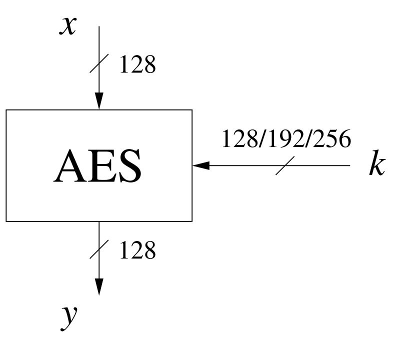
ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Nhập môn An Toàn Thông Tin

The Advanced Encryption Standard (AES)

Hệ AES



AES được xây dựng dựa trên một số phép toán trên trường hữu hạn.

Nội dung

Định nghĩa (Trường)

Một trường F là một tập với các tính chất sau:

- Các phần tử của F tạo thành một nhóm với phép toán $+$ với phần tử đơn vị là 0.
- Các phần tử của F ngoại trừ 0 tạo thành một nhóm với phép toán $\times$ với phần tử đơn vị là 1.
- Các phần tử của F cùng với hai phép toán $+$ và $\times$ thỏa mãn luật phân phối, tức là:

$$a \times (b + c) = (a \times b) + (a \times c), \text{ với mọi } a, b, c \in F.$$

Ví dụ

- Tập số thực $\mathbb{R}$ cùng với hai phép toán $+$ và $\times$ tạo thành một trường.
- Tập $\mathbb{Z}_p$ với hai phép toán $+$ và $\times$ theo modun nguyên tố p là một trường. Trường này có hữu hạn phần tử.

Điều kiện tồn tại trường hữu hạn

Số phần tử của trường F được gọi là ***cấp*** hay ***lực lượng*** của trường F.

Định lý

Tồn tại trường cấp n nếu $n = p^m$ với p là số nguyên tố và m là một số nguyên dương. Số p được gọi là đặc số của trường hữu hạn.

Ví dụ

- Tồn tại trường hữu hạn có 11 phần tử $GF(11)$.
- Tồn tại trường hữu hạn có 81 phần tử $GF(81)$.
- Tồn tại trường hữu hạn có 256 phần tử $GF(2^8)$ (cũng gọi là trường AES).
- Không tồn tại trường với 12 phần tử. Tại sao?

Kiểu trường hữu hạn

$$\begin{tikzcd}
 & GF(p^m) \arrow[dl, "m=1"'] \arrow[dr, "m>1"] & \\
GF(p)\text{: trường nguyên tố } \mathbb{Z}_p & & GF(p^m)\text{: trường mở rộng}
\end{tikzcd}$$

Nhận xét

Hai trường hay được dùng trong mật mã là $GF(p)$ và $GF(2^m)$.

Trường mở rộng $GF(2^m)$

Các phần tử của $GF(2^m)$ là các đa thức:

$$a_{m-1}x^{m-1} + \cdots + a_1x + a_0 = A(x) \in GF(2^m)$$

với $a_i \in GF(2) = \{0, 1\}$.

Ví dụ

Các phần tử của trường $GF(2^3) = GF(8)$ là các đa thức

$$A(x) = a_2x^2 + a_1x + a_0$$

$GF(2^3)$ có $2^3 = 8$ phần tử:

$$GF(2^3) = \{ \begin{array}{llll} 0, & 1, & x, & x+1 \\ x^2, & x^2+1, & x^2+x, & \\ x^2+x+1 & & & \end{array} \}$$

Câu hỏi

Làm thế nào để tính toán $(+, -, \times, /)$ trên $GF(2^m)$?

Tính toán như trên đa thức thông thường với các hệ số được tính trên $GF(p)$.

Cộng và trừ

Ví dụ (Tính toán trên $GF(2^3)$)

$$\begin{aligned} A(x) &= x^2 + x + 1 \\ B(x) &= x^2 + 1 \\ A(x) + B(x) &= x \end{aligned}$$

Nhân

Ví dụ (Tính toán trên $GF(2^3)$)

$$\begin{aligned} A(x) &= x^2 + x + 1 \\ B(x) &= x^2 + 1 \\ A(x) \times B(x) &= (x^2 + x + 1)(x^2 + 1) \\ &= x^4 + x^3 + x + 1 \notin GF(2^3) \end{aligned}$$

Nhắc lại: với trường nguyên tố $GF(7) = \{0, 1, \ldots, 6\}$

$$3 \cdot 4 = 12 = 5 \mod 7$$

Phép nhân trên trường mở rộng

Ta sẽ chia lấy dư cho một đa thức bất khả quy, là đa thức tương tự như số nguyên tố.

Định nghĩa (Phép nhân trên trường mở rộng $GF(2^m)$)

Xét $A(x), B(x) \in GF(2^m)$ và xét

$$P(x) = \sum_{i=0}^{m} p_i x^i, \qquad p_i \in GF(2)$$

là một đa thức bất khả quy. Phép nhân của hai phần tử $A(x), B(x)$ có kết quả là

$$C(x) = A(x) \cdot B(x) \mod P(x).$$

Phép nhân trên trường mở rộng

Ví dụ (Tính toán trên $GF(2^3)$)

$$\begin{aligned} A(x) &= x^2 + x + 1 \\ B(x) &= x^2 + 1 \\ A(x) \times B(x) &= (x^2 + x + 1)(x^2 + 1) \\ &= x^4 + x^3 + x + 1 \notin GF(2^3) \end{aligned}$$

Ta chọn đa thức bất khả quy là

$$P(x) = x^3 + x + 1$$

Khi đó

$$A(x) \cdot B(x) = x^2 + x \mod P(x)$$

vì $(x^4 + x^3 + x + 1)/(x^3 + x + 1) = x + 1$; và dư $x^2 + x$.

Đa thức bất khả quy

- Không phải mọi đa thức đều bất khả quy. Ví dụ,

$$x^4 + x^3 + x + 1 = (x^2 + x + 1)(x^2 + 1)$$

  không phải là bất khả quy.
- Trong AES, người ta sử dụng đa thức bất khả quy

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

Trường hữu hạn trong SageMath

https://cocalc.com

- Trường hữu hạn $K = GF(2^3)$ với mo đun $P(x) = x^3 + x + 1$:

```
K.<x> = GF(2^3, name='x', modulus=x^3 + x +1)
```

- Phép nhân trong SageMath:

```
A=x^2 + x + 1
B=x^2 + 1
C=A*B
print (C)
```


Phần tử nghịch đảo trong mở rộng trường

- Phần tử nghịch đảo $A^{-1}(x)$ của $A(x) \in GF(2^m)$ là phần tử thỏa mãn

$$A(x) \cdot A^{-1}(x) = 1 \mod P(x)$$

- Tương tự như trong trường nguyên tố, phần tử nghịch đảo có thể tính dùng thuật toán Euclid mở rộng.

```
sage: K.<x> = GF(2^3, name='x', modulus=x^3 + x +1)
sage: A=x^2 + x + 1
sage: A^-1
x^2
sage: x^2 * A
1
```

Trường AES trong SageMath

```
sage: K.<x>=GF(2^8, name='x', modulus=x^8+x^4+x^3+x+1)
sage: (x^7+x^6+x)^-1
x^5 + x^3 + x^2 + x + 1
sage: (x^7+x^6+x)*(x^5+x^3+x^2+x+1)
1
```


X \ Y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00	01	8D	F6	CB	52	7B	D1	E8	4F	29	C0	B0	E1	E5	C7
1	74	B4	AA	4B	99	2B	60	5F	58	3F	FD	CC	FF	40	EE	B2
2	3A	6E	5A	F1	55	4D	A8	C9	C1	0A	98	15	30	44	A2	C2
3	2C	45	92	6C	F3	39	66	42	F2	35	20	6F	77	BB	59	19
4	1D	FE	37	67	2D	31	F5	69	A7	64	AB	13	54	25	E9	09
5	ED	5C	05	CA	4C	24	87	BF	18	3E	22	F0	51	EC	61	17
6	16	5E	AF	D3	49	A6	36	43	F4	47	91	DF	33	93	21	3B
7	79	B7	97	85	10	B5	BA	3C	B6	70	D0	06	A1	FA	81	82
8	83	7E	7F	80	96	73	BE	56	9B	9E	95	D9	F7	02	B9	A4
9	DE	6A	32	6D	D8	8A	84	72	2A	14	9F	88	F9	DC	89	9A
A	FB	7C	2E	C3	8F	B8	65	48	26	C8	12	4A	CE	E7	D2	62
B	0C	E0	1F	EF	11	75	78	71	A5	8E	76	3D	BD	BC	86	57
C	0B	28	2F	A3	DA	D4	E4	0F	A9	27	53	04	1B	FC	AC	E6
D	7A	07	AE	63	C5	DB	E2	EA	94	8B	C4	D5	9D	F8	90	6B
E	B1	0D	D6	EB	C6	0E	CF	AD	08	4E	D7	E3	5D	50	1E	B3
F	5B	23	38	34	68	46	03	8C	DD	9C	7D	A0	CD	1A	41	1C

Bảng trên tính nghịch đảo trong $GF(2^8)$. Ví dụ, nghịch đảo của

$$x^7 + x^6 + x = (11000010)_2 = (C2)_{hex} = (XY)$$

được cho bởi ô tại dòng C, cột 2:

$$(2F)_{hex} = (00101111)_2 = x^5 + x^3 + x^2 + x + 1.$$

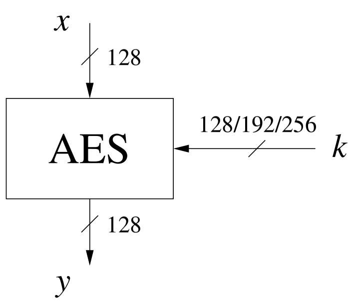
Nội dung

1. Trường hữu hạn
2. AES
3. Giải mã AES

Sơ lược lịch sử

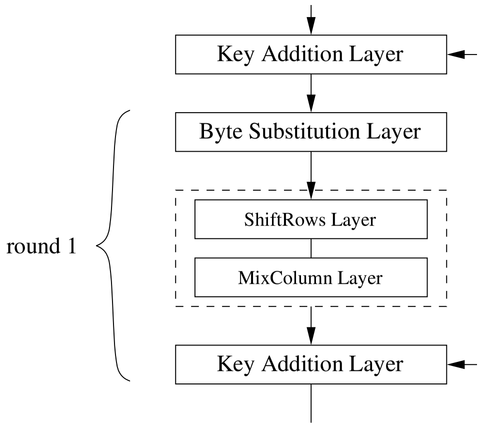
- 1997: Kêu gọi đề xuất Chuẩn mã hóa nâng cao AES bởi NIST
- 1998: Có 15 thuật toán đề xuất
- Tháng 8 năm 1999: Chọn 5 thuật toán vào vòng cuối
  1. Mars bởi IBM
  2. RC6 bởi RSA Laboratories
  3. Rijndael bởi Joan Daemen và Vincent Rijmen
  4. Serpent bởi Ross Anderson, Eli Biham và Lars Knudsen
  5. Twofish bởi Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall và Neils Ferguson
- Tháng 10 năm 2000: ***Rijndael*** đã được chọn làm AES

AES



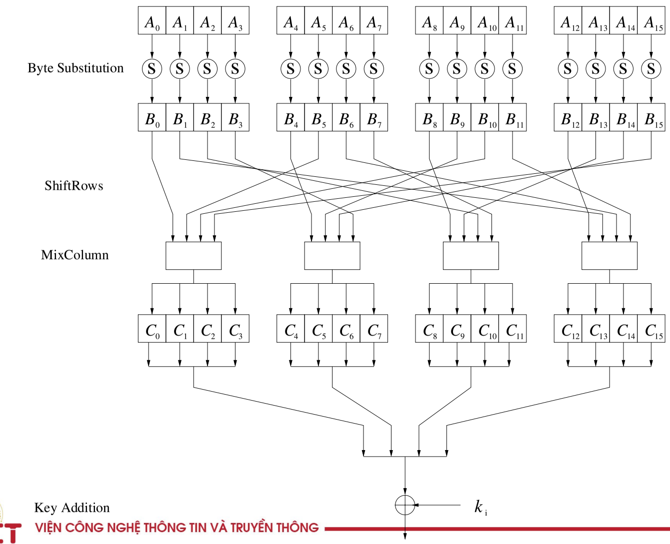
K	Số vòng
128	10
192	12
256	14

Cấu trúc một vòng của AES

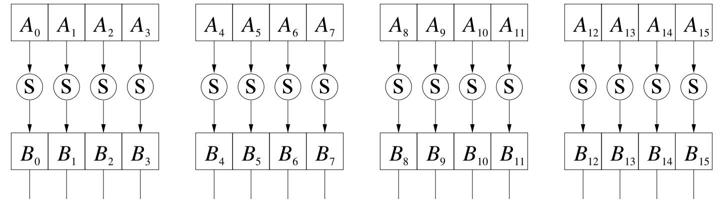


- Chú ý: Riêng vòng cuối cùng không có thao tác MixColumn.

1 vòng: 128 bit được tách thành 16 bytes $(16 \times 8 = 128)$

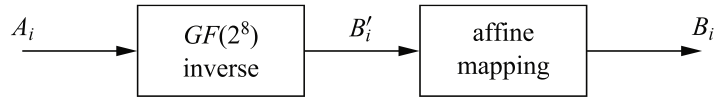


Byte Substitution layer



Hình: Dùng 16 S-box giống nhau: $S(A_i) = B_i$

- Hỏi: Bảng S-box được xây dựng thế nào?
- Trả lời: Coi $A_i \in GF(2^8)$ và tính nghịch đảo $A_i^{-1} = B_i'$; sau đó đưa qua một phép biến đổi tuyến tính.



Phép biến đổi tuyến tính $B_i' \to B_i$

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix} \equiv \begin{pmatrix} 1&0&0&0&1&1&1&1 \\ 1&1&0&0&0&1&1&1 \\ 1&1&1&0&0&0&1&1 \\ 1&1&1&1&0&0&0&1 \\ 1&1&1&1&1&0&0&0 \\ 0&1&1&1&1&1&0&0 \\ 0&0&1&1&1&1&1&0 \\ 0&0&0&1&1&1&1&1 \end{pmatrix} \begin{pmatrix} b_0' \\ b_1' \\ b_2' \\ b_3' \\ b_4' \\ b_5' \\ b_6' \\ b_7' \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \bmod 2.$$

Ví dụ

- Giả sử input

$$A_i = (11000010)2 = (C2)_{hex} \quad \in GF(2^8)$$

- Ta có

$$A_i^{-1} = B_i' = (2F) = (00101111)_2 \qquad \in GF(2^8)$$

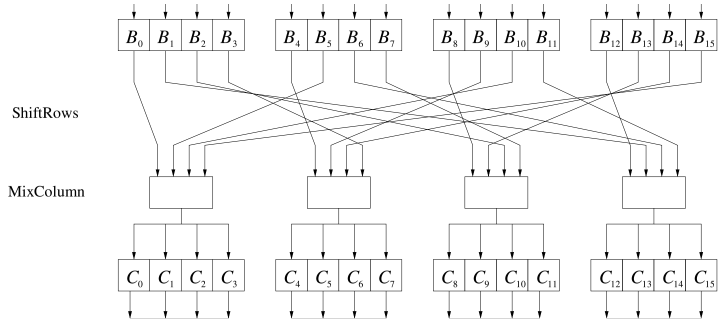
- Qua phép biến đổi tuyến tính ta được

$$B_i = (0010\ 0101)_2 = (25)_{hex}$$

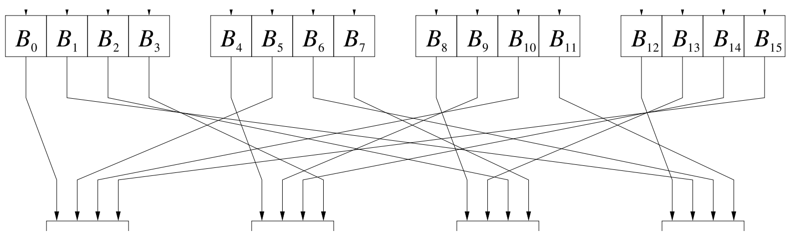
S-box: $S((C2)_{hex}) = S(C, 2) = (25)_{hex}$.

x \ y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Diffusion Layer = Shift Rows và Mix Column



Shift Rows



B_0	B_4	B_8	B_{12}
B_1	B_5	B_9	B_{13}
B_2	B_6	B_{10}	B_{14}
B_3	B_7	B_{11}	B_{15}

$\rightarrow$

B_0	B_4	B_8	B_{12}
B_5	B_9	B_{13}	B_1
B_{10}	B_{14}	B_2	B_6
B_{15}	B_3	B_7	B_{11}

Mix Column

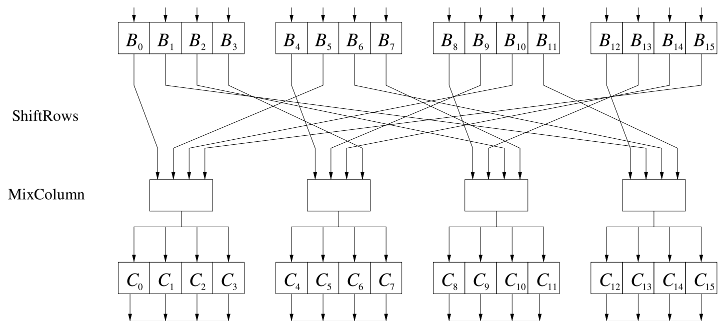
- Là một phép biến đổi tuyến tính biến đổi

$$\text{MixColumn}(B) = C$$

- Mỗi cột 4 byte được xem như một vector, và được nhân với một ma trận cố định trước.
- Phép cộng và phép nhân các hệ số được thực hiện trong $GF(2^8)$.
- Ma trận của phép biến đổi tuyến tính MixColumn là

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix}$$

Ví dụ



$$\begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} B_0 \\ B_5 \\ B_{10} \\ B_{15} \end{pmatrix}$$

Ví dụ

- Giả sử input của MixColumn là

$$B = (25, 25, \ldots, 25).$$

- Do ma trận MixColumn, ta chỉ cần tính toán theo đa thức trong $GF(2^8)$ với $02 \cdot 25$ và $03 \cdot 25$:

$$\begin{aligned} 02 \cdot 25 &= x \cdot (x^5 + x^2 + 1) \\ &= x^6 + x^3 + x, \\ 03 \cdot 25 &= (x+1) \cdot (x^5 + x^2 + 1) \\ &= (x^6 + x^3 + x) + (x^5 + x^2 + 1) \\ &= x^6 + x^5 + x^3 + x^2 + x + 1 \end{aligned}$$

Ví dụ (tiếp)

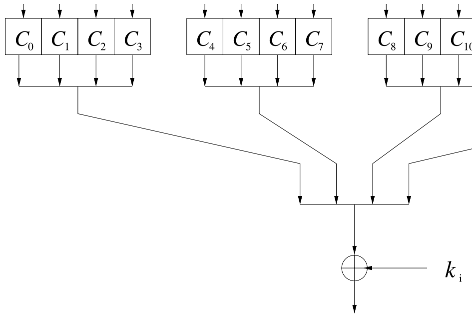
- Thực hiện phép cộng trong $GF(2^8)$ ta được kết quả của C:

$$\begin{array}{rcccc} 01 \cdot 25 = & & x^5+ & x^2+ & 1 \\ 01 \cdot 25 = & & x^5+ & x^2+ & 1 \\ 02 \cdot 25 = & x^6+ & x^3+ & x & \\ 03 \cdot 25 = & x^6+ & x^5+x^3 & x^2+x+ & 1 \\ \hline C_i = & & x^5+ & x^2+ & 1 \end{array}$$

- Vậy output của C là:

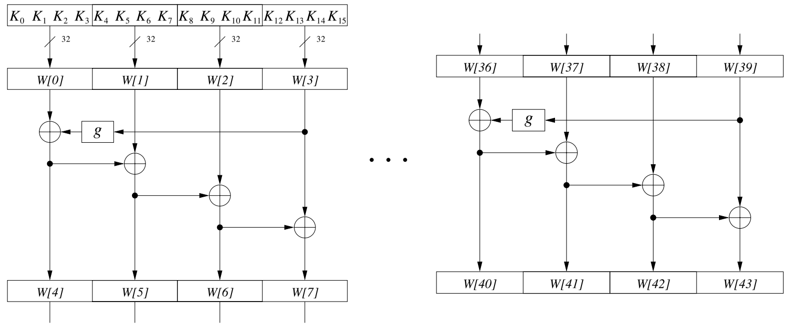
$$C = (25, 25, \ldots, 25).$$

Key Addition Layer



- Input: Gồm 16-byte ma trận C và 16-byte khóa con k_i
- Output: $C \oplus k_i$
- Các khóa con được sinh trong thủ tục mở rộng khóa (Key schedule).

Key schedule cho AES với khóa 128 bit

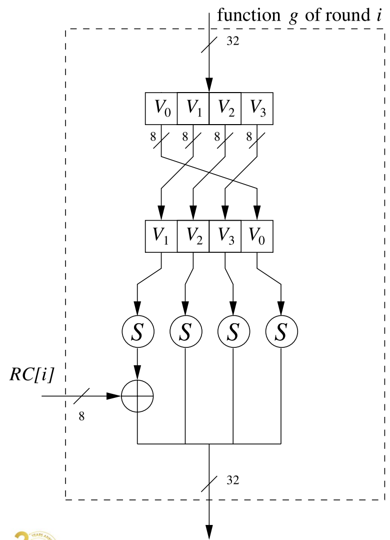


- Có 10 vòng và 11 lần Key Addition Layer; mỗi Key Addition Layer cần khóa con 128 bit;
- Các khóa con này được chia thành $W[0]$, $W[1]$, …, $W[43]$, và được tính (trên $GF(2^8)$) bởi

$$W[4i] = W[4(i-1)] + g(W[4i-1]).$$

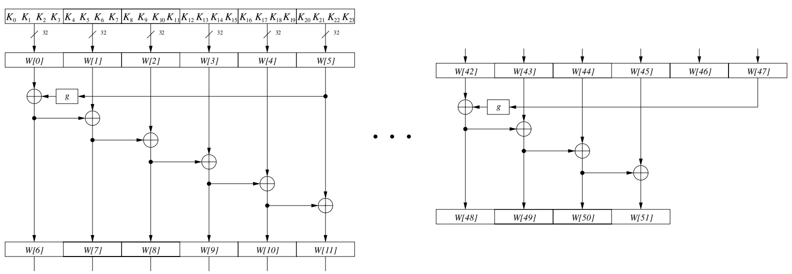
$$W[4i+j] = W[4i+j-1] + W[4(i-1)+j]$$

Hàm g ở vòng thứ i sử dụng $RC[i]$



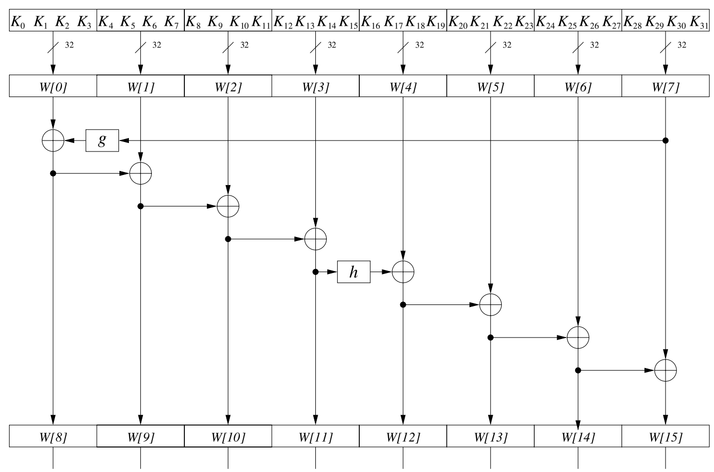
$$RC[1] = x^0 = (00000001)_2,$$
$$RC[2] = x^1 = (00000010)_2,$$
$$RC[3] = x^2 = (00000100)_2,$$
$$\vdots$$
$$RC[10] = x^9 = (00110110)_2.$$

AES-192: Key schedule

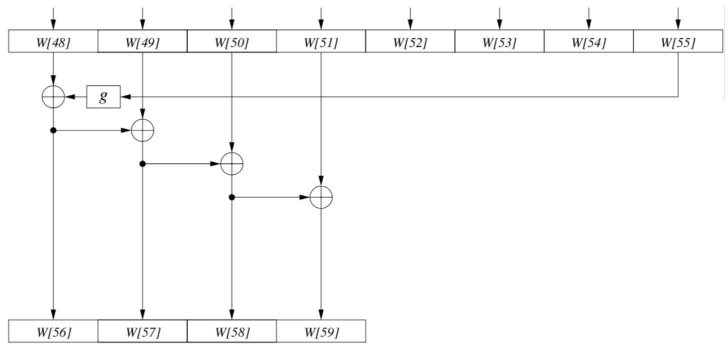


- AES với 192 bit có 12 vòng, vậy cần 13 Key Addition layer.
- Mỗi Key Addition Layer cần 128 bit khóa;
- Vậy cần 52 khóa con $W[0], \ldots, W[51]$ mỗi khoá 32 bit $= 4$ byte $(4 \times 13 = 52)$.

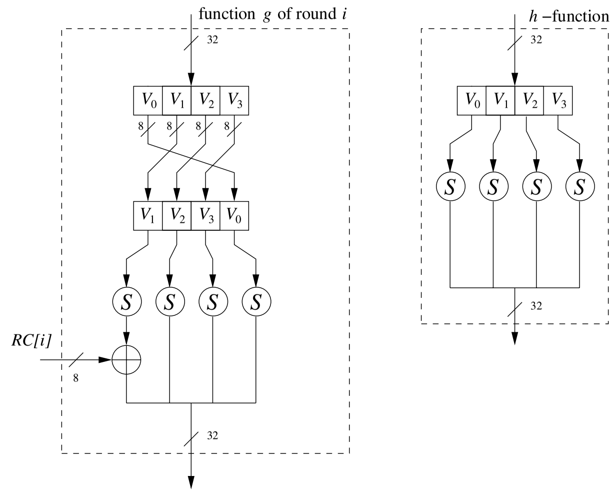
AES-256: Key schedule vòng 1



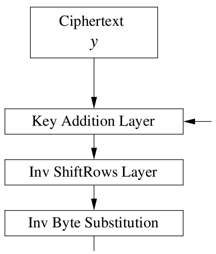
AES-256: Key schedule vòng cuối



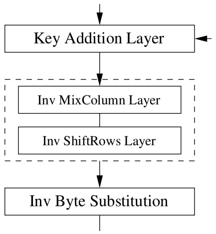
AES256: hàm g và h



Sơ đồ giải mã

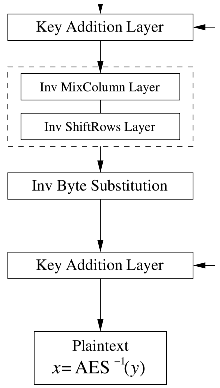


Hình: Vòng n



Hình: Vòng $n-1$

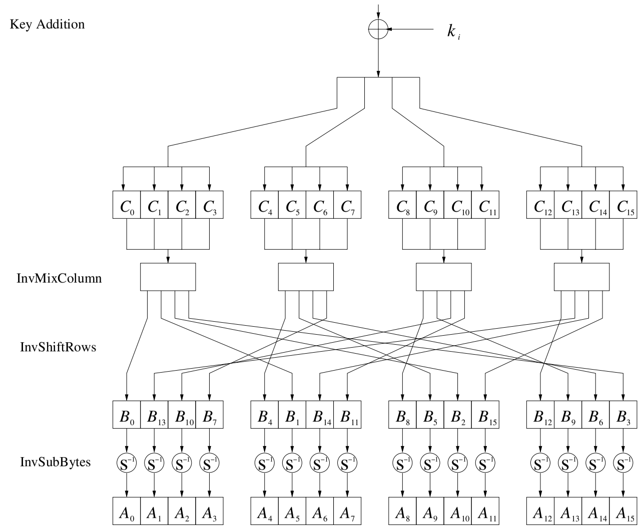
. . .



Hình: Vòng 1

Nội dung

1. Trường hữu hạn
2. AES
3. Giải mã AES



Hình: Mỗi vòng trong sơ đồ giải mã

InvMixColumn: Hàm ngược của MixColumn

- Là phép biến đổi ngược của MixColumn

$$\text{InvMixColumn}(C) = B$$

- Phép cộng và phép nhân các hệ số được thực hiện trong $GF(2^8)$;
- Ma trận của phép biến đổi tuyến tính InvMixColumn là

$$\begin{pmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{pmatrix}$$

InvSubBytes: Hàm ngược của SubBytes

- Ta nhắc lại phép toán SubBytes:

$$A_i \longrightarrow \boxed{\begin{array}{c}GF(2^8)\\ \text{inverse}\end{array}} \xrightarrow{B_i'} \boxed{\begin{array}{c}\text{affine}\\ \text{mapping}\end{array}} \longrightarrow B_i$$

- Để tính InvSubBytes, ta tính ngược lại:

$$B_i \to B_i' \to A_i$$

InvSubByte: $B_i' \to A_i$

- Ta có $A_i = (B_i')^{-1} \in GF(2^8)$
- Ví dụ, nghịch đảo của

$$(2F)_{hex} = (00101111)_2 = x^5 + x^3 + x^2 + x + 1.$$

là

$$x^7 + x^6 + x = (11000010)_2 = (C2)_{hex}$$

InvShiftRows: Hàm ngược của ShiftRows

B_0	B_4	B_8	B_{12}
B_1	B_5	B_9	B_{13}
B_2	B_6	B_{10}	B_{14}
B_3	B_7	B_{11}	B_{15}

$\longrightarrow$

B_0	B_4	B_8	B_{12}
B_{13}	B_1	B_5	B_9
B_{10}	B_{14}	B_2	B_6
B_7	B_{11}	B_{15}	B_3

InvSubBytes: Biến đổi tuyến tính ngược $B_i \to B_i'$

$$\begin{pmatrix} b_0' \\ b_1' \\ b_2' \\ b_3' \\ b_4' \\ b_5' \\ b_6' \\ b_7' \end{pmatrix} \equiv \begin{pmatrix} 0&1&0&1&0&0&1&0 \\ 0&0&1&0&1&0&0&1 \\ 1&0&0&1&0&1&0&0 \\ 0&1&0&0&1&0&1&0 \\ 0&0&1&0&0&1&0&1 \\ 1&0&0&1&0&0&1&0 \\ 0&1&0&0&1&0&0&1 \\ 1&0&1&0&0&1&0&0 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} \bmod 2,$$

Key Schedule của AES^{-1}

- Ở vòng thứ n của AES^{-1}, ta cần khóa con cuối cùng,
- Ở vòng thứ $n-1$ của AES^{-1}, ta cần khóa con trước khóa con cuối,$\cdots$
- Tóm lại, ta cần tính các khóa con theo thứ tự ngược lại. Ví dụ, với AES^{-1}, thứ tự ta cần là

$$(W[40], W[41], W[42], W[43]) \rightarrow (W[0], W[1], W[2], W[3])$$

- Trên thực tế, ta sẽ tính trước toàn bộ
  11 khóa con (nếu cho AES-128),
  13 khóa con (nếu cho AES-192), hoặc
  15 khóa con (nếu cho AES-256) và lưu lại.

Bảng tính InvSubBytes

x \ y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

SOICT
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

soict.hust.edu.vn/ fb.com/groups/soict