TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN CÔNG NGHỆ TRI THỨC

NHẬP MÔN MÃ HOÁ MẬT MÃ

ĐỒ ÁN MÔN HỌC CUỐI KÌ

Các thành viên trong nhóm:

1. Lê Trọng Anh Tú MSSV: 20127091
2. Phan Tuấn Khải MSSV: 20127524

Giáo viên hướng dẫn: Nguyễn Đình Thúc
Ngô Đình Hy
Nguyễn Văn Quang Huy

Mục lục

1 Giới thiệu

Sau 11 tuần được học ở lớp lý thuyết môn *Nhập môn mã hoá mật mã*, nhóm đã làm được 1 hệ thống giao tiếp an toàn giữa 2 người trên mạng cục bộ áp dụng những kiến thức như hệ mã khoá công khai, hệ mã khoá bí mật và cơ chế xác thực thông tin chữ ký số.

2 Bảng phân công công việc

STT	Sinh viên thực hiện	Nội dung	Phần trăm đóng góp
1	Lê Trọng Anh Tú	Thiết lập thông số khi bắt đầu giao tiếp	50%
2		Cài đặt giao thức trao đổi khoá Diffie−Hellman	
3		Thiết lập khoá ma trận cho mã ma trận	
4	Phan Tuấn Khải	Thiết lập kết nối giữa 2 bên thông qua *socket*	50%
5		Thiết kế thuật toán mã hoá và giải mã trên mã ma trận	
6		Cài đặt chữ kí bằng hệ mã RSA và hàm băm mật mã	

3 Hệ thống giao tiếp an toàn giữa 2 người

3.1 Thiết lập thông số ban đầu

Các thông số liệt kê dưới đây được công khai trên kênh truyền và nằm trong file `parameter.txt`

- p là số nguyên tố 1024-bits [1]
- g là căn nguyên thuỷ (hay phần tử sinh) của nhóm $\mathbb{Z}_p^*$ (cách tìm g được trình bày ở phần **Phụ lục**)

Một cải tiến có thể áp dụng để đảm bảo an toàn hơn là ta có thể thay đổi g và p qua mỗi ngày. Việc thay đổi này giúp cho kẻ tấn công không có đủ thời gian để phá với những thông số cũ.

3.2 Trao đổi khoá giữa 2 bên

Đầu tiên, ta cần thiết lập kết nối giao tiếp giữa 2 file `Alice.py` và `Bob.py` thông qua **socket**.
Alice và Bob tiến hành trao đổi khoá thông qua **giao thức Diffie−Hellman**.

- Alice và Bob chọn một số ngẫu nhiên trong khoảng $(2, p-1)$ với p được lấy từ `parameter.txt` và tính $g^a \ (mod\ p)$, $g^b \ (mod\ p)$ với a, b lần lượt là số được chọn ngẫu nhiên bởi Alice và Bob và g được lấy từ `parameter.txt`
- Alice sẽ gửi cho Bob $g^a \ (mod\ p)$ và Bob sẽ gửi lại cho Alice $g^b \ (mod\ p)$. Những thứ được công khai trên kênh truyền là 2 giá trị $g^a \ (mod\ p)$ và $g^b \ (mod\ p)$. Theo như bài toán **DLP**, kẻ tấn công sẽ không thể tìm ra được a và b trong thời gian đa thức.
- Alice sau khi nhận $g^b \ (mod\ p)$ và đang sở hữu a nên tính được $g^{ab} \ (mod\ p)$. Tương tự như vậy, Bob sau khi nhận được $g^a \ (mod\ p)$ và đang sở hữu b cũng tính được $g^{ab} \ (mod\ p)$. Vậy Alice và Bob đã thống nhất được 1 khoá bí mật chung được sử dụng cho quá trình mã hoá và giải mã là $g^{ab} \ (mod\ p)$.

[1] Số nguyên tố p được lấy từ `https://asecuritysite.com/encryption/getprimen`

Về vấn đề an toàn trong giao thức trao đổi khoá cũng như về bài toán **DLP** nêu trên, nhóm xin được phép trình bày trong phần **Phụ lục.**

3.3 Tạo khoá ma trận

Nhóm sử dụng mã ma trận cho việc mã hoá và giải mã được diễn ra nhanh chóng → cần đưa khoá bí mật đã trao đổi ở phần trên thành 1 khoá ma trận có kích thước $12 * 12$ (mod 251)

- Lý do chọn ma trận vuông có bậc là 12 vì nhóm chọn số nguyên tố 1024-bits → khả năng cao khoá chung có xấp xỉ 309 chữ số ở hệ thập phân, do lấy mod 251 ở mỗi phần tử nên ta sẽ lấy 1 lần 2 chữ số để không cần sử dụng phép mod → sinh ra được tổng cộng 155 phần tử. Đồng thời, ta cần sinh khoá ma trận với tổng cộng $12 * 2$ (các phần tử trên đường chéo) + $11 * 12$ (các phần tử ở ma trận tam giác trên và ma trận tam giác dưới) = $12 * 13 = 156$ phần tử vừa sát với 155 nên ta sẽ chọn bậc của ma trận là 12 (những vị trí hết số để gán giá trị thì sẽ được mặc định gán giá trị 1)
- Nhóm sử dụng modulo 251 vì đây là số nguyên tố nên nó sẽ nguyên tố cùng nhau với tất cả các số nhỏ hơn nó. *Tính chất này sẽ được sử dụng trong phần sinh khoá ma trận thoả khoá khả nghịch với modulo 251.* Đồng thời nhóm cũng làm trên **ASCII** nên sẽ lấy số nguyên tố lớn nhất nhỏ hơn 255 là 251

Định nghĩa mã ma trận

- ***Gen(n, p)*** sinh ra khoá ma trận K (mod p) khả nghịch có kích thước $n * n$
- ***Enc(K, M)*** sinh ra bản mã C là 1 vector thoả $C = K * M$ (mod p) và gửi C trên kênh truyền
- ***Dec(K, C)*** sinh ra tin nhắn M là 1 vector thoả $M = K^{-1} * C$ (mod p)
- Khoá K được sử dụng trong quá trình mã hoá và giải mã thì chỉ có Alice và Bob biết được → khoá bí mật.

Thuật toán sinh khoá trong mã ma trận

1. Tạo ma trận **L** là ma trận tam giác dưới thoả $gcd(det(L), p) = 1$
2. Tạo ma trận **U** là ma trận tam giác trên thoả $gcd(det(U), p) = 1$
3. Trả ra khoá **K** = **L** * **U**

Lúc này, ta tiến hành sinh khoá ma trận **K** bằng khoá được trao đổi ở giao thức **Diffie−Hellman** như sau

- Khởi tạo 2 ma trận **L** và **U** có kích thước $12 * 12$ và có tất cả phần tử mang giá trị 0
- Tạo các phần tử nằm trên đường chéo của ma trận **L** và **U** thoả những phần tử đó nguyên tố cùng nhau với p (mà ở đây $p = 251$). Do vậy, ma trận mà ta sinh được sẽ có định thức luôn nguyên tố cùng nhau với p
  - Nhóm sử dụng $p = 251$ nên việc lấy 2 chữ số luôn đảm bảo phần tử đó nguyên tố cùng nhau với 251
  - Theo như thực nghiệm thì độ dài khoá bí mật rất dài (khả năng lớn hơn 50 chữ số rất cao) nên khả năng dính vào trường hợp không còn chữ số nào trong việc sinh phần tử trên đường chéo là rất thấp (việc không còn chữ số nào trong khoá bí mật sẽ dẫn đến tồn tại một vài phần tử trong đường chéo mang giá trị 0 → không thoả được điều kiện $gcd(det(L), p) = 1$)

- Với giá trị còn lại của khoá bí mật, tiến hành đưa các giá trị đó vào các phần tử phù hợp với định nghĩa của ma trận tam giác trên **U** và ma trận tam giác dưới **L**. Nếu đã hết giá trị thì các phần tử đó sẽ được gán giá trị 1
- Cuối cùng, khoá $\mathbf{K} = \mathbf{L} * \mathbf{U}$ là khoá sẽ được sử dụng trong mã ma trận
- Chương trình sinh khoá trên không mang yếu tố ngẫu nhiên nên việc thực hiện chương trình này giúp cho cả 2 bên Alice và Bob vẫn có được khoá ma trận bí mật chung mà không bị lộ một vài thông tin của khoá ra ngoài

3.4 Cơ chế xác thực tin nhắn – Digital Signature

Nhóm sử dụng phương pháp tạo chữ kí cho 1 tin nhắn nào đó dựa trên ý tưởng của hệ mã RSA

- ***Gen(k)*** tạo ra cặp khoá (e, d) với độ dài của khoá là k
  - e là khoá được công khai trên kênh truyền, còn d là khoá bí mật chỉ có người tạo ra biết được
- ***Sign(d, m)*** dùng để tạo ra chữ kí $\sigma = m^d \ (mod\ n)$ và gửi trên kênh truyền cặp giá trị (c, σ)
- ***Verify(e, c, σ)*** dùng để kiểm tra $\sigma == m^e \ (mod\ n)$ hay không (người bên phía chứng nhận sẽ giải mã c ra m rồi tiến hành kiểm tra chữ kí)

3.5 Quá trình thực hiện của hệ thống

1. Alice và Bob lấy thông tin của các thông số được công khai trên kênh truyền từ file `parameter.txt` bao gồm p và g
2. Thiết lập kết nối để giao tiếp với nhau bằng **socket**
3. Alice và Bob gửi các giá trị eA, eB, nA, nB lên kênh truyền, hỗ trợ cho việc xác thực tin nhắn bằng chữ kí số. Các giá trị trên sẽ bị kẻ tấn công nhìn thấy. Đồng thời ta cũng có thể xem những giá trị này thông qua file `public_msg.txt`
4. Alice và Bob thống nhất 1 khoá bí mật sử dụng chung thông qua giao thức **Diffie–Hellman**
5. Tạo khoá **K** là ma trận khả nghịch cấp 12 từ khoá bí mật đã thống nhất ở bước 4, đồng thời cũng tính luôn $\mathbf{K}^{-1}$ cho quá trình giải mã diễn ra nhanh chóng
6. Alice và Bob bắt đầu giao tiếp trên kênh truyền. Mỗi lần gửi tin nhắn thì hệ thống sẽ gửi 2 giá trị, giá trị thứ nhất là bản mã c được mã hoá từ tin nhắn m và giá trị thứ hai là chữ kí σ của tin nhắn m
   (a) Do tin nhắn có thể rất dài nên ta sử dụng **hàm băm (hash function)** để đưa về 1 chuỗi nhỏ hơn và xử lý trên chuỗi đã được băm (sử dụng hàm băm **SHA256**)
   (b) Ta chỉ sử dụng khoá **K** 10 lần. Sau lần thứ 10 ta phải thiết lập khoá mới để tránh kẻ tấn công có thể thu thập đủ cặp (**M**, **C**) và giải được khoá **K** bằng phương pháp **Gauss**

PHỤ LỤC

A Tìm phần tử sinh trong $\mathbb{Z}_|^*$

Theo định nghĩa, phần tử sinh g trong nhóm Z_p^* là phần tử thoả $ord(g) = ord(Z_p^*) = p - 1$

Đồng thời theo *định nghĩa bậc của phần tử* thì $ord(g) = p - 1 \Leftrightarrow g^{p-1} = 1 \ (mod\ p)$ và không tồn tại i với $0 < i < p - 1$ sao cho $g^i = 1 \ (mod\ p)$

Do p là số nguyên tố lớn nên p lẻ $\rightarrow p = 2k + 1 \rightarrow g^{2k} = 1 \ (mod\ p) \rightarrow g^k = \pm 1 \ (mod\ p)$

Xét trường hợp $g^k = 1 \ (mod\ p)$ thì không phù hợp với *định nghĩa của phần tử sinh* khi tồn tại $k < p - 1$ thoả $g^k = 1 \ (mod\ p) \Rightarrow$ phần tử sinh g sẽ thoả mãn $g^k = -1 \ (mod\ p)$

Vậy cách để tìm phần tử sinh là đầu tiên cho $g = 2$ và kiểm tra phương trình $g^{\frac{p-1}{2}} = -1 \ (mod\ p)$. Nếu phương trình trên thoả thì g là phần tử sinh, còn không thì $g = g + 1$ và tiếp tục kiểm tra.

B Bài toán logarit rời rạc (DLP)

Cho số nguyên tố p, g là 1 căn nguyên thuỷ của Z_p^*. Với $p \nmid h$, bài toán **DLP** yêu cầu tìm $x \in Z_p^*$ thoả $g^x = h \ (mod\ p)$. Bài toán này được xem là một bài toán khó bởi vì chưa có thuật toán nào hiệu quả (chạy trong thời gian đa thức) để giải được.

C An toàn của giao thức trao đổi khoá Diffie−Hellman

Để nói về độ an toàn của giao thức trao đổi khoá **Diffie−Hellman**, các nhà mật mã học đã đưa ra 2 giả định sau đây.

- *Giả định Diffie−Hellman dạng tính toán (Computational Diffie−Hellman assumption)* nói rằng khi ta có $g^a \ (mod\ p)$ và $g^b \ (mod\ p)$ thì không thể tính được $g^{ab} \ (mod\ p)$ trong thời gian đa thức.
- Tuy nhiên, ta lại thường xét tới một giả định yếu hơn nhưng là vừa đủ để đánh giá giao thức trao đổi khoá **Diffie−Hellman** là an toàn. *Giả định Diffie−Hellman dạng quyết định (Decisional Diffie−Hellman assumption)* nói rằng đưa cho mình 2 cặp ($g^a \ mod\ p$, $g^b \ mod\ p$, $g^{ab} \ mod\ p$) và ($g^a \ mod\ p$, $g^b \ mod\ p$, $g^c \ mod\ p$ với c là số ngẫu nhiên thuộc $\mathbb{Z}_|^*$) thì kẻ tấn công không thể phân biệt được 2 cặp này trong thời gian đa thức.

D Tóm tắt một số kiến thức về ma trận

Định lý 1: Cho **A** là ma trận vuông cấp n. Nếu **A** là ma trận tam giác trên hay ma trận tam giác dưới thì định thức của ma trận **A** được tính bằng tích của các phần tử trên đường chéo.

$$det(A) = \prod_{i=1}^{n} a_{i,i}$$

Định lý 2: Cho **A** là ma trận vuông cấp n. **A** là ma trận khả nghịch khi và chỉ khi định thức của **A** khác 0 $\rightarrow$ Nếu **A** là ma trận vuông cấp n có các phần tử thuộc Z_p^* thì **A** khả nghịch trong modulo p khi và chỉ khi $gcd(det(A), p) = 1$

Định lý 3: Cho **A**, **B**, **C** là các ma trận vuông cấp n. Nếu **C** = **A** * **B** thì det(**C**) = det(**A**) * det(**B**)

E Hệ mã khoá công khai RSA

Cho p và q là 2 số nguyên tố lớn khác nhau. Ta tính thêm các thông số sau: $n = pq$, $\phi(n) = (p-1)(q-1)$, $e \xleftarrow{\$} Z_n^*$, $d \in Z_n^*$ thoả $ed \equiv 1 \ (mod\ \phi(n))$. Giữ p, q, $\phi(n)$, d là các thông số bí mật và công khai n, e lên kênh truyền.

- ***Gen(k)*** sinh ra khoá công khai e, n và khoá bí mật d có độ dài k
- ***Enc(n, e, m)*** tính $c = m^e \ (mod\ n)$ và gửi c trên kênh truyền
- ***Dec(n, d, c)*** tính $m = c^d \ (mod\ n)$

F Hàm băm (hash function)

Định nghĩa: Hàm băm (Hash function) là hàm $h : \{0,1\}^* \to \{0,1\}^n$ (đưa chuỗi có độ dài bất kì về đúng độ dài n, chuỗi đầu vào có thể dài hơn hoặc ngắn hơn n) thoả các tính chất sau.

- **Tính một chiều (One way function):** khi có x thì ta có thể tính $h(x)$ trong thời gian đa thức, nhưng khi có $h(x)$ thì không tồn tại thuật toán chạy trong thời gian đa thức có thể tìm ngược lại x
- **Tính kháng đụng độ:** (dùng để bảo vệ tính toàn vẹn của dữ liệu, 2 dữ liệu khác nhau sẽ có các giá trị băm khác nhau) $x \neq x' \Rightarrow h(x) \neq h(x')$
- **Tính kháng đụng độ 2:** (dùng để chống lại giả mạo tin nhắn) không tồn tại thuật toán trong thời gian đa thức có thể tìm được 1 cặp (x, x') với $x \neq x'$ thoả $h(x) = h(x')$