Introduction on Information security

Chapter 3 – User Authentication

Riccardo Spolaor, Ph.D rspolaor@sdu.edu.cn

Shandong University, School of Computer Science and Technology

User Authentication

- fundamental security building block
  - basis of access control & user accountability

- is the process of verifying an identity claimed by or for a system entity

- has two steps:
  - **identification** - specify identifier
  - **verification** - bind entity (person) and identifier

- User authentication != message authentication

Means of User Authentication

- four means of authenticating user's identity

- based on something the individual
  - knows - e.g. password, PIN
  - possesses - e.g. key, token, smartcard
  - is (static biometrics) - e.g. fingerprint, retina
  - does (dynamic biometrics) - e.g. voice, sign

- can use alone or combined
- all can provide user authentication
- all have issues!

Password Authentication

- widely used user authentication method
  - user provides name/login and password
  - system compares password with that saved for specified login
- authenticates ID of user logging and
  - that the user is authorized to access system
  - determines the user's privileges
  - is used in discretionary access control

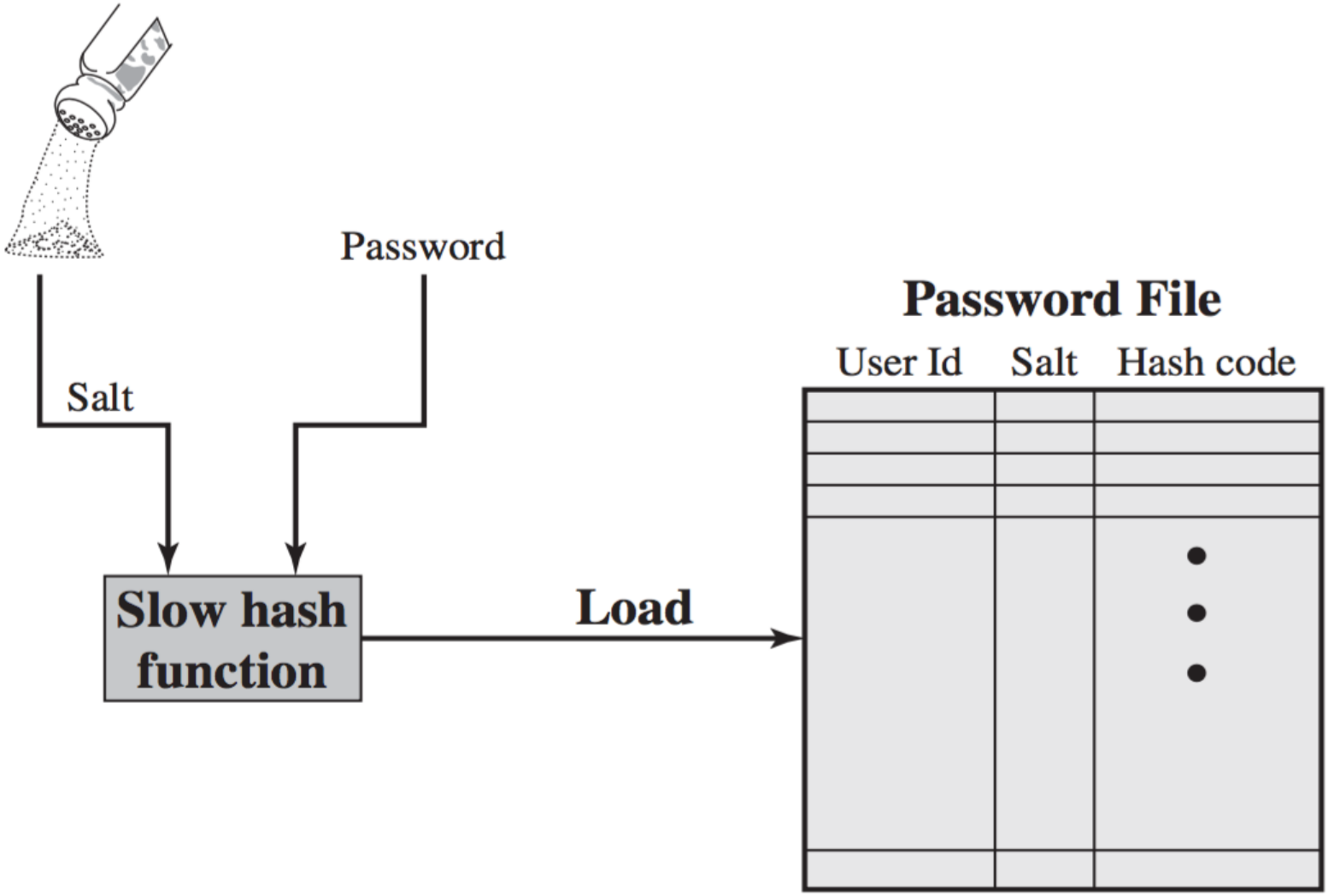
Password Vulnerabilities

- offline dictionary attack
- specific account attack
- popular password attack
- password guessing against single user
- workstation hijacking
- exploiting user mistakes
- exploiting multiple password use
- electronic monitoring

Countermeasures

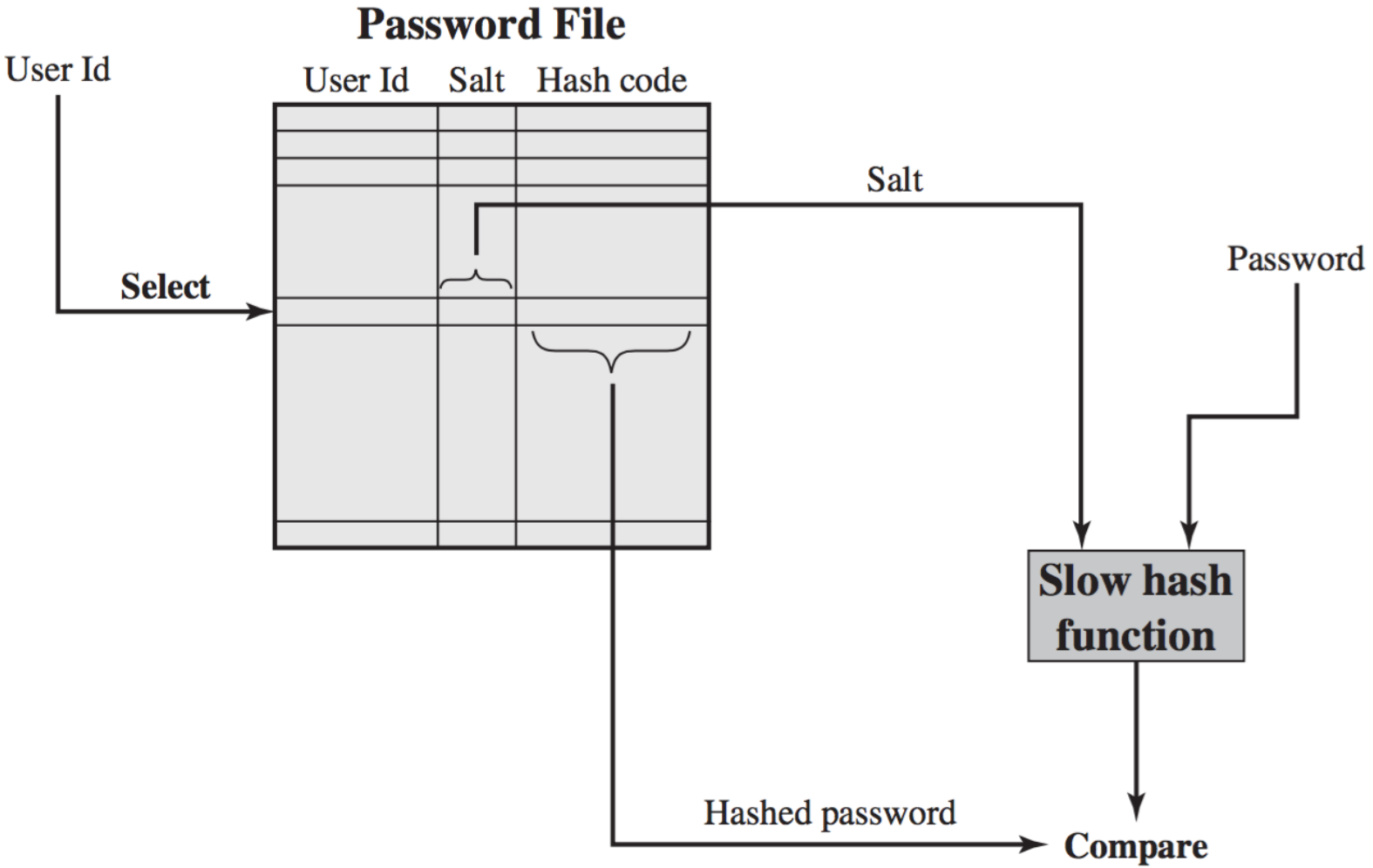
- stop unauthorized access to password file
- intrusion detection measures
- account lockout mechanisms
- policies against using common passwords but rather hard to guess passwords
- training & enforcement of policies
- automatic workstation logout
- encrypted network links

Use of Hashed Passwords



(a) Loading a new password

Use of Hashed Passwords



(b) Verifying a password

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
  - 0 value repeatedly encrypted 25 times
  - output translated to 11 character sequence
- now regarded as woefully insecure
  - e.g., supercomputer, 50 million tests, 80 min
- sometimes still used for compatibility

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 hash algorithm called Bcrypt
  - uses 128-bit salt to create 192-bit hash value

Improved Implementations

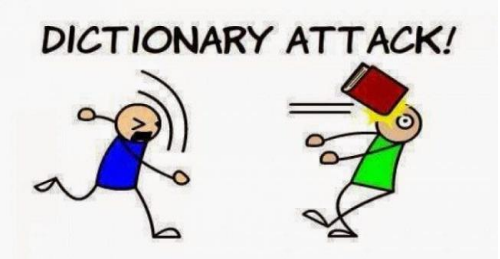
- have other, stronger, hash/salt variants

- many systems now use MD5
  - with 48-bit salt
  - password length is unlimited
  - is hashed with 1000 times inner loop
  - produces 128-bit hash
  - It Has been broken! https://www.avira.com/en/blog/md5-the-broken-algorithm

- OpenBSD uses Blowfish block cipher based hash algorithm called Bcrypt
  - uses 128-bit salt to create 192-bit hash value

Password Cracking

- dictionary attacks
  - try each word then obvious variants in **large dictionary** against hash in password file



- rainbow table attacks
  - precompute tables of hash values for all salts
  - a mammoth table of hash values
  - e.g., 1.4GB table cracks 99.9% of alphanumeric Windows passwords in 13.8 secs
  - not feasible if larger salt values used

Password Choices

- 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!

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 file

- still have vulnerabilities
  - exploit O/S bug
  - accident with permissions making it readable
  - users with same password on other systems
  - access from unprotected backup media
  - sniff passwords in unprotected network traffic

Using Better Passwords

- clearly have problems with passwords
- goal to eliminate guessable passwords
- whilst still easy for user to remember
- techniques:
  - user education
  - computer-generated passwords
  - reactive password checking
  - proactive password checking

Proactive Password Checking

- rule enforcement plus user advice, e.g.
  - 8+ chars, upper/lower/numeric/punctuation
  - may not suffice
- password cracker
  - time and space issues
- Markov Model
  - generates guessable passwords
  - hence reject any password it might generate
- Bloom Filter
  - use to build table based on dictionary using hashes
  - check desired password against this table

TIME IT TAKES A HACKER TO BRUTE FORCE YOUR PASSWORD IN 2022

Number of Characters	Numbers Only	Lowercase Letters	Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters, Symbols
4	Instantly	Instantly	Instantly	Instantly	Instantly
5	Instantly	Instantly	Instantly	Instantly	Instantly
6	Instantly	Instantly	Instantly	Instantly	Instantly
7	Instantly	Instantly	2 secs	7 secs	31 secs
8	Instantly	Instantly	2 mins	7 mins	39 mins
9	Instantly	10 secs	1 hour	7 hours	2 days
10	Instantly	4 mins	3 days	3 weeks	5 months
11	Instantly	2 hours	5 months	3 years	34 years
12	2 secs	2 days	24 years	200 years	3k years
13	19 secs	2 months	1k years	12k years	202k years
14	3 mins	4 years	64k years	750k years	16m years
15	32 mins	100 years	3m years	46m years	1bn years
16	5 hours	3k years	173m years	3bn years	92bn years
17	2 days	69k years	9bn years	179bn years	7tn years
18	3 weeks	2m years	467bn years	11tn years	438tn years

HIVE SYSTEMS

› Learn about our methodology at hivesystems.io/password

What if the pasword has been into a leaked database?

What if the pasword has been into a leaked database?

Number of Characters	Numbers Only	Lowercase Letters	Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters, Symbols
4	Instantly	Instantly	Instantly	Instantly	Instantly
5	Instantly	Instantly	Instantly	Instantly	Instantly
6	Instantly	Instantly	Instantly	Instantly	Instantly
7	Instantly	Instantly	Instantly	Instantly	Instantly
8	Instantly	Instantly	Instantly	Instantly	Instantly
9	Instantly	Instantly	Instantly	Instantly	Instantly
10	Instantly	Instantly	Instantly	Instantly	Instantly
11	Instantly	Instantly	Instantly	Instantly	Instantly
12	Instantly	Instantly	Instantly	Instantly	Instantly
13	Instantly	Instantly	Instantly	Instantly	Instantly
14	Instantly	Instantly	Instantly	Instantly	Instantly
15	Instantly	Instantly	Instantly	Instantly	Instantly
16	Instantly	Instantly	Instantly	Instantly	Instantly
17	Instantly	Instantly	Instantly	Instantly	Instantly
18	Instantly	Instantly	Instantly	Instantly	Instantly

Token Authentication

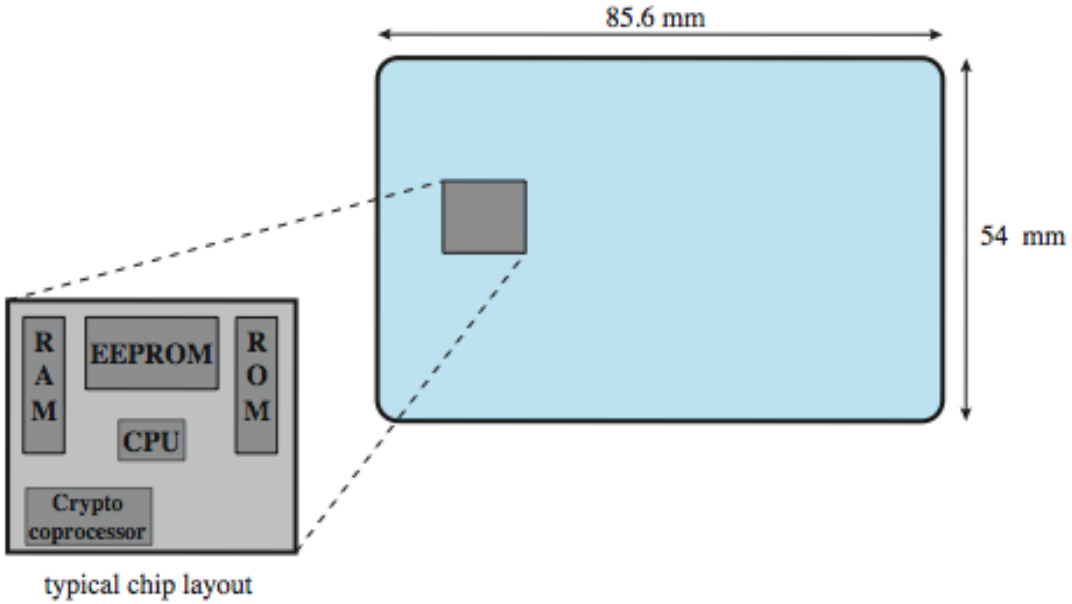
- Authentication object that the user possesses:
  - embossed card
  - magnetic stripe card
  - memory card
  - smartcard

Memory Card

- store but do not process data
- magnetic stripe card, e.g. bank card
- electronic memory card
- used alone for physical access
- with password/PIN for computer use
- drawbacks of memory cards include:
  - need special reader
  - loss of token issues
  - user dissatisfaction

Smartcard

- credit-card like
- has own processor, memory, I/O ports
  - wired or wireless access by reader
  - may have crypto co-processor
  - ROM, EEPROM, RAM memory
- executes protocol to authenticate with reader/computer
- also have USB dongles



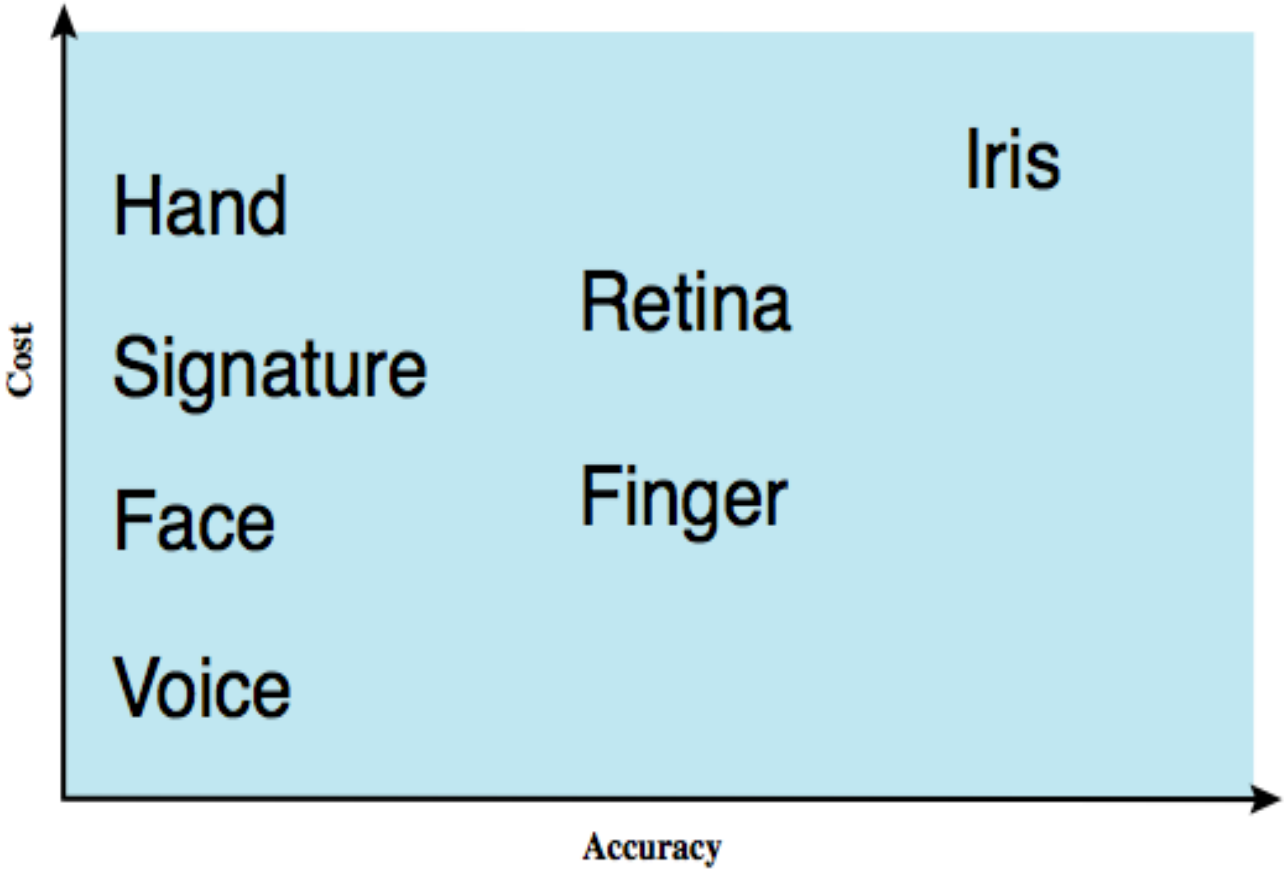
typical chip layout

Biometric Authentication

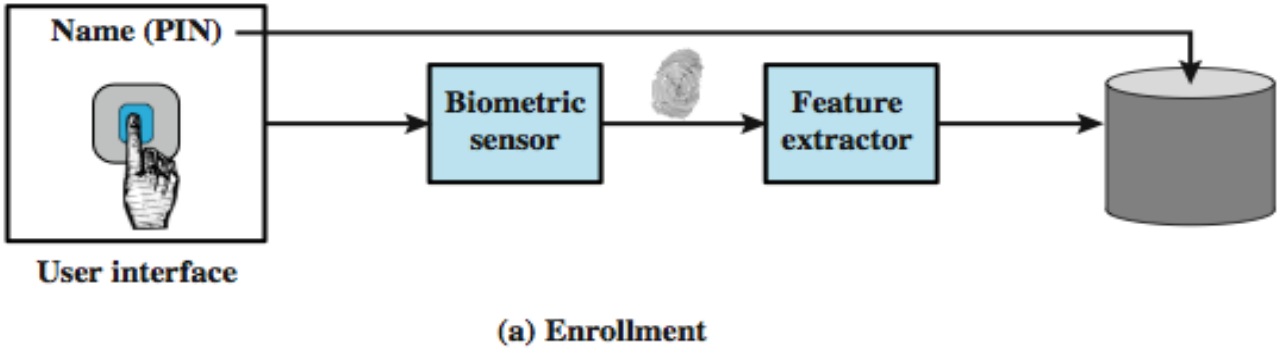
Authenticate user based on one of their physical characteristics

Biometric Authentication

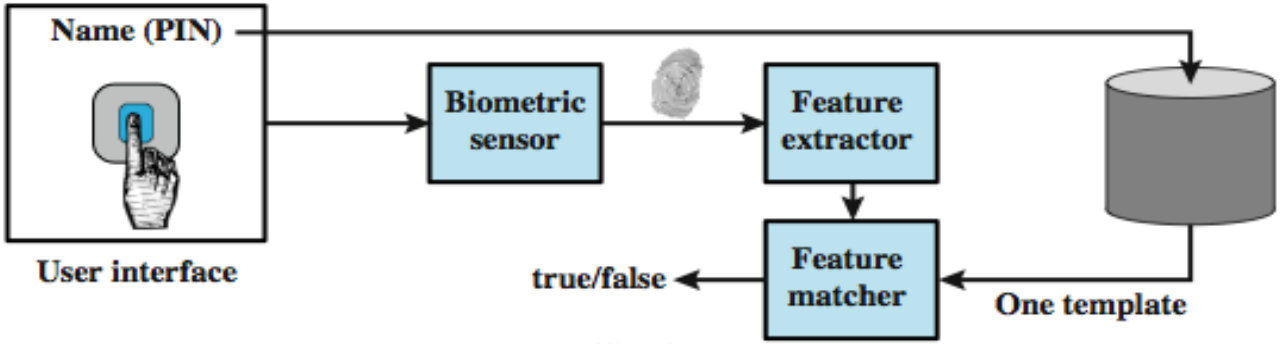
Authenticate user based on one of their physical characteristics



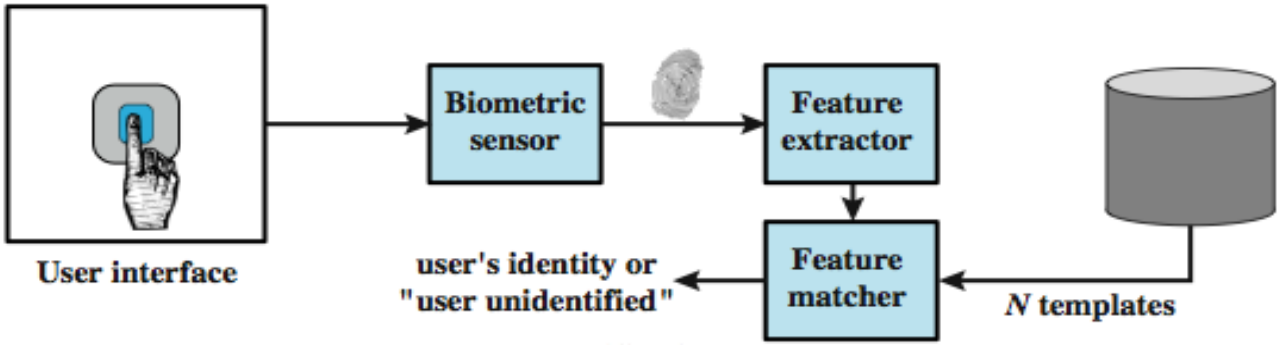
Operations of a Biometric System



(a) Enrollment



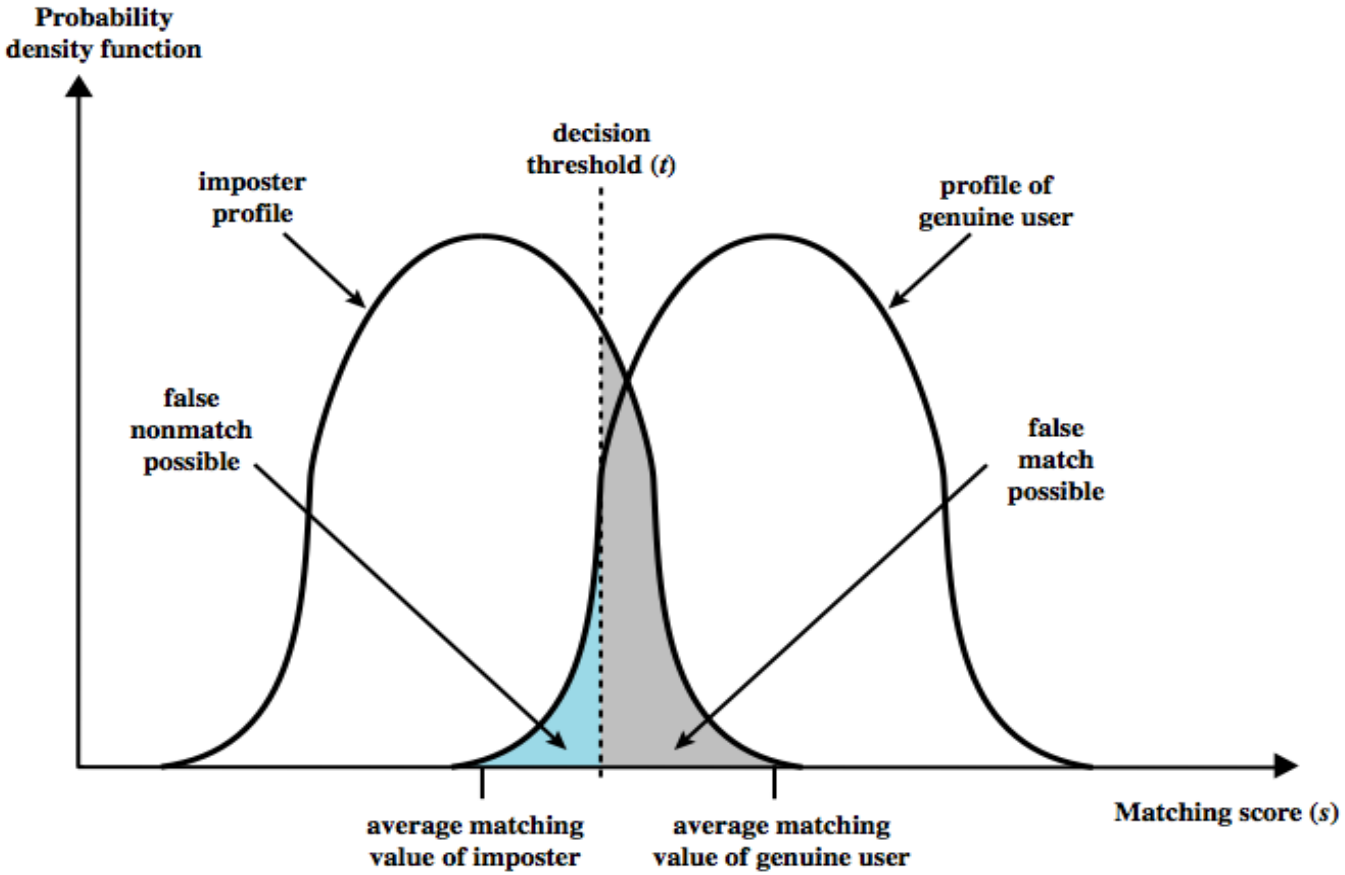
(b) Verification



(c) Identification

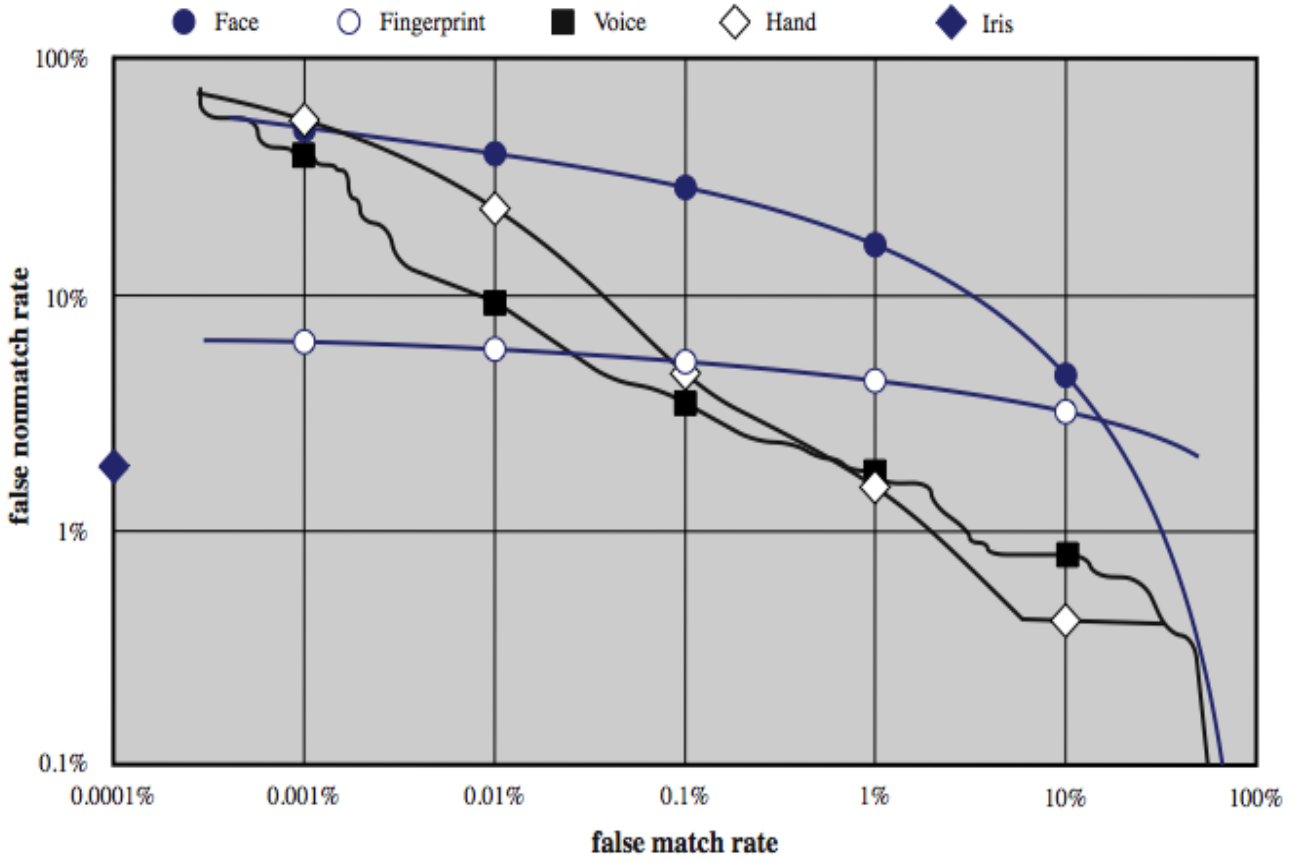
Biometric Accuracy

- never get identical templates
- problems of false match / false non-match



Biometric Accuracy

- can plot characteristic curve
- pick threshold balancing error rates



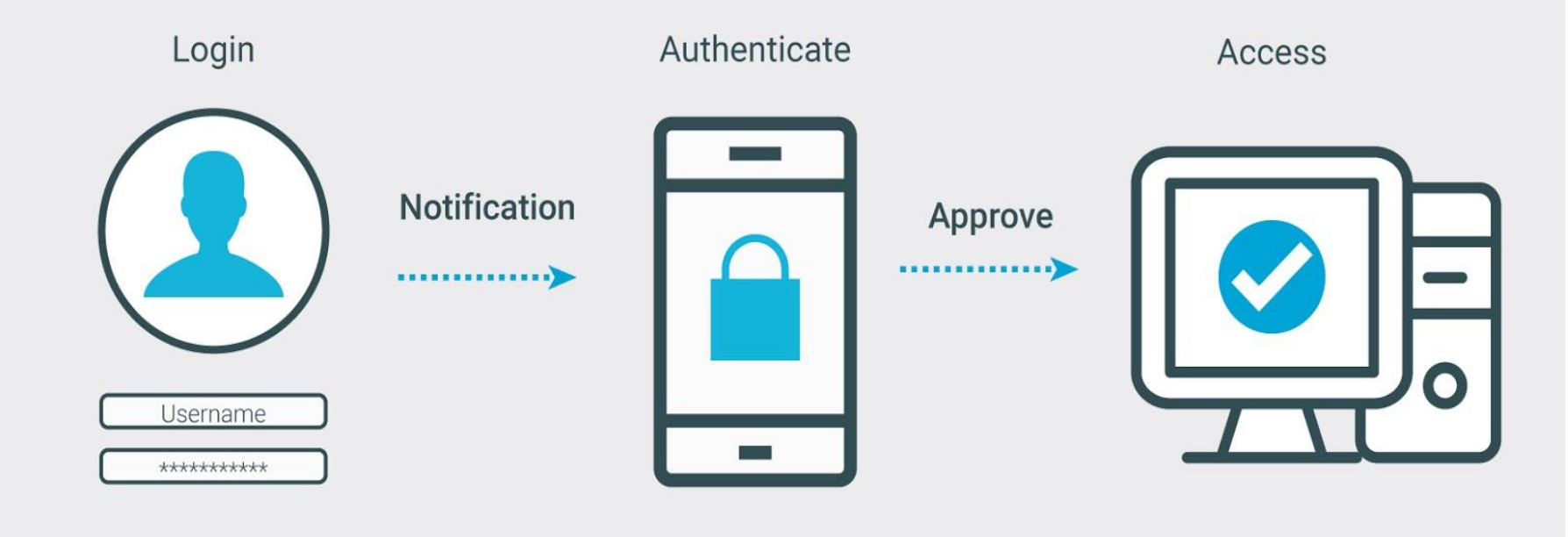
Remote User Authentication

- authentication over network more complex
  - problems of eavesdropping, replay
- generally use challenge-response
  - user sends identity
  - host responds with random number
  - user computes f(r,h(P)) and sends back
  - host compares value from user with own computed value, if match user authenticated
- protects against a number of attacks

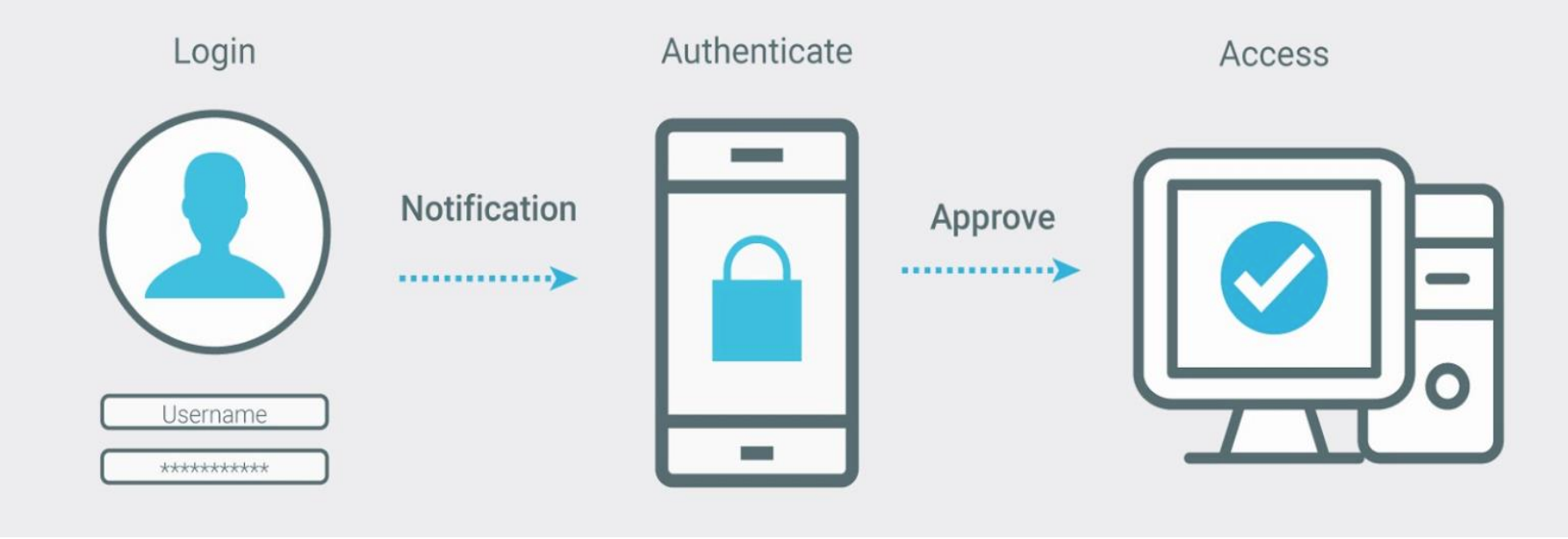
Authentication Security Issues

- client attacks
- host attacks
- eavesdropping
- replay
- trojan horse
- denial-of-service

2FA Authentication



2FA Authentication



Summary

- introduced user authentication
  - using passwords
  - using tokens
  - using biometrics
- remote user authentication issues