MỤC LỤC

LỜI CẢM ƠN

Em xin chân thành gửi lời cảm ơn tới các thầy cô của trường, các thầy cô trong Ban giám hiệu và thầy cô trong Bộ môn Tin học của trường Đại học Dân lập Hải Phòng đã tận tình giảng dạy, giúp đỡ và tạo mọi điều kiện cho chúng em trong suốt thời gian học tập tại trường.

Và em cũng xin gửi lời cảm ơn tới thầy Trịnh Nhật Tiến – Giáo viên hướng dẫn - đã tận tình, hết lòng hướng dẫn em trong suốt quá trình nghiên cứu để hoàn thành đồ án tốt nghiệp này. Em mong thầy luôn luôn mạnh khoẻ để nghiên cứu và giảng dạy, đào tạo nguồn nhân lực cho đất nước.

Một lần nữa em xin chân thành cảm ơn.

Hải Phòng, ngày ..... tháng ..... năm 2011

Sinh viên thực hiện

Phạm Thị Phượng

Chương 1. MỘT SỐ KHÁI NIỆM CƠ BẢN

1.1. CÁC KHÁI NIỆM TRONG TOÁN HỌC

1.1.1. Khái niệm số nguyên tố

Số nguyên tố là số tự nhiên lớn hơn 1 và chỉ có hai ước là 1 và chính nó.

Ví dụ: Các số 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31,37 ,43... là các số nguyên tố. Trong đó số 2 là số nguyên tố chẵn duy nhất.

Số nguyên tố có vai trò và ý nghĩa to lớn trong số học và lý thuyết mật mã. Bài toán kiểm tra tính nguyên tố của một số nguyên dương n và phân tích một số n ra thừa số nguyên tố là các bài toán rất được quan tâm.

1.1.2. Khái niệm số nguyên tố cùng nhau

Một ước chung d >0 của các số nguyên a_1, a_2, ...a_n, trong đó mọi ước chung của a_1, a_2,...a_n đều là ước của d, thì d được gọi là ước chung lớn nhất (UCLN) của a_1, a_2 , ...a_n . Kí hiệu d = bgd(a_1, a_2, ...a_n) hay d= UCLN(a_1, a_2, ...a_n).

Nếu gcd (a_1, a_2 ...a_n)=1,thì các số a_1, a_2,...a_n được gọi là số nguyên tố cùng nhau.

Ví dụ: Hai số 8 và 13 là hai số nguyên tố cùng nhau vì có gcd (8,13) =1

1.1.3. Một số khái niệm trong đại số

1/. Khái niệm Nhóm:

Nhóm là bộ các phần tử (*G*, *) thỏa mãn các tính chất:

+ Kết hợp: ($x * y$) $* z = x * (y * z)$ với mọi x, y, z $\in G$

+ Tồn tại phần tử trung lập $e \in G$: $e * x = x * e = x$, $\forall x \in G$

+ Tồn tại phần tử nghịch đảo $x' \in G$: $x' * x = x * x' = e$

2/. Khái niệm Nhóm con:

Nhóm con của G là tập $S \subset G$, $S \neq \phi$ thỏa mãn các tính chất sau:

+Phần tử trung lập *e* của G nằm trong *S*

+ S khép kín đối với phép tính(*) trong G, tức là với mọi $x, y \in S$ thì $x * y \in S$

+ S khép kín đối với phép lấy nghịch đảo trong G, tức là $x^{-1} \in$ S với mọi x $\in$ S.

3/. Khái niệm Nhóm Cyclic:

G được gọi là nhóm Cyclic nếu tồn tại g $\in$ G sao cho mọi phần tử trong G đều là một luỹ thừa nguyên nào đó của g.

Ví dụ: Nhóm (Z^+, +) gồm các số nguyên dương là Cyclic với phần tử sinh g =1.

4/. Tập hợp thặng dư thu gọn theo modulo:

Kí hiệu Z_n^* ={ x$\in$ Z_n , x là nguyên tố cùng nhau với n}. Tức là x phải khác 0.

Z_n^* được gọi là tập thặng dư theo mod n có số phần tử là ϕ(n).

1.1.4. Một số khái niệm về độ phức tạp của thuật toán

1.1.4.1. Khái niệm bài toán

Bài toán được diễn đạt bằng hai phần:

Input: Các dữ liệu vào của bài toán.

Output: Các dữ liệu ra của bài toán(kết quả).

Không mất tính chất tổng quát của bài toán giả thiết các dữ liệu trong bài toán đều là số nguyên.

1.1.4.2. Khái niệm thuật toán

"Thuật toán" được hiểu đơn giản là cách thức để giải một bài toán. Cũng có thể được hiểu bằng hai quan niệm: Trực giác hay Hình thức như sau:

1/. Quan niệm trực giác về "thuật toán"

Một cách trực giác, thuật toán được hiểu là một dãy hữu hạn các qui tắc(chỉ thị, mệnh lệnh) mô tả một quá trình tính toán, để từ dữ liệu đã cho (Input) ta nhận được kết quả (Output) của bài toán.

2/. Quan niệm toán học về "thuật toán"

Một cách hình thức, người ta quan niệm thuật toán là một máy tính Turing. Thuật toán được chia thành hai loại: Đơn định và không đơn định.

Thuật toán đơn định (Deterministic): Là thuật toán mà kết quả của mọi phép toán đều được xác định duy nhất.

Thuật toán không đơn định (Nondeterministic): Là thuật toán có ít nhất một phép toán mà kết quả của nó là không duy nhất.

1.1.4.3. Hai mô hình tính toán

Hai quan niệm về thuật toán ứng với hai mô hình tính toán.

Ứng với hai mô hình tính toán có hai cách biểu diễn thuật toán.

1/. Mô hình ứng dụng: Thuật toán được biểu diễn bằng ngôn ngữ tựa Algol.

+ Đơn vị nhớ: Một ô nhớ chứa toàn bộ dữ liệu.

+ Đơn vị thời gian: Thời gian để thực hiện một phép tính cơ bản trong số học hay logic như cộng, trừ, nhân, chia.....

2/. Mô hình lý thuyết:

Thuật toán được biểu diễn bằng ngôn ngữ máy Turing.

+ Đơn vị nhớ: Một ô chứa một tín hiệu. Với mã nhị phân thì đơn vị nhớ là 1 bit.

+ Đơn vị thời gian: Thời gian để thực hiện một bước chuyển hình trạng.

1.1.4.4. Khái niệm độ phức tạp của thuật toán

1/. Chi phí của thuật toán (Tính theo một bộ dữ liệu đầu vào)

Chi phí phải trả cho một quá trình tính toán gồm chi phí về thời gian và bộ nhớ:

Chi phí thời gian của một quá trình tính toán là thời gian cần thiết để thực hiện một quá trình tính toán. Với thuật toán tựa Algol: chi phí thời gian là số các phép tính cơ bản thực hiện trong quá trình tính toán.

Chi phí bộ nhớ của một quá trình tính toán là số ô nhớ cần thiết để thực hiện môt quá trình tính toán.

Gọi A là một thụât toán, e là dữ liệu vào của bài toán đã được mã hoá bằng cách nào đó. Thuật toán A tính trên dữ liệu vào e phải trả một giá nhất định. Ta kí hiệu: $t_A(e)$ là giá thời gian và $I_A(e)$ là giá bộ nhớ.

2/. Độ phức tạp về bộ nhớ (Trong thường hợp xấu nhất)

$L_A(n)$ =max{ $I_A(e)$, với $|e| \leq n$}, n là "kích thuớc" đầu vào của thuật toán.

3/. Độ phức tạp thời gian (Trong trường hợp xấu nhất)

$T_A(n)$ = max{ $t_A(e)$, với $|e| \leq n$}.

4/. Độ phức tạp tiệm cận

Độ phức tạp PT(n) được gọi là tiệm cận tới hàm f(n), kí hiệu O(f(n)) nếu tồn tại các số n_0., c mà PT(n) $\leq$ c.f(n), $\forall n \geq n_0$.

5/.Độ phức tạp đa thức

Độ phức tạp PT(n) được gọi là đa thức, nếu nó tiệm cận tới đa thức p(n).

6/. Thuật toán đa thức

Thuật toán được gọi là đa thức, nếu độ phức tạp về thời gian(trong trường hợp xấu nhất) của nó là đa thức.

1.2. HỆ MÃ HOÁ

1.2.1. Khái niệm mã hoá dữ liệu

Để đảm bảo được an toàn thông tin lưu trữ trong máy tính (giữ gìn thông tin cố định) hay đảm bảo an toàn thông tin trên đường truyền tin (trên mạng máy tính), người ta phải "che giấu" các thông tin này.

"Che" thông tin (dữ lệu) hay còn gọi là "mã hoá" thông tin là thay đổi hình dạng thông tin gốc, và người khác khó nhận ra.

"Giấu" thông tin (dữ liệu) là cất giấu thông tin trong bản tin khác, và người khác khó nhận ra.

1/. Hệ mã hoá

Việc mã hoá phải theo nguyên tắc nhất định, quy tắc đó gọi là Hệ mã hoá.

Hệ mã hoá được định nghĩa là một bộ năm (P,C,K,E,D) trong đó:

P: tập hữu hạn các bản rõ có thể.

C: tập hữu hạn các bản mã có thể.

K: tập hữu hạn các khoá có thể.

E: tập các hàm lập mã.

D: là tập các hàm giải mã.

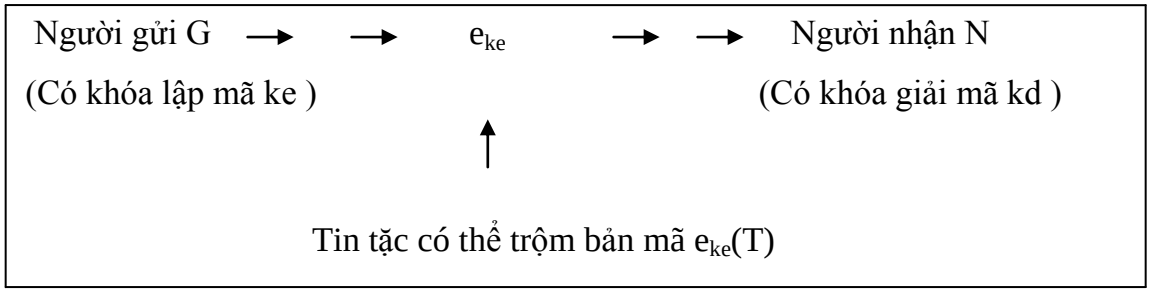
Với khóa lập mã ke $\in$ K, có hàm lập mã $e_{ke} \in E$, $e_{ke}: P \rightarrow C$,

Với khoá giải mã kd $\in$ K, có hàm lập mã $e_{kd} \in D$, $e_{ke}: C \rightarrow P$,

sao cho $d_{kd}(e_{ke}(x))=x, \quad \forall x \in P.$

Ở đây x được gọi là bản rõ, $e_{ke}(x)$ được gọi là bản mã.

2/. Mã hoá và giải mã



Người gửi G muốn bán tin T cho người nhận N. Để bảo đảm bí mật, G mã hoá bản tin bằng khoá lập mã ke, nhận được bản mã $e_{ke}(T)$, sau đó gửi cho N. Tin tặc có thể trộm bản mã $e_{ke}(T)$, nhưng cũng "khó" hiểu được bản tin gốc T nếu không có khoá giải mã kd.

Người nhận N nhận được bản mã, họ dùng khoá giải mã kd, để giải mã $e_{ke}(T)$, sẽ nhận được bản tin gốc $T = d_{kd}(e_{ke}(T))$.

1.2.2. Phân loại hệ mã hoá

Người ta chia làm hai loại Hệ mã hóa chính đó là: Hệ mã hoá khoá đối xứng (hay Hệ mã hóa khóa bí mật) và Hệ mã hoá khóa bất đói xứng (hay Hệ mã hóa khoá công khai).

1.2.2.1. Hệ mã hoá khoá đối xứng

Hệ mã hoá khoá đối xứng là Hệ mã hoá khoá mà biết được khoá lập mã thì có thể "dễ" tính được khoá giải mã và ngược lại. Đặc biệt một số Hệ mã hoá có khoá lập mã và khoá giải mã trùng nhau (ke =kd), như Hệ mã hoá "dịch chuyển" hay DES.

Hệ mã hoá khoá đối xứng còn gọi là Hệ mã hoá khoá bí mật, hay khoá riêng, vì phải giữ bí mật cả hai khoá. Trước khi dùng Hệ mã hoá khoá đối xứng, người ta gửi và nhận phải thoả thuận thuật toán mã hoá và khoá chung (lập mã hay giải mã), khoá phải được giữ bí mật.

Độ an toàn của khoá này phụ thuộc vào khoá.

Ví dụ

+ Hệ mã hoá cổ điển là Mã hoá khoá đối xứng: dễ hiểu, dễ thực thi, nhưng có độ an toàn không cao. Vì giới hạn tính toán chỉ trong phạm vi bảng chữ cái, sử dụng trong bản tin cần mã, ví dụ là Z_{26} nếu dùng các chữ cái tiếng Anh. Với hệ mã hoá cổ điển, nếu biết khoá lập mã hay thuật toán lập mã, có thể "dễ" xác định được bản rõ, vì "dễ" tìm được khoá giải mã.

+ Hệ mã hoá DES (1973) là Mã hoá khoá đối xứng hiện đại, có độ an toàn cao.

a). Đặc điểm của Hệ mã hoá khoá đối xứng

Ưu điểm:

Hệ mã hoá khoá đối xứng mã hoá và giải mã nhanh hơn Hệ mã hoá khoá công khai.

Han chế:

1/. Mã hoá khoá đối xứng chưa thật an toàn với lý do sau

Người mã hoá và người giải mã phải có "chung"một khoá. Khoá phải được giữ bí mật tuyệt đối, vì biết khoá này "dễ" xác định được khoá kia và ngược lại.

2/. Vấn đề thoả thuận khoá và quản lý khoá chung là khó khăn và phức tạp, Người gửi và người nhận phải luôn thống nhất với nhau về khoá. Việc thay đổi khoá là rất khó và dễ bị lộ. Khoá chung phải được gửi cho nhau trên kênh an toàn.

Mặt khác khi hai người (lập mã, giải mã) cũng biêt "chung" một bí mật, thì càng khó giữ được bí mật!

b). Nơi sử dụng Hệ mã hoá khoá đối xứng

Hệ mã hoá khoá đối xứng thường được sử dụng trong một môi trường chung có thể dễ dàng trao chuyển bí mật, chẳng hạn trong cùng một mạng nội mạng nội bộ, Hệ mã hoá khoá đối xứng thường dùng để mã hoá những bản tin lớn, vì tốc độ mã hoá và giải mã nhanh hơn Hệ mã hoá khoá công khai.

1.2.2.2. Hệ mã hoá khoá công khai

Hệ mã hoá khoá phi đối xứng là Hệ mã hoá có khoá lập mã và khoá giải mã khác nhau (ke ≠ kd) biết được khoá này cũng "khó" tính được khoá kia.

Hệ mã hoá này còn được gọi là Hệ mã hoá khoá công khai, vì:

Khoá lập mã cho công khai, còn gọi là khoá công khai (Public key).

Khoá giải mã giữ bí mật, còn gọi là khoá riêng (Private key) hay khoá bí mật.

Một người bất kì có thể dùng khoá công khai để mã hoá bản tin, nhưng chỉ người nào có đúng giải mã thì mới có khả năng đọc được bản rõ.

Hệ mã hoá khoá công khai hay Hệ mã hoá khoá đối xứng do Diffie và Hellman phát minh vào những năm 1970.

a). Đặc điểm của Hệ mã hoá khoá công khai

Ưu điểm:

Thuật toán được viết một lần, công khai cho nhiều lần dùng, cho nhiều người dùng, họ chỉ cần giữ bí mật khoá riêng của mình.

Khi biết các tham số ban đầu của hệ mã hoá, việc tính ra cặp khoá công khai và bí mật phải là "dễ", tức là trong thời gian đa thức.

Người gửi có bản rõ là P và khoá công khai, thì "dễ" tạo ra bản mã C.

Người nhận có bản mã C và khoá bí mật, thì "dễ" giải được thành bản rõ P.

Người mã hoá dùng khoá công khai, người giải mã giữ khoá bí mật. Khả năng lộ khoá bí mật khó hơn vì chỉ có một người giữ gìn.

Nếu thám mã biết khoá công khai, cố gắng tìm khoá bí mật, thì chúng phải đương đầu với bái toán "khó".

Nếu thám mã biết khoá công khai và bản mã C, thì việc tìm ra bản rõ P cũng là bài toán "khó", số phép thử là vô cùng lớn, không khả thi.

Hạn chế

Hệ mã hoá khoá công khai: mã hoá và giải mã chậm hơn hệ mã hoá khoá đối xứng.

b). Nơi sử dụng Hệ mã hoá khoá công khai

Hệ mã hoá khoá công khai thường được sử dụng chủ yếu trên các mạng công khai như Internet, khi mà việc trao chuyển khoá bí mật tương đối khó khăn.

Đặc trưng nổi bật của hệ mã hoá công khai là khoá công khai (public key) bản mã (ciphertext) đều có thể gửi trên một kênh truyền tin không an toàn. Có biết cả khoá công khai và bản mã, thì thám mã cũng không dễ khám phá được bản rõ.

Nhưng vì tốc độ mã hoá và giải mã chậm, nên hệ mã hoá khoá công khai chỉ dùng để mã hoá những bản tin ngắn, ví dụ như mã hoá bí mật gửi đi.

Hệ mã hoá khoá công khai thường được sử dụng cho cặp người dùng thoả thuận khoá bí mật của Hệ mã hoá khoá riêng.

1.2.3. Hệ mã hoá đối xứng cổ điển

Khái niệm:

Hệ mã hoá đối xứng đã được dùng từ rất sớm, nên còn gọi là hệ mã hoá đối xứng - cổ điển (gọi ngắn gọn là Hệ mã hoá đối xứng cổ điển).

Lập mã: thực hiện theo các bước sau:

1/. Nhập bản rõ kí tự: RÕ_CHỮ.

2/. Chuyển RÕ_CHỮ ==> RÕ_SỐ.

3/. Chuyển RÕ_SỐ ==> MÃ_SỐ

4/. Chuyển MÃ_SỐ ==> MÃ_CHỮ.

Giải mã: Thực hiện theo các bước sau:

1/. Nhập bản mã kí tự: MÃ_CHỮ

2/. Chuyển MÃ_CHỮ ==> MÃ_SỐ.

3/. Chuyển MÃ_SỐ ==> RÕ_SỐ

4/. Chuyển RÕ_SỐ ==> RÕ_CHỮ.

Để chuyển từ CHỮ sang SỐ hay ngược lại từ SỐ về CHỮ, người ta theo một qui ước nào đó, ví dụ chữ cái thay bằng số theo modulo 26 như sau:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	X	Y	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	23	24	25	26

Để thực hiện mã hoá hay giải mã với các "số", người ta dùng các phép toán số học theo modulo 26.

<u>Các hệ mã hoá cổ điển</u>

Mã hoá cổ điển gồm nhiều hệ, ví dụ:

Hệ mã hoá dịch chuyển: Khoá có "chìa". (Thể hiện bằng 1 giá trị).

Hệ mã hoá Affine: Khoá có 2 "chìa". (Thể hiện bằng 2 giá trị).

Hệ mã hoá thay thế: Khoá có 26 "chìa". (Thể hiện bằng 16 giá trị).

Hệ mã hoá VIGENERE: Khoá có m "chìa". (Thể hiện bằng m giá trị).

Hệ mã hoá HILL: Khoá có ma trận "chìa". (Chùm chìa khoá).

1.2.3.1. Hệ mã hoá dịch chuyển

Sơ đồ:

Đặt P = C = K = Z_{26} . Bản mã y và bản rõ x $\in$ Z_{26}.

Với khoá k $\in$ K ta định nghĩa:

Hàm mã hoá: $y = e_k(x) = (x+k) \bmod 26$

Hàm giải mã: $y = d_k(y) = (y-k) \bmod 26$

Độ an toàn: Độ an toàn của mã dịch chuyển rất thấp.

Tập khoá K chỉ có 26 khoá, nên việc phá khóa (thám mã) có thể thực hiện được dễ dàng bằng cách thử kiểm tra từng khoá: k = 1, 2, 3, 4,..., 26.

1.2.3.2. Hệ mã hoá Thay thế (Hoán vị toàn cục)

Sơ đồ

Đặt P = C = Z_{26} , Bản mã y và bản rõ x $\in$ Z_{26} .

Tập khoá K là tập mọi hoán vị trên Z_{26}, ta định nghĩa:

Mã hoá: $y = e_\pi(x) = \pi(x)$

Giải mã: $x = d_\pi(y) = \pi^{-1}(y)$

Độ an toàn Độ an toàn của mã thay thế: thuộc loại cao.

Tập khoá K có 26! khoá (> 4.10^{26}), nên việc phá khoá (thám mã) có thể thực hiện bằng cách duyệt tuần tự 26! khoá, tốn rất nhiều thời gian!

Hiện nay với hệ mã này, người ta có phương pháp thám mã khác nhanh hơn.

1.2.3.3. Hệ mã hoá AFFINE

Sơ đồ

Đặt P = C = X_{26} . Bản mã y và bản rõ x $\in$ Z_{26}

Tập khoá K = {(a,b), với a, b $\in$ Z_{26} , UCLN(a,26) = 1}

Với khoá k = (a,b) $\in$ K, ta định nghĩa:

Phép mã hoá $y = e_k(x) = (a\,x + b) \bmod 26$

Phép giải mã $x = d_k(y) = a^{-1}(y-b) \bmod 26$

Độ an toàn : Độ an toàn của hệ mã hoá Affine là rất thấp.

+ Điều kiện UCLN(a,26) = 1 để đảm bảo a có phần tử nghịch đảo a^{-1} mod 26, tức là thuật toán giải mã d_k luôn thực hiện được.

+ Số lượng a$\in$ Z_{26} nguyên tố với 26 là (26) = 12, đó là:

1, 3, 5, 7, 9, 11, 15, 17, 19, 21,23, 25

Các số nghịch đảo theo(mod) 26 tương ứng:1, 9, 21, 15, 3, 19, 7, 23,11, 5, 17, 25

+ Số lượng b $\in$ Z_{26} là 26.

+ Số các khóa (a,b) có thể là 12*26 =312. Rất ít!

Như vậy việc dò khoá mật rất dễ dàng.

1.2.3.4. Hệ mã hoá VIGENERE

Sơ đồ

Đặt P = C = K =$(Z_{26})^m$, m là số nguyên dương, các phép toán thực hiện trong Z_{26}.

Bản mã Y và bản rõ X $\in$(Z_{26})^m. Khoá k = (k_1, k_2,...,k_m) gồm m phần tử.

Mã hóa Y = (y_1, y_2,...,y_m) = e_k(x_1, x_2,,...x_m) = (x_1 +k_1, x_2 +k_2 ,...x_m+ k_m) mod m.

Giải mã X = (x_1, x_2, ...,x_m) = d_k(y_1, y_2,...,y_m) = (y_1 - k_1, y_2 - k_2,...,y_m- k_m) mod m.

Độ an toàn: Độ an toàn của mã VIGENERE là tương đối cao.

Nếu khoá gồm m kí tự khác nhau, mỗi kí tự có thể được ánh xạ vào 1 trong m kí tự có thể, do đó hệ mật này được gọi là hệ thay thế đa biểu.

Như vậy số khoá (độ dài m)có thể có trong mật Vigenere là 26^m.

Nếu dùng phương pháp "tấn công vét cạn", thám mã phải kiểm tra 26^m khoá.

Hiện nay với hệ mã này, người ta có phương pháp thám mã khác nhanh hơn.

1.2.3.5. Hệ mã hoá Hoán vị cục bộ

Sơ đồ

Đặt P = C = Z_{26}^m , m là số nguyên dương. Bản mã Y và bản rõ X $(Z_{26})^m$.

Tập khoá K là tập tất cả các hoán vị của {1, 2, ..., m}.

Với mỗi khoá k = $\pi \in$ K, k = (k_1, k_2,....,k_m) gồm m phần tử, ta định nghĩa:

Mã hoá Y = (y_1, y_2,...y_m) = e_k(x_1, x_2,...x_m) = ($x_{k(1)}$, $x_{k(2)}$,....$x_{k(m)}$)

Giải mã X = (x_1,x_2,....x_m) = d_k(y_1,y_2,...y_m) = ($y_{k(1)}^{-1}$,$y_{k(2)}^{-1}$,...,$y_{k(m)}^{-1}$)

Trong đó $k^{-1} = \pi^{-1}$ là hoán vị ngược của μ.

Độ an toàn:

Nếu dùng phương pháp "tấn công vét cạn", thám mã phải kiểm tra số khoá có thể là: 1!+ 2!+3!+...+m! trong đó m $\leq$ 26.

Hiện nay với hệ mã này, người ta có phương pháp thám mã khác nhanh hơn.

1.2.3.6. Hệ mã hoá HILL

Sơ đồ:

Đặt P = C = Z_{26}^{m} , m là số nguyên dương. Bản mã Y và bản rõ X $\in$ $(Z_{26})^{m}$

Tập khoá K = {K $\in$ Z_{26}^{m*m} det (K,26) = 1}. (K phải có K^{-1}).

Mỗi khoá k là môt "chùm chìa khoá" (một ma trận "các chìa khoá").

Với mỗi k $\in$ K định nghĩa:

Hàm lập mã: $Y = (y_1,y_2,...,y_m) = e_k(x_1,x_2,...,x_m) = (x_1,x_2,...,x_m)*K$

Hàm giải mã: $X = (x_1,x_2,...x_m) = = d_k(y_1,y_2,...y_m) = (y_1,y_2,...,y_m)*K$

Độ an toàn:

Nếu phương pháp "tấn công vét cạn", thám mã phải kiểm tra số khoá có thể với m lần lượt là 2,3,4,..trong đó m lớn nhất bằng độ dài bản rõ.

1.2.4. Hệ mã hoá công khai

1.2.4.1. Hệ mã hoá RSA

Sơ đồ:

* Tạo cặp khoá (bí mật, công khai) (a,b):

Chọn bí mật số nguyên lớn p, q, tính n = p*q, công khai n, đặt P = C = Z_n.

Tính bí mật (n) = (p-1).(q-1). Chọn khoá công khai b < (n), nguyên tố với (n).

Khoá bí mật a là phần tử nghịch đảo của b theo mod (n): a*b $\equiv$ 1(mod (n)).

Tập cặp khoá (bí mật, công khai) K = {(a,b)/a,b $\in$ Z_n, a*b$\equiv$ 1}(mod ϕ(n)).

Với bản rõ x$\in$ P và bản mã y$\in$ C, định nghĩa:

Hàm mã hoá: $y = e_k(x) = x^b \bmod n$

Hàm giải mã: $x = d_k(y) = y^a \bmod n$

Độ an toàn:

1/. Hệ mã hoá RSA là tất định, tức là với bản rõ **x** và một khoá bí mật **a**, thì chỉ có một bản mã **y**.

2/. Hệ mật RSA an toàn, khi giữ được bí mật khoá giải mã **a**, **p**, **q**, (**n**).

Nếu biết được **p** và **q**, thì thám mã dễ dàng tính được (**n**) = (q-1)*(p-1).

Nếu biết được (**n**), thì thám mã sẽ tính được **a** theo thuật toán Eulide mở rộng.

Nhưng phân tích **n** thành tích của **p** và **q** là bài toán "khó".

Độ an toàn của hệ mật RSA dựa vào khả năng giải bài toán phân tích số nguyên dương **n** thành tích của 2 số nguyên tố lớn **p** và **q**.

1.2.4.2. Hệ mã hoá Elgamal

Sơ đồ:

**** Tạo cặp khoá (bí mật, công khai) (a,h):***

Chọn số nguyên tố p sao cho bài toán logarit rời rạc trong Z_p là "khó "giải.

Chọn phần tử nguyên thuỷ $g \in Z_p^*$. Đặt $P = Z_p^*$, $C = Z_p^* \times Z_p^*$

Chọn khoá bí mật là $a \in Z_p^*$. Tính khoá công khai $h \equiv g^a \bmod p$.

Định nghĩa tập khoá: $K = \{(p, g, a, h): h \equiv g^a \bmod p\}$

Các giá trị p, g, h được công khai, phải giữ bí mật a.

Với bản rõ $x \in P$ và bản $y \in C$, với khoá $k \in K$ định nghĩa:

*** *Lập mã***: Chọn ngẫu nhiện bí mật $r \in Z_{p-1}$, bản mã là $y = e_k(x,r) = (y_1,y_2)$

Trong đó $y_1 = g^r \bmod p$ và $y_2 = x*h^r \bmod p$.

*** *Giải mã***: $d_k(y_1,y_2) = y_2(y_1^a)^{-1} \bmod p$

Độ an toàn

1/. Hệ mã hoá Elgamal là không tất định, tức là với một bản rõ x và 1 khoá bí mật a thì có thể có nhiều hơn một bản mã y, vì trong công thức lập mã còn có thành phần ngẫu nhiên **r**.

2/. Độ an toàn của hệ mật Elgamal dựa vào khả năng giải bài toán logarit rời rạc trong Z_p. Theo giả thiết trong sơ đồ, thì bài toán này phải "khó "giải.

Cụ thể như sau: Theo công thức lập mã: $y = e_k(x, r) = (y_1,y_2)$

Trong đó : $y_1 = g^r \bmod p$ và $y_2 = x*h^r \bmod p$.

Như vậy muốn xác định rõ bản **c** từ công thức y_2, thám mã phải biết được **r**. Giá trị này có thể tính được từ công thức y_1, nhưng lại gặp bài toán logarit rời rạc.

1.3. CHỮ KÝ SỐ

1.3.1. Giới thiệu về chữ ký số

Để chứng thực nguồn gốc hay hiệu lực của một tài liệu (ví dụ : đơn xin học, giấy báo nhập học, ...) lâu nay người ta dùng chữ kí "tay", ghi vào phía dưới của mỗi tài liệu. Như vậy người kí phải trực tiếp "kí tay" vào tài liệu.

Ngày nay các tài liệu được số hoá người ta cũng có nhu cầu chứng thực nguồn gốc hay hiệu lực của các tài liệu này. Rõ ràng không thể "kí tay "vào tài liệu, vì chúng không được in ấn trên giấy.

Tài liệu số (hay tài liệu "điện tử") là một xâu các bít (0 hay 1), xâu bít có thể rất dài(nếu in trên giấy có thể hàng nghìn trang). "Chữ kí" để chứng thực một xâu bít tài liệu cũng không thể là một xâu các bít nhỏ đặt phía dưới xâu bít tài liệu. Một "chữ kí" như vậy chắc chắn sẽ bị kẻ gian sao chép để đặt dưới một tài liệu khác bất hợp pháp.

Những năn 80 của thế kỉ 20, các nhà khoa học đã phát minh ra "chữ kí số" để chứng thực một "tài liệu số". Đó chính là "bản mã" của xâu bít tài liệu.

Người ta tạo ra "chữ kí số" (chữ kí điện tử) trên " tài liệu số" giống như tạo ra "bản mã"của tài liệu với "khoá lập mã".

Như vậy "kí số" trên " tài liệu số" là "kí" trên từng bít tài liệu. kẻ gian khó thể giả mạo "chữ kí số" nếu nó không biết "khoá lập mã".

Để kiểm tra một "chữ kí số" thuộc về một " tài liệu số", người ta giải mã "chữ kí số" bằng "khoá giải mã", và so sánh với tài liệu gốc.

Ngoài ý nghĩa để chứng thực nguồn gốc hay hiệu lực của các tài liệu số hoá, mặt mạnh của "chữ kí số" hơn "chữ kí tay" là ở chỗ người ta có thể "kí" vào tài liệu từ rất xa trên mạng công khai. Hơn thế nữa có thể "kí" bằng các thiết bị cầm tay (ví dụ: điện thoại di động) tại khắp mọi nơi miễn là kết nối được vào mạng, đỡ tốn thời gian, sức lực, chi phí.

"Kí số" thực hiện trên từng bít tài liệu, nên độ dài của "chữ kí số" ít nhất cũng bằng độ dài của tài liệu. Do đó thay vì kí trên tài liệu dài, người ta thường dùng "hàm băm" để tạo "đại diện" cho tài liệu, sau đó mới "kí số " lên "đại diện" này.

1.3.2. Sơ đồ chữ kí số

Sơ đồ chữ kí số là bộ năm (P, A, K, S,V) trong đó:

P là tập hợp các văn bản có thể.

A là tập hữu hạn các chữ kí có thể.

K là tập hữu hạn các khoá có thể.

S là tập các thuật toán kí.

V là tập các thuật toán kiểm thử.

Với mỗi khoá k $\in$ K, có thuật toán kí $Sig_k \in S$, Sig_k: P$\rightarrow$ A có thuật toán kiểm tra chữ kí $Ver_k \in V$, Ver_k : P $\times$ A{đúng, sai}, thoả mãn điều kiện sau với mọi x $\in$ P, y $\in$ A:

$$Ver_k(x,y) = \begin{cases} \text{Đúng, nếu } y = Sig_k(x) \\ \text{Sai, nếu } y \neq Sig_k(x) \end{cases}$$

Người ta dùng hệ mã hoã khoá công khai để lập " sơ đồ chữ kí số". Ở đây khoá bí mật **a** dùng làm khoá "kí", khoá công khai b làm khoá kiểm tra "chữ kí".

Ngược lại với việc mã hoá, dùng làm khoá công khai b để lập mật mã, dùng khoá bí mật **a** để giải mã. Điều này là hoàn toàn tự nhiên, vì "kí" cần giữ bí mật nên phải dùng khoá bí mật a để "kí" còn "chữ kí " là công khai cho mọi người biết nên họ dùng công khai b để kiểm tra.

1.3.3. Phân loại "chữ ký số"

Có nhiều loại chữ kí tuỳ theo cách phân loại, sau đây là một số cách:

Cách 1: Phân loại chữ kí theo đặc trưng kiểm tra chữ kí gồm có:

+ Chữ kí khôi phục thông điệp: Là loại chữ kí, trong đó người gửi chỉ cần "chữ kí", người nhận có thể khôi phục lại được thông điệp, đã được "kí" bởi "chữ kí" này.

Ví dụ: Chữ kí RSA là chữ kí khôi phục thông điệp.

+ Chữ kí không khôi phục thông điệp thông điệp: Là loại chữ kí, trong đó người gửi chỉ cần gửi "chữ kí", phải gửi kèm cả thông điệp đã được "kí" bởi chữ kí này. Ngược lại người nhận sẽ không có được thông điệp gốc.

Ví dụ: Chữ kí Elgamal là chữ kí đi kèm thông điệp.

Cách 2: Phân loại chữ kí theo mức an toàn gồm có:

1) Chữ kí "không thể phủ nhận": Nhằm tránh việc nhân bản chữ kí để sử dụng nhiều lần, tốt nhất là người gửi tham gia trực tiếp vào việc kiểm thử chữ kí. Điều đó được thực hiện bằng một giao thức kiểm thử, dưới dạng một giao thức mớii hỏi và trả lời.

Ví dụ:Chữ kí không phủ định (Chaum – van Antverpen).

2) Chữ kí "một lần":

Để đảm bảo an toàn, "Khoá kí" chỉ dùng một lần (one time) trên một tài liệu.

Ví dụ: Chữ kí một lần Lamport, chữ kí Fail – stop (Van Heyst & Pedersen).

Cách 3: Phân loại chữ kí theo ứng dụng đặc trưng gồm có:

Chữ kí "mù" (Blind Signature).

Chữ kí "nhóm" (Group Signature).

Chữ kí "bội" (Multy Signature).

Chữ kí "mù nhóm" (Blind Group Signature).

Chữ kí "mù bội" (Blind Multy Signature).

1.3.4. Chữ ký RSA

Sơ đồ chữ ký

*** Tạo cặp khóa (bí mật, công khai) (a,b):**

Chọn bí mật số nguyên tố lớn **p, q**, tính **n = p*q**, công khai **n**, đặt **P = C= Z_n**

Tính bí mật **ϕ(n) = (p-1).(q-1)**. Chọn khóa công khai **b < ϕ(n),** nguyên tố với **ϕ(n).**

Khóa bí mật a là phần tử nghịch đảo của b theo mod ϕ(n): a*b $\equiv$ 1(modϕ(n)).

Tập cặp khóa(bí mật, công khai):

K= { (a,b)/ a,b $\in$ Z_n, a * b $\equiv$ 1 (mod ϕ(n) **)}.**

*** Ký số:** Chữ ký trên **x**$\in$ ***P*** là **y= $Sig_k(x) = x^a$** (mod **n**), **y** $\in$ **A.** (R_1).

*** Kiểm tra chữ ký: Ver_k (x,y)** = đúng $\Leftrightarrow$ **x** $\equiv$ y^b (mod **n**). (R_2).

Chú ý:

- So sánh sơ đồ chữ ký RSA và sơ đồ mã hoá RSA ta thấy có sự tương ứng.
- Việc ký chẳng qua là mã hoá, việc kiểm thử lại chính là việc giải mã:

Việc "ký số" vào x tương ứng với việc "mã hoá" tài liệu **x**.

Kiểm thử chữ ký chính là việc giải mã "chữ ký", để kiểm tra xem tài liệu đã giải mã có đúng là tài liệu trước khi ký không. Thuật toán và khoá kiểm thử "chữ ký" là công khai, ai cũng có thể kiểm thử chữ ký được.

Ví dụ Ký trên **x = 2**

*** Tạo cặp khoá (bí mật, công khai) (a, b):**

Chọn số bí mật số nguyên tố **p=3, q=5**, tính **n=p*q=** 3*5= 15, công khai **n**.

Đặt **P= C= Z_n= Z_{15}**. Tính bí mật **ϕ(n) = (p-1).(q-1)** = 3*4= 8.

Chọn khoá công khai **b= 3** < **ϕ(n)** , số nguyên tố **ϕ(n)** =8

Khoá bí mật **a= 3**, là phần tử nghịch đảo của **b** theo mod **ϕ(n):** a*b $\equiv$ 1(modϕ(n)).

*** Ký số:** Chữ ký trên **x** =2 $\in$ ***P*** là

y= Sig_k(x) = x^a (mod **n**) = 2^3 (mod 15) = 8, **y** $\in$ **A.**

*** Kiểm tra chữ ký:** **Ver_k (x,y)** = đúng $\Leftrightarrow$ **x** $\equiv$ **y^b** (mod **n**) $\Leftrightarrow$ **2** $\equiv$ **8^3** (mod n)

1.3.5. Chữ kí ELGAMAL

1.3.5.1. Sơ đồ chữ kí Elgamal

*** Tạo cặp khoá (bí mật, công khai) (a, h)**

Chọn số nguyên tố **p** sao cho bài toán logarit rời rạc trong $\mathbf{Z_p}$ là "khó" giải.

Chọn phần tử nguyên thuỷ $\mathbf{g} \in \mathbf{Z_p}^*$. Đặt $\mathbf{P} = \mathbf{Z_p}^*$, $\mathbf{A} = \mathbf{Z_p}^* \times \mathbf{Z_{p-1}}^*$

Chọn khoá bí mật là $\mathbf{a} \in \mathbf{Z_p}^*$, Tính khoá công khai $\mathbf{h} \equiv \mathbf{g^a} \bmod \mathbf{p}$.

Định nghĩa tập khoá: $\mathbf{K} = \{(p, g, a, h): \mathbf{h} \equiv \mathbf{g^a} \bmod \mathbf{p}\}$.

Các giá trị **p, g, h** được công khai, phải giữ bí mật **a**.

***Kí số**

Dùng 2 khoá kí: khoá **a** và khoá ngẫu nhiên bí mật $\mathbf{r} \in \mathbf{Z_{p-1}}^*$

(Vì $\mathbf{r} \in \mathbf{Z_{p-1}}^*$, nên nguyên tố cùng **p-1**, do đó tồn tại $\mathbf{r^{-1}} \bmod \mathbf{(p-1)}$).

Chữ kí trên $\mathbf{x} \in \mathbf{P}$ là $\mathbf{y} = \mathbf{Sig_k}\ \mathbf{(x, r)} = \mathbf{(\gamma, \delta)}$, $\mathbf{y} \in \mathbf{A}$

Trong đó $\gamma \in \mathbf{Z_p}^*$, $\delta \in \mathbf{Z_{p-1}}$:

$$\gamma = \mathbf{g^r} \bmod \mathbf{p} \text{ và } \delta = \mathbf{(x - a^*)^* r^{-1}} \bmod \mathbf{(p-1)}$$

***Kiểm tra chữ kí:** $\mathbf{Ver_k(x, y, \delta)}$ = đúng $\leftrightarrow$ $\mathbf{h^{\gamma} * \gamma^{\delta}} \equiv \mathbf{g^x} \bmod \mathbf{p}$

Chú ý: Nếu chữ kí được kí đúng, kiểm thử sẽ thành công vì:

$$\mathbf{h^{\gamma} * \gamma^{\delta}} \equiv \mathbf{g^{a\gamma} * g^{r*\delta}} \bmod \mathbf{p} \equiv \mathbf{g^{(a\gamma + r*\delta)}} \bmod \mathbf{p} \equiv \mathbf{g^x} \bmod \mathbf{p}$$

Do $\delta = \mathbf{(x - a^*\gamma) * r^{-1}} \bmod (p-1)$ nên $\mathbf{(a^*\gamma + r^*\delta)} \equiv \mathbf{x} \bmod (p-1)$.

Ví dụ: Chữ ký Elgamal trên dữ liệu **x= 112**.

*** Tạo cặp khoá (bí mật, công khai) (a, h):**

Chọn số nguyên tố **p** = 463 . Đặt $\mathbf{P = Z_p^*}$, $\mathbf{A = Z_p^* \times Z_{p-1}^*}$

Chọn phần tử nguyên thuỷ **g**= 2 $\in \mathbf{Z_p^*}$.

Chọn khoá bí mật là **a**=211 $\in \mathbf{Z_p^*}$.

Tính khoá công khai $\mathbf{h \equiv g^a}$ mod $\mathbf{p = 2^{211}}$ mod 463 = 249.

Định nghĩa tập khoá: $\mathbf{K = \{(p, g, a, h): h \equiv g^a}$ mod **p**}.

Các giá trị **p, g, h** được công khai, phải giữ bí mật **a**.

***Kí số:** Chọn ngẫu nhiên bí mật r = 235 $\in \mathbf{Z_{p-1}^*}$**.** Khoá kí là (**a**, **r**).

Vì **r** $\in \mathbf{Z_{p-1}^*}$, nên nguyên tố cùng **p-1**, do đó tồn tại $\mathbf{r^{-1}}$ mod **(p-1).** Cụ thể:

UCLN (**r**, **p-1**) = UCLN (235, 462) = 1

nên $\mathbf{r}^{-1}$ mod **(p-1)** = 235^{-1} mod 462 = 289.

Chữ kí trên dữ liệu **x =** 112 là **(γ,δ) =** (16,18).

Trong đó: $\gamma = \mathbf{g^r}$ mod **p** = 2^{235} mod 463 = 16

$\delta = \mathbf{(x-a*\gamma)*r^{-1}}$ mod **(p-1)** = (**112**-211*16)* 289 mod 462 = **108**

***Kiểm tra chữ kí:** $\mathbf{Ver_k(x, y, \delta)}$ = đúng $\leftrightarrow$ $\mathbf{h^{\gamma}*\gamma^{\delta} \equiv g^x}$ mod **p**

$\mathbf{h^{\gamma} * \gamma^{\delta}}$ = $249^{16} * 16^{108}$ mod 463 = **132**

$\mathbf{g^x}$ mod **p** = 2^{112} mod 463 = **132**.

Hai giá trị đó bằng nhau, như vậy chữ ký là đúng.

1.3.6. Chữ kí DSS

Sơ đồ chuẩn chữ kí DSS

Sơ đồ

*** Tạo cặp khoá (bí mật, công khai) (a, h):**

+ Chọn số nguyên tố **p** sao cho bài toán logarit rời rạc trong $\mathbf{Z_p}$ là "khó" giải.

Chọn **q** là ước nguyên tố của **p-1**. Tức là p-1 = **t***q hay p = **t*** q+1.

(Số nguyên tố p cỡ 512 bít, q cỡ 160 bít).

+ Chọn $\mathbf{g} \in \mathbf{Z_p^*}$ là căn bậc **q** của 1 mod **p**, (**g** là phần tử sinh của $\mathbf{Z_p^*}$).

Tính $\boldsymbol{\alpha} = \mathbf{g^t}$, chọn khoá bí mật a $\in \mathbf{Z_p^*}$, tính khoá công khai $\mathbf{h} \equiv \boldsymbol{\alpha}^{\mathbf{a}} \mod \mathbf{p}$.

+ Đặt $\mathbf{P} = \mathbf{Z_q^*}$, $\mathbf{A} = \mathbf{Z_q^*} \times \mathbf{Z_q^*}$, $\mathbf{K} = \{(p, q, \alpha, a, h) / a \in \mathbf{Z_p^*}, h \equiv \alpha^a \mod p\}$.

+ Với mỗi khoá **(p, q, α, a, h), k' = a** bí mật, **k" = (p, q, α, h)** công khai.

*** Kí số** : Dùng 2 khoá kí: khoá a và khoá ngẫu nhiên bí mật $\mathbf{r} \in \mathbf{Z_q^*}$.

Chữ kí trên $\mathbf{x} \in \mathbf{Z_p^*}$ là $\mathbf{Sig_{k'}(x, r) = (\gamma, \delta)}$ trong đó:

$\boldsymbol{\gamma} = (\boldsymbol{\alpha}^{\mathbf{r}} \mod \mathbf{p}) \mod \mathbf{q}$, $\boldsymbol{\delta} = ((\mathbf{x} + \mathbf{a} * \boldsymbol{\gamma})) * \mathbf{r}^{-1} \mod \mathbf{q}$

(Chú ý $\mathbf{r} \in \mathbf{Z_q^*}$, để đảm bảo tồn tại $\mathbf{r}^{-1} \mod \mathbf{q}$)

*** Kiểm tra chữ kí**: Với $\mathbf{e_1} = \mathbf{x} * \boldsymbol{\delta}^{-1} \mod \mathbf{q}$, $\mathbf{e_2} = \boldsymbol{\gamma} * \boldsymbol{\delta}^{-1} \mod \mathbf{q}$.

$\mathbf{Ver_{k''}(x, \gamma, \delta)}$ = đúng $\leftrightarrow$ $(\boldsymbol{\alpha}^{\mathbf{e1}} * \mathbf{h}^{\mathbf{e2}} \mod \mathbf{p}) \mod \mathbf{q} = \boldsymbol{\gamma}$.

Ví dụ

*** Tạo cặp khoá (bí mật, công khai) (a,h):**

Chọn **p =** 7649, **q** = 239 là ước nguyên tố của p-1, **t** = 32 .

Tức là p-1 = **t*q** hay p = **t*** q+1 = 32 * q+1 = 32*239 + 1 = 7649.

Chọn **g**= 3 $\in$ $\mathbf{Z_{7649}}$ là phần tử sinh **α** = $\mathbf{g^t}$ mod p = 7098^{85} mod 7649 =7098.

Chọn khoá mật **a =** 85 khoá công khai **h** $\equiv$ $\mathbf{\alpha^a}$ mod **p** = 7098^{85} mod 7649 = 5387

*** Ký số** :Dùng 2 khoá kí: **a** và khoá ngẫu nhiên **r =** 58 $\in$ $\mathbf{Z_q^*}$, r^{-1} mod q = 136.

+ Chữ kí trên **x** $\in$ $\mathbf{Z_p^*}$ là $\mathbf{Sig_k}$, **(x, r) = (γ, δ)** trong đó:

γ = ($\mathbf{\alpha^r}$ mod **p**) mod **q =** (7089^{58} mod 7649) mod 239 = 593 mod 239 = 115.

δ = ((x+ a *γ))* $\mathbf{r^{-1}}$ mod **q =** (1246 + 85 * 115)* 136 mod 239 = 87.

*** Kiểm tra chữ kí**: **(γ, δ) =** (115,87) là chữ ký trên x = 1246.

Với $\mathbf{e_1}$ = **x* δ**$^{-1}$ mod **q** = 1246 *11 mod q = 83,

$\mathbf{e_2}$ **= γ* δ**$^{-1}$ mod **q** = 115*11 mod q = 70.

Điều kiện kiểm thử đúng ? ($\mathbf{\alpha^{e1}}$***** $\mathbf{h^{e2}}$ mod **p**) mod **q = γ,** với **δ**$^{-1}$ = 1 .

(7089^{58} mod 7649) mod 239 = 593 mod 239 = 115

1.3.7. Chữ ký không thể phủ định

1.3.7.1. Sơ đồ chữ ký không thể phủ định (Chaum – van Antverpen)

*** Chuẩn bị các tham số:**

Chọn số nguyên tố **p** sao cho bài toán log rời rạc trong $\mathbf{Z_p}$ là khó.

p = 2*q+1, q cũng là số nguyên tố.

Gọi **P** là nhóm nhân con của $\mathbf{Z_p^*}$ theo q (gồm các thặng dư bậc hai theo mod p).

Chọn phần tử sinh **g** của nhóm **P** cấp q.

Đặt **P = A= P**, **K = {(p, g,a, h): a** $\in \mathbf{Z_p^*}$ **,h** $\equiv \mathbf{g^a}$ mod **p**}

1/. Thuật toán ký: Dùng khoá bí mật k' = **a** để kí lên **x**:

Chữ ký là **y = Sig$_{k'}$(x) = x^a** mod **p**.

2/. Giao thức kiểm thử: Dùng khoá công khai k'' = **(p, g, h)**.

Với **x, y** $\in$ **P**, người nhận N cùng người gửi G thực hiên giao thức kiểm thử:

+ N chọn ngẫu nhiên $\mathbf{e_1, e_2} \in \mathbf{Z_q^*}$

+ N tính $\mathbf{c = y^{e1} h^{e2}}$ mod p và gửi cho G

+ G tính $\mathbf{d} = c^{a^{-1} \bmod q} \bmod p$ và gửi cho N

+ N chấp nhận y là chữ kí đúng, nếu $d \equiv x^{e1} g^{e2} \bmod p$.

3/. Giao thức chối bỏ:

+ N chọn ngẫu nhiên $\mathbf{e_1, e_2} \in \mathbf{Z_q^*}$

+ N tính $\mathbf{c = y^{e1} h^{e2}}$ mod p, và gửi cho G.

+ G tính d = $c^{a^{-1} \bmod q}$ mod *p* và gửi cho N.

+ N thử điều kiện $\mathbf{d \neq x^{e1} g^{e2}}$ (mod p).

+ N chọn ngẫu nhiên $\mathbf{f_1, f_2} \in \mathbf{Z_q^*}$.

+ N tính $\mathbf{C = y^{f1} * \beta^{f2}}$ mod p và gửi cho G.

+ G tính $\mathbf{D} = C^{a^{-1} \bmod q}$ mod *p* và gửi cho N.

+ N thử điều kiện $\mathbf{D \neq x^{f1} g^{f2}}$ (mod p).

+ N kết luận y là chữ kí **giả mạo** nếu:

$(d^* \alpha^{-e2})^{f1} \equiv (D * \alpha^{-f2})^{e1}$ (mod p) (thay **α** bằng **g**).

Ví dụ: Ký trên **x** = 229

*** Chuẩn bị các tham số:**

Chọn số nguyên tố **p** = 467 = 2 *q+1, q = 233 cũng là số nguyên tố.

Chọn phần tử sinh của nhóm **P** là **g = 4 (P** là nhóm nhân con q của $\mathbf{Z_p^*}$**)**.

Đặt **P = A= *P*, K = {(p, g,a, h): a** $\in$ $\mathbf{Z_p^*}$ **,h** $\equiv$ $\mathbf{g^a}$ mod **p**}

Chọn khoá mật **a = 121,** chọn khóa công khai **h** $\equiv$ $\mathbf{g^a}$ mod **p=** 4^{121}mod 467= 422.

1/. Thuật toán ký: Dùng khoá bí mật k' = **a** để kí lên **x=** 299 .

Chữ ký là **y = $\mathbf{Sig_{k'}(x) = x^a}$** mod **p=** 299^{121} mod 467 = 9.

2/. Giao thức kiểm thử: Dùng khoá công khai k" = **(p, g, h)** = (467, 4, 422).

Với **x,y** $\in$ **P**, người nhận N cùng người gửi G thực hiện giao thức kiểm thử:

+ N chọn ngẫu nhiên $\mathbf{e_1}$= 48, $\mathbf{e_2}$ = 213 $\in$ $\mathbf{Z_q^*}$

+N tính **c =** $\mathbf{y^{e1}\ h^{e2}}$ mod p = 116 và gửi cho G.

+ G tính **d =** $c^{a^{-1} \bmod q}$ mod *p* = 235 và gửi cho N.

+ N chấp nhận y là chữ kí đúng, nếu **d**$\equiv$ $\mathbf{x^{e1}\ g^{e2}}$ mod p.

N thử điều kiện **d** $\equiv$ $\mathbf{x^{e1}\ g^{e2}}$ mod p **.**

Rõ ràng $235 \equiv 229^{48}\ 8\ 4^{213} \pmod{467}$.

N chấp nhận **y** = 9 đúng là chữ ký của G trên **x** = 229.

3/. Giao thức chối bỏ

Giả sử G gửi tài liệu **x** = 226 với chữ ký **y** = 183. Giao thức chối bỏ thực hiện:

+ N chọn ngẫu nhiên $\mathbf{e_1}$= 47, $\mathbf{e_2}$ = 137 $\in$ $\mathbf{Z_q^*}$.

+ N tính $\mathbf{c = y^{e1}\, h^{e2}}$ mod p = 306 , và gửi cho G.

+ G tính d = $c^{a^{-1} \bmod q}$ mod p = 148 và gửi cho N.

+ N thử điều kiện $\mathbf{d \neq x^{e1}\ g^{e2}}$ (mod p).

Điều kiện trên không đúng vì: $184 \neq 226^{47} * 4^{137} \equiv 145 \bmod 467$.

N lại tiếp tục thực hiện bước 5 của giao thức.

+ N chọn ngẫu nhiên $\mathbf{f_1}$= 225, $\mathbf{f_2}$ = 19 $\in$ $\mathbf{Z_q^*}$.

+ N tính $\mathbf{C = y^{f1} * \beta^{f2}}$ mod p = 348, và gửi cho G.

+ G tính $\mathbf{D = c^{a^{-1} \bmod q}}$ mod p = 426, và gửi cho N.

+ N thử điều kiện $\mathbf{D \neq x^{f1}\, g^{f2}}$ (mod p).

D = 426 trong khi $\mathbf{x^{f1}\, g^{f2}}$ (mod p) = $226^{225} * 4^{19} \equiv 295 \bmod 467$.

Điều kiện 8 là đúng, nên N thực hiện bước 9:

+ N kết luận y là chữ kí **giả mạo** nếu:

$(d^* \alpha^{-e2})^{f1} \equiv (D * \alpha^{-f2})^{e1}$ (mod p) (thay **α** bằng **g**).

N tính giá trị của 2 vế đồng dư $\equiv$ $(d^* \alpha^{-e2})^{f1} \equiv (184 * 4^{-137})^{225} \equiv 79 \bmod 467$

$(D * \alpha^{-f2})^{e1} \equiv (426 * 4^{-19})^{17} \equiv 79 \bmod 467$

Hai giá trị đó bằng nhau. Kết luận chữ ký **y** là **giả mạo.**

Chương 2. GIAO THỨC PHÂN PHỐI KHOÁ MẬT

2.1. KHÁI NIỆM PHÂN PHỐI KHOÁ MẬT

Phân phối khoá mật là **cơ chế** để một tổ chức **chọn khoá mật**, sau đó truyền khoá mật, hay chỉ truyền **"vật liệu công khai"** và **"cách thức"** tạo **khoá mật** đến cặp người dùng muốn có chung **khoá mật**.

Hơn thế nữa bảo dảm rằng thám mã khó có thể khám phá hay tráo đổi khoá mật của họ. Phương pháp thiết lập khoá chung này phải nhờ một tổ chức tin cậy (TT) điều phối.

Vấn đề đặt ra là bằng cách nào để trung tâm được uỷ quyền (TT) có thể chuyển một cách an toàn khoá mật đến cặp người dùng U và V muốn có chung khoá mật $K_{u,v}$? hay chỉ chuyển vật liệu công khai hay cách thức tạo khoá mật cho họ.

Mặt khác giảm được lượng thông tin cần truyền đi và cất giữ của mỗi cặp người dùng. Hơn thế nữa bảo đảm rằng kẻ thám mã khó thể khám phá hay tráo đổi khoá mật của cặp người dùng. Hiện có hai phương pháp chính:

+ Phương pháp thông thường:

Trung tâm được uỷ quyền (TT) chuyển từng khoá mật cho cặp người dùng U. Phương pháp này phải dùng nhiều thông tin truyền đi và cất giữ, mặt khác độ an toàn thấp khi truyền khoá trên mạng công khai. Mặt khác TT cũng biết được khoá mật.

+ Phương pháp hiệu quả:

Trung tâm được uỷ quyền (TT) chỉ chuyển vật liệu công khai và cách thức tạo khoá mật đến cặp người dùng U và V, trong khi người dùng vẫn giữ gìn vật liệu riêng (bí mật) để thiết lập khoá.

Phương pháp này không phải dùng nhiều thông tin truyền đi và cất giữ, mặt khác độ an toàn cao, vì chỉ truyền trên mạng vật liệu công khai và cách thức tạo khoá, chứ không trực tiếp khóa mật

2.1.1. Phân phối khoá theo phương pháp thông thường

Giả sử, có một mạng không an toàn gồm n người dùng, Trung tâm được uỷ quyền (TT) phân phối khoá riêng cho mỗi cặp người dùng.

Theo phương pháp thông thường, tổng số khoá riêng giữa 2 người dùng nhiều nhất là (n-1) + (n-2) + (n-3) + ...+2+1 = n(n-1)/2.

Như vậy người dùng phải lưu trữ (n-1) khoá. TT phải tạo ra n(n-1)/2 khoá và chuyển mỗi khoá cho duy nhất một cặp người dùng.

Phương pháp này chỉ nên sử dụng khi số người dùng không nhiều. Nếu n lớn thì giải pháp này không thực tế, vì lượng thông tin rất lớn cần phải truyền đi khó bảo đảm an toàn, mặt khác vì người dùng phải cất giữ nhiều khoá mật. Đó là các khoá mật của (n-1) người dùng khác.

2.1.2. Phân phối khoá theo phương pháp hiệu quả

Phương pháp phân phối khoá hiệu quả đạt được hai tiêu chí sau:

+ Bảo đảm an toàn các thông tin về khoá mật.

Tức là bảo đảm rằng thám mã khó có thể khám phá hay tráo đổi khoá mật.

+ Giảm được thông tin cần truyền đi và cất giữ, trong khi vẫn cho phép mỗi cặp người dùng tính toán được khoá mật.

Hiện nay có nhiều phương pháp phân phối khoá hiệu quả, trung tâm được uỷ quyền (TT) chỉ chuyển vật liệu công khai và cách thức tạo khoá mật đến cặp người dùng. Người dùng tự tính khoá chung cho họ.

Thám mã có trộm được tin trên đường truyền, cũng khó tính được khoá mật vì không biết vật liệu bí mật của người dùng.

Sau đây sẽ giới thiệu một số phương pháp phân phối khoá hiệu quả: Sơ đồ phân phối khoá Blom, Diffie-Hellman, Kerboros,...

2.2. GIAO THỨC PHÂN PHỐI KHOÁ BLOM

Ý tưởng chính

Ta giả thiết rằng có một mạng gồm n người sử dụng.

Giả sử rằng các khoá được chọn trên trường hữu hạn Z_p, trong đó p là số nguyên tố ($p \geq n$).

Cho k là số nguyên, $1 \leq k \leq n-2$. Giá trị k để hạn chế kích thước lớn nhất mà sơ đồ vẫn duy trì được độ mật.

TT sẽ truyền đi k+1 phần tử của Z_p, cho người sử dụng trên kênh an toàn (so với n-1 trong sơ đồ phân phối trước cơ bản). Mỗi cặp người sử dụng U và V sẽ có khả năng tính khoá $K_{u, v} = K_{v, u}$ như trước đây.

Điều kiện an toàn như sau: tập bất kì gồm nhiều nhất k người sử dụng không liên kết từ {U, V} phải không có khả năng xác định bất kì thông tin nào về $K_{u, v}$.

2.2.1. Giao thức khoá Blom với k =1

Sơ đồ

1/. Số nguyên tố p công khai, với người sử dụng U, phần tử $r_u \in Z_p$ là công khai, khác nhau.

2/. TT chọn 3 phần tử ngẫu nhiên bí mật a, b, c $\in Z_p$ (không cần khác biệt) và thiết lập đa thức:

$$f(x, y) = (a + b*(x + y) + c*x*y) \mod p.$$

3/. Với người sử dụng U, TT tính đa thức: $g_u(x) = f(x, r_u) \mod p$ và truyền $g_u(x)$ đến U trên kênh an toàn.

$g_u(x)$ là đa thức tuyến tính theo x, có thể viết:

$g_u(x) = f(x, r_u) \mod p = (a+b.(x+ r_u) + c.x. r_u \mod p) \mod p$ hay

$g_u(x) = a_u + b_u*x$, trong đó:

$$a_u = a + b*r_u \mod p$$

$$b_u = b + c*r_u \mod p$$

4/. Nếu U và V muốn liên lạc với nhau, họ sẽ dùng khoá chung:

$K_{u, v} = K_{v, u} = f(r_u, r_v) = (a + b*(r_u+ r_v) + c.r_u.r_v) \mod p.$

U tính $K_{u, v} = f(r_u, r_v) = g_u(r_v) =(a+b.(r_v+ r_u) + c.r_v. r_u) \mod p.$

V tính $K_{u, v} = f(r_u,r_v) = g_v(r_u) =(a+b.(r_u+ r_v) + c.r_u.r_v) \mod p.$

Do tính chất đối xứng của đa thức f(x,y), nên $K_{u,v}= K_{v,u}$.

Ví dụ 1

1/. Giả sử có 3 người sử dụng là U,V và W. Chọn số nguyên tố p =17,

Các phần tử công khai của họ là $r_u = 12$, $r_v = 7$, $r_w = 1$.

2/. TT chọn ngẫu nhiên, bí mật a = 8, b = 7, c = 2. Khi đó đa thức f như sau:

f(x, y) = (8 + 7*(x + y) + 2*x*y) mod 17.

3/. TT tính các đa thức và gửi cho U, V, W tương ứng là:

$g_u(x) = f(x, 12) = (8 + 7*(x + 12) + 12*2*x) \bmod 17 = 7 + 14*x.$

$g_v(x) = f(x, 7) = (8 + 7*(x + 7) + 7*2*x) \bmod 17 = 6 + 4*x.$

$g_w(x) = f(x, 1) = (8 + 7*(x + 1) + 12*2*x) \bmod 17 = 15 + 9*x.$

4/. Khi U và V muốn liên lạc với nhau, người dùng tự tính khoá chung:

U tính $K_{u,v} = g_u(r_v) = f(r_u,r_v) = (a+b.(r_v + r_u) + c.r_v. r_u) \bmod p$

$= 7 + 14*7 \bmod 17 = 3.$

V tính $K_{u,v} = g_v(r_u) = f(r_u,r_v) = (a+b.(r_u + r_v) + c.r_u.r_v) \bmod p$

$= 6 + 4*12 \bmod 17 = 3.$

* 3 khoá tương ứng với 3 cặp người dùng là:

$K_{u,v} = f(r_u,r_v) = (8 + 7*(12 + 7) + 2*12*7) \bmod 17 = 3.$

$K_{u,w} = f(r_u,r_w) = (8 + 7*(12 + 1) + 2*12*71) \bmod 17 = 4.$

$K_{v,w} = f(r_v,r_w) = (8 + 7*(7 + 1) + 2*7*71) \bmod 17 = 10.$

Mức an toàn

a). Sơ đồ Blom với k = 1 an toàn với 1 đối thủ.

Định lý:

Theo sơ đồ Blom với k =1, **khoá của một cặp đối tác** là an toàn không điều kiện trước bất kì người sử dụng thứ ba.

TT: Không một người sử dụng nào có thể xác định được thông tin về khoá của 2 người sử dụng khác.

***Chứng minh*:**

Giả sử người sử dụng thứ ba là W muốn thử tính khoá.

$K_{u, v} = (a + b*(r_u + r_v) + c*r_u*r_v)$ mod p

Trong đó các giá trị r_u, r_v là công khai, còn a, b, c không được biết.

W tìm biết được các giá trị:

$a_w = a + b*r_w$ mod p.

$b_w = b + c*r_w$ mod p.

Vì chúng là hệ số của đa thức $g_w(x)$ được TT gửi đến cho W.

Ta sẽ chỉ ra rằng thông tin mà W biết phù hợp với giá trị tùy ý $t \in Z_p$ của khoá $K_{u,v}$.

Xét phương trình ma trận sau:

$$\begin{matrix} 1 & r_u + r_v & r_u r_v \\ 1 & r_w & 0 \\ 0 & 1 & r_w \end{matrix} \begin{matrix} a \\ b \\ c \end{matrix} = \begin{matrix} t \\ a_w \\ b_w \end{matrix}$$

(2) Tức là hệ các phương trình:

$K_{u,v} = (a+b *(r_u + r_v) + c *r_u *r_v) \bmod p = t$ (1)

$a+b *r_w \bmod p = a_w$

$b+c *r_w \bmod p = b_w$ (3)

(1) thể hiện giả thiết rằng $K_{u,v} = t$, (2),(3) cho thấy W biết a, b và c từ $g_w(x)$.

Định thức của ma trận hệ số là:

$$\{ (1*r_w *r_w) + (1 *1 *r_u r_v) +(0 *(r_u + r_v) *0) \}-$$

$$\{ 0*r_w *r_u r_v) + (1 *1 *0) +(1 *(r_u + r_v) *r_w)\}=$$

$$= \{r_w^2 + r_u r_v\} – \{(r_u + r_v)*r_w\} = (r_w - r_u)(r_w - r_v).$$

Vì $r_w \neq r_u$ và $r_w \neq r_v$ nên định thức ma trận hệ số khác không. Do đó phương trình ma trận có nghiệm duy nhất cho a, b, c.

Nói cách khác, bất kì giá trị t nào thuộc Z_p cũng có thể nhận là khoá $K_{u,v}$.

b). Sơ đồ không an toàn với liên minh 2 đối thủ

Định lý

Liên minh của 2 người sử dụng {W, X}, (không phải là cặp người dùng {U, V}) sẽ có khả năng xác định khoá $K_{u,v}$ bất kì U và V.

Chứng minh

Hai người W và X cùng biết rằng các đẳng thức sau:

$a_w = a + b*r_w$.

$b_w = b + c*r_{w.}$

$a_x = a + b*r_x$.

$b_x = b + c*r_x$.

Như vậy, họ có bốn phương trình trong đó ba ẩn chưa biết và dễ dàng tính ra nghiệm duy nhất cho a, b, c. Một khi đã biết a, b, và c, họ có thể thiết lập đa thức f(x, y) và tính khoá bất kì mà họ muốn .

2.2.2. Giao thức phân phối khoá Blom với k > 1

Sơ đồ

Để tạo lập sơ đồ phân phối khoá chống lại được liên minh k đối thủ, TT sẽ dùng đa thúc f(x,y) dạng sau:

$$f(x,y) = \sum_{i=0}^{k} \sum_{j=o}^{k} a_{i,j}\, x^i y^j \mod p.$$

Trong đó: $a_{i,j} \in Z_p$ ($0 \le i \le k$, $0 \le j \le k$) và $a_{i,j} = a_{j,i}$ với mọi i, j.

Các phần còn lại của giao thức như sơ đồ phân phối khoá Blom với k=1.

2.3. GIAO THỨC PHÂN PHỐI KHOÁ DIFFIE - HELLMAN

Sơ đồ

1/. Chọn số nguyên tố p sao cho bài toán logarit rời rạc Z_p là "*khó giải*".

Chọn α là phần tử nguyên thuỷ của Z_p^*

Giá trị α và p là công khai (Người dùng hoặc TT chọn).

Người dùng U chọn số mũ bí mật a_u ($0 \le a_u \le p - 2$) và tính giá trị công khai tương ứng :

$$b_u = \alpha^{a_u} \mod p.$$

Người sử dụng U sẽ có một dấu xác nhận của tt về ID(U) và b_u :

$$C(U) = (ID(U), b_u, sig_{TT}(ID(U), b_u)) .$$

2/. Để có khoá chung với V, người dùng U (có a_u) tính

$$K_{u,v} = b_v^{a_u} \mod p = \alpha^{a_u a_v} \mod p$$

3/. Để có khoá chung với U, người dùng V(có a_v) tính:

$$K_{u,v} = b_u^{a_v} \mod p = \alpha^{a_u a_v} \mod p$$

Rõ ràng 2 khoá là như nhau và bằng $\alpha^{a_u a_v} \mod p$

Ví dụ:

1/. Cho số nguyên tố p = 25307, sao cho bài toán logarit rời rạc Z_p .Chọn phần tử nguyên thuỷ $\alpha = 2 \in Z^*_p$,

Giá trị α và p là công khai (Người dùng hoặc TT chọn).

Người dùng U chọn số mũ bí mật $a_u = 3578$ và tính giá trị công khai tương ứng:

$$b_u = \alpha^{a_u} \bmod p = 2^{3578} \bmod 25307 = 6113.$$

Người dùng V chọn số mũ bí mật $a_v = 19956$. và tính giá trị công khai tương ứng: $b_v = \alpha^{a_v} \bmod p = 2^{19956} \bmod 25307 = 7984$.

2/. U (có a_u) tính khoá chung:

$$K_{u,v} = b_v^{a_u} \bmod p = 7984^{3578} \bmod 25307 = 3694.$$

3/. V (có a_v) tính khoá chung:

$$K_{u,v} = b_u^{a_v} \bmod p = 6113^{19956} \bmod 25307 = 3694.$$

Hai giá trị khoá trên là bằng nhau.

Mức an toàn

1/. Với loại tấn công chủ động, không cần lo lắng nhiều, vì:

Người dùng có dấu xác nhận C(U) của trung tâm được uỷ quyền TT, điều này ngăn chặn người dùng khác U có thể biến đổi thông tin nào đó trong dấu xác nhận.

$C(U) = (ID(U), b_u, sig_{TT}(ID(U), b_u))$.

2/. Với loại tấn công thụ động, cũng không cần lo lắng nhiều, vì:

Người dùng W (khác U,V) "khó" có thể tính được khoá chung $K_{u,v}$ của U,V. Cụ thể khi biết $b_u = \alpha^{a_u} \bmod p$ và $b_v = \alpha^{a_v} \bmod p$, thì cũng "khó" có thể tính được khoá chung của U và V là $K_{u,v} = \alpha^{a_u a_v} \bmod p$ (1).

Muốn tính đươc (1), W phải tính được a_u từ b_u. Nhưng đó là các trường hợp riêng của bài toán Logarit rời rạc. Như vậy chỉ cần bài toán Logarit rời rạc là "**khó**" giải thì sơ đồ phân phối khoá Diffie-Hellman sẽ "an toàn" trước kiểu tấn công loại này.

Bài toán Diffie-Hellman: Đó là vấn đề trên.

Cho trước số nguyên p, phần tử nguyên thuỷ $\alpha \in Z_p^*$, phần tử $\beta, \gamma \in Z_p^*$

Yêu cầu: Tính $\beta^{\log_\alpha \gamma} \bmod p \; (= \gamma^{\log_\alpha \beta} \bmod p)$?

Chương 3 GIAO THỨC THOẢ THUẬN KHOÁ MẬT

3.1. KHÁI NIỆM THOẢ THUẬN KHOÁ MẬT

Nếu không muốn dùng dịch vụ phân phối khoá qua trung tâm được uỷ quyền TT, cặp người dùng phải tự thoả thuận khoá (trao đổi) khoá bí mật.

Thoả thuận khoá mật là giao thức để cặp người dùng (hoặc nhiều hơn) liên kết với nhau cùng thiết lập khoá mật, bằng cách liên lạc trên kênh công khai.

Phương pháp thiết lập khoá chung kiểu này không nhờ Tổ chức tin cậy TT điều phối, cặp người dùng tự "Thoả thuận khoá mật".

Hiện nay có hai phương pháp chính để "Thoả thuận khoá mật":

+ Phương pháp thông thường

Khi cặp người dùng thông nhất có một khoá mật chung, thì một trong hai người chọn khoá ngẫu nhiên K, sau đó truyền nó một cách an toàn đến người kia bằng phương pháp nào đó, ví dụ bằng mã khoá công khai hay phương pháp "giấu tin". Phương pháp này phải dùng nhiều thông tin truyền đi và cất giữ, mặt khác độ an toàn thấp vì phải truyền đi "trọn vẹn " một khoá trên mạng công khai.

+ Phương pháp hiệu quả

Phương pháp hiệu quả để thoả thuận khoá phải đạt được hai tiêu chí sau:

+ Bảo đảm an toàn các thông tin về khoá mật .

Tức là bảo đảm rằng thám mã khó có thể khám phá hay tráo đổi khoá mật.

+ Giảm được thông tin cần truyền đi và cất giữ, trong khi vẫn cho phép mỗi cặp người dùng tính toán được khoá mật.

Theo phương pháp hiệu quả, người dùng không truyền cho nhau trên mạng "trọn vẹn" một khoá K, mà chỉ truyền vật liệu công khai và cách thức tạo khoá K đến cặp người dùng U và V.

Phương pháp này không phải dùng nhiều thông tin truyền đi và cất giữ, mặt khác độ an toàn cao, vì người dùng chỉ truyền trên mạng vật liệu công khai và cách thức tạo khoá mật, chứ không truyền trực tiếp khoá mật.

Thám mã có trộm được tin trên đường truyền, cũng khó tính được khoá mật vì không biết vật liệu bí mật của từng người dùng.

3.2. GIAO THỨC THỎA THUẬN KHOÁ DIFFIE - HELLMAN

Sơ đồ

Chuẩn bị:

Chọn số nguyên tố p sao cho bài toán logarit rời rạc trong Z_p là "*khó giải*".

Chọn α là phần tử nguyên thuỷ của Z_p^*

Giá trị α và p là công khai

1/. Người dùng U chọn a_u ngẫu nhiên, bí mật ($0 \leq a_u \leq p - 2$).

Tính :

$b_u = \alpha^{a_u} \bmod p$ và gửi nó đến V.

2/. Người dùng V chọn a_v ngẫu nhiên, bí mật ($0 \leq a_v \leq p - 2$).

Tính :

$b_v = \alpha^{a_v} \bmod p$ và gửi nó đến U.

3/. U tính khoá

$$K_{u,v} = (\alpha^{a_v})^{a_u} \bmod p.$$

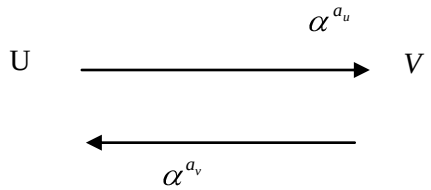
4/. V tính khoá chung

$$K_{v,u} = (\alpha^{a_v})^{a_u} \bmod p.$$

Hai giá trị khoá bằng nhau!

Mức an toàn của sơ đồ

Thông tin trao đổi trong giao thức được mô tả như sau:



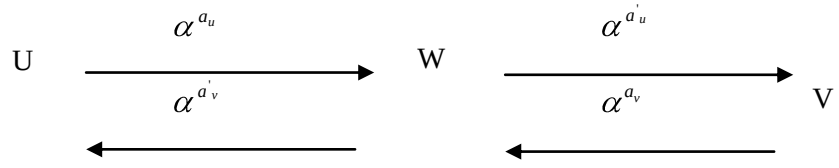
1/. Hạn chế

Không có xác thực danh tính U và V

Giao thức này dễ bị tổn thương trước đối phương tích cực – những người sử dụng cách tấn công "***kẻ xâm nhập vào giữa cuộc***". Đó là tình tiết của vở "The Lucy show", trong đó nhân vật Vivian Vance đang dùng bữa tối với người bạn, còn Lucille Ball đang trốn dưới bàn. Vivian và người bạn của cô đang cầm tay nhau dưới bàn. Lucy cố tránh bị phát hiện đã nắm lấy tay của cả hai người, còn hai người trên bàn vẫn nghĩ rằng họ đang cầm tay nhau.

Cuộc tấn công kiểu "kẻ xâm nhập vào giữa cuộc" trên giao thức trao đổi khoá Diffie-Hellman hoạt động cũng như vậy. W sẽ ngăn chặn các bức điện trao đổi giữa U và V và thay thế bằng các bức điện của anh ta.

Ta có sơ đồ như sau:



Cuối giao thức, U thiết lập thực sự khoá mật $\alpha^{a_u a_v'}$ cùng với W, còn V thiết lập khoá mật $\alpha^{a'_u a_v}$ với W.

Khi U cố giải mã bức điện để gửi cho V, W cũng có khả năng giải mã nó song V thì không thể, (tương tự tình huống nắm tay nhau nếu V gửi bức điện cho U).

2/. Cải tiến

Bổ sung xác thực danh tính U và V.

Điều cơ bản đối với U và V là bảo đảm rằng, họ đang trao đổi khoá cho nhau mà không có W. Trước khi trao đổi khoá, U và V có thể thực hiện những giao thức tách bạch để thông báo danh tính của nhau, nhờ đó họ sẽ nhận ra kẻ không phải là U hay V.

3.3. GIAO THỨC THOẢ THUẬN KHOÁ TRẠM TỚI TRẠM

Giao thức thoả thuận khoá "Trạm tới Trạm" (STS) là cải tiến của giao thức phân khoá Diffie-Hellman, trong đó bổ sung phần xác thực danh tính của người dùng. STS được gọi là giao thức thoả thuận khoá có xác thực, nhờ trung tâm tin cậy TT.

Sơ đồ

Chọn số nguyên tố p sao cho bài toán logarit rời rạc trong Z_p là "*khó giải*".

Chọn α là phần tử nguyên thuỷ của Z_p^*

Giá trị α và p là công khai, có dấu xác nhận.

Ppk&ttksửa_ok sử dụng U có chữ ký với thuật toán xác minh ver_u.

TT cũng có chữ ký với thuật toán xác minh công khai ver_{TT} .

Người sử dụng U có dấu xác nhận:

$$C(U) = (ID(U), ver_u , sig_{TT}(ID(U), ver_u)).$$

1/. U chọn số ngẫu nhiên a_u, bí mật ($0 \leq a_u \leq p - 2$).

Tính:

$\alpha^{a_u} \bmod p$ và gửi nó đến V.

2/. V chọn số ngẫu nhiên a_v, bí mật ($0 \leq a_v \leq p - 2$).

Tính:

$\alpha^{a_v} \bmod p$

$y_v = sig_v(\alpha^{a_v}, \alpha^{a_u})$. Gửi (C(V), α^{a_v}, y_v) đến U.

V tính khoá chung:

$K_{v,u} = (\alpha^{a_v})^{a_u} \bmod p$

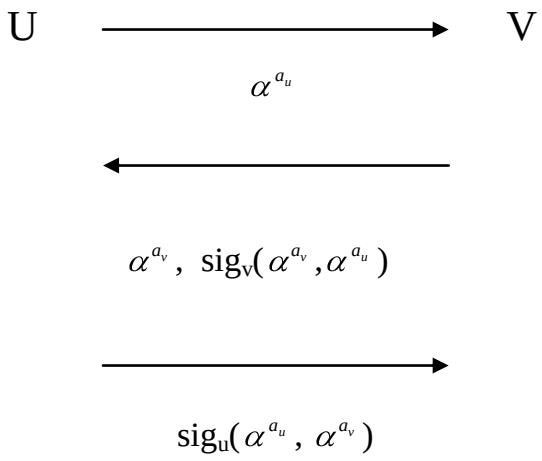
U tính khoá chung : $K_{u,v} = (\alpha^{a_u})^{a_v} \bmod p$

U dùng ver_v để xác minh y_v và xác minh C(V) nhờ ver_{TT}.

Tính: $y_u = sig_u(\alpha^{a_u}, \alpha^{a_v})$ và gửi (C(U), y_u) đến V.

3/. Dùng ver_u để xác minh y_u và C(U) bằng ver_{TT}.

STS là giao thức 3 lần truyền tin. Thông tin được trao đổi như sau:



1/. Giao thức có thể bảo vệ trước tấn công của "kẻ xâm nhập giữa cuộc"

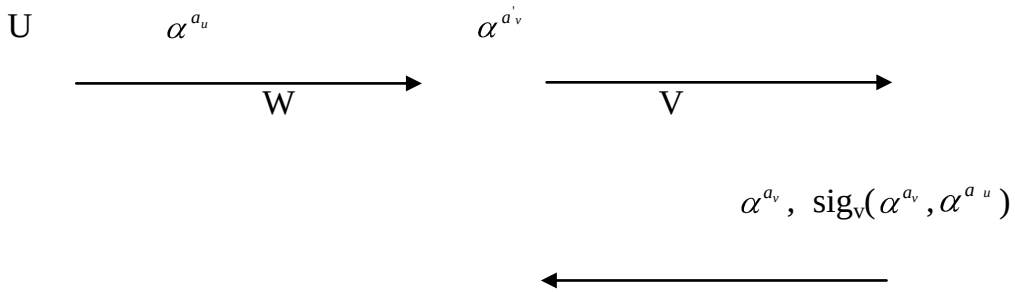
+ Giống như giao thức phân phối khoá Diffie- Hellman (PP DH) kẻ tấn công W chặn bắt α^{a_u} và thay nó bằng $\alpha^{a'_u}$. Sau đó W chặn bắt α^{a_v}, $sig_v(\alpha^{a_v}, \alpha^{a'_u})$ từ V.

+ W cũng muốn thay α^{a_v} của V bằng $\alpha^{a'_u}$. Điều này có nghĩa anh ta cũng phải thay

$y_v = sig_v(\alpha^{a_v}, \alpha^{a'_u})$ bằng $sig_v(\alpha^{a'_v}, \alpha^{a_u})$.

Đáng tiếc W khó thể tính $sig_v(\alpha^{a'_v}, \alpha^{a_u})$ vì không biết thuật toán kí sig_v của V.

+ Tương tự, W không thể thay $sig_u(\alpha^{a_u}, \alpha^{a'_v})$ bằng $sig_v(\alpha^{a'_u}, \alpha^{a_v})$ do anh ta không biết thuật toán kí của U.



W muốn thay bằng để gửi cho U, nhưng khó thể tính .

$sig_v(\alpha^{a_v}, \alpha^{a_u})$

W muốn thay bằng $\alpha^{a'_u}$ để gửi cho V, nhưng khó thể tính $sig_v(\alpha^{a'_u}, \alpha^{a_v})$=?

2/. Giao thức STS không đưa ra sự khẳng định khoá.

Tức là trong bước 2), α^{a_v} và y_v được gửi tới U, nhưng chưa bảo đảm thật an toàn.

Trong bước 1/, 3/ α^{a_v} và y_u được gửi tới V, nhưng chưa bảo đảm thật an toàn.

Có thể bảo đảm an toàn y_v và y_u bằng cách:

Trong bước 2/: mã hoá y_v bằng khoá K:

$$y_v = e_K(sig_v(\alpha^{a_v}, \alpha^{a_u})) = e_K(y_v).$$

Trong bước 3/: mã hoá y_u bằng khoá K:

$$y_u = e_K(sig_u(\alpha^{a_u}, \alpha^{a_v})) = e_K(y_u)$$

Chương 4 THỬ NGHIỆM CHƯƠNG TRÌNH

4.1. CHƯƠNG TRÌNH PHÂN PHỐI KHÓA BLOM VỚI K > 1

4.1.1. Cấu hình hệ thống.

+Phần cứng

Yêu cầu phần cứng của chương trình: CPU Khoảng 15- 20M.

+ Phần mềm

Yêu cầu phần mềm của chương trình: Tubo C++ phiên bản 4.9.9.2, Hệ điều hành Windown XP.

4.1.2. Các thành phần của chương trình.

Thành phần của chương trình gồm :

+ Input: - Số lượng người dùng, hệ số k, số nguyên tố p.

- Các phần tử công khai và hệ số a ngẫu nhiên bí mật.

+ Output: - Khóa tương ứng giữa những cặp người dùng.

4.1.3. Chương trình.

```
#include <iostream>
#include <cmath>
using namespace std;
//-------------------------------------------------
int a[1000][1000];//cac so ngau nhien bi mat ma TT chon
int k;          //he so k
int p;          //so nguyen to p
int n;          //so luong nguoi dung
int r[1000];    //phan tu cong khai cua n nguoi dung
//-----------------------------------------------------------------------
void gx(int y)
{
   int heso_x[100] = {0};
   for(int i=0;i<=k;i++)
  {
      for(int j=0;j<=k;j++)
     {
         heso_x[i] += (int)(a[i][j]*pow((double)y,j));
      }
```

```
 heso_x[i] %= p;
        if(i>1)
            cout << " + " << heso_x[i] << "*x^" << i;
        else if(i==1)
            cout << " + " << heso_x[i] << "*x";
        else if(i==0)
            cout << heso_x[i];
    }
}
//------------------------------------------------------------------------
int f(int x, int y)
{
    int kq = 0;
    for(int i=0;i<=k;i++)
  {
        for(int j=0;j<=k;j++)
        {
            kq += a[i][j]*pow((double)x,i)*pow((double)y,j);
        }
    }
    kq %= p;
    return kq;
}
```

```
//---------------------------------------------------------------------
void nhapdulieu()
{
   cout << "So luong nguoi dung = ";
   cin >> n;
   cout << "k = ";
   cin >> k;
   cout << "p = ";
   cin >> p;
   cout << "Cac phan tu cong khai cua " << n << " nguoi dung lan luot la:\n";
   for(int i=0;i<n;i++)
   {
      cout << "r["<< i+1 <<"] = ";
      cin >> r[i];
   }
   cout << "Cac he so a[i][j] do TT chon ngau nhien bi mat la:\n";
   for(int i=0;i<=k;i++)
      for(int j=0;j<=i;j++)
      {
         cout << "a[" << j << "][" << i << "] = ";
         cin >> a[i][j];
         a[j][i] = a[i][j];
      }}
```

```
int main()
 {
   nhapdulieu();
   cout << "\nDa thuc gui cho "<< n <<" nguoi dung lan luot la:\n";
   for(int i=0;i<n;i++)
   {
      cout << "G" << i+1 << "(x) = ";   gx(r[i]);
      cout << "\n";
   }
   cout << "\nKhoa tuong ung giua nhung cap nguoi dung la:\n";
   for(int i=0;i<n;i++)
   {
      for(int j=0;j<=i;j++)
     {
         if(i!=j)
         {
             cout << "K["<< j+1 <<"]["<< i+1 <<"] = " << f(r[i],r[j]) << "\n";
         }
      }
   }
   cout << "\n";
   system("pause");
}
```

4.1.4. Hướng dẫn sử dụng chương trình.

+ Khởi động TC để vào chương trình.

- Sinh khóa
- Khai báo hệ số k.
- Khai báo số nguyên tố p.
- Khai báo số lượng người dùng.
- Khai báo các phần tử công khai.

+ Trước khi chạy chương trình nhấn F9 để kiểm tra lỗi

+ Nếu không báo lỗi nhấn tổ hợp phím CTRL+ F9 để chạy chương trình.

Kết quả thử nghiệm của chương trình:

Số lượng người dùng = 3

k = 4

p = 83

Cac phan tu cong khai cua 3 nguoi dùng lan lươt la:

r[1] = 3

r[2] = 6

r[3] = 9

Cac he so a[i][j] do TT chon ngau nhien bi mat la:

a[0][0] = 3

a[0][1] = 3

a[1][1] = 3

a[1][2] = 3

a[2][2] = 3

a[0][3] = 3

a[1][3] = 3

a[2][3] = 3

a[3][3] = 3

a[0][4] = 3

a[0][4] = 3

a[2][4] = 3

a[3][4] = 3

a[4][4] = 3

Da thuc gui cho 3 nguoi dung lan luot la:

G1(x) = 35+ 45*x+ 82*x^2+ 50*x^3 + 12*x^4

G2(x) = 72+ 57*x+ 73*x^2+ 34*x^3 + 7*x^4

G1(x) = 17+ 13*x+ 19*x^2+ 57*x^3 + 9*x^4

Khoa tuong ung giua nhung cap nguoi dung la:

K[1][2] = 61

K[1][3] = 5

K[2][3] = 21

Pess any key to continue...

4.2. CHƯƠNG TRÌNH THỎA THUẬN KHÓA DIFFIE - HELLMAN

4.2.1. Cấu hình hệ thống.

+Phần cứng

Yêu cầu phần cứng của chương trình: CPU: Khoảng 15- 20M.

+ Phần mềm

Yêu cầu phần mềm của chương trình: Tubo C++ phiên bản 4.9.9.2, Hệ điều hành Windown XP.

4.2.2. Các thành phần của chương trình.

Thành phần của chương trình gồm :

+ Input- Số nguyên tố p.

- Khóa bí mật của người dùng U.
- Khóa bí mật của người dùng V.

Output: Khóa chung của U và V.

4.2.3. Chương trình.

```
#include <iostream>
#include <cmath>
using namespace std;

int p, alpha, aU, aV, bU, bV, K_VU, K_UV;
//---------------------------------------------------------------



int mod(int x,int n,int m)//ham tinh x^n mod m
{
  int p;
  if (n==0) return 1;
  p=mod(x,n/2,m);
  if (n%2==0)
     return (p*p)%m;
  else
     return (p*p*x)%m;
}

//-----------------------------------------------------------------------
```

```
int main()
{
   cout << "p = ";
   cin >> p;

   cout << "alpha = ";
   cin >> alpha;

   cout << "Nguoi dung U chon so ngau nhien bi mat aU = ";
   cin >> aU;
   cout << "Nguoi dung V chon so ngau nhien bi mat aV = ";
   cin >> aV;
   bU = mod(alpha,aU,p);
   bV = mod(alpha,aV,p);
   K_UV = mod(mod(alpha,aV,p),aU,p);
   K_VU = mod(mod(alpha,aU,p),aV,p);
   cout << "\nNguoi dung U gui den V gia tri bU = " << bU;
   cout << "\nNguoi dung V gui den U gia tri bV = " << bV;
   cout << "\nU tinh khoa chung K_UV = " << K_UV << "\nV tinh khoa chung K_VU
= " << K_VU;
   cout << "\n";
   system("pause");
}
```

4.2.4. Hướng dẫn sử dụng chương trình phân phối khóa Diffeie – Hellman.

+ Khởi động TC để vào chương trình.

- Khai báo hệ số k.
- Khai báo số nguyên tố p.
- Khai báo alpha.
- Khai báo khóa bí mật của U.
- Khai báo khóa bí mật của V.

+ Trước khi chạy chương trình nhấn F9 để kiểm tra lỗi

+ Nếu không báo lỗi nhấn tổ hợp phím CTRL+ F9 để chạy chương trình.

Kết quả thử nghiệm của chương trình:

p = 83

alpha = 56

Nguoi dung U gui den V gia tri bU = 8

Nguoi dung V gui den U gia tri bV = 51

U tinh khoa chung K_UV = 37

V tinh khoa chung K_VU = 37

Press any key to continue...

KẾT LUẬN:

1/. Tìm hiểu và nghiên cứu qua tài liệu để hệ thống lại các vấn đề sau:

+ Tổng quan về phân phối khóa và thỏa thuận khóa mật:

Như đã trình bày ở trên, nghiên cứu phân phối khóa và thỏa thuận khóa mật để đáp ứng những yêu cầu mới trong các lĩnh vực bảo mật thông tin trên đường truyền.

+ Một số bài toán về An toàn thông tin trong phân phối khóa và thỏa thuận khóa mật:

- Bài toán bảo mật thông tin.
- Bài toán về xác thực thông tin.
- Bài toán về toàn vẹn thông tin.....

+ Phương pháp giải quyết các bài toán: Có nhiều cách khắc phục về vấn đề an toàn thông tin, nhưng việc tìm ra cách khắc phục tốt nhất thì chúng ta còn phải trải qua quá trình nghiên cứu lâu dài.

Chúng ta có thể chọn một số cách sau: mã hóa, ký số, chứng thực số, hàm băm....

2/. Thử nghiệm chương trình Phân phối khóa và thỏa thuận khóa mật.

TÀI LIỆU THAM KHẢO

[1]. Phan Đình Diệu. Lý thuyết mật mã và An toàn thông tin, 2004

[2]. Trịnh Nhật Tiến. Bài giảng môn An toàn dữ liệu, 2005