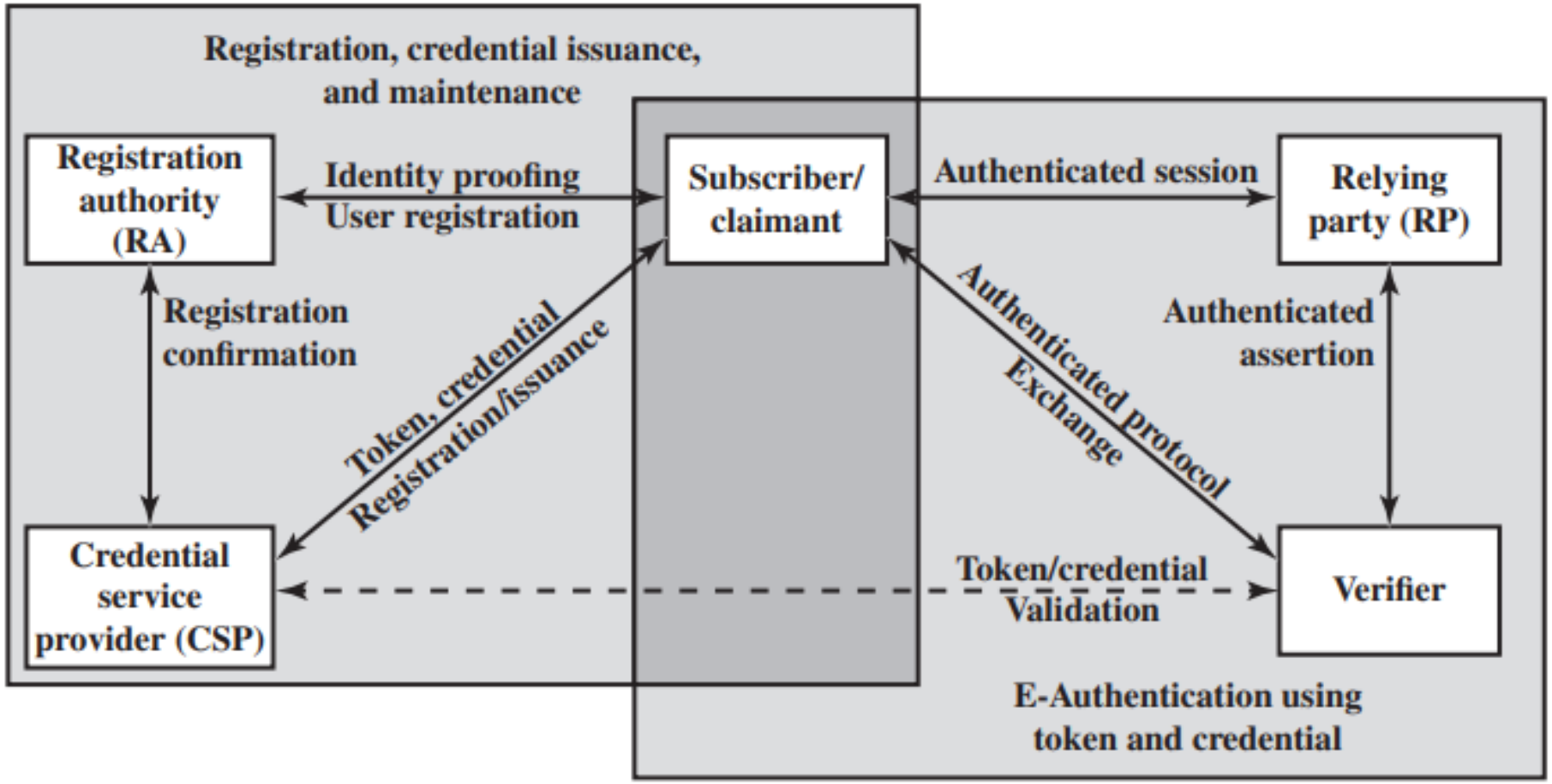
User Authentication

Model for Electronic User Authentication



Model for Electronic User Authentication

- An applicant applies to a registration authority (RA) to become a subscriber of a credential service provider (CSP).
- CSP issues some sort of electronic credential to the subscriber.
- The credential is a data structure that authoritatively binds an identity and additional attributes to a token possessed by a subscriber, and can be verified when presented to the verifier in an authentication transaction.

Model for Electronic User Authentication

- Once a user is registered as a subscriber, the actual authentication process can take place between the subscriber and one or more systems.
- The party to be authenticated is called a claimant and the party verifying that identity is called a verifier.
- The verifier passes on an assertion about the identity of the subscriber to the relying party (RP).

Means of Authentication

- Something the individual knows: Examples includes a password, a personal identification number (PIN), or answers to a prearranged set of questions.
- Something the individual possesses: Examples include electronic keycards, smart cards, and physical keys. This type of authenticator is referred to as a token.
- Something the individual is (static biometrics): Examples include recognition by fingerprint, retina, and face.
- Something the individual does (dynamic biometrics): Examples include recognition by voice pattern, handwriting characteristics, and typing rhythm.

Which of these is the most common?

Password Based Authentication

- A widely used line of defense against intruders is the password system.
- The password serves to authenticate the ID of the individual logging on to the system.
- ID provides security in the following ways:
  - The ID determines whether the user is authorized to gain access to a system
  - The ID determines the privileges accorded to the user.
  - The ID is used in what is referred to as discretionary access control.

The Vulnerability of Passwords

- Offline dictionary attack.
- Specific account attack
- Popular password attack
- Password guessing against single user
- Workstation hijacking
- Exploiting user mistakes
- Electronic monitoring

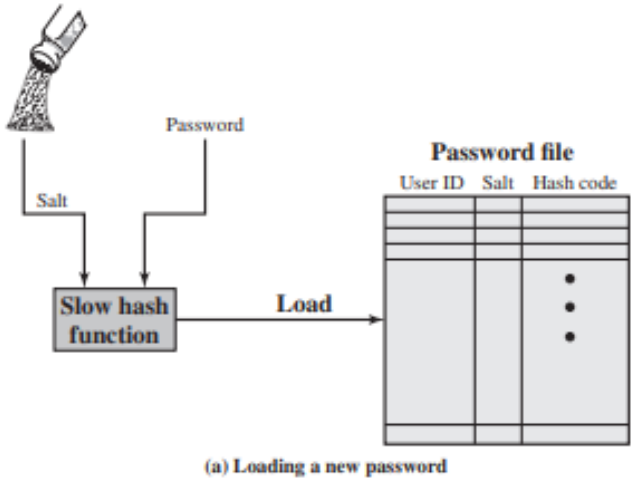
Persistent popularity of passwords

- Techniques that utilize client-side hardware, such as fingerprint scanners and smart card readers, require the implementation of the appropriate user authentication software to exploit this hardware on both the client and server systems.
- Physical tokens, such as smart cards, are expensive and/or inconvenient to carry around, especially if multiple tokens are needed.
- Schemes that rely on a single sign-on to multiple services, using one of the non-password techniques create a single point of security risk.
- Automated password managers that relieve users of the burden of knowing and entering passwords have poor support for roaming and synchronization across multiple client platforms, and their usability had not be adequately researched.

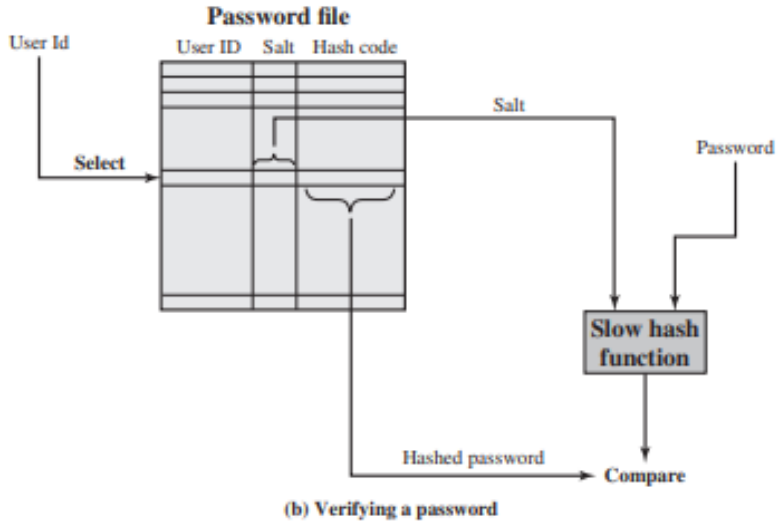
Defense against password attacks

- Upto 5 login attempts
- Automatic logout
- Password policies (user education)
- Reactive password checking
- Separate passwords from user ids
- Salting the hash
- Resetting the passwords
- Intentionally slow down authentication

Use of Hashed Passwords



(a) Loading a new password



(b) Verifying a password

Unix Implementations

1. **User Password and Salt:**
   - User passwords are 8 digits long, resulting in a 56-bit value.
   - A 12-bit salt value is generated, typically based on the account creation time.
2. **Hash Function and Iterations:**
   - The hashing algorithm used is based on DES (Data Encryption Standard) and is run 25 times.
   - Each iteration involves rehashing the result of the previous iteration.
3. **Resulting 64-Bit Value:**
   - After running the hash function 25 times, the final hash value is 64 bits long.
4. **Conversion to 11-Character Sequence:**
   - The resulting 64-bit hash value is converted to an 11-character sequence.

Password Cracker

- Password crackers are software tools or programs designed to attempt to break into user accounts by guessing or cracking passwords.
- These tools use various techniques to try different combinations of characters, words, and patterns to find the correct password or its corresponding hash value.
- Password crackers are often used by security professionals and ethical hackers to assess the strength of passwords and identify potential vulnerabilities in password security.

Password Cracker

1. **Brute Force Attack:** Brute force password crackers systematically try all possible combinations of characters, starting from the shortest passwords and gradually increasing the length until the correct password is found. Brute force attacks are time-consuming but can be effective for weak passwords.
2. **Dictionary Attack:** Dictionary password crackers use a precompiled list of commonly used passwords, words from dictionaries, and common character combinations to speed up the password guessing process. This approach is more efficient than brute force for finding weak passwords.
3. **Rainbow Table Attack:** Rainbow table password crackers use precomputed tables of password hashes and their corresponding plaintext values. They match the stored hashed passwords with entries in the rainbow table to find the original passwords. Salted hashes make rainbow table attacks less effective.

Password Cracker

4. **Hybrid Attack:** Hybrid password crackers combine elements of both dictionary attacks and brute force attacks to create more targeted and efficient password guessing strategies.
5. **Rule-Based Attack:** Rule-based password crackers apply specific rules or patterns to generate passwords based on common patterns, keyboard layouts, or known password patterns. This approach speeds up the password guessing process for common password structures.
6. **Mask Attack:** Mask attacks allow password crackers to specify a pattern or mask for the password. For example, if parts of the password are known or can be guessed, a mask attack can be used to test only those parts, reducing the search space.

Password Selection Strategies

- Four basic techniques are in use:
  - ✓User education
  - ✓Computer-generated passwords
  - ✓Reactive password checking
  - ✓Complex password policy or proactive password checker

Approaches to proactive password checking

- Rule Enforcement
  - ✓For example, the following rules could be enforced:
    - All passwords must be at least eight characters long.
    - In the first eight characters, the passwords must include at least one each of uppercase, lowercase, numeric digits, and punctuation marks.
- Password Checker
  - ✓When a user selects a password, the system checks to make sure that it is not on the disapproved list.
  - ✓There are two problems with this approach:
    - Space: The dictionary must be very large to be effective.
    - Time: The time required to search a large dictionary may itself be large. In addition, to check for likely permutations of dictionary words, either those words must be included in the dictionary, making it truly huge, or each search must also involve considerable processing.

Approaches to proactive password checking

- Bloom Filter
  - ✓A Bloom filter of order k consists of a set of k-independent hash functions H1(x), H2(x), c, Hk(x), where each function maps a password into a hash value in the range 0 to N – 1. That is

$$H_i(X_j) = y \qquad 1 \le i \le k; \qquad 1 \le j \le D; \qquad 0 \le y \le N - 1$$

where

X_j = jth word in password dictionary

D = number of words in password dictionary

  - ✓A hash table of N bits is defined, with all bits initially set to 0.
  - ✓For each password, its k hash values are calculated, and the corresponding bits in the hash table are set to 1. Thus, if Hi (Xj) = 67 for some (i, j), then the sixty-seventh bit of the hash table is set to 1; if the bit already has the value 1, it remains at 1.

Token Based Authentication

- Objects that a user possesses for the purpose of user authentication are called tokens.

Table 3.2 **Types of Cards Used as Tokens**

Card Type	Defining Feature	Example
Embossed	Raised characters only, on front	Old credit card
Magnetic stripe	Magnetic bar on back, characters on front	Bank card
Memory	Electronic memory inside	Prepaid phone card
Smart Contact Contactless	Electronic memory and processor inside Electrical contacts exposed on surface Radio antenna embedded inside	Biometric ID card

Drawbacks

- **Requires special reader:** This increases the cost of using the token and creates the requirement to maintain the security of the reader's hardware and software.
- **Token loss:** A lost token temporarily prevents its owner from gaining system access. Thus there is an administrative cost in replacing the lost token. In addition, if the token is found, stolen, or forged, then an adversary now need only determine the PIN to gain unauthorized access.
- **User dissatisfaction:** Although users may have no difficulty in accepting the use of a memory card for ATM access, its use for computer access may be deemed inconvenient.

Smart Cards

- Physical characteristics: Smart tokens include an embedded microprocessor. A smart token that looks like a bank card is called a smart card.
- User interface: Manual interfaces include a keypad and display for human/token interaction.
- Electronic interface: A smart card or other token requires an electronic interface to communicate with a compatible reader/writer.
- A card may have one or both of the following types of interface:
  - Contact: A contact smart card must be inserted into a smart card reader with a direct connection to a conductive contact plate on the surface of the card (typically gold-plated).
  - Contactless: A contactless card requires only close proximity to a reader. Both the reader and the card have an antenna, and the two communicate using radio frequencies.

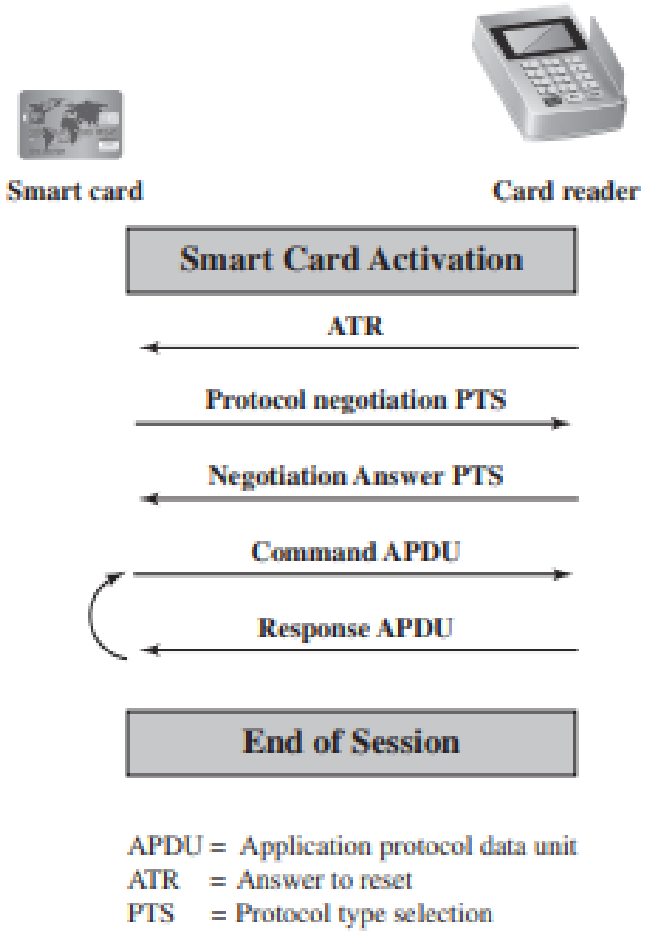
Smart Cards

- Authentication protocol: The purpose of a smart token is to provide a means for user authentication.
  - Static: With a static protocol, the user authenticates himself or herself to the token, and then the token authenticates the user to the computer.
  - Dynamic password generator: In this case, the token generates a unique password periodically (e.g., every minute). This password is then entered into the computer system for authentication, either manually by the user or electronically via the token.
  - Challenge-response: In this case, the computer system generates a challenge, such as a random string of numbers. The smart token generates a response based on the challenge.

Electronic Identity Cards

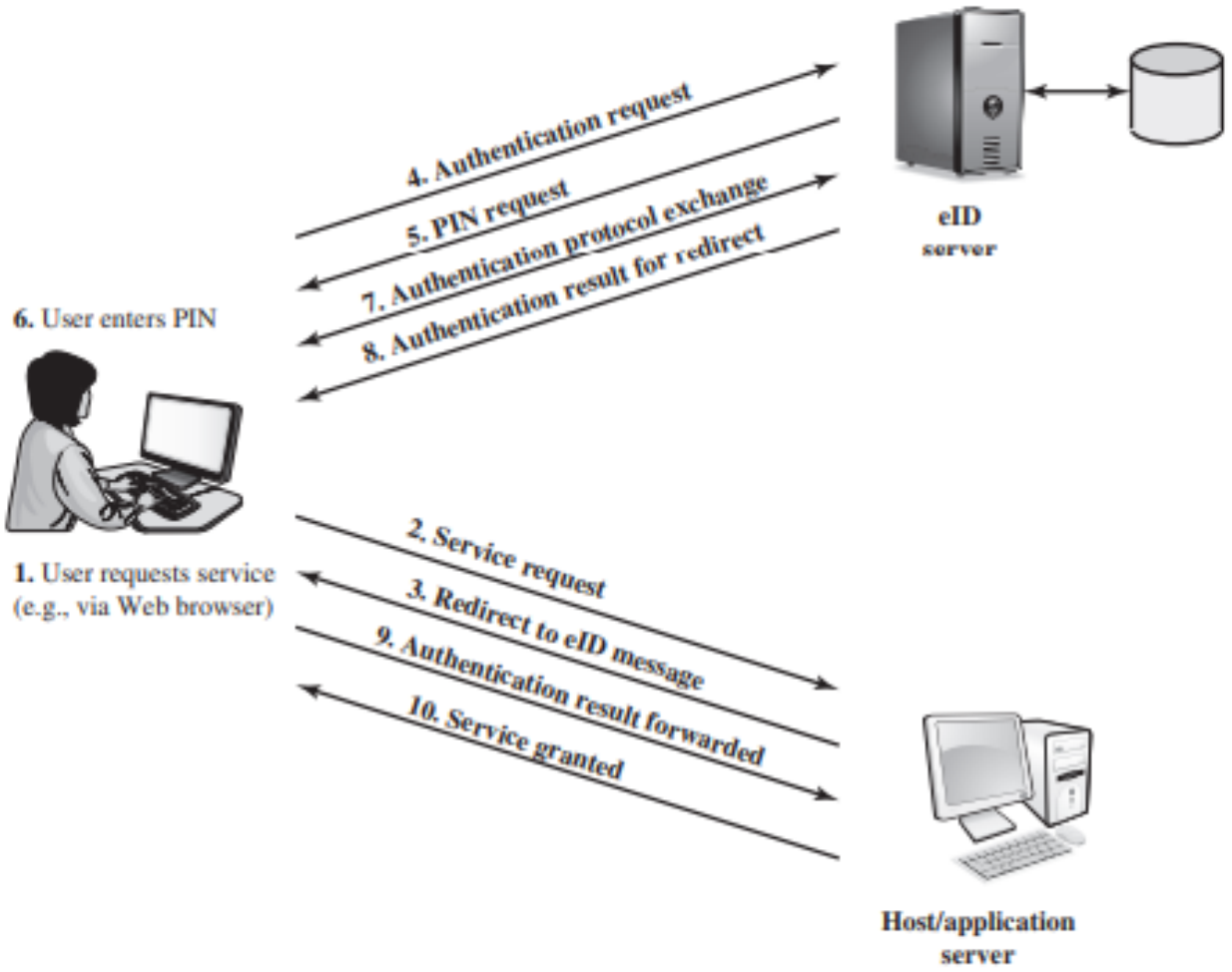
- An application of increasing importance is the use of a smart card as a national identity card for citizens.
- An eID card can provide stronger proof of identity and be used in a wider variety of applications.
- An eID card is a smart card that has been verified by the national government as valid and authentic.

Smart Card/Reader Exchange



APDU = Application protocol data unit
ATR = Answer to reset
PTS = Protocol type selection

User Authentication with eID



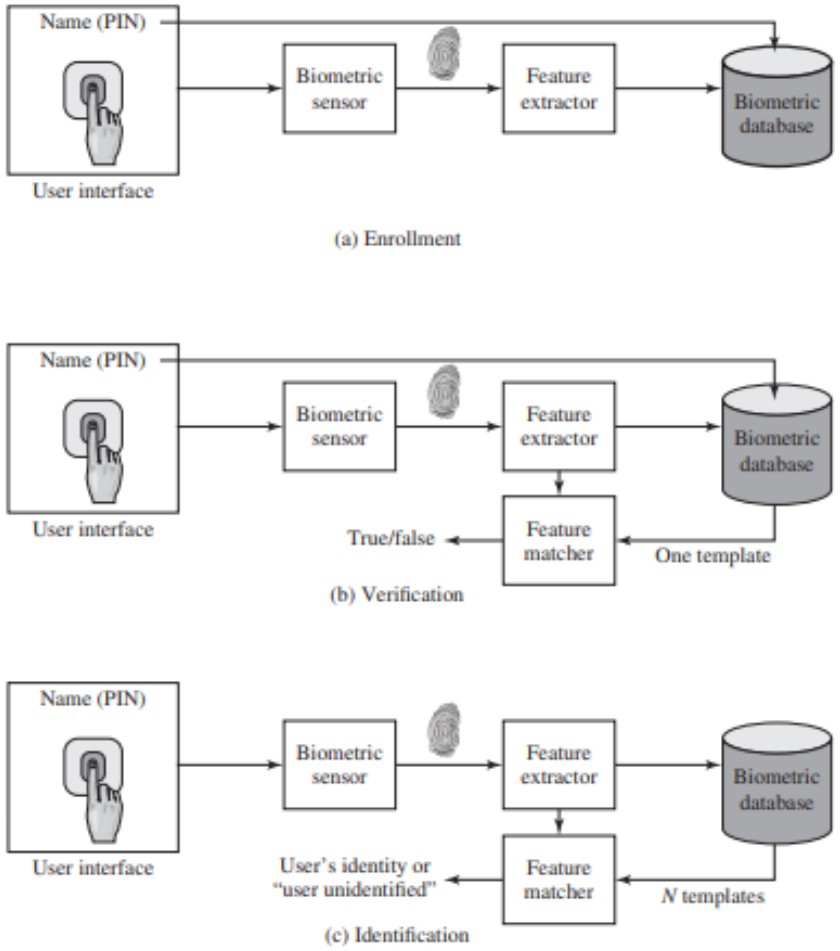
Biometric Authentication

- A biometric authentication system attempts to authenticate an individual based on his or her unique physical characteristics.
- These include static characteristics, such as fingerprints, hand geometry, facial characteristics, and retinal and iris patterns; and dynamic characteristics, such as voiceprint and signature.
- Compared to passwords and tokens, biometric authentication is both technically more complex and expensive.

Physical Characteristics Used in Biometric Applications

- Facial characteristics
- Fingerprint
- Hand geometry
- Retinal pattern
- Iris
- Signature
- Voice

A Generic Biometric System



Sample Problems

- An iris biometric system
- Security Problems for ATM machine

THE END