MH4311 Cryptography

Lecture 18

Secret Sharing

Wu Hongjun

Lecture Outline

- Classical ciphers
- Symmetric key encryption
- Hash function and Message Authentication Code
- Public key encryption
- Digital signature
- **Key establishment and management**
  - Key generation
  - Key establishment
    - Key establishment using symmetric key cryptography
    - Key establishment using public key cryptography
      - PKI, Certificate: TLS/SSL
      - SSH
  - **Secret sharing**
    - **Simple secret sharing**
    - **Shamir's threshold secret sharing scheme**
    - **Threshold public key cryptosystem**
- Introduction to other cryptographic topics

Recommended Reading

- CTP: Section 13.1
- Wiki

  Secret Sharing:

  - http://en.wikipedia.org/wiki/Secret_sharing
  - http://en.wikipedia.org/wiki/Shamir%27s_Secret_Sharing
  - http://en.wikipedia.org/wiki/Threshold_cryptosystem

Secret Sharing

- Secret sharing
  - Distribute a secret among a group of participants, each of them is allocated a share of the secret.
  - The secret can be reconstructed only when a sufficient number of shares are combined together; individual shares are of no use on their own

Secret Sharing

- Threshold secret sharing
  - Distribute a secret among *n* participants
  - The secret can be reconstructed only when **at least *t*** shares are combined together
    - *t*-out-of-*n* secret sharing scheme
    - Alternatively, denoted as (*t*, *n*) secret sharing scheme
  - Application example:
    - In the early 1990s in Russia, control of nuclear weapons depended upon a two-out-of-three access mechanism
      - Three parties: president, defense minister, defense ministry
      - Any two parties can control nuclear weapons

Secret Sharing: (n, n) secret sharing

- n-out-of-n secret sharing
  - The simplest threshold secret sharing
  - Distribute a secret among n participants
  - The secret can be reconstructed only when all the shares are combined together

Secret Sharing: (n, n) secret sharing

- The n-out-of-n secret sharing scheme
  - Let the secret be encoded as an integer S
  - Generate n-1 random number r_i $(1 \le i \le n\text{-}1)$, where each r_i is the same size as that of S
  - Let $r_n = S \oplus r_1 \oplus r_2 \oplus r_3 \oplus \ldots \oplus r_{n\text{-}1}$
  - r_i $(1 \le i \le n)$ are the n shares of the secret s
    - Reconstructing S requires all the n shares
    - $S = r_1 \oplus r_2 \oplus r_3 \oplus \ldots \oplus r_{n\text{-}1} \oplus r_n$

Secret Sharing: (n, n) secret sharing

- Security
  - Unconditionally secure
    - With less than n shares, no information of S can be recovered

Secret Sharing: (n, n) secret sharing

- Insecure n-out-of-n secret sharing scheme
  - Suppose that a secret key K is to be shared among n participants
  - Divide K into small pieces (K is αn-bit, each k_i is α-bit)

    $$K = k_1 \parallel k_2 \parallel k_3 \parallel \ldots. \parallel k_n$$
  - The i-th participant receives k_i
- Attack
  - With t shares, $t \cdot \alpha$ bits of K are known
    - Then recovering K requires only $2^{(n-t) \cdot \alpha}$ computations, instead of $2^{n \cdot \alpha}$ computations (less than n shares can be used to reconstruct the secret)

Secret Sharing: (n, n) secret sharing

- (n, n) secret sharing is not robust
  - If one share is lost, the secret cannot be recovered
  - So (t, n) secret sharing is needed.

Secret Sharing: Shamir's Secret Sharing Scheme

- Shamir's secret sharing scheme
  - (t, n) secret sharing scheme
  - Based on simple math
- Basic idea
  - 2 points are sufficient to define a line
  - 3 points are sufficient to define a parabola
  - 4 points to define a cubic curve

=> it takes t points to define a polynomial of degree t-1

Secret Sharing: Shamir's Secret Sharing Scheme

- Shamir's scheme over finite field GF(p)
  (p is a large prime, the secret S is an integer less than p)
  - Generate t-1 random integers a_i (i = 1, 2, 3, …, t-1)
    (each a_i is a random integer less than p)
  - Let $a_0 = S$
  - Build the polynomial over GF(p):
    $$f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + \ldots. + a_{t-1}x^{t-1}$$
  - Compute n points: (j, $f(j)$) for j = 1, 2, 3, …, n
  - Each participant is given a point (a share)
- Reconstructing the secret S
  - Given t points, solve the linear equations to determine the coefficients of the polynomial
  - $S = a_0$

Secret Sharing: Shamir's Secret Sharing Scheme

- Instead of solving linear equations, we can use Lagrange interpolation to find the coefficient a_0 efficiently
- Lagrange polynomial
  - Given t points (x_j, y_j) of a polynomial with degree t-1, the interpolation polynomial in the Lagrange form can be written as

$$L(x) = \sum_{j=1}^{t} y_j \lambda_j(x), \text{ where } \lambda_j(x) = \Big(\prod_{\substack{1 \le i \le t \\ i \ne j}} \frac{x - x_i}{x_j - x_i}\Big) \bmod p$$

- The coefficient a_0 is given as

$$a_0 = L(0) = \Big(\sum_{j=1}^{t} y_j \lambda_j(0)\Big) \bmod p$$

Secret Sharing: Shamir's Secret Sharing Scheme

- Security
  - Unconditionally secure
    - With less than t shares (points), no information of S can be recovered

Threshold public key cryptosystem

Threshold public key cryptosystem

- In a public key cryptosystem, the simple (n, n) secret sharing scheme and Shamir's (t, n) secret sharing scheme can be used to share a private key, and the private key can be successfully reconstructed if sufficient number of shares are given
  - But reconstructing the private key may not be a good idea since the hacker may get this private key
  - How to design such a threshold public key cryptosystem?

Threshold public key cryptosystem

- (n, n) threshold public key cryptosystem
  - Example: (3,3) threshold ElGamal encryption scheme
    - Let the private key $x = x_1+x_2+x_3 \bmod p\text{-}1$, where x_1, x_2 are random integers
    - x_1,x_2, x_3 are given to three participants A_1, A_2 and A_3, respectively

    - After receiving a ciphertext (c_1,c_2),
      - A_1 computes $w_1 = (c_1)^{x1} \bmod p$
      - A_2 computes $w_2 = (c_1)^{x2} \bmod p$
      - A_3 computes $w_3 = (c_1)^{x3} \bmod p$
      - The message is decrypted as: $(w_1 \cdot w_2 \cdot w_3)^{-1} \cdot c_2 \bmod p = m$

        (the message is decrypted without each participant revealing its share x_i to others)

Threshold public key cryptosystem

- (t, n) threshold public key cryptosystem
  - A Simple Example: (2, 3) threshold ElGamal encryption scheme
    - Let the private key $x = x_1+x_2+x_3$ mod p-1, where x_1, x_2 are random numbers
    - x_1, x_2 are given to participant A_1, respectively
      x_2, x_3 are given to participant A_2, respectively
      x_1, x_3 are given to participant A_3, respectively

(continued in the next page)

Threshold public key cryptosystem

- (t, n) threshold public key cryptosystem
  - A Simple Example: (2, 3) threshold ElGamal encryption scheme (cont.)
    - After receiving ciphertext (c_1,c_2),
      - If only A1 and A2 are available, then
        - » A_1 computes $w_1 = (c_1)^{x1} \bmod p$
        - » A_2 computes $w_2 = (c_1)^{x2} \bmod p$; $w_3 = (c_1)^{x3} \bmod p$
        - » The message is decrypted as: $(w_1 \cdot w_2 \cdot w_3)^{-1} \cdot c_2 \bmod p = m$
      - If only A1 and A3 are available, then
        - » A_1 computes $w_1 = (c_1)^{x1} \bmod p$, $w_2 = (c_1)^{x2} \bmod p$
        - » A_3 computes $w_3 = (c_1)^{x3} \bmod p$
        - » The message is decrypted as: $(w_1 \cdot w_2 \cdot w_3)^{-1} \cdot c_2 \bmod p = m$
      - If only A2 and A3 are available, then
        - » A_2 computes $w_2 = (c_1)^{x2} \bmod p$, $w_3 = (c_1)^{x3} \bmod p$
        - » A_3 computes; $w_1 = (c_1)^{x1} \bmod p$
        - » The message is decrypted as: $(w_1 \cdot w_2 \cdot w_3)^{-1} \cdot c_2 \bmod p = m$

Threshold public key cryptosystem

- (t, n) threshold public key cryptosystem based on Shamir's secret sharing scheme
  - Example: Threshold ElGamal encryption scheme
    - Consider a slightly modified ElGamal encryption scheme: the large prime p satisfies p-1= 2q, q is a prime, and the order of g is q
    - The private key x is shared using Shamir's secret sharing scheme over GF(***q***)
      - Generate t-1 random integers a_i (i = 1, 2, 3, …, t-1)
        (each a_i is an integer less than q)
      - Let $a_0 = S$
      - Build the polynomial over GF(q):
        $f(z) = a_0 + a_1 z + a_2 z^2 + a_3 z^3 + \ldots. + a_{t-1} z^{t-1}$
      - Compute n points: $(u_j, z_j) = (j, f(j))$ for $j = 1, 2, 3, \ldots, n$
      - Each participant is given a point (u_j, z_j)

Threshold public key cryptosystem

- (t, n) threshold public key cryptosystem based on Shamir's secret sharing scheme
  - Example: Threshold ElGamal encryption scheme (cont.)
    - After receiving a ciphertext (c_1, c_2), the ciphertext is decrypted by t participants as follows:

–Each participant computes $w_j = (c_1)^{z_j} \bmod p$ for $1 \le j \le t$

–Compute $\lambda_j(0) = \prod_{\substack{1 \le i \le t \\ i \ne j}} \frac{0 - u_i}{u_j - u_i} \bmod q$ (using public information)

–Then the message is obtained as $(\prod_{j=1}^{t} (w_j)^{\lambda_j(0)})^{-1} c_2 \bmod p$

Proof. $\prod_{j=1}^{t} (w_j)^{\lambda_j(0)} \bmod p = \prod_{j=1}^{t} (c_1)^{z_j \cdot \lambda_j(0)} \bmod p = (c_1)^{\sum_{j=1}^{t} z_j \cdot \lambda_j(0)} \bmod p$

$= (c_1)^{a_0} \bmod p = (c_1)^{x} \bmod p$

Summary

- Secret sharing
  - (n, n) secret sharing
  - Shamir's secret sharing scheme
  - Threshold public key cryptosystem
    - (n, n) threshold public key cryptosystem
    - (t, n) threshold public key cryptosystem
    - (t, n) threshold ElGamal encryption scheme based on Shamir's secret sharing scheme