18.704
Fall 2004

Elliptic Curve Cryptography

Contents

1 Abstract

The aim of this paper is to give a basic introduction to Elliptic Curve Cryptography (ECC). We will begin by describing some basic goals and ideas of cryptography and explaining the cryptographic usefulness of elliptic curves. We will then discuss the discrete logarithm problem for elliptic curves. We will describe in detail the Baby Step, Giant Step method and the MOV attack. The latter will require us to introduce the Weil pairing. We will then proceed to talk about cryptographic methods on elliptic curves. We begin by describing the kinds of curves for which the discrete log problem is not known to be easy. We will then introduce the basic cryptographic problem of sending and encrypted message, and describe two encryption methods: the Diffie-Hellman Key Exchange, and the Massey-Omura Encryption. We will conclude our paper by summing up the results and mentioning briefly some developments which we did not have room to address fully in this paper. [1]

2 Basics of Cryptography

Cryptography aims to analyze which problems can be made easy while making others hard.[Lu] In practical terms, if I want to send my friend a message nobody else will be able to read, I want it to be easy for me to encode, easy for my friend to decrypt, and hard for an adversary to decrypt. One problem that is thought to be hard, on which many cryptographic systems are based, is the discrete log problem. No fast methods for solving this problem in general exist, though it is possible that they will someday be found. The discrete log problem is more difficult for elliptic curves than for finite fields, which means that the same size encryption key will yield greater security if we use ECC. Put another way, we can use a smaller key to get the same amount of security, which speeds up the computations we want to speed up (i.e. mine and my friend's).

This paper will often talk about problems being practical, easy, and tractable, or, conversely, impractical, hard, and intractable. A problem is considered tractable if the computation time is polynomial in the length of the input. For example, say we are working with a group of size N. We can then write down any member of N using $\log(N)$ digits. For the problem to be considered tractable, we will then need an algorithm that runs in time that is polynomial in $\log(N)$. An algorithm that runs in time N, for

[1] When no source is cited for a specific item of information, that information generally comes from [Wa].

instance, will be much too slow. If we use 1024 bits to write down N, a perfectly reasonable number, N can equal 2^{1023}, which is huge.

3 Discrete Logarithm Problem for Elliptic Curves

3.1 Problem Statement

The classical discrete logarithm problem is the following: Given that there is some integer k such that $a^k \equiv b \pmod{p}$, where p is prime, find k. Since the order of a must divide $p-1$, k can be defined $\pmod{p-1}$.

Similarly, we can define the discrete log problem for elliptic curves. Switching to additive notation, we have the problem of finding k (given that k exists) such that $kP = Q$, where P, Q are points on the curve $E(\mathbb{F}_q)$, with $q = p^n$ for some prime p.

Our notation is the following: $E(\mathbb{F}_q)$ is the set of points on E whose coordinates lie in $\mathbb{F}_q$. $\mathbb{F}_q$ denotes $\mathbb{F}_{p^n}$. We will write $E(\mathbb{F}_q)$ with coefficients in $\mathbb{F}_q$. kP is defined as $\underbrace{P + P + \ldots + P}_{k}$, with standard addition of points on elliptic curves.

3.2 Attacks on the Elliptic Curve Discrete Logarithm Problem

In cryptography, an attack is a method of solving a problem. Specifically, the aim of an attack is to find a fast method of solving a problem on which an encryption algorithm depends. The known methods of attack on the elliptic curve (EC) discrete log problem that work for all curves are slow, making encryption based on this problem practical. However, several efficient methods for solving the EC discrete log problem for specific types of elliptic curves are known. This means that one should make sure that the curve one chooses for one's encoding does not fall into one of the several classes of curves on which the problem is tractable.

Below, we describe the Baby Step, Giant Step Method, which works for all curves, but is slow. We then describe the MOV attack, which is fast for certain types of curves.

3.2.1 Baby Step, Giant Step Method

This is one of the fastest general methods of solving the EC discrete log problem. (In fact, it can be applied to an arbitrary group.) The algorithm runs in approximately $\sqrt{N}$ time and $\sqrt{N}$ space, where $N = \#E(\mathbb{F}_q)$. This is not fast enough to be practical.

Problem:

Find k such that $kP = Q$ on $E(\mathbb{F}_q)$, with $\#E(\mathbb{F}_q) = N$, assuming that such a k exists.

Algorithm:

1. Pick an integer $m > \sqrt{N}$.
2. Compute mP.
3. For $i = 0$ to $i = m - 1$ compute (and store) iP.
4. For $j = 0$ to $j = m - 1$ compute (and store) $Q - jmP$.
5. Sort the lists from steps 3, 4 in some consistent way.
6. Compare the lists from steps 3, 4 until a pair i, j such that $iP = Q - jmP$ is found.
7. Return $k \equiv i + jm \pmod{N}$.

Proof:

Since we chose m such that $m^2 > N$, there is a solution $k < m^2$. Let $k_0 \equiv k \pmod{m}$, $0 \leq k_0 < m$. Let $k_1 = (k - k_0)/m$. Then we have $k = k_0 + mk_1$, with $0 \leq k_1 < m$. In the algorithm given, we try all i in the range of values of k_0 and all j in the range for k_1 until we find i, j such that $iP = Q - jmP \implies (i + jm)P = Q$. Thus the value returned is always a solution, and the algorithm always halts, since k_0, k_1 must exist.

Time Analysis:

Step 1 takes $O(\log N)$ time. Steps 2 and 3 take $O(m+1) = O(\sqrt{N})$ and $\Omega(m + 1) = \Omega(\sqrt{N})$ time. Step 4 can be done in $O(\sqrt{N})$ time as well. The sort in step 5 can be performed in $O(\log(N)\sqrt{N})$ time. Step 6 can then be done in $O(\sqrt{N})$ time. Finally, step 7 can also be done in $O(\sqrt{N})$, so if we ignore logarithmic factors ($\sqrt{N}$ is already large enough for the problem to be intractable), we find that the running time is on the order of $\sqrt{N}$. The storage space required is also on the order of $\sqrt{N}$, as that is how much space is required to store the lists in steps 3 and 4. This algorithm is too slow to be of practical use in breaking codes, as it is exponential in the length $\log N$ of the input.

3.2.2 The MOV Attack

The MOV attack (named for Menezes, Okamoto, and Vanstone) reduces the discrete log problem on an elliptic curve $E(\mathbb{F}_q)$ to the discrete log problem in $\mathbb{F}_{q^m}^\times$ for some m. The problem can then be solved fairly quickly using an index calculus attack (not described in this paper), as long as m is small. A small m can always be obtained for certain types or curves.

We will start by defining a Weil pairing for curves $E(\mathbb{F})$. We will not actually prove that such a pairing exists for the curves we are considering.

Consider $E(\mathbb{F})$, and let N be an integer not divisible by the characteristic[2] of $\mathbb{F}$. Let $E[N]$ be the set of points on the curve with order dividing N whose coordinates are in the algebraic closure[3] of $\mathbb{F}$. We claim (without giving proof) that $E[N] \simeq \mathbb{Z}_n \oplus \mathbb{Z}_n$.

Let $\mu_N = \{x \in \mathbb{F} | x^N = 1\}$. That is, μ_N is the group of Nth roots of unity in $\mathbb{F}$. Since the characteristic of $\mathbb{F}$ doesn't divide N, $x^N = 1$ has no multiple roots, and therefore there are N distinct roots of unity in $\mathbb{F}$.

Definition 1: A *Weil pairing* is a map

$$e_N : E[N] \times E[N] \to \mu_n$$

such that

1. e_N is bilinear in each variable.
2. e_N is nondegenerate in each variable. That is, if $e_N(S,T) = 1$ for all S, then $T = \mathcal{O}$, and if $e_N(S,T) = 1$ for all T, then $S = \mathcal{O}$. [4]
3. $e_N(T,T) = 1 \forall T$.
4. $e_N(T,S) = e_N(S,T)^{-1} \forall S,T$.
5. If σ is an automorphism of $\bar{\mathbb{F}}$ that preserves the coefficients of E, then $e_N(\sigma S, \sigma T) = \sigma(e_N(S,T)) \forall S,T$.
6. If α is a separable endomorphism of E, $e_N(\alpha(S), \alpha(T)) = e_N(S,T)^{\deg \alpha}$.

[2] The *characteristic* of a field $\mathbb{F}$ is the smallest positive integer p such that $p \times 1_{\mathbb{F}} = 0$ if such a p exits, and 0 otherwise.[Du]

[3] The *algebraic closure* of a field $\mathbb{F}$ is a field $\bar{\mathbb{F}}$ such that $\bar{\mathbb{F}}$ is algebraic over $\mathbb{F}$ and every polynomial over $\mathbb{F}$ splits completely over $\bar{\mathbb{F}}$.[Du]

[4] $\mathcal{O}$ denotes the point at infinity on our curve $E(\mathbb{F})$.

The following propositions will be useful:

Proposition 1: Let S, T form a basis for $E[N]$. Then $e_N(S,T)$ is a primitive Nth root of unity.

Proof:
Let $\zeta = e_N(S,T)$, and let d be such that $\zeta^d = 1$. Then by bilinearity $e_N(S, dT) = e_N(S,T)^d = \zeta^d = 1$. Similarly, $e_N(T, dT) = e_N(T,T)^d = 1^d = 1$ by the properties 1 and 3 of e_N. For any point $P \in E[N]$, $P = aS + bT$ for some $a, b \in \mathbb{Z}$. We therefore have $e_N(P, dT) = e_N(aS + bT, dT) = e_N(S, dT)^a e_N(T, dT)^b = 1$. Since this holds for all points P, by property 2 we have $dT = \mathcal{O}$. Thus $\text{ord}(T)|d \Rightarrow n|d$, and therefore $\zeta = e_N(S,T)$ is always a primitive nth root of unity when S, T form a basis.

Proposition 2: If $E[N] \subseteq E(\mathbb{F})$, then $\mu_N \subset \mathbb{F}$.

Let σ be an automorphism of $\bar{\mathbb{F}}$ such that σ is the identity map on $\mathbb{F}$. Let S, T form a basis for $E[N]$. Since S, T have coefficients in $\mathbb{F}$, by property 5 of the Weil pairing, we have

$$\zeta = e_N(S,T) = e_N(\sigma S, \sigma T) = \sigma(e_N(S,T)) = \sigma(\zeta).$$

By the fundamental theorem of Galois theory, $\sigma(\zeta) = \zeta \Rightarrow \zeta \in \mathbb{F}$. By Proposition 1, ζ is a primitive root of unity. Since the above holds for all primitive roots of unity ζ, we have $\mu_N \subset \mathbb{F}$.

We can now describe the MOV attack. Since from algebra we know that $\bar{\mathbb{F}_q} = \bigcup_{i \geq 1} \mathbb{F}_{q^i}$, we can pick an m such that $E[N] \subseteq E(\mathbb{F}_{q^m})$. By the proposition above, we have $\mu_N \subset \mathbb{F}_{q^m}$.

Problem:
Find k such that $kP = Q$ on $E(\mathbb{F}_q)$, with $\#E(\mathbb{F}_q) = N$, assuming that such a k exists. Use a reduction of the discrete log problem on the curve $E(\mathbb{F}_q)$ to the discrete log problem in $\mathbb{F}_{q^m}^\times$.

Algorithm:
Until $\text{lcm}(d_1, d_2, \ldots, d_k) = N$, perform the following steps, incrementing i by 1 for each repetition:

1. Select a random point $S_i \in E(\mathbb{F}_{q^m})$.

2. Compute the order M_i of S_i.

3. Let $d_i = \gcd(M_i, N)$. Let $T_i = (M_i/d_i)S_i$.

4. Let $\zeta_{1i} = e_N(P, T_i), \zeta_{2i} = e_N(Q, T_i)$.

5. Solve the discrete log problem $\zeta_{1i}^{k_i} = \zeta_{2i}$ in $F_{q^m}^{\times}$. This gives $k_i \pmod{d_i}$.

Now use the values of $k_i \pmod{d_i}$ to find a $k \pmod{N}$ such that $k \equiv k_i \pmod{d_i} \forall i$. This is the k we are looking for.

Proof:

We can clearly select a point on the curve (step 1), calculate its order (step 2), and find T_i (step 3). Note that the order of T_i is $d_i|N$, so $T_i \in E[N]$. Let $\zeta = e_N(R, T_i)$, where R is an arbitrary point on $E(\mathbb{F}_{q^m})$ and T_i is as in step 3. Then

$$\zeta^d = e_N(R, T_i)^d = e_N(R, dT_i) = e_N(R, M_iS_i) = e_N(R, \mathcal{O}) = 1,$$

and so $\zeta_1, \zeta_2 \in \mu_d \subseteq F_{q^m}^{\times}$, and we can solve $\zeta_{2i} = \zeta_{1i}^{k_i}$ in $F_{q^m}^{\times}$.

Now let $kP = Q$, and define l_i such that $k \equiv l_i \pmod{d_i}$. Then $e_N(kP, T_i) = e_N(Q, T_i) \Rightarrow e_N(P, T_i)^k = e_N(Q, T_i) \Rightarrow \zeta_{1i}^k = \zeta_{2i}$. Since $\zeta_{1i}^d = 1$, this implies that $\zeta_{1i}^{l_i} \equiv \zeta_{2i} \pmod{d_i} \Rightarrow l_i \equiv k_i \pmod{d_i}$, and so k must equal $k_i \pmod{d_i}$. Thus finding the necessary k_i provides the answer k.

Time Analysis:

Weil pairings can be computed reasonably quickly, and given $k \equiv k_i \pmod{d_i}$, we can reasonably quickly find $k \pmod{N}$. Thus the running time of this algorithm will be determined by (1) how long it takes to compute each k_i, and (2) for how many i, k_i must be computed.

The time it takes to compute each k_i depends on the size of the field $F_{q^m}^{\times}$ in which it must be computed. The larger m is, the longer the computation takes. There is no general theorem that determines how large an m is necessary for all types of curves. However, we know that for curves such that $\#E(\mathbb{F}_q) = q + 1$ (these curves are called *supersingular*), if there exists a point $P \in E(\mathbb{F}_q)$ of order N, then $E[N] \subseteq E(\mathbb{F}_{q^2})$. (We omit the proof of this statement.)

To answer question (2), we first claim (without proof) that $E(\mathbb{F}_{q^m}) \simeq \mathbb{Z}_{n_1} \oplus \mathbb{Z}_{n_2}$ for some n_1, n_2 with $n_1|n_2$. The largest possible order of an element in $E(\mathbb{F}_q)$ divides the largest order of an element in $E(\mathbb{F}_{q^m})$, so $N|n_2$, since no element in $E(\mathbb{F}_{q^m})$ can have order $> n_2$.

Let B_1, B_2 be points generating $E(\mathbb{F}_{q^m})$ such that B_1, B_2 have orders n_1, n_2, respectively. Then any point S_i chosen in the algorithm above can be written as $a_1B_1 + a_2B_2$ for some a_1, a_2 depending on i. Let p be a prime, and let $p^e \| N$. Then $p^e | n_2$. If $p \nmid a_2$, then $p^e | n_2 \Rightarrow p^e | M_i$, where M_i is the order of S_i. Then $p^e | d_i = \gcd(M_i, N)$. Since S_i is picked randomly, a_2 for S_i is also random, and so the probability that $p \nmid a_2$ is $1 - 1/p$. Thus the probability that $p^e | d_i$ is $\geq 1 - 1/p$ for every i and every $p^e \| N$. This probability is sufficiently low that only a few d_i should be needed for $p^e | \text{lcm}(d_1, d_2, \ldots, d_k)$ to be true for all p. We will therefore not need to iterate the algorithm too many times.

4 Cryptographic Methods Using Elliptic Curves

In this section, we will describe two cryptographic methods based on the difficulty of the discrete log problem for elliptic curves. Many other methods are used as well, but we do not have room to give all of them here. These methods are generally also available over finite fields, but give more security per bit of data if elliptic curves are used instead.

4.1 Choosing a Curve

For each of the cryptographic methods depended on the difficulty of the EC discrete log problem, we must begin by choosing an elliptic curve that is not susceptible to the known fast attacks on the discrete log problem, such as the MOV attack described in the previous section. The curve must therefore satisfy the following restrictions:

- There exists a large prime p dividing $\#E(\mathbb{F}_q)$, so that the problem is not susceptible to the Pollard-ρ-attack. [not presented]
- $\#E(\mathbb{F}_q) \neq q$ (i.e. the curve is not anomalous). This prevents the problem from being susceptible to the Semaev-Smart-Satoh-Araki attack. [not presented]
- The order of P does not divide $q^k - 1$ for all k such that $1 \leq k \leq C$, where C is a sufficiently large constant so that it is difficult to solve the discrete logarithm problem in $\mathbb{F}_{q^C}^{\times}$. This is necessary for MOV not to generate a solution quickly. [see previous section][Bha]

There exist several methods of choosing these curves. The simplest one is to pick a curve $E(\mathbb{F}_q) : y^2 = x^3 + ax + b$ at random by selecting $a, b \in F_q$

such that $4a^3 + 27b^2 \neq 0$ if q is odd and $b \neq 0$ if q is a power of 2. We then check the conditions given above. A large fraction of the time, the conditions will be satisfied. If they are not, we try a different a, b.[Bha]

4.2 Setup for Encryption Algorithms

Consider the following problem of cryptography:

Alice wants to send Bob a message m, usually assumed to be an integer. However, she does not want the eavesdropper Eve to be able to read the message as well. Therefore, Alice uses and encryption key to encrypt the message, and sends the resulting cyphertext (rather then the plaintext) to Bob. Bob then uses a decryption key to decrypt they message. Obviously, Eve must be prevented from finding the decryption key, as otherwise she would also be able to decode the message.

There are two possibilities in this scenario. Perhaps Alice and Bob were able to communicate secretly in advance and agree on a key, and perhaps they were not. If they were, the encryption and decryption keys may be the same. If they were not, they must establish a public encryption key that allows Alice to encode the message and a different private decryption key that allows Bob to decrypt the message.

4.3 Diffie-Hellman Key Exchange

The following series of steps describes the Diffie-Hellman Key Exchange, a public key encryption system that allows Alice and Bob to set up a symmetric private key. Since the symmetric systems are generally faster than the public key systems, this is quite useful.

1. Alice and Bob publicly agree on $E(\mathbb{F}_q)$, chosen so that the discrete log problem is hard, as described above. They also agree on a point $P \in E(\mathbb{F}_q)$ of high (usually prime) order.

2. Alice chooses a secret $a \in \mathbb{Z}$, computes aP, and sends it to Bob.

3. Bob chooses a secret $b \in \mathbb{Z}$, computes bP, and sends it to Alice.

4. Alice computes $a(bP) = abP$.

5. Bob computes $b(aP) = abP$.

6. Alice and Bob now have the same point abP. They use a publicly agreed on method, such as taking the last 256 bits of the y-coordinate of the point, to extract a key.

Without performing a detailed analysis, we see that no step in the algorithm above will be intractable.

In order to obtain the key as well, Eve needs to find abP from the publicly available $P, aP, bP \in E(\mathbb{F}_q)$. This is known as the Diffie-Hellman Problem. If Eve could solve the discrete log problem on $E(\mathbb{F}_q)$, she could solve $kP = (aP)$ to obtain a, and then multiply bP by a to get abP. It is not known, however, whether Eve could compute abP in some other way that does not require solving the discrete log problem.

The Decision Diffie-Hellman Problem asks if given $P, aP, bP, Q \in E(\mathbb{F}_q)$ Eve can determine whether or not $Q = abP$. As it turns out, the Weil pairing can be used to answer this question for some types of elliptic curves.

4.4 Massey-Omura Encryption

Now consider the situation in which Alice wants to send Bob a message Eve will be unable to read. Alice and Bob have not communicated privately to set up a key. Conceptually, the following algorithm will work: Alice sends Bob a box with her lock on it. Bob adds his own lock and sends the box back. Alice removes her lock and sends the box on to Bob. Bob removes his lock and reads the message. This method can be implemented using elliptic curves:

1. Alice and Bob publicly agree on $E(\mathbb{F}_q)$, chosen so that the discrete log problem is hard, as described above. Let $N = \#E(\mathbb{F}_q)$.

2. Alice represents her message as a point $P \in E(\mathbb{F}_q)$.

3. Alice chooses a secret $a \in \mathbb{Z}$ such that $\gcd(a, N) = 1$, computes aP, and sends it to Bob.

4. Bob chooses a secret $b \in \mathbb{Z}$ such that $\gcd(b, N) = 1$, computes $b(aP) = baP$, and sends it to Alice.

5. Alice finds $a^{-1} \in \mathbb{Z}_n$, computes $a^{-1}(baP) = a^{-1}baP$, and sends it to Bob.

6. Bob finds $b^{-1} \in \mathbb{Z}_n$, computes $b^{-1}(a^{-1}baP) = b^{-1}a^{-1}baP$, and takes the result to be the message.

Without performing a detailed analysis, we see that no step given above will take ‘too long.’

To show that the above encryption method is valid, we need to show that (1) Alice can represent her message as a point on $E(\mathbb{F}_q)$ and (2) that $b^{-1}a^{-1}baP = P$. It is clear that all the other steps can be performed.

We will give a method for encoding a message m as a point P on a curve $E(\mathbb{F}_p) : y^2 = x^3 + Ax + B$ (given in Weierstrass normal form). A similar method exists for $E(\mathbb{F}_q)$. As always, we assume that the message is an integer.

Let m be a message such that $0 \leq m < p/100$. For $i = 0$ to 99 let $x_i = 100m + i$. Compute each $s_i = x_i^3 + Ax_i + B$ for i in the range. It is possible to test whether s_i is a square and compute its square root if it is. If s_i is a square, we're done, as we can use the point $P = (x_i, y_i)$ on our curve, where y_i is the root of s_i. The message m can then be obtained from P by simply taking $\lfloor x_i \rfloor$. s_i is an essentially random element of $\mathbb{F}_p^{\times}$, which is cyclic and has even order (pick an odd p), so the probability of each s_i being a square is about $1/2$. Therefore, the probability that some s_i is a square is $1 - 2^{-100}$, which is quite high. We could obviously have used 10^k for some $k > 2$ in place of 100 to increase this probability, but that is unnecessary. If no s_i is a square, pick another curve.

To show that $b^{-1}a^{-1}baP = P$, it is enough to show that $a^{-1}aR = R$ for $R \in E(\mathbb{F}_q)$, as the a's and b's commute and are symmetric.

Note that we chose a such that $\gcd(a, N) = 1$, so a^{-1} exists. $a^{-1}a \equiv 1 \pmod{N}$ by the definition of a^{-1}, so $a^{-1}a = 1 + kN$ for some k. Since the group $E(\mathbb{F}_q)$ has order N, $R \in E(\mathbb{F}_q)$ has order dividing N, and therefore $NR = \mathcal{O}$. Thus $a^{-1}aR = (1 + kN)R = R + k(NR) = R + k\mathcal{O} = R$, as needed.

Let $a' = a^{-1}, b' = b^{-1}, P' = abP$. The eavesdropper Eve then knows $P, a'P, b'P$ and needs to find $a'b'P$, which is again the Diffie-Hellman problem, as in the previous encryption method.

5 Conclusion

We have now given a basic introduction to Elliptic Curve Cryptography. We introduced the discrete log problem. We then gave a general, but slow method of attack on this problem. There exist methods that take constant rather than $\sqrt{N}$ space, but there are no know general methods that run faster than $\sqrt{N}$ time. This means that the EC discrete log problem is hard. It is known to be easy only for a few specific classes of elliptic curves, such as supersingular curves (due to the MOV attack). When using an EC for encryption, it is easy to pick one that does not fall into any of these classes.

Two basic encryption methods were presented in this paper. Along the way, we defined a Weil pairing, which is very useful in ECC. In addition to the results given above for which it is relevant, there is also an encryption method based on Weil pairings. We have shown that ECC is a useful and theoretically interesting field.

References

[Bha] Bhandari, A. K.; Nagraj, D.S.; Ramkrishna, B.; Venkataramana, T. N. (editors). *Elliptic Curves, Modular Forms and Cryptography.* New Delhi, India: Hindustan Book Agency, 2003.

[Du] Dummit, David S. and Foote, Richard M. *Abstract Algebra.* New York, NY: John Wiley and Sons, Inc., 1999.

[Ga] Galbraith, Steven. *Elliptic Curve Cryptography According to Steven Galbraith.* .

[Ko] Koblitz, Neal. *A Course in Number Theory and Cryptography.* New York, NY: Springer-Verlag, 1994.

[Lu] Luhrs, Christopher. *Personal communication.* 12/10/2004.

[Wa] Washingtion, Lawrence C. *Elliptic Curves: Number Theory and Cryptography.* Boca Raton, FL: CRC Press, 2003.

[We] Weisstein, Eric. *MathWorld.* 12/9/2004.
`http://mathworld.wolfram.com`.