

Sicurezza di Rete

Riassunto dei principali argomenti

Candidati:

Davide Bianchi

Matteo Danzi

Indice

1	Introduzione	2
2	Crittografia e integrità	2
2.1	Funzioni Hash	2

Sommario

Questa dispensa è scritta per la parte teorica del corso di Programmazione e sicurezza di Rete. Il codice L^AT_EX è disponibile a <https://github.com/alx79/dispense-info-univr.git>

1 Introduzione

Il fatto di garantire una protezione a determinati assets implica il garantire di alcune proprietà:

1. **Confidenzialità:** un utente non dovrebbe venire a conoscenza di cose che non è autorizzato a conoscere (riservatezza dei dati, privacy);
2. **Disponibilità:** rendere disponibili ad un utente autorizzato le informazioni che può avere e che richiede;
3. **Integrità:** impedire l'alterazione di dati e informazioni in maniera diretta o indiretta (anche in seguito a incidenti);
4. **Autenticità:** ad un utente deve essere garantita l'autenticità delle informazioni che riceve;
5. **Tracciabilità:** le azioni di un utente devono essere tracciate in modo univoco, in modo evitare eventuali casi di ripudiabilità.

Ciò che può compromettere le caratteristiche sopra elencate sono le minacce e gli attacchi. Viene definita *minaccia* una possibile violazione della sicurezza, mentre invece un *attacco* è una violazione effettiva della sicurezza.

Gli attacchi possono essere sostanzialmente di 4 tipologie:

- *attivi:* tentativi di alterare il funzionamento di un sistema;
- *passivi:* tentativi di carpire informazioni senza intaccare i meccanismi del sistema;
- *interni:* effettuati da un'entità interna al sistema;
- *esterni:* effettuati da un'entità esterna al sistema.

Gli attacchi (o le minacce) sono suddivisi in classi:

- *disclosure:* accesso non autorizzato alle informazioni;
- *deception:* accettazione di dati falsi;
- *disruption:* interruzione o prevenzione di informazioni corrette;
- *usurpation:* controllo non autorizzato di alcune parti del sistema.

2 Crittografia e integrità

Lo scopo storico della crittografia è quello di garantire la privacy, ossia come garantire che un'informazione ricevuta provenga effettivamente dall'utente che ci si aspetta l'abbia mandata.

2.1 Funzioni Hash

Una funzione hash è una funzione che trasforma un messaggio di lunghezza arbitraria in uno di lunghezza fissa (viene chiamato *hash* o *digest* del messaggio originale). Le funzioni hash attualmente più utilizzate sono MD5 e SHA.

Per soddisfare le condizioni di sicurezza, gli algoritmi che gestiscono le funzioni hash dovrebbero avere le seguenti caratteristiche:

- *coerenti:* a input uguali corrispondono output uguali;
- *casuali:* per impedire l'interpretazione del messaggio originale;
- *univoci:* la probabilità che due messaggi generino due hash uguali deve essere remota;
- *non invertibili:* deve essere impossibile (o computazionalmente complesso) risalire dal digest al messaggio originale.

Le funzioni hash vengono anche usate come fingerprint per verificare che nessuno sia intervenuto sul messaggio originale (altrimenti i due digest sarebbero diversi, vedi esempio).

Ora daremo un esempio di come possa avvenire una comunicazione sfruttando le funzioni hash. Alcune definizioni:

- m è il messaggio in chiaro;
- $H(m)$ è l'hash del messaggio;
- $c(x)$ è la funzione di cifratura;
- A e B sono due utenti.

Indichiamo inoltre come $H_A(m)$ l'hash del messaggio scritto da A .

Esempio. A scrive un messaggio e ne utilizza il testo come input di una funzione di hash, che genera il digest $H_A(m)$. A poi manda $c(m + H_A(m))$ a B .

B decifra e separa il contenuto del messaggio cifrato che ha ricevuto, e calcola con la funzione di hash un hash denominato $H_B(m)$. Se vale

$$H_B(m) = H_A(m)$$

il messaggio è autentico.

Se i due utenti non sono interessati a mantenere occultato il messaggio, viene utilizzato un MAC (Message Authentication Code), un segreto condiviso conosciuto da entrambi gli utenti. In questo caso viene mandato al destinatario il pacchetto con

$$m + H_A(m + s)$$

Usando un MAC si ha anche garanzia di autenticità, grazie al segreto condiviso. Qui sorge un nuovo problema: come poter scambiare con l'altro utente un segreto condiviso su un canale protetto? Per ovviare a questo problema è stato proposto un meccanismo di *firma digitale*, che **non usa chiavi segrete**.