Azzolini Riccardo 2021-05-06

Gestione delle chiavi pubbliche

1 Distribuzione delle chiavi pubbliche

Quando si usa la crittografia asimmetrica, a ogni utente A è associata una coppia di chiave pubblica KP_A e chiave privata KR_A. La chiave pubblica KP_A deve essere distribuita a tutti gli utenti che vogliono inviare ad A messaggi segreti e/o verificare firme generate da A.

Esistono varie soluzioni per la distribuzione delle chiavi pubbliche. In seguito verranno presentate alcune delle principali:

- annuncio pubblico;
- elenco (directory) pubblico;
- autorità di distribuzione delle chiavi;
- certificati a chiave pubblica.

2 Annuncio pubblico

La più semplice soluzione per la distribuzione delle chiavi pubbliche è l'**annuncio pubblico**: gli utenti distribuiscono le loro chiavi pubbliche a coloro con cui devono comunicare (ad esempio, nel caso della firma digitale la chiave può essere accodata al messaggio firmato) oppure le trasmettono in broadcast a un'intera comunità. Con questo schema si ha però il problema della **contraffazione**: un qualunque attaccante E può creare una chiave pubblica e distribuirla fingendosi di essere un altro utente A, e così, finché ciò non viene scoperto, E è in grado di leggere tutti i messaggi diretti ad A (e falsificare le firme di A).

3 Elenco pubblico

Con l'annuncio pubblico la contraffazione è possibile perché non si ha alcuna prova di quale sia l'identità associata a una chiave pubblica. Per ottenere un livello di sicurezza maggiore serve allora una qualche autorità, un ente fidato, che garantisca le associazioni tra identità e chiavi pubbliche.

Una possibilità è che le chiavi degli utenti vengano registrate presso un **elenco pubblico** (o *directory pubblica*) gestita da un'autorità fidata (ad esempio il datore di lavoro). La registrazione, cioè l'inserimento nella directory dell'identità e della chiave pubblica di un utente, deve essere effettuata dall'utente stesso di persona o tramite una modalità di autenticazione sicura, in modo che l'autorità che gestisce l'elenco possa *verificare l'identità* dell'utente al fine di garantire l'associazione tra identità e chiave. L'utente deve poi poter sostituire in ogni momento la sua chiave pubblica (sempre con le opportune verifiche di identità), ad esempio nel caso in cui la corrispondente chiave privata sia compromessa.

La directory deve essere accessibile elettronicamente tramite un protocollo di comunicazione autentica e sicura, in modo che chi deve comunicare con un utente possa ottenere la sua chiave pubblica e verificarne la correttezza. La comunicazione autentica e sicura impedisce a un attaccante di sostituire la chiave restituita dalla directory.

4 Autorità di distribuzione

L'elenco pubblico è una soluzione locale, legata a una singola "realtà", ad esempio un'azienda, ma un utente può avere una chiave pubblica personale, a prescindere ad esempio da dove lavora. La naturale evoluzione dell'elenco pubblico è dunque un ente che mantenga una directory di chiavi e gestisca la registrazione non nell'ambito di un singolo dominio, ma a livello più generale: un'**autorità di distribuzione**. Essa ha appunto lo stesso ruolo dell'autorità che gestisce un elenco pubblico, ma siccome fornisce chiavi che vengono usate in scenari "più aperti", ad esempio nella rete Internet invece che solo nell'intranet di un'azienda, migliora la sicurezza esercitando un controllo più rigido sulla distribuzione delle chiavi, tramite l'uso di un apposito protocollo per la distribuzione delle chiavi agli utenti che le richiedono.

4.1 Protocollo di distribuzione delle chiavi pubbliche

Un possibile protocollo per la distribuzione delle chiavi pubbliche è il seguente:

1. Quando un utente A vuole richiedere la chiave pubblica di B, invia all'autorità di distribuzione un messaggio contenente la richiesta della chiave (rappresentata ad esempio dall'identificatore di B) e un timestamp.
2. L'autorità risponde con un messaggio contenente
   - la chiave pubblica di B,
   - la richiesta originaria (ID_B),
   - il timestamp originario,

il tutto cifrato con la chiave privata dell'autorità per dare prova che la chiave sia stata restituita dall'autorità e non sostituita da un'attaccante.

Il timestamp presente nella risposta dell'autorità, che è uguale a quello presente nella richiesta, dà ad A la prova che la risposta sia appunto stata generata in seguito a questa richiesta, impedendo così attacchi replay.

Questi due messaggi coincidono sostanzialmente con i primi due passi del protocollo Needham-Schroeder a chiave pubblica, anche se i contenuti dei messaggi sono leggermente diversi: la richiesta è $\text{Request}, \text{Time}_1$ (ad esempio $\text{ID}_B, \text{Time}_1$) invece di ID_A, ID_B e Time_1 viene aggiungo anche alla risposta. Quando due utenti A e B devono comunicare tra di loro allora anche B richiede in modo analogo la chiave di A, e aggiungendo i messaggi per la doppia autenticazione tra A e B si completa l'esecuzione di Needham-Schroeder. Se S è l'autorità di distribuzione, i messaggi scambiati sono:

(1) $A \to S$: $\text{Request}_B, \text{Time}_1$
(2) $S \to A$: $\{KP_B, \text{Request}_B, \text{Time}_1\}_{KR_S}$
(3) $A \to B$: $\{\text{ID}_A, N_a\}_{KP_B}$
(4) $B \to S$: $\text{Request}_A, \text{Time}_2$
(5) $S \to B$: $\{KP_A, \text{Request}_A, \text{Time}_2\}_{KR_S}$
(6) $B \to A$: $\{N_a, N_b, \text{ID}_B\}_{KP_A}$
(7) $A \to B$: $\{N_b\}_{KP_B}$

(dopodiché A e B potrebbero ad esempio scambiarsi una chiave segreta per proseguire la comunicazione con la cifratura simmetrica).

5 Certificati a chiave pubblica

L'autorità di distribuzione può diventare un collo di bottiglia nel rispondere alle richieste, poiché *ogni volta* che un utente A vuole comunicare con B deve richiedere all'autorità la chiave KP_B anche se la conosce già, al fine di verificarne la validità. Per evitare questo problema sarebbe necessario poter riutilizzare più volte le informazioni ricevute dall'autorità di distribuzione, entro un determinato periodo di validità. A tale scopo sono stati definiti i **certificati a chiave pubblica**.

Un certificato contiene l'identità dell'utente, la sua chiave pubblica, il timestamp di emissione, il periodo di validità, e altre informazioni. Per dare una prova della correttezza del certificato, esso è *firmato* con la chiave privata dell'autorità che lo rilascia, la quale non si chiama più autorità di distribuzione, ma prende invece il nome di **autorità di certificazione** (**CA**, **Certification Authority**). Grazie alla firma, il certificato di un utente può essere distribuito senza ulteriori accorgimenti dall'utente stesso agli altri con cui vuole comunicare, e può essere verificato da chiunque abbia la chiave pubblica della CA.

Come l'autorità di distribuzione, la CA ha l'importante ruolo di tenere traccia dell'associazione tra l'identità di un utente e la sua chiave pubblica, verificandola in qualche modo (ad esempio richiedendo all'utente di presentare di persona un documento d'identità) prima di rilasciare un certificato: il certificato rilasciato è la prova che la CA ha verificato l'identità dell'utente. Di conseguenza, un certificato ha un valore solo per chi riconosce la CA che l'ha emesso e si fida di essa. Ad esempio, come caso limite, un utente potrebbe generare da solo il proprio certificato, oppure un certificato contraffatto: dal punto di vista degli algoritmi crittografici funzionerebbe tutto correttamente, ma questo certificato non darebbe alcuna prova dell'identità dell'utente.

5.1 Certificati X.509

Lo standard di fatto per la definizione e gestione dei certificati a chiave pubblica è **X.509**, che definisce:

- la struttura dei certificati;
- i protocolli di autenticazione.

Esso fa parte della serie di raccomandazioni ITU-T X.500, la quale definisce un servizio di directory (un server che gestisce informazioni su utenti e/o servizi) e prevede l'uso delle chiavi pubbliche per recuperare informazioni dalla directory. Tuttavia, i certificati X.509 possono essere usati anche per applicazioni/protocolli diversi, tanto è vero che attualmente essi vengono impiegati principalmente per il protocollo HTTPS usato in ambito Web.

Un certificato X.509 viene emesso da una CA e contiene le seguenti informazioni:

- il numero di versione X.509 (1, 2 o 3);
- il numero di serie del certificato, che è unico all'interno della CA (ciascun certificato emesso da una stessa CA deve avere un numero di serie diverso); tra tutti i certificati emessi da una stessa CA;
- l'identificatore dell'algoritmo di firma ed eventuali parametri di tale algoritmo;
- il nome X.500 della CA;
- il periodo di validità (da - a);
- il nome X.500 del proprietario del certificato, ovvero l'identità a cui il certificato si riferisce;
- la chiave pubblica, che comprende:
  - l'algoritmo per cui la chiave è stata generata (ad esempio RSA);
  - la chiave vera e propria;
  - eventuali altri parametri dell'algoritmo;

- dei campi extension (solo nella versione 3 di X.509), tra cui ad esempio il *key usage*, cioè l'utilizzo previsto per questa chiave (ad esempio, una chiave potrebbe essere considerata valida per verificare le firme generate dall'utente ma non per cifrare messaggi segreti da inviare all'utente);
- la firma digitale della CA.

I *nomi X.500* sono stati pensati per identificare le entità all'interno di una directory X.500. Siccome tali directory sono organizzate in modo gerarchico su degli attributi (nazione, organizzazione, unità dell'organizzazione, ecc.), un nome X.500 è modellato come una serie di attributi.

Riassumendo, un certificato X.509 contiene principalmente

- l'identità dell'utente (sotto forma di nome X.500),
- la chiave pubblica dell'utente,
- le informazioni sulla CA che l'ha rilasciato,
- l'indicazione del periodo di validità,

il tutto firmato con la chiave privata della CA.

5.2 Revoca di certificati

Normalmente un certificato è utilizzabile fino alla scadenza del suo periodo di validità, ma per vari motivi può essere **revocato** prima della scadenza:

- perché la chiave privata del possessore del certificato è compromessa;
- perché l'utente non è più certificato dalla CA;
- perché la chiave privata della CA è compromessa;
- perché è cambiata l'intestazione del certificato;
- ecc.

A tale scopo, ogni CA mantiene una lista di certificati revocati, chiamata **Certificate Revocation List** (**CRL**); le informazioni contenute in tale lista, anch'esse fissate dallo standard X.509, sono principalmente:

- le informazioni sulla CA;
- le date di quest'aggiornamento e del prossimo aggiornamento;
- le informazioni sui certificati revocati;
- la firma della CA.

La CA si impegna a mantenere aggiornata la CRL, e ogni applicazione che usa un certificato (non scaduto) deve prima verificare, tramite appositi protocolli, che esso non sia presente in una CRL.

5.3 Certificati emessi da CA diverse

Come detto prima, il ruolo di una CA è molto importante, poiché si delega a essa la verifica delle identità degli utenti. Accettare o meno un certificato significa dunque accettare o meno la procedura di verifica dell'identità che la CA ha svolto.

Quando due utenti A e B si scambiano i propri certificati per comunicare, se entrambi i certificati sono emessi dalla stessa CA allora sia A che B:

- conoscono la chiave pubblica della CA, dunque possono verificare le firme dei certificati;
- soprattutto, sanno di potersi fidare della procedura di verifica dell'identità svolta dalla CA.

Può invece succedere che il certificato di ciascuno dei due utenti sia rilasciato da una CA che l'altro utente non conosce:

- A ha un certificato rilasciato da una CA X_1, che si indica con la notazione $X_1\langle\langle A\rangle\rangle$ e che B non riesce a validare;
- B ha un certificato rilasciato da una CA X_2, $X_2\langle\langle B\rangle\rangle$, che A non riesce a validare.

In questo caso si applica il concetto di *transitività della fiducia*: se A sapesse che la propria CA X_1 riconosce la CA X_2, cioè che esiste una **relazione di fiducia** tra X_1 e X_2, allora A potrebbe a sua volta fidarsi di X_2 e dunque accettare il certificato $X_2\langle\langle B\rangle\rangle$.

Al fine di modellare la relazione di fiducia in modo che sia verificabile automaticamente, le CA X_1 e X_2 rilasciano ciascuna un certificato per l'altra:

- il certificato della chiave pubblica di X_2 rilasciato da X_1, $X_1\langle\langle X_2\rangle\rangle$, dà la prova che X_1 riconosce X_2;
- viceversa, il certificato della chiave pubblica di X_1 rilasciato da X_2, $X_2\langle\langle X_1\rangle\rangle$, comprova che X_2 riconosce X_1.

In generale, se una CA firma la chiave pubblica di un'altra CA, ovvero emette un certificato per tale chiave, certifica di riconoscere l'operato di quella CA e quindi tutti i certificati da essa rilasciati.

Se esiste questa relazione di fiducia, l'utente A che vuole comunicare con B invia non solo il proprio certificato rilasciato da X_1, ma anche il certificato che la CA di B, X_2, ha emesso per X_1:

(1) $A \to B$: $X_1\langle\langle A\rangle\rangle$, $X_2\langle\langle X_1\rangle\rangle$

Così, l'utente B:

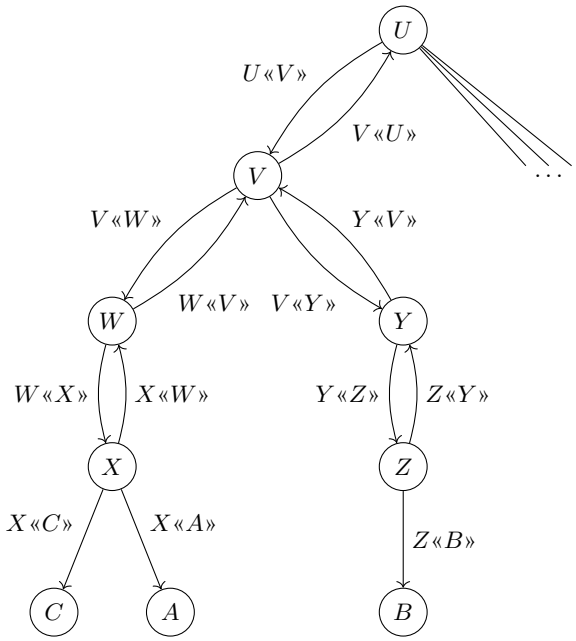
1. sa che si può fidare della propria CA X_2 e conosce la chiave pubblica di quest'ultima, che usa per verificare il certificato $X_2«X_1»$, dal quale ottiene la chiave pubblica di X_1 e la prova della relazione di fiducia tra le CA;
2. usa la chiave pubblica di X_1 appena ottenuta per verificare il certificato $X_1«A»$, che grazie alla relazione di fiducia tra le CA può accettare pur non conoscendo direttamente la CA che l'ha rilasciato.

Analogamente, l'utente B invia ad A il proprio certificato e il certificato della propria CA emesso dalla CA di A:

(2) $B \to A$: $X_2«B»$, $X_1«X_2»$

5.4 Gerarchia di CA

Quando ci sono un numero elevato di CA non è pratico avere una relazione di fiducia diretta tra ogni coppia di CA. La soluzione è organizzare le CA in una **gerarchia** ad albero e prevedere una relazione di fiducia solo tra nodi padre e figlio:

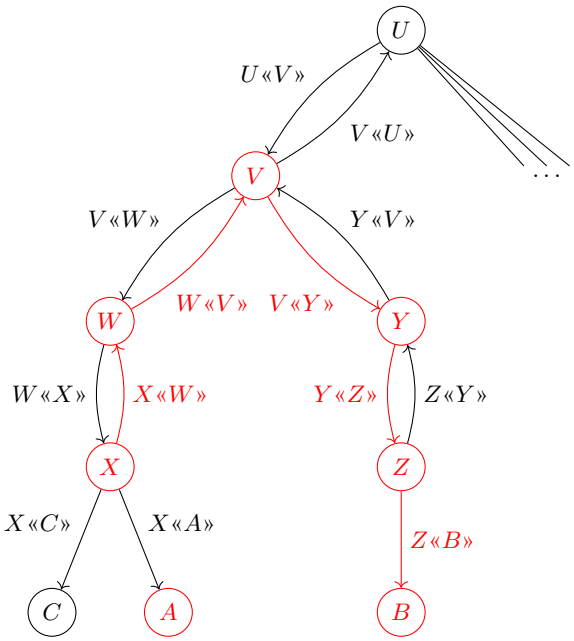


la radice dell'albero (nel disegno U) è la **CA root**, che ha relazioni di fiducia con (emette certificati per ed è certificata da) le **CA intermedie**, e così via fino agli **end user** (qui A, B e C), gli utenti finali dei certificati, che sono le foglie dell'albero.

Per verificare un certificato che ha ricevuto, un utente finale deve avere tutti i certificati nel percorso tra la sua CA e quella del mittente, che insieme prendono il nome di **catena**

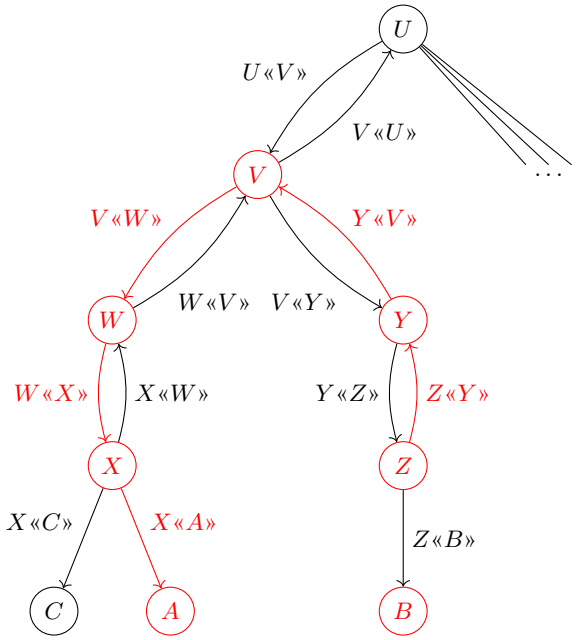
di certificati (*certificate chain*, chiamata anche *chain of trust*, *trust path* o *certificate path*). Ad esempio, nel disegno precedente, l'utente A può verificare il certificato di B utilizzando la catena

$$X\text{«}W\text{»},\ W\text{«}V\text{»},\ V\text{«}Y\text{»},\ Y\text{«}Z\text{»},\ Z\text{«}B\text{»}$$



Viceversa, B può verificare il certificato di A utilizzando la catena

$$Z\text{«}Y\text{»},\ Y\text{«}V\text{»},\ V\text{«}W\text{»},\ W\text{«}X\text{»},\ X\text{«}A\text{»}$$



Nella pratica esistono tantissime CA, organizzate non in un'unica gerarchia ma in più gerarchie separate, ciascuna con una diversa root. Una CA root emette certificati per se stessa, dunque la fiducia si fonda su di essa. In ambito Web, i certificati delle CA root sono già installati nei browser, il che determina quali siano le CA di cui i browser si fidano (ma in caso di certificati scaduti o non riconosciuti, solitamente i browser chiedono all'utente se proseguire comunque).