TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN ĐIỆN TỬ - VIỄN THÔNG

BỘ MÔN ĐIỆN TỬ HÀNG KHÔNG VŨ TRỤ

Môn học:

LÝ THUYẾT MẬT MÃ

Giảng viên: PGS.TS. Đỗ Trọng Tuấn

Email: dotrongtuan@gmail.com

Mục tiêu học phần

Cung cấp kiến thức cơ bản về mật mã đảm bảo an toàn và bảo mật thông tin:

- ✓ Các phương pháp mật mã khóa đối xứng; Phương pháp mật mã khóa công khai;
- ✓ Các hệ mật dòng và vấn đề tạo dãy giả ngẫu nhiên;
- ✓ Lược đồ chữ ký số Elgamal và chuẩn chữ ký số ECDSA;
- ✓ Độ phức tạp xử lý và độ phức tạp dữ liệu của một tấn công cụ thể vào hệ thống mật mã;
- ✓ Đặc trưng an toàn của phương thức mã hóa;
- ✓ Thám mã tuyến tính, thám mã vi sai và các vấn đề về xây dựng hệ mã bảo mật cho các ứng dụng.

Nội Dung

1. Chương 1. Tổng quan
2. Chương 2. Mật mã khóa đối xứng
3. Chương 3. Hệ mật DES
4. **Chương 4. Hệ mật AES**
5. Chương 5. Dãy giả ngẫu nhiên và hệ mật dòng
6. Chương 6. Kỹ thuật quản lý khóa

Tài liệu tham khảo

1. A. J. Menezes, P. C. Van Oorschot, S. A. Vanstone, *Handbook of applied cryptography*, CRC Press 1998.
2. B. Schneier, *Applied Cryptography*. John Wiley Press 1996.
3. M. R. A. Huth, *Secure Communicating Systems*, Cambridge University Press 2001.
4. W. Stallings, *Network Security Essentials*, *Applications and Standards,* Prentice Hall. 2000.

Nhiệm vụ của Sinh viên

1. Chấp hành nội quy lớp học
2. Thực hiện đầy đủ bài tập
3. Nắm vững ngôn ngữ lập trình Matlab

Chương 4. Hệ mật AES

4.1. Giới thiệu sơ lược hệ mật AES
4.2. Cấu trúc hệ mật AES
4.3. Mở rộng bộ khóa hệ mật AES
4.4. Cách triển khai hệ mật AES
4.5. Thám mã hệ mật AES

4.1. Sơ lược hệ mật AES

The Advanced Encryption Standard (AES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST) in December 2001.

In February 2001, NIST announced that a draft of the Federal Information Processing Standard (FIPS) was available for public review and comment. Finally, AES was published as FIPS 197 in the Federal Register in December 2001.

4.1. Sơ lược hệ mật AES

The Advanced Encryption Standard (AES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST) in December 2001.

The criteria defined by NIST for selecting AES fall into three areas:

1. Security

2. Cost

3. Implementation.

4.2. Cấu trúc hệ mật AES

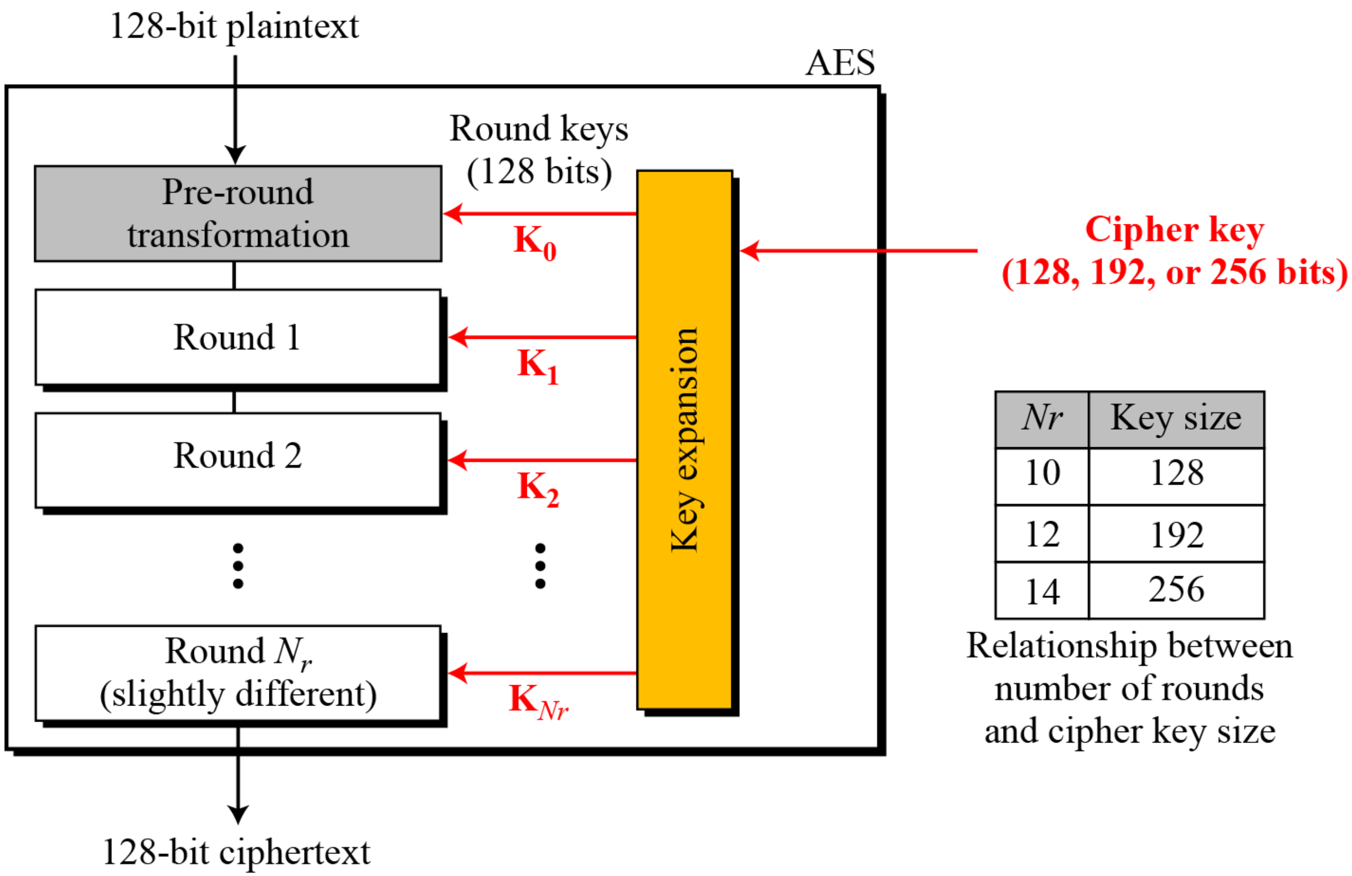
AES is a non-Feistel cipher that encrypts and decrypts a data block of 128 bits. It uses 10, 12, or 14 rounds. The key size, which can be 128, 192, or 256 bits, depends on the number of rounds.

AES has defined three versions, with 10, 12, and 14 rounds.

Each version uses a different cipher key size (128, 192, or 256), **but the round keys are always 128 bits**.

4.2. Cấu trúc hệ mật AES

General design of AES encryption cipher

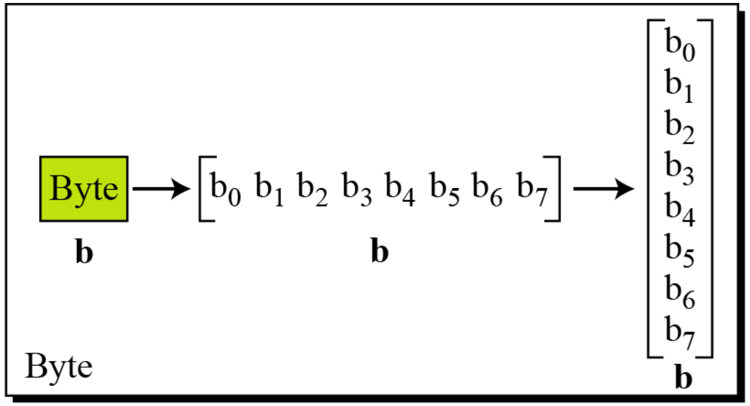


Nr	Key size
10	128
12	192
14	256

Relationship between number of rounds and cipher key size

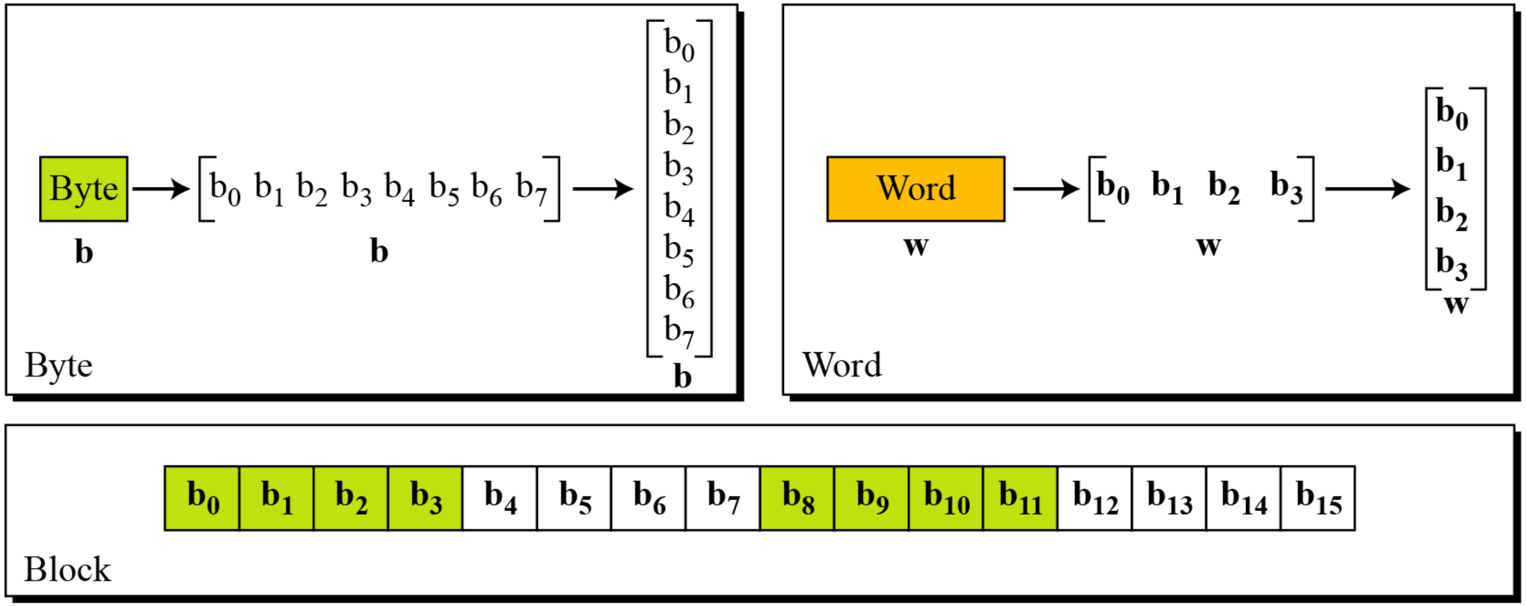
4.2. Cấu trúc hệ mật AES

Data units used in AES



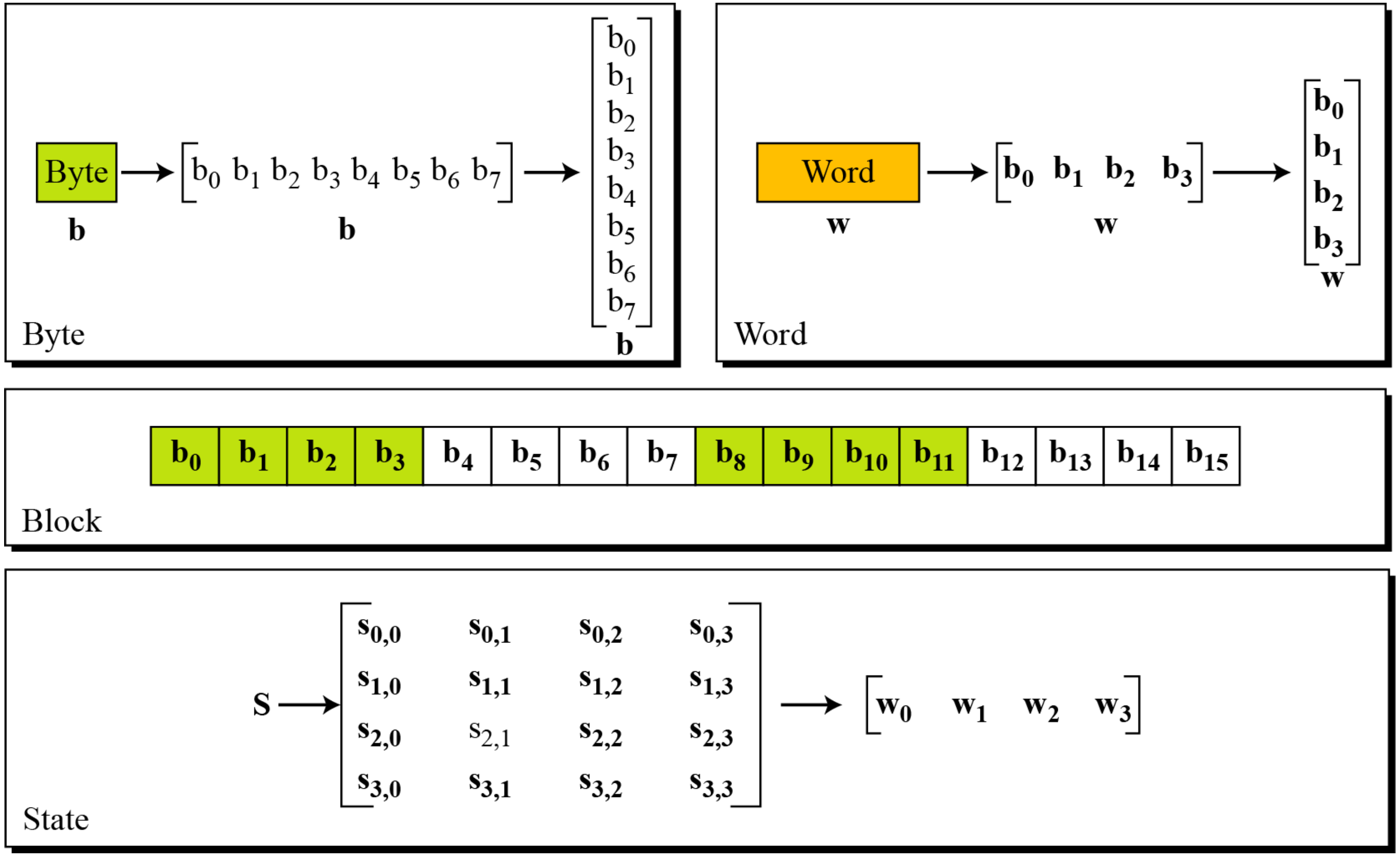
4.2. Cấu trúc hệ mật AES

Data units used in AES



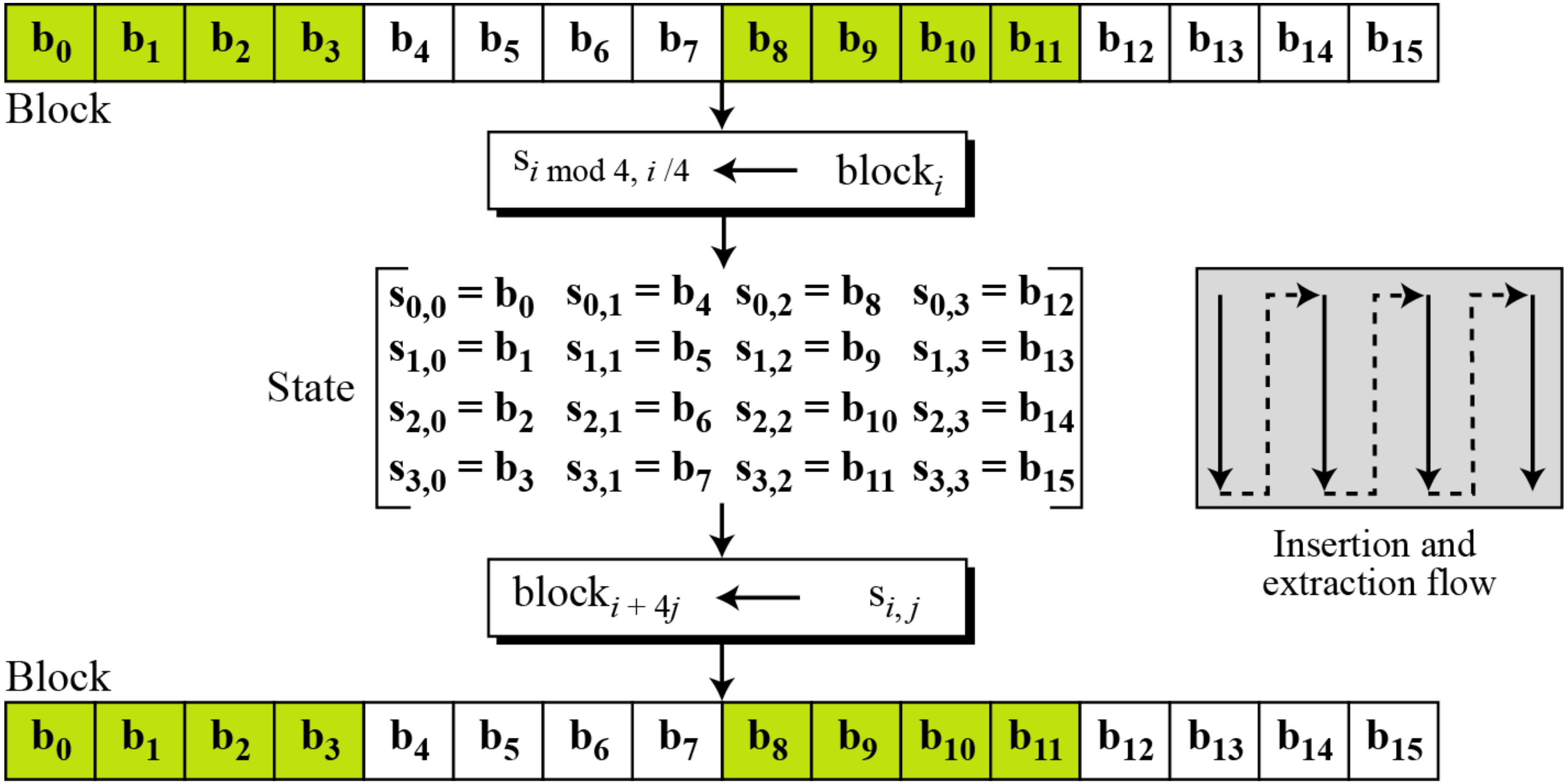
4.2. Cấu trúc hệ mật AES

Data units used in AES



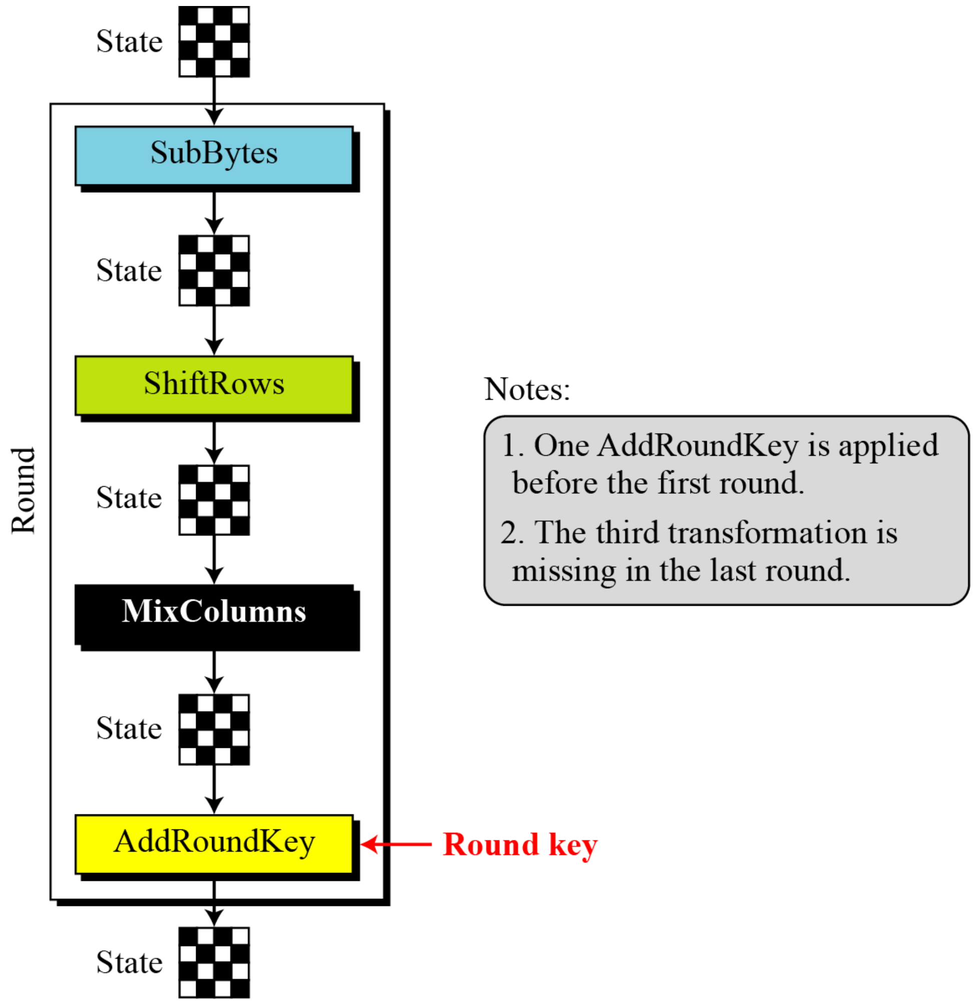
4.2. Cấu trúc hệ mật AES

Block-to-state and state-to-block transformation



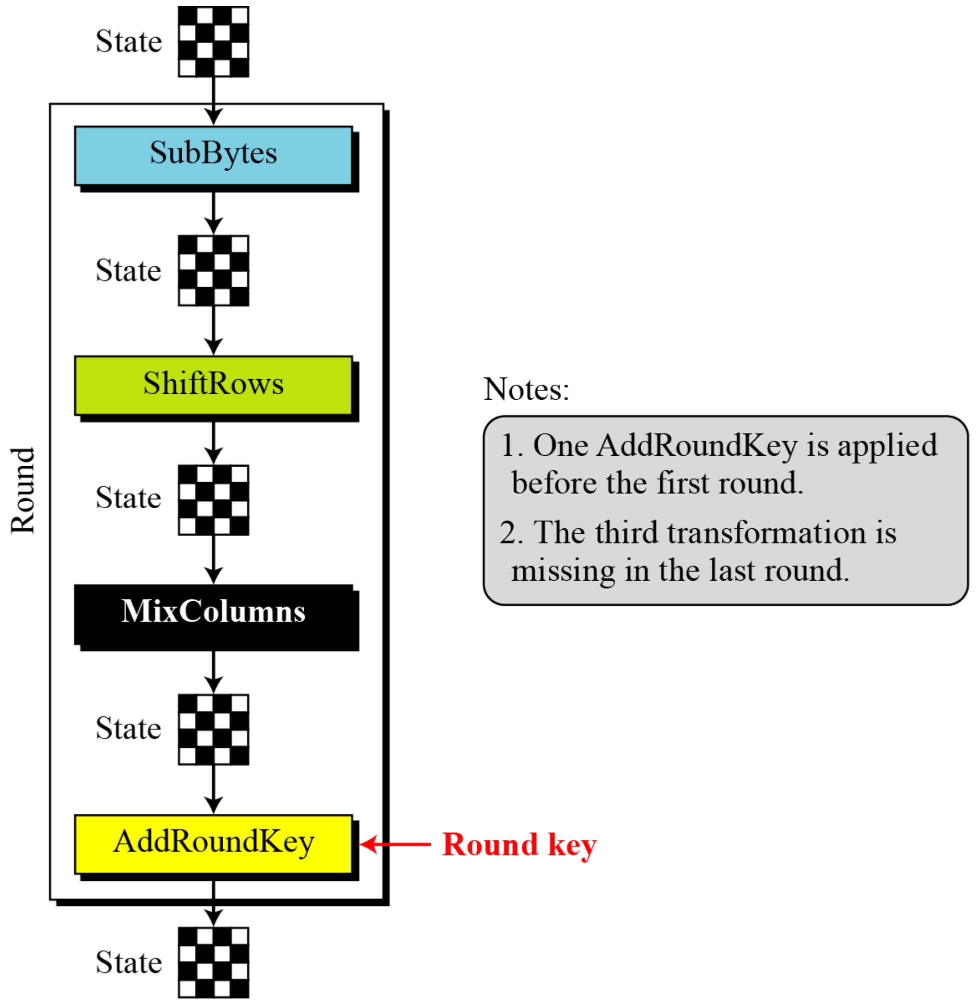
4.2. Cấu trúc hệ mật AES

Structure of each round at the encryption site



4.2. Cấu trúc hệ mật AES

To provide security, AES uses four types of transformations: substitution, permutation, mixing, and key-adding.



4.2. Cấu trúc hệ mật AES

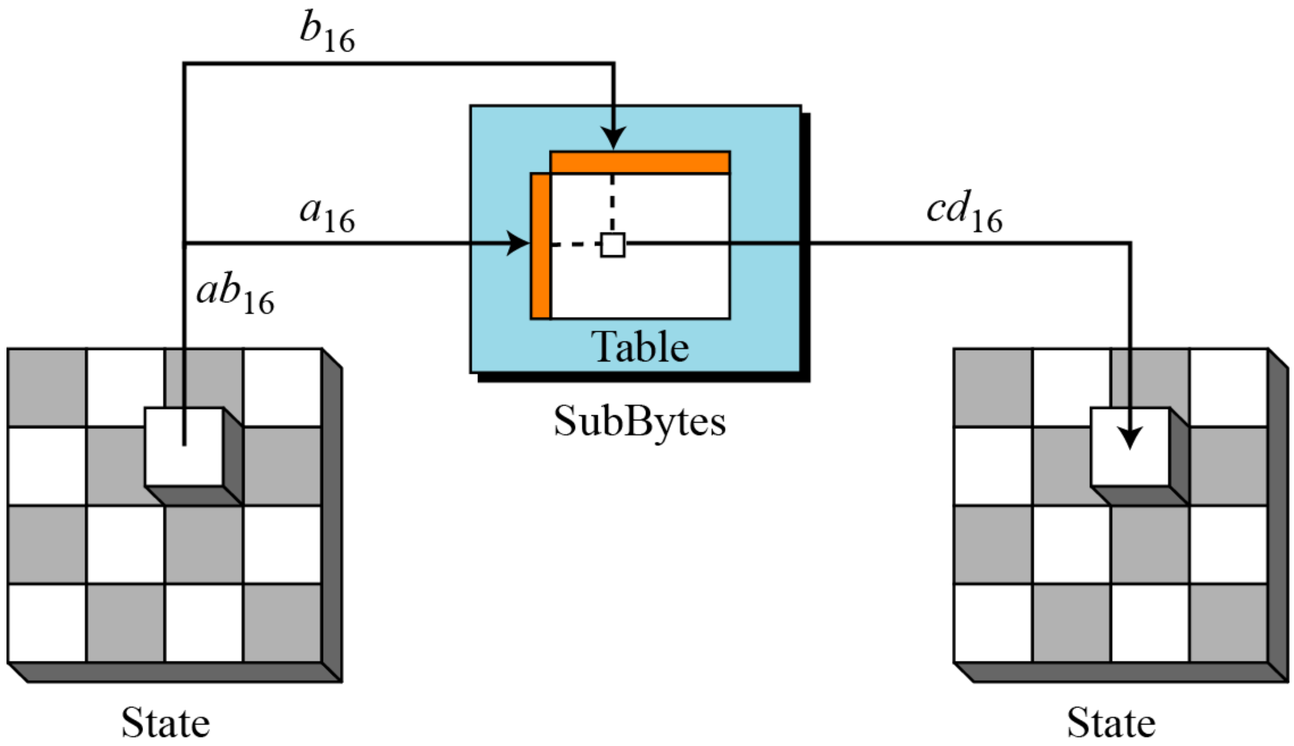
AES, like DES, uses substitution. AES uses two invertible transformations.

SubBytes

The first transformation, SubBytes, is used at the encryption site. To substitute a byte, we interpret the byte as two hexadecimal digits.

The SubBytes operation involves 16 independent byte-to-byte transformations.

4.2. Cấu trúc hệ mật AES



4.2. Cấu trúc hệ mật AES

SubBytes table

	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	CB	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

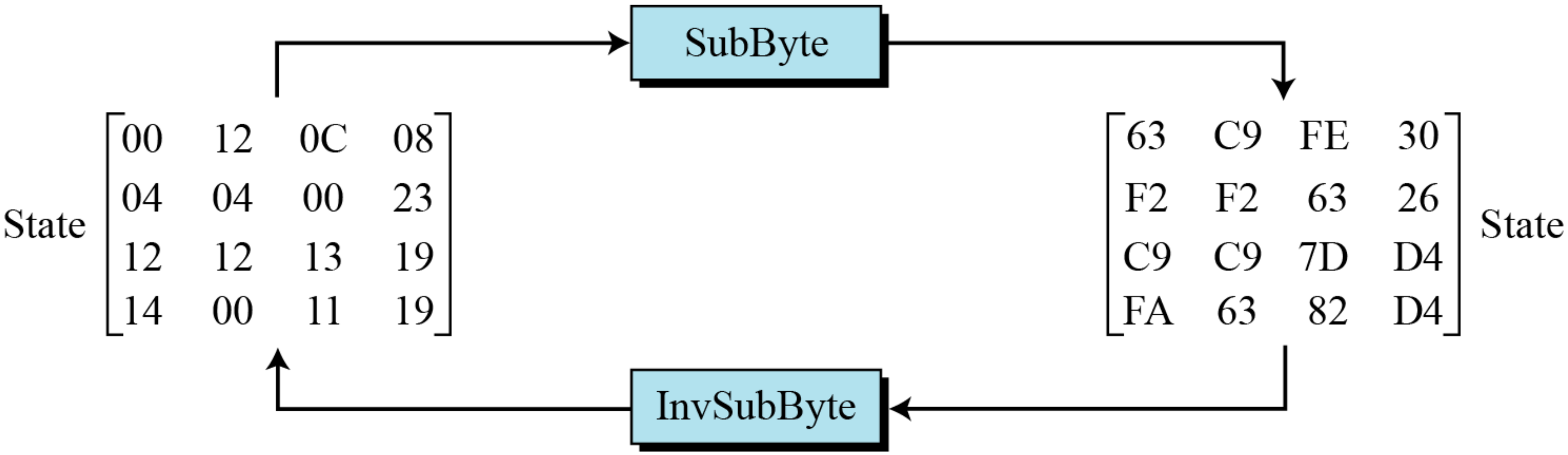
4.2. Cấu trúc hệ mật AES

invSubBytes table

	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

4.2. Cấu trúc hệ mật AES

This figure shows how a state is transformed using the SubBytes transformation. The figure also shows that the InvSubBytes transformation creates the original one. Note that if the two bytes have the same values, their transformation is also the same.



4.2. Cấu trúc hệ mật AES

Transformation Using the GF(2^8) Field

AES also defines the transformation algebraically using the GF(2^8) field with the irreducible polynomials $(x^8 + x^4 + x^3 + x + 1)$.

subbyte: $\rightarrow \quad \mathbf{d} = \mathbf{X}\,(s_{r,c})^{-1} \oplus \mathbf{y}$

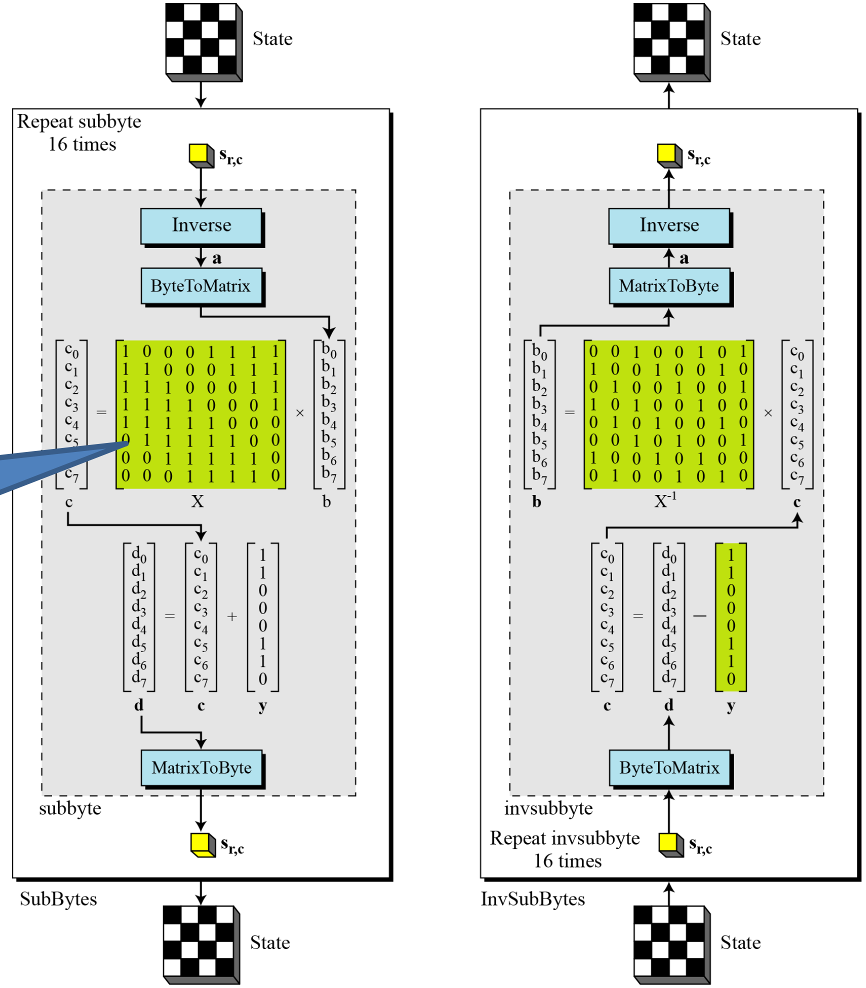
invsubbyte: $\rightarrow \quad [\mathbf{X}^{-1}(\mathbf{d} \oplus \mathbf{y})]^{-1} = [\mathbf{X}^{-1}(\mathbf{X}\,(s_{r,c})^{-1} \oplus \mathbf{y} \oplus \mathbf{y})]^{-1} = [(s_{r,c})^{-1}]^{-1} = s_{r,c}$

The SubBytes and InvSubBytes transformations are inverses of each other.

4.2. Cấu trúc hệ mật AES

SubBytes and InvSubBytes processes

Constant matrix
X



4.2. Cấu trúc hệ mật AES

SubBytes (S)

$\{$

for $(r = 0 \text{ to } 3)$

for $(c = 0 \text{ to } 3)$

$S_{r,c} = \text{subbyte}(S_{r,c})$

$\}$

subbyte (byte)

$\{$

$a \leftarrow \text{byte}^{-1}$ *// Multiplicative inverse in GF(2^8) with inverse of 00 to be 00*

ByteToMatrix (a, **b**)

for $(i = 0 \text{ to } 7)$

$\{$

$\mathbf{c}_i \leftarrow \mathbf{b}_i \oplus \mathbf{b}_{(i+4)\bmod 8} \oplus \mathbf{b}_{(i+5)\bmod 8} \oplus \mathbf{b}_{(i+6)\bmod 8} \oplus \mathbf{b}_{(i+7)\bmod 8}$

$\mathbf{d}_i \leftarrow \mathbf{c}_i \oplus \text{ByteToMatrix}(0x63)$

$\}$

MatrixToByte (**d**, d)

$\text{byte} \leftarrow d$

$\}$

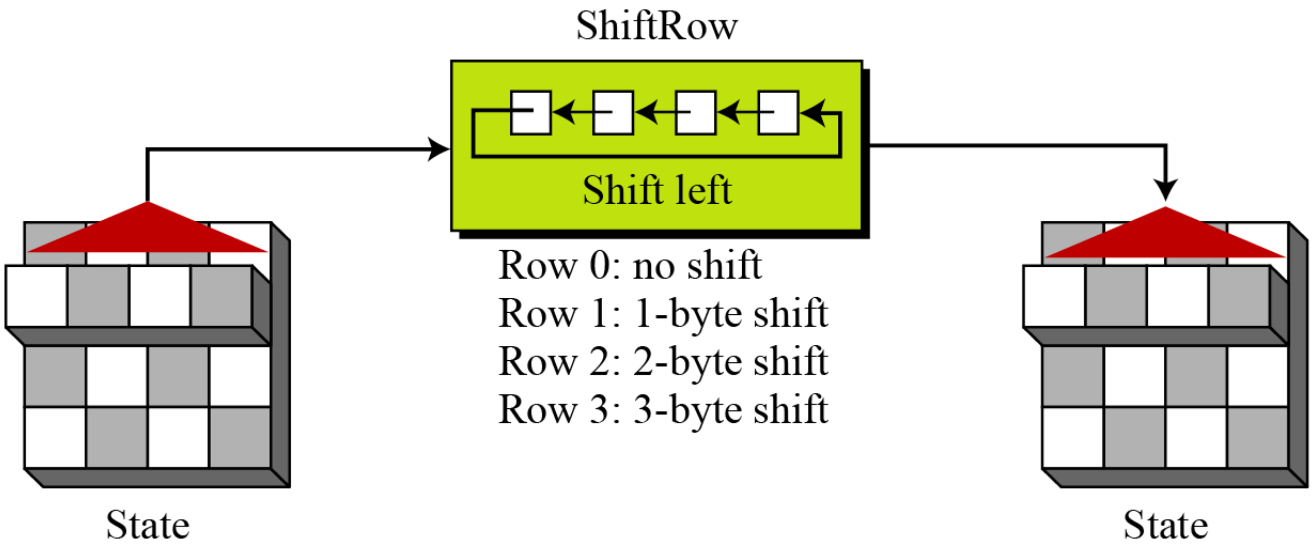
4.2. Cấu trúc hệ mật AES

Another transformation found in a round is shifting, which permutes the bytes.

ShiftRows

In the encryption, the transformation is called ShiftRows.

ShiftRows transformation = Permutation



4.2. Cấu trúc hệ mật AES

InvShiftRows

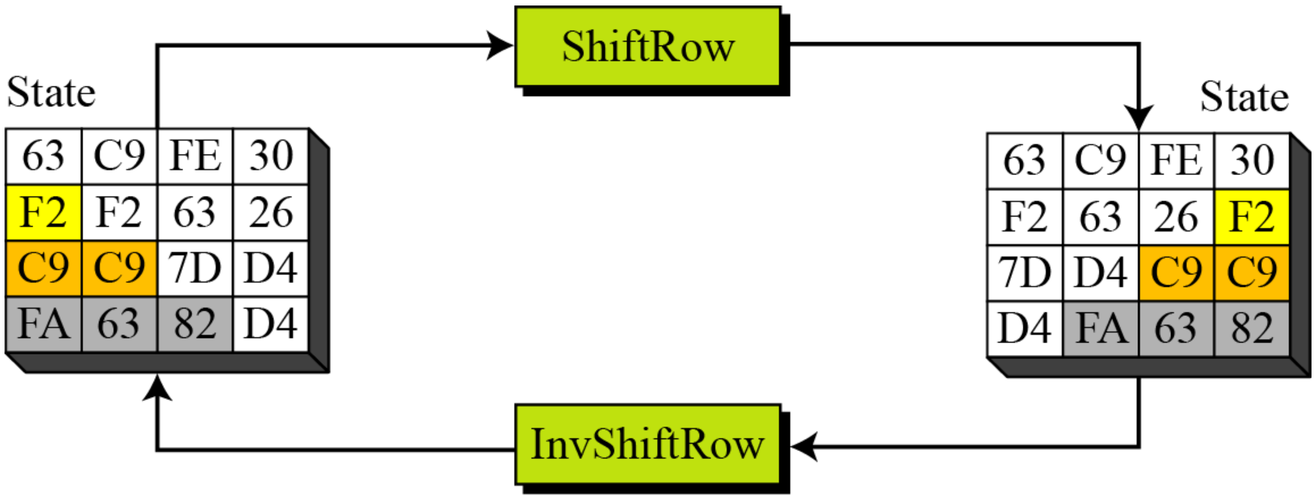
In the decryption, the transformation is called InvShiftRows and the shifting is to the right.

```
ShiftRows (S)
{
    for (r = 1 to 3)
        shiftrow (s_r, r)                // s_r is the rth row
}
shiftrow (row, n)                        // n is the number of bytes to be shifted
{
   CopyRow (row, t)                      // t is a temporary row
   for (c = 0 to 3)
        row_{(c − n) mod 4} ← t_c
}
```

4.2. Cấu trúc hệ mật AES

This figure shows how a state is transformed using ShiftRows transformation. The figure also shows that InvShiftRows transformation creates the original state.

ShiftRows transformation example



4.2. Cấu trúc hệ mật AES

Mixing

We need an interbyte transformation that changes the bits inside a byte, based on the bits inside the neighboring bytes. We need to mix bytes to provide diffusion at the bit level.

Mixing bytes using matrix multiplication

$$\begin{array}{r} a\mathbf{x} + b\mathbf{y} + c\mathbf{z} + d\mathbf{t} \longrightarrow \\ e\mathbf{x} + f\mathbf{y} + g\mathbf{z} + h\mathbf{t} \longrightarrow \\ i\mathbf{x} + j\mathbf{y} + k\mathbf{z} + l\mathbf{t} \longrightarrow \\ m\mathbf{x} + n\mathbf{y} + o\mathbf{z} + p\mathbf{t} \longrightarrow \end{array} \begin{bmatrix} \square \\ \square \\ \square \\ \square \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix} \times \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \\ \mathbf{t} \end{bmatrix}$$

New matrix | **Constant matrix** | Old matrix

4.2. Cấu trúc hệ mật AES

Mixing

$$\begin{matrix} a\mathbf{x} + b\mathbf{y} + c\mathbf{z} + d\mathbf{t} \\ e\mathbf{x} + f\mathbf{y} + g\mathbf{z} + h\mathbf{t} \\ i\mathbf{x} + j\mathbf{y} + k\mathbf{z} + l\mathbf{t} \\ m\mathbf{x} + n\mathbf{y} + o\mathbf{z} + p\mathbf{t} \end{matrix} \rightarrow \begin{bmatrix} \blacksquare \\ \blacksquare \\ \blacksquare \\ \blacksquare \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix} \times \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \\ \mathbf{t} \end{bmatrix}$$

New matrix | **Constant matrix** | Old matrix

Constant matrices used by MixColumns and InvMixColumns

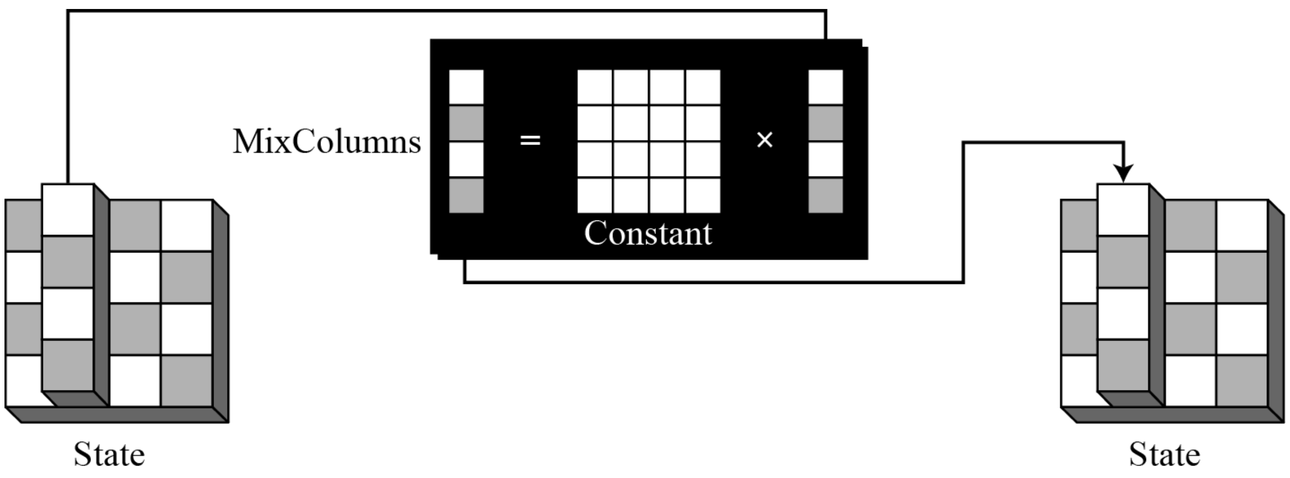
$$\underset{\mathrm{C}}{\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}} \xleftrightarrow{\text{Inverse}} \underset{\mathrm{C}^{-1}}{\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix}}$$

4.2. Cấu trúc hệ mật AES

MixColumns

The MixColumns transformation operates at the column level; it transforms each column of the state to a new column.

MixColumns transformation



4.2. Cấu trúc hệ mật AES

InvMixColumns

The InvMixColumns transformation is basically the same as the MixColumns transformation.

The MixColumns and InvMixColumns transformations are inverses of each other.

4.2. Cấu trúc hệ mật AES

```
MixColumns (S)
{
    for (c = 0 to 3)
        mixcolumn (s_c)
}
```

```
mixcolumn (col)
{
  CopyColumn (col, t)                    // t is a temporary column
```

$$\mathbf{col}_0 \leftarrow (0x02) \bullet \mathbf{t}_0 \oplus (0x03 \bullet \mathbf{t}_1) \oplus \mathbf{t}_2 \oplus \mathbf{t}_3$$

$$\mathbf{col}_1 \leftarrow \mathbf{t}_0 \oplus (0x02) \bullet \mathbf{t}_1 \oplus (0x03) \bullet \mathbf{t}_2 \oplus \mathbf{t}_3$$

$$\mathbf{col}_2 \leftarrow \mathbf{t}_0 \oplus \mathbf{t}_1 \oplus (0x02) \bullet \mathbf{t}_2 \oplus (0x03) \bullet \mathbf{t}_3$$

$$\mathbf{col}_3 \leftarrow (0x03 \bullet \mathbf{t}_0) \oplus \mathbf{t}_1 \oplus \mathbf{t}_2 \oplus (0x02) \bullet \mathbf{t}_3$$

```
}
```

4.2. Cấu trúc hệ mật AES

Figure below shows how a state is transformed using the MixColumns transformation. The figure also shows that the InvMixColumns transformation creates the original one.

The MixColumns transformation example

$$\text{State}\ \begin{bmatrix}63 & C9 & FE & 30\\ F2 & 63 & 26 & F2\\ 7D & D4 & C9 & C9\\ D4 & FA & 63 & 82\end{bmatrix} \underset{\textbf{InvMixColumn}}{\overset{\textbf{MixColumn}}{\rightleftarrows}} \begin{bmatrix}62 & 02 & 27 & 26\\ CF & 92 & 91 & 0D\\ 0C & 0C & F4 & D6\\ 99 & 18 & 30 & 74\end{bmatrix}\ \text{State}$$

4.2. Cấu trúc hệ mật AES

Key Adding

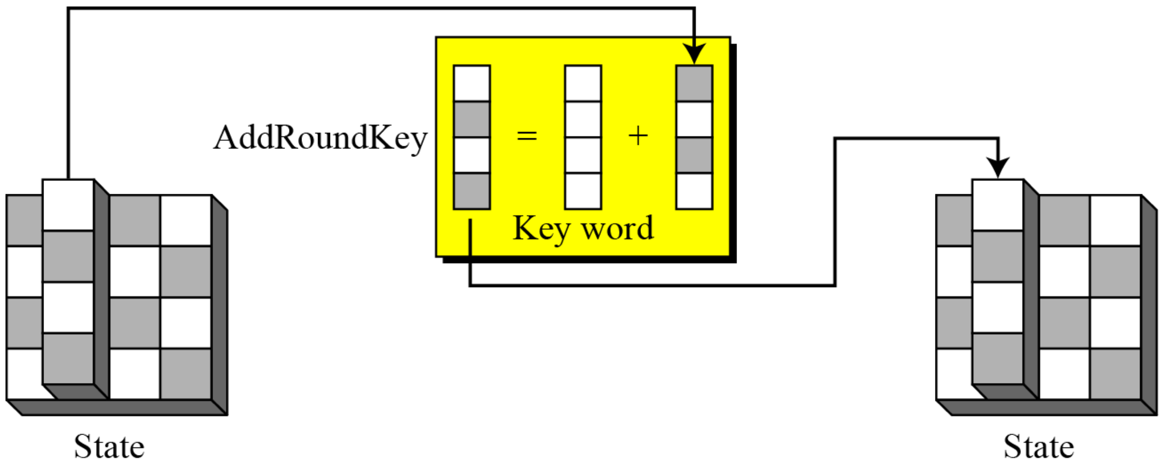
AddRoundKey

- *AddRoundKey proceeds one column at a time.*
- *AddRoundKey adds a round key word with each state column matrix*
- *The operation in AddRoundKey is matrix addition.*

The AddRoundKey transformation is the inverse of itself.

4.2. Cấu trúc hệ mật AES

AddRoundKey transformation



AddRoundKey (S)

{

for (c = 0 to 3)

$\mathbf{s}_c \leftarrow \mathbf{s}_c \oplus \mathbf{w}_{\text{round} + 4c}$

}

4.3. Mở rộng bộ khóa hệ mật AES

KEY EXPANSION

- *To create round keys for each round, AES uses a key-expansion process.*
- *If the number of rounds is N_r , the key-expansion routine creates $N_r + 1$ 128-bit round keys from one single 128-bit cipher key.*

- Key Expansion in AES-128
- Key Expansion in AES-192 and AES-256
- Key-Expansion Analysis

The key-expansion mechanism in AES has been designed to provide several features that thwart the cryptanalyst.

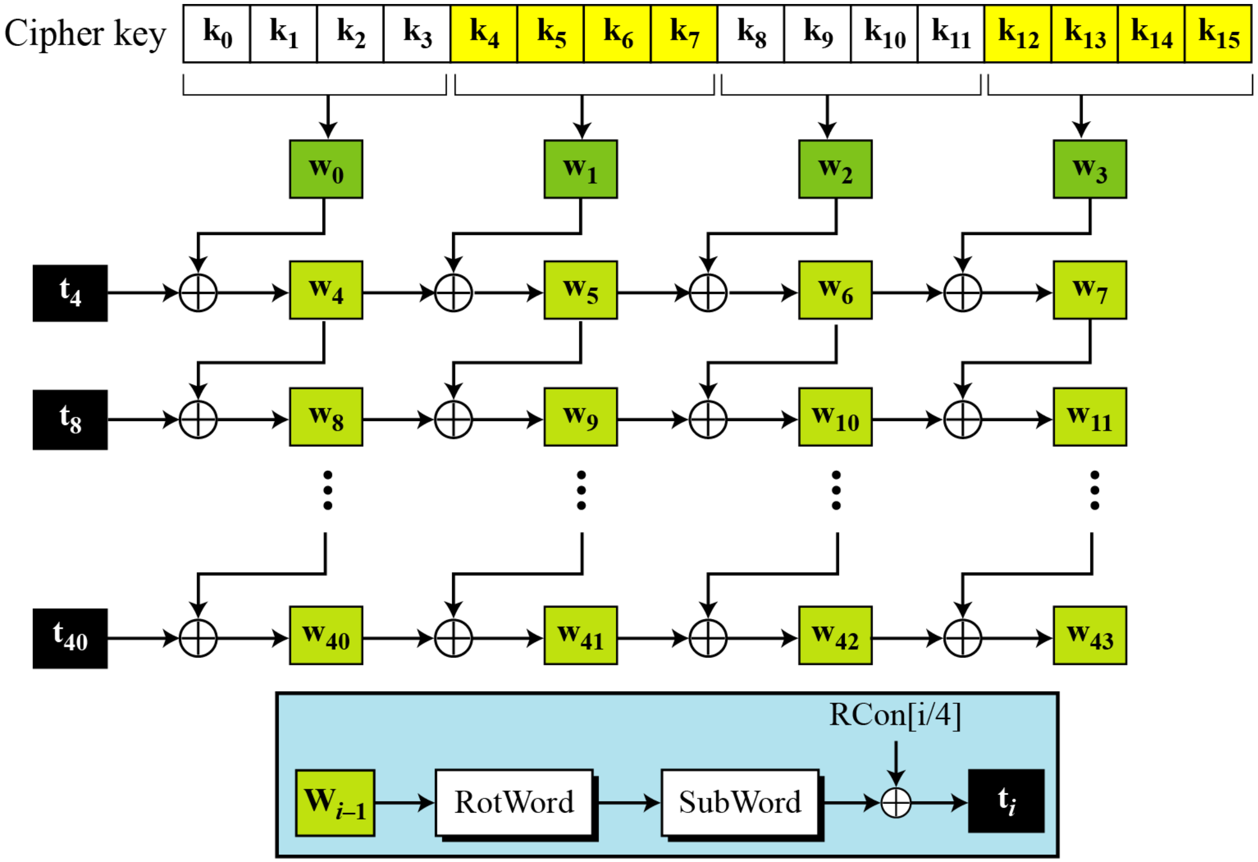
4.3. Mở rộng bộ khóa hệ mật AES

To create round keys for each round, AES uses a key-expansion process. ***If the number of rounds is N_r , the key-expansion routine creates $N_r + 1$ 128-bit round keys from one single 128-bit cipher key.***

Round	*Words*			
Pre-round	$\mathbf{w}_0$	$\mathbf{w}_1$	$\mathbf{w}_2$	$\mathbf{w}_3$
1	$\mathbf{w}_4$	$\mathbf{w}_5$	$\mathbf{w}_6$	$\mathbf{w}_7$
2	$\mathbf{w}_8$	$\mathbf{w}_9$	$\mathbf{w}_{10}$	$\mathbf{w}_{11}$
. . .	. . .			
N_r	$\mathbf{w}_{4N_r}$	$\mathbf{w}_{4N_r+1}$	$\mathbf{w}_{4N_r+2}$	$\mathbf{w}_{4N_r+3}$

4.3. Mở rộng bộ khóa hệ mật AES

Key Expansion in AES-128



Making of t_i (temporary) words $i = 4\ N_r$.

4.3. Mở rộng bộ khóa hệ mật AES

Key Expansion in AES-128

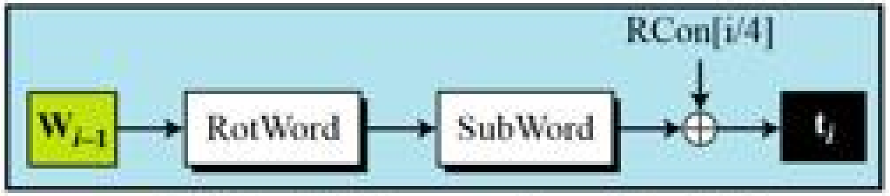
The process is as follows:

1. The first four words ($\mathbf{w}_0$, $\mathbf{w}_1$, $\mathbf{w}_2$, $\mathbf{w}_3$) are made from the cipher key. The cipher key is thought of as an array of 16 bytes (k_0 to k_{15}). The first four bytes (k_0 to k_3) become $\mathbf{w}_0$; the next four bytes (k_4 to k_7) become $\mathbf{w}_1$; and so on. In other words, the concatenation of the words in this group replicates the cipher key.
2. The rest of the words ($\mathbf{w}_i$ for $i = 4$ to 43) are made as follows:
   a. If (i mod 4) $\neq 0$, $\mathbf{w}_i = \mathbf{w}_{i-1} \oplus \mathbf{w}_{i-4}$. this means each word is made from the one at the left and the one at the top.
   b. If (i mod 4) $= 0$, $\mathbf{w}_i = \mathbf{t} \oplus \mathbf{w}_{i-4}$. Here $\mathbf{t}$, a temporary word, is the result of applying two routines, SubWord and RotWord, on $\mathbf{w}_{i-1}$ and XORing the result with a round constants, RCon. In other words, we have,

$$\mathbf{t} = \text{SubWord}(\text{RotWord}(\mathbf{w}_{i-1})) \oplus \text{RCon}_{i/4}$$

4.3. Mở rộng bộ khóa hệ mật AES

Key Expansion in AES-128



Making of t_i (temporary) words $i = 4\,N_r$

RotWord

The **RotWord** (rotate word) routine is similar to the ShiftRows transformation, but it is applied to only one row. The routine takes a word as an array of four bytes and shifts each byte to the left with wrapping.

SubWord

The **SubWord** (substitute word) routine is similar to the SubBytes transformation, but it is applied only to four bytes. The routine takes each byte in the word and substitutes another byte for it.

4.3. Mở rộng bộ khóa hệ mật AES

Key Expansion in AES-128

Each round constant, RCon, is a 4-byte value in which the rightmost three bytes are always zero.

Round	*Constant (RCon)*	*Round*	*Constant (RCon)*
1	(**01** 00 00 00)$_{16}$	6	(**20** 00 00 00)$_{16}$
2	(**02** 00 00 00)$_{16}$	7	(**40** 00 00 00)$_{16}$
3	(**04** 00 00 00)$_{16}$	8	(**80** 00 00 00)$_{16}$
4	(**08** 00 00 00)$_{16}$	9	(**1B** 00 00 00)$_{16}$
5	(**10** 00 00 00)$_{16}$	10	(**36** 00 00 00)$_{16}$

4.3. Mở rộng bộ khóa hệ mật AES

Key Expansion in AES-128

The key-expansion routine can either use the above table when calculating the words or use the $GF(2^8)$ field to calculate the leftmost byte dynamically, as shown below (prime is the irreducible polynomial):

RC_1	$\rightarrow x^{1-1}$	$= x^0$	mod *prime*	$= 1$	$\rightarrow 00000001$	$\rightarrow 01_{16}$
RC_2	$\rightarrow x^{2-1}$	$= x^1$	mod *prime*	$= x$	$\rightarrow 00000010$	$\rightarrow 02_{16}$
RC_3	$\rightarrow x^{3-1}$	$= x^2$	mod *prime*	$= x^2$	$\rightarrow 00000100$	$\rightarrow 04_{16}$
RC_4	$\rightarrow x^{4-1}$	$= x^3$	mod *prime*	$= x^3$	$\rightarrow 00001000$	$\rightarrow 08_{16}$
RC_5	$\rightarrow x^{5-1}$	$= x^4$	mod *prime*	$= x^4$	$\rightarrow 00010000$	$\rightarrow 10_{16}$
RC_6	$\rightarrow x^{6-1}$	$= x^5$	mod *prime*	$= x^5$	$\rightarrow 00100000$	$\rightarrow 20_{16}$
RC_7	$\rightarrow x^{7-1}$	$= x^6$	mod *prime*	$= x^6$	$\rightarrow 01000000$	$\rightarrow 40_{16}$
RC_8	$\rightarrow x^{8-1}$	$= x^7$	mod *prime*	$= x^7$	$\rightarrow 10000000$	$\rightarrow 80_{16}$
RC_9	$\rightarrow x^{9-1}$	$= x^8$	mod *prime*	$= x^4 + x^3 + x + 1$	$\rightarrow 00011011$	$\rightarrow 1B_{16}$
RC_{10}	$\rightarrow x^{10-1}$	$= x^9$	mod *prime*	$= x^5 + x^4 + x^2 + x$	$\rightarrow 00110110$	$\rightarrow 36_{16}$

4.3. Mở rộng bộ khóa hệ mật AES

Key Expansion in AES-128

KeyExpansion ([key_0 to key_{15}], [$\mathbf{w}_0$ **to** $\mathbf{w}_{43}$])

{

for (i = 0 to 3)

$\mathbf{w}_i \leftarrow \text{key}_{4i} + \text{key}_{4i+1} + \text{key}_{4i+2} + \text{key}_{4i+3}$

for (i = 4 to 43)

{

if (i mod 4 $\neq$ 0) $\mathbf{w}_i \leftarrow \mathbf{w}_{i-1} + \mathbf{w}_{i-4}$

else

{

$\mathbf{t} \leftarrow \text{SubWord}(\text{RotWord}(\mathbf{w}_{i-1})) \oplus \text{RCon}_{i/4}$ *// **t** is a temporary word*

$\mathbf{w}_i \leftarrow \mathbf{t} + \mathbf{w}_{i-4}$

}

}

}

Key Expansion in AES-128

Each round key in AES depends on the previous round key. The dependency, however, is nonlinear because of SubWord transformation. The addition of the round constants also guarantees that each round key will be different from the previous one.

4.3. Mở rộng bộ khóa hệ mật AES

Ví dụ

This table shows how the keys for each round are calculated assuming that the 128-bit cipher key agreed upon by Alice and Bob is (24 75 A2 B3 34 75 56 88 31 E2 12 00 13 AA 54 87)$_{16}$.

Round	*Values of t's*	*First word in the round*	*Second word in the round*	*Third word in the round*	*Fourth word in the round*
—		w_{00} = 2475A2B3	w_{01} = 34755688	w_{02} = 31E21200	w_{03} = 13AA5487
1	AD20177D	w_{04} = 8955B5CE	w_{05} = BD20E346	w_{06} = 8CC2F146	w_{07} = 9F68A5C1
2	470678DB	w_{08} = CE53CD15	w_{09} = 73732E53	w_{10} = FFB1DF15	w_{11} = 60D97AD4
3	31DA48D0	w_{12} = FF8985C5	w_{13} = 8CFAAB96	w_{14} = 734B7483	w_{15} = 2475A2B3
4	47AB5B7D	w_{16} = B822deb8	w_{17} = 34D8752E	w_{18} = 479301AD	w_{19} = 54010FFA
5	6C762D20	w_{20} = D454F398	w_{21} = E08C86B6	w_{22} = A71F871B	w_{23} = F31E88E1
6	52C4F80D	w_{24} = 86900B95	w_{25} = 661C8D23	w_{26} = C1030A38	w_{27} = 321D82D9
7	E4133523	w_{28} = 62833EB6	w_{29} = 049FB395	w_{30} = C59CB9AD	w_{31} = F7813B74
8	8CE29268	w_{32} = EE61ACDE	w_{33} = EAFE1F4B	w_{34} = 2F62A6E6	w_{35} = D8E39D92
9	0A5E4F61	w_{36} = E43FE3BF	w_{37} = 0EC1FCF4	w_{38} = 21A35A12	w_{39} = F940C780
10	3FC6CD99	w_{40} = DBF92E26	w_{41} = D538D2D2	w_{42} = F49B88C0	w_{43} = 0DDB4F40

4.3. Mở rộng bộ khóa hệ mật AES

Ví dụ

The concept of weak keys, as we discussed for DES in Chapter 3, does not apply to AES. Assume that all bits in the cipher key are 0s. The following shows the words for some rounds:

Pre-round:	00000000	00000000	00000000	00000000
Round 01:	62636363	62636363	62636363	62636363
Round 02:	9B9898C9	F9FBFBAA	9B9898C9	F9FBFBAA
Round 03:	90973450	696CCFFA	F2F45733	0B0FAC99
...	...	...	...	...
Round 10:	B4EF5BCB	3E92E211	23E951CF	6F8F188E

The words in the pre-round and the first round are all the same. In the second round, the first word matches with the third; the second word matches with the fourth. However, after the second round the pattern disappears; every word is different.

4.3. Mở rộng bộ khóa hệ mật AES

Key-expansion algorithms in the AES-192 and AES-256 versions are very similar to the key expansion algorithm in AES-128, with the following differences:

1. In AES-192, the words are generated in groups of six instead of four.
   a. The cipher key creates the first six words ($\mathbf{w}_0$ to $\mathbf{w}_5$).
   b. If $i \bmod 6 \neq 0$, $\mathbf{w}_i \leftarrow \mathbf{w}_{i-1} + \mathbf{w}_{i-6}$; otherwise, $\mathbf{w}_i \leftarrow \mathbf{t} + \mathbf{w}_{i-6}$.

2. In AES-256, the words are generated in groups of eight instead of four.
   a. The cipher key creates the first eight words ($\mathbf{w}_0$ to $\mathbf{w}_7$).
   b. If $i \bmod 8 \neq 0$, $\mathbf{w}_i \leftarrow \mathbf{w}_{i-1} + \mathbf{w}_{i-8}$; otherwise, $\mathbf{w}_i \leftarrow \mathbf{t} + \mathbf{w}_{i-8}$.
   c. If $i \bmod 4 = 0$, but $i \bmod 8 \neq 0$, then $\mathbf{w}_i = \text{SubWord}(\mathbf{w}_{i-1}) + \mathbf{w}_{i-8}$.

4.4. Cách triển khai hệ mật AES

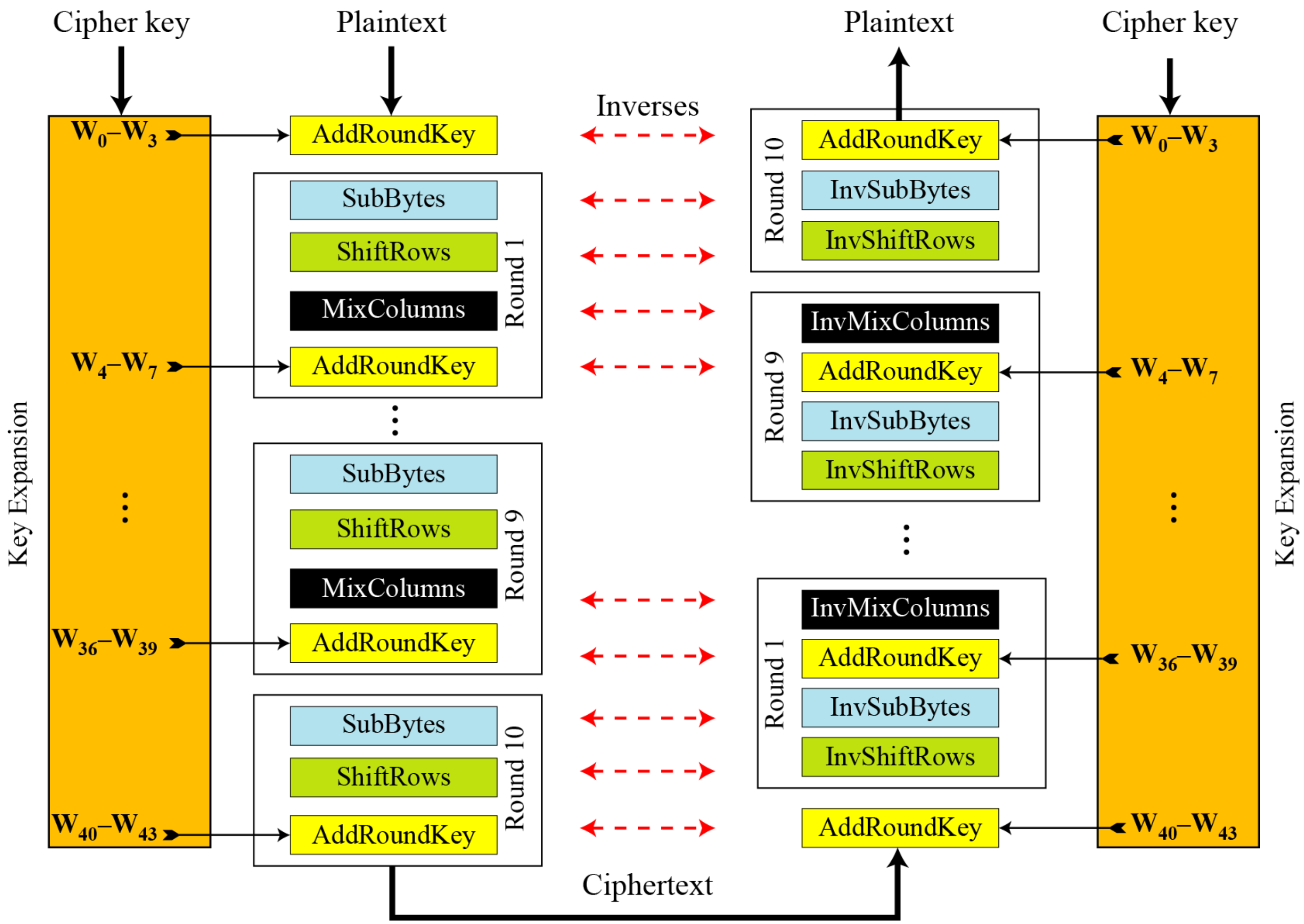
- *AES uses four types of transformations for encryption and decryption.*
- *In the standard, the encryption algorithm is referred to as the cipher and the decryption algorithm as the inverse cipher.*

- Original Design
- Alternative Design

4.4. Cách triển khai hệ mật AES

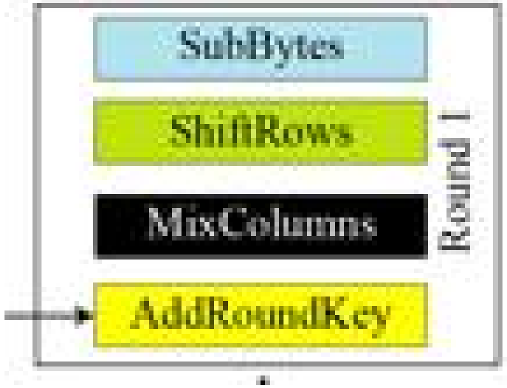
Original Design

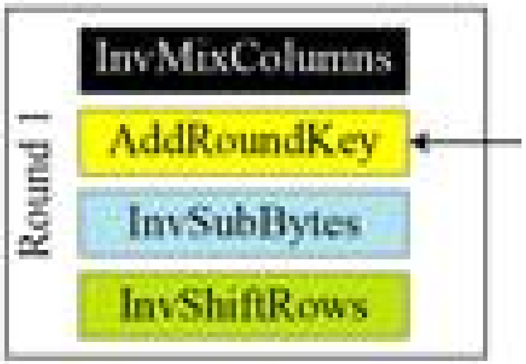
Ciphers and inverse ciphers of the original design



4.4. Cách triển khai hệ mật AES

In the original design, the order of transformations in each round is not the same in the cipher and reverse cipher.





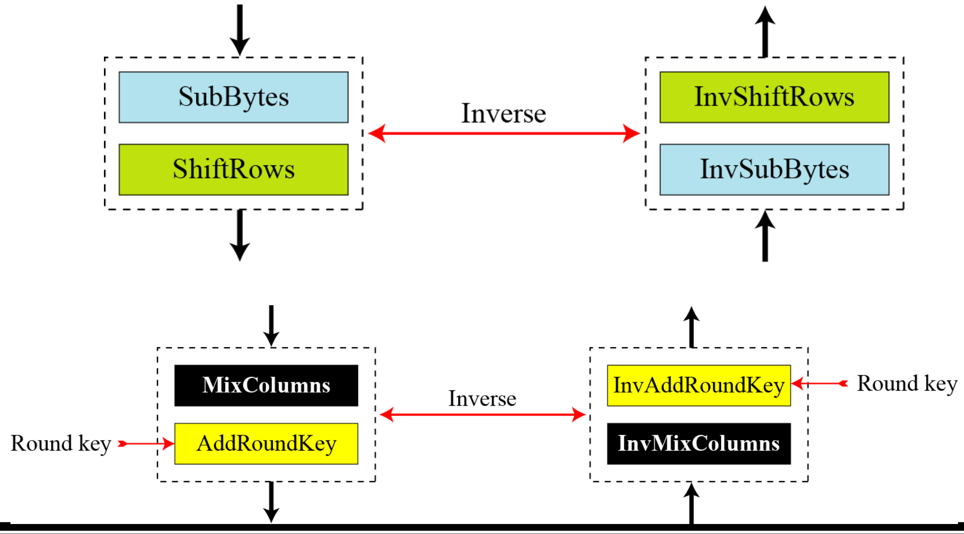
First, the order of SubBytes and ShiftRows is changed in the reverse cipher.

Second, the order of MixColumns and AddRoundKey is changed in the reverse cipher.

4.4. Cách triển khai hệ mật AES

Alternative Design

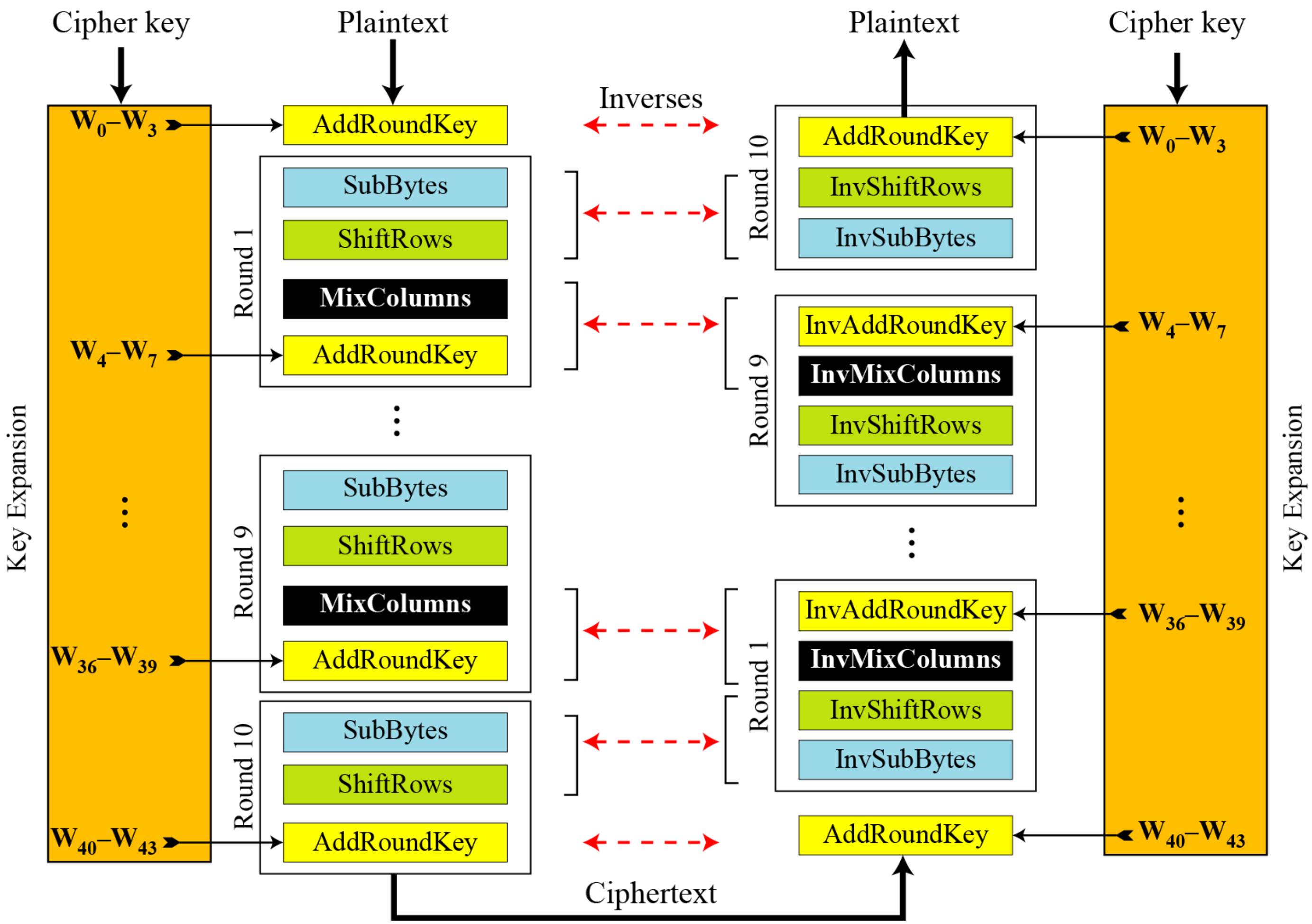
- In this version, the transformation in the reverse cipher are rearranged to make the order of transformations the same in the cipher and reverse cipher.
- In this design, invertibility is provided for a pair of transformations, not for each single transformation



4.4. Cách triển khai hệ mật AES

Alternative Design

Cipher and reverse cipher in alternate design



4.4. Cách triển khai hệ mật AES

Alternative Design

Changing Key-Expansion Algorithm

Instead of using InvRoundKey transformation in the reverse cipher, the key-expansion algorithm can be changed to create a different set of round keys for the inverse cipher.

Note that:

- The round key for the pre-round operation and the last round should not be changed.
- The round keys for round 1 to 9 need to be multiplied by the constant matrix

4.4. Cách triển khai hệ mật AES

Alternative Design

Ví dụ

The following shows the ciphertext block created from a plaintext block using a randomly selected cipher key.

Plaintext:	00	04	12	14	12	04	12	00	0C	00	13	11	08	23	19	19
Cipher Key:	24	75	A2	B3	34	75	56	88	31	E2	12	00	13	AA	54	87
Ciphertext:	BC	02	8B	D3	E0	E3	B1	95	55	0D	6D	FB	E6	F1	82	41

4.4. Cách triển khai hệ mật AES

Round	*Input State*	*Output State*	*Round Key*
Pre-round	00 12 0C 08 04 04 00 23 12 12 13 19 14 00 11 19	24 26 3D 1B 71 71 E2 89 B0 44 01 4D A7 88 11 9E	24 34 31 13 75 75 E2 AA A2 56 12 54 B3 88 00 87
1	24 26 3D 1B 71 71 E2 89 B0 44 01 4D A7 88 11 9E	6C 44 13 BD B1 9E 46 35 C5 B5 F3 02 5D 87 FC 8C	89 BD 8C 9F 55 20 C2 68 B5 E3 F1 A5 CE 46 46 C1
2	6C 44 13 BD B1 9E 46 35 C5 B5 F3 02 5D 87 FC 8C	1A 90 15 B2 66 09 1D FC 20 55 5A B2 2B CB 8C 3C	CE 73 FF 60 53 73 B1 D9 CD 2E DF 7A 15 53 15 D4

4.4. Cách triển khai hệ mật AES

3	1A 90 15 B2 66 09 1D FC 20 55 5A B2 2B CB 8C 3C	F6 7D A2 B0 1B 61 B4 B8 67 09 C9 45 4A 5C 51 09	FF 8C 73 13 89 FA 4B 92 85 AB 74 0E C5 96 83 57
4	F6 7D A2 B0 1B 61 B4 B8 67 09 C9 45 4A 5C 51 09	CA E5 48 BB D8 42 AF 71 D1 BA 98 2D 4E 60 9E DF	B8 34 47 54 22 D8 93 01 DE 75 01 0F B8 2E AD FA
5	CA E5 48 BB D8 42 AF 71 D1 BA 98 2D 4E 60 9E DF	90 35 13 60 2C FB 82 3A 9E FC 61 ED 49 39 CB 47	D4 E0 A7 F3 54 8C 1F 1E F3 86 87 88 98 B6 1B E1
6	90 35 13 60 2C FB 82 3A 9E FC 61 ED 49 39 CB 47	18 0A B9 B5 64 68 6A FB 5A EF D7 79 8E B2 10 4D	86 66 C1 32 90 1C 03 1D 0B 8D 0A 82 95 23 38 D9

4.4. Cách triển khai hệ mật AES

7	18 0A B9 B5 64 68 6A FB 5A EF D7 79 8E B2 10 4D	01 63 F1 96 55 24 3A 62 F4 8A DE 4D CC BA 88 03	62 04 C5 F7 83 9F 9C 81 3E B3 B9 3B B6 95 AD 74
8	01 63 F1 96 55 24 3A 62 F4 8A DE 4D CC BA 88 03	2A 34 D8 46 2D 6B A2 D6 51 64 CF 5A 87 A8 F8 28	EE EA 2F D8 61 FE 62 E3 AC 1F A6 9D DE 4B E6 92
9	2A 34 D8 46 2D 6B A2 D6 51 64 CF 5A 87 A8 F8 28	0A D9 F1 3C 95 63 9F 35 2A 80 29 00 16 76 09 77	E4 0E 21 F9 3F C1 A3 40 E3 FC 5A C7 BF F4 12 80
10	0A D9 F1 3C 95 63 9F 35 2A 80 29 00 16 76 09 77	BC E0 55 E6 02 E3 0D F1 8B B1 6D 82 D3 95 F8 41	DB D5 F4 0D F9 38 9B DB 2E D2 88 4F 26 D2 C0 40

4.4. Cách triển khai hệ mật AES

This figure shows the state entries in one round, round 7.

States in a single round

18	0A	B9	B5
64	68	6A	FB
5A	EF	D7	79
8E	B2	10	D4

Input State

→

7C	FB	A1	90
36	80	AA	FC
1D	E3	BF	7E
7B	4B	F4	C4

After SubBytes

→

C4	DE	F7	9E
4C	95	C0	35
FD	7B	69	C7
59	E3	1E	BA

After ShiftRows

→

2A	34	D8	46
2D	6B	A2	D6
51	64	CF	5A
87	A8	F8	28

After MixColumns

→

01	63	F1	96
55	24	3A	62
F4	8A	DE	4D
CC	BA	88	03

Output State
(After RoundKey)

One may be curious to see the result of encryption when the plaintext is made of all 0s.

Plaintext:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Cipher Key:	24 75 A2 B3 34 75 56 88 31 E2 12 00 13 AA 54 87
Ciphertext:	63 2C D4 5E 5D 56 ED B5 62 04 01 A0 AA 9C 2D 8D

4.4. Cách triển khai hệ mật AES

The avalanche effect

Plaintext 1:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Plaintext 2:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	0<u>1</u>
Ciphertext 1:	63	2C	D4	5E	5D	56	ED	B5	62	04	01	A0	AA	9C	2D	8D
Ciphertext 2:	26	F3	9B	BC	A1	9C	0F	B7	C7	2E	7E	30	63	92	73	13

4.4. Cách triển khai hệ mật AES

The following shows the effect of using a cipher key in which all bits are 0s.

Plaintext:	00	04	12	14	12	04	12	00	0c	00	13	11	08	23	19	19
Cipher Key:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Ciphertext:	5A	6F	4B	67	57	B7	A5	D2	C4	30	91	ED	64	9A	42	72

4.5. Thám mã hệ mật AES

AES was designed after DES. Most of the known attacks on DES were already tested on AES.

Brute-Force Attack

AES is definitely more secure than DES due to the larger-size key.

Statistical Attacks

Numerous tests have failed to do statistical analysis of the ciphertext.

Differential and Linear Attacks

There are no differential and linear attacks on AES as yet.

4.5. Thám mã hệ mật AES

Statistical Attacks

Numerous tests have failed to do statistical analysis of the ciphertext.

Differential and Linear Attacks

There are no differential and linear attacks on AES as yet.