Computing Discrete Logarithms[⋆]

Robert Granger[1] and Antoine Joux[2]

[1] Surrey Centre for Cyber Security, University of Surrey, United Kingdom
[2] CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

1 Introduction

Let G be a multiplicatively-written finite cyclic group, let $g \in G$ be a generator and let $h \in G$. The discrete logarithm problem (DLP) for (G, g, h) is the computational problem of determining an integer x such that $h = g^x$. Note that the integer x is uniquely determined modulo the group order. Just as for the continuous logarithm function, one also writes $x = \log_g h$ and refers to x as the *discrete logarithm* of h to the base g.

The DLP has been central to public key cryptography ever since its inception by Diffie and Hellman in 1976 [15], and its study can be traced at least as far back as 1801, when discrete logarithms featured in Gauß' *Disquisitiones Arithmeticae*, referred to there as *indices* with respect to a primitive root modulo a prime [23, art. 57–60]. Indeed, the multiplicative group $\mathbb{F}_p^\times$ of the field $\mathbb{F}_p$ of integers modulo a prime p is perhaps the most natural example of a group in which the DLP can be posed – which is presumably why Diffie and Hellman used this setting for their famous key agreement protocol – and it is still believed to be hard for well-chosen primes.

In general, if the DLP is hard in a particular group then one can instantiate numerous important cryptographic protocols. So the issue at hand is: how hard is it to compute discrete logarithms in various groups? In this chapter we shall describe some cryptographically relevant DLPs and present some of the key ideas and constructions behind the most efficient algorithms known that solve them. Since the topic encompasses such a large volume of literature, for the finite field DLP we limit ourselves to a selection of results reflecting recent advances in fixed characteristic finite fields. We start by briefly recalling the so-called generic algorithms, which do not exploit any representational properties of group elements and may thus be applied to *any* finite cyclic group, and then recall the more sophisticated approach known as the *index calculus method*, which may be applied whenever the representation of elements of a group can be imbued with a suitable notion of smoothness. In §2 we introduce elliptic curves and pairings over finite fields and consider various discrete logarithm algorithms. Then in §3 we consider some groups in which the DLP is easier than for the strongest elliptic curves, including some families of weak curves. In §4 we focus on discrete logarithm algorithms for XTR and algebraic tori when defined over extension fields, and finally in §5 we present some of the key insights behind the breakthroughs between 2012 and 2014 that led to the downfall of finite fields of fixed characteristic in cryptography.

First, we introduce some useful notation for describing the running time of discrete logarithm algorithms (or equivalently the complexity or hardness of the DLP),

[⋆] This material will be published in revised form in Computational Cryptography edited by Joppe W. Bos and Martijn Stam and published by Cambridge University Press. See www.cambridge.org/9781108795937.

which has become customary. Let N be the order of a group G. We define

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

where $\alpha \in [0, 1]$, $c > 0$ and log denotes the natural logarithm. When there is no ambiguity we often omit the subscript N, and sometimes write $L(\alpha)$ to mean $L(\alpha, c)$ for some $c > 0$. Observe that $L(0) = (\log N)^{c+o(1)}$, which therefore represents polynomial time, while $L(1) = N^{c+o(1)}$ represents exponential time. If an algorithm has a running time of $L(\alpha)$ for some $0 < \alpha < 1$ it is said to be of *subexponential* complexity.

1.1 Generic algorithms

In the context of the DLP a generic algorithm is one that applies in the generic group model, in which elements have a randomly selected unique encoding and one can only perform group operations and check for equality of elements (by comparing the encodings), see [79]. In this context it was shown by Shoup [79] and Nechaev [67] that the DLP has an exponential running time $\Omega(\sqrt{N})$ if N is prime. This result implies that a subexponential algorithm *must* exploit a suitable group representation. We now describe the main examples of generic algorithms for the DLP in any finite cyclic group G, given a generator g and a target element h, where $|G| = N$.

First, if N is composite and its prime factorisation is known, then one can apply the Pohlig-Hellman algorithm [71], which reduces the DLP in G to DLPs in prime order subgroups. In particular, let $N = p_1^{e_1} \cdot \ldots \cdot p_r^{e_r}$. By the Chinese remainder theorem it is sufficient to solve the DLP in each of the subgroups of order $\mathbb{Z}/p_i^{e_i}\mathbb{Z}$ for $i = 1, \ldots, r$, and one can project the DLP into each of them by powering g and h by the respective cofactors $N/p_i^{e_i}$. Let $x_i = \log_g h \pmod{p_i^{e_i}}$. If $e_i = 1$ then one needs only to solve a DLP in a prime order subgroup. If $e_i > 1$ then the digits of the p_i-ary expansion of x_i can be computed sequentially, starting from the least significant digit via projecting and applying a Hensel lifting approach, each time solving a DLP in a subgroup of order p_i. For this reason, groups of large prime order, or those whose order possesses a large prime factor, are used in practice.

Moving on to algorithms for solving the DLP, we start with the time-memory trade-off known as the *Baby-Step-Giant-Step* method (BSGS), attributed to Shanks. Let $M = \lceil \sqrt{N} \rceil$. One first computes a table $\{(j, g^j) \mid j \in \{0 \ldots M-1\}\}$ (the baby steps) and sorts it according to the second component. Letting $k = g^{-M}$, one then computes $h, hk, hk^2, \ldots$ (the giant steps) until a collision $hk^i = g^j$ is found, at which point one knows that $\log_g h = iM + j$. The algorithm requires $O(\sqrt{N})$ storage and $O(\sqrt{N})$ group operations. Its precise bit complexity depends on the cost of group operations and on the implementation of the search for collisions.

An alternative approach is *Pollard's rho* method [72]. It requires some heuristic assumptions but preserves the expected $O(\sqrt{N})$ running time, while reducing the storage requirement to $O(1)$. The heuristic can be removed at the cost of introducing an extra logarithm factor in the runtime [37]. The core idea is to define pseudorandom sequences (a_i), (b_i) in $\mathbb{Z}/N\mathbb{Z}$ and $(x_i) \in G$ such that $x_i = g^{a_i}h^{b_i}$. To construct the sequence, we iterate a function $f : G \to G$ that allows the tracking of the exponent and behaves in a pseudo-random fashion. A typical choice is to partition G as a disjoint union $G = G_1 \cup G_2 \cup G_3$ and then define f by setting $f(x) = x^2$ when $x \in G_1$, $f(x) = gx$ when $x \in G_2$ and $f(x) = hx$ when $x \in G_3$.

Once f is defined, we construct the sequence (x_i) iteratively, starting from a random x_0 and computing $x_{i+1} = f(x_i)$. Eventually, since G is finite, one must have $x_j = x_j$ for some $i \neq j$. For such a collision one has $\log_g h = \frac{a_j - a_i}{b_i - b_j}$, provided the denominator is invertible modulo N. In fact, the sequence is ultimately periodic and one has $x_j = x_{j+\ell}$ for some $\ell, j_0 > 0$ and every $j \geq j_0$. In this context, one uses a cycle-finding algorithm to find a collision, for instance Floyd's cycle-finding algorithm which discovers a collision of the form $x_i = x_{2i}$.

Because of its ultimate periodicity, the sequence (x_i) has a tail and a cycle, depicted in Figure 1. This is why Pollard called it the 'ρ' method. For a random function f the expected length of the tail and the cycle is $\sqrt{\pi N/8}$ and therefore the expected time to solve the DLP is $\sqrt{\pi N/2}$. For concrete choices of f, a similar behavior is seen in practice and because of the assumption, the algorithm is heuristic. Due to the negligible storage requirements, the rho method is usually preferred over the BSGS method. For large computations with generic algorithms, the method of choice is often parallel collision search as introduced in [84].

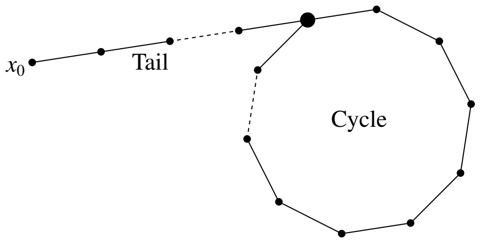

Fig. 1. Illustration of the shape of Pollard's rho sequences

1.2 The index calculus method

The index calculus method (ICM) – meaning, rather opaquely, a 'method for calculating the index' – is an approach to solving DLPs that can be far more efficient than generic methods, depending on the group and element representation under consideration, as well as the ingenuity of the mathematician. However, the basic template is the basis for all subexponential algorithms and so the use of the definite article is probably justified. The method was first published by Kraitchik in the 1920's in the context of the DLP in prime fields [52, 53], and has been independently discovered many times since, see [56, 62, 69] and the references therein.

We now describe the two stages of the ICM for (G, g, h) abstractly, i.e., without reference to a particular group. First, one must choose a subset $\mathcal{F} \subseteq G$ known as the *factor base*, such that $\langle \mathcal{F} \rangle = G$, and to which g is usually adjoined if it is not already in $\mathcal{F}$. Informally, the first stage of the ICM is finding the logarithms of all elements in $\mathcal{F}$; this stage is usually divided into two parts, namely, *relation generation* and *linear algebra*. The second stage is the individual logarithm stage, i.e., expressing an arbitrary element over $\mathcal{F}$ so as to infer its discrete logarithm.

More formally, let $A = \mathbb{Z}/N\mathbb{Z}$ and consider the surjective group homomorphism

$$\phi \colon A^{|\mathcal{F}|} \to G\,, \quad (e_f)_{f \in \mathcal{F}} \mapsto \prod_{f \in \mathcal{F}} f^{e_f}\,.$$

The aforementioned steps are as follows.

- **Relation generation:** Find vectors $(e_f)_{f\in\mathcal{F}}$ in $\ker\phi$, known as *relations*, which thus generate a subset $R \subseteq \ker\phi$.
- **Linear algebra:** Compute a non-zero element $(x_f)_{f\in\mathcal{F}} \in R^{\perp}$, i.e., one satisfying $\sum_{f\in\mathcal{F}} x_f e_f = 0$ for all $(e_f)_{f\in\mathcal{F}} \in R$. Taking the logarithm of the multiplicative relations, we see that the vector of logarithms of the elements of $\mathcal{F}$ (in any basis) form a solution. Assuming that the set of equations is large enough, one does not expect any other solutions.
- **Individual logarithm:** Find a preimage $(e_f)_{f\in\mathcal{F}} \in \phi^{-1}(h)$; it then follows that $\log_g h = \sum_{f\in\mathcal{F}} e_f \log_g f$.

Provided that sufficiently many linearly independent relations have been found, the discrete logarithms of elements of $\mathcal{F}$ can be computed, up to a non-zero scalar multiple, which can be normalised by insisting that $\log_g g = 1$.

In order to apply the ICM to a particular group and element representation, one needs to be able to define a suitable factor base. In order to do this one usually requires a natural notion of norm, primes and consequently smoothness, or the ability to impose analogues of these algebraically. An example of the former will be seen in §5, while an example of the latter will be seen in §4.

2 Elliptic curves

2.1 Elliptic curves of finite fields: a quick summary

An elliptic curve is a mathematical object which can be presented through several complementary points of view. When used for cryptographic purposes, the main focus is usually on elliptic curves over finite fields and one often focuses on the following definition.

Definition 1. *An* Elliptic curve *in Weierstrass form is a smooth projective curve given by an homogeneous equation:*

$$Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3.$$

When the coefficients $(a_1, a_2, a_3, a_4, a_6)$ belong to $\mathbb{F}_q$, we say that the curve is defined over $\mathbb{F}_q$.

Let p denote the characteristic of $\mathbb{F}_q$. As soon as $p \geq 5$, it is possible via a linear change of coordinates to change the equation into a reduced *Weierstrass equation:*

$$Y^2Z = X^3 + aXZ^2 + bZ^3.$$

Most of the time, we consider elliptic curves given by such a reduced equation. Let us briefly recall that the projective plane consists of all classes of non-zero triples (X, Y, Z) obtained from the equivalence relation that identifies (X, Y, Z) and (X', Y', Z') whenever there exists an invertible (i.e., non-zero in the case of $\mathbb{F}_q$) value λ such that $X = \lambda X'$, $Y = \lambda Y'$ and $Z = \lambda Z'$. The equivalence class associated to (X, Y, Z) is usually denoted by $(X : Y : Z)$.

A projective point with $Z = 0$ is said to lie at infinity. On the Weierstrass equation, we see that $Z = 0$ implies $X = 0$. As a consequence, there is a single point at infinity on the elliptic curve defined by that equation, the point with class $(0 : 1 : 0)$. It is simply called the point at infinity on E and written O_E. All other points have $Z \neq 0$; using the equivalence, they can be written as $(x : y : 1)$.

The pair (x, y) then satisfy the affine equation:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad \text{or} \quad y^2 = x^3 + ax + b.$$

It is a frequent practice to describe an elliptic curve by such an affine equation, together which the implicit convention that the curve also contains the point O_E at infinity.

The use of the projective form formalises the meaning of O_E, it can also by useful for faster computations. Indeed, it may avoid the need to compute inverses while adding points.

Main invariant. For the curve to be smooth, it should not have any singular points. This can be tested on a reduced Weierstrass equation by computing the discriminant:

$$\Delta = -16(4a^3 + 27b^2).$$

The curve is smooth, if and only if, $\Delta \neq 0$.

Moreover, the reduced Weierstrass form allows many distinct but isomorphic curve equations. To see that, it suffices to consider changes of variables of the form $(x, y) = (u^2x', u^3y')$. The change transforms the equation

$$y^2 = x^3 + ax + b$$

into

$$y'^2 = x'^3 + a'x' + b',$$

where $a' = a/u^4$ and $b' = b/u^6$.

The j-invariant of the curve is a simple way to classify isomorphic curves. It is given by:

$$j = -1728 \cdot \frac{64a^3}{\Delta}.$$

Two isomorphic curves have the same j-invariant, furthermore, over an algebraically closed field, two curves with the same j-invariant are isomorphic. However, over finite fields, the situation is more complex. Indeed, from the above formulae we see that $u^2 = a'b/ab'$ and we need to distinguish the case where u^2 is a quadratic residue or not, in the considered finite field.

In the first, u itself exists in the same field and the two curves are isomorphic. In the second, u belongs to a quadratic extension and the curves are said to be *quadratic twists*. Note that when $a = a' = 0$ or $b = b' = 0$, the situation is more complex since we can only compute u^4 or u^6 rather than u^2.

To distinguish quadratic twists over a finite field $\mathbb{F}_q$, one also computes the so-called minimal polynomial of Frobenius $X^2 - tX + q$. This is equivalent to point counting and the number of points defined over $\mathbb{F}_q$ is $q + 1 - t$ (including the point at infinity). Two isomorphic curves have the same number of points and when going from a curve to a quadratic twist, the parameter t changes its sign.

Since point counting can be done efficiently using the Schoof–Elkies–Atkin method (SEA), introduced by Schoof in [75], we can always assume that t is known. This makes elliptic curves quite useful for discrete logarithm based cryptosystem where it is essential to know the cardinality of the group being used.

Lines and the group law The main interest of elliptic curves, especially for cryptography, is that they can be equipped with a group law (denoted additively) whose neutral element is the point at infinity. This law can be defined geometrically as follows. Take an elliptic curve given by a reduced Weierstrass equation and consider the intersection of an arbitrary line with the curve. Three cases are to be considered: the line at infinity with equation $Z = 0$, a vertical line with equation $x = x_0$ and finally other lines with equation $y = \lambda x + \mu$.

In the first case, substituting Z by 0 in the projective equation of E, we find $x^3 = 0$. As a consequence, the intersection is reduced to the point O_E with multiplicity 3. For a vertical line, the affine equation becomes $y^2 = x_0^3 + ax_0 + b$. When the right-hand side is 0 we get as intersection a double point $(x_0, 0)$. When it is a quadratic residue, the intersection is formed of the two points $\left(x_0, \pm\sqrt{x_0^3 + ax_0 + b}\right)$. When it is a non-residue, the line doesn't meet the curve on affine points. However, by considering points over a quadratic field extension, we recover the two points of intersection. Furthermore, considering the projective equation of the same line, i.e., $X = x_0 Z$, we see that a vertical line also meets the curve at O_E.

Finally, for a line $y = \lambda x + \mu$, replacing y by this expression in the curve equation, we obtain:

$$x^3 - \lambda^2 x^2 + (a - 2\lambda\mu)\, x + (b - \mu^2) = 0.$$

Counting roots with multiplicities, this polynomial can have 0, 1 or 3 roots. Over a well-chosen field extension, we can always find three roots (counting multiplicities). Let x_1, x_2 and x_3 denote these (not necessarily distinct) roots. For each x_i, we get an intersection point $(x_i, \lambda x_i + \mu)$.

As a consequence, we see that any line intersects the curve E three times (counting multiplicities and intersection with O_E). The group law on E can be created from the simple rule that the sum of such three points of intersection is always O_E.

From the line at infinity, we find that $3O_E$ is zero, it is therefore natural to choose O_E as the neutral element for the group law. Then, considering a vertical line that meets the curve at (x_0, y_0) and $(x_0, -y_0)$. Since, $(x_0, y_0) + (x_0, -y_0) + O_E = O_E$, we see that points that are symmetric about the x-axis are opposites. We thus denote by $-P$ the reflection of P about the x-axis.

Finally, consider a line meeting the curve at P, Q and R, implying that $P + Q + R = O_E$. We can deduce that the sum $P + Q$ is equal to $-R$. From this, we recover the usual addition formulae on an elliptic curve.

More precisely, if P and Q are symmetric about the x-axis, their sum is O_E. Otherwise, we compute the slope λ of the line through P and Q. If the points are equal, the line is tangent to the curve and:

$$\lambda = \frac{2y_P}{3x_P^2 + a}.$$

If they are distinct, we have:

$$\lambda = \frac{y_P - y_Q}{x_P - x_Q}.$$

We then find that $x_{P+Q} = \lambda^2 - (x_P + x_Q)$ and that $y_{P+Q} = -(y_P + \lambda\,(x_{P+Q} - x_P))$.

Divisors, functions and pairings In truth, this idea of considering the intersection of the curve with a line is a particular case of a more general construction. To explain the more general viewpoint, we need to introduce two essential mathematical objects called divisors and functions of the curve.

Divisors of E. A divisor on the curve E is simply a mapping from the set of points on the curve to the integers of $\mathbb{Z}$ that is non-zero only on finitely many points. A frequent representation consists in listing the points where the mapping is non-zero together with the value at each of these points. With this representation a divisor D is written as a formal finite sum:

$$D = \sum_{i=1}^{n_D} e_i(P_i).$$

Given a divisor D in the above form, we define its degree as $\deg D = \sum_{i=1}^{n_D} e_i$. We also define the *support* of D as the set of points appearing with a non-zero coefficient in D. The set of divisors can be naturally given a group structure where the addition of two divisors is defined as the sum of the underlying mappings. When considering this group law, we can see that deg is a group homomorphism to $\mathbb{Z}$. Thus, its kernel called the set of degree-0 divisors is also a group.

Functions and function field of a curve. The concept of functions on a curve will generalise the idea of a line. Given an elliptic curve E and its (reduced) Weierstrass equation, we proceed in two steps. First, we consider the ring of bivariate polynomials in X and Y modulo the curve equation $Y^2 - (X^3 + a\,X + b)$. For example, in that ring, Y^2 and $X^3 + aX + b$ are representations of the same element. This ring is an integral domain and we can then build its field of fractions. This field is called the *function field* of the curve. Informally, we are thus considering rational fractions in X and Y modulo the curve's equation.

Let f/g be a fraction representing an element of the function field. Then, for any point P of E, we can compute the value $f(P)$ by replacing in f the variables X and Y by the values of the coordinates of P. In the same way, we can compute $g(P)$. Finally, when $g(P)$ is non-zero, the evaluation of f/g at P is defined as $f(P)/g(P)$. Moreover, if f_1/g_1 and f_2/g_2 represent the same element of the function field and if, in addition, $g_1(P) \neq 0$ and $g_2(P) \neq 0$, then $(f_1/g_1)(P) = (f_2/g_2)(P)$. Indeed, when f_1/g_1 and f_2/g_2 represent the same function, then $f_1g_2 - f_2g_1$ is a multiple of the curve equation. Thus, the value of $f_1g_2 - f_2g_1$ at P is 0, which implies the equality of evaluations of f_1/g_1 and f_2/g_2 at P. Furthermore, this allows us to define f/g at every point P on E. If $g(P) \neq 0$, we use the value $f(P)/g(P)$ as before. Otherwise, when $g(P) = 0$, we consider $f(P)$. More precisely, if $g(P) = 0$ and $f(P) \neq 0$ we say that $(f/g)(P) = \infty$. In the final case where both $g(P) = 0$ and $f(P) = 0$, it is always possible to find F and G such that $f/g = F/G$ and $(F(P), G(P)) \neq (0, 0)$. We can then use F/G to define the value at P. This definition is valid since (when defined) the value is independent of the choice of representative of the function.

When $(f/g)(P) = 0$, we say that P is a *zero* of f/g and when $(f/g)(P) = \infty$, we say that P is a *pole* of f/g. It is also possible to define multiplicities of zeroes and poles.

Evaluation of a function at a point can easily be generalised into an evaluation on a divisor in the following way. If D is given as $D = \sum_i e_i(P_i)$ and F is an element of the function field, with no zero or pole in the support of D, we define $F(D)$ by the following formula;

$$F(D) = \prod_i F(P_i)^{e_i},$$

i.e., as the product (with multiplicities) of evaluations on all points in the support of D.

Principal divisors and the group law. To every non zero function F in the function field, we can associate a divisor $\mathrm{Div}(F)$ that regroups the information on its zeroes and poles. In the following equation that defines $\mathrm{Div}(F)$, the notation $\mathcal{Z}_F$ stands for the set of its zeroes and $\mathcal{P}_F$ for the set of poles. Furthermore, when P is a zero or a pole, m_P denotes its multiplicity. We now define the divisor of F as:

$$\mathrm{Div}(F) = \sum_{P \in \mathcal{Z}_F} m_P(P) - \sum_{P \in \mathcal{P}_F} m_P(P).$$

Note that for all $\lambda \in \mathbb{F}_p^\times$ we have $\mathrm{Div}(\lambda F) = \mathrm{Div}(F)$. Indeed, multiplication by a non-zero constant does not change the zeroes or poles of a function. Furthermore, $\mathrm{Div}(1)$ is the zero divisor (empty in our notation), $\mathrm{Div}(FG) = \mathrm{Div}(F) + \mathrm{Div}(G)$ and $\mathrm{Div}(1/F) = -\mathrm{Div}(F)$. Thus, the Div operator is a morphism for the multiplicative group of the function field to the additive group of divisors. Divisors in the image of Div are called *principal divisors*. In addition, it can be shown that the degree of a principal divisor is zero.

As a consequence, principal divisors form a subgroup of degree-0 divisors and we can form the quotient group. This yields back the group law for the elliptic curve. The most relevant property is that for every degree-0 divisor D, there exists a unique point P and function F (up to equivalence) such that D can be written as $(P)-(O_E)+\mathrm{Div}(F)$. It is thus possible to label the elements of the above quotient group by points of E, which equips E with the group law.

Conversely, this correspondence can be used to test whether a given divisor is principal. We first check that the degree is 0 then evaluate the expression given the divisor on the curve. More precisely, for $D = \sum_i e_i(P_i)$, we compute $v(D) = \sum_i e_i P_i$. A degree-0 divisor D is principal, if and only if, $v(D)$ is the point at infinity O_E.

Furthermore, if D is a principal divisor, there exists a unique (up to multiplication by a constant) function F such that $\mathrm{Div}(F) = D$.

Weil's pairing. The Weil pairing is a non-degenerate, antisymmetric, bilinear function from the n-torsion subgroup $E[n]$ of the curve E to the n-th roots of unity in $\overline{\mathbb{F}}_p$. We recall that the n-torsion $E[n]$ is the set of all points P defined over the algebraic closure $\overline{\mathbb{F}}_p$ such that $nP = O_E$. When n is coprime to p, we know that $E[n]$ is isomorphic to the group $(\mathbb{Z}/n\mathbb{Z})^2$.

Let P and Q be two n-torsion points. We denote by D_P the divisor $D_P = (P)-(O_E)$ and by D_Q the divisor $(Q) - (O_E)$. We see that nD_P and nD_Q are principal. Let F_P and F_Q be functions such that $\mathrm{Div}(F_P) = nD_P$ and $\mathrm{Div}(F_Q) = nD_Q$. The Weil pairing $e_n(P, Q)$ is then defined as $F_P(D_Q)/F_Q(D_P)$.

When P and Q are defined over a small degree extension of $\mathbb{F}_p$, there exists an efficient algorithm due to Miller [66] that quickly computes this value $e_n(P, Q)$. Due to this efficiency, the Weil pairing (and its cousin the Tate pairing [18]) have been used to construct a large variety of, so-called, pairing-based cryptographic protocols.

The main parameter that governs the concrete use of a pairing is the *embedding degree*. Given a curve E with cardinality N_E over $\mathbb{F}_p$ and q a prime divisor of N_E, there is a cyclic subgroup of order q defined over $\mathbb{F}_p$. The embedding degree of this subgroup is the smallest integer k such that $\mathbb{F}_{p^k}$ contains a primitive q-th root of unity. It gives the smallest field $\mathbb{F}_{p^k}$ in which one can compute a non-degenerate pairing involving points of order q.

2.2 Discrete logarithm algorithms for families of weak curves

Supersingular and low-embedding degree curves The pairings we just described can be used as a tool to transport the discrete logarithm problem on an elliptic curve to a discrete logarithm problem in a finite field. This is the MOV method, introduced by Menezes, Okamoto and Vanstone [63]. It initially used the Weil pairing but can also rely on the Tate pairing [18].

The idea, in order to solve $Q = nP$ in a cyclic group of order q is to find a third point R, also of order q such that $e_q(P, R) \neq 1$. Then, by linearity of the pairing, we have $e_q(Q, R) = e_q(P, R)^n$. This moves the discrete logarithm to the group of q-th roots of unity in the finite field $\mathbb{F}_{p^k}$, where k is the embedding degree.

Due to the sub-exponential algorithms for computing discrete logarithms in $\mathbb{F}_{p^k}^{\times}$ (discussed briefly in §4 when p is of cryptographic size), this gives a better than generic algorithm as long as k remains small.

An especially weak case for which the MOV method was initially proposed is the case of supersingular curves that have cardinality $p + 1$ and embedding degree 2 since $p + 1$ divides $p^2 - 1$.

Descent and cover methods The discrete logarithm techniques that follow only apply (to this day) to curves defined over a finite field $\mathbb{F}_{p^k}$, where the extension degree k has small factors. In particular, we currently do not know how to use them for curves defined over prime fields $\mathbb{F}_p$. Similarly, after studying state-of-the-art methods together with various speed-ups, [19] concludes that logarithms on curves over prime degree extension fields cannot be computed faster than by generic methods for cryptographic sizes.

Gaudry-Hess-Smart (GHS) method. This method, described in [22, 64, 20, 36], consists of finding a so-called cover of the curve E defined over $\mathbb{F}_{p^k}$ by a curve H of genus $g > 1$ defined over the smaller field $\mathbb{F}_p$. More precisely, a cover is a surjective map from H to E expressed by rational functions on H. It is particularly useful when the genus g is not too large, ideally when $g = k$. The existence of the cover can be used to transport the discrete logarithm problem from E to the Jacobian of H, where it becomes easier. It turns that the conditions permitting the existence of a cover are such that the most studied and more vulnerable cases are for $k = 2$ and $k = 3$.

Gaudry-Semaev method. This method developed in [77, 21] is an index calculus technique that remains in the curve E (the general method of [21] is summarised in §4.2). Its basic idea is to write arbitrary points on the curve as a sum of a small number of points with abcissa in the small field $\mathbb{F}_p$. This is done by solving multivariate systems of polynomial equations in k variables, with a degree growing exponentially with k. Again, this is only achievable for small values of k.

Combining both, Joux-Vitse method. For certain curves with $k = 6$, the situation is especially bad. As shown in [45], it is possible to combine both attacks. First moving the discrete logarithm to the Jacobian of a genus 3 curve of $\mathbb{F}_{p^2}$, where a variant of Gaudry-Semaev method can be used. Discrete logarithms can then be computed for these specific curves, even for cryptographically meaningful group sizes.

Diem's asymptotic analysis. In two articles [13, 14], Diem showed that elliptic curve discrete logarithms over finite fields $\mathbb{F}_{q^n}$ can be solved asymptotically faster than by generic methods. In [13], he achieves an asymptotic complexity of the form $\exp(O(\max(\log q, n^2)))$ and in [14], he improves it to $\exp(O(\max(\log q, n\log(q)^{1/2}, n^{3/2})))$. As far as we know, Diem's methods have not been used in any large size computation. The main obstruction seems to be the need to solve algebraic systems of equations of large degrees and number of variables, which are not practically accessible with current algebraic methods.

3 Some group descriptions with easier discrete logarithms

3.1 Addition modulo an integer

A basic theorem in group theory is that every cyclic group of order N is isomorphic to its structure group $(\mathbb{Z}/N\mathbb{Z}, +)$. Moreover, in this group representation, solving the discrete logarithm problem is trivial. Indeed, since the group law is additive, the discrete logarithm problem is, given a generator of the group, i.e., a number x coprime with N and a value y to find n such that $y = nx \pmod{N}$. This implies $n = yx^{-1} \pmod{N}$, where x^{-1} is obtained from Euclid's extended GCD algorithm.

A classical question that arises when presenting discrete logarithm based cryptography to pure mathematicians is related to the remark. Why should the discrete logarithm problem be considered to be hard since it is so easy in the (isomorphic) structure group? In fact, computing discrete logarithms is simply a way to explicitly describe the isomorphism between a cyclic group and the corresponding $(\mathbb{Z}/N\mathbb{Z}, +)$. Furthermore, solving the problem not only requires an explicit expression but also an efficiently computable one.

3.2 Matrix groups

Since the discrete logarithm problem can be defined for any cyclic group, it is quite natural to consider the subgroup of the square matrices over some finite field $\mathbb{F}_p$ generated by an invertible matrix G. In particular, this was suggested in [70].

This problem was studied by Menezes and Wu in [65]. It turns out that it can be reduced to discrete logarithms in finite fields. Furthermore, the matrix computations are more expensive for the participants than the corresponding computations in finite fields. As a consequence, this particular instantiation of the discrete logarithm problem does not provide any specific advantage.

We briefly describe here the main idea of the construction. Let A and B be two n by n matrices over $\mathbb{F}_p$. We want to find ℓ such that $B = A^\ell$ knowing that such an integer exists.

We first consider the characteristic polynomial p_A of A. In general, the complete form of the attack depends on the factorisation of p_A. However, the attack is easier to describe when p_A is an irreducible polynomial of degree n. For simplicity of exposition, we limit ourselves to that case. In that case, p_A has n distinct conjuguate roots in $\mathbb{F}_{p^n}$ and these roots are eigenvalues of A. Let α denote one of these eigenvalues, the others can be written as α^{p^i} with i in $[1 \dots n-1]$.

As a consequence, A can be diagonalised by writting $A = C^{-1}A_D C$ where A_D is a diagonal matrix whose entries are the values α^{p^i} with i in $[0 \dots n-1]$. We see that $B = C^{-1}A_D^{\ell}C$, thus $B_D = CBC^{-1}$ is diagonal with entries $\alpha^{\ell\, p^i}$. Taking the logarithm of the first entry α^{ℓ} relative to α in $\mathbb{F}_{p^n}$ is thus enough to completely recover ℓ.

3.3 Particularly bad curves

Singular curves A first example of bad curves covers a degenerate case, the case of curves with a zero discriminant. These curves have a singular point and are not valid elliptic curves. However, they can be obtained by reducing modulo a prime p an elliptic curve E with rational coefficients. In that case, we say that E has bad reduction at p.

In this situation, there is a singular point on the curve and we denote by E^{ns} the set of regular points. It is interesting to know that on this set, the usual geometric construction of an elliptic curve group law still works and yields a group law on E^{ns} .

We can distinguish two main cases. In the first one, the curve E is given by $y^2 = x^3$ (possibly after a change of variable) in the second by $y^2 = x^3 + Ax^2$ with $A \neq 0$. In both cases, the point $(0, 0)$ is singular on E and the set E^{ns} consists of all the other points.

In the first case, i.e., on $y^2 = x^3$, every point of E^{ns} can be written as (ℓ^2, ℓ^3), with $\ell \neq 0$. Moreover, given $P = (x_P, y_P)$ the corresponding value is simply given by $\ell_P = y_p/x_P$. Let P and Q be the two points corresponding to the values ℓ_P and ℓ_Q. The slope of the line through P and Q is:

$$\lambda = \frac{y_Q - y_P}{x_Q - x_P} = \frac{\ell_Q^2 + \ell_P\ell_Q + \ell_P^2}{\ell_Q + \ell_P}.$$

Let $R = (x_R, y_R)$ be the third point of intersection with E. We find that $x_R = \lambda^2 - (x_P + x_Q)$ and $y_R = \lambda(x_R - x_P) + y_p$. Developing the expressions we can write:

$$x_R = \frac{\ell_P^2\ell_Q^2}{\ell_Q^2 + 2\,\ell_P\ell_Q + \ell_P^2} = \left(\frac{\ell_P\ell_Q}{\ell_P + \ell_Q}\right)^2,$$

and

$$y_R = -\left(\frac{\ell_P\ell_Q}{\ell_P + \ell_Q}\right)^3.$$

Thus, the sum of P and Q corresponds to the values $\ell_{P+Q} = \frac{\ell_P\ell_Q}{\ell_P+\ell_Q}$. We can remark that:

$$\frac{1}{\ell_{P+Q}} = \frac{1}{\ell_P} + \frac{1}{\ell_Q}.$$

Adding the natural convention that the point at infinity has an infinite associated value (with inverse 0), we see that addition on E^{ns} boils down to addition in $\mathbb{F}_p$. The group isomorphism to the structure group is explicit and efficient to compute and the discrete logarithm is thus easy.

For the second case, i.e. the curve $y^2 = x^3 + Ax^2$, we start by writing $A = \alpha^2$. This leads to two subcases depending on whether A is a square in $\mathbb{F}_p$ or not. In the first case, we turn point addition into multiplication in $\mathbb{F}_p^\times$, while in the second it becomes multiplication in the subgroup of order $p + 1$ of the quadratic extension $\mathbb{F}_{p^2}^\times$.

As before, we express the points of the curve E^{ns} as functions of a parameter ℓ_P given by the following formula:

$$\ell_P = \frac{y_P + \alpha\, x_P}{y_P - \alpha\, x_P},$$

with the convention that it is equal to 1 for the point at infinity. The coordinates x_P and y_P can be recovered from ℓ_P by computing:

$$x_P = \frac{4\alpha^2 \ell_p}{(\ell_P - 1)^2} \quad \text{and} \quad y_P = \frac{4\alpha^3 \ell_p(\ell_P + 1)}{(\ell_P - 1)^3}.$$

Finally, we can check that $\ell_{P+Q} = \ell_P \ell_Q$. Thus, the discrete logarithm is transported to the multiplicative group of a finite field $\mathbb{F}_p$ or $\mathbb{F}_{p^2}$. As a consequence, the discrete logarithm becomes much easier than on (general) elliptic curves.

Anomalous curves An elliptic curve over a prime field $\mathbb{F}_p$ is said to be anomalous when its trace is equal to 1 or equivalently its cardinality is equal to p. On such curves, the discrete logarithm problem becomes easy. In fact, two distinct methods have been proposed to explain this fact. The first one by Semaev [76] defines an additive pairing sending a point P to an element of $\mathbb{F}_p$ by defining the function f_P with divisor $p(P) - p(O)$ (which is unique up to a multiplicative constant) and evaluating the ratio of the function f_P and its x-derivative at another (fixed) point.

The second approach by Smart [80] considers an arbitrary lift of the curve and the point P to the p-adic numbers and uses a multiplication by p and a p-adic elliptic logarithm. All of these computations can be done with low p-adic precision.

For simplicity, we present here a heuristic version of the second method that bypasses the use of p-adic elliptic logarithms. As usual given the curve E defined modulo p and two non-zero points P and Q we want to solve the equation $Q = nP \pmod{p}$. Indeed, since p is prime, any non-zero point is a generator of E.

We start by considering an arbitrary Weirstrass equation modulo p^2 that reduces to the equation of E modulo p. We denote by E_2 this lifted curve modulo p^2. Each point of E, including the point at infinity, can be lifted to E_2 in p distinct ways via Hensel's Lemma. Thus E_2 contains p^2 points with coordinates modulo p^2. Furthermore, the usual group law contruction can be applied to E_2. We thus get an abelian group with order p^2. Its structure is either $(\mathbb{Z}/p\mathbb{Z})^2$ or $\mathbb{Z}/p^2\mathbb{Z}$. Heuristically, for a random lifting, we expect $\mathbb{Z}/p^2\mathbb{Z}$ to occur more frequently.

Let us now assume that E_2 with the elliptic curve addition is a cyclic group of order p^2. The reduction modulo p of points is a surjective group homomorphism to E, whose kernel is formed of the p-distinct lifting of the point at infinity. We denote by r_E this reduction. The kernel of r_E is thus a subgroup of order p.

Let P_2 and Q_2 be two arbitrary liftings of P and Q. This implies that $r_E(Q_2 - nP_2) = Q - nP = 0$. Thus $Q_2 - nP_2$ is a lift of the point at infinity O_E. As a consequence, $p(Q_2 - nP_2)$ is zero on E_2. Thus, $(pQ_2) = n(pP_2)$. In addition, pP_2 and pQ_2 are in the kernel of r_E.

To see how this leads to a recovery of n, let us study the structure of the kernel of r_E, i.e., of the subgroup formed by all lifts of the point at infinity. To do this, it is useful to consider a weighted projective description of the Weirstrass equation, where the variables X, Y and Z respectively have weights 2, 3 and 1. The Weierstrass equation then has total weight 6 and can be written as:

$$Y^2 + a_1XYZ + a_3YZ^3 = X^3 + a_2X^2Z^2 + a_4XZ^4 + a_6Z^6.$$

Note that each a_i corresponds here to the Z^i term.

In this weighted notation, (X, Y, Z) and (X', Y', Z') represent the same point if and only if there exists an invertible element λ modulo p^2 such that $X' = \lambda^2 X$, $Y' = \lambda^3 Y$ and $Z' = \lambda Z$.

The liftings of the point at infinity are the triple where Z is a multiple of p. Up to equivalence, they can all be written as $(1, 1, \ell p)$. The point $(1, 1, 0)$ is the zero of the group law. For all the other points of this form, we can put them into the equivalent form $(\ell^{-2}, \ell^{-3}, p)$.

If we further assume that $a_1 = 0$, which can be achieved by using a reduced Weierstrass form for E_2, the situation is equivalent to considering the point (ℓ^{-2}, ℓ^{-3}) on the singular curve of equation $y^2 = x^3$. As we saw previously, the logarithm can be obtained by just mapping this point to the slope, we previously called ℓ, in the finite field.

4 Discrete logarithms for XTR and algebraic tori

The Diffie-Hellman key exchange protocol [15] and El Gamal encryption and signatures [16] were formulated in the multiplicative group of a prime field $\mathbb{F}_p$. While the fastest algorithms for solving the DLP in $\mathbb{F}_p^\times$ are subexponential (for details we refer the reader to the survey article [31]), given a subgroup of prime order l, the fastest known discrete logarithm algorithms that operate purely within the subgroup are generic. As Schnorr observed, one can therefore base protocols in the subgroup in order to speed up exponentiations, and obtain shorter signatures for example, provided that the complexity of both attacks is above the required security threshold [74].

However, other than by using the discrete logarithm of a subgroup element relative to a generator, for prime fields there does not seem to be a way to reduce the size of the representation of elements: each requires $\lceil \log_2 p \rceil$ bits. One way to overcome this representational (and operational) inefficiency is to instead consider subgroups of the multiplicative group of extension fields, i.e., $\mathbb{F}_{p^n}^\times$. This idea is the basis of the entirety of Chapter 10, so presently we only briefly mention a couple of important examples.

The cryptosystem LUC [81], developed by Smith and Skinner in 1995, represents elements of the order $p + 1$ subgroup of $\mathbb{F}_{p^2}^\times$ by their trace from $\mathbb{F}_{p^2}$ to $\mathbb{F}_p$. As a result, just one element of $\mathbb{F}_p$ is needed to represent an element, thus providing the optimal compression factor of 2 for the full subgroup. Building upon this idea, in 1999, Brouwer, Pellikaan and Verheul described a compression method for elements of the order $p^2 - p + 1$ subgroup of $\mathbb{F}_{p^6}^\times$, again using the trace function, but this time to $\mathbb{F}_{p^2}$ [9]. This reduces the representation to just two elements of $\mathbb{F}_p$, providing the optimal compression factor of 3 for the full subgroup. Very soon afterwards, Lenstra and Verheul developed the cryptosystem XTR[3], extending the compression method in [9] by developing a more efficient exponentiation for the trace representation than is available in the usual field representation [59, 57].

Observe that for both LUC and XTR, the relevant subgroups do not embed into a proper subfield of $\mathbb{F}_{p^n}$, for $n = 2$ and $n = 6$ respectively. Indeed, they are the so-called cyclotomic subgroups of $\mathbb{F}_{p^n}^{\times}$, which have order $\Phi_n(p)$ where $\Phi_n(x)$ is the n-th cyclotomic polynomial, defined by

$$\Phi_n(x) := \prod_{1 \leq k \leq n,\ \gcd(k,n)=1} (x - \zeta_n^k),$$

and ζ_n is a primitive (complex) n-th root of unity. Note that the degree of $\Phi_n(x)$ is simply $\phi(n)$, where $\phi(\cdot)$ is the Euler totient function. Furthermore, since

$$x^n - 1 = \prod_{d|n} \Phi_d(x), \tag{1}$$

for each $d|n$ the subgroup of $\mathbb{F}_{p^n}^{\times}$ of order $\Phi_d(p)$ embeds into $\mathbb{F}_{p^d}$. Hence, for $d < n$ one can solve the DLP in the order $\Phi_d(p)$ subgroup of $\mathbb{F}_{p^n}^{\times}$ by applying subexponential algorithms to $\mathbb{F}_{p^d}^{\times}$, rather than to $\mathbb{F}_{p^n}^{\times}$. So the subgroup of order $\Phi_n(p)$ may be regarded as the 'cryptographically strongest' subgroup of $\mathbb{F}_{p^n}^{\times}$, and this subgroup is always used in cryptographic applications[4].

Interestingly, the first listed author of this chapter was informed in 2005 by the second listed editor of this book that the real purpose of XTR was to stimulate research into the DLP in finite fields of small extension degree, and possibly in the cyclotomic subgroups [82], which was confirmed by Lenstra when the conversation was raised in 2013 [55]. For the former possibility, generally referred to as the medium prime case, research has progressed steadily, with several $L_{p^n}(1/3, c)$ algorithms being developed with generally decreasing c [4, 3, 46, 1]. These algorithms are mainly motivated by pairing-based cryptography. On the other hand, the latter possibility would seem at first to not be possible, thanks to the argument of the previous paragraph. Let $G_{p,n}$ denote the subgroup of $\mathbb{F}_{p^n}^{\times}$ of order $\Phi_n(p)$. If there were a hypothetical non-generic algorithm for solving the DLP in $G_{p,n}$ which was more efficient than solving the DLP via the embedding into $\mathbb{F}_{p^n}$, then by solving it there and also in $\mathbb{F}_{p^d}$ for each $d \mid n$, $d < n$, by (1) and the Chinese remainder theorem[5], one would have solved the DLP in $\mathbb{F}_{p^n}$ more efficiently than was thought possible. It has been argued that with such a security reduction one can be confident in the DLP security of $G_{p,n}$, as it is equivalent to the DLP security of $\mathbb{F}_{p^n}$, which is well studied.

However, this reduction can be viewed in another way: to attack the DLP in $\mathbb{F}_{p^n}$ one can try to invent algorithms for attacking the DLP $G_{p,n}$ directly. Indeed, this is what Granger and Vercauteren did in 2005 [28], as we explain shortly.

[3] XTR, pronounced '*X-T-R*', is the phonetic pronunciation of the acronym ECSTR, which stands for Efficient Compact Subgroup Trace Representation.

[4] Only if $\Phi_n(p) \leq n$, which is never the case for cryptographic applications, may this subgroup embed into a proper subfield of $\mathbb{F}_{p^n}$ [58].

[5] One may ignore the cryptographically small GCD's of such factors, as they can be computed with generic methods.

4.1 Algebraic tori, rationality and compression

In their original paper Lenstra and Verheul proposed to allow the base field for XTR to itself be an extension field, so henceforth we allow the base field to be $\mathbb{F}_q$ where q is a prime power p^m with $m \geq 1$.

In 2003 Rubin and Silverberg proposed torus-based cryptography, based on the observation that $G_{q,n}$ can be identified with the $\mathbb{F}_q$-rational points on the algebraic torus T_n of dimension $\phi(n)$, which has some cryptographically exploitable properties [73]. As well as showing that one can interpret LUC and XTR in terms of quotients of the algebraic tori T_2 and T_6 by certain actions of the symmetric groups S_2 and S_3 respectively, they observed that whenever T_n is rational, i.e., there exists a rational map to $\phi(n)$-dimensional affine space, one can compress (almost all of) its elements by a factor of $n/\phi(n)$ relative to the $\mathbb{F}_{q^n}$ representation and use this smaller representation for communications. T_n is known to be rational if n is the product of at most two prime powers, and is conjectured to be rational for all n [85, 51], although no other examples are currently known. Were this conjecture to be proven then one could obtain arbitrarily large compression factors for elements of the cyclotomic subgroup of $\mathbb{F}_{q^n}^{\times}$. The rationality of T_2 gives a simple analogue to LUC, while the rationality of T_6 gives an analogue to XTR, with an advantage that in these analogues one can freely multiply elements, unlike in LUC and XTR.

We now formally define the algebraic torus.

Definition 2. *Let $k = \mathbb{F}_q$ and $L = \mathbb{F}_{q^n}$. The torus T_n is the intersection of the kernels of the norm maps $N_{L/F}$, for all subfields $k \subset F \subsetneq L$:*

$$T_n(k) := \bigcap_{k \subset F \subsetneq L} Ker[N_{L/F}].$$

The following lemma provides two useful properties of T_n [73].

Lemma 1. *1. $T_n(\mathbb{F}_q) \cong G_{q,n}$, and thus $\#T_n(\mathbb{F}_q) = \Phi_n(q)$;*

2. If $h \in T_n(\mathbb{F}_q)$ is an element of prime order not dividing n, then h does not lie in a proper subfield of $\mathbb{F}_{q^n}/\mathbb{F}_q$.

4.2 Gaudry's algorithm

The attack of Granger and Vercauteren [28] may be seen as a version of an algorithm due to Gaudry, which is a general index calculus algorithm that may be applied to any abelian variety once a computationally convenient element representation and group law have been specified [21]. We briefly recall it here. Let $A/\mathbb{F}_q$ be an abelian variety of dimension d on which we would like to solve the DLP. We assume that except for a negligible proportion of elements, there is an explicit embedding of A into affine space of dimension $d + d'$, i.e., an element $P \in A$ defined over $\mathbb{F}_q$ can be represented as

$$P = (x_1, \ldots, x_d, y_1, \ldots, y_{d'}),$$

where $x_i, y_i \in \mathbb{F}_q$. Since A has dimension d, we assume that for any $x_1, \ldots, x_d \in \overline{\mathbb{F}_q}$ there are only finitely many $y_1, \ldots, y_{d'} \in \overline{\mathbb{F}_q}$ such that the corresponding P is on A. For the factor base, or more appropriately the decomposition base, let

$$\mathcal{F} := \{(x_1, 0, \ldots, 0, y_1, \ldots, y_{d'}) \in A : x_1, y_i \in \mathbb{F}_q\},$$

which one may assume is an absolutely irreducible curve whose closure under the group law is not a strict abelian subvariety of A; for otherwise, a random linear change of variables can be applied to the x_i-coordinates until these two properties hold. Hence, one may assume that $|\mathcal{F}| \approx q$.

Let $P \in A$ and let $Q \in \langle P \rangle$, with the group operation written additively. In order to find $\log_P Q$ we construct linear combinations $R = aP + bQ$ with a, b uniformly random integers modulo the group order and attempt to express R as a sum of d elements of $\mathcal{F}$, i.e.,

$$R = aP + bQ = P_1 + \cdots + P_d, \tag{2}$$

where $P_i \in \mathcal{F}$, since this will heuristically occur with probability $\approx 1/d!$ as $q \to \infty$. When this occurs, we call (2) a relation. One can then proceed with the usual index calculus method. The crux of this method is how to test whether a random element of A decomposes over $\mathcal{F}$. Since A is an abelian variety and therefore an algebraic group, one can express the right hand side of (2) as

$$P_1 + \cdots + P_d = (\phi_1(P_1, \ldots, P_d), \ldots, \phi_{d+d'}(P_1, \ldots, P_d)),$$

where $\phi_1, \ldots, \phi_{d+d'}$ are rational functions of the coordinates used. By setting this expression equal to R one obtains a set of equations, which together with the equations arising from membership of A or $\mathcal{F}$ results in a system that will generically be of dimension zero, whose solutions can be found by a Gröbner basis computation, or sometimes by faster methods, depending on A and its element and group law representation.

4.3 The Granger-Vercauteren attack

The algorithm of Granger and Vercauteren uses the affine representation of elements of an algebraic torus, and the group law induced in this representation by field multiplication, i.e., the usual group law. The key insight of the work is that for a T_n which possesses a rational parameterisation, only $\phi(n)$ elements of a factor base need to be added in order to generate a random element of the group with constant probability. In comparison with using the field representation and defining the decomposition base to be the set of monic linear polynomials, for example, the probability of generating a relation is $1/\phi(n)!$ rather than $1/n!$. Therefore, it is the very compression which made torus-based cryptography attractive, that enables a significant speed up to be made when computing discrete logarithms. We now describe the algorithm for T_2 and T_6 respectively. In the following we assume q is odd.

Algorithm for $T_2(\mathbb{F}_{q^n}) \subset \mathbb{F}_{q^{2n}}^\times$. By the discussion in §4 the prime order subgroup would be in $T_{2n}(\mathbb{F}_q) \subsetneq T_2(\mathbb{F}_{q^n})$, but since we exploit the rationality of T_2 rather than T_{2n}, we work with $T_2(\mathbb{F}_{q^n})$, or more precisely the dimension n variety $(\mathrm{Res}_{\mathbb{F}_{q^n}/\mathbb{F}_q} T_2)(\mathbb{F}_q)$, where Res denotes the Weil restriction of scalars (see [73]).

Let $\mathbb{F}_{q^n} \cong \mathbb{F}_q[t]/(f(t))$ with $f(t) \in \mathbb{F}_q[t]$ an irreducible monic polynomial of degree n. We shall use the polynomial basis $\{1, t, \ldots, t^{n-1}\}$. For a non-square $\delta \in \mathbb{F}_{q^n} \backslash \mathbb{F}_q$, let $\mathbb{F}_{q^{2n}} = \mathbb{F}_{q^n}[\gamma]/(\gamma^2 - \delta)$, with basis $\{1, \gamma\}$. From Definition 2 we have

$$T_2(\mathbb{F}_{q^n}) = \{(x, y) \in \mathbb{F}_{q^n} \times \mathbb{F}_{q^n} : x^2 - \delta y^2 = 1\}.$$

Rather than use two elements of $\mathbb{F}_{q^n}$ to represent each point, as the torus T_2 is one-dimensional and rational, one can use the following affine representation:

$$T_2(\mathbb{F}_{q^n}) = \left\{ \frac{z-\gamma}{z+\gamma} : z \in \mathbb{F}_{q^n} \right\} \cup \{O\}, \tag{3}$$

where O is the point at infinity. Note that for $g = g_0 + g_1\gamma \in T_2(\mathbb{F}_{q^n})$ in the $\mathbb{F}_{q^{2n}}$ representation, the corresponding affine representation is $z = -(1+g_0)/g_1$ if $g_1 \neq 0$, while -1 and 1 map to $z = 0$ and $z = O$ respectively. Since $T_2(\mathbb{F}_{q^n})$ has q^n+1 elements, this representation is optimal: note that this map is really from $T_2(\mathbb{F}_{q^n})$ to $\mathbb{P}^1(\mathbb{F}_{q^n})$.

We define the decomposition base as follows:

$$\mathcal{F} = \left\{ \frac{a-\gamma}{a+\gamma} : a \in \mathbb{F}_q \right\} \subset T_2(\mathbb{F}_{q^n}),$$

which contains precisely q elements since the above map is a birational isomorphism from T_2 to $\mathbb{A}^1$. Now let P be a generator and $Q \in \langle P \rangle$. To find relations we test whether for random integers a, b modulo the group order, $R = aP + bQ$ decomposes as a sum of n points in $\mathcal{F}$, i.e.,

$$R = P_1 + \cdots + P_n, \tag{4}$$

with $P_1, \ldots, P_n \in \mathcal{F}$. In the affine representation this becomes

$$\frac{r-\gamma}{r+\gamma} = \prod_{i=1}^{n} \left(\frac{a_i-\gamma}{a_i+\gamma} \right),$$

where the $a_i \in \mathbb{F}_q$ are unknowns and $r \in \mathbb{F}_{q^m}$ is the affine representation of R. As the right hand side is symmetric in the a_i we may expand it in terms of the elementary symmetric polynomials $\sigma_i(a_1, \ldots, a_n)$ of the a_i:

$$\frac{r-\gamma}{r+\gamma} = \frac{\sigma_n - \sigma_{n-1}\gamma + \cdots + (-1)^n\gamma^n}{\sigma_n + \sigma_{n-1}\gamma + \cdots + \gamma^n}.$$

Reducing modulo the defining polynomial of γ, we obtain:

$$\frac{r-\gamma}{r+\gamma} = \frac{b_0(\sigma_1, \ldots, \sigma_n) - b_1(\sigma_1, \ldots, \sigma_n)\gamma}{b_0(\sigma_1, \ldots, \sigma_n) + b_1(\sigma_1, \ldots, \sigma_n)\gamma},$$

where b_0, b_1 are linear in the σ_i and have coefficients in $\mathbb{F}_{q^n}$. Reducing the right hand side to the affine representation (3) we obtain the equation

$$b_0(\sigma_1, \ldots, \sigma_n) - b_1(\sigma_1, \ldots, \sigma_n)r = 0.$$

Since the σ_i are in $\mathbb{F}_q$, by expressing this equation on the polynomial basis of $\mathbb{F}_{q^n}$ we obtain n linear equations over $\mathbb{F}_q$ in the n unknowns σ_i. If there is a solution $(\sigma_1, \ldots, \sigma_n)^T$ to this linear system, we see whether it corresponds to a solution of (4) by checking whether the following polynomial splits completely over $\mathbb{F}_q$:

$$p(x) := x^n - \sigma_1 x^{n-1} + \sigma_2 x^{n-2} - \cdots + (-1)^n \sigma_n.$$

Whenever it does, the roots $a_1, \ldots, a_n$ will be the affine representations of elements of $\mathcal{F}$ which sum to R, i.e., we have found a relation.

In terms of complexity, when $n! \approx q$ the full algorithm runs in time $L_{q^n}(1/2, c)$ for some $c > 0$. Experiments in the computer algebra system Magma [6] reported in [28] demonstrated that it would be faster than Pollard's rho in a (at the time standard) 160 bit subgroup, when q^{2n} was between 400 and 1000 bits, thus indicating its efficacy for some practical parameters.

Algorithm for $T_6(\mathbb{F}_{q^n}) \subset \mathbb{F}_{q^{6n}}^\times$. As we saw before, the prime order subgroup would be in $T_{6n}(\mathbb{F}_q) \subset T_6(\mathbb{F}_{q^n})$, but since we exploit the rationality of T_6 and not T_{6n} we shall work with $T_6(F_{q^n})$, or rather its Weil restriction $(\mathrm{Res}_{\mathbb{F}_{q^n}/\mathbb{F}_q} T_6)(\mathbb{F}_q)$. The central difference between this algorithm and the T_2 algorithm is that for the present case the equations to be solved in the decomposition step are no longer linear.

Let $\mathbb{F}_{q^n} \cong \mathbb{F}_q[t]/(f(t))$, with $f(t)$ an irreducible polynomial of degree n, and use the polynomial basis $\{1, t, t^2, \ldots, t^{n-1}\}$. For the birational map from $T_6(\mathbb{F}_{q^n})$ to $\mathbb{A}^2(\mathbb{F}_{q^n})$ we use the specifications for CEILIDH, the compression mechanism described by Rubin and Silverberg [73]. Assume that $q^n \equiv 2$ or 5 mod 9, and for $(r, q) = 1$ let ζ_r denote a primitive r-th root of unity in $\overline{\mathbb{F}_{q^n}}$. Let $x = \zeta_3$ and let $y = \zeta_9 + \zeta_9^{-1}$. Then $x^2 + x + 1 = 0$ and $y^3 - 3y + 1 = 0$. Furthermore let $\mathbb{F}_{q^{3n}} = \mathbb{F}_{q^n}(y)$ and $\mathbb{F}_{q^{6n}} = \mathbb{F}_{q^{3n}}(x)$. The bases we use are $\{1, y, y^2 - 2\}$ for the degree three extension and $\{1, x\}$ for the degree two extension. Let $V(f)$ be the zero set of $f(\alpha_1, \alpha_2) = 1 - \alpha_1^2 - \alpha_2^2 + \alpha_1\alpha_2$ in $\mathbb{A}^2(\mathbb{F}_{q^n})$. The following are inverse birational maps:

- $\psi : \mathbb{A}^2(\mathbb{F}_{q^n}) \setminus V(f) \xrightarrow{\sim} T_6(\mathbb{F}_{q^n}) \setminus \{1, x^2\}$, defined by

$$\psi(\alpha_1, \alpha_2) = \frac{1 + \alpha_1 y + \alpha_2(y^2 - 2) + (1 - \alpha_1^2 - \alpha_2^2 + \alpha_1\alpha_2)x}{1 + \alpha_1 y + \alpha_2(y^2 - 2) + (1 - \alpha_1^2 - \alpha_2^2 + \alpha_1\alpha_2)x^2}, \tag{5}$$

- $\rho : T_6(\mathbb{F}_{q^n}) \setminus \{1, x^2\} \xrightarrow{\sim} \mathbb{A}^2(\mathbb{F}_{q^n}) \setminus V(f)$, which is defined as follows: for $\beta = \beta_1 + \beta_2 x$, with $\beta_1, \beta_2 \in \mathbb{F}_{q^{3n}}$, let $(1 + \beta_1)/\beta_2 = u_1 + u_2 y + u_3(y^2 - 2)$, then $\rho(\beta) = (u_2/u_1, u_3/u_1)$.

We define the decomposition base as follows:

$$\mathcal{F} = \left\{ \frac{1 + (at)y + (1 - (at)^2)x}{1 + (at)y + (1 - (at)^2)x^2} : a \in \mathbb{F}_q \right\}$$

which clearly contains q elements. Note that we use $\psi(at, 0)$ rather than $\psi(a, 0)$ since the latter would map to the strict subvariety $T_6(\mathbb{F}_q)$. Since $(\mathrm{Res}_{\mathbb{F}_{q^n}/\mathbb{F}_q} T_6)(\mathbb{F}_q)$ is $2n$-dimensional, to find relations we need to solve

$$R = P_1 + \cdots + P_{2n}\,, \tag{6}$$

with $P_1, \ldots, P_{2n} \in \mathcal{F}$. Assuming that R is expressed in affine form, i.e., $R = \psi(r_1, r_2)$, we obtain

$$\frac{1 + r_1 y + r_2(y^2 - 2) + (1 - r_1^2 - r_2^2 + r_1 r_2)x}{1 + r_1 y + r_2(y^2 - 2) + (1 - r_1^2 - r_2^2 + r_1 r_2)x^2} = \prod_{i=1}^{2n} \left(\frac{1 + (a_i t)y + (1 - (a_i t)^2)x}{1 + (a_i t)y + (1 - (a_i t)^2)x^2} \right).$$

Upon expanding the product of the numerators and denominators, the right hand side becomes

$$\frac{b_0 + b_1 y + b_2(y^2 - 2) + \big(c_0 + c_1 y + c_2(y^2 - 2)\big)\, x}{b_0 + b_1 y + b_2(y^2 - 2) + (c_0 + c_1 y + c_2(y^2 - 2))\, x^2} \tag{7}$$

with b_i, c_i polynomials over $\mathbb{F}_{q^n}$ of degree $4n$ in $a_1, \ldots, a_{2n}$. In general, these polynomials have a large number of terms and are thus slow to compute with. However, as before by construction these polynomials are symmetric in the $a_1, \ldots, a_{2n}$, so one can rewrite the b_i and c_i in terms of the $2n$ elementary symmetric polynomials $\sigma_j(a_1, \ldots, a_{2n})$ for $j = 1, \ldots, 2n$. This dramatically reduces the degree and size of these polynomials: in particular they become quadratic and as a consequence the number of terms is much lower, being bounded by $4n + \binom{2n}{2} + 1$.

To generate a system of quadratic equations, we use the embedding of $T_6(\mathbb{F}_{q^n})$ into $T_2(\mathbb{F}_{q^{3n}})$ and consider the Weil restriction of the following equality:

$$\frac{b_0 + b_1 y + b_2(y^2 - 2)}{c_0 + c_1 y + c_2(y^2 - 2)} = \frac{1 + r_1 y + r_2(y^2 - 2)}{1 - r_1^2 - r_2^2 + r_1 r_2}\,.$$

This leads to 3 quadratic equations over $\mathbb{F}_{q^n}$ or equivalently, to $3n$ quadratic equations over $\mathbb{F}_q$ in the $2n$ unknowns $\sigma_1, \ldots, \sigma_{2n}$. Observe that amongst these equations there must be at least n dependencies arising from the fact that we used the embedding into T_2 rather than T_6.

The properties of such systems, which have the same structure but differ only by the coefficients of R, were investigated using the Magma implementation of the F4 algorithm [17]. It was found that: the ideal generated is zero-dimensional; the Gröbner basis with respect to the lexicographic ordering satisfies the so-called Shape Lemma, i.e., the basis is of the form:

$$\sigma_1 - g_1(\sigma_{2n}),\ \sigma_2 - g_2(\sigma_{2n}),\ \ldots,\ \sigma_{2n-1} - g_{2n-1}(\sigma_{2n}),\ g_{2n}(\sigma_{2n})\,,$$

where $g_i(\sigma_{2n})$ is a univariate polynomial in σ_{2n} for each i; and in all cases it holds that $\deg(g_{2n}) = 3^n$, rather than the bound of 2^{2n} that one would expect from Bezout's theorem.

Provided that n is not prohibitively large, such systems can be solved in a reasonable time. To test if a random point R decomposes over $\mathcal{F}$, one computes the roots of $g_{2n}(\sigma_{2n})$ in $\mathbb{F}_q$, and then substitutes these in the other g_i to find the values of the other σ_i. For each such solution we then test if the polynomial

$$p(x) := x^{2n} - \sigma_1 x^{2n-1} + \sigma_2 x^{2n-2} - \cdots + (-1)^{2n}\sigma_{2n}$$

splits completely over $\mathbb{F}_q$. Whenever it does, the roots a_i for $i = 1, \ldots, 2n$ lead to a relation of the form (6).

In terms of complexity, when $n! \approx q$ the full algorithm runs in time $L_{q^n}(1/2, c')$ for some $c' > 0$. In terms of experimental results, Table 2 of [28] gave expected running times for attacking $T_6(\mathbb{F}_{q^n})$, for $n = 1, \ldots, 5$. In particular, it showed that in the group $T_{30}(\mathbb{F}_q)$ – which can be embedded into $T_6(\mathbb{F}_{q^5})$ – discrete logarithms are easier than previously expected. Indeed, this group was proposed in [83] and [58] for cryptographic use and keys of length 960 bits were recommended, i.e., , with $q \approx 2^{32}$. The experiments showed that even with a Magma implementation it would be feasible to compute discrete logarithms in $T_{30}(\mathbb{F}_q)$ with $q \approx 2^{20}$, and the attack for $q \approx 2^{32}$ would be about 1000 times faster than Pollard's rho, albeit with a far larger memory constraint. In light of this attack, the security offered by the DLP in finite fields of the form $\mathbb{F}_{q^{30}}$ needed to be reassessed.

5 Discrete logarithms in finite fields of fixed characteristic

Progress in cryptanalytic algorithms, just as in science more generally, usually evolves by small increments but with occasional revolutionary steps forward [54]. One example of such a step forward could arguably be the rapid development of efficient algorithms for solving the DLP in finite fields of fixed characteristic, that took place from late 2012 to mid 2014, thanks to the present authors and their collaborators. Between these times, the fastest algorithm for solving this problem went from having complexity $L(1/3)$ to being quasi-polynomial [25, 39, 26, 2, 30, 33, 32], rendering such fields entirely unsuitable for discrete logarithm-based cryptography, including pairing-based cryptography over small characteristic supersingular curves. These events constitute a perfect example of Prof. Lenstra's (perhaps jocular, but no doubt in part quite serious) contention that no problem based on number theory should ever be considered truly secure, even if it has remained impenetrable for several decades[6].

Since 2014, there have been many surveys of the state-of-art in discrete logarithm algorithms for finite fields [41, 43, 31]. Therefore, in the present section we focus only on the key ideas behind the fixed characteristic breakthroughs, to give a flavour of what was behind them, as well as the central results.

5.1 Key insights

If one performs the basic index calculus method as described in §1.2, in $\mathbb{F}_{q^n}$ for fixed q and $n \to \infty$, then by using a theorem due to Odlyzko [69] and Lovorn [61], the distribution of smooth polynomials naturally leads to an $L(1/2)$ complexity algorithm. It may therefore have been assumed for some years that this is the best complexity that can be achieved.

A key avenue to improving these index calculus algorithms is to find an approach that generates the relations faster. A first idea, is to create relations between elements with smaller norms, in order to increase the smoothness probability and also reduce the size of the factor base. This is the basis of Coppersmith's celebrated 1984 algorithm [10, 11] and all subsequent $L(1/3)$ algorithms for larger characteristic. These algorithms become heuristic because the considered relations involve equality between elements which are neither independent nor uniformly distributed. A long standing open problem is to find a way to lift these heuristics.

Alternatively, it turned out that the lack of independence can be used to speed-up index calculus for certain fields. This was first described by Joux in 2012 [38], building upon Joux and Lercier's medium prime function field sieve method [40].

This gives the hope to be able to generate field equations between elements that have *better than expected* smoothness properties. This idea is more subtle than one might assume in retrospect, seeing the breakthroughs it led to. Indeed, before these breakthroughs, this possibility does not even seem to have ever been considered. Most likely because from a complexity analysis perspective it seemed essential that the expected smoothness properties of generated elements hold.

It is an instantiation of this second idea that initiated the aforementioned progress in this area. The technique was discovered independently and at approximately the

[6] This perspective is attributed to Prof. Lenstra by the first listed author, he having worked with him for four years and having discussed such matters a few times; any error in attribution is entirely his.

same time by the present authors, in two different but essentially isomorphic approaches. In particular, it was shown how to produce a family of polynomials of high degree that are smooth by construction, and which thus lead to useful relations, in contrast to uniformly generated polynomials of the same degree, which have only an exponentially small probability of being smooth. The family of polynomials and its exploitation lay the foundation for two independent and theoretically distinct quasi-polynomial algorithms: the first due to Barbulescu, Gaudry, Joux, Thomé in 2013 [2] and the second due to Granger, Kleinjung and Zumbrägel in 2014 [32].

In terms of historical development, the approach of the BGJT quasi-polynomial algorithm grew naturally from Joux's 2013 paper [39], which is an extension of the previously mentioned method of [38] for medium characteristic. On the other hand, the GKZ quasi-polynomial algorithm grew from observations in [30] and the techniques of [25], which combined independent observations with a specialisation of the field representation in [40], which was itself motivated by the Granger-Vercauteren algorithm [28].

5.2 Polynomial-time relation generation

In order to give a flavour of the ideas behind the breakthrough techniques, we now describe the polynomial time relation generation methods published by Göloğlu, Granger, McGuire and Zumbrägel [25], and Joux [39], both in February 2013.

Both methods start with a family of finite fields $\mathbb{F}_{p^n}$ in which the DLP is to be solved, with p fixed and $n \to \infty$. Each of these is embedded into a corresponding field of the form $\mathbb{F}_Q = \mathbb{F}_{q^{kn}}$, with $k \geq 2$ fixed and $n \approx q$, by setting $q = p^\ell$ with $\ell = \lceil \log_p n \rceil$, increasing the extension degree by a factor of $k \lceil \log_p n \rceil$, which does not significantly impact the complexity of the resulting algorithms.

The GGMZ method Let $\mathbb{F}_{p^n} \hookrightarrow \mathbb{F}_{q^{kn}}$ be the target field and let $\mathbb{F}_{q^k}$ be represented arbitrarily. In order to represent $\mathbb{F}_{q^{kn}}$, the GGMZ method uses an extremely unbalanced version of the field representation employed in the Joux-Lercier function field sieve [40]. In particular, let $f = X^q$ and $g = \frac{h_0}{h_1}$ for some $h_0, h_1 \in \mathbb{F}_{q^k}[X]$ of low degree[7] $\leq d_h$, so that there exists a monic irreducible $I \in \mathbb{F}_{q^k}[X]$ of degree n such that $h_1(X^q)X - h_0(X^q) \equiv 0 \pmod{I}$. For such h_0, h_1, I we define $\mathbb{F}_{q^{kn}} := \mathbb{F}_{q^k}[X]/(I)$.

Let x be a root of I in $\mathbb{F}_{q^{kn}}$ and let $y = f(x) = x^q$. Then by construction we have $x = g(y) = \frac{h_0(y)}{h_1(y)}$, giving an isomorphism between two representations of $\mathbb{F}_{q^{kn}}$, namely $\mathbb{F}_{q^k}(x)$ and $\mathbb{F}_{q^k}(y)$. The factor base is as simple as could be expected, consisting of $h_1(x^q)$ and all linear polynomials on the x-side; the y-side factor base is unnecessary since for all $d \in \mathbb{F}_{q^k}$ one has $(y + d) = (x + d^{1/q})^q$.

For $a, b, c \in \mathbb{F}_{q^k}$, consider elements of $\mathbb{F}_{q^{kn}}$ of the form $xy + ay + bx + c$. Using the above field isomorphisms we have the following identity:

$$x^{q+1} + ax^q + bx + c = \frac{1}{h_1(y)}(yh_0(y) + ayh_1(y) + bh_0(y) + ch_1(y)). \qquad (8)$$

A key observation is that the l.h.s. of Eq. (8) has a very special form, and provably splits completely over $\mathbb{F}_{q^k}$ with probability $\approx 1/q^3$, which is exponentially higher

[7] In [25] h_1 was not specified and was thus implicitly 1; h_1 was introduced in [30] in order to increase the number of representable extension degrees.

than the probability that a uniformly random polynomial of the same degree splits completely over $\mathbb{F}_{q^k}$, which is $\approx 1/(q+1)!$. Indeed, for $k \geq 3$ consider the polynomial $X^{q+1} + aX^q + bX + c$. For $ab \neq c$ and $a^q \neq b$, this polynomial may be transformed (up to a scalar) into

$$F_B(\overline{X}) = \overline{X}^{q+1} + B\overline{X} + B\,, \quad \text{with} \quad B = \frac{(b-a^q)^{q+1}}{(c-ab)^q}\,,$$

via $X = \frac{c-ab}{b-a^q}\,\overline{X} - a$. Observe that the original polynomial splits completely over $\mathbb{F}_{q^k}$ whenever F_B splits completely over $\mathbb{F}_{q^k}$ and we have a valid transformation from $\overline{X}$ to X. The following theorem provides the precise number of $B \in \mathbb{F}_{q^k}$ for which $F_B(\overline{X})$ splits completely over $\mathbb{F}_{q^k}$.

Theorem 1. (Bluher [5]) *The number of elements $B \in \mathbb{F}_{q^k}^{\times}$ s.t. the polynomial $F_B(\overline{X}) \in \mathbb{F}_{q^k}[\overline{X}]$ splits completely over $\mathbb{F}_{q^k}$ equals*

$$\frac{q^{k-1}-1}{q^2-1} \quad \textit{if } k \textit{ is odd}\,, \qquad \frac{q^{k-1}-q}{q^2-1} \quad \textit{if } k \textit{ is even}\,.$$

By using the expression for B in terms of a, b, c one can generate triples (a, b, c) for which the l.h.s. of Eq. (8) *always* splits over $\mathbb{F}_{q^k}$. In particular, firstly compute the set $\mathcal{B}$ of all $B \in \mathbb{F}_{q^k}$ for which F_B splits over $\mathbb{F}_{q^k}$. Then for any $a, b \neq a^q$ and $B \in \mathcal{B}$ there is a uniquely determined c for which the l.h.s. splits. By Theorem 1, there are $\approx q^{3k-3}$ such triples, giving the aforementioned probability $1/q^3$. For such triples, whenever the r.h.s. of Eq. (8) splits, one obtains a relation amongst the factor base elements.

This just leaves the case $k = 2$, for which there are no such F_B. However, the set of triples for which the l.h.s. splits non-trivially can be shown to be

$$\{(a, a^q, c) \mid a \in \mathbb{F}_{q^2} \text{ and } c \in \mathbb{F}_q, c \neq a^{q+1}\}.$$

So for $k \geq 2$, assuming the r.h.s. splits with probability $1/(d_h + 1)!$, there will be sufficiently many relations when $q^{2k-3} > (d_h + 1)!$. Then for fixed d_h and $q \to \infty$ the cost of computing the logarithms of all of the factor base elements is heuristically $O(q^{2k+1})$ operations in $\mathbb{Z}/(q^{kn}-1)\mathbb{Z}$ as one can use sparse (weight q) linear algebra techniques; for fixed k this complexity is polynomial in $\log Q = q^{1+o(1)}$, as claimed.

Joux's method Joux's method [39] also applies to fields of the form $\mathbb{F}_Q = \mathbb{F}_{q^{kn}}$ (with $k = 2$ being used for the exposition and initial examples), but the crucial degree n extension is built in a slightly different, but analogous manner. In particular, we have $\mathbb{F}_Q = \mathbb{F}_{q^k}(x) = \mathbb{F}_{q^k}[X]/(I)$, where $I \mid h_1(X)X^q - h_0(X)$ for some $h_0, h_1 \in \mathbb{F}_{q^k}[X]$ of low degree $\leq d_h$. This leads to the field equation $x^q = \frac{h_0(x)}{h_1(x)}$. The factor base consists of $h_1(x)$ and all linear polynomials in x.

Joux's method starts with the identity

$$\prod_{\mu \in \mathbb{F}_q} (X - \mu) = X^q - X.$$

If one substitutes X by $\frac{\alpha X + \beta}{\gamma X + \delta}$ with $\alpha, \beta, \gamma, \delta \in \mathbb{F}_{q^k}$ and $\alpha\delta - \beta\gamma \neq 0$, multiplying by $(\gamma X + \delta)^{q+1}$ gives

$$(\gamma X+\delta) \prod_{\mu \in \mathbb{F}_q} ((\alpha X+\beta) - \mu(\gamma X+\delta)) = (\alpha X+\beta)^q(\gamma X+\delta) - (\alpha X+\beta)(\gamma X+\delta)^q. \tag{9}$$

Observe that the r.h.s. of Eq. (9) has the same monomial degrees as the l.h.s. of Eq. (8), and automatically splits completely over $\mathbb{F}_{q^k}$ by virtue of the l.h.s. of Eq. (9). Applying the field equation $x^q = \frac{h_0(x)}{h_1(x)}$ to the r.h.s. of Eq. (9) produces

$$\tfrac{1}{h_1(x)}(\alpha^q h_0(x) + \beta^q h_1(x))(\gamma x + \delta) - (\alpha x + \beta)(\gamma^q h_0(x) + \delta^q h_1(x)),$$

and if this degree d_h+1 polynomial also splits over $\mathbb{F}_{q^k}$ then one has a relation amongst factor base elements.

In order to count the number of distinct splitting polynomials that one can obtain in this manner, first observe that the total number of $(\alpha,\beta,\gamma,\delta)$-transformations is $|\,\mathrm{PGL}_2(\mathbb{F}_{q^k})| = q^{3k} - q^k$. Second, observe that two transformations will give the same relation (up to multiplication by a scalar in $\mathbb{F}_{q^k}^{\times}$) if there exists an element of $\mathrm{PGL}_2(\mathbb{F}_q)$ which when multiplied by the first transformation gives the second. Hence the total number of distinct transformations is $\approx q^{3k-3}$, just as we found for the GGMZ method. From a practical perspective, in order to avoid repetitions one should compute a set of coset representatives for the quotient $\mathrm{PGL}_2(\mathbb{F}_{q^k})/\,\mathrm{PGL}_2(\mathbb{F}_q)$; by contrast the GGMZ method already achieves this implicitly.

5.3 $L(1/4 + o(1))$ and quasi-polynomial algorithms

The two methods just described mean that the first stage of index calculus is (at least heuristically) solvable in polynomial time. So the remaining problem is to compute individual logarithms. However, due to the extension degree being $n = O(q)$ and the factor base being only polynomial in the size of the field, this is now much harder than before. In particular, if one uses the usual descent method from [40] then the elimination of an element - i.e., expressing it as a product of elements of lower degree, modulo the field polynomial – becomes harder as the degree becomes smaller, with degree two eliminations being the bottleneck. However, with independent and distinct methods GGMZ [25] and Joux [39] showed how to eliminate degree two elements efficiently. For reasons of space we refer the reader to the original papers for their expositions (or to the survey article [31]), and note that these methods spawned the building blocks of the individual logarithm stages of the two aforementioned quasi-polynomial algorithms.

In [39] Joux also gave a new elimination method which relies on solving multivariate bilinear quadratic systems via Gröbner basis computations, whose cost increases with the degree. Balancing the costs of the Gröbner basis descent and the classical descent (whose cost decreases with the degree) results in a heuristic $L(1/4+o(1))$ algorithm, which was the first algorithm to break the long-standing $L(1/3)$ barrier. This can be tweaked for fields of the present form to obtain an $L(1/4)$ algorithm [26].

Soon afterwards in June 2013, Barbulescu, Gaudry, Joux and Thomé announced an algorithm for solving the DLP [2] in the fields $\mathbb{F}_{q^{kn}}$ with $k \geq 2$ fixed and $n \leq q + d$ with d very small, which for $n \approx q$ has heuristic quasi-polynomial time complexity

$$(\log q^{kn})^{O(\log\log q^{kn})} = \exp(O((\log\log q^{kn})^2)). \tag{10}$$

Since (10) is smaller than $L(\alpha)$ for any $\alpha > 0$, this constituted a very significant breakthrough for the DLP in finite fields of fixed characteristic. Moreover, when the cardinality of the base field $\mathbb{F}_{q^k}$ can be written as $q^k = L_{q^{kn}}(\alpha)$, the algorithm results in complexity $L(\alpha)$, thus providing an improvement over the original function field sieve algorithms whenever $\alpha < 1/3$. As for the $L(1/4)$ method, this algorithm relies on unproven heuristics. Moreover, it is an asymptotic improvement whose cross-over point with previous techniques is too high to make it usable in record computations.

In February 2014 Granger, Kleinjung and Zumbrägel developed an alternative quasi-polynomial algorithm for fields of essentially the same form. Just as the BGJT elimination step may be viewed as a generalisation of Joux's degree two elimination method, the GKZ elimination step depends on the degree two elimination method of GGMZ (albeit combined with another crucial but simple idea). Thanks to the algebraic nature of the elimination method, the only assumption required for the algorithm to be rigorously proven to work is one regarding the existence of a suitable field representation. In particular, the following theorems were proven in [32].

Theorem 2. *Given a prime power $q > 61$ that is not a power of 4, an integer $k \geq 18$, coprime polynomials $h_0, h_1 \in \mathbb{F}_{q^k}[X]$ of degree at most two and an irreducible degree n factor I of $h_1 X^q - h_0$, the DLP in $\mathbb{F}_{q^{kn}} \cong \mathbb{F}_{q^k}[X]/(I)$ can be solved in expected time*

$$q^{\log_2 n + O(k)}.$$

That the degree of h_0, h_1 is at most two is essential to eliminating smoothness heuristics, since this ensures that the cofactor of the r.h.s. of Eq. (8) has degree at most one, and is thus automatically 1-smooth. This theorem is reproved by a slightly easier approach that gives better parameters for q and k in [24]. A simple application of Kummer theory shows that such h_1, h_0 exist when $n = q - 1$, which gives the following easy corollary when $m = ik(p^i - 1)$.

Theorem 3. *For every prime p there exist infinitely many explicit extension fields $\mathbb{F}_{p^m}$ in which the DLP can be solved in expected quasi-polynomial time*

$$\exp\left((1/\log 2 + o(1))(\log m)^2\right).$$

In practice it is very easy to find polynomials h_0, h_1 for general extension degrees as per Theorem 2, and heuristically it would appear to be all but guaranteed. However, proving their existence seems to be a hard problem. The idea of using an alternative field representation arising from torsion points of elliptic curves to obviate this issue is a very natural one. Such a field representation was initially introduced by Couveignes and Lercier in [12]. At least three teams of researchers have developed this idea [60, 50, 44] in order to build an analogue of the GKZ algorithm using this alternative field representation. As of the time of writing, the work of Kleinjung and Wesolowski [50] is the only one containing a full proof. Previously, only an $L(1/2)$ complexity had been proven rigorously for arbitrary extension degrees, so this is a very significant theoretical result. More precisely, [50] proved:

Theorem 4. *Given any prime number p and any positicve integer n, the discrete logarithm problem in the group $\mathbb{F}_{p^n}$ can be solved in expected time $(pn)^{2\log_2(n)+O(1)}$.*

5.4 Practical impact

From the perspective of mathematical cryptology, rigorously proving the correctness of new DLP algorithms is of central theoretical interest. However, in terms of real

Table 1. Large-scale discrete logarithm computations in finite fields of small or medium characteristic. Details of uncited results can be found in the number theory mailing list [68]

bitlength	charact.	Kummer	who/when	running time
127	2	no	Coppersmith 1984 [10]	$L(1/3, 1.526..1.587)$
401	2	no	Gordon, McCurley 1992 [27]	$L(1/3, 1.526..1.587)$
521	2	no	Joux, Lercier 2001	$L(1/3, 1.526)$
607	2	no	Thomé 2002	$L(1/3, 1.526..1.587)$
613	2	no	Joux, Lercier 2005	$L(1/3, 1.526)$
556	medium	yes	Joux, Lercier 2006 [40]	$L(1/3, 1.442)$
676	3	no	Hayashi *et al.* 2010 [35]	$L(1/3, 1.442)$
923	3	no	Hayashi *et al.* 2012 [34]	$L(1/3, 1.442)$
1175	medium	yes	Joux 24 Dec 2012	$L(1/3, 1.260)$
619	2	no	CARAMEL 29 Dec 2012	$L(1/3, 1.526)$
1425	medium	yes	Joux 6 Jan 2013	$L(1/3, 1.260)$
1778	2	yes	Joux 11 Feb 2013	$L(1/4 + o(1))$
1971	2	yes	GGMZ 19 Feb 2013	$L(1/3, 0.763)$
4080	2	yes	Joux 22 Mar 2013	$L(1/4 + o(1))$
809	2	no	CARAMEL 6 Apr 2013	$L(1/3, 1.526)$
6120	2	yes	GGMZ 11 Apr 2013	$L(1/4)$
3164	2	yes	GGMZ May 2013	$L(1/3, 0.763)$
6168	2	yes	Joux 21 May 2013	$L(1/4 + o(1))$
1303	3	no	AMOR 27 Jan 2014	$L(1/4 + o(1))$
4404	2	no	GKZ 30 Jan 2014	$L(1/4 + o(1))$
9234	2	yes	GKZ 31 Jan 2014	$L(1/4 + o(1))$
1551	3	no	AMOR 26 Feb 2014	$L(1/4 + o(1))$
3796	3	no	Joux, Pierrot 15 Sep 2014	$L(0 + o(1))$
1279	2	no	Kleinjung 17 Oct 2014	$L(0 + o(1))$
4841	3	no	ACCMORR, 18 Jul 2016	$L(0 + o(1))$
30750	2	yes	GKLWZ, 10 July 2019	$L(0 + o(1))$

world cryptographic impact, what matters far more is how practical the algorithms are and whether they can be used to solve previously unsolvable DLP instances. Furthermore, as is well known to practitioners and computational number theorists, carrying out large-scale implementations often leads to new theoretical insights that can, in turn, result in superior algorithms. Hence, the value of practical considerations should not be overlooked.

Shortly after GGMZ and Joux discovered their methods, a period of intense competition began, both in theory [25, 39, 26, 2, 30, 33, 42, 32] as already alluded to, and in practice, see Table 1. As one can see these computational records dwarfed those that had been set previously, leading small characteristic pairing-based cryptography to be entirely eschewed by the cryptographic community. Without doubt, this 'academic arms race' accelerated and stretched the development of the new discrete logarithm algorithms, and as such were scientifically extremely beneficial.

As of the time of writing the largest such (publicly known) DLP computation was completed in the field $\mathbb{F}_{2^{30750}}$, by Granger, Kleinjung, Lenstra, Wesolowski and Zumbrägel; this was announced in July 2019 and required approximately 2900 core years [29]. The main purpose of the computation was to test the GKZ quasi-polynomial descent method at scale for the first time, in order to assess its reach when the number of core hours expended is comparable to the number expended during the largest DLP computations in prime fields and integer factorisation efforts. At the time of the announcement [29] the record for the former was in a field of bitlength 768, set in June 2016 [49]; the current record is in a field of bitlength 795, announced in December 2019 [7]. For the latter, at the time of the announcement [29] the record was the factorisation of a 768-bit RSA challenge modulus, set in December 2009 [47]. Also in December 2019, the factorisation of a 795-bit RSA challenge modulus was announced [7], which was swiftly improved upon in February 2020 by the solving of an 829-bit RSA challenge [8]. For Mersenne numbers, an implementation of Coppersmith's factorisation factory idea resulted in January 2015 in the factorisation of the 17 remaining unfactored moduli of the form $2^n - 1$ with $1007 \leq n \leq 1199$ [48].

In terms of remaining hard open problems in the area of finite field discrete logarithms, there are two central – and natural – ones. The first challenging problem is to find a classical polynomial time algorithm for fixed characteristic DLPs, either heuristic or rigorous. The second, probably far more challenging problem is to develop quasi-polynomial classical algorithms for medium and large characteristic fields. As there is far less structure for prime fields in particular, it seems that fundamentally new ideas will be required.

6 Conclusion

As is well known, the DLP and the integer factorisation problem can be solved in polynomial time using a sufficiently large quantum computer [78]. At present, such computers are not available, despite a wide-spread worry or excitement that they might come soon. To be ready when this occurs, a large part of the cryptographic community is currently working on post-quantum secure alternatives. However, the flexibility of discrete logarithms for constructing cryptographic protocols is currently unsurpassed. As a consequence, it remains essential to study the security of discrete logarithms against classical computers. New sporadic breakthroughs could possibily occur and would likely also affect factoring and the RSA cryptosystem.

Acknowledgements

This work has been supported in part by the European Union's H2020 Programme under grant agreement number ERC-669891.

Bibliography

[1] R. Barbulescu and S Duquesne. Updating Key Size Estimations for Pairings. *J. Cryptology*, 32:1298–1336, 2019.

[2] Razvan Barbulescu, Pierrick Gaudry, Antoine Joux, and Emmanuel Thomé. A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. In *Advances in Cryptology—EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 1–16. Springer, 2014.

[3] Razvan Barbulescu, Pierrick Gaudry, and Thorsten Kleinjung. The Tower Number Field Sieve. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology – ASIACRYPT 2015*, pages 31–55, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[4] Razvan Barbulescu and Cécile Pierrot. The multiple number field sieve for medium- and high-characteristic finite fields. *LMS Journal of Computation and Mathematics*, 17(A):230–246, 2014.

[5] Antonia W. Bluher. On $x^{q+1} + ax + b$. *Finite Fields and Their Applications*, 10(3):285–305, 2004.

[6] Wieb Bosma, John Cannon, and Catherine Playoust. The Magma algebra system. I. The user language. *J. Symbolic Comput.*, 24(3-4):235–265, 1997. Computational algebra and number theory (London, 1993).

[7] Fabrice Boudot, Pierrick Gaudry, Aurore Guillevic, Nadia Heninger, Thomé Emmanuel, and Paul Zimmermann. 795-bit factoring and discrete logarithms. NMBRTHRY list, 02/12/2019.

[8] Fabrice Boudot, Pierrick Gaudry, Aurore Guillevic, Nadia Heninger, Thomé Emmanuel, and Paul Zimmermann. Factorization of RSA-250. NMBRTHRY list, 28/02/2020.

[9] A. E. Brouwer, R. Pellikaan, and E. R. Verheul. Doing More with Fewer Bits. In Kwok-Yan Lam, Eiji Okamoto, and Chaoping Xing, editors, *Advances in Cryptology - ASIACRYPT'99*, pages 321–332, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[10] Don Coppersmith. Evaluating Logarithms in GF(2^n). In *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, STOC '84, pages 201–207, New York, NY, USA, 1984. ACM.

[11] Don Coppersmith. Fast Evaluation of Logarithms in Fields of Characteristic Two. *IEEE Trans. Inf. Theor.*, 30(4):587–594, 1984.

[12] Jean Marc Couveignes and Reynald Lercier. Elliptic periods for finite fields. *Finite Fields and Their Applications*, 15(1):1–22, 2009.

[13] Claus Diem. On the discrete logarithm problem in elliptic curves. *Compositio Mathematica*, 147(1):75 – 104, 2011.

[14] Claus Diem. On the discrete logarithm problem in elliptic curves II. *Algebra Number Theory*, 7(6):1281–1323, 2013.

[15] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22:644–654, 1976.

[16] T. Elgamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.

[17] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases (F_4). *J. Pure Appl. Algebra*, 139 (1-3):61–88, 1999.

[18] G. Frey and H. Ruck. A remark considering m-divisibility in the divisor class group of curves. *Mathematics of Computation*, 1994.

[19] Steven D. Galbraith and Shishay W. Gebregiyorgis. Summation Polynomial Algorithms for Elliptic Curves in Characteristic Two. In Willi Meier and Debdeep Mukhopadhyay, editors, *Progress in Cryptology – INDOCRYPT 2014*, pages 409–427, Cham, 2014. Springer International Publishing.

[20] Steven D. Galbraith, Florian Hess, and Nigel P. Smart. Extending the GHS Weil Descent Attack. In Lars R. Knudsen, editor, *Advances in Cryptology—EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 29–44. Springer Verlag, April 28–May 2, 2002.

[21] Pierrick Gaudry. Index Calculus for Abelian Varieties of Small Dimension and the Elliptic Curve Discrete Logarithm Problem. *J. Symb. Comput.*, 44(12):1690–1702, December 2009.

[22] Pierrick Gaudry, Florian Hess, and Nigel P. Smart. Constructive and Destructive Facets of Weil Descent on Elliptic Curves. *J. Cryptology*, 15(1):19–46, 2002.

[23] Carl F. Gauss. *Disquisitiones Arithmeticae*. Translated by Arthur A. Clarke. Yale University Press, 1965.

[24] Faruk Göloglu and Antoine Joux. A simplified approach to rigorous degree 2 elimination in discrete logarithm algorithms. *Math. Comput.*, 88(319):2485–2496, 2019.

[25] Faruk Göloğlu, Robert Granger, Gary McGuire, and Jens Zumbrägel. On the Function Field Sieve and the Impact of Higher Splitting Probabilities - Application to Discrete Logarithms in $\mathbb{F}_{2^{1971}}$ and $\mathbb{F}_{2^{3164}}$. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology—CRYPTO 2013*, volume 8043 of *LNCS*, pages 109–128. Springer, 2013.

[26] Faruk Göloğlu, Robert Granger, Gary McGuire, and Jens Zumbrägel. Solving a 6120-bit DLP on a Desktop Computer. In *Selected Areas in Cryptography—SAC 2013*, volume 8282 of *LNCS*, pages 136–152. Springer, 2014.

[27] Daniel M. Gordon and Kevin S. McCurley. Massively Parallel Computation of Discrete Logarithms. In Ernest F. Brickell, editor, *Advances in Cryptology — CRYPTO' 92*, pages 312–323, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg.

[28] R. Granger and F. Vercauteren. On the Discrete Logarithm Problem on Algebraic Tori. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, pages 66–85, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[29] Robert Granger, Thorsten Kleinjung, Arjen K. Lenstra, Benjamin Wesolowski, and Jens Zumbrägel. Discrete Logarithms in GF(2^{30750}). NMBRTHRY list, 10/07/2019.

[30] Robert Granger, Thorsten Kleinjung, and Jens Zumbrägel. Breaking '128-bit Secure' Supersingular Binary Curves - (Or How to Solve Discrete Logarithms in $\mathbb{F}_{2^{4\cdot 1223}}$ and $\mathbb{F}_{2^{12\cdot 367}}$). In *Advances in Cryptology—CRYPTO 2014*, volume 8617 of *LNCS*, pages 126–145. Springer, 2014.

[31] Robert Granger, Thorsten Kleinjung, and Jens Zumbrägel. Indiscreet logarithms in finite fields of small characteristic. *Advances in Mathematics of Communications*, 12(2):263–286, 2018.

[32] Robert Granger, Thorsten Kleinjung, and Jens Zumbrägel. On the discrete logarithm problem in finite fields of fixed characteristic. *Transactions of the American Mathematical Society*, 370:3129–3145, 2018.

[33] Robert Granger, Thorsten Kleinjung, and Jens Zumbrägel. On the Powers of 2. Available from `eprint.iacr.org/2014/300`, 29th Apr 2014.

[34] Takuya Hayashi, Takeshi Shimoyama, Naoyuki Shinohara, and Tsuyoshi Takagi. Breaking Pairing-Based Cryptosystems Using η T Pairing over GF(3^{97}). In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, volume 7658 of *Lecture Notes in Computer Science*, pages 43–60. Springer, 2012.

[35] Takuya Hayashi, Naoyuki Shinohara, Lihua Wang, Shin'ichiro Matsuo, Masaaki Shirase, and Tsuyoshi Takagi. Solving a 676-Bit Discrete Logarithm Problem in GF(3^{6n}). In Phong Q. Nguyen and David Pointcheval, editors, *Public Key Cryptography – PKC 2010*, pages 351–367, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[36] Florian Hess. The GHS Attack Revisited. In Eli Biham, editor, *Advances in Cryptology—EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 374–387. Springer Verlag, May 4–8, 2003.

[37] Jeremy Horwitz and Ramarathnam Venkatesan. Random Cayley Digraphs and the Discrete Logarithm. In Claus Fieker and David R. Kohel, editors, *Algorithmic Number Theory*, pages 416–430, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[38] Antoine Joux. Faster Index Calculus for the Medium Prime Case. Application to 1175-bit and 1425-bit Finite Fields. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology—EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 177–193. Springer, 2013.

[39] Antoine Joux. A New Index Calculus Algorithm with Complexity $L(1/4 + o(1))$ in Small Characteristic. In Tanja Lange, Kristin Lauter, and Petr Lisoněk, editors, *Selected Areas in Cryptography—SAC 2013*, volume 8282 of *LNCS*, pages 355–379. Springer, 2014.

[40] Antoine Joux and Reynald Lercier. The Function Field Sieve in the Medium Prime Case. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006*, pages 254–270, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[41] Antoine Joux, Andrew Odlyzko, and Cécile Pierrot. *The Past, Evolving Present, and Future of the Discrete Logarithm*, pages 5–36. Springer International Publishing, Cham, 2014.

[42] Antoine Joux and Cécile Pierrot. Improving the Polynomial time Precomputation of Frobenius Representation Discrete Logarithm Algorithms. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014*, pages 378–397, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[43] Antoine Joux and Cécile Pierrot. Technical history of discrete logarithms in small characteristic finite fields - The road from subexponential to quasi-polynomial complexity. *Des. Codes Cryptogr.*, 78(1):73–85, 2016.

[44] Antoine Joux and Cécile Pierrot. Algorithmic aspects of elliptic bases in finite field discrete logarithm algorithms. Preprint. Available from: *https://eprint.iacr.org/2019/782*, 2019.

[45] 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 David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology—EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 9–26. Springer Verlag, April 15–19, 2012.

[46] Taechan Kim and Razvan Barbulescu. Extended Tower Number Field Sieve: A New Complexity for the Medium Prime Case. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016*, pages 543–571, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[47] Thorsten Kleinjung, Kazumaro Aoki, Jens Franke, Arjen K. Lenstra, Emmanuel Thomé, Joppe W. Bos, Pierrick Gaudry, Alexander Kruppa, Peter L. Montgomery, Dag Arne Osvik, Herman te Riele, Andrey Timofeev, and Paul Zimmermann. Factorization of a 768-Bit RSA Modulus. In Tal Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, pages 333–350, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[48] Thorsten Kleinjung, Joppe W. Bos, and Arjen K. Lenstra. Mersenne Factorization Factory. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014*, pages 358–377, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[49] Thorsten Kleinjung, Claus Diem, Arjen K. Lenstra, Christine Priplata, and Colin Stahlke. Computation of a 768-Bit Prime Field Discrete Logarithm. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017*, pages 185–201, Cham, 2017. Springer International Publishing.

[50] Thorsten Kleinjung and Benjamin Wesolowski. Discrete logarithms in quasi-polynomial time in finite fields of fixed characteristic. To appear in *Journal of the American Mathematical Society*.

[51] A. A. Klyachko. On the Rationality of Tori with Cyclic Splitting Field (Russian). *Arithmetic and Geometry of Varieties*, pages 73–78, 1988.

[52] Maurice Kraitchik. *Théorie des nombres*, volume 1. Paris: Gauthier-Villars, 1922.

[53] Maurice Kraitchik. *Recherches sur la théorie des nombres*, volume 1. Paris: Gauthier-Villars, 1924.

[54] Thomas S. Kuhn. *The Structure of Scientific Revolutions, 3rd ed.* University of Chicago Press, 1996.

[55] *Personal communication between Robert Granger and Arjen K. Lenstra.*, 2013.

[56] A. K. Lenstra and H. W. Lenstra Jr. Algorithms in number theory. Technical Report 87-008, University of Chicago, 1987.

[57] A. K. Lenstra and E. R. Verheul. An Overview of the XTR Public Key System. In *Public Key Cryptography and Computational Number Theory*, pages 151–180. Verlages Walter de Gruyter, 2001.

[58] Arjen K. Lenstra. Using cyclotomic polynomials to construct efficient discrete logarithm cryptosystems over finite fields. In Vijay Varadharajan, Josef Pieprzyk, and Yi Mu, editors, *Information Security and Privacy*, pages 126–138, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.

[59] Arjen K. Lenstra and Eric R. Verheul. The XTR Public Key System. In Mihir Bellare, editor, *Advances in Cryptology — CRYPTO 2000*, pages 1–19, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[60] Guido Lido. *Discrete logarithm over finite fields of small characteristic*. Master's thesis, Universita di Pisa, 2016.

[61] R. Lovorn. *Rigorous Subexponential Algorithms for Discrete Logarithms over Finite Fields*. Ph. D. thesis, University of Georgia, 1992.

[62] Kevin S. McCurley. The discrete logarithm problem. In *Cryptology and computational number theory, Proc. Symp. in Applied Mathematics*, volume 42, pages 49–74. American Mathematical Society, 1990.

[63] A. Menezes, T. Okamoto, and S. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transaction on Information Theory*, 39:1639–1646, 1993.

[64] Alfred Menezes and Minghua Qu. Analysis of the Weil Descent Attack of Gaudry, Hess and Smart. In David Naccache, editor, *CT-RSA 2001*, volume 2020 of *LNCS*, pages 308–318. Springer Verlag, April 8–12, 2001.

[65] Alfred Menezes and Yihong Wu. The Discrete Logarithm Problem in GL(n, q). *Ars Comb.*, 47, 1997.

[66] Victor S. Miller. The Weil Pairing, and Its Efficient Calculation. *J. Cryptology*, 17(4):235–261, September 2004.

[67] V. I. Nechaev. On the complexity of a deterministic algorithm for a discrete logarithm. *Mat. Zametki*, 55:91–101, 1994.

[68] *NumberTheoryList: https://listserv.nodak.edu/cgi-bin/wa.exe?A0=NMBRTHRY.*

[69] A. M. Odlyzko. Discrete logarithms in finite fields and their cryptographic significance. In Thomas Beth, Norbert Cot, and Ingemar Ingemarsson, editors, *Advances in Cryptology*, pages 224–314, Berlin, Heidelberg, 1985. Springer Berlin Heidelberg.

[70] R.W.K. Odoni, V. Varadharajan, and P.W. Sanders. Public key distribution in matrix rings. *Electronics Letters*, 20(9):386 – 387, 1984.

[71] Stephen C. Pohlig and Martin E. Hellman. An improved algorithm for computing logarithms over gf(p) and its cryptographic significance (corresp.). *IEEE Trans. Inf. Theory*, 24(1):106–110, 1978.

[72] John M. Pollard. Monte Carlo Methods for Index Computation (mod p). *Mathematics of Computation*, 32:918–924, 1978.

[73] Karl Rubin and Alice Silverberg. Torus-Based Cryptography. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003*, pages 349–365, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[74] C. P. Schnorr. Efficient signature generation by smart cards. *J. Cryptology*, 4:161–174, 1991.

[75] RenÃ© Schoof. Counting points on elliptic curves over finite fields. *Journal de thÃ©orie des nombres de Bordeaux*, 7(1):219–254, 1995.

[76] I. A. Semaev. Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p. *Math. Comp.*, 67(221):353–356, 1998.

[77] Igor Semaev. Summation polynomials and the discrete logarithm problem on elliptic curves. Cryptology ePrint Archive, Report 2004/031, 2004.

[78] Peter W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM J. Comput.*, 26(5):1484–1509, October 1997.

[79] Victor Shoup. Lower Bounds for Discrete Logarithms and Related Problems. In Walter Fumy, editor, *Advances in Cryptology — EUROCRYPT '97*, pages 256–266, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.

[80] Nigel P. Smart. The Discrete Logarithm Problem on Elliptic Curves of Trace One. *J. Cryptology*, 12(3):193–196, 1999.

[81] Peter Smith and Christopher Skinner. A public-key cryptosystem and a digital signature system based on the lucas function analogue to discrete logarithms. In Josef Pieprzyk and Reihanah Safavi-Naini, editors, *Advances in Cryptology — ASIACRYPT'94*, pages 355–364, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg.

[82] *Personal communication between Robert Granger and Martijn Stam*, 2005.

[83] Marten van Dijk, Robert Granger, Dan Page, Karl Rubin, Alice Silverberg, Martijn Stam, and David Woodruff. Practical Cryptography in High Dimensional Tori. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, pages 234–250, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[84] Paul C. van Oorschot and Michael J. Wiener. Parallel Collision Search with Cryptanalytic Applications. *J. Cryptology*, 12(1):1–28, 1999.

[85] V. E. Voskresenskiĭ. *Algebraic Groups and Their Birational Invariants*. Translations of Mathematical Monographs, **179**, American Mathematical Society, 1998.