# Reti di Calcolatori e Sicurezza

## 5. Autenticazione

Cap. 9,20(seconda metà) Schneier
Capitolo 7 Kurose

Thanks to Giampaolo Bella for slides
draft!!



# Autenticazione

- Stabilisce l'identità di una "parte" ad un'altra
- Le parti possono essere utenti o computer
  1. computer-computer (*stampa in rete, delega,…*)
  2. utente-utente (*protocolli di sicurezza, …*)
  3. computer-utente (*autenticare un server web,…*)
  4. utente-computer (*per accedere a un sistema…*)
- Spesso richieste varie combinazioni
- Proprietà primaria
  - *Richiesta da un corretto controllo d'accesso*

Thanks to Giampaolo Bella for slides draft!!



# Tipi di authenticazione

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

Thanks to Giampaolo Bella for slides draft!!

# Autenticazione indiretta


- Protocollo di autenticazione
  1. U → S : username
  2. S → U : challenge
  3. U → S : response


- Protocollo di autenticazione indiretto
  - U → $S_R$ : logon request (chi sono e cosa voglio)
  - $S_R$ → $S_A$ : authentication request
  - $S_A$ → $S_R$ : authentication response (A/R)
  - $S_R$ → U : logon response

Thanks to Giampaolo Bella for slides draft!!



# 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
   {“salve Bob, sono il server web della tua banca”}$K^{-1}$bancaxy
3. Bob scarica il certificato per la chiave pubblica della banca
   {“banca XY”, Kbancaxy}$K^{-1}$CA
   lo verifica ed estrae Kbancaxy
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

Thanks to Giampaolo Bella for slides
draft!!

# 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,…*


Thanks to Giampaolo Bella for slides draft!!



# What I know

Thanks to Giampaolo Bella for slides draft!!



# Authentication

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

Apriti Sesamo!

Thanks to Giampaolo Bella for slides draft!!



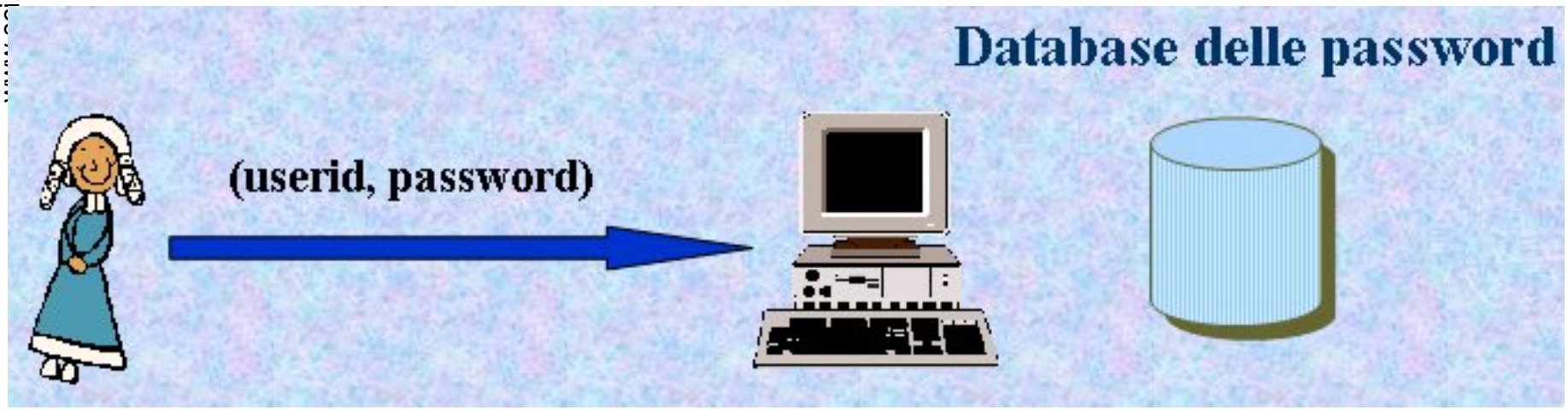
# 1. Autenticazione su conoscenza

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

Thanks to Giampaolo Bella for slides
draft!!

# Gestione delle password

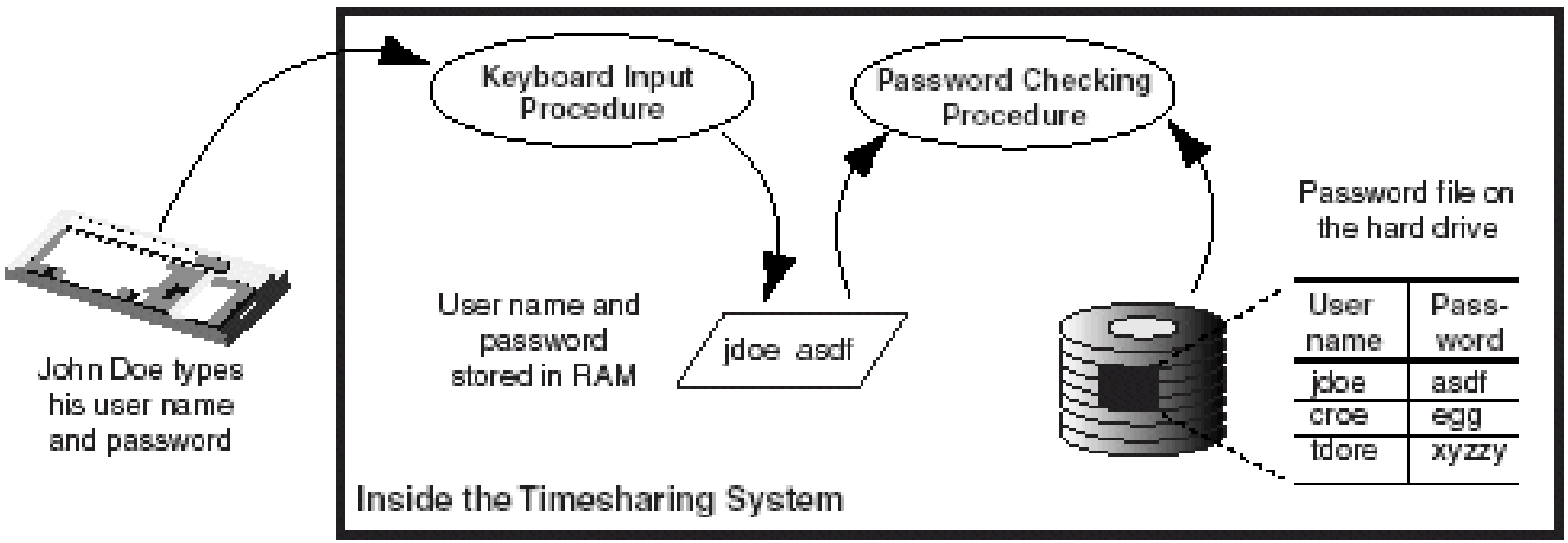
- Il sistema deve memorizzare una rappresentazione delle password. Come??




Thanks to Giampaolo Bella for slides draft!!

# Compatible Time Sharing System (CTSS)

- 1960, MIT
- Password memorizzate in chiaro su file di sistema protetto da politica di sicurezza




Thanks to Giampaolo Bella for slides draft!!



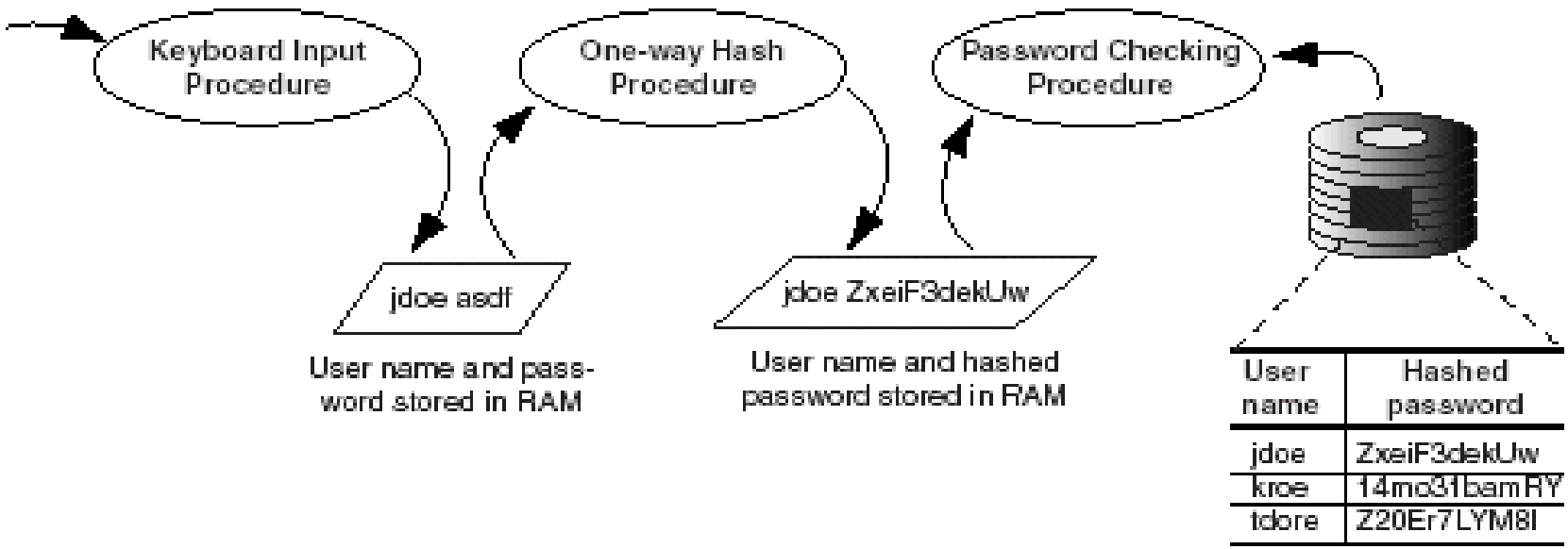
# CTSS

- 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!! ☹

Thanks to Giampaolo Bella for slides draft!!

# CTSS + hashing

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






draft!!

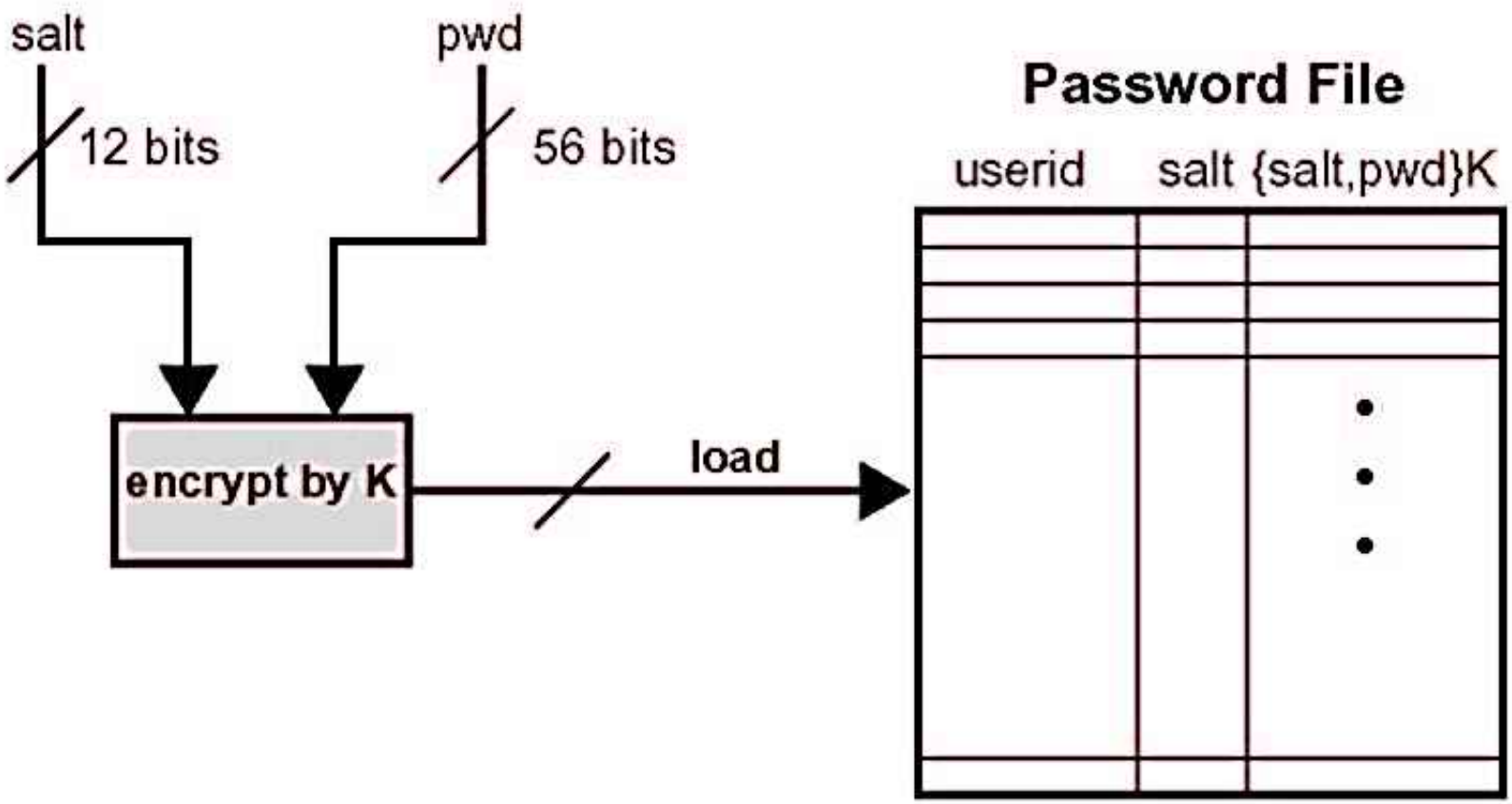


# Password in Unix (cenni)

- 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”

Thanks to Giampaolo Bella for slides draft!!

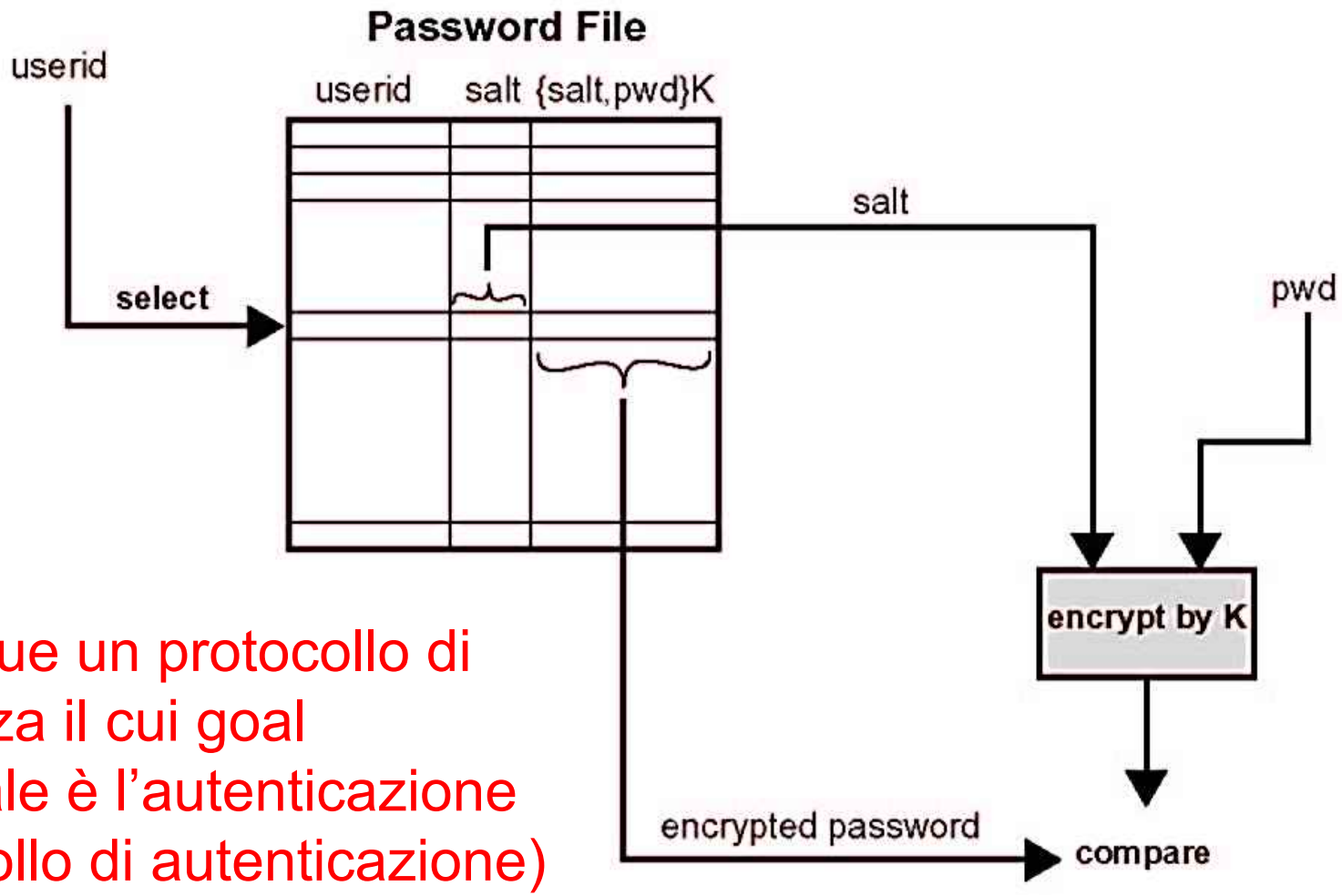
# Unix: aggiunta di nuova password




Si esegue un protocollo di sicurezza

Thanks to Giampaolo Bella for slides draft!!

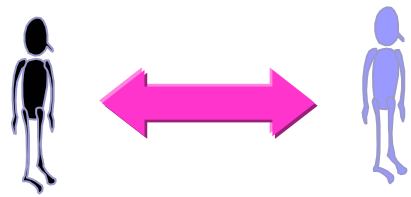
# Unix: verifica di una password




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

Thanks to Giampaolo Bella for slides draft!!

# 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,

Thanks to Giampaolo Bella for slides draft!!



# 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!

Thanks to Giampaolo Bella for slides draft!!



# … E difese

- Guessing attack
  - Audit-log
  - Limite agli sbagli
- Social enginerring
  - 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)

Thanks to Giampaolo Bella for slides draft!!

# Scelta della password

## Delicata a causa dei rischi di guessing


Thanks to Giampaolo Bella for slides draft!!



# 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)

Thanks to Giampaolo Bella for slides
draft!!



# 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



Thanks to Giampaolo Bella for slides draft!!



# 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



Thanks to Giampaolo Bella for slides draft!!



# 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,…*

Thanks to Giampaolo Bella for slides draft!!



# 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

Thanks to Giampaolo Bella for slides

draft!!



# 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

Thanks to Giampaolo Bella for slides draft!!



# 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

Thanks to Giampaolo Bella for slides draft!!



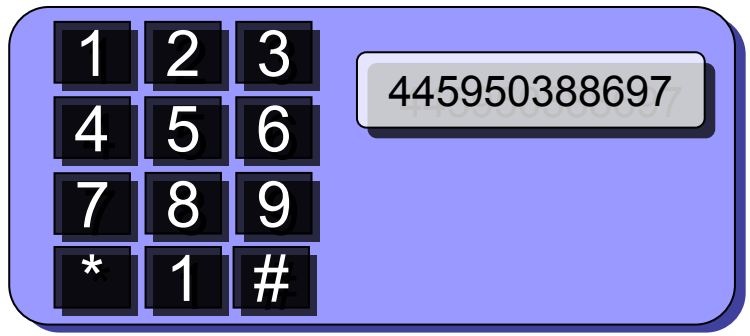
# Tipi di token

- **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**

Thanks to Giampaolo Bella for slides draft!!



# Smart token




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

Thanks to Giampaolo Bella for slides draft!!

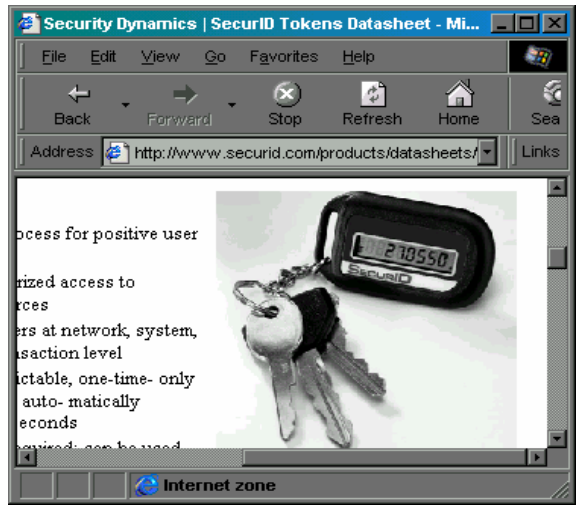


# Smart token – funzionamento

- 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

Thanks to Giampaolo Bella for slides draft!!

# Smart token commerciali



Thanks to Giampaolo Bella for slides draft!!



# 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

Thanks to Giampaolo Bella for slides draft!!



# 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.

Thanks to Giampaolo Bella for slides draft!!

# Problema

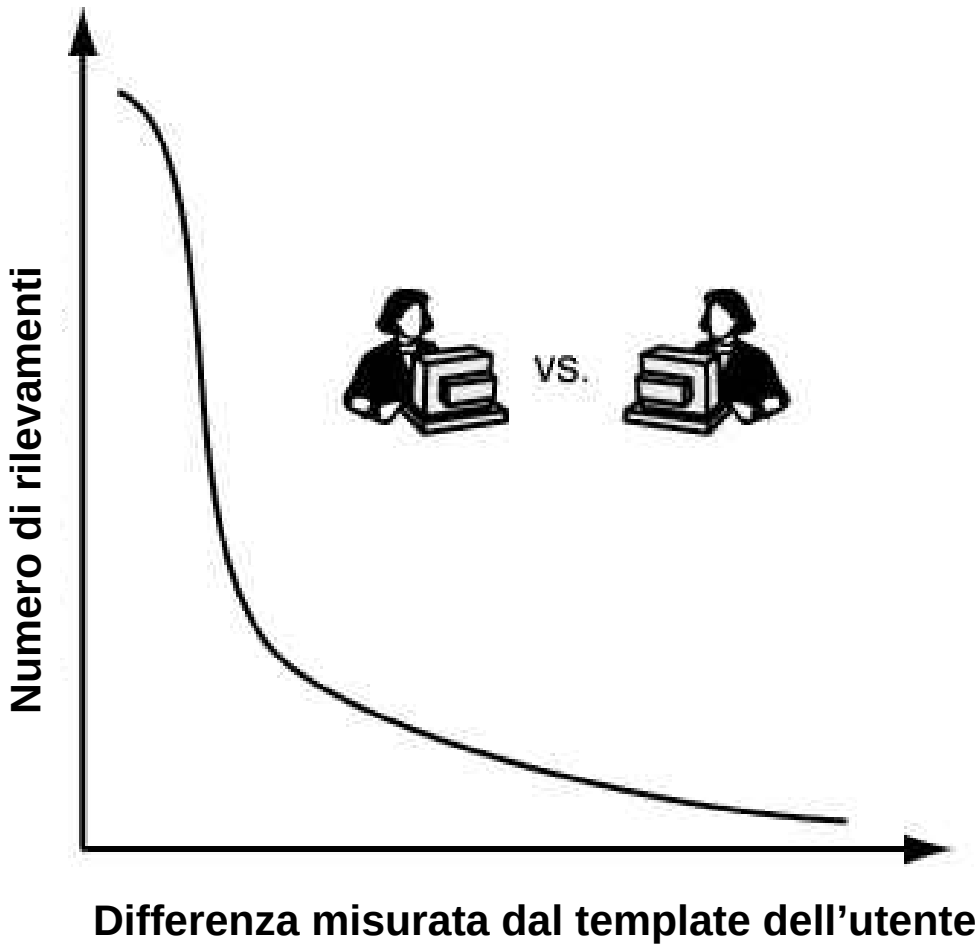
- Confrontare la caratteristica appena misurata dall'utente col template di quell'utente
- Distinguendola dal template di un altro utente!

| Today's Biometric Signature from Cathy: | Cathy's Stored Biometric Pattern: | Tim's Stored Biometric Pattern: |
|---|---|---|
| 389 | 390 | 284 |
| 416 | 418 | 570 |
| 501 | 502 | 534 |
| 468 | 471 | 501 |
| 353 | 355 | 399 |
| | Distance = 4 from that signature | Distance = 199 from that signature |

Thanks to Giampaolo Bella for slides draft!!

# Rilevamenti

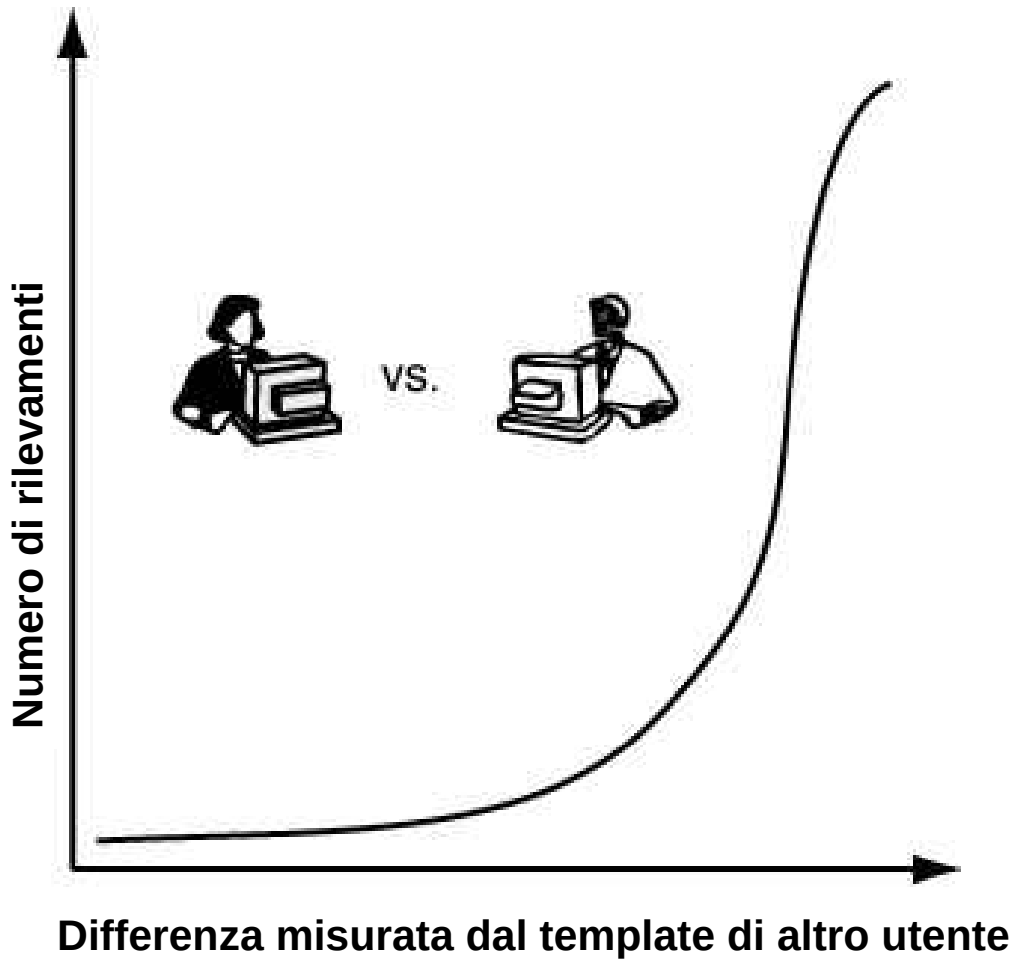
- Aumentando il numero di rilevamenti in fase di autenticazione diminuisce la distanza dal template del giusto utente




Thanks to Giampaolo Bella for slides draft!!

# Rilevamenti

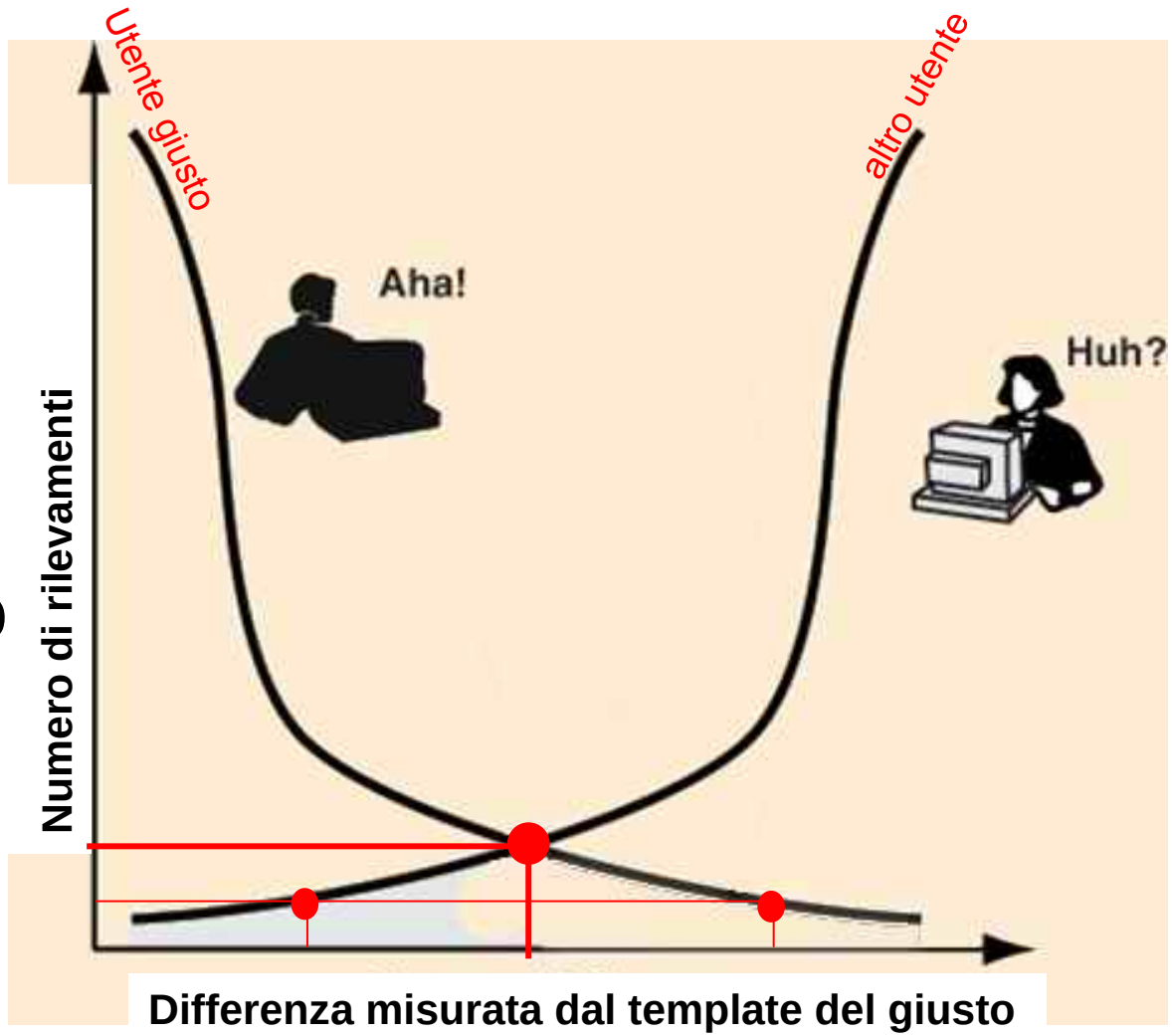
- Aumentando il numero di rilevamenti in fase di autenticazione aumenta la distanza dal template di un altro utente




draft!!

# Punto di “equal error rate”

- Vorremmo diminuire i rilevamenti per praticità
- Però arriviamo al punto di massima indecisione!




**Differenza misurata dal template del giusto utente e dal template di altro utente**



# 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

Thanks to Giampaolo Bella for slides draft!!

# 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

Thanks to Giampaolo Bella for slides draft!!

# Dall'inchiostro…

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

Thanks to Giampaolo Bella for slides draft!!

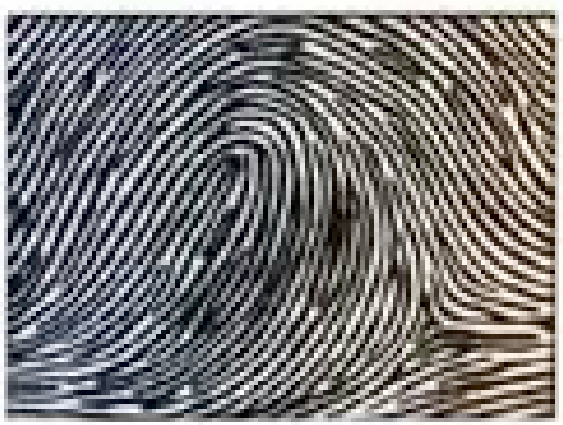
# … ai lettori ottici

- Ora basta poggiare un attimo il dito sul lettore

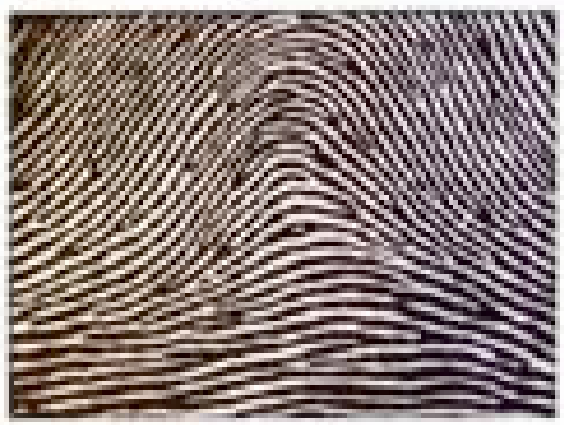

Thanks to Giampaolo Bella for slides draft!!

# Classificazioni di impronte

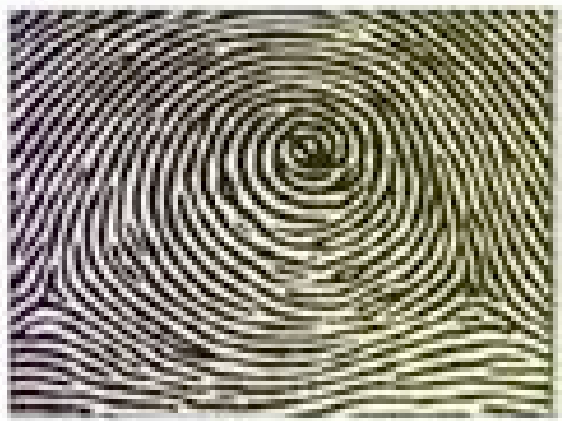
- Classificate in tre grandi gruppi in base allo “schema” predominante



loop



arch



whorl

Thanks to Giampaolo Bella for slides draft!!



# Schema Loop

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

Thanks to Giampaolo Bella for slides draft!!



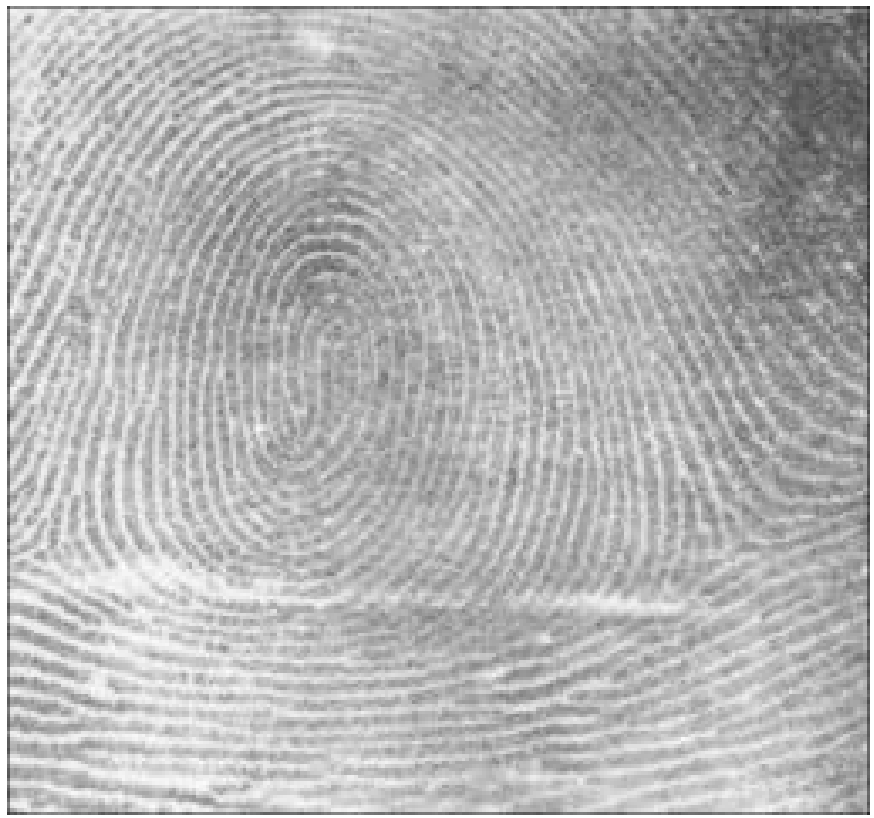
# Schema Arch

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

Thanks to Giampaolo Bella for slides draft!!

# Schema Whorl

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

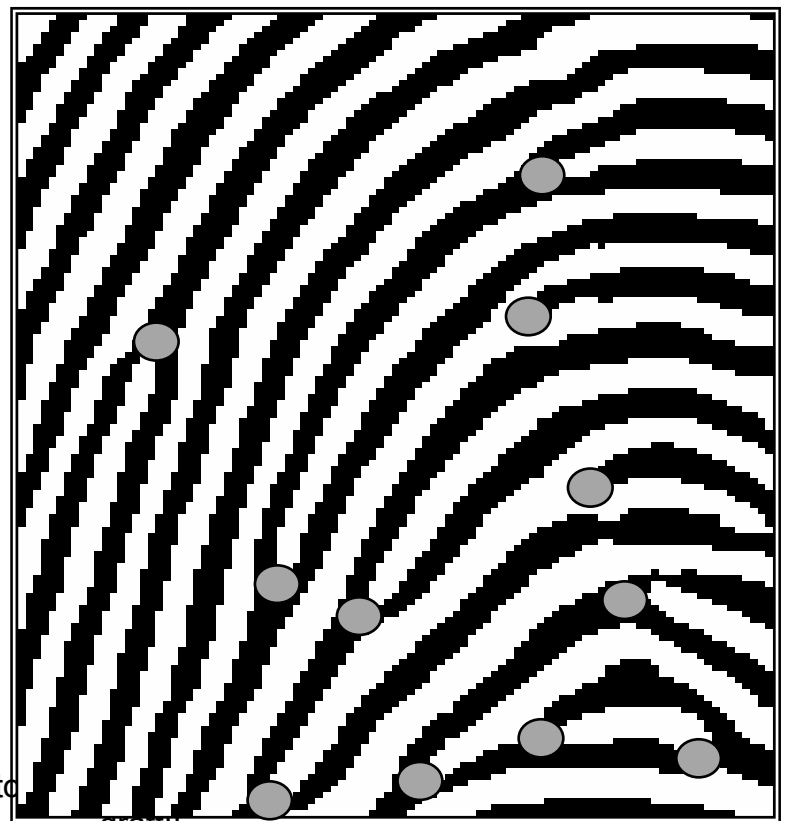


Thanks to Giampaolo Bella for slides draft!!



# 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



Thanks to
draft!!



# 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

Thanks to Giampaolo Bella for slides draft!!



Thanks to Giampaolo Bella for slides draft!!



# 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!

Thanks to Giampaolo Bella for slides draft!!



# 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)

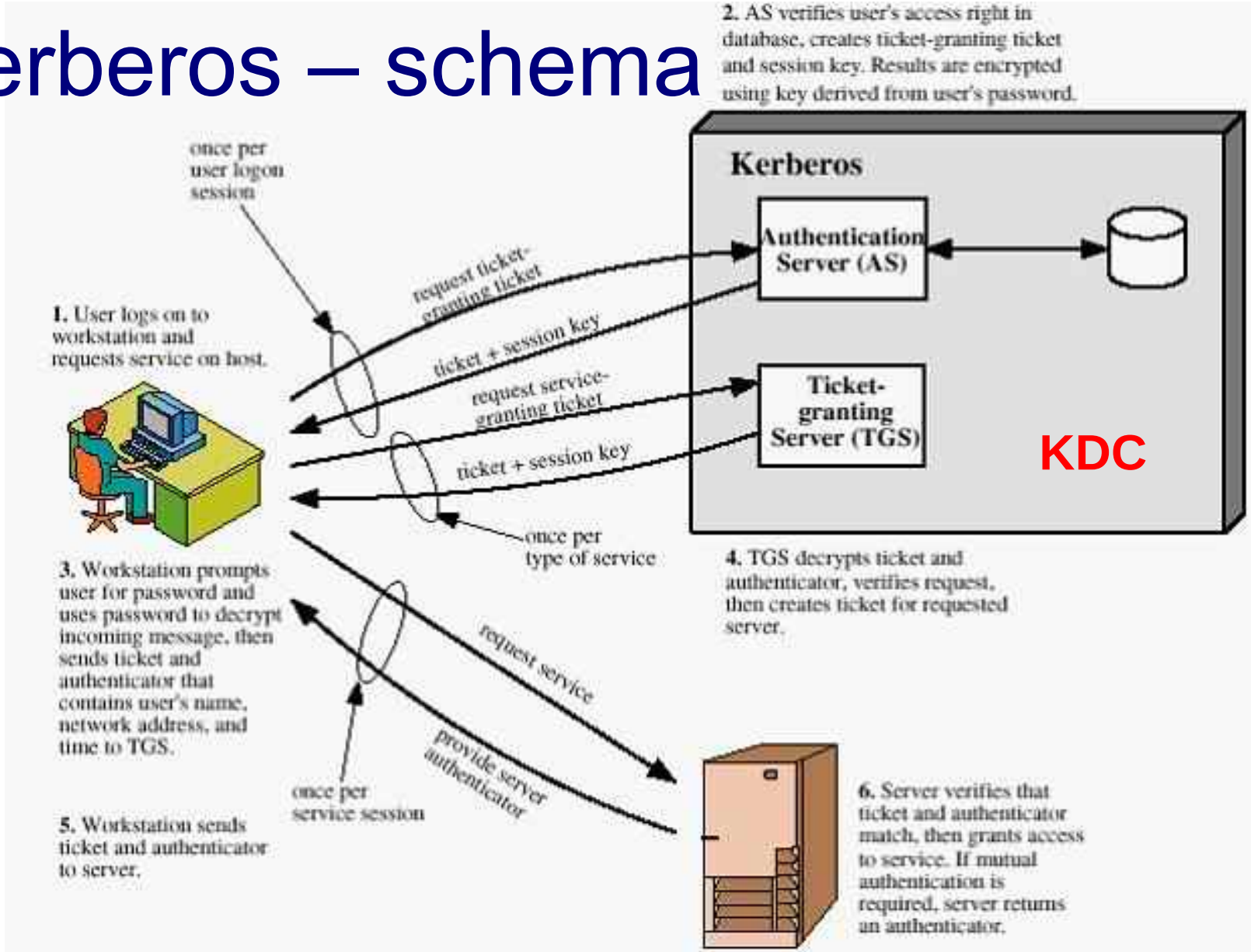
Thanks to Giampaolo Bella for slides draft!!

# 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

Thanks to Giampaolo Bella for slides draft!!

# Kerberos – schema






draft!!



# 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



Thanks to Giampaolo Bella for slides draft!!

# Kerberos – eventi

**I. AUTENTICAZIONE**

1. $A \rightarrow AS : A, TGS, T1$
2. $AS \rightarrow A : \{authK, TGS, Ta, \underbrace{\{A, TGS, authK, Ta\}K_{tgs}}_{\text{authTicket}}\}K_a$

**II. AUTORIZZAZIONE**

3. $A \rightarrow TGS : \overbrace{\{A, TGS, authK, Ta\}K_{tgs}}^{\text{authTicket}}, \overbrace{\{A, T2\}authK}^{\text{autenticatore 1}}, B$
4. $TGS \rightarrow A : \{servK, B, Ts, \underbrace{\{A, B, servK, Ts\}K_b}_{\text{servTicket}}\}authK$

**III. SERVIZIO**

5. $A \rightarrow B : \overbrace{\{A, B, servK, Ts\}K_b}^{\text{servTicket}}, \overbrace{\{A, T3\}servK}^{\text{autenticatore 2}}$
6. $B \rightarrow A : \{T3+1\}servK$

Thanks to Giampaolo Bella for slides draft!!



# Kerberos – gestione delle chiavi

- AS genera authK al tempo Ta
  TGS genera servK al tempo Ts
- Validità
  - di authK (ossia di Ta) in ore, diciamo La
  - di servK (ossia di Ts) in minuti, diciamo Ls
  - di un autenticatore (ossia di T1, T2 e T3) in sec.
- Tgs può generare servK solo qualora sia
  $$Ts + Ls \leq Ta + La$$
  altrimenti problema di **cascata dell'attacco**

Thanks to Giampaolo Bella for slides draft!!



# Cascade attack

**Def.** Un attacco ne provoca altri direttamente

- Supponiamo che C abbia violato una chiave di sessione (di autorizzazione) scaduta authK che B condivise con A
- Con semplice decodifica ottiene servK ancora valida se non si impone Ts + Ls ≤ Ta + La

**III. SERVICE**

5. C →B : {A,B,servK,Ts}Kb, {A,T3'}servK
6. B →A : {T3'+1}servK (intercettato)

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

Thanks to Giampaolo Bella for slides draft!!



# Discussione

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

Thanks to Giampaolo Bella for slides draft!!

# Autenticazione fra domini

**Dominio = realm Kerberos**

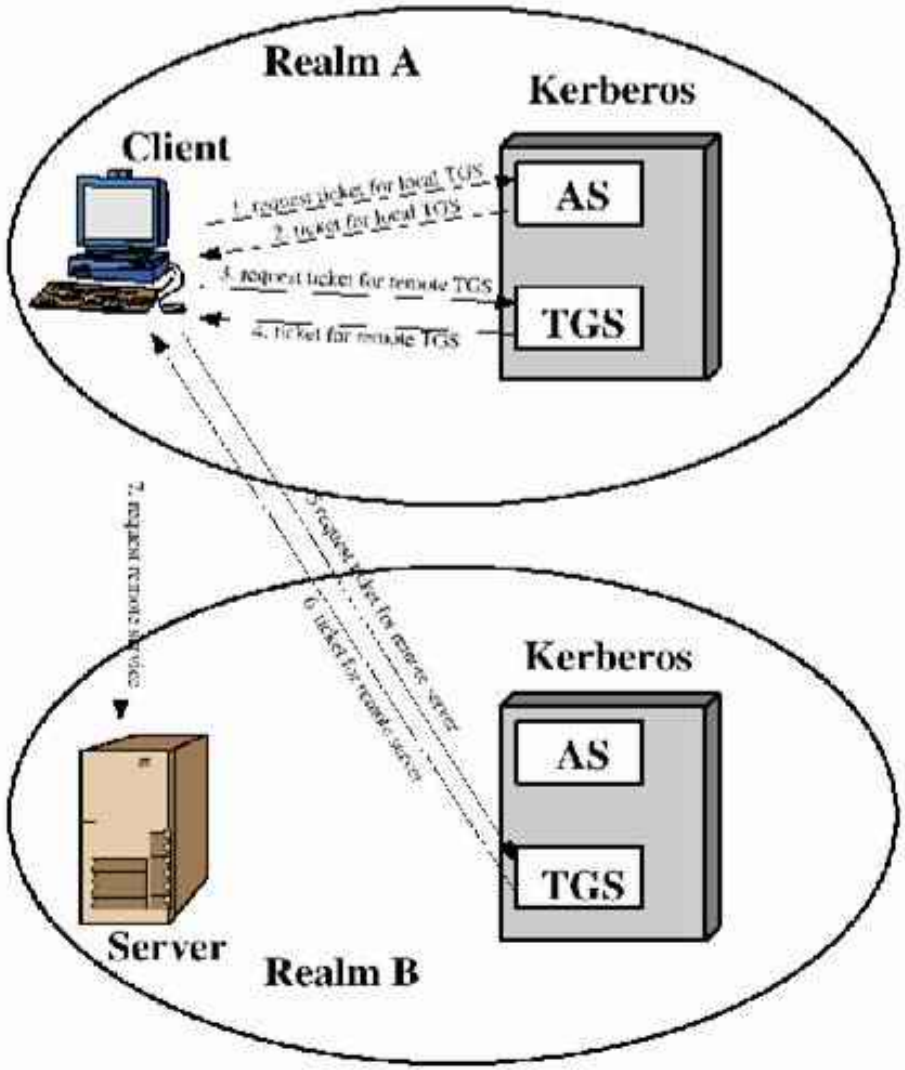



Figure 4.2 Request for Service in Another Realm



# 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)

Thanks to Giampaolo Bella for slides draft!!



# 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

Thanks to Giampaolo Bella for slides draft!!