Algoritmo IDEA

Di Simone Galdino

IDEA – Cenni Storici

- IDEA (International Data Encryption Algorithm) fu creato da X.Lay e J.Massey
- Nel 1991 Lay e Massey ne crearono una prima versione, chiamata PES (Proposed Encryption Algorithm).
- Questa era facilmente attaccabile dalla crittoanalisi differenziale, quindi gli autori crearono l'IPES (Improved Proposed Encryption Algorithm)
- Questo è l'algoritmo oggi conosciuto come IDEA.

L'Idea di IDEA

- IDEA fa parte della versione ufficiale del PGP, ed è fondamentale per la sicurezza della posta elettronica.
- IDEA nasce con la volontà di creare un algoritmo semplice da applicare e difficile da forzare.
- L'intento era sostituire il DES, già forzato con un attacco in simultanea da più calcolatori, utilizzando un algoritmo resistente ad eventuali Sig.X dotati di calcolatori con elevata potenza di calcolo.
- Per ora, con IDEA, pare che i due autori abbiano raggiunto il loro scopo.

IDEA – Caratteristiche (1)

- IDEA è un algoritmo di cifratura a blocchi, come il DES.
- Proprio come tutti gli algoritmi simmetrici, IDEA ha una parola chiave, usata sia per cifrare che per decifrare.
- Tale chiave ha lunghezza di 128 bit, contro i 64 (ridotti a 56) bit usati nel DES.
- Intuitivamente, si capisce già che una parola chiave di tali dimensioni rende la probabilità di trovare la chiave, per chi non ne sia a conoscenza, estremamente bassa.
- La ricerca esaustiva, infatti, è praticamente impossibile: le chiavi possibili sono 2^128!

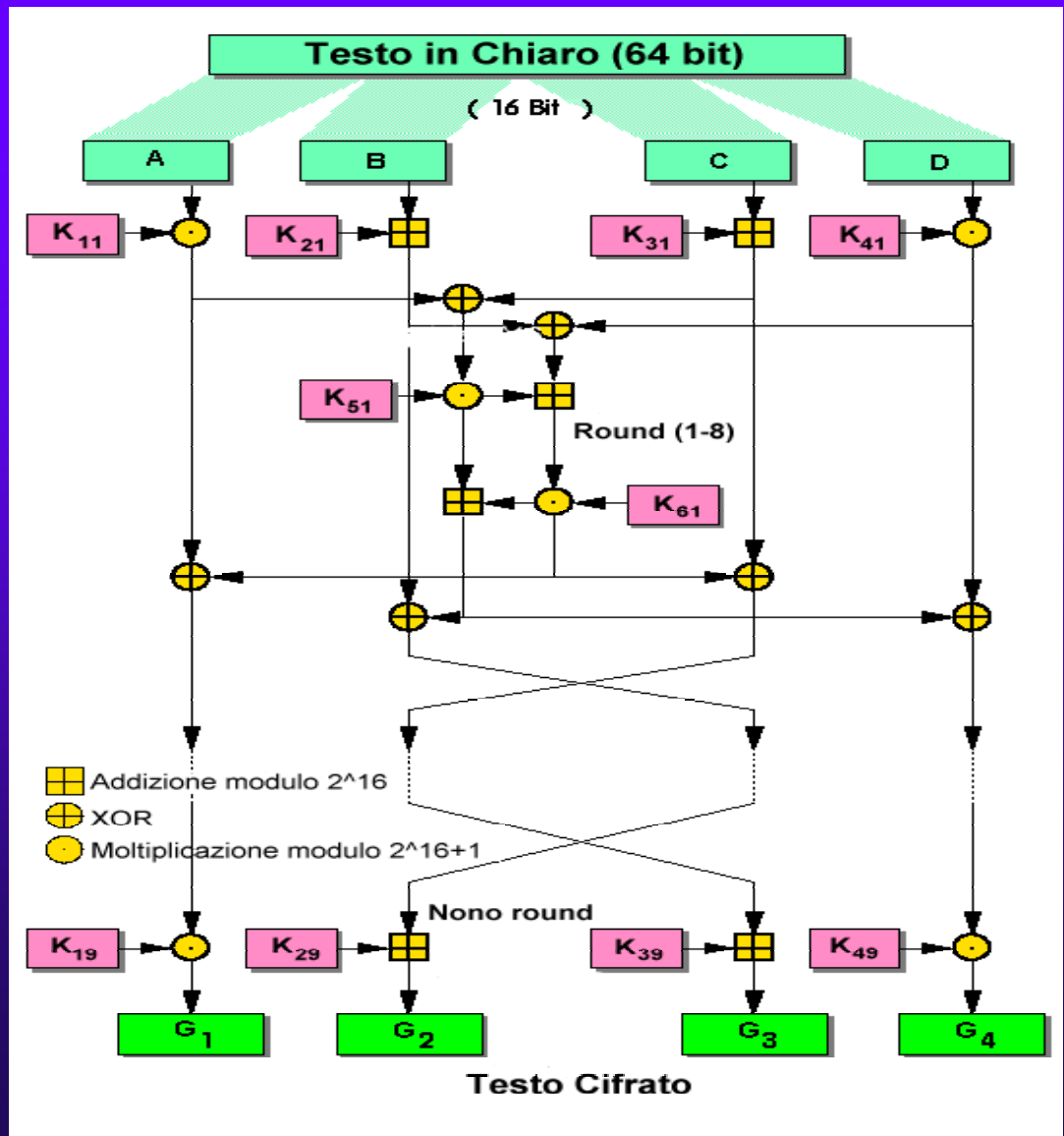
IDEA – Caratteristiche (2)

- IDEA fa uso di 8 round che si ripetono uguali, più il 9° e ultimo passo leggermente differente.
- Per questo, causa il funzionamento dell'algoritmo stesso, come vedremo, IDEA fa uso di ben 52 sottochiavi (6 per ognuno degli 8 passi e 4 per il 9° e ultimo passo).
- Il testo in chiaro è processato dall'algoritmo in blocchi da 64 bit.
- L'algoritmo fa uso, oltre che dello XOR (⊕) come il DES, anche dell'addizione modulo 2^16 (⊞) e della moltiplicazione modulo 2^16+1 (⊙).

IDEA – MUL mod 2^16+1

- Come detto, IDEA fa uso delle operazioni XOR, ADD mod 2^16 e MUL mod 2^16+1.
- Per poter decodificare il testo cifrato, queste operazioni devono essere invertibili. Le prime due (XOR e ADD) sappiamo esserlo.
- La moltiplicazione modulo M è invertibile se e solo se entrambi i fattori sono primi con M.
- Notiamo che 2^16+1=65537, che è numero primo. Per questo la moltiplicazione, nel nostro caso, è sempre invertibile, tranne per 0, che viene considerato come 65537.

IDEA – Algoritmo



IDEA – Gestione della chiave (1)

- Come detto, IDEA usa chiavi di 128 bit.
- Al primo passo, la chiave viene suddivisa in 8 blocchi da 16 bit ciascuno.
- Le prime 6 chiavi così generate vengono usate direttamente nel primo passo dell'algoritmo.
- Nei passi successivi, la chiave usata nel passo precedente viene shiftata ciclicamente di 25 posizioni a sinistra, per poi essere suddivisa nuovamente in 8 blocchi da 16 bit.

IDEA – Gestione della chiave (2)

- Nell'ultimo passo vengono utilizzati soltanto i primi 4 blocchi ottenuti dalla chiave shiftata.
- La notazione usata nello schema, per le chiavi, è questa:

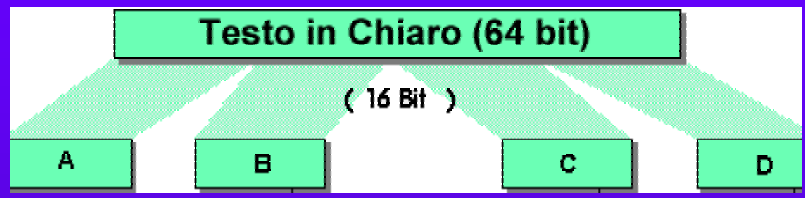
$$K_{ij}$$

Dove i=numero della sottochiave

j=numero del round attuale

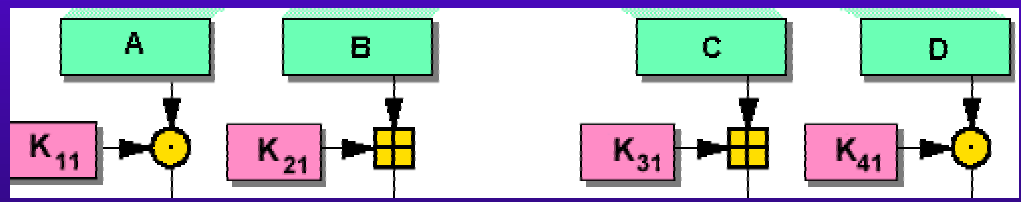
IDEA – Algoritmo (1)

- Ad ogni passo, il blocco del messaggio da cifrare (64bit) viene suddiviso in 4 blocchi da 16 bit, che chiameremo A, B, C, D.



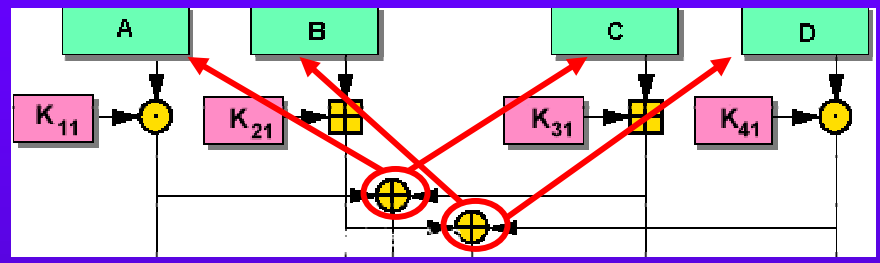
- Ad ogni passo i (1<=i<=8) avvengono le seguenti operazioni:

1. MUL (A, K_{1i})
2. ADD (B, K_{2i})
3. ADD (C, K_{3i})
4. MUL (D, K_{4i})

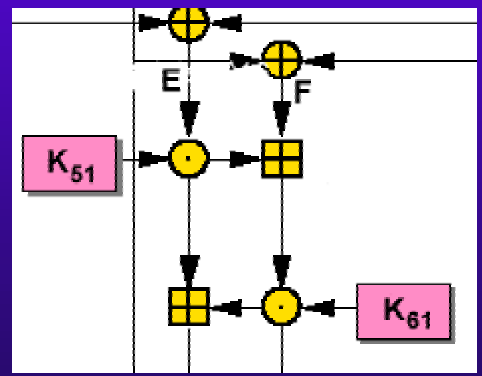


IDEA – Algoritmo (2)

5. E = XOR (A, C)
6. F = XOR (B, D)

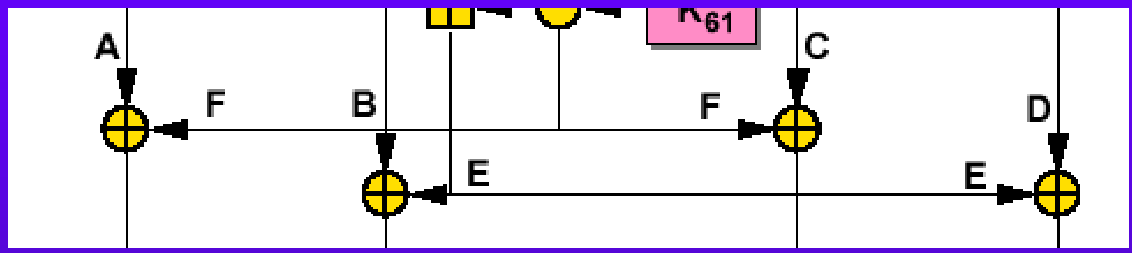


7. MUL (E, K_{5i})
8. SUM (E, F)
9. MUL (F, K_{6i})
10. SUM (F, E)

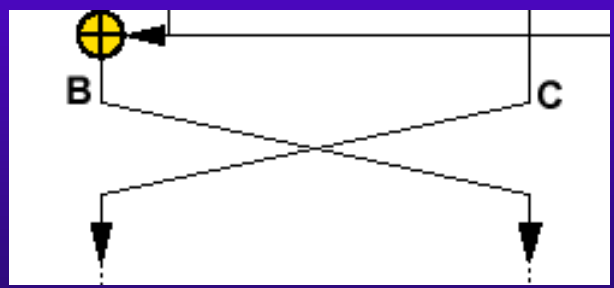


IDEA – Algoritmo (3)

11. XOR (A, F), XOR (C, F)
12. XOR (B, E), XOR (D, E)



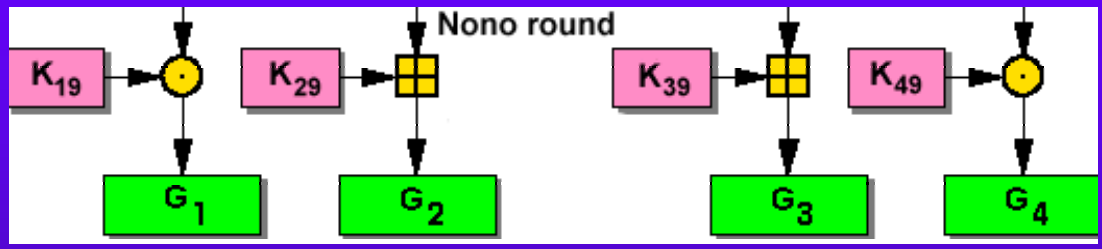
13. scambio B e C



IDEA – Algoritmo (4)

- Nell'ultimo round, il 9°, vengono eseguiti meno passaggi. Questi:

1. MUL (A, K_{19})
2. ADD (B, K_{29})
3. ADD (C, K_{39})
4. MUL (D, K_{49})



- A questo punto, l'algoritmo è terminato, e G_1, G_2, G_3 e G_4 vengono semplicemente giustapposti per ottenere il testo cifrato da inviare.

IDEA – Decifrazione (1)

- La decifrazione del messaggio cifrato, segue lo stesso identico schema.
- Ma:
  - Le chiavi utilizzate sono gli inversi delle chiavi usate per la cifratura rispetto a MUL e ADD.
- Le prime sei chiavi del primo passo sono:

$K_{1d} = K_{19}^{-1}$
$K_{2d} = \sim K_{29}$
$K_{3d} = \sim K_{39}$
$K_{4d} = K_{49}^{-1}$
$K_{5d} = K_{18}$
$K_{6d} = K_{28}$

IDEA – Decifrazione (2)

- Per ogni passo successivo della decifrazione le chiavi sono generate allo stesso modo, fino al 9° e ultimo passo, in cui si hanno le chiavi:

$$K_{49d} = K_{11}^{-1}$$
$$K_{50d} = \sim K_{21}$$
$$K_{51d} = \sim K_{31}$$
$$K_{52d} = K_{41}^{-1}$$

- N.B. il simbolo ~ indica l'inverso più uno.

  ES. ~1110001 = (NOT 1110001) + 1 = 0001110 + 1 = 0001111

IDEA – Sicurezza (1)

- Finora IDEA è sopravvissuto a qualsiasi tentativo di attacco sferratogli.
- Tramite un rapido calcolo, si può verificare che, pur utilizzando un supercalcolatore in grado di generare un miliardo di chiavi al secondo, si impiegherebbero comunque 10^22 anni per portare a termine una ricerca esaustiva.
- Nonostante questo, non si può affermare con certezza che IDEA sia un algoritmo sicuro.

IDEA – Sicurezza (2)

- Questo per due motivi:
  - Una persona molto fortunata potrebbe indovinare la chiave per caso. Ma la probabilità che questo accada è molto molto prossima allo 0;
  - E' possibile che un giorno si trovi il punto debole dell'algoritmo e lo si riesca a forzare.
- Per ora, comunque, IDEA è l'algoritmo più sicuro presente in circolazione.

IDEA – Vantaggi

- La forza di IDEA sta nella segretezza della chiave, e nel fatto, come abbiamo visto, che si ha un range di possibili chiavi pari a 2^128.
- L'Algoritmo è abbastanza facile da applicare ed è anche veloce.
- Si potrebbe rendere ancora più veloce, riducendo il numero di round da 9 a 4. E' stato infatti dimostrato che IDEA diventa impossibile (per ora) da forzare dopo il quarto passo.
- Il suo utilizzo è free, ossia libero da qualsiasi licenza.
- Può essere implementato sia via software che hardware, con semplici componenti che lavorano a 16 bit.