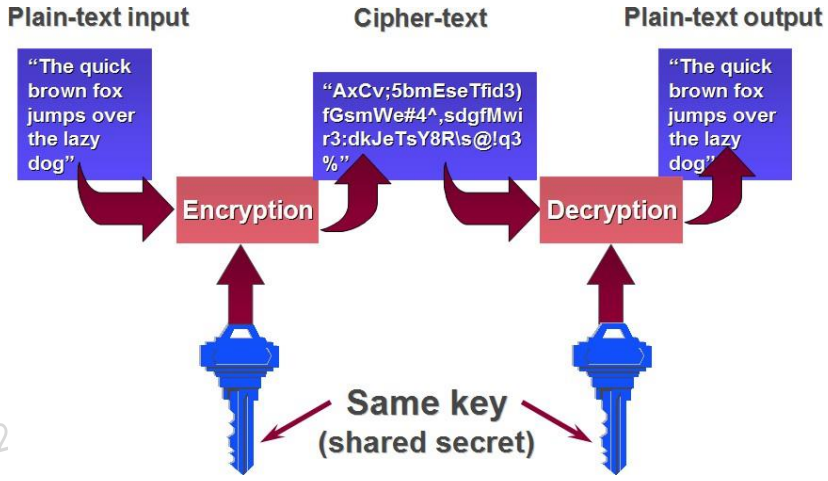
1. Khái quát về mã hóa khóa đối xứng

Mã hóa khóa đối xứng (Symmetric key encryption) hay còn gọi là mã hóa khóa bí mật (Secret key encryption) sử dụng một khóa bí mật (Secret key) duy nhất cho cả quá trình mã hóa và giải mã. Khóa bí mật được sử dụng trong quá trình mã hóa và giải mã còn được gọi là *khóa chia sẻ* (Shared key) do bên gửi và bên nhận cần chia sẻ khóa bí mật một cách an toàn trước khi có thể thực hiện việc mã hóa và giải mã. Hình 3.14 minh họa quá trình mã hóa và giải mã sử dụng chung một khóa bí mật chia sẻ.



Hình 3.14. Mã hóa khóa đối xứng (Symmetric key encryption)

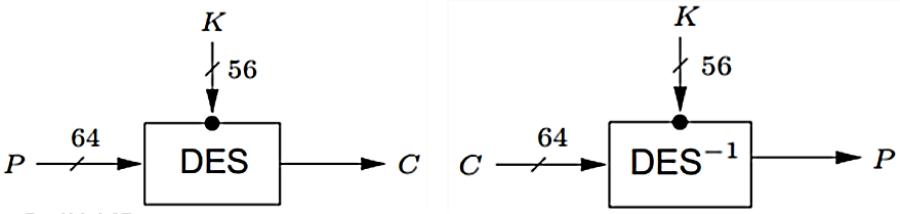
Các hệ mã hóa khóa đối xứng thường sử dụng khóa với kích thước tương đối ngắn. Một số kích thước khóa được sử dụng phổ biến là 64, 128, 192 và 256 bit. Do sự phát triển nhanh về tốc độ tính toán của máy tính, nên các khóa có kích thước nhỏ hơn 128 bit được xem là không an toàn và hầu hết các hệ mã hóa khóa đối xứng đảm bảo an toàn hiện tại sử dụng khóa có kích thước từ 128 bit trở lên. Ưu điểm nổi bật của các hệ mã hóa khóa đối xứng là có độ an toàn cao và tốc độ thực thi nhanh. Tuy nhiên, nhược điểm lớn nhất của các hệ mã hóa khóa đối xứng là việc quản lý và phân phối khóa rất khó khăn, đặc biệt là trong các môi trường mở như mạng Internet do các bên tham gia phiên truyền thông cần thực hiện việc trao đổi các khóa bí mật một cách an toàn trước khi có thể sử dụng chúng để mã hóa và giải mã các thông điệp trao đổi.

Một số hệ mã hóa khóa đối xứng tiêu biểu, gồm DES (Data Encryption Standard), 3- DES (Triple-DES), AES (Advanced Encryption Standard), IDEA (International Data Encryption Algorithm), Blowfish, Twofish, RC4 và RC5. Phần tiếp theo của mục này là mô tả các giải thuật mã hóa DES, 3-DES và AES do chúng là các giải thuật đã và đang được sử dụng rộng rãi nhất trên thực tế.

1.1 Giải thuật mã hóa DES và 3-DES

1.1.1 DES

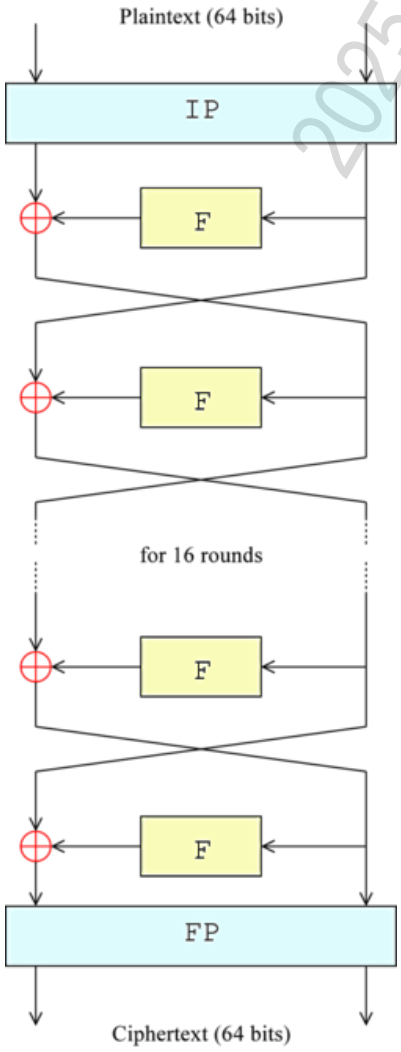
DES (Data Encryption Standard) được phát triển tại IBM với tên gọi Lucifer vào đầu những năm 1970 và được chấp nhận là chuẩn mã hóa ở Mỹ vào năm 1977. DES được sử dụng rộng rãi trong những năm 1970 và 1980. DES là dạng mã hóa khối với khối dữ liệu vào kích thước 64 bit và khóa 64 bit, trong đó thực sử dụng 56 bit (còn gọi là kích thước hiệu dụng của khóa) và 8 bit dùng cho kiểm tra chẵn lẻ. Một ưu điểm của DES là sử dụng chung một giải thuật cho cả khâu mã hóa và khâu giải mã, như minh họa trên Hình 3.15, trong đó *P* là khối bản rõ 64 bit, K là khóa với kích thước hiệu dụng 56 bit, *C* là khối bản mã 64 bit, *DES* biểu diễn khâu mã hóa và DES^{-1} biểu diễn khâu giải mã. Hiện nay DES được coi là không an toàn do nó có không gian khóa nhỏ, dễ bị vét cạn và tốc độ tính toán của các hệ thống máy tính ngày càng nhanh.



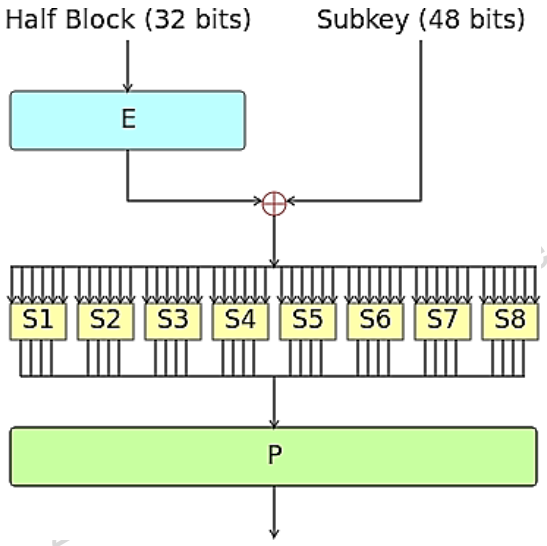
Hình 3.15. Các khâu mã hóa và giải mã của DES

Với mỗi khối dữ liệu đầu vào 64 bit, DES thực hiện 3 bước xử lý như minh họa trên Hình 3.16 để chuyển nó thành khối mã 64 bit tương ứng. Các bước cụ thể gồm:

- Bước 1: Hoán vị khởi tạo (IP – Initial Permutation);
- Bước 2: 16 vòng lặp chính thực hiện xáo trộn dữ liệu sử dụng hàm Feistel (F). Sau mỗi vòng lặp, các kết quả trung gian được kết hợp lại sử dụng phép ⊕ (XOR);
- Bước 3: Hoán vị kết thúc (FP – Final Permutation).



Hình 3.16. Các bước xử lý chuyển khối rõ 64 bit thành khối mã 64 bit của DES



Hình 3.17. Các bước xử lý của hàm Feistel (F)

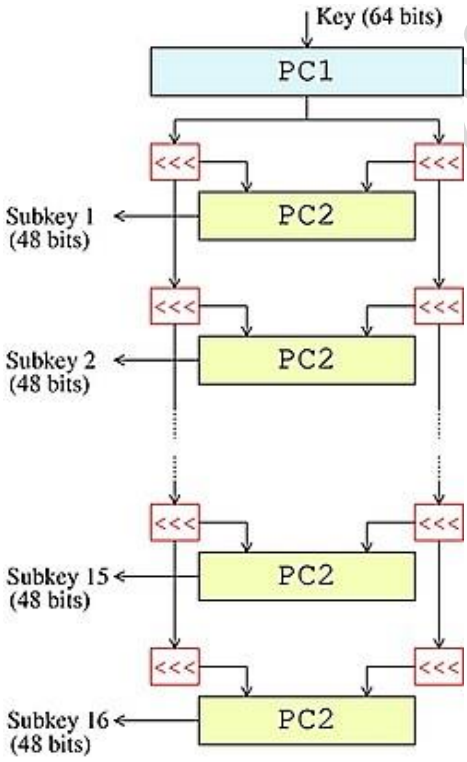
Hàm Feistel (F) là hạt nhân trong các vòng lặp xử lý dữ liệu của DES. Trước hết, khối 64 bit được chia thành 2 khối 32 bit và được xử lý lần lượt. Hàm Feistel được thực hiện

trên một khối dữ liệu 32 bit (Half Block 32 bits) gồm 4 bước xử lý như minh họa trên Hình 3.17. Cụ thể, các bước xử lý như sau:

- E (Expansion): thực hiện mở rộng 32 bit khối đầu vào thành 48 bit bằng cách nhân đôi một nửa số bit.
- ⊕: Trộn khối 48 bit kết quả ở bước E với khóa phụ 48 bit. Có 16 khóa phụ (Subkey) được tạo từ khóa chính để sử dụng cho 16 vòng lặp.
- Si (Substitution): Khối dữ liệu 48 bit được chia thành 8 khối 6 bit và được chuyển cho các bộ thay thế (S1-S8). Mỗi bộ thay thế Si sử dụng phép chuyển đổi phi tuyến tính để chuyển 6 bit đầu vào thành 4 bit đầu ra theo bảng tham chiếu. Các bộ thay thế là thành phần nhân an ninh (Security core) của DES.
- P (Permutation): khối 32 bit đầu ra từ các bộ thay thế được sắp xếp bằng phép hoán vị cố định (Fixed permutation) cho ra đầu ra 32 bit.

DES sử dụng một thủ tục sinh 16 khóa phụ từ khóa chính để sử dụng trong 16 vòng lặp hàm Feistel. Hình 3.18 minh họa thủ tục sinh 16 khóa phụ từ khóa chính của DES. Các bước xử lý chính của thủ tục sinh khóa phụ như sau:

- 56 bit khóa được chọn từ khóa gốc 64 bit bởi PC1 (Permuted Choice 1). 8 bit còn lại được hủy hoặc dùng để kiểm tra chẵn lẻ;
- 56 bit được chia thành 2 phần 28 bit, mỗi phần được xử lý riêng;
- Mỗi phần được quay trái 1 hoặc 2 bit;
- Hai phần được ghép lại và 48 bit được chọn làm khóa phụ 1 (Subkey 1) bởi PC2;
- Lặp lại bước trên để tạo 15 khóa phụ còn lại.



Hình 3.18. Thủ tục sinh các khóa phụ từ khóa chính của DES

Như đã đề cập, giải thuật DES có thể sử dụng cho cả khâu mã hóa và giải mã. Trong khâu giải mã các bước xử lý tương tự khâu mã hóa. Tuy nhiên, các khóa phụ sử dụng cho các vòng lặp được sử dụng theo trật tự ngược lại: khóa phụ số 16, 15,…, 2, 1 được sử dụng cho các vòng lặp số 1, 2,…, 15, 16 tương ứng.

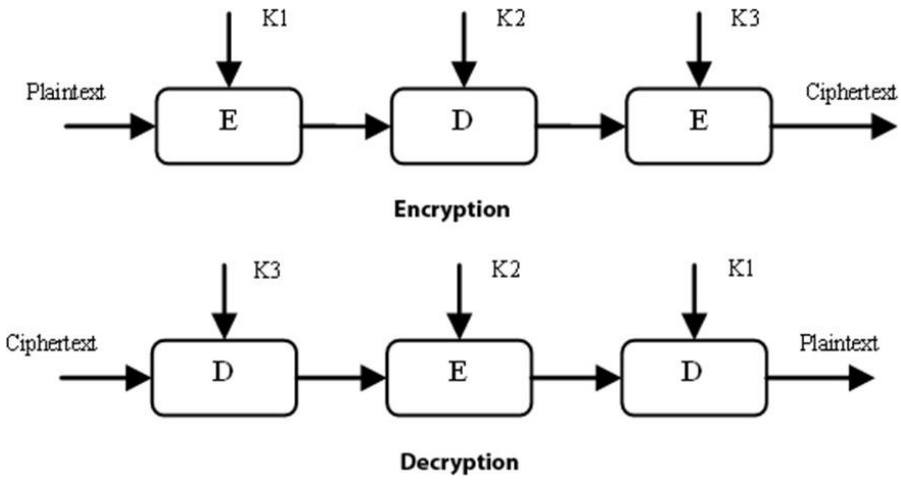
1.1.2 3-DES

3-DES hay Triple DES có tên đầy đủ là Triple Data Encryption Algorithm (TDEA) được phát triển từ giải thuật DES bằng cách áp dụng DES 3 lần cho mỗi khối dữ liệu đầu vào 64 bit. 3-DES sử dụng một bộ gồm 3 khóa DES: K1, K2, K3, trong đó mỗi khóa kích thước hiệu dụng là 56 bit. 3-DES cho phép lựa chọn các bộ khóa:

- Lựa chọn 1: cả 3 khóa độc lập, với tổng kích thước bộ khóa là 168 bit;
- Lựa chọn 2: K1 và K2 độc lập, K3 = K1, với tổng kích thước bộ khóa là 112 bit;
- Lựa chọn 3: 3 khóa giống nhau, K1 = K2 = K3, với tổng kích thước bộ khóa là 56 bit.

Hình 3.19 biểu diễn quá trình mã hóa và giải mã với giải thuật 3-DES, trong đó khâu mã hóa được ký hiệu là E và khâu giải mã được ký hiệu là D. Theo đó, ở bên gửi bản rõ (Plaintext) được mã hóa bằng khóa K1, giải mã bằng khóa K2

và mã hóa bằng khóa K3 để cho ra bản mã (Ciphertext). Ở bên nhận, quá trình giải mã bắt đầu bằng việc giải mã bằng khóa K3, sau đó mã hóa bằng khóa K2 và cuối cùng giải mã bằng khóa K1 để khôi phục bản rõ. Ưu điểm của 3-DES là nâng cao được độ an toàn nhờ tăng kích thước khóa. Tuy nhiên, nhược điểm chính của 3-DES là tốc độ thực thi chậm do phải thực hiện DES lặp 3 lần cho mỗi khâu mã hóa và giải mã.



Hình 3.19. Mã hóa và giải mã với giải thuật 3-DES

2. Giải thuật mã hóa AES

2.1 Giới thiệu

AES (Advanced Encryption Standard) là một chuẩn mã hóa dữ liệu được Viện Tiêu chuẩn và Công nghệ Mỹ (NIST) công nhận năm 2001. AES được xây dựng dựa trên Rijndael cipher phát triển và công bố năm 1998 bởi 2 nhà mật mã học người Bỉ là Joan Daemen và Vincent Rijmen. AES là dạng mã hóa khối, với khối dữ liệu vào có kích thước là 128 bit và khóa bí mật với kích thước có thể là 128, 192, hoặc 256 bit. AES

được thiết kế dựa trên mạng hoán vị-thay thế (Substitution-permutation network) và nó có thể cho tốc độ thực thi cao khi cài đặt bằng cả phần mềm và phần cứng. Đặc biệt, giải thuật AES đã được tích hợp vào các bộ vi xử lý gần đây của hãng Intel dưới dạng tập lệnh AES-NI, giúp tăng đáng kể tốc độ thực thi các thao tác mã hóa và giải mã dựa trên AES.

AES vận hành dựa trên một ma trận vuông 4x4, được gọi là *state* (trạng thái). Ma trận này gồm 16 phần tử, mỗi phần tử là 1 byte dữ liệu. State được khởi trị là khối 128 bit bản rõ và qua quá trình biến đổi sẽ chứa khối 128 bit bản mã ở đầu ra. Như đã đề cập, AES hỗ trợ 3 kích thước khóa và kích thước của khóa quyết định số vòng lặp chuyển đổi cần thực hiện để chuyển bản rõ thành bản mã như sau:

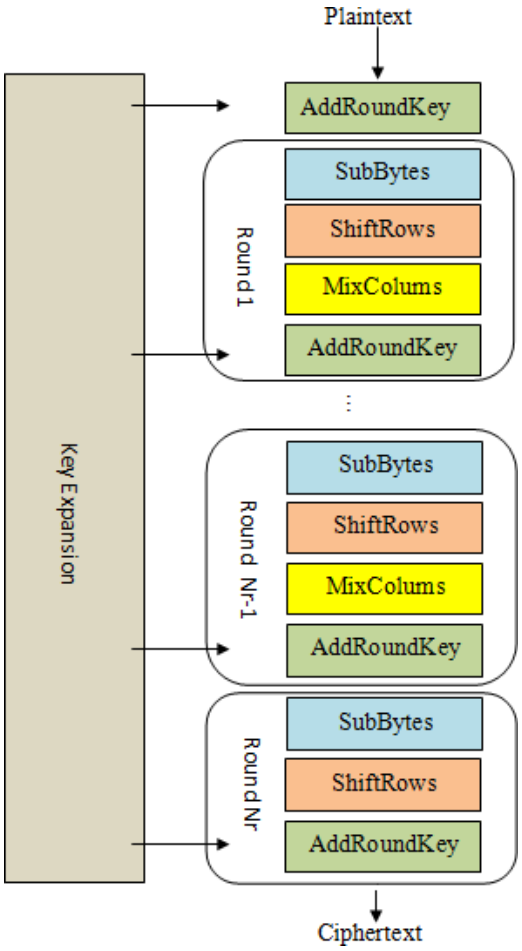
- 10 vòng lặp với khóa 128 bit;

- 12 vòng lặp với khóa 192 bit;
- 14 vòng lặp với khóa 256 bit.

2.2 Mô tả khái quát giải thuật

Giải thuật AES cho mã hóa dữ liệu, như minh họa trên Hình 3.20, gồm các bước xử lý chính như sau:

- Mở rộng khóa (Key Expansion): các khóa vòng (Round key) dùng trong các vòng lặp được sinh ra từ khóa chính AES sử dụng thủ tục sinh khóa Rijndael.



Hình 3.20. Các bước xử lý mã hóa dữ liệu của AES

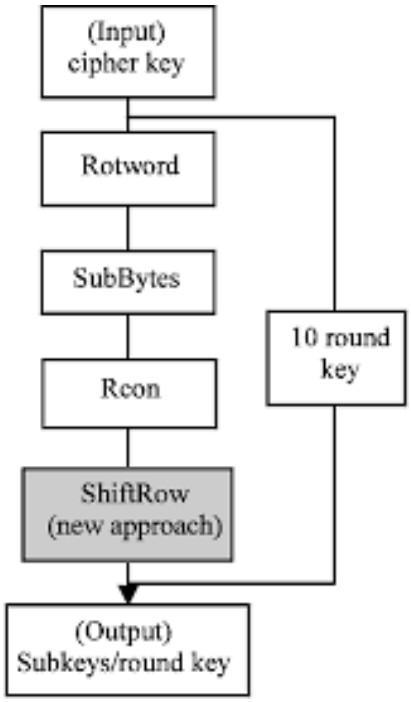
- Vòng khởi tạo (Initial Round): Thực hiện hàm AddRoundKey, trong đó mỗi byte trong *state* được kết hợp với khóa vòng sử dụng phép XOR.
- Các vòng lặp chính (Rounds): Có 4 hàm biến đổi dữ liệu được thực hiện trong mỗi vòng, gồm:
  + SubBytes: hàm thay thế phi tuyến tính, trong đó mỗi byte trong *state* được thay thế bằng một byte khác sử dụng bảng tham chiếu S-box;
  + ShiftRows: hàm đổi chỗ, trong đó mỗi dòng trong *state* được dịch

một số bước theo chu kỳ;

+ MixColumns: trộn các cột trong *state*, kết hợp 4 bytes trong mỗi cột.
+ AddRoundKey.

- Vòng cuối (Final Round): Tương tự các vòng lặp chính, nhưng chỉ thực hiện 3 hàm biến đổi dữ liệu, gồm:
  + SubBytes;
  + ShiftRows;
  + AddRoundKey.

2.3 Mở rộng khóa



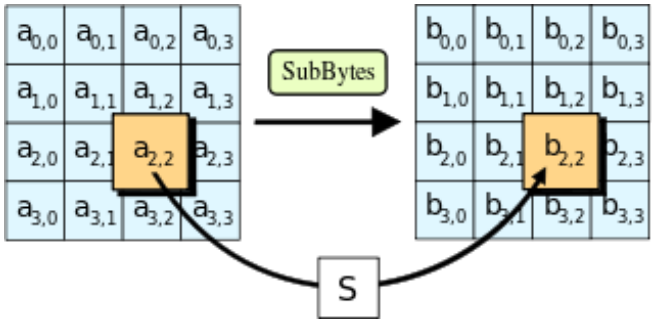
Hình 3.21. Thủ tục sinh khóa Rijndael

Khâu mở rộng khóa AES sử dụng thủ tục sinh khóa Rijndael để sinh các khóa vòng (Round key) cho các vòng lặp xử lý như biểu diễn trên Hình 3.21. Thủ tục Rijndael nhận đầu vào là khóa chính AES (cipher key) và xuất ra một khóa vòng (Subkey/Round key) sau mỗi vòng lặp. Một vòng lặp của thủ tục Rijndael gồm các khâu:

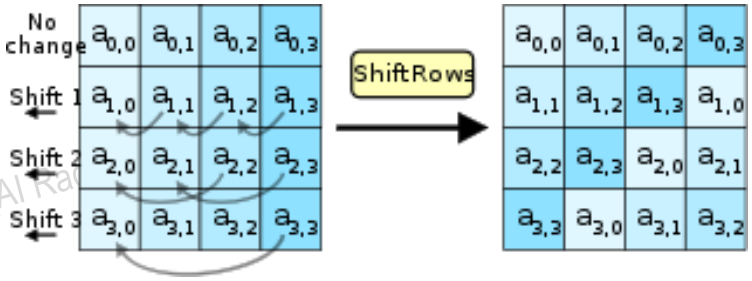
- Rotword: quay trái 8 bit từng từ 32 bit từ khóa gốc;
- SubBytes: thực hiện phép thay thế sử dụng bảng tham chiếu S-box.
- Rcon: tính toán giá trị Rcon(i) = $x^{(i-1)}$ mod $x^8 + x^4 + x^3 + x + 1$
- ShiftRow: thực hiện đổi chỗ tương tự hàm ShiftRows của AES.

2.4 Các hàm xử lý chính

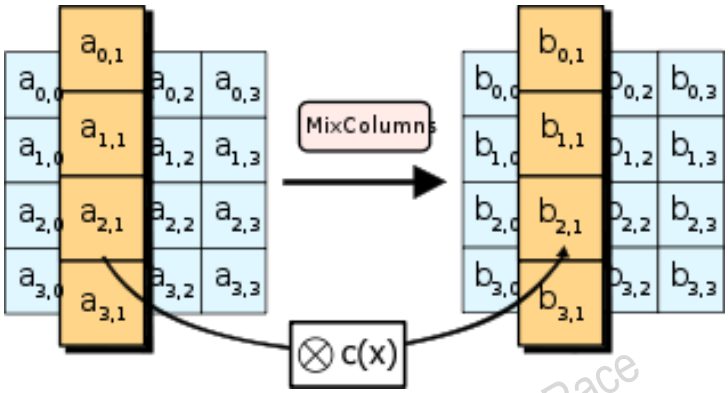
Hàm SubBytes: Mỗi byte trong ma trận *state* được thay thế bởi 1 byte trong Rijndael S-box, hay $b_{ij} = S(a_{ij})$ như minh họa trên Hình 3.22. S-box là một bảng tham chiếu phi tuyến tính, được tạo ra bằng phép nhân nghịch đảo một số cho trước trong trường GF(2^8). Nếu như trong khâu mã hóa S-box được sử dụng thì bảng S-box *đảo* được sử dụng trong khâu giải mã.



Hình 3.22. Hàm SubBytes sử dụng Rijndael S-box



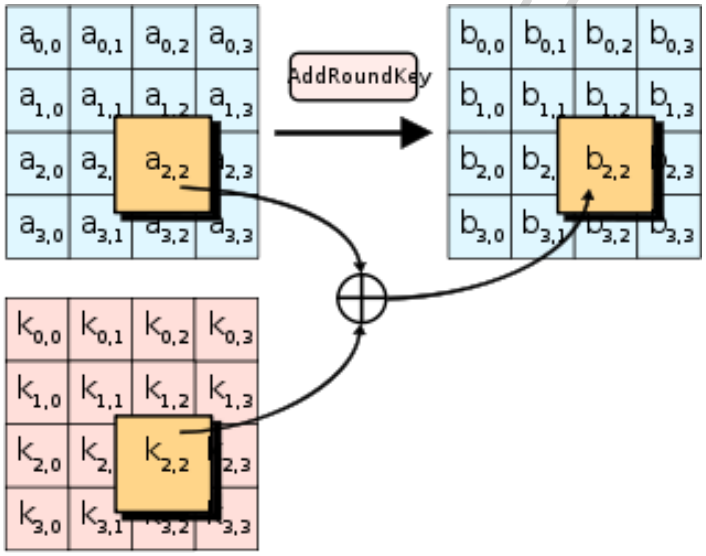
Hình 3.23. Hàm ShiftRows



Hình 3.24. Hàm MixColumns

Hàm ShiftRows: Các dòng của ma trận *state* được dịch theo chu kỳ sang trái theo nguyên tắc: hàng số 0 giữ nguyên, hàng số 1 dịch 1 byte sang trái, hàng số 2 dịch 2 byte và hàng số 3 dịch 3 byte, như minh họa trên Hình 3.23.

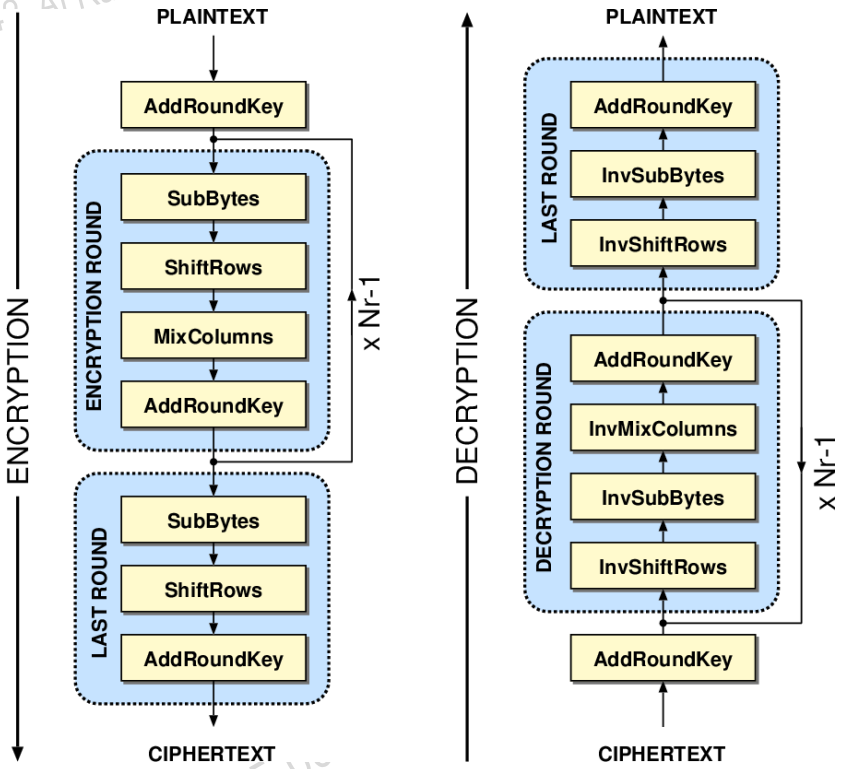
Hàm MixColumns: Mỗi cột của ma trận *state* được nhân với một đa thức c(x), như minh họa trên Hình 3.24. Đa thức $c(x) = 3x^3 + x^2 + x + 2$.



Hình 3.25. Hàm AddRoundKey

Hàm AddRoundKey: Mỗi byte của ma trận *state* được kết hợp với một byte tương ứng của khóa vòng sử dụng phép ⊕ (XOR), như minh họa trên Hình 3.25.

2.5 Giải mã



Hình 3.26. Quá trình mã hóa và giải mã trong AES

Khâu giải mã trong AES cũng gồm các bước xử lý tương tự như khâu mã hóa. Hình

3.26 biểu diễn quá trình mã hóa và giải mã trong AES. Theo đó, ngoài bước Mở rộng khóa, quá trình giải mã gồm Vòng khởi tạo (AddRoundKey), Các vòng lặp chính (Decryption round) và Vòng cuối (Last round) để chuyển khối mã thành khối rõ. Điểm khác biệt chính của khâu giải mã so với khâu mã hóa là các *hàm đảo* được sử dụng, như

các hàm đảo InvSubBytes, InvShiftRows và InvMixColumns tương ứng thay cho các hàm SubBytes, ShiftRows và MixColumns.