Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field

Alfred Menezes & Scott Vanstone
Dept. of Combinatorics and Optimization, University of Waterloo
Waterloo, Ontario, Canada, N2L 3G1.

Tatsuaki Okamoto
NTT Laboratories
Take, Yokosuka-Shi, 238-03 Japan.

Abstract

Previously, no general-purpose algorithm was known for the elliptic curve logarithm problem that ran in better than exponential time. In this paper we demonstrate the reduction of the elliptic curve logarithm problem to the logarithm problem in the multiplicative group of an extension of the underlying finite field. For the class of supersingular elliptic curves, the reduction takes probabilistic polynomial time, thus providing a probabilistic subexponential time algorithm for the former problem. The implications of our results to public key cryptography are discussed.

1 Introduction

The discrete logarithm problem for a general group G can be stated as follows: Given $\alpha \in G$ and $\beta \in G$, find an integer x such that $\beta = \alpha^x$, provided that such an integer exists. The integer x is called the discrete logarithm of β to the base α. In this paper, we shall consider the case where G is an elliptic curve group E, and where α is a point $P \in E$.

In [8] and [12], Koblitz and Miller described how the group of points on an elliptic curve over a finite field can be used to construct public key cryptosystems. The security of these cryptosystems is based upon the presumed intractability of the problem of computing logarithms in the elliptic curve group. The best algorithms that are known for solving this problem, are the exponential square root attacks (for example, see [13]) that apply to any finite group and have a running time proportional to the square root of the largest prime factor dividing the order of the group. In [12], Miller argues that the index-calculus methods, which produced dramatic results in the computation of discrete logarithms in (the multiplicative group of) a finite field (see [3], [13]), do not extend to elliptic curve groups. Consequently, if the elliptic curve is chosen so that its order is divisible by a large prime then, even the best attacks take exponential time.

The method we propose in this paper, reduces the elliptic curve logarithm problem in a curve E over a finite field F_q to the discrete logarithm problem in a suitable extension field F_{q^k} of F_q. This is achieved by establishing an isomorphism between $<P>$, the subgroup of E generated by P, and the subgroup of n^{th} roots of unity in F_{q^k}, where n denotes the order of P. The isomorphism is given by the Weil pairing.

Since the index-calculus methods for computing logarithms in a finite field have running times that are subexponential, the reduction is useful for the purpose of computing elliptic curve logarithms provided that k is small. This is indeed the case for special classes of elliptic curves, including the

curves recommended for implementation in [12], [8], [6], [2], and [9]. More precisely, we prove the following two results about the reduction.

Theorem 11 *If $E(F_q)$ is a supersingular curve, then the reduction of the elliptic curve logarithm problem in $E(F_q)$ to the discrete logarithm problem in F_{q^k} is a probabilistic polynomial time (in ln q) reduction.*

Corollary 12 *Let P be an element of order n in a supersingular elliptic curve $E(F_q)$, and let $R = lP$ be a point in $E(F_q)$. If q is a prime, or if q is a prime power $q = p^m$, where p is small, then the new algorithm can determine l in probabilistic subexponential time.*

The remainder of the paper is organized as follows. In Section 2, we list some of the properties of elliptic curves that we will use. In Section 3, we describe the reduction, and in Section 4, we mention some special curves for which the reduction is especially useful. Finally, in Section 5, we discuss some of the implications of our results for cryptography.

2 Background on Elliptic Curves

In this Section, we review some of the theory of elliptic curves over finite fields which we will use. For further information, we refer the reader to the book by Silverman [19].

Let $E(F_q)$ be an elliptic curve over F_q, the finite field on q elements. Let $q = p^m$, where p is the characteristic of F_q. If p is greater than 3, then $E(F_q)$ is the set of all solutions in $F_q \times F_q$ to an affine equation

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

with $a, b \in F_q$, $4a^3 + 27b^2 \neq 0$, together with an additive identity element O, called the point at infinity. If $p = 2$, then an affine equation for $E(F_q)$ is

$$y^2 + a_3 y = x^3 + a_4 x + a_6, \tag{2}$$

with $a_3, a_4, a_6 \in F_q$, $a_3 \neq 0$, if the curve has j-invariant equal to 0, and

$$y^2 + xy = x^3 + a_2 x^2 + a_6, \tag{3}$$

with $a_2, a_6 \in F_q$, $a_6 \neq 0$, if the curve has j-invariant not equal to 0. There is a natural addition defined on the points of $E(F_q)$ that is given by the "tangent and chord method", and involves a few arithmetic operations in F_q. Under this addition, the points of $E(F_q)$ form an abelian group of rank 1 or 2. By Hasse's Theorem, the order of the group is $q + 1 - t$, where $|t| \leq 2\sqrt{q}$. The type of the group is (n_1, n_2), where $n_2 | n_1$, and furthermore $n_2 | q - 1$. The next result determines whether or not an elliptic curve of a certain order exists.

Lemma 1 ([18], 4.2) *There exists an elliptic curve of order $q + 1 - t$ over F_q if and only if one of the following conditions holds:*
(i) $t \not\equiv 0 \pmod p$ and $t^2 \leq 4q$.
(ii) m is odd and one of the following holds:
(1) $t = 0$.
(2) $t^2 = 2q$ and $p = 2$.
(3) $t^2 = 3q$ and $p = 3$.
(ii) m is even and one of the following holds:
(1) $t^2 = 4q$.
(2) $t^2 = q$ and $p \not\equiv 1 \pmod 3$.
(3) $t = 0$ and $p \not\equiv 1 \pmod 4$. ■

Let $\#E(F_q) = q + 1 - t$ denote the order of a curve. $E(F_q)$ is said to be supersingular if p divides t. From the preceding result, we can deduce that $E(F_q)$ is supersingular if and only if $t^2 = 0$, q, $2q$, $3q$, or $4q$. The following result gives the group structure of the supersingular curves. $\mathbf{Z}_n$ denotes the cyclic group on n elements.

Lemma 2 ([18], 4.8) *Let $\#E(F_q) = q + 1 - t$.*
(i) If $t^2 = q$, $2q$, or $3q$, then $E(F_q)$ is cyclic.
(ii) If $t^2 = 4q$, then either $E(F_q) \cong \mathbf{Z}_{\sqrt{q}-1} \oplus \mathbf{Z}_{\sqrt{q}-1}$ or $E(F_q) \cong \mathbf{Z}_{\sqrt{q}+1} \oplus \mathbf{Z}_{\sqrt{q}+1}$, depending on whether $t = 2\sqrt{q}$ or $t = -2\sqrt{q}$ respectively.
(iii) If $t = 0$ and $q \not\equiv 3 \pmod 4$, then $E(F_q)$ is cyclic. If $t = 0$ and $q \equiv 3 \pmod 4$, then either $E(F_q)$ is cyclic, or $E(F_q) \cong \mathbf{Z}_{(q+1)/2} \oplus \mathbf{Z}_2$. ■

The curve E can also be viewed as an elliptic curve over any extension field K of F_q; $E(F_q)$ is a subgroup of $E(K)$. The Weil Conjecture enables one to compute $\#E(F_{q^k})$ from $\#E(F_q)$ as follows. Let $t = q + 1 - \#E(F_q)$. Then $\#E(F_{q^k}) = q^k + 1 - \alpha^k - \beta^k$, where α, β are complex numbers determined from the factorization of $1 - tT + qT^2 = (1 - \alpha T)(1 - \beta T)$.

An n-torsion point P is a point satisfying $nP = O$. Let $E(F_q)[n]$ denote the subgroup of n-torsion points in $E(F_q)$, where $n \neq 0$. We will write $E[n]$ for $E(\overline{F_q})[n]$, where $\overline{F_q}$ denotes the algebraic closure of F_q. If n and q are relatively prime, then $E[n] \cong \mathbb{Z}_n \oplus \mathbb{Z}_n$. If $n = p^e$, then either $E[p^e] \cong \{O\}$ if E is supersingular, or else $E[p^e] \cong \mathbb{Z}_{p^e}$ if E is non-supersingular.

The following result provides necessary and sufficient conditions for $E(F_q)$ to contain all of the n-torsion points in $E(\overline{F_q})$. For definition of the terms in condition (iii), see [18].

Lemma 3 ([18], 3.7) *If* $\gcd(n, q) = 1$, *then* $E[n] \subset E(F_q)$ *if and only if the following three conditions hold:*
(i) $n^2 \mid \#E(F_q)$
(ii) $n \mid q - 1$
(iii) Either $\phi \in \mathbb{Z}$ *or* $\vartheta\left(\frac{t^2-4q}{n^2}\right) \subset End_{F_q}(E)$. ■

Let n be a positive integer relatively prime to q. The *Weil pairing* is a function

$$e_n : E[n] \times E[n] \longrightarrow \overline{F_q}.$$

For a definition of the Weil pairing, see the Appendix. We list some useful properties of the Weil pairing.

(i) *Identity:* For all $P \in E[n]$, $e_n(P, P) = 1$.

(ii) *Alternation:* For all $P_1, P_2 \in E[n]$, $e_n(P_1, P_2) = e_n(P_2, P_1)^{-1}$.

(iii) *Bilinearity:* For all $P_1, P_2, P_3 \in E[n]$, $e_n(P_1 + P_2, P_3) = e_n(P_1, P_3)\, e_n(P_2, P_3)$, and $e_n(P_1, P_2 + P_3) = e_n(P_1, P_2)e_n(P_1, P_3)$.

(iv) *Non-degeneracy:* If $P_1 \in E[n]$, and if $e_n(P_1, P_2) = 1$ for all $P_2 \in E[n]$, then $P_1 = O$.

(v) If $E[n] \subseteq E(F_{q^k})$, then $e_n(P_1, P_2) \in F_{q^k}$, for all $P_1, P_2 \in E[n]$.

Miller has developed an efficient probabilistic polynomial-time algorithm for computing the Weil pairing [11]. By a probabilistic polynomial algorithm, we mean a randomized algorithm whose expected running time is bounded by a polynomial in the size of the input. By a probabilistic subexponential algorithm with input x, we mean a randomized algorithm with expected running time $L[\alpha, x]$, where $0 < \alpha < 1$ and

$$L[\alpha, x] = e^{(c+o(1))\,(ln\, x)^{\alpha}(ln\, ln\, x)^{1-\alpha}}.$$

For a brief description of Miller's algorithm, see the Appendix. An implementation of the algorithm in MACSYMA, is given in [7].

The following result from [7] provides a method for partitioning the elements of an elliptic curve $E(F_q)$ into the cosets of $<P>$, the subgroup of $E(F_q)$ generated by a point P of maximum order.

Lemma 4 *Let* $E(F_q)$ *be an elliptic curve with group structure* (n_1, n_2), *and let* P *be an element of maximum order* n_1. *Then for all* $P_1, P_2 \in E(F_q)$, P_1 *and* P_2 *are in the same coset of* $<P>$ *if and only if* $e_{n_1}(P, P_1) = e_{n_1}(P, P_2)$. ■

The next result is similar to, and has a similar proof, as Lemma 4. For completeness, we include it here.

Lemma 5 *Let* $E(F_q)$ *be an elliptic curve such that* $E[n] \subseteq E(F_q)$, *and where* n *is a positive integer coprime to* q. *Let* $P \in E[n]$ *be a point of order* n. *Then for all* $P_1, P_2 \in E[n]$, P_1 *and* P_2 *are in the same coset of* $<P>$ *within* $E[n]$ *if and only if* $e_n(P, P_1) = e_n(P, P_2)$.

Proof. If $P_1 = P_2 + kP$, then clearly

$$\begin{aligned} e_n(P, P_1) &= e_n(P, P_2)e_n(P, P)^k \\ &= e_n(P, P_2). \end{aligned}$$

Conversely, suppose that P_1 and P_2 are in different cosets of $<P>$ within $E[n]$. Then we can write $P_1 - P_2 = a_1P + a_2Q$, where (P, Q) is a generating pair for $E[n] \cong \mathbb{Z}_n \oplus \mathbb{Z}_n$, and where $a_2Q \neq O$. If $b_1P + b_2Q$ is any point in $E[n]$, then

$$\begin{aligned} e_n(a_2Q, b_1P + b_2Q) &= e_n(a_2Q, P)^{b_1} e_n(Q, Q)^{a_2b_2} \\ &= e_n(P, a_2Q)^{-b_1}. \end{aligned}$$

If $e_n(P, a_2Q) = 1$ then by the non-degeneracy property of e_n, we have that $a_2Q = O$, a contradiction. Thus $e_n(P, a_2Q) \neq 1$. Finally,

$$\begin{aligned} e_n(P, P_1) &= e_n(P, P_2)e_n(P, P)^{a_1} e_n(P, a_2Q) \\ &\neq e_n(P, P_2). \end{aligned}$$ ■

For the algorithms that follow, it is essential that we are able to pick points P uniformly and randomly on an elliptic curve $E(F_q)$ in probabilistic polynomial time. This can be accomplished

as follows. We first randomly choose an element $x_1 \in F_q$. If x_1 is the x-coordinate of some point in $E(F_q)$, then we can find y_1 such that $(x_1, y_1) \in E(F_q)$ by solving a root finding problem in F_q. There are various techniques for finding the roots of a polynomial over F_q in probabilistic polynomial time; for example, see [1] if q is a prime, and [15] if q is a prime or a prime power. We then set $P = (x_1, y_1)$ or $(x_1, -y_1)$ if the curve has equation (1) (respectively, $P = (x_1, y_1)$ or $(x_1, y_1 + a_3)$, and $P = (x_1, y_1)$ or $(x_1, y_1 + x_1)$ if the curve has equation (2) or (3)). From Hasse's theorem, the probability that x_1 is the x-coordinate of some point in $E(F_q)$ is at least $1/2 - 1/\sqrt{q}$. Note that with the method just described the probability of picking a point of order 2 is twice the probability of picking any other point, however there are at most three points of order 2.

Finally, for future reference, we state the following results.

Lemma 6 *Let G be a group and $\alpha \in G$. Let $n = \prod_{i=1}^{k} p_i^{\beta_i}$ be the prime factorization of n. Then α has order n if and only if*
(i) $\alpha^n = 1$, and
(ii) $\alpha^{n/p_i} \neq 1$ for each i, $1 \leq i \leq k$. ■

Lemma 7 *Let G be an abelian group of type (cn, cn). If elements $\{\alpha_i\}$ are selected uniformly and randomly from G, then the elements $\{c\alpha_i\}$ are uniformly distributed about the elements of the subgroup of G of type (n, n).* ■

3 The Reduction

Let $E(F_q)$ be an elliptic curve over the finite field F_q, with group structure $\mathbb{Z}_{n_1} \oplus \mathbb{Z}_{n_2}$, where $n_2 | n_1$. Given the defining equation for $E(F_q)$, we can compute $\#E(F_q)$ in polynomial time by using Schoof's algorithm [17]. Also, we can determine n_1 and n_2 in probabilistic polynomial time by an algorithm due to Miller [11], given the integer factorization of $\gcd(\#E(F_q), q-1)$. We further assume that $\gcd(\#E(F_q), q) = 1$; it follows that $E[n_1] \cong \mathbb{Z}_{n_1} \oplus \mathbb{Z}_{n_1}$.

Let $P \in E(F_q)$ be a point of order n, where n divides n_1, and let $R \in E(F_q)$. We assume that n is known. The *elliptic curve logarithm problem* is the following: Given P and R, determine the unique integer l, $0 \leq l \leq n-1$, such that $R = lP$, provided that such an integer exists.

Since $e_n(P, P) = 1$, we deduce from Lemma 4 that $R \in < P >$ if and only if $nR = O$ and $e_n(P, R) = 1$, conditions which can be checked in probabilistic polynomial time. Henceforth, we will assume that $R \in <P>$.

We first describe an algorithm for obtaining partial information about l by solving a discrete logarithm problem in the field F_q itself, in the case that P has maximum order.

Algorithm 1

Input: An element $P \in E(F_q)$ of maximum order n_1, and $R = lP$.
Output: An integer $l' \equiv l \pmod{n'}$, where n' is a divisor of n_2.
(i) Pick a random point $T \in E(F_q)$.
(ii) Compute $\alpha = e_{n_1}(P, T)$ and $\beta = e_{n_1}(R, T)$.
(iii) Compute l', discrete logarithm of β to the base α in F_q.

Theorem 8 *Algorithm 1 correctly computes $l' \equiv l \pmod{n'}$, where n' is some divisor of n_2.*

Proof. Let n' denote the order of α; n' divides n_2 because of the following reasons. Let $G \in E(F_q)$ be an element of order n_2 such that the pair of points (P, G) generates $E(F_q)$, and let $T = c_1 P + c_2 G$. Then

$$\begin{aligned} \alpha^{n_2} &= e_{n_1}(P, T)^{n_2} \\ &= e_{n_1}(P, P)^{c_1 n_2} e_{n_1}(P, c_2 n_2 G) \\ &= e_{n_1}(P, O) \\ &= 1. \end{aligned}$$

Now, since

$$\begin{aligned} \beta &= e_{n_1}(R, T) = e_{n_1}(lP, T) \\ &= e_{n_1}(P, T)^l = \alpha^l \\ &= \alpha^{l'}, \end{aligned}$$

we can then determine l' by computing the discrete logarithm of β to the base α in F_q. ■

Since there are n_2 cosets of $<P>$ within $E(F_q)$, we deduce from Lemma 4 that the probability that $n' = n_2$ is $\phi(n_2)/n_2$. If n_2 is small however, then this method does not provide us with any significant information about l. In the remainder of this

Section, we describe a technique for computing l modulo n.

Let k be the smallest positive integer such that $E[n] \subseteq E(F_{q^k})$; it is clear that such an integer k exists.

Theorem 9 *There exists $Q \in E[n]$, such that $e_n(P,Q)$ is a primitive n^{th} root of unity.*

Proof. Let $Q \in E[n]$. Then, by the bilinearity of the Weil pairing, we have that

$$\begin{aligned} e_n(P,Q)^n &= e_n(P,nQ) \\ &= e_n(P,O) \\ &= 1. \end{aligned}$$

Thus $e_n(P,Q) \in \mu_n$, where μ_n denotes the subgroup of the n^{th} roots of unity in F_{q^k}.
There are n cosets of $<P>$ within $E[n]$, and by Lemma 5 we deduce that as Q varies among the representatives of these n cosets, $e_n(P,Q)$ varies among all of the elements of μ_n. The result now follows. ■

Let $Q \in E[n]$ such that $e_n(P,Q)$ is a primitive n^{th} root of unity. The next result is easy to prove.

Theorem 10 *Let $f : <P> \longrightarrow \mu_n$ be defined by $f : R \mapsto e_n(R,Q)$. Then f is a group isomorphism.* ■

We can now describe the method for reducing the elliptic curve logarithm problem to the discrete logarithm problem in a finite field.

Algorithm 2

Input: An element $P \in E(F_q)$ of order n, and $R \in <P>$.
Output: An integer l such that $R = lP$.
(i) Determine the smallest integer k such that $E[n] \subseteq E(F_{q^k})$.
(ii) Find $Q \in E[n]$ such that $\alpha = e_n(P,Q)$ has order n.
(iii) Compute $\beta = e_n(R,Q)$.
(iv) Compute l, the discrete logarithm of β to the base α in F_{q^k}.

Note that the output of Algorithm 2 is correct since

$$\begin{aligned} \beta &= e_n(lP,Q) \\ &= e_n(P,Q)^l \\ &= \alpha^l. \end{aligned}$$

Remarks
The reduction described in this section takes exponential time (in $\ln q$) in general, as k is exponentially large in general. Algorithm 2 is incomplete as we have not provided methods for determining k, and for finding the point Q. We shall accomplish this in the next section for the supersingular elliptic curves.

4 Supersingular Curves

In this Section, we prove that the reduction of Section 3 takes probabilistic polynomial time for supersingular curves, resulting in a probabilistic subexponential time algorithm for computing elliptic curve logarithms in these curves.

Let $E(F_q)$ be a supersingular elliptic curve of order $q+1-t$ over F_q, and let $q = p^m$. By Lemmas 1 and 2, E falls into one of the following classes of curves.
(I) $t = 0$ and $E(F_q) \cong \mathbb{Z}_q$.
(II) $t = 0$ and $E(F_q) \cong \mathbb{Z}_{(q+1)/2} \oplus \mathbb{Z}_2$ (and $q \equiv 3 \pmod 4$).
(III) $t^2 = q$ (and m is even).
(IV) $t^2 = 2q$ (and $p = 2$ and m is odd).
(V) $t^2 = 3q$ (and $p = 3$ and m is odd).
(VI) $t^2 = 4q$ (and m is even).

Let P be a point of order n in $E(F_q)$. Since $n_1 \mid (q+1-t)$, and $p \mid t$, we have $\gcd(n_1, q) = 1$. By applying the Weil conjecture and using Lemma 2, one can easily determine the smallest positive integer k such that $E[n_1] \subseteq E(F_{q^k})$, and hence $E[n] \subseteq E(F_{q^k})$. For convenience, we summarize the relevant information in Table 1. Note that for each class of curves, the structure of $E(F_{q^k})$ is of the form $\mathbb{Z}_{cn_1} \oplus \mathbb{Z}_{cn_1}$, for appropriate c. We now proceed to give a detailed description of the reduction.

Algorithm 3

Input: An element P of order n in a supersingular curve $E(F_q)$, and $R \in <P>$.
Output: An integer l such that $R = lP$.
(i) Determine k and c from Table 1.
(ii) Pick a random point $Q' \in E(F_{q^k})$ and set $Q = (cn_1/n)Q'$.
(iii) Compute $\alpha = e_n(P,Q)$ and $\beta = e_n(R,Q)$.
(iv) Compute the discrete logarithm l' of β to the

Class of curve	t	Group structure	n_1	k	Type of $E(F_{q^k})$	c
I	0	cyclic	$q+1$	2	$(q+1, q+1)$	1
II	0	$\mathbf{Z}_{(q+1)/2} \oplus \mathbf{Z}_2$	$(q+1)/2$	2	$(q+1, q+1)$	2
III	$\pm\sqrt{q}$	cyclic	$q+1 \mp \sqrt{q}$	3	$(\sqrt{q^3} \pm 1, \sqrt{q^3} \pm 1)$	$\sqrt{q} \pm 1$
IV	$\pm\sqrt{2q}$	cyclic	$q+1 \mp \sqrt{2q}$	4	(q^2+1, q^2+1)	$q \pm \sqrt{2q} + 1$
V	$\pm\sqrt{3q}$	cyclic	$q+1 \mp \sqrt{3q}$	6	(q^3+1, q^3+1)	$(q+1)(q \pm \sqrt{3q}+1)$
VI	$\pm 2\sqrt{q}$	$\mathbf{Z}_{\sqrt{q}\mp 1} \oplus \mathbf{Z}_{\sqrt{q}\mp 1}$	$\sqrt{q} \mp 1$	1	$(\sqrt{q} \mp 1, \sqrt{q} \mp 1)$	1

Table 1: Some information about supersingular curves

base α in F_{q^k}.
(v) Check whether $l'P = R$. If this is so, then $l = l'$ and we are done. Otherwise, the order of α must be less than n, so go to (ii).

Note that by Lemma 7, Q is a random point in $E[n]$. Note also that the probability that the field element α has order n is $\phi(n)/n$. This follows from Lemma 5 and the facts that there are $\phi(n)$ elements of order n in F_{q^k}, and there are n cosets of $<P>$ within $E[n]$.

We now proceed to prove the main result of this Section.

Proof of Theorem 11. We assume that a basis of the field F_q over its prime field is explicitly given. To do arithmetic in F_{q^k}, we need to find an irreducible polynomial $f(x)$ of degree k over F_q. This can be done in probabilistic polynomial time, for example by the method given in [15] (the method in [15] is described for prime fields, but is also applicable when q is a prime power). We then have $F_{q^k} \cong F_q[x]/I_f$, where I_f denotes the ideal generated by $f(x)$. Note that the constant polynomials in $F_q[x]$ form a subfield isomorphic to F_q. The point Q' can be chosen in probabilistic polynomial time since $Q' \in E(F_{q^k})$ and $k \leq 6$, and then Q can be determined in polynomial time. The elements α and β can be computed in probabilistic polynomial time by Miller's algorithm. Since

$$\frac{n}{\phi(n)} \leq 6 ln\ ln\ n$$

(see [16]), the expected number of iterations before we find a Q such that $e_n(P, Q)$ has order n is $O(ln\ ln\ n)$. Finally, observe that $l'P = R$ can be tested in polynomial time, and that $n = O(q)$. ■

Note that the discrete logarithm problem in F_{q^k} that we solve in (iv) has a base element α of order n, where $n < q^k - 1$. The probabilistic subexponential algorithms of [3], [4] and [5] for computing discrete logarithms in a finite field require that the base element be primitive. Using these algorithms, we obtain the proof of Corollary 12.

Proof of Corollary 12. The problem of finding the logarithm of β to the base α in F_{q^k} can be solved in probabilistic subexponential time as follows. We first obtain the integer factorization of $q^k - 1$ in probabilistic subexponential time using one of the many techniques available for integer factorization (for example [20] for a practical algorithm with a heuristic running time analysis, and [14] for an algorithm with a rigorous running time analysis). Observe that we apriori have the following partial factorizations of $q^k - 1$:
(I), (II) $q^2 - 1 = (q+1)(q-1)$.
(III) $q^3 - 1 = (q-1)(q+1-\sqrt{q})(q+1+\sqrt{q})$.
(IV) $q^4-1 = (q-1)(q+1)(q+1-\sqrt{2q})(q+1+\sqrt{2q})$.
(V) $q^6 - 1 = (q-1)(q+1)(q+1-\sqrt{3q})(q+1+\sqrt{3q})(q^2+q+1)$.
We then select random elements γ in F_{q^k}, until γ has order $q^k - 1$; the expected number of trials is $(q^k-1)/\phi(q^k-1)$ which is $O(ln\ ln\ q)$ since $k \leq 6$. The order of γ can be checked in polynomial time using Lemma 6. By solving two discrete logarithm problems in F_{q^k}, we find the unique integers s and t, $0 \leq s, t \leq q^k - 1$, such that $\alpha = \gamma^s$ and $\beta = \gamma^t$. Since $\beta = \alpha^{l'}$, we obtain the congruence $sl' \equiv t \pmod{q^k - 1}$. Let $w = \gcd(s, q^k - 1)$, and let $v = (q^k - 1)/w$ be the order of α. Then $l' = (s/w)^{-1}(t/w) \pmod{v}$.

The logarithms in F_{q^k} can be computed in prob-

abilistic subexponential time in $ln\ q^k$ (and consequently also subexponential in $ln\ q$) using, for example, the algorithm in [4] if q is prime and $k = 1$, [5] if q is prime and $k > 1$, or [3] if q is the proper power of a small prime. ■

In solving the elliptic curve logarithm problem in practice, one would first factor n. Using this factorization, we can easily check the order of α. Thus to find Q, we repeatedly choose random points in $E[n]$ until α has order n. This avoids the possibility of having to solve several discrete logarithm problems before l' is in fact equal to l. Note however that this modified reduction is different from the reduction described in Algorithm 3, and in particular is no longer a probabilistic polynomial time reduction to the discrete logarithm problem in a finite field.

The dominant step of the algorithm as modified in the previous paragraph is the final stage of computing discrete logarithms in F_{q^k}. The expected running time of the algorithm is thus $L[1/2, q^k]$ if q is a prime, and $L[1/3, q^k]$ if q is the power of a small prime.

We conclude that for the supersingular curves, the elliptic curve discrete logarithm problem is more tractable than previously believed. Among these special elliptic curves are the following:

(A) $y^2 + y = x^3 + b$ over F_{2^m}, m odd (class I).
(B) $y^2 = x^3 - ax$ over F_p, where $p > 3$ is a prime, a is a quadratic non-residue in F_p, and $p \equiv 3 \pmod 4$ (class I).
(C) $y^2 = x^3 - ax$ over F_p, where $p > 3$ is a prime, a is a quadratic residue in F_p, and $p \equiv 3 \pmod 4$ (class II).
(D) $y^2 = x^3 + b$ over F_p, where $p > 3$ is a prime, and $p \equiv 2 \pmod 3$ (class I).

We will discuss these curves further in the next section.

5 Cryptographic Implications

In order to implement the Diffie-Hellman and El Gamal protocols [8], one would like a cyclic group which is relatively easy to exponentiate in, and one for which the discrete logarithm problem is intractable.

Elliptic curve cryptosystems have the potential to be implemented efficiently with relatively small block size, and high security. (This was, of course, the motivation for studying such systems.) With current schemes, such as RSA and discrete exponentiation in a finite field, block sizes in excess of 500 bits (and preferably 1,000 bits) are necessary for adequate security. The results of the preceding section demonstrate that some care must be exercised in selecting an elliptic curve over a finite field. This is not unlike the situation with RSA where the prime numbers must be judiciously chosen. It is now clear that the curve

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

over F_{2^m} is no more secure than using the cyclic group of non-zero elements in $F_{2^{2m}}$. Since it appears that the cost of computations on the curve is higher than the cost of computations in $F_{2^{2m}}$, such a curve is inferior for cryptographic purposes to other existing systems. Similar statements are valid for the classes of curves (B), (C) and (D) of Section 4.

The curve $y^2 + y = x^3$ over F_{2^m} was first considered for the implementation of elliptic curve cryptosystems by Koblitz [8]. In [2], the authors suggested the particular values $m = 61$ and $m = 127$. Since the discrete logarithm problem in the fields $F_{2^{122}}$ and $F_{2^{254}}$ are very tractable using the index-calculus methods, these curves are clearly inadequate for cryptographic purposes. The particular values $m = 191$ and $m = 251$ were suggested in [9]. These curves should also be avoided by the comments made in the previous paragraph. The class of curves (B) and (C) were suggested by Miller [12]. Finally, the class of curves (D) was suggested in [2] for the implementation of elliptic curve cryptosystems, and by Kaliski [6] for the implementation of secure pseudorandom number generators.

The following cyclic curves over F_{2^m} (m odd)

$$E_1 : y^2 + y = x^3 + x$$

and

$$E_2 : y^2 + y = x^3 + x + 1$$

are much more attractive since they are easily implementable (see [9]), and give a security level that is apparently equivalent to the multiplicative group of $F_{2^{4m}}$ ($k = 4$).

It should be noted that although the supersingular curves over F_{2^m} have received the most attention to date, this does not mean that the more general class of curves is unattractive. Some work has been done on the implementation of non-supersingular curves over F_{2^m} and curves over F_q, q odd, and this will be reported in [10].

6 Acknowledgements

We wish to thank Neal Koblitz for conjecturing that the supersingular curves have a small k value. We would also like to thank Victor Miller for sending us a copy of his unpublished manuscript [11].

References

1. L. Adleman, K. Manders and G. Miller, "On taking roots in finite fields", *Proceedings of the 20th Annual Symposium on the Foundations of Computer Science* (1979), 175-178.

2. A. Bender and G. Castagnoli, "On the implementation of elliptic curve cryptosystems", *Advances in Cryptology – Proceedings of Crypto '89*, Lecture Notes in Computer Science, **435** (1990), Springer-Verlag, 417-426.

3. D. Coppersmith, "Fast evaluation of logarithms in fields of characteristic two", *IEEE Transactions on Information Theory*, **IT-30** (1984), 587-594.

4. D. Coppersmith, A. Odlyzko and R. Schroeppel, "Discrete logarithms in $GF(p)$", *Algorithmica*, **1** (1986), 1-15.

5. T. ElGamal, "A subexponential-time algorithm for computing discrete logarithms over $GF(p^2)$", *IEEE Transactions on Information Theory*, **IT-31** (1985), 473-481.

6. B. Kaliski, "A pseudorandom bit generator based on elliptic logarithms", *Advances in Cryptology – Proceedings of Crypto '86*, Lecture Notes in Computer Science, **293** (1987), Springer-Verlag, 84-103.

7. B. Kaliski, "Elliptic curves and cryptography: A pseudorandom bit generator and other tools", PhD thesis, M. I. T., January 1988.

8. N. Koblitz, "Elliptic curve cryptosystems", *Mathematics of Computation*, **48** (1987), 203-209.

9. A. Menezes and S. Vanstone, "The implementation of elliptic curve cryptosystems", *Advances in Cryptology – Proceedings of Auscrypt '90*, Lecture Notes in Computer Science, **453** (1990), Springer-Verlag, 2-13.

10. A. Menezes and S. Vanstone, "Elliptic curve cryptosystems and their implementation", in preparation.

11. V. Miller, "Short programs for functions on curves", unpublished manuscript, 1986.

12. V. Miller, "Uses of elliptic curves in cryptography", *Advances in Cryptology – Proceedings of Crypto '85*, Lecture Notes in Computer Science, **218** (1986), Springer-Verlag, 417-426.

13. A. Odlyzko, "Discrete logarithms and their cryptographic significance", *Advances in Cryptology – Proceedings of Eurocrypt '84*, Lecture Notes in Computer Science, **209** (1985), Springer-Verlag, 224-314.

14. C. Pomerance, "Fast, rigorous factorization and discrete logarithms algorithms", *Discrete Algorithms and Complexity* (1987), 119-143.

15. M. Rabin, "Probabilistic algorithms in finite fields", *SIAM Journal on Computing*, **9** (1980), 273-280.

16. J. Rosser and L. Schoenfield, "Approximate formulas for some functions of prime numbers", *Illinois Journal of Mathematics*, **6** (1962), 64-94.

17. R. Schoof, "Elliptic curves over finite fields and the computation of square roots mod p", *Mathematics of Computation*, **44** (1985), 483-494.

18. R. Schoof, "Nonsingular plane cubic curves over finite fields", *Journal of Combinatorial Theory*, **A 46** (1987), 183-211.

19. J. Silverman, *The Arithmetic of Elliptic Curves*, Springer-Verlag, New York, 1986.

20. R. Silverman, "The multiple polynomial quadratic sieve", *Mathematics of Computation*, 48 (1987), 329-339.

Appendix (Weil Pairing)

We give a brief introduction to the theory of divisors, define the Weil pairing, and outline Miller's algorithm for computing the Weil pairing. For a more thorough treatment of this subject, we refer to [19] and [11].

Let $K = F_q$ and let $\overline{K}$ denote its algebraic closure. Let E be an elliptic curve defined over K. If L is any field containing K, then $E(L)$ denotes the set of points on the curve whose coordinates are both in L. We will write E for $E(\overline{K})$.

A *divisor* D is a formal sum of points in E, $D = \sum_{P \in E} n_P(P)$, where $n_P \in \mathbb{Z}$, and $n_P = 0$ for all but finitely many $P \in E$. The *degree* of D is the integer $\sum n_P$. The divisors of degree 0 form an additive group, denoted D^0. The *support* of D is the set $\{P \mid n_P \neq 0\}$.

If E is defined by the equation $f(x, y) = 0$, $f \in K[x, y]$, then the *function field* $K(E)$ of E over K is the field of fractions of the ring $K[x, y]/I_f$. Similarly, $\overline{K}(E)$ is the field of fractions of $\overline{K}[x, y]/I_f$.

Let f be a non-zero function in $\overline{K}(E)$, i.e. $f \in \overline{K}(E)^\star$. For each $P \in E$, define $v_P(f)$ to be $n > 0$ or $-n < 0$ if f has a zero or a pole of order n at P, respectively. We associate the divisor $\sum v_P(f)(P)$ to f, and denote it by (f). One can verify that $(f) \in D^0$. A divisor $D = \sum n_P(P)$ is said to be *principal* if $D = (f)$ for some $f \in \overline{K}(E)^\star$. One can also verify that D is principal if and only if $\sum n_P = 0$ and $\sum n_P P = O$.

Let D_l denote the set of all principal divisors. Then D_l forms a subgroup of D^0. If $D_1, D_2 \in D^0$, we write $D_1 \sim D_2$ if $D_1 - D_2 \in D_l$. For each $D \in D^0$ there exists a unique point $P \in E$ such that $D \sim (P) - (O)$.

If $D = \sum n_P(P)$ is a divisor and $f \in \overline{K}(E)^\star$ such that D and (f) have disjoint supports, then we define $f(D) = \prod_{P \in E} f(P)^{n_P}$.

Now, let m be an integer coprime to q and let $P, Q \in E[m]$. Let $A, B \in D^0$ such that $A \sim (P) - (O)$ and $B \sim (Q) - (O)$, and A and B have disjoint supports. Let $f_A, f_B \in \overline{K}(E)$ be such that $(f_A) = mA$ and $(f_B) = mB$. Then the Weil pairing $e_m(P, Q)$ is

$$e_m(P, Q) = \frac{f_A(B)}{f_B(A)}.$$

Let $D_1, D_2 \in D^0$ with $D_1 = (P_1) - (O) + (f_1)$, $D_2 = (P_2) - (O) + (f_2)$, where $P_1, P_2 \in E$ and $f_1, f_2 \in \overline{K}(E)$. Then $D_1 + D_2 = (P_3) - (O) + (f_1 f_2 f_3)$, where $P_3 = P_1 + P_2$, and $f_3 = l/v$, where l is the equation of the line through P_1 and P_2, and v is the equation of the vertical line through P_3.

If $D = \sum n_P(P)$ is a principal divisor, then we can find $f \in \overline{K}(E)$ such that $D = (f)$ by first writing $D = \sum n_P((P) - (O))$, and then repeatedly using the method of the previous paragraph to compute each term of the summation. Notice that if $P \in E(K)$ for each P in the support of D, then $f \in K(E)$, and all intermediate computations take place in K. The problem with this method is that the bivariate rational function f may itself be of exponential size, relative to the size of the input D. Hence instead of writing f explicitly, i.e. writing down all of the non-zero coefficients and the corresponding monomials of f, we represent f by a straight-line program. The straight-line program will be of polynomial size, and f can be evaluated at points P in polynomial time (provided that $f(P)$ is defined). As a result of the method of this construction, the straight-line program representing f may be undefined on at most all points of the supports of the divisors occurring in the intermediate steps.

To find f_A and f_B to compute $e_m(P, Q)$, we pick random points $T, U \in E(K)$. Let $A = (P + T) - (T)$, $B = (Q + U) - (U)$. We then compute straight-line programs for f_A and f_B by the method described above. Finally, we compute

$$e_m(P, Q) = \frac{f_A(Q + U) f_B(T)}{f_B(P + T) f_A(U)}.$$

Let $1 = a_1, a_2, ..., a_t = m$ be a fixed addition chain for m, where $t \leq \log_2 m$. The value $e_m(P, Q)$ will be computed successfully provided that $P + T$

and T are distinct from a_iU and $a_i(U+Q)$, and also $Q+U$ and U are distinct from a_iT and $a_i(P+T)$, for each i, $1 \leq i \leq t$. The number of pairs of points (T,U) which do not satisfy the conditions above is at most $8tN$, where $N = \#E(K)$. Hence the probability that a randomly selected pair of points (T,U) does not satisfy the conditions is $8t/N$. Thus if $m = O(N)$, then the probability that $e_m(P,Q)$ is not computed successfully is negligible. If the computation fails, then we pick a new pair of random points $T, U \in E(K)$ and repeat.