AN TOÀN VÀ BẢO MẬT THÔNG TIN

GVTH: TS. Nguyễn Đào Trường

Nội dung

- Chương 1: Tổng quan về an toàn và bảo mật thông tin.
- Chương 2: Các phương pháp mã hóa cổ điển
- Chương 3: Chuẩn mã dữ liệu DES - AES
- Chương 4: Mật mã công khai
- Chương 5: Các sơ đồ chữ ký số
- Chương 6: Hàm băm
- Chương 7: Giao thức mật mã

Chương 3: Chuẩn mã dữ liệu DES (Data Encryption Standard)

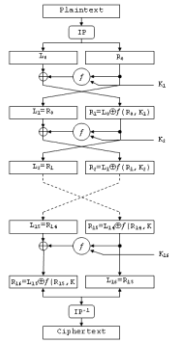
1.Giới thiệu chung về DES

- Ngày 13/5/1973 ủy ban quốc gia về tiêu chuẩn của Mỹ công bố yêu cầu về hệ mật mã áp dụng cho toàn quốc. Điều này đã đặt nền móng cho chuẩn mã hóa dữ liệu, hay là DES.
- Lúc đầu Des được công ty IBM phát triển từ hệ mã Lucifer, công bố vào năm 1975.
- Sau đó Des được xem như là chuẩn mã hóa dữ liệu cho các ứng dụng.

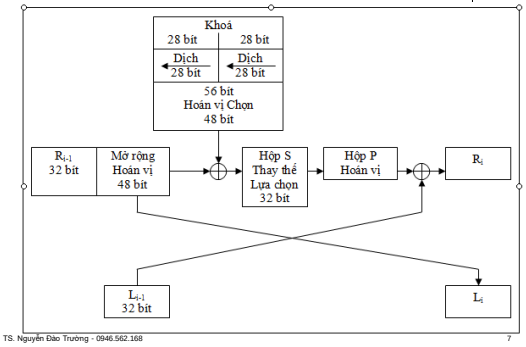
2. Đặc điểm của thuật toán DES

- DES là thuật toán mã hóa khối, độ dài mỗi khối là 64 bit .
- 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.
- Des xuất ra bãn mã 64 bit.
- Thuật toán thực hiện 16 vòng
- Mã hoá và giải mã được sử dụng cùng một khoá.
- DES được thiết kế để chạy trên phần cứng.

3. Mô tả thuật toán



3. Mô tả thuật toán



3. 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.

3. 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

3. 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:
   ⊕ 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.
   K_1 đến K_{16} lập nên một lịch khóa.

3. Mô tả thuật toán

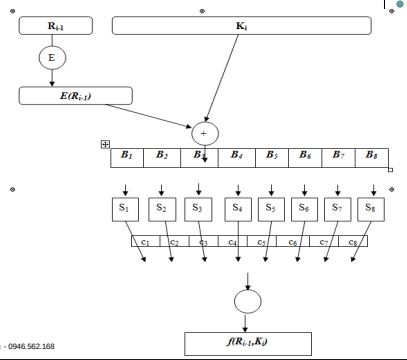
3. Tại vòng thứ 16, R16 đổi chỗ cho L16. Sau đó ghép 2 nửa R16, L16 cho đi qua hoàn vị nghịch đảo của hoàn vị IP sẽ tính được bản mã. Bản mã cũng có độ dài 64 bít.

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

3. Mô tả thuật toán

Hàm f



Hàm f

Hàm f lấy đối số đầu 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.

Hàm f

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

Hàm f

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 bit.

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 bit.

S-box

- 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.

S-box

Hộp S1

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 S2

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

S-box

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

S-box

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

S-box

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

S-box

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).

Hàm f

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)= $f(R_i,K_i)$.

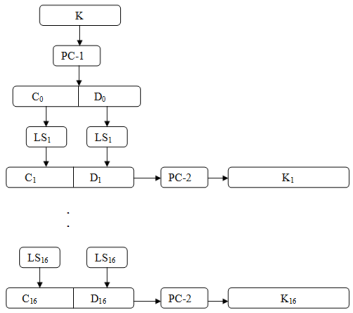
Hoán vị P

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

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.
- Các bit kiểm tra sự bằng nhau là được bỏ qua khi tính lịch khóa.

Sơ đồ tính khóa K1, K2, …, K16



Khóa K

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

Khóa K

Các hoán vị cố định PC-1 và PC-2:

Bảng trật tự khoá (PC-1):

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

Bảng trật tự nén(PC-2):

14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32

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.

Đặc điểm của mã DES

Tính chất bù của mã DES:

DES có tính chất bù:

$$E_{\bar{K}}(\bar{P}) = \bar{C} \Leftrightarrow E_K(P) = C$$

trong đó :

$\bar{A}$ là phần bù của A theo từng bít (1 thay bằng 0 và ngược lại).

E_K là bản mã hóa của E với khóa K. P và C là văn bản rõ (trước khi mã hóa) và văn bản mã (sau khi mã hóa).

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.

Đặc điểm của mã DES

Các khóa yếu trong mã Des:

Ngoài ra DES còn có 4 khóa yếu (weak keys). Khi sử dụng khóa yếu thì mã hóa (E) và giải mã (D) sẽ cho ra cùng kết quả:

$E_K(E_K(P)) = P$ or equivalently, $E_K = D_K$

Bên cạnh đó, còn có 6 cặp *khóa nửa yếu* (semi-weak keys). Mã hóa với một khóa trong cặp, *K*1, tương đương với giải mã với khóa còn lại, *K*2:

$E_{K1}(E_{K2}(P))=P$ or equivalently $E_{K1}=D_{K2}$

Tuy nhiên có thể dễ dàng tránh được những khóa này khi thực hiện thuật toán, có thể bằng cách thử hoặc chọn khóa một cách ngẫu nhiên. Khi đó khả năng chọn phải khóa yếu là rất nhỏ.

Đặc điểm của mã DES

Triple DES:

Triple-DES chính là DES với hai chìa khoá 56 bit. Cho một bản tin cần mã hoá, chìa khoá đầu tiên được dùng để mã hoá DES bản tin đó.

Kết quả thu được lại được cho qua quá trình giải mã DES nhưng với chìa khoá là chìa khoá thứ hai.

Bản tin sau qua đã được biến đổi bằng thuật toán DES hai lần như vậy lại được mã hoá DES một lần nữa với chìa khoá đầu tiên để ra được bản tin mã hoá cuối cùng.

Quá trình mã hoá DES ba bước này được gọi là Triple-DES.

Ví dụ mã hóa giải mã DES

Một bản rõ mang nội dung: "0123456789ABCDEF".
Sử dụng khoá (ở dạng thập phân): "133457799BBCDFFI". Khoá này ở dạng nhị phân là một chuỗi bit như sau (không có bit kiểm tra):
0001001001101001010110111100100110110111101101111111111000

- Chuyển đổi IP, chúng ta lấy ra L_0 và R_0:

L_0 = 11001100000000001100110011111111

$L_0 = R_0$ = 11110000101010101111000010101010

- 16 vòng mã hoá được thực hiện như sau:

Ví dụ mã hóa giải mã DES

$E(R_0)$	=	011110100001010101010101011110000101010101010101
K_1	=	000110110000001011101111111111000111000001110010
$E(R_0) \oplus K_1$	=	011000010001011110111010100001100110010100100111
Đầu ra S-Box	=	010111001000001010110100110010111
$f(R_0,K_1)$	=	001000110010101010100110111011
$L_2=R_1$	=	11101111010010100110010101000100

$E(R_1)$	=	011101011110101001010100001100001010101000001001
K_2	=	011110011010111011010011011011110010011110010101
$E(R_1) \oplus K_2$	=	000011000100010010001101111010110110001111101100
Đầu ra S-Box	=	111110001101000000111010101011110
$f(R_1,K_2)$	=	001111001010101110000111101000111
$L_3=R_2$	=	11001100000000001011101110001001

Ví dụ mã hóa giải mã DES

$E(R_2)$	=	111001011000000000000101011101011101000010100111
K_3	=	010101011111110010001010010000101100111110011001
$E(R_2) \oplus K_3$	=	101100000111110010001000111110000010011111001010
Đầu ra S-Box	=	001001110001000011100001010111111
$f(R_2,K_3)$	=	010011010001011001101110101110000
$L_4=R_3$	=	101000100101110000001011111110100

$E(R_3)$	=	010100000100001011111000000001010111111110101001
K_4	=	011100101010110111010110110111001101010000111101
$E(R_3) \oplus K_4$	=	001000101110111100101111011011110010101010110100
Đầu ra S-Box	=	001000011110110100111100111010
$f(R_3,K_4)$	=	101110110010001101110111001100
$L_5=R_4$	=	011101110010001000000000001000101

Ví dụ mã hóa giải mã DES

$E(R_4)$	=	101110101110100100001000000000000000001000001010
K_5	=	011111001110110000000111111010110101001110101000
$E(R_4) \oplus K_5$	=	110001100000010100000011111010110101000110100010
Đầu ra S-Box	=	010100001100100000110001111101011
$f(R_4,K_5)$	=	001010000001001110101110111000011
$L_6=R_5$	=	100010100100111101001100110111

$E(R_5)$	=	110001010100001001011111110100001100000110101111
K_6	=	011000111010010100111110010100000111101100101111
$E(R_5) \oplus K_6$	=	101001101110011101100001100000001011101010000000
Đầu ra S-Box	=	010000011110011010011000011110
$F(R_5,K_6)$	=	100111100100010111001101001001100
$L_7=R_6$	=	111010010110011111001101011010011

Ví dụ mã hóa giải mã DES

$E(R_6)$	=	111101010010101100001111111001011010101101010011
K_7	=	111011001000010010110111111111011000011000101111100
$E(R_6) \oplus K_7$	=	000110011010111110111100000001001110110011111101111
Đầu ra S-Box	=	000100000111010101000000101011101
$F(R_6,K_7)$	=	100011000000010100011100001001111
$L_8=R_7$	=	000001100100101010111010000100000

$E(R_7)$	=	000000000110000100101010101111101000000101000000
K_8	=	111101111000101000111010110000010011101111111011
$E(R_7) \oplus K_8$	=	111101110100100001101111100111001111010110111011
Đầu ra S-Box	=	011011000001100001111010101110
$F(R_7,K_8)$	=	001111000000111010000110111111001
$L_9=R_8$	=	110101010110100101001011110010000

Ví dụ mã hóa giải mã DES

$E(R_8)$	=	011010101010101101010010101001011111001010000001
K_9	=	111000001101101111101011111101101111100111000001
$E(R_8) \oplus K_9$	=	100010100111000010111001010010001101101001000000
Đầu ra S-Box	=	000100010000110001010111101111
$F(R_8,K_9)$	=	001000100011011001111100011010
$L_{10}=R_9$	=	001001000111110011000110011111010

$E(R_9)$	=	000100001000000111111100101100000110000111111010
K_{10}	=	101100011111001101000111101110100100011001001111
$E(R_9) \oplus K_{10}$	=	101000010111000010111111011010101000010110111011
Đầu ra S-Box	=	110110100000010001010010011110101
$F(R_9,K_{10})$	=	011000101011110010011100001000010
$L_{11}=R_{10}$	=	101101111101010111010111110110010

Ví dụ mã hóa giải mã DES

$E(R_{10})$	=	01011010111111101010111110101011111101101001
K_{11}	=	00100001010111111101001111011110101001110000110
$E(R_{10}) \oplus K_{11}$	=	01111011101000010111100000110100001011000100011
Đầu ra S-Box	=	01110011000001011010001000000001
$f(R_{10},K_{11})$	=	11100001000001001111101000000010
$L_{12}=R_{11}$	=	11000101011110000011110001111000

$E(R_{11})$	=	01100000101010111111000000011111100000111110001
K_{12}	=	01110101011100011111010110010100011001111101001
$E(R_{11}) \oplus K_{12}$	=	00010101111011010000001011000101111100100000011000
Đầu ra S-Box	=	01111011100010110010011000110101
$f(R_{11},K_{12})$	=	11000010011010001100111111101010
$L_{13}=R_{12}$	=	01110101101111010001100001011000

Ví dụ mã hóa giải mã DES

$E(R_{12})$	=	00111010101111011111101010001111000000101111000
K_{13}	=	10010111110001011010001111110101011101001000001
$E(R_{12}) \oplus K_{13}$	=	10101101011110000001010110111010111000010110001
Đầu ra S-Box	=	10011010110100011000101101001111
$f(R_{12},K_{13})$	=	11011101101110110010100100100010
$L_{14}=R_{13}$	=	00011000110000110001010101011010
$E(R_{13})$	=	00001111000101100000011010001010101010111110100
K_{14}	=	01011111010000111010111111100101110011100111010
$E(R_{13}) \oplus K_{14}$	=	01010000010101011011000101111000010011011001110
Đầu ra S-Box	=	01100100011110011001101011110001
$f(R_{13},K_{14})$	=	10110111001100011000111001010101
$L_{15}=R_{14}$	=	11000010100011100100101100000110

Ví dụ mã hóa giải mã DES

$E(R_{14})$	=	11100000010101000101100101001010101100000001011011
K_{15}	=	10111111100100011000110100111101001111110000101 0
$E(R_{14}) \oplus K_{15}$	=	01011111111000101110101000111011111111111101010001
Đầu ra S-Box	=	10110010111010001000110100111100
$f(R_{14},K_{15})$	=	01011011100000010010011101101110
$L_{16}=R_{15}$	=	01000011010000100011001000110100

$E(R_{15})$	=	00100000011010100000010000011010010000011010100 0
K_{16}	=	11001011001111011000101100001110000101111111010 1
$E(R_{15}) \oplus K_{16}$	=	11101011010101111000111100010100010101100101110 1
Đầu ra S-Box	=	10100111100000011001001000010 1
$f(R_{15},K_{16})$	=	11001000110000000100111110011000
R_{16}	=	00001010010011001101100110010101

Đề Test thử

- Cho bản rõ mang nội dung: *x="0123D56789ABCDE8".*
- Cho khoá *K=183457799B3CDFF2*

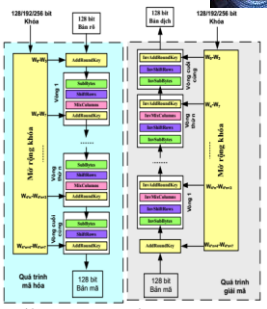
Trong hệ cơ số 16, Thực hiện mã hóa văn bản rõ trên theo thuật toán DES
Bằng phần mềm Demo

Mật mã AES (Advanced Encryption Standard)

- Tổng quan.
  - AES (viết tắt của từ tiếng anh: Advanced Encryption Standard, hay Tiêu chuẩn mã hóa nâng cao) là một thuật toán mã hóa khối được chính phủ Hoa Kỳ áp dụng làm tiêu chuẩn mã hóa.
  - Thuật toán được xây dựng dựa trên Rijndael Cipher phát triển bởi 2 nhà mật mã học người Bỉ: Joan Daemen và Vincent Rijmen.
  - AES làm việc với các khối dữ liệu 128bit và độ dài khóa 128bit, 192bit hoặc 256bit. Các khóa mở rộng sử dụng trong chu trình được tạo ra bởi thủ tục sinh khóa Rijndael.
  - Hầu hết các phép toán trong thuật toán AES đều thực hiện trong một trường hữu hạn của các byte. Mỗi khối dữ liệu đầu vào 128bit được chia thành 16byte, có thể xếp thành 4 cột, mỗi cột 4 phần tử hay một ma trận 4x4 của các byte, nó gọi là ma trận trạng thái.
  - Tùy thuộc vào độ dài của khóa khi sử dụng 128bit, 192bit hay 256bit mà thuật toán được thực hiện với số lần lặp khác nhau.

Mật mã AES (Advanced Encryption Standard)

- Các bước xử lý chính.
  - Quá trình mở rộng khóa sử dụng thủ tục sinh khóa Rijndael.
  - Quá trình mã hóa.



n là số lần lặp 9, 11, 13 tương ứng với khóa đầu vào 128, 192, 256

Mật mã AES (Advanced Encryption Standard)

- Xây dựng thuật toán.
  - Xây dựng bảng S-box.
  - Bảng S – box thuận.
    - Bảng S-box thuận được sinh ra bằng việc xác định nghịch đảo cho một giá trị nhất định trên GF(2^8) = GF(2)[x] / (x^8+x^4+x^3+x+1) (trường hữu hạn Rijindael). Giá trị 0 không có nghịch đảo thì được ánh xạ với 0. Những nghịch đảo được chuyển đổi thông qua phép biến đổi affine.
    - Công thức tính các giá trị bảng S-box và bảng S- box tương ứng: (trang bên)

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Mật mã AES (Advanced Encryption Standard)

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0x	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1x	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2x	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3x	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4x	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5x	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6x	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7x	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8x	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9x	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
ax	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
bx	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
cx	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
dx	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
ex	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
fx	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Mật mã AES (Advanced Encryption Standard)

- Bảng S-box nghịch đảo.
  - S-box nghịch đảo chỉ đơn giản là S-box chạy ngược. Nó được tính bằng phép biến đổi affine nghịch đảo các giá trị đầu vào. Phép biến đổi affine nghịch đảo được biểu diễn như sau: (trang sau)

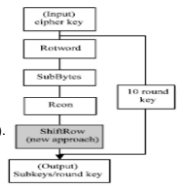
$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Mật mã AES (Advanced Encryption Standard)

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0x	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
1x	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
2x	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
3x	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
4x	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
5x	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
6x	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
7x	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
8x	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
9x	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
ax	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
bx	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
cx	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
dx	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
ex	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
fx	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

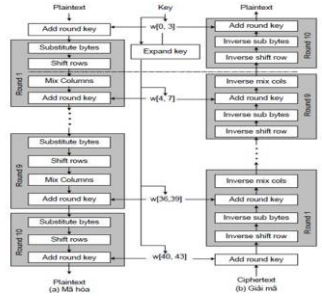
Mật mã AES (Advanced Encryption Standard)

- Giải thuật sinh khóa vòng:
  - Quá trình sinh khóa gồm 4 bước:
    - Rotword: quay trái 8 bít
    - SubBytes
    - Rcon: tính giá trị Rcon(i) Trong đó:
      - Rcon(i) = x(i-1) mod (x^8 + x^4 + x^3 + x + 1).
    - ShiftRow



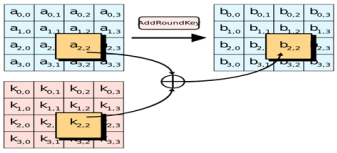
Mật mã AES (Advanced Encryption Standard)

- Quá trình mã hóa.
  - Sơ đồ tổng quát



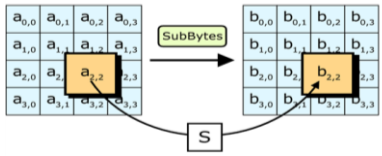
Mật mã AES (Advanced Encryption Standard)

- Hàm AddRoundKey.
  - Được áp dụng từ vòng lặp thứ 1 tới vòng lặp Nr
  - Trong biến đổi Addroundkey(), một khóa vòng được cộng với state bằng một phép XOR theo từng bit đơn giản.
  - Mỗi khóa vòng gồm có 4 từ (128 bit) được lấy từ lịch trình khóa. 4 từ đó được cộng vào mỗi cột của state, sao cho:



Mật mã AES (Advanced Encryption Standard)

- Hàm SubBytes.
  - Biến đổi SubBytes() thay thế mỗi byte riêng rẽ của state Sr,c bằng một giá trị mới S'r,c sử dụng bảng thay thế (S - box) được xây dựng ở trên.

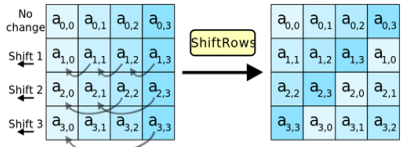


Mật mã AES (Advanced Encryption Standard)

- Hàm ShiftRow.
  - Trong biến đổi ShiftRows(), các byte trong ba hàng cuối cùng của trạng thái được dịch vòng đi các số byte khác nhau (độ lệch). Cụ thể:
  - S'r,c = Sr,(c + shift (r, Nb)) mod Nb (Nb = 4)
  - Trong đó giá trị dịch shift (r, Nb) phụ thuộc vào số hàng r như sau:
    - Shift(1,4) = 1, shift(2,4) = 2, shift(3,4) = 3.
    - Hàng đầu tiên không bị dịch, ba hàng còn lại bị dịch tương ứng:
    - Hàng thứ 1 giữ nguyên.
    - Hàng thứ 2 dịch vòng trái 1 lần.
    - Hàng thứ 3 dịch vòng trái 2 lần.
    - Hàng thứ 4 dịch vòng trái 3 lần.

Mật mã AES (Advanced Encryption Standard)

- Hàm ShiftRow.

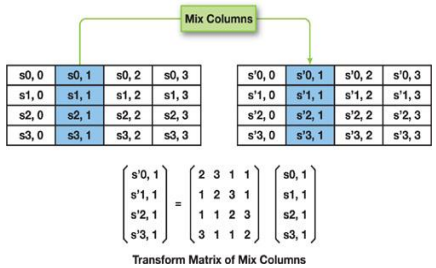


Mật mã AES (Advanced Encryption Standard)

- Hàm MixColumns.
  - Biến đổi MixColumns() tính toán trên từng cột của state. Các cột được coi như là đa thức trong trường GF(28) và nhân với một đa thức a(x) với:
  - a(x) = (03)x^3 +(01)x^2 +(01)x + (02)
  - Biến đổi này có thể được trình bày như phép nhân một ma trận, mà mỗi byte được hiểu như là một phần tử trong trường GF(28): s'(x) = a(x) XOR s(x):
  - Mô tả bằng ma trận như sau :

Mật mã AES (Advanced Encryption Standard)

- Hàm MixColumns.



Mật mã AES (Advanced Encryption Standard)

- Quá trình giải mã.
  - Tổng quan.
    - Thuật toán giải mã khá giống với thuật toán mã hóa về mặt cấu trúc nhưng 4 hàm sử dụng là 4 hàm ngược của quá trình mã hóa.

Mã Hóa	Giải Mã
AddRoundKey()	InvAddRoundKey()
SubBytes()	InvSubBytes()
ShiftRows()	InvShiftRows()
MixColumns()	InvMixColumns()

Mật mã AES (Advanced Encryption Standard)

- Thuật toán giải mã.

```
InvCipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
begin
byte state[4,Nb]
state = in
AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])
for round = Nr-1 downto 1
   InvShiftRows(state)
   InvSubBytes(state)
   AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
   InvMixColumns(state)
end for
InvShiftRows(state)
InvSubBytes(state)
AddRoundKey(state, w[0, Nb-1])
out = state
end
```

Trong đó :
- In[] : Mảng dự liệu đầu vào Input.
- Out[] : Mảng dữ liệu đầu ra Output.
- Nr : Số vòng lặp.(Nr = 10).
- Nb : Số cột(Nb = 4).
- W[] : Mảng các w[i] có độ dài 4 bytes.

Mật mã AES (Advanced Encryption Standard)

- Các dạng tấn công vào AES và phương pháp phòng chống
  - Side-channel attack.
    - Side Channels (Kênh kề) được định nghĩa là các kênh đầu ra không mong muốn từ một hệ thống.
    - Tấn công kênh bên hay còn gọi là Tấn công kênh kề là loại tấn công để thực hiện trong các loại tấn công mạnh chống lại quá trình triển khai mã hóa, và mục tiêu của loại tấn công này là phân tích các nguyên tố, các giao thức, modul, và các thiết bị trong mỗi hệ thống.
    - Phân loại :
      - Tấn công thời gian.
      - Tấn công dựa vào lỗi.
      - Tấn công phân tích năng lượng.
      - Tấn công phân tích điện từ.

Mật mã AES (Advanced Encryption Standard)

- Các dạng tấn công vào AES và phương pháp phòng chống
  - **Known attacks.**
    - Vào năm 2002, Nicolas Courtois và Josef Pieprzyk phát hiện một tấn công trên lý thuyết gọi là tấn công XSL và chỉ ra điểm yếu tiềm tàng của AES.
    - Tuy nhiên, một vài chuyên gia về mật mã học khác cũng chỉ ra một số vấn đề trong cơ sở toán học của tấn công này và cho rằng các tác giả đã có sai lầm trong tính toán. Việc tấn công dạng này có thực sự trở thành hiện thực hay không vẫn còn để ngỏ và cho tới nay thì tấn công XSL vẫn chỉ là suy đoán.

Mật mã AES (Advanced Encryption Standard)

- Các dạng tấn công vào AES và phương pháp phòng chống
  - Các phương pháp phòng chống.
    - Phương pháp 1: Mã hóa cực mạnh
      - Sử dụng các biện pháp để tăng tính bảo mật của các thuật toán mã hóa.
    - Phương pháp 2: Bảo vệ dữ liệu theo phương pháp vật lý
      - Nếu một kẻ tấn công không thể tiếp cận vật lý với dữ liệu, dĩ nhiên khả năng đánh cắp khóa mã hóa sẽ khó khăn hơn. Vì vậy, trước những cuộc tấn công qua âm thanh tiềm tàng, bạn có thể sử dụng các giải pháp bảo vệ vật lý như đặt laptop vào các hộp cách ly âm thanh, không để ai lại gần máy tính khi đang giải mã dữ liệu hoặc sử dụng các nguồn âm thanh băng rộng tần số đủ cao để gây nhiễu.
    - Phương pháp 3: Kết hợp cả 2 cách trên.

Hết chương 3