AN TOÀN & BẢO MẬT THÔNG TIN

Giảng viên: Ths Phạm Thanh Bình

Bộ môn Kỹ thuật máy tính & mạng

http://dhthuyloi.blogspot.com

Chương 2:

MẬT MÃ ĐỐI XỨNG

(tiếp)

- Những vấn đề cơ bản của mật mã
- Các kỹ thuật mã hoá cổ điển
- Chuẩn mã hoá dữ liệu DES
- Chuẩn mã hoá cải tiến AES

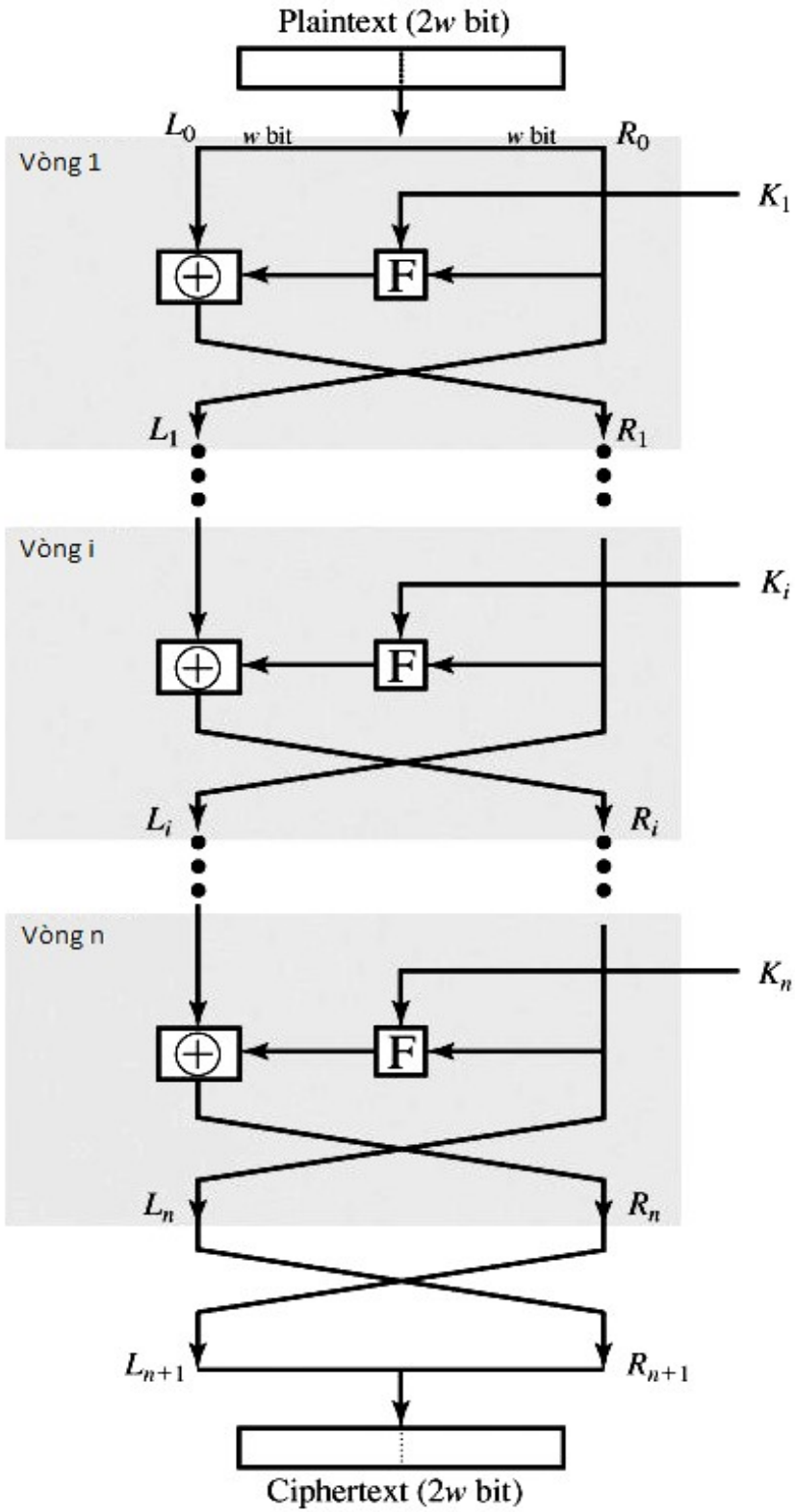
Mật mã khối và DES

- **Mật mã luồng** (Stream cipher) là loại mật mã lần lượt mã hóa từng bit hay từng byte dữ liệu số của một luồng dữ liệu.
- **Mật mã khối** (Block cipher) mã hóa mỗi lần cả một khối plaintext và thường kết xuất ra một khối ciphertext có chiều dài như khối plaintext.
- Độ dài tiêu biểu của một khối thường là 64 bit hay 128 bit.

Mật mã khối Feistel

- Mật mã Feistel được coi là cơ sở của các mật mã khối hiện đại
- Đầu vào là một khối plaintext dài m bít và khoá K dài k bít.
- Sau đó liên tục tác động lên plaintext bằng các kỹ thuật thay thế và hoán vị, lặp lại nhiều lần, để thu được ciphertext ở đầu ra cũng có chiều dài m bít.

- Kỹ thuật thay thế có tác dụng làm *xáo trộn* thông tin thống kê của plaintext, làm phức tạp hóa mối quan hệ thống kê giữa ciphertext và mật khoá, nhằm ngăn cản nỗ lực tìm khoá.
- Kỹ thuật hoán vị có tác dụng làm *khuếch tán* thông tin thống kê của plaintext, pha loãng các cấu trúc thống kê của plaintext ra một phạm vi rộng hơn, làm phức tạp hóa mối quan hệ thống kê giữa ciphertext và plaintext.
- Có thể lặp lại phép thay thế và hoán vị nhiều lần để tăng độ phức tạp, nâng cao tính bảo mật


Plaintext (2w bit)
L0
w bit
w bit
R0
Vòng 1
K1
F
L1
R1
Vòng i
Ki
F
Li
Ri
Vòng n
Kn
F
Ln
Rn
Ln+1
Rn+1
Ciphertext (2w bit)

- Plaintext có chiều dài $m = 2w$ bit, được chia thành hai nửa, L_0 và R_0.
- Hai nửa dữ liệu đi qua n vòng xử lý rồi được tổ hợp lại thành khối ciphertext.
- Mỗi vòng xử lý i có các đầu vào L_{i-1} và R_{i-1} lấy từ vòng trước cùng với khóa con K_i, lấy từ khóa K gốc.
- Các khóa con K_i đều khác nhau và khác khóa K.

Cấu trúc của 1 vòng xử lý:

- Đầu tiên, **phép thay thế** được thực hiện trên L_0, bằng cách XOR L_0 với $F(R_0,K_1)$.

  Hàm F là một phép biến đổi, càng phức tạp càng tốt
- Sau đó thực hiện **phép hoán vị** bằng cách hoán đổi giá trị của hai nửa dữ liệu
- Kết quả thu được: $R_1 = L_0 \oplus F(R_0,K_1)$

  $L_1 = R_0$
- Lấy L_1, R_1 làm đầu vào của vòng thứ 2, lặp lại *n* vòng.

- Công thức tổng quát của mỗi vòng:

$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$$

$$L_i = R_{i-1}$$

- **Kích thước khối (*m*)**: Kích thước khối càng lớn càng bảo mật tốt, nhưng làm giảm tốc độ của thuật toán. Thường thì $m = 64$ bít.
- **Kích thước khóa (*k*)**: Kích thước khóa càng lớn càng bảo mật tốt, nhưng làm giảm tốc độ của thuật toán. Trước đây DES sử dụng $k=56$ bít, nhưng hiện không còn an toàn nữa. Ngày nay $k = 128$ bít là phổ biến.

Thời gian trung bình để dò tìm khóa theo thuật toán brute-force

Kích thước khóa (bit)	Số lượng khóa tối đa	Thời gian cần cho máy 1 phép giải mã /μs	Thời gian cần cho máy 10^6 phép giải mã /μs
32	$2^{32}=4{,}3 \times 10^9$	$2^{31}\mu s = 35{,}8$ phút	2,15 milli giây
56	$2^{56}=7{,}2 \times 10^{16}$	$2^{55}\mu s = 1142$ năm	10,01 giờ
128	$2^{128}= 3{,}4 \times 10^{38}$	$2^{127}\mu s = 5{,}4 \times 10^{24}$ năm	$5{,}4 \times 10^{18}$ năm
168	$2^{168}= 3{,}7 \times 10^{50}$	$2^{167}\mu s = 5{,}9 \times 10^{36}$ năm	$5{,}9 \times 10^{30}$ năm
26 ký tự (hoán vị)	$26! = 4 \times 10^{26}$	$2 \times 10^{26}\mu s = 6{,}4 \times 10^{12}$ năm	$6{,}4 \times 10^6$ năm

- **Số vòng xử lý (*n*)**: Một vòng đơn thì tính bảo mật không cao, nhưng nhiều vòng kết hợp sẽ làm tăng khả năng an ninh. Số vòng thường là 16.
- **Thuật toán sinh khóa con**: Thuật toán càng phức tạp thì càng khó bị phân tích phá mã.
- **Hàm F**: Hàm càng phức tạp thì càng khó bị phân tích phá mã.

Bài tập 1:

Lập trình mô phỏng hoạt động của mật mã Feistel đơn giản với 2 vòng xử lý:

- Nhập một khối plaintext từ bàn phím, khối có độ dài $m = 2w = 16$ bít.
- Nhập khóa K (dài 8 bít) từ bàn phím. Khóa K_i được sinh ra từ khóa K nhờ phép dịch trái K i lần.
- Hàm F thực hiện phép cộng giữa R_{i-1} với K_i.
- Hiện khối ciphertext ra màn hình.

Bài tập 2:

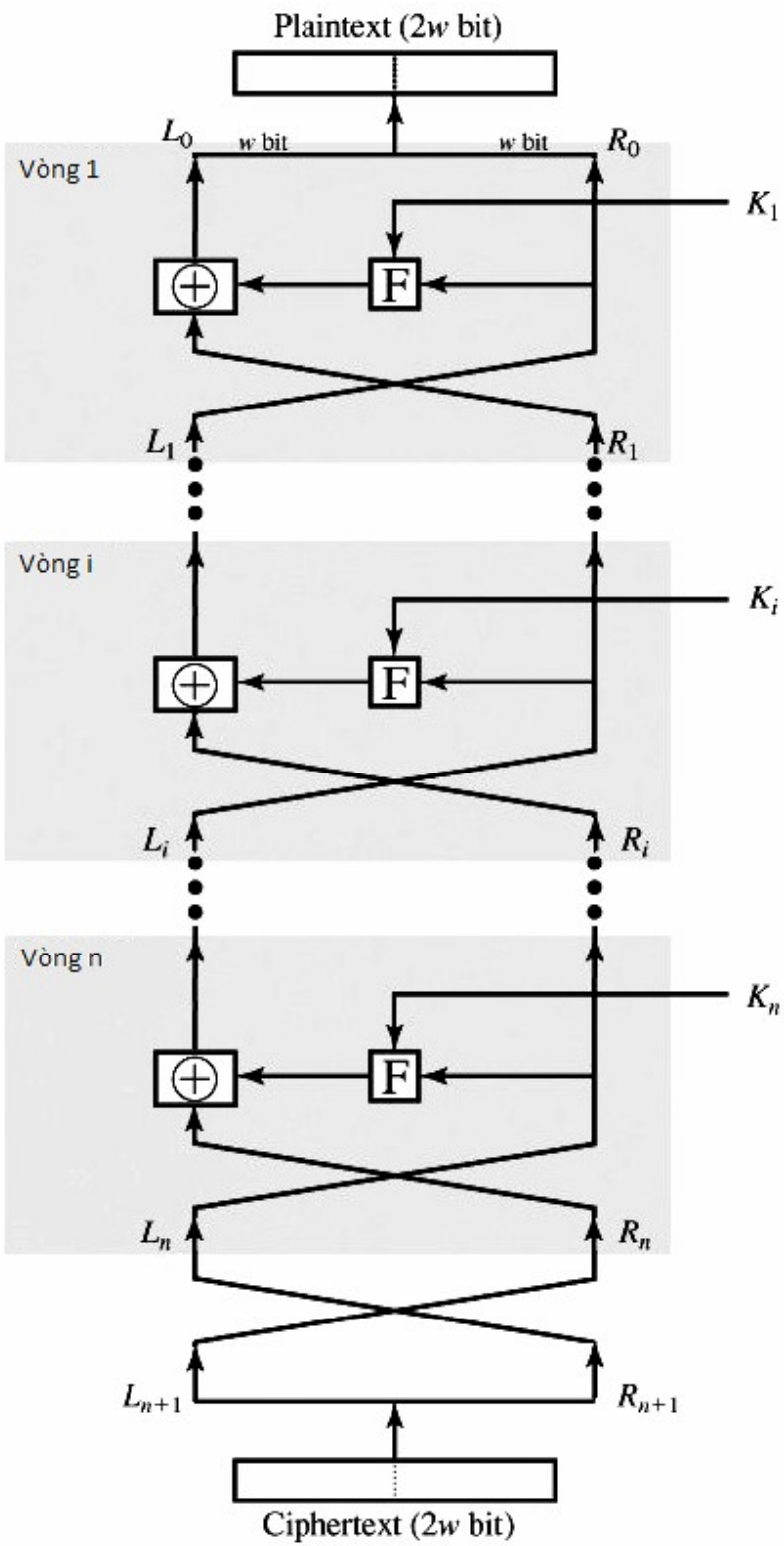
Lập trình mô phỏng hoạt động của mật mã Feistel đơn giản với 2 vòng xử lý:

- Nhập chuỗi plaintext từ bàn phím, chia chuỗi thành các khối dài $m = 2w = 16$ bít.
- Nhập khóa K (dài 8 bít) từ bàn phím. Khóa K_i được sinh ra từ khóa K nhờ phép dịch trái K i lần.
- Hàm F thực hiện phép cộng giữa R_{i-1} với K_i.
- Hiện chuỗi ciphertext ra màn hình.

Bài tập 3:

- Lập trình giải mã chuỗi ciphertext thu được từ bài tập 2.

(xem hướng dẫn ở trang sau)


Plaintext (2w bit)
L_0
w bit
w bit
R_0
Vòng 1
K_1
F
L_1
R_1
Vòng i
K_i
F
L_i
R_i
Vòng n
K_n
F
L_n
R_n
L_{n+1}
R_{n+1}
Ciphertext (2w bit)

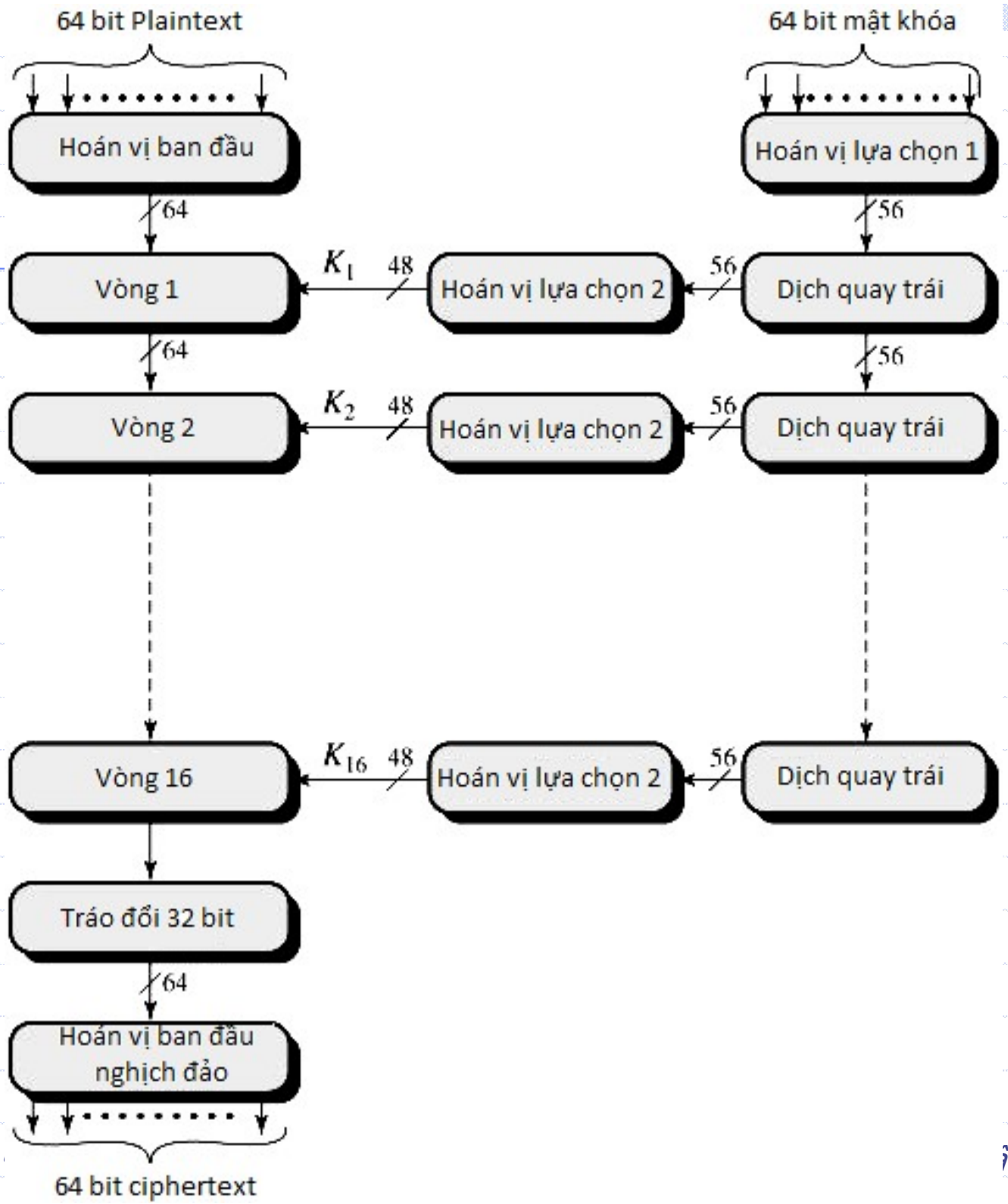
- Công thức tổng quát của mỗi vòng:

$$L_{i-1} = R_i \oplus F(L_i, K_i)$$

$$R_{i-1} = L_i$$

Chuẩn mã hoá dữ liệu DES

- Chuẩn mã hoá dữ liệu DES (Data Encryption Standard) được thông qua năm 1977 bởi National Bureau of Standards (NBS).
- DES sử dụng cấu trúc Feistel với độ dài khối $m = 64$ bít, và độ dài khóa $k = 56$ bít
- DES được sử dụng rộng rãi trong một thời gian dài (hơn 20 năm), cho tới khi nó bị bẻ gãy bởi tấn công brute-force năm 1998.


64 bit Plaintext
64 bit mật khóa
Hoán vị ban đầu
Hoán vị lựa chọn 1
64
56
Vòng 1
K_1
48
Hoán vị lựa chọn 2
56
Dịch quay trái
64
56
Vòng 2
K_2
48
Hoán vị lựa chọn 2
56
Dịch quay trái
Vòng 16
K_{16}
48
Hoán vị lựa chọn 2
56
Dịch quay trái
Tráo đổi 32 bit
64
Hoán vị ban đầu nghịch đảo
64 bit ciphertext

Chuẩn mã hoá cải tiến AES

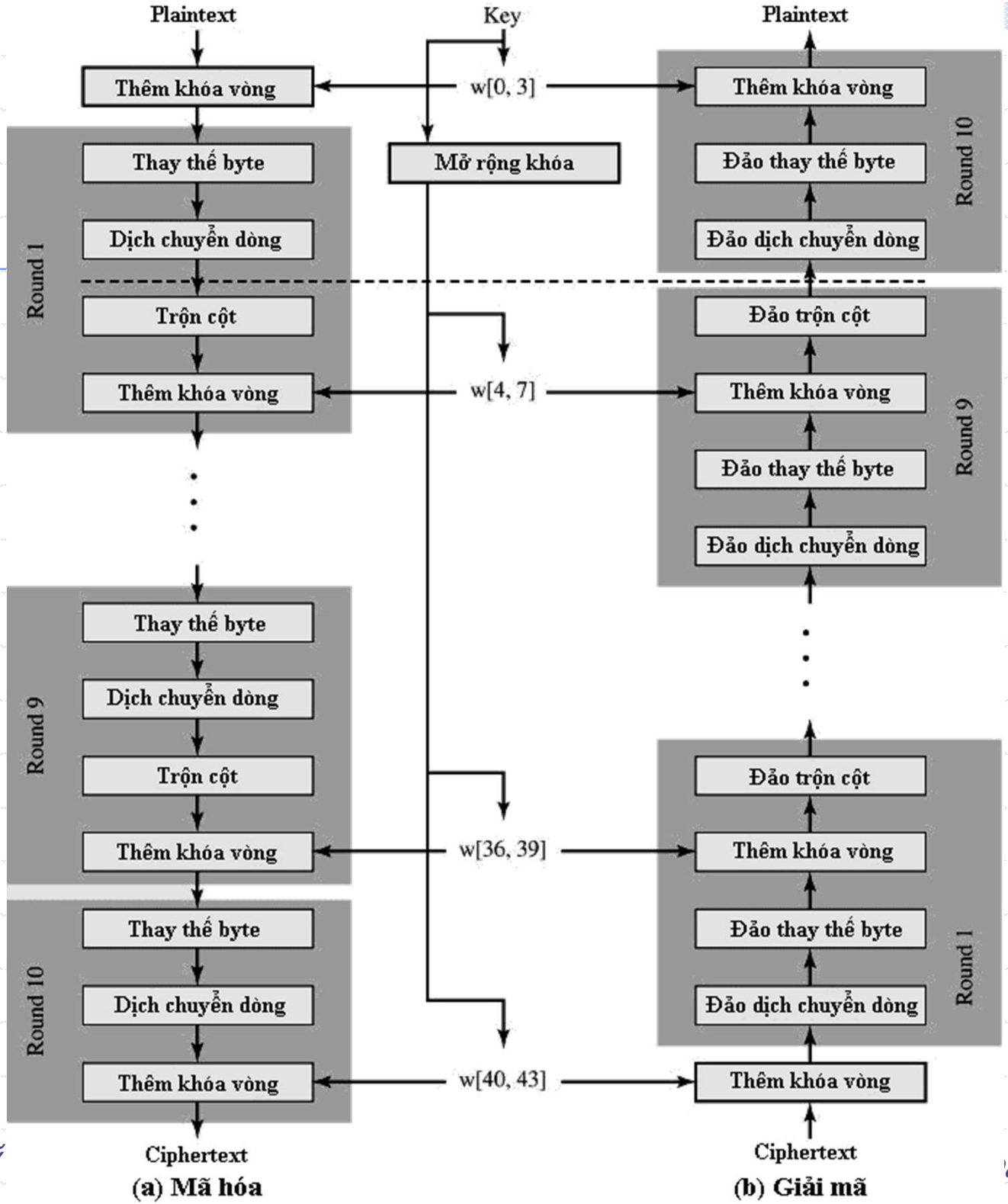
- Chuẩn mã hoá cải tiến AES (*Advanced Encryption Standard*) được ban hành bởi NIST (*National Institute of Standards and Technology*) vào năm 2001.
- AES là mật mã khối để thay thế cho DES trong các ứng dụng thương mại. Nó sử dụng kích thước khối 128 bit và kích thước mật khoá 128, 192 hay 256 bit

Các tham số của AES

Kích thước khoá (word/byte/bit)	4/16/128	6/24/192	8/32/256
Kích thước khối plaintext (word/byte/bit)	4/16/128	4/16/128	4/16/128
Số vòng	10	12	14
Kích thước khoá mỗi vòng (word/byte/bit)	4/16/128	4/16/128	4/16/128
Kích thước khoá mở rộng (word/byte)	44/176	52/208	60/240

AES không mang cấu trúc Feistel. Mỗi vòng của nó chứa bốn hàm phân biệt:

- Thay thế byte
- Hoán vị
- Các phép số học trên một trường hữu hạn
- XOR với mật khoá


Plaintext
Thêm khóa vòng
Thay thế byte
Dịch chuyển dòng
Trộn cột
Thêm khóa vòng
Round 1
Thay thế byte
Dịch chuyển dòng
Trộn cột
Thêm khóa vòng
Round 9
Thay thế byte
Dịch chuyển dòng
Thêm khóa vòng
Round 10
Ciphertext
(a) Mã hóa
Key
Mở rộng khóa
w[0, 3]
w[4, 7]
w[36, 39]
w[40, 43]
Plaintext
Thêm khóa vòng
Đảo thay thế byte
Đảo dịch chuyển dòng
Round 10
Đảo trộn cột
Thêm khóa vòng
Đảo thay thế byte
Đảo dịch chuyển dòng
Round 9
Đảo trộn cột
Thêm khóa vòng
Đảo thay thế byte
Đảo dịch chuyển dòng
Round 1
Thêm khóa vòng
Ciphertext
(b) Giải mã

Giải thích:

- **Thay thế byte**: Thực hiện một phép thay thế từng byte cho khối.
- **Dịch chuyển hàng**: Chỉ là một hoán vị đơn giản.
- **Trộn cột**: Một phép thay thế sử dụng số học
- **Thêm khoá vòng**: Một phép XOR bít đơn giản của khối hiện tại với một phần của khoá mở rộng.

Các đặc trưng của AES

- Có khả năng chống lại tất cả các tấn công đã được biết đến.
- Đảm bảo tốc độ thực hiện trên nhiều nền tảng kiến trúc.
- Có thiết kế đơn giản

Hết Phần 2_3