EVALUATION REPORT FOR CRYPTREC: SECURITY LEVEL OF CRYPTOGRAPHY – ECDLP MATHEMATICAL PROBLEM

S.D. GALBRAITH AND N.P. SMART

Abstract. This report discusses the elliptic curve discrete logarithm problem and the known methods to solve it. We consider the implications of these methods for choosing the domain parameters in elliptic curve based cryptographic schemes. We also study special classes of elliptic curves. In particular, we discuss the security of Koblitz curves.

Contents

1. Introduction

The Elliptic Curve Discrete Logarithm Problem (ECDLP) is the foundation of a number of cryptographic protocols, for example EC-DSA, EC-DH, EC-MQV, EC-IES etc. In this report we discuss the current state of knowledge about the difficulty of the ECDLP. Before doing so we set up some notation which will be used throughout.

Let $K = \mathbb{F}_q$ denote a finite field. In practical systems one always either chooses q to be a large prime or one chooses q to be a power of two, these correspond to the cases of odd and even characteristic respectively. An elliptic curve over K is usually given in one of two forms

$$E : Y^2 = X^3 + aX + b$$

in the odd case, or

$$E : Y^2 + XY = X^3 + aX^2 + b$$

in the even case. In both cases we assume $a, b \in K$ are chosen to make the curve non-singular.

The set of points on an elliptic curve E over K, including the point at infinity which we write as $\mathcal{O}_E$, forms a finite abelian group denoted by $E(K)$. We write the group operation on points of $E(K)$ additively. For positive integers n we write

$$[n]P = \underbrace{P + P + \cdots + P}_{n \text{ times}}.$$

We sometimes also write this as nP. This is extended to all integers $n \in \mathbb{Z}$ using the inverse $-P$ of a point.

We write the order of this group as

$$\#E(K) = N.$$

We will always assume (see Section 2.2 for the reason) that $N = h \cdot l$ where l is a large prime number and h is small and called the *cofactor*.

Let $P = (x, y)$ be a point on an elliptic curve E over K. We write

$$\langle P \rangle = \{[n]P : n \in \mathbb{Z}\}.$$

This is the subgroup of $E(K)$ generated by the point P. The main problem which motivates elliptic curve cryptography is the following.

Elliptic Curve Discrete Logarithm Problem (ECDLP): Given $P, Q \in E(K)$ find the value of λ, if it exists, such that

$$Q = [\lambda]P.$$

For suitably chosen fields, curves and points this problem is believed to be computationally infeasible to solve.

Most cryptographic protocols in ECC actually rely for their security on weaker problems such as the elliptic curve Diffie-Hellman problem (EC-DHP) or the elliptic curve Decision Diffie-Hellman problem (EC-DDH). The cryptographic technique which we have been asked to study is the ECDLP, and so it will be the main focus of the present report. In order to accurately determine the security of cryptographic systems it is essential to also study the EC-DHP and EC-DDH. We recommend that the CRYPTREC organisation make further studies on these problems. In most instances it is conjectured that all three problems are polynomial time equivalent. One should however be aware that there are some elliptic curves for which there is strong evidence for a separation between these problems (see Joux and Nguyen [20]).

There is a certain family of curves which occurs quite frequently in elliptic curve cryptography and they are called the 'Koblitz curves'. Originally this term was used for curves defined over a subfield k of K, where we consider the group of points over the larger field for cryptographic purposes. The reason for considering these curves was that they possess an endomorphism structure which allows an efficient implementation of the various protocols. Nowadays, especially in standards such as SEC [1] [2], the name 'Koblitz curve' is used for any elliptic curve which possesses a special endomorphism structure which enables efficient implementation.

There are two common classes of Koblitz curves:

- The classical case of curves over $\mathbb{F}_2$ (sometimes called anomalous binary curves or ABC curves) which are given by
$$Y^2 + XY = X^3 + aX^2 + 1$$
where $a \in \{0, 1\}$. These curves are defined over $\mathbb{F}_2$ but we work in the group of points defined over the field $\mathbb{F}_{2^n}$.
- The more recent case of Koblitz curves over a large prime field which have a suitable endomorphism.

Further details of these cases are given in Section 4.

2. Known generic attacks

In this section we consider a number of generic attacks against the ECDLP. The name 'generic' refers to the fact that such attacks will apply in any group and not just an elliptic curve group. What makes elliptic curves particularly attractive is that for well chosen parameter sets the following generic attacks are the best possible with current knowledge.

We always assume we have a discrete logarithm problem given by
$$Q = [\lambda]P,$$
where P and Q are given and the goal is to find λ.

2.1. **Exhaustive search.** This is the most elementary attack. One simply computes
$$R = [\mu]P$$
for $\mu = 1, 2, 3, \ldots$, and checks whether $R = Q$. When equality is reached we conclude $\mu = \lambda$.

This algorithm requires $O(1)$ storage, but requires $O(N)$ time in both the worst and average case.

2.2. **Pohlig-Hellman.** The discrete logarithm problem in a group G (for instance, $G = E(K)$) is only as hard as the discrete logarithm problem in the largest subgroup of prime order in G. This observation is due to Pohlig and Hellman [31], and their method works in an arbitrary finite abelian group.

To explain the Pohlig-Hellman algorithm, suppose we have a finite cyclic abelian group G whose order is given by

$$N = \#G = \prod_{i=1}^{t} p_i^{e_i}.$$

The case of non-cyclic groups may be handled analogously. We assume that the number N can be factored (this assumption is valid for the ECDLP since group orders for elliptic curve cryptography are rather small compared to current factoring records).

Now suppose we are given two points $P, Q \in G$ such that there exists an integer λ such that

$$Q = [\lambda]P$$

Our aim is to find λ by first finding it modulo $p_i^{e_i}$ and then using the Chinese Remainder Theorem to recover it modulo N.

From basic group theory we know that there is a group isomorphism

$$\phi : G \to C_{p_1^{e_1}} \times \cdots \times C_{p_t^{e_t}},$$

where C_{p^e} is a cyclic subgroup of G of prime power order p^e. The projection of ϕ to the component C_{p^e} is given by

$$\phi_p : \left\{ \begin{array}{ccc} G & \to & C_{p^e} \\ R & \longmapsto & [N/p^e]R \end{array} \right.$$

The map ϕ_p is a group homomorphism so if we have $Q = [\lambda]P$ in $\langle P \rangle$ then we will have $\phi_p(Q) = [\lambda]\phi_p(P)$ in C_{p^e}. But the discrete logarithm in C_{p^e} would only be determined modulo p^e. Therefore, solving the discrete logarithm problem in C_{p^e} determines λ modulo p^e. Doing this for all primes p dividing N would allow us to solve for λ using the Chinese Remainder Theorem.

The only problem is that we have not shown how to solve the discrete logarithm problem in C_{p^e}. We shall now show how this is done, by reducing to solving e discrete logarithm problems in the group C_p.

Suppose $P, Q \in C_{p^e}$ and that there is an λ such that

$$Q = [\lambda]P.$$

Clearly λ is only defined modulo p^e and we can write

$$\lambda = \lambda_0 + \lambda_1 p + \cdots + \lambda_{e-1}p^{e-1}.$$

We find $\lambda_0, \lambda_1, \ldots$ in turn, using the following inductive procedure. Suppose we know λ', the value of λ modulo p^t, i.e.

$$\lambda' = \lambda_0 + \cdots + \lambda_{t-1}p^{t-1}.$$

We now wish to determine λ_t and so compute λ modulo p^{t+1}. We write

$$\lambda = \lambda' + p^t\lambda'',$$

and we have that

$$Q = [\lambda']P + [\lambda'']\left([p^t]P\right).$$

So if we set

$$Q' = Q - [\lambda']P \text{ and } P' = [p^t]P,$$

then

$$Q' = [\lambda'']P'.$$

Now P' is an element of order p^{e-t} so to obtain an element of order p, and hence a discrete logarithm problem in C_p we need to multiply the above equation by $s = p^{e-t-1}$. So setting

$$Q'' = [s]Q' \text{ and } P'' = [s]P'$$

we obtain the discrete logarithm problem in C_p given by

$$Q'' = [\lambda_t]P''.$$

So assuming we can solve discrete logarithms in C_p we can find λ_t and so find x.

Since the intermediate steps in the Pohlig-Hellman algorithm are quite simple, the difficulty of solving a general discrete logarithm problem will be dominated by the time required to solve the discrete logarithm problem in the cyclic subgroups of prime order.

For elliptic curve cryptography it follows that the security depends on the largest prime factor of the order of the group, say l. For efficiency we prefer that the subgroup contain a large proportion of all the points on the curve.

Implication 1. *For elliptic curve cryptography we select an elliptic curve such that*

$$\#E(K) = N = h \cdot l$$

where l is a large prime and h is very small. Usually one chooses $h = 1, 2$ or 4.

2.3. **Baby-Step/Giant-Step.** In our above discussion of the Pohlig-Hellman algorithm we assumed we had an algorithm to solve the discrete logarithm problem in cyclic groups of prime order. We shall now describe a general method of solving such problems which is more efficient than exhaustive search. This method is due to Shanks and is called the Baby-Step/Giant-Step method. Once again this is a generic method which applies to any cyclic finite abelian group.

Again we fix notation as follows. We have a public cyclic subgroup $G = \langle P \rangle$ of some elliptic curve group $E(K)$, which we can now assume to have prime order l. We are also given a point $Q \in G$ and are asked to find the value of λ modulo l such that

$$Q = [\lambda]P.$$

We assume there is some fixed encoding of the elements of G, so in particular it is easy to store, sort and search a list of elements of G.

The principle behind the Baby-Step/Giant-Step method is a standard divide and conquer approach found in many areas of computer science. We first write

$$\lambda = \lambda_0 + \lambda_1 \lceil \sqrt{l} \rceil.$$

Since $\lambda \leq l$ we have that $0 \leq \lambda_0, \lambda_1 < \lceil \sqrt{l} \rceil$.

We now compute the list of Baby-Steps

$$P_i = [i]P \text{ for } 0 \leq i < \lceil \sqrt{l} \rceil.$$

The pairs

$$(P_i, i)$$

are stored in a table so that one can easily search for items indexed by the first entry in the pair. This can be accomplished by sorting the table on the first entry or, more efficiently, by the use of hash-tables. To compute and store the Baby-Steps clearly requires

$$O(\lceil\sqrt{l}\rceil)$$

time, and a similar amount of storage.

We now compute the Giant-Steps. Let $P' = [\lceil\sqrt{l}\rceil]P$ and compute

$$Q_j = Q - [j]P' \text{ for } 0 \leq j < \lceil\sqrt{l}\rceil.$$

We then try to find a match in the table of Baby-Steps, i.e. we try to find a value P_i such that $P_i = Q_j$. If such a match occurs we have

$$\lambda_0 = i \text{ and } \lambda_1 = j$$

since, in that case,

$$[i]P = Q - [j\lceil\sqrt{l}\rceil]P,$$

and so

$$[i + j\lceil\sqrt{l}\rceil]P = Q.$$

Notice that the time to compute the Giant-Steps is at most

$$O(\lceil\sqrt{l}\rceil).$$

Hence, the overall time and space complexity of the Baby-Step/Giant-Step method is

$$O(\sqrt{l}).$$

This complexity is for both the worst and average cases of running the algorithm.

It is known (see Shoup [36]) that the Baby-Step/Giant-Step method is the fastest possible method for solving the discrete logarithm problem in a 'black box group'. Black box groups are a theoretical tool which allow the analysis of algorithms in an idealised setting. A black box group is a group modelled in such a way that the representations of field elements provides no structure. Of course in any particular group there may be a special purpose algorithm which works faster, but in general the Baby-Step/Giant-Step method is the best possible.

In conclusion, combining the Baby-Step/Giant-Step method with the Pohlig-Hellman algorithm, if we wish a discrete logarithm problem in a group G to be as difficult as a work effort of 2^{80} operations, then we need the group G to have a prime order subgroup of order at least 2^{160}.

Implication 2. *This means that for elliptic curve cryptography we select a curve such that*

$$\#E(K) = N = h \cdot l$$

where

$$l > 2^{160}.$$

2.4. **Pollard methods.** The trouble with the Baby-Step/Giant-Step method is that, although its run time is bounded by $O(\sqrt{l})$, it also requires $O(\sqrt{l})$ space. This space requirement makes the algorithm infeasible in practice. Hence, it is desirable to reduce the large space requirement while still obtaining a time complexity of $O(\sqrt{l})$. Pollard achieved this [32], but the method only has an expected running time rather than an absolute bound on the running time. The resulting algorithm is therefore of the "Las Vegas" type.

The methods for reducing the space complexity all make use of random walks, and a number of techniques exist in the literature almost all of which are due to Pollard (such as the rho and lambda and kangaroo methods).

These algorithms were all developed for serial computers. In real life when one uses random walk based techniques to solve discrete logarithm problems one uses a parallel version due to van Oorschot and Wiener [30]. The parallel methods are easily distributed over the internet. We now describe the parallel Pollard method as it is used in practice.

Suppose we are given the discrete logarithm problem

$$Q = [\lambda]P$$

in a subgroup $G = \langle P \rangle$ of an elliptic curve. The order of P is assumed to be a prime l. We first construct an easily computable function

$$h : G \to \{1, \ldots, k\},$$

where k is usually around 16. For example, such a function can be obtained by selecting a suitable window of 4 bits in the binary representation of elements of G, and mapping to an integer obtained from these bits in the natural way.

Then we define a set of "multipliers" M_i, these are produced by generating random integers $a_i, b_i \in [0, \ldots, l-1]$ and then setting

$$M_i = [a_i]P + [b_i]Q.$$

(In fact, when working in a cyclic group such as $\langle P \rangle$ it is possible to set the $b_i = 0$ as long as the t_0 below are always taken to be distinct).

To start a random walk we pick random $s_0, t_0 \in [0, \ldots, l-1]$ and compute

$$P_0 = [s_0]P + [t_0]Q,$$

the random walk is then defined on the triples (P_i, s_i, t_i) where

$$\begin{aligned} P_{i+1} &= P_i + M_{h(P_i)}, \\ s_{i+1} &= s_i + a_{h(P_i)} \pmod{l}, \\ t_{i+1} &= t_i + b_{h(P_i)} \pmod{l}. \end{aligned}$$

Hence, for every P_i we record the values of s_i and t_i such that

$$P_i = [s_i]P + [t_i]Q.$$

Suppose we have m processors, each processor starts a from a different starting position using the same process to determine the next element in the deterministic 'random' walk. When two processors, or even the same processor, meet an element of the group that has been seen before then we obtain an equation

$$[s_i]P + [t_i]Q = [s'_j]P + [t'_j]Q,$$

from which we can solve for the discrete logarithm λ with high probability (the algorithm succeeds unless $t_i \equiv t'_j \pmod{l}$). It can be shown [30] that we expect that after roughly $\sqrt{\pi l/2}/m$ iterations of these parallel walks that we will find a collision and so solve the discrete logarithm problem. Hence the method requires $O(\sqrt{l}/m)$ computation time.

However, as described above, each processor needs to return every element in its computed random walk to a central server who then stores all the computed elements. This is highly inefficient as the storage requirements will be very large, namely $O(\sqrt{l})$. We can reduce the storage requirements to any value as follows.

We define a function d on the group

$$d : G \rightarrow \{0, 1\}$$

such that $d(g) = 1$ around $1/2^t$ of the time. For example, the function d is often defined by returning $d(g) = 1$ if a certain subset of t of the bits representing a group element R are set to zero. The elements in G for which $d(g) = 1$ will be called *distinguished.*

Now it is only the distinguished group elements which are transmitted back to the central server. In other words, only 1 in 2^t points in the random walk are sent to the server. This means that one expects the random walks to need to continue another 2^t steps in the worst case before a collision occurs between two random walks. Hence, the computing time now becomes

$$O\left(\sqrt{l}/m + 2^{t-1}\right)$$

in the average case, whilst the storage becomes

$$O\left(\sqrt{l}/2^t\right).$$

Both these complexity estimates are for the average case running time and storage requirements. Since the algorithms are Las Vegas in nature there is the possibility that they never terminate (although this is highly unlikely). In addition the distribution of the running time can have quite a large variance.

One might be tempted to balance the above two equations by choosing $2^{t-1} \approx \sqrt{\pi l/2}/m$ so that the time requirement is $O(\sqrt{l}/m)$ while the space requirement is $O(m)$. In practice this is not what is done, since any single processor probably cannot execute $2^t = O(\sqrt{l}/m)$ operations. Instead, the space requirement is chosen in such a way that the central server has enough memory, and that single processors can produce distinguished points at an convenient rate (for instance, one or two distinguished points each day).

It must be remembered that all the distinguished points are sent to a central server where they are stored. This leads to practical memory and network considerations which are an obstacle to performing this method in very large groups.

Finally, we note that a parallel random walk attack similar to the one mentioned above is known for finding collisions in cryptographic hash functions. This means that elliptic curve cryptosystems should have parameter sizes chosen according to the same security criteria as hash function output sizes.

Implication 3. *When choosing elliptic curve parameter sizes it must be remembered that the work effort required is essentially reduced by a linear factor in terms of the number of processors available to an attacker.*
Elliptic curve parameter sizes should be chosen to be at least as large as recommended hash function output sizes.

2.5. **Practical considerations.** As already remarked it is usual to choose

$$l \approx N,$$

and general theory implies that $N \approx q$. So in most fielded elliptic curve cryptosystems we have

$$l \approx q$$

which means that the size of the field has a strong influence on the difficulty of the ECDLP.

The parallel Pollard method with distinguished points is the method of choice to solve the generic ECDLP. In 1997 Certicom announced a series of elliptic curve challenges, and the ones which have been successfully solved have all been done using the parallel Pollard method with distinguished points. Currently the record stands at an elliptic curve over a 97 bit finite field for a general curve and a Koblitz curve over a 109 bit characteristic 2 finite field. We will see in Section 3.1 why the parallel Pollard method can be made more efficient for Koblitz curves.

The amount of computing time needed to solve the 97 bit or 109 bit ECDLP problems in the challenges is about comparable with, or even greater than, the computing time needed to factor a 512 bit RSA modulus.

To get some idea of the size of resources deployed in these challenges, one typically would use $m = 9500$ machines and taking the proportion of distinguished points at about $1/2^{32}$, i.e. taking $t = 32$. Hence, for the 97 bit challenge we would expect to need storage of around 116159 points. The elapsed time needed to find the solution would be be on average equivalent to the time needed to perform

$$2^{35}$$

group operations. Hence, if each group operation could be performed in one tenth of a micro second, which is actually quite slow, one would expect an average elapsed time of 60 days. Assuming that is all 9500 machines operated on this problem for that length of time.

The MIPS estimates in Tables 1 and 2 for the ECDLP and the FACTORING problem are taken from the paper [23] and allow one to compare comparable key sizes for ECC and RSA keys. A similar comparison is given in [24] where similar conclusions are reached

Table 1. MIPS years to solve a generic ECDLP using parallel Pollard methods

q	$\sqrt{\pi q/2}$	MIPS Years
160	2^{80}	8.5×10^{11}
186	2^{93}	7.0×10^{15}
234	2^{117}	1.2×10^{23}
354	2^{177}	1.3×10^{41}
426	2^{213}	9.2×10^{51}

Table 2. MIPS years to factor an RSA modulus using NFS

RSA Size	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}

3. Known special attacks

We now consider attacks which apply to certain special classes of elliptic curves.

3.1. **Equivalence classes.**

The use of equivalence classes to accelerate the Pollard methods was first noticed by Gallant, Lambert and Vanstone [13] and Wiener and Zuccherato [43].

The principle behind the Pollard methods is that two random walks on a set of size l are likely to have a collision after approximately $\sqrt{\pi l/2}$ steps. Hence, if the size of the set can be reduced then the time required to find collisions is also reduced. The principle behind the equivalence classes method is that, if there is a convenient equivalence relation on the set, then one can consider a random walk on the set of *equivalence classes* rather than the whole set. The only difficulty is that it must be easy to construct a random walk which is well-defined on equivalence classes.

A basic example which works for all elliptic curves is the following, which is based on inverses of a point. If $P = (x, y)$ then $-P$ is either $(x, -y)$ in the odd case or $(x, y + x)$ in the even case. In both cases it is computationally trivial to pass from P to $-P$ (this is quite unlike the case of finite fields, where computing g^{-1} from g is a complicated arithmetic operation). Define the equivalence relation $P \sim Q$ if $Q = \pm P$ and consider the set of equivalence classes $S = E(K)/\sim$. Note that $\#S \approx \#E(K)/2$.

We want to define a random walk on the set S as we did in Section 2.4 but we have to take care of the following: when we have P_i being mapped to $P_{i+1} = P_i + M_{h(P_i)}$ we must also have $-P_i$ mapped to $\pm P_{i+1}$. This means that we must be very careful in evaluating the function h which determines the choice of multiplier. A way to define h on equivalence classes is to impose an ordering on the finite field elements. Then, for a given point $P = (x, y)$ one may compute $-P = (x, y')$ and then determine whether $y \leq y'$ or not. Finally one defines the function h as before, but in terms of the uniquely specified point whose y-coordinate has the minimal value.

Now that h is uniquely defined on equivalence classes it remains to construct the random walk on equivalence classes. One way to do this is to define $P_{i+1} = P_i + M_{h(P_i)}$ if y is the minimal value, and to define $P_{i+1} = P_i - M_{h(P_i)}$ if y' is the minimum value. The corresponding values of s_i and t_i are updated as $s_{i+1} = s_i \pm a_{h(P_i)}$ accordingly. Note that if $P_i = (x, y)$ is the point with minimal y-coordinate and $-P_i = (x, y')$ is the other point, then $-P_{i+1} = -(P_i + M_{h(P_i)}) = (-P_i) - M_{h(-P_i)}$ which shows that the map is well-defined on equivalence classes. Note that, in the notation of Section 2.4, we have $P_i = [s_i]P + [t_i]Q$ and $-P_i = [-s_i]P + [-t_i]Q$.

Finally, we must consider the distinguished points. We can use the same definition as before for distinguished points of $E(K)$ and now define an equivalence $\{\pm P\}$ of S to be distinguished if either P or $-P$ is distinguished. The central server stores distinguished points and the values s_i, t_i. When two processors find the same distinguished equivalence class then we have

$$[s_i]P + [t_i]Q = \pm([s_j']P + [t_j']Q).$$

The discrete logarithm problem can then be solved with high probability.

This trick of using the equivalence classes $\{\pm P\}$ can be applied to all elliptic curves. The way to abstract this method is to think of the process ± as being an easily computed endomorphism of the elliptic curve. Therefore, whenever we have an easily computed endomorphism of a curve then we are able to perform a similar method to speed up the Pollard methods. We now give further examples.

Consider the case of an elliptic curve E defined over $\mathbb{F}_2$ but considered as a group over an extension field $\mathbb{F}_{2^m}$. In this case there is the Frobenius endomorphism

$$\pi : (x, y) \mapsto (x^2, y^2)$$

on the curve E. On points $P = (x, y) \in E(\mathbb{F}_{2^m})$ we have $\pi^m(P) = (x^{2^m}, y^{2^m}) = P$. In fact, there is an integer λ such that $\pi(P) = [\lambda]P$ for every point $P = (x, y) \in E(\mathbb{F}_{2^m})$. Note that computing $\pi(P)$ from its definition as the Frobenius map is much faster than computing $[\lambda]P$ using elliptic curve point multiplication algorithms, and this is the key to the speed-ups using Frobenius expansions.

We can define an equivalence relation $P \sim Q$ if $P = \pm\pi^n(Q)$ for some n. We want to construct a random walk on the set of equivalence classes $S = E(\mathbb{F}_{2^m})/\sim$.

Given a point $P_i = [s_i]P + [t_i]Q$ we must obtain a new point in the random walk. We need a function h which maps all points in the same equivalence class to a number in $\{1, \ldots, 16\}$ or so. This is done by searching through the equivalence class to find a point which has a certain property (e.g., the y-coordinated is minimal subject to an ordering condition). Once such a function h is provided there are two ways to proceed.

The first approach makes use of the fact that we found a unique representative of the equivalence class while computing h. If the input point is $P_i = [s_i]P + [t_i]Q$ and if the unique representative is $P_i' = (-1)^j \pi^k(P_i)$ then we obtain the corresponding s_i' and t_i' via

$$s_i' = (-1)^j \lambda^k s_i \pmod{l} \qquad \text{and} \qquad t_i' = (-1)^j \lambda^k t_i \pmod{l}.$$

One can then define a random walk using multipliers as before, and compute the s_{i+1} and t_{i+1} accordingly.

Another approach is to use a "next step" function which is well-defined on equivalence classes (instead of using fixed multipliers M_i). Such a process (which is used in [13]) is to define the walks as

$$P_{i+1} = P_i + \pi^{h(P_i)}(P_i)$$

so that $s_{i+1} = (1 + \lambda^{h(P_i)} s_i)$ etc.

Either method can be used. In both cases we are using equivalence classes of size $2m$ and so the expected running time of the algorithm is improved from approximately $\sqrt{\pi l/2}$ steps to approximately $\sqrt{\pi l/(4m)}$ steps.

The Frobenius endomorphism is particularly useful but it only occurs for elliptic curves defined over subfields. However, even with elliptic curves defined over prime fields there are sometimes endomorphisms available. An example of this occurs with the elliptic curve

$$E : y^2 = x^3 + A.$$

This elliptic curve has j-invariant $j(E) = 1728$ and it has the endomorphism

$$\phi : (x, y) \longmapsto (\beta x, y)$$

where $\beta^3 = 1$. As with the ± and Frobenius maps, this is actually an automorphism (in other words, it has no kernel). Also, it is clear that this map has order three.

Hence we can obtain an equivalence relation as $P \sim Q$ if $P = \pm\phi^j(Q)$ for some j. The equivalence classes have size 6. All the techniques developed above can be applied in this case and the algorithm runs faster by a factor of $\sqrt{6}$.

Similarly, the elliptic curve

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

(which has j-invariant $j(E) = 0$) has the automorphism

$$(x, y) \longmapsto (-x, iy)$$

of order 4, where $i^2 = -1$. This can also be used to speed up the Pollard methods by a factor of 2.

With all of these methods there is a risk of getting trapped in small cycles. In practice these are easily eliminated and we refer to [13, 43] for details.

The computational time of the equivalence classes method is the following: If the equivalence classes have size m and the group is of size l then the expected running time of the method is

$$c\sqrt{\pi l/2m}$$

where the constant c depends on the cost of computing a fixed representative for each equivalence class. Of course, from the point of view of computational complexity, the constant c is $O(m)$ and so the equivalence classes method is asymptotically slower! But in practice it gives a very significant improvement to the running time, as the constant c is rather small.

We now discuss how this method effects the choice of parameters for elliptic curve cryptosystems. Suppose we want to develop a cryptosystem which requires a work effort (for a single processor) of 2^{80} to be broken. Then we could take an elliptic curve over a finite field which has a subgroup of prime order l where $l > 2^{160}$. Now suppose we would like to take advantage of the efficiency benefits enabled by the Frobenius endomorphism, and so we take an elliptic curve E over $\mathbb{F}_2$ and consider the group $E(\mathbb{F}_{2^m})$. One might expect that taking $m = 163$ gives the desired level of security. In fact, with $m = 163$ one obtains an elliptic curve with a subgroup of order $l < 2^{163}$ (there is always some co-factor). The work effort to solve the ECDLP on such a curve using the Pollard method with equivalence classes is at most

$$c\sqrt{\pi l/2m} < c\sqrt{\pi 2^{163}/(2 \cdot 2 \cdot 163)} \leq c2^{81}/10.2 < c2^{78}.$$

In fact, to obtain 2^{80} security it is necessary to take the extension degree m to be at least 168.

Every elliptic curve over $\mathbb{F}_q$ has some non-trivial endomorphisms. So why can't they be used to improve the Pollard methods? The reason why they can't be used is that it is not usually efficient to find a canonical representative of an equivalence class. For instance, consider an elliptic curve $E(\mathbb{F}_q)$ with l points and consider an integer n which is coprime to l. Inspired by the above methods we might impose the equivalence relation on $E(\mathbb{F}_q)$ that $P \sim Q$ if and only if $P = \pm[n]^j Q$ for some j. The first observation is that the powers n^j may generate a large subgroup of $\mathbb{Z}_l^*$ and so the equivalence classes may be very large. More importantly, to define the mapping h on equivalence classes it is necessary to have some canonical representative. The task of searching through the equivalence class is very expensive since each step is an elliptic curve point multiplication. In the notation above, the "constant" c is a significant multiple of m and so the complexity is not reduced. In general, the equivalence classes method is a slower algorithm than the usual Pollard method.

Implication 4. *Koblitz curves are slightly less secure then random curves over the same field.*
In particular, Koblitz curves over $\mathbb{F}_{2^{163}}$ only require a work effort of approximately 2^{78} to break the ECDLP, rather than approximately 2^{81}.

3.2. **Weil pairing and Tate pairing attacks.** Menezes, Okamoto and Vanstone [27] were the first to show that the ECDLP may be transformed into a discrete logarithm problem in a finite field. Their method used the Weil pairing. This method was generalised by Frey and Rück [9] using the Tate pairing. We recall the Frey-Rück attack here.

Let $K = \mathbb{F}_q$ and let P and Q be points in $E(K)$ of order l. Suppose that l is coprime to q (see Section 3.4 for the case where $l|q$).

Let k be a positive integer such that the field $\mathbb{F}_{q^k}$ contains the lth roots of unity (in other words, $l|(q^k - 1)$ and k is the order of q in $\mathbb{Z}_l^*$). Let $G = E(\mathbb{F}_{q^k})$ and write $G[l]$ for the subgroup of points of order l. Write $lG = \{[l]P : P \in G\}$ and write G/lG for the quotient group. Similarly, write $(\mathbb{F}_{q^k}^*)^l = \{\alpha^l : \alpha \in \mathbb{F}_{q^k}^*\}$ and write $\mathbb{F}_{q^k}^*/(\mathbb{F}_{q^k}^*)^l$ for the quotient group. The groups $G[l]$, G/lG and $\mathbb{F}_{q^k}^*/(\mathbb{F}_{q^k}^*)^l$ all have exponent l (i.e., they are a product of cyclic groups of order l).

The Tate pairing is a mapping

$$\langle \cdot, \cdot \rangle : G[l] \times G/lG \to \mathbb{F}_{q^k}^*/(\mathbb{F}_{q^k}^*)^l. \tag{1}$$

The Tate pairing satisfies the following properties [9]:

1. (Well-defined) $\langle 0, Q\rangle \in (\mathbb{F}_{q^k}^*)^l$ for all $Q \in G$ and $\langle P, Q\rangle \in (\mathbb{F}_{q^k}^*)^l$ for all $P \in G[l]$ and all $Q \in lG$.
2. (Non-degeneracy) For each point $P \in G[l] - \{0\}$ there is some point $Q \in G$ such that $\langle P, Q\rangle \notin (\mathbb{F}_{q^k}^*)^l$.
3. (Bilinearity) For any integer n, $\langle [n]P, Q\rangle \equiv \langle P, [n]Q\rangle \equiv \langle P, Q\rangle^n$ modulo lth powers in $\mathbb{F}_{q^k}^*$.

There are more properties, but these are the ones which are used for the attack.

The Weil pairing is a similar function, but in general there is no relationship between the Tate pairing and the Weil pairing, as they are defined on different sets. However, when E is an elliptic curve such that $l^2 \| \#E(\mathbb{F}_{q^k})$ and P, Q are independent points in $E(\mathbb{F}_{q^k})[l]$ then we have that the Weil pairing is

$$e_l(P, Q) = \langle P, Q\rangle / \langle Q, P\rangle.$$

A consequence of the non-degeneracy property of the Weil pairing is that the Tate pairing is not symmetric. The Weil pairing requires working over the field $\mathbb{F}_q(E[l])$ generated by the coordinates of all the l-division points. In general, one would expect the Weil pairing to require a larger field than that used for the Tate pairing. One observation is that for elliptic curves these fields are usually the same.

Theorem 1. *(Koblitz) Let E be an elliptic curve over $\mathbb{F}_q$ and let l be a prime dividing $\#E(\mathbb{F}_q)$. Suppose that $l \not| (q-1)$. The $E[l] \subset E(\mathbb{F}_{q^k})$ if and only if $l|(q^k-1)$.*

The Tate pairing may be computed using the following process: Since $[l]P = 0$ there is some function f on the curve E such that the divisor of the function f, which is denoted (f), is equal to $l(P) - l(\mathcal{O}_E)$. Then $\langle P, Q\rangle = f(Q')$ where Q' is a divisor in the same class as Q such that the support of Q' is disjoint with the support of (f). This computation is easily and efficiently implemented in practice by using the double-and-add algorithm and evaluating all the intermediate functions at Q' (see [9], [10] for details).

The value $f(Q')$ lies in $\mathbb{F}_{q^k}$ and is only determined up to a multiple of an lth power. By raising it to the power $(q^k - 1)/l$ we obtain a precise lth root of unity.

One subtlety when implementing the Tate pairing is finding a divisor class Q' with support disjoint from the partial terms in the addition chain for lP. In the elliptic curve case this is done by taking $Q' = (Q + S) - (S)$ where Q is the target point and where S is an arbitrary point (not necessarily of order l).

We now recall how the Tate pairing is used to attack the ECDLP. The method proceeds as follows:

1. Choose random points $R \in E(\mathbb{F}_{q^k})$ until $\langle P, R\rangle \notin (\mathbb{F}_{q^k}^*)^l$.
2. Compute $\zeta_1 = \langle P, R\rangle, \zeta_2 = \langle Q, R\rangle \in \mathbb{F}_{q^k}^*$.
3. Map the ζ_i to lth roots of unity (by raising to the power $(q^k - 1)/l$). This step is actually optional since the linear algebra in the index calculus method should be performed modulo l.
4. Solve the discrete logarithm problem $\zeta_2 = \zeta_1^\lambda$ in the subgroup of order l of the finite field $\mathbb{F}_{q^k}^*$ using an index calculus method.

Note that the index calculus algorithms for solving the discrete logarithm problem in $\mathbb{F}_{q^k}^*$ have subexponential complexity in terms of the field size q^k (their performance is comparable with integer factorisation algorithms). Since the original problem is in a group of size q it is necessary that the subexponential complexity in terms of q^k be smaller than the complexity $O(\sqrt{q})$ of the Pollard methods. Hence, this strategy is only practical when k is relatively small.

It is known that supersingular curves are vulnerable to the Frey-Rück attack (since the value k is always less than or equal to 6). There are also non-supersingular curves which are vulnerable to this attack (e.g., curves of trace two over $\mathbb{F}_q$). However, if one chooses non-supersingular curves at random then the probability of finding a curve which is vulnerable to the Frey-Rück attack is negligible (more precisely, the probability is $O(1/\sqrt{q})$). Even more is true (as was shown by Balasubramanian and Koblitz [5]), if one chooses non-supersingular elliptic curves at random *so that the number of points is a prime*, then the probability of finding a curve which is vulnerable to the Frey-Rück attack is incredibly small (much smaller than $1/\sqrt{q}$).

Implication 5. *One should choose a curve such that*

$$l \not| q^k - 1$$

for all "small" values of k*, e.g.,* $k \leq 30$*.*
This test will eliminate all supersingular curves and curves of trace two, plus a few others.
We emphasise that this test is trivial to perform and that the probability of a random non-supersingular curve failing this test is negligible.

3.3. **Determining whether the ECDLP has a solution.** The definition of the ECDLP often includes the task of deciding whether Q lies in the subgroup generated by P or not.

In elliptic curve systems the point P usually has prime order l and the order N of the group $E(K)$ is known to all users.

If $N = l$ (i.e., the subgroup generated by P is the full set of points on the curve) then it is sufficient simply to test whether the point Q lies on the curve (i.e., whether it satisfies the curve equation). If so, then $Q \in \langle P \rangle$.

More generally there is some non-trivial cofactor $h = N/l$. Usually l^2 does not divide N (and this can be checked by all users of the system). This means that there is a unique subgroup of $E(K)$ of order l and it is the subgroup generated by P. Hence, if Q has order l then Q must lie in $\langle P \rangle$ and the ECDLP does have some solution λ. Hence, the simplest test is to simply check whether $[l]Q = \mathcal{O}_E$.

If the subgroup of points of order l in $E(K)$ is not cyclic then $l^2|N$ and the problem of checking whether the ECDLP has a solution is more difficult. Of course, this case usually does not arise in cryptography since, in that case, $l \leq \sqrt{N}$ which means the system has poor efficiency. Nevertheless, the Weil pairing may be used to determine whether $Q \in \langle P \rangle$ or not (no field extension is necessary in this case, since the theory implies that $l|(q-1)$). Briefly, if the pairing of P and Q is one then $Q \in \langle P \rangle$ while if the pairing is not one then $Q \notin \langle P \rangle$ and so the ECDLP does not have a solution.

Implication 6. *Deciding whether the ECDLP has a solution or not is rather easy in practice.*

3.4. **The anomalous curves attack.** An elliptic curve E defined over a prime field $\mathbb{F}_p$ is said to be anomalous if it has exactly p points.

In 1997 several researchers independently announced related methods to reduce the discrete logarithm problem on an anomalous elliptic curve $E/\mathbb{F}_p$ to the discrete logarithm problem in the additive group $\mathbb{Z}/p\mathbb{Z}$ of the integers modulo p. Nigel Smart [40] posted an announcement on the internet briefly describing a method to solve this problem. At the same time a preprint appeared by Satoh and Araki [4], which used identical methods. It then became known that Semaev [35] had already submitted a paper on this topic, and that Rück [33] had generalised this method to deal with Abelian varieties. In this section we will discuss the method of Semaev and Rück, restricted to the case of elliptic curves. This method is computationally more efficient than the methods proposed by Smart and Satoh–Araki. The papers [4], [40] give a very readable description of the alternative (p-adic logarithm) method. The paper by Voloch [42] puts the method into a more theoretical framework which demonstrates the analogy between these methods and the Tate pairing.

We note that anomalous curves $E/\mathbb{F}_p$ are very rare. There is approximately $1/(4\sqrt{p})$ chance of a random curve being anomalous. Also, this phenomenon has no impact for the case of elliptic curves over fields of small characteristic.

Suppose $E/\mathbb{F}_p$ is an elliptic curve such that $\#E(\mathbb{F}_p) = p$. In this section we will describe the mapping

$$E[p] \longrightarrow \mathbb{Z}/p\mathbb{Z} \tag{2}$$

which is the key element of the attack on anomalous elliptic curves.

We will describe the map (2) in two stages.

The first stage is a mapping described by Serre from $E[p]$ into $\Omega^1(E)$ (the space of holomorphic differentials on E). Given $P \in E[p]$ we know that there is some function f which has divisor $(f) = p(P) - p(\mathcal{O}_E)$. Define $\varphi(P)$ to be the differential $\omega_P := \frac{1}{f} df$. Then one can show that the map $\varphi : P \mapsto \omega_P$ is a well-defined homomorphism into $\Omega^1(E)$.

To complete the description of (2) we need the homomorphism $\Omega^1(E) \to \mathbb{Z}/p\mathbb{Z}$. We may expand any differential $\omega \in \Omega^1(E)$ in terms of a uniformizer t at a point P_0. In other words, we may write $\omega = \sum_j a_j t^j dt$. The Riemann-Roch theorem shows that a_0 cannot be zero since ω is holomorphic (so it has no poles) and also the divisor of ω has degree $2g - 2 = 0$ (and so it cannot have any zeroes either).

The map $\Omega^1(E) \to \mathbb{Z}/p\mathbb{Z}$ is simply $\omega = \sum_j a_j t^j dt \mapsto a_0$. That this is a homomorphism follows from the fact that one may add the power series $\sum_j a_j t^j$ in a termwise manner.

We now show how this map is used to solve the ECDLP on anomalous curves. The attack is very simple, we merely evaluate the map φ from the previous section on both P and Q. Note that this map can be efficiently computed using a version of the double-and-add method like that used to compute the Tate pairing. Since this is a homomorphism into the additive group $\mathbb{Z}/p\mathbb{Z}$, the discrete logarithm is simply $\lambda \equiv \varphi(Q)/\varphi(P) \pmod{p}$. This can be solved efficiently using the Euclidean algorithm.

Implication 7. *In the case of large odd characteristic one should always have*

$$l \neq q.$$

We emphasise that this test is trivial to perform and that the probability of a random non-supersingular curve failing this test is negligible.

3.5. **Weil descent.** This method applies to elliptic curves over field extensions of the form $\mathbb{F}_{q^n}$ where q is a prime or prime power and where $n > 1$. The principle is to transform the ECDLP from $E(\mathbb{F}_{q^n})$ to a discrete logarithm problem on the Jacobian of a curve C over $\mathbb{F}_q$. Since subexponential algorithms exist for the discrete logarithm problem in high genus curves, this gives a possible method of attack against the ECDLP.

The technique of Weil descent to solve the elliptic curve discrete logarithm problem (ECDLP) was first proposed by Frey [8]. This strategy was detailed further by Galbraith and Smart [11]. These papers were rather general in their scope, but were not detailed enough to give precise and efficient algorithms to solve the ECDLP for specific curves.

The work of Gaudry, Hess and Smart [16] was less general than the earlier works but gave much more powerful and efficient techniques. In particular, they gave a very efficient algorithm to reduce the ECDLP to the discrete logarithm in a Jacobian of a hyperelliptic curve over $\mathbb{F}_q$. We refer to the method of [16] as the GHS attack.

Menezes and Qu [28] analysed the GHS attack in some detail and demonstrated that it did not apply to the case when $q = 2$ and n is prime. Since this is the common case in real world applications, the work of Menezes and Qu means that the GHS attack does not apply to most deployed systems. However, there are a few fielded elliptic curve systems which use the fields $\mathbb{F}_{2^{155}}$ and $\mathbb{F}_{2^{185}}$, in the IPSEC standards. Hence there is considerable interest as to whether the GHS attack makes all curves over these fields vulnerable.

In [41] Smart examined the GHS attack for elliptic curves with respect to the field extension $\mathbb{F}_{2^{155}}/\mathbb{F}_{2^{31}}$ and concluded that such a technique was unlikely to work for any curve defined over $\mathbb{F}_{2^{155}}$. Jacobson, Menezes and Stein [19] also examined the field $\mathbb{F}_{2^{155}}$, this time using the GHS attack down to the subfield $\mathbb{F}_{2^5}$. They concluded that such a strategy could be used in practice to attack around 2^{33} isomorphism classes of elliptic curves defined over $\mathbb{F}_{2^{155}}$. Since there are about 2^{156} isomorphism classes of elliptic curves defined over $\mathbb{F}_{2^{155}}$, the probability of finding one where the GHS attack is applicable is negligible. Further analysis of the GHS attack has been given by Maurer, Menezes and Teske [26].

We now give some details of the GHS attack.

Let us first set up some notation. Throughout this section we let E denote an elliptic curve over the field $K = \mathbb{F}_{q^n}$ where $q = 2^r$. Let k denote the subfield $\mathbb{F}_q$. To simplify the discussion, and since those cases are the most important, we always assume that r and n are odd. We also assume that n is a prime. We stress that it is easy to obtain analogous results in the more general case.

We assume the curve is given by an equation of the form

$$E : Y^2 + XY = X^3 + aX^2 + b \text{ where } a \in \{0,1\}, b \in K.$$

We may assume that $a \in \{0,1\}$ since r and n are odd. We have points $P, Q \in E(K)$ of large prime order l (we may assume that $l \approx q^n$) and we aim to solve the discrete logarithm problem.

Define $\sigma : K \to K$ to be the q-power Frobenius automorphism, and let $\pi : K \to K$ denote the absolute Frobenius automorphism $\pi : \alpha \mapsto \alpha^2$. Therefore, $\sigma = \pi^r$.

The first step is to construct the Weil restriction of scalars $W_{E/k}$ of E over k, this is an n-dimensional abelian variety over k with the property that $W_{E/k}(k) \cong E(K)$. The variety $W_{E/k}$ is then intersected with $n-1$ carefully chosen hyperplanes so as to obtain a hyperelliptic curve C over the field k. Let g denote the genus of C.

In addition, the GHS attack gives an explicit and efficient group homomorphism from $E(K)$ to the Jacobian $J_C(k)$ of the curve C. Assuming some mild conditions, the Jacobian of C will contain a subgroup of order l and the image of the subgroup of order l in $E(K)$ will be a non-trivial subgroup of order l in $J_C(k)$.

One can then translate the original ECDLP into a discrete logarithm problem on the Jacobian of the curve C over $k = \mathbb{F}_q$. The available algorithms have complexity depending on q^g where g is the genus of the curve C. Hence the method is only successful when g is not too large.

One of the key results of [16] is the following.

Theorem 2 (Gaudry, Hess and Smart [16])**.** *The genus of C is equal to either 2^{m-1} or $2^{m-1} - 1$, where m is determined as follows. Let $b_i = \sigma^i(b)$, then m is given by*

$$m = m(b) = \dim_{\mathbb{F}_2}\left(\mathrm{Span}_{\mathbb{F}_2}\left\{(1, b_0^{1/2}), \ldots, (1, b_{n-1}^{1/2})\right\}\right).$$

In particular we have $1 \le m \le n$. If m is too small then the size of $J_C(k)$, which is $\approx q^g$, will be too small to contain a subgroup of size l. If m is too large then, although we can translate discrete logarithm problems to the hyperelliptic setting, the genus of the resulting curve is high and so the algorithms for solving the discrete logarithm problem are not effective. Hence, this case does not help us to solve the original ECDLP in practice.

Menezes and Qu proved the following theorem which characterises the smallest value of $m > 1$ and the elliptic curves which give rise to such m.

Theorem 3 (Menezes and Qu [28])**.** *Keeping the notation as above, and considering the GHS technique for Weil restriction of E from K down to k. Let n denote an odd prime, let t denote the multiplicative order of two modulo n and let $s = (n-1)/t$. Then*

(1) *The polynomial $x^n - 1$ factors over $\mathbb{F}_2$ as $(x-1)f_1 f_2 \cdots f_s$ where the f_i's are distinct irreducible polynomials of degree t. For $1 \leq i \leq s$ define*
$$B_i = \{b \in \mathbb{F}_{q^n} : (\sigma - 1)f_i(\sigma)b = 0\}.$$
(2) *For all $1 \leq i \leq s$ and all $b \in B_i$ the elliptic curves*
$$\begin{aligned} Y^2 + XY &= X^3 + b, \\ Y^2 + XY &= X^3 + \alpha X^2 + b \end{aligned}$$
have $m(b) \leq t + 1$, where α is a fixed element of K of trace one.

(3) *If $m(b) = t + 1$ then E must be one of the previous curves for some i and some $b \in B_i$.*

(4) *The cardinality of the set $\cup_{i=1}^{s} B_i$ is $qs(q^t - 1) + q$.*

In particular $m(b) = t + 1$ is the smallest attainable value of $m(b)$, which is greater than one, for the field $\mathbb{F}_{q^n}$ using the GHS technique for Weil restriction down to $\mathbb{F}_q$.

Menezes and Qu use the above theorem to show that if n is a prime in the range $160 \leq n \leq 600$ and if $q = 2$ then the GHS attack will be infeasible. These results are analysed in further detail by Jacobsson, Menezes and Stein [19] and Maurer, Menezes and Teske [26].

A new approach to Weil descent was very recently developed by Galbraith, Hess and Smart in [12]. They show that one can sometimes apply the GHS attack to a curve which has a large value of $m(b)$. The idea is to find an isogenous curve $E'(K)$ which has a small value of $m(b')$ and an isogeny
$$E(K) \longrightarrow E'(K).$$
The discrete logarithm problem in $E(K)$ can then be mapped in to the discrete logarithm problem in $E'(K)$ and then this can be mapped using the GHS method to the discrete logarithm problem in the Jacobian of a hyperelliptic curve of low genus. Efficient methods to find the isogenous curve and the isogeny are given in the paper [12], as well as a study as to how effective this extension to the GHS method is in practice.

While the work of Menezes and Qu shows that the GHS attack is not generally applicable to elliptic curve discrete logarithm problems over $\mathbb{F}_{2^p}$ where p is prime, it shows that there are some choices for p which are more dangerous than others (for instance, the case $p = 127$). The danger of these special primes is not serious in practice, since all primes in the range $160 < p < 600$ do not suffer from this risk.

One very important point is that the GHS attack is only one particular way to perform the general Weil descent strategy. One cannot deduce that a given instance of the elliptic curve discrete logarithm problem is hard simply by showing that the GHS attack cannot be applied. Hence, it is very important to think more generally about Weil descent as an attack on the ECDLP.

If we consider random elliptic curves over finite fields of the form $\mathbb{F}_{2^p}$ then Weil descent can only be performed with respect to the degree p extension $\mathbb{F}_{2^p}/\mathbb{F}_2$. The abelian variety A which arises from Weil descent therefore is defined over $\mathbb{F}_2$ and has dimension p. The difficulty of the ECDLP depends on whether there exist curves of 'small' genus on this variety. Unless the abelian variety A has some very special properties then it seems to be very unlikely that A always has such curves in it. Indeed, the analysis of the GHS attack has reinforced the opinions of researchers that Weil descent is only applicable to a very small proportion of elliptic curves if the extension field is chosen to have prime degree p.

Note that almost all research on Weil descent has been performed in characteristic 2, since this is the most important case. In fact, the ideas are easily applied to other finite fields $\mathbb{F}_{p^n}$ where p is odd and $n > 1$. The results in these cases are not as strong as in the case of characteristic two, but we still recommend against using such systems if n has a factor which lies between 4 and 10.

Implication 8. *The GHS and the Weil Descent methodology imply that one should take $q = 2^p$, where p is a prime in the even characteristic case.*
We emphasise that the Weil descent methods do not apply to elliptic curves over prime fields $\mathbb{F}_p$.

3.6. **Special finite fields.** There are implementation advantages from using elliptic curves over finite fields $\mathbb{F}_p$ where p is of a special form such as a generalised Mersenne number. These are primes of the form

$$p = f(2^w)$$

where w is a multiple of the word size and f is a sparse polynomial with coefficients drawn from $\{-1, 0, 1\}$. As an example we have

$$p = 2^{192} - 2^{64} - 1 = f(2^{64})$$

where

$$f = x^3 - x - 1.$$

We are not aware of any security risk associated with using these particular finite fields. There are no results in the theory of elliptic curve cryptography which suggest that some prime fields are more or less secure than others.

4. Koblitz curves versus general curves

The term 'Koblitz curves' originally referred to certain elliptic curves over the field $\mathbb{F}_2$. Koblitz [22] pointed out that there are two advantages to performing elliptic curve cryptography in the group $E(\mathbb{F}_{2^n})$ when E is defined over $\mathbb{F}_2$, namely:

(1) It is easy to compute the group order $\#E(\mathbb{F}_{2^n})$.
(2) The arithmetic can been made faster by using Frobenius expansions.

The first advantage above is no longer important as there are now extremely efficient algorithms for counting points on elliptic curves over fields of small characteristic [17]. The second advantage is still of interest, as it can lead to cryptographic systems with improved performance. Hence Koblitz curves have remained very popular for implementations of elliptic curve cryptography.

Since we always want curves whose group order is divisible by a large prime it follows that we must take the extension degree n to be prime (otherwise the curve have large subgroups corresponding to the subfields of $\mathbb{F}_{2^n}$ and so the group order has various significant factors). Hence, Koblitz curves over $\mathbb{F}_2$ are not generally at risk from the Weil descent attack.

One is not constrained to using elliptic curves over $\mathbb{F}_2$ but can use curves over any field of small characteristic (such as $\mathbb{F}_{2^2}$ or $\mathbb{F}_3$). These curves are sometimes known as 'Koblitz curves' and sometimes as 'subfield curves'. Nevertheless, the case of curves over $\mathbb{F}_2$ remains the most important in applications.

More recently the definition of Koblitz curves has been extended by Gallant, Lambert and Vanstone [14] to the case of elliptic curves over prime fields $\mathbb{F}_p$ which have convenient endomorphisms. The speedup for curves over $\mathbb{F}_2$ can be realised in this case too by using endomorphisms.

We will now discuss Koblitz curves in more detail. We separate the discussion into two parts. First we discuss the more traditional Koblitz curves (those over small fields, and in particular $\mathbb{F}_2$) and second we discuss Koblitz curves over large prime fields.

4.1. **Koblitz curves in characteristic 2.** The SEC standard [2] gives 20 predefined curves in characteristic two, a number of which appear in other standards such as ANSI X9.62, WAP WTLS or NIST FIPS 186.2. Of these 20 predefined curves six are of Koblitz form in that they possess a convenient endomorphism which can be used to speed up the group law.

The curves, labelled sect163k1, sect233k1, sect239k1, sect283k1, sect409k1 and sect571k1 are all anomalous binary curves of the form

$$Y^2 + XY = X^3 + aX^2 + 1$$

where $a \in \{0, 1\}$. These curves possess the endomorphism given by the action of the Frobenius map

$$(x, y) \longmapsto (x^2, y^2).$$

Using techniques of Solinas [39] one can improve the algorithms for point multiplication considerably, and hence obtain very efficient implementations both in hardware and software.

However, the existence of the Frobenius endomorphism of order n combined with the techniques of Section 3.1 mean that the curves are not as secure as a general curve over the same finite field. However, the effect of this reduction in security is modest. For example with the curve sect163k1 one would expect to require

$$\sqrt{\frac{q\pi}{2 \cdot h}} \approx 2^{81}$$

operations to break a general elliptic curve over $\mathbb{F}_{2^{163}}$ while the Koblitz curve only requires

$$\sqrt{\frac{q\pi}{4 \cdot 163 \cdot h}} \approx 2^{77}$$

operations. For larger finite fields the effect of choosing a Koblitz curve is similar. Table 3 demonstrates this by showing the difference between the security of a general curve and a Koblitz curve for the field sizes in the above mentioned standard, with the specified cofactor. For the security of general curves in the table we assume the cofactor is two, as this is the most common case for randomly chosen curves.

To summarise the results of this section. Despite being anomalous, Koblitz curves are not susceptible to the anomalous curves attack (since $p = 2$). Despite being over a field of the form $\mathbb{F}_{2^m}$, Koblitz curves are not at risk from Weil descent since the extension degree m is prime. Nevertheless, there is a slight loss of security from the use of equivalence classes in the parallel Pollard methods.

Table 3. Reduction in Security for Koblitz in Characteristic Two

Curve	Field Size	Cofactor	General Curve Security	Koblitz Curve Security
sect163k1	2^{163}	2	2^{81}	2^{77}
sect233k1	2^{233}	4	2^{116}	2^{111}
sect239k1	2^{239}	4	2^{119}	2^{114}
sect283k1	2^{283}	4	2^{141}	2^{136}
sect409k1	2^{409}	4	2^{204}	2^{198}
sect571k1	2^{571}	4	2^{285}	2^{279}

4.2. **Koblitz curves in characteristic p.** Gallant, Lambert and Vanstone [14] point out in characteristic p one can also use endomorphisms to speed up the point multiplication, as long as the curve is chosen with the correct properties. In [2] there are 15 predefined curves over fields of large prime characteristic. Of these four are not chosen at random, but are chosen to have efficiently computable endomorphisms. The curve names of these four curves are secp160k1, secp192k1, secp224k1, secp256k1.

These curves all have the form

$$Y^2 = X^3 + b$$

and possess the endomorphism

$$\phi : (x, y) \longmapsto (\beta x, y),$$

where β is a cube root of one in $\mathbb{F}_p$. The characteristic p of the base field needs to be chosen so that $p \equiv 1 \pmod 3$, unlike the other curves in the standard the field of definition of these curves is not chosen to be a generalised Mersenne prime.

Note that the performance improvement with using these curves is not as marked as the performance improvement one obtains by using anomalous binary curves. But the corresponding reduction in security is also not as marked, since the endomorphism is of order 3.

The endomorphism ring of these curves is contained in the ring of integers of the quadratic field $\mathbb{Q}(\sqrt{-3})$. We discuss whether this is could pose a security threat in Section 5.

The Pollard methods utilising equivalence classes as discussed in Sections 2.4 and 3.1 apply in this case. The equivalence classes have size 6. Table 4 demonstrates this by showing the difference between the security of a general curve and a Koblitz curve for the field sizes in the above mentioned standard. Unlike the case of characteristic two, in the standards the Koblitz curves for large prime characteristic always have minimal cofactor, i.e. $h = 1$.

Implication 9. *The only known security reduction for Koblitz curves, compared with random curves, is the use of equivalence classes to speed up the Pollard methods.*

5. Possible special attacks

In this section we indicate some mathematical structures which are present with elliptic curves but which have not yielded attacks on the ECDLP.

TABLE 4. Reduction in Security for Koblitz in Large Prime Characteristic

Curve	Field Size	General Curve Security	Koblitz Curve Security
secp160k1	2^{160}	2^{80}	2^{79}
secp192k1	2^{192}	2^{96}	2^{95}
secp224k1	2^{224}	2^{112}	2^{111}
secp256k1	2^{256}	2^{128}	2^{127}

5.1. **Endomorphisms and complex multiplication.** Elliptic curves over finite fields have a lot of mathematical structure and theory behind them. One of these structures is the *endomorphism ring*. Let E be an elliptic curve over $\mathbb{F}_q$. Then

$$\mathrm{End}_{\overline{\mathbb{F}}_q}(E) = \{\text{rational maps } \phi : E(\overline{\mathbb{F}}_q) \to E(\overline{\mathbb{F}}_q) \text{ such that } \phi(\mathcal{O}_E) = \mathcal{O}_E\}.$$

The endomorphism ring always contains all the multiplication by n maps $[n]$ for $n \in \mathbb{Z}$. It also contains the qth power Frobenius map

$$(x, y) \longmapsto (x^q, y^q).$$

We have seen some other endomorphisms earlier in this report.

It is easy to phrase the ECDLP as a problem involving the endomorphism ring: Given $P, Q \in E(\mathbb{F}_q)$ find $\phi \in \mathrm{End}_{\overline{\mathbb{F}}_q}(E)$ such that $Q = \phi(P)$. This problem also seems to be hard.

From now on we restrict to the case of non-supersingular (which are also called ordinary) elliptic curves.

Write $\#E(\mathbb{F}_q) = q + 1 - t$. Then it is known that $|t| \le 2\sqrt{q}$. Define

$$\Delta = t^2 - 4q.$$

The number Δ is negative and satisfies $\Delta \equiv 0, 1 \pmod 4$. This number is central to the theory of the endomorphism ring.

Let c be the largest positive integer such that $c^2 | \Delta$ and $\Delta/c^2 \equiv 0, 1 \pmod 4$. Let $d = \Delta/c^2$. Then the endomorphism ring is a subring of the ring $\mathcal{O}_L = \mathbb{Z}[(d + \sqrt{d})/2]$ which is the ring of integers of the imaginary quadratic number field $L = \mathbb{Q}(\sqrt{d}) = \mathbb{Q}(\sqrt{\Delta})$.

There is a natural measure of complexity of the ring $\mathcal{O}_L$ (equivalently, the field L), namely its *ideal class number*. It is well-known in algebraic number theory that fields of small class number are easier to handle than fields of large class number.

Let us say that an elliptic curve E over $\mathbb{F}_q$ has *small class number* if the class number of the corresponding field L under the above relationship is "small" (say, less than 200). We say that E has *large class number* otherwise, particularly if the class number of L is larger than 2^{80}.

The intuition from algebraic number theory has lead some researchers to speculate that the ECDLP for elliptic curves of small class number might be easier to solve than the ECDLP for elliptic curves with large class number.

We must stress at this point that the only supporting evidence for this hypothesis is the small speedup to the Pollard methods from using equivalence classes for the two curves $j = 0$ and $j = 1728$ (which both have class number 1).

Indeed, many experts have tried to find a way to attack the ECDLP using the theory of the endomorphism ring and their efforts have been totally unsuccessful.

There is **no evidence** that elliptic curves with small class number are insecure for cryptography.

5.2. **Imagined attacks.** Some attacks which have been suggested for the ECDLP on elliptic curves with small class number rely on lifting from characteristic p to characteristic zero (in particular, lifting to the appropriate ring class field associated to the endomorphism ring).

In fact, it can be argued that any kind of lifting attack to number fields will be unsuccessful (this is the thrust of the work of Koblitz and others [18] on the refutation of the so-called “Xedni-calculus attack” [38] on the ECDLP). In other words, the size of the class number does not influence the success or otherwise of these kind of lifting attacks.

5.3. **Koblitz curves revisited.** All Koblitz curves have small class number. In fact, the Koblitz curves in the SEC standard all have class number 1. Nevertheless, there have been no methods found to attack the ECDLP on such curves which utilise this structure, except for the equivalence classes method.

5.4. **Trace three curves.** A recent proposal [29] has been to use elliptic curves of “trace three” for cryptography.

This scheme uses elliptic curves E over $\mathbb{F}_p$ such that the number of points is $p-2$. Since $p-2=p+1-3$ these curves have “trace three” in the sense of their Frobenius endomorphism. Such curves can be constructed using the CM method, which produces elliptic curves with small class number.

The first questions to ask about this scheme are what is special about trace three curves, and what is their advantage? The proposers of this scheme claim that trace three curves resist the Frey-Rück attack with very high probability and so they are secure. But, as we have explained in Section 3.2, *any* randomly chosen non-supersingular curve resists the Frey-Rück attack with very high probability. And, in any case, checking whether a curve is vulnerable to this attack is completely elementary and trivial.

There seems to be no advantage of the trace three curves with respect to parameter selection, efficiency of arithmetic, ease of use, memory requirements etc. In other words, using trace three curves gives no improvement or benefit over random curves. Indeed, the authors of [29] could have just as readily written their paper about “trace five” curves or “trace -1” curves or “trace 675432345661” curves and they would have found analogous results.

There are other special families of curves (such as the Koblitz curves) which yield valuable efficiency improvements. If a user wants to restrict their attention from random curves to special ones, then they should expect some computational advantages in return. In conclusion, there seems to be no motivation for designing cryptosystems to only use trace three curves.

6. Conclusion

The security criteria for elliptic curves, so that the discrete logarithm problem is difficult, is that one should choose curves E over finite fields $K=\mathbb{F}_q$ such that if

$$\#E(K) = h \cdot l$$

where l is prime, then

- $l > 2^{160}$.

- l should not divide $q^k - 1$ for $k \leq 30$.
- $l \neq q$.
- q should be equal to a large prime, or a prime power of two.

In addition one may need to take into account a slight reduction in security if the curve has an efficiently computable endomorphism (as in the case of Koblitz curves), As we have seen, this reduction in security is not enough to cause the systems to be insecure. But care should be taken when selecting parameter sizes to ensure that one is getting the desired security.

In general it is better to take a random rather than a special curve to protect oneself against possible future attacks. But some environments may demand the use of special curves either for interoperability or reasons of constrained resources. In such situations one should use the special curves, but one should not forget that this decision carries with it a slightly reduced security confidence.

The various standards, of which SEC [1] [2] are the most complete, specify a number of recommended curves either of the random or special variety, assuming one chooses from these lists a curve which has a value of l greater than 2^{160} we see no medium term risk in using such curves in a cryptographic system.

The security of most elliptic curve cryptosystems actually relies on the difficulty of problems which are related (but not equal) to the ECDLP, such as the EC-DHP and EC-DDH. We advise that CRYPTREC should initiate further study of these important computational problems.

The ECDLP has been studied by a large number of experts over the last few years. These experts have built upon the huge body of literature on elliptic curves built up over the centuries. Despite this intensive work by some of the worlds most able cryptographers and mathematicians the ECDLP is still an exponential problem in general. We can therefore conclude with some confidence that the ECDLP is a hard enough problem upon which to base security protocols well into the current century.

References

[1] Standards for Efficient Cryptography. SEC1 : Elliptic Curve Cryptography. Version 1.0, **http://www.secg.org/**, 2000.

[2] Standards for Efficient Cryptography. SEC2 : Recommended Elliptic Curve Domain Parameters. Version 1.0, **http://www.secg.org/**, 2000.

[3] L. Adleman, J. De Marrais, and M.-D. Huang. A subexponential algorithm for discrete logarithms over the rational subgroup of the Jacobians of large genus hyperelliptic curves over finite fields. *ANTS-1: Algorithmic Number Theory*, Springer-Verlag LNCS 877, 28–40, 1994.

[4] K. Araki and T. Satoh. Fermat quotients and the polynomial time discrete logarithm algorithm for anomalous elliptic curves. *Commentarii Mathematici Universitatis Sancti Pauli*, **47**, 81–92, 1998.

[5] R. Balasubramanian and N. Koblitz, The improbability that an elliptic curve has subexponential discrete log problem under the Menezes-Okamoto-Vanstone algorithm. *J. Cryptology*, **11**, 141–145, 1998.

[6] I.F. Blake, G. Seroussi and N.P. Smart. *Elliptic Curves in Cryptography.* Cambridge University Press, 1999.

[7] I. Duursma, P. Gaudry and F. Morain, Speeding up the discrete log computation on curves with automorphisms. *Advances in Cryptology - ASIACRYPT '99*, Springer-Verlag LNCS 1716, 103–121, 1999.

[8] G. Frey. How to disguise an elliptic curve. Talk at Waterloo workshop on the ECDLP, 1998. **http://cacr.math.uwaterloo.ca/conferences/1998/ecc98/slides.html**.

[9] G. Frey and H.-G. Rück. A remark concerning m-divisibility and the discrete logarithm problem in the divisor class group of curves. *Math. Comp.*, **62**, 865–874, 1994.

[10] G. Frey, M. Müller and H.-G. Rück. The Tate pairing and the discrete logarithm applied to elliptic curve cryptosystems, *IEEE Trans. Inform. Theory*, **45**, 1717–1719, 1999.

[11] S.D. Galbraith and N.P. Smart. A cryptographic application of Weil descent. *Cryptography and Coding, 7th IMA Conference*, Springer-Verlag, LNCS 1746, 191–200, 1999.

[12] S. D. Galbraith, F. Hess and N. P. Smart. Extending the GHS Weil Descent Attack. Preprint 2001. Available from:
`http://www.cs.bris.ac.uk/~nigel/weil_descent.html`

[13] R. Gallant, R. Lambert and S. Vanstone, Improving the parallelised Pollard lambda search on binary anomalous curves. *Math. Comp.*, **69**, 1699–1705, 2000.

[14] R. P. Gallant, R. J. Lambert and S. A. Vanstone, Faster point multiplication on elliptic curves with efficient endomorphisms, *Advances in Cryptology - CRYPTO 2001*, Springer-Verlag LNCS 2139, 190–200, 2001.

[15] P. Gaudry. An algorithm for solving the discrete logarithm problem on hyperelliptic curves. *Advances in Cryptology - EUROCRYPT 2000*, Springer-Verlag LNCS 1807, 19–34, 2000.

[16] P. Gaudry, F. Hess and N.P. Smart. Constructive and destructive facets of Weil descent on elliptic curves. To appear *J. Cryptology.*

[17] R. Harley, Counting points using the AGM, Eurocrypt 2001 rump session talk.

[18] M. Jacobson, N. Koblitz, J.H. Silverman, A. Stein and E. Teske. Analysis of the Xedni Calculus Attack. *Designs, Codes and Cryptography*, **20**, 41–64 2000.

[19] M. Jacobson, A. Menezes and A. Stein. Solving elliptic curve discrete logarithm problems using Weil descent. Preprint, 2001.

[20] A. Joux and K. Nguyen. Separating Decision Diffie-Hellman from Diffie-Hellman in cryptographic groups, Preprint, 2001. Available from:
`http://www.exp-math.uni-essen.de/~nguyen/dhdp_paper.ps`

[21] N. Koblitz. Elliptic curve cryptography. *Math. Comp.*, **48**, 203–209, 1987.

[22] N. Koblitz. CM curves with good cryptographic properties, *Advances in Cryptology - CRYPTO '91*, Springer-Verlag 576, 279–287, 1992.

[23] N. Koblitz, A. Menezes and S. Vanstone. *The state of elliptic curve cryptography.* Designs, Codes and Cryptography, **19**, 173–193, 2000.

[24] A. Lenstra and E. Verheul. Selecting cryptographic keysizes. to appear in *J. Cryptology*, 2001.

[25] R. Lercier. Computing isogenies in $\mathbb{F}_{2^n}$. *Algorithmic Number Theory Symposium- ANTS II*, Springer-Verlag LNCS 1122, 197–212, 1996.

[26] M. Maurer, A. Menezes and E. Teske. Analysis of the GHS Weil descent attack on the ECDLP over characteristic two finite fields of composite degree. Preprint, 2001.

[27] A. Menezes, T. Okamoto and S. Vanstone. Reducing elliptic curve logarithms to logarithms in finite fields. *IEEE Trans. on Infor. Th.*, **39**, 1639–1646, 1993.

[28] A. Menezes and M. Qu. Analysis of the Weil descent attack of Gaudry, Hess and Smart. *Topics in Cryptology - CT-RSA 2001*, Springer-Verlag LNCS 2020, 308–318, 2001.

[29] A. Miyaji, M. Nakabayashi and S. Takano, New explicit conditions of elliptic curve traces for FR-reduction, *IEICE Trans. Fundamentals*, Vol. E84 A, No. 5, 2001.

[30] P. van Oorschot and M. Wiener. Parallel collision search with applications to hash functions and discrete logarithms. *2nd ACM Conference on Computer and Communications Security*, 210–218, ACM Press 1994.

[31] S. Pohlig and M. Hellman. An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance. *IEEE Trans. on Infor. Th.*, **24**, 106–110, 1978.

[32] J. Pollard. Monte Carlo methods for index computations mod p. *Math. Comp.*, **32**, 918–924, 1978.

[33] H.-G. Rück, On the discrete logarithm in the divisor class group of curves, *Math. Comp.*, **68**, No.226, 805–806, 1999.

[34] R. Schoof. Elliptic curves over finite fields and the computation of square roots mod p. *Math. Comp.*, **44**, 483–494, 1985.

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

[36] V. Shoup. Lower bounds for discrete logarithm and related problems. *Advances in Cryptology - EUROCRYPT '97*, Springer-Verlag LNCS 1233, 313-328, 1997.

[37] J. H. Silverman. *The Arithmetic of Elliptic Curves.* GTM 106, Springer-Verlag, 1986.

[38] J.H. Silverman. The Xedni Calculus and the Elliptic Curve Discrete Logarithm Problem. *Designs, Codes and Cryptography*, **20**, 5–40, 2000.

[39] J. Solinas. Efficient arithmetic on Koblitz curves. *Designs, Codes and Cryptography*, **19**, 195–249, 2000.

[40] N. P. Smart. The discrete logarithm problem on elliptic curves of trace one. *J. Cryptology*, **12**, 193–196, 1999.

[41] N.P. Smart. How secure are elliptic curves over composite extension fields? *Advances in Cryptology - EUROCRYPT '01*, Springer-Verlag LNCS 2045, 30–39, 2001.

[42] J.F. Voloch. The discrete logarithm problem on elliptic curves and descents. Preprint, 1997.

[43] M. J. Wiener and R. J. Zuccherato. Faster attacks on elliptic curve cryptosystems. *Selected Areas in Cryptography - SAC 1999*, Springer-Verlag LNCS 1556, 190–200, 1999.