Elliptic Curve Cryptography

Bimal Roy
Indian Statistical Institute, Kolkata.

(Based on a talk by Prof. Rana Barua)

Public Key Cryptography

- Two functions E_{k_1} and D_{k_2} inverse of each other, i.e.,

$$E_{k_1}(D_{k_2}(x)) = x = D_{k_2}(E_{k_1}(x))$$

- Functional forms of $E(.)$ and $D(.)$ are known, but from k_1 (or k_2), k_2 (or k_1) can not be computed "easily". One of k_1 or k_2 is made public and the other is kept private.
- To communicate between users A and B
  - A: public key k_1, private key k_2.
  - B: public key k_3, private key k_4.
- Encryption for a message M that A wants to send to B:

$$C = D_{k_2}(E_{k_3}(M))$$

Decryption:

$$D_{k_4}(E_{k_1}(C)) = M$$

ElGamal Public Key Cryptosystem

- Key Generation:
  1. Choose a suitable large prime p
  2. Choose a generator g of the cyclic group $\mathbb{Z}_p^*$
  3. Choose a cyclic $G =< g >$ of prime order p.
  4. choose $x_A \leftarrow \mathbb{Z}_p$ and compute $y_A = g^{x_A} \mod p$.
  5. Public key of Alice is (g, y_A) and secret key is x_A.
- Encryption: Given message $m \in \mathbb{Z}_p^*$,
  1. choose $r \in \mathbb{Z}_p$ and compute $h = g^r \mod p$
  2. send ciphertext $(h, y_A^r.m \mod p)$
- Decryption: On receiving ciphertext (h, z), compute

$$m = (h^{x_A})^{-1}.z \mod p$$

Security of ElGamal

- Discrete Logarithm Problem.
- Diffie-Hellman Problem.

Discrete Logarithm:

- Instance: A multiplicative group $(G, .)$, an element $\alpha \in G$ of order n, and an element $\beta \in <\alpha>$, the cyclic group generated by α.
- Problem: Find the unique integer $a, 0 \leq a \leq n-1$, s.t. $\alpha^a = \beta$. The integer a is called the discrete log of β to base α and is denoted by $log_\alpha \beta$.
- Computing the discrete log is probably difficult in suitable groups. Thus the exponentiation function is (probably) a one-way function in suitable groups G, i.e. a function which is easy to compute but computationally infeasible to invert.

Computational Diffie-Hellman Problem

- Instance: A multiplicative group $(G, .)$, an element $\alpha \in G$ of order n, and an element $\beta \in <\alpha>$, the cyclic group generated by α.
- Problem: Compute α^{ab} .
- Diffie-Hellman Problem is stronger than the DLP

Questions What groups G should be chosen for ElGamal Cryptosystems?

- Obvious choice is $\mathbb{Z}_p^*$, for large primes p
  p should be carefully chosen to avoid known algorithms for DLP. e.g. $p-1$ should contain at least one large prime factor.
- Elliptic Curves
- Hyperelliptic curves
- Others

Reasons for using ECC

- Shorter secret key. Lenstra and Verheul made some comparative security estimates. They have argued that in order for a ECDLP based cryptosystem to be secure one should take $p \equiv 2^{160}$. To achieve the same level of security in case of $\mathbb{Z}_p^*$ p needs to be at least 2^{1880}
- Memory efficient implementation.
- Higher speed.

Elliptic Curve over a Finite Field

- An elliptic curve E over a finite field $K = \mathbb{F}_q(\mathbb{Z}_p, p > 3)$ is given by an equation

$$y^2 = x^3 + ax + b, a, b \in K$$

, where $4a^3 + 27b^2 \neq 0$

- The set of K-rational points on E is

$$E(K) = \{(x, y) \in K \times K : y^2 = x^3 + ax + b\} \cup \{\mathcal{O}\}.$$

Elliptic Curve over a Finite Field

The set $E(L)$ is an abelian group under the "chord-and-tangent law"'. Consider $E/K : y^2 = x^3 + ax + b$. Addition formulae are as follows:

1. $P + \mathcal{O} = \mathcal{O} + P = P$, for all $P \in E(L)$.
2. $-\mathcal{O} = \mathcal{O}$.
3. If $P = (x, y) \in E(L)$, then $-P = (x, -y)$.
4. If $Q = -P$, then $P + Q = \mathcal{O}$.
5. If $P = (x_1, y_1) \in E(L), Q = (x_2, y_2) \in E(L), P = -Q$, then $P + Q = (x_3, y_3)$, where $x_3 = \lambda^2 - x_1 - x_2$, $y_3 = \lambda(x_1 - x_3) - y_1$, and

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} \text{ if } P \neq Q;$$

$$\lambda = \frac{3x_1^2 + a}{2y_1} \text{ if } P = Q.$$

Elliptic Curve over a Finite Field

Suppose P and Q are both points on the elliptic curve then $P + Q$ is always another point on the elliptic curve which is defined as follows. Draw a line through P and Q (if $P = Q$ take the Tangent line). The line intersects the curve in a third Point. Reflect that point through the x-axis to find $R = P + Q$

Elliptic Curve over a Finite Field

- (Hasse's Theorem) $\#E(\mathbb{F}_q) = q + 1 - t, |t| \leq \sqrt{q}$. Consequently, $\#E(\mathbb{F}_q) \approx q$.
- (Schoof's Algorithm) $\#E(\mathbb{F}_q)$ can be computed in polynomial time.
- Let E be an elliptic curve defined over $\mathbb{F}_q$. Then $E(\mathbb{F}_q) \cong \mathbb{Z}_{n_1} \oplus \mathbb{Z}_{n_2}$, where $n_2 | n_1$ and $n_2 | (q-1)$.
- $E(\mathbb{F}_q)$ is cyclic if and only if $n_2 = 1$.
- $P \in E$ is an n-torsion point if $nP = \mathcal{O}$ and $E[n]$ is the set of all n-torsion points.
- If $\gcd(n, q) = 1$, then $E[n] \cong \mathbb{Z}_n \oplus \mathbb{Z}_n$

Point Compress

- Let E be an elliptic curve over $\mathbb{Z}_p$.
- Define

$$PC : E - \{\mathcal{O}\} \mapsto \mathbb{Z}_p \times \mathbb{Z}_2$$

as follows

$$PC(P) = (x, y(\mod 2)), \text{ where } P = (x, y) \in E.$$

Simplified ECIES

Let E be an elliptic curve over $\mathbb{Z}_p$ s.t. E contains a cyclic subgroup $H =< P >$ of prime order n in which the DLP is infeasible

- Pick $m \leftarrow \mathbb{Z}_n$ and set $Q = mP$
  Public key : P, Q, n Private key; m
- Encrypt: Given message $x \in \mathbb{Z}_p^*$ choose a secret random no. $k \in \mathbb{Z}_n^*$
  - Compute $y_1 = PC(kP)$ and $kQ = (x0, y0), x_0 \neq 0$.
  - Compute $y_2 = xx_0(\mod p)$.

  Ciphertext is (y_1, y_2)
- Decrypt: Given cipher (y_1, y_2)
  - Compute $(x_0, y_0) = mPC^{-1}y_1$
  - Compute $x = y_2(x_0^{-1})(\mod p)$

Public Key Signature Scheme

A signature scheme is given by following algorithms:

- $Setup(1^k)$: A PPT algorithm which takes a security parameter as input and outputs public parameters Params.
- $KG(Params)$: A PPT algorithm which takes Params as input and outputs a public-private key pair (PK, SK).
- $SIG(m, SK, Params)$: A PPT algorithm which takes a message m, a secret key SK and Params as input and outputs a signature σ.
- $VER(m, \sigma, PK, Params)$: A deterministic polynomial time algorithm which takes a message m, a signature σ, a public key PK and Params as input and outputs T if σ is a valid signature on message m, else it returns F.

Security Notion of Signature Scheme

A signature scheme is said to be EUF-CMA (existentially unforgeable against chosen message attack) secure if no probabilistic polynomial time algorithm has a non-negligible advantage in the following game.

$Game_{SIG,\mathcal{A}}^{EUF-CMA}(1^k)$

- $L \leftarrow \phi$
- $Params \leftarrow Setup(1^k)$
- $(PK, SK) \leftarrow KG(Params)$
- $(m, \sigma) \leftarrow \mathcal{A}^{\mathcal{O}}(SK, Params)$
- $x \leftarrow VER(m, \sigma, PK, Params)$

Advantage of $\mathcal{A}$ is defined as $Adv(\mathcal{A}) = Pr(x = true \wedge m \notin L)$

Setup

- Select an elliptic curve E defined over $\mathbb{Z}_p$. The number of points in $E(\mathbb{Z}_p)$ should be divisible by a large prime n.
- Select a point $P \in E(\mathbb{Z}_p)$ of order n.
- Select an integer d in the interval $[1, n-1]$.
- Compute $Q = dP$.
- A's public key is $(E; P; n; Q)$; A's private key is d.

ECDSA(contd)

ECDSA signature generation. To sign a message m, A does the following:

- Select a random integer k in the interval $[1, n-1]$.
- Compute $kP = (x_1; y_1)$ and $r = x_1 (\mod n)$.
- Compute $k^{-1} \mod n$.
- Compute $s = k - 1[h(m) + dr] \mod n$, where h is the Secure Hash Algorithm (SHA-1).
- The signature for the message m is the pair of integers $(r; s)$.

ECDSA(contd)

ECDSA signature verification. To verify A's signature $(r; s)$ on m, B should:

- Compute $w = s^{-1} \mod n$ and $h(m)$.
- Compute $u_1 = h(m)w \mod n$ and $u_2 = rw \mod n$.
- Compute $u_1 P + u_2 Q = (x_0; y_0)$ and $v = x_0 \mod n$. Accept the signature if and only if $v = r$.

The parameter n should have about 160 bits. If this is the case, then ECDSA signatures have size 320 bits (same as DSA).

Thank You