

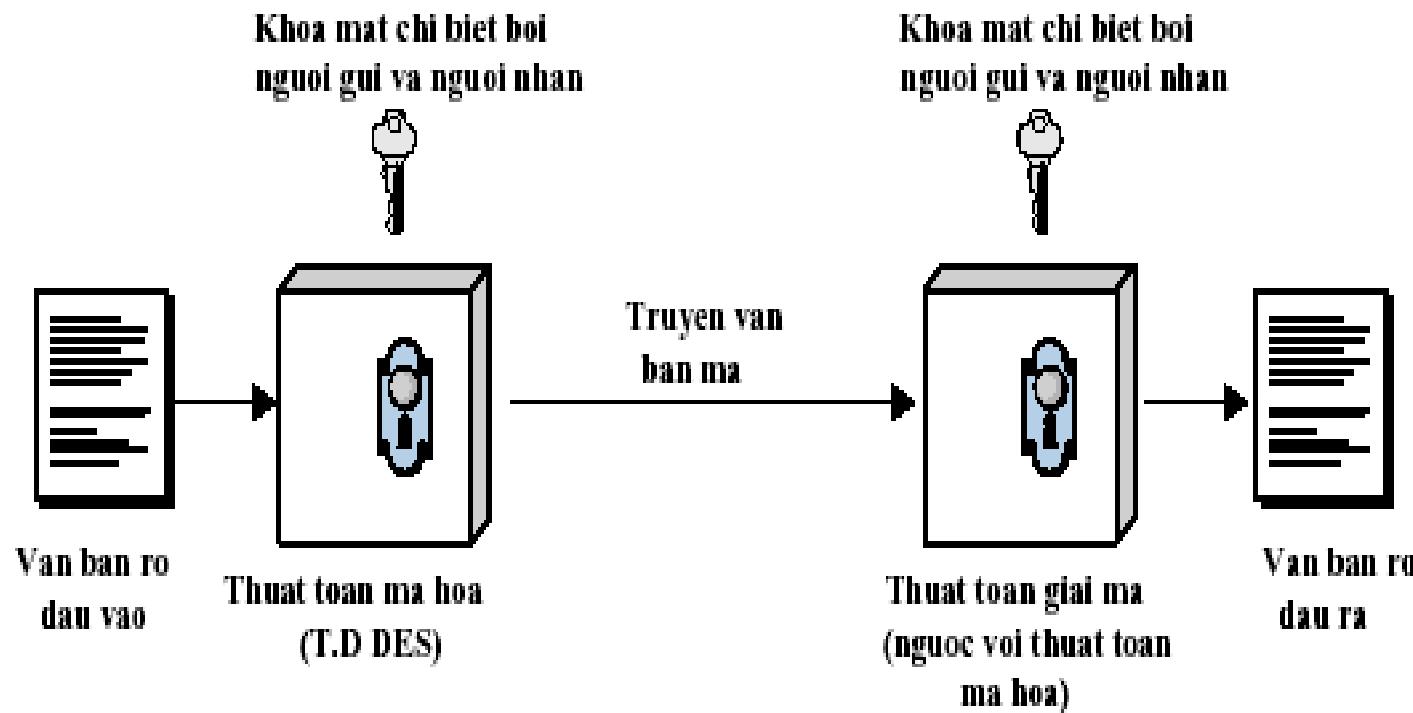
# Các vấn đề trình bày

1. Quá trình phát triển của mật mã hiện đại
2. Nguyên tắc xây dựng thuật toán khóa bí mật
3. Chuẩn mã hóa dữ liệu – DES
4. Chuẩn mật mã nâng cao – AES
5. Một số thuật toán khóa đối xứng: Twofish, Mars, RC6, Serpent
6. Một số phương pháp thám mã hệ mật khóa bí mật

# 1. Quá trình phát triển

- Mật mã truyền thống (mật mã đối xứng, mật mã với một khoá), cho đến khi phát minh ra mật mã với khoá công khai, đã là phương pháp duy nhất của mật mã.
- Ngày nay phương pháp này vẫn tiếp tục được phát triển.

# Mô hình đơn giản của mật mã truyền thống



# Mô tả

- *Bản rõ (plain text)*: các tin tức rõ nghĩa ban đầu.
- *Bản mã (cipher text)*: dạng biến đổi của bản rõ.
- Quá trình mã hoá bao gồm việc sử dụng thuật toán và khoá nào đó.
- Khoá (*key*): đó là một giá trị, được gọi là *khoá mật*, không phụ thuộc vào bản rõ.

# (tiếp)

- Khi có bản rõ X và khoá mật K, nhờ thuật toán mã hoá mà bản mã  $Y = [Y_1, Y_2, \dots, Y_M]$ . Điều này có thể viết dưới dạng công thức sau:

$$Y = E_K(X)$$

- Người nhận tin tức, giả thiết rằng bằng một cách nào đó, cũng có khoá mật K, cần phải có khả năng thực hiện biến đổi ngược:

$$X = D_K(Y).$$

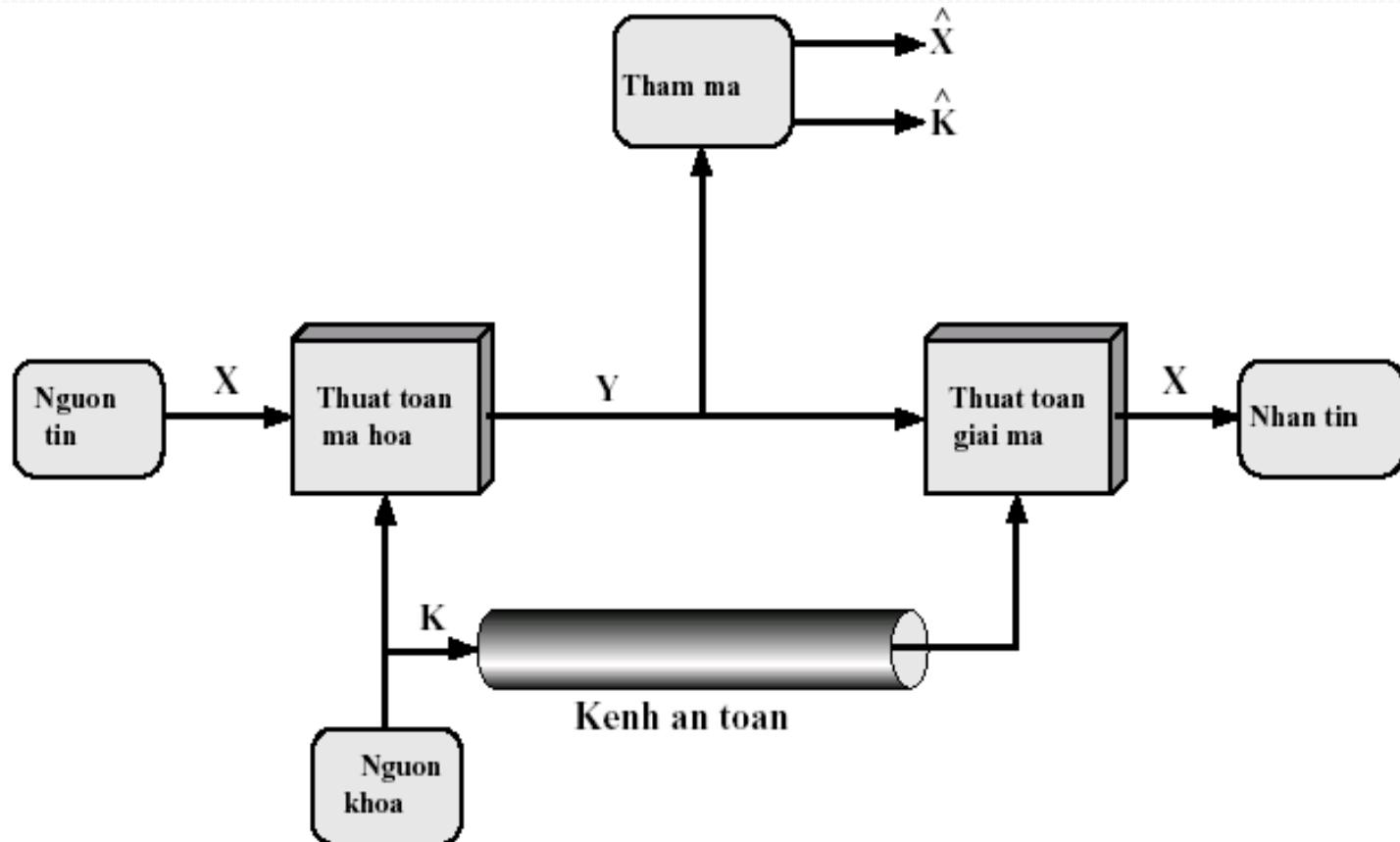
# Nhận xét

- Kết quả đạt được, khi thực hiện thuật toán, phụ thuộc vào việc sử dụng khoá.
- Sự thay đổi khoá dẫn đến việc thay đổi kết quả đạt được của thuật toán.

# Độ tin cậy của mật mã truyền thống

- Thuật toán mật mã cần phải phức tạp, để không có khả năng giải mã, khi chỉ có văn bản mã.
- Thứ hai, yếu tố cơ bản độ tin cậy của mật mã truyền thống là khoá mật, trong đó chính thuật toán mật mã không cần bí mật.

# Mô hình của mật mã truyền thống



# Phân loại mật mã khóa đối xứng

- Mã khối - thực hiện biến đổi khối dữ liệu với một kích thước không đổi.
- Mã dòng - thực hiện biến đổi tuần tự từng bit hoặc byte riêng rẽ.

# Thám mã

- Quá trình khôi phục giá trị X hoặc là K, hoặc cả hai được gọi là thám mã.
- Chiến thuật thám mã được sử dụng phụ thuộc vào sơ đồ mã hoá và vào những thông tin có được trong khi tiến hành.

# Các dạng thám mã

Dạng thám mã	Các số liệu mà thám mã biết
Khi chỉ có bản mã (ciphertext only)	<ul style="list-style-type: none"><li>❖ Thuật toán mã hóa</li><li>❖ Bản mã</li></ul>
Khi biết bản rõ (know plaintext)	<ul style="list-style-type: none"><li>❖ Thuật toán mã hóa</li><li>❖ Bản mã</li><li>❖ Có một hoặc một vài cặp tương ứng của giữa bản mã và bản rõ, được tạo ra từ cùng một khóa mật</li></ul>

# (tiếp)

Phân tích với  
bản mã chọn  
lựa (chosen  
ciphertext)

- ❖ Thuật toán mã hoá
- ❖ Bản mã
- ❖ Văn bản mã chọn lựa để phù hợp với văn bản rõ, được mã hoá cùng một khoá mật, được thực hiện bởi người thám mã

# (tiếp)

Phân tích với  
bản rõ chọn  
lựa (chosen  
plaintext)

- ❖ Thuật toán mã hoá
- ❖ Bản mã
- ❖ Văn bản rõ chọn lựa để phù hợp với văn bản mã, được tạo ra cùng một khoá mật, được thực hiện bởi người thám mã

# (tiếp)

Phân tích với bản  
chọn lựa  
(Chosen text)

- ❖ Thuật toán mã hoá
- ❖ Bản mã
- ❖ Văn bản rõ chọn lựa để phù hợp với văn bản mã, được tạo ra cùng một khoá mật, được thực hiện bởi người thám mã
- ❖ Văn bản mã chọn lựa để phù hợp với văn bản rõ, được mã hoá cùng một khoá mật, được thực hiện bởi người thám mã

# Nhận xét

- Bài toán phức tạp nhất từ tất cả các bài toán được trình bày trong bảng này là trường hợp khi tiến hành người thám mã *chỉ có văn bản mã*.
- Trong một số trường hợp nào đó, thậm chí còn không biết cả thuật toán mã hoá, nhưng về cơ bản chúng ta giả thiết rằng người thám mã biết thuật toán mã hoá.

# (tiếp)

- Một xu thế thám mã là thử chọn tất cả các khả năng của khoá.
- Tuy nhiên, nếu không gian về khả năng của khoá rất lớn, thì xu thế này tỏ ra không thực tế.

## 2. Nguyên tắc xây dựng hệ mật khóa bí mật

- Khi thiết kế mật mã thì vấn đề đảm bảo độ vững chắc của thuật toán là một vấn đề quan trọng nhất.
- ❖ Đánh giá độ bền vững của thuật toán là một trong các vấn đề lâu nhất và khó nhất.

# Các toán tử sử dụng trong mật mã khóa bí mật

- Phép hoán vị.
- Phép thay thế.
- Các phép toán số học: dịch vòng, XOR,

...

# Các sơ đồ mật mã nguyên thủy

- Sơ đồ Feistel
- Mạng hoán vị - thay thế (SPN)
- Sơ đồ kết hợp

## 2.1. Sơ đồ Feistel

- Rất nhiều thuật toán của mật mã khối đối xứng, được sử dụng ngày nay, được dựa trên cấu trúc gọi là “Mật mã khối Feistel” (Feistel block cipher).
- Thí dụ: DES, RC6, ...

# Các điều kiện tiên quyết tạo ra mật mã Feistel

- Giả thiết mật mã khối biến đổi n bit văn bản rõ thành khối văn bản mã có cùng độ dài → Số lượng các khối khác nhau sẽ là  $2^n$ .
- Một phép biến đổi như vậy, để đảm bảo khả năng giải mã phải là phép biến đổi thuận nghịch.

# Thí dụ: biến đổi thuận nghịch

<b>Biến đổi thuận nghịch</b>	
Văn bản rõ	Văn bản mã
00	11
01	10
10	00
11	01

# Thí dụ: biến đổi thuận nghịch

## Biến đổi không thuận nghịch

Văn bản rõ	Văn bản mã
00	11
01	10
10	01
11	01

# Mật mã Feistel

- Feistel đã đề nghị về việc xây dựng một loại mật mã khối, trong đó đồng thời sử dụng liên tiếp toán tử chuyển vị và toán tử thay thế, để nhận được độ an toàn cao hơn so với bất kỳ loại mật mã nào chỉ ứng dụng riêng biệt các toán tử.

# (tiếp)

- Phát triển mật mã khối (product cipher) với độ dài khóa  $k$  bit và khối biến đổi  $n$  bit, cho phép có tổng cộng  $2^k$  khả năng biến đổi.
- Mật mã (ideal block cipher) có  $2^n!$

# Ý tưởng

- Dựa trên ý tưởng của Claude Shannon về ý định gia công về một loại mật mã, trong đó có sự sử dụng cả hai chức năng khuếch tán (diffusion) và hỗn loạn (confusion).

# Khuếch tán và hỗn loạn

- Shannon đề xuất ra chúng để chống lại thám mã dựa trên phân tích thống kê.
- Tất cả các đặc trưng thống kê của bản mã là độc lập với khóa riêng biệt được sử dụng.

# Khuếch tán

- Bản chất của khuếch tán liên quan đến việc tán xạ mạnh của đặc trưng thống kê của văn bản rõ vào đặc trưng thống kê theo dải rộng của văn bản mã.
- Điều này đạt được bằng cách làm sao cho mỗi bit của văn bản rõ có ảnh hưởng tới nhiều bit của văn bản mã, hoặc có thể chỉ ra được mỗi phần tử bất kỳ của văn bản mã phụ thuộc vào một tập các phần tử của văn bản rõ.

# Thí dụ

- Ứng dụng phương pháp khuếch tán là mã hoá tin tức  $M = m_1, m_2, m_3, \dots$ , nhờ toán tử trung bình:

$$y_n = \sum_{i=1}^k m_{n+i} \pmod{26}$$

# Nhận xét

- Có thể chứng minh được rằng, trong trường hợp này, đặc trưng thống kê của văn bản rõ “được phân bố” theo văn bản mã.
- Bởi vậy, trong văn bản mã, đặc trưng tần suất sử dụng các chữ cái, sẽ tiến tới phân bố đều.

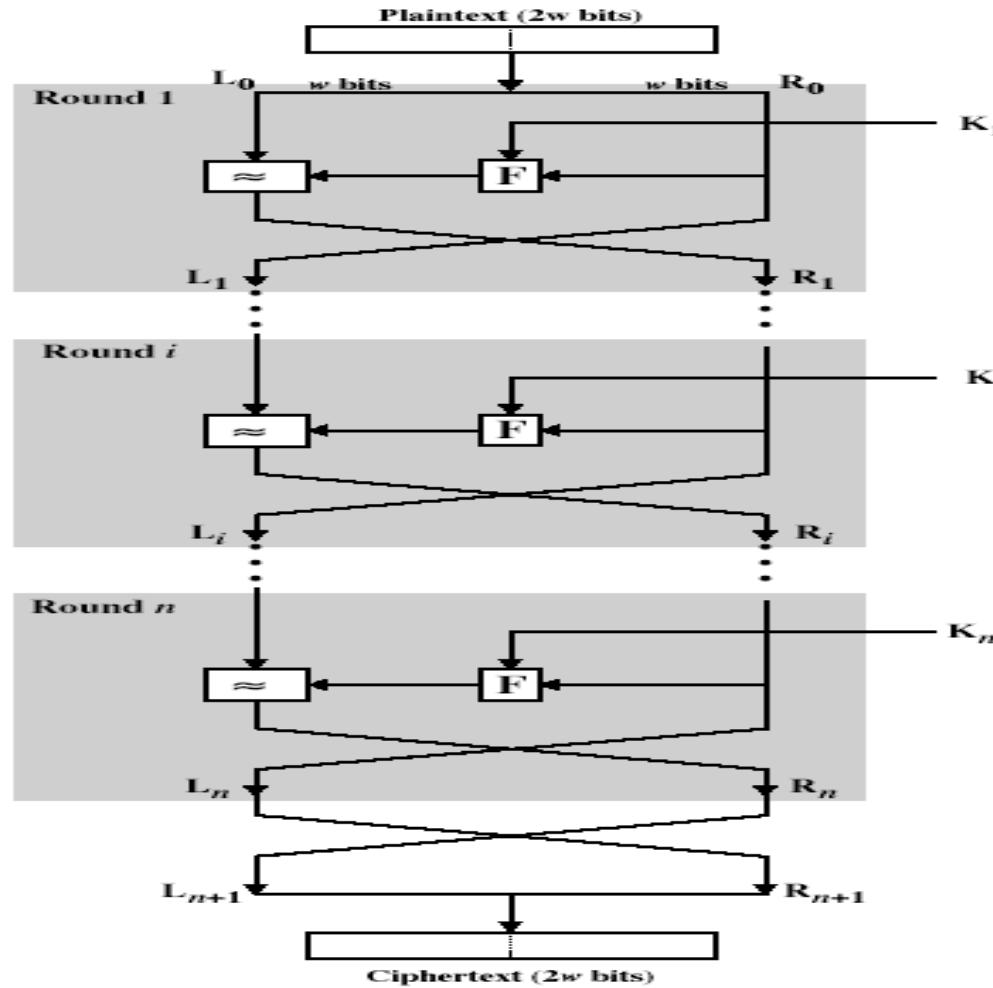
# Hỗn loạn

- Hiệu ứng hỗn loạn nhằm làm cho mối quan hệ giữa các đặc trưng thống kê của bản mã và giá trị của khóa mã trở nên phức tạp, chống lại khả năng cố gắng khôi phục khóa.
- Đồng thời, ngay cả khi nếu đổi phương có khả năng xác định được các đặc tính thống kê nào đó của văn bản mã, thì việc phức tạp của sự sử dụng khoá để nhận được văn bản mã cần chứng tỏ đạt được điều, mà mọi thử nghiệm muốn xác định khoá dựa trên các quan hệ thống kê đặc biệt này là không tưởng.
- Điều đó đạt được bằng các thuật toán chuyển vị phức tạp.

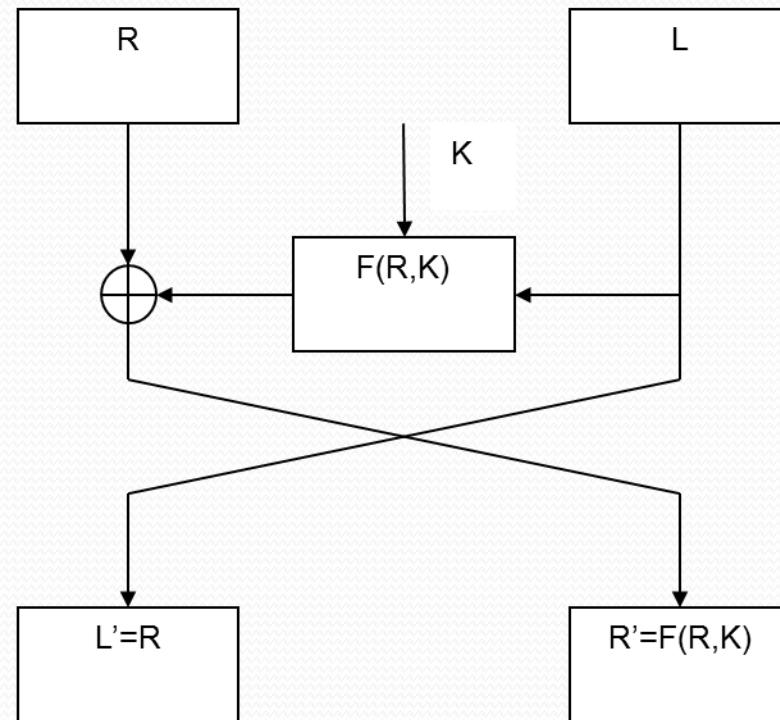
# Nhận xét

- Khái niệm khuếch tán và hỗn loạn đã thể hiện sự thành công đến mức trở thành quan điểm miêu tả bản chất của các đặc trưng mong đợi của mật mã khối.
- Các thuật ngữ này trở thành cơ sở đối với tất cả việc xây dựng các mật mã khối hiện đại.

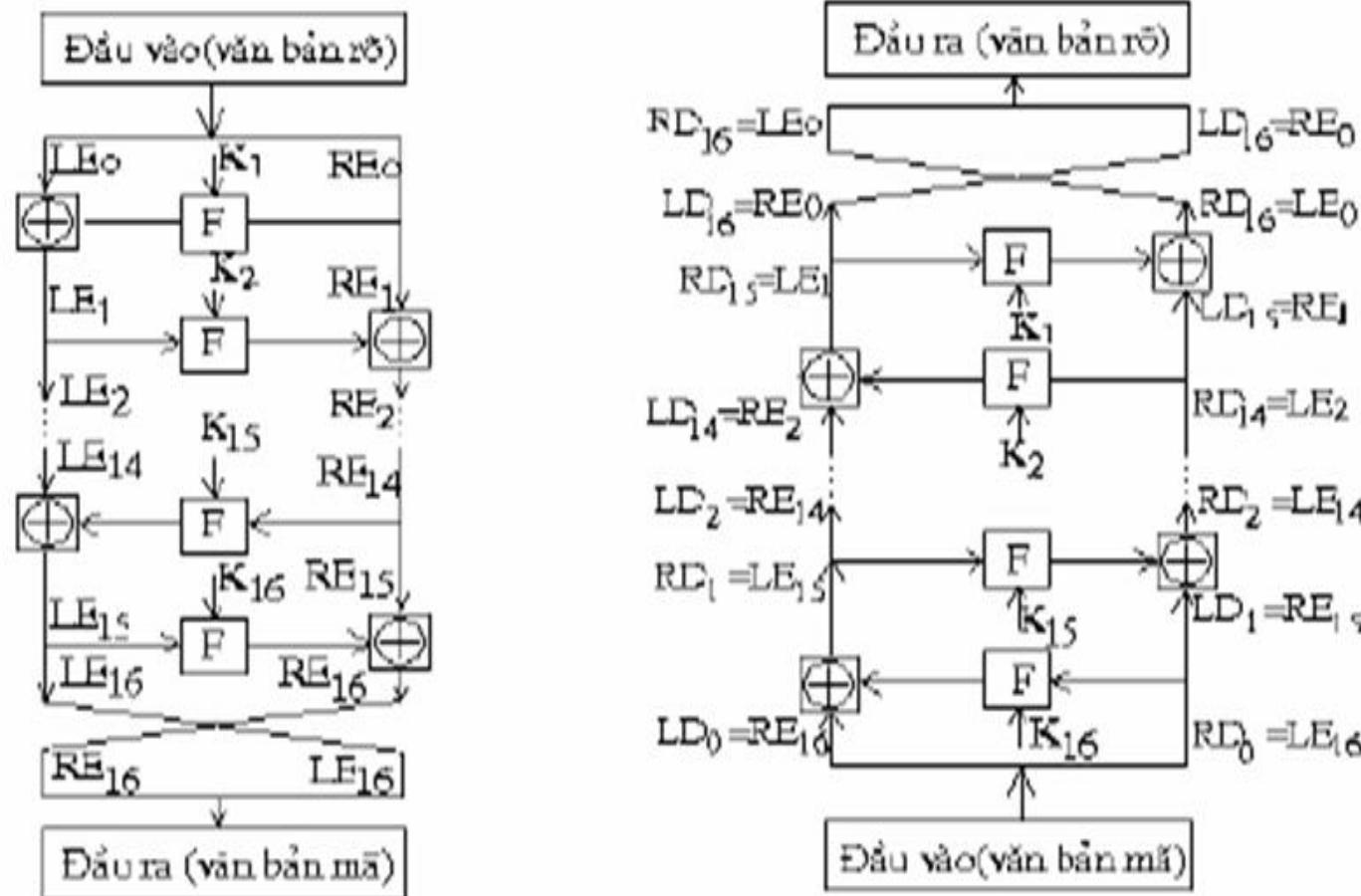
# Cấu trúc mật mã Feistel



# Sơ đồ một vòng mã hóa của Feistel



# Thuật toán mã/giải mã



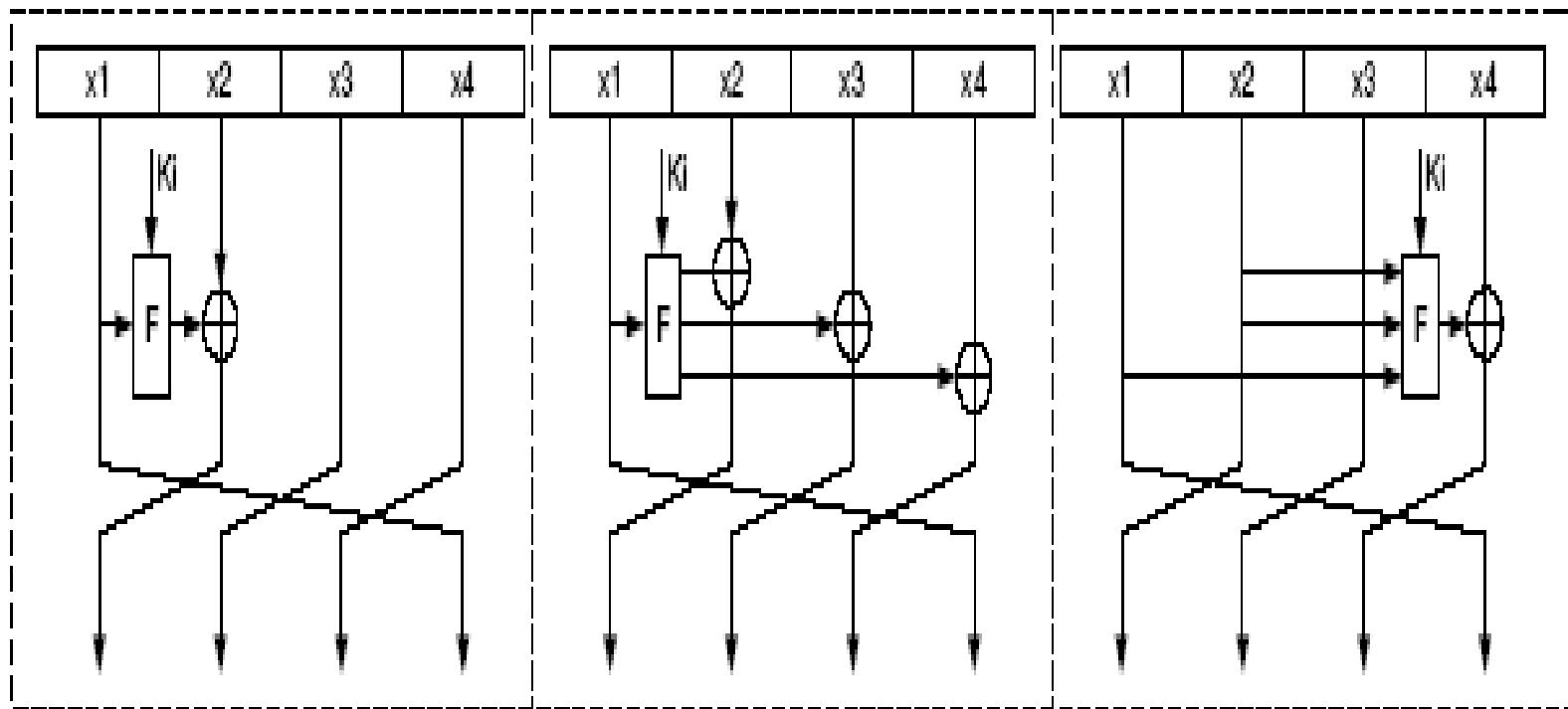
# Ưu điểm của mật mã Feistel

- Quá trình mã hóa và giải mã trùng nhau, chỉ khác nhau ở thứ tự khóa con, điều này sẽ tiết kiệm được nữa tài nguyên khi thực hiện thuật toán trên phần cứng.
- Hàm F có thể chọn với độ khó bất kỳ, vì không phải tìm hàm nghịch.

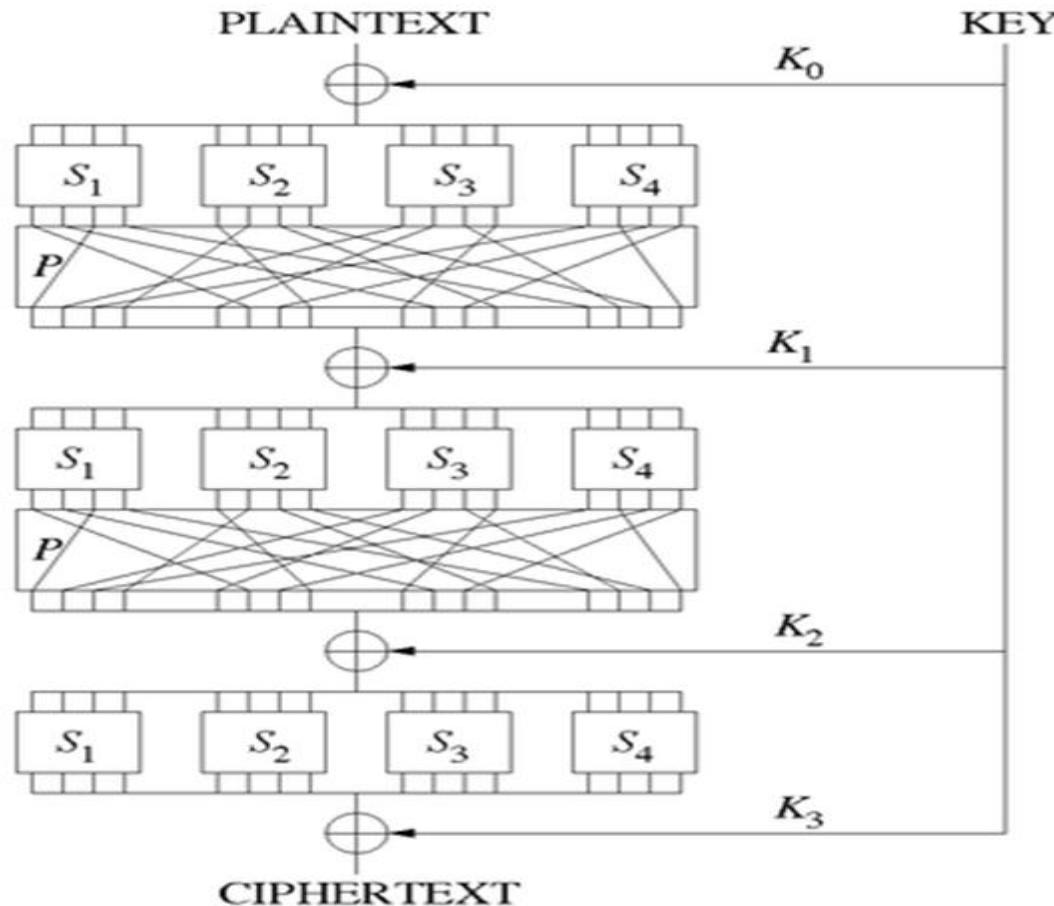
# Nhược điểm của mật mã Feistel

- Vì mỗi vòng mã chỉ thực hiện biến đổi nũa khối dữ liệu, nên cần số vòng mã hóa lớn để đảm bảo độ an toàn của hệ mật, điều này làm giảm đáng kể tốc độ mã.
- Ngoài ra xây dựng trên cơ sở mạng Feistel tồn tại lớp khóa tương đương, nên làm không gian khóa giảm đi một nǔa.

# Một số kiểu mở rộng của mật mã Feistel



## 2.2. Mạng hoán vị-thay thế



### **3. Chuẩn mã hóa dữ liệu DES**

#### 3.1. Lịch sử phát triển

➤ Chuẩn mật mã dữ liệu DES (Data Encryption Standard), được chấp nhận vào năm 1977 bởi văn phòng tiêu chuẩn quốc gia (NBS) của Mỹ (hiện là viện quốc gia của tiêu chuẩn và công nghệ – NIST).

# (tiếp)

- Tên chính thức là chuẩn xử lý thông tin của liên bang số 46 (FIPS PUB 46).
- Những năm 60 của thế kỷ XX, IBM bắt đầu dự án trong lĩnh vực mật mã do Feistel dẫn đầu → tạo ra LUCIPHER (kích thước khối dữ liệu 64 bit và sử dụng khoá dài 128 bit).

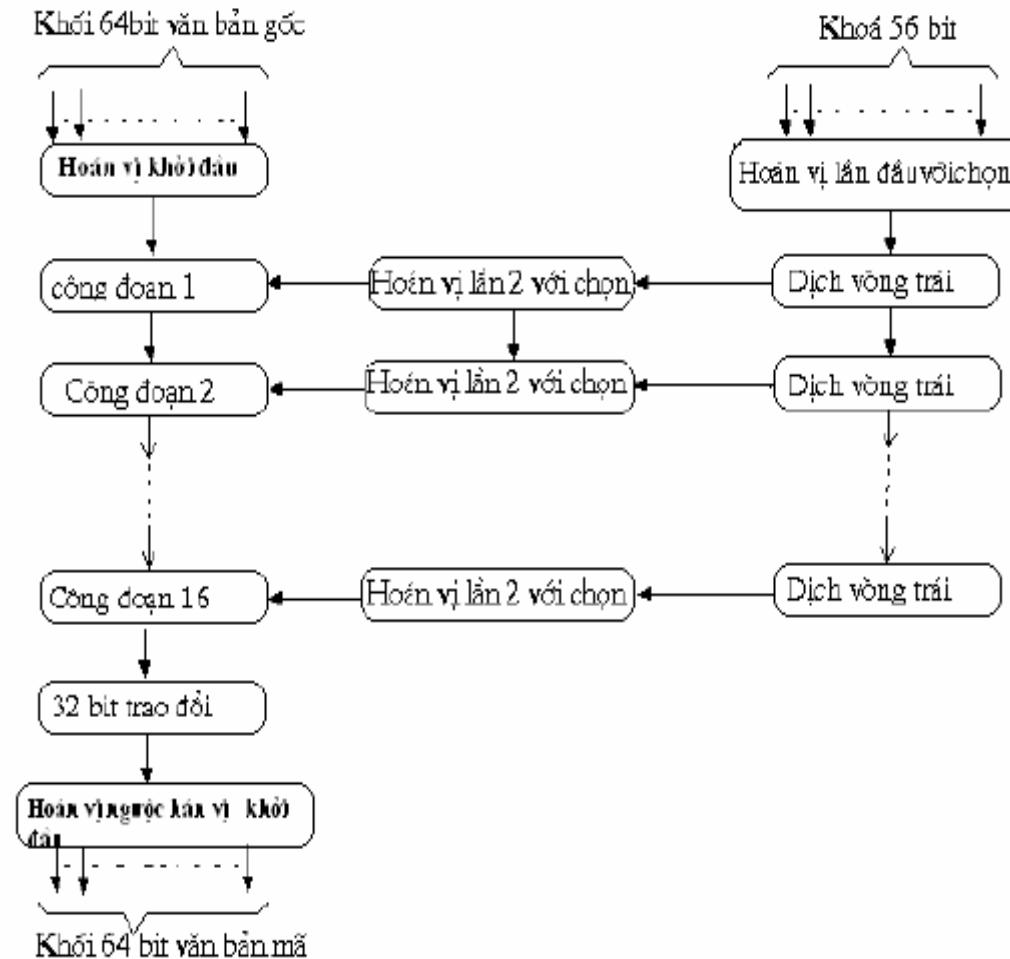
# (tiếp)

- Dựa trên kết quả của LUCIPHER, IBM tạo ra phương án thương mại hóa mật mã (tích hợp trong một IC). Dự án được dẫn đầu bởi Tuchman và Carl Meyer.
- IBM đưa ra trong cuộc thi các kết quả của dự án Tuchman-Meyer → vào năm 1977 nó đã được công nhận là chuẩn mật mã dữ liệu (DES).

# Nhận xét

- Thuật toán LUCIFER, hãng IBM đã sử dụng khoá dài 128 bit.
- Trong DES, khoá có độ dài 56 bit.
- Có 2 phê phán chính:
  - Độ dài khoá
  - Cấu trúc của ma trận S

## 3.2. Mã hóa DES



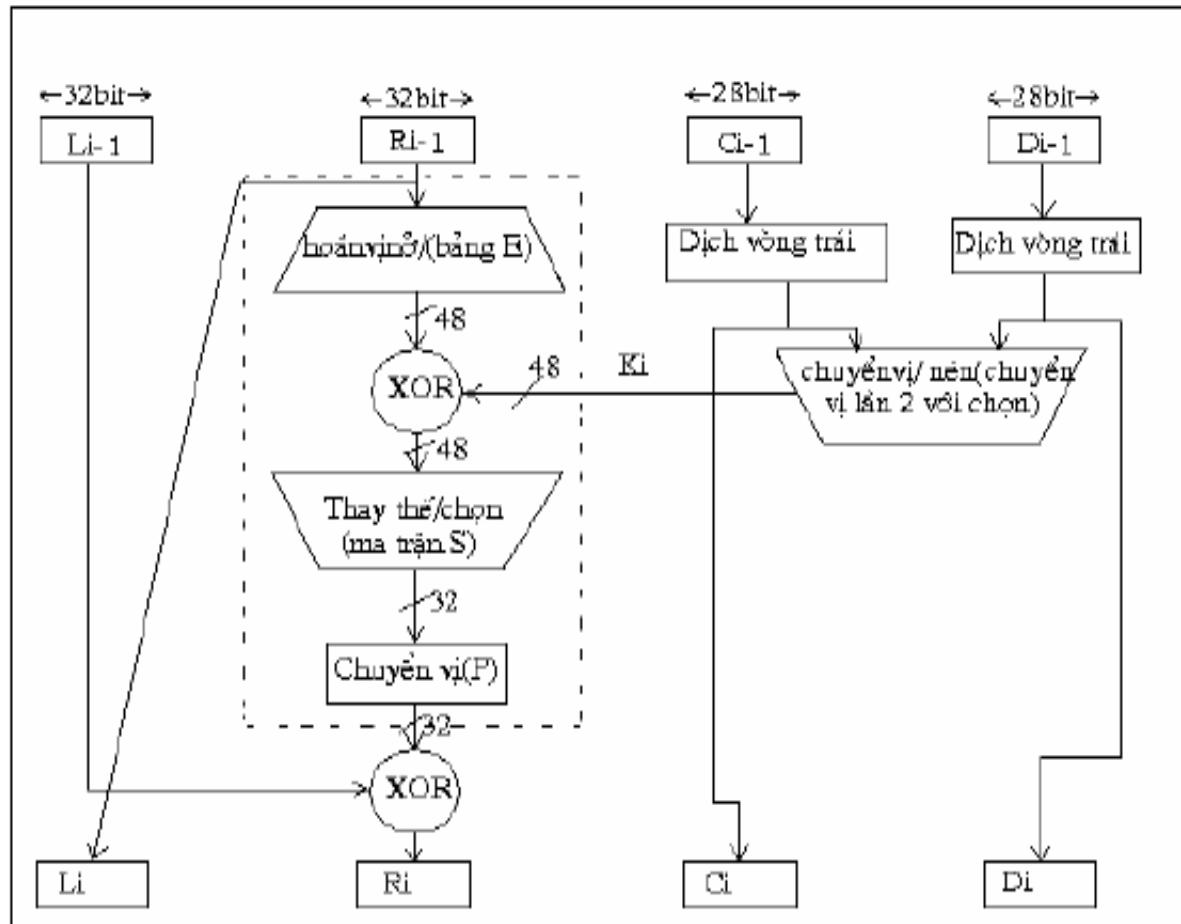
# Quá trình biến đổi văn bản rõ gồm ba bước

- Khối 64 bit của bản rõ được thực hiện biến đổi ở khối hoán vị khởi đầu (IP),
- Thực hiện 16 vòng biến đổi với cùng một chức năng, trong đó sử dụng các toán tử thay thế và hoán vị.
- Hoán vị cuối ( $IP^{-1}$ ).

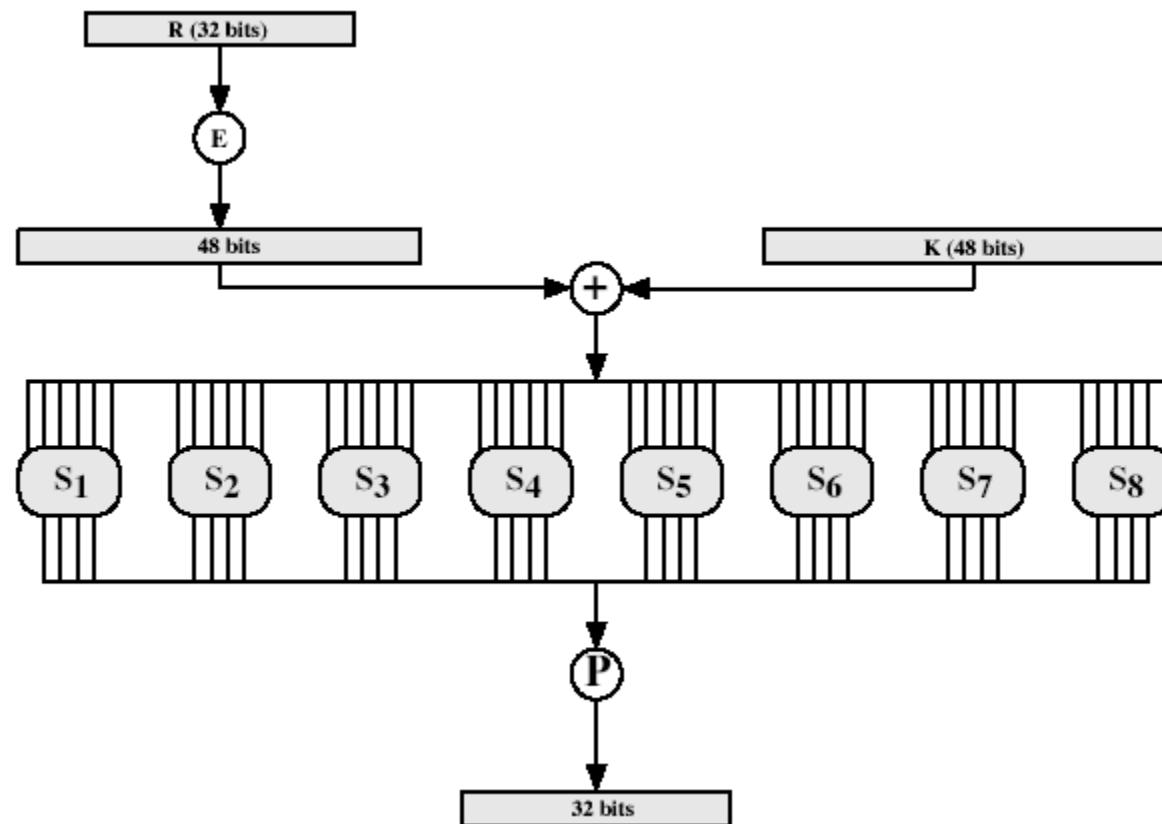
# Thủ tục sinh khóa vòng

- Hoán vị đầu.
- Dịch vòng trái.
- Hoán vị lần 2 với chọn.

# Một vòng mã hóa DES



# Hàm F(R,K)



# Các bảng hoán vị

a, Hoán vị khởi đầu(IP)

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

b, Hoán vị, ngược hoán vị khởi đầu ( $IP^{-1}$ )

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

# (tiếp)

c, Hoán vị với phép mở ( $\oplus$ )

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

### 3.3. Giải mã DES

- Để giải mã sử dụng cùng một thuật toán, nhưng đối với khoá để giải mã, khi tham gia vào thuật toán sẽ theo trình tự ngược.

## 3.4. Thảo luận về DES

- Hiệu ứng thác lũ
- Độ an toàn của DES

# Hiệu ứng thác lũ

- Đặc tính mong đợi → cần phải có độ nhạy cao của đầu ra (bản rõ, khóa) với sự thay đổi của dữ liệu đầu vào (bản mã), (hiệu ứng thác lũ - avalanche effect).

# (tiếp)

- Thuật toán DES thể hiện đạt được hiệu ứng thác lũ mạnh.
- Trường hợp 1:
  - Hai đoạn văn bản rõ, chỉ khác nhau có một bit là: “00000000 00000000 00000000 00000000 00000000 00000000 00000000” và “10000000 00000000 00000000 00000000 00000000 00000000 00000000”.
  - Khóa “0000001 1001011 0100100 1100010 0011100 0011000 0011000 0110010”.

# (tiếp)

- Trường hợp 2:
  - Bản rõ: “01101000 10000101 00101110 01111010 00010011 01110110 11101011 10100100”.
  - Hai khoá khác nhau, chúng khác nhau chỉ một bit ở vị trí thứ nhất là:”1110010 1111011 1101111 0011000 0011101 0000100 0110001 1101110” và “0110010 1111011 1101111 0011000 0011101 0000100 0110001 1101110”.

# Hiệu ứng thác lũ của DES

a) Sự thay đổi trong văn bản rõ		b) Sự thay đổi trong khoá	
Công đoạn	Số bit khác nhau	Công đoạn	Số bit khác nhau
0	1	0	0
1	6	1	2
2	21	2	14
3	35	3	28
4	39	4	32
5	34	5	30
6	32	6	32
7	31	7	35
8	29	8	34
9	42	9	40
10	44	10	38
11	32	11	31
12	30	12	33
13	30	13	28
14	26	14	26
15	29	15	34
16	34	16	35

# Độ an toàn của DES

- Sử dụng 56 bit khoá.

Độ dài khoá (bit)	Số lượng khoá khác nhau	Thời gian cần thiết khi tốc độ một lần giải mã/ $\mu$ sek	Thời gian cần thiết khi tốc độ $10^6$ lần giải mã/ $\mu$ sek
32	$2^{32} = 4,3 \times 10^9$	$2^{31} \mu\text{sek} = 35,8 \text{ ph}$	$2,15 \mu\text{sek}$
56	$2^{56} = 7,2 \times 10^{16}$	$2^{55} \mu\text{sek} = 1142 \text{ năm}$	10,01 giờ
128	$2^{128} = 3,4 \times 10^{38}$	$2^{127} \mu\text{sek} = 5,4 \times 10^{24} \text{ năm}$	$5,4 \times 10^{18} \text{ năm}$
26!	$26! = 4 \times 10^{26}$	$2 \times 10^{25} \mu\text{sek} = 6,4 \times 10^{12} \text{ năm}$	$6,4 \times 10^{18} \text{ năm}$

# Nhận xét

- 1977 Diffie và Hellman đã tin tưởng rằng đến một lúc nào đó công nghệ sẽ cho phép tạo ra thiết bị gồm từ một triệu ( $10^6$ ) các thiết bị mã hóa song song, trong đó mỗi thiết bị thực hiện mã hóa trong một chu trình mất 1  $\mu$ s.
- Theo đánh giá tại thời gian đó giá cả của thiết bị như vậy vào khoảng 20 triệu đôla Mỹ.

# Đề xuất của Viewner

Giá thành thiết bị (đôla)	Thời gian tìm kiếm mong đợi
100.000	35 giờ
1.000.000	3,5 giờ
10.000.000	21 phút

# Cuộc thi tuyển của RSA

- Cuộc thi đã được công bố bởi RSA vào 29 tháng 2 năm 1977.
- Một trong các người tham gia Rocke Veser.
- Xây dựng chương trình chọn khóa và thực hiện nó trên mạng Internet.

# (tiếp)

- Đề án bắt đầu vào ngày 18 tháng 3 năm 1997.
- Kết thúc thành công sau 96 ngày, sau khi lựa chọn khoảng một phần tư của tất cả các khả năng tổ hợp, đã tìm ra được khoá đúng.

# Cấu trúc bên trong của thuật toán DES

- Chỉ trích hộp S của DES được giữ bí mật.
- Tuy nhiên hộp S của DES khá hoàn thiện.

# Những tiêu chuẩn trong kiến trúc DES

- Cấu trúc của ma trận S.
- Hàm P.

# Ma trận S

- Không có một bit nào ở đầu ra của các ma trận S này lại có thể tiệm cận theo quan hệ hàm tuyến tính với các bit đầu vào (vì ma trận S là thành phần phi tuyến duy nhất).
- Mỗi dòng của ma trận S cần phải bao gồm tất cả 16 khả năng tổ hợp các bit đầu vào.

# (tiếp)

- Nếu các giá trị đầu vào của các ma trận S khác nhau chỉ một bit, thì các giá trị đầu ra cần phải khác nhau ít nhất là hai bit.
- Nếu các giá trị đầu vào của các ma trận S khác nhau hai bit ở giữa, thì các giá trị đầu ra cần phải khác nhau ít nhất hai bit.

# (tiếp)

- Nếu các giá trị đầu vào của các ma trận S khác nhau hai bit đầu tiên, và trùng nhau ở hai bit cuối cùng, thì các giá trị đầu ra không được trùng nhau.
- Đối với hiệu số khác nhau bất kì của 6 bit giá trị đầu vào không lớn hơn 8 từ 32 cặp giá trị đầu vào, có cùng một hiệu như vậy có thể cho một và chỉ một giá trị cố định của hiệu đầu ra.

# Hoán vị P

- 4 bit đầu ra của S ở vòng i cần phải phân bổ sao cho để hai bit trong chúng ảnh hưởng tới “các bit ở giữa” của vòng  $i + 1$ , còn hai bit còn lại thì ảnh hưởng tới các bit bên ngoài.
- 4 bit đầu ra của các S trong vòng tiếp sau, cần phải ảnh hưởng tới kết quả của sáu ma trận khác nhau và hai từ bốn bit đầu ra này không được rơi vào đầu vào của một ma trận S

# (tiếp)

- Đối với hai ma trận  $S$ :  $S_j$  và  $S_k$ , nếu bit đầu ra nào đó của ma trận  $S_j$  trong vòng tiếp theo ảnh hưởng tới các bit ở giữa của  $S_k$ , thì không có bit đầu ra nào của  $S_k$  được ảnh hưởng tới các bit ở giữa của  $S_j$ .
- Chẳng hạn, khi  $j = k$  không có bit đầu ra nào của  $S_j$  được gây ảnh hưởng tới các bit ở giữa của  $S_j$ .

# Số lượng vòng mã hóa

- Độ bền thám mã của mã Feistel phụ thuộc vào ba tham số kiến trúc:
- số lượng các vòng mã hóa,
- dạng hàm F
- và thuật toán tính toán khoá.

# Nhận xét

- Số lượng vòng mã hoá càng lớn thì sự khó khăn trong thám mã càng lớn, thậm chí ngay cả khi hàm F tương đối yếu.
- Trong trường hợp chung, số lượng các vòng mã hoá cần phải chọn sao cho đối với tất cả các phương pháp thám mã đã biết, phải bỏ ra một công sức lớn hơn khi thám mã bằng cách chọn tất cả các khả năng của khoá.

## 3.5. Thám mã vi sai và thám mã tuyến tính

- Bị phân tích chủ yếu là ở chỗ dễ bị tổn thương theo quan điểm thám mã với việc sử dụng chọn tất cả các khả năng của khoá, theo nguyên nhân độ dài của khoá nhỏ (56 bit).
- Khóa ngày càng dài (2-DES, 3-DES) → thám mã vét cạn không hiệu quả.

# Thám mã vi sai

- Cho đến năm 1990 trong các tài liệu công khai chưa thấy xuất hiện về phương pháp thám mã vi sai.
- Lần đầu tiên phương pháp thám mã vi sai đã được công bố công khai đối với mã khối FEAL, trong công trình của Murphy.
- Sau đó không lâu là công trình của Biham và Shamir.

# (tiếp)

- Phương pháp thám mã vi sai được đánh giá là phương pháp đầu tiên cho phép bẻ khoá DES với mức độ phức tạp của bài toán nhỏ hơn  $2^{55}$ .
- Theo Biham, với sự trợ giúp của phương pháp đã cho có thể dẫn đến thám mã thành công DES với độ phức tạp  $2^{47}$ , nhưng cần phải có  $2^{47}$  bản rõ chọn lựa.

# Nhận xét

- Mặc dù  $2^{47}$  rõ ràng rất nhỏ hơn rất nhiều so với  $2^{55}$ , nhưng sự cần thiết phải có  $2^{47}$  bản rõ chọn lựa, làm cho phương pháp thám mã đã cho trở nên thuận tuý lý thuyết.

# (tiếp)

- Mặc dù phương pháp thám mã vi sai là công cụ mạnh, nhưng để chống lại DES, nó tỏ ra không hoàn toàn hiệu quả.
- Nguyên nhân ở chỗ theo điều khẳng định của một thành viên trong nhóm IBM, nơi đã tạo ra DES, kết luận rằng phương pháp thám mã vi sai đã được biết ngay khi tìm ra DES từ năm 1974.

# (tiếp)

- Để thám mã vi sai phiên bản LUCIFER với tám vòng mã hoá đòi hỏi tất cả  $2^{56}$  bản rõ lựa chọn.
- Để bẻ khoá DES với tám vòng mã hoá, đòi hỏi tất cả  $2^{14}$  bản rõ lựa chọn.

# (tiếp)

- Sử dụng phương pháp thám mã vi sai trong thực tế là không đơn giản chút nào.
- Việc miêu tả chi tiết phương pháp luận phù hợp có thể tìm thấy trong tài liệu của Biham.

# Mô tả

- Nghiên cứu một khối văn bản rõ m bit, bao gồm hai nửa khối là  $m_o$  và  $m_1$  bit.
- Trong mỗi vòng của thuật toán mã hoá DES, nửa bên phải của khối dữ liệu đầu vào được biến đổi thành nửa bên trái của khối dữ liệu đầu ra.

# (tiếp)

- Nửa bên phải của khối dữ liệu đầu ra là hàm của nửa khối dữ liệu bên trái dữ liệu đầu vào và phân khoá.
  - Như vậy, trong kết quả của bất kì vòng mã hoá nào chỉ tạo nên một khối mới 32 bit.

# (tiếp)

- Nếu kí hiệu mỗi khối mới qua  $m_i$  ( $2 \leq i \leq 17$ ), thì nhận được ở đầu ra của mỗi vòng mã hoá các khối khôi trung gian, sẽ liên hệ với nhau như sau:

$$m_{i+1} = m_{i-1} \oplus f(m_i, K_i), \quad i = 1, 2, \dots, 16.$$

# (tiếp)

- Thám mã vi sai đề xuất khảo sát hai bản tin  $m$  và  $m'$ , đối với chúng biết được sai phân XOR (XOR difference) là:  $m = m \text{ XOR } m'$ , và tiếp tục khảo sát sai phân giữa hai bản tin trung gian, chúng ta có kết quả sau:

$$\begin{aligned}\Delta m_{i+1} &= m_{i+1} \oplus m'_{i+1} = [m_{i-1} \oplus f(m_i, k_i)] \oplus [m'_{i-1} \oplus f(m'_i, k_i)] \\ &= \Delta m_{i-1} \oplus [f(m_i, k_i) \oplus f(m'_i, k_i)]\end{aligned}$$

# (tiếp)

- Giả thiết rằng, đối với nhiều cặp của dữ liệu đầu vào của hàm F có cùng sai phân sẽ sinh cùng một sai phân ở đầu ra, khi sử dụng cùng một phân khoá.
- Một cách hình thức sẽ nói rằng X kéo theo Y với xác suất p, nếu đối với một số p của các cặp, trong đó đầu vào với sai phân XOR là X, để cặp đầu ra phù hợp với sai phân XOR là Y.

# Nhận xét

- Mong muốn rằng có hàng loạt giá trị X với xác suất cao kéo theo sai phân xác định của giá trị đầu ra.
- Khi đó nếu đổi với chúng ta với xác suất lớn biết được các giá trị  $\Delta m_{i-1}$  và  $\Delta m_i$ , thì chúng ta với một xác suất lớn có thể xác định được cả  $\Delta m_{i+1}$ .
- Còn nếu chúng ta xác định được nhiều cặp giá trị sai phân như thế, thì chúng ta có thể xác định được phân khoá  $K_i$ , được sử dụng trong hàm F.

# Nhận xét

- Chiến lược chung của thám mã vi sai, được dựa trên phương pháp luận nêu trên cho mỗi mỗi vòng mã hóa.
- Thủ tục được bắt đầu bằng việc khảo sát hai văn bản rõ m' và m' với giá trị sai phân xác định đã cho và mẫu có thể có của sai phân sau mỗi vòng để sinh ra một sai phân có thể cho bản mã.

# (tiếp)

- Thực tế có 2 mẫu sai phân có thể cho 2 nửa 32 bit:  
 $(\Delta m_{17} || m_{16})$ .
- Sau đó thực hiện mã hóa m và m', để xác định hiệu thật sự khi không biết khoá, và sau đó so sánh kết quả với đánh giá xác suất được tính toán.

# (tiếp)

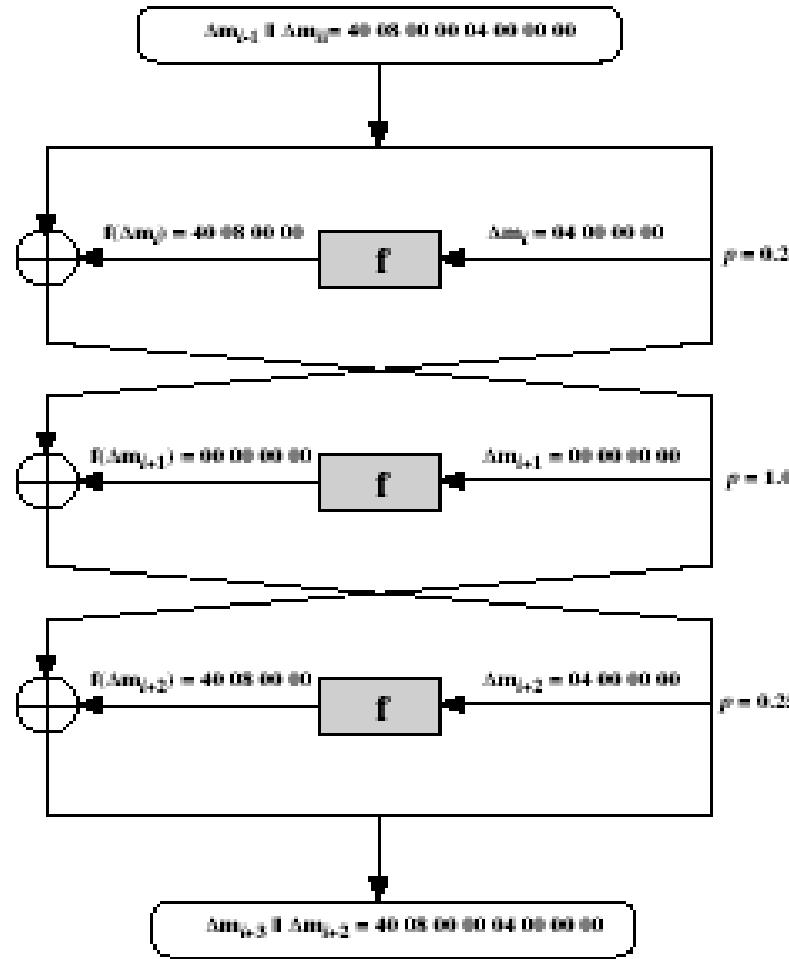
- Nếu kết quả trùng nhau, nghĩa là:

$$E_k(m) \oplus E_k(m') = (\Delta m_{17} || m_{16}).$$

thì có thể cho rằng các đánh giá xác suất đối với tất cả cả các vòng phù hợp với thực tiễn. Điều này cho phép suy ra một vài bit khoá.

Thủ tục được diễn ra theo cách lặp lại nhiều lần.

# Lan truyền vi sai qua 3 vòng của DES



# (tiếp)

- Sau 3 vòng, xác suất của sai phân lối ra là:
  - $0.25 \times 1 \times 0.25 = 0.0625$

# Thám mã tuyến tính

- Còn một hướng mới hơn trong thám mã là thám mã tuyến tính.
- Phương pháp thám mã tuyến tính có liên quan đến sự tìm kiếm xấp xỉ tuyến tính, để mô tả biến đổi thực hiện của DES.

# (tiếp)

- Phương pháp này cho phép tìm được khoá DES khi có  $2^{43}$  bản rõ biết, so sánh với thám mã vi sai cần  $2^{47}$  bản rõ chọn lựa.
- Là một sự cải tiến.
- Vì dễ dàng hơn đạt được biết bản rõ hơn là bản rõ chọn lựa.

# Nguyên tắc

- Giả sử đỗi với mã khối với n bit của khối căn bản rõ, cũng như khối văn bản mã và khối khoá m bit.
- Khối văn bản rõ sẽ kí hiệu là:  $P[1], \dots, P[n]$ , khối văn bản mã sẽ kí hiệu là:  $e[1], \dots, e[n]$ , còn khối khoá kí hiệu  $K[1], \dots, K[m]$ .

# (tiếp)

- Khi đó chúng ta xác định được :

$$A[i,j,\dots,k] = A[i] \oplus A[j] \oplus \dots \oplus A[k],$$

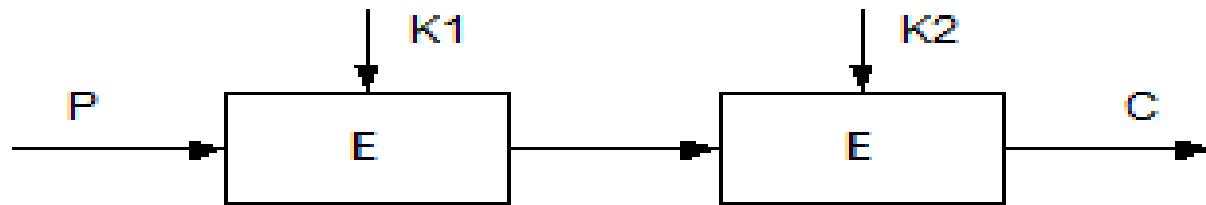
- Mục tiêu của phương pháp thám mã tuyến tính là tìm phương trình tuyến tính thích hợp dạng:

$$P[\alpha_1, \alpha_2, \dots, \alpha_a] \oplus C[\beta_1, \beta_2, \dots, \beta_b] = K[\gamma_1, \gamma_2, \dots, \gamma_c]$$

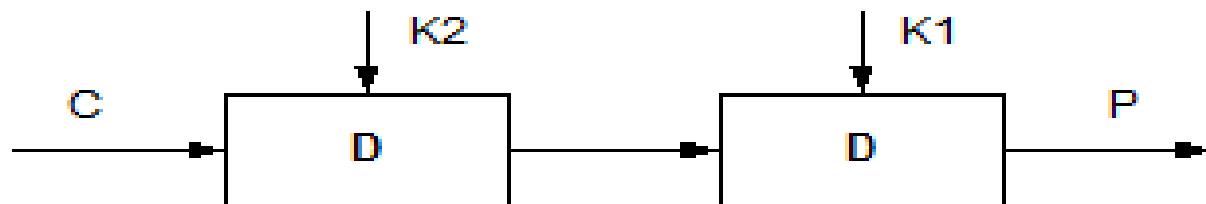
# Các biến dạng của DES

- DES “bội đôi”
- DES “bội ba” với 2 khóa
- DES “bội ba” với 3 khóa

# DES “bội đôi”



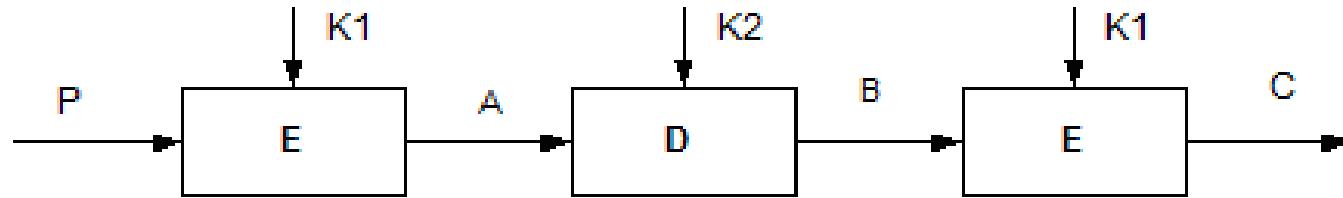
quá trình mã hóa "bội đôi"



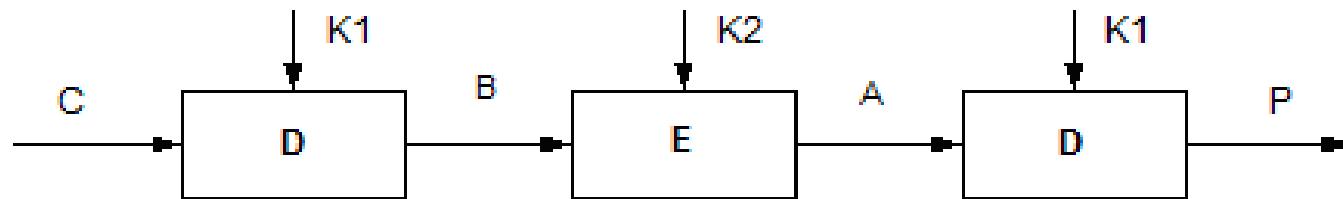
quá trình giải mã "bội đôi"

# DES “bội ba” với hai khoá

- Đề xuất của Tuchman



quá trình mã hóa DES "bội ba"



quá trình giải mã DES "bội ba"

# Các chế độ công tác của mật mã khối

- DES trong những trường hợp khác nhau, đã xác định bốn chế độ công tác (FIPS PUB 74, 81).
- AES thêm CTR.
- Chế độ không hồi tiếp (non feedback mode): ECB, CTR.
- Chế độ có hồi tiếp (feedback mode): CBC, CFB, OFB.

# Mô tả

Chế độ	Miêu tả	Lĩnh vực ứng dụng điển hình
Quyền sách mã điện tử (ECB-Electronic code book)	Mỗi một khối 64 bit của văn bản rõ được mã hoá không phụ thuộc lẫn nhau vào cùng một khoá	Bảo vệ việc truyền các giá trị riêng (chẳng hạn, khoá mã)
Sự liên kết các khối mã hoá(CBC-cipher block chaining)	Khối đầu vào số liệu đối với thuật toán mã hoá được tính toán như là tổng XOR của khối 64bit dạng của văn bản rõ và khối 64bit văn bản mã lượt trước	. Truyền số liệu ở dạng công dụng chung. . Xác thực

# (tiếp)

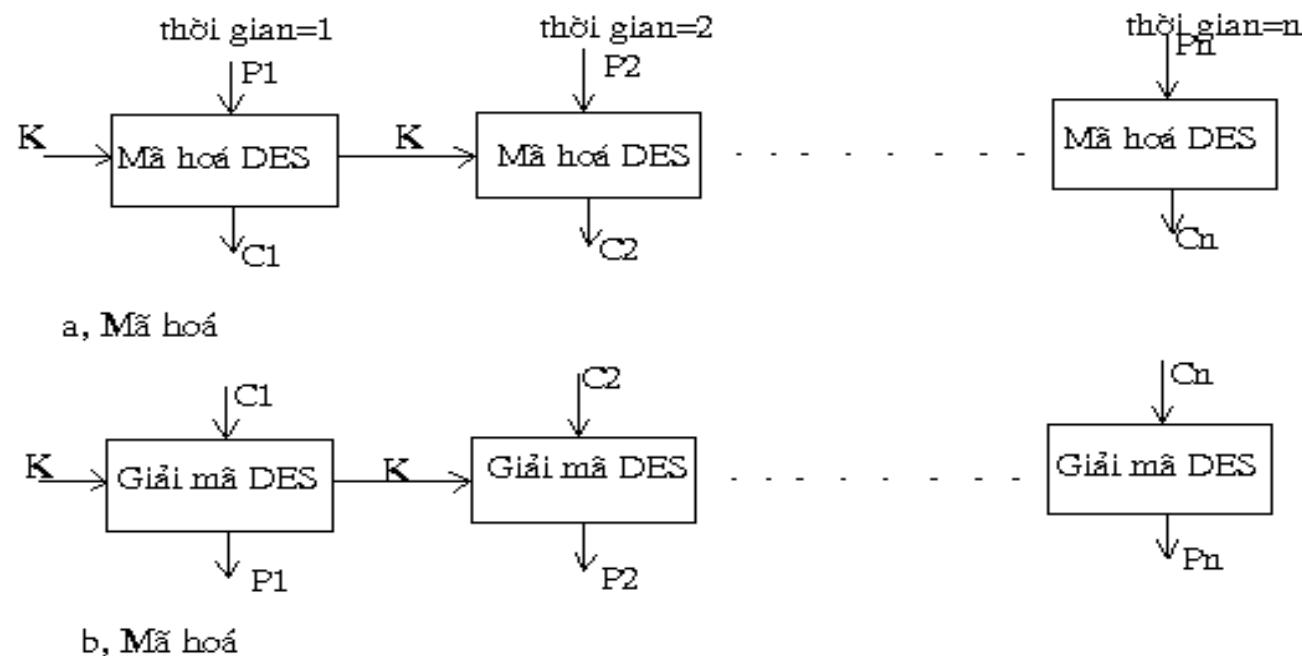
Liên hệ ngược mã hoá(CFB- cipher Feedback)	Các số liệu đầu vào được gia công từng phần theo jbit kết quả nhận được ở bước trước của văn bản mã, được sử dụng như là số liệu đầu vào đối với thuật toán mã hoá với mục đích nhận được sự	. Truyền liên tục số liệu theo công dụng chung. . Xác thực
--	--	---

	giả ngẫu nhiên liên tiếp, tổng XOR theo j bit với khối văn bản rõ sẽ xác định trình tự từng phần của văn bản mã.	
--	--	--

# (tiếp)

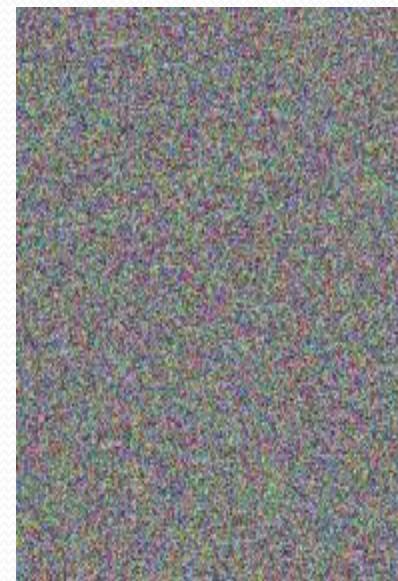
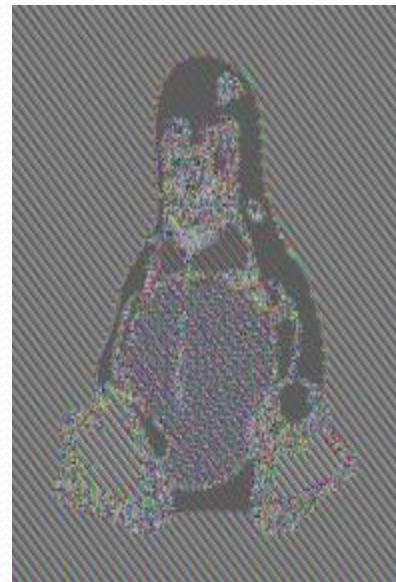
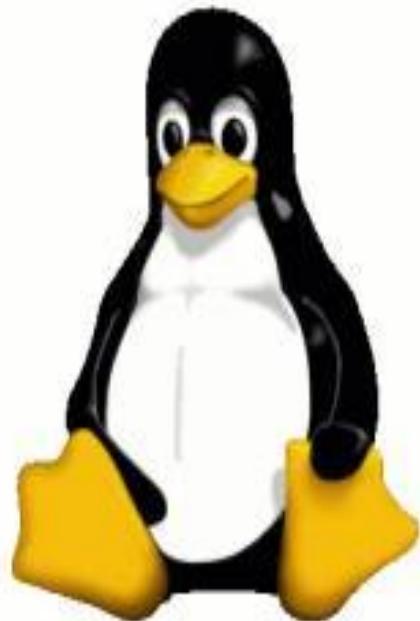
Liên hệ ngược đầu ra(OFB-output Feedback)	Tương tự như chế độ CFB, nhưng làm số liệu đầu vào đổi với thuật toán mã hoá được sử dụng các số đầu ra nhận được trước đây của DES	Truyền liên tục số liệu theo kênh có nhiều(chẳng hạn kênh liên lạc vệ tinh)
---	---	---

# ECB

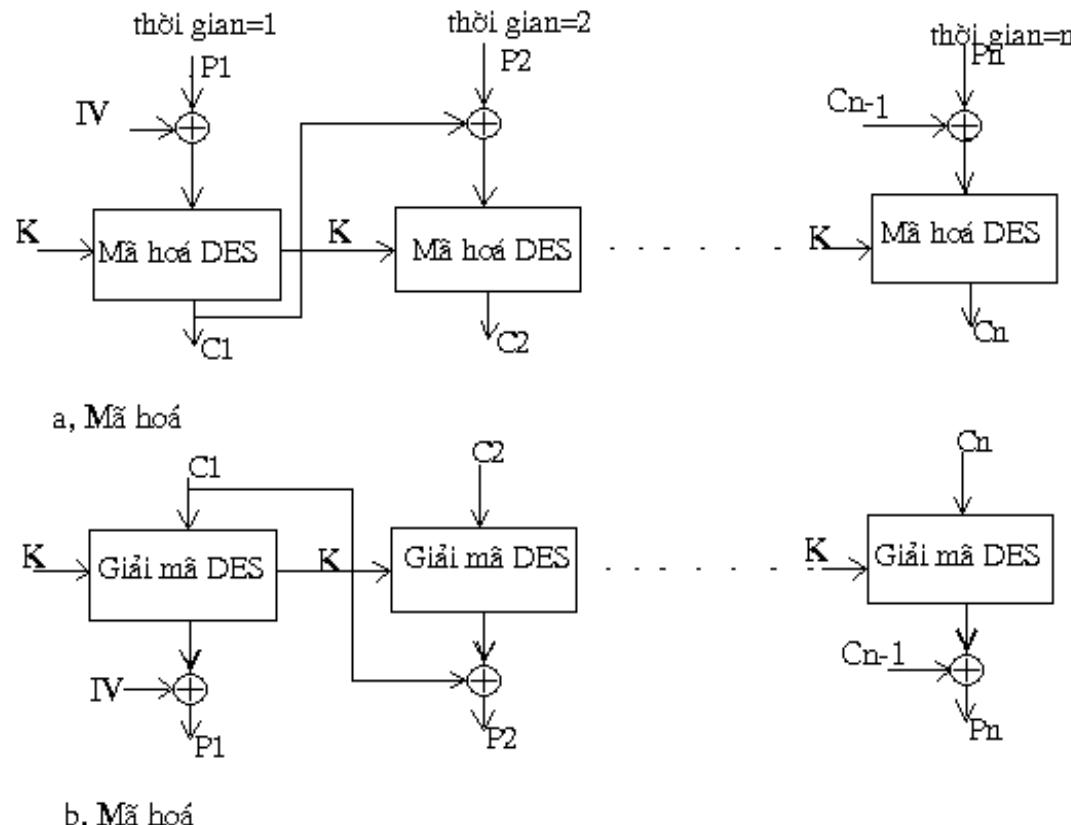


Hình 3.11. Chế độ quyển sách mã hoá điện tử

# So sánh các chế độ (bản rõ, ECB, các chế độ khác)

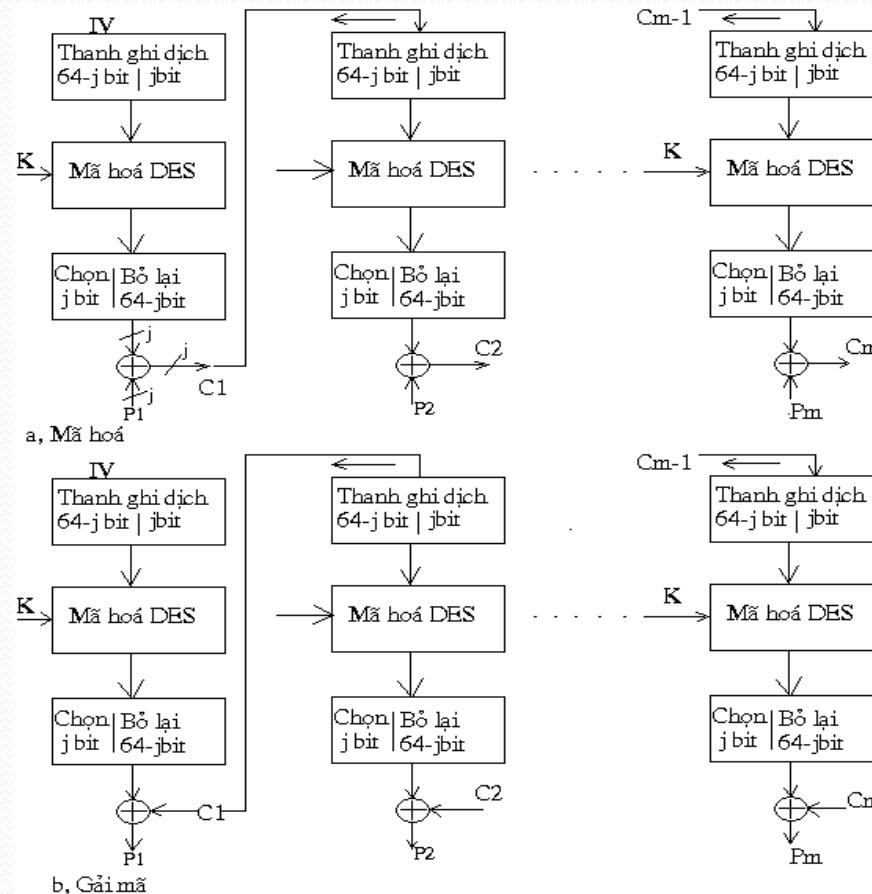


# Chế độ CBC



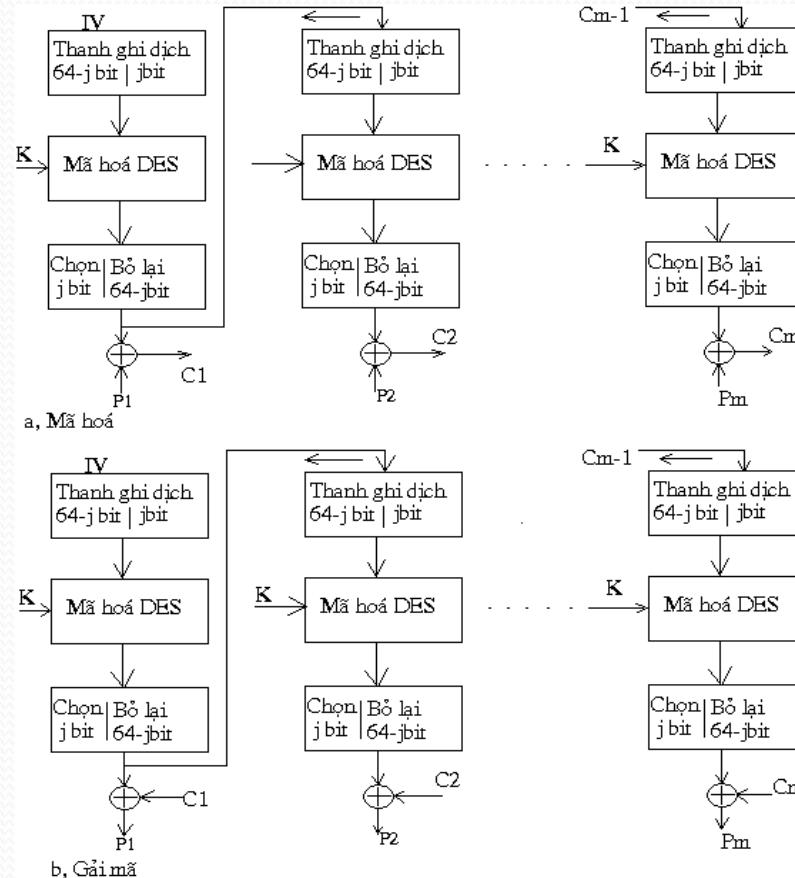
Hình 3.12.Chế độ xâu chuỗi các khối văn bản mã (CBC)

# Chế độ CFB



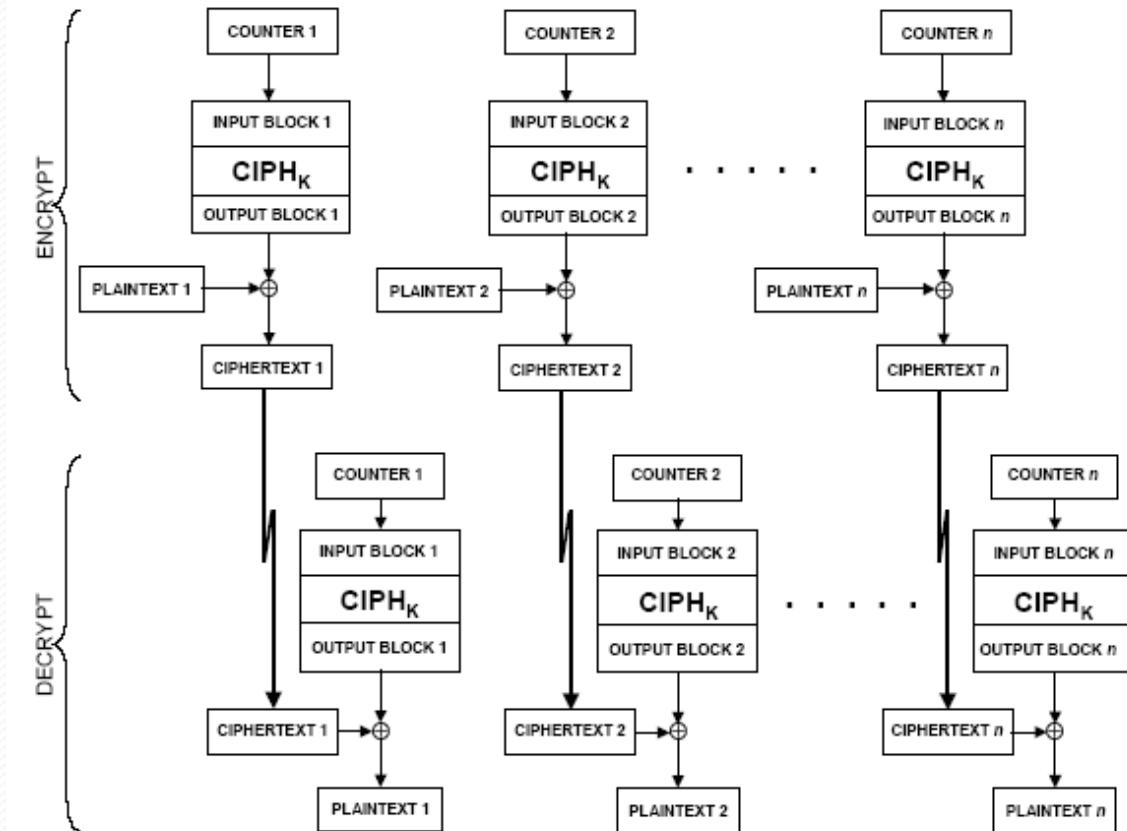
Hình 3.13. Chế độ liên hệ ngược mã hoá j bit (CFB)

# Chế độ OFB



Hình 3.14. Chế độ liên hệ ngược jbit theo đầu ra (OFB)

# Chế độ CTR (The Counter mode )



## 4. Thuật toán AES

- DES và các thuật toán cải tiến (DES-X, G-DES, D\_DES, T\_DES).
  - Các thuật toán phát triển thay thế IDEA, RC5, ...
- Không đáp ứng được các nhu cầu hiện tại và tương lai.

# (tiếp)

- Trước tình hình đó, NIST đã mở ra một cuộc thi nhằm tìm kiếm thuật toán mới thay thế cho DES (sẽ được gọi là AES – Advanced Encryption Standard).
- Các yêu cầu cơ bản đối với các thuật toán chung khảo AES là:
  - Có tốc độ nhanh hơn so với DES,
  - ít nhất có độ an toàn không kém T\_DES
  - và có khả năng thực hiện tối ưu trên cả phần cứng và phần mềm
  - khối dữ liệu có độ dài 128 bít, và có khả năng làm việc với các khóa có độ dài khác nhau – 128, 192, và 256 bít.

# Các thuật toán chung khảo

- Vòng 1: 15 thuật toán

CAST-256, CRYPTON, DEAL, DFC, E<sub>2</sub>, FROG, HPC,  
LOKI97, MAGENTA, MARS, RC6, Rijndael, SAFER+,  
Serpent, and Twofish.

- 5 thuật toán chung khảo:

MARS, RC6, Rijndael, Serpent, and Twofish.

# Các tiêu chuẩn đánh giá

- Ý tưởng thiết kế:
  - Dựa trên các thiết kế có sẵn?
  - Mã hoàn toàn mới?
  - Mạng Feistel/ S-P?
  - Thiết kế của các vòng mã hóa.
  - Sử dụng bao nhiêu vòng?
  - ...

# (tiếp)

- Các loại toán tử được sử dụng: XOR, kích thước các hộp S được sử dụng ( $4 \times 4$ , ...,  $8 \times 32$ ), ....
- Công nghệ thực hiện: Bitslicing, PHT, Decorrelation, ....
- Mục tiêu tối ưu: Xác định thuật toán sẽ được thiết kế để thực hiện tối ưu trên nền tảng nào – Pentium/MMX/Pro, 64/32/8 bit processors, smart card, phần cứng.
- Và một số tiêu chuẩn khác.

# Một số nhận xét

- **Rijndael**: được phát triển từ thuật toán Square,
- dựa trên cấu trúc cơ sở là mạng chuyển vị-thay thế (S-P Network),
- có thể thực hiện trên 10, 12, hoặc 14 vòng (phụ thuộc vào độ dài khối dữ liệu và khóa 128, 192 hoặc 256 bit),
- thực hiện hiệu quả trên cả phần cứng và phần mềm (dễ dàng thực hiện, có tốc độ làm việc cao, yêu cầu ít bộ nhớ).

# (tiếp)

- Với thuật toán Rijndael nguyên bản có thể sử dụng các khóa và các khối dữ liệu với kích thước bất kỳ là bội số của 32 bít (nhỏ nhất là 128 bít và lớn nhất là 256 bít).
- Các tính toán của AES được thực hiện trên trường hữu hạn.
- Trong chế độ mã hóa, mỗi vòng của AES (ngoại trừ vòng cuối cùng) được thực hiện qua 4 biến đổi : 1), SubBytes; 2), ShiftRows; 3), MixColumns; 4), AddRoundKey.

# (tiếp)

- Các tấn công trên AES đã cố gắng thực hiện trên mã với số vòng được rút bớt (ví dụ như trên 7 vòng với 128 bit khóa, 8 vòng với 192 bit khóa, 9 vòng với 256 bit khóa).
- **Ưu điểm** chính của AES là sự mềm dẻo, nhanh và thuật toán được thực hiện rất tinh xảo.
- **Nghi ngại** chính đối với nó là sự an toàn của thuật toán (liên quan đến số vòng thực hiện của nó) .

# MARS

- Không giống với phần lớn các thuật toán mã khối khác, IBM thiết kế MARS với một cấu trúc mới lạ, nó có cấu trúc bất đồng nhất (heterogeneous).
- Thuật toán sử dụng các khóa biến đổi giữa 128 và 448 bít (biến đổi trên 32 bít).
- Thuật toán bao gồm 32 vòng với 2 kiểu cấu trúc, được chia thành 8 phần thực hiện.

# (tiếp)

- Các thành phần cơ bản trong 1 vòng thường là các toán tử cộng số nguyên, cộng mod2 và dịch vòng. Vì vậy thuật toán có hiệu suất hoạt động rất cao trên hầu hết các nền (platform) thực hiện (có một số hạn chế khi thực hiện trên smart card).
- MARS khác với tất cả các thuật toán AES chung kết khác là không dựa trên các cấu trúc đã biết vì vậy độ an toàn của thuật toán rất khó ước lượng.

# (tiếp)

- Nói chung ưu điểm chính của MARS là nó rất bền vững, trong thuật toán có sử dụng nhiều các cơ chế “fail stop” hơn so với các thuật toán AES chung khác.
- Nhờ có cấu trúc bất đồng nhất và sự đa dạng của các toán tử bền vững, vì vậy thậm chí với các tấn công thành công trên một thành phần nào đó của thuật toán, cũng sẽ không dẫn tới 1 tấn công thành công trên toàn bộ thuật toán.
- Thực tế, với MARS số vòng tối thiểu sử dụng của nó là 20.

# RC6

- Được phát triển từ RC5, được xây dựng dựa trên cấu trúc cơ sở Feistel.
- Nó có thể được thực hiện trên các biến thể khác nhau (kích thước khối dữ liệu đầu vào, kích thước khóa, số lượng vòng – thuật toán gồm 20 vòng trong cuộc thi AES), vì thế thuật toán rất mềm dẻo đối với tất cả các cấp độ của độ an toàn và sự hiệu quả.

# (tiếp)

- Trên thực tế, RC6 được xem như 2 quá trình mã hóa RC5 song song.
- RC6 bao gồm 20 vòng, tác giả của thuật toán đã xác nhận rằng với 16 vòng, thuật toán có thể tấn công với độ phức tạp  $2^{119}$ .
- Thuật toán có thể ít phù hợp trên một số nền nào đó vì trong thuật toán có sử dụng các toán tử dịch vòng trên 32 bít và các phép nhân số nguyên, nhưng khi các toán tử này được hỗ trợ, RC6 sẽ thực hiện nhanh hơn so với tất cả các thuật toán AES chung kết khác.

# Serpent

- Được thiết kế dựa trên mạng chuyển vị-thay thế (S-P Network).
- Các tác giả thiết kế thuật toán này hướng tới việc tuân thủ dựa trên các thiết kế đã có và coi trọng tính an toàn của thuật toán hơn là tính mới lạ và tốc độ của thuật toán.
- Trong mỗi vòng của thuật toán bao gồm 8 hộp S dựa trên các hộp S của mã DES, nó được thiết kế cho phép tất cả các toán tử có thể thực hiện song song.

# (tiếp)

- Thuật toán bao gồm 32 vòng. Các tác giả của thuật toán khẳng định rằng 16 vòng đã đảm bảo độ an toàn của thuật toán (32 vòng sẽ đảm bảo khả năng chống lại các kiểu tấn công trong tương lai).
- Điều này dễ ràng tạo cho thuật toán một sự an toàn cần thiết (Serpent được nhìn nhận là thuật toán an toàn nhất trong các thuật toán chung khảo AES),

# (tiếp)

- Nhưng sự trả giá của nó là hiệu suất thấp của thuật toán so với tất cả các thuật toán chung khảo AES.
- Tuy nhiên, vì yêu cầu ít bộ nhớ khi thực hiện, vì vậy thuật toán rất thích hợp để thực hiện trên smart card (chính điều này giúp cho Serpent chiến thắng thuật toán CAST-256, mặc dù chúng có cùng hiệu năng và độ an toàn).

# Twofish

- Thiết kế dựa trên thuật toán Blowfish (được thiết kế dựa trên cấu trúc cơ sở là mạng Feistel).
- Twofish là thuật toán nhanh và không yêu cầu nhiều bộ nhớ khi thực hiện.

# (tiếp)

- Cấu trúc của thuật toán rất phức tạp và do đó rất khó phân tích (để đạt được sự mềm dẻo trong cấu trúc, các tác giả của thuật toán đã sử dụng một số “kỹ xảo - tricks”, sự an toàn của các kỹ thuật này không rõ ràng, vì vậy trong 5 thuật toán chung khảo AES, Twofish là thuật toán khó phân tích nhất), điều này làm cho nó giống như thuật toán MARS.
- Có ưu điểm hơn là thiết kế được dựa trên một thuật toán đã được nghiên cứu và đã phổ biến.

# (tiếp)

- Twofish bao gồm 16 vòng, tính năng đặc biệt của nó là sử dụng các hộp S được tính toán trước phụ thuộc vào khóa, và sử dụng khóa theo thời gian biểu tương đối phức tạp.
- Trong thiết kế, thuật toán sử dụng 1 số thành phần của các thiết kế khác - ví dụ biến đổi PHT (Pseudo-Hadamard Transform) của họ mã SAFER.

# (tiếp)

- Trên hầu hết các nền phần mềm Twofish chậm hơn một chút so với Rijndael (với 128 bít khóa), nhưng có phần nhanh hơn đối với 256 bít khóa.
- Các tác giả của thuật toán đã đề cập đến việc tấn công thuật toán trên 6 vòng và tấn công liên quan đến khóa lên tới 10 vòng.
- Trên thực tế (đến năm 2006) không có kiểu tấn công nào lên thuật toán hiệu quả hơn kiểu tấn công “tìm khóa theo phương pháp vét cạn – brute force key search”.

# Kết quả

- Rijndael: 86 positive, 10 negative
- Serpent: 59 positive, 7 negative
- Twofish: 31 positive, 21 negative
- RC6: 23 positive, 37 negative
- MARS: 13 positive, 83 negative

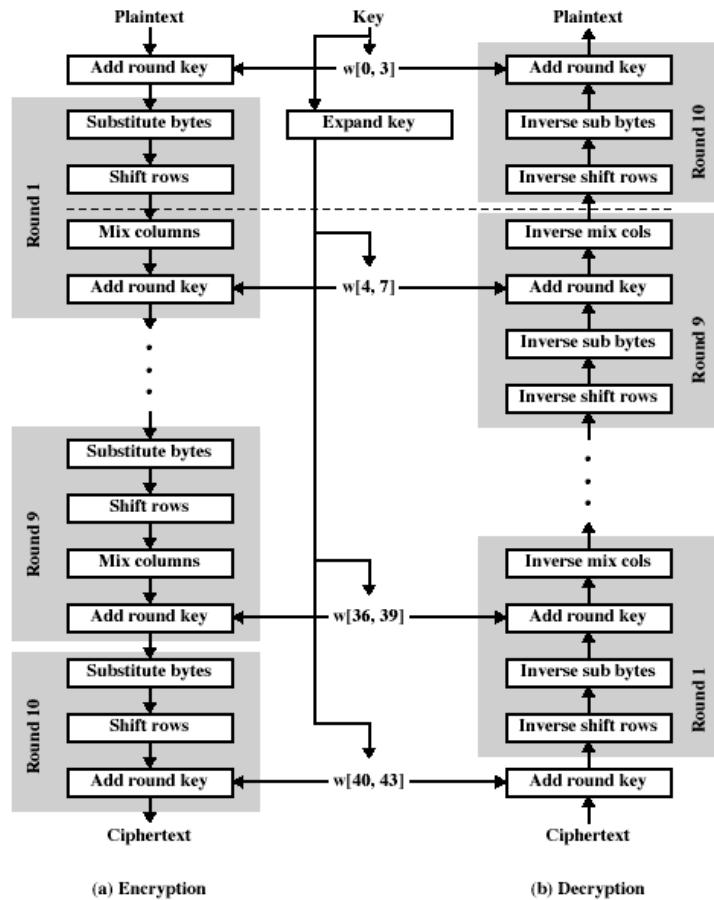
# Thuật toán AES

- Hầu hết các tính toán của AES được thực hiện trên trường hữu hạn.
- AES hoạt động trên các mảng của byte kích thước  $4 \times 4$ , được gọi là state (trạng thái).

# 4 phép biến đổi

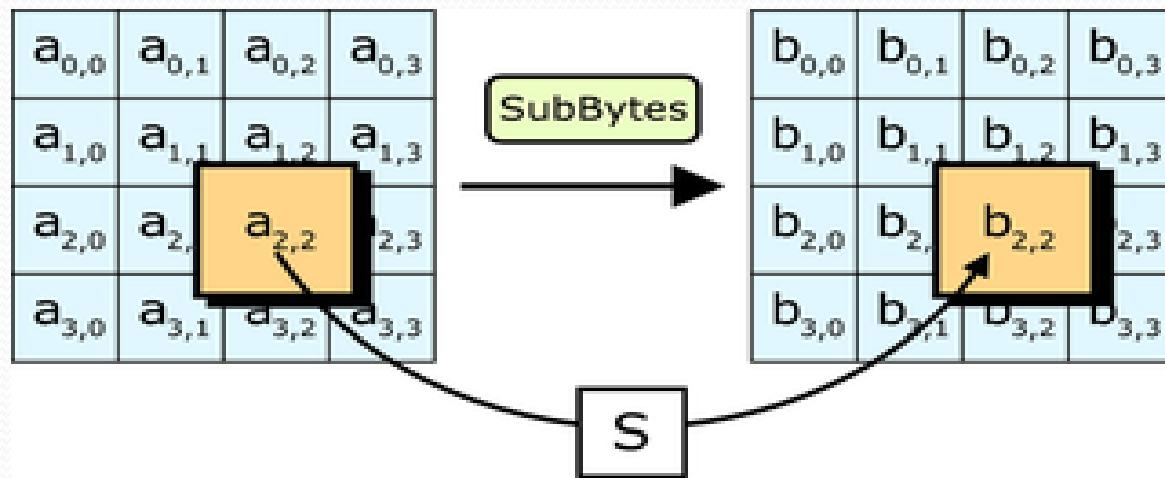
- AddRoundKey: cộng (XOR) khóa của chu kỳ vào trạng thái hiện hành. Độ dài của khóa của chu kỳ bằng với kích thước của trạng thái.
- SubBytes: thay thế phi tuyến mỗi byte trong trạng thái hiện hành thông qua bảng thay thế (S-box).
- MixColumns: trộn thông tin của từng cột trong trạng thái hiện hành. Mỗi cột được xử lý độc lập.
- ShiftRows: dịch chuyển xoay vòng từng dòng của trạng thái hiện hành với di số khác nhau.

# Sơ đồ AES



# SubBytes

- Trong bước SubBytes, mỗi byte trong state được thay thế bởi một byte của hộp S đóng vai trò là bảng tìm kiếm.

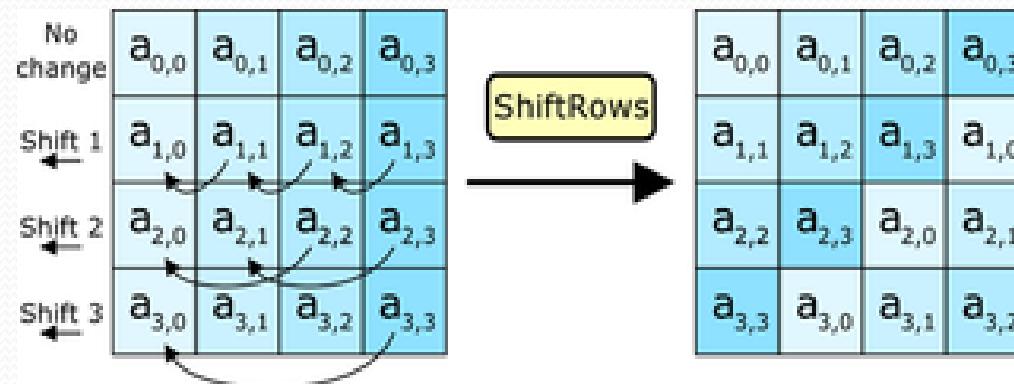


# Nhận xét

- Bước thay thế có tính phi tuyến.
- Hộp S được sinh ra từ hàm ngược (inverse function) trên  $GF(2^8)$  để có được tính chất phi tuyến cao nhất.

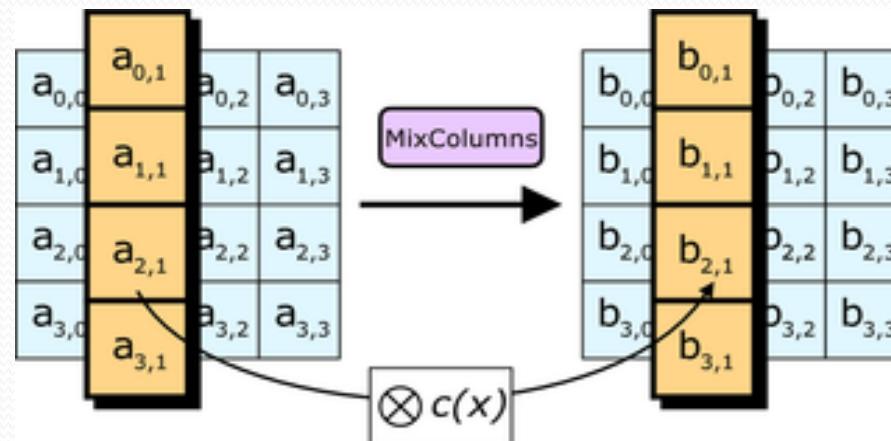
# ShiftRows

- Trong bước ShiftRows, các byte trong mỗi hàng của state được dịch vòng về phía trái. Số lượng các bước dịch vòng là khác nhau ở mỗi hàng.



# MixColumns

- Trong MixColumns, mỗi cột của state được nhân với một đa thức cố định  $c(x)$  theo  $\text{mod}(x^4 + 1)$ .

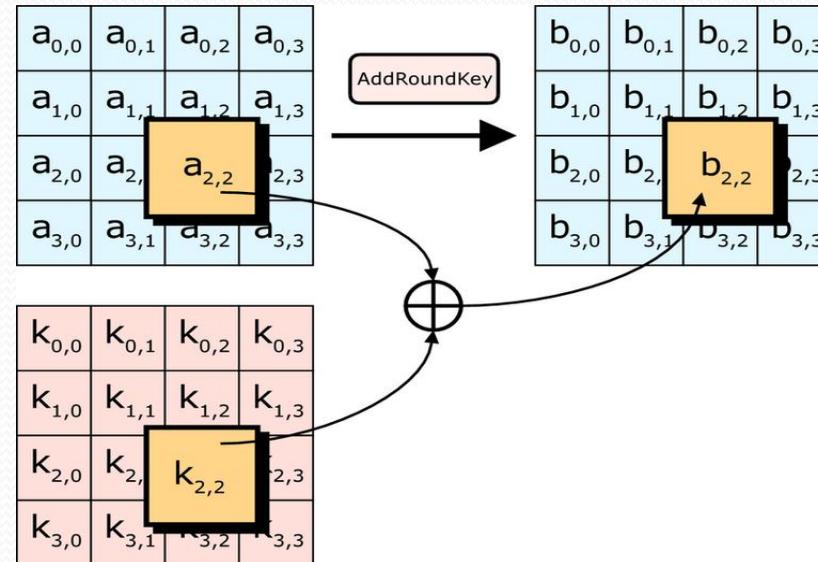


# Nhận xét

- Trong hàm MixColumns mỗi byte của lối vào sẽ ảnh hưởng lên tất cả 4 byte ở lối ra.
- Cùng với ShiftRows, MixColumns sẽ tạo ra tính khuếch tán (diffusion) của mã.
- Mỗi cột được xử lý như một đa thức trên  $GF(2^8)$  và được nhân với một đa thức cố định  $c(x) = '03'x^3 + '01'x^2 + '01'x + '02'$  theo modul  $(x^4 + 1)$ .

# AddRoundKey

- Mỗi byte của state được kết hợp tương ứng với 1 byte của khóa vòng bằng sử dụng toán tử XOR.



# Độ bền của thuật toán

- Thám mã vi sai: các tấn công vi sai là có thể nếu có thể dự đoán được sự sai khác lan truyền trên tất cả các vòng mà có tỉ số lan truyền lớn hơn đáng kể  $2^{1-n}$ , với n là độ dài của khối dữ liệu.
- Trong AES với 4 vòng đưa ra, tỉ số lan truyền khoảng  $2^{-150}$  và với 8 vòng là khoảng  $2^{-300}$ . Vì vậy nó đủ nhỏ để chống lại thám mã vi sai.

# (tiếp)

- Thám mã tuyển tính: với AES tương quan vào ra qua 4 vòng vào khoảng  $2^{-75}$  và qua 8 vòng là  $2^{-150}$ .
- Vì vậy nó đủ nhỏ để chống lại thám mã tuyển tính.

# Ưu điểm

- Phương diện thực hiện:
  - Tốc độ nhanh
  - Tiêu tốn ít tài nguyên
  - Vòng biến đổi song song
  - Không sử dụng các phép toán số học, nó không ảnh hưởng đến giới hạn trên hoặc dưới của cấu trúc bộ vi xử lý.

# (tiếp)

- Nó không sử dụng các thành phần của thuật toán mã khác.
- Tính bí mật của thuật toán không dựa trên sự không rõ ràng.
- Thiết kế kín của thuật toán không cho phép có đủ cơ hội để che giấu một cửa sập.

# Nhược điểm

- Phần giải mã ít phù hợp để thực hiện trên Smart card hơn phần mã hóa, nó yêu cầu nhiều chỉ thị và sổ chu kì thực hiện nhiều???.
- Trên phần mềm: mã và giải mã sử dụng các chỉ thị và/hoặc các bảng khác nhau.
- Trên phần cứng: giải mã chỉ sử dụng một phần mạch đã được thiết kế thực hiện cho mã hóa.