Advanced Encryption Standard (AES)

Raj Jain
Washington University in Saint Louis
Saint Louis, MO 63130
Jain@cse.wustl.edu

Audio/Video recordings of this lecture are available at:

http://www.cse.wustl.edu/~jain/cse571-11/

Overview

1. AES Structure
2. AES Round Function
3. AES Key Expansion
4. AES Decryption

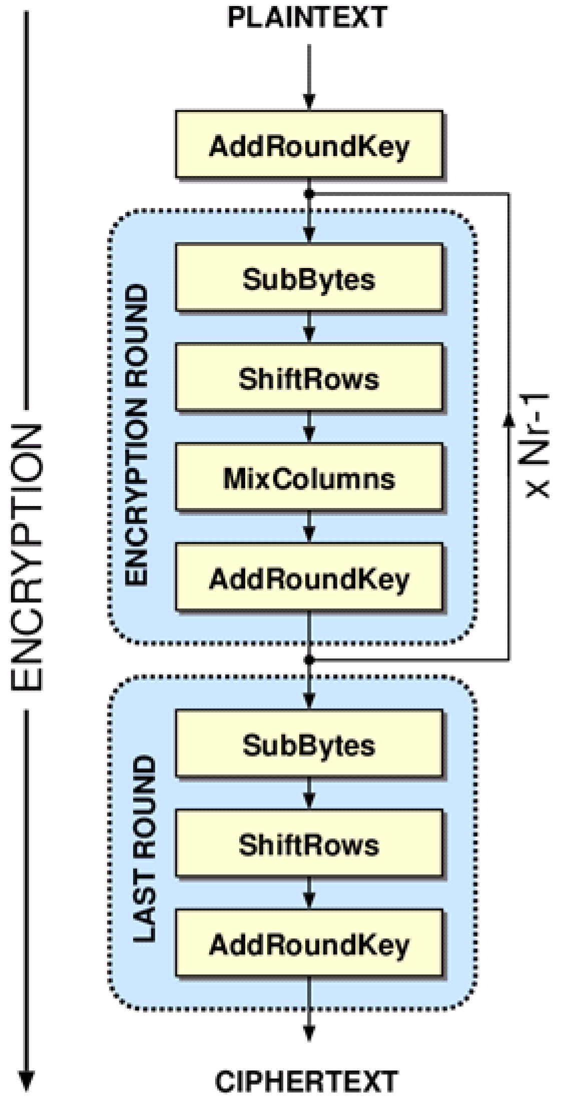
These slides are based on Lawrie Brown's slides supplied with William Stalling's book "Cryptography and Network Security: Principles and Practice," 5th Ed, 2011.

Advanced Encryption Standard (AES)

- Published by NIST in Nov 2001: FIPS PUB 197
- Based on a competition won by Rijmen and Daemen (Rijndael) from Belgium
- 22 submissions, 7 did not satisfy all requirements
  15 submissions 5 finalists: Mars, RC6, Rijndael, Serpent, Twofish. Winner: Rijndael.
- Rijndael allows many block sizes and key sizes
- AES restricts it to:
  - Block Size: 128 bits
  - Key sizes: 128, 192, 256 (AES-128, AES-192, AES-256)
- An iterative rather than Feistel cipher
  - operates on entire data block in every round
- Byte operations: Easy to implement in software

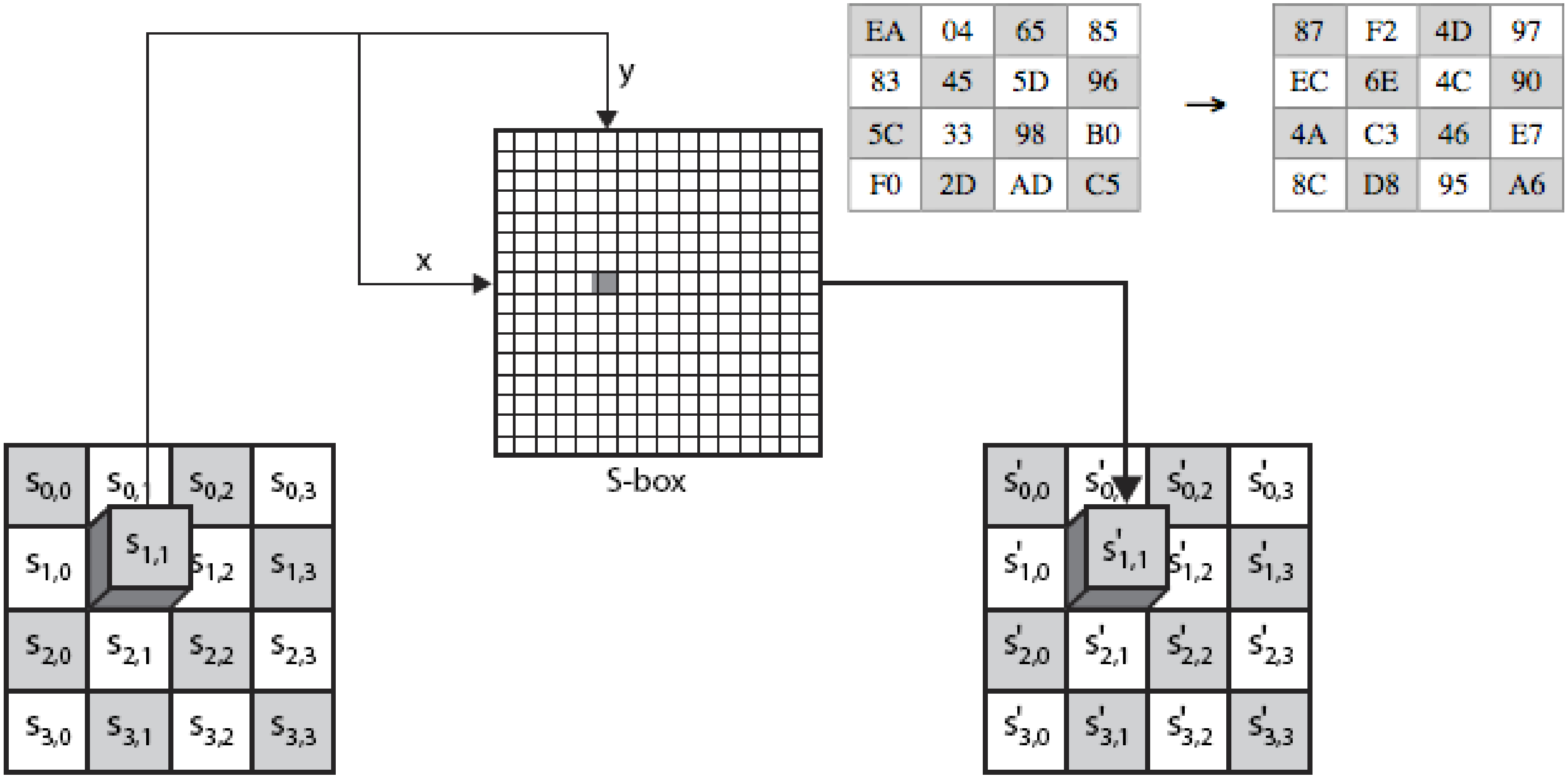
Basic Structure of AES

- # Rounds $N_r = 6 + \max\{N_b, N_k\}$
- N_b = 32-bit words in the block
- N_k = 32-bit words in key
- AES-128: 10
- AES-192: 12
- AES-256: 14



1. Substitute Bytes

- Each byte is replaced by byte indexed by row (left 4-bits) & column (right 4-bits) of a 16x16 table



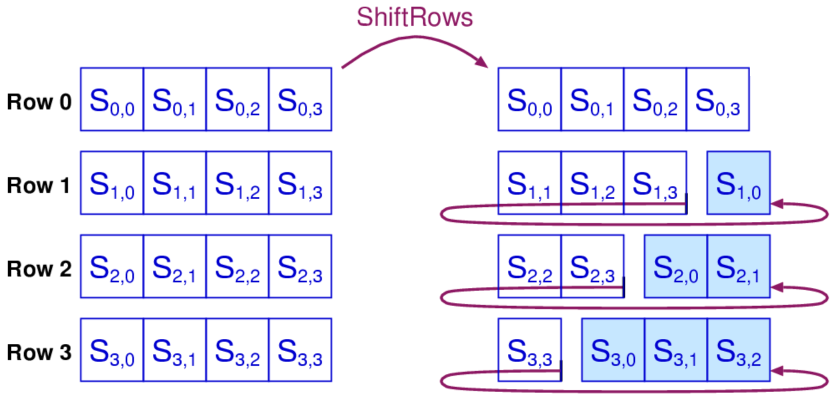
EA	04	65	85
83	45	5D	96
5C	33	98	B0
F0	2D	AD	C5

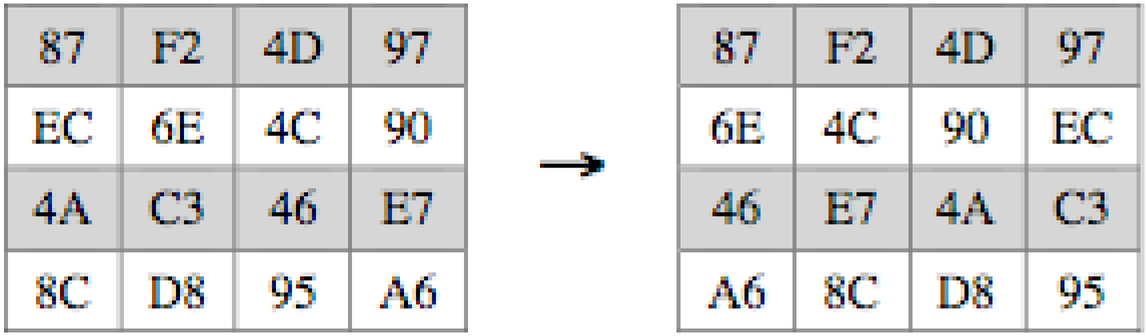
→

87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

2. Shift Rows

- 1st row is unchanged
- 2nd row does 1 byte circular shift to left
- 3rd row does 2 byte circular shift to left
- 4th row does 3 byte circular shift to left


ShiftRows
Row 0 $S_{0,0}$ $S_{0,1}$ $S_{0,2}$ $S_{0,3}$
Row 1 $S_{1,0}$ $S_{1,1}$ $S_{1,2}$ $S_{1,3}$
Row 2 $S_{2,0}$ $S_{2,1}$ $S_{2,2}$ $S_{2,3}$
Row 3 $S_{3,0}$ $S_{3,1}$ $S_{3,2}$ $S_{3,3}$
$S_{0,0}$ $S_{0,1}$ $S_{0,2}$ $S_{0,3}$
$S_{1,1}$ $S_{1,2}$ $S_{1,3}$ $S_{1,0}$
$S_{2,2}$ $S_{2,3}$ $S_{2,0}$ $S_{2,1}$
$S_{3,3}$ $S_{3,0}$ $S_{3,1}$ $S_{3,2}$



87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

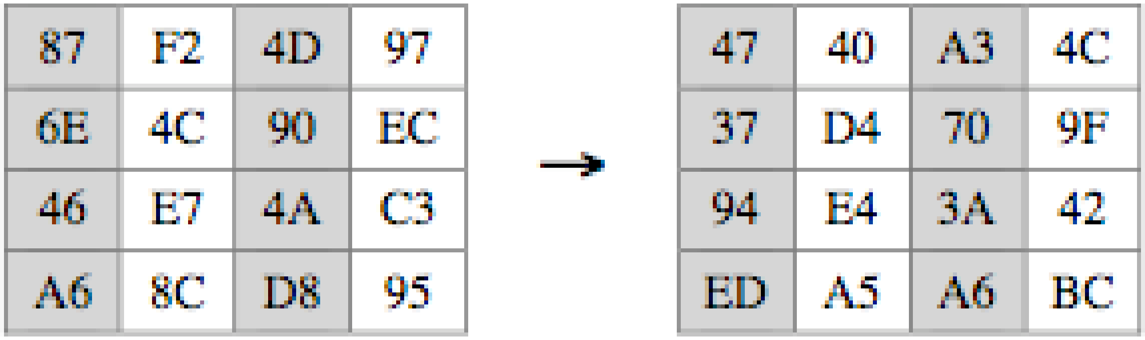
→

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

3. Mix Columns

- Effectively a matrix multiplication in GF(2^8) using prime polynomial $m(x) = x^8+x^4+x^3+x+1$

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



87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

→

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

$$(\{02\} \bullet \{87\}) \oplus (\{03\} \bullet \{6E\}) \oplus \{46\} \oplus \{A6\} = \{47\}$$

AES Arithmetic

- Uses arithmetic in the finite field GF(2^8) with irreducible polynomial

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

  which is `(100011011)` or `{11B}`

- Example:

  $\{02\} \bullet \{87\} \bmod \{11B\}$

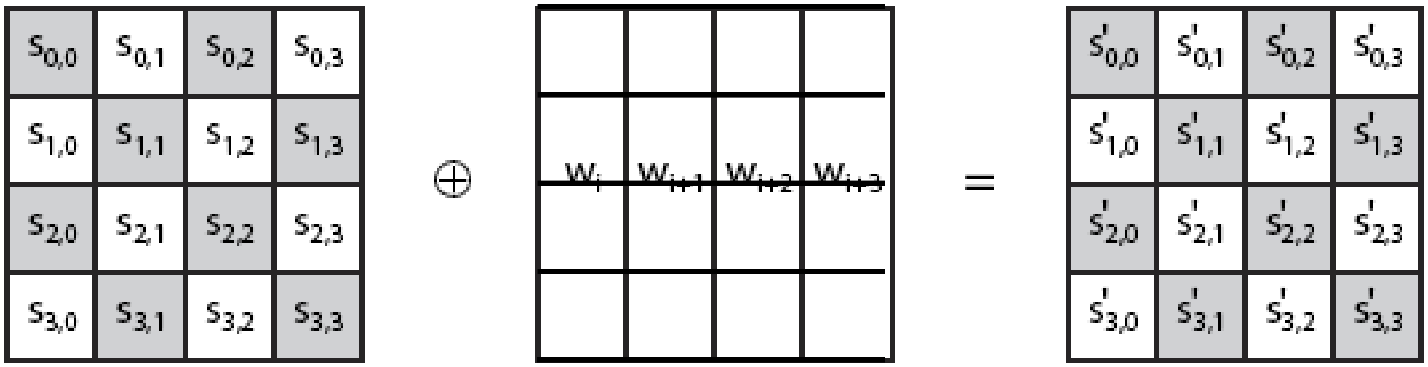
  $= (1\ 0000\ 1110) \bmod \{11B\}$

  $= (1\ 0000\ 1110) \oplus (1\ 0001\ 1011)$

  $= (0001\ 0101)$

4. Add Round Key

- XOR state with 128-bits of the round key



$$\begin{array}{|c|c|c|c|}\hline s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ \hline s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ \hline s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ \hline s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|}\hline & & & \\ \hline w_i & w_{i+1} & w_{i+2} & w_{i+3} \\ \hline & & & \\ \hline \end{array} = \begin{array}{|c|c|c|c|}\hline s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ \hline s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ \hline s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ \hline s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \\ \hline \end{array}$$

AES Key Expansion

- Use four byte words called w_i. Subkey = 4 words.

For AES-128:

- First subkey (w3,w2,w1,w0) = cipher key
- Other words are calculated as follows:

$$w_i = w_{i-1} \oplus w_{i-4}$$

for all values of i that are not multiples of 4.

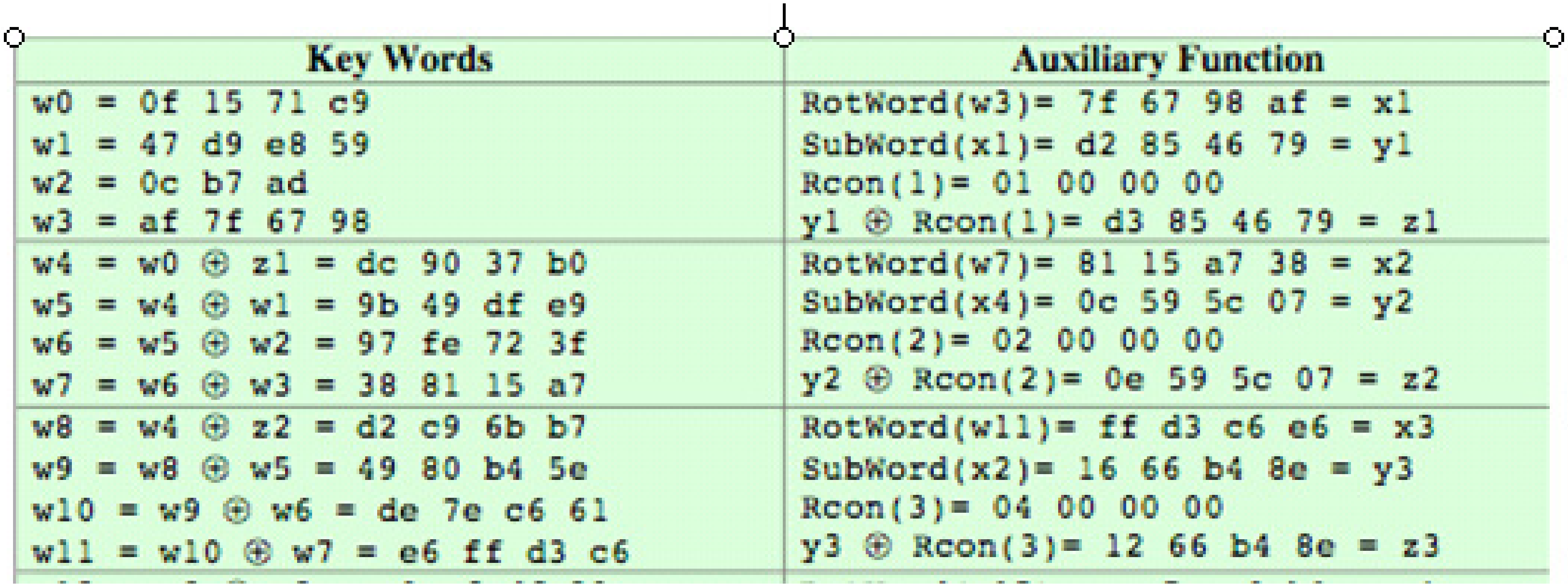
- For the words with indices that are a multiple of 4 (w_{4k}):

1. RotWord: Bytes of w_{4k-1} are rotated left shift (nonlinearity)
2. *SubWord: SubBytes* fn is applied to all four bytes. (Diffusion)
3. The result r_{sk} is XOR'ed with w_{4k-4} and a round constant r_{conk} (breaks Symmetry):

$$w_{4k} = r_{sk} \oplus w_{4k-4} \oplus r_{conk}$$

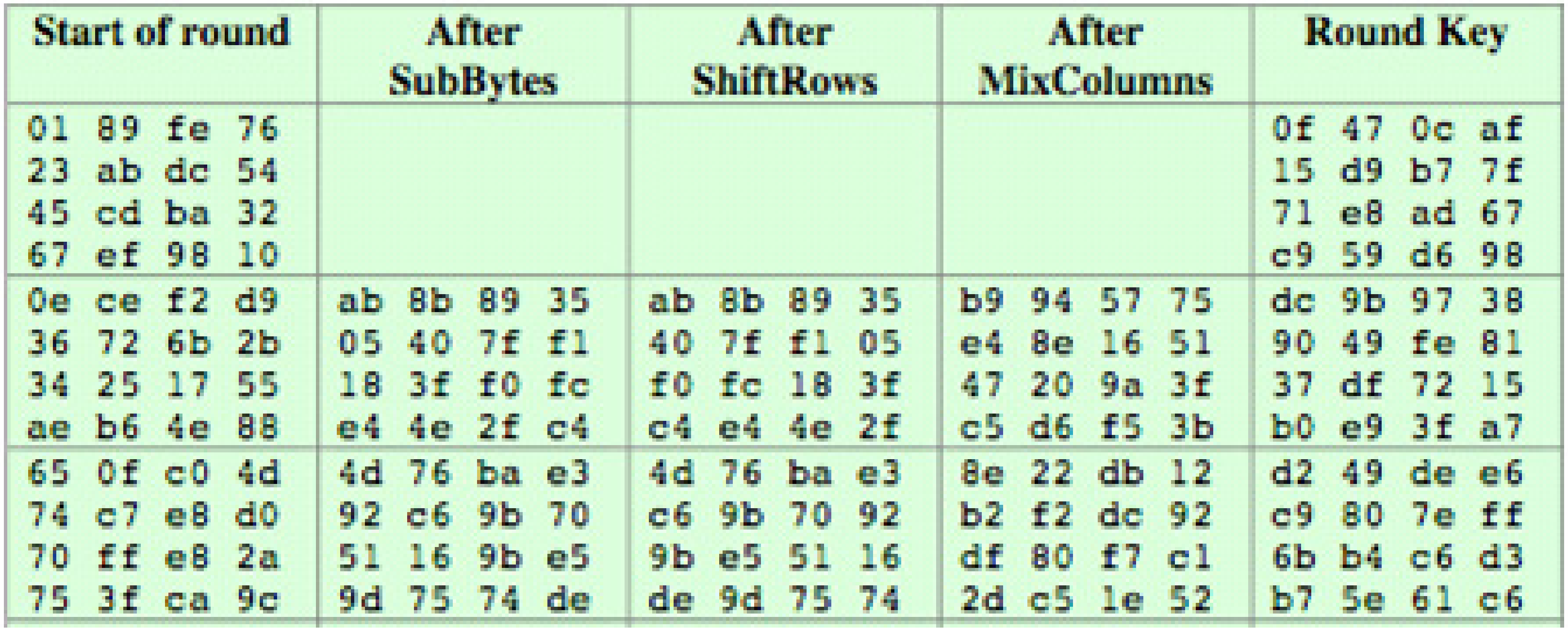
- For AES-192 and AES-256, the key expansion is more complex.

AES Example Key Expansion



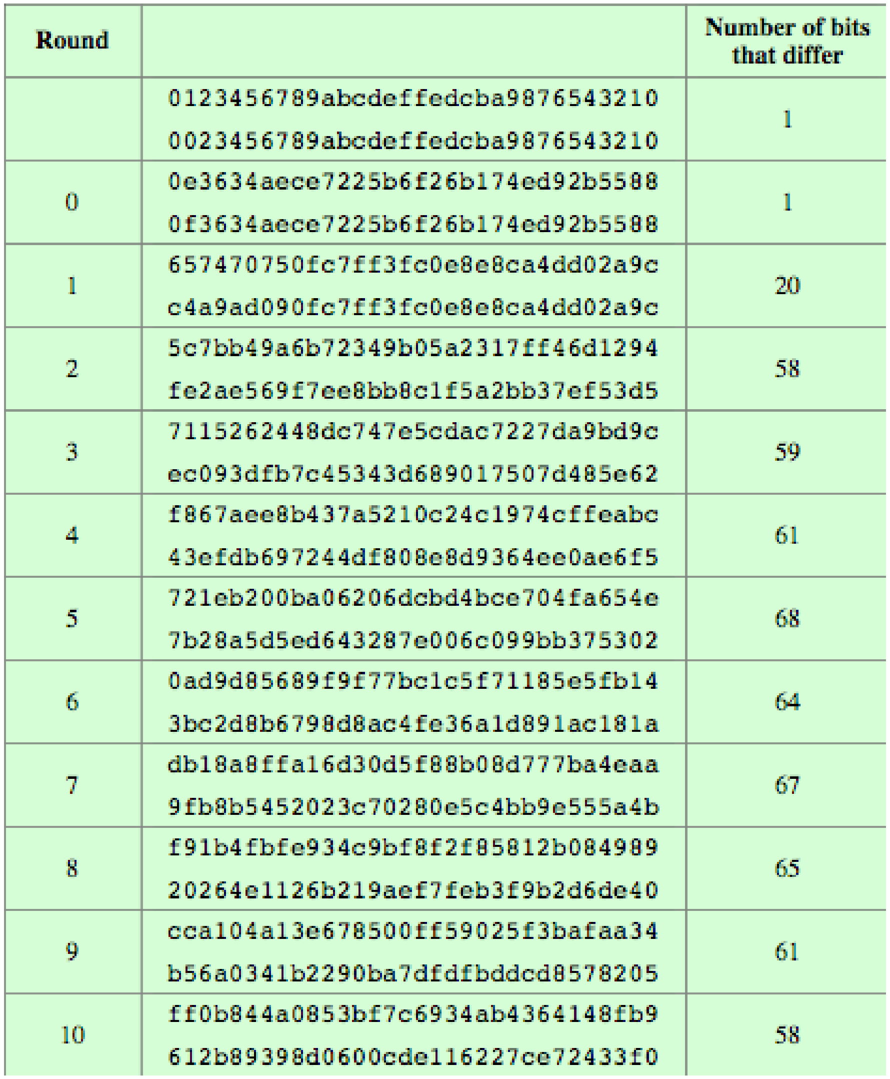
Key Words	Auxiliary Function
w0 = 0f 15 71 c9	RotWord(w3)= 7f 67 98 af = x1
w1 = 47 d9 e8 59	SubWord(x1)= d2 85 46 79 = y1
w2 = 0c b7 ad	Rcon(1)= 01 00 00 00
w3 = af 7f 67 98	y1 ⊕ Rcon(1)= d3 85 46 79 = z1
w4 = w0 ⊕ z1 = dc 90 37 b0	RotWord(w7)= 81 15 a7 38 = x2
w5 = w4 ⊕ w1 = 9b 49 df e9	SubWord(x4)= 0c 59 5c 07 = y2
w6 = w5 ⊕ w2 = 97 fe 72 3f	Rcon(2)= 02 00 00 00
w7 = w6 ⊕ w3 = 38 81 15 a7	y2 ⊕ Rcon(2)= 0e 59 5c 07 = z2
w8 = w4 ⊕ z2 = d2 c9 6b b7	RotWord(w11)= ff d3 c6 e6 = x3
w9 = w8 ⊕ w5 = 49 80 b4 5e	SubWord(x2)= 16 66 b4 8e = y3
w10 = w9 ⊕ w6 = de 7e c6 61	Rcon(3)= 04 00 00 00
w11 = w10 ⊕ w7 = e6 ff d3 c6	y3 ⊕ Rcon(3)= 12 66 b4 8e = z3

AES Example Encryption



Start of round	After SubBytes	After ShiftRows	After MixColumns	Round Key
01 89 fe 76 23 ab dc 54 45 cd ba 32 67 ef 98 10				0f 47 0c af 15 d9 b7 7f 71 e8 ad 67 c9 59 d6 98
0e ce f2 d9 36 72 6b 2b 34 25 17 55 ae b6 4e 88	ab 8b 89 35 05 40 7f f1 18 3f f0 fc e4 4e 2f c4	ab 8b 89 35 40 7f f1 05 f0 fc 18 3f c4 e4 4e 2f	b9 94 57 75 e4 8e 16 51 47 20 9a 3f c5 d6 f5 3b	dc 9b 97 38 90 49 fe 81 37 df 72 15 b0 e9 3f a7
65 0f c0 4d 74 c7 e8 d0 70 ff e8 2a 75 3f ca 9c	4d 76 ba e3 92 c6 9b 70 51 16 9b e5 9d 75 74 de	4d 76 ba e3 c6 9b 70 92 9b e5 51 16 de 9d 75 74	8e 22 db 12 b2 f2 dc 92 df 80 f7 c1 2d c5 1e 52	d2 49 de e6 c9 80 7e ff 6b b4 c6 d3 b7 5e 61 c6

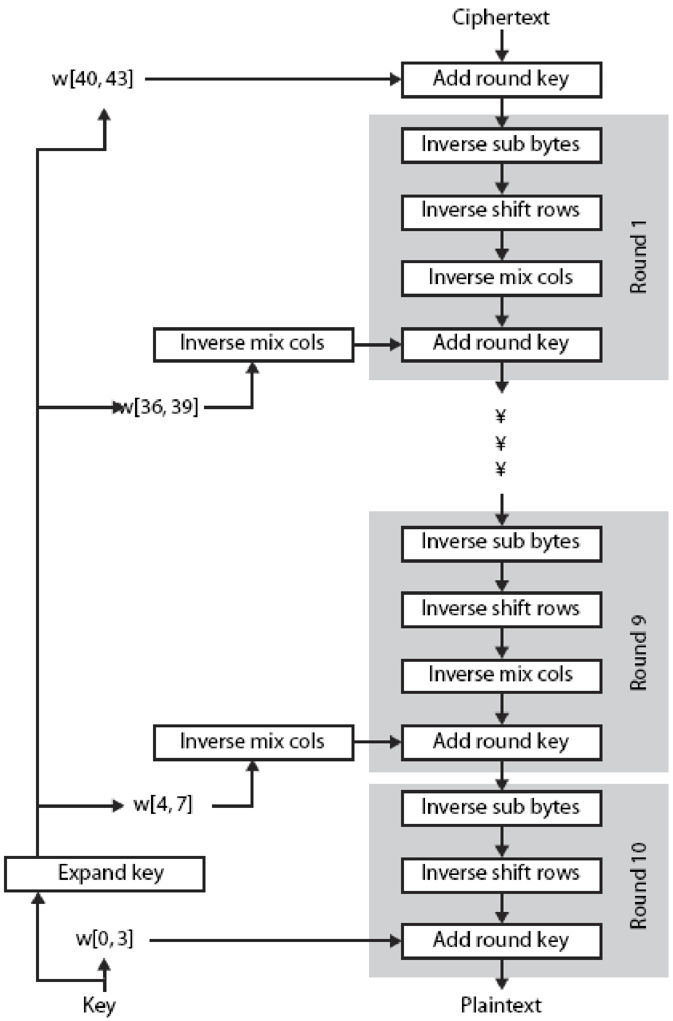
AES Example Avalanche



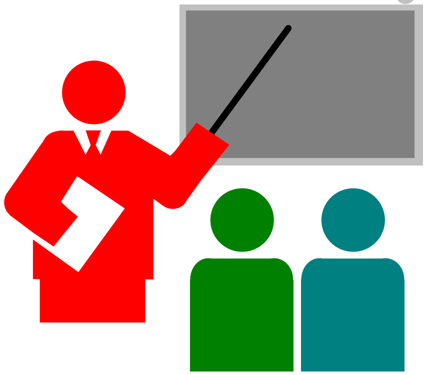
Round		Number of bits that differ
	0123456789abcdeffedcba9876543210 0023456789abcdeffedcba9876543210	1
0	0e3634aece7225b6f26b174ed92b5588 0f3634aece7225b6f26b174ed92b5588	1
1	657470750fc7ff3fc0e8e8ca4dd02a9c c4a9ad090fc7ff3fc0e8e8ca4dd02a9c	20
2	5c7bb49a6b72349b05a2317ff46d1294 fe2ae569f7ee8bb8c1f5a2bb37ef53d5	58
3	7115262448dc747e5cdac7227da9bd9c ec093dfb7c45343d689017507d485e62	59
4	f867aee8b437a5210c24c1974cffeabc 43efdb697244df808e8d9364ee0ae6f5	61
5	721eb200ba06206dcbd4bce704fa654e 7b28a5d5ed643287e006c099bb375302	68
6	0ad9d85689f9f77bc1c5f71185e5fb14 3bc2d8b6798d8ac4fe36a1d891ac181a	64
7	db18a8ffa16d30d5f88b08d777ba4eaa 9fb8b5452023c70280e5c4bb9e555a4b	67
8	f91b4fbfe934c9bf8f2f85812b084989 20264e1126b219aef7feb3f9b2d6de40	65
9	cca104a13e678500ff59025f3bafaa34 b56a0341b2290ba7dfdfbddcd8578205	61
10	ff0b844a0853bf7c6934ab4364148fb9 612b89398d0600cde116227ce72433f0	58

AES Decryption

- AES decryption is not identical to encryption
- But each step has an inverse



Summary



1. AES encrypts 128 bit blocks with 128-bit, 192-bit or 256-bit keys using 10, 12, or 14 rounds, respectively.
2. Is not a Feistel cipher ⇒All 128 bits are encrypted
3. Each round = 4 steps of SubBytes, ShiftRows, MixColumns, and AddRoundKey.
4. Last round has only 3 steps. No MixColumns.
5. Decryption is not the same as encryption (as in DES). Decryption consists of inverse steps.

Homework 5

5.4 Given the plaintext [0001 0203 0405 0607 0809 0A0B 0C0D 0E0F] and the key [0101 0101 0101 0101 0101 0101 0101 0101]

a. Show the original contents of state, displayed as a 4x4 matrix.

b. Show the value of state after initial AddRoundKey.

c. Show the value of State after SubBytes.

d. Show the value of State after ShiftRows.

e. Show the value of State after MixColumns.

Lab Homework 4

- This homework requires two computers with OpenSSH and telnet client and servers installed. You can use CSE571XPC2 client and CSE571XPS server or your own computers.
- Start wireshark on the client machine.
- telnet to the server and login with your username and password. Logout.
- Use "follow the TCP stream option" (right click on the packet) to see your username and password on the screen. Capture the screen and circle your password.
- ssh to the server and login with your username and password. Logout.
- Stop wireshark and read the trace. Capture the screen. Circle the password characters.
  Note the difference in the two logins?