Robert Granger
School of Computer and Communication Sciences
École polytechnique fédérale de Lausanne
robert.granger@epfl.ch

Indiscreet discrete logarithms

In 2013 and 2014 a revolution took place in the understanding of the discrete logarithm problem (DLP) in finite fields of small characteristic. Consequently, many cryptosystems based on cryptographic pairings were rendered completely insecure, which serves as a valuable reminder that long-studied so-called hard problems may turn out to be far easier than initially believed. In this article, Robert Granger gives an overview of the surprisingly simple ideas behind some of the breakthroughs and the many computational records that have so far resulted from them.

By way of motivation, we begin with the landmark work of Diffie and Hellman, who in 1976 introduced the following very well known key-agreement protocol, which allows two parties — referred to as Alice and Bob — to agree a shared secret key over an insecure channel which can then be used for secure communications between them [13]. To do so, Alice and Bob agree in advance on two public system parameters: p a prime integer and g a primitive root modulo p. We denote by $\mathbb{F}_p$ the finite field of p elements, represented as usual as the quotient $\mathbb{Z}/p\mathbb{Z}$ with coset representatives always in $\{0,\ldots,p-1\}$; g thus generates the multiplicative group $\mathbb{F}_p^\times$. To establish a shared key, Alice picks a secret integer $a \in \{1,\ldots,p-1\}$, computes g^a and sends this to Bob. Likewise, Bob picks a secret integer $b \in \{1,\ldots,p-1\}$, computes g^b and sends this to Alice. Using their respective secrets Alice and Bob can both compute the shared key

$$g^{ab} = (g^b)^a = (g^a)^b.$$

In order for this key to be secure, it is necessary that it is hard to compute g^{ab} from the public information (g, g^a, g^b), for some notion of 'hard' that is discussed later on. The task of doing so is known as the *Diffie–Hellman problem* (DHP). One way to solve the DHP is to recover a from (g, g^a) and then compute the shared key as Alice does (or equivalently recover b from (g, g^b) and compute the shared key as Bob does). Hence, it is also necessary that the problem of recovering a from (g, g^a) — known as the *discrete logarithm problem* (DLP), since it is the inverse of exponentiation — is hard to solve too. Note that finite cyclic groups other than $\mathbb{F}_p^\times$ may also be used to instantiate this protocol. We therefore formalise the DLP more generally with the following.

Definition 1. Given a finite cyclic group $(G, \cdot)$, a generator $g \in G$ and another group element $h \in G$, the DLP is the problem of finding an integer t such that $h = g^t$. The integer t — denoted by $\log_g h$ — is uniquely determined modulo the group order and is called the *discrete logarithm* of h with respect to the base g.

Although the DHP is clearly reducible to the DLP, in the sense that an algorithm to solve the latter provides an algorithm to solve the former, it is not known whether the converse holds in general; there are however several positive results in this direction [7, 39, 40]. Since there are no known algorithms which solve the DHP directly, research on the hardness of the DHP has focused almost entirely on the hardness of the DLP, which explains its cryptographic importance. We note that while necessary, the hardness of the DHP is by no means sufficient to ensure the security of the protocol, since several other issues must also be addressed, see [50, Chapter 18], for example.

The above key-agreement protocol can easily be extended to a public-key encryption scheme, in which Alice can send a message m to Bob over an insecure channel, without having first agreed on a secret key with him [14]. In particular, Bob chooses a key-pair (b, g^b) consisting of a private key b, which is kept secret, and a public key g^b, which is published. To encrypt a message $m \in \{1,\ldots,p-1\}$ to Bob, Alice chooses a random $a \in \{1,\ldots,p-1\}$ and sends to him $(c_1, c_2) = (g^a, m(g^b)^a)$. Bob decrypts by computing $m = c_2/c_1^b$, using his private key b. One can also obtain a digital signature scheme in a similar manner [14] and there are a large number of variations and cryptosystems with more complex properties, all of which rely on the hardness of the DLP in one form or another. Hence, having groups in which the DLP is hard is essential.

Pairing-based cryptography

An interesting family of protocols which are pertinent to this story are those which arose from the invention of pairing-based cryptography in 2000, allowing cryptographic functionalities such as identity-based non-interactive key distribution [47], one-round tripartite key-agreement [25] and identity-based encryption [8] (and later several hundred others). All of these rely on the existence of certain non-degenerate efficiently computable bilinear maps, known as pairings. Such a map has the form

$$e : G_1 \times G_2 \to G_3,$$

where G_1 and G_2 are abelian groups of exponent $l \in \mathbb{N}$, which by convention are written in additive notation with identity

element 0, and G_3 is a cyclic group of order l, written in multiplicative notation with identity element 1.

The non-degeneracy condition is that for all $P \in G_1 \backslash \{0\}$ there is a $Q \in G_2$ such that $e(P,Q) \neq 1$, and for all $Q \in G_2 \backslash \{0\}$ there is a $P \in G_1$ such that $e(P,Q) \neq 1$. The bilinearity condition is that for all $P, P' \in G_1$ and all $Q, Q' \in G_2$ one has

$$e(P+P',Q) = e(P,Q)\,e(P',Q),$$
$$e(P,Q+Q') = e(P,Q)\,e(P,Q'),$$

which implies that for all $P \in G_1$, all $Q \in G_2$ and all $a, b \in \mathbb{Z}$ one has

$$e(aP,bQ) = e(P,Q)^{ab}. \tag{1}$$

Although such maps arise naturally from the Tate and Weil pairings on arbitrary abelian varieties over local or finite fields, for efficiency purposes they are usually instantiated using elliptic curves over finite fields. In this case, for the Tate pairing [16] we have $G_1 = E(\mathbb{F}_q)[l]$, the group of l-torsion points on an elliptic curve E defined over $\mathbb{F}_q$, with $q = p^r$ and l coprime to p. G_3 is the group μ_l of l-th roots of unity in $\overline{\mathbb{F}}_q$, which embeds into $\mathbb{F}_{q^n}^{\times}$, with n the order of q modulo l, also known as the embedding degree. Finally, G_2 is the quotient group $E(\mathbb{F}_{q^n})[l]/lE(\mathbb{F}_{q^n})[l]$, whose coset representatives we do not describe here. For the definition of the pairing itself and other technical conditions we refer the interested reader to [5, Chapter 9], which contains a comprehensive introduction to the area.

The property (1) was originally exploited using the linearity of the pairing in the first argument only, in order to transfer a DLP from an elliptic curve group to a DLP in an extension of the underlying base field [41]. Indeed, if the input DLP in G_1 is (P, aP), one selects an appropriate $Q \in G_2$ such that $e(P,Q) \neq 1$ — which exists by the non-degeneracy condition — and computes $g = e(P,Q)$ and $g^a = e(aP,Q)$. The *raison d'être* of this transfer is that in general the best algorithms for solving the finite field DLP have lower complexity than the best algorithms for solving the elliptic curve DLP, so even though the inputs to each problem have different sizes, namely $n\log_2 q$ and $\log_2 q$, if the embedding degree is small enough then the transferred DLP will be easier to solve.

However, it is the full bilinearity that enables interesting cryptographic applications. Such applications require that the embedding degree is small enough that the pairing itself is efficiently computable, but large enough that the DLP in $\mathbb{F}_{q^n}^{\times}$ is hard. Therefore, while as with the key-agreement protocol each pairing-based protocol comes with a set of problems other than the DLP which must be hard in order for it to be secure, all are vulnerable to developments in discrete logarithm algorithms for the finite field DLP.

Hardness of the DLP

Let $(G,\cdot)$ be a finite cyclic group of known order N in which the group operation is assumed to be computable in unit time, and let g be a generator of G. First observe that exponentiation in G can be performed in time polynomial in the bitlength of N, i.e., $\lceil \log_2 N \rceil$, since one can take the binary expansion of an exponent e and compute g^e via the square-and-multiply algorithm or its variants. Hence, for a group G to be useful for discrete logarithm-based cryptography, discrete logarithms should not be computable in polynomial time, and should preferably be *much* harder to compute.

Note that there are groups in which the DLP is easy. For instance, if $G = (\mathbb{Z}/N\mathbb{Z},+)$ and $1 \leq g \leq N-1$ is coprime to N, then g is a generator and 'exponentiation by e' is just $eg \pmod N$. Thus the discrete logarithm of any element h is just $hg^{-1} \pmod N$. As all cyclic groups of order N are isomorphic to $(\mathbb{Z}/N\mathbb{Z},+)$, one might expect the DLP in such groups to be easy. However, since computing the image of such an isomorphism requires solving a DLP with respect to a generator, this need not be so. Indeed, the representation of group elements may obscure the cyclic structure to varying degrees, which dictates the apparent hardness of the DLP in each group.

If one insists upon not exploiting any information regarding the representation of group elements, i.e., the worst case from a cryptanalytic perspective, then the DLP in such groups can be analysed using the *generic group model*. This model stipulates that group elements are represented using random encodings, while the group operation is performed using an oracle which takes as input the encodings of two elements and outputs the encoding of the group operation applied to the input elements. In this case it can be shown that the DLP requires $\Omega(\sqrt{N})$ oracle calls in order to be solved with high probability, i.e., at least $c\sqrt{N}$ for some constant $c > 0$, for N sufficiently large [43, 49]. Since this is exponential in the size of the problem, namely $\log N$, we see that in the ideal case from a cryptography perspective, the DLP is exponentially hard. Of course, the DLP can be no harder than exponential since a naive enumeration of powers of the generator will solve it too. Note that a square root complexity can be achieved using a standard time-space trade-off known as Baby Step/Giant Step, or a memory efficient version based on random walks, due to Pollard [46]. Further note that if the prime factorisation of N is known then the DLP can always be reduced to a set of DLPs in prime order subgroups, by projecting into them via exponentiation by their cofactors, using a form of Hensel lifting for prime-power order subgroups, and applying the Chinese remainder theorem [45]. So in practice the DLP can be assumed to be in a group of prime order. These results imply that in order to solve the DLP in a time faster than exponential, one *must* exploit representational properties of elements of the group. In some scenarios this is possible, as for multiplicative groups of finite fields for example, while in others it is apparently not, as for the group of $\mathbb{F}_p$-rational points on a suitably chosen elliptic curve. The latter explains the popularity of elliptic curve cryptography, first proposed in 1985 independently by Miller [42] and Koblitz [33], which essentially achieves optimal security per bit as a result.

In general, the hardness of a DLP is measured by the complexity of the fastest algorithm known to solve it. The following function is often used in this regard:

$$L_N(\alpha,c) = \exp\big((c+o(1))(\log N)^{\alpha}(\log\log N)^{1-\alpha}\big),$$

where $\alpha \in [0,1]$, $c > 0$, $\log$ denotes the natural logarithm and the $o(1)$ denotes a function that tends to zero as $N \to \infty$. Observe that $L_N(0,c) = (\log N)^{c+o(1)}$, which thus represents algorithms which run in polynomial time, while $L_N(1,c) = N^{c+o(1)}$ represents algorithms which run in exponential time. For $0 < \alpha < 1$, the function $L_N(\alpha,c)$ represents algorithms which are said to run in *subexponential* time. As is customary we often omit the subscript N and the constant c when convenient.

The first subexponential algorithm for the finite field DLP was shown by Adleman in 1979 to have *heuristic* complexity $L(1/2)$ [1], see the next section. It is termed heu-

ristic because the analysis relied on unproven assumptions. In 1984 Coppersmith proposed the first (again, heuristic) $L(1/3)$ algorithm for the DLP in binary fields, i.e., in $\mathbb{F}_{2^n}$ [11], which generalises to arbitrary families of extension fields $\mathbb{F}_{q^n}$ with a fixed base field, also referred to as small characteristic fields. The later development of the number field sieve [37] and the function field sieve [2,3,27,28] led to heuristic $L(1/3)$ algorithms for all finite fields [20,29]. Between 1984 and 2013, no algorithms went below the $L(1/3)$ barrier, although the far less important constant c was occasionally lowered for some (q,n) families. It thus seemed plausible that this was the natural complexity of the finite field DLP, and cryptographers were fairly confident that it would not be broken any time soon, excepting of course the possible development of a large-scale quantum computer, which threatens all DLPs as well as the integer factorisation problem, thanks to Shor's algorithm from 1994 [48].

Each of the subexponential algorithms exploits the property that field elements, when viewed as elements in the parent ring, can be factored into a product of irreducible elements. Thanks to this property a very natural and broadly applicable framework first discussed by Kraitchik in the 1920s [34,35] can be applied, namely, *index calculus*, which we now introduce.

Index calculus

The term index calculus — which literally means 'calculating the index' — originates from the at least two-centuries-old name used by Gauss for the discrete logarithm of an integer modulo p relative to a primitive root, namely, the index [17, art. 57–60]. For a group G written multiplicatively and a generator g, let $h \in G$ be an element whose discrete logarithm with respect to g is to be computed. The index calculus method consists of the following three steps.

1. *Relation generation:* Choose a subset $\mathcal{F} \subset G$ which we call the factor base, and find multiplicative relations between its elements. Observe that each such relation provides a linear equation in the logarithms of the factor base elements with respect to any generator, modulo $|G|$.
2. *Linear algebra:* Once at least $|\mathcal{F}|$ linearly independent equations between logarithms of elements of $\mathcal{F}$ have been generated, obtain these logarithms by solving the corresponding linear system.
3. *Individual logarithms:* Find an expression for h as a product of factor base elements, for example by computing hg^e for random e until this factors completely over $\mathcal{F}$, from which one can easily deduce $\log_g h$.

The elements of $\mathcal{F}$ are usually chosen to be the set of 'prime' elements whose 'norm' is less than some bound (for some notions of prime and norm), since such a choice generates the maximum number of elements of G amongst all sets of the same cardinality. How steps (1) and (3) are performed in practice depends very much on the group in question and the ingenuity of the cryptanalyst. In order to illustrate the method we now present a very simple example.

Example. Let $p = 1009$. Then $g = 11$ is a generator of $G = \mathbb{F}_p^\times$. Let $\mathcal{F} = \{2,3,5,7\}$. Relations are obtained by computing $g^e \bmod p$ for random $e \in \{1,\ldots,p-1\}$ and then using trial division to check whether this integer is a product of the primes in $\mathcal{F}$. The following relations were quickly obtained:

$$\begin{aligned} 11^{796} \bmod p &= 15 = 3 \cdot 5, \\ 11^{678} \bmod p &= 315 = 3^2 \cdot 5 \cdot 7, \\ 11^{992} \bmod p &= 63 = 3^2 \cdot 7, \\ 11^{572} \bmod p &= 10 = 2 \cdot 5. \end{aligned}$$

Note that the factorisations occur in the parent ring $\mathbb{Z}$ of $\mathbb{F}_p \cong \mathbb{Z}/p\mathbb{Z}$. These relations yield the following equations mod $p-1$:

$$\begin{aligned} 796 &\equiv \log_{11} 3 + \log_{11} 5, \\ 678 &\equiv 2\log_{11} 3 + \log_{11} 5 + \log_{11} 7, \\ 992 &\equiv 2\log_{11} 3 + \log_{11} 7, \\ 572 &\equiv \log_{11} 2 + \log_{11} 5. \end{aligned}$$

Writing this linear system in matrix form we have:

$$\begin{bmatrix} 796 \\ 678 \\ 992 \\ 572 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 2 & 1 & 1 \\ 0 & 2 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \log_{11} 2 \\ \log_{11} 3 \\ \log_{11} 5 \\ \log_{11} 7 \end{bmatrix}.$$

The matrix is invertible mod $p-1$ and solving the system yields the solutions: $\log_{11} 2 = 886$, $\log_{11} 3 = 102$, $\log_{11} 5 = 694$ and $\log_{11} 7 = 788$, as one can easily verify.

For an individual logarithm, the first non-trivial case is $h = 13$, for which $\log_{11} 13$ can be computed as follows. Testing random e, we quickly find that

$$\begin{aligned} hg^e &= 13 \cdot 11^{53} \bmod p \\ &= 720 = 2^4 \cdot 3^2 \cdot 5, \end{aligned}$$

and hence

$$\begin{aligned} \log_{11} 13 &= (4\log_{11} 2 + 2\log_{11} 3 \\ &\quad + \log_{11} 5 - 53) \bmod p - 1 \\ &= 357. \end{aligned}$$

A basic question arising from this approach is how large should the factor base be in order to optimise the running time, as $p \to \infty$? This depends on the density of smooth numbers, whose definition we now recall.

Definition 2. A positive integer is said to be B-smooth if all of its prime divisors are at most B.

The following result on the asymptotic density of smooth numbers amongst the integers is due to Canfield, Erdős and Pomerance [9].

Theorem 1. *A uniformly random integer in $\{1,\ldots,M\}$ is B-smooth with probability*

$$P = u^{-u(1+o(1))}, \quad \textit{where } u = \frac{\log M}{\log B},$$

provided that $3 \leq u \leq (1-\varepsilon)\frac{\log M}{\log\log M}$ for some $\varepsilon > 0$.

Clearly, the larger the factor base then the higher the probability that a uniformly random element of $\mathbb{F}_p^\times$, viewed as an integer, is smooth with respect to the factor base. However, one then requires more relations. On the other hand, a smaller factor base means fewer relations are needed, but each is harder to find. Theorem 1 indicates how to optimise this trade-off; in particular, using the very aptly defined function $L_N(\alpha,c)$ it implies the following.

Corollary 1. *Let $M = L_N(2\alpha,\mu)$ and $B = L_N(\alpha,\beta)$. Then the expected number of trials until a uniformly random number in $\{1,\ldots,M\}$ is B-smooth is $L_N\left(\alpha,\frac{\alpha\mu}{\beta}\right)$.*

For the $\mathbb{F}_p^\times$ index calculus algorithm we have $M = N = L(1,1)$. Corollary 1 implies that we should set the smoothness bound to be $B = L(1/2,\beta)$ for some unknown $\beta > 0$. Since we need about $|F| \approx B/\log B \leq B$ relations, the estimated running time is

$$L(\tfrac{1}{2},\beta)\cdot L(\tfrac{1}{2},\tfrac{1}{2\beta}) = L(\tfrac{1}{2},\beta+\tfrac{1}{2\beta}).$$

This complexity is minimised for $\beta = 1/\sqrt{2}$, resulting in a running time of $L(1/2,\sqrt{2})$ for step 1. For step 2, as the matrices generated are incredibly sparse, i.e., have very few non-zero entries, by using either Lanczos' algorithm [36] or Wiedemann's algorithm [52], the complexity is about $B^2 = L(1/2,2\beta) = L(1/2,\sqrt{2})$ as well. Step 3 is obviously of lower complexity since only one relation is needed.

For fixed q and $n \to \infty$, one needs to employ a different, but equally natural notion of smoothness in order to apply an analogous algorithm to solve the DLP in $\mathbb{F}_{q^n}^{\times}$; note that the norm of an element in this scenario is its degree.

Definition 3. An element in $\mathbb{F}_q[x]$ of positive degree is said to be b-smooth if all of its irreducible factors are of degree at most b.

The following result on the asymptotic density of smooth polynomials amongst those of the same degree is due to Odlyzko [44] and Lovorn [38].

Theorem 2. *A uniformly random polynomial $f \in \mathbb{F}_q[X]$ of degree m is b-smooth with probability $P = u^{-u(1+o(1))}$, where $u = \frac{m}{b}$, provided that $m^{1/100} \leq b \leq m^{99/100}$.*

With this notion of smoothness and a corollary to Theorem 2 analogous to Corollary 1 but with N now q^n rather than p, the algorithm given for $\mathbb{F}_p^{\times}$ applies to $\mathbb{F}_{q^n}^{\times}$ *mutatis mutandis* and one can show that it also has a running time of $L(1/2,\sqrt{2})$. Note that as described above the algorithm is heuristic since there is no guarantee that the relations generated produce a linear system of full rank. However, it can be made rigorous by using an elementary argument due to Enge and Gaudry [15].

Obtaining faster algorithms

The previous analysis demonstrates that when elements of the field in question are generated uniformly at random, an $L(1/2)$ complexity is optimal for the index calculus algorithm employed. To obtain algorithms of better complexity, there are (at least) two approaches that one could attempt to employ.

The first approach is to generate relations between elements of smaller norm than before, and for complexity analysis purposes assume that any such subset of elements has the same smoothness density as uniformly random elements of that norm, which is a *smoothness heuristic*. This is precisely what the $L(1/3)$ algorithms do. In particular, elements of norm $\log(L(2/3))$ are produced and the factor base consists of elements of norm $\log(L(1/3))$. Applying Corollary 1 for the integers (or its analogue for polynomials) once again means that the running times of steps 1 and 2 are both $L(1/3)$. Since the factor base is now smaller, in order to obtain an $L(1/3)$ complexity for step 3 one needs to employ a *descent* strategy. A descent begins by expressing the target element as a product of elements of lower norm, mod p or mod the field-defining polynomial, rather than in $\mathbb{N}$ or the polynomial ring, respectively. When such an expression has been obtained we say that the element has been *eliminated*, since one need no longer compute its logarithm directly; only the logarithms of the elements in the obtained product are needed. By iteratively eliminating all of the elements featured in the product one obtains expressions for the target element of lower and lower norm, until finally one has an expression over the factor base, from which one can easily deduce the target logarithm. Subject to the above smoothness heuristic, techniques to do this have an $L(1/3)$ complexity, but with a smaller c than for steps 1 and 2.

The second approach — which seems not to have even been appreciated as a possibility prior to 2013, perhaps due to the desire to assume the above smoothness heuristic for the sake of the complexity analysis — is to generate relations between elements which have *higher smoothness probabilities* than uniformly random elements of the same norm. While no method is known for achieving this over the integers — and thus for the DLP in $\mathbb{F}_p^{\times}$ — the breakthrough results from 2013 onwards all came about because two ways to do this usefully for polynomial rings — and thus for the DLP in small characteristic extension fields — were independently discovered at essentially the same time. The first was due to Göloğlu, Granger, McGuire and Zumbrägel (referred to hereafter as GGMZ) [18], while the second was due to Joux [26]. Although the ingredients of the two methods are somewhat different, they may be viewed as being essentially isomorphic. Both methods produce relations for a factor base whose size is only *polynomial* in the bitlength of the field in question, in *polynomial time*, as the smoothness probability is exponentially larger than before; indeed, such relations are smooth by construction. Since the factor base in this scenario is very small, usually consisting of degree one elements over a suitable base field, step 3 must now descend further which leads to complexities worse than $L(1/3)$ when using the old, or 'classical' techniques, with the complexity being highest for degree two elimination. Hence, in order to make these insights fully applicable, new descent strategies were also needed.

GGMZ proposed a polynomial-time method for eliminating degree two elements on the fly, i.e., on an element by element basis as required, while Joux proposed a polynomial time method for computing the logarithms of degree two elements in batches, as well as a technique to eliminate elements of very small degree, which leads to a heuristic $L(1/4+o(1))$ algorithm. The two degree two elimination methods led respectively to two very different *quasi-polynomial time* algorithms for solving the DLP in small characteristic extension fields, this complexity arising from the descent step. In particular, Joux's approach led to the first such algorithm in mid 2013 and is due to Barbulescu, Gaudry, Joux and Thomé [4], while Göloğlu et al.'s approach led to the second in early 2014 and is due to Granger, Kleinjung and Zumbrägel (referred to hereafter as GKZ), appearing in the preprint [22] and its final version [23]. Since the GKZ algorithm is somewhat simpler and far more practical than the former, and is rigorously proven for an infinite family of extensions of every base field, we detail only this one in this article.

The GKZ algorithm

Unlike in the $L(1/2)$ algorithms, how the target field is represented is now of paramount importance. The GKZ algorithm applies to fields of the form $\mathbb{F}_{q^{kn}}$, with $18 \leq k = o(\log q)$ and $n \approx q$ (see Theorem 5 for additional technical conditions). Any small characteristic field $\mathbb{F}_{q'^n}$ can be embedded into such a field by setting $q = q'^{\lceil \log_{q'} n \rceil}$, thereby increasing the extension degree by a factor of $k\lceil \log_{q'} n \rceil$. The use of such an embedding does not significantly affect the

algorithm's resulting complexity. The field setup used in [23] can be either from [18] (with a small modification from [21]), or from [26], both of which may be seen in the context of the Joux–Lercier doubly-rational function field sieve variant [28].

Let $h_0, h_1 \in \mathbb{F}_{q^k}[X]$ be coprime and of degree $d_h \leq 2$, such that $h_1(X)X^q - h_0(X) \equiv 0 \pmod{I}$ for an irreducible degree n polynomial $I \in \mathbb{F}_{q^k}[X]$. Heuristically, such h_0, h_1 can always be found. Let x be a root of I in $\mathbb{F}_{q^{kn}}$, so that $\mathbb{F}_{q^{kn}} = \mathbb{F}_{q^k}(x)$. Observe that by the choice of field-defining polynomial we have $x^q = h_0(x)/h_1(x)$. The factor base is:

$$\mathcal{F} = \{f \in \mathbb{F}_{q^k}[X] \,|\, \deg(f) \leq 1\} \cup \{h_1(x)\}.$$

Let $g \in \mathbb{F}_{q^{kn}}^{\times}$, let $h \in \langle g \rangle$, let $h = g^t$ with the integer t to be computed and let $N = q^{kn} - 1$ be the order of the multiplicative group of $\mathbb{F}_{q^{kn}}$. Thanks to a small adaptation of the argument given by Diem [12], which is an adaptation of that given by Enge and Gaudry [15], one does not need to compute the logarithms of the factor base elements, as we now sketch. Let $F = |\mathcal{F}|$ and let the elements of $\mathcal{F}$ be $f_1, \ldots, f_F$. One constructs a matrix $R = (r_{i,j}) \in (\mathbb{Z}/N\mathbb{Z})^{(F+1)\times F}$ and column vectors $\alpha, \beta \in (\mathbb{Z}/N\mathbb{Z})^{F+1}$ as follows. For each i with $1 \leq i \leq F+1$ choose $\alpha_i, \beta_i \in \mathbb{Z}/N\mathbb{Z}$ uniformly and independently at random and apply the to-be-explained randomised descent algorithm to $g^{\alpha_i} h^{\beta_i}$ to express this as

$$g^{\alpha_i} h^{\beta_i} \equiv \prod_{j=1}^{F} f_j^{r_{i,j}} \pmod{I}. \tag{2}$$

One then computes a lower row echelon form R' of R by using invertible row transformations and applies these row transformations to α and β, resulting in vectors α' and β' respectively. Since the first row of R' vanishes, we have $g^{\alpha'_1} h^{\beta'_1} = 1$ and hence $\alpha'_1 + t\beta'_1 \equiv 0 \pmod{N}$. If $\gcd(\beta'_1, N) = 1$ then one can invert β'_1 to compute t. One can prove that β'_1 is uniformly distributed in $\mathbb{Z}/N\mathbb{Z}$ (cf. [23, Lemma 2.1]) and so the algorithm succeeds with (the very high) probability $\phi(N)/N$; if it does not then one simply repeats the algorithm until it is successful.

We therefore need only describe the descent procedure for carrying out (2), which need only be applied $F+1$ times, i.e., a polynomial number of times. This depends on the recursive application of degree two elimination, as we now explain.

The descent

First note that any element in $\mathbb{F}_{q^{kn}}^{\times}$ can be lifted to an irreducible element of degree 2^e in $\mathbb{F}_{q^k}[X]$, provided that $2^e > 4n$, thanks to a Dirichlet-type theorem due to Wan [51, Theorem 5.1], so one applies this to each featured $g^{\alpha_i} h^{\beta_i}$ before descending to the factor base. Second, we claim the following.

Proposition 1. *Let $Q \in \mathbb{F}_{q^k}[X]$ be an irreducible polynomial of degree $2d \geq 2$. Then Q can be expressed mod I as a product of at most $q+2$ irreducible polynomials of degree dividing d, in time $\mathrm{poly}(q, d)$.*

To see that this implies a quasi-polynomial time algorithm, observe that if Q is irreducible of degree 2^e, then one application of Proposition 1 leads to at most $q+2$ irreducibles of degree dividing 2^{e-1}. Applying it to each of these leads to at most $(q+2)^2$ irreducibles of degree dividing 2^{e-2}. Recursively lowering the degrees in this way eventually leads to at most $(q+2)^e$ degree one polynomials and takes time at most $(q+2)^e \mathrm{poly}(q) = (q+2)^{\log_2 n} \mathrm{poly}(q)$ to compute, as d is at most $2^{e-1} \approx n \approx q$. The running time for the algorithm is therefore $q^{\log_2 n + O(k)}$, which is quasi-polynomial in q^{kn} as claimed.

We now show that in order to prove Proposition 1 it is sufficient for there to be an efficient elimination method for irreducible degree two polynomials in $\mathbb{F}_{q^{kd}}[X]$, expressing each as a product of at most $q+2$ linear polynomials mod I. Let Q be as in Proposition 1. Observe that over the degree d extension $\mathbb{F}_{q^{kd}}$ the polynomial Q factors into d irreducible quadratics $Q_1 \cdot \cdots \cdot Q_d$. Applying the hypothesised degree two elimination method to any one of these quadratics — say Q_1 — expresses it as a product of at most $q+2$ linear polynomials over $\mathbb{F}_{q^{kd}}$ mod I. Then applying the norm map with respect to the extension $\mathbb{F}_{q^{kd}}/\mathbb{F}_{q^k}$ to both sides of the expression maps Q_1 back to Q and the linear polynomials to powers of irreducible polynomials of degree dividing d (the degree depending on the base field of the respective constant terms), which thus eliminates Q as per the proposition.

We now describe such a degree two elimination method which first featured in [18]. Let $Q_1 \in \mathbb{F}_{q^{kd}}[X]$ be an irreducible quadratic to be eliminated mod I. For $a, b, c \in \mathbb{F}_{q^{kd}}$ consider the following equivalence mod I:

$$\begin{aligned} X^{q+1} + aX^q + bX + c \\ \equiv \frac{1}{h_1(X)} (&Xh_0(X) + ah_0(X) \\ &+ bXh_1(X) + ch_1(X)). \end{aligned} \tag{3}$$

Denote the left-hand side and the numerator of the right-hand side of (3) by $L(X)$ and $R(X)$, respectively. The condition $Q_1 \mid R(X)$ can be expressed as

$$b = u_0 a + v_0, \quad c = u_1 a + v_1, \tag{4}$$

for some $u_i, v_i \in \mathbb{F}_{q^{kd}}$ (at least in general; some degenerate cases are easily obviated, see [23, Section 3.1]). Note that since $d_h \leq 2$, the cofactor of Q_1 in $R(X)$ has degree at most one and so no smoothness heuristics are required as long as $L(X)$ splits completely over $\mathbb{F}_{q^{kd}}$. Crucially, although the degree of $L(X)$ is $q+1$, such polynomials split completely over $\mathbb{F}_{q^{kd}}$ with probability $\approx 1/q^3$, which is exponentially larger than the $1/(q+1)!$ one expects for uniformly random polynomials of this degree. Indeed, for $kd \geq 3$ if $ab \neq c$ and $b \neq a^q$, $L(X)$ may be transformed (up to a scalar) into

$$\begin{aligned} &F_B(\overline{X}) = \overline{X}^{q+1} + B\overline{X} + B, \\ &\text{with } B = \frac{(b - a^q)^{q+1}}{(c - ab)^q}, \end{aligned} \tag{5}$$

via $X = \frac{c-ab}{b-a^q}\overline{X} - a$. $L(X)$ splits whenever F_B splits and the transformation from $\overline{X}$ to X is valid. The following theorem is due to Bluher [6].

Theorem 3. *The number of elements $B \in \mathbb{F}_{q^{kd}}^{\times}$ such that the polynomial $F_B(\overline{X}) \in \mathbb{F}_{q^{kd}}[\overline{X}]$ splits completely over $\mathbb{F}_{q^{kd}}$ equals*

$$\begin{aligned} &\frac{q^{kd-1} - 1}{q^2 - 1} \quad \textit{if } kd \textit{ is odd}, \\ &\frac{q^{kd-1} - q}{q^2 - 1} \quad \textit{if } kd \textit{ is even}. \end{aligned}$$

In both cases the number of such B is $\approx q^{kd-3}$. Let $\mathcal{B}$ be the set of all $B \in \mathbb{F}_{q^{kd}}^{\times}$ such that F_B splits completely over $\mathbb{F}_{q^{kd}}$. Since for any $B \in \mathcal{B}$ one can freely choose a and any $b \neq a^q$, while the expression for B in (5) determines c uniquely, there are $\approx q^{3kd-3}$ such $L(X)$ which split completely over $\mathbb{F}_{q^{kd}}$, which explains the $1/q^3$ splitting probability. One way to intuit why this number is so large is that the subset of polynomials of the form $L(X)$ which split completely over $\mathbb{F}_{q^{kd}}$ may be seen to arise from taking the homogeneous eval-

uation of Möbius transformations of X in $X^q - X = \prod_{\alpha \in \mathbb{F}_q}(X-\alpha)$. In particular, for $a',b',c',d' \in \mathbb{F}_{q^{kd}}$ with $a'd'-b'c' \neq 0$ one has

$$\begin{aligned}(c'X+d')^{q+1}&\left(\left(\frac{a'X+b'}{c'X+d'}\right)^q - \frac{a'X+b'}{c'X+d'}\right)\\ &= (a'X+b')^q(c'X+d')\\ &\quad -(a'X+b')(c'X+d')^q \qquad (6)\\ &= (c'X+d')\\ &\quad \times \prod_{\alpha\in\mathbb{F}_q}(a'X+b'-\alpha(c'X+d')), \qquad (7)\end{aligned}$$

where (6) is of the same form as $L(X)$ (up to a scalar) and (7) is a product of linear polynomials. Indeed, this is precisely how Joux approached obtaining such $L(X)$ [26, Section 4.2]. Joux also showed that the number of such polynomials is

$$\begin{aligned}&|\operatorname{PGL}_2(\mathbb{F}_{q^{kd}})| / |\operatorname{PGL}_2(\mathbb{F}_q)|\\ &\quad = (q^{3kd}-q^{kd})/(q^3-q) \approx q^{3kd-3},\end{aligned}$$

broadly matching the number arising from the approach already described.

The following theorem due to Helleseth and Kholosha [24, Theroem 5] (generalised to arbitrary characteristic) characterises the set $\mathcal{B}$.

Theorem 4.

$$\mathcal{B} = \left\{ \frac{(z^{q^2}-z)^{q+1}}{(z^q-z)^{q^2+1}} \,\middle|\, z \in \mathbb{F}_{q^{kd}} \backslash \mathbb{F}_{q^2} \right\}.$$

Combining Theorem 4 with the expression for B in (5) and the expressions for b and c in (4), to eliminate Q_1 one needs to find an $(A,Z) \in \mathbb{F}_{q^{kd}} \times (\mathbb{F}_{q^{kd}} \backslash \mathbb{F}_{q^2})$ satisfying

$C/\mathbb{F}_{q^{kd}}$:

$$\begin{aligned}&(Z^{q^2}-Z)^{q+1}(-u_1A^2+(v_1-u_0)A+v_0)^q\\ &-(Z^q-Z)^{q^2+1}(-A^q+u_0+Au_1)^{q+1} = 0.\end{aligned}$$

That there are sufficiently many points on C was proven in [23] by analysing the action of $\operatorname{PGL}_2(\mathbb{F}_q)$ on Z in order to prove that there is an absolutely irreducible factor of C, and then applying the Weil bound. One also needs to consider so-called *descent traps*, which are elements that divide $h_1(X)X^{q^{kd+1}} - h_0(X)$ for $d \geq 0$ which can not be eliminated in the above manner and so must be avoided during the descent. Computing points on C is efficient since one can take any $Z \in \mathbb{F}_{q^{kd}} \backslash \mathbb{F}_{q^2}$ which gives a polynomial in A, whose $\mathbb{F}_{q^{kd}}$ roots can be computed by taking the greatest common denominator with $A^{q^{kd}} - A$, for instance, which takes time polynomial in $\log q^{kn}$. The above algorithm and considerations lead to the following [23, Theorem 1.2].

Theorem 5. *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 $h_1X^q - h_0$, the DLP in $\mathbb{F}_{q^{kn}} \cong \mathbb{F}_{q^k}[X]/(I)$ can be solved in expected time*

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

Thanks to Kummer theory, such h_1, h_0 are known to exist when $n = q-1$, which gives the following easy corollary when $m = ik(p^i-1)$ [23,Theorem 1.1].

Theorem 6. *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*

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

One may also replace the prime p in Theorem 6 by a (fixed) prime power p^r by setting $k = 18r$. Proving the existence of h_0, h_1 for general extension degrees as per Theorem 5 seems to be a hard problem, even though in practice it is very easy to find such polynomials and heuristically is almost certain.

Computational records and impact

Since early 2013 several computational records have been set using the techniques from [18, 19, 21, 23, 26, 30], which have dwarfed previous records, demonstrating categorically their superiority. Moreover, such large scale computations also help to inform one of potential pitfalls (cf. the traps noted in [21, Remark 1]), and can also lead to theoretical insights that give rise to novel or improved algorithms.

In practice, since the running time is dominated by the descent one first computes the logarithms of the factor base elements (cf. [18, Section 3] and [26, Section 4.2]), so that only one descent is needed. A descent usually consists of: several classical elimination steps; the GKZ elimination for irreducibles of small even degree (for which $kd \geq 4$ suffices in practice) and Joux's elimination method for irreducibles of small odd degree [26]; and finally degree two elimination, either from GGMZ or [26]. The crossover points between these techniques should be determined using a dynamic programming bottom-up approach [21].

Table 1 contains a selection of discrete logarithm computations in finite fields. All details may be found in [10]. At the time

Bitlength	Charact.	Kummer	Who and when	Complexity
127	2	no	Coppersmith, 1984	$L(1/3, [1.526, 1.587])$
401	2	no	Gordon and McCurley, 1992	$L(1/3, [1.526, 1.587])$
521	2	no	Joux and Lercier, 2001	$L(1/3, 1.526)$
607	2	no	Thomé, 2002	$L(1/3, [1.526, 1.587])$
613	2	no	Joux and Lercier, 2005	$L(1/3, 1.526)$
556	medium	yes	Joux and Lercier, 2006	$L(1/3, 1.442)$
676	3	no	Hayashi et al., 2010	$L(1/3, 1.442)$
923	3	no	Hayashi et al., 2012	$L(1/3, 1.442)$
1175	medium	yes	Joux, 24 December 2012	$L(1/3, 1.260)$
1425	medium	yes	Joux, 6 January 2013	$L(1/3, 1.260)$
1778	2	yes	Joux, 11 February 2013	$L(1/4 + o(1))$
1971	2	yes	GGMZ, 19 February 2013	$L(1/3, 0.763)$
4080	2	yes	Joux, 22 March 2013	$L(1/4 + o(1))$
6120	2	yes	GGMZ, 11 April 2013	$L(1/4)$
6168	2	yes	Joux, 21 May 2013	$L(1/4 + o(1))$
1303	3	no	AMOR, 27 January 2014	$L(1/4 + o(1))$
4404	2	no	GKZ, 30 January 2014	$L(1/4 + o(1))$
9234	2	yes	GKZ, 31 January 2014	$L(1/4 + o(1))$
3796	3	no	Joux and Pierrot, 15 September 2014	$L([o(1), 1/4 + o(1)])$
1279	2	no	Kleinjung, 17 October 2014	$L([o(1), 1/4 + o(1)])$
4841	3	no	Adj et al., 18 July 2016	$L([o(1), 1/4 + o(1)])$

Table 1 A selection of discrete logarithm computations in finite fields.

of writing the largest example DLP to have been solved was in the field of 2^{9234} elements, which took ≈ 45 core years of computation. The impact on cryptography can be seen from the solution of DLPs in the fields of bitlength 4404 and 4841, which both arise from what were designed to be industry-standard 128-bit secure supersingular curves. These took ≈ 5 and ≈ 200 core years, respectively. Note that these fields can not be represented by a Kummer extension; fields which admit such a representation make the computation much easier due to the presence of factor base automorphisms and other descent advantages, making them ideal for setting records. Since parameters would have to increase significantly to counter the quasi-polynomial time algorithms, thus making the cryptosystems inefficient, and since further algorithmic developments in this area should be expected, small characteristic supersingular curves (or those with low embedding degree) should be considered completely insecure for pairing-based cryptography.

In summary, we see that long-studied so-called hard problems can suddenly become easy with the right ideas, while basing cryptography on unproven computational assumptions is inherently risky in general. Whether the prime field DLP or the integer factorisation problem will remain hard remains to be seen: the current record bitlength for both of these problems is 768, which took ≈ 5300 [32] and ≈ 1700 [31] core years, respectively. However, the ideas behind the breakthroughs do not seem to be extendable to these scenarios since there is no analogue of the tremendously useful polynomial $X^q - X$ around which one can build such an algorithm.

Acknowledgements
The author would like to thank Arjen Lenstra for his useful comments.

References

1 L. M. Adleman, A subexponential algorithm for the discrete logarithm problem with applications to cryptography, *20th Annual Symposium on Foundations of Computer Science,* IEEE, 1979, pp. 55–60.

2 L. M. Adleman, The function field sieve, *Algorithmic Number Theory*, Springer, 1994, pp. 108–121.

3 L. M. Adleman, M.-D.A. Huang, Function field sieve method for discrete logarithms over finite fields, *Inform. and Comput.* 151(1) (1999), 5–16.

4 R. Barbulescu, P. Gaudry, A. Joux and E. Thomé, A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic, *Advances in Cryptology–EUROCRYPT 2014*, LNCS 8441, Springer, 2014, pp. 1–16.

5 I. F. Blake, G. Seroussi, N. P. Smart, *Advances in Elliptic Curve Cryptography*, Cambridge University Press, 2005.

6 A. W. Bluher, On $x^{q+1} + ax + b$, *Finite Fields Appl.* 10(3) (2004), 285–305.

7 B. den Boer, Diffie–Hellman is as strong as discrete log for certain primes, *Proceedings on Advances in Cryptology–CRYPTO '88*, Springer, 1990, pp. 530–539.

8 D. Boneh, M. Franklin, Identity-based encryption from the Weil pairing, *Advances in Cryptology–CRYPTO 2001*, LNCS 2139, Springer, 2001, pp. 213–229.

9 E. R. Canfield, P. Erdős and C. Pomerance, On a problem of Oppenheim concerning 'factorisatio numerorum', *J. Number Theory*, 17(1) (1983), 1–28.

10 Computations of discrete logarithms sorted by date, https://members.loria.fr/LGremy/dldb/index.html.

11 D. Coppersmith, Fast evaluation of logarithms in fields of characteristic two, *IEEE Trans. Inform. Theory* 30(4) (1984), 587–594.

12 C. Diem, On the discrete logarithm problem in elliptic curves, *Compositio Math.* 147 (2011), 75–104.

13 W. Diffie and M. E. Hellman, New directions in cryptography, *IEEE Trans. Inform. Theory*, 22(6) (1976), 644–654.

14 T. ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithms, *Advances in Cryptology–CRYPTO '84*, LNCS 196, Springer, 1985, pp. 10–18.

15 A. Enge and P. Gaudry, A general framework for subexponential discrete logarithm algorithms, *Acta Arithmetica* 102 (2002), 83–103.

16 S. D. Galbraith, K. Harrison and D. Soldera, Implementing the Tate Pairing, *Proceedings of the 5th International Symposium on Algorithmic Number Theory, ANTS-V*, Springer, 2002, pp. 324–337.

17 C. F. Gauß, *Disquisitiones Arithmeticae*, Leipzig, 1801. Translated by A. A. Clarke. Yale University Press, 1965.

18 F. Göloğlu, R. Granger, G. McGuire and J. Zumbrägel, On the function field sieve and the impact of higher splitting probabilities: application to discrete logarithms in $\mathbb{F}_{2^{1971}}$ and $\mathbb{F}_{2^{3164}}$, *Advances in Cryptology–CRYPTO 2013*, LNCS 8043, Springer, 2013, pp. 109–128.

19 F. Göloğlu, R. Granger, G. McGuire and J. Zumbrägel, Solving a 6120-bit DLP on a desktop computer, *Selected Areas in Cryptography–SAC 2013*, LNCS 8282, Springer, 2014, pp. 136–152.

20 D. M. Gordon, Discrete logarithms in $GF(p)$ using the number field sieve, *SIAM J. Discrete Math.* 6(1) (1993), 124–138.

21 R. Granger, T. Kleinjung and J. Zumbrägel, Breaking '128-bit secure' supersingular binary curves, *Advances in Cryptology–CRYPTO 2014*, LNCS 8617, Springer, 2014, pp. 126–145.

22 R. Granger, T. Kleinjung and J. Zumbrägel, On the powers of 2, *IACR Cryptology ePrint Archive* (2014), eprint.iacr.org/2014/300.

23 R. Granger, T. Kleinjung and J. Zumbrägel, On the discrete logarithm problem in finite fields of fixed characteristic, to appear in *Transactions of the AMS*.

24 T. Helleseth and A. Kholosha, $x^{2^l+1} + x + a$ and related affine polynomials over $GF(2^k)$' *Cryptogr. Commun.* 2(1) (2010), 85–109.

25 A. Joux, A one round protocol for tripartite Diffie–Hellman, *Algorithmic Number Theory*, Springer, 2000, pp. 385–393.

26 A. Joux, A new index calculus algorithm with complexity $L(1/4 + o(1))$ in small characteristic, *Selected Areas in Cryptography–SAC 2013*, LNCS 8282, Springer, 2014, pp. 355–379.

27 A. Joux and R. Lercier, The function field sieve is quite special, *Algorithmic Number Theory*, Springer, 2002, pp. 431–445.

28 A. Joux and R. Lercier, The function field sieve in the medium prime case, *Advances in Cryptology–EUROCRYPT 2006*, LNCS 4117, Springer, 2006, pp. 254–270.

29 A. Joux, R. Lercier, N. Smart and F. Vercauteren, The number field sieve in the medium prime case, *Advances in Cryptology–CRYPTO 2006,* Springer, 2006, pp.326–344.

30 A. Joux and C. Pierrot, Improving the polynomial time precomputation of Frobenius representation discrete logarithm algorithms, *Advances in Cryptology–ASIACRYPT 2014*, LNCS 8873, Springer, 2014, pp. 378–397.

31 T. Kleinjung, K. Aoki, J. Franke, A. K. Lenstra, E. Thomé, J. W. Bos, P. Gaudry, A. Kruppa, P. L. Montgomery, D. A. Osvik, H. J. J. te Riele, A. Timofeev and P. Zimmermann, Factorization of a 768-Bit RSA Modulus, *Advances in Cryptology–CRYPTO 2010*, LNCS 6223, Springer, 2010, pp. 333–350.

32 T. Kleinjung, C. Diem, A. K. Lenstra, C. Priplata and C. Stahlke, Computation of a 768-bit Prime Field Discrete Logarithm, *Advances in Cryptology–EUROCRYPT 2017*, LNCS 10210, Springer, 2017, pp. 333–350.

33 N. Koblitz, Elliptic Curve Cryptosystems, *Mathematics of Computation* 48 (1987), 203–209.

34 M. Kraitchik, *Théorie des nombres*, Vol. 1, Gauthiers-Villars, 1922.

35 M. Kraitchik, *Recherches sur la théorie des nombres*, Gauthiers-Villars, 1924.

36 C. Lanczos, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, *J. Research Nat. Bur. Standards* 45 (1950), 255–282.

37 A.K. Lenstra and H.W. Lenstra, Jr (eds.), *The Number Field Sieve*, Springer, 1993.

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

39 U.M. Maurer, Towards the equivalence of breaking the Diffie–Hellman protocol and computing discrete logarithms, *Advances in Cryptology–CRYPTO '94*, LNCS 839, Springer, 1994, pp. 271–281.

40 U.M. Maurer and S. Wolf, Diffie–Hellman Oracles, *Advances in Cryptology–CRYPTO '96*, LNCS 1109, Springer, 1996, pp. 268–282.

41 A. Menezes, S. Vanstone and T. Okamoto, Reducing elliptic curve logarithms to logarithms in a finite field, *Proceedings of the Twenty-third Annual ACM Symposium on Theory of Computing (STOC '91)*, ACM, 1991, pp. 80–89.

42 V. Miller, Uses of Elliptic Curves in Cryptography, *Advances in Cryptology–CRYPTO 1985*, LNCS 218, Springer, 1985, 417–426.

43 V.I. Nechaev, On the complexity of a deterministic algorithm for a discrete logarithm, *Mat. Zametki* 55(2) (1994), 91–101, 189.

44 A.M. Odlyzko, Discrete logarithms in finite fields and their cryptographic significance, *Advances in Cryptology–CRYPTO '84*, LNCS 209, Springer, 1985, pp. 224–314.

45 S.C. Pohlig and M.E. Hellman, An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance (Corresp.), *IEEE Trans. Inform. Theory* 24(1) (1978), 106–110.

46 J.M. Pollard, Monte Carlo methods for index computation (mod p), *Math. Comp.* 32(143) (1978), 918–924.

47 R. Sakai, K. Ohgishi and M. Kasahara, Cryptosystems based on pairing, *Symposium on Cryptography and Information Security*, Okinawa, Japan, 2000, pp. 26–28.

48 P.W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM J. Computing* 26(5) (1997), 1484–1509.

49 V. Shoup, Lower bounds for discrete logarithms and related problems, *Advances in Cryptology–EUROCRYPT '97*, LNCS 1223, Springer, 1997, pp. 256–266.

50 N.P. Smart, *Cryptography Made Simple*, Springer, 2016.

51 D. Wan, Generators and irreducible polynomials over finite fields, *Math. Comp.* 66(219) (1997), 1195–1212.

52 D.H. Wiedemann, Solving sparse linear equations over finite fields, *IEEE Trans. Inform. Theory* 32 (1986), 54–62.