Integrità Autenticazione Autorizzazione

Damiano Carra

Università degli Studi di Verona
Dipartimento di Informatica

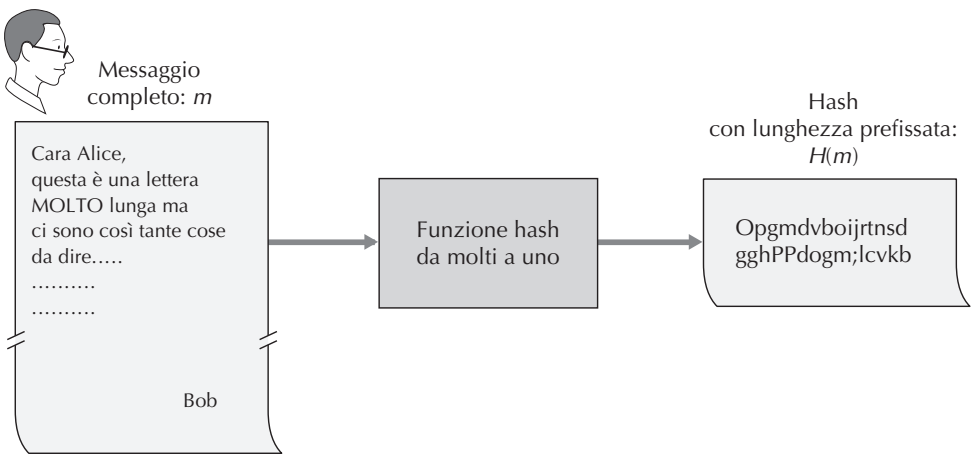
Parte I: Integrità

Considerazioni sulla crittografia

- L'obbiettivo storico della crittografia è garantire privacy
  - Dati non possono essere letti da non autorizzati
- Problema: dato un messaggio che sembra provenire da un utente, come posso essere sicuro che esso arriva effettivamente da tale utente?
  - Autenticità e integrità del messaggio
  - Problema intrinsecamente diverso da quello di proteggere i contenuti
- Per affrontare questi argomenti ci servono altri strumenti
  - Funzioni hash

Funzioni hash

- Una funzione hash trasforma un messaggio di lunghezza arbitraria in output di lunghezza fissa
  - Chiamato hash o digest del messaggio originale



Funzioni hash

- Per soddisfare le condizioni di sicurezza stabilite per le funzioni hash, gli algoritmi devono essere:
  - Coerenti
    - Input uguali corrispondono output uguali
  - Casuali, o apparire tali
    - Per impedire l'interpretazione accidentale del messaggio originale
  - Univoci
    - La probabilità che due messaggi generino il medesimo hash deve essere virtualmente nulla
  - Non invertibili
    - Risalire al messaggio originale dall'output deve essere impossibile

Funzioni hash

- Le funzioni hash non invertibili vengono normalmente utilizzate per assegnare un'impronta digitale a un messaggio o a un file
  - Come le impronte dei polpastrelli, un'impronta hash è univoca e costituisce una prova dell'integrità e dell'autenticità del messaggi
  - Se A e B vogliono accertarsi che nessuno sia intervenuto sul contenuto del messaggio in fase di transizione utilizzano proprio una funzione hash non invertibile

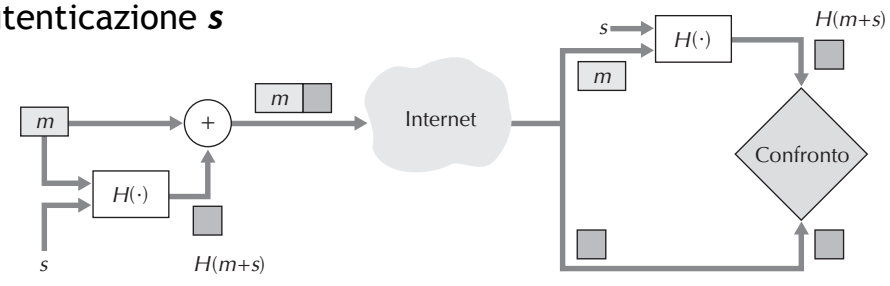
Esempio – Messaggio cifrato

- A scrive un messaggio e ne utilizza il testo come input di una funzione hash non invertibile
- Il risultato della funzione hash viene accodato al messaggio e ne costituisce l'impronta digitale
  - Il tutto viene cifrato e inviato a B
- B decifra ciò che ha ricevuto, separa il messaggio dall'impronta e utilizza il testo del messaggio come input della medesima funzione hash utilizzata da A
- Se i due hash corrispondono, B è certo che nessun altro sia *intervenuto* sul messaggio

Approccio simile al checksum usato negli header dei protocolli (con l'aggiunta della crittografia)

Esempio – Messaggio non cifrato

- In alcune situazioni, A e B non sono interessati ad occultare il contenuto del messaggio
  - Ma vogliono comunque preservare l'integrità
- Message Authentication Code (MAC)
  - Serve disporre di un codice segreto condiviso → una chiave di autenticazione ***s***



Legenda:

m = Messaggio

s = Segreto condiviso

Funzioni hash: considerazioni

- Funzioni hash più comuni:
  - l'algoritmo MD5 (Message Digest 5)
  - l'algoritmo SHA (Secure Hash Algorithm)

- Altri usi delle funzioni hash
  - UNIX utilizza funzioni hash per gestire le password
    - Piuttosto che memorizzare la password si memorizza la sua hash
    - Quando un utente fa il login, Unix fa l'hash della password digitata e lo confronta con il valore in memoria

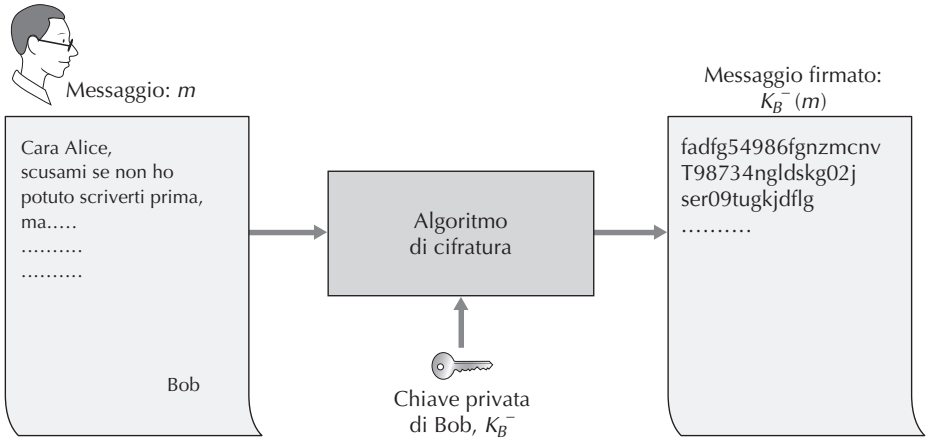
Autenticità dei messaggi

- Finora abbiamo considerato l'integrità dei messaggi
  - Dato un messaggio, come posso essere sicuro che non sia stato manipolato?
- Con il MAC, c'è anche garanzia di autenticità
  - La chiave di autenticazione "s" è segreta e associata ad un'utente
  - Tuttavia "s" deve essere preventivamente distribuita tra chi comunica
    - Attraverso un canale sicuro
- Come è possibile garantire l'autenticità senza distribuire chiavi segrete?
  - Si parla di firma digitale
  - Attuabile con crittografia asimmetrica

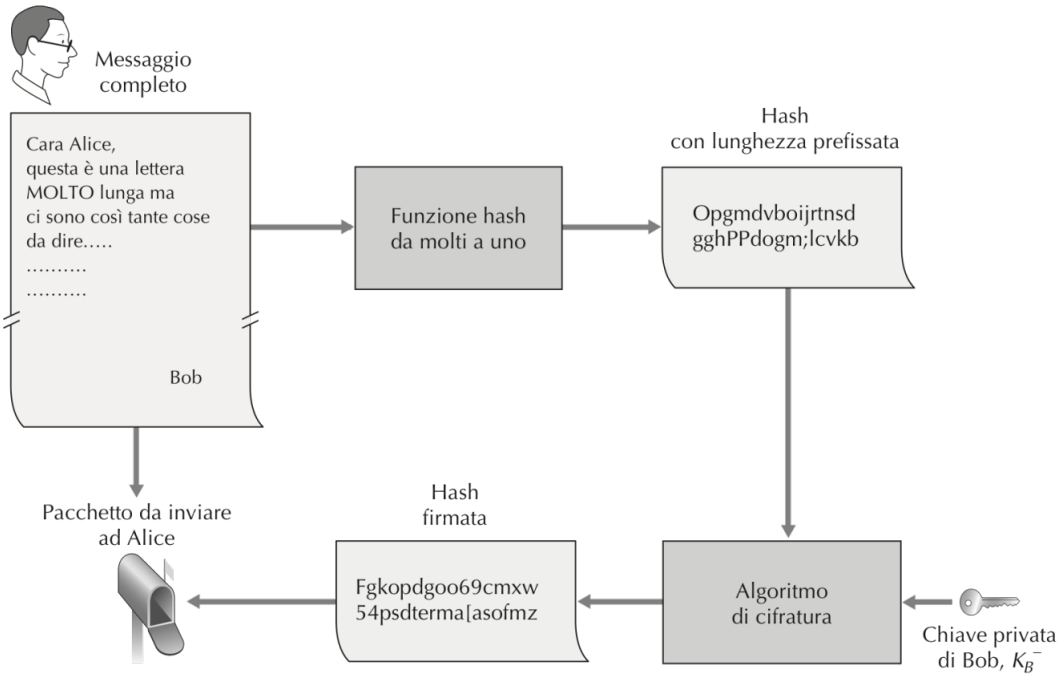
Firma digitale

- Una firma digitale è l'equivalente informatico di una firma convenzionale
  - Una firma digitale è (in generale) non ripudiabile
- Si sfrutta RSA in modo inverso a quanto fatto per cifrare
  - L'algoritmo di cifratura diventa l'algoritmo di verifica
  - L'algoritmo di decifratura diventa l'algoritmo di firma
- Aspetti computazionali
  - Firmare l'intero documento (= cifrarlo con la chiave privata) è molto oneroso
  - Approccio più efficiente se RSA viene combinato con le funzioni hash

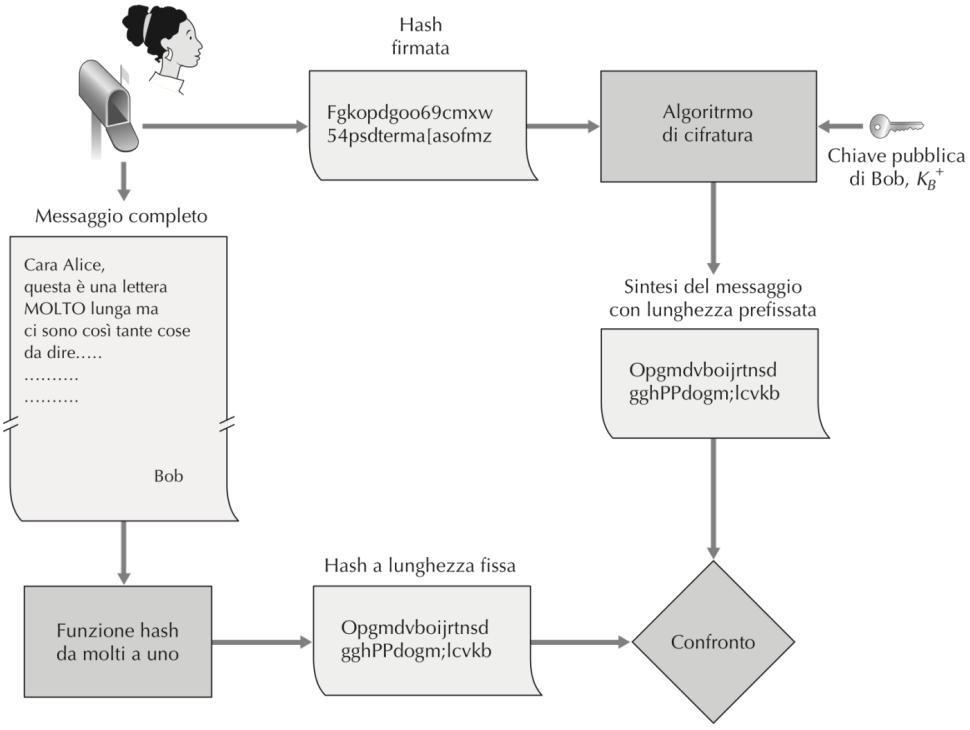
Firma di un documento (oneroso)



Firma di un documento (più efficiente)

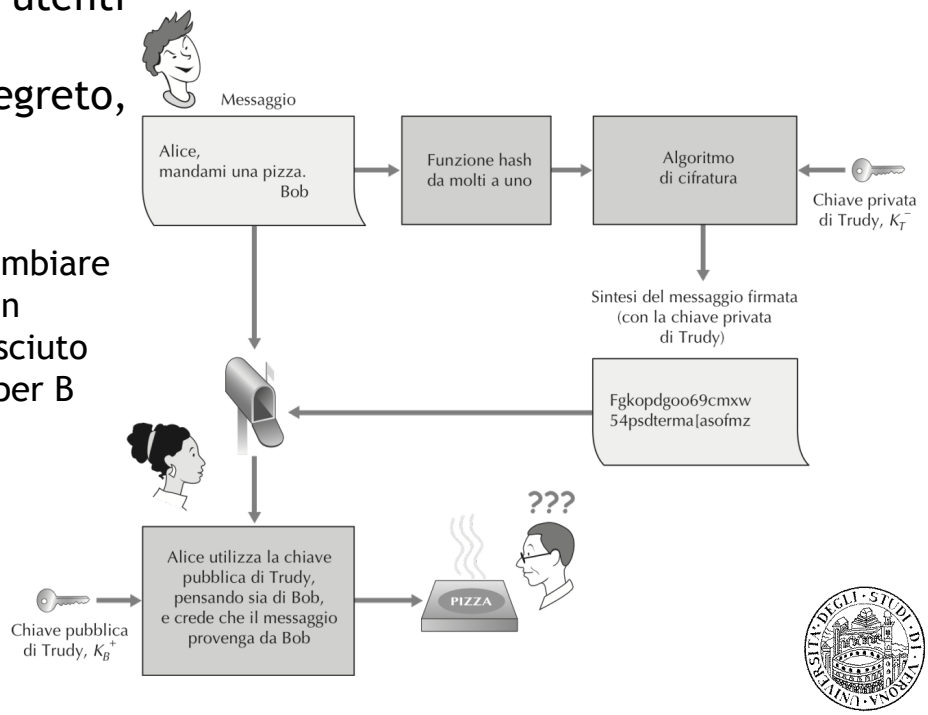


Controllo dell'integrita' del messaggio firmato



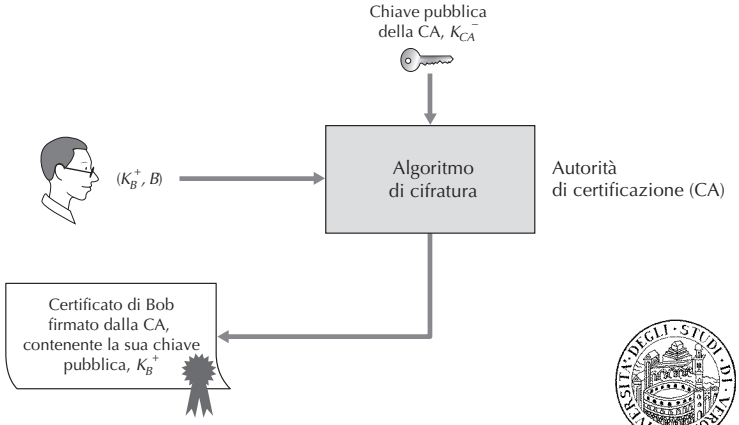
Firma digitale: problemi

- Per quanto due utenti A e B possano scambiare un segreto, non è garantita autenticità
  - A potrebbe scambiare la chiave con un perfetto sconosciuto che si spaccia per B



Autorità di certificazione

- E' necessaria la certificazione della chiave pubblica
  - Utenti, browser, router e così via devono avere la certezza che la chiave pubblica sia proprio quella del corrispondente
- Autorità di certificazione (CA, certification authority)
  - convalida l'identità ed emette certificati
- ITU (international telecommunication union) e IETF (Internet engineering task force) hanno sviluppato standard per le autorità di certificazione



Certificati X.509 e infrastruttura a chiave pubblica

❑ Campi essenziali di un certificato X.509

Nome campo	Descrizione
Versione	Numero di versione della specifica X.509
Numero seriale	Identificatore unico del certificato fornito dalla CA
Firma	Specifica l'algoritmo utilizzato dalla CA per firmare il certificato
Nome dell'emittente	Identificativo della CA che rilascia il certificato, in formato DN [RFC 4514]
Periodo di validità	Inizio e fine del periodo di validità del certificato
Nome del soggetto	Identificativo dell'entità la cui chiave pubblica è associata al certificato (in formato DN)
Chiave pubblica del soggetto	Chiave pubblica del soggetto e indicazioni dell'algoritmo da utilizzare

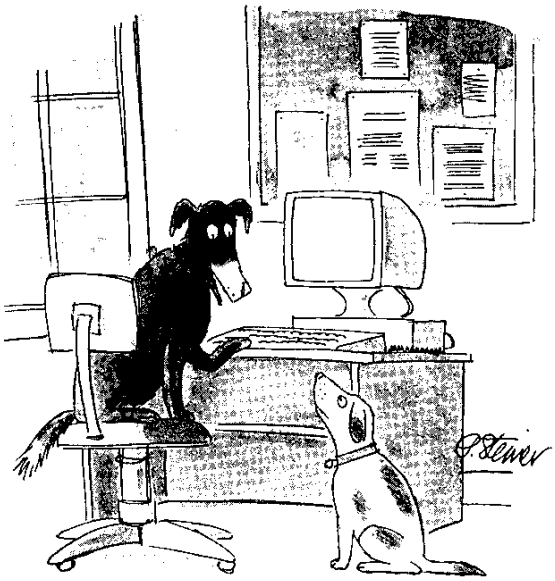
❑ Serve un'infrastruttura per gestire i certificati

- PKI: Public Key Infrastructure
- Gerarchica, per gestire la mole di richieste
- Argomento trattato più avanti

Parte II: Autenticazione

Autenticazione

- L'autenticazione è il servizio di sicurezza che permette di garantire l'identità degli interlocutori



"On the Internet, nobody knows you're a dog."

by Peter Steiner

The New Yorker

5 Luglio 1993

Autenticazione

- Gli interlocutori possono essere utenti o computer
  - computer-computer (stampa in rete, delega,…)
  - utente-utente (protocolli di sicurezza, …)
  - computer-utente (autenticare un server web,…)
  - utente-computer (per accedere a un sistema…)
- Spesso richieste varie combinazioni
- Proprietà primaria
  - Richiesta da un corretto controllo d'accesso

Tipologie di autenticazione

- Si posso distinguere quattro categorie di sistemi di autenticazione:
  - Locale: l'utente accede in locale al servizio, che effettua l'autenticazione
  - Diretta: l'utente accede da remoto al servizio, che effettua direttamente l'autenticazione
  - Indiretta: l'utente accede da remoto a diversi servizi, che si appoggiano su un servizio di autenticazione separato
    - ad es. RADIUS, Kerberos
  - "Off-line": i servizi possono prendere decisioni autonome anche senza dover contattare ogni volta l'autorità di autenticazione
    - ad es. PKI

Fattori di autenticazione

- Basata su qualcosa che l'utente
  - Conosce: segreti
    - password, PIN

  - Possiede: cose fisiche o elettroniche
    - Chiavi convenzionali, carte magnetiche o smart card

  - E': caratteristiche biometriche
    - Impronte digitali, dell'iride, tono di voce

Autenticazione a fattori multipli

- Per ottenere un'autenticazione più forte si possono combinare diversi fattori
  - Qualcosa che si possiede e si conosce

+ PIN

  - Qualcosa che si possiede e si è

+

Qualcosa che si conosce: Username e Password

- Il metodo di autenticazione più semplice è quello basato su username e password
  - L'utente inserisce un nome che lo identifica (lo username), solitamente non segreto, e una parola segreta (la password)
- Vantaggi:
  - Semplice per l'utente
  - Economico
  - Non richiede di immagazzinare un segreto lato client
- Svantaggi:
  - Spesso gli utenti scelgono password deboli
  - Spesso i metodi di autenticazione basati su password sono deboli

Attacchi alle password

- Attacchi possibili:
  - Intercettazione (se la password passa in chiaro)
  - Guessing/cracking
    - si può fare un attacco a pura forza bruta...
    - o più spesso un "attacco a dizionario", provando parole di senso compiuto (o loro minime variazioni)
- Una password dovrebbe essere abbastanza lunga, non essere una parola di senso compiuto, e dovrebbe essere cambiata di frequente
  - Ma questo si scontra con la "comodità" e la "pigrizia" degli utenti...
- Altri attacchi: social engineering, trojan horse...

One-time password

- Con il termine "one-time password" ci si riferisce a sistemi in cui viene generata una nuova password ad ogni accesso da parte dell'utente, per risolvere il problema dell'intercettazione
- Queste password "monouso" vengono generate sulla base di un contatore (esiste quindi una sequenza di password successive) o più spesso sulla base dell'istante temporale
- Spesso i sistemi one-time password si appoggiano
  - su "token", dispositivi hardware che forniscono all'utente la password da inserire
  - SMS inviato su cellulare dell'utente

- Spesso la one-time password viene utilizzata congiuntamente ad un PIN

Qualcosa che si possiede

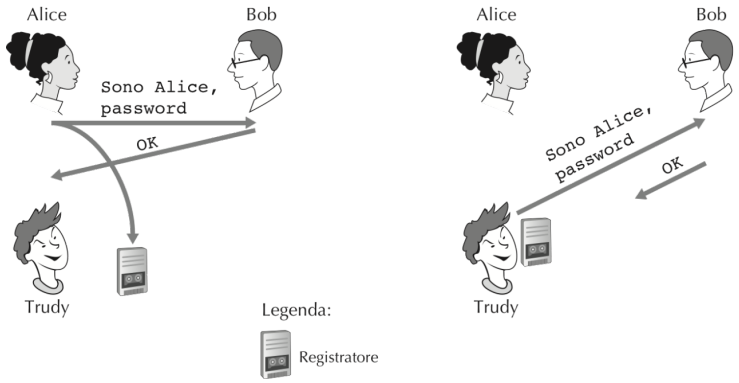
- I token citati precedentemente in realtà appartengono alla categoria dell'autenticazione basata sul possesso
  - Fornisce prova dell'identità
- Altri esempi:
  - Smart card
  - Carte magnetiche
- L'autenticazione dimostra solo l'identità del token, non quella dell'utente
  - Token persi, rubati, clonati
- Spesso si combina possesso e conoscenza
  - Bancomat: carta + PIN

Qualcosa che si è

- Possesso di caratteristiche univoche fornisce prova dell'identità
  - Fisiche: impronte digitali, forma della mano, impronta della retina o del viso, …
  - Comportamentali: firma, timbro di voce, scrittura, "keystroke dynamic",…
- Tecnica recente (in termini di fattibilità) e promettente
- Punti deboli
  - L'autenticazione si basa su una misura e un confronto con un template
    - Misure imprecise, anche quando viene creato il template
    - Possibilità di falsi positivi e falsi negativi
  - Non sostituibili se compromesse
    - Impronte digitali falsificate

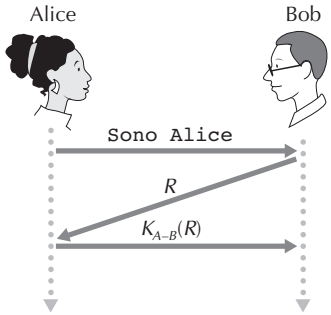
Autenticazione diretta

- Le tipologie di autenticazione viste finora si possono usare senza modifiche per l'autenticazione locale
- Se l'autenticazione avviene da remoto (autenticazione diretta) ci sono altri problemi
  - Un intruso potrebbe registrare e replicare le informazioni



Autenticazione diretta

- Soluzione
  - L'autenticazione avviene su un canale sicuro
  - Oppure si aggiunge una "sfida"

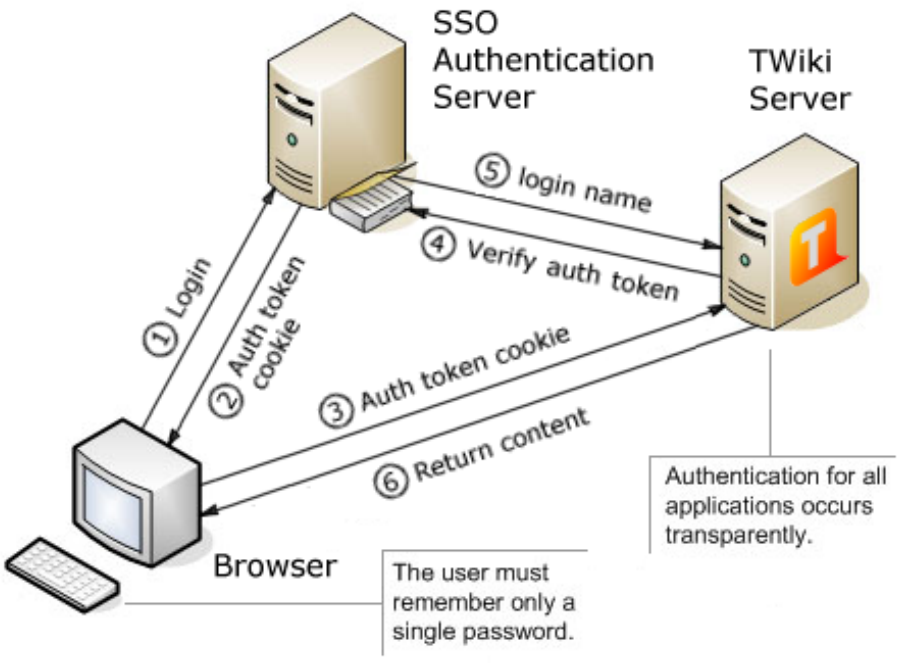


Autenticazione indiretta

- Quando molti sistemi / applicazioni condividono gli stessi utenti, si ricorre spesso a sistemi di autenticazione indiretta
  - Le informazioni sugli utenti vengono centralizzate sul sistema di autenticazione, e gli altri sistemi si appoggiano su di esso

- Due esempi di sistemi di autenticazione indiretta sono:
  - RADIUS(Remote Authentication Dial In User Service), nato per l'accesso remoto dial-up
  - Kerberos, usato per l'autenticazione e il single sign-on (SSO) tra applicazioni all'interno di un "dominio" amministrativo
    - ad esempio all'interno di un'azienda

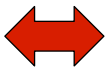
Autenticazione indiretta: Esempio



Autenticazione “off-line”

- Basata su certificati
  - Emessi da una autorità di certificazione (CA)
  - Distribuiti da un’infrastruttura a chiave pubblica
    - Public Key Infrastructure (PKI)

- Certificato reale
  - Cartaceo
    - ad es. Passaporto
  - Emesso da un’autorità riconosciuta
  - Associa l’identità di una persona al suo aspetto fisico



- Certificato digitale
  - Elettronico
  - Emesso da una CA riconosciuta
    - Firmato con la chiave privata della CA
  - Associa l’identità di una persona ad una chiave pubblica

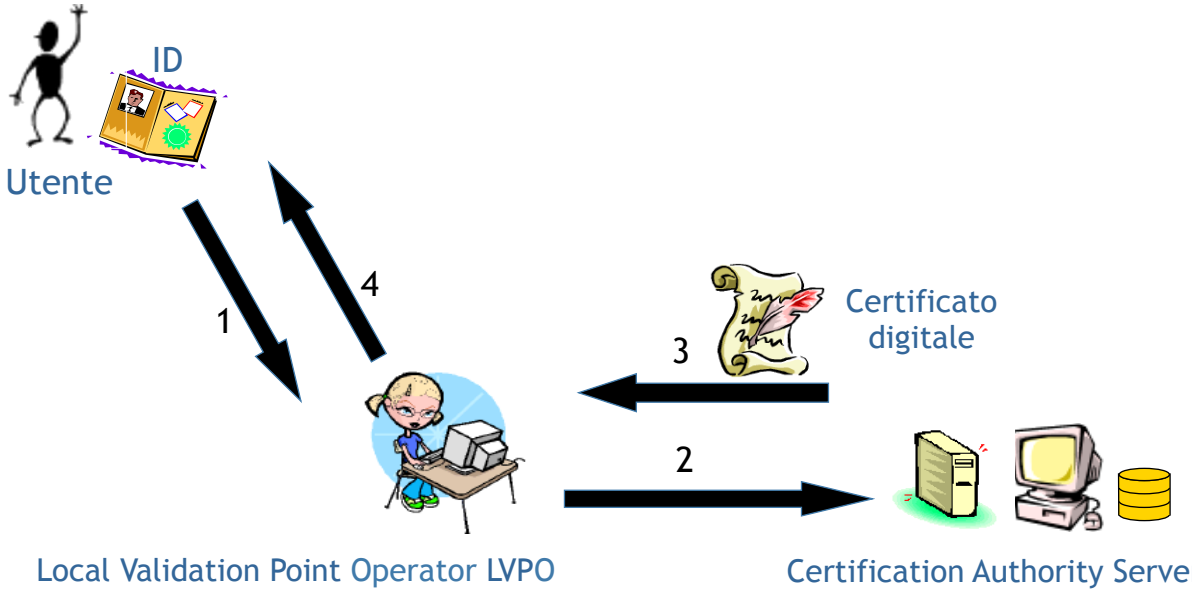
I 10 compiti di una CA

1. Identificare con certezza la persona che fa richiesta della certificazione della chiave pubblica
2. Rilasciare e rendere pubblico il certificato
3. Garantire l'accesso telematico al registro delle chiavi pubbliche
4. Informare i richiedenti sulla procedura di certificazione e sulle tecniche per accedervi
5. Dichiarare la propria politica di sicurezza

I 10 compiti di una CA

6. Attenersi alle norme sul trattamento di dati personali
7. Non rendersi depositario delle chiavi private
8. Procedere alla revoca o alla sospensione dei certificati in caso di richiesta dell'interessato o venendo a conoscenza di abusi o falsificazioni, ecc.
9. Rendere pubblica la revoca o la sospensione delle chiavi
10. Assicurare la corretta manutenzione del sistema di certificazione

Ottenere un certificato digitale



Ottenere un certificato digitale

- L'utente genera sul proprio PC una coppia di chiavi
  - I browser comuni offrono il servizio
  - La chiave privata è memorizzata localmente in un file nascosto
  - Maggiore sicurezza: generare la coppia di chiavi tramite SmartCard collegata al PC - la chiave privata non esce mai dalla SmartCard (protetta da PIN)
- L'utente invia alla CA una richiesta di certificato, insieme alla chiave pubblica generata (a meno che non sia la CA a generare la coppia di chiavi per l'utente)

Ottenere un certificato digitale

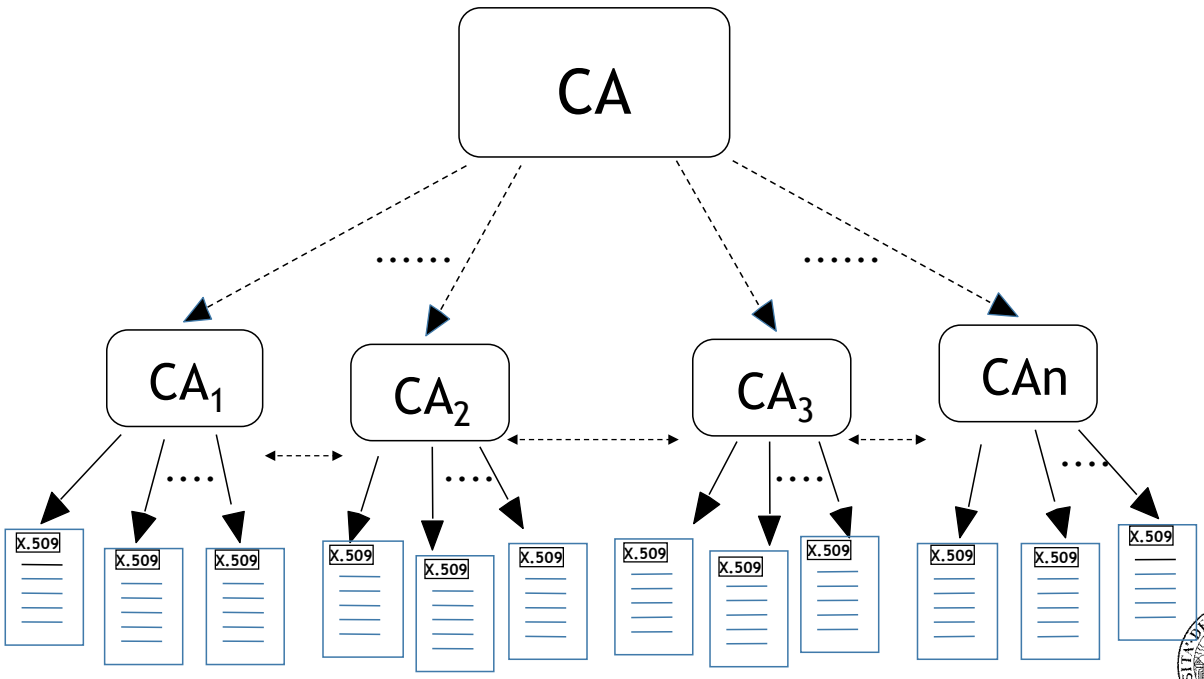
- La CA autentica il richiedente, di solito chiedendogli di recarsi di persona ad uno sportello di LVP (Local Validation Point) collegato con la CA
- Verificata l'identità, la CA emette il certificato, lo invia al richiedente tramite posta elettronica ed inserisce la chiave certificata nel registro delle chiavi pubbliche

L'intera procedura accade nell'ambito di una **PKI**
(Public Key Infrastructure)

PKI (Public Key Infrastructure)

- Struttura minima: CA+LVP
  - Ammesse più LVP
  - LVP è lo sportello per l'autentica classica dell'utente
  - LVPO il suo operatore
- Struttura gerarchica: alcune CA certificano altre, ottenendo una "catena di fiducia"
  - Struttura ad albero
  - La Root CA certifica le CA di primo livello
  - Le primo livello certificano le CA di secondo livello
  - Le CA di ultimo livello certificano il singolo utente

PKI a struttura gerarchica



Certificati: utilizzo

- Bob invia ad Alice il proprio certificato, firmato dalla CA
- Alice verifica la firma della CA sul certificato di Bob, e se è corretta estrae la chiave pubblica di Bob dal certificato
  - Alice deve già avere il certificato della CA, per poterne verificare la firma
  - Il certificato della CA è autofirmato
- A questo punto Alice ha ottenuto la chiave pubblica di Bob, la cui identità è garantita dalla CA

Certificati: problemi

- È comunque necessario ottenere in qualche modo sicuro il certificato della CA
  - Il problema della distribuzione delle chiavi pubbliche rimane, ma su scala molto più ridotta
- Un certificato può essere revocato, ad esempio se il proprietario si accorge del furto della chiave privata corrispondente
  - Ma la verifica della firma della CA su un certificato revocato va a buon fine!
  - la CA pubblica una lista dei certificati revocati, da essa firmata, che andrebbe controllata per accertarsi della validità di un certificato
- Il sistema implica una fiducia nella CA, ma... chi lo garantisce?

Problemi comuni a tutti gli schemi

- Alice vuole una propria coppia di chiavi pubblica/privata, ma...
  - Come la genera?
  - Come custodisce la chiave privata, in modo che nessuno se ne impossessi?
  - Come trasporta la chiave privata, in modo da poterla utilizzare in qualunque luogo?

- La sicurezza dipende molto anche dagli utenti

Parte III: Autorizzazione

Autorizzazione (Controllo degli accessi)

- Il servizio di controllo dell'accesso (detto anche autorizzazione) garantisce che l'accesso alle risorse sia limitato ai soli utenti che ne hanno diritto
- Soggetti diversi possono avere diritto a diverse modalità di interazione con le risorse
- I soggetti sono in possesso di privilegi sugli oggetti in accordo con le politiche definite sul sistema
  - Soggetti: utenti, applicazioni, altri sistemi...
  - Privilegi: lettura, scrittura, esecuzione, proprietà
  - Oggetti: file, funzioni, applicazioni, altri sistemi...

Controllo degli accessi

- Le politiche di controllo dell'accesso definiscono l'attribuzione dei privilegi di accesso dei soggetti sugli oggetti
- I meccanismi di controllo dell'accesso specificano come le relazioni tra i soggetti e gli oggetti (i privilegi) sono rappresentate
- Due principi utili:
  - Privilegio minimo
    - Ad un soggetto dovrebbero essere concessi solo i privilegi minimi necessari a compiere l'azione che deve compiere
  - Separazione dei compiti
    - Nessun soggetto dovrebbe avere abbastanza potere per sovvertire il sistema

Meccanismi di controllo dell'accesso

- Matrice di controllo dell'accesso
  - Le righe contengono i soggetti, le colonne gli oggetti, nelle caselle sono rappresentati i permessi

	File 1	File 2	Progr.1	Progr.2
Alice	`rwx`	`rwx,own`	`x`	`rwx`
Bob	`rwx,own`	`r`	`x`	`rwx`
Progr.1	`rw`	`rw`	`-`	`x`

- Soggetti e oggetti possono essere molto numerosi, la matrice sparsa, sorgono problemi di scalabilità
  - Si può memorizzare la matrice per righe/colonne
  - Si possono effettuare raggruppamenti per gestire i privilegi in modo "omogeneo" (gruppi/ruoli)

Access Control List (ACL)

- Si memorizza la matrice di controllo dell'accesso per colonne
- Ciascuna risorsa viene memorizzata con la lista dei soggetti che possono interagire con questa, e con i relativi permessi
  - Per esempio il filesystem Unix
- Sono adatte a contesti in cui la protezione è orientata ai dati
  - È semplice gestire i permessi associati a un oggetto
  - Non sono adatte se si vogliono gestire centralmente i permessi di ciascun soggetto e/o se si vogliono introdurre meccanismi di delega temporanea

Capabilities

- Si memorizza la matrice di controllo dell'accesso per righe
- A ciascun soggetto è associato l'insieme degli oggetti con cui può interagire, si memorizza la lista di relazioni che il soggetto ha con gli oggetti e i relativi permessi
  - Permette di gestire in modo efficiente i permessi associati a un singolo utente
    - Rende semplice anche un meccanismo di delega temporanea
  - Per determinare tutti i soggetti che hanno diritto di interagire con un oggetto si deve scorrere la lista

Politiche di controllo dell'accesso

- Si distinguono due diversi approcci:
  - DAC (Discretionary Access Control)
  - MAC (Mandatory Access Control)
- Gli approcci possono essere combinati con una suddivisione degli utenti in gruppi e ruoli
  - Un gruppo è una lista di soggetti
  - Un ruolo è un insieme prefissato di permessi di accesso che uno o più soggetti possono acquisire per un certo periodo di tempo, spesso corrispondente a una funzione all'interno di un'organizzazione

DAC

- In un modello DAC i singoli utenti possono a loro discrezione concedere e revocare permessi su oggetti che sono sotto il loro controllo
  - In genere ci si basa sul concetto di proprietà (ownership): ogni oggetto ha un proprietario, cioè il soggetto che ne definisce i diritti di accesso
    - Eventualmente, il proprietario può assegnare la proprietà a un altro soggetto
- DAC è flessibile, utilizzabile in molti ambiti (per esempio i sistemi operativi Unix, Windows...)
- DAC non permette di controllare la diffusione dell'informazione (chi ha permessi in lettura su un file potrebbe inviarlo a chi non ha permessi)

MAC

- In un modello MAC la politica di controllo dell'accesso è determinata centralmente dal sistema, non dai singoli utenti
- Utilizzato per esempio in ambito militare, basandosi su una classificazione degli oggetti e dei soggetti
  - Ad esempio: TopSecret, Secret, Confidential, Classified
- Per esempio nel modello di Bell-LaPadula il sistema forza il rispetto delle seguenti regole:
  - No read up: non è possibile leggere informazioni classificate a livelli più alti del proprio.
  - No write down: non è possibile scrivere informazioni classificate a livelli più bassi del proprio.
- Meno flessibile ma più robusto del modello DAC

Gruppi e ruoli

- I gruppi permettono di gestire insiemi di soggetti/oggetti in modo omogeneo
  - Ciò semplifica l'associazione dei permessi agli utenti
    - L'accesso alle risorse è basato sui permessi dei gruppi.
    - Diversi gruppi possono avere proprietà sovrapposte
    - Modificare i diritti di un gruppo permette di cambiare direttamente quelli di tutte le entità appartenenti
- I ruoli definiscono insiemi di proprietà e responsabilità solitamente associate alla struttura organizzativa a cui fa capo il sistema
  - RBAC (Role Based Access Control)