Designs, Codes and Cryptography, 19, 173–193 (2000)

The State of Elliptic Curve Cryptography

NEAL KOBLITZ koblitz@math.washington.edu
Dept. of Mathematics, Box 354350, University of Washington, Seattle, WA 98195, USA.

ALFRED MENEZES ajmeneze@cacr.math.uwaterloo.ca
Dept. of C&O, University of Waterloo, Waterloo, Ontario, Canada, N2L 3G1.

SCOTT VANSTONE
Dept. of C&O, University of Waterloo, Waterloo, Ontario, Canada, N2L 3G1.

Abstract. Since the introduction of public-key cryptography by Diffie and Hellman in 1976, the potential for the use of the discrete logarithm problem in public-key cryptosystems has been recognized. Although the discrete logarithm problem as first employed by Diffie and Hellman was defined explicitly as the problem of finding logarithms with respect to a generator in the multiplicative group of the integers modulo a prime, this idea can be extended to arbitrary groups and, in particular, to elliptic curve groups. The resulting public-key systems provide relatively small block size, high speed, and high security. This paper surveys the development of elliptic curve cryptosystems from their inception in 1985 by Koblitz and Miller to present day implementations.

Keywords: Elliptic curves, public-key cryptography

1. Introduction

Since the introduction of public-key cryptography by Diffie and Hellman [14] in 1976, the cryptographic importance of the apparent intractability of the discrete logarithm problem has been recognized. ElGamal [16] first described how this problem may be utilized in public-key encryption and digital signature schemes. ElGamal's methods have been refined and incorporated into various protocols to meet a variety of applications, and one of its extensions forms the basis for the U.S. government digital signature algorithm (DSA) [56].

Although the discrete logarithm problem as first employed by Diffie and Hellman in their key agreement protocol was defined explicitly as the problem of finding logarithms with respect to a generator in the multiplicative group of the integers modulo a prime, this idea can be extended to arbitrary groups. Let G be a finite group of order n, and let α be an element of G. The *discrete logarithm problem* for G is the following: given an element $\beta \in G$, find an integer x, $0 \leq x \leq n-1$, such that $\alpha^x = \beta$, if such an integer exists (i.e., if β is in the subgroup of G generated by α). Groups that have been proposed for cryptographic use include the multiplicative group of characteristic two finite fields (see, for example, Agnew *et al* [2]), subgroups of the multiplicative group of the integers modulo a prime (Schnorr [68]), the group of units of Z_n where n is a composite integer (McCurley [46]), the group of points on an elliptic curve defined over a finite field (Koblitz [29] and Miller [52]), the jacobian of a hyperelliptic curve defined over a finite field (Koblitz [31]), and the class group of an imaginary quadratic number field (Buchmann and Williams [9]).

Elliptic curves have been extensively studied for over a hundred years, and there is a vast literature on the topic. Originally pursued mainly for aesthetic reasons, elliptic curves have recently become a tool in several important applied areas, including coding theory (Driencourt and Michon [15] and van der Geer [19]); pseudorandom bit generation (Kaliski [26, 27]); and number theory algorithms (Goldwasser and Kilian [20] for primality proving and Lenstra [41] for integer factorization).

In 1985, Koblitz [29] and Miller [52] independently proposed using the group of points on an elliptic curve defined over a finite field in discrete log cryptosystems. The primary advantage that elliptic curve systems have over systems based on the multiplicative group of a finite field (and also over systems based on the intractability of integer factorization) is the absence of a subexponential-time algorithm (such as those of "index-calculus" type) that could find discrete logs in these groups. Consequently, one can use an elliptic curve group that is smaller in size while maintaining the same level of security. The result is smaller key sizes, bandwidth savings, and faster implementations, features which are especially attractive for security applications where computational power and integrated circuit space is limited, such as smart cards, PC (personal computer) cards, and wireless devices.

Elliptic curves also appear in the so-called elliptic curve analogues of the RSA cryptosystem, as first proposed by Koyama *et al* [38]. In these systems, one works in an elliptic curve defined over the ring Z_n (n a composite integer), and the order of the elliptic curve group serves as the trapdoor. The security of these schemes is based on the difficulty of factoring n. The work of several people, including Kurosawa, Okada, and Tsujii [39], Pinch [61], Kaliski [28], and Bleichenbacher [7] subsequently showed that these elliptic curve analogues do not have any significant advantages over their RSA counterparts. For this reason, they are not considered in this paper.

The remainder of the paper is organized as follows. §2 begins with a brief review of elliptic curves. For an elementary introduction to elliptic curves, the reader is referred to Chapter 6 of Koblitz's books [36, 37]. Charlap and Robbins [10, 11] present elementary self-contained proofs for some of the basic theory. For more sophisticated treatments, see Silverman [73, 74]. The elliptic curve analogues of discrete log cryptosystems are discussed in §3. §4 studies the elliptic curve discrete logarithm problem, whose apparent intractability is the basis for the security of elliptic curve systems. §5 considers various issues that arise in implementation.

We will use the following notation. F_q denotes the finite field of q elements and $\overline{\mathrm{F}}_q$ denotes the algebraic closure of F_q. By Z_n we denote the integers modulo n. The cardinality of a set S is denoted by $\#S$.

2. Background on Elliptic Curves

Assume first that F_q has characteristic greater than 3. An *elliptic curve* E over F_q is the set of all solutions $(x, y) \in \overline{\mathrm{F}}_q \times \overline{\mathrm{F}}_q$ to an equation

$$y^2 = x^3 + ax + b, \tag{1}$$

where $a, b \in \mathrm{F}_q$ and $4a^3 + 27b^2 \neq 0$, together with a special point ∞ called the *point at infinity*.

It is well known that E is an (additively written) abelian group with the point ∞ serving as its identity element. The rules for group addition are summarized below.

Addition Formulas for the Curve (1). Let $P = (x_1, y_1) \in E$; then $-P = (x_1, -y_1)$. If $Q = (x_2, y_2) \in E$, $Q \neq -P$, then $P + Q = (x_3, y_3)$, where

$$\begin{aligned} x_3 &= \lambda^2 - x_1 - x_2 \\ y_3 &= \lambda(x_1 - x_3) - y_1, \end{aligned}$$

and

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & \text{if } P \neq Q \\ \dfrac{3x_1^2 + a}{2y_1} & \text{if } P = Q. \end{cases}$$

If F_q is a field of characteristic 2, then there are two types of elliptic curves over F_q. An *elliptic curve E of zero j-invariant* over F_q is the set of all solutions $(x, y) \in \overline{\mathrm{F}}_q \times \overline{\mathrm{F}}_q$ to an equation

$$y^2 + cy = x^3 + ax + b, \tag{2}$$

where $a, b, c \in \mathrm{F}_q$, $c \neq 0$, together with the point at infinity ∞. An *elliptic curve E of non-zero j-invariant* over a field F_q of characteristic 2 is the set of solutions $(x, y) \in \overline{\mathrm{F}}_q \times \overline{\mathrm{F}}_q$ to an equation

$$y^2 + xy = x^3 + ax^2 + b, \tag{3}$$

where $a, b \in \mathrm{F}_q$, $b \neq 0$, together with the point at infinity ∞. In both cases, E is an (additively written) abelian group with the point ∞ serving as the identity. The addition formulas for the two types of curves over F_{2^m} are given below.

Addition Formulas for the Curve (2). Let $P = (x_1, y_1) \in E$; then $-P = (x_1, y_1 + c)$. If $Q = (x_2, y_2) \in E$ and $Q \neq -P$, then $P + Q = (x_3, y_3)$, where

$$x_3 = \begin{cases} \left(\dfrac{y_1 + y_2}{x_1 + x_2}\right)^2 + x_1 + x_2 & P \neq Q \\ \dfrac{x_1^4 + a^2}{c^2} & P = Q \end{cases}$$

and

$$y_3 = \begin{cases} \left(\dfrac{y_1 + y_2}{x_1 + x_2}\right)(x_1 + x_3) + y_1 + c & P \neq Q \\ \left(\dfrac{x_1^2 + a}{c}\right)(x_1 + x_3) + y_1 + c & P = Q. \end{cases}$$

Addition Formulas for the Curve (3). Let $P = (x_1, y_1) \in E$; then $-P = (x_1, y_1 + x_1)$. If $Q = (x_2, y_2) \in E$ and $Q \neq -P$, then $P + Q = (x_3, y_3)$, where

$$x_3 = \begin{cases} \left(\dfrac{y_1 + y_2}{x_1 + x_2}\right)^2 + \dfrac{y_1 + y_2}{x_1 + x_2} + x_1 + x_2 + a & P \neq Q \\ x_1^2 + \dfrac{b}{x_1^2} & P = Q \end{cases}$$

and

$$y_3 = \begin{cases} \left(\dfrac{y_1 + y_2}{x_1 + x_2}\right)(x_1 + x_3) + x_3 + y_1 & P \neq Q \\ x_1^2 + \left(x_1 + \dfrac{y_1}{x_1}\right) x_3 + x_3 & P = Q. \end{cases}$$

If E is an elliptic curve over a finite field F_q, then let $E(\mathrm{F}_q)$ denote the points in E having both coordinates in F_q, including the point ∞; the points in $E(\mathrm{F}_q)$ are also known as F_q-*rational points*. $E(\mathrm{F}_q)$ is an abelian group of rank 1 or 2. We have $E(\mathrm{F}_q) \cong C_{n_1} \oplus C_{n_2}$, where C_n denotes the cyclic group of order n, n_2 divides n_1, and furthermore $n_2 | q - 1$. A well-known theorem of Hasse states that $\#E(\mathrm{F}_q) = q + 1 - t$, where $|t| \leq 2\sqrt{q}$. The curve E is said to be *supersingular* if $t^2 = 0, q, 2q, 3q$, or $4q$; otherwise the curve is *non-supersingular*.

If q is a power of 2 and E is supersingular, then $\#E(\mathrm{F}_q)$ is odd; if q is a power of 2 and E is non-supersingular, then $\#E(\mathrm{F}_q)$ is even. A result of Waterhouse [81] states that if q is a prime, then for each t satisfying $|t| \leq 2\sqrt{q}$ there exists at least one elliptic curve E defined over F_q with $\#E(\mathrm{F}_q) = q + 1 - t$; if q is a power of 2, then for each odd t satisfying $|t| \leq 2\sqrt{q}$ there exists at least one (non-supersingular) elliptic curve E defined over F_q with $\#E(\mathrm{F}_q) = q + 1 - t$. More generally, Schoof [70] derived a formula for the number of isomorphism classes of elliptic curves defined over F_q with $\#E(\mathrm{F}_q) = q + 1 - t$, for each t satisfying $|t| \leq 2\sqrt{q}$.

Example (*elliptic curve over* Z_{23}). Consider the elliptic curve E: $y^2 = x^3 + x + 1$ defined over Z_{23}. Then $\#E(\mathrm{Z}_{23}) = 28$, $E(\mathrm{Z}_{23})$ is cyclic, and a generator of $E(\mathrm{Z}_{23})$ is $P = (0, 1)$. The points in $E(\mathrm{Z}_{23})$, expressed as multiples of P, are shown below:

$P = (0, 1)$	$2P = (6, -4)$	$3P = (3, -10)$	$4P = (-10, -7)$
$5P = (-5, 3)$	$6P = (7, 11)$	$7P = (11, 3)$	$8P = (5, -4)$
$9P = (-4, -5)$	$10P = (12, 4)$	$11P = (1, -7)$	$12P = (-6, -3)$
$13P = (9, -7)$	$14P = (4, 0)$	$15P = (9, 7)$	$16P = (-6, 3)$
$17P = (1, 7)$	$18P = (12, -4)$	$19P = (-4, 5)$	$20P = (5, 4)$
$21P = (11, -3)$	$22P = (7 - 11)$	$23P = (-5, -3)$	$24P = (-10, 7)$
$25P = (3, 10)$	$26P = (6, 4)$	$27P = (0, -1)$	$28P = \infty$.

□

Example (*elliptic curve over* F_{2^3}). Consider the elliptic curve E: $y^2 + xy = x^3 + x^2 + 1$ defined over F_{2^3}. F_{2^3} is constructed using the primitive irreducible polynomial $f(x) =$

$x^3 + x + 1$ and a root α. Then $\#E(\mathrm{F}_{2^3}) = 14$, and $E(\mathrm{F}_{2^3})$ is cyclic. A generator of $E(\mathrm{F}_{2^3})$ is $P = (\alpha, \alpha^5)$. The points in $E(\mathrm{F}_{2^3})$, expressed as multiples of P, are shown below:

$$\begin{array}{llll} P = (\alpha, \alpha^5) & 2P = (\alpha^3, 0) & 3P = (\alpha^2, \alpha^5) & 4P = (\alpha^5, 0) \\ 5P = (\alpha^4, \alpha^3) & 6P = (\alpha^6, \alpha^6) & 7P = (0, 1) & 8P = (\alpha^6, 0) \\ 9P = (\alpha^4, \alpha^6) & 10P = (\alpha^5, \alpha^5) & 11P = (\alpha^2, \alpha^3) & 12P = (\alpha^3, \alpha^3) \\ 13P = (\alpha, \alpha^6) & 14P = \infty. & & \end{array}$$

□

3. Elliptic Curve Cryptosystems

Discrete log cryptosystems are typically described in the setting of the multiplicative group of the integers modulo a prime p. Such systems can be modified to work in the group of points on an elliptic curve. For instance, the Diffie–Hellman key agreement protocol can be adapted for elliptic curves as follows. First note that a "random" point on an elliptic curve E can serve as a key, since Alice and Bob can agree in advance on a method to convert it to an integer (for example, they can take the image of its x-coordinate under some agreed upon simple map from F_q to the natural numbers).

So suppose that E is an elliptic curve over F_q, and Q is an agreed upon (and publicly known) point on the curve. Alice secretly chooses a random integer k_A and computes the point $k_A Q$, which she sends to Bob. Likewise, Bob secretly chooses a random k_B, computes $k_B Q$, and sends it to Alice. The common key is $P = k_A k_B Q$. Alice computes P by multiplying the point she received from Bob by her secret k_A; Bob computes P by multiplying the point he received from Alice by his secret k_B. An eavesdropper who wanted to spy on Alice and Bob would have to determine $P = k_A k_B Q$ knowing Q, $k_A Q$, and $k_B Q$, but not k_A or k_B. The eavesdropper's task is called the "Diffie–Hellman problem for elliptic curves."

It is not hard to modify the Diffie–Hellman protocol for the purpose of message transmission, using an idea of ElGamal [16]. Suppose that the set of message units has been embedded in E in some agreed upon way, and Bob wants to send Alice a message $M \in E$. Alice and Bob have already exchanged $k_A Q$ and $k_B Q$ as in Diffie–Hellman. Bob now chooses another secret random integer l, and sends Alice the pair of points $(lQ, M + l(k_A Q))$. To decipher the message, Alice multiplies the first point in the pair by her secret k_A and then subtracts the result from the second point in the pair.

We next describe the elliptic curve analogue (ECDSA) of the U.S. government digital signature algorithm (DSA). The ECDSA is an ANSI standard and is also being considered by the ANSI X9F1 and IEEE P1363 standards committees as a digital signature standard (see §5.3).

ECDSA Key Generation. E is an elliptic curve defined over F_q, and P is a point of prime order n in $E(\mathrm{F}_q)$; these are system-wide parameters. For simplicity, we shall suppose that q is a prime, although the construction can easily be adapted to a prime power q as well. Each entity A does the following:

1. Select a random integer d in the interval $[1, n-1]$.

2. Compute $Q = dP$.
3. A's public key is Q; A's private key is d.

ECDSA Signature Generation. To sign a message m, A does the following:

1. Select a random integer k in the interval $[1, n-1]$.
2. Compute $kP = (x_1, y_1)$ and $r = x_1 \bmod n$ (where x_1 is regarded as an integer between 0 and $q-1$). If $r = 0$ then go back to step 1.[1]
3. Compute $k^{-1} \bmod n$.
4. Compute $s = k^{-1}\{h(m) + dr\} \bmod n$, where h is the Secure Hash Algorithm (SHA-1 [57]). If $s = 0$, then go back to step 1.[2]
5. The signature for the message m is the pair of integers (r, s).

ECDSA Signature Verification. To verify A's signature (r, s) on m, B should do the following:

1. Obtain an authenticated copy of A's public key Q.
2. Verify that r and s are integers in the interval $[1, n-1]$.
3. Compute $w = s^{-1} \bmod n$ and $h(m)$.
4. Compute $u_1 = h(m)w \bmod n$ and $u_2 = rw \bmod n$.
5. Compute $u_1P + u_2Q = (x_0, y_0)$ and $v = x_0 \bmod n$.
6. Accept the signature if and only if $v = r$.

Discussion. The only significant difference between ECDSA and DSA is in the generation of r. The DSA does this by taking the random element $(\alpha^k \bmod p)$ and reducing it modulo q, thus obtaining an integer in the interval $[1, q-1]$. (In the DSA, q is a 160-bit prime divisor of $p-1$, and α is an element of order q in F_p^*.) The ECDSA generates the integer r in the interval $[1, n-1]$ by taking the x-coordinate of the random point kP and reducing it modulo n.

To obtain a security level similar to that of the DSA, the parameter n should have about 160 bits. If this is the case, then DSA and ECDSA signatures have the same bitlength (320 bits).

Instead of using system-wide parameters, we could fix the underlying finite field F_q for all entities, and let each entity select its own elliptic curve E and point $P \in E(\mathrm{F}_q)$. In this case, the defining equation for E, the point P, and the order n of P must also be included in the entity's public key. If the underlying field F_q is fixed, then hardware or software can be built to optimize computations in that field. At the same time, there are an enormous number of choices of elliptic curves E over the fixed F_q.

4. Security

The basis for the security of elliptic curve cryptosystems such as the ECDSA is the apparent intractability of the following *elliptic curve discrete logarithm problem* (ECDLP): given an elliptic curve E defined over F_q, a point $P \in E(\mathrm{F}_q)$ of order n, and a point $Q \in E(\mathrm{F}_q)$, determine the integer l, $0 \leq l \leq n-1$, such that $Q = lP$, provided that such an integer exists.

The Pohlig–Hellman algorithm [62] reduces the determination of l to the determination of l modulo each of the prime factors of n. Hence, in order to achieve the maximum possible security level, n should be prime. The best algorithm known to date for ECDLP is the Pollard ρ-method [63], as modified by Gallant, Lambert and Vanstone [18], and Wiener and Zuccherato [82], which takes about $(\sqrt{\pi n})/2$ steps, where a *step* here is an elliptic curve addition. Van Oorschot and Wiener [59, 60] showed how the Pollard ρ-method can be parallelized so that if r processors are used, then the expected number of steps by each processor before a single discrete logarithm is obtained is $(\sqrt{\pi n})/(2r)$. For elliptic curves E defined over a subfield F_{2^l} of F_{2^m}, the parallelized Pollard ρ-method for the ECDLP in $E(\mathrm{F}_{2^m})$ can be sped up to an expected running time of $(\sqrt{\pi nl/m})/(2r)$ (see [18, 82]).

An elliptic curve E over F_p is said to be *prime-field-anomalous* if $\#E(\mathrm{F}_p) = p$. Semaev [72], Smart [77] and Satoh and Araki [64] independently showed how to efficiently compute an isomorphism between $E(\mathrm{F}_p)$, where E is a prime-field-anomalous curve, and the *additive* group of F_p. This gives a polynomial-time algorithm for the ECDLP in $E(\mathrm{F}_p)$. The attack does not appear to extend to any other class of elliptic curves. Consequently, by verifying that the number of points on an elliptic curve does not equal the cardinality of the underlying field, one can easily ensure that the Semaev–Smart–Satoh–Araki attack does not apply.

Menezes, Okamoto and Vanstone (MOV) ([49]; see also Menezes [48]) used the Weil pairing on an elliptic curve E to embed the group $E(\mathrm{F}_q)$ in the multiplicative group of the field F_{q^k} for some integer k. This reduces the ECDLP in $E(\mathrm{F}_q)$ to the discrete logarithm problem (DLP) in $\mathrm{F}_{q^k}^*$. A necessary condition for $E(\mathrm{F}_q)$ to be embedded in $\mathrm{F}_{q^k}^*$ is that n divide $q^k - 1$; and in [5] it is proved that this condition is also sufficient under a mild assumption.[3] Now in $\mathrm{F}_{q^k}^*$ we can hope to use a version of the index-calculus algorithm with subexponential running time

$$\exp((c + o(1))(\log q^k)^{1/3}(\log\log q^k)^{2/3}). \tag{4}$$

See Coppersmith [12] for the case when q a power of 2, and Gordon [21] and Schirokauer [67] for the case when q is a prime and $k = 1$. No algorithm with running time (4) is known when q is odd and $k > 1$, but we adopt the "optimistic" supposition that the time estimate (4) is the complexity of the discrete logarithm problem in $\mathrm{F}_{q^k}^*$ for all q and $k \geq 1$.

Note that k must be less than $\log^2 q$, since otherwise the index-calculus algorithm for $\mathrm{F}_{q^k}^*$ will take fully exponential time (in $\log q$). For the very special class of supersingular curves, it is known that $k \leq 6$. For these curves, the MOV reduction gives a subexponential-time algorithm for the ECDLP. However, a randomly generated elliptic curve has an exponentially small probability of being supersingular; and, as shown by Koblitz [33] (see also Balasubramanian and Koblitz [5]), for most randomly generated elliptic curves we have $k > \log^2 q$.

No subexponential-time algorithm is known for the ECDLP for any class of elliptic curves other than the ones discussed above. Miller [52] discusses the index-calculus method as it might apply to elliptic curve groups. He comments that unlike in the case of F_q^*, where there are natural candidates for the factor base Γ (prime numbers of small size or small degree irreducible polynomials), there appear to be no likely candidates in $E(\mathrm{F}_q)$. The most natural ones for elliptic curves over F_p seem to be points of small height in $E(\mathrm{Q})$, Q the field of rational numbers (the height of a point is related to the number of bits needed to represent the point). However, Miller points out that there are very few points of small height in $E(\mathrm{Q})$. Furthermore, even if such a set Γ exists, finding an efficient method for lifting a point in $E(\mathrm{F}_p)$ to a point in $E(\mathrm{Q})$ looks hopeless. Miller's argument against the possibility of index-calculus attacks has been elaborated on and explored in more detail by J. Silverman and Suzuki [76], who support his conclusions.

A very interesting line of attack on the ECDLP was recently proposed by J. Silverman [75]. His "xedni calculus" turns the index calculus method "on its head" (hence the name). Given a discrete log problem on an elliptic curve over F_p, he first lifts the points in question (actually, r different integer linear combinations of them, where $r \leq 9$) to points in the plane over Q, and then he considers elliptic curves $E(\mathrm{Q})$ that pass through these r points. If $E(\mathrm{Q})$ can be chosen to have rank $< r$ — i.e., so that there is an integer linear dependence relation among the r points — then the ECDLP is solved. In general, the probability of rank $< r$ is negligible. However, Silverman's idea is to impose a number of "Mestre conditions" modulo ℓ for small primes ℓ in order to increase this probability. (Each Mestre condition [51] forces $\#E(\mathrm{F}_\ell)$ to be as small as possible.) Although the xedni calculus attack is clever and elegant, a careful analysis [25] showed that it is extremely impractical. One intriguing aspect of Silverman's algorithm is that it can be adapted (with no important changes) to solve both the discrete log problem in the multiplicative group of F_p and the integer factorization problem. Thus, if it had turned out to be efficient, it would have attacked all major public-key cryptosystems that are in practical use.

Other work has treated problems that are related to the ECDLP. Frey and Rück [17] used a variant of the Tate pairing for abelian varieties over local fields to extend the MOV reduction algorithm to jacobian groups of curves of genus g over finite fields. Adleman, DeMarrais and Huang [1] (see also Stein, Müller and Thiel [80]) presented a subexponential-time algorithm for the discrete logarithm problem in the jacobian of a large genus hyperelliptic curve over a finite field. More precisely, there exists a number c, $0 < c \leq 2.181$, such that for all sufficiently large $g \geq 1$ and all odd primes p with $\log p \leq (2g+1)^{0.98}$, the expected running time of the algorithm for computing logarithms in the jacobian of a genus g hyperelliptic curve over F_p is conjectured to be

$$\exp((c + o(1))(\log p^{2g+1})^{1/2}(\log\log p^{2g+1})^{1/2}).$$

However, in the case of elliptic curves (which are hyperelliptic curves of genus $g = 1$) the algorithm is worse than naive exhaustive search.

In 1994, Scheidler, Buchmann and Williams [65] used a non-group structure, the so-called infrastructure of the principal ideals of a real quadratic number field, to implement the Diffie–Hellman key agreement protocol. To overcome some difficulties with implementing such a scheme, Scheidler, Stein and Williams [66] extended the ideas to (odd

Table 1. Computing power needed to compute elliptic curve logarithms with the Pollard ρ-method.

Field size (in bits)	Size of n (in bits)	$(\sqrt{\pi n})/2$	MIPS years
163	160	2^{80}	8.5×10^{11}
191	186	2^{93}	7.0×10^{15}
239	234	2^{117}	1.2×10^{23}
359	354	2^{177}	1.3×10^{41}
431	426	2^{213}	9.2×10^{51}

characteristic) real quadratic congruence function fields; see also Müller, Vanstone and Zuccherato [54] for the case of even characteristic quadratic congruence function fields. Stein [79] (and Zuccherato [85] in the case of even characteristic) showed that the discrete logarithm problem in real quadratic congruence function fields of genus 1 is equivalent to the ECDLP. No subexponential-time algorithm is known for the former problem.

The security of the elliptic curve Diffie–Hellman key agreement protocol relies on the intractability of the elliptic curve Diffie–Hellman problem (ECDHP): given an elliptic curve E defined over F_q and points P, k_1P, $k_2P \in E(\mathrm{F}_q)$, compute the point k_1k_2P. Clearly ECDHP polynomial-time reduces to ECDLP. Boneh and Lipton [8] proved that if the ECDLP cannot be solved in subexponential time, then neither can ECDHP.

Software Attacks. We assume that a million-instructions-per-second (MIPS) machine can perform 4×10^4 elliptic curve additions per second, i.e., about 2^{40} elliptic curve additions per year. (This estimate is indeed conservative – an application-specific integrated circuit (ASIC) for performing elliptic curve additions over the field $\mathrm{F}_{2^{155}}$ (see [3]) has a 40 MHz clock-rate and can perform roughly 40,000 elliptic curve operations per second. Also, the software implementation by Schroeppel *et al* [71] on a SPARC IPC (rated at 25 MIPS) performs 2,000 elliptic curve additions per second.) The term *MIPS year* denotes the computational power of a MIPS computer utilized for one year. Table 1 shows the computing power required for various values of n to compute a single discrete logarithm using the Pollard ρ-method.

For instance, if 10,000 computers each rated at 1,000 MIPS are available, and $n \approx 2^{160}$, then a single elliptic curve discrete logarithm can be computed in 85,000 years. Odlyzko [58] has estimated that if 0.1% of the world's computing power were available for one year to work on a collaborative effort to break some challenge cipher, then the computing power available would be 10^8 MIPS years in 2004 and between 10^{10} and 10^{11} MIPS years in 2014.

To put the numbers in Table 1 in some perspective, Table 2 (due to Odlyzko [58]) shows the estimated computing power required to factor integers with current versions of the general number field sieve.

Hardware Attacks. For well-funded attackers, a more promising approach might be to build special-purpose hardware for a parallel search using the Pollard ρ-method. Van Oorschot and Wiener [59] provide a detailed study of such a possibility. In their 1994 study, they

Table 2. Computing power needed to factor integers using the general number field sieve.

Bitsize of integer to be factored	MIPS years
512	3×10^4
768	2×10^8
1024	3×10^{11}
1280	1×10^{14}
1536	3×10^{16}
2048	3×10^{20}

estimated that if $n \approx 10^{36} \approx 2^{120}$, then a machine with $m = 325{,}000$ processors that could be built for about US$10 million would compute a single discrete logarithm in about 35 days.

Discussion. It should be pointed out that in the software and hardware attacks described above, computation of a single elliptic curve discrete logarithm has the effect of revealing a *single* user's private key. Roughly the same effort must be repeated in order to determine another user's private key.

In [6], Blaze *et al* report on the minimum key lengths required for secure symmetric-key encryption schemes. They come to the following conclusions:

> To provide adequate protection against the most serious threats – well-funded commercial enterprises or government intelligence agencies – keys used to protect data today should be at least 75 bits long. To protect information adequately for the next 20 years in the face of expected advances in computing power, keys in newly-deployed systems should be at least 90 bits long.

Extrapolating these conclusions to the case of elliptic curves, we see that n should be at least 150 bits for short-term security and at least 180 bits for medium-term security. This extrapolation is justified by the following considerations:

1. Exhaustive search through a k-bit symmetric-key cipher takes about the same time as the Pollard ρ-algorithm applied to an elliptic curve having a $2k$-bit parameter n.

2. Exhaustive searches with a symmetric-key cipher and the Pollard ρ-algorithm can both be parallelized with a linear speedup.

3. A basic operation with elliptic curves (addition of two points) is computationally more expensive than a basic operation in a symmetric-key cipher (encryption of one block).

4. In both symmetric-key ciphers and elliptic curve systems, a "break" has the same effect: it recovers a single private key.

5. Implementation Issues

Since the elliptic curve discrete logarithm problem appears to be harder that the discrete logarithm problem in F_p^* (or the problem of factoring a composite integer n), one can use an elliptic curve group that is significantly smaller that F_p^* (respectively, n). For example, an elliptic curve $E(\mathrm{F}_q)$ with a point $P \in E(\mathrm{F}_q)$ whose order is a 160-bit prime offers approximately the same level of security as DSA with a 1024-bit modulus p and RSA with a 1024-bit modulus n.

In order to get a rough idea of the computational efficiency of elliptic curve systems, let us compare the times to compute

(i) kP where $P \in E(\mathrm{F}_{2^m})$, E is a non-supersingular curve, $m \approx 160$, and k is a random 160-bit integer (this is an operation in ECDSA); and

(ii) $\alpha^k \bmod p$, where p is a 1024-bit prime and k is a random 160-bit integer (this is an operation in DSA).

Let us assume that a field multiplication in F_q, where $\log_2 q = l$, takes l^2 bit operations; then a modular multiplication in (ii) takes $(1024/160)^2 \approx 41$ times longer than a field multiplication in (i). Computation of kP by repeated doubling and adding on the average requires 160 elliptic curve doublings and 80 elliptic curve additions. From the addition formula for non-supersingular curves (see §2), we see that an elliptic curve addition or doubling requires 1 field inversion and 2 field multiplications. (The cost of field addition is negligible, as is the cost of a field squaring especially if a normal basis representation is used.) Assume also that the time to perform a field inversion is equivalent to that of 3 field multiplications (this is what has been reported in practice; see Schroeppel *et al* [71] and De Win *et al* [83]). Hence, computing kP requires the equivalent of 1200 field multiplications, or $1200/41 \approx 29$ 1024-bit modular multiplications. On the other hand, computing $\alpha^k \bmod p$ by repeated squaring and multiplying requires an average of 240 1024-bit modular multiplications. Thus, the operation in (i) can be expected to be about 8 times faster than the operation in (ii).[4] Since multiplication in F_{2^m} is in fact substantially faster than modular multiplication, even more impressive speedups can be realized in practice.

Another important consequence of using a smaller group in elliptic curve systems is that low-cost and low-power implementations are feasible in restricted computing environments, such as smart cards, pagers, hand-held computers, and cellular telephones. For example, an ASIC built for performing elliptic curve operations over the field $\mathrm{F}_{2^{155}}$ (see Agnew, Mullin and Vanstone [3]) has only 12,000 gates and would occupy less that 5% of the area typically designated for a smart card processor. By comparison, a chip designed to do modular multiplication of 512-bit numbers (see Ivey *et al* [24]) has about 50,000 gates, while the chip designed to do field multiplications in $\mathrm{F}_{2^{593}}$ (see Agnew *et al* [2]) has about 90,000 gates.

Another advantage of elliptic curve systems is that the underlying field F_q and a representation for its elements can be selected so that the field arithmetic (addition, multiplication, and inversion) can be optimized. This is not the case for systems based on discrete log (re-

spectively, integer factorization), where the prime modulus p (respectively, the composite modulus n) should not be chosen to have a special form that would be likely to make the cryptanalyst's task easier (using the number field sieve).

With our current knowledge, elliptic curve systems over prime order fields F_p appear to provide the same level of security as elliptic curve systems over characteristic two fields F_{2^m} when $p \approx 2^m$. Because it appears that arithmetic in F_{2^m} can be implemented more efficiently in hardware and software than arithmetic in F_p (on platforms where specialized arithmetic co-processors for performing the finite field arithmetic are not available), elliptic curves over F_{2^m} have seen wider use in commercial implementations.

Construction of an elliptic curve cryptosystem requires some basic steps:

1. Selecting an underlying field F_q.
2. Selecting a representation for the elements of F_q.
3. Implementing the arithmetic in F_q.
4. Selecting an appropriate elliptic curve E over F_q.
5. Implementing the elliptic curve operations in E.

§5.1 surveys some of the field representations used in elliptic curve implementations that have been reported in the literature. Techniques for selecting suitable elliptic curves are discussed in §5.2. Finally, §5.3 summarizes the current efforts underway to standardize elliptic curve cryptosystems.

5.1. Representation of the Underlying Field

The representation used for the elements of the underlying field F_q can have a significant impact on the feasibility, cost, and speed of an elliptic curve system. It must be emphasized, however, that the representation used for a particular field F_q does not appear to affect its security.

Elliptic Curves over F_p. To minimize the time to perform modular multiplication, the prime p may be chosen to be of the form $p = 2^k - 1$ (called a *Mersenne prime*); see the patent of Crandall [13]. See De Win *et al* [84] for a report of a software implementation of ECDSA over F_p, and Bailey and Paar [4] for an implementation report of elliptic curve arithmetic over finite fields F_{p^m} where p is of the form $2^k \pm c$ for some small c.

Elliptic Curves over F_{2^m}. The field F_{2^m} can be viewed as a vector space of dimension m over F_2. That is, there exists a set of m elements $\{\alpha_0, \alpha_1, \ldots, \alpha_{m-1}\}$ in F_{2^m} such that each $\alpha \in F_{2^m}$ can be written uniquely in the form

$$\alpha = \sum_{i=0}^{m-1} a_i \alpha_i, \quad \text{where } a_i \in \{0, 1\}.$$

We can then represent α as the binary vector $(a_0, a_1, \ldots, a_{m-1})$. Addition of field elements is performed by bitwise XOR-ing the vector representations. There are many different bases of F_{2^m} over F_2.

1. *Trinomial bases*

   If $f(x)$ is an irreducible polynomial of degree m over F_2, then the field F_{2^m} can be represented as the set of polynomials of degree less than m over F_2, where multiplication of polynomials is performed modulo $f(x)$. That is, in the above notation $\alpha_i = x^i$, $0 \leq i \leq m-1$. Such a representation is called a *polynomial basis representation*. A *trinomial basis representation* is a polynomial basis representation in which the polynomial $f(x)$ has the form $f(x) = x^m + x^k + 1$. Such representations have the advantage that reduction modulo $f(x)$ can be performed efficiently, both in software and in hardware. For a detailed description of the field arithmetic in $F_{2^{155}}$ using a trinomial basis representation, see Schroeppel *et al* [71].

2. *Optimal normal bases*

   A *normal basis* of F_{2^m} over F_2 is a basis of the form

   $$\{\beta, \beta^2, \beta^{2^2}, \ldots, \beta^{2^{m-1}}\},$$

   where $\beta \in F_{2^m}$; such a basis always exists. Since squaring is a linear operator in F_{2^m}, we have

   $$\alpha^2 = \sum_{i=0}^{m-1} a_i \beta^{2^{i+1}} = \sum_{i=0}^{m-1} a_{i-1} \beta^{2^i} = (a_{m-1}, a_0, \ldots, a_{m-2}).$$

   Thus, a normal basis representation of F_{2^m} has the advantage that squaring a field element is accomplished by a simple rotation of the vector representation, an operation that is easily implemented in hardware.

   Multiplication in a normal basis representation is more complicated. The so-called *optimal normal bases*[5] (see Mullin *et al* [55]) appear to give the most efficient implementation of field arithmetic (with respect to both speed and complexity of hardware architecture). For a report on a hardware implementation of an elliptic curve cryptosystem over $F_{2^{155}}$ using an optimal normal basis, see Agnew, Mullin and Vanstone [3].

   Another advantage of normal bases is that square roots of elements in F_{2^m} can be efficiently computed. This is useful for recovering points when using the following compression technique. Let $P = (x_1, y_1)$ be a point on the elliptic curve $y^2 + xy = x^3 + ax^2 + b$ defined over F_{2^m}. Define $\widetilde{y_1}$ to be 0 if $x_1 = 0$; if $x_1 \neq 0$, then $\widetilde{y_1}$ is defined to be the rightmost bit of the field element $y_1 x_1^{-1}$. P can now be represented as $(x_1, \widetilde{y_1})$. Given x_1 and $\widetilde{y_1}$, y_1 can be recovered using the following technique from Menezes and Vanstone [50]. First, if $x_1 = 0$, then $y_1 = \sqrt{b}$. If $x_1 \neq 0$, then the change of variables $(x, y) \rightarrow (x, xz)$ transforms the curve equation to $z^2 + z = x + a + bx^{-2}$. Compute $\alpha = x_1 + a + bx_1^{-2}$. To solve the quadratic equation $z^2 + z = \alpha$, let $z = (z_0, z_1, \ldots, z_{m-1})$ and $\alpha = (a_0, a_1, \ldots, a_{m-1})$ be the vector representations of z and α, respectively. Then $z^2 + z = (z_{m-1} + z_0, z_0 + z_1, \ldots, z_{m-2} + z_{m-1})$. Each choice

$z_0 = 0$ or $z_0 = 1$ uniquely determines a solution $\overline{z}$ to $z^2 + z = \alpha$, by comparing the components of $z^2 + z$ and α. The correct solution $\overline{z}$ is selected by comparison with the bit $\widetilde{y_1}$. Finally, y_1 is recovered as $y_1 = x_1\overline{z}$.

3. *Using subfields*

   Suppose that $m = lr$, where l is small (e.g., $l = 8$ or $l = 16$). Then the field F_{2^m} can be viewed as an extension field of degree r over F_{2^l}. If $\{\alpha_0, \alpha_1, \ldots, \alpha_{r-1}\}$ is a basis for F_{2^m} over F_{2^l}, then each element $\alpha \in \mathrm{F}_{2^m}$ can be uniquely written in the form

$$\alpha = \sum_{i=0}^{r-1} a_i \alpha_i, \quad \text{where } a_i \in \mathrm{F}_{2^l}.$$

   Field multiplication in F_{2^m} now involves performing several operations in the field F_{2^l}. Since l is small, arithmetic in F_{2^l} can be sped up significantly, for example, by precomputing "log" and "antilog" tables. The drawback of this method is the space required for the tables. See Harper, Menezes and Vanstone [23] for an implementation report when $l = 8$, and De Win *et al* [83] and Guajardo and Paar [22] for a report when $l = 16$.

5.2. Selecting an Appropriate Elliptic Curve

By an "appropriate" elliptic curve, we mean an elliptic curve E defined over a finite field F_q satisfying the following conditions:

(i) To resist the Pollard ρ-attack mentioned in §4, $\#E(\mathrm{F}_q)$ should be divisible by a sufficiently large prime n (for example, $n > 2^{160}$).

(ii) To resist the Semaev–Smart–Satoh–Araki attack mentioned in §4, $\#E(\mathrm{F}_q)$ should not be equal to q.

(iii) To resist the MOV reduction attack mentioned in §4, n should not divide $q^k - 1$ for all $1 \le k \le C$, where C is large enough so that it is computationally infeasible to find discrete logarithms in $\mathrm{F}_{q^C}^*$. ($C = 20$ suffices in practice.)

We shall say that a positive integer u is *B-almost prime* if u is divisible by a prime factor $\ge u/B$.

Below we give an overview of four techniques for selecting an appropriate elliptic curve.

Using Hasse's Theorem. This technique can be used for picking curves over F_{2^m} where m is divisible by a small integer $l \ge 1$.

If E is an elliptic curve defined over F_q, then E can be viewed as an elliptic curve over any extension F_{q^k} of F_q; $E(\mathrm{F}_q)$ is a subgroup of $E(\mathrm{F}_{q^k})$. Hasse's theorem enables one to compute $\#E(\mathrm{F}_{q^k})$ from $\#E(\mathrm{F}_q)$ as follows. Let $t = q + 1 - \#E(\mathrm{F}_q)$. Then $\#E(\mathrm{F}_{q^k}) = q^k + 1 - \alpha^k - \beta^k$, where α and β are complex numbers determined from the factorization of $1 - tT + qT^2 = (1 - \alpha T)(1 - \beta T)$.

To select an appropriate curve over F_{2^m}, we first pick an elliptic curve over a small field F_{2^l}, where l divides m, compute $\#E(F_{2^l})$ exhaustively, and then use Hasse's theorem to determine $\#E(F_{2^m})$. If conditions (i), (ii) and (iii) above (with $q = 2^m$) are not satisfied, then another curve is selected and the process is repeated. Since the number of elliptic curves over F_{2^l} is relatively small, for a fixed m it may not be possible to construct an appropriate curve using this method.

Koblitz [34] observed that if one uses exponents k of small Hamming weight when computing kP in $E(F_{2^m})$, then one gets doubling of points "almost 3/4 for free" for some anomalous curves E defined over F_{2^l} (where m is a multiple of l). He provides a list of anomalous curves defined over F_2 (respectively F_4, F_8 and F_{16}) and extension degrees m such that $\#E(F_{2^m})$ (respectively, $\#E(F_{4^m})$, $\#E(F_{8^m})$ and $\#E(F_{16^m})$) has a prime factor of at least 30 decimal digits, and there exists an optimal normal basis in F_{q^m}. For these curves, if one uses exponents k of low Hamming weight, then any string of ≤ 4 zeros in k (respectively, exactly 2, 3, 4 zeros) can be handled with a single addition of points. In [78] Solinas, building on earlier work of Meier and Staffelbach [47], shows how to compute kP very efficiently in $E(F_{2^m})$ for arbitrary k, where E is an anomalous curve defined over F_2. (Note: the Semaev–Smart–Satoh–Araki algorithm mentioned before does not apply to these anomalous curves, which are used not over a prime field, but rather over a large degree extension of their field of definition.)

The Global Method. Another possibility is to choose an elliptic curve defined over a number field and then reduce it modulo a prime ideal such that the resulting curve over a finite field satisfies conditions (i), (ii) and (iii). For instance, we could start with the equation (1) with $a, b \in \mathbb{Q}$ and then consider the same equation modulo p for large primes p, where we want the number N_p of points on the curve over F_p to be a prime or a prime times a small factor. Here N_p is always divisible by $\#E_{\text{tors}}$, the number of points of finite order on the original elliptic curve over Q. But the ratio $N_p/\#E_{\text{tors}}$ will often be prime. It should be noted that $\#E_{\text{tors}} \leq 16$ by a deep theorem of B. Mazur [45], and $\#E_{\text{tors}} = 1$ for most "random" curves. For more discussion of primality of N_p, see [30].

Example: Consider the curve $y^2 = x^3 - m^2x$, where m is an integer parameter. (This is the family of curves that arises from the famous Congruent Number Problem, first studied by the ancient Greeks; see [35].) Now consider this curve modulo a prime p not dividing m, where $p \equiv 1 \pmod 4$. (Note: if $p \equiv 3 \pmod 4$, then the curve is supersingular.) It was Gauss who found a simple formula for N_p. First one has to write p as a sum of two squares: $p = a^2 + b^2$ (this is a very easy computational task), where without loss of generality we suppose that a is odd. We choose the sign of a by requiring that $a + b \equiv \left(\frac{m}{p}\right) \pmod 4$. Then $N_p = p + 1 - a$. Since our original elliptic curve over Q has exactly four points of finite order (namely $(0, 0)$, $(\pm m, 0)$, ∞), it follows that 4 divides N_p. But often $N_p/4$ is prime. □

The Complex Multiplication Method. The method of complex multiplication (CM) allows the choice of an elliptic curve order *before* the curve is explicitly constructed. Thus, orders can be generated and tested to satisfy conditions (i), (ii) and (iii); a curve is constructed only

when these conditions are met. The CM method is efficient provided that the finite field size q and the order $\#E(\mathrm{F}_q) = q+1-t$ are chosen so that the CM-field $\mathrm{Q}(\sqrt{t^2-4q})$ has small class number. For elliptic curves over F_p, the CM method is also called the *Atkin-Morain method* (see [53]); over F_{2^m}, it is called the *Lay-Zimmer method* (see [40]). The CM method is fast in practice. Lay and Zimmer [40] report timings of about 3 minutes on a SPARC 2 (excluding the time for precomputation) for the construction of an elliptic curve over $\mathrm{F}_{2^{191}}$ whose order is twice a prime.

Choosing a Curve at Random. Another approach to selecting an appropriate elliptic curve E over F_q is to select random parameters $a, b \in \mathrm{F}_q$ (subject to the constraint that $4a^3+27b^2 \neq 0$ if q is odd, and $b \neq 0$ if q is a power of 2). One then computes $u = \#E(\mathrm{F}_q)$ and factors u. This process is repeated until conditions (i), (ii) and (iii) are satisfied.

In the case of elliptic curves over F_p, the following theorem shows that, if the coefficients a and b are selected uniformly at random, then the orders of the resulting elliptic curves are roughly uniformly distributed. Similar results for the case of elliptic curves over F_{2^m} can be deduced from the work of Waterhouse [81] and Schoof [70].

THEOREM (LENSTRA [41]) *There exist effectively computable positive constants c_1 and c_2 such that for each prime $p \geq 5$ and for any subset S of integers in the interval $[p+1-\sqrt{p},\ p+1+\sqrt{p}]$, the probability r_S that a random pair $(a,b) \in F_p \times F_p$ determines an elliptic curve E: $y^2 = x^3+ax+b$ with $\#E(F_p) \in S$ is bounded as follows:*

$$\frac{\#S-2}{2\lfloor\sqrt{p}\rfloor+1} \cdot c_1(\log p)^{-1} \ \leq\ r_S \ \leq\ \frac{\#S}{2\lfloor\sqrt{p}\rfloor+1} \cdot c_2(\log p)(\log\log p)^2.$$

For fixed B and sufficiently large q, it is thus reasonable to assume that the probability of B-almost primality of the order of a randomly chosen elliptic curve over F_q is roughly equal to the probability of B-almost primality of a random integer of the same order of magnitude as q. If q is a power of 2, then one considers random *even* integers of the same order of magnitude as q. For fixed B and $q = 2^m$, the latter probability is asymptotic to $\sum_{j=1}^{B/2} \frac{1}{j\log(q/2j)} \approx \frac{1}{m}\log_2(B/2)$. For example, if $q = 2^{175}$ and we want an elliptic curve whose order is divisible by $n > 2^{160}$ (so $B = 2^{15}$), we expect to try about 13 curves before finding one whose order is B-almost prime.

In 1985 Schoof [69] presented a polynomial-time algorithm for computing the number of F_q-points on an elliptic curve defined over F_q in the case when q is odd; the algorithm was later extended to the case of q a power of 2 by Koblitz [32]. Schoof's algorithm has a worst-case running time of $O((\log q)^8)$ bit operations, and is rather inefficient in practice for the values of q of practical interest (i.e., $q > 2^{160}$). In the last few years a lot of work has been done on improving and refining Schoof's algorithm. Lercier and Morain [44] implemented Schoof's algorithm incorporating ideas of Atkin, Elkies and Couveignes. They reported timings of 4 and 3 minutes on a DecAlpha 3000/500 for computing the orders of elliptic curves over $\mathrm{F}_{2^{155}}$ and over a 155-bit prime field, respectively. A new record for elliptic curve point counting over prime fields was established in 1995 by Lercier and Morain [44], who computed the order of a curve over a 499-decimal digit (1658-bit) prime field; the computation took the equivalent of roughly 4200 hours on a DEC 3000-M300X. In the

case of characteristic two finite fields, the current record was established in June 1998 by A. Joux and R. Lercier, who computed the order of a curve over $F_{2^{1663}}$; the computation took the equivalent of roughly 330 days on a DEC Alpha. They used the Schoof–Elkies–Atkin algorithm and incorporated newer ideas of Lercier [42]. Cryptographically suitable elliptic curves over fields as large as $F_{2^{196}}$ can be randomly generated in a few hours on a workstation [43].

5.3. Standards Activities

The two primary objectives of industry standards are to promote interoperability and to facilitate widespread use of well-accepted techniques. Standards for elliptic curve systems are currently being drafted by various accredited standards bodies around the world; some of this work is summarized below.

1. The Elliptic Curve Digital Signature Algorithm (ECDSA) was adopted in January 1999 as an official American National Standards Institute (ANSI) standard. The ANSI X9 (Financial Services) working group is also drafting a standard for elliptic curve key agreement and transport protocols.

2. Elliptic curves are in the draft IEEE P1363 standard (Standard Specifications for Public-Key Cryptography), which includes encryption, signature, and key agreement mechanisms. Elliptic curves over F_p and over F_{2^m} are both supported. For the characteristic two finite fields, polynomial bases and normal bases of F_{2^m} over an arbitrary subfield F_{2^l} are supported. P1363 also includes discrete log systems in subgroups of the multiplicative group of the integers modulo a prime, as well as RSA encryption and signatures. The latest drafts are available from the web site `http://stdsbbs.ieee.org/`.

3. The OAKLEY Key Determination Protocol of the Internet Engineering Task Force (IETF) describes a key agreement protocol that is a variant of Diffie–Hellman. It allows for a variety of groups to be used, including elliptic curves over F_p and F_{2^m}. The document makes specific mention of elliptic curve groups over the fields $F_{2^{155}}$ and $F_{2^{210}}$. A draft is available from the web site `http://www.ietf.cnri.reston.va.us/`.

4. ECDSA is specified in the draft document ISO/IEC 14888: Digital signature with appendix – Part 3: Certificate-based mechanisms.

5. The ISO/IEC 15946 draft standard specifies various cryptographic techinques based on elliptic curves including signature schemes, public-key encyrption schemes, and key establishment protocols.

6. The ATM Forum Technical Committee's Phase I ATM Security Specification draft document aims to provide security mechanisms for Asynchronous Transfer Mode (ATM) networks. Security services provided include confidentiality, authentication, data integrity, and access control. A variety of systems are supported, including RSA, DSA, and elliptic curve systems.

As these drafts become officially adopted by the appropriate standards bodies, one can expect elliptic curve systems to be widely used by providers of information security.

Notes

1. This is a security condition: if $r = 0$, then the signing equation $s = k^{-1}\{h(m) + dr\} \bmod n$ does not involve the private key d.
2. If $s = 0$ then $s^{-1} \bmod n$ does not exist; this is required in step 3 of signature verification. Note that if k is chosen at random, then the probability that either $r = 0$ or $s = 0$ is negligibly small.
3. More precisely, let m be a prime factor of n that does not divide $q - 1$. Then the MOV algorithm for discrete logs in the subgroup of $E(\mathrm{F}_q)$ of order m can be carried out in $\mathrm{F}_{q^k}^*$ if and only if $m|q^k - 1$.
4. It must be emphasized that such a comparison is very rough, as it does not take into account the various enhancements that are possible for each system.
5. Here *optimality* refers to the minimum possible number of interconnections between the components of the multiplicands.

References

1. L. Adleman, J. DeMarrais and M. Huang, A subexponential algorithm for discrete logarithms over the rational subgroup of the jacobians of large genus hyperelliptic curves over finite fields, *Algorithmic Number Theory*, Lecture Notes in Computer Science, Springer-Verlag, 877 (1994) pp. 28–40.
2. G. Agnew, R. Mullin, I. Onyszchuk and S. Vanstone, An implementation for a fast public-key cryptosystem, *Journal of Cryptology*, Vol. 3 (1991) pp. 63–79.
3. G. Agnew, R. Mullin and S. Vanstone, An implementation of elliptic curve cryptosystems over $F_{2^{155}}$, *IEEE Journal on Selected Areas in Communications*, Vol. 11 (1993) pp. 804–813.
4. D. Bailey C. Paar, Optimal extension fields for fast arithmetic in public-key algorithms, Advances in Cryptology—CRYPTO '98, Lecture Notes in Computer Science, Springer-Verlag, 1462 (1998) pp. 472–485.
5. R. Balasubramanian and N. Koblitz, The improbability that an elliptic curve has subexponential discrete log problem under the Menezes–Okamoto–Vanstone algorithm, *Journal of Cryptology*, Vol. 11 (1998) pp. 141–145.
6. M. Blaze, W. Diffie, R. Rivest, B. Schneier, T. Shimomura, E. Thompson, and M. Wiener, Minimal key lengths for symmetric ciphers to provide adequate commercial security, January 1996, available from `http://theory.lcs.mit.edu/~rivest/publications.html`.
7. D. Bleichenbacher, On the security of the KMOV public key cryptosystem, Advances in Cryptology—CRYPTO '97, Lecture Notes in Computer Science, Springer-Verlag, 1294 (1997) pp. 235–248.
8. D. Boneh and R. Lipton, Algorithms for black-box fields and their applications to cryptography, Advances in Cryptology—CRYPTO '96, Lecture Notes in Computer Science, Springer-Verlag, 1109 (1996) pp. 283–297.
9. J. Buchmann and H. Williams, A key-exchange system based on imaginary quadratic fields, *Journal of Cryptology*, Vol. 1 (1988) pp. 107–118.
10. L. Charlap and D. Robbins, *An Elementary Introduction to Elliptic Curves*, CRD Expository Report No. 31, Institute for Defense Analysis, Princeton (December 1988).
11. L. Charlap and D. Robbins, *An Elementary Introduction to Elliptic Curves II*, CRD Expository Report No. 34, Institute for Defense Analysis, Princeton (December 1988).
12. D. Coppersmith, Fast evaluation of logarithms in fields of characteristic two, *IEEE Transactions on Information Theory*, Vol. 30 (1984) pp. 587–594.
13. R. Crandall, Method and apparatus for public key exchange in a cryptographic system, U.S. patent number 5,159,632 (October 1992).
14. W. Diffie and M. Hellman, New directions in cryptography, *IEEE Transactions on Information Theory*, Vol. 22 (1976) pp. 644–654.

15. Y. Driencourt and J. Michon, Elliptic codes over a field of characteristic 2, *Journal of Pure and Applied Algebra*, Vol. 45 (1987) pp. 15–39.
16. T. ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithms, *IEEE Transactions on Information Theory*, Vol. 31 (1985) pp. 469–472.
17. G. Frey and H. Rück, A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves, *Mathematics of Computation*, Vol. 62 (1994) pp. 865–874.
18. R. Gallant, R. Lambert and S. Vanstone, Improving the parallelized Pollard lambda search on binary anomalous curves, to appear in *Mathematics of Computation*.
19. G. van der Geer, Codes and elliptic curves, *Effective Methods in Algebraic Geometry*, Birkhäuser (1991) pp. 159–168.
20. S. Goldwasser and J. Kilian, Almost all primes can be quickly certified, *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, (1986) pp. 316–329
21. D. Gordon, Discrete logarithms in $GF(p)$ using the number field sieve, *SIAM Journal on Discrete Mathematics*, Vol. 6 (1993) pp. 124–138.
22. J. Guajardo and C. Paar, Efficient algorithms for elliptic curve cryptosystems, Advances in Cryptology—CRYPTO '97, Lecture Notes in Computer Science, Springer-Verlag, 1294 (1997) pp. 342–356.
23. G. Harper, A. Menezes and S. Vanstone, Public-key cryptosystems with very small key lengths, Advances in Cryptology—EUROCRYPT '92, Lecture Notes in Computer Science, Springer-Verlag, 658 (1993) pp. 163–173.
24. P. Ivey, S. Walker, J. Stern and S. Davidson, An ultra-high speed public key encryption processor, *Proceedings of IEEE Custom Integrated Circuits Conference*, Boston (1992) 19.6.1–19.6.4.
25. M. Jacobson, N. Koblitz, J. Silverman, A. Stein and E. Teske, Analysis of the xedni calculus attack, to appear in *Designs, Codes and Cryptography*.
26. B. Kaliski, A pseudorandom bit generator based on elliptic logarithms, Advances in Cryptology—CRYPTO '86, Lecture Notes in Computer Science, Springer-Verlag, 293 (1987) pp. 84–103.
27. B. Kaliski, One-way permutations on elliptic curves, *Journal of Cryptology*, Vol. 3 (1991) pp. 187–199.
28. B. Kaliski, A chosen message attack on Demytko's elliptic curve cryptosystem, *Journal of Cryptology*, Vol. 10 (1997) pp. 71–72.
29. N. Koblitz, Elliptic curve cryptosystems, *Mathematics of Computation*, Vol. 48 (1987) pp. 203–209.
30. N. Koblitz, Primality of the number of points on an elliptic curve over a finite field, *Pacific Journal of Mathematics*, Vol. 131 (1988) pp. 157–165.
31. N. Koblitz, Hyperelliptic cryptosystems, *Journal of Cryptology*, Vol. 1 (1989) pp. 139–150.
32. N. Koblitz, Constructing elliptic curve cryptosystems in characteristic 2, Advances in Cryptology—CRYPTO '90, Lecture Notes in Computer Science, Springer-Verlag, 537 (1991) pp. 156–167.
33. N. Koblitz, Elliptic curve implementation of zero-knowledge blobs, *Journal of Cryptology*, Vol. 4 (1991) pp. 207–213.
34. N. Koblitz, CM-curves with good cryptographic properties, Advances in Cryptology—CRYPTO '91, Lecture Notes in Computer Science, Springer-Verlag, 576 (1992) pp. 279–287.
35. N. Koblitz, *Introduction to Elliptic Curves and Modular Forms*, 2nd edition, Springer-Verlag (1993).
36. N. Koblitz, *A Course in Number Theory and Cryptography*, 2nd edition, Springer-Verlag (1994).
37. N. Koblitz, *Algebraic Aspects of Cryptography*, Springer-Verlag (1998).
38. K. Koyama, U. Maurer, T. Okamoto and S. Vanstone, New public-key schemes based on elliptic curves over the ring Z_n, Advances in Cryptology—CRYPTO '91, Lecture Notes in Computer Science, Springer-Verlag, 576 (1993) pp. 252–266.
39. K. Kurosawa, K. Okada and S. Tsujii, Low exponent attack against elliptic curve RSA, Advances in Cryptology—ASIACRYPT '94, Lecture Notes in Computer Science, Springer-Verlag, 917 (1995) pp. 376–383.
40. G. Lay and H. Zimmer, Constructing elliptic curves with given group order over large finite fields, Algorithmic Number Theory, Lecture Notes in Computer Science, Springer-Verlag, 877 (1994) pp. 250–263.
41. H. W. Lenstra, Factoring integers with elliptic curves, *Annals of Mathematics*, Vol. 126 (1987) pp. 649–673.
42. R. Lercier, Computing isogenies in F_{2^n}, Algorithmic Number Theory, Proceedings Second Intern. Symp., ANTS-II, (Henri Cohen, ed.), Lecture Notes in Computer Science, Springer-Verlag, 1122 (1996) pp. 197–212.
43. R. Lercier, Finding good random elliptic curves for cryptosystems defined F_{2^n}, Advances in Cryptology—EUROCRYPT '97, Lecture Notes in Computer Science, Springer-Verlag, 1233 (1997) pp. 379–392.

44. R. Lercier and F. Morain, Counting the number of points on elliptic curves over finite fields: strategies and performances, Advances in Cryptology—EUROCRYPT '95, Lecture Notes in Computer Science, Springer-Verlag, 921 (1995) pp. 79–94.
45. B. Mazur, Modular curves and the Eisenstein ideal, *Inst. Hautes Études Sci. Publ. Math.*, Vol. 47 (1977) pp. 33–186.
46. K. McCurley, A key distribution system equivalent to factoring, *Journal of Cryptology*, Vol. 1 (1988) pp. 95–105.
47. W. Meier and O. Staffelbach, Efficient multiplication on certain nonsupersingular elliptic curves, Advances in Cryptology—CRYPTO '92, Lecture Notes in Computer Science, Springer-Verlag, 740 (1993) pp. 333–344.
48. A. Menezes, it Elliptic Curve Public Key Cryptosystems, Kluwer Academic Publishers, Boston (1993).
49. A. Menezes, T. Okamoto and S. Vanstone, Reducing elliptic curve logarithms to logarithms in a finite field, *IEEE Transactions on Information Theory*, Vol. 39 (1993) pp. 1639–1646.
50. A. Menezes and S. Vanstone, Elliptic curve cryptosystems and their implementation, *Journal of Cryptology*, Vol. 6 (1993) pp. 209–224.
51. J. F. Mestre, Formules explicites et minoration de conducteurs de variétés algébriques, *Compositio Math.*, Vol. 58 (1986) pp. 209–232.
52. V. Miller, Uses of elliptic curves in cryptography, Advances in Cryptology—CRYPTO '85, Lecture Notes in Computer Science, Springer-Verlag, 218 (1986) pp. 417–426.
53. F. Morain, Building cyclic elliptic curves modulo large primes, Advances in Cryptology—EUROCRYPT '91, Lecture Notes in Computer Science, Springer-Verlag, 547 (1991) pp. 328–336.
54. V. Müller, S. Vanstone and R. Zuccherato, Discrete logarithm based cryptosystems in quadratic function fields of characteristic 2, *Designs, Codes and Cryptography*, Vol. 14 (1998) pp. 159–178.
55. R. Mullin, I. Onyszchuk, S. Vanstone and R. Wilson, Optimal normal bases in $GF(p^n)$, *Discrete Applied Mathematics*, Vol. 22 (1988/89) pp. 149–161.
56. National Institute for Standards and Technology, Digital signature standard, FIPS Publication 186 (1993).
57. National Institute for Standards and Technology, Secure hash standard, FIPS Publication 180-1 (1995).
58. A. Odlyzko, The future of integer factorization, *CryptoBytes–The Technical Newsletter of RSA Laboratories*, Vol. 1, No. 2 (Summer 1995) pp. 5–12.
59. P. van Oorschot and M. Wiener, Parallel collision search with application to hash functions and discrete logarithms, *Proceedings of the 2nd ACM Conference on Computer and Communications Security*, Fairfax, Virginia (2–4 November 1994) pp. 210–218.
60. P. van Oorschot and M. Wiener, Parallel collision search with cryptanalytic applications, *Journal of Cryptology*, Vol. 12 (1999) pp. 1–28.
61. R. Pinch, Extending the Wiener attack to RSA-type cryptosystems, *Electronics Letters*, Vol. 31 (1995) pp. 1736–1738.
62. S. Pohlig and M. Hellman, An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance, *IEEE Transactions on Information Theory*, Vol. 24 (1978) pp. 106–110.
63. J. Pollard, Monte Carlo methods for index computation mod p, *Mathematics of Computation*, Vol. 32 (1978) pp. 918–924.
64. T. Satoh and K. Araki, Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves, *Commentarii Mathematici Universitatis Sancti Pauli*, Vol. 47 (1998) pp. 81–92.
65. R. Scheidler, J. Buchmann and H. Williams, A key-exchange protocol using real quadratic fields, *Journal of Cryptology*, Vol. 7 (1994) pp. 171–199.
66. R. Scheidler, A. Stein and H. Williams, Key-exchange in real quadratic congruence function fields, *Designs, Codes and Cryptography*, Vol. 7 (1996) pp. 153–174.
67. O. Schirokauer, Discrete logarithms and local units, *Philosophical Transactions of the Royal Society of London A*, Vol. 345 (1993) pp. 409–423.
68. C. Schnorr, Efficient signature generation by smart cards, *Journal of Cryptology*, Vol. 4 (1991) pp. 161–174.
69. R. Schoof, Elliptic curves over finite fields and the computation of square roots mod p, *Mathematics of Computation*, Vol. 44 (1985) pp. 483–494.
70. R. Schoof, Nonsingular plane cubic curves, *Journal of Combinatorial Theory*, Series A, Vol. 46 (1987) pp. 183–211.
71. R. Schroeppel, H. Orman, S. O'Malley and O. Spatscheck, Fast key exchange with elliptic curve systems, Advances in Cryptology—CRYPTO '95, Lecture Notes in Computer Science, Springer-Verlag, 963 (1995) pp. 43–56.

72. I. Semaev, Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p, *Mathematics of Computation*, Vol. 67 (1998) pp. 353–356.
73. J. Silverman, *The Arithmetic of Elliptic Curves*, Springer-Verlag, New York (1986).
74. J. Silverman, *Advanced Topics in the Arithmetic of Elliptic Curves*, Springer-Verlag, New York (1994).
75. J. Silverman, The xedni calculus and the elliptic curve discrete logarithm problem, to appear in it Designs, Codes and Cryptography.
76. J. Silverman and J. Suzuki, Elliptic curve discrete logarithms and the index calculus, to appear in Advances in Cryptology—ASIACRYPT '98, Lecture Notes in Computer Science, Springer-Verlag (1998).
77. N. Smart, The discrete logarithm problem on elliptic curves of trace one, to appear in *Journal of Cryptology*.
78. J. Solinas, An improved algorithm for arithmetic on a family of elliptic curves, Advances in Cryptology—CRYPTO '97, Lecture Notes in Computer Science, Springer-Verlag, 1294 (1997) pp. 357–371.
79. A. Stein, Equivalences between elliptic curves and real quadratic congruence function fields, *Journal de Théorie des Nombres de Bordeaux*, Vol. 9 (1997) pp. 75–95.
80. A. Stein, V. Müller and C. Thiel, Computing discrete logarithms in real quadratic congruence function fields of large genus, *Mathematics of Computation*, Vol. 68 (1999) pp. 807–822.
81. W. Waterhouse, Abelian varieties over finite fields, *Ann. Sci. École Norm. Sup.*, 4^{e} série, Vol. 2 (1969) pp. 521–560.
82. M. Wiener and R. Zuccherato, Fast attacks on elliptic curve cryptosystems," to appear in Fifth Annual Workshop on Selected Areas in Cryptography – SAC '98, Lecture Notes in Computer Science, Springer-Verlag (1999).
83. E. De Win, A. Bosselaers, S. Vandenberghe, P. De Gersem and J. Vandewalle, A fast software implementation for arithmetic operations in $GF(2^n)$, Advances in Cryptology—ASIACRYPT '96, Lecture Notes in Computer Science, Springer-Verlag, 1163 (1996) pp. 65–76.
84. E. De Win, S. Mister, B. Preneel and M. Wiener, On the performance of signature schemes based on elliptic curves, Algorithmic Number Theory, Proceedings Third Intern. Symp., ANTS-III (J. P. Buhler, ed.), Lecture Notes in Computer Science, Springer-Verlag, 1423 (1998) pp. 252–266.
85. R. Zuccherato, The equivalence between elliptic curve and quadratic function field discrete logarithms in characteristic 2, Algorithmic Number Theory, Proceedings Third Intern. Symp., ANTS-III (J. P. Buhler, ed.), Lecture Notes in Computer Science, Springer-Verlag, 1423 (1998) pp. 621–638.