CHƯƠNG 3: MÃ HÓA ĐỐI XỨNG HIỆN ĐẠI

Mục đích: *Giới thiệu chung về các hệ mã hóa đối xứng hiện đại: nguyên tắc mã hóa, giải mã. Đưa các ví dụ minh họa để sinh viên nắm được các hệ mã hóa.*

Yêu cầu: *Sinh viên nắm được nguyên tắc mã hóa, giải mã của các hệ mã hóa hiện đại. Vận dụng được vào thực hiện các bài tập mã hóa.*

3.1 Mã khối hiện đại

Bây giờ chúng ta xét các mã khối hiện đại. Đây là kiểu mã được sử dụng rộng rãi nhất của các thuật toán mã hoá. Đồng thời nó cũng được sử dụng kết hợp với các thủ tục khác nhằm cung cấp các dịch vụ an toàn và xác thực.

Trước hết chúng ta tập trung vào chuẩn mã dữ liệu DES (Data Encryption Standards) để minh hoạ cho các nguyên lý mã khối. Trước hết chúng ta xét hai kiểu xử lý thông tin khác nhau trên bản rõ. Một kiểu chia dữ liệu thành từng khối để xử lý, kiểu kia xử lý trực tiếp từng đơn vị thông tin.

3.1.1 Phân biệt mã khối với mã dòng.

- Mã khối (block) xử lý bản tin theo từng khối, lần lượt mỗi khối được mã hoặc giải mã. Có thể xem giống như phép thế với các ký tự lớn – mỗi khối gồm 64 bít hoặc nhiều hơn.
- Mã dòng xử lý bản tin theo từng bít hoặc bite, lần lượt mỗi bít hoặc bite được mã hoá hoặc giải mã. Chẳng hạn như mã khoá tự động Vigenere.
- Rất nhiều mã hiện nay là mã khối. Chúng có khả năng ứng dụng rộng rãi hơn. Rất nhiều ứng dụng mã đối xứng trên mạng sử dụng mã khối.

Các nguyên lý mã khối

- Hầu hết các mã khối đối xứng dựa trên cấu trúc mã Fiestel, do nhà bac học Fiestel đề xuất năm 1973. Đây là điều cần thiết, vì cần phải có khả năng giải mã các bản mã một cách có hiệu quả.

- Mã khối được coi giống như phép thế cực lớn. Cần bảng có 2^{64} đầu vào cho mã khối 64 bít, bảng như vậy là rất lớn. Do đó có thể thay thế bằng cách tạo các khối nhỏ hơn.
- Sử dụng ý tưởng dùng mã tích. Ở đây sẽ kết hợp giữa mã thay thế và mã hoán vị, đồng thời sử dụng nhiều vòng lặp như vậy.

3.1.2 Claude Shannon và mã phép thế hoán vị

Năm 1949, Shannon đưa ra ý tưởng mạng phép thế và hoán vị (S-P networks) – là mã tích phép thế và hoán vị hiện đại với mục đích là cản trở việc thám mã dựa vào các phân tích thống kê. Giả sử kẻ thám mã biết một số tính chất thống kê của bản rõ như bảng phân bố tần suất của các chữ cái, bộ các chữ cái. Nếu các đặc trưng thống kê này được phản ánh trong bản mã, thì kẻ thám mã sẽ tìm cách tìm được khoá hoặc một phần khoá hoặc tìm mò ra bản rõ. Shannon muốn có một bản mã lý tưởng, ở đó mọi đặc trưng thống kế đều độc lập với khoá riêng được dùng, như vậy kẻ thám mã sẽ không có cơ sở để tìm khoá.

Mạng S-P đã tạo nên cơ sở cho mã khối hiện đại. Mạng S-P dựa trên hai thao tác mã cơ bản mà ta đã biết: phép thế (S-box) và hoán vị (P-box). Chúng sẽ tạo nên độ rối loạn và khuếch tán của bản tin.

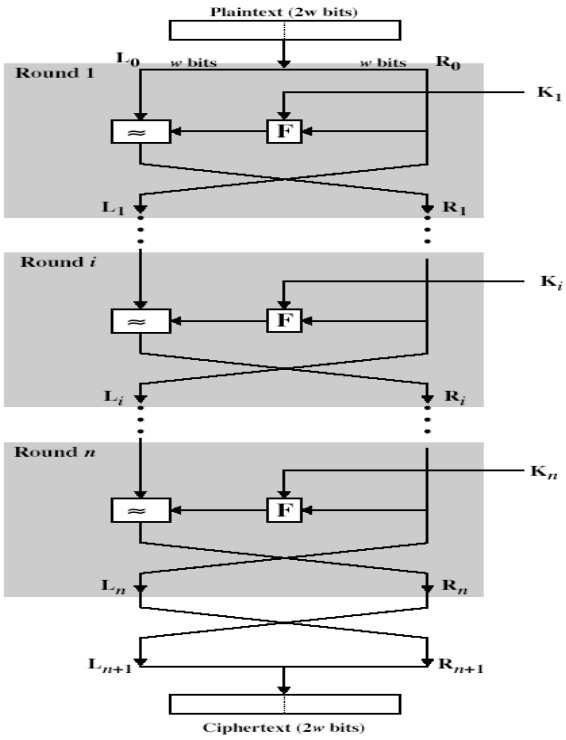
Rối loạn và khuếch tán

- Một tính chất quan trọng của mã tốt là mã cần phải che dấu hoàn toàn các tính chất thống kê của bản tin gốc. Như ta đã thấy mã bộ đệm một lần có thể làm được điều đó, do tính ngẫu nhiên của khoá đệm và độ dài bằng bản tin của nó.
- Shannon nghiên cứu và đề xuất phương pháp thực tế hơn là kết hợp các thành phần khác nhau của bản rõ để xử lý qua nhiều lần và nhận được bản mã.
- **Khuếch tán** là làm tan biến cấu trúc thống kê của bản rõ trên bản mã. Điều đó đạt được nếu mỗi bit của bản rõ tác động đến giá trị của rất nhiều bit trên bản mã hay mỗi bit của bản mã chịu tác động của nhiều bit bản rõ.

- **Rối loạn** là làm cho quan hệ giữa bản mã và khoá càng phức tạp càng tốt. Bản mã có tính rối loạn cao sẽ làm cho việc tìm mò khoá trở nên rất khó khăn, ngay cả khi kẻ tấn công có các đặc trưng thống kê của bản mã và biết cách khoá tác động đến bản mã.

3.1.3 Cấu trúc mã Fiestel

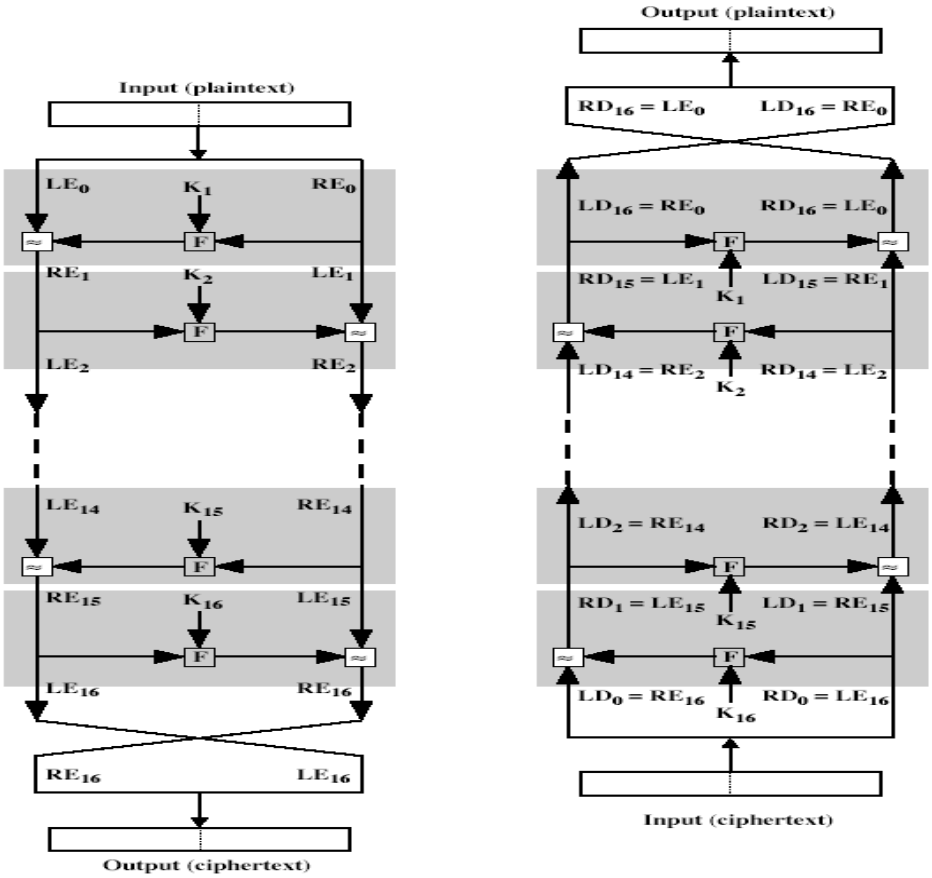
- Horst Fiestel sáng tạo nên mã Fiestel dựa trên mã tích nghịch đảo được, tức là kết hợp mã thế với mã hoán vị và qui trình giải mã là giống với mã hoá, chỉ cần thay đổi vai trò khối bản mã với khối bản rõ và thứ tự các khoá con được dùng. Từ khoá chính sinh ra cho mỗi vòng lặp một khoá con.
- Chia khối đầu vào thành 2 nửa bằng nhau:
  - Thực hiện phép thế trên nửa trái. Sử dụng hàm vòng trên nửa phải và khoá con, rồi tác động đến nửa trái.
  - Sau đó hoán vị các nửa, nửa phải chưa được xử lý.
  - Xử lý vòng tiếp theo.
- Đây là một thể hiện của mã thế kết hợp với hoán vị của Shannon. Ta xem xét cụ thể cấu trúc mã Fiestel gồm n vòng:



Nguyên tắc thiết kế mã khối Fiestel

- Tăng kích thước khối sẽ làm tăng độ an toàn nhưng làm giảm tốc độ mã
- Tăng kích thước khoá sẽ làm tăng độ an toàn – tìm khoá khó hơn, nhưng làm chậm mã.
- Tăng số vòng làm tăng độ an toàn nhưng làm chậm mã
- Phát sinh khoá con càng phức tạp làm cho việc thám mã khó hơn nhưng làm chậm mã
- Hàm vòng càng phức tạp làm cho việc thám mã khó hơn nhưng làm chậm mã
- Phần mềm mã hoá/giải mã nhanh và khó thám mã là tiêu chí hay được đề cập đến đối với ứng dụng và kiểm nghiệm thực tế.

Giải mã khối Fiestel



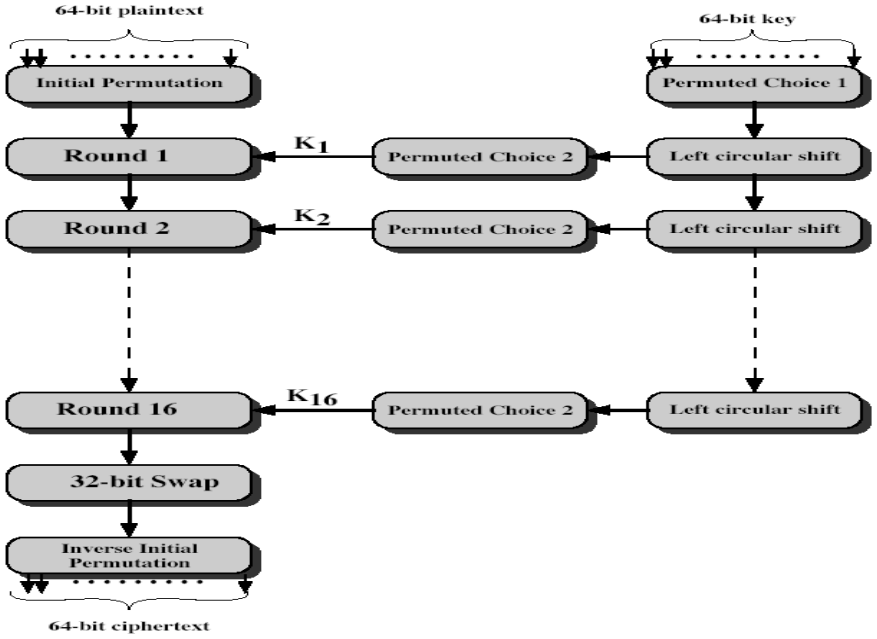
3.2 Chuẩn mã dữ liệu (DES)

DES (Data Encryption Standards) là mã khối sử dụng rộng rãi nhất trên thế giới trong thời gian vừa qua. Nó được đưa ra năm 1977 bởi NBS – văn phòng chuẩn Quốc gia Hoa kỳ (bây giờ là NIST - Viện chuẩn và công nghệ Quốc gia). DES là mã khối với mỗi khối dữ liệu 64 bít và dùng khoá dài 56 bít. Nó được sử dụng rộng rãi và đã được tranh luận kỹ về mặt an toàn.

3.2.1 Lịch sử DES:

Cuối những năm 1960, IBM phát triển mã Lucifer, được lãnh đạo bởi Fiestel. Ban đầu Lucifer sử dụng khối dữ liệu 64 bít và khoá 128 bít. Sau đó tiếp tục phát triển như mã thương mại. Năm 1973 NBS yêu cầu đề xuất chuẩn mã Quốc gia. IBM đề nghị bản sửa đổi Lucifer, sau này gọi là DES. Đã có các tranh luận về thiết kế của DES. Vì chuẩn của DES được công khai, mọi người đóng góp ý kiến về tốc độ, độ dài khoá và mức độ an toàn, khả năng thám mã. Người ta đề xuất chọn khoá 56 bít thay vì 128 để tăng tốc đọ xử lý và đưa ra các tiêu chuẩn thiết kế một chuẩn mã dữ liệu. Các suy luận và phân tích chứng tỏ rằng thiết kế như vậy là phù hợp. Do đó DES được sử dụng rộng rãi, đặc biệt trong lĩnh vực tài chính.

3.2.2 Sơ đồ mã DES



- **Hoán vị ban đầu IP**: đây là bước đầu tiên của tính toán dữ liệu, hoán vị IP đảo thứ tự các bít đầu vào: các bít chẵn sang nửa trái và các bít lẻ sang nửa phải. Hoán vị trên dễ dàng thực hiện trên phần cứng.

  Mỗi số trong hệ 16 biểu diễn bởi 4 bit, 16 số được thể hiện bởi 64 bit. Mỗi bit có một vị trí xác định qua hoán vị ban đầu (xem bảng phụ lục cuối tài liệu).

Ví dụ

IP(675a6967 5e5a6b5a) = (ffb2194d 004df6fb)

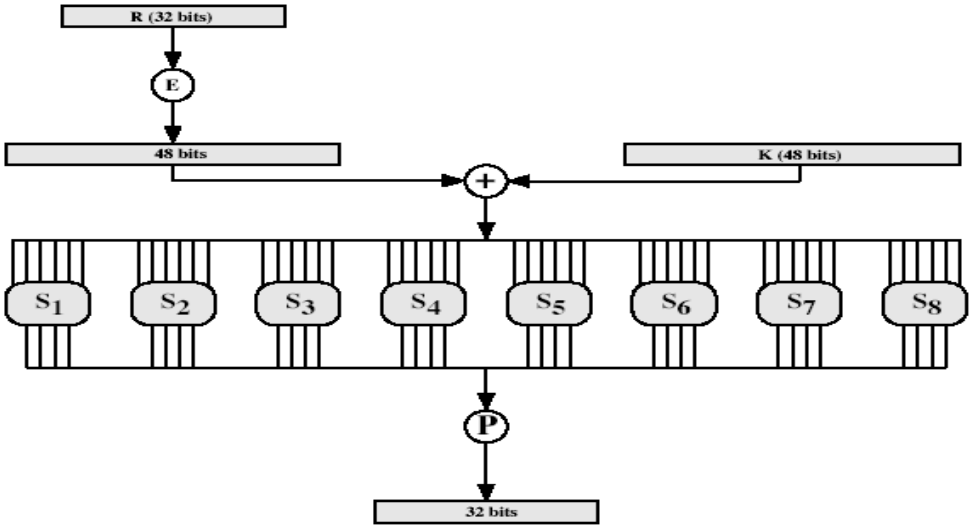
- **Cấu tạo một vòng của DES**

Sử dụng hai nửa 32 bít trái và 32 bít phải. Như đối với mọi mã Fiestel, nửa phải của vòng trước được chuyển qua nửa trái của bước sau và lấy đầu ra của hàm vòng trên nửa phải và khoá con cộng cơ số 2 với nửa trái. Có thể biểu diễn bằng công thức như sau:

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

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

Ở đây F lấy 32 bít nửa phải R, mở rộng thành 48 bít nhờ hoán vị E, rồi cộng vào với khoá con 48 bít. Sau đó chia thành 8 cụm 6 bít và cho qua 8 S-box để nhận được kết quả 32 bít. Đảo lần cuối sử dụng hoán vị 32 bít P nhận được 32 bít đầu ra, rồi cộng với nửa trái để chuyển thành nửa phải của bước sau.



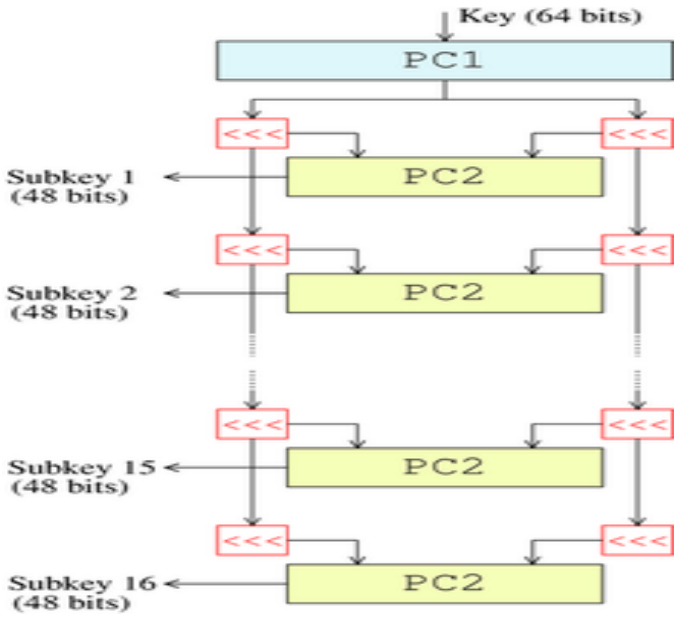
- **Các hộp thế S (xem phụ lục cuối tài liệu)**

  Có 8 hộp S khác nhau ánh xạ 6 bít vào 4 bít. Các hộp S box thực hiện các phép thế, chúng được cấu tạo không có qui luật và cố định. Mỗi S box là hộp 4 x 16 bít, mỗi hàng là một hoán vị của 16 phần tử. Giả sử ta có 6 bít đầu vào. Ta lấy hai bít ngoài 1-6 ghép lại được số nhị phân xác định chọn hàng từ 0 đến 3 trong S box. Bốn bít từ 2 đến 5 là một số nhị phân xác định cột từ 0 đến 15 trong S box. Lấy phần tử tương ứng trên hàng và cột mới được xác định, đây là một số từ 0 đến 15, chuyển sang số nhị phân ta được 4 bít đầu ra. Như vậy 48 bít chia thành có 8 cụm 6 bít, qua 8 S box được chuyển thành 8 cụm 4 bít, tổng cộng là 32 bít Việc chọn hàng trong các S box phụ thuộc cả dữ liệu và khoá - đặc trưng này được gọi là khoá tự xác định

Ví dụ:

S(18 09 12 3d 11 17 38 39) = 5fd25e03

- **Sinh khoá con của DES**
  - Tạo 16 khoá con sử dụng cho 16 vòng của DES. 56 bit khoá đầu vào được sử dụng như bảng 8 x 8, trong đó cột thứ 8 không sử dụng.
  - Hoán vị ban đầu của khoá PC1 và tách 56 bít thành hai nửa 28 bít.
  - 16 giai đoạn bao gồm
    - Ở mỗi vòng nửa trái và nửa phải được dịch trái vòng quanh tương ứng 1 và 2 bit. Hai nửa này được dùng tiếp cho vòng sau.
    - Đồng thời hai nửa cũng cho qua hoán vị PC2 và chọn mỗi nửa 24 bít gộp lại thành 48 bít để sinh khoá con..
  - Ứng dụng thực tế trên cả phần cứng và phần mềm đều hiệu quả



Các thông số cụ thể về hoán vị ban đầu, các hộp Box và thuật toán sinh khoá của DES được cho cuối tài liệu trong phần phụ lục.

- **Giải mã DES**

  Giải mã làm ngược lại quá trình mã hoá**.** Với thiết kế Fiestel thực hiện mã hoá tiếp với các khoá con từ SK16 ngược lại về SK1. Nhận thấy rằng hoán vị ban đầu IP sẽ trả lại tác dụng của hoán vị cuối FP. Vòng đầu với SK16 sẽ trả lại tác dụng của vòng mã thứ 16. Vòng thứ 16 với SK1 sẽ trả lại tác dụng của vòng mã đầu tiên. Hoán vị cuối FP trả lại tác dụng hoán vị ban đầu IP. Như vậy đã khôi phục lại được dữ liệu ban đầu.

3.2.3 Tính chất của DES

- **Tác dụng đồng loạt.** Khi ta thay đổi 1 bit trong khoá sẽ gây ra tác động đồng loạt làm thay đổi nhiều bit trên bản mã. Đây là tính chất mong muốn của khoá trong thuật toán mã hoá. Nếu thay đổi 1 bít đầu vào hoặc khoá sẽ kéo theo thay đổi một nửa số bít đầu ra. Do đó không thể đoán khoá được. Co thể nói rằng DES thể hiện tác động đồng loạt mạnh.
- **Sức mạnh của DES – kích thước khoá.**

Độ dài của khoá trong DES là 56 bít có $2^{56} = 7.2 \times 10^{16}$ giá trị khác nhau. Đây là con số rất lớn nên tìm kiếm duyệt rất khó khăn. Các thành tựu gần đây chỉ ra rằng thời gian cần thiết để giải một trang mã DES mà không biết khoá là: sau một vài tháng trên Internet trong năm 1997; một vài ngày trên thiết bị phần cứng tăng cường trong năm 1998; sau 22 giờ nếu kết hợp các biện pháp trong năm 1999. Như vậy vẫn có thể đoán được bản rõ sau một khoảng thời nhất định, nếu có nguồn lực máy tính mạnh. Chính vì vậy bây giờ người ta đã xét một vài biến thể của DES nhằm nâng cao sức mạnh cho DES.

- **Sức mạnh của DES – tấn công thời gian.**

  Đây là dạng tấn công vào cài đặt thực tế của mã. Ở đây sử dụng hiểu biết về quá trình cài đặt thuật toán mà suy ra thông tin về một sô khoá con hoặc mọi khoá con. Đặc biệt sử dụng kết luận là các tính toán chiếm khoảng thời gian khác nhau phụ thuộc vào giá trị đầu vào của nó. Do đó kẻ thám mã theo dõi thời gian thực hiện mà phán đoán về khoá. Có thể kẻ thám mã sáng tạo ra các loại card thông minh phán đoán khoá, mà còn phải bàn bạc thêm về chúng.

- **Sức mạnh của DES – tấn công thám mã.**

  Có một số phân tích thám mã trên DES, từ đó đề xuất xây dựng một số cấu trúc sâu về mã DES. Rồi bằng cách thu thập thông tin về mã, có thể đoán biết được tất cả hoặc một số khoá con đang dùng. Nếu cần thiết sẽ tìm duyệt những khoá còn lại. Nói chung, đó là những tấn công dựa trên phương pháp thống kê bao gồm: thám mã sai phân, thám mã tuyến tính và tấn công khoá liên kết.

- **Thám mã sai phân**

  Một trong những thành tựu công khai gần đây trong thám mã là phương pháp thám mã sai phân. Nó được biết đến bởi NSA trong những năm 70, chẳng hạn trong thiết kế DES. Murphy, Birham và Shamir công bố phương pháp sai phân năm 1990. Đây là phương pháp mạnh để phân tích mã khối.

Nó sử dụng phân tích hầu hết các mã khối hiện tại với mức độ thành công khác nhau. Nhưng DES có thể kháng cự lại các tấn công đó. Thám mã sai phân là tấn công thống kê chống lại các mã Fiestel. Mã Fiestel dùng các cấu trúc mã chưa được sử dụng trước kia như thiết kế S-P mạng có đầu ra từ hàm f chịu tác động bởi cả đầu vào và khoá. Do đó không thể tìm lại được giá trị bản rõ mà không biết khoá.

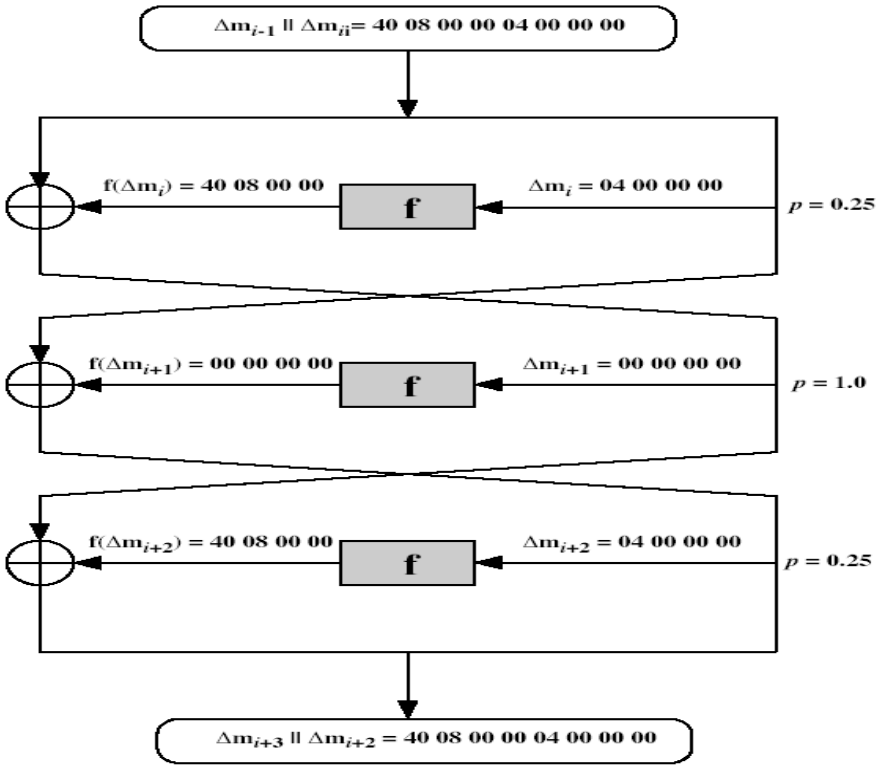
Thám mã sai phân so sánh hai cặp mã có liên quan với nhau

- Với sự khác biệt đã biết ở đầu vào
- Khảo sát sự khác biệt ở đầu ra
- Khi với cùng khoá con được dùng
- Trong công thức sau với hai đầu vào khác nhau, vế trái là sự khác biệt mã ở cùng vòng thứ i được biểu diễn qua sự khác biệt mã ở vòng trước đó i-1 và sự khác biệt của hàm f trong ngoặc vuông.

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

Sự khác biệt ở đầu vào cho sự khác biệt ở đầu ra với một xác suất cho trước.

- Nếu tìm được một thể hiện đầu vào - đầu ra với xác suất cao. Thì có thể luận ra khoá con được sử dụng trong vòng đó
- Sau đó có thể lặp lại cho nhiều vòng (với xác suất giảm dần)
  - Cặp đúng cho bít khoá như nhau
  - Cặp sai cho giá trị ngẫu nhiên
- Đối với số vòng lớn, xác suất để có nhiều cặp đầu vào 64 bít thoả mãn yêu cầu là rất nhỏ.
- Birham và Shamir chỉ ra rằng làm như thế nào để các đặc trưng lặp của 13 vòng có thể bẻ được DES 16 vòng đầy đủ.



- Qui trình thám mã như sau: thực hiện mã hoá lặp lại với cặp bản rõ có XOR đầu vào biết trước cho đến khi nhận được XOR đầu ra mong muốn
- Khi đó có thể tìm được
  - nếu vòng trung gian thỏa mãn XOR yêu cầu thì có cặp đúng
  - nếu không thì có cặp sai, tỷ lệ sai tương đối cho tấn công đã biết trước dựa vào thống kê.
- Sau đó có thể tạo ra các khoá cho các vòng theo suy luận sau

- **Thám mã tuyến tính**

  Đây là một phát hiện mới khác. Nó cũng dùng phương pháp thống kê. Ở đây cần lặp qua các vòng với xác suất giảm, nó được phát triển bởi Matsui và một số người khác vào đầu những năm 90. Cơ sở của phương pháp dựa trên tìm xấp xỉ tuyến tính. Và có nhận định rằng có thể tấn công DES với 2^{47} bản rõ đã biết. Như vậy thám mã tuyến tính vẫn không khả thi trong thực tế.

- Tìm xấp xỉ tuyến tính với xác suất p != ½

$$P[i_1,i_2,...,i_a] (+) C[j_1,j_2,...,j_b] = K[k_1,k_2,...,k_c]$$

trong đó i_a, j_b, k_c là các vị trí bit trong bản rõ, mã, khoá.

- Điều kiện trên cho phương trình tuyến tính của các bít khoá. Để nhận được 1 bít khoá sử dụng thuật toán lân cận tuyến tính
- Sử dụng một số lớn các phương trình thử nghiệm. Hiệu quả cho bởi $|p - 1/2|$

Trong quá trình tìm hiểu DES người ta đã hệ thống lại các tiêu chuẩn thiết kế DES. Như báo cáo bởi Copperscmith trong [COPP94]:

- Có 7 tiêu chuẩn đối với S box được cung cấp để đảm bảo
  - tính phi tuyến tính
  - chống tham mã sai phân
  - Rối loạn tốt
- Có 3 tiêu chuẩn cho hoán vị P để tăng độ khuếch tán

- **Các nguyên lý mã khối**

  Các nguyên lý cơ bản của mã khối giống như Fiestel đề xuất trong những năm 70:

  - Có một số vòng: càng nhiều càng tốt; tấn công tốt nhất phải tìm tổng thể
  - Trong mỗi vòng có hàm cung cấp độ rối loạn là phi tuyến, tác động đồng loạt
  - Qui trình sinh khoá con phức tạp, khoá tác động đồng loạt đến bản mã.

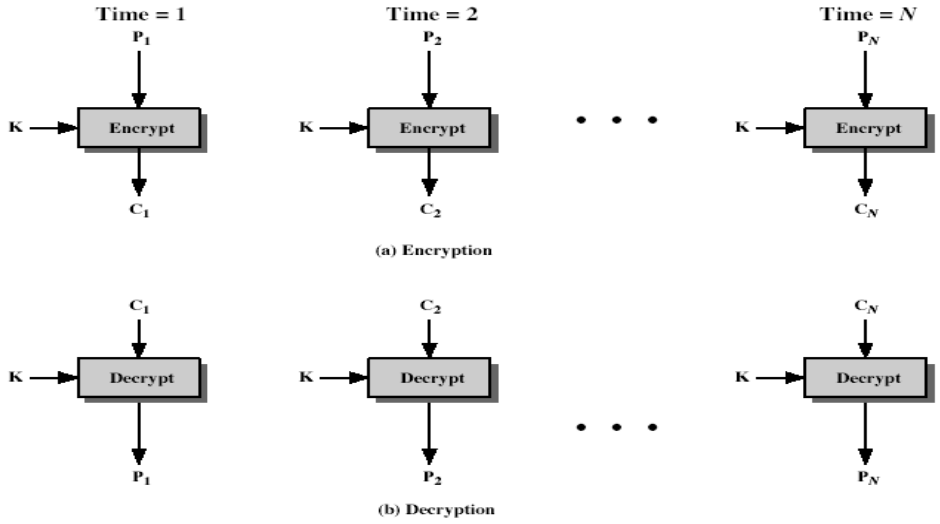
3.2.4 Các kiểu thao tác của DES

Mã khối mã các block có kích thước cố định. Chẳng hạn DES mã các block 64 bít với khoá 56 bít Cần phải có cách áp dụng vào thực tế vì các thông tin cần mã có kích thước tùy ý. Trước kia có 4 kiểu thao tác được định nghĩa cho DES

theo chuẩn ANSI: **ANSI X3.106-1983 Modes of Use.** Bây giờ mở rộng them có 5 cách cho DES và chuẩn mã nâng cao (AES – Advanced Encryption Standards). Trong đó có kiểu áp dụng cho khối và có kiểu áp dụng cho mã dòng.

1. Sách mật mã điện tử (Electronic Codebook Book - ECB)

- Mẫu tin được chia thành các khối độc lập, sau đó mã từng khối
- Mỗi khối là giá trị cần thay thế như dùng sách mã, do đó có tên như vậy
- Mỗi khối được mã độc lập với các mã khác $C_i = DES_{K1}(P_i)$
- Khi dùng: truyền an toàn từng giá trị riêng lẻ



- Ưu và nhược của ECB
  - Lặp trên bản mã được chỉ rõ lặp trên bản tin
    - Nếu dóng đúng khối
    - Đặc biệt với hình ảnh
    - Hoặc với bản tin mà thay đổi rất ít

    sẽ trở thành đối tượng để thám mã
  - Nhược điểm là các khối được mã độc lập
  - Được sử dụng chủ yếu khi gửi một ít dữ liệu

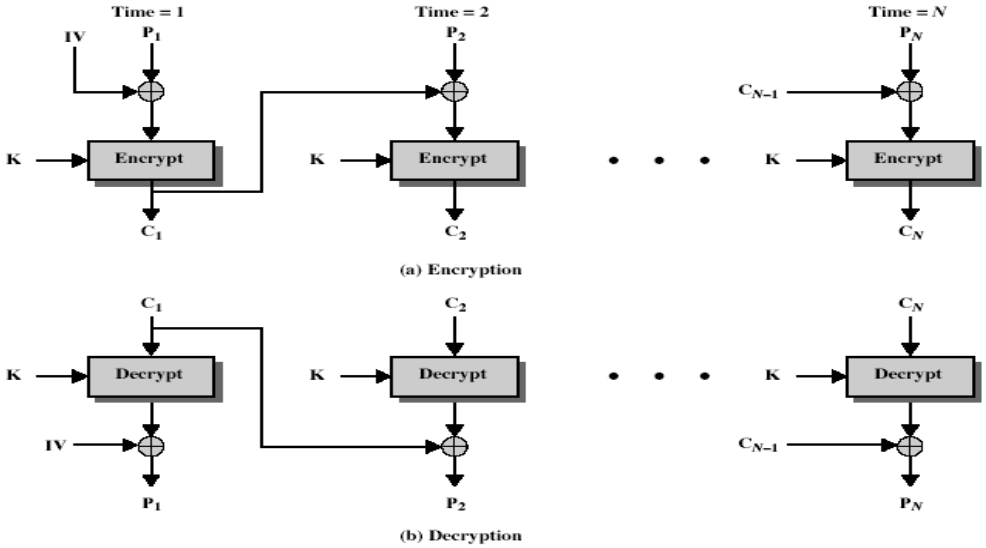
2. Dây chuyền mã khối (Cipher Block Chaining - CBC)

- Các mẫu tin được chia thành các khối
- Nhưng chúng được liên kết với nhau trong quá trình mã hoá
- Các block được sắp thành dãy, vì vậy có tên như vậy
- Sử dụng véctơ ban đầu IV để bắt đầu quá trình

$$C_i = DES_{K1}(P_i \text{ XOR } C_{i-1})$$

$$C_{-1} = IV$$

- Dùng khi: mã dữ liệu lớn, xác thực



- **Ưu và nhược của CBC**
  - Mỗi khối mã phụ thuộc vào tất cả các khối bản rõ
  - Sự thay đổi của bản tin ở đâu đó sẽ kéo theo sự thay đổi của mọi khối mã
  - Cần giá trị véc tơ ban đầu IV được biết trước bởi người gửi và người nhận
    - Tuy nhiên nếu IV được gửi công khai, kẻ tấn công có thể thay đổi bít đầu tiên và thay đổi cả IV để bù trừ
    - Vậy IV cần phải có giá trị cố định trước hoặc mã hoá trong chế độ ECB và gửi trước phần còn lại của mẩu tin
  - Ở cuối bản tin, để kiểm soát các block ngắn còn lại

- Có thể bổ sung các giá trị không phải dữ liệu như NULL
- Hoặc dùng bộ đệm cuối với số byte đếm kích thước của nó.

Ví dụ

[b1 b2 b3 0 0 0 0 5] <- 3 data bytes,

vậy có 5 bytes dành cho đệm và đếm.

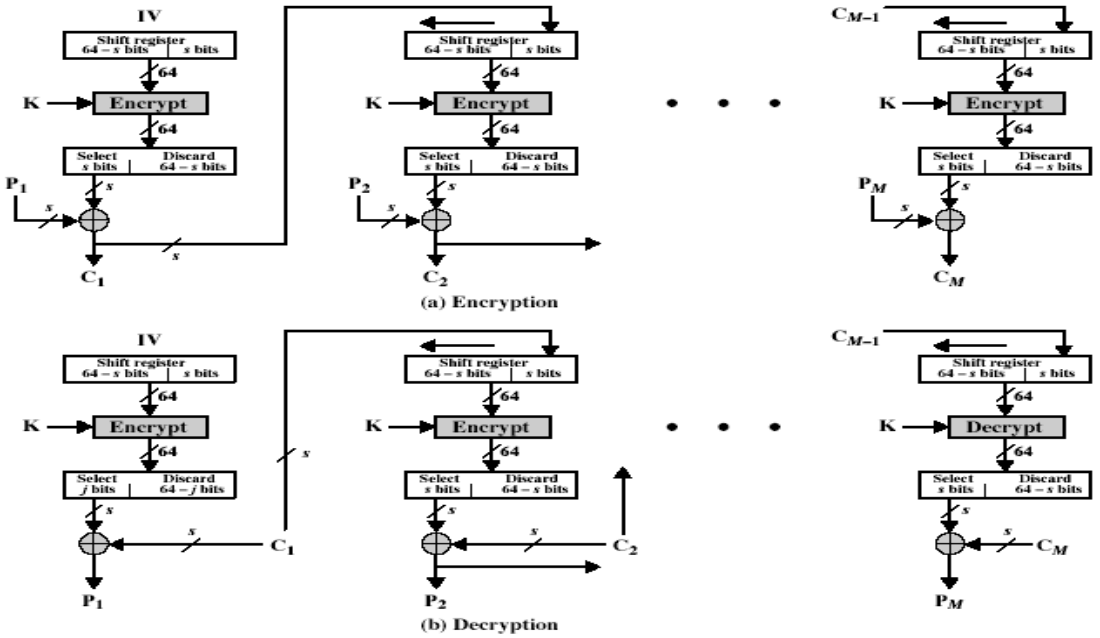
3. Mã phản hồi ngược (Cipher FeedBack - CFB)

- Bản tin coi như dòng các bít
- Bổ sung vào đầu ra của mã khối
- Kết quả phản hồi trở lại cho giai đoạn tiếp theo, vì vậy có tên như vậy.
- Nói chung cho phép số bít phản hồi là 1, 8, 64, hoặc tuỳ ý: ký hiệu tương ứng là CFB1, CFB8, CFB64,…
- Thường hiệu quả sử dụng cả 64 bít

$$C_i = P_i \text{ XOR } DES_{K1}(C_{i-1})$$

$$C_{-1} = IV$$

- Được dùng cho mã dữ liệu dòng, xác thực



Ưu và nhược điểm của mã phản hồi ngược

- Được dùng khi dữ liệu đến theo byte/bit
- Chế độ dòng thường gặp nhất
- Hạn chế là cần ngăn chuồng khi mã khối sau mỗi n bit
- Nhận xét là mã khối được dùng ở chế độ mã ở cả hai đầu
- Lỗi sẽ lan ra một vài block sau lỗi

4. Phản hồi ngược đầu ra (Output FeedBack - OFB)

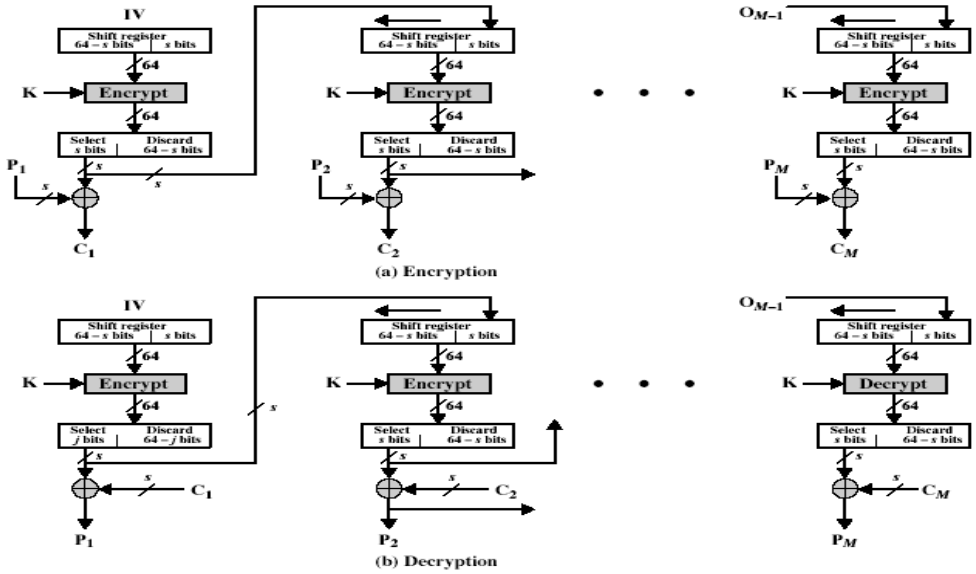
- Mẩu tin xem như dòng bit
- Đầu ra của mã được bổ sung cho mẩu tin
- Đầu ra do đó là phản hồi, do đó có tên như vậy
- Phản hồi ngược là độc lập đối với bản tin
- Có thể được tính trước

$$C_i = P_i \text{ XOR } O_i$$

$$O_i = DES_{K1}(O_{i-1})$$

$$O_{-1} = IV$$

- Được dùng cho mã dòng trên các kênh âm thanh



Ưu điểm và nhược điểm của OFB

- Được dùng khi lỗi phản hồi ngược lại hoặc ở nơi cần mã trước khi mẩu tin sẵn sàng

- Rất giống CFB
- Nhưng phản hồi là từ đầu ra của mã và độc lập với mẩu tin
- Là biến thể của mã Vernam, suy ra không sử dụng lại với cùng một dãy (Key + IV)
- Người gửi và người nhận phải đồng bộ, có phương pháp khôi phục nào đó là cần thiết để đảm bảo việc đó.
- Nguyên bản chỉ rõ m bit phản hồi ngược theo các chuẩn
- Các nghiên cứu tiếp theo chỉ ra rằng chỉ có OFB64 là dùng được

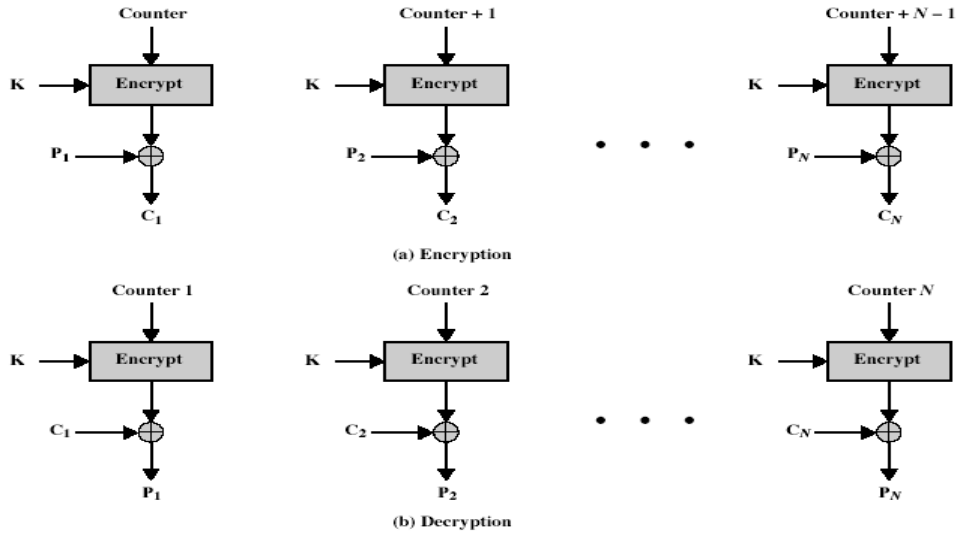
5. Bộ đếm CTR (Counter)

- Là chế độ mới, tuy đã được đề xuất từ lâu
- Giống như OFB, nhưng mã giá trị đếm thay vì giá trị phản hồi tuỳ ý.
- Cần phải có khoá khác và giá trị đếm cho mỗi khối bản rõ (không bao giờ dùng lại)

$$C_i = P_i \text{ XOR } O_i$$

$$O_i = DES_{K1}(i)$$

- Được dùng mã trên mạng với tốc độ cao
- Ưu và nhược điểm của CTR
  - Hiệu quả
    - Do có thể mã song song
    - Chuẩn bị trước nếu cần
    - Tốt cho các kết nối với tốc độ rất cao
  - Truy cập ngẫu nhiên đến các khối dữ liệu mã
  - Tính an toàn có thể chứng minh được
  - Nhưng phải tin tưởng không bao giờ dùng lại khoá/đếm, nếu không có thể bẻ.



3.3 Chuẩn mã nâng cao (AES)

3.3.1 Nguồn gốc

Rõ ràng cần phải thay thế DES, vì có những tấn công về mặt lý thuyết có thể bẻ được nó. Một số tấn công nghiên cứu thấu đáo khoá đã được trình diễn. Người ta thấy rằng, cần sử dụng Triple DES (sử dụng DES ba lần liên tiếp) cho các ứng dụng đòi hỏi tăng cường bảo mật, nhưng quá trình mã và giải mã chậm, đồng thời với khối dữ liệu nhỏ. Do đó Viện chuẩn quốc gia Hoa kỳ US NIST ra lời kêu gọi tìm kiếm chuẩn mã mới vào năm 1997. Sau đó có 15 đề cử được chấp nhận vào tháng 6 năm 1998. Và được rút gọn còn 5 ứng cử viên vào tháng 6 năm 1999. Đến tháng 10 năm 2000, mã Rijndael được chọn làm chuẩn mã nâng cao và được xuất bản là chuẩn FIPS PUB 197 vào 11/2001.

Yêu cầu của AES

- Là mã khối đối xứng khoá riêng.
- Kích thước khối dữ liệu 128 bit và độ dài khoá là tùy biến: 128, 192 hoặc 256 bit.
- Chuẩn mã mới phải mạnh và nhanh hơn Triple DES. Mã mới có cơ sở ls thuyết mạnh để thời gian sống của chuẩn khoảng 20-30 năm (cộng thêm thời gian lưu trữ).
- Khi đưa ra thành chuẩn yêu cầu cung cấp chi tiết thiết kế và đặc tả đầy đủ. Đảm bảo rằng chuẩn mã mới cài đặt hiệu quả trên cả C và Java.

- NIST in rút gọn mọi đề xuất, phân tích và không phân loại.

3.3.2 Tiêu chuẩn triển khai của AES

- Tiêu chuẩn ban đầu:
    - An toàn - chống đỡ mọi tấn công thám mã về thực tế
    - Giá trị về mặt tính toán
    - Các đặc trưng cài đặt và thuật toán.
- Tiêu chuẩn cuối cùng:
    - An toàn tổng thể
    - Dễ cài đặt phần mềm và phần cứng
    - Chống được tấn công về mặt cài đặt
    - Mềm dẻo trong mã / giải mã, khoá và các yếu tố khác
- Danh sách các ứng cử viên Chuẩn mã nâng cao được rút gọn:
    - MARS (IBM): phức tạp, nhanh, biên độ tin cậy cao
    - RC6 (USA): đơn giản, rất nhanh, biên độ tin cậy thấp
    - Rijndael (Bỉ): rõ ràng, nhanh, biên độ tin cậy tốt
    - Serpent (Châu Âu): chậm, rõ ràng, biên độ tin cậy rất cao
    - Twofish (USA): phức tạp, rất nhanh, biên độ tin cậy cao
- Sau đó tục phân tích và đánh giá. Tập trung vào việc so sánh các thuật toán khác nhau:
    - Ít vòng nhưng phức tạp với nhiều vòng đơn giản hơn.
    - Nêu rõ cải tiến các mã đã có với các đề xuất mới.

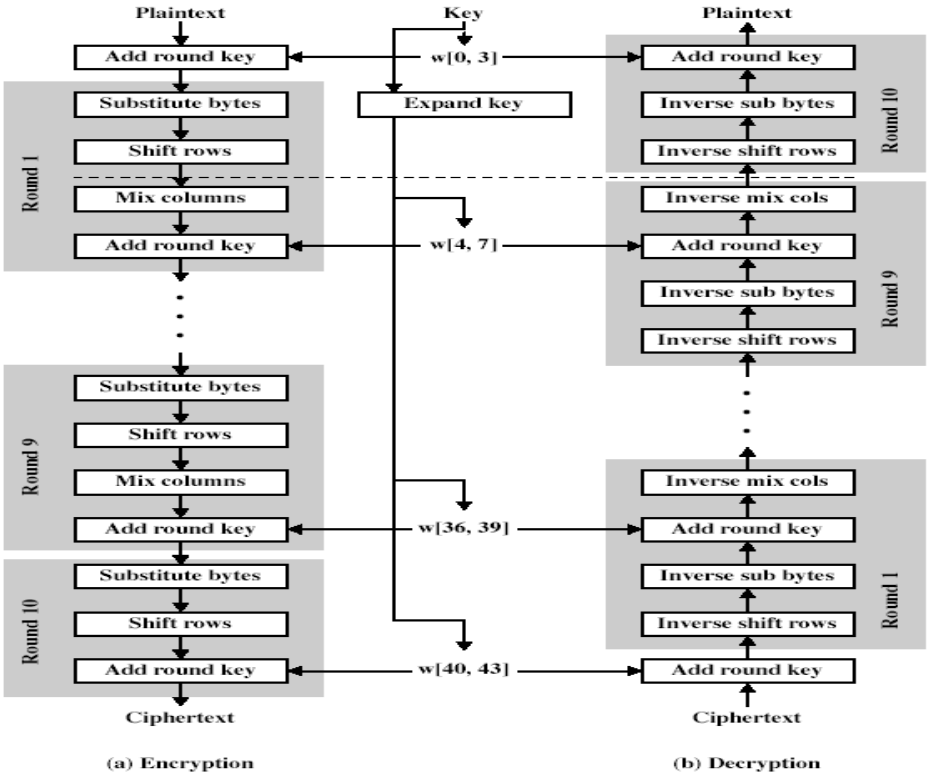
3.3.3 Chuẩn mã nâng cao AES – Rijndael

Cuối cùng Rijndael được chọn là chuẩn mã nâng cao. Nó được thiết kế bởi Rijmen – Daemen ở Bỉ, có các đặc trưng sau:

- Có 128/192/256 bit khoá và 128 bit khối dữ liệu.
- Lặp hơi khác với Fiestel
    - Chia dữ liệu thành 4 nhóm – 4 byte
    - Thao tác trên cả khối mỗi vòng

- Thiết kế để:
  - chống lại các tấn công đã biết
  - tốc độ nhanh và nén mã trên nhiều CPU
  - Đơn giản trong thiết kế

- Xử lý khối dữ liệu 128 bit như 4 nhóm của 4 byte: 128 = 4*4*8 bit. Mỗi nhóm nằm trên một hàng. Ma trận 4 hàng, 4 cột với mỗi phần tử là 1 byte coi như trạng thái được xử lý qua các vòng mã hoá và giải mã.
- Khoá mở rộng thành mảng gồm 44 từ 32 bit w[i].
- Có tùy chọn 9/11/13 vòng, trong đó mỗi vòng bao gồm
  - Phép thế byte (dùng một S box cho 1 byte)
  - Dịch hàng (hoán vị byte giữa nhóm/cột)
  - Trộn cột (sử dụng nhân ma trận của các cột)
  - Cộng khoá vòng (XOR trạng thái dữ liệu với khoá vòng).
  - Mọi phép toán được thực hiện với XOR và bảng tra, nên rất nhanh và hiệu quả.
- Sơ đồ Rijndael



- **Phép thế Byte**
  - Phép thê byte đơn giản
  - Sử dụng một bảng 16 x 16 byte chứa hoán vị của tất cả 256 giá trị 8 bit
  - Mỗi byte trạng thái được thay bởi byte trên hàng xác định bởi 4 bit trái và cột xác định bởi 4 bit phải.

    Chẳng hạn {95} được thay bởi hàng 9, cột 5, mà giá trị sẽ là {2A}.
  - S box được xây dựng sử dụng hoán vị các giá trị trong GF(2^8) đã được xác định trong chương trước.
  - Thiết kế để chống mọi tấn công đã biết
- **Dịch hàng**
  - Dịch hàng vòng quanh trên mỗi hàng
    - Hàng 1 không đổi
    - Hàng 2 dịch vòng quanh 1 byte sang trái
    - Hàng 3 dịch vòng quanh 2 byte sang trái
    - Hàng 4 dịch vòng quanh 3 byte sang trái
  - Giải mã thực hiện dịch ngược lại sang phải
  - Vì trạng thái được xử lý bởi cột, bước này thực chất là hoán vị byte giữa các cột.
- **Trộn các cột**
  - Mỗi cột được xử lý riêng biệt.
  - Mỗi byte được thay bởi 1 giá trị phụ thuộc vào tất cả 4 byte trong cột
  - Nhân ma trận hiệu quả trong GF(2^8), sử dụng đa thức nguyên tố

$$m(x) = x^8+x^4+x^3+x+1$$

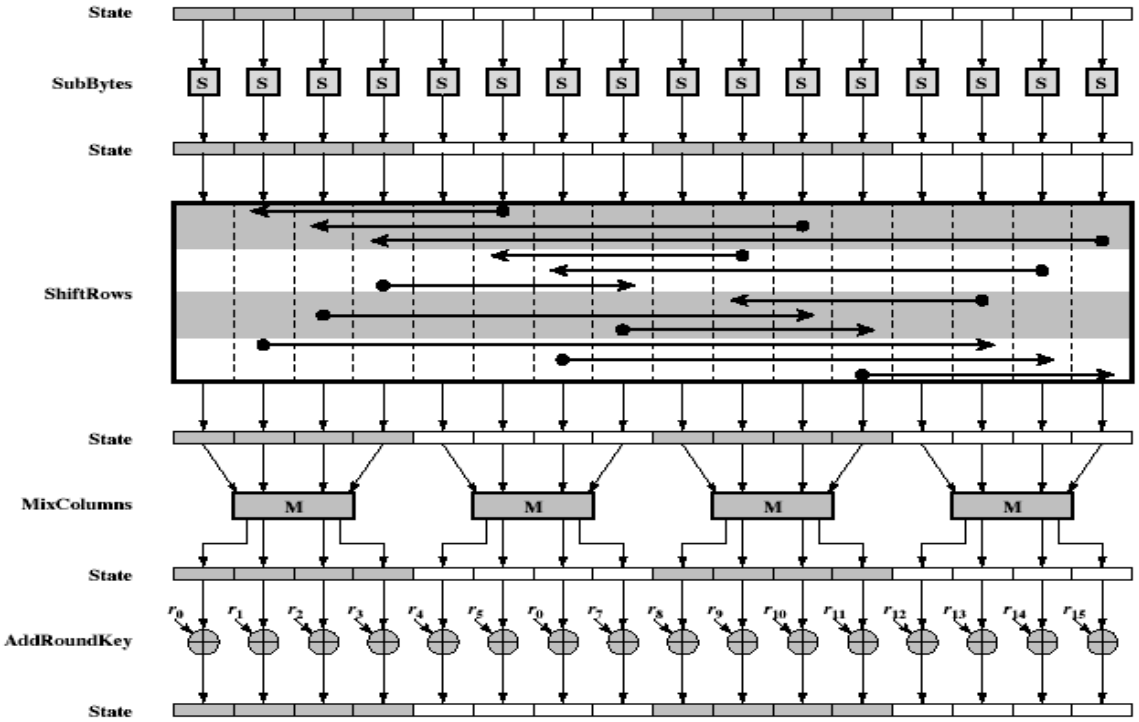
$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

Trộn cột

- Có thể biểu diễn mỗi cột mới là nghiệm của 4 phương trình
  - để tìm ra byte mới trong mỗi cột
- Mã yêu cầu sử dụng ma trận nghịch đảo
  - Với hệ số lớn thì tính toán khó khăn hơn
- Có các đặc trưng khác của cột như sau:
  - Mỗi cột là một đa thức bậc 3 gồm 4 số hạng
  - Với mỗi phần tử là một byte tương ứng với phần tử trong $GF(2^8)$.
  - Các đa thức nhân tính theo Modulo (x^4+1).

- **Cộng khoá quay vòng**
  - XOR trạng thái với 128 bit khoá quay vòng
  - Xử lý lại bằng cột (hiệu quả qua một loạt các thao tác bit)
  - Nghịch đảo cho giải mã hoàn toàn xác định, vì khi XOR với nghịch đảo của bản thân nó, XOR trùng với đảo bit của khoá quay vòng.
  - Thiết kế để đơn giản nhất có thể
    - Dạng mã Vernam với khoá mở rộng
    - Đòi hỏi thêm một số bước tăng độ phức tạp/tính an toàn.
- **Một vòng AES**
- **Mở rộng khoá AES**
  - Dùng khoá 128 bit (16 byte) và mở rộng thành mảng gồm 44/52/60 từ 32 bit.
  - Bắt đầu bằng việc copy khoá vào 4 từ đầu
  - Sau đó tạo quay vòng các từ mà phụ thuộc vào giá trị ở các vị trí trước và 4 vị trí sau
    - 3 trong 4 trường hợp chỉ là XOR chúng cùng nhau
    - Mỗi cái thứ 4 có S box kết hợp quay và XOR với hằng số trước đó, trước khi XOR cùng nhau

- Thiết kế chống các tấn công đã biết



- **Giải mã AES**
  - Giải mã ngược lại không duy nhất vì các bước thực hiện theo thứ tự ngược lại.
  - Nhưng có thể xác định mã ngược tương đương với các bước đã làm đối với mã
    - Nhưng sử dụng ngược lại với từng bước
    - Với khoá con khác nhau
  - Thực hiện được vì kết quả không thay đổi khi
    - Đổi lại phép thế byte và dịch các hàng
    - Đổi lại việc trộn các cột và bổ sung khoá vòng
  - Lý do mở rộng khoá: các tiêu chuẩn thiết kế bao gồm
    - Giả sử biết một phần khoá, khi đó không đủ để biết nhiều hơn, tức là các khoá con khác hoặc khoá nói chung.
    - Phép biến đổi nghịch đảo được.
    - Nhanh đối với nhiều kiểu CPU.

- Sử dụng hằng số vòng để làm mất tính đối xứng
- Khuếch tán bit khoá thành khoá con cho các vòng
- Có đủ tính phi đối xứng để chống thám mã
- Đơn giản trong việc giải mã

o Các khía cạnh cài đặt:

- có thể cài đặt hiệu quả trên CPU 8 bit
  - Phép thế byte làm việc trên các byte sử dụng bảng với 256 đầu vào.
  - Dịch hàng là phép dịch byte đơn giản
  - Cộng khoá vòng làm việc trên byte XOR
  - Các cột hỗn hợp yêu cầu nhân ma trận trong GF(2^8) mà làm việc trên giá trị các byte, có thể đơn giản bằng cách tra bảng
- có thể cài đặt hiệu quả trên CPU 32 bit
  - Xác định lại các bước để sử dụng từ 32 bit
  - Có thể tính trước 4 bảng với 256 đầu vào
  - Sau đó mỗi cột trong mỗi vòng có thể tính bằng cách tra 4 bảng và 4 XOR
  - Cần 16 Kb để lưu các bảng
- Những nhà thiết kế tin tưởng rằng việc cài đặt rất hiệu quả này là yếu tố cơ bản trong việc chọn nó là mã AES.