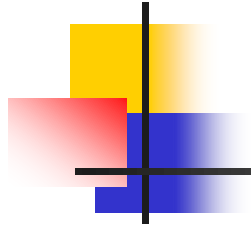




Sicurezza Informatica

- Prof. Stefano Bistarelli
- bista@dipmat.unipg.it
- <http://www.sci.unich.it/~bista/>





Autenticazione

- Stabilisce l'identità di una “parte” ad un'altra
- Le parti 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 richiede varie combinazioni
- Proprietà primaria
 - Richiesta da un corretto controllo d'accesso



Tipi di autenticazione

- Locale
 - Desktop systems
- Diretta
 - Sul server (file, login, ..)
- Indiretta
 - Windows domain, radius, kerberos, nis
- Off-line
 - PKI ...

Autenticazione indiretta



■ Protocollo di autenticazione

1. $U \rightarrow S$: username
2. $S \rightarrow U$: challenge
3. $U \rightarrow S$: response



– Protocollo di autenticazione indiretto

- $U \rightarrow S_R$: logon request (chi sono e cosa voglio)
- $S_R \rightarrow S_A$: authentication request
- $S_A \rightarrow S_R$: authentication response (A/R)
- $S_R \rightarrow U$: logon response



Computer-utente – esempio

Bob intende autenticare il server web della sua banca

1. Bob invia una richiesta al server
2. Il server replica con qualcosa del tipo
 $\{\text{"salve Bob, sono il server web della tua banca"}\}_{K^{-1}\text{bancaxy}}$
3. Bob scarica il certificato per la chiave pubblica della banca
 $\{\text{"banca XY", } K_{\text{bancaxy}}\}_{K^{-1}\text{CA}}$
lo verifica ed estrae K_{bancaxy}
4. Bob usa la chiave ottenuta al passo 3 per verificare il certificato ottenuto al passo 2
5. Se Bob ottiene qualcosa di intelligibile, allora autentica il server, altrimenti no

Autenticazione utente-computer

Basata su qualcosa che l'utente

1. **Conosce**: segreti

- *Password, PIN, ...*



2. **Possiede**: cose fisiche o elettroniche

- *Chiavi convenzionali, carte magnetiche o smart*



3. **E'**: caratteristiche biometriche

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





What I know

Authenticati

- Come un sistema può associare *con certezza* una identità ad una persona
- Password: uso antico!!



Apriti Sesamo!

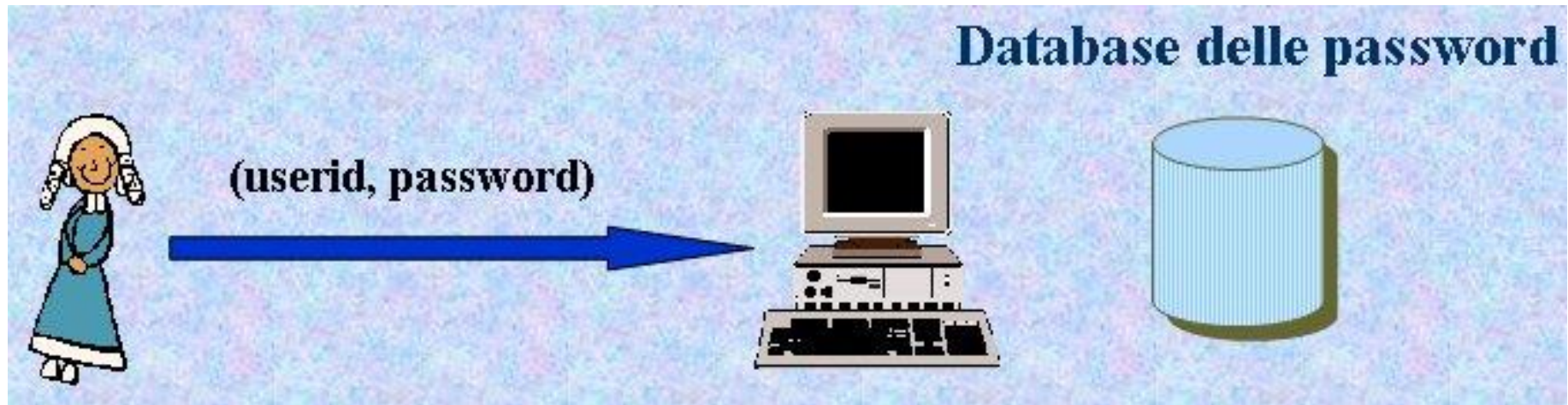


1. Autenticazione su conoscenza

- Conoscenza fornisce prova dell'identità
- Coppia userid-password
- Antico e diffuso
- Semplice da implementare
- Economico
- Debole

Gestione delle password

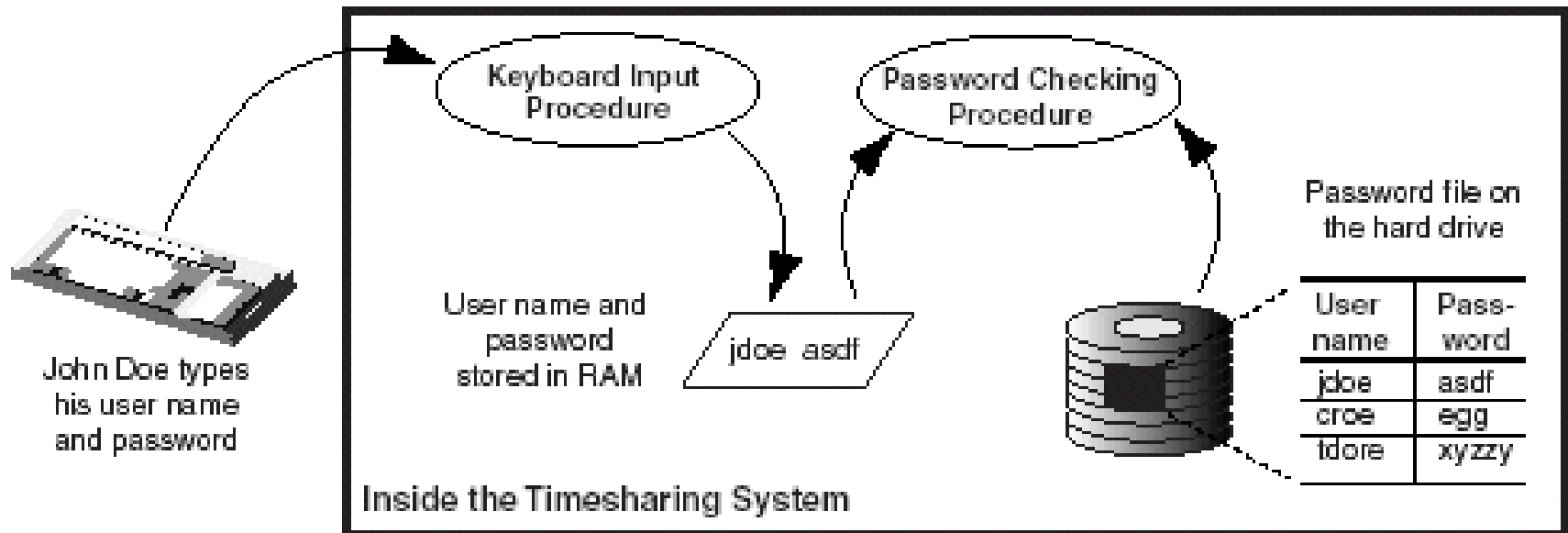
- Il sistema deve memorizzare una rappresentazione delle password. Come??



Compatible Time Sharing System (CTSS)

■ 1960, MIT

- Password memorizzate in chiaro su file di sistema protetto da politica di sicurezza



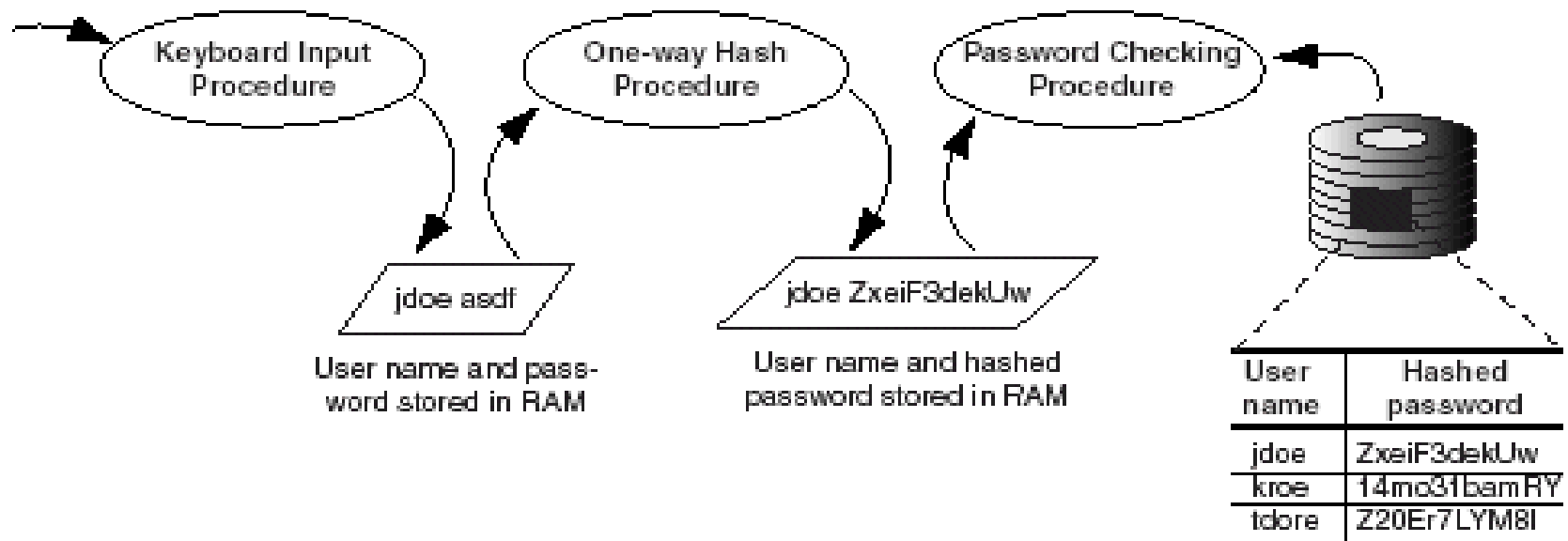


CTSS

- Limiti intrinseci
 - Il controllo d'accesso si basa sull'autenticazione (sempre)
 - L'autenticazione si basa sul controllo d'accesso (CTSS)
- La storia registra numerose violazioni di questo schema
 - Memorizzate in chiaro in un file protetto!!
 - Problema! Nessuna protezione contro chi si impossessa del file!! ☹️

CTSS + hashing

- 1967, Cambridge University
- Il file delle password memorizza l'hash delle password

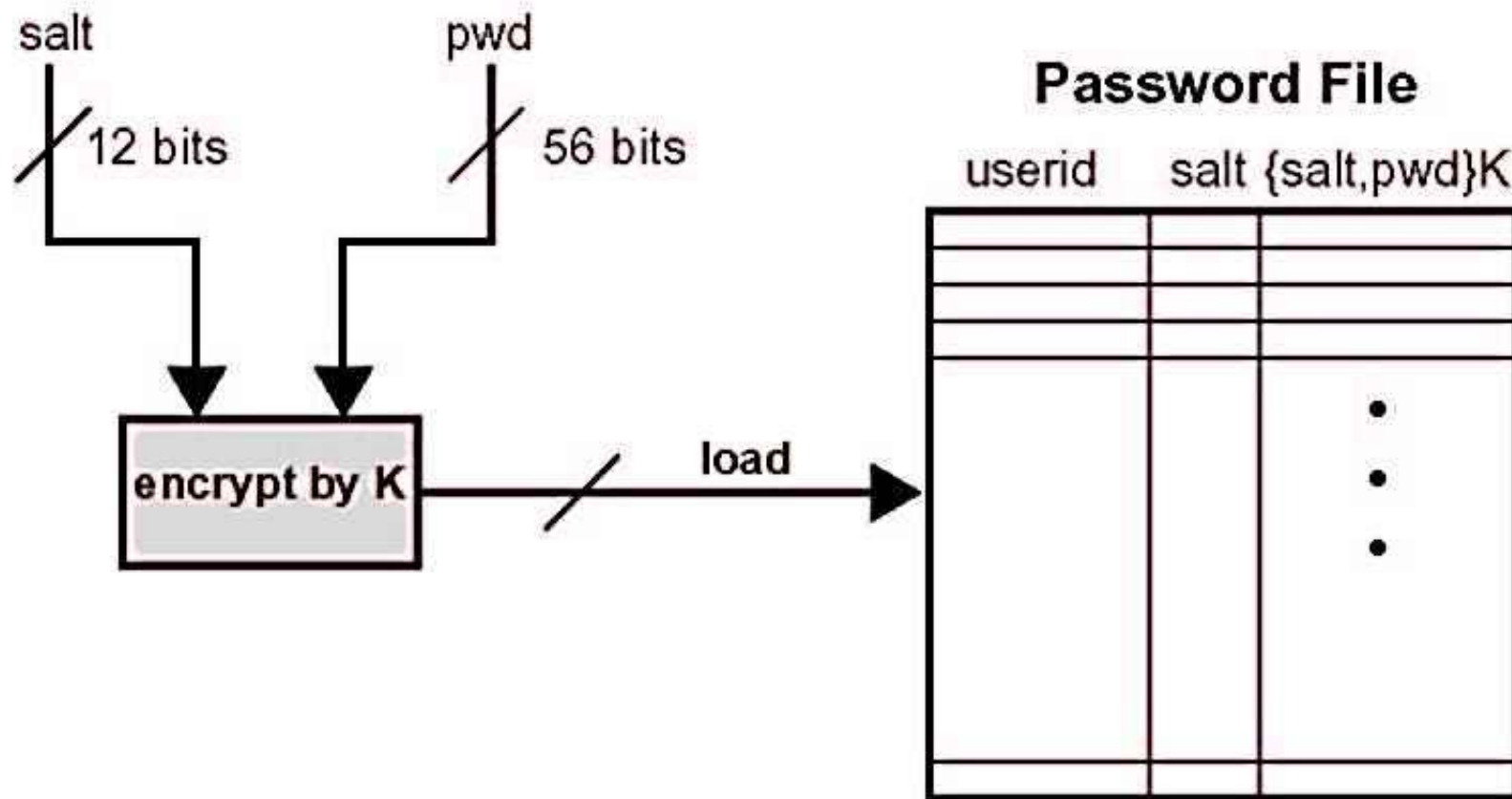




Password in Unix (cenni)

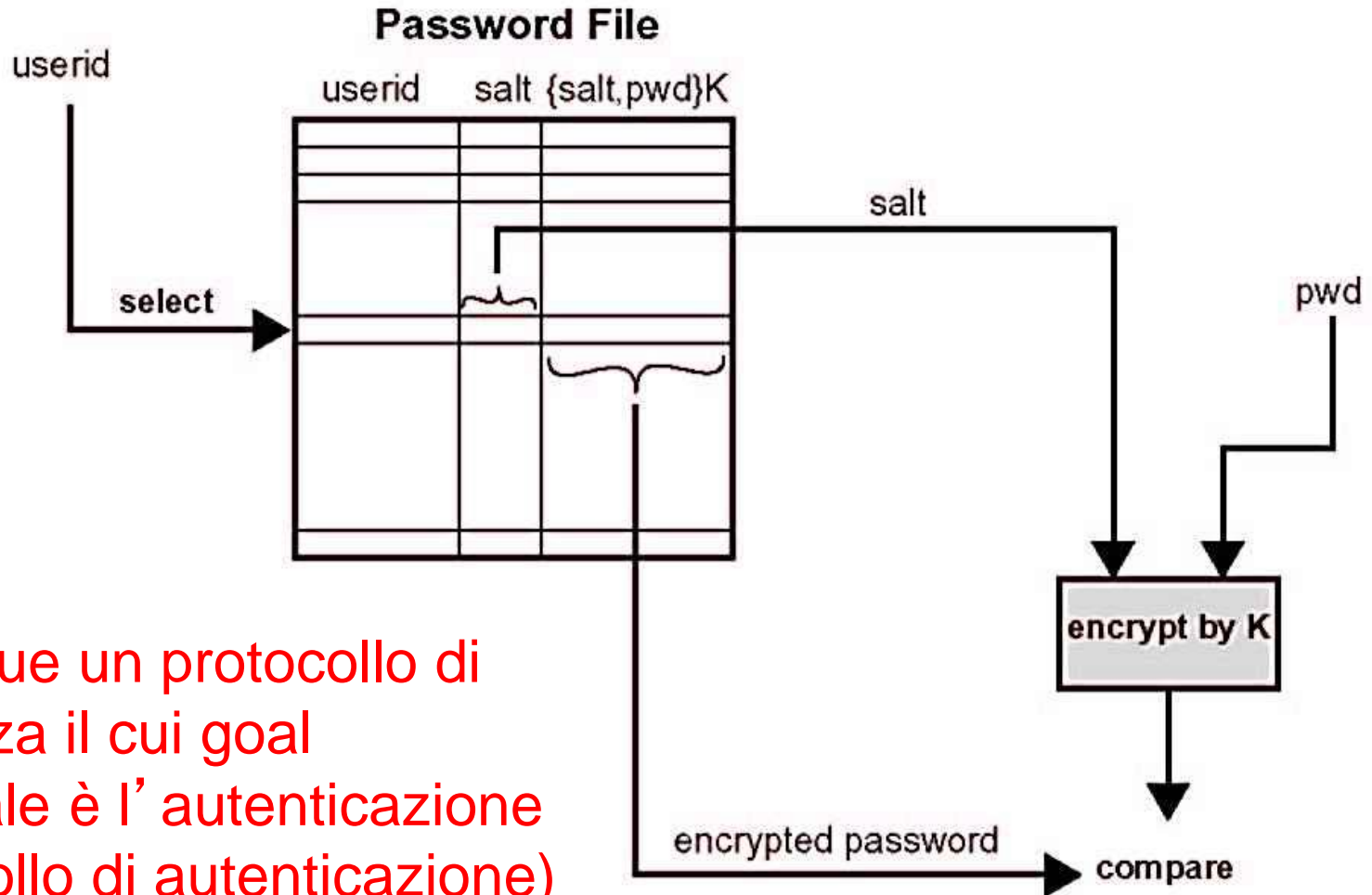
- Memorizzate codificate insieme a **salt**
- Salt: sequenza di bit generata dal sistema
- Siffatto file delle password memorizzato in directory “shadow”, in nessun modo accessibile da alcun utente eccetto “root”

Unix: aggiunta di nuova password



Si esegue un protocollo di sicurezza

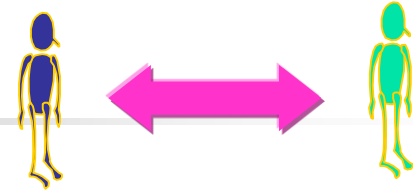
Unix: verifica di una password



Si esegue un protocollo di sicurezza il cui goal principale è l'autenticazione (protocollo di autenticazione)



Pericoloso??



- Inserimento utenti
 - No password
 - Default password
 - Inserita dall'utente al momento della consegna
 - Consegnata dall'amministratore
 - Obbligo di modifica
 - Entro un certo tempo max
 - Controllo della password scelta
 - Lunghezza, caratteri,



Vulnerabilità delle password ...

- Le password rischiano

1. **Guessing**: indovinate (dizionario)
2. **Snooping**: sbirciate mentre vengono inserite (Shoulder surfing)
3. **Sniffing**: intercettate durante trasmissione in rete (Keystroke sniffing)
4. **Spoofing**: acquisite da terze parti che impersonano l'interfaccia di login (Trojan login)
5. Van Eck sniffing

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



... E difese

- Guessing attack
 - Audit-log
 - Limite agli sbagli
- Social engineering
 - Cambio password abilitato solo in specifiche circostanze
 - Policy!!!
- Sniffing attack
 - Shoulder surfing
 - Password blinding
 - Keystroke sniffing
 - Memory protection
- Trojan login
 - Special key to login
- Offline dictionary attack
 - Shadow password (unix)

Scelta della password

Delicata a causa dei rischi di guessing



Copyright © 2001 United Feature Syndicate, Inc.



Norme fondamentali

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



Controlli automatici su password

- 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



Alternativa ai controlli

- La password sia generata da un apposito sistema in maniera pseudorandom
 - Non sempre ben accetto (difficoltà nel ricordare)
- Ricorrere a **one-time password** (monouso)
 - Distribuzione improponibile per un uso continuativo



Tecniche di violazione

- **Tentativi standard**: indipendenti dall'utente
 - *Password tipiche, parole brevi (1-3 caratteri), parole in un dizionario elettronico (decine di migliaia)*
- **Tentativi non-standard**: informazioni sull'utente
 - *Hobby, nomi parenti, compleanno, indirizzi, numeri di polizze, di targhe, di telefono,...*



Distribuzione iniziale di password

L'utente si reca dall'amministratore e si autentica tradizionalmente

L'amministratore prepara l'account e l'utente digita la password

- Rischio potenziale per il sistema!

L'amministratore prepara l'account e sceglie la password iniziale

- L'utente la cambierà al primo utilizzo



2. Autenticazione su possesso

■ Possesso di un **token** fornisce prova dell'identità

- *Carte magnetiche*
- *Smart card*
- *Smart token*
- Ogni token memorizza una chiave (pwd)
- Recente e poco diffuso
- Non proprio economico
- Più robusto delle password



Pro e contro del sistema

- L'autenticazione dimostra solo l'identità del token, non quella dell'utente
 - Token persi, rubati, falsificati
 - Rubando un token si impersona l'utente
- + Idea: combinare possesso e conoscenza
 - *Bancomat: carta + PIN*
- + Vantaggio: molto difficile estrarre un segreto da un token

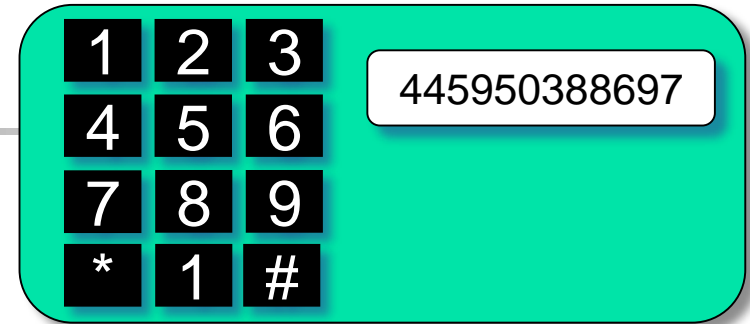


Tipi di token

Carte magnetiche (obsolete)

- Smart card per memorizzare pwd robusta
 - **Memory card**: ha una memoria ma non capacità computazionali
 - Impossibile controllare o codificare il PIN
 - PIN trasmesso in chiaro soggetto a sniffing
 - **Microprocessor card**: ha memoria e microprocessore
 - Possibile controllo o codifica PIN
- **Smart token**

Smart token



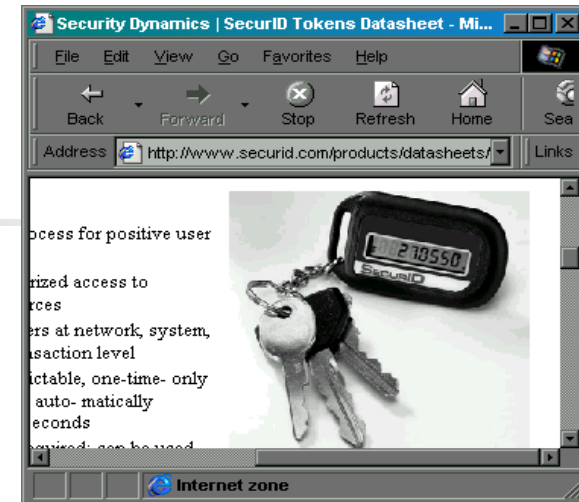
- Protetto da PIN
- Microprocessor card + tastierina e display
- Vero e proprio computer!
- Svantaggi: costoso e fragile



Smart token – funzionamento

- Chiave segreta (**seme**) memorizzata dalla fabbrica, condivisa col server
- Prende info esterne (*PIN, ora,...*) per generare one-time password
- Password sul display, rinnovata ogni 30-90 secondi
- Sincronizzazione col server grazie a seme ed algoritmo comune

Smart token commerciali



3 Autenticazione su caratteristiche

■ 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 moderna e promettente
- **Template**: rappresentazione digitale delle caratteristiche univoche del dato biometrico



Funzionamento

Fase iniziale di campionamento

- Esecuzione di più misurazioni sulla caratteristica d'interesse
- Definizione di un template
- Autenticazione: confronto fra la caratteristica appena misurata rispetto al template
- Successo se i due corrispondono a meno di una tolleranza, che va definita attentamente
- Perfetta uguaglianza tecnicamente imposs.



Problema

- Confrontare la caratteristica appena misurata dall'utente col template di quell'utente
- Distinguendola dal template di un altro utente!

Today's
Biometric
Signature
from Cathy:

389
416
501
468
353

Cathy's
Stored
Biometric
Pattern:

390
418
502
471
355

Distance = 4
from that
signature

Tim's
Stored
Biometric
Pattern:

284
570
534
501
399

Distance = 199
from that
signature



Discussione

- Forma di autenticazione più forte anche se tecnicamente meno accurata
 - Eliminate in pratica le impersonificazioni
- 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

Esempio: le 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



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

[illegible]

... ai lettori ottici

■ Ora basta poggiare un attimo il dito sul lettore



Classificazioni di impronte

- Classificate in tre grandi gruppi in base allo “schema” predominante



loop



arch



whorl

Schema Loop

- Schemi circolari che escono verso l'esterno
- Caratterizza circa il 60% della popolazione



Schema Arch

- Cerchi che escono da entrambi i lati
- Raro: caratterizza solo il 5% della popolazione



Schema Whorl

- Cerchi concentrici, nessuna linea esce dall'immagine
- Caratterizza circa il 35% della popolazione

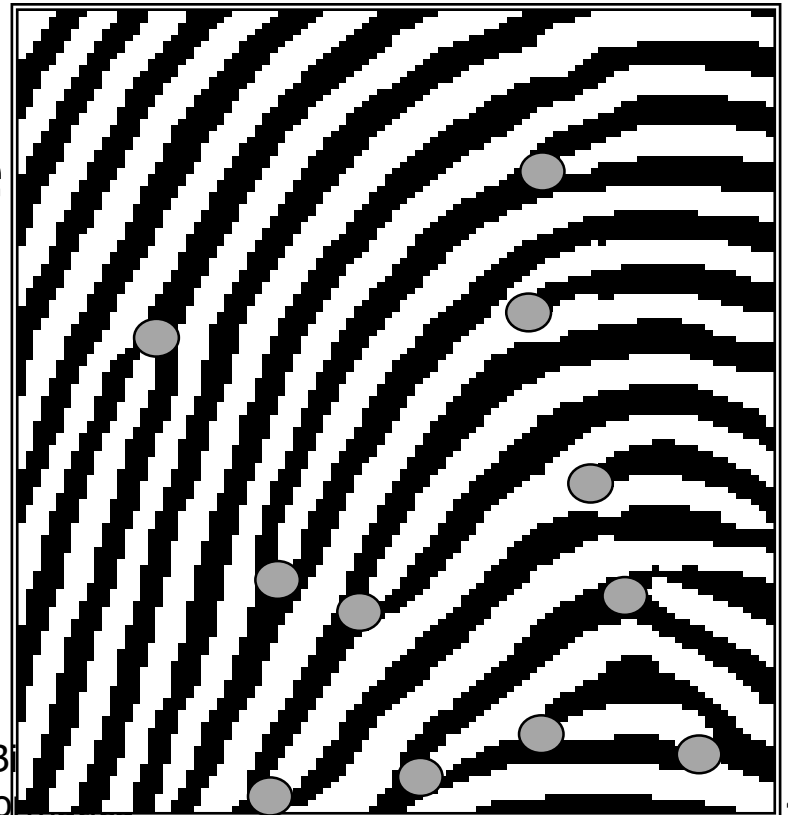


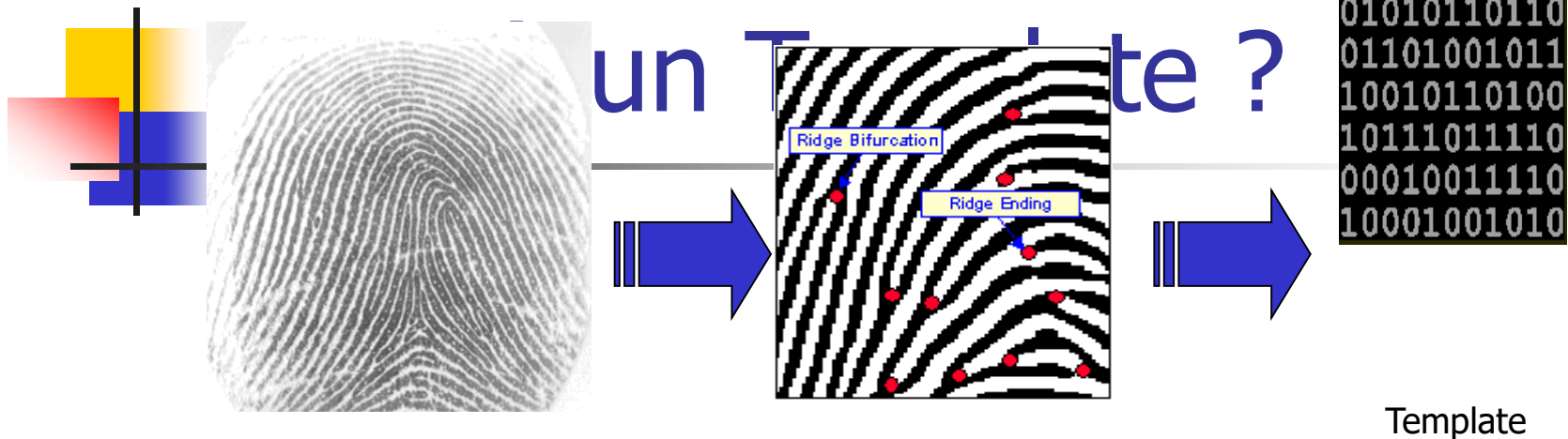
Riconoscimento di impronte

- Necessita di algoritmi avanzati per il riconoscimento delle immagini digitali

- **Minuzie**

- rappresentano la fine e il punto di biforcazione delle linee
- uniche per individuo
- standard nei sistemi di riconoscimento





- Template

- Contiene le caratteristiche univoche per ogni dato biometrico
- Es: contiene le coordinate geometriche delle minuzie

- Minuzie

- Rappresentano la fine e il punto di biforcazione delle linee
- Sono uniche da individuo ed individuo ed oggi uno standard nei sistemi di riconoscimento basati su fingerprints

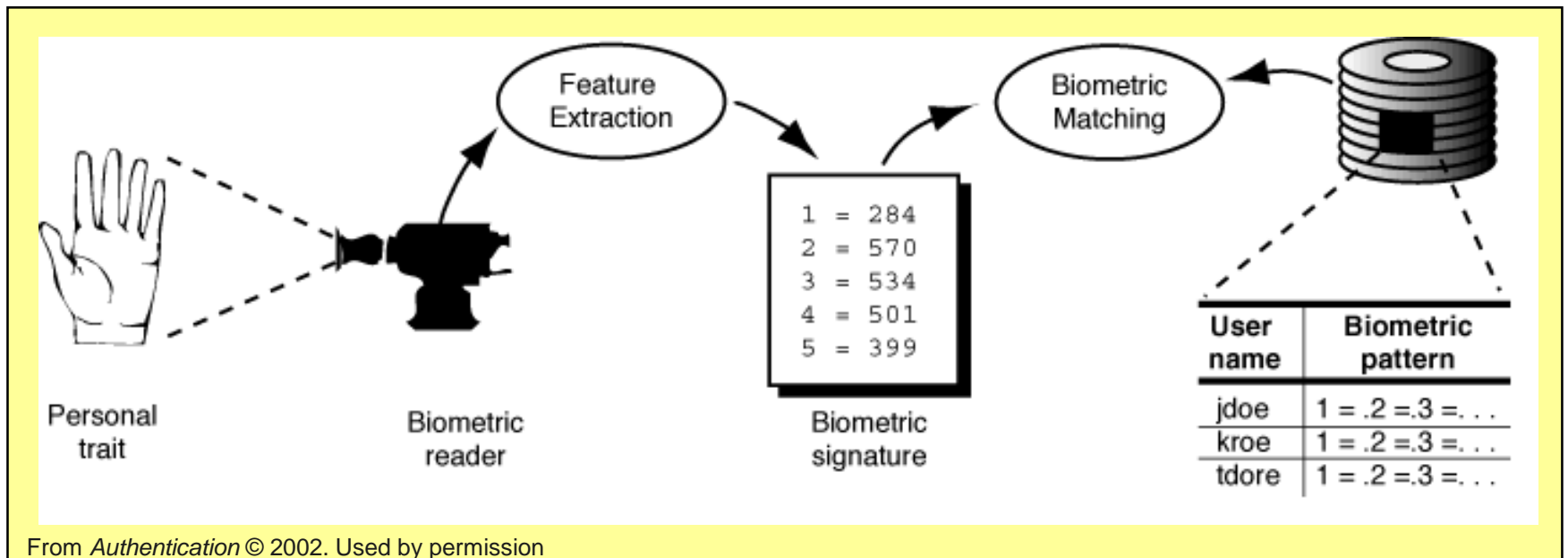


Sommario

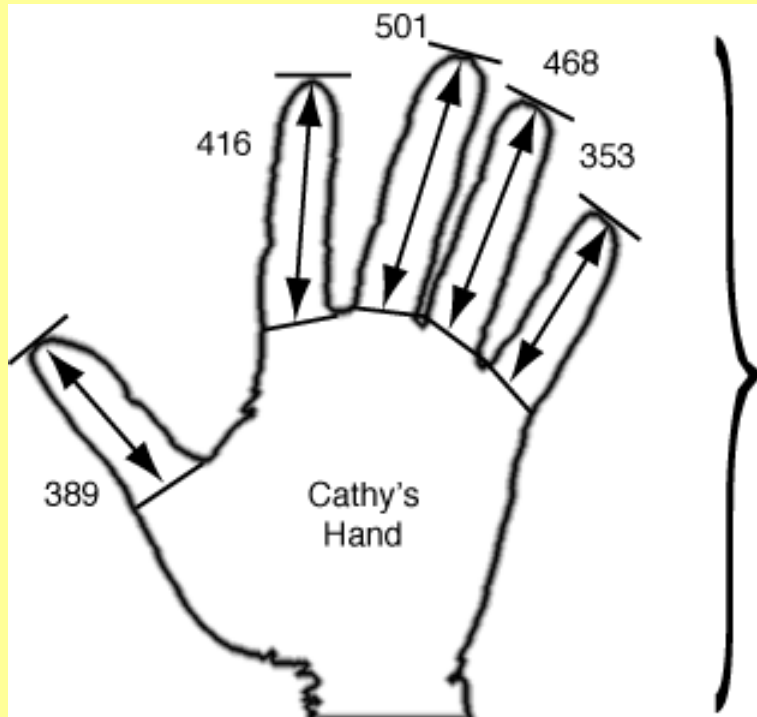
- Le impronte digitali
- Identificazione e verifica d'identità

confrontare il template acquisito con il template memorizzato

- “Biometric template”: Matching è sempre approssimativo! Mai esatto!



From *Authentication* © 2002. Used by permission



Today's
Biometric
Signature
from Cathy:

389
416
501
468
353

Cathy's
Stored
Biometric
Pattern:

390
418
502
471
355

Distance = 4
from that
signature

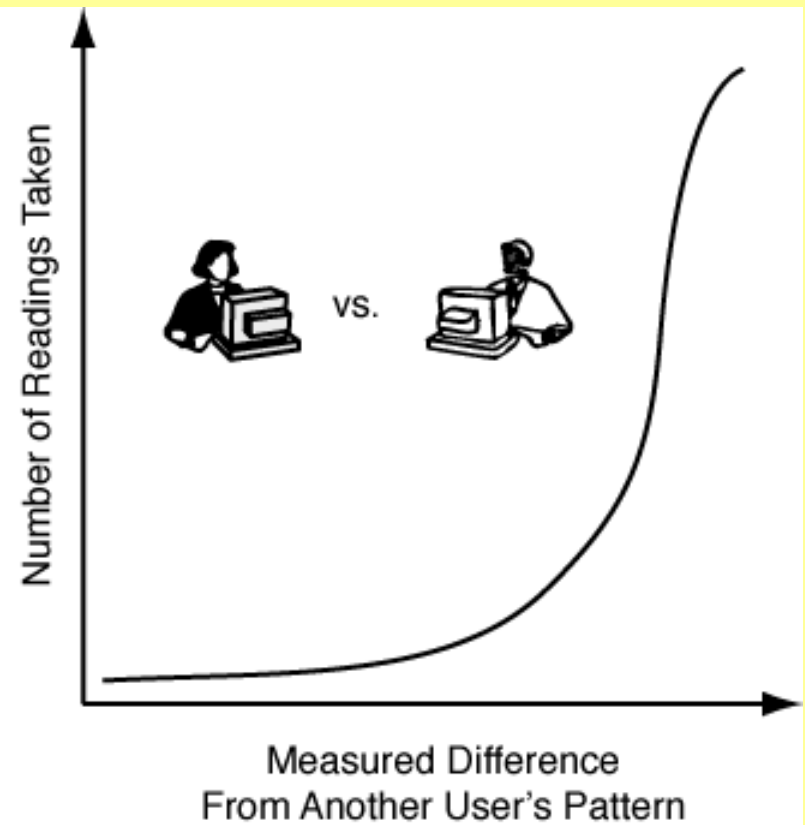
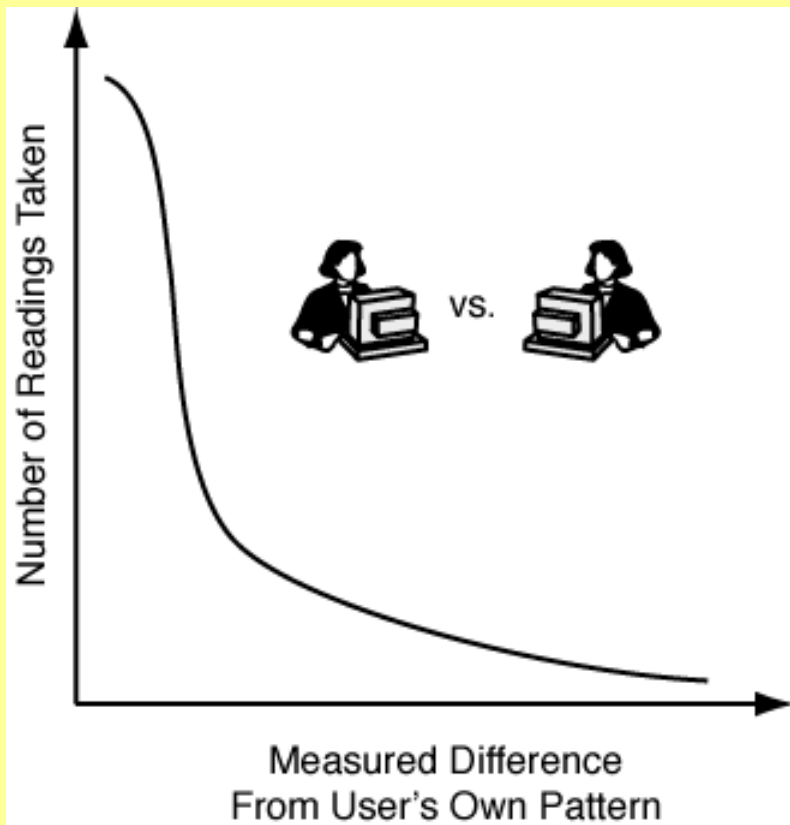
Tim's
Stored
Biometric
Pattern:

284
570
534
501
399

Distance = 199
from that
signature

From *Authentication* © 2002. Used by permission

one user's pattern versus another's pattern



From *Authentication* © 2002. Used by permission



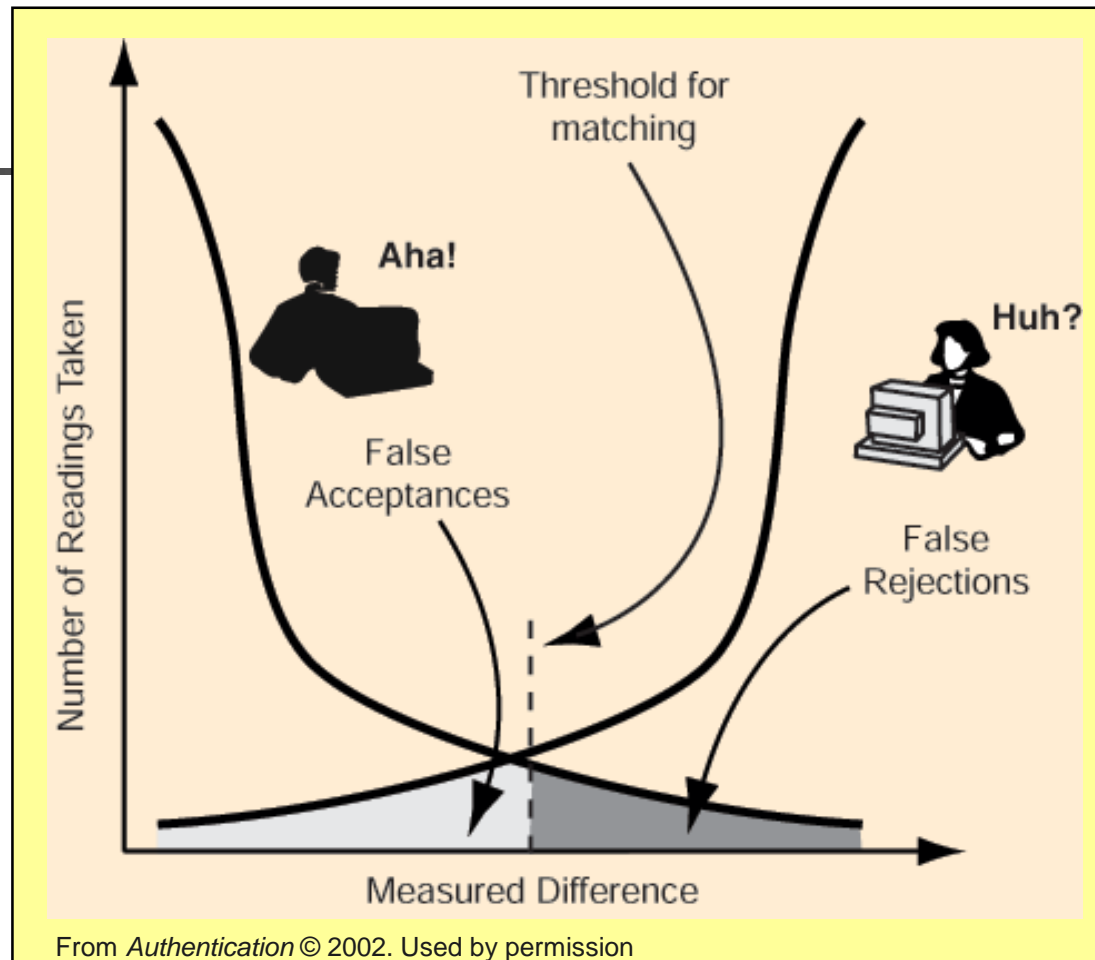
Measurement Trade-Offs

Dobbiamo bilanciare FAR e FRR

- basso FAR = sopporta di più gli attacchi
 - Meno tollerante nei match corretti
 - quindi – aumenta il FRR
- basso FRR = facile da usare
 - Più tollerante ai match corretti
 - Tuttavia, sopporta di meno gli attacchi e potrebbe matchare incorrettamente un attaccante
 - quindi – aumenta il FAR

Equal error rate = punto dove FAR = FRR

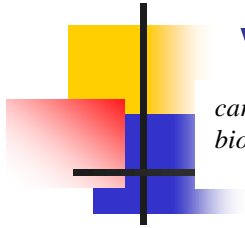
Instead, I KK – doesn't recognize me





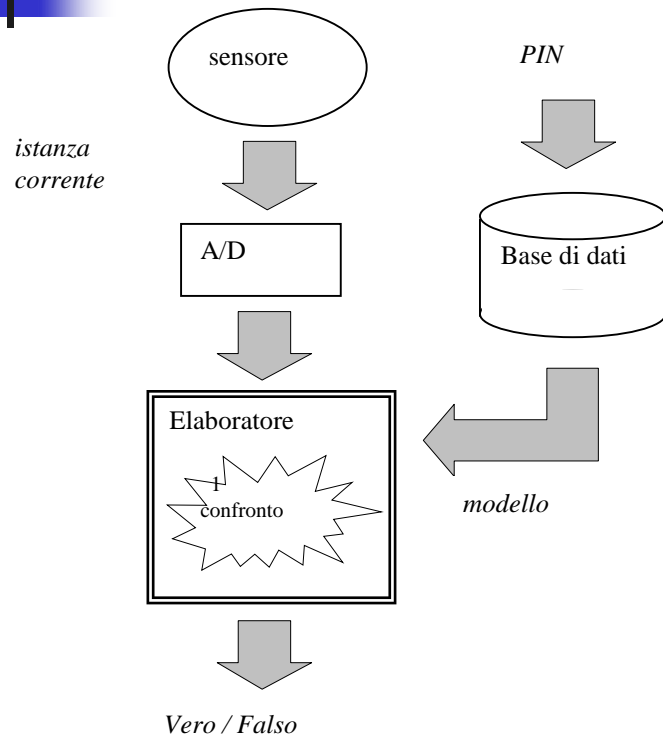
Sommario

- Le impronte digitali
- Identificazione e verifica d'identità



Verifica Identità

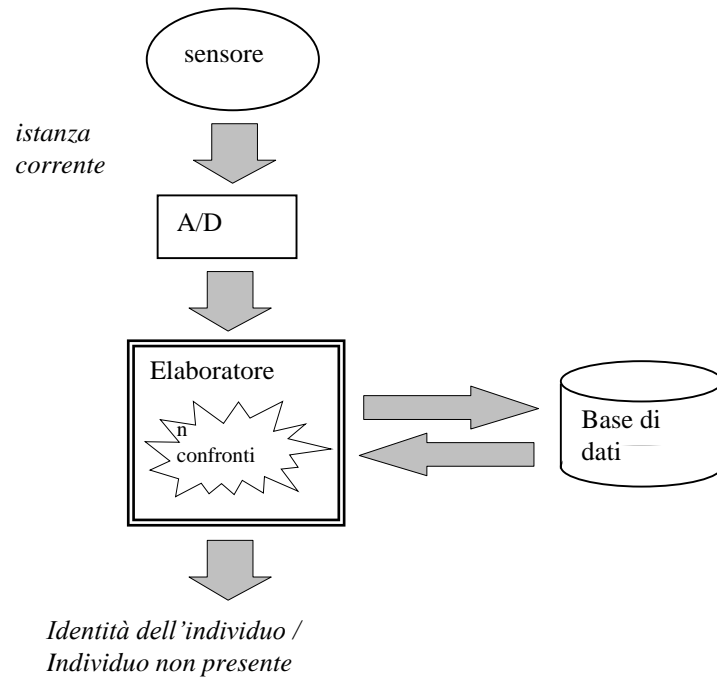
*caratteristica
biometrica*



Riconoscimento/Identificazione

e

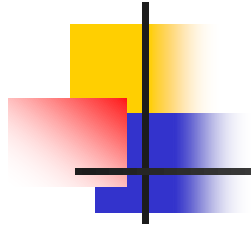
caratteristica
biometrica





Quale tecnica di autenticazione?

- 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





Aggiunta di autenticazione!

- Una password per accedere al sistema
- Una per accedere al file system
- Una per la rete
- La stampa
- La posta
- ...

Scomodo! Ancor peggio con token!



Accesso singolo

Def. Usare unica credenziale di autenticazione per accedere a tutti i servizi

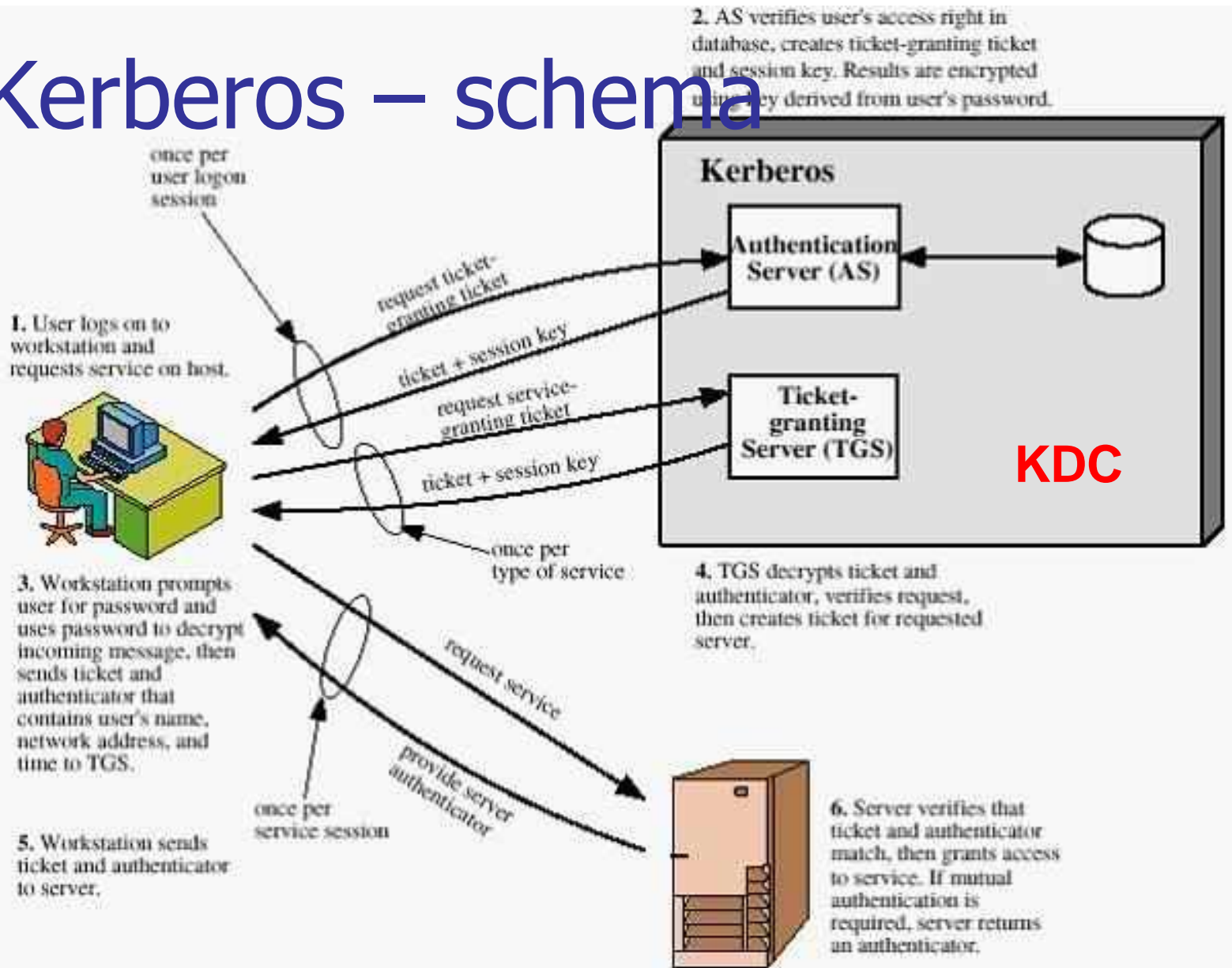
- Soluzione comoda ma poco robusta
 - *Un' unica password per tutto*
- **Kerberos** è un protocollo reale che si occupa di questo problema (e non solo)

Kerberos (fine anni '80)

- In mitologia Greca: cane a tre teste guardiano delle porte dell' Ade
- Goal: segretezza, autentica (ad accesso singolo), **temporalità**
 - le chiavi usate hanno validità limitata onde prevenire **replay attack**
- Usa i **timestamp**, che richiedono macchine sincronizzate, contro replay attack



Kerberos – schema





Kerberos – caratteristiche

- 3 fasi: autenticazione, autorizzazione, servizio
- Ultime 2 opzionali e trasparenti all'utente
- Ognuna fornisce credenziali per successiva
 - Fase I fornisce **authKey** e **authTicket** per la II
 - Fase II fornisce **servKey** e **servTicket** per la III
- Ogni tipo di chiave di sessione ha sua durata
- Una authKey può criptare diverse servKey

Kerberos – eventi

I. AUTENTICAZIONE

1. $A \rightarrow AS : A, TGS, T1$

2. $AS \rightarrow A : \{authK, TGS, Ta, \underbrace{\{A, TGS, authK, Ta\}_{K_{tgs}}}_{authTicket}\}_{K_a}$

II. AUTORIZZAZIONE

3. $A \rightarrow TGS : \underbrace{\{A, TGS, authK, Ta\}_{K_{tgs}}}_{authTicket}, \underbrace{\{A, T2\}_{authK}}_{autenticatore\ 1}, B$

4. $TGS \rightarrow A : \underbrace{\{servK, B, Ts, \{A, B, servK, Ts\}_{K_b}\}_{authK}}_{servTicket}$

III. SERVIZIO

5. $A \rightarrow B : \underbrace{\{A, B, servK, Ts\}_{K_b}}_{servTicket}, \underbrace{\{A, T3\}_{servK}}_{autenticatore\ 2}$

6. $B \rightarrow A : \{T3+1\}_{servK}$

Kerberos – gestione delle chiavi

- AS genera authK al tempo T_a
TGS genera servK al tempo T_s
- Validità
 - di authK (ossia di T_a) in ore, diciamo L_a
 - di servK (ossia di T_s) in minuti, diciamo L_s
 - di un autenticatore (ossia di T_1 , T_2 e T_3) in sec.
- Tgs può generare servK solo qualora sia
$$T_s + L_s \leq T_a + L_a$$
altrimenti problema di **cascata dell' attacco**

Attacchi consequenziali

Def. Un attacco ne provoca altri direttamente

- Supponiamo che C abbia violato una chiave di sessione (di autorizzazione) scaduta authK che B condivide con A
- Con semplice decodifica ottiene servK ancora valida se non si impone $T_s + L_s \leq T_a + L_a$

III. SERVICE

5. **C** \rightarrow B : $\{A, B, \text{servK}, T_s\}_{K_b}, \{A, \text{T3}'\}_{\text{servK}}$

6. B \rightarrow A : $\{T3'+1\}_{\text{servK}}$ (intercettato)

- C può accedere a B per la durata residua di L_s



Discussione

- Replay attack su N-S simmetrico nell'ipotesi che chiavi di sessione **vecchie** siano insicure
 - Vecchio: genericamente, del passato – non esiste temporalità
- consequential attack su Kerberos nell'ipotesi che chiavi di sessione **scadute** siano insicure
 - Scaduto: specificatamente, il cui intervallo di validità sia scaduto – esiste temporalità

Autenticazione fra domini

Dominio = realm Kerberos

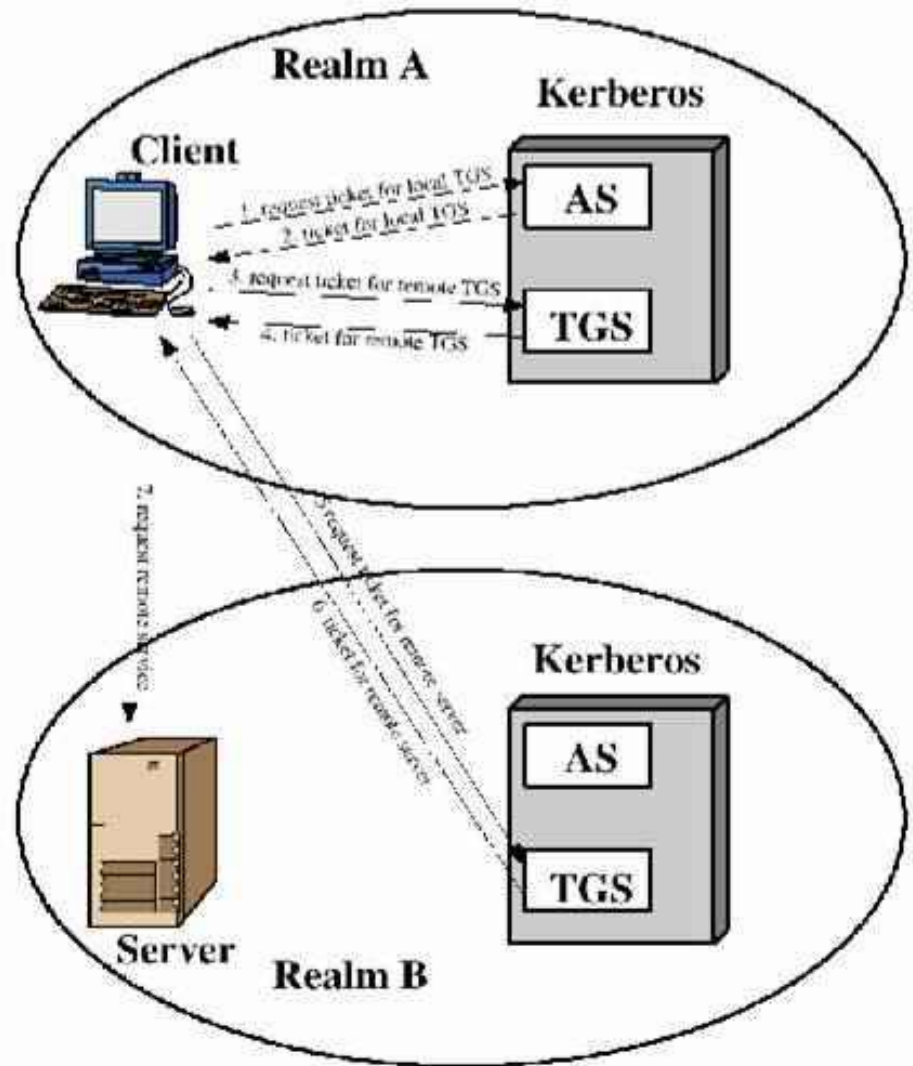


Figure 4.2 Request for Service in Another Realm



Kerberos in pratica

- **Versione IV**: ristretta a un singolo realm
- **Versione V**: funzionamento inter-realm
- Altre differenze: scelta dell'algoritmo di crittografia impossibile in IV (DES); scelta del lifetime impossibile in IV
- Versione V è uno standard di vastissimo utilizzo (specificato in RFC1510)



Usare Kerberos

- Serve un KDC sul proprio dominio
- Tutte le applicazioni partecipanti devono essere servizi Kerberos
- Problema: gli algoritmi crittografici americani non possono essere esportati
 - I sorgenti di Kerberos non possono lasciare gli USA
 - Il crittosistema va implementato localmente



Discussion:
