Crittografia simmetrica e algoritmo DES

Con analisi degli algoritmi

Samuele Pozzani

Contents

1 Introduzione

Con **crittografia simmetrica**, o **crittografia a chiave privata**, si intende una tecnica di **cifratura**. Rappresenta un metodo semplice per cifrare testo in chiaro dove la chiave di crittazione è la stessa chiave di decrittazione, rendendo l'algoritmo molto performante e semplice da implementare. Tuttavia presuppone che le due parti siano già in possesso delle chiavi, richiesta che non rende possibile uno scambio di chiavi con questo genere di algoritmi. Lo scambio avviene attraverso algoritmi a chiave asimmetrica o pubblica, generalmente più complessi sia da implementare che da eseguire ma permettono questo scambio in modo sicuro. Dopodiché la comunicazione verrà crittata usando solo algoritmi a chiave simmetrica per garantire una comunicazione sicura ma veloce.

2 Funzionamento

In questo genere di algoritmi si suppone che entrambe le parti conoscano già la chiave con cui crittare e decrittare il messaggio. Il mittente ha un messaggio **P** (PlainText o testo in chiaro). Il mittente critta il messaggio **P** con la chiave **k** usando un algoritmo di crittografia simmetrica chiamato **S**. Il messaggio risultante sarà **C** (CypherText o messaggio cifrato). In formule diventa:

$$S(P, k) = C$$

A questo punto al destinatario arriva un messaggio cifrato che riesce a decrittare poiché è in possesso della chiave privata. Ora il ricevente applica l'algoritmo di decrittazione **D** con la stessa chiave che ha usato il mittente per crittare il testo. Diventa:

$$D(C, k) = P$$

Se un attaccante ha intercettato il messaggio lungo il mezzo di comunicazione, avrà il messaggio crittato ma non la chiave che è stata scambiata in modo sicuro dai due interlocutori. Se l'attaccante vorrà leggere il messaggio crittato potrà solo usare metodi di decrittazione che richiedono elevate capacità di calcolo.

Nel caso di una comunicazione reale, questo colloquio viene criptato tramite un algoritmo a chiave pubblica, più complesso ma che non richiede nessuna trasmissione della chiave sul mezzo di comunicazione.

3 Componenti comuni nelle varie implementazioni

Tra i vari algoritmi di crittazione possiamo trovare alcune operazioni comuni, poiché aggiungono generalmente maggior sicurezza nel testo cifrato e sono operazioni rapide per la macchina.

Spesso una stessa operazione viene ciclata più volte, riferendosi a questi passaggi come **cicli** o **round**. Ad esempio in AES la stessa routine viene ripetuta 10 volte. In DES il testo in chiaro subisce 16 volte la crittazione insieme alla chiave prima di terminare. Una volta disegnato l'algoritmo viene molto facile ripeterlo, rendendo più complesso un lavoro di decrittazione forzata tramite brute force. Se l'algoritmo di decrittazione è ben disegnato e non si riescono ad avere informazioni sulla chiave, questo è l'unico metodo attraverso cui è possibile la decrittazione del messaggio cifrato.

Tra i vari algoritmi simmetrici possiamo riconoscere alcuni parametri standard come la **lunghezza della chiave** e la **dimensione del blocco**.

La lunghezza della chiave è misurata in bit e ha valori che oscillano tra 32 bit e 512 bit. Generalmente la lunghezza della chiave è un valore fisso nonostante esistano alcuni algoritmi come AES che impiegano lunghezze variabili.

Ogni algoritmo generalmente cerca di crittare una stringa di bit attraverso una chiave in un'altra stringa di bit della medesima lunghezza. La lunghezza di questa stringa è pari alla dimensione del blocco. Algoritmi più datati avevano questo valore pari a 64bit in media. Oggi si preferisce adottare dimensioni di 128 bit.

Un problema che affligge la dimensione del blocco è il paradosso del compleanno che rilascia informazioni sulla chiave ogni volta che avviene una collisione. Possiamo ritenere sicura solo la radice quadrata di tutte le combinazioni possibili. Per esempio con una dimensione di 64 bit, che genererebbe $\mathbf{2^{64}}$ possibili combinazioni, potremo impiegarne solo $\mathbf{2^{32}}$ prima di cominciare a rivelare informazioni sulla chiave.

4 Metodi di Crittazione a Blocchi di Cifre

Generalmente la dimensione del blocco scelta è della medesima lunghezza della chiave perché risulta semplice per l'implementazione di un algoritmo. Tuttavia è bene fare attenzione ad alcuni metodi che possono compromettere la sicurezza dell'algoritmo. Nei seguenti algoritmi individuiamo

$\mathbf{k_i}$ è l'i-esima cifra della chiave
$\mathbf{P_i}$ è l'i-esima cifra del testo in chiaro
$\mathbf{C_i}$ è l'i-esima cifra del testo cifrato

con

$\mathbf{i} = 1...n$
n = dimensione del blocco e lunghezza della chiave

4.1 Electronic Code Book (ECB)

$$S(P_i, k_i) = C_i$$

È l'implementazione più semplice, in cui l'unica cosa che nasconde il testo in chiaro è una cifra della chiave. Questo metodo risulta essere tanto semplice quanto insicuro. Infatti è sufficiente per l'attaccante raccogliere un numero sufficiente di campioni per scoprire la chiave. Su questo metodo si basa il Cifrario di Cesare.

4.2 Cipher Block Chaining (CBC)

$$S((IV \oplus P_1), K_1) = C_1$$
$$S((C_i - 1 \oplus P_i), K_i) = C_i$$

In questo metodo si aggiunge un fattore di casualità inserendo nell'algoritmo anche la cifra precedentemente crittata; più precisamente, si effettua uno XOR prima di crittare il testo. In questo modo non vi è una associazione univoca tra chiave e testo in chiaro ma si aggiunge la dipendenza dalla cifra precedente. Inserendo la dipendenza dalla cifra precedente, si crea la necessità di aggiungere un elemento per crittare la prima cifra del blocco, chiamata Vettore di Inizializzazione (IV nelle formule).

4.2.1 Cipher Feed-Back (CFB)

$$S(IV, k_1) \oplus P_1 = C_1$$
$$S(C_i - 1, k_i) \oplus P_i = C_i$$

Molto simile al CBC ma l'operazione di XOR con il testo in chiaro viene eseguita dopo la crittazione. Si critta prima la chiave con la cifra precedente o il Vettore di Inizializzazione nel caso della prima cifra. RIspetto a CBC è sempre presente la dipendenza dalla cifra precedente, ma soffre ancora del Problema di Malleabilità, anche se solo localmente alla singole cifra.

5 Algoritmi

5.1 DES (Data Encryption Standard)

Impiega una chiave di 56 bit e opera su blocchi di 64 bit. DES è uno degli algoritmi a chiave simmetrica più famoso, pubblicato nel 1976 da IBM e scelto come standard per la Federal Information Processing Standard. È diventato in seguito lo standard fino a quando non fu decrittato nel 1997 in 3 giorni di calcolo. Nell'anno successivo fu sufficiente un giorno soltanto impiegando un cluster di computer e con l'avanzare i tempi si riducono ulteriormente. Il suo successore fu 3DES.

5.2 3DES (Triple DES)

Quando DES non fu più sicuro, si cercò un metodo che mantenesse le meccaniche del DES ma che permettesse di avere una chiave più lunga. In questo algoritmo si esegue una tripla crittazione impiegando 3 chiavi DES standard, a 56 bit, ottenendo una chiave a 168 bit. È possibile anche invertire il secondo passaggio, ovvero eseguire una crittazione e una decrittazione. Tuttavia non modifica la sicurezza generale dell'algoritmo.

Anche questo algoritmo oggi non viene più impiegato poiché le tecnologie si stanno evolvendo e molti algoritmi di crittazione non risultano abbastanza forti da sopportare le elevate capacità di calcolo dei computer moderni, soprattutto con l'avvento delle GPGPU. 3DES ha lasciato il posto a AES, il nuovo standard ormai.

5.3 AES (Advanced Encryption Standard)

Nel 1999 si presentarono vari algoritmi candidati a diventare lo standard di crittografia simmetrica. Questi candidati furono MARS proposto dalla IBM, RC6, Serpent, Twofish e Rijndael. Tutti questi algoritmi furono testati per efficienza e sicurezza su varie architetture, sia hardware che software. Tra questi ricevette un feedback positivo Rijndael che nel 2000 divenne il nuovo standard con il nome di AES. Fu dapprima impiegato dal governo degli USA e dopodiché il suo successo divenne globale.

AES lavora su blocchi a dimensione fissa di 128 bit. Ha una chiave di 128 bit ma possono essere impiegate chiavi più lunghe da 192 e 256 bit per crittare documenti di particolare importanza.

Fonte: it.wikipedia.org

6 Algoritmo DES

In crittografia il Data Encryption Standard (DES) è un algoritmo di cifratura scelto come standard dal Federal Information Processing Standard (FIPS) per il governo degli Stati Uniti d'America nel 1976 e in seguito diventato di utilizzo internazionale. Si basa su un algoritmo a chiave simmetrica con chiave a 64 bit (ma solo 56 utili poiché 8 sono di controllo).

6.1 Come funziona l'algoritmo

L'algoritmo di crittazione DES elabora il messaggio dividendolo in blocchi da 64 bit permutandoli in 2^{64} possibili combinazioni. Inizialmente il blocco di 64 bit viene diviso in due parti da 32 bit L e R. Anche la chiave di crittografia deve essere di 64 bit.
Esempio: Il messaggio M è 0123456789ABCDEF in esadecimale, quindi in formato binario:
M = 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
L = 0000 0001 0010 0011 0100 0101 0110 0111
R = 1000 1001 1010 1011 1100 1101 1110 1111

6.1.1 Passo 1: Creazione di 16 sottochiavi

Per prima cosa la chiave K è permutata attraverso una tabella detta PC-1. PC-1 contiene solamente 56 numeri, infatti la chiave permutata conterrà 56 bit poichè l'ottavo bit di ogni byte viene usato come controllo di parità. Se il primo numero di PC-1 è 57 significa che il primo bit dalla chiave permutata K+ è il 57-esimo bit della chiave K.
Successivamente K+ viene divisa in due blocchi da 28 bit: C_0 e D_0.

A questo punto, in accordo con una tabella di iterazione, C_0 e D_0 subiscono 16 rotazioni verso sinistra fino ad arrivare a C_{16} e D_{16} che sono uguali a C_0 e D_0. Vengono poi concatenate le chiavi C_n e D_n ottenendo così 16 chiavi da 56 bit.
Ognuna di queste chiavi viene poi permutata attraverso una nuova tabella PC-2 (che contiene solamente 48 bit) nella stessa modalità della precedente permutazione. Da ciò si ottengono le 16 sottochiavi K_n da 48 bit ciascuna.

6.1.2 Passo 2: Codifica dei blocchi del messaggio

Inizialmente il messaggio M viene permutato attraverso la tabella IP e poi viene diviso in L_0 e R_0, entrambi da 32 bit.
Ora si procede attraverso 16 iterazioni usando la funzione f che opera su due blocchi del messaggio da 32 bit e una sottochiave da 48 bit.

Quindi, con 1 <= n <= 16:

$$L_n = R_(n-1)$$
$$R_n = L_(n-1) \oplus f(R_(n-1), K_n)$$

La funzione f prevede che venga inizialmente eseguito uno XOR tra il blocco del messaggio e la sottochiave; per fare ciò il blocco del messaggio deve essere permutato in modo da raggiungere i 48 bit della sottochiave grazie alla tabella E. A questo punto è possibile calcolare lo XOR bit a bit.
La funzione f non termina qui: sono ora necessarie le 8 S-tables per far tornare la lunghezza del blocco a 32 bit. Il risultato dello XOR si può dividere in 8 blocchi da 6 bit ciascuno; in ogni blocco viene preso il primo e l'ultimo bit che, uniti, indicano il numero della riga (da 0 a 3); poi vengono presi i 4 bit centrali che indicano il numero della colonna (da 0 a 15). Grazie a queste coordinate è possibile trovare un numero (di 4 bit) all'interno delle S-tables che andrà a sostituire il blocco di partenza. Una volta fatto ciò per tutti gli 8 blocchi si ottiene una stringa da 32 bit. A questo punto l'ultima stringa da 32 bit viene permutata attraverso la tabella P e R_n diventa la XOR tra L_{n-1} e il risultato della funzione f.

L'ultimo passo sta nel concatenare R_{16} con L_{16} e permutarlo attraverso l'ultima tabella IP^{-1}. Per decodificare il messaggio è necessario seguire la stessa procedura inverttendo l'ordine in cui le sottochiavi vengo applicate.