AES 10/10/22

DES 56 bit key

By 1998, DES Challenge

- Double IDLE CPU cycles
- Many people in parallel, running regular computers would try all the keys.

* Key broke in ~ 2 months

NIST (National Institute of Standards Tech)

New Govt Standard Private key Encryption

1) Must support blocks sizes of 128 bits, 192 bits + 256 bits (shift to longer block size if nece.)
2) More Secure Triple DES $56 \times 2 = 112$
   $C = E_{K_1}(E_{K_2}(E_{K_1}(P)))$, effectively ~~128~~ bits
3) Public Domain
4) Secure for at least 30 yrs.

Rijndael (Pronounced "Rain Doll")

John Daemon + Vincent Rijment

→ Winner of contest

FYI, algorithm based on the field $GF(2^8)$ with polynomial $m(x) = x^8 + x^4 + x^3 + x + 1$.

Only describe 128 bit version

Alg

1. $S = AddRoundKey(P, K_0)$

2. for $i = 1$ to 10:
   $S = SubBytes(S)$
   $S = ShiftRows(S)$
   if $(i < 10)$ $S = MixCols(S)$
   $S = AddRoundKey(S, K_i)$

3. Return S

State Matrix

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

1st 4 2nd 4 3rd 4

b_{00} = byte #1
b_{10} = byte #2
b_{20} = byte #3
b_{30} = byte #4

Read columns not rows.

example

A4	B1	89	69
37	2C	AF	78
DE	45	F2	D4
09	67	00	59

typically denoted in hex.

Add Round Key

return $S \oplus K_i$

Sub bytes

for each byte b_{ij} replace it with

$Sbox(b_{ij})$

Examples:

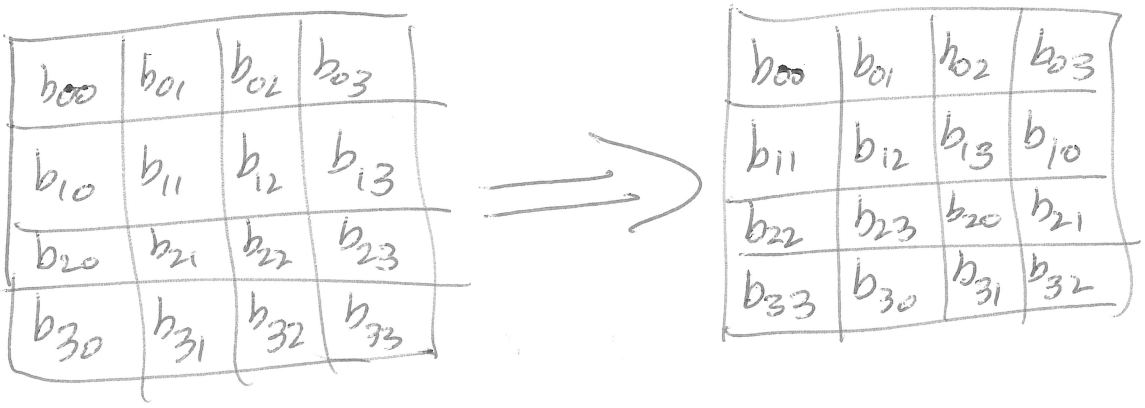
$Sbox(C4) = 1C$

$Sbox(2B) = F1$

$Sbox(A7) = 5C$

This is easy just a table lookup BUT for security created using math in the field $GF(2^8)$ w/poly $m(x)$.

Shift Rows



Mix Cols

$$\text{Matrix Enc} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}$$

Matrix Multiply in the field $GF(2^8)$ mod $m(x) = x^8 + x^4 + x^3 + x + 1$

Sample State Matrix

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} A4 & B1 & 89 & 69 \\ 37 & 2C & AF & 78 \\ DE & 45 & F2 & D4 \\ 09 & 67 & 00 & 59 \end{bmatrix}$$

row 2 x col 3

Row 2 Col 3 entry = $01 \times 89 + 02 \times AF + 03 \times F2 + 01 \times 00$

these objects represent polynomials

$01 \times Any = Any$

$02 \times$ byte = left shift by 1 spot

$02 \times AF = 10101111 << 1$

if overflow 1, = 1 0101 1110

then remove it and replace it with 0001 1011 then XOR w/ 8 LSBits

$$= 0101\ 1110$$
$$\oplus\ 0001\ 1011$$
$$01000101 \quad (45)$$

$03 \times B = 01 \times B + 02 \times B$, Note: + is XOR

$03 \times F2 = 01 \times F2 \oplus 02 \times F2$

$= F2$
$= \oplus FF$
$0D$

$02 \times F2 = 1\ 1110\ 0100$
$\oplus\ 1\ 1011$
$1111\ 1111 \quad (FF)$

Goal: $01 \times 89 + 02 \times AF + 03 \times F2 + 01 \times 00$

$= 89 \oplus 45 \oplus 01 \oplus 00 =$

$= \boxed{C1}$

1001
0101
1101
0001