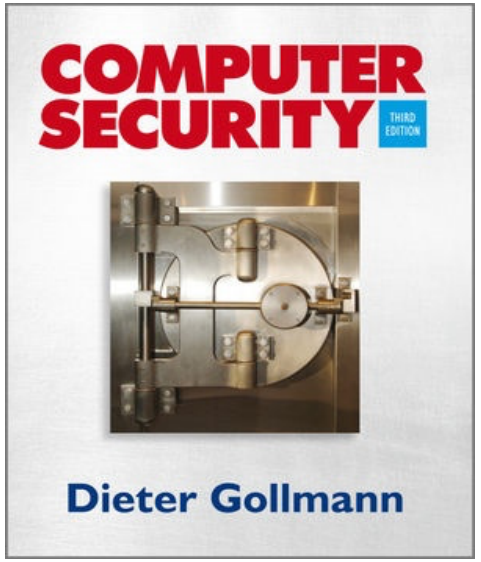
502049 – Introduction to Information Security

Chapter 3: Identification & Authentication



Agenda

- User authentication
- Identification & authentication
- Passwords
  - how to get the password to the user
  - forgotten passwords
  - password guessing
  - password spoofing
  - compromise of the password file
- Biometrics
- TOCTTOU

Introduction

- A secure system might have to track the identities of the users requesting its services.
- Authentication: process of verifying a user's identity.
- Two reasons for authenticating a user:
  - The user identity is a parameter in access control decisions.
  - The user identity is recorded when logging security relevant events in an audit trail.
- It is not always necessary or desirable to base access control on user identities.
- Much stronger case for using identities in audit logs.

Identification & Authentication

- When logging on to a computer you enter
  - user name and
  - password
- The first step is called identification:
  - You announce who you are.
- The second step is called authentication;
  - You prove that you are who you claim to be.
- To distinguish this type of 'authentication' from other interpretations, we refer here to user authentication: the process of verifying a claimed user identity.
- Authentication by password is widely accepted and not too difficult to implement.

Something You Have

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

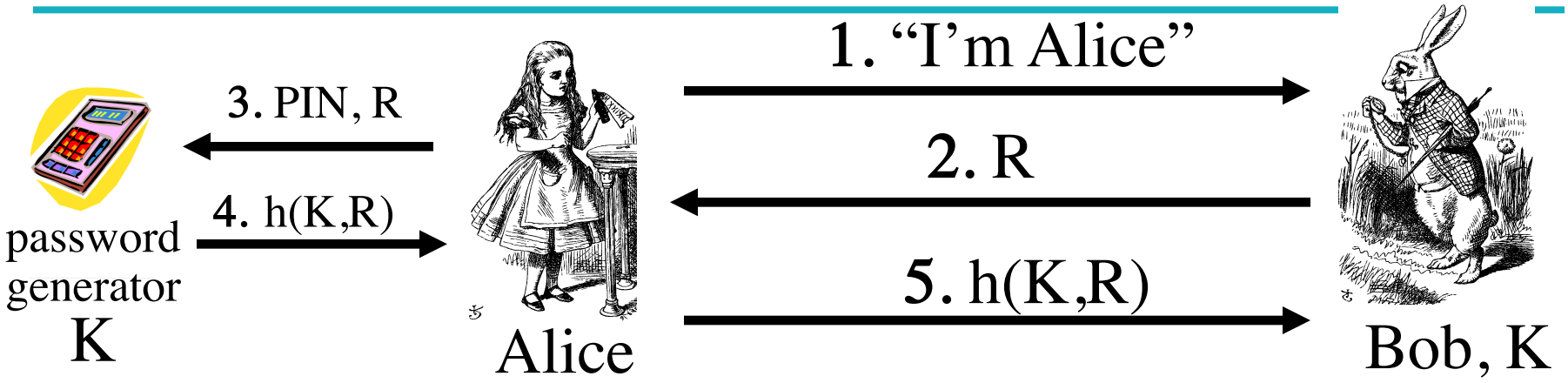
Are You Who You Say You Are?

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

Something You Know

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

Password Generator



- Alice receives random "challenge" R from Bob
- Alice enters PIN and R in password generator
- Password generator hashes symmetric key K with R
- Alice sends "response" h(K,R) back to Bob
- Bob verifies response
- Note: Alice has pwd generator and knows PIN

Trouble with Passwords

- “Passwords are one of the biggest practical problems facing security engineers today.”
- “Humans are incapable of securely storing high-quality cryptographic keys, and they have unacceptable speed and accuracy when performing cryptographic operations. (They are also large, expensive to maintain, difficult to manage, and they pollute the environment. It is astonishing that these devices continue to be manufactured and deployed.)”

Why Passwords?

- Why is “something you know” more popular than “something you have” and “something you are”?
- **Cost**: passwords are free
- **Convenience**: easier for sysadmin to reset pwd than to issue a new thumb

Keys vs Passwords

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

- **Passwords**
- Spse 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
  - incorrect
  - Pikachu
  - 102560
  - AustinStamp
- Good Passwords?
  - jflej,43j-EmmL+y
  - 09864376537263
  - P0kem0N
  - FSa7Yago
  - 0nceuP0nAt1m8
  - PokeGCTall150

Password Experiment

- User compliance hard to achieve
- In each case, 1/3rd did not comply
  - And about 1/3rd of those easy to crack!
- Assigned passwords sometimes best
- If passwords not assigned, best advice is…
  - Choose passwords based on passphrase
  - Use pwd cracking tool to test for weak pwds
- Require periodic password changes?

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
  - May only require **one** weak password!

Password Retry

- Suppose system locks after 3 bad passwords. How long should it lock?
  - 5 seconds
  - 5 minutes
  - Until SA restores service
- What are +'s and -'s of each?

Password File?

- Bad idea to store passwords in a file
- But we need to verify passwords
- Solution? **Hash** passwords
  - Store $y = h(\text{password})$
  - Can verify entered password by hashing
  - If Trudy obtains the password file, she does not (directly) obtain passwords
- But Trudy can try a *forward search*
  - Guess x and check whether $y = h(x)$

Dictionary Attack

- 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 of computing hashes in dictionary, actual attack is trivial
- Can we prevent this forward search attack? Or at least make it more difficult?

Salt

- Hash password with salt
- Choose random salt s and compute

$$y = h(\text{password}, s)$$

  and store (s,y) in the password file
- Note that the salt s is not secret
  - Analogous to IV
- Still easy to verify salted password
- But lots more work for Trudy
  - Why?

Password Cracking: Do the Math

- Assumptions:
- Pwds are 8 chars, 128 choices per character
  - Then $128^8 = 2^{56}$ possible passwords
- There is a **password file** with 2^{10} pwds
- Attacker has **dictionary** of 2^{20} common pwds
- **Probability** 1/4 that password is in dictionary
- **Work** is measured by number of hashes

Password Cracking: Case I

- Attack 1 specific password ***without*** using a dictionary
  - E.g., administrator's password
  - Must try $2^{56}/2 = 2^{55}$ on average
  - Like exhaustive key search
- Does **salt** help in this case?

Password Cracking: Case II

- Attack 1 specific password ***with*** dictionary
- With **salt**
  - Expected work: $1/4\ (2^{19}) + 3/4\ (2^{55}) \approx 2^{54.6}$
  - In practice, try all pwds in dictionary…
  - …then work is at most 2^{20} and probability of success is $1/4$
- What if **no salt** is used?
  - One-time work to compute dictionary: 2^{20}
  - Expected work is of same order as above
  - But with precomputed dictionary hashes, the "in practice" attack is essentially free…

Password Cracking: Case III

- Any of 1024 pwds in file, ***without*** dictionary
  - Assume all 2^{10} passwords are distinct
  - Need 2^{55} **comparisons** before expect to find pwd
- If **no salt** is used
  - Each computed hash yields 2^{10} comparisons
  - So expected work (hashes) is $2^{55}/2^{10} = 2^{45}$
- If **salt** is used
  - Expected work is 2^{55}
  - Each comparison requires a hash computation

Password Cracking: Case IV

- Any of 1024 pwds in file, ***with*** dictionary
  - Prob. one or more pwd in dict.: $1 - (3/4)^{1024} \approx 1$
  - So, we ignore case where no pwd is in dictionary
- If **salt** is used, expected work less than 2^{22}
  - See book, or slide notes for details
  - Work ≈ size of dictionary / P(pwd in dictionary)
- What if **no salt** is used?
  - If dictionary hashes not precomputed, work is about $2^{19}/2^{10} = 2^9$

Other Password Issues

- Too many passwords to remember
  - Results in password reuse
  - Why is this a problem?
- Who suffers from bad password?
  - Login password vs ATM PIN
- Failure to change default passwords
- Social engineering
- Error logs may contain “almost” passwords
- Bugs, keystroke logging, spyware, etc.

Passwords

- The bottom line…
- **Password attacks are too easy**
  - Often, one weak password will break security
  - Users choose bad passwords
  - Social engineering attacks, etc.
- Trudy has (almost) all of the advantages
- All of the math favors bad guys
- Passwords are a **BIG** security problem
  - And will continue to be a problem

Password Cracking Tools

- Popular password cracking tools
  - Password Crackers
  - Password Portal
  - L0phtCrack and LC4 (Windows)
  - John the Ripper (Unix)
- 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

Resetting Passwords

- When setting up a new user account some delay in getting the password may be tolerated.
- If you have forgotten your password but are in the middle of an important task you need instant help.
- Procedures for resetting passwords are the same as listed previously, but now reaction should be instant.
  - Global organisations must staff a hot desk round the clock,
  - On a web site, auxiliary information may authenticate a user: mother's maiden name, phone number, name of pet, …
- Password support can become a major cost factor.
- Staff at hot desk needs proper security training.

Lesson

- Security mechanisms may fail to give access to legitimate users.
- Your security solution must be able to handle such situations efficiently.

Guessing Passwords

- Exhaustive search (brute force): try all possible combinations of valid symbols up to a certain length.
- Intelligent search: search through a restricted name space, e.g. passwords that are somehow associated with a user like name, names of friends and relatives, car brand, car registration number, phone number,…, or try passwords that are generally popular.
- Typical example for the second approach: dictionary attack trying all passwords from an on-line dictionary.
- You cannot prevent an attacker from accidentally guessing a valid password, but you can try to reduce the probability of a password compromise.

Defences

- Set a password: if there is no password for a user account, the attacker does not even have to guess it.
- Change default passwords: often passwords for system accounts have a default value like “manager”.
  - Default passwords help field engineers installing the system; if left unchanged, it is easy for an attacker to break in.
  - Would it then be better to do without default passwords?
- Avoid guessable passwords:
  - Prescribe a minimal password length.
  - Password format: mix upper and lower case, include numerical and other non-alphabetical symbols.
  - Today on-line dictionaries for almost every language exist.

Defences

- Password ageing: set an expiry dates for passwords to force users to change passwords regularly.
- Prevent users from reverting to old passwords, e.g. keep a list of the last ten passwords used.
- Limit login attempts: the system can monitor unsuccessful login attempts and react by locking the user account (completely or for a given time interval) to prevent or discourage further attempts.
- Inform user: after successful login, display time of last login and the number of failed login attempts since, to warn the user about recently attempted attacks.

Password Security

- Is security highest if users are forced to use long passwords, mixing upper and lower case characters and numerical symbols, generated for them by the system, and changed repeatedly?
  - Users may have difficulty memorizing complex passwords.
  - Users may have difficulty dealing with frequent password changes.
  - Users may find ways of re-using their favourite password.
- Passwords will be written on a piece of paper kept close to the computer.
  - Security experts routinely look out for passwords on notes posted on computer terminals.
  - Is it always a bad idea to write down your password?

Password Security

- People are best at memorizing passwords they use regularly.
- Passwords work reasonably well in situations where they are entered quite frequently, but not so with systems used only occasionally.
- Good advice:
  - When changing a password, type it immediately several times.
  - Do not change passwords before weekends or holidays.

Lesson

- Don’t look at security mechanisms in isolation.
- Putting too much emphasis on one security mechanism may actually weaken the system.
- Users will find ways of circumventing security to be able to do their job properly.
- There is a trade-off between the complexity of passwords and the faculties of human memory.

Phishing and Spoofing

- Identification and authentication through username and password provide unilateral (đơn phương) authentication.
- Computer verifies the user's identity but the user has no guarantees about the identity of the party that has received the password.
- In phishing and spoofing attacks a party voluntarily sends the password over a channel, but is misled about the end point of the channel.

Spoofing Attacks

- Attacker starts a program that presents a fake login screen and leaves the computer.
- If the next user coming to this machine enters username and password on the fake login screen, these values are captured by the program.
  - Login is then typically aborted with a (fake) error message and the spoofing program terminates.
  - Control returned to operating system, which now prompts the user with a genuine login request.

Countermeasures

- Display number of failed logins: may indicate to the user that an attack has happened.
- Trusted path: guarantee that user communicates with the operating system and not with a spoofing program; e.g., Windows has a secure attention key CTRL+ALT+DEL for invoking the operating system logon screen.
- Mutual authentication: user authenticated to system, system authenticated to user.

Phishing

- Phishing: attacker impersonates the system to trick a user into releasing the password to the attacker.
  - E.g., a message could claim to come from a service you are using, tell you about an upgrade of the security procedures, and ask you to enter your username and password at the new security site that will offer stronger protection.
- Take care to enter your passwords only at the “right” site (but how do you know?)
- Social engineering: attacker impersonates the user to trick a system operator into releasing the password to the attacker.

Protecting the Password File

- Operating system maintains a file with user names and passwords
- Attacker could try to compromise the confidentiality or integrity of this password file.
- Options for protecting the password file:
  - cryptographic protection,
  - access control enforced by the operating system,
  - combination of cryptographic protection and access control, possibly with further measures to slow down dictionary attacks.

One-way Functions

- For cryptographic protection we can use one-way functions (cryptographic hash functions).
- Definition: A one-way function *f* is a function that is relatively easy to compute but hard to reverse.
  - Given an input *x* it is easy to compute *f*(*x*), but given an output *y* it is hard to find *x* so that *y* = *f*(*x*)
- Instead of the password *x*, the value *f*(*x*) is stored in the password file; when a user logs in entering a password *x'*, the system applies the one-way function *f* and compares *f*(*x'*) with the expected value *f*(*x*).

Password Salting

- Following common practice we refer to $f(x)$ as the encrypted password; to be precise, it is the hash of x. (More about encryption and hashing later.)
- To slow down dictionary attacks, a salt is appended to the password before encryption and stored with the encrypted password.
  - If two users have the same password, they will now have different entries in the file of encrypted passwords.
  - Example: Unix uses a 12 bit salt.

Access Control Settings

- Only privileged users must have write access to the password file.
  - Otherwise, an attacker could get access to the data of other users simply by changing their password, even if it is protected by cryptographic means.
- If read access is restricted to privileged users, then passwords in theory could be stored unencrypted.
- If password file contains data required by unprivileged users, passwords must be “encrypted”; such a file can still be used in dictionary attacks.
  - Typical example is **`/etc/passwd`** in Unix; many versions of Unix thus store encrypted passwords in a shadow password file that is not publicly accessible.

Lesson

- You have seen examples for two security design principles.
- Combining mechanisms can enhance protection.
  - Use of encryption and access control to guard password files.
- Separate security relevant data from data that should be openly available.
  - In Unix, **`/etc/passwd`** contains both types of data; shadow password files achieve the desired separation.

Caching Passwords

- Our description of login has been quite abstract: password travels directly from user to the password checking routine.
- In reality, it will be held temporarily in intermediate storage locations like buffers, caches, or a web page.
- The management of these storage locations is normally beyond the control of the user; a password may be kept longer than the user has bargained for.

Single Sign-on

- Having to remember many passwords for different services is a nuisance; with a single sign-on service, you have to enter your password only once.
- A simplistic single-sign on service could store your password and do the job for you whenever you have to authenticate yourself.
  - Such a service adds to your convenience but it also raises new security concerns.
- System designers have to balance convenience and security; ease-of-use is an important factor in making IT systems really useful, but many practices which are convenient also introduce new vulnerabilities.

Single Sign-on

- A hassle to enter password(s) repeatedly
  - Alice would like to authenticate only once
  - “Credentials” stay with Alice wherever she goes
  - Subsequent authentications transparent to Alice
- Kerberos — a single sign-on protocol
- Single sign-on for the Internet?
  - Microsoft: Passport
  - Everybody else: Liberty Alliance
  - Security Assertion Markup Language (SAML)

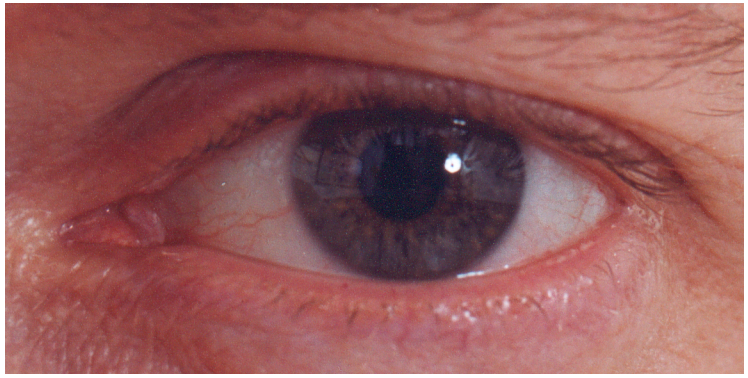
Web Cookies

- Cookie is provided by a Website and stored on user's machine
- Cookie indexes a database at Website
- Cookies **maintain state** across sessions
  - Web uses a stateless protocol: HTTP
  - Cookies also maintain state within a session
- Sorta like a single sign-on for a website
  - But, very, very weak form of authentication
- Cookies also create privacy concerns

2-factor Authentication

- Requires any 2 out of 3 of
  - Something you **know**
  - Something you **have**
  - Something you **are**
- Examples
  - ATM: Card and PIN
  - Credit card: Card and signature
  - Password generator: Device and PIN
  - Smartcard with password/PIN

Biometrics



Something You Are

- Biometric
  - **"You are your key"** — Schneier
- Examples
  - Fingerprint
  - Handwritten signature
  - Facial recognition
  - Speech recognition
  - Gait (walking) recognition
  - "Digital doggie" (odor recognition)
  - Many more!

Why Biometrics?

- May be better than passwords
- But, 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.
- But biometrics not really that popular
  - Has not lived up to its promise/hype (yet?)

Ideal Biometric

- **Universal** — applies to (almost) everyone
  - In reality, no biometric applies to everyone
- **Distinguishing** — distinguish with certainty
  - In reality, cannot hope for 100% certainty
- **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, and ???

Identification vs Authentication

- **Identification** — Who goes there?
  - Compare **one-to-many**
  - Example: FBI fingerprint database
- **Authentication** — Are you who you say you are?
  - Compare **one-to-one**
  - Example: Thumbprint mouse
- Identification problem is more difficult
  - More “random” matches since more comparisons
- We are (mostly) 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 a weak point in real-world use
- 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 (also, terrorists in airports, etc.)
  - Often, less than ideal enrollment conditions
  - Subject will try to confuse recognition phase
- Cooperative subject makes it much easier
  - We are focused on authentication
  - So, we can assume subjects are 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 ⇒ low fraud, high insult
  - 30% voiceprint match ⇒ high fraud, low insult
- **Equal error rate:** rate where fraud == insult
  - A way to ***compare*** different biometrics

Fingerprint History

- 1823 — Professor Johannes Evangelist Purkinje discussed 9 fingerprint patterns
- 1856 — Sir William Hershel used fingerprint (in India) on contracts
- 1880 — Dr. Henry Faulds article in *Nature* about fingerprints for ID
- 1883 — Mark Twain's *Life on the Mississippi* (murderer ID'ed by fingerprint)

Fingerprint History

- 1888 — Sir Francis Galton developed classification system
  - His system of “minutia” can be used today
  - Also verified that fingerprints do not change
- Some countries require fixed number of “points” (minutia) to match in criminal cases
  - In Britain, at least 15 points
  - In US, no fixed number of points

Fingerprint Comparison

- Examples of **loops**, **whorls**, and **arches**
- Minutia extracted from these features

Loop (double)

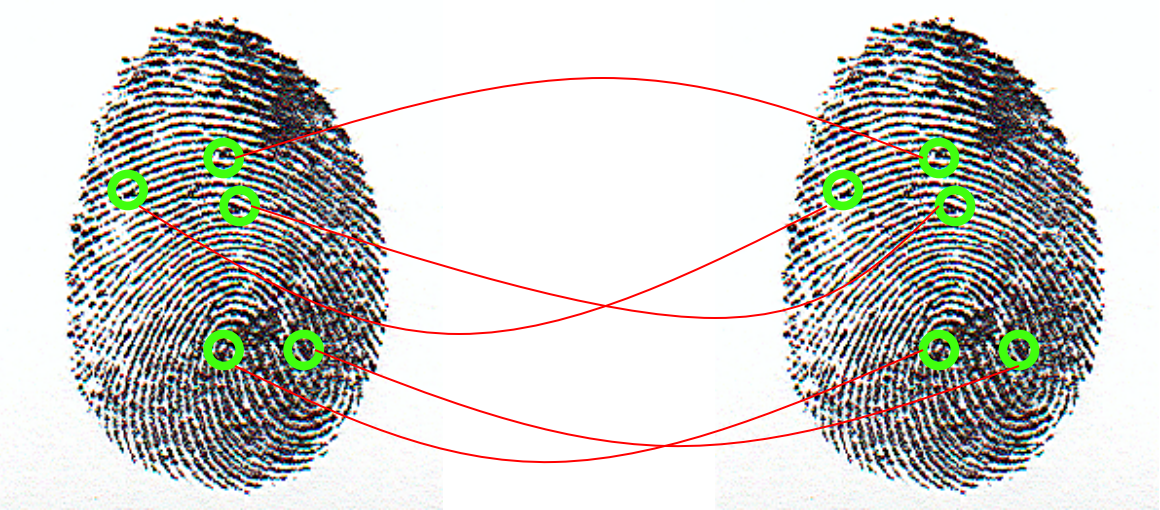
Whorl

Arch

Fingerprint: Enrollment

- Capture image of fingerprint
- Enhance image
- Identify “points”

Fingerprint: Recognition



- Extracted points are compared with information stored in a database
- Is it a statistical match?
- Aside: Do identical twins' fingerprints differ?

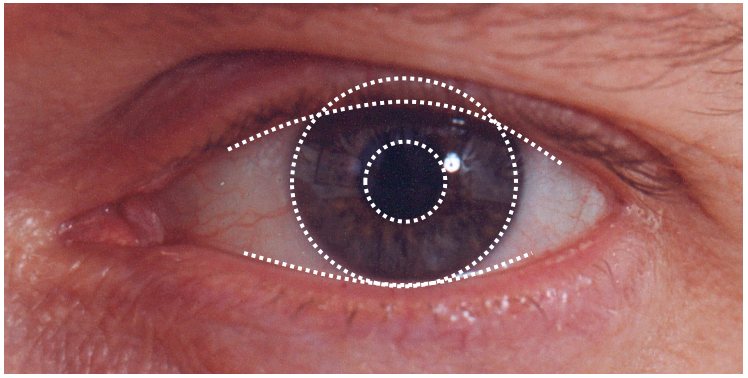
Hand Geometry

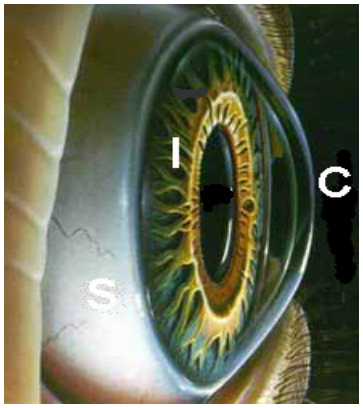
- A popular biometric
- Measures shape of hand
  - Width of hand, fingers
  - Length of fingers, etc.
- Human hands not so unique
- Hand geometry sufficient for many situations
- OK for authentication
- Not useful for ID problem

Hand Geometry

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

Iris Patterns





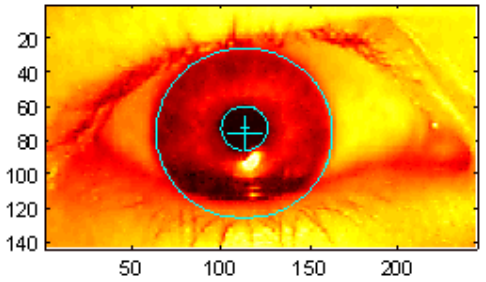
- Iris pattern development is “chaotic”
- Little or no genetic influence
- Even for identical twins, uncorrelated
- Pattern is stable through lifetime

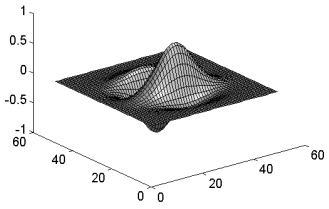
Iris Recognition: History

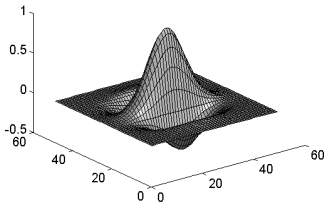
- 1936 — suggested by ophthalmologist
- 1980s — James Bond film(s)
- 1986 — first patent appeared
- 1994 — John Daugman patents new-and-improved technique
  - Patents owned by Iridian Technologies

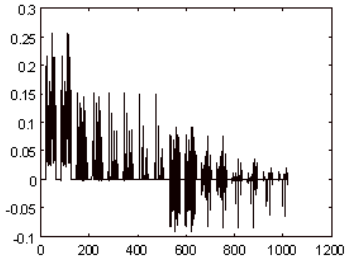
Iris Scan

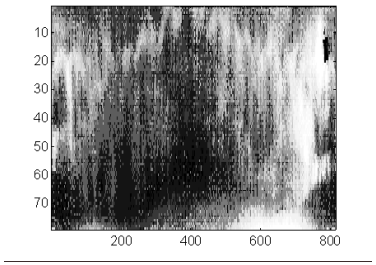
- Scanner locates iris
- Take b/w photo
- Use polar coordinates…
- 2-D wavelet transform
- Get 256 byte iris code











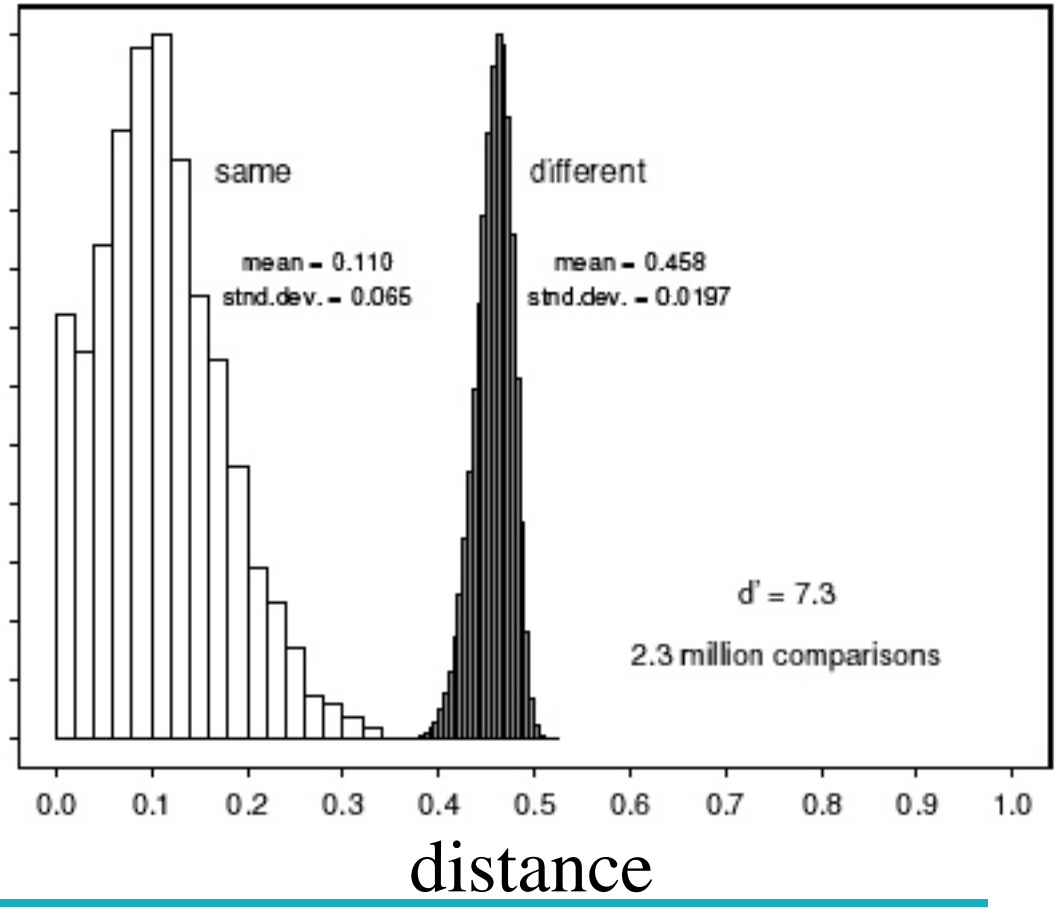
Measuring Iris Similarity

- 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

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 $1.3*10^{5}$

== equal error rate



Attack on Iris Scan

- Good **photo** of eye can be scanned
  - Attacker could use photo of eye

- Afghan woman was authenticated by iris scan of old photo
  - Story can be found here
- To prevent attack, scanner could use light to be sure it is a "live" iris

Equal Error Rate Comparison

- Equal error rate (EER): fraud == insult rate
- **Fingerprint** biometrics used in practice have EER ranging from about 10^{-3} to as high as 5%
- **Hand geometry** has EER of about 10^{-3}
- In theory, **iris scan** has EER of about 10^{-6}
  - Enrollment phase may be critical to accuracy
- Most biometrics much worse than fingerprint!
- Biometrics useful for authentication…
  - …but for identification, not so impressive today

Biometrics: The Bottom Line

- Biometrics are hard to forge
- But attacker could
  - Steal Alice's thumb
  - Photocopy Bob's fingerprint, eye, etc.
  - Subvert software, database, "trusted path" …
- And how to revoke a "broken" biometric?
- **Biometrics are not foolproof**
- Biometric use is relatively limited today
- That should change in the (near?) future

More on Authentication

- If you are dissatisfied with the level of security provided by passwords, what else can you do?
- In general, the following options are open.
- You can be authenticated on the basis of
  - something you know,
  - something you hold,
  - who you are,
  - what you do,
  - where you are.

Identification & Verification

- Biometrics are used for two purposes:
  - Identification: 1:*n* comparison tries to identify the user from a database of *n* persons.
  - Verification: 1:1 comparison checks whether there is a match for a given user.
- Authentication by password: clear reject or accept at each authentication attempt.
- Biometrics: stored reference features will hardly ever match precisely features derived from the current measurements.

Failure Rates

- Measure similarity between reference features and current features.
- User is accepted if match is above a predefined threshold.
- New issue: false positives and false negatives
- Accept wrong user (false positive): security problem.
- Reject legitimate user (false negative): creates embarrassment and an inefficient work environment.

Technology Analysis

- Based on a (given) databases of biometric samples.
- Measures performance of the algorithms extracting and comparing biometric features.
- False match rate (FMR):

$$FMR = \frac{\text{number of successful false matches}}{\text{number of attempted false matches}}$$

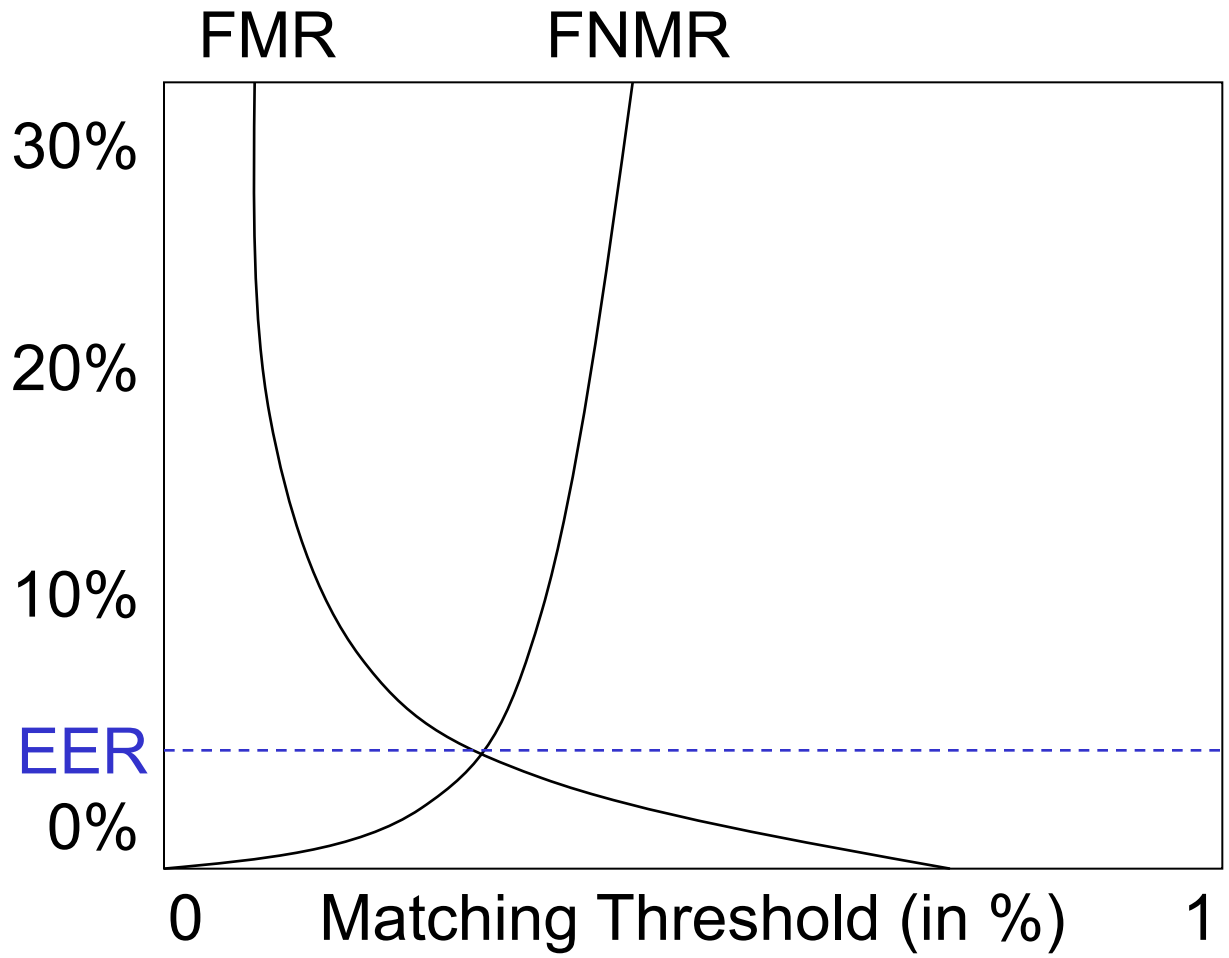
- False non-match rate (FNMR):

$$FNMR = \frac{\text{number of rejected genuine matches}}{\text{number of attempted genuine matches}}$$

Equal-error Rate

- By setting the matching threshold, we can trade off a lower false match rate against a higher false non-match rate, and vice versa.
- Finding the right balance between those two errors depends on the application.
- Equal error rate (EER): given by the threshold value where FMR and FNMR are equal.
- Currently, the best state-of-the-art fingerprint recognition schemes have an EER of about 0.5 - 2%.
- Iris pattern recognition has a superior performance.
- State of the art: http://atvs.ii.uam.es/fvc2006.html.

FMR, FNMR, EER



Scenario Analysis

- Records error rates in actual field trials; measures performance of fingerprint reader (hardware and software) capturing templates at log-in time.
  - Failure-to-capture rate (FTC): frequency of failing to capture a sample.
  - Failure-to-extract rate (FTX): frequency of failing to extract a feature from a sample.
- Failure-to-acquire rate: frequency of failing to acquire a biometric feature: FTA = FTC + FTX·(1 - FTC)
- False accept rate for the entire biometric scheme: FAR = FMR · (1 - FTA).
- False reject rate: FRR = FTA + FNMR · (1 - FTA).

Madrid Error (2004)

- False positive identification rate for a database with *n* persons:
  - FPIR = (1 - FTA) · (1 - (1 - FMR)^n).
- A fingerprint found in the Madrid train bombing was compared against a database of 530 million entries.
- A match was found and linked by four experts with 100% confidence to a US citizen (B. Mayfield).
- They were wrong: the guy did not even have a passport and had not left the country.
- Criteria for matching features had to be re-appraised.
  - http://www.henrytempleman.com/madrid_error, http://www.onin.com/fp/problemidents.html

Forged Fingers

- Fingerprints, and biometric traits in general, may be unique but they are no secrets.
- You are leaving your fingerprints in many places.
- Rubber fingers have defeated many a commercial fingerprint recognition systems in the past.
  - Minor issue if authentication takes place in the presence of security personnel.
  - When authenticating remote users additional precautions have to be taken to counteract this type of fraud.
- User acceptance: so far fingerprints have been used for tracing criminals.

What You Do

- People perform mechanical tasks in a way that is both repeatable and specific to the individual.
- Experts look at the dynamics of handwriting to detect forgeries.
- Users could sign on a special pad that measures attributes like writing speed and writing pressure.
- On a keyboard, typing speed and key strokes intervals can be used to authenticate individual users.

Where You Are

- Some operating systems grant access only if you log on from a certain terminal.
  - A system manager may only log on from an operator console but not from an arbitrary user terminal.
  - Users may be only allowed to log on from a workstation in their office.
- Decisions of this kind will be even more frequent in mobile and distributed computing.
- Global Positioning System (GPS) might be used to established the precise geographical location of a user during authentication.

ĐẠI HỌC TÔN ĐỨC THẮNG
TON DUC THANG UNIVERSITY

Authorization

Authentication vs Authorization

- Authentication — Are you who you say you are?
  - Restrictions on who (or what) can access system
- **Authorization** — Are you allowed to do that?
  - Restrictions on actions of authenticated users
- Authorization is a form of **access control**
- But first, we look at system certification…

System Certification

- Government attempt to certify “security level” of products
- Of historical interest
  - Sorta like a history of authorization
- Still important today if you want to sell a product to the government
  - Tempting to argue it’s a failure since government is so insecure, but…

Orange Book

- Trusted Computing System Evaluation Criteria (TCSEC), 1983
  - Universally known as the “orange book”
  - Name is due to color of it’s cover
  - About 115 pages
  - Developed by U.S. DoD (NSA)
  - Part of the “rainbow series”
- Orange book generated a pseudo-religious fervor among some people
  - Less and less intensity as time goes by

Orange Book Outline

- Goals
  - Provide way to assess security products
  - Provide general guidance/philosophy on how to build more secure products
- Four ***divisions*** labeled D thru A
  - D is lowest, A is highest
- Divisions split into numbered ***classes***

D and C Divisions

- D — minimal protection
  - Losers that can't get into higher division
- C — discretionary protection, i.e., don't enforce security, just have means to detect breaches (audit)
  - C1 — discretionary security protection
  - C2 — controlled access protection
  - C2 slightly stronger than C1 (both vague)

B Division

- B — mandatory protection
- B is a ***huge*** step up from C
  - C: break security, you might get caught
  - B: “mandatory”, so you can’t break it
- B1 — labeled security protection
  - All data labeled, which restricts what can be done with it
  - This access control cannot be violated

B and A Divisions

- B2 — structured protection
  - Adds covert channel protection onto B1
- B3 — security domains
  - On top of B2 protection, adds that code must be ***tamperproof*** and “small”
- A — verified protection
  - Like B3, but ***proved*** using formal methods
  - Such methods still (mostly) impractical

Orange Book: Last Word

- Also a 2nd part, discusses rationale
- Not very practical or sensible
- But some people insist we'd be better off if we'd followed it
- Others think it was a dead end
  - And resulted in lots of wasted effort
  - Aside… people who made the orange book, now set security education standards

Common Criteria

- Successor to the orange book (ca. 1998)
  - Due to inflation, more than 1000 pages
- An international government standard
  - And it reads like it…
  - Won't ever stir same passions as orange book
- CC is relevant in practice, but usually only if you want to sell to the government
- Evaluation Assurance Levels (EALs)
  - 1 thru 7, from lowest to highest security

EAL

- Note: product with high EAL may not be more secure than one with lower EAL
  - Why?
- Similarly, product with an EAL may not be any more secure than one without
  - Why?

EAL 1 thru 7

- EAL1 — functionally tested
- EAL2 — structurally tested
- EAL3 — methodically tested, checked
- EAL4 — ***designed***, tested, reviewed
- EAL5 — semiformally designed, tested
- EAL6 — verified, designed, tested
- EAL7 — formally … (blah blah blah)

Common Criteria

- EAL4 is most commonly sought
  - Minimum needed to sell to government
- EAL7 requires formal proofs
  - Author could only find 2 EAL7 products…
- Who performs evaluations?
  - Government accredited labs, of course (for a hefty fee, like 6 figures)

Summary: Security Mechanisms Fail

- You may be worried about failures that wrongly permit an action.
- You should be equally worried about failures that wrongly deny access.
- Forgotten passwords, false biometric rejection: system not available to legitimate users.
- You have to implement measures that deal with such failures; this may be quite expensive.
- Even worse, you may believe that technology is perfect (or forget about this issue) and your system may fail in quite damaging ways.