Key Distribution

Riccardo Longo

Distribuzione delle Chiavi

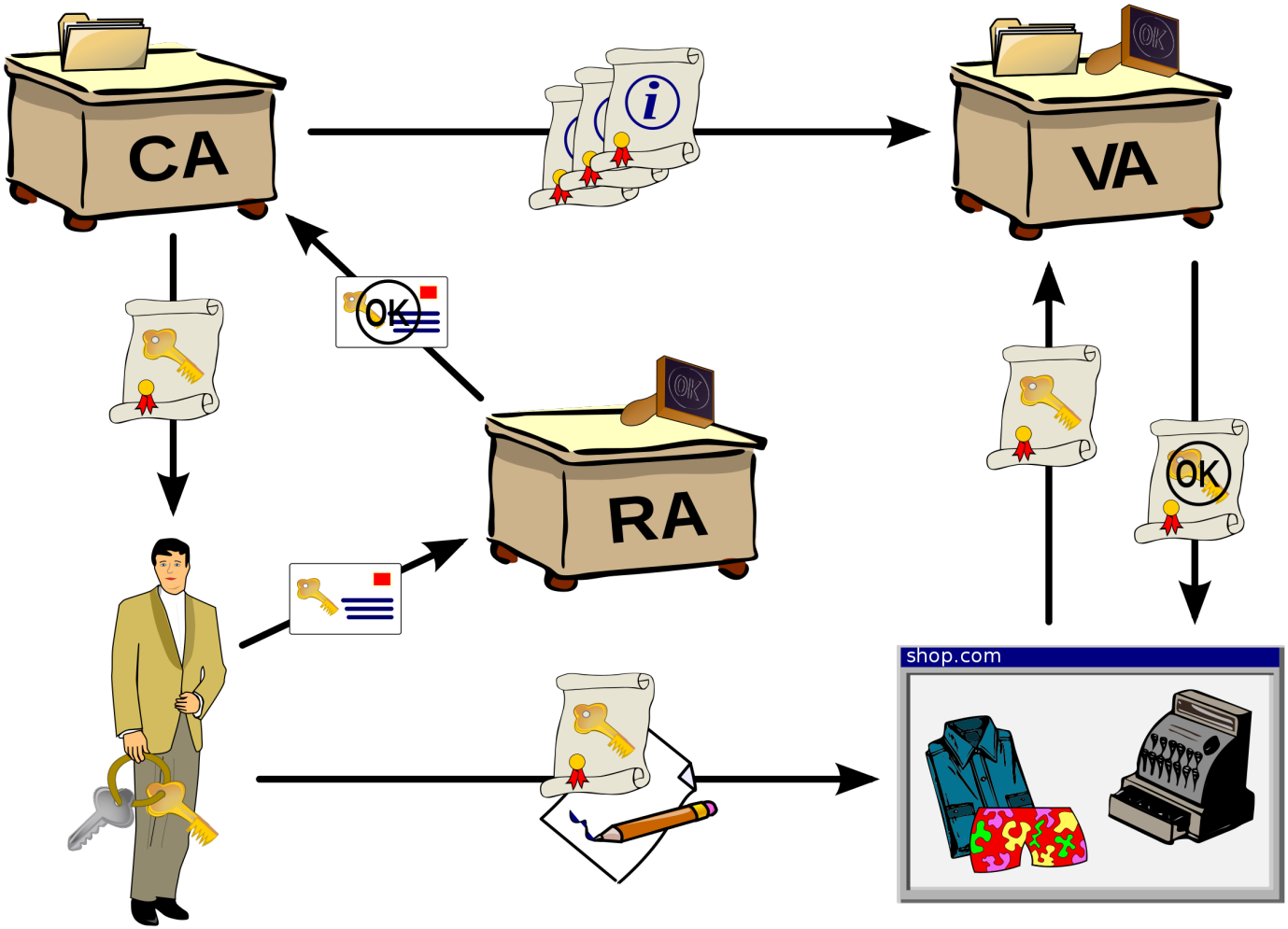
La gestione delle chiavi è una questione critica:

- Le chiavi **simmetriche** richiedono condivisione **privata** ed **autenticata**
  - Scambio chiavi **di persona**
  - Corriere fidato
  - Canale cifrato (dipende da scambio precedente)
- Le chiavi **pubbliche** richiedono **verificabilità** e **autenticazione**
  - Identità corretta
  - Informazioni aggiornate

Public Key Infrastructure

- È un insieme di ruoli, regole e procedure per la gestione di **certificati**:
  - **Creazione**
  - **Distribuzione**
  - **Gestione**
  - **Uso**
  - **Revoca**
- Cura l'associazione di **chiavi pubbliche** ad **identità**

PKI



Certification Authority

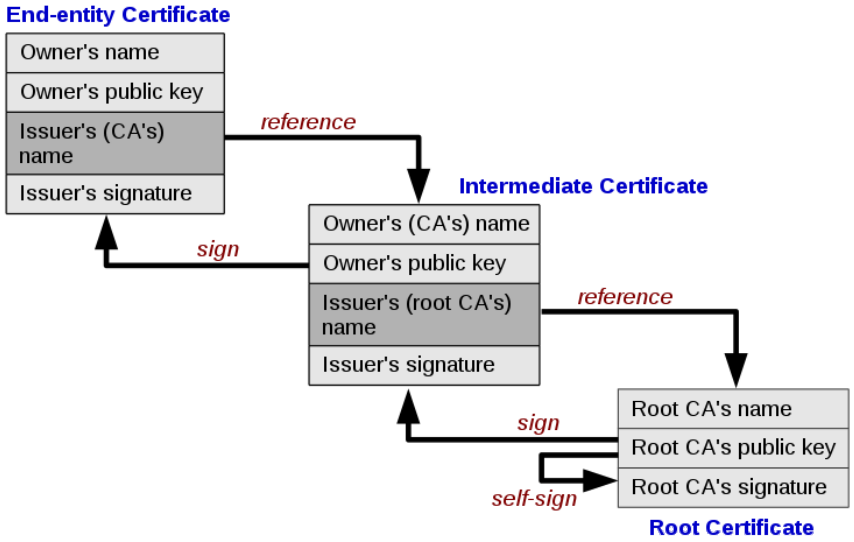
- **Trusted Third Party**
- Considerata **affidabile** sia da mittente che destinatario
- Assicura la **validità** dei certificati firmandoli
- Verifica l'**identità** di chi richiede un certificato
- **Distribuisce** i certificati a chi li usa

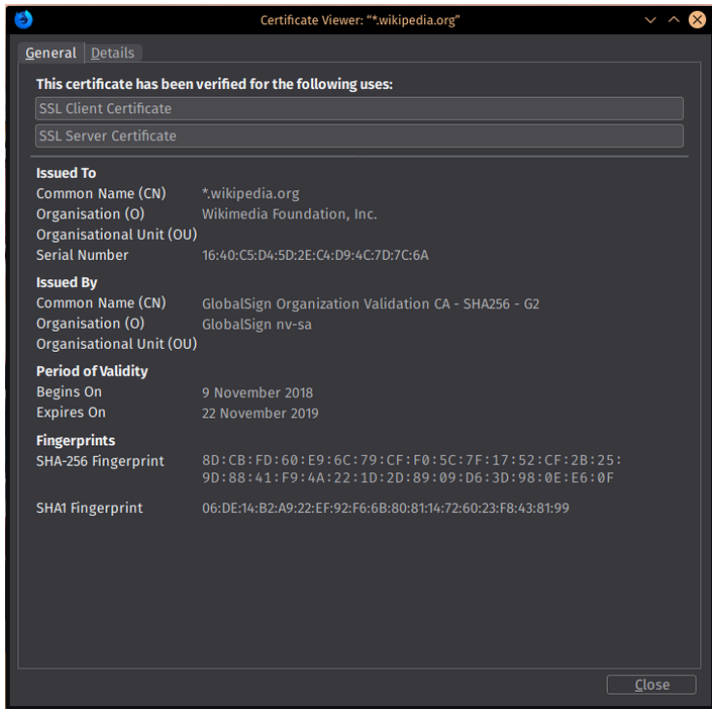
Web of Trust

- Gli utenti si autenticano **reciprocamente**
- Mi fido di chiavi **firmate** da chi **mi fido**
- Fiducia **distribuita**
- Basato su **keyserver**
- Espone **relazioni** tra utenti
- Può essere **difficile entrare**
- Standard **PGP**

Certificati

- Contengono:
  - Informazioni sull'**identità** del proprietario
  - **Chiave pubblica** del proprietario
  - Periodo di **validità**
  - Info sulla **CA**
  - **Firma** della CA
- **Gerarchia** di CA per firmare certificati di diversa importanza





Revoca

- Un certificato può non essere più valido **prima della scadenza**:
  - **Compromissione** o **perdita** della chiave privata
  - **Emissione** errata o **malevola**
  - **Cambio** identità o permessi
  - **Sostituzione** con chiavi più nuove
- **CRL**: Certificate Revocation List
  - Lista di certificati revocati firmata dalla CA
  - Controllo **online**
  - Pesante da mantenere
- **OCSP**: Online Certificate Status Protocol
  - Client richiede verifica di validità del **singolo** certificato
  - Espone cronologia alla CA
  - Pesante per la CA
- **OCSP Stapling**: al certificato viene allegato una verifica aggiornata della sua validità

Key Agreement

- Gli interlocutori si accordano su una chiave **simmetrica**
- Scambio sicuro su **canale insicuro**
- **Entrambi** influenzano il risultato
- Per avere sicurezza l'accordo dev'essere **autenticato**
  - Chiavi **pubbliche**
  - Segreto (password) **condivisa**
  - Verifica successiva via **altri canali**

Diffie-Hellman Key Exchange

- Scambio di chiavi **non autenticato**
- Va integrato con autenticazione per evitare attacchi attivi
- Progenitore della crittografia a chiave pubblica, basato su **DLOG**
- Varianti basate su gruppi $\mathbb{Z}_p$ o **curve ellittiche** (ECDH)
- **Standard** alla base di molti protocolli

DH

privato pubblico

- **Parametri** di sistema: $\mathbb{G}$ gruppo di ordine n, g generatore
- **Alice** sceglie $1 < a < n$ random, pubblica $A = g^a$
- **Bob** sceglie $1 < b < n$ random, pubblica $B = g^b$
- **Alice** calcola il segreto $S_a = B^a$
- **Bob** calcola il segreto $S_b = A^b$
- I segreti coincidono:

$$S_a = B^a = (g^b)^a = g^{ba} = g^{ab} = (g^a)^b = A^b = S_b$$

Station To Station Protocol

- Basato su **DH**
- Aggiunge **autenticazione** e verifica della chiave
- Ha alcune varianti
  - Per fornire solo mutua autenticazione
  - Usando un **MAC** per verificare la chiave
  - Evitare **unknown key-share attacks**

STS base

- **Setup**:
  - Alice e Bob hanno **coppia di chiavi asimmetriche** per firmare e conoscono la chiave pubblica dell'altro
  - Sono stati stabiliti i parametri per DH
- **Alice** genera x random, invia g^x a Bob
- **Bob** genera y random, calcola g^y ed il segreto $K = (g^x)^y$
- **Bob** firma la coppia **ordinata** (g^y, g^x), cifra la firma usando K, invia ad Alice:

$$g^y, E_K(S_B(g^y, g^x))$$

- **Alice** calcola il segreto $K = (g^y)^x$, decifra e controlla la firma di Bob
- **Alice** firma la coppia **ordinata** (g^x, g^y), cifra la firma usando K, invia a Bob:

$$E_K(S_A(g^x, g^y))$$

Man In The Middle

- Attaccante attivo
- Posto tra i due interlocutori
- **Intercetta** e **modifica** il traffico
- Sventato principalmente con **autenticazione**
- Varianti:
  - **Man in the Browser**: particolarmente pericoloso: si inserisce prima della cifratura TLS
  - **Boy in the Browser**: Malware che cambia il routing per effettuare un classico MITM
  - **Man on the Side**: attaccante vede il traffico, può inserire messaggi ma non cancellarli o modificarli

Forward Secrecy

- Requisito di sicurezza per i protocolli di comunicazione moderni
- Prevede che le **sessioni passate** restino sicure anche se la **chiave a lungo termine** viene compromessa e in presenza di **attaccanti attivi**
- Ogni sessione dev'essere cifrata con **chiavi effimere random** ed **indipendenti** dalla chiave a lungo termine
- La sicurezza si riferisce alle chiavi, quindi il cifrario deve resistere alla **crittanalisi**