Symmetric Cryptography - AES

Data and Information Management: ELEN 3015

School of Electrical and Information Engineering,
University of the Witwatersrand

Sources

Bruen and Forcinito, “Crypto,... ”, p.102 - 109

fips-197 (AES standard)

Overview

1. Motivation
2. Introduction to algorithm and concepts
3. Key schedule
4. Rounds
5. Decryption

1. Motivation for AES

DES has reached the end of its credibility

Several reasons $\rightarrow$ main one: key-space

Rijndael was selected to replace DES

DES is a Feistel Cipher

AES is a substitution-permutation network

2. Introduction - Advanced Encryption Standard

Also known as Rijndael

Developed by Vincent Rijmen and Joan Daemen

Rijndael $\rightarrow$ Key and blocklengths can be 128, 192 and 256 bits

AES $\rightarrow$ Key - 128, 192 or 256 bits, blocklength - 128 bits

2. Introduction - Advanced Encryption Standard

Consider only 128-bit AES

Input is 16 bytes (128 bits), output 16 bytes (128 bits)

10 Rounds $\rightarrow$ 9 full rounds + final round

Every round has own subkey

2. Introduction - Advanced Encryption Standard

Rijndael uses $GF(2^8)$

Every byte is mapped onto a symbol in $GF(2^8)$

Irreducible polynomial for $GF(2^8) \rightarrow m(x) = x^8 + x^4 + x^3 + x + 1$ (not primitive)

Because of symbol size, operations can be performed on entire bytes at a time

2. Introduction - Advanced Encryption Standard

Representing the input data

Input consists of 128 bits (16 bytes)
$A_{0,0}, A_{1,0}, A_{2,0}, A_{3,0}, A_{0,1}, \cdots, A_{0,3}, A_{1,3}, A_{2,3}, A_{3,3}$
Bytes are arranged in a 4×4 matrix:

$$A = \begin{pmatrix} A_{0,0} & A_{0,1} & A_{0,2} & A_{0,3} \\ A_{1,0} & A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,0} & A_{2,1} & A_{2,2} & A_{2,3} \\ A_{3,0} & A_{3,1} & A_{3,2} & A_{3,3} \end{pmatrix}$$

A also called the State (Intermediate Cipher result that can be pictured as a rectangular array of bytes)

2. Introduction - Advanced Encryption Standard

Representing the input data

Byte values presented as the concatenation of its individual bit values between braces in the order $\{b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0\}$.

Bytes also interpreted as finite field elements using a polynomial representation:
$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0 = \sum_{i=0}^{7} b_i x^i$.

Example:

$\{01100011\} \rightarrow$

2. Introduction - Advanced Encryption Standard

Representing the input data

Byte values presented as the concatenation of its individual bit values between braces in the order $\{b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0\}$.

Bytes also interpreted as finite field elements using a polynomial representation:
$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0 = \sum_{i=0}^{7} b_i x^i$.

Example:

$\{01100011\} \rightarrow x^6 + x^5 + x + 1 \rightarrow 63_{16}$

2. Introduction - Advanced Encryption Standard

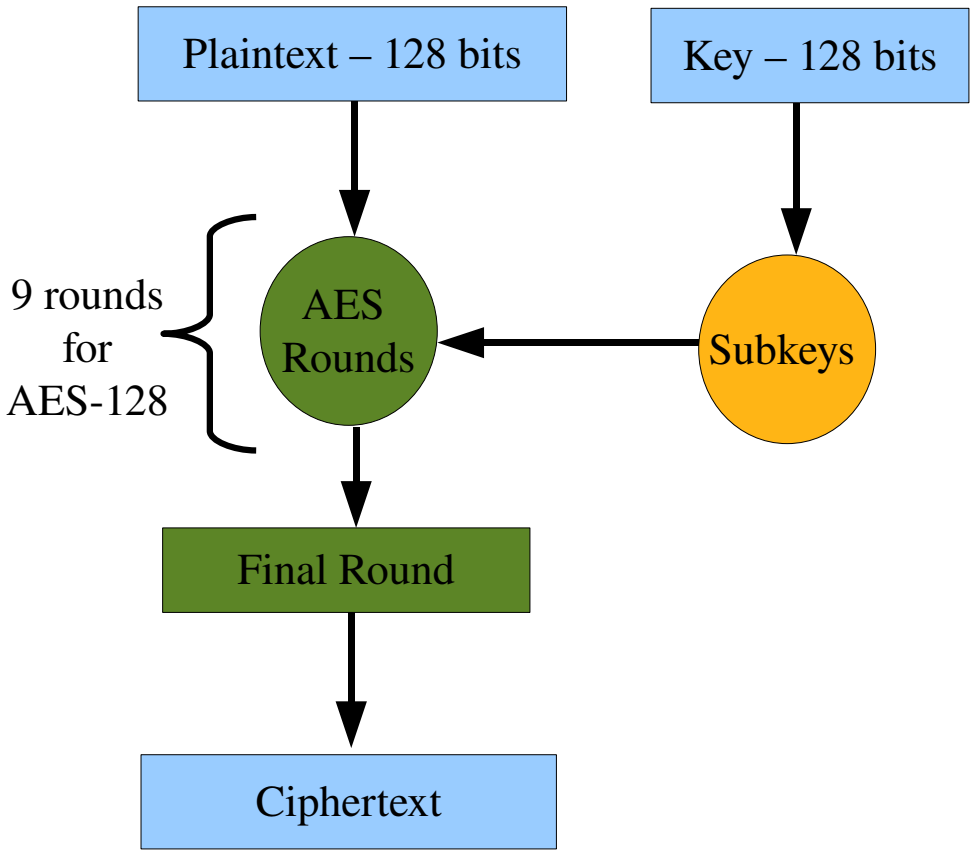
Multiplication of two elements

Multiplication performed modulo $m(x)$

$a(x) \cdot b(x) \mod m(x) = p(x)$

Inverses calculated using Extended Euclidean algorithm

2. Introduction - Advanced Encryption Standard



3. Key Schedule

Key 128-bits = 16 bytes

W-matrix: Matrix with dimensions (4×44)

Bytes are placed in columns w_0, w_1, w_2 and w_3

Creating rest of the columns:

$$w_j = \begin{cases} w_{j-4} + w_{j-1}, & \text{if } 4 \nmid j \\ w_{j-4} + n_{j-1} + v_j, & \text{if } 4 \mid j \end{cases}$$

3. Key Schedule

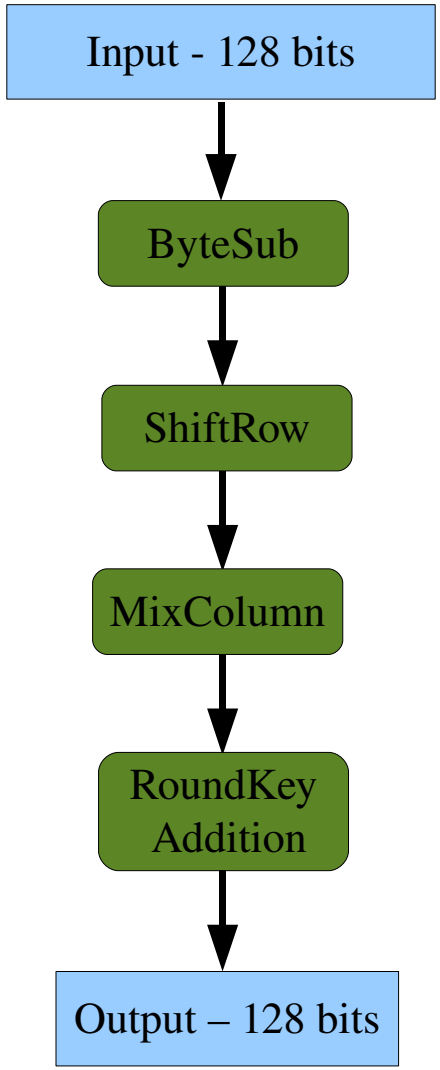
n_{j-1} is calculated as follows

- $\lambda = \text{S-Box}(w_{j-1})$
- n_{j-1} = Cyclic shift λ (Top byte moves to bottom, moving every other byte up one place)

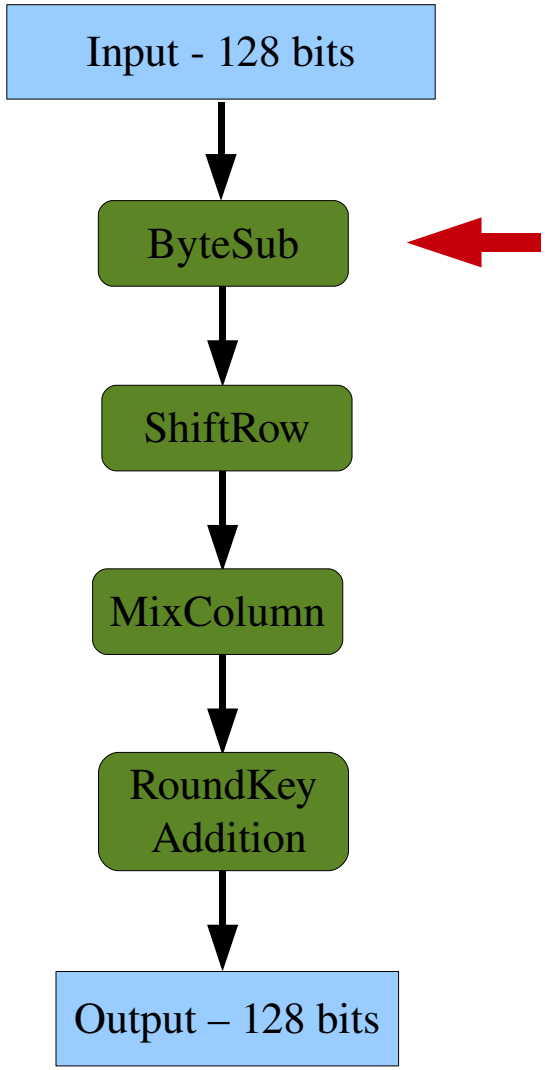
v_j is given as

$$v_j = \begin{pmatrix} \alpha^{(j-4)/4} \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

4. Rounds


Input - 128 bits
ByteSub
ShiftRow
MixColumn
RoundKey Addition
Output – 128 bits

4.1. ByteSub Transformation



4.1. ByteSub Transformation

First step in a round

Nonlinear transformation $\rightarrow$ resistant to linear and differential attacks

Each of the 16 bytes in the A matrix is replaced with a new byte using S-Box substitution $\mapsto B$

4.1. S-Box

	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

4.1. Using the S-Box for the Code

16×16 matrix

NB: S-Box given in hexadecimal

Used to replace a byte with a coded byte:

If byte is represented by ab_{16}, then

- $a \rightarrow$ row
- $b \rightarrow$ column

Example:

Input byte $= 53_{16}$

4.1. Using the S-Box for the Code

16×16 matrix

NB: S-Box given in hexadecimal

Used to replace a byte with a coded byte:

If byte is represented by ab_{16}, then

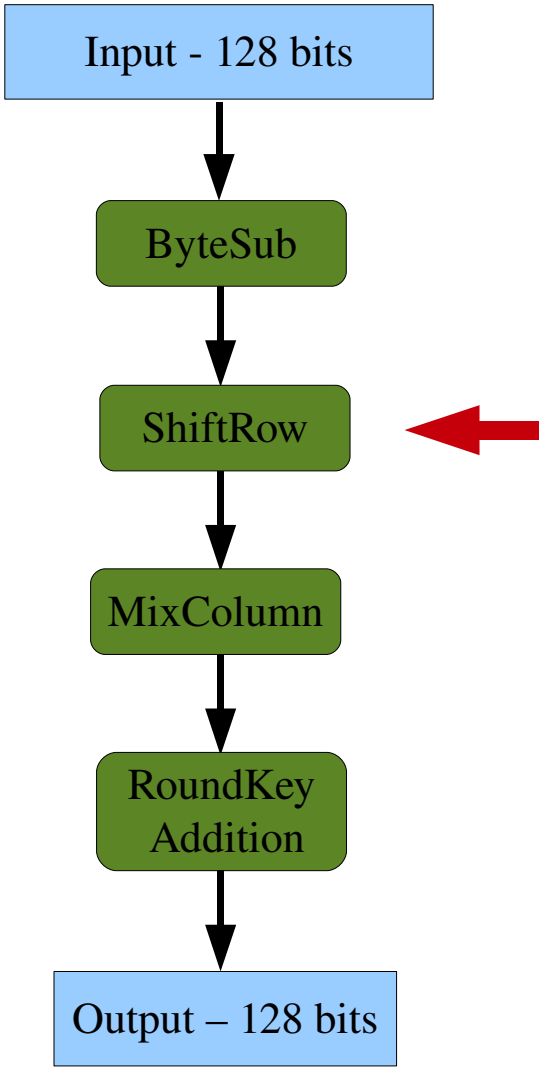
- $a \rightarrow$ row
- $b \rightarrow$ column

Example:

Input byte $= 53_{16}$

Output $= ed_{16}$

4.2. The ShiftRow Transformation



4.2. The ShiftRow Transformation

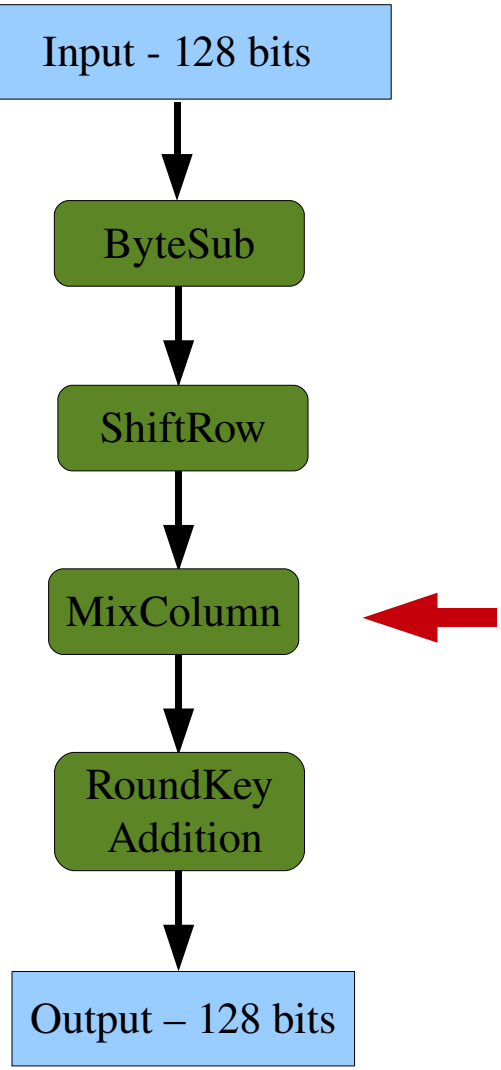
Second step

Linear step which introduces diffusion

Row j of matrix is shifted cyclically to the left by j offsets

$$C = \begin{pmatrix} B_{0,0} & B_{0,1} & B_{0,2} & B_{0,3} \\ B_{1,1} & B_{1,2} & B_{1,3} & B_{1,0} \\ B_{2,2} & B_{2,3} & B_{2,0} & B_{2,1} \\ B_{3,3} & B_{3,0} & B_{3,1} & B_{3,2} \end{pmatrix}$$

4.3. The MixColumn Transformation



4.3. The MixColumn Transformation

Third step

Creates high diffusion between the columns over multiple rounds of the code

$$MC = \begin{pmatrix} \alpha & \alpha+1 & 1 & 1 \\ 1 & \alpha & \alpha+1 & 1 \\ 1 & 1 & \alpha & \alpha+1 \\ \alpha+1 & 1 & 1 & \alpha \end{pmatrix} \begin{pmatrix} B_{0,0} & B_{0,1} & B_{0,2} & B_{0,3} \\ B_{1,1} & B_{1,2} & B_{1,3} & B_{1,0} \\ B_{2,2} & B_{2,3} & B_{2,0} & B_{2,1} \\ B_{3,3} & B_{3,0} & B_{3,1} & B_{3,2} \end{pmatrix}$$

6. The MixColumn Transformation

Example:

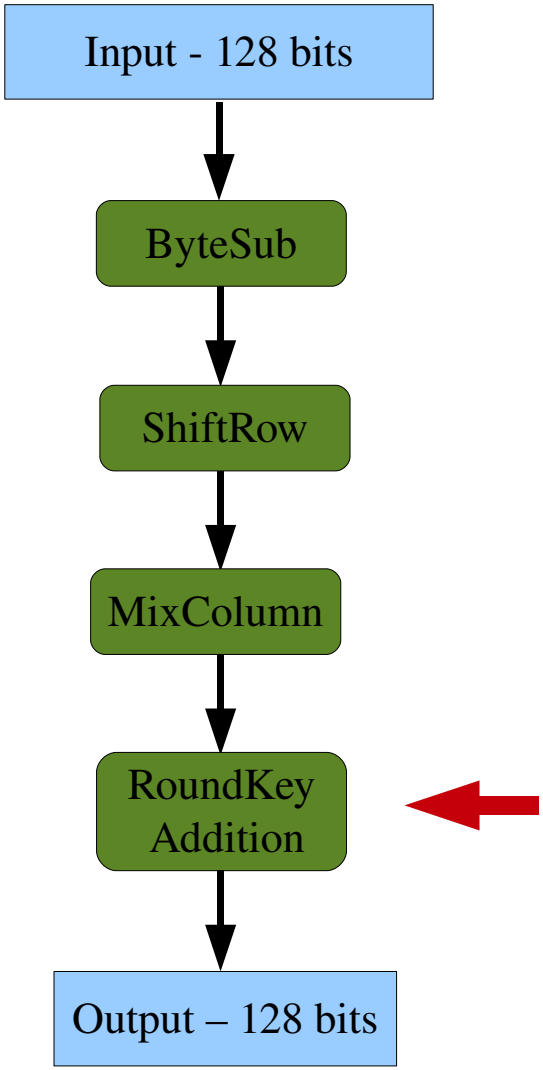
$$
\begin{aligned}
MC &= \begin{pmatrix} \alpha & \alpha+1 & 1 & 1 \\ 1 & \alpha & \alpha+1 & 1 \\ 1 & 1 & \alpha & \alpha+1 \\ \alpha+1 & 1 & 1 & \alpha \end{pmatrix} \begin{pmatrix} 81_{16} & B_{0,1} & B_{0,2} & B_{0,3} \\ 0_{16} & B_{1,2} & B_{1,3} & B_{1,0} \\ 9_{16} & B_{2,3} & B_{2,0} & B_{2,1} \\ 2a_{16} & B_{3,0} & B_{3,1} & B_{3,2} \end{pmatrix} \\
&= \begin{pmatrix} x_0 & D_{0,1} & D_{0,2} & D_{0,3} \\ x_1 & D_{1,1} & D_{1,2} & D_{1,3} \\ x_2 & D_{2,1} & D_{2,2} & D_{2,3} \\ x_3 & D_{3,1} & D_{3,2} & D_{3,3} \end{pmatrix}
\end{aligned}
$$

6. The MixColumn Transformation

Example:

$$
\begin{aligned}
MC &= \begin{pmatrix} \alpha & \alpha+1 & 1 & 1 \\ 1 & \alpha & \alpha+1 & 1 \\ 1 & 1 & \alpha & \alpha+1 \\ \alpha+1 & 1 & 1 & \alpha \end{pmatrix} \begin{pmatrix} 81_{16} & B_{0,1} & B_{0,2} & B_{0,3} \\ 0_{16} & B_{1,2} & B_{1,3} & B_{1,0} \\ 9_{16} & B_{2,3} & B_{2,0} & B_{2,1} \\ 2a_{16} & B_{3,0} & B_{3,1} & B_{3,2} \end{pmatrix} \\
&= \begin{pmatrix} 3a_{16} & D_{0,1} & D_{0,2} & D_{0,3} \\ b0_{16} & D_{1,1} & D_{1,2} & D_{1,3} \\ ed_{16} & D_{2,1} & D_{2,2} & D_{2,3} \\ c5_{16} & D_{3,1} & D_{3,2} & D_{3,3} \end{pmatrix}
\end{aligned}
$$

4.4. The RoundKey Addition



4.4. RoundKey Addition

Final step

At end of jth round:

- Find (4×4) $E_j = [w_{4j}; w_{4j+1}; w_{4j+2}; w_{4j+3}]$
- Add E_j and MC

(E_j is a submatrix of W, consisting of the columns $4j$, $4j + 1$, $4j + 2$ and $4j + 3$)

4.5. Overview of Rijndael

Step 1: Create W-matrix from key, add (w_0, w_1, w_2, w_3) to input data

Step 2. Nine rounds (ByteSub, ShiftRow, MixColumn, RoundKeyAdd) using appropriate columns of W

Step 3. Final round (ByteSub, ShiftRow, RoundKeyAdd) with last subkey

5. Decryption of AES

Decryption based on fact that every substep (ByteSub, Shiftrow, MixColumn and RoundKey Addition) is invertible.

Inverse of ByteSub step: Inverse map exists for S-box

Inverse of Shiftrow: shifting row to the right same number of times as was shifted left

Inverse of MixColumns: Inverse exists for $M \rightarrow M^{-1}$

- Encryption j round: $MC + E_j = H$
- Thus $M^{-1}MC + M^{-1}E_j = M^{-1}H$
- $C = M^{-1}E_j + M^{-1}H$

5. Overview: Decryption of AES

Step 1: Perform RoundKey Addition, using last RoundKey

Step(2): Nine rounds of Inverse ByteSub, Inverse ShiftRow, Inverse MixColumn and Inverse Roundkey Addition, using roundkeys in opposite order.

Step(3): Final round consisting of Inverse ByteSub, Inverse ShiftRow, Roundkey Addition, using keyword.

Summary

1. Motivation
2. Introduction to algorithm and concepts
3. Key schedule
4. Rounds
5. Decryption