

3TDEA

Triple Data Encryption Algorithm

$$c = C(D(C(m, k_1), k_2), k_3)$$

Per mantenere la retrocompatibilità, si sceglie le stesse chiavi per k_1, k_2, k_3 . Usare D antidi C in mezzo non aumenta la sicurezza

2TDEA

$$c = C(D(C(m, k_1), k_2), k_1)$$

la chiave è 112 bit sia in 2TDEA

sia in 3TDEA

Attacchi meet in the middle

$$D(c, k_3) = D(C(m, k_1), k_2)$$

$$C(D(c, k_3), k_1) = C(m, k_1)$$

Dato una coppia (m, c) :

1. Per ogni k_1 , si calcola e si salva $C(m, k_1)$
 2^{56} operazioni di cifratura

2. Per ogni (k_2, k_3) calcolo $C(D(m, k_3), k_2)$
e si cerca nella tabella
 $2^{56} \cdot 2^{56}$ elementi da enumerare

Costo: $O(2^b + 2^b \cdot 2^b)$ con $b = (k \mid$

Su chiavi da 56 bit: costa tant. quanto

un attacco forte bruto

AES

Cifrario \rightarrow blochi di 128 bit

chiavi da 128, 192 o 256 bit

| | chiave | dim. blocco | num. fasi |
|---------|--------|-------------|-----------|
| AES 128 | 128 | 128 | 10 |
| AES 192 | 192 | 128 | 12 |
| AES 256 | 256 | 128 | 14 |

Chiave e messaggio vengono caricati in tabella da 4x4 byte.

| | | |
|---|---|---|
| | b _{0,0} b _{0,1} - - - | K _{0,0} K _{0,1} - - - |
| B | b _{1,0} - - - | W K _{1,0} - - - |

| | | |
|-----------|---------|----------------|
| Messaggio | - - - - | chiave - - - - |
| | - - - - | - - - - |



\hookrightarrow regole e'

i multipli di 4 $W[i] = W[-4] \oplus W[-1]$

alt

$$W[i] = W[-4] \oplus T(W[-1])$$

transformation non lineare

m

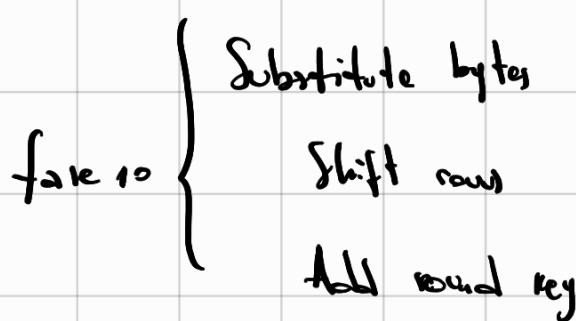
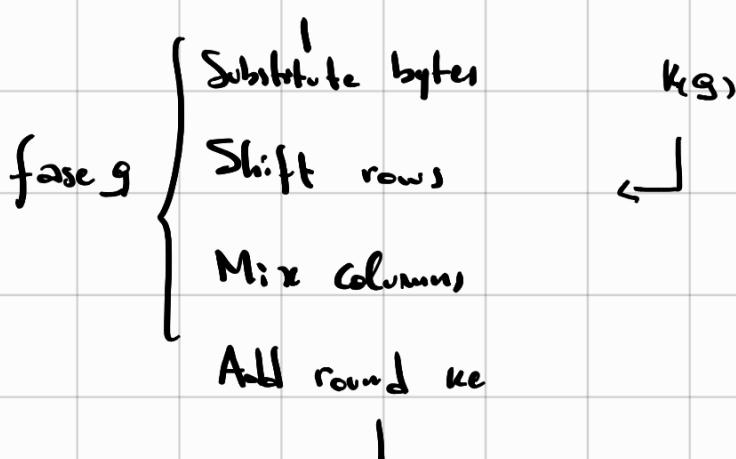
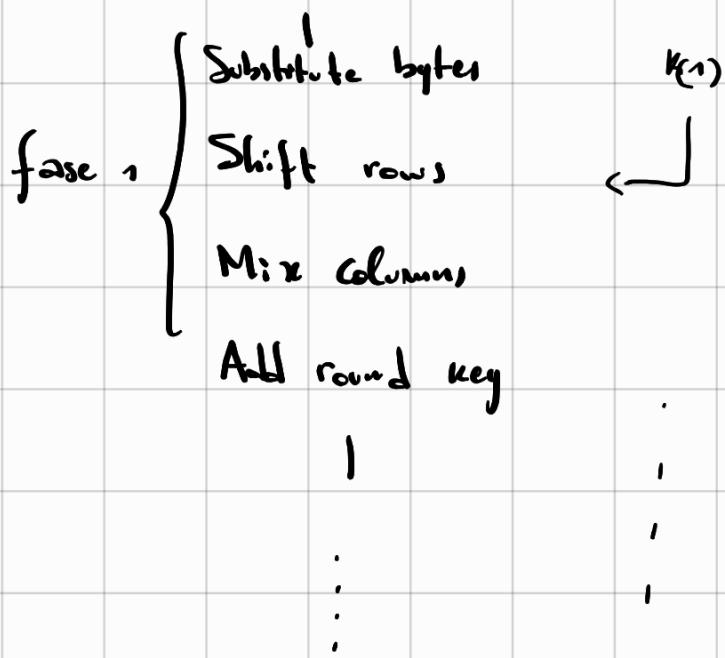
k

| 128 bit

| 128 bit

Summa mod 2

(S-box)



$\hookrightarrow c (128 \text{ bit})$

Substitute bytes:

Ogni byte del blocco B e' trasformato mediante

una S-box

$$B \xrightarrow{\text{sub. bytes}} A$$

$$a_{i,j} = S_{box}(b_{i,j})$$

Contiene tutti i numeri da 0 a 255, li permuta.

Ogni byte $b_{i,j}$ viene sostituito con il suo inverso multiplicative

in $GF(2^8)$, moltiplicato per una matrice 8×8 bit

e sommat. con un vettore colonna

→ non e' lineare, le altre operazioni lo sono

ovv. un byte si come un polinomio di grado 7

$$1 + x^1 + \dots + x^7$$

si sommano mod 2

il prodotto si fa come prodotto tra polinomi a tris

per un polinomio irriducibile

Shift rows:

I byte di ogni riga vengono shiftati ciclicamente
di 0, 1, 2, 3 posizioni ripetutamente.

ovv. I byte di ogni colonna si disperdono su 4 colonne

Mix columns

Ogni colonna del blocco (vista come un vettore di 4 elementi) viene moltiplicata per una matrice 4×4 byte
(la moltiplicazione e' eseguita mod 2^8 = la somma
mod 2)

M e' scelta in modo tale che ogni byte della colonna venga mappato in un nuovo byte che e' funzione dei 4 byte presenti.

Add round key

Ogni byte della matrice si posta in 8 bit a bit con un byte della chiave dello stesso corrispondente

Sicurezza

Tutti i bit della chiave sono bit di sicurezza

Esistono attacchi meno costosi di forza bruta se le fai sono 6.

Si conoscono attacchi side channel che non sfruttano le debolezze del cifrario, bensì quelle del sistema che lo implementa

Si fa uso di padding nel caso in cui le lunghezze del

svantaggio: non sia un multiplo della lunghezza del blocco

ott. blocchi uguali nel messaggio producono
blocchi uguali nel cifrogramma

Esistono tecniche di composizione di blocchi

Cipher Block Chaining (CBC)

- Si compongono i blocchi fra loro

$$\text{Cifratura: } c_i = C(m \oplus c_{i-1}, k)$$

$\uparrow c$ del blocco $i-1$

$c_0 :=$ stringa di $b = 1$ blocco / bit concordata

con l'altra utente (scambiabile
anche in chiaro)

ott. fatti messaggi uguali in blocchi diversi

svantaggio: la cifratura è sequenziale, non si parallelizza

$$\text{Decifratura: } m_i = c_{i-1} \oplus D(c_i, k)$$

ott. si può fare in parallelo, serve solo il cifrogramma
e non la sua decifratura

Eventuali errori nella trasmissione del blocco: componere
la decifratura dei blocchi $i, i-1$

Esercizio

1) Suppongo funzione di cifratura \rightarrow blochi da 128 bit. Chiedere

$$Cl(m_1 \oplus m_2, k) = Cl(m_1, k) \oplus Cl(m_2, k)$$

Come attaccarlo?

Definisco $e^{(i)} = 0 \dots 0 \dots 1 \dots 0$

vettore di tutti 0 tranne in posizione i :

Chiedo la decifratura di tutti gli $e^{(i)}$.

$$\forall i, 1 \leq i \leq 128 \quad f^{(i)} = D(e^{(i)}, k)$$

$$m = D(c, k) = D\left(\bigoplus_{i=1}^n e^{(i)}, k\right)$$

2) Sbox AES: 16 blocchi, 8 ingressi, 8 uscite

Qual è il numero totale di funzioni che si

potrebbero scegliere per ciascun blocco?

$$\{1, 2, \dots, 2^8\} \rightarrow \{1, 2, \dots, 2^8\}$$

$$(2^8)^{2^8} = 2^{8 \cdot 2^8} = 2^{2^M}$$

3) $c_i = m_{i-1} \oplus L(m_i \oplus c_{i-1}, k)$ mo, co noti e pubblici

1) Come si decifra?

2) Cosa succede se ci si corrompe la trasmissione?

$$c_i \oplus m_{i-1} = C(m_i \oplus c_{i-1}, k)$$

$$D(c_i \oplus m_{i-1}, k) = D(C(m_i \oplus c_{i-1}, k), k)$$

$$D(C(m_i \oplus c_{i-1}, k)) = m_i \oplus c_{i-1}$$

$$\rightarrow m_i = D(c_i \oplus m_{i-1}, k) \oplus c_{i-1}$$

Se c_i e' danneggiato, m_i e' corrotto e lo sono

anche tutti i msg con $j > i$.