Advanced Encryption Standard (AES)

Math 4175

§4.6c. Advanced Encryption Standard

When DES was originally proposed as a standard, one of the criticisms was the relatively short size of the key space.

On January 2, 1997, NIST began the process of selecting a replacement for DES, which would be called the Advanced Encryption Standard (AES).

The basic criteria for the new standard were:

- AES should have a block length of 128 bits,
- three allowable key length of 128, 192 and 256 bits,
- and should be available worldwide on a royalty-free basis.

Of the 21 submitted cryptosystems, 15 met all the necessary criteria and were accepted as AES candidates.

§4.6c. Advanced Encryption Standard

AES candidates were evaluated for their suitability according to three main criteria:

- Security: which was absolutely essential, and algorithm found not to be secure would not be considered further.
- Cost: computational efficiency with effective speed and memory, implementation of software, hardware and smart cards.
- algorithm and implementation characteristics: flexibility and algorithm simplicity.

Five finalists (MARS, RC6, Rijindael, Serpent, and Twofish), who were all felt to be secure, were selected. On October 2, 2002 Rijndael (Dutch pronounciation: Rhine Dahl, developed by two Belgian cryptographers, Vincent Rijmen and Joan Daemen) was selected to be the AES, because its security, performance, efficiency, implementability and flexibility were judged to be superior to other finalists.

§4.6c. Advanced Encryption Standard

The selection process for the AES was notable for its openness and its international flavor. Submissions were evaluated by the cryptographic community with public discussions.

AES is built upon the basic SPN architecture.

- Block size is 128 bits.
- There are three possible key lengths, namely 128, 192 and 256 bits.
- Number of rounds $N = 10$, 12, or 14, depends respectively on the key length.
- From a key, $N + 1$ round keys are generated by a key schedule.

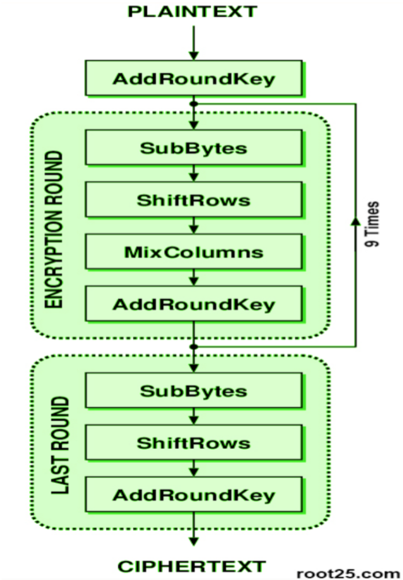
§4.6.1. Advanced Encryption Standard

We will give a description of AES with 10 rounds.

Four layers used to form rounds:

- SubByte (SB) – A non-linear S-box
- ShiftRow (SR) – An index permutation
- MixColumn (MC) – A linear "hybrid" transformation
- AddRoundKey (ARK) – Standard key mixing

We will give precise descriptions of all of these operations and discuss the construction of key schedule. Of course, the standard key mixing is the bit by bit X-or operation.



root25.com

AES Encyption

1. ARK, using 0th round key
2. Nine "standard" rounds of SB, SR, MC, ARK, using round keys 1 through 9
3. Tenth "special" round of SB, SR, ARK

§4.6c. SubByte

Each input of 128 bits to a normal round can be represented more efficiently as follows:

Recall that 1 byte is equal to 8 bits. So the 128 input bits to a normal round can be grouped into 16 bytes:

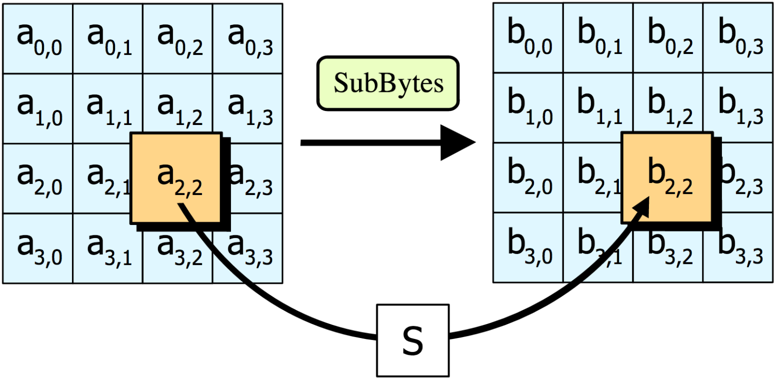
This can be arranged in a 4 × 4 matrix:

$$A = \begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} = \begin{bmatrix} x_0 & x_4 & x_8 & x_{12} \\ x_1 & x_5 & x_9 & x_{13} \\ x_2 & x_6 & x_{10} & x_{14} \\ x_3 & x_7 & x_{11} & x_{15} \end{bmatrix}$$

Since each $a_{i,j}$ has 8 bits, it can be represented by two hexadecimal notations. For example, the string 01010011 is **53** in hexadecimal notation.

§4.6c. SubByte

In the SubByte step, AES S-box carry each byte in the current state matrix to a corresponding byte in the next state matrix.


$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
SubBytes
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$
S

§4.6c. SubByte

In contrast to S-boxes in DES, which are apparently “random” substitutions, the AES S-box can be defined algebraically by using the field operations just like S-AES.

Let $a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0$ be a byte, that is, a binary string of length 8.

We can identify this byte with an element in the field F as follows:

$$y = a_7\mathbf{x}^7 + a_6\mathbf{x}^6 + a_5\mathbf{x}^5 + a_4\mathbf{x}^4 + a_3\mathbf{x}^3 + a_2\mathbf{x}^2 + a_1\mathbf{x} + a_0.$$

We now describe the construction of the permutation

$$\pi_S : \{0,1\}^8 \to \{0,1\}^8$$

that will be part of the SubByte.

§3.6c. SubByte

The permutation π_S incorporates operations in the finite field

$$F = \mathbb{Z}_2[\mathbf{x}]/(\mathbf{x}^8 + \mathbf{x}^4 + \mathbf{x}^3 + \mathbf{x} + 1).$$

Let $a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0$ be a binary string of length 8. We identify this byte with the field element

$$y = a_7\mathbf{x}^7 + a_6\mathbf{x}^6 + a_5\mathbf{x}^5 + a_4\mathbf{x}^4 + a_3\mathbf{x}^3 + a_2\mathbf{x}^2 + a_1\mathbf{x} + a_0.$$

Let $y^{-1} = b_7\mathbf{x}^7 + b_6\mathbf{x}^6 + b_5\mathbf{x}^5 + b_4\mathbf{x}^4 + b_3\mathbf{x}^3 + b_2\mathbf{x}^2 + b_1\mathbf{x} + b_0$ be the inverse of y in F when $y \neq 0$. (with $0^{-1} = 0$).

§4.6c. SubByte

Then, compute

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix} \oplus \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \end{pmatrix}$$

We define $\pi_S(a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0) = w_7 w_6 w_5 w_4 w_3 w_2 w_1 w_0$.

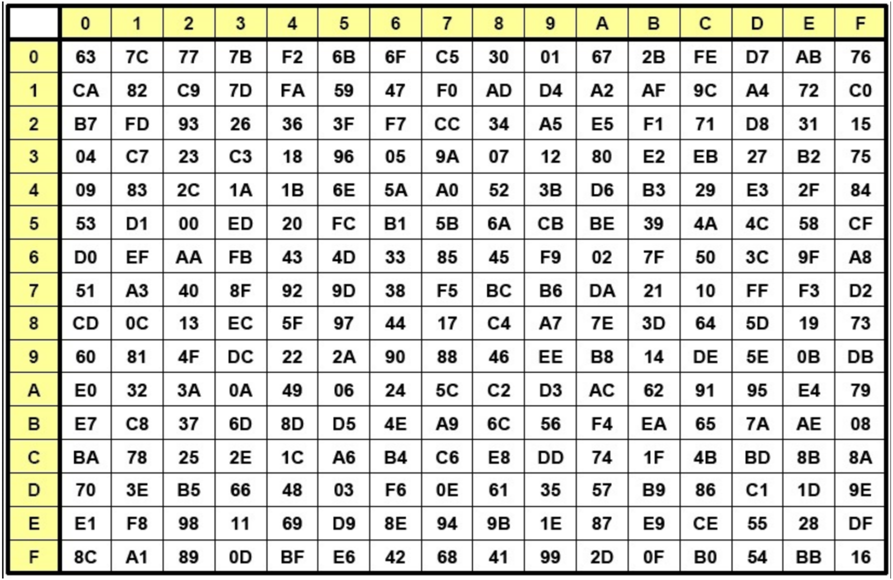
§4.6c. SubByte

Example: Suppose that we begin with the byte 01010011, which is **53** in hexadecimal, that represents the field element $y = \mathbf{x}^6 + \mathbf{x}^4 + \mathbf{x} + 1$. One can verify that the multiplicative inverse in the field F is $y^{-1} = \mathbf{x}^7 + \mathbf{x}^6 + \mathbf{x}^3 + \mathbf{x}$. Therefore, in binary notation, we have $y^{-1} = 11001010$. Then

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} \oplus \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Thus $\pi_S(01010011) = 11101101$, which in hexadecimal notation means $\pi_S(\mathbf{53}) = \mathbf{ED}$.

This computation can be checked by verifying that the entry in row 5 and column 3 of the table is ED.



	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

§4.6c. Shift Row

ShiftRow: Let the output of SubByte be:

$$B = \begin{bmatrix} b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{1,0} & b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3} \end{bmatrix}$$

For $0 \leq i \leq 3$, the i-th row of the matrix is left-shifted cyclically to yield:

$$\begin{bmatrix} b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{1,1} & b_{1,2} & b_{1,3} & b_{1,0} \\ b_{2,2} & b_{2,3} & b_{2,0} & b_{2,1} \\ b_{3,3} & b_{3,0} & b_{3,1} & b_{3,2} \end{bmatrix} = \begin{bmatrix} c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{1,0} & c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,0} & c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,0} & c_{3,1} & c_{3,2} & c_{3,3} \end{bmatrix} = C$$

C is the output of the ShiftRow.

§4.6c. MixColumn

MixColumn: Still viewing a byte as an element of F, we perform the following matrix multiplication to get:

$$D = \begin{bmatrix} d_{0,0} & d_{0,1} & d_{0,2} & d_{0,3} \\ d_{1,0} & d_{1,1} & d_{1,2} & d_{1,3} \\ d_{2,0} & d_{2,1} & d_{2,2} & d_{2,3} \\ d_{3,0} & d_{3,1} & d_{3,2} & d_{3,3} \end{bmatrix}$$

$$= \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{1,0} & c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,0} & c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,0} & c_{3,1} & c_{3,2} & c_{3,3} \end{bmatrix}$$

D is the output of the MixColumn Layer.
Note: $02 = 00000010$ corresponds to x, $03 = 00000011$ corresponds to $x + 1$ and $01 = 00000001$ corresponds to 1.

§4.6c. AddRoundKey

AddRoundKey: A round key is generated from the given 128 bit key, and bitwise XOR-ed with output D of MixColumn:

$$\begin{aligned} E &= \begin{bmatrix} e_{0,0} & e_{0,1} & e_{0,2} & e_{0,3} \\ e_{1,0} & e_{1,1} & e_{1,2} & e_{1,3} \\ e_{2,0} & e_{2,1} & e_{2,2} & e_{2,3} \\ e_{3,0} & e_{3,1} & e_{3,2} & e_{3,3} \end{bmatrix} \\ &= \begin{bmatrix} d_{0,0} & d_{0,1} & d_{0,2} & d_{0,3} \\ d_{1,0} & d_{1,1} & d_{1,2} & d_{1,3} \\ d_{2,0} & d_{2,1} & d_{2,2} & d_{2,3} \\ d_{3,0} & d_{3,1} & d_{3,2} & d_{3,3} \end{bmatrix} \oplus \begin{bmatrix} k_{0,0} & k_{0,1} & k_{0,2} & k_{0,3} \\ k_{1,0} & k_{1,1} & k_{1,2} & k_{1,3} \\ k_{2,0} & k_{2,1} & k_{2,2} & k_{2,3} \\ k_{3,0} & k_{3,1} & k_{3,2} & k_{3,3} \end{bmatrix} \end{aligned}$$

§4.6c. Key Schedule

Now it only remains to describe the key schedule for AES.

- We describe how to construct the key schedule for the 10-round version of AES.
- The key schedule for 12 and 14 versions are similar with minor modifications.
- We need 11 round keys, each of which consists of 16 bytes (128 bits).
- The key scheduling algorithm is word oriented, where a word here consists of 4 bytes (32 bits).
- The original key consists of 128 bits, which are arranged into a 4×4 matrix of bytes (loaded by down columns, just like the plaintext).
- Label the four columns as $W(0), W(1), W(2), W(3)$, where $W(i)$ stands for i-th Word. This matrix is the round key for the 0th round.
- Then this matrix is expanded by adjoining 40 more columns, as follows.

§4.6c. Key Schedule

- The new columns are generated recursively.
- Suppose that columns up to $W(i-1)$ have been defined.
- If i is not a multiple of 4, then

$$W(i) = W(i-4) \oplus W(i-1).$$

- Otherwise, we write $W(i-1) = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$ and define $W(i)$ as follows:

§4.6c. Key Schedule

- $$W(i) = W(i-4) \oplus \begin{bmatrix} S(b) \oplus x^{(i-4)/4} \\ S(c) \\ S(d) \\ S(a) \end{bmatrix}$$

  (where S is the S-box defined in SubByte and $x^{(i-4)/4}$ is considered as an element in the field F.)

- The **round key** for the ith round consists of the columns

$$W(4i), \quad W(4i+1), \quad W(4i+2), \quad W(4i+3)$$

§4.6c. Analysis of AES

- We have now described all the operations required to perform an encryption operation in the AES.
- In order to decrypt, one needs to perform all the operations in the reverse order and use the key schedule in reverse order.
- The operations ShiftRows, SubBytes, MixColumns, and AddRoundKey must be replaced by their inverse operations. AddRoundKey is its own inverse.
- Indeed, it may be more efficient to simply construct an 'inverse AES cipher' to perform the decryption, in stead of performing decryption case by case.
- The AES is secure against all known attacks. There are apparently no known attacks on AES that are faster than brute force exhaustive search, which would take billions of years on current hardware.
- In June 2003, the U.S. Government announced that AES could be used to protect classified information.