Cifrari Simmetrici

Return

Indice

Introduzione

La crittografia simmetrica studia i **Cifrari a chiave segreta**, questi vengono usati per proteggere la riservatezza di documenti, generazione di numeri pseudo casuali, autentificazione e identificazione.

I cifrari hanno quattro caratteristiche:

- **Robustezza**: resitere agli attacchi di crittanalisi
- **Velocità**: nelle realizzazioni con hardware specializzato elaborano diversi milioni di bit al secondo
- **Efficacia**: gestiscono stringhe binarie di lunghezza variabile
- **Efficienza**: il testo crifrato è lungo quanto il testo in chiaro

> Funzionamento di un cifrario simmetrico:

- Mittente A e destinatario B condividono una chiave segreta K_{AB}.
- A deve inviare un messaggio m a B, lo cifra con la chiave K_{AB} ottenendo il testo cifrato $c=E_{K_{AB}}(m)$ e lo invia a B.
- B riceve il messaggio c e lo decifra con la chiave K_{AB} ottenendo il messaggio originale $m=D_{K_{AB}}(c)$.

In generale m è una parte del testo in chiaro M che A vorrà inviare a B. Proprio sulla natura di m si basa la distinzione che viene effettuata sugli algoritmi simmetrici:

- **Cifrari a flusso**: sono basati su **One-Time Pad** e cifrano un bit alla volta. La chiave è lunga quanto il messaggio e viene usata una sola volta.
- **Cifrari a blocchi**: cifrano un blocco di n bit alla volta, ispirati al **cifrario poligrafico** e **cifrario composto**.

Cifrari a Flusso

Utilizzano una trasformazione variabile al progredire del testo ottenuta tramite funzioni **XOR**.

- **cifratura**: $c_i = m_i \oplus k_i$
- **decifratura**: $m_i = c_i \oplus k_i$

La chiave è una sequenza pseudocasuale di bit lunga quanti il testo da cifrare.

Sia in trasmissione che in ricezione si utilizzano dei **generatori di flusso di chiave** (i prng) per garantire la sincronizzazione tra i due flussi.

Le chiavi in questo modo sono periodiche, quindi per garantire sicurezza si deve avere un periodo più lungo e ogni volta si utilizza un seed diverso per generare le chiavi.

Ad oggi esistono due tipi di cifrario a flusso:

- a flusso sincrono
- con auto-sincronizzazione

Nei primi il flusso di bit di chiave viene generato in modo indipendente dal flusso dei bit di testo. Nel momento in cui sorgente e destinazione si disallineano, devono fare ripartire i generatori di chiave e scegliere un diverso punto di inizio della sequenza.

Nei cifrari con autosincronizzazione, il flusso dei bit dipende dal flusso dei bit di testo crifrato. La causa più comune di diseallinamento è la perdità di integrità del testo cifrato che può essere causata da eventi casuali o intenzionali.

Cifrari a Blocchi

Nei cifrari a blocchi il testo in chiaro viene diviso in blocchi di lunghezza fissa (Aggiungendo padding se necessario) e ogni blocco viene cifrato separatamente. La lunghezza della chiave è la determinante della robustezza del cifrario.

Rete di Feistel

La rete di Feistel è una architettura per cifrari a blocchi che utilizza una funzione F non lineare per cifrare i blocchi.

Si parte dal testo in chiaro m di lunghezza w e lo si divide in due parti L_i e R_i di lunghezza $w/2$ bit. La rete quindi procede in n passaggi con la seguente formula: $$ L_{i+1} = R_i$$ $$ R_{i+1} = L_i \oplus F(R_i, K_i) $$ Dove K_i è una sottochiave generata dalla chiave principale K con una apposita funzione.

La rete genera quindi **confusione** grazie alle sostituzioni operate da F e **diffusione** grazie alla permutazione dei bit tra L_i e R_i.

DES - Data Encryption Standard

DES è un cifrario a blocchi che utilizza una rete di Feistel che prevede 16 iterazioni con chiave di 56 bit e blocchi di 64 bit. La funzione F è composta da una espansione e permutazione che porta il vettore da 32 a 48 bit, una somma modulo 2 con 48 bit di sottochiave, una sostituzione che riporsta il vettore a 32 bit e una permutazione senza chiave finale.

Provando ad attaccare il cifrario, si sono scoperte due nuove tecniche di crittanalisi:

- **Attacco differenziale**: si basa sull'analisi delle differenze tra le coppie di testo in chiaro e il loro corrispondente testo cifrato. Si cerca di trovare una relazione tra le differenze.
- **Attacco lineare**: si basa sull'analisi delle relazioni lineari tra le coppie di testo in chiaro e il loro corrispondente testo cifrato. Si cerca di trovare una relazione tra i bit del testo in chiaro e quelli del testo cifrato.

Triple DES

Il Triple DES consiste nell'esecuzione di tre volte l'agoritmo DES. Ci sono due varianti:

- **EEE** dove vengono effettuate tre cifrature con 3 sottochiavi diverse
- **EDE** dove viene effettuata una cifratura, una decifratura e una cifratura con 2 sottochiavi diverse.

AES - Advanced Encryption Standard

AES è un cifrario che utilizza chiavi e blocchi di 128 bit, espandibili per multipli di 32 bit. Le operazioni utilizzate sono somme a modulo 2 e scorrimenti. Segue uno schema semplice e lineare detto **square**. Ogni round è composto da 4 operazioni:

- **sostituzione** di byte: ogni byte del blocco viene sostituito con un altro byte secondo una tabella di sostituzione
- **permutazione** shift rows: i byte del blocco vengono spostati in modo circolare
- **operazioni aprimetiche su $GF(2^8)$**: i byte del blocco vengono sommati con una chiave di round (mix columns)
- **operazioni di somma modulo due** fra i dati in ingresso e la chiave di round.

Modalità di cifratura

ECB - Electronic Code Book

Avviene un solo bit alla volta, non propaga errori, ma non è sicuro in quanto ogni blocco utilizza la stessa chiave. Se due blocchi sono uguali, il testo cifrato sarà uguale.

CBC - Cipher Block Chaining

Ogni blocco del testo viene messo in XOR con il blocco precedente cifrato. La chiave è la stessa per tutti i blocchi, ma il testo cifrato cambia in base al blocco precedente. Questo metodo propaga gli errori, quindi se un blocco viene alterato, anche i successivi saranno alterati.

Per impedire all'attaccante di alterare il testo cifrato, si utilizza un **IV** (Initialization Vector) che viene cifrato con la chiave e viene messo in XOR con il primo blocco del testo in chiaro. L'IV deve essere casuale e unico per ogni sessione di cifratura.

CFB - Cipher Feedback

Converte idealmente una cifratura a blocchi in una a flusso. L'input delle funzione di cifratura è dato da un registro a scorrimento di 64 bit che contiene il Vettore di inizializzazione. I primi n bit significativi dell'output vengono messi in XOR con i primi n bit del testo da cifrare, gli n bit successivi vengono accolti dal registro a scorrimento e così via.

OFB - Output Feedback

Uguale alla precedente, ma il registro a scorrimento viene alimentato con l'output della funzione di cifratura invece che con il testo cifrato.

Implementa un cifrario a flusso sincrono, mentre CFB realizza un cifrario a flusso autosincronizzante.

CTR - Counter

utilizza un contatore della stessa dimensione del blocco di testo su cui operare. Ad ogni blocco deve corrispondere un valore diverso del contatore che viene incrementato ad ogni giro. Il suo valore viene cifrato con la chiave e messo in XOR con il blocco di testo in chiaro.

Generazione di bit e numeri pseudo-casuali

CFB, OFB e CTR possono essere utilizzate per generare sequenze di bit casuali ovvero come PRBG (Pseudo Random Bit Generator). Mentre ECB e CBC come PRNG (Pseudo Random Number Generator).

Integrità e origine di un messaggio

Per garantire l'integrità di un messaggio e avere conferma dell'autore utilizzando la crittografia simmetrica si cifra il messaggio con la chiave condivisa. Se il destinatario riuscirà a decifrare il messaggio, avrà la certezza che il messaggio è stato cifrato da chi possiede la chiave condivisa. Un metodo alternativo consiste nel fare l'hash con la chiave del messaggio usando la chiave condivisa e concatenarlo al messaggio originale. Se il destinatario riesce a ricostruirire lo stesso hash con la chiave condivisa, avrà la certezza che il messaggio è integro e proviene da chi possiede la chiave condivisa. questo metodo è detto HMAC (Hashed Message Authentication Code). Sia nel priimo che nel secondo metodo possono esserci delle possibilità di ripudio e falsificazione.

Integrità, origine e non ripudio.

Una firma digitale deve possedere 5 requisiti:

- consentire a chiunque di identificare univocamente il filrmatario
- non imitabile
- non trasportabile
- non ripudiabile
- rendere inalterabile il documento firmato

Garantisce quindi integrità, origine e non ripudio. L'unico modo per garantire questi tre requisiti con chiave simmetriche è quello di ricorrere a una terza parte fidata.

1. A invia al **registro atti privati (RAP)** il messaggio $A||E_{K_A}(A||M)$
2. RAP invia a A una ricevuta $E_{K_A}(A||T||M||E_R(A||T||M))$, cifrando con un numero RNG la concatenazione fra l'identificazione di A, un timestamp e il messaggio cifrato. Il RAP nel frattempo salva la voce $A||T||M||R||Firma$
3. A invia a B il documento e la ricevuta $A||T||M||E_R(A||T||M)$
4. B interroga RAP e verifica l'autenticità del messaggio ottenendo come risposta $B||A||T||M||E_R(A||T||M)$
5. RAP comunica a B l'esito della verifica del messaggio $E_{K_B}(A||T||M)$

Risolti i problemi di autenticità, non ripudio e non falsificazione. introducendo:

- necessità di avere l'autorità sempre online
- l'autorità non deve costituire un collo di bottiglia
- l'autorità non deve creare documenti falsi
- l'autorità deve memorizzare le chiavi in una memoria sicura

Gestione delle chiavi

Autorità per la distribuzione delle chiavi

Un centro di distribuzione delle chiavi è un ente fidato che fa da intermediario fra coloro che vogliono comunicare. L'obiettivo è quello di trovare una soluzione scalabile per la distribuzione delle chiavi poichè per n utenti servirebbero n^2 scambi di chiavi.

Prima di vedere come A e B possano comunicare è necessario assumere che A e B abbiano prima effettuato uno scambio di chiavi tra A e T e tra B e T. Queste chiavi sono dette **chiavi master** vengono utilizzate per comunicare le chiavi di sessione in modo sicuro.

Le fasi sono quindi:

1. A chiede a T una chiave per comunicare con B, manda un messaggio del tipo $R_a||A||B$ con R_a numero pseudo-casuale valido una sola volta (nonce).
2. T reinvia ad A il nonce, l'identificativo di B, la chiave generata con un PRNG (chiave master), e un messaggio cifrato con K_B che A deve mandare a B per informarlo della chiave. il messaggio è quindi $E_{K_A}(||R_A||B||K||E_{K_B}(A||K))$. inviando ad A il messaggio da inoltrare a B riduce le interrogazioni al centro di distribuzione delle chiavi.
3. A riceve il messaggio, lo decifra usando la chiave K_A e invia a B il messaggio $E_{K_B}(A||K)$ cifrato con la chiave master di B.
4. B riceve il messaggio, lo decifra usando la chiave K_B e ottiene la chiave di sessione K. Per accertarsi che il messaggio provenga da A, B invia ad A un messaggio cifrato con la chiave di sessione $E_K(B||R_B)$ dove R_B è un nonce generato da B.
5. A riceve il messaggio e per provare a B la sua identità invia a B il nonce - 1 $E_K(R_B-1)$ cifrato con la chiave di sessione.

Il centro di distribuzione delle chiavi deve comunque attribuire un tempo di vita limitato ad ogni chiave di sessione per garantire maggiore sicurezza.

Protocollo Diffie-Hellman

Con Diffie-Hellman viene meno il problema di doversi accordare su una chiave- Lo scambio di chiavi si basa sul **calcolo del logaritmo discreto**.

Le fasi sono:

1. A e B decidono un numero primo p e un generatore g (generatore = numero le cui potenze modulo p sono comprese fra 1 e $p-1$).
2. A sceglie a caso un numero X_A. B sceglie a caso un numero X_B. Questi numeri devono restare segreti e devono essere compresi fra 1 e $p-1$.
3. A calcola $Y_A=g^{X_A} \mod p$ e lo invia a B.
4. B calcola $Y_B=g^{X_B} \mod p$ e lo invia a A.
5. A calcola $K_A=Y_B^{X_A} \mod p$ e B calcola $K_B=Y_A^{X_B} \mod p$. A questo punto A e B hanno la chiave di sessione $K_A=K_B=K$ dunque K è la chiave.

Un intrusore che conosce p e g (perchè sono pubblici) e che riesce ad intercettare Y_A e Y_B non riesce a calcolare la chiave di sessione in quanto il calcolo del logaritmo discreto è computazionalmente difficile.

Il problema di Diffie-Hellman rimane l'identificazione di A e B.