INFORMATION SECURITY FALL 2022

Week # 7

Lecture # 16, 17 and 18

Dr. Aqsa Aslam

CHAPTER 3

USER AUTHENTICATION

LEARNING OBJECTIVES

After studying this chapter, you should be able to:

- Discuss the four general means of authenticating a user's identity.
- Explain the mechanism by which hashed passwords are used for us authentication.
- Understand the use of the Bloom filter in password management.
- Present an overview of token-based user authentication.
- Discuss the issues involved and the approaches for remote user authentication.
- Summarize some of the key security issues for user authentication.

User Authentication

- User authentication is a fundamental security building block
- User authentication is the basis for:
  - ***Access control*** *and* ***User accountability***
- User authentication encompasses two functions
  1. ***Identification:***
     - The user identifies herself to the system by presenting a credential
       - *such as user ID.*
  2. ***Verification:***
     - The system verifies the user by the exchange of authentication information.
- User Authentication is distinct from message authentication
  - *(when communicating parties are concerned with the integrity of the exchanges messages)*

- ***Identification*** → *means by which a user provides a claimed identity to the system*
- ***User authentication*** → *means of establishing the validity of the claim*

A Model for Digital User Authentication

- *NIST SP 800-63-3 (Digital Authentication Guideline, October 2016)* ***defines digital user authentication as the process of establishing confidence in user identities that are presented electronically to an information system.***
  - *Systems can use the authenticated identity to determine if the authenticated individual is authorized to perform particular functions*
    - Such as database transactions or access to system resources.

A Model for Digital User Authentication

- The NIST SP 800-63-2 model
  - *The initial requirement for performing user authentication is that the user must be registered with the system.*
  - *An applicant applies to a **registration authority (RA)** to become a **subscriber** of a **credential service provider (CSP).***
  - ***RA** is a trusted entity*
    - That establishes and vouches for the identity of an applicant to a CSP.
  - *The **CSP** then engages in an exchange with the subscriber.*
    - The CSP issues some sort of electronic credential to the subscriber.
      - *The credential (a data structure) binds an identity to a token possessed by the subscriber*
  - *Once a user is registered as a subscriber*
    - the actual authentication process can take place between the subscriber and one or more systems
      - *that perform authentication and, subsequently, authorization*
  - *The party to be authenticated is called a **claimant***
  - *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).***
    - The **RP** can use the authenticated information provided by the verifier to make access control or authorization decisions.

A Model for Digital User Authentication

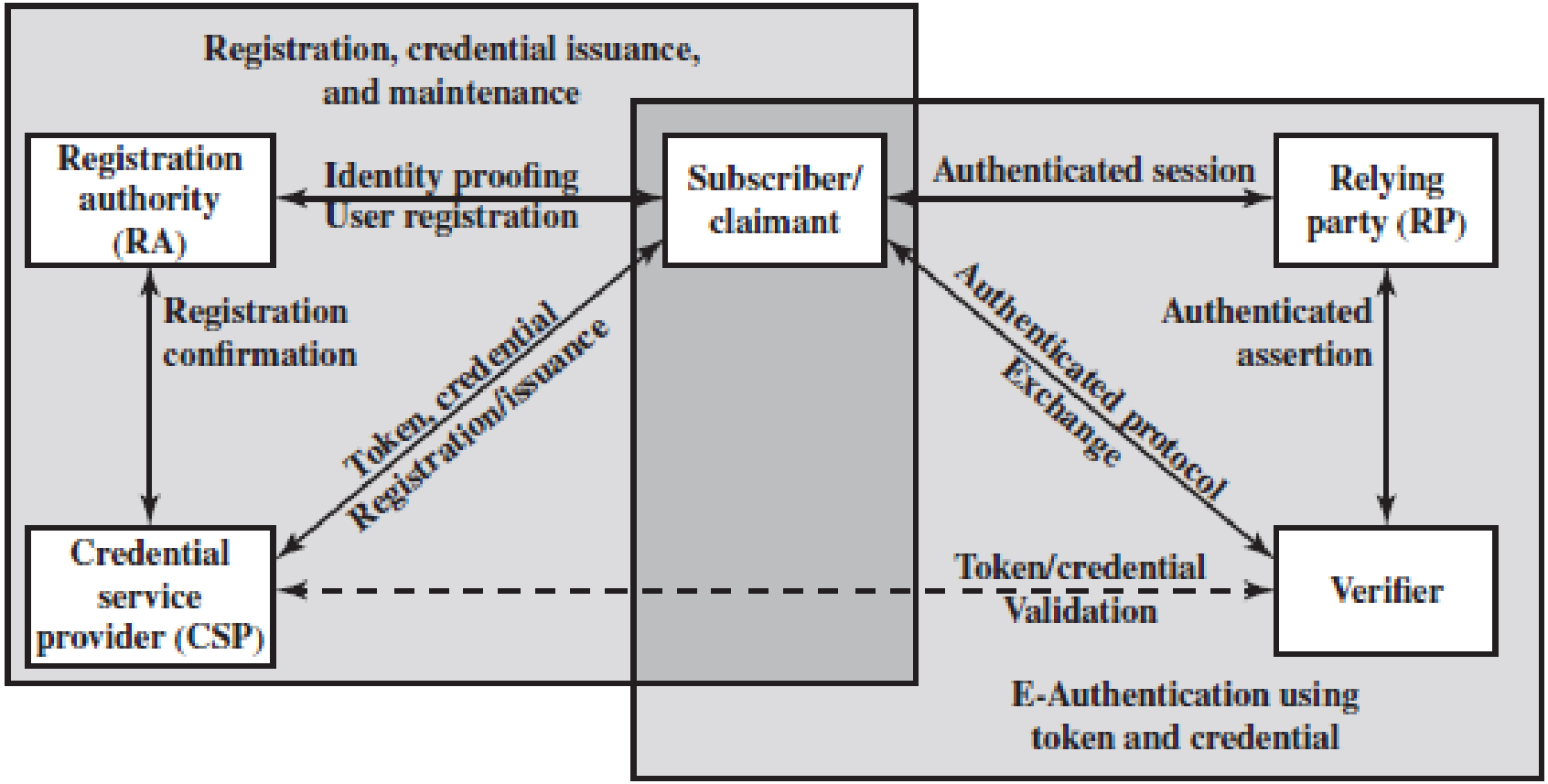


Figure 3.1 The NIST SP 800-63-3 E-Authentication Architectural Model

Means of Authentication

- There are four general means of authenticating a user's identity, which can be used alone or in combination:
- **Something the individual knows:**
  - *Examples include 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*
- All of these methods, properly implemented and used, can provide secure user authentication.
  - ***However, each method has problems.***

Means of Authentication

- ***Multifactor authentication*** refers to the use of more than one of the authentication means in the preceding
- The strength of authentication systems is largely determined by the ***number of factors incorporated by the system***
  - *Implementations that use* ***two factors are considered*** *to* ***be stronger than those that use only one factor***
  - *Systems that incorporate* ***three factors are stronger than systems that only incorporate two of the factors***
    - And so on.

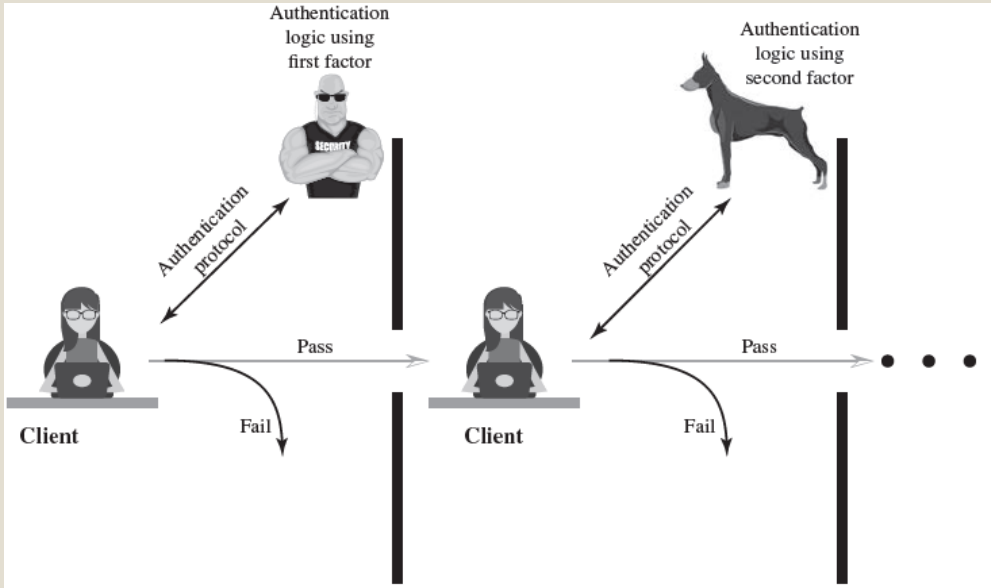


Figure 3.2 **Multifactor Authentication**

Risk assessment for user authentication

- Three separate concepts
  - ***Assurance level, Potential impact, and Areas of risk.***
- **Assurance level:** the degree of certainty that a user has presented a credential that refers to his/her identity
  - ***Level 1:*** *little confidence (an online forum)*
  - ***Level 2:*** *some confidence (professional organizations)*
  - ***Level 3:*** *High confidence (patent office applicants)*
  - ***Level 4:*** *Very high confidence (employees accessing restricted/sensitive services)*
- **Potential impact:** ***Potential Impact*** A concept closely related to that of assurance level is potential impact. FIPS
  - ***defines three levels of potential impact on organizations or individuals should there be a breach of security (in our context, a failure in user authentication)***
  - *low, moderate, impact*

Risk assessment for user authentication

Table 3.2 **Maximum Potential Impacts for Each Assurance Level**

	Assurance Level Impact Profiles			
Potential Impact Categories for Authentication Errors	**1**	**2**	**3**	**4**
Inconvenience, distress, or damage to standing or reputation	Low	Mod	Mod	High
Financial loss or organization liability	Low	Mod	Mod	High
Harm to organization programs or interests	None	Low	Mod	High
Unauthorized release of sensitive information	None	Low	Mod	High
Personal safety	None	None	Low	Mod/ High
Civil or criminal violations	None	Low	Mod	High

PASSWORD-BASED AUTHENTICATION

- A widely used line of defense against intruders is the password system
  - *User provides name/login and password*
  - *System compares password with the one stored for that specified login*
- The user ID:
  - ***Determines that the user is authorized to access the system***
    - In some systems, only those who already have an ID filed on the system are allowed to gain access.
  - ***Determines the user's privileges***
    - Some systems have guest or anonymous accounts, and users of these accounts have more limited privileges than others.
  - ***Is used in discretionary access control***
    - For example, by listing the IDs of the other users, a user may grant permission to them to read files owned by that user.

The Vulnerability of Passwords

- The main forms of attack against ***password-based authentication and countermeasures***
- **Offline dictionary attack:**
  - *Strong access controls are used to protect the system's password file.*
    - Experience shows that ***determined hackers*** can frequently bypass such controls and ***gain access*** to the file.
    - The attacker obtains the ***system password file*** and ***compares*** the password hashes ***against*** of commonly used passwords.
  - ***Countermeasures:***

1. *Controls to prevent unauthorized access to the password file*
2. *Intrusion detection measures to identify a compromise*
3. *Rapid reissuance of passwords should the password file be compromised.*

- **Specific account attack:**
  - *The attacker targets a* ***specific account*** *and* ***submits password guesses*** *until the correct password is discovered.*
  - ***Countermeasure:***
    - An **account lockout mechanism,** which locks out access to the ***account after a number of failed login attempts.***
      - ***Typical practice*** *is no more than* ***five*** *access* ***attempts.***

The Vulnerability of Passwords

- **Popular password attack (against a wide range of IDs)**
  - *A variation of the preceding attack is to use **a popular password and try it against a wide range of user IDs.***
    - A user tendency is to choose a password that is easily remembered:
  - *Countermeasures:*
    - policies to inhibit the selection by users of common passwords and
    - scanning the IP addresses of authentication requests and client cookies for submission patterns.

- **Password guessing against single user:**
  - *The attacker attempts to **gain knowledge** about **the account holder and system password policies** and uses that knowledge to **guess** the password.*
  - *Countermeasures:*
    - training in and **enforcement of password policies** that make passwords **difficult** to guess.
    - Such policies address the secrecy
      1. *minimum length of the password*
      2. *character set*
      3. *prohibition against using well-known user identifiers, and*
      4. *length of time before the password must be changed.*

The Vulnerability of Passwords

- **Workstation hijacking:**
  - *The attacker waits until* ***a logged-in workstation is unattended.***
  - *Countermeasure :*
    - automatically logging the workstation out after a period of inactivity.
    - **Intrusion detection** schemes can be used to detect changes in **user behavior.**
- **Exploiting user mistakes:**
  - *If the system assigns a password, then the user is more likely to* ***write it down because it is difficult to remember***
    - This situation creates the potential for an adversary to read the written password.
  - *Many computer systems are shipped with* ***preconfigured passwords for system administrators.***
    - Unless these preconfigured passwords are changed, they are easily guessed.
  - *Countermeasures:*
    - user training
    - intrusion detection, and
    - simpler passwords combined with another authentication mechanism.

The Vulnerability of Passwords

- **Exploiting multiple password use:**
  - *Attacks can also become much more effective or* ***damaging if different network devices share the same or a similar password for a given user.***
  - ***Countermeasures*** *include a policy that forbids the same or similar password on particular network devices.*

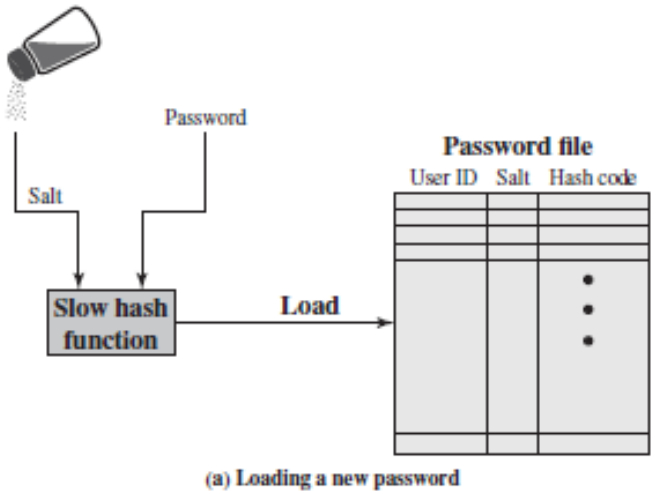
- **Electronic monitoring:**
  - *If a password is* ***communicated across a network*** *to log on to a remote system, it is vulnerable* ***to eavesdropping (E.g. sniffing).***
  - *Simple encryption will not fix this problem, because the encrypted password is, in effect, the password and can be observed and reused by an adversary.*
  - ***Countermeasures:***
    - **We need additional techniques**

Use of hashed passwords

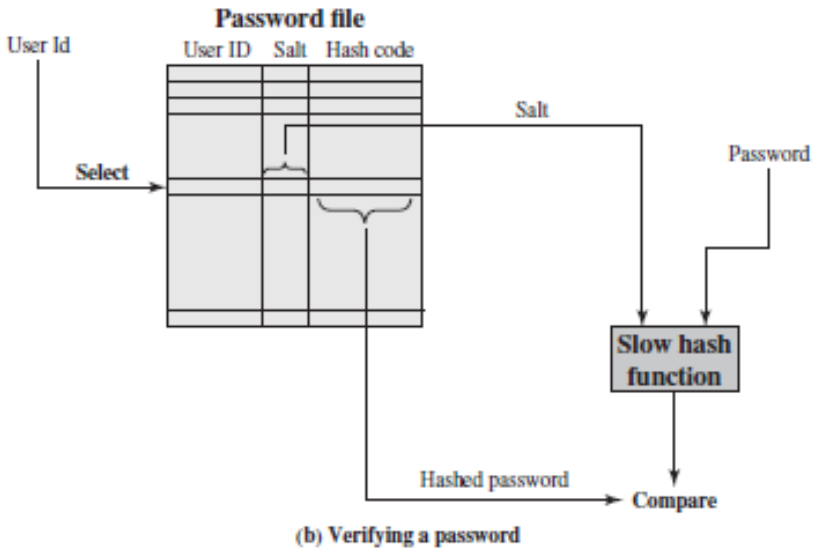
- A widely used password security technique
  - *use of hashed passwords and a salt value*
  - *This scheme is found on virtually all UNIX operating system*

1. To load a new password into the system, the user selects or is assigned a password. . (see Figure 3.3a).
   - *This password is **combined with a fixed-length salt value***
     - use a pseudorandom or random number.
       - *can be anything **like time, date etc.***

- The hashed password is then stored,
  - *together with a plaintext copy of the salt, in the password file for the corresponding user ID*



(a) Loading a new password

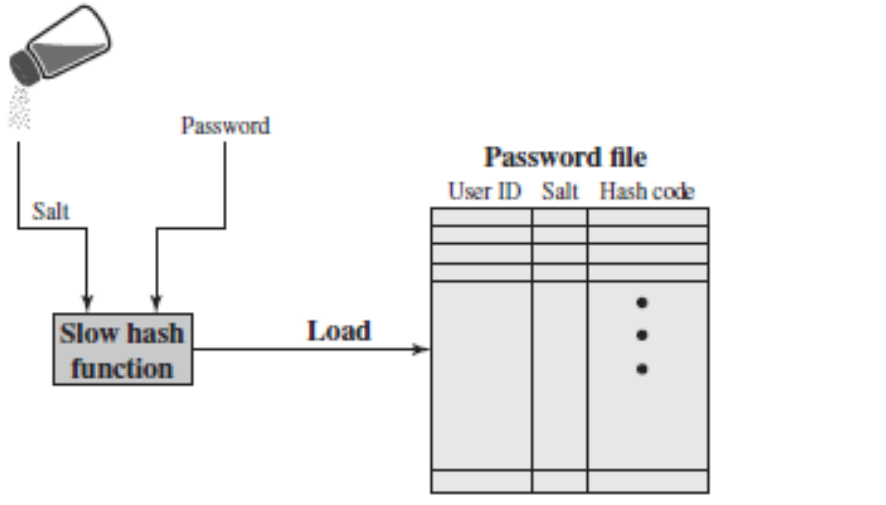


(b) Verifying a password

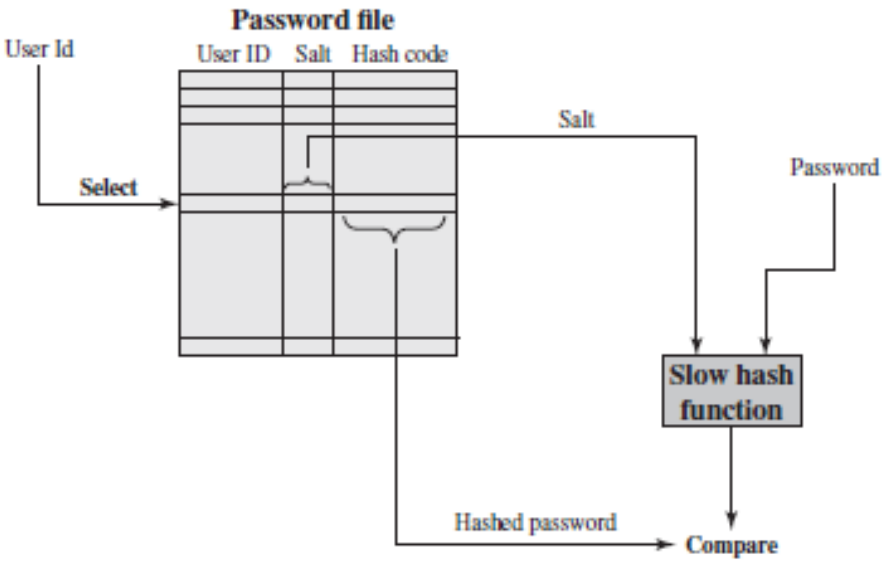
Figure 3.3 **UNIX Password Scheme**

Use of hashed passwords

- When a user attempts to log on to a UNIX system, the user provides:
  - ***an ID and a password*** *(see Figure 3.3b).*
  - *The operating system uses:*
    - ***the ID to index into the password file and retrieve the plaintext salt and the encrypted password.***
    - **The salt and user-supplied password are used as input to the encryption routine.**
  - *If the result matches the stored value, the password is accepted.*



(a) Loading a new password



(b) Verifying a password

Figure 3.3 **UNIX Password Scheme**

Why a salt value?

- The **salt serves** three purposes:
  1. *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.*
  2. *It greatly increases the **difficulty of offline dictionary attacks.***
     - For a **salt of length *b* bits**, the number of possible passwords is increased by a factor of 2*b*,
       - *Increasing the difficulty of guessing a password in a dictionary attack.*
  3. *Nearly **impossible** to tell if a person **used the same password on multiple systems***

The vulnerability of Passwords

- Two threat to the UNIX password scheme
  - *Gaining access on a machine and then run a password guessing program on that machine with little resource consumption*
  - *Obtaining a copy of the password file, then a cracker program can be run on another machine*
- The attacker should be able to check many thousands of possible passwords with little resource consumption.
- If an opponent is able to obtain a copy of the password file, then a cracker program can be run on another machine at leisure.
- This enables the opponent to run through millions of possible passwords in a reasonable period.

Passwords must NOT be too short, NOT be too easy to guess

UNIX Implementation

- Original scheme
  - *8 character password form 56-bit key*
  - *12-bit salt used to modify DES encryption into a one-way hash function*
    - This process is repeated for a total of **25 encryptions.**
  - *output translated to 11 character sequence*
- The crypt(3) routine is designed to discourage guessing attacks.
- Now regarded as woefully insecure
  - *e.g. supercomputer, 50 million tests, 80 min*
- Sometimes still used for compatibility

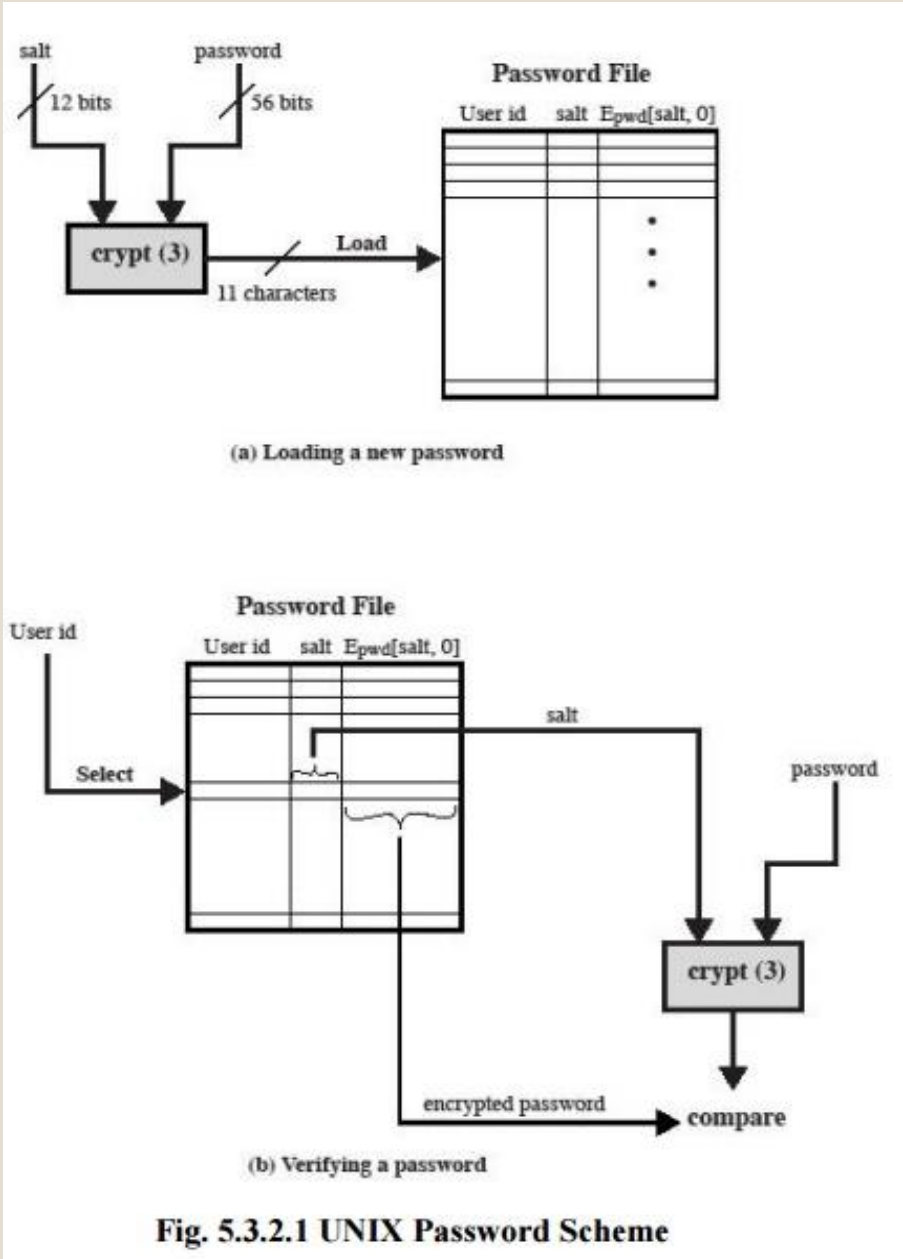


Fig. 5.3.2.1 UNIX Password Scheme

Improved implementations

- Have other, stronger, hash/salt variants
- Many systems now use MD5
  - *with 48-bit salt*
  - *password length is unlimited*
  - *is hashed with 1000 times inner loop*
  - *produces 128-bit hash*
- OpenBSD uses Blowfish block cipher based and hash algorithm called Bcrypt
  - *uses 128-bit salt to create 192-bit hash value*

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
- **less computer processing time and more storage than a brute-force attack**
- Can be **countered** by using a **sufficiently large salt** value and a sufficiently **large hash length**

Exploiting easily guessable passwords

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

Modern Approaches

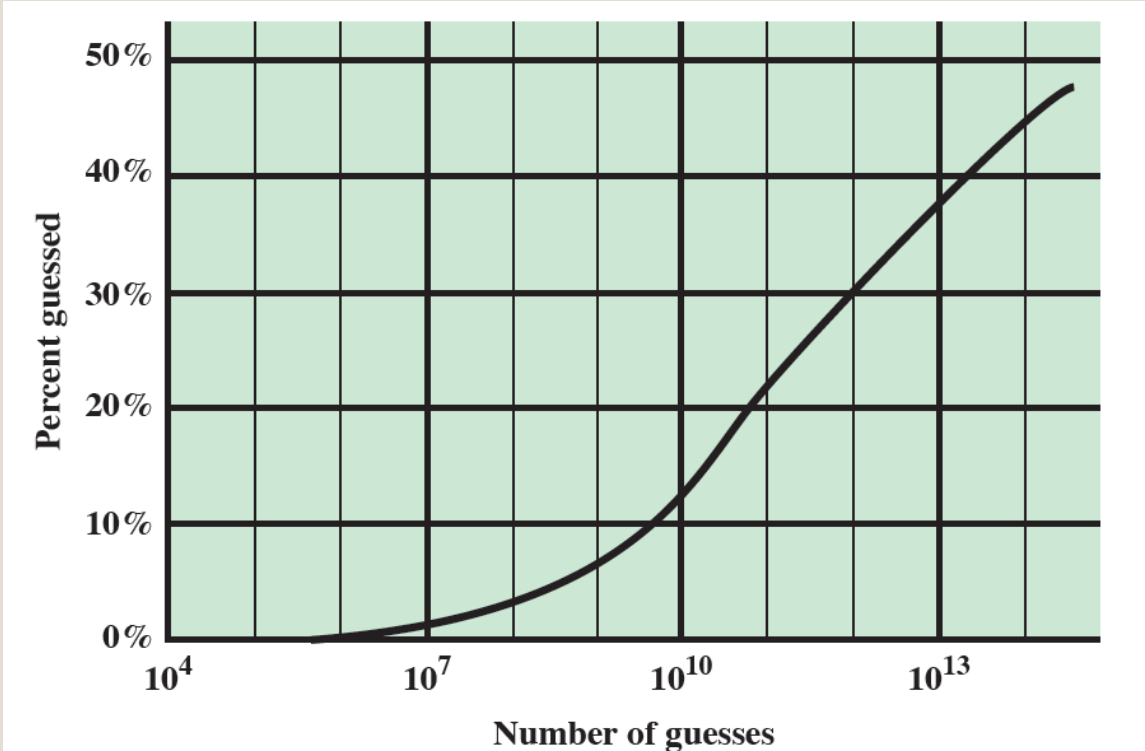
- **Complex password policy:** **Forcing users to pick stronger passwords**
  - *Users are doing a better job of selecting passwords, and*
  - *organizations are doing a better job of forcing users to pick stronger passwords,*
- But password-cracking techniques have also improved
  - *The **processing capacity** available for password cracking has increased dramatically (GPUs)*
  - *The use of **sophisticated algorithms** to generate potential passwords*
  - *Studying examples and structures of actual passwords in use*
    - ***(may employ modern machine learning techniques)***
      - ***Using large datasets of leaked passwords as training data***

Password choices/concerns

- users may pick short passwords
  - *e.g. 3% were 3 chars or less, easily guessed*
  - *system can reject choices that are too short*
- users may pick guessable passwords
  - *so crackers use lists of likely passwords*
  - *e.g. one study of 14000 encrypted passwords guessed nearly 1/4 of them*
  - *would take about 1 hour on fastest systems to compute all variants, and only need 1 break!*

Another case study

- An analysis of passwords used by 25,000 students at a research university with a complex password policy.
- The graph shows the percentage of passwords that have been recovered as a function of the number of guesses.
  - *Over 10% recovered after 10^10 guesses*
  - *After 10_{13} guesses, almost 40% of the passwords are recovered.*

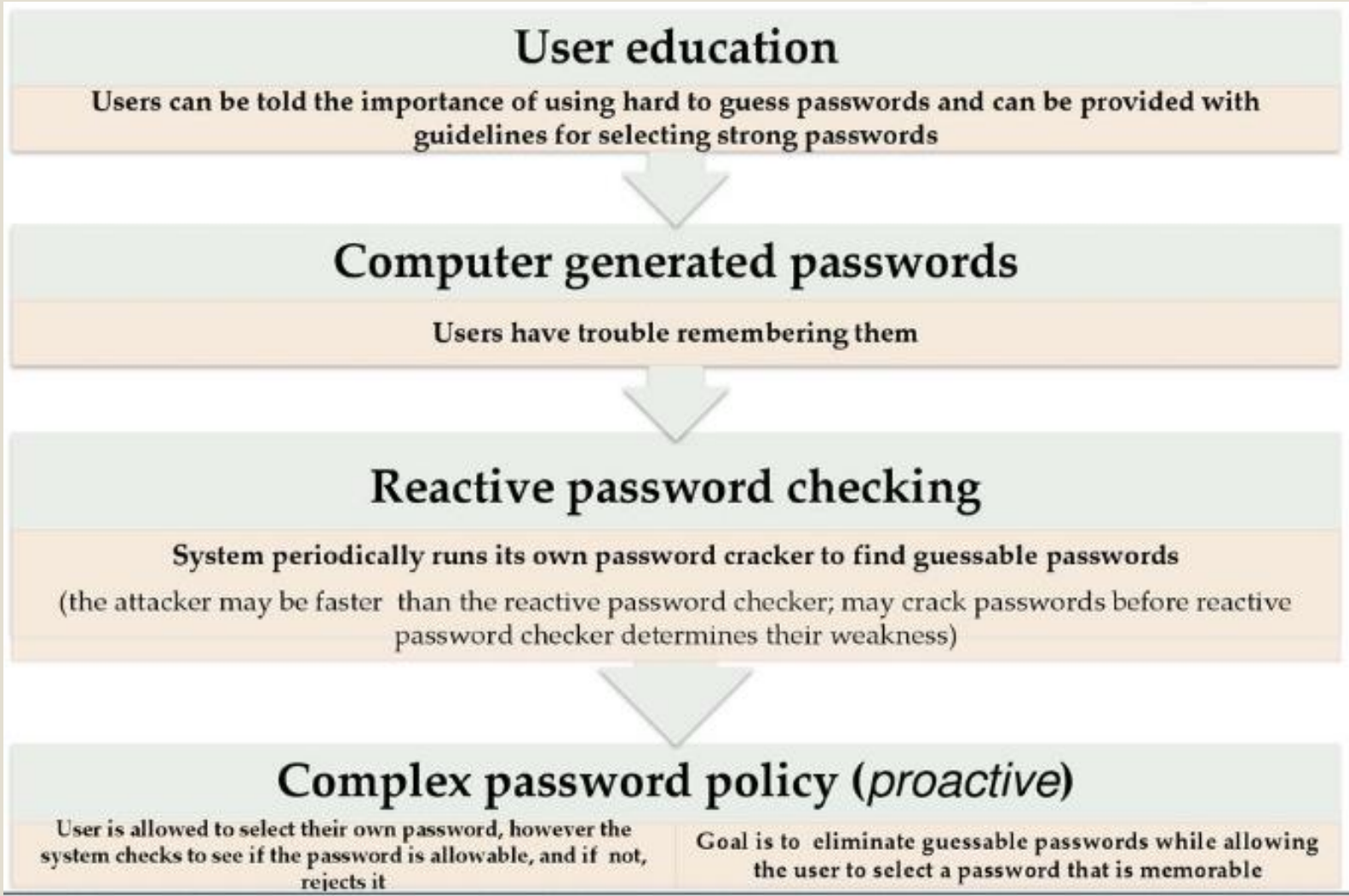


Password File Access Control

- Can block offline guessing attacks by denying access **to encrypted passwords**
  - ***make available only to privileged users***
  - ***often using a separate shadow password***
- Still have vulnerabilities
  - *Weakness in the OS that allows access to the file*
  - *Accident with permissions making it readable*
  - *Users with same password on other systems*
  - *Access from unprotected backup media*
  - *Sniff passwords in unprotected network traffic*

Password select strategies

- Goal to eliminate guessable passwords
  - *Still easy for user to remember*
- Techniques



Proactive Password Checking

- Rule enforcement plus user advice, e.g.
  - *8+ chars, upper/lower/numeric/punctuation*
  - *may not suffice*
- Password cracker
  - *Procedure is simply to compile a large dictionary of possible "bad" passwords.*
  - *When a user selects a password, the system checks to make sure that it is not on the disapproved list.*
    - time and space issues
- Bloom Filter
  - *technique [SPAF92a, SPAF92b] for developing an effective and efficient proactive password checker that is based on rejecting words on a list has been implemented on a number of systems, including Linux.*
  - *use to build table based on dictionary using hashes*
  - *check desired password against this table*

TOKEN-BASED AUTHENTICATION

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

Table 3.3 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

Memory Card

- Can store but do not process data
- The most common is the magnetic stripe card
  - *Can be reprogrammed by inexpensive card reader*
- Can include an internal electronic memory
- Can be used alone for physical access
  - *Hotel room*
  - *ATM*
- Provides significantly greater security when combined with a password or PIN
- Drawbacks of memory cards include:
  - *Requires a special reader*
    - Cost of the reader and keeping it secure
  - *Loss of token*
    - Prevents the owner from gaining system access
    - Adversary needs only to find the PIN
  - *User dissatisfaction:*
    - for computer access may be inconvenient

Smart token

- A wide variety of devices qualify as smart tokens.
- These can be categorized along three dimensions that are not mutually exclusive:
  - ***Physical characteristics:-***
  - *Smart tokens include an **embedded microprocessor.***
  - *A smart token that looks like a **bank card is called a smart card**.*
    - Other tokens can look **like calculators, keys, or other small portable objects**
  - ***Interface:-***
  - *Manual interfaces include a keypad and display for human/token interaction.*
  - *Smart tokens with an electronic interface communicate with a compatible reader/writer*

Smart token

- **Authentication protocol:** Classified the authentication protocols into three categories:
  - *Static:*
    - The user authenticates himself or herself to the token and then the token authenticates the user to the computer.
  - *Dynamic password generator*
    - 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.*
      - *The token and the computer system must be initialized and kept synchronized so that the computer knows the password that is current for this token*
  - *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.
      - *For example, public-key cryptography could be used and the token could encrypt the challenge string with the token's private key.*
      - *The simplest example of a challenge–response protocol is password authentication, where the challenge is asking for the password and the valid response is the correct password.*

Smart cards

- Most important category of smart token
  - *Has the appearance of a credit card*
  - *Has an electronic interface*
  - *May use any of the smart token protocols*
- Contain:
  - *An entire microprocessor*
    - Processor
    - Memory
    - I/O ports
- 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

Smart cards

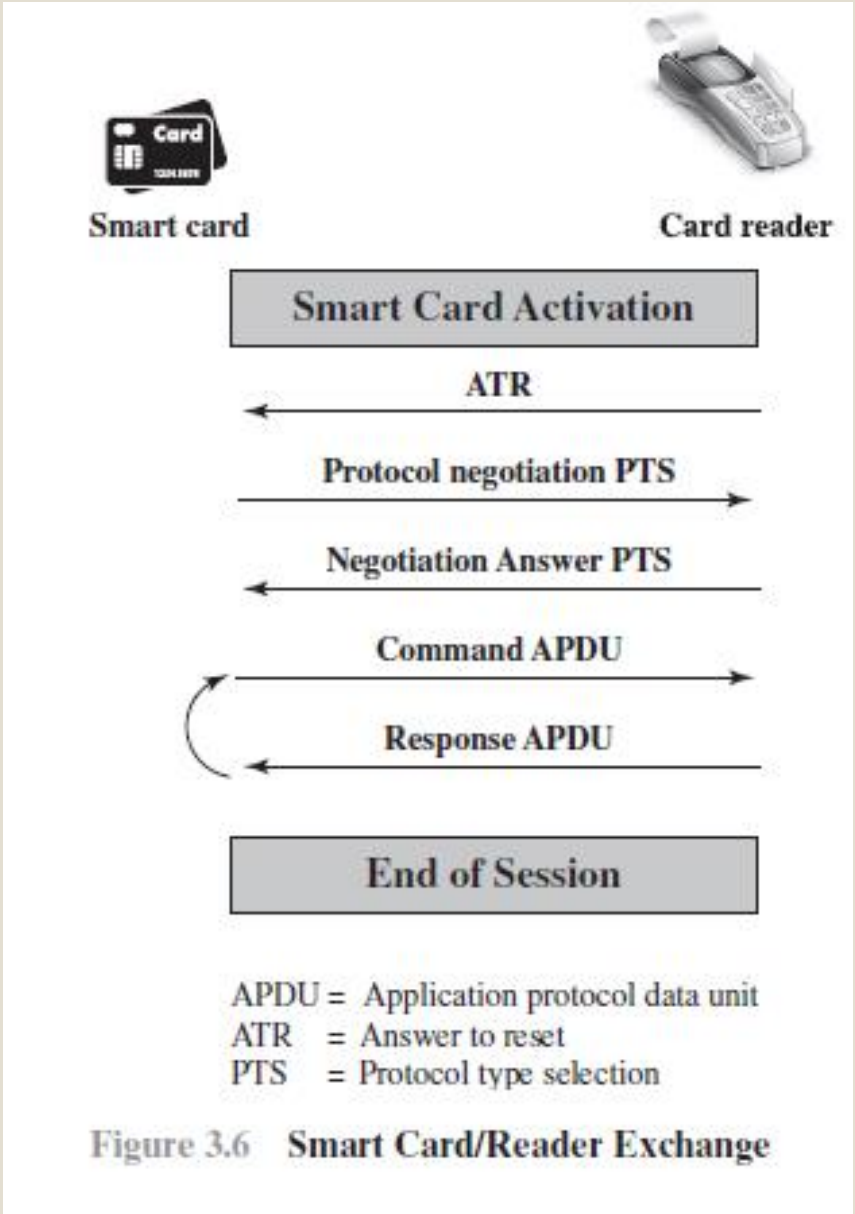


Figure 3.6 **Smart Card/Reader Exchange**

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, and similar cards such as a driver's license, for access to government and commercial services

Can provide stronger proof of identity and can be used in a wider variety of applications

In effect, is a smart card that has been verified by the national government as valid and authentic

Most advanced deployment is the German card *neuer Personalausweis*

Has human-readable data printed on its surface

- Personal data
- Document number
- Card access number (CAN)
- Machine readable zone (MRZ)

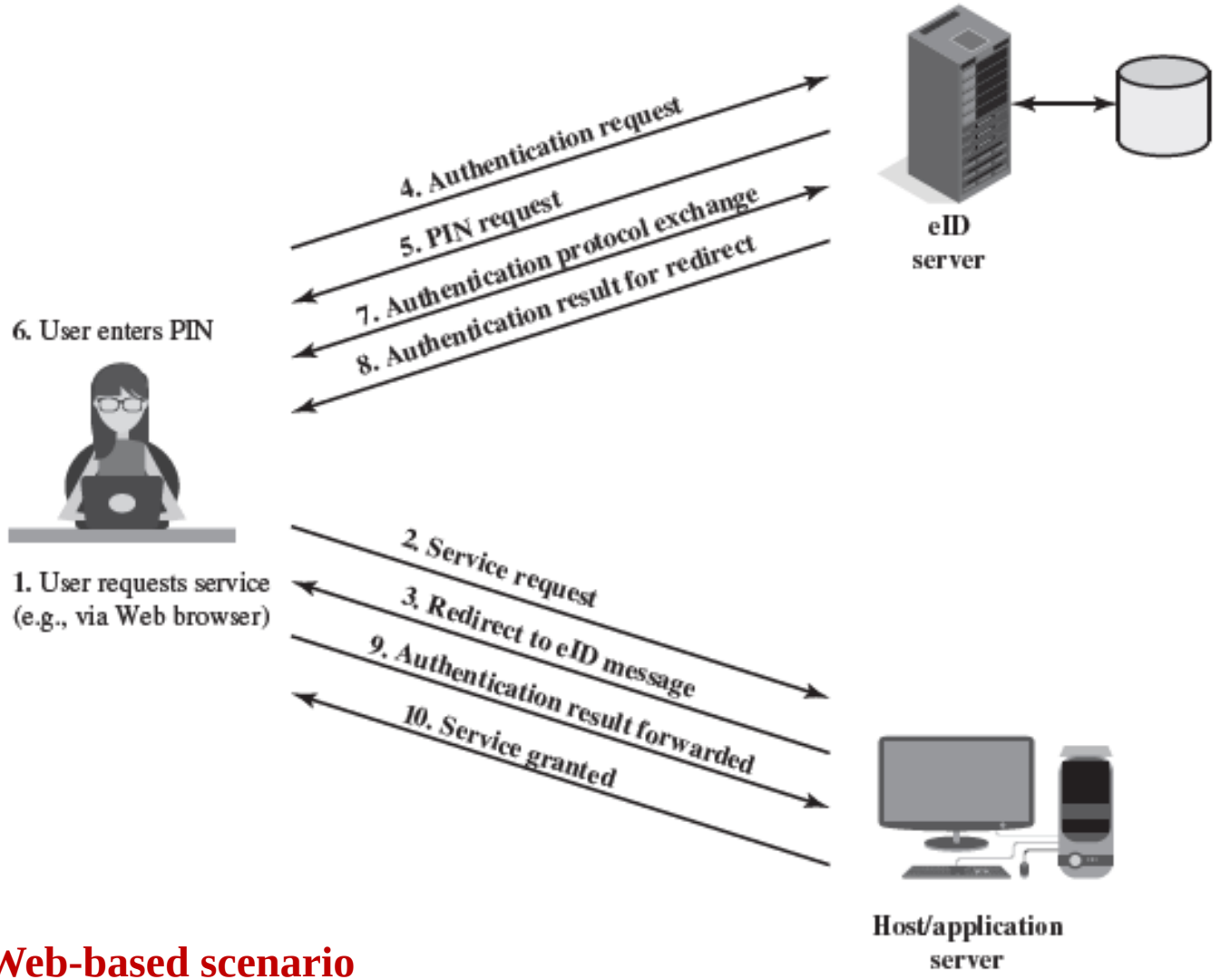
Electronic Identity Cards (eID)

Table 3.4 **Electronic Functions and Data for eID Cards**

Function	Purpose	PACE Password	Data	Uses
ePass (mandatory)	Authorized offline inspection systems read the data.	CAN or MRZ	Face image; two fingerprint images (optional); MRZ data	Offline biometric identity verification reserved for government access
eID (activation optional)	Online applications read the data or access functions as authorized.	eID PIN	Family and given names; artistic name and doctoral degree: date and place of birth; address and community ID; expiration date	Identification; age verification; community ID verification; restricted identification (pseudonym); revocation query
	Offline inspection systems read the data and update the address and community ID.	CAN or MRZ		
eSign (certificate optional)	A certification authority installs the signature certificate online.	eID PIN	Signature key; X.509 certificate	Electronic signature creation
	Citizens make electronic signature with eSign PIN.	CAN		

CAN = card access number
MRZ = machine-readable zone
PACE = password authenticated connection establishment
PIN = personal identification number

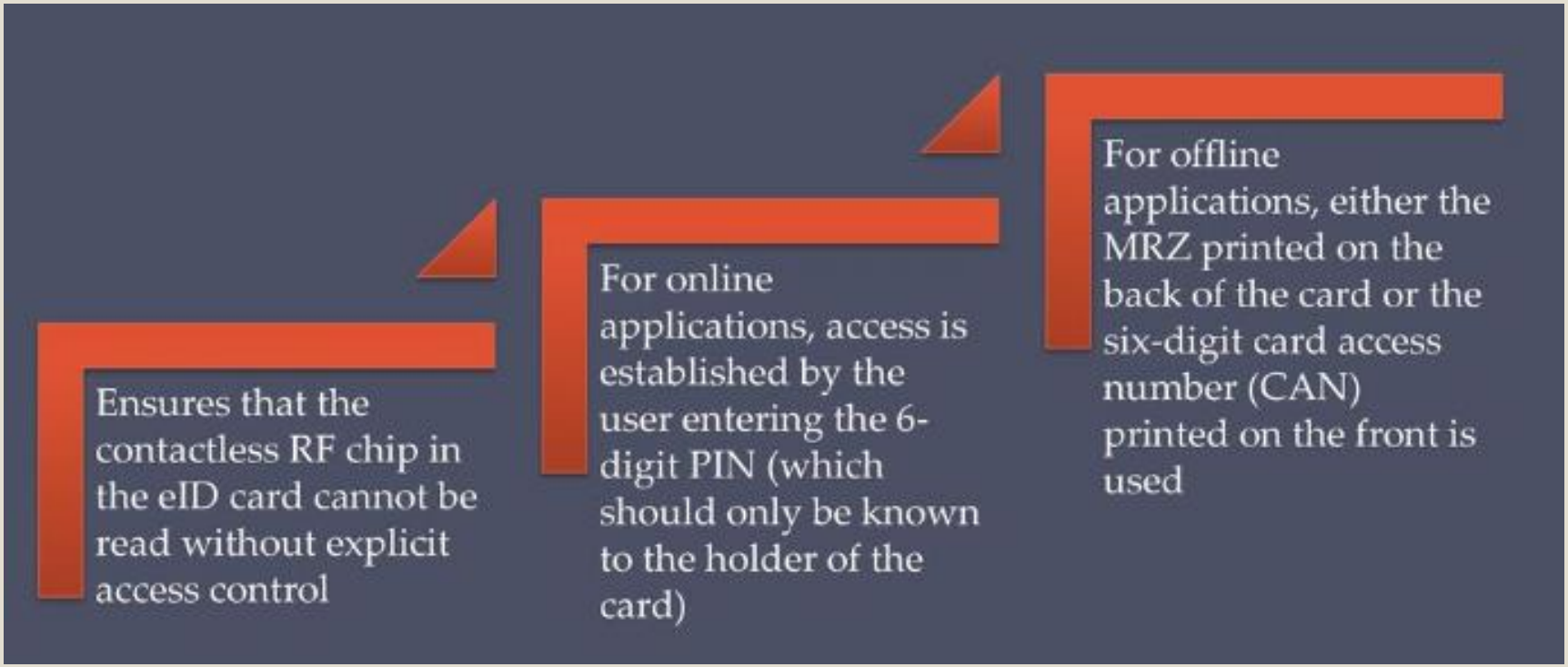
User authentication with eID



Web-based scenario

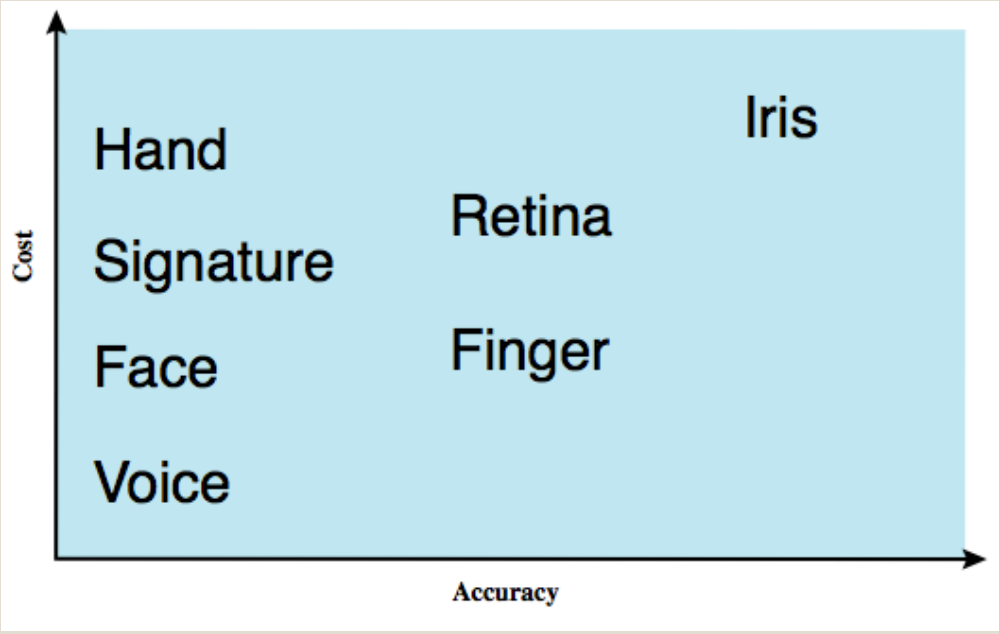
Figure 3.7 **User Authentication with eID**

Password Authenticated Connection Establishment (PACE)



Biometric Authentication

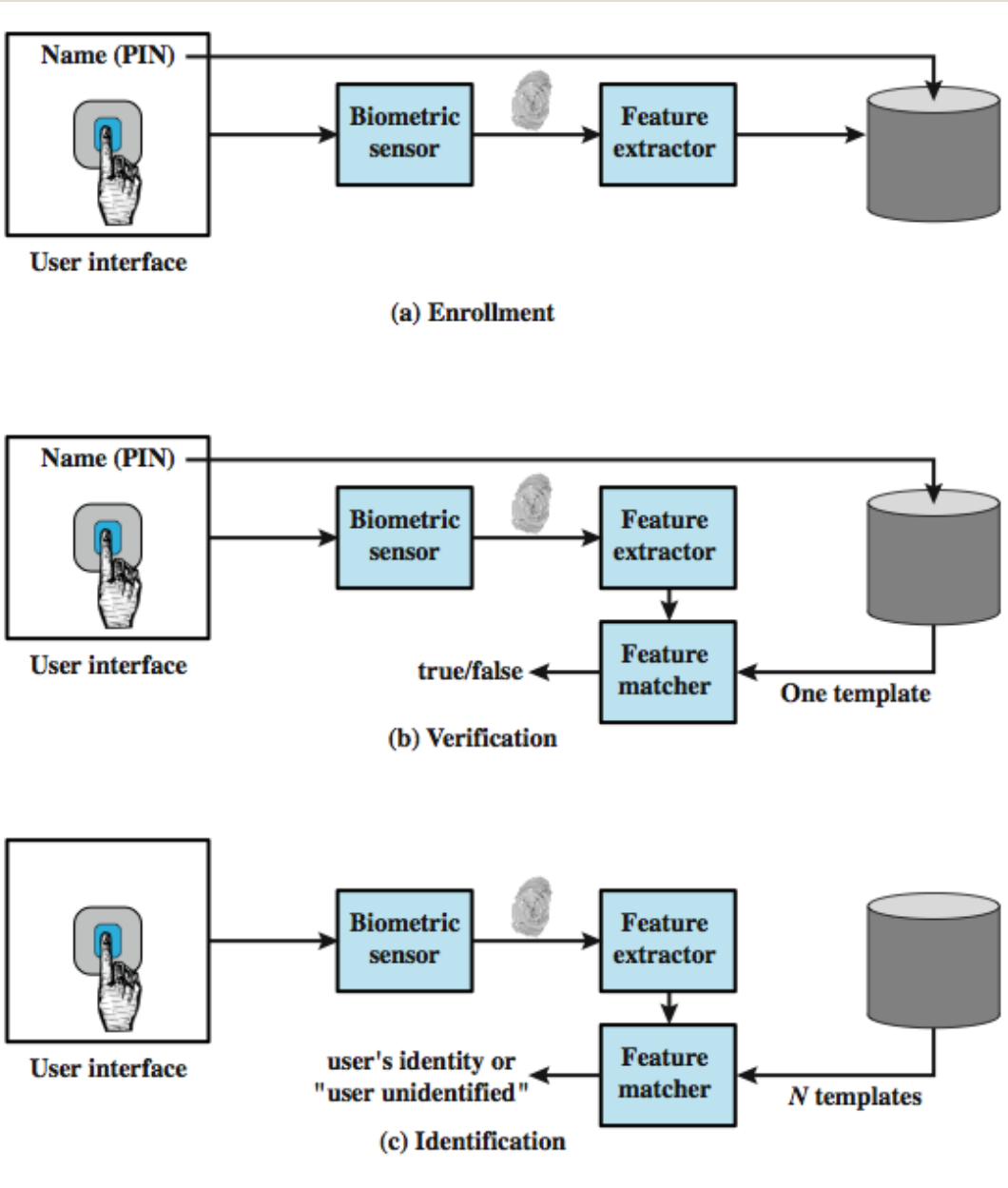
- Attempts to authenticate an individual based on unique physical characteristics
- Based on pattern recognition
- Is technically complex and expensive when compared to passwords and tokens
- Physical characteristics used include:
  - *Facial characteristics*
  - *Fingerprints*
  - *Hand geometry*
  - *Retinal pattern*
  - *Iris*
  - *Signature*
  - *Voice*



Operation of a biometric system

Enrollment is analogous to assigning a password to a user.

Verification is analogous to a user logging on to a system by using a memory card or smart card coupled with a password or PIN.

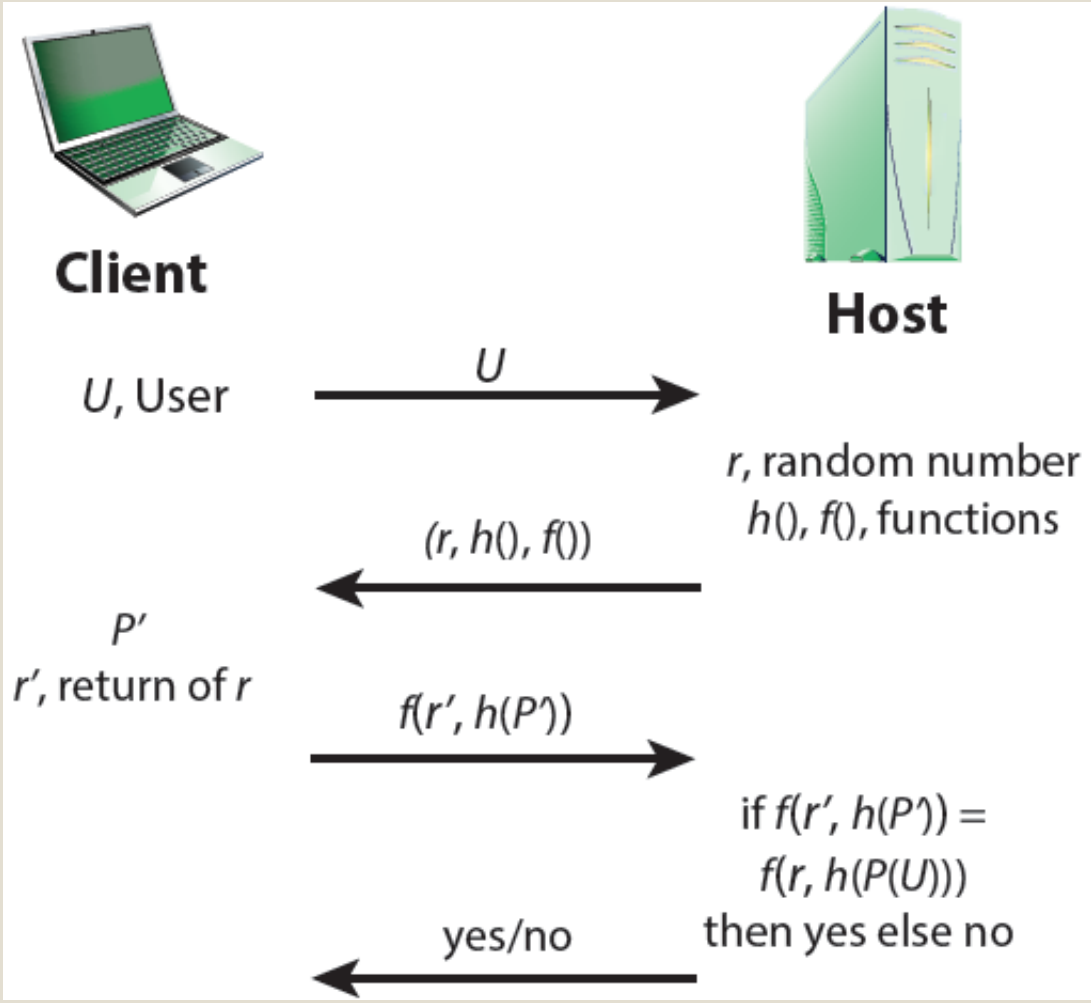


REMOTE USER AUTHENTICATION

- Authentication over a network, the Internet, or a communications link is more complex
- Additional security threats such as:
  - ***Eavesdropping, capturing a password, replaying an authentication sequence that has been observed***
- Generally use challenge-response
  - *user sends identity*
  - *host responds with random number r*
  - *user computes f(r,h(P)) and sends back*
  - *host compares value from user with own computed value, if match user authenticated*
- The simplest form of user authentication is local authentication, in which a user attempts to access a system that is locally present, such as
  - ***a stand-alone office PC or an ATM machine**.*

Protocol for a password verification

- Similar approach for token and biometric verification
- Figure 3.13a provides a simple example of a challenge-response protocol



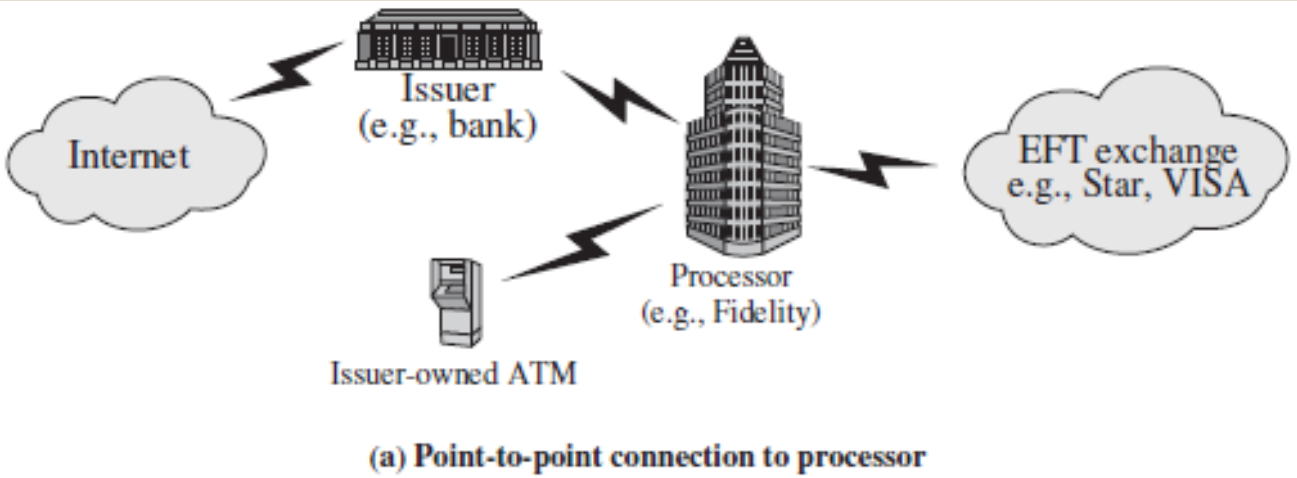
SECURITY ISSUES FOR USER AUTHENTICATION

Some Potential Attacks, Susceptible Authenticators, and Typical Defenses

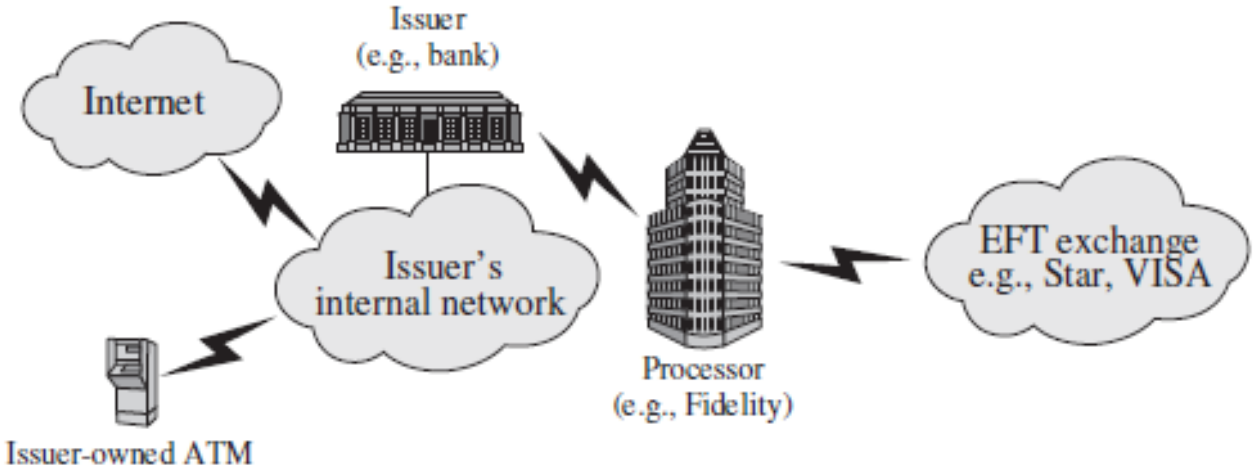
Table 3.5 **Some Potential Attacks, Susceptible Authenticators, and Typical Defenses**

Attacks	Authenticators	Examples	Typical Defenses
Client attack	Password	Guessing, exhaustive search	Large entropy; limited attempts
	Token	Exhaustive search	Large entropy; limited attempts; theft of object requires presence
	Biometric	False match	Large entropy; limited attempts
Host attack	Password	Plaintext theft, dictionary/exhaustive search	Hashing; large entropy; protection of password database
	Token	Passcode theft	Same as password; 1-time passcode
	Biometric	Template theft	Capture device authentication; challenge response
Eavesdropping, theft, and copying	Password	"Shoulder surfing"	User diligence to keep secret; administrator diligence to quickly revoke compromised passwords; multifactor authentication
	Token	Theft, counterfeiting hardware	Multifactor authentication; tamper resistant/evident token
	Biometric	Copying (spoofing) biometric	Copy detection at capture device and capture device authentication
Replay	Password	Replay stolen password response	Challenge-response protocol
	Token	Replay stolen passcode response	Challenge-response protocol; 1-time passcode
	Biometric	Replay stolen biometric template response	Copy detection at capture device and capture device authentication via challenge-response protocol
Trojan horse	Password, token, biometric	Installation of rogue client or capture device	Authentication of client or capture device within trusted security perimeter
Denial of service	Password, token, biometric	Lockout by multiple failed authentications	Multifactor with token

CASE STUDY: SECURITY PROBLEMS FOR ATM SYSTEMS



(a) Point-to-point connection to processor



(b) Shared connection to processor

Figure 3.15 **ATM Architectures** Most small to mid-sized issuers of debit cards contract processors to provide core data processing and electronic funds transfer (EFT) services. The bank's ATM machine may link directly to the processor or to the bank.