

Protocolli basilari per la segretezza

Vediamo come utilizzare la misura crittografica per ottenere **segretezza** tra due agenti che vogliano comunicare.

Segretezza del messaggio m per A e B , due modi per ottenerla:

- **Crittografia simmetrica**

- **Prerequisito:** chiave k_{ab} sia condivisa tra A e B **solì**, cioè:

$$A \rightarrow B : m_{k_{ab}}$$

- **Crittografia asimmetrica**

- **Prerequisito 1:** B abbia una chiave privata valida (sicura, non scaduta)
- **Prerequisito 2:** A possa verificare che k_b è di B (ovvero A **conosce il proprietario della chiave pubblica**, ha un'associazione valida tra il messaggio e la chiave pubblica di B , e non ipoteticamente di un terzo agente C)

$$A \rightarrow B : m_{k_b}$$

Queste ultime due assunzioni, **insieme**, garantiscono segretezza. Se una di queste due viene a mancare la priorità di segretezza non viene garantita. Serve ovviamente **certificazione** per garantire il prerequisito **2** della crittografia asimmetrica.

In questo caso A non ha la certezza che B sia chi dice di essere. Il focus del protocollo si incentra sul come mandare ad un altro agente un messaggio che sia segreto, ovvero che non sia intelligibile se intercettato (e la misura crittografica garantisce questo per entrambi i casi simmetrico ed asimmetrico).

Protocolli basilari per l'autenticazione

Vediamo come utilizzare la misura crittografica per ottenere autenticazione tra due agenti che vogliano comunicare. A differenza della segretezza, qui c'è bisogno di un'evidenza del con chi si stia parlando.

Autenticazione di A con B , due modi per ottenerla:

- **Crittografia simmetrica**

- **Prerequisito 1:** chiave k_{ab} sia condivisa fra A e B **solì**
- **Prerequisito 2:** B possa verificare prerequisito **1**

$$A \rightarrow B : \text{"Sono io"}_{k_{ab}}$$

- **Crittografia asimmetrica**

- **Prerequisito 1:** A abbia una chiave privata valida
- **Prerequisito 2:** B possa verificare che k_a è di A .
 A cifrerà dunque con la sua chiave privata il messaggio che spedisce a B :

$$A \rightarrow B : //Sono io//_{k_a^{-1}}$$

Notiamo ancora una volta come segretezza e autenticazione siano due aspetti differenti.

Nel caso di crittografia simmetrica i prerequisiti che portano all'autenticazione implicano anche segretezza del messaggio (in quanto esso è cifrato e solo il diretto interessato può decifrarlo).

Discorso differente si ha per la crittografia asimmetrica. Nel punto 2 infatti il messaggio di A è decifrabile da chiunque possenga la chiave pubblica di quest'ultima.

Non è il contenuto del messaggio a fornire autenticazione (nell'esempio infatti si utilizza una stringa in chiaro) ma la misura crittografica apposta su di esso. Autenticiamo la controparte perché, viste le assunzioni, sappiamo per certo che solo il legittimo agente mittente possa aver cifrato (con la propria chiave pubblica) il messaggio che stiamo tentando di decifrare.

Combinare segretezza e autenticazione

Per ottenere entrambe le proprietà di segretezza ed autenticazione dobbiamo sommare le assunzioni delle due.

Come detto prima, nel caso della crittografia simmetrica la somma dei prerequisiti è già presente nell'autenticazione, differente è il caso della crittografia asimmetrica.

- **Crittografia simmetrica**

- **Prerequisito 1:** chiave k_{ab} sia condivisa fra A e B soli
- **Prerequisito 2:** B possa verificare prerequisito 1

$$A \rightarrow B : m_{k_{ab}}$$

- **Crittografia asimmetrica**

- **Prerequisito 1:** B abbia una chiave privata valida
- **Prerequisito 2:** A possa verificare che k_b è di B
- **Prerequisito 3:** A abbia una chiave privata valida
- **Prerequisito 4:** B possa verificare che k_a è di A

$$A \rightarrow B : \{m_{k_a^{-1}}\}_{k_b}$$

oppure

$$A \rightarrow B : \{m_{k_b}\}_{k_a^{-1}}$$

Ottenere integrità

Qualunque messaggio nella rete può essere alterato. Anche utilizzando l'hash la proprietà di integrità non è garantita perché l'attaccante potrebbe ricalcolare l'hash per il messaggio manomesso e spedirlo, facendolo passare per vero.

Aggiungiamo quindi all'hash, come irrobustimento della misura di integrità, un segreto (condiviso o asimmetrico).

1. **Caso asimmetrico:** $A \rightarrow B : m, h(m)_{k_a^{-1}}$ (**DS, Digital signature**)

2. **Caso simmetrico:** $A \rightarrow B : m, h(m, k_{sessione})$

Utilizziamo in questo caso il sistema simmetrico, ottenendo chiavi di sessione da utilizzare per costruire l'hash (**MAC, Message authentication code**)

L'unica differenza tra queste due misure, in termini di "bontà" della misura, è **la scalabilità**.

Nel primo caso chiunque può verificare la firma, nel secondo solo il ricevente, visto che c'è bisogno della chiave di sessione per farlo.

Alle misure di integrità, in principio, non importa cifrare il contenuto del messaggio, ma renderlo **inalterabile**.

Ancora una volta vediamo come una proprietà non implica le altre (in questo caso non implica segretezza).

Digital Signature

Nel caso della firma digitale il protocollo si compone di due step:

1. Generare la firma
2. Verificare la firma

Come nel mondo reale la firma digitale serve a **certificare a terzi l'integrità**. **Garantisce anche l'autenticazione ed eredita come limiti quelli della cifratura asimmetrica** (da cui deriva).

Analizziamo gli step:

1. *A*, partendo da un testo in chiaro, effettua:

1. L'hashing, trasformando il testo in chiaro in digest, ovvero:

$$cleartext \rightarrow digest$$

2. L'encryption, utilizzando la propria chiave privata, ovvero:

$$digest \rightarrow encrypted\ digest$$


Il testo in chiaro e l'artefatto prodotto al punto 2 costituiscono **insieme** la firma digitale.

$$[cleartext, encrypted\ digest]$$

2. Ricevuto il messaggio, B esegue la verifica della firma:

1. applica la funzione hash alla prima componente, il *cleartext*
2. decodifica la seconda componente, l' *encrypted digest*
3. confronta i due risultati
4. ottiene garanzia di integrità **se e solo se** essi combaciano, cioè se:

$$sign_A(m) = [m, h(m)_{k_a^{-1}}]$$

 **Domanda d'esame:** dov'è che la firma digitale nasconde l'encryption?

Risposta: nell'hash.

Autenticazione, segretezza ed integrità

Sommiamo ancora una volta i prerequisiti di autenticazione, segretezza e integrità.

- Segretezza del messaggio m per A e B
- Autenticazione di A con B
- Integrità di m nella transazione da A a B

Omettiamo per semplicità il caso della crittografia simmetrica ed analizziamo quello asimmetrico.

- **Prerequisito 1:** B abbia una chiave privata valida
- **Prerequisito 2:** A possa verificare che k_b è di B
- **Prerequisito 3:** A abbia una chiave privata valida
- **Prerequisito 4:** B possa verificare che k_a è di A

Otteniamo quindi un messaggio cifrato con la chiave pubblica di B e su cui A appone la propria firma digitale.

$$A \rightarrow B : sign_A(m_{k_b})$$

Nota bene:

- Ci sono in questa descrizione anche le assunzioni sull'hash.
- Nessuna delle 3 misure viste fino ad ora offre una garanzia di freshness (sono quindi soggetti a replay attack).

Certificati

Un certificato, in generale, non può prescindere da:

- **un identificativo**
- **un ruolo (qualità o proprietà) associato all'identificativo**
- **un marker dell'ente certificatore**

 **Domanda d'esame:** cosa significa autenticare un messaggio?

Risposta: riconoscere che questo abbia un preciso autore.

Il certificato deve essere integro e autentico per essere valido, dobbiamo quindi garantire la proprietà di autenticazione ed integrità. Garantiamo queste proprietà con la firma digitale. Indichiamo quindi il certificato "firmato" come:

$$\{A, K_a\}_{K_{CA}^{-1}}$$

Il certificato è **pubblico** ma non per questo non è robusto, infatti se la firma dietro al certificato perde di validità anche il certificato viene invalidato. **Il certificato descrive un'identità. Esprime l'identità che possiede la chiave privata associata alla chiave pubblica.**

Overview di alcuni protocolli più evoluti

Citiamo ora alcuni protocolli più evoluti.

- Basati su crittografia simmetrica:
 - **Diffie-Hellmann (DH)**
 - **RSA Key Exchange**
- Basati su crittografia asimmetrica:
 - **Public-Key Infrastructure (PKI)**

Protocollo Diffie-Hellmann

Indichiamo ora gli step del protocollo Diffie-Hellmann per lo scambio di chiavi in un sistema crittografico che voglia essere simmetrico.

- A e B concordano due parametri **pubblici** α e β **coprime tra loro**
- A genera X_a random (costruisce un segreto), quindi:

$$Y_a = \alpha^{X_a} \bmod \beta$$

- B genera X_b random (costruisce un segreto), quindi:

$$Y_b = \alpha^{X_b} \bmod \beta$$

- A e B eseguono il protocollo di scambio:

1. $A \rightarrow B : Y_a$
2. $B \rightarrow A : Y_b$

- Alla ricezione di **1**, B calcola:

$$Y_a^{X_b} \bmod \beta$$

- Alla ricezione di **2**, A calcola:

$$Y_b^{X_a} \bmod \beta$$

A e B , negli ultimi due passaggi del protocollo, **ottengono lo stesso valore** (la stessa chiave) grazie alle proprietà delle potenze. Attraverso operazioni locali, ciascuno ha calcolato qualcosa che ha calcolato anche l'altro.

Il **problema del logaritmo discreto**, introdotto dalla presenza del modulo, rende il risalire ad X_a (o X_b) computazionalmente non fattibile per un attaccante, in quanto richiede tempo esponenziale (**non può fare forging del segreto**). Questa misura previene che, anche in presenza di Dolev-Yao, questo possa risalire ai valori generati da A e B .

Notare bene che questo protocollo **non prevede misure di autenticazione**. Dati i parametri α e β chiunque può utilizzarli per contattare un agente.