Chapter 5.3

The elliptic curve discrete logarithm problem (ECDLP)

Recall the DLP is based on solving the congruence

$$h \equiv g^x \pmod{p} \quad \text{in } \mathbb{F}_p^*$$

How can we do something similar with an Elliptic curve E over $\mathbb{F}_p$?

In order to do this, we can take a group of points $E(\mathbb{F}_p)$ of an elliptic curve over a finite field $\mathbb{F}_p$, and publish two points P & Q in $E(\mathbb{F}_p)$, and our secret is an integer n that makes

$$Q = \underbrace{P + P + P + \dots + P}_{n \text{ additions on } E} = \underline{n}P$$

An attacker must figure out how many times P is added to itself in order to get Q.
Remember that the addition law is actually quite complicated compared to traditional addition.

Formal Definition

Let E be an elliptic curve over the Finite Field $\mathbb{F}_p$, and let P & Q be points in $E(\mathbb{F}_p)$. The ECDLP is the problem of finding an integer n such that $Q = nP$. we denote this integer n by

$$n = \log_P(Q)$$

The elliptic discrete log of Q w.r.t P

Ex. $E: Y^2 = X^3 + 8X + 7$ over $\mathbb{F}_{73}$

The points $P = (32, 53)$ and $Q = (39, 17)$ are both in $E(\mathbb{F}_{73})$, and

$$Q = 11 \cdot P \quad \text{so} \quad \log_P Q = 11$$

Similarly, $R = (35, 47) \in E(\mathbb{F}_{73})$ and $S = (58, 4) \in E(\mathbb{F}_{73})$

So, $R = 37P$ and $S = 28P$, $\therefore$

$$\log_P(R) = 37 \quad \text{and} \quad \log_P(S) = 28$$

$\# E(\mathbb{F}_{73}) = 82$, but P satisfies $41\mathcal{O}$ Thus P has order $41 = 82/2$ so only half the points in $E(\mathbb{F}_{73})$ are multiples of P

5.31 The Double and Add Algorithm

- Quite difficult to recover the value of n from the two points P and $Q = nP$ in $E(\mathbb{F}_p)$
- In order to use the function
$$\mathbb{Z} \longrightarrow E(\mathbb{F}_p), \quad n \longrightarrow nP$$
we need to efficiently compute nP from the known values n and P

The Algorithm

Input. Point $P \in E(\mathbb{F}_p)$ and integer $n \geq 1$

1. set $Q = P$ and $R = \mathcal{O}$
2. Loop while $n > 0$
   3. If $n \equiv 1 \pmod 2$ set $R = R + Q$
   4. set $Q = 2Q$ and $n = \lfloor n/2 \rfloor$
   5. If $n > 0$, continue with loop @ step 2
6. Return point R, which equals nP

Ex. Use the Double and Add Algorithm to compute nP in $E(\mathbb{F}_p)$ for

$n = 947$, $E: Y^2 = X^2 + 14X + 19$, $p = 3623$, $P = (6, 730)$

The binary expansion of n is

$$n = 947 = 1 + 2 + 2^4 + 2^5 + 2^7 + 2^8 + 2^9$$

The final result is $947P = (3492, 60)$

We can also use a slightly different technique in order to reduce the time required to compute nP. The idea is to write n using sums and differences of powers of 2. Doing so there are generally fewer terms. Remember that performing a subtraction is as easy as adding them, since $-(x,y) = (x,-y)$

Proposition 5.18: Let n be a positive integer and let $k = [\log n] + 1$ which means that $2^k > n$. Then we can always write

$$n = u_0 + u_1 \cdot 2 + u_2 \cdot 4 + u_3 \cdot 8 + \dots + u_k \cdot 2^k$$

with $u_0, u_1, \dots, u_k \in \{-1, 0, 1\}$ and at most $\frac{1}{2}k$ of the u_i nonzero

5.32 How hard is the ECDLP?

In order to solve $Q = nP$, an attacker chooses random integers $j_1, \dots, j_r$ and $k_1, \dots, k_r$ b/t 1 and p and makes two lists of points:

List 1: $j_1P, j_2P, j_3P, \dots, j_rP$

List 2: $k_1P + Q, k_2P + Q, \dots$

As soon as she finds a match (collision) between the two lists, she is done, since if she finds $j_u P = k_v P + Q$, then $Q = (j_u - k_v) P$

However, the ECDLP appears to be much more difficult than the DLP

The fastest known algorithm to solve ECDLP in $E(\mathbb{F}_p)$ takes approx $\sqrt{p}$ steps

5.4 Elliptic Curve cryptography

5.4.1 Elliptic Diffie-Hellman key exchange

Alice and Bob agree to use a particular elliptic curve $E(\mathbb{F}_p)$ and a particular point $P \in E(\mathbb{F}_p)$. Alice chooses a secret integer n_A and bob chooses a secret integer n_B

They compute the associated multiples

$$\underbrace{Q_A = n_A P}_{\text{Alice computes}} \quad \text{and} \quad \underbrace{Q_B = n_B P}_{\text{Bob computes}}$$

and they exchange the values of Q_A and Q_B. Alice then uses her secrect multiplier to compute $n_A Q_B$, and Bob similarly computes $n_B Q_A$. They now have the showed secrect value

$$n_A Q_B = (n_A n_B) P = n_B Q_A$$

which they can use as a key to communicate privately via a symmetric cipher.

Ex. Alice and Bob ~~er~~ use the following prime, curve, and point:

$$p = 3851, \quad E: Y^2 = X^3 + 324X + 1287, \quad P = (920, 303) \in E(\mathbb{F}_{3851})$$

Alice & Bob choose their respective secret values $n_A = 1194$ and $n_B = 1759$, and then

Alice computes $Q_A = 1194P = (2097, 2178) \in E(\mathbb{F}_{3851})$
Bob computes $Q_B = 1759P = (\cancel{38}\,3684, 3125) \in E(\mathbb{F}_{3851})$

Alice sends Q_A to Bob and Bob sends Q_B to Alice.

Finally,
Alice computes $n_A Q_B = 1194(3684, 3125) = (3347, 1242)$
Bob computes $n_B Q_A = 1759(2097, 2178) = (3347, 1242) \in E(\mathbb{F}_{3851})$

They have now ~~to~~ exchanged their secret point $(3347, 1242)$, and discard the y-coordinate and treat only the value $x = 3347$ as a secret shared value

One way for Eve to discover Alice and Bob's secret is to solve the ECDLP

$$nP = Q_A$$

since if Eve can solve this problem, then she knows n_A and can use it to compute $n_A Q_B$

Definition Let $E(\mathbb{F}_p)$ be an elliptic curve over a finite field and let $P \in E(\mathbb{F}_p)$. The Elliptic Curve Diffie-Hellman Problem is the problem of computing the value of $n_1 n_2 P$ from the known values of $n_1 P$ and $n_2 P$

Remark: A point $Q \in E(\mathbb{F}_p)$ consists of two coordinates $Q = (x_Q, y_Q)$, where x_Q and y_Q are elements of the finite field $\mathbb{F}_p$, thus it appears that Alice must send Bob two numbers in $\mathbb{F}_p$. However those two numbers modulo p do not contain as much information as two arbitrary numbers, since they are related by the formula

$$y_Q^2 = x_Q^3 + Ax_Q + B \in \mathbb{F}_p$$

Note that Eve knows A and B, so if she can guess the correct value of x_Q, she can find the value of y_Q since there are only two possible values.

Thus, there is little reason for Alice to send both coordinates of Q_A to Bob, since the y coord. contains such little info.

So Bob, with the x value, ends up computing ~~either~~

$$\pm n_B Q_A = \pm(n_A n_B)P$$

and Alice computes one of

$$\pm(n_A n_B)P$$

They still end up using the same x-coordinate, since it is the same regardless of which y they choose.

5.4.2 Elliptic ElGamal PKC (public key cryptosystem)

Public Parameter Creation	
A trusted party chooses and publishes a (large) prime p, an elliptic curve E over $\mathbb{F}_p$, and a point P in $E(\mathbb{F}_p)$	
Alice	Bob
Key Creation	
Chooses a private key n_A. Computes $Q_A = n_A P$ in $E(\mathbb{F}_p)$. Publishes public key Q_A	
Encryption	
	Chooses plaintext $M \in E(\mathbb{F}_p)$. Chooses an ephemeral key k. Uses Alices public key Q_A to compute $C_1 = kP \in E(\mathbb{F}_p)$ and $C_2 = M + kQ_A \in E(\mathbb{F}_p)$. Sends ciphertext (C_1, C_2) to Alice
Decryption	
Computes $C_2 - n_A C_1 \in E(\mathbb{F}_p)$. This quantity is equal to M.	

practical difficulties with elliptic ElGamel cryptosystem:

1) There is no way to attach plaintext messages to points in $E(\mathbb{F}_p)$
2) The Elliptic ElGamel has a 4-to-1 message expansion, compared to the 2-1 expansion ratio using $\mathbb{F}_p$