CMPE 209 Network Security

User Authentication & Authentication Protocols

Dr. Younghee Park

SAN JOSÉ STATE UNIVERSITY

RFC 4949

RFC 4949 defines user authentication as:

"**The process of verifying an identity claimed by or for a system entity.**"

Authentication Process

- Fundamental building block and primary line of defense
- Basis for access control and user accountability
- Two steps
  - **Identification step**
    - Presenting an identifier to the security system
  - **Verification step**
    - Presenting or generating authentication information that corroborates the binding between the entity and the identifier

General Architecture

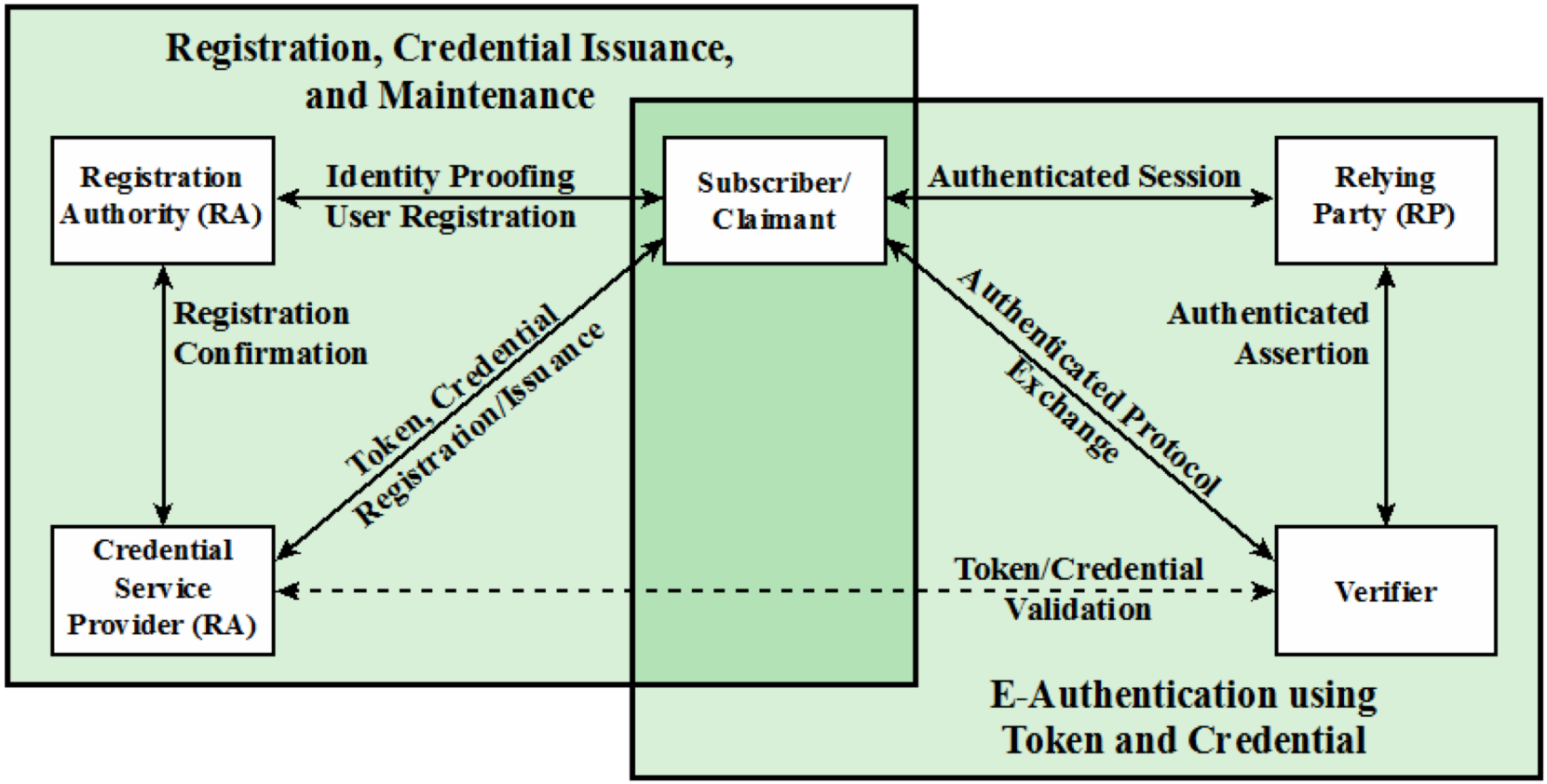


Figure 3.1 The NIST SP 800-63-2 E-Authentication Architectural Model

Authenticating User Identity (4 Methods)

- **Something the individual knows**
  - Password, PIN, answers to prearranged questions
- **Something the individual possesses (token)**
  - Smartcard, electronic keycard, physical key
- **Something the individual is (static biometrics)**
  - Fingerprint, retina, face
- **Something the individual does (dynamic biometrics)**
  - Voice pattern, handwriting, typing rhythm

Password Authentication

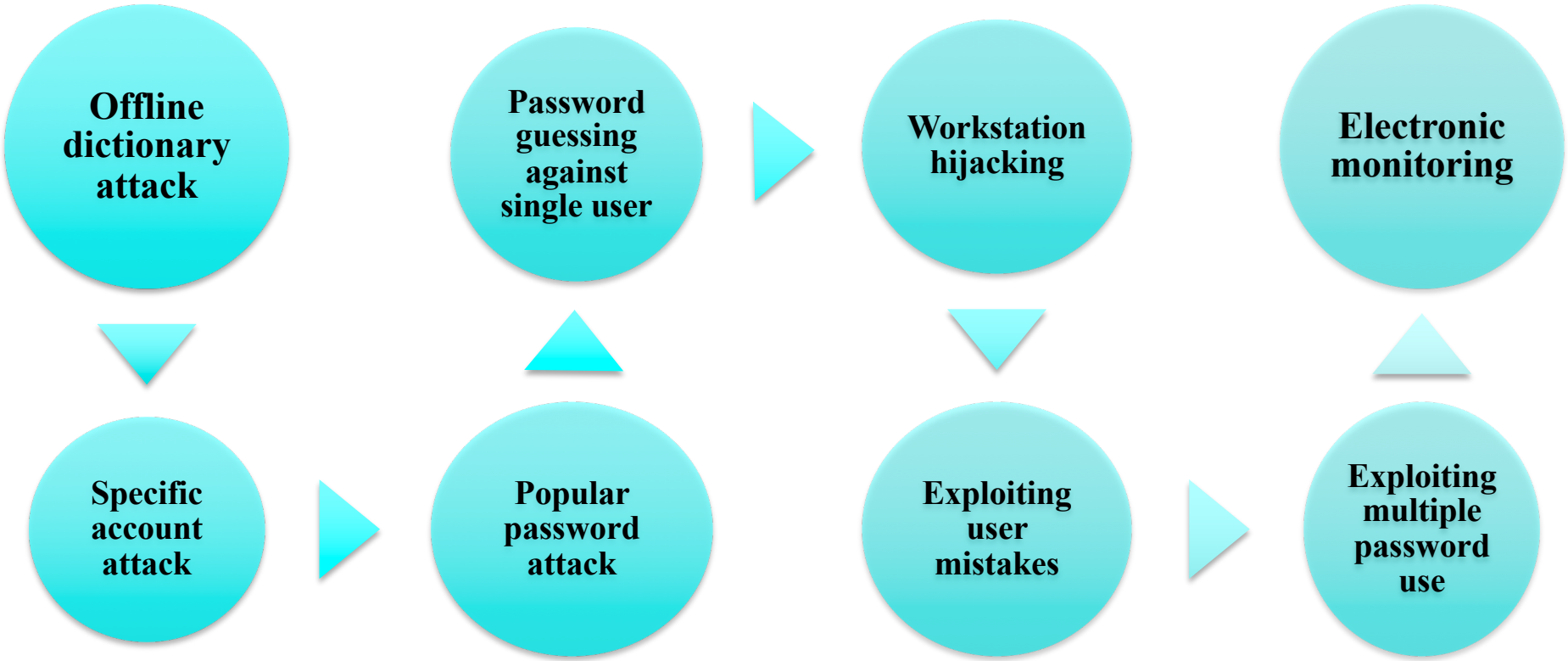
- Widely used line of defense against intruders
  - User provides name/login and password
  - System compares password with the one stored for that specified login
- The user ID:
  - Determines that the user is authorized to access the system
  - Determines the user's privileges
  - Is used in discretionary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

Authentication Mechanisms

- Password-based authentication
  - Use a secret quantity (the password) that the verifier states to prove he/she knows it.
  - Threat: password guessing/dictionary attack

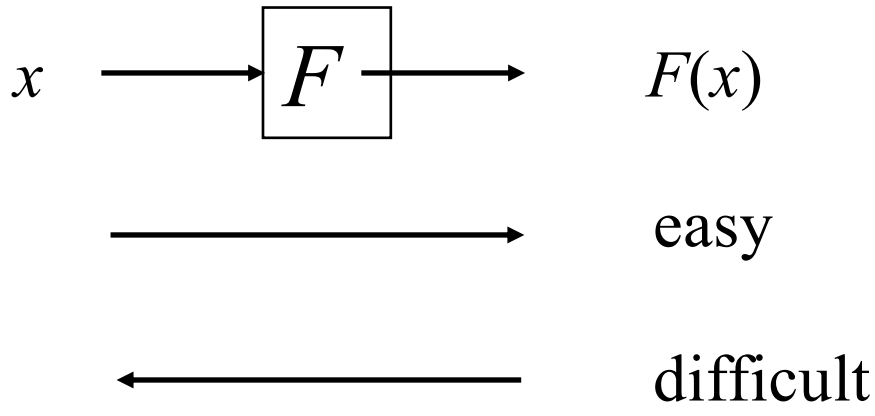
    - How is the password stored?
    - How does the system check the password?
    - How easy is it to guess the password
      - Easy-to-remember passwords tend to be easy to guess
      - Password files is difficult to keep secret

Password Vulnerabilities



One-Way Hash Function

- One-way hash function F
  - $F(x)$ is easy to compute
  - From $F(x)$, x is difficult to compute
  - Example: $F(x) = g^x \bmod p$, where p is a large prime number and g is a primitive root of p.



Storing Passwords

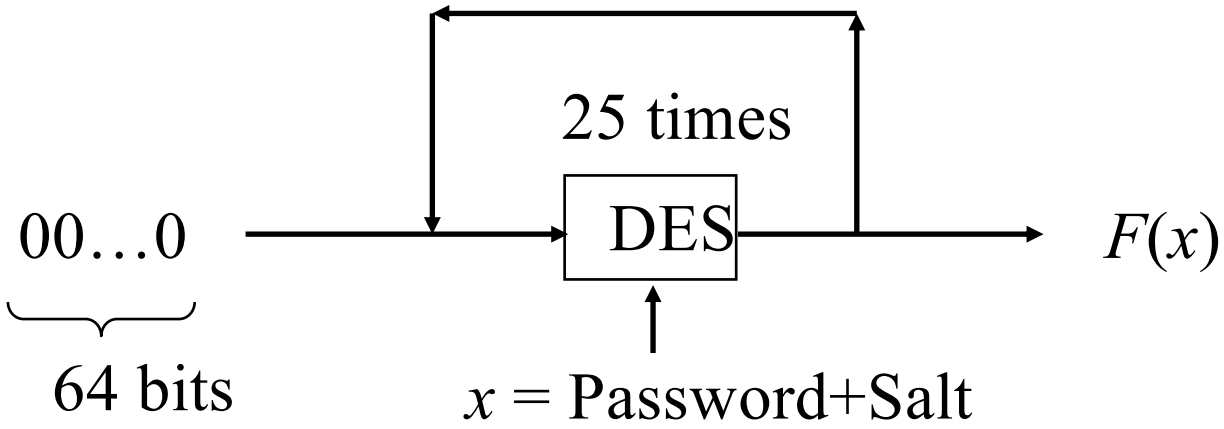
- For each user, system stores

  (*user name*, *F*(*password*))

  in a password file, where *F* is a one-way hash function
- When a user enters the password, system computes *F*(*password*); a match provides proof of identity

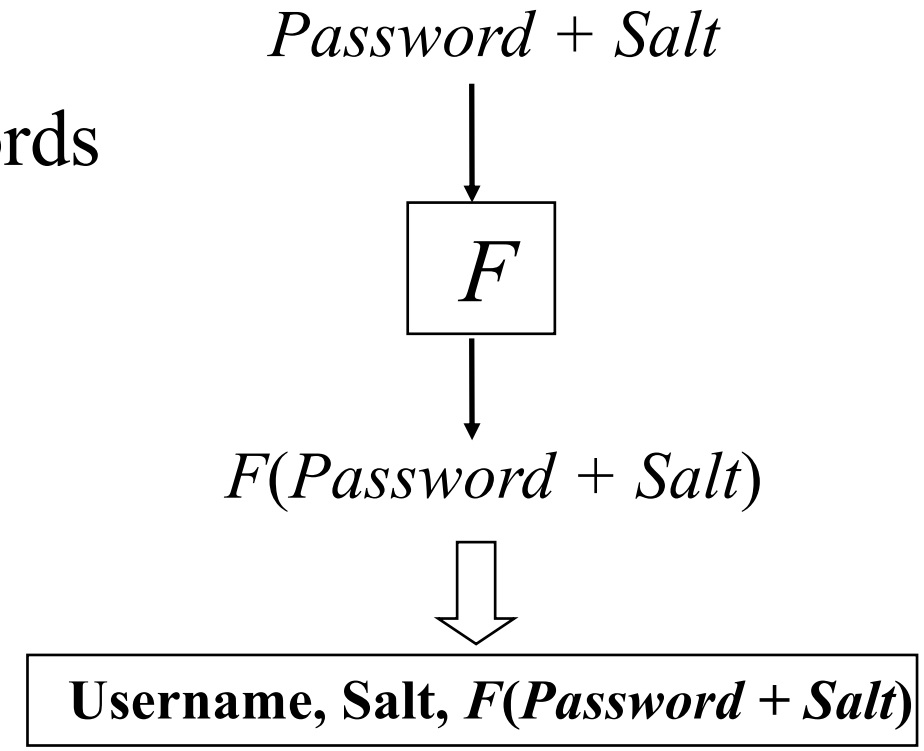
What is *F*?

- Crypt Algorithm (Unix)
  - Designed by Bob Morris and Ken Thompson
  - Use Data Encryption Standard (DES) encryption algorithm
  - User password and salt is used as the encryption key to encrypt a 64-bit block of zeros
  - The salt is 12 bits. The password takes 8 characters
  - This process is repeated 25 times



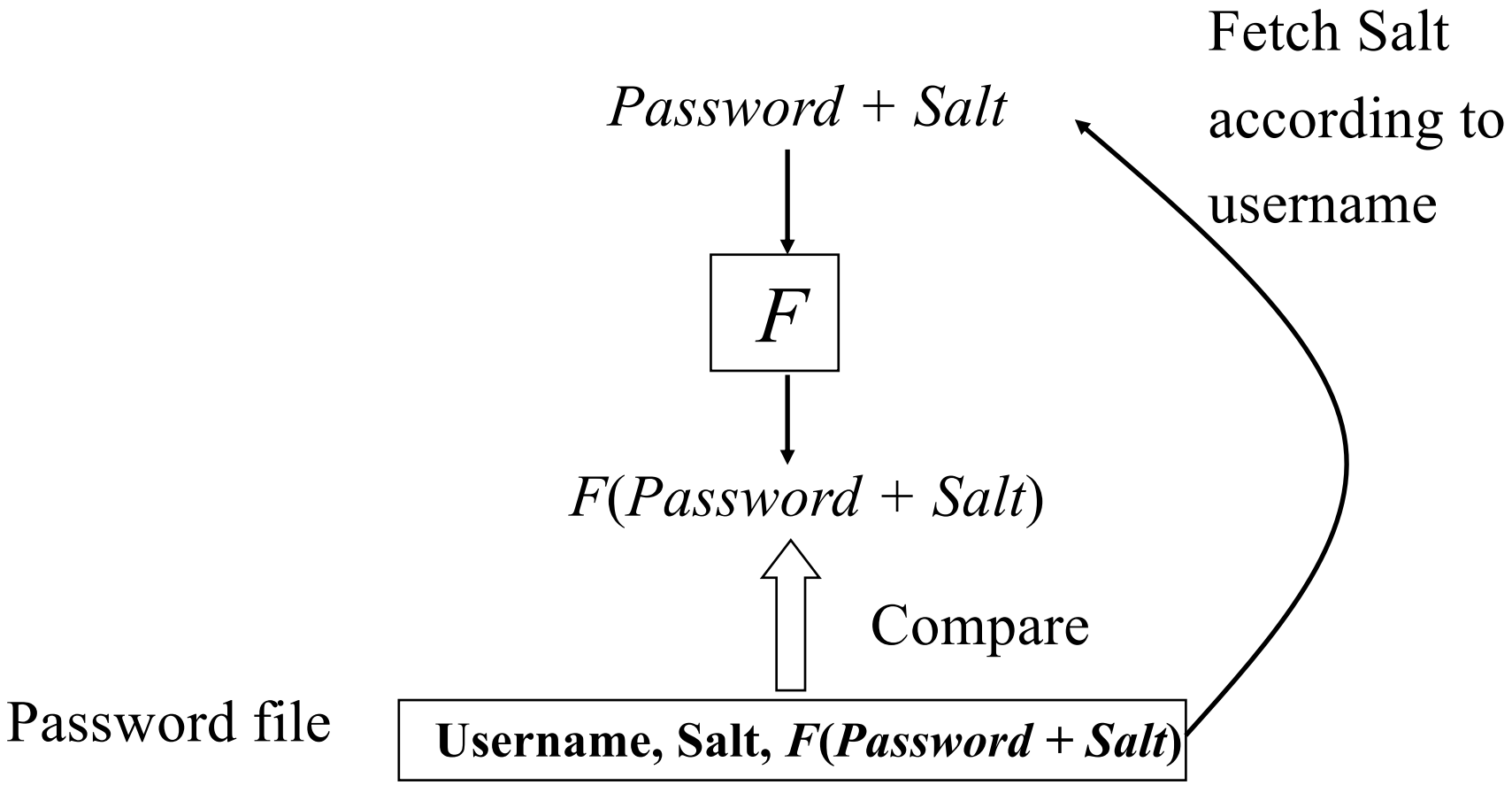
Password Salt

- To make the dictionary attack a bit more difficult
- Salt is a 12-bit number between 0 and 4095
- Derived from the system clock and the process identifier
- Storing the passwords



Password Salt (Cont'd)

- Verifying the passwords



UNIX Password Scheme

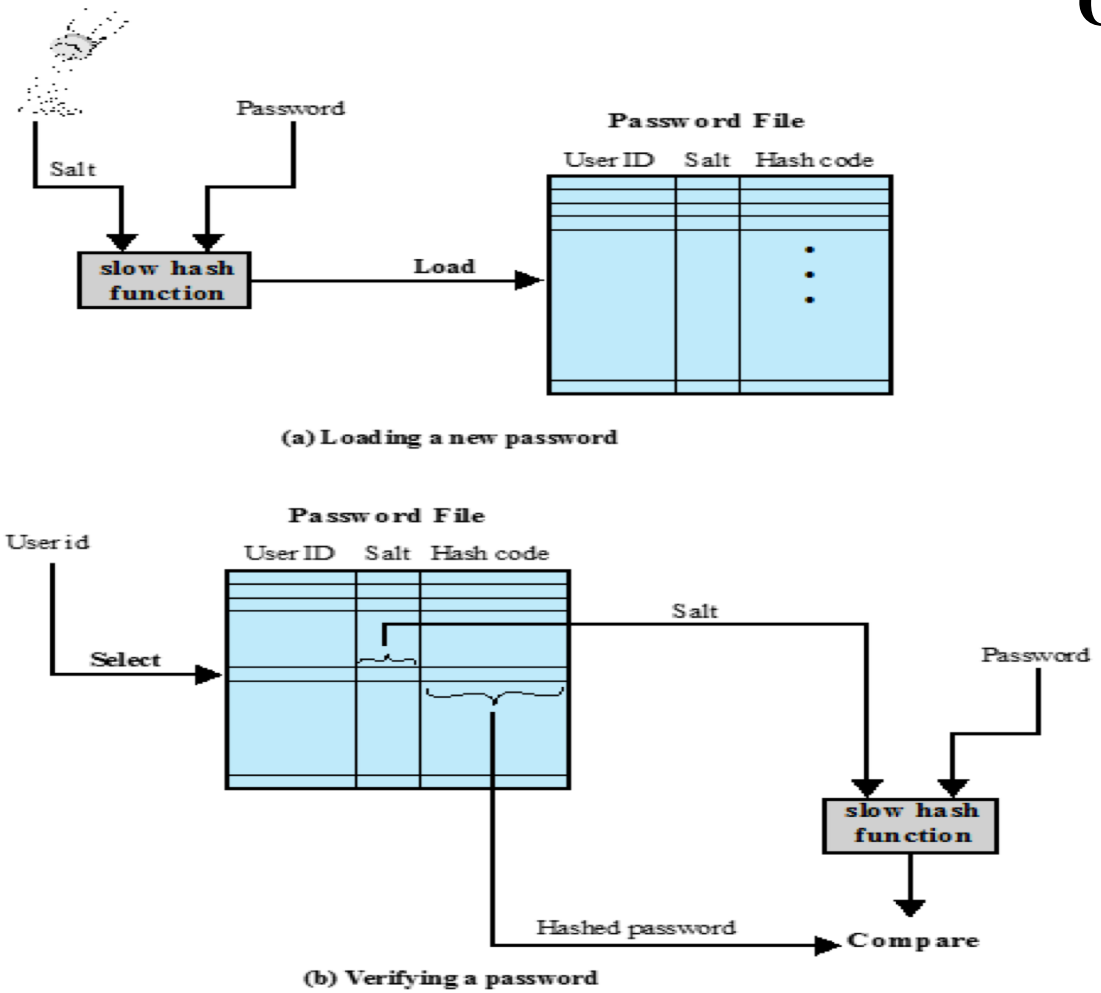


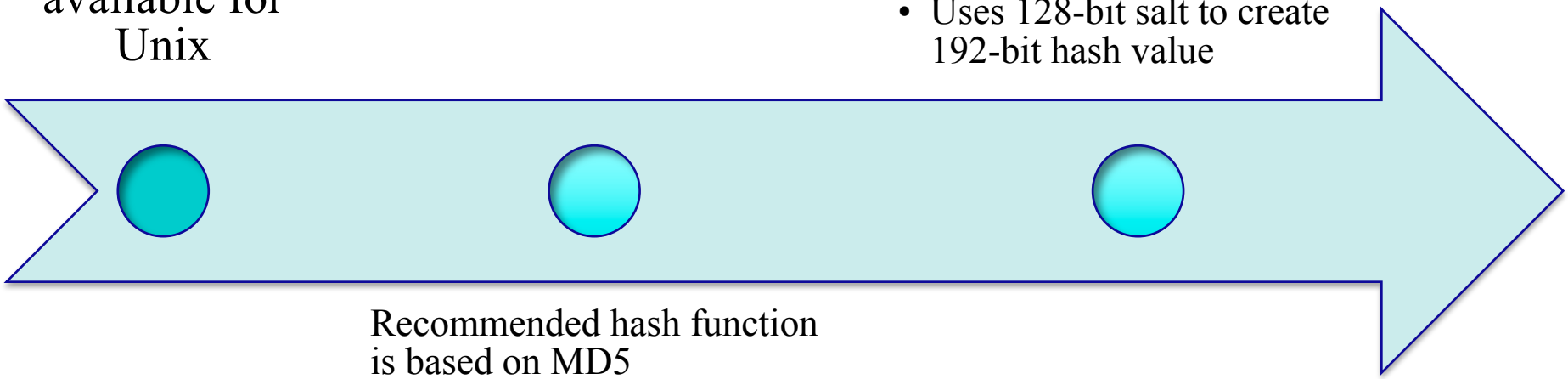
Figure 3.2 UNIX Password Scheme

Original scheme :

- Up to eight printable characters in length
- Use DES
  - 8*7bits =56 bits
  - 12-bit salt used to modify DES encryption into a one-way hash function
- A data input consisting of a 64-bit block of zeros. It repeatedly encrypts 25 times
- 64-bit output is translated to 11 character sequence

Improved Implementation

Much stronger hash/salt schemes available for Unix



Recommended hash function is based on MD5

- Salt of up to 48-bits
- Password length is unlimited
- Produces 128-bit hash
- Uses an inner loop with 1000 iterations to achieve slowdown

OpenBSD uses Blowfish block cipher based hash algorithm called Bcrypt

- Most secure version of Unix hash/salt scheme
- Uses 128-bit salt to create 192-bit hash value

A Little Bit Detail

- An entry in the password file

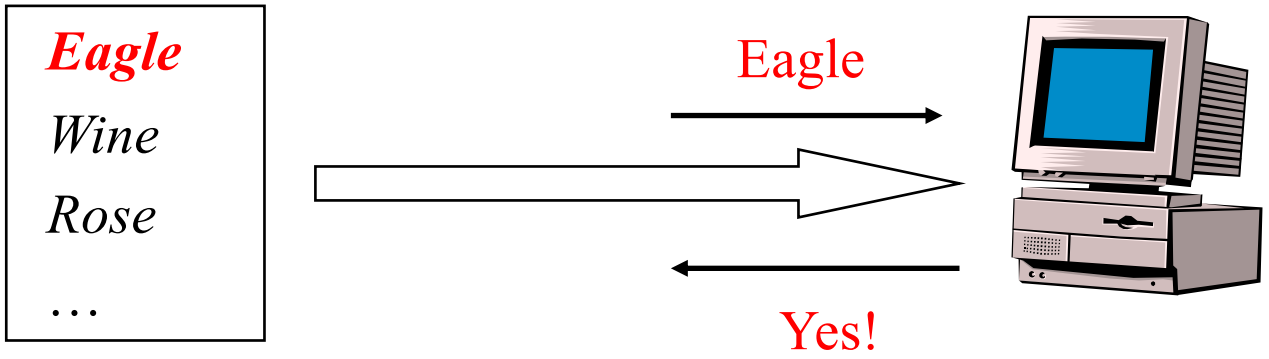
nomad:HrLNrZ3VS3TF2:501:100:Simple Nomad:/home/nomad:/bin/bash

- The salt controls the bit expansion in DES
  - Determine which bits to duplicate
  - Not a standard DES
- Shadowed password file

  nomad:***:501:100:Simple Nomad:/home/nomad:/bin/bash

Dictionary Attacks

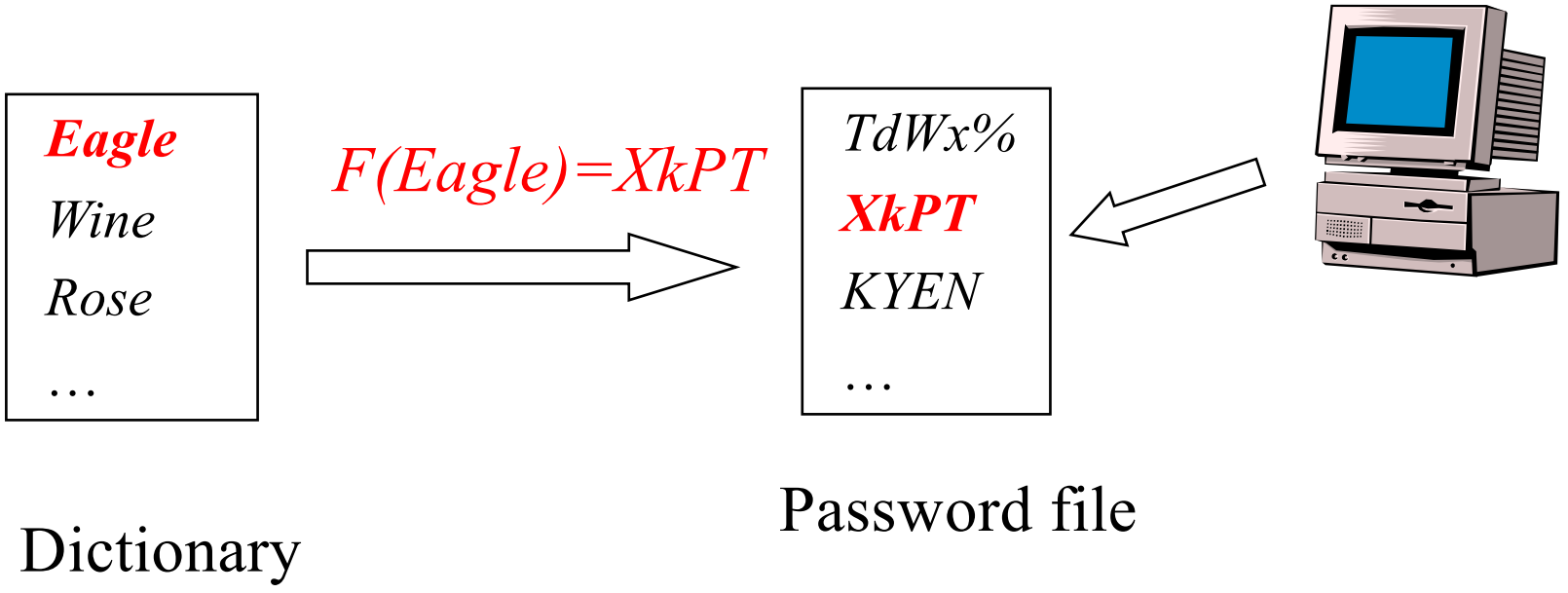
- Attack 1:
  - Create a dictionary of common words and names and their simple transformations
  - Use these to guess the password



Dictionary

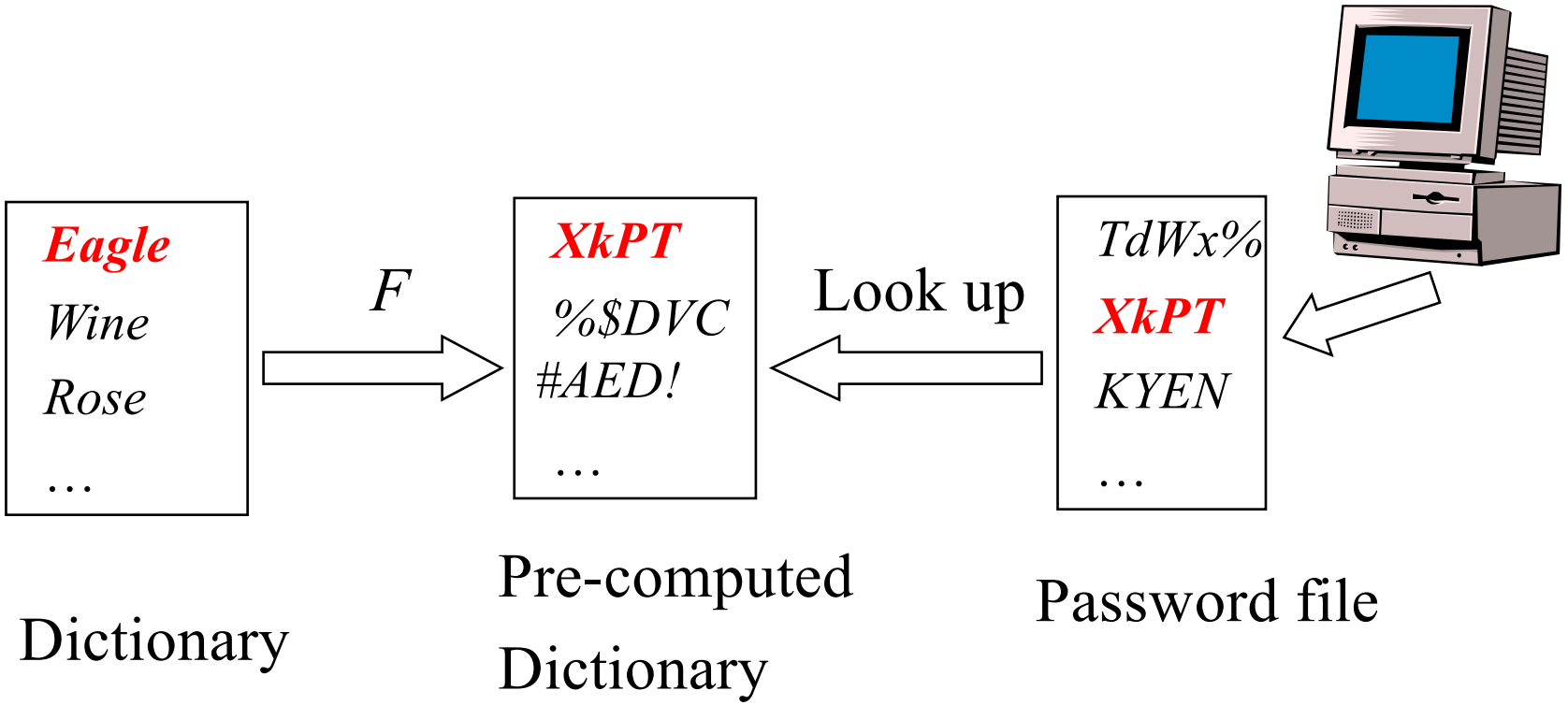
Dictionary Attacks (Cont'd)

- Attack 2:
  - Usually *F* is public and so is the password file
    - In Unix, *F* is crypt, and the password file is /etc/passwd.
  - Compute *F*(*word*) for each word in the dictionary
  - A match gives the password



Dictionary Attacks (Cont'd)

- Attack 3:
  - To speed up search, pre-compute *F*(dictionary)
  - A simple look up gives the password



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 1996
- Uses a combination of brute-force and dictionary techniques

Modern Approaches

- Complex password policy
  - Forcing users to pick stronger passwords
- However password-cracking techniques have also improved
  - The processing capacity available for password cracking has increased dramatically
  - The use of sophisticated algorithms to generate potential passwords
  - Studying examples and structures of actual passwords in use

Password File Access Control

Can block *offline guessing attacks* by denying access to encrypted passwords

Make available only to privileged users

Shadow password file

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 backup media**
- **Sniff passwords in network traffic**

Password Selection Strategies

User education

Users can be told the importance of using hard to guess passwords and can be provided with guidelines for selecting strong passwords

Computer generated passwords

Users have trouble remembering them

Reactive password checking

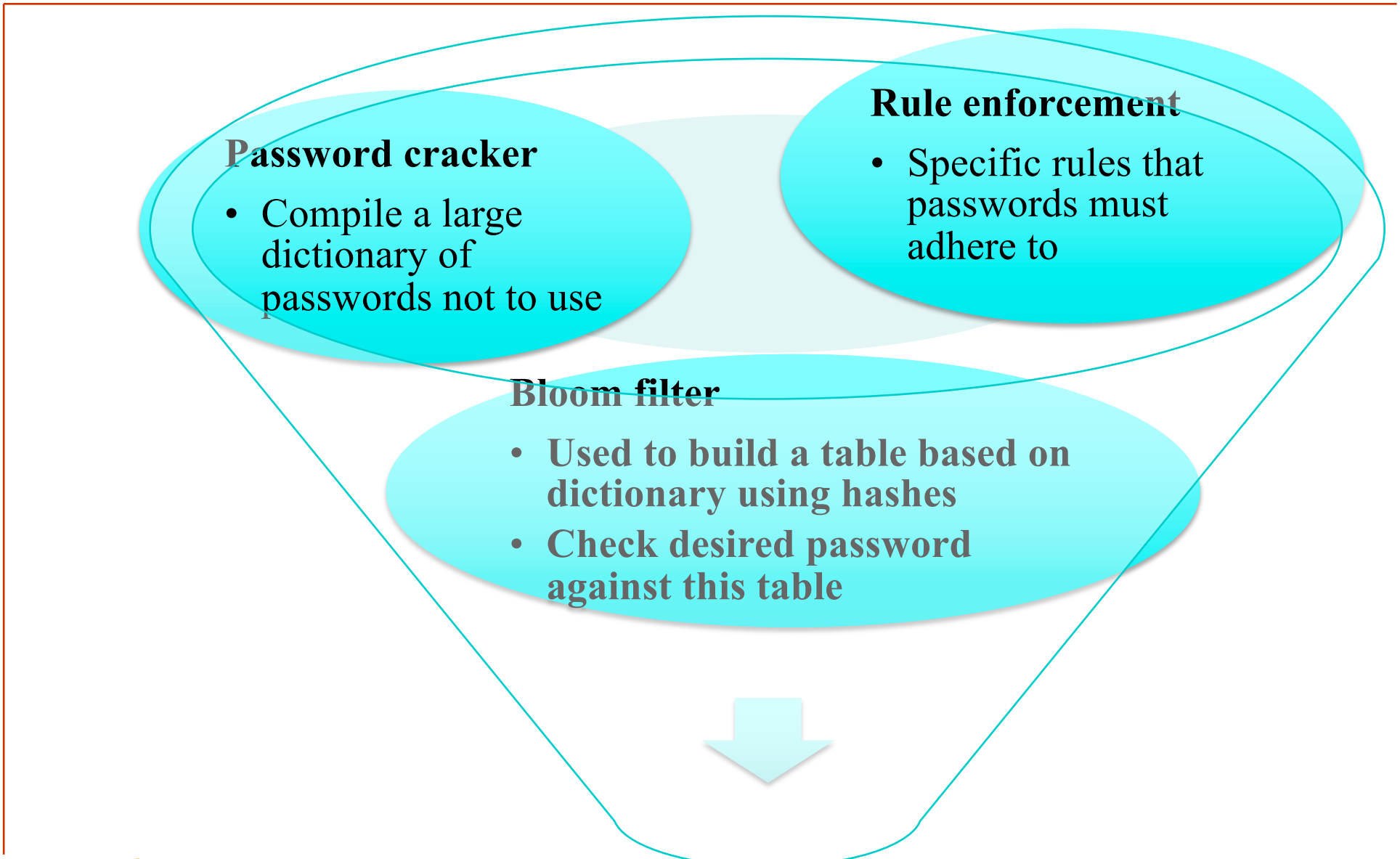
System periodically runs its own password cracker to find guessable passwords

Complex password policy

User is allowed to select their own password, however the system checks to see if the password is allowable, and if not, rejects it

Goal is to eliminate guessable passwords while allowing the user to select a password that is memorable

Proactive Password Checking

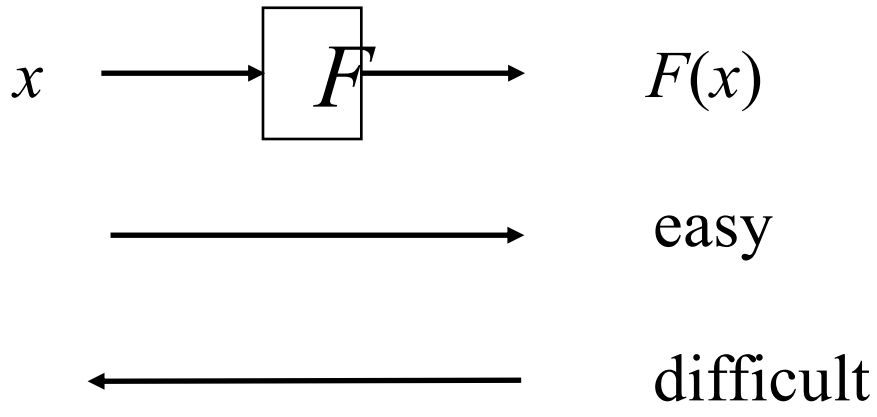


One-time Passwords

- Use the password exactly once!

Lamport's Scheme (S/Key)

- Take advantage of One-Way function
- One-way hash function F
  - F(x) is easy to compute
  - From F(x), x is difficult to compute



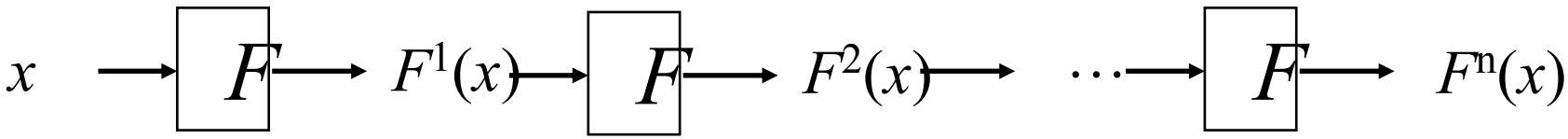
S/Key (Cont'd)

- Pre-computation

The System

1. Randomly generate x

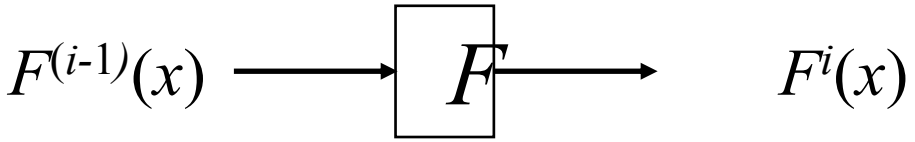
2. Compute the following



3. Save (*username*, $F^n(x)$), and give x to the user.

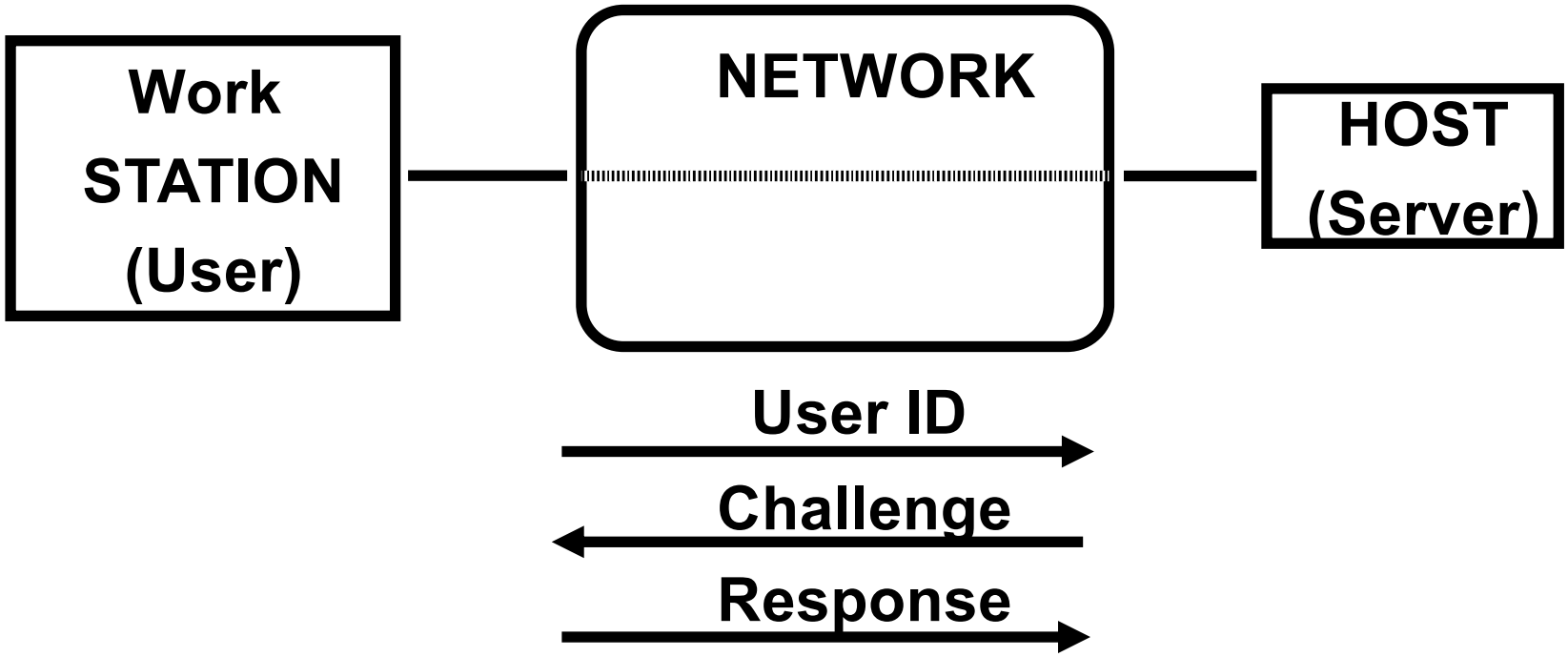
S/Key (Cont'd)

- Authentication
  - The first time, the user supplies $F^{(n-1)}(x)$.
  - The system checks if $F(F^{(n-1)}(x))=F^n(x)$. If yes, the user is authenticated and the system replaces $F^n(x)$ with $F^{(n-1)}(x)$.
  - The second time, the user supplies $F^{(n-2)}(x)$.
  - The third time, …

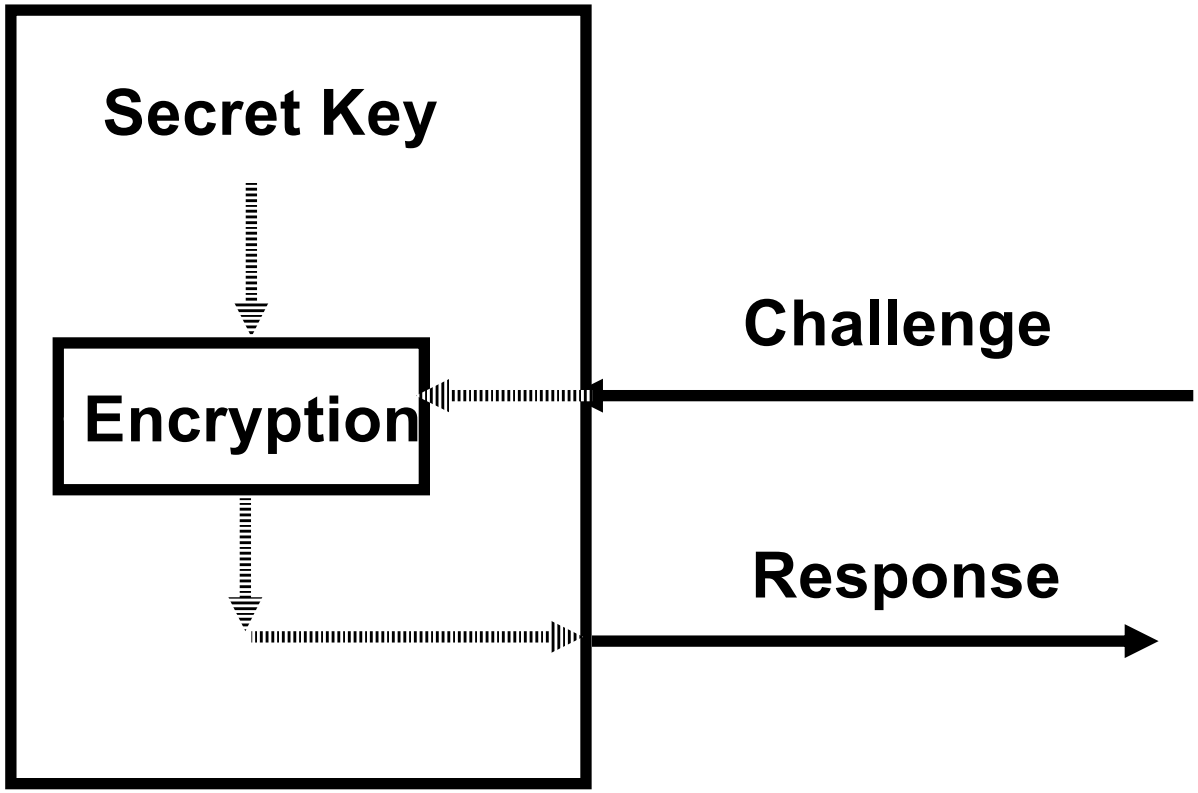


Challenge Response

- A non-repeating challenge from the host instead of a clock is used
- Note that the device requires a keypad.



Challenge Response (Cont'd)



Challenge Response (Cont'd)

- Problems with challenge/response schemes
  - Key database is extremely sensitive
  - This can be avoided if public key algorithms are used; however, the outputs would be too long for users to input conveniently

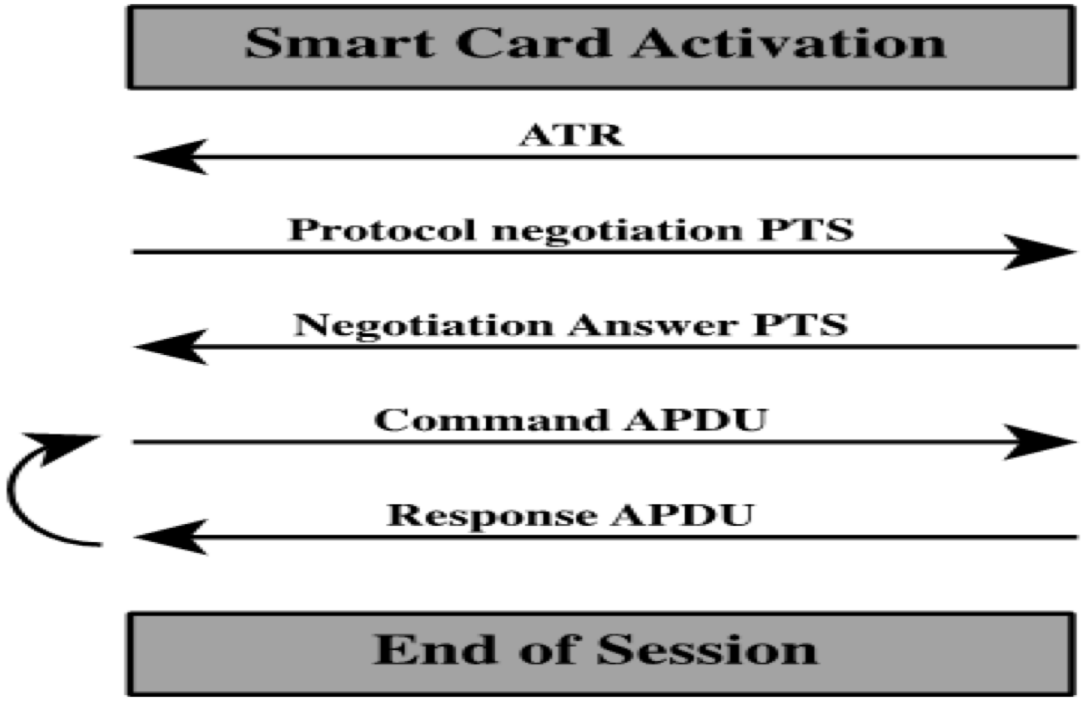
Table 3.2

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

Types of Cards Used as Tokens

Smart card

Card reader



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

Figure 3.5 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)

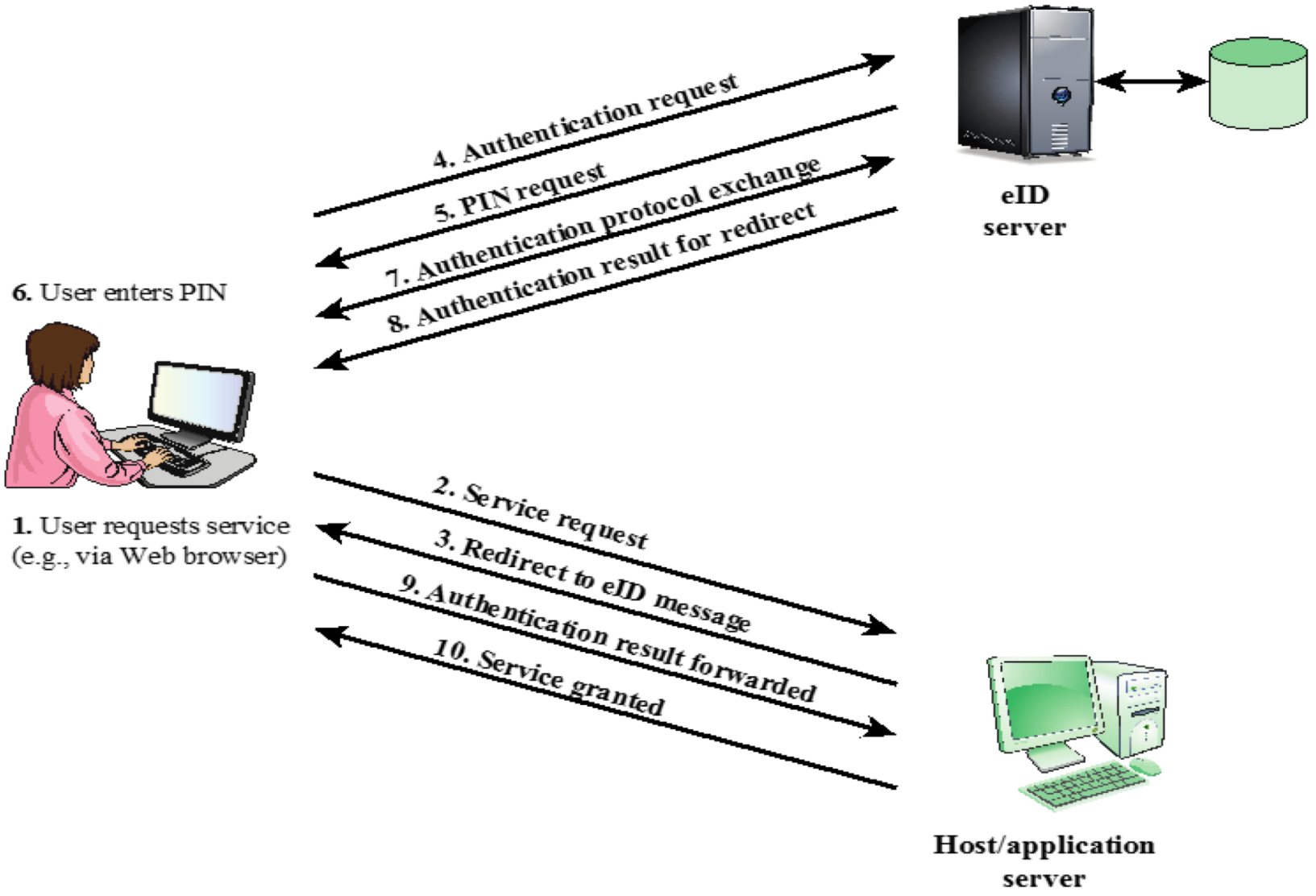


Figure 3.6 User Authentication with eID

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

Cost vs Accuracy

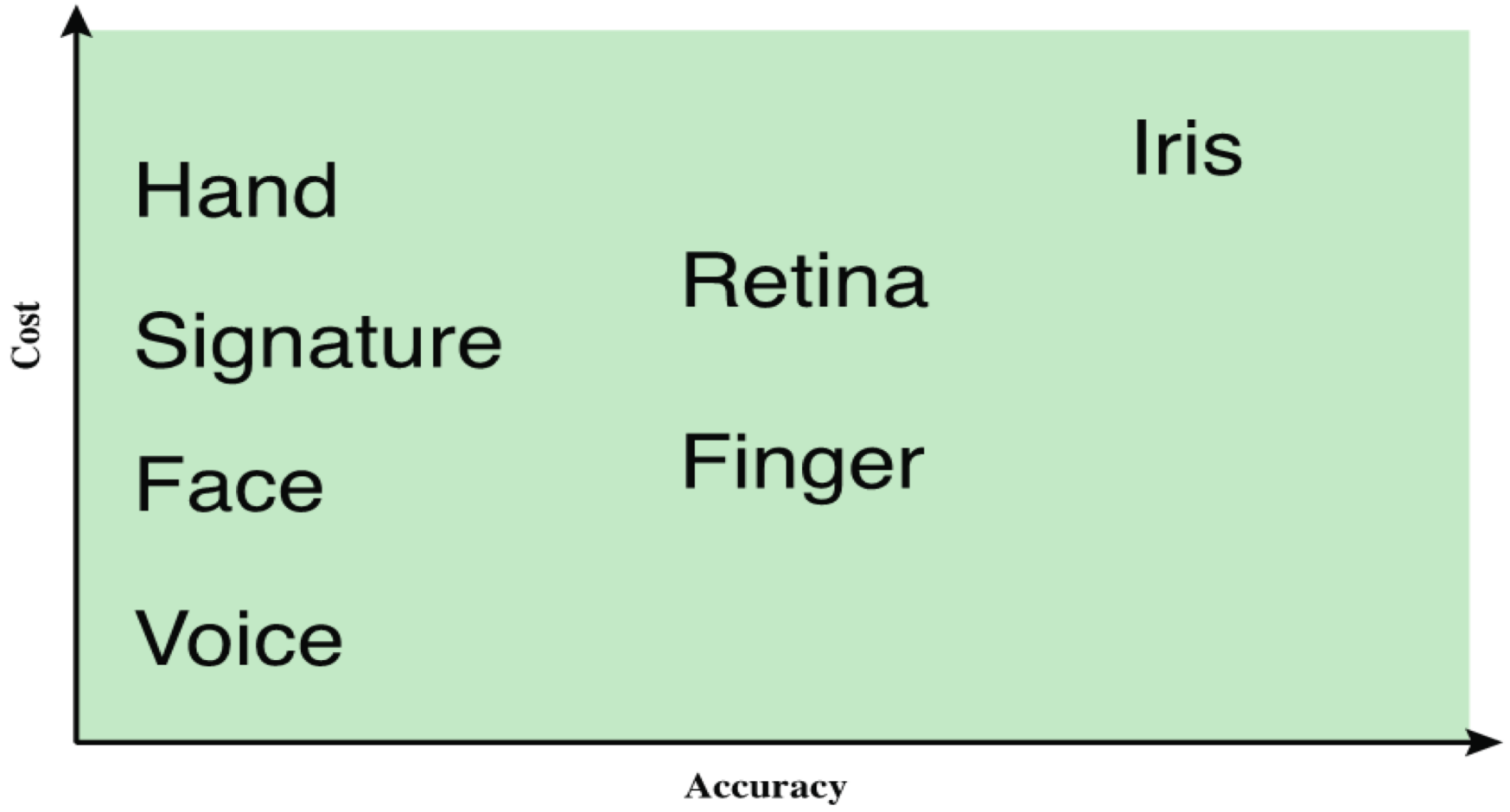


Figure 3.7 Cost Versus Accuracy of Various Biometric Characteristics in User Authentication Schemes.

Trade-off

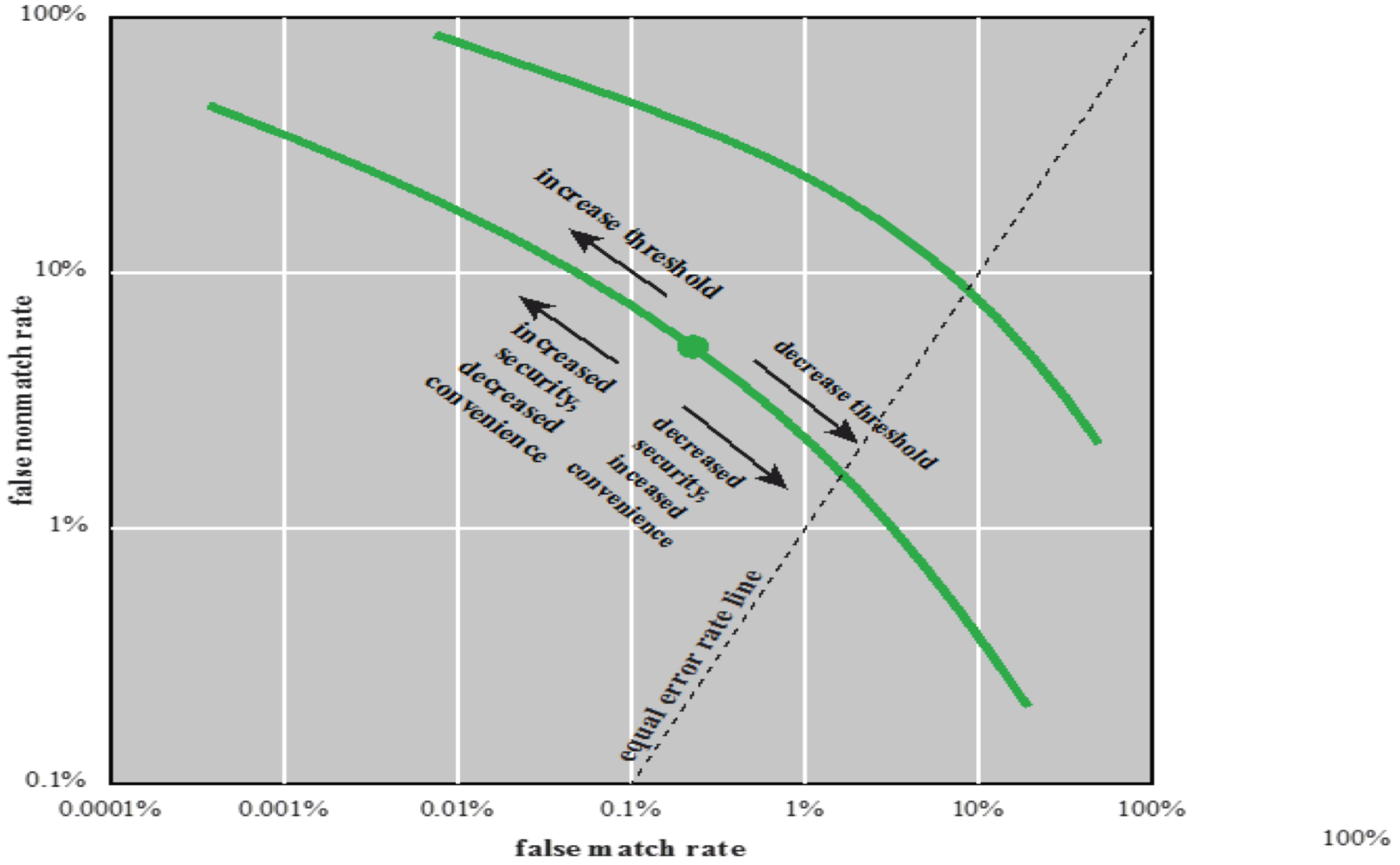


Figure 3.10 Idealized Biometric Measurement Operating Characteristic Curves (log-log scale)

Actual Biometric Measurement

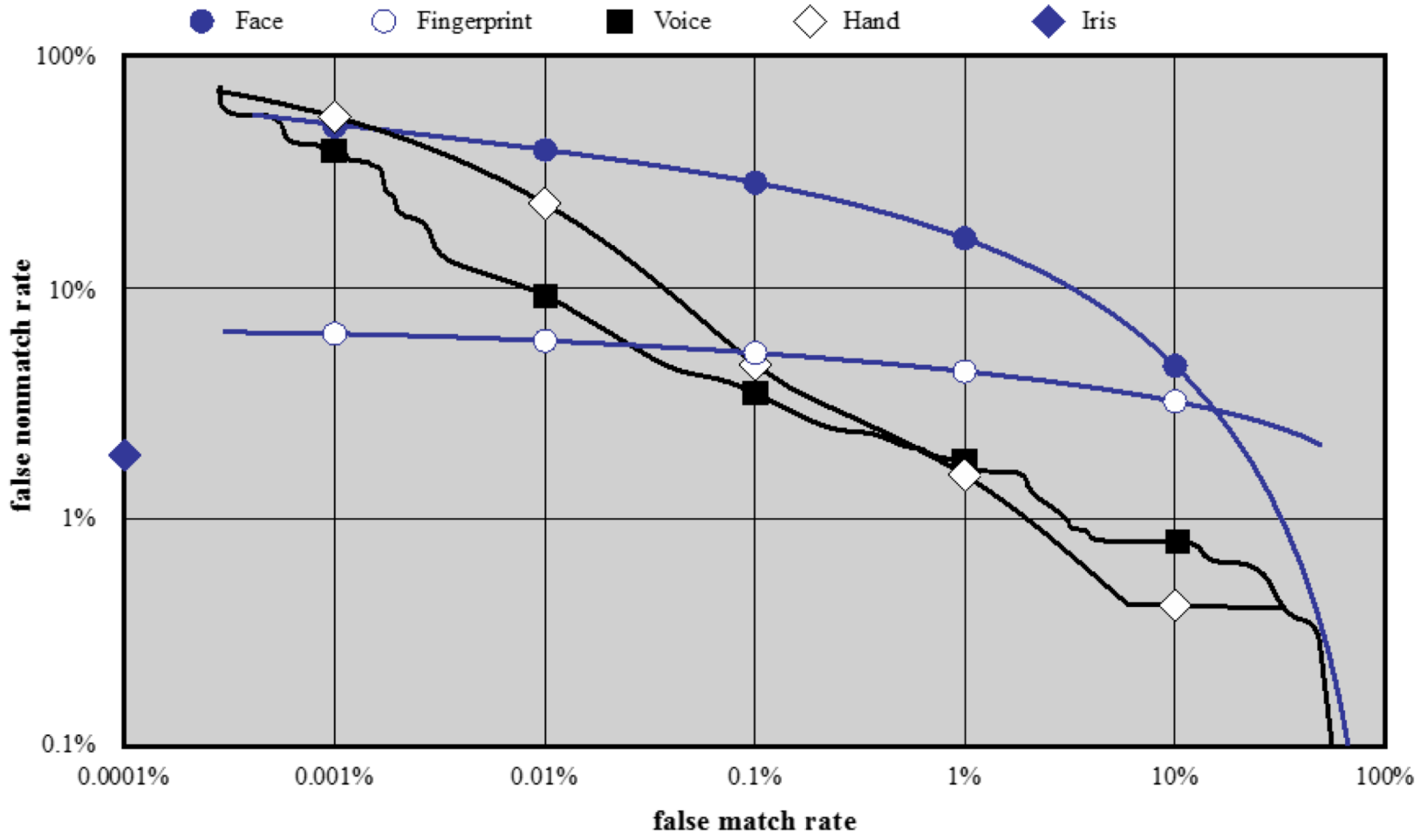
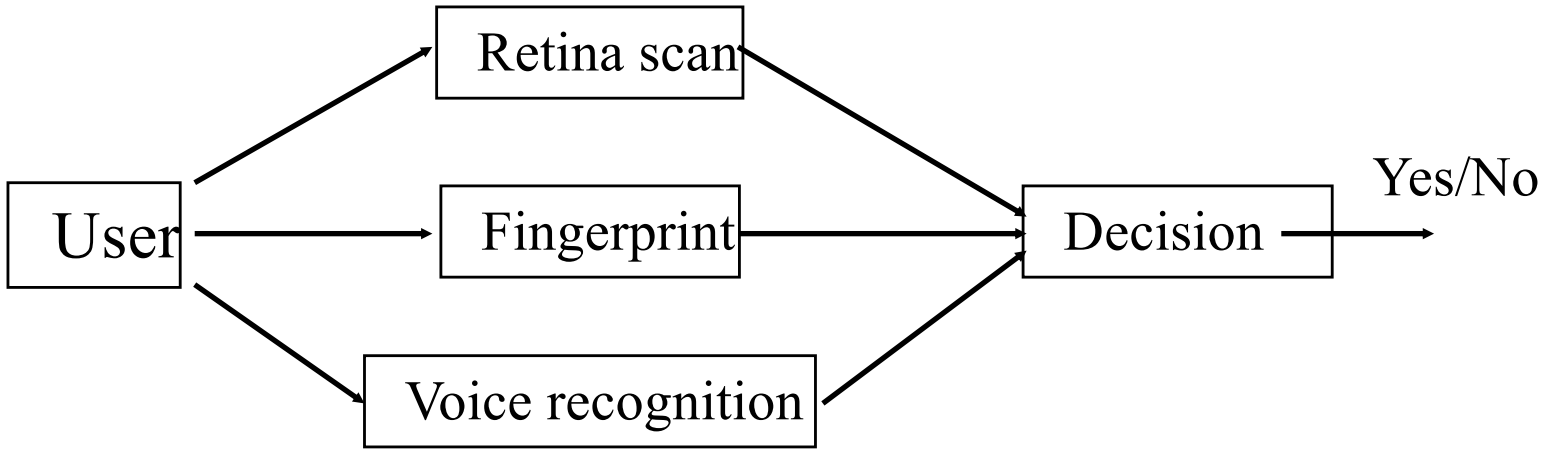


Figure 3.11 Actual Biometric Measurement Operating Characteristic Curves, reported in [MANS01]. To clarify differences among systems, a log-log scale is used.

Multimodal Biometrics

- Use multiple Biometrics together.
  - AND: Accept only when all are passed
    - Why do we need this?
  - OR: Accept as long as at least one is passed
    - Why do we need this?
  - Others



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 rely on some form of a challenge-response protocol to counter threats

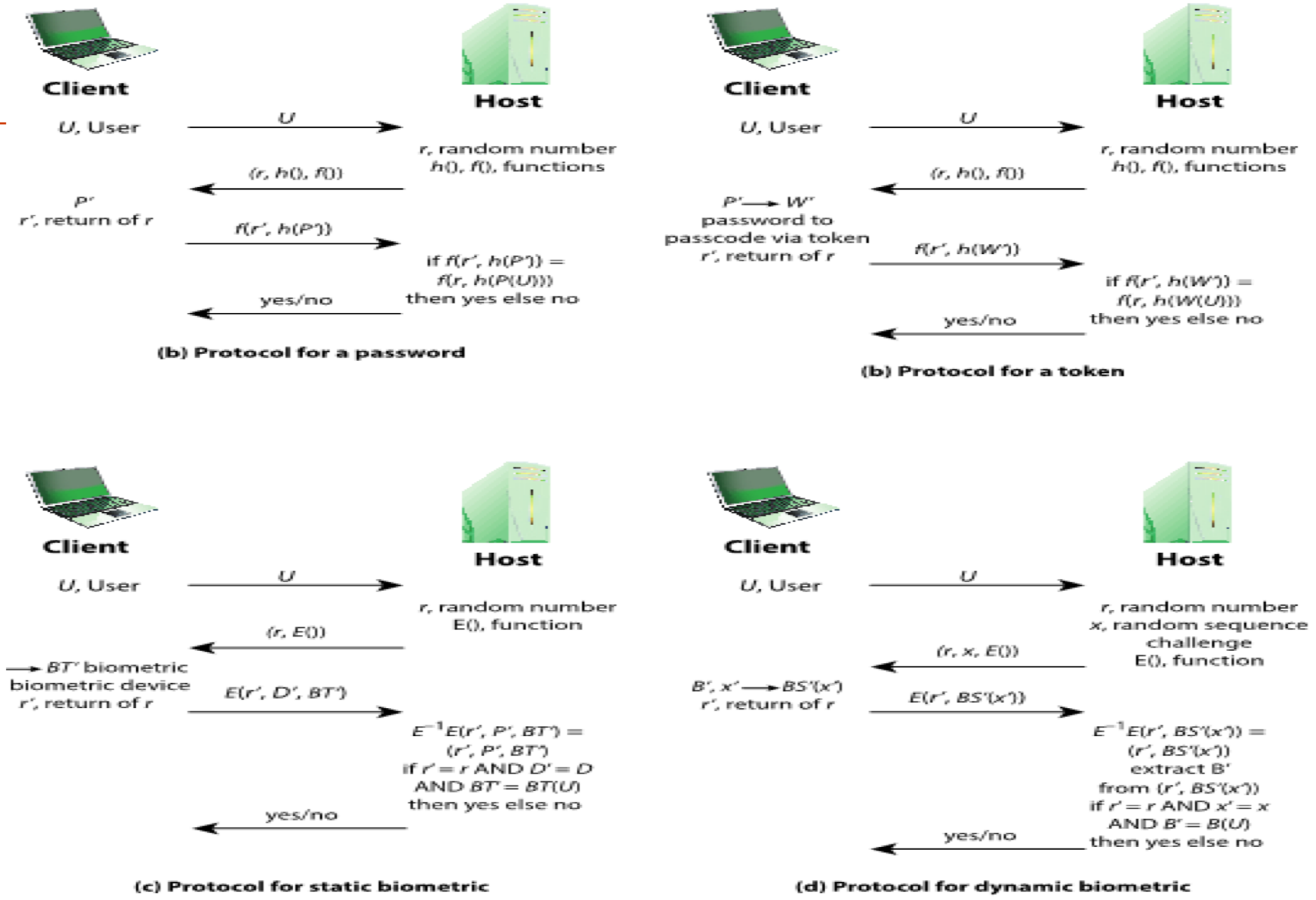
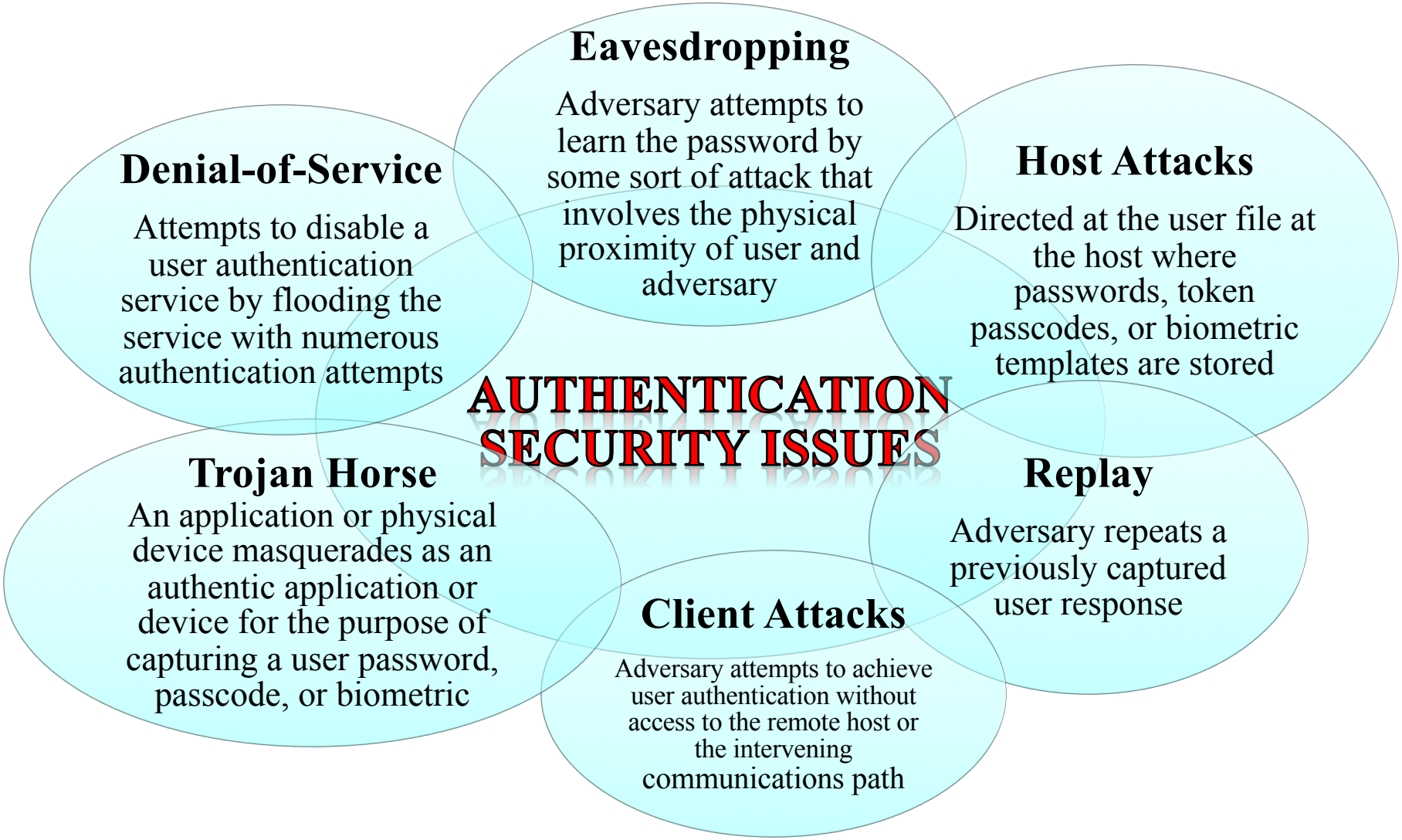


Figure 3.12 Basic Challenge-Response Protocols for Remote User Authentication

Table 3.4 Attacks vs 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	Password, token,	Lockout by multiple	Multifactor with token

Security Issues



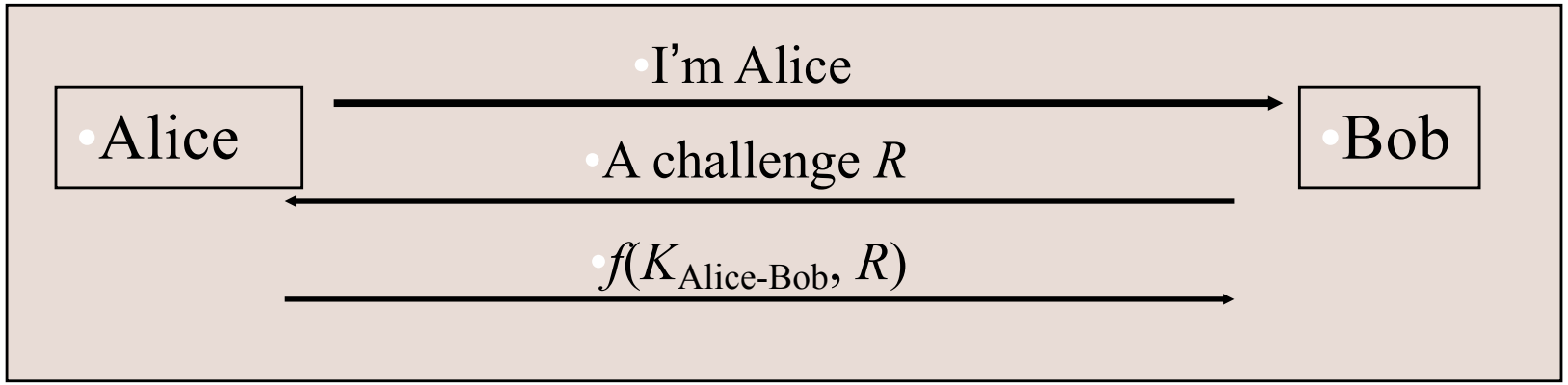
Authentication Mechanisms (Cont'd)

- Cryptographic authentication protocols
  - Basic idea:
    - A prover proves some information by performing a cryptographic operation on a quantity that the verifier supplies.
  - Usually reduced to the knowledge of a secret value
    - A symmetric key
    - The private key of a public/private key pair

Authentication Handshakes

- Secure communication almost always includes an initial authentication handshake.
  - Authenticate each other
  - Establish session keys
  - *This process is not trivial; flaws in this process undermines secure communication*
- This topic is about typical flaws

Authentication with Shared Secret



- Weaknesses
  - Authentication is not mutual; Trudy can convince Alice that she is Bob
  - Trudy can hijack the conversation after the initial exchange
  - If the shared key is derived from a password, Trudy can mount an off-line password guessing attack
  - Trudy may compromise Bob's database and later impersonate Alice

Authentication with Shared Secret (Cont'd)

- A variation
  - Requires reversible cryptography
  - Other variations are possible
- Weaknesses
  - All the previous weaknesses remain
  - Trudy doesn't have to see R to mount off-line password guessing if R has certain patterns (e.g., concatenated with a timestamp)
    - Trudy sends a message to Bob, pretending to be Alice

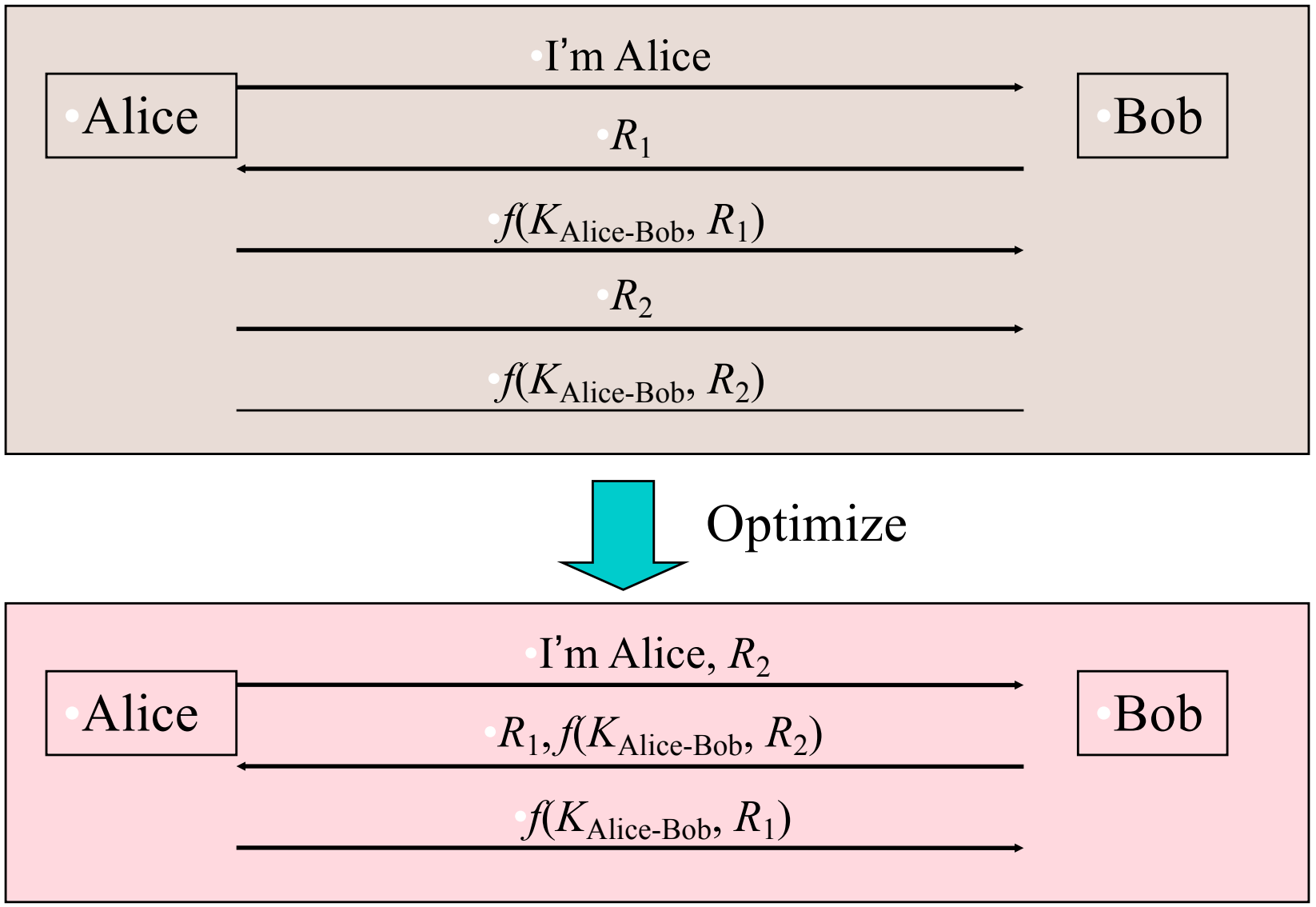
Authentication with Public Key

- Bob's database is less risky
- Weaknesses
  - Authentication is not mutual; Trudy can convince Alice that she is Bob
  - Trudy can hijack the conversation after the initial exchange
  - Trudy can trick Alice into signing something
    - Use different private key for authentication

Authentication with Public Key (Cont'd)

A variation

Mutual Authentication



Mutual Authentication (Cont'd)

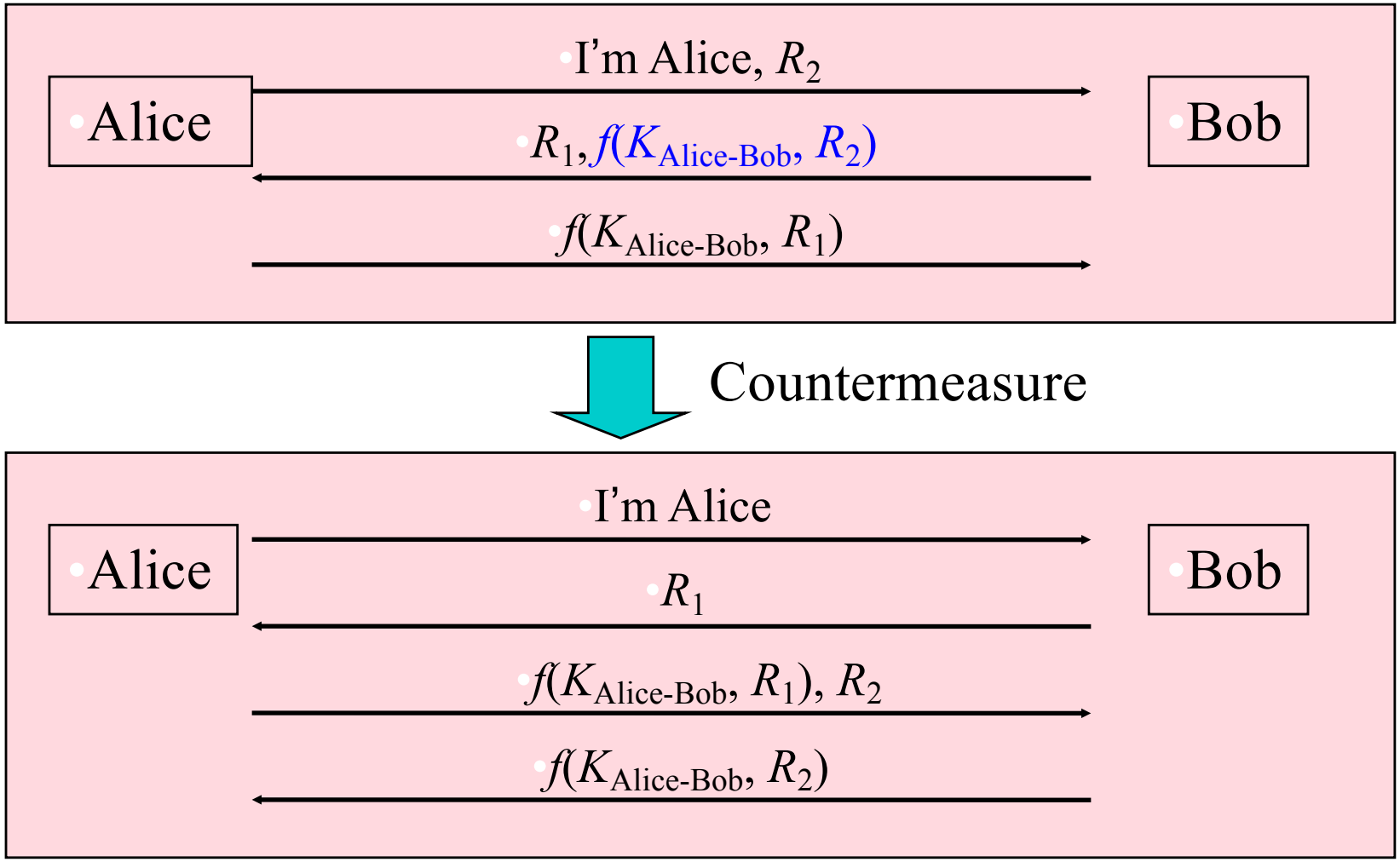
- Reflection attack

Reflection Attacks (Con'td)

- Lesson: Don't have Alice and Bob do exactly the same thing
  - Different keys
    - Totally different keys
    - $K_{Alice\text{-}Bob} = K_{Bob\text{-}Alice} + 1$
  - Different Challenges
  - The initiator should be the first to prove its identity
    - Assumption: initiator is more likely to be the bad guy

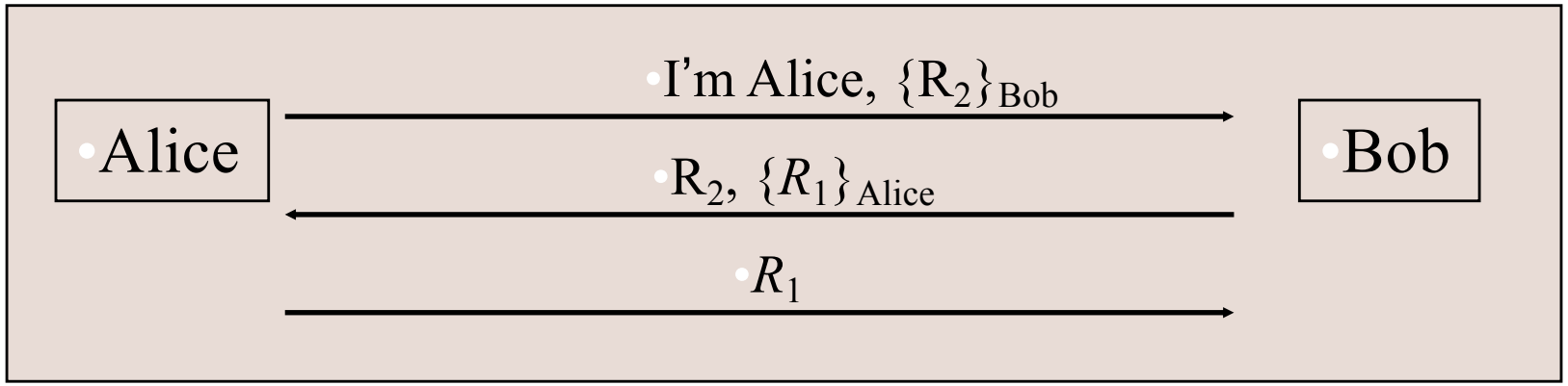
Mutual Authentication (Cont'd)

- Password guessing



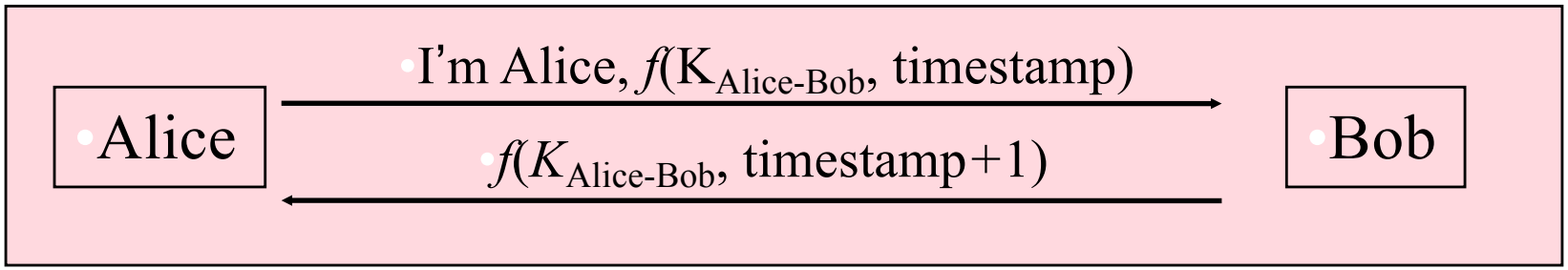
Mutual Authentication (Cont'd)

- Public keys
  - Authentication of public keys is a critical issue



Mutual Authentication (Cont'd)

- Mutual authentication with timestamps
  - Require synchronized clocks
  - Alice and Bob have to encrypt different timestamps



Integrity/Encryption for Data

- Communication after mutual authentication should be cryptographically protected as well
  - Require a session key established during mutual authentication

Session Keys

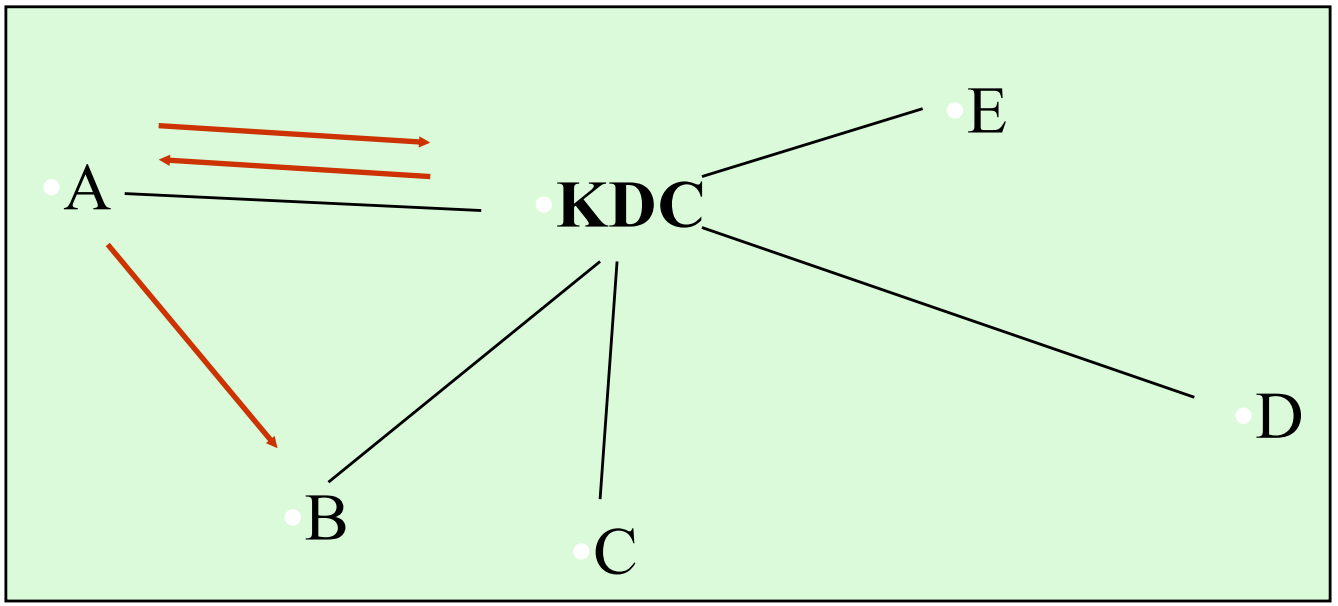
- More the same symmetric key used, more likely to be broken
- Generate and use a symmetric key for use during a specific communication for data only
- PK schemes good for encrypting random data; not good otherwise - especially if possible plaintext is from a small set
  - Forward search attack
  - Encrypt all possibilities with public key

Two-Way Public Key Based Authentication (Cont'd)

- A better approach
  - Alice and Bob establish the session key with Diffie-Hellman key exchange
  - Alice and Bob signs the quantity they send
  - Trudy can't learn anything about the session key even if she compromises both Alice and Bob

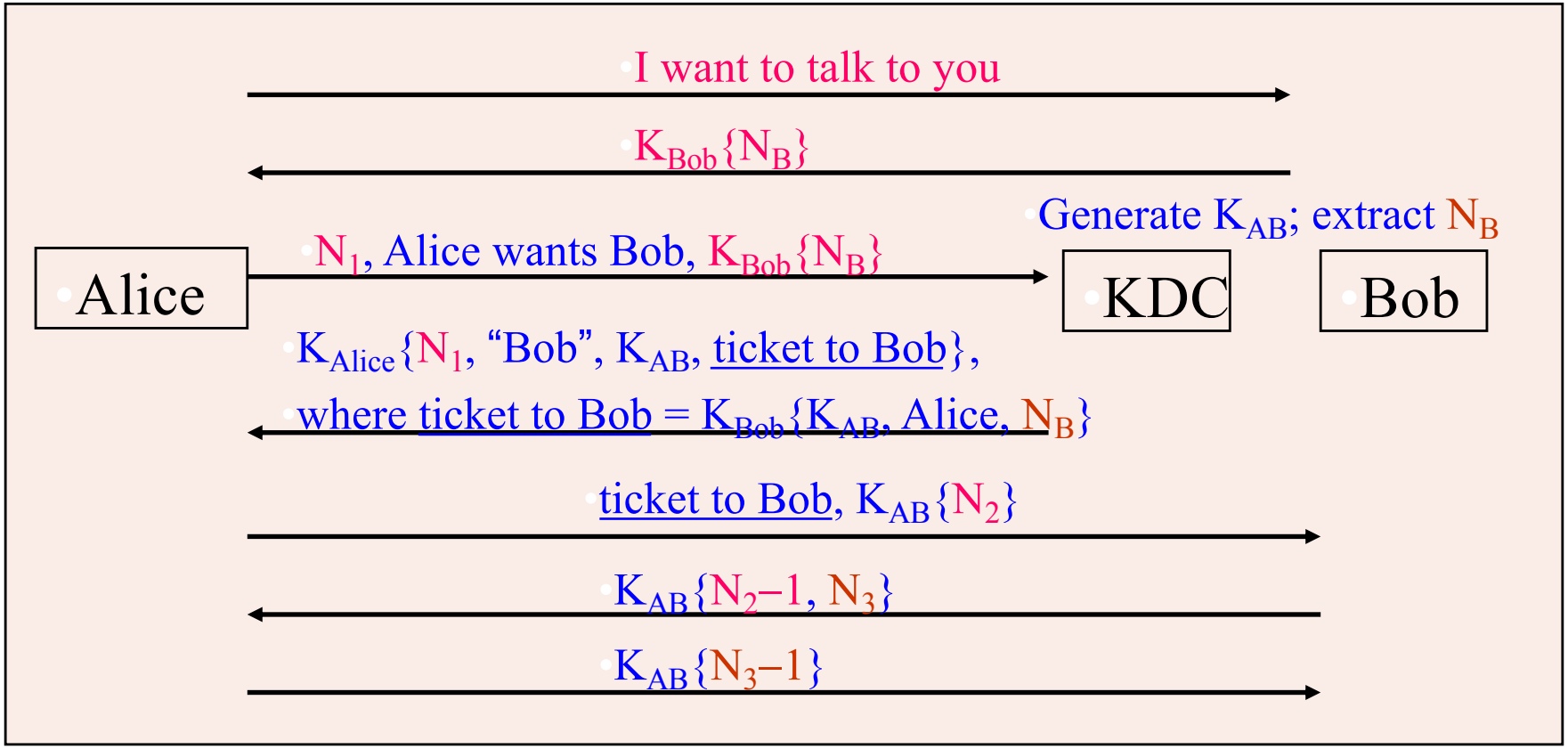
Trusted Intermediaries

- Problem: authentication for large networks
- Key Distribution Center (KDC)
  - Secret key cryptography



Disadvantages: high risk; single point of failure; performance bottleneck

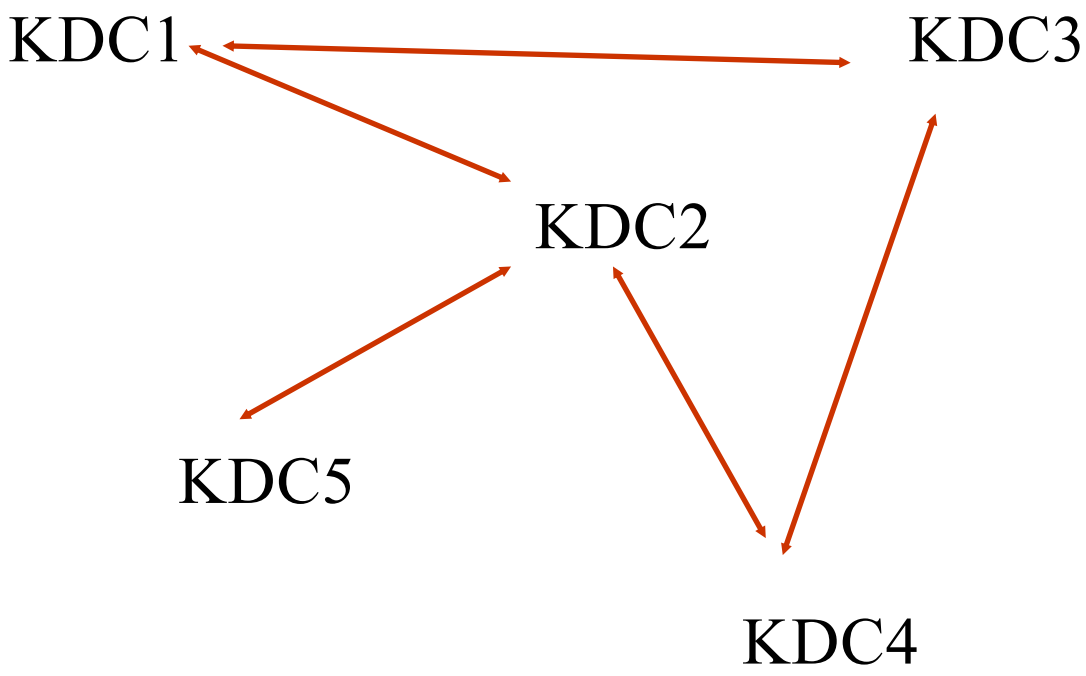
Expanded Needham-Schroeder Protocol



- **Classic protocol for authentication with KDC**
  - **Many others have been modeled after it (e.g., Kerberos)**
- **Nonce: A number that is used only once**
  - **Deal with replay attacks**
- **The additional two messages assure Bob that the initiator has talked to KDC since Bob generates N_B**

Multiple Trusted Intermediaries

- Multiple KDC domains
  - KDCs share keys between each other



Trusted Intermediaries

- Certification Authorities (CAs)
  - Public key cryptography
- Certificates
  - Signed messages that specify an identity and the corresponding public key
  - Signed with the well-known public key of a CA

CAs (Cont'd)

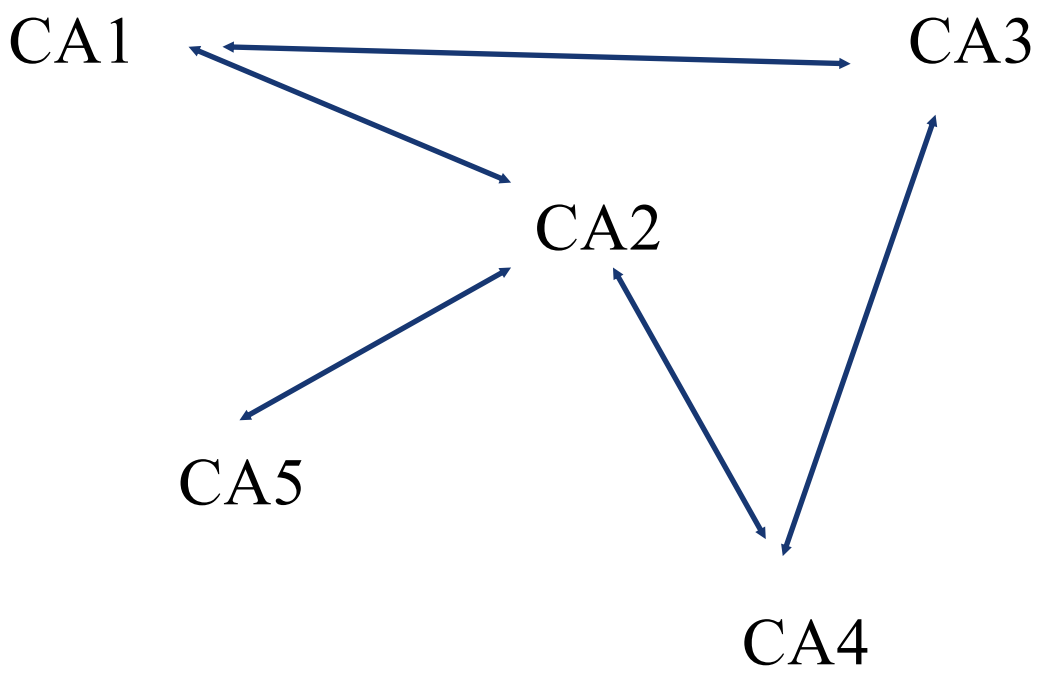
- Advantages
  - Doesn't have to be online
  - Lower risk compared with KDCs
  - Allow the network to operate even if CAs crash
  - Certificates can be public
  - A compromised CA cannot decrypt previously secured traffic

CAs (Cont'd)

- Certificate revocation
  - Problem: how to deal with revoked certificates (before they expire)
  - Certificate Revocation List (CRL)
    - List of revoked certificates
  - Timely and reliable distribution of CRLs is a critical and difficult problem.

Multiple Trusted Intermediaries (Cont'd)

- Multiple CA domains
  - CAs issue certificates to each other



Summary

- **Electronic user authentication principles**
  - A model for electronic user authentication
  - Means of authentication
  - Risk assessment for user authentication
- **Password-based authentication**
  - The vulnerability of passwords
  - The use of hashed passwords
  - Password cracking of user-chosen passwords
  - Password file access control
  - Password selection strategies
- **Token-based authentication**
  - Memory cards
  - Smart cards
  - Electronic identity cards
- **Biometric authentication**
  - Physical characteristics used in biometric applications
  - Operation of a biometric authentication system
  - Biometric accuracy
- **Remote user authentication**
  - Password protocol
  - Token protocol
  - Static biometric protocol
  - Dynamic biometric protocol
- **Security issues for user authentication**
- **One-time passwords**
  - S/Key
  - Time synchronized
  - Challenge response