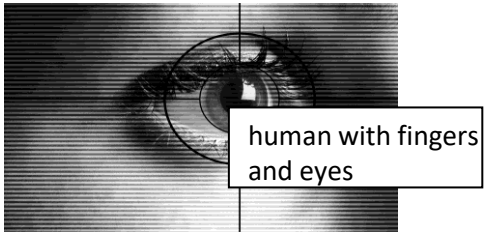
User Authentication and Applications

Dr. Chen Zhang

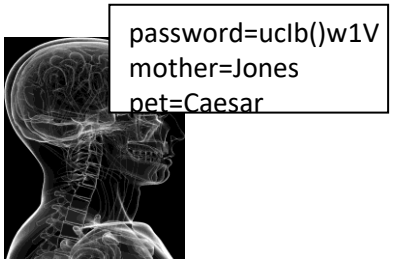
Department of Computer Science
The Hang Seng University of Hong Kong

Authentication

- The act of proving an assertion, such as the identity of a computer system user.
- Fundamental building block and primary line of defense
- Basics for access control
- The determination of **identity**, usually based on a combination of
  - something the person knows
  - something the person has/possesses
  - something the person is
  - something the individual does



Something you are



Something you know

Something you have

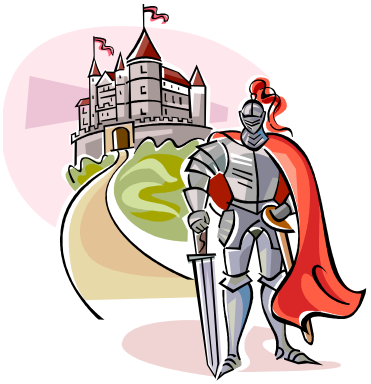
Means of Authentication

the four means of authenticating user identity are based on:

something the individual knows	something the individual possesses (token)	something the individual is (static biometrics)	something the individual does (dynamic biometrics)
• password, answers to prearranged questions	• smartcard, electronic keycard, physical key	• fingerprint, retina, face	• voice pattern, handwriting, typing rhythm

Authentication Process

- An authentication process consists of two steps:
  - Identification step
    - Presenting an identifier to the security system (I set my identifier as CHEH).
  - Verification step
    - Presenting or generating authentication information that verifies the binding between the entity and the identifier (prove I am CHEN).



E-channel

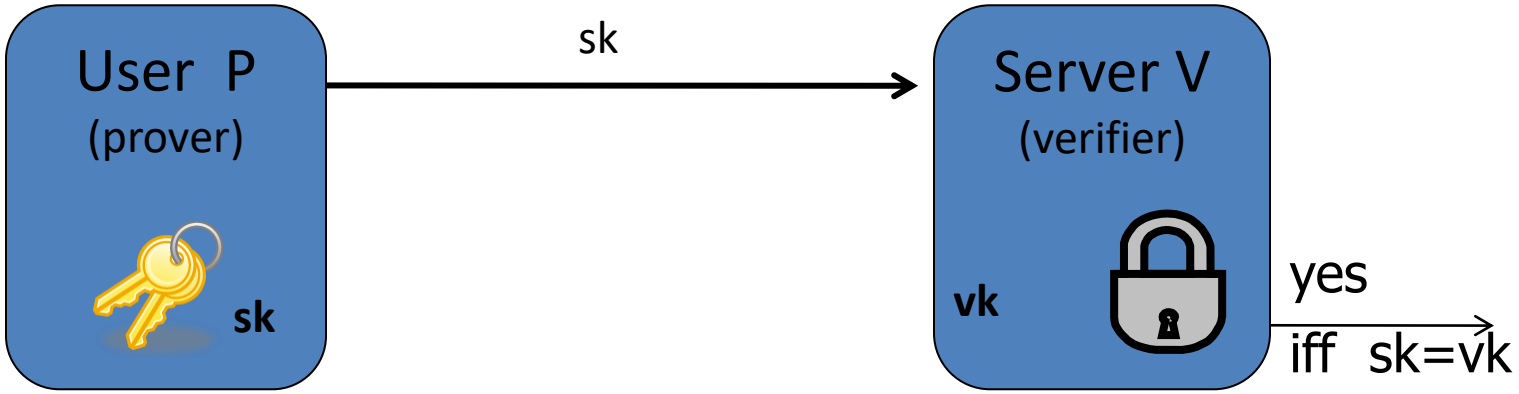
User Authentication with passwords

Password Authentication

- Widely used line of defense against intruders
  - user provides name/login and password
  - system compares password with the one stored for that specified login
- The user ID:
  - determines whether the user is authorized to access the system
  - determines the user's privileges
  - is used in access control

Basic Password Protocol (incorrect version)

- **PWD**: finite set of passwords
- Algorithm G (KeyGen):
  - choose rand pw in PWD.
  - output sk = vk = pw. (vk is the key stored in server for verification)



Basic Password Protocol (incorrect version)

- Problem: vk must be kept secret
  - Compromise of server exposes all passwords
  - Never store passwords in the clear!

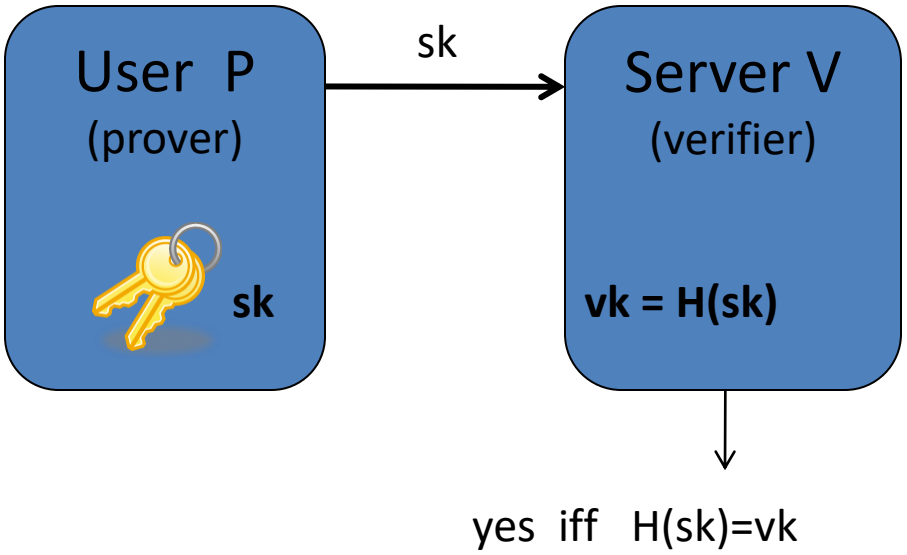
password file on server

Alice	pw_{alice}
Bob	pw_{bob}
…	…

Basic Password Protocol: version 1

H: one-way hash function from PWD to X

"Given H(x), it is difficult to find y such that H(y)=H(x)"



password file on server

Alice	$H(pw_A)$
Bob	$H(pw_B)$
…	…

Weak Passwords and Dictionary Attacks

- People often choose passwords from a small set:
  - The 6 most common passwords (sample of 32×10^6 pwds):
    123456, 12345, Password, iloveyou, princess, abc123
    ('123456' appeared 0.90% of the time)
  - 23% of users choose passwords in a dictionary of size 360,000,000

- **Dictionary Attacks:**
  - Online dictionary attacks: performed by trying each password in a separate request to the server, and is therefore visible to the server and also limited in speed by the server's processing power and the network capacity between the client and the server.
    - Defeated by doubling response time after every failure
    - Harder to block when attacker commands a bot-net
  - Offline Dictionary attacks: obtain the ciphertext and try each password against the ciphertext.

Offline Dictionary Attacks

- Suppose attacker obtains vk = H(pw) from server
  - **Offline** attack: hash all words in Dict until a word w is found such that H(w) = vk
  - Time O(|Dict|) per password
- Off the shelf tools
  - 2,000,000 brute-force guesses/sec
  - Scan through 360,000,000 guesses in few minutes
    - Will recover 23% of passwords

Password Crackers

- Many tools for this
  - John the ripper
  - Cain and Abel
  - Passware (Commercial)

* When using these tools, you need to comply with legal and ethical guidelines and ensure authorized use to protect personal privacy and data security.

Batch Offline Dictionary Attacks

- Suppose attacker steals pwd file F
  - Obtains hashed pwds for **all** users
- Batch dict. attack:
  - Build list L containing **(w, H(w))** for all w ∈ Dict
  - Find intersection of L and F
- Total time: **O(|Dict| + |F|)**
- Much better than a dictionary attack on each password

Preventing Batch Dictionary Attacks

- Public salt:
  - When setting password, pick a random n-bit salt S
  - When verifying pw for A, test if $\mathbf{H(pw, S_A) = h_A}$
- Recommended salt length, n = 64 bits
  - Pre-hashing dictionary does not help
- Batch attack time is now: $O(\ |\mathbf{Dict}| \times |\mathbf{F}|\ )$

Procedure of Storing Hashed Passwords with Salt

- To load a new password into the system, this password is combined with a fixed-length **salt** value.
  - Salt from the time when the password is assigned to the user.
    - Some old implementation.
  - Salt from a pseudorandom or random number.
- The **password** and **salt** serve as inputs to a hashing algorithm to produce a fixed-length hash code.
  - The hash algorithm is designed to be **slow** to execute to thwart attacks .
- The hashed password is then stored, together with a plaintext copy of the salt, in the password file for the corresponding user ID.

When users login…

- To login, the user provides an ID and a password.
- The OS uses the ID to index into the password file and retrieve the plaintext salt and the hashed password.
- The salt and user-supplied password are used as input to the slow hash routine.

Use of Hashed Passwords

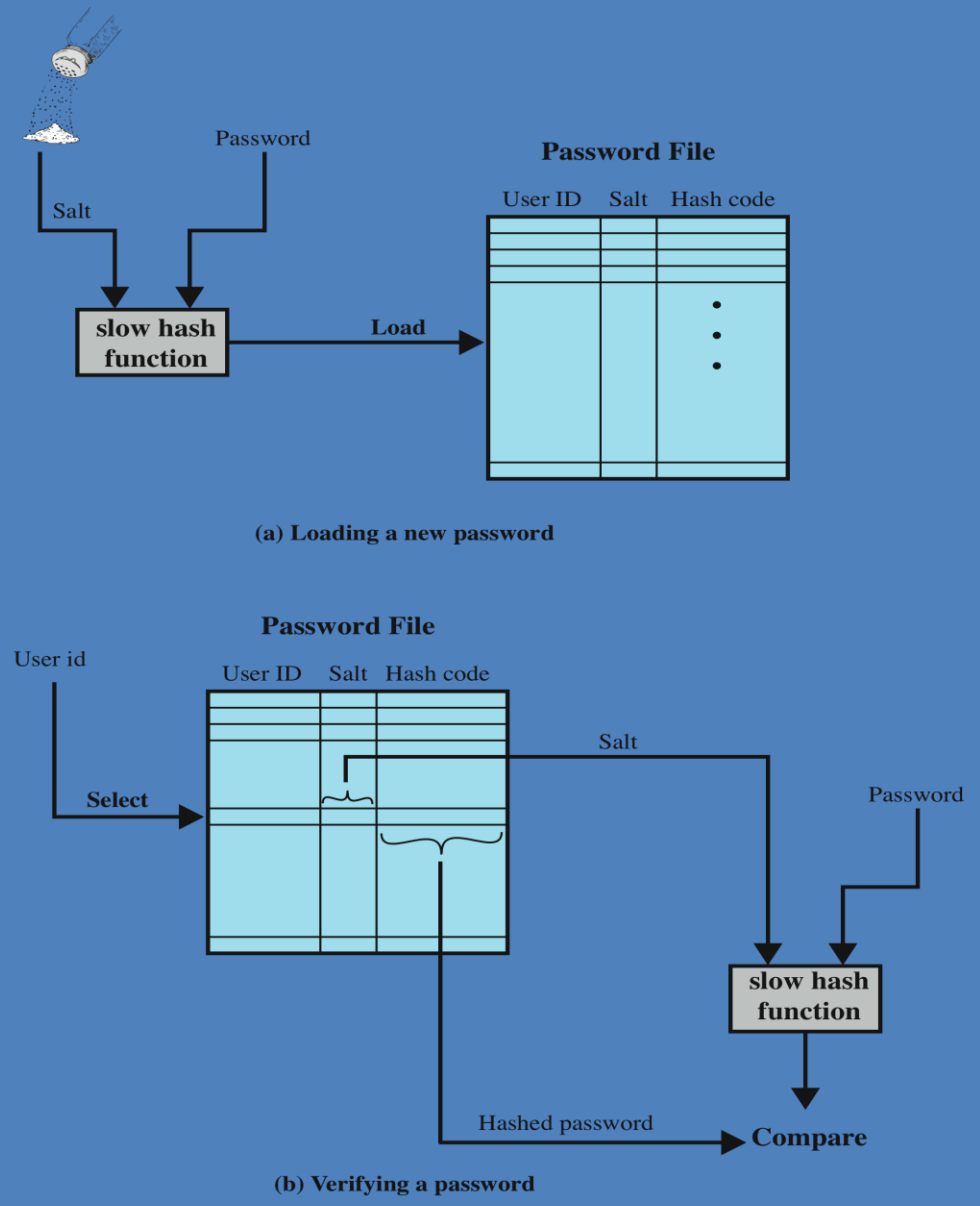


Figure 3.1 UNIX Password Scheme

Summary on Purposes of Using Salt

- It prevents duplicate passwords from being visible in the password file.
  - Even if two users choose the same password, those passwords will be assigned different salt values.

- Hence, the hashed passwords of the two users will differ.

Summary on Purposes of Using Salt (Cont'd)

- It greatly increases the difficulty of offline dictionary attacks.
  - As noted previously,

    batch attack time is now: O**(|Dict| × |F|)**
- It becomes nearly impossible to find out whether a person with passwords on two or more systems has used the same password on all of them.

Further Defenses

- **Slow hash function** H: (0.1 sec to hash pw)
  - Example: H(pw) = SHA1(SHA1(… SHA1(pw) …))
  - Unnoticeable to user, but makes offline dictionary attack harder

Alice	$\mathbf{S_A}$	$H(pw_A , S_A , \mathbf{r_A})$
Bob	$\mathbf{S_B}$	$H(pw_B , S_B , \mathbf{r_B})$
…	…	…

- **Secret salts**:
  - When setting pwd choose short random r (8 bits)
  - When verifying pw for A, try all values of r_A: 128 times slow down on average
  - It also serves for purpose of slowing down attacker.

Password Complexity also Matters

- Password length is only part of the problem. Many people, when permitted to choose their own password, pick a password that is guessable.
  - such as their own name, their street name, a common dictionary word, and so forth.
- General guessing strategy:
  - 1. Try the user's name, initials, account name, and other relevant personal information
  - 2. Try words from various dictionaries (accessible online)
  - 3. Try various permutations on the words from step 2.
    - first letter uppercase or a control character, making the entire word uppercase, reversing the word, changing the letter "o" to the digit "zero," and so on.
  - 4. Try various capitalization permutations on the words from step 2 that were not considered in step 3.

Remarks on the Common Password Problem

- Users tend to use the same password at many sites
  - Password at a high security site can be exposed by a break-in at a low security site

- Standard solution:
  - The client side software converts a common password pw into a unique site password

    $$pw' \leftarrow H(\, pw, \text{user-id}, \text{server-id} \,)$$

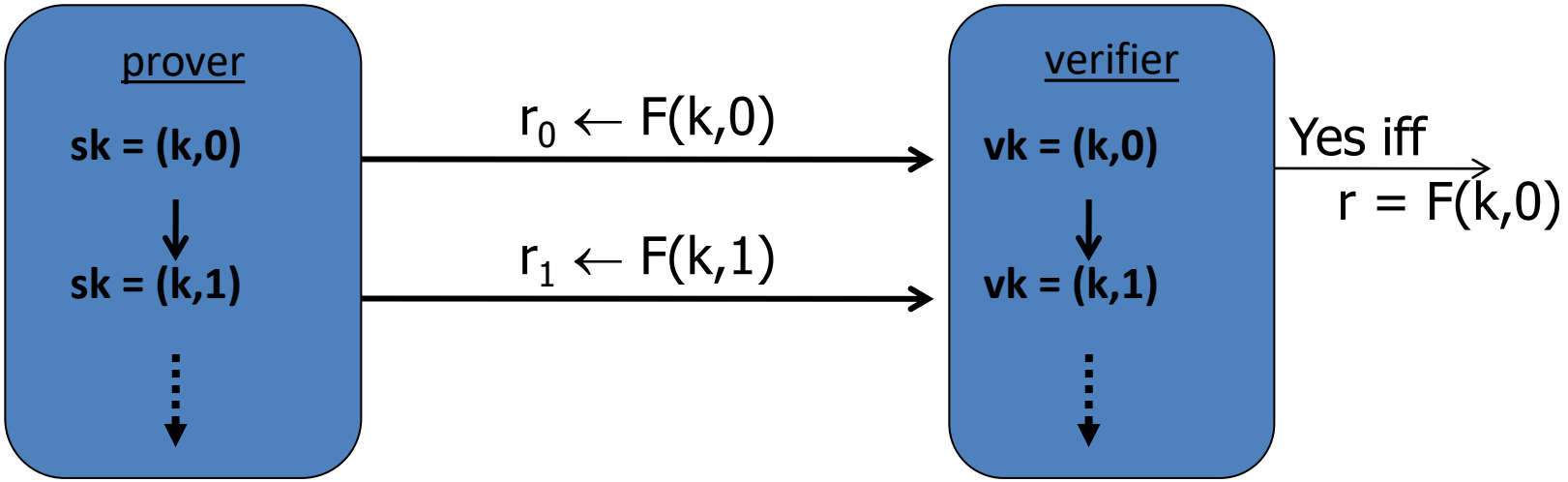
    pw' is sent to server

User Authentication via One-time Passwords

The SecurID system

- https://www.rsa.com/products/securid/
- Algorithm G: (setup)
  - Choose random key $k \leftarrow K$
  - Output **sk = (k,0)** ; **vk = (k,0)**

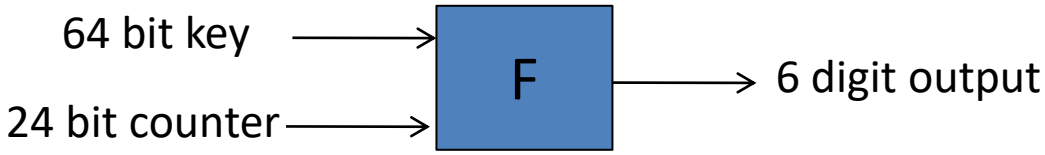
- Identification:



The SecurID system

- "Thm": if F is a secure PRF then protocol is secure against eavesdropping

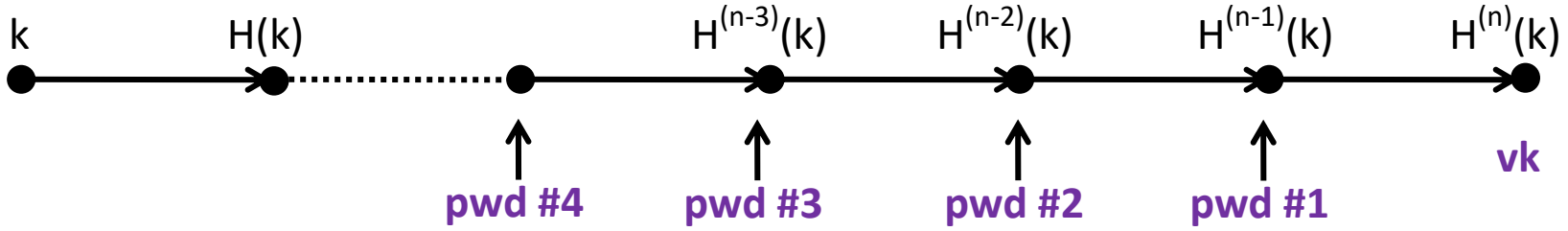
- RSA SecurID uses a custom PRF:



- Advancing state: $sk \leftarrow (k, i+1)$
  - Time based: every 60 seconds
  - User action: every button press

The Hash Chain

- Notation: $H^{(n)}(x) = \underbrace{H(H(\ldots H(x)\ldots))}_{\text{n times}}$
- Algorithm G: (setup)
  - Choose random key $k \leftarrow K$
  - Output **sk = (k,i); vk = $H^{(i+1)}(k)$, initially set i=n-1**
- Identification:



The Hash Chain

- Identification (in detail):

  - Prover (**sk=(k,i)**): send **$t \leftarrow H^{(i)}(k)$** ; set **$sk \leftarrow (k, i-1)$**

    (Initially, i is set to n-1. i can be n-1, n-2, …, 1)

  - Verifier(**$vk=H^{(i+1)}(k)$**): if **$H(t)=vk$** then **$vk \leftarrow t$**, output "yes"

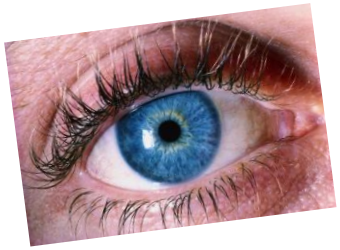
  - Notes: vk can be made public; but need to generate new sk after n-1 logins ($n \approx 10^6$)

- "Thm": The hash chain is secure against eavesdropping.
  The provided H is one-way on n-iterates

User Authentication/Identification with Biometrics

Biometric Authentication

- Attempts to authenticate an individual based on unique physical characteristics based on pattern recognition
- Technically complex and expensive when compared to passwords and tokens.
- physical characteristics used include:
  - facial characteristics
  - fingerprints
  - hand geometry
  - retinal pattern
  - signature
  - voice



Requirements for Biometrics

- **Universality.** Almost every person should have this characteristic.
- **Distinctiveness.** Each person should have noticeable differences in the characteristic.
- **Permanence.** The characteristic should not change significantly over time.
- **Collectability.** The characteristic should have the ability to be effectively determined and quantified.

Pros and Cons of Using Biometrics

- Benefits:
  - hard to forget
  - Individually unique

- Problems:
  - Biometrics are not generally secret
  - Cannot be changed, unlike passwords

- $\Rightarrow$ Primarily used as a second factor authentication

General Operation of a Biometric System

- Each individual who is to be included in the database of authorized users must first be enrolled in the system.
  - Similar to assigning a password to a user.
  - The system maintains the user a name (ID), perhaps a password, and the biometric value.
- Depending on application, user authentication on a biometric system involves either verification or identification.
  - For biometric **verification**, the user enters a Personal Identification Number (PIN) and also uses a biometric sensor.
    - The system extracts the corresponding feature and compares that to the template.
    - Match? ---> the system authenticates the user.
  - For an **identification** system, the individual uses the biometric sensor but presents no additional information.
    - The system then compares the presented template with the set of stored templates.
    - Match? ---> the user is identified.

General Operation of a Biometric System

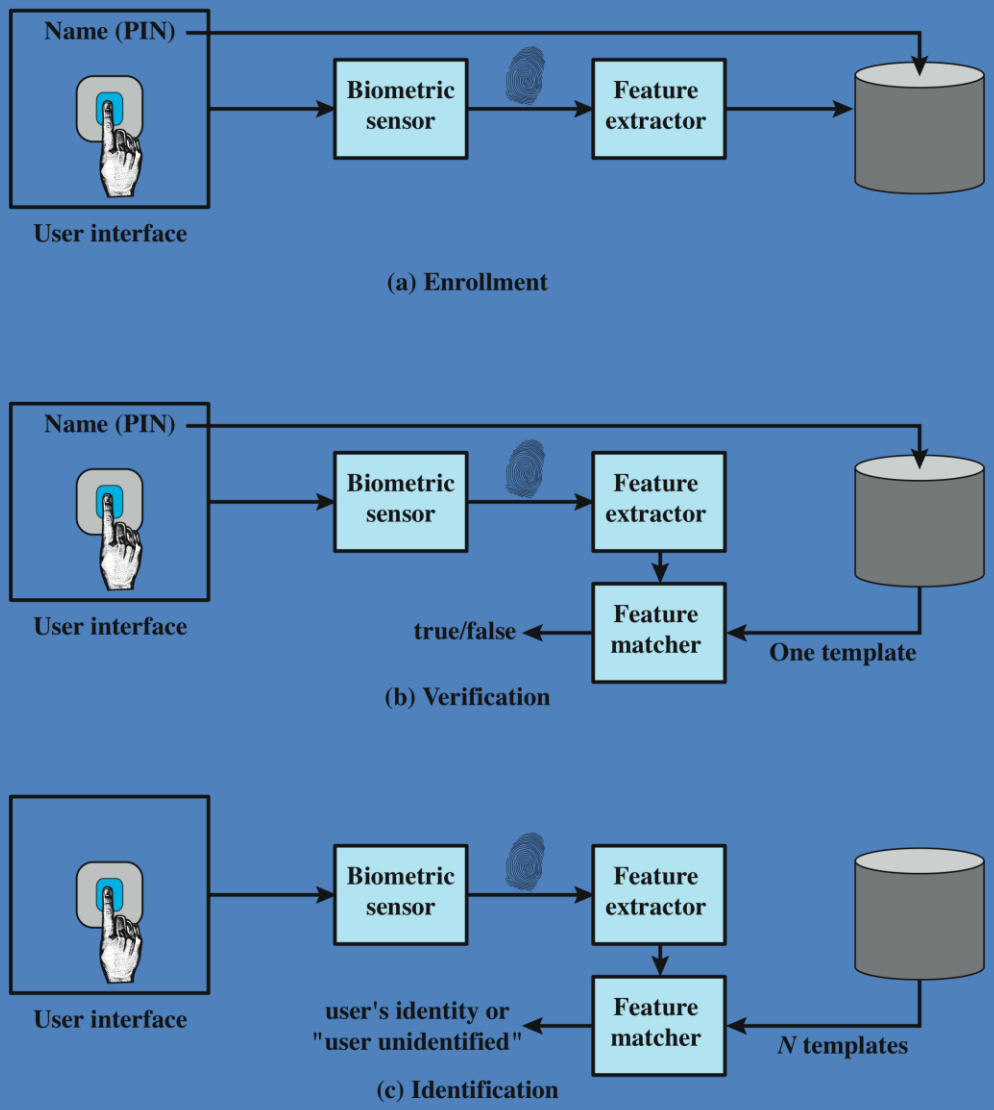


Figure 3.6 A Generic Biometric System Enrollment creates an association between a user and the user's biometric characteristics. Depending on the application, user authentication either involves verifying that a claimed user is the actual user or identifying an unknown user.

Biometric Authentication Accuracy

Biometric Authentication Accuracy

- In any biometric scheme, some physical characteristic of the individual is mapped into a **digital representation**.
  - For each individual, a single digital representation, or template, is stored in the computer.

- When the user is to be authenticated, the system compares the stored template to the presented template.
  - Given the complexities of physical characteristics, we cannot expect that there will be an exact match between the two templates.

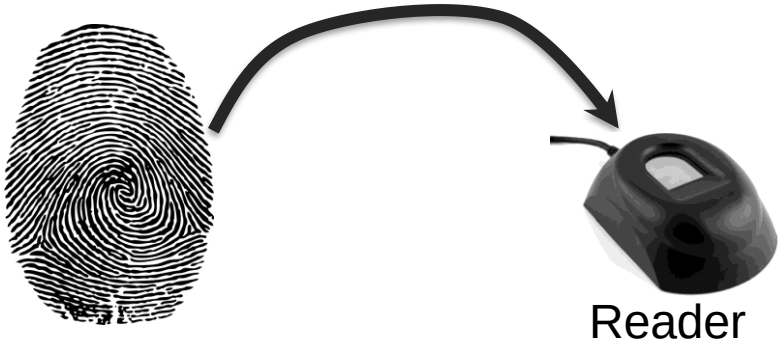
- System uses an algorithm to generate a **matching score** (typically a single number) that **quantifies the similarity** between the input and the stored template.

Accuracy

- The concept of accuracy does not apply to user authentication schemes using passwords.
  - either matches exactly or not.

- In the case of biometric parameters, the system instead must determine how closely a presented biometric characteristic matches a stored characteristic.

Biometric Authentication Accuracy



Biometric

Reader

Feature vector

Comparison algorithm

Reference vector

Distance (feature vector, reference vector) < threshold?

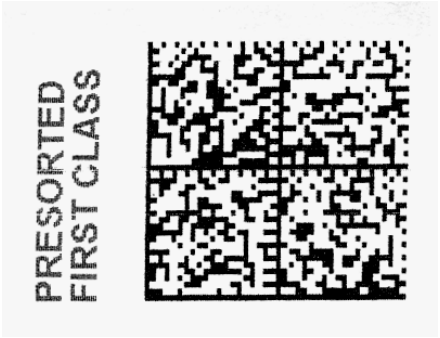
matches

doesn't match

Other Authentication/Identification Techniques

Barcodes

- Developed in the 20th century to improve efficiency in grocery checkout.
- First-generation barcodes represent data as a series of **variable-width, vertical lines** of ink, which is essentially a one-dimensional encoding scheme.
- Some more recent barcodes are rendered as **two-dimensional patterns** using dots, squares, or other symbols that can be read by specialized optical scanners, which translate a specific type of barcode into its encoded information.



Authentication via Barcodes

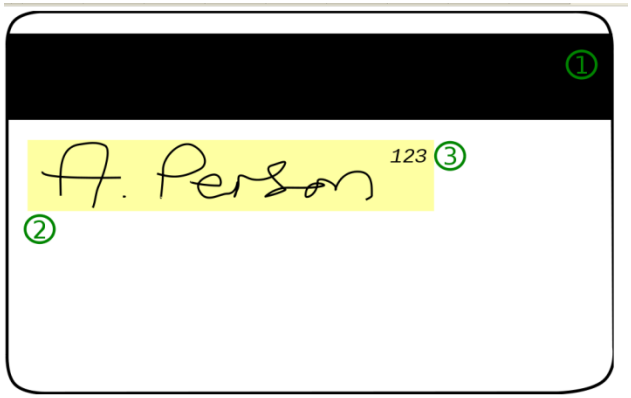
- Since 2005, the airline industry has been incorporating two-dimensional barcodes into boarding passes, which are created at flight check-in and scanned before boarding.
- In most cases, the barcode is encoded with an **internal unique identifier** that allows airport security to look up the corresponding passenger's record with that airline.
- Staff then verifies that the boarding pass was in fact purchased in that person's name (using the airline's database), and that the person can provide photo identification.
- In most other applications, however, barcodes provide convenience but not security. Since barcodes are simply images, they are extremely easy to duplicate.

Two-dimensional barcode

Public domain image from http://commons.wikimedia.org/wiki/File:Bpass.jpg

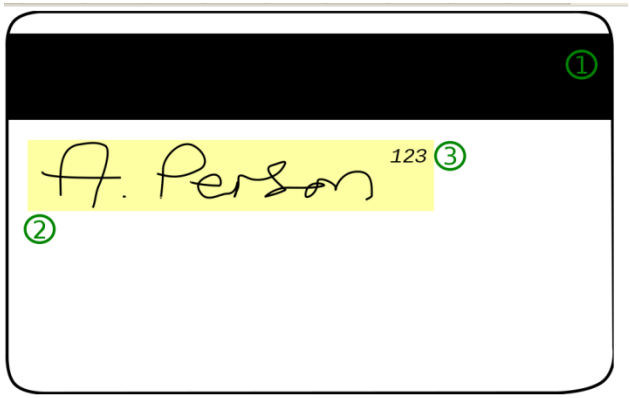
Magnetic Stripe Cards

- Plastic card with a magnetic stripe containing personalized information about the card holder.
- The first track of a magnetic stripe card contains the cardholder's full name in addition to an account number, format information, and other data.
- The second track may contain the account number, expiration date, information about the issuing bank, data specifying the exact format of the track, and other discretionary data.



Magnetic Stripe Card Security

- One vulnerability of the magnetic stripe medium is that it is easy to read and reproduce.
- Magnetic stripe readers can be purchased at relatively low cost, allowing attackers to read information off cards.
- When coupled with a magnetic stripe writer, which is only a little more expensive, an attacker can easily clone existing cards.
- So, many uses require card holders to enter a PIN to use their cards (e.g., as in ATM and debit cards in the U.S.).



Smart Cards

- **Smart cards** incorporate an integrated circuit, optionally with an on-board microprocessor, which microprocessor features reading and writing capabilities, allowing the data on the card to be both accessed and altered.
- Smart card technology can provide secure authentication mechanisms that protect the information of the owner and are extremely difficult to duplicate.

Circuit interface

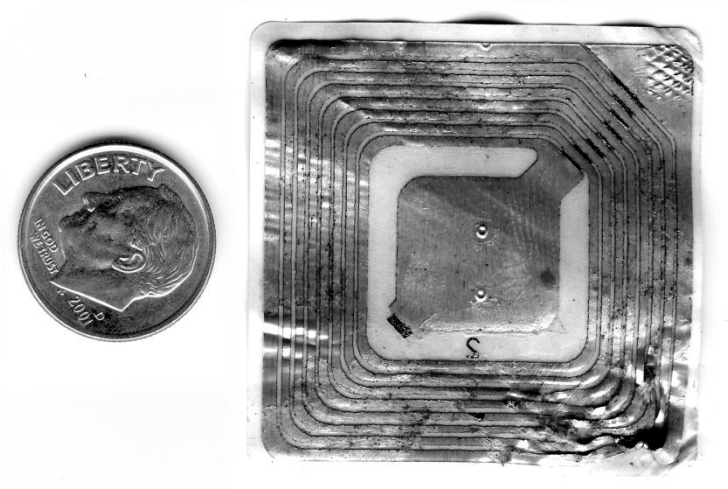
Public domain image from http://en.wikipedia.org/wiki/File:Carte_vitale_anonyme.jpg

Smart Card Authentication

- They are commonly employed by large companies and organizations as a means of strong authentication using cryptography.
- Smart cards may also be used as a sort of “electronic wallet,” containing funds that can be used for a variety of services, including parking fees, public transport, and other small retail transactions.

RFIDs

- **Radio frequency identification, or RFID,** is a rapidly emerging technology that relies on small transponders to transmit identification information via radio waves.
- RFID chips feature an integrated circuit for storing information, and a coiled antenna to transmit and receive a radio signal.



RFID Technology

- RFID tags must be used in conjunction with a separate reader or writer.
- While some RFID tags require a battery, many are passive and do not.
- The effective range of RFID varies from a few centimeters to several meters, but in most cases, since data is transmitted via radio waves, it is not necessary for a tag to be in the line of sight of the reader.

RFID Technology

- This technology is being deployed in a wide variety of applications.
- Many vendors are incorporating RFID for consumer-product tracking.
  - Car key fobs.
  - Electronic toll transponders.