# Mã hoá AES

**Plaintext của mã hoá AES:**

* Mã hoá AES mã hoá một *Plaintext (bản rõ)* có kích thước khối bằng 128-bit thành một *Ciphertext (bản mờ).*
* Plaintext có thể là bất cứ dạng dữ liệu nào, chẳng hạn như văn bản, hình ảnh, âm thanh,…

**Key của mã hoá AES:**

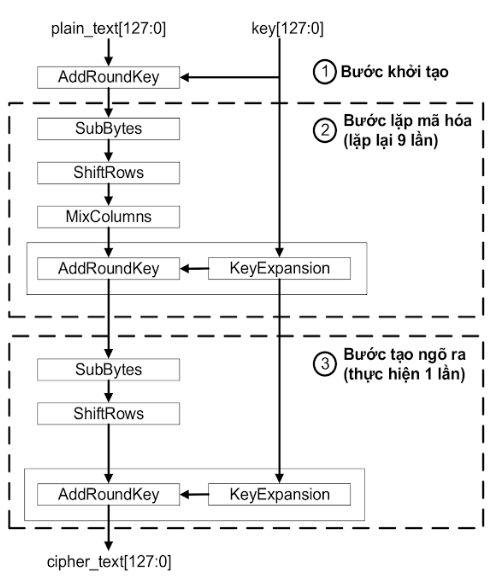
* Key là một giá trị bí mật dùng để mã hoá và giải mã AES
* Trong đó Key có kích thước khác nhau gồm: 128-bit, 192-bit và 256-bit. 🡪 Số vòng lặp của thuật toán lần lượt là 10, 12 và 14.
* Độ bảo mật của quá trình mã hoá phụ thuộc vào độ dài, ngẫu nhiên và tính khó đoán của khoá.
* Quá trình mã hóa trong thuật toán AES được thực hiện thông qua một loạt các Round. Mỗi Round bao gồm các bước nhất định để biến đổi dữ liệu đầu vào.

|  |  |  |  |
| --- | --- | --- | --- |
| Khoá ban đầu (bit) | 128 | 192 | 256 |
| Plaintext (bit) | 128 | 128 | 128 |
| Số vòng lặp | 10 | 12 | 14 |
| Khoá vòng lặp (bit) | 128 | 128 | 128 |
| Khoá mở rộng (Byte) | 176 | 208 | 240 |

1: Mã hoá AES

**Các bước mã hoá AES:**

* + Quá trình mã hoá được thực hiện thông qua 5 chức năng:  **AddRoundKey, SubBytes, ShiftRows, MixColumns và KeyExpansion**.
* Bước 1: Khởi tạo: Khối dữ liệu bản rõ cần mã hoá kết hợp với khoá thông qua *AddRoundKey*.
* Bước 2: Lặp mã hoá: Kết quả của bước 1 được thực thi tuần tự các chức năng *SubBytes, ShiftRows, MixColumns và AddRoundKey*. Bước này được lặp lại (n – 1) lần, trong đó n là số vòng lặp. *KeyExpansion* được thực hiện song song với bước *AddRoundKey* để tạo khoá cho vòng kế tiếp.
* Bước 3: Tạo bản mờ: Ở vòng lặp cuối, kết quả của bước 2 được sử dụng để thực thi tuần tự *SubBytes, ShiftRows và AddRoundKey* để tạo ra bản mờ.



**Plaintext**

**Key**

**1. Bước khởi tạo**

**2. Bước lặp mã hoá**

**3. Bước tạo bản mờ**

**Ciphertext**

2: Sơ đồ mã hoá AES

# ShiftRows

* Phép ShiftRows là một trong các bước của quá trình mã hóa và giải mã trong thuật toán AES.
* Với mục tiêu nhằm phân tán dữ liệu trong khối dữ liệu để ngăn chặn các mẫu lặp lại và cải thiện tính bảo mật của thuật toán.
* Phép ShiftRows lấy kết quả và thực thi sau phép SubBytes

|  |  |  |  |
| --- | --- | --- | --- |
| **D4** | **E0** | **B8** | **1E** |
| **27** | **BF** | **B4** | **41** |
| **11** | **98** | **5D** | **52** |
| **AE** | **F1** | **E5** | **30** |

3: Kết quả của phép SubBytes

* **Các bước thực hiện phép ShiftRows :** được áp dụng trên mỗi hàng của khối dữ liệu

*Theo các quy tắc cụ thể:*

* Hàng đầu tiên không thay đổi.
* Hàng thứ hai được dịch chuyển sang trái một byte.
* Hàng thứ ba được dịch chuyển sang trái hai byte.
* Hàng thứ tư được dịch chuyển sang trái ba byte.

|  |  |  |  |
| --- | --- | --- | --- |
| **27** | **BF** | **B4** | **41** |

|  |  |  |  |
| --- | --- | --- | --- |
| **BF** | **B4** | **41** | **27** |

4: Dịch 1 byte sang trái

|  |  |  |  |
| --- | --- | --- | --- |
| **27** | **BF** | **B4** | **41** |

|  |  |  |  |
| --- | --- | --- | --- |
| **B4** | **41** | **27** | **BF** |

5: Dịch 2 byte sang trái

|  |  |  |  |
| --- | --- | --- | --- |
| **27** | **BF** | **B4** | **41** |

|  |  |  |  |
| --- | --- | --- | --- |
| **41** | **27** | **BF** | **B4** |

6: Dịch 3 byte sang trái

**Thực hiện phép ShiftRows:**

|  |  |  |  |
| --- | --- | --- | --- |
| **D4** | **E0** | **B8** | **1E** |
| **27** | **BF** | **B4** | **41** |
| **11** | **98** | **5D** | **52** |
| **AE** | **F1** | **E5** | **30** |

|  |  |  |  |
| --- | --- | --- | --- |
| **D4** | **E0** | **B8** | **1E** |
| **BF** | **B4** | **41** | **27** |
| **5D** | **52** | **11** | **98** |
| **30** | **AE** | **F1** | **E5** |

Kết quả

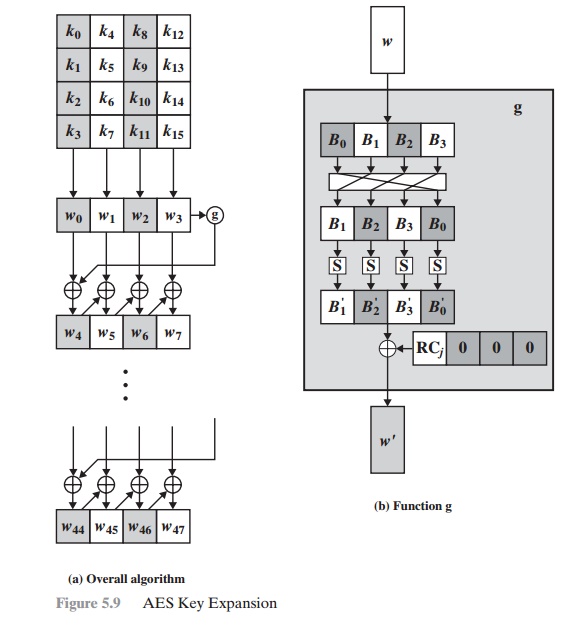
7: Kết quả sau khi thực thi hàm ShiftRows

# KeyExpansion

* Chức năng KeyExpansion là chức năng mở rộng khoá chính sinh ra những khoá con (SubKeys) đây là một chức năng quan trọng để mã hoá và giải mã AES và đảm bảo tính bảo mật của thuật toán.
* Số vòng lập mã hoá cũng thay đổi tuỳ theo độ dài của khoá dẫn đến độ an toàn cũng tăng. Với độ dài khoá là 128-bit, 192-bit, 256-bit thì mỗi lần sử dụng phép KeyExpansion có số vòng lặp lần lượt là 10, 12, 14.

9: Hàm g

8: Qui trình tạo ra khoá con



* Giả sử: *key-128* [[1]](#footnote-1)= 2B 7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C

## Hàm g

* Hàm g xử lý một word 4 byte cuối trong khoá:

|  |  |  |  |
| --- | --- | --- | --- |
| *09* | *CF* | *4F* | *3C* |

* **Rcon (RC)** là một word gồm 4 byte được định nghĩa **Rcon[j] = (RC[j],0,0,0)**.
* Với *RC[1] = 1*, *RC[j] = 2 \* RC[ j - 1 ]*, phép nhân được định nghĩa trên trường GF(28). Các giá trị của *RC[j]* trong cơ số 16.

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

**Các bước thực hiện của hàm g:**

* Bước 1: Dịch sang trái 1 byte.

|  |  |  |  |
| --- | --- | --- | --- |
| *CF* | *4F* | *3C* | *09* |

* Bước 2: Từng byte một được qua một phép **SubBytes.**

|  |  |  |  |
| --- | --- | --- | --- |
| *CF 🡪 8A* | *4F 🡪 84* | *3C 🡪 EB* | 1. 🡪 *01* |

* Bước 3: Sử dụng hàm *XOR* giữa kết quả của bước 2 và giá trị *Rcon[j].* Trong đó, j là số thứ tự vòng lặp của KeyExpainsion.
  + Ví dụ ở vòng lặp thứ 3: RC[3] = 04 🡪 *Rcon[3] = 04 00 00 00*

## Thực thi sinh khoá con

* Bước 1: Chia *key-128* thành 4 word, mỗi word 4 byte.

|  |  |  |  |
| --- | --- | --- | --- |
| 2B 7E 15 16 | 28 AE D2 A6 | AB F7 15 88 | 09 CF 4F 3C |
| W0 | W1 | W2 | W3 |

* Bước 2: Đưa W3 đi qua *hàm g.* Ở vòng lặp thứ 1: *Rcon[1] = 01 00 00 00.*

Ta sẽ có kết quả của hàm g ở vòng lặp KeyExpasion thứ 1:

|  |  |  |  |
| --- | --- | --- | --- |
| 9A | 84 | EB | 01 |

* Bước 3: Sử dụng hàm *XOR* giữa W0 và kết quả *hàm g để ra W4.*

W5 = W1 *xor W4 ;* W6 = W2 *xor W5 ;* W7 = W3 *xor W6.*

|  |  |  |  |
| --- | --- | --- | --- |
| B1 FA FE 17 | 99 54 2C B1 | 32 A3 39 39 | 3B 6C 76 05 |
| W4 | W5 | W6 | W7 |

* W[4] W[5] W[6] W[7] sẽ là khoá K1.
* Những khoá con phía sau sẽ dùng khoá trước nó để biến đổi.

# Cơ sơ toán học và hiệu suất

## Hiệu suất

Tài liệu tham khảo

[[AES] - Lý thuyết về mã hóa AES-128 ~ VLSI TECHNOLOGY (nguyenquanicd.blogspot.com)](https://nguyenquanicd.blogspot.com/2019/09/aes-bai-1-ly-thuyet-ve-ma-hoa-aes-128.html)

1. Key-128: Là một khoá có 128-bit [↑](#footnote-ref-1)