UNIVERSITY OF MASSACHUSETTS
Dept. of Electrical & Computer Engineering

Introduction to Cryptography
ECE 597XX/697XX

Part 4

The Advanced Encryption Standard (AES)

Israel Koren

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Content of this part

- Overview of the AES algorithm
- Galois Fields
- Internal structure of AES
  - Byte Substitution layer
  - Diffusion layer
  - Key Addition layer
  - Key schedule
- Decryption
- Practical issues

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

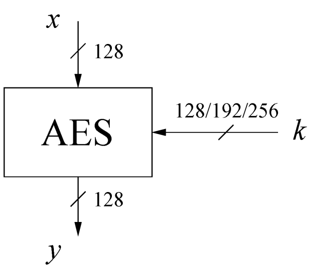
Some Basic Facts

- AES is the most widely used symmetric cipher today
- The algorithm for AES was chosen by the US *National Institute of Standards and Technology* (NIST) in a multi-year selection process
- The requirements for all AES candidate submissions were:
  - Block cipher with **128-bit block size**
  - **Three supported key lengths**: 128, 192 and 256 bit
  - Security relative to other submitted algorithms
  - **Efficiency** in software and hardware

Chronology of the AES Selection

- **The need for a new block cipher announced by NIST in January, 1997**
- **15 candidates algorithms accepted in August, 1998**
- **5 finalists announced in August, 1999:**
  - ***Mars*** **– IBM Corporation**
  - ***RC6*** **–** ***RSA*** **Laboratories**
  - ***Rijndael*** **– J. Daemen & V. Rijmen**
  - ***Serpent*** **– E. Biham et al.**
  - ***Twofish*** **– B. Schneier et al.**
- **In October 2000,** ***Rijndael*** **was chosen as the AES**
- **AES was formally approved as a US federal standard in November 2001**

AES: Overview



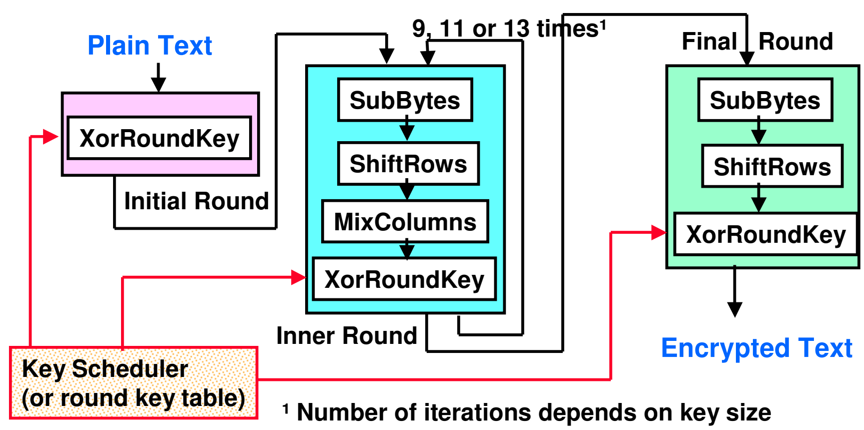
- The number of rounds depends on the chosen key length:

Key length (bits)	Number of rounds
128	10
192	12
256	14

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

AES: Overview

- Iterated cipher with 10/12/14 rounds for key length of 128, 192, 256 bits, respectively (Nr+1 round keys)
- Each round consists of "Layers"



[1] Number of iterations depends on key size

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Internal Structure of AES

- AES is a byte-oriented cipher
- The state A (i.e., the 128-bit data path) can be arranged in a 4x4 matrix:

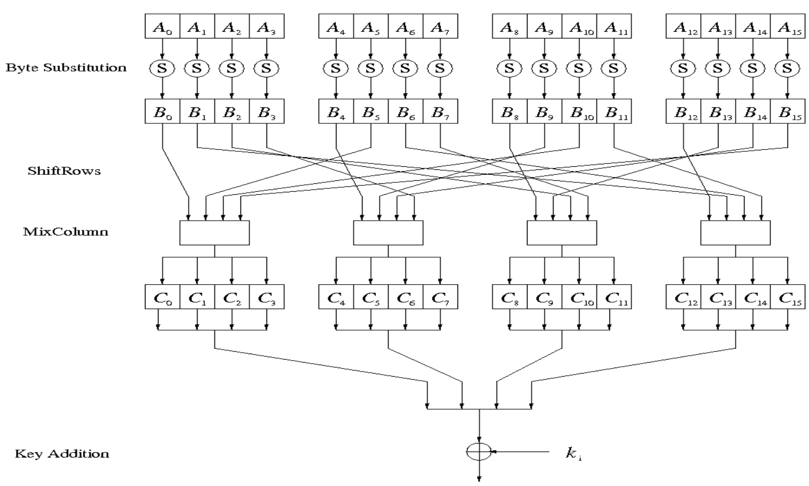
A_0	A_4	A_8	A_{12}
A_1	A_5	A_9	A_{13}
A_2	A_6	A_{10}	A_{14}
A_3	A_7	A_{11}	A_{15}

S_{00}	S_{01}	S_{02}	S_{03}
S_{10}	S_{11}	S_{12}	S_{13}
S_{20}	S_{21}	S_{22}	S_{23}
S_{30}	S_{31}	S_{32}	S_{33}

with $A_0, \ldots, A_{15}$ denoting the 16-byte input of AES

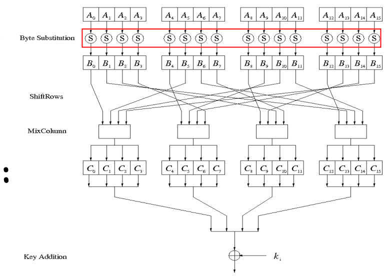
Internal Structure of AES

- Round function for rounds $1,2,\ldots,N_r-1$:



- Note: In the last round, the MixColumn tansformation is omitted

Byte Substitution Layer



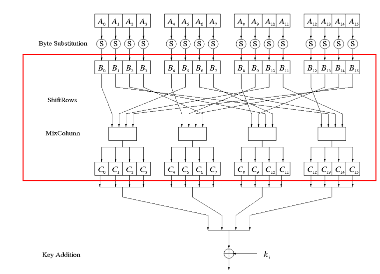
- The Byte Substitution layer consists of 16 S-Boxes with the following properties:

  The S-Boxes are

  - identical
  - the only nonlinear elements of AES, i.e., ByteSub(A_i) + ByteSub(A_j) ≠ ByteSub(A_i + A_j), for i,j = 0,…,15
  - bijective, i.e., there exists a one-to-one mapping of input and output bytes
    ⇒ S-Box can be uniquely reversed

- In software implementations, the S-Box is usually realized as a lookup table

Diffusion Layer



The Diffusion layer

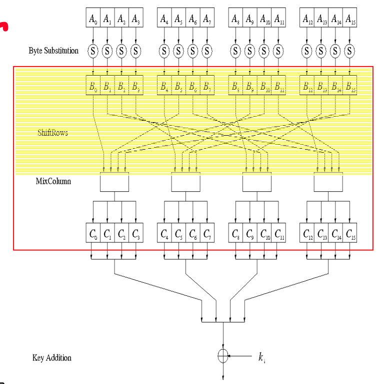
- provides diffusion over all input state bits

- consists of two sublayers:
  - ShiftRows Sublayer: Permutation of the data on a byte level
  - MixColumn Sublayer: Matrix operation which combines ("mixes") blocks of four bytes

- performs a linear operation on state matrices A, B, i.e.,

$$\mathrm{DIFF}(A) + \mathrm{DIFF}(B) = \mathrm{DIFF}(A + B)$$

ShiftRows Sublayer



- Rows of the state matrix are shifted cyclically:

Input matrix

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}

Output matrix

B_0	B_4	B_8	B_{12}	no shift
B_5	B_9	B_{13}	B_1	← one position left rotate
B_{10}	B_{14}	B_2	B_6	← two positions left rotate
B_{15}	B_3	B_7	B_{11}	← three positions left rotate

Introduction to Galois Fields

- Substitution & Mix-column steps based on Galois field arithmetic
- A Galois field consists of a finite set of elements with the operation: add, subtruct, multiply and invert
- A group is a set of elements with one operation that is closed and associative, the set has a neutral (identity) element „1" and each element a has an inverse so that $a \circ a^{-1} = 1$ and $a \circ 1 = a$
- A group is commutative if the operation is commutative.
- The set {0,1,...,m-1} with the addition mod m is a group but

this set with the operation multiply mod m is not.

- A field is a set of elements that form an additive group with the operation + and a multiplicative group (except 0) with the operation ×, and the ditributivity rule holds.
- A finite field of order *m* has $m = p^n$ elements with *p* a prime number.
- For n=1 the field GF(p) consists of the integers 0,1,...,p-1 and add/multiply mod p – all non-zero elements have an inverse- special case of a ring.

Galois Fields

- GF(5): multiplicative inverses for 0,1,2,3,4 are: none,1,3,2,4
- GF(2) and its extension field *GF*(2^8) are important for AES
  - AES operates on bytes that have 256 possible values
  - But 2^8 is not a prime and we cannot use add/multiply mod 2^8 (why)
- Define the extension field *GF*(2^8) as consisting of 256 polynomials

$$A(x) = a_7x^7 + \cdots + a_1x + a_0, \quad a_i \in GF(2) = \{0,1\}$$

$$A = (a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0)$$

- *GF*(2^m): $$C(x) = A(x) + B(x) = \sum_{i=0}^{m-1} c_i x^i, \quad c_i \equiv a_i + b_i \bmod 2$$

$$C(x) = A(x) - B(x) = \sum_{i=0}^{m-1} c_i x^i, \quad c_i \equiv a_i - b_i \equiv a_i + b_i \bmod 2.$$

- Example: A=(1101 0011) + B=(0101 1010) in binary and polynomial notation

Extension Field

- Multiplication would generate a polynomial of degree 2m – we divide the product by a given polynomial and use the remainder
  - The modulo reduction should use an irreducible polynomial
- Define multiplication in *GF*(2^m) as: $C(x) \equiv A(x) \cdot B(x) \bmod P(x).$
  where *P(x)* is an irreducible polynomial
- For AES $P(x) = x^8 + x^4 + x^3 + x + 1$
- Example A=(0010 0010), B=(0001 0101) ⇒ C(1011 1100)

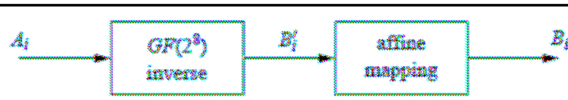
Inversion

- Basic operation (but not the only one) behind the substitution step in AES
- Definition: $A^{-1}(x) \cdot A(x) = 1 \bmod P(x)$
- For small fields commonly done using lookup tables
- Example: Inverse of A=(0010 0010)=(22)$_{16}$ is (5A)

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

X \ Y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00	01	8D	F6	CB	52	7B	D1	E8	4F	29	C0	B0	E1	E5	C7
1	74	B4	AA	4B	99	2B	60	5F	58	3F	FD	CC	FF	40	EE	B2
2	3A	6E	5A	F1	55	4D	A8	C9	C1	0A	98	15	30	44	A2	C2
3	2C	45	92	6C	F3	39	66	42	F2	35	20	6F	77	BB	59	19
4	1D	FE	37	67	2D	31	F5	69	A7	64	AB	13	54	25	E9	09
5	ED	5C	05	CA	4C	24	87	BF	18	3E	22	F0	51	EC	61	17
6	16	5E	AF	D3	49	A6	36	43	F4	47	91	DF	33	93	21	3B
7	79	B7	97	85	10	B5	BA	3C	B6	70	D0	06	A1	FA	81	82
8	83	7E	7F	80	96	73	BE	56	9B	9E	95	D9	F7	02	B9	A4
9	DE	6A	32	6D	D8	8A	84	72	2A	14	9F	88	F9	DC	89	9A
A	FB	7C	2E	C3	8F	B8	65	48	26	C8	12	4A	CE	E7	D2	62
B	0C	E0	1F	EF	11	75	78	71	A5	8E	76	3D	BD	BC	86	57
C	0B	28	2F	A3	DA	D4	E4	0F	A9	27	53	04	1B	FC	AC	E6
D	7A	07	AE	63	C5	DB	E2	EA	94	8B	C4	D5	9D	F8	90	6B
E	B1	0D	D6	EB	C6	0E	CF	AD	08	4E	D7	E3	5D	50	1E	B3
F	5B	23	38	34	68	46	03	8C	DD	9C	7D	A0	CD	1A	41	1C

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources



Substitution step

- Inverse: $B'_i = A_i^{-1}$

$B' = (b'_7, \ldots, b'_0)$

- Multiply by a constant matrix and add a constant

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix} \equiv \begin{pmatrix} 1&0&0&0&1&1&1&1 \\ 1&1&0&0&0&1&1&1 \\ 1&1&1&0&0&0&1&1 \\ 1&1&1&1&0&0&0&1 \\ 1&1&1&1&1&0&0&0 \\ 0&1&1&1&1&1&0&0 \\ 0&0&1&1&1&1&1&0 \\ 0&0&0&1&1&1&1&1 \end{pmatrix} \begin{pmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \bmod 2$$

Typically replaced by a table lookup

x \ y	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

Example:

$A_i = (11000010)_2 = (C2)_{hex}$

From previous slide:

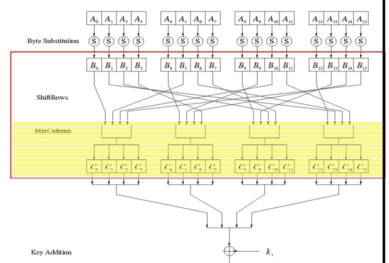
$A_i^{-1} = B'_i = (2F)_{hex} = (00101111)_2$

$B_i = (00100101) = (25)_{hex}$

Adapted fr

MixColumn Sub-layer

- ♦Linear transformation which mixes each column of the state matrix



- ♦Each 4-byte column is considered as a vector and multiplied by a fixed 4x4 matrix, e.g.,

$$\begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \cdot \begin{pmatrix} B_0 \\ B_5 \\ B_{10} \\ B_{15} \end{pmatrix}$$

where 01, 02 and 03 are given in hexadecimal notation

- ♦All arithmetic is done in the Galois field $GF(2^8)$

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

MixColumn Sublayer

- ♦ **MixColumns:** $\alpha=x=02_{16}$; $\beta=x+1=03_{16}$, ⊗ and ⊕ are mod 2 multiply and add; both modulo AES generator polynomial

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

$$s_{o,j} = (\alpha \otimes s_{o,j}) \oplus (\beta \otimes s_{1,j}) \oplus s_{2,j} \oplus s_{3,j}$$
$$s_{1,j} = s_{o,j} \oplus (\alpha \otimes s_{1,j}) \oplus (\beta \otimes s_{2,j}) \oplus s_{3,j}$$
$$s_{2,j} = s_{o,j} \oplus s_{1,j} \oplus (\alpha \otimes s_{2,j}) \oplus (\beta \otimes s_{3,j})$$
$$s_{3,j} = (\beta \otimes s_{o,j}) \oplus s_{1,j} \oplus s_{2,j} \oplus (\alpha \otimes s_{3,j})$$

Example: 03 ⊗ 5d = e7 ≡ $(x+1) \otimes (x^6 + x^4 + x^3 + x^2 + 1)$

Example: 02⊗bf=17e mod P(x)=65 ≡

$$f(x) = (x) \otimes (x^7 + x^5 + x^4 + x^3 + x^2 + x + 1)$$

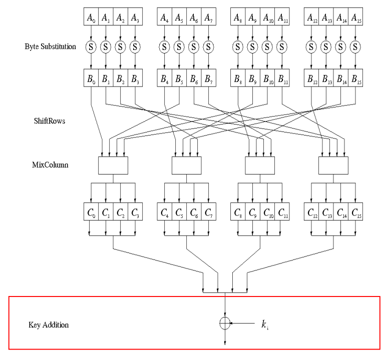
If f(x) is of degree 8:

$$f(x) \bmod (x^8 + x^4 + x^3 + x + 1) = f(x) \oplus P(x)$$

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Key Addition Layer

- **Inputs:**
  - 16-byte state matrix S
  - 16-byte subkey k_i
- **Output:** $S \oplus k_i$
- The subkeys are generated in the key schedule



S_{00}	S_{01}	S_{02}	S_{03}
S_{10}	S_{11}	S_{12}	S_{13}
S_{20}	S_{21}	S_{22}	S_{23}
S_{30}	S_{31}	S_{32}	S_{33}

$\oplus$

k_{00}	k_{01}	k_{02}	k_{03}
k_{10}	k_{11}	k_{12}	k_{13}
k_{20}	k_{21}	k_{22}	k_{23}
k_{30}	k_{31}	k_{32}	k_{33}

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Key Schedule

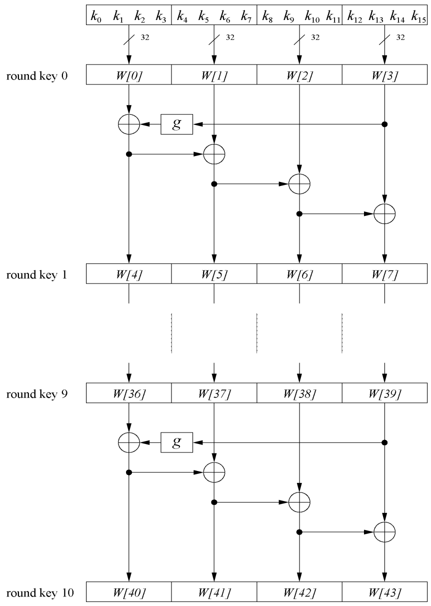
- Subkeys are derived recursively from the original 128/192/256-bit input key
- Each round has 1 subkey, plus 1 subkey at the beginning of AES

Key length (bits)	Number of subkeys
128	11
192	13
256	15

- Key whitening: Subkey is used both at the input and output of AES
  $\Rightarrow$ # subkeys = # rounds + 1
- There are different key schedules for the different key sizes

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Key Schedule



Example: Key schedule for 128-bit key AES

- Word-oriented: 1 word = 4 bytes = 32 bits
- 11 subkeys are stored in *W[0]…W[3], W[4]…W[7], … , W[40]…W[43]*
- First subkey *W[0]…W[3]* is the original AES key

$$W[4i] = W[4(i-1)] + g(W[4i-1])$$

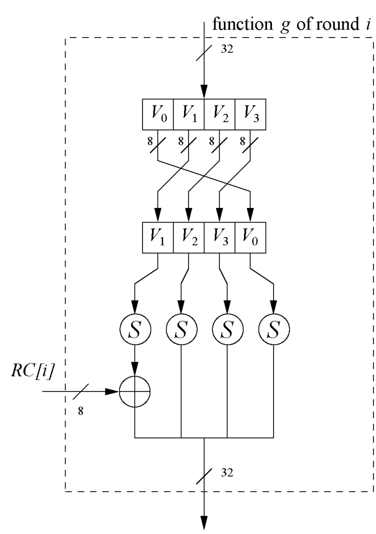
$$W[4i+j] = W[4i+j-1] + W[4(i-1)+j].$$

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Key Schedule

- ♦ **Function *g* rotates its four input bytes and performs a bytewise S-Box substitution ⇒ nonlinearity**
- ♦ **Round coefficient *RC* is only added to leftmost byte and varies from round to round:**

$RC[1] = x^0 = (00000001)_2$
$RC[2] = x^1 = (00000010)_2$
$RC[3] = x^2 = (00000100)_2$
...
$RC[10] = x^9 = (00110110)_2$



- ♦ ***x^i* represents an element in a Galois field *GF*(2^8)**

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

AES key Schedule

- **Nr=10,12,14 rounds**
- **Nk=4,6,8 words**

```
KeyExpansion(byte key[4 * Nk], word w[4 * (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 < 4 * (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
```


Example

Plaintext = 32 43 f6 a8 88 5a 30 8d 31 31 98 a2 e0 37 07 34

128-bit key = 2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c

$$\begin{bmatrix} 32 & 88 & 31 & e0 \\ 43 & 5a & 31 & 37 \\ f6 & 30 & 98 & 07 \\ a8 & 8d & a2 & 34 \end{bmatrix}$$

(a) Initial state matrix.

$$\begin{bmatrix} 2b & 28 & ab & 09 \\ 7e & ae & f7 & cf \\ 15 & d2 & 15 & 4f \\ 16 & a6 & 88 & 3c \end{bmatrix}$$

(b) Key added in round 1.

$$\begin{bmatrix} 19 & a0 & 9a & e9 \\ 3d & f4 & c6 & f8 \\ e3 & e2 & 8d & 48 \\ be & 2b & 2a & 08 \end{bmatrix}$$

(c) State matrix - end of round 1.

$$\begin{bmatrix} d4 & e0 & b8 & 1e \\ 27 & bf & b4 & 41 \\ 11 & 98 & 5d & 52 \\ ae & f1 & e5 & 30 \end{bmatrix}$$

(d) After SubBytes.

- **MixColumns:**

$$s_{0,0} = (\alpha \otimes s_{0,0}) \oplus (\beta \otimes s_{1,0}) \oplus s_{2,0} \oplus s_{3,0}$$
$$= (02 \otimes d4) \oplus (03 \otimes bf) \oplus 5d \oplus 30$$
$$= 1b8 \oplus 1c1 \oplus 5d \oplus 30 = 04$$

$$\begin{bmatrix} d4 & e0 & b8 & 1e \\ bf & b4 & 41 & 27 \\ 5d & 52 & 11 & 98 \\ 30 & ae & f1 & e5 \end{bmatrix}$$

(e) After ShiftRows.

$$\begin{bmatrix} 04 & e0 & 48 & 28 \\ 66 & cb & f8 & 06 \\ 81 & 19 & d3 & 26 \\ e5 & 9a & 7a & 4c \end{bmatrix}$$

(f) After MixColumns.

$$s_{1,0} = s_{0,0} \oplus (\alpha \otimes s_{1,0}) \oplus (\beta \otimes s_{2,0}) \oplus_{3,0}$$
$$= d4 \oplus (02 \otimes bf) \oplus (03 \otimes 5d) \oplus 30 = d4 \oplus 17e \oplus e7 \oplus 30 = 17d$$

$$17d \bmod \quad p(x) = 17d \oplus p(x) = 7d \oplus (x^4 + x^3 + x + 1) = 7d \oplus 1b = 66$$

Effect of bit flips

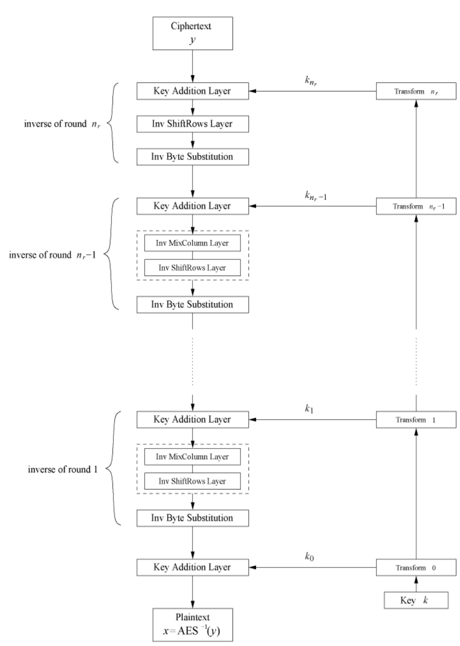
- Plaintext:
  32 43 f6 a8 88 5a 30 8d 31 31 98 a2 e0 37 07 34
- 128-bit key:
  2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c
- Ciphertext:
  39 25 84 1d 02 dc 09 fb dc 11 85 97 19 6a 0b 32

- A single bit flip in the plaintext:
  30 43 f6 a8 88 5a 30 8d 31 31 98 a2 e0 37 07 34
- Results in the ciphertext:
  c0 06 27 d1 8b d9 e1 19 d5 17 6d bc ba 73 37 c1
- A single bit flip in the key:
  2a 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c
- Results in the ciphertext:
  c4 61 97 9e e4 4d e9 7a ba 52 34 8b 39 9d 7f 84

- A single bit flip results in a totally scrambled output

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Decryption



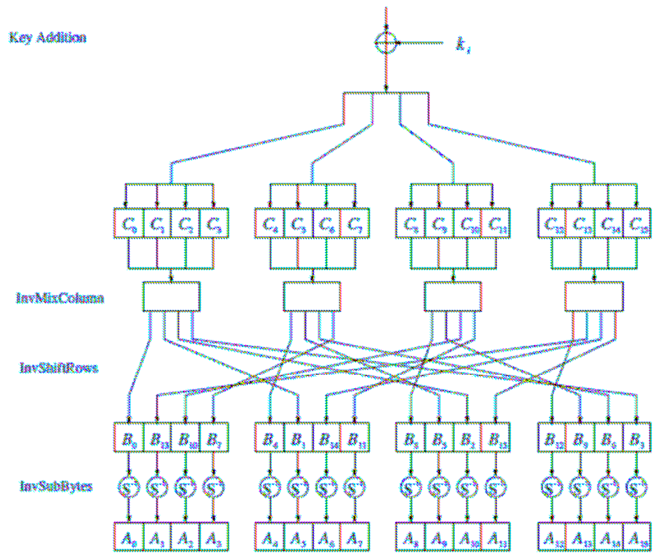
- AES is not based on a Feistel network

⇒ All layers must be inverted for decryption:

- **MixColumn layer** → Inv MixColumn layer
- **ShiftRows layer**→ Inv ShiftRows layer
- **Byte Substitution layer** → Inv Byte Substitution layer
- **Key Addition layer is its own inverse**

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Decryption- details



ECE597/697 Koren Part.4 .27

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Decryption – Inv Mixcolumn

- **Inv MixColumn layer:**
  - **To reverse the MixColumn operation, each column of the state matrix *C* must be multiplied with the** inverse of the 4x4 matrix, **e.g.,**

$$\begin{pmatrix} B_0 \\ B_1 \\ B_2 \\ B_3 \end{pmatrix} = \begin{pmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{pmatrix} \cdot \begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix} \qquad \begin{pmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{pmatrix} \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix}$$

where 09, 0*B*, *0D* and 0*E* are given in hexadecimal notation

- **All arithmetic done in the Galois field *GF*(2^8)**

ECE597/697 Koren Part.4 .28

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Decryption – Inv Shift Rows

- **Inv ShiftRows layer:**
  - All rows of the state matrix B are shifted to the opposite direction:

Input matrix

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}

Output matrix

B_0	B_4	B_8	B_{12}	no shift
B_{13}	B_1	B_5	B_9	→ one position right rotate
B_{10}	B_{14}	B_2	B_6	→ two positions right rotate
B_7	B_{11}	B_{15}	B_3	→ three positions right rotate

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Decryption – Inv S-Box

- **Inv Byte Substitution layer:**
  - Since the S-Box is bijective, it is possible to construct an inverse, such that

$$A_i = S^{-1}(B_i) = S^{-1}(S(A_i))$$

⇒ The inverse S-Box is used for decryption. It is usually realized as a lookup table

- **Decryption key schedule:**
  - Subkeys are needed in reversed order (compared to encryption)
  - In practice, for encryption and decryption, the same key schedule is used. This requires that all subkeys must be computed before the encryption of the first block can begin

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Implementation in Software

- One requirement of AES was the possibility of an efficient software implementation
- Straightforward implementation is well suited for 8-bit processors (e.g., smart cards), but inefficient on 32-bit or 64-bit processors
- A more sophisticated approach: Merge all round functions (except the key addition) into one table look-up
  - This results in four tables with 256 entries, where each entry is 32 bits wide
  - One round can be computed with 16 table look-ups
- Typical SW speeds are more than 1.6 Gbit/s on modern 64-bit processors

Security

- Brute-force attack: Due to the key length of 128, 192 or 256 bits, a brute-force attack is not possible
- Analytical attacks: There is no efficient analytical attack known that is sufficiently better than brute-force (e.g., complexity of 2^{126})
- Side-channel attacks:
  - Many side-channel attacks have been published
  - Note that side-channel attacks do not attack the underlying algorithm but the implementation of it

Lessons Learned

- AES is a block cipher which supports three key lengths of 128, 192 and 256 bit. It provides excellent long-term security against brute-force attacks.
- AES has been studied intensively since the late 1990s and no attacks have been found that are better than brute-force.
- AES is not based on Feistel networks. Its basic operations use Galois field arithmetic and provide strong diffusion and confusion.
- AES is part of numerous open standards such as Ipsec (Internet Protocol Security) or TLS (Transport Layer Security), in addition to being the mandatory encryption algorithm for US government applications. It is likely to be the dominant encryption algorithm for many years to come.
- AES is efficient in software and hardware.

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources