**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**---------------------------------------**

****

**BÀI TẬP LỚN**

**MÔN: AN TOÀN VÀ BẢO MẬT THÔNG TIN**

**XÂY DỰNG CHƯƠNG TRÌNH MÃ HÓA VÀ GIẢI MÃ AES (SỬ DỤNG NGÔN NGỮ C++, C#)**

GVHD:

Lớp:

Nhóm

Thành viên nhóm:

Hà Nội- Năm 2024

# LỜI CẢM ƠN

Em xin chân thành cảm ơn các thầy, cô khoa Công nghệ thông tin – Trường Đại học Công Nghiệp Hà Nội đã tận tình dạy dỗ, truyền đạt cho chúng em những kiến thức bổ ích và quý báu trong suốt những năm học đã qua.

Chúng em xin tỏ lòng biết ơn sâu sắc đến thầy Phạm Văn Hiệp, người đã trực tiếp hướng dẫn, giúp đỡ và truyền đạt cho chúng em những kinh nghiệm để đề tài này được thực hiện và hoàn thành.

Chúng em xin cảm ơn gia đình và bạn bè đã động viên giúp đỡ em trong suốt thời gian em làm bài báo cáo bài tập lớn.

Vì thời gian có hạn, trình độ hiểu biết của chúng em còn nhiều hạn chế. Cho nên trong bài báo cáo không tránh được những thiếu sót, em rất mong nhận được sự đống góp ý kiến của cô cũng như bạn bè để bài báo cáo của chúng em được hoàn thiện hơn.

**Em xin chân thành cảm ơn!**

Hà Nội, ngày … tháng … năm 2022

Nhóm sinh viên

# LỜI MỞ ĐẦU

Sự phát triển vượt bậc của công nghệ mạng dẫn đến vấn đề an toàn thông tin trong là rất quan trọng. Có nhiều phương pháp để trao đổi thông tin mật, trong đó phương pháp mã hóa thông tin được coi là xuất hiện sớm nhất, tuy nhiên phương pháp này làm cho người ta dễ phát hiện. Trong đề tài này sẽ sử dụng phương pháp mã hóa AES (advanced encryption standard) để mã hóa thông tin mật. Nội dung báo cáo gồm 3 chương chính sau:

**Chương 1: Tổng quan**

Giới thiệu tổng quan về mã hóa thông tin, phương pháp mã hóa AES.

**Chương 2: Kết quả nghiên cứu**

Thuật toán, sơ đồ thuật toán, ví dụ minh họa của mã hóa AES, đưa ra công cụ hướng dẫn cài đặt và chạy chương trình

**Chương 3: Phần kiến thức lĩnh hội và bài học kinh nghiệm.**

Tổng kết các phần trong bài tập lớn đã làm được,rút ra bài học kinh nghiệm đã rút ra.

# CHƯƠNG 1: TỔNG QUAN VỀ MÃ HÓA AES

* 1. Chuẩn mã nâng cao (AES – Advanced Encryption Standard)

# Giới thiệu

Vào năm 1999, cục tiêu chuẩn quốc gia Hoa Kỳ (NIST) đã ban hành một phiên bản mới của tiêu chuẩn DES chỉ ra rằng DES chỉ nên được sử dụng cho các hệ thống kế cũ và DES ba lần được sử dụng. DES ba lần có hai ưu điểm đảm bảo cho việc sử dụng rộng rãi trong vài năm tới. Đầu tiên, với độ dài khóa 168-bit, nó khắc phục được lỗ hổng đối với cuộc tấn công vét cạn của DES. Thứ hai, thuật toán mã hóa cơ bản trong DES ba lần cũng giống như trong DES. Thuật toán này đã được giám sát kỹ lưỡng hơn bất kỳ thuật toán mã hóa nào khác trong một khoảng thời gian dài và không có cuộc tấn công phá mã hiệu quả nào dựa trên thuật toán thay vì vét cạn được tìm thấy. Do đó, DES ba lần có khả năng chống phá mã rất tốt. Nếu bảo mật là yếu tố duy nhất được xem xét, thì DES ba lần sẽ là lựa chọn thích hợp cho thuật toán mã hóa tiêu chuẩn trong nhiều thập kỷ tới.

Hạn chế chính của DES ba lần là thuật toán tương đối chậm trong phần mềm. DES ban đầu được thiết kế để triển khai bằng phần cứng giữa những năm 1970 và không tạo ra mã phần mềm hiệu quả. DES ba lần, có số vòng gấp ba lần DES, do đó thực hiện chậm hơn DES ban đầu. Một nhược điểm phụ là cả DES và DES ba lần đều sử dụng kích thước khối 64-bit. Vì lý do cả hiệu quả và bảo mật, kích thước khối lớn hơn là cần thiết.

Vì những nhược điểm này, DES ba lần không phải là ứng cử viên thích hợp để sử dụng lâu dài. Để thay thế, vào năm 1997 NIST đã đưa ra lời kêu gọi đề xuất Tiêu chuẩn mã hóa nâng cao (AES) mới, tiêu chuẩn này phải có sức mạnh bảo mật bằng hoặc tốt hơn DES ba lần và cải thiện đáng kể hiệu quả. Ngoài các yêu cầu chung này, NIST quy định rằng AES phải là mật mã khối đối xứng với độ dài khối 128 bit và hỗ trợ độ dài khóa có thể là 128, 192 và 256 bit.

Trong vòng đánh giá đầu tiên, 15 thuật toán được đề xuất đã được chấp nhận. Vòng thứ hai thu hẹp còn 5 thuật toán. NIST đã hoàn thành quá trình đánh giá của mình và xuất bản tiêu chuẩn cuối cùng vào tháng 11 năm 2001. NIST đã chọn Rijndael làm thuật toán AES được đề xuất. Hai nhà nghiên cứu đã phát triển và gửi Rijndael cho AES đều là những nhà mật mã học đến từ Bỉ: Tiến sĩ Joan Daemen và Tiến sĩ Vincent Rijmen.

Cuối cùng, AES được thiết kế để thay thế DES ba lần, nhưng quá trình này sẽ mất một số năm. NIST dự đoán rằng DES ba lần vẫn sẽ là một thuật toán được sử dụng trong tương lai gần.

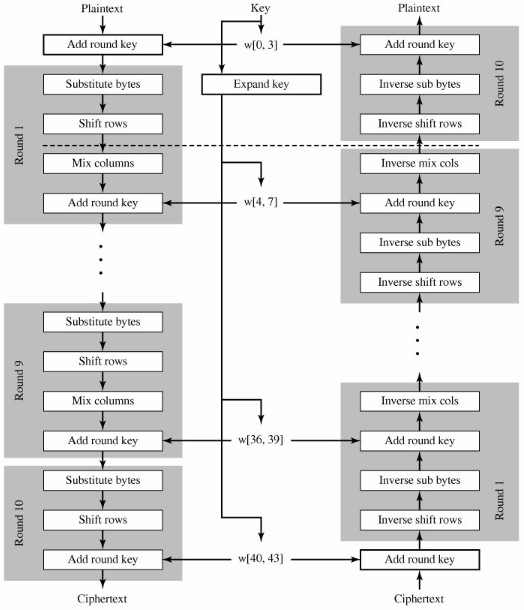
Bảng 4.11 liệt kê tham số của AES tùy thuộc vào kích thước của khóa. Trong phần này ta lựa chọn khóa 128 bits là kích thước thông dụng thường được triển khai trong thực tế.

*Bảng 4.11. Tham số của AES*

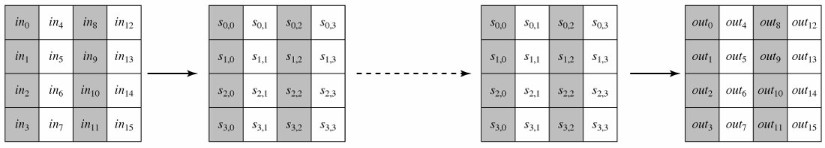
|  |  |  |  |
| --- | --- | --- | --- |
| Kích thước khóa (words/bytes/bits) | 4/16/128 | 6/24/192 | 8/32/256 |
| Kích thước khối của bản rõ (words/bytes/bits) | 4/16/128 | 4/16/128 | 4/16/128 |
| Số vòng | 10 | 12 | 14 |
| Kích thước khóa tại mỗi vòng (words/bytes/bits) | 4/16/128 | 4/16/128 | 4/16/128 |
| Kích thước khóa mở rộng (words/bytes) | 44/176 | 52/208 | 60/240 |

# Mã hóa và giải mã

Cấu trúc tổng thể của mã hóa AES được mô tả trên hình 4.10. Đầu vào cho thuật toán mã hóa và giải mã là một khối 128 bít, khối bít này được mô tả là một ma trận vuông, mỗi ô là 1 byte. Khối này được sao chép vào một mảng trạng thái, được sửa đổi ở mỗi giai đoạn mã hóa hoặc giải mã. Sau giai đoạn cuối cùng, mảng trạng thái này được sao chép vào một ma trận đầu ra. Các hoạt động này được mô tả trong hình 4.11. Tương tự, khóa 128 bit được mô tả như một ma trận vuông, mỗi phần tử là một byte. Khóa này sau đó được mở rộng thành một mảng các từ (word), mỗi từ là bốn byte và tổng chiều dài khóa là 44 từ cho khóa 128 bit như hình 4.12. Lưu ý rằng thứ tự của các byte trong ma trận là theo cột. Vì vậy, bốn byte đầu tiên của bản rõ 128 bit đầu vào chiếm cột đầu tiên của ma trận, bốn byte thứ hai chiếm cột thứ hai, v.v. Tương tự, bốn byte đầu tiên của khóa mở rộng, tạo thành một từ, chiếm cột đầu tiên của ma trận w.



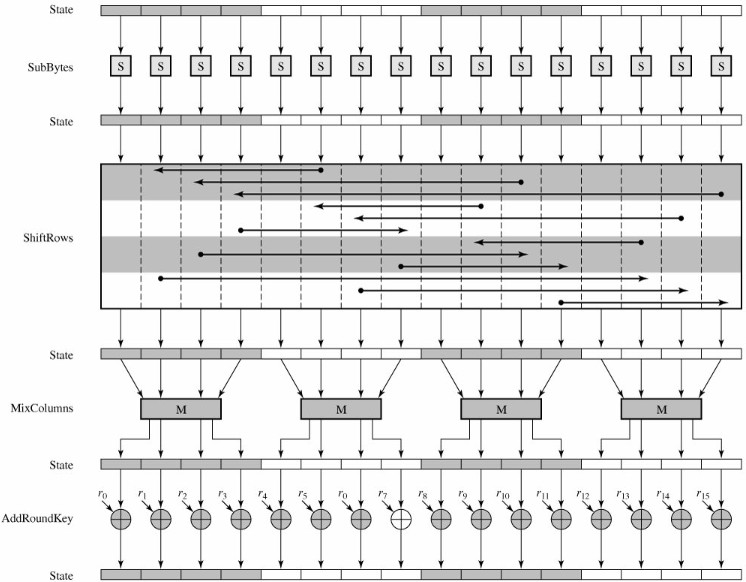
*Hình 4.10. Cấu trúc mã hóa và giải mã AES*



*Hình 4.11. Đầu vào, mảng trạng thái và đầu ra*



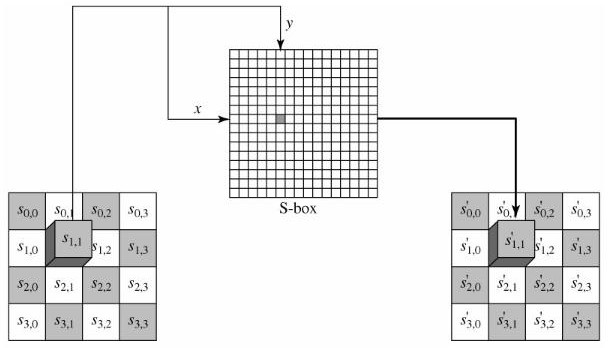
*Hình 4.12. Khóa và mở rộng khóa*

Cấu trúc của thuật toán AES tương đối đơn giản. Cả thuật toán mã hóa và giải mã đều bắt đầu giai đoạn AddRoundKey, tiếp theo là 9 vòng, mỗi vòng đầy đủ 4 giai đoạn: Thay thế các bytes (Substitute bytes) sử dụng hộp S để thực hiện việc thay thế từng byte của khối; dịch các dòng (ShiftRows) đơn giản là thực hiện hoán vị; trộn cột (MixColumns) là phép thay thế sử dụng các phép toán số học trên Z256; AddRoundKey đơn giản chỉ là phép XOR của khối hiện tại với một phần của khóa được mở rộng. Vòng cuối cùng chỉ có 3 giai đoạn. Hình 4.13 minh họa một vòng mã hóa đầy đủ.

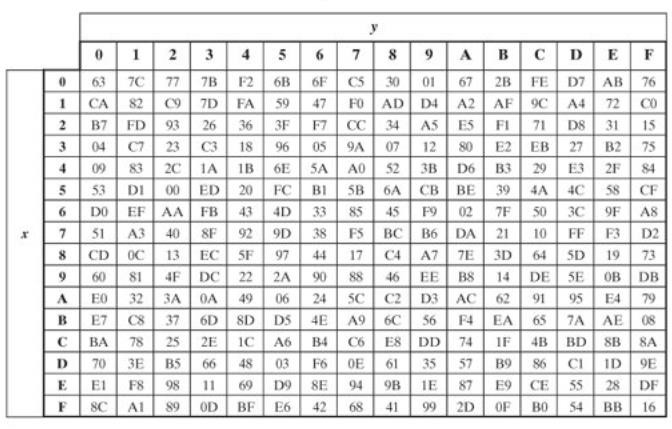
*Hình 4.13. Minh họa một vòng mã AES*

Tiếp theo ta đi xét chi tiết các giai đoạn trong một vòng mã hóa.

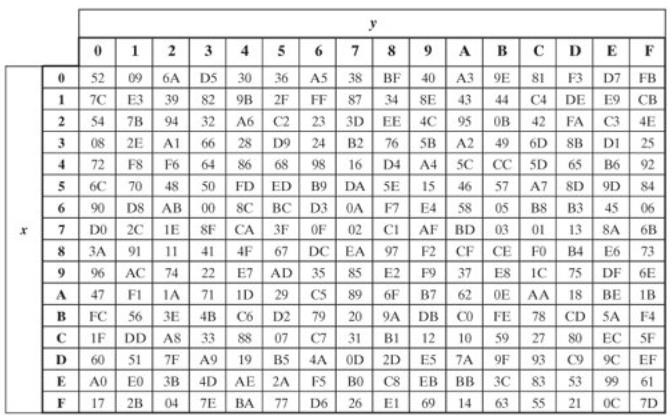
# Thay thế byte

Thay thế byte đơn giản chỉ là tra cứu trong bảng 16 x 16, mỗi ô là 1 byte và được gọi là hộp S như bảng 4.12 và hộp S đảo bảng 4.13. Minh họa việc tra cứu hộp S như hình 4.14.

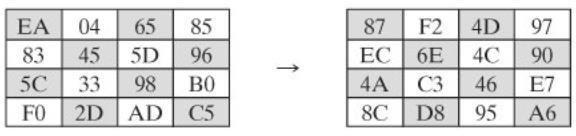
*Hình 4.14. Phép thay thế byte sử dụng hộp S Bảng 4.12. Hộp S*



*Bảng 4.13. Hộp S đảo (inverse S box)*



Ví dụ minh họa phép thay thế byte như hình 4.15



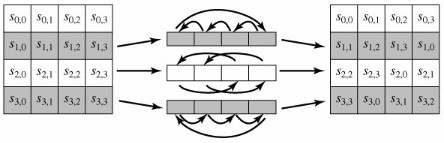
*Hình 4.15. Minh họa phép thay thế byte*

Để tìm byte thay thế của byte EA, ta tra dòng E và cột A trong hộp S thu được byte

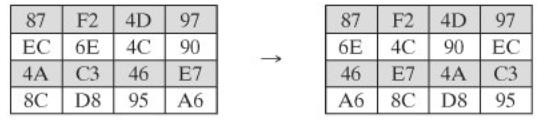
87. Như vậy, byte EA sẽ được thay thế bằng byte 87. Tương tự, byte 04 ta tra dòng 0 và

cột 4 thu được F2. Ta làm tương tự cho các byte còn lại sẽ thu được ma trận kết quả sau khi thực hiện phép thay thế.

# Dịch dòng (Shiftrows)

Hình 4.16 minh họa phép dịch dòng. Dòng đầu tiên của ma trận trạng thái được giữ nguyên, dòng thứ hai quay trái 1 byte, dòng thứ 3 quay trái 2 byte và dòng cuối cùng quay trái 3 byte. Hình 4.17 ví dụ minh họa phép dịch dòng.

*Hình 4.16. Minh họa phép dịch dòng*

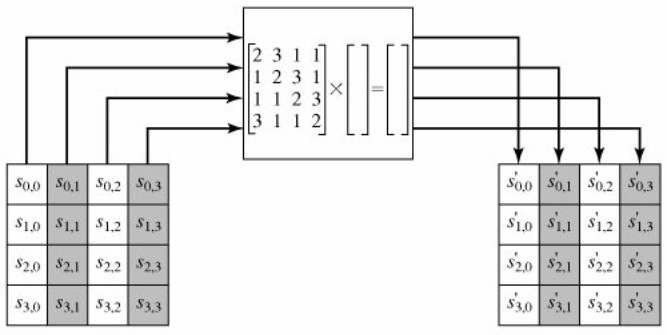


*Hình 4.17. Ví dụ minh họa phép dịch dòng*

Đối với thuật toán giải mã ta sử dụng phép dịch dòng ngược. Tức là, dòng đầu tiên của ma trận trạng thái giữ nguyên, dòng thứ 2 quay phải 1 byte, dòng thứ 3 quay phải 2 bytes và dòng cuối cùng quay phải 3 bytes.

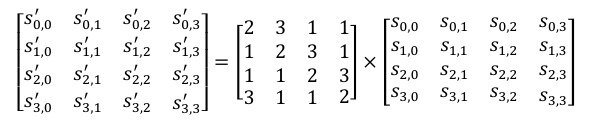
# Trộn cột

Phép trộn cột được thực hiện như minh họa trên hình 4.18.



*Hình 4.18. Minh họa phép trộn cột*

Như vậy, kết quả của phép trộn cột sẽ được xác định theo công thức sau:



Áp dụng phép nhân hai ma trận ta thu được:

𝑠′ = (2 ∙ 𝑠0,𝑗) + (3 ∙ 𝑠1,𝑗) + 𝑠2,𝑗 + 𝑠3,𝑗

0,𝑗

𝑠′ = 𝑠0,𝑗 + (2 ∙ 𝑠1,𝑗) + (3 ∙ 𝑠2,𝑗) + 𝑠3,𝑗

1,𝑗

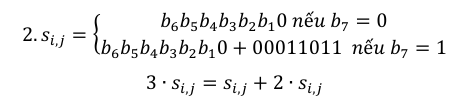
𝑠′ = 𝑠0,𝑗 + 𝑠1,𝑗 + (2 ∙ 𝑠2,𝑗) + (3 ∙ 𝑠3,𝑗)

2,𝑗

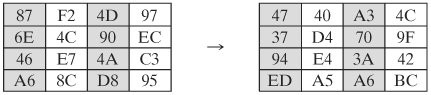
𝑠′ = (3 ∙ 𝑠0,𝑗) + 𝑠1,𝑗 + 𝑠2,𝑗 + (3 ∙ 𝑠3,𝑗)

3,𝑗

Trong đó, phép nhân (.) được thực hiện theo luật sau: Giả sử *si,j* được biểu diễn dưới dạng 8 bít *b7b6b5b4b3b2b1b0* khi nhân với 2 sẽ được thực hiện theo công thức sau:



Phép cộng (+) trong các công thức trên là phép XOR bit.



*Hình 4.9. Minh họa phép trộn cột*

Ta diễn giải cách xác định phần tử đầu tiên trong ma trận sau khi thực hiện phép trộn cột. 𝑠′ = 2 ∙ (87) + 3. (6𝐸) + 46 + 𝐴6.

0,0

Chuyển các số từ hệ 16 sang hệ 2 thu được *87h = 10000111*. Do bít *b7 = 1* nên *2.(87)*

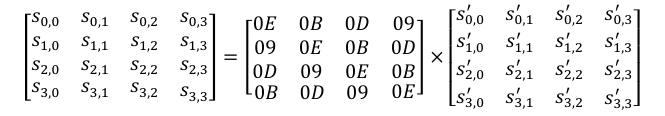
*= 00001110 XOR 00011011 = 00010101, 6Eh = 01101110, 46h = 01000110, A6h = 10100110* và *3.(6E) = 6E + 2.(6E)*. Do bít *b7* của *6E* là 0 nên *2.(6E) = 11011100*. Do đó,

*3.(6E) = 01101110 XOR 11011100 = 10110010*.

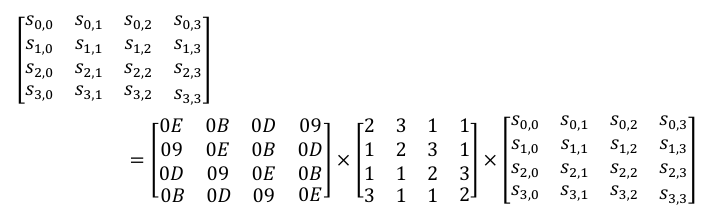
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2.(87) | = | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 3.(6E) | = | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 46 | = | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| A6 | = | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| XOR |  | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 = 47h |

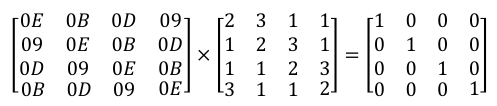
Tính toán tương tự cho các phần tử còn lại ta thu được trạng thái sau khi thực hiện phép trộn cột.

Phép chuyển đổi đảo trộn cột (inverse mix column transform) trong thuật toán giải mã được thực hiện như sau:



Thay thế công thức của phép trộn cột vào thì ta thu được công thức sau.



Như vậy, thì công thức sau phải được thỏa mãn:

Ta chứng minh phần tử đầu tiên thỏa mãn yêu cầu. Thật vậy, 2.(0E) + 0B + 0D + 3.(09) = 00011100 + 00001011 + 00001101 + 3.(09). Trong đó, 3.(09) = 09 + 2.(09) =

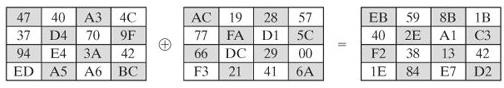
00001001 XOR 00010010 = 00011011. Cuối cùng ta thu được.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2.(0E) | = | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0B | = | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 0D | = | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 3.(09) | = | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| XOR |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 = 1 |

Các phần tử còn lại người đọc tự chứng mình như là một bài tập.

# Cộng với khóa (add round key)

Phép cộng với khóa là thực hiện phép XOR bít của 128 bít của ma trận trạng thái và 128 bít của khóa tương ứng của vòng. Hình 4.10 minh họa ví dụ thực hiện phép cộng khóa, ma trận đầu tiên là trạng thái và ma trận thứ 2 là khóa của vòng.



*Hình 4.10. Ví dụ minh họa phép cộng khóa*

# Mở rộng khóa

Thuật toán mở rộng khóa có đầu vào là 4 từ (16 bytes) khóa và tạo ra một mảng đầu ra 44 từ (176 bytes). Mã giả của thuật toán được mô tả như sau:

KeyExpansion (byte key[16], word w[44])

{

word temp for(i=0;i<4;i++)

w[i] = (key[4\*i], key[4\*i+1], key[4\*i+2], key[4\*i+3]) for(i=4,i<44;i++)

{

temp = w[i-1] if(i mod 4 = 0)

temp = SubWord(RotWord(temp)) XOR Rcon[i/4] w[i] = w[i-4] XOR temp

}

}

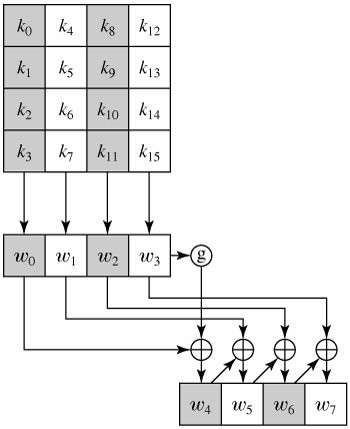
Trong đó, phép toán **RotWord** là thực hiện phép quay trái 1 byte, tức là đầu vào l từ có 4 byte [b0, b1, b2, b3] thì kết quả sau khi thực hiện phép quay trái 1 byte sẽ là [b1, b2, b3, b0]. Phép toán **SubWord** là phép thay thế byte sử dụng bảng S. Hằng số cho mỗi vòng khóa Rcon[j] = (RC[j], 0, 0, 0), với RC[1] = 1, RC[j] = 2.RC[j-1] và phép nhân (.) được thực hiện theo luật như trong thuật toán trộn cột. Giá trị của RC[j] được xác định như bảng

4.14 ở hệ thập lục phân (hexadecimal).

*Bảng 4.14. Giá trị của RC[j]*

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| j | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| RC[j] | 01 | 02 | 04 | 08 | 10 | 20 | 40 | 80 | 1B | 36 |

Hình 4.11 minh họa cách xác định khóa của một vòng.



*Hình 4.11. Minh họa cách xác định khóa của vòng 1*

Ví dụ minh họa cách xác định khóa cho vòng thứ 9 khi khóa tại vòng 8 là EA D2 73 21 B5 8D BA D2 31 2B F5 60 7F 8D 29 2F tương ứng w[32] = [EA, D2, 73, 21], w[33] = [B5, 8D, BA, D2], w[34] = [31, 2B, F5, 60] và w[35] = [7F, 8D, 29, 2F]. Giá trị của khóa

tại vòng 9 được xác định như bảng sau:

*Bảng 4.15. Ví dụ xác định khóa tại vòng 8*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Giá trị i ở hệ thập phân** | **temp** | **Sau khi thực hiện phép RotWord** | **Sau khi thực hiện phép SubWord** | **Rcon(9)** | **Sau khi XOR với Rcon** | **w[i-4]** | **w[i] = temp XOR w[i-4]** |
| 36 | 7F8D292F | 8D292F7F | 5DA515D2 | 1B000000 | 46A515D2 | EAD27321 | AC7766F3 |
| 37 | AC7766F3 | AC7766F3 | AC7766F3 | 1B000000 | AC7766F3 | B58DBAD2 | 19FABC21 |
| 38 | 19FABC21 | 19FABC21 | 19FABC21 | 1B000000 | 19FABC21 | 312BF560 | 28B14941 |
| 39 | 28B14941 | 28B14941 | 28B14941 | 1B000000 | 28B14941 | 7F8D292F | 575C606E |

Như vậy, khóa của vòng 9 sẽ là AC 77 66 F3 19 FA BC 21 28 B1 49 41 57 5C 60 6E.