TDEA

Triple Data Encryption Algorithm.

3TDEA $C_3(m, k_1, k_2, k_3) = C_{\text{DES}}(D_{\text{DES}}(C_{\text{DES}}(m, k_1), k_2), k_3)$

2TDEA $C_2(m, k_1, k_2) = C_{\text{DES}}(D_{\text{DES}}(C_{\text{DES}}(m, k_1), k_2), k_1)$

La decifrazione al secondo passo non aumenta la sicurezza rispetto a 3 cifrature, ma è preferita per retrocompatibilità con i sistemi a cifratura singola (usando $k_1 = k_2 = k_3$).

$2 \cdot 56 = 112$ bit di sicurezza in entrambi i casi; si prefereisce 2TDEA perché richiede una chiave in meno.

Meet-in-the-middle

Il motivo per cui 3TDEA non ha $3 \cdot 56 = 168$ bit di sicurezza è che è vulnerabile ad un attacco meet-in-the-middle.

$$C_{\text{DES}}(D_{\text{DES}}(c, k_3), k_2) = C_{\text{DES}}(m, k_1)$$

- è nota una coppia (m, c);
- memorizziamo in una tabella il risultato di $C_{\text{DES}}(m, k_1)$ per ogni k_1 possibile;
- per ogni coppia (k_2, k_3), se $C_{\text{DES}}(D_{\text{DES}}(c, k_3), k_2)$ è contenuto nella tabella alla posizione k_1 allora probabilmente (k_1, k_2, k_3) è la chiave.

Costo: $\text{O}(2^{56} + 2^{56} \cdot 2^{56}) = \text{O}(2^{112})$ operazioni di cifratura e decifrazione.

Il meet-in-the-middle su 2TDEA ha sempre costo 2^{112}, quindi non è meglio di forza bruta.