An toàn và An ninh thông tin

Nguyễn Linh Giang

Bộ môn Truyền thông và Mạng máy tính

Khoa CNTT, ĐHBK HN

Một số hệ mật khóa công khai

Nội dung

- Trao đổi khóa Diffie-Hellman
- Chữ ký ElGamal
- Hệ mật Knapsack

Khái quát hệ Diffie-Hellman

- Được đề cập trong một hội thảo do Diffie-Hellman đưa ra vào 1976
- Là sự kết hợp của hai mô hình xác thực và mật của hệ KCK
- Việc sinh ra các cặp khoá là hoàn toàn khác nhau đối với người sử dụng
- Sử dụng cơ chế trao đổi khoá trực tiếp không qua trung gian xác thực

Mục đích ra đời

- Sử dụng để áp dụng cho các ứng dụng có độ mật cao bằng phương pháp trao đổi khoá (key exchange)
- Với nguyên tắc hai người sử dụng có thể trao đổi một khoá an toàn - được dùng để mã hoá các tin nhắn
- Thuật toán tự giới hạnchỉ dùng cho các ứng dụng sử dụng kĩ thuật trao đổi khoá

Cơ sở hình thành thuật toán

- Dựa trên nguyên tắc toán học :với m là một số nguyên tố thì

  "Có thể tính toán dễ dàng $y=a^i \bmod m$ nhưng việc tính ngược lại là rất khó và với m lớn thì dường như là không thể"
- Dựa trên phép tính logarit rời rạc

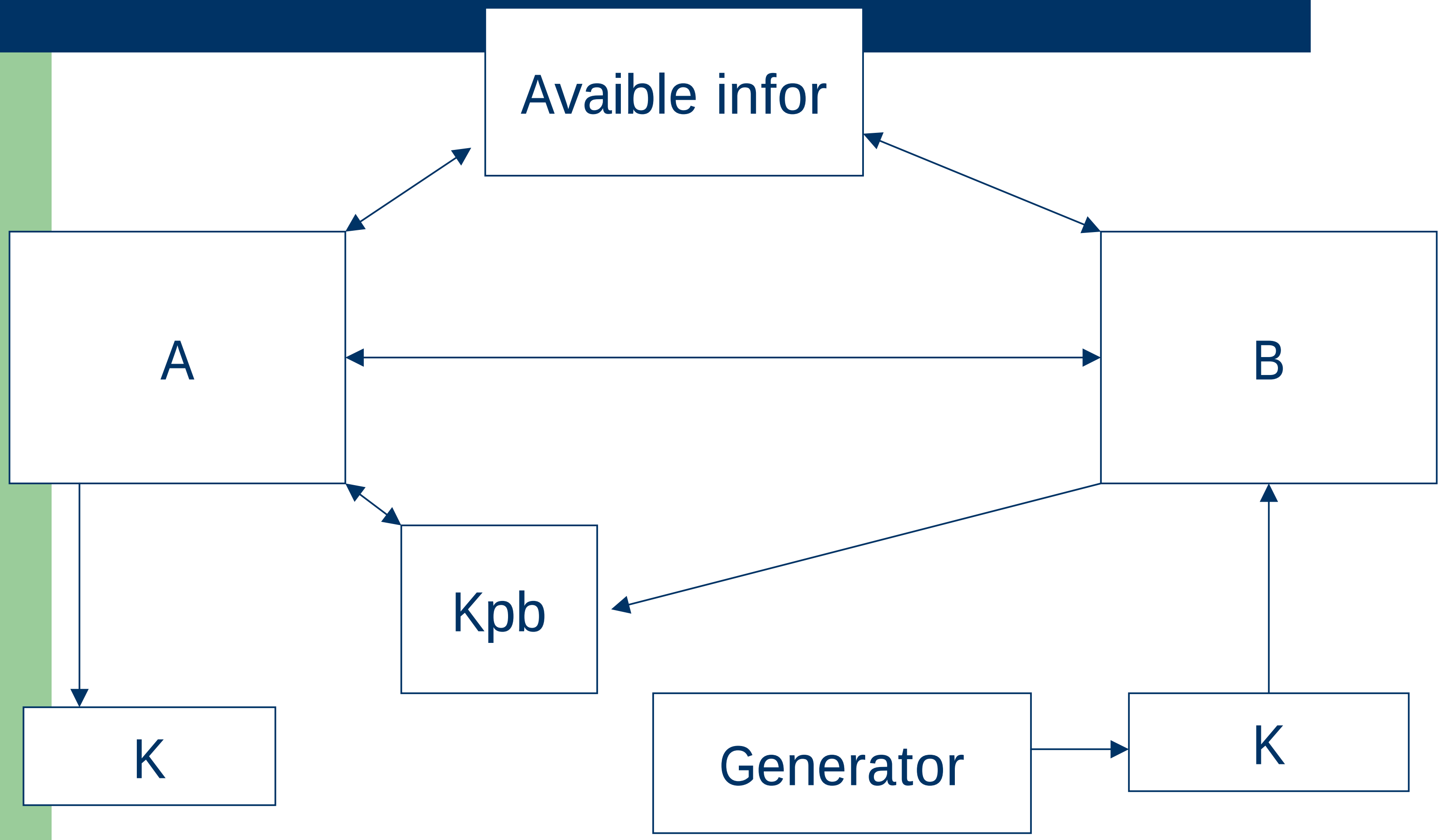
Thuật toán logarit rời rạc

- Một số nguyên tố p
- Một gốc nguyên thuỷ a của p : là các số mà luỹ thừa của nó thuộc (1,p-1)
- Với b bất kì nguyên sẽ luôn ∃i sao cho $b= a^i$ mod p

  Đây thuật toán logarit rời rạc .

  Được coi là cơ sở để hình thành thuật toán này .

Mô hình chung của thuật toán



Thuật toán sinh khóa

- Lựa chọn số nguyên tố p và gốc nguyên thuỷ a
- Khoá của người i
  - Khóa riêng x_i : chọn sao cho x_i <p-1
  - Khoá công khai y_i : $y_i = a^{x_i} \bmod p$
- Khoá của người j
  - Khoá riêng x_j : chọn sao cho x_j <p-1
  - Khoá công khai y_j : $y_j = a^{x_j} \bmod p$
- Khoá mật chung : $K=(y_j)^{x_i} \bmod p=(y_i)^{x_j} \bmod p$

Trao đổi khóa Diffie-Hellman

Thành phần khoá chung cho cả hai

q : số nguyên tố

α : $\alpha < q$ và α là một gốc nguyên thuỷ của q

↓

Người sử dụng i sinh khóa

Lựa chọn khoá riêng x_i : $x_i < q$

Tính toán : $y_i == (\alpha^{x_i} \bmod q)$

↓

Người sử dụng j sinh khóa

Lựa chọn khoá riêng x_j : $x_j < q$

Tính toán : $y_j == (\alpha^{x_j} \bmod q)$

↓

Sinh khoá mật bởi người sử dụng i

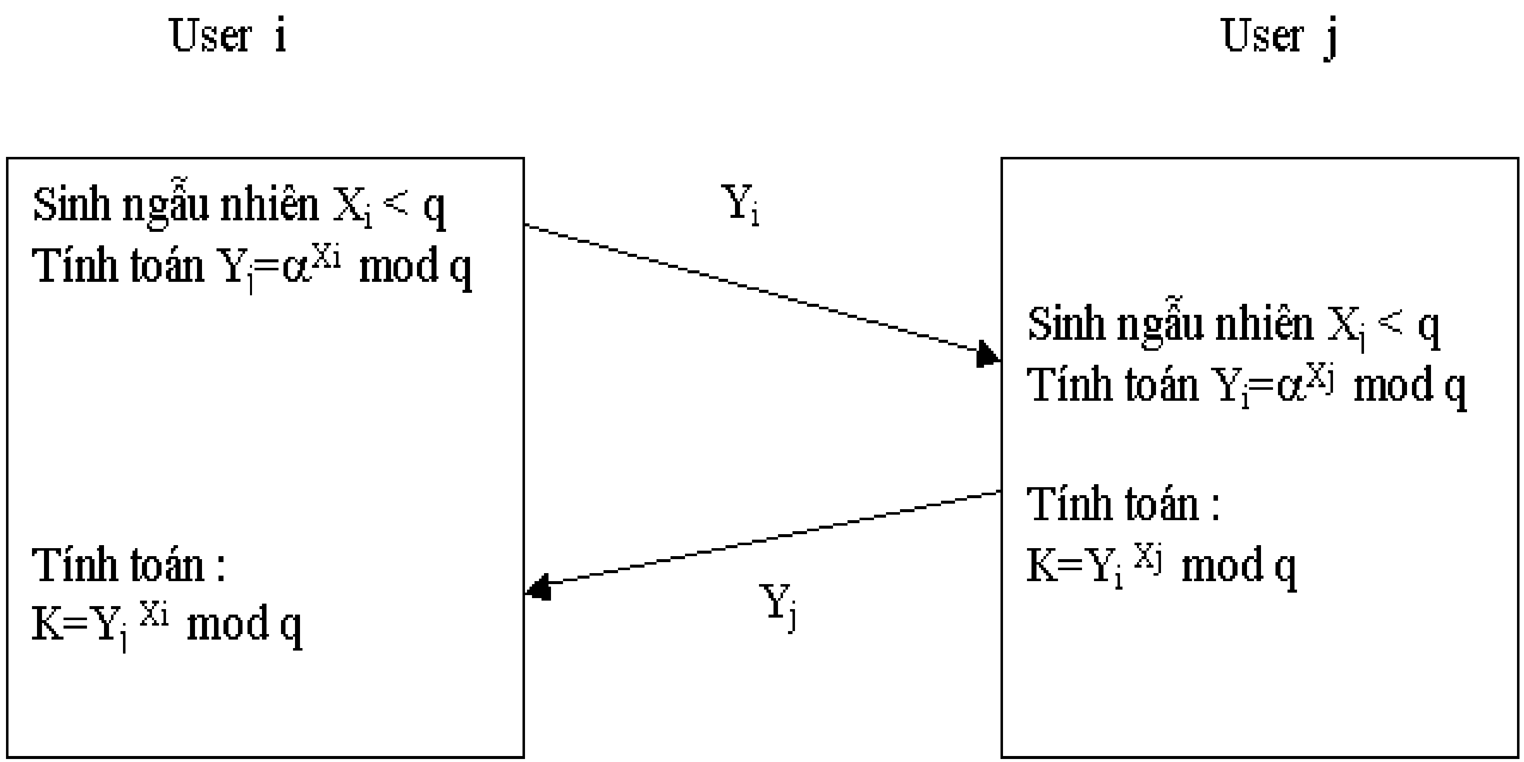
$K = (y_j^{x_i} \bmod q)$

↓

Sinh khoá mật bởi người sử dụng j

$K = (y_i^{x_j} \bmod q)$

Thuật toán trao đổi khoá



Tính an toàn của hệ mật

- Thám mã có sẵn các thông tin :p,a,Y_i,Y_j
- Để có thể giải được K ,X bắt buộc thám mã phải sử dụng thuật toán logarit rời rạc : rất khó nếu p lớn
- Nếu chọn p lớn: việc tính toán ra X, K dường như không thể trong thời gian thực

Hệ mật và thám mã

- Thám mã có thể tấn công vào các thông tin : p ,a,Y_j,Y_j
- Và sử dụng thuật toán rời rạc để tính ra X, sau đó tính ra K
- Quan trọng nhất là độ phức tạp của thuật toán logarit phụ thuộc vào chọn số nguyên tố p

Lĩnh vực ứng dụng

- Tự quá trình thuật toán đã hạn chế ứng dụng chỉ sử dụng cho quá trình trao đổi khoá mật là chủ yếu
- Sử dụng trong chữ kí điện tử.
- Các ứng dụng đòi hỏi xác thực người sử dụng.

ElGamal

- Tạo khóa: p, q, α, a, y=α^a mod p
- Tạo chữ ký:
  - Chọn ngẫu nhiên k, 1 $\leq$ k $\leq$ p-1, gcd(k, p-1)=1
  - Tính r = α^k mod p
  - Tính k^{-1} mod (p-1)
  - Tính s = $k^{-1} * (h(m) - ar)$ mod (p-1)
  - Chữ ký là (r,s)

El Gamal (cont)

- Xác minh chữ ký
  - Xác minh $1 \leq r \leq p-1$
  - Tính $v_1 = y^r r^s \bmod p$
  - tính h(m) and $v_2 = \alpha^{h(m)} \bmod p$
  - Đồng ý nếu $v_1 = v_2$

$$s \equiv k^{-1}\{h(m) - ar\} \pmod{p-1}$$

$$ks \equiv h(m) - ar \pmod{p-1}$$

$$\alpha^{h(m)} \equiv \alpha^{ar+ks} \equiv (\alpha^a)^r r^s \pmod{p}$$

ElGamal (cont)

- Chú ý:
  - k phải đơn nhất đối với mỗi bản tin được ký
    - $(s_1-s_2)k=(h(m_1)-h(m_2)) \bmod (p-1)$
  - Tấn công giả mạo có thể được thiết lập nếu các hàm băm không được dùng

ElGamal (cont)

- Hiệu năng
  - Tạo chữ ký
    - Một module theo hàm mũ
    - Một thuật toán ơclid
    - Cả hai có thể được thực hiện offline
  - Xác minh
    - Three modular exponentiations
- Các chữ ký ElGamal được tạo ra cho các bài toán xác thực, chứng thực

Thuật toán mã hoá công khai Knapsach

- Bài toán Subset Sum
- Mô tả thuật toán Knapsack

Bài toán Subset Sum

- Thuật toán Knapsach được xây dựng dựa trên bài toán Subset Sum

$I = (s_1, \ldots, s_n, T)$, trong đó $s_1, \ldots, s_n$ và T là các số nguyên dương. Các s_i được gọi là sizes và T gọi là target sum.

Câu hỏi là có hay không có một vector $x = (x_1, \ldots, x_n)$ trong đó $x_1, \ldots, x_n \in \{0, 1\}$ sao cho $\sum_{i=1}^{n} x_i s_i = T$?

Thuật toán Knapsack

Cho $s = (s_1, \ldots, s_n)$ là một danh sách các số nguyên tăng nhanh, $p > \sum_{i=1}^{n} s_i$ là số nguyên tố, và $1 \leq a \leq p - 1$.

Với $1 \leq i \leq n$, định nghĩa

$$t_i = a s_i \bmod p$$

Đặt $t = (t_1, \ldots, t_n)$

$$P = \{0, 1\}^n$$

$$C = \{0, \ldots, n(p-1)\}$$

$K = \{(s, p, a, t)\}$ trong đó s, p, a, t được xây dựng như trên.

Thuật toán Knapsack

- KU = {t} là khoá công khai.
- KR = {p, a, s} là khoá mật.
- Hàm mã hoá

$$E_K(x_1, \ldots, x_n) = \sum_{i=1}^{n} x_i t_i$$

- Hàm giải mã

Với $0 \leq y \leq n(p-1)$, định nghĩa $z = a^{-1} y \bmod p$ và giải quyết bài toán $(s_1, \ldots, s_n, z)$ và có được $D_K(y) = (x_1, \ldots, x_n)$ (Hàm giải mã)