

MH4311 Cryptography

Lecture 6

Block Cipher (Part 3, AES)

Wu Hongjun

Lecture Outline

- **Classical ciphers**
- **Symmetric key encryption**
 - **One-time pad & information theory**
 - **Block cipher**
 - **Introduction**
 - **DES, Double DES, Triple DES**
 - **[AES](#)**
 - **Modes of Operation**
 - **Attacks**
 - **Stream cipher**
- **Hash function and Message Authentication Code**
- **Public key encryption**
- **Digital signature**
- **Key establishment and management**
- **Introduction to other cryptographic topics**

Recommended Reading

- CTP Section 3.6
- FIPS 197 (complete AES specifications)
<http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>
- Wikipedia:
 - AES
http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

Advanced Encryption Standard (AES)

- **AES**
 - **Block cipher**
 - **128-bit block size**
 - **Substitution-permutation network**
 - **Three different key sizes & round numbers**
 - **AES-128: 128-bit key + 10 rounds**
 - **AES-192: 192-bit key + 12 rounds**
 - **AES-256: 256-bit key + 14 rounds**

AES: History

- **1997: NIST called for algorithm to replace DES**
- **1998: 15 ciphers submitted for competition**
- **2001: Rijndael was approved as AES (FIPS 197)**
 - **Designers: Joan Daemen, Vincent Rijmen**



AES: Applications

- **AES**
 - **Free**
 - **Simple & elegant design**
 - **High security**
 - **Excellent performance (hardware & software)**
- **USA government standard**
 - **AES-128 for SECRET information**
 - **AES-192 and AES-256 for TOP SECRET information**
- **Commercial applications**
 - **Too many ...**

Mathematical Preliminaries

- 1. Euclidean Algorithm, Extended Euclidean Algorithm**
- 2. Group, Ring, Field, Finite Field**
- 3. Finite field $\text{GF}(2^8)$:**
Addition, Multiplication, Multiplicative Inverse
- 4. Polynomials with coefficients in $\text{GF}(2^8)$**
Addition, Multiplication, Multiplicative Inverse

Mathematical Preliminaries

1. Euclidean Algorithm and Extended Euclidean Algorithm

Euclidean Algorithm

- used to find the GCD efficiently
- uses the following property repeatedly:
$$\text{GCD}(a, b) = \text{GCD}(b, a \bmod b)$$
- Example:
$$\text{GCD}(12, 5) = ?$$

Mathematical Preliminaries

Extended Euclidean Algorithm

- This algorithm is used to find x and y satisfying
$$ax + by = \text{GCD}(a, b)$$
- when $\text{GCD}(a, b) = 1$, x is the multiplicative inverse of a modulo b , and y is the multiplicative inverse of b modulo a .
- Basic idea of the algorithm: at each step (the i -th step) in the Euclidean algorithm, find

$$r_i = ax_i + by_i$$

Mathematical Preliminaries

Extended Euclidean Algorithm

- **Example** $a = 12, b = 5$, find $\text{GCD}(a,b)$

$$\begin{array}{ll} & \text{GCD}(a, b) \\ r_1 = a \bmod b = 2 & \text{GCD}(b, r_1) \\ r_2 = b \bmod r_1 = 1 & \end{array}$$

$$\begin{array}{l} r_1 = a - 2b \\ r_2 = b - 2r_1 = b - 2(a - 2b) = 5b - 2a \end{array}$$

Euclidean Algorithm: $\text{GCD}(a,b) = 1$

Extended Euclidean Algorithm: $1 = 5b - 2a$

Mathematical Preliminaries

2. Group, Ring, Field

Mathematical Preliminaries

2.1 Group

- A group is a set, G , together with an operation \bullet that combines any two elements a and b to form another element, denoted $a \bullet b$ or ab . To qualify as a group, the set and operation, (G, \bullet) , must satisfy:
 - Closure: For all a, b in G , the result of the operation, $a \bullet b$, is also in G .
 - Associativity: $(a \bullet b) \bullet c = a \bullet (b \bullet c)$.
 - Identity element: There exists an element e in G , such that for every element a in G , the equation $e \bullet a = a \bullet e = a$ holds.
 - Inverse element: For each a in G , there exists an element b in G such that $a \bullet b = b \bullet a = e$
- Example:
 - The set of integers \mathbb{Z} together with integer addition

Mathematical Preliminaries

2.2 Ring

- **Wikipedia:** A ring is an algebraic structure which generalizes the main properties of the addition and the multiplication of integers, real numbers, complex numbers and square matrices
- **Example:** The set of integers \mathbb{Z} together with integer addition and multiplication

Mathematical Preliminaries

Ring

- A ring is **a set** R equipped with **two operations**

$$+ : R \times R \rightarrow R \text{ and } \cdot : R \times R \rightarrow R,$$

called *addition* and *multiplication*. To qualify as a ring, the set and two operations, $(R, +, \cdot)$, must satisfy:

- $(R, +)$ is required to be an *abelian group* under addition
 - Abelian group has additional property: Commutativity:
 $a \cdot b = b \cdot a$
- (R, \cdot) is required to satisfy
 - Closure of multiplication
 - Associativity of multiplication
 - Existence of multiplicative identity element
- The distributive laws:
 - $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$
 - $(a + b) \cdot c = (a \cdot c) + (b \cdot c)$

Mathematical Preliminaries

Field

- A field is a commutative ring whose nonzero elements form a group under multiplication
- Example
 - the field of rational numbers

Mathematical Preliminaries

Finite field (or Galois Field)

- A field that contains a finite number of elements
- For every prime number p and positive integer n , there exists a finite field with p^n elements.
- Examples
 - GF(2)
0,1
addition and multiplication modulo 2
(XOR and AND)
 - GF(7)
0,1,2,3,4,5,6
addition and multiplication modulo 7



Évariste Galois
(Oct 1811- May 1832)

Mathematical Preliminaries

3. The finite field $\text{GF}(2^8)$

Mathematical Preliminaries

- An element in the finite field $\text{GF}(2^8)$ can be denoted using

- Binary notation: consisting of eight bits (one byte)

$b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0$

(each b_i is one bit)

- There are 256 elements in this finite field

00000000

00000001

00000010

00000011

00000100

.....

.....

11111100

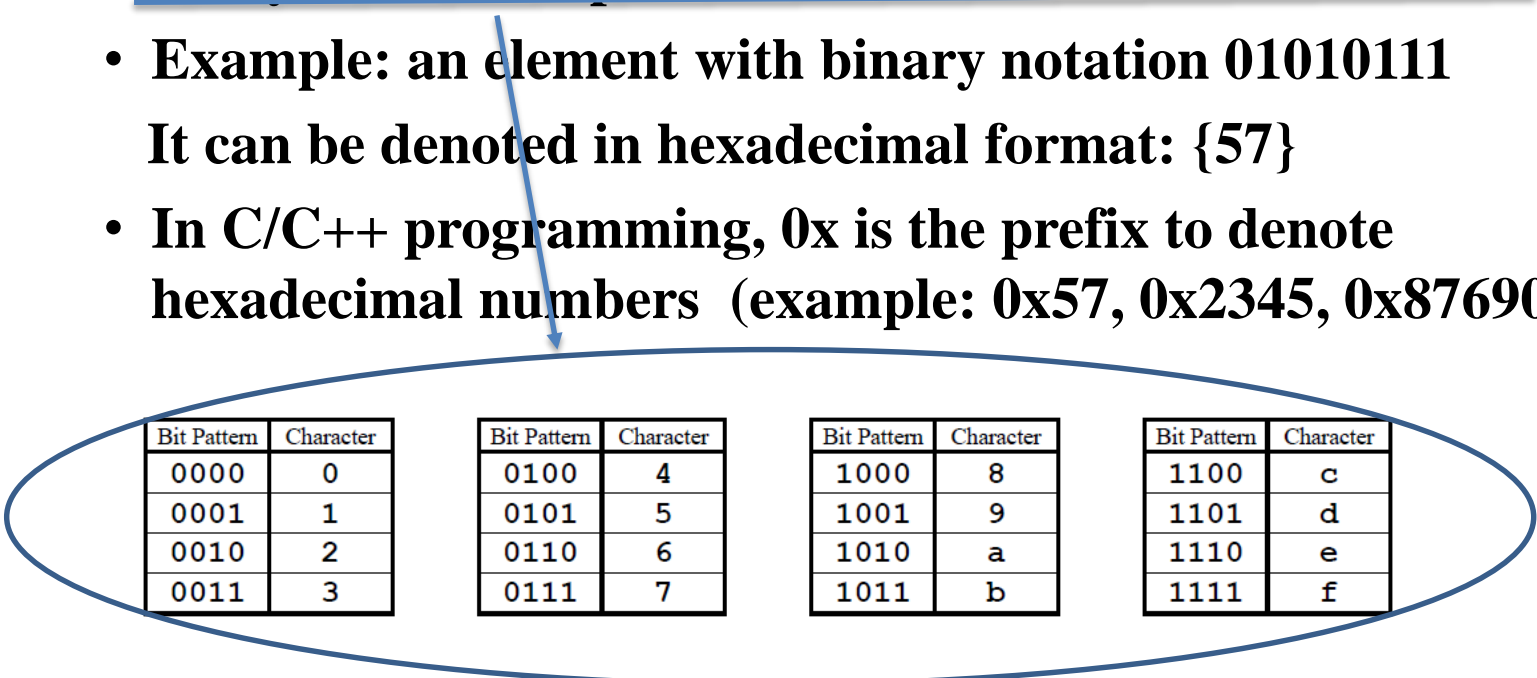
11111101

11111110

11111111

Mathematical Preliminaries

- An element in the finite field $\text{GF}(2^8)$ can be denoted using
 - Hexadecimal notation:
 - Hexadecimal: base-16 number system
 - Every four bits represented as a hexadecimal character
 - Example: an element with binary notation 01010111
It can be denoted in hexadecimal format: {57}
 - In C/C++ programming, 0x is the prefix to denote hexadecimal numbers (example: 0x57, 0x2345, 0x87690)



Bit Pattern	Character
0000	0
0001	1
0010	2
0011	3

Bit Pattern	Character
0100	4
0101	5
0110	6
0111	7

Bit Pattern	Character
1000	8
1001	9
1010	a
1011	b

Bit Pattern	Character
1100	c
1101	d
1110	e
1111	f

Mathematical Preliminaries

- **An element in the finite field $\text{GF}(2^8)$ can be denoted using**

- **Polynomial notation: b is considered as a polynomial with binary coefficients (either 0 or 1):**

$$b_7 x^7 + b_6 x^6 + b_5 x^5 + b_4 x^4 + b_3 x^3 + b_2 x^2 + b_1 x + b_0$$

- **Example: a byte with hexadecimal value {57} (binary notation: 01010111) corresponds with polynomial:**

$$x^6 + x^4 + x^2 + x + 1$$

Mathematical Preliminaries

- Addition in finite field $\text{GF}(2^8)$:

$$(x^6 + x^4 + x^2 + x + 1) + (x^7 + x + 1) = x^7 + x^6 + x^4 + x^2 \quad (\text{polynomial notation});$$

$$\{01010111\} \oplus \{10000011\} = \{11010100\} \quad (\text{binary notation});$$

$$\{57\} \oplus \{83\} = \{\text{d4}\} \quad (\text{hexadecimal notation}).$$

Bit Pattern	Character
0000	0
0001	1
0010	2
0011	3

Bit Pattern	Character
0100	4
0101	5
0110	6
0111	7

Bit Pattern	Character
1000	8
1001	9
1010	a
1011	b

Bit Pattern	Character
1100	c
1101	d
1110	e
1111	f

Mathematical Preliminaries

- **Multiplication in finite field $\text{GF}(2^8)$**
 - denoted here by \bullet
 - defined as: multiplication of **binary** polynomials modulo an irreducible **binary** polynomial of degree 8
 - irreducible polynomial: indivisible by any polynomial other than 1 and itself
 - In AES, the following irreducible polynomial is used:

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

Mathematical Preliminaries

- Multiplication in finite field $\text{GF}(2^8)$ (cont.)

- Example:

$$\begin{aligned}
 & \frac{(x^6 + x^4 + x^2 + x + 1)}{\quad} \frac{(x^7 + x + 1)}{\quad} = x^{13} + x^{11} + x^9 + x^8 + x^7 + \\
 & \quad x^7 + x^5 + x^3 + x^2 + x + \\
 & \quad x^6 + x^4 + x^2 + x + 1 \\
 & = x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1 \\
 & x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1 \text{ modulo } (x^8 + x^4 + x^3 + x + 1) \\
 & = \underline{x^7 + x^6 + 1}.
 \end{aligned}$$

$$\{57\} \bullet \{83\} = \{c1\}$$

Mathematical Preliminaries

- **Multiplicative inverse in finite field $\text{GF}(2^8)$**
 - **Extended Euclidean Algorithm is used to find the inverse**
 - For given a and b , find x and y satisfying
$$ax + by = \gcd(a, b)$$
 - If $\gcd(a, b) = 1$, then $ax \bmod b = 1$, i.e.,
 x is the modular multiplicative inverse of a modulo b

Mathematical Preliminaries

4. Polynomials with coefficients in $\text{GF}(2^8)$

Mathematical Preliminaries

- **Polynomials with coefficients in $\text{GF}(2^8)$**
 - Let $[a_0, a_1, a_2, a_3]$ denote four bytes. We have the following polynomials with coefficients in $\text{GF}(2^8)$:
$$a(x) = a_3x^3 + a_2x^2 + a_1x + a_0$$
 - The above polynomials are the elements of a polynomial ring used in AES
 - 2^{32} elements in the polynomial ring in AES
 - Addition
 - Coefficient addition is the addition over $\text{GF}(2^8)$, i.e., XOR
 - Multiplication:
 - Coefficient multiplication is the multiplication over $\text{GF}(2^8)$
 - Reduction polynomial is $x^4 + 1$
 - » This reduction polynomial is reducible, so some elements in this ring do not have multiplicative inverse

Mathematical Preliminaries

- Addition in the polynomial ring of AES

$$a(x) = a_3x^3 + a_2x^2 + a_1x + a_0$$

$$b(x) = b_3x^3 + b_2x^2 + b_1x + b_0$$

$$a(x) + b(x) = (a_3 \oplus b_3)x^3 + (a_2 \oplus b_2)x^2 + (a_1 \oplus b_1)x + (a_0 \oplus b_0)$$

Mathematical Preliminaries

- **Multiplication in the polynomial ring in AES**

Denoted here as \otimes

$$a(x) = a_3x^3 + a_2x^2 + a_1x + a_0$$

$$b(x) = b_3x^3 + b_2x^2 + b_1x + b_0$$

$$\begin{aligned} d(x) &= a(x) \otimes b(x) \\ &= (a(x) \bullet b(x)) \bmod (x^4 + 1) \end{aligned}$$

Mathematical Preliminaries

- **Multiplication in the polynomial ring in AES (cont.)**

- We first compute $a(x) \bullet b(x)$:

$$a(x) = a_3x^3 + a_2x^2 + a_1x + a_0$$

$$b(x) = b_3x^3 + b_2x^2 + b_1x + b_0$$

$$c(x) = a(x) \bullet b(x)$$

$$= c_6x^6 + c_5x^5 + c_4x^4 + c_3x^3 + c_2x^2 + c_1x + c_0$$

$$c_0 = a_0 \bullet b_0$$

$$c_4 = a_3 \bullet b_1 \oplus a_2 \bullet b_2 \oplus a_1 \bullet b_3$$

$$c_1 = a_1 \bullet b_0 \oplus a_0 \bullet b_1$$

$$c_5 = a_3 \bullet b_2 \oplus a_2 \bullet b_3$$

$$c_2 = a_2 \bullet b_0 \oplus a_1 \bullet b_1 \oplus a_0 \bullet b_2$$

$$c_6 = a_3 \bullet b_3$$

$$c_3 = a_3 \bullet b_0 \oplus a_2 \bullet b_1 \oplus a_1 \bullet b_2 \oplus a_0 \bullet b_3$$

Mathematical Preliminaries

- **Multiplication in the polynomial ring in AES (cont.)**

$$\begin{aligned}d(x) &= a(x) \otimes b(x) \\ &= (a(x) \bullet b(x)) \bmod (x^4 + 1)\end{aligned}$$

Since

$$x^i \bmod (x^4 + 1) = x^{i \bmod 4}$$

we have

$$d(x) = c_3 x^3 + (c_6 + c_2)x^2 + (c_5 + c_1)x + (c_4 + c_0)$$

Mathematical Preliminaries

- **Multiplication in the polynomial ring in AES (cont.)**

Let $d(x) = d_3x^3 + d_2x^2 + d_1x + d_0$

We have $d_0 = (a_0 \bullet b_0) \oplus (a_3 \bullet b_1) \oplus (a_2 \bullet b_2) \oplus (a_1 \bullet b_3)$

$$d_1 = (a_1 \bullet b_0) \oplus (a_0 \bullet b_1) \oplus (a_3 \bullet b_2) \oplus (a_2 \bullet b_3)$$

$$d_2 = (a_2 \bullet b_0) \oplus (a_1 \bullet b_1) \oplus (a_0 \bullet b_2) \oplus (a_3 \bullet b_3)$$

$$d_3 = (a_3 \bullet b_0) \oplus (a_2 \bullet b_1) \oplus (a_1 \bullet b_2) \oplus (a_0 \bullet b_3)$$

Mathematical Preliminaries

- **Multiplication in the polynomial ring in AES (cont.)**
 - For a fixed polynomial $a(x)$, $d(x) = a(x) \otimes b(x)$ can be written in a matrix form:
(we get this simple matrix form because of the use of the simple reduction polynomial x^4+1 in this ring)

$$\begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} a_0 & a_3 & a_2 & a_1 \\ a_1 & a_0 & a_3 & a_2 \\ a_2 & a_1 & a_0 & a_3 \\ a_3 & a_2 & a_1 & a_0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Mathematical Preliminaries

- **Multiplication in the polynomial ring of AES (cont.)**

In AES, an invertible $a(x)$ is used:

$$a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$$

$$a^{-1}(x) = \{0b\}x^3 + \{0d\}x^2 + \{09\}x + \{0e\}$$

Mathematical Preliminaries

- **The polynomial ring in AES is different from the finite field $\text{GF}(2^8)$**
 - **Elements**
 - Element in the polynomial ring in AES (**coefficients in $\text{GF}(2^8)$**):
$$a_3x^3 + a_2x^2 + a_1x + a_0$$
 - Element in the finite field $\text{GF}(2^8)$ (**binary coefficients**):
$$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0$$
 - The indeterminate x in the ring notation is not related to the indeterminate x in the finite field notation
 - **Multiplication**
 - Polynomial ring in AES:
 - polynomial multiplication (coefficient multiplication is over $\text{GF}(2^8)$)
 - reduction polynomial $x^4 + 1$ is reducible
 - Finite field in AES
 - polynomial multiplication (coefficient multiplication is over $\text{GF}(2)$)
 - Reduction polynomial is irreducible and with degree 8

AES: state

16 bytes (the same as the block size)

Represented as a 2D array

– Four rows, four columns



AES: overall

State = Plaintext

AddRoundKey(State, RoundKey₀)

for $i = 1$ to $r-1$,

 SubBytes(State)

 ShiftRows(State)

 MixColumns(State)

 AddRoundKey(State, RoundKey _{i})

end for;

SubBytes(State)

ShiftRows(State)

~~MixColumns(State)~~

AddRoundKey(State, RoundKey _{r})

Ciphertext = state

$s_{0,0}$	$s_{0,1}$	$s_{0,2}$	$s_{0,3}$
$s_{1,0}$	$s_{1,1}$	$s_{1,2}$	$s_{1,3}$
$s_{2,0}$	$s_{2,1}$	$s_{2,2}$	$s_{2,3}$
$s_{3,0}$	$s_{3,1}$	$s_{3,2}$	$s_{3,3}$

$r-1$ rounds

The last round

$r+1$ round keys are
used

AES: SubByte

State = Plaintext

AddRoundKey(State, Key₀)

for $i = 1$ to $r-1$,

SubBytes(State)

ShiftRows(State)

MixColumns(State)

AddRoundKey(State, RoundKey _{i})

end for;

SubBytes(State)

ShiftRows(State)

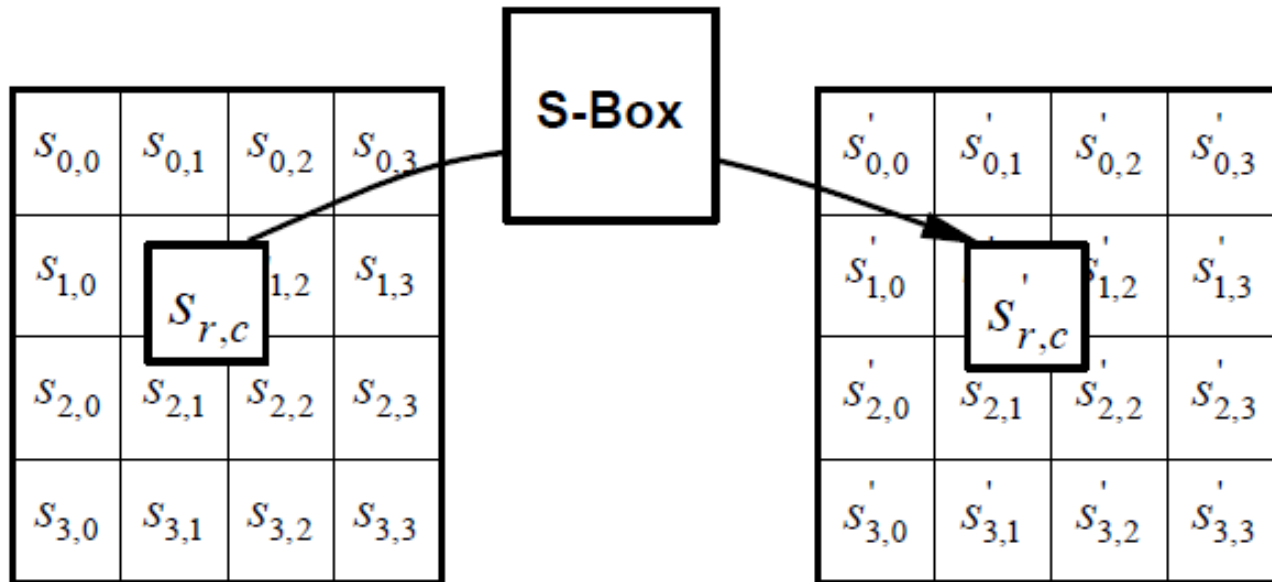
~~MixColumns(State)~~

AddRoundKey(State, RoundKey _{r})

Ciphertext = state

AES: SubByte

Apply Sbox to each byte in the state (parallel)
(Sbox is an substitution table)



AES: SubByte

- **SBox (substitution table) in AES**
 - 8-bit input; 8-bit output, invertible
 - Given input a , two steps to compute $b' = S(a)$
 - $b = a^{-1}$ in $GF(2^8)$ (0 is mapped to 0)
 - Write b in binary notation, apply the following transformation to b to compute the output b'

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

AES: ShiftRows

State = Plaintext

AddRoundKey(State, Key₀)

for $i = 1$ to $r-1$,

SubBytes(State)

ShiftRows(State)

MixColumns(State)

AddRoundKey(State, RoundKey _{i})

end for;

SubBytes(State)

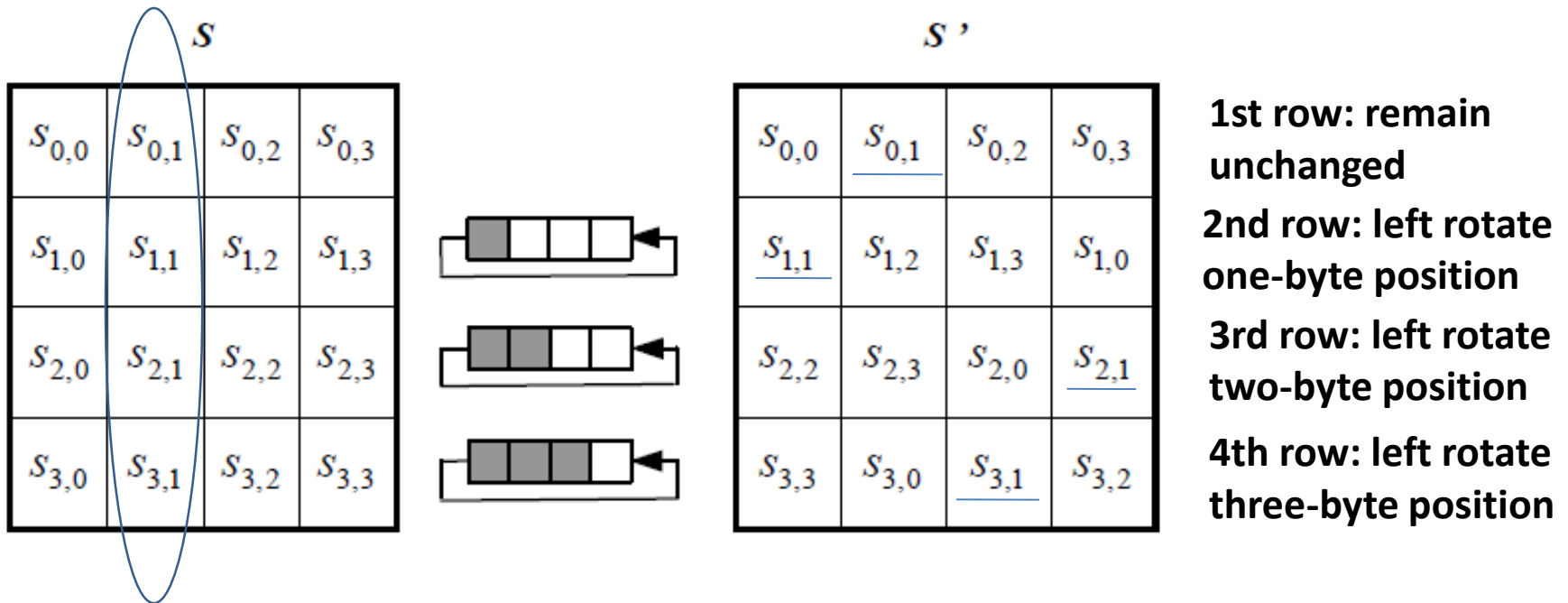
ShiftRows(State)

~~MixColumns(State)~~

AddRoundKey(State, RoundKey _{r})

Ciphertext = state

AES: ShiftRows



Reason for ShiftRows: four elements in one column relocated to 4 different columns after the ShiftRows

AES: MixColumns

State = Plaintext

AddRoundKey(State, Key₀)

for $i = 1$ to $r-1$,

SubBytes(State)

ShiftRows(State)

MixColumns(State)

AddRoundKey(State, RoundKey _{i})

end for;

SubBytes(State)

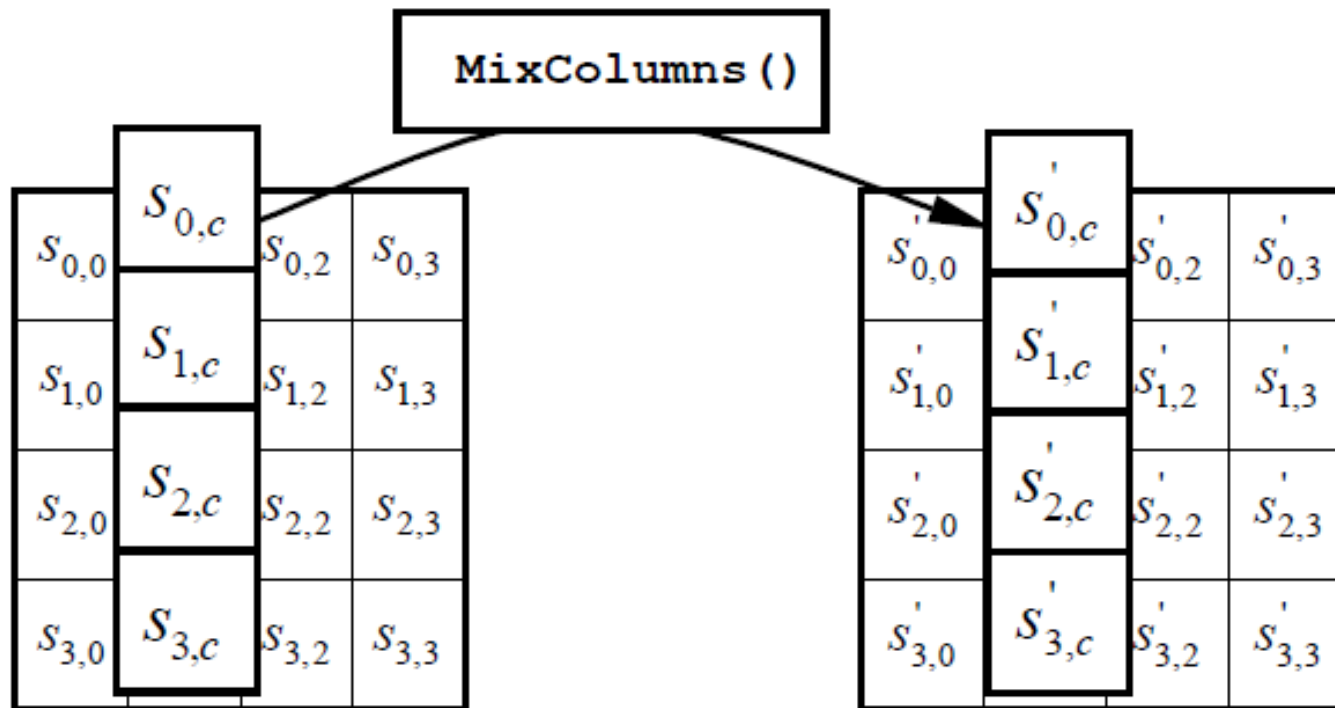
ShiftRows(State)

~~MixColumns(State)~~

AddRoundKey(State, RoundKey _{r})

Ciphertext = state

AES: MixColumns



$$a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$$

$$s'(x) = a(x) \otimes s(x)$$

AES: MixColumns

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

$$\begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} a_0 & a_3 & a_2 & a_1 \\ a_1 & a_0 & a_3 & a_2 \\ a_2 & a_1 & a_0 & a_3 \\ a_3 & a_2 & a_1 & a_0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$$

AES: MixColumns

- Important properties of MixColumn:
 - any input byte affects all four output bytes
 - If there are α non-zero bytes in the input;
 β non-zero bytes in the output;
then $(\alpha + \beta) \geq 5$

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

The AES MixColumn generates a maximum distance separable (MDS) code. For a 4-byte x , the distance between any $(x, \text{Mixcolumn}(x))$ is at least 5 over $\text{GF}(2^8)$.

AES: AddRoundKey

State = Plaintext

AddRoundKey(State, Key₀)

$l = 0$

for $i = 1$ to $r-1$,

 SubBytes(State)

 ShiftRows(State)

 MixColumns(State)

AddRoundKey(State, RoundKey _{i})

$l = 4i$

end for;

SubBytes(State)

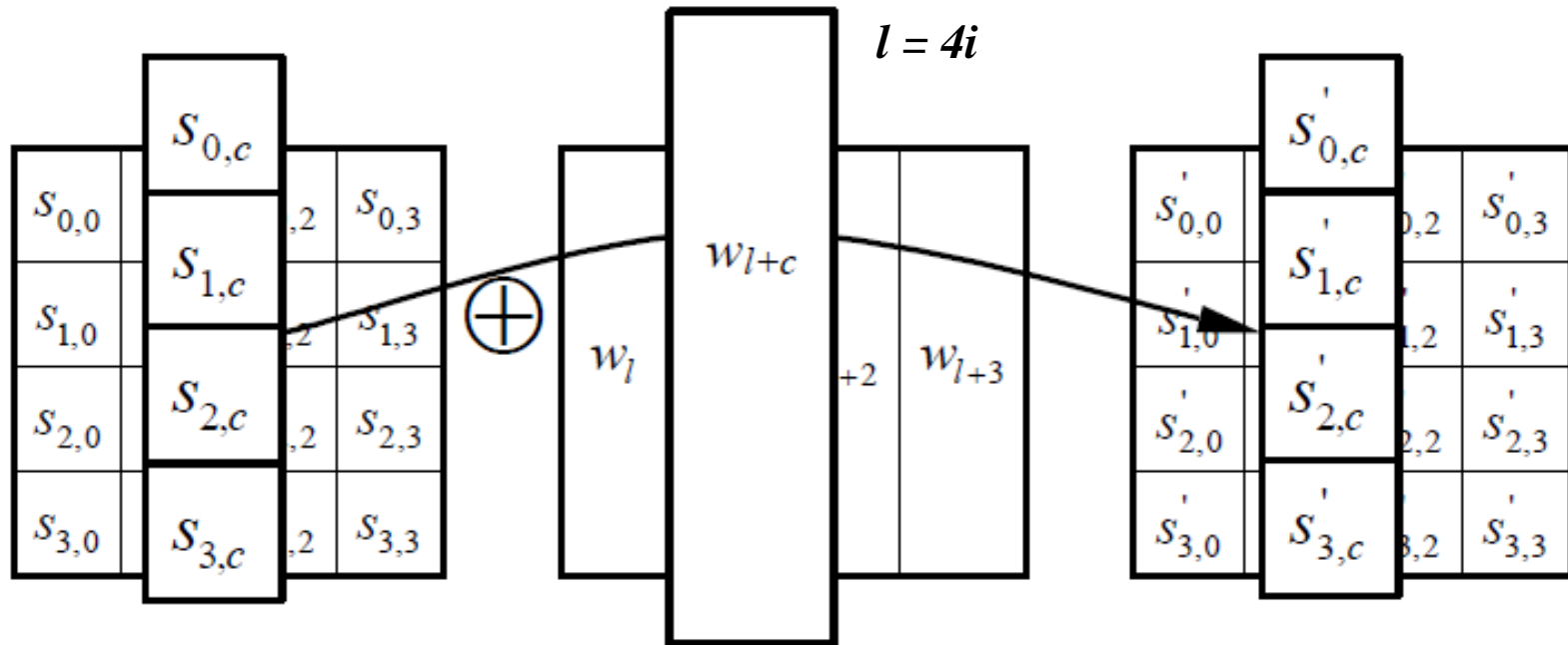
ShiftRows(State)

~~MixColumns(State)~~

AddRoundKey(State, RoundKey _{r})

Ciphertext = state

AES: AddRoundKey



AES: Key Schedule

- Each round key is 128-bit;
- Round keys are represented as an array of 32-bit words: $w[i]$
 - The first round key is $w[0], w[1], w[2], w[3]$
 - The second round key is $w[4], w[5], w[6], w[7]$
 -
- Secret key is represented as an array of bytes: $\text{key}[0], \text{key}[1], \text{key}[2], \dots$

AES: Key Schedule

- **Two functions are used in the key schedule**
 - **SubWord()**
 - 4-byte input
 - Apply Sbox to each input byte
 - **RotWord()**
 - Input: 4-byte [a0, a1, a2, a3]
 - Output: [a1, a2, a3, a0]

AES: Key Schedule

- A 32-bit round constant is used for generating each round key
 - $\text{Rcon}(i)$: $[2^{i-1}, 0, 0, 0]$, where 2^{i-1} is the power of 2 in the finite field $\text{GF}(2^8)$, $i > 0$.

$2^0 = 0\text{x}1$	$2^1 = 0\text{x}2$	$2^2 = 0\text{x}4$	$2^3 = 0\text{x}8$
$2^4 = 0\text{x}10$	$2^5 = 0\text{x}20$	$2^6 = 0\text{x}40$	$2^7 = 0\text{x}80$
$2^8 = 0\text{x}1\text{B}$	$2^9 = 0\text{x}36$	
 - We use different constants for different rounds to prevent slide-attack and some potential attacks (such as the invariance attack)

AES: Key Schedule

- **Example: AES-128 Key schedule**

$RCon[1] \leftarrow 01000000$
 $RCon[2] \leftarrow 02000000$
 $RCon[3] \leftarrow 04000000$
 $RCon[4] \leftarrow 08000000$
 $RCon[5] \leftarrow 10000000$
 $RCon[6] \leftarrow 20000000$
 $RCon[7] \leftarrow 40000000$
 $RCon[8] \leftarrow 80000000$
 $RCon[9] \leftarrow 1B000000$
 $RCon[10] \leftarrow 36000000$

Round constants

$W[0] = (K[0], K[1], K[2], K[3])$
 $W[1] = (K[4], K[5], K[6], K[7])$
 $W[2] = (K[8], K[9], K[10], K[11])$
 $W[3] = (K[12], K[13], K[14], K[15])$

for $i \leftarrow 0$ **to** 3

do $w[i] \leftarrow (key[4i], key[4i + 1], key[4i + 2], key[4i + 3])$ } Load the key into $w[]$

for $i \leftarrow 4$ **to** 43

do $\left\{ \begin{array}{l} temp \leftarrow w[i - 1] \\ \text{if } i \equiv 0 \pmod{4} \\ \quad \text{then } temp \leftarrow SUBWORD(ROTWORD(temp)) \oplus RCon[i/4] \\ w[i] \leftarrow w[i - 4] \oplus temp \end{array} \right.$

return $(w[0], \dots, w[43])$

11 round keys

AES: all the key schedules

```
KeyExpansion(byte key[4*Nk], word w[Nb*(Nr+1)], Nk)
begin
    word temp

    i = 0

    while (i < Nk)
        w[i] = word(key[4*i], key[4*i+1], key[4*i+2], key[4*i+3])
        i = i+1
    end while

    i = Nk

    while (i < Nb * (Nr+1))
        temp = w[i-1]
        if (i mod Nk = 0)
            temp = SubWord(RotWord(temp)) xor Rcon[i/Nk]
        else if (Nk > 6 and i mod Nk = 4)
            temp = SubWord(temp)
        end if
        w[i] = w[i-Nk] xor temp
        i = i + 1
    end while
end
```

	Key Length (<i>Nk</i> words)	Block Size (<i>Nb</i> words)	Number of Rounds (<i>Nr</i>)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

AES: Decryption

(functions are different from that in encryption)

State = ciphertext

AddRoundKey(State, RoundKey_{*r*})

for *i* = 1 to *r*-1,

 InvShiftRows(State)

 InvSubBytes(State)

 AddRoundKey(State, RoundKey_{*r-i*})

 InvMixColumns(State)

end for;

r-1 rounds

InvShiftRows(State)

InvSubBytes(State)

AddRoundKey(State, RoundKey₀)

~~InvMixColumns(State)~~

The last round

plaintext = state

r+1 round keys are
used

AES: Encryption & Decryption

State = Plaintext

AddRoundKey(State, RoundKey₀)

for $i = 1$ to $r-1$,
 SubBytes(State)
 ShiftRows(State)
 MixColumns(State)
 AddRoundKey(State, RoundKey _{i})
end for;

SubBytes(State)
ShiftRows(State)
~~MixColumns(State)~~
AddRoundKey(State, RoundKey _{r})

ciphertext = state

State = ciphertext

AddRoundKey(State, RoundKey _{r})

for $i = 1$ to $r-1$,
 InvShiftRows(State)
 InvSubBytes(State)
 AddRoundKey(State, RoundKey _{$r-i$})
 InvMixColumns(State)
end for;

InvShiftRows(State)
InvSubBytes(State)
AddRoundKey(State, RoundKey₀)
~~InvMixColumns(State)~~

plaintext = State

AES Implementation

- **In practice, the implementation of the cipher must be correct**
 - **How to check the correctness of the implementation?**
 - Using the test vectors provided in the standard
<http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>
- **Efficient in**
 - **Hardware**
 - **Embedded system (normally 8-bit processors)**
 - **Software (32-bit processor)**
 - **Precompute Sbox and MixColumn operations, store the results in tables. Using table lookup for fast software computation of Sbox and MixColumn.**
 - **Around 20 clock cycles/byte**

AES Implementation

- **Starting from 2009, Intel implemented the AES round function and step function of the key schedule in CPUs (part of AVX instruction set)**
 - **Mainly to address the cache-timing attack on AES**
 - **For software implementation on CPUs with memory cache, the table lookup timing may vary depending on whether the element of the table is in the memory cache. Such information can be used to recover the key**
 - **Results in extremely fast AES**
 - **around 4 clock cycles/byte**
 - **AMD follows Intel**

Summary

- **Mathematical preliminaries**
 - **Extended Euclidean Algorithm**
 - **$\text{GF}(2^8)$**
 - **Polynomial rings (with coefficients in $\text{GF}(2^8)$)**
- **AES**
 - **Encryption**
 - **Substitution-Permutation Network**
 - **Round function**
 - different round numbers for different key sizes
 - **Key schedule**
 - different for different key sizes