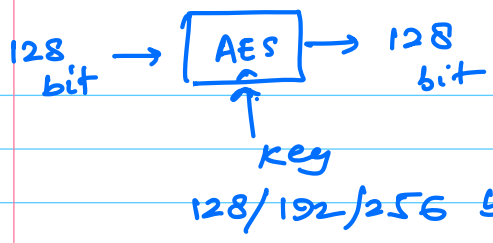
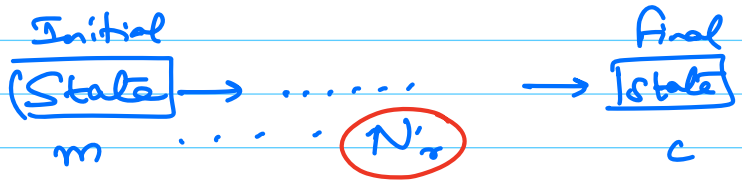
11 Feb 2025



+ some mode of operation (for domain extension)

Input: state array of 4×4 bytes = 16 bytes = 128 bits



	secret key size (bits)	No. of rounds (N_r)
AES-128	128	10
AES-192	192	12
AES-256	256	14

AES-128

Round keys $(K_0, K_1, K_2, \ldots, K_{10})$ = AES keyschedule (K)

$K_i \in \{0,1\}^{128}$

K ← secret key

State[0] = input msg

m_0	m_4		
m_1	m_5		
m_2	.		
m_3	:		

key whitening → State[0] = State[0] ⊕ K_0

for (i = 1 to 9) ← first 9 rounds

{

Round(i, State[i], K_i);

}

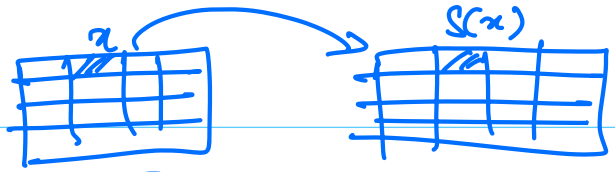
Final Round (10, State[10], K_{10});

→ Last step in final Round will also be key XOR

Round function:

inverse in $GF(2^8)$

(i) SubBytes

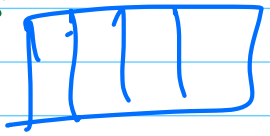


$S(x) = A x^{-1} + b$

(ii) ShiftRows

→ circular rotation of each row by a constant

(iii) MixColumns



→ Replace each column by a different column

(iv) AddRoundKey

→ new state

$C = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$ → 1 column of 4 bytes

$\equiv a_0 + a_1 x + a_2 x^2 + a_3 x^3$

where $a_i \in GF(2^8)$

$C * M(x) = (\ldots)$ → degree may increase

divide by a polynomial of degree = 4

$(x^4 + \ldots)$

$$(a_0 + a_1 x + a_2 x^2 + a_3 x^3) * (b_0 + b_1 x + b_2 x^2 + b_3 x^3) \quad \text{mod } (x^4+1)$$

$$= a_0 b_0 +$$
$$(a_0 b_1 + a_1 b_0)\, x +$$
$$(a_0 b_2 + a_1 b_1 + a_2 b_0)\, x^2 +$$
$$(a_0 b_3 + a_1 b_2 + a_2 b_1 + a_3 b_0)\, x^3 +$$
$$(a_1 b_3 + a_2 b_2 + a_3 b_1)\, x^4 +$$
$$(a_2 b_3 + a_3 b_2)\, x^5 +$$
$$(a_3 b_3)\, x^6$$

Notice:

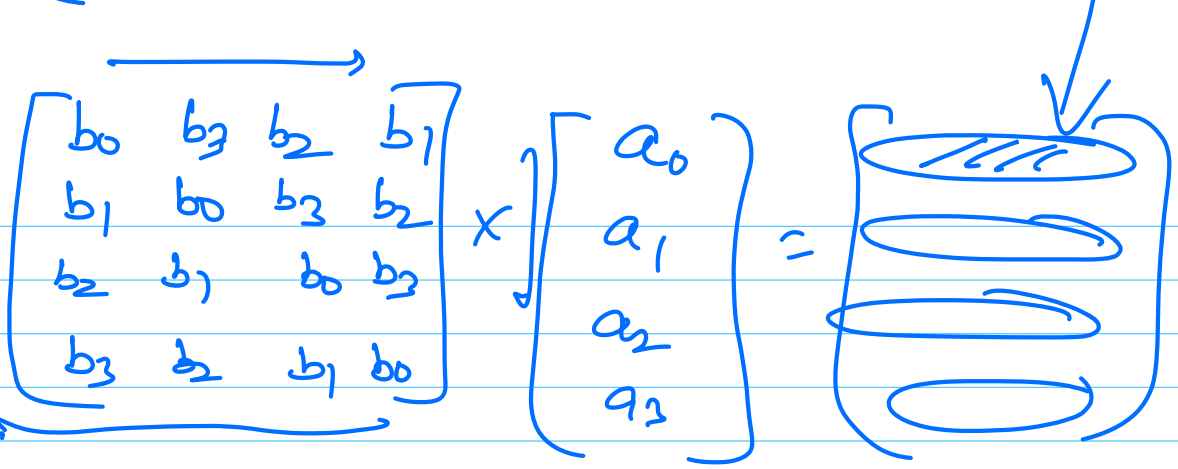
$$x^4 / (x^4+1) = \frac{x^4+1+1}{x^4+1} = 1$$

$$x^5 / (x^4+1) = x$$

$$x^6 / (x^4+1) = x^2$$

$\Rightarrow$

$$(a_0 b_0 + a_1 b_3 + a_2 b_2 + a_3 b_1)$$
$$+ (a_0 b_1 + a_1 b_0 + a_2 b_3 + a_3 b_2)\, x$$
$$+ (a_0 b_2 + a_1 b_1 + a_2 b_0 + a_3 b_3)\, x^2$$
$$+ (a_0 b_3 + a_1 b_2 + a_2 b_1 + a_3 b_0)\, x^3$$



Unfortunately, (x^4+1) is not irreducible

$$(x^4+1) = (x^2+1)^2$$

$\Rightarrow$ this matrix arose from a polynomial

$$b(x) = (b_0 + b_1 x + b_2 x^2 + b_3 x^3)$$

Ensure that $b(x)$ is coprime to (x^4+1)

Why is last round different?

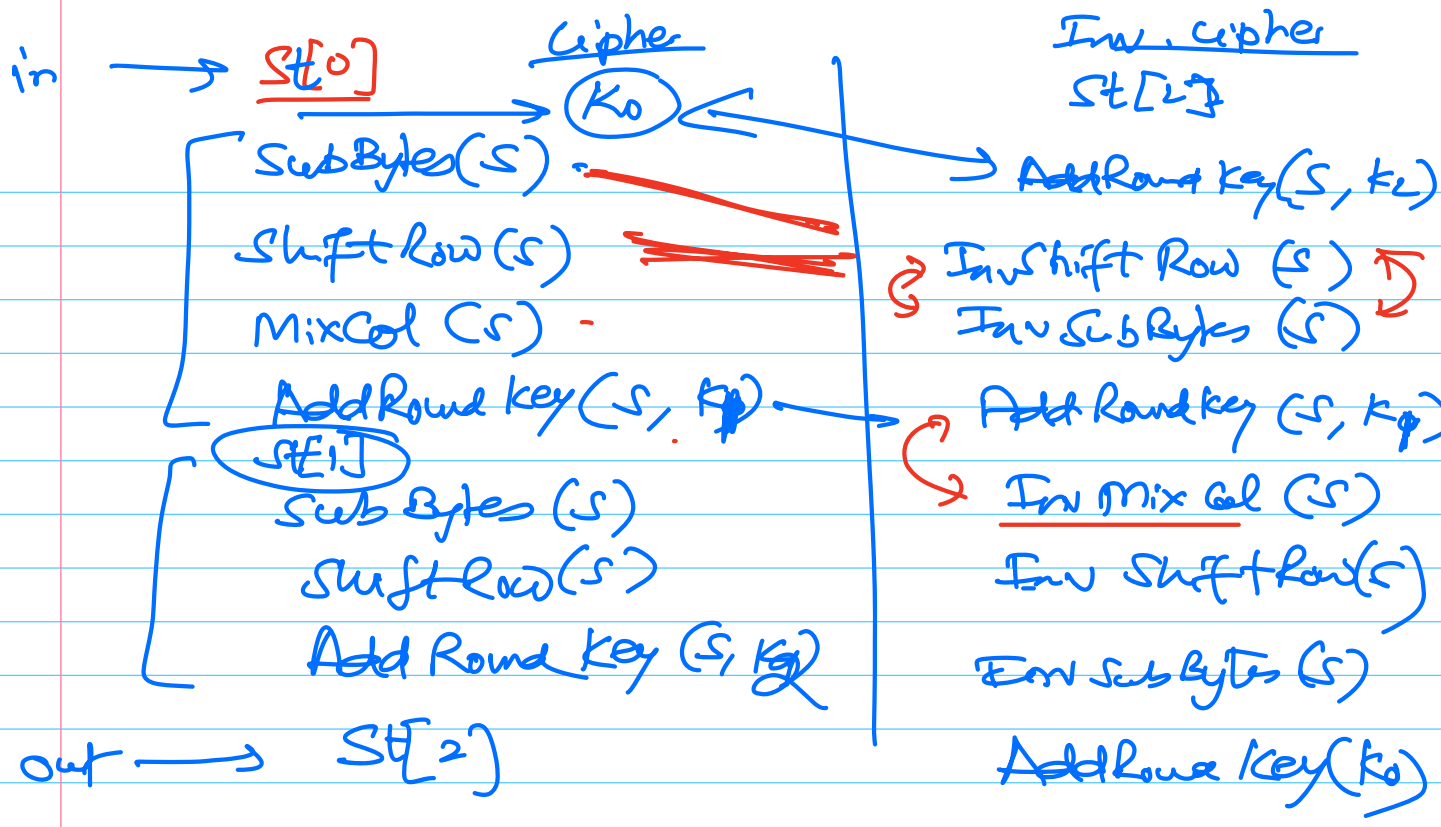
to make the structure of encryption & decryption operations same

SubBytes

ShiftRow

~~MixCol~~

~~Add~~ RoundKey



MixCol(S) → M.S

AddRound(S, K) → S ⊕ K

→ modify the key to reverse the operation

$$M \cdot (S \oplus \vec{m}^{-1} K) = MS \oplus K$$

Cipher Round keys:

$K_0, K_1, K_2 \ldots K_{10}$

Inv round keys:

$K_{10}, (K_9)', (K_8)' \ldots (K_1)', K_0$

modified keys

SPN ciphers -

Substitution permutation Network

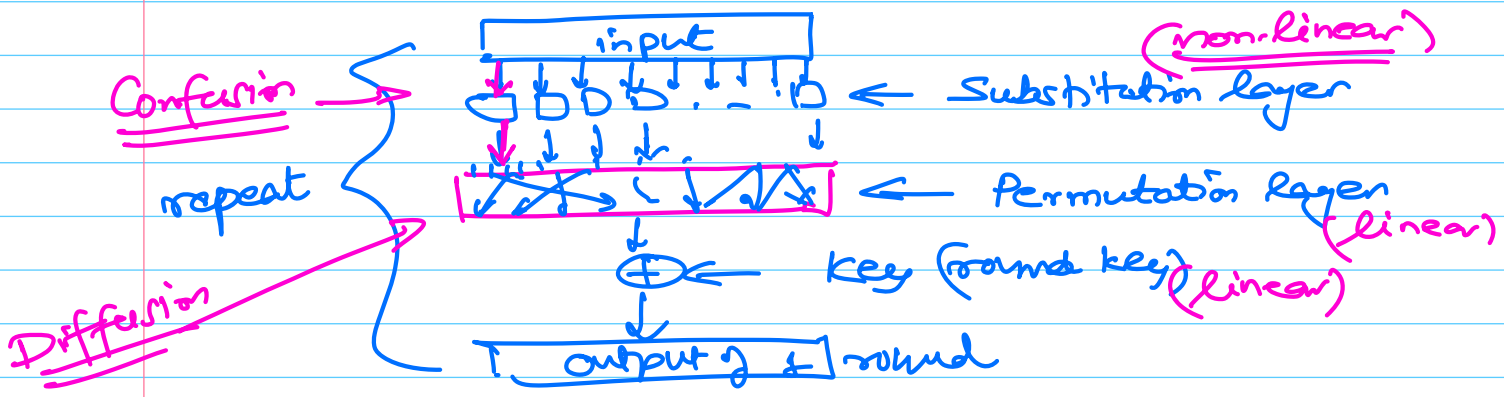
AES design document

— NIST

for ref.

13-02-2025

SPN structure:



'if a cryptosystem is linear Then it is easy to break. → $C = Ax + k$

$= [A'][x\ 1]$

Gaussian elimination → solve for A'.

Ref. → Hill cipher

Avalanche effect — a small input change causes drastic change in the output

(Caused by combination of Subs. & Perm. layers)