Chương 4

HỆ MÃ HÓA ĐỐI XỨNG

Giáo viên: Lê Quốc Anh

Nội dung

1. Hệ mã hóa hiện đại
2. Hệ mã hóa dòng (Stream cipher)
3. Hệ mã hóa khối (Block cipher)

1. Hệ mã hóa hiện đại

- **Mã hóa hiện đại (modern cryptography):** mã hóa đối xứng (symmetric cipher, secret key cryptography – 1 khóa), bất đối xứng (asymmetric cipher, public key cryptography – 2 khóa), hàm băm (hash functions – không có khóa).

Slide: 3

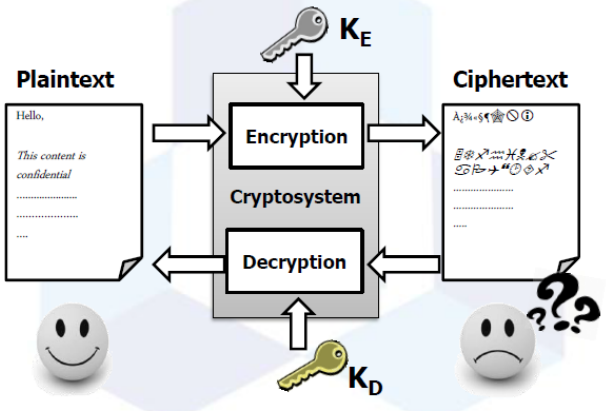
1. Hệ mã hóa hiện đại

- ***Hệ thống mã hóa đối xứng (Symmetric cryptosystem)*** là hệ thống mã hóa sử dụng một khóa bí mật chia sẻ (shared-secret-key) cho cả hai quá trình mã hóa và giải mã.
- ***Hệ thống mã hóa bất đối xứng (Asymmetric cryptosystem)*** là hệ thống mã hóa sử dụng một khóa công khai (public key) và một khóa bí mật (private key) cho quá trình mã hóa và giải mã.
- Hệ thống mã hóa bất đối xứng còn được gọi là hệ thống mã hóa khóa công khai (public-key cryptosystem)

Slide: 4

1. Hệ mã hóa hiện đại

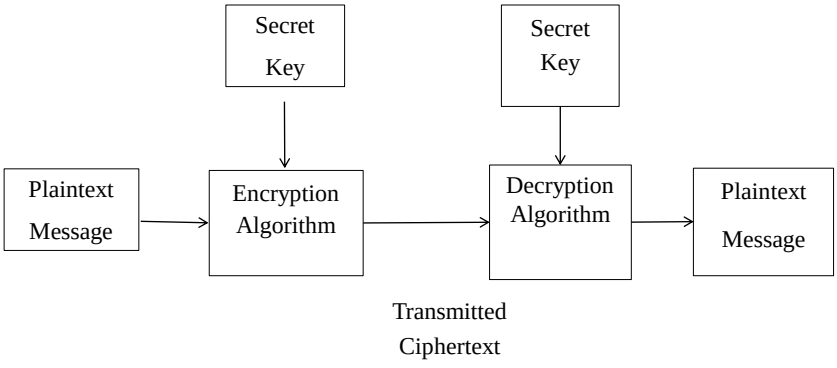
- Mã hóa đối xứng: $K_E = K_D$
- Mã hóa bất đối xứng: $K_E \neq K_D$



Slide: 5

Mã hóa đối xứng (symmetric cipher)

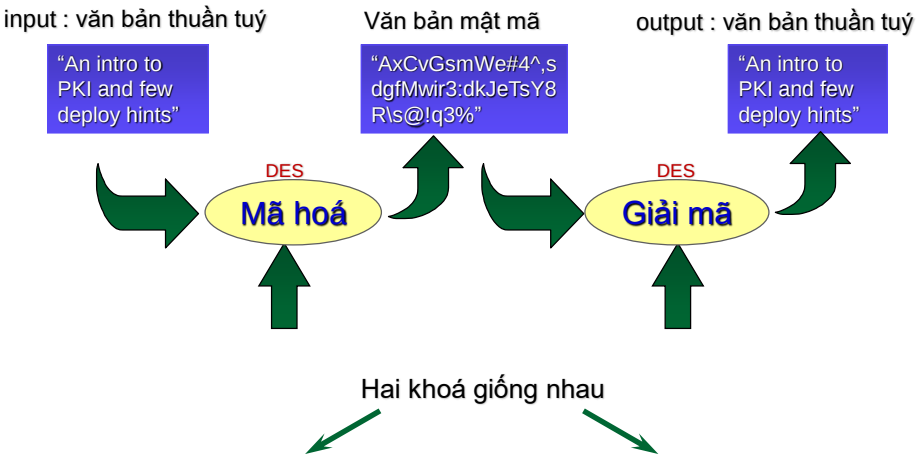
Sơ đồ mã hóa đối xứng



Một số thuật toán mã hóa đối xứng: DES, AES, Blowfish, RC5, RC6 …

Slide: 6

Mã hóa đối xứng (symmetric cipher)



Slide: 7

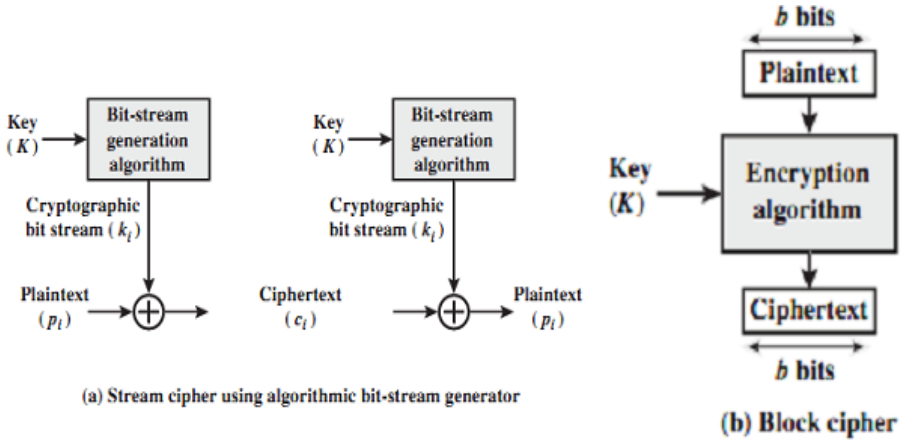
Phân loại hệ mã đối xứng

- **Mã dòng (stream cipher)** là một kỹ thuật mã hóa mà mã hóa một dòng dữ liệu số một bit hoặc một byte tại một thời điểm.

- **Mã khối (block cipher)** là một cơ chế mã hóa/giải mã mà trong đó một khối của bản rõ được xử lý như một tổng thể và dùng để tạo ra khối bản mã có độ dài bằng nhau. Thông thường kích cỡ khối là 64 hoặc 128 bit được sử dụng.

Slide: 8

Stream Ciphers và Block Ciphers



Slide: 9

Stream Ciphers

Mã dòng có các đặc tính sau:

- Kích thước một đơn vị mã hóa: gồm k bít. Bản rõ được chia thành các đơn vị mã hóa:

$$P \rightarrow p_0 p_1 p_2 \dots p_{n-1} \qquad (p_i : k \text{ bít})$$

- Một bộ sinh dãy số ngẫu nhiên: dùng một khóa K ban đầu để sinh ra các số ngẫu nhiên có kích thước bằng kích thước đơn vị mã hóa:

$$StreamCipher(K) \rightarrow S = s_0 s_1 s_2 \dots s_{n-1} \qquad (s_i : k \text{ bít})$$

- Mỗi số ngẫu nhiên được XOR với đơn vị mã hóa của bản rõ để có được bản mã.

$$c_0 = p_0 \oplus s_0,\ c_1 = p_1 \oplus s_1 \dots \ ; C = c_0 c_1 c_2 \dots c_{n-1}$$

Slide: 10

Stream Ciphers

- Quá trình giải mã được thực hiện ngược lại, bản mã *C* được XOR với dãy số ngẫu nhiên *S* để cho ra lại bản rõ ban đầu:
- Trong ví dụ p=111100000011 đơn vị mã hóa có chiều dài k = 4 bít, n = 3:

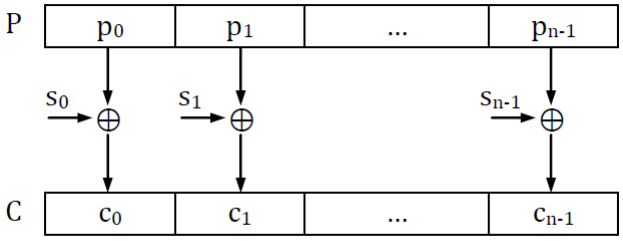
$$p_0 = 1111,\ p_1 = 0000,\ p_2 = 0011$$
$$s_0 = s_1 = s_2 = K = 0101$$
$$c_0 = 1010,\ c_1 = 0101,\ c_2 = 0110$$

Ví dụ này không phải là mã dòng vì s_0, s_1, s_2 lặp lại khóa K

Slide: 11

Stream Ciphers

- Đối với mã dòng, các số *si* được sinh ra phải đảm bảo một độ ngẫu nhiên nào đó (chu kỳ tuần hoàn dài):



Mô hình mã dòng

Điểm quan trọng nhất của các mã dòng là bộ sinh số ngẫu nhiên

Slide: 12

Stream Ciphers

- Quá trình giải mã được thực hiện ngược lại, bản mã *C* được XOR với dãy số ngẫu nhiên *S* để cho ra lại bản rõ ban đầu:
- Trong ví dụ p=111100000011 đơn vị mã hóa có chiều dài k = 4 bít, n = 3, với các khóa phát sinh ngẫu nhiên sau: s0=0101, s1=1010, s2=1100, s3=0011

Cho biết kết quả sau khi mã hóa

Slide: 13

Các hệ mã dòng

- ***Chú ý:*** 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 loại trừ (XOR).
- Bảng chân lý phép cộng theo modulo 2 giống như bảng chân lý của phép toán XOR

a	b	$c = a + b \bmod 2$
0	0	$0 + 0 = 0 \bmod 2$
0	1	$0 + 1 = 1 \bmod 2$
1	0	$1 + 0 = 1 \bmod 2$
1	1	$1 + 1 = 0 \bmod 2$

Slide: 14

Các hệ mã dòng

- Hàm mã hóa và giải mã được thực hiện bởi cùng một phép toán là phép cộng theo modulo 2(hay phép XOR)
- Vì:

$$\text{Decryption: } y_i + z_i = \underbrace{(x_i + z_i)}_{\text{encryption}} + z_i = x_i + (z_i + z_i) \equiv x_i \bmod 2.$$

- Trong đó với z_i=0 và z_i=1 thì

$$z_i + z_i \equiv 0 \bmod 2$$

Slide: 15

Các hệ mã dòng

- Ví dụ: mã hóa ký tự 'A' bởi Alice
- Ký tự 'A' trong bảng mã ASCII được tướng ứng với mã $65_{10}=1000001_2$ được mã hóa bởi hệ khóa $z_1,\ldots,z_7=0101101$
- Hàm mã hóa:
- Hàm giải mã:

Slide: 16

Hệ mã dòng A5/1

- A5/1 được dùng trong mạng điện thoại GSM, để bảo mật dữ liệu trong quá trình liên lạc giữa máy điện thoại và trạm thu phát sóng vô tuyến.
- Đơn vị mã hóa của A5/1 là một bít.
- Bộ sinh số mỗi lần sẽ sinh ra hoặc bít 0 hoặc bít 1 để sử dụng trong phép XOR.

Slide: 17

Tiny A5/1

- Mô hình thu nhỏ của A5/1 gọi là TinyA5/1.
- Cơ chế thực hiện của bộ sinh số TinyA5/1 là như sau:
- Bộ sinh số gồm 3 thanh ghi X, Y, Z.
  - Thanh ghi X gồm 6 bit, ký hiệu là $(x_0, x_1, \ldots, x_5)$.
  - Thanh ghi Y gồm 8 bit $(y_0, y_1, \ldots, y_7)$.
  - Thanh ghi Z lưu 9 bit $(z_0, z_1, \ldots, z_8)$.
- Khóa K ban đầu có chiều dài 23 bít và lần lượt được phân bố vào các thanh ghi: K -> XYZ

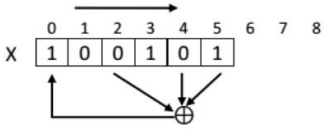
Slide: 18

Tiny A5/1

– Các thanh ghi X, Y, Z được biến đổi theo 3 quy tắc:

1) ***Quay X*** gồm các thao tác:

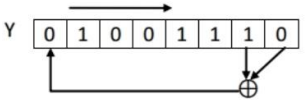
- $t = x_2 \oplus x_4 \oplus x_5$
- $x_j = x_{j-1}$ với j = 5, 4, 3, 2, 1
- $x_0 = t$



Ví dụ: giả sử X là 100101, dẫn đến t = 0⊕0⊕1 = 1, vậy sau khi quay giá trị của X là 110010.

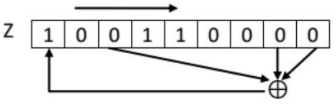
2) ***Quay Y***: tương tự như quay X, quay Y là như sau:

- $t = y_6 \oplus y_7$
- $y_j = y_{j-1}$ với $j = 7, 6, 5, ..., 1$
- $y_0 = t$



3) **Quay Z**:

- $t = z_2 \oplus z_7 \oplus z_8$
- $z_j = z_{j-1}$ với $j = 8, 7, 6, ..., 1$
- $z_0 = t$



Tiny A5/1

– Hàm **maj(x, y, z)** nếu trong 3 bít x, y, z có từ hai bít 0 trở lên thì hàm trả về giá trị 0, nếu không hàm trả về giá trị 1.

– Tại bước sinh số thứ i, các phép tính sau được thực hiện:

- m = **maj(x_1, y_3, z_3)**
- If x_1 = m then thực hiện quay X
- If y_3 = m then thực hiện quay Y
- If z_3 = m then thực hiện quay Z

– Và bít được sinh ra là: $s_i = x_5 \oplus y_7 \oplus z_8$

– Bít s_i được XOR với bít thứ i trong bản rõ để có được bít thứ i trong bản mã theo quy tắc của mã dòng.

Slide: 20

Ví dụ: mã hóa bản rõ P=111 (chữ h) với khóa K là 100101. 01001110.100110000

Ban đầu giá trị của các thanh ghi X, Y, Z là:

X = 100101

Y = 01001110

Z = 100110000

Bước 0: $x_1 = 0, y_3 = 0, z_3 = 1 \rightarrow$ m = 0 $\rightarrow$ quay X, quay Y

X = 110010

Y = 10100111 $\rightarrow s_0 = 0 \oplus 1 \oplus 0 = 1$

Z = 100110000

Bước 1: $x_1 = 1, y_3 = 0, z_3 = 1 \rightarrow$ m = 1 $\rightarrow$ quay X, quay Z

X = 111001

Y = 10100111 $\rightarrow s_1 = 1 \oplus 1 \oplus 0 = 0$

Z = 010011000

Bước 2: $x_1 = 1, y_3 = 0, z_3 = 0 \rightarrow$ m = 0 $\rightarrow$ quay Y, quay Z

X = 111001

Y = 01010011 $\rightarrow s_2 = 1 \oplus 1 \oplus 0 = 0$

Z = 001001100

Vậy bản mã là C = 111 $\oplus$ 100 = 011 (chữ D)

Hệ mã dòng A5/1

- A5/1 hoạt động giống như TinyA5/1.
- Kích thước thanh ghi X, Y, Z lần lượt là 19, 22 và 23 bít

1) **Quay X**:

- $t = x_{13} \oplus x_{16} \oplus x_{17} \oplus x_{18}$
- $x_j = x_{j-1}$ với $j = 18, 17, 16 \ldots, 1$
- $x_0 = t$

2) **Quay Y**:

- $t = y_{20} \oplus y_{21}$
- $y_j = y_{j-1}$ với $j = 21, 20, 19, \ldots, 1$
- $y_0 = t$

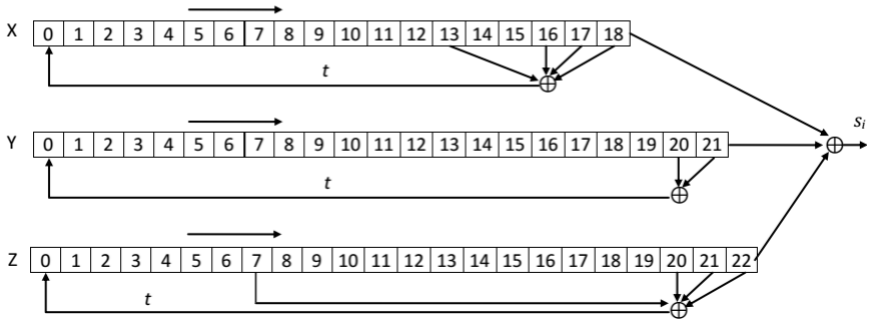
3) **Quay Z**:

- $t = z_7 \oplus z_{20} \oplus z_{21} \oplus z_{22}$
- $z_j = z_{j-1}$ với $j = 22, 21, 20, \ldots, 1$
- $z_0 = t$

Slide: 22

Hệ mã dòng A5/1

- Hàm maj được tính trên 3 bít x_8, y_{10}, z_{10}. Sau khi quay xong bít sinh ra là: $s_i = x_{18} \oplus y_{21} \oplus z_{22}$.
- Toàn bộ quá trình sinh dãy số của A5/1 được minh họa qua hình bên dưới:



Slide: 23

Mã khối (Block Cipher)

- Trong máy tính các chữ cái được biểu diễn bằng mã ASCII.

Bản tin: `attack`

Mã ASCII: `97 116 116 97 99 107`

Biểu diễn nhị phân: `01100001 01110100 01110100 01100001 01100011 01101011`

- Trong bản tin nhị phân cũng tồn tại một số đặc tính thống kê nào đó mà người phá mã có thể tận dụng để phá bản mã, dù rằng bản mã bây giờ tồn tại dưới dạng nhị phân.
- Mã hóa hiện đại quan tâm đến vấn đề *chống phá mã trong các trường hợp biết trước bản rõ (known-plaintext), hay bản rõ được lựa chọn (chosen-plaintext)*.

Slide: 24

Mã khối (Block Cipher)

Chữ cái	Nhị phân
A	000
B	001
C	010
D	011
E	100
F	101
G	110
H	111

- Ví dụ: chúng ta sử dụng bản rõ là các chữ cái của một *ngôn ngữ* gồm có 8 chữ cái A, B, C, D, E, F, G, H trong đó mỗi chữ cái được biểu diễn bằng 3 bít.
- Như vậy nếu có bản rõ là 'head' thì biểu diễn nhị phân tương ứng là: 111100000011
- Giả sử dùng một khóa K gồm 4 bít 0101 để mã hóa bản rõ trên bằng phép XOR ⊕:

bản rõ:	1111 0000 0011	(head)
khóa:	0101 0101 0101	
bản mã:	1010 0101 0110	(FBCG)

Trong phép mã hóa trên, đơn vị mã hóa không phải là một chữ cái mà là một khối 4 bít. Để giải mã, lấy bản mã XOR một lần nữa với khóa thì có lại bản rõ ban đầu.

Slide: 25

Mã khối (Block Cipher)

Mã khối an toàn lý tưởng

- Phép toán XOR có một hạn chế là chỉ cần biết *một cặp khối* bản rõ và bản mã, người ta có thể dễ dàng suy ra được khóa và dùng khóa đó để giải các khối bản mã khác (knownplaintext attack). :

bản rõ:	1111 0000 0011	(head)
khóa:	0101 0101 0101	
bản mã:	1010 0101 0110	(FBCG)

Nếu biết bản mã c_0 = 1010 có bản rõ tương ứng là p_0 = 1111, thì có thể dễ dàng suy ra khóa là 0101.

Slide: 26

Mã khối (Block Cipher)

- Do đó để chống phá mã trong trường hợp known-plaintext hay choosen-plaintext, chỉ có thể là làm cho *P* và *C* không có mối liên hệ toán học. Điều này chỉ có thể thực hiện được nếu ta lập một bản tra cứu ngẫu nhiên giữa bản rõ và bản mã.

→đây là *mã khối an toàn lý tưởng vì* Người gởi cũng như người nhận phải biết toàn bộ bảng trên để mã hóa và giải mã

→Không khả thi vì kích thước khối lớn thì số dòng của bảng khóa cũng lớn và gây trở ngại cho việc lưu trữ cũng như trao đổi khóa giữa người gởi và người nhận

Bản rõ	Bản mã
0000	1110
0001	0100
0010	1101
0011	0001
0100	0010
0101	1111
0110	1011
0111	1000
1000	0011
1001	1010
1010	0110
1011	1100
1100	0101
1101	1001
1110	0000
1111	0111

Slide: 27

Các mô hình ứng dụng mã khối

- Mã khối (như mã DES) được áp dụng để mã hóa một khối dữ liệu có kích thước xác định. Để mã hóa một bản tin dài, bản tin được chia ra thành nhiều khối ($P=P_0P_1P_2...P_{n-1}$) và áp dụng mã khối cho từng khối một. Có nhiều mô hình áp dụng mã khối là
  - Electronic Code Book (ECB)
  - Cipher Block Chaining Mode (CBC)
  - Cipher Feedback Mode (CTR)
  - Output Feedback Mode (OFB)
  - Counter Mode

Slide: 28

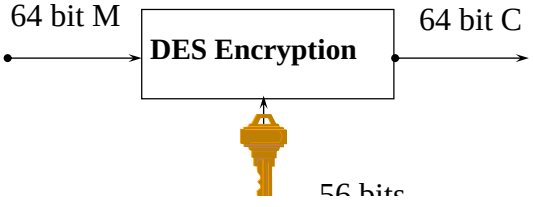
Mã DES
(Data Encryption Standard)

- DES được công nhận vào năm 1977 bởi Viện nghiên cứu quốc gia về chuẩn của Mỹ (NIST –National Institut of Standards and Technology), chuẩn hóa 1979.
- Là mã thuộc hệ mã Feistel gồm 16 vòng, ngoài ra DES có thêm một hoán vị khởi tạo trước khi vào vòng 1 và một hoán vị khởi tạo sau vòng 16
- Kích thước của khối là 64 bít.
- Ví dụ bản tin "*meetmeafterthetogaparty*" biểu diễn theo mã ASCII thì mã DES sẽ mã hóa làm 3 lần, mỗi lần 8 chữ cái (64 bít): *meetmeaf - tertheto - gaparty.*

Slide: 29

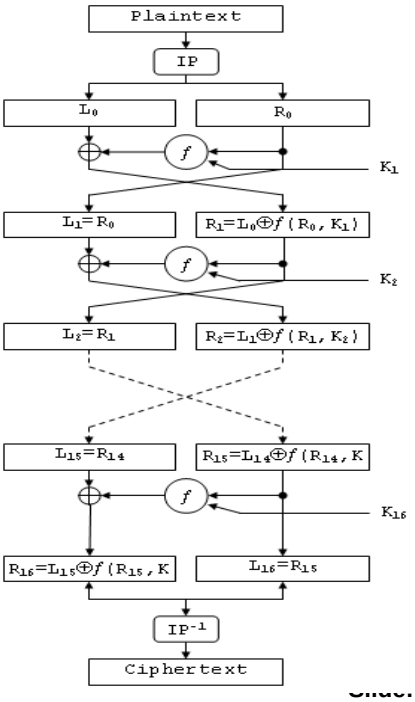
Đặc điểm của thuật toán DES

- Khóa dùng trong DES có độ dài toàn bộ là 64 bit. Tuy nhiên chỉ có 56 bit thực sự được sử dụng; 8 bit còn lại chỉ dùng cho việc kiểm tra.
- Mỗi vòng của DES dùng khóa con có kích thước 48 bít được trích ra từ khóa chính.
- Des xuất ra bãn mã 64 bit.
- Mã hoá và giải mã được sử dụng cùng một khoá.
- DES được thiết kế để chạy trên phần cứng.

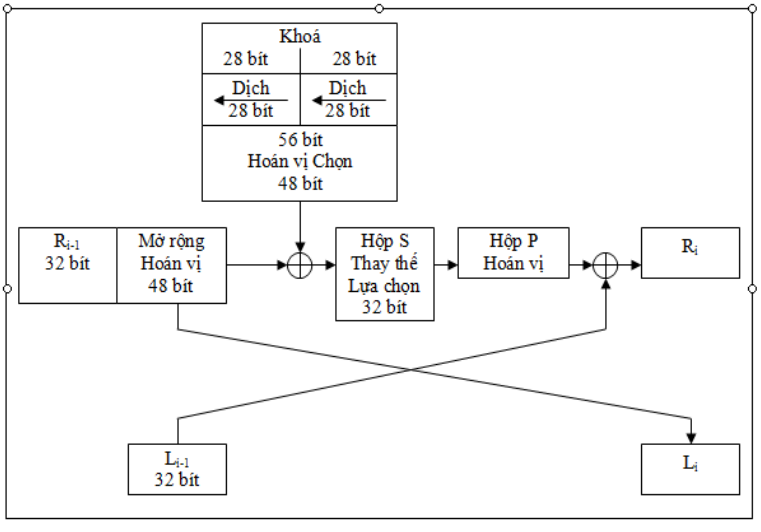


Slide: 30

Mô tả thuật toán



Mô tả thuật toán



Một vòng lặp DES

Mô tả thuật toán

Thuật toán được thực hiện trong 3 giai đoạn:

1. Cho bản rõ x (64bit) được hoán vị khởi tạo IP (Initial Permutation) tạo nên xâu bit $x_{0.}$

$$x_0=IP(x)=L_0R_0$$

L_0 là 32 bit đầu tiên của x_0.
R_0 là 32 bit cuối của x_0.

Slide: 33

Mô tả thuật toán

Bộ chuyển vị IP

Hoán vị khởi đầu nhằm đổi chỗ khối dữ liệu vào , thay đổi vị trí của các bít trong khối dữ liệu vào. Ví dụ, hoán vị khởi đầu chuyển bít 1 thành bít 58, bít 2 thành bít 50, bít 3 thành bít 42,...

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Slide: 34

Mô tả thuật toán

2. Từ L_0 và R_0 sẽ lặp 16 vòng, tại mỗi vòng tính:

   $L_i=R_{i-1}$

   $R_i=L_{i-1}\oplus f(R_{i-1},K_i)$ với i= 1, 2,…,16

   với:

   $\oplus$ là phép XOR của hai xâu bit:

   $0 \oplus 0=0$, $1 \oplus 1=0$

   $1 \oplus 0=1$, $0 \oplus 1=1$

   f là hàm mà ta sẽ mô tả sau.

   K_i là các xâu có độ dài 48 bit được tính như là các hàm của khóa K.

Slide: 35

Mô tả thuật toán

3. Tại vòng thứ 16, R16 đổi chỗ cho L16. Sau đó ghép 2 nửa R16 L16 cho đi qua hoán vị nghịch đảo của hoàn vị IP sẽ tính được bản mã. Bản mã cũng có độ dài 64 bít.

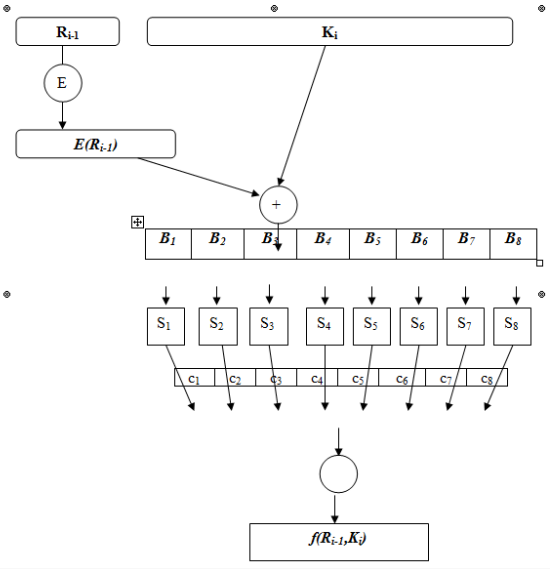
Hoán vị IP^{-1}

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Slide: 36

Mô tả thuật toán

- Hàm F



Slide: 37

Mô tả thuật toán

Hàm F

Hàm f lấy đối số đầu vào là xâu nhập R_{i-1} (32 bit) đối số thứ hai là K_i (48 bit) và tạo ra xâu xuất có độ dài 32 bit. Các bước sau được thực hiện.

1. Đối số đầu R_{i-1} sẽ được "mở rộng" thành xâu có độ dài 48 bit tương ứng với hàm mở rộng E cố định. E(R_i) bao gồm 32 bit từ R_i, được hoán vị theo một cách thức xác định, với 16 bit được tạo ra 2 lần.

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Hàm mở rộng E

Slide: 38

Mô tả thuật toán

2. Tính $E(R_{i-1}) \oplus K_i$ kết quả được một khối có độ dài 48 bit. Khối này sẽ được chia làm 8 khối $B=B_1B_2B_3B_4B_5B_6B_7B_8$. Mỗi khối này có độ dài là 6 bít.

3. Bước kế tiếp là cho các khối Bi đi qua hộp S_i sẽ biến một khối có độ dài 6 bit thành một khối C_i có độ dài 4 bít.

Slide: 39

Mô tả thuật toán

Hộp S

- Mỗi hộp S-box là một bảng gồm 4 hàng và 16 cột được đánh số từ 0. Như vậy mỗi hộp S có hàng 0,1,2,3. Cột 0,1,2,…,15. Mỗi phần tử của hộp là một số 4 bít. Sáu bít vào hộp S sẽ xác định số hàng và số cột để tìm kết quả ra.
- Mỗi khối Bi có 6 bít kí hiệu là b1, b2, b3, b4, b5 và b6. Bít b1 và b6 được kết hợp thành một số 2 bít, nhận giá trị từ 0 đến 3, tương ứng với một hàng trong bảng S. Bốn bít ở giữa, từ b2 tới b5, được kết hợp thành một số 4 bít, nhận giá trị từ 0 đến 15, tương ứng với một cột trong bảng S.

Slide: 40

Mô tả thuật toán

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Hộp S_1

15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

Hộp S_2

Slide: 41

Mô tả thuật toán

Hộp S3

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

Hộp S4

7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

Slide: 42

Mô tả thuật toán

Hộp S5

2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

Hộp S6

12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

Slide: 43

Mô tả thuật toán

Hộp S7

4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

Hộp S8

13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Slide: 44

Mô tả thuật toán

- **Ví dụ:** Ta có B1=011000 thì b_1b_6=00 (xác định r=0), $b_2b_3b_4b_5$=1100 (xác định c=12), từ đó ta tìm được phần tử ở vị trí (0,12) --> S1(B1)=0101 (tương ứng với số 5).

$b_2b_3b_4b_5$=1100

b_1b_6=00

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Hộp S1

- Mỗi xâu xuất 4 bit của các hộp S được đưa vào các Cj tương ứng: Cj = Sj(Bj) (1<=j<=8).

Slide: 45

Mô tả thuật toán

- 4. Xâu bit C = $C_1C_2C_3C_4C_5C_6C_7C_8$ có độ dài 32 bit được hoán vị tương ứng với hoán vị cố định P. Kết quả có P(C)= $f(R_i,K_i)$.

Hoán vị P

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Slide: 46

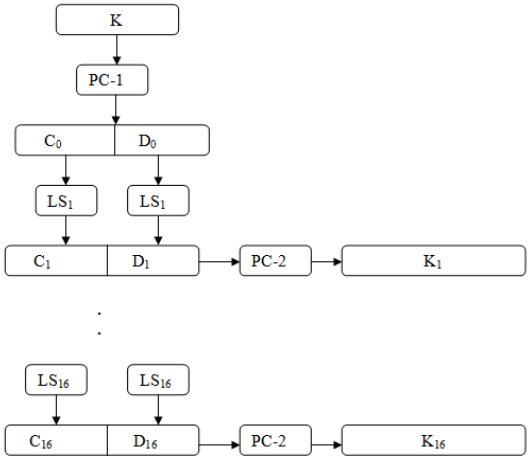
Mô tả thuật toán

Khóa K

- K là một xâu có độ dài 64 bit trong đó 56 bit dùng làm khóa và 8 bit dùng để kiểm tra sự bằng nhau (phát hiện lỗi).
- Các bit ở các vị trí 8, 16,…, 64 được xác định, sao cho mỗi byte chứa số lẻ các số 1, vì vậy từng lỗi có thể được phát hiện trong mỗi 8 bit.

Slide: 47

Mô tả thuật toán



Sơ đồ tính khóa $k_1, k_2, \ldots k_{16}$

Slide: 48

Mô tả thuật toán

Quá trình tạo các khóa con (subkeys) từ khóa K được mô tả như sau:

Cho khóa K 64 bit, loại bỏ các bit kiểm tra và hoán vị các bit còn lại của K tương ứng với hoán vị cố định PC-1. Ta viết PC1(K) = C_0D_0, với C_0 bao gồm 28 bít đầu tiên của PC-1(k) và D_0 là 28 bit còn lại.

Trong đó bảng số bít dịch trái tại mỗi vòng là:

Vòng i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Số bít dịch	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Slide: 49

Giải mã

- Việc giải mã dùng cùng một thuật toán như việc mã hoá.
- Để giải mã dữ liệu đã được mã hoá, quá trình giống như mã hoá được lặp lại nhưng các chìa khoá phụ được dùng theo thứ tự ngược lại từ K_{16} đến K_1, nghĩa là trong bước 2 của quá trình *mã hoá dữ liệu đầu vào* ở trên R_{i-1} sẽ được XOR với K_{17-i} chứ không phải với K_i.

Slide: 50

Ví dụ mã hóa

Mã hóa bản rõ sau trong dạng thập lục phân (Hexadecimal) ***0123456789ABCDEF***

Sử dụng khóa thập lục phân ***133457799BBCDFF1***

Slide: 51

L_0	=	11001100000000000110011001111111
$L_1 = R_0$	=	11110000101010101111000010101010

$E(R_0)$	=	01111010000101010101010101111010000101010101010101
K_1	=	00011011000000101110111111111110001110000011110010
$E(R_0) \oplus K_1$	=	01100001000101111011101010000110011001010010011 1
Output S-hộp	=	01011100100000101011010110010111
$f(R_0, K_1)$	=	00100011010010101010100110111011
$L_2 = R_1$	=	11101111010010100110010101000100

$E(R_1)$	=	01110101111010100101010000110000101010100000 1001

K_2	=	01111001101011101101100111011011110010011110 0101
$E(R_1) \oplus K_2$	=	00001100010001001000110111101011011000111110 1100
Output S-hộp	=	11111000110100000011101010101110
$f(R_1, K_2)$	=	00111100101010111000011110100011
$L_3 = R_2$	=	11001100000000010111011100001001

Slide: 52

$E(R_2)$	=	111001011000000000000010101110101110100001010011
K_3	=	010101011111110010001010010000101100111110011001
$E(R_2) \oplus K_3$	=	101100000111110010001000111110000010011111001010
S-box output	=	00100111000100001110000101101111
$f(R_2, K_3)$	=	01001101000101100110111010110000
$L_4 = R_3$	=	10100010010111000000101111110100

$E(R_3)$	=	01010000010000101111100000000101010111111110101001
K_4	=	01110010101011011101011011011011001101010000011101
$E(R_3) \oplus K_4$	=	00100010111011110010111011011110010010101010110100
S-box output	=	00100001111011011001111100111010
$f(R_3, K_4)$	=	10111011001000110111011101001100
$L_5 = R_4$	=	011101110

$E(R_4)$	=	10111010111010010000010000000000000000001000001010
K_5	=	01111100111011000000011111101011010100111010100 0
$E(R_4) \oplus K_5$	=	11000110000001010000001111101011010100011010 0010
Xuất S-hộp	=	01010000110010000011000111101011
$f(R_4, K_5)$	=	00101000000100111010110111000011
$L_6 = R_5$	=	10001010010011111010011000110111

Slide: 53

$E(R_5)$	=	11000101010000100101111111101000011000001101011111
K_6	=	01100011101001010011111100101000001111011001011111
$E(R_5) \oplus K_6$	=	10100110111001110110000110000000101110101000000 0
S-box output	=	01000001111100110100110000111101
$f(R_5, K_6)$	=	10011110010001011100110100101100
$L_7 = R_6$	=	11101001011001111100110101101001

$E(R_6)$	=	11110101001010110000111111100101101010110101010011
K_7	=	11101100100001001011011111111011000011000101111100
$E(R_6) \oplus K_7$	=	00011001101011111011100000010011101100111111011111
S-box output	=	00010000011101010100000010101101
$f(R_6, K_7)$	=	10001100000001010001110000100111
$L_8 = R_7$	=	00000110010010101011101000010000

$E(R_7)$	=	00000000110000100101010101011111010000001010000 0

K_8	=	11110111100010100011101011000001001110111111111011
$E(R_7) \oplus K_8$	=	11110111010010000110111110011110011110110101011011
S-box output	=	01101100000110000111110010101110
$f(R_7, K_8)$	=	00111100000011101000011011111001
$L_9 = R_8$	=	11010101011010010100101110010000

Slide: 54

$E(R_8)$	=	011010101010101101010010101001010111110010100001
K_9	=	111000001101101111101011111011011110011110000001
$E(R_8) \oplus K_9$	=	100010100111000010111001010010001001101100100000
S-box output	=	00010001000011000101011101110111
$f(R_8, K_9)$	=	00100010001101100111110001101010
$L_{10} = R_9$	=	00100100011111001100011001111010

$E(R_9)$	=	000100001000001111111001011000001100000111111010 0
K_{10}	=	101100011111001101000111101110100100011001001111
$E(R_9) \oplus K_{10}$	=	101000010111000010111110110110101000010110111011
S-box output	=	11011010000001000101001001110101
$f(R_9, K_{10})$	=	01100010101111001001110000100010
$L_{11} = R_{10}$	=	10110111110101011101011110110010
$E(R_{10})$	=	01011010111111101010101111101010111111011010 0101
K_{11}	=	00100001010111111101001111011110110100111000011 0
$E(R_{10}) \oplus K_{11}$	=	01111011101000010111100000110100001011100010 0011
S-box output	=	01110011000001011101000100000001
$f(R_{10}, K_{11})$	=	11100001000001001111101000000010
$L_{12} = R_{11}$	=	11000101011110000011110001111000

Slide: 55

$E(R_{11})$		011000001010101111110000000111111000001111110001
K_{12}		011101010111000111111010110010100011001111110100 1
$E(R_{11}) \oplus K_{12}$		000101011101101000000101100010111110010000011000
S-box output		01111011100010110010011000110101
$f(R_{11}, K_{12})$		11000010011010001100111111101010
$L_{13} = R_{12}$		01110101101111010001100001011000

$E(R_{12})$	=	00111010101111011111101010001111000000101111000 0
K_{13}	=	10010111110001011101000111111010101111010010000 01
$E(R_{12}) \oplus K_{13}$	=	10101101011110000010101101110101101110001011000 1
S-box output	=	10011010110100011000101101001111
$f(R_{12}, K_{13})$	=	11011101101110110010100100100010
$L_{14} = R_{13}$	=	00011000110000110001010101011010

$E(R_{13})$	=	000011110001011000000110100010101010101011110100
K_{14}	=	010111110100001110110111111100101110011100111010
$E(R_{13}) \oplus K_{14}$	=	010100000101010110110001011110000100110111001110

Slide: 56

S-box output	=	01100100011110011001101011110001
$f(R_{13}, K_{14})$	=	10110111001100011000111001010101
$L_{15} = R_{14}$	=	11000010100011001001011000001101

$E(R_{14})$	=	11100000010101000101100101001010110000000010110 11
K_{15}	=	10111111100100011000110100111101001111110000 1010
$E(R_{14}) \oplus K_{15}$	=	01011111110001011101010001110111111111110101 0001
S-box output	=	10110010111010001000110100111100
$f(R_{14}, K_{15})$	=	01011011100000010010011101101110
$L_{16} = R_{15}$	=	01000011010000100011001000110100

$E(R_{15})$	=	00100000011010100000010000011010010000011010 1000
K_{16}	=	11001011001111011000101100001110000101111111 0101
$E(R_{15}) \oplus K_{16}$	=	11101011010101111000111100010100010101100101 1101
S-box output	=	10100111100000110010010000101001
$f(R_{15}, K_{16})$	=	11001000110000000100111110011000
R_{16}	=	00001010010011001101100110010101

Slide: 57

Ví dụ mã hóa

Kết luận:

Cuối cùng áp dụng IP-1 cho R16L16 ta nhận được bản rõ trong dạng thập lục phân sau: ***85E813540F0AB405***

Slide: 58

Đặc điểm của hệ mã DES

Không gian khóa

-DES có $2^{56} = 10^{17}$ khoá

-Nếu biết được một cặp "tin/mã" có thể thử tất cả 10^{17} khả năng này để tìm ra khoá cho kết quả khớp nhất.

-Nếu một phép thử 10^{-6}s thì sẽ mất 10^{11}s tức là 7300 năm

-Vào năm 1976 và 1977, Diffie và Hellman đã ước lượng rằng có thể chế tạo được một máy tính chuyên dụng để vét cạn không gian khoá DES trong ½ ngày với cái giá 20 triệu đô la

-Đến năm 1990, hai nhà toán học người Do Thái - Biham và Shamir - đã phát minh ra phương pháp phá mã vi sai, đây là một kỹ thuật sử dụng những phỏng đoán khác nhau trong bản rõ để đưa ra những thông tin trong bản mã

Slide: 59

Đặc điểm của hệ mã DES

Tính bù

Nếu ta ký hiệu $\overline{U}$ Là phần tử bù của U (ví dụ 0100101 là phần bù của 1011010) thì DES có tính chất sau:

$$y = DES(x,k) \rightarrow \overline{y} = DES(\overline{x}, \overline{k})$$

=> Nếu biết bản mã y, bản rỏ x và khóa k thì biết

Do tính bù, ta có thể giảm độ phức tạp của tấn công duyệt toàn bộ xuống 2 lần (tương ứng với 1 bít) với điều kiện là ta có thể lựa chọn bản rõ.

Slide: 60

Đặc điểm của hệ mã DES

Khóa yếu:

Khoá yếu là các khoá mà theo thuật toán sinh khoá con thì tất cả 16 khoá con đều như nhau:

$$K_1 = K_2 = ... = K_{15} = K_{16}$$

=>Việc mã hóa và giải mã đối với khoá yếu là giống hệt nhau

Khoá yếu (Hex)				C_0	D_0
0101	0101	0101	0101	$\{0\}^{28}$	$\{0\}^{28}$
FEFE	FEFE	FEFE	FEFE	$\{1\}^{28}$	$\{1\}^{28}$
1F1F	1F1F	0E0E	0E0E	$\{0\}^{28}$	$\{1\}^{28}$
E0E0	E0E0	F1F1	F1F1	$\{1\}^{28}$	$\{0\}^{28}$

Slide: 61

Đặc điểm của hệ mã DES

Đồng thời còn có 6 cặp khoá nửa yếu (semi-weak key) khác với thuộc tính như sau:

$$y = DES(x,k_1) \text{ và } y = DES(x,k_2)$$

Nghĩa là với 2 khoá khác nhau nhưng mã hoá ra cùng một bản mã từ cùng một bản rõ:

C_0	D_0	Semi-weak key (Hex)								C_0	D_0
$\{01\}^{14}$	$\{01\}^{14}$	01FE	01FE	01FE	01FE	FE01	FE01	FE01	FE01	$\{10\}^{14}$	$\{10\}^{14}$
$\{01\}^{14}$	$\{10\}^{14}$	1FE0	1FE0	0EF1	0EF1	E01F	E01F	F10E	F10E	$\{10\}^{14}$	$\{01\}^{14}$
$\{01\}^{14}$	$\{0\}^{28}$	01E0	01E0	01F1	01F1	E001	E001	F101	F101	$\{10\}^{14}$	$\{0\}^{28}$
$\{01\}^{14}$	$\{1\}^{28}$	1FFE	1FFE	0EFE	0EFE	FE1F	FE1F	FE0E	FE0E	$\{10\}^{14}$	$\{1\}^{28}$
$\{0\}^{28}$	$\{01\}^{14}$	011F	011F	010E	010E	1F01	1F01	0E01	0E01	$\{0\}^{28}$	$\{10\}^{14}$
$\{1\}^{28}$	$\{01\}^{14}$	E0FE	E0FE	F1FE	F1FE	FEE0	FEE0	FEF1	FEF1	$\{1\}^{28}$	$\{10\}^{14}$

Slide: 62

Triple DES (3DES)

-Hệ mã DES (hay chuẩn mã hóa dữ liệu) với không gian khóa có 2^{55} khóa nên thực tế hiện nay có thể bị thám mã trong một khoảng thời gian ngắn.

-Tìm kiếm một hệ mã mới là điều tất yếu.

-Tận dụng hệ mã DES, chúng ta đưa ra một phương pháp mã hóa mới bằng cách mã hóa nhiêu lần.

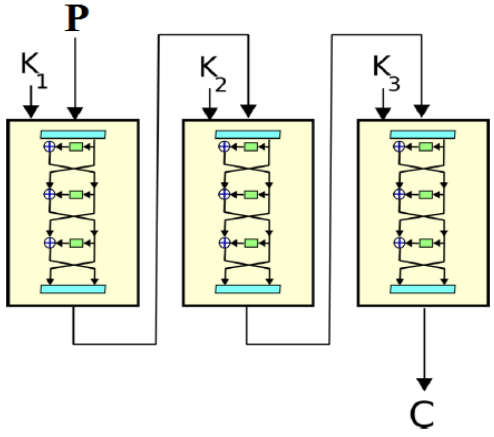
-Chúng ta sử dụng 2 khóa để mã hóa hai lần $C = E_{K2}(E_{K1}(P))$

-Cách này gọi là double DES hay 2DES, khóa của hệ mã theo mô hình này là 112 bit tuy nhiên về mặt lý thuyết thì hệ mã này không an toàn hơn DES.

Slide: 63

Triple DES (3DES)

-Chúng ta có thể sử dụng thuật toán DES với 3 khóa để mã hóa được gọi là Triple DES (TDES) hay 3DES



Slide: 64

Triple DES (3DES)

-Bản mã C = $DES_{K3}(DES_{K2}(DES_{K1}(M))$, mô hình này gọi là EEE,

-Một biến thể khác của mô hình này gọi là EDE với bước ở giữa sử dụng thuật toán giải mã của DES:

$$C = DES_{K3}(DES^{-1}_{K2}(DES_{K1}(M)).$$

-Khóa của Triple DES là 168 bit. Các chứng minh về mặt lý thuyết và các tấn công đối với Triple DES cho thấy hệ mã này vẫn sẽ còn được sử dụng trong một tương lai dài

Slide: 65

Mã AES (Advanced Encryption Standard)

- Tiêu chuẩn mã hóa tiên tiến (Advanced Encryption Standard - AES) được giới thiệu vào năm 2001 bởi NIST với mục đích thay thế DES có nhiều hạn chế, và được sử dụng rộng rãi trong các ứng dụng hiện nay, thuật toán được thiết kế bởi Joan Daemen và Vincent Rijmen với tên gọi ban đầu là Rijndael.
- Khóa có kích thước 256 bít là ***"an toàn mãi mãi"*** bất kể những tiến bộ trong ngành kỹ thuật máy tính.
- AES là thuật toán mã hóa đối xứng dạng khối 128-bit

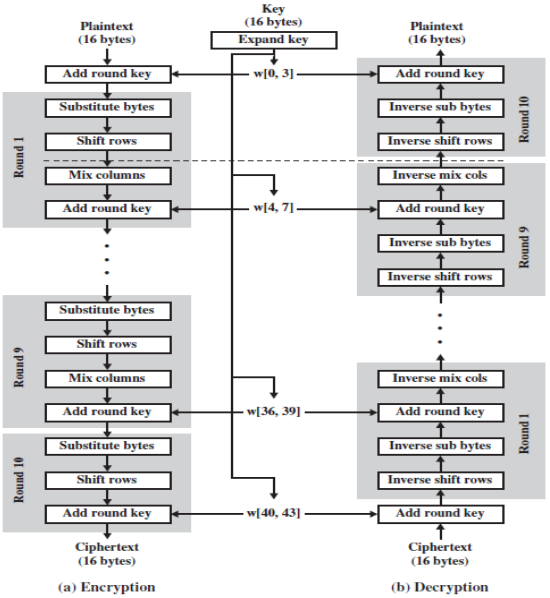
Slide: 66

Mã AES
(Advanced Encryption Standard)

- Như DES, mã AES là mã khối, nhiều vòng, nhưng mã AES không phải là một mã Feistel. Thuật toán AES khá phức tạp.
- Đặc điểm chính của AES:
  - Cho phép lựa chọn kích thước khối mã hóa là 128, 192 hay 256 bít.
  - Cho phép lựa chọn kích thước của khóa một cách độc lập với kích thước khối: là 128, 192 hay 256 bít.
  - Số lượng vòng có thể thay đổi từ 10 đến 14 vòng tùy thuộc vào kích thước khóa.

Slide: 67

Mã AES
(Advanced Encryption Standard)

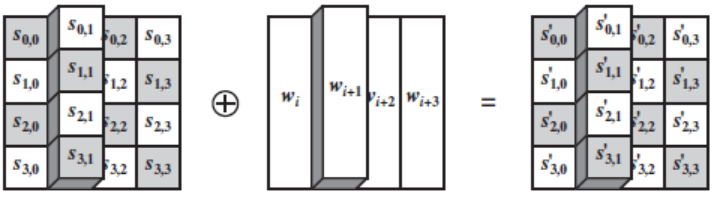


Slide: 68

Mã AES (Advanced Encryption Standard)

- Quá trình mã hóa bao gồm 4 bước:

B1: AddRoundKey - mỗi byte của khối được kết hợp với khóa con, các khóa con này được tạo ra từ quá trình tạo khóa con Rijndael.

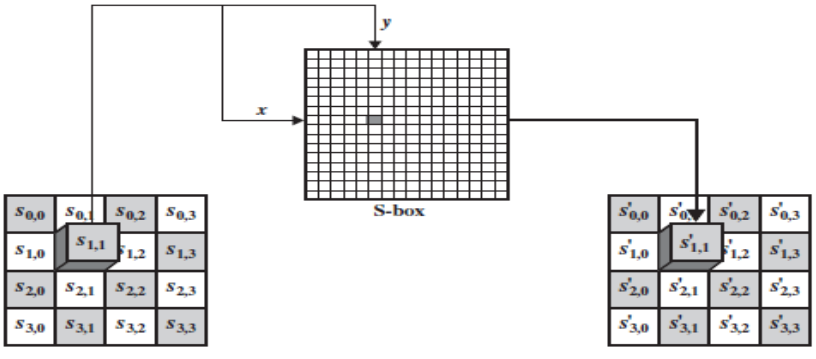


Add round key transformation

Slide: 69

Mã AES (Advanced Encryption Standard)

- B2: SubBytes - đây là quá trình thay thế trong đó mỗi byte sẽ được thay thế bằng một byte khác theo bảng tra.

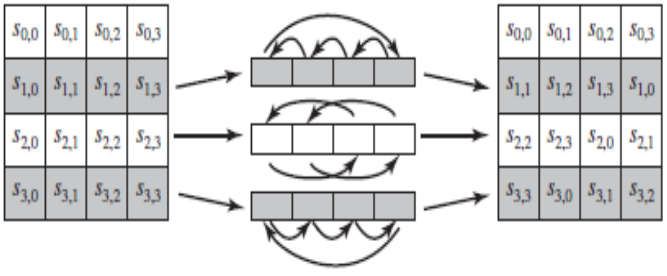


Substitute byte transformation

Slide: 70

Mã AES
(Advanced Encryption Standard)

- B3: ShiftRows - đổi chỗ, các hàng trong khối được dịch vòng.

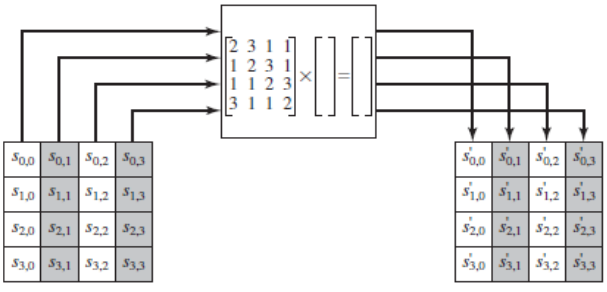


Shift row transformation

Slide: 71

Mã AES
(Advanced Encryption Standard)

- B4: MixColumns - quá trình trộn làm việc theo các cột trong khối theo một chuyển đổi tuyến tính.
- Tại chu trình cuối thì MixColumns được thay thế bằng AddRoundKey



Mix column transformation

Slide: 72

Mã AES (Advanced Encryption Standard)

- Độ an toàn của AES làm cho AES được sử dụng ngày càng nhiều và trong tương lai sẽ chiếm vai trò của DES và Triple DES.

Slide: 73

Ưu và nhược điểm của mã hóa đối xứng

Ưu điểm: độ an toàn cao (phụ thuộc vào thuật toán và khóa), quá trình mã hóa và giải mã nhanh do đó mã hóa đối xứng được sử dụng phổ biến trong việc truyền dữ liệu.

Slide: 74

Ưu và nhược điểm của mã hóa đối xứng

Hạn chế: Một số vấn đề cần quan tâm của hệ thống mã hóa đối xứng liên quan đến khóa, bao gồm:

- Do quá trình mã hóa và giải mã sử dụng chung một khóa nên khóa (secret key) sử dụng cần phải được bảo quản an toàn tuyệt đối.
- Vấn đề phân phối khóa
- Vấn đề quản lý khóa (với hệ thống có n nút khác nhau thì số lượng khóa cần thiết cho hệ thống là n(n+1)/2
- Không cung cấp tính chống thoái thác thông tin

Xin chân thành cảm ơn!