Introduction to Computer Security

Chapter 3: User Authentication

Chi-Yu Li (2019 Spring)
Computer Science Department
National Chiao Tung University

User Authentication

- Fundamental building block and primary line of defense
- Basis for access control and user accountability

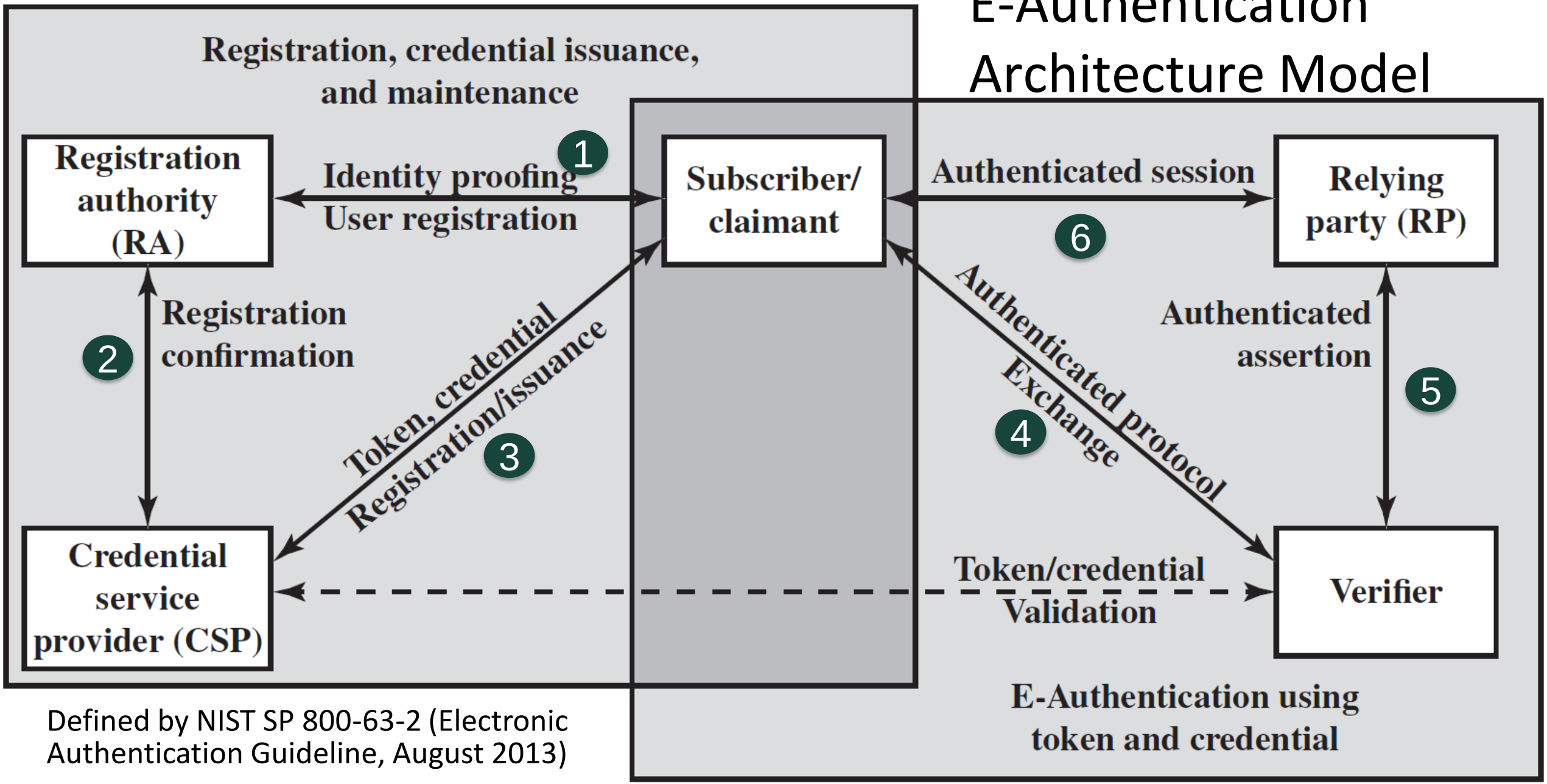
Definition of User Authentication (RFC 4949)

- The process of verifying an identity claimed by or for a system entity
- Consists of 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

Outline

- Electronic User Authentication Model
- Password-based Authentication
- Token-based Authentication
- Biometric Authentication
- Remote user Authentication
- Security Issues for User Authentication

E-Authentication Architecture Model



Defined by NIST SP 800-63-2 (Electronic Authentication Guideline, August 2013)

Cornerstone: Credential and Token

- Credential
  - Paper credential: documents that attest to the identity
    - e.g., passports, driver's licenses, and student ID cards
  - E-authentication credential: an object or data structure
    - Authoritatively binds an identity (via an identifier) and (optionally) additional attributes,
    - To at least one token (or authenticator) possessed and controlled by a subscriber

Mere possession of a valid electronic credential is ***rarely*** a sufficient basis for successful authentication

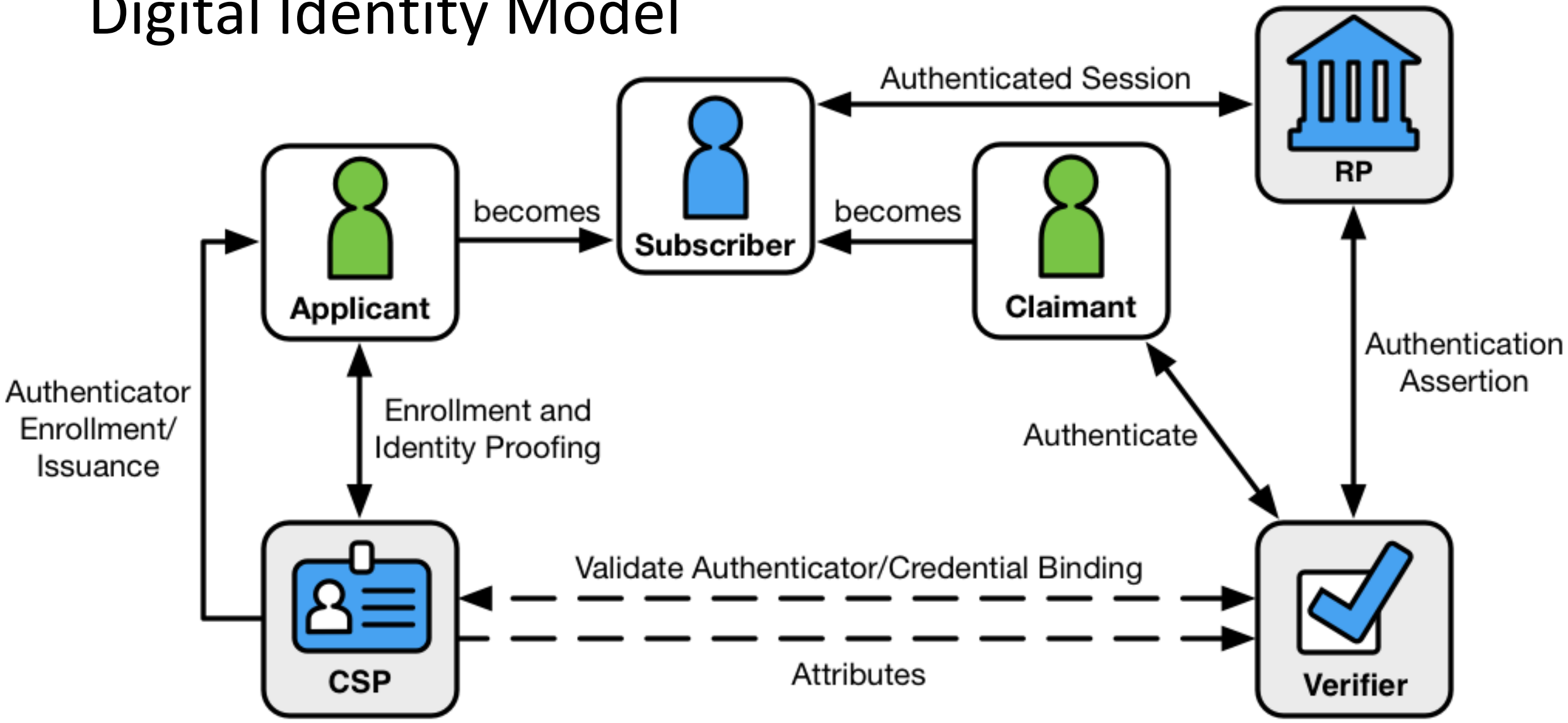
    - e.g., password database entries (possessed by the Verifier), X.509 public key certificates (possessed by the Claimant)

Cornerstone: Credential and Token (Cont.)

- Token: something that the Claimant possess and controls is used to authenticate the Claimant's identity
  - Typically a cryptographic module or password
  - Also named as authenticator

- In other words, authentication establishes confidence that
  - The Claimant has possession of an authenticator(s) bound to the credential, and (optional) the attribute values of the subscriber
    - Attribute values: s(he) is a Taiwan citizen, or a student at NCTU

Digital Identity Model



Defined by NIST SP 800-63-3 (Digital Identity Guidelines, Jun 2017)

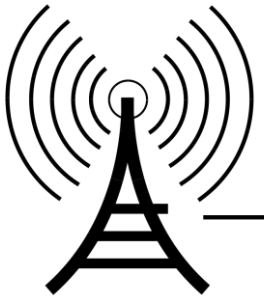
Example: Do Cellular Services Follow the E-Auth?

U.S. Carriers

at&t

verizon

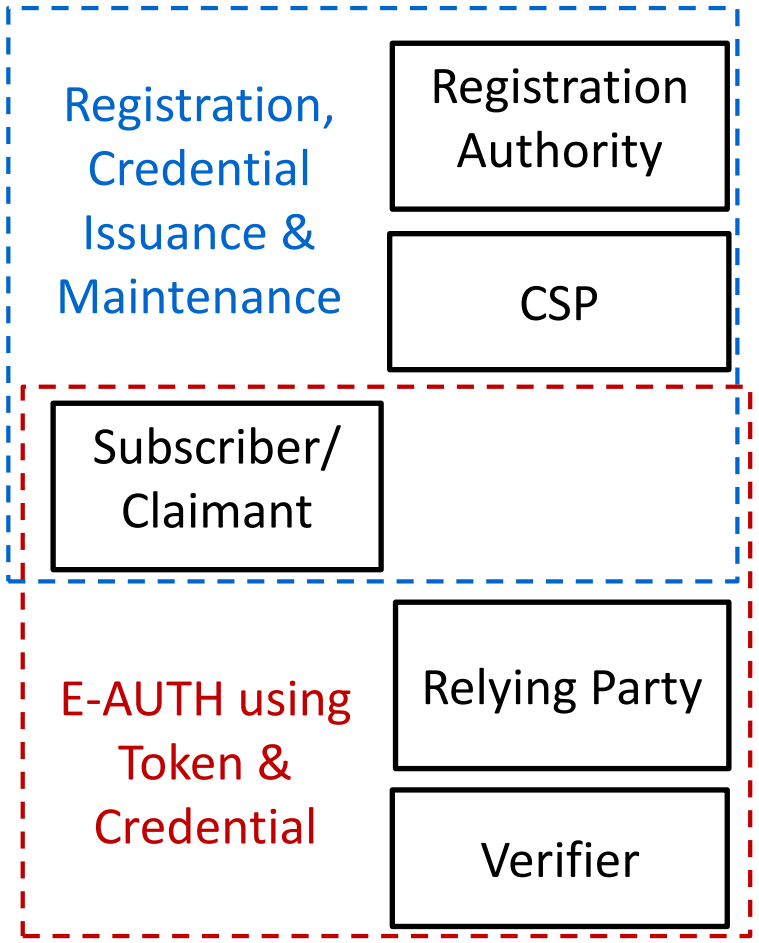
Sprint



Base Station

4G LTE Core Network

Role for Each Entity



at&t Store

4G LTE Core Network

CSP: Credential Service Provider

Tokens (Authenticators)

- Something the individual knows
  - e.g., password, answers to prearranged questions
- Something the individual possesses
  - e.g., electronic keycards, smart cards
- Something the individual is (static biometrics)
  - e.g., fingerprint, retina, face
- Something the individual does (dynamic biometrics)
  - e.g., voice pattern, handwriting

Password-based Authentication

- Widely used line of defense against intruders
  - User provides name or identifier (ID) and password
  - System compares password with the stored one
    - A password file indexed by user ID: store usernames or hash values of passwords
- The user ID
  - Determines that the user is authorized to access the system
  - Determines the user's privileges
  - Used in discretionary access control

Local Login

登入 Facebook

電子郵件或電話號碼

密碼

登入

Remote Login

Attacks and Countermeasures

- Offline dictionary attack
  - Obtain the system's password file (passwords stored in hash values)
    - CVE cases: https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=password
  - Search for valid passwords with hash values of commonly used passwords
    - Tools: http://www.openwall.com/john/, http://project-rainbowcrack.com/
  - Countermeasure: prevent unauthorized access to the password file, intrusion detection measures to identify a compromise, etc.
- Specific account attack
  - Submit password guesses until the correct password is discovered or the account Is blocked (more than 5 failure times)

Attacks and Countermeasures (Cont.)

- Popular password attack
  - Try popular passwords, e.g., 123456, against a wide range of user IDs
    - Assume that adversary obtains user IDs in advance
  - Countermeasure: inhibiting the selection by users of common passwords, scanning the IP addresses of auth requests and client cookies for submission patterns
- Password guessing against single user
  - Gain knowledge about the account holder and system password policies, and uses that knowledge to guess the password
  - Countermeasure: enforcement of password policies that make passwords difficult to guess (e.g., minimum length of the password)

Attacks and Countermeasures(Cont.)

- Workstation hijacking
  - Wait until a logged-in workstation is unattended
  - Countermeasure: automatically logging the workstation out after a period of inactivity

- Exploiting user mistakes
  - Mistakes: write it down, share it via any ways, keep preconfigured passwords, etc.
  - Countermeasure: user training, intrusion detection, simpler passwords combined with another auth. mechanism, etc.

Attacks and Countermeasures(Cont.)

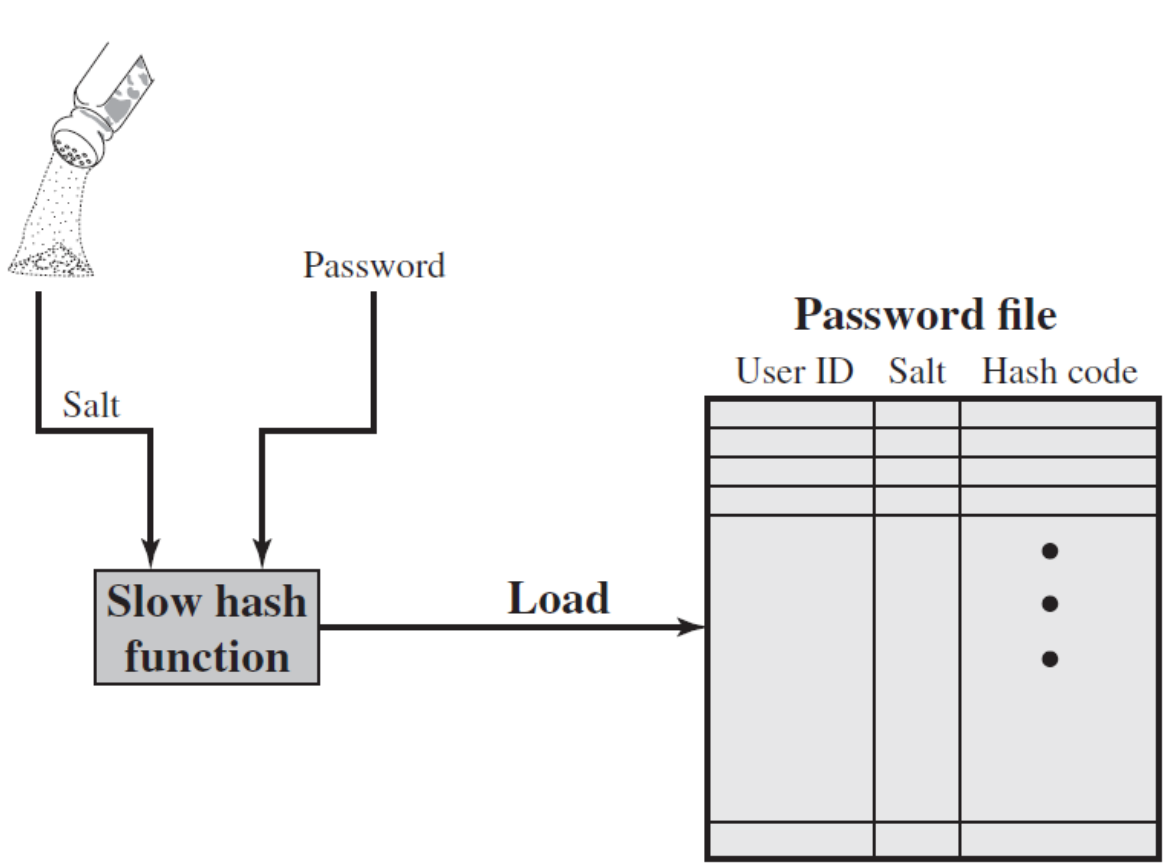
- Exploiting multiple password uses
  - Different network devices share the same or similar password for a given user
  - Countermeasure: a policy that forbids it
- Password sniffing/phishing
  - Passwords are transmitted without encryption, e.g., http or ftp
  - Phishing web pages
  - Countermeasure: encryption, inputting passwords with trusted devices and environments, etc.

Still: Most Commonly Used User Authentication

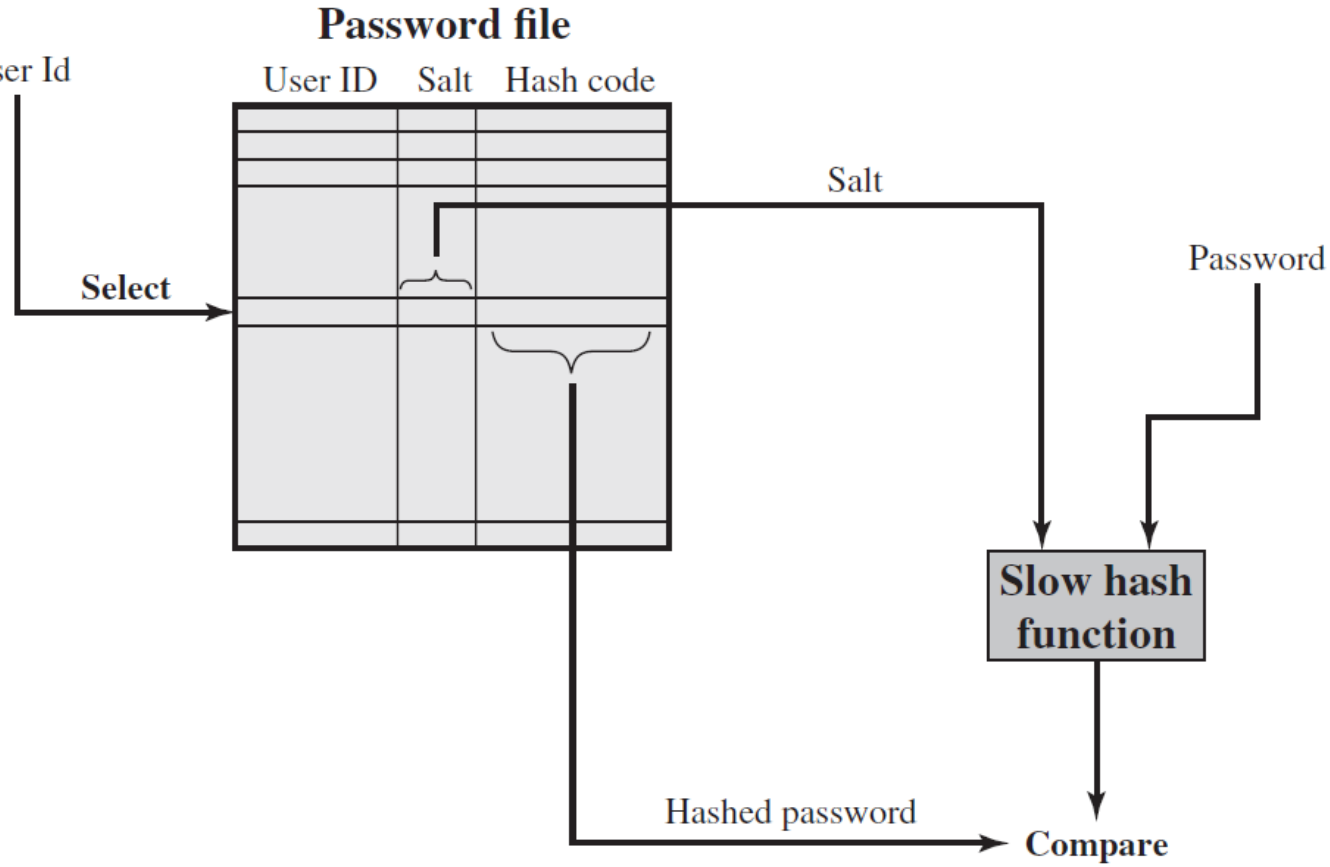
- Reasons for the persistent popularity of passwords
  - Cheap, convenient for use, and easy to implement
  - Other techniques based on client-side hardware require the implementation of the software on both client and server
    - e.g., fingerprint scanners and smart card readers
  - Physical tokens are expensive and/or inconvenient
    - e.g., smart cards
  - Biometric tokens are expensive and/or not sufficiently accurate

Use of Hashed Passwords

- A widely used password security technique
  - On all UNIX variants and other OSes



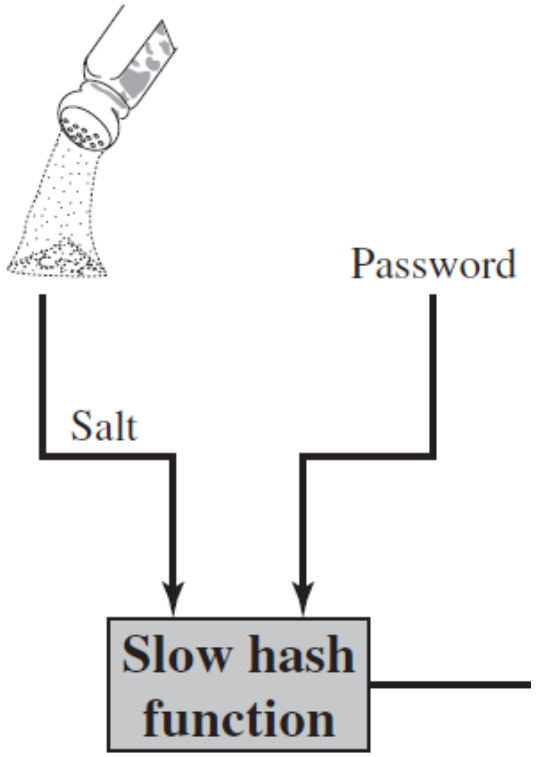
(a) Loading a new password



(b) Verifying a password

Why Salt?

- Purpose I: prevents duplicate passwords from being visible in the password file
- Purpose II: 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
- Purpose III: Nearly impossible to find out whether a person has used the same password on multiple systems



Two Threats

- Threat I: password guessing on the machine
  - Attackers gain access on a machine using a guest account or by some other means
  - Run a password guessing program, called a password cracker
- Threat II: password guessing on another machine
  - They can have a copy of the password file on another machine, and then run the cracker
  - Run through millions of possible passwords in a reasonable period

Old Implementation of UNIX Password Scheme

- Password: up to 8 characters in length (56-bit value using 7-bit ASCII)
  - Serving as the key input to DES
- Modified DES encryption
  - An one-way hash function with a data input of a 64-bit block of zeros
- Repeated for a total of 25 encryptions
- Has been regarded as inadequate (50 million password guesses in about 80 minutes)
  - But, still often required for compatibility with existing account management software or multivendor environments

Improved Implementations

- Recommended hash function is based on MD5
  - Salt: up to 48-bit
  - Password length is unlimited
  - A 128-bit hash value
  - Slowdown: an inner loop with 1000 iterations
- Bcrypt: developed for OpenBSD based on the Blowfish symmetric block cipher
  - Most secure version of Unix hash/salt scheme
  - A 128-bit salt and a 192-bit hash value
  - Configurable cost variables (number of iterations)

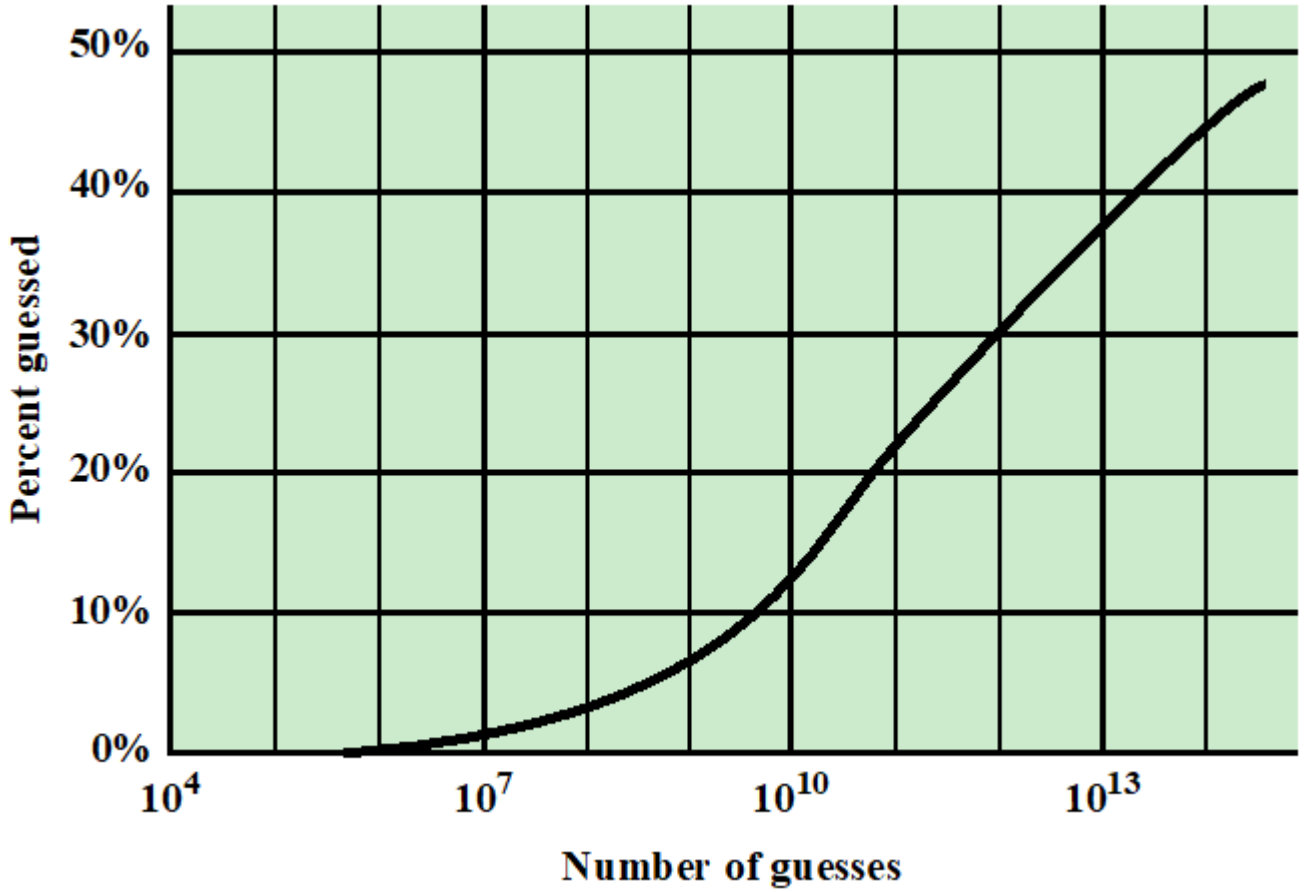
Password Cracking of User-chosen Passwords

- Traditional approaches
  - Dictionary attack
    - Prepare a large dictionary of possible password and try each
    - Each password must be hashed using each salt value and then compared to stored hash values
    - Countermeasure: slow hash functions
  - Password crackers exploit that fact that people choose easily guessable passwords
  - Rainbow table attacks
    - Pre-compute tables of hash values for all salts (a mammoth table)
    - Example: using 1.4GB rainbow table to crack 99.9% of all alphanumeric Windows password hashes in 13.8s (http://lasecwww.epfl.ch/~oechslin/publications/crypto03.pdf)
    - Countermeasure: a sufficiently large salt value and a large hash length
    - Notable open-source password cracker (developed in 1996 and still in use): Join the Ripper
      - A combination of brute-force and dictionary techniques

Modern Approaches for Password Cracking

- 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 algorithm (e.g., Markov modeling + natural language) to generate better password candidates
    - http://www.cs.cornell.edu/~shmat/shmat_ccs05pwd.pdf
  - Studying examples and structures of actual passwords in use
    - Apply data mining techniques to studying public password files (leaked by security vulnerability)
    - E.g., an SQL injection attack against online games, Rockyou.com
      → a data breach resulting in the exposure of over 32M plaintext passwords in 2009

Percentage of Passwords Recovered



- An analysis of the passwords used by over 25000 students at a research university with a complex password policy [1]
  - Using a database consisting of a collection of leaked password files [2]

[1] http://www.cs.umd.edu/~jkatz/security/downloads/passwords_revealed-weir.pdf
[2] https://www.cylab.cmu.edu/files/pdfs/tech_reports/CMUCyLab13013.pdf

Major Countermeasures

- Password file access control
- Password selection strategies
- Proactive password checking

Password File Access Control

- Two mechanisms
  - Access control: makes the password file 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
  - Weakness in physical security may provide access to backup media
  - Sniffing network traffic

Password Selection Strategies

- User education
  - Users can be told the importance of using hard-to-guess passwords
- Computer-generated passwords
  - FIPS 181 Automated Password Generator: http://pass.rasm.se/
  - But, users have trouble remembering them
- Reactive password checking
  - System periodically runs its own password cracker to find guessable passwords
- Complex password policy or proactive password checker
  - Rejecting guessable passwords

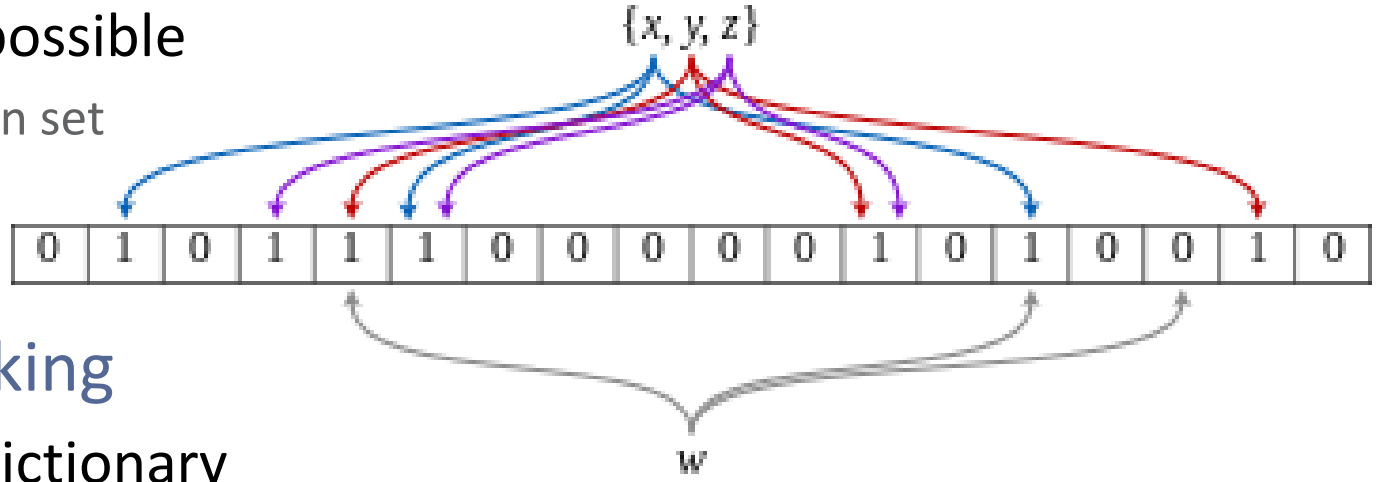
Proactive Password Checking

- Rule enforcement
  - Specific rules that passwords must adhere to
  - e.g., must be at least eight characters long, must include at least one for each of uppercase and lowercase
- Password cracker
  - Compile a large dictionary of “bad” passwords not to use
  - But, it is space-consuming and time-consuming

Can we use a hash function to address the issues?

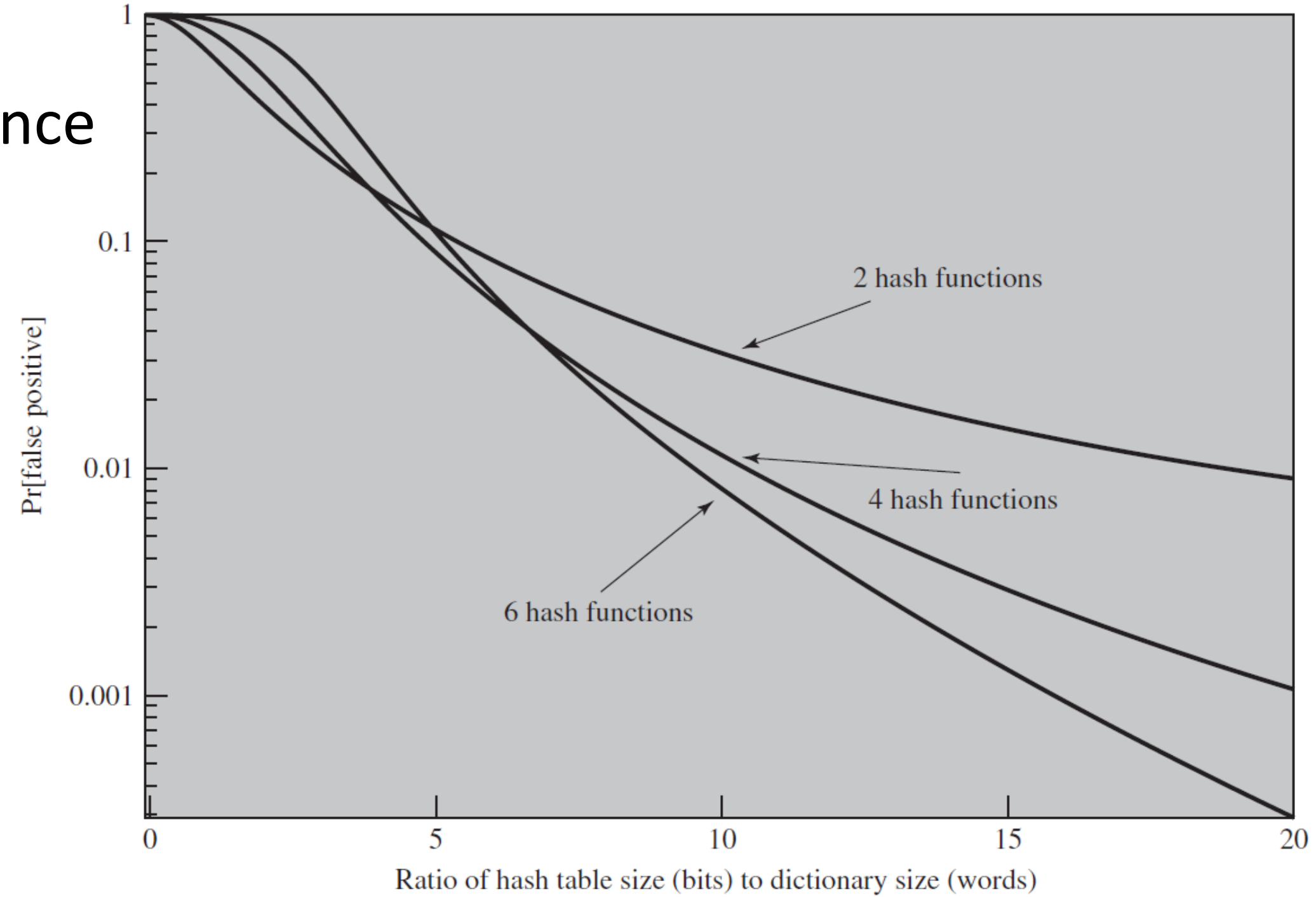
Proactive Password Checking (Cont.)

- Bloom filter: a space-efficient probabilistic data structure
  - Used to test whether an element is a member of a set
  - A bit array of *m* bits, and *k* different hash functions
  - But, false positive matches are possible
    - Result: possibly in set or definitely in set
  - Space advantage: do not store the data items
- Applied to the password checking
  - Used to build a table based on dictionary
  - Check desired password against this table



$m = 18$ and $k = 3$ (From Wikipedia)

Performance of Bloom Filter



Outline

- Electronic User Authentication Model
- Password-based Authentication
- Token-based Authentication
- Biometric Authentication
- Remote user Authentication
- Security Issues for User Authentication

Token-based Authentication

- 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	Band card
Electronic memory	Electronic memory inside	ATM, credit cards
Smart Contact Contactless	Electronic memory and processor inside Electrical contacts exposed on surface Radio antenna embedded inside	SIM card Biometric ID card

Memory Cards

- Functions
  - Can store but do not process data
  - Can include an internal electronic memory
- Most common: the bank card with a magnetic stripe on the back
- Usage
  - Alone for physical access (e.g., Hotel room)
  - Combined with a password or PIN: provides significantly greater security
- Drawbacks
  - A special reader is required
  - Token loss
  - User dissatisfaction

Smart Tokens

- Categorized along four dimensions
  - Physical characteristics
    - Include an embedded microprocessor
    - Like a bank card: smart card
    - Others: calculators, keys, small portable objects
  - User interface
    - Manual interfaces include a keypad and display for interaction

  - Electronic interface
    - Required by a smart card or other token to communicate with a compatible reader/writer
    - Contact: direct contact between a card reader and a conductive contact plate on the card
    - Contactless: both the reader and the card have an antenna

Smart Tokens (Cont.)

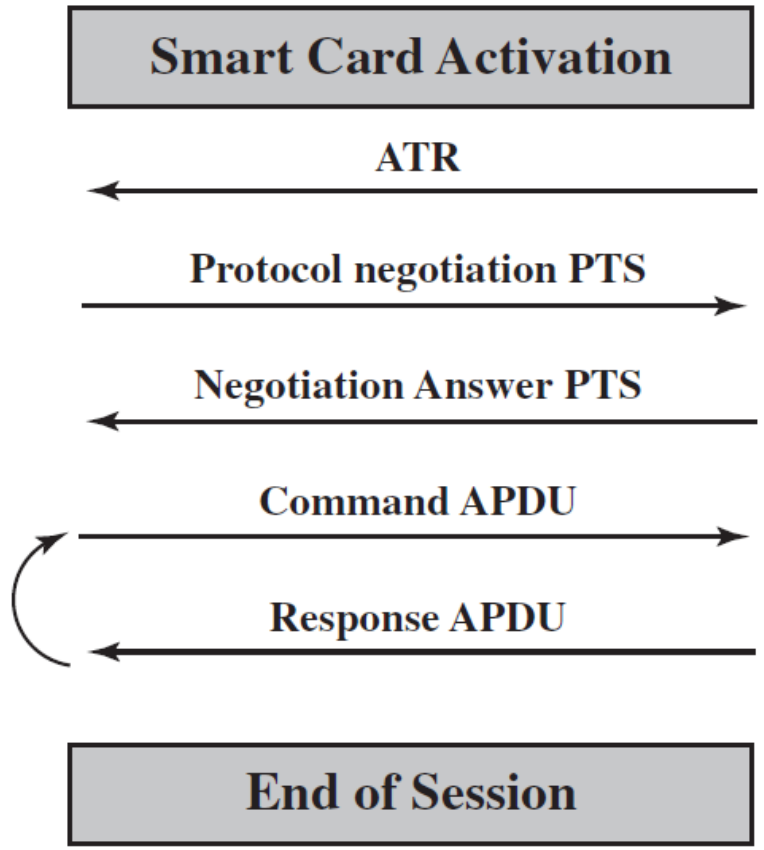
- Authentication protocol
  - Static
    - User authenticates himself or herself to the token
    - Token authenticates the user to the computer
  - Dynamic password generator
    - Token generates a unique password periodically (e.g., every minute)
    - Initialized and synchronized for the token and the computer
  - Challenge-response
    - Computer generates a challenge
    - Token generates a response to the challenge

Most Important: Smart Cards

- Contains an entire microprocessor
  - Including processor, memory, and I/O ports
  - (optional) a special co-processing circuit for cryptographic operation
- Three types of memory
  - Read-only memory (ROM)
  - Electrically erasable programmable ROM (EEPROM)
  - Random access memory (RAM)

Smart card

Card reader



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

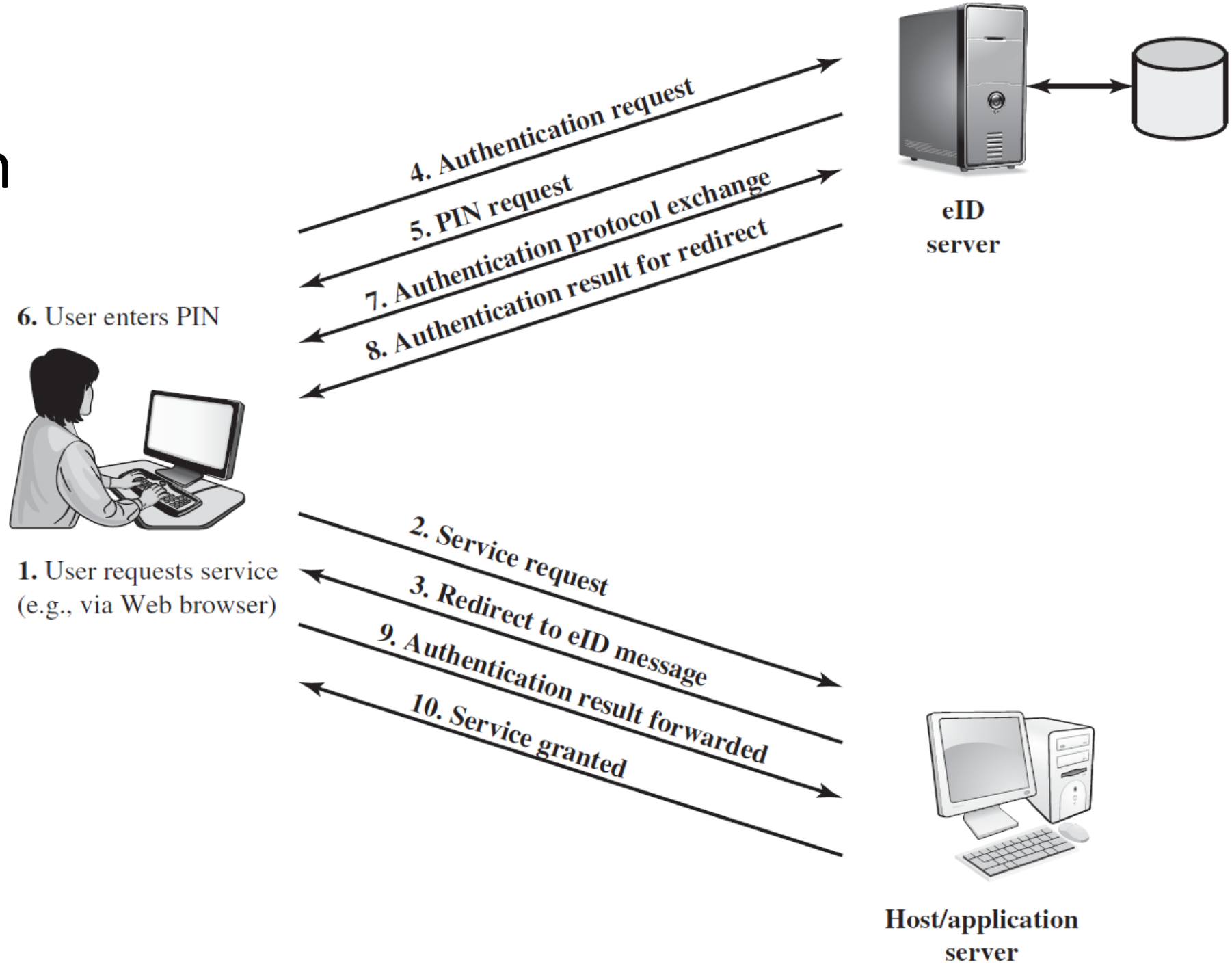
Smart Cards: Electronic Identity (eID) Cards

- Verified by the national government as valid and authentic
  - Most advanced eID: German eID card
- Three eID functions
  - ePass: government use; offline (e.g., electronic passport)
    - Stores a digital representation of the identity (e.g., face and fingerprint images)
  - eID: general-purpose use; offline and online
    - Stores an identity record (e.g., name, date of birth, address)
  - eSign: generating a digital signature
    - Stores a private key and a certificate verifying the key (e.g., X.509 certificate)

German

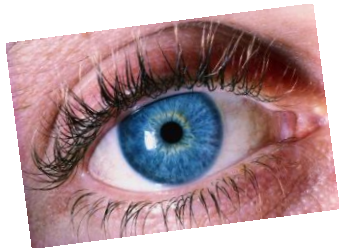
Taiwan

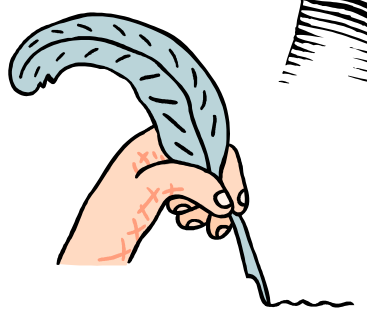
Online User Authentication with the eID Function



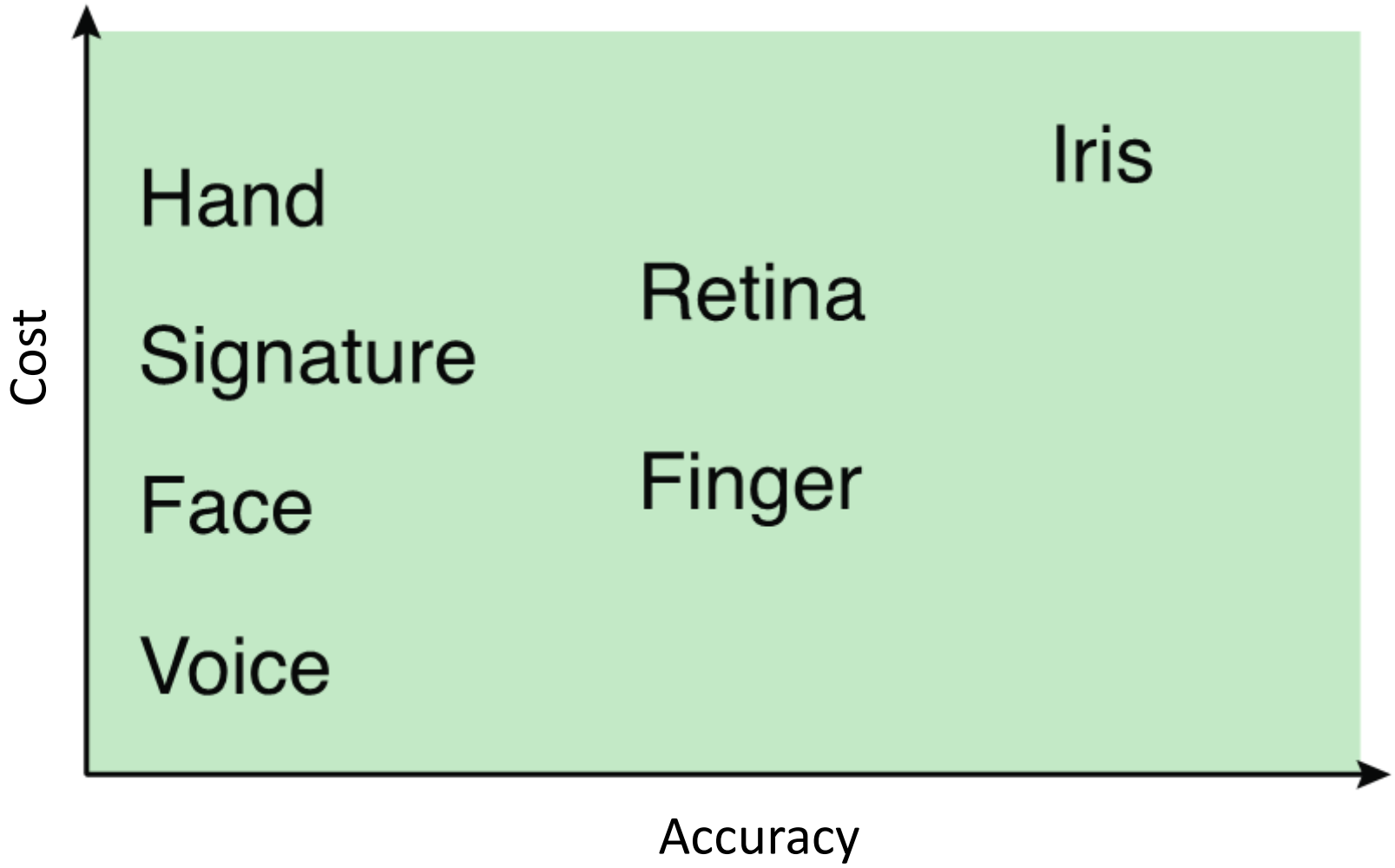
Biometric Authentication

- Authentication based on unique physical characteristics
  - Static: facial characteristics, fingerprints, hand geometry
  - Dynamic: signature, voice
- Relies on pattern recognition technologies
  - More complex and expensive than passwords and tokens
  - Not yet to mature as a standard tool



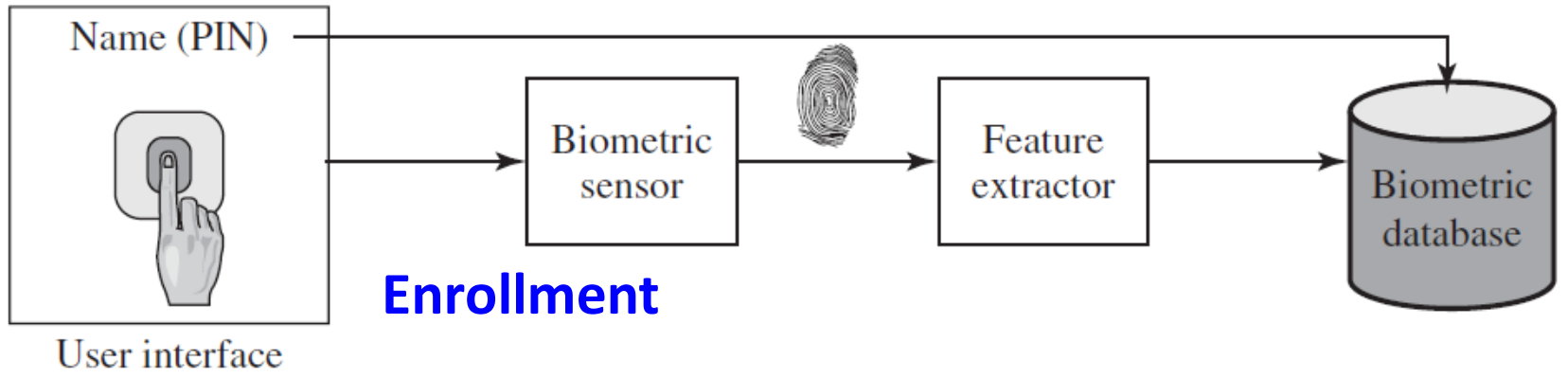


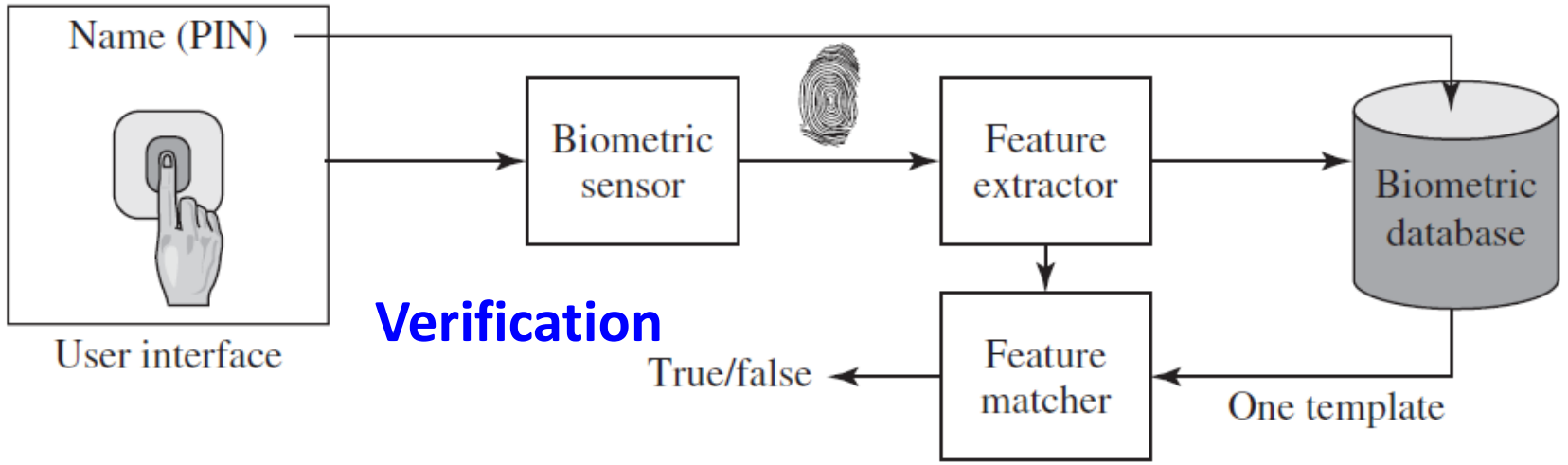
Cost vs. Accuracy of Various Biometric Characteristics

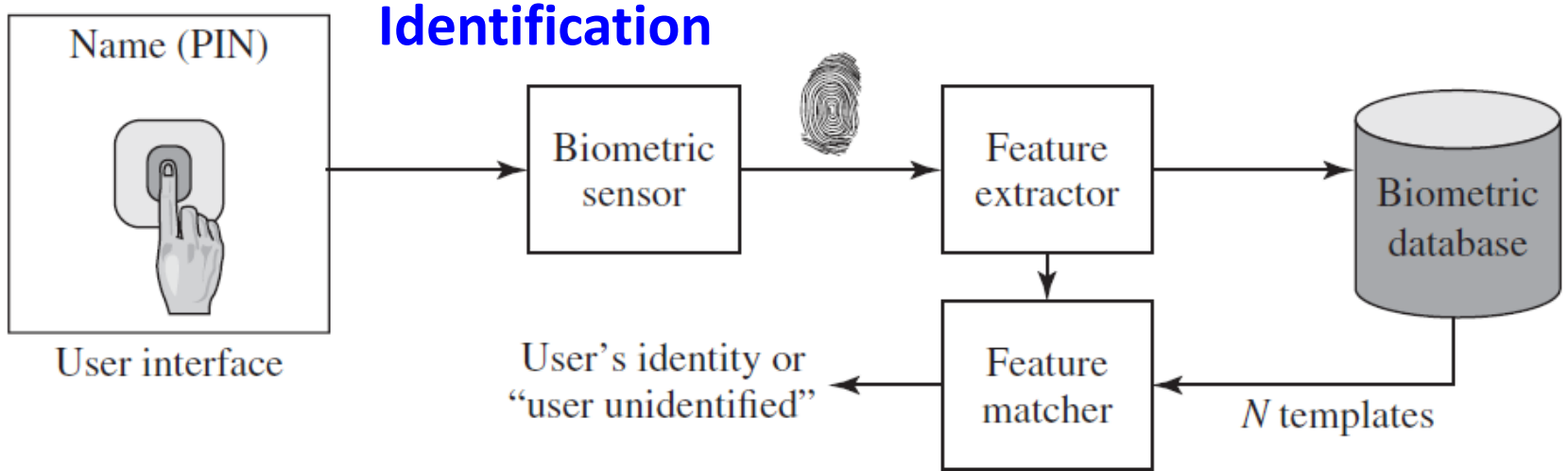


Biometric Auth System Operation

- **Verification**: analogous to a user logging on to a system by using a smart card and a PIN
- **Identification**: user presents biometric info without other info; system compares it with stored templates

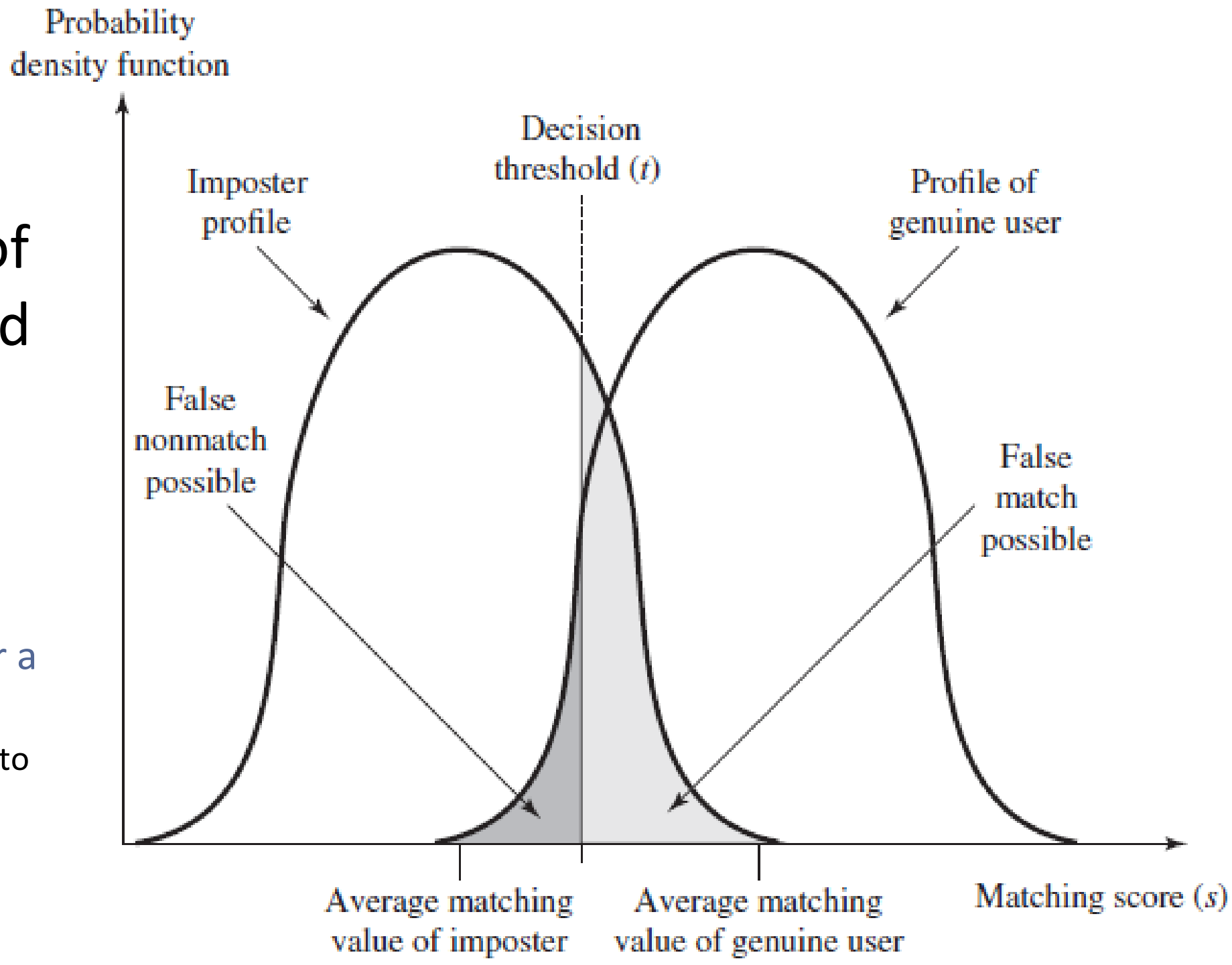






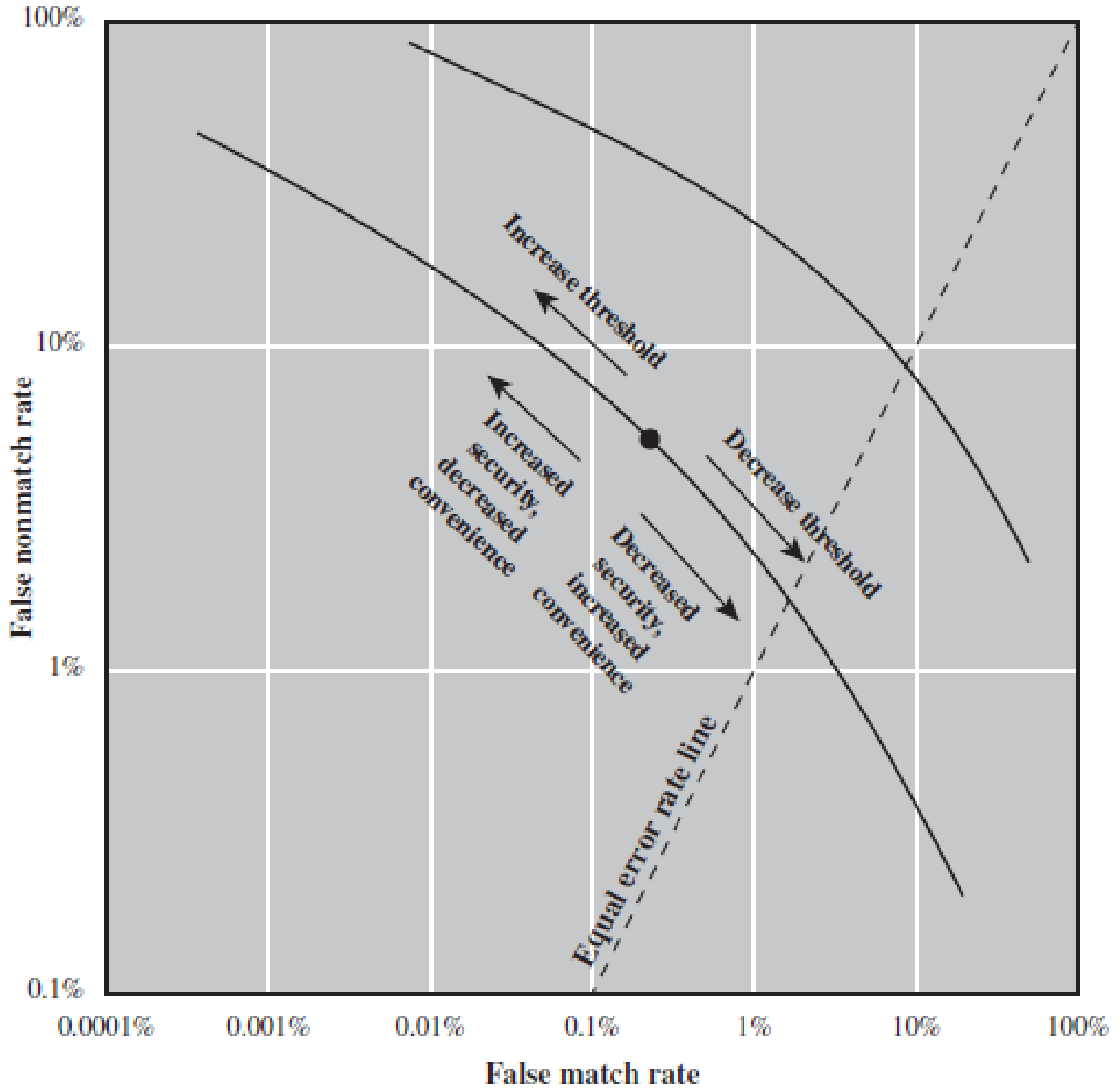
Profiles of a Biometric Characteristic of an Imposter and an Authorized User

- Dilemma: matching score would vary for a single user
  - e.g., fingerprint: due to sensor noise

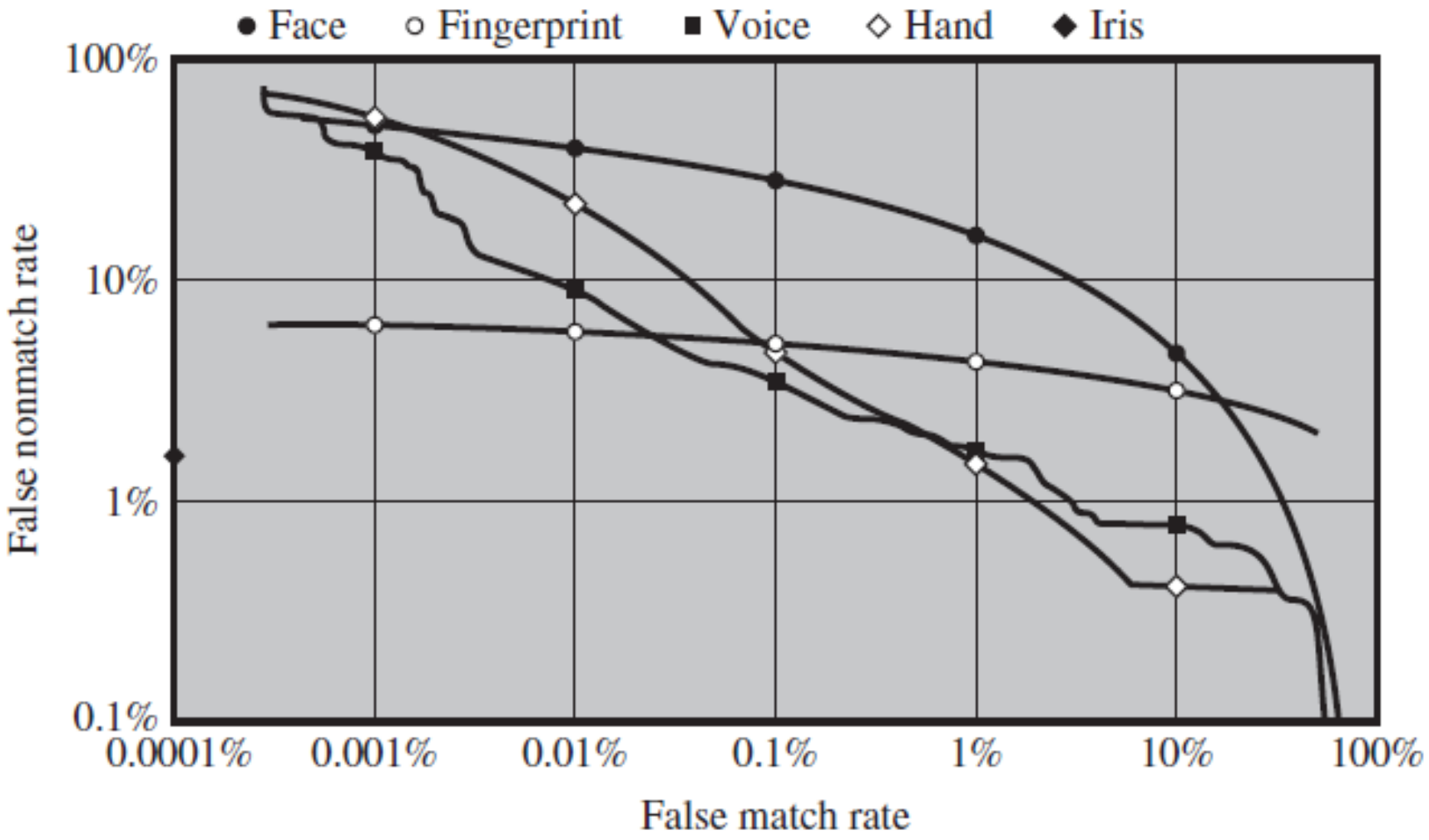


Idealized Biometric Measurement Operating Characteristic Curves (log-log scale)

- Tradeoff between security and convenience
  - Inconvenience: a valid user is denied access



Actual Biometric Measurement Operating Characteristic Curves (log-log scale)



Outline

- Electronic User Authentication Model
- Password-based Authentication
- Token-based Authentication
- Biometric Authentication
- Remote user Authentication
- Security Issues for User Authentication

Remote User Authentication

- More complex than local authentication: over a network, the Internet, or a communication link
- Why?
  - More security threats: eavesdropping, capturing a password, and replaying an authentication sequence that has been observed
- General solution: challenge-response protocols

Protocols for a Password and a Token

Password

Client	Transmission	Host
U, user	$U \rightarrow$	
	$\leftarrow \{r, \text{h}(), \text{f}()\}$	random number h(), f(), functions
P' password r', return of r	$\text{f}(r', \text{h}(P')) \rightarrow$	
	$\leftarrow$ yes/no	if $\text{f}(r', \text{h}(P')) = \text{f}(r, \text{h}(P(U)))$ then yes else no

Token

Client	Transmission	Host
U, user	$U \rightarrow$	
	$\leftarrow \{r, \text{h}(), \text{f}()\}$	r, random number h(), f(), functions
$P' \rightarrow W'$ password to passcode via token r', return of r	$\text{f}(r', \text{h}(W')) \rightarrow$	
	$\leftarrow$ yes/no	if $\text{f}(r', \text{h}(W')) = \text{f}(r, \text{h}(W(U)))$ then yes else no

Protocols for Static and Dynamic Biometric

Static

Client	Transmission	Host
U, user	$U \rightarrow$	
	$\leftarrow \{ r, \mathrm{E}() \}$	r, random number E(), function
$B' \rightarrow BT'$ biometric D' biometric device r', return of r	$\mathrm{E}(r', D', BT') \rightarrow$	$\mathrm{E}^{-1}\mathrm{E}(r', P', BT') =$ (r', P', BT')
	$\leftarrow$ yes/no	if $r' = r$ and $D' = D$ and $BT' = BT(U)$ then yes else no

Dynamic

Client	Transmission	Host
U, user	$U \rightarrow$	
	$\leftarrow \{ r, x, \mathrm{E}() \}$	r, random number x, random sequence challenge E(), function
$B', x' \rightarrow BS'(x')$ r', return of r	$\mathrm{E}(r', BS'(x')) \rightarrow$	$\mathrm{E}^{-1}\mathrm{E}(r', BS'(x')) =$ $(r', BS'(x'))$ extract B' from $BS'(x')$
	$\leftarrow$ yes/no	if $r' = r$ and $x' = x$ and $B' = B(U)$ then yes else no

Security Issues for User Authentication

- Client attacks: masquerade as a legitimate user
  - Guessing, exhaustive search, and false match
  - Countermeasures: strong passwords and limited attempts
- Host attacks: steals the user file where passwords, token passcodes, or biometric templates are stored
  - Theft of plaintext, passcode, and template
  - Countermeasures: strong access control
- Eavesdropping
  - Shoulder surfing, keystroke logging, copying biometric
  - Countermeasures: multifactor authentication and anomaly detection

Security Issues for User Authentication (Cont.)

- Relay: repeats a previously captured user response
  - Replays stolen password, passcode, and biometric template
  - Countermeasures: a random number in challenge-response protocols
- Trojan horse: installation of rogue client or capture device
  - e.g., rogue ATM or credit card scanner
  - Countermeasures: authentication of client or capture device within trusted security perimeter
- Denial of service: lockout by multiple failed authentications
  - Countermeasures: multifactor with token

Questions?