# Elliptic Curves and their use in Cryptography

Victor S. Miller
Center for Communications Research
Thanet Road
Princeton, NJ 08540, USA

21 March 1997

**Abstract**

The security of many cryptographic protocols depends on the difficulty of solving the so-called "discrete logarithm" problem, in the multiplicative group of a finite field. Although, in the general case, there are no polynomial time algorithms for this problem, constant improvements are being made – with the result that the use of these protocols require much larger key sizes, for a given level of security, than may be convenient.

An abstraction of these protocols shows that they have analogues in any group. The challenge presents itself: find some other groups for which there are no good attacks on the discrete logarithm, and for which the group operations are sufficiently economical. In 1985, the author suggested that the groups arising from a particular mathematical object known as an "elliptic curve" might fill the bill. In this paper I review the general cryptographic protocols which are involved, briefly describe elliptic curves and review the possible attacks against such cryptosystems. Finally, I mention other applications that elliptic curves have had upon the analysis of other cryptosystems not involving them.

## 1 Introduction

The art of Cryptography (*secret writing*) is to design a method or system which allows two parties to communicate secret messages to one another without anyone else being able to understand them. Any designer of a cryptographic system must be well-versed in the art of cryptanalysis – breaking such systems without having any of the special information known to both parties.

Before the 1970s all such systems relied on both parties possessing a shared secret – *the private key* – which was not known to anyone else. The logistics of arranging this, known as *key distribution*, are formidable in practice. In 1976 this changed significantly: Whitfield Diffie and Martin Hellman [6] showed the problem of key distribution could be made insignificant, as long as the two parties could rely on each other's identity (this is a completely different problem, which we won't discuss). A few years later, Rivest, Shamir and Adleman [28]

devised another method for using cryptography with a publicly available key (again relying on the identity of the other party). Both of these methods made fundamental use of the arithmetic in some algebraic object. Although these algebraic objects – the integers modulo a prime number, or the integers modulo a composite number – were fairly basic (to number theorists), their use still presented great challenges to any would be adversaries.

For the Diffie-Hellman protocol an adversary needed to solve the "discrete logarithm" problem, and for the RSA protocol he needed to factor large numbers. Since the time that these two papers appeared there have been significant improvements in solving both of these problems [14]. Nevertheless, solving neither problem is easy. In this paper, I discuss another, more complicated, algebraic object, known as an "Elliptic Curve". Elliptic Curves have been objects of intense study by pure mathematicians for well over 100 years, and have many deep and interesting properties. They have had an impact on many important problems in mathematics, the most spectacular, being on the solution of the centuries old "Fermat's Last Theorem" [38].

In this paper, I'll discuss how they may be used, very naturally, in the art of cryptography, and appear to be immune from the attacks that have been mounted against other cryptosystems. Many other protocols have been developed whose security depends on the intractibility of the original discrete logarithm [31, 7] problem. However, these protocols work in almost exactly the same way when used with any group – in particular the group given by an elliptic curve.

In addition, elliptic curves have recently been used in the cryptanalysis of other cryptographic systems which are not based on the use of elliptic curves.

## 2 Prehistory

In 1976 Diffie and Hellman [6] proposed an ingenious solution to the problem of two parties communicating over a public channel agreeing on a common value known only to the two of them.

The security of this protocol depended on the conjectural difficulty of a particular problem (now known as the "Diffie-Hellman Problem"). They pointed out that this problem was easily solved by, in turn, solving the so-called "discrete logarithm" problem (see below). They conjectured that this was the best way.

It is well known that the set of residues of integers modulo $p$ (a prime), admits a "primitive root" $g$: That is for every other non-zero residue, $x$, modulo $p$, there is an integer $a$ such that

$$x = g^a \bmod p.$$

We then write $a = \mathrm{ind}_g x$

If **Alice** (A) and **Bob** (B) wish to agree on a common secret value, Alice generates an integer $a$ between 0 and $p-1$ at random. Similarly, Bob generates an integer $b$. Alice sends the value $g^a \bmod p$ to Bob, and Bob sends $g^b \bmod p$ to Alice. Both may easily calculate $g^{ab} \bmod p$: Alice calculates $(g^b)^a \bmod p$ (since

she know $a$ and and has received $g^b \bmod p$); and Bob does the same with $a$ and $b$ interchanged. An eavesdropper (**Charlie** – (C)), only sees $g^a \bmod p$ and $g^b \bmod p$, and must try to reconstruct $g^{ab} \bmod p$. Charlie's task is precisely the "Diffie-Hellman" problem.

One way of solving this problem is to solve the "discrete logarithm" problem:

Given $g^a \bmod p$, find $a$.

In their paper, and a subsequent paper by Pohlig and Hellman [26], they outlined a method of finding discrete logarithms which took roughly $\sqrt{q}$ operations, where $q$ was the largest prime dividing $p-1$ [1]. This gave an important criterion that needed to be satisfied by any prime $p$ used in this protocol:

The largest prime divisor $q$ of $p-1$ must be "large".

Pohlig and Hellman conjectured that their algorithm was the best possible. For the original problem they were proved wrong. However, in a sense (to be explained), they were correct (see section 3). John Pollard [27] gave another algorithm with approximately the same running time, but with a much smaller storage requirement.

Unknown to them, a description of a much better method for calculating discrete logarithms had been given by the number theorists Western and Miller in 1968 [37] This method was rediscovered, and precisely analyzed by Adleman in 1978. It was dubbed "the index calculus" by Andrew Odlyzko [25]. Odlyzko's paper is an excellent discussion of the best algorithms known for discrete logarithms through 1985. For another point of view and an updated version see McCurley's survey [18].

The fact that there was a much faster method to solve the discrete logarithm problem made the use of the Diffie-Hellman protocol more expensive (since the modulus p needed to be chosen much larger), and so, a bit less attractive.

It was realized by many researchers that the "Diffie-Hellmann" protocol would work with any finite field, not just the integers modulo $p$. A natural candidate to use was a finite field of characteristic 2. First, these fields are well-known to workers in the field of error-correcting codes; and, second, the implementation of the field multiplication operations in hardware can be quite inexpensive when compared with those modulo a large prime. There was also the slight bonus that it was possible (when $2^n - 1$ was a Mersenne prime), to have the order of the multiplicative group of the field to be a prime. This was the strongest possible case with respect to the Pohlig and Hellmann attack. It was soon realized, however, that a method very similar to the method of Western, Miller, and Adleman (the "index calculus") would work for such fields [9]. Nevertheless, the speed and economy of field operations in such fields led a few companies (such as Hewlett-Packard) to manufacture chips to implement "field arithmetic" in the finite field of $2^{127}$ elements, for use with such a protocol.

[1] According to Nechaev [24], the "Baby-Step Giant-Step" method was known to A. O. Gelfond in 1962, and the Pohlig-Hellman method was known to Nechaev in 1965.

In 1983 Blake, Fuji-hara, Mullin and Vanstone (BFMV) [2], presented a way of making the index calculus in these fields much more efficient. Later that year, Don Coppersmith [4] built on these ideas to come up with a new immensely faster algorithm for the discrete logarithm in characteristic 2. As an example of its capabilities he solved a discrete logarithm problem over the field of $2^{127}$ elements in less than a minute, as compared with the BFMV estimate of a few months.

A number of researchers realized that the only thing necessary to make the Diffie-Hellmann protocol work, was a way of multiplying two elements together. Thus this protocol would work in any finite group. In 1984 I considered what would happen if an "elliptic curve" (something which was one of the main topics of my mathematical research) were used. I came to the conclusion that any analogies of the index calculus that I could think of could be shown to doomed to failure. Simultaneously, Neal Koblitz [12], of the University of Washington, also was thinking of using Elliptic curves for this purpose. I presented my results [23] at the Crypto 85 conference, and argued that although, it was more complex to do arithmetic operations on elliptic curves, that their apparent immunity from index calculus attacks made them quite attractive. An additional advantage of elliptic curves, is that there are many different curves of about the same size. The Pohlig-Hellman paper above gave a criterion which must be satisfied for good security. Using elliptic curves gives many more chances to satisfy this.

In an unpublished manuscript in 1986 [22], I gave an efficient method for calculating the so-called "Weil Pairing". This gave an explicit connection between the arithmetic in an elliptic curve and multiplication in a finite field. In particular, this reduced the problem of calculating a discrete logarithm on an elliptic curve to calculating a discrete logarithm in the multiplicative group of a finite field. However, all was not lost, as the finite field involved was usually stupendously large – and so effectively unapproachable. Menezes, Okamoto and Vanstone in 1991 [20] used these results about the efficient calculation of the Weil-pairing to show that using a member of a certain small class (the "supersingular curves") of elliptic curves, gave not much more cryptographic strength than using the original Diffie-Hellman construction. Also see the thesis of Kaliski [11].

# 3 Generic Methods

A number of the algorithms used for calculating "discrete logarithms" don't use any special properties of the elements of the group. These algorithms are the "Baby-Step Giant-Step" method of Shanks [33, page 419], the "lambda" and "catching kangaroos" method of Pollard [27], and the Pohlig-Hellman algorithm [26].

Shoup [34] calls a method of finding discrete logarithms *generic* if the only properties of the underlying group that it uses, are the fact that elements may be multiplied, inverted, and that each element has a unique encoding as a bit string. This specifically excludes methods such as the *index calculus* which make

essential use of a particular encoding of elements. All of the methods mentioned above fall into this category.

Shoup shows (generalizing a result of Nechaev [24]) that any generic probabilistic algorithm for solving the discrete logarithm problem must take expected time at least $\sqrt{q}$ where $q$ is the largest prime divisor of the order of the group. If we can arrange that $q$ is nearly the size of the group, then we've made the adversary's (Charlie) task exponentially more difficult than the work for legitimate users (Alice and Bob). This is the ideal state of affairs for the designers and users of a cryptosystem.

# 4 Index Calculus

If we can't use any particular property of the encoding of groups elements, then Nechaev and Shoup's results, essentially show that the best we can do in solving the discrete logarithm problem is to use a "square root" algorithm like Baby-Step Giant Step, or one of Pollard's algorithms. However, as I previously stated, a much better algorithm exists for the case that the group is the integers modulo a prime $p$. Here is an outline of how it works:

There are two phases:

1. The equation gathering phase.
2. Solving for the individual logarithm.

We first choose a size $N$ for the "factor base" (to be explained shortly). Our "factor base" consists of the first $N$ prime number $l_1, \ldots, l_N$. We then repeat the following step until we have $N$ equations:

> Pick $a$ at random between 0 and $p-1$. Calculate $g^a \bmod p$, and let $m$ be an integer between 1 and $p-1$ such that $m \equiv g^a \bmod p$. If $m$ factors completely into the primes in the factor base, we have
>
> $$g^a \equiv l_1^{e_1} \ldots l_N^{e_N} \bmod p$$
>
> for some integer $e_1, \ldots, e_N$. In that case we have an equation
>
> $$a = e_1 \mathrm{ind}_g l_1 + \cdots + e_N \mathrm{ind}_g l_N.$$

We treat $\mathrm{ind}_g l_i$ as the unknowns, and solve for them once we have $N$ equations.

In the second phase, we wish to find $\mathrm{ind}_g x$. As above, we pick $a$ at random, and calculate $g^a x \bmod p$. We treat this residue as an integer, and see if it factor completely into the prime $l_i$. If it does, we have our answer:

$$a + \mathrm{ind}_g x = f_1 \mathrm{ind}_g l_1 + \cdots + f_N \mathrm{ind}_g l_N.$$

To choose the best value of $N$, we have a trade off: if we make $N$ bigger, the probabilty that a random residue will completely factor increases, but the

size of the linear system that we must solve also gets bigger. To choose the best value involves some delicate number theory: see [18].

There are many variations and refinements of the above theme [1, 30, 36], some of which have made spectacular improvements on the basic algorithm. However, all of them are based on the idea of finding some means of being able to factor, or decompose, elements in some object (like the integers) that can be associated to our original problem.

# 5 Elliptic Curves

## 5.1 General Facts

Elliptic curves are an example of an *algebraic group*. That is, elements of the group are given by a set of $n$-tuples of elements of some field, satisfying a system of polynomial equations, and the group laws (composition and inversion) are given by rational functions of the input $n$-tuples (ratios of two polynomials). This means that the group laws can be calculated fairly economically. The study of elliptic curves is a very large subject, which can only be touched on here. A standard reference is Silverman [35]. The book of Menezes [19] is a good elementary introduction, as well as an excellent reference for the impact of elliptic curves on modern cryptosystems.

Recall that a *field* is an algebraic object with addition, multiplication, a 0 element, a unit (denoted by 1), and inversion of non-zero elements. There are many different kinds of fields, but, in this paper, we will mostly work with finite fields (also called *Galois Fields*), GF($q$), where $q$ is a power of some prime $p$.

Algebraic groups fall into two classes (roughly): *affine groups*, and *complete groups*. The affine groups are those that have natural definitions in terms of matrix multiplication. The additive group and the multiplicative group are examples of these. For example, if $F$ is field, define the additive group of $F$:

$$A(F) = \left\{ \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix} : a \in F \right\},$$

and the multiplicative group of $F$:

$$M(F) = \left\{ \begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix} : a, b \in F, ab = 1 \right\}.$$

From the point of view of cryptography, the discrete logarithm problem in a matrix group of dimension $n$, is easily reducible to the discrete logarithm problem in the multiplicative group of a field extension $K$ over $F$ of degree $m$, where $m \leq n$. More concretely, if $F$ is a finite field with $q$ elements, then $K$ is a finite field of $q^m$ elements. Thus, using such a group provides about the same amount of security as the original discrete logarithm problem.

Elliptic Curves are an example of the complete algebraic groups. In a down to earth description, an elliptic curve $E$ defined over a field $F$ is the set of

solutions of a cubic equation in $x$ and $y$, $E(x,y)=0$, where the coefficients of $E$ are all in the field $F$.

We need one, simple, fundamental fact:

> If $f(x)=x^3-a_1x^2+a_2x-a_3$, then the sum of the three roots of $f(x)=0$ is $a_1$.

This shows that

**Lemma 1** *If two points $P_1=(x_1,y_1)$ and $P_2=(x_2,y_2)$ are on the curve $E(x,y)=0$, and $x_1,x_2,y_1,y_2\in F$, then the line joining $P_1$ and $P_2$ intersects $E$ in a third point $P_3=(x_3,y_3)$, and $x_3,y_3\in F$. We use the notation $P_3=P_1*P_2$.*

Proof: Let $y=\lambda x+\nu$ be the line joining $P_1$ and $P_2$. Since, $x_1,x_2,y_1,y_2\in F$ we must have $\lambda,\nu\in F$. If we substitute $\lambda x+\nu$ for $y$ in $E(x,y)=0$, we get a cubic equation in $x$ (since $E$ has degree 3), all of whose coefficients are in $F$. Two of the roots of this cubic are $x_1$ and $x_2$. If the third root is $x_3$ then their sum is in $F$. Thus $x_3\in F$. Finally, since $y_3=\lambda x_3+\nu$, we must have $y_3\in F$. $\square$

We remark that we can define $P*P$ by using the tangent line to the curve $E$ at the point $P$.

Suppose that $O$ is a point on $E(x,y)$ with the property that $O*O=O$. Such a point is called a *flex* (the tangent to the curve at that point has contact of order 3), and every cubic curve has precisely 9 of them. We then define

$$P+Q=(P*Q)*O.$$

That this new "+" is commutative is straightforward. To show that it is associative, is not. The flex $O$ is the zero element for this group law, and $-P=P*O$. A convenient way of thinking of this group law is that the 3 points which are the intersection of a line with the curve have sum equal to 0.

It is standard to make a linear change of coordinates so that the cubic curve can be written in the following form:

$$E: y^2+a_1xy+a_3y=x^3+a_2x^2+a_4x+a_6,$$

where $a_1,\ldots,a_6\in F$. In this case we choose the flex $O$ to be the "point at infinity".

When we're working in a field whose characteristic is not 2 (that is $1+1\neq 0$), we can "complete the square", and use the form

$$y^2=x^3+a_2x^2+a_4x+a_6.$$

In this case, the point at infinity corresponds to the collection of all vertical lines. So that the group law consists of finding the third point of interesection of the line, and "flipping" it – changing the $y$-coordinate. When the characteristic is not 3 (i.e. $1+1+1\neq 0$), then we may assume, by changing coordinates, that $a_2=0$.

When the characteristic is 2, and $F$ is a finite field there are two standard forms:

$$y^2+xy=x^3+a_2x^2+a_6,$$

where either $a_2 = 0$, or $a_2$ is a particular fixed non-zero element of $F$ (two possibilities). This curve is called *ordinary*. The other form is

$$y^2 + a_3 y = x^3 + a_4 x + a_6.$$

These curves are known as *supersingular*. There are only a small number of isomorphism classes of these curve. See [19, Chapter 3] for details.

So the elements of our group, denoted by $E(F)$ are the set of solutions to $E(x, y) = 0$, along the with point at infinity. The Hasse-Weil theorem states that

> The size $\#E(F)$ is $q + 1 - t$, where $t \leq 2\sqrt{q}$.

Thus, the size of the group of an elliptic curve defined over $F$ is roughly the same size, $q - 1$ as the size of the multiplicative group of non-zero points of $F$. However, unlike that case, where we have only one group for a given $F$, as we vary the parameters $a_i$ we can get many groups. In fact, essentially every possible value of $t$, subject to the above inequality, can be attained. When the characteristic is 2, however, and we are working with ordinary curves, $t$ must be odd. Even better, every group order $q+1-t$, when $t \leq \sqrt{q}$, is taken on roughly with equal probability. This fact is exploited by Hendrik Lenstra [15] to give a fast algorithm for factoring integers.

So, we know that if $E$ is an elliptic curve, then $E(F)$ is an abelian group. However, not all such groups can occur. In fact

> $E(F)$ is either a cyclic group or the direct sum of two cyclic groups.

More is known. Rück in [29] gives a criterion which tells which groups can (and do) occur. Furthermore, Howe [10] refines this by analyzing the probability of these groups occuring.

## 5.2 Cryptographic Use

In almost any discussion of algorithms for factoring or discrete logarithm computation, we use the useful terminology of Carl Pomerance:

> An integer is *smooth* is it has no large prime divisors.

The size of *large* is, in practice, an adjustable parameter of the problem. When we are dealing with polynomials over a finite field, our measure of "largeness" is the degree of the polynomial.

In the case of the original Diffie-Hellman protocol, if $q$, the number of points in a finite field, had the property that $q - 1$ were smooth, then we were out of luck for using the field $F$. However, when we use elliptic curves, it is very likely (essentially certain), that there are a number of curves $E$ defined over $F$ whose groups have prime order (or twice a prime, if we are working in characteristic 2). In fact one can say that a curve chosen at random has a probability of roughly $1/\log q$ of having such a group order.

This wonderful structure of elliptic curves, would not be of any practical use to us, unless we could actually find the curves whose group orders are divisible by a large prime. In [32], Rene Schoof gives an algorithm that computes $\#E(F)$, in time polynomial in $\log q$, only given the coefficients $a_i$. Schoof's original algorithm was speeded up greatly, using ideas of Atkin, Elkies and the author (in an unpublished manuscript), see [21, 5, 13].

However, the most striking thing about elliptic curves, is that, with an extremely small set of exceptions, there appear to be no non-generic algorithms for solving the discrete logarithm in them. In my paper [23] I analyzed what would happen if one were to look for a factor base in the direct analogy of the situation with $\mathrm{GF}(p)$: "lifting points" to points on a elliptic curve defined over the field of rational numbers. That is, given a point $P \in E(F)$ on an elliptic curve $E$ over a finite field $F$, find an elliptic curve $\tilde{E}$ defined over the rational field $\mathbb{Q}$, and a point $Q \in \tilde{E}(\mathbb{Q})$, such that $Q \equiv P \bmod p$.

I argued that, compared to the case of integers, that points on elliptic curves over the rationals are very sparse: List all the points on the curve in order by their "height" (number of bits in the numerator and denominator of the $x$-coordinate). Then, in order to have a probability of $c > 0$ of lifting a random point $P \in E(F)$ to a point $Q \in \tilde{E}(\mathbb{Q})$, one would need to consider points of height at least $2^{cp}$. This is clearly impossible.

A similar argument holds when the field $F = \mathrm{GF}(2^n)$. In that case, instead of integers, we work with polynomials with coefficients in $\mathrm{GF}(2)$.

## 5.3 The Weil Pairing

If $E$ is an elliptic curve over a field $F$, and $n$ is a positive integer then we define

$$E[n](F) = \{P \in E(F) | nP = 0\}.$$

the points of order $n$ in $E(F)$. The *Weil pairing* is something like an algebraic inner product between elements of the group $E(F)$. More specifically, for every positive integer $m$, we have a map

$$e_m : E[m](F) \times E[m](F) \to F^*.$$

This map satisfies

$$\begin{aligned}
e_m(P,P) &= 1, P \in E[m](F) \\
e_m(P+Q,R) &= e_m(P,R)e_m(Q,R), P,Q,R \in E[m](F) \\
e_m(P,Q) &= e_m(Q,P)^{-1}, P,Q \in E[m](F) \\
e_{mn}(P,Q) &= e_m(P,nQ), P \in E[m](F), Q \in E[mn](F) \\
e_m(P,Q) &\neq 1, \text{ if } P \in E[m](F), P \neq 0, \text{ then there is a } Q \in E[m](K).
\end{aligned}$$

In the above, $K$ is another field containing $F$. In [22] I showed how the Weil pairing may be computed efficiently: $e_m(P,Q)$ may be computed in about $\log m$

elliptic curve operations in $F$. The usefulness of the Weil pairing in understanding the elliptic discrete logarithm problem, is that if there are two *independent* point $P, Q \in E(F)$ of order $m$, and $e_m(P,Q)$ is a primitive $m$-th root of 1, then

$$e_m(aP, Q) = e_m(P, Q)^a.$$

This means that if we know $P$ and $Q$, and we want to find the discrete logarithm of a multiple of $P$ to base $P$, we can use the Weil-pairing to reduce it to an ordinary discrete logarithm problem.

At first, this would seem to say that the elliptic curve discrete logarithm is no harder than the ordinary discrete logarithm. However, the one sticking point is that in order to find two *independent* points of order $m$, the field $K$ must contain all $m$-th roots of 1. Almost all the time, this forces the field $K$ to be stupendously large – much too large to work with. One notable exception [20] is the case when $E$ is supersingular, in which case, the field $K$ need only be an extension of degree 2.

In [8] a similar analysis is given using the *Tate-pairing*. This analysis allows the extension field $K$ to be slightly smaller most of the time. However, $K$ still needs to contain all the $m$-th roots of unity, and so must be quite large almost all the time.

# 6 Elliptic Curves and other problems

As stated above, Lenstra [15] used the properties of elliptic curves in order to give a fast algorithm for factoring integers. Unlike other fast algorithm, such as the number field sieve [14], the smaller the smallest prime divisor is, the faster the algorithm runs.

In [16], Ueli Maurer, used elliptic curves to show how one could reduce the problem of calculating discrete logarithms to the Diffie-Hellman problem, for some prime moduli. Later work by Maurer and Wolf [17] and Boneh and Lipton [3] finished off this program, by making use of a reasonable conjecture about the distribution of the orders of elliptic curves (the same conjecture used by Lenstra in his factoring method).

# 7 Conclusion

The study of elliptic curves includes much beautiful and deep number theory. Until recently this study was almost exclusively the province of pure mathematicians. Now elliptic curves can claim their place as one of the important subjects in the study of cryptography. Not only are they useful theoretically, but are already having great practical impact.

# References


[1] Leonard M. Adleman and Jonathan DeMarrais. A subexponential algorithm for discrete logarithms over all finite fields. *Mathematics of Computation*, 61:1–15, 1993.

[2] Ian F. Blake, R. Fuji-Hara, R. C. Mullin, and Scott A. Vanstone. Computing logarithms in finite fields of characteristic two. *SIAM J. Algebraic and Discrete Methods*, 5(2):276–285, 1984.

[3] Dan Boneh and Richard Lipton. Algorithms for Black-Box fields and their application to cryptography. In Neal Koblitz, editor, *Advances in Cryptology – Crypto '96*, volume 1109 of *Lecture Notes in Computer Science*, pages 283–297, Berlin, Heidelberg, New York, 1996. Springer–Verlag.

[4] Don Coppersmith. Fast evaluation of logarithms in fields of characteristic two. *IEEE Trans. Inf. Thy.*, 30(4):587–594, 1984.

[5] Jean-Marc Couveignes and François Morain. Schoof's algorithm and isogeny cycles. In Leonard M. Adleman and Ming-Deh Huang, editors, *Algorithmic Number Theory*, volume 877 of *Lecture Notes in Computer Science*, pages 60–70, Berlin, Heidelberg, New York, 1994. Springer–Verlag.

[6] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE Trans. Inf. Thy.*, 22:644–654, 1976.

[7] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inf. Thy.*, 31:469–472, 1985.

[8] Gerhard Frey and Han-Georg Rück. A remark concerning $m$-divisibility and the discrete logarithm in the divisor class group of curves. *Mathematics of Computation*, 62(206):865–874, 1994.

[9] Martin E Hellman and Justin M. Reyneri. Fast computation of discrete logarithms in GF($q$). In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *Advances in Cryptology: Proceedings of Crypto '82*, New York, 1983. Plenum Press.

[10] Everett W. Howe. On the group orders of elliptic curves over finite fields. *Compositio Math.*, 85:229–247, 1993.

[11] Burton S. Kaliski. *Elliptic Curves and Cryptography: A Pseudorandom bit Generator and other Tools.* PhD thesis, MIT, Cambridge, January 1988. MIT/LCS/TR-411.

[12] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48:203–209, 1987.

[13] Frank Lehmann, Markus Maurer, Volker Müller, and Victor Shoup. Counting the number of points on elliptic curves over finite fields of characteristic greater than three. In Leonard M. Adleman and Ming-Deh Huang, editors, *Algorithmic Number Theory*, volume 877 of *Lecture Notes in Computer Science*, pages 60–70, Berlin, Heidelberg, New York, 1994. Springer–Verlag.

[14] Arjen K. Lenstra and Hendrik W. Lenstra, Jr., editors. *The development of the Number Field Sieve*, volume 1554 of *Lecture Notes in Mathematics*. Springer–Verlag, Berlin, Heidelberg, New York, 1993.

[15] Hendrik W. Jr. Lenstra. Factoring integers with elliptic curves. *Mathematics of Computation*, 126:649–673, 1987.

[16] Ueli Maurer. Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms. In Yvo Desmedt, editor, *Advances in Cryptology – Crypto '94*, volume 838 of *Lecture Notes in Computer Science*, pages 271–281, Berlin, Heidelberg, New York, 1994. Springer–Verlag.

[17] Ueli Maurer and Stefan Wolf. Diffie-Hellman oracles. In Neal Koblitz, editor, *Advances in Cryptology – Crypto '96*, volume 1109 of *Lecture Notes in Computer Science*, pages 268–282, Berlin, Heidelberg, New York, 1996. Springer–Verlag.

[18] Kevin S. McCurley. The discrete logarithm problem. In Carl Pomerance, editor, *Cryptology and Computational Number Theory*, volume 42 of *Proc. Symp. Appl. Math.*, pages 49–74. American Math. Soc., Providence, 1990.

[19] Alfred J. Menezes. *Elliptic Curve Public Key Cryptosystems*. International Series in Engineering and Computer Science. Kluwer Academic Publishers, Boston, Dordrecht, London, 1993.

[20] Alfred J. Menezes, T. Okamoto, and Scott A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Trans. Inf. Thy.*, 39(5):1639–1646, 1993.

[21] Alfred J. Menezes, Scott A. Vanstone, and Robert J. Zuccherato. Counting points on elliptic curves over $\mathbb{F}_{2^m}$. *Mathematics of Computation*, 60(201):407–420, 1993.

[22] Victor S. Miller. Short programs for functions on curves. IBM, Thomas J. Watson Research Center, 1986.

[23] Victor S. Miller. Use of elliptic curves in cryptography. In H. Williams, editor, *Advances in Cryptology – Crypto '85*, volume 218 of *Lecture Notes in Computer Science*, pages 417–426, Berlin, Heidelberg, New York, 1986. Springer–Verlag.

[24] V. I. Nechaev. Complexity of determinate algorithm for the discrete logarithm. *Mathematical Notes*, 55(2):165–172, 1994. Translated from *Matematicheskie Zametki*, 55(2)91–101, 1994.

[25] Andrew M. Odlyzko. Discrete logarithms and their cryptographic significance. In N. Cot T. Beth and I. Ingemarsson, editors, *Advances in Cryptology – EUROCRYPT '84*, volume 209 of *Lecture Notes in Computer Science*, pages 224–314, Berlin, Heidelberg, New York, 1985. Springer–Verlag.

[26] Steven Pohlig and Martin Hellman. An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance. *IEEE Trans. Inf. Thy.*, 24:106–110, 1978.

[27] John Pollard. Monte Carlo methods for index computation (mod p). *Mathematics of Computation*, 32:918–924, 1978.

[28] Ronald Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Comm. Assoc. Comput. Mach.*, 21:120–126, 1978.

[29] Hans-Georg Rück. A note on elliptic curves over finite fields. *Mathematics of Computation*, 49(179):301–304, July 1987.

[30] Oliver Schirokauer, Damian Weber, and Thomas Denny. Discrete logarithms: the effectiveness of the index calculus method. In Henri Cohen, editor, *Algorithmic Number Theory: ANTS II*, pages 327–352, Bordeaux, May 1996. Université Bordeaux.

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

[32] Rene Schoof. Elliptic curves over finite fields and the computation of square roots mod $p$. *Mathematics of Computation*, 44:483–494, 1985.

[33] Daniel Shanks. Class number, a theory of factorization, and genera. In Donald J. Lewis, editor, *1969 Number Theory Institute*, volume XX of *Proc. Symp. Pure Math.*, pages 415–440. American Math. Soc., Providence, 1971.

[34] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *Advances in Cryptology – Eurocrypt '97*, Lecture Notes in Computer Science, Berlin, Heidelberg, New York, May 1997. Springer–Verlag.

[35] Joseph H. Silverman. *The Arithmetic of Elliptic Curves*, volume 106 of *Graduate Texts in Mathematics*. Springer–Verlag, Berlin, Heidelberg, New York, first edition, 1986.

[36] Damian Weber. Computing discrete lograrithms with the general number field sieve. In Henri Cohen, editor, *Algorithmic Number Theory: ANTS II*, pages 377–389, Bordeaux, May 1996. Université Bordeaux.

[37] A. E. Western and J. C. P. Miller. *Tables of Indices and Primitive Roots*, volume 9 of *Royal Society Mathematical Tables*. Cambridge University Press, Cambridge, 1968.

[38] Andrew Wiles. Modular elliptic curves and Fermat's last theorem. *Annals of Math.*, 141(3):443–551, 1995.