KHOA HỌC KHÁM PHÁ

# MẬT MÃ

# THE CODE BOOK

THE SCIENCE OF SECRECY FROM ANCIENT EGYPT TO QUANTUM CRYPTOGRAPHY

TỪ CỔ ĐIỂN ĐẾN LƯỢNG TỬ

SIMON SINGH

NHÀ XUẤT BẢN TRẺ

CHƯƠNG 1

# MẬT MÃ CỔ ĐIỂN

## 1.1 MỞ ĐẦU - MỘT SỐ HỆ MẬT ĐƠN GIẢN

Đối tượng cơ bản của mật mã là tạo ra khả năng liên lạc trên một kênh không mật cho hai người sử dụng (tạm gọi là Alice và Bob) sao cho đối phương (Oscar) không thể hiểu được thông tin được truyền đi. Kênh này có thể là một đường dây điện thoại hoặc một mạng máy tính. Thông tin mà Alice muốn gửi cho Bob (bản rõ) có thể là một văn bản tiếng Anh, các dữ liệu bằng số hoặc bất cứ tài liệu nào có cấu trúc tuỳ ý. Alice sẽ mã hoá bản rõ bằng một khoá được xác định trước và gửi bản mã kết quả trên kênh. Oscar có bản mã thu trộm được trên kênh song không thể xác định nội dung của bản rõ, nhưng Bob (người đã biết khoá mã) có thể giải mã và thu được bản rõ.

Ta sẽ mô tả hình thức hoá nội dung bằng cách dung khái niệm toán học như sau:

***Định nghĩa 1.1***

*Một hệ mật là một bộ 5 $(\mathcal{P},\mathcal{C},\mathcal{K},\mathcal{E},\mathcal{D})$ thoả mãn các điều kiện sau:*

1. *$\mathcal{P}$ là một tập hữu hạn các bản rõ có thể.*
2. *$\mathcal{C}$ là một tập hữu hạn các bản mã có thể.*
3. *$\mathcal{K}$ (không gian khoá) là tập hữu hạn các khoá có thể.*
4. *Đối với mỗi $k \in \mathcal{K}$ có một quy tắc mã $e_k$: $\mathcal{P} \to \mathcal{C}$ và một quy tắcv giải mã tương ứng $d_k \in D$. Mỗi $e_k$: $\mathcal{P} \to \mathcal{C}$ và $d_k$: $\mathcal{C} \to \mathcal{P}$ là những hàm mà:*

$$d_k(e_k(x)) = x \text{ với mọi bản rõ } x \in \mathcal{P}.$$

Trong tính chất 4 là tính chất chủ yếu ở đây. Nội dung của nó là nếu một bản rõ x được mã hoá bằng $e_k$ và bản mã nhận được sau đó được giải mã bằng $d_k$ thì ta phải thu được bản rõ ban đầu x. Alice và Bob sẽ áp dụng thủ tục sau dùng hệ mật khoá riêng. Trước tiên họ chọn một khoá ngẫu nhiên K $\in \mathcal{K}$. Điều này được thực hiện khi họ ở cùng một chỗ và không bị Oscar theo dõi hoặc khi họ có một kênh mật trong trường hợp họ ở xa nhau. Sau đó giả sử Alice muốn gửi một thông baó cho Bob trên một kênh không mật và ta xem thông báo này là một chuỗi:

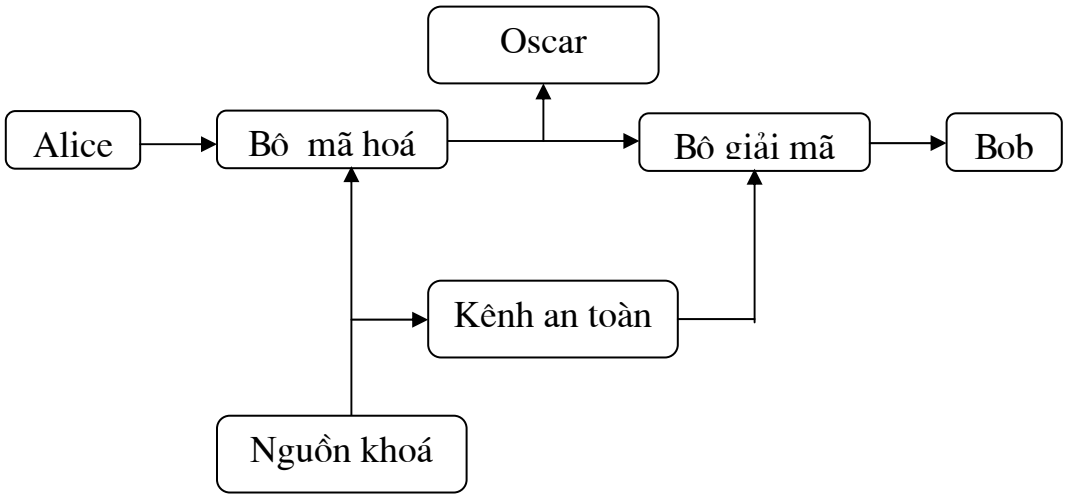
$$x = x_1,x_2 ,. . .,x_n$$

với số nguyên n $\geq 1$ nào đó. Ở đây mỗi ký hiệu của mỗi bản rõ $x_i \in \mathcal{P}$, $1 \leq i \leq n$. Mỗi $x_i$ sẽ được mã hoá bằng quy tắc mã $e_k$ với khoá K xác định trước đó. Bởi vậy Alice sẽ tính $y_i = e_k(x_i)$, $1 \leq i \leq n$ và chuỗi bản mã nhận được:

$$y = y_1,y_2 ,. . .,y_n$$

sẽ được gửi trên kênh. Khi Bob nhận đươc $y_1,y_2 ,. . .,y_n$ anh ta sẽ giải mã bằng hàm giải mã $d_k$ và thu được bản rõ gốc $x_1,x_2 ,. . .,x_n$. Hình 1.1 là một ví dụ về một kênh liên lạc

***Hình 1.1. Kênh liên lạc***




Rõ ràng là trong trường hợp này hàm mã hoá phải là hàm đơn ánh ( tức là ánh xạ 1-1), nếu không việc giải mã sẽ không thực hiện được một cách tường minh. Ví dụ

$$y = e_k(x_1) = e_k(x_2)$$

trong đó $x_1 \neq x_2$ , thì Bob sẽ không có cách nào để biết liệu sẽ phải giải mã thành $x_1$ hay $x_2$ . Chú ý rằng nếu $\mathcal{P} = \mathcal{C}$ thì mỗi hàm mã hoá là một phép hoán vị, tức là nếu tập các bản mã và tập các bản rõ là đồng nhất thì mỗi một hàm mã sẽ là một sự sắp xếp lại (hay hoán vị ) các phần tử của tập này.

### *1.1.1 Mã dịch vòng ( shift cipher)*

Phần này sẽ mô tả mã dịch (MD) dựa trên số học theo modulo. Trước tiên sẽ điểm qua một số định nghĩa cơ bản của số học này.

***Định nghĩa 1.2***

*Giả sử a và b là các số nguyên và m là một số nguyên dương. Khi đó ta viết a ≡ b (mod m) nếu m chia hết cho b-a. Mệnh đề a ≡ b (mod m) được gọi là " a đồng dư với b theo modulo m". Số nguyên m được gọi là mudulus.*

Giả sử chia a và b cho m và ta thu được thương nguyên và phần dư, các phần dư nằm giữa 0 và m-1, nghĩa là $a = q_1m + r_1$ và $b = q_2m + r_2$ trong đó $0 \le r_1 \le m-1$ và $0 \le r_2 \le m-1$. Khi đó có thể dễ dàng thấy rằng a ≡ b (mod m) khi và chỉ khi $r_1 = r_2$ . Ta sẽ dùng ký hiệu a mod m (không dùng các dấu ngoặc) để xác định phần dư khi a được chia cho m (chính là giá trị $r_1$ ở trên). Như vậy: a ≡ b (mod m) khi và chỉ khi a mod m = b mod m. Nếu thay a bằng a mod m thì ta nói rằng a được rút gọn theo modulo m.

*Nhận xét:* Nhiều ngôn ngữ lập trình của máy tính xác định a mod m là phần dư trong dải - m+1,.. ., m-1 có cùng dấu với a. Ví dụ -18 mod 7 sẽ là -4, giá trị này khác với giá trị 3 là giá trị được xác định theo công thức trên. Tuy nhiên, để thuận tiện ta sẽ xác định a mod m luôn là một số không âm.

Bây giờ ta có thể định nghĩa số học modulo m: $Z_m$ được coi là tập hợp {0,1,. . .,m-1} có trang bị hai phép toán cộng và nhân. Việc cộng và nhân trong $Z_m$ được thực hiện giống như cộng và nhân các số thực ngoài trừ một điểm làcác kết quả được rút gọn theo modulo m.

Ví dụ tính 11× 13 trong $Z_{16}$ . Tương tự như với các số nguyên ta có 11 ×13 = 143. Để rút gọn 143 theo modulo 16, ta thực hiện phép chia bình thường: 143 = 8 × 16 + 15, bởi vậy 143 mod 16 = 15 trong $Z_{16}$ .

Các định nghĩa trên phép cộng và phép nhân $Z_m$ thảo mãn hầu hết các quy tắc quyen thuộc trong số học. Sau đây ta sẽ liệt kê mà không chứng minh các tính chất này:

1. Phép cộng là đóng, tức với bất kì $a,b \in Z_m$ ,$a+b \in Z_m$
2. Phép cộng là giao hoán, tức là với a,b bất kì $\in Z_m$
   $$a+b = b+a$$
3. Phép cộng là kết hợp, tức là với bất kì $a,b,c \in Z_m$
   $$(a+b)+c = a+(b+c)$$
4. 0 là phần tử đơn vị của phép cộng, có nghĩa là với a bất kì $\in Z_m$
   $$a+0 = 0+a = a$$

5. Phần tử nghịch đảo của phép cộngcủa phần tử bất kì ($a \in Z_m$ ) là m-a, nghĩa là a+(m-a) = (m-a)+a = 0 với bất kì $a \in Z_m$ .
6. Phép nhân là đóng , tức là với a,b bất kì $\in Z_m$ , $ab \in Z_m$ .
7. Phép nhân là gioa hoán , nghĩa là với a,b bất kì $\in Z_m$ , ab = ba
8. Phép nhân là kết hợp, nghĩa là với $a,b,c \in Z_m$ , (ab)c = a(cb)
9. 1 là phần tử đơn vị của phép nhân, tức là với bất kỳ $a \in Z_m$

$$a \times 1 = 1 \times a = a$$

10. Phép nhân có tính chất phân phối đối với phép cộng, tức là đối với $a,b,c \in Z_m$ , (a+b)c = (ac)+(bc) và a(b+c) = (ab) + (ac)

Các tính chất 1,3-5 nói lên rằng $Z_m$ lập nên một cấu trúc đại số được gọi là một nhóm theo phép cộng. Vì có thêm tính chất 4 nhóm được gọi là nhóm Aben (hay nhóm gioa hoán).

Các tính chất 1-10 sẽ thiết lập nên một vành $Z_m$ . Ta sẽ còn thấy nhiều ví dụ khác về các nhóm và các vành trong cuốn sách này. Một số ví dụ quên thuộc của vành là các số nguyên Z, các số thực R và các số phức C. Tuy nhiên các vành này đều vô hạn, còn mối quan tâm của chúng ta chỉ giới hạn trên các vành hữu hạn.

Vì phần tử ngược của phép cộng tồn tại trong $Z_m$ nên cũng có thể trừ các phần tử trong $Z_m$ . Ta định nghĩa a-b trong $Z_m$ là a+m-b mod m. Một cách tương có thể tính số nguyên a-b rồi rút gon theo modulo m.

Ví dụ : Để tính 11-18 trong $Z_{31}$, ta tính 11+13 mod 31 = 24. Ngược lại, có thể lấy 11-18 được -7 rồid sau đó tính -7 mod 31 = 24.

Ta sẽ mô tả mã dịch vòng trên hình 1.2. Nó được xác định trên $Z_{26}$ (do có 26 chữ cái trên bảng chữ cái tiếng Anh) mặc dù có thể xác định nó trên $Z_m$ với modulus m tuỳ ý. Dễ dàng thấy rằng, MDV sẽ tạo nên một hệ mật như đã xác định ở trên, tức là $d_K(e_K(x)) = x$ với mọi $x \in Z_{26}$ .

***Hình 1.2: Mã dịch vòng***

Giả sử $\mathcal{P} = \mathcal{C} = \mathcal{K} = Z_{26}$ với $0 \le k \le 25$ , định nghĩa:

$$e_K(x) = x + K \bmod 26$$

và

$$d_K(x) = y - K \bmod 26$$

$(x,y \in Z_{26})$

*Nhận xét:* Trong trường hợp K = 3, hệ mật thường được gọi là mã Caesar đã từng được Julius Caesar sử dụng.

Ta sẽ sử dụng MDV (với modulo 26) để mã hoá một văn bản tiếng Anh thông thường bằng cách thiết lập sự tương ứnggiữa các kí tự và các thặng dư theo modulo 26 như sau: A $\leftrightarrow$ 0,B $\leftrightarrow$ 1, . . ., Z $\leftrightarrow$ 25. Vì phép tương ứng này còn dùng trong một vài ví dụ nên ta sẽ ghi lại để còn tiện dùng sau này:

| A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

| N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

Sau đây là một ví dụ nhỏ để minh hoạ

*Ví dụ 1.1:*

Giả sử khoá cho MDV là K = 11 và bản rõ là:

*wewillmeetatmidnight*

Trước tiên biến đổi bản rõ thành dãy các số nguyên nhờ dùng phép tương ứng trên. Ta có:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 22 | 4 | 22 | 8 | 11 | 11 | 12 | 4 | 4 | 19 |
| 0 | 19 | 12 | 8 | 3 | 13 | 8 | 6 | 7 | 19 |

sau đó cộng 11 vào mỗi giá trị rồi rút gọn tổng theo modulo 26

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 7 | 15 | 7 | 19 | 22 | 22 | 23 | 15 | 15 | 4 |
| 11 | 4 | 23 | 19 | 14 | 24 | 19 | 17 | 18 | 4 |

Cuối cùng biến đổi dãy số nguyên này thành các kí tự thu được bản mã sau:

HPHTWWXPPELEXTOYTRSE

Để giả mã bản mã này, trước tiên, Bob sẽ biến đổi bản mã thành dãy các số nguyên rồi trừ đi giá trịcho 11 ( rút gọn theo modulo 26) và cuối cùng biến đổi lại dãy nàythành các ký tự.

Nhận xét: Trong ví dụ trên , ta đã dùng các chữ in hoa ch o bản mã, các chữ thường cho bản rõ đêr tiện phân biệt. Quy tắc này còn tiếp tục sử dụng sau này.

Nếu một hệ mật có thể sử dụng được trong thực tế thì nó phảo thoả mãn một số tính chất nhất định. Ngay sau đây sé nêu ra hai trong số đó:

1. Mỗi hàm mã hoá $e_K$ và mỗi hàm giải mã $d_K$ phải có khả năng tính toán được một cách hiệu quả.

2. Đối phương dựa trên xâu bản mã phải không có khả năng xác định khoá K đã dùng hoặc không có khả năng xác định được xâu bản rõ x.

Tính chất thứ hai xác định (theo cách khá mập mờ) ý tưởng ý tưởng "bảo mật". Quá trình thử tính khoá K (khi đã biết bản mã y) được gọi là mã thám (sau này khái niệm này sẽ đực làm chính xác hơn). Cần chú ý rằng, nếu Oscar có thể xác định được K thì anh ta có thể giải mã được y như Bob bằng cách dùng $d_K$. Bởi vậy, việc xác định K chí ít cũng khó như việc xác định bản rõ x.

Nhận xét rằng, MDV (theo modulo 26) là không an toàn vì nó có thể bị thám theo phương pháp vét cạn. Do chỉ có 26 khoá nên dễ dàng thử mọi khoá $d_K$ có thể cho tới khi nhận được bản rõ có nghĩa. Điều này được minh hoạ theo ví dụ sau:

Ví du 1.2

Cho bản mã

JBCRCLQRWCRVNBJENBWRWN

ta sẽ thử liên tiếp các khoá giải mã $d_0$ ,$d_1$ .. . và y thu được:

j b c r c l q r w c r v n b j e n b w r w n
i a b q b k p q v b q u m a i d m a v q v m
h z a p a j o p u a p t l z h c l z u p u l
g y z o z i n o t z o s k y g b k y t o t k
j x y n y h m n s y n r j e x f a j x s n s j
e w x m x g l m r x m q i w e z i w r m r i
d v w l w f k l q w l p h v o d y h v q l q h
c u v k v e j k p v k o g u c x g u p k p g
b t u j u d i j o u j n f t b w f o j o f
a s t i t c h i n t i m e s a v e s n i n e

Tới đây ta đã xác định được bản rõ và dừng lại. Khoá tương ứng K = 9.

Trung bình có thể tính được bản rõ sau khi thử 26/2 = 13 quy tắc giải mã.

Như đã chỉ ra trong ví dụ trên , điều kiện để một hệ mật an toàn là phép tìm khoá vét cạn phải không thể thực hiện được; tức không gian khoá phải rất lớn. Tuy nhiên, một không gian khoá lớn vẫn chưa đủ đảm bảo độ mật.

### *1.1.2 Mã thay thế*

Một hệ mật nổi tiếng khác là hệ mã thay thế. Hệ mật này đã được sử dụng hàng trăm năm. Trò chơi đố chữ "*cryptogram*" trong các bài báo là những ví dụ về MTT. Hệ mật này được nếu trên hình 1.3.

Trên thực tế MTT có thể lấy cả $\mathcal{P}$ và $\mathcal{C}$ đều là bộ chữ cái tiếng anh, gồm 26 chữ cái. Ta dùng $Z_{26}$ trong MDV vì các phép mã và giải mã đều là các phép toán đại số. Tuy nhiên, trong MTT, thích hợp hơn là xem phép mã và giải mã như các hoán vị của các kí tự.

***Hình 1.3 Mã thay thế***

Cho $\mathcal{P}=\mathcal{C}=Z_{26}$ . $\mathcal{K}$ chứa mọi hoán vị có thể của 26 kí hiệu 0,1, . . . ,25
Với mỗi phép hoán vị $\pi \in \mathcal{K}$, ta định nghĩa:

$$e\pi(x) = \pi(x)$$

và

$$d\pi(y) = \pi^{-1}(y)$$

trong đó $\pi^{-1}$ là hoán vị ngược của $\pi$.

Sau đây là một ví dụ về phép hoán vị ngẫu nhiên $\pi$ tạo nên một hàm mã hoá (cũng nhưb trước, các kí hiệu của bản rõ được viết bằng chữ thường còn các kí hiệu của bản mã là chữ in hoa).

| a | b | c | d | e | f | g | h | i | j | k | l | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| X | N | Y | A | H | P | O | G | Z | Q | W | B | T |

| n | o | p | q | r | s | t | u | v | w | x | y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| S | F | L | R | C | V | M | U | E | K | J | D | I |

Như vậy, eπ (a) = X, eπ (b) = N,. . . . Hàm giải mã là phép hoán vị ngược. Điều này được thực hiện bằng cách viết hàng thứ hai lên trước rồi sắp xếp theo thứ tự chữ cái. Ta nhận được:

| A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| d | l | r | y | v | o | h | e | z | x | w | p | T |

| N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| b | g | f | j | q | n | m | u | s | k | a | c | I |

Bởi vậy dπ (A) = d, dπ(B) = 1, . . .

Để làm bài tập, bạn đọc có giải mã bản mã sau bằng cách dùng hàm giải mã đơn giản:

M G Z V Y Z L G H C M H J M Y X S S E M N H A H Y C D L M H A.

Mỗĩ khoá của MTT là một phép hoán vị của 26 kí tự. Số các hoán vị này là 26!, lớn hơn $4 \times 10^{26}$ là một số rất lớn. Bởi vậy, phép tìm khoá vét cạn không thể thực hiện được, thậm chí bằng máy tính. Tuy nhiên, sau này sẽ thấy rằng MTT có thể dễ dàng bị thám bằng các phương pháp khác.

### *1.1.3 Mã Affine*

MDV là một trường hợp đặc biệt của MTT chỉ gồm 26 trong số 26! các hoán vị có thể của 26 phần tử. Một trường hợp đặc biệt khác của MTT là mã Affine được mô tả dưới đây. trong mã Affine, ta giới hạn chỉ xét các hàm mã có dạng:

$$e(x) = ax + b \bmod 26,$$

a,b $\in Z_{26}$ . Các hàm này được gọi là các hàm Affine (chú ý rằng khi a = 1, ta có MDV).

Để việc giải mã có thể thực hiện được, yêu cầu cần thiết là hàm Affine phải là đơn ánh. Nói cách khác, với bất kỳ y $\in Z_{26}$, ta muốn có đồng nhất thức sau:

$$ax + b \equiv y \pmod{26}$$

phải có nghiệm x duy nhất. Đồng dư thức này tương đương với:

$$ax \equiv y-b \pmod{26}$$

Vì y thay đổi trên $Z_{26}$ nên y-b cũng thay đổi trên $Z_{26}$ . Bởi vậy, ta chỉ cần nghiên cứu phương trình đồng dư:

$$ax \equiv y \pmod{26} \qquad (y \in Z_{26}).$$

Ta biết rằng, phương tfình này có một nghiệm duy nhất đối với mỗi y khi và chỉ khi UCLN(a,26) = 1 (ở đây hàm UCLN là ước chung lớn nhất của các biến của nó). Trước tiên ta giả sử rằng, UCLN(a,26) = d >1. Khi đó, đồng dư thức $ax \equiv 0 \pmod{26}$ sẽ có ít nhất hai nghiệm phân biệt trong $Z_{26}$ là x = 0 và x = 26/d. Trong trường hợp này, e(x) = ax + b mod 26 không phải là một hàm đơn ánh và bởi vậy nó không thể là hàm mã hoá hợp lệ.

Ví dụ, do UCLN(4,26) = 2 nên 4x +7 không là hàm mã hoá hợp lệ: x và x+13 sẽ mã hoá thành cùng một giá trị đối với bất kì $x \in Z_{26}$ .

Ta giả thiết UCLN(a,26) = 1. Giả sử với $x_1$ và $x_2$ nào đó thảo mãn:

$$ax_1 \equiv ax_2 \pmod{26}$$

Khi đó

$$a(x_1 - x_2) \equiv 0 \pmod{26}$$

bởi vậy

$$26 \mid a(x_1 - x_2)$$

Bây giờ ta sẽ sử dụng một tính chất của phép chia sau: Nếu USLN(a,b)=1 và $a \mid bc$ thì $a \mid c$. Vì $26 \mid a(x_1 - x_2)$ và USLN(a,26) = 1 nên ta có:

$$26 \mid (x_1 - x_2)$$

tức là

$$x_1 \equiv x_2 \pmod{26}$$

Tới đây ta chứng tỏ rằng, nếu UCLN(a,26) = 1 thì một đồng dư thức dạng $ax \equiv y \pmod{26}$ chỉ có (nhiều nhất) một nghiệm trong $Z_{26}$ . Do đó , nếu ta cho x thay đổi trên $Z_{26}$ thì ax mod 26 sẽ nhận được 26 giá trị khác nhau theo modulo 26 và đồng dư thức $ax \equiv y \pmod{26}$ chỉ có một nghiệm y duy nhất.

Không có gì đặc biệt đối vơi số 26 trong khẳng định này. Bởi vậy, bằng cách tương tự ta có thể chứng minh được kết quả sau:

***Định lí 1.1***

*Đồng dư thức $ax \equiv b \bmod m$ chỉ có một nghiệm duy nhất $x \in Z_m$ với mọi $b \in Z_m$ khi và chỉ khi UCLN(a,m) = 1.*

Vì $26 = 2 \times 13$ nên các giá trị $a \in Z_{26}$ thoả mãn UCLN(a,26) = 1 là a = 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23 và 25. Tham số b có thể là một phần tử bất kỳ trong $Z_{26}$. Như vậy, mã Affine có $12 \times 26 = 312$ khoá có thể ( dĩ nhiên con số này quá nhỉ để bảo đảm an toàn).

Bây giờ ta sẽ xét bài toán chung với modulo m. Ta cần một định nghĩa khác trong lý thuyết số.

***Định nghĩa 1.3***

*Giả sử $a \geq 1$ và $m \geq 2$ là các số nguyên. UCLN(a,m) = 1 thì ta nói rằng a và m là nguyên tố cùng nhau. Số các số nguyên trong $Z_m$ nguyên tố cùng nhau với m thường được ký hiệu là $\phi(m)$ ( hàm này được gọi là hàm Euler).*

Một kết quả quan trọng trong lý thuyết số cho ta giá trị của $\phi(m)$ theo các thừa số trong phép phân tích theo luỹ thừa các số nguyên tố của m. ( Một số nguyên p >1 là số nguyên tố nếu nó không có ước dương nào khác ngoài 1 và p. Mọi số nguyên m >1 có thể phân tích được thành tích của các luỹ thừa các số nguyên tố theo cách duy nhất. Ví dụ $60 = 2^3 \times 3 \times 5$ và $98 = 2 \times 7^2$ ).

Ta sẽ ghi lại công thức cho $\phi(m)$ trong định lí sau:

***Định lý 1.2.*** ( thiếu )

Giả sử $m = \prod p_i$

Trong đó các số nguyên tố $p_i$ khác nhau và $e_i > 0$ ,1

Định lý này cho thấy rằng, số khoá trong mã Affine trên $Z_m$ bằng $m\phi(m)$, trong đó $\phi(m)$ được cho theo công thức trên. ( Số các phép chọn của b là m và số các phép chọn của a là $\phi(m)$ với hàm mã hoá là $e(x) = ax + b$). Ví dụ, khi m = 60, $\phi(60) = 2 \times 2 \times 4 = 16$ và số các khoá trong mã Affine là 960.

Bây giờ ta sẽ xét xem các phép toán giải mã trong mật mã Affine với modulo m = 26. Giả sử UCLN(a,26) = 1. Để giải mã cần giải phương trình đồng dư $y \equiv ax+b \pmod{26}$ theo x. Từ thảo luận trên thấy rằng, phương trình

này có một nghiệm duy nhất trong $Z_{26}$ . Tuy nhiên ta vẫn chưa biết một phương pháp hữu hiệu để tìm nghiệm. Điều cần thiết ở đây là có một thuật toán hữu hiệu để làm việc đó. Rất mayb là một số kết quả tiếp sau về số học modulo sẽ cung cấp một thuật toán giải mã hữu hiệu cần tìm.

***Định nghĩa 1.4***

*Giả sử $a \in Z_m$ . Phần tử nghịch đảo (theo phép nhân) của a là phần tử $a^{-1} \in Z_m$ sao cho $aa^{-1} \equiv a^{-1}a \equiv 1 \pmod m$.*

Bằng các lý luận tương tự như trên, có thể chứng tỏ rằng a có nghịch đảo theo modulo m khi và chỉ khi UCLN(a,m) =1, và nếu nghịch đảo này tồn tại thì nó phải là duy nhất. Ta cũng thấy rằng, nếu $b = a^{-1}$ thì $a = b^{-1}$ . Nếu p là số nguyên tố thì mọi phần tử khác không của $Z_P$ đều có nghịch đảo. Một vành trong đó mọi phần tử đều có nghịch đảo được gọi là một trường.

Trong phần sau sẽ mô tả một thuật toán hữu hiệu để tính các nghịch đảo của $Z_m$ với m tuỳ ý. Tuy nhiên, trong $Z_{26}$ , chỉ bằng phương pháp thử và sai cũng có thể tìm được các nghịch đảo của các phần tử nguyên tố cùng nhau với 26: $1^{-1} = 1$, $3^{-1} = 9$, $5^{-1} = 21$, $7^{-1} = 15$, $11^{-1} = 19$, $17^{-1} = 23$, $25^{-1} = 25$. (Có thể dễ dàng kiểm chứng lại điều này, ví dụ: $7 \times 5 = 105 \equiv 1 \bmod 26$, bởi vậy $7^{-1} = 15$).

Xét phương trình đồng dư $y \equiv ax+b \pmod{26}$. Phương trình này tương đương với

$$ax \equiv y-b \pmod{26}$$

Vì UCLN(a,26) =1 nên a có nghịch đảo theo modulo 26. Nhân cả hai vế của đồng dư thức với $a^{-1}$ ta có:

$$a^{-1}(ax) \equiv a^{-1}(y-b) \pmod{26}$$

Áp dụng tính kết hợp của phép nhân modulo:

$$a^{-1}(ax) \equiv (a^{-1}a)x \equiv 1x \equiv x.$$

Kết quả là $x \equiv a^{-1}(y-b) \pmod{26}$. Đây là một công thức tường minh cho x. Như vậy hàm giải mã là:

$$d(y) = a^{-1}(y-b) \bmod 26$$

Hình 1.4 cho mô tả đầy đủ về mã Affine. Sau đây là một ví dụ nhỏ

***Hình 1.4 Mật mãA ffine***

Cho $\mathcal{P} = \mathcal{C} = Z_{26}$ và giả sử

$\mathcal{P} = \{ (a,b) \in Z_{26} \times Z_{26} : UCLN(a,26) = 1 \}$

Với $K = (a,b) \in \mathcal{K}$ , ta định nghĩa:

$$e_K(x) = ax + b \bmod 26$$

và

$$d_K(y) = a^{-1}(y-b) \bmod 26,$$

$x,y \in Z_{26}$

*Ví dụ 1.3*

Giả sử K = (7,3). Như đã nêu ở trên, $7^{-1} \bmod 26 = 15$. Hàm mã hoá là

$$e_K(x) = 7x+3$$

Và hàm giải mã tương ứng là:

$$d_K(x) = 15(y-3) = 15y - 19$$

Ở đây, tất cả các phép toán đều thực hiện trên $Z_{26}$. Ta sẽ kiểm tra liệu $d_K(e_K(x)) = x$ với mọi $x \in Z_{26}$ không?. Dùng các tính toán trên $Z_{26}$ , ta có

$$\begin{aligned} d_K(e_K(x)) &= d_K(7x+3) \\ &= 15(7x+3)-19 \\ &= x + 45 - 19 \\ &= x. \end{aligned}$$

Để minh hoạ, ta hãy mã hoá bản rõ "*hot*". Trước tiên biến đổi các chữ h, o, t thành các thặng du theo modulo 26. Ta được các số tương ứng là 7, 14 và 19. Bây giờ sẽ mã hoá:

$$\begin{aligned} 7 \times 7 + 3 \bmod 26 &= 52 \bmod 26 = 0 \\ 7 \times 14 + 3 \bmod 26 &= 101 \bmod 26 = 23 \\ 7 \times 19 + 3 \bmod 26 &= 136 \bmod 26 = 6 \end{aligned}$$

Bởi vậy 3 ký hiệu của bản mã là 0, 23 và 6 tương ứng với xâu ký tự AXG. Việc giải mã sẽ do bạn đọc thực hiện như một bài tập.

### *1.1.4 Mã Vigenère*

Trong cả hai hệ MDV và MTT (một khi khoá đã được chọn) mỗi ký tự sẽ được ánh xạ vào một ký tự duy nhất. Vì lý do đó, các hệ mật còn được gọi hệ thay thế đơn biểu. Bây giờ ta sẽ trình bày ( trong hùnh 1.5) một hệ mật không phải là bộ chữ đơn, đó là hệ mã Vigenère nổi tiếng. Mật mã này lấy tên của Blaise de Vigenère sống vào thế kỷ XVI.

Sử dụng phép tương ứng A $\Leftrightarrow$ 0, B $\Leftrightarrow$ 1, . . . , Z $\Leftrightarrow$ 25 mô tả ở trên, ta có thể gắn cho mỗi khoa K với một chuỗi kí tự có độ dài m được gọi là từ khoá. Mật mã Vigenère sẽ mã hoá đồng thời m kí tự: Mỗi phần tử của bản rõ tương đương với m ký tự.

Xét một ví dụ nhỏ

*Ví dụ 1.4*

Giả sử m =6 và từ khoá là CIPHER. Từ khoá này tương ứng với dãy số K = (2,8,15,4,17). Giả sử bản rõ là xâu:

*thiscryptosystemisnotsecure*

***Hình 1.5 Mật mã Vigenère***

Cho m là một số nguyên dương cố định nào đó. Định nghĩa $\mathcal{P} = \mathcal{C} = \mathcal{K} = (Z_{26})^m$ . Với khoá K = $(k_1, k_2, \ldots, k_m)$ ta xác định :

$$e_K(x_1, x_2, \ldots, x_m) = (x_1+k_1, x_2+k_2, \ldots, x_m+k_m)$$

và

$$d_K(y_1, y_2, \ldots, y_m) = (y_1-k_1, y_2-k_2, \ldots, y_m-k_m)$$

trong đó tất cả các phép toán được thực hiện trong $Z_{26}$

Ta sẽ biến đổi các phần tử của bản rõ thành các thặng dư theo modulo 26, viết chúng thành các nhóm 6 rồi cộng với từ khoá theo modulo 26 như sau:

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 19 | 7 | 8 | 18 | 2 | 17 | 24 | 15 | 19 | 14 | 18 | 24 |
| 2 | 8 | 15 | 7 | 4 | 17 | 2 | 8 | 15 | 7 | 4 | 17 |
| 21 | 15 | 23 | 25 | 6 | 8 | 0 | 23 | 8 | 21 | 22 | 15 |
| 18 | 19 | 4 | 12 | 8 | 18 | 13 | 14 | 19 | 18 | 4 | 2 |
| 2 | 8 | 15 | 7 | 4 | 17 | 2 | 8 | 15 | 7 | 4 | 17 |
| 20 | 1 | 19 | 19 | 12 | 9 | 15 | 22 | 8 | 15 | 8 | 19 |

| | | |
|---|---|---|
| 20 | 17 | 4 |
| 2 | 8 | 15 |
| 22 | 25 | 19 |

Bởi vậy, dãy ký tự tương ứng của xâu bản mã sẽ là:

V P X Z G I A X I V W P U B T T M J P W I Z I T W Z T

Để giải mã ta có thể dùng cùng từ khoá nhưng thay cho cộng, ta trừ cho nó theo modulo 26.

Ta thấy rằng các từ khoá có thể với số độ dài m trong mật mã Vigenère là $26^m$, bởi vậy, thậm chí với các giá trị m khá nhỏ, phương pháp tìm kiếm vét cạn cũng yêu cầu thời gian khá lớn. Ví dụ, nếu m = 5 thì không gian khoá cũng có kích thước lớn hơn $1{,}1 \times 10^7$ . Lượng khoá này đã đủ lớn để ngaen ngừa việc tìm khoá bằng tay( chứ không phải dùng máy tính).

Trong hệ mật Vigenère có từ khoá độ dài m,mỗi ký tự có thể được ánh xạ vào trong m ký tự có thể có (giả sử rằng từ khoá chứa m ký tự phân biệt). Một hệ mật như vậy được gọi là hệ mật thay thế đa biểu (polyalphabetic). Nói chung, việc thám mã hệ thay thế đa biểu sẽ khó khăn hơn so việc thám mã hệ đơn biểu.

### *1.1.5 Mật mã Hill*

Trong phần này sẽ mô tả một hệ mật thay thế đa biểu khác được gọi là mật mã Hill. Mật mã này do Lester S.Hill đưa ra năm 1929. Giả sử m là một số nguyên dương, đặt $\mathcal{P} = C = (Z_{26})^m$ . Ý tưởng ở đây là lấy m tổ hợp tuyến tính của m ký tự trong một phần tử của bản rõ để tạo ra m ký tự ở một phần tử của bản mã.

Ví dụ nếu m = 2 ta có thể viết một phần tử của bản rõ là $x = (x_1,x_2)$ và một phần tử của bản mã là $y = (y_1,y_2)$. Ở đây, $y_1$cũng như $y_2$ đều là một tổ hợp tuyến tính của $x_1$và $x_2$. Chẳng hạn, có thể lấy

$$y_1 = 11x_1 + 3x_2$$
$$y_2 = 8x_1 + 7x_2$$

Tất nhiên có thể viết gọn hơn theo ký hiệu ma trận như sau

$$(y_1\ y_2) = (x_1\ x_2) \begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix}$$

Nói chung, có thể lấy một ma trận K kích thước m × m làm khoá. Nếu một phần tử ở hàng i và cột j của K là $k_{i,,j}$ thì có thể viết $K = (k_{i,,j})$, với $x = (x_1, x_2, \ldots ,x_m) \in \mathcal{P}$ và $K \in \mathcal{K}$, ta tính $y = e_K(x) = (y_1, y_2, \ldots ,y_m)$ như sau:

$$(y_1,\ldots,y_m)\ (x_1, \ldots ,x_m) \begin{pmatrix} k_{1,1} & k_{1,2} & \ldots & k_{1,m} \\ k_{2,1} & k_{2,2} & \ldots & k_{2,m} \\ \ldots & \ldots & \ldots & \ldots \\ k_{m,1} & k_{m,2} & \ldots & k_{m,m} \end{pmatrix}$$

Nói một cách khác y = xK.

Chúng ta nói rằng bản mã nhận được từ bản rõ nhờ phép biến đổi tuyến tính. Ta sẽ xét xem phải thực hiện giải mã như thế nào, tức là làm thế nào để tính x từ y. Bạn đọc đã làm quen với đại số tuyến tính sẽ thấy rằng phải dùng ma trận nghịch đảo $K^{-1}$ để giả mã. Bản mã được giải mã bằng công thức $y\ K^{-1}$ .

Sau đây là một số định nghĩa về những khái niệm cần thiết lấy từ đại số tuyến tính. Nếu $A = (x_{i,j})$ là một ma trận cấp l × m và $B = (b_{1,k})$ là một ma trận cấp m × n thì tích ma trận $AB = (c_{1,k})$ được định nghĩa theo công thức:

$$c_{1,k} = \sum_{j=1}^{m} a_{i,j}\, b_{j,k}$$

Với $1 \le i \le l$ và $1 \le k \le l$. Tức là các phần tử ở hàng i và cột thứ k của AB được tạo ra bằng cách lấy hàng thứ i của A và cột thứ k của B, sau đó nhân tương ứng các phần tử với nhau và cộng lại. Cần để ý rằng AB là một ma trận cấp $l \times n$.

Theo định nghĩa này, phép nhân ma trận là kết hợp (tức (AB)C = A(BC)) nhưng noiâ chung là không giao hoán ( không phải lúc nào AB = BA, thậm chí đố với ma trận vuông A và B).

Ma trận đơn vị $m \times m$ (ký hiệu là $I_m$ ) là ma trận cấp $m \times m$ có các số 1 nằm ở đường chéo chính và các số 0 ở vị trí còn lại. Như vậy ma trận đơn vị $2 \times 2$ là:

$$I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$I_m$ được gọi là ma trận đơn vị vì $AI_m = A$ với mọi ma trận cấp $l \times m$ và $I_mB = B$ với mọi ma trận cấp $m \times n$. Ma trận nghịch đảo của ma trận A cấp $m \times m$ ( nếu tồn tại) là ma trận $A^{-1}$ sao cho $AA^{-1} = A^{-1}A = I_m$ . Không phải mọi ma trận đều có nghịch đảo, nhưng nếu tồn tại thì nó duy nhất.

Với các định nghĩa trên, có thể dễ dàng xây dựng công thức giải mã đã nêu: Vì y = xK, ta có thể nhân cả hai vế của đẳng thức với $K^{-1}$ và nhận được:

$$yK^{-1} = (xK)K^{-1} = x(KK^{-1}) = xI_m = x$$

( Chú ý sử dụng tính chất kết hợp)

Có thể thấy rằng, ma trận mã hoá ở trên có nghịch đảo trong $Z_{26}$:

$$\begin{bmatrix} 11 & 8 \\ 3 & 7 \end{bmatrix}^{-1} = \begin{bmatrix} 7 & 18 \\ 23 & 11 \end{bmatrix}$$

vì

$$\begin{bmatrix} 12 & 8 \\ 3 & 7 \end{bmatrix} \begin{bmatrix} 8 & 18 \\ 23 & 11 \end{bmatrix} = \begin{bmatrix} 11\times7+8\times23 & 11\times18+8\times11 \\ 3\times7+7\times23 & 3\times18+7\times11 \end{bmatrix}$$

$$= \begin{bmatrix} 261 & 286 \\ 182 & 131 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

(Hãy nhớ rằng mọi phép toán số học đều được thực hiện theo modulo 26).

Sau đây là một ví dụ minh hoạ cho việc mã hoá và iải mã trong hệ mật mã Hill.

*Via dụ 1.5*

Giả sử khoá $K = \begin{bmatrix} 11 & 8 \\ 3 & 7 \end{bmatrix}$

Từ các tính toán trên ta có:

$$K^{-1} = \begin{bmatrix} 7 & 18 \\ 23 & 11 \end{bmatrix}$$

Giả sử cần mã hoá bản rõ "July". Ta có hai phần tử của bản rõ để mã hoá: (9,20) (ứng với Ju) và (11,24) (ứng với ly). Ta tính như sau:

$$(9,20) \begin{bmatrix} 11 & 8 \\ 3 & 7 \end{bmatrix} = (99+60, 72+140) = (3,4)$$

và

$$(11,24) \begin{bmatrix} 11 & 8 \\ 3 & 7 \end{bmatrix} = (121+72, 88+168) = (11,22)$$

Bởi vậy bản mã của July là DELW. Để giải mã Bob sẽ tính

$$(3,4) \begin{bmatrix} 7 & 18 \\ 23 & 11 \end{bmatrix} = (9,20)$$

và

$$(11,22 \begin{bmatrix} 7 & 18 \\ 23 & 11 \end{bmatrix} = (11,24)$$

Như vậy Bob đã nhận được bản đúng.

Cho tới lúc này ta đã chỉ ra rằng có thể thực hiện phép giải mã nếu K có một nghịch đảo. Trên thực tế, để phép giải mã là có thể thực hiện được, điều kiện cần là K phải có nghịch đảo. ( Điều này dễ dàng rút ra từ đại số tuyến tính sơ cấp, tuy nhiên sẽ không chứng minh ở đây). Bởi vậy, chúng ta chỉ quan tâm tới các ma trận K khả nghịch.

Tính khả nghịch của một ma trận vuông phụ thuộc vào giá trị định thức của nó. Để tránh sự tổng quát hoá không cần thiết, ta chỉ giới hạn trong trường hợp 2×2.

***Định nghĩa 1.5***

*Định thức của ma trận* $A = (a_{i,j})$ *cấp* $2\times 2$ *là giá trị*

$$det\ A = a_{1,1}\, a_{2,2} - a_{1,2}\, a_{2,1}$$

*Nhận xét:* Định thức của một ma trận vuông cấp mm có thể được tính theo các phép toán hằng sơ cấp: hãy xem một giáo trình bất kỳ về đại số tuyến tính.

Hai tính chất quan trọng của định thức là $\det I_m = 1$ và quy tắc nhân $\det(AB) = \det A \times \det B$.

Một ma trận thức K là có nghịch đảo khi và chỉ khi định thức của nó khác 0. Tuy nhiên, điều quan trọng cần nhớ là ta đang làm việc trên $Z_{26}$ . Kết quả tương ứng là ma trận K có nghịch đảo theo modulo 26 khi và chỉ khi UCLN(det K,26) = 1.

Sau đây sẽ chứng minh ngắn gọn kết quả này.

Trước tiên, giả sử rằng UCLN(det K,26) = 1. Khi đó det K có nghịch đảo trong $Z_{26}$ . Với $1 \le i \le m$, $1 \le j \le m$, định nghĩa $K_{i\,j}$ ma trận thu được từ K bằng cách loại bỏ hàng thứ i và cột thứ j. Và định nghĩa ma trận $K^*$ có phần tử (i,j) của nó nhận giá trị$(-1)$ $\det K_{j\,i}$ ($K^*$ được gọi là ma trận bù đại số của K). Khi đó có thể chứng tỏ rằng:

$$K^{-1} = (\det K)^{-1} K^* .$$

Bởi vậy K là khả nghịch.

Ngược lại K có nghịch đảo $K^{-1}$ . Theo quy tắc nhân của định thức

$$1 = \det I = \det (KK^{-1}) = \det K \det K^{-1}$$

Bởi vậy det K có nghịch đảo trong $Z_{26}$ .

Nhận xét: Công thức đối với ở trên không phải là một công thức tính toán có hiệu quả trừ các trường hợp m nhỏ ( chẳng hạn m = 2, 3). Vớim lớn, phương pháp thích hợp để tính các ma trận nghịch đảo phải dựa vào các phép toán hằng sơ cấp.

Trong trường hợp 2×2, ta có công thức sau:

***Định lý 1.3***

*Giả sử $A = (a_{i\ j})$ là một ma trận cấp $2 \times 2$ trên $Z_{26}$ sao cho $\det A = a_{1,1}a_{2,2} - a_{1,2}\ a_{2,1}$ có nghịch đảo. Khi đó*

$$A^{-1} = (\det A)^{-1} \begin{bmatrix} a_{2,2} & -a_{1,2} \\ -a_{2,1} & a_{1,1} \end{bmatrix}$$

Trở lại ví dụ đã xét ở trên . Trước hết ta có:

$$\det \begin{bmatrix} 11 & 8 \\ 3 & 7 \end{bmatrix} = 11 \times 7 - 8 \times 3 \bmod 2$$
$$= 77 - 24 \bmod 26 = 53 \bmod 26$$
$$= 1$$

Vì $1^{-1} \mod 26 = 1$ nên ma trận nghịch đảo là

$$\begin{bmatrix} 11 & 8 \\ 3 & 7 \end{bmatrix}^{-1} = \begin{bmatrix} 7 & 18 \\ 23 & 11 \end{bmatrix}$$

Đây chính là ma trận đã có ở trên.

Bây giờ ta sẽ mô tả chính xác mật mã Hill trên $Z_{26}$ (hình 1.6)

***Hình 1.6 Mật mã HILL***

Cho m là một số nguyên dương có định. Cho $\mathcal{P} = \mathcal{C} = (Z_{26})^m$ và cho

$\mathcal{K}$ = { các ma trận khả nghịch cấp m × m trên $Z_{26}$}

Với một khoá $K \in \mathcal{K}$ ta xác định

$$e_K(x) = xK$$

và $$d_K(y) = yK^{-1}$$

Tất cả các phép toán được thực hiện trong $Z_{26}$

### *1.1.5 Mã hoán vị (MHV)*

Tất cả các hệ mật thảo luận ở trên ít nhiều đều xoay quanh phép thaythế: các ký tự của bản rõ được thay thế bằng các ký tự khác trongbản mã. Ý tưởng của MHV là giữ các ký tự của bản rõ không thay đổi nhưng sẽ thay đôỉi vị trí của chúng bằng cách sắp xếp lại các ký tự này. MHV (còn được gọi là mã chuyển vị) đã được dùng từ hàng trăm năm nay. Thật ra thì sự phân biệt giữa MHV và MTT đã được Giovani Porta chỉ ra từ 1563. Định nghĩa hình thức cho MHV được nêu ra trên hình 1.7.

Không giống như MTT, ở đây không có các phép toán đại số nào cần thực hiện khi mã hoá và giải mã nên thích hợp hơn cả là dùng các ký tự mà không dùng các thặng dư theo modulo 26. Dưới đây là một ví dụ minh hoạ

Ví dụ 1.6

Giả sử m = 6 và khoá là phép hoán vị ( $\pi$ ) sau:

| 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|
| 3 | 5 | 1 | 6 | 4 | 2 |

***Hình 1.7 Mã hoán vị***

Cho m là mộ số nguyên dương xác định nào đó. Cho $\mathcal{P} = \mathcal{C} = (Z_{26})^m$ và cho $\mathcal{K}$ gồm tất cả các hoán vị của {1, . . ., m}. Đối một khoá $\pi$ ( tức là một hoán vị) ta xác định

$$e_\pi(x_1, \ldots, x_m) = (x_{\pi(1)}, \ldots, x_{\pi(m)})$$

và

$$d_\pi(x_1, \ldots, x_m) = (y_{\pi^{-1}(1)}, \ldots, y_{\pi^{-1}(m)})$$

trong đó $\pi^{-1}$ là hoán vị ngược của $\pi$

Khi đó phép hoán vị ngược $\pi^{-1}$ sẽ là:

| 1 | 2 | 3 | 4` | 5 | 6 |
|---|---|---|---|---|---|
| 3 | 6 | 1 | 5 | 2 | 4 |

Bây giờ giả sử có bản rõ

*Shesellsseashellsbytheseashore*

Trước tiên ta nhóm bản rõ thành các nhóm 6 ký tự:

shesel | lsseas | hellsb | ythese | ashore

Bây giờ mỗi nhóm 6 chữ cái được sắp xếp lại theo phép hoán vị π, ta có:

EESLSH | SALSES | LSHBLE | HSYEET | HRAEOS

Như vậy bản mã là

EESLSH SALSES LSHBLE HSYEET HRAEOS

Như vậy bản mã đã được mã theo cách tương tự banừg phép hoán vị đảo $\pi^{-1}$.

Thực tế mã hoán vị là trường hợp đặc biệt của mật mã Hill. Khi cho phép hoán vị π của tập {1, . . . ,m}, ta có thể xác định một ma trận hoán vị m × m thích hợp $K_\pi = \{ k_{i,j}\}$ theo công thức:

$$k_{i,j} = \begin{cases} 1 & \text{nếu } j = \pi(i) \\ 0 & \text{với các trường hợp còn lại} \end{cases}$$

( ma trận hoán vị là ma trận trong đó mỗi hàng và mỗi cột chỉ có một số "1", còn tất cả các giá trị khác đều là số "0". Ta có thể thu được một ma trận hoán vị từ ma trận đơn vị bằng cách hoán vị các hàng hoặc cột).

Dễ dàng thấy rằng, phép mã Hill dùng ma trận $K_\pi$ trên thực tế tương đương với phép mã hoán vị dùng hoán vị π. Hơn nữa $K^{-1}{}_\pi = K_\pi{}^{-1}$ tức ma trận nghịch đảo của $K_\pi$ là ma trận hoán vị xác định theo hoán vị $\pi^{-1}$. Như vậy, phép giải mã Hill tương đương với phép giải mã hoán vị.

Đối với hoán vị π được dung trong ví dun trên, các ma trận hoán vị kết hợp là:

$$K_\pi = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} \quad \text{và } K^{-1}{}_\pi = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Bạn đọc có thể kiểm tra để thấy rằng, tích của hai ma trạn này là một ma trận đơn vị.

### *1.1.7 Các hệ mã dòng*

Trong các hệ mật nghiên cứu ở trên, cácb phần tử liên tiếp của bản rõ đều được mã hoá bằng cùng một khoá K. Tức xâu bản mã y nhạn được có dạng:

$$y = y_1y_2 \ldots = e_K(x_1)\, e_K(x_2) \ \ldots$$

Các hệ mật thuộc dạng này thường được gọi là các mã khối. Một quan điểm sử dụng khác là mật mã dòng. Ý tưởng cơ bản ở đây là tạo ra một dòng khoá $z = z_1z_2 \ldots$ và dùng nó để mã hoá một xâu bản rõ $x = x_1x_2 \ldots$ theo quy tắc:

$$y = y_1y_2 \ldots = e_{z1}(x_1)\, e_{z2}(x_1) \ldots$$

Mã dòng hoạt động như sau. Giả sử $K \in \mathcal{K}$ là khoá và $x = x_1x_2 \ldots$ là xâu bản rõ. Hàm $f_i$ được dùng để tạo $z_i$ ($z_i$ là phần tử thứ i của dòng khoá) trong đó $f_i$ là một hàm của khoá K và i-1 là ký tự đầu tiên của bản rõ:

$$z_i = f_i\,(K, x_1, \ldots, x_{i-1})$$

Phần tử $z_i$ của dòng khoá được dùng để mã $x_i$ tạo ra $y_i = e_{iz}(x_i)$. Bởi vậy, để mã hoá xâu bản rõ $x_1\, x_2 \ldots$ ta phải tính liên tiếp: $z_1, y_1, z_2, y_2$ ...

Việc giải mã xâu bản mã $y_1y_2 \ldots$ có thể được thực hiện bằng cách tính liên tiếp: $z_1, x_1, z_2, x_2$ ...

Sau đây làb định nghĩa dưới dạng toán học:

***Định nghĩa 1.6.***

*Mật mã dòng là một bộ $(\mathcal{P},\mathcal{C},\mathcal{K},\mathcal{L},\mathcal{F},\mathcal{E},\mathcal{D})$ thoả mãn dược các điều kiện sau:*

1. *$\mathcal{P}$ là một tập hữu hạn các bản rõ có thể.*
2. *$\mathcal{C}$ là tập hữu hạn các bản mã có thể.*
3. *$\mathcal{K}$ là tập hữu hạn các khoá có thể ( không gian khoá)*
4. *$\mathcal{L}$ là tập hữu hạn các bộ chữ của dòng khoá.*
5. *$\mathcal{F} = (f_1 f_2 \ldots)$ là bộ tạo dòng khoá. Với $i \geq 1$*

$$f_i : \mathcal{K} \times \mathcal{P}^{i-1} \rightarrow \mathcal{L}$$

6. *Với mỗi $z \in \mathcal{L}$ có một quy tắc mã $e_z \in \mathcal{E}$ và một quy tắc giải mã tương ứng $d_z \in \mathcal{D}$. $e_z : \mathcal{P} \rightarrow \mathcal{C}$ và $d_z : \mathcal{C} \rightarrow \mathcal{P}$ là các hàm thoả mãn $d_z(e_z(x)) = x$ với mọi bản rõ $x \in \mathcal{P}$.*

Ta có thể coi mã khối là một trường hợp đặc biệt của mã dòng trong đó dùng khoá không đổi: $Z_i = K$ với mọi $i \geq 1$.

Sau đây là một số dạng đặc biệt của mã dòng cùng với các ví dụ minh hoạ. Mã dòng được gọi là đồng bộ nếu dòng khoá không phụ thuộc vào xâu bản rõ, tức là nếu dòng khoá đực tạo ra chỉ là hàm của khoá K. Khi đó ta coi K là một "mần" để mở rộng thành dòng khoá $z_1 z_2 \ldots$

Một hệ mã dòng được gọi là tuần hoàn với chu kỳ d nếu $z_{i+d} = z_i$ với số nguyên $i \geq 1$. Mã Vigenère với độ dài từ khoá m có thể coi là mã dòng tuần hoàn với chu kỳ m. Trong trường hợp này, khoá là $K = (k_1, \ldots k_m)$. Bản thân K sẽ tạo m phần tử đầu tiên của dòng khoá: $z_i = k_i$, $1 \leq i \leq m$. Sau đó dòng khoá sẽ tự lặp lại. Nhận thấy rằng, trong mã dòng tương ứng với mật mã Vigenère, các hàm mã và giải mã được dùng giống như các hàm mã và giải mã được dùng trong MDV:

$$e_z(x) = x+z \text{ và } d_z(y) = y-z$$

Các mã dòng thường được mô tả trong các bộ chữ nhi phân tức là $\mathcal{P}= \mathcal{C}=\mathcal{L}= Z_2$. Trong trường hợp này, các phép toán mã và giải mã là phép cộng theo modulo 2.

$$e_z(x) = x +z \bmod 2 \text{ và } d_z(x) = y +z \bmod 2.$$

Nếu ta coi "0" biểu thị giá trị "sai" và "1" biểu thị giá trị "đúng" trong đại số Boolean thì phép cộng theo moulo 2 sẽ ứng với phép hoặc có loại trừ. Bởi vậy phép mã (và giải mã ) dễ dàng thực hiện bằng mạch cứng.

Ta xem xét một phương pháp tạo một dòng khoá (đồng bộ ) khác. Giả sử bắt đầu với $(k_1, \ldots, k_m)$ và $z_i = k_i$, $1 \leq i \leq m$ ( cũng giống như trước đây), tuy nhiên bây giờ ta tạo dòng khoá theo một quan hệ đệ quy tuyến tính cấp m:

$$z_{i+m} = \sum_{j=0}^{m-1} c_j z_{i+j} \bmod 2$$

trong đó $c_0, \ldots, c_{m-1} \in Z_2$ là các hằng số cho trước.

*Nhận xét:*

Phép đệ quy được nói là có bậc m vì mỗi số hạng phụ thuộc vào m số hạng đứng trước. Phép đệ quy này là tuyến tính bởi vì $Z_{i+m}$ là một hàm tuyến tính của các số hạng đứng trước. Chú ý ta có thể lấy $c_0 = 1$ mà không làm mất tính tổng quát. Trong trường hợp ngược lại phép đệ quy sẽ là có bậc m-1.

Ở đây khoá K gồm 2m giá trị $k_1, \ldots, k_m, c_0, \ldots, c_{m-1}$. Nếu $(k_1, \ldots, k_m)= (0, \ldots, 0)$ thì dòng khoá sẽ chứa toàn các số 0. Dĩ nhiên phải tránh điều này vì khi đó bản mã sẽ đồng nhất với bản rõ. Tuy nhiên nếu chọn thích hợp các hằng số $c_0, \ldots, c_{m-1}$ thì một véc tơ khởi đầu bất kì khác $(k_1, \ldots, k_m)$ sẽ tạo nên một dòng khoá có chu kỳ $2^m$ -1. Bởi vậy một khoá ngắn sẽ tạo nên một dòng khoá có chu kỳ rất lớn. Đây là một tính chất rất đáng lưu tâm vì ta sẽ thấy ở phần sau, mật mã Vigenère có thể bị thám nhờ tận dụng yếu tố dòng khoá có chu kỳ ngắn.

Sau đây là một ví dụ minh hoạ:

*Ví dụ 1.7*

Giả sử m = 4 và dòng khoá được tạo bằng quy tắc:

$$z_{i+4} = z_i + z_{i+1} \bmod 2$$

Nếu dòng khoá bắt đầu một véc tơ bất kỳ khác với véc tơ (0,0,0,0) thì ta thu được dòng khoá có chu kỳ 15. Ví dụ bắt đầu bằng véc tơ (1,0,0,0), dòng khoá sẽ là:

1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1

Một véc tơ khởi đầu khác không bất kỳ khác sẽ tạo một hoán vị vòng (cyclic) của cùng dòng khoá.

Một hướng đáng quan tâm khác của phương pháp tạo dòng khoá hiệu quả bằng phần cứng là sử dụng bộ ghi dịch hồi tiếp tuyến tính (hay LFSR). Ta dùng một bộ ghi dịch có m tầng. Véc tơ $(k_1, \ldots, k_m)$ sẽ được dùng để khởi tạo ( đặt các giá trị ban đầu) cho thanh ghi dịch. Ở mỗi đơn vị thời gian, các phép toán sau sẽ được thực hiện đồng thời.

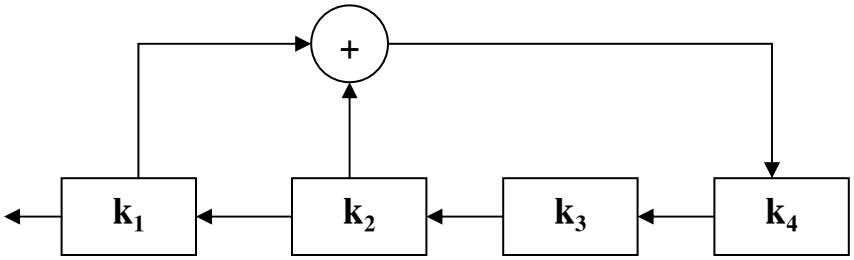
1. $k_1$ được tính ra dùng làm bit tiếp theo của dòng khoá.
2. $k_2, \ldots, k_m$ sẽ được dịch một tầng về phía trái.
3. Giá trị mới của  sẽ được tính bằng:

$$\sum_{j=0}^{m-1} c_j k_{j+1}$$

(đây là hồi tiếp tuyến tính)

Ta thấy rằng thao tác tuyến tính sẽ được tiến hành bằng cách lấy tín hiệu ra từ một số tầng nhất định của thanh ghi (được xác định bởi các hằng số $c_j$ có giá trị "1" ) và tính tổng theo modulo 2 ( là phép hoặc loại trừ ). Hình 1.8 cho mô tả của LFSR dùng để tạo dòng khoá cho ví dụ 1.7.

***Hình 1.8 Thanh ghi dịch hồi tiếp tuyến tính (LFSR)***




Một ví dụ về mã dòng không đồng bộ là mã khoá tự sinh được cho ở hình 1.9. Hình như mật mã này do Vigenère đề xuất.

***Hình 1.9. Mật mã khoá tự sinh***

Cho $\mathcal{P} = C = \mathcal{K} = \mathcal{L} = Z_{26}$
Cho $z_1 = K$ và $z_i = x_{i-1}$ $(i \geq 2)$
Với $0 \leq z \leq 25$ ta xác định
$e_z(x) = x + z \bmod 26$
$d_z(y) = y - z \bmod 26$
$(x,y \in Z_{26})$

Lý do sử dụng thuật nghữ "khoá tự sinh" là ở chỗ: bản rõ được dùng làm khoá ( ngoài "khoá khởi thuỷ" ban đầu K).

Sau đây là một ví dụ minh hoạ

*Ví dụ 1.8:*

Giả sử khoá là k = 8 và bản rõ là *rendezvous*. Trước tiên ta biến đổi bản rõ thành dãy các số nguyên:

17 4 13 3 4 25 21 14 20 18

Dòng khoá như sau:

8 17 4 13 3 4 25 21 14 20

Bây giờ ta cộng các phần tử tương ứng rồi rút gọn theo modulo 26:

25 21 17 16 7 3 20 9 8 12

Bản mã ở dạng ký tự là: ZVRQHDUJIM
Bây giờ ta xem Alice giải mã bản mã này như thế nào. Trước tiên Alice biến đổi xâu kí tự thành dãy số:

25 21 17 16 7 3 20 9 8 12

Sau đó cô ta tính:

$$x_1 = d_8(25) = 25 - 8 \bmod 26 = 17$$

và $$x_2 = d_{17}(21) = 21 - 17 \bmod 26 = 4$$

và cứ tiếp tục như vậy. Mỗi khi Alice nhận được một ký tự của bản rõ, cô ta sẽ dùng nó làm phần tử tiếp theo của dòng khoá.

Dĩ nhiên là mã dùng khoá tự sinh là không an toàn do chỉ có 26 khoá.
Trong phần sau sẽ thảo luận các phương pháp thám các hệ mật mã mà ta đã trình bày.

---

## 1.2 MÃ THÁM CÁC HỆ MÃ CỔ ĐIỂN

Trong phần này ta sẽ bàn tới một vài kỹ thuật mã thám. Giả thiết chung ở đây là luôn coi đối phương Oscar đã biết hệ mật đang dùng. Giả thiết này được gọi là nguyên lý Kerekhoff. Dĩ nhiên, nếu Oscar không biết hệ mật được dùng thì nhiệm vụ của anh ta sẽ khó khăn hơn. Tuy nhiên ta không muốn độ mật của một hệ mật lại dựa trên một giả thiết không chắc chắn là Oscar không biết hệ mật được sử dụng. Do đó, mục tiêu trong thiết kế một hệ mật là phải đạt được độ mật dưới giả thiết Kerekhoff.

Trước tiên ta phân biệt các mức độ tấn công khác nhau vào các hệ mật. Sau đây là một số loại thông dụng nhất.

*Chỉ có bản mã:*

Thám mã chỉ có xâu bản mã y.

*Bản rõ đã biết:*

Thám mã có xâu bản rõ x và xâu bản mã tương ứng y.

*Bản rõ được lựa chọn:*

Thám mã đã nhận được quyền truy nhập tạm thời vào cơ chế mã hoá. Bởi vậy, thám mã có thể chọn một xâu bản rõ x và tạo nên xâu bản mã y tương ứng.

*Bản mã được lựa chọn:*

Thám mã có được quyền truy nhập tạm thời vào cơ chế giải mã. Bởi vậy thám mã có thể chọn một bản mã y và tạo nên xâu bản rõ x tương ứng.

Trong mỗi trường hợp trên, đối tượng cần phải xác định chính là khoá đã sử dụng. Rõ ràng là 4 mức tấn công trên đã được liệt kê theo độ tăng của sức mạnh tấn công. Nhận thấy rằng, tấn công theo bản mã được lựa chọn là thích hợp với các hệ mật khoá công khai mà ta sẽ nói tới ở chương sau.

Trước tiên, ta sẽ xem xét cách tấn công yếu nhất, đó là tấn công chỉ có bản mã. Giả sử rằng, xâu bản rõ là một văn bản tiếng Anh thông thường không có chấm câu hoặc khoảng trống ( mã thám sẽ khó khăn hơn nếu mã cả dấu chấm câu và khoảng trống).

Có nhiều kỹ thuật thám mã sử dụng các tính chất thống kê của ngôn ngữ tiếng Anh. Nhiều tác giả đã ước lượng tần số tương đối của 26 chữ cái theo các tính toán thống kê từ nhiều tiểu thuyết, tạp chí và báo. Các ước lượng trong bảng 1.1 lấy theo tài liệu của Beker và Piper.

***Bảng 1.1 Xác suất xuất hiện của 26 chữ cái:***

| Kí tự | Xác suất | Kí tự | Xác suất | Kí tự | Xác suất |
|---|---|---|---|---|---|
| A | .082 | J | .002 | S | .063 |
| B | .015 | K | .008 | T | .091 |
| C | .028 | L | .040 | U | .028 |
| D | .043 | M | .024 | V | .010 |
| E | .0127 | N | .067 | W | .023 |
| F | .022 | O | .075 | X | .001 |
| G | .020 | P | .019 | Y | .020 |
| H | .061 | Q | .001 | Z | .001 |
| I | .070 | R | .060 | | |

Từ bảng trên, Beker và Piper phân 26 chữ cái thành 5 nhóm như sau:

1. E: có xác suất khoảng 1,120
2. T, A, O, I, N, S, H, R : mỗi ký tự có xac suất khoảng 0,06 đến 0,09
3. D, L : mỗi ký tự có xác suất chừng 0,04
4. C, U, M, W, F, G, Y, P, B: mỗi ký tự có xác suất khoảng 0,015 đến 0,023
5. V, K, J, X, Q, Z mỗi ký tự có xác suất nhỏ hơn 0,01

Việc xem xét các dãy gồm 2 hoặc 3 ký tự liên tiếp ( được gọi là bộ đôi - diagrams và bộ ba - Trigrams )cũng rất hữu ích. 30 bộ đôi thông dụng nhất ( theo hứ tự giảm dần ) là: TH, HE, IN, ER, AN, RE, ED, ON, ES, ST, EN, AT, TO, NT, HA, ND, OU, EA, NG, AS, OR, TI, IS, ET, IT, AR, TE, SE, HI và OF. 12 bộ ba thông dụng nhất (theo thứ tự giảm dần ) là: THE, ING, AND, HER, ERE, ENT, THA, NTH, WAS, ETH, FOR và DTH.

### *1.2.1 Thám hệ mã Affine*

Mật mã Affine là một ví dụ đơn giản cho ta thấy cách thám hệ mã nhờ dùng các số liệu thống kê. Giả sử Oscar đã thu trộm được bản mã sau:

***Bảng 1.2: Tần suất xuất hiện của 26 chữ cái của bản mã***

| Kí tự | Tần suất | Kí tự | Tần suất | Kí tự | Tần suất | Kí tự | Tần suất |
|---|---|---|---|---|---|---|---|
| A | 2 | H | 5 | O | 1 | U | 2 |
| B | 1 | I | 0 | P | 3 | V | 4 |
| C | 0 | J | 0 | Q | 0 | W | 0 |
| D | 6 | K | 5 | R | 8 | X | 2 |
| E | 5 | L | 2 | S | 3 | Y | 1 |
| F | 4 | M | 2 | T | 0 | Z | 0 |
| G | 0 | N | 1 | | | | |

Ví Dụ 1.9:

Bản mã nhận được từ mã Affine:

FMXVEDRAPHFERBNDKRXRSREFMORUDSDKDVSHVUFEDKPKDLYEVLRHHRH

Phân tích tần suất của bản mã này được cho ở bảng 1.2

Bản mã chỉ có 57 ký tự. Tuy nhiên độ dài này cũng đủ phân tích thám mã đối với hệ Affine. Các ký tự có tần suất cao nhất trong bản mã là: R ( 8 lần xuất hiện), D (6 lần xuất hiện ), E, H, K (mỗi ký tự 5 lần ) và F, S, V ( mỗi ký tự 4 lần).

Trong phỏng đoán ban đầu, ta giả thiết rằng R là ký tự mã của chữ e và D là kí tự mã của t, vì e và t tương ứng là 2 chữ cái thông dụng nhất. Biểu thị bằng số ta có: $e_K(4) = 17$ và $e_K(19) = 3$. Nhớ lại rằng $e_K(x) = ax + b$ trong đó a và b là các số chưa biết. Bởi vậy ta có hai phương trình tuyến tính hai ẩn:

$$4a + b = 17$$
$$19a + b = 3$$

Hệ này có duy nhất nghiệm a = 6 và b = 19 ( trong $Z_{26}$ ). Tuy nhiên đây là một khoá không hợp lệ do UCLN(a,26) = 2 1. Bởi vậy giả thiết của ta là không đúng.

Phỏng đoán tiếp theo của ta là: R là ký tự mã của e và E là mã của t. Thực hiện như trên, ta thu được a =13 và đây cũng là một khoá không hợp lệ. Bởi vậy ta phải thử một lần nữa: ta coi rằng R là mã hoá của e và H là mã hoá của t. Điều này dẫn tới a = 8 và đây cũng là một khoá không hợp lệ. Tiếp tục, giả sử rằng R là mã hoá của e và K là mã hoá của t. Theo giả thiết này ta thu được a = 3 và b = 5 là khóa hợp lệ.

Ta sẽ tính toán hàm giải mã ứng với K = (3,5) và gải mã bản mã để xem liệu có nhận được xâu tiếng Anh có nghĩa hay không. Điều này sẽ khẳng định tính hợp lệ của khoá (3,5). Sau khi thực hiện các phép toán này, ta có $d_K(y) = 9y - 19$ và giải mã bản mã đã cho, ta được:

*algorithmsarequitegeneraldefinitionsof*
*arithmeticprocesses*

Như vậy khoá xác định trên là khoá đúng.

### 1.2.2. Thám hệ mã thay thế

Sau đâyta phân tích một tình huống phức tạp hơn, đó là thay thế bản mã sau

Ví dụ 1.10

Bản mã nhận được từ MTT là:

YIFQFMZRWQFYVECFMDZPCVMRZWNMDZVEJBTXCDDUMJ
NDIFEFMDZCDMQZKCEYFCJMYRNCWJCSZREXCHZUNMXZ
NZUCDRJXỷYMTMEYIFZWDYVZVYFZUMRZCRWNZDZJT
XZWGCHSMRNMDHNCMFQCHZJMXJZWIEJYUCFWDINZDIR

Phân tích tần suất của bản mã này đươch cho ở bảng 1.3.

***Bảng 1.3. Tần suất xuất hiện của 26 chưz cái trong bản mã.***

| Ký tự | Tần suất | Ký tự | Tần suất | Ký tự | Tần suất | Ký tự | Tần suất |
|---|---|---|---|---|---|---|---|
| A | 0 | H | 4 | O | 0 | U | 5 |
| B | 1 | I | 5 | P | 1 | V | 5 |
| C | 15 | J | 11 | Q | 4 | W | 8 |
| D | 13 | K | 1 | R | 10 | X | 6 |
| E | 7 | L | 0 | S | 3 | Y | 10 |
| F | 11 | M | 16 | T | 2 | Z | 20 |
| G | 1 | N | 9 | | | | |

Do Z xuất hiện nhiều hơn nhiều so với bất kỳ một ký tự nào khác trong bản mã nên có thể phỏng đoán rằng, $d_Z(Z) = e$. các ký tự còn lại xuất hiện ít nhất 10 lần ( mỗi ký tự ) là C, D, F, J, R, M, Y. Ta hy vọng rằng, các ký tự này là mã khoá của ( một tập con trong ) t, a, c, o, i, n, s, h, r, tuy nhiên sự khác biệt về tần suất không đủ cho ta có được sự phỏng đoán thích hợp.

Tới lúc này ta phải xem xét các bộ đôi, đặc biệt là các bộ đôi có dạng -Z hoặc Z- do ta đã giả sử rằng Z sẽ giải mã thành e. Nhận thấy rằng các bộ đôi thường gặp nhất ở dạng này là DZ và ZW ( 4 lần mỗi bộ ); NZ và ZU ( 3 lần mỗi bộ ); và RZ, HZ, XZ, FZ, ZR, ZV, ZC, ZD và ZJ ( 2 lần mỗi bộ ). Vì ZW xuất hiện 4 lần còn WZ không xuất hiện lần nào và nói chung W xuất hiện ít hơn so với nhiều ký tự khác, nên ta có thể phỏng đoán là $d_K(W) = d$. Vì DZ xuất hiện 4 lần và ZD xuất hiện 2 lần nên ta có thể nghĩ rằng $d_K(D) \in \{r,s,t\}$, tuy nhiên vẫn còn chưa rõ là ký tự nào trong 3 ký tự này là ký tự đúng.

Nêu tiến hành theo giả thiết $d_K(Z) = e$ và $d_K(W) = d$ thì ta phải nhìn trở lại bản mã và thấy rằng cả hai bộ ba ZRW và RZW xuất hiện ở gần đầu của bản mã và RW xuất hiện lại sau đó vì R thường xuất hiện trong bản mã và nd là một bộ đôi thường gặp nên ta nên thử $d_K(R) = n$ xem là một khả năng thích hợp nhất.

Tới lúc này ta có:

```
- - - - - - end - - - - - - - - - e - - - - ned- - - e - - - - - - - - -
YIFQFMZRWQFYVECFMDZPCVMRZWNMDZVEJBTXCDDUMJ
- - - - - - - - e- - - - e - - - - - - - - n - - d - - - en - - - - e - - - -e
NDIFEFMDZCDMQZKCEYFCJMYRNCWJCSZREXCHZUNMXZ
- e - - - n - - - - - n - - - - - - ed - - - e - - - - - - ne - nd- e- e - -
NZUCDRJXYYSMRTMEYIFZWDYVZVYFZUMRZCRWNZDZJJ
- ed - - - - - n - - - - - - - - - - e - - - ed - - - - - - - d - - - e - - n
XZWGCHSMRNMDHNCMFQCHZJMXJZWIEJYUCFWDJNZDIR
```

Bước tiếp theo là thử $d_K(N) = h$ vì NZ là một bộ đôi thường gặp còn ZN không xuất hiện. Nếu điều này đúng thì đoạn sau của bản rõ ne - ndhe sẽ gợi ý rằng $d_K(C) = a$. Kết hợp các giả định này, ta có:

```
- - - - - -end- - - - - a- - -e -a - - nedh- -e- - - - - -a - - - - -
YIFQFMZRWQFYVECFMDZPCVMRZWNMDZVEJBTXCDDUMJ
h -  - - - - - - a- - - e - a- - - a - - - nhad - a - -en -a -  e - h- -e
NDIFEFMDZCDMQZKCEYFCJMYRNCWJCSZREXCHZUNMXZ
he - a - n- - - - - - n - - - - - - ed - - - e- - - e - - neandhe  -e - -
NZUCDRJXYYSMRTMEYIFZWDYVZVYFZUMRZCRWNZDZJJ
- ed  - a - - -nh - - - ha - - - a- e - - - - ed - - - - -a -d - - he- -n
XZWGCHSMRNMDHNCMFQCHZJMXJZWIEJYUCFWDJNZDIR
```

Bây giờ ta xét tới M là ký tự thường gặp nhất sau Z. Đoạn bản mã RNM mà ta tin là sẽ giải mã thành nh- gợi ý rằng h- sẽ bắt đầu một từ, bởi vậy chắc là M sẽ biểu thị môt nguyên âm. Ta đã sử dụng a và e, bởi vậy, phỏng đoán rằng $d_K(M) = i$ hoặc o. Vì ai là bộ đôi thường gặp hơn ao nên bộ đôi CM trong bản mã gợi ý rằng, trước tiên nên thử $d_K(M) = i$. Khi đó ta có:

```
- - - - -iend- - - - - a -i - e -a -inedhi - e- - - - - -a - - -i  -
YIFQFMZRWQFYVECFMDZPCVMRZWNMDZVEJBTXCDDUMJ
h - - - - - i - ea - i - e  -a - - -a  - i -nhad -a - en - -a - e -hi -e
NDIFEFMDZCDMQZKCEYFCJMYRNCWJCSZREXCHZUNMXZ
he  - a - n - - - - -in -i - - - -  ed - - -e - - - e - ineandhe - e - -
NZUCDRJXYYSMRTMEYIFZWDYVZVYFZUMRZCRWNZDZJJ
- ed - a - - inhi - - hai - - a - e - i- -ed- - - - - a - d - - he - -n
XZWGCHSMRNMDHNCMFQCHZJMXJZWIEJYUCFWDJNZDIR
```

Tiếp theo thử xác định xem chữ nào được mã hoá thành o. Vì o là một chữ thường gặp nên giả định rằng chữ cái tương ứng trong bản mã là một trong các ký tự D,F,J,Y. Y có vẻ thích hợp nhất, nếu không ta sẽ có các xâu dài các nguyên âm, chủ yếu là aoi ( từ CFM hoặc CJM ). Bởi vậy giả thiết rằng $d_K(Y) = o$.

Ba ký tự thường gặp nhất còn lại trong bản mã là D,F,J, ta phán đoán sẽ giải mã thành r,s,t theo thứ tự nào đó. Hai lần xuất hiện của bộ ba NMD gợi ý rằng $d_K(D) = s$ ứng với bộ ba his trong bản rõ ( điều này phù hợp với giả định trước kia là $d_K(D) \in \{r,s,t\}$ ). Đoạn HNCMF có thể là bản mã của chair, điều này sẽ cho $d_K(F) = r$ (và $d_K(H) = c$ ) và bởi vậy (bằng cách loại trừ ) sẽ có $d_K(J) = t$.
Ta có:

o- r - riend - ro - - arise - a - inedhise - - t - - - ass - it
YIFQFMZRWQFYVECFMDZPCVMRZNMDZVEJBTXCDDUMJ
hs - r - riseasi - e - a - orationhadta - - en - -ace - hi - e
NDIFEFMDZCDMQZKCEYFCJMYRNCWJCSZREZCHZUNMXZ
he - asnt - oo - in - i - o - redso - e - ore - ineandhesett
NZUCDRJXYYSMRTMEYIFZWDYVZVYFZUMRZCRWNZDZJJ
- ed - ac - inhischair - aceti - ted - - to - ardsthes - n
XZWGCHSMRNMDHNCMFQCHZJMXJZWIEJYUCFWDJNZDIR

Bây giờ việc xác định bản rõ và khoá cho ví dụ 1.10 không còn gì khó khăn nữa. Bản rõ hoàn chỉnh như sau:

*Our friend from Pais examined his empty glass with surprise, as if evaporation had taen place while he wasn't looking. I poured some more wine and he settled back in his chair, face tilted up towards the sun.*

### 1.2.3. Thám hệ mã Vigenère

Trong phần này chúng ta sẽ mô tả một số phương pháp thám hệ mã Vigenère. Bước đầu tiên là phải xác định độ dài từ khoá mà ta ký hiệu là m. Ở đây dùng hai kỹ thuật. Kỹ thuật thứ nhất là phép thử Kasiski và kỹ thuật thứ hai sử dụng chỉ số trùng hợp.

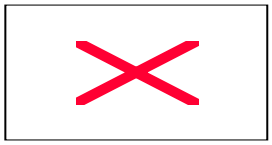
Phép thử Kasiski lần đầu tiên được Kasiski Friendrich mô tả vào năm 1863. Kỹ thuật này được xây dựng trên nhận xét là: hai đoạn giống nhau của bản rõ sẽ được mã hoá thành cùng một bản mã khi chúng xuất hiện trong bản rõ cách nhau x vị trí, trong đó $x \equiv o \bmod m$. Ngược lại, nếu ta thấy hai đoạn giống nhau của bản mã ( mỗi đoạn có độ dài ít nhất là 3 ) thì đó là một dấu hiệu tốt để nói rằng chúng tương ứng với các đoạn bản rõ giống nhau.

Phép thử Kasiski như sau. Ta tòm trong bản mã các cặp gồm các đoạn như nhau có độ dài tối thiểu là 3 và ghi lại khoảng cách giữa các vị trí bắt đầu của hai đoạn. Nếu thu được một vài giá trị $d_1$, $d_2$ ,. . . thì có thể hy vọng rằng m sẽ chia hết cho ước chung lớn nhất của các $d_i$.

Việc xác minh tiếp cho giá trị của m có thể nhận được bằng chỉ số trùng hợp. Khái niệm này đã được Wolfe Friedman đưa ra vào 1920 như sau:

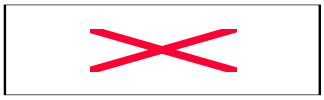
Định nghĩa 1.7.

Giả sử $x = x_1x_2 \ldots x_n$ là một xâu ký tự. Chỉ số trùng hợp của x (ký hiệu là $I_c(x)$) được định nghĩa là xác suất để hai phần tử ngẫu nhiên của x là đồng nhất. Nếu ký hiệu các tần suất của A,B,C,. . . ,Z trong x tương ứng là $f_0$,$f_1$ ,. . . $f_{25}$ , có thể chọn hai phần tử của x theo ??? cách. Với mỗi i, $0 \le i \le 25$, có ??? cách chọn hai phần tử là i. Bởi vậy ta có công thức:



Ghi chú: Hệ số nhị thức ?????? xác định số cách chọn một tập con k đối tượng từ một tập n đối tượng.

Bây giờ, giả sử x là một xâu văn bản tiếng Anh. Ta kí hiệu các xác suất xuất hiện của các kí tự A,B,. . .,Z trong bảng 1.1 là $p_0$,...$p_{25}$. Khi đó:



do xác suất để hai phần tử ngẫu nhiên đều là A là $p_0^2$, xác suất để cả hai phần tử này đều bằng B bằng $p_1^2$ . . . Tình hình tương tự cũng xảy ra nếu x là một bản mã nhận được theo một hệ mã thay thế đơn bất kì. Trong trường hợp này, từng xác suất riêng rẽ sẽ bị hoán vị nhưng tổng ??? sẽ không thay đổi.

Bây giờ giả sử có một bản mã $y = y_1y_2. \ldots y_n$ được cấu trúc theo mật mã Vigenère. Ta xác định các xâu con m của $y(y_1,y_2,. \ldots ,y_m)$ bằng cách viết ra bản mã thành một hình chữ nhật có kích thước $m\times(n/m)$. Các hàng của ma trận này là các xâu con $y_i$, $1 \le i \le m$. Nếu m thực sự là độ dài khoá thì mỗi $I_c(y_i)$ phải xấp xỉ bằng 0,065. Ngược lại, nếu m không phải là độ dài khoá thì các xâu con $y_i$ sẽ có vẻ ngẫu nhiên hơn vì chúng nhận được bằng cách mã dịch vòng với các khoá khác nhau. Xét thấy rằng, một xâu hoàn toàn ngẫu nhiên sẽ có:

Hai giá trị 0,065 và 0,038 đủ cách xa nhau để có thể xác định được độ dài từ khoá đúng ( hoặc xác nhận giả thuyết đã được làm theo phép thử Kasiski). Hai kỹ thuật này sẽ được minh hoạ qua ví dụ dưới đây:

Ví dụ 1.11.

Bản mã nhận được từ mật mã Vigenère.

CHEEVOAHMAERATBTAXXWTNXBEEOPHBSBQMQEQERBW
RVXUOAKXAOSXXWEAHBWGJMMQMNKGRFVGXWTRZXWIAK
LXFPSKAUTEMNDCMGTSXMXBTUIADNGMGPSRELXNJELX
VRVPRTULHDNQWTWDTYGBPHXTFEALJHASVBFXNGLLCHR
ZBWELEKMSJIKNBHWRJGNMGJSGLXFEYPHAGNRBIEQJT
AMRVLCRREMNDGLXRRIMGNSNRWCHRQHAEYEVTAQEBBI
PEEWEVKAKOEWADREMXMTBHHCHRTKDNVRZCHRCLQOHP
WQAIIWXNRMGWOIIFKEE

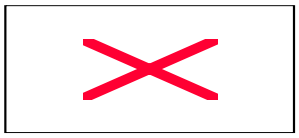
Trước tiên, ta hãy thử bằng phép thử Kasiski xâu bản mã CHR xuất hiện ở bốn vị trí trong bản mã, bắt đầu ở các vị trí 1, 166,236 và 286. Khoảng cách từ lần xuất hiện đầu tiên tới 3 lần xuất hiện còn lại tương ứng là 165,235 và 285. UCLN của 3 số nguyên này là 5, bởi vậy giá trị này rất có thể là độ dài từ khoá.

Ta hãy xét xem liệu việc tính các chỉ số trùng hợp có cho kết luận tương tự không. Với m = 1 chỉ số trùng hợp là 0,045. Với m = 2, có 2 chỉ số là 0,046 và 0,041. Với m = 3 ta có 0,043; 0,050; 0,047. Với m = 4 các chỉ số là 0,042; 0,039; 0,046; 0,040. Với m = 5 ta có các giá trị 0,063; 0,068; 0,069; 0,061 và 0,072. Điều này càng chứng tỏ rằng độ dại từ khoá là 5.

Với giả thiết trên, làm như thế nào để xác định từ khoá? Ta sẽ sử dụng khái niệm chỉ số trùng hợp tương hỗ của hai xâu sau:

***Định nghĩa 1.8.***

Giả sử $x = x_1x_2. . .x_n$ và $y = y_1y_2. . .y_{n'}$ là các xâu có n và n' kí tự anphabet tương ứng. Chỉ số trùng hợp tương hỗ của x và y ( kí hiệu là $MI_c(x,y)$) được xác định là xác suất để một phần tử ngẫu nhiên của x giống với một phần tử ngẫu nhiên của y. Nếu ta kí hiệu các tần suất của A,B,. . .,Z trong x và y tương ứng là $f_0,f_1,. . .,f_{25}$ thì $MI_c(x,y)$ sẽ được tính bằng:



Với các giá trị m đã xác định, các xâu con $y_i$ thu được bằng mã dịch vòng bản rõ. Giả sử K = ($k_1$,$k_2$,. . .,$k_m$) là từ khoá. Ta sẽ xem xét có thể đánh giá $MI_c(y_i,y_j)$ như thế nào. Xét một kí tự ngẫu nhiên trong $y_i$ và một kí tự ngẫu nhiên trong $y_j$ . Xác suất để cả hai kí tự là A bằng $p_{-ki}\, p_{-kj}$, xác suất để cả hai là B bằng $p_{1-ki}\, p_{1-kj}$,. . .( Cần chú ý rằng tất cả các chỉ số dưới đều được rút gọn theo modulo 26). Bởi vậy có thể ước lượng rằng:

$$MI_c(y_i, y_i) \approx \sum_{h=0}^{25} p_{h-ki}\, p_{h-kj} = \sum_{h=0}^{25} p_h\, p_{h+ki-kj}$$

Ta thấy rằng, giá trị ước lượng này chỉ phụ thuộc vào kiếu hiệu $k_i$-$k_j$ mod 26 ( được gọi là độ dịch tương đối của $y_i$ và $y_j$). Cũng vậy, ta thấy rằng:

$$\sum_{h=0}^{25} p_h\, p_{h+1} = \sum_{h=0}^{25} p_h\, p_{h-1}$$

Bởi vậy độ dịch tương đối *l* sẽ dẫn đến cùng một ước lượng $MI_c$ như độ dịch tương đối 26-*l* .

Ta lập bảng các ước lượng cho độ dịch tương đối trong phạm vi từ 0 đến 13.( Xem bảng 1.4).

***Bảng 1.4. Các chỉ số trùng hợp tương hỗ tính được.***

| **Độ dịch tương đối** | **Giá trị tính được của $MI_c$** |
|---|---|
| 0 | 0.065 |
| 1 | 0,039 |
| 2 | 0,032 |
| 3 | 0,034 |
| 4 | 0,044 |
| 5 | 0,033 |
| 6 | 0,036 |
| 7 | 0,039 |
| 8 | 0,034 |
| 9 | 0,034 |
| 10 | 0,038 |
| 11 | 0,045 |
| 12 | 0,039 |
| 13 | 0,043 |

Xét thấy rằng, nếu độ dịch tương đối khác 0 thì các ước lượng này thay đổi trong khoảng từ 0.031 đến 0,045; ngược lại nếu độ dịch tương đối bằng 0 thì ước lượng bằng 0,065. Có thể dùng nhận xét này để tạo nên một phỏng đoán thích hợp cho $l = k_i - k_j$ (độ dịch tương đối của $y_i$ và $y_j$) như sau: Giả sử cố định $y_i$ và xét việc mã hoá $y_j$ bằng $e_0, e_1, e_2 \ldots$ Ta kí hiệu các kết quả bằng $y_j^0, y_j^1, \ldots$ Dễ dàng dùng các chỉ số $MI_c(y_i, y_j^g)$, $0 \le g \le 25$ theo công thức sau:

$$MI_c(x, y^g) = \frac{\sum_{i=0}^{25} f_i f'_{i-g}}{n.n'}$$

Khi $g = l$ thì $MI_c$ phải gần với giá trị 0,065 vì độ dịch tương đối của $y_i$ và $y_j$ bằng 0. Tuy nhiên, với các giá trị $g \neq l$ thì $MI_c$ sẽ thay đổi giữa 0,031 và 0,045.

Bằng kỹ thuật này, có thể thu được các độ dịch tương đối của hai xâu con $y_i$ bất kỳ. Vấn đề còn lại chỉ là 26 từ khoá có thể và điều này dễ dàng tìm được bằng phương pháp tìm kiếm vét cạn.

Trở lại ví dụ 1.11 để minh hoạ.

Ví dụ 1.11( tiếp ):

Ở trên đã giả định rằng, độ dài từ khoá là 5. Bây giờ ta sẽ thử tính các độ dịch tương đối. Nhờ máy tính, dễ dàng tính 260 giá trị $MI_c(y_i, y_j^g)$, trong đó $1 \le i \le j \le 5$; $0 \le g \le 25$. Các giá trị này được cho trên bảng 1.5. Với mỗi cặp ( i,j), ta tìm các giá trị của $MI_c(y_i, y_j^g)$ nào gần với 0,065. Nếu có một giá trị duy nhất như vậy( Đối với mỗi cặp (i,j) cho trước), thì có thể phán đoán đó chính là giá trị độ dịch tương đối.

Trong bảng 1.5 có 6 giá trị như vậy được đóng khung. Chúng chứng tỏ khá rõ ràng là độ dịch tương đối của $y_1$ và $y_2$ bằng 9; độ dịch tương đối của $y_2$ và $y_3$ bằng 13; độ dịch tương đối của $y_2$ và $y_5$ bằng 7; độ dịch tương đối của $y_3$ và $y_5$ bằng 20; của $y_4$ và $y_5$ bằng 11. Từ đây có các phương trình theo 5 ẩn số $K_1, K_2, K_3, K_4, K_5$ như sau:

$$K_1 - K_2 = 9$$
$$K_1 - K_2 = 16$$
$$K_2 - K_3 = 13$$
$$K_2 - K_5 = 17$$
$$K_3 - K_5 = 20$$
$$K_4 - K_5 = 11$$

Điều này cho phép biểu thị các $K_i$ theo $K_1$ ;

$$K_2 = K_1 + 17$$
$$K_3 = K_1 + 4$$
$$K_4 = K_1 + 21$$
$$K_5 = K_1 + 10$$

Như vậy khoá có khả năng là ( $K_1$, $K_1$+17, $K_1$+4, $K_1$+21, $K_1$+10) với giá trị $K_1$ nào đó $\in Z_{26}$. Từ đây ta hy vọng rằng, từ khoá là một dịch vòng nào đó của AREVK. Bây giờ , không tốn nhiều công sức lắm cũng có thể xác định được từ khoá là JANET. Giải mã bản mã theo khoá này, ta thu được bản rõ sau:

*The almond tree was in tentative blossom. The days were longer often ending with magnificient evenings of corrugated pink skies. The hunting seasun was over, with hounds and guns put away for six months. The vineyards were busy again as the well-organized farmers treated their vinesand the more lackadaisical neighbors hurried to do the pruning they have done in November.*

***Bảng 1.5. Các chỉ số trùng hợp tương hỗ quan sát được.***

| i | j | Giá trị của $MI_c(y_j,y_j^g)$ |
|---|---|---|
| 1 | 2 | 0,028 0,027 0,028 0,034 0,039 0,037 0,026 0,025 0,052<br>0,068 0,044 0,026 0,037 0,043 0,037 0,043 0,037 0,028<br>0,041 0,041 0,041 0,034 0,037 0,051 0,045 0,042 0,036 |
| 1 | 3 | 0,039 0,033 0,040 0,034 0,028 0,053 0,048 0,033 0,029<br>0,056 0,050 0,045 0,039 0,040 0,036 0,037 0,032 0,027<br>0,037 0,047 0,032 0,027 0,039 0,037 0,039 0,035 |
| 1 | 4 | 0,034 0,043 0,025 0,027 0,038 0,049 0,040 0,032 0,029<br>0,034 0,039 0,044 0,044 0,034 0,039 0,045 0,044 0,037<br>0,055 0,047 0,032 0,027 0,039 0,037 0,039 0,035 |
| 1 | 5 | 0,043 0,033 0,028 0,046 0,043 0,044 0,039 0,031 0,026<br>0,030 0,036 0,040 0,041 0,024 0,019 0,048 **0,070** 0,044<br>0,028 0,038 0,044 0,043 0,047 0,033 0,026 |
| 2 | 3 | 0,046 0,048 0,041 0,032 0,036 0,035 0,036 0,020 0,024<br>0,039 0,034 0,029 0,040 **0,067** 0,061 0,033 0,037 0,045<br>0,033 0,033 0,027 0,033 0,045 0,052 0,042 0,030 |
| 2 | 4 | 0,046 0,034 0,043 0,044 0,034 0,031 0,040 0,045 0,040<br>0,048 0,044 0,033 0,024 0,028 0,042 0,039 0,026 0,034<br>0,050 0,035 0,032 0,040 0,056 0,043 0,028 0,028 |
| 2 | 5 | 0,033 0,033 0,036 0,046 0,026 0,018 0,043 **0,080** 0,050<br>0,029 0,031 0,045 0,039 0,037 0,027 0,026 0,031 0,039<br>0,040 0,037 0,041 0,046 0,045 0,043 0,035 0,030 |
| 3 | 4 | 0,038 0,036 0,040 0,033 0,036 0,060 0,035 0,041 0,029<br>0,058 0,035 0,035 0,034 0,053 0,030 0,032 0,035 0,036<br>0,036 0,028 0,043 0,032 0,051 0,032 0,034 0,030 |
| 3 | 5 | 0,035 0,038 0,034 0,036 0,030 0,043 0,043 0,050 0,025<br>0,041 0,051 0,050 0,035 0,032 0,033 0,033 0,052 0,031<br>0,027 0,030 **0,072** 0,035 0,034 0,032 0,043 0,027 |
| 4 | 5 | 0,052 0,038 0,033 0,038 0,041 0,043 0,037 0,048 0,028<br>0,028 0,036 **0,061** 0,033 0,033 0,032 0,052 0,034 0,027<br>0,039 0,043 0,033 0,027 0,030 0,039 0,048 0,035 |

### *1.2.4.Tấn công với bản rõ đã biết trên hệ mật Hill.*

Hệ mã Hill là một hệ mật khó pha hơn nếu tấn công chỉ với bản mã. Tuy nhiên hệ mật này dễ bị phá nếu tấn công bằng bản rõ đã biết. Trước tiên, giả sử rằng, thám mã đã biết được giá trị m đang sử dụng. Giả sử thám mã có ít nhất m cặp véc tơ khác nhau xj = $(x_{1,j}, x_{2,j}, , . . ., x_{m,j})$ và $y_j = (y_{1,j}, y_{2,j},...,y_{m,j})$ $(1 \le j \le m)$ sao cho $y_j = e_K(x_j)$, $1 \le j \le m$. Nếu xác định hai ma trận: $X = (x_{i,j})$ $Y = (y_{i,j})$ cấp m×m thì ta có phương trình ma trận Y = XK, trong đó ma trận K cấp m×m là khoá chưa biết. Với điều kiện ma trận Y là khả nghịch. Oscar có thể tính $K = X^{-1}Y$ và nhờ vậy phá được hệ mật. ( Nếu Y không khả nghịch thì cần phải thử các tập khác gồm m cặp rõ - mã).

Ví dụ 1.12.

Giả sử bản rõ *Friday* được mã hoá bằng mã Hill với m = 2, bản mã nhận được là PQCFKU.

Ta có $e_K(5,17) = (15,16)$, $e_K(8,3) = (2,5)$ và $e_K(0,24) = (10,20)$. Từ hai cặp rõ - mã đầu tiên, ta nhận được phương trình ma trận:

$$\begin{pmatrix} 15 & 16 \\ 2 & 5 \end{pmatrix} = \begin{pmatrix} 5 & 17 \\ 8 & 3 \end{pmatrix} K$$

Dùng định lý 1.3, dễ dàng tính được:

$$\begin{pmatrix} 5 & 17 \\ 8 & 3 \end{pmatrix}^{-1} = \begin{pmatrix} 9 & 1 \\ 2 & 15 \end{pmatrix}$$

Bởi vậy:

$$K = \begin{pmatrix} 9 & 1 \\ 2 & 15 \end{pmatrix} \begin{pmatrix} 15 & 16 \\ 2 & 5 \end{pmatrix} = \begin{pmatrix} 7 & 19 \\ 8 & 3 \end{pmatrix}$$

Ta có thể dùng cặp rõ - mã thứ 3 để kiểm tra kết quả này.

Vấn đề ở đây là thám mã phải làm gì nếu không biết m?. Giả sử rằng m không quá lớn, khi đó thám má có thể thử với m = 2,3,. . . cho tới khi tìm được khoá. Nếu một giá trị giả định của m không đúng thì mà trận m×m tìm được theo thuật toán đã mô tả ở trên sẽ không tương thích với các cặp rõ - mã khác. Phương pháp này, có thể xác định giá trị m nếu chưa biết.

### *1.2.5. Thám mã hệ mã dòng xây dựng trên LFSR.*

Ta nhớ lại rằng, bản mã là tổng theo modulo 2 của bản rõ và dòng khoá, tức $y_i = x_i + z_i \bmod 2$. Dòng khóa được tạo từ $(z_1,z_2,\ldots,z_m)$ theo quạn hệ đệ quy tuyến tính:

$$z_{m+1} = \sum_{j=0}^{m-1} c_j z_{i+1} \bmod 2$$

trong đó $c_0,\ldots,c_m \in Z_2$ (và $c_0 = 1$)

Vì tất cả các phép toán này là tuyến tính nên có thể hy vọng rằng, hệ mật này có thể bị phá theo phương pháp tấn công với bản rõ đã biết như trường hợp mật mã Hill. Giả sử rằng, Oscar có một xâu bản rõ $x_1x_2\ldots x_n$ và xâu bản mã tương ứng $y_1y_2\ldots y_n$ . Sau đó anh ta tính các bít dòng khoá $z_i = x_i+y_i \bmod 2$, $1 \le i \le n$. Ta cũng giả thiết rằng Oscar cũng đã biết giá trị của m. Khi đó Oscar chỉ cần tính $c_0, \ldots, c_{m-1}$ để có thể tái tạo lại toàn bộ dòng khoá. Nói cách khác, Oscar cần phải có khả năng để xác định các giá trị của m ẩn số.

Với $i \ge 1$ bất kì ta có :

$$z_{m+1} = \sum_{j=0}^{m-1} c_j z_{i+j} \bmod 2$$

là một phương trình tuyến tính n ẩn. Nếu $n \ge 2n$ thì có m phương trình tuyến tính m ẩn có thể giải được.

Hệ m phương trình tuyến tính có thể viết dưới dạng ma trận như sau:

$$(z_{m+1}, z_{m+2},\ldots,z_{2m}) = (c_0, c_1,\ldots,c_{m-1}) \begin{bmatrix} z_1 & z_2 & \ldots & z_m \\ z_2 & z_3 & \ldots & z_{m+1} \\ \ldots & \ldots & \ldots & \ldots \\ z_m & z_{m+1} & \ldots & z_{2m-1} \end{bmatrix}$$

Nếu ma trận hệ số có nghịch đảo ( theo modulo 2 )thì ta nhận được nghiệm:

$$(c_0, c_1,\ldots,c_{m-1}) = (z_{m+1}, z_{m+2},\ldots,z_{2m}) \begin{bmatrix} z_1 & z_2 & \ldots & z_m \\ z_2 & z_3 & \ldots & z_{m+1} \\ \ldots & \ldots & \ldots & \ldots \\ z_m & z_{m+1} & \ldots & z_{2m-1} \end{bmatrix}^{-1}$$

Trên thực tế, ma trận sẽ có nghịch đảo nếu bậc của phép đệ quy được dùng để tạo dòng khoá là m.(xem bài tập). Minh hoạ điều này qua một ví dụ.

Ví dụ 1.13.

Giả sử Oscar thu được xâu bản mã

101101011110010

tương ứng với xâu bản rõ

011001111111001

Khi đó anh ta có thể tính được các bít của dòng khoá:

110100100001010

Ta cũng giả sử rằng, Oscar biết dòng khoá được tạo từ một thanh ghi dịch phản hồi (LFSR) có 5 tầng. Khi đó, anh ta sẽ giải phương trình mà trận sau ( nhận được từ 10 bít đầu tiên của dòng khoá):

$$(0,1,0,0,0)=(c_0,c_1,c_2,c_3,c_4)\begin{bmatrix}1&1&0&1&0\\1&0&1&0&0\\0&1&0&0&1\\1&0&0&1&0\\0&0&1&0&0\end{bmatrix}$$

Có thể kiểm tra được rằng:

$$\begin{bmatrix}1&1&0&1&0\\1&0&1&0&0\\0&1&0&0&1\\1&0&0&1&0\\0&0&1&0&0\end{bmatrix}^{-1}=\begin{bmatrix}0&1&0&0&1\\1&0&0&1&0\\0&0&0&0&1\\0&1&0&1&1\\1&0&1&1&0\end{bmatrix}$$

Từ đó ta có:

$$(c_0,c_1,c_2,c_3,c_4)=(0,1,0,0,0)\begin{bmatrix}0&1&0&0&1\\1&0&0&1&0\\0&0&0&0&1\\0&1&0&1&1\\1&0&1&1&0\end{bmatrix}$$

$$= (1, 0, 0, 1, 0)$$

Như vậy phép đệ quy được dùng để tạo dòng khoá là:

$$z_{i+5} = z_i + z_{i+3} \bmod 2$$

## 1.3. CÁC CHÚ GIẢI VÀ TÀI LIỆU DẪN

Nhiều tài liệu về mật mã cổ điển đã có trong các giáo trình, chẳng hạn như giáo trình của Beker và Piper [BP82] và Denning [DE82]. Xác suất đánh

giá cho 26 kí tự được lấy của Beker và Piper. Cũng vậy, việc phân tích mã Vigenère được sửa đổi theo mô tả của Beker và Piper. Rosen [Ro93] là một tài liệu tham khảo tốt về lý thuyết số. Cơ sở của Đại số tuyến tính sơ cấp có thể tìm thấy trong sách của Anton [AN91]. Cuốn " Những người mã thám " của Kahn [KA67] là một cấu chuyện hấp dẫn và phong phú về mật mã cho tới năm 1967, trong đó Kahn khẳng định rằng mật mã Vigenère thực sự không phải là phát minh của Vigenère.

Mật mã Hill lần đầu tiên được mô tả trong [HI29]. Các thông tin về mật mã dòng có thể tìm được trong sách của Rueppel [RU86].

## BÀI TẬP

1.1. Dưới đây là 4 bản mã thu được từ mã thay thế. Một bản thu được từ mã Vigenère, một từ mật mã Affine và một bản chưa xác định. Nhiệm vụ ở đây là xác định bản rõ trong mỗi trường hợp.

Hãy mô tả các bước cần thực hiện để giải mã mỗi bản mã ( bao gồm tất cả các phân tích thống kê và các tính toán cần thực hiện).

Hai bản rõ đầu lấy từ cuốn " The diary of samuenl marchbanks " của Robertson Davies, Clack Iriwin,1947; bản rõ thứ tư lấy từ " Lake wobegon days" của Garrison Keillor, Viking Penguin, 1985.

*a) Mã thay thế:*
EMGLXUDCGDNCUSWYXFPHNSFCYKDPUMLWGYICOXYFIPJCK
QPKUGKMGOLICGINCGACKFNIFACYKZSCKXECJCKFHYFXCG
0IDPKZCNKSHICGIWYGKKGKGOLDSILKGOIUFIGLEDFPWZU
GFZCCNDGYYFFUSZCNXEOJNCGYEOWEUPXEZGACGNFGLKNF
ACIGOYCKXCJUCIUZCFZCCNDGYYSFEUEKUZCSOCSZCCNC
IACZEJNCFFZEJZEGMXCYHCJUMGKUSI

*Chỉ dẫn:* F sẽ giải mã thành W.

*b) Hệ mã Vigenère*
KCCPKBGUFDPHQTYAVINRRTMVGRKDNBVFDETDGLLTXRGUD
DKBTMBPVGEGLTGCKQRACQCWDNAWCRXIZAKSTLEWRPTYC
QKYVXCHKFTPONCQQRHJVAJUWETMCMSPKQDYHJVDAHCTRL
SVSKCGCZQQDZXGSFRLFWCWSJTBHAFSIASPRJAHKJRJUMP
FFSQNRWXCVYCGAONWDDKACKAWBBIKFTIOVKCGGHJVLNHI
CWHJVLNHIQIBTKHJVNPIST

*c) Hệ mã Affine.*

KQEREJEBCPPCJCRKIEACUZBKRVPKRBCIBQCARBJCVFCUP
KRIOFKPACUZQEPBKRXPEIIEABDKPBCPFCDCAFIEABDKP
BCPFEQPKAZBKRHAIBKAPCCIBURCCDKDCCJCIDFUIXPAFF
ERBICZDFKABICBBENEFCUPJCVKABPCYDCCDPKBCOCPERK
IVKSCPICBRKIJPKABI

*d) Hệ mã chưa xác định được.*

BNVSNSIHQCEELSSKKYERISJKXUMBGYKAMQLJTYAVFBKVT
DVBPVVRJYYLAOKYMPQSCGDLFLLPROYGEFEBUUALRWXM
MASAZLGLEDFJBZAVVPXWYCGJXASCBYEHOSNMULKCEAHTQ
OKMFLEBKFXLRRFDTZXCIWBJSICBGAWDVYDHAVFJXZIBKC
GJIWEAHTTOEWTUHKRQVVRGZBXYIREMMASCSPBNLHJGBLR
FFJELHWEYLWISTFVVYFJCMHYURUFSFMGESIGRLWALSWM
NUHSIMYYITCCQPZSICEHBCCMZFEGVJYOCDEMMPGHVAAMU
ELCMOEHVLTIPSUYILVGFLMVWDVYDBTHERAYISYSGKVSUU
HYHGGCKTMBLRX

1.2. a) Có bao nhiêu ma trận khả nghịch cấp 2×2 trên $Z_{26}$ .

b) Giả sử p là số nguyên tố. Hãy chứng tỏ số các ma trận khả nghịch cấp 2×2 trên $Z_p$ là $(p^2-1)(p^2-p)$.

*Chỉ dẫn:* Vì p là số nguyên tố nên $Z_p$ là một trường. Hãy sử dụng khẳng định sau: Một ma trận là khả nghịch trên một trường là khả nghịch khi và chỉ khi các hàng của nó là các véc tơ độc lập tuyến tính ( tức không tồn tại một tổ hợp tuyến tính các hàng khác 0 mà tổng của chúng là một véc tơ toàn số 0).

c) Với p là số nguyên tố và m là một số nguyên ≥ 2. Hãy tìm công thức tính số các ma trận khả nghịch cấp m×m trên $Z_p$.

1.3. Đôi khi chọn một khoá mà phép mã và giải mã là đồng nhất rất hữu ích. Trong trường hợp mất mã Hill, ta phải tìm các ma trận K sao cho $K = K^{-1}$ ( ma trận này được gọi là ma trận đối hợp). Trên thực tế, Hill đã đề nghị sử dụng các ma trận này làm khoá trong các hệ mật của mình. Hãy xác định số các ma trận đối hợp trên $Z_{26}$ trong trường hợp m = 2.

*Chỉ dẫn:* Hãy dùng công thức trong định lý 1.3 và để ý rằng detA = ± với một ma trận đối hợp trên $Z_{26}$.

1.4. Giả sử ta đã biết rằng bản rõ " *conversation* " sẽ tạo nên bản mã " HIARRTNUYTUS " ( được mã theo hệ mã Hill nhưng chưa xác định được m). Hãy xác định ma trận mã hoá.

1.5. Hệ mã Affine - Hill là hệ mã Hill được sửa đổi như sau: Giả sử m là một số nguyên dương và $\mathcal{P} = C = (Z_{26})^m$. Trong hệ mật này, khoá K gồm các cặp (L,b), trong đó L là mọt ma trận khả nghịch cấp m×m trên $Z_{26}$ và $b \in (Z_{26})^m$. Với $x = (x_1,\ldots,x_m) \in \mathcal{P}$ và $K = (L,b) \in \mathcal{K}$, ta tính $y = e_K(x) = (y_1,\ldots,y_m)$ theo công thức y = xL + b. Bởi vậy, nếu $L = (l_{i,j})$ và $b = (b_1,\ldots,b_m)$ thì:

$$(y_1,\ldots,y_m) = (x_1,\ldots,x_m)\begin{bmatrix} l_{1,1} & l_{1,2} & . & . & . & l_{1,m} \\ l_{2,1} & l_{2,2} & . & . & . & l_{2,m} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ l_{m,1} & l_{m,2} & . & . & . & l_{m,m} \end{bmatrix} + (b_1,\ldots,b_m)$$

Giả sử Oscar đã biết bản rõ là "*adisplayedequation*" và bản mã tương ứng là " DSRMSIOPLXLJBZULLM". Oscar cũng biết m =3. Hình tính khoá và chỉ ra tất cả các tính toán cần thiết.

1.6. Sau đây là cách thám mã hệ mã Hill sử dụng phương pháp tấn công chỉ với bản mã. Giả sử ta biết m = 2. Chia các bản mã thành các khối có độ dài 2 kí tự ( các bộ đôi). Mỗi bộ đôi này là bản mã của một bộ đôi của bản rõ nhờ dùng một ma trận mã hoá chưa biết. Hãy nhặt ra các bộ đôi thường gặp nhất trong bản mã và coi rằng đó là mã của một bộ đôi thường gặp trong danh sách ở bảng 1.1 ( ví dụ TH và ST). Với mỗi giả định, hãy thực hiện phép tấn công với bản rõ đã biết cho tới khi tìm được ma trận giải mã đúng.

Sau đây là một ví dụ về bản mã để bạn giải mã theo phương pháp đã nêu:

LMQETXYEAGTXCTUIEWNCTXLZEWUAISPZYVAPEWLMGQWYA
XFTCJMSQCADAGTXLMDXNXSNPJQSYVAPRIQSMHNOCVAXFV.

1.7. Ta sẽ mô tả một trường hợp đặc biệt của mã hoán vị. Giả sử m, n là các số nguyên dương. Hãy viết bản rõ theo thành từng hàng thành một hình chữ nhật m×n. Sau đó tạo ra bản mã bằng cách lấy các cột của hình chữ nhật này. Ví dụ, nếu m = 4, n = 3 thì ta sẽ mã hoá bản rõ "*cryptography*" bằng cách xây dựng hình chữ nhật :

*cryp*
*togr*
*aphy*

Bản mã sẽ là: 'CTAROPYGHPRY'

a) Hãy mô tả cách Bob giải mã một bản mã ( với m, n đã biết)
b) Hãy giải mã bản mã sau: ( nhận được theo phương pháp đã nêu):

MYAMRARUYIQTENCTORAHROYWDSOYEOUARRGDERNOGW

1.8. Có 8 phép đệ quy tuyến tính bậc 4 khác nhau trên $Z_2$ với $c_0 = 1$. Hãy xác định những phép đệ quy nào tạo được dòng khoá có chu kỳ 15 ( với véc tơ khởi tạo khác 0).

1.9. Mục đích của bài tập này để chứng minh khẳng định ở phần 1.2.5 là : ma trận hệ số cấp m×m có nghịch đảo. Điều này tương đương với khẳng định rằng, các hàng ma trận này là các véc tơ độc lập tuyến tính trên $Z_2$.

Giả sử rằng phép đệ quy có dạng:

$$z_{m+i} = \sum_{j=0}^{m-1} c_j z_{i+j} \ mod 2$$

( $z_1,. . .,z_m$) là véc tơ khởi tạo. Với i ≥ 1 ta xác định:

$$v_i = (z_i,. . .,z_{i+m-1})$$

Chú ý rằng, ma trận hệ số có các véc tơ $v_1,. . .,v_m$ là các hàng của nó. Bởi vậy, nhiệm vụ của ta là chứng tỏ rằng m véc tơ này là độc lập tuyến tính.

Hãy chứng minh hai khẳng định sau:

a) Với i ≥ 1 bất kì:

$$v_{m+i} = \sum_{j=0}^{m-1} c_j v_{i+j} \ mod 2$$

b)Chọn h là số nguyên nhỏ nhất sao cho tồn tại một tổ hợp tuyến tính không tầm thường của các véc tơ $v_1,. . .,v_h$ có tổng là véc tơ (0, . . . , 0) theo modulo 2. Khi đó:

$$v_h = \sum_{j=0}^{h-2} c_j v_{j+1} \ mod 2$$

( Các $\alpha_j$ không đồng nhất bằng 0). Để ý rằng, h ≤ m+1 vì m+1 là véc tơ bất kỳ trong không gian tuyến tính m chiều đều phụ thuộc tuyến tính .

c) Hãy chứng tỏ rằng dòng khoá phải thảo mãn phép đệ quy:

$$z_{h-1+i} = \sum_{j=0}^{h-2} \alpha_j z_{j+i} \ mod 2$$

với bất kì i ≥ 1.

d) Ta nhận thấy rằng, nếu $h \leq m$ thì dòng khoá thảo mãn phép đệ quy tuyến tính có bậc nhỏ hơn m. Điều này mâu thuẫn. Bởi vậy $h = m + 1$ và ma trận phải là khả nghịch.

1.10. Hãy giải mã bản mã sau ( thu được từ mã khoá tự sinh ) bằng phương pháp tìm khoá vét cạn.

MALVVMAFBHBUQPTSOXALTGVWWRG

1.11. Ta sẽ mô tả một hệ mã dòng là biến thể của mã Vigenère như sau. Với một từ khoá độ dài m cho trước ( $k_1,. . .,k_m$ ), ta tạo dòng khoá theo quy tắc $z_i=k_i$ $(1 \leq i \leq m)$, $z_{i+m} = z_i+1 \bmod 26$ ( $i \geq m+1$). Nói cách khác, mỗi lần dùng từ khoá ta sẽ thay mỗi kí tự bằng kí tự đứng sau nó theo modulo 26. Ví dụ, nếu SUMMER là từ khoá thì ta dùng SUMMER để mã hoá 6 kí tự đầu.,sau đó dùng TVNNFS để mã hoá 6 kí tự tiếp theo và cú tiếp tục như vậy.

Hãy mô tả cách có thể dùng khái niệm chỉ số trùng hợp như thế nào để trước hết là xác định độ dài từ khoá và sau đó là tìm từ khoá.

Hãy kiểm tra phương pháp của bạn bằng cách bằng cách phân tích bản mã sau:

IYMYSILONRFNCQXQJEDSHBUIBCJUZBOLFQYSCHATPEQGQ
JEJNGNXZWHHGUFSUKULJQACZKKJOAAHGKEMTAFGMKVRDO
PXNEHEKZNKFSKIFRQVHHOVXINPHMRTJPYWQGJWPUUKFP
OAWPMRKKQZWLQDYAZDRMLPBJKJOBWIWPSEPVVQMBCRYVC
RUZAAOUMBCHDAGDIEMSZFZHALIGKEMJJFPCIWKRMLMPIN
AYOFIREAOLDTHITDVRMSE

Bản rõ được lấy từ "The codebreakers" của D.Kahn, 1967.

# CHƯƠNG 13

# CÁC CHỨNG MINH KHÔNG TIẾT LỘ THÔNG TIN

## 13.1.CÁC HỆ THỐNG CHỨNG MINH TƯƠNG HỖ

Một cách đơn giản, một hệ thống chứng minh không tiết lộ thông tin sẽ cho phép một đối tượng thuyết phục được một đối tượng khác tin một điều nào đó mà không để lộ một tý thông tin nào về phép chứng minh. Trước tiên ta sẽ thảo luận ý tưởng về một hệ thống chứng minh tương hỗ. Trong một hệ thống chứng minh tương hỗ có hai thành viên: teggy và Vic. Teggy là người chứng minh và Vic là người kiểm tra. Teggy biết một điều gì đó và cô ta muốn chứng minh cho Vic rằng cô ta biết điều đó.

Điều cần thiết là phải mô tả được các kiểu tính toán mà Peggy và Vic được phép thực hiện và các tác động qua lại xảy ra. Ta có thể coi các thuật toán mà Peggy và Vic thực hiện là các thuật toán xác suất. Peggy và Vic sẽ thực hiện các tính toán riêng và mỗi người đều có một bộ tạo số ngẫu nhiên riêng. Họ sẽ liên lạc với nhau qua một kênh truyền tin. Thoạt đầu cả Peggy và Vic đều có một giá trị x. mục đích của phép chứng minh tương hỗ là Peggy phảI thuyết Vic rằng x có một tính chất xác đình nào đó. Chính xác hơn x là câu trả lời có của một bái toán quyết định xác định $\prod$.

Phép chứng minh tương hỗ (là một giao thức hỏi-đáp) gồm một số vòng xác định. Trong mỗi vòng .Peggy và Vic luân phiên thực hiện các công việc sau:

1. Nhận một thông báo từ nhóm khác .
2.Thực hiện một tính toán riêng.
3. Gửi một thông báo toiư nhóm khác

Một vòng đIển hình của giao thức sẽ gồm một yêu cầu của Vic và một đáp ứng của Peggy. Tới cuối phép chứng minh ,Vic hoặc sẽ chấp nhận hoặc từ chối tuỳ thuộc vào việc liệu Peggy có đáp ứng thành công các yêu câù của Vic hay không. Ta định nghĩa giao thức là một hệ thông chứng minh tương hỗ đối với vái toán quyết định $\prod$ nếu hai tính chất sau được thoả mãn mỗi khi Vic tuân theo giao thức đó:

Tính đầy đủ

Nếu x là câu trả lời có của hai bái toán quyết định ∏ thì Vic sẽ luôn luôn chấp nhận chứng minh của Peggy.

Tính đúng đắn

Nếu x là câu trả lời không của ∏ thì xác suất để Vic chấp nhận phép chứng minh là rất nhỏ.

Ta hạn chết chỉ xét các hệ thống chứng minh tương hỗ mà các tính toán do Vic thức hiện nằm trong thoì gian đa thức song không hàn chế thời gian tính toán mà prggy thực hiên.(Peggy được coi là "toàn năng").

Ta sẽ bắt đầu bằng việc trình bày một hệ thống chứng minh tương hỗ cho bái toán đồ thị không dẳng cấu. Bái toán đẳng cẩu đồ thị được mô tả trên hình 13.1. Đây là một bái toán thú vị mà cho tới nay người ta chưa biết thuật giải nào có thời gian đa thức tuy rằng không được coi là bái toán NP đầy đủ.

Hình 13.1 . tính đẳng cấu đồ thị

*Đặc trưng của bái toán* : 2 đồ thị n đỉnh $G_1=(V_1,E_1)$ và $G_2=(V_2,E_2)$
*Câu hỏi*: liệu có một song ánh $\pi$: $V_1 \rightarrow V_2$ sao cho $\{u,v\} \in E_1$ khi và chỉ khi $\{\pi(u), \pi(v)\} \in E_2$ không ?. (nói cách khác liệu $G_1$ và $G_2$ có đẳng cấu không ?)

Sau đây sẽ trình bày một hệ thống chứng minh tương hỗ cho phép Peggy chứng tỏ với Vic rằng 2 đồ thị chỉ ra là không đẳng cấu. Để đơn giản, giả sử rằng mỗi đồ thị G1 và G2 có tập đỉnh {1..n}. Hệ thông chứng minh tương hỗ đối với tính không đẳng cấu đồ thị được mô tả trên hình 13.2.

Hình 13.2. Một hệ thống chứng minh tương hỗ đối với tính không đẳng cấu đồ thị

Đầu vào :mỗi đồ thị G1 và G2 có tập đỉnh {1,....,n}

1. Hãy lặp lại các bước sau n lần:
2. Vic chọn một số ngẫu nhiên I=1 hoặc 2 và một phép hoán vị ngẫu nhiên π của {1,...,m}.Vic sẽ tính H là ảnh của G theo hoán vị π và gửi H cho Peggy.
3. Peggy xác định giá trị j sao cho $G_j$ là đẳng cấu với H và gửi j cho Vic
4. Vic sẽ kiểm tra xem liệu i=j không .
5. Vic chấp nhận chứng minh của Peggy nếu I=j trong mỗi vòng.

Hình 13.3 các đồ thị không đẳng cấu của Peggy và yêu cầu của Vic

??????????????????????

Ví dụ nhỏ sau đây sẽ minh hoạ cho hoạt động của thuật toán

Ví dụ 13.1

Giả sử G1 = (V, $E_1$)và $G_2$=(V, $E_2$) trong đó V = (1, 2, 3, 4), $E_1$ = {12, 14, 23, 34}, E2 ={112,13,14,34}.

Gỉa sử ở một vòng nào đó của giao thức,Vic trao cho Peggy đồ thị H=(V, $E_3$) trong đó $E_3$={13, 14, 23, 24}(xem hình 13.3). Đồ thị H là đẳng cấu với $G_1$. (Một phép đẳng cấu từ H vào G1 là phéo hoán vị (1 3 4 2)). Bởi vậy Peggy sẽ trả lời "1"

Dễ dàng nhận thấy rằng, hệ thống chứng minh này thoả mãn tính đầy đủ và tính đúng đắn. Nếu $G_1$ không đẳng cấu với $G_2$ thì j sẽ bằng i ở mỗi vòng và Vic sẽ chấp nhận với xác suất 1. Bởi vậy giao thức là đầy đủ.

Mặt khác, giả sử rằng $G_1$ đẳng cấu với $G_2$. Khi đó một đồ thị yêu cầu bất kỳ H được Vic đưa ra đẳng cấu với cả $G_1$ và $G_2$. Peggy sẽ không có cách nào để xác định xem H là phiên bản đẳng cấu nào của $G_1$ hay $G_2$ nên Peggy không còn cách nào khác hơn là phải trả lời bằng cách giả định j=1 hoặc 2. Cách duy nhất để Vic chấp nhận là xem Peggy có khả năng phán đoán tất cả n phép chọn i do Vic thực hiện hay không. Xác suất Peggy thực hiện điều này là $2^n$. Bởi vậygiao thức là đúng đắn.

Chú ý rằng các tính toán của Vic đều trong thời gian đa thức. Ta không thể nói tý gì về thời gian tính toán củ Peggy vì bái toán đồ thị đẳng cấu chưa có một thuật giải nào theo thờigian đa thức. Tuy nhiên hãy nhớ lại rằng ta đã cho Peggy có năng lực tính toán không hạn chế và bởi vậy đều này được chấp nhận theo "các quy tắc của trò chơi".

---

## 13.2. CÁC CHỨNG MINH KHÔNG TIẾT LỘ THÔNG TIN HOÀN THIỆN.

Mặc dù các hệ thống chứng minh tương hỗ khã hay ho nhưng kiểu chứng minh thú vị nhất lại là kiêu chứng minh không để lộ thông tin. Vấn đề là Peggy phảI thuyết phục Vic rằng x có một tính chất xác định nào đó, nhưng vào lúc kết thúc giao thức Vic vẫn không có chút ý niệm nào về cách chứng minh (cho chính anh ta ) rằng có tính chất đó. Đây là một khái niệm rất khó định nghĩa hình thức ,bởi vậy ta sẽ đưa ra trước khi định nghĩa nó. Trên hình 13.4 mô tả một phép chứng minh tương hỗ không tiết lộ thông tin đối với tính đẳng cấu của đồ thị. Ví dụ nhỏ sau sẽ minh hoạ cho hoạt động của thuật toán.

Đầu vào :hai đồ thị G1 và G2 mỗi đồ thị có tập đỉnh {1...n}

1. Lặp lại các bước sau n lần
2. Peggy chọn một phứp hoán vị ngẫy nhiên π vủa {1...n} cô ta tính H là ảnh của $G_1$ theo π và gửi H cho Vic
3. Vic chọn một số nguyên ngẫu nhiên I=1 hoặc 2 và gửi nó cho Peggy
4. Peggy tính một phép hoán vị của {1....n} sao cho H là ảnh của $G_1$ theo p. Peggy sẽ gửu p cho Vic (nếu i= 1 thì Peggy sẽ xác định p=π nếu I=2 thì Peggy sẽ xác định p là hợp của δ và π trong δ là một phép hoán vị cố định nào đó sao cho ảnh của $G_2$ theo δ là $G_1$)
5. vic sẽ kiểm tra xem H có phải là ảnh của $G_1$ theo p hay không
6. vic sẽ chấp nhận chứng minh của Peggy nếu H là ảnh của $G_1$ ở mỗi một trong n vòng.

Ví dụ 13.2:

Giả sử $G_1 = (V, E_1)$ và $G_2 = (V, E_2)$, trong đó V = {1, 2, 3, 4}, $E_1$ = {12, 13, 14, 34} và $E_2$={12, 13, 23, 24}. Một phép đẳng cấu từ $G_2$ sang $G_1$ là hoán vị δ=(4 1 2 3).

Bây giờ giả sử ở trong vòng nào đó của giao thức Peggy chọn hoán vị π = (2 4 1 3). Khi đó H có tập cạnh {12, 13, 23, 24} (xem hình 13.5)

Nếu yêu cầu của Vic là i=1 thì Peggy sẽ cho Vic phép hoán vị π và Vic sẽ kiểm tra xem ảnh của $G_1$ theo π có phải là H không. Nếu yêu cầu của Vic là i=2 thì thì Peggy sẽ cho Vic phép hợp $p=\pi_0$ δ =(3 2 1 4) và Vic sẽ kiểm tra xem ảnh của G2 theo p có phải là H không.

Dễ dàng diểm tra được tính đầy đủ và tính đúng đắn của giao thức. Không khó khăn thấy rằng xác suất để Vic chấp nhận sẽ bằng 1 nếu $G_1$ đẳng cấu với $G_2$. Mặt khác nếu $G_1$ không đẳng cấu với G2 thì chỉ có một cách để Peggy lừa dối được Vic là có ta phải giả định đúng giá trị i mà Vic sẽ chọn ở

mỗi vòng và ghi một bản sao đẳng cấu (ngẫu nhiên ) của $G_1$ lên băng liên lạc. Xác suất để Peggy giả định đúng các yêu cầu của Vic là $2^n$.

??????????????????????????????

Tất cả các tính toán của Vic có thể thực hiện được trong thời gian đa thức (như một hàm của n là số các đỉnh trong $G_1$ và $G_2$). Mặc dù không cần thiết lắm nhưng ta cũng thấy rằng các tính toán của Peggy cũng có thể được thực hiện trong thời gian đa thức miễn là cô ta biết được sự tồn tạI của phép hoán vị δ là $G_1$.

Tại sao ta lại coi hệ thống chứng minh là hệ thông chứng minh không tiết lộ thông tin. Lý do là ở chỗ mặc dù Vic đã bị thuyết phục rằng $G_1$ là đẳng cấu với $G_2$ nhưng anh ta vẫn không thu thêm được tý kiến thức nào để giúp tìm được phép hoán vị δ đưa $G_2$ về $G_1$. Tất cả những đIều mà Vic thấy trong mỗi vòng của phép chứng minh là một bản sao ngẫu nhiên của các độ thị này mà không cần tới sự giúp đỡ của Peggy. Vì các đồ thị H được chọn một cách độc lập và ngẫy nhiên ở mỗi phần của phép chứng minh nên đIều này không giúp đỡ được gì vho Vic trong việc tìm một phép dẳng cấu từ $G_1$ sang $G_2$.

Ta hãy xem xét kĩ lưỡng thông tin mà Vic thu được nhờ tham gia vào hệ thông chứng minh tương hỗ. Có thể biểu thị cách nhình của Vic về phép chứng minh tương bằng một " bản sao " chứa các thông tin sau:

—

1.Các đồ thị $G_1$ và $G_2$
2. Tất cả các thông báo được Peggy và Vic gửi đi.
3. Các số ngẫu nhiên mà Vic dùng để tào các yêu cầu của mình.

Bởi vậy một bản sao T đối với phép chứng minh tương hỗ về phép đẳng cấu đồ thị sẽ có dạng sau:

$$T = ((G_1, G_2):(H_1, i_1, p_1): \ldots (H_n, i_n, p_n))$$

Điểm mấu chốt (tạo cơ sở cho định nghĩa hình thức về phép chứng minh không tiết lộ thông tin ) là Vic (hay vất kỳ người nào khác) có thể giả mạo

các bản sao (mà không cần phải tham gia vào hệ chứng minh tương hỗ) ”giống như” các bản sao thực tế. Điều này có thể thực hiện được miễn là các đồ thị $G_1$ và $G_2$ là đẳng cấu. Việc giả mạo được thực hiện theo thuật toán mô tả trên hình 13.6. thuật toán giả mạo là một thuật toán xác suất theo thời gian đã thức. Theo nhôn ngữ của phép chứng minh không tiết lộ thông tin một thuật toán giả mạo thường được gọi là một bộ mô phỏng.

Sự kiện một bộ mô phỏng có thể giả mạo các bản sao có một hệ quả rất quan trọng. Bất kỳ kết quả nào mà Vic (hay bất kì ai khác ) có thể tính từ một bản sao cũng có thể tính được từ một bản sao giả mạo. Bởi vậy ,việc tham gia vào hệ thông chứng minh sẽ không làm tăng khả năng tính toán của Vic; đặc biệt là điều này không cho phép Vic tự chứng minh được rằng $G_1$ và $G_2$ là đẳng cấu. Hơn nữa, Vic cũng không thể thuyết phục được ai khác rằng $G_1$và $G_2$ là đẳng cấu bằng cách chỉ cho họ bản soa T bởi vì không có cách nào để phân biệt một bản sao hợp lệ với một bản sao giả mạo.

Ta sẽ chính xác hoá ý tưởng về một bản sao giả mạo “giống như” một bản sao thực và đưa ra một định nghĩa chặt chẽ theo thuật ngữ về các phân bố xác suất.

*Định nghĩa 13.1*

*Giả sử ta có một chứng minh tương hỗ thời gian đa thức cho bái toán quyết định $\prod$ và một bộ mô phỏng thời gian đa thức S. Kí hiệu tập tất cả các bản sao có thể cho kết quả có x là F(x). Các bản sao giả mạo có thể được tạo bởi S là F(x). với bản sao bất kỳ $T \in \tau(x)$ ,cho bản sao giả mạo có thể được tạo bởi S là F(x). với bản sao bất kì $T \in \tau(x)$ cho $p_\tau(T)$ là xác suất để T là một bản sao được tạo từ phép chứng minh tương hỗ. Tương tự, với $T \in F(x)$, cho $p_\tau(T)$ là xác suất để T là một bản sao (giả mạo) được tạo bởi S, Giả sử rằng $\tau(x) = F(x)$ và với bất kỳ $T \in \tau(x)$ ta có $p_\tau(T) = p_F(T)$ (nói cách khác tập các bản sao thực đồng nhất với tập các bản sao giả mạo và hai phân bố xác suất là như nhau). Khi đó ta định nghĩa hệ thống chứng minh tương hỗ là hệ thông chứng minh không tiết lộ thiing tin hoàn thiện đối với Vic.*

Hình 13.6 thuật toán giả mạo cho các bản sao đối với phép đẳng cấu đồ thị

Đầu vào : hai đồ thị G1 và G2 mỗi đồ thị có tập đỉnh {1...n}
1. T=($G_1$, $G_2$)
2. For j=1 to n do
3. Chọn ngẫu nhiên $i_j$=1 hoặc 2;
4. Chọn $p_j$ là một hoán vị ngẫu nhiên của{1...n}
5. Tính $H_j$ là ảnh của G1 theo $p_j$
6. Ghép ($H_j$, $i_j$, $p_j$) vào cuối của T

Dĩ nhiên là có thể định nghĩa đặc tính không tiết lộ thông tin theo kiểu mà ta thiéc. Tuy nhiên điều quan trọng là định nghĩa phải giữ nội dung cơ bản của đặc tính này. Ta đã coi rằng một hệ thống chứng minh tương hỗ là hệ không tiết lộ thông tin cho Vic nếu tồn tại một hệ mô phỏng rạo ra các bản sao có phân bố xác suất đồng nhất với phân bố xác suất của các bản sao được tạo ra khi Vic tham gia thực sự vào giao thức. (đây là một khái niêm tương đối nhưng mạnh hơn khái niệm về các phân mốt xác suất không có khả năng phân biệt nêu trong chương 12). Ta đã biết rằng một bản sao sẽ chứa tất cả các thông tin mà Vic thu lượm được nhờ tham gia vào giao thức. Bởi vậy, quả là hợp lý khi ta xem rằng bất cứ việc gì mà Vic có thể thực hiện được sau khi tham gia vào gia thức cũng chỉ như việc mà anh ta có thể thực hiện được nếu sử dụng hệ mô phỏng để tào một bản sao giả mạo. Mặc dù ta không định nghĩa" thông tin"(hiểu biết )bằng cách tiếp cận này nhưng bất cứ đIều gì được coi là thông tin thì Vic không thu lượm được tý nào!

Bây giờ ta sẽ chứng tỏ rằng hệ thống chứng minh tương hỗ đối với tính đẳng cấu đồ thị là một hệ thống chứng minh không tiết lộ thông tin đối với Vic.

*Định lý 13.1*

*Hệ thông chứng minh tương hỗ đối với tính đẳng cấu đồ thị là một hệ thống chứng minh không tiết lộ thông tin hoàn thiện đối với Vic.*

*Chứng minh:*

Giả sử $G_1$ và $G_2$ là các đồ thị đẳng cấu có n đỉnh. Một bản sao T (thực hoặc giả mạo) sẽ gồm n bộ dạng(H, i, ρ)trong đó i=1 hoặc 2, p là một phép hoán vị của {1,...,n} và H là ảnh của $G_1$ theo hoán vị ρ. Ta goim một bộ ba như vậy là một bộ ba hợp lệ và ký hiệu nó là ???????????. Trước tiên ta sẽ tính |??????| là số các bộ ba hợp lệ. Hiển nhiên là |????| = 2×n! vì mỗi phép chọn i và p sẽ xác định một đồ thị duy nhất H.

Ở mỗi vòng cho trước j bất kỳ của thuật toán giả mạo rõ ràng là mỗi bộ ba hợp lệ (H, i, ρ)là bộ ba thứ j ở bản sao thực là gì? Trong hệ thống chứng minh tương hỗ, trước tiên Peggy dẽ chọn một phép hoán vị ngẫu nhiên π sau đó tính H là ảnh của $G_1$ theo π. Phéhoán vị p được xác định là π nếu i = 1và nó đươc xác định là hợp của hai phép hoán vị π nếu i = 2.

Giả sử giá trị vủa i được chọn ngẫu nhiên bởi Vic. Nếu i = 1 thì tất cả n! phép hoán vị ρ là đồ các suất vì trong trường hợp này ρ = π và π đã được chọn là một phép hoán vị ngẫu nhiên. Mặt khác, nếu i = 2 thì $\rho = \pi_0\delta$ ,trong đó π là ngẫu nhiên và δ là cố định. Trong trường hợp này mỗi phép hoán vị có thể đều có xác suất bằng nhau. Xét thấy, vì cả hai trường hợp i = 1 và i = 2 đều vào xác suất bằng nhau và mỗi phép hoán vị ρ đồng xác suất (không phụ thuộc vào giá trị của i) và bởi vì i và p cùng xác định H nên suy ra mọi bộ ba trong R chắc chắn sẽ đồng xác suất.

Vì một bản sao gồm n bộ ngẫu nhiên độc lập ghép với nhau nên đối với mỗi bản sao có thể có T ta có:

$$p_\tau(T) = p_F(T) = \frac{1}{(2*n!)^n}$$

Trong chứng minh trên đã giả thiết Vic tuân thủ giao thức khi anh ta tham gia vào hệ thống chứng minh tương hỗ. Tình hình sẽ phức tạp hơn nhiệu nếu Vic không tuân theo giao thức. Phải chăng một phép chứng minh tương hỗ vẫn còn giữ được đặc tính không để lộ thông tin ngay cả khi Vic đi chéch khỏi giao thức?.

Trong trường  hợp phép đẳng cấu đồ thị, cách duy nhất mà Vic có thể đi chệch khỏi giao thức chọn các yêu cầu i của mình theo cách không ngẫu nhiên. về mặt trực giác có vẻ như đIều này không cung cấp cho Vic một chút "hiểu biết" nào. Tuy nhiên các bản sao được tạo bởi bộ mô phỏng sẽ không còn giống như các bản sao do Vic tạo ra nếu anh ta đi chệch khỏi giao thức. Ví dụ ,giả sử Vic chọn i = 1 trong mỗi vòng vủa phép chứng minh. Khi đó một bản sao của phép chứng minh tương hỗ sẽ có $i_j = 1$ với $1 \le j \le n$, trong

khi đó một bản sao được tào bởi bộ mô phỏng sẽ có $i_j = 1$ với $1 \leq j \leq n$, chỉ với xác suất xuất hiện bằng2.

Điều khó khăn ở đây là phải chứng tỏ rằng cho dù Vic "không trung thực" đi chệch khỏi giao thức nhưng vẫn tồn tại trong bộ mô phỏng thời gian với thời gian đa thức tạo ra các bản sao giả mạo giống như các bản sao được tạo bởi Peggy và Vic (không trung thực) trong phép chứng minh tương hỗ. Cũng như ở trên, câu "giống như" được hình thức hoá bằng cách nói rằng hai phân bố xacs suất này là đồng nhất.

Sau đây là một định nghĩa hình thức hơn nữa.

*Định nghĩa13.2*

*Giả sử rằng ta có một hệ thống chứng minh tương hỗ thưo thời gian đa thức cho một bái toán quyết định cho trước $\prod$. Cho V* là một thuật toán xác suất theo thời gian đa thức mà người kiểm tra (có thể không trung thực)sử dụng dể tào các yêu cầu của mình (tức là V* biểu thị cho một người kiểm tra trung thực hoặc không trung thực). Ký hiệu tập tất cả các bản sao có thể (được tào ra do kết quả của phép chứng minh tương hỗ mà Peggy và V* thực hiện với câu trả lời có x của $\prod$) là ?????(V*,x). giả sử rằng với mỗi V* như vậy tồn tại một thuật toán xác suất theo thời gian đa thức S*=S*(V*) (bộ mô phỏng) tạo ra một bản sao giả mạo. ký hiệu tập các bản sao giả mạo có thể bằng ???(V*,x). Với một bản sao bất kỳ $T \in$ ???? (V*,x) cho ???(T) là xác suất để T là một bản sao dó V* tạo ra ki tham gia vào phép chứng minh tương hỗ. Tương tự ,với $T \in F(x)$, cho ????(T) là xác suất để T là một bản sao (giả mạo)được tạo bởi S*. Giả sử rằng T(V*,x)và với bất kỳ kỳ $T \in$ ??????(V*,x), giả sử rằng $p_{Fv}(T)$ =?????(T). khi đó hệ thống chứng minh tương hỗ được gọi là một hệ thống chứng minh không tiết lộ thông tin hoàn thiện(không điều kiện).*

Trong hợp đặc biệt khi $V^*$ giống như Vic (khi Vic là trung thực) thì định nghĩa trên giống như định nghĩa 13.1.

Hình 13.7 thuật toán giả mạo cho V* đối với các bản sao cho tnsh đẳng cấu đồ thị

Đầu vào: hai đồ thị đẳng cấu G1 và G2 ,mỗi đồ thị có tập đỉnh {1...n}

1. T = ($G_1$, $G_2$)
2. For j = 1 to n do
3. Xác định tàng thái cũ bằng trạng thái (V*)
4. Repeat
5. Chọn ngẫu $i_j$=1 hoặc 2
6. Chọn $p_j$ là phép hoán vị ngẫu nhiên của {1...n}
7. Tính $H_j$ là ảnh của $G_i$ theo $\rho_j$
8. Gọi V* với đầu vào $H_j$ ta thu được một yêu cầu I',
9. If $i_j = I'_j$ then
   ghép ($H_j$, $i_j$, $\rho_j$) vào đuôi của T
   Else
   Thiết lập lại V* bằng cách xác định trạng thái (V*)

   = trạng thái cũ
10. Until $i_j$=$i'_j$

Để chứng minh rằng hệ thống chứng minh là không tiết lộ thông tin hoàn thiện ta cần một phép biến đổi chung để xây dựng một bộ mô phỏng S* từ V* bất kỳ. Ta sẽ tiếp tục thực hiện việc này đối với hệ thống chứng minh cho tính đẳng cấu đồ thị. Bộ mô phỏng sẽ đóng vai trò của Peggy sử dụng V* như một "chương trình con" có khả năng khởi tạo lại. Nói một cách không hình thức S* sẽ cố gắng giả định một yêu cầu $i_j$ mà V*sẽ đưa ra trong mỗi vòng j. tức là S* sẽ tạo ra một bộ ba hợp lệ ngẫu nhiên có dạng ($H_j$, $i_j$, $\rho_j$) và thực hiện thuật toán V* đẻ thấy được yêu cầu của nó dành cho vòng j. nếu giả định $i_j$ giống như yêu cầu $i'_j$(như được tạo bởi V*) thì bộ ba ($H_j$, $i_j$, $\rho_j$) sẽ được gắn vào bản sao giả mạo. nếu không thị bộ ba này sẽ bị loại bỏ, S* sẽ giả định một yêu cầu mới $i_j$ và thuật toán V* sẽ được khởi động lại sau khi thiết lập lại trạng thái của nó về tràng thái bắt đầu của vòng hiện thời . thuật ngữ "trạng thái "được hiểu là các giá trị của tất cả các biến dùng trong thuật toán.

Bây giờ ta sẽ đưa ra một mô tả chi tiết hơn về thuật toán mô phỏng S*.ở thời đIúm bát kỳ cho trước, trong khi thực hiên chương trình V* trạng thái hiện thời của V* sẽ được ký hiệu là state (V*). Một mô tả giả mã của thuật toán mô phỏng được cho ở hình 13.7

Có khả năng bộ mô phỏng sẽ không dừng lại nếu không xảy ra $i_j = i'_j$. tuy nhiên có thể chứng tỏ rằng thời gian chạy trung bình của bộ mô phỏng là thời gian đa thức và hai phân bố xác suất ????????(T)*và ???????*(T)là đồng nhất.

*Định lý 13.2*

*Hệ thống chứng minh tương hỗ cho tính đẳng cấu đồ thị là một hệ thông chứng minh không tiết lộ thông tin hoàn thiện.*

*Chứng minh:*

Trước tiên ta thấy rằng bất luận V* tạo ra các yêu cầu của nó ra sao, xác suất để giả định $i'_j$ là bằng 1/2. Như vậy trung bình S* phảI tạo được hai bộ ba để tạo được hai bộ ba ,để tạo được một bộ ba gắn voà bản sao giả mạo. Do đó thời gian chạy trung bình là thời gian đa thức theo n .

Nhiệm vụ khó khăn hơn là phảI chứng tỏ rằng hai phân bố xác suất ????????(T)*và ???????????*(T) là như nhau.ở định lý 13.1(trong đó Vic là người kiểm tra trung thực) ta đã tính được hai phân bố xác suất và thấy rằng chúng là đồng nhất. Ta cũng đã sử dụng một yếu tố là các bộ ba (H, i, ρ) được ở các vòng khác nhau của phép chứng minh là độc lập. Tuy nhiên trong bái toán này ta không có cách tính toán tường minh hai phân bố xác suất. Hơn nữa các bộ ba được tạo ở các vòng khác nhau của phép chứng minh lại không độc lập. Ví dụ yêu cầu mà $V^*$ đưa ra vòng j có thể phụ phuộc theo 1 kiểu rất phức tạp nào đó vào các yêu cầu ở các vòng trước và vào cách Peggy đáp ứng các yêu cầu đó.

Cách khắc phục các khó khăn này là phải xem xét các phân bố xác xuất trên các bản sao bộ phận có thể có trong quá trình mô phỏng hoặc chứng minh tương hỗ và sau đó tiếp tục bằng phương pháp quy nạp trên số các vòng. Với $0 \le j \le n$ ta xác định các phân bố xác xuất $p_{\tau,v,j}(T)$ và $p_{\tau,v,n}(T)$ trên tập các bản sao bộ phận $T_j$ xuất hiện ở cuối vòng j. Chú ý rằng $p_{\tau,v,j}(T) = p_{\tau,v}(T)$và $p_{\tau,v,n}(T) = p_{\tau,v}(T)$. Bởi vậy nếu có thể chứng tỏ rằng hai phân bố $p_{\tau,v,j}(T)$ và $p_{\tau,v,j}(T)$ là đồng nhất với mọi j thì ta có điều cần chứng minh .

Trường hợp j = 0 ứng với khi bắt đầu thuật toán : lúc này bản sao chỉ gồm hai đồ thị $G_1$ và $G_2$ .Bởi vậy các phân bố xác suất là đồng nhất khi j = 0 .Ta sẽ sử dụng điều kiện để bắt đầu phép quy nạp.

Trước tiên ta giả sử hai phân bố xác suất $p_{\tau,v,j-1}(T)$, và $p_{\tau,v,j-1}(T)$ trên $\tau_{j-1}$ là đồng nhất với giá trị $j \geq 1$ nào đó. Sau đó ta sẽ chứng tỏ rằng hai phân bố xác suất $p_{\tau,v,j}(T)$ và $p_{\tau,v,j}(T)$ trên $\tau_j$ là đồng nhất .

Xét điều sẽ xảy ra trong vòng j của phép chứng minh tương hỗ. Xác suất để yêu cầu của V là $i_j=1$ là một số thực p nào đó và xác suất để yêu cầu của V $i_j = 2$ là $1-p_i$. ở đây $p_j$ phụ thuộc vào trạng thái của thuật toán V khi bắt đầu vòng lặp j. ở trên đã nhận xét rằng trong phép chứng minh tương hỗ tất cả các đồ thị H có thể đều được Peggy chọn với xác suất như nhau (không phụ thuộc vào giá trị $p_j$), vì mọi phép hoán vị đều đồng khả năng đối với mỗi yêu cầu $i_j$ có thể .Bởi vậy xác suất đ ể bộ ba thứ j ở trên bản sao (H, i,p) bằng $p_i/n!$ nếu i=1 và bằng $(1-p_1)/n!$ nếu i=2.

Tiếp theo ta sẽ thực hiện phân tích tương tự cho phép mô phỏng .Trong một bước lặp cho trước bất kỳ của vòng lặp REPEAT, S sẽ chọn một đồ thị H bất kỳ với xác suất 1/n! .Xác suất để i=1 và yêu cầu của V là 1 bằng $p_1/2$ ; xác suất để i=2 và yêu cầu của V là 2 bằng $(1-p_j)/2$. Ở mỗi trạng thái này, (H, i, ρ) được coi là bộ ba thứ j của bản sao. Với xác suất bằng 1/2 sẽ không có gì được viết tiếp lên băng trong lần lặp cho trước bất kỳ của vòng lặp REPEAT .

Trước hết sẽ xét trường hợp i =1. Như đã nêu ở trên, xác suất để yêu cầu của V=1 là $p_1$. Xác suất để một bộ ba (H,i,p) được coi là bộ ba thứ j trong bản sao ((H,i,p) được viết tiếp lên bảng) trong bước lặp thứ i của vòng lặp REPEAT bằng:

$$\frac{P_1}{2^l \times n!}$$

Bởi vậy, Xác suất để (H, i, ρ) là bộ ba thứ j trong bản sao là:

$$\frac{p_1}{2 \times n!}\left(1+\frac{1}{2}+\frac{1}{4}+\ldots\right)=\frac{p_1}{n!}$$

Trường hợp i = 2 được phân tích theo cách tương tự : Xác suất để (H,i,p) được coi là bộ ba thứ j trong bản sao bằng $(1-p_1)/n!$.

Như vậy hai phân bố xác suất trên các bản sao bộ phận tại cuối vòng j là đồng nhất. Theo quy nạp, hai phân bố xác suất $p_{\tau,v,j-1}(T)$ và $p_{\tau,v,j-1}(T)$ là như nhau. Định lý được chứng minh …

Việc xem xét hệ thống chứng minh tương hỗ đối với tính không đẳng cấu đồ thị cũng rất thú vị. Không quá khó khăn để chứng minh rằng, hệ thống chứng minh này là hệ thống không tiết lộ thông tin hoàn thiện nếu Vic tuân thủ giao thức ( tức là nếu Vic chọn mỗi đồ thị yêu cầu là một phiên bản đẳng cấu ngẫu nhiên của $G_1$, trong đó i =1 hoặc i =2 được chọn ngẫu nhiên ). Hơn nữa nếu là Vic tạo mỗi đồ thị yêu cầu bằng cách lấy một phiên bản đẳng cấu của $G_1$ hoặc $G_2$ thì giao thức vẫn đảm bảo không tiết lộ thông tin ngay cả khi Vic chọn các yêu cầu của mình một cách không ngẫu nhiên. Tuy nhiên, giả sử rằng, kẻ gây rối Oscar đưa cho Vic một đồ thị H ( H là đẳng cấu với $G_1$ hoặc $G_2$) nhưng Vic không biết $G_i$ nào là đẳng cấu với H nếu Vic sử dụng H này làm một trong các đồ thị yêu cầu của mình trong các hệ thống chứng minh tương hỗ thì Peggy sẽ cho Vic một phép đẳng cấu mà trước đó anh ta không biết và không thể tính toán được cho chính mình. Trong tình huống này, về mặt trực giác hệ thống chứng minh sẽ không còn là một hệ thống tiết lộ thông tin và bản sao do hệ thống này tạo ra khó có thể giả mạo bằng bộ mô phỏng .

Có thể biến đổi phép chứng minh tính không đẳng cấu đồ thị để nó là một hệ thống không tiết lộ thông tin hoàn thiện, tuy nhiên ta sẽ không trình bày chi tiết ở đây .

Bây giờ ta sẽ trình bày một số ví dụ khác về các hệ thống không tiết lộ thông tin hoàn thiện. Một phép chứng minh không tiết lộ thông tin hoàn thiện cho các thặng dư bậc hai ( Modulo n = pq, trong đó p , q là các số nguyên tố) được cho ở hình 13.8 .

Hình 13.8. Hệ thống chứng minh tương hỗ không tiết lộ thông tin hoàn thiện cho các thặng dư bậc hai

Đầu vào: Một số nguyên dương n có phân tích n = pq không được biết, trong đó p, q là các số nguyên tố và $x \in QR(n)$.

1. Lập lại các bước sau $\log_2 n$ lần :
2. Peggy chọn một số ngẫu nhiên $v \in Z_n$ và tính
   $y=y^2 \bmod n$.
   Peggy gửi y cho Vic.
3. Vic chọn một số ngẫu nhiêni = 0 hoặc i = 1 và gửi nó cho Peggy.
4. Peggy tính
   $z = u^j v \bmod n$,
   trong đó u là căn bậc hai của x và gửi z cho Vic .
5. Vic kiểm tra xem liệu có thoả mãn :
   $z^2 \equiv x^i y (\bmod n)$.
6. Vic sẽ chấp nhận chứng minh của Peeggy nếu tính toán ở bước 5 được kiểm tra cho mỗi vòng (trong $\log_2 n$ vòng ) .

Peggy đang phải chứng tỏ x là một thặng dư bậc hai. ở mỗi vòng cô ta sẽ tạo ra một thặng dư bậc hai ngẫu nhiên y và gửi nó cho Vic. Sau đó tuỳ thuộc vào yêu cầu của Vic, Peggy hoặc sẽ đưa cho Vic một căn bậc hai của y hoặc một căn bậc hai của xy.

Rõ ràng là giao thức đầy đủ. Để chứng minh tính đúng đắn ta thấy rằng nếu x không phải là một thặng dư bậc hai thì Peeggy chỉ có thể trả lời một trong hai yêu cầu có thể vì trong trường hợp này y là một thặng dư bậc hai khi và chỉ khi xy không phải một thặng dư bậc hai. Bởi vậy Peggy sẽ bị tóm ở một vòng cho trước bất kỳ của giao thức với xác suất 1/2 và xác suất để Peggy đánh lừa được Vic trong toàn bộ n vòng chỉ bằng $2^{-\log_2 n} = 1/n$ ( lý do có $\log_2 n$ vòng là do cỡ đặc trưng của bái toán tỷ lệ với số bit trong biểu diễn nhị phân của n là $\log_2 n$ ). Bởi vậy xác suất đánh lừa của Peggy sẽ là một hàm mũ âm của cỡ đặc trưng của bái toán giống như trong phép chứng minh không tiết lộ thông tin cho tính đẳng cấu đồ thị.

Có thể chỉ ra tính không tiết lộ thông tin hoàn thiện đối với Vic theo cách tương tự như bái toán đẳng cấu đồ thị. Vic có thể tạo ra bộ ba (y,i,z) bằng cách trước tiên chọn i và z xác định:

$$y = z^2 (x^I)^{-1} \bmod n$$

Các bộ ba được tạo theo cách này có cùng phân bố xác suất các bộ ba được tạo trong giao thức với giả thiết Vic chọn các yêu cầu của mình một cách ngẫu nhiên. Tính không tiết lộ thông tin hoàn thiện (với v tuỳ ý) có thể được chứng minh theo phương pháp tương tự như đối với bái toán đẳng cấu đồ thị. Nó đòi hỏi phải xây dựng một bộ mô phỏng s để giả định các yêu cầu của v và chỉ giữ lại các bộ ba ứng với các giả định đúng.

Để minh hoạ thêm cho vấn đề này ta sẽ đưa ra một ví dụ nữa về phép chứng minh không tiết lộ thông tin hoàn thiện, đây là một phép chứng minh cho một bái toán quyết định có liên quan đến bái toán logarit rời rạc. Bái toán này được gọi là bái toán thành viên của nhóm con ( được mô tả ở hình 13.9 ). Dĩ nhiên là số nguyên k ( nếu nó tồn tại ) chính là logarit rời rạc của $\beta$

*Hình 13.9*. Thành viên của nhóm con.

*Đặc trưng của bái toán* **:** Hai số nguyên dương n và l, và hai phần tử phân biệt $\alpha, \beta \in Z_n$ trong đó $\alpha$ có cấp l trong $Z_n$.
Vấn đề : phải chăng $\beta = \alpha^k$ đối với một số nguyên tố k nào đó sao cho $0 \le k \le n-1$ ?(nói một cách khác là phải chăng $\beta$ là một thành viên của nhóm $Z_n$ được tạo bởi $\alpha$ ?)

Hình 13.10 Mô tả một phép chứng minh không tiết lộ thông tin hoàn thiện cho bái toán thành viên nhóm con. Việc phân tích giao thức nỳ tương tự như các giao thức mà ta đã xem xét ; các chi tiết được giành cho bạn đọc xem xét.

Hình 13.10. Hệ thống chứng minh tương hỗ không tiết lộ thông tin hoàn thiện cho thành viên của nhóm con.

Đầu vào:Một số nguyên dương n và hai phần tử phân biệt $\alpha,\beta \in Z_n$ trong đó cấp của $\alpha$ được ký hiệu bằng l và được công khai .

1. Lập lại các bước sau $\log_2 n$ lần :
2. Peggy chọn một số ngẫu nhiên j sao chi $0 \le j \le l - 1$ và tính $\gamma = \alpha^j \bmod n$ Peggy gửi $\gamma$ cho Vic.
3. Vic chọn một số ngẫu nhiên I = 0 hoặc i = 1 và gửi nó cho Peggy .
4. Peggy tính $h = j + ik \bmod l$ trong đó $k = \log_\alpha \beta$ và gửi cho Vic .
5. Vic kiểm tra xem liệu có thoả mãn đồng dư thức sau không : $\alpha^h \equiv \beta^i \gamma \pmod n$.
6. Vic sẽ chấp nhận chứng minh của Peeggy nếu tính toán ở bước 5 được kiểm tra cho mỗi vòng trong $\log_2 n$ vòng .

## 13.3 CÁC CAM KẾT BÍT

Hệ thống chứng minh không tiết lộ thông tin đối với bái toán đẳng cấu đồ thị là một hệ thống thú vị, tuy nhiên sẽ là hữu ích hơn nếu có các hệ thống chứng minh không tiết lộ thông tin cho các bái toán được coi là NP đầy đủ. Về mặt lý thuyết, không tồn tại các phép chứng minh không tiết lộ thông tin hoàn thiện cho các bái toán NP đầy đủ. Tuy nhiên ta có thể mô tả các hệ thống chứng minh có dạng không tiết lộ thông tin về mặt tính toán. Các hệ thống chứng minh thực tế sẽ được mô tả ở phần sau ; trong phần này ta sẽ mô tả kỹ thuật cam kết bít là một công cụ quan trọng được dùng trong hệ thống chứng minh .

Giả sử Peggy viết một thông báo lên một mẩu giấy rồi đặt nó vào một két sắt mà cô ta biết mã số. Sau đó Peggy trao két sắt cho Vic. Mặc dù Vic không biết thông báo là gì cho tới khi két được mở nhưng ta sẽ coi rằng Peggy đã bị ràng buộc với thông báo của mình vì cô ta không thể thay đổi nó. Hơn nữa, Vic không thể biết thông báo là gì ( giả sử Vic không biết mã số của két ).

Trừ phi Peggy mở két cho anh ta. ( Hãy nhớ lạI là ta đã dùng lập luận tương tự ở chương 4 để mô tả ý tưởng về một hệ mật công khai, tuy nhiên trong trường hợp đó Vic là người có thể mở két bởi vì anh ta là người nhận thông báo ).

Giả sử thông báo là một bít = 0 và Peggy sẽ mã hoá b theo cách nào đó. Dạng đã mã hoá của b đôI khi được gọi blob và phương pháp mã hoá được gọi là một sơ đồ cam kết bít. Nói chung , một sơ đồ cam kết bít là một hàm f: {0,1} x X $\rightarrow$ Y, trong đó X và Y là các tập hữu hạn. Một phép mã hoá của b là giá trị bất kỳ f(b,x), x$\in$X. Ta có thể định nghĩa một cách phi hình thức hai tính chất mà một sơ đồ cam kết phải thoả mãn .

Tính chất giấu kín:

Với một bít b = 0 hoặc 1, Vic không thể xác định được giá trị của b từ blob f(b,x).

Tính ràng buộc :

Sau đó Peggy có thể mở được blob bằng cách tiết lộ giá trị x dùng mã hoá b để thuyết phục Vic rằng b là giá trị đã mã. Peggy không thể mở một blob bởi cả hai giá trị 0 và 1.

Nếu Peggy muốm cam kết ( ràng buộc) một xâu bit bất kỳ thì một cách đơn giản là cô ta phảI ràng buộc từng bit một cách độc lập .

Một phương pháp để thực hiện cam kết bit là sử dụng hệ mật xác suất Goldwasser - micali mô tả ở phần 12.4 hãy nhớ lại rằng trong hệ mật này n = pq trong đó p, q là các số nguyên tố và m $\in$ ???QR(n). Các số nguyên m và n là công khai và chỉ có Peggy biết phân tích n = pq trong sơ đồ cam kết bit ta có X = Y = $Z_n^*$ và :

$$f(b,x)=m^b x^2 \bmod n$$

Peggy sẽ mã hoá giá trị b bằng cách chọn một số ngẫu nhiên x và tính y=f(b,x) ; giá trị y chính là blob .

Sau đó khi peggy muốn mở y, cô ta sẽ tiết lộ các giá trị b và x. Khi đó Vic có thể kiểm tra thấy rằng :

$$y \equiv m^b x^2 \bmod n$$

Ta xem xét tính giấu kín và tính ràng buộc. Một blob là một phép mã hoá của 0 hoặc 1, và sẽ không để lộ thông tin về giá trị bản rõ x miễn là bái toán các thặng dư bậc hai là không có khả năng giảI ( ta đã thảo luận kỹ đIều này chương 12 ). Bởi vậy sơ đồ có tính giấu kín .

Liệu sơ đồ có tính ràng buộc không ? Nếu ta giả sử là không thì

$$m\, x_1^2 \equiv x_2^2 (\bmod n)$$

Với các giá trị $x_1$, $x_2$ nào đó thuộc Zn. Tuy nhiên

$m \equiv (x_2 x_1^{-1})^2 \bmod n$

điều này mâu thuẫn bởi vì m ∈??????QR(n)

Các sơ đồ ràng buộc bit sẽ được dùng để xây dựng các phép chứng minh không tiết lộ thông tin. Tuy nhiên chúng còn có một ứng dụng tuyết vời khác vào một bái toán tung đồng xu qua đIện thoại. Giả sử Alice và Bob muốn đưa ra một quyết định nào đó dựa trên phép tung đồng xu ngẫu nhiên nhưng họ không ở cùng một địa đIểm .ĐIều này có nghĩa là không thể thực hiện được công việc một người tung đồng xu thực còn người kia kiểm tra phép thử này. Sơ đồ ràng buộc bit sẽ cho một phương pháp thoát khỏi tình trạng bế tắc này. Một trong hai người ( chẳng hạn Alice ) sẽ chọn một bit ngẫu nhiên b và tính blob y .Cô ta sẽ trao y cho Bob. Bây giờ Bob sẽ giả định giá trị của b và rồi Alice sẽ mở blob để tiết lộ b. ở đây, tính chất giấu kín có nghĩa là Bob không có khả năng tính b theo y đã cho, và tính chất ràng buộc có nghĩa là Alice không thể thay đổi được lựa chọn của mình sau khi Bob tiết lộ giả định của anh ta .

Sau đây là một ví dụ khác về sơ đồ ràng buộc bit dựa trên bái toán logarithm rời rạc. Từ phần 5.1.2 ta đã có : Nếu $p \equiv 3 \pmod 4$ là một số nguyên tố sao cho bái toán logarithm trong $Z_p$ không giảI được thì bit bậc thấp nhất thứ hai của một logarit rời rạc là an toàn. Trên thực tế, đối với các số nguyên tố $p \equiv 3 \pmod 4$, người ta chứng minh rằng thuật toán Monte - Carlo bất kỳ cho bái toán về bit thứ hai sẽ có xác suất sai bằng $1/2 - \varepsilon$ với $\varepsilon > 0$ có thể được dùng để giảI toán logarit rời rạc trong $Z_p$. Kết quả mạnh hơn nhiều này là cơ sở cho sơ đồ ràng buộc bit .

Sơ đồ ràng buộc này sẽ có $X = \{1, \ldots, p-1\}$ và $Y = Z_p$ .Bit bậc thấp nhất thứ hai của số nguyên x ( ký hiệu là SLB (x)) được xác định như sau :

$$SLB = \begin{cases} 0 \text{ Nếu } x \equiv 0,1 \pmod 4 \\ 1 \text{ Nếu } x \equiv 2,3 \pmod 4 \end{cases}$$

sơ đồ ràng buộc bit được xác định bởi :

$$f(b,x) = \begin{cases} \alpha^x \bmod p & \text{Nếu } SLB(x) = b \\ \alpha^{p-1} \bmod p & \text{Nếu } SLB(x) \neq b \end{cases}$$

Nói cách khác bit b sẽ được mã bằng cách chọn một một phần tử ngẫu nhiên có bit cuối cùng thứ hai là b và nâng $\alpha$ lên luỹ thừa x modulo p.( Chú ý rằng $SLB(p-x) \neq SLB(x)$ vì $p \equiv 3 \pmod 4$).

Sơ đồ thoả mãn tính ràng buộc và theo các nhận xét đã nêu, nó cũng thoả mãn tính giấu kín nếu bái toán logarit rời rạc trong $Z_p$ là không giảI được .

---

## 13.4 .CÁC CHỨNG MINH KHÔNG TIẾT LỘ THÔNG TIN VỀ MẶT TÍNH TOÁN .

Trong phần này ta sẽ đưa ra một hệ thống chứng minh không tiết lộ thông tin cho bái toán quyết định NP đầy đủ là bái toán về khả năng tô màu một đồ thị bằng ba màu, bái toán này được nêu ở hình 13.11.

Hệ thống chứng minh sẽ sử dụng một đồ thị cam kết ( ràng buộc ) bit: để xác định ,ta sẽ áp dụng sơ đồ ràng buộc bit được mô tả ở 13.3 ( dựa trên mã hoá xác suất ). Giả sử Peggy biết hàm ϕ ba màu của đồ thị G và cô ta muốn thuyết phục Vic rằng có thể tô màu G bằng ba màu theo kiểu không tiết lộ thông tin .Không mất tính tổng quát, giả sử rằng G có tập đỉnh V={1 .. n}. Ký hiệu m ={E}. Hệ thống chứng minh sẽ được mô tả theo các thuật ngữ cuả sơ đồ ràng buộc f:{0,1} x X $\rightarrow$Y ( được đưa ra công khai ). Vì không thể mã hoá một màu bằng một bit nên ta thay màu 1 bằng hai bit 01, màu hai bằng 10, màu ba bằng 11.Khi đó ta sẽ mã hoá mỗi bit trong hai bit (biểu thị màu ) bằng hàm f.

Hình 13.11.khả năng tô đồ thị bằng ba mầu.

Đặc trưng của bái toán :Một đồ thị G = (V,E) có n đỉnh.
Vấn đề :Liệu có thể tô G bằng đúng 3 mầu hay không?
(Theo các thuật ngữ toán học có chăng một hàm ϕ:V(G)→{1,2,3} sao cho {u,v}∈ E thì ϕ (u)= ϕ (v)?).

Hệ thống chứng minh tương hỗ được trình bày trên hình 13.12.Một cách không hình thức ,quá trình xẩy ra như sau:ở mỗi vòng ,Peggy sẽ quy

định một mầu là một hoán vị của phép tô màu xác định $\phi$.Vic sẽ yêu cầu Peggy mở các blob ứng với các điểm cuối của một cạnh nào đó được chọn ngẫu nhiên.Peggy sẽ thực hiện các điều đó và rồi Víc sẽ kiểm tra xem các quy định có tuân thủ theo dòng đòi hỏi không.Chú ý rằng mọi tính toán của Víc là theo thời gian đa thức và tính toán của Peggy cũng vậy ,miễn là cô ta biết được sự tồn tại của một phép tô 3 mầu $\phi$.

Sau đây là một ví dụ nhỏ để minh hoạ:

Ví dụ 13.3

Giả sử G là một đồ thị (V,E) trong đó :

$$V = \{1, 2, 3, 4, 5\}$$

và

$$E = \{12, 14, 15, 23, 34, 45\}.$$

Giả sử Peggy biết phép tô 3 mầu ? trong đó $\phi(1)=1$, $\phi(2)=\phi(4)=2$, và $\phi(3)=\phi(5)=3$.Ta cũng giả sử rằng các tham số của sơ đồ ràng buộc bít là n=321389 và m=156897 ,bởi vậy $f(b,x)=m^b x^2 \bmod n$,trong đó b=0,1 và $x \in Z_n^*$ .

Giả sử Peggy chọn phép hoán vị $\Pi$ =(1, 3, 5) ở một vòng nào đó cho phép chứng minh. Khi đó cô ta tính :

$C_1 = 1$
$C_2 = 3$
$C_3 = 2$
$C_4 = 3$
$C_5 = 2$

và sẽ mã hoá phép tô mầu này ở dạnh nhị phân bằng một bộ 10:

0 1 1 1 1 0 1 1 1 0

sau đó tính các ràng buộc cho 10 bít này .Giả sử cô làm như sau:

| b | x | F(b,x) |
|---|---|---|
| 0 | 147658 | 176593 |
| 1 | 318856 | 205585 |
| 1 | 14497 | 189102 |
| 1 | 285764 | 294039 |
| 1 | 128589 | 230968 |
| 0 | 228569 | 77477 |

| | | |
|---|---|---|
| 1 | 53369 | 305090 |
| 1 | 194634 | 276484 |
| 1 | 202445 | 292707 |
| 0 | 177561 | 290599 |

Say đó Peggy trao cho Vic 10 giá trị f(b,x) đã tính ở trên

Tiếp theo ,giả sử rằng Víc chọn cạnh 34 làm yêu cầu của mình.Sau đó Peggy sẽ mở 4 blob :2 lob ứng với đình 3 ,2 lob ứng với đỉnh 4.Như vậy Peggy sẽ trao cho Víc các cặp được sắp sau:

(b,x) = (1,128089), (0, 228569), (1, 53369), (1, 194634)

Víc sẽ kiểm tratrước hết xem 2 mấu có khác nhau không :10 là mã hoá của mầu 2 và 11 là mã hoá của mầu 3 .Như vậy diều vừa kiểm tra là được thỏ mãn. Tiếp theo, Víc sẽ kiểm tra thấy rằng 4 cam kết là hợp lệ.Đây là điều phải chứng minh.

Cũng như trong các hệ thống chứng minh đã được nghiên cứu ở trên Víc sẽ chấp nhận một phép chứng minh hợp lệ với xác suất =1 ,bởi vậy ta có được tính đầy đủ .Xác suất để Víc sẽ chấp nhận bằng bao nhiêunếu G không thể tô bằng 3 mầu ? Trong trường hợp này ,đối với phếp tô mầu bất kì phải có ít nhâts một cạnh ij để i và j có cùng mầu .Cơ hội để Víc chọn một cạnh như vậy it nhất là 1/m.Xác xuất để Peggy đánh lừa được Víc trong toàn bộ $m^2$ vòng nhiều nhất là

$$(1- 1/m )^n$$

vì $(1- 1/m )^m \rightarrow e^{-1}$ khi $m\rightarrow\infty$ nên ta thấy rằng $(1- 1/m )^n \rightarrow e^{-m}$ và giá trị này tiến tới 0 theo hàm mũ như là hàm của m | E | .Bởi vậy ta cũng có được tính đúng đắn.

Trở lại xem xét khía cạnh không tiết lộ thông tin của hệ thống chứng minh. Tất cả những cái mà Víc thấy trong mỗi vòng đã cho của giao thức là một phép tô 3 mầu đã mã của G, cùng với hai mầu phân biệt của các đỉnh trên một cạnh cụ thể như đã được Peggy cam kết trước đó. Vì các mầu được hoán vị ở mỗi vòng nên dường như là Víc không thể kết hợp các thông tin từ các vòng khác nhau để xây dựng phép tô 3 mầu .

Hệ thống chứng minh này không phải là một hệ thống chứng minh không tiết lộ thông tin hoàn thiện mà chỉ là một dạng yếu hơn của nó và được gọi là một hệ thống chứng minh không tiết lộ thông tin về mặt tính toán .Tính

không tiết lộ thông về mặt tính toán được định nghĩa giống như tính không tiế lộ thông tin hoàn thiện ngoại trừ một điểm là các phân bố xác suất tương ứng của các bản sao chỉ đòi hỏi không phân phân biệt theo đa thức (theo cách hiểu ở chương 12) chứ không nhất thiết phải là đồng nhất .

Hình 12.13.một hệ thống chứng minh tương hỗ không tiết lộ thông tin về mặt tính toán cho bái toán xét khả năng tô đồ thị bằng 3 mầu .

Đầu vào : Một đồ thị G = (E,V) trên tập đỉnh {1,. . .,n}

1. lặp lại các bước sau $m^2$ lần .
2. Cho ϕ là một đồ thị tô 3 mầu của G .Peggy sẽ chọn một hoán vị ngẫu nhiên Π của {1,2,3}.Với 1≤i≤n,cô ta xác định

   $C_i = \Pi(\phi(i))$

   Và viết $c_i$ như một xâu bít có độ dái hai:

   $C_i=c_{i1}c_{i2}$

   Sau đó ,với i≤1≤n cô ta chọn 2 phần tử ngẫu nhiên $r_{i1,}r_{i2}$?thuộc X và tính $r_{ij=}f(c_{ij,}r_{ij})$,j=1,2rồi gửi danh sách cho Víc

   $(r_{11} ,r_{12} ,. . .,r_{n1} ,r_{n2})$
3. Víc chọn một cách ngẫu nhiên {u,v} ∈ E và gửi nó cho Peggy.
4. Peggy gửi $(c_{u1},c_{u2},r_{u1},r_{u2})$ và $(c_{v1},c_{v2},r_{v1}.r_{v2})$ cho Víc.
5. Víc kiểm tra xem có thoả mãn các bất đẳng thức ,và đẳng thức sau không?

   $(c_{u1},c_{u2})\# (c_{v1},c_{v2})$

   $(c_{u1},c_{u2})\# (0,0)$

   $(c_{v1},c_{v2})\#(0,0)$

   $R_{u,j}=f(c_{uj},r_{uj}),j=1,2$

   $R_{v,j}=f(c_{vj},r_{vj}),j=1,2$
6. Víc sẽ chấp nhận phép chứng minh của Peggy nếu toán ở bước 5 được kiểm tra ở mỗi một trong $m^2$ vòng .

Chúng ta bắt đầu bằng việc chỉ ra cách các bản sao có thể được giả mạo như thế nào. Sau đây sẽ đưa ra một thuật toán trực tiếp giả mạo bản sao (các bản sao này không thể phân biệt được với các bản sao được tạo bởi Vic trung thực). Nếu Vic không tuân theo giao thức thì có thể xây dựng một bộ mô phỏng dùng thuật toán $V^{\times}$ như một chương trình con có thể gọi lại được để xây dựng các bản sao giả mạo. Cả hai thuật toán giả mạo này đều theo dạng các thuật toán tương đối cho hệ thống chứng minh tính đẳng cấu đồ thị.

Ở đây ta chỉ xem xét trường hợp khi Vic tuân thủ giao thức. Một bản sao T của phép chứng minh tương hỗ về tính khả tô đồ thị (bằng ba màu) sẽ có dạng:

$$(G:A_1,.. .. ..,A_{m2})$$

trong đó $A_j$ chứa 2m blob đã được tính bởi Peggy, cạnh u v được Vic chọn, các màu được Peggy gán cho các đỉnh u và v ở vòng j, và 4 số ngẫu nhiên được Peggy dùng để mã hoá các màu của hai đỉnh này. Một bản sao được giả mạo bằng thuật toán giả mạo được mô tả trên hình 13.13.

Hình 13.13. Thuật toán giả mạo các bản sao về tính khả tô đồ thị bằng ba màu.

Đầu vào: Một đồ thị G=(V,E) có tập đỉnh V={1,.. .. ,n}

1. T=(G)
2. For j=1 to $m^2$ do
3. Chọn ngẫu nhiên một cạnh $\{u, v\} \in E$
4. Chọn ngẩu nhiên các màu khác nhau $d = d_1d_2$ và $e = e_1e_2$ trong đó $d_1, d_2, e_1, e_2, \in \{0, 1\}$
5. Chọn $r_{i,j}$ là một phần tử ngẩu nhiên của X, với $1 \le i \le n$, j = 1,2
6. Với $1 \le i \le n$, j = 1,2, hãy xác định

$$R_{i,j} = \begin{cases} f(1,r_{i,j}) & \text{nếu } i \neq u,v \\ f(d_j,r_{i,j}) & \text{nếu } i = u \\ f(e_j,r_{i,j}) & \text{nếu } i = v \end{cases}$$

7. Ghép ($R_{1,1}. . . .,R_{n,2}$, u, v, $d_1$, $d_2$, $r_{d,1}$, $r_{d,2}$, $e_1$, $e_2$, $r_{e,1}$, $r_{e,2}$) vào đầu cuối

Phép chứng minh tính không tiết lộ thông tin (về mặt tính toán) đối với Vic đòi hỏi chứng tỏ rằng hia phân bố xác suất trên các bản sao (được tạo bởi Vic khi tham gia vào giao thức và được tạo bởi bộ mô phỏng ) là không thể phân biệt. ặ đây ta bỏ qua việc đó và chỉ đưa ra vái nhận xét. Cần chú ý rằng hai phân bố xác suất không đồng nhất. Sở dĩ như vậy trên thực tế tất cả các $R_{i,,j}$ trên một bản sao giả mạo là các *blob* mã hoá cho 1 ;trong khi đó các $_{Ri,\,j}$ trên một bản sao thực hiện là các *blob* mã hoá cho các số 1 và 0 với xác suấtgần bằng nhau. Tuy nhiên có thể chỉ ra rằng, không thể phân biệt được hai phân bố xác suất này trong thuời gian đa thức, nếu sơ đồ cam kết bítử dụng là an toàn. Chính xác hơn, điều đó có nghĩa lầ phân bố xác xuất trên các

*blob* mã hoá các màu c là không thể phân biệt với phân bố xác suất trên *blob* mã hoá cho các màu d nếu c ≠ d.

Bạn đọc đã làm quen với lý thuyết NP- đây đủ sẽ nhận thấy rằng nếu có một phép chứng minh không tiết lộ thông tin cho trước một bái tóan NP-đầy đủ nào đó thì ta có thể thu một phép chứng minh không tiết lộ thông tin cho một bái toán NP-đầy đủ bất kỳ khá. Điều này có thể được thức hiên bằng cách áp dụng phép biến đổi đa thức một bái toán NP-đầy đủ cho trước vào một bái toán tô đồ thị bằng ba màu.

---

## 13.5. CÁC LUẬN CỨ KHÔNG TIẾT LỘ THÔNG TIN

Ta sẽ nhắc lại các tính chất cơ bản của phép chứng minh không tiết lộ thông tin về mặt tính toán cho bái toán về tính khả tô đồ thị bằng ba màu nêu ở phần trên. Ở đây không cần giả thiết nào để chứng minh cho tính đầy đủ và tính đúng đắn của giao thức mà chỉ cần một giả thiết về mặt tính toán để chứng minh tính không tiết lộ thông tin, đó là sơ đò cam kết bit phải là một sơ đồ an toàn. Nhận thấy rằng, nếu Peggy và Vic tham gia trong giao thức thì Vic sau đó có thể cố gắng phá sơ đồ ràng buộc bít được dùng (ví dụ, nêu sơ đồ được xây dựng trên một sơ đồ thặng dư bậc hai thì Vic sẽ cố gắng thức hiên phân tích n). Nếu Vic có thể phá được sơ đồ ràng buộc bít thì anh có thể giải mã được các *blob* (được Peggy dùng trong giao thức) và rút ra phép toán tô ba màu.

Phân tích này sẽ phụ thuộc vào các tính chất của các blob dùng trong giao thức. Mặc dù tính ràng buộc của các *blob* là không điều kiện song tính dấu kín lại đưa trên giả thiết về mặt tính toán.

Một phương án khá thú vị là dùng các *blob* trong đó tính che dấu là không điều kiện nhưng tính ràng buộc lại đòi hỏi một giả thiết về mặt tính toán. Điều này dẫn tới một giao thức gọi là luận cứ không tiếtlộ thông tin hơi khác với phép chứng minh không tiết lộ thông tin. Bạn đọc nhớ lại rằng, cho tới nay ta vẫn giả sử rằng Peggy có đầy đủ sức mạnh, còn trong luận cứ không tiết lộ thông tin, ta sẽ coi rằng các tính toán của Peggy được thực hiện theo thời gian đa thức (trên thực tế giả thiết này không gây ra một chút khó khăn nào vì các tính toán của Peggy là theo thời gian đa thức nếu cô ta biết phép tô màu của G).

Ta sẽ mô tả hai sơ đồ ràng buộc bit thuộc loại này và sau đó đánh giá các kiểu sử dụng chúng trong giao thức tô đồ thị bằng ba màu.

Sơ đồ đầu tiên được xây dựng trên bái toán các thặng dư bậc hai. Giả sử n = pq, trong đó p và q là các số nguyên tố và cho m $\in$QR(n) (chú ý rằng trong sơ đồ trước m là một thặng dư giả bậc hai). Trong sơ dồ nàyPeggy không nhất thiết phải biết phân tích của n và căn bậc hai của m. Bởi vậy Vic hoặc phải xây dựng được các giá trị này hoặc chúng phải được thu nhận từ một người thứ ba (tin cậy được).

Cho X= $Z_n^*$ và Y= QR(n) và định nghĩa

$$f(b ,n) =m^b x^2 \bmod n$$

Cũng như trước đây, Peggy sẽ mã hoá giá trị b bằng cách chọn một giá trị ngẫu nhiên x và tính blob y = f(b,x). Trong sơ đồ này, tất cả các blob đều là các thặng dư bậc hai. Hơn nữa một giá trị bất kỳ y$\in$ QR(n) có thể là bản mã hoá của 0 hay của 1. Giả sử y= $x^2$ mod n và m= $k^2$ mod n. Khi đó:
y= f(0,x) = f(1,x,$k^{-1}$ mod n)

Điều đó có nghĩa là sơ đồ này đạt được tính dấu kín không điều kiện. Vậy điều kiện gì sẽ xảy ra đối với tính ràng buộc ? Peggy có thể mở một blob bất kỳ cho trước thành 0 hoặc 1 khi và chỉ khi cô ta có thể tính được k (là một căn bậc hai của m). Như vậy, để cho sơ đồ này là ràng buộc (về mặt tính toán), cần phải giả thiết rằng Peggy không có khả năng tính căn bậc hai của m. (Nếu Peggy có đầy đủ sức mạnhthì dĩ nhiên cô ta có thể làm được điều đó. Đó là lý do phải giả thiết Peggy có khả năng tính toán hạn chế).

Để làm ví dụ cho một sơ đồ cam kết bit thứ hai thuộc loại này, xét một sơ đồ xây dựng trên bái toán logarithm rời rạc. Cho p là một số nguyên tố sao cho bái toán logarithm rời rạc trong $Z_p^*$ là một bái toán bất khả giải, cho $\alpha$ là một phần tử nguyên thuỷ của $Z_p^*$ và cho $\beta \in Z_p^*$ . Giá trị $\beta$ phải được chọn bởi Vic hoặc một người thứ ba tin cậy (chứ không phải bởi Peggy). Sơ đồ này sẽ có X= {0,.....,p-1}, Y= $Z_p^*$ và f được xác định bằng:

$$f(b,x) = \beta^b \alpha^x \bmod p$$

Không khó khăn lắm có thể thấy rằng sơ đồ này có tính dấu kín không điều kiện, và nó có tính dàng buộc khi và chỉ khi Peggy không có khả năng tính được logarit rời rạc $\log_\alpha \beta$.

Bây giờ giả sử ta dùng một trong hai sơ đồ cam kết bit trên trong giao thức về tính khả thi đồ thị bằng ba màu. Dễ dàng thấy rằng, giao thức vẫn giữ được tính đầy đủ. Tuy nhiên điều kiện đúng đắn ở đây sẽ phụ thuộc vào mặt giả thiết về mặt tính toán: giao thức là đúng đắn khi và chỉ khi sơ đồ dàng buộc bit thoả mãn tính ràng buộc. Điều gì sẽ xảy ra đối với khía cạnh không tiết lộ thông tin của giao thức? Bởi vì sơ đồ cam kết bit đảm bảo tính dấu kín không điều kiện nên giao thức này sẽ trở thành một giao thức không tiết lộ thông tin hoàn thiện chứ không chỉ là một giao thức không tiết lộ thông tin về mặt tính toán nữa. Như vật ta đã có một luận cứ không tiết lộ thông tin hoàn thiện.

Bảng 13.1. So sánh các tính chất của phép chứng minh và các luận cứ

| Tính chất | Chứng minh không tiết lộ thông tin | Luận cứ không tiết lộ thông tin |
|---|---|---|
| Đầy đủ | Không điều kiện | Không điều kiện |
| Đúng đắn | Không điều kiện | Về mặt tính toán |
| Không tiết lộ thông tin | Về mặt tính toán | Hoàn thiện |
| Giấu kín | Không điều kiện | Về mặt tính toán |
| Ràng buộc | Về mặt tính toán | Không điều kiện |

Tuỳ theo áp dụng cụ thể mà người ta có thể thích dùng một luận cứ hơn là dùng một phép nhứng minh. Và khi nào thì ta phải đưa ra một giả thiết về mặt tính toán cho Peggy hay cho Vic? Một so sánh tóm lược về các tính chất của các phép chứng minhvà các luận cứ được nêu ở bảng 13.1. ở cột "chứng minh không tiết lộ thông tin ",các giả thiết về mặt tính toán có liên quan tới năng lực tính toán của Peggy. Trong cột "Luận cứ không tiết lộ thông tin", các giả thiết về mặt tính toán có liên quan tới năng lực tính toán của Vic.

---

## 13.6. CÁC CHÚ GIẢI VÀ TÁI LIỆU HƯỚNG DẪN

Phần lớn các kiến thức trong chương này đều dựa theo tái liệu của Brassard, Chaum và Crépeau [BBC 88] và của Goldreich, Micali và Wigderson [GMW 91]. Các sơ đồ cam kết (ràng buộc) bít và các thoả luận về sự khác nhau giữa các phép chứng minh và các luận cứ có thể tìm thấy trong [BBC 88](tuy nhiên cần để ý rằng thuật ngữ "luận cứ " lần đầu tiên được sử dụng trong [BC 90]. Các chứng minh không tiết lộ thông tin cho tính đẳng cấu đồ thị , tính không đẳng cấu đồ thị và tính khả tô đồ thị ba màu có thể tìm

được trong [GMW 91]. Một bái báo khác có liên quan là bái báo của Goldwasser, Micali và Rackoff [GMR 89], trong bái báo này các hệ thống chứng minh tương hỗlần đầu tiên được định nghĩa một cách hình thức. Chứng minh không tiết lộ thông tin cho bái toán các thặng dư bậc hai được lấy từ bái báo này.

Ý tưởng tung đồng tiền bằng điện thoại là của Blum [B1 28]. Một minh hoạ có tính chất giải trí và rất không hình thức cho khái niệm không tiết lộ thông tin được Quisquater và Guilỏutình bày trong [QG 90]. Cũng có thể xem trong [Jo 88] của Johnson là một tổng quan chặt chẽ hơn về mặt toán học cho các hệ thống chứng minh tương hỗ.

## BÁI TẬP

13.1. Xét một hệ thống chứng minh tương hỗ cho bái toán các thặng dư không bậc hai được mô tả ở hình 13.14. Hãy chứng tỏ rằng hệ thống là đúng đắn và đầy đủ và giải thích tại sao giao thức này không tiết lộ thông tin.

13.2. Hãy tạo ra một hệ thống chứng minh tương hỗ cho bái toán không là thành viên của nhóm con. Hãy chứng mỉnh rằng giao thức của bạn là đúng đắn và đầy đủ.

13.3. Xét một phép chứng minh không tiết lộ thông tin cho các thặng dư bậc hai được trình bày ở hình 13.8.

Hình 13.14. Một hệ thống chứng minh tương hỗ cho các thặng dư không bậc hai.

Đầu vào: Một số nguyên n có phân tích n =pq chưa biết, trong đó p và q là các số nguyên tố, và x $\in$ ?????????

1. Lặp lại các bước sau $\log_2 n$ lần:
2. Vic chọn một số ngẫu nhiên $v \in Z_n^*$ và tính
   $y = v^2 \bmod n$
   Vic chọn ngẫu nhiên i= 0 hoặc 1 và gửi cho Peggy:
   $z = x^i y \bmod n$
3. Nếu $z \in QR(n)$ thì Peggy xác định j= 0, ngược lại Peggy sẽ xác định j=1. Sau đó cô ta gửi j cho Vic.
4. Vic sẽ kiểm tra xem liệu i=j hay không.
5. Vic chấp nhận phép chứng minh của Peggy nếu tính toán ở bước 4 được kiểm tra ở mỗi vòng trong $\log_2 n$ vòng.

a.) Xác định một bộ ba hợp lệ là một bộ ba có dạng (y,i,z), trong đó $y \in QP(n)$, i=0 hoặc 1, $z \in Z_n^*$ và $z^2 \equiv x^i y \pmod n$. Hãy chứng minh rằng số các bộ ba hợp lệ là 2(p-1)(q-1), và mỗi một bộ ba như vậy sẽ được tạo với xác suất như nhau nếu Peggy và Vic tuân theo giao thức.

b) Hãy chỉ ra rằng Vic có thể tạo được các bộ ba có cùng phân bố xác suất mà không biết phân tích n = pq.

c) Hãy chứng minh rằng giao thức này là một giao thức không tiết lộ thông tin hoàn thiên đối với Vic.

13.4. Xét phép chứng minh không tiết lộ thông tin cho bái toán thành viên của nhóm con đã được mô tả ở hình 13.10.

a) Hãy chứng tỏ rằng giao thức này đúng đắn và đầy đủ.

b) Xác minh một bộ ba hợp lệ là một bộ ba có dạng ($\gamma$, i, h), trong đó $\gamma \in Z_n^*$, i = 0 hoặc 1, $0 \le h \le l - 1$ và $\alpha^h \equiv \beta^i \gamma \pmod n$. Hãy chứng tỏ rằng số các bộ ba hợp lệ là $2l$ và mỗi bộ ba như vậy sẽ được tạo với xác suất băng nhau nếu Peggy va Vic tuân thhủ giao thức.

c) Hãy chứng tỏ rằng có thể tạo được các bộ ba có cùng phân bố xác suất mà không cần biết logarithm rời rạc $\log_\alpha \beta$.

d) Chứng minh răng giao thức này là một giao thức không tiết lộ thông tin hoàn thiệt đối với Vic.

13.5. Chứng minh rằng sơ đồ cam kết bít logarithm rời rạc được trình bày ở phần 13.5 là có tính giấu kín không điều kiện và chứng minh rằng nó có tính ràng buộc khi và chỉ khi Peggy không thể tính $\log_\alpha \beta$.

13.6. Giả sử ta sử dụng sơ đồ ràng buộc bịt theo các thặng dư bậc hai được mô tả ở phần 13.5 để có một luận cứ không tiết lộ thông tin cho phép tô đồ thị bằng ba màu. Bằng cách dùng thuật toán giả mạo nêu ở hịnh 13.13. Hãy chứng minh rằng giao thức này là một giao thức không tiết lộ thông tin hoàn thiện đối với Vic.

## TÁI LIỆU ĐỌC THÊM

Kahn [KA 67], Koblit[Ko 87], Konheim[Ko 81], Kranakis[Kr 86], Merezes[Me 93], Meyer và Matyas[MM 82], Patterson [Pa 87], Pomerance[Po 90A], Rueppel [Ru 86], Salomaa[Sa 90], Schneier[Sc 93], Seberry và Pieprzk[SP 89], Simmons [Si 92B], van Tilborg [vT 88] và Welsh [We 88].

Bạn đọc nên đọc thêm một số giao trình và sách chuyên khảo khác về mật mã học sau đây: BecKer và Piper [BP 93], Beutelspacher[Be 94], Brassard[Br 88], Biham và Shamir[BS 93], Denning[De 82],

Các tạp chí nghiên cứu chủ yếu trong mật mã học là Journal of Cryptology và Designs, Codes and Cryptography. Journal of Cryptography là tạp chí của Hiệp hội nghiên cứu mật mã quốc tế (IACR). Hiệp hội này cũng tái trợ hai Hội nghị chính về mật mã học được tổ chức hàng năm là CRYPTO vµ EUROCRYPT.

CRYPTO đã đươc tổ chức từ năm 1981 ở Santa Barabara. Các báo cáo khoa học ở CRYPTO đã được xuất bản hàng năm đáng kể từ 1982:
CRYPTO’ 82[CRS 83] , CRYPTO’ 83[CH 84], CRYPTO’ 84[BC 85] CRYPTO’ 85[WI 96], CRYPTO’ 86[Oo 87], CRYPTO’ [Po 88] CRYPTO’ 88[Go 90], CRYPTO’ 89[BR 90], CRYPTO’ 90[MV 91] CRYPTO’ 9[FE 92], CRYPTO’ 92[BR 93], CRYPTO’ 93[ST 94]
Và CRYPTO’ 94[DE 94].
EUROCRYPT đã được tổ chức hàng năm kể từ năm 1982 (trừ các năm 1983 và 1986), các bái báo khoa học đã công bố gồm:
EUROCRYPT’ 82[BE 83], EUROCRYPT’ 84[BCI 85], EUROCRYPT’ 85[PX86], EUROCRYPT’ 87[CP 88], EUROCRYPT’ 88[GU 88A], EUROCRYPT’ 90[DA 91], EUROCRYPT’ 91[DA 91A], EUROCRYPT’ 92[RU 93] và EUROCRYPT’ 93[HE 94].

Loại hội nghị thứ ba là AUCRYPT / ASIACRYPT đã được tổ chức với sự kết hợp của IACR. Các báo cáo khoa học đã được xuất bản bao gồm AUCRYPT’ 90[SP90], ASIACRYPT’ 91[IRM 93] và AUCRYPT’ 92[SZ92]

# CHƯƠNG 10

# CÁC MÃ XÁC THỰC

## 10.1 MỞ ĐẦU

Ta đã dành nhiều thời gian để nghiên cứu các hệ mật được dùng để đảm bảo độ mật .Mã xác thực sẽ cung cấp phương pháp bảo đảm tình toàn vẹn của bản tin,mghĩa là bản tin phải không bị can thiệp một cách bất hợp pháp và nó thực sự được gửi đi từ mày phát.

Mục đích của chương này là phải có được khả năng xá thực ngay cả khi có một đối phương tích cực-Oscar là người có thể quan sát các bản tin trong kênh.Mục đích này có thể đạt được bằng cách thiết lập một ‘’khoa riêng’’K bằng cách để Alice và Bob chungchung một khoá bí mật trước hki mỗi bản tin được gửi đi.

Trong chương này ta sẽ nghiên cứu đảm bảo xacs thực chứ không phải các mã đảm bảo độ mật.Trong mã này,khoá sẽ được dùng dể tính một mã xác thực cho phép Bob kiểm tra được tính xác thực của thông báo mà anh ta nhận được.Một ứng dụng khác của mã xác thực là để kiểm tra xem các số liệu trong một file lớn có bị can thiệp vào một cách hợp pháp hay không.Nhãn xác thực sẽ được lưu cùng với số liệu:KHOá ĐƯẻc dùng để tạo và kiểm tra dấu xác thực được lưu một cách tách bạch trong một’’vùng’’an toàn.

Ta cũng sẽ chỉ ra rằng,về nhiều khía cạnh mã xác thực cũng tương tự như một sơ đồ chữ kí hoặc tương tự như một maw xác thực thông báo(MAC).Sự khác biệt chính là sự an toàn của một maw xác thực là không điều kiện biên,trong khi đó các sơ đồ chữ kí và MAC lại được nghiên cứu theo quan điểm độ an toàn tính toán.Cũng vậy,khi một maw xác thực (hoặc MAC) được dùng,một bản tin chỉ có thể được kiểm tra bởi người nhận hợp pháp.Trong khi đó baats cứ mỗi ai cũng có thể xác minh được chữ kí bằng cách dùng một thuật toán xác minh công khai.

Bây giờ ta sẽ đưa ra một định nghia hình thức cho thuật ngữ được sử dụng khi nghiên cứu các mã xác thực.

***Định nghĩa 10.1***

*Một mã xác thực là một bộ 4(S,R,K,C)thoả mãn các điều kiện sau :*

1. *S là tập hữu hạn các trạng thái nguồn có thể*

2. *A là tập hợp các nhãn xác thực có thể*
3. *K là một tập hữu hạn các khoá có thể (không gian khoá)*
4. *Với mỗi $k \in K$ có một quy tắc xác thực $e_k : S \rightarrow R$*

*Tập bản tin được xác định bằng $M = S \rightarrow R$*

Nhận xét:

Chú ý một trạng thái nguồn tương đương với một bản rõ.Một bản tin gồm một bản rõ với một nhãn xác thực kèm theo,một cách chính xác hơn có thể coi đó là là một bản tin đã được xác nhận.Một quy tắc xác thực không nhất thiết phải là hàm đơn ánh.

Đểê phát một thông báo (đã được kí).Alice và Bob phải tuân theo giao thức sau.Trước tiên họ phải chộn một khoá ngẫu nhiên $K \in K$.Điều này được thuwc hiện một cách bí mật như trong hệ mật khoá bi mật.Sau đó giả sử rằng Alice muốn gửi một trạng thái nguồn $s \in S$ cho Bob trong một kênh không an toàn>Alice sẽ tính $a = e_k(s)$ và gửi bản tin (s,a)cho Bob.Khi nhận được (s,a) Bob tính $a' = e_K(s)$.Nếu a=a' thì Bob chấp nhận bản tin là xác thực,ngược lại Bob sẽ loại bỏ nó.

Ta sẽ nghiên cứu hai kiểu tấn công khác nhau mà Oscar có thể tiến hành.Trong cả hai loại này,Oscar sẽ là''kẻ xâm nhập vào giưa cuộc''.Các phép tấn công này được mô tả như sau:

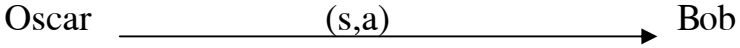
**Giả mạo**

Oscar đưa ra một bản tin (s,a) vào kênh và hi vọng nó sẽ được chấp nhận .Phương pháp này được mô tả trong hình 10.1.

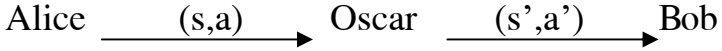
**Thay thế**

Oscar quan sát một bản tin trong (s,a)kênh ,sau đó anh ta biến đổi nó thành(s',a'),trong đó s'=s và hi vọng được Bob chấp nhận như một bản tin xác thực .Bởi vậy anh ta tin sẽ lái được Bob đi tới trạng thái nguồn mới này.Phương pháp này được mô tả như hình 10.2.

**Hình 10.1. Việc giả mạo bởi Oscar**




**Hình 10.2 . Phép thay thế của Oscar**.



Gắn với mỗi phong pháp này là một xác xuất lừa bịp,là xác suất để Oscar thành công trong việc lừa Bob nếu anh ta (Oscar) tuân thủ một chiến lược tối ưu .Các xác suất này được kí hiệu là $Pd_0$(trường hợp giả mạo)và $Pd_1$(trường hợp thay thế) .Để tình $Pd_0$ và $Pd_1$ ta cần phải xác định các phân bố xác suất trên *S* và*K*.Các xác suất này được kí hiệu tương ứng là $p_s$ và $p_k$ .

Giả sử rằng Oscar đã biết mã xác thực và hai phân bố xác suất này.Chỉ có một thông tin mà Alice và Bob có nhưng mà Oscar không được biết là giá trị của khoá K .Điều này tương tự với cách mà chúng ta đã nghiên cứu độ an toàn không điều kiện của các hệ mật khoá bí mật.

## 10.2.TÍNH XÁC SUẤT LỪA BỊP

Trong phần này sẽ xét đến việc tính các xác suất lừa bịp.Ta bắt đầu về một mã xác thực.

*Ví dụ 10.1*

Giả sử $K=R=Z$

*và* $K=Z_3xZ_3$

Với mỗi $(i,j)\in K$ và mỗi $s\in S$ ta xác định

$$e_k(s) = i.s+j \bmod 3$$

Để thuận tiện cho việc nghiên cứu ta dùng ma trận xác thực (ma trận này tạo bằng tất cả các giá trị $e_k(s)$).Với mỗi khoá $K\in K$ và với mỗi $s\in S$ ta đặt nhãn xác thực $e_k(s)$ vào hàng K và cột s của một ma trận M kích thước *K* x*S* .Mảng M được mô tả trên hình 10.3.

*Hình 10.3.Ma trận xác thực*

| Khoá | 0 | 1 | 2 |
|---|---|---|---|
| (0,0) | 0 | 0 | 0 |
| (0,1) | 1 | 1 | 1 |
| (0,2) | 2 | 2 | 2 |
| (1,0) | 0 | 1 | 2 |
| (1,1) | 1 | 2 | 0 |
| (1,2) | 2 | 0 | 1 |
| (2,0) | 0 | 1 | 2 |
| (2,1) | 1 | 0 | 2 |
| (2,2) | 2 | 1 | 0 |

Giả sử rằng khoá được chọn một cách ngẫu nhiên,tức là $p_k(K)=1/9$ đối với mọi $K\in K$. Ta không phải xác định phân bố xác suất $p_S$ vì trong thí dụ này nó khong có ý nghĩa gì.

Trước tiên xét cách tấn công giả mạo,Oscar sẽ chọn ra một trạng thái nguồn s và cố gắng phỏng ddoand\s một nhãn xác thực ''đúng''.Kí hiệu $K_0$ là khoá đang sử dụng (mà Oscar không biết).ócả sẽ thành công trong việc đánh lừa Bob nếu anh ta phỏng đoán $a_0=e_{K0}(s)$.Tuy nhiên với bất kì $s\in S$ và $a\in R$ dễ dàng thấy rằng ,chỉ có đúng 3(chứ không phải là 9)quy tắc xác thực $K\in K$ sao cho $e_k(s)$ =a.(Nói cách khác mỗi kí hiệu chỉ xuất hiện 3 lần trong mỗi cột của ma trận xác thực ).Bởi vậy dẫn tới $Pd_0=1/3$.

Phân tích phép thay thế có phức tạp hơn một chút.Giả sử Oscar đã quan sát được trên kênh 1 bản tin (0.0).Nhờ đó anh ta đã biết một thông tin nào đó về khoá:anh ta biết rằng :

$$K_0\in\{(0,0),(1,0),(2,0)\}$$

Bây giờ ,giả sử Oscar thay bản tin (0,0) bằng bản tin (1,1).Khi đó anh ta sẽ lừa bịp thành công khi và chỉ khi $K_0=(1,1)$ ,xác suất để $K_0$ là khoá bằng 1/3 vì khoá nằm trong tập {(0,0),(1,0),(2,0)}.

Có thể thực hiện một phân tích tương tự đối với bất kì một phép thay thế nào mà Oscar tiến hành.Nói chung nếu Oscar quan sát một bản tin (s,a) và thấy nó bằng một bản tin bất kì (s',a') trong đó s'=s thì anh ta sẽ đánh lừa được Bob với xác suất 1/3.Ta có thể thấy rõ điều này như sau .Việc quan sát được (s,a) sẽ hạn chế khóa và một trong ba khả năng.Trong khi đó với một phép chọn (s',a') chỉ có một khoá chứ không phải ba khoá có thể )theo quy tắc a là nhãn xác thực của s'.

Bây giờ ta sẽ thảo luận cách tính toán tổng quát cho các xác suất lừa bịp.Trước tiên ta hãy xát $Pd_0$.Cũng như trên $K_0$ là khoá được chọn bởi Alice và Bob.Với $s\in S$ và $a\in R$ ta xác định payoff(s,a)là xác suất để Bob chấp nhận bản tin (s,a) là bản tin xác thực .Dễ dàng thấy rằng :

$$\text{Payoff}(s,a) = \text{prob}(a=e_K(s))$$
$$=\sum_{K\in K\ (ek(s)=a)} p_K(K)$$

Nghĩa là payoff(s,a) được tính bằng cách chọn các hàng của ma trận xác thực có phần tử a nằm trong cột s và lấy tổng xác suất của các khoá K tương ứng.

Để cơ hội thành công là lớn nhất.Oscar phỉa chọn (s,a) sao cho payoff(s,a) là cực đại .Bởi vậy:

$$Pd_0 =\max\{\text{payoff}(s,a): s\in S.a\in R\} \qquad (10.1)$$

Chú ý rằng $Pd_0$ không phụ thuộc vào phân bố xác suất $p_S$

Việc tính $Pd_1$ có khó hơn một chút và nó có thể phụ thuộc vào $p_S$.Trước tiên ta sẽ xét bài toán sau:Giả sử Oscar quan sát được thông báo (s,a) trong kênh.Oscar sẽ thay (s,a) bằng một bản tin (s',a') nào đó ,trong đó $s' \neq s$.Khi đó,với $s,s' \in S$ ,$s \neq s'$ và $a,a' \in R$ ta định nghĩa payoff(s',a';s,a) là xác suất để phép thay thế (s,a) bằng (s',a') thành công(để đánh lừa Bob) .Khi đó có thể tính như sau :

$$Payoff(s',a';s,a) = prob(a' = e_{Ko}(s') \mid a = e_{Ko}(s))$$

$$= \frac{prob(a' = e_K(s') \Lambda a = e_K(s))}{prob(a = e_K(s))}$$

Tử số của phân số này được tính bằng cách chọn các hàng của ma trận xác thực có giá trị a trong cột s và giá trị a' trong cột s'và lấy tổng các xác suất của các khoá tương ứng.Vì Oscar muốn tăng cực đại cơ hội đánh lừa Bob nên anh ta tính :

$$P_S = \max\{payoff(s',a';s,a); s' \in S, s \neq s', a \in R\}$$

Đại lượng p,kí hiệu để Oscar đánh lừa Bob bằng một phép thay thế khi đã quan sát được bản tin (s,a) trên kênh.

Bây giờ phải làm thế nào để tính để tinhs xác suất lừa bịp $Pd_1$?Rõ ràng là ở đây ta ta phải tính trung bình các giá trị của lượng $p_S$ theo các xác suất $p_M(s,a)$ quan sát các bản tin trên kênh.Nghĩa là $Pd_1$ được tính bằng :

$$Pd_1 = \sum_{(S,a) \in M} p_M(s,a).p_M \qquad (10.2)$$

Phân bố xác suất $p_M$ như sau:

$$P_M(s,a) = p_s(s) x\, p_K(a \mid s)$$

$$= p_S(s) x \sum_{(K \in K;\, ek(s)=a)} p_K(K)$$

$$= p_S(s) x payoff(s,a)$$

Trong ví dụ 10.1:

$$Payoff(s,a) = 1/3$$

Với $\forall s',a',s,a, s \neq s'$ .Bởi vậy $Pd_1 = 1/3$ đối với mọi phân tố xác suất $p_S$ (nói chung $Pd_1$ phụ thuộc vào $p_S$).

Trong ví dụ sau đây sẽ xét việc tính $Pd_0$ và $Pd_1$ .

Ví dụ 10.2:

Xét ma trận trên hình 10.4Giả sử các phân bố xác suất trên *S và K* là:

$$P_S(i) = 1/4$$

$1 \leq i \leq 4$ và

$p_K(1) = 1/2$ ; $p_K(2) = p_K(3) = 1/4$

***Hình 10.4 Ma trận xác thực***

| Khoa | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 2 |
| 2 | 2 | 2 | 1 | 2 |
| 3 | 1 | 2 | 2 | 1 |

Các giá trị payoff(s,a) như sau :

Payoff(1,1) =3/4 Payoff(1,1) =1/4
Payoff(2,1) =1/2 Payoff(2,2) =1/2
Payoff(3,1) =3/4 Payoff(3,2) =1/4
Payoff(4,1) =1/4 Payoff(4,2) =3/4

Bởi vậy $Pd_0$=3/4 .Chiến lược đánh lừa tối ưu của Oscar là đưa một thông báo bất kì trong số các thông báo (1,1),(3,1) hoặc (4,2) vào kênh.

Bây giờ ta sẽ chuyển sang tính $Pd_1$.Trước hết ta đưa các giá trị khác nhau của payoff(s',a';s,a).

| | (1,1) | (1,2) | (2,1) | (2,2) | (3,1) | (3,2) | (4,1) | (4,2) |
|---|---|---|---|---|---|---|---|---|
| (1,1)<br>(1,2) | | | 2/3<br>0 | 1/3<br>1 | 2/3<br>1 | 1/3<br>0 | 1/3<br>1 | 2/3<br>0 |
| (2,1)<br>(2,2) | 1<br>1/2 | 0<br>1/2 | | | 0<br>1/2 | 1<br>1/2 | 0<br>1/2 | 1<br>1/2 |
| (3,1)<br>(3,2) | 2/3<br>1 | 1/3<br>0 | 2/3<br>0 | 1/3<br>1 | | | 0<br>1 | 1<br>0 |
| (4,1)<br>(4,2) | 1<br>2/3 | 0<br>1/3 | 0<br>2/3 | 1<br>1/3 | 0<br>1 | 1<br>0 | | |

Như vậy ta có $p_{1.1}$=2/3,$p_{2.2}$=1/2,$p_{3.3}$=1 với mọi giá trị s,a khác .Khi đó việc đánh giá $Pd_1$ sẽ trở nên rất đơn giản:$Pd_1$=7/8.Chiến lược thay thế tối ưu của Oscar là:

(1,1) → (2,1)
(1,2) → (2,2)
(2,1) → (1,1)
(2,2) → (1,1)
(3,1) → (4,2)
(3,2) → (1,1)
(4,1) → (1,1)
(4,2) → (3,1)

Chiến lược này thực sự dẫn đến $Pd_1=7/8$
Việc tính toán $Pd_1$ trong ví dụ 10.2 dễ hiểu nhưng khá dài dòng .Trên thực tế có thể đơn giản hóa việc tính $Pd_2$ dựa trên nhận xét là ta đã thực hiện việc chia cho đại lượng payoff(s,a) khi tính $P_{s,a}$ và sau đó Lại nhân với payoff(s,a) khi tính $Pd_1$ .Dĩ nhiên là hai phép tính này loại bỏ nhau.Giả sử định nghĩa :

$q_{s,a}=\max\{\sum_{\{K\in K:ek(s)=a,ek(s')=a'_1\}} p_K(K) : s'\in S, s'\neq s, a'\in A\}$
Với mọi s,a. Khi đó có công thức đơn giản hơn sau:

## 10.3.CÁC GIỚI HẠN TỔ HỢP

Ta đã thấy ràng độ an toàn của một mã xác định được đo bằng
Các xác xuất lừa bịp . Bởi vậy cần xây dựng các mã sao cho các xác Xuất này nhỏ tới mức có thể .Tuy nhiên những khía canh khác cũng Rất qoan trọng .Ta xem xét một số vấn đề cấn qoan tâm trong mã xác thực .
1.Các xác xuất lừa bịp $Pd_0$ và $Pd_1$ phải đủ nhỏ để đạt được mức an toàn mong muốn .
2.số các trạng thái nguồn phải đủ lớn để có thể truyền các thông tin cần thiết bằng cách gán một nhãn xác thực vào một trạng thái nguồn .
3. Kích thước của không gian khóa phải được tối thiểu hóa và các giá trị của khóa phải truyền qua một kênh an toàn (Cần chú ý rằng phải thay đổi khóa sau mỗi lần truyền tin giống như khi dùng OTP).
Trong phần này sẽ xác địinh giới hạn dưới đối với các xác suất lừa bịp và chúng được tính theo các tham số của mã.Hãy nhớ lại rằng ta đã định nghĩa mã xác thực là một bộ bốn (S,R,K,E).Trong phần này ta sẽ ký hiệu $|R|=l$
Giả sử cố định một trạng thái nguồn $s\in S$.Khi đó có thể tính :

$$\sum_{a\in R} payoff(s,a)=\sum_{a\in R}\sum_{(K\in K:ek(s)=a\}} p_K(K)$$
$$=\sum_{K\in K} p_K(K)$$
$$=1$$
Bởi vậy với mỗi $s\in S$,tồn tại một nhãn xác thực a(s) sao cho :
$Payoff(s,a(s))\geq 1/l$.
Dễ dàng rút ra định lý sau:

***Đinh lý 10.1***

*Giả sử (S,R,K,E) là một mã xác thực .Khi đó* $Pd_0 \geq 1/l$ *trong đó* $l = |R|$*.Ngoài ra* $Pd_0 = 1/l$ *khi và chỉ khi :*

$$\sum_{\{K \in K : ek(s)=a\}} p(K) = 1/l \qquad (10.4)$$

*với mỗi* $s \in S, a \in R$.

Baauy giờ ta sẽ chuyển sang phương pháp thay thế .Giả sử cố định s,a và $s', s \neq s'$.Ta có:

$$\sum_{a' \in R} payoff\ (s', a'; s, a) = \sum_{a' \in R} \frac{\sum_{\{K \in K : ek(s)=a, ek(s')=a'\}} p_K(K)}{\sum_{\{K \in K : ek(s)=a\}} p_K(K)}$$

$$= \frac{\sum_{\{K \in K : ek(s)=a\}} p_K(K)}{\sum_{\{K \in K : ek(s)=a\}} p_K(K)} = 1$$

Như vậy tồn tại một nhãn thực $a'(s', s, a)$ sao cho :

Payoff$(s', a'(s', s, a) : s, a) \geq 1/l$

Định lý sau sẽ rút ra kết quả :

***Định lý10.2***

*Giả sử (S,R,K,E) là một mã xác thực .Khi đó* $Pd_1 >= 1/l$ *trong đó* $L = |R|$*.Ngoài ra* $Pd_1 \geq 1/l$ *khi và chỉ khi :*

$$\frac{\sum_{\{K \in K : ek(s)=a, ek(s')=a'\}} p_K(K)}{\sum_{\{K \in K : ek(s)=a\}} p_K(K)} = 1/l$$

*Với mỗi* $s, s' \in S, s = s', a, a' \in R$

*Chứng minh*

Ta có : $Pd_1 = \sum_{(s,a) \in M} p_M(s,a).p_{s,a} \geq \sum_{(s,a) \in M} p_M(s,a)/l = 1/l$

Ngoài ra dấu bằng chỉ tồn tại khi và chỉ khi $p_{s,a} = 1/l$ với mỗi (s,a) .Tuy nhiên điều kiện này lại tương đương với điều kiện :

Payoff$(s', a'; s, a) = 1/l$ với mọi (s,a).

*Định lý 10.3*

*Giả sử (S,R,K,E) là một mã xác thực trong đó* $l = |R|$*.Khi đó*$Pd_0 = Pd_1 = 1/l$ *khi và chỉ khi :*

$$\sum_{\{K \in K : ek(s)=a, ek(s')=a'\}} p_K(K) = 1/l^2 \qquad (10.6)$$

*Với mọi* $s,s' \in S, a,a' \in R, s \neq s'$
*Chứng minh*
Các phương trình (10.4)và (10.5) boa hàm phương trình (10.6).Ngược lại , phương trình (10.6) kéo theo các phương trình (10.4) và(10.5).
Nừu các khóa là đồng khả năng thì ta nhận được hệ quả sau:
*Hệ quả 10.4*:
Giả sử (S,R,K,e) là một mã xác thực ,trong đó l=$|R|$ và các khoá chọn đồng xác suất.Khi đó $Pd_0=Pd_1=1/l$ khi và chi khi :

$$|\{K \in K : e_K(s)=a, e_K(s')=a'\}| = |K|/l^2 \qquad (10.7)$$

Với mọi $s,s' \in S, s' \neq s, a,a' \in R$.

### 10.3.1.Các mạng trực giao

Trong phần này ta xét các mối liên quan giưa các mã xác thực và các cấu trúc tổ hợp được gọi là các mảng trực giao.Trước tiên ta sẽ đưa ra các định nghĩa:

***Định nghĩa 10.2:***
*Một mạng trực giao 0A(n,k,λ)là một mảng kích thước* $\lambda n^2 xk$ *chứa n kí hiệu sao cho trong hai cột bất kì của mảng mỗi cặp trong* $n^2$ *cặp kí hiệu chỉ xuất hiện trong đúng* $\lambda$ *hàng.*
Các mạng trực giao là các cấu trúc đã được nghiên cứu kĩ trong lí thuyets thiết kế tổ hợp và tương đương với các cấu trúc khác như các hình vuông Latinh trực giao hỏi các lưới ...
Trong hình 10.5 ta đưa ra một mảng trực giao 0A(3.3.1) nhận được từ ma trận xác thực ở hình 10.3.

*Hình 10.5. 0A(3.3.1)*

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 2 & 2 & 2 \\ 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \\ 0 & 2 & 1 \\ 1 & 0 & 2 \\ 2 & 1 & 0 \end{bmatrix}$$

Có thể dùng một mảng trực giao bất kì 0A(n,k,λ) để xây dựng một mã xác thực có $Pd_0=Pd_1=1/n$ như được nêu trong định lí sau:

*Định lí 10.5.*
*Giả sử có một mảng trực giao 0A(n,k,λ).Khi đó cùng tồn tại một mã xác thực (S,A,K,E).trong đó* $|S|=k, |R|=n, |K|=\lambda n^2$ *và* $Pd_0=Pd_1=1/n$.
*Chứng minh:*
Hãy dùng mỗi hàng của mảng trực giao làm một quy tắc xác thực với xác suất như nhau bằng $1/(\lambda n^2)$.Mối liên hệ tương ứng giưa mảng trực giao và mã xác thực được cho ở bảng dưới đây.Vì phương trình (10.7) được thoả mãn nên ta có thể áp dụng hệ quả 10.4 để thu được một mã xác thực có các tính chất đã nêu.

| Mảng trực giao | Mã xác thực |
|---|---|
| Hàng | Quy tắc xác thực |
| Cột | Trạng thái nghuồn |
| Kí hiệu | Nhãn xác thực |

### 10.3.2.Phương pháp xây dựng và các giới hạn đối với các 0A

Giả sử ta xây dựng một mã xác thực từ một 0A(n,k,λ).Tham số n sẽ xác định số các nhãn (tức là độ an toàn của mã).Tham số k xác định số các trạng thái nguồn mà mã có thể thích ứng.Tham số λ chỉ quan hệ tới số khoá (là $\lambda n^2$ ).Dĩ nhiên trường hợp λ=1là trường hợp mong muốn nhất tuy nhiên ta sẽ thấy rằng đôi khi cần phải dùng các mảng trực giao có λ lớn hơn.Giả sử ta muốn xây dựng một mã xác thực ới tập nguồn xác định S và có một mức an toàn ε xác định (tức là để $Pd_0<\varepsilon$ và $Pd_1<\varepsilon$).Khi đó mảng trực giao thích hợp phải thoả mãn các điều kiện sau:

1. $n\geq 1/\varepsilon$
2. $k\geq |S|$.(Xét thấy có thể loại một hoặc một số cột khỏi mảng trực giao và mảng kết quả vẫn còn là một mảng trực giao,bởi vậy không đòi hỏi $k=|S|$).
3. λ được tối thiểu hoá ,tuỳ thuộc vào các điếu kiện trên được thoả mãn

Trước tiên xét các mảng trực giao có λ=1 .Với một giá trị n cho trước ,ta cần làm cực đại hoá số cột,sau đây là một số điều kiện cần để tồn tại .

Định lí 10.6.

*Giả sử tồn tại một 0A(n,k,λ) .Khi đó k≥ n+1*

*Chứng minh:*

Cho A là một 0A(n,k,l) trên tập kí hiệu X={0,1...n-1}.Giả sử π là một phép hoán vị của X và ta hoán vị các kí hiệu trong một cột bất kì của A theo phép giao hoán π.Kết quả là ta lại có một 0A(n,k,l).Bởi vậy bằng cách áp dụng liên tiếp các phép vị kiểu này ,có thể xem (mà không làm mất tính tổng quát) rằng hàng đầu tiên cuả A là (00...0).

Tiếp theo ta sẽ chỉ ra rằng mỗi kí hiệu chỉ xuất hiện đùng n lần trong mỗi cột của A.Hãy chọn hai cột (chẳng hạn c và c')và cho X là một kí hiệu bất kì .Khi đó với mỗi kí hiệu x' tồn tại một hàng duy nhất của A trong đó x ở cột c và x' ở cột c'.Cho x' thay đổi trên X ta thấy rằng x xuất hiện đúng n lần trong cột c.

Vì hàng thứ nhất là (00...0) nên ta đã vét cạn các khả năng xuất hiện của các cặp được sắp (0.0).Bởi vậy không có một hàng nào khác có nhiều hơn một kí hiệu o.Bây giờ ta sẽ đếm số các hàng chứa ít nhất một kí hiệu 0.Tổng số là 1+k(n-1).Tuy nhiên tổng này không thể lớn hơn tổng số các hàng trong A (bằng $n^2$).Bởi vậy $1+k(n-1)\leq n^2$ hay $k\leq n+1$ như mong muốn .

Bây giờ ta sẽ đưa ra một cấu trúc cho mảng trực giao có λ=1 ,trong đó k=n .Trong thực tế đây chính là cấu trúc đã dùng để thu được mảng trực giao nêu ở hình 10.5.

***Định lí 10.7***

*Giả sử p là một số nguyên tố.Khi đó tồn tại một mảng trực giao 0A(p.p.1).*

*Chứng minh:*

Mảng này sẽ là một cấp $p^2\times p$,trong đó các hàng được lập chỉ số trong $Z_P x Z_P$ và các cột được lập chỉ số trong $Z_P$ .Phần tử ở hàng (i,j) và cột x được tính bằng i.x+j mod p.

Giả sử chọn hai cột x và y,x≠y,và hai kí hiệu a,b.Ta cần tìm một hàng duy nhất (i,j) sao cho a nằm trong cột x và y nằm trong cột y của hàng (i,j).Vì thế cần giải hai phương trình:

a=i.x+j

b=i.y+j

theo các ẩn i và j (trong đó tất cả các phép tính số học được thực hiện trong trường Z).Nhưng hệ này có nghiệm duy nhất:

$$i=(a-b)(x-y)^4 \bmod p$$
$$j=a-y.x \bmod p$$

Bởi vậy ta có một mảng trực giao.

Nhận xét rằng một 0A(n,n,1) bất kì có thể mở rộng thêm một cột để tạo thành 0A(n,n+1,1)(xem các bài tập ).Vì thế dùng định lí 10.7 có thể nhận được vô hạn các 0A đạt được giới hạn của định lí 10.6 với dấu bằng.

Định lí 10.6 cho biết rằng $\lambda>1$ nếu $k>n+1$.Ta sẽ chứng minh một kết quả tổng quát hơn khi đặt giới hạn dưới của $\lambda$ như một hàm của n và k.Tuy nhiên,trước tiên cần đưa ra một bất đẳng thức quan trọng sẽ dùng trong chứng minh.

**Bổ đề 10.8.**

Giả sử $b_1....b_m$ là các số thực.Khi đó:

$$m\sum_{m=1}^{n} b_1^2 \geq (\sum_{m=1}^{n} b_1)^2$$

*Chứng minh*

áp dụng bất đẳng thức Jensen(Định lí 2.5) với $f(x)=-x^2$ và $a_1=1/m.1\leq i\leq m$.Hàm f là liên tục là và lõm.Vì thế ta nhận được :

$$\sum_{i=1}^{m} \frac{b_1^2}{m} \leq \left(\sum_{i=1}^{m} \frac{b_1}{m}\right)^2$$

Từ đây dễ dàng rút ra kết quả mong muốn.

Định lí 10.9.

Giả sử tồn tại một 0A(n,k,λ).Khi đó

$$\lambda \geq \frac{k(n-1)+1}{n^2}$$

X

*Chứng minh*

Cho A là một 0A(n,k,λ) trên tập kí hiệu X={0,1.....n-1},trong đó hàng đầu tiên của A là (0,0....0)(giả thiết này không làm mất tính tổng quát như đã thấy trong định lí 10.6).

Kí hiệu các tập hàng của Alà R và $r_1$ là hàng đầu tiên,cho $R_1$=R\{r1}.Với một hàng bất r của A,kí hiệu $x_r$ chỉ số lần xuất hiện của 0 trong hàng r.Có thể dễ dàng tình được tổng số lần xuaat hiện của 0 trong $R_1$.Vì mỗi kí hiệu phải xuất hiện đúng λn lần trong mỗi cột của Anên ta có:

$$\sum_{r\in R_1} x_r = k(\lambda.n-1)$$

Bây giờ số lần xuất hiện cặp được sắp (0,0) ở các hàng trong $R_1$ là:

$$\sum_{r\in R_1} x_r(x_r-1) = \sum_{r\in R_2} x_r^2 - \sum_{r\in R_1} x_r$$

$$= \sum_{r\in R_2} x_r^2 - k(\lambda.n-1)$$

áp dụng bổ đề (10.8) ta có:

$$\sum_{r\in R_1} x_r^2 \geq \frac{(k(\lambda.n-1))^2}{\lambda.n^2-1}$$

và bởi vậy :

$$\sum_{r\in R_1} x_r(x_r-1) \geq \frac{(k(\lambda.n-1))}{\lambda..n^2-1} - k(\lambda.n-1)$$

Mặt khác,trong một cặp cột cho trước bất kì,cặp được sắp (0,0) xuất hiện trong đúng λ hàng .Vì có k(k-1)cặp các cột được sắp nên dẫn đến số lần xuất hiện của cặp được sắp (0,0) trong các hàng của R đúng bằng (λ-1)k(k-1).Bởi vậy ta có:

$$(\lambda\text{-}1)k(k\text{-}1) \geq \frac{(k(\lambda.n-1))^2}{\lambda.n^2-1} - k(\lambda.n-1)$$

và do đó :

$$((\lambda\text{-}1)k(k\text{-}1)+k(\lambda n\text{-}1)(\lambda n^2\text{-}1) \geq (k(\lambda n\text{-}1))^2$$

Khai triển ta có:

$$\lambda^2kn^2\text{-}\lambda k.n^2\text{-}\lambda^2n^2+\lambda^2n^3\text{-}\lambda k+k+\lambda\text{-}\lambda n \geq \lambda^2kn^2\text{-}2\lambda kn+k$$

hay:

$$\text{-}\lambda^2n^2+\lambda^2n^3 \geq \lambda kn^2+\lambda k\text{-}\lambda+\lambda n\text{-}2\lambda kn$$

hoặc $\lambda^2(n^3\text{-}n^2) \geq \lambda(k(n\text{-}1)^2+n\text{-}1)$

Cuối cùng,chia hai vế cho λ(n-1) ta có :

$$\lambda n^2 \geq k(n\text{-}1)+1$$

Đây chính là giới hạn cần tìm.

Kết quả sau thiết lập sự tồn tại của một lớp vô hạn các mảng trực giao đạt được giới hạn nêu trên với đấu ‘’=’’.

***Định lí 10.10***.

*Giả sử p là một số nguyên tố và d≥2 là một số nguyên.Khi đó tồn tại một mảng trực giao 0A(p.($p^d$-1)/(p-1).$p^{d-2}$*

*Chứng minh:*

Kí hiệu $(Z_P)^d$ là không gian véc tơ chứa tất cả bộ d trên $Z_P$.Ta sẽ xây dựng A (là một $0A(p,(p^d-1)/(p-1),p^{d-2})$ trong đó các hàng và các cột được lập chỉ số theo các véc tơ trong $(Z_P)^d$.Các phần tử của A sẽ là các phần tử của $Z_P$.Tập hợp các hàng được xác định là $R=(Z_p)^d$):tập các cột là :

$$C = \{(c_1...c_d)\in(Z_p)^d: \exists j,0\le j\le d-1 ,c_1=...=c_j=0,c_{j+1}=1\}$$

R chứa tất cả các véc tơ trong $(Z_P)^d$,bởi vậy $|R|=p^d$.C chứa tất cả các véc tơ khác không có toạ độ khác 0 đầu tiên bằng 1.Nhận thấy rằng:

$$|C|=\frac{p^c-1}{p-1}$$

và không có hai véc tơ nào trong C là các bội vô hướng của nhau.

Bây giờ vuới mỗi véc tơ $r' \in R$ và mỗi $c'\in C$ ta định nghĩa:

$$A(r'.c')=r'.c'$$

Trong đó "."kí hiệu tích trong hai véc tơ (được rút gọn theo mod p).

Ta sẽ chứng minh A là mảng trực giao mong muốn.Cho $b',c'\in C$ là hai cột khác nhau và cho $x,y\in Z_P$.Ta sẽ tính số hàng r' để $A(r',b')=x$ và $A(r',b')=y$.Kí hiệu $r'=(r_1,r_2....r_d).b'=(b_1,b_2....b_d)$ và $c'=(c_1,c_2....c_d)$.Hai phương trình $r'.b'=x$ và $r'.c'=y$ có thể được viết thành hai phương trình tuyến tính trong $Z_P$

$$b_1.r_1+...+b_d.r_d=x$$
$$c_1.r_1+...+c_d.r_d=y.$$

Đây là hai phương trình tuyến tính với d ẩn $r_1...r_d$.Vì các bội b'và c' không phải là các bội vô hướng của nhau nên hai phương trình trên là độc lập tuyến tính.Bởi vậy hệ này có không gian nghiệm (d-2) chiều.Nghĩa là số các nghiệm (số các hàng trong đó x nằm ở cột b' và y ở cột c')bằng $p^{d-2}$ theo mong muốn.

Ta sẽ làm một ví dụ nhỏ minh hoạ cách xây dựng này:

*Ví dụ 10.3*

Giả sử lấy p=2,d=3,khi đó ta sẽ xây dựng một 0A(2,7,2).Ta có :

R={000,001,010,011,100,101,110,111}

và C={001,010,011,100,101,110,111}

Ta nhận được kết quả là mảng trực giao như trên hình 10.6

Hình 10.6.Một 0A(2,7,2).
$$\begin{bmatrix} 0&0&0&0&0&0&0\\ 1&0&1&0&1&0&1\\ 0&1&1&0&0&1&1\\ 1&1&0&0&1&1&0\\ 0&0&0&1&1&1&1\\ 1&0&1&1&0&1&0\\ 0&1&1&1&1&0&0\\ 1&1&0&1&0&0&1 \end{bmatrix}$$

***10.3.3Đặc trưng của mã xác thực*** .

Cho tới giờ ta đã nghiên cứu các mã xác thực nhận được từ các mảng trực giao.Ta cũng đã xem xét các điều kiện tồn tại cần thiết về việc xây dựng các mảng trực giao .Vấn đề ở đây là liệu có các phương pháp khác tốt hơn các mảng trực giao không?Tuy nhiên hai định lí đặc trưng sẽ cho biết rằng nếu chỉ giới hạn mối quan tâm tới các mã xác thực có xác suất lừa bịp nhỏ tới mức co thể thì vấn đề trên không cần phải đặt ra nữa.

Trước tiên ta sẽ chứng minh một định lí đảo một phần của định lí 10.5.

*Định lí 10.11*.

*Giả sử (S,A,K,E)là một mã xác thực trong đó* $|R|=n$ *và* $Pd_0=Pd_1=1/n$.*Khi đó* $|K|\geq n^2$.*Hơn nữa* $|K|=n^2$ *khi và chỉ khi có một mảng trực giao 0A(n.k.l) trong đó* $|S|=k$ *và* $p_K(K)=1/n^2$ *với mọi khoá* $K\in K$ .

*Chứng minh*:

Cố định hai trạng thái nguồn tuỳ ý s và s' ,s=s' và xét phương trình (10.6).Với mỗi cặp được sắp (a,a') của các nhãn xác thực ta xác định :

$$K_{a,a'}=\{K\in K : e_K(s)=a, e_K(s')=a'\}.$$

Khi đó $|K|>0$ với mọi cặp (a,a').Cũng thấy rằng các tập $K_{a,a'}$ này rời nhau (có $n^2$ tập).Bởi vậy $K\geq n^2$.

Bây giờ giả sử rằng $|K|=n^2$ .Khi đó trị $|K_{a,a'}|=1$,với mọi cặp (a,a') và từ phương trình (10.6) ,cho ta thấy rằng $p_K(K)=1/n^2$ với mọi khoá $K\in K$.

Vấn đề còn lại là phải chứng tỏ ma trận xác thực sẽ tạo nên ma trận trực giao 0A(n,k,l) .Xét các cột lấy chỉ số theo các trạng thái nguồn s và s'.Vì $|K_{a,a'}|=1$ với mọi (a,a') nên mỗi cặp được sắp xuất hiện dúng một lần trong hai cột này.Vì s,s' là tuỳ ý nên mỗi cặp được sắp xuất hiện đúng một lần trong hai cột bất kì.

Đặc trưng sau đây có khó hơn một chút chúng ta chỉ phát biểu mà không chứng minh .

*Định lí 10.2*

*Giả sử (S,A,K,E) là một mã xác thực ,trong đó* $|A|=n$ *và* $Pd_0=Pd_1=1/n$.*Khi đó* $|K|\geq k(n-1)+1$.*Hơn nữa* $|K|=k(n-1)+1$ *khi và chỉ khi có một mảng trực giao* $0A(n,k,\lambda)$,*ở đây* $|S|=k,\lambda=(k(n-1)+1)/n^2$ và $p_K(K)=1/(k(n-1)+1)$ *với mọi khoá* $K\in K$.

Nhận xét.Chú ý rằng định lí 10.10 tạo ra một lớp vô hạn các mảng trực giao đạt được giới hạn ở định lí 10.12 với dấu "=".

## 10.4.CÁC GIỚI HẠN ENTROPY

Trong phần này chúng ta dùng kĩ thuật entropy để nhận được các giới hạn về các xác suất lừa bịp .Trước tiên ta sẽ xét các giới hạn đối với $Pd_0$.

*Định lí 10.13*

*Giả sử (S,R.K,E) là một mã xác thực .Khi đó*

$$LogPd_0\geq H(K|M)-H(K)$$

*Chứng minh:*

Từ phương trình (10.1) ta có :

$$Pd_0\geq \max\{payoff(s,a):s\in S,a\in R\}$$

Vì giá trị cực của payoff(s,a) phải lớn hơn trung bình các trọng số của chúng nên ta nhận được:

$$Pd_0\geq\sum_{s\in S,a\in R}p_M(s,a)payoff(s,a)$$

Như vậy thoe bất đẳng thức Jensen(định lí (2.5) ta có :

$$LogPd_0\geq\log\sum_{s\in S,a\in R}p_M(s,a)payoff(s,a)$$
$$\geq\sum_{s\in S,a\in R}p_M(s,a)\log payoff(s,a)$$

Theo phần 10.2:

$$P_M(s,a)=p_s(s)x\ payoff(s,a)$$

Ta thấy rằng:

$$Log\ Pd_0\geq\sum_{s\in S,a\in R}p_s(s)payoff(s,a)\ \log payoff(s,a)$$

Bây giờ ta thấy rằng $payoff(s,a)=p_R(a|s)$(tức là xác suất để a là nhãn xác thực với điều kiện s là trạng thái nguồn ).Bởi vậy:

$$LogPd_0\geq\sum_{s\in S,a\in R}p_s(s).p_R(a|s)\ \log p_R(a|s)\ =-H(A|S)$$

Theo định nghĩa của entropy có điều kiện .Ta sẽ hoàn chỉnh chứng minh định lí bằng cách chỉ ra rằng: $-H(A|S)=H(K|M)-H(K)$.Điều kiện này được rút ra từ các đồng nhất thức cơ bản của entropy.Một mặt ta có :

$$H(K,A,S)=H(A|K,S)+H(A|S)+H(S)$$

Mặt khác ta tính:

$$H(K,A,S)=H(A|K,S)+H(K,S)=H(S)+H(K)$$

ậ đây ta có sử dụng điều kiện $H(A|K,S)=0$ vì khoá và trạng thái nguồn sẽ xác định nhãn xác thực một cách duy nhất .Ta cũng dùng đẳng thức $H(A|S)=H(K)+H(S)$ vì nguồn và khoá là các biến cố độc lập.

So sánh hai biểu thức biểu thị H(K,S,A) ta có:

$$-H(A,S)=H(K|A,S)-H(K)$$

Tuy nhiên thông báo m=(s,a) được xác định gồm một trạng thái nguồn và một trạng thái nhãn xác thực(nghĩa là $M=SxA$).Bởi vậy:

$$H(K|A,S)=H(K|M)$$

Định lí được chứng minh.

Sau đây ta sẽ chỉ đưa ra mà không chứng minh giới hạn tương tự cho $Pd_1$.

*Định lí 10.4*

*Giả sử rằng (S,A,K,E) là một mã xác thực .Khi đó*

$$Log Pd_1 \geq H(K/M^2)-H(K/M)$$

Cần phải xác định giới hạn entropy theo biến ngẫu nhiên $M^2$.Giả sử ta xác thực hai trạng thái nguồn khác nhau dùng cùng một khoá K.Theo cách này ta nhận được một cặp được sắp các banr tin $(m_1,m_2)\in MxM$.Để xác định phân bố xác suất trên MxM,cần phải xác định xác suất trên SxS với điều kiện $p_{sxs}(s,s)=0$ với mọi $s\in S$(nghĩa là không cho phép lặp lại trạng thái nguồn ).Các phân bố xác suất trên K và SxS sẽ dẫn đến phân bố xác suất trên $MxM$ tương tự như phân bố xác suất trên K và S sẽ tạo nên một phân bố xác suất trên M Để minh hoạ cho hai giới hạn trên ,xét cấu trúc mảng trực giao cơ bản và chỉ ra rằng cả hai giới hạn trong định lí 10.13 và 10.14 đều đạt được với dấu bằng.Trước hết ta dễ thấy rằng:

$$H(K)=\log \lambda n^2$$

Vì mỗi một trong $\lambda n^2$ quy tắc xác thực đều được chọn đồng xác suất.Tiếp theo ta sẽ quay lại việc tính toán $H(K|M)$.Nừu đã quan sát được một bản tin m=(s,a) nào đó thì điều này sẽ giới hạn các khóa sẽ nằm trong tập con có lực lượng $\lambda n$.Mỗi khoá trong $\lambda n$ khóa này sẽ có

tập con như nhau .Vì thế $H(K|m)=\log\lambda n$ với bản tin n bất kì .Khi đó ta có :

$$H(K|M)=\sum_{m\in M}p_M(m)H(K|m)$$
$$=\sum_{\in M}p_M(m)\log\lambda n$$
$$=\log\lambda n$$

Như vậy ta có:

$$H(K|M)-H(K)=\log\lambda n-\log\lambda n^2=-\log n=\log Pd_0$$

Như vậy giới hạn thoả mãn với dấu "=".

Nừu ta quan sát được hai bản tin (được tạo ra theo cùng một khoá và các trạng thái nguồn khác nhau )thì số các khoá có thể giảm xuống còn $\lambda$.Lập luận tương tự như trên ta thấy rằng $H(K|M^2)=\log\lambda$.Khi đó:

$$H(K|M)-H(K)=\log\lambda-\log\lambda n$$
$$=-\log n=-Pd_1$$

Như vậy giới hạn này được thoả mãn với dấu "=".

## 10.5.CÁC CHÚ GIẢI VÀ TÀI LIỆU DẪN

Các mã xác thực được phát minh vào năm 1974 bởi Gilbert.Mac-Williams và Sloane [GMS 74ư.Nhiếu phần lí thuyết về các mã xác thực đã được Simones phát triển,ông đã chứng minh nhiều kết quả cơ bản trong lĩnh vực này.Hai bài tổng quan hữa ích của Simones là [Si92] và [Si88].Massey cũng trình bày một tổng quan khá hay khác trong [Ma86].Các mối liên hệ giữa các mảng trực giao và các mã xác thực đã là mối quan tâm của nhiều nhà nghiên cứu..Cách trình bày ở đây dựa vào ba bài báo của Stinson[St 88],[St 90]và [St 92].Các mảng trực giao đã được nghiên cứu trong hơn 45 năm bởi các nhà nghiên cứu trong lĩnh vực thống kê và trong lí thuyết thiết kế tổ hợp.Ví dụ,giới hạn trong định lí 10.9 lần đầu tiên được chứng minh bởi Placket và Berman vào 1945 trong [PB 45].Nhiều kết quả thú vị về các mảng trực giao có thể tìm được trong nhiều giáo trình khác nhau về lí thuyết thiết kế tổ hợp(chẳng hạn như trong [BJL 8] của Beth,Jungickel và Lenz).

Cuối cùng việc sử dụng kĩ thuật entropy trong việc nghiên cứu các mã xác thực do Simone đưa ra .Giới hạn của định lí 10.13 đã được Simone chứng minh trước tiên trong [Si 85];một cánh chứng minh của định lí 10.14 có thể tìm được trong [Wa 90] của Walker.

## BÀI TẬP

10.1.Hãy tính $Pd_0$ và $Pd_1$ của mã xác thực được biểu thị trong ma trận sau :

| Khoá | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 1 | 1 | 2 | 3 |
| 2 | 1 | 2 | 3 | 1 |
| 3 | 2 | 1 | 3 | 1 |
| 4 | 2 | 3 | 1 | 2 |
| 5 | 3 | 2 | 1 | 3 |
| 6 | 3 | 3 | 2 | 1 |

Các phân bố xác suất trên S và K như sau:

$$P_s(1)=p_s(4)=1/6\ ,p_s(2)=p_s(3)=1/3$$

$$p_K(1)=p_K(6)=1/4,\ p_K(2)=p_K(3)=p_K(4)=p_K(5)=1/8.$$

Nêu các chiến lược thay thế và giả mạo tối ưu .

10.2.Ta đã biết cấu trúc đối với một mảng trực giao 0A(p,p,1)khi p là số nguyên tố.Hãy chứng tỏ rằng luôn có thể mở rộng 0A(p,p,1)thêm một cột nữa để tạo thành 0A(p,p+1,1).Hãy minh hạo cấu trúc của bạn trong trường hợp p=5.

10.3.Giả sử A là một cấu trúc $0A(n_1,k,\lambda_1)$ trên tập kí hiệu $\{1,...,n_1\}$ và giả sử B là một $0A(n_2,k,\lambda_2)$ trên tập kí hiệu $\{1,...,n_2\}$Ta xây dựng C là một $0A(n_1,n_2,k,\lambda_1\lambda_2)$ trên tập kí hiệu $\{1...n_1\}x\{1...n_2\}$ như sau :với mỗi hàng $r_1=(x_1...x_k)$ của A và với mỗi hàng $s_1=\{y_1...y_k\}$ của B ta xác định một hàng $t_1$ của C là:

$$t_1=((x_1,y_1),...,(x_k,y_k)).$$

Hãy chứng manh rằng C thực sự là một $0A(n_1n_2,k,\lambda_1\lambda_2)$.

10.4.Hãy xây dựng một mảng trực giao 0A(3,13,3).

10.5Hãy viết một chương trình máy tính để tính H(K),H(K|M) và $H(K|M^2)$cho mã xác thực ở bài toán 10.1Phân bố xác suất trên cavcs dãy của hai nguồn là :

$$p_{S^2}(1.2) = p_{S^2}(1.3) = p_{S^2}(1.4) = 1/18$$

$$p_{S^2}(2.1) = p_{S^2}(2.3) = p_{S^2}(2.4) = 1/9$$

$$p_{S^2}(3.1) = p_{S^2}(3.2) = p_{S^2}(3.4) = 1/9$$

$$p_{S^2}(4.1) = p_{S^2}(4.2) = p_{S^2}(4.3) = 1/18$$

Hãy so sánh giới hạn entropy của $Pd_0$ và $Pd_1$ với các giá trị mà bạn tính được trong bài tập 10.1.

Chỉ dẫn:Để tính $p_K(k|m)$ hãy dùng công thức Bayes:

$$p_K(k \mid m) = \frac{p_M(m \mid k)\, p_K(k)}{p_M(m)}$$

Ta đã biết cách tính $p_M(m)$.Để tính $p_M(m \mid k)$ hãy viết m=(s,a) và nhận xét thấy rằng :$p_M(m \mid k)=p_S(s)$ nếu $e_K(s)=a$ và $p_M(m \mid k)=0$ trong trường hợp ngược lại .

# CHƯƠNG 9
# CÁC SƠ ĐỒ ĐỊNH DANH

## 9.1 GIỚI THIỆU.

Các kỹ thuật mật mã cho phép nhiều bài toán dường như không thể giải được thành có thể giải được. Một bài toán như vậy là bài toán xây dựng các sơ đồ định danh mật. Trong nhiều trường hợp cần thiết phải "chứng minh" bằng phương tiện điện tử danh tính của ai đó. Dưới đây là một số trường hợp điển hình:

1. Để rút tiền từ một máy thủ quỹ tự động (ATM), ta dùng thẻ cùng với số định danh cá nhân (PIN) có 4 chữ số.
2. Để trả tiền cho các cuộc mua bán trên điện thoại dùng thẻ tín dụng, tất cả đều cần số thẻ tín dụng (và thời hạn dùng thẻ)
3. Để trả tiền cho các cú gọi điện thoại đường dài (dùng thẻ gọi) chỉ cần số điện thoại và PIN 4 chữ số.
4. Để vào mạng máy tính, cần tên hợp lệ của người sử dụng và mật khẩu tương ứng.

Thực tế, các kiểu sơ đồ này thường không được thực hiện theo cách an toàn. Trong các giao thức thực hiện trên điện thoại, bất kì kẻ nghe trộm nào cũng có thể dùng thông tin định danh cho mục đích riêng của mình. Những người này cũng có thể là người nhận thông tin. Các mưu đồ xấu trên thẻ tín dụng đều hoạt động theo cách này. Thẻ ATM an toàn hơn một chút song vẫn còn những điểm yếu. Ví dụ, ai đó điều khiển đường dây liên lạc có thể nhận được tất cả các thông tin được mã hoá trên dải từ tính của thẻ cũng như thông tin về PIN. Điều này cho phép một kẻ mạo danh tiếp cận vào tài khoản nhà băng. Cuối cùng, việc chui vào mạng máy tính từ xa cũng là vấn đề nghiêm trọng do các ID và mật khẩu của người sử dụng được truyền trên mạng ở dạng không mã. Như vậy, họ là những vùng dễ bị tổn thương đối với những người điều khiển mạng máy tính.

Mục đích của sơ đồ định danh là: ai đó "nghe" như Alice tư xưng danh với Bob không thể tự bịa đặt mình là Alice. Ngoài ra, chúng ta sẽ cố gắng giảm xác suất để chính Bob có thể thử mạo nhận là Alice sau khi cô ta tự xưng danh với anh ta. Nói cách khác, Alice muốn có khả năng chứng minh danh tính của mình bằng phương tiện điện tử mà không cần đưa ra chút thông tin nào hết về danh tính của mình.

Một vài sơ đồ định danh như vậy đã được nêu ra. Một mục đích thực tế là tìm một sơ đồ đủ đơn giản để có thể thực hiện được trên thẻ thông minh, đặc biệt là thẻ tín dụng gắn thêm một chíp có khả năng thực hiện các tính toán số học. Vì thế, thẻ đòi hỏi cả khối lượng tính toán lẫn bộ nhớ nhỏ đến mức có thể. Thẻ như vậy an toàn hơn các thẻ ATM hiện tại. Tuy nhiên, điều quan trọng cần chú ý là sự an toàn "đặc biệt" liên quan đến người điều khiển

đường dây thông tin. Vì nó là thẻ để chứng minh danh tính nên không cần bảo vệ chống mất thẻ. Song nó vẫn cần thiết có PIN để biết ai là chủ nhân thực sự của thẻ.

Trong các phần sau sẽ mô tả một số sơ đồ định danh thông dụng nhất. Tuy nhiên, trước hết hãy xét một sơ đồ rất đơn giản dựa trên hệ thống mã khoá riêng bất kì, chẳng hạn như DES. Giao thức mô tả trên hình 9.1 được gọi là giao thức "yêu cầu và trả lời", trong đó giả thiết rằng, Alice đang tự xưng danh với Bob cô và Bob chia nhau một khoá mật chung K, khoá này chỉ là hàm mã $e_K$.

***Hình 9.1: Giao thức Yêu cầu và đáp ứng:***

1. Bob chọn một yêu cầu x- là một chuỗi ngẫu nhiên 64 bit. Bob gửi x cho Alice
2. Alice tính $y = e_K(x)$
   gửi nó cho Bob.
3. Bob tính:
   $y' = e_K(x)$
   và xác minh $y' = y$.

Ta sẽ minh hoạ giao thức này bằng ví dụ nhỏ dưới dây.

*Ví dụ 9.1*

Giả sử Alice và Bob dùng hàm mã làm luỹ thừa tính modulo:

$$e_K(x) = x^{102379} \bmod 167653.$$

Giả sử yêu cầu của Bob x = 77835. Khi đó Alice sẽ trả lời với y = 100369.

Mọi sơ đồ định danh thực sự đều là các giao thức "Yêu cầu và đáp ứng" song các sơ đồ hiệu quả nhất lại không yêu cầu các khoá chia sẻ (dùng chung). ý tưởng này sẽ được tiếp tục trong phần còn lại của chương này.

## 9.2 Sơ đồ định danh Schnorr.

Ta bắt đầu bằng việc mô tả sơ đồ định danh Schnorr - là một trong những sơ đồ định danh thực tiễn và đáng chú ý nhất. Sơ đồ này đòi hỏi một người được uỷ quyền có tín nhiệm mà ta ký hiệu là TA. Ta sẽ chọn các tham số cho sơ đồ như sau:

1. p là số nguyên tố lớn (tức $p \geq 2^{512}$) sao cho bài toán logarithm rời rạc trong Zp là không giải được.
2. q là ước nguyên tố lớn của p-1 (tức $q \geq 2^{140}$).
3. $\alpha \in Z_p^*$ có bậc q (có thể tính $\alpha$ như (p-1) ?? ) đều được công khai.

TA sẽ đóng một dấu xác nhận cho Alice. Khi Alice muốn nhận được một dấu xác thực từ TA, cô phải tiến hành các bước như trên hình 9.2. Vào

thời điểm cuối, khi Alice muốn chứng minh danh tính của cô trước Bob, cô thực hiện giao thức như trên hình 9.3.

Như đã nêu ở trên, t là một tham số mật. Mục đích của nó là ngăn kẻ mạo danh - chẳng hạn Olga - khỏi phỏng đoán yêu cầu r của Bob. Ví dụ, nếu Olga đoán đúng giá trị r, cô ta có thể chọn giá trị bất kỳ cho y và tính

$$\gamma = \alpha^y v^\gamma \bmod p$$

Cô sẽ đưa cho Bob $\gamma$ như trong bước 1 và sau đó khi nhận được yêu cầu r, cô sẽ cung cấp giá trị y đã chọn sẵn. Khi đó $\gamma$ sẽ được Bob xác minh như trong bước 6.

***Hình 9.2 Cấp dấu xác nhận cho Alice.***

1. TA thiết lập danh tính của Alice bằng cách lập giấy chứng minh thông thường chẳng hạn như xác nhận ngày sinh, hộ chiếu ... Sau đó TA thiết lập một chuỗi ID (Alice) chứa các thông tin định danh của cô ta.
2. Alice bí mật chọn một số mũ ngẫu nhiên a, $0 \le a \le q-1$. Alice tính:
   $v = \alpha^{-a} \bmod p$
   và gửi v cho TA
3. TA tạo ra một chữ kí:
   $s = sig_{TA}(I,v)$.
   Dấu xác nhận
   $C(Alice) = (ID(Alice),v,s)$
   và đưa cho Alice

Xác suất để Olga phỏng đoán đúng r là $2^{-t}$ nếu r được Bob chọn ngẫu nhiên. Như vậy, t = 40 là giá trị hợp lý với hầu hết các ứng dụng, (tuy nhiên, chú ý rằng, Bob sẽ chọn r ngẫu nhiên mỗi lần Alice xưng danh với anh ta. Nếu Bob luôn dùng cùng một r thì Olga có thể mạo danh Alice bằng phương pháp mô tả ở trên).

Có hai vấn đề nảy sinh trong giao thức xác minh. Trước hết, chữ kí s chứng minh tính hợp lệ của dấu xác nhân của Alice. Như vậy, Bob xác minh chữ ký của TA trên dấu xác nhận của Alice để thuyết phục chính bản thân mình rằng dấu xác nhận là xác thực. Đây là xác nhận tương tự như cách đã dùng ở chương 8.

Vấn đề thứ hai của giao thức liên quan đến mã số mật a. Giá trị a có chức năng tương tự như PIN để thuyết phục Bob rằng, người thực hiện giao thức định danh quả thực là Alice. Tuy nhiên có một khác nhau quan trọng so với PIN là: trong giao thức định danh, a không bị lộ. Thay vào đó, Alice (hay chính xác hơn là thẻ thông minh của cô) chứng minh rằng, cô (thẻ) biết giá trị a trong bước 5 bằng cách tính y trong khi trả lời đòi hỏi r do Bob đưa ra. Vì a không bị lộ nên kĩ thuật này gọi là chứng minh không tiết lộ thông tin.

***Hình 9.3. sơ đồ định danh Schnorr***

1. Alice chọn một số ngẫu nhiên k, $0 \le k \le q-1$ và tính:
   $\gamma = \alpha^k \bmod p$.

2. Alice gửi dấu xác nhận của mình cho C(Alice) = (ID(Alice),v,s) và $\gamma$ cho Bob.
3. Bob xác minh chữ kí của TA bằng cách kiểm tra xem có thoả mãn
   *ver(*ID(Alice),v,s) = true hay không.
4. Bob chọn một số ngẫu nhiên r, $1 \le r \le 2^t$ và đưa nó cho Alice.
5. Alice tính:
   $y = k + ar \bmod q$
   và đưa y cho Bob.
6. Bob xác minh xem có thoả mãn đồng dư thức sau không
   $\gamma \equiv \alpha^y v^r \pmod p$.

Các đồng dư sau đây chứng minh rằng Alice có khả năng chứng minh danh tính của cô cho Bob:

$$\alpha^y v^r \equiv \alpha^{k+ar} v^r \pmod p$$
$$\equiv \alpha^{k+ar} v^{ar} \pmod p$$
$$\equiv \alpha^k \pmod p$$
$$\equiv \gamma \pmod p$$

Như vậy sẽ chấp nhận bằng chứng về danh tính của Alice và giao thức được gọi là có tính đầy đủ.

Dưới đây là một ví dụ nhỏ minh hoạ khía cạnh "thách thức và đáp ứng" của giao thức.

*Ví dụ 9.2*

Giả sử p=88667, q = 1031, t=10. Phần tử $\alpha$ = 70322 có bậc q thuộc $Z_p^*$. Giả sử số mã mật của Alice a = 755. Khi đó:

$$v = \alpha^{-a} \pmod p$$
$$= 70322^{1031-755} \bmod 88667$$
$$= 13136$$

Giả sử Alice chọn k = 543, sau đó cô tính:

$$\gamma = \alpha^k \bmod p$$
$$= 70322^{543} \bmod 88667$$
$$= 84109$$

và gửi $\gamma$ cho Bob. Giả thiết Bob đưa ra yêu cầu r = 1000. Khi đó Alice tính:

$$y = k + ar \bmod q$$
$$= 543 + 755 \times 1000 \bmod 1031$$
$$= 851$$

và gửi y cho Bob. Sau đó Bob xác minh xem

$$84109 \equiv 70322^{851} 13136^{1000} \pmod{88667}$$

Nếu đúng, Bob sẽ tin rằng anh ta đang liên lạc với Alice.

Tiếp theo ta hãy xem xét cách ai đó có thể mạo danh Alice. Olga - kẻ đang cố mạo danh Alice bằng cách làm giả dấu xác nhận:

C'(Alice) = (ID(Alice), v', s'),

trong đó v'≠v. Song s' được giả thiết là chữ kí của (ID(Alice), v', s') và nó được Bob xác minh trong bước 3 của giao thức. Nếu sơ đồ chữ kí của TA là an toàn, Olga sẽ không thể làm giả chữ kí s' (mà sau này sẽ bị Bob xác minh).

Biện pháp khác sẽ cho Olga dùng dấu xác nhận đúng của Alice C(Alice) = (ID(Alice), v, s) (nhớ lại rằng, các dấu xác nhận không mật và thông tin trên dấu xác nhận bị lộ mỗi lần thực hiện giao thức định danh). Tuy nhiên Olga sẽ không thể mạo danh Alice trừ phi cô ta cũng biết giá trị a. Đó là vì "yêu cầu" r trong bước 4. ở bước 5, Olga sẽ phải tính y mà y là hàm của a. Việc tính a từ v bao hàm việc giải bài toán logarithm rời rạc là bài toán mà ta đã giả thiết là không thể giải được.

Có thể chứng minh một định lí chính xác hơn về tính an toàn của giao thức như sau:

***Định lí 9.1.***

*Giả sử Olga biết giá trị γ nhờ đó cô có xác suất $\varepsilon \geq 1/2^{t-1}$ để giả mạo Alice thành công trong giao thức xác minh. Khi đó Olga có thể tính a trong thời gian đa thức.*

*Chứng minh*

Với một phần $\varepsilon$ trên $2^t$ yêu cầu r, Olga có thể tính giá trị y (sẽ được Bob chấp nhận trong bước 6). Vì $\varepsilon \geq 1/2^{t-1}$ nên ta có $2^t/\varepsilon \geq 2$ và bởi vậy, Olga có thể tính được các giá trị $y_1, y_2, r_1$ và $r_2$ sao cho

$$y_1 \neq y_2$$

và $\gamma \equiv \alpha^{y_1} v^{\hat{I}_1} \equiv \alpha^{y_2} v^{\hat{I}_2} \pmod p$

hay $\alpha^{y_1 - y_2} \equiv v^{r_1 - r_2} \pmod p$

Vì $v = \alpha^{-a}$ nên ta có:

$$y_1 - y_2 \equiv a(r_1 - r_2) \pmod q$$

Xét thấy $0 < | r_1 - r_2 | < 2^t$ và $q > 2^t$ là nguyên tố. Vì UCLN$(r_1 - r_2, q) = 1$ và Olga có thể tính:

$$a = (y_1 - y_2)(r_1 - r_2)^{-1} \bmod q$$

như mong muốn…

Định lý trên chứng minh rằng, bất kỳ ai có cơ hội (không phải không đáng kể) thực hiện thành công giao thức định danh đều phải biết (hoặc có thể tính trong thời gian đa thức) số mũ mật a của Alice. Tính chất này thường được gọi là tính đúng đắn (sound). Dưới đây là ví dụ minh hoạ:

*Ví dụ 9.3*

Giả sử ta cũng có các tham số như trong ví dụ 9.2: p = 88667, q = 1031, t= 10, $\alpha$ = 70322, a = 755 và v = 13136. Giả sử Olga nghiên cứu thấy rằng:

$\alpha^{851}v^{1000} \equiv \alpha^{454}v^{19} (\text{mod } p)$.

khi đó có thể tính:

$a = (851 - 454)(1000 - 19)^{-1} \text{ mod } 1031 = 755$

và như vậy sẽ khám phá ra số mũ mật của Alice. …

Chúng ta đã chứng minh rằng, giao thức có tính đúng đắn và đầy đủ. Song tính đúng đắn và đầy đủ chưa đủ để bảo đảm rằng giao thức là an toàn. Chẳng hạn, nếu Alice để lộ số mũ mật a của mình khi chứng minh danh tính của cô với Olga thì giao thức vẫn còn đúng đắn và đầy đủ. Tuy nhiên nó sẽ hoàn toàn không an toàn vì sau đó Olga có thể mạo danh Alice.

Điều này thúc đẩy động cơ xem xét thông tin mật đã cho người xác minh - người cũng tham gia trong giao thức - biết (trong giao thức này, thông mật là a). Hy vọng là không có thông tin nào về a có thể bị gia tăng bởi Olga khi Alice chứng minh danh tính của mình cho cô ta, để sau đó Olga có thể giả dạng như Alice.

Nói chung, có thể hình dung tình huống khi Alice chứng minh danh tính của mình với Olga trong một số tình huống khác nhau. Có lẽ Olga không chọn các yêu cầu của cô (tức các giá trị r) theo kiểu ngẫu nhiên. Sau vài lần thực hiện giao thức, Olga sẽ cố gắng xác định giá trị a để sau đó có thể mạo danh Alice. Nếu Olga không thể xác định được chút thông tin nào về a qua tham gia với số lần đa thức thực hiện giao thức và sau đó thực hiện một lượng tính toán đa thức thì giao thức có thể được gọi là an toàn.

Hiện tại vẫn chưa chứng minh được rằng giao thưc Schnorr là an toàn, song trong phần tiếp sau, ta sẽ đưa ra một cải tiến về sơ đồ này (do Okmoto đưa ra) mà có thể chứng minh được nó là an toàn khi cho trước giả thuyết tính toán nào đó.

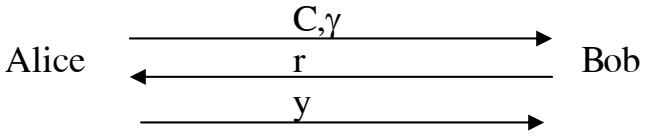
Sơ đồ Schnorr đã được thiết kế với tốc độ nhanh và hiệu quả theo quan điểm cả về tính toán lẫn lượng thông tin cần thiết để trao đổi trong giao thức. Nó cũng được thiết kế nhằm tối thiểu hoá lượng tính toán mà Alice phải thực hiện. Đây là những đặc tính tốt vì trong thực tế, các tính toán của Alice sẽ phải tính trên các thẻ thông minh có khả năng tính toán thấp trong khi các tính toán của Bob lại trên các máy lớn.

Vì mục đích thảo luận, ta hãy giả sử rằng, ID(Alice) là chuỗi 512 bit, v cũng gồm 512 bit, còn s bằng 320 bit nến DSS được dùng như sơ đồ chữ kí. Kích thước tổng cộng của dấu xác nhận C(Alice) (cần được lưu trên thẻ của Alice) là 1444 bit.

Xét các tính toán của Alice: Bước 1 cần lấy mũ theo modulo, bước 5 so sánh một phép công modulo và một phép nhân modulo. Đó là phép luỹ

thừa modulo mạnh về tính toán song có thể tính toán gián tiếp nếu muốn. Còn các tính toán trực tiếp được Alice thực hiện bình thường.

Việc tính số bit cần thiết trong quá trình liên lạc để thực hiện giao thức cũng khá đơn giản. Có thể mô tả thông tin được liên lạc ở dạng đồ hình như sau




Alice đưa cho Bob 1444 + 512 = 1956 bit thông tin trong bước 2: Bob đưa cho Alice 40 bit trong bước 4 và Alice đưa cho Bob 160 bit trong bước 6. Như vậy các yêu cầu về liên lạc rất mức độ.

## 9.3 Sơ đồ định danh của Okamoto.

Trong phần này ta sẽ đưa ra một biến thể của sơ đồ Schnorr do Okamoto đưa ra. Sơ đồ cải tiến này Zp không giải được.

Để thiết lập sơ đồ, TA chọn p và q như trong sơ đồ Schnorr. TA cũng chọn hai phần tử $\alpha_1$ và $\alpha_2 \in Z_p^*$ đều có bậc q. Giá trị $c = \log_{\alpha 1}\alpha_2$ được giữ bí mật kể cả đối với Alice. Ta sẽ giả thiết rằng, không ai có thể giải được (thậm chí Alice và Olga liên minh với nhau) để tính ra giá trị c. Như trước đây, TA chọn sơ đồ chữ kí số và hàm hash. Dấu xác nhận mà TA đã phát cho Alice được xây dựng như mô tả ở hình 9.4.

Dưới đây là một ví dụ về sơ đồ Okamoto.

*Ví dụ 9.4.*

Cũng như ví dụ trước, ta lấy p = 88667, q = 1031, t = 10. Cho $\alpha_1$ = 58902 và cho $\alpha_2$ = 73611 (cả $\alpha_1$ lẫn $\alpha_2$ đều có bậc q trong $Z_p^*$). Giả sử $a_1$=846, $a_2$ = 515, khi đó v = 13078.

Giả sử Alice chọn $k_1$ = 899, $k_2$ = 16, khi đó γ = 14573. Nếu Bob đưa ra yêu cầu r = 489 thì Alice sẽ trả lời $y_1$ = 131 và $y_2$ = 287. Bob sẽ xác minh thấy:

$58902^{131}78611^{287}1378^{489} \equiv 14574 \pmod{88667}$.

Vì thế Bob chấp nhận bằng chứng của Alice về danh tính của cô. …

Việc chứng minh giao thức là đầy đủ không khó (tức là Bob sẽ chấp nhận bằng chứng về danh tính của cô). Sự khác nhau giữa sơ đồ của Okamoto và Schnorr là ở chỗ, ta có thể chứng minh rằng sơ đồ Okamota an toàn miễn là bài toán logarithm rời rác không giải được.

**Hình 9.4: Đóng dấu xác nhận cho Alice.**

1. TA thiết lập danh tính của Alice và phát chuỗi định danh ID(Alice).
2. Alice chọn bí mật hai số mũ ngẫu nhiên $a_1$,$a_2$ trong đó $0 \le a_1,a_2 \le q - 1$ Alice tính:

$$v = \alpha_1^{-a_1}\alpha_1^{-a_2} \bmod p$$

và đưa cho TA.

3. TA tạo chữ kí

$$s = sig_{TA}(I,v).$$

và đưa dấu xác nhận

$$C(Alice) = (ID(Alice),v,s)$$

cho Alice

Phép chứng minh về tính an toàn rất tinh tế. Đây là ý kiến chung: Như trước đây, Alice tự định danh với Olga trong nhiều thời gian đa thức thông qua thực hiện giao thức. Khi đó ta giả thiết rằng Olga có khả năng nghiên cứu một số thông tin về các giá trị $a_1,a_2$. Nếu như vậy thì Olga và Alice kết hợp với nhau có khả năng tính được logarithm rời rạc c trong thời gian đa thức. Điều này mâu thuẫn với giả định ở trên và chứng tỏ rằng Olga chắc không thể nhận được chút thông tin nào về các số mũ của Alice thông qua việc tham gia vào giao thức.

Phần đầu tiên của giao thức này tương tự với chứng minh tính đầy đủ trong sơ đồ Schnorr.

*Định lý 9.2.*

*Giả sử Olga biết a giá trị $\gamma$ mà nhờ nó cô có xác suất thành công $\varepsilon \geq 1/2^{t-1}$ khi đánh giá Alice trong giao thức xác minh. Khi đó, Olga có thể tính các giá trị $b_1,b_2$ trong thời gian đa thức sao cho*

$$v \equiv \alpha_1^{-b_1}\alpha_1^{-b_2} \bmod p .$$

*Chứng minh:*

Với phần $\varepsilon$ trên $2^t$ yêu cầu có thể r, Olga có thể tính các giá trị $y_1$, $y_2$, $z_1$, $z_2$, r và s với $r \neq s$ và:

$$\gamma \equiv \alpha_1^{y_1}\alpha_2^{y_2} v^r \equiv \alpha_1^{z_1}\alpha_2^{Z_2} v^8 (\text{mod } p).$$

Ta định nghĩa: $b_1 = (y_1 - z_1)(r - s)^{-t} \bmod q$

và $b_1 = (y_2 - z_2)(r - s)^{-t} \bmod q$

Khi đó dễ dàng kiểm tra thấy rằng:

$$v \equiv \alpha_1^{-b_1}\alpha_2^{-b_2} (\text{mod } p)$$

như mong muốn….

***Hình 9.5. Sơ đồ định danh Okamoto.***

1. Alice chọn các số ngẫu nhiên $k_1$, $k_2$, $0 \le k_1, k_2 \le q-1$ và tính:
$$\gamma = \alpha_1^{k_1}\alpha_2^{k_2} \pmod p.$$
2. Alice gửi dấu xác nhận của cô C(Alice) = (ID(Alice),v,s) và γ cho Bob.
3. Bob xác minh chữ kí của TA bằng cách kiểm tra xem có thoả mãn đồng nhất thức:
$$ver_{TA}(ID(Alice),v,s) = true$$
4. Bob chọn số ngẫu nhiên r, $1 \le r \le 2^t$ và đưa nó cho Alice.
5. Alice tính:
$$y_1 = k_1 + a_1 r \bmod q$$
và $$y_2 = k_2 + a_2 r \bmod q$$
và đưa $y_1, y_2$ cho Bob.
6. Bob xác minh xem:
$$\gamma \equiv \alpha_1^{y_1}\alpha_2^{y_2} v^r \pmod p$$ hay không.

Bây giờ ta tiếp tục chỉ ra cách Alice và Olga cùng tính giá trị c.

***Định lý 9.3.***

*Giả sử Olga biết giá trị γ (mà với nó cô có xác suất giả danh Alice thành công là $\varepsilon \ge 1/2^{t-1}$) trong giao thức xác minh. Khi đó, Alice và Olga có thể cùng nhau tính* $\log_{\alpha_1}\alpha_2$ *trong thời gian đa thức với xác suất 1-1/q.*

*Chứng minh*

Theo định lý 9.2, Olga có khả năng xác định các giá trị $b_1$ và $b_2$ sao cho
$$v \equiv \alpha_1^{b_1}\alpha_2^{b_2} \pmod p$$
Giả thiết rằng Alice để lộ các giá trị $a_1$ và $a_2$ cho Olga biết. Dĩ nhiên:
$$v \equiv \alpha_1^{a_1}\alpha_2^{a_2} \pmod p$$
vì thế $$\alpha_1^{a_1-b_1} \equiv \alpha_2^{b_2-a_2} \pmod p$$
giả sử rằng (a1,a2) ≠ (b1,b2), khi đó $(a1-b1)^{-1}$ tồn tại và logarithm rời rạc:
$$c = \log_{\alpha_1}\alpha_2 = (a_1-b_1)(b_2-a_2)^{-1} \bmod q$$
có thể tính được trong thời gian đa thức.

Phần còn lại là xem xét xác suất để (a1,a2) = (b1,b2). Nếu xảy ra điều này thì giá trị c không thể tính theo mô tả ở trên. Tuy nhiên, ta sẽ chỉ ra rằng (a1,a2) = (b1,b2) sẽ chỉ xảy ra với xác suất rất bé 1/q, vì thế giao thức nhờ đó Alice và Olga tính được c sẽ hầu như chắc chắn thành công.

Định nghĩa:
$$A = \{ (a_1', a_2') \in Z_p \times Z_q : \alpha_1^{-a_1'}\alpha_2^{-a_2'} \equiv \alpha_1^{-a_1'}\alpha_2^{-a_2'} \pmod p \}$$
Nghĩa là A gồm tất cả các cặp được sắp có thể và chúng có thể là các số mũ mật của Alice. Xét thấy rằng:
$$A = \{a_1 - c\theta, a_2 + \theta : \theta \in Z_P\},$$

Trong đó c = $\log_{\alpha_1} \alpha_2$. Như vậy A chứa q cặp được sắp.

Cặp được sắp (b1,b2) do Olga tính chắc chắn ở trong tập A. Ta sẽ chỉ ra rằng, giá trị của cặp (b1,b2) độc lập với cặp (a1,a2) chứa các số mũ mật của Alice. Vì (a1,a2) được Alice chọn đầu tiên một cách ngẫu nhiên nên xác suất để (a1,a2) = (b1,b2) là 1/q.

Như vậy, (b1,b2) là "độc lập" với (a1,a2). Cặp (a1,a2) của Alice là một trong q cặp được sắp có thể trong A và không có thông tin nào về nó (là cặp "đúng") đã bị Alice để lộ cho Olga biết khi cô xưng danh với Olga. (Một cách hình thức, Olga biết một cặp trong A chứa số mũ của Alice song cô ta không biết nó là cặp nào).

Ta hãy xét thông tin được trao đổi trong giao thức định danh. Về cơ bản, trong mỗi lần thực hiện giao thức, Alice chọn γ, Olga chọn r và Alice để lộ $y_1$ và $y_2$ sao cho:

$$\gamma = \alpha_1^{y_1} \alpha_1^{y_2} v^r \pmod p.$$

Ta nhớ lại rằng, Alice tính:

$$y_1 = k_1 + a_1 r \bmod q$$

và $$y_2 = k_2 + a_2 r \bmod q$$

trong đó

$$\gamma = \alpha_1^{k_1} \alpha_1^{k_2} \bmod q$$

Chú ý rằng $k_1$ và $k_2$ không bị lộ (mà $a_1$ và $a_2$ cũng không).

Bốn phần tử cụ thể $(\gamma,r,y_1,y_2)$ được tạo ra trong thực hiện giao thức tuỳ thuộc vào cặp $(a_1,a_2)$ của Alice vì $y_1$ và $y_2$ được định nghĩa theo $a_1$ và $a_2$. Tuy nhiên ta sẽ chỉ ra rằng, mỗi bộ bốn như vậy có thể được tạo ra như nhau từ cặp được sắp bất kì khác $(a'_1, a'_2) \in A$. Để thấy rõ, giả thiết $(a'_1, a'_2) \in A$, tức là $a'_1 = a_1 - c\theta$ và $a'_2 = a_2 + \theta$, trong đó $0 \le \theta \le q - 1$.

Có thể biểu diễn $y_1$ và $y_2$ như sau:

$$\begin{aligned} y_1 &= k_1 + a_1 r \\ &= k_1 + (a'_1 + c\theta) r \\ &= (k_1 + rc\theta) + a'_1 r \end{aligned}$$

và

$$\begin{aligned} y_2 &= k_2 + a_2 r \\ &= k_2 + (a'_2 - \theta) r \\ &= (k_2 - r\theta) + a'_2 r \end{aligned}$$

Trong đó tất cả các phép tính số học đều thực hiện trong $Z_p$. Nghĩa là bộ bốn $(\gamma,r,y_1,y_2)$ cũng phù hợp với cặp được sắp $(a'_1, a'_2)$ bằng việc dùng các phép chọn ngẫu nhiên $k'_1 = k_1 + rc\theta$ và $k'_2 = r\theta$ để tạo ra γ. Cần chú ý rằng, các giá trị $k_1$ và $k_2$ không bị Alice làm lộ nên bộ $(\gamma, r, y_1, y_2)$ không cho biết thông tin gì về cặp nào trong A được Alice dùng làm số mũ mật của cô. Đây là điều phải chứng minh.…

Việc chứng minh tính an toàn này khá tinh vi và tối ưu. Chắc nó sẽ hữu dụng để lắp mới các đặc điểm của giao thức, dẫn tới bằng chứng về sự an toàn. Như vậy, Alice chọn 2 số mũ mật cao hơn là chọn một. Có tổng cộng q cặp trong A tương đương với cặp $(a_1,a_2)$ của Alice. Điều này dẫn đến mâu thuẫn cơ bản là, việc hiểu biết hai cặp khác nhau trong A sẽ cho một phương pháp hiệu quả tính toán logarithm rời rạc c. Alice dĩ nhiên chỉ biết một cặp trong A; nếu ta chứng minh rằng Olga có thể giả danh Alice thì Olga có thể tính một cặp trong A khác với cặp của Alice (với xác suất cao). Như vậy Alice và Olga có thể cùng nhau tìm hai cặp trong A và tính c - cho mâu thuẫn như mong muốn.
Dưới đây là một ví dụ nhỏ minh hoạ việc Alice và Olga tính toán $\log_{\alpha_1} \alpha_2$:

*Ví dụ 9.5*.

Giống như trong ví dụ 9.4, ta lấy p =88667, q = 1031, t = 10 và giả sử v = 13078.
Giả thiết Olga đã xác định được rằng:

$$\alpha_1^{131}\alpha_2^{287}v^{489} \equiv \alpha_1^{890}\alpha_2^{303}v^{199} \pmod p$$

Khi đó cô tính:

$$b_1 = (131 - 890)(489 - 199)^{-1} \bmod 1031 = 456$$

và

$$b_2 = (287 - 303)(489 - 199)^{-1} \bmod 1031 = 519$$

Dùng các giá trị $a_1$ và $a_2$ do Alice đưa cho, giá trị c tính như sau:

$$c = (846 - 456)(519 - 515)^{-1} \bmod 1031 = 613$$

giá trị thực tế này là $\log_{\alpha_1} \alpha_2$ mà có thể xác minh bằng cách tính:

$$58902^{613} \bmod 88667 = 73611.$$

Cuối cùng, cần nhấn mạnh rằng, mặc dù không có chứng minh đã biết nào chứng tỏ sơ đồ Schnorr an toàn (thậm chí giả thiết rằng, bài toán logarithm rời rạc không giải được) song ta cũng không biết bất kì nhược điểm nào của sơ đồ này. Thực sự sơ đồ Schnorr được ưa thích hơn sơ đồ Okamoto do nó nhanh hơn.

## 9.4 Sơ đồ định danh Guillou - quisquater.

Trong phần này sẽ mô tả một sơ đồ định danh khác do Guillou và Quisquater đưa ra dựa trên RSA.

Việc thiết lập sơ đồ như sau: TA chọn 2 số nguyên tố p và q và lập tích n =pq. Giá trị của p và q được giữ bí mật trong khi n công khai. Giống như trước đây, p và q nên chọn đủ lớn để việc phân tích n không thể thực hiện được. Cũng như vậy, TA chọn số nguyên tố đủ lớn b giữ chức năng tham số mật như số mũ mật trong RSA. Giả thiết b là số nguyên tố dài 40 bít. Cuối cùng TA chọn sơ đồ chữ kí và hàm hash.

***Hình 9.6: Phát dấu xác nhận cho Alice***

1. TA thiết lập định danh cho Alice và phát chuỗi định danh ID(Alice).

2. Alice chọn bí mật một số nguyên u, trong đó $0 \le u \le n-1$. Alice tính:
$$v = (u^{-1})^b \bmod n$$
và đưa u cho TA
3. TA tạo ra chữ kí:
$$s = sig_{TA}(I,v)$$
Dấu xác nhận:
$$C(Alice) = (ID(Alice), v, s)$$
và đưa cho Alice

Dấu xác nhận do TA phát cho Alice được xây dựng như mô tả trong hình 9.6. Khi Alice muốn chứng minh danh tính của cô cho Bob, cô thực hiện giao thức hình 9.7. Ta sẽ chứng minh rằng, sơ đồ Guillou - Quisquater là đúng đắn và đầy đủ. Tuy nhiên, sơ đồ không được chứng minh là an toàn (mặc dù giả thiết hệ thống mã RSA là an toàn).

*Ví dụ 9.6:*

Giả sử TA chọn p = 467, q = 479, vì thế n = 223693. Giả sử b = 503 và số nguyên mật của Alice u = 101576. Khi đó cô tính:
$$\begin{aligned} v &= (u^{-1})^b \bmod n \\ &= (101576^{-1})^{503} \bmod 223693 \\ &= 24412. \end{aligned}$$

***Hình 9.7: Sơ đồ định danh Guillou - Quisquater.***

1. Alice chọn số ngẫu nhiên k, trong đó $0 \le k \le n-1$ và tính:
$$\gamma = k^b \bmod n$$
2. Alice đưa cho Bob dấu xác nhận của cô C(Alice) = (ID(Alice), v, s) và γ.
3. Bob xác minh chữ kí của TA bằng cách kiểm tra xem có thoả mãn hay không đồng dư thức:
$$ver(ID(Alice), v, s) = true.$$
4. Bob chọn số ngẫu nhiên r, $0 \le r \le b-1$ và đưa nó cho Alice.
5. Alice tính:
$$y = k\, u^r \bmod n$$
và đưa y cho Bob
6. Bob xác minh rằng
$$\gamma \equiv v^r y^b \pmod n$$

Giả sử Bob trả lời bằng yêu cầu r = 375. Khi đó Alice sẽ tính
$$\begin{aligned} y &= k u^r \bmod n \\ &= 187485 \times 101576^{375} \bmod 223693 \\ &= 93725 \end{aligned}$$
và đưa nó cho Bob. Bob xác minh thấy:
$$24412 \equiv 89888^{375} 93725^{503} \pmod{223693}$$
vì thế Bob chấp nhận bằng chứng về danh tính của Alice. …

Giống như trường hợp tổng quát, việc chứng minh tính đầy đủ rất đơn giản:

$v^r y^b \equiv (u^{-b})^r (ku^r)^b \pmod n$

$\equiv u^{-br} k^b u^{br} \pmod n$

$\equiv k^b \pmod n$

$\equiv \gamma \pmod n$

Bây giờ ta xét đến tính đúng đắn. Ta sẽ chứng minh sơ đồ là đúng đắn miễn là không dễ dàng tính được u từ v. Vì v được lập từ u bằng phép mã RSA nên đây là giả thiết có vẻ hợp lý.

***Định lí 9.4***

*Giảsử Olga biết giá trị γ nhờ nó cô có xác suất thành công trong việc giả danh Alice là $\varepsilon > 1/b$ trong giao thức xác minh. Khi đó Olga có thể tính u trong thời gian đa thức.*

*Chứng minh*

Với γ nào đó, Olga có thể tính giá trị $y_1$, $y_2$, $r_1$, $r_2$ với $r_1 \neq r_2$ sao cho:

$$\gamma \equiv v^{r_1} y^b \equiv v^{r_2} y_2^b \pmod n$$

không mất tính tổng quát, giả sử rằng $r_1 > r_2$. Khi đó ta có:

$$v^{r_1 - r_2} \equiv (y_2 / y_1)^b \pmod n$$

vì 0 < r1- r2 <b và b là số nguyên tố nên $t = (r_1 - r_2)^{-1} \bmod b$ tồn tại và Olga có thể tính nó trong thời gian đa thức bằng thuật toán Euclidean. Vì thế ta có:

$$v^{(r_1 - r_2)t} \equiv (y_2 / y_1)^{bt} \pmod n.$$

xét thấy, $(r_1 - r_2)t = lb + 1$

với số nguyên dương *l* nào đó, vì thế:

$$v^{lb+1} \equiv (y_2/y_1)^{bt} \pmod n$$

hay tương đương,

$$v \equiv (y_2/y_1)^{bt} (v^{-1})^{lb} \pmod n.$$

Nâng cả hai vế lên luỹ thừa $b^{-1} \bmod \phi(n)$ ta có:

$$u^{-1} \equiv (y_2/y_1)^t (v^{-1})^l \pmod n.$$

cuối cùng tính modulo đảo của cả hai vế của đồng dư thức này, ta nhận được công thức sau cho u:

$$u = (y_2/y_1)^t v^l \bmod n$$

Olga có thể dùng công thức này để tính u trong thời gian đa thức. …

*Ví dụ 9.7*

Giống như ví dụ trước, giả sử rằng n = 223963, b = 503, u = 101576 và v = 89888. Giả thiết Olga nghiên cứu thấy rằng:

$v^{401} 103386^b \equiv v^{375} 93725^b \pmod n$

Trước tiên cô tính:

$t = (r_1 - r_2)^{-1} \bmod b$

$= (401 - 375)^{-1} \bmod 503$

$= 445$

Tiếp theo cô tính:

$l = ((r_1 - r_2)t - 1)/b$

$= ((401 - 375)445 - 1)/503 = 23$

Cuối cùng cô có thể nhận được giá trị u mật như sau:

$u = (y_1/y_2)^t v^l \bmod n$

$= (103386/93725)^{445} 89888^{23} \bmod 233693$

$= 101576$

và như vậy, số mũ mật của Alice đã bị lộ. …

### *9.4.1Sơ đồ định danh dựa trên tính đồng nhất.*

Sơ đồ định danh Guillou - Quisquater có thể chuyển thành sơ đồ định danh dựa trên tính đồng nhất. Điều này có nghĩa là không cần các dấu xác nhận. Thay vào đó, TA tính giá trị của u như một hàm của chuỗi ID của Alice bằng cách dùng một hàm hash công khai h trong phạm vi $Z_n$ như chỉ ra trên hình 9.8. Giao thức định danh lúc này làm việc như mô tả trong hình 9.9. Giá trị v được tính từ chuỗi ID của Alice thông qua hàm hash công khai. Để tiến hành giao thức định danh, Alice cần biết giá trị u, giá trị này chỉ TA là có thể tính được (giả thiết hệ thống mã khoá công khai RSA là an toàn). Nếu Olga cố tự xưng mình là Alice cô sẽ không thành công nếu không biết u.

***Hình 9.8: Phát giá trị u cho Alice***

1. Thiết lập danh tính của Alice và phát chuỗi định danh ID(Alice):
2. TA tính

   $u = (h(ID(Alice)^{-1})^a \bmod n$

   và đưa u cho Alice

***Hình 9.9: Sơ đồ định danh dựa trên tính đồng nhất Guillou - Quisquater.***

1. Alice chọn một số ngẫu nhiên k, $0 \le k \le n - 1$ và tính:

   $\gamma = k^b \bmod n$
2. Alice đưa ID(Alice) và $\gamma$ cho Bob
3. Bob tính:

   $v = h(ID(Alice))$
4. Bob chọn số ngẫu nhiên r, $0 \le r \le b-1$ và đưa nó cho Alice.
5. Alice tính:

   $y = ku^r \bmod n$

   và đưa y cho Bob
6. Bob xác minh xem có thoả mãn hay không điều kiện dưới đây:

   $\gamma = v^r y^b (\bmod n)$

## 9.5 Chuyến sơ đồ định danh thành sơ đồ chữ kí.

Có một phương pháp chuẩn để chuyển sơ đồ định danh thành sơ đồ chữ kí. ý tưởng cơ bản là thay thế người xác minh (Bob) bằng hàm hash công khai h. Trong sơ đồ chữ kí thực hiện theo phương pháp này, thông báo không

bị chặt ra (băm) trước khi được kí: Quá trình băm được tích hợp thành thuật toán kí.

Sau đây sẽ minh hoạ biện pháp này bằng việc chuyển sơ đồ Schnorr thành sơ đồ chữ kí (hình 9.10). Thực tế, có khả năng đưa hàm hash h vào SHS và làm giảm được modulo q. Do SHS tạo ra xâu bit có độ dài 160 và q là số nguyên tố 160 bit, nên việc giảm được modulo q chỉ cần thiết khi bản tóm lược của thông báo do SHS tạo ra vượt quá q. Thậm chí trong trường hợp này, chỉ cần trừ q khỏi kết quả.

Trong quá trình chuyển từ sơ đồ định danh thành sơ đồ chữ kí, ta đã thay yêu cầu 40 bit bằng bản tóm lược thông báo 160 bit, 40 bit là đủ đối với một yêu cầu (challenge) vì kẻ mạo danh cần có khả năng phỏng đoán yêu cầu để tính trước câu trả lời (mà sẽ được chấp nhận). Song trong phạm vi của sơ đồ chữ kí, ta cần cac bản tóm lược thông báo có kích thước lớn hơn nhiều để ngăn chặc sự tấn công thông qua việc tìm kiếm các va chạm trong hàm hash.

***Hình 9.10: Sơ đồ chữ kí Schnorr.***

Cho p là số nguyên tố 512 bít sao cho bài toán logarithm rời rạc trong $Z_P$ là không giải được; cho q là số nguyên tố 160 bít chia hết cho p-1. Giả sử $\alpha \in Z_p^*$ là căn bậc q của 1modulo p. Cho h là hàm hash trong phạm vi $Z_p^*$. Định nghĩa $P = Z_p^*$ .$A = Z_p^* \times Z_P$ và định nghĩa:

$K = \{(p, q, \alpha, a, v) : v \equiv \alpha^{-a} (\text{mod } p)\}$

Các giá trị p, q, $\alpha$ và v là công khai còn a mật.

Với K = (p, q, $\alpha$, a, v) và với số ngẫu nhiên k mật $\in Z_p^*$, ta định nghĩa:

$sig_K(x,k) = (\gamma, y)$

trong đó

$\gamma = \alpha^k \bmod p$

và

$y = k + ah(x,\gamma) \bmod q.$

với $x, \gamma \in Z_p^*$ và $y \in Z_P$, định nghĩa

$ver(x, \gamma, y) = \text{true} \Leftrightarrow \gamma \equiv \alpha^y v^{h(x,y)} (\text{mod } p)$

## 9.6 Các chú giải và tài liệu tham khảo

Sơ đồ định danh Schnorr nêu trong tài liệu [Sc91], sơ đồ Okamoto được đưa ra trong [OK93] còn sơ đồ Guillou - quisquater có thể tìm thấy trong [GQ88]. Một sơ đồ khác chứng minh sự an toàn dưới giả thiết tính toán hợp lý là của Brickell và McCurley [BM92].

Các sơ đồ định danh phổ biến khác chứa đựng trong sơ đồ Fiege - Fiat - Shamir [FFS88] và sơ đồ nhân hoán vị [SH90]. Sơ đồ Fiege - Fiat - Shamir được chứng minh là an toàn nhờ dùng kĩ thuật tri thức không.

Phương pháp xây dựng sơ đồ chữ kí từ các sơ đồ định danh là do Fiat và Shamir đưa ra [FS87]. Chúng cũng được mô tả và phiên bản dựa trên tính đồng nhất của sơ đồ định danh của chính họ.

Các tổng quan về các sơ đồ định danh này đã được công bố trong công trình của Burmester, Desmedt và Beth [BDB92] và công trình của Waleffe và Quisquater [DWQ93].

## Các bài tập

**9.1**. Xét sơ đồ định danh sau đây: Alice sở hữu khoá mật n = pq, p và q là những số nguyên tố và $p \equiv q \equiv 3 \pmod 4$. Các giá trị n và ID(Alice) đều do TA kí như thường lệ và lưu trên dấu xác nhận của Alice. Khi Alice muốn tự xưng danh với Bob, Bob sẽ đưa cho Alice một thặng dư bình phương theo modulo n gọi là x. Sau đó Alice sẽ tính căn bình phương y của x và đưa nó cho Bob. Khi đó Bob xác minh xem $y^2$ có $\equiv x \pmod n$ hay không. Hãy giải thích tại sao sơ đồ này không an toàn.

**9.2**. Giả sử Alice đang dùng sơ đồ Schnorr với q = 1201, p =122503, t = 10 còn $\alpha$= 11538.

a/ Hãy kiểm tra xem $\alpha$ có bậc q trên $Z_p^*$ không.

b/ Giả thiết số mũ mật của Alice là $\alpha$ = 357, hãy tính v.

c/ Giả sử k = 868, hãy tính $\gamma$.

d/ Giả sử Bob đưa ra yêu cầu r = 501, hãy tính câu trả lời y của Alice.

e/ Thực hiện các tính toán của Bob khi xác minh y

**9.3.** Giả thiết, Alice dùng sơ đồ Schnorr với p, q, t như trong bài tập 9.2. v=51131 và giả sử Olga có thể nghiên cứu thấy rằng:

$$\alpha^3 v^{148} \equiv \alpha^{151} v^{1077} \pmod p$$

Hãy chỉ ra cách Olga có thể tính số mũ mật a của Alice.

9.4. Giả sử Alice đang dùng sơ đồ Okamoto với q = 1201, p = 122503, t= 10, $\alpha_1$=60497 và $\alpha_2$ = 17163

a/ Giả sử các số mũ mật của Alice $a_1$=432, $a_2$ = 423, hãy tính v.

b/ Giả sử $k_1$ = 389, $k_2$ = 191, tính $\gamma$

c/ Giả thiết Bob đưa ra yêu cầu r = 21. Hãy tính câu trả lời $y_1$ và $y_2$ của Alice

d/ Thực hiện các tính toan của Bob để xác minh $y_1$và $y_2$.

9.5/ Cũng giả thiết rằng Alice dùng sơ đồ Okamoto với p, q, t, $\alpha_1$và $\alpha_2$ như trong bài tập 9.4, và v = 119504.

a/ Hãy kiểm tra xem phương trình sau có thoả mãn không:

$$\alpha_1^{70} \alpha_2^{1033} v^{877} = \alpha_1^{248} \alpha_2^{883} v^{992} \pmod p$$

b/ Dùng thông tin trên để tính $b_1$ và $b_2$ sao cho:

$$\alpha_1^{-b_1}\alpha_2^{-b_2} \equiv v(\text{mod } p)\,.$$

c/ Giả sử rằng Alice để lộ $\alpha_1$=484 và $\alpha_2$ =935. Hãy chỉ ra cách Alice và Olga cùng nhau tính $\log_{\alpha_1}\alpha_2$.

**9.6.** Giả sử rằng, Alice đang dùng sơ đồ Quisquater với p = 503, q = 379 và b= 509.

a/ Giả sử giá trị u mật của Alice = 155863 tính v.

b/ Giả sử k = 123845, hãy tính γ.

c/ Giả thiết Bob đưa ra yêu cầu r = 487. Hãy tính câu trả lời y của Alice

d/ Thực hiện các tính toán của Bob để xác minh y

**9.7**. Giả sử Alice đang dùng sơ đồ Quisquater với n = 199543, b = 523 và v=146152. Giả thiết Olga đã khám phá ra rằng

$v^{456}101360^b = v^{257}36056^b(\text{mod } n)$

Hãy nêu cách Olga có thể tính u.

# CHƯƠNG 8
# PHÂN PHỐI VÀ THOẢ THUẬN VỀ KHOÁ

## 8.1 GIỚI THIỆU:

Chúng ta đã thấy rằng, hệ thống mã khoá công khai có ưu điểm hơn hệ thống mã khoá riêng ở chỗ không cần có kênh an toàn để trao đổi khoá mật. Tuy nhiên, đáng tiếc là hầu hết các hệ thống mã khoá công khai đều chậm hơn hệ mã khoá riêng, chẳng hạn như DES. Vì thế thực tế các hệ mã khoá riêng thường được dùng để mã các bức điện dài. Nhưng khi đó chúng ta lại trở về vấn đề trao đổi khoá mật.

Trong chương này, chúng ta sẽ thảo luận vài biện pháp thiết lập các khoá mật. Ta phân biệt giữa phân phối khoá và thoả thuận về khoá. Phân phối khoá được định nghĩa là cơ chế một nhóm chọn khoá mật và sau đó truyền nó đến các nhóm khác. Còn thoả thuận khoá là giao thức để hai nhóm (hoặc nhiều hơn) liên kết với nhau cùng thiết lập một khoá mật bằng cách liên lạc trên kênh công khai. Trong sơ đồ thoả thuận khoá, giá trị khoá được xác định như hàm của các đầu vào do cả hai nhóm cung cấp.

Giả sử, ta có một mạng không an toàn gồm n người sử dụng. Trong một số sơ đồ, ta có người uỷ quyền được tín nhiệm (TA) để đáp ứng những việc như xác minh danh tính của người sử dụng, chọn và gửi khoá đến người sử dụng ... Do mạng không an toàn nên cần được bảo vệ trước các đối phương. Đối phương (Oscar) có thể là người bị động, có nghĩa là hành động của anh ta chỉ hạn chế ở mức nghe trộm bức điện truyền trên kênh. Song mặt khác, anh ta có thể là người chủ động. Một đối phương chủ động có thể làm nhiều hành vi xấu chẳng hạn:

1. Thay đổi bức điện mà anh ta nhận thấy là đang được truyền trên mạng.
2. Cất bức điện để dùng lại sau này.
3. Cố gắng giả dạng làm những người sử dụng khác nhau trên mạng.

Mục tiêu của đối phương chủ động có thể là một trong những cái nêu sau đây:

1. Lừa U và V chấp nhận khoá "không hợp lê" như khoá hợp lệ (khoá không hợp lệ có thể là khoá cũ đã hết hạn sử dụng, hoặc khoá do đối phương chọn).
2. Làm U hoặc V tin rằng, họ có thể trao đổi khoá với người kia khi họ không có khoá.

Mục tiêu của phân phối khoá và giao thức thoả thuận khoá là, tại thời điểm kết thúc thủ tục, hai nhóm đều có cùng khoá K song không nhóm khác nào biết được (trừ khả năng TA). Chắc chắn, việc thiết kế giao thức có kiểu an toàn này khó khăn hơn nhiều trước đối phương chủ động.

Trước hết ta xem xét ý tưởng về sự phân phối khoá trước trong mục 8.2. Với mỗi cặp người sử dụng {U,V}, TA chọn một khoá ngẫu nhiên $K_{U,V}=K_{V,U}$ và truyền "ngoài dải" đến U và V trên kênh an toàn. (Nghĩa là, việc truyền khoá không xảy ra trên mạng do mạng không an toàn ). Biện pháp này gọi là an toàn không điều kiện song nó đòi hỏi một kênh an toàn giữa TA và những người sử

dụng trên mạng. Tuy nhiên điều quan trọng hơn là mỗi người phải lưu n -1 khoá và TA cần truyền tổng cộng $\binom{n}{2}$khoá một cách an toàn (đôi khi bài toán này được gọi là bài toán $n^2$). Thậm chí với một số mạng tương đối nhỏ, giá để giải quyết vấn đề này là khá đắt và như vậy giải pháp hoàn toàn không thực tế.

Trong phần 8.2.1, chúng ta thảo luận một sơ đồ phân phối trước khoá an toàn không điều kiện khá thú vị do Blom đưa ra. Sơ đồ cho phép giảm lượng thông tin mật mà người sử dụng cần cất giữ trên mạng. Mục 8.2.2 cũng đưa ra một sơ đồ phân phối trước khoá an toàn về mặt tính toán dựa trên bài toán logarithm rời rạc.

Một biện pháp thực tế hơn là TA phân phối khoá trực tiếp. Trong sơ đò như vậy, TA làm việc như một người chủ khoá (key server). TA chia khoá mật $K_U$ cho mỗi người sử dụng U trên mạng. Khi U muốn liên lạc với V, cô ta yêu cầu TA cung cấp khoá cho phiên làm việc (session key). TA tạo ra khoá session K và gửi nó dưới dạng mã hoá cho U và V để giải mã. Hệ thống mã Kerboros mô tả trong mục 8.3 là dựa trên biện pháp này.

Nếu như cảm thấy vấn đề phân phối khoá thông qua TA không thực tế hoặc không mong muốn thì biện pháp chung là dùng giao thức thoả thuận khoá. Trong giao thức thoả thuận khoá, U và V kết hợp chọn một khoá bằng cách liên lạc với nhau trên kênh công khai. ý tưởng đáng chú ý này do Martin và Diffie đưa ra độc lập với Merkle. ở đây mô tả vài giao thưc thoả thuận khoá phổ thông hơn. Giao thức đầu tiên của Diffie và Hellman được cải tiến để ứng phó với các đối phương tích cực được nêu trong phần 8.4.1. Hai giao thức đáng quan tâm nữa cũng được xem xét: sơ đồ MTI nên trong 8.4.2 và sơ đồ Girault nêu trong mục 8.4.3

## 8.2 Phân phối khoá trước

theo phương pháp cơ bản, TA tạo ra $\binom{n}{2}$ khoá và đưa mỗi khoa cho duy nhất một cặp người sử dụng trong mạng có n người sử dụng. Như đã nêu ở trên, ta cần một kênh an toàn giữa TA và mỗi người sử dụng để truyền đi các khoá này. Đây là một cải tiến quan trọng vì số kênh an toàn cần thiết giảm từ $\binom{n}{2}$ xuống còn n. Song nếu n lớn, giải pháp này cũng không thực tế cả về lượng thông tin cần truyền đi an toàn lẫn lượng thông tin mà mỗi người sử dụng phải cất giữ an toàn (nghĩa là các khoá mật của n-1 người sử dụng khác).

như vậy, điều cần quan tâm là cố gắng giảm được lượng thông tin cần truyền đi và cất giữ trong khi vẫn cho phép mỗi cặp người sử dụng U và V có khả năng tính toán khoá mật $K_{U,V}$. Một sơ đồ ưu việt hơn thoả mãn yêu cầu này là sơ đồ phân phối khoá trước của Blom.

### *8.2.1 Sơ đồ Blom*.

Như trên, giả thiết rằng có một mạng gôm n người sử dụng. Để thuận tiện, giả sử rằng các khoá được chọn trên trường số hữu hạn $Z_P$, $p \geq n$ là số nguyên tố. Cho k là số nguyên, $1 < k < n - 2$. Giá trị k để hạn chế kích thước lớn nhất mà sơ đồ vẫn duy trì được mật độ. Trong sơ đồ Blom, TA sẽ truyền đi k +1 phần tử của $Z_P$ cho mỗi 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 $K_{U,V}$. (chú ý rằng, ta đang xét sự an toàn không điều kiện).

Trước hết, xét trường hợp đặc biệt của sơ đồ Blom khi k =1. ở đây TA sẽ truyền đi 2 phần tử của $Z_P$ cho mỗi người sử dụng trên kênh an toàn và người sử dụng riêng W sẽ không thể xác định được bất kì thông tin nào về $K_{U,V}$ nếu $W \neq U,V$. Sơ đồ Blom được đưa ra trong hình 8.1. Ta sẽ minh hoạ sơ đồ Blom với k = 1 trong ví dụ sau:

***Hình 8.1: Sơ đồ phân phối khoá của Blom (k =1)***

1. Số nguyên tố p công khai, còn với mỗi người sử dụng U, phần tử $r_U \in Z_P$ là công khai. Phần tử $r_U$ phải khác biệt.
2. Ta chọn 3 phần tử ngẫu nhiên a, b, c $\in Z_P$ (không cần khác biệt) và thiết lập đa thức

## 8.3.Kerboros

trong các phương pháp phân phối trước khoá xem xét trong các phần trước đó, mỗi cặp người sử dụng cần tính một khoá cố định. Nếu dùng cùng một khoá trong một thời gian dài sẽ dễ bị tổn thương, vì thế người ta thường thích dùng phương pháp trực tiếp trong đó khoá của phiên lamà việc mới chỉ được tạo ra mỗi khi hai người sử dụng muốn liên lạc với nhau (gọi là tính tươi mới của khoá).

Nếu dùng phân phối khoá trực tiếp thì người sử dụng mạng không cần phải lưu các khoá khi muốn liên lạc với những người sử dụng khác (Tuy nhiên mỗi người đều được chia sẻ khoá với TA). Khoá của phiên làm việc (khóa session) sẽ được truyền đi theo yêu cầu của TA. Đó là sự đáp ứng của TA để đảm bảo khoá tươi.

Korobos là hệ thống dịch vụ khóa phổ cập dựa trên mã khoá riêng. Trong phần này sẽ đưa ra một tổng quan về giao thức phát hành khoá session trong Korobos. Mỗi người sử dụng U sẽ chia sẻ khoá DES mật $K_U$ cho TA. Trong phiên bản gần đây nhất của Korobos (version 5), mọi thông báo cần truyền được mã hoá theo chế độ xích khối (CBC) như mô tả trong 3.4.1

Như trong mục 8.2.2, ID(U) chỉ thông tin định danh công khai cho U. Khi có yêu cầu khoá session gửi đến TA, TA sẽ tạo ra một khoá session mới ngẫu nhiên K. Cũng vậy, TA sẽ ghi lại thời gian khi có yêu cầu T và chỉ ra thời gian (thời gian tồn tại) L để K có hiệu lực. Điều đó có nghĩa là khoá K chỉ có hiệu lực từ T đến T+L. Tất cả thông tin này đều được mã hoá và được truyênông dân đến U và V. Trước khi đi đến các chi tiết hơn nữa, ta sẽ đưa ra giao thức trong hình 8.4. thông tin được truyền đi trong giao thức được minh hoạ như sau:

***Hình 8.4: Truyền khoá session trong Korobos.***

1.

Ta sẽ giải thích điều sắp sửa xảy ra trong các bước của giao thức. Mặc dù không có chứng minh hình thức rằng Kerobos là an toàn trước đối thủ tích cực, song ít nhất ta cũng có thể đưa ra lí do nào đó về các đặc điểm của giao thức.

Như nêu ở trên, TA tạo ra K, T và L trong bước 2. Trong bước 3, thông tin này cùng với ID(V) được mã hoá bằng khoá $K_U$ (được U và TA chia sẻ) để tạo lập $m_1$. Cả hai bức điện đã mã hoá này được gửi đến U.

U có thể dùng khoá của mình giải mã $m_1$, nhận được K, T và L. Cô sẽ xác minh xem thời gian hiện tại có nằm trong khoảng T đến T + L hay không. Cô cũng kiểm tra khoá session K được phát ra cho liên lạc giữa cô và V bằng cách xác minh thông tin ID(V) đã giải mã từ $m_2$.

Tiếp theo, U sẽ làm trễ thời gian $m_2$ và $m_3$ đến V. Cũng như vậy, U sẽ dùng khoá session K mới để mã T và ID(U) và gửi kết quả $m_3$ đến V.

Khi V nhận được $m_3$ và $m_3$ từ U, V giải mã $m_2$ thu được T, K, L và ID(U). Khi đó, anh ta sẽ dùng khoá session mới K để giải mã $m_3$ và xác minh xem T và

ID(U) nhận được từ $m_2$ và $m_3$ có như nhau không. Điều này đảm bảo cho V rằng khoá session được mã bằng $m_2$ cũng là khoá đã dùng để mã $m_3$. Khi đó V dùng K để mã T+1 và gửi kết quả $m_4$ trở về U.

Khi U nhận được $m_4$, cô dùng K giải mã nó và xác minh xem kết quả có bằng T+1 không. Công đoạn này đảm bảo cho U rằng khoá session K đã được truyền thành công đến V vì K đã được dùng để tạo ra $m_4$.

Điều quan trọng cần lưu ý là các chức năng khác nhau của các thông báo dùng trong giao thức, $m_1$ và $m_2$ dùng để bảo đảm an toàn trong việc truyền khoá session. Còn $m_3$ và $m_4$ dùng để khẳng định khoá, nghĩa là cho phép U và V có thể thuyết phục nhau rằng họ sở hữu cùng một khoá session K. Trong hầu hết các sơ đồ phân phối khoá, sự khẳng định khoá đượoc coi như một đặc tính. Thường thì nó được thực hiện tương tự kiểu Kerobos, U dùng K để mã ID(U) và T dùng để mã trong $m_2$. Tương tự, V dùng K để mã T+1.

Mục đích của thời gian hệ thống T và thời hạn L để ngăn đối phương tích cực khỏi "lưu" thông báo cũ nhằm tái truyền lại sau này (đây được gọi là tấn công kiểu chơi lại - relay attack). Phương pháp này hiệu quả vì các khoá không được chấp nhận là hợp lệ một khi chúng quá hạn.

Một trong hạn chế của Kerobos là mọi người sử dụng trong mạng đều phải có đồng hồ đồng bộ với nhau vì cần có thời gian hiện tại để xác định khoá session K cho trước là hợp lệ. Thực tế, rất khó có được sự đồng bộ hoàn hảo nên phải cho phép có khoảng thay đổi nào đó về thời gian.

***Hình 8.5: Trao đổi khoá Diffie - Hellman***

## 8.4 Trao đổi khoá Diffie - Hellman

Nếu ta không muốn dùng dịch vụ khoá trực tiếp thì buộc phải dùng giao thức thoả thuận khoá để trao đổi khoá mật. Trước hết, giao thức thoả thuận khoá nổi tiếng nhất là giao thức trao đổi khoá Diffie - Hellman. Giả sử rằng, p là số nguyên tố, $\alpha$ là phần tử nguyên thuỷ của $Z_P$ và chúng đều là những tham số công khai. Giao thức trao đổi khoá Diffie - Hellman được đưa ra trong mục 8.5.

Cuối giao thức, U và V tính ra cùng một khoá:

Giao thức này cũng tương tự với sơ đồ phân phối khoá trước của Diffie - Hellman đã mô tả trước đây. Sự khác nhau ở chỗ các số mũ $a_U$, $a_V$ của U và V đều được chọn lại mỗi lần thực hiện giao thức thay vì cố định. Cũng như vậy, trong giao thức này, cả U lẫn V đều được đảm bảo khoá tươi vì khoá session phụ thuộc vào cả hai số mũ ngẫu nhiên $a_U$ và $a_V$.

### *8.4.1 Giao thức trạm tới trạm.*

Trao đổi khoá Diffie - Hellman được đề xuất như sơ đồ sau:

(Sơ đồ)

Đáng tiếc là giao thức dễ bị tổn thương trước đối phương tích cực - những người sử dụng tấn công "kẻ xâm nhập vào giữa cuộc" (Intuder - in -middle - attack). 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 nắm tay nhau dưới bàn. Lucy cố tránh bị phát hiện đã nắm tay của cả hai người, còn hai người vẫn nghĩ rằng họ đang nắm tay nhau.

Cuộc tấn công kiểu "kẻ xâm nhập giữa cuộc" trên giao thức trao đổi khoá Diffie - Hellman cũng như vậy. W sẽ chặn bắt được các bức điện trao đổi giữa U và V và thay thế bằng các bức điện của anh ta như sơ đồ dưới đây:

(sơ đồ)

Tại thời điểm cuối của 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 không thể, (tương tự tình huống nắm tay nhau nếu V gửi bức điện cho U).

Rõ ràng, điều cơ bản đối với U và V là bảo đảm rằng, họ đang trao đổi khoá với nhau mà không có W. Trước khi trao đổi khoá, U và V có thể thực hiện những giao thưc tách bạch để thiết lập danh tính cho nhau, ví dụ, nhờ dùng một trong các sơ đồ định danh mô tả trong chương 9. Tuy nhiên, điều này có thể đưa đến việc không bảo vệ được trước tấn công kẻ xâm nhập giữa cuộc nếu W vẫn duy trì một cách đơn giản sự tấn công thụ động cho đến khi U và V đã chứng minh danh tính của họ cho nhau. Vì thế giao thức thoả thuận khoá tự nó cần xác thực được các danh tính của những người tham gia cùng lúc khoá được thiết lập. Giao thức như vậy được gọi là giao thức thoả thuận khoá đã xác thực.

Ta sẽ mô tả một giao thức thoả thuận khoá là cải tiến của sơ đồ trao đổi khoá Diffie - Hellman. Giao thức giả thiết số nguyên tố p và phần tử nguyên thuỷ $\alpha$ là công khai và nó dùng với các dấu xác nhận. Mỗi người sử dụng U sẽ có một sơ đồ chữ kí với thuận toán xác minh $ver_U$. TA cũng có sơ đồ chữ kí với thuật toán xác minh công khai $ver_{TA}$. Mỗi người sử dụng U có dấu xác nhận:

$C(U) = (ID(U), ver_U, sig_{TA}(ID(U), ver_U))$

Trong đó ID(U) là thông tin định danh cho U

***Hình 8.6 Giao thức trạm tới trạm đơn giản.***

Thoả thuận khoá đã xác thực do Diffie - Hellman, van Oorschot và Viener đưa ra được gọi là giao thức trạm đến trạm (viết tắt là STS). Giao thức đưa ra trên hình 8.6 đơn giản hơn một chút: nó có thể được dùng để có thể phù hợp với các giao thức của ISO 9798-3.

Thông tin được trao đổi trong sơ đồ STS đã đơn giản hoá (gồm cả các dấu xác nhận) được minh hoạ như sau:

(sơ đồ)
Ta hãy xem cách bảo vệ này trước tấn công kẻ xâm nhập giữa cuộc. Như trước đây, W sẽ chặn bắt $\alpha^{a_U}$ và thay nó bằng

### *8.4.2. Các giao thức thoả thuận khoá MTI*

Matsumoto, Takashima và Imai đã xây dựng vài giao thức thoả thuận khoá đáng chú ý bằng cách biến đổi giao thức trao đổi khoá của Diffie - Hellman. Các giao thức này được gọi là MTI. Giao thức này không đòi hỏi U và V phải tính bất kì chữ kí nào. Chúng là các giao thức hai lần vì chỉ có hai lần truyền thông tin riêng biệt (một từ U đến V và một từ V đến U). Trái lại, giao thức STS được gọi là giao thức ba lần.

***Hình 8.7: Giao thức thoả thuận khoá MTI.***

Ta đã đưa ra một trong các giao thức MIT. Việc thiết lập chúng giống như giao thức phân phối khoá trước Diffie – Hellman. Giả thiết số nguyên tố p và phần tử nguyên thuỷ $\alpha$ là công khai. Mỗi người sử dụng U đều có chuỗi ID(U), số mũ mật $a_U$ ($0 \le a_U \le p-2$) và giá trị công khai tương ứng:

TA có sơ đồ chữ kí với thuật toán xác minh (công khai) $ver_{TA}$ và thuật toán kí mật $sig_{TA}$.

Mỗi người sử dụng U sẽ có dấu xác nhận:

$$C(U) = (ID(U), b_U, sig_{TA}(ID(U), b_U)).$$

Trong đó $b_U$ được thiết lập như trên.

Giao thức thoả thuận khoá MTI được đưa ra trên hình 8.7. Cuối giao thức U và V đều tính cùng một khoá:

K =

Dưới đây là ví dụ minh hoạ giao thức này:

*Ví dụ 8.3*.

Giả sử p = 27803, $\alpha$ = 5. Giả sử U chọn $a_U$ = 21131: sau đó cô ta tính:

$$b_U = 5^{21131} \bmod 27803 = 21420.$$

được đóng trên giấy xác nhận của cô. Cũng như vậy, V chọn $a_V$ = 17555. Sau đó anh ta sẽ tính:

$$b_V = 5^{17555} \bmod 27803 = 17100.$$

được dặt trên giấy xác nhận của anh.

Bây giờ giả sử rằng U chọn $r_U$ =169, sau đó cô gửi giá trị:

$$s_U = 5^{169} \bmod 27803 = 6268.$$

đến V. Lúc đó giả sử V chọn $r_V = 23456$, sau đó anh ta gửi giá trị:

$s_U = 5^{23456} \bmod 27803 = 26759$

đến U.

Bây giờ U tính khoá:

$K_{U,V} =$

$= 6268^{17555}\ 21420^{23456} \bmod 27803$

$= 21600.$

Như vậy, U và V đã tính cùng một khóa. …

Thông tin được truyền trong giao thức được miêu tả như sau:

(sơ đồ)

Hãy xét độ mật của sơ đồ. Không khó khăn nhận thấy rằng, độ mật của giao thức MTI trước tấn công thụ động đúng bằng bài toán Diffie – Hellman. Cũng như nhiều giao thức, việc chứng minh tính an toàn trước tấn công chủ động không phải đơn giản, chúng ta sẽ không thử chứng minh bất cứ điều gì về điều này và tự hạn chế đến một số đối số không hình thức.

Đây là một mối nguy hiểm có thể xem xét: Khi không dùng chữ kí trong suốt quá trình thực hiện giao thức, có thể xuất hiện tình huống không có sự bảo vệ nào trước tấn công xâm nhập vào điểm giữa. Quả thực, có khả năng W có thể chọn các giá trị mà U và V gửi cho nhau. Dưới đây mô tả một tình huống quan trọng có thể xuất hiện:

(sơ đồ)

Trong trường hợp này, U và V sẽ tính các khoá khác nhau: U tính

$K =$

Trong khi đó V tính:

$K =$

Tuy nhiên, W không thể tính toán ra khoá của U và V vì chúng đòi hỏi phải biết số mũ mật $a_U$ và $a_V$ tương ứng. Thậm chí ngay cả khi U và V tính ra các khoá khác nhau (mà dĩ nhiên là không dùng chúng) thì W cũng không thể tính được khoá nào trong chúng. Nói cách khác, cả U lẫn V đều được bảo đảm rằng, người sử dụng khác trên mạng chỉ có thể tính được khoá mà họ tính được. Tính chất này đôi khi được gọi là *xác thực khoá ẩn (implicit key authentication)*

### *8.4.3 Thoả thuận khoá dùng các khoá tự xác nhận*

Trong phần này, ta mô tả một phương pháp thoả thuận khoá do chính Girault đưa ra không cần dấu xác nhận. Giá trị của khoá công khai và danh tính người sở hữu nó sẽ ngầm xác thực lẫn nhau.

Sơ đồ Girault kết hợp các tính chất của RSA và các logarithm rời rạc. Giả sử $n = pq$, $p = p_1+1$, $q = 2q_l+1$, còn p, q, $p_1$và $q_1$ đều là các số nguyên tố lớn. Nhóm nhân $Z_n^*$ là đẳng cấu với $Z_p^* \times Z_q^*$. Bậc cực đại của phần tử bất kì trong $Z_n^*$ bởi vậy là bội chung nhỏ nhất của p - 1 và q - 1, hoặc $2p_1q_1$. Cho $\alpha$ là phân tử có

bậc $2p_1q_1$. Khi đó nhóm cyclic của $Z_n^*$ do $\alpha$ tạo ra là thiết lập thích hợp của bài toán logarithm rời rạc.

Trong sơ đồ Girault, chỉ TA biết được phân tích nhân tử của n. Các giá trị n và $\alpha$ là công khai, song p, q, $p_1$ và $q_1$ đều là mật. TA chọn số mũ mã công khai RSA, kí hiệu là e. Số mũ giải mã tương ứng bí mật là d (nhớ rằn $d = e^{-1} \bmod \phi(n)$).

Mỗi người sử dụng U có một chuỗi ID(U) như trong các sơ đồ trước đây. U nhận được khoá tự xác nhận công khai $p_U$ từ TA như nêu trên hình 8.8. Nhận xét rằng, U cần TA giúp đỡ để tạo $p_U$. Cũng chú ý rằng:

$$b_U = p_U^e + ID(U) \bmod n$$

***Hình 8.8: Nhận khoá tự xác nhận từ TA***

1. U chọn số mũ mật $a_U$ và tính:

   $b_U =$
2. U đưa $a_U$ và $b_U$ cho TA
3. TA tính:

   $p_U = (b_U - ID(U))^d \bmod n$
4. TA đưa $p_U$ cho U

Có thể tính từ $p_U$ và ID(U) bằng thông tin công khai có sẵn.

Giao thức thoả thuận khoá Girault được đưa ra trên hình 8.9. Thông tin truyền đi trong giao thức như sau:

$$U \quad \frac{ID(U),\ p_U,\ \alpha^{r_U} \bmod n}{ID(V),\ p_V,\ \alpha^{r_V} \bmod n} \quad V$$

Cuối giao thức, U và V tính khoá:

$$K = \alpha^{r_U a_V + r_V a_U} \bmod n$$

Dưới đây là một ví dụ về trao đổi khoá trong sơ đồ Girault.

*Ví dụ 8.4:*

Giả sử p =839, q = 863. Khi đó n = 724057 và $\phi(n) = 722356$. Phần tử $\alpha$=5 có bậc $2p_1q_1 = \phi(n)/2$. Giả sử TA chọn d = 125777 làm số mũ giải mã RSA, khi đó e = 84453.

Giả sử U có ID(U) = 500021 và $a_U$ = 111899. Khi đó $b_U$ = 488889 và $p_U$ =650704. Cũng giả thiết rằng V có ID(V) = 500022 và $a_U$ = 123456. Khi đó $b_V$ = 111692 và $p_V$ = 683556.

Bây giờ U và V muốn trao đổi khoá. Giả sử U chọn $r_U$ =56381, nghĩa là $s_U$=171007. Tiếp theo, giả sử V chọn $r_V$ = 356935, nghĩa là $s_V$ =320688.

Khi đó cả U lẫn V sẽ tính cùng một khoá K = 42869. …

***Hình 8.9: Giao thức thoả thuận khoá của Girault***

1. U chọn $r_U$ ngẫu nhiên và tính

   $s_u =$
2. U gửi ID(U), $p_U$ và $s_U$ cho V.
3. V chọn $r_V$ ngẫu nhiên và tính

$$s_V = \alpha^{r_V} \bmod n$$

4. V gửi ID(V), $p_V$ và $s_V$ cho U
5. U tính:

$$K = s_V^{a_U} (p_V^e + ID(V))^{r_U} \bmod n$$

Và V tính:

$$K = s_U^{a_V} (p_U^e + ID(U))^{r_V} \bmod n$$

Xét cách các khoá tự xác thực bảo vệ chống lại một kiểu tấn công. Vì các giá trị $b_U$, $p_U$ và ID(U) không được TA kí nên không có cách nào để ai đó xác minh trực tiếp tính xác thực của chúng. Giả thiết thông tin này bị W - người muốn giả danh U - giả mạo (tức là không hợp tác với TA để tạo ra nó). Nếu W bắt đầu bằng ID(U) và giá trị giả $b'_U$. Khi đó không có cách nào để cô ta tính được số mũ $a'_U$ tương ứng với $b'_U$ nếu bài toán logarithm rời rạc khó giải. Không có $a'_U$, W không thể tính được khoá.

Tình huống tương tự nếu W hoạt động như kẻ xâm nhập giữa cuộc. W sẽ có thể ngăn được U và V tính ra khoá chung, song W không thể đồng thời thực hiện các tính toán của U và V. Như vậy, sơ đồ cho khả năng xác thực ngầm như giao thức MTI.

Bạn đọc có thể tự hỏi tại sao U được yêu cầu cung cấp các giá trị $a_U$ cho TA. Quả thực, TA có thể tính $p_U$ trực tiếp từ $b_U$ mà không cần biết $a_U$ song điều quan trọng ở đây là TA sẽ được thuyết phục rằng, U biết $a_U$ trước khi TA tính $p_U$ cho U.

Điểm này được minh hoạ bằng cách chỉ ra sơ đồ có thể bị tấn tông nếu TA phát bừa bãi các khoá công khai $p_U$ cho những người sử dụng mà không kiểm tra trước hết xem họ có sở hữu các $a_U$ tương ứng với các $b_U$ của họ hay không. Giả sử W chọn một giá trị giả $a'_U$ và tính giá trị tương ứng:

$$b'_U = \alpha^{a'_U} \bmod n$$

Đây là cách anh ta có thể xác định khoá công khai tương ứng

$$p'_U = (b'_U - ID(U))^d \bmod n$$

W sẽ tính:

$$p'_W = b'_W - ID(U) + ID(W)$$

và sau đó đưa $b'_W$ và ID(W) cho TA. Giả sử TA phát ra khoá công khai

$$p'_W = (b'_W - ID(W))^d \pmod n$$

cho W. Nhờ dùng yếu tố:

$$b'_W - ID(W) \equiv b'_U - ID(U) \pmod n$$

có thể suy ra rằng: $p'_W = p'_U$.

Cuối cùng, giả sử U và V thực hiện giao thức còn W thay thế thông tin như sau:

$$\text{U} \quad \underset{\text{ID(V), } p_V,\ \alpha^{r_v} \bmod n}{\overset{\text{ID(U), } p_U,\ \alpha^{r_u} \bmod n}{\rightleftarrows}} \quad \text{V} \quad \underset{\text{ID(V), } p'_V,\ \alpha^{r_v} \bmod n}{\overset{\text{ID(U), } p'_U,\ \alpha^{r'_U} \bmod n}{\rightleftarrows}} \quad \text{W}$$

Xét thấy V sẽ tính khoá:

$$K' = \alpha^{r_U' a_V + r_V a_U'} \bmod n$$

trong khi U sẽ tính khoá

$$K = \alpha^{r_U a_V + r_V a_U} \bmod n$$

W có thể tính K' như sau:

$$K' = s_V^{a_U'} (p_V^e + ID(V))^{r_U'} \bmod n$$

Như vậy, W và V chia sẻ nhau một khoá, song V nghĩ anh ta đang chia khoá với U. Như vậy, W sẽ có thể giải mã được bức điện mà V gửi cho U.

## 8.5 Các chú ý và tài liệu tham khảo.

Blom đã đưa ra sơ đồ phân phối khoá của ông trong [BL85]. Các bài báo có tính chất tổng quát hoá cũng có trong một số bài báo khác của ông [BDSHKVY93] và của Beimel và Chor [BC94].

Diffie và Hellman đưa ra thuật toán trao đổi khoá của họ trong [DH76]. ý tưởng về trao đổi khoá cũng được Merkle đưa ra độc lập trong [ME78]. Những ý kiến về trao đổi khoá xác thực được lấy từ Diffie, Van Oorschot và Wiener [DVW92].

Phiên bản thứ 5 về Kerobos được mô tả trong [KN93]. Còn bài báo gần đây nhất về Kerobos xem trong [SC94] của Schiller.

Các giao thức của Matsumoto, Takashima và Imai có thể tìm thấy trong [MTI86]. Phân phối khoá tự xác nhận được giới thiệu trong Girault [GIR91]. Sơ đồ mà ông đưa ra thực sự là sơ đồ phân phối khoá trước: Bản cải tiến sơ đồ thoả thuận khoá dựa trên [RV94].

Hai tổng quan gần đây về phân phối khoá và thoả thuận khoá là của Rueppel và Van Oorschot [RV94] và Van Tilburg [VT93].

## Bài tập

8.1 Giả sử sơ đồ Blom với k =1 được thực hiện cho tập 4 người sử dụng, U, V, W và X. Giả thiết p = 7873, $r_U$ = 2365, $r_V$ =6648, $r_W$ = 1837 còn $r_X$ = 2186. Các đa thức mật g như sau:

$g_U(x) = 6018 + 6351x$

$g_V(x) = 3749 + 7121x$

$g_W(x) = 7601 + 7802x$

$g_X(x) = 635 + 6828x$

a/ Tính khoá cho mỗi cặp người sử dụng, xác minh rằng mỗi cặp nhận được một khoá chung (nghĩa là $K_{U,V} = K_{V,U}$ v.v...)

b/ Chỉ ra cách W và X cùng nhau tính khoá $K_{V,U}$

8.2 Giả thiết sơ đồ Blom với k = 2 được thực hiện cho tập 5 người sử dụng U, V, W, X và Y. Giả thiết p = 97, $r_U$ = 14, $r_V$ = 38, $r_W$ = 92, $r_X$ =69 còn $r_Y$ = 70. Các đa thức mật g như sau:

$g_U(x) = 15 + 15x + 2x^2$

$g_V(x) = 95 + 77x + 83x^2$
$g_W(x) = 88 + 32x + 18x^2$
$g_X(x) = 62 + 91x + 59x^2$
$g_Y(x) = 10 + 82x + 52x^2$

a/ Chỉ ra cách U và V tính khoá $K_{U,V} = K_{V,U}$
b/ Chỉ ra cách W, X và Y cùng nhau tính khoá $K_{U,V}$

***Hình 8.10: Bài toán MTI***

*Bài toán:* I =(p, α, β, γ, δ, ε) trong đó p là số nguyên tố, $\alpha \in Z^*_P$ là phần tử nguyên thuỷ còn β, γ, δ, ε $\in Z^*_P$
*Mục tiêu:* Tính $\beta^{\log_\alpha \gamma} \delta^{\log_\alpha \varepsilon} \bmod p$

8.3. Giả thiết U và V tiến hành trao đổi khoá theo sơ đồ Diffie - Hellman với p = 27001 và α = 101. Giả sử U chọn $a_U$ = 21768 và V chọn $a_V$ = 9898. Hãy chỉ ra các tính toán mà U và V thực hiện và xác định khoá mà họ tính được.

8.4. Giả thiết U và V tiến hành giao thức MTI với p = 30113, α = 52. Giả sử U có $a_U$ = 12385. Hãy chỉ ra các tính toán mà cả U và V thực hiện và xác định khoá mà họ tính được.

8.5. Nếu đối phương thụ động cố gắng tính K do U và V xây dựng bằng giao thưc MTI (hình 8.10), khi đó anh ta phải đối mặt với bài toán MTI. Chứng minh rằng thuật toán bất kì giải được bài toán MTI thì cũng có thể giải được bài toán Diffie - Hellman và ngược lại.

8.6. Xét sơ đồ định danh Girault trong đó p = 167, q = 179 và vì thế n = 29893. Giả sử α = 2 và e = 11101.

a/ Tính d.

b/ Cho trước ID(U) = 10021 và $a_U$ = 9843, tính $b_U$ và $p_U$. Cho trước ID(V) = 10022 và $a_V$ = 7692, hãy tính $b_V$ và $p_V$

c/ Chỉ ta cách có thể tính $b_U$ từ $p_U$ và ID(V) bằng cách dùng số mũ công khai e. Tương tự, chỉ ra cách tính $b_V$ từ $p_V$ và ID(V).

d/ Giả sử U chọn ra $r_U$ = 15556 và V chọn ra $r_V$ = 6420. Hãy tính $s_U$ và $s_V$ và chỉ ra cách U và V tính khoá chung của họ.

# *CHƯƠNG 7*

# CÁC HÀM HASH

## 7.1 CÁC CHỮ KÍ VÀ HÀM HASH.

Bạn đọc có thể thấy rằng các sơ dồ chữ kí trong chương 6 chỉ cho phép kí các bức điện nhỏ.Ví dụ, khi dùng DSS, bức điện 160 bit sẽ được kí bằng chữ kí dài 320 bít. Trên thực tế ta cần các bức điện dài hơn nhiều. Chẳng hạn, một tài liệu về pháp luật có thể dài nhiều Megabyte.

Một cách đơn giản để gải bài toán này là chặt các bức điện dài thành nhiều đoạn 160 bit, sau đó kí lên các đoạn đó độc lập nhau. Điều này cũng tương tự như mã một chuoĩ dài bản rõ bằng cách mã của mỗi kí tự bản rõ độc lập nhau bằng cùng một bản khoá. (Ví dụ: chế độ ECB trong DES).

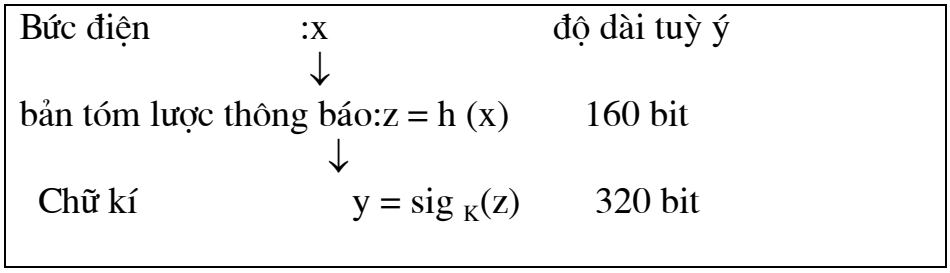
Biện pháp này có một số vấ đề trong việc tạo ra các chữ kí số. Trước hết, với một bức điện dài, ta kết thúc bằng một chữ kí rất lớn ( dài gấp đôi bức điện gốc trong trường hợp DSS). Nhược điểm khác là các sơ đồ chữ kí "an toàn" lại chậm vì chúng dùng các pháp số học phức tạp như số mũ modulo. Tuy nhiên, vấn đề nghiêm trọng hơn với phép toán này là búc điện đã kí có thể bị sắp xếp lại các đoạn khác nhau,hoặc một số đoạn trong chúng có thể bị loại bỏ và bức điện nhận được vẫn phải xác minh được. Ta cần bảo vệ sự nguyên vẹn của toàn bộ bức điện và điều này không thể thực hiện được bằng cách kí độc lập từng mẩu nhỏ của chúng.

Giải pháp cho tất cả các vấn đề này là dùng hàm Hash mã khoá công khai nhanh. Hàm này lấy một bức điện có độ dài tuỳ ý và tạo ra một bản tóm lược thông báo có kích thước qui định (160 bit nếu dùng DSS).

Sau đó bản tóm lược thông báo sẽ được kí. Vơi DSS, việc dùng hàm Hash được biểu diễn trê hình 7.1.

Khi Bob muốn kí bức điện x, trước tiên anh ta xây dựng một bnr tóm lược thông báo $z = h(x)$ và sau đó tính $y = sig_K(z)$. Bob truyền cặp ( x, y) trên kênh. Xét thấy có thể thực hiện xác minh (bởi ai đó ) bằng cách trước hết khôi phục bản tóm lược thông báo $z = h(x)$ bằng hàm h công khai và sau đó kiểm tra xem $ver_k(x,y)$ có = true, hay không.

**Hình 7.1.Kí một bản tóm lược thông báo**




---

## 7.2. HÀM HASH KHÔNG VA CHẠM

Chúng ta cần chú ý rằng,việc dùng hàm hash h không làm giảm sự an toàn của sơ đồ chữ kí vì nó là bản tóm lược thông báo được chữ kí không phải là bức điện. Điều cần thiết đối với h là cần thoả mãn một số tinhs chất nào đó để tranh sự giả mạo.

Kiểu tấn công thông thường nhất là Oscar bắt đầu bằng một bức diện được kí hợp lệ (x, y), $y = sig_K(h(x))$,(Cặp (x, y) là bức điện bất kì được Bob kí trước đó). Sau đó anh ta tính $z = h(x)$ và thử tìm $x \neq x'$ sao cho $h(x') = h(x)$. Nếu Oscar làm được như vậy, $(x', y)$ sẽ là bức điện kí hợp lệ, tức một bức điện giả mạo. Để tránh kiểu tấn công này, h cần thoả mãn tính không va chạm như sau:

Định nghĩa 7.1

Hàm hash h là hàm không va chạm yếu nếu khi cho trước một bức điện x, không thể tiến hành về mặt tính toán để tìm một bức điện $x \neq x'$ sao cho $h(x') = h(x)$.

Một tấn công kiểu khác như sau: Trước hết Oscar tìm hai bức điện $x \neq x'$ sao cho $h(x) = h(x')$. Sau đó Oscar đưa x cho Bob và thyết phục Bob kí bản tóm lược thông báo h(x) để nhận được y. Khi đố $(x',y)$ là thông báo (bức điện ) giả mạo hợp lệ.

Đây là lí do đưa ra một tính chất không va chạm khác.

Định nghĩa 7.2.

Hàm Hash h là không va chạm mạnh nếu không có khả năng tính toán để tìm ra bức điênk x và $x'$ sao cho $x \neq x'$ và $h(x) = h(x')$.

Nhận xét rằng: không va chạm mạnh bao hàm va chạm yếu.

Còn đây là kiểu tấn công thứ 3: Như đã nói ở phần 6.2 việc giả mạo các chữ kí trên bản tóm lược thông báo z ngẫu nhiên thường xảy ra với sơ đồ chữ kí. Giả sử Oscar tính chữ kí trên bản tóm lược thông báo z ngẫu nhiên như vậy. Sau đó anh ta tìm x sao cho z= h(x). Nếu làm được như vậy thì (x,y) là bức điện giả mạo hợp lệ. Để tránh được tấn công này, h cần thoả mãn tính chất một chiều (như trong hệ mã khoá công khai và sơ đồ Lamport).

Định nghĩa 7.3.
Hàm Hash h là một chiều nếu khi cho trước một bản tóm lược thông báo z, không thể thực hiện về mặt tính toán để tìm bức điện x sao cho h(x) = z.

Bây giờ ta sẽ chứng minh rằng, tính chất không va chạm mạnh bao hàm tính một chiều bằng phản chứng. Đặc biệt ta sẽ chứng minh rằng, có thể dùng thuật toán đảo với hàm Hash như một chương trình con (giả định ) trong thuật toán xác suất Las Vegas để tìm các va chạm.
Sự rút gọn này có thể thực hiện với một giả thiết yếu về kích thước tương đối của vùng và miền (domain and range) của hàm Hash. Ta cũng sẽ giả thiết tiếp là hàm Hash h: $X \to Z$, X,Z là các tập hữu hạn và $|X| \geq 2|Z|$. Đây là giả thiết hợp lí :Nếu xem một phần tử của X được mã như một xâu bít có độ dài $\log_2|X|$ và phần tử của Z được mã hoá như một xâu bít có độ dài $\log_2|X|$ thì bản tóm lược thông báo z = h(x) ít nhất cũng ngắn hơn bức điện x một bít (ta sẽ quan tâm đến tình huống vùng X là vô hạn vì khi đó có thể xem xét các bức điện dài tuỳ ý. Lập luận đó của ta cũng áp dụng cho tình huống này).

Tiếp tục giả thiết là ta có một thuật toán đảo đối với h, nghĩa là có một thuật toán A chấp nhận như đầu vào bản tóm lược thông báo $z \in Z$ và tìm một phần tử $A(z) \in X$ sao cho h(A(z)) = z.

Ta sẽ chứng minh định lí dưới đây:

Định lí 7.1:
Giả sử h: $X \to Z$ là hàm Hash, trong đó $|X|$ và $|Z|$ hữu hạn và $|X| \geq 2|Z|$. Cho A là thuật toán đảo đối với h. Khi đó tồn tại một thuật toán Las Vagas xác suất tìm được một va chạm đối với h với xác suất ít nhất là1/2.

Chứng minh :

Xét thuật toán B đưa ra trong hình 7.2. Rõ ràng B là một thuật toán xác suất kiểu Las Vegas vì nó hoặc tìm thấy một va chạm, hoặc cho câu trả lời không. Vấn đề còn lại là ta phải tịnh xac suất thành công, Với x bất kỳ thuộc X, định nghĩa $x \sim x_1$ nếu $h(x) = h(x_1)$. Dễ thấy rằng, ~ là quan hệ tương đương. Ta định nghĩa:

$$[x] = \{x_1 \in X: x \sim x_1\}$$

Mỗi lớp tương đương [x] chứa ảnh đảo của một phần tử thuộc Z nên số các lớp tương đương nhiều nhất là $|Z|$. Kí hiệu tập các lớp tương đương là C.

Bây giờ giả sử, x là phần tử $\in X$ được chọn trong bước 1. Với giá trị x này, sẽ có $|[x]|$ giá trị $x_1$ có thể cho phép trở lại bước 3. $|[x]|-1$ các giá trị $x_1$ này khác với x và như vậy bước 4 thành công. (Chú ý rằng thuật thoán A không biết biểu diễn các lớp tương đương [x] đã chon trong bước 1). Như vậy, khi cho trước lựa chọn cụ thể $x \in X$, xác suất thành công là $(|[x)|-1/|[x]|$.

Hình.7.2 Dùng thuật toán đảo A để tìm các va chạm cho hàm Hash

```
1.chọn một ssó ngẫu nhiên x ∈X
2.Tính z=h(x)
3.Tinh x1= A(Z)
4. if x1 ≠ x then
        x và x1 va chạm dưới h (thành công)
   else
        Quit (sai)
```

Xác suất thành công của thuật toán B bằng trung bình cộng tất cả các lựa chon x có thể:

$$\begin{aligned} P(\text{thành công}) &= (1/|X|)\sum_{x\in X}(|[x]|-1)/|[x]| \\ &= (1/|X|)\sum_{c\in C}\sum_{x\in C}(|c|-1)/|c| \\ &= 1/|X|\sum_{c\in C}(|c|-1) = (1/|X|)\sum_{c\in C}|c| - \sum_{c\in C}1 \\ &>= (|X| - |Z|) / |X| \\ &>= ((|X| - |Z|)/2) / |X| = \tilde{a} \end{aligned}$$

Như vậy, ta đã xây dựng thuật toán Las Vegas có xác suất thành công ít nhất bằng 1/2.

Vì thế, đó là điều kiện đủ để hàm Hash thoả mãn tính chất không va chạm mạnh vì nó bao hàm hai tính chất khác.Phần còn lại của chương này ta chỉ quan tâm đến các hàm Hash không va chạm mạnh.

## 7.3 TẤN CÔNG NGÀY SINH NHẬT(birthday)

Trong phần này, ta sẽ xác định điều kiện an toàn cần thít ch hàm Hash và điều kiện này chỉ phụ thuộc vào lực lượng của tập Z (tương đương về kích thước của bảng thông báo ).Điều kiện cần thiết nà rút ra tư phương pháp tìm kiếm đơn giản ác va chạm mà người ta đã biết đến dưới cái tên tấn công ngày sinh nhật (birthday phương pháparradox), trong bài toán:một nhóm 23 người ngẫu nhiên, có ít nhất 2 người có ngày sinh trùng nhau với xác suất ít nhất là1/2.(Dĩ nhiên, đây chưa phải là nghịch lí,song đó là trực giác đối lập có thể xảy ra). Còn lí do của thuật ngữ "tấn công ngày sinh nhật " sẽ rõ ràng khi ta tiếp tuch trình bày.

Như trước đây, ta hãy giả sử rằng :h:X→Z là hàm Hash, X,Z hữu hạn và $|X| >= 2|Z|$.Định nghĩa $|X| = m$ và $|Z| = n$.Không khó khăn nhận thấy rằng, có ít nhất n va chạm và vấn đề đặt ra là cách tìm chúng. Biện pháp đơn sơ nhất là chọn k phần tử ngẫu nhiên phân biệt $x_1,x_2\ldots..x_k \in X$, tính $z_1 = h(x_1),1<= i <= k$ và sau đó xác định xem liệu có xảy ra va chạm nào không (bằng cách, chẳng hạn như sáp xếp lại các $z_i$).

Quá trình này tương tự với việc ném k quả bóng vào thùng và sau đó kiểm tra xem liệu có thùng nào chứa ít nhất hai quả hay không (k qủa bóng tương đương với k giá trị $x_i$ ngẫu nhiên và n thùng tương ứng với n phần tử có thể trong Z).

Ta sẽ giới hạn dưới của xác suất tìm thấy một va chạm theo phương pháp này.Do chỉ quan tâm đến giới hạn dưới về xác suất va chạm nên ta sẽ giả sử rằng $|h^{-1}(z)| \approx m/n$ với mọi $z \in Z$. (đây là giả thiết hợp lí :Nếu các ảnh đảo không xấp xỉ bằng nhau thì xác suất tìm thấy một va chạm sẽ tăng lên ).

Vì các ảnh đảo đều có kích thước bằng nhau và các $x_{i\ được}$ chọn một cách ngẫu nhiên nên các $z_i$ nhận được có thể xem như các phần tử ngẫu nhiên của Z. Song việc tính toán xác suất để các phần tử ngẫu nhiên $z_1, z_{2,\ldots} z_k \in Z$ là riêng biệt khá đơn giản.Xét các $z_i$ theo thứ tự $z_1, \ldots,z_k$. Phép chọn $z_1$ đầu tiên là tuỳ ý. Xác suất để $z_2 \neq z_1$ là 1-1/n; xác suất để $z_3 \neq z_1$ và $z_2$ là 1- 2/n. vv…

Vì thế ta ước lượng xác suất để không có va chạm nào là:

$(1-1/n)(1-2/n)\ldots (1-(k-1/n)) = (1-1/n)$

Nếu x là số thực nhỏ thì 1- x ≈ $e^{-x}$. Ước lượng này nhận dược từ hai số hạng đầu tiên của cá chuỗi khai triển.

$$e^{-x} = 1 - x + x^2/2! - x^3/3! \ ...$$

Khi đó xác suất không có va chạm nào là :

$$\prod_{i=1}^{k-1}(1-\frac{i}{n}) \approx \prod_{i=1}^{k-1} e^{-1/n} = e^{-k(k-1)/n}$$

Vì thế ta ước lượng xác suất để có ít nhất một va chạm là

$$1-e^{-k(k-1)/n}$$

Nếu kí hiệu xác suất này là ε thì có thể giải phương trình đối với k (như một hàm của n và ε)

$$1-e^{-k(k-1)/n} \approx 1 -\varepsilon$$

$$-k(k-1)/n \approx \ln(1-\varepsilon)$$

$$k^2 - k \approx n\ln 1/(1-\varepsilon)$$

Nếu bỏ qua số hạng k thì :

$$k= \sqrt{n ln\frac{1}{1-\varepsilon}}$$

Nếu lấy ε = 0.5 thì

$$k \approx 1.17\sqrt{n}$$

Điều này nói lên rằng, việc chặt (băm) trên $\sqrt{n}$ phần tử ngẫu nhiên của X sẽ tạo ra một va chạm với xác suấtt 50%. Chú ý rằng, cách chọn ε khác sẽ dẫn đến hệ số hằng số khác song k vẫn tỷ lên với $\sqrt{n}$ .

Nếu X là tập người,Y là tập gồm 365 ngỳ trong năm (không nhuận tức tháng 2 có 29 ngày) còn h(x) là ngày sinh nhật của x, khi đó ta sẽ giả guyết bằng nhgịch lý ngày sinh nhật. Lấy n = 365, ta nhận được k ≈ 22,3. Vì vậy, như đã nêu ở trên, sẽ có ít nhất 2 người có ngày sinh nhật trùng nhau trong 23 người ngẫu nhiên với xác suất ít nhất bằng 1/2.

Tấn công ngày sonh nhật đặt giới hạn cho các kích thước các bản tóm lược thông báo. bản tóm lược thông báo 40 bit sẽ không an toàn vì có thể tìm thấy một va chạm với xác suất 1/2 trên $2^{20}$ (khoảng1.000.000)đoạn chặt ngẫu nhiên. Từ đây cho thấy rằng, kích thước tối thiểu chấp nhận được của bản tóm lược thông báo là 128 bit (tấn công ngày sinh nhật cần trên $2^{64}$ đoạn chặt trong trường hợp này). Đó chính là lý do chọn bản tóm lược thông báo dài 160 bit trong sơ đồ DSS.

**Hình7.3. Hàm hash chaum-Van heyst-Plitzmann.**

Giả sử p là số nguyên tố lớn và q =(p-1)/2 cũng là số nguyên tố. Cho $\alpha$ và $\beta$ là hai phần tử nguyên thuỷ của Zp. Giá trị $\log_\alpha\beta$ không công khai và giả sử rằng không có khả năng tính toán được giá trị của nó.
Hàm Hash:

$h: \{0,...,q-1\}\times\{0,...,q-1\} \rightarrow Zp\setminus\{0\}$

được định nghĩa như sau:

$h(x1,x2) = \alpha^{x_1}\beta^{x_2} \bmod p$

## 7.3. hàm hash logarithm rời rạc

Trong phần này ta sẽ mô tả một hàm Hash do Chaum-Van Heyst và Pfĩtmann đưa ra. Hàm này an toàn do không thể tính được logarithm rời rạc. Hàm Hast này không đủ nhanh để dùng trong thực tế song nó đơn giản và cho một ví dụ tốt về một hàm Hash có thể an toàn dưới giả thuyết tính toán hợp lý nào số. Hàm Hash Caum-Van Heyst- Pfĩtmann được nêt trong hình 7.3. Sau đây sẽ chứng minh một định lý liên quan đến sự an toàn của hàm Hast này.

**Định lý 7.2.**

**N***ếu cho trước một va chạm với hàm Hash Chaum-Van Heyst-Pfĩtmann h có thể tính được logarithm rời rạc $\log_\alpha\beta$ một cách có hiệu quả.*

*Chứng minh*

Giả sử cho trước va chạm

$h(x_1,x_2) = h(x_3,x_4)$

trong đó $(x_1,x_2) \neq (x_3,x_4)$. Như vậy ta có đồng dư thức sau:

$\alpha^{x_1}\beta^{x_2} = \alpha^{x_3}\beta^{x_4}$

hay

$\alpha^{x_1}\beta^{x_2} \equiv \alpha^{x_3}\beta^{x_4} \pmod p$

Ta kí hiệu

$D = UCLN\ (x_4-x_2,p-1)$

Vì p-1 =2q ,q là số nguyên tố nên d ∈ {1, 2, q, p-1}. Vì thế, ta có 4 xác suất với d sẽ xem xét lần lượt dwois đây.

Trước hết ,giả sử d =1 ,khi đó cho

$$y= (x_4-x_2)^{-1} \bmod (p-1)$$

ta có

$$\beta \equiv \beta^{(x_4-x_2)y} (\bmod p)$$
$$\equiv \alpha^{(x_1-x_2)y} (\bmod p)$$

Vì thế, có thể tính loarithm rời rạc $\log_\alpha \beta$ như sau:

$$\log_\alpha \beta = (x_1-x_3)\ (x_4-x_2)^{-1} \bmod (p-1)$$

Tiếp theo, giả sử d=2. Vì p-1 =2q, lẻ nên UCLN($x_4$-$x_2$,q) =1. Giả sử:

$$y=(x_4-x_2)^{-1} \bmod q$$

xét thấy $(x_4-x_2)y = kq+1$

với số nguyên k nào đó. Vì thế ta có:

$$\beta^{(x_4-x_2)y} \equiv \beta^{kq+1} (\bmod p)$$
$$\equiv (-1)^k \beta (\bmod p)$$
$$\equiv \pm \beta (\bmod p)$$

Vì $\beta^q \equiv -1 (\bmod p)$

Nên

$$\alpha^{(x4-x2)y} \equiv \beta^{(x1-x3)} (\bmod p)$$
$$\equiv \pm \beta (\bmod p)$$

Từ đó suy ra rằng:

$$\log_\alpha \beta = (x_1-x_3)y \bmod (p-1)$$
$$\log_\alpha \beta = (x_1-x_3)y \bmod (p-1)$$

Ta có thể dễ dàng kiểm tra thấy một trong hai xác suất trên là đúng. Vì thế như trong trường hợp d =1, ta tính được $\log_\alpha \beta$.

Xác suất tiếp theo là d = q. Tuy nhiên

$$q-1 \geq x_1 \geq 0$$

và

$$q-1 \geq x_3 \geq 0$$

nên

$$(q-1) \geq x_4-x_2 \geq -(q-1)$$

do vậy UCLN($x_4$-$x_2$,p-1) không thể bằng q, nói cách khác trường hợp này không xảy ra.

Xác suất cuối cùng là d = p-1. Điều nàychỉ xảy ra khi x2 =x4. Song khi đó ta có

$$\alpha^{x_1}\beta^{x_2} \equiv \alpha^{x_3}\beta^{x_4} (\bmod p)$$

nên $\alpha^{x_1} \equiv \alpha^{x_3} \pmod p$

và $x_1 = x_2$. Như vậy $(x_1,x_2) = (x_3,x_4) \Rightarrow$ mâu thuẫn. Như vậy trường hợp này cũng không thể có.

Vì ta đã xem xét tất cả các giá trị có thể đối với d nên có thể kết luận rằng ,hàm Hash h là không va chạm mạnh miễn là không thể tính được logarithm rời rạc $\log_\alpha\beta$ trong $Z_p$.

Ta sẽ minh hoạ lý thuyết nêu trên bằng một ví dụ.

Ví dụ 7.1

Giả sử p =12347 (vì thế q = 6173), $\alpha = 2$, $\beta = 8461$. Giả sử ta được đưa trước một va chạm

$$\alpha^{5692}\ \beta^{144} \equiv \alpha^{212}\ \beta^{4214} \pmod{12347}$$

Như vậy $x_1 = 5692$, $x_2 = 144$, $x_3 = 212$, $x_4 = 4214$. Xét thấy UCLN $(x_4 - x_2, p-1) = 2$ nên ta bắt đầu bằng việc tính

$$y = (x_4 - x_2)^{-1} \bmod q$$
$$= (4214 - 144)^{-1} \bmod 6173 = 4312$$

Tiếp theo tính

$$y = (x_1 - x_3) \bmod (p-1)$$
$$= (5692 - 212)\ 4312 \bmod 12346$$
$$= 11862$$

Xét thấy đó là trường hợp mà $\log_\alpha\beta \in \{y', y'+q \bmod (p-1)\}$. Vì

$$\alpha^y \bmod p = 2^{12346} = 9998$$

nên ta kết luận rằng:

$$\log_\alpha\beta = y' + q \bmod (p-1)$$
$$= 11862 + 6173 \bmod 12346$$
$$= 5689$$

như phép kiểm tra, ta có thể xác minh thấy rằng

$$2^{5689} = 8461 \pmod{12347}$$

Vì thế , ta các định được $\log_\alpha\beta$.

---

## 7.5.**các hàm hash mở rộng**

Cho đến lúc này, ta đã xét các hàm Hash trong vùng hữu hạn. Bây giờ ta nghiên xéu cách có thể mở rộng một hàm Hash không va chạm mạnh từ vùng hữu hạn sang vùng vô hạn. Điều này cho phép ký các bức điện có độ dài tuỳ ý. Gỉa sử h: $(Z_2)^m \rightarrow (Z_2)^t$ là một hàm hash không va chạm mạnh ,trong đó $m \geq t-1$. Ta sẽ dùng h đêu xây dựng hàm hash không va chạm mạnh h: $X \rightarrow (Z_2)^t$ trong đó

$$X = \bigcup_{i=m}^{\infty} (Z_2)^t$$

Trước tiên xét trường hợp m ≥ t+2.

Ta sẽ xem các phần tử của X như các xây bit. |x| chỉ độ dàI của x (tức số các bit trong x) và x||y ký hiệu sự kết hợp các xây x và y. Giả sử |x| = n > m. Có thể biểu thị x như một chuỗi kết hợp.

$$X = x_1||x_2||...||x_k$$

Trong đó

$$|x_1| = |x_2| = ... = |x_{k-1}| = m- t-1$$

và $|x_k| = m- t- 1- d$

Hình 7.4. Mở rộng hàm hash h thành h* (m ≥t+2)

1. For i= 1 to k-1 do
   $y_i = x_i$
2. $y_k = x_k ||0^d$
3. cho $y_{k+1}$ là biểu diễn nhị phân của d
4. $g_i = h(0I+1||y_1)$
5. for i=1 to k do
   $g_i+1 = h(g_i||1||y_i+1)$
6. $h^*(x) = g_k +1$

Trong đó m- t- 2 ≥ d ≥0. Vì thế ta có

$$k= \left\lceil \frac{n}{m-t-1} \right\rceil$$

Ta định nghĩa h*(x) theo thuật toán biểu kiễn trong hình 7.4.

Kí hiệu $y(x) = y1||y2||...||y_{k-1}$

Nhận xét rằng $y_k$ được lập từ $x_k$ bằng cách chèn thêm d số 0 vào bên phảI để tất cả các khối $y_i$ (k ≥ i ≥ 1)đều có chiều dàI m-t-1. Cũng như trong bước 3 $y_{k+1}$ sẽ được đệm thêm về bên tráI các số 0 sao cho $|y_{k+1}| = m-t-1$.

Để băm nhỏ x ,trước hết ta xây dựng hàm y(x) và sau đó "chế biến" các khối $y_1...y_{k+1}$ theo một khuôn mẫu cụ thể. Điều quan trọng là y(x) ≠y(x') khi

x≠x. Thực tế $y_{k+1}$ được định nghĩa theo cách các phép ánh xạ x → y(x)là một đơn ánh.

Định lý sau đây chứng minh rằng h* là an toàn khi h an toàn.

*Định lý 7.3*

*Giả sử h:* $(Z_2)^n \to (Z_2)$ *là hàm hash không va chạm mạnhm≥t+2. Khi đó hàm h*:* $\cup_{i=m}^{\infty}$ $(Z_2)^t \to (Z_2)^t$ *được xây dựng như trên hình 7.4 là hàm hash không*
*và chạm mạnh.*

*Chứng minh:*

Giả sử rằng ,ta có thể tìm được x ≠x' sao cho h*(x) = h*(x'). Nếu cho trước một cặp như vậy, ta sẽ chỉ ra cách có thể tìm được một va chạm đối với h trong thời gian đa thức. Vì h được giả thiết là không va chạm mạnh nên dẫn đến một mâu thuẫn như vậy h sẽ được chứng minh là không va chạm mạnh.

Kí hiệu $y(x)= y_1||..||y_{k+1}$

Và $y(x') = y_1'||...||y_{k+1}'$

ở đây x và x' được đệm thêm d và d' số 0 tương ứng trong bước 2. Kí hiệu tiếp các giá trị được tính trong các bước 4 và 5 là $g_1,g_2....,g_{k+1}$ và $g_1',.....,g_{k+1}'$ tương ứng.

Chúng ta sẽ đồng nhất hai trường hợp tuỳ thuộc vào việc có hay không |x| ≡|x'| (mod m-t-1).

*Trường hợp1*: |x| ≠|x'| (mod m-t-1)

Tại đây d ≠d' và $y_{k+1} \neq y'_{k+1}$. Ta có:

$$\begin{aligned} H(g_k||1||y_{k+1}) &= g_{k+1} \\ &= h^*(x) \\ &= h^*(x') \\ &= g'_{t+1} \\ &= h(g'_{t+1}||1||y'_{t+1)} \end{aligned}$$

là một va chạm đối với h vì $y_{k+1} \neq y'_{k+1.}$

*Trường hợp2:* |x| ≡|x'| (mod m-t-1)

Ta chia trường hợp này thành hai trường hợp con:

*Trường hợp 2a*: |x| = |x'|.

Tạ đây ta có k= l và $y_{k+1} = y'_{k+1}$. Ta vắt đầu như trong trường hợp 1:

$$h(g_k||1||y_{k+1}) = g_{k+1}$$
$$= h^*(x)$$
$$= h^*(x')$$
$$= h(g'_k||1||y'_k+1)$$

Nếu $g_k = g'_k$ thì ta tìm thấy một va chạm đối với h, vì thế giả sử $g_k = g'_k$ khi đó ta sẽ có:

$$h(g_{k-1}||1||y_k) = g_k$$
$$=g'_k$$
$$=h(0^{i+1}||y_1)$$

Hoặc là tìm thấy một va chạm đối với h hoặc $g_{k-1} =g'_{k}-_1$ và $y_k = y'_k$. Giả sử không tìm thấy va chạm nào ,ta tiếp tục thực hiện ngược các bước cho đến khi cuối cùng nhận được :

$$h(0_{i+1}||y_1) = g_1$$
$$=g'_{i-k+1}$$
$$=g(g'_{i-k}||1||y'_{i-k+1}).$$

Nhưng bit thứ (t+1) của $0^{i+1}||y_1$ bằng 0 và bit thứ (t+1) của $g'_{i-k+1}||1||y'_{i-k+1}$ bằng 1. Vì thế ta tịm thấy một va chạm đối với h.

Vì đã xét hết các trường hợp có thể nên ta có kết luận mong muốn.

Cấu trúc của hình 7.4 chỉ được dùng khi m>= t+2. Bây giờ ta hãy xem xét tình huống trong đó m = t+1. Cần dùng một cấu trúc khác cho h. Như trước đây, giả sử |x|=n>m. Trước hết ta mã x theo cách đặc biệt. Cách này dùng hàm f có định nghĩa như sau:

f(0) = 0
f(1) = 01

Thuật toán để xây dựng h*(x)được miêu tả trong hình 7.5

Phép mã $x \rightarrow y = y(x)$ được định nghĩa trong vước 1 thoả mãn hai tính chất quan trọng sau:

1. nếu x ≠x' thì y(x)≠ y(x') (tức là x→ y(x) là một đơn ánh)
2. Không tồn tạI hai chuỗi x≠ x' và chuỗi z sao cho y(x)= z||y(x'). Nói cách khác không cho phép mã hoá nào là fpsstix của phép mã khác. ĐIều này dễ dàng thấy được do chuỗi y(x) bắt đầu bằng 11 và không tồn tạI hai số 1 liên tiếp trong phần còn lạI của chuỗi).

Hình 7.5 Mở rộng hàm hash h thành h* (m = t+1)

*Định lý 7*

1. Giả sử $y = y_1y_2...y_k = 11||f(x_1)||....||f(x_n)$
2. $g_1 = h(0^1||y_1)$
3. for i=1 to k-1 do
   $g_{i+1} = h(g_i||y_{i+1})$
4. $h^*(x) = g_k$

*Giả sử h: $(Z_2)^n \rightarrow (Z_2)$ là hàm hash không va chạm mạnh. Khi đó hàm h*: $\cup_{i=m}^{\infty}$ $(Z_2)^t \rightarrow (Z_2)^t$ được xây dựng như trên hình 7.5 là hàm hash không va chạm mạnh.*

*Chứng minh:*

Giả sử rằng ta có thể tìm được x ≠x' sao cho h*(x)=h*(x'). Kí hiệu:

$$y(x) = y_1y_2....y_k$$

và $$y(x') = y'_1y'_2....y'_l$$

Ta xét hai trường hợp:

Trường hợp 1: k=l

Như trong định lý 7.3 hoặc ta tìm thấy một va chạm đỗi với h hoặc ta nhận được y = y' song đIều này lạI bao hàm x = x', dẫn đến mâu thuẫn.

*Trường hợp2*: k≠ l

Không mất tính tổng quát ,giả sử l>k . trường hợp này xử lý theo kiểu tương tự. Nếu giả thiết ta không tìm thấy va chạm nào đối với h ,ta có dãy các phương trình sau:

$y_k = y'_l$
$y_{k-1} = y'_{l-1}$
.....
$y_1 = y'_{l-k+1}$

Song đIều này mâu thuẫn với tính chất "không posfixx" nêu ở trên. Từ đây ta kết luận rằng h* là hạm không va chạm.

Ta sẽ tổng kết hoá hai xây dựng trong phần này và số các ứng dụng của h cần thiết để tính h* theo định lý sau:

***Định lý 7.5***

*Giả sử h: $(Z_2)^n \to (Z_2)$ là hàm hash không va chạm mạnh,ở đây m>=t+1. Khi đó tồn tạI hàm không va chạm mạnh*

$$h^*: \cup_{i=m}^{\infty} (Z_2)^t \to (Z_2)^t$$

*Số lần h được tính trong ước lượng h* nhiều nhất bằng :*

$$l + \left\lceil \frac{n}{m-t-1} \right\rceil \text{ nếu } m>=t+2$$

$2n+2$ *nếu* $m= t+2$

*trong đó* $|x|=n$.

---

7.6 các hàm hash dựa trên các hệ mật

Cho đến nay, các phương pháp đã mô tả để đưa đến nhứng hàm hash hầu như đều rất chậm đối với các ứng dụng thực tiễn. Một biện pháp khác là dùng các hệ thống mã hoá bí mật hiện có để xây dừng các hàm hash. Giả sử rằng (P,C,K,E,D) là một hệ thống mật mã an toàn về mặt tính toán. Để thuận tiện ta cũng giả thiết rằng $P = C = K = (Z_2)^n$.ở đâychọn n>=128 để xây ngăn chặn kiểu tấn công ngày sinh nhật. ĐIều này loạI trừ việc dùng DES (vì độ dài khoá của DES khác với độ dài bản rõ).

Giả sử cho trước một xâu bit:

x= x1||x2||....||xk

trong đó xi $\in$ $(Z_2)^n$, 1≤ i ≤ (nếu số bit trong x không phải là bội của n thì cần chèn thêm vào x theo cách nào đó. Chẳng hạn như cách làm trong nục 7.5. Để đơn giản ta sẽ bỏ qua đIểm này).

Ý tưởng cơ bản là bắt đầu bằng một "giá trị ban đầu" cố định $g_0$ =IV và sau đó ta xây dựng $g_1$,...,$g_k$ theo quy tắc thiết lập :

$$g_i = f(x_i, g_{i-1}).$$

ở đây f là hàm kết hợp toàn bộ các phép mã hoá của hệ mật được dùng. Cuối cùng ta định nghĩa bản tóm lược của thông báo h(x) =$g_k$.

Vài hàm hash kiểu này đã được đề xuất và nhiều loại trong chúng tỏ ra không an toàn (không phụ thuộc vào việc liệu hệ mật cơ bản có an toàn hay không ). Tuy nhiên , có 4 phương án khác nhau có vẻ an toàn của sơ đồ này :

$$g_i = e_{g_{i-1}}(x_i) \oplus x_i$$
$$g_i = e_{g_{i-1}}(x_i) \oplus x_I \oplus g_{i-1}$$
$$g_i = e_{g_{i-1}}(x_i \oplus g_{i-1}) \oplus x_I$$
$$g_i = e_{g_{i-1}}(x_i \oplus g_{i-1}) \oplus x_I \oplus g_{i-1.}$$

## 7.7 **Hàm hash MD4.**

Hàm hash MD4 được Riverst đề xuất năm 1990 và một hiên bản mạnh là MD5 cũng được đưa ra năm 1991. Chuẩn hàm hash an toàn (hay SHS) phức tạp hơn song cũng dưa tên các phương pháp tương tự. Nó được công bố trong hồ sơ liên bang năm 1992 và được chấp nhận làm tiêu chuẩn vào ngày 11/5/1993. Tất cả các hàm hash trên đều rất nhanh nên trên thực tế chúng dùng để kí các bức điện dài.

Trong phần này sẽ mô tả chi tiết MD4 và thảo luận một số cảI tiến dùng trong MD5 và SHS.

Cho trước một xâu bit trước hết ta tạo một mạng:

M = M[0] M[1]... M[N-1] .

trong đó M[i] là xâu bit có độ dàI 32 và N ≡ 0 mod 16. Ta sẽ gọi M[i] là *từ*. M được xây dựng từ x bằng thuật toán trong hình 7.6.

Hình 7.6 Xây dựng M trong MD4

1. d = 447-(|x| mod 512)
2. giả sử $\ell$ là kí hiệu biểu diễn nhị phân của |x| mod $2^{64}$.|l| = 64
3. M = x||1||$0^d$||$\ell$

Trong việc xây dựng M, ta gắn số 1 ssơn lẻ vào x, sau đó sẽ gài thêm các số 0 đủ để độ dài trở nên đồng dư với 448 modulo 512.,cuối cùng nối thêm 64 bit chưa biểu diễn nhị phân về độ dàI (ban đầu) của x(được rút gọn theo móulo $2^{64}$ nếu cần). Xâu kết quả M có độ dàI chia hết cho 512. Vì thế khi chặt M thành các từ 32 bit , số từ nhận được là N-sẽ chia hết cho 16.

Bây giờ, tiếp tục xây dựng bản tóm lược thông báo 128 bit. Hình 7.7 đưa ra mô tả thuật toán ở mức cao. Bản tóm lược thông báo được xây dựng như sự kết nối 4 từ A,B,C và D mà ta sẽ gọi là các thanh ghi. Bốn thanh ghi được khởi động như trong bước 1. Tiếp theo ta xử lí bảng M 16 bit từ cùng lúc. Trong mỗi vòng lặp ở bước 2, đầu tiên lấy 16 từ "tiếp theo" của M và lưu chúng trong bảng X (bước 3). Các giá trị của bốn thanh ghi dịch sau đó sẽ được lưu lại (bước 4). Sau đó ta sẽ thực hiện ba vòng "băm" (hash). Mỗi vaòng gồm một phép toán thực hiện trên một trong 16 từ trong X. Các phép toán được thực hiện trong ba vòng tạo ra các giá trị mới trong bốn thanh ghi. Cuối cùng ,bốn thanh ghi được update (cập nhật) trong bước 8 bằng cách cộng ngược các giá trị lưu trước đó trong bước 4. Phép cộng này được xác định là cộng các số nguyên dương ,được rút gọn theo modelo $2^{32}$.

Ba vòng trong MD4 là khác nhau (không giông như DES. 16 vòng đều như nhau). Trước hết ta sẽ mô tả vàI phép toán khác nhau trong ba vòng này. Trong phần sau,ta kí hiệu X và Y là các từ đầu vào và mỗi phép toán sẽ tạo ra một từ đầu ra. Dưới đây là phép toán được dùng:

X$\wedge$Y là phép "AND" theo bit giữa X và Y
X$\vee$Y là phép "OR" theo bit giữa X và Y
X$\oplus$Y là phép "XOR" theo bit giữa X và Y
$\neg$X chỉ phần bù của X
X+Y là phép cộng theo modulo $2^{32}$.
X<< s phép dịch vòng tráI X đI s vị trí (31>= s >=0).

Chú ý rằng, tất cả các phép toán trên đều tất nhanh và chỉ có phép số học duy nhất được dùng là phép cộng modulo $2^{32}$. Nếu MD4 được ứng dụng thì cần tính đến kiến trúc cơ bản của máy tính mà nó chạy trên đó để thực hiện

chính xác phép cộng. Giả sử $a_1a_2a_3a_4$ là 4 byte trong từ xem mỗi $a_i$,như một số nguyên trong dảI 0-255 được biểu diễn dưới dạng nhị phân. Trong kiến trúc kiểu endian lớn (chẳng hạn như trên trạm Sunsparc) từ này biểu diễn số nguyên.

$$a_1 2^{24} + a_2 2^{16} + a_3 2^8 + a_4$$

Trong kiến trúc kiểu endian nhỏ (chẳng hạn họ intel 80xxx). Từ này biểu diễn số nguyên:

$$a_4 2^{24} + + a_3\ 2^{16} + a_2\ 2^8 + a_1$$

MD4 giả thiết dùng kiến trúc kiểu endian nhỏ. ĐIều quan trọng là bản tóm lược thông báo độc lập với kiến trúc cơ bản. Vì thể nếu muốn chạy MD4 trên máy tính endian lớn cần thực hiện phép cộng X+Y như sau:

1. Trao đổi $x_1$ và $x_4$; $x_2$ và $x_3$; $y_1$ và $y_4$; $y_2$ và $y_3$
2. Tính $Z = X+Y \bmod 2^{32}$
3. Trao đổi $z_1$ và $z_4$ ; $z_2$ và $z_3$.

Hình 7.7 hàm hash MD4

1. A= 67452301 (hệ hexa)
   B = efcdab89 (hệ hexa)
   C = 98badcfe (hệ hexa)
   D = 10325476 (hệ hexa)
2. for i = 0 to N/16-1 do
3. for i = 1 to 15 do
   X[i] = M[16i+j]
4. AA = A
   BB = B
   CC = C
   DD = D
5. round1
6. round2
7. round3
8. A = A+AA
   B = B+ BB
   C = C + CC
   D = D + DD

Các vòng 1, 2 và 3 của MD4 dùng tương ứng ba hàm f, g, và h. Mỗi hàm này là một hàm boolean tính theo bit dùng 2 từ làm đầu vào và tạo ra một từ tại đẩu ra. Chúng được xác định như sau:

$f(X,Y,Z) = (X \wedge Y) \vee((-X) \wedge Z)$

$g(X,Y,Z) = (X \wedge Y) \vee(X \wedge Z) \vee(Y \wedge Z)$

$h(X,Y,Z) = X \oplus Y \oplus Z$

Các hình 7.8-7.10 sẽ mô tả đầy đủ các vòng 1,2 và 3 của MD4.

MD4 được thiết kế chạy rất nhanh và quả thực phần mềm chạy trên máy Sun SPARC có tốc độ 1.4 Mbyte/s. Mặt khác, khó có thể nói đIều gì cụ thể về độ mật của hàm hash, chẳng hạn như MD4 vì nó không dựa trên vàI toán khó đã nghiên cứu kĩ (ví dụ như phân tích nhân tử trên bàI toán logarithm rời rạc). Vì thế trong trường hợp Dé sự tin cậy vào độ an toàn của hệ thống chỉ có thể đạt được về thời gian và như vậy có thể hi vọng hệ thống vừa được nghiên cứu và không tìm thấy sự không an toàn nào.

Hình 7.8 : Vòng 1 của MD4 .(round 1)

```
1.   A = (A+ f(B,C,D) + X[0]) << 3
2.   D = (D + f(A,B,C) +X[1]) << 7
3.   C = (C + f(D,A,C) +X[2]) << 11
4.   B = (B + f(C,D,A) +X[3]) << 19
5.   A = (A + f(B,C,D) +X[4]) << 3
6.   D = (D + f(A,B,C) +X[5]) << 7
7.   C = (C + f(D,A,C) +X[6]) << 11
8.   B = (B + f(C,D,A) +X[7]) << 19
9.   A = (A + f(B,C,D) +X[8]) << 3
10.  D = (D + f(A,B,C) +X[9]) << 7
11.  C = (C + f(D,A,C) +X[10]) << 11
12.  B = (B + f(C,D,A) +X[11]) << 19
13.  A = (A + f(B,C,D) +X[12]) << 3
14.  D = (D + f(A,B,C) +X[13]) << 7
15.  C = (C + f(D,A,C) +X[14]) << 11
16.  B = (B + f(C,D,A) +X[15]) << 19
```

Mặc dù MD4 vẫn chưa bị phá song các phiên bản yếu cho phép bỏ qua hoặc vòng thứ nhất hoặc thứ ba dều có thể bị phá không khó khăn gì. nghĩa là dễ dàng tìn thấy các va chạm đối với các phiên bản chỉ có hai vòng. Phiên vản mạnh của MD5 là MD5 được công bố năm 1991. MD5 dùng vòng thay cho ba và chậm hơn 30% so với MD4 (khoảng 0.9 Mbyte/s trên cùng máy).

Chuẩn hàm hash an toàn phức tạp và chậm hơn. Ta sẽ không mô tả đầy đủ song sẽ chỉ ra một vàI cảI tiến trên nó.

1. SHS được thiết kế để chạy trên máy kiến trúc endian lớn hơn là trên máy endian nhỏ.
2. SHA tạo ra các bản tóm lược thông báo 5 thanh ghi (160 bit).
3. SHS xử lí 16 từ của bức đIện cùng một lúc như MD4. Tuy nhiên, 16 từ trước tiên được ”mở rộng” thành 80 từ ,sau đó thực hiện một dãy 80 phép tính ,mỗi phép tính trên một từ.
   Hình 7.9 Vòng 2 củaMD4.

```
1.   A = (A +g(B,C,D) + X[0] + 5A827999) <<3
2.   D = (D +g(A,B,C) + X[4] + 5A827999) <<5
3.   C = (C +g(D,A,B) + X[8] + 5A827999) <<9
4.   B = (B +g(C,D,A) + X[12] + 5A827999) <<13
5.   A = (A +g(B,C,D) + X[1] + 5A827999) <<3
6.   D = (D +g(A,B,C) + X[1] + 5A827999) <<5
7.   C = (C +g(D,A,B) + X[5] + 5A827999) <<9
8.   B = (B +g(C,D,A) + X[13] + 5A827999) <<13
9.   A = (A +g(B,C,D) + X[2] + 5A827999) <<3
10.  D = (D +g(A,B,C) + X[6] + 5A827999) <<5
11.  C = (C +g(D,A,B) + X[10] + 5A827999) <<9
12.  B = (B +g(C,D,A) + X[14] + 5A827999) <<13
13.  A = (A +g(B,C,D) + X[3] + 5A827999) <<3
14.  D = (D +g(A,B,C) + X[7] + 5A827999) <<5
15.  C = (C +g(D,A,B) + X[11] + 5A827999) <<9
16.  B = (B +g(C,D,A) + X[15] + 5A827999) <<13
```

Dùng hàm mở rộng sau đây: Cho trước 16 từ X[0] ...X[15], ta tính thêm 64 từ nữa theo quan hệ đệ quy.

X[j] = X[j-3]⊕ X[j-8]⊕ X[j-14]⊕ X[j-16], 79 ≥ j ≥ 16 7.1

Kết quả của phương trình (7.1) là mỗi một trong các từ X[16]...X[79] được thiết lập bằng cách cộng ⊕ với một tập con xác định nào đó của các từ X[0]...X[15].

Ví dụ: Ta có:

X[16] = X[0] ⊕X[2]⊕X[8]⊕X[13]
X[17] = X[1] ⊕X[3]⊕X[9]⊕X[14]
X[18] = X[2] ⊕X[4]⊕X[10]⊕X[15]
X[19] = X[0] ⊕X[2]⊕X[3]⊕X[5]⊕X[8]⊕X[11]⊕X[13]
X[79] = X[1] ⊕X[4]⊕X[15]⊕X[8]⊕X[12]⊕X[13].

Một đề xuất đòi hỏi sửa lại SHS liên quan đến hàm mở rộng trong đó đề nghị đặt lại phương trình 7.1 như sau:

X[j] = X[j-3] ⊕X[j-8]⊕X[j-14]⊕X[j-16] <<1; 79≥ j ≥ 16 (7.2)

Hình 7.10 : Vòng ba của MD4.

```
1.  A = (A + h(B,C,D) + X[0] + 6ED9EBA1) <<3
2.  D = (D + h(A,B,C) + X[8] + 6ED9EBA1) <<9
3.  C = (C + h(D,A,B) + X[4] + 6ED9EBA1) << 11
4.  B = (B + h(C,D,A) + X[12] + 6ED9EBA1) << 15
5.  A = (A + h(B,C,D) + X[2] + 6ED9EBA1) <<3
6.  D = (D + h(A,B,C) + X[10] + 6ED9EBA1) <<9
7.  C = (C + h(D,A,B) + X[6] + 6ED9EBA1) << 11
8.  B = (B + h(C,D,A) + X[14] + 6ED9EBA1) << 15
9.  A = (A + h(B,C,D) + X[1] + 6ED9EBA1) <<3
10. D = (D + h(A,B,C) + X[9] + 6ED9EBA1) <<9
11. C = (C + h(D,A,B) + X[13] + 6ED9EBA1) << 11
12. B = (B + h(C,D,A) + X[13] + 6ED9EBA1) << 15
13. A = (A + h(B,C,D) + X[3] + 6ED9EBA1) <<3
14. D = (D + h(A,B,C) + X[11] + 6ED9EBA1) <<9
15. C = (C + h(D,A,B) + X[7] + 6ED9EBA1) << 11
16. B = (B + h(C,D,A) + X[15] + 6ED9EBA1) << 15
```

Như trước đây , toán tử ''<<1'' là phép dịch vòng trái một vị trí.

---

## 7.8 **nhãn thời gian (timestamping)**.

Một khó khăn trong sơ đồ chữ kí là thuật toán kí có thể bị tổn thương. Chẳng hạn, giả sử Oscar có khả năng xác định số mũ mật a của Bob trên bất kì bức điện nào mà anh ta muốn. Song còn vấn đề khác (có thể nghiêm trọng hơn )là : từ đây người ta sẽ đặt câu hỏi về tính xác thực của tất cả các bức điện mà Bob kí, kể cả những bức điện mà anh ta kí trước khi Oscar đánh cắp được thuật toán.

Từ đây lại có thể nảy sinh tình huống không mong muốn khác : giả sử Bob kí một bức điện và sau đó từ chối là đã không kí nó. Bob có thể công khai thuật toán kí của mình sau đó công bố rằng chữ kí của anh ta trên bức điện đang nói trên là giả mạo.

Lí do có các kiểu sự kiện này là do không có các nào các định bức điện được kí khi nào. Nhãn thời gian có thể cung cấp bằng chứng rằng, bức điện đã được kí vào thời điểm cụ thể nào đó. Khì đó nế thuật toán kí của Bob có nhược điểm (bị tổn thương) thì bất kì chữ kí nào anh ta kí trước đó sẽ không còn hợp lệ. ĐIều này giống với kiểu thực hiên các thẻ tín dụng: Nếu ai đó làm mất thẻ tín dụng và thông báo cho nhà băng đã phát hành thì thẻ mất hiệu lực. Song các cuộc mua bán thực hiện trước khi mất nó thì vẫn không bị ảnh hưởng.

Trong phần này sẽ mô tả một vàI phương pháp gắn nhãn thời gian. Trước hết,nhận xét rằng, Bob có thể tạo ra một nhãn thời gian có sức thuyết phục trên chữ kí của anh ta. Đầu tiên, Bob nhận được một thông tin "hiện thời " có sẵn công khai nào đó, thông tin này không thể dự đoán được trước khi nó xảy ra. Ví dụ thông tin chứa tất cả các lợi thế về môn bóng chày của các liên minh chính từ ngày trước đó, hay các giá trị của tất cả cổ phần đwocj lên danh sách trong sở giao dịch chứng khoán NewYork. Ta kí hiệu thông tin này bằng chữ *pub*.

Bây giờ giả sử Bob muốn dán nhãn thời gian trên chữ kí của mình trên bức điện x. Giả thiết rằng, h là hàm hash công khai biết trước. Bob sẽ thực hiện theo thuật toán trong hình 7.11.Sau đây là cách sơ đồ làm việc : sự có mặt của thông tin pub có nghĩa là bob không thể tạo ra được y trước ngày đang nói đến. Còn một thực tế là y công bố trong một tờ báo ra ngày tiếp theo chứng tỏ

rằng bob đã không tính y sau ngày được nói đến. Vì thế chữ kí y của bob bị hạn chế trong thời hạn một ngày. Cũng nhận xét thấy rằng, bob không để lộ bức điện x trong sơ đồ này vì chỉ có x được công bố ... Nếu cần bob có thể chứng minh rằng x là bức điện mà anh ta đã kí và dán nhãn thời gian một cách đơn giản là làm lộ nó.

Cũng không khó khăn tạo ra tạo ra các nhãn thời gian nếu có một cơ quan dịch vụ dán nhãn đáng tin cậy. Bob có thể tính $z = h(x)$ và $y = sig_k(z)$ và sau đó gửi (z và x ) đến cơ quan làm dịch vụ dán nhãn thời gian (TSS). TSS sau đó sẽ gắn ngày D và kí (đánh dấu)bộ ba (z,y,D). Công việc này sẽ hoàn hảo miễn là thuật toán kí của TSS an toàn và TSS không thể bị mua chuộc để lùi ngày dãn nhãn của thời gian. (chú ý rằng phương pháp này chỉ được thiết lập khi bob đã kí một bức điện trước một thời gian nào đó. Nếu bob muốn thiết lập cáI anh ta đã kí nó sau ngày nào đó ,anh ta có thể kết hợp thông tin công khai pub nào đó như phương pháp trước đó).

Hình 7.11 :Dán nhãn thời gian lên chữ kí trên bức điện x.

1. Bob tính $z = h(x)$.
2. Bob tính $z' = h(z \| pub)$.
3. Bob tính $y = sig_k(z')$.
4. Bob công bố (z,pub,y) trên tờ báo ra ngày hôm sau.

Nếu như không muốn tin vô điều kiện vào TSS thì có thể tăng độ an toàn lên bằng cách liên kết các thông báo đã dán nhãn thời gian. Trong sơ độ như vậy, bob sẽ gửi một bộ ba được xếp thứ tự (z,x,ID)(Bob) cho TSS. ở đây, z là bản tóm lược thông báo của bức điện x,y là chữ kí của bob trên z ,còn ID(Bob) là thông tin định danh của Bob. TSS sẽ dãn nhãn thời gian một chuỗi bộ ba có dạng này. Kí hiệu $(z_n,y_n,ID_n)$ là bộ ba thứ tự n được TSS dán nhãn thời gian và cho $t_n$ là kí hiệu thời gian lúc thực hiện yêu cầu thứ n.

Hình 7.12: Dán nhãn thời gian $(z_n,y_n,ID_n)$.

1. TSS tính $L_n = (t_{n-1},ID_{n-1},Z_{n-1}y_{n-1},h(L_{n-1}))$
2. TSS tính $C_n = (n, t_n, z_n, ID_n, L_n)$
3. TSS tính $S_n = sig_{TSS}(h(C_n))$
4. TSS gửi $(C_n, S_n, ID_n)$ cho $ID_n$.

TSS sẽ dán nhãn thời gian lên bộ ba thứ n bằng fthuật toán nêu trên hình 7.12. $L_n$ là "thông tin liên kết để nối yêu cầu thứ n vào yêu cầu trước đó. ($L_0$ được chọn làm thông tin gia nào đó (được xác định trước đây)để quá trình được bắt đầu)".

Bây giờ nếu được yêu cầu (challenge). Bob có thể để lộ bức điện $x_n$ của mình, và sau đó có thể xác minh $y_n$. Tiếp theo , các minh chữ kí $s_n$ của TSS. Nếu muốn thì có thể đòi $ID_{n-1}$ hoặc $ID_{n+1}$ để tạo ra nhãn thời gian ($C_{n-1}$, $S_{n-1}$, $ID_n$) và ($C_{n+1}$, $S_{n+1}$, $ID_{n+2}$) tương ứng của chúng. Các chữ kí của TSS có thể được kiểm tra theo nhãn thời gian này. Dĩ nhiên , quá trình này có thể tiếp tục tới mức mong muốn, trước hay sau đó.

---

## 7.9.CÁC CHÚ Ý VỀ TÀI LIỆU DẪN

Hàm hash log rời rạc được mô tả trong mục 7.4 là của chaum , van heijst và pfitzmann [CvHP92]. Còn hàm hash có thể chứng mình đwocj là an toàn liễn là hợp số n không thể phân tích thành nhân tư là do gibson [Gib91] đưa ra (bài tập 7.4 có mô tả sơ đồ này).

Cơ sở cho việc m mở rộng hàm hash trong mục 7.5 là của Damgard [DA90] Merkle cũng đưa ra các phương pháp tương tự [ME90].

Các thông tin liên qua tới việc xây dựng các hàm hash dựa trên các hệ thông mã khoá bí mật. Bạn đọc có thể xem trong [PGV94] của Preneel, Govaerts và Vandewalle.

Thuật toán MD4 được đưa ra trong [Ri91] của Rivest, còn tiêu chuẩn hash an toàn được mô tả trong [NBS93]. Tấn công hai trong ba vòng MD4 là của Boer và Bossalaer [DBB92]. Các hàm hash gần đây kể cả N-hash là của [MOI90] và Snefru [ME90A].

Ngoài ra có thể tìm thấy tổng quan về kĩ thuật băm trong Preneel, Govaerts, Vandewalle [PGV93].

**Bài tập**

7.1. Giả sử h: $X \to Y$ là hàm hash. Với y bất kỳ $\in Y$, cho:

$h^{-1}(y) = \{ x: h(x) = y\}$

và ký hiệu $s_y = | h^{-1}(y)|$.

Định nghĩa N =

*CHƯƠNG 6*

# CÁC SƠ ĐỒ CHỮ KÍ SỐ

## 6.1 GIỚI THIỆU.

Trong chương này, chúng ta xem xét các sơ đồ chữ kí số (còn được gọi là chữ kí số ). Chữ kí viết tay thông thường trên tàI liệu thường được dùng để xác người kí nó. Chữ kí được dùng hàng ngày chẳng hạn như trên một bức thư nhận tiền từ nhà băng, kí hợp đồng...

Sơ đồ chữ kí là phương pháp kí một bức điện lưu dưới dang điện từ. Chẳng hạn một bức điện có ký hiệu được truyền trên mạng máy tinh. Chương trình này nghiên cứu vài sơ đồ chữ kí. Ta sẽ thảo luận trên một vàI khác biệt cơ bản giửa các chữ kí thông thường và chữ kí số.

Đầu tiên là một vấn đề kí một tài liệu. Với chữ kí thông thường, nó là một phần vật lý của tài liệu. Tuy nhiên, một chữ kí số không gắn theo kiểu vật lý vào bức điện nên thuật toán được dùng phải "không nhìn thấy" theo cách nào đó trên bức điện.

Thứ hai là vấn đề về kiểm tra. Chữ kí thông thường được kiểm tra bằng cách so sánh nó với các chữ kí xác thực khác. ví dụ, ai đó kí một tấm séc để mua hàng, người bán phải so sánh chữ kí trên mảnh giấy với chữ kí nằm ở mặt sau của thẻ tín dụng để kiểm tra. Dĩ nhiên, đây không phải là phương pháp an toàn vì nó dể dàng giả mạo. Mắt khác, các chữ kí số có thể được kiểm tra nhờ dùng một thuật toán kiểm tra công khai. Như vậy, bất kỳ ai cũng có thể kiểm tra dược chữ kí số. Việc dùng một sơ đồ chữ kí an toàn có thể sẽ ngăn chặn dược khả năng giả mạo.

Sự khác biệt cơ bản khác giữa chữ kí số và chữ kí thông thường bản copy tài liệu được kí băng chữ kí số đồng nhất với bản gốc, còn copy tài liệu có chữ kí trên giấy thường có thể khác với bản gốc. Điều này có nghĩa là phải cẩn thận ngăn chăn một bức kí số khỏi bị dung lạI. Ví dụ, Bob kí một bức điện xác nhận Alice có khả năng làm điều đó một lần. Vì thế, bản thân bức điện cần chứa thông tin (chẳng hạn như ngày tháng) để ngăn nó khỏi bị dung lại.

Một sơ đồ chữ kí số thường chứa hai thành phần: thuật toán kí và thuật toán xác minh. Bob có thể kí điện x dùng thuật toán kí an toàn. Chữ kí sig(x) nhận được có thể kiểm tra bằng thuật toán xác minh công khai ver. Khi cho trước cặp (x,y), thuật toán xác minh có giá trị TRUE hay FALSE tuỳ thuộc vào chữ kí được thực như thế nào. Dưới đây là định nghĩa hình thức của chữ kí:

**Định nghĩa 6.1**

Một sơ đồ chữ kí số là bộ 5( $\mathcal{P}$,$\mathcal{A}$, $\mathcal{K}$,$\mathcal{S}$,$\mathcal{V}$) thoả mãn các điều kiện dưới đây:

1. $\mathcal{P}$ là tập hữu hạn các bức điện có thể.
2. $\mathcal{A}$ là tập hữu hạn các chữ kí có thể.
3. $\mathcal{K}$ không gian khoá là tập hữu hạn các khoá có thể.
4. Với mỗi K thuộc $\mathcal{K}$ tồn tạI một thuật toán kí $sig_k \in \mathcal{S}$ và là một thuật toán xác minh $ver_k \in \mathcal{V}$. Mỗi $sig_k : \mathcal{P} \to \mathcal{A}$ và $ver_k: \mathcal{P}\times\mathcal{A} \to \{true,false\}$ là những hàm sao cho mỗi bức điện $x \in \mathcal{P}$ và mối chữ kí $y \in \mathcal{A}$ thoả mãn phương trình dưới đây.

$$ver_k \begin{cases} \text{True nếu } y=sig(x) \\ \\ \text{False nếu } y \# sig(x) \end{cases}$$

Với mỗi k thuộc $\mathcal{K}$ hàm $sig_k$ và $ver_k$ là các hàm thơì than đa thức. $Ver_k$ sẽ là hàm công khai $sig_k$ là mật. Không thể dể dàng tính toán để giả mạo chữ kí của Bob trên bức điện x. Nghĩa là x cho trước, chỉ có Bob mới có thể tính được y để $ver_k$ = True. Một sơ đồ chữ kí không thể an toàn vô điều kiện vì Oscar có thể kiểm tra tất cả các chữ số y có thể có trên bức điện x nhờ dùng thuật toán ver công khai cho đến khi anh ta tìm thấy một chữ kí đúng. Vi thế, nếu có đủ thời gian. Oscar luôn luôn có thể giả mạo chữ kí của Bob. Như vậy, giống như trường hợp hệ thống mã khoá công khai, mục đích của chúng ta là tìm các sơ đồ chữ kí số an toan về mặt tính toán.

Xem thấy rằng, hệ thống mã khoá công khai RSA có thể dùng làm sơ đồ chữ kí số, Xem hình 6.1.

Như vậy, Bob kí bức điện x dùng qui tắc giải mã RSA là $d_k$,. Bob là người tạo ra chữ kí vì $d_k = sig_k$ là mật. Thuật toán xác minh dùng qui tắc mã RSA $e_k$. Bất kì ai củng có xác minh chữ kí vi $e_k$ được công khai.

Chú ý rằng, ai đó có thể giả mạo chữ kí của Bob trên một bức điện " ngẩu nhiên" x bằng cách tìm $x=e_k(y)$ với y nào đó; khi đó $y= sig_k(x)$. Một pháp xung quanh vấn đề khó khăn này là yêu cầu bức điện chưa đủ phần dư để chữ kí giả mạo kiểu này không tương ứng với bức điện đây nghĩa là x trừ một xác suất rất bé. Có thể dùng các hàm hash trong việc kết nối với các sơ đồ chữ kí số sẽ loại trừ được phương pháp giả mạo này (các hàm hash được xét trong chương 7).

Hình 6.1 sơ đồ chữ kí RSA

Cho n= pq, p và q là các số nguyên tố. Cho $\mathcal{P} = \mathcal{A} = Z_n$ và định nghĩa $\mathcal{P}$= {(n,p,q,a,b):=n=pq,p và q là nguyên tố, $ab \equiv 1(\mathrm{mod}(\phi(n)))$ }. Các giá trị n và b là công khai, ta định nghĩa :

$sig_k(x) = x^a \bmod n$

và $ver_k(x,y) = \mathrm{true} \Leftrightarrow x \equiv y^b \pmod n$

$(x,y \in Z_n)$

Cuối cùng, ta xét tóm tắt các kết hợp chữ kí và mã khoá công khai. Giả sử rằng, Alice tính toán chư kí của ta y= $sig_{Alice}(x)$ và sau đó mã cả x và y bằng hàm mã khoá công khai $e_{Bob}$ của Bob, khi đó cô ta nhận được $z = e_{Bob}(x,y)$. Bản mã z sẽ được truyền tới Bob. Khi Bob nhận được z, anh ta sẽ trước hết sẽ giải mã hàm $d_{Bob}$ để nhận được (x,y). Sau đó anh ta dung hàm xác minh công khai của Alice để kiểm tra xem $ver_{Alice}(x,y)$ có bằng True hay không.

Song nếu đầu tiên Alice mã x rồi sau đó mới kí tên bản mã nhận được thì sao?. Khi đó cô tính :

$$y = sig_{Alice}(e_{Bob}(x)).$$

Alice sẽ truyền cặp (z,y) tới Bob. Bob sẽ giải mã z, nhận x và sau đó xác minh chữ kí y trên x nhờ dùng $ver_{Alice}$. Một vấn đề tiềm ẩn trong biện pháp này là nếu Oscar nhận được cặp (x,y) kiểu này, được ta có thay chữ kí y của Alice bằng chữ kí của mình.

$$y' = sig_{Oscar}(e_{Bob}(x)).$$

(chú ý rằng,Oscar có thể kí bản mã $e_{Bob}(x)$ ngay cả khi anh ta không biết bản rõ x). Khi đó nếu Oscar truyền(x, y ) đến Bob thì chữ kí Oscar được Bob xác minh bằng $ver_{Oscar}$ và Bob có thể suy ra rằng, bản rõ x xuất phát từ Oscar. Do khó khăn này, hầu hết người sử dụng được khuyến nghị nếu kí trước khi mã.

---

## 6.2 SƠ ĐỒ CHỮ KÍ ELGAMAL

Sau đây ta sẽ mô tả sơ đồ chữ kí Elgamal đã từng dưới thiệu trong bài báo năm 1985. Bản cả tiến của sơ đồ này đã được Viện Tiêu chuẩn và Công Nghệ Quốc Gia Mỹ (NIST) chấp nhận làm chữ kí số. Sơ đồ Elgamal (E.) được

thiết kế với mục đích dành riêng cho chữ kí số, khác sơ đồ RSA dùng cho cả hệ thống mã khoá công khai lẫn chữ kí số.

Sơ đồ E, là không tất định giống như hệ thống mã khoá công khai Elgamal. Điều này có nghĩa là có nhiều chữ kí hợp lệ trên bức điện cho trước bất kỳ. Thuật toán xác minh phải cố khải năng chấp nhận bất kì chữ kí hợp lệ khi xác thực. Sơ đồ E. được môt tả trên hình 6.2

Nếu chữ kí được thiết lập đúng khi xác minh sẽ thành công vì :

$$\beta^{\gamma}\gamma^{\delta} \equiv \alpha^{a\gamma}\alpha^{k\gamma} (\text{mod } p)$$
$$\equiv \alpha^{x} (\text{mod } p)$$

là ở đây ta dùng hệ thức :

$$a\gamma + k\delta \equiv x \ (\text{mod } p-1)$$

Hình 6.2 sơ đồ chữ kí số Elgamal.

Cho p là số nguyên tố sao cho bài toán log rời rạc trên $Z_p$ là khó và giả sử $\alpha \in Z_n$ là phần tử nguyên thuỷ $\mathcal{P} = Z_p^*$ , $\mathcal{A} = Z_p^* \times Z_{p-1}$ và định nghĩa :

$$\mathcal{K} = \{(p,\alpha,a,\beta) : \beta \equiv \alpha^a (\text{mod } p)\}.$$

Giá trị $p,\alpha,\beta$ là công khai, còn a là mật.

Với $\mathcal{K} = (p,\alpha,a,\beta)$ và một số ngẫu nhiên (mật) $k \in Z_{p-1}$. định nghĩa :

$$Sig_k(x,y) = (\gamma,\delta),$$

trong đó $\gamma = \alpha^k \bmod p$

và $\delta = (x-a)\, k^{-1} \bmod (p-1)$

Với $x,\gamma \in Z_p$ và $\delta \in Z_{p-1}$ , ta định nghĩa :

$$Ver(x,\gamma,\delta) = \text{true} \Leftrightarrow \beta^{\gamma}\gamma^{\delta} \equiv \alpha^{x} (\text{mod } p).$$

Bob tính chữ kí bằng cách dùng cả gía trị mật a (là một phần của khoá) lẫn số ngẫu nhiên mật k (dùng để kí lên bức điện x ). Việc xác minh có thực hiện duy nhất bằng thông báo tin công khai.

Chúng ta hãy xét một ví dụ nhỏ minh hoạ.

Ví dụ 6.1

Giả sử cho p = 467, $\alpha$ =2,a = 127; khi đó:

$\beta = \alpha^a \bmod p$
$= 2^{127} \bmod 467$
$= 132$

Nếu Bob muốn kí lên bức điện x = 100 và chọn số ngẫu nhiên k =213 (chú ý là UCLN(213,466) =1 và $213^{-1} \bmod 466 = 431$. Khi đó

$\gamma = 2^{213} \bmod 467 = 29$

và $\delta = (100-127 \times 29)\ 431 \bmod 466 = 51.$

Bất kỳ ai củng có thể xác minh chữ kí bằng các kiểm tra :

$$132^{29}\ 29^{51} \equiv 189 \pmod{467}$$

và

$$2^{100} \equiv 189 \pmod{467}$$

Vì thế chữ kí là hợp lệ.

Xét độ mật của sơ đồ chữ kí E. Giả sử, Oscar thử giả mạo chữ kí trên bức điện x cho trước không biết a. Nếu Oscar chọn $\gamma$ và sau đó thử tìm giá trị $\delta$ tương ứng, anh ta phải tính logarithm rời rạc $\log_\gamma \alpha^x\beta^{-\gamma}$. Mặt khác, nếu đầu tiên ta chọn $\delta$ và sau đó thử tim $\gamma$ và thử giải phương trình:

$$\beta^\gamma \gamma^\delta \equiv \alpha^x \pmod p.$$

để tìm $\gamma$. Đây là bài toán chưa có lời giải nào: Tuy nhiên, dường như nó chưa được gắn với đến bài toán đã nghiên cứu kĩ nào nên vẫn có khả năng có cách nào đó để tính $\delta$ và $\gamma$ đồng thời để ($\delta$, $\gamma$)là một chữ kí. Hiện thời không ai tìm được cách giải song củng ai không khẳng định được rằng nó không thể giải được.

Nếu Oscar chọn $\delta$ và $\gamma$ và sau đó tự giải tìm x, anh ta sẽ phảI đối mặt với bài toán logarithm rời rạc, tức bài toán tính $\log_\alpha$ ??? Vì thế Oscar không thể kí một bức điện ngẫu nhiên bằng biện pháp này. Tuy nhiên, có một cách để Oscar có thể kí lên bức điện ngẫu nhiên bằng việc chọn $\gamma$, $\delta$ và x đồng thời: giả thiết i và j là các số nguyên $0 \le i \le p-2$, $0 \le j \le p-2$ và UCLN(j,p-2) = 1. Khi đó thực hiện các tính toán sau:

$\gamma = \alpha^i \beta^j \bmod p$
$\delta = -\gamma\, j^{-1} \bmod (p-1)$
$x = -\gamma\, i\, j^{-1} \bmod (p-1)$

trong đó $j^{-1}$ được tính theo modulo (p-1) (ở đây đòi hỏi j nguyên tố cùng nhau với p-1).

Ta nói rằng (γ, δ )là chữ kí hợp lệ của x. Điều này được chứng minh qua việc kiểm tra xác minh :

????

Ta sẽ minh hoạ bằng một ví dụ :

Ví dụ 6.2.

Giống như ví dụ trước cho p = 467, α = 2, β =132. Giả sữ Oscar chọn i = 99,j = 179; khi đó $j^{-1}$ mod (p-1) = 151. Anh ta tính toán như sau:

$\gamma = 2^{99}132^{197} \bmod 467 = 117$

$\delta = -117 \times 151 \bmod 466 = 51.$

$x = 99 \times 41 \bmod 466 = 331$

Khi đó (117, 41) là chữ kí hợp lệ trên bức điện 331 hư thế đã xác minh qua phép kiểm tra sau:

$132^{117}\ 117^{41} \equiv 303 \pmod{467}$

và $2^{331} \equiv 303 \pmod{467}$

Vì thế chữ kí là hợp lệ.

Sau đây là kiểu giả mạo thứ hai trong đó Oscar bắt đầu bằng bức điện được Bob kí trước đây. Giả sử (γ, δ ) là chữ kí hợp lệ trên x. Khi đó Oscar có khả năng kí lên nhiều bức điện khác nhau. Giả sử i, j, h là các số nguyên, $0 \le h, i, j \le p-2$ và UCLN (h γ - j δ, p-1) = 1. Ta thực hiện tính toán sau:

$\lambda = \gamma^h \alpha^i \beta^j \bmod p$

$\mu = \delta\lambda(h\gamma - j\delta)^{-1} \bmod (p-1)$

$x' = \lambda(hx + i\delta)^{-1} \bmod (p-1),$

trong đó $(h\gamma - j\delta)^{-1}$ được tính theo modulo (p-1). Khi đó dễ dàng kiểm tra điệu kiện xác minh :

$\beta^{\lambda} \lambda^{\mu} \equiv \alpha^{x'} \pmod{p}$

vì thế (λ, μ)là chữ kí hợp lệ của x'.

Cả hai phương pháp trên đều tạo các chữ kí giả mạo hợp lệ song không xuất hiện khả năng đối phương giả mạo chữ kí trên bức điện có sự lựu chọn của chính họ mà không phải giải bài toán logarithm rời rạc, vì thế không có gì nguy hiểm về độ an toàn của sơ đồ chữ kí Elgamal.

Cuối cùng, ta sẽ nêu vài cách có thể phái được sơ đồ này nếu không áp dụng nó một cách cẩn thận (có một số ví dụ nữa về khiếm khuyết của giao thức, một số trong đó là xét trong chương 4). Trước hết, giá trị k ngẫu nhiên được dùng để tính chữ kí phải giữ kín không để lộ. vì nếu k bị lộ, khá đơn giản để tính :

$$A = (x\text{-}k\,\gamma)\delta^{-1} \bmod (p\text{-}1).$$

Dĩ nhiên, một khi a bị lộ thì hệ thống bị phá và Oscar có thể dễ dang giả mạo chữ kí.

Một kiểu dung sai sơ đồ nữa là dùng cùng giá trị k để kí hai bức điện khác nhau. điều này cùng tạo thuận lợi cho Oscar tinh a và phá hệ thống. Sau đây là cách thực hiện. Giả sử $(\gamma, \delta_1)$ là chữ kí trên $x_1$ và $(\gamma, \delta_2)$ là chữ kí trên $x_2$. Khi đó ta có:

$$\beta^{\gamma}\gamma^{\delta_1} \equiv \alpha^{x_1} \pmod p$$

và

$$\beta^{\gamma}\gamma^{\delta_2} \equiv \alpha^{x_2} (\text{mod} p).$$

Như vậy

$$\alpha^{x_1-x_2} \equiv \alpha^{\delta_1-\delta_2} \pmod p.$$

Nếu viết $\gamma = \alpha^k$, ta nhận được phương trình tìm k chưa biết sau.

$$\alpha^{x_1-x_2} \equiv \alpha^{k(\delta_1-\delta_2)} \pmod p$$

tương đương với phương trình

$$x_1 - x_2 \equiv k(\delta_1 - \delta_2) \pmod{p-1}.$$

Bây giờ giả sử d =UCLN$(\delta_1 - \delta_2, p-1)$. Vì d | (p-1) và d | $(\delta_1-\delta_2)$ nên suy ra d | $(x_1-x_2)$. Ta định nghĩa:

$$x' = (x_1 - x_2)/d$$
$$\delta' = (\delta_1 - \delta_2)/d$$
$$p' = (p-1)/d$$

Khi đó đồngdư thức trở thành:

$$x' \equiv k\,\delta' \pmod{p'}$$

vì UCLN$(\delta', p') = 1$,nên có thể tính:

$$\varepsilon = (\delta')^{-1} \bmod p'$$

Khi đó giá trị k xác định theo modulo $p'$ sẽ là:

$$k = x' \varepsilon \bmod p'$$

Phương trình này cho d giá trị có thể của k

$$k = x' \varepsilon + i\, p' \bmod p$$

với i nào đó, $0 \le i \le d-1$. Trong số d giá trị có có thế này, có thể xác định được một giá trị đúng duy nhất qua việc kiểm tra điều kiện

$$\gamma \equiv \alpha^k \pmod p$$

---

## 6.3 CHUẨN CHỮ KÍ SỐ.

Chuẩn chữ kí số(DSS) là phiên bản cải tiến của sơ đồ chữ kí Elgamal. Nó được công bố trong Hồ Sơ trong liên bang vào ngày 19/5/94 và được làm chuẩn vào 1/12/94 tuy đã được đề xuất từ 8/91. Trước hết ta sẽ nêu ra những thay đổi của nó so với sơ đồ Elgamal và sau đó sẽ mô tả cách thực hiện nó.

Trong nhiều tinh huống, thông báo có thể mã và giải mã chỉ một lần nên nó phù hợp cho việc dùng với hệ mật Bất kì (an toàn tại thời điểm được mã). Song trên thực tế, nhiều khi một bức điện được dùng làm một tài liệu đối chứng, chẳng hạn như bản hợp đồng hay một chúc thư và vì thế cần xác minh chữ kí sau nhiều năm kể từ lúc bức điện được kí. Bởi vậy, điều quan trọng là có phương án dự phòng liên quan đến sự an toàn của sơ đồ chữ kí khi đối mặt với hệ thống mã. Vì sơ đồ Elgamal không an toàn hơn bài toán logarithm rời rạc nên cần dung modulo p lớn. Chắc chắn p cần ít nhất là 512 bít và nhiều người nhất trí là p nên lấy p=1024 bít để có độ an toàn tốt.

Tuy nhiên, khi chỉ lấy modulo p =512 thì chữ kí sẽ có 1024 bít. Đối với nhiều ứng dụng dùng thẻ thông minh thì cần lại có chữ kí ngắn hơn. DSS cải tiến sơ đồ Elgamal theo hướng sao cho một bức điện 160 bít được kí bằng chữ kí 302 bít song lại p = 512 bít. Khi đó hệ thống làm việc trong nhóm con $Z_n^*$ kích thước $2^{160}$. Độ mật của hệ thống dựa trên sự an toàn của việc tìm các logarithm rời rạc trong nhóm con $Z_n^*$.

Sự thay đổi đầu tiên là thay dấu " - " bằng "+" trong định nghĩa δ, vì thế:

$$\delta = (x + \alpha\, \gamma)k^{-1} \bmod (p-1)$$

thay đổi kéo theo thay đổi điều kiện xác minh như sau:

$$\alpha^x \beta^\gamma \equiv \gamma^\delta \pmod p \quad (6.1)$$

Nếu UCLN $(x + \alpha\gamma, p-1) = 1$ thì $\delta^{-1} \bmod (p-1)$ tồn tại và ta có thể thay đổi điều kiện (6.1) như sau:

$$\alpha^{x\delta^{-1}} \beta^{\gamma\delta^{-1}} \equiv \gamma \pmod p \quad (6.2)$$

Đây là thay đổi chủ yếu trong DSS. Giả sử q là số nguyên tố 160 bít sao cho $q \mid (q-1)$ và $\alpha$ là căn bậc q của một modulo p. (Dễ dàng xây dựng một $\alpha$ như vậy: cho $\alpha_0$ là phần tử nguyên thuỷ của $Z_p$ và định nghĩa $\alpha = \alpha_0^{(p-1)/q} \bmod p$).

Khi đó $\beta$ và $\gamma$ cũng sẽ là căn bậc q của 1. vì thế các số mũ Bất kỳ của $\alpha$, $\beta$ và $\gamma$ có thể rút gọn theo modulo q mà không ảnh hưởng đến điều kiện xác minh (6.2). Điều rắc rối ở đây là $\gamma$ xuất hiện dưới dạng số mũ ở vế trái của (6.2) song không như vậy ở vế phải. Vì thế, nếu $\gamma$ rút gọn theo modulo q thì cũng phải rút gọn toàn bộ vế trái của (6.2) theo modulo q để thực hiện phép kiểm tra. Nhận xét rằng, sơ đồ (6.1) sẽ không làm việc nếu thực hiện rút gọn theo modulo q trên (6.1). DSS được mô tả đầy đủ trong hinh 6.3.

Chú ý cần có $\delta \not\equiv 0 \pmod q$ vì giá trị $\delta^{-1} \bmod q$ cần thiết để xác minh chữ kí (điều này tương với yêu cầu UCLN$(\delta, p-1) = 1$ khi biến đổi (6.1) thành (6.2). Nếu Bob tính $\delta \equiv 0 \pmod q$ theo thuật toán chữ kí, anh ta sẽ loại đi và xây dựng chữ kí mới với số ngẫu nhiên k mới. Cần chỉ ra rằng, điều này có thể không gần vấn đề trên thực tế: xác xuất để $\delta \equiv 0 \pmod q$ chắc sẽ xảy ra cở $2^{-160}$ nên nó sẽ hầu như không bao giờ xảy ra.

Dưới đây là một ví dụ minh hoạ nhỏ

Hình 6.3. Chuẩn chữ kí số.

Giả sử p là số nguyên tố 512 bít sao cho bài toán logarithm rời rạc trong $Z_p$ khong Giải được, cho p là số nguyên tố 160 bít là ước của (p-1). Giả thiết $\alpha \in Z_p$ là căn bậc q của 1modulo p: Cho $\mathcal{P} = Z_p$ . $\mathcal{A} = Z_q \times Z_p$ và định nghĩa :

$$\mathcal{A} = \{(p,q,\alpha,a,\beta) : \beta \equiv \alpha^a \pmod p\}$$

các số p, q, $\alpha$ và $\beta$ là công khai, có a mật.

Với K = (p,q,$\alpha$ ,a,$\beta$ )và với một số ngẫu nhiên (mật) k ,$1 \le k \le q-1$, ta định nghĩa:

$$sig_k(x,k) = (\gamma, \delta)$$

trong đó $\gamma = (\alpha^k \bmod p) \bmod q$

và $\delta = (x + a\gamma)k^{-1} \bmod q$

Với $x \in Z_p$ và $\gamma, \delta \in Z_q$ , qua trình xác minh sẽ hoàn toàn sau các tính toán :

$$e_1 = x\delta^{-1} \bmod q$$

$$e_2 = \gamma\delta^{-1} \bmod q$$

$$ver_k(x, \gamma, \delta) = true \Leftrightarrow (\alpha^{e_1}\beta^{e_2} \bmod p) \bmod q = \gamma$$

Ví dụ 6.3:

Giả sử q =101, p = 78 q+1 =7879.3 là phần tử nguyên thuỷ trong $Z_{7879}$ nên ta có thể lấy: $\alpha = 3^{78} \bmod 7879 = 170$

Giả sử a =75, khi đó :

$$\beta = \alpha^a \bmod 7879 = 4576$$

Bây giờ giả sữ Bob muốn kí bức điện x = 1234 và anh ta chọn số ngẫu nhiên k =50, vì thế :

$$k^{-1} \bmod 101 = 99$$

khi đó $\gamma = (170^{30} \bmod 7879) \bmod 101$

$= 2518 \bmod 101$

$= 94$

và $\delta = (1234 + 75 \times 94) \bmod 101$

$= 96$

Chữ kí (94, 97) trên bức điện 1234 được xác minh bằng các tính toán sau:

$$\delta^{-1} = 97^{-1} \bmod 101 = 25$$

$$e_1 = 1234 \times 25 \bmod 101 = 45$$
$$e_2 = 94 \times 25 \bmod 101 = 27$$
$$(170^{45}\ 4567^{27} \bmod 7879) \bmod = 2518 \bmod 101 = 94$$

vì thế chữ kí hợp lệ.

Khi DSS được đề xuất năm 1991, đã có một vài chỉ trích đưa ra. Một ý kiến cho rằng, việc xử lý lựa chọn của NIST là không công khai. Tiêu chuẩn đã được Cục An ninh Quốc gia (NSA) phát triển mà không có sự tham gia của khôi công nghiệp Mỹ. Bất chấp những ưu thế của sơ đồ, nhiều người đã đóng chặt cửa không tiếp nhận.

Còn những chỉ trích về mặt kĩ thuật thì chủ yếu là về kích thước modulo p bị cố định = 512 bít. Nhiều người muốn kích thước này có thể thay đổi được nếu cần, có thể dùng kích cỡ lớn hơn. Đáp ứng những đòi hỏi này, NIST đã chọn tiêu chuẩn cho phép có nhiều cở modulo, nghĩa là cỡ modulo bất kì chia hết cho 64 trong phạm vi từ 512 đến 1024 bít.

Một phàn nàn khác về DSS là chữ kí được tạo ra nhanh hơn việc xác minh nó. Trong khi đó, nếu dùng RSA làm sơ đồ chữ kí với số mũ xác minh công khai nhỏ hơn (chẳng hạn = 3) thì có thể xác minh nhanh hơn nhiều so với việc lập chữ kí. Điều này dẫn đến hai vấn đề liên quan đến những ứng dụng của sơ đồ chữ kí:

1.Bức điện chỉ được kí một lần, song nhiều khi lại cần xác minh chữ kí nhiều lần trong nhiều năm. Điều này lại gợi ý nhu cầu có thuật toán xác minh nhanh hơn.

2.Những kiểu máy tính nào có thể dùng để kí và xác minh ?. Nhiều ứng dụng, chẳng hạn các thẻ thông minh có khả năng xử lý hạn chế lại liên lạc với máy tính mạnh hơn. Vi thế có nhu cầu nhưng thiết kế một sơ đồ để có thực hiện trên thẻ một vài tính toán. Tuy nhiên, có những tình huống cần hệ thống mình tạo chữ kí, trong những tình huống khác lại cần thẻ thông minh xác minh chữ kí. Vì thế có thể đưa ra giải pháp xác định ở đây.

Sự đáp ứng của NIST đối với yêu cầu về số lần tạo xác minh chữ kí thực ra không có vấn đề gì ngoài yêu cầu về tốc độ, miễn là cả hai thể thực hiện đủ nhanh.

## 6.4 CHỮ KÍ MỘT LẦN

Trong phần, này chúng ta mô tả cách thiết lập đơn giản một sơ đồ chữ kí một lần từ hàm một chiều. Thuật ngữ "một lần" có nghĩa là bức điện được kí chỉ một lần (dĩ nhiên chữ kí có thể xác minh nhiều lần tuỳ ý). Sơ đồ mô tả là sơ đồ chữ kí Lamport nêu hình 6.4.

Sơ đồ làm viêc như sau: Bức điện được kí là một bức điện nhị phân k bít. Một bít được kí riêng biệt nhau. Giá trị $z_{i,j}$ tương ứng với bít thứ i của bức điện có giá trị j (j =0,1). Mỗi $z_{i,j}$ là ảnh hưởng đến $y_{i,j}$ dưới tác động của hàm một chiều f. Bít thứ i của bức điện được kí nhờ là ảnh gốc(nghịch ảnh - priemage) $y_{i,j}$ của $z_{i,j}$ (tương ứng với bít thứ i của bức điện ). Việc xác minh chỉ đơn giản là kiểm tra xem mỗi phần tử trong chữ kí có là ảnh gốc của phần tử

***Hình 6.4. Sơ đồ chữ kí Lamport***

Cho k là số nguyên dương và cho $\mathcal{P} = \{0,1\}^k$ . Giả sử f:Y → Z là hàm một chiều và cho $\mathcal{A} = Y^k$ . Cho $y_{i,j} \in Y$ được chọn ngẫu nhiên. $1 \le i \le k$, j =0,1 và giả sử $z_{i,j} = f(y_{i,j})$. Khoá K gồm 2k giá trị y và 2k giá trị z. Các giá trị của i giữ bí mật trong khi các giá trị của z công khai.

Với $K = (y_{i,j}, z_{i,j} : 1 \le i \le k, j = 0,1)$ , ta định nghĩa :

$$sig_k(x_1 \ldots x_k) = (????\text{tự đánh vào})$$

và $ver_k(x_1 \ldots x_k, a_1 \ldots a_k) = true \Leftrightarrow f(a_i) = ????\text{tự đánh vào}$

khoá công khai thích hợp hay không.

Sau đây sẽ minh hoạ sơ đồ bằng việc xem xét một thực hiện dùng hàm mũ $f(x) = \alpha^x \bmod p$. $\alpha$ là một phần tử nguyên thuỷ modulo p.

Ví dụ 6.4

7879 là số nguyên tố và 3 là phần tử nguyên thuỷ thuộc $Z_{7879}$. Định nghĩa:

$$f(x) = 3^x \bmod 7879$$

Giả sử Bob muốn kí một bức điện có 3 bít. Anh ta chọn 6 số tự nhiên (mật)

$y_{1,0} = 5831$ $y_{2,1} = 2467$
$y_{1,1} = 735$ $y_{3,0} = 4285$
$y_{2,0} = 803$ $y_{3,1} = 6449$

Khi đó, anh ta tính các ảnh của y dưới hàm f

$z_{1,0} = 2009$ $z_{2,1} = 4721$
$z_{1,1} = 3810$ $z_{3,0} = 268$
$z_{2,0} = 4672$ $z_{3,1} = 5731$

Các ảnh của z này được công khai. Bây giờ giả sử Bob muốn ký bức điện

$x = (1, 1, 0)$

chữ kí trên x là:

$(y_{1,1}, y_{2,1}, y_{3,0}) = (735, 2467, 4285)$

Để xác minh chữ kí, chỉ cần tính toán như sau:

$3^{735} \bmod 7879 = 3810$
$3^{4675} \bmod 7879 = 4721$
$2^{4285} \bmod 7879 = 268$

Vì thế, chữ kí hợp lệ.

Oscar không thể giả mạo chữ kí vì anh ta không thể đảo được hàm một chiều f(x) để có các giá trị y mật. Tuy nhiên, sơ đồ được dùng để kí chỉ một bức điện. Bởi vì nếu cho trước chữ kí của 2 bức điện khác nhau. Oscar sẽ dễ dàng xây dựng chữ kí cho bức điện khác.

Ví dụ, giã sử các bức điện (0, 1, 1) và (1, 0, 1) đều được kí bằng cùng một sơ đồ. Bức điện (0, 1, 1) có chữ kí $(y_{1,0}, y_{2,1}, y_{3,1})$ còn bức điện (1,0,1) có chữ kí $(y_{1,1}, y_{2,0}, y_{3,1})$. Nếu cho trước 2 chữ kí này, Oscar có thể xây dựng các chữ kí của bức điện (1,1,1) là $(y_{1,1}, y_{2,1}, y_{3,1})$ và chữ kí cho bức điện (0,0,10 là $(y_{1,0}, y_{2,0}, y_{3,1})$.

Mặc dù sơ đồ này hoàn toàn tốt song nó không được sử dụng trong thực do kích thước chữ kí. Ví dụ, nếu ta dùng hàm số mũ modulo như trong ví dụ ở trên thì yêu cầu an toàn đòi hỏi p dài ít nhất 512 bít. Điều này, có nghĩa mỗi bít của bức điện chữ kí dùng 512 bít. Kết quả chữ kí dài hơn bức điện 512 lần.

Bây giờ xét một cải tiến của Bos và Chaum cho phép chữ kí ngăn hơn một chút song không giảm độ mật. Trong sơ đồ Lamport, lý do Oscar không thể giả mão chữ kí trên bức điện (thứ hai) khi biết chữ kí ở bức điện là: các

ảnh của y (tương ứng với một bức điện ) không bao giờ là tập con của các ảnh của y (tương ứng với bức điện khác).

Giả sử ta có tập $\mathcal{B}$ gồm các tập con của B sao cho $B_1 \subseteq B_2$ chỉ khi $B_1 = B_2$ với mọi $B_1, B_2 \in \mathcal{B}$. Khi đó $\mathcal{B}$ được gọi là thoả mãn tính chất Sperner. Cho trước một tập B có lực lượng n chẵn, khi đó kích thước cực đại của tập $\mathcal{B}$ gồm các tập con B có tính chất Sperner là $\binom{2n}{n}$. Điều này dễ dàng nhận được bằng cách lấy tất cả các tập con n của B: rõ ràng không có tập con n nào nhận được trong tập con n khác

Bây giờ, giã sử ta muốn ki một bức điện k bít như trước đây, ta chọn n đủ lớn để.

$$2^k \leq \binom{2n}{n}$$

Cho | B | =n và giả sữ $\mathcal{B}$ chỉ tập các tập con n của B. Giả sử $\phi:\{0,1\}^k \rightarrow \mathcal{B}$ là đơn ánh trong công khai đã biết. Khi đó, có thể liên kết mỗi bức điện có thể với một con n trong $\mathcal{B}$. Ta sẽ có 2n giá trị của y, và 2n giá trị của z và mỗi bức điện được kí bằng n ảnh của y. Hình 6.5 mô tả đầy đủ sơ đồ Bos- chaum.

***Hình 6.5 Sơ đồ chữ kí Bos - chaum.***

Cho k là số nguyên dương và giả sử $\mathcal{P}=\{0,1\}^k$. Cho n là số nguyên sao cho $2^k \leq \binom{2n}{n}$ và B là tập có lực lượng n và cho

$$\phi: \{0,1\}^k \rightarrow \mathcal{B}$$

là một đơn ánh , trong đó $\mathcal{B}$ là tập tất cả các con n của B. Giả sử f: Y→Z là hàm một chiều và $\mathcal{A} = Z^n$. Cho ??????????????

Ưu điểm của sơ đồ Bos- chaum là các chữ kí ngăn hơn sơ đồ Lamport. Ví dụ, ta muốn ký một bức điện 6 bit (k = 6). Vì $2^6$ =64 và $\binom{8}{2}$ =70 nên có thể lấy n =4 và bức điện 6 bit được kí bằng 4 giá trị của y so với 6 của sơ đồ Lamport. Như vậy khoá k sẽ ngắn hơn, nó gồm 8 giá trị của z so với 12 của sơ đồ Lamport.

Sơ đồ Bos-Chaum đòi hỏi hàm đơn ánh $\phi$ để kết hợp tập con n của tập 2n với mỗi x nhị phân bội k ($x_1$ …. $x_k$). Ta sẽ đưa ra một thuật toán đơn giản để thực hiện điều này (hinh 6.6). Ví dụ, áp dụng thuật toán này với x = (0,1,0,0,1,1) sẽ tạo ra.

$$\phi(x) = \{2,4,6,8\}$$

Nói chung, n trong sơ đồ Bos-Chaum lớn bao nhiêu so với k ?. Ta cần thoả mãn bất phương trình $2^k \le \binom{2n}{n}$ Nếu đánh giá hệ số của nhị thức

$$\binom{2n}{2} = (2n)!/(n!)^2$$

Hình 6.6 ***Tính $\phi$ trong sơ đồ Bos - chaum***

1. $X = \sum_{i-1}^{k} x_i 2^{i-2}$
2. $\phi(x) = 0$
3. $t = 2n$
4. $e = n$
5. While $t > 0$ do
6. $t = t - 1$
7. if $x > \binom{t}{e}$ then
8. $x = x - \binom{t}{e}$
9. $e = e - 1$
10. $\phi(x) = \phi(x) \cup \{t+1\}$

băng công thức Stirling $2^{2n} / \sqrt{\pi n}$ Sau vài phép biến đổi đơn giản, bất kỳ đẳng thức trở thành

$$k \le 2n - \log_2 (\pi n)/2$$

Một cách gần đúng, $n \approx k/2$. Như vậy, ta đã giảm được khoảng 50% kích thước chữ kí bằng sơ đồ Bos - chaum.

---

## 6.5 CÁC CHỮ KÍ KHÔNG CHỐI ĐƯỢC

Các chữ kí không chối được do Chaum và Antwerpen đưa ra từ năm 1989. Chúng có vài đặc điểm mới. Nguyên thuỷ nhất trong các chữ kí này là chữ kí không thể xác minh được nếu không hợp tác với người ký là Bob. Như vậy sẽ bảo được Bob trước khả năng các tài liệu được anh ta ký bị nhân đôi và phân phối bằng phương pháp điện tử mà không có sự đồng ý của anh ta. Việc xác minh được thực hiên bằng giao thức yêu cầu và đáp ứng (Challege and repotocol).

Song liệu có cần sự hợp tác của Bob để xác minh chữ kí (nhằm ngăn chặn Bob từ chối không nhận đã ký trước đó) không? Bob có thể truyền thống chữ kí hợp lệ là giả mạo và từ chối xác minh nó, hoặc thực hiện giao thức theo cách để chữ kí không thể được xác minh. Để ngăn chặn tình huống này xảy ra, sơ đồ chữ kí không chối được đã kết hợp giao thức từ chối (theo giao thức này, Bob có thể chứng minh chữ kí là giả mạo). Như vậy, Bob sẽ có khả năng chứng minh trước toà rằng chữ kí bị lừa dối trên thực tế là giả mạo. (Nếu anh ta không chấp nhận tham vào giao thức từ chối, điều này được xem như bằng chứng chứng tỏ chữ kí trên thực tế là thật).

Như vậy, sơ đồ chữ kí không chối được gồm 3 thành phần: thuật toán ký, giao thức xác minh và giao thức từ chối (disavowal). Đầu tiên ta sẽ đưa ra thuật toán ký và giao thức xác minh của sơ đồ chữ kí không từ chối được của chaum - VanAntwerpen trên hình 6.7.

Xét vai trò của p và q trong sơ đồ này. Sơ đồ tồn tại trong $Z_p$; tuy vậy cần có khả năng tính toán theo nhóm nhân con G của $Z_p^*$ có bậc nguyên tố. Củ thể, ta có khả năng tính được các phần tử nghịch đảo Modulo |G| - là lý do giải thích tại sao |G| phải là số nguyên tố. Để tiện lợi, lấy p=2q+1, q là số

nguyên tố. Theo cách này, nhóm con G lớn đến mức có thể là điều đáng mong muốn vì cả bức điện lẫn chữ kí đều là phần tử thuộc G.

Trước hết, cần chứng minh rằng, Alice sẽ chấp nhận một chữ kí hợp lệ. Trong các tính toán sau đây, tất cả các số mũ được rút gọn theo modulo q. Đầu tiên, nhận xét:

$$d \equiv c^{\alpha^{-1}} \pmod{p}$$

***Hình 6.7. Sơ đồ chữ kí không chấp nhận chaum - Van Antwerpen.***

Cho p =2q +1 là số nguyên tố sao cho q là nguyên tố và bài toán logarithm rời rạc trong $Z_p$ là không thể giải được. Giả sử $\alpha \in Z_p$ là phần tử bậc q. Cho $1 \le a \le q-1$ và được định nghĩa $\beta = \alpha^a \bmod p$. Giả sử G biểu nhóm con bội $Z_p^*$ bậc q (G là gồm các thặng dư bình thường modulo p). Cho $\mathcal{P} = \mathcal{A} = G$ và định nghĩa :

$\mathcal{K} = \{p, \alpha, a, \beta): \beta \equiv \alpha^a \pmod{p}\}$

Các giá trị p, $\alpha$ và $\beta$ công khai, còn a mật.

Với k = (p, $\alpha$, a, $\beta$ ) và x $\in$G , định nghĩa :

$$y = sig_k(x) = x^a \bmod p$$

Với x,y $\in$ G, việc xác minh được thực hiện qua giao thức sau:

1. Alice chọn $e_1, e_2$ ngẫu nhiên, $e_1, e_2 \in Z_p^*$
2. Alice tính $c = y^{e_1}\beta^{e_2} \bmod p$ và gửi cho no đến Bob
3. Bob tính $d = c^{a \bmod q} \bmod p$ và gữi nó cho Alice
4. Alice chấp nhận y là chữ kí hợp lệ khi và chỉ khi

$$d \equiv x^{e_1}\alpha^{e_2} \pmod{p}$$

Vì:

$$\beta \equiv \alpha^a \pmod{p}$$

Ta có:

????Chua viết

Tương tự

$$y = x^a \pmod{p}$$

có nghĩa là:

?????????? chưa viết

Vì thế $d \equiv x^{e_1}\alpha^{e_2} \pmod p$
như mong muốn.

Dưới đây là một ví dụ nhỏ.

Ví dụ 6.5

Giả sử lấy p = 467, vì 2 là phần tử nguyên thuỷ nên $2^2$ =4 là phần tử sinh của G, các thặng dư bình phương modulo 467. Vì thế ta có thể lấy $\alpha$ =4. Giả thiết a = 101, khi đó

$$\beta = \alpha^a \bmod 467 = 499$$

Bob sẽ ký bức điện x= 119 với chữ ký

$$y = 119^{101} \bmod 467 = 129$$

Bây giờ giả sử Alice muốn xác minh chữ ký y, cô ta chọn các số ngẫu nhiên chẳng hạn $e_1$=38, $e_2$=397. Cô tính c=13, ngay lúc đó Bob sẽ trả lời với d =9,Alice kiểm tra câu trả lời bằng việc xác minh xem:

$$119^{38}\, 4^{397} = 9 \pmod{467}$$

vì thế Alice chấp nhận chữ ký là hợp lệ.

Tiếp theo, ta chứng minh rằng, Bob không thể lừa Alice chấp nhận chữ ký giả mạo (Fradulart) như là chữ ký hợp lệ trừ một xác suất rất bé. Kết quả này không phụ thuộc vào bất kỳ giả thiết tính toán nào, điều đó có nghĩa độ an toàn là vô điều kiện.

**Định lý 6.1**

Nếu $y \not\equiv x^a \pmod p$ thì Alice sẽ nhận y như la một chữ ký hợp lệ trên x với xác xuất 1/q.

Chứng minh

Trước hết, nhânj xét rằng mỗi yêu cầu (challenge) c có thể tương ứng chính xác với q cặp được sắp $(e_1,e_2)$ (đó là vì cả y lẫn β đều là các phần tử của nhóm nhân G có bậc nguyên tố q). Bây giờ, khi Bob nhận được yêu cầu c, anh ta không có cách nào để biết về q cặp được sắp $(e_1,e_2)$ có thể mà Alice đã

dùng để xây dựng c. Ta nói rằng, nếu $y \not\equiv x^a \pmod p$ thì đáp dụng ứng (respond) $d \in G$ mà Bob có thể là sẽ chỉ phủ hợp chính xác một trong q cặp được $(e_1,e_2)$.

Vì $\alpha$ sinh ra G, nên ta có thể viết một phần tữ bất kỳ thuộc G như một số mũ của $\alpha$, trong đó số mũ được xác minh duy nhất theo modulo q. Vì thế có thể viết $c = \alpha^i, d = \alpha^j$, $x = \alpha^k$ và $y = \alpha^{\ell}$ với $i, j, k, \ell \in Z_p$ và mọi phép tính số học là theo modulo q. Xét 2 đồng dư thức sau:

$$c \equiv y^{e_1}\beta^{e_2} \pmod p$$
$$d \equiv x^{e_1}\alpha^{e_2} \pmod p$$

Hệ thống này tương đương hệ đồng thức sau:

$$i \equiv \ell e_1 + a e_2 \pmod q$$
$$j \equiv k e_1 + e_2 \pmod q$$

Bầy giờ giả thiết rằng:

$$y \not\equiv x^a \pmod p$$

Comment [NVH1]:

nên rút ra : $\ell \not\equiv a k \pmod q$

Vì thế, ma trận hệ số của các đồng dư thức theo modulo q này có định thức khác 0 và như vây tồi tại nghiệm duy nhất cho hệ thống thống. Nghiã là, mỗi $d \in G$ là một đáp ứng với một trong q cặp $(e_1,e_2)$ được sắp có thể. Hệ thống quả là, xác suất để Bob đưa cho Alice một đáp ứng(trả lời) d cần được xác minh đúng bằng 1/q. Định lý được chứng minh.

***Hình 6.8***. Thủ tục từ chối**.**

1. Alice chọn $e_1,e_2$ một cách ngẫu nhiên, $e_1,e_2 \in Z_q^*$
2. Alice tính $c = y^{e_1}\beta^{e_2} \bmod p$ và gửi nó cho Bob.
3. Bob tính d = ?
4. Alice xác minh xem d có $\not\equiv x^{e_1}\alpha^{e_2} \pmod p$ không
5. Alice chọn $f_1,f_2$ ngẫu nhiên , $f_1,f_2 \in Z_q^*$
6. Alice tính $C = y^{f_1}\beta^{f_2} \bmod p$ và gửi cho Bob
7. Bob tính D = ????????
8. Alice xác minh xem D có $\equiv x^{f_1}\alpha^{f_2} \pmod p$ không
9. Alice kết luận rằng y là giả mạo khi và chỉ khi

$$(d\,\alpha^{-e_2})^{f_1} \equiv (D\,\alpha^{-f_2})^{e_1} \pmod p$$

Bây giờ quay trở lại giao thức từ chối. Giao thức này gồm hai 2 thực hiện giao thức xác minh và được nêu trong hình 6.8.

Các bước 1- 4 và 5- 8 gồm 2 lần thực hiện không thành công giao thức xác minh. Bước 9 bước "tính kiểm tra phù hợp" cho Alice xác định xem liệu có phải đang lập các câu trả lời của anh ta theo thứ tự chỉ ra hay không.

Dưới đây là ví dụ minh hoạ.

Ví dụ 6.6

Như trước đây, giả sử p = 467, $\alpha$ = 4, a = 101, và $\beta$ = 449. Giả thiết bức điện x = 286 được ký y = 83 và Bob muốn thiết phục Alice rằng chữ ký không hợp lệ.

Giả sử Alice bắt đầu bằng việc chọn các giá trị ngẫu nhiên $e_1$=45, $e_2$=237. Alice tính c =305 và Bob trả lời d = 109. Sau đó Alice tính

$$286^{125}4^{237} \bmod 467 = 149$$

Vì 149 # 109 nên Alice thực hiện bước 5 của giao thức.

Bây giờ giả sử Alice chọn giá trị ngẫu nhiên $f_1$= 125, $f_2$= 9. Alice tính C = 270 và Bob trả lời với D =68. Alice tính

$$186^{125}4^{9} \bmod 467 = 25$$

Vì 25 # 68 nên Alice tiếp tục sang bước 9 của giao thức kiểm tra tính phù hợp. Bước kiểm tra này thành công vì:

$$(109 \times 4^{-9})^{125} \equiv 188 \pmod{467}$$

và

$$(68 \times 4^{-9})^{45} \equiv 188 \pmod{467}$$

Vì thế Alice tin rằng chữ ký không hợp lệ. □

Bây giờ, ta phải chứng minh hai vấn đề:

1. Bob có thể thuyết phục Alice rằng, chữ ký không hợp lệ là giả mạo.
2. Bob không thể là Alice tin rằng chữ ký không hợp lệ là giả mạo trừ một xác suất rất bé.

Định lý 6.2

Nếu $y \not\equiv x^a \pmod p$ và cả Alice lẫn Bob thực hiện theo giao thức từ chối thì

$$(d\,\alpha^{-e_2})^{f_1} \equiv (D\,\alpha^{-f_2})^{e_1} \pmod p$$

*Chứng minh:*

Dùng các yếu tố

$$d \equiv ???$$

$$c \equiv y^{e_1}\beta^{e_2} (\text{mod } p)$$

và

$$\beta \equiv \alpha^a \text{ (mod } p)$$

Ta có:

$$(d\ \alpha^{-e_2})^{f_1} \equiv ????$$

Tượng tự, dùng các yếu tố $D \equiv ????$

$$(D\ \alpha^{-f_2})^{e_1} \equiv y^{e_1\ f_2} \text{ (mod } p)$$

vì thế phép kiểm tra tính phù hợp trong bước 9 thành công. □

Bây giờ xét xác suất để Bob có thể thử từ chối một chữ ký hợp lệ. Trường hợp này không giả thiết Bob thực hiên theo thủ tục. Nghĩa là Bob có thể không xây dưng D Và d như trong giao thức. Vì thế trong định lý tiếp theo chỉ là giả thiết rằng, Bob có thể tạo ra các D và d thoả mãn điều kiện trong các bước 4,8và 9 của giao thức nêu trên hình 6.8.

Định lý 6.3

Giả sử $y \equiv x^a \text{ (mod } p)$ và Alice thực hiện theo giao thức từ chối. Nếu

$$d \not\equiv x^{e_1}\ \alpha^{e_2} (\text{mod } p)$$

và

$$D \equiv x^{f_1}\ \alpha^{f_2} (\text{mod } p)$$

thì xác suất để:

$$(d\alpha^{-e_2})^{f_1} \not\equiv (D\alpha^{-f_2})^{e_1} \text{ (mod } p) = 1\text{-}1/q$$

chứng minh: giả sử rằng, các đồng dư thức sau được thoả mãn

$$y \not\equiv \alpha^a \text{ (mod } p)$$

$$d \not\equiv x^{e_1}\ \alpha^{e_2} (\text{mod } p)$$

$$D \not\equiv x^{f_1}\ \alpha^{f_2} (\text{mod } p)$$

$$(d\alpha^{-e_2})^{f_1} \equiv (D\alpha^{-f_2})^{e_1} \text{ (mod } p)$$

ta sẽ nhận được mâu thuẩn như trình bay sau đây:

có thể viết lại bước 9- bước kiểm tra tính phù hợp như sau

$$D \equiv d_0^{f_1}\ \alpha^{f_2}$$

trong đó

$$d_0 = d^{1/e_1}\alpha^{-e_2/e_1} \text{mod } p$$

là giá trị chỉ phụ thuộc vào các bước 1- 4 trong giao thức.

Áp dụng định lý 6.1, ta kết luận được y là chữ ký hợp lệ đối với $d_0$ với xác suất 1- 1/q. Song ta đã giả thiết y là chữ ký hợp lệ đối với x, nghĩa là là ta có (với xác suất cao)

$$x^a \equiv d_0^{\ a} \text{ (mod } p)$$

có nghĩa là $x = d_0$

Tuy nhiên do

$$d \not\equiv x^{e_1} \alpha^{e_2} (\text{mod } p)$$

có nghĩa là

$$x \not\equiv d^{1/e_1} \alpha^{-e_2/e_1} (\text{mod } p)$$

Và từ chỗ

$$d_0 \equiv d^{1/e_1} \alpha^{-e_2/e_1} (\text{mod } p)$$

suy ra $x \# d_0 \Rightarrow$ ta nhận được mâu thuẩn.

Như vậy Bob có thể lừa dối Alice theo cách này với xác suất 1/q.

---

## 6.6 CÁC CHỮ KÝ FAIL- STOP

Sơ đồ chữ ký Fail- stop dùng để tăng độ mật trước khả năng một đối thủ mạnh có thể giả mạo chữ ký. Nếu Oscar khả năng giả mạo chữ ký của Bob thì Bob có khả năng chứng minh được (với xác suất cao) rằng chữ ký của Oscar là giả mạo.

Phần này sẽ mô tả một sơ đồ Fail- stop do Van Heyst va Pedersen đua ra năm 1992. Đầu là sơ đồ chữ ký 1 lần (chỉ một bức điện có thể ký bằng một cho trước chỉ 1 lần). Hệ thống gồm các thuật toán ký, thuật toán xác minh và thuật toán "chứng minh giả mạo". Hình 6.9 mô tả các thuật toán ký và xác minh của sơ đồ Fail- stop của Van Heyst va Pedersen.

Không khó khăn nhận thấy rằng, chữ ký do Bob tao ra sẽ thoả mãn điều kiện xác minh nên ta lại trở các kía cạnh an toàn toàn của sơ đồ này và các thức làm việc của tính chất Fail- Safe (tự động ngừng khi có sai số). Trước hết, ta thiết lập vài yếu tố quan trọng có liên quan đến các khoá của sơ đồ. Đầu tiên đưa ra một định nghĩa: Hai khoá ($\gamma_1$, $\gamma_2$, $a_1$, $a_2$, $b_1$, $b_2$) và ($\gamma_1$', $\gamma_2$', $a_1$', $a_2$', $b_1$', $b_2$') là tương đương nếu $\gamma_1 = \gamma_1$', $\gamma_2 = \gamma_2$'. Và dễ dàng nhận thấy tồi tại $q^2$ khoá trong lớp tương đương bất kỳ.

Sau đây là vài bổ đề.

**Bổ đề 6.4**

Giả sử K và K'là các khoá tương đương và giả thiết chữ ký $ver_K(x,y)$ = true(đúng). Khi đó chữ ký $ver_{K'}(x,y)$ = true.

Chứng minh

Giả sử $K =(\gamma_1, \gamma_2, a_1, a_2, b_1, b_2)$ và $K'= (\gamma_1', \gamma_2', a_1', a_2', b_1', b_2')$ trong đó :

$$\gamma_1= \alpha^{a_1}\beta^{a_2} \bmod p =\alpha^{a'_1}\beta^{a'_2} \bmod p$$
$$\gamma_2= \alpha^{b_1}\beta^{b_2} \bmod p =\alpha^{b'_1}\beta^{b'_2} \bmod p$$

Giả sử x được bằng cách dùng K và tạo ra các chữ ký $y =(y_1, y_2)$ trong đó:

$$y_1= a_1+xb_1 \bmod q$$
$$y_2= a_2+xb_2 \bmod q$$

Hình 6.9 Sơ đồ chữ ký Fail- stop.

Cho p = 2q+1 là số nguyên tố sao q là nguyên tố và bài toán logarithm rời rạc trong $Z_p$ là khó giải. cho $\alpha \in Z_p^*$ là phần tử bậc q. Giả sử $1 \le a_0 \le q-1$ và định nghĩa $\beta =\alpha^{a_0} \bmod p$. Các giá trị p, q, $\alpha$, $\beta$ và $a_0$ đều do người có thẩm quyền (được tin cậy) chọn. Các số p, q, $\alpha$ và $\beta$ công khai và cố định còn $a_0$ được giữ bí mật.

Cho $\mathcal{P} =Z_p$ và $\mathcal{A} = Z_q \times Z_q$. khoá có dạng:

$$K =(\gamma_1, \gamma_2, a_1, a_2, b_1, b_2)$$

trong đó $a_1, a_2, b_1, b_2 \in Z_q$

$$\gamma_1= \alpha^{a_1}\beta^{a_2} \bmod p$$

còn $$\gamma_2= \alpha^{b_1}\beta^{b_2} \bmod p$$

Với $K =(\gamma_1, \gamma_2, a_1, a_2, b_1, b_2)$ và $x \in Z_p^*$, ta định nghĩa

$$sig_k(x) =(y_1, y_2)$$

trong đó

$$y_1= a_1+xb_1 \bmod q$$

còn $$y_2= a_2+xb_2 \bmod q$$

Với $y =(y_1, y_2) \in Z_q \times Z_q$ ta có:

Xác minh ver(x,y) = true $\Leftrightarrow \gamma_1\gamma_2^x \equiv \alpha^{y_1}\beta^{y_2} \pmod p$

Bây giờ giả sử ta xác minh y bằng cách dùng K'

$$\alpha^{y_1}\beta^{y_2} \equiv \alpha^{a'_1 + xb'_1}\beta^{a' + xb'_2} \pmod p$$
$$\equiv \alpha^{a'_1}\beta^{a'_2}(\alpha^{b'_1}\beta^{b'_2})^x \pmod p$$
$$\equiv \gamma_1\gamma_2^x \bmod p$$

Như vậy, y cũng sẽ được xác minh bằng K'.

Bổ đề 6.5

Giả sử K là khoá còn $y = sig_{K'}(x)$. Khi đó tồn tại đúng q khoá K' tương đương với K sao cho $y= sig_{K'}(x)$.

*Chứng minh*

Giả sử $\gamma_1$và $\gamma_2$là các thành phần công khai của K. Ta muốn xác định số bội 4 $(a_1, a_2, b_1, b_2)$sao cho các đồng dư thức sau đây được thoả mãn.

$$\gamma_1 \equiv \alpha^{a_1}\beta^{a_2} \pmod p$$
$$\gamma_2 \equiv \alpha^{b_1}\beta^{b_2} \pmod p$$
$$y_1 \equiv a_1 + xb_1 \pmod q$$
$$y_2 \equiv a_2 + xb_2 \pmod q.$$

Vì $\alpha$ sinh ra G nên tồn tại các số mũ duy nhất $c_1, c_2, a_0 \in Z_q$ sao cho

$$\gamma_1 \equiv \alpha^{c_1} \pmod p$$
$$\gamma_2 \equiv \alpha^{c_2} \pmod p$$

và $$\beta \equiv \alpha^{a_0} \pmod p$$

vì thế nó điều kiện cần và đủ để hệ các đồng dư thức sau đây được thoả mãn:

$$c_1 \equiv a_1 + a_0a_2 \pmod q$$
$$c_2 \equiv b_1 + a_0b_2 \pmod q$$
$$y_1 \equiv a_1 + xb_1 \pmod q$$
$$y_2 \equiv a_2 + xb_2 \pmod q$$

Hệ thống này có thể viết dưới dạng phương trình ma trận trong $Z_q$ như sau:

X

$$\begin{pmatrix} 1 & a_0 & 0 & 0 \\ 0 & 0 & 1 & a_0 \\ 1 & 0 & x & 0 \\ 0 & 1 & 0 & x \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \\ y_1 \\ y_2 \end{pmatrix}$$

có thể thấy ma trận hệ thống số của phương trình có hạng là 3( hạng của một ma trận là số cực đại của các hàng độc lập tuyến tính mà nó có). Rõ ràng, hạng ít nhất bằng 3 vì các hàng 1, 2 và 4 là độc lập tuyến tính trên $Z_p$. còn hạng nhiều nhất cũng bằng 3 vì:

$$r_1 + x\, r_2 - r_3 - a_0 r_4 = (0,0,0,0).$$

Với $r_1$ chỉ hàng thứ i của ma trận.

Hệ phương trình này có ít nhâts một nghiệm nhận được bằng cách dùng khoá K.Vì hàng của ma trận hệ số bằng 3 nên suy ra rằng chiều của không gian nghiệm là 4-3=1 và có chính xác q nghiệm.

Tương tự như vậy ta có thể chứng minh được kết qủa sau:

Bổ đề 6.6

Giả sử K là khoá $y=sig_K(x)$ còn $ver_K(x',y')$=true, trong đó $x' \# x$. Khi đó tồn tại ít nhất một khoá $K'$ tương đương với K sao cho $y=sig_{K'}(x)$ và $y'= sig_{K'}(x')$

Ta hãy làm sáng tỏ hai bổ đề trên về độ mật của sơ đồ. Khi cho trước y là chữ kí hợp lệ của x, sẽ tồn tại q khoá có thể để x sẽ được kí bằng y. Song với bức điện bất kì $x' \neq x$, q khoá này sẽ tạo ra q khoá khác nhau trên $x'$. Điều đó dẫn đến định lí sau đây:

Định lí 6.7:

Nếu cho trước $sig_K(x)=y$ và $x' \neq x$. Oscar có thể tính $sig_K(x')$ với xác suất là 1/q.

Chú ý rằng, định lí này không phụ thuộc vào khả năng tính toán của Oscar: Mức an toàn qui định đạt được vì Oscar không thể nói về q khoá có thể mà Bob đang dùng. Như vậy độ an toàn ở đây là vô điều kiện.

Tiếp tục xem xét về khái niệm Fail- Stop. Khi cho trước chữ ký y trên bức điện x. Oscar không thể tính ra được chữ ký $y'$ của Bob trên bức điện $x'$ khác. Điều này cũng có thể hiểu rằng, Oscar có thể tính được chữ ký giả mạo

$y'' = sig_K(x')$(sẽ được chứng minh ). Tuy nhiên, nếu đưa cho Bob một chữ ký giả mạo hợp lệ, thì anh ta có thể tạo ra "một bằng chứng về sự giả mạo " với xác suất 1-1/q. Bằng chứng về sự giả mạo là giá trị $a_0=\log_\alpha \beta$ (chỉ người có thẩm quyền trung tâm biết ).

Giả sử Bob sở hữu cặp $(x',y'')$ sao cho $ver(x',y'')$= true và $y''\neq sig_K(x')$.

Nghĩa là:

$$\gamma_1\gamma_2^{x'} \equiv \alpha^{y''_1}\beta^{y''_2}(\text{mod } p)$$

trong đó :$y''=(y''_1,y''_2)$. Bây giờ Bob có thể tính chữ ký của mình trên $x'$ là $y' = (y'_1,y'_2)$. Khi đó :

$$\gamma_1\gamma_2^{x'} \equiv \alpha^{y'_1}\beta^{y'_2}(\text{mod } p)$$

vì thế $$\alpha^{y''_1}\beta^{y''_2} \equiv \alpha^{y'_1}\beta^{y'_2}(\text{mod } p)$$

Nếu viết $\beta = \alpha^{a_0} \text{mod } p$, ta có :

$$\alpha^{y''_1+ a_0 y''_2} \equiv \alpha^{y'_1+ a_0 y'_2}(\text{mod } p)$$

hay:

$$y''+a_0y''_2 \equiv y'_1+a_0y'(\text{mod } q)$$

hoặc:

$$y''_1- y_1' \equiv a_0(y'_2-y''_2)(\text{mod } q)$$

Xét thấy $y'_1 \not\equiv y''_2(\text{mod } q)$ vì $y'$ là giả mạo. Vì thế $(y'_2-y''_2)^{-1} \text{mod } q$ tồn tại và

$$a_0 =\log_\alpha \beta = (y''_1-y'_1)\ (y'_2-y''_2)^{-1} \text{ mod } q$$

Dĩ nhiên, bằng việc chấp nhận bằng chứng về sự giả mạo như vậy,ta giả thiết Bob không thể tự tính được logarithm rời rạc $\log_\alpha \beta$. Đây là gải thiết về mặt tính toán.

Cuối cùng, chú ý rằng,sơ đồ chữ kí là một lần vì khóa k của Bob có thể tính dễ dàng nếu hai bức điện đều dùng K để ký. Dưới đây là ví dụ minh hoạ cách Bob tạo một bằng chứng về sự giả mạo.

Vi dụ 6.7

Cho p=4367=2.1733+1. Phần tử $\alpha$ =4 có bậc là 1733 trong $Z_{3467}^*$
Giả sử $a_o$ =1567, ta có:

$$\beta = 4^{1567} \bmod 346=514$$

(Bob biết $\alpha$ và $\beta$ song không biết $a_0$). Giả sử Bob tập khoá bằng cách dùng $a_1$ = 888, $a_2$ = 1042, $b_1$ = 786, $b_2$ = 999. Khi đó

$$\gamma_1=4^{888}514^{1024} \bmod 3476=3405$$

và $$\gamma_2=4^{786}514^{999} \bmod 3476=2281$$

Tiếp theo, giả sử Bob nhận được chữ kí giả mạo (822,55) trênê bức điện 3383. Đây là chữ ký hợp lệ vì thoả mãn điều kiện xác minh.

$$3405 \times 2281^{3384} \equiv 2282 \pmod{3476}$$

và $$4^{822}514^{55} \equiv 2282 \pmod{3476}$$

Mặt khác đây không phải là chữ kí đã được Bob xây dựng. Bob có thể tíng chữ kí của mình như sau:

(888+3383×786 mod 1733.1024+3383×999 mod 1733 )=(1504.1291)

Sau đó anh ta tính tiếp log rời rạc bí mật

$$a_0 =(822-1504)(1291-55)^{-1} \bmod 1733 =1567.$$

Đây là bằng chứng về sự giả mạo.

---

## 6.7 CÁC CHÚ GIẢI VỀ TÀI LIỆU DẪN

Mitchell, Piper và Wild [MPW 92] đã đưa ra một tổng quan đầy đủ về các sơ đồ chữ kí. Bài này cũng có hai phương pháp giả mạo chữ kí của Elgamal mà ta đã đưa ra trong 6.2.

Sơ đồ chữ kí Elgamal đã được nêu trong [EL 85], tiêu chuẩn chữ kí số được công bố đầu tiên vào 8/1991 bởi NIST và được chấp nhận làm tiêu chuẩn vào 12/94 [NBS 94]. Một cuộc thoả luận dài về DSS và những cuộc tranh cãi xung quanh nó vào 7/1992 được đăng trên Communication of the ACM.

Sơ đồ Lamport được mô tả trong bài báo của Diffie_Hellman [DH 76] năm 1976. Bản cải tiến củaBob và Chaum được nêu trong [BC 93]. Sơ đồ chữ

kí không chối nêu trong mục 6.5 do Chaum và Van Antwerpen đưa ra trong [CVA 90]. Sơ đồ chữ kí Fail-Stop trong mục 6.6 là của Van Heyst và Pederson [VHP 93].

Một số ví dụ về các sơ đồ chữ kí "phá được "gồm các sơ đồ của ông Schorss- Sshamir[OSS 85](cũng bị phá bởi Estes EAKMM 86) và sơ đồ hoán vị Birational của Shamir [SH94] (bị Coppessnuth, Steru và Vandeney CSV 94). Cuối cùng ESIGN là sơ đồ chữ kí của Fujioka.Okamoto và Meyaguchi [FOM 91]. Một số phiên bản của sơ đồ này đã bị phá. Song một sửa đổi trong [FOM 91] lại không bị phá.

---

## BÀI TẬP

6.1. Giả thiết Bob đang dùng sơ dồ Elgamal, anh ta kí hai bức điện $x_1$ và $x_2$ bằng chữ kí $(\gamma, \delta_1)$ và $(\gamma, \delta_2)$ tương ứng (giá trị này của $\gamma$ giống nhau trong cả hai chữ kí ). Cũng giả sử UCLN $(\gamma_1-\gamma_2, p-1)=1$.

a) Hãy cho biết cách tính k hiệu quả khi biết thông tin này
b) Hãy mô tả cách sơ đồ chữ kí có thể bị phá.
c) Giả sử p=31847, $\alpha$ =5, và $\beta$ =25703. Tính k và a khi cho trước chữ kí (2397 2,31396 ) với bức điện x=8990 và chữ kí (23972, 20481 trên bứ c điện x=31415)

6.2. Giả sử I thực hiên sơ đò Elgamal với p=31847, $\alpha$ =5,và $\beta$ =26379. Hãy viết phương trìng thực hiện công việc sau:
a) Xác minh chữ kí (20679,11082 ) trên bức điện x=20543
b) Xác định số mũ mật a bằng cách dùng thuật toán tối ưu hoá thời gian - bộ nhớ của Shark, sau dó xác định giá trị k ngẫu nhiên dùng trong việc kí lên bức điện x.

6.3. Giả sử Bob dùng sơ đố chữ kí Elgamal như trong ví dụ 6.1:p=467,$\alpha$ =2 $\beta$=132.Giả sử Bob kí lên bức điện x=100 bằng chữ kí (29,51).Hãy tính chữ kí giả mạo mà Oscar có thể lập bằng cách dùng h=100,i=45 và j =293.Hãy kiểm tra xem chữ ký vừa nhận được có thoả mãn điều kiện xác minh không.

6.4. Chứng minh rằng phương pháp giả mạo thứ hai trên sơ đồ Elgamal (mô tả trong mục 6.2) cũng tạo ra chữ kí thoả mãn điều kiện xác minh.

6.5. Sau đây là phương án của sơ đồ Elgamal :Khoá được xây dựng tương tự theo mghĩa như trước đây:Bob chọn $\alpha \in Z_p^*$ là phần tử nguyên thuỷ. a là số mũ mật ($0 \le a \le p-2$) sao cho UCLN (a,p-1) =1 và $\alpha^a \bmod p$. Khoá K = ($\alpha$, a, $\beta$ ), ở đây $\alpha$ và $\beta$ công khai còn a mật. Cho $x \in Z_p$ là bức điện được kí. Bob tính chữ kí sig (x)=($\gamma$, $\delta$ ), trong đó:

$$\gamma = \alpha^k \bmod p$$

còn $\delta = (x - k\gamma)^{a-1} \bmod (p-1)$.

Sự khác nhau duy nhất so với sơ đồ Elgamal ban đầu là ở cách tính $\delta$. Hãy trả lời các câu hỏi sau liên quan đến sơ đồ cải tiến này:

a) Mô tả cách xác minh một chữ kí ($\gamma$, $\delta$ ) trên bức điện x bằng cách dùng công khai khoá của Bob.
b) Mô tả ưu điểm về mặt tính toán của sơ đồ cải tiến.
c) So sánh tóm tắt độ an toàn của sơ đồ cải tiến và sơ đồ ban đầu.

6.6. Giả sử Bob dùng DSS với q = 101, p = 7879, $\alpha$ = 170, a = 75 còn $\beta$ = 4567 như trong ví dụ 6.3. Xác định chữ kí của Bob trên bức điện x=5011, bằng cách dùng giá trị ngẫu nhiên k =49 và chỉ ra cách xác minh chữ kí nhận được.

6.7. Trong sơ đồ Lamport,giả sử rằng hai bức điện x và $x'$ bội k (k-tuple) đều do Bob kí. Cho $\ell = d(x,x')$ là toạ độ trên đó x và $x'$ khác nhau. Hãy chỉ ra cách Oscar có thể kí $2^\ell - 2$ bức điện mới.

6.8. Trong sơ đồ Bob-Chaum với k = 6, n = 4, giả sử rằng các bức điện x = (0, 1,0,0,1,1) và $x'$ = (1,1,0,1,1) đều được kí. Xác định bức điện mới được Oscar kí khi biết chữ kí trên x và $x'$.

6.9. Trong sơ đồ Bob- Chaum, giả sử rằng hia bức điện x và $x'$ là các bội k đều do Bob kí Cho $l = |\phi(x) \cup \phi(x')|$. Hãy chỉ ra cách Oscar có thể kí $\binom{\ell}{n} - 2$ bức điện mới.

6.10. Giả sử Bob đang dùng chữ kí không chối được của Chaum –Van Antwerpen như trong ví dụ 6.5. Nghĩa là p = 467, $\alpha$ = 4, a = 101, $\beta$ = 449. Giả sử Bob được trình chữ kí y = 25 trên bức điện x =157 và anh ta muốn chứng minh rằng nó giả mạo. Giả sử số ngẫu nhiên của Alice là $e_1$ = 46, $e_2$ = 123, $f_1$ =198, $f_2$ =11 trong thủ tục từ chối. Hãy tính các yêu cầu c, d, của Alice và các câu trả lời C, D của Bob; chỉ ra rằng phép kiểm tra tính phù hợp của Alice sẽ thành công.