Advance Encryption Standard

Topics

- **Origin of AES**
- Basic AES
- Inside Algorithm
- Final Notes

Origins

- A replacement for DES was needed
  - Key size is too small
- Can use Triple-DES – but slow, small block
- US NIST issued call for ciphers in 1997
- 15 candidates accepted in Jun 98
- 5 were shortlisted in Aug 99

The AES Cipher - Rijndael

- Rijndael was selected as the AES in Oct-2000
  - Designed by Vincent Rijmen and Joan Daemen in Belgium
  - Issued as FIPS PUB 197 standard in Nov-2001
- An **iterative** rather than **Feistel** cipher
  - processes data as block of 4 columns of 4 bytes (128 bits)
  - operates on entire data block in every round
- Rijndael design:
  - simplicity
  - has 128/192/256 bit keys, 128 bits data
  - resistant against known attacks
  - speed and code compactness on many CPUs

V. Rijmen

J. Daemen

AES Competition Requirements

- Private key symmetric block cipher
- 128-bit data, 128/192/256-bit keys
- Stronger & faster than Triple-DES
- Provide full specification & design details
- Both C & Java implementations

AES Evaluation Criteria

- initial criteria:
  - security – effort for practical cryptanalysis
  - cost – in terms of computational efficiency
  - algorithm & implementation characteristics

- final criteria
  - general security
  - ease of software & hardware implementation
  - implementation attacks
  - flexibility (in en/decrypt, keying, other factors)

Features of AES

- SP Network: It works on an SP network structure rather than a Feistel cipher structure, as seen in the case of the DES algorithm.
- Key Expansion: It takes a single key up during the first stage, which is later expanded to multiple keys used in individual rounds.
- Byte Data: The AES encryption algorithm does operations on byte data instead of bit data. So it treats the 128-bit block size as 16 bytes during the encryption procedure.
- Key Length: The number of rounds to be carried out depends on the length of the key being used to encrypt data. The 128-bit key size has ten rounds, the 192-bit key size has 12 rounds, and the 256-bit key size has 14 rounds

High Level Description

Key Expansion

- Round keys are derived from the cipher key using Rijndael's key schedule

Initial Round

- AddRoundKey : Each byte of the state is combined with the round key using bitwise xor

Rounds

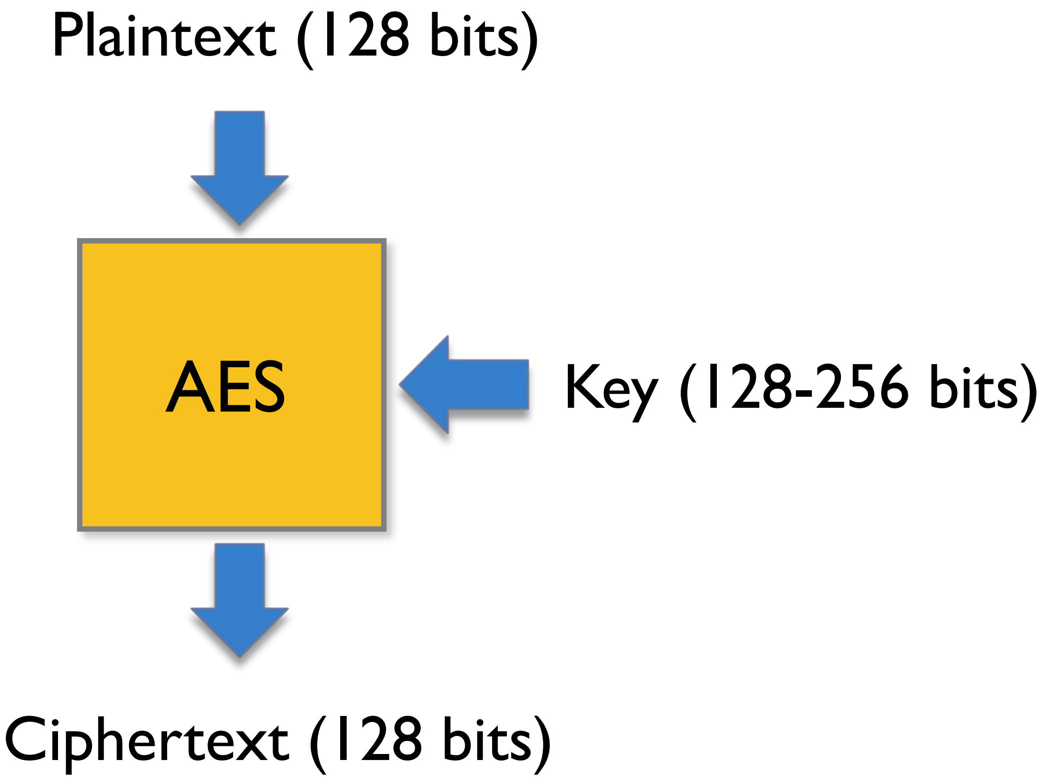
- SubBytes : non-linear substitution step
- ShiftRows : transposition step
- MixColumns : mixing operation of each column.
- AddRoundKey

Final Round

- SubBytes
- ShiftRows
- AddRoundKey

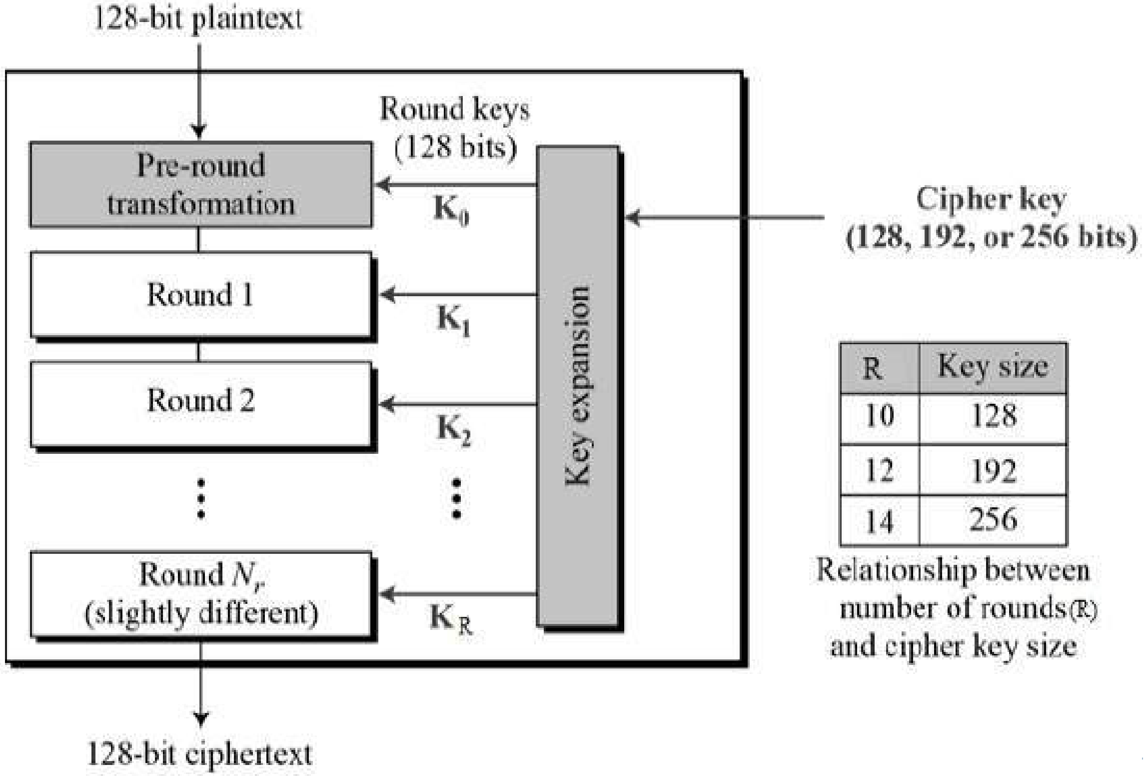
No MixColumns

AES Conceptual Scheme



Block Cipher

The Advanced Encryption Standard (AES), also called Rijndael, is a symmetric block cipher with a block length of 128 bits and support for key lengths of 128, 192, and 256 bits. It was published by NIST (National Institute of Standards and Technology) in 2001. Here, we assume a key length of 128 bits, which is likely to be the one most commonly implemented.



R	Key size
10	128
12	192
14	256

Relationship between number of rounds(R) and cipher key size

AES Block Cipher

The AES Algorithm:

- AES operates on a 4 × 4 column-wise order array of bytes, called the *state*. For instance, if there are 16 bytes, these bytes are represented as this two-dimensional array: $$\begin{bmatrix} b_0 & b_4 & b_8 & b_{12} \\ b_1 & b_5 & b_9 & b_{13} \\ b_2 & b_6 & b_{10} & b_{14} \\ b_3 & b_7 & b_{11} & b_{15} \end{bmatrix}$$

- The key size used for an AES cipher specifies the number of transformation rounds that convert the plaintext into the ciphertext . The number of rounds are as follows:

10 rounds for 128-bit keys.
12 rounds for 192-bit keys.
14 rounds for 256-bit keys.

- Each round consists of several processing steps, including one that depends on the encryption key itself. A set of reverse rounds are applied to transform ciphertext back into the original plaintext using the same encryption key.

AES Block Cipher

The AES Encryption Algorithm:

- The AES algorithm can be broken into three phases: the initial round, the main rounds, and the final round. All of the phases use the same sub-operations in different combinations as follows:

  Initial Round

  AddRoundKey

  Main Rounds (1,2…Nr-1)

  SubBytes

  ShiftRows

  MixColumns

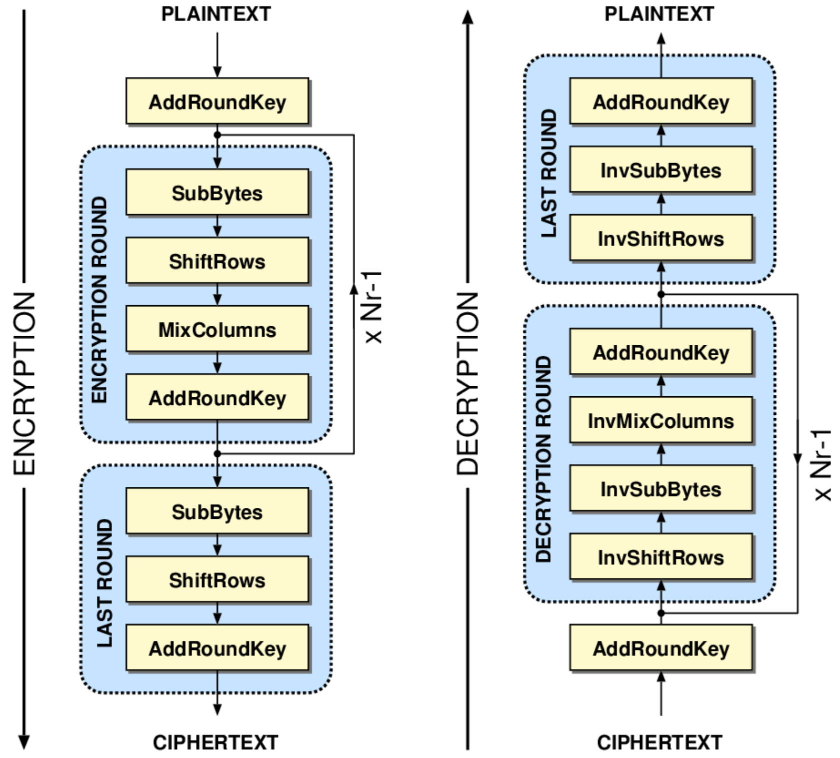
  AddRoundKey

  Final Round (Nr)

  SubBytes

  ShiftRows

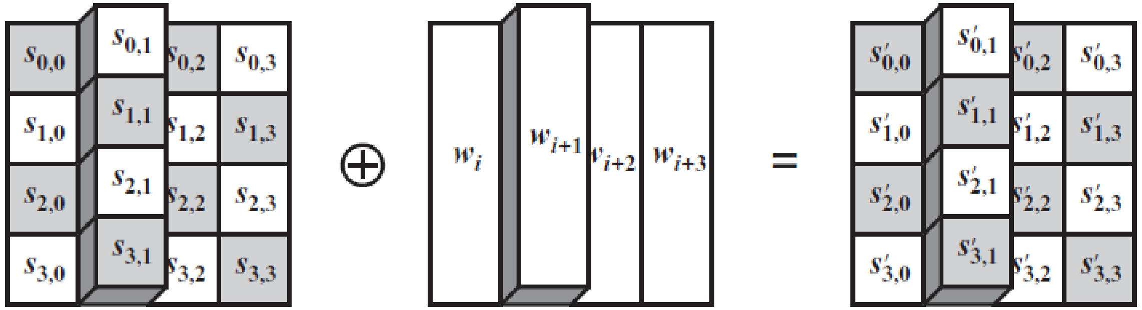
  AddRoundKey



Note that in the above figure, KeyExpansion: round keys are derived from the cipher key using key expansion algorithm. AES requires a separate 128-bit round key block for each round plus one more.

AES Block Cipher

AddRoundKey: In this operation, the 128 bits of **State** are bitwise XORed with the 128 bits of the round key. Here is an example where the first matrix is State, and the second matrix is the round key.



e.g.

$$\begin{pmatrix} 54 & 4F & 4E & 20 \\ 77 & 6E & 69 & 54 \\ 6F & 65 & 6E & 77 \\ 20 & 20 & 65 & 6F \end{pmatrix} \oplus \begin{pmatrix} 54 & 73 & 20 & 67 \\ 68 & 20 & 4B & 20 \\ 61 & 6D & 75 & 46 \\ 74 & 79 & 6E & 75 \end{pmatrix} \begin{pmatrix} 00 & 3C & 6E & 47 \\ 1F & 4E & 22 & 74 \\ 0E & 08 & 1B & 31 \\ 54 & 59 & 0B & 1A \end{pmatrix}$$

$$\text{e.g., } 69 \oplus 4B = 22$$

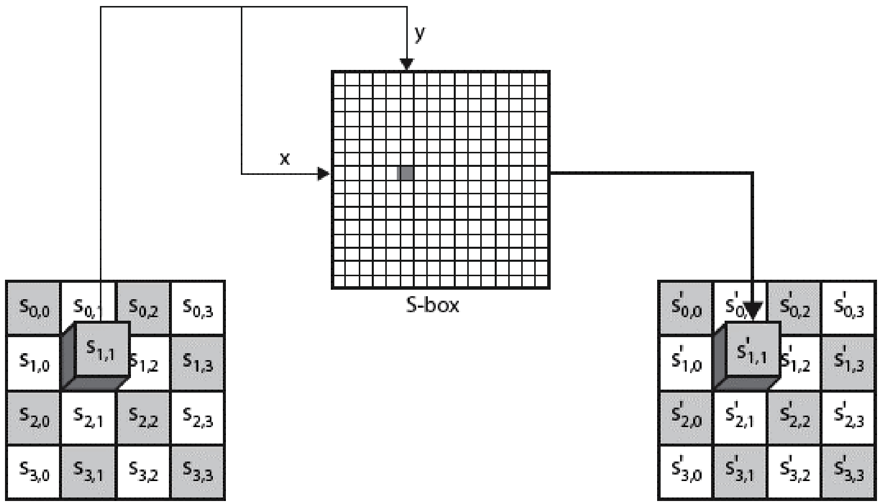
$$\begin{array}{r} 0110\ 1001 \\ 0100\ 1011 \\ \hline 0010\ 0010 \end{array}$$

AES Block Cipher

SubBytes: A nonlinear substitution step where each entry (byte) of the current state matrix is substituted by a corresponding entry in the AES S-Box. For instance: byte (6E) is substituted by the entry of the S-Box in row 6 and column E, i.e., by (9F). (The byte input is broken into two 4-bit halves. The first half determines the row and the second half determines the column).

e.g.:

$$\text{state} = \begin{pmatrix} 00 & 3C & 6E & 47 \\ 1F & 4E & 22 & 74 \\ 0E & 08 & 1B & 31 \\ 54 & 59 & 0B & 1A \end{pmatrix} \Rightarrow \text{S\_box(State)} = \begin{pmatrix} 63 & EB & 9F & A0 \\ C0 & 2F & 93 & 92 \\ AB & 30 & AF & C7 \\ 20 & CB & 2B & A2 \end{pmatrix}$$



AES Encryption Cipher

ShiftRows: A transposition step where the four rows of the state are shifted cyclically to the left by offsets of 0, 1, 2, and 3.

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

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

e.g.:

$$\begin{pmatrix} 63 & EB & 9F & A0 \\ C0 & 2F & 93 & 92 \\ AB & 30 & AF & C7 \\ 20 & CB & 2B & A2 \end{pmatrix} \Rightarrow \begin{pmatrix} 63 & EB & 9F & A0 \\ 2F & 93 & 92 & C0 \\ AF & C7 & AB & 30 \\ A2 & 20 & CB & 2B \end{pmatrix}$$

AES Block Cipher

MixColumns: a linear mixing operation which multiplies fixed matrix against current State Matrix:

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

$$\begin{aligned} s'_{0,j} &= (2 \cdot s_{0,j}) \oplus (3 \cdot s_{1,j}) \oplus s_{2,j} \oplus s_{3,j} \\ s'_{1,j} &= s_{0,j} \oplus (2 \cdot s_{1,j}) \oplus (3 \cdot s_{2,j}) \oplus s_{3,j} \\ s'_{2,j} &= s_{0,j} \oplus s_{1,j} \oplus (2 \cdot s_{2,j}) \oplus (3 \cdot s_{3,j}) \\ s'_{3,j} &= (3 \cdot s_{0,j}) \oplus s_{1,j} \oplus s_{2,j} \oplus (2 \cdot s_{3,j}) \end{aligned}$$

Unlike standard matrix multiplication, MixColumns performs matrix multiplication as per Galois Field (2^8).

e.g.:

$$\begin{pmatrix} 02\,03\,01\,01 \\ 01\,02\,03\,01 \\ 01\,01\,02\,03 \\ 03\,01\,01\,02 \end{pmatrix} \begin{pmatrix} 63 & EB & 9F & A0 \\ 2F & 93 & 92 & C0 \\ AF & C7 & AB & 30 \\ A2 & 20 & CB & 2B \end{pmatrix} = \begin{pmatrix} BA & 84 & E8 & 1B \\ 75 & A4 & 8D & 40 \\ F4 & 8D & 06 & 7D \\ 7A & 32 & 0E & 5D \end{pmatrix}$$

AES Block Cipher

The AES Decryption Algorithm:

❑ AddRoundKey:

Add Roundkey transformation is identical to the forward add round key transformation, because the XOR operation is its own inverse.

❑ Inverse SubBytes:

This operation can be performed using the inverse S-Box. It is read identically to the S-Box matrix.

❑ InvShiftRows:

Inverse Shift Rows performs the circular shifts in the opposite direction for each of the last three rows, with a one-byte circular right shift for the second row, and so on.

❑ InvMixColumns:

The inverse mix column transformation is defined by the following matrix multiplication in Galois Field (2^8):

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

Example

Plaintext – Two One Nine Two

T	w	o		O	n	e		N	i	n	e		T	w	o
54	77	6F	20	4F	6E	65	20	43	69	6E	25	20	54	77	6F

Plaintext in Hex Format
54 77 6F 20 4F 6E 65 20 43 69 6E 25 20 54 77 6F

Encryption Key – Thats my Kung Fu

T	h	a	t	s		m	y		K	u	n	g		F	u
54	68	61	74	73	20	6D	79	20	4B	75	6E	67	20	46	75

Encryption Key in Hex Format
54 68 61 74 73 20 6D 79 20 4B 75 6E 67 20 46 75

Key generation

Keys generated for every round

- Round 0: 54 68 61 74 73 20 6D 79 20 4B 75 6E 67 20 46 75
- Round 1: E2 32 FC F1 91 12 91 88 B1 59 E4 E6 D6 79 A2 93
- Round 2: 56 08 20 07 C7 1A B1 8F 76 43 55 69 A0 3A F7 FA
- Round 3: D2 60 0D E7 15 7A BC 68 63 39 E9 01 C3 03 1E FB
- Round 4: A1 12 02 C9 B4 68 BE A1 D7 51 57 A0 14 52 49 5B
- Round 5: B1 29 3B 33 05 41 85 92 D2 10 D2 32 C6 42 9B 69
- Round 6: BD 3D C2 B7 B8 7C 47 15 6A 6C 95 27 AC 2E 0E 4E
- Round 7: CC 96 ED 16 74 EA AA 03 1E 86 3F 24 B2 A8 31 6A
- Round 8: 8E 51 EF 21 FA BB 45 22 E4 3D 7A 06 56 95 4B 6C
- Round 9: BF E2 BF 90 45 59 FA B2 A1 64 80 B4 F7 F1 CB D8
- Round 10: 28 FD DE F8 6D A4 24 4A CC C0 A4 FE 3B 31 6F 26

AddRound

54	4F	4E	20
77	6E	69	54
6F	65	6E	77
20	20	65	6F

Plaintext

$\oplus$

XOR

54	73	20	67
68	20	4B	20
61	6D	75	46
74	79	6E	75

Round 0 Key

00	3C	63	47
1F	4E	22	74
0E	08	1B	31
54	59	0B	1A

New State Array

Sub-Bytes and Shift Row

63	EB	9F	A0
C0	2F	93	92
AB	30	AF	C7
20	CB	2B	A2

New State Array

63	EB	9F	A0
C0	2F	93	92
AB	30	AF	C7
20	CB	2B	A2

Old State Array

63	EB	9F	A0
2F	93	92	C0
AF	C7	AB	30
A2	20	CB	2B

New State Array

Mix Columns

$$\underbrace{\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}}_{\text{Constant Matrix}} \times \underbrace{\begin{bmatrix} 63 & EB & 9F & A0 \\ 2F & 93 & 92 & C0 \\ AF & C7 & AB & 30 \\ A2 & 20 & CB & 2B \end{bmatrix}}_{\text{Old State Array}} \longrightarrow \underbrace{\begin{bmatrix} BA & 84 & E8 & 1B \\ 75 & A4 & 8D & 40 \\ F4 & 8D & 06 & 7D \\ 7A & 32 & 0E & 5D \end{bmatrix}}_{\text{New State Array}}$$

Add Round Key

BA	84	E8	1B
75	A4	8D	40
F4	8D	06	7D
7A	32	0E	5D

Old State Array

XOR

E2	91	B1	D6
32	12	59	79
FC	91	E4	A2
F1	88	E6	93

Round 1 Key

58	15	59	CD
47	B6	D4	39
08	1C	E2	DF
8B	BA	E8	CE

New State Array

Final Round

Final State Array after Round 10

29	57	40	1A
C3	14	22	02
50	20	99	D7
5F	F6	B3	3A

AES Final Output

29 C3 50 5F 57 14 20 F6 40 22 99 B3 1A 02 D7 3A

↓

Ciphertext

AES Security

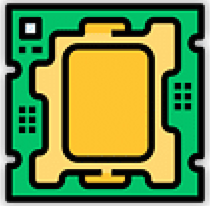
- AES was designed after DES.
- Most of the known attacks on DES were already tested on AES.
- Brute-Force Attack
  - AES is definitely more secure than DES due to the larger-size key.
- Statistical Attacks
  - Numerous tests have failed to do statistical analysis of the ciphertext
- Differential and Linear Attacks
  - There are no differential and linear attacks on AES as yet.

Applications

Wireless security

Encrypted browsing

General file encryption



Processor security

DES vs AES

DES Algorithm	AES Algorithm
Key Length - 56 bits	Key Length - 128, 192, 256 bits
Block Size - 64 bits	Block size - 128 bits
Fixed no. of rounds	No. of rounds dependent on key length
Slower and less secure	Faster and more secure