CryptoCurrency and Blockchain (4)

金融科技導論

陳君明

jmchen@crypto.tw

Threshold Cryptography
Multi-Party Computation

Introduction to Threshold Signature

- Goal : (with parameter (t, n))
  - A group of n people wants to collectively sign a message
  - Each member can create his signature
  - Any one can calculate the signature of the group upon receiving any t signatures of the n members

- Before dealing with signatures, we deal with secrets

Introduction to Threshold Secret Sharing

- Goal : (with parameter (t, n))
  - A group of n people wants to collectively own a group secret
  - Each member owns his share of secret
  - Any one can calculate the group secret upon knowing any t secret shares

- This can be done by polynomial interpolation

Lagrange Interpolation

- Problem: Construct a quadratic polynomial $p(x)$ with
  $$p(1) = 5,\ p(2) = 9,\ \text{and } p(3) = 7.$$

- Solution: $p(x)$
  $$= 5 \cdot \frac{(x-2)(x-3)}{(1-2)(1-3)} + 9 \cdot \frac{(x-1)(x-3)}{(2-1)(2-3)} + 7 \cdot \frac{(x-1)(x-2)}{(3-1)(3-2)}$$
  $$= -3x^2 + 13x - 5$$

Lagrange Interpolation

- Lagrange Interpolation Formula

$$p(x) = \sum_{i=0}^{k} p_i(x) = \sum_{i=0}^{k} y_i \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}$$

  is the unique polynomial of degree $\leq k$ passing through the $k+1$ points (x_i, y_i), where $x_i \neq x_j$ for $i \neq j$

- Note that $p(x_i) = y_i$ since $p_i(x_i) = y_i$ and $p_j(x_i) = y_j \prod_{k \neq j} \frac{x_i - x_k}{x_j - x_k} = 0$

- Denote the factor of recovery $\prod_{j \neq i} \frac{x - x_j}{x_i - x_j}$ by $r_i(x ; x_0, \dots, x_k)$

Adi Shamir's Scheme (1979)

- Suppose there is a trusted secret distributor with trusted channels
- Set $p(x) = a_0 + a_1 x + \ldots + a_{t-1} x^{t-1}$ of degree $t-1$
  - Let a_0 be the secret
  - Choose $a_1, \ldots, a_{t-1}$ randomly
- Distribute $p(1), p(2), \ldots, p(n)$ to n participants
- t of the n points $(1, p(1)), (2, p(2)), \ldots, (n, p(n))$ can recover $p(x)$, hence the secret a_0 [$=p(0)$]
- $t-1$ of the m points can not obtain any information about a_0
- The coefficient of recovery is $r_i(0\,; x_1, \ldots, x_t)$ in $\mathbb{Q}$ or $\mathbb{F}_q$

Feldman's Verifiable Secret Sharing

- Participant *i* can verify if the value v_i received is equal to $p(i)$
- The distributor has to make commitments to the polynomial p
  - Assuming discrete logarithm problem is hard on **additive** cyclic group $G = <g>$
  - Publish $c_0 = a_0 \cdot g, \ldots, c_{t-1} = a_{t-1} \cdot g$ as elements of G before distribution
- Participant *i* verifies if $v_i \cdot g = c_0 + (i \cdot c_1) + \ldots + (i^{t-1} \cdot c_{t-1})$ holds
  - LHS = $p(i) \cdot g = a_0 \cdot g + (i \cdot a_1) \cdot g + \ldots + (i^{t-1} \cdot a_{t-1}) \cdot g$ = RHS
- If no participants fail the examination, this guarantees that the distributor did not cheat
- Note that the distributor knows the secret a_0

Curve25519, EdDSA

Curve25519

- $p = 2^{255} - 19$ 是質數
- 定義橢圓曲線

$$E: y^2 = x^3 + 486662x^2 + x$$

  - Montgomery Curve
  - Birationally Equivalent to a Twisted Edward Curve
- 考慮橢圓曲線群 $E\left(\mathbb{F}_{p^2}\right)$，即允許座標 $x, y \in \mathbb{F}_{p^2}$
  - 代入任意 $x \in \mathbb{F}_p$，都可以開根號解出 $y \in \mathbb{F}_{p^2}$
- Base point

$$Q = \left(9, \sqrt{39420360}\right)$$

Edwards Curve

- Curve equation

$$E: ax^2 + y^2 = 1 + dx^2y^2$$

  - Special case $a = 1$, called Untwisted
  - In Ed25519, $a = -1$
- Addition and Doubling

$$(x_1, y_1) + (x_2, y_2) = \left(\frac{x_1y_2 + y_1x_2}{1 + dx_1x_2y_1y_2}, \frac{y_1y_2 - ax_1x_2}{1 - dx_1x_2y_1y_2}\right)$$

- Neural point (0,1)
- **Theorem**. Every twisted Edwards curve is birationally equivalent to an Montgomery curve

Edwards Curve

- Montgomery curve

$$M: Bv^2 = u^3 + Au^2 + u$$

- Twisted Edwards curve

$$E: ax^2 + y^2 = 1 + dx^2y^2$$

- The birational map is defined by

$$x = \frac{u}{v}, y = \frac{u-1}{u+1}$$

- With the coefficients

$$a = \frac{A+2}{B}, d = \frac{A-2}{B}$$

Edwards Curve

- Curve25519

$$M: v^2 = u^3 + 486662u^2 + u$$
$$A = 486662, B = 1$$

- The coefficients of the twisted Edwards curve

$$a = \frac{A+2}{B} = 486664, d = \frac{A-2}{B} = 486660$$

- Twisted Edwards curve

$$E: 486664x^2 + y^2 = 1 + 486660x^2y^2$$

- Changing variable $x \mapsto \sqrt{-486664} \cdot x$

$$E: -x^2 + y^2 = 1 - \frac{121665}{121666}x^2y^2$$

Edwards Curve

- Base point for Curve25519

$$Q = \left(9, \sqrt{39420360}\right)$$

- The coordinate $u = 9$
- The corresponding point on twisted Edwards curve

$$x = \frac{u}{v}, y = \frac{u-1}{u+1} = \frac{4}{5}$$

- The coordinate x is chosen to be positive

EdDSA

- Public Parameters
  1. Odd prime p, and base field $\mathbb{F}_p$
  2. Encoding length b such that $2^{b-1} > p$
  3. Cryptographic hash function $H(x)$ has $2b$ bits output
  4. A non-square element $d \in \mathbb{F}_p$
  5. A non-zero square element $a \in \mathbb{F}_p$
  6. Elliptic curve $E: ax^2 + y^2 = 1 + dx^2y^2$ over $\mathbb{F}_p$
  7. Base point $B \neq (0,1)$ on E of prime order L
  8. Integer $c = 2$ or 3, the base 2 logarithm of cofactor
  9. Integer n with $c \leq n < b$
     - Secret scalar $s = 2^n + \sum_{i=c}^{n-1} 2^i h_i$

EdDSA

- KeyGen
  1. Master private key: b bits string k
  2. Compute hash value $H(k) = (h_0, h_1, \ldots, h_{2b-1})$
  3. Compute secret scalar with least significant b bits
  $$s = 2^n + \sum_{i=c}^{n-1} 2^i h_i$$
  4. Compute EC scalar multiplication $A = [s]B$
  5. Public key: EC point A
  6. Private key: s, most significant b bits $(h_b, \ldots, h_{2b-1})$

EdDSA

- Sign
  1. Input message M and secret key s, $(h_b, \ldots, h_{2b-1})$
  2. Compute $r = H(h_b, \ldots, h_{2b-1}, M)$
  3. Compute EC scalar multiplication $R = [r]B$
  4. Compute scalar for verify
  $$S \equiv r + s \cdot H(R, A, M) \bmod L$$
  5. Signature: (R, S)

EdDSA

- Verify
  1. Input message M, signature (R, S) and public key A
  2. Compute EC scalar multiplication $P = [2^c S]B$
  3. Compute EC operation
  $$Q = [2^c]R + [2^c H(R, A, M)]A$$
  4. Accept the signature if $P = Q$

EdDSA

- Correctness

$$
\begin{aligned}
[2^c S]B &= \left[2^c\left(r + sH(R, A, M)\right)\right]B \\
&= [2^c r]B + [2^c sH(R, A, M)]B \\
&= [2^c]R + [2^c H(R, A, M)]A
\end{aligned}
$$

Ed25519 vs ECDSA NIST P-256

- According to Ed25519 Web site (https://ed25519.cr.yp.to/)
  - Westmere CPU (Intel Xeon E5620, hydra2)
  - Signing: 87548 cycles (109000 messages per second)
  - Verification: 273364 cycles
- According to the eBACS
  - **E**CRYPT **B**enchm**a**rking of **C**ryptographic **S**ystems
  - Ed25519 takes at range (time on real computer)
    - 524288 - 1048576 for optimized implementation for AMD 64
    - 1048576 - 4194304 for reference implementation
  - ECDSA with NIST P-256 takes at range
    - 1048576 - 8388608 for openssl implementation