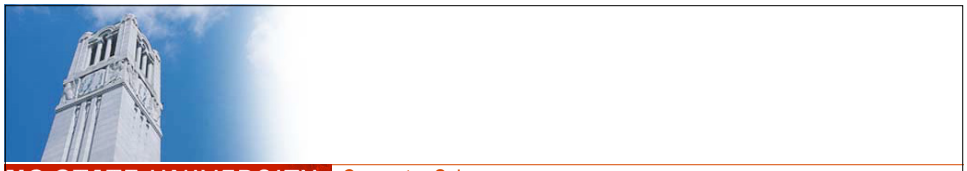


NC STATE UNIVERSITY Computer Science

CSC 774 Advanced Network Security

Topic 2.6 ID Based Cryptography #2

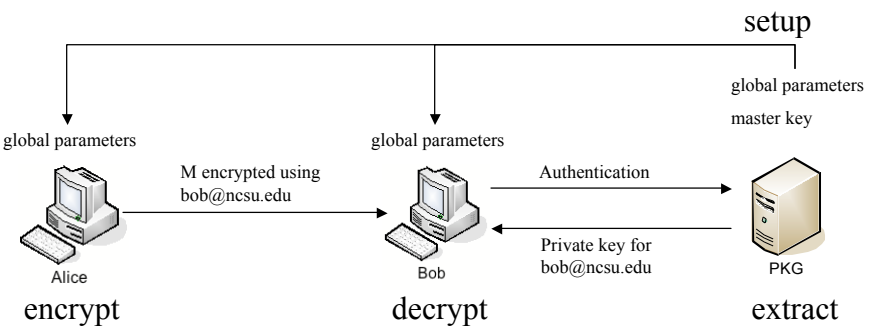
Slides by An Liu

Outline

- Applications
- Elliptic Curve Group over real number and F_p
- Weil Pairing
- BasicIdent
- FullIdent
- Extensions
- Escrow ElGamal Encryption

NC STATE UNIVERSITY Computer Science

Identity-Based Encryption

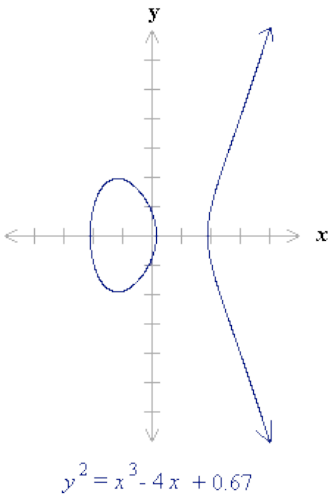


Applications

- Revocation of public keys
  - bob@ncsu.edu || 2006
  - bob@ncsu.edu || 2006-10-20
  - Send message into the future
- Delegation of decryption keys
  - Delegation to a laptop (use date as public key)
  - Delegation of duties (use subject as public key)

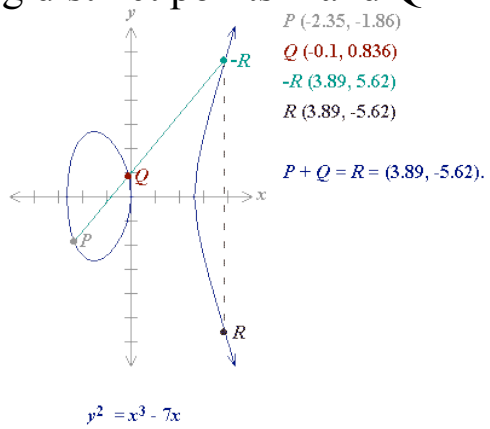
Elliptic Curve Group over Real Numbers

- $y^2 = x^3 + ax + b$
  - x, y, a, b are real numbers
- If $4a^3 + 27b^2 \neq 0$, a group can be formed.
  - points on curve and infinity point
  - Additive group



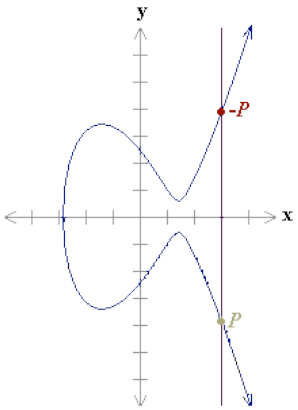
Elliptic Curve Addition: A Geometric Approach

- Adding distinct points P and Q



* The negative of a point P is its reflection in the x-axis.

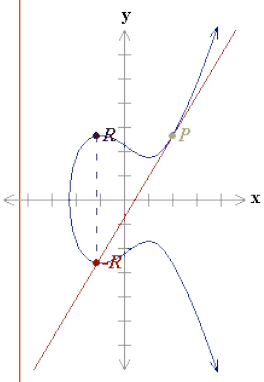
- Adding the points P and -P



$P + (-P) = O$

$y^2 = x^3 - 6x + 6$

- Doubling the point P



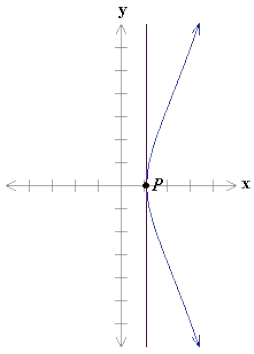
P (2, 2.65)

$-R$ (-1.11, -2.64)

R (-1.11, 2.64)

$2P = R$ = (-1.11, 2.64).

$y^2 = x^3 - 3x + 5$



P (1.1,0)

Since $y_P = 0$, $2P = O$, the point at infinity.

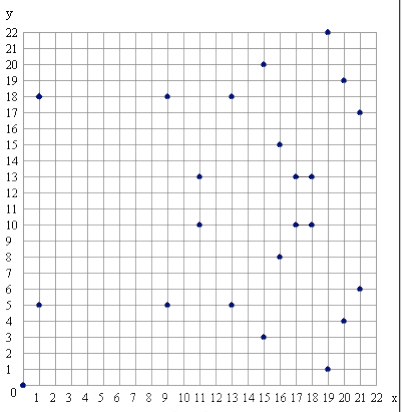
$y^2 = x^3 + 5x - 7$

Elliptic Curve Addition: An Algebraic Approach

- Adding distinct points P and Q (P+Q=R)
  - $P(x_P,y_P)$ and $Q(x_Q,y_Q)$ are not negative each other
  - $s = (y_P - y_Q)/(x_P - x_Q)$
  - $x_R = s^2 - x_P - x_Q$, $y_R = -y_P + s(x_P - x_R)$
- Doubling the point P (2P=R)
  - $y_P \neq 0$
  - $s = (3x_P^2 + a)/2y_P$
  - $x_R = s^2 - 2x_P$, $y_R = -y_P + s(x_P - x_R)$

Elliptic Curve Groups over Fp

- Calculations over real number are slow and inaccurate.
- $y^2 \bmod p = x^3 + ax + b \bmod p$
  - x, y, a, b are in F_p
- finite set of points
- no geometric approach



Elliptic curve equation: $y^2 = x^3 + x$ over F_{23}

Elliptic Curve Groups over Fp (Cont'd)

- Adding distinct points P and Q (P+Q=R)
  - $P(x_P, y_P)$ is not $-Q = (x_Q, -y_Q \bmod p)$
  - $s = (y_P - y_Q)/(x_P - x_Q) \bmod p$
  - $x_R = s^2 - x_P - x_Q \bmod p$
  - $y_R = -y_P + s(x_P - x_R) \bmod p$
- Doubling the point P (2P=R)
  - $y_P \neq 0$
  - $s = (3x_P^2 + a)/2y_P \bmod p$
  - $x_R = s^2 - 2x_P \bmod p$, $y_R = -y_P + s(x_P - x_R) \bmod p$

Elliptic Curve Discrete Logarithm Problem (ECDLP)

- Discrete Logarithm Problem
  - For multiplicative group Z_p^*, given r, q, p, find k such that $r = q^k \bmod p$.
  - Foundation of many cryptosystems.
- Scalar multiplication
  - P, 2P, 3P=2P+P, 4P=3P+P,… , kP (additive notation)
- ECDLP
  - Given points Q, P, find k such that kP=Q

Weil Pairing

- Bilinear map
  - A map e: $G_1 \times G_1 \rightarrow G_2$
  - $\forall P,Q \in G_1$, $\forall a,b \in Z$, $e(aP, bQ) = e(P, Q)^{ab}$
- Weil Pairing
  - bilinear map
    - G_1 is the group of points of an elliptic curve over F_p
    - G_2 is a subgroup of $F_{p^2}^*$
  - efficiently computable
    - Miller's algorithm

Weil Pairing (Cont'd)

- Elliptic Curve Group in this paper
  - P, q are primes, $p = 2 \bmod 3$, $p = 6q - 1$
  - E is the elliptic curve defined by $y^2 = x^3 + 1$ over F_p
  - G_q is the group with order $q = (p+1)/6$ generated by $P \in E/F_p$
- Modified Weil pairing
  - $\hat{e}: G_q \times G_q \rightarrow \mu_q$
  - μ_q is the subgroup of $F_{p^2}^*$ containing all elements of order q
  - Non-degenerate: $\hat{e}(P, P) \in F_{p^2}$ is generator of μ_q

Weil Diffie-Hellman Assumption (WDH)

- Given $< P, aP, bP, cP >$ for random $a,b,c \in Z_q^*$, $P \in E/F_p$, compute $W = \hat{e}(P,P)^{abc} \in F_{p^2}$
- When p is a random k-bit prime, there is no probabilistic polynomial time algorithm for the WDH problem.

MapToPoint algorithm

- Convert arbitrary string $ID \in \{0,1\}^*$ to a point $Q_{ID} \in E/F_p$ of order q
- hash function G: $\{0,1\}^* \rightarrow F_p$
- Steps:
  - $y_0 = G(ID)$, $x_0 = (y_0^2 - 1)^{1/3} = (y_0^2 - 1)^{(2p-1)/3}$
  - $Q = (x_0, y_0) \in E/F_p$, $Q_{ID} = 6Q$

BasicIdent – Setup

- Use the elliptic curve group we already defined
- Choose arbitrary $P \in E/F_p$ of order q
- Pick random $s \in Z_q^*$ and set $P_{pub} = sP$
- Choose hash functions
  - H: $F_{p^2} \rightarrow \{0,1\}^n$
  - G: $\{0,1\}^* \rightarrow F_p$
- Message space $M = \{0,1\}^n$, ciphertext space is $C = E/F_p \times \{0,1\}^n$
- System parameters are <p, n, P, P_{pub}, G, H>. Master-key is *s*.

BasicIdent (Cont'd)

- Extract (get private key from ID)
  1. Use MapToPoint to map ID to a point Q_{ID}
  2. Private key corresponding to ID is $d_{ID} = sQ_{ID}$

- Encrypt (encrypt M with ID)
  1. Use MapToPoint to map ID to a point Q_{ID}
  2. Choose random $r \in Z_q$
  3. $C = <rP, M \oplus H(g_{ID}^r)>$ where $g_{ID} = \hat{e}(Q_{ID}, P_{pub}) \in F_{p^2}$

BasicIdent (Cont'd)

- Decrypt (decrypt $C = <U,V>$)
  - If U is not a point of order q, reject the ciphertext
  - Otherwise, $M = V \oplus H(\hat{e}(d_{ID}, U))$
- Why M can be recovered?

  $\hat{e}(d_{ID}, U) = \hat{e}(sQ_{ID}, rP) = \hat{e}(Q_{ID}, P)^{sr} = \hat{e}(Q_{ID}, P_{pub})^{r} = g_{ID}^{r}$

  $V \oplus H(\hat{e}(d_{ID}, U)) = M \oplus H(g_{ID}^{r}) \oplus H(g_{ID}^{r}) = M$

FullIdent

- BasicIdent is not chosen ciphertext secure.
- Setup
  - In addition to BasicIdent, pick another two hash functions:
    - H_1: $\{0,1\}^n \times \{0,1\}^n \rightarrow F_q$
    - G_1: $\{0,1\}^n \rightarrow \{0,1\}^n$
- Extract
  - Same as BasicIdent

FullIdent (Cont'd)

- Encrypt (encrypt M using ID)
  1. Use MapToPoint to convert ID into point Q_{ID}
  2. Choose random $\sigma \in \{0,1\}^n$
  3. Set $r = H_1(\sigma, M)$
  4. $C = < rP, \sigma \oplus H(g_{ID}^r), M \oplus G_1(\sigma) >$ where $g_{ID} = \hat{e}(Q_{ID}, P_{pub}) \in F_{p^2}$

FullIdent (Cont'd)

- Decrypt (decrypt $C=<U,V,W>$)
  1. Compute $V \oplus H(\hat{e}(d_{ID}, U)) = \sigma$
  2. Compute $W \oplus G_1(\sigma) = M$
  3. Set $r = H_1(\sigma, M)$
  4. If $U \neq rP$, reject.

Extensions & Observations

- Tate pairing and other curves can improve the speed
- Distributed PKG
- IBE implies signatures
  - Master-key s is private key (sign)
  - Global system parameters is public key (verify)
  - Signature of M: sQ_M
  - Verification: encrypt random M' use ID=M, then decrypt use sQ_M

Escrow ElGamal Encryption

- Setup
  - Use same elliptic curve
  - Pick a random $s \in Z_q$, $Q = sP$
  - Choose hash function: $F_{p^2} \rightarrow \{0,1\}^n$
  - System parameters: $< p, n, P, Q, H >$
  - s is the escrow key
- Keygen
  - User randomly choose $x \in Z_q$ as private key
  - Public key is $P_{pub} = xP$

Escrow ElGamal Encryption (Cont'd)

- Encrypt
  - Pick random $r \in Z_q$
  - $C = < rP, M \oplus H(g^r) >$ where $g = \hat{e}(P_{pub}, Q) \in F_{p^2}$
- Decrypt ($C = <U,V>$)
  - $V \oplus H(\hat{e}(U, xQ)) = M$
- Escrow-decrypt
  - $V \oplus H(\hat{e}(U, sP_{pub})) = M$