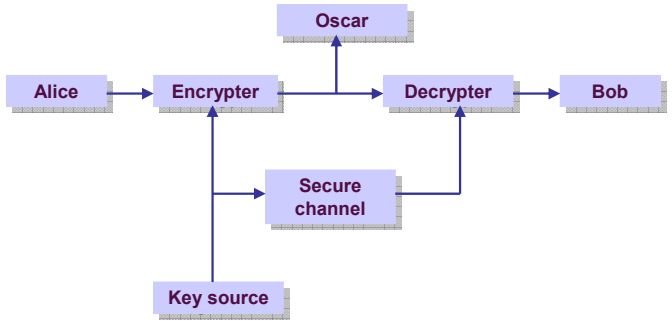
Mật mã hoá dữ liệu

PTIT
Lê Thị Thanh

2. Mật mã cổ điển

a. Các hệ mật thay thế đơn giản
b. Các hệ mật mã thay thế đa biểu
c. Mật mã hoán vị.
d. Các hệ mật mã tích
e. Chuẩn mã dữ liệu (DES)
f. Chuẩn mã dữ liệu tiên tiến (AES)

1. Giới thiệu



Định nghĩa

- *A cryptosystem is a five-tuple (P,C,K,E, D) where the following conditions are satisfied:*
  - **1.** P *is a finite set of possible plaintexts*
  - **2.** C *is a finite set of possible ciphertexts*
  - **3.** K, *the keyspace, is a finite set of possible keys*
  - **4.** *For each K, there is an encryption rule* $e_K \in E$. *and a corresponding decryption rule* $d_K \in D$. *Each* $e_K: P \rightarrow C$ *and* $d_K: C \rightarrow P$ *are functions such that* $d_K(e_K(x)) = x$ *for every plaintext*

a. Mật mã thay thế đơn giản

i. Mật mã dịch vòng
ii. Mật mã thay thế

i. Mật mã dịch vòng

- Giả sử: $P = C = K = Z_{26}$ với $0 \le k \le 25$
- $e_k = x + k \bmod 26$
- $d_k = y - k \bmod 26$
- $x, y \in Z_{26}$

Ký tự	A	B	C	D	E	F	G	H	I	J	K	L	M
Mã tương ứng	0	1	2	3	4	5	6	7	8	9	10	11	12
Ký tự	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Mã tương ứng	13	14	15	16	17	18	19	20	21	22	23	24	25

ii. Mật mã thay thế

- $P = C = Z_{26}$ và k chứa mọi hoán vị có thể có của 26 ký tự từ 0 tới 25. Với mỗi hoán vị $\pi \in K$, ta định nghĩa: $e_\pi(x) = \pi(x)$ và $d_\pi(y) = \pi^{-1}(y)$ trong đó π^{-1} là hoán vị ngược của π

Ký tự bản rõ	a	b	c	d	e	f	g	h	i	j	k	l	m
Ký tự bản mã	X	N	Y	A	H	P	O	G	Z	Q	W	B	T
Ký tự bản rõ	n	o	p	q	r	s	t	u	v	w	x	y	z
Ký tự mã	S	F	L	R	C	V	M	U	E	K	J	D	I

ii. Mật mã thay thế

- Như vậy $e_\pi(a) = X$; $e_\pi(b) = N$, ...
- Hàm giải mã là phép hoán vị ngược. Điều này thực hiện bằng cách viết hàng thứ hai lên trước rồi sắp xếp theo thứ tự chữ cái.

Ký tự bản mã	A	B	C	D	E	F	G	H	I	J	K	L	M
Ký tự bản rõ	d	l	r	y	v	o	h	e	z	x	w	p	t
Ký tự bản mã	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Ký tự mã	b	g	f	j	q	n	m	u	s	k	a	c	i

b. Polyalphabetic cipher

- MDV và MTT là các loại mật mã thay thế đơn ký tự.
- Vigenère là một loại mật mã thay thế đa ký tự.
- Sử dụng phép thay thế tương ứng: A ↔ 0, B ↔ 1, ..., Z ↔ 25.
- Khoá k = CIPHER ↔ (2, 8, 15, 7, 4, 17)
- Các phép cộng được thực hiện theo modulo 26

Bản rõ	12	4	4	19	12	4	0	19	18	20	13	18	4	19
Khoá	2	8	15	7	4	17	2	8	15	7	4	17	2	8
Bản mã	14	12	19	0	16	21	2	1	7	1	17	9	6	1

Polyalphabetic Cipher

- The most common method used is Vigenère cipher
- Vigenère cipher starts with a 26 x 26 matrix of alphabets in sequence. First row starts with 'A', second row starts with 'B', etc.
- Like the ADFGVX cipher, this cipher also requires a keyword that the sender and receiver know ahead of time
- Each character of the message is combined with the characters of the keyword to find the ciphertext character

Vigenère Cipher Table

	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
A	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
B	B C D E F G H I J K L M N O P Q R S T U V W X Y A B
C	C D E F G H I J K L M N O P Q R S T U V W X Y Z A B
D	D E F G H I J K L M N O P Q R S T U V W X Y Z A B C
E	E F G H I J K L M N O P Q R S T U V W X Y Z A B C D
F	F G H I J K L M N O P Q R S T U V W X Y Z A B C D E
G	G H I J K L M N O P Q R S T U V W X Y Z A B C D E F
H	H I J K L M N O P Q R S T U V W X Y Z A B C D E F G
I	I J K L M N O P Q R S T U V W X Y Z A B C D E F G H
J	J K L M N O P Q R S T U V W X Y Z A B C D E F G H I
K	K L M N O P Q R S T U V W X Y Z A B C D E F G H I J
L	L M N O P Q R S T U V W X Y Z A B C D E F G H I J K

Vigenère Cipher Table (cont'd)

	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
N	N O P Q R S T U V W X Y Z A B C D E F G H I J K L M
O	O P Q R S T U V W X Y Z A B C D E F G H I J K L M N
P	P Q R S T U V W X Y Z A B C D E F G H I J K L M N O
Q	Q R S T U V W X Y Z A B C D E F G H I J K L M N O P
R	R S T U V W X Y Z A B C D E F G H I J K L M N O P Q
S	S T U V W X Y Z A B C D E F G H I J K L M N O P Q R
T	T U V W X Y Z A B C D E F G H I J K L M N O P Q R S
U	U V W X Y Z A B C D E F G H I J K L M N O P Q R S T
V	V W X Y Z A B C D E F G H I J K L M N O P Q R S T U
W	W X Y Z A B C D E F G H I J K L M N O P Q R S T U V
X	X Y Z A B C D E F G H I J K L M N O P Q R S T U V W
Y	Y Z A B C D E F G H I J K L M N O P Q R S T U V W X
Z	Z A B C D E F G H I J K L M N O P Q R S T U V W X Y

Polyalphabetic Cipher

- E.g., Message = SEE ME IN MALL
- Take keyword as INFOSEC
- Vigenère cipher works as follows:

S E E M E I N M A L L
I N F O S E C I N F O

A R J A W M P U N Q Z

Polyalphabetic Cipher

- To decrypt, the receiver places the keyword characters below each ciphertext character
- Using the table, choose the row corresponding to the keyword character and look for the ciphertext character in that row
- Plaintext character is then at the top of that column

Polyalphabetic Cipher

- Decryption of ciphertext:

A R J A W M P U N Q Z
I N F O S E C I N F O

S E E M E I N M A L L

- Best feature is that same plaintext character is substituted by different ciphertext characters (i.e., polyalphabetic)

Vigenère Cipher

- Easiest way to handle Vigenère cipher is to use arithmetic modulo 26
- This approach dispenses with the need for the table
- Keyword is converted to numbers and corresponding numbers in message and keyword are added modulo 26
- “thiscryptosystemisnotsecure. “

Mã Affine

- Một trường hợp đặc biệt của mật mã thay thế là mật mã Affine Cipher,bản rõ và bản mã là e(x) ≡ ax + b mod 26 (a,b € Z_{26}).
- Các hàm này gọi là hàm Affine
- Để có thể giải mã: hàm Affine phải đơn ánh
- Tức là đồng nhất thức: ax + b ≡ y mod 26 phải có nghiệm duy nhất. Vì y thay đổi trên Z_{26} nên y-b cung thay đổi trên Z_{26}. Do vậy, chỉ cần xét phương trình: ax ≡ y mod 26

Mã Affine

- Phương trình này chỉ có một nghiệm duy nhất đối với mỗi y khi và chỉ khi gcd(a,26) = 1.
- Định lý 3.2: đồng dư thức ax ≡ b mod m chỉ có một nghiệm duy nhất x € Z_m với mọi b € Z_m khi và chỉ khi gcd(a,m) = 1
- Định nghĩa3.4: giả sử a ≥ 1 và m ≥ 2 là các số nguyên. gcd(a,m) = 1 thì ta nói a và m là các số nguyên tố cùng nhau. Số các số nguyên trong Z_m nguyên tố cùng nhau với m thường được ký hiệu là φ(m)–hàm phi Euler

Mã Affine

- Định nghĩa 3.5: giả sử a € Z_m. Phần tử nghịch đảo (theo phép nhân) của a là phần tử a^{-1} € Z_m sao cho a. a^{-1} = a^{-1}.a = 1 mod m.
- Định lý 3.3: giả sử $m = \prod_{i=1}^{n} p_i^{e_i}$
- Trong đó các số nguyên p_i khác nhau và e_i>0, 1≤ i ≤ n, khi đó:

$$\varphi(m) = \prod_{i=1} \left(p_i^{e_i} - p_i^{e_i-1}\right)$$

Mã Affine

- Xét phương trình đồng dư y ≡ ax + b mod 26
- Phương trình tương đương: ax ≡ y –b mod26
- Vì gcd(a,26) = 1 nên a có nghịch đảo theo modulo 26.
- a^{-1}(ax) ≡ a^{-1}(y-b)(mod 26)
- Áp dụng tính kết hợp của phép nhân modulo:
- a^{-1}(ax) = (a^{-1}a)x = 1.x
- → x ≡ a^{-1}(y-b)(mod 26)
- Như vậy hàm giải mã là: d(y) ≡ a^{-1}(y-b)mod 26

Mã Affine

- Xét phương trình đồng dư $y \equiv ax + b \bmod 26$
- Phương trình tương đương: $ax \equiv y - b \bmod 26$
- Vì gcd(a,26) = 1 nên a có nghịch đảo theo modulo 26.
- $a^{-1}(ax) \equiv a^{-1}(y-b) \pmod{26}$
- Áp dụng tính kết hợp của phép nhân modulo:
- $a^{-1}(ax) = (a^{-1}a)x = 1.x$
- $\rightarrow x \equiv a^{-1}(y-b) \pmod{26}$
- Như vậy hàm giải mã là: $d(y) \equiv a^{-1}(y-b) \bmod 26$

Mã Affine

- Ví dụ: giả sử $k = (7,3) \rightarrow 7^{-1} \bmod 26 = 15$
- Hàm mã hoá là: $e_k(x) \equiv 7x + 3$
- Và hàm giải mã tương ứng là:

  $d_k(x) = 15(y-3) = 15y - 19$
- $d_k(e_k(x)) = x$ với mọi $x \in Z_{26}$?
- $d_k(e_k(x)) = d_k(7x + 3)$
- $= 15(7x + 3) - 19 = x + 45 - 19 = x$

Phân tích mật mã Affine

- Các số khả nghịch trong Z_{26}
  - 1 có $1^{-1} = 1$
  - 3 có $3^{-1} = 9$
  - 5 có $5^{-1} = 21$
  - 7 có $7^{-1} = 15$
  - 9 có $9^{-1} = 3$
  - 11 có $11^{-1} = 19$
  - 15 có $15^{-1} = 7$
  - 17 có $17^{-1} = 23$
  - 19 có $19^{-1} = 11$
  - 21 có $21^{-1} = 5$
  - 23 có $23^{-1} = 17$
  - 25 có $25^{-1} = 25$

Phân tích mật mã Affine

- Số thứ tự bảng chữ cái Anh ngữ

0	1	2	3	4	5	6	7	8	9	10	11	12
A	B	C	D	E	F	G	H	I	J	K	L	M
13	14	15	16	17	18	19	20	21	22	23	24	25
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Giải thuật phân tích mã Affine

- Thống kê xác suất xuất hiện của các ký tự
- Tìm ký tự xuất hiện nhiều nhất, thử với suy đoán nó là e (trong Anh ngữ).
- Tìm ký tự xuất hiện nhiều kế tiếp, thử với suy đoán nó là t (trong Anh ngữ).
- Giải cặp phương trình trên để tìm a, b với UCLN của a và 26 phải không lớn hơn 1.
- Với cặp a, b tìm được kiểm chứng xem bản rõ dịch được có nghĩa hay không.

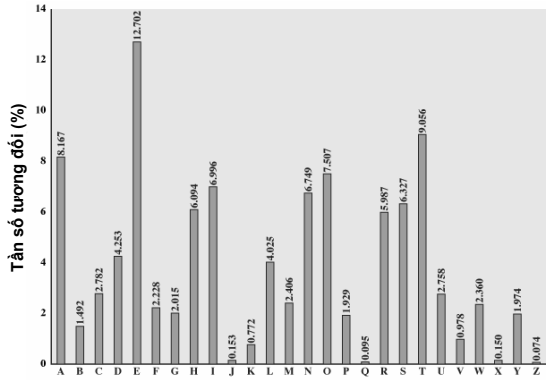
Ví dụ

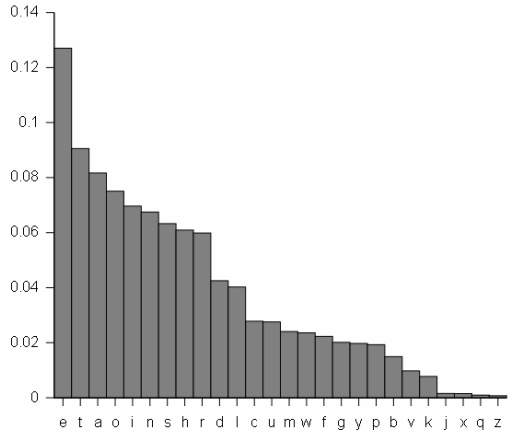
- Phân tích chuỗi mã:
- FMXVEDKAPHFERBNDKRXRSREFMORUD SDKDVSHVUFEDKAPRKDLYEVLRHHRH

Bảng xác suất xuất hiện

Letter	Frequency	Letter	Frequency
A	2	N	1
B	1	O	1
C	0	P	2
D	7	Q	0
E	5	R	8
F	4	S	3
G	0	T	0
H	5	U	2
I	0	V	4
J	0	W	0
K	5	X	2
L	2	Y	1
M	2	Z	0

Các tần số chữ cái tiếng Anh





Phép thử 1

- Chữ E thành chữ **R**: 4a + b = **17** (mod 26)(1)
- Chữ T thành chữ **D**: 19a + b = **3** (mod 26)(2)
- Trừ theo vế (2) cho (1): 15a = -14 (mod 26)
- Hay a = $(-14)15^{-1}$ = (-14)7 = 12.7 = 84 = 6 (mod 26).
- UCLN(a,m) = UCLN(6,26) = 2 không hợp lệ

Phép thử 2

- Chữ E thành chữ **R**: 4a + b = **17** (mod 26)(1)
- Chữ T thành chữ **E**: 19a + b = **4** (mod 26)(2)
- Trừ theo vế (2) cho (1): 15a = -13 (mod 26)
- Hay a = $(-13)15^{-1}$ = (-13)7 = 13.7 = 91 = 13 (mod 26).
- UCLN(a,m) = UCLN(13,26) = 13 không hợp lệ

Phép thử 3

- Chữ E thành chữ **R**: 4a + b = **17** (mod 26)(1)
- Chữ T thành chữ **H**: 19a + b = **7** (mod 26)(2)
- Trừ theo vế (2) cho (1): 15a = -10 (mod 26)
- Hay a = $(-10)15^{-1}$ = (-10)7 = -70 = 8 (mod 26).
- UCLN(a,m) = UCLN(13,26) = 2 không hợp lệ

Phép thử 4

- Chữ E thành chữ **R**: 4a + b = **17** (mod 26)(1)
- Chữ T thành chữ **K**: 19a + b = **10** (mod 26)(2)
- Trừ theo vế (2) cho (1): 15a = -7 (mod 26)
- Hay a = (-7)15^{-1} = (-7)7 = -49 = 3 (mod 26).
- UCLN(a,m) = UCLN(13,26) = 1 là hợp lệ
- Thế a = 3 vào (1) ta được b = 5.

Phép thử 4

- Vậy mã hoá: y = 3x + 5 (mod 26)
- Suy ra: 3x = y – 5 (mod 26)
- x = (y – 5)3^{-1} = (y – 5)9 = 9y – 45 = 9y + 7 (mod 26)
- Giải mã:
- ALGORITHMSAREQUITEGENERALDEFINITIONSOFARITHMETICPROCESSES
- Đọc là: algorithms are quite general definitions of arithmetic processes

c. Mật mã hoán vị

- Ý tưởng của mã hoán vị là giữ các ký tự của bản rõ nhưng không thay đổi nhưng sẽ thay đổi vị trí của chúng bằng cách sắp xếp lại các ký tự này.
- Ví dụ: giả sử m = 6 và khoá là phép hoán vị sau:

1	2	3	4	5	6
3	5	1	6	4	2

c. Mật mã hoán vị

- Khi đó phép hoán vị ngược sẽ là:

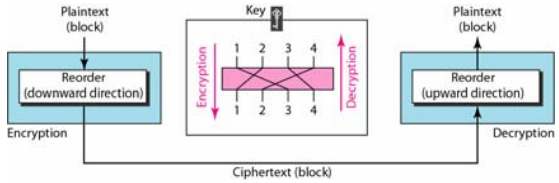
1	2	3	4	5	6
3	6	1	5	2	4

- Giả sử ta có bản rõ: asecondclasscarriageonthetrain → asecon|dclass|carria|geonth|etrain
- Mỗi nhóm 6 chữ cái lại được sắp xếp theo phép hoán vị π, ta có
  EOANCS|LSDSAC|RICARA|OTGHNE|RIENAT
- Cuối cùng có bảng mã sau
  EOANCSLSDSACRICARAOTGHNERIENAT

c. Mật mã hoán vị – định nghĩa

- Cho m là một số nguyên dương xác định nào đó.
- P = C = $(Z_{26})^m$ và cho K là tất cả mọi hoán vị có thể có của { 1,2,..., m }.
- Đối với một khoá π (tức là một phép hoán vị nào đó), ta xác định:
- $e_\pi = (x_1, ..., x_m) = (x_{\pi(1)}, ..., x_{\pi(m)})$
- $d_\pi = (x_1, ..., x_m) = (y_{\pi^{-1}(1)}, ..., y_{\pi^{-1}(m)})$

Ví dụ mật mã hoán vị



- Mã hoá và giải mã theo mật mã hoán vị bản tin: “Hello my dear” theo khóa K trong sơ đồ

Ví dụ: Mật mã Hill

- Do Lester S.Hill đưa ra năm 1929
- Giả sử m là một số nguyên dương, đặt P=C=$(Z_{26})^m$
- ý tưởng: 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: 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).

Mật mã Hill

- Ở đây y_1, y_2 đều là một tổ hợp tuyến tính của x_1, x_2:
  - $y_1 = 11\,x_1 + 3\,x_2$
  - $y_2 = 8\,x_1 + 7\,x_2$
- Hoặc:

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

Mật mã Hill

- Khoá K: ma trận kích thước m x m

$$(y_1 \quad \dots \quad y_m) = (x_1 \quad \dots \quad x_m)\begin{bmatrix} k_{11} & k_{12} & \dots & k_{1m} \\ k_{21} & k_{22} & \dots & k_{2m} \\ \vdots & \vdots & & \vdots \\ k_{m1} & k_{m2} & \dots & k_{mm} \end{bmatrix}$$

Mật mã Hill

- Vấn đề là làm sao để tính x từ y?
- Dùng ma trận nghịch đảo: $x = yk^{-1}$
- Ví dụ: giả sử khoá

$$k = \begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix}$$

- Ma trận nghịch đảo:

$$k^{-1} = \begin{pmatrix} 7 & 18 \\ 23 & 11 \end{pmatrix}$$

Mật mã Hill

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

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

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

Mật mã Hill

- Do vậy bản mã của July là DELW. Để giải mã Bob sẽ tính:
- (3 4).k^{-1}= (9 20)
- Và (11 22) k^{-1}= (11 24)

Lưu ý rằng các phép toán được thực hiện trên Z_{26}

d. Các hệ mật mã tích

- Xét C = P, S1 = (P, P, K1, E1, D1), S2 = (P, P, K2, E2, D2) là hai hệ mật tự đồng cấu có cùng không gian bản mã và bản rõ
- S1 X S2 = (P, P, K1 X K2, E, D)
- Khoá của hệ mật tích có dạng k = (k_1, k_2) với $k_1 \epsilon K_1$ và $k_2 \epsilon K_2$.
- Quy tắc mã hoá: Với mỗi k = (k_1, k_2) ta có một quy tắc mã e_k xác định theo công thức: $e_{(k1,k2)}(x) = e_{k2}(e_{k1}(x))$

d. Các hệ mật mã tích

- Quy tắc giải mã: $d_{(k1,k2)}(y) = d_{k1}(d_{k2}(y))$
- Trước tiên mã hoá x bằng e_{k1} rồi mã lại bản kết quả bằng e_{k2}. Quá trình giải mã thực hiện tương tự nhưng theo thứ tự ngược lại.
- $d_{(k1,k2)}(e_{(k1,k2)}(x)) = d_{(k1,k2)}(e_{k2}(e_{k1}(x)))$
  $= d_{k1}(d_{k2}(e_{k2}(e_{k1}(x))))$
  $= d_{k1}(e_{k1}(x))$
  $= x$

d. Các hệ mật mã tích

- Các hệ mật mã đều có các phân bố xác xuất ứng với các không gian khoá của chúng.
- Chọn k_1 có phân bố xác xuất $p(k_1)$ rồi chọn một cách độc lập k_2 có phân bố xác xuất $p(k_2)$
- $P(k_1,k_2) = p(k_1).\ p(k_2)$

d. Các hệ mật mã dòng

- Trong các hệ mật mã nghiên cứu ở trên, các phần tử liên tiếp của bản rõ đều được mã hóa bằng cùng một khoá k. Chuỗi ký tự mã hoá nhận được có dạng:
- $y = y_1y_2... = e_k(x_1).\ e_k(x_2)...$
- Các hệ mật thuộc dạng này thường được gọi là các mã khối.
- Một ý tưởng khác là tạo ra một dòng khoá $z = z_1z_2...$ để mã hoá một chuỗi ký tự $x = x_1x_2...$

d. Các hệ mật mã dòng

- Trong các hệ mật mã nghiên cứu ở trên, các phần tử liên tiếp của bản tin đều được mã hóa bằng cùng một khoá k. Chuỗi ký tự mã hoá nhận được có dạng:
- $y = y_1y_2... = e_k(x_1).\ e_k(x_2)...$
- Các hệ mật thuộc dạng này thường được gọi là các mã khối.

d. Các hệ thống mật mã dòng

- Một ý tưởng khác là tạo ra một dòng khoá $z = z_1z_2...$ để mã hoá một chuỗi ký tự $x = x_1x_2...$ theo quy tắc: $y = y_1y_2... = e_{z1}(x_1).\ e_{z2}(x_2)...$
- **Hoạt động của mật mã dòng:** giả sử $k \in K$ là khoá và $x = x_1x_2...$ là chuỗi ký tự mã hoá.
- $z_i = f_i(k, x_1, ..., x_{i-1})$
- z_i dùng để mã x_i tạo ra $y_i = e_{iz}(x_i)$.
- Do vậy, để mã hoá chuỗi ký tự $x_1x_2...$ ta phải tính liên tiếp $z_1, y_1, z_2, y_2...$

d. Các hệ thống mật mã dòng

- Việc giải mã chuỗi y_1, y_2 có thể được thực hiện bằng cách tính liên tiếp $z_1, x_1, z_2, x_2...$
- Định nghĩa: mật mã dòng là một bộ (P, C, K, L, F, E, D) thoả mãn các điều kiện sau:
  - P là một tập hữu hạn các bản tin cần mã hoá.
  - C là một tập hữu hạn các bản mã có thể.
  - K là tập hữu hạn các khoá có thể (hoặc không gian khoá)
  - L là tập hữu hạn các bộ chữ của dòng khoá.
  - $F = (f_1f_2...)$ là bộ tạo dòng khoá. Với $i \geq 1$

d. Các hệ thống mật mã dòng

- Định nghĩa (tt):
  - F= ($f_1 f_2$...) là bộ tạo dòng khoá. Với i ≥ 1
  - $f_i : K \times P^{i-1} \rightarrow L$
  - Với mỗi z ϵ L có một quy tắc mã hoá e_z ϵ E và một quy tắc giải mã tương ứng d_z ϵ D. e_z: P, và d_z: C → P là các hàm thoả mãn $d_z(e_z(x))$ mọi bản tin x thuộc 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 ≥ 1

e. Chuẩn mã dữ liệu

- Vào khoảng 1970, tiến sĩ Horst Feistel đã đặt nền móng đầu tiên cho chuẩn mã hóa dữ liệu DES với phương pháp mã hóa FeistelCipher.
- Vào năm 1976 Cơ quan Bảo mật Quốc gia Hoa Kỳ (NSA) đã công nhận DES dựa trên phương pháp Feistel là chuẩn mã hóa dữ liệu [25]. Kích thước khóa của DES ban đầu là 128 bit nhưng tại bản công bố FIPS kích thước khóa được rút xuống còn 56 bit.

e. Chuẩn mã dữ liệu

- Kích thước khối văn bản rõ là 64 bit. DES thực hiện mã hóa dữ liệu qua 16 vòng lặp mã hóa, mỗi vòng sử dụng một khóa chu kỳ 48 bit được tạo ra từ khóa ban đầu có độ dài 56 bit. DES sử dụng 8 bảng hằng số S-box để thao tác.
- Biểu diễn thông điệp nguồn n $x \in P$ bằng dãy 64bit. Khóa *k* có 56 bit. Thực hiện mã hóa theo ba giai đoạn:

e. Chuẩn mã dữ liệu

- **Giai đoạn 1**. Tạo dãy 64 bit 0 *x* bằng cách hoán vị *x* theo hoán vị IP (Initial Permutation).
- Biểu diễn $x_0 = IP(x) = L_0 R_0$, L_0 gồm 32 bit bên trái của x_0, R_0 gồm 32 bit bên phải của *x*0.
- **Giai đoạn 2:** Thực hiện 16 vòng lặp từ 64 bit thu được và 56 bit của khoá *k* (chỉ sử dụng 48 bit của khoá k trong mỗi vòng lặp).
- 64 bit kết quả thu được qua mỗi vòng lặp sẽ là đầu vào cho vòng lặp sau.

e. Chuẩn mã dữ liệu

- Các cặp từ 32 bit *Li*, *Ri* (với i 1≤ *i* ≤16) được xác định theo quy tắc sau:
  - $L_i = R_{i-1}$
  - $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$
- với ⊕ biểu diễn phép toán XOR trên hai dãy bit, *K*1, *K*2, ..., *K*16 là các dãy 48 bit phát sinh từ khóa *K* cho trước (Trên thực tế, mỗi khóa *Ki* được phát sinh
- bằng cách hoán vị các bit trong khóa *K* cho trước).

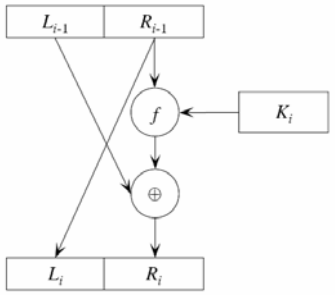
e. Chuẩn mã dữ liệu

- **Giai đoạn 3:** Áp dụng hoán vị ngược IP^{-1} đối với dãy bit $R_{16}\,L_{16}$, thu được từ *y* gồm 64 bit. Như vậy, $y=IP^{-1}(R_{16}\,L_{16})$.
- Hàm *f* được sử dụng ở bước 2 có hai tham số:
  - *Tham số thứ nhất A* là một dãy 32 bit, *tham số thứ hai J* là một dãy 48 bit. Kết quả của hàm *f* là một dãy 32 bit. Các bước xử lý của hàm *f* (*A*,*J*) như sau:

e. Chuẩn mã dữ liệu

- *A* (32 bit) được mở rộng thành dãy 48 bit bằng hàm mở rộng *E*, E(A) là một dãy 48 bit được phát sinh từ *A* bằng cách hoán vị theo một thứ tự nhất định 32 bit của *A*, trong đó có 16 bit của *A* được lặp lại hai lần trong *E*(*A*).
- $E(A) \oplus J$ = B (48 bit)
- Biểu diễn *B* thành từng nhóm 6 bit như sau:
- $B = B_1B_2B_3\,B_4\,B_5B_6$

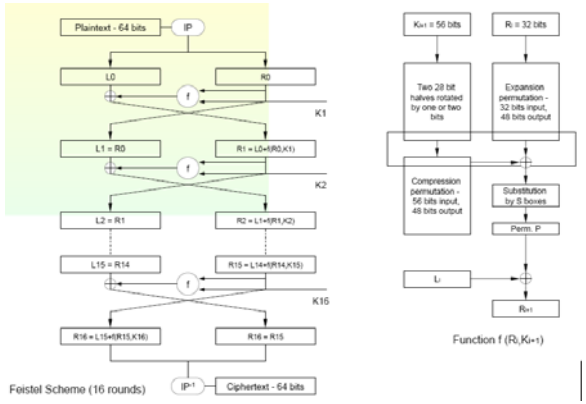
e. Chuẩn mã dữ liệu



e. Chuẩn mã dữ liệu

- *Sử dụng 8 ma trận S_1S_2 ... S_8, kích thước 4x16*
- *Xét dãy $B_j = B_1B_2B_3\ B_4\ B_5B_6$; $S_j(B_j) = s_{rc}$*
- *r: b_1b_6 ; c = $b_2b_3\ b_4\ b_5$*
- *$C_j = S_j(B_j)$, $1 \le j \le 8$*
- *C = $C_1C_2C_3\ C_4\ C_5C_6\ C_7C_8$; hoán vị P của dãy C cho ta hàm F(A,J)*

Giải thuật DES



Khả năng phá mã DES

- Khóa 56 bit có 2^{56} = 7,2 x 10^{16} giá trị có thể
- Phương pháp vét cạn tỏ ra không thực tế
- Tốc độ tính toán cao có thể phá được khóa
  - 1997 : 70000 máy tính phá mã DES trong 96 ngày
  - 1998 : Electronic Frontier Foundation (EFF) phá mã DES bằng máy chuyên dụng (250000$) trong < 3 ngày
  - 1999 : 100000 máy tính phá mã trong 22 giờ
- Thực tế DES vẫn được sử dụng không có vấn đề nhờ các luật an ninh của liên bang.
- Nếu cần an ninh hơn : 3DES hay chuẩn mới AES

3DES

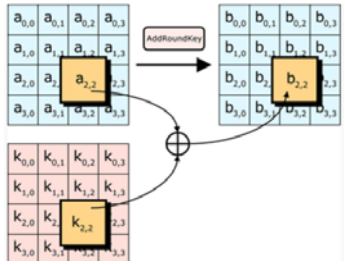
- Sử dụng 3 khóa và chạy 3 lần giải thuật DES
  - Mã hóa : C = $E_{K_3}[D_{K_2}[E_{K_1}[p]]]$
  - Giải mã : p = $D_{K_1}[E_{K_2}[D_{K_3}[C]]]$
- Độ dài khóa thực tế là 168 bit
  - Không tồn tại K_4 = 56 sao cho C = $E_{K_4}(p)$
- Vì sao 3 lần : tránh " **man-in-the-middle attack** "
  - $C = E_{K_2}(E_{K_1}(p)) \Rightarrow X = E_{K_1}(p) = D_{K_2}(C)$
  - Nếu biết một cặp (p, C)
    - Mã hóa p với 2^{56} khóa và giải mã C với 2^{56} khóa
    - So sánh tìm ra K_1 và K_2 tương ứng
    - Kiểm tra lại với 1 cặp (p, C) mới; nếu OK thì K_1 và K_2 là khóa

f. AES chuẩn mã hoá tiên tiến

- Advanced encryption standard.
- Được chấp nhận làm tiêu chuẩn liên bang sau 5 năm tiêu chuẩn hoá bởi NIST
- Tác giả: Joan Daemen và Vincent Rijmen
- Còn được đặt tên Rijndael
- Trên thực tế AES và Rijndael khác nhau.
- DES dùng mạng Feistel, Rijndael dùng mạng thay thế – hoán vị

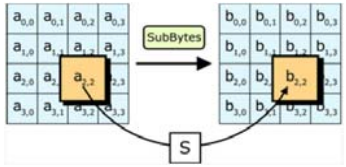
f. AES – mô tả thuật toán

- Addround key



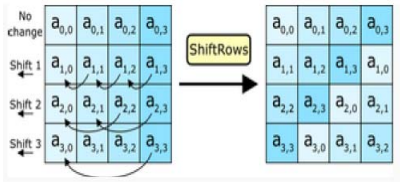
f. AES – mô tả thuật toán

- SubByte



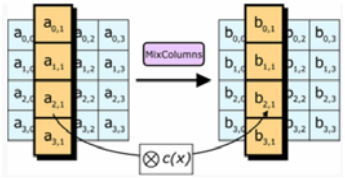
f. AES – mô tả thuật toán

- ShiftRows



f. AES – mô tả thuật toán

- MixColumns

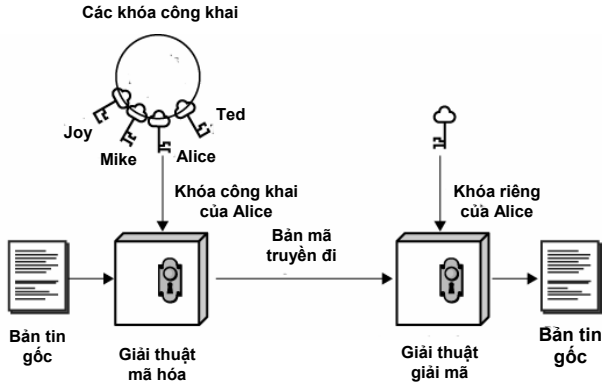


Chương 3: Mật mã hoá công khai

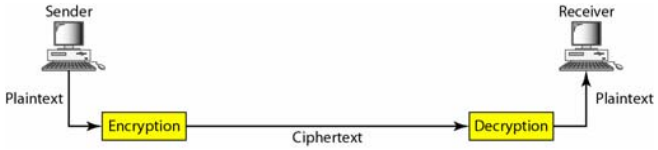
3.1Những khái niệm về mật mã hoá khoá công khai

- Còn gọi là mật mã hai khóa hay bất đối xứng
- Các giải thuật khóa công khai sử dụng 2 khóa
  - Một khóa công khai
    - Ai cũng có thể biết
    - Dùng để mã hóa thông báo và thẩm tra chữ ký
  - Một khóa riêng
    - Chỉ nơi giữ được biết
    - Dùng để giải mã thông báo và ký (tạo ra) chữ ký
- Có tính bất đối xứng
  - Bên mã hóa không thể giải mã thông báo
  - Bên thẩm tra không thể tạo chữ ký

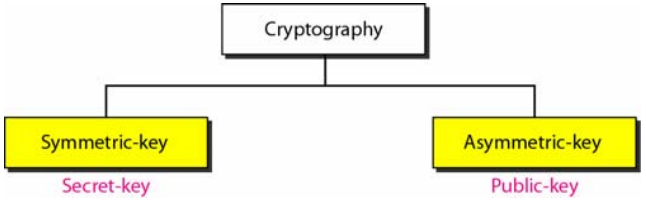
3.1 Mã hóa khóa công khai



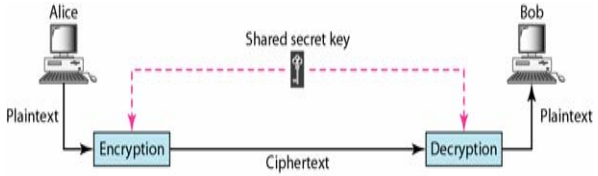
Mô hình mật mã hoá



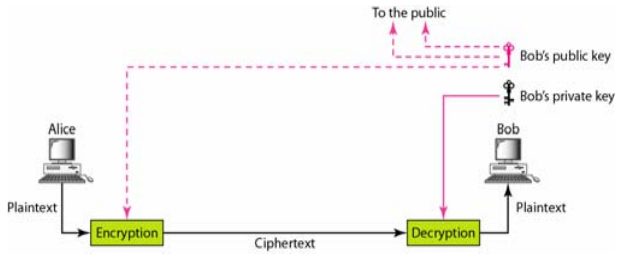
Hai loại mật mã hoá



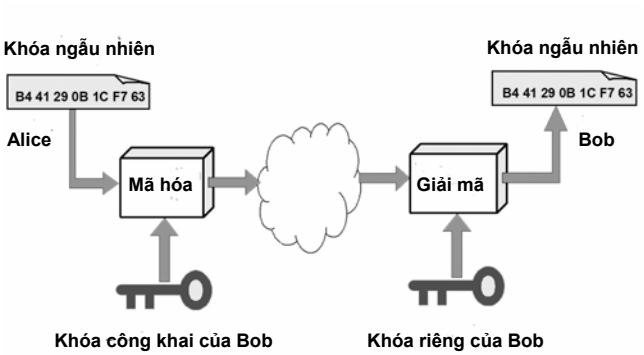
Mô hình mật mã cổ điển/đối xứng



Mô hình mật mã hoá khoá công khai



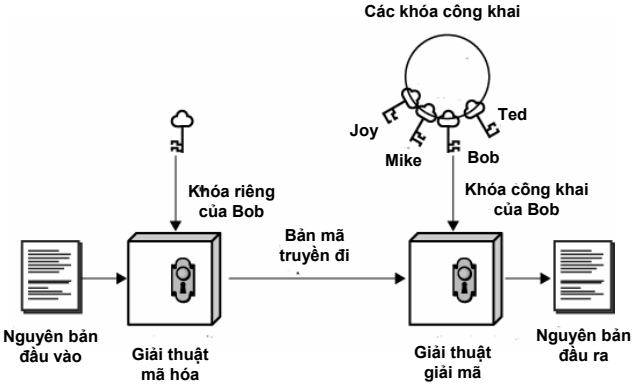
a. Trao đổi khóa



3.1 Những khái niệm về mật mã hoá công khai

- Những hạn chế của mật mã đối xứng
  - Vấn đề phân phối khóa
    - Khó đảm bảo chia sẻ mà không làm lộ khóa bí mật
    - Trung tâm phân phối khóa có thể bị tấn công
  - Không thích hợp cho chữ ký số
    - Bên nhận có thể làm giả thông báo nói nhận được từ bên gửi
- Mật mã khóa công khai đề xuất bởi Whitfield Diffie và Martin Hellman vào năm 1976
  - Khắc phục những hạn chế của mật mã đối xứng
  - Có thể coi là bước đột phá quan trọng nhất trong lịch sử của ngành mật mã
  - Bổ sung chứ không thay thế mật mã đối xứng

3.1.1 Xác thực



3.1.2 Ứng dụng mật mã khóa công khai

- Có thể phân ra 3 loại ứng dụng
  - Mã hóa/giải mã
    - Đảm bảo sự bí mật của thông tin
  - Chữ ký số
    - Hỗ trợ xác thực văn bản
  - Trao đổi khóa
    - Cho phép chia sẻ khóa phiên trong mã hóa đối xứng
- Một số giải thuật khóa công khai thích hợp cho cả 3 loại ứng dụng; một số khác chỉ có thể dùng cho 1 hay 2 loại

a. Các điều kiện cần thiết

- Bên B dễ dàng tạo ra được cặp (KU_b, KR_b)
- Bên A dễ dàng tạo ra được C = $E_{KU_b}(M)$
- Bên B dễ dàng giải mã M = $D_{KR_b}(C)$
- Đối thủ không thể xác định được KR_b khi biết KU_b
- Đối thủ không thể xác định được M khi biết KU_b và C
- Một trong hai khóa có thể dùng mã hóa trong khi khóa kia có thể dùng giải mã
  - $M = D_{KR_b}(E_{KU_b}(M)) = D_{KU_b}(E_{KR_b}(M))$
  - Không thực sự cần thiết

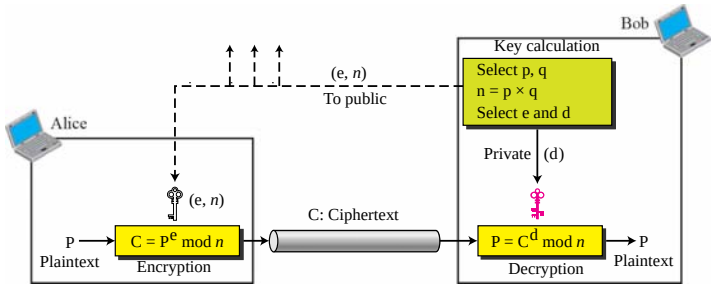
3.2 Số học modulo

- x mod n = phần dư của x khi chia x cho n
- Mệnh đề:
  [(a mod n) + (b mod n)] mod n = (a+b) mod n
  [(a mod n) - (b mod n)] mod n = (a-b) mod n
  [(a mod n) * (b mod n)] mod n = (a*b) mod n
- Do đó:
  $(a \bmod n)^d \bmod n = a^d \bmod n$
- Ví dụ: x=14, n=10, d=2:
  $(x \bmod n)^d \bmod n = 4^2 \bmod 10 = 6$
  $x^d = 14^2 = 196 \quad x^d \bmod 10 = 6$

3.3 Hệ thống mật mã hóa RSA

- Đề xuất bởi Ron Rivest, Adi Shamir và Len Adleman (MIT) vào năm 1977
- Hệ mã hóa khóa công khai phổ dụng nhất
- Mã hóa khối với mỗi khối là một số nguyên < n
  - Thường kích cỡ n là 1024 bit ≈ 309 chữ số thập phân
- Đăng ký bản quyền năm 1983, hết hạn năm 2000
- An ninh vì chi phí phân tích thừa số của một số nguyên lớn là rất lớn

Mật mã RSA – mô hình hoạt động



Tạo khóa RSA

- Mỗi bên tự tạo ra một cặp khóa công khai - khóa riêng theo các bước sau :
  - Chọn ngẫu nhiên 2 số nguyên tố đủ lớn $p \neq q$
  - Tính $n = pq$
  - Tính $\Phi(n) = (p-1)(q-1)$
  - Chọn ngẫu nhiên khóa mã hóa e sao cho $1 < e < \Phi(n)$ và $\gcd(e, \Phi(n)) = 1$
  - Tìm khóa giải mã $d \leq n$ thỏa mãn $e.d \equiv 1 \bmod \Phi(n)$
- Công bố khóa mã hóa công khai $K_p = \{e, n\}$
- Giữ bí mật khóa giải mã riêng $K_e = \{d, n\}$
  - Các giá trị bí mật p và q bị hủy bỏ

Thực hiện RSA

- Để mã hóa 1 thông báo nguyên bản M, bên gửi thực hiện
  - Lấy khóa công khai của bên nhận $KU = \{e, n\}$
  - Tính $C = M^e \bmod n$
- Để giải mã bản mã C nhận được, bên nhận thực hiện
  - Sử dụng khóa riêng $KR = \{d, n\}$
  - Tính $M = C^d \bmod n$
- Lưu ý là thông báo M phải nhỏ hơn n
  - Phân thành nhiều khối nếu cần

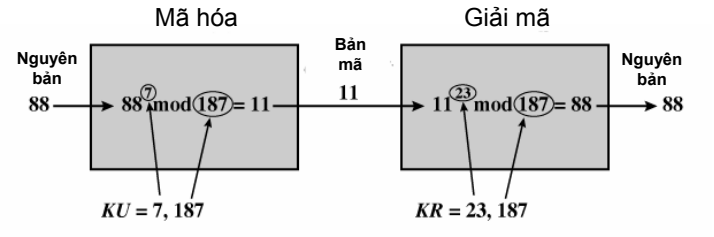
Vì sao RSA khả thi

- Theo định lý Euler
  - $\forall$ a, n : $\gcd(a, n) = 1 \Rightarrow a^{\Phi(n)} \bmod n = 1$
  - $\Phi(n)$ là số các số nguyên dương nhỏ hơn n và nguyên tố cùng nhau với n
- Đối với RSA có
  - $n = pq$ với p và q là các số nguyên tố
  - $\Phi(n) = (p - 1)(q - 1)$
  - $ed \equiv 1 \bmod \Phi(n) \Rightarrow \exists$ số nguyên k : $ed = k\Phi(n) + 1$
  - $M < n$
- Có thể suy ra
  - $C^d \bmod n = M^{ed} \bmod n = M^{k\Phi(n)+1} \bmod n = M \bmod n = M$

Ví dụ tạo khóa RSA

- Chọn 2 số nguyên tố p = 17 và q = 11
- Tính $n = pq = 17 \times 11 = 187$
- Tính $\Phi(n) = (p - 1)(q - 1) = 16 \times 10 = 160$
- Chọn e : $\gcd(e, 160) = 1$ và $1 < e < 160$; lấy e = 7
- Xác định d : $de \equiv 1 \bmod 160$ và $d \leq 187$
  Giá trị d = 23 vì $23 \times 7 = 161 = 1 \times 160 + 1$
- Công bố khóa công khai $KU = \{7, 187\}$
- Giữ bí mật khóa riêng $KR = \{23, 187\}$
  - Hủy bỏ các giá trị bí mật p = 17 và q = 11

Ví dụ thực hiện RSA



Chọn tham số RSA

- Cần chọn p và q đủ lớn
- Thường chọn e nhỏ
- Thường có thể chọn cùng giá trị của e cho tất cả người dùng
- Trước đây khuyến nghị giá trị của e là 3, nhưng hiện nay được coi là quá nhỏ
- Thường chọn e = $2^{16} - 1 = 65535$
- Giá trị của d sẽ lớn và khó đoán

An ninh của RSA

- Khóa 128 bit là một số giữa 1 và một số rất lớn
  340.282.366.920.938.000.000.000.000.000.000.000.000
- Có bao nhiêu số nguyên tố giữa 1 và số này
  ≈ n / ln(n) = 2^{128} / ln(2^{128}) ≈
  3.835.341.275.459.350.000.000.000.000.000.000.000
- Cần bao nhiêu thời gian nếu mỗi giây có thể tính được 10^{12} số
  Hơn 121,617,874,031,562,000 năm (khoảng 10 triệu lần tuổi của vũ trụ)
- An ninh nhưng cần đề phòng những điểm yếu

Phá mã RSA

- Phương pháp vét cạn
  - Thử tất cả các khóa riêng có thể
    - Phụ thuộc vào độ dài khóa
- Phương pháp phân tích toán học
  - Phân n thành tích 2 số nguyên tố p và q
  - Xác định trực tiếp Φ(n) không thông qua p và q
  - Xác định trực tiếp d không thông qua Φ(n)
- Phương pháp phân tích thời gian
  - Dựa trên việc đo thời gian giải mã
  - Có thể ngăn ngừa bằng cách làm nhiễu

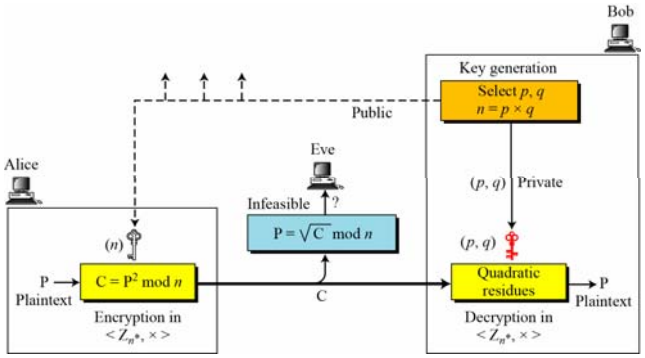
Phân tích thừa số RSA

- An ninh của RSA dựa trên độ phức tạp của việc phân tích thừa số n
- Thời gian cần thiết để phân tích thừa số một số lớn tăng theo hàm mũ với số bit của số đó
  - Mất nhiều năm khi số chữ số thập phân của n vượt quá 100 (giả sử làm 1 phép tính nhị phân mất 1 ηs)
- Kích thước khóa lớn đảm bảo an ninh cho RSA
  - Từ 1024 bit trở lên
  - Gần đây nhất năm 1999 đã phá mã được 512 bit (155 chữ số thập phân)

Mật mã Rabin

- Mật mã hoá Rabin có thể coi như RSA với giá trị của e và d cố định. Bản mã $C \equiv P^2 \pmod n$ và bản rõ là $P \equiv C^{1/2} \pmod n$

Mật mã Rabin



Mật mã Rabin – tạo khoá

- Chọn hai số nguyên tố p, q có kích thước xấp xỉ nhau sao cho $p \equiv q \equiv 3 \pmod 4$.
- Tính n = p.q
- Khoá công khai là n, khoá bí mật là (p,q)

Mật mã Rabin – mã hoá

- Alice nhận khoá công khai của Bob.
- Biểu thị bản tin dưới dạng một số nguyên. nằm trong dải [0, n-1].
- Tính $C = P^2 \pmod n$.
- Gửi bản mã cho Bob

Mật mã Rabin – giải mã

- Bob phải tìm 4 căn bậc 2 của C mod n.
- Sử dụng thuật toán Euclide mở rộng để tính a, b sao cho ap + bq = 1.
- Tính $r = c^{(p+1)/4} \bmod p$.
- Tính $s = c^{(q+1)/4} \bmod q$.
- Tính x = (aps + bqr) mod n.
- Tính y = (aps – bqr) mod n.
- Bốn giá trị căn bậc 2 của c mod n là x,-x mod n, y và –y mod n.

Mật mã Rabin – ví dụ

- Mã hoá và giải mã với bản rõ x = 9, p = 7, q = 11.
- X = 25, p = 19, q = 23
- p = 23, q = 7

3.5.1 Mật mã hoá Merkle Hellman- giới thiệu

- Định nghĩa dãy siêu tăng: Dãy các số nguyên dương $a_1, a_2, \ldots a_n$ được gọi là siêu tăng nếu $a_i > a_1 + a_2 + \ldots + a_{i-1}$ với mọi $2 \le i \le n$
- Bài toán xếp balo: xếp một đống các gói có trọng lượng khác nhau vào ba lô để ba lô có một trọng lượng cho trước.

3.5.1 Mật mã hoá Merkle Hellman

- Trên phương diện toán học: cho tập các giá trị $M_1, M_2, ..., M_n$ và một tổng S. Hãy tính các giá trị b_i để:
  - $S = b_1 M_1 + b_2 M_2 + ... + b_n M_n$
  - với b ϵ [0,1]
  - $b_i = 1$: gói M_i được xếp vào ba lô
  - $b_i = 0$: gói M_i không khđược xếp vào ba lô

3.5.2 Bài toán xấp ba lô

- Giải bài toán xếp ba lô trong trường hợp dãy $M = \{M_1, M_2, ..., M_n\}$ là dãy siêu tăng.
- Việc tìm $b = (b_1, b_2, ..., b_n)$ tương đương với bài toán tìm biểu diễn nhị phân của S
- Giải thuật:
  - Vào: Dãy siêu tăng $M = \{M_1, M_2, ..., M_n\}$ và một số nguyên S là tổng của một tập con trong M
  - Ra: $(b_1, b_2, ..., b_n)$ với $b_i = 0,1$ sao cho:
    $S = b_1 M_1 + b_2 M_2 + ... + b_n M_n$

3.5.3 Giải thuật Merkle Hellman

- (1) i←n
- (2) While i ≥ 1 do
  - a. If $S \geq M_i$ then $x_i \leftarrow 1$ và $S \leftarrow S - M_i$
    - Else $x_i \leftarrow 0$
  - b. $i \leftarrow i - 1$
- (3) return (b)
- Nếu M không phải dãy siêu tăng thì lời giải của bài toán là một trong 2^n phương án có thể xảy ra. Điều này không dễ với n lớn!!!

3.5.3 Mật mã hoá Merkle Hellman – tạo khoá

1) Chọn một dãy siêu tăng $M_1, M_2, ..., M_n$ và một modul $M > M_1, M_2, ..., M_n$
2) Chọn một số nguyên ngẫu nhiên W sao cho $1 \leq W \leq M - 1$, (W, M) =1
3) Chọn một phép hoán vị ngẫu nhiên π của các số nguyên { 1, 2, ..., n}
4) Tính ai = $WM_{\pi(i)}$ mod M với i = 1, 2, .., n
5) Khoá công khai là tập các số $(a_1, a_2, ..., a_n)$
6) Khoá bí mật là $(\pi, M, W(M_1, M_2, ..., M_n))$

3.5.4 Mật mã hoá Merkle Hellman-thuật toán mã khoá công khai

- B mã hoá bản tin m để gửi cho A bản mã cần giải mã.
- Mã hoá
  - nhận khoá công khai của A: $(a_1, a_2, ..., a_n)$
  - biểu thị bản tin m như một chuỗi nhị phân có độ dài n. Với $m = m_1, m_2, ..., m_n$
  - tính số nguyên $c= m_1a_1+ m_2a_2 + ...+ m_na_n$
  - Gửi bản mã cho A

3.5.5 Giải mã Merkle Hellman

- Giải mã:
  - Tính $s = W-1c \bmod M$
  - Sử dụng giải thuật xếp balo trong trường hợp dãy siêu tăng để tìm các số nguyên $r_1, r_2, ..., r_n \epsilon \{0,1\}$ sao cho: $d = r_1 M_1+ r_2M_2 + ...+ r_n M_n$
  - Các bit của bản rõ là $m_i = r_{\pi(i)}$, i = 1,2, ..., n

3.6 Thuật toán trao đổi khoá Diffie Hellman

- Giải thuật mật mã khóa công khai đầu tiên
- Đề xuất bởi Whitfield Diffie và Martin Hellman vào năm 1976
  - Malcolm Williamson (GCHQ - Anh) phát hiện trước mấy năm nhưng đến năm 1997 mới công bố
- Chỉ dùng để trao đổi khóa bí mật một cách an ninh trên các kêch thông tin không an ninh
- Khóa bí mật được tính toán bởi cả hai bên
- An ninh phụ thuộc vào độ phức tạp của việc tính logarithm rời rạc

3.6.1 Thiết lập Diffie-Hellman

- Các bên thống nhất với nhau các tham số chung
  - q là một số nguyên tố đủ lớn
  - α là một nguyên căn của q
    - $\alpha \bmod q$, $\alpha^2 \bmod q$,..., $\alpha^{q-1} \bmod q$ là các số nguyên giao hoán của các số từ 1 đến q - 1
- Bên A
  - Chọn ngẫu nhiên làm khóa riêng $X_A < q$
  - Tính khóa công khai $Y_A = \alpha^{X_A} \bmod q$
- Bên B
  - Chọn ngẫu nhiên làm khóa riêng $X_B < q$
  - Tính khóa công khai $Y_B = \alpha^{X_B} \bmod q$

3.6.2 Trao đổi khóa Diffie-Hellman

- Tính toán khóa bí mật
  - Bên A biết khóa riêng X_A và khóa công khai Y_B
    $K = Y_B^{X_A} \bmod q$
  - Bên B biết khóa riêng X_B và khóa công khai Y_A
    $K = Y_A^{X_B} \bmod q$
- Chứng minh

$$Y_A^{X_B} \bmod q = (\alpha^{X_A} \bmod q)^{X_B} \bmod q$$
$$= \alpha^{X_A X_B} \bmod q$$
$$= \alpha^{X_B X_A} \bmod q$$
$$= (\alpha^{X_B} \bmod q)^{X_A} \bmod q$$
$$= Y_B^{X_A} \bmod q$$

Ví dụ Diffie-Hellman

- Alice và Bob muốn trao đổi khóa bí mật
- Cùng chọn q = 353 và $\alpha = 3$
- Chọn ngẫu nhiên các khóa riêng
  - Alice chọn $X_A = 97$, Bob chọn $X_B = 233$
- Tính toán các khóa công khai
  - $Y_A = 3^{97} \bmod 353 = 40$ (Alice)
  - $Y_B = 3^{233} \bmod 353 = 248$ (Bob)
- Tính toán khóa bí mật chung
  - $K = Y_B^{X_A} \bmod 353 = 248^{97} \bmod 353 = 160$ (Alice)
  - $K = Y_A^{X_B} \bmod 353 = 40^{233} \bmod 353 = 160$ (Bob)

3.7 Mật mã hoá trên đường cong elliptic

- Elliptic Curve Cryptography - ECC
- Mã hoá công khai dựa trên độ khó của các bài toán khó giải quyết.
- Cho đến nay chỉ còn hai bài toán:
  - Bài toán logarithm rời rạc
  - Bài toán phân tích thừa số của số nguyên.
- 1985: Neal Koblitz và Victor S.Miller đã độc lập nghiên cứu và đưa ra đề xuất ứng dụng lý thuyết đường cong elliptic trên trường hữu hạn.
- Được phát hiện lần đầu tiên vào thế kỷ 17 dưới dạng công thức Diophantine: $y_2 - x_3 = C$ với $C \in Z$

3.7.1 Khái niệm về đường cong elliptic

- 3.7.1.1 Công thức Weierstrase
- 3.7.1.2 Đường cong elliptic trên trường R^2
- 3.7.1.3 Đường cong elliptic trên trường hữu hạn
- 3.7.1.4 Bài toán logarithm rời rạc trên đường cong elliptic (ECDLP)

3.7.1.1 Công thức Weierstrase

- Đường cong elliptic E(K) được định nghĩa trên trường K bằng công thức Weierstrase:
- $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ trong đó $a_1, a_2, a_3, a_4, a_5, a_6 \in K$
- Số các điểm nguyên trên E(K) ký hiệu là #E hoặc #E(K)

3.7.1.2 Đường cong elliptic trên trường R^2

- Đường cong elliptic trên trường số thực R^2 là tập hợp các điểm (x,y) thoả mãn:
  - $y^2 = x^3 + a_4x + a_6$
  - Điểm O tại vô cực.
- Phép cộng:
  - điểm tại vô cực O là điểm cộn với điểm nào cũng ra chính điểm đó.
  - $P + (-P) = (x,y) + (x,-y) = O$

Thuật toán cộng trên đường cong Elliptic

- Input:
  - E(R) với các tham số $a_4x + a_6$
  - Điểm $P(x_1,y_1) \in E(R)$ và $Q(x_2,y_2) \in E(R)$
- Output:
  - $R = P + Q$, $R = (x_3,y_3) \in E(R)$
  - If P = O then R← Q
  - If Q = O then R← P
  - If $x_1 = x_2$ then
    - If $y_1 = y_2$ then $\theta \leftarrow (3x_1^2 + a_4)/2y_1$
    - Else if $y_1 = -y_2$ then R← O
  - Else $\theta \leftarrow (y_2 - y_1)/(x_2 - x_1)$
  - End if

Thuật toán cộng trên đường cong Elliptic

- $X3 = \theta^2 \leftarrow (3x_1^2 + a_4)/2y_1$

3.8 Mật mã hoá McElice

- Cho G là một ma trận sinh của mã Goppa
- C[n,k,d], với n = 2m, d = 2t + 1, k = n – mt
- S là một ma trận khả nghịch cấp k x k trên Z_2
- P là ma trận hoán vị cấp n x n
- Đặt G' = S G P
- Cho P = $(Z_2)^2$ và ký hiệu K = {(G, S, P, G')}
- G, S, P được giữ bí mật
- G' được công khai

3.8 Mật mã hoá McElice

- Với K = {(G, S, P, G')} ta xây định nghĩa: $e_k(x, e) = xG' + e$ với e ϵ $(Z_2)^n$ và là một vector ngẫu nhiên có trọng số t
- Bob giải bản mã y ϵ $(Z_2)^n$ theo các bước sau:
  - Tính $y_1 = yP^{-1}$
  - $y_1 = x_1 + e_1$, x_1 ϵ C
  - Tính x_0 ϵ $(Z_2)^k$ sao cho $x_0G = x_1$
  - Tính $x = x_0S^{-1}$