Università degli studi dell'Aquila

Tesina per Combinatoria

Crittografia a chiave Simmetrica: Il DES

Author: Daniele Di Pompeo

January 2014

Contents

List of Tables

Chapter 1

Algoritmi di cifratura

1.1 Algoritmi simmetrici

La categoria degli algoritmi di cifratura a chiave segreta o detti algoritmi simmetrici prevedono che sia la fase di cifratura che di decifratura avvenga con l'utilizzo della stessa chiave. Ovvero il mittente ed il destinatario utilizzino la stessa chiave per lo scambio di messaggi.

Gli algoritmi a chiave simmetrica si possono suddividere in due sotto gruppi:

- **a trasposizione**: gli algoritmi che fanno parte di questa famigli prevedono la cifratura dei messaggio utilizzando tecniche anche complesse di permutazioni del testo in chiaro.
- **a sostituzione**: gli algoritmi di questa famiglia a differenza di quella a trasposizione prevedono la sostituzione di ogni lettera del testo in chiaro, secondo dettagliate tecniche, con una lettera dell'alfabeto segreto

Tra i più famosi ed antichi algoritmi a chiave simmetrica con tecnica della sostituzione si ricorda il *cifrario di cesare*. Questo cifrario venne usato dall'imperatore Cesare per inviare informazioni riservate alle proprie truppe durante l'epoca romana. Esendo basato su un alfabeto segreto di 26 lettere si possono creare 26 differenti cifrari utilizzando la stessa tecnica. Ad ogni lettera del testo in chiaro si sostituisce la corrispondente lettera dell'alfabeto segreto. La chiave di questo cifrario è data dal numero dei posti di cui è stato traslato l'alfabeto segreto.

I cifrari a sostituzione possono essere suddivisi a loro volta in macro categorie che si distinguono per le tecniche di sostituzione adottate. Possono essere:

- **Cifrari Monoalfabetici**: ognu lettera del testo in chiaro viene sempre cifrata con la stessa lettera dell'alfabeto segreto.
  - *Additivi*
  - *Moltiplicativi*
  - *Affini*
- **Cifrari Monoalfabetici in lingue non naturali**:
  - *a flusso*: questi algoritmi prevedono la cifratura a stati dei blocchi di informazione in chiaro. Ovvero la determinata cifratura non varia al variare dei passi dell'algoritmo. Prevedono la cifratura simbolo a simbolo.
  - *a blocchi*: sono algoritmi di cifratura che prevedono la scomposizione del testo in chiaro in blocchi di lunghezza finita, che varia da algoritmo ad algoritmo. Tra i principali algoritmi di annovera l'algoritmo di **Feitsel** che associa ad un testo in chiaro (S_0,D_0) un testo cifrato della stessa lunghezza (S_n,D_n). Intorno agli inizi del 1900 la comunità crittografica si iniziava a domandare se il **DES**[1] si potesse ancora utilizzare come standard per la cifratura. Nel 1991 *Lai & Massey* proposero una nuova idea di algoritmo di cifratura a blocchi che prese il nome di *Proposed Encryption Standard* con l'intento di codificare blocchi di testo in chiaro di 64bit in blocchi di testo cifrato di 64bit tramite una chiave di 128bit. Nel 1992 fu brevettato sotto il nome di *IDEA* e ad oggi risulta essere ancora inviolato ed è utilizzato come standard di cifratura nei software PGP.
- **Cifrari Polialfabetici**: sono quegli algoritmi che prevedono la cifratura di ogni simbolo dell'alfabeto in chiaro tramite simboli dell'alfabeto cifrata non sempre uguali. Fanno parte di questa famiglia di algoritmi:
  - *Cifrario di Alberti*
  - *Cifrario di Bellaso*

1.2 Algoritmi asimmetrici

Gli algoritmi a chiave pubblica o asimmetrici si differenziano dagli algoritmi a chiave simmetrica dalla metotologia di cifratura e decifratura. Nella famiglia degli *simmetrici* un'attore capace di cifrare un messaaggio è anche capace di decifrarlo, ovvero per la comunicazione riservata di un messaggio si utilizza la stessa chiave sia nell'operazione

[1] analizzato nel prossimo capitolo

di cifratura che decifratura; mentre nelle algoritmi a chiave asimmetrica non è garantita l'operazione di decifratura se si è cifrato il messaggio, e vice-versa.

Questo è dovuto alla tecnica di scambio della chiave, che negli algoritmi simmetrici avviene prima dell'invio dell'informazioni mentre negli algoritmi asimmetrici invece si parte dal concetto che il mittente ed il destinatario siano in possesso di una coppia di chiave, una pubblica ed una privata. Sostanzialmente il mittente del messaggio cifrerà il messaggio con la chiave pubblica del destinatario il quale eseguirà la decifratura con la sua chiave segreta.

La robustezza intrinseca degli algoritmi a chieve pubblica, o asimmetrici, è legata alla complessità di fattorizzazione in numeri primi. Risulta essere di estrema facilità il calcolo della funzione conoscendo i fattori primi di partenza ma risulta essere nota l'impossibilità di tornare ai fattori di partenza conoscendo la funzione generata. Ovvero un sistema a chiave asimmetrica deve garantire:

- Sistema di cifratura: $D(E(m)) = m$
- Sistema di firma digitale: $E(D(m)) = m$

avendo posto $D = chiaveprivata, E = chiavepubblicaem = messaaggio.$

Il primo algoritmo che sfrutta il principio della cifratura a chiave pubblica fu progettato nel 1976 da *Diffie & Hellmann.*
Nella famiglia degli algoritmi a chiave pubblica si riportano alcuni dei più famosi:

- **RSA**: scoperta da *Rivest, Shamir, Adlemann.* L'algoritmo rende computazionalmente impossibile il calcolo della fattorizzazione della funzione di cifratura grazie all'utilizzo di numeri primi dell'ordine delle 100 cifre. Ad oggi lo standard RSA garantisce la sua inviolabilità con numeri di grandezza dell'ordine dei 2048 bit.
- **Rabbin**: utilizzato per la prima volta nel 1979, basa le fondamente della sua sicurezza sulla complessità computazionale nel calcolare la radice quadrata in Zn, essendo n = p*q, due primi. É stato dimostrato che il tempo necessario per calcolare il testo in chiaro partendo dal testo cifrato è pari alla scomposizione di n in fattori primi.
- **El-Gamal**: progettato nel 1985 basa tutta la sua sicurezza sull'utilizzo di una funzione unidirezionale a sua volta basata sull'impossibilità di calcolare un logaritmo partirtendo da una funzione discreta. É anche noto per essere un algoritmo stocastico ovvero utilizza una generazione randomica di chiavi.

Chapter 2

Data Encryption Standard

In questo capitolo verrà analizzato nel dettaglio l'algoritmo a chiave simmetrica Data Encryption Standard noto come il DES

2.1 Brevi cenni storici

Nei primi anni del 1970 negli ambienti del ICT si presentava la necessità di utilizzare un algoritmo di cifratura che potesse essere ritenuto robusto e stabile.

Nel 1973 il National Bureau Standard commissionò alla comunità crittografica internazionale la realizzazione di un algoritmo di cifratura che potesse essere usato come standard. Nel 1974 dai laboratori dell'IBM fu rilasciato l'algoritmo *LUCIFER*. Successivamente fu sottoposto all'analisi del NBS che solamente nel 1975 lo rese pubblico. Nel 1976 l'algoritmo fu standardizzato sotto il nome *DES*. Il passaggio tra l'IBM e l'ente NBS fu gradito ad una parte della comunità crittografica dubbiosa del fatto che il NBS avesse potuto inserito delle trapdoor nell'algoritmo. Oltre alla critica su un possibile inserimento di trapdoor da parte del NBS fu criticato dalla comunità crittografica la scelta di utilizzare una chiave di cifratura troppo corta quindi soggetta a possibili attacchi rendendo l'algoritmo troppo fragile.

L'algoritmo, nonostante le critiche sulla sua fragilità, risulta essere un algoritmo fortemente utilizzato nel commercio elettronico grazie alla sua estrema velocità nel cifrare e decifrare il testo e grazie alla sua ragionevole robustezza.qqq

Solamente nel 1998 dopo un ventennio di utilizzo il DES fu sostituito dal *Rijndael* sotto il nome **Advanced Encryption Standard**, che consentiva l'utilizzo di chiavi a 128, 192 e 256 bit.

2.2 L'algoritmo

Il DES fa parte della famiglia degli algoritmi a chiave simmetrica con cifratura a blocchi. Ovvero mittente e destinario utilizzano una chiave condivisa per comunicare in maniera sicura ed il testo in chiaro è suddiviso in blocchi di lunghezza prefissata.

Per il DES i ricercatori dell'IBM scelsero di suddividere i blocchi di testo in chiaro in blocchi di 64 bit sui quali operare le operazioni per cifrare il testo. L'algoritmo di cifratura e decifratura prende il nome di *sistema Feistel*, dal nome di un matematico che prese parte allo sviluppo di *LUCIFER*, algoritmo padre del DES. Il DES è un cifrario binario monoalfabetico a blocchi che opera per trasposizione e sostituzione riducendo al minimo la criticità di tipo statistico e matematico. Ad oggi l'unico attacco valido all'algoritmo è quello della ricerca esaustiva delle possibili chiavi. Quindi la criticità è nella lunghezza della chiave (64 bit).

La chiave L'algoritmo prevede la cifratura del messaggio in chiaro tramite una chiave di *64 bit* di lunghezza fissa scomposta in 9 blocchi da 8 bit ciascuno. Per ogni blocco l'ultimo bit è utilizzato per il controllo di disparità. Il bit di disparità viene settato ad 1 o a 0 per mantere il numero di bit 1 dispari. L'utilizzo del bit di disparità porta ad avere solamente *56 bit* indipendenti per costruire la chiave quindi per il DES si possono creare solamente 2^{56} chiavi differenti.

Il DES prevede la cifratura del testo tramite un meccanismo a round e per ogni round prevede la creazione di una chiave derivata dalla chiave base.
Di seguito è riportato l'algoritmo per procedere al calcolo della chiave di round:

1. *Eliminazione bit parità*: il primo passo è eliminare i bit di parità inseriti precedentemente e permutare i restanti bit secondo una prefissata tabella. In output a tale procedimento si ottiene una sequenza di 56bit (C_0, D_0)
2. *Operazione di shifting*: per $1 \leq i \leq 16$ si considerano $C_i = LS_i(C_{i-1})$ e $D_i = LS_i(D_{i-1})$. Con LS si intende l'operazione di left shifting di 1 o 2 bit in base alla tabella seguente

Table 2.1: Tabella di shifting della chiave di round

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bit di scorrimento	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

3. *Selezione di 48 bit*: dalla sequenza di 56 bit ottenuta precedentemente C_iS_i si individuano i 48 bit in base alla seguente tabella ottenedo così la chiave di round K_i.

TABLE 2.2: Tabella della permutazione della chiave di round

14	17	11	24	1	5	3	28	15	6	21	10
23	29	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	23

Ogni bit della chiave iniziale K viene utilizzato in circa 14 raound su 16, questa caratteristica porta ad avere un buon sistema di cifratura.

L'algoritmo di cifratura L'algoritmo di cifratura del messaggio m di 64bit di lunghezza è composto da tre fasi:

1. *Permutazione Iniziale*: i bit iniziali di m sono permutati con una permutazione fissa, seguendo la tabella successiva (Table 2.3), in modo da ottenere $m_0 = IP(m)$. Vengono indicati con L_0, i primi 32bit del messaggio e con R_0 i successivi 32bit.

TABLE 2.3: Permutazione iniziale

58	50	42	34	26	18	10	2	60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6	64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1	59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5	63	55	47	39	31	23	15	7

2. *16 Round*: la caratteristica del DES è legata al numero di round necessari per produrre in output il testo cifrato. Considerando $1 \leq i \leq 16$, con i numero del round, si effettuano le seguenti operazioni:

   - $L_i = R_{i-1}$
   - $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$

   avendo definito: f come la funzione del DES, analizzata più nel dettaglio nel paragrafo successivo e K_i è la chiave di round di 48 bit generata a partire dalla chiave iniziale K.

3. *Permutazione finale*: eseguiti i 16 round si invertono i L_{16} e R_{16} in modo da ottenere il testo cifrato $c = IP^{-1}(R_{16}L_{16})$.

L'inserimento dela permutazione iniziale sembra essere stato fatto non a fini crittografici per aumentarne la sicurezza ma solamente per migliorare l'inseirmento dei bit nei chip disponibili negli anni '70

L'agoritmo di decifratura É possibile decifrare il messaggio $c = L_{16}R_{16}$, in output all'algoritmo di cifratura, eseguendo gli stessi procedimenti per generarlo ma considerando le 16 chiavi $K_1, .., K_{16}$ nell'ordine inverso.

La funzione $f(R, K_i)$ La funzione f, detta *fuzione Feistel*, è composta da una funzione di espansione $E(R)$ che trasposforma l'input R di 32bit in un output di 48bit secondo la tabella seguente. Quindi si calcola $E(R) \oplus K_i$ che produce un output di 48bit.

TABLE 2.4: Permutazione di espansione

32	1	2	3	4	5	4	5	6	7	8	9
8	9	10	11	12	13	12	13	14	15	16	17
16	17	18	19	20	21	20	21	22	23	24	25
24	25	26	27	28	29	28	29	30	31	32	1

Successivamente il messaaggio entra in 8 S-box che producono 8 output $C_1...C_8$ di 4 bit ognuno. In fine la stringa in output viene permutata secondo una specifica tabella.

Gli S-box Sono il cuore della funzione f del DES e permettono di ottenere la stringa cifrata. Ogni S-box è composto da una matrice 4x16 che prende in input la stringa di 6 bit uscente dall'operazione $E(R) \oplus K_i$. La stringa ottenuta Bj=b1b2...b6 consente di individuare la riga e la colonna dell'S-box$_j$ da selezionare. I bit b_1b_6 individuano la riga mentre i restanti bit $b_2b_3b_4b_5$ individuano la colonna dell'S-box$_j$, viene così individuato il valore da selezionare dall'S-box. Fino alla metà degli anni '90 il procedimento e gli S-box utilizzati dall'algoritmo furono mantenuti segreti.

TABLE 2.5: Esempio di un S$_i$-box

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Le caratteristiche degli S-box sono legate al fatto che nel 1974 erano disponibili chip dalla computazione e dalla memoria limitata. Gli S-box del DES sono:

1. prendono in input 6 bit e produco in output 4 bit
2. l'output di ogni S-box non doveva assumere una qualsiasi similitudine ad una funzione lineare degli input. Se fosse accaduto ciò si sarebbe venuta a creare una situazione che avrebbe reso l'algoritmo vulnerabile
3. Ogni riga di ogni S-box contiene esattamente i numeri da 0 a 15

4. dati due input ad un S-box che differiscono di un solo bit l'output prodotto deve differire di almeno di 2 bit

5. se due input di un S-box hanno i primi 2 bit differenti ma gli ultimi due identici i due output devono essere differenti

6. date le 32 coppie in ingresso con le loro relative XOR si calcolcano le XOR dei relativi output e non più di 8 XOR in output possono essere uguali.

L'ultimo punto è chiaramente una tecnica di prevenzione rispetto ad un'attacco mediante *Crittanalisi differenziale*.

Il DES si può considerare un buon sistema di cifratura perchè grazie alla funzione di espansione $E(R)$ in pochi round ogni bit del testo cifrato dipende da ogni bit del testo in chiaro. Infatti in un sistema di cifratura ottimale ogni bit di testo cifrato dovrebbero dipendere da ogni bit del testo in chiaro. Ciò è ottenuto nel DES grazie all'utilizzo della funzione di espansione $E(R)$.

2.3 Attacchi all'algoritmo

LA parte di crittanalisi degli algoritmi di cifratura si basa sull'assunto che l'algoritmo di cifratura/decifratura sia noto al "nemico". Questo assunto è un buon punto di partenza in ragione del fatto che in letteratura si sono verificati episodi di algoritmi di cifratura/decifratura violati non appena divenuti di dominio pubblico.

Questa ipotesi iniziale è avallata anche dal principio di *Kerckhoff*, secondo cui il nemico è a conoscenza dell'intero sistema di cifratura/decifratura. Secondo questo assunto un sistema crittoanalitico è sicuro quanto è più complessa l'individuazione della chiave di cifratura/decifratura.

2.3.1 Tipologia di attacchi al sistema

- **Testo cifrato**: è la tipologia di attacco che nel passato veniva utilizzata per decifrare testi cifrati scritti su pergamena.

  L'attaccante ha a disposizione una collezione di testi cifrati. Si dice che l'attacco ha pieno successo se l'attaccante riesce a recuperare il corrispondente testo in chiaro o meglio se riesce a dedurre la chiave di crifratura. Sono considerati risultati positivi anche il recupero di parte dell'informazione cifrata.

- **Testo in chiaro noto**: l'attaccante ha a disposizione sia del testo cifrato si del testo in chiaro ciò permette di dedurre la chiave di cifratura.

  Durante la seconda guerra mondiale questa tipologia di attacco fu utilizzata dagli alleati per decifrare testi riservati dei nazisti. Gli alleati avevano testi in chiaro noti corti detti *crib*, che si riferivano a informazioni generali sul campo di battaglia. Attraverso queste breve sequenze note di testo in chiaro potevano cercare di arrivare ad individuare la chiave di cifratura.

- **Testo in chiaro scelto**: l'attaccante riesce a produrre del testo in chiaro da far decifrare per ottenere un testo cifrato con l'obiettivo di ottenere quante più informazioni sul testo cifrato da quale dedurre quante più informazioni sugli schemi di cifratura. Nel peggiore dei casi queste informazioni conducono a identificare la chiave di cifratura.

  Nella seconda guerra mondiale gli alleati inducevano il nemico a reinviare messaggi relativi ad azioni note, come per esempio sulla bonifica di campi minati. Questa tecnica detta *gardering* portò alla conoscenza di maggiori informazioni riguardo agli schemi di cifratura utilizzati con *Enigma*[1]

- **Chiavi correlate**: in questo attacco è a disposizione dell'attaccante la possibilità di verificare le risposte dell'algoritmo con l'utilizzo di svariate chiavi di cifrazione inizialmente ignote, ma di cui conosce sostanzialmente le proprietà per la loro creazione.

  Un esempio di utilizzo di questa tecnica di attacco si ha nella possibilità di violare reti WIFI protette con l'algoritmo *WEP*. Questo utilizza l'algoritmo del RC4, famoso algoritmo a flusso nel quale la chiave non deve essere usata una sola volta, per generare la chiave di cifratura. La chiave del WEP è una concatenazione tra la chiave inserita manualmente dall'utente, per questo si presuppone che venga sostituita di rado, con una seconda per non violare il RC4. La seconda parte del chiave è il *Vettore di Inizializzazione* di 24bit. Utilizzando il paradosso del compleanno ci si attende che ogni 4096 pacchetti due di essi condividono lo stesso *VI* e quindi lo stesso RC4. Per ovviare a questo bug di sicurezza si è passati a proteggere le reti WIFI con l'algoritmo WPA.

[1] Enigma è il nome in codice della macchina di cifratura utilizzata dai nazisti durante la seconda guerra mondiale per produrre ipotetici testi cifrati. Questa tecnica di cifratura fu violata da Rejeweski, Zygalski, Rozycki intorno al 1939 e inviaro le informazioni agli Inglesi per contrastare la potenza dei Tedeschi.

2.3.2 Crittanalisi Differenziale

Nel 1990 Biham e Shamir presentarono alla comunità crittografica un nuovo concetto di crittoanalisi chiamata **Crittanalisi Differenziale**. Questa tecnica prevede di confrontare due testi cifrati e calcolarne le differenze partendo da una coppia di testi in chiaro opportunamente scelti e ricavarne informazioni relative alla chiave di cifratura utilizzata. É intuitivo considerare che la differenza tra due sequenze di bit si possa eseguire attraverso l'operazione di XOR, quindi avendo inserito nell'algoritmo la chiave tramite operazione di XOR su $E(R_{i-1}$ eseguendo nuovamente l'operazione di XOR su due sequenze in input si riesce ad eliminare la casualità inserita nel sistema dalla chiave.

Esempio dell'analisi differenziale Si esegue l'analisi differenziale su un sistema DES a 4 round.
Si ipotizza, sotto l'ipotesi di Kerckhoff, che l'attaccante abbiamo piena conoscenza del sistema, ovvero conosca l'input e il corrispettivo output prodotto ed inoltre ha accesso agli S-box del sistema, di seguito riportati. L'attaccante vuole solamente ottenere informazioni sulla chiave utilizzata dall'algoritmo.

TABLE 2.6: S_1-box

010	010	001	110	011	100	111	000
001	100	110	010	000	111	101	011

TABLE 2.7: S_2-box

100	000	110	101	111	001	011	010
101	011	000	111	110	010	001	100

Con un attenta analisi probabilistica su i due S-box, ci si accorge che prese le coppie di input $f(R_0, K_1)$ e $f(R_0^*, K_1)$ con

$$f(R_0, K_1) \oplus f(R_0^*, K_1) = 0011$$

al S_1-box ben 12 producono lo stesso output 011. Questo discostamento notevole rispetto al valore atteso di soli due output con valore uguale ci aiuta nel calcolo della chiave. Anche l'S_2-box ha un punto debole. Infatti considerando tra le 16 coppie in input le coppie con

$$f(R_0, K_1) \oplus f(R_0^*, K_1) = 1100$$

ben 8 producono un output 010.

Ora considerando:

$$R_0' = R_0 \oplus R_0^* = 001100$$

poichè la funzione di espansione genera un output:

$$E(R') = 00111100$$

quindi abbiamo che la somma XOR in input ai due S-box verifica le nostre premesse:

$$input(S_1) = 0011 \tag{2.1}$$

$$input(S_2) = 1100 \tag{2.2}$$

Ipotizzando di poter considerare i due input indipendenti si ottiene:

$$\left.\begin{array}{l} P(2.1) = \frac{12}{16} \\ P(2.2) = \frac{8}{16} \end{array}\right\} \rightarrow P(2.1) * P(2.2) = \frac{12}{16} * \frac{8}{16}$$

con P ad indicare la probabilità di ottenere in uscita l'output. Purtroppo i due input non sono indipendenti avendo utilizzato la funzione di espansione che porta i bit 3 e 4 sia nel S_1-box che nel S_2-box. Quindi