CONTROLLI DEGLI ACCESSI E AUTENTICAZIONE

Vincenzo Calabrò

Sicurezza Informatica: definizione informale

Sicurezza Informatica?

- **Prevenzione** o **protezione** contro accesso, distruzione o alterazione di risorse/informazioni da parte di utenti **non autorizzati**

È sufficiente?

Sicurezza (Schneier 2000)

"***La sicurezza non è un prodotto, ma un processo***"

- Concetto mai assoluto – qual è il contesto?
- Sicurezza da che cosa?
- Che livello di sicurezza si vuole garantire?

"***La sicurezza è una catena e la sua resistenza è determinata dall'anello più debole***"

La sicurezza informatica non è …

… non è crittografia

Crittografia *scienza esatta* come branca della matematica

- *Impossibile violare RSA in tempo polinomiale*

Sicurezza *scienza inesatta* perché basata su persone e macchine

- *Acquisto on-line potenzialmente insicuro*

… non è password

Sistema molto debole!

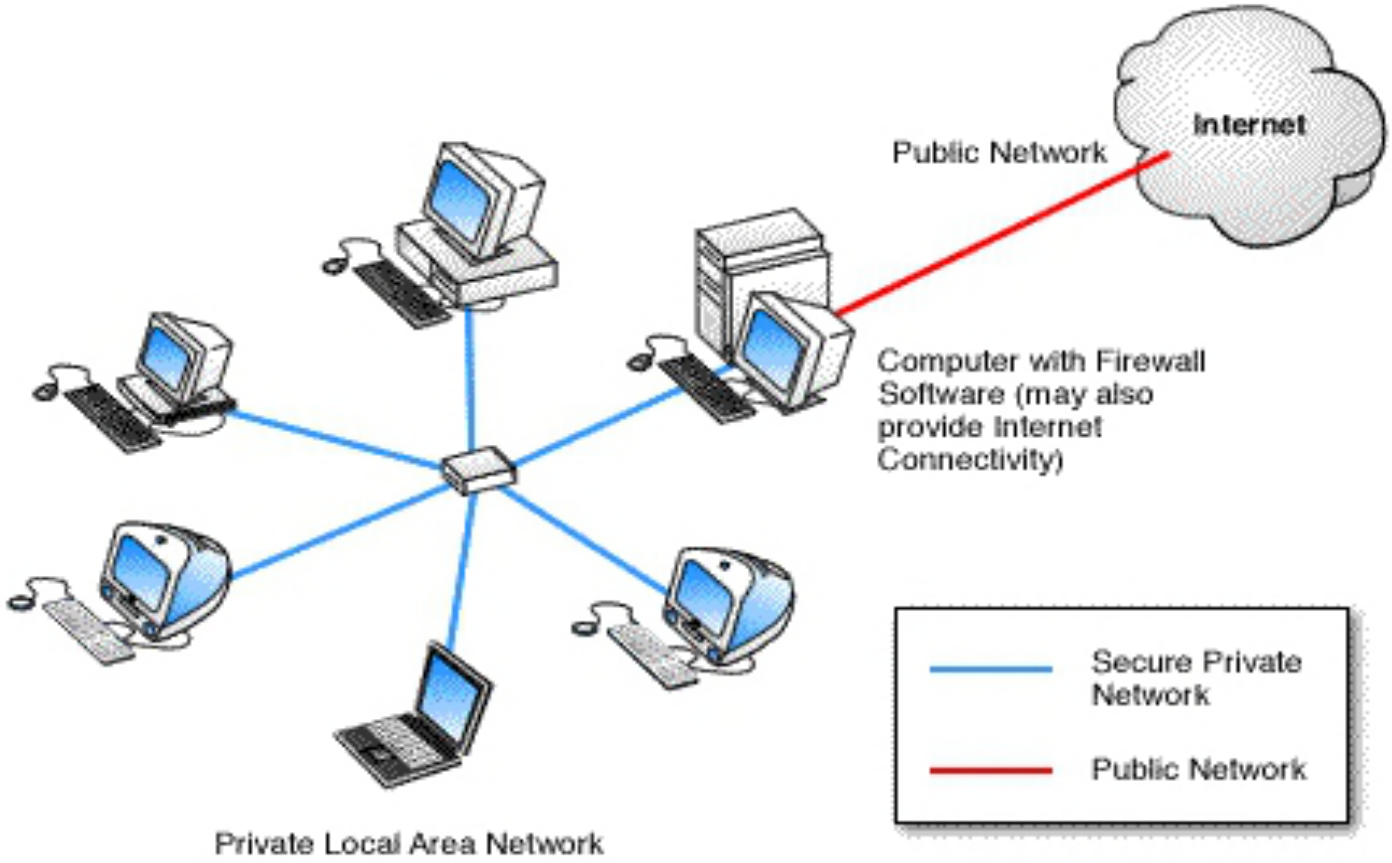
La password più diffusa è *love*

- Attacchi dizionario
- Attacchi a forza bruta
- Ottenere l’accesso al file delle password

Problemi:

- Come scegliere una buona password?
- Come ricordare una buona password?
- Usare una password per sempre?

… non è firewall



Sicurezza: stato dell'arte

La sicurezza:

- E' una proprietà di vari livelli architetturali (SO, rete, applicativo, …)
- Non è un semplice predicato booleano!
- È **costosa** nel senso di risorse computazionali, gestione, mentalità e utilizzo
- Rimane un campo aperto anche per i colossi dell'informatica
- Richiederebbe spesso il ridisegno di un sistema preesistente, il che non è sempre possibile

Come proteggersi: progettazione della sicurezza

Pianificazione della sicurezza (1)

Prevenzione

- Utilizzare tecnologie che rendano il sistema non vulnerabile
  - Crittografia
  - Backup
- Fare in modo che solo gli utenti autorizzati accedano alle risorse
  - Uso di badge

Rilevamento

- Verificare che il sistema funzioni come previsto
  - Uso di file di log

Reazione

- In caso si subisca un attacco, rilevare l'attacco in tempo reale ed avere un "piano B" pronto

Pianificazione della sicurezza (2)

Esempio:

- Pianificazione della rete con hardware adeguato (router, switch ecc.) insieme alla divisione della rete in aree a livello di sicurezza variabile.
- Controllo dell'integrità delle applicazioni (bugs free) e verifica della correttezza delle configurazioni.
- Utilizzo di software che controllino e limitino il traffico di rete dall'esterno verso l'interno e viceversa (es. firewall, router screening ecc.)
- Utilizzo di applicazioni che integrino algoritmi di crittografia in grado di codificare i dati prima della loro trasmissione in rete (es. PGP, SSH, SSL ecc.).

I trade-off della SI (1)

Non esiste sicurezza in senso assoluto:

- La ***prevenzione*** può:
  - Ridurre la possibilità che abbia luogo una violazione
  - Ridurre i danni che una violazione può portare
  - Ma spesso l'attacco avviene bypassando il modulo che si occupa di prevenzione!
- Il ***rilevamento*** di attacchi alla sicurezza:
  - Non sempre è possibile in tempo reale
  - Funziona soprattutto per attacchi già noti

I trade-off della SI (2)

La *sicurezza* ha un *costo* in termini di:

- Acquisto, applicazione e gestione di misure aggiuntive
- Aumento del carico del sistema (diminuzione di prestazioni)
- Accettazione dell'utente

La prevenzione va bilanciata rispetto al valore delle risorse da proteggere ed al danno che una violazione a queste porterebbe

Sicurezza Informatica: definizione informale

Sicurezza Informatica?

- **Prevenzione** o **protezione** contro accesso, distruzione o alterazione di risorse/informazioni da parte di utenti **non autorizzati**

Essenziali anche:

- **Rilevamento**
- **Reazione**

Pianificazione della sicurezza (1)

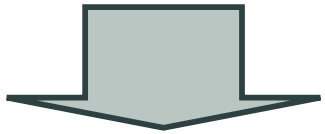
Prevenzione, rilevamento e reazione

Per *prevenire* intanto si deve specificare:

- Da chi (o cosa) ci si vuole proteggere
- Quali proprietà di sicurezza devono essere soddisfatte dal sistema
- Quali sono gli utenti autorizzati e quali non
- Che livello di segretezza hanno le risorse
  - *Top secret, Secret, Riservato, Non classificato*
- ……

Pianificazione della sicurezza (2)

1. Definire delle regole di accesso e controllare che chi accede alle risorse soddisfi tali regole
   - politiche e meccanismi



CONTROLLO DEGLI ACCESSI

2. Verificare che il sistema funzioni come previsto

3. Trasmettere i dati che devono rimanere confidenziali non in chiaro
   - crittografia e protocolli crittografici

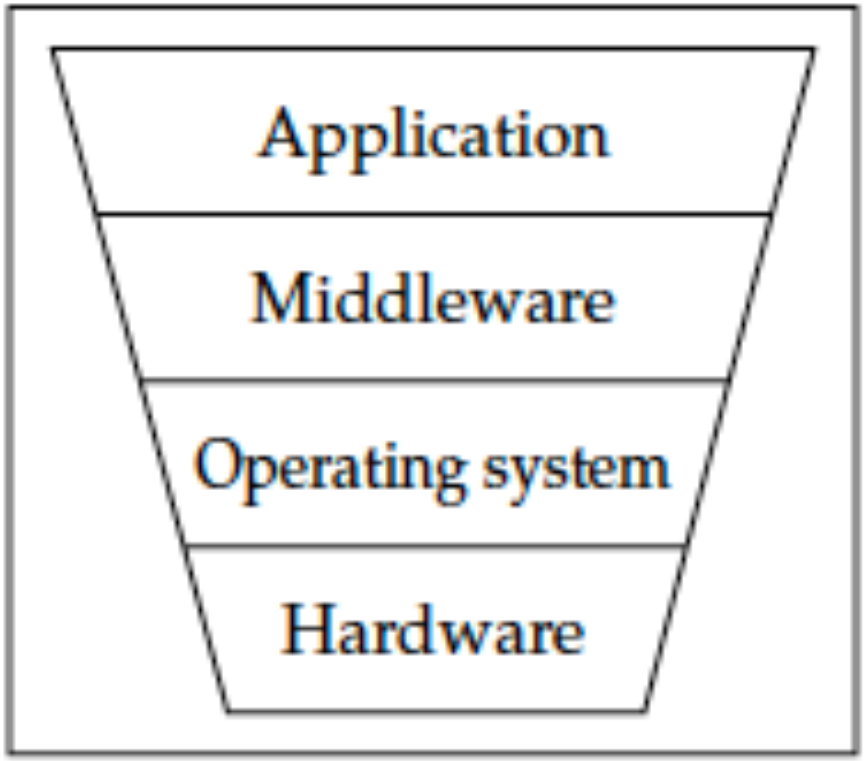
Pianificazione della sicurezza (3)

Controllo degli accessi:

1. Definire una ***politica di sicurezza*** che limiti quali operazioni gli utenti possono fare sulle risorse
2. Scegliere un ***meccanismo*** per far rispettare la politica
3. Verificare che sia la politica che il meccanismo siano valide e coerenti

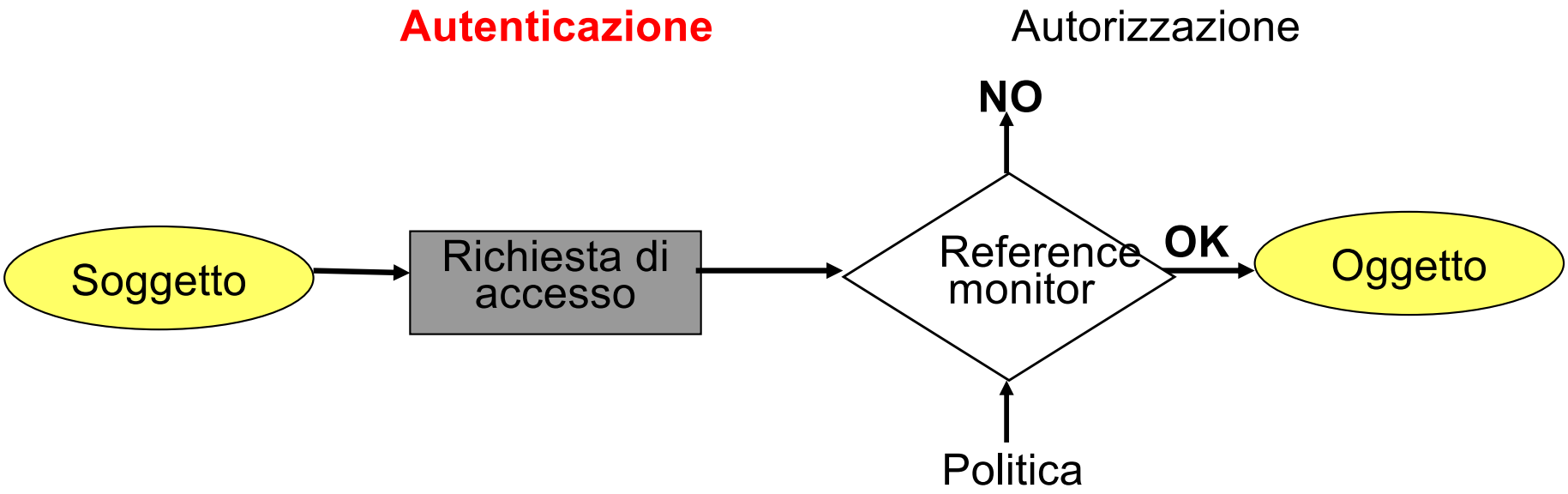
Pianificazione della sicurezza (4)

A che livello opera il controllo degli accessi?



- HW: a quali parti di memoria può accedere un *processo*
- SO: accesso a *file* o altre risorse (e.g., porte)
- Middleware e Application: politiche di sicurezza più complesse

Pianificazione della sicurezza (5)



- Valido a tutti i livelli
- A seconda del livello il soggetto può essere un utente, un processo, etc. e l'oggetto un dato, una risorsa, un file, la memoria, etc.
- Il sistema deve garantire che tutte le richieste passino attraverso il *reference monitor* (una specie di guardiano)

Politiche vs Meccanismi (1)

- Politiche: regole ad alto livello che descrivono gli accessi autorizzati al sistema
  - Un insieme di proprietà di sicurezza
  - Dipendono dall'applicazione da proteggere
  - Esempio: in una banca
    - *Autenticazione* dei clienti agli sportelli, ATM e su web
    - *Non-repudiation* delle transazioni
    - *Integrità* dei conti correnti dei clienti
    - *Segretezza* dei clienti e dei dati interni
    - *Disponibilità* di un sistema di allarme
    - *Separation of duties* (nessun conflitto di interessi)

- Meccanismi: funzioni di basso livello (hardware o software) che implementano le politiche

Politiche vs Meccanismi (2)

La separazione di politiche e meccanismi permette:

- analisi dei requisiti di accesso indipendentemente dall'implementazione
- confronto tra diverse politiche per il controllo degli accessi
- confronto tra diversi meccanismi che implementano la stessa politica
- sviluppo di meccanismi che implementano più politiche

Politiche di Sicurezza

Le politiche possono venire descritte in:

- ***Linguaggio naturale*** (semplice da capire ma di solito impreciso)
- ***Matematica*** (precisa, ma difficile da capire)
- ***Linguaggio ad hoc***, una specie di pseudo-codice che cerca di bilanciare la precisione con la facilità di comprensione

Composizione di politiche

- Se le politiche sono in conflitto, le discrepanze possono diventare una vulnerabilità

Controllo degli Accessi - Terminologia

Gli elementi sono:

- Soggetti, entità attiva, utente, processo o macchina, che richiede l'accesso
- Oggetti, entità passiva, file o risorsa
- Operazioni di accesso (o ***modalità di accesso***) di soggetti su oggetti
  - Varia da accessi alla memoria (read/write) a chiamate di funzioni in sistemi orientati agli oggetti

- Permesso (o diritto di accesso): possibilità di accedere ad una risorsa secondo una certa modalità
- Privilegio: in genere un insieme di permessi dati direttamente ad un ruolo specifico come l'amministratore, l'operatore, ecc.

Controllo degli accessi - Modalità di accesso

A livello elementare:

- Osservare un oggetto
- Alterare un oggetto

In genere:

- A seconda del *tipo di oggetto*, ci possono essere diverse modalità di accesso:
  - **`read/write/append`**: vedere/cambiare i contenuti di un oggetto
  - **`execute`**: eseguire un processo
  - **`select`**, **`insert`**, **`update`**, **`delete`**: operazioni dell'ambito di un DBMS

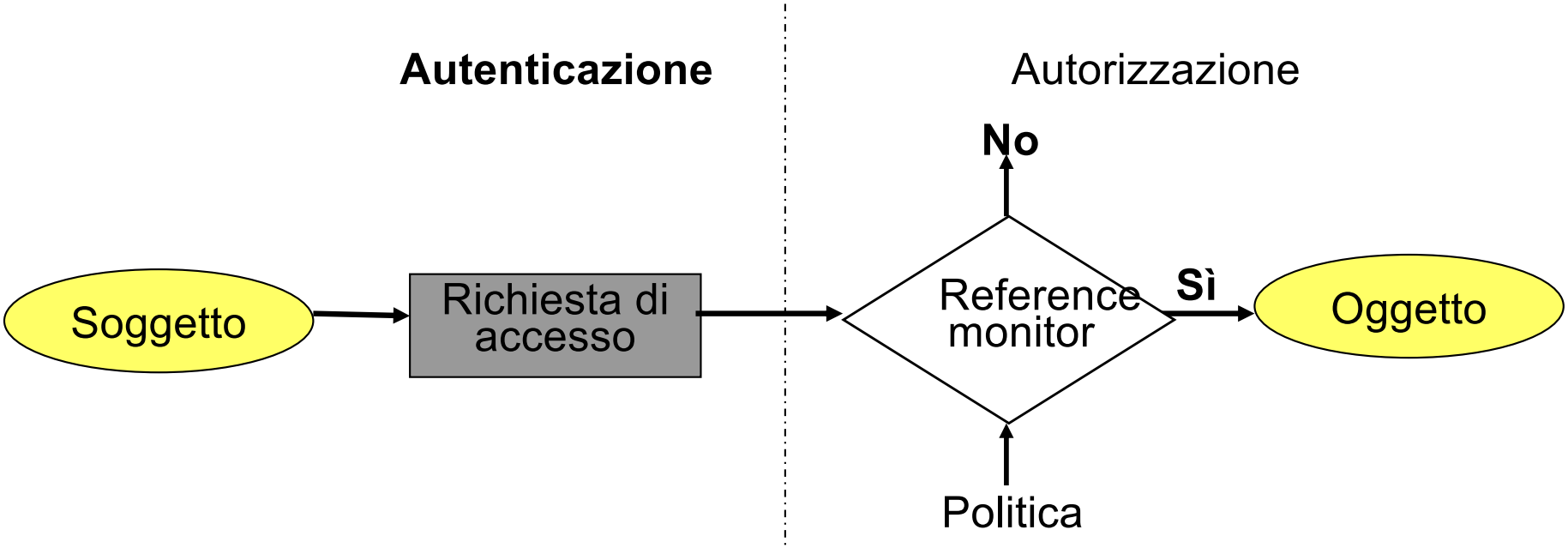
Sistema per il Controllo degli Accessi

- Un sistema di controllo degli accessi *regola le operazioni che possono venire eseguite sui dati e sulle risorse che devono essere protette*
- L'obiettivo è di controllare le operazioni eseguite dai soggetti per *prevenire azioni che potrebbero danneggiare i dati o le risorse*
- Il controllo in genere viene fatto dal sistema operativo
  - **Reference monitor**: macchina astratta che applica il controllo degli accessi, una guardia che media tutte le richieste di accesso e decide come rispondere alle richieste provenienti dai soggetti

Reference Monitor

Funzionamento:

- Un *soggetto* attivo *chiede accesso* ad un *oggetto* passivo per eseguire una specifica *operazione di accesso*
- Il *monitor* concede o meno l'accesso



Correttezza del controllo dell'accesso implica:

- Corretta *identificazione/autenticazione*
- Corretta *definizione* (e implementazione) della *politica di sicurezza*

Autenticazione vs Autorizzazione

- Autenticazione: il reference monitor verifica **l'identità** dell'utente che fa la richiesta

- Autorizzazione: il reference monitor decide se concedere o meno **l'accesso**

Politiche per il controllo degli accessi

Chi stabilisce la politica?

La responsabilità di stabilire una politica può venire assegnata a:

- Al proprietario della risorsa, che può decidere a chi è permesso l'accesso
  - Politiche discrezionarie in quanto il controllo degli accessi è a discrezione del proprietario
- Ad una politica a livello di sistema che stabilisce gli accessi
  - Tali politiche sono chiamate mandatorie

Attenzione:

- Esistono altre interpretazioni dei termini discrezionario e mandatorio

Discretionary Access Control (DAC)

Il proprietario della risorsa:

- Decide gli accessi: chi può accedere e in quale modalità
- Ha l'autorità di passare i propri privilegi ad altri utenti
  - *Delegation of duty*

Controllano solo gli accessi diretti:

- Nessun controllo su cosa accade all'informazione una volta che vi è stato l'accesso
- Vulnerabile a Trojan horse, un programma di utilità che contiene codice nascosto che esegue funzioni non legittime

DAC e Trojan horse

s_2 ottiene accesso ai dati di s_1

- s_1 non se ne accorge
- il meccanismo DAC è rispettato

Mandatory Access Control (MAC)

Il sistema impone una politica multilivello

- Gli accessi vengono gestiti in modo **centralizzato**
- Vengono assegnati livelli di sicurezza a soggetti e oggetti
  - Gli utenti non hanno controllo sul livello di sicurezza che viene loro assegnato
- La modalità di accesso è un attributo dell'oggetto
  - Impone restrizioni sul flusso di informazione
  - Non sono possibili attacchi di tipo Trojan horse

Storicamente 2 tipi di politiche:

- *Secrecy* based → confidenzialità
- *Integrity* based → integrità

Modello Bell-La Padula

Risorse e soggetti vengono *classificati* in livelli di sicurezza

- Livello del soggetto: autorizzazione/livello di fiducia associato all'utente
- Livello dell'oggetto: sensibilità dell'informazione

Chiamata anche *politica multi-livello*

- I livelli possono formare un reticolo
- Es. di livelli:
  - Trusted/Untrusted
  - Public/Secret/Top Secret

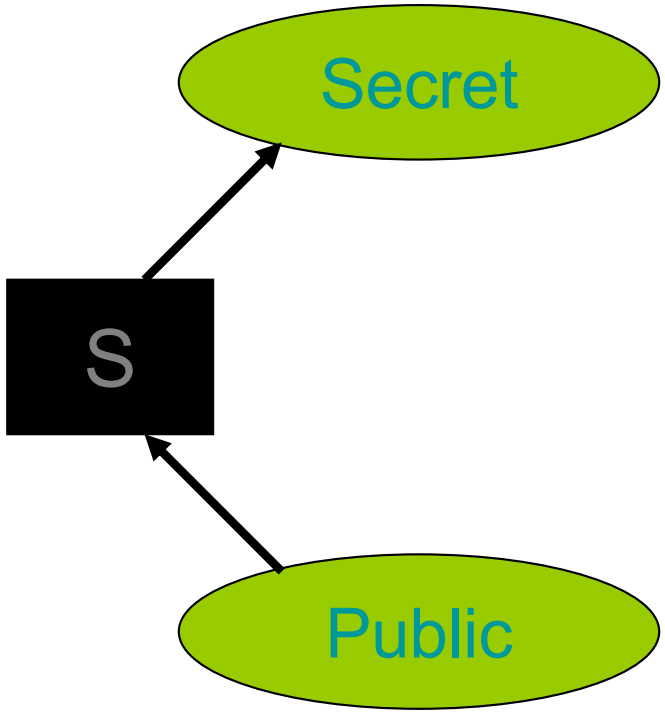
Le regole imposte sono:

- **No read-up** — Garantisce la segretezza
- **No write-down** — Previene il downgrading dell'informazione

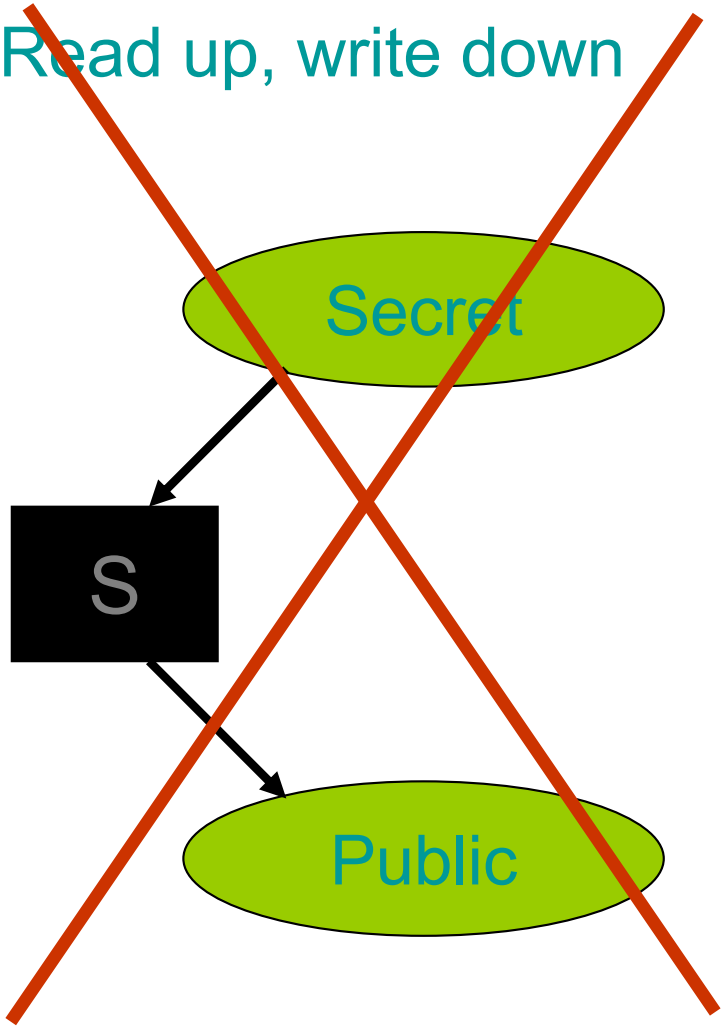
Usato per garantire segretezza

Confidenzialità

Read down, write up



Read up, write down



Biba Model

Risorse e soggetti vengono *classificati* in livelli di integrità

- Livello del soggetto: livello di fiducia associato al soggetto (quanto fidato sia l'utente)
- Livello dell'oggetto: sensibilità dell'informazione e fiducia nella validità dell'informazione

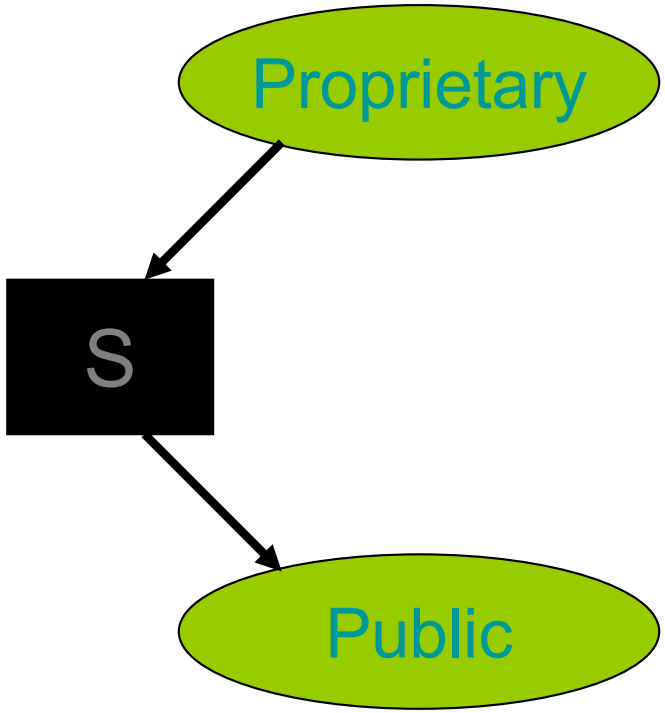
Le regole sono:

- **No write-up**
- **No read-down**

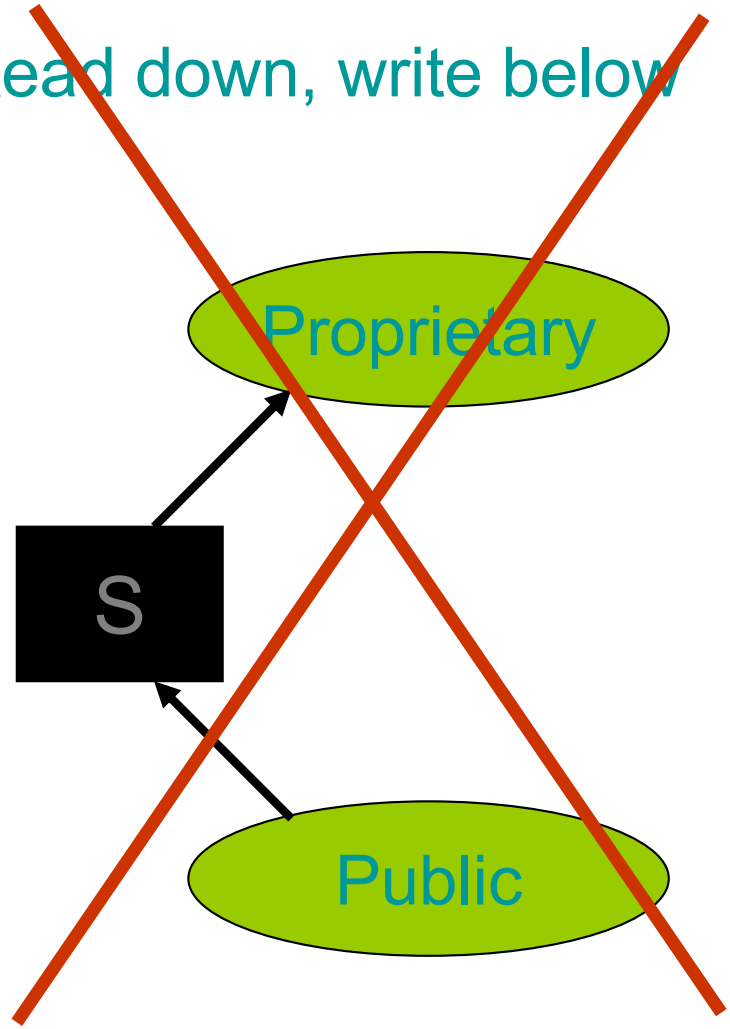
Usato per garantire integrità

Integrità

Read up, write down



Read down, write below



Role-Based Access Control

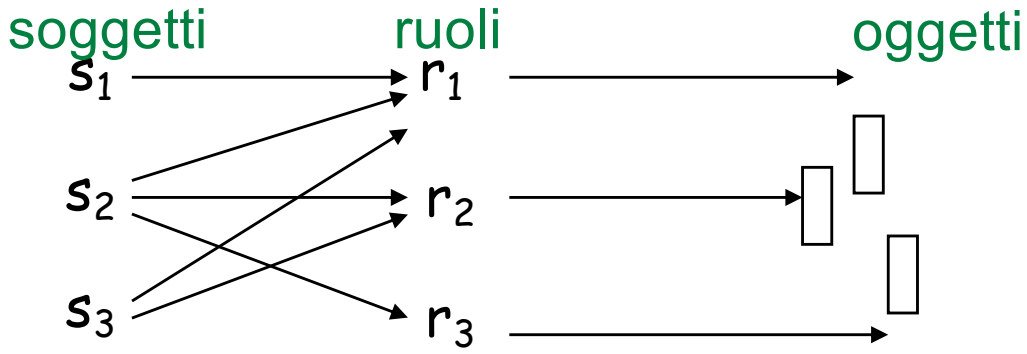
Basato sull'idea che uno stesso soggetto può aver bisogno di permessi diversi a seconda dell'attività (*ruolo*) che svolge

- Tom come system administrator (`root`)
- Tom come utente `tommy`
- Tom come consulente della banca A
- Tom come consulente della compagnia B

} Ruoli

L'accesso agli oggetti è mediato dai ruoli

- *s* con ruolo *r* ha tutti i permessi associati al ruolo *r*



Vantaggi di RBAC

Supporta:

- Facile la revoca di un permesso
- *Separation of duty*
- Gerarchie di ruoli
- Anonymity (parziale)
- *Least privilege*

NB:

- Gruppo: insieme di utenti
- Ruolo: insieme di permessi/privilegi

Principio del Privilegio Minimo

Principle of Least Privilege

- Un soggetto/sistema dovrebbe sempre avere l'insieme minimo di privilegi necessari per svolgere il suo compito/attività

Cos'è un privilegio?

- Abilità di accedere o modificare una risorsa

Esempi?

Autorizzazione e controllo degli accessi

Sicurezza in termini di…

Autenticazione

- Chi sei?

Autorizzazione (controllo degli accessi)

- Cosa puoi fare?
- Si basa su di una politica

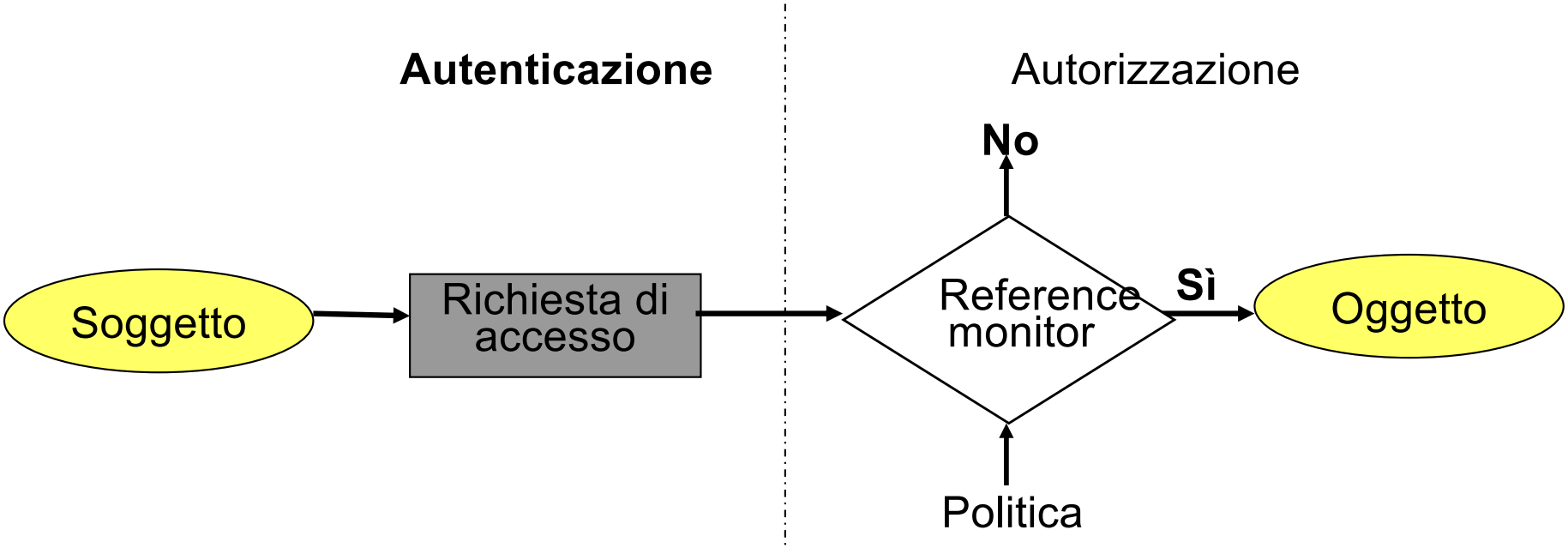
Enforcement mechnism

- Come viene implementata la politica?

Reference Monitor

Funzionamento:

- Un *soggetto* attivo *chiede accesso* ad un *oggetto* passivo per eseguire una specifica *operazione di accesso*
- Il *monitor* concede o meno l'accesso



NB: Differenza tra

- Politica di sicurezza
- Meccanismo

Esempio: Accesso ad un Night Club (1)

Autenticazione

- Controllo documenti

Politica-Controllo degli accessi

- Maggiorenni (>18) - possono entrare
- > 21 - possono bere
- Sulla lista VIP – possono accedere all'area VIP

Enforcement Mechanism

- Muri, porte, buttafuori, luchetti

Esempio: Accesso ad un Night Club (2)
Altre considerazioni

Biglietti:

- Anonimi o con indicato il nome?
- Con o senza data?

Come fare se uno vuole uscire e poi tornare?

Quindi…

Riusciamo a definire dei modelli logici per la gestione degli accessi?

Primo approccio: distinguiamo chi stabilisce la politica

- DAC (Discretionary Access Control)
  - il proprietario della risorsa
- MAC (Mandatory Access Control)
  - il sistema

MultiLevel Security (MLS) - 1

Utilizzata quando soggetti e oggetti di livelli di sicurezza diversi usano lo stesso sistema

Il livello di sicurezza serve per classificare le informazioni

- specificato come *L(O)* and *L(S)*
- in ambito militare: TOP SECRET >SECRET > CONFIDENTIAL >UNCLASSIFIED
- In ambito lavorativo:
  - Informazione ristretta all'amministratore delegato, all'amministratore delegato e al suo vice, all'amministratore delegato, al suo vice e alla sua segretaria, ecc.

MultiLevel Security (MLS) - 2

Modello Bell-LaPadula – Confidenzialità

- Obiettivo: prevenire letture non autorizzate

Regole:

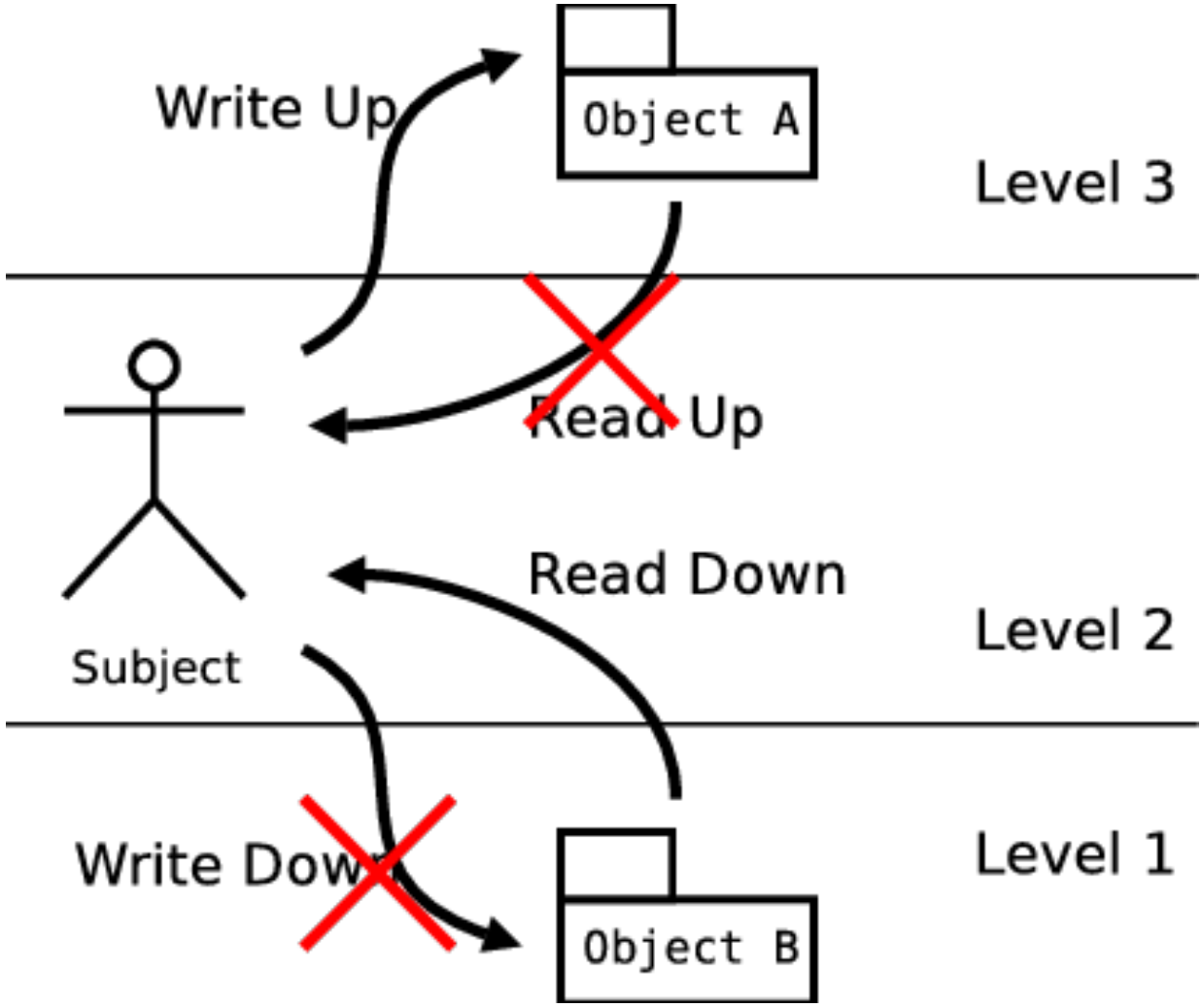
- *Simple security property*
  - un soggetto *S* può leggere un oggetto *O* solo se $L(O) \leq L(S)$
- **-Property (Star Propeerty)*
  - un soggetto *S* può scrivere un oggetto *O* solo se $L(S) \leq L(O)$

In altri termini:

- No read up
- No write down

MultiLevel Security (MLS) - 3

Modello Bell-LaPadula – Confidenzialità



MultiLevel Security (MLS) - 4

Modello Biba - Integrità

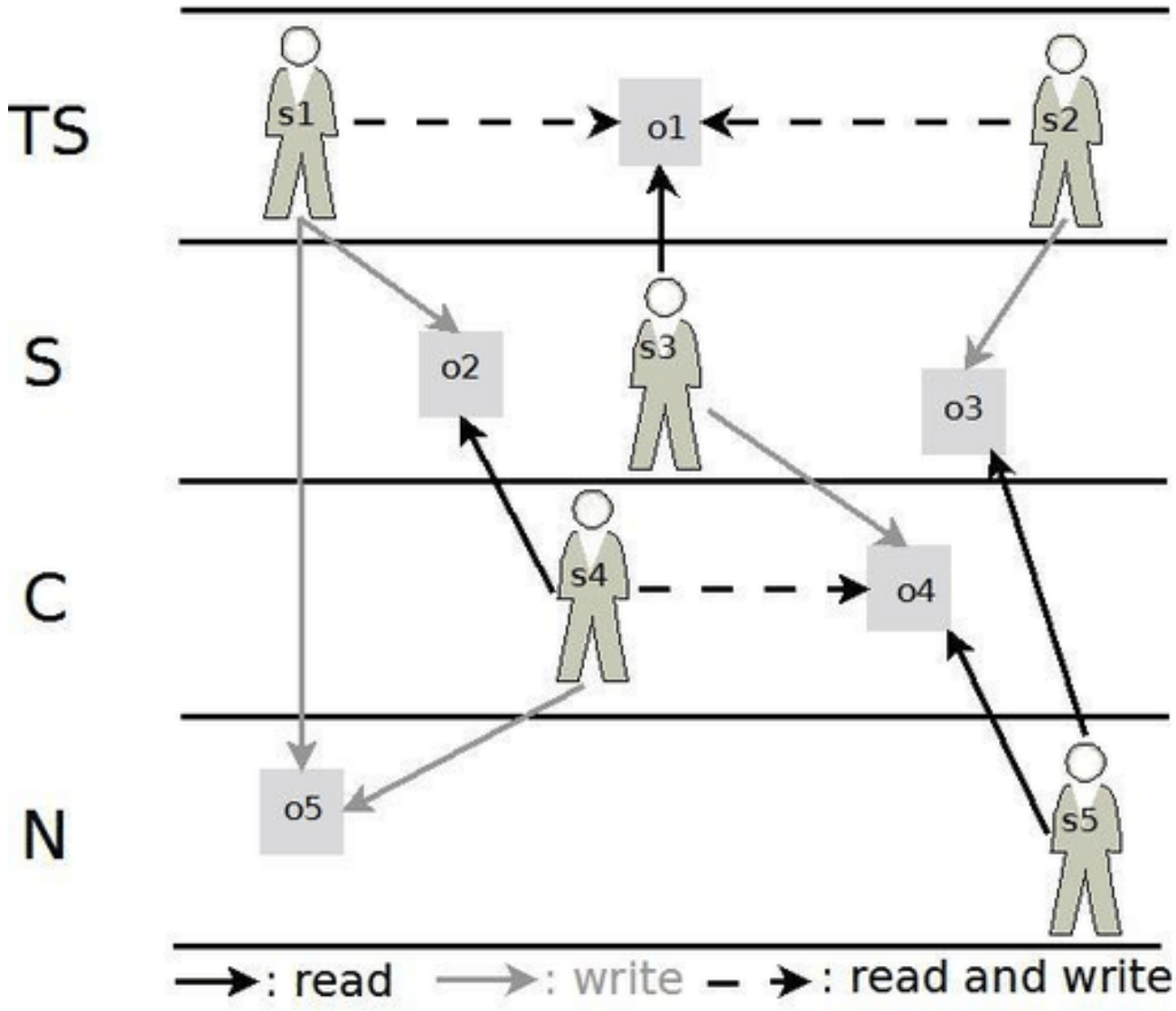
- Obiettivo: preservare l'integrità delle informazioni

Regole:

- *Simple integrity property (No Write up)*
  - un soggetto *S* può scrivere un oggetto *O* solo se $L(S) \geq L(O)$
  - *S* può modificare *O* solo se ha un rango superiore (o uguale), altrimenti meglio non fidarsi…
- **-Property (Star Propeerty) (No Read Down)*
  - un soggetto *S* può scrivere un oggetto *O* solo se $L(S) \leq L(O)$
  - *Meglio "copiare" informazioni solo da oggetti più fidati*

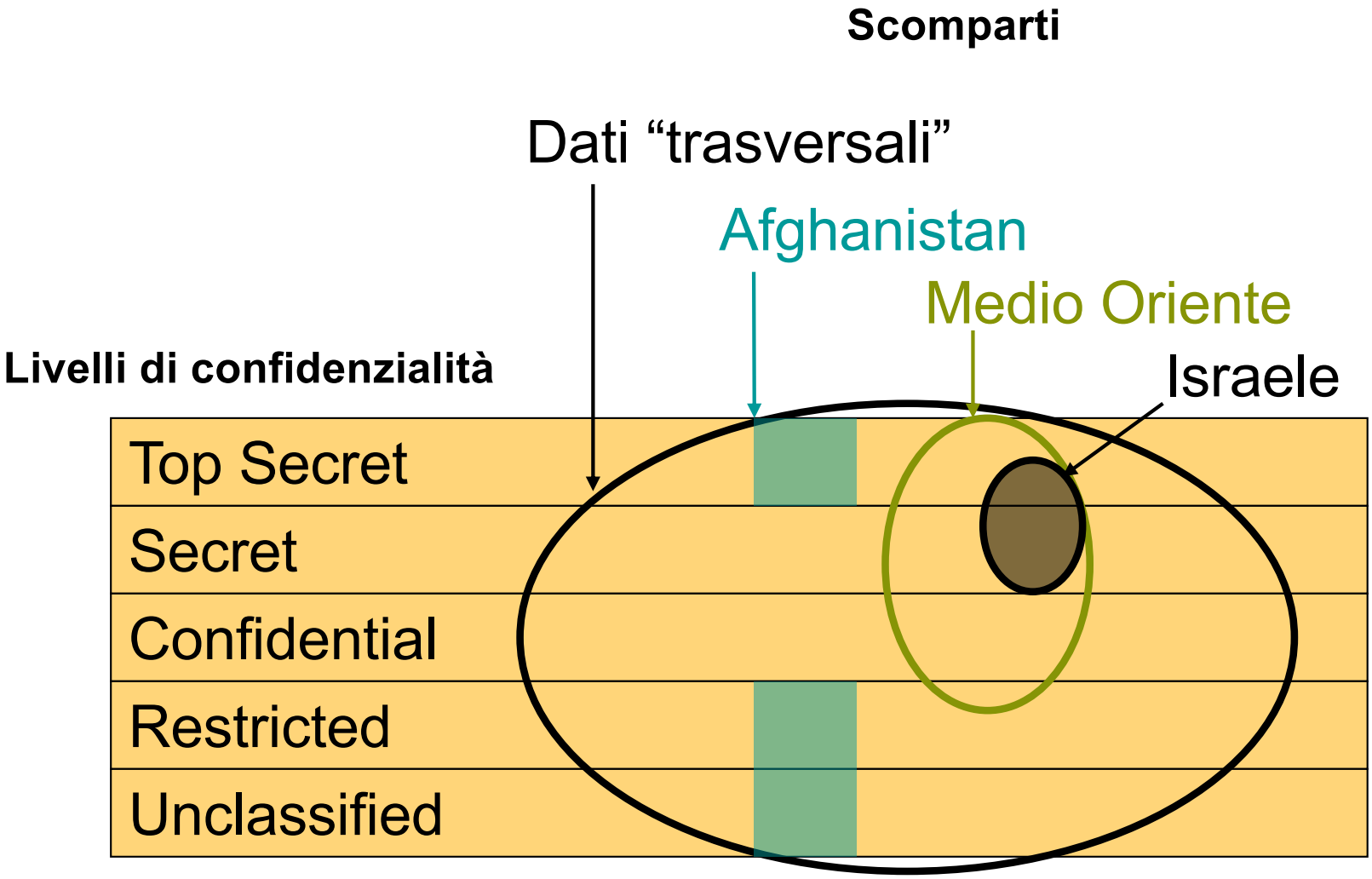
MultiLevel Security (MLS) - 5

Modello Biba - Integrità



MultiLevel Security (MLS) - 6

Si possono complicare le cose ulteriormente...



MultiLevel Security (MLS) - 7

In ambito militare: classificazione di personale e dati

- Classe = ⟨rango, scomparto⟩
- Classe di S sempre indicata con L(S)

Relazione di dominanza

- $L(S_1) \leq L(S_2)$ sse $\text{rango}_1 \leq \text{rango}_2$ and $\text{scomparto}_1 \subseteq \text{scomparto}_2$
- Esempio:

  ⟨Restricted, Israele⟩ ≤ ⟨Secret, Medio Oriente⟩

Si applica a:

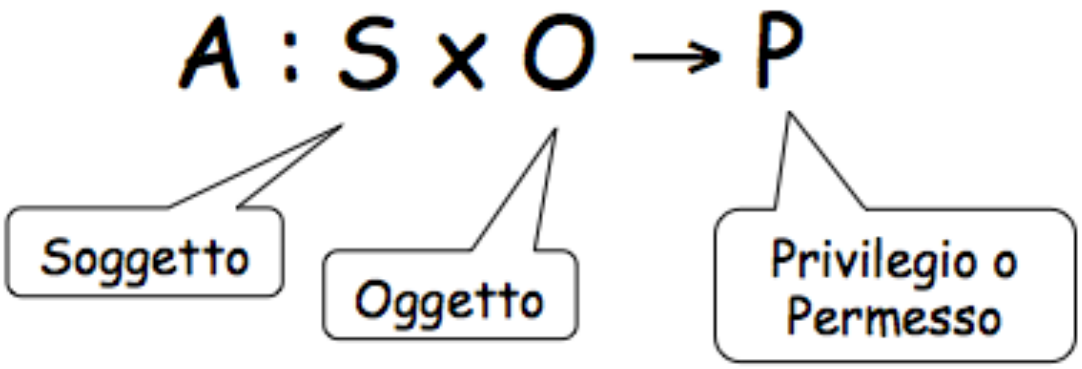
- soggetti – utenti o processi
- oggetti – documenti o risorse

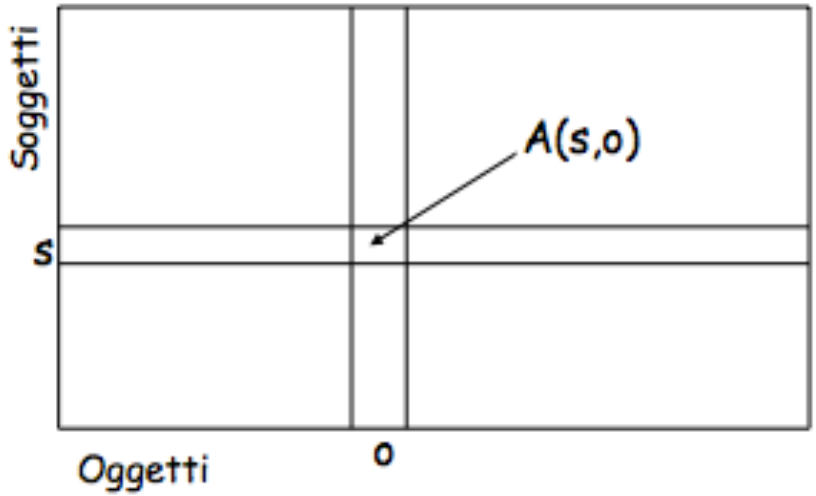
Altro problema:

Come memorizzare le politiche per il controllo degli accessi?

Come memorizzare una politica? (1)

Mediante una matrice degli accessi





La matrice in generale è grande e sparsa:

- Memorizzata per colonne: access control list (ACL)
  - *Lista di soggetti* che possono accedere ad una data risorsa
- Memorizzata per righe: capability list
  - *Lista di risorse* a cui può accedere un utente
- Memorizzate triple non-nulle: tabelle di autorizzazione
  - DBMSs

Matrice per il controllo degli accessi [Lampson]

Esempio 1:

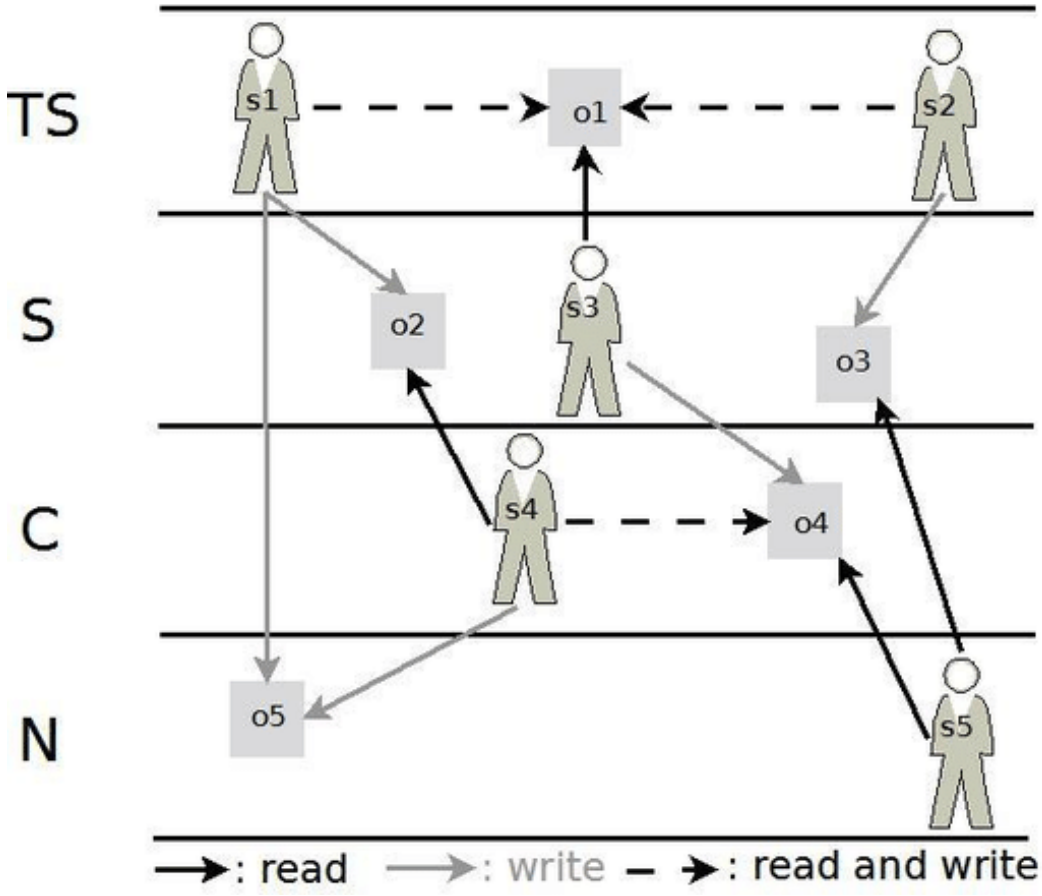
Oggetti

Soggetti

	File 1	File 2	File 3	...	File n
User 1	read	write	-	-	read
User 2	write	write	write	-	-
User 3	-	-	-	read	read
...					
User m	read	write	read	write	read

Matrice per il controllo degli accessi [Lampson]

Esempio 2: Biba model

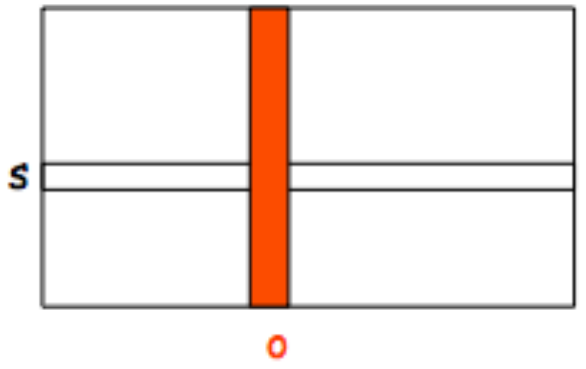


	o1	o2	o3	o4	o5
s1	read write	write			write
s2	read write		write		
s3	read			write	
s4		read		read write	write
s5			read	read	

Come memorizzare una politica? (2)

Access control list

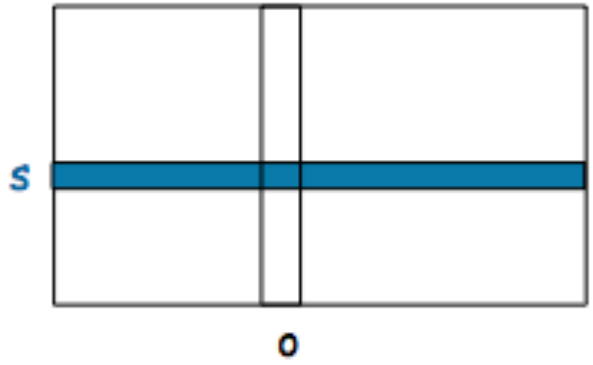
- Implementa la matrice degli accessi per colonne
- Memorizza insieme all'*oggetto o* la colonna contenente la lista dei *soggetti s* che possono accedere all'oggetto e con quale permesso
  - compatta
  - facile sommario per-oggetto
  - difficile revocare un soggetto
  - utile:
    - in caso di molti oggetti
    - in caso di pochi soggetti



Come memorizzare una politica? (3)

Capability list

- Implementa la matrice degli accessi per righe
- È come se il soggetto avesse un "biglietto" per ogni oggetto/risorsa
  - Elenca gli *oggetti o* ai quali il *soggetto s* può accedere e con quale privilegio
- Due varianti (implementative):
  - Memorizzare la riga della matrice insieme all'utente (controllo del SO)
  - Memorizzare dei ticket non modificabil
- Caratteristiche:
  - utile in caso di delegazione
  - facile delegare a qualcuno
  - difficile revocare una capability (accesso ad un dato oggetto)



ACL vs Capabilities (1) – Esempio reale

ACL:

- Il mio nome è nella lista

CL:

- Ecco il biglietto

ACL vs Capabilities (2)

ACL:

- Associa una lista a ciascun oggetto
- Controlla la presenza del soggetto/gruppo nella lista
- Ha bisogno dell'autenticazione: deve conoscere l'identità del soggetto

CL:

- Se la capability è un ticket non modificabile e univoco, i soggetti se lo possono passare
- Si controlla il possesso del biglietto e non l'identità del soggetto (altra forma di autenticazione…)

ACL vs Capabilities (3)

ACL – delega:

- Chiedo che il soggetto venga inserito nella lista?

CL – delega:

- Semplice (anche a run-time): consegno il ticket al delegato

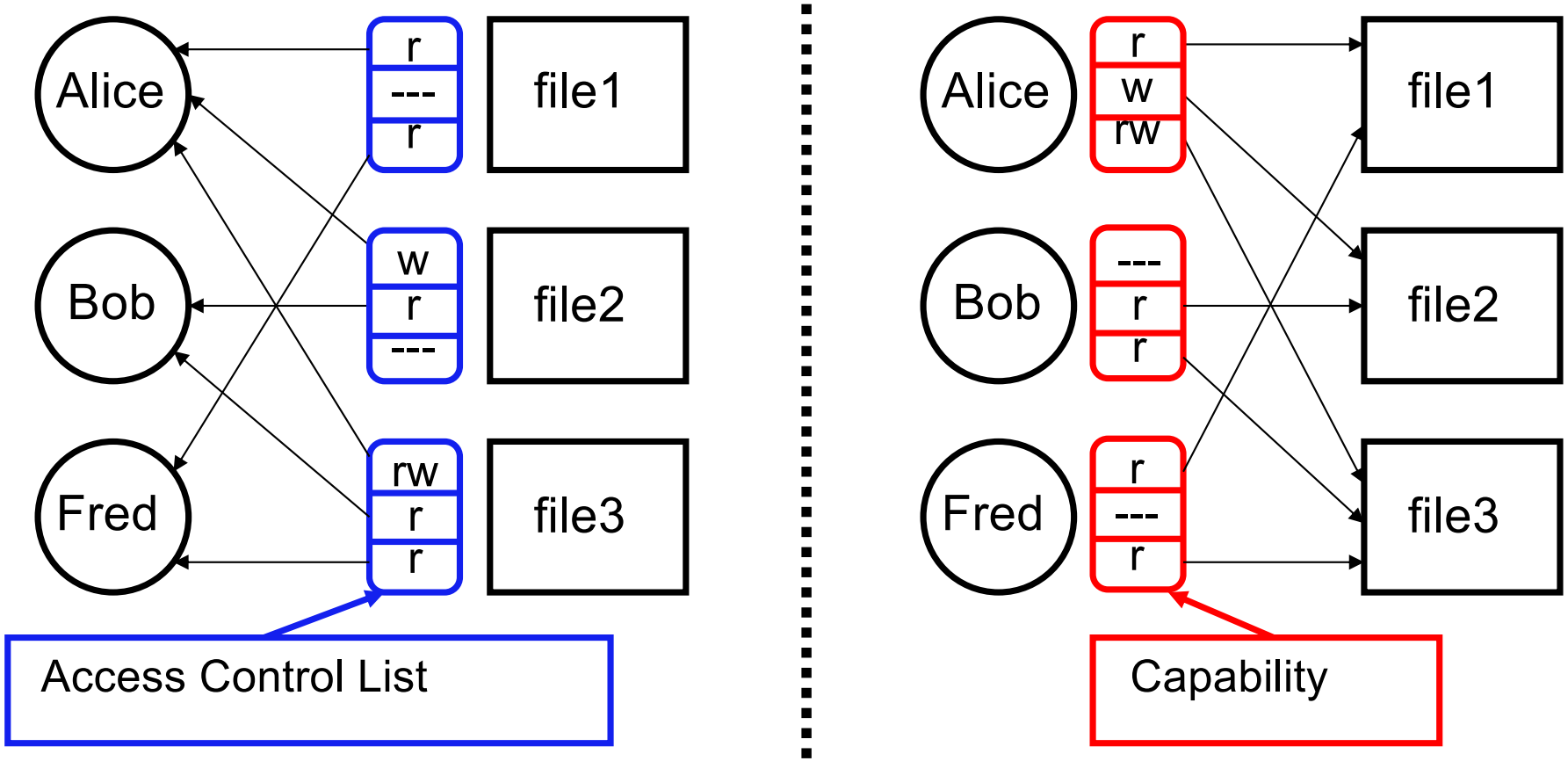
ACL - revoca:

- Semplice: rimuovi soggetto dalla lista

CL - revoca:

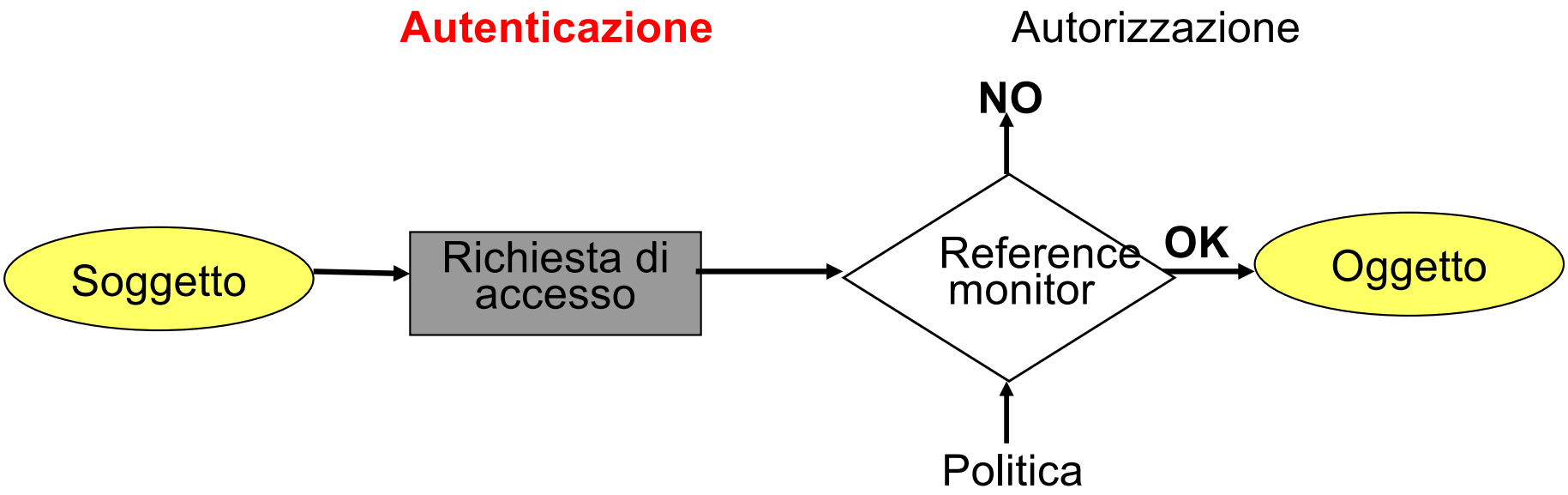
- Bisogna recuperare il ticket…

ACLs vs Capabilities (4)



Autenticazione uomo-macchina

Perché autenticazione? (1)



Perché autenticazione?

Nella vita reale una qualche forma di autenticazione viene eseguita prima di un'azione:

- L'impiegato della banca chiede di vedere la C.I. prima di farvi incassare un assegno
- Il noleggiatore di un'auto chiede di vedere la patente di guida prima di consegnare l'auto

Nei sistemi automatici il principio è lo stesso:

- **Autenticazione prima di autorizzazione**
- **Oppure**: l'identità di un utente viene registrata nei log di sistema quando è importante tenere traccia degli utenti che hanno richiesto un certo servizio

Autenticazione - Definizione

Verificare l'identità di un utente

- Mutua autenticazione: nel caso in cui si debba stabilire l'identità delle due "parti" che interagiscono

- Le parti possono essere utenti o computer
  - **computer-computer** (stampa in rete, …)
  - **utente-utente** (posta elettronica certificata, …)
  - **computer-utente** (autenticare un server web,…)
  - **utente-computer** (per accedere ad un sistema,…)

Autenticazione utente-computer (1)

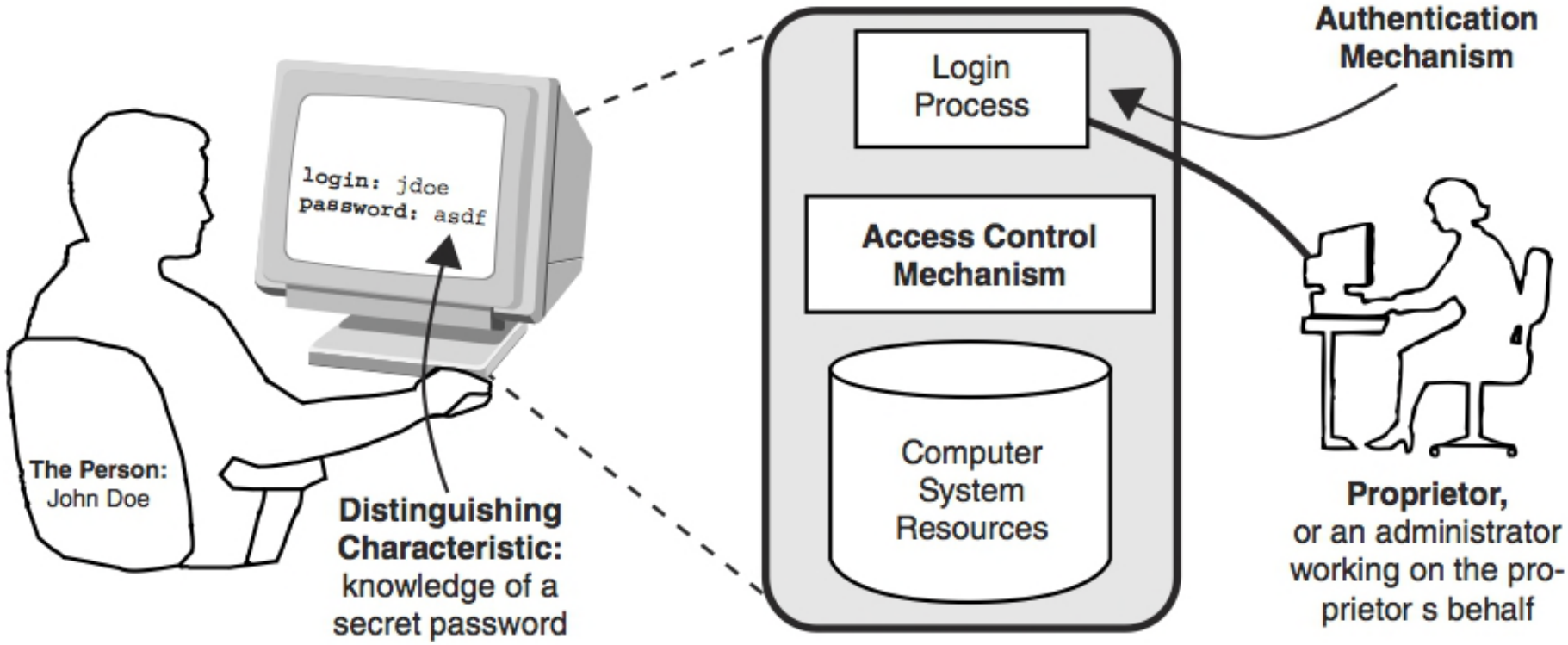


FIGURE 1.1: *Five elements of authentication.* These elements are: the person, the distinguishing characteristic, the proprietor, the authentication mechanism, and the access control mechanism.

Autenticazione utente-computer (2)

Basata su qualcosa che l'utente:

Conosce: un informazione segreta

- Password, PIN, …

Possiede: cosa fisica

- Chiavi convenzionali, carte magnetiche o smart card

È: caratteristiche biometriche

- Impronte digitali, dell'iride, tono di voce,…

Autenticazione nella Rete – Precisazione

Nel mondo reale

Sto parlando con Mario

Sto parlando con qualcuno che possiede
le seguenti credenziali di Mario:

- Password
- Chiave privata
- Impronta digitale
- …

Nella Rete

Autenticazione nella Rete – Precisazione

- Vignetta di Peter Steiner
- The New Yorker, 5 Luglio 1993 issue [Vol.69 no. 20] page 61

Autenticazione basata sulla conoscenza

Autenticazione basata sulla conoscenza

Funzionamento di base:

- L'utente ha una *password* segreta
- Quando fa login l'utente immette
  - Username, per identificarsi
  - Password, per autenticarsi
- Il sistema controlla la *password* per autenticare l'utente

Problemi da gestire (e risolvere):

- Il sistema **dove e come memorizza** la *password*?
- Il sistema **come verifica** la *password*?
- Quanto facile è **indovinare** la *password*?

Password memorizzate in chiaro

- Le password vengono memorizzate in chiaro in un file protetto da un meccanismo di controllo dell'accesso
- La validazione avviene confrontando la password inserita con quella memorizzata

- Metodo usato da Compatible Time Sharing System (CTSS), uno dei primi sistemi operativi *time-sharing (*1961-MIT, in uso fino al 1973)

Problema:

- Nessuna protezione contro chi si impossessa del file!

Password memorizzate cifrate (1)

- L'intero file o solo la parte relativa alle password viene **cifrata**
- Cifrata = crittata oppure *hash*
  - Funzione hash *h*, proprietà:
    - *h* : stringhe → stringhe
    - dato *h(password)*, **è difficile indovinare la password**
- Per la verifica:
  - Il sistema decifra la password memorizzata e la confronta con quella inserita dall'utente, oppure
  - Il sistema calcola l'hash della password inserita dall'utente e la confronta con quella cifrata

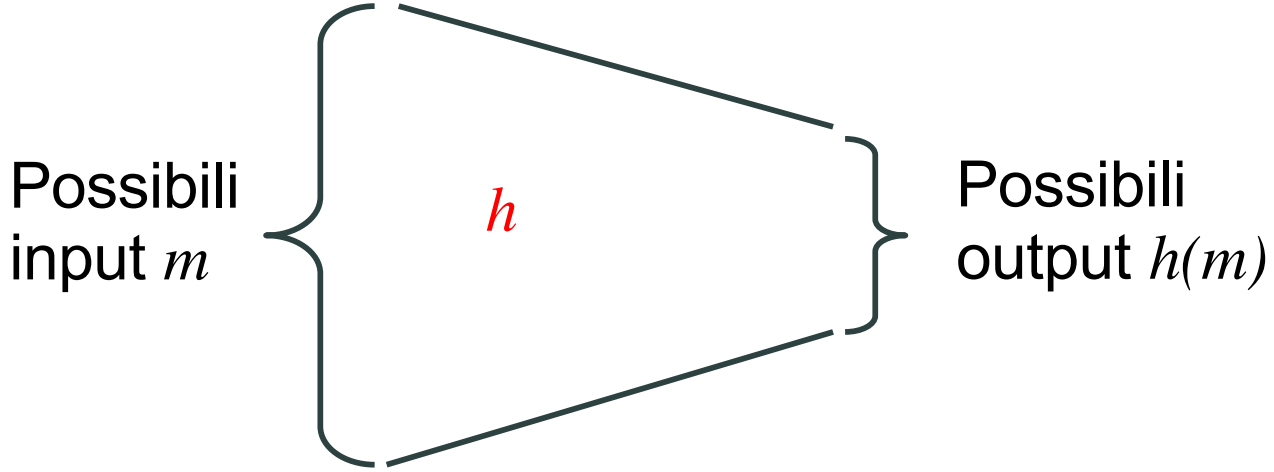
Problema:

- Due utenti con la stessa password hanno entrate identiche nel file delle password

Funzione Hash (1)

Una **funzione hash** è una funzione computazionalmente efficiente che mappa stringhe binarie di lunghezza arbitraria in stringhe di lunghezza costante, dette **hash value** (o **message digest**).

- Detta anche one-way function, checksum o message digest
- h: $\{0,1\}^* \rightarrow \{0,1\}^n$



Funzione Hash (2) - Proprietà

Semplice da calcolare

- Dato x, è facile calcolare $h(x)$

Compressione

- h mappa input di lunghezza arbitraria a output $h(x)$ di lunghezza n definita

One way (= irreversibile)

- Dato y, è difficile trovare x t.c. $y=h(x)$ [preimage resistance]
- Dato m, è difficile trovare m' t.c. $h(m) = h(m')$ [second preimage resistance]

Collision-resistant

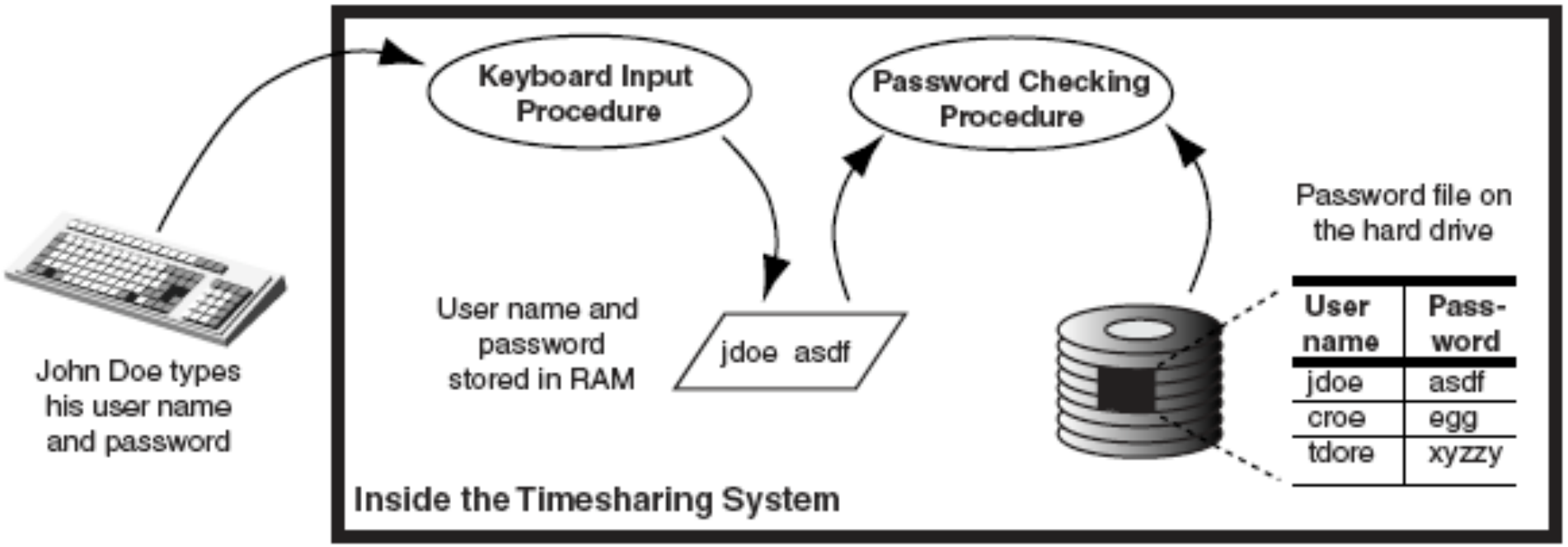
- Difficile trovare m, m' distinti t.c. $h(m)=h(m')$

Effetto valanga

- Una piccola modifica di m deve alterare tutto $h(m)$

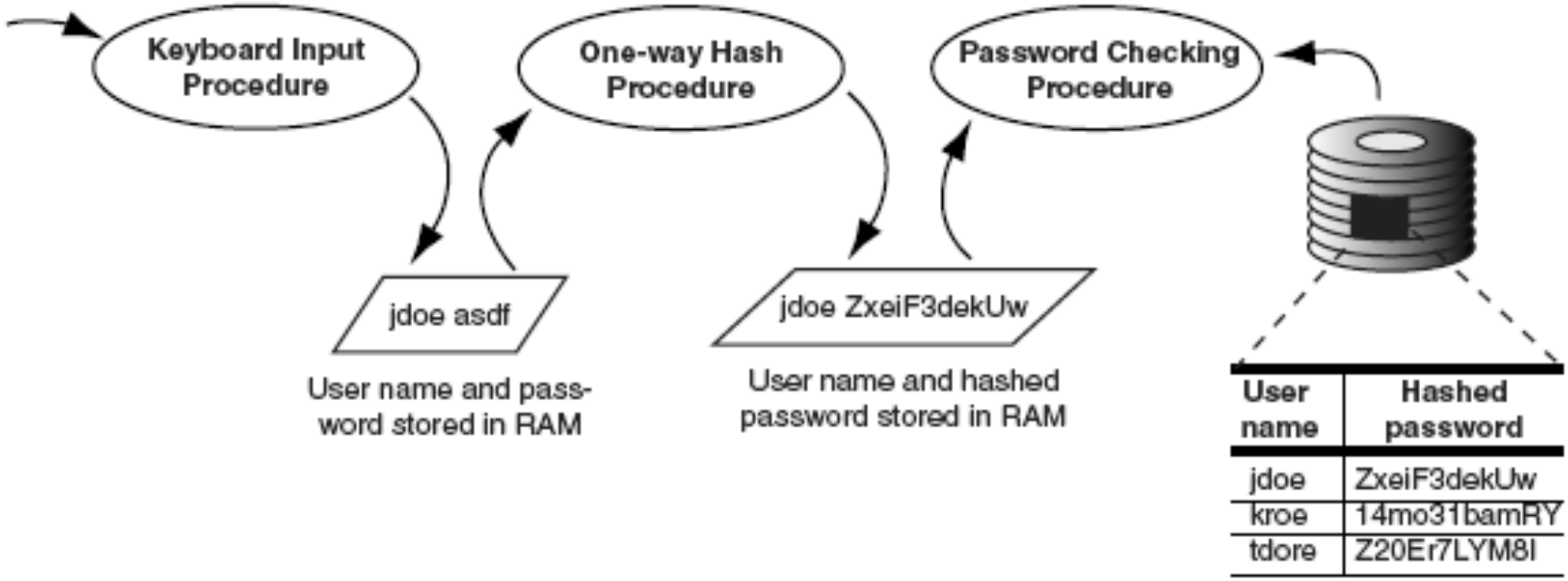
Password memorizzate cifrate (2)

In CTSS, prima:



Password memorizzate cifrate (2)

In CTSS, poi:



Password memorizzate cifrate (3)

Cosa può fare un attaccante se recupera il file delle password (cifrate)?

Possibili attacchi:

- **Attacco esaustivo (o a forza bruta)**
  - L'attaccante prova in modo sistematico tutte le possibili password
  - **Esempio**: Password costruite sull'alfabeto {A,...,Z} (quindi 26 caratteri) e di lunghezza variabile tra 1 e 8
    - $26^1 + 26^2 + \ldots + 26^8 = 5 * 10^{12}$ combinazioni da provare

Password memorizzate cifrate (4)

Possibili attacchi:

- **Attacco dizionario**
  - L'attaccante prova le password probabili
    - Gli utenti preferiscono le password corte a quelle lunghe
    - Gli utenti legano la propria password a qualcosa che ha un particolare significato per loro (Nome, Data di nascita, Nome del cane, Squadra di calcio preferita, …)

Funzionamento:

- L'attaccante calcola in anticipo *h(x)* per tutte le *x* in un dizionario di password comuni
- Se recupera il file delle password deve "solo" confrontare gli hash con quelli che lui ha calcolato in precedenza

- **Riusciamo a prevenire questo attacco oppure a renderlo più difficile?**

Password memorizzate cifrate (5)

In Linux/Unix:

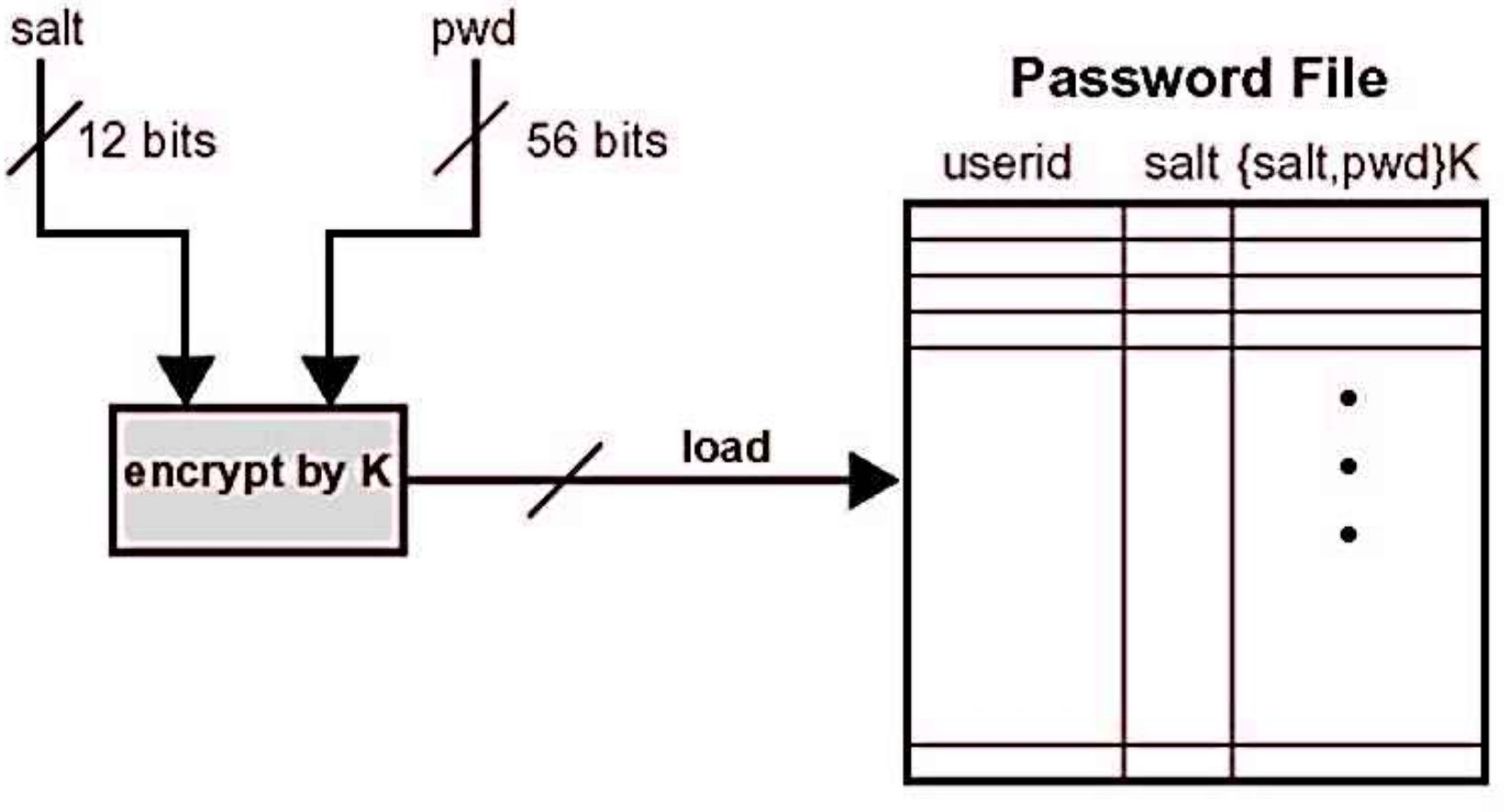
- Prima crittate con DES (25xDES), ora con una funzione hash (prima MD5, ora SHA-512)

- Idea: Viene memorizzato hash(**salt** + password)
  - **Salt**: sequenza di 12 (ora 48-128) bit generata del sistema
  - Salt unico per ogni utente e memorizzato in chiaro

Password memorizzate cifrate (6)

In Linux/Unix:

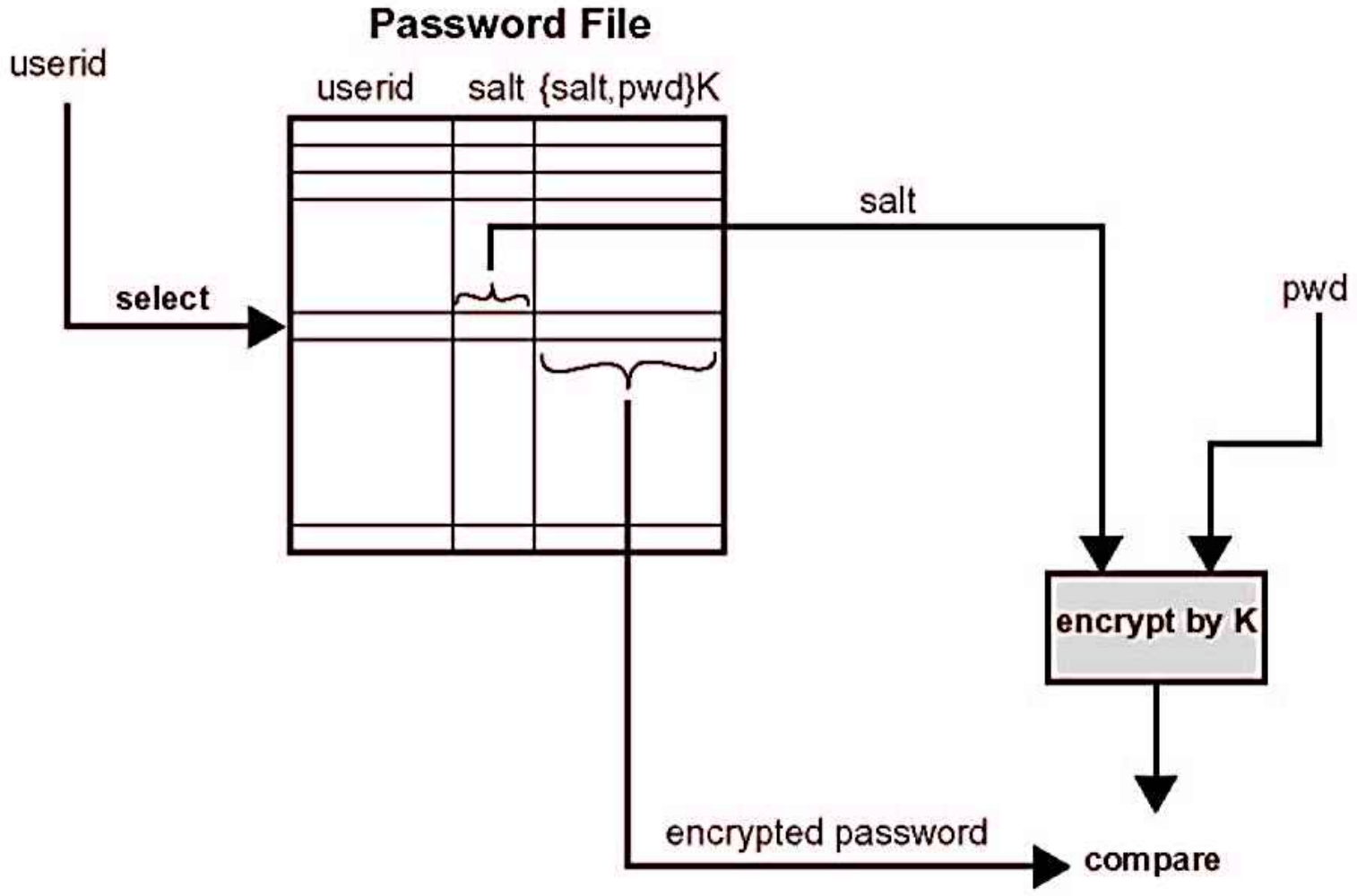
- Aggiunta di una nuova password:



Password memorizzate cifrate (7)

In Linux/Unix:

- Verifica di una password:



Password memorizzate cifrate (8)

Vantaggi:

1. Due utenti con la stessa password hanno hash diverso
   - il salt si basa sulla data e l'ora in cui è stato creato l'account, quindi sarà diverso in quanto non è possibile che la creazione di due account su uno stesso sistema avvenga contemporaneamente

2. L'attaccante deve **ricalcolare** gli hash delle parole presenti nel dizionario **per ciascun utente**, molto più lavoro!!

Altri problemi delle password?

Vulnerabilità delle password…

Le password rischiano di:

- essere indovinate (**guessing**)
- essere sbirciate mentre vengono inserite (shoulder surfing o **snooping** = ficcare il naso)
- venire intercettate durante trasmissione in rete (**sniffing** = annusare e keystroke sniffing)
- venire acquisite da terze parti che impersonano l'interfaccia di login (Trojan login - **spoofing** = truffa)
- venire "rubate" da un social engineer

Chiunque conosca la password di un utente può impersonare quell'utente col sistema!

… e difese

Dal guessing attack:

- Limite agli sbagli permessi
- Verifica dei log
- Vincolare il formato della password (per rendere la password “forte”)

Dallo snooping e sniffing attack:

- Schermare la password scritta (*******)
- Protezione della memoria per *keystroke sniffing*

Dall’offline dictionary attack

- Shadow password (Unix) => file delle password accessibile solo da utenti con *root privilege*

Password: buona gestione (1)

Cause principali di vulnerabilità:

- password **immutata** per lungo tempo
- **condivisione** di password:
  - con colleghi o amici
  - su computer/applicazioni/servizi diversi
- **scelta** di password deboli
- scrittura delle password su pezzi di carta

Password: buona gestione (2)

Linee guida:

- Cambiare password frequentemente
- Non condividere la password con altri
- Non usare la stessa password per autenticazioni diverse
- Usare almeno 8 caratteri
- Non usare una parola del dizionario
- Bilanciare
  - Semplicità (facile da ricordare, non serve trascriverla)
  - Complessità (difficile da intuire, robusta verso guessing)

Password: buona gestione (3)

- Nella realtà:

RANK	PASSWORD	CHANGE FROM 2015
1	123456	Unchanged
2	password	Unchanged
3	12345	2 ↗
4	12345678	1 ↘
5	football	2 ↗
6	qwerty	2 ↘
7	1234567890	5 ↗
8	1234567	1 ↗
9	princess	12 ↗
10	1234	2 ↘
11	login	9 ↗
12	welcome	1 ↘
13	solo	10 ↗
14	abc123	1 ↘
15	admin	NEW
16	121212	NEW
17	flower	NEW
18	password	6 ↗
19	dragon	3 ↘
20	sunshine	NEW
21	master	4 ↘
22	hottie	NEW
23	loveme	NEW
24	zaq1zaq1	NEW
25	password1	NEW

Report annuale di SplashData: lista delle peggiori password (2016)

Password: buona gestione (4)

- Nella realtà:

RANK	PASSWORD	CHANGE FROM 2014
1	123456	Unchanged
2	password	Unchanged
3	12345678	1 ↗
4	qwerty	1 ↗
5	12345	2 ↘
6	123456789	Unchanged
7	football	3 ↗
8	1234	1 ↘
9	1234567	2 ↗
10	baseball	2 ↘
11	welcome	NEW
12	1234567890	NEW
13	abc123	1 ↗
14	111111	1 ↗
15	1qaz2wsx	NEW
16	dragon	7 ↘
17	master	2 ↗
18	monkey	6 ↘
19	letmein	6 ↘
20	login	NEW
21	princess	NEW
22	qwertyuiop	NEW
23	solo	NEW
24	passw0rd	NEW
25	starwars	NEW

Report annuale di SplashData: lista delle peggiori password (2015)

Password: buona gestione (5)

- Nella realtà:
  - (2016) Sembra che la password dell'account di posta privata su Gmail di John Podesta fosse "password"
  - (2016) Zuckeberg usava la stessa password nei siti di LinkedIn, Twitter e Pinterest ed era "dadada"

Password: buona gestione (6)

Sw per la gestione delle password:

1. crea per noi delle password robuste
2. funge da database di tutte le nostre password
   - Le memorizza cifrate!!!
3. compila automaticamente i campi dedicati

Ne esistono anche di gratuiti!

- LastPass, KeePass, Norton Identity Safe, ecc.

Password: buona amministrazione

Controlli automatici su password per evitare grosse debolezze:

- Restrizioni sulla lunghezza e sul minimo numero di caratteri
  - Richiesta combinazione di caratteri alfanumerici
- Controllo rispetto a dizionari
  - Rifiuto delle parole del linguaggio naturale
- Verifica del massimo tempo di validità
  - L'utente deve cambiare la password quando scade
- Limitare il numero di tentativi di inserimento
  - Quanti tentativi? E dopo quanto sbloccare l'accesso?

Password: distribuzione iniziale

L'utente si reca dall'amministratore:

- scomodo per l'utente
- pericoloso per il sistema

No password o password di default

Password spedita via posta, email, inserita dall'utente al momento della consegna del login o detta al telefono

- Il messaggio può venire intercettato!!

L'amministratore prepara l'account con una password iniziale (*pre-expired password*):

- obbligo di modifica entro un tempo massimo
- controllo della password scelta (lunghezza, caratteri, …)

Autenticazione utente-computer

Basata su qualcosa che l'utente:

Conosce: un informazione segreta

- Password, PIN, …

Possiede: cosa fisica

- Chiavi convenzionali, carte magnetiche o smart card

È: caratteristiche biometriche

- Impronte digitali, dell'iride, tono di voce,…

Autenticazione basata sul possesso

Autenticazione basata sul possesso

Il possesso di un token fornisce la prova dell'identità

Possesso di cosa?

- Carte magnetiche
- *Smart card* per memorizzare una password robusta
  - *Memory card:* ha una memoria ma **non** capacità computazionali
  - *Microprocessor card:* ha memoria **e microprocessore**
- *Smart token*
  - *Generatori di one-time password*
- *RFID (Radio Frequency IDentification)*

Vantaggi e svantaggi

Svantaggi

- L'autenticazione dimostra solo l'identità del token, non quella dell'utente
  - Se in possesso di un token si impersona l'utente
  - Token persi, rubati, falsificati
- Costi della progettazione del token
- Token perso/rubato: come gestire il periodo transitorio?

Vantaggio: difficile estrarre un segreto da un token

Idea: combinare possesso e conoscenza

- Bancomat: carta + PIN
- Homebanking: password + codice generato da OTP

Carta magnetica (1)

- Inventata nel 1960 dall'IBM
- L'identificativo dell'utente è memorizzato sulla carta
- Molto diffusa
- Spesso utilizzata insieme ad un PIN

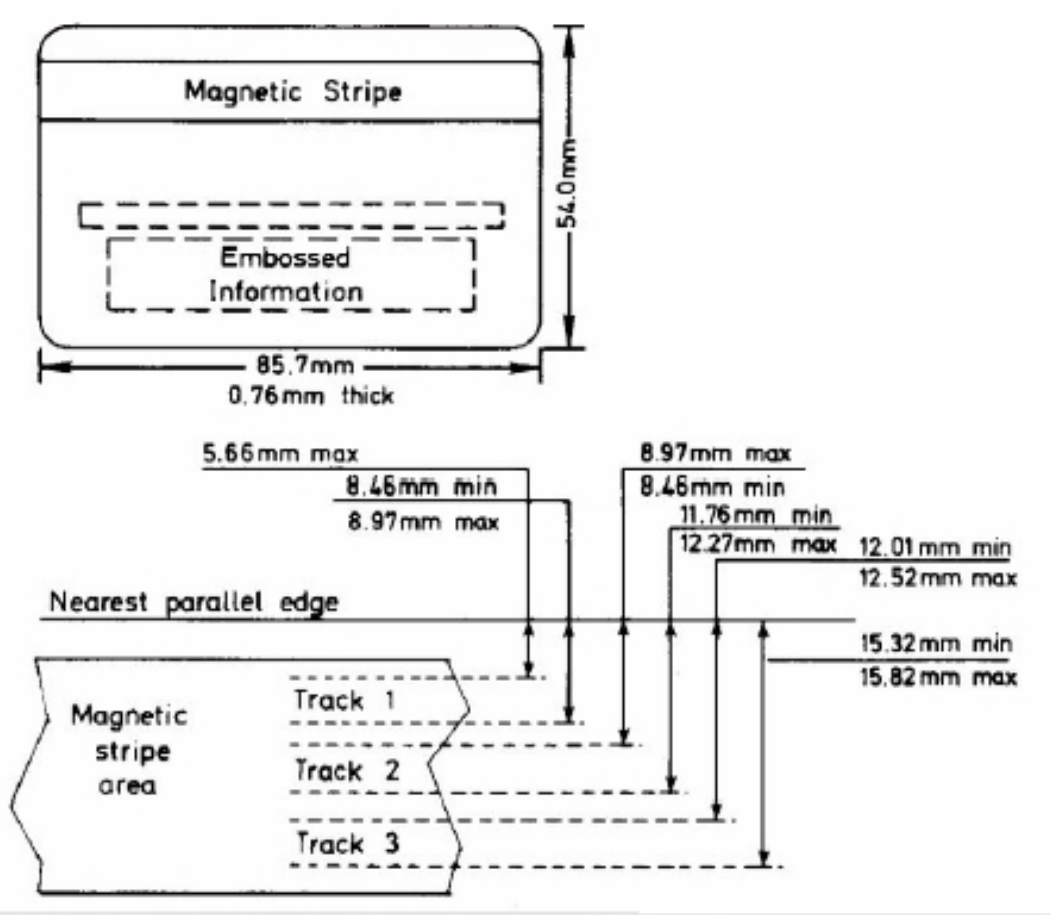
Per la verifica:

- Sistemi *on-line*: il PIN viene verificato tramite un server centralizzato esterno
- Sistemi *off-line*: il PIN viene verificato sulla carta

Svantaggi:

- Poca memoria: in genere 250 byte
- Facile da copiare/forgiare (forging)

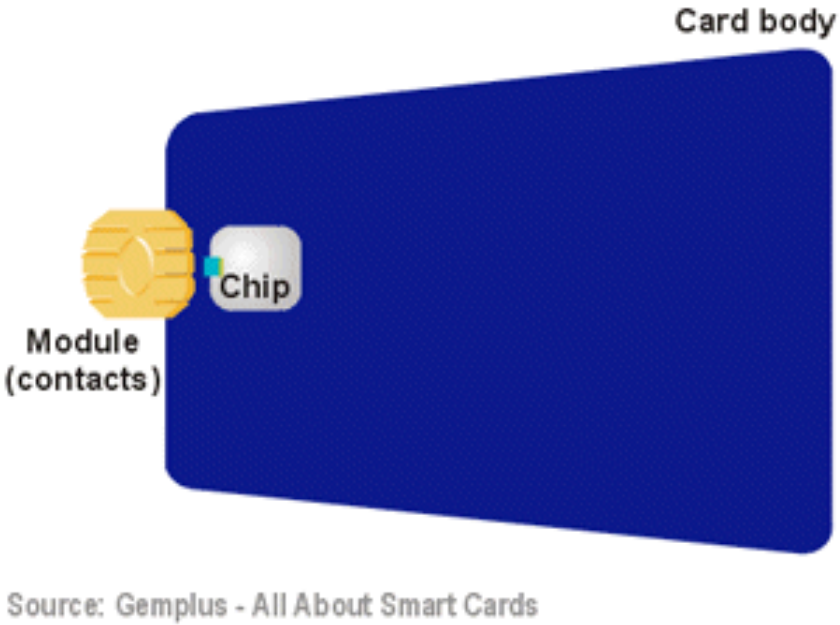
Carta magnetica (2)



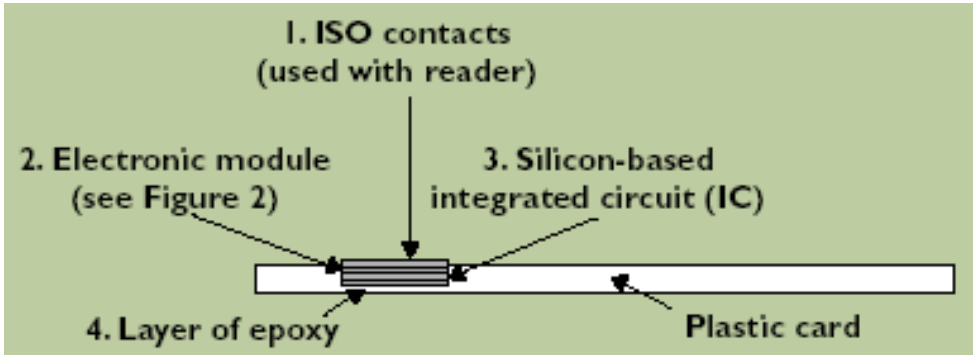
- Lo standard ISO 7810 specifica le **dimensioni** della carta e il **formato** della striscia magnetica, dei dati, flessibilità della carta, ecc.
- In genere l'informazione minima necessaria è contenuta sia nel Track 1 che nel Track 2
- Solo il Track 1 può contenere caratteri (nome del proprietario)
- Il Track 3 non è quasi mai usato

Smart card

Spessore: 0.76 mm



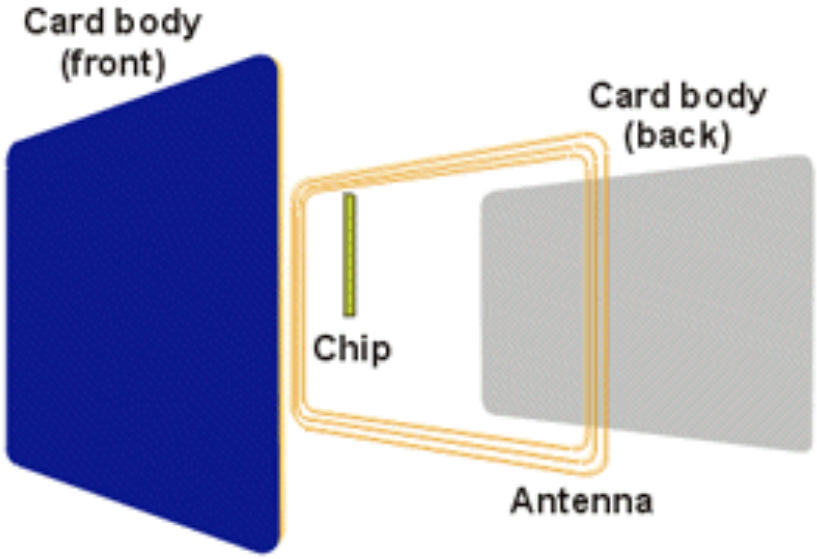
Source: Gemplus - All About Smart Cards



- Inventata nel 1968 in Germania, brevettata negli anni ‘70, messa in commercio solo negli anni ‘80
- Definita nello standard ISO 7816

Perché *smart*?

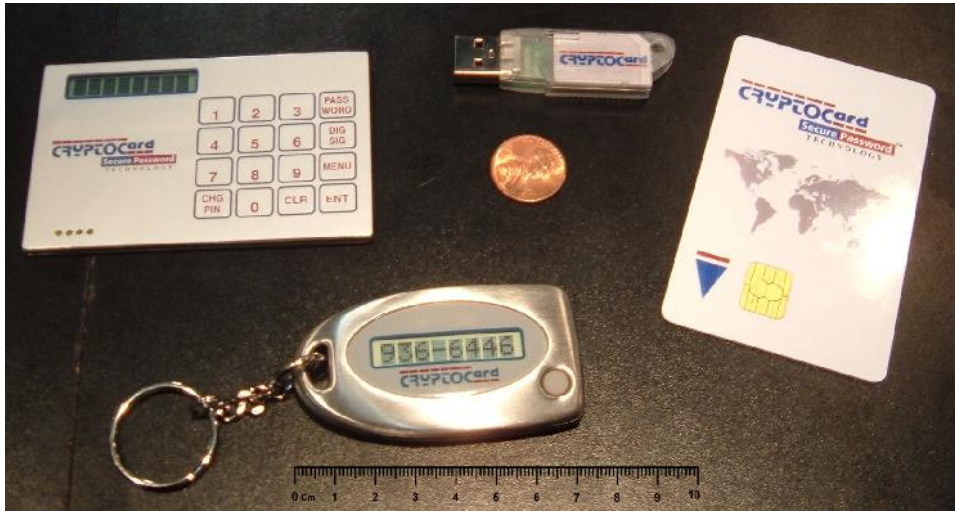
- Contiene un *microprocessore*
  - 8, 16, 32 bit
- Ha più memoria (RAM, ROM, EEPROM, Flash)
- Ha un *co-processore crittografico*
  - Può venire utilizzata per un challenge-response
- I/O channel (Contact/Contactless)



Source: Gemplus - All About Smart Cards

Smart card: applicazioni

- Fin dagli anni ‘90 come carta di credito
- Ora anche come bancomat
- Usate dai telefoni cellulari:
  - la SIM (Subscriber Identity Module) memorizza l’identità dell’utente e il suo PIN
- Altri usi:
  - Denaro elettronico
    - Mondex (UK) e Proton (Benelux)
  - Titolo di viaggio elettronico
  - Pay TV
  - Badge



Smart token o eToken

- In genere utilizzato come:
  - *One-Time Password (OTP) generator*
  - Contenitore di *credenziali digitali*
- A volte protetto da PIN
- Vero e proprio computer!

Svantaggi:

- Costo della progettazione del token
- Gestione dei token
- Token perso: come gestire il periodo transitorio?

Smart token per generare OTP

Funzionamento:

- Chiave segreta (*seme*) memorizzata all'interno del token dal produttore, condivisa con il server
- Per generare one-time password prende il *seme* ed informazioni esterne (*PIN, ora,…*)
- La password viene visualizzata sul display, rinnovata ogni 30-90 secondi
- La sincronizzazione con il server avviene grazie a *seme* ed algoritmo comune e orologio sincronizzato

RSA SecurID (1)

RSA SecurID (2)

Usa una autenticazione a due livelli:

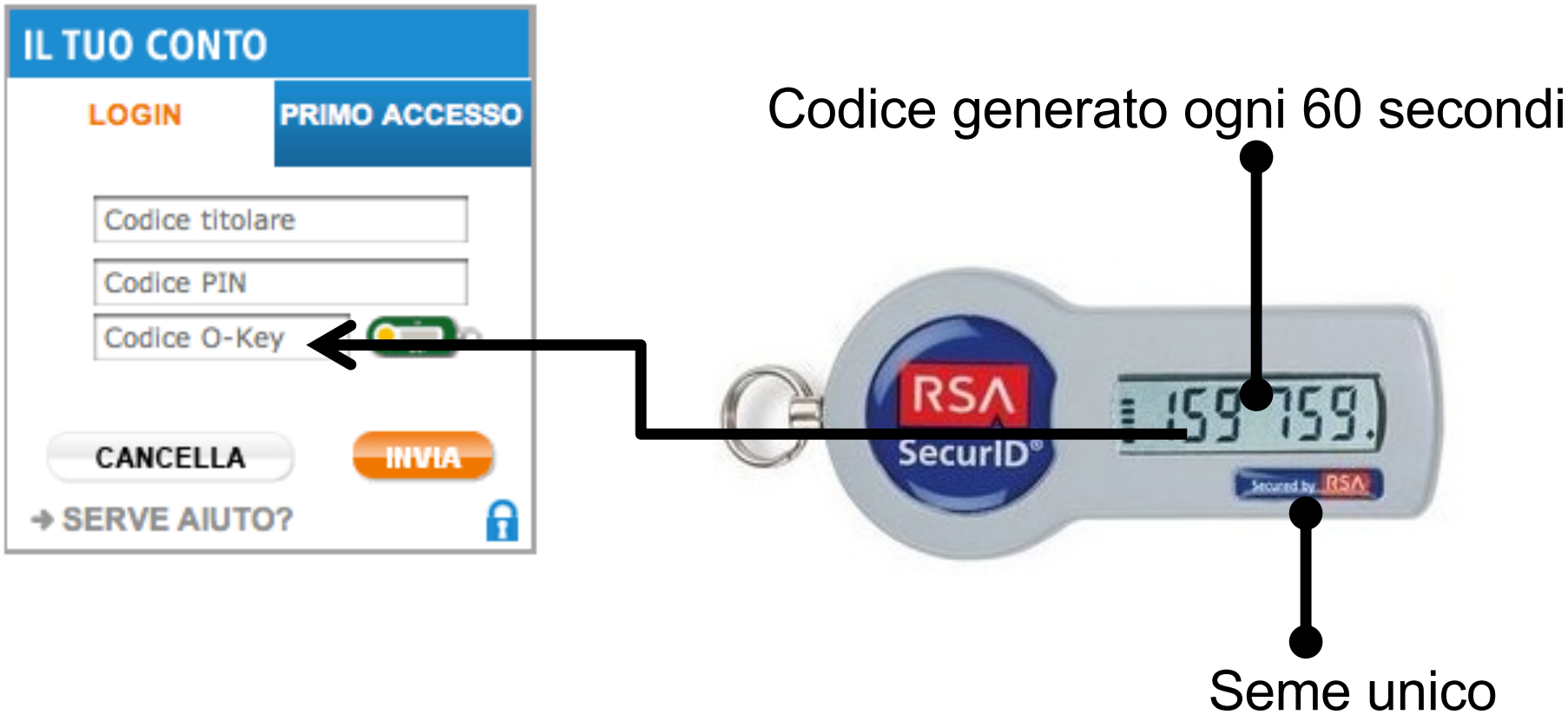
- PIN segreto
- Codice generato dal token

Dentro al token c'è un <u>orologio sincronizzato con il server</u>

- A intervalli regolari (o se si accende il display) il tempo e il *seme* vengono utilizzati e combinati per generare la password
- Gli intervalli sono di 60 secondi

RSA SecurID (3)

Processo di autenticazione (1): a lato client



RSA SecurID (4)

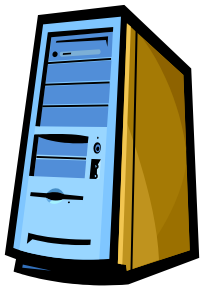
Processo di autenticazione (2): come viene generato il codice del token?

Autenticatore

159759

Algoritmo

Seme | Tempo



RSA Server

159759

Algoritmo

Seme | Tempo

RSA SecurID – Attacco (5)

Cyber attack all'RSA!

- Il 17 Marzo 2011 RSA ha comunicato di essere rimasta vittima di un "sofisticato attacco finalizzato al furto di dati riservati"
- Probabile funzionamento dell'attacco (da un post in un blog «Anatomy of an Attack»:
  - Inviate nel giro di 2 giorni ad un ristretto numero di dipendenti di RSA alcune email con un allegato "maligno" in formato Microsoft Excel con oggetto "*2011 Recruitment plan.xls*"
    - Il file conteneva un malware che sfruttava una vulnerabilità presente in Adobe Flash Player (pubblicata 3 giorni prima dell'attacco!)
    - Permetteva di eseguire un'applicazione (*Poison Ivy)* di grado di comandare da remoto il funzionamento del sistema violato
  - Sottratte le credenziali per l'accesso dell'utente vittima, entrato in altri sistemi dove ha sottratto le credenziali di accesso di altri impiegati che avevano accesso ad informazioni sensibili

eToken PRO

Dispositivo USB contenente delle credenziali digitali

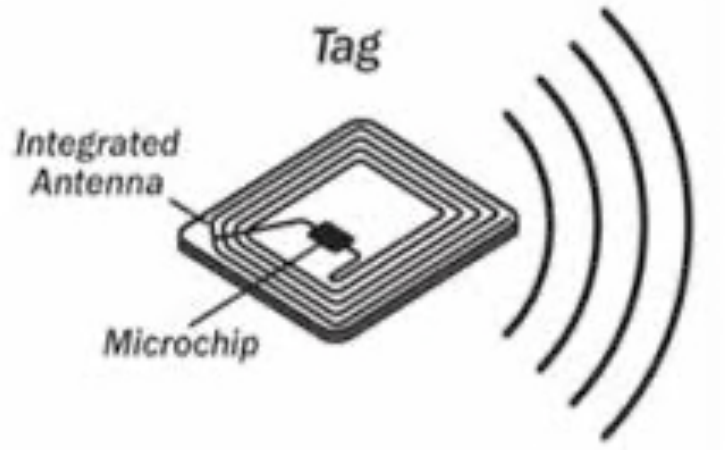
Funzionamento:

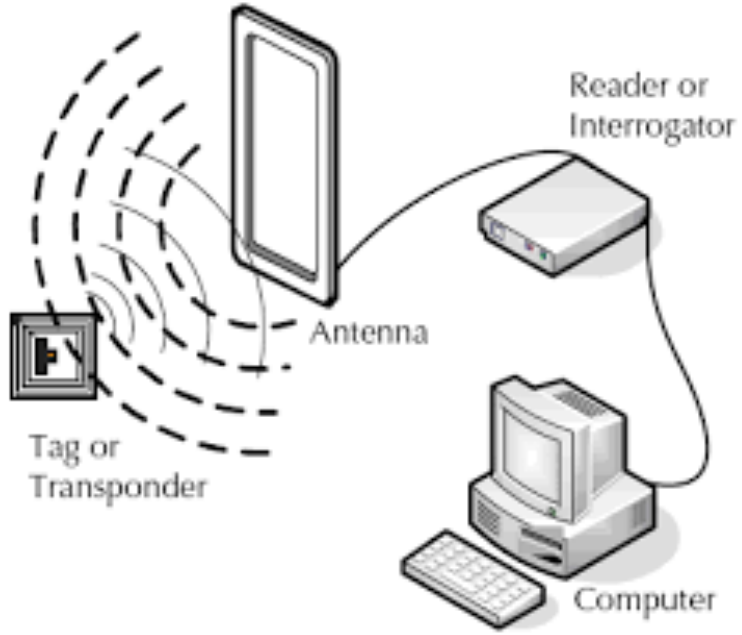
1. L'utente inserisce l'*eToken PRO* nella porta USB
2. L'utente digita la propria password
3. Le chiavi contenute in *eToken PRO* sono a disposizione!

RFID – Radio Frequency Identifier (1)

Costituito da tre elementi fondamentali:

- Una o più etichette RFID (o *tag/Transponder*)
- Un apparecchio di lettura e/o scrittura
- Sistema informativo di gestione dei dati per il trasferimento dei dati da e verso i lettori.





RFID (2)

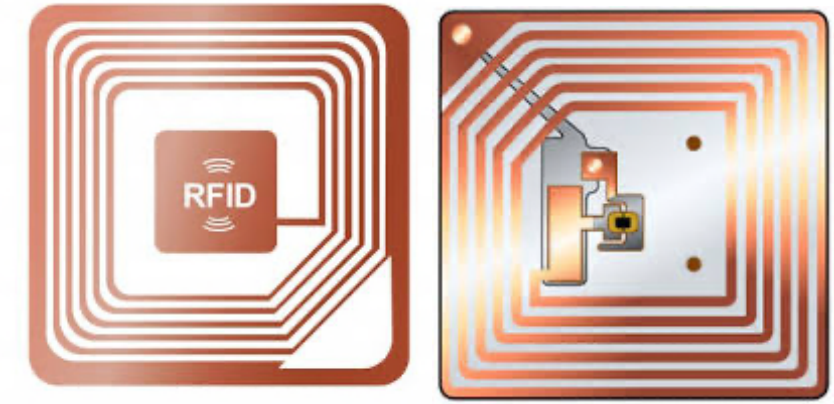
L'etichetta RFID può essere attiva, passiva, semi-passiva o semi-attiva.

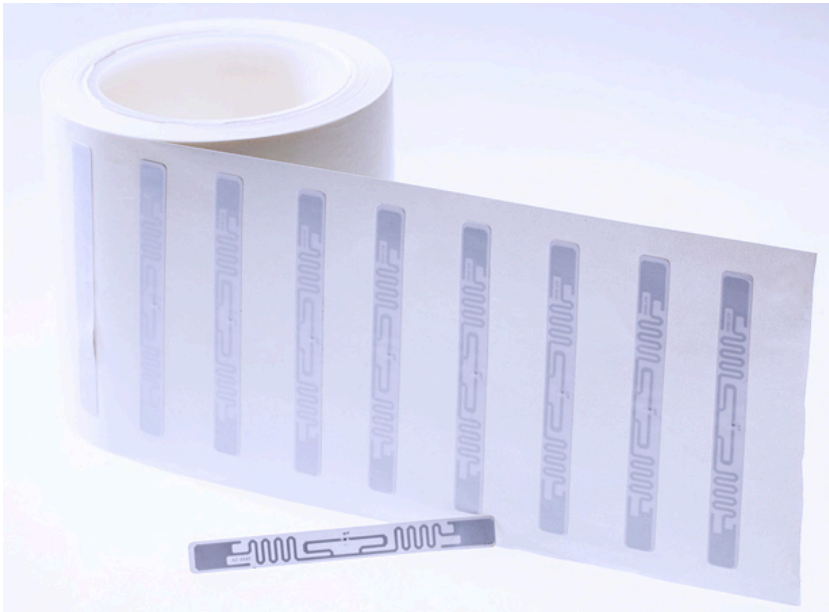
Se è attiva, dispone di:

- una batteria per alimentarla
- una o più antenne per inviare e ricevere segnali
- uno o più tag

Un’etichetta attiva ha una distanza operativa maggiore di una passiva (max 200m).

RFID (3)





Autenticazione basata sul caratteristiche biometriche

Autenticazione basata su caratteristiche

Possesso di caratteristiche univoche fornisce prova dell'identità

- **Fisiche**: impronta digitale, forma della mano, impronta della retina, dell'iride o del viso, …
- **Comportamentali**: firma, timbro di voce, scrittura, "keystroke dynamic",…

Caratteristiche biometriche

Idealmente, la caratteristica biometrica da usare dovrebbe avere le seguenti proprietà:

- **Universalità**: tutte le persone dovrebbero possederla
- **Unicità**
- **Stabilità**: non deve cambiare nel tempo (e non può venire modificata!)
- **Facilità di rilevamento**
- **Accettabilità**
- **Difficoltà di contraffazione**

Funzionamento (1)

1. **Fase iniziale di campionamento**
   - Esecuzione di più misurazioni sulla caratteristica d'interesse
   - Definizione di un **template**
     - **NB:** la perfetta uguaglianza è tecnicamente impossibile!

2. **Autenticazione: confronto fra la caratteristica appena misurata rispetto al template**
   - Acquisizione + confronto + decisione
   - Successo se i due corrispondono **a meno di una tolleranza**, che va definita attentamente

Biometria: utilizzi

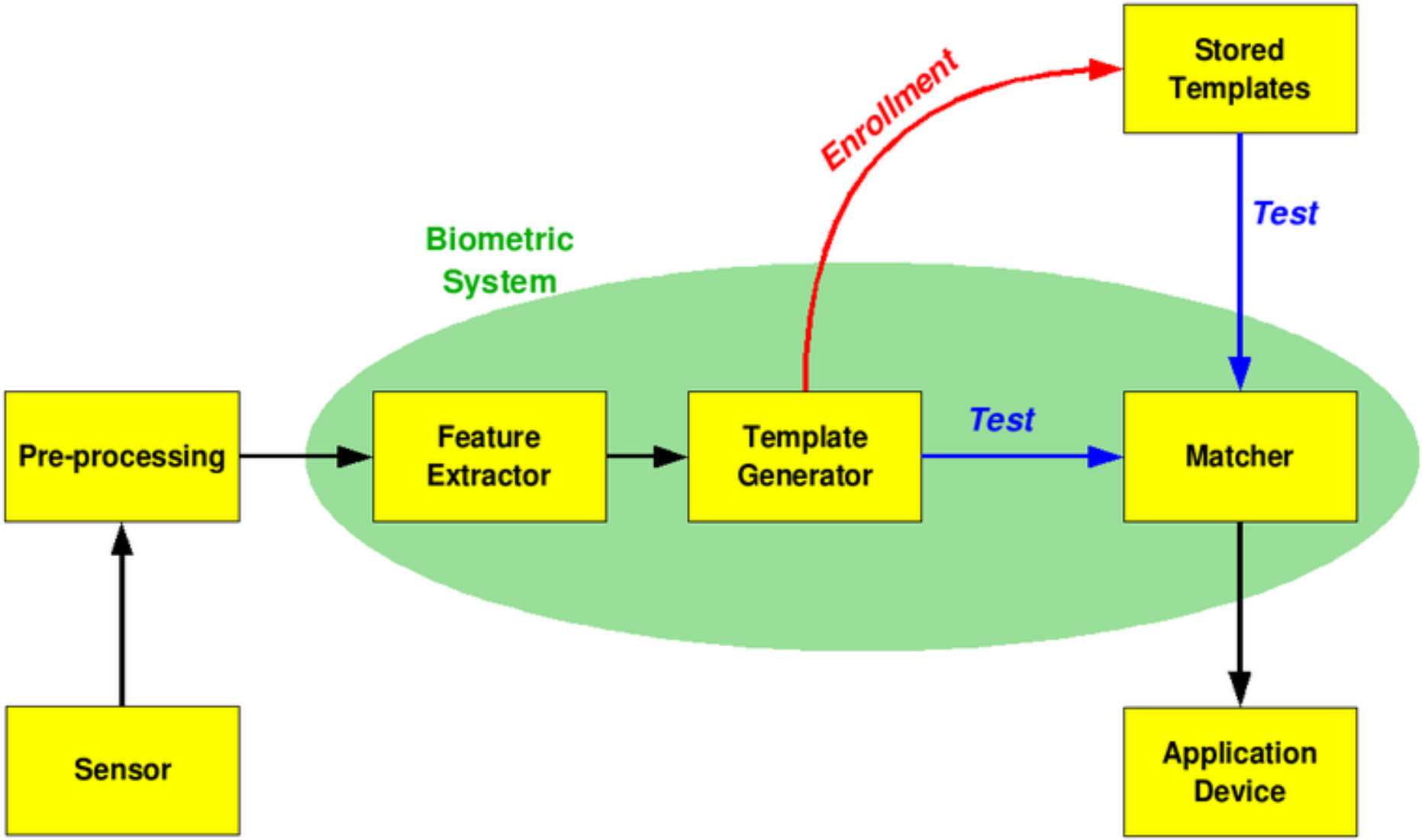
Identificazione:

- Confronto *1:n*: cerca di identificare la persona in un database di *n* persone

Verifica

- Confronto *1:1*: controlla se il template della persona corrisponde con il template di riferimento salvato nel database.

Funzionamento (2)



Problema

La perfetta uguaglianza tra due campioni/template di uno stesso utente è tecnicamente impossibile

Punto cruciale:

- Confrontare la caratteristica appena misurata dall'utente con il *template* di quell'utente
- … eventualmemente distinguendola dal *template* di un altro utente!

Alcune metriche della biometria (1)

FAR – False Acceptance Rate

- Probabilità che il sistema accetti un impostore (che il sistema associ il campione in input con un template presente nel DB, ma relativo ad un altro utente)

FRR – False Reject Rate

- Probabilità che il sistema non riconosca un utente valido (che il sistema non associ il campione in input con il template nel DB relativo ad uno stesso utente)

Soglia di tolleranza (1)

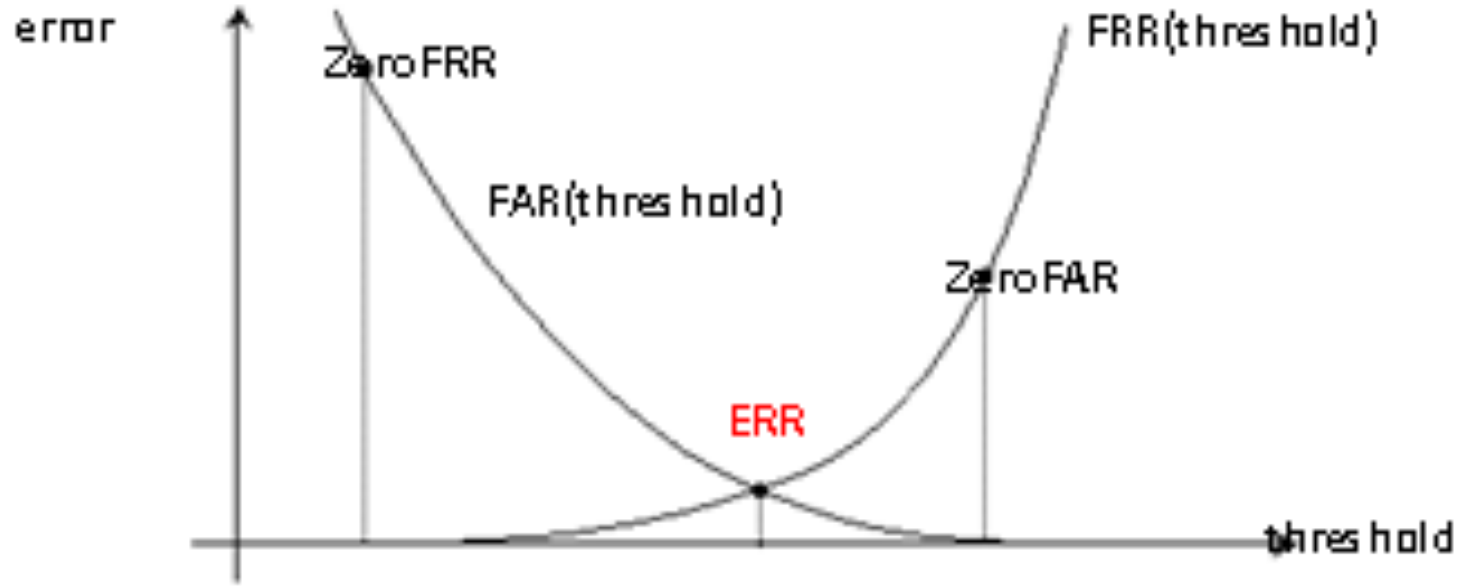
La *soglia di tolleranza dell'errore* è cruciale e dipende dall'applicazione

Errore **= differenza tra valore misurato e valore template**

- Se la soglia è troppo alta, allora si **accettano impostori** (FAR alto)
- Se la soglia è troppo bassa, allora **non si accettano utenti legittimi** (FRR alto)

- Si devono **bilanciare** i 2 valori

Soglia di tolleranza (2)



Alcune metriche della biometria (2)

FTE – Failure To Enroll Rate

- Probabilità che il sistema non riesca a creare un template dal campione in input
- Causato da input di bassa qualità

Template Capacity

- Numero massimo di dati che possono venire memorizzati dal sistema

Possibili attacchi

- Utilizzare una caratteristica biometrica contraffatta (guanto, lente a contatto, maschera)
- Riutilizzare un vecchio template
- Intercettare la comunicazione tra il sensore e il database dove è memorizzato il template, sovrascrivendo la decisione finale
- Modificare il template nel database

Vantaggi/Svantaggi (1)

Forma di autenticazione più forte anche se tecnicamente meno accurata

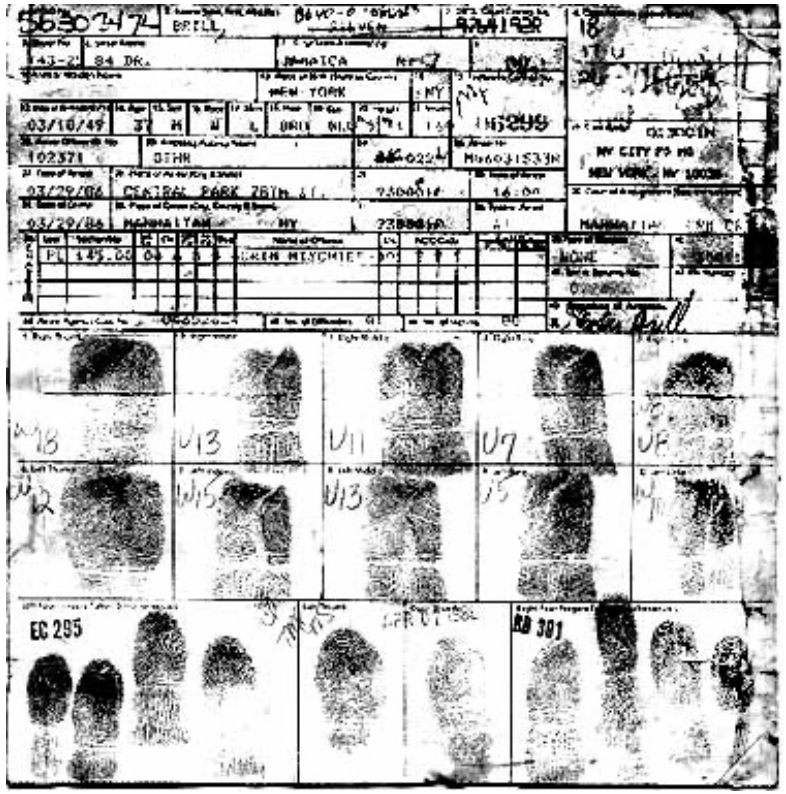
- Eliminate in pratica le impersonificazioni
  - Qualcosa che si conosce può venire indovinato o rubato
  - Qualcosa che si possiede può venire rubato
- Problema dell'accuratezza del template

Impronte digitali

- Piccole righe che si formano su mani e piedi ancor prima della nascita
- Restano inalterate per tutta la vita dell'individuo (a meno di incidenti)
- **Il pattern presente sulle dita è unico per ogni individuo**
- Il riconoscimento di impronte digitali è uno dei metodi più comuni ed affidabili per il riconoscimento di identità

Dall’inchiostro…

Una volta si premeva il dito sull’inchiostro e poi sulla carta con movimento rotatorio



… ai lettori ottici

Ora basta poggiare un attimo il dito sul lettore

Classificazioni di impronte (1)

Classificate in tre grandi gruppi in base allo "schema" predominante

Arch

Whorl

Loop

Accidental

Classificazioni di impronte (2)

- Loop
- 65% of all fingerprints


89s. Loop.

89t. Loop.

89u. Loop.

- Arch
- Plain and tented arch


106


89v. Tented arch.

- Whorl
- 30% of all fingerprints
- One complete circle


92b


92c

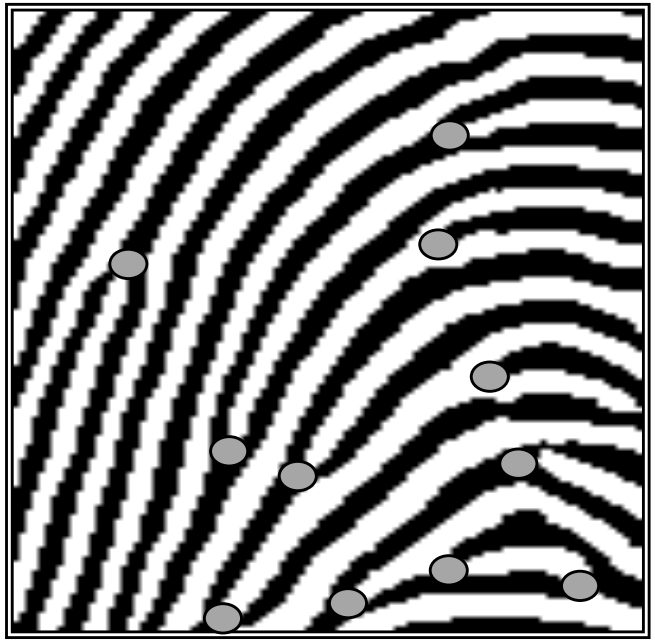

92d

Riconoscimento di impronte (1)

Il campione digitale memorizza le minuzie presenti nell'impronta

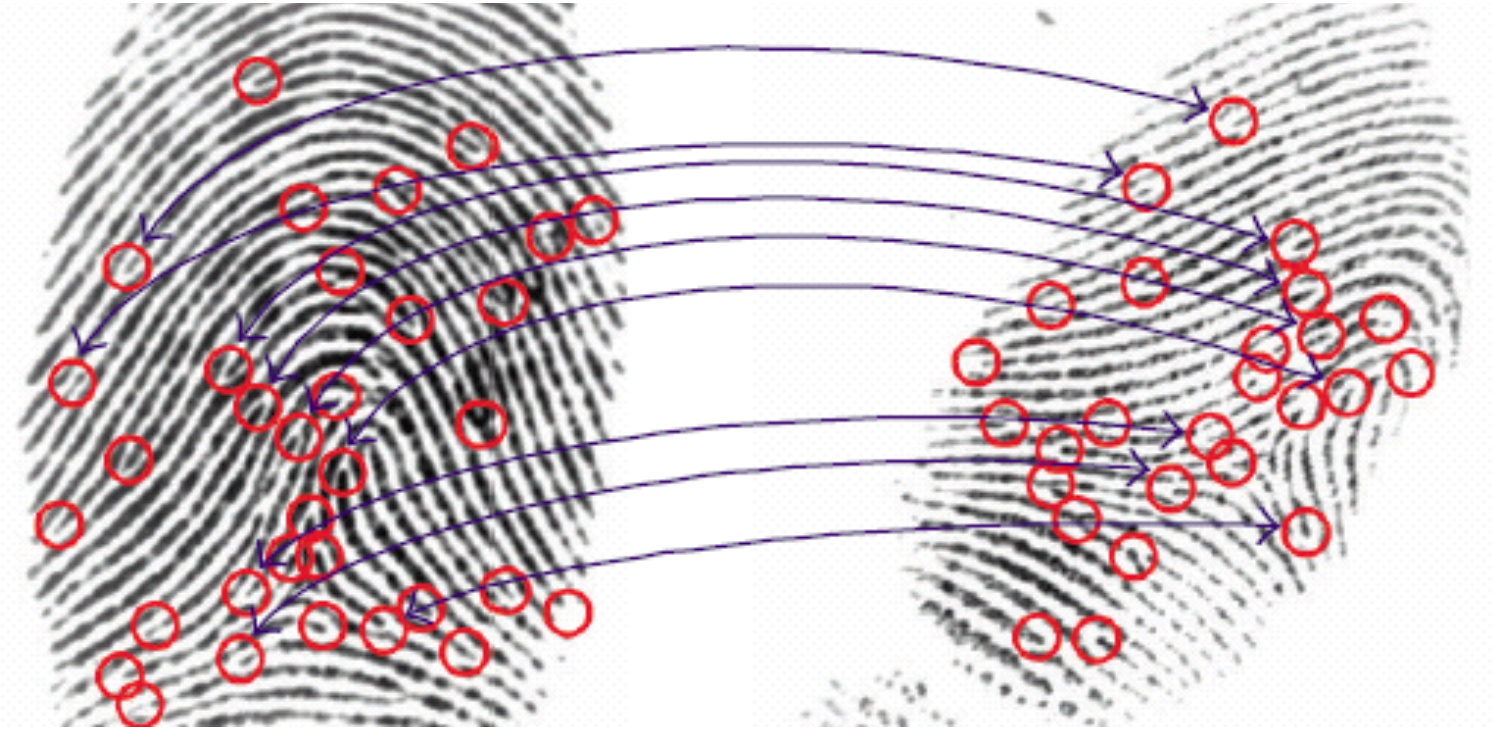
Minuzie

- rappresentano la **fine** e il **punto di biforcazione** delle linee
- uniche per individuo



Riconoscimento di impronte (2)

In fase di autenticazione vengono confrontati i punti di minuzie



Vantaggi/Svantaggi

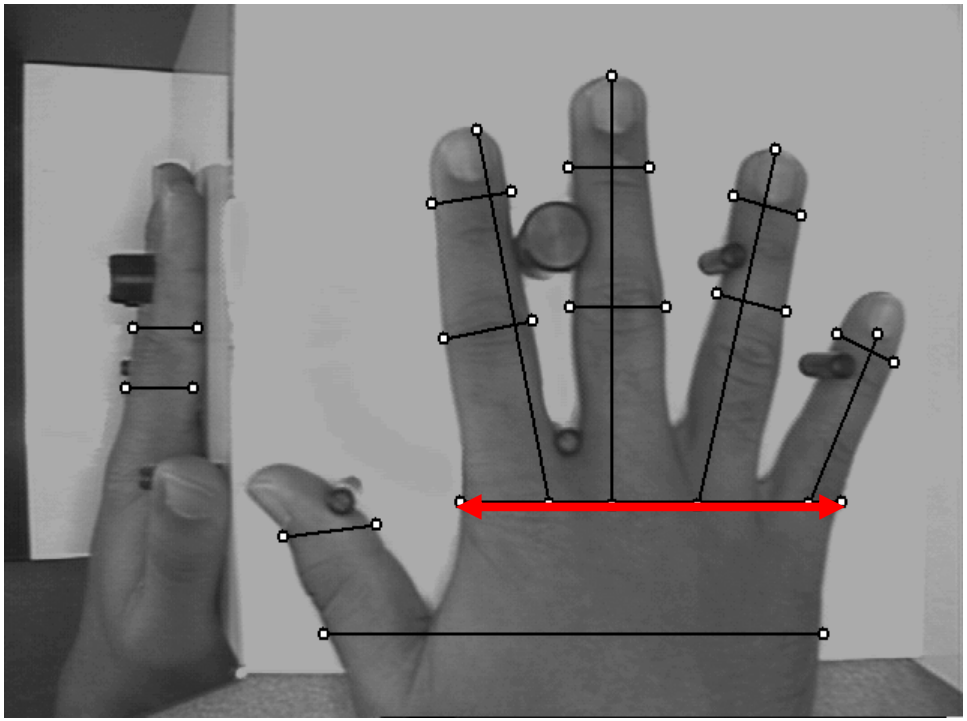
Vantaggi:

- Non intrusivo
- Facile da usare
- Economico
- Di piccole dimensioni
- Low power

Svantaggi:

- Ferite e/o cicatrici possono falsare il campione
- Esistono persone con poche minuzie

Forma della mano



Vantaggi/Svantaggi

Vantaggi

- Più affidabile delle impronte digitali
- Template piccoli
  - 10 byte, mentre per le impronte digitali 250-1000

Svantaggi

- Richiede scanner molto grandi
- Più costoso dei dispositivi per il rilevamento delle impronte digitali
- Come gestire (frequenti) ferite alle mani?

Riconoscimento facciale

Richiede un'espressione neutrale

Vantaggi:

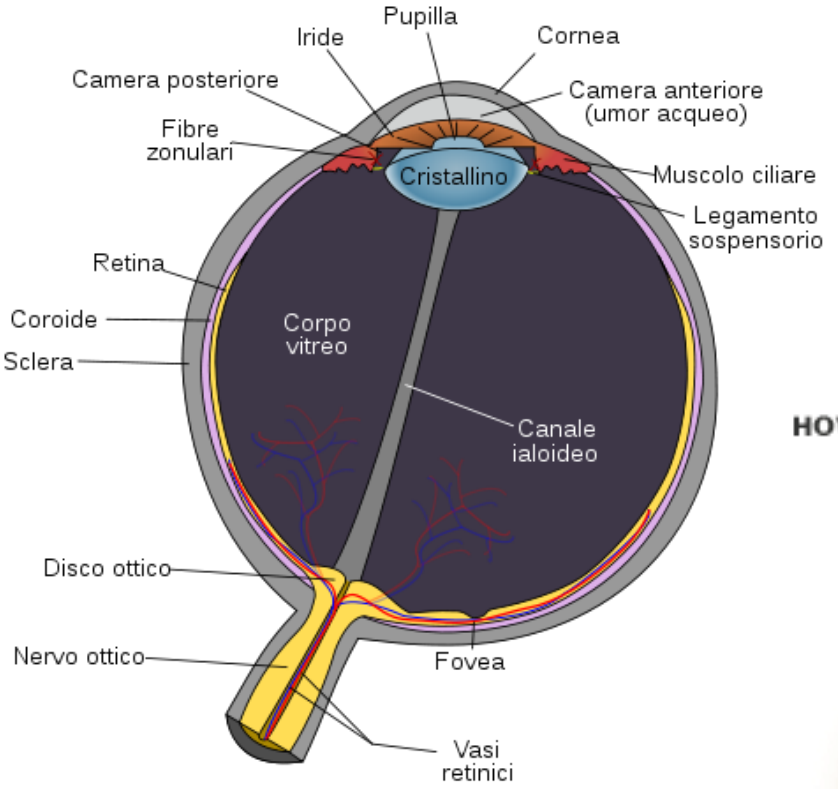
- Non intrusivo
- La rilevazione può avvenire anche ad una certa distanza

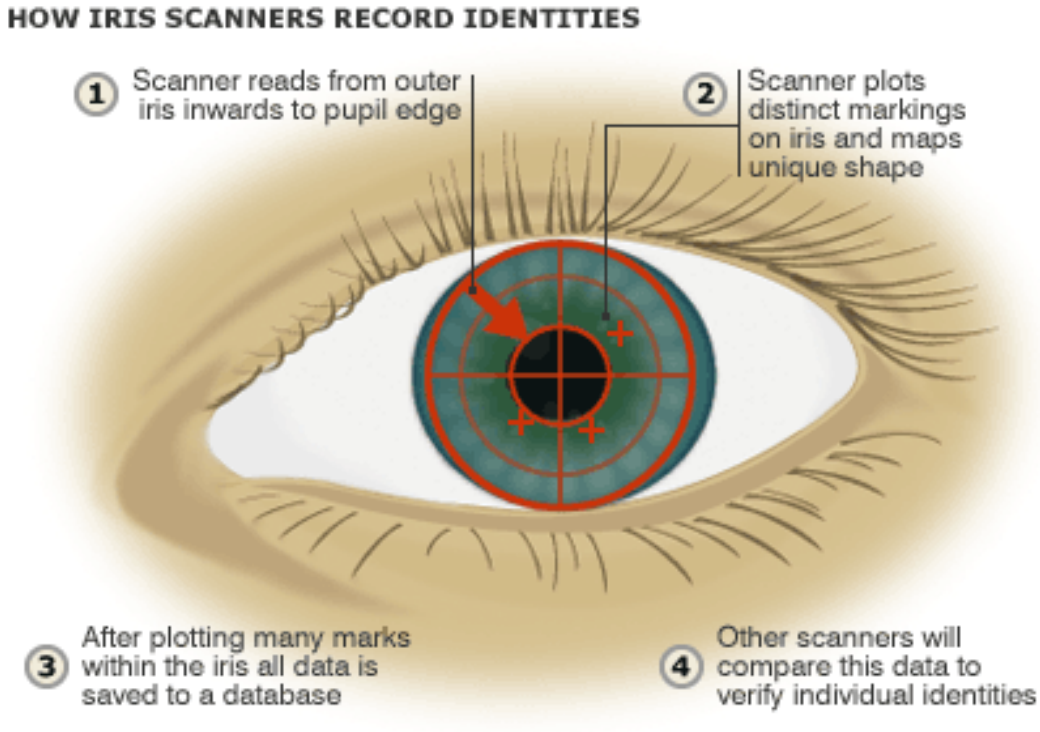
Svantaggi:

- Non sempre accettato il fatto che venga fatta una foto
- Fluttuazione dei valori FRR e FAR
- **Quale espressione?**

Scanning iride (1)

- Occhiali, lenti a contatto e operazioni con il laser non modificano l'iride





Scanning iride (2)

Vantaggi:

- Molto accurato

Svantaggi:

- L'utente deve guardare "esattamente" nel lettore e per un tempo piuttosto lungo
  - FTE (fail to enroll) alto
- Esposizione alla luce (anche se bassa)

Alcuni confronti

Caratteristica biometrica	Universale?	Unico?	Permanenza	Archiviabile?	Performance	Accettazione	Attacchi
Faccia	alto	basso	medio	alto	basso	alto	basso
Impronta digitale	medio	alto	alto	medio	alto	medio	alto
Forma della mano	medio	medio	medio	alto	medio	medio	medio
Iride	alto	alto	alto	medio	alto	basso	alto

Vedere anche: `http://www.bromba.com/faq/biofaqe.htm`

Applicazioni correnti

- Banche
- Immigrazione negli USA
- Ambito militare
- Giochi olimpici in Grecia, 2004: i tedeschi avevano una smart card con impronta digitale
- Aeroporto di Francoforte, dal 2000, dipendenti autenticati tramite scanning dell’iride

Vantaggi/Svantaggi (2)

Ancora poco utilizzata: costosa e intrusiva

- Non sempre accettata dagli utenti
- Gli scanner di retina sono accurati ma si temono conseguenze sulla retina…

Dibattiti politici e sociali per potenziale mancanza di *privacy*

Quale tecnica di autenticazione? (1)

- Tecnicamente la più forte è quella basata sulle caratteristiche univoche dell'utente
- Bilancio costi-benefici: metodi più deboli possono andare bene in certi casi
- Le password sono il meccanismo più antico ma sono e saranno nel breve futuro quello più utilizzato
- Combinare le tecniche:
  - Google 2-step verification
- Da un punto di vista prettamente tecnico il miglior metodo di autenticazione è:
  - autenticazione biometrica fra utente e token
  - mutua autenticazione basata su crittografia fra token e sistema

Quale tecnica di autenticazione? (2)

Aggiungere altri tipi di controllo:

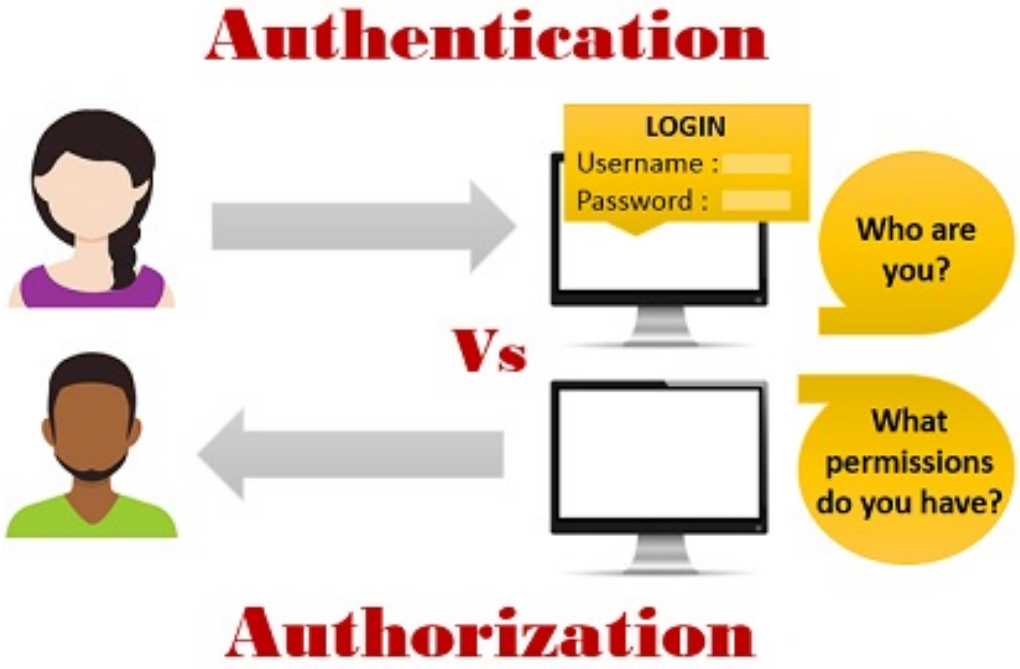
- Dove si trova l'utente?
  - Dal PC dell'ufficio?
  - Da un qualsiasi terminale utente?
  - Da quale zona geografica?
  - Da dentro la LAN?
- In che fascia oraria si sta loggando?

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
- **"Off-line":** i servizi possono prendere decisioni autonome anche senza dover contattare ogni volta l'autorità di autenticazione

Autenticazione vs Autorizzazione

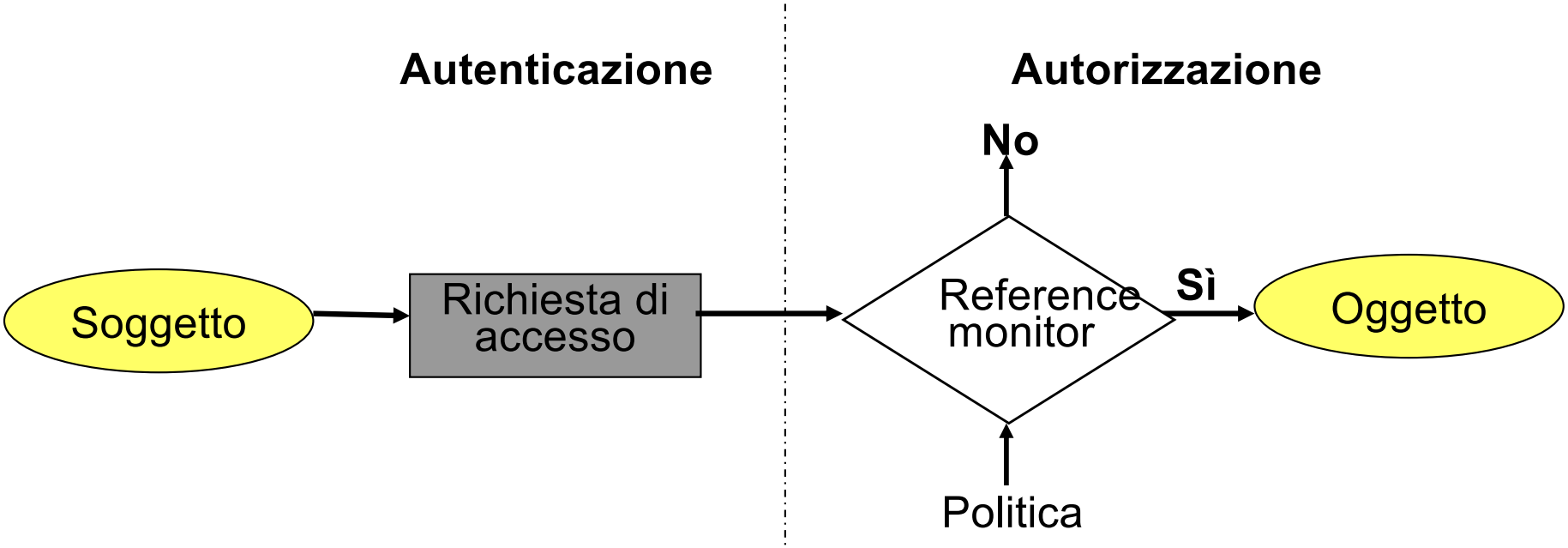


The authentication and authorization are used in respect of information security which enables the security on an automated information system. The terminologies are interchangeably used but are distinct. The identity of a person is assured by authentication. On the other hand, authorization checks the access list that the authenticated person has. In other words, the authorization includes the permissions that a person has given.

Reference Monitor

Funzionamento:

- Un *soggetto* attivo *chiede accesso* ad un *oggetto* passivo per eseguire una specifica *operazione di accesso*
- Il *monitor* concede o meno l'accesso



Correttezza del controllo dell'accesso implica:

- Corretta *identificazione/autenticazione*
- Corretta *definizione* (e implementazione) della *politica di sicurezza*

Autenticazione utente-computer

Basata su qualcosa che l'utente:

Conosce: un informazione segreta

- Password, PIN, …

Possiede: cosa fisica

- Chiavi convenzionali, carte magnetiche o smart card

È: caratteristiche biometriche

- Impronte digitali, dell'iride, tono di voce,…

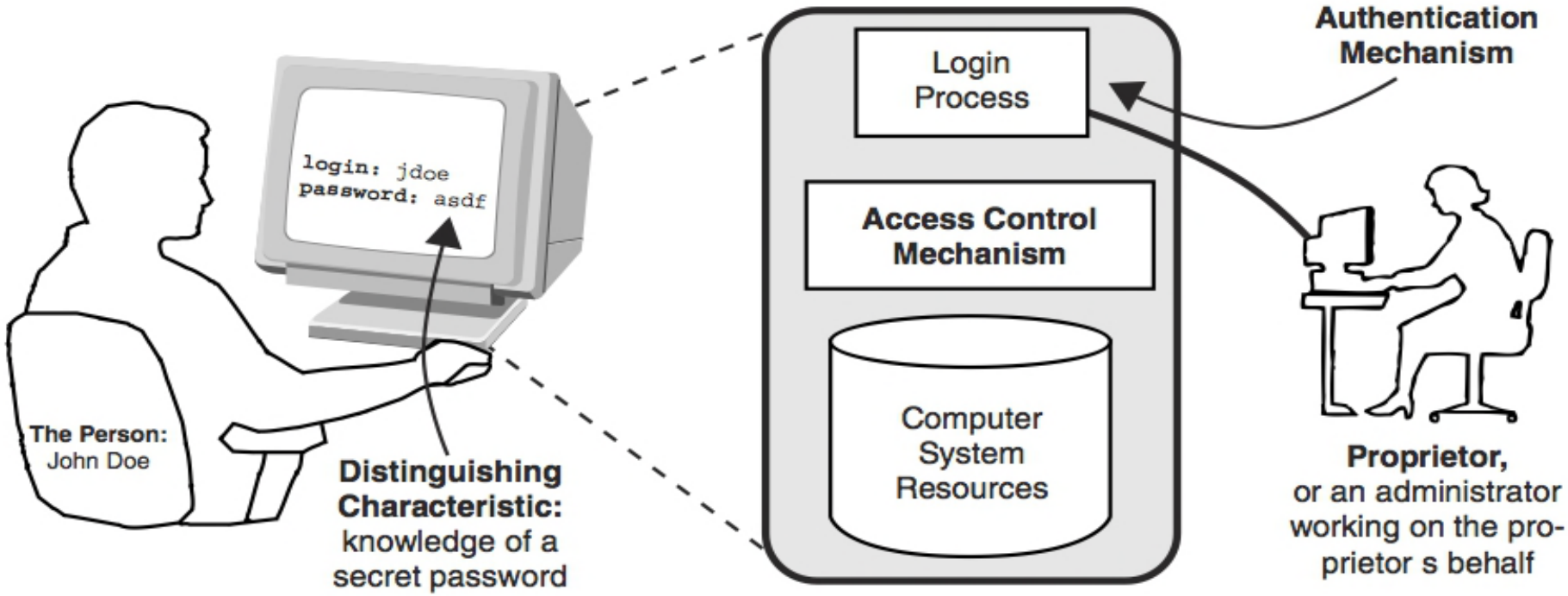
Tipologie di autenticazione

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

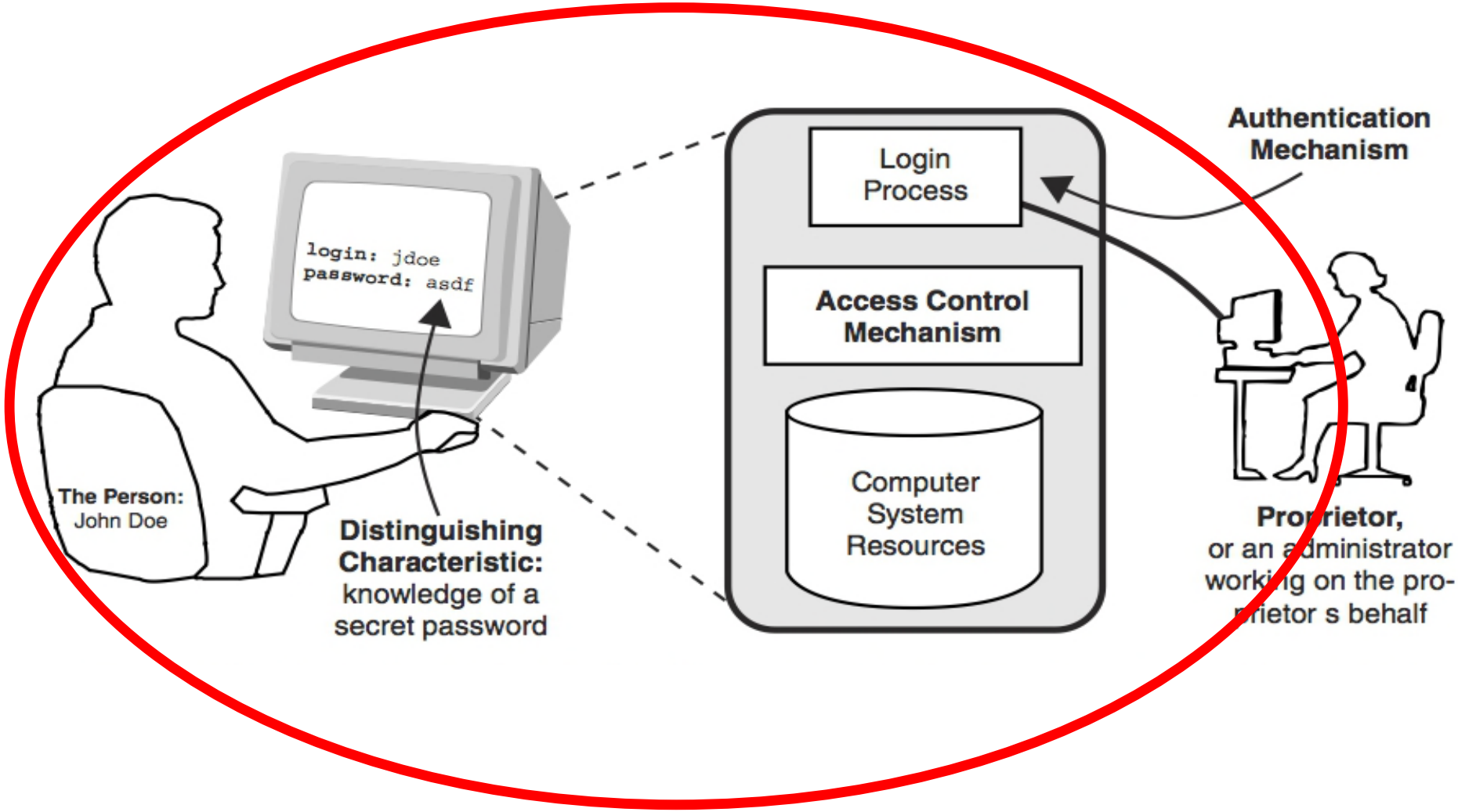
- **Indiretta**: l'utente accede da remoto a diversi servizi, che si appoggiano su un servizio di autenticazione separato
- **"Off-line":** i servizi possono prendere decisioni autonome anche senza dover contattare ogni volta l'autorità di autenticazione

Autenticazione utente-computer



L'utente lavora su di un terminale con una connessione **locale e diretta** ad un host

Autenticazione utente-computer



Assunzione: il server **NON** è remoto

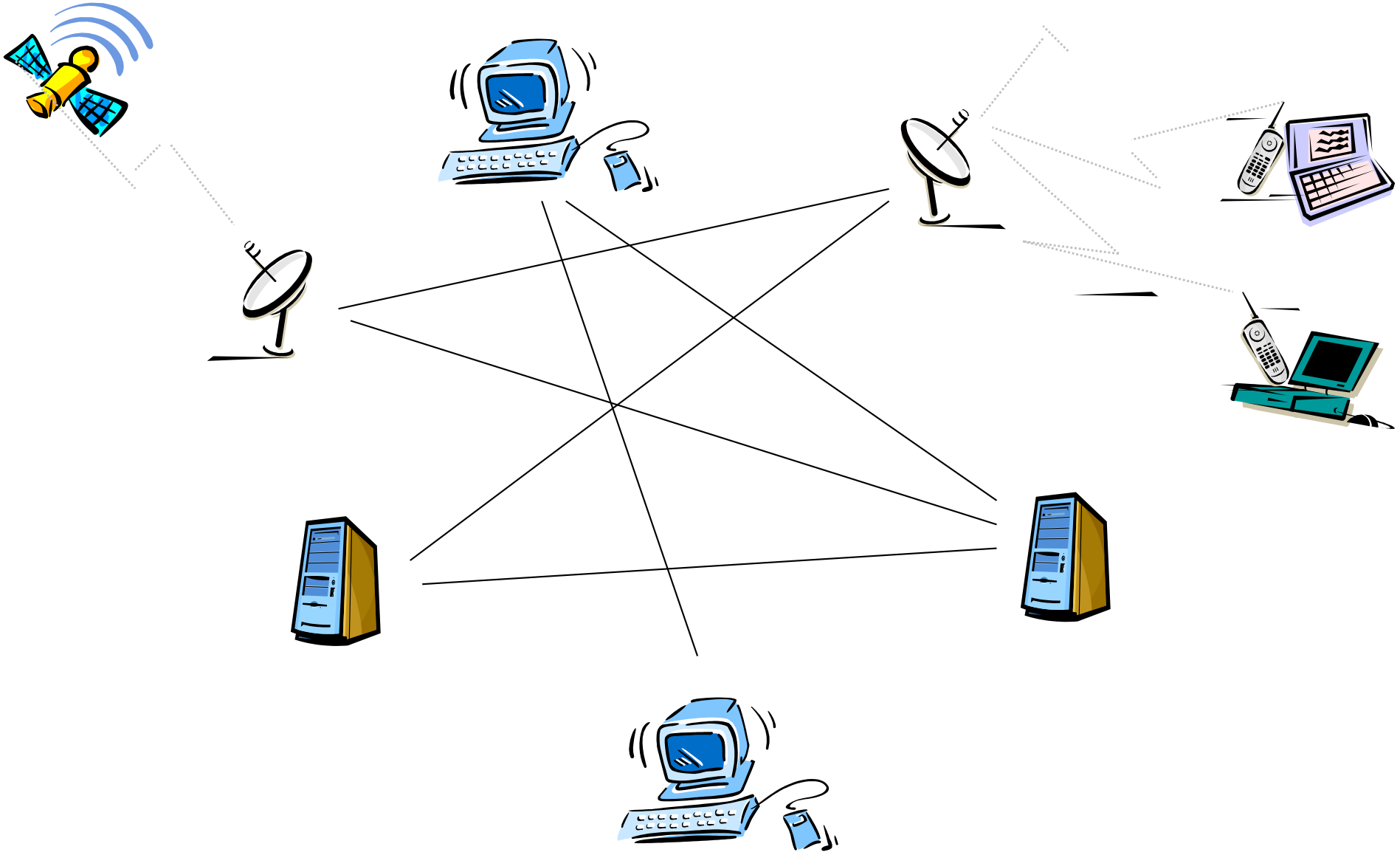
Autenticazione in sistemi distribuiti

Esempio: Utente-Applicazione Web?

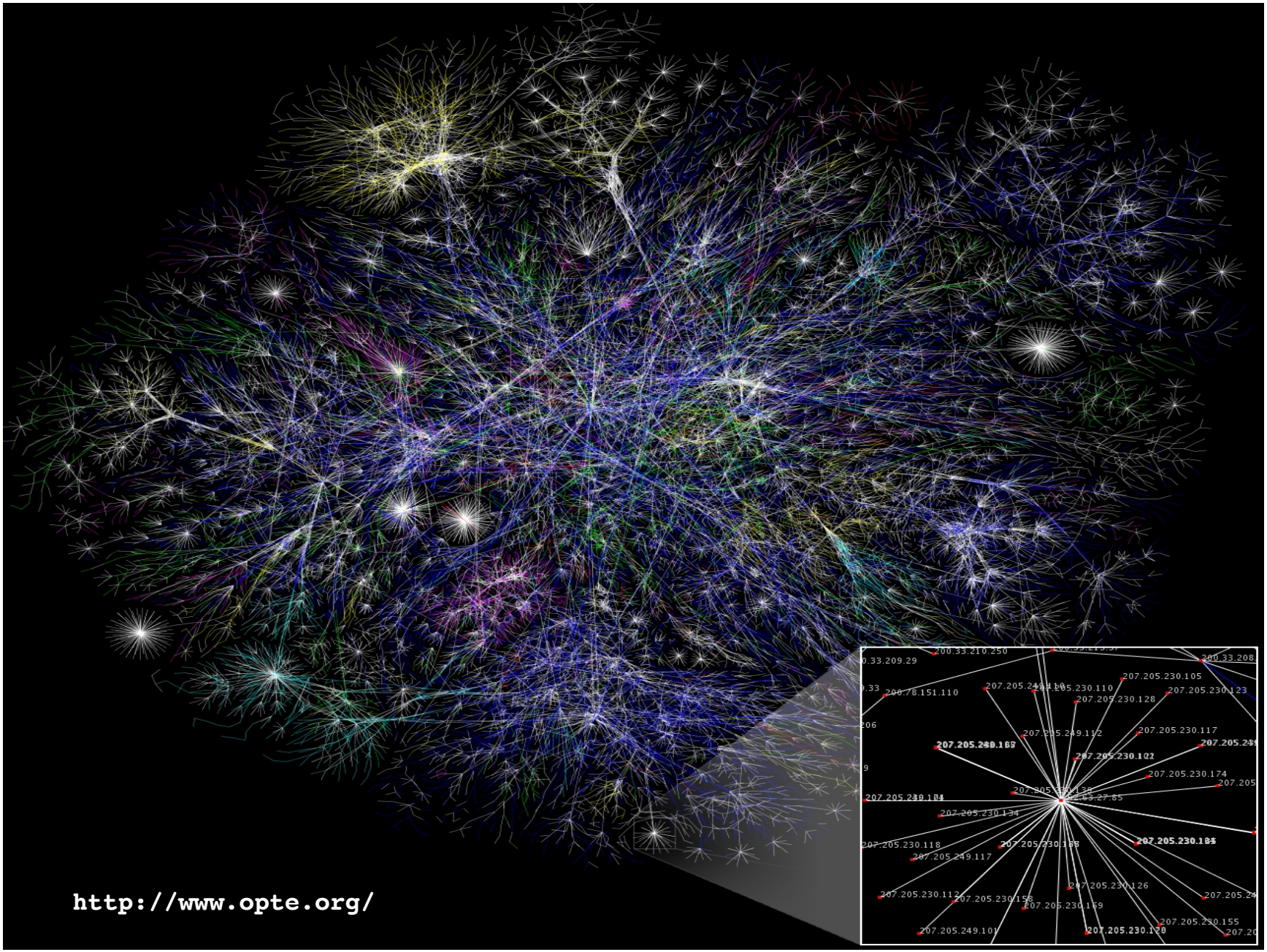
Problema: il server **è** remoto

Autenticazione in sistemi distribuiti?

Internet, Web e mobilità



Internet in realtà

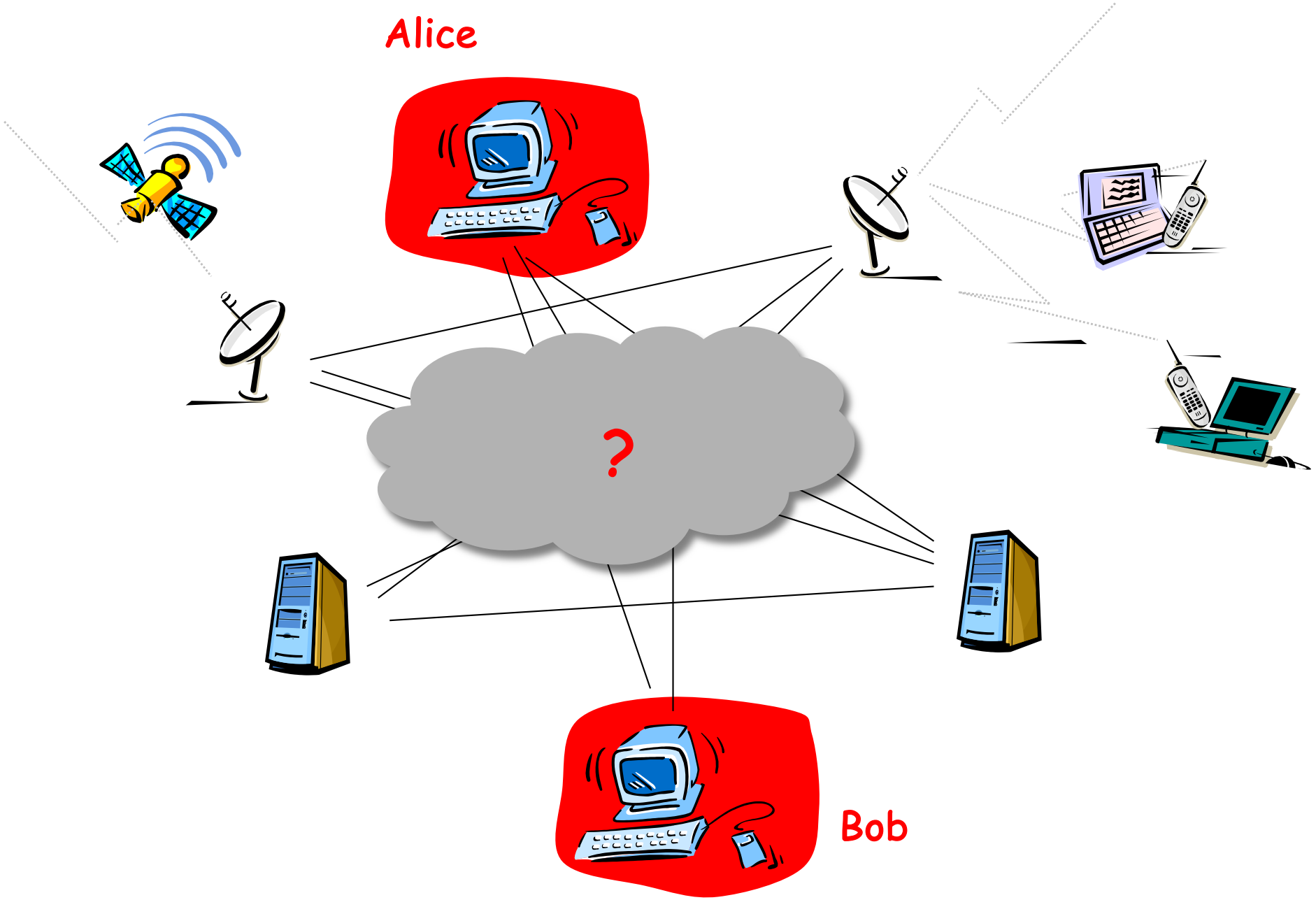


Internet canale insicuro

Perché insicuro?

- Un canale è un mezzo che due entità utilizzano per comunicare
- Nel caso in cui il canale sia a disposizione di altre entità oltre a quelle che devono comunicare tra loro il canale si dice **insicuro**
- Internet è un canale **condiviso** da migliaia di utenti quindi è un canale molto insicuro!

Comunicazione tramite Internet



Perché tanto interesse?

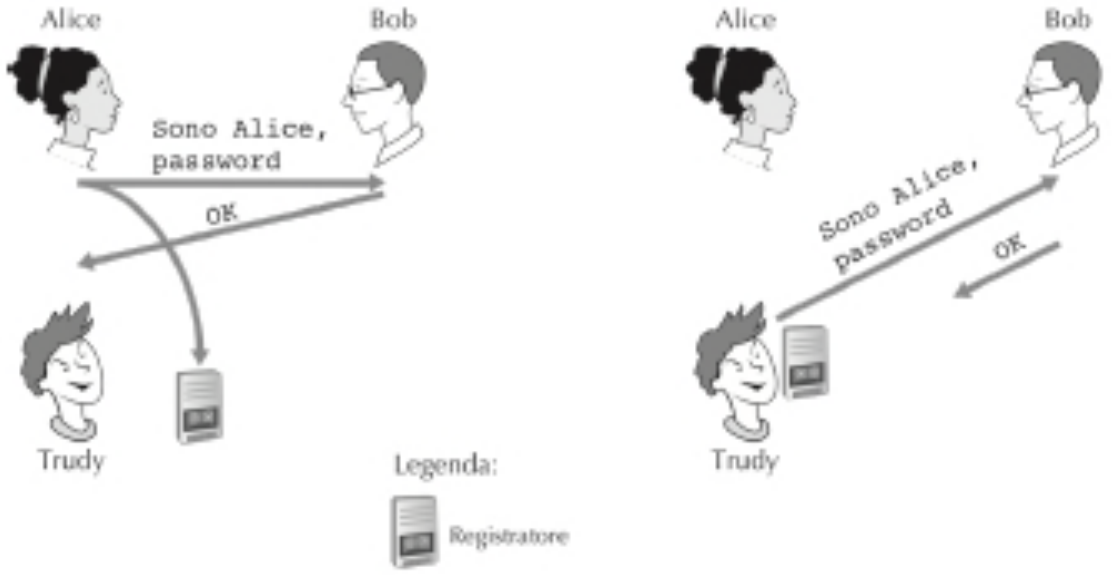
Chi possono essere Alice e Bob?

- *browser/server Web* per transazioni elettroniche (e.g., acquisti on-line)
- *on-line banking* client e server
- *router* che si scambiano gli aggiornamenti delle tabelle di routing
- … beh, Bob e Alice *in carne e ossa*, che si scambiano messaggi privati!

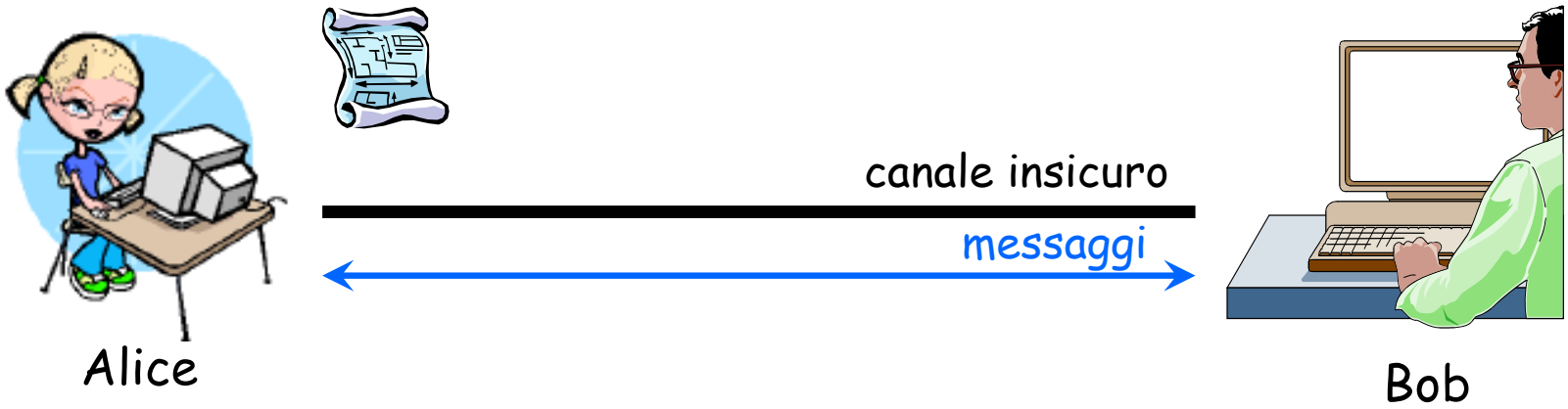
In tutti i casi:

- L'informazione deve venire *protetta* dalla divulgazione e modifica lungo la rete
- L'*autenticazione dell'utente* è necessaria: quando uno fa login ci deve essere un modo per verificarne l'identità

Autenticazione da remoto: problema



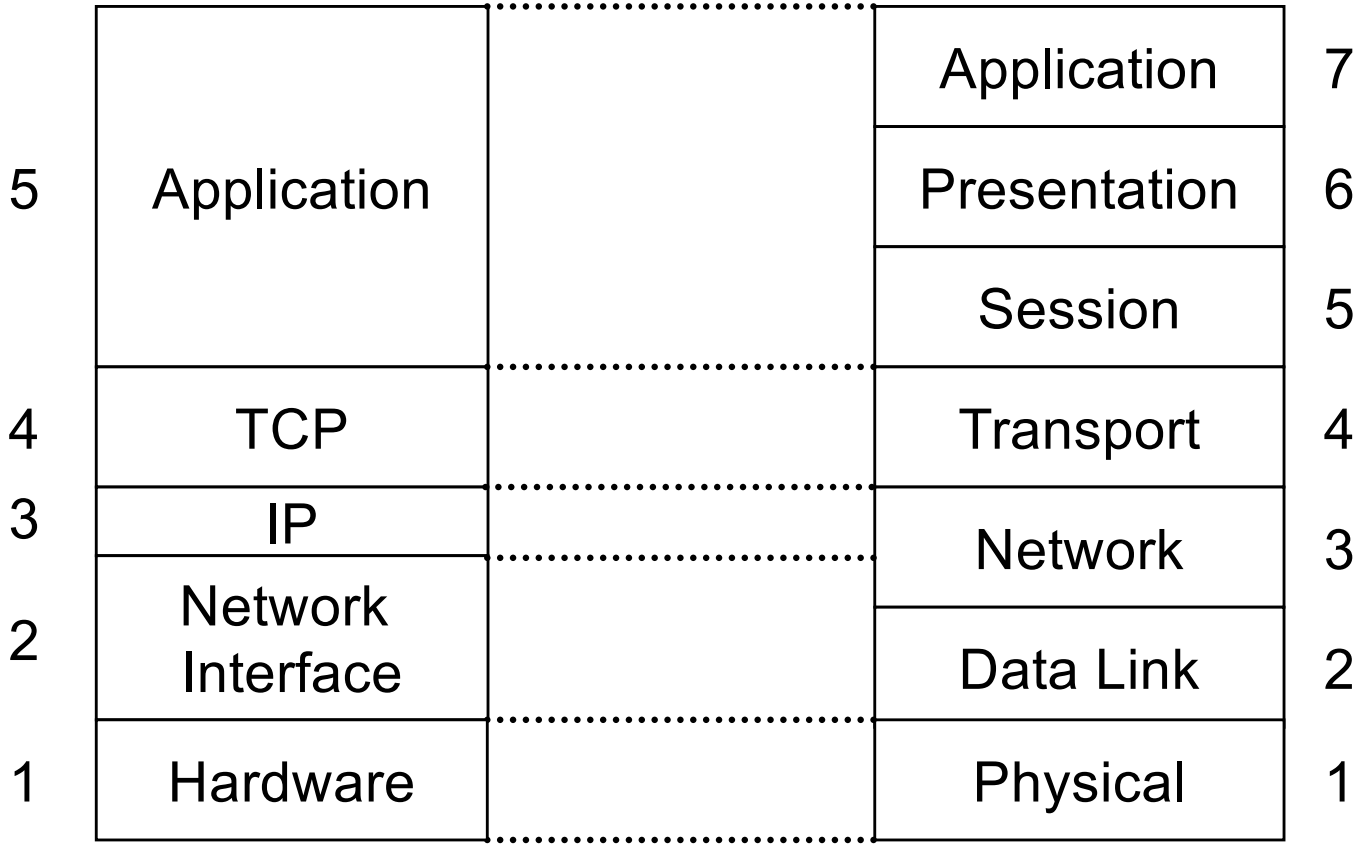
Altre proprietà oltre all'autenticazione?



- Confidenzialità: solo Alice e Bob devono "capire" il contenuto dei messaggi
- Integrità: i messaggi non devono venire modificati
- Autenticazione: Alice e Bob vogliono essere sicuri che il loro interlocutore sia effettivamente chi loro si aspettano
- Non-repudiation: Alice non può negare di aver spedito un messaggio a Bob (e viceversa)

Come comunicano Alice e Bob?

Usano Internet come canale di comunicazione, quindi usano i protocolli dello stack TCP/IP:
stratificazione di programmi e protocolli



Cos'è un protocollo di comunicazione? (1)

Protocollo "umano":

- "*Che ore sono?*"
- "*Ho una domanda*"

… messaggi specifici comunicati

… azioni specifiche intraprese quando un messaggio viene ricevuto

Protocolli di rete:

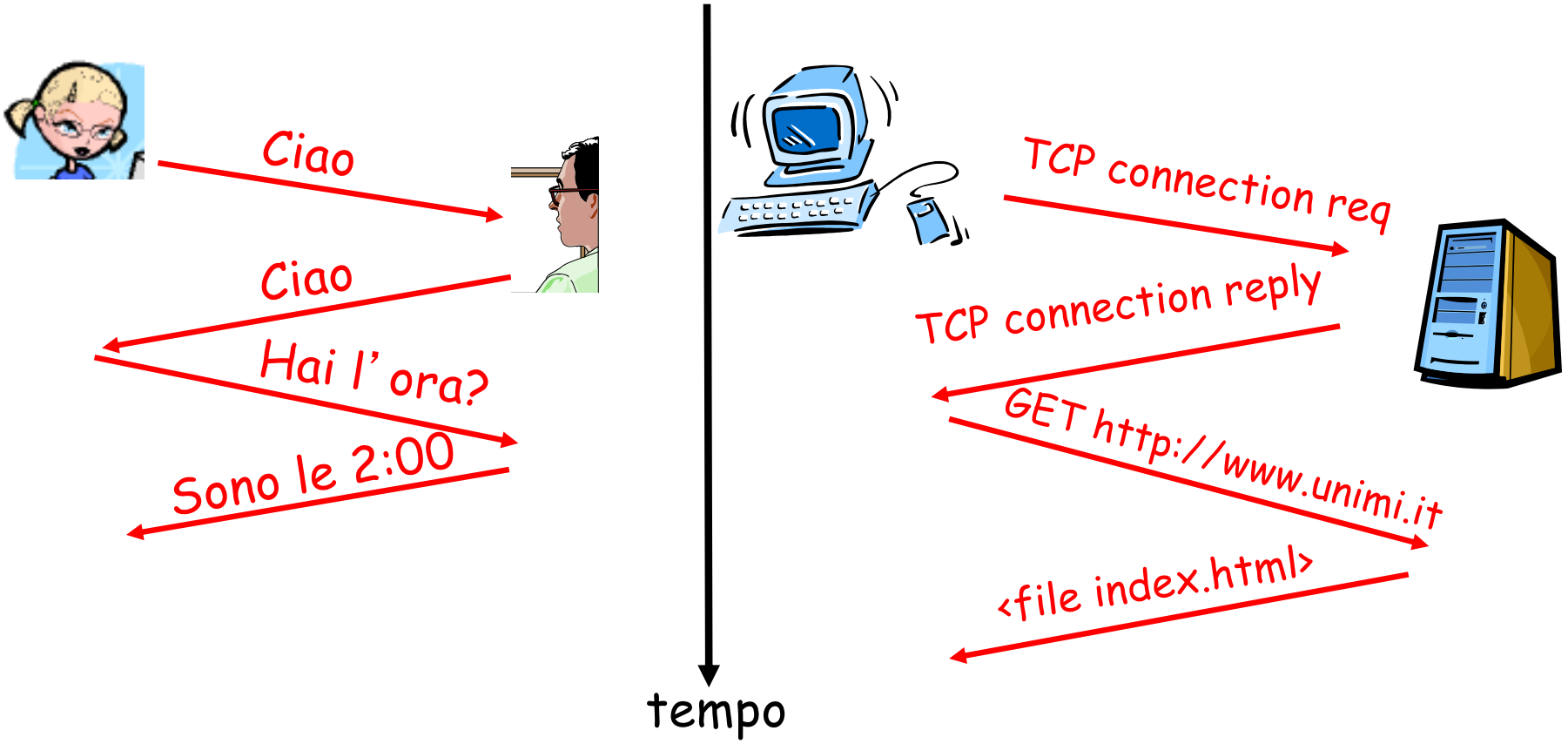
- PC invece che umani
- Trasmettono pacchetti da un host all'altro

Tutta la comunicazione di Internet viene governata da protocolli

I protocolli definiscono il formato, l'ordine dei messaggi spediti e ricevuti tra entità della rete, e le azioni intraprese dopo la trasmissione/ricezione

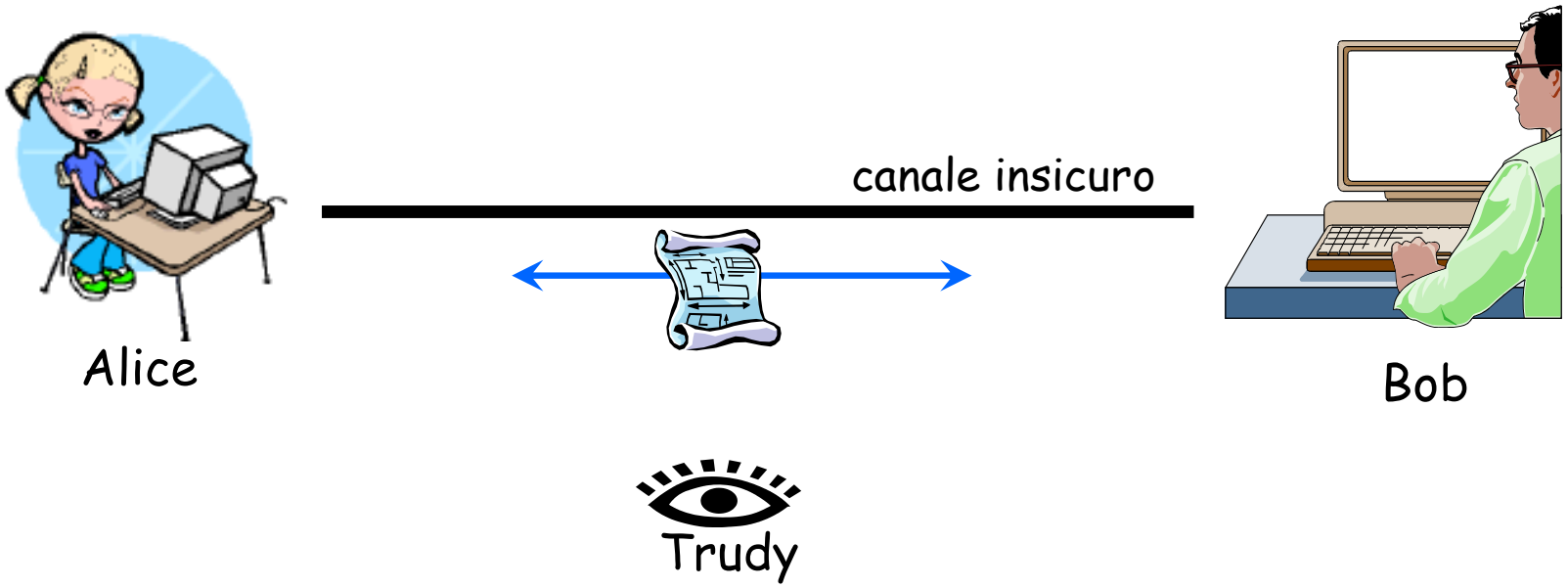
Cos'è un protocollo di comunicazione? (2)

Un protocollo umano e uno di reti di computer:



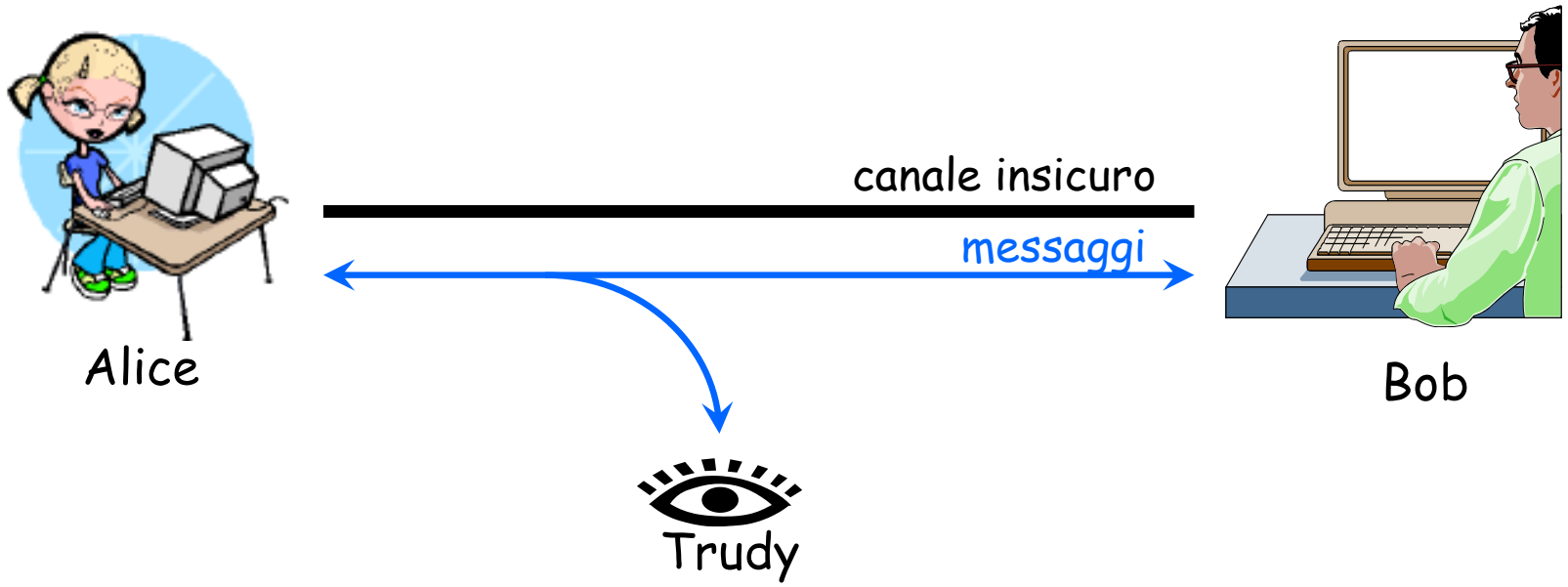
- In un protocollo *crittografico* parte o tutto il messaggio sono crittati

Cosa può fare un attaccante? (1)



L'attaccante è un host malevolo connesso ad Internet che può voler modificare il normale comportamento del protocollo per scopi illeciti

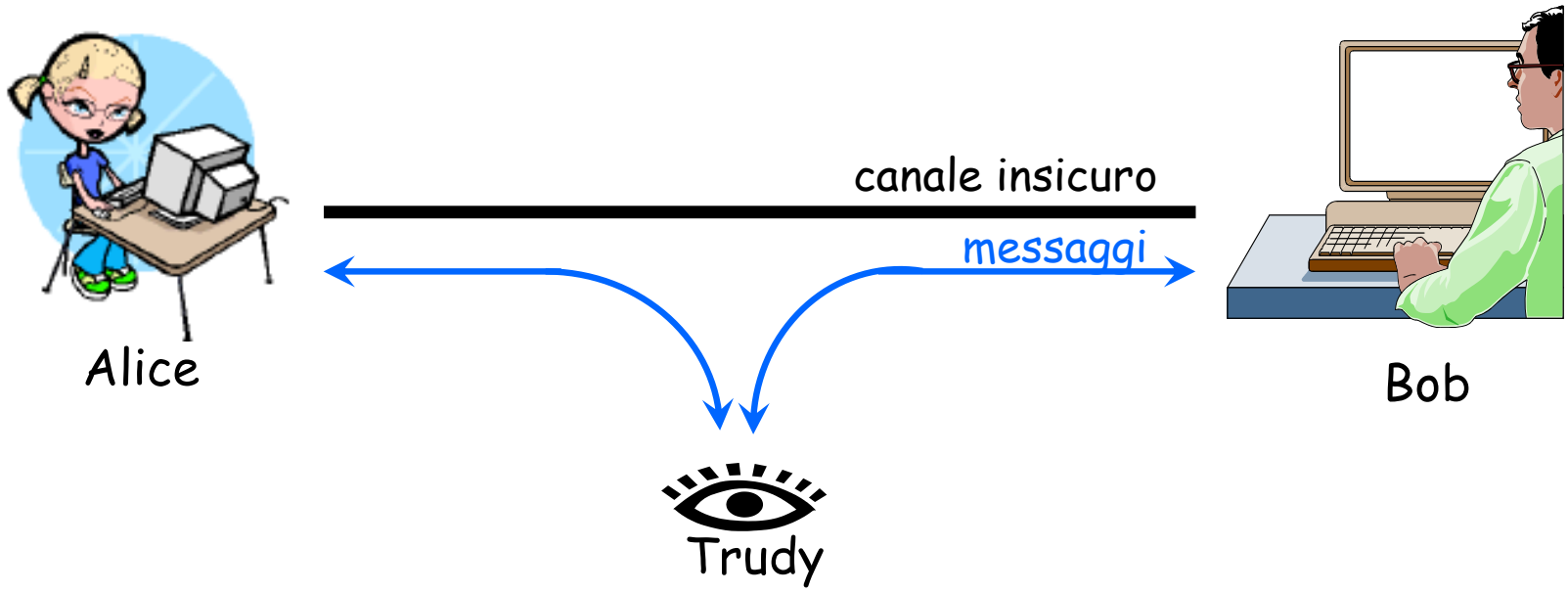
Cosa può fare un attaccante? (2)



L'attaccante può intercettare il messaggio ascoltando il canale: minaccia alla *confidenzialità*

- solo leggere il messaggio (*passivo*)
- cancellare il messaggio senza inoltrarlo a Bob (*attivo*)

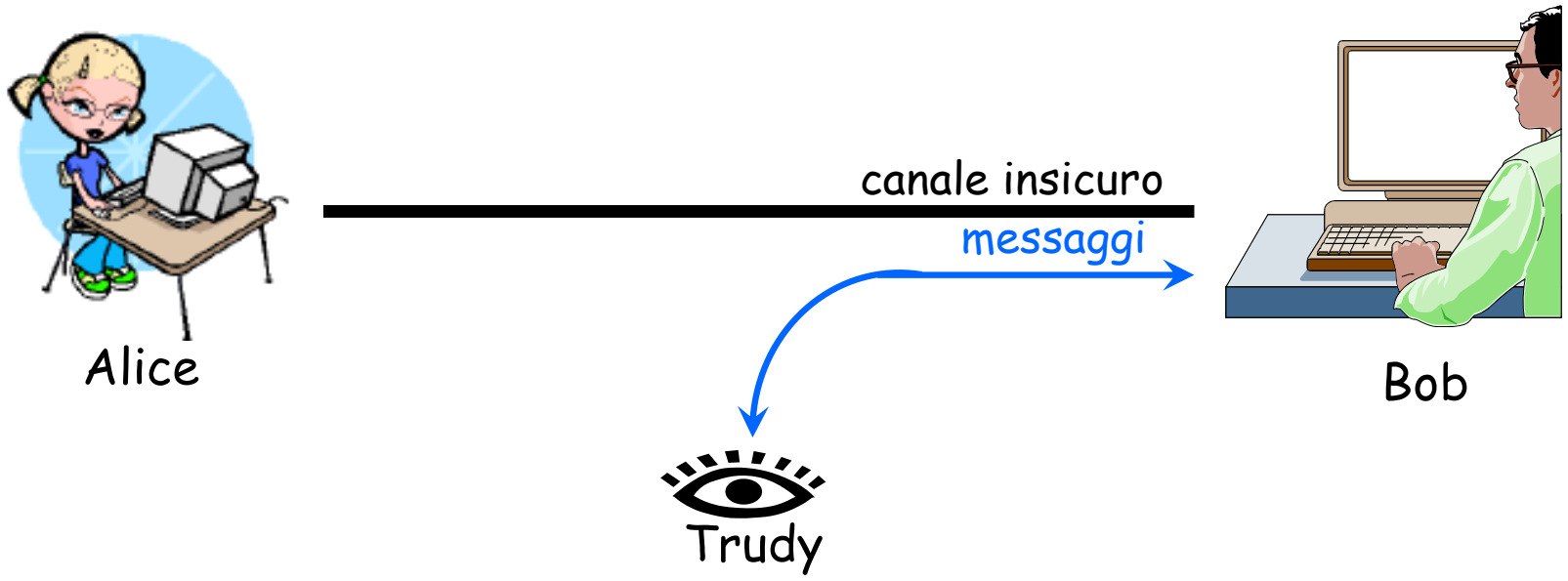
Cosa può fare un attaccante? (3)



L'attaccante può modificare i messaggi in transito: minaccia all'*integrità*

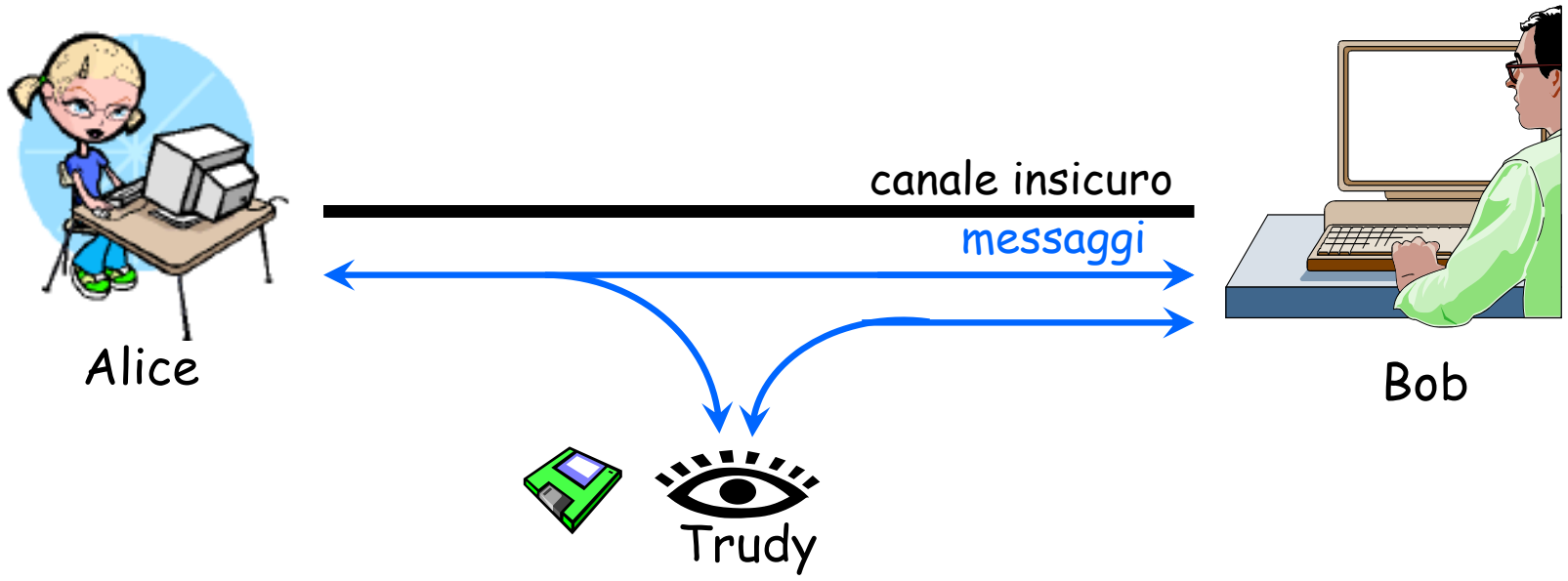
- Come può Bob distinguere tra un messaggio legittimo e uno contraffatto?

Cosa può fare un attaccante? (4)



L'attaccante può spedire messaggi arbitrari fingendo di essere Alice: minaccia all'*autenticazione.*

Cosa può fare un attaccante? (5)



L'attaccante può riutilizzare vecchi messaggi salvati (*replay attack*): minaccia all'*autenticazione.*

Soluzioni?

Crittografia - Schema generale (1)

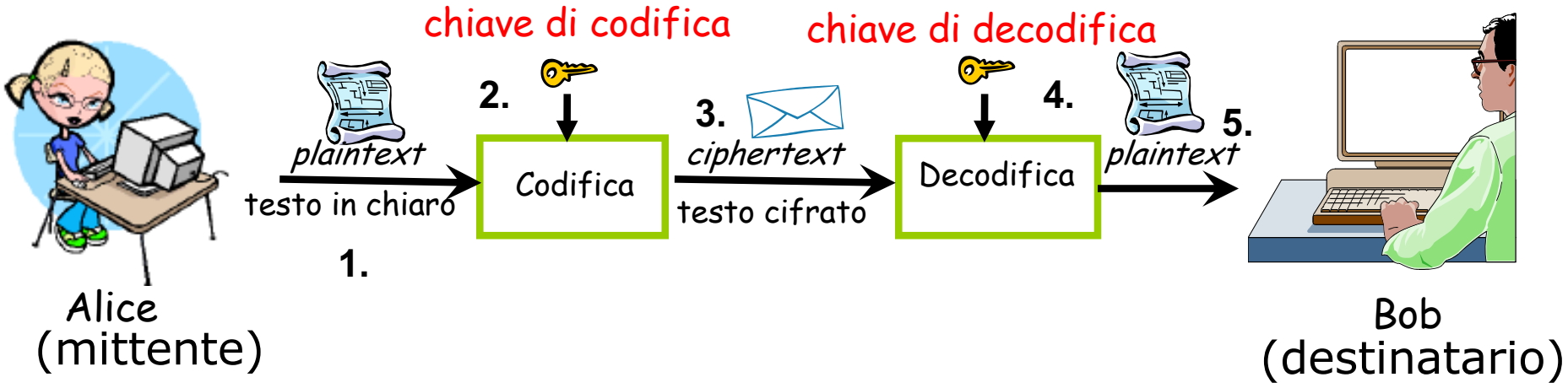
Alice
(mittente)

M
(messaggio)

Bob
(destinatario)

Trudy
(attaccante)

Crittografia - Schema generale (2)



Crittografia simmetrica:

- Stessa chiave per codifica e decodifica (da tenere segreta!)

Crittografia asimmetrica:

- Una coppia di chiavi per ciascun utente: *chiave pubblica* per la codifica, *chiave privata* per la decodifica

Attacchi comuni e contromisure (1)

Osservazione passiva dei messaggi

- Contromisura: Uso di crittografia per garantire segretezza

Modifica di messaggi

- Contromisura: Funzioni hash oppure crittografia per legare insieme informazioni

Replay

- Riutilizzo di vecchi messaggi
- Contromisura: specificare il contesto in modo che un messaggio non possa venire utilizzato più volte in una stessa esecuzione o in esecuzioni diverse del protocollo

Attacchi comuni e contromisure (2)

Attacco alla freshness

- Utilizzo di informazioni “vecchie” (chiave, nonce,…)
- Contromisura: utilizzare dei meccanismi per fare in modo che un partecipante legittimo riconosca se l’informazione è *fresh* o meno

Sessioni parallele

- L’attaccante combina i messaggi di sessioni diverse
- Contromisura: utilizzare dei meccanismi per fare in modo che un partecipante legittimo riconosca da quale sessione proviene il messaggio

Attacchi comuni e contromisure (3)

Reflection Attack

- Un tipo di *replay attack* che utilizza il protocollo contro se stesso (e quindi fa uso anche di una *sessione parallela*):
  - L'attaccante rispedisce l'informazione o il messaggio al mittente che l'ha spedito, facendo in modo di ottenere da lui il messaggio che deve spedirgli nella sessione del protocollo in cui i ruoli sono invertiti
- Contromisura: specificare il destinatario designato (quindi legittimo)

Attacchi comuni e contromisure (4)

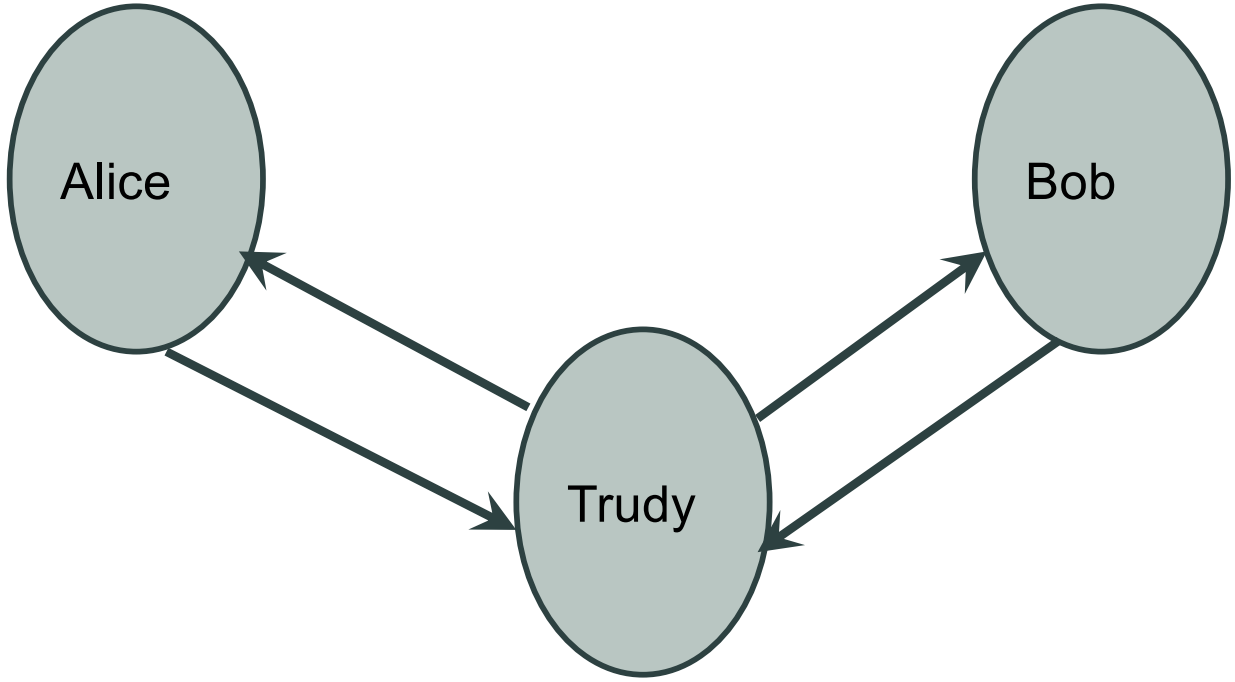
Men-in-the-middle attack

- L'attaccante crea delle connessioni indipendenti con i 2 partecipanti e li impersona (entrambi **o solo uno**)

Attacchi comuni e contromisure (5)

Men-in-the-middle attack

- L'attaccante si "mette in mezzo" alla normale esecuzione di un protocollo
- Fa uso di sessioni parallele dello stesso protocollo
- Ha senso solo quando Trudy ne trae vantaggio



Attacchi comuni e contromisure (6)

Men-in-the-middle attack

- Contromisura: specificare il mittente e/o il destinatario designato del messaggio

Quindi…

Si deve parlare di protocolli di autenticazione:

- per proteggere anche durante la trasmissione l'informazione

- "estendendo" la definizione di autenticazione che abbiamo dato

Autenticazione nella Rete – Definizioni

Entity Authentication (def. in standard ISO 1991)

- Garantisce l'identità dell'utente con cui in tempo reale si stanno scambiando messaggi all'interno di una sessione di protocollo

Message Authentication

- Garantisce che il messaggio ricevuto proviene dalla sorgente indicata, senza dare alcuna indicazione di quando il messaggio sia stato effettivamente creato

Autenticazione nella Rete – Precisazione

Nel mondo reale

Sto parlando con Mario

Sto parlando con qualcuno che **possiede** le seguenti credenziali di Mario:

- Password
- Chiave privata
- Impronta digitale
- …

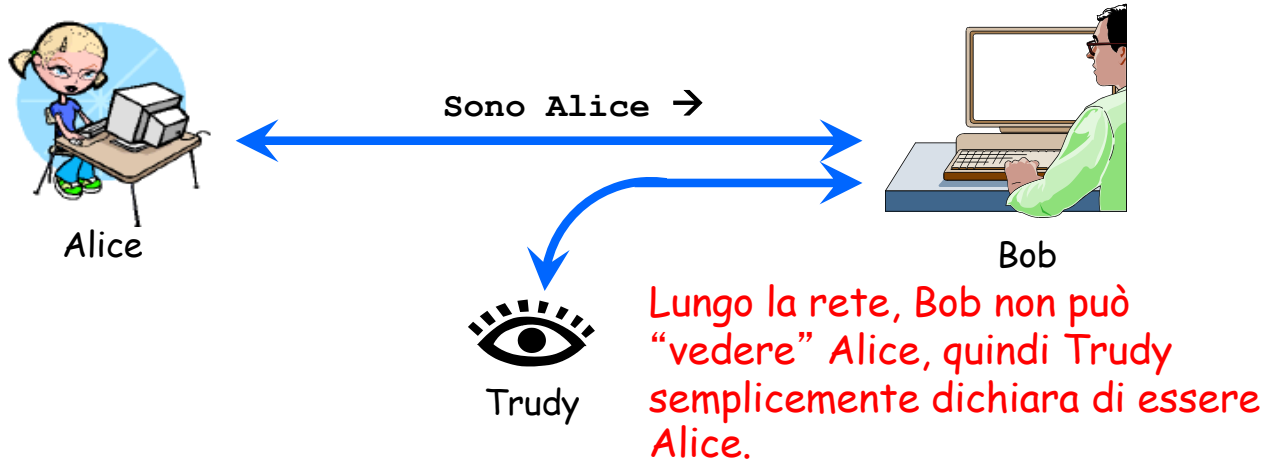
Nella Rete

Proviamo a definire insieme un semplice protocollo di autenticazione?

Protocollo di Autenticazione (1)

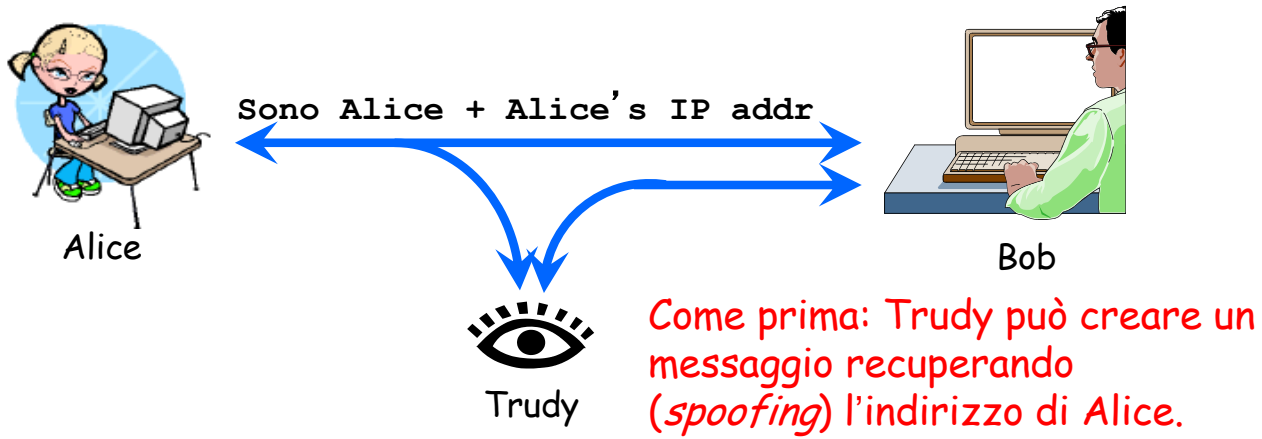
Obiettivo: Bob vuole che Alice "dimostri" la sua identità.

Protocollo 1: Alice dice "Sono Alice".



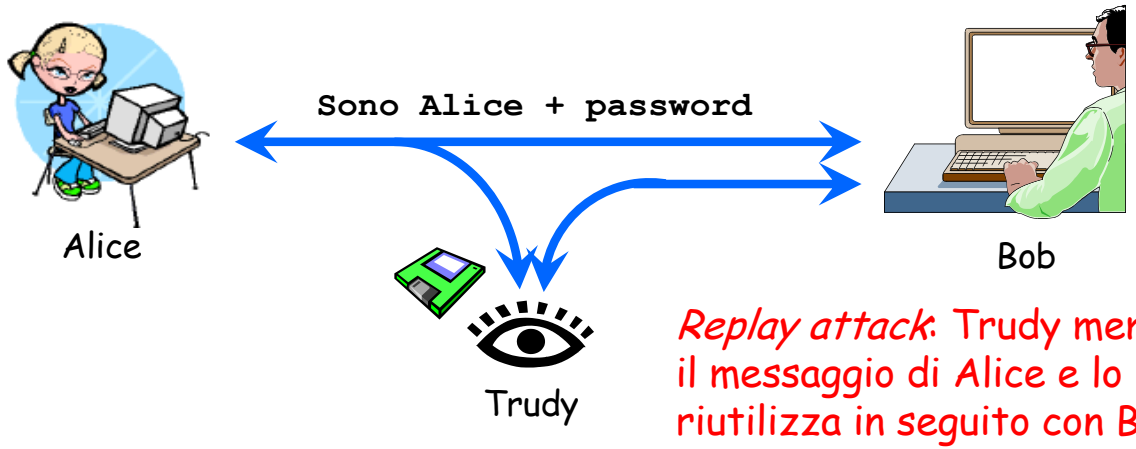
Protocollo di Autenticazione (2)

Protocollo 2: Alice dice “Sono Alice” e spedisce il suo indirizzo IP per “provarlo”.



Protocollo di Autenticazione (3)

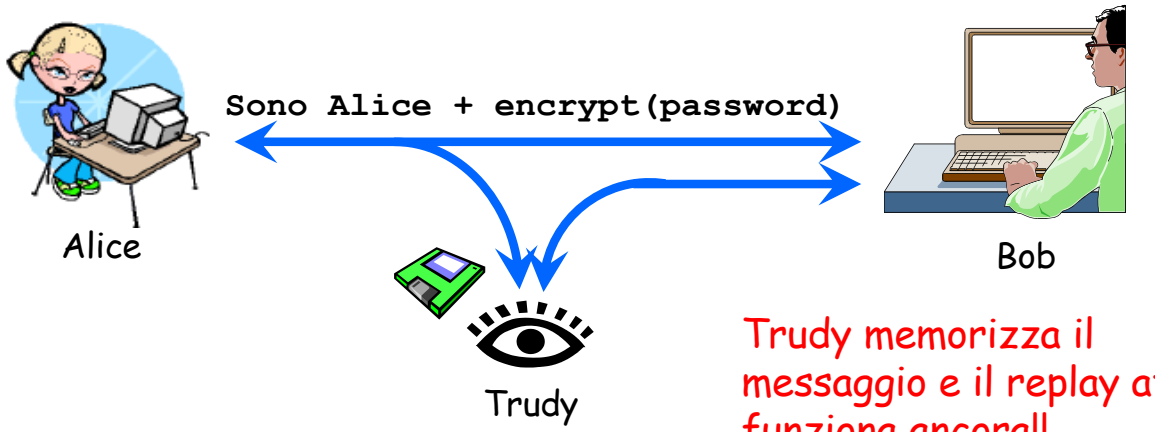
Protocollo 3: Alice dice “Sono Alice” e spedisce la sua password (segreta…) per “dimostrarlo”.



Replay attack: Trudy memorizza il messaggio di Alice e lo riutilizza in seguito con Bob.

Protocollo di Autenticazione (4)

Protocollo 4: Alice dice “Sono Alice” e spedisce la sua password segreta cifrata per “dimostrarlo”.



Protocollo di Autenticazione (5)

Obiettivo (aggiornato): Fare in modo che Alice “dimostri” la sua identità e che non ci siano replay attack.

Alice

Bob

Trudy

Protocolli challenge-response

Protocolli challenge-response (1)

Chiamato anche test di autenticazione

- Usato per ottenere autenticazione **mutua** o **unilaterale**
- Di solito rappresenta la parte di set-up di un protocollo più grande

- Un partecipante prova la propria identità dimostrando di conoscere un segreto (non necessariamente condiviso) e un'informazione che varia nel tempo (per evitare replay attack), senza rivelare esplicitamente il segreto

Protocolli challenge-response (2)

Funzionamento di base (con chiave condivisa):

- **Challenge (sfida)**
  - il sistema o l'altro partecipante presenta all'utente una *stringa* contenente l'informazione che varia nel tempo
- **Response (risposta)**
  - L'utente cifra la stringa con la chiave condivisa: $\{stringa\}_K$ e la spedisce all'altro partecipante
- **Autenticazione**
  - Il partecipante controlla la $\{stringa\}_K$

NB: I dati segreti rimangono tali: non viene spedita in chiaro né una password né una chiave segreta

Protocolli challenge-response (3)

Sia il challenge che il response dovrebbero essere "fresh"

- Altrimenti il *replay attack* è possibile
- Attacchi con sessioni parallele sono sempre possibili …

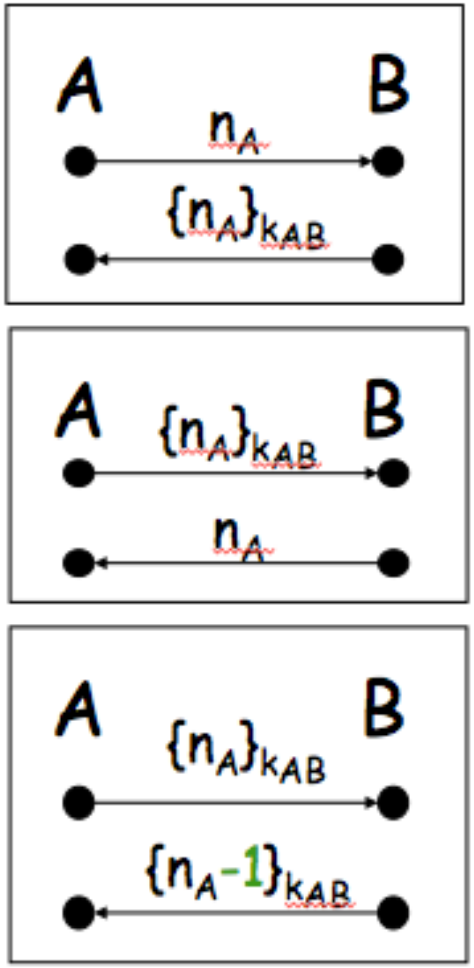
L'informazione che varia nel tempo (detta Nonce dal 1978) può essere:

- Numero casual (detto anche nonce)
- Timestamp
- Numeri in sequenza
- Chiavi a breve termine

Protocolli challenge-response – Nonce (1)

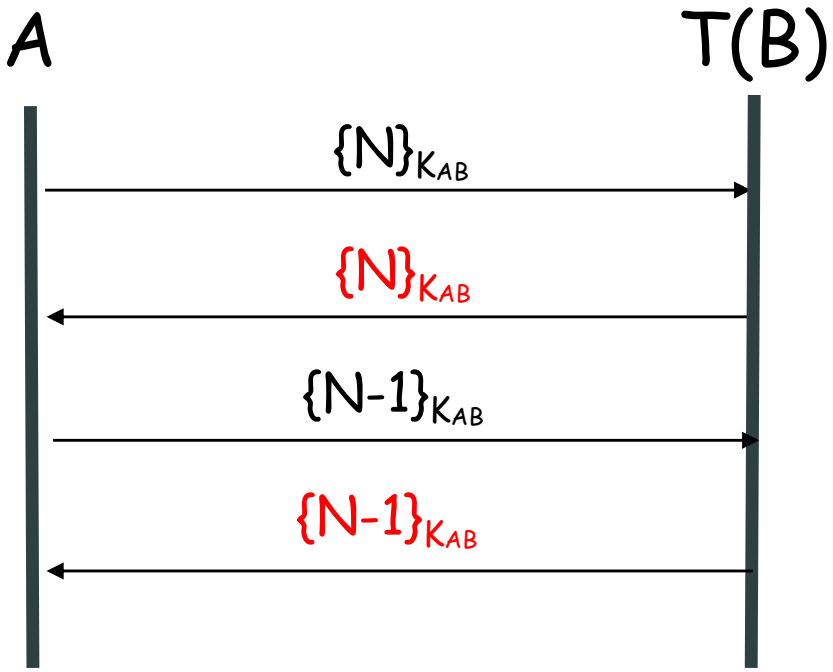
- Sequenza casuale di bit (32-128)
- Generata ogni volta (fresh!) dal mittente come challenge
  - **Non prevedibile**
  - Verificata solo nel response
- Non controllata dal destinatario
  - Poco pratico memorizzarla
- Mai riutilizzata

Q: Possibile il reflection attack (come?)



Protocolli challenge-response – Nonce (2)

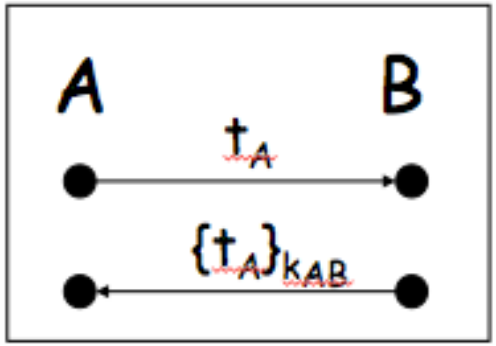
Può subire un *reflection attack* (non molto utile però…)

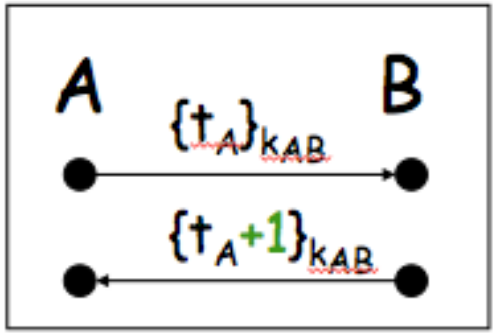


Da qui in poi *A* crede di aver autenticato *B* e comunicherà con lui cifrando i messaggi con la chiave K_{AB}, quindi *T* non sarà comunque in grado di venire a conoscenza dei messaggi in chiaro

Protocolli challenge-response - Timestamp

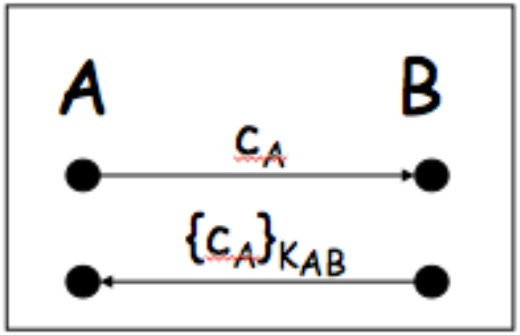
- Ora nel computer locale, ad esempio in millisecondi
- Controllabile dal destinatario
  - **Prevedibile**
  - Il destinatario deve memorizzare i timestamp **recenti** per evitare riutilizzi
- Richiede sincronizzazione





Protocolli challenge-response – Numeri in sequenza

- Il mittente mantiene un **contatore**
  - Viene incrementato di 1 dopo ogni challenge
  - Deve essere legato ai dati che identificano il canale e la coppia di partecipanti



- Il destinatario memorizza il valore più recente
  - Non accetta valori troppo vecchi

- Simile ai timestamp ma
  - Locale al mittente
  - Non richiede sincronizzazione

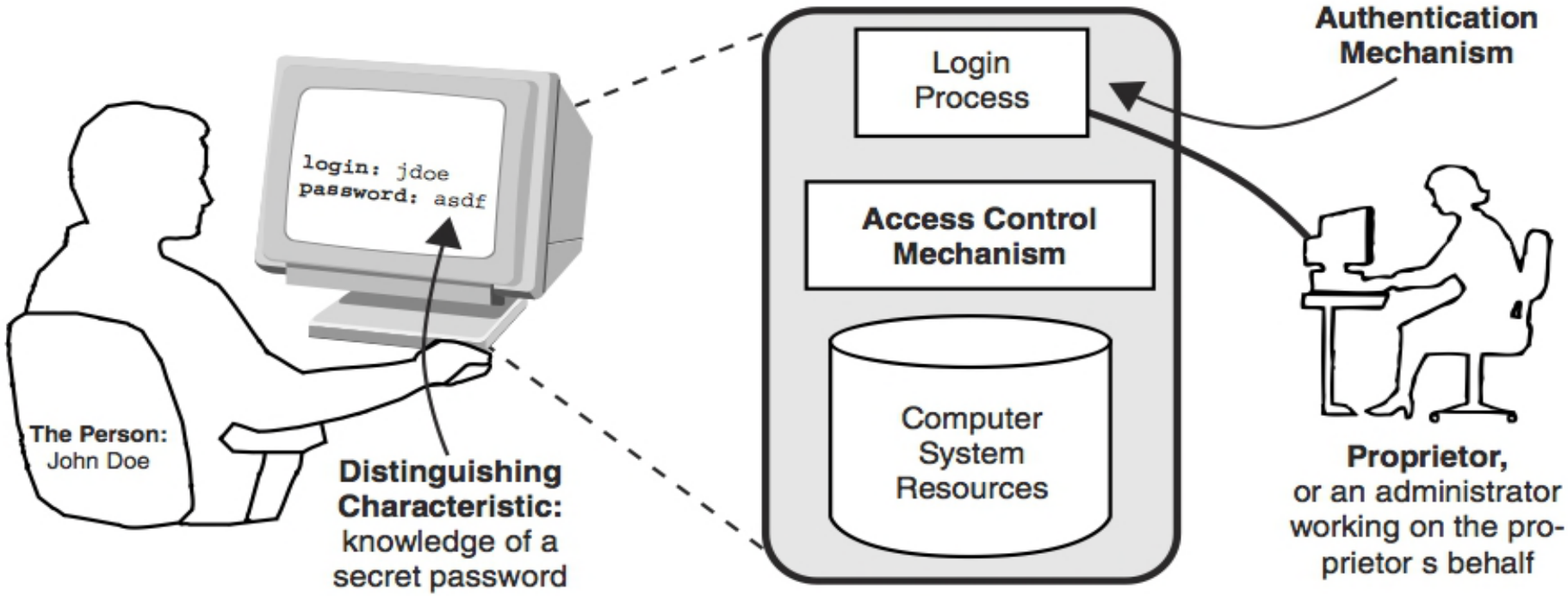
Tipologie di autenticazione

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
- **"Off-line":** i servizi possono prendere decisioni autonome anche senza dover contattare ogni volta l'autorità di autenticazione

Autenticazione utente-computer



L'utente lavora su di un terminale con una connessione **locale e diretta** ad un host

Autenticazione in sistemi distribuiti

Esempio: Utente-Applicazione Web?

Problema: il server **è** remoto

Cosa può fare un attaccante?

L'attaccante può:

1. intercettare il messaggio ascoltando il canale: minaccia alla *confidenzialità*
   - solo leggere il messaggio (*passivo*)
   - cancellare il messaggio senza inoltrarlo a Bob (*attivo*)
2. modificare i messaggi in transito: minaccia all'*integrità*
3. spedire messaggi arbitrari fingendo di essere Alice: minaccia all'*autenticazione*
4. riutilizzare vecchi messaggi salvati (*replay attack*): minaccia all'*autenticazione*

Quindi…

Si deve parlare di protocolli di autenticazione:

- per proteggere anche durante la trasmissione l'informazione

- "estendendo" la definizione di autenticazione che abbiamo dato

Autenticazione nella Rete – Definizioni

Entity Authentication (def. in standard ISO 1991)

- Garantisce l'identità dell'utente con cui in tempo reale si stanno scambiando messaggi all'interno di una sessione di protocollo

Message Authentication

- Garantisce che il messaggio ricevuto proviene dalla sorgente indicata, senza dare alcuna indicazione di quando il messaggio sia stato effettivamente creato

Protocolli challenge-response

Protocolli challenge-response (1)

Funzionamento di base (con chiave condivisa):

- **Challenge (sfida)**
  - il sistema o l'altro partecipante presenta all'utente una *stringa* contenente l'informazione che varia nel tempo
- **Response (risposta)**
  - L'utente cifra la stringa con la chiave condivisa: $\{stringa\}_K$ e la spedisce all'altro partecipante
- **Autenticazione**
  - Il partecipante controlla la $\{stringa\}_K$

NB: I dati segreti rimangono tali: non viene spedita in chiaro né una password né una chiave segreta

Protocolli challenge-response (2)

Sia il challenge che il response dovrebbero essere "fresh"

- Altrimenti il *replay attack* è possibile
- Attacchi con sessioni parallele sono sempre possibili …

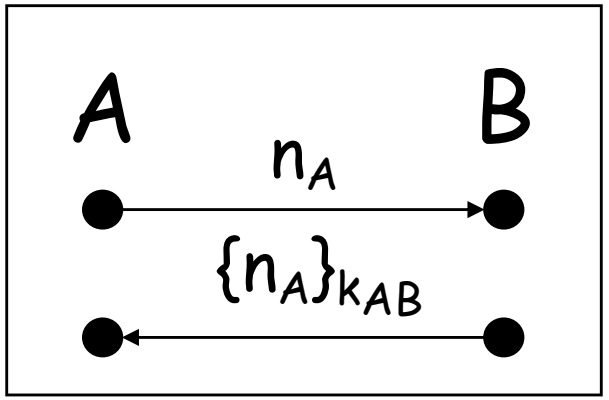
L'informazione che varia nel tempo (detta Nonce dal 1978) può essere:

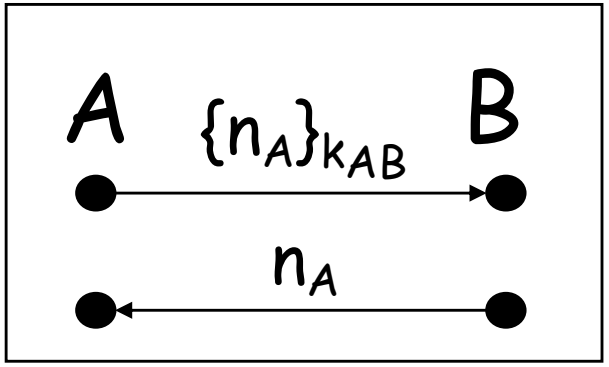
- Numero casuale (detto anche nonce)
- Timestamp
- Numeri in sequenza
- Chiavi a breve termine

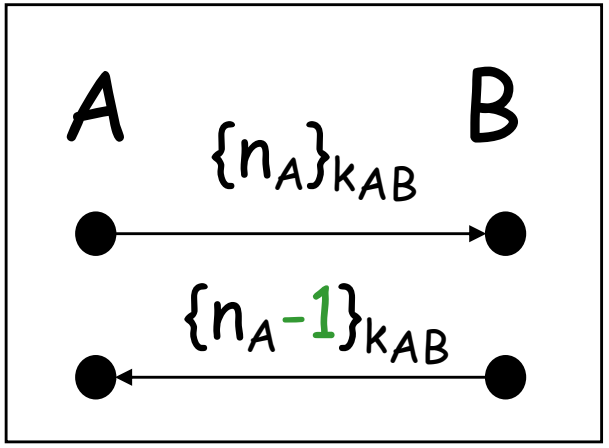
Protocolli challenge-response – Nonce (1)

- Sequenza casuale di bit (32-128)
- Generata ogni volta (fresh!) dal mittente come challenge
  - **Non prevedibile**
  - Verificata solo nel response
- **Non controllata dal destinatario**
  - Poco pratico memorizzarla
- Mai riutilizzata

Q: Possibile il reflection attack (come?)

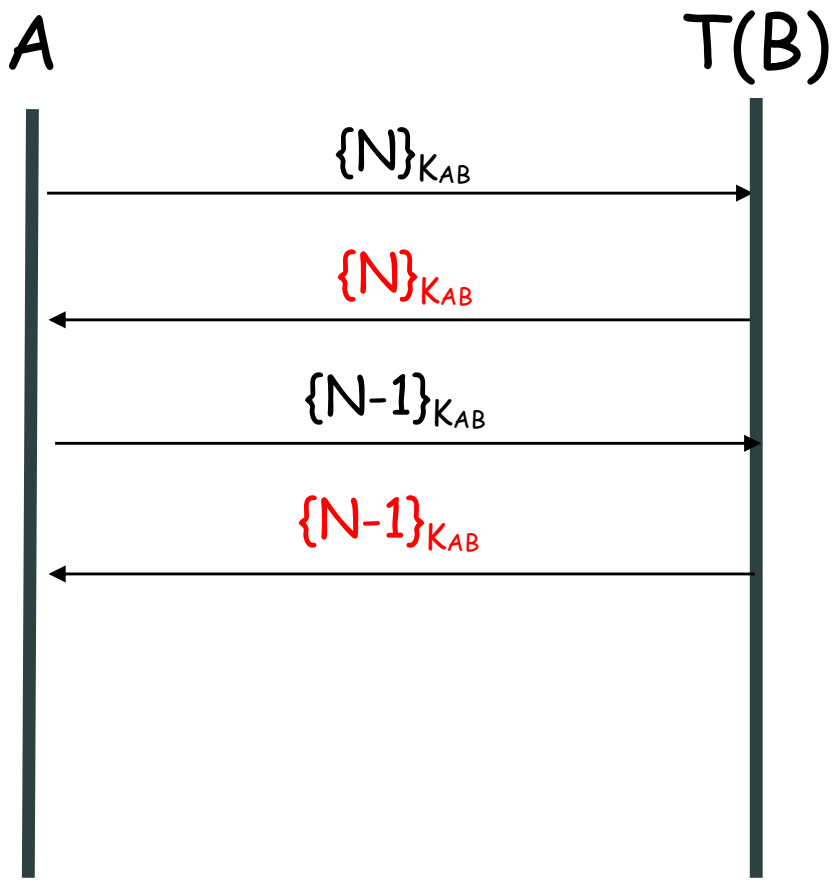






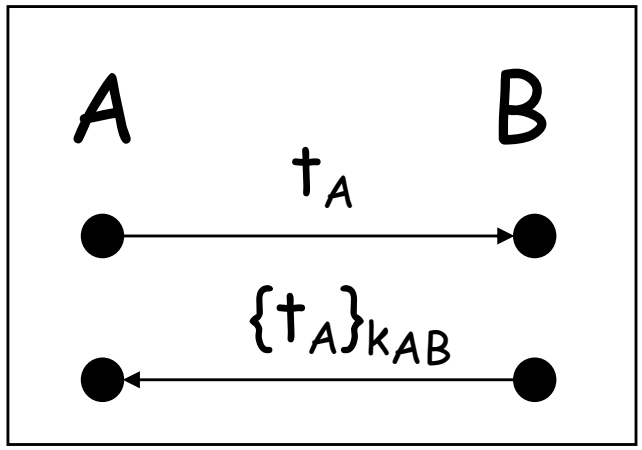
Protocolli challenge-response – Nonce (2)

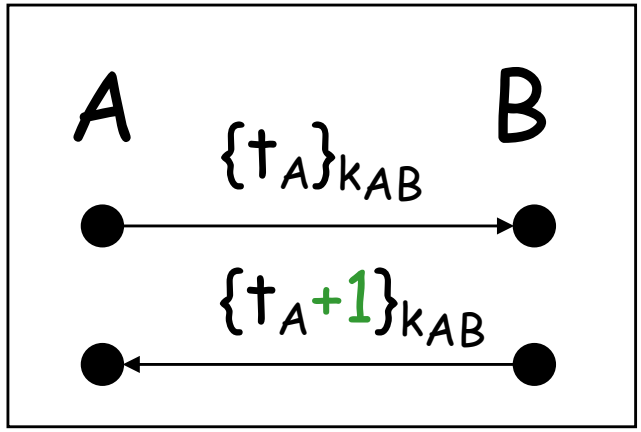
Reflection attack (non molto utile però…)



Protocolli challenge-response - Timestamp

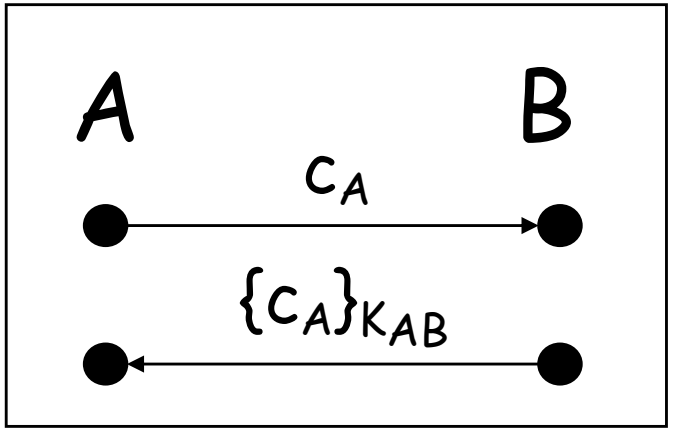
- Ora nel computer locale, ad esempio in millisecondi
- Controllabile dal destinatario
  - **Prevedibile**
  - Il destinatario deve memorizzare i timestamp **recenti** per evitare riutilizzi
- Richiede sincronizzazione





Protocolli challenge-response – Numeri in sequenza

- Il mittente mantiene un **contatore**
  - Viene incrementato di 1 dopo ogni challenge
  - Deve essere legato ai dati che identificano il canale e la coppia di partecipanti

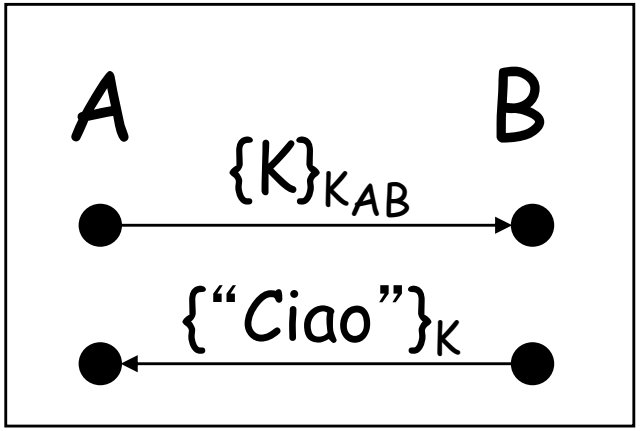


- Il destinatario memorizza il valore più recente
  - Non accetta valori troppo vecchi

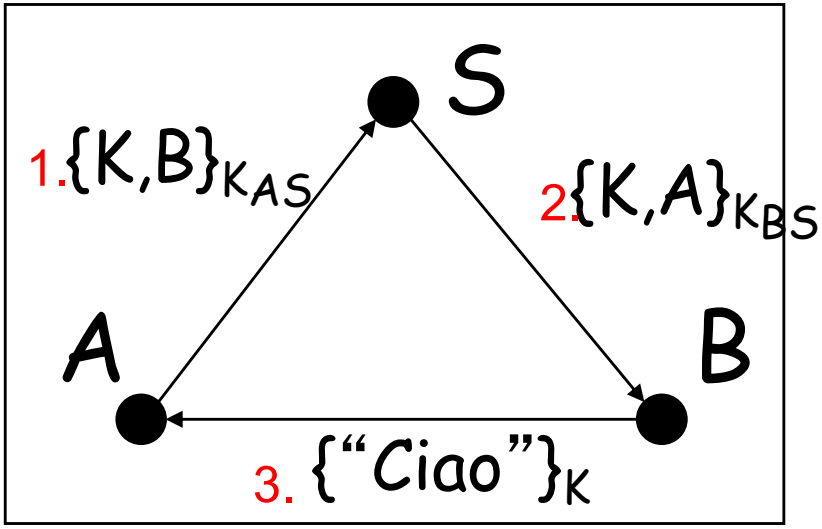
- Simile ai timestamp ma
  - Locale al mittente
  - Non richiede sincronizzazione

Protocolli challenge-response – Chiave a breve termine

- Il mittente genera una chiave *K*
  - La spedisce cifrata
- Il destinatario risponde usando *K*
- Si ottine la distribuzione di una chiave di sessione allo stesso tempo



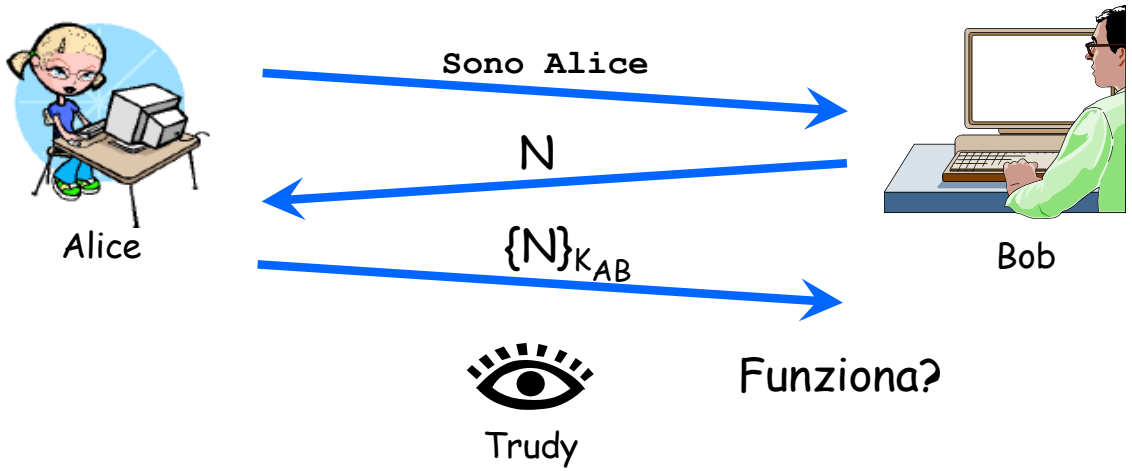
- Spesso ottenuto mediante un terzo partecipante



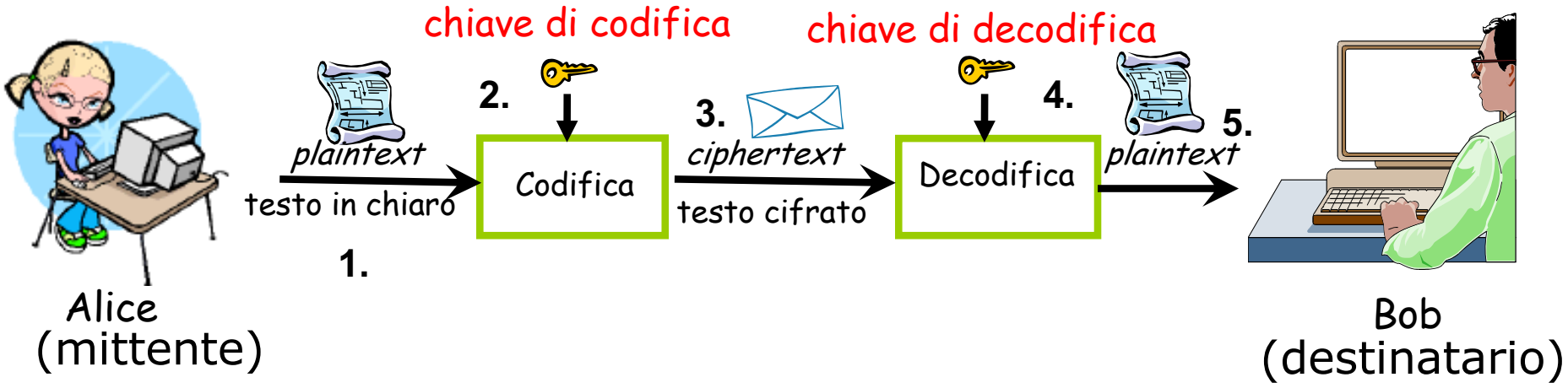
Protocollo di Autenticazione (5)

Obiettivo (aggiornato): Fare in modo che Alice "dimostri" la sua identità e che non ci siano replay attack.

Protocollo 5: Alice dice "Sono Alice", Bob spedisce ad Alice un nonce *N* (un numero casuale usato solo una volta). Alice deve rispedire *N cifrato* con una chiave segreta condivisa K_{AB}.



Crittografia - Schema generale (2)



Crittografia simmetrica:

- Stessa chiave per codifica e decodifica (da tenere segreta!)

Crittografia asimmetrica:

- Una coppia di chiavi per ciascun utente: *chiave pubblica* per la codifica, *chiave privata* per la decodifica

Protocollo di Autenticazione (6)

Problema: Il protocollo 5 richiede una chiave condivisa

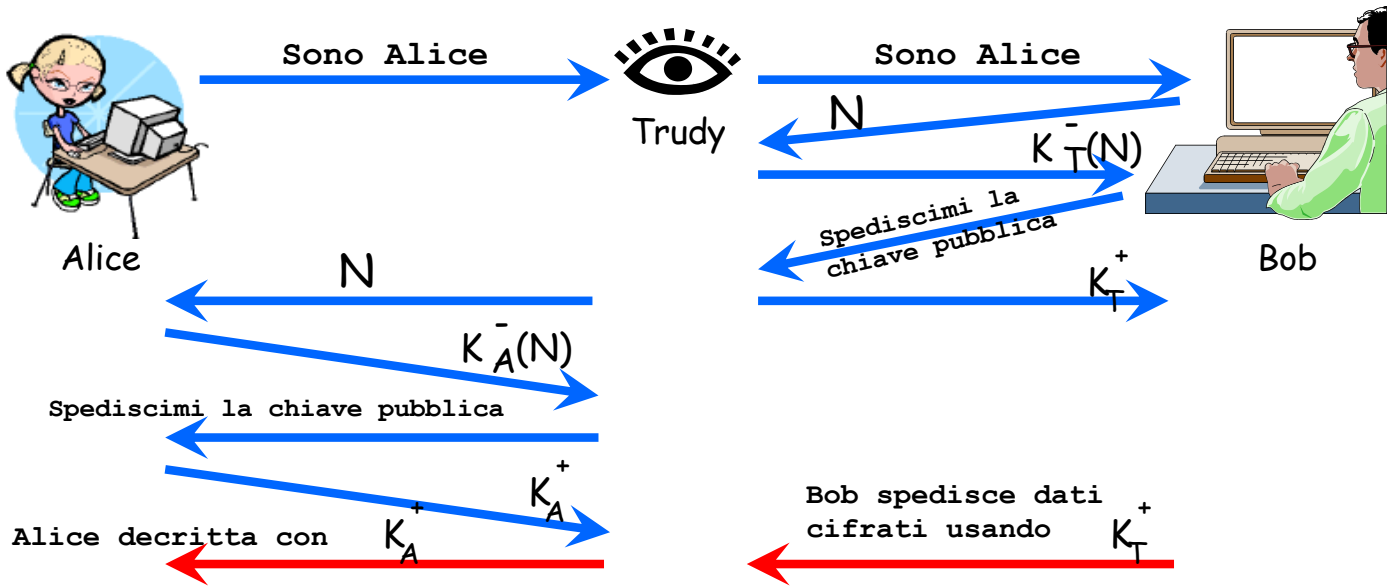
- Come possono Bob e Alice accordarsi sulla chiave?
- Possiamo utilizzare tecniche a chiave pubblica?

Protocollo 6: usare nonce e crittografia a chiave pubblica.

Bob calcola $K_A^+(K_A^-(N)) = N$ e sa che solo Alice può avere la chiave privata che cifra N in modo che $K_A^+(K_A^-(N)) = N$

Protocollo 6: un attacco!

Man (woman) in the middle attack: Trudy finge di essere Alice (con Bob) e di essere Bob (con Alice).



Trudy riesce a leggere tutti i messaggi!!

Soluzione: chiavi pubbliche "certificate"!!

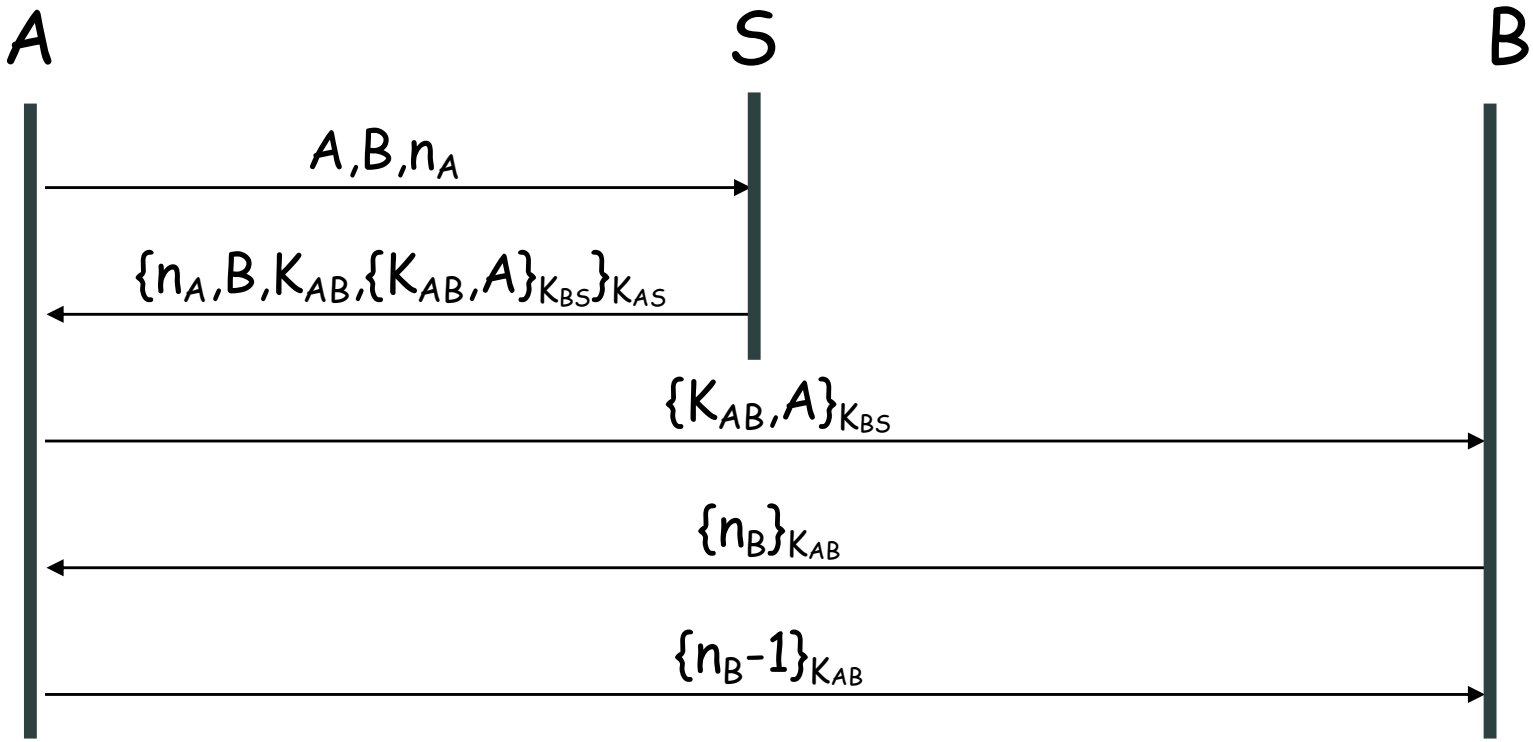
Needham-Schroeder a chiave condivisa (1)

- Progettato nel 1978 da R. Needham e M. Schroeder
- **Protocollo a chiave segreta** per stabilire una chiave di sessione utilizzabile da due entità di rete per proteggere le comunicazioni successive

Caratteristiche:

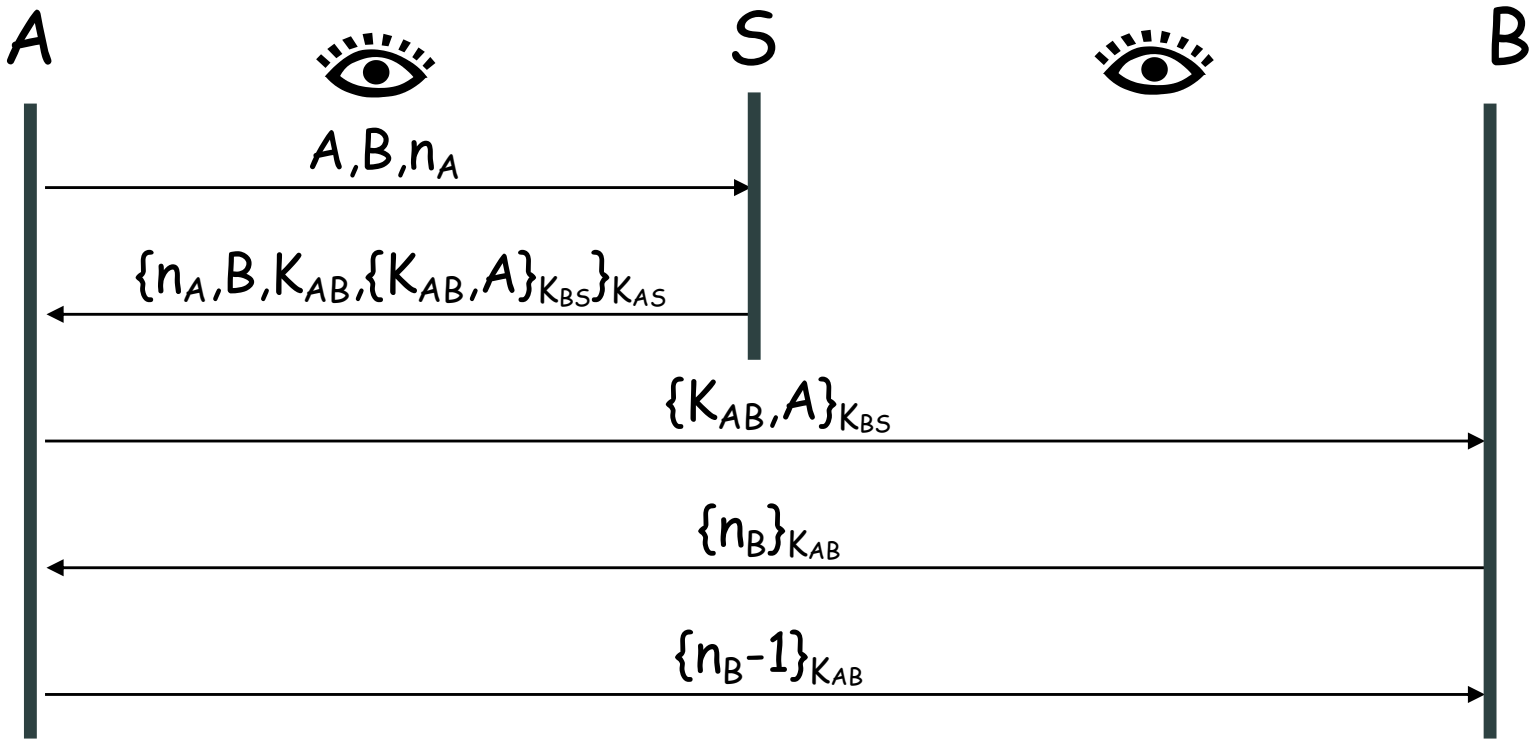
- Utilizzo di *crittografia simmetrica*
- Utilizzo di un *KDC* per generare la chiave di sessione
- Mutua autenticazione basata su *nonce*
- Chiavi condivise:
  - Una a lungo termine condivisa con il server
  - Una di sessione generata dal server

Needham-Schroeder a chiave condivisa (2)



- *S* (il KDC, server fidato) crea la chiave (di sessione) condivisa K_{AB}

Needham-Schroeder a chiave condivisa (2)



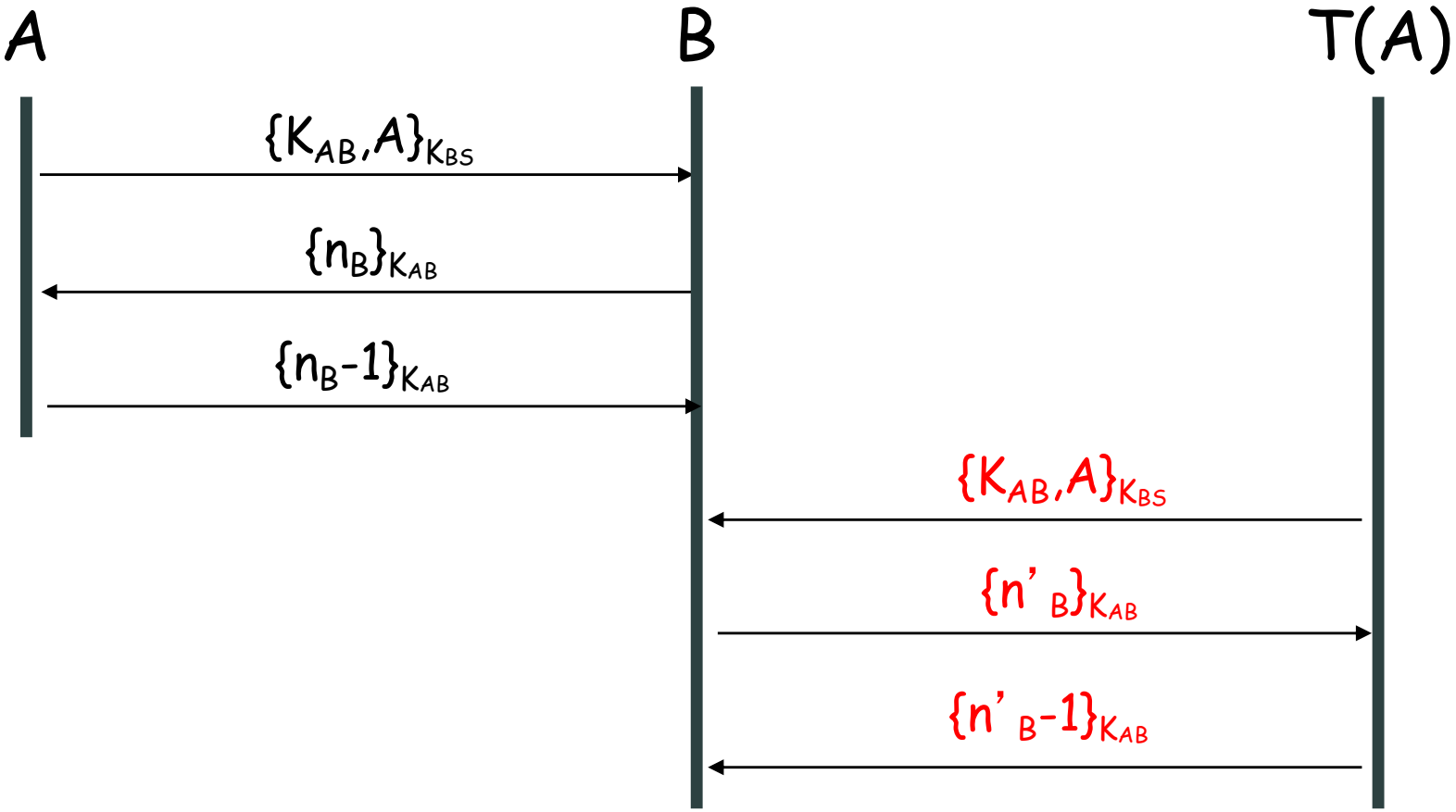
NB: L'attaccante può leggere tutti i messaggi trasmessi lungo il canale

Possibile attacco (1)

- Trovato da Denning e Sacco (1981!)

- Un replay attack in cui Trudy finge di essere A con B usando una **vecchia** chiave compromessa K_{AB}:
  1. *Trudy* rispedisce a *B* il messaggio $\{K_{AB}, A\}_{K_{BS}}$
  2. *B* lo accetta visto che non riesce a capire che si tratta di una vecchia chiave

Possibile attacco (2)



- NB: Omessi i primi due messaggi con il server

Possibile attacco (3) - Soluzione

- Utilizzare un timestamp o un nonce nel messaggio $\{K_{AB},A\}_{K_{BS}}$ per garantire la freshness della chiave:

  1. A → B: $\{K_{AB},A, t_{exp}\}_{K_{BS}}$
  2. B → A: $\{n_B\}_{K_{AB}}$
  3. A → B: $\{n_B-1\}_{K_{AB}}$

- Il problema c'era anche in Kerberos, risolto usando un timestamp