Università degli Studi di Verona

Sicurezza delle reti

Riassunto dei principali argomenti

Davide Bianchi

November 15, 2018

Contents

1 Introduzione

Definizione 1.0.1 (Information Security) *Protezione delle informazioni e dei sistemi per impedirne l'accesso non autorizzato, uso, divulgazione, modifica o distruzione.*

Definizione 1.0.2 (Network Security) *Protezione dell'accesso a risorse situate all'interno di una rete.*

Nella sicurezza si distinguono una **policy**, un **meccanismo** e una **compliance**. Una security policy specifica il comportamento che il sistema può o non può assumere. I meccanismi di sicurezza sono l'implementazione di una data policy. Diciamo quindi che una security policy ϕ deve rimanere valida per un sistema P in ogni ambiente malevolo E, ovvero $P \parallel E \models \phi$.

Le politiche di sicurezza sono spesso formulate per arrivare ad alcune proprietà standard, le più comuni sono:

- Confidenzialità: non ci sono fughe di informazioni;
- Integrità: non ci sono modifiche alle informazioni;
- Disponibilità: non ci sono "danneggiamenti" ai servizi;
- Accountability[1]: le azioni sono sempre riconducibili ai diretti responsabili;
- Autenticazione: l'origine dei dati può essere identificata con sicurezza.

Contromisure per la protezione. Le principali tecniche di contromisura consistono in:

- Prevenzione di breach;
- Rilevamento di attacchi in corso;
- Reazione ad un possibile attacco.

2 Cenni di crittografia

2.1 Introduzione

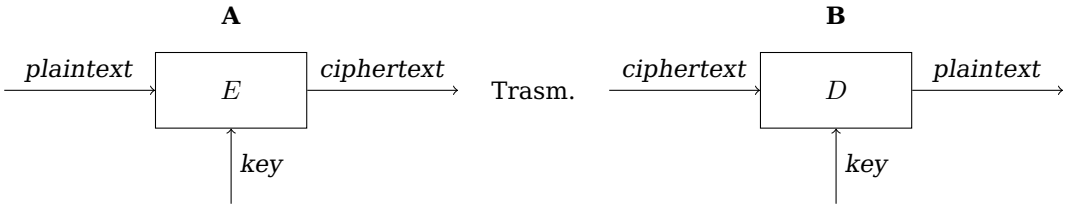
Iniziamo dando alcune definizioni fondamentali. Si useranno i termini *ciphertext* e *plaintext* per indicare rispettivamente il testo cifrato e quello in chiaro.

Definizione 2.1.1 (Crittografia) *Insieme dei metodi per rendere un messaggio non leggibile ad altri.*

Definizione 2.1.2 (Steganografia) *Insieme dei metodi per nascondere l'esistenza di un messaggio in un altro contenuto.*

Definizione 2.1.3 (Crittoanalisi) *Analisi del ciphertext per ottenere il plaintext corrispondente.*

Un generico sistema crittografico è strutturato come:



[1] La traduzione più vicina è *responsabilità*.

In crittografia si distinguono le due categorie *a chiave simmetrica* e *a chiave asimmetrica*. La differenza sta nel fatto che nella crittografia a chiave simmetrica le due entità che si scambiano il messaggio devono condividere una stessa chiave (che deve essere trasmessa su un canale sicuro), mentre nella crittografia a chiave asimmetrica le chiavi sono differenti e sono 2 per ogni entità, una pubblica e una privata. Nella crittografia a chiave asimmetrica si elimina il problema della condivisione della chiave; inoltre la chiave pubblica può essere compromessa da attaccanti senza che la chiave privata venga compromessa, e senza che venga compromessa la segretezza del messaggio.

Un altro aspetto fondamentale della crittografia è che la cifratura e la decodifica sono facili, *se le chiavi sono note*. Da ciò consegue che la sicurezza debba risiedere nella chiave, non nell'algoritmo in se.

2.2 Crittoanalisi

La scienza di recuperare il messaggio in chiaro senza conoscere il ciphertext si basa sostanzialmente su due differenti approcci:

- attacco brute-force;
- attacco crittoanalitico.

Attacco brute-force. Un attacco bruteforce è semplice: consiste nel provare tutte le chavi possibili fino ad indovinare quella corretta. Questa tipologia di attacco in generale è sempre possibile nella sua semplicità, tuttavia, se la dimensione dello spazio delle chiavi inizia ad essere elevata, il tempo che si deve impiegare diventa insostenibile, per cui in questi casi è necessario ricorrere ad altri stratagemmi.

Attacco crittoanalitico. In questo caso si assume che l'attaccante conosca l'algoritmo utilizzato nella cifratura dei messaggi; si trova quindi una qualche debolezza nell'algoritmo che permetta di farlo fallire.

In tal senso, si tende a rendere noto un algoritmo affinchè il maggior numero di persone tenti di attaccarlo, per aumentare al massimo le possibilità che venga trovata una falla. (In contrasto con la cosiddetta **security by obscurity**).

Tipologie di attacco. Consideriamo ora i possibili attacchi che un sistema crittografico deve affrontare per essere affidabile:

- *known cypertext attack:* questo attaccante è il meno aggressivo e conosce solamente il testo cifrato;
- *known plaintext attack:* conosce entrambi i tipi di testo;
- *choosen plaintext:* può scegliere il plaintext da codificare e analizzare il ciphertext ottenuto;
- *adaptive choosen plaintext:* può liberamente scegliere il plaintext da far codificare e comportarsi di conseguenza, sulla base del risultato appena ottenuto.
- *chosen ciphertext:* l'attaccante può scegliere differenti ciphertext e avere accesso al plaintext decriptato, per infine ricavare la chiave.

2.3 Notazione

La notazione usata è la seguente:

- $\mathcal{A}$ è l'alfabeto, ovvero un insieme finito di simboli;

- $\mathcal{M} \subseteq \mathcal{A}^{*}\star$ è il messaggio, dove $M \in \mathcal{M}$ è il *plaintext*;
- $\mathcal{C}$ è il messaggio cifrato, il cui alfabeto può anche differire da quello usato per M;
- $\mathcal{K}$ indica lo spazio delle chiavi;
- ogni $e \in \mathcal{K}$ denota una funzione biiettiva da $\mathcal{M}$ a $\mathcal{C}$, viene indicata con E_e ed è la funzione di cifratura;
- ogni $d \in \mathcal{K}$ è una funzione biiettiva da $\mathcal{C}$ a $\mathcal{M}$, indicata con D_d ed è la funzione di decodifica.

Data la notazione soprastante, indichiamo con cifrario un insieme $\{E_e \mid e \in \mathcal{K}\}$ e il suo corrispondente $\{D_d \mid d \in \mathcal{K}\}$ tale che per ogni $e \in \mathcal{K}$ esiste un solo $d \in \mathcal{K}$, in modo tale che $D_d = E_e^{-1}$. La coppia $\langle e, d\rangle$ forma una coppia di chiavi, dove e e d possono anche essere identiche (come nel caso della crittografia simmetrica).

3 Crittografia a chiave simmetrica

Nella crittografia a chiave simmetrica le chiavi sono le stesse ($e = d$), e i due interlocutori condividono una chiave. I cifrari possibili nel caso della crittografia simmetrica sono di 3 categorie:

- *cifrari a blocchi:* dividono il testo in blocchi di lunghezza fissa e cifrano un blocco alla volta;
- *cifrari a flusso:* cifrari a blocchi in cui la dimensione di ogni blocco è fissata a 1;
- *codes:* cifrari che lavorano su parole a lunghezza variabile.

3.1 Tecniche di sostituzione

Sono tutti quei cifrari che sostituiscono una lettera con un'altra lettere, basandosi su una qualche regola di sostituzione, come il cifrario di Cesare e la permutazione casuale.

Cifrario di Cesare. Il messaggio viene cifrato sostituendo ogni lettera l del messaggio con la $l + k$ esima lettera dell'alfabeto; la chiave quindi è data dalla coppia $(l, l + k)$.

Il cifrario di Cesare è facile da attaccare in quanto basta un attacco *bruteforce*, quindi è sufficiente provare tutte le combinazioni (che sono in totale 26).

Permutazione casuale. Supponiamo di usare come cifrario una permutazione casuale dell'alfabeto, ovvero sostituendo ad ogni lettera dell'alfabeto un'altra lettera, in modo totalmente casuale. In tal caso l'attacco bruteforce richiederebbe tempo eccessivo (ci sono $26!$ pssibili combinazioni da provare, che sono decisamente troppe).

La tecnica usata per attaccare questo tipo di crittografia è l'*analisi delle frequenze*, ovvero l'analisi delle lettere che capitano di più in una data lingua, e associare la lettera del messaggio cifrato con una data frequenza con quella nella lingua del messaggio con una frequenza simile.

Cifrario di Vigenère. Il cifrario di Vigenère riprende l'idea del cifrario di Cesare. L'idea è la seguente: presa una chiave (es. *key*), si ripete la chiave tante volte quanto è lungo il testo (eventualmente troncando l'ultima ripetizione), e si codifica la lettera con il corrispondete cifrario di Cesare.

K E Y K E Y K E Y K E Y
P R O V A D I T E S T O

La prima lettera del cipher text sarà la lettera ottenuta dal cifrario di Cesare di chiave (K, P), la seconda con la chiave (E, R) e così via.

Anche questo cifrario è semplice da attaccare, si parte dalla divisione del ciphertext in gruppi di lunghezza pari a quella della chiave, e si esegue l'analisi delle frequenze su ogni gruppo.

3.2 Cifrari a trasposizione

Funzionano in maniera leggermente diversa. Dato un blocco di lunghezza l e $\mathcal{K}$ un insieme di permutazioni su $\{1 \dots t\}$, si ha che

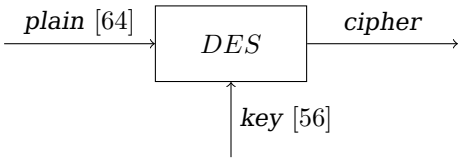
$$E_e(m) = m_{e(1)}, \dots, m_{e(2)}$$

Per decodificare si applica la permutazione inversa ad ogni carattere del ciphertext.

3.3 Cifrario di Feistel

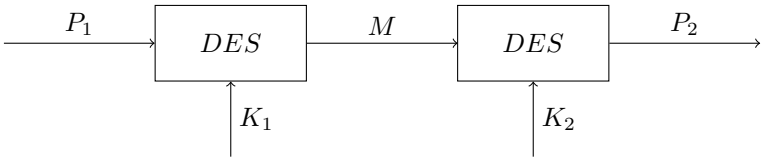
3.4 Data Encryption Standard (DES)

È un cifrario a blocchi che lavora su blocchi di 64 bit. Fu in effetti il primo standard di crittografia, e ne furono rilasciate versioni aggiornate che lavorassero su chiavi di lunghezza maggiore (3-DES). Non è ancora stato violato, ma è possibile ridurre in tempo lineare lo spazio delle chiavi da 2^{56} a 2^{43}.



3.4.1 Double DES e 3-DES

La variante DD, che usa DES due volte consecutive, è soggetta ad un attacco del tipo *Meet-in-the-Middle*.



L'attacco funziona nel seguente modo:

1. Dato un $C = E_{K_2}(E_{K_1}(P))$, sia $X = E_{K_1}(P) = D_{K_2}(C)$;
2. Dati P e C, cifrare P per ogni possibile chiave (sono 2^{56});
3. Generare una tabella con tutti i risultati, ordinati secondo X;
4. Decifrare C con tutte le possibili K_2, cercando un matching con quelle i risultati ottenuti prima. Ogni coppia è una possibile coppia valida, basta confrontare i risultati con P e C iniziali.

In effetti, guardando lo schema sopra, si nota che al massimo occorrono 2^{56} operazioni per violare questo protocollo.

Con 3-DES invece, un tentativo con la soluzione bruteforce necessita di almeno 2^{112} operazioni, un numero notevolmente più alto. Al momento infatti non esistono soluzioni per violare 3-DES.

3.5 Advanced Encryption Standard (AES)

Proposto come rimpiazzo di DES nel 1991, fu selezionato nel 2001. Infatti il DES iniziava a non andare più bene, in larga parte perchè era disegnato per i software degli anni 70 ed era abbastanza lento. AES funziona in maniera più snella e lavora su chiavi molto più lunghe (128, 192 e 256 bit).

3.6 Cipher block chaining

Come ci si comporta quando la lunghezza del messaggio eccede la dimensione del blocco? In tal caso, ci sono molteplici possibilità:

1. Splittare il messaggio in m blocchi, e cifrarli individualmente. Questa opzione è soggetta a pesanti limtazioni, la prima data dal fatto che identici plaintext vengono mappati su identici ciphertext (*information leak*); la seconda invece limita di parecchio le possibilità di individuare eventuali manomissioni del messaggio da parte di terzi (*integrity*);

2. Si può pensare in alternativa di far dipendere un carattere del ciphertext da quello precedente: dato un valore iniziale, il successivo carattere sarà cifrato con uno XOR tra il carattere precedentemente cifrato e il carattere da cifrare. In sostanza, dato un certo C_0,

$$C_i = E_K(P_i \oplus C_{i-1})$$
$$P_i = C_{i-1} \oplus D_K(C_i) \quad \text{(per la decodifica)}$$

Con la seconda soluzione, i caratteri cifrati dipendono strettamente da quelli precedenti, quindi è impossibile che due plaintext uguali vengano mappati su ciphertext uguali.

3.7 Posizionamento dei sistemi crittografici

Distinguiamo i casi di *link encryption* e *end-to-end encryption*. Nella link encryption ci sono sistemi di cifratura ad ogni collegamento, quindi i dati vengono decifrati e cifrati ad ognuno dei singoli collegamenti. Nella crittografia end-to-end invece, ciò che accade è che i sistemi di cifratura sono posizionati all'origine e alla destinazione dei dati, però in tal caso è necessario l'utilizzo di chiavi condivise tra i due interlocutori.

Guardando la questione da un punto di vista alternativo, ossia quello dello stack OSI, possiamo osservare come la link encryption sia applicata ai livelli più bassi dello stack, mentre man mano si sale viene applicata una crittografia di tipo end-to-end. Idealmente, serve che la crittografia end-to-end protegga i dati contenuti nei pacchetti, ma che lasci inalterati gli header, per permettere l'inoltro dei pacchetti. La link encryption protegge invece i dati di inoltro da monitoraggio e analisi da parte di terzi.

3.8 Distribuzione delle chiavi

Come già detto in precedenza, la crittografia a chiave simmetrica richiede che le parti condividano una chiave. Ciò può costituire un problema, dal momento che terzi malintenzionati possono sempre tentare di rubare la chiave sfruttando qualche falla nel sistema di condivisione della stessa.

Tipicamente non viene usata una sola chiave, ma una gerarchia di chiavi. Si ha quindi:

- *session key:* usata per crittografare dati per una sola sessione logica;
- *master key:* usata per cifrare le sessioni.

I problemi principali che si possono incontrare quindi sono i seguenti:

- la gerarchia delle chiavi è necessaria per reti molto vaste, ma è necessaria una sorta di garanzia sulle chiavi;
- il tempo di vita della chiave di sessione deve essere il minore possibile;
- l'uso di un sistema automatico di ditribuzione delle chiavi necessita la fiducia da parte degli utenti;
- il sistema di distribuzione è decentralizzato;
- è necessario stabilire una politica di controllo sull'uso delle chiavi.

4 Crittografia a chiave pubblica

Notazione. Oltre alla notazione specificata nella sezione 2.3, specifichiamo con PU_b e PR_b rispettivamente la chiave pubblica di B e la chiave privata di B.

4.1 Struttura del sistema

Questo tipo di crittografia elimina il problema della distribuzione delle chiavi, in quanto ogni utente ha due chiavi, una pubblica (che tutti possono vedere), e una privata (che *dovrebbe* rimanere incognita).

Ogni utente genera una coppia di chiavi, una pubblica e una privata. Quella pubblica viene inserita in un registro. Supponiamo che B voglia inviare un messaggio ad A. La procedura è la seguente:

1. B cifra il messaggio con la chiave pubblica di A;
2. A riceve il messaggio cifrato trasmesso da B;
3. A decifra il messaggio ricevuto usando la sua chiave privata.

Il vantaggio è evidente: senza doversi scambiare le chiavi, A è certa che il messaggio non sia stato letto in precedenza, in quanto è decifrabile solo con la chiave privata che solo lei possiede. Le applicazioni sono molteplici: si va dalla firma digitale, alla cifratura/decifratura di contentuti, fino allo scambio di chiavi.

4.2 Requisiti necessari per il funzionamento

Come per la crittografia a chiave simmetrica, ci sono dei requisiti fondamentali al sistema per garantire un processo di crittografia che sia ottimale:

- deve essere facile generare la coppia di chiavi;
- deve essere facile, per il mittente A, generare $C = E(PU_b, M)$;
- deve essere facile, per il destinatario B, calcolare $M = D(PR_b, C)$