Lecture 7
User Authentication

Dr. Alshaimaa Abo-alian
A_alian@cis.asu.edu.eg

Lecture Outline

- What is user authentication?
- User authentication factors
- Password based authentication
    - Password Vulnerabilities
    - How to store passwords
- Token-based Authentication
- Biometric Authentication

What is User Authentication?

- Sometimes called Entity Authentication or Identification

- The process whereby one party (**verifier**) is assured of the identity of a second party (**prover**) involved in a protocol.

- It facilitates access control for systems by checking to see if a user's credentials match the credentials in a database of authorized users.

What is User Authentication?

- **User authentication is distinct from message authentication:**

  - Message authentication is a process of verifying the **content of a received message** has not been altered
  - User authentication is a process of verifying the identity claimed by **an entity**.

  - In user authentication, the prover/claimant is active at the time of verification.
  - Message authentication provides no timeliness guarantees

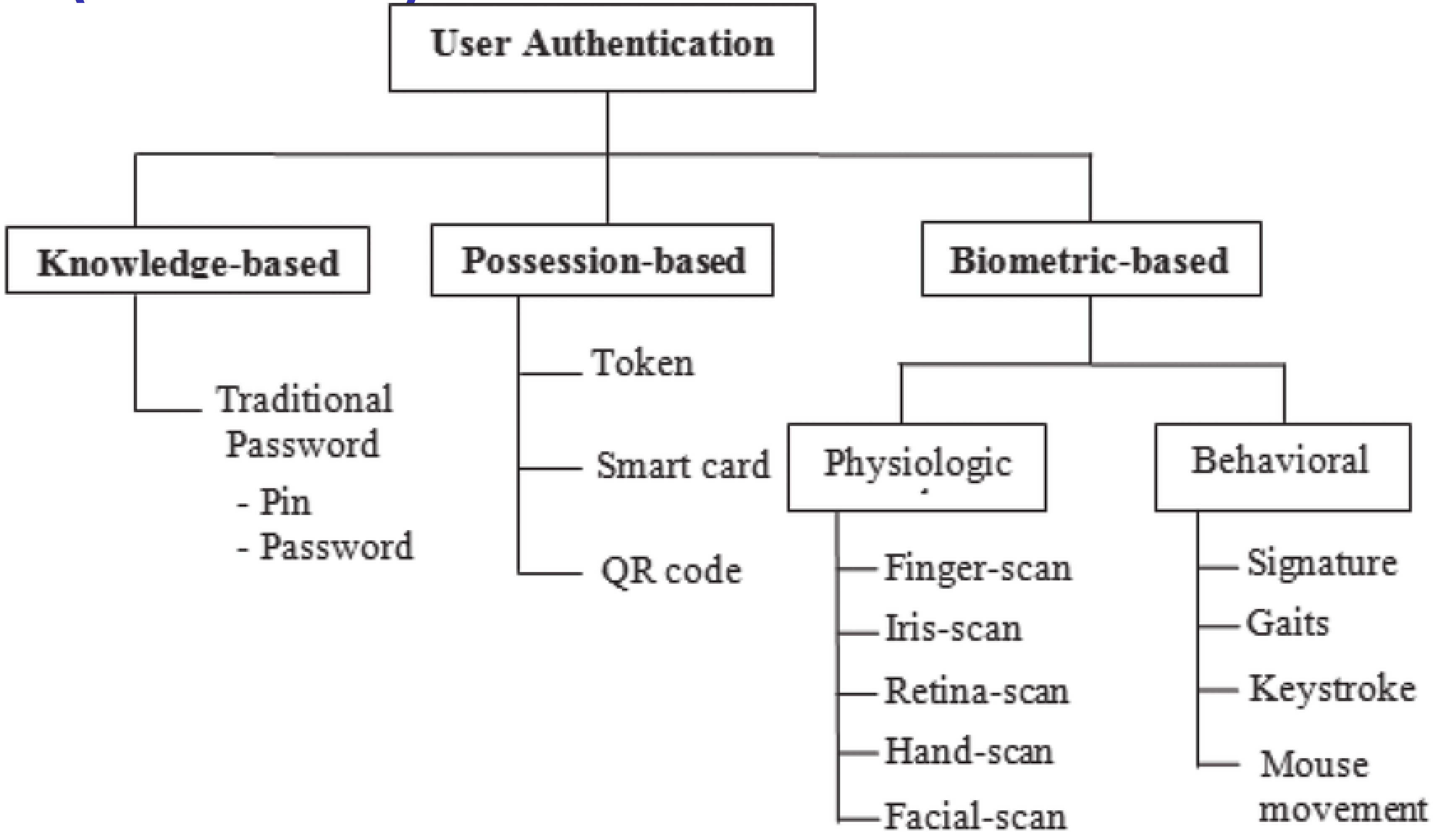
Two Steps of Authentication

1. Identification step: presenting an identifier to the security system
   - ➢Example: user ID
   - ➢Generally unique but not secret

2. Verification step: presenting or generating authentication information that acts as evidence to prove the binding between the attribute and that for which it is claimed.
   - ➢Example: password, PIN, biometric information
   - ➢Often secret or cannot be generated by others

User authentication is primary line of defense in computer security; other security controls rely on it

User Authentication Means (Factors)



User Authentication Means (Factors)

There are 3 general means, or **authentication factors**, of authenticating a user's identity:

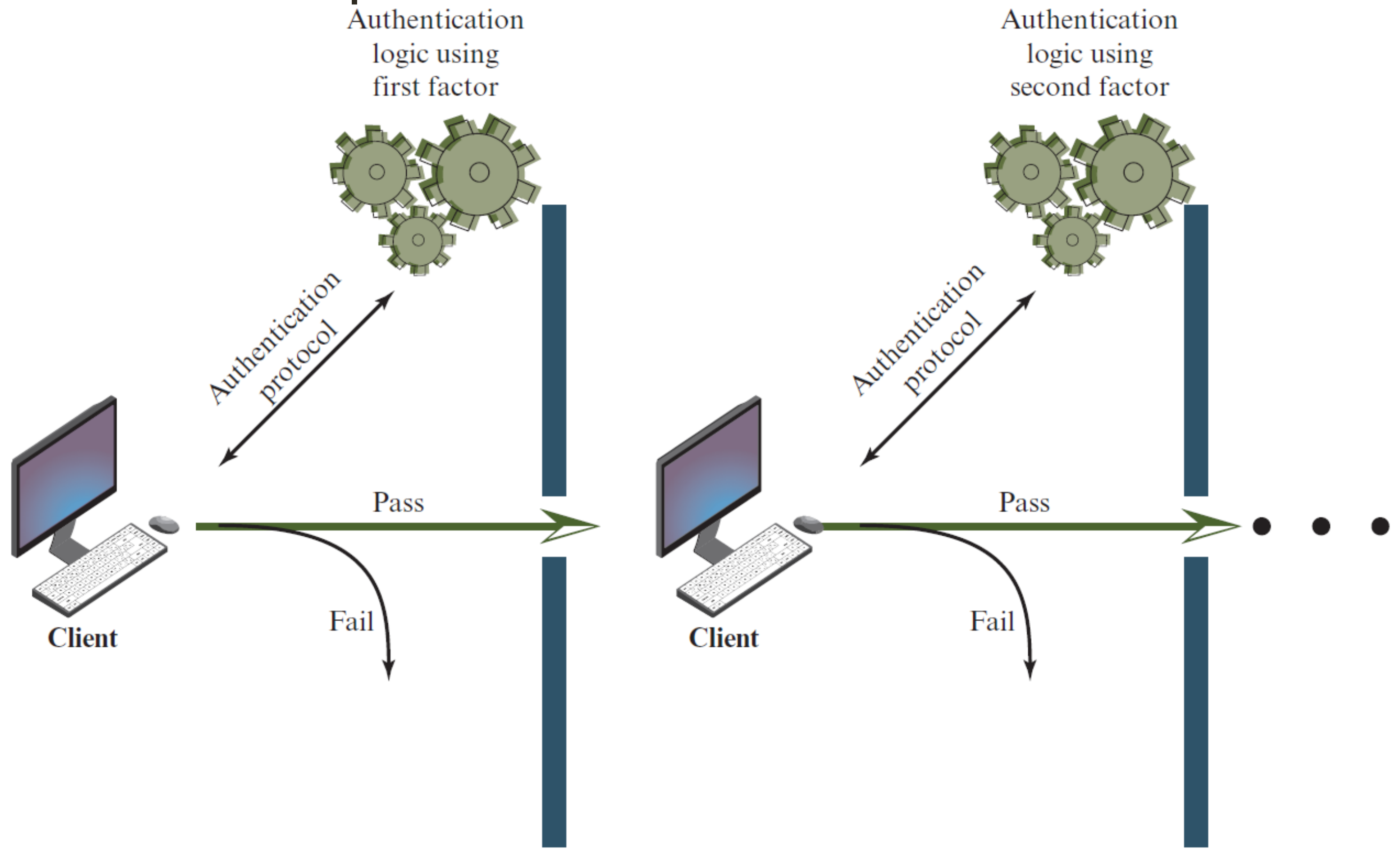
1. **Knowledge factor (something the individual knows):** Such as passwords, passphrases, personal identification numbers (PINs), etc.
2. **Possession factor (something the individual possesses):** physical hardware possessed by the authorized user to connect to the client computer or portal. Such as smart cards and USB tokens
3. **Inherence factor (something the individual is or does):** characteristics, called **biometrics**, that are unique or almost unique to the individual. These include:
   a) **Static physiological biometrics:** such as fingerprint, retina, and face
   b) **Dynamic behavioral biometrics:** such as voice, handwriting, and keystroke.

User Authentication Factors

Factor	Examples	Properties
Knowledge	User ID Password PIN	Can be shared Many passwords easy to guess Can be forgotten
Possession	Smart Card QR Code RFID	Can be shared Can be duplicated (cloned) Can be lost or stolen
Inherence	Fingerprint Face Iris Voice	Not possible to share False positives and false Negatives possible Forging is difficult

Multifactor Authentication

- Refers to the use of two or more authentication factors. such as a PIN plus a hardware token


Authentication logic using first factor
Authentication protocol
Client
Pass
Fail
Authentication logic using second factor
Authentication protocol
Client
Pass
Fail

Password-based Authentication

- Most commonly used
- User provides username (ID) and password
- System compares password with the one stored for that specified ID.
- The user ID:
  - ✓Determines that the user is authorized to access the system
  - ✓Determines the user's privileges
  - ✓Is used in discretionary access control

Password Threats

- Offline dictionary attack
- Password guessing against single user
- Computer hijacking
- Electronic monitoring
- Specific account attack
- Popular password attack
- Exploiting user mistakes
- Exploiting multiple password use

Password Threats

1. Offline Dictionary Attack Attacker obtains access to ID/password (hash) database; use dictionary to find passwords

➢ Countermeasures:

1. Strong access control to prevent unauthorized access to the password file
2. Intrusion detection measures to identify a compromise
3. Rapid reissuance of passwords if compromised
4. **Strong hashes and salts**

Password Threats

2. Specific Account Attack: Attacker submits password guesses on specific account

➢ Countermeasures:

lock account after a number of failed attempts (Typical practice is no more than five access attempts)

3. Popular Password Attack Try popular password with many IDs

➢ Countermeasures:

1. Control password selection
2. Block computers that make multiple attempts for different accounts

Password Threats

4. Password Guessing Against Single User Gain knowledge about user and use that to guess password

➢ Countermeasures: control password selection such as

- √ Minimum length of the password
- √ Character set
- √ Prohibition against using well-known user identifiers
- √ Length of time before the password must be changed.

5. Computer Hijacking Attackers gains access to computer that user currently logged in to

➢ Countermeasures: auto-logout

Password Threats

6. Exploiting User Mistakes Users write down password, share with friends, use pre-configured passwords

- ➢ Countermeasures: user training, multifactor authentication

7. Exploiting Multiple Password Use Passwords re-used across different systems/accounts, make easier for attacker to access resources once one password discovered

- ➢ Countermeasure: control selection of passwords on multiple account/devices

Password Threats

8. Electronic Monitoring (Eavesdropping) Attacker intercepts passwords sent across network

- ➢ Countermeasure: Simple encryption will not fix this problem because the encrypted password can be observed and reused by the attacker

How Should Passwords Be Stored?

Storing Passwords in the Clear ID; P

Insider attack: normal user reads the database and learns other users' passwords

- ➢ Countermeasure: access control on password database

Outsider attack: attacker gains unauthorized access to database and learns all passwords

- ➢ Countermeasure: do not store passwords in the clear

How Should Passwords Be Stored?

Encrypting the Passwords ID; E(K; P)

- Encrypted passwords are stored
- When user submits password, it is encrypted and compared to the stored value

➢ Drawback: Secret key, K, must be stored; if attacker can read database, then likely they can also read K

How Should Passwords Be Stored?

Hashing the Passwords ID;H(P)

- Hashes of passwords are stored
- When user submits password, it is hashed and compared to the stored value
- Practical properties of hash functions:
  - Variable sized input; produce a fixed length, small output
  - No collisions
  - One-way function

→ If attacker gains database, practically impossible to take a hash value and directly determine the original password

Brute Force Attack on Hashed Passwords

- **Aim**: given one (or more) target hash value, find the original password
- Start with large set of possible passwords (e.g., from dictionary, all possible n-character combinations)
- Calculate hash of possible password, compare with target hash
  - if match, original password is found
  - else, try next possible password

→Attack duration depends on size of possible password set

Pre-calculated Hashes And Rainbow Tables

- How to speed up brute force attack? Use hash values calculated by someone else
- Possible passwords and corresponding hashes stored in database
- Attacker performs lookup on database for target hash
- How big is such a database of pre-calculated hashes?
  - In raw form, generally too big to be practical (100’s or 1000's of TB)
  - Using specialized data structures (e.g. Rainbow tables), can obtain manageable size, e.g. 1 TB

Pre-calculated Hashes And Rainbow Tables

- Tradeoff: reduce search time, but increase storage space
- Countermeasures:
  - ✓ Longer passwords
  - ✓ Slower hash algorithms
  - ✓ Salting the password before hashing

Salting Passwords

ID; Salt; H(P||Salt)

- When ID and password initially created, generate random s-bit value (salt), concatenate with password and then hash
- When user submits password, salt from password database is concatenated, hashed and compared
- If attacker gains database, they know the salt; same effort to find password as brute force attack
- BUT pre-calculated values (e.g. Rainbow tables) are no longer feasible
  - ➢ Space required increased by factor of 2^s

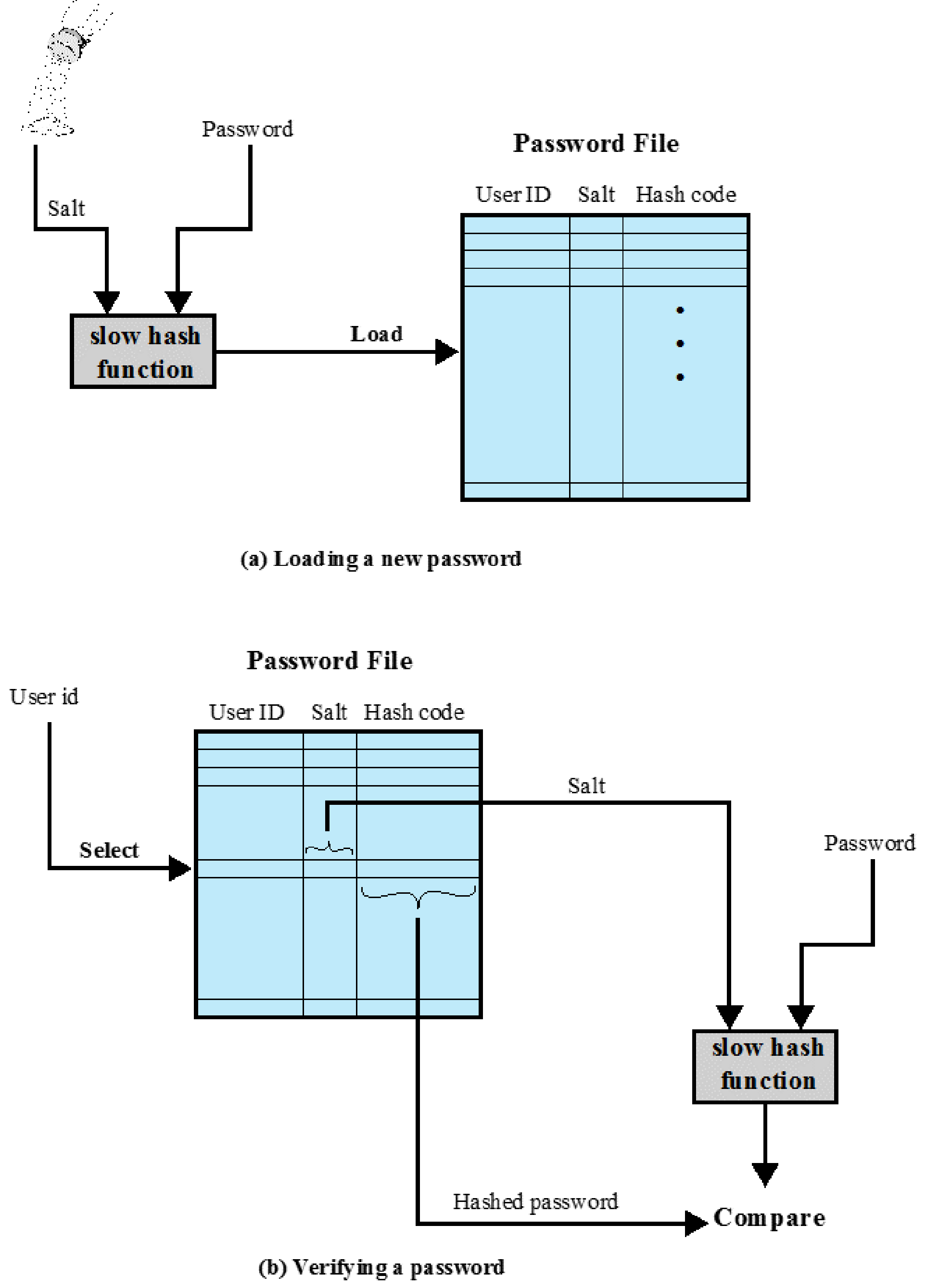


Figure 3.3 UNIX Password Scheme

Password Cracking

Dictionary attacks

- Develop a large dictionary of possible passwords and try each against the password file
- Each password must be hashed using each salt value and then compared to stored hash values

Rainbow table attacks

- Pre-compute tables of hash values for all salts
- A mammoth table of hash values
- Can be countered by using a sufficiently large salt value and a sufficiently large hash length

Password crackers exploit the fact that people choose easily guessable passwords

- Shorter password lengths are also easier to crack

John the Ripper

- Open-source password cracker first developed in in 1996
- Uses a combination of brute-force and dictionary techniques

Password Selection Strategies

1. User education Ensure users are aware of importance of hard-to-guess passwords; advise users on strategies for selecting passwords
2. Computer-generated passwords Generate random or pronounceable passwords (but poorly accepted by users)
3. Proactive password checking User is allowed to select their own password, however the system checks to see if the password is allowable, and if not, rejects it

➔ Should balance between user acceptability and strength.

Proactive Password Checking

Rule enforcement

- Specific rules that passwords must adhere to

Password checker

- Compile a large dictionary of passwords not to use

Bloom filter

- Used to build a table based on hash values
- Check desired password against this table

Token-based Authentication

- Objects that a user possesses for purpose of user authentication are called tokens
- Examples:

1. Memory Cards
2. Smart Tokens
    a. Smart Cards
    b. Electronic Identity Cards (eID)

Memory Cards

- Can store but do not process data
- The most common is the magnetic stripe card
- Can include an internal electronic memory
- Can be used alone for physical access, e.g. hotel room,
- Provides greater security when combined with a password or PIN, e.g. ATM
- Drawbacks include
    - Requires a special reader
    - Loss of token
    - User dissatisfaction

Smart Tokens

- **Physical characteristics:**
  - Include an embedded microprocessor
  - A smart token that looks like a bank card
  - Can look like calculators, keys, small portable objects
- **User Interface:** manual interfaces include a keypad and display for interaction
- **Electronic interfaces:** communicate with a compatible reader/writer
  - **Contact interface:** must be inserted into a smart card reader. Transmission of commands/ data takes place over these physical contact points.
  - **Contactless interfaces:** requires only close proximity to a reader. Both the reader and the card have an antenna

Smart Tokens

- **Authentication protocol:**

1. Static: Similar to a memory token.
2. Dynamic password generator: The token generates a unique password periodically (e.g., every minute).
3. Challenge-response: the computer system generates a challenge, such as a random string of numbers. The smart token generates a response based on the challenge.

Smart Cards

- Most important category of smart token
  - Has the appearance of a credit card
  - May use any of the smart token protocols
- Contain an entire microprocessor
  - Processor
  - Memory
  - I/O ports or embedded antenna
- Typically include three types of memory:
  - **Read-only memory (ROM):** Stores data that does not change during the card's life
  - **Electrically erasable programmable ROM (EEPROM)**: Holds application data and programs
  - **Random access memory (RAM):** Holds temporary data generated when applications are executed

Electronic Identity Cards (eID)

- Use of a smart card as a national identity card for citizens
- Can serve the same purposes as other national ID cards for access to government and commercial services
- Can provide stronger proof of identity and can be used in a wider variety of applications

Biometric Authentication

- Attempts to authenticate an individual based on unique physical/behavioral characteristics
- Based on pattern recognition
- Technically complex and expensive when compared to passwords and tokens
- Characteristics used include:
  - ✓ Facial characteristics
  - ✓ Fingerprints
  - ✓ Hand geometry
  - ✓ Retinal pattern
  - ✓ iris
  - ✓ signature
  - ✓ Voice
  - ✓ Keystrokes

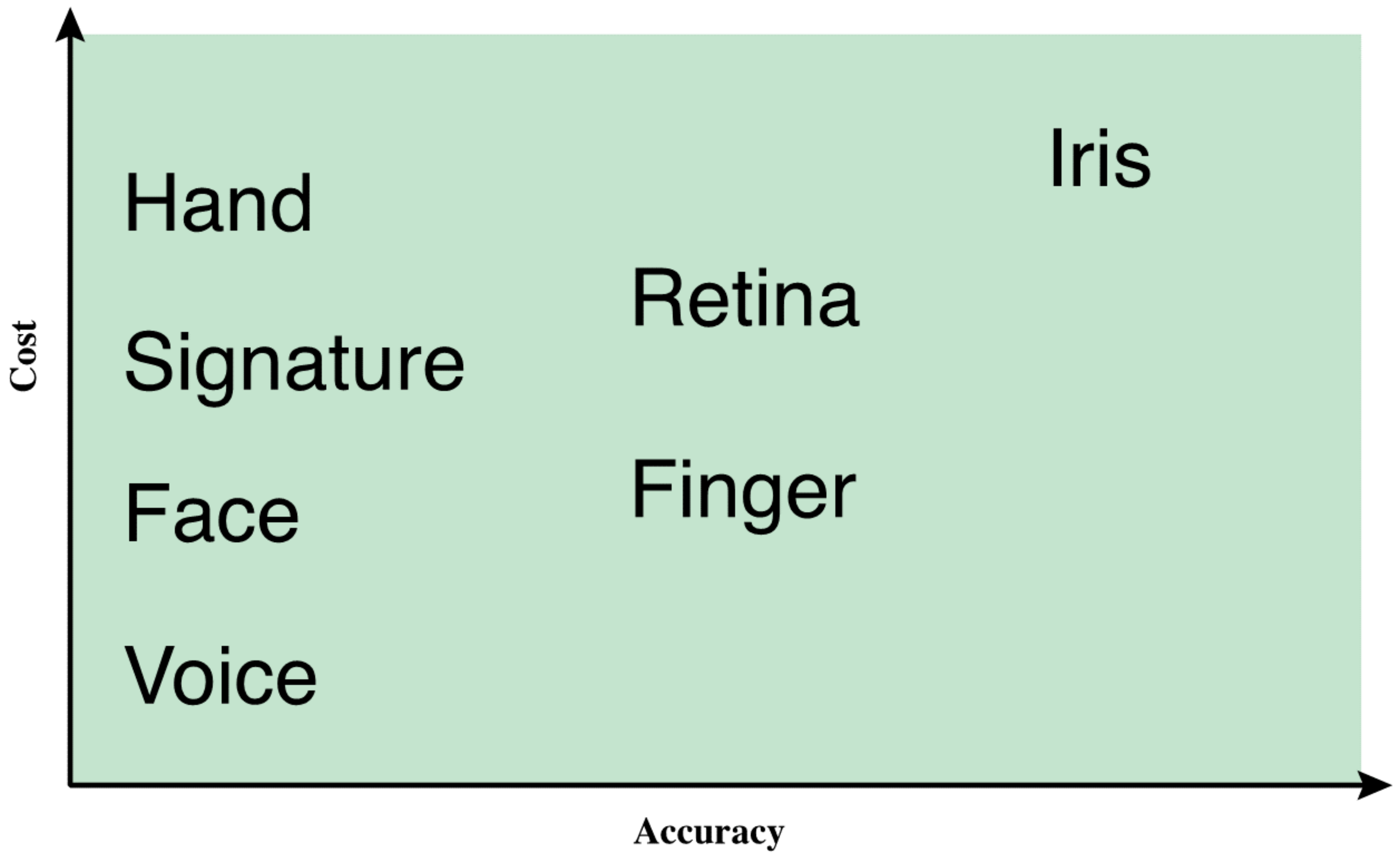


Figure 3.8 Cost Versus Accuracy of Various Biometric Characteristics in User Authentication Schemes.

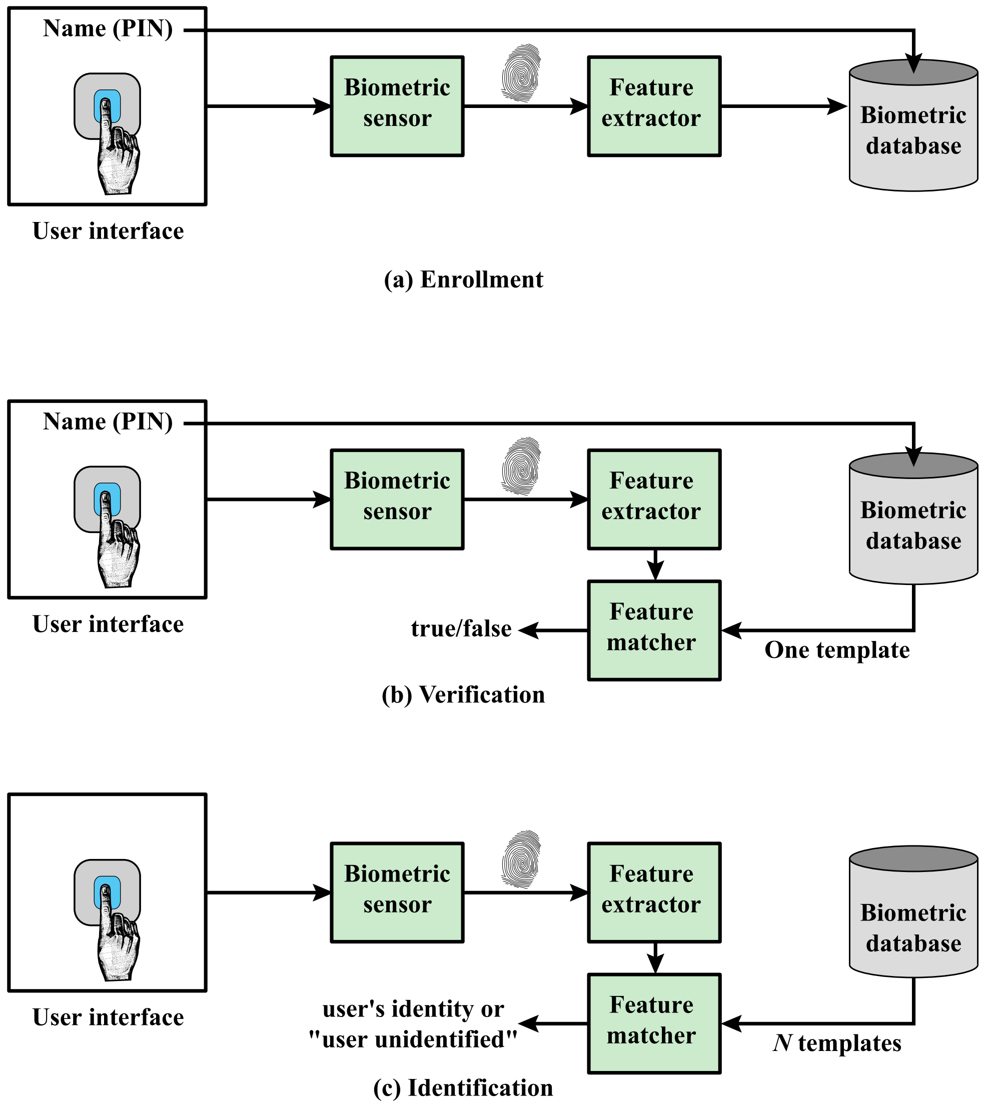


Figure 3.9 A Generic Biometric System. Enrollment creates an association between a user and the user's biometric characteristics. Depending on the application, user authentication either involves verifying that a claimed user is the actual user or identifying an unknown user.

Biometric Accuracy

- When the user is to be authenticated, the system compares the stored template (samples) to the presented template.
- The system uses a matching score that quantifies the similarity between the input and the stored template.
- False match rate (**false positive**; FP): where the model incorrectly predicts the positive class (i.e., incorrectly authenticate the user).
- False nonmatch rate (**false negative**; FN): where the model incorrectly predicts the negative class (i.e., incorrectly reject the user)

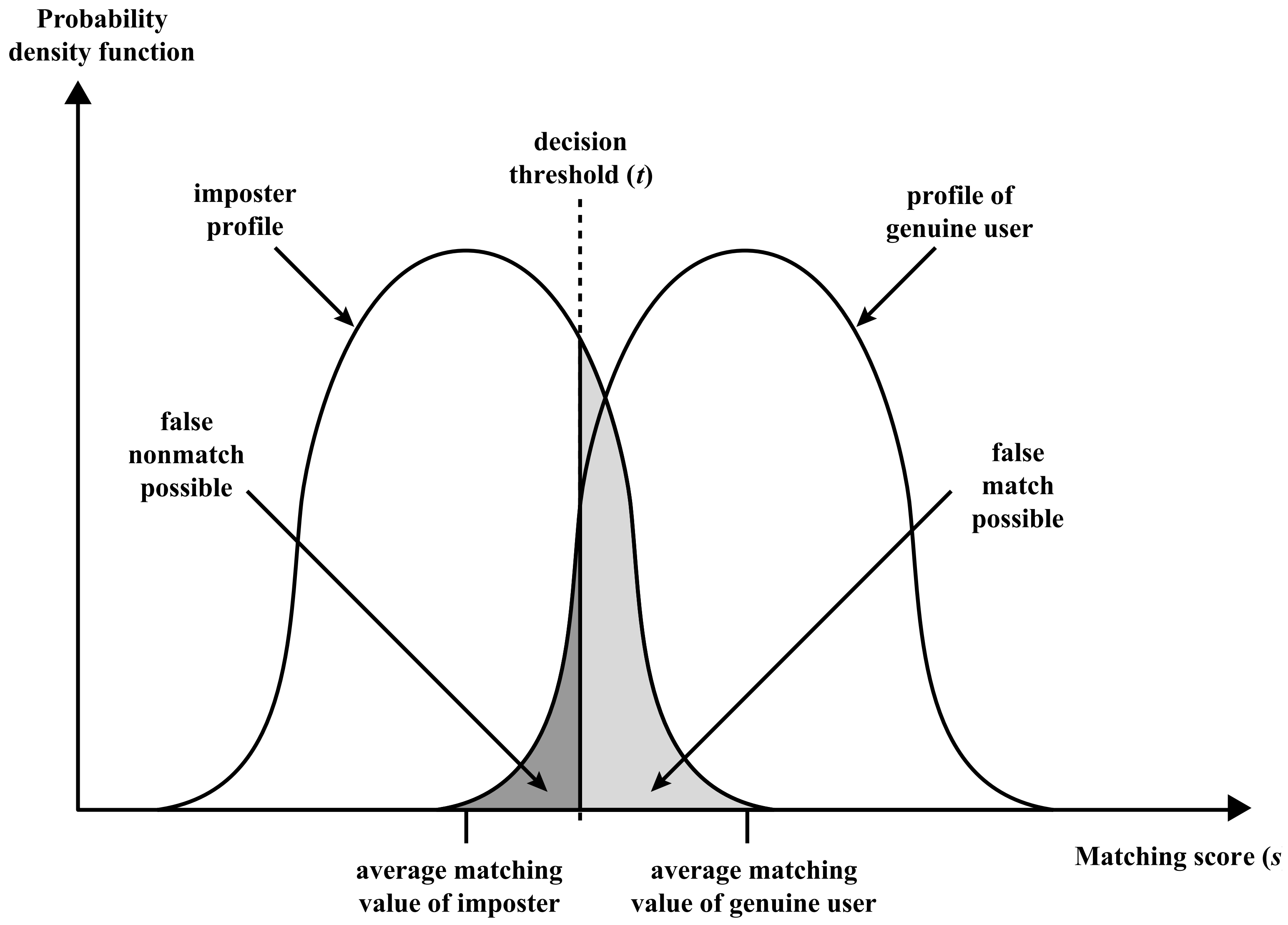


Figure 3.10 Profiles of a Biometric Characteristic of an Imposter and an Authorized Users In this depiction, the comparison between presented feature and a reference feature is reduced to a single numeric value. If the input value (*s*) is greater than a preassigned threshold (*t*), a match is declared.

LECTURE REFERENCE

“Computer Security: Principles and Practice”, 4/e, by William Stallings and Lawrie Brown

Chapter 3 “User Authentication”.

Thank you