

Recap: Point addition on elliptic curve

[Washington Chapter 2.6]

Suppose that

$$P = (x_1, y_1) \quad Q = (x_2, y_2)$$

$$m = (y_2 \oplus \neg y_1) \otimes \boxed{1} \otimes (x_2 \oplus \neg x_1)$$

$$c = y_2 \oplus \neg m \otimes x_1$$

$$x_3 = m \otimes m \oplus \neg x_1 \oplus x_2$$

$$y_3 = \neg m \otimes x_3 \oplus \neg c$$

computation time
for field inversion $O(\log^3 p)$
Bottleneck of point addition

$$P \oplus Q = (x_3, y_3)$$

Jacobian Coordinate

$$M = y_2 \oplus \neg y_1 \quad Z_3 = x_2 \oplus \neg x_1$$

$$m = M \otimes 1/Z_3$$

$$c = y_1 \oplus \neg m \otimes x_1 = y_1 \oplus \neg M \otimes 1/Z_3 \otimes x_1$$

$$= 1/Z_3 \otimes [y_1 \otimes Z_3 \oplus \neg M \otimes x_1]$$

$$c = 1/Z_3 \otimes C$$

$$x_3 = m \otimes m \oplus \neg x_1 \oplus \neg x_2$$

$$= \underbrace{M \otimes M \otimes 1/Z_3 \otimes 1/Z_3}_{m^2 \otimes 1/Z_3^2} \oplus \neg x_1 \oplus \neg x_2$$

$$= 1/Z_3^2 \otimes [M^2 \oplus \neg x_1 \otimes Z_3^2 \oplus \neg x_2 \otimes Z_3^2]$$

$$= X \otimes 1/Z_3^2$$

$$y_3 = \neg m \otimes x_3 \oplus \neg c$$

$$= \neg M \otimes 1/Z_3 \otimes X \otimes 1/Z_3^2 \oplus 1/Z_3 \otimes C$$

$$= 1/Z_3^3 \otimes [\neg M \otimes X \oplus C \otimes Z_3^2] = Y \otimes 1/Z_3^3$$

$$M = y_2 \oplus \neg y_1, \quad Z_3 = x_2 \oplus \neg x_1$$

$$C = y_1 \otimes Z_3 \oplus \neg M \otimes x_1$$

$$X_3 = M^2 \oplus \neg x_1 \otimes Z_3^2 \oplus \neg x_2 \otimes Z_3^2$$

$$Y_3 = \neg M \otimes X_3 \oplus C \otimes Z_3^2$$

The bottleneck operation is \otimes
which takes $O(\log p + \log \log p)$.

$$(x_1, y_1) \oplus (x_2, y_2) = (X_3, Y_3, Z_3) = (x_3, y_3)$$

When $x_3 = X_3^1 / Z_3^2$
 $y_3 = Y_3^1 / Z_3^3$

still need inversion which takes $O(\log^3 p)$ anyway 😞

Scalar Multiplication

Calculate 16P

$P \oplus P = 2P$

$2P \oplus 2P = 4P$

$4P \oplus 4P = 8P$

$8P \oplus 8P = 16P$

$(x_1, y_1) \oplus (x_1, y_1) = (X_2, Y_2, Z_2)$

$(X_2, Y_2, Z_2) \oplus (X_2, Y_2, Z_2) = (X_4, Y_4, Z_4)$

$(X_4, Y_4, Z_4) \oplus (X_4, Y_4, Z_4) = (X_8, Y_8, Z_8)$

$(X_8, Y_8, Z_8) \oplus (X_8, Y_8, Z_8) = (X_{16}, Y_{16}, Z_{16})$

$4 \times O(\log_p \log \log p)$

If we need m point additions,

computation time $\approx m \times O(\log_p \log \log p) + O(\log^3 p)$

$(X_{16}^{(9)} / Z_{16}^2, Y_{16}^{(9)} / Z_{16}^2)$
 $\uparrow O(\log^3 p)$
 lazy division.

In the previous algorithm, $m = O(\log^2 p)$

computation time $= O(\log^2 p) O(\log_p \log \log p) + O(\log^3 p)$

$= O(\log^3 p \log \log p)$

Double-based Number Representation (DBNS)

$57 = 2^5 + 2^4 + 2^3 + 2^2$ (binary representation)
 $O(\log p)$ terms

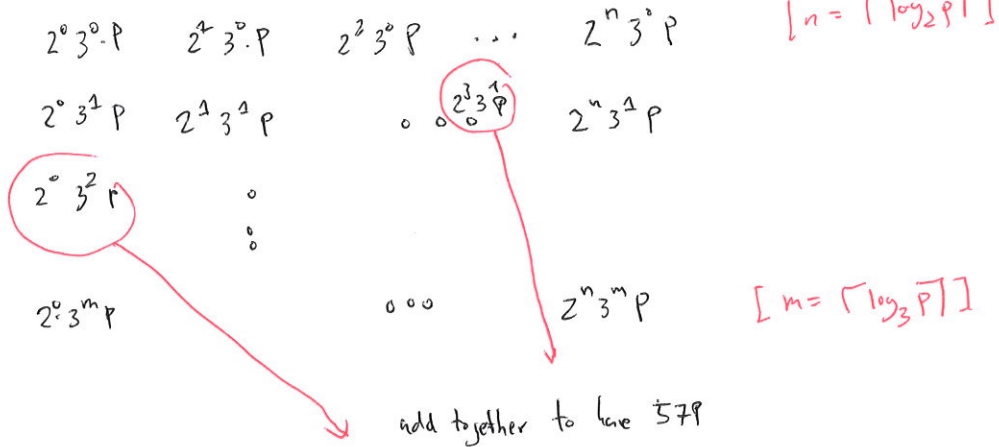
$57 = 2 \cdot 3^0 + 3^1 + 3^0$ (ternary representation)
 $O(\log p)$ terms

merge

$57 = 2^3 \cdot 3^1 + 3^2 = 48 + 9$ [Double-based Number representation]

Situation: P is fixed for many scalar multiplications, but a changed.
 \hookrightarrow scalar

Pre-computation



How to find DBNS?

- We want to find a representation with smallest # terms \rightarrow smallest # additions.

Greedy algorithm Suppose that we want to calculate $aP = Q$

1: $Q \leftarrow \infty$

2: Find the largest $2^i z^j$ ^{no larger} ~~smaller~~ than a [i and j are integers]

3: $Q \leftarrow Q \oplus 2^i z^j P$

4: $a \leftarrow a - 2^i z^j$

5: If $a=0$, then terminate.

Otherwise, go to Step 2

Theorem # terms = # point additions = $O(\log p / \log \log p)$

Computation time = $O(\log p / \log \log p) \cdot O(\log p \cdot \log \log p) + O(\log^3 p)$
 $= O(\log^3 p)$

Theorem # terms = $\Omega(\log p / \log \log p)$ for any algorithm for DBNS.

Public Key Cryptography

[Washington, Chapter 6.9]

Alice

Certificate Authority (CA)

Bob

1: Receive his private key pr_B for (CA)

Have to be done every time Alice want to send a new message.
 Inefficient!!

- 2. Receive Bob's public key pu_B from CA
- 3. Encrypt message using pu_B
 $M \rightarrow E_{pu_B}(M)$

Before communication
 After communication

$$E_{pu_B}(M) \rightarrow D_{pr_B}(E_{pu_B}(M)) = M$$

4: Decrypt message using pr_B

Identity-based Cryptography

Certificate Authority

Bob

1: Receive his private key pr_B from CA

Before communication

- 2. Receive via $pu_B = \text{Bob's email address}$ to generate $E_{pu_B}(M)$

3: Decrypt using pr_B

Cryptographic Hash Function: $H: \text{same set} \rightarrow \text{random bits length } n.$

It should be hard to find a and b such that $H(a) = H(b).$

Given b , It is hard to find a such that $H(a) = b.$

Most common: MD5, SHA-1

Recommended: SHA-3

Field Extension Prime field on $\mathbb{F}_7 = \{0, 1, \dots, 6\}$

Irreducible polynomial: ~~$x^2 + x + 2$~~ $x^2 + x + 2 \in \mathbb{F}_7[x]$

There is no solution to this equation. :-)

x	$x^2 + x + 2$
0	
1	
2	
3	
4	
5	
6	

Bonus Question
Fill this table.

Include α ($\sqrt{-5}$) to \mathbb{F}_7 !

$$\mathbb{F}_{7^2} = \{a + b\alpha : a, b \in \mathbb{F}_7\}$$

Operation: $(3 + 6\alpha) \oplus (6 + 5\alpha)$
 $= (3 \oplus 6) + (5 \oplus 6)\alpha = 2 + 4\alpha$

$$(3 + 6\alpha) \otimes (6 + 5\alpha) = (6 \otimes 5)\alpha^2 + (3 \otimes 6 \oplus 5 \otimes 3)\alpha + (3 \otimes 3) = 2\alpha^2 \oplus 2\alpha \oplus 4 = 2\alpha$$

Theorem \mathbb{F}_{7^2} with the operation defined above is a field.

Weil Pairing A function e with following properties: [output of e is \mathbb{F}_{p^2}]

1. $e(S_1 \oplus S_2, T) = e(S_1, T) \otimes e(S_2, T)$

$e(S, T_1 \oplus T_2) = e(S, T_1) \otimes e(S, T_2)$

$[e(nS, T) = e(S, T)^n]$

$[e(S, nT) = e(S, T)^n]$

2. $e(S, T) = 1$ for all T only if $S = \infty$

$e(S, T) = 1$ for all S only if $T = \infty$

3. ~~$e(T, T) = 1$ for all T~~ 3. $e(T, T) \neq 1$ for all T .

4. ~~$e(S, T)$ can be computed efficiently.~~

5. We cannot calculate $e(\sum a_i T_i)^{abc}$ from T, aT, bT, cT

Pairing Friendly Curve: $y^2 = x^3 + 1$ when $p = 6l - 1$ for some prime l

Warning: This curve is not suitable for Diffie-Hellman Key Exchange Protocol.

Certificate Authority
 Publish P_s, P_{pub} to everyone
 when $P_{pub} = sP$
 s secret of CA

Bob
 1. Receive $pr_B = [s \cdot H_2(\text{Bob's email})]$
common Hash function give a point on elliptic curve
calculate at CA

Before communication

communication

Alice
 2: Calculate $g = e(H_2(\text{Bob's email}), P_{pub})$
common function
 3. Calculate rP when r is a random number
 4: Send rP and $M \oplus H_2(g^r)$
common Hash function
bit-wise xor

5: Receive (u, v)
 6: Calculate $h = e(pr_B, u)$
 7: $M = v \oplus H_2(h)$

$$h = e(pr_B, u) = e(s \cdot H_2(\text{Bob's}), rP) = e(H_2(\text{Bob's}), P)^{sr} = e(H_2(\text{Bob's}), sP)^r = e(H_2(\text{Bob's}), P_{pub})^r = g^r !$$

$g^r =$

$$h = g^r$$

$$H_2(h) = H_2(g^r)$$

$$v \oplus H_2(h) = M \oplus H_2(g^r) \oplus H_2(h) = M \oplus H_2(h) \oplus H_2(h) = M$$