K15, 14H52.

Key words and phrases. Elliptic curve cryptography, discrete logarithm problem, index calculus, trace zero variety.

The systems can be solved by Gröbner bases methods. The complexity of this attack depends on the size of $\mathbb{F}_q$ and the dimension of the abelian variety: Asymptotically in q, and regarding n as a constant, it has complexity $\tilde{O}(q^{2-2/(n-1)})$, which is lower than that of generic attacks on T_n and on $E(\mathbb{F}_{q^n})$ for $n \geq 5$. This leads to the lowest-complexity attack on the DLP in $E(\mathbb{F}_{q^n})$ for prime n. Other attacks, of comparable or lower complexity but which only apply to specific elliptic curves, are discussed in [GHS02, Die03, Die06, DK13, DS]. We apply Gaudry's index calculus algorithm to T_n, and demonstrate that it is feasible for $n = 3$ and q up to about 30 bits. For $n = 5$ we show that the bottleneck of the algorithm is the Gröbner basis computation. Using some tricks from [BFP08, JV12] we are able to produce relations and to solve a DLP for very small q, but the attack this yields is not feasible over fields of cryptographic size, therefore it is presently not a threat to the DLP in T_5 or $E(\mathbb{F}_{q^5})$.

We also analyze the algorithm asymptotically in n and q, and we see that the complexity is exponential in n. This is mostly due to the fact that in order to produce relations, the algorithm solves polynomial systems whose size (number of equations, number of indeterminates, degrees of the equations) depends on n, and that the Gröbner basis methods have a large complexity in these parameters. We conclude that one can only hope to produce relations with this method for small values of n.

The paper is organized as follows. We recall the functionality of index calculus algorithms and the most important definitions in connection with the trace zero variety in Section 2. Then we describe the application of Gaudry's algorithm to the trace zero variety in Section 3, and we analyze its complexity in Section 4. In Section 5, we present explicit equations and Magma experiments for $n = 3, 5$. Finally, we compare the index calculus attack with other attacks on the DLP in T_n in Section 6, and discuss the implications of our results for trace zero elliptic curve cryptosystems in Section 7.

Acknowledgements. We thank Pierrick Gaudry, Peter Schwabe, Vanessa Vitse, and Bo-Yin Yang for useful discussions on the material of this paper. We are grateful to the mathematics department of the University of Zürich for access to their computing facilities. The authors were supported by the Swiss National Science Foundation under grant no. 123393.

2. Preliminaries

2.1. **Index calculus.** The security of several public key cryptosystems, including ElGamal and DSA, is based on the hardness of the discrete logarithm problem.

Definition 2.1. Let G be a finite additive group. Given two elements $P \in G$ and $Q \in \langle P \rangle$, the *discrete logarithm problem (DLP)* is

$$\text{find an element } \ell \in \mathbb{Z}/(\operatorname{ord} P)\mathbb{Z} \text{ such that } \ell P = Q.$$

The number ℓ is called the *discrete logarithm of Q in base P*, and denoted by $\log_P Q$.

A combination of the Pollard–Rho Algorithm and the Pohlig–Hellman Algorithm can solve an instance of the DLP in any group G of known order $|G|$ in time $\tilde{O}(\sqrt{p})$, where p is the largest prime factor of $|G|$.

However, when a concrete group is given, its properties can often be exploited in order to devise more efficient attacks. A particularly powerful such class of attacks are *index calculus algorithms*, which exploit the algebraic structure of the groups that they work in. There are index calculus algorithms that compute the DLP in multiplicative groups of finite fields (namely the number field sieve for prime fields [Adl79, Gor93, JL03] and the function field sieve for fields of small to medium characteristic [Cop84, Adl94, ADH94, Sch02, JL02, JL06, Jou13a, GGMZ13a, Jou13b, GGMZ13b, BBD+14, BGJT13]), elliptic curves over extension fields [Sem04, Gau09, Die11, Die13], Picard groups of hyperelliptic curves and more generally $C_{a,b}$ curves [ADH94, Gau00, Eng02, Die06, DT08, EG07, Eng08, EGT11, VJS14], and even general abelian varieties [Gau09].

The general outline of an index calculus attack is as follows (see e.g. [EG02]). Let us assume that the goal is to compute a discrete logarithm $\ell = \log_P Q$ of an element $Q \in \langle P \rangle$ in some group G. Since we are only working in the cyclic subgroup, we may assume that $G = \langle P \rangle$.

1. **Factor base:** Choose a factor base $\mathcal{F} = \{P_1, \ldots, P_k\} \subseteq \langle P \rangle$.

2. **Relation collection:** Construct relations of the form $\alpha_j P + \beta_j Q = \sum_{i=1}^{k} m_{ij} P_i$ for $j = 1, \ldots, r > k$.
3. **Linear algebra:** Given the matrix $M = (m_{ij}) \in (\mathbb{Z}/\operatorname{ord}(P)\mathbb{Z})^{k \times r}$, compute a non-zero column vector $\gamma = (\gamma_1, \ldots, \gamma_r)^\intercal$ in the right kernel of M.
4. **Individual logarithm:** Output $\ell = -(\sum_{j=1}^{r} \alpha_j \gamma_j)(\sum_{j=1}^{r} \beta_j \gamma_j)^{-1}$ if $\sum \beta_j \gamma_j$ is invertible in $\mathbb{Z}/\operatorname{ord}(P)\mathbb{Z}$, otherwise return to step 2.

It is easy to see that this gives the correct result: Since γ is in the right kernel of M, we have $M\gamma = 0$, or equivalently

$$\sum_{j=1}^{r} m_{ij}\gamma_j = 0 \quad \text{for all } i = 1, \ldots, k.$$

Multiplying all relations from step 2 by γ_j, summing over j, and using the above equality gives

$$\sum_{j=1}^{r} \alpha_j \gamma_j P + \sum_{j=1}^{r} \beta_j \gamma_j Q = \sum_{j=1}^{r} \sum_{i=1}^{k} m_{ij} \gamma_j P_i = \sum_{i=1}^{k} \left(\sum_{j=1}^{r} m_{ij} \gamma_j \right) P_i = 0.$$

Therefore,

$$Q = -\left(\sum_{j=1}^{r} \alpha_j \gamma_j \right) \left(\sum_{j=1}^{r} \beta_j \gamma_j \right)^{-1} P = \ell P.$$

Algorithms that function in this way have been used for many years to compute discrete logarithms in groups where a concept of factorization is available. However, it was not until 2009 that Gaudry [Gau09] published an algorithm that works in abelian varieties of dimension at least 2. His idea is to translate the condition for a relation into a system of polynomial equations and to solve the system with Gröbner basis methods in order to obtain relations. We give more details on his approach in Section 3, where we apply it to the trace zero variety. The heuristic complexity of his attack is $\tilde{O}(q^{2-2/d})$ asymptotically for $q \to \infty$, where the dimension $d \geq 2$ and all other parameters associated to the variety (like the degrees of the defining equations and the size of the representation) are assumed to be constant or bounded by constants.

Since its publication, Gaudry's algorithm has been applied mostly to the Weil restriction of elliptic curves defined over extension fields. In fact, Gaudry suggests this application himself in his original article [Gau09]. A similar algorithm for elliptic curves was developed independently by Diem [Die11]. The algorithm of Gaudry and Diem was implemented by Joux and Vitse [JV12]. With several further improvements and variations, including a specialized implementation of the Gröbner basis algorithm F4 [JV11] using an idea of Traverso [Tra88], they were able to solve an instance of an oracle-assisted static Diffie–Hellman problem in $E(\mathbb{F}_{2^{155}})$, which is related to, but easier than, the DLP in the same group [GJV10]. Faugère, Perret, Petit, and Renault [FPPR12], Petit and Quisquater [PQ12], and Shantz and Teske [ST13] studied the polynomial systems that arise during this attack. They come to the conclusion that these systems are of a special shape and that special-purpose Gröbner basis techniques may lead to a significant speed-up. The application of the algorithm to Edwards curves was studied by Faugère, Gaudry, Huot, and Renault in [FGHR12, FGHR13].

Notice that this approach only threatens elliptic curves defined over extension fields and does not affect groups $E(\mathbb{F}_p)$ where p is a prime. The best attack on such groups is the Pollard–Rho attack, and the current record for computing a discrete logarithm in $E(\mathbb{F}_p)$, for p a 112-bit prime, is held by Bos, Kaihara, Kleinjung, Lenstra, and Montgomery [BKK+09], using a parallelized version of the Pollard–Rho Algorithm. Some improvements which take into account the use of the negation map in running the Pollard–Rho Algorithm are discussed in [BLS11].

Besides elliptic curves, Gaudry's algorithm for abelian varieties has been applied to the Weil restriction of hyperelliptic curves of small genus by Nagao [Nag10] and to algebraic tori by Granger and Vercauteren [GV05]. In this paper, we apply Gaudry's attack to the trace zero variety.

2.2. **The Trace Zero Variety.** Throughout this paper, let E be a smooth elliptic curve defined over a finite field $\mathbb{F}_q$ by an affine Weierstraß equation. For any extension field $\mathbb{F}$ of $\mathbb{F}_q$, the $\mathbb{F}$-rational points $E(\mathbb{F})$ on E form a group with neutral element $\mathcal{O}$, the point at infinity. When

$\mathbb{F} = \mathbb{F}_{q^n}, n \geq 1$, is a finite extension, $E(\mathbb{F}_{q^n})$ is a finite group of order about q^n. We denote by $+$ the group operation and by φ the Frobenius endomorphism on E

$$\varphi : E \to E, \quad (X, Y) \mapsto (X^q, Y^q), \quad \mathcal{O} \mapsto \mathcal{O}.$$

Throughout the paper, we denote field elements by uppercase and indeterminates by lowercase letters.

Definition 2.2. For a field extension $\mathbb{F}_{q^n}|\mathbb{F}_q$ of degree $n > 1$, the *trace map* is defined by

$$\mathrm{Tr} : E(\mathbb{F}_{q^n}) \to E(\mathbb{F}_q), \quad P \mapsto P + \varphi(P) + \ldots + \varphi^{n-1}(P).$$

When n is prime, the kernel of the trace map is called the *trace zero subgroup* of $E(\mathbb{F}_{q^n})$. We denote it by T_n.

The trace zero subgroup is isomorphic to the group of $\mathbb{F}_q$-rational points of the *trace zero variety* V_n, which is an $(n-1)$-dimensional subvariety of the Weil restriction of E: Fixing a basis $\{\zeta_0, \ldots, \zeta_{n-1}\}$ of $\mathbb{F}_{q^n}|\mathbb{F}_q$, we have $V_n(\mathbb{F}_q) \cong T_n$ via

$$(X_0, \ldots, X_{n-1}, Y_0, \ldots, Y_{n-1}) \mapsto (X_0\zeta_0 + \ldots + X_{n-1}\zeta_{n-1}, Y_0\zeta_0 + \ldots + Y_{n-1}\zeta_{n-1}). \tag{1}$$

In this paper, we consider the case $n \geq 3$, when the trace zero variety has dimension at least 2.

We study the hardness of the DLP in T_n, which is of interest in cryptography for various reasons, as explained in the Introduction. In particular, the DLP in T_n is as hard as the DLP in $E(\mathbb{F}_{q^n})$. This is shown for the analogous case of algebraic tori in [GV05], and more generally for exact sequences of abelian varieties in [GS06]. The result as we state it here is Proposition 2.4 in [GM13].

Proposition 2.3. *Let E be an elliptic curve defined over $\mathbb{F}_q$, and let T_n be the trace zero subgroup of $E(\mathbb{F}_{q^n})$ for some prime number n. Then the sequence*

$$0 \longrightarrow E(\mathbb{F}_q) \longrightarrow E(\mathbb{F}_{q^n}) \stackrel{\varphi-\mathrm{id}}{\longrightarrow} T_n \longrightarrow 0$$

is exact, and the DLP in $E(\mathbb{F}_{q^n})$ has the same complexity as the DLP in T_n.

In [GM14] we wrote an equation for the x-coordinates of the points in T_n using the Semaev polynomial. We briefly summarize how to write such an equation, starting with the definition and the main result from [Sem04].

Definition 2.4. Let $\mathbb{F}_q$ be a finite field of characteristic at least 5, and let E be a smooth elliptic curve defined over $\mathbb{F}_q$ by the affine equation

$$E : y^2 = x^3 + Ax + B.$$

The m-th *summation polynomial* or *Semaev polynomial* f_n is defined recursively by

$$\begin{aligned} f_3(z_1, z_2, z_3) &= (z_1 - z_2)^2 z_3^2 - 2((z_1 + z_2)(z_1 z_2 + A) + 2B)z_3 + (z_1 z_2 - A)^2 - 4B(z_1 + z_2) \\ f_m(z_1, \ldots, z_m) &= \mathrm{Res}_z(f_{m-k}(z_1, \ldots, z_{m-k-1}, z), f_{k+2}(z_{m-k}, \ldots, z_m, z)) \end{aligned}$$

for $m \geq 4$ and $m - 3 \geq k \geq 1$, where Res denotes the resultant.

Theorem 2.5 ([Sem04], Theorem 1)**.** *For any $m \geq 3$, let $Z_1, \ldots, Z_m$ be elements of the algebraic closure $\overline{\mathbb{F}}_q$ of $\mathbb{F}_q$. Then $f_m(Z_1, \ldots, Z_m) = 0$ if and only if there exist $Y_1, \ldots, Y_m \in \overline{\mathbb{F}}_q$ such that the points (Z_i, Y_i) are on E and $(Z_1, Y_1) + \ldots + (Z_m, Y_m) = \mathcal{O}$ in the group $E(\overline{\mathbb{F}}_q)$. Furthermore, f_m is absolutely irreducible and symmetric of degree 2^{m-2} in each variable. The total degree is $(m-1)2^{m-2}$.*

Remark 2.6. The original definition from [Sem04] is for elliptic curves defined over fields of characteristic at least 5. However, polynomials with the same properties can be defined also for characteristic 2 and 3. Therefore, all results of this paper hold, with the appropriate adjustments, over finite fields of any characteristic.

The Semaev polynomial is used in [GM14] to give the following equation for the x-coordinates of the points of T_n.

Proposition 2.7 ([GM14, Proposition 3, Remark 5]). *Let n be an odd prime, and let T_n be the trace zero subgroup of $E(\mathbb{F}_{q^n})$. Then*

$$T_n \subseteq \{(X,Y) \in E(\mathbb{F}_{q^n}) \mid f_n(X, X^q, \ldots, X^{q^{n-1}}) = 0\} \cup \{\mathcal{O}\}.$$

Moreover, we have

$$\begin{aligned} T_3 &= \{(X,Y) \in E(\mathbb{F}_{q^3}) \mid f_3(X, X^q, X^{q^2}) = 0\} \cup \{\mathcal{O}\} \\ T_5 \cup (E[3](\mathbb{F}_q) + (E[2] \cap T_5)) &= \{(X,Y) \in E(\mathbb{F}_{q^5}) \mid f_5(X, X^q, \ldots, X^{q^4}) = 0\} \cup \{\mathcal{O}\}. \end{aligned}$$

In the case when $n = 3$ or 5, for any root $X \in \mathbb{F}_{q^n}$ of $f_n(x, x^q, \ldots, x^{q^{n-1}}) = 0$ it can be decided efficiently whether $(X,Y) \in T_n$ by checking $Y \in \mathbb{F}_{q^n}$ and, if $n = 5$, by checking in addition that $X \notin \mathcal{L} := \{X_{Q+R} \mid Q + R = (X_{Q+R}, Y_{Q+R}) \in E[3](\mathbb{F}_q) + (E[2] \cap T_5),\ Q \neq \mathcal{O}\}$, where $|\mathcal{L}| \leq 16$.

As discussed in [GM14] at the end of Section 3, Weil restriction of $f_n(x, x^q, \ldots, x^{q^{n-1}}) = 0$ with respect to the coordinates

$$\begin{aligned} x &= x_0\zeta_0 + \ldots + x_{n-1}\zeta_{n-1} \\ y &= y_0\zeta_0 + \ldots + y_{n-1}\zeta_{n-1} \end{aligned}$$

and reduction modulo the polynomials $x_i^q - x_i$ yield exactly one equation

$$\tilde{f}_n(x_0, \ldots, x_{n-1}) = 0. \tag{2}$$

Its zeros describe the x-coordinates of the points of $V_n(\mathbb{F}_q)$ as given by Proposition 2.7 and via the isomorphism (1). Therefore, we henceforth use (2) as an equation for the trace zero subgroup. It has total degree $(n-1)2^{n-2}$.

3. An index calculus algorithm for the trace zero variety

Following the ideas of Gaudry [Gau09], we propose the following index calculus algorithm to compute discrete logarithms in T_n. When $n = 2$, then V_n is one-dimensional, and the attack cannot be applied. Therefore, we only consider $n \geq 3$. Furthermore, we assume that T_n is cyclic, which is the most relevant case in cryptography.

Remark 3.1. When T_n is not cyclic, some of the probability estimates in Section 4 may be wrong and the algorithm may not function as expected. However, these problems can be overcome using classical randomization techniques (see [Gau09, Remark 2], [EG02]).

The algorithm takes as input two points $P, Q \in T_n$ such that $T_n = \langle P \rangle$, and it outputs the discrete logarithm $\log_P Q$, i.e. a number $\ell = \log_P Q \in \mathbb{Z}/\operatorname{ord}(P)\mathbb{Z}$ such that $\ell P = Q$ in T_n. Below, we describe the different steps of the algorithm in detail. We always identify T_n and $V_n(\mathbb{F}_q)$ via the isomorphism (1).

3.1. **Setup.** Following the suggestion of Semaev [Sem04], we carry out the index calculus algorithm working only with the x-coordinates of points in T_n. We choose a basis $\{\zeta_0, \ldots, \zeta_{n-1}\}$ of the extension $\mathbb{F}_{q^n}|\mathbb{F}_q$ and represent an affine point $P = (X,Y) \in T_n$ via the coordinates

$$P = (X_0, \ldots, X_{n-1}),$$

where $X = X_0\zeta_0 + X_1\zeta_1 + \ldots + X_{n-1}\zeta_{n-1}$. So by writing $(X_0, \ldots, X_{n-1}) \in T_n$ we mean that there exists a Y such that $(X,Y) \in T_n$. We use (2) as an equation for T_n.

3.2. **Factor base.** We define the factor base

$$\mathcal{F} = \{(0, \ldots, 0, X_{n-2}, X_{n-1}) \in T_n\}.$$

These are the $\mathbb{F}_q$-rational points of a curve $\mathcal{C}$ in V_n obtained by intersecting V_n with the hyperplanes $\{x_0 = 0\}, \ldots, \{x_{n-3} = 0\}$. Since V_n has dimension $n-1$, intersecting with $n-2$ hyperplanes generically gives a curve $\mathcal{C}$. Thus $\mathcal{F} = \mathcal{C}(\mathbb{F}_q)$ has about q elements by the Theorem of Hasse–Weil, provided that $\mathcal{C}$ is absolutely irreducible.

Remark 3.2. Important properties of the factor base are that it has about q elements (this will be used in the complexity analysis, see Section 4) and that its elements can be described via algebraic equations (this will allow us to describe relations via a polynomial system, see Section 3.3). A further very important property is that the factor base must generate a large part of T_n, so that many elements of T_n decompose over the factor base. For this reason, the curve $\mathcal{C}$ should not be contained in any proper abelian subvariety of V_n. Notice that this can easily be detected in practice, since the algorithm will find practically no relations when $\mathcal{C}$ is an abelian subvariety of V_n.

Moreover, the fact that $|\mathcal{F}| \approx q$ can be proven (with the Theorem of Hasse–Weil) only if we assume that $\mathcal{C}$ is smooth and absolutely irreducible. In practice, if setting $x_0 = \ldots = x_{n-3} = 0$ does not produce a factor base with the desired properties, we simply make a different choice of hyperplanes. In our exposition we assume that the choice we have made is a good one. This is true in all our experiments.

Using equation (2), we see that any element $(0, \ldots, 0, X_{n-2}, X_{n-1}) \in \mathcal{F}$ satisfies the equation $\tilde{f}_n(0, \ldots, 0, X_{n-2}, X_{n-1}) = 0$. Conversely, the $\mathbb{F}_q$-solutions (X_{n-2}, X_{n-1}) of

$$\tilde{f}_n(0, \ldots, 0, x_{n-2}, x_{n-1}) = 0 \tag{3}$$

yield x-coordinates of points in $\mathcal{F}$ via $(X_{n-2}, X_{n-1}) \mapsto X_{n-2}\zeta_{n-2} + X_{n-1}\zeta_{n-1}$, provided that the corresponding y-coordinates are in $\mathbb{F}_{q^n}$ and up to a few exceptions, as explained above (see also Proposition 2.7). Therefore, enumerating the factor base essentially amounts to finding all solutions of (3).

3.3. **Relation collection.** Since V_n has dimension $n-1$, we search for relations of the form

$$R = P_0 + \ldots + P_{n-2}, \tag{4}$$

where $R = \alpha P + \beta Q \in T_n$ is given and $P_0, \ldots, P_{n-2} \in \mathcal{F}$ are to be found. We write $U = U_0\zeta_0 + U_1\zeta_1 + \ldots + U_{n-1}\zeta_{n-1}$ for the x-coordinate of R.

Following [Sem04], we use the Semaev polynomial to describe a relation. If the points $P_0, \ldots, P_{n-2}$ with x-coordinates $X_{P_0}, \ldots, X_{P_{n-2}}$ are given, then according to Theorem 2.5 they satisfy (4) if and only if $f_n(X_{P_0}, \ldots, X_{P_{n-2}}, U) = 0$. Therefore, candidates for x-coordinates of the P_i can be found by solving

$$f_n(x_{P_0}, \ldots, x_{P_{n-2}}, U) = 0 \tag{5}$$

for the x_{P_i}. We apply Weil restriction to equation (5) using the coordinates

$$x_{P_i} = x_{i,0}\zeta_0 + x_{i,1}\zeta_1 + \ldots + x_{i,n-1}\zeta_{n-1}$$

and obtain n equations

$$F_j(x_{0,0}, \ldots, x_{n-2,n-1}, U_0, \ldots, U_{n-1}) = 0, \quad j = 0, \ldots, n-1. \tag{6}$$

Solving this system over $\mathbb{F}_q$ is equivalent to solving equation (5) over $\mathbb{F}_{q^n}$, and yields possible x-coordinates for the points P_i.

In addition to requiring that the P_i's sum to R, we must ensure that they belong to the factor base. Therefore, we set $x_{i,0} = \ldots = x_{i,n-3} = 0$ for $i = 0, \ldots, n-1$, and we include an equation of the form (3) in system (6) for each P_i. This means that in order to find a relation, we solve the system

$$\begin{aligned}
F_0(0, \ldots, 0, x_{0,n-2}, x_{0,n-1}, \ldots, 0, \ldots, 0, x_{n-2,n-2}, x_{n-2,n-1}, U_0, \ldots, U_{n-1}) &= 0 \\
&\vdots \\
F_{n-1}(0, \ldots, 0, x_{0,n-2}, x_{0,n-1}, \ldots, 0, \ldots, 0, x_{n-2,n-2}, x_{n-2,n-1}, U_0, \ldots, U_{n-1}) &= 0 \\
\tilde{f}_n(0, \ldots, 0, x_{0,n-2}, x_{0,n-1}) &= 0 \\
&\vdots \\
\tilde{f}_n(0, \ldots, 0, x_{n-2,n-2}, x_{n-2,n-1}) &= 0
\end{aligned} \tag{7}$$

over $\mathbb{F}_q$. The system has $2n-1$ equations in $2(n-1)$ indeterminates, two indeterminates for each of the P_i's. The first n equations are the Weil descent of the n-th Semaev polynomial, where a constant has been plugged in for the last indeterminate. Therefore, they each have total degree at most $(n-1)2^{n-2}$. They describe the condition that the points P_i sum to R. The last $n-1$

equations also have total degree at most $(n-1)2^{n-2}$. They guarantee that the solution points P_i belong to the factor base.

Since the system has more equations than unknowns, one would expect that it generically has no solutions over the algebraic closure and that, when it has solutions, then it is zero-dimensional. This is verified in our experiments. Then, by the Shape Lemma (see e.g. [KR00, Theorem 3.7.25]), the system may be solved by computing a lexicographic Gröbner basis and then finding the $\mathbb{F}_q$-roots of a univariate polynomial. Notice that, in order to find the $\mathbb{F}_q$-roots of a polynomial $f(x) \in \mathbb{F}_q[x]$, one would first find the divisor $g(x)$ of $f(x)$ which is the product of all linear factors of $f(x)$ over $\mathbb{F}_q$, then factor $g(x)$, whose degree equals the number of solutions of the system over $\mathbb{F}_q$. Again, this is the case only after a generic change of coordinates. In the examples we computed however, a change of coordinates was never needed.

Whenever a given point R decomposes over the factor base, i.e. when a relation of the form (4) exists, this gives a solution of system (7). The converse, however, is not true. For example, when the solutions of the system give x-coordinates where one of the corresponding y-coordinates is not in $\mathbb{F}_{q^n}$, then this does not produce a valid relation. In theory, it is also possible that a system produces more than one relation. However, we expect this to be extremely rare, since it would produce a relation among the elements of the factor base. In accordance with this intuition, we never encountered a system with more than one solution in our experiments.

Remark 3.3. Joux and Vitse [JV12] propose considering relations that involve one factor base point less than suggested by Gaudry, i.e. only $n-2$ points in our case. This reduces the probability of finding relations by a factor q, but in some cases it can make the difference between a manageable and an unmanageable system. We consider this idea in Section 5.2.

Finally, we need to produce more relations than there are factor base elements, i.e. about q, by solving the system sufficiently many times (see Section 4 for an estimate) for different random points R.

3.4. **Linear algebra.** The relation collection phase of the algorithm produces a sparse matrix of size about $q \times q$ with entries 0 or 1. Notice that, while it is theoretically possible to have a row whose entries are positive numbers greater than 1, this should be extremely rare and in fact we never encountered such a relation in our experiments. The rows of the matrix correspond to the factor base elements, and the columns correspond to the different relations. Generically a column has $n-1$ non-zero entries, one for each factor base element that appears in the corresponding relation. Assuming that more relations have been produced than there are factor base elements, the matrix has more columns than rows. Therefore, there exists a non-zero vector in its right kernel. The task of the linear algebra step is to find such a vector, where the computations must be performed not over $\mathbb{Z}$, but modulo the order of P in T_n. Standard methods to solve such sparse linear systems are Wiedemann's Algorithm and Lanczos' Algorithm (see [Wie86, LO90]).

Remark 3.4. Since there are efficient and well-studied methods for solving sparse linear systems, we do not treat this step in detail. Notice however that the efficient implementation of the linear algebra step is far from trivial, especially since the algorithms are hard to parallelize. One recent record-breaking implementation on GPUs is presented in [Jel13, Jel14]. Moreover, in practice a filtering step can make a big difference, see e.g. [Bou12]. This is a preprocessing of the matrix, where duplicate relations are removed, points that appear in only one relation (corresponding to rows with only one nonzero entry) are removed, and excess relations are removed until there are exactly $|\mathcal{F}|+1$ of them left. We do not employ such sophisticated techniques in our experiments, since we treat only small examples and our emphasis is on finding relations and not on the linear algebra step.

3.5. **Individual logarithm.** Once the linear system has been solved, computing the actual discrete logarithm is easy. Denoting by $(\gamma_1, \ldots, \gamma_r)$ the vector in the kernel of the matrix computed in the previous step and by α_j, β_j the values of α, β corresponding to the j-th relation we have

$$\log_P Q = -\left(\sum_{j=1}^{r} \gamma_j \alpha_j\right)\left(\sum_{j=1}^{r} \gamma_j \beta_j\right)^{-1},$$

provided that $\sum \gamma_j \beta_j$ is invertible modulo the order or P. If not, one must collect more relations in order to produce a different matrix and find a different vector γ. Notice that $\sum \gamma_j \beta_j$ is invertible with high probability, especially if $\mathrm{ord}(P)$ is prime.

4. Complexity analysis

We now analyze the complexity of the index calculus algorithm presented in the previous section. We make the same heuristic assumptions as Gaudry [Gau09] and other work based on Gaudry's results, e.g. [GV05, JV12]. Our analysis is in q and n and therefore more precise than that of Gaudry, who disregards the dependency on n. By disregarding the dependency on n in our analysis, one obtains the result of Gaudry. For simplicity we use the $\tilde{O}$-notation, which ignores logarithmic factors in both n and q.

4.1. **Setup.** Diem [Die11] shows that the n-th Semaev polynomial and its Weil restriction can be computed with a randomized algorithm in expected time polynomial in $\tilde{O}(e^{n^2})$.

Remark 4.1. We do not have to compute the full Weil restriction of $f_n(x_i, x_i^q, \ldots, x_i^{q^{n-1}})$ or of $f_n(x_{P_0}, x_{P_1}, \ldots, x_{P_{n-2}}, u)$, since we only need to evaluate the polynomials on the x-coordinates of points in the factor base. Therefore, when computing the Weil restriction, we work with the coordinates $x_{P_i} = x_{i,n-2}\zeta_{n-2} + x_{i,n-1}\zeta_{n-1}$. In practice, this procedure is much quicker than first computing the usual Weil restriction and then setting $x_{i0} = \ldots = x_{i,n-3} = 0$, and the complexity is lower than the one given in [Die11]. However, since this term will not dominate the final complexity of the index calculus algorithm, the complexity estimate by Diem suffices for our purposes.

We choose to treat u, the x-coordinate of R, as an indeterminate. Then we only have to compute the Weil restriction once to obtain system (7). Each time we plug a value for the x-coordinate of R into system (7), we obtain a system which possibly produces a relation.

4.2. **Factor base.** In order to enumerate the factor base, we go through all values $X_{n-2} \in \mathbb{F}_q$, compute the solutions of $\tilde{f}_n(0, \ldots, 0, X_{n-2}, x_{n-1}) = 0$ over $\mathbb{F}_q$, and check whether the solution gives a point in T_n. Since the degree of $\tilde{f}_n$ in x_{n-1} is bounded by $(n-1)2^{n-2}$, computing all solutions takes $\tilde{O}((n-1)2^{n-2})$ operations in $\mathbb{F}_q$ (see [GvzG99, Corollary 14.16]). Typically, there are only few solutions. Checking whether the y-coordinate corresponding to $X = X_{n-2}\zeta_{n-2} + X_{n-1}\zeta_{n-1}$ is in $\mathbb{F}_{q^n}$ is much cheaper. Altogether, enumerating the factor base costs

$$\tilde{O}(q(n-1)2^{n-2}).$$

4.3. **Relation generation.** Assuming that most *different* unordered $(n-1)$-tuples of factor base elements sum to *different* points in T_n, then $|\mathcal{F}|^{n-1}/(n-1)!$ points of T_n decompose over the factor base. Since T_n has about q^{n-1} elements, this means that the probability of a point $R \in T_n$ splitting over the factor base is $1/(n-1)!$. Therefore, in order to generate q relations, we expect to have to try to decompose $q(n-1)!$ points, i.e. solve $q(n-1)!$ systems.

In order to solve each system, we follow the approach that is most efficient in practice: We first compute a Gröbner basis with respect to the degree reverse lexicographic term order, and we then use a Gröbner walk algorithm to convert it to a lexicographic Gröbner basis. Afterwards, we factor a univariate polynomial. The complexity of the last step is negligible compared to the first two.

To estimate the complexity of the Gröbner basis computation, we use the bound on the complexity of Faugère's F5 algorithm [Fau02]. We assume that the system is semi-regular, which is true generically. Then according to [BFSY05, Proposition 6], the complexity of computing a degree reverse lexicographic Gröbner basis of our system is

$$O\left(\binom{d_{\mathrm{reg}} + 2n - 2}{2n-2}^{\omega}\right),$$

where $2 \leq \omega \leq 3$ is the linear algebra constant (i.e. the exponent in the complexity of matrix multiplication) and d_{reg} is the degree of regularity of the system (this is also called the regularity index, see [KR05, Definition 5.1.8]).

We estimate d_{reg} using a standard bound from commutative algebra

$$d_{\mathrm{reg}} \leq (2n-2)((n-1)2^{n-2} - 1) + 1 = (2n-2)(n-1)2^{n-2} - 2n + 3.$$

Hence the complexity of computing a degree reverse lexicographic Gröbner basis of our system is

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

Now using the FGLM algorithm [FGLM93], we may compute from this basis a lexicographic Gröbner basis in

$$O((2n-2)\cdot D^3),$$

where D is the degree of the ideal generated by the degree reverse lexicographic Gröbner basis (i.e. the number of solutions counted with multiplicity in $\overline{\mathbb{F}}_q$). Using as a bound on D the product of the degrees of $2n-2$ of the equations of the system, we get

$$D \leq ((n-1)2^{n-2})^{2n-2}.$$

Therefore, this is not more expensive than F5.

Taking into account that we have to do this $q(n-1)!$ times, the total cost of the relation collection step is

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

4.4. **Linear algebra.** Using Lanczos' or Wiedemann's Algorithm, the cost of solving a sparse linear system of size about $q \times q$, where each column has $n-1$ non-zero entries, is

$$O((n-1)q^2)$$

(see e.g. [EK97]).

4.5. **Individual logarithm.** The cost of computing the individual logarithm is negligible compared to the complexities above.

Putting everything together, we get that the algorithm has a total complexity of

$$\tilde{O}\left(\binom{(2n-2)(n-1)2^{n-2}+1}{2n-2}^{\omega}(n-1)!q+(n-1)q^2\right).$$

4.6. **Double large prime variation.** As suggested by Gaudry, we may use the double large prime variation [Thé03, GTTD07] in order to rebalance the complexity of the relation collection and the linear algebra step in q. Then one must collect $q^{2-2/(n-1)}$ relations instead of q and solve a linear system of size $q^{1-1/(n-1)} \times q^{1-1/(n-1)}$ instead of $q \times q$. Then the overall cost of the algorithm becomes

$$\tilde{O}\left(\binom{(2n-2)(n-1)2^{n-2}+1}{2n-2}^{\omega}(n-1)!q^{2-2/(n-1)}\right).$$

Hence we have proven the following heuristic result.

Theorem 4.2. *Let $T_n, n \geq 3$, be the trace zero subgroup of an elliptic curve. Then there exists a probabilistic algorithm that computes discrete logarithms in T_n in heuristic time*

$$\tilde{O}\left(\binom{(2n-2)(n-1)2^{n-2}+1}{2n-2}^{\omega}(n-1)!q^{2-2/(n-1)}\right)$$

where n is constant and q tends to infinity. The constant in the $\tilde{O}$ does not depend on q or n.

The heuristic nature of Theorem 4.2 is due to the following heuristic assumptions, which are standard assumptions in this context, see e.g. [Gau09]. First of all, we assume that (after a randomization of coordinates) there is a choice of hyperplanes which, upon intersection with V_n, produce an absolutely irreducible smooth curve in V_n, whose $\mathbb{F}_q$-rational points define a factor base of cardinality about q (see Remark 3.2), and so that the sums of $n-1$ factor base points produce about $q^{n-1}/(n-1)!$ different elements of $V_n(\mathbb{F}_q)$. Secondly, we assume that the systems to be solved are either empty or zero-dimensional, and semi-regular. Finally, we assume that (after a randomization of coordinates), T_n is cyclic, as explained in Remark 3.1.

Remark 4.3. In particular, q must be sufficiently large compared to n. More precisely, we need that

$$q^{n-1} > (n-1)! q^{2(1-1/(n-1))(1-2/(n-1))}.$$

This is due to the fact that we need to be able to find enough relations: Taking into account the double large prime variation, we need to produce $q^{2-2/(n-1)}$ relations, and the probability of finding a relation is $1/((n-1)!q^{1-2/(n-1)})$. Therefore we expect to have to try to decompose about $(n-1)!q^{2(1-1/(n-1))(1-2/(n-1))}$ points, hence T_n must have at least that many points.

If we allow the constant in the $\tilde{O}$ to depend on n but not on q, Theorem 4.2 gives the heuristic complexity of $\tilde{O}(q^{2-2/(n-1)})$ from [Gau09]. Our analysis makes the exponential dependency on n explicit. The exponential dependency of the complexity on n was already pointed out by Gaudry and is due to the cost of the Gröbner basis computation. Notice that one cannot hope to get subexponential complexity in n for generic systems, due to the complexity bound for F5, which is exponential in n in our situation.

5. Explicit equations and experiments

We now study the systems of polynomial equations that describe the relations and the overall behavior of our algorithm for $n = 3, 5$. All computations were done with Magma version 2.19.3 [BCP97] on one core of an Intel Xeon X7550 Processor (2.00 GHz) on a Fujitsu Primergy RX900S1. Our implementation is only meant to be a proof of concept. It is a straightforward implementation of the algorithm described in Section 3, and we use the built-in Magma routines wherever possible, e.g. for Gröbner basis computation, polynomial factorization, and linear algebra. Our timings are only meant as an indication, and they could be improved significantly by a special-purpose implementation, using current state-of-the-art methods such as [BBD+14], and by choosing convenient parameters, such as finite fields where particularly efficient arithmetic is possible. We concentrate mostly on the computation of the equations of the trace zero subgroup, the factor base, and the relation generation. In particular, we did not implement any filtering (except for not allowing duplicate relations), we did not implement the double large prime variation, and our implementation is not parallelized.

5.1. **Explicit equations for $n = 3$.** When $n = 3$, the trace zero variety has dimension 2. Therefore, the index calculus attack on T_3 is not more efficient than generic (square root) attacks on T_3. Since $n = 3$ is the case where all equations are small enough to be written down explicitly, we present them nevertheless, together with some experimental data that allows us to make predictions on the feasibility of this attack for different values of q.

For simplicity, we assume that $3 \mid q-1$ and write $\mathbb{F}_{q^3} = \mathbb{F}_q[\zeta]/(\zeta^3 - \mu)$ as a Kummer extension of $\mathbb{F}_q$ with basis $1, \zeta, \zeta^2$. For cases where this is not possible, one may use a normal basis, which gives similar equations. We also assume that $\mathbb{F}_q$ does not have characteristic 2 or 3 and that E is given by an equation in short Weierstraß form

$$E : y^2 = x^3 + Ax + B.$$

Our approach also works when $\mathbb{F}_q$ has characteristic 2 or 3, but in this case the definition of the Semaev polynomial and all equations given below must be adjusted (see Remark 2.6).

The third Semaev polynomial is

$$f_3(z_1, z_2, z_3) = (z_1 - z_2)^2 z_3^2 - 2((z_1 + z_2)(z_1 z_2 + A) + 2B)z_3 + (z_1 z_2 - A)^2 - 4B(z_1 + z_2),$$

and the Weil restriction of $f_3(x, x^q, x^{q^2})$ is

$$\begin{aligned} \tilde{f}_3(x_0, x_1, x_2) \quad = \quad & -3x_0^4 - 12\mu^2 x_0 x_2^3 - 12\mu x_0 x_1^3 + 18\mu x_0^2 x_1 x_2 \\ & + 9\mu^2 x_1^2 x_2^2 - 6Ax_0^2 + 6A\mu x_1 x_2 - 12Bx_0 + A^2. \end{aligned}$$

We write points of T_3 as tuples (X_0, X_1, X_2) that satisfy $\tilde{f}_3(X_0, X_1, X_2) = 0$. For the factor base, we choose those points with $X_0 = 0$, namely

$$\mathcal{F} = \{(0, X_1, X_2) \in T_3\}.$$

These are precisely the points in T_3 that satisfy

$$\tilde{f}_3(0, X_1, X_2) = 9\mu^2 X_1^2 X_2^2 + 6A\mu X_1 X_2 + A^2 = (3\mu X_1 X_2 + A)^2 = 0. \tag{8}$$

If $A = 0$, then this is equivalent to

$$X_1 X_2 = 0,$$

and it is particularly easy to enumerate the factor base: One simply checks which x-coordinates $(0, X_1, 0)$ and $(0, 0, X_2)$, for $X_1, X_2 \in \mathbb{F}_q$, give points in T_3. If, on the other hand, $A \neq 0$, then every solution of (8) satisfies $X_1 \neq 0$, and moreover (8) is equivalent to

$$X_2 = -\frac{A}{3\mu X_1}. \tag{9}$$

In this case, it is also fairly easy to enumerate the factor base: For every $X_1 \in \mathbb{F}_q^*$, one computes X_2 according to (9) and checks whether this yields a point of T_3.

Now we need to find relations of the form

$$R = P_0 + P_1,$$

where R with x-coordinate $U = U_0 + U_1\zeta + U_2\zeta^2$ is given and P_1, P_2 are in $\mathcal{F}$. We denote by $x_{P_0} = x_{01}\zeta + x_{02}\zeta^2$ the indeterminates representing the x-coordinate of P_0 and by $x_{P_1} = x_{11}\zeta + x_{12}\zeta^2$ those representing the x-coordinate of P_1. Then we have to solve

$$f_3(x_{P_0}, x_{P_1}, U) = 0,$$

or equivalently, its Weil restriction. Assuming that $A \neq 0$, which is the general case, from (9) we get

$$x_{02} = -\frac{A}{3\mu x_{01}} \quad \text{and} \quad x_{12} = -\frac{A}{3\mu x_{11}}.$$

Plugging this into the above system and multiplying the first two equations by $27\mu x_{01}^2 x_{11}^2$ and the third equation by $81\mu^2 x_{01}^2 x_{11}^2$ allows us to eliminate the two indeterminates x_{02} and x_{12} from the system that describes a relation. We obtain

$$\begin{aligned}
0 &= 36{x_{01}}^3 x_{11}\mu^2 A U_1 U_2 + 36 x_{01} {x_{11}}^3 \mu^2 A U_1 U_2 - 72 {x_{01}}^2 {x_{11}}^2 \mu^2 A U_1 U_2 \\
&\quad -12 x_{01} x_{11} A^2 U_0 U_2 \mu + 54 {x_{01}}^4 {x_{11}}^2 \mu^2 U_0 U_1 + 54 {x_{01}}^2 {x_{11}}^4 \mu^2 U_0 U_1 \\
&\quad -18 {x_{01}}^3 {x_{11}}^2 \mu^2 A U_2 - 18 {x_{01}}^2 {x_{11}}^3 \mu^2 A U_2 - 108 {x_{01}}^3 {x_{11}}^3 \mu^2 U_0 U_1 \\
&\quad +18 x_{01} {x_{11}}^4 \mu^2 A U_2 + 18 {x_{01}}^4 x_{11} \mu^2 A U_2 + 6 {x_{01}}^2 A^2 U_0 U_2 \mu + 6 {x_{11}}^2 A^2 U_0 U_2 \mu \\
&\quad -108 {x_{01}}^2 {x_{11}}^2 B U_0 \mu - 36 {x_{01}}^2 {x_{11}}^2 A {U_0}^2 \mu + 6 {x_{01}}^2 x_{11} A^2 U_1 \mu + 6 x_{01} {x_{11}}^2 A^2 U_1 \mu \\
&\quad +18 {x_{01}}^3 x_{11} A {U_0}^2 \mu - 6 x_{01} x_{11} A^2 {U_1}^2 \mu + 18 x_{01} {x_{11}}^3 A {U_0}^2 \mu + 3 {x_{11}}^4 A^2 \mu \\
&\quad +3 {x_{01}}^4 A^2 \mu - 54 {x_{01}}^4 {x_{11}}^3 \mu^2 U_0 - 54 {x_{01}}^3 {x_{11}}^4 \mu^2 U_0 - 54 {x_{01}}^3 {x_{11}}^3 \mu^3 {U_2}^2 \\
&\quad +27 {x_{01}}^2 {x_{11}}^4 \mu^3 {U_2}^2 + 27 {x_{01}}^4 {x_{11}}^2 \mu^3 {U_2}^2 + 18 x_{01} {x_{11}}^3 A^2 \mu + 18 {x_{01}}^3 x_{11} A^2 \mu \\
&\quad +39 {x_{01}}^2 {x_{11}}^2 A^2 \mu + 3 {x_{11}}^2 A^2 {U_1}^2 \mu + 3 {x_{01}}^2 A^2 {U_1}^2 \mu - 6 {x_{01}}^3 A^2 U_1 \mu \\
&\quad -6 {x_{11}}^3 A^2 U_1 \mu + 2 x_{01} A^3 U_0 + 2 {x_{11}} A^3 U_0 \\
0 &= -72 {x_{01}}^2 {x_{11}}^2 A U_0 U_1 \mu + 36 x_{01} {x_{11}}^3 A U_0 U_1 \mu - 12 x_{01} x_{11} A^2 U_1 U_2 \mu \\
&\quad +36 {x_{01}}^3 x_{11} A U_0 U_1 \mu - 6 x_{01} x_{11} A^3 + 3 {x_{01}}^2 A^2 {U_0}^2 + 2 {x_{11}} A^3 U_1 + 2 x_{01} A^3 U_1 \\
&\quad +3 {x_{11}}^2 A^2 {U_0}^2 - 54 {x_{01}}^3 {x_{11}}^4 \mu^2 U_1 + 27 {x_{01}}^2 {x_{11}}^4 \mu^2 {U_1}^2 + 27 {x_{01}}^4 {x_{11}}^2 \mu^2 {U_1}^2 \\
&\quad -54 {x_{01}}^3 {x_{11}}^3 \mu^2 {U_1}^2 - 54 {x_{01}}^4 {x_{11}}^3 \mu^2 U_1 - 6 {x_{11}}^3 A^2 U_2 \mu - 6 {x_{01}}^3 A^2 U_2 \mu \\
&\quad -108 {x_{01}}^2 {x_{11}}^3 B \mu - 108 {x_{01}}^3 {x_{11}}^2 B \mu - 2 {x_{11}}^2 A^3 + 27 {x_{01}}^4 {x_{11}}^4 \mu^2 - 2 {x_{01}}^2 A^3 \\
&\quad -6 x_{01} x_{11} A^2 {U_0}^2 + 54 {x_{01}}^2 {x_{11}}^4 \mu^2 U_0 U_2 + 54 {x_{01}}^4 {x_{11}}^2 \mu^2 U_0 U_2 \\
&\quad -108 {x_{01}}^3 {x_{11}}^3 \mu^2 U_0 U_2 - 36 {x_{01}}^2 {x_{11}}^2 \mu^2 A {U_2}^2 + 18 x_{01} {x_{11}}^3 \mu^2 A {U_2}^2 \\
&\quad +18 {x_{01}}^3 x_{11} \mu^2 A {U_2}^2 - 18 {x_{01}}^3 {x_{11}}^2 A U_0 \mu - 18 {x_{01}}^2 {x_{11}}^3 A U_0 \mu + 6 x_{01} {x_{11}}^2 A^2 U_2 \mu \\
&\quad +6 {x_{01}}^2 x_{11} A^2 U_2 \mu - 108 {x_{01}}^2 {x_{11}}^2 B U_1 \mu + 18 x_{01} {x_{11}}^4 A U_0 \mu + 18 {x_{01}}^4 x_{11} A U_0 \mu \\
&\quad +6 {x_{01}}^2 A^2 U_1 U_2 \mu + 6 {x_{11}}^2 A^2 U_1 U_2 \mu \\
0 &= -216 {x_{01}}^2 {x_{11}}^2 A U_0 U_2 \mu^2 + 108 x_{01} {x_{11}}^3 A U_0 U_2 \mu^2 + 108 {x_{01}}^3 x_{11} A U_0 U_2 \mu^2 \\
&\quad +18 {x_{11}}^2 A^2 U_0 U_1 \mu + 18 {x_{01}}^2 A^2 U_0 U_1 \mu + 18 {x_{01}}^2 x_{11} A^2 U_0 \mu + 108 x_{01} {x_{11}}^2 B A \mu \\
&\quad +18 x_{01} {x_{11}}^2 A^2 U_0 \mu + 108 {x_{01}}^2 x_{11} B A \mu - 36 x_{01} x_{11} A^2 U_0 U_1 \mu - 162 {x_{01}}^4 {x_{11}}^3 \mu^3 U_2 \\
&\quad -162 {x_{01}}^3 {x_{11}}^4 \mu^3 U_2 + 9 {x_{11}}^2 A^2 {U_2}^2 \mu^2 + 9 {x_{01}}^2 A^2 {U_2}^2 \mu^2 - 162 {x_{01}}^3 {x_{11}}^3 A \mu^2 \\
&\quad +81 {x_{01}}^2 {x_{11}}^4 {U_0}^2 \mu^2 - 54 {x_{01}}^2 {x_{11}}^4 A \mu^2 + 81 {x_{01}}^4 {x_{11}}^2 {U_0}^2 \mu^2 - 162 {x_{01}}^3 {x_{11}}^3 {U_0}^2 \mu^2 \\
&\quad -54 {x_{01}}^4 {x_{11}}^2 A \mu^2 + A^4 - 324 {x_{01}}^2 {x_{11}}^2 B {U_2} \mu^2 + 54 {x_{01}}^4 x_{11} A U_1 \mu^2 \\
&\quad +54 {x_{01}}^3 x_{11} A {U_1}^2 \mu^2 - 18 x_{01} x_{11} A^2 {U_2}^2 \mu^2 + 54 x_{01} {x_{11}}^4 A U_1 \mu^2 + 54 x_{01} {x_{11}}^3 A {U_1}^2 \mu^2 \\
&\quad -54 {x_{01}}^3 {x_{11}}^2 A U_1 \mu^2 - 54 {x_{01}}^2 {x_{11}}^3 A U_1 \mu^2 - 108 {x_{01}}^2 {x_{11}}^2 A {U_1}^2 \mu^2 \\
&\quad +162 {x_{01}}^4 {x_{11}}^2 \mu^3 U_1 U_2 - 324 {x_{01}}^3 {x_{11}}^3 \mu^3 U_1 U_2 + 162 {x_{01}}^2 {x_{11}}^4 \mu^3 U_1 U_2 \\
&\quad -18 {x_{11}}^3 A^2 U_0 \mu + 6 {x_{11}} A^3 U_2 \mu - 18 {x_{01}}^3 A^2 U_0 \mu + 6 x_{01} A^3 U_2 \mu.
\end{aligned} \tag{10}$$

This system only involves the two indeterminates x_{01}, x_{11}. All equations have degree 4 in both x_{01} and x_{11}. The first and third equations have total degree 7, and the second equation has total degree 8. We have computed that the system (10) has regularity 14 for almost all points R

(and regularity 12 or 13 for some special choices of R). This means that the highest degree of all polynomials appearing during the Gröbner basis computation is at most 14. This moderate number suggests that the Gröbner basis computation is not very costly, and our experiments (see below) show that this is indeed true.

For a given x-coordinate U of a point $R \in T_n$, the $\mathbb{F}_q$-solutions (X_{01}, X_{11}) of the above system with $X_{01}, X_{11} \neq 0$ give candidates for x-coordinates

$$X_0 = X_{01}\zeta - \frac{A}{3\mu X_{01}}\zeta^2 \quad \text{and} \quad X_1 = X_{11}\zeta - \frac{A}{3\mu X_{11}}\zeta^2$$

of the points P_0, P_1 in the relation.

Example 5.1. We give a toy example. Let $q = 2^{12}-3, \mathbb{F}_{q^3} = \mathbb{F}_q/(\zeta^3-2)$, and $E : y^2 = x^3+x+21$. Then T_3 has order 16715869, which is a 24-bit prime, and we take

$$P = 3961 + 199\zeta + 4028\zeta^2$$

as a generator. We choose a random

$$Q = 3342 + 3020\zeta + 4031\zeta^2,$$

of which we wish to compute the discrete logarithm. The elements of the factor base satisfy

$$6x_{i1}x_{i2} + 1 = 0, \quad i = 0, 1,$$

and we compute that there are exactly 4002 such points. Now we choose random $\alpha = 4297188$ and $\beta = 10382682$, which gives $U = 2960 + 1129\zeta + 1917\zeta^2$, and we solve the system

$$\begin{aligned}
0 &= 439x_{01}^4x_{11}^3 + 1215x_{01}^4x_{11}^2 + 2556x_{01}^4x_{11} + 2274x_{01}^4 + 439x_{01}^3x_{11}^4 + 1663x_{01}^3x_{11}^3 \\
&\quad +1537x_{01}^3x_{11}^2 + 3403x_{01}^3x_{11} + 2023x_{01}^3 + 1215x_{01}^2x_{11}^4 + 1537x_{01}^2x_{11}^3 + 1961x_{01}^2x_{11}^2 \\
&\quad +2070x_{01}^2x_{11} + 2326x_{01}^2 + 2556x_{01}x_{11}^4 + 3403x_{01}x_{11}^3 + 2070x_{01}x_{11}^2 + 3534x_{01}x_{11} \\
&\quad +716x_{01} + 2274x_{11}^4 + 2023x_{11}^3 + 2326x_{11}^2 + 716x_{11} \\
0 &= 2x_{01}^4x_{11}^4 + 3670x_{01}^4x_{11}^3 + 938x_{01}^4x_{11}^2 + 609x_{01}^4x_{11} + 3670x_{01}^3x_{11}^4 + 2217x_{01}^3x_{11}^3 \\
&\quad +3400x_{01}^3x_{11}^2 + 405x_{01}^3x_{11} + 3667x_{01}^3 + 938x_{01}^2x_{11}^4 + 3400x_{01}^2x_{11}^3 + 2586x_{01}^2x_{11}^2 \\
&\quad +426x_{01}^2x_{11} + 94x_{01}^2 + 609x_{01}x_{11}^4 + 405x_{01}x_{11}^3 + 426x_{01}x_{11}^2 + 115x_{01}x_{11} \\
&\quad +345x_{01} + 3667x_{11}^3 + 94x_{11}^2 + 345x_{11} \\
0 &= 518x_{01}^4x_{11}^3 + 1692x_{01}^4x_{11}^2 + 2117x_{01}^4x_{11} + 518x_{01}^3x_{11}^4 + 2070x_{01}^3x_{11}^3 + 1976x_{01}^3x_{11}^2 \\
&\quad +1677x_{01}^3x_{11} + 1945x_{01}^3 + 1692x_{01}^2x_{11}^4 + 1976x_{01}^2x_{11}^3 + 3431x_{01}^2x_{11}^2 + 2162x_{01}^2x_{11} \\
&\quad +1057x_{01}^2 + 2117x_{01}x_{11}^4 + 1677x_{01}x_{11}^3 + 2162x_{01}x_{11}^2 + 1979x_{01}x_{11} + 71x_{01} \\
&\quad +1945x_{11}^3 + 1057x_{11}^2 + 71x_{11} + 3474.
\end{aligned}$$

We get $X_{01} = 1770, X_{11} = 1515$, and from these we compute $X_{02} = 338, X_{12} = 3029$, which gives a relation

$$P_0 + P_1 = R$$

for some choice of y-coordinates. After collecting 4002 more such relations and solving the linear system, we obtain $\log_P Q = 419$.

Finally, we present implementation results for fields of different size in Table 1. For primes q of $10, 12, 14, 16, 18, 20, 30, 40, 50, 60, 70$, and 80 bits, we chose the smallest possible value μ, and we chose curves E, given by the coefficients A, B, that yield cyclic trace zero subgroups T_3 of prime order. Where we were able to compute it, we list the exact size of the factor base. In all cases, it is close to $q - q^{1/2}$. We also list the number of points R we had to try in order to find $|\mathcal{F}| + 1$ distinct relations.

Times are given in seconds, and numbers in normal font stand for computations that we were able to perform, while numbers in bold represent expected times, extrapolated from timings we were able to obtain. For example, when we are able to compute one relation, this allows us to predict the time it would take to collect q relations (experimentally this requires solving about $2q$ polynomial systems). Where we were not able to carry out a computation or make a prediction, we write "–".

For all field sizes, we were able to solve the system at least a few times. For comparison, we give the time taken to compute a lexicographic Gröbner basis of the straightforward system consisting of 5 equations in 4 indeterminates ("large system"), as well as the time taken to compute a lexicographic Gröbner basis of system (10) consisting of 3 equations in 2 indeterminates ("small

TABLE 1. Index calculus algorithm for $n = 3$, timings in seconds

$\log_2 \lvert T_3 \rvert$	20	24	28	32	36	40
q	$2^{10} - 3$	$2^{12} - 3$	$2^{14} - 3$	$2^{16} - 15$	$2^{18} - 93$	$2^{20} - 3$
μ	5	2	2	2	2	2
A	2	1	1	1	1	1
B	0	21	11	5	10	25
$\lvert\mathcal{F}\rvert$	900	4002	16380	65388	261822	1045962
number of R's tried	2208	8263	32828	130533	522935	2091965
time for GB of large system	0.01773	0.01698	0.01705	0.01792	0.01686	0.01703
time for GB of small system	0.00102	0.00169	0.00167	0.00124	0.00146	0.00135
time to solve small system	0.00115	0.00180	0.00173	0.00134	0.00159	0.00136
time to enumerate $\mathcal{F}$	0.07	0.28	1.15	5.24	23.59	104.86
time to collect relations	3.52	13.53	49.71	197.17	803.95	2845.01
time linear algebra	0.01	0.30	5.22	108.29	129.69	–
total time	3.60	14.25	56.08	310.70	957.23	–
$\log_2 \lvert T_3 \rvert$	60	80	100	120	140	160
q	$2^{30} - 105$	$2^{40} - 87$	$2^{50} - 51$	$2^{60} - 93$	$2^{70} - 267$	$2^{79} - 67$
μ	2	2	2	2	5	3
A	1	1	1	1	1	1
B	24	49	40	193	15	368
$\lvert\mathcal{F}\rvert$	$\mathbf{2^{30}}$	$\mathbf{2^{40}}$	$\mathbf{2^{50}}$	$\mathbf{2^{60}}$	$\mathbf{2^{70}}$	$\mathbf{2^{79}}$
number of R's tried	$\mathbf{2^{31}}$	$\mathbf{2^{41}}$	$\mathbf{2^{51}}$	$\mathbf{2^{61}}$	$\mathbf{2^{71}}$	$\mathbf{2^{80}}$
time for GB of large system	0.02683	0.12645	0.12817	0.13431	0.15000	0.14102
time for GB of small system	0.00146	0.00231	0.00244	0.00249	0.00304	0.00262
time to solve small system	0.00171	0.00291	0.00342	0.00351	0.00467	0.00442
time to enumerate $\mathcal{F}$	$\mathbf{2^{17.2}}$	$\mathbf{2^{28.3}}$	$\mathbf{2^{38.5}}$	$\mathbf{2^{48.7}}$	$\mathbf{2^{59.4}}$	$\mathbf{2^{68.4}}$
time to collect relations	$\mathbf{2^{21.8}}$	$\mathbf{2^{32.5}}$	$\mathbf{2^{42.8}}$	$\mathbf{2^{52.8}}$	$\mathbf{2^{63.2}}$	$\mathbf{2^{72.1}}$

system"). This shows that this little trick to simplify the system saves a considerable amount of time in practice. Therefore, in the following, we work with the small system.

Next we list in the table the average time taken to solve the small system once. This includes computing the lexicographic Gröbner basis, factoring a univariate polynomial (of degree 6 in our experiments), which gives the value(s) of one indeterminate, and computing the corresponding value(s) of the other indeterminate. For the Gröbner basis computation, we use Magma's `GroebnerBasis()`, which computes a degree reverse lexicographic Gröbner basis using Faugère's F4 algorithm [Fau99] and subsequently a lexicographic Gröbner basis using the FGLM algorithm [FGLM93].

Finally, we give the actual or extrapolated times for the full execution of the different steps of our algorithm. First we give the time to enumerate the factor base, then the time to collect $\lvert\mathcal{F}\rvert + 1$ relations, and then the time to solve the linear system, using the sparse linear algebra routine `ModularSolution(Lanczos:=true)` of Magma, which is an implementation of Lanczos' algorithm. We also give the total time to compute one discrete logarithm with our algorithm.

We see that the largest trace zero subgroup where we can compute a full discrete logarithm with our prototype implementation has 36-bit size. The attack takes approximately 15 minutes. For the 40-bit trace zero subgroup, we can compute sufficiently many relations in about 47 minutes, but we are not able to solve the linear system of size about $2^{20} \times 2^{20}$ in Magma. A specialized implementation presented in [BBD+14, Jel13, Jel14] solves a linear system of size about $2^{22} \times 2^{22}$ in less than 5 days using a sophisticated implementation of Lanczos' algorithm, running on a high performance computer. This means that our attack is certainly feasible for a 40-bit trace zero subgroup. However, we can do much better by rebalancing the cost of relation collection and linear algebra.

Let us consider e.g. the group T_3 of 60 bits, with $q \approx 2^{30}$, as given in Table 1. We rebalance the complexity with a relatively straightforward approach. Using a factor base of $q^r = 2^{30r}$ elements, where $0 < r < 1$, the probability of finding a relation becomes $q^{2r-2}/2$. Hence in order to find q^r relations, we need to solve $2q^{2-r} = 2^{61-30r}$ systems. Since we know that solving a linear system

of size $2^{22} \times 2^{22}$ is possible, we set $q^r = 2^{22}$ and get $r = 0.73$. This means that we would have to collect 2^{39} relations, which would take $2^{39.8}$ seconds or about 30 years. Assuming that solving a linear system of size $2^{23} \times 2^{23}$ is possible, we would need about 15 years to collect relations, etc. We stress that these predictions correspond to the time required by our simple implementation. With an optimized and parallel implementation of the relation collection step (notice that the relation search can trivially be parallelized), it would become faster by a considerable factor. Hence we conclude that with an optimized implementation, computing a discrete logarithm in a 60-bit trace zero subgroup with this index calculus algorithm is feasible.

5.2. **Explicit equations for $n = 5$.** We proceed simliarly for $n = 5$, but we do not write down the equations in this case because they are too large. We assume that $5 \mid q - 1$ and write $\mathbb{F}_{q^5} = \mathbb{F}_q(\zeta)/(\zeta^5 - \mu)$. Then $1, \zeta, \zeta^2, \zeta^3, \zeta^4$ is a basis of $\mathbb{F}_{q^5}|\mathbb{F}_q$, which we use for Weil restriction.

The fifth Semaev polynomial f_5 has total degree 32. The same is true for $\tilde{f}_5(x_0, \ldots, x_4)$, which we use as an equation for T_5. The factor base is

$$\mathcal{F} = \{(0, 0, 0, X_3, X_4) \in T_5\},$$

and all its elements satisfy the equation

$$\tilde{f}_5(0, 0, 0, x_3, x_4) = 0. \tag{11}$$

It has total degree 32 and degree 30 in each x_3 and x_4. Although this polynomial does not have such a simple shape as the corresponding one for $n = 3$, it is still easy to enumerate the factor base: For every $X_3 \in \mathbb{F}_q$, solve $\tilde{f}_5(0, 0, 0, X_3, x_4) = 0$ for x_4 in $\mathbb{F}_q$.

Following an idea of Joux and Vitse [JV12] (see Remark 3.3), we look for relations of the form

$$R = P_0 + P_1 + P_2, \tag{12}$$

where P_0, P_1, P_2 are elements of the factor base. We obtain a system of 8 equations in 6 indeterminates: The first 5 equations are the Weil restriction of $f_4(x_{P_0}, x_{P_1}, x_{P_2}, U)$ and correspond to (12). They have total degree 12 and degree 4 in each indeterminate. The last 3 equations correspond to the condition that the points belong to the factor base and are of the form (11). For a given U, we solve this system in order to obtain possible relations. However, the system is too large to be solved with Magma. Even over the relatively small field $\mathbb{F}_{1021}$, our computation did not finish after several weeks of computation and using more than 300 GB of memory.

Hence we use a hybrid approach along the lines of [YCC04, BFP08]. This allows us to find some relations, but it is not fast enough for an attack of realistic cryptographic size. Nevertheless, we give some experimental results, timings, and extrapolations. The hybrid method is often used where a direct Gröbner basis computation is too costly, since it is a trade-off between exhaustive search and Gröbner basis techniques. The main idea is to choose fixed values for a small number of variables and to solve the system in the remaining indeterminates. In order to find all solutions of the system, all choices for the fixed variables have to be tried. Therefore, this requires computing many Gröbner bases of smaller systems instead of computing one Gröbner basis of a large system.

In our case, it is enough to choose one fixed value in order to solve the system readily. We fix $x_{03} = X_{03} \in \mathbb{F}_q$ and use the factor base equation $\tilde{f}_5(0, 0, 0, X_{03}, x_{04}) = 0$ to determine possible values of x_{04}. Although this equation has degree 30 in x_{04}, there are usually only very few solutions, most frequently 1, 2, or 3. In every case where $x_{04} = X_{04}$ gives a point in the factor base, we plug $x_{03} = X_{03}$ and $x_{04} = X_{04}$ into the system to obtain a new system of 7 equations in the 4 indeterminates $x_{13}, x_{14}, x_{23}, x_{24}$. The first five equations each have total degree 8 and degree 4 in every indeterminate. By trying all $X_{03} \in \mathbb{F}_q$, we find out whether R decomposes over the factor base.

We give some timings and extrapolations in Table 2. As before, numbers in normal font are times we measured, and numbers in bold are predictions. After giving the parameters of the fields and curves we used, we indicate the number of points R which we tried to decompose (we expect $6q$), the total number of polynomial systems to be solved for this (we expect $6q^2$), the time for the solution of one system (this is equal to the time for computing a Gröbner basis, since the rest of the computation to solve the system is negligible), the time to enumerate the factor base, the time to collect about q relations, the time for the linear algebra step, and the time for the total attack.

TABLE 2. Index calculus algorithm for $n = 5$, timings in seconds

$\log_2 \lvert T_5 \rvert$	20	22	27	32	36	40
q	$2^5 - 1$	$2^6 - 23$	$2^7 - 27$	$2^8 - 15$	$2^9 - 21$	$2^{10} - 3$
μ	2	2	2	3	2	2
A	1	1	1	1	1	1
B	16	3	3	13	18	1
$\lvert\mathcal{F}\rvert$	40	70	110	230	520	970
number of R's tried	886	884	2424	**5784**	**11784**	**24528**
number of systems solved	17719	30934	244824	**1393944**	**5785944**	**25043088**
time for GB of one system	1.30	1.31	1.28	1.21	1.22	1.32
time to enumerate $\mathcal{F}$	0.02	0.04	0.07	0.18	0.43	0.89
time to collect relations	25004	38219	171085	**821328**	**3818016**	**15084720**
time linear algebra	0.01	0.01	0.01	0.01	0.01	0.01
total time	25164	43618	171085	**821328**	**3818016**	**15084720**
$\log_2 \lvert T_5 \rvert$	60	80	100	120	140	160
q	$2^{15} - 157$	$2^{20} - 5$	$2^{25} - 61$	$2^{30} - 173$	$2^{35} - 547$	$2^{40} - 195$
μ	3	2	2	2	5	2
A	1	1	1	1	1	1
B	7	10	17	5	3	12
$\lvert\mathcal{F}\rvert$	32600	1051440	$\mathbf{2^{25}}$	$\mathbf{2^{30}}$	$\mathbf{2^{35}}$	$\mathbf{2^{40}}$
number of R's tried	$\mathbf{2^{20}}$	$\mathbf{2^{25}}$	$\mathbf{2^{30}}$	$\mathbf{2^{35}}$	$\mathbf{2^{40}}$	$\mathbf{2^{45}}$
number of systems solved	$\mathbf{2^{35}}$	$\mathbf{2^{45}}$	$\mathbf{2^{55}}$	$\mathbf{2^{65}}$	$\mathbf{2^{75}}$	$\mathbf{2^{85}}$
time for GB of one system	1.34	1.33	7.09	6.93	146.16	147.89
time to enumerate $\mathcal{F}$	38.80	1530.91	$\mathbf{2^{17.1}}$	$\mathbf{2^{22.9}}$	$\mathbf{2^{28.7}}$	$\mathbf{2^{34.0}}$
time to collect relations	$\mathbf{2^{34.3}}$	$\mathbf{2^{45.4}}$	$\mathbf{2^{57.8}}$	$\mathbf{2^{67.7}}$	$\mathbf{2^{82.2}}$	$\mathbf{2^{92.2}}$
time linear algebra	89.12	–	–	–	–	–
total time	$\mathbf{2^{34.3}}$	$\mathbf{2^{45.4}}$	$\mathbf{2^{57.8}}$	$\mathbf{2^{67.7}}$	$\mathbf{2^{82.2}}$	$\mathbf{2^{92.2}}$

The numbers show that we are able to compute a discrete logarithm in the 27-bit group T_5 in about 2 days and that a discrete logarithm in the 32-bit, 36-bit, and 40-bit groups T_5 can be computed in about 10, 44, and 165 days, respectively. Clearly, this approach is far from feasible for any group of cryptographic size.

We see that it is very costly to find a relation with this approach, for two reasons. Firstly, we are searching for relations that involve only 3 points of the factor base. While the probability that a point decomposes into a sum of 4 points of the factor base is $1/4! = 1/24$, the probability that it decomposes into a sum of 3 points of the factor base is $1/(3!q) = 1/(6q)$ (see Section 4). This means that we expect to have to try about $6q$ points R in order to find one that decomposes. Notice that we can still hope to find enough relations, even though the probability of finding a relation has decreased by a factor q (Joux and Vitse [JV12] have shown that such an approach is indeed advantageous in certain situations): Assuming that most distinct unordered 3-tuples of factor base elements sum to distinct points of T_5, this means that about $q^3/6$ points $R \in T_5$ decompose into a sum of 3 factor base elements. This number is much larger than q. Therefore, it is a realistic assumption that we find about q relations.

Secondly, every time we wish to check whether a given point R decomposes into a sum of 3 factor base points, we do not have to solve one system, but $O(q)$ systems, namely a small number of systems for every $X_{03} \in \mathbb{F}_q$. In practice, not all X_{03} yield valid X_{04}, therefore the number of systems to be solved is actually smaller.

6. Comparison with other attacks and discussion

We now compare the index calculus attack on the DLP in T_n with other known attacks.

6.1. **Pollard–Rho.** Assuming that T_n is cyclic of prime order, the Pollard–Rho Algorithm performs $O(q^{(n-1)/2})$ steps, and each step consists essentially of a point addition and hence has complexity $\tilde{O}(1)$. Comparing this to the complexity of the index calculus algorithm in q, which is $\tilde{O}(q^{2-2/(n-1)})$, we see that the index calculus algorithm has smaller complexity for $n \geq 5$. More

precisely, when $n = 3$ then Pollard–Rho and index calculus have the same complexity, when $n = 5$ the advantage of the index calculus attack comes only from the large prime variation (because without the large prime variation, index calculus has complexity $\tilde{O}(q^2)$), and when $n > 5$, the index calculus method always has lower complexity, independently of the large prime trick. The larger n, the larger the advantage of the index calculus algorithm over Pollard–Rho in this analysis.

However, the Pollard–Rho Algorithm has to perform only an elliptic curve point addition in each step, while the index calculus algorithm has to compute a Gröbner basis, which is much more expensive. Even in the case $n = 3$, where the system is much more manageable than for larger n, we can solve less than a thousand systems per second (cf. Table 1), whereas elliptic curve point addition can be performed at a rate of 25000 to 150000 per second (depending on the size of the field; we measured this by adding random points of T_3 in Magma, an optimized implementation can achieve much better values). For larger values of n, the difference becomes much more extreme, since the cost of elliptic curve point addition increases only at the same rate as that of finite field arithmetic in $\mathbb{F}_{q^n}$, whereas the cost of the Gröbner basis computation increases considerably. In fact the degree of the equations grows exponentially and the number of equations and variables grows linearly in n. This is reflected in the large complexity in n of the index calculus algorithm (see Theorem 4.2).

We conclude that in practice index calculus can be more efficient than Pollard–Rho only for moderate values of $n > 3$ and very large values of q. We do not know the precise crossover point.

Notice also that the variant of the index calculus algorithm for T_5 that uses the trick of Joux and Vitse and the hybrid approach has complexity $\tilde{O}(q^3)$ in q, therefore it is not better than the Pollard–Rho Algorithm for $n = 5$. It would be better only for $n > 5$.

6.2. **Index calculus on the whole curve.** The index calculus algorithm of Gaudry may also be used to compute discrete logarithms in $E(\mathbb{F}_{q^n})$ by working in the n-dimensional Weil restriction of E with respect to $\mathbb{F}_{q^n}|\mathbb{F}_q$. This is one of the original applications suggested by Gaudry in [Gau09]. From a complexity theoretic point of view, it does not make sense to attack the DLP in $E(\mathbb{F}_{q^n})$ when one wants to solve a DLP in T_n, since the complexity of Gaudry's algorithm in q depends on the dimension of the variety and therefore has complexity $\tilde{O}(q^{2-2/n})$ in $E(\mathbb{F}_{q^n})$ and complexity $\tilde{O}(q^{2-2/(n-1)})$ in T_n.

From a practical point of view, however, the systems one gets when performing index calculus on the whole curve may be more manageable, since they consist only of the Weil restriction of the Semaev polynomial, whereas in our approach, the system contains also the equations of the factor base. Moreover, when working in the whole curve, the Semaev polynomial may easily be symmetrized, which gives a system of smaller degree and with fewer solutions, whereas it is not obvious how to do this in our case. Also, when working in the whole curve, factor base elements may be represented by one $\mathbb{F}_q$-coordinate only, where we need two for the trace zero variety. Therefore, our system has twice as many indeterminates. On the other hand, the advantage of working in the trace zero variety is that relations contain $n-1$ factor base elements, and therefore one uses f_n to describe relations, whereas when working on the whole curve, relations contain n factor base elements, thus one has to use f_{n+1}. Summarizing, when working on the whole curve, one has a system of n equations in n indeterminates of total degree 2^{n-1}. In contrast, when working in the trace zero variety one has a system of $2n-1$ equations in $2n-2$ indeterminates of total degree $(n-1)2^{n-2}$.

Such subtleties are not evident in the original complexity analysis of Gaudry, which is only in q (and n is taken to be constant) and where the Gröbner basis computation thus has constant complexity. When performing an analysis similar to the one of Section 4 for Gaudry's algorithm on the whole curve, one obtains

$$\tilde{O}\left(\binom{n2^{n-1}+1}{n}^{\omega} n! q^{2-2/n}\right),$$

which is smaller in n. Therefore, which attack performs better depends on the relation between q and n.

In both cases the feasibility of the Gröbner basis computation plays an important role in practice.

6.3. **Cover attacks.** Cover attacks, also referred to as transfer attacks, were first proposed by Frey [Fre99] and further studied by many authors, including Galbraith and Smart [GS99], Gaudry, Hess, and Smart [GHS02], and Diem [Die03]. The aim of such attacks is to transfer the DLP from the algebraic variety one is considering to the Picard group of a curve of larger (but still rather low) genus, where the DLP is then solved using index calculus methods. There exist different constructions, each of them specific to a certain type of curve or variety, and there are constructions for cover attacks on $E(\mathbb{F}_{q^n})$ and on T_n directly.

For example, combining the results of [Die03] and [DK13], it is sometimes possible to map the DLP to the Picard group of a genus 5 curve (which is usually not hyperelliptic), where it can be solved in $\tilde{O}(q^{4/3})$. This is better than Gaudry's index calculus in $E(\mathbb{F}_{q^5})$, which has complexity $\tilde{O}(q^{8/5})$, and the index calculus attack on T_5, which has complexity $\tilde{O}(q^{3/2})$. However, the index calculus attack on T_5 applies to all curves, whereas only a very small proportion of curves is affected by the cover attack.

Diem and Scholten [DS, DS03] propose a cover attack for the trace zero variety directly. It works best for trace zero varieties of genus 2 curves, but it also applies to some trace zero varieties of elliptic curves. Namely, when $g = 1$ and $n = 5$, the DLP may sometimes be transferred to a curve of genus 4, where it can be solved in $\tilde{O}(q^{4/3})$. Again, this is better than the complexity of the index calculus attack, but it only affects a small number of curves (in fact, in [DS03] the authors find only one curve vulnerable to this attack). The same is true for $g = 1$ and $n = 7$, where the DLP may sometimes be mapped to a curve of genus 8 (in this case the authors cannot find any examples, although they can prove that vulnerable curves exist).

7. Conclusions on the hardness of the DLP

We conclude that applying Gaudry's index calculus algorithm for abelian varieties to the trace zero variety, as presented in this paper, yields an attack in T_n that has smaller complexity than generic algorithms whenever $n \geq 5$ when the complexity is measured asymptotically in q. Although there sometimes exist cover attacks with even better complexity, the index calculus attack can be applied to trace zero varieties of all elliptic curves, while cover attacks apply only to a small proportion of curves.

Since the DLP in T_n has the same complexity as the DLP in $E(\mathbb{F}_{q^n})$, we get that the DLP in $E(\mathbb{F}_{q^n})$ may be attacked in complexity $\tilde{O}(q^{2-2/(n-1)})$ when E is defined over $\mathbb{F}_q$. This is better than all known direct attacks on the DLP in $E(\mathbb{F}_{q^n})$ for $n \geq 5$.

For general n, we have seen that the complexity of our index calculus attack on T_n depends exponentially on n and that it becomes infeasible for rather small values of n. This is due to the fact that the algorithm has to solve many polynomial systems, whose size (i.e. number of equations, number of indeterminates, degrees of the equations) depends on n, and that a Gröbner basis computation quickly becomes unmanageable. In fact, already for $n = 5$ we cannot solve the system with standard Gröbner basis software. By using some tricks (namely, considering relations that involve one point less, using a hybrid approach), we were nevertheless able to produce relations. However this does not yield a practical attack, since it multiplies the complexity of the relation search by a factor q^2.

Specialized Gröbner basis techniques in the spirit of [JV11, FPPR12, PQ12] would be needed in order to efficiently solve the systems that arise in this index calculus attack, and more research needs to be done on this topic in order to make our index calculus attack feasible in practice.

References

[AC07] R. M. Avanzi and E. Cesena, *Trace zero varieties over fields of characteristic 2 for cryptographic applications*, Proceedings of the First Symposium on Algebraic Geometry and Its Applications (SAGA '07), 2007, pp. 188–215.

[ACD+06] R. Avanzi, H. Cohen, C. Doche, G. Frey, T. Lange, K. Nguyen, and F. Vercauteren, *Handbook of elliptic and hyperelliptic curve cryptography*, Discrete Mathematics and its Applications, Chapman & Hall/CRC, Boca Raton, 2006.

[ADH94] L. M. Adleman, J. DeMarrais, and M.-D. A. Huang, *A subexponential algorithm for discrete logrithms over the rational subgroup of the Jacobians of large genus hyperelliptic curves over finite fields*, Algorithmic Number Theory (ANTS I) (L. M. Adleman and M.-D. A. Huang, eds.), LNCS, vol. 877, Springer, 1994, pp. 28–40.

[Adl79] L. M. Adleman, *A subexponential algorithm for discrete logarithms with applications to cryptography*, Proceedings of the 20th Annual Symposium on Foundations of Computer Science, IEEE, 1979, pp. 55–60.

[Adl94] ———, *The function field sieve*, Algorithmic Number Theory (ANTS I), LNCS, vol. 877, Springer, 1994, pp. 108–121.

[BBD+14] R. Bărbulescu, C. Bouvier, J. Detrey, P. Gaudry, H. Jeljeli, E. Thomé, M. Videau, and P. Zimmermann, *Discrete logarithm in GF(2^{809}) with FFS*, Public-Key Cryptography – PKC 2014 (H. Krawczyk, ed.), LNCS, vol. 8383, Springer, 2014, pp. 221–238.

[BCP97] W. Bosma, J. Cannon, and C. Playoust, *The Magma algebra system. I. The user language*, J. Symbolic Comput. **24** (1997), 235–265.

[BFP08] L. Bettale, J.-C. Faugère, and L. Perret, *Hybrid approach for solving multivariate systems over finite fields*, J. Math. Cryptol. **2** (2008), 1–22.

[BFSY05] M. Bardet, J.-C. Faugère, B. Salvy, and B.-Y. Yang, *Asymptotic behaviour of the index of regularity of quadratic semi-regular polynomial systems*, The Effective Methods in Algebraic Geometry Conference (MEGA '05) (P. Gianni, ed.), 2005, pp. 1–14.

[BGJT13] R. Bărbulescu, P. Gaudry, A. Joux, and E. Thomé, *A quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic*, To appear in Proceedings of EUROCRYPT '14, available at http://arxiv.org/abs/1306.4244, 2013.

[BKK+09] J. W. Bos, M. E. Kaihara, T. Kleinjung, A. K. Lenstra, and P. L. Montgomery, *Playstation 3 computing breaks 2^{60} barrier: 112-bit prime ECDLP solved*, Available at http://lacal.epfl.ch/112bit_prime, 2009.

[BLS11] D. J. Bernstein, T. Lange, and P. Schwabe, *On the correct use of the negation map in the Pollard rho method*, Public Key Cryptography – PKC 2011 (D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, eds.), LNCS, vol. 6571, Springer, 2011, pp. 128–146.

[Bou12] C. Bouvier, *The filtering step of discrete logarithm and integer factorization algorithms*, Available at http://hal.inria.fr/hal-00734654, 2012.

[Ces10] E. Cesena, *Trace zero varieties in pairing-based cryptography*, Ph.D. thesis, Università degli studi Roma Tre, Available at http://ricerca.mat.uniroma3.it/dottorato/Tesi/tesicesena.pdf, 2010.

[Cop84] D. Coppersmith, *Fast evalution of logarithms in fields of characteristic two*, IEEE Trans. Inform. Theory **30** (1984), 587–594.

[Die03] C. Diem, *The GHS attack in odd characteristic*, Ramanujan Math. Soc. **18** (2003), no. 1, 1–32.

[Die06] ———, *An index calculus algorithm for plane curves of small degree*, Algorithmic Number Theory (ANTS VII) (F. Hess, S. Pauli, and M. Pohst, eds.), LNCS, vol. 4076, Springer, 2006, pp. 543–557.

[Die11] ———, *On the discrete logarithm problem in elliptic curves*, Compos. Math. **147** (2011), 75–104.

[Die13] ———, *On the discrete logarithm problem in elliptic curves II*, Algebra & Number Theory **7** (2013), 1281–1323.

[DK13] C. Diem and S. Kochinke, *Computing discrete logarithms with special linear systems*, Available at http://www.math.uni-leipzig.de/~diem/preprints, 2013.

[DS] C. Diem and J. Scholten, *An attack on a trace-zero cryptosystem*, Available at http://www.math.uni-leipzig.de/diem/preprints.

[DS03] ———, *Cover attacks – A report for the AREHCC project*, Available at http://www.math.uni-leipzig.de/~diem/preprints, 2003.

[DT08] C. Diem and E. Thomé, *Index calculus in class groups of non-hyperelliptic curves of genus three*, J. Cryptology **21** (2008), 593–611.

[EG02] A. Enge and P. Gaudry, *A general framework for subexponential discrete logarithm algorithms*, Acta Arith. **102** (2002), 83–103.

[EG07] ———, *An $L(1/3+\varepsilon)$ algorithm for the discrete logarithm problem for low degree curves*, Advances in Cryptology: Proceedings of EUROCRYPT '07 (M. Naor, ed.), LNCS, vol. 4515, Springer, 2007, pp. 379–393.

[EGT11] A. Enge, P. Gaudry, and E. Thomé, *An $L(1/3)$ discrete logarithm algorithm for low degree curves*, J. Cryptology **24** (2011), 24–41.

[EK97] E. Eberly and K. Kaltofen, *On randomized Lanczos algorithms*, Proceedings of the 1997 international symposium on Symbolic and algebraic computation (ISSAC '97) (W. W. Küchlin, ed.), ACM, 1997, pp. 176–183.

[Eng02] A. Enge, *Computing discrete logarithms in high-genus hyperelliptic Jacobians in provably subexponential time*, Math. Comp. **71** (2002), 729–742.

[Eng08] ———, *Computing discrete logarithms in curves over finite fields*, Finite Fields Appl. (G. L. Mullen, D. Panario, and I. E. Shparlinski, eds.), Contemp. Math., vol. 461, AMS, 2008, pp. 119–139.

[Fau99] J.-C. Faugère, *A new efficient algorithm for computing Gröbner bases (F4)*, J. Pure Appl. Algebra **139** (1999), no. 1, 61–88.

[Fau02] ———, *A new efficient algorithm for computing Gröbner bases without reduction to zero (F5)*, Proceedings of the 2002 international symposium on Symbolic and algebraic computation (ISSAC '02), ACM, 2002, pp. 75–83.

[FGHR12] J.-C. Faugère, P. Gaudry, L. Huot, and G. Renault, *Using symmetries and fast change of ordering in the index calculus for elliptic curves discrete logarithm*, Proceedings of the Third International Conference on Symbolic Computation and Cryptography (SCC '12), 2012, pp. 113–118.

[FGHR13] ———, *Using symmetries in the index calculus for elliptic curves discrete logarithm*, To appear in J. Cryptology, Springer, DOI: 10.1007/s00145-013-9158-5, 2013.

[FGLM93] J.-C. Faugère, P. Gianni, D. Lazard, and T. Mora, *Efficient computation of zero-dimensional Gröbner bases by change of ordering*, J. Symbolic Comput. **16** (1993), no. 4, 329–344.

[FPPR12] J.-C. Faugère, L. Perret, C. Petit, and G. Renault, *Improving the complexity of index calculus algorithms in elliptic curves over binary fields*, Advances in Cryptology: Proceedings of EUROCRYPT '12 (D. Pointcheval and T. Johansson, eds.), LNCS, vol. 7237, Springer, 2012, pp. 27–44.

[Fre98] G. Frey, *How to disguise an elliptic curve*, Talk at the 2nd workshop on Elliptic Curve Cryptography (ECC '98), 1998.

[Fre99] ———, *Applications of arithmetical geometry to cryptographic constructions*, Proceedings of the 5th International Conference on Finite Fields and Applications, Springer, 1999, pp. 128–161.

[Gau00] P. Gaudry, *An algorithm for solving the discrete log problem on hyperelliptic curves*, Advances in Cryptology: Proceedings of EUROCRYPT '00 (B. Preneel, ed.), LNCS, vol. 1807, Springer, 2000, pp. 19–34.

[Gau09] ———, *Index calculus for abelian varieties of small dimension and the elliptic curve discrete logarithm problem*, J. Symbolic Comput. **44** (2009), no. 12, 1690–1702.

[GGMZ13a] F. Göloğlu, R. Granger, G. McGuire, and J. Zumbrägel, *On the function field sieve and the impact of higher splitting probabilities: Application to discrete logarithms in* $\mathbb{F}_{2^{1971}}$, To appear in Proceedings of CRYPTO '13, available at http://eprint.iacr.org/2013/074, 2013.

[GGMZ13b] ———, *Solving a 6120-bit DLP on a desktop computer*, To appear in Proceedings of SAC '13, available at http://eprint.iacr.org/2013/306, 2013.

[GHS02] P. Gaudry, F. Hess, and N.P. Smart, *Constructive and destructive facets of Weil descent*, J. Cryptology **15** (2002), no. 1, 19–46.

[GJV10] R. Granger, A. Joux, and V. Vitse, *New timings for oracle-assisted SDHP on the IPSEC Oakley 'well known group' 3 curve*, NMBRTHRY list, available at https://listserv.nodak.edu/cgi-bin/wa.exe?A2=ind1007&L=NMBRTHRY&P=R156&1=NMBRTHRY&9=A&J=on&d=No+Match%3BMatch, 2010.

[GLS11] S. D. Galbraith, X. Lin, and M. Scott, *Endomorphisms for faster elliptic curve cryptography on a large class of curves*, J. Cryptology **24** (2011), no. 3, 446–469.

[GLV01] R. P. Gallant, R. J. Lambert, and S. A. Vanstone, *Faster point multiplication on elliptic curves with efficient endomorphisms*, Advances in Cryptology: Proceedings of CRYPTO '01 (J. Kilian, ed.), LNCS, vol. 2139, Springer, 2001, pp. 190–200.

[GM13] E. Gorla and M. Massierer, *An optimal representation for the trace zero variety*, Preprint, 2013.

[GM14] ———, *Point compression for the trace zero subgroup over a small degree extension field*, To appear in Des. Codes Cryptogr., Springer, DOI: 10.1007/s10623-014-9921-0, 2014.

[Gor93] D. M. Gordon, *Discrete logarithms in GF(p) using the number field sieve*, SIAM J. Discrete Math. **6** (1993), no. 1, 124–138.

[Gor11] E. Gorla, *Torus-based cryptography*, Encyclopedia of Cryptography (S. Jajodia and H. v. Tilborg, eds.), Springer, Berlin–Heidelberg–New York, 2nd ed., 2011, pp. 1306–1308.

[GS99] S. D. Galbraith and N. P. Smart, *A cryptographic application of Weil descent*, Cryptography and Coding. Proceedings of the 7th IMA International Conference (M. Walker, ed.), LNCS, vol. 1746, Springer, 1999, pp. 191–200.

[GS06] S. D. Galbraith and B. A. Smith, *Discrete logarithms in generalized Jacobians*, Available at http://uk.arxiv.org/abs/math.NT/0610073, 2006.

[GTTD07] P. Gaudry, E. Thomé, N. Thériault, and C. Diem, *A double large prime variation for small genus hyperelliptic index calculus*, Math. Comp. **76** (2007), 475–492.

[GV05] R. Granger and F. Vercauteren, *On the discrete logarithm problem on algebraic tori*, Advances in Cryptology: Proceedings of CRYPTO '05 (V. Shoup, ed.), LNCS, vol. 3621, Springer, 2005, pp. 66–85.

[GvzG99] J. Gerhard and J. von zur Gathen, *Modern computer algebra*, Cambridge University Press, Cambridge, 1999.

[Jel13] H. Jeljeli, *Accelerating iterative SpMV for discrete logarithm problem using GPUs*, Available at http://hal.inria.fr/hal-00734975, 2013.

[Jel14] ———, *Resolution of linear algebra for the discrete logarithm problem using GPU and multi-core architectures*, Available at http://hal.inria.fr/hal-00946895, 2014.

[JL02] A. Joux and R. Lercier, *The function field sieve is quite special*, Algorithmic Number Theory (ANTS V) (C. Fieker and D. R. Kohel, eds.), LNCS, vol. 2369, Springer, 2002, pp. 431–445.

[JL03] ———, *Improvements to the general number field sieve for discrete logarithms in prime fields. A comparison with the Gaussian integer method*, Math. Comp. **72** (2003), no. 242, 953–976.

[JL06] ———, *The function field sieve in the medium prime case*, Advances in Cryptology: Proceedings of EUROCRYPT '06 (S. Vaudenay, ed.), LNCS, vol. 4004, Springer, 2006, pp. 254–270.

[Jou13a] A. Joux, *Faster index calculus for the medium prime case: Application to 1175-bit and 1425-bit finite fields*, Advances in Cryptology: Proceedings of EUROCRYPT '13 (T. Johansson and P. Nguyen, eds.), LNCS, vol. 7881, Springer, 2013, pp. 177–193.

[Jou13b] ———, *A new index calculus algorithm with complexity* $L(1/4 + o(1))$ *in very small characteristic*, To appear in Proceedings of SAC '13, available at http://eprint.iacr.org/2013/095, 2013.

[JV11] A. Joux and V. Vitse, *A variant of the F4 algorithm*, Topics in cryptology CT-RSA 2011, LNCS, vol. 6558, Springer, 2011, pp. 356–375.

[JV12] ———, *Elliptic curve discrete logarithm problem over small degree extension fields. Application to the static Diffie-Hellman problem on $E(\mathbb{F}_{q^5})$*, To appear in J. Cryptology, Springer, DOI: 10.1007/s00145-011-9116-z, 2012.

[Kob91] N. Koblitz, *CM-curves with good cryptographic properties*, Advances in Cryptology: Proceedings of CRYPTO '91 (J. Feigenbaum, ed.), LNCS, vol. 576, Springer, 1991, pp. 179–287.

[KR00] M. Kreuzer and L. Robbiano, *Computational commutative algebra 1*, Springer, Berlin–Heidelberg–New York, 2000.

[KR05] ———, *Computational commutative algebra 2*, Springer, Berlin–Heidelberg–New York, 2005.

[Lan01] T. Lange, *Efficient arithmetic on hyperelliptic curves*, Ph.D. thesis, Univerität GHS Essen, Available at http://www.hyperelliptic.org/tanja/preprints.html, 2001.

[Lan04] ———, *Trace zero subvarieties of genus 2 curves for cryptosystem*, Ramanujan Math. Soc. **19** (2004), no. 1, 15–33.

[LO90] B. A. LaMacchia and A. M. Odlyzko, *Solving large sparse linear systems over finite fields*, Advances in Cryptology: Proceedings of CRYPTO '90 (A. J. Menezes and S. A. Vanstone, eds.), LNCS, vol. 537, Springer, 1990, pp. 109–133.

[Nag10] K. Nagao, *Decomposition attack for the Jacobian of a hyperelliptic curve over an extension field*, Algorithmic Number Theory (ANTS IX) (G. Hanrot, F. Morain, and E. Thomé, eds.), LNCS, vol. 6197, Springer, 2010, pp. 285–300.

[PQ12] C. Petit and J. Quisquater, *On polynomial systems arising from a Weil descent*, Advances in Cryptology: Proceedings of ASIACRYPT '12 (X. Wang and K. Sako, eds.), LNCS, vol. 7658, Springer, 2012, pp. 451–466.

[RS02] K. Rubin and A. Silverberg, *Supersingular abelian varieties in cryptology*, Advances in Cryptology: Proceedings of CRYPTO '02 (M. Yung, ed.), LNCS, vol. 2442, Springer, 2002, pp. 336–353.

[RS09] ———, *Using abelian varieties to improve pairing-based cryptography*, J. Cryptology **22** (2009), no. 3, 330–364.

[Sch02] O. Schirokauer, *The special function field sieve*, SIAM J. Discrete Math. **16** (2002), 81–98.

[Sem04] I. Semaev, *Summation polynomials of the discrete logarithm problem on elliptic curves*, Available at http://eprint.iacr.org/2004/031, 2004.

[ST13] M. Shantz and E. Teske, *Solving the elliptic curve discrete logarithm problem using Semaev polynomials, Weil descent and Gröbner basis methods – an experimental study*, Available at http://eprint.iacr.org/2013/596, 2013.

[Thé03] N. Thériault, *Index calculus attack for hyperelliptic curves of small genus*, Advances in Cryptology: Proceedings of ASIACRYPT '03 (C. S. Laih, ed.), LNCS, vol. 2894, Springer, 2003, pp. 75–92.

[Tra88] C. Traverso, *Gröbner trace algorithms*, Proceedings of the 1988 international symposium on Symbolic and algebraic computation (ISSAC '88), LNCS, vol. 358, Springer, 1988, pp. 125–138.

[VJS14] M. D. Velichka, M. J. Jacobson, Jr., and A. Stein, *Computing discrete logarithms in the Jacobian of high-genus hyperelliptic curves over even characteristic finite fields*, Math. Comp. **83** (2014), no. 286, 935–963.

[Wie86] D. H. Wiedemann, *Solving sparse linear equations over finite fields*, IEEE Trans. Inform. Theory **IT-32** (1986), no. 1, 54–62.

[YCC04] B.-Y. Yang, J.-M. Chen, and N. T. Courtois, *On asymptotic security estimates in XL and Gröbner bases-related algebraic cryptanalysis*, Information and Communications Security (ICICS '04) (J. López, S. Qing, and E. Okamoto, eds.), LNCS, vol. 3269, Springer, 2004, pp. 401–413.

Elisa Gorla, Institut de mathématiques, Université de Neuchâtel, Rue Emile-Argand 11, 2000 Neuchâtel, Switzerland

E-mail address: elisa.gorla@unine.ch

Maike Massierer, Mathematisches Institut, Universität Basel, Rheinsprung 21, 4051 Basel, Switzerland

E-mail address: maike.massierer@unibas.ch