Authentication: Are you who you say you are?

Access Control

- Access control for physical security

- Access control for computer security involves restricted access to computer system resources
  - File systems
  - Cloud computers
  - High performance computers
  - …
- Another foundation of computer security, other than crypto

Two parts of access control

- **Authentication:** Are you who you say you are?
  - Determine whether access is allowed
  - Authenticate human to machine
  - Or authenticate machine to machine
    - Authentication over network is different than local machine
- **Authorization:** Are you allowed to do that?
  - Once you have access, what can you do?
  - Enforces limits on actions
- Note: “access control” often used as synonym for authorization

Are You Who You Say You Are?

- How to authenticate human to a machine?
- Can be based on…
  - Something you **know**
    - For example, a password
  - Something you **have**
    - For example, a ATM card or smartcard
  - Something you **are**
    - For example, your fingerprint

Something You Know

- Passwords
  - Computer can verify that you know, and something **nobody** else can guess—even with access to unlimited computing resources

- Lots of things act as passwords!
  - PIN
  - Social security number
  - Mother's maiden name
  - Date of birth
  - Name of your pet, etc.

Why Passwords?

- Why is “something you know” more popular than “something you have” and “something you are”?
- **Cost**: passwords are free
  - ID card/biometric device cost money
- **Convenience**: easier for admin to reset pwd than to issue a new thumb

An ideal password

- Something that you know
- Something that your computer can verify that you know
- Something that something nobody else can guess
- But these standards are difficult to meet in reality

Keys vs Passwords

- **Crypto keys**
- Suppose key is 64 bits
- Then 2^{64} keys
- Choose key at random…
- …then attacker must try about 2^{63} keys

- **Passwords**
- Suppose passwords are 8 characters, and 256 different characters
- Then $256^8 = 2^{64}$ pwds
- Users do not select passwords at random
- Attacker has far less than 2^{63} pwds to try (**dictionary attack**)

Good and Bad Passwords

- Bad passwords
  - frank
  - Fido
  - password
  - 4444
  - Pikachu
  - 10251960
  - AustinStamp
- Good Passwords?
  - jflej(43j-EmmL+y
  - 09864376537263
  - B1ngh@mt0n
  - FSa7Yago

Passphrase: Four score and seven years ago

Most common passwords in 2023

Rank	Password	Time taken to crack	Number of times used
1	123456	< 1 Second	4,524,867
2	admin	< 1 Second	4,008,850
3	12345678	< 1 Second	1,371,152
4	123456789	< 1 Second	1,213,047
5	1234	< 1 Second	969,811
6	12345	< 1 Second	728,414
7	password	< 1 Second	710,321
8	123	< 1 Second	528,086
9	Aa123456	< 1 Second	319,725
10	1234567890	< 1 Second	302,709

Attacks on Passwords

- Attacker could…
  - Target one particular account
  - Target any account on system
  - Target any account on any system
  - Attempt denial of service (DoS) attack
- Common attack path
  - Outsider → normal user → administrator
    - attempt to upgrade level of privilege
  - May only require **one** weak password!

Password Retry

- Suppose system locks after 3 bad passwords. How long should it lock?
  - 5 seconds
    - Insufficient to deter an automatic attack
  - 5 minutes
    - DOS
  - Admin manually resets

Password File?

- Bad idea to store passwords in a file
- But we need to verify passwords, **how**?
- Symmetric key crypto? Public key crypto? Crypto hash?
- Cryptographic solution: **hash** the pwd
  - Store **y = h(password)**
  - Can verify entered password by hashing
  - If Trudy obtains "password file," she does not obtain passwords
- But Trudy can try a *forward search*
  - Guess x and check whether $y = h(x)$

Dictionary Attack – Forward Search

- Trudy pre-computes $h(x)$ for all x in a **dictionary** of common passwords
- Suppose Trudy gets access to password file containing hashed passwords
  - She only needs to compare hashes to her pre-computed dictionary
  - After one-time work, actual attack is trivial
- Can we prevent this attack? Or at least make attacker's job more difficult?

Salt

- Hash password with **salt**
- Choose random salt s for each user and compute

  $y = h(password, s)$

  and store (s,y) in the password file
- Append salt to each password before hash
- Note: The salt s is not secret
- Easy to verify salted password
- But Trudy **must re-compute** dictionary hashes for each user
  - Lots more work for Trudy!

Case study: Linux password

- **/etc/passwd**: stores the password file
- **/etc/shadow**: readable only from the root account
  - root:1Etg2ExUZ$F9NTP7omafhKIlqaBMqng1:15651:0:99999:7:::
  - $1 = MD5 hashing algorithm
    $2 = Blowfish algorithm
    $2a= eksblowfish algorithm
    $5 = SHA-256 algorithm
    $6 = SHA-512 algorithm
  - $Etg2ExUZ → Salt = 'Etg2ExUZ'
  - $F9NTP7omafhKIlqaBMqng1 → hashed value of (salt + user password)
- Run: **openssl** passwd -1 -salt Etg2ExUZ redhat
  - 1Etg2ExUZ$F9NTP7omafhKIlqaBMqng1

Other Password Issues

- Too many passwords to remember
  - Results in password reuse; Why is this a problem?
  - Password manager software
    - Master key to reveal other passwords
- Failure to change default passwords
- Social engineering by, say, claiming to be admin
  - 34% of users would give away, and 70% if offered a candy bar
- Error logs may contain “almost” passwords
- Bugs, keystroke logging, spyware, etc.
- Who suffers from bad password?
  - Login password (company) vs ATM PIN (only yourself)

Password Cracking Tools

- Popular password cracking tools
  - Password Crackers
  - Password Portal
  - L0phtCrack and LC4 (Windows)
  - John the Ripper (Unix)
  - Come with preconfigured dictionaries
- Admins should use these tools to test for weak passwords since attackers will
- Good articles on password cracking
  - Passwords - Conerstone of Computer Security
  - Passwords revealed by sweet deal

The bottom line…

- **Password cracking is too easy**
  - One weak password may break security
  - Users choose bad passwords
  - Social engineering attacks, etc.
- Trudy has (almost) all of the advantages
- Passwords are a **BIG** security problem
  - And will continue to be a big problem

Biometrics: authentication based on something you are

Quiz

- Which of the following best describes the purpose of adding a **salt** to a password before hashing?
  - A) To encrypt the password, making it unreadable to unauthorized users.
  - B) To ensure that the hash output for the same password is different even if the password is used by multiple users, thereby guarding against forward search attacks.
  - C) To speed up the password authentication process by adding additional data to the password.
  - D) To compress the password into a smaller format for easier storage.

Quiz

- What is true about hashing a password **p** with salt **s**?
  - A)Salt s is used as the key to encrypt password p
  - B)Salt s is public and stored along with hashed password p
  - C) The salt s is kept secret in the same way as the password p, ensuring both are secure from unauthorized access.
  - D)Salt s makes password p taste salty

Quiz

- What is the primary reason passwords are hashed before being stored in a database?
  - A) To compress the passwords, reducing the amount of storage space required.
  - B) To encrypt the password so that it can be easily decrypted by the system for authentication.
  - C) To transform the passwords into a fixed-size string of characters, regardless of the password's length.
  - D) To ensure that even if the database is compromised, the actual passwords are not easily retrievable by attackers.

Recap: Are You Who You Say You Are?

- How to authenticate human to a machine?
- Can be based on…
  - Something you **know**
    - For example, a password
  - Something you **have**
    - For example, a ATM card or smartcard
  - Something you **are**
    - For example, your fingerprint

Something You Are

- Biometric
  - **“You are your key”**
- Examples
  - Fingerprint
  - Handwritten signature
  - Facial recognition
  - Speech recognition
  - Gait (walking) recognition
  - Many more!

Why Biometrics?

- More secure replacement for passwords
- Cheap and reliable biometrics needed
  - Today, an active area of research
- Biometrics **are** used in security today
  - Thumbprint mouse
  - Palm print for secure entry
  - Fingerprint to unlock car door, etc.
  - Facial recognition to unlock phones
- Biometrics are getting increasingly popular

Ideal Biometric

- **Universal** — applies to (almost) everyone
  - Most ppl have readable fingerprints
  - In reality, no biometric applies to everyone
- **Distinguishing** — distinguish with certainty
  - In reality, cannot hope for 100% certainty
  - Some with lower error rates
- **Permanent** — physical characteristic being measured never changes
  - In reality, OK if it to remains valid for long time
- **Collectable** — easy to collect required data
  - Depends on whether subjects are cooperative
- Also, safe, user-friendly, etc., etc.

Biometric Modes

- **Identification** — Who goes there?
  - Compare **one-to-many**
  - Example: The FBI fingerprint database
    - Suspicious fingerprint compared to millions of fingerprint
- **Authentication** — Are you who you say you are?
  - Compare **one-to-one**
  - Example: Fingerprint unlock phones
- Identification problem is more difficult
  - More “random” matches since more comparisons
- We are interested in authentication

Enrollment vs Recognition

- Enrollment phase
  - Subject's biometric info put into database
  - Must carefully measure the required info
  - OK if slow and repeated measurement needed
  - Must be very precise
  - May be weak point of many biometric
    - difficult to obtain results that are comparable to those obtained under lab conditions

- Recognition phase
  - Biometric detection, when used in practice
  - Must be quick and simple
  - But must be reasonably accurate

Cooperative Subjects?

- Authentication — cooperative subjects
- Identification — uncooperative subjects
- For example, facial recognition
  - Used in Las Vegas casinos to detect known cheaters (terrorists in airports, etc.)
  - Often do not have ideal enrollment conditions
  - Subject will try to confuse recognition phase
- Cooperative subject makes it much easier
  - We are focused on authentication
  - So, subjects are generally cooperative

Biometric Errors

- **Fraud rate** versus **insult rate**
  - Fraud — Trudy mis-authenticated as Alice
  - Insult — Alice not authenticated as Alice
- For any biometric, can decrease fraud or insult, but other one will increase
- For example
  - 99% voiceprint match $\Rightarrow$ low fraud, high insult
  - 30% voiceprint match $\Rightarrow$ high fraud, low insult
- **Equal error rate: rate where fraud == insult**
  - A way to compare different biometrics

Fingerprint Comparison

- The widespread use of fingerprinting only became possible in 1892
  - Francis Galton developed a classification system based on "minutia" that enabled efficient searching, and he verified that fingerprints do not change over time
- Examples of different types of minutia: **loops**, **whorls**, and **arches**

Loop (double)

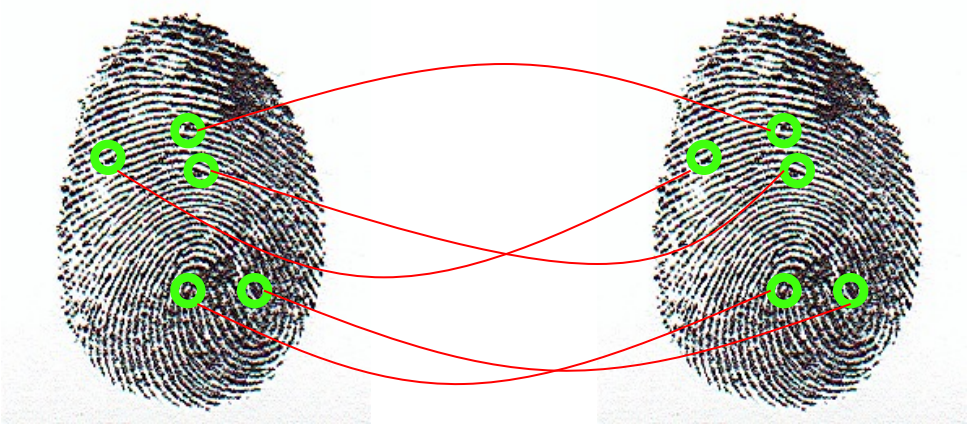
Whorl

Arch

Fingerprint: Enrollment

- Capture image of fingerprint
- Enhance image
- Identify points

Fingerprint: Recognition



- Extracted points are compared with information stored in a database
- British system: 16 points, US: not fixed
- The system then determines whether a statistical match occurs

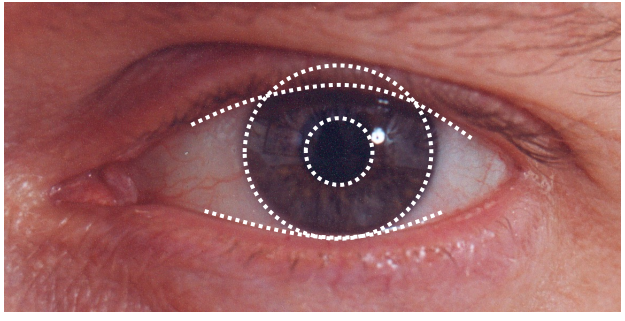
Hand Geometry

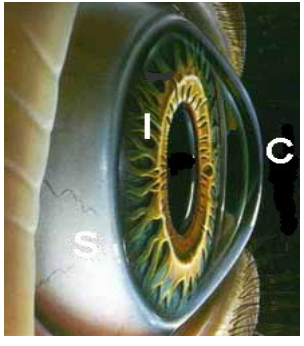
- A popular biometric
- Measures shape of hand
  - Width of hand, fingers
  - Length of fingers, etc.
- Human hands not unique
- Hand geometry sufficient for many situations
- OK for authentication
- Not useful for ID problem

Hand Geometry

- Advantages
  - Quick — 1 minute for enrollment, 5 seconds for recognition
  - Hands are symmetric
- Disadvantages
  - Cannot use on very young or very old
  - Relatively high equal error rate

Iris Patterns

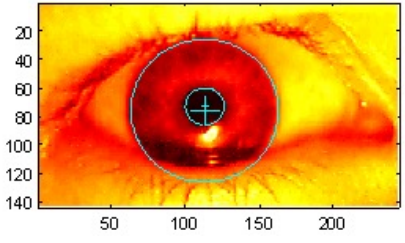


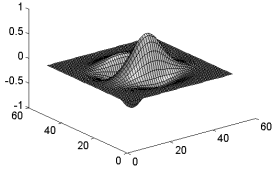


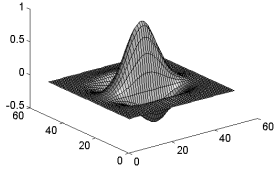
- In theory, one of the best for authentication
- Iris pattern development is “chaotic”
  - minor variations lead to large differences
- Little or no genetic influence
- Different even for identical twins
- Pattern is stable through lifetime

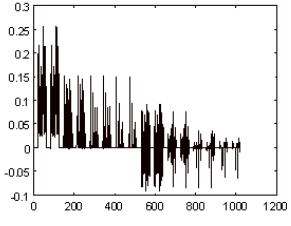
Iris Scan

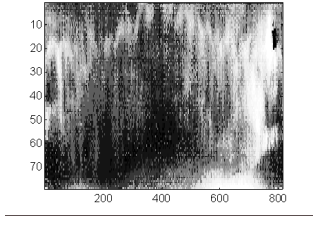
- Scanner locates iris
- Take b/w photo
- Use polar coordinates…
- 2-D wavelet transform
- Get 256-byte iris code









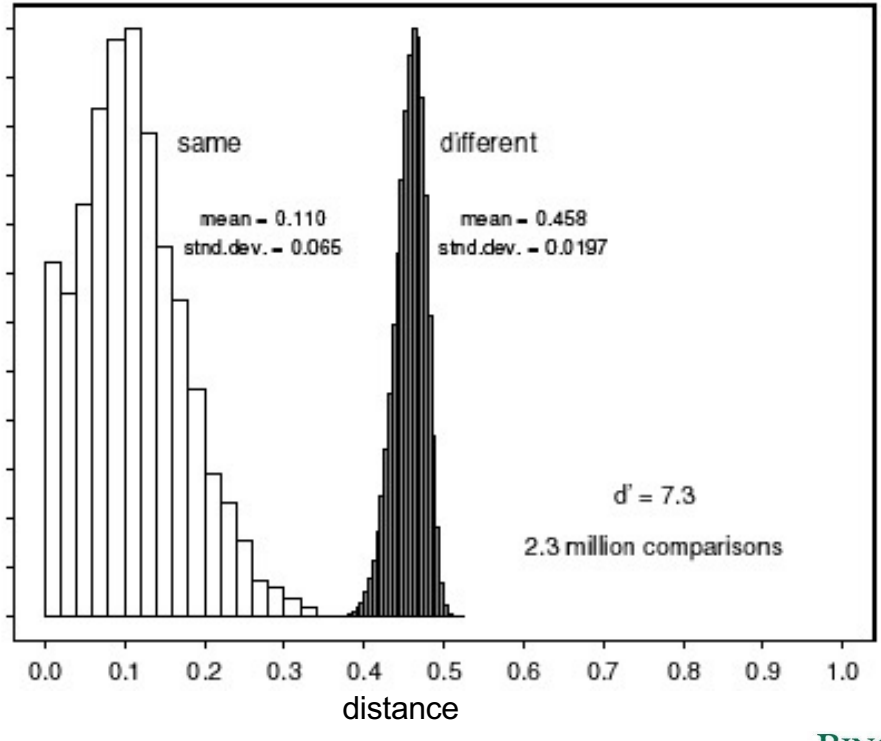


Measuring Iris Similarity

- Based on Hamming distance
- Define $d(x,y)$ to be
  - # of non match bits / # of bits compared
  - $d(0010,0101) = 3/4$ and $d(101111,101001) = 1/3$
- Compute $d(x,y)$ on 2048-bit iris code
  - Perfect match is $d(x,y) = 0$
  - For same iris, expected distance is 0.08
  - At random, expect distance of 0.50
  - Accept iris scan as match if distance < 0.32

Iris Scan Error Rate: 2.3 million comparisons

distance	Fraud rate
0.29	1 in $1.3*10^{10}$
0.30	1 in $1.5*10^{9}$
0.31	1 in $1.8*10^{8}$
0.32	1 in $2.6*10^{7}$
0.33	1 in $4.0*10^{6}$
0.34	1 in $6.9*10^{5}$
0.35	1 in $1.3*10^{5}$



Attack on Iris Scan

- Good **photo** of eye can be scanned
  - Attacker could use photo of eye
- Afghan woman was positively identified after 17 years by iris scan of old photo on National Geographic magazine cover in 1984
- To prevent attack, scanner could use light on the "eye" to be sure it is a "live" iris
  - increase the cost of the system
  - cost is always an issue

Equal Error Rate Comparison

- Equal error rate (EER): fraud == insult rate
- **Fingerprint** biometric has EER of about 5%, but cheap
- **Hand geometry** has EER of about 10^{-3}
- In theory, **iris scan** has EER of about 10^{-6}
  - But in practice, may be hard to achieve
  - Enrollment phase must be extremely accurate
- Most biometrics much worse than fingerprint!
- Biometrics useful for authentication…
  - …but identification biometrics almost useless today

Biometrics: The Bottom Line

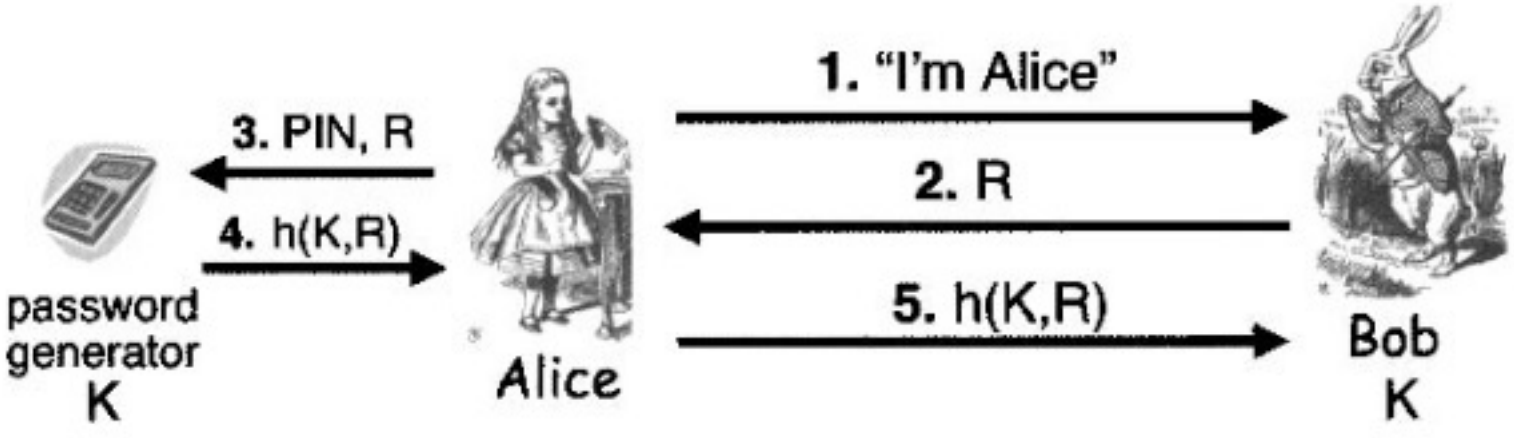
- Biometrics are hard to forge
- But attacker could
  - Photocopy Bob's fingerprint, eye, etc.
  - Subvert software, database, "trusted path" …
- And how to revoke a "broken" biometric?
  - Passwords can be revoked
- **Biometrics are not foolproof**

Authentication based on something you have

Something You Have

- Something in your possession
- Examples include following…
  - Car key
  - Laptop computer (or MAC address)
  - Password generator
  - ATM card, smartcard, etc.
  - Your phone

Password generator



- Alice wants to authenticate herself
- Bob sends random challenge R to Alice
- Alice inputs R, PIN into passwd generator
- Password generator produce a response
- Alice sends the response to Bob

2-factor Authentication

- Requires any 2 out of 3 of
  - Something you know
  - Something you have
  - Something you are
- Examples
  - Password generator:
    - PIN(something you know)
    - Generator(something you have)
  - ATM: Card and PIN
  - Credit card: Card and signature