Meccanismi Asimmetrici

Return

Indice

Introduzione

Nei meccanismi iasimmetrici si ha una coppia di chiavi per ogni soggetto: una **chiave privata** (usata per firmare e decifrare) e una **chiave pubblica** (usata per verificare la firma o cifrare). Le due chiavi sono legate da una funzione pseudo-unidirezionale.

La chiaave privata deve essere tenuta con la massima segretezza, la chiaave pubblica invece deve potere essere associata univocamente ad un solo soggetto.

- **chiave privata**: robustezza, segretezza, integrità
- **chiave pubblica**: disponibilità, autenticità, robustezza, integrità

Tramite una **trapdoor one-way function** (funzione unidirezionale con porta d'accesso) è possibile ottenere la chiave pubblica dalla chiave privata.

Che si voglia cifrare o firmare un documento, si userà sempre una coppia di trasformazioni in cui una operazione è semplice e una difficile per chi non conosce la chiave privata. Nella cifratura è facile cifrare e difficile decifrare. Nella firma è difficile firmare e facile verificare.

Gli algoritmi asimmetrici hanno due principali vantaggi:

- si può ottenere la **non ripudiabilità**: il mittente non può negare di aver firmato un messaggio, in quanto solo lui possiede la chiave privata.
- si possono implementare meccanicsmi di **identificazione attiva**.

Per garantire riservatezza e autenticazione è lo stesso, cambia solo la coppia di chiavi che viene usata.

> **Cifratura** (riservatezza): il mittente cifra il messaggio con la chiave pubblica del destinatario e lo invia. Il destinatario decifra il messaggio con la sua chiave privata. A manda a B il messaggio cifrato $E_{P_B}(M)$, dove E_{P_B} è la cifratura con la chiave pubblica di B. B decifra il messaggio con la sua chiave privata $D_{S_B}(E_{P_B}(M))=M$.

> **Firma** (autenticazione): il mittente cifra il messaggio con la sua chiave privata e lo invia. Il destinatario verifica il messaggio con la chiave pubblica del mittente. A manda a B il messaggio firmato $S_{S_A}(M)$, dove S_{S_A} è la firma con la chiave privata di A. B verifica il messaggio con la chiave pubblica di A $V_{P_A}(E_{S_A}(M))$.

Come faccio ad associare ad una chiave pubblica un proprietario? Se ogni utente distribuisse autonomamente la propria chiave pubblica si potrebbe incoorere in un attacco di **man in the middle** Serve quindi un ente fidato per la distribuzione delle chiavi pubbliche. Questo ente è chiamato **PKI** (Public Key Infrastructure) e si occupa di generare le chiavi, distribuirle e gestirle e generare i certificati.

Il Certificato

Rilascio del certificato

vediamo come viene rilasciato e utilizzato un certificato:

1. L'utente X costriusce e invia il messaggio $mX = X||IX||PX||IPX$ dove
   - X è l'identificativo dell'utente
   - IX è un campo di informazioni sull'utente
   - PX è la chiave pubblica dell'utente
   - IPX è un campo di informazioni sulla chiave
2. Il server T costruisce una struttura dati $mS: $S=mT||IT\n=mT||mX$$
3. Il server calcola l'hash del messaggio, lo firma e lo pubblica: $$H(m)\S_{S_T}(H(m))\Cert(PX,T)=m||S_{S_T}(H(m))$$ *"io, T, certifico che X è il proprietario della chiave pubblica PX"*

Il *man in the middle* non è più possibile perchè non conosce la chiave del server T.

Formato del certificato

il certificato è formato da una parte di dati e una di firma digitale. La parte di dati contiene:

- **versione**: versione del certificato
- **serial number**: numero di serie del certificato (univoco)
- **protocollo di firma**: algoritmo di firma usato da T
- **emittente**: identificativo dell'emittente
- **validità**: periodo di validità del certificato
- **soggetto**: identificativo del soggetto
- **informazioni sulla chaive pubblica del soggetto**: quando e come è stata generata

Ci sono diverse estensioni che sono state aggiunte nelle ultime versioni del certificato:

- aggiungere informazioni sulla chiave
- aggiungere informazioni sulle politiche di emissione
- aggiungere infomrazioni sull'utente e/o sull'ente di certificazione
- aggiungere ulteriori informazioni

PKI - Public Key Infrastructure

Il sistema che permette di ottenere certificati e pubblicare le chiavi pubbliche è detto PKI (Public Key Infrastructure). Comprende:

- **Registration Authority (RA)**: entità che si occupa di raccogliere le richieste di certificato e di verificare l'identità del richiedente
- **Certification Authority (CA)**: rilascia il certificato per le chiavi e lo pubblica sul database. non deve essere contattata dall'esterno, e deve uscire solo sul DB per sicurezza.
- **Directory (DB)**: ospita i certificati delle chiavi e li mette a disposizione.

In fase di registrazione, l'utente invia la richiesta di certificato alla RA, che verifica l'identità dell'utente e invia la richiesta alla CA. La CA genera il certificato e lo pubblica sul DB. A regime l'interazione è solo tra Utente e DB.

Directory (DB)

È un database per salvare i certificati delle chiavi delgli utenti. Deve sempre essere disponibile 24 ore su 24 e 7 giorni su 7.

Segue lo standard X.500, che definisce un protocollo per la gestione di directory distribuite. Le directory sono organizzate in una struttura ad albero, in cui ogni nodo rappresenta un oggetto e ogni oggetto ha un identificativo unico (DN - Distinguished Name). In questo caso gli oggetti sono i certificati delle chiavi pubbliche.

Le ricerche possono essere fatte tramite nome assoluto (univoco) o tramite nome relativo (non univoco). Le directory possono essere distribuite su più server, in modo da garantire la disponibilità e la ridondanza dei dati, X.500 gestisce automaticamente e efficientemente la distribuzione e la ricerca dei dati.

Ricerca di un certificato

Per richiedere un certificato della propria chiave, l'utente si rivolge a un ente di certificazione. Nel modello a tre parti (Utente, RA, CA) l'utente chiede un certificato alla RA specificando:

- dati personali
- algoritmo di firma
- chiave pubblica
- firma del messaggio di richiesta

Protocolli di gestione

Oltre al modello a tre parti, esiste anche il modello centralizzato (a due parti) in cui l'utente si rivolge direttamente alla CA per richiedere il certificato. La generazione delle chiavi avviene sul server e l'unico messaggio che viene inviato è originato dalla CA per comunicare all'utente le chiavi che ha generato. Questo modello però permette a due enti di avere la chiave privata dell'utente, quindi non è sicuro. Viene usato ad esempio nelle aziende quando l'ente di certificazione è l'azienda stessa che vuole permettere ai dipendenti di cifrari i dati, in questo modo, in caso di azioni malevole da parte di un dipendente, l'azienda può accedere ai dati cifrati.

Proof Of Possession - POP

La CA deve essere in grado di dimostrare che il soggetto richiedende possiede la chiave privata associata alla chiave pubblica. Altrimenti si possono verificare diversi casi:

- un utente si fa rilasciare un certificato per la sua chiave spacciandosi per un altro utente emettendo documenti sotto falso nome.
- (di conseguenza) un utente può ripudiare la propria firma, in quanto non è in grado di dimostrare di possedere la chiave privata associata alla chiave pubblica.

Il metodo migliore per avere la prova di possesso è il **POP a tempo di firma**. Questo metodo consiste nel inserire nei messaggi un riferimento al certificato e firmare con la propria chiave privata. In modo che se il destinatario riesce a leggere il certificato ha la prova dell'identità.

> Esempio: Per provare la propria identità, l'utente A che ha richiesto un certificato alla CA T, riceve un nonce da T e lo firma con la propria chiave privata, lo invia a T e T verifica la firma con la chiave pubblica di A. Se la verifica ha successo, T può rilasciare il certificato.

La soluzione alternativa (utilizzata effettivamente) è il **POP a tempo di registrazione**. In questo caso l'utente A richiede il certificato mandando la propria chiave pubblica concatenata alla propria chiave pubblica firmata con la chiave privata. Se CA, verificata la firma, trova due chiavi pubbliche identiche allora A è in possesso della chiave privata associata e rilascia il certificato.

Revoca dei Certificati

Quando si ha il sospetto che la propria chiave privata sia stata compromessa è necessario revocare il certificato, quindi ripudiarla. Il ripudio è l'azione con la quale un utente spezza l'associazione tra chiave pubblica e proprietario, sancita da un certificato.

Una volta notificato il ripudio alla RA, la CA emette la revoca del certificato. Come fanno gli utenti a sapere che un certifato è stato revocato? I metodi si classificano in:

- **PULL**: l'informazione è pubblicata su una repository pubblica e gli utenti la consultano quando ne hanno bisogno.
- **PUSH**: Ogni utente specifica i certificati di interesse, la CA ha la responsabilità di notificarli quando ci sono cambiamenti.
- **ONLINE STATUS CHECKING**: Richiesta connessione tra CA e utente a tempo di verifica
- **OFFLINE STATUS CHECKING**

CRL - Certificate Revocation List

È un metodo PULL che consente la notifica offline. La CA rilascia periodicamente una lista (CRL) di certificati revocati. La struttura dati contiene i seguenti campi:

- Nome della CA che ha emesso la lista
- Data di emissione della lista
- Data della prossima emissione
- Certificati revocati: ID del certificato, data di revoca

La CRL aumenta periodicamente di dimensioni, la soluzione utilizzata è di dividere la lista in sottoliste in base all'ID del certificato.

Se l'utente ha bisogno di informazioni sui certificati revocati in tempo reale si utilizza il metodo **OCSP** (Online Certificate Status Protocol).

OCSP - Online Certificate Status Protocol

È un protocollo PULL online. I server offrono il serrvizio di consultazione in tempo reale dello stato dei certificati, fanno uso anche delle CRL per ottimizzare le richieste. Questi server sono detti **OCSP Responder**. Si dividono in:

- **Trusted Respondere**: il server firma con una coppia chiave:certificato indipendere da quella della CA.
- **Delegated responder**: il server firma con la coppia chiave:certificato in base alla CA che lo ha emesso.

Modelli di Fiducia

Non tutti gli utenti si affidano alla stessa CA, quindi è necessario un modello di fiducia che permetta di stabilire la fiducia tra le CA attraverso un **cammino di fiducia**.

Il protocollo di ricerca del percorso di fiducia è il seguente:

1. Il certificato del nostro "destinatario" è firmato da una CA_x
2. Cerchiamo le entità che hanno certificato CA_x.
3. Tra le entità c'è una ROOT AUTHORITY? (no -> torniamo a cercare per ogni entità dal passo 1)
4. Se c'è una ROOT AUTHORITY abbiamo trovato il cammino di fiducia.

È possibile avere più cammini di fiducia, è anche possibile che due CA diverse si certifichino a vicenda, in questo caso si ha una **cross certification**.

Trovato un cammino di fiducia si verificano le firme di tutti i certificati, si controlla che il soggetto di ogni certificato sia colui che ha emesso il certificato successivo, si controlla il periodo di validità e si verifica che il certificato non sia revocato.

PKI

La PKI è una infrastruttura che comprende la **Registration Authority (RA)**, la **Certification Authority (CA)** e la **Directory (DB)**. Affinchè si possa parlare di PKI è necessiario che abbia alcuni requisiti:

- **Key history**: necessario mantenere una history delle chiavi
- **Cross certification**: necessaria per la fiducia tra le CA
- **Key backup & recovery**: necessaria per il recupero delle chiavi
- **Aggiornamento automatico della chiave**: sarebbe bene avere l'aggoirnamento della chiave in odo automatico alla scadenza
- **Supporto al non ripudio**: è fondamentale garantire agli utenti di potere comunicare certi che sia garantita la non ripudiabilità
- **Timestamping**

Certificati e Reti Sicure

Riprendendo Diffie-Hellman, il problema principale è garantire che Y_A e Y_B provengano effettivamente da A e B, ovvero manca identificazione.

Una prima opzione per risolvere il problema è il **Fixed Diffie-Hellman**. Questo meccanismo prevede una entità di certificazione che crea un certificato contenente p,g,Y per entrambe le entità che vogliono comunicare. Queste comunicheranno quindi scambiandosi il certificato e non più i valori Y. La chiave genereata poi non è la chiave definitiva, ma è detta **pre-master key**. La chiave definitiva è generata tramite una randomizzazione concatenando la pre-master key con due nonce generati da entrambe le entità e scambiati. La chiave definitiva è quindi: $$K_{final} = H(K_{pre-master}||nonce_A|nonce_B)$$

Una seconda opzione è il **Diffie-Hellman Ephemeral**. All'inizio della comunicazione i due partecipanti scelgono una chiave privata, calcolano la chiave pubblica, la firmano con la privata e la inviano all'altro partecipante concatenando il certificato della firma.