Information Security
Identification and authentication

Advanced User Authentication III

2015-02-06

Amund Hunstad

Guest Lecturer, amund@foi.se

LiU expanding reality

Agenda for lecture II within this part of the course

Background

Statistics in user authentication

Biometric systems

Tokens

Statistics✔

Generic biometric system✔

Design cycle✔

Multibiometrics

Security threats✔

Attacks

A. Jain, A. Ross and K. Nandakumar, Chapters 1, 6 & 7 in "Introduction to Biometrics”

Agenda for lecture III within this part of the course

Background

Statistics in user authentication

Biometric systems

Tokens

Attacks

Multibiometrics

Fingerprints

Iris

Face etc

Attacks on tokens

A. Jain, A. Ross and K. Nandakumar, Chapters 6 & 7, 2-5 in "Introduction to Biometrics"

Ross Anderson, Security Engineering, Chapter 16

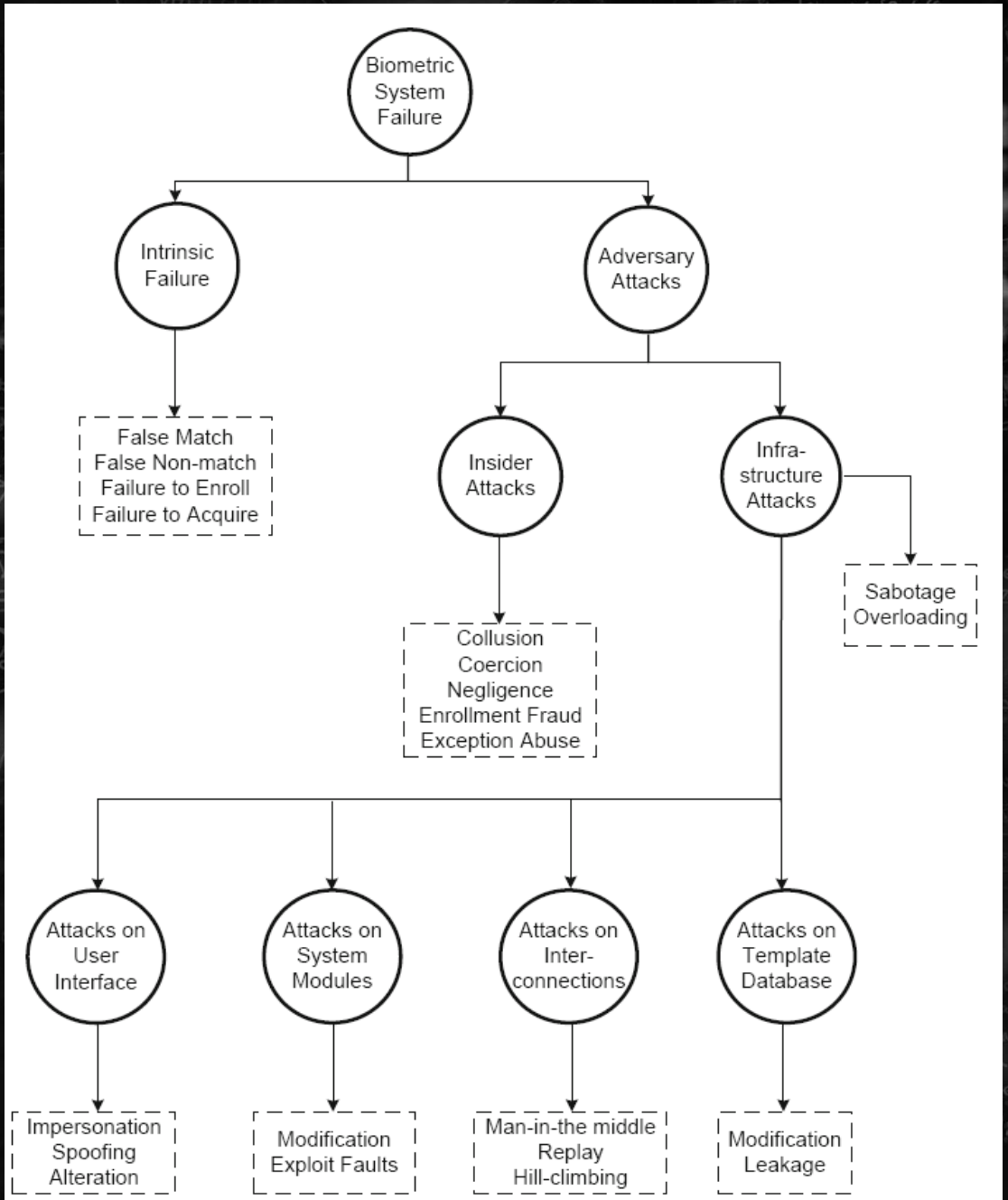
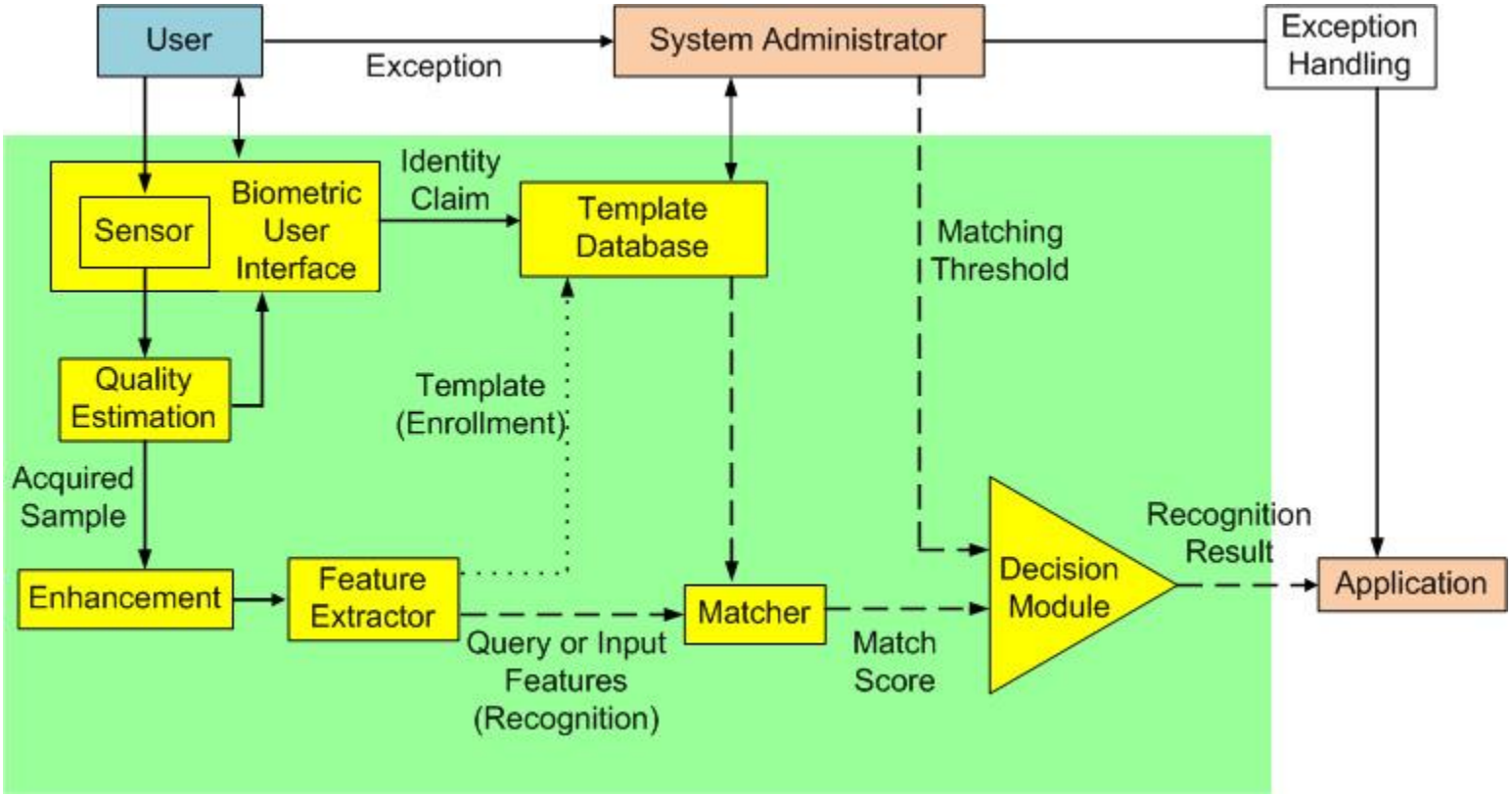


Fig. 7.2 Taxonomy of attacks that can be mounted against a biometric system.

Generic biometric system: Building blocks



Types of adversary attacks

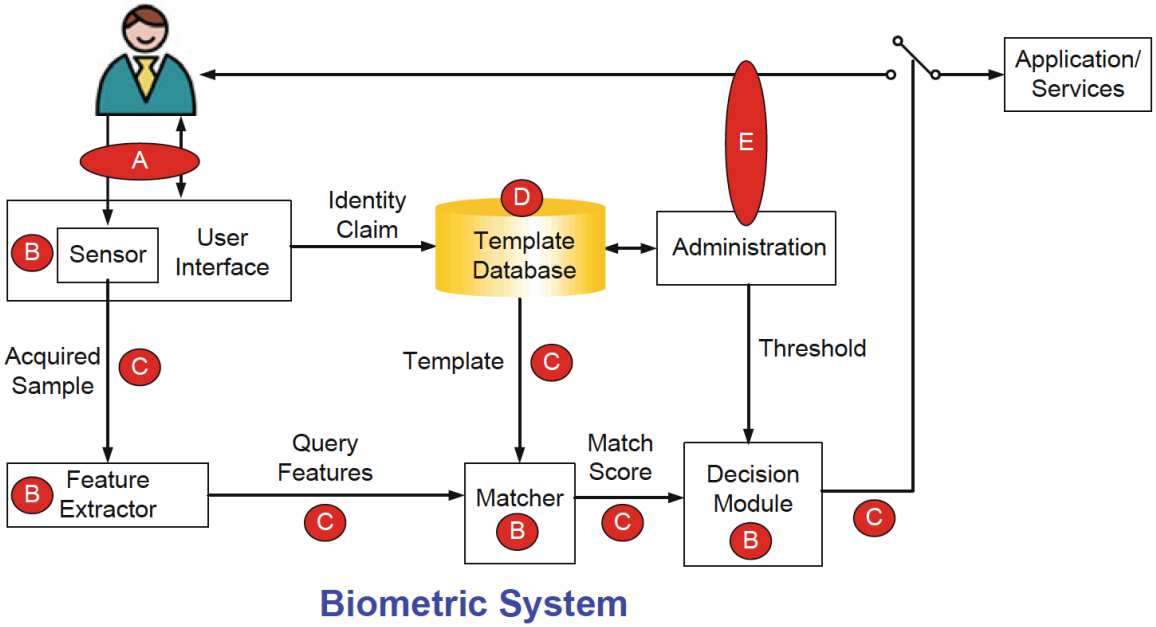
A: User-biometric system interface

B: Biometric system modules

C: Interconnections betweeen biometric modules

D: Templates database

E: Attacks through insiders (admin or enrolled users)



Biometric System

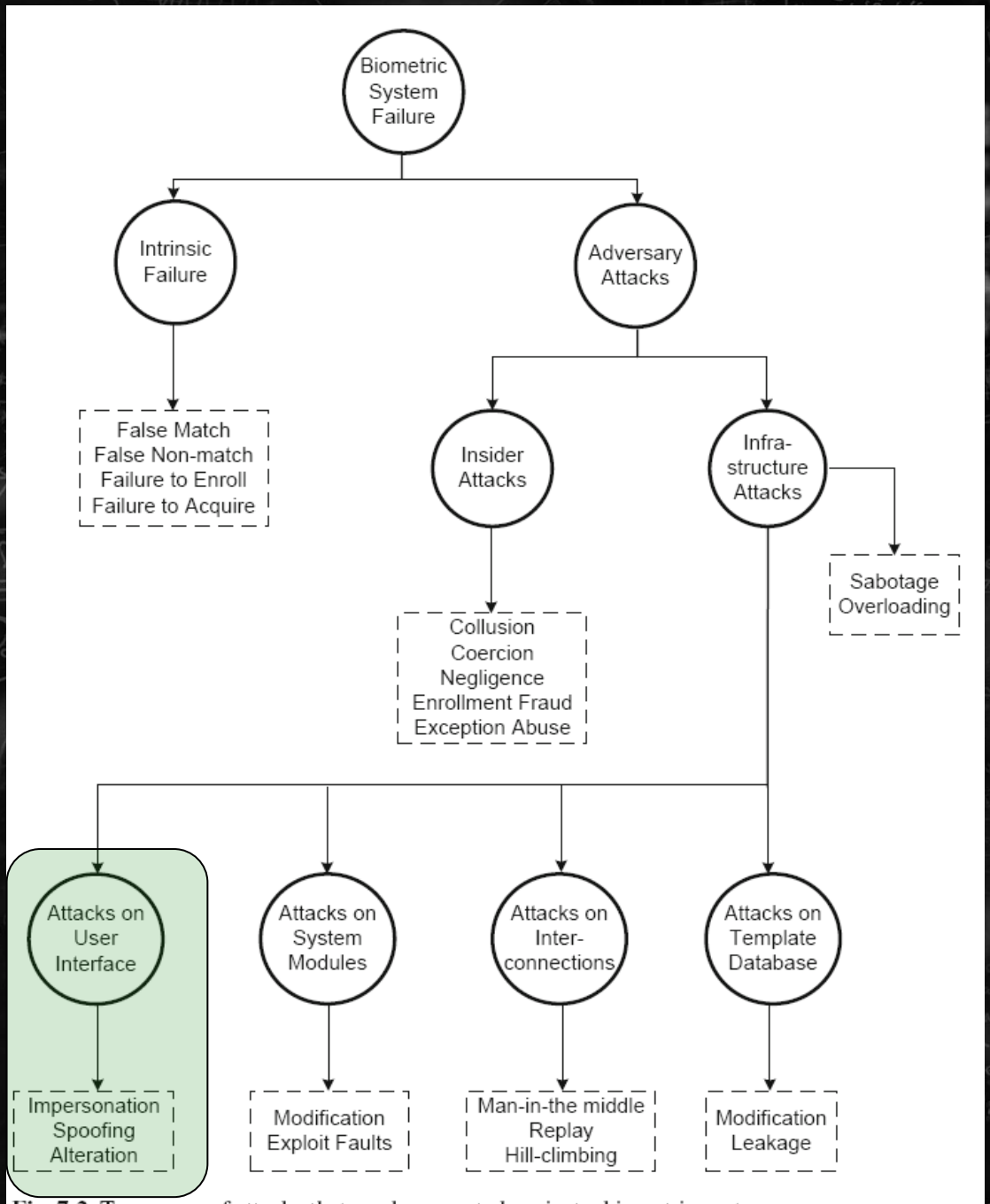
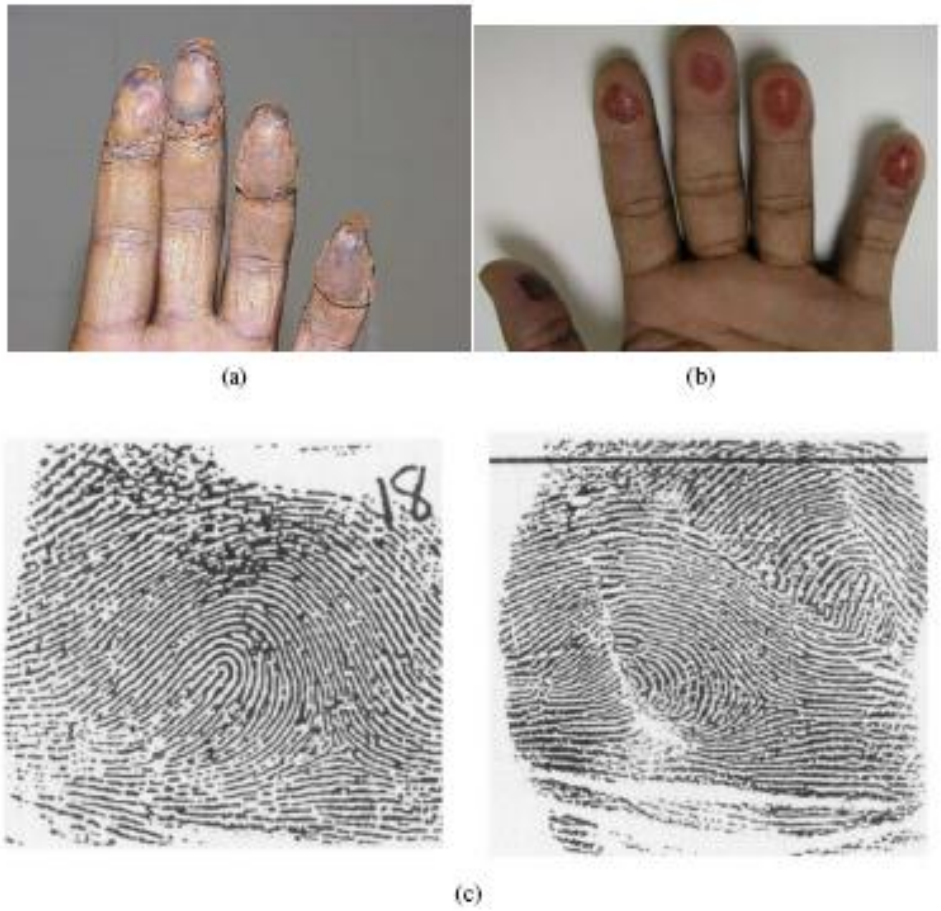


Fig. 7.2 Taxonomy of attacks that can be mounted against a biometric system.

Attacks at the user interface: Obfuscation



Attacks at the user interface: Spoofing

(a)

(b)

(c)

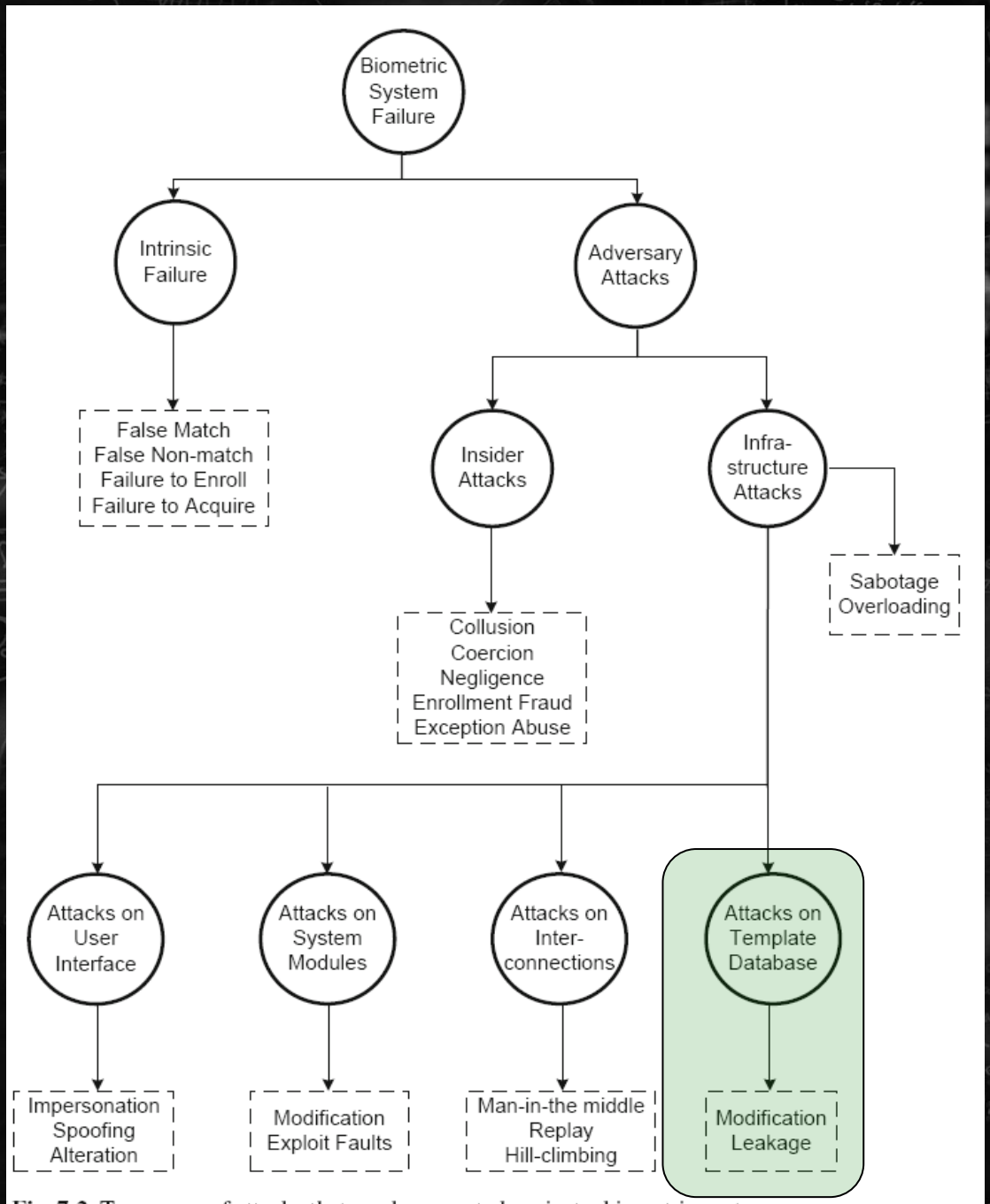


Fig. 7.2 Taxonomy of attacks that can be mounted against a biometric system.

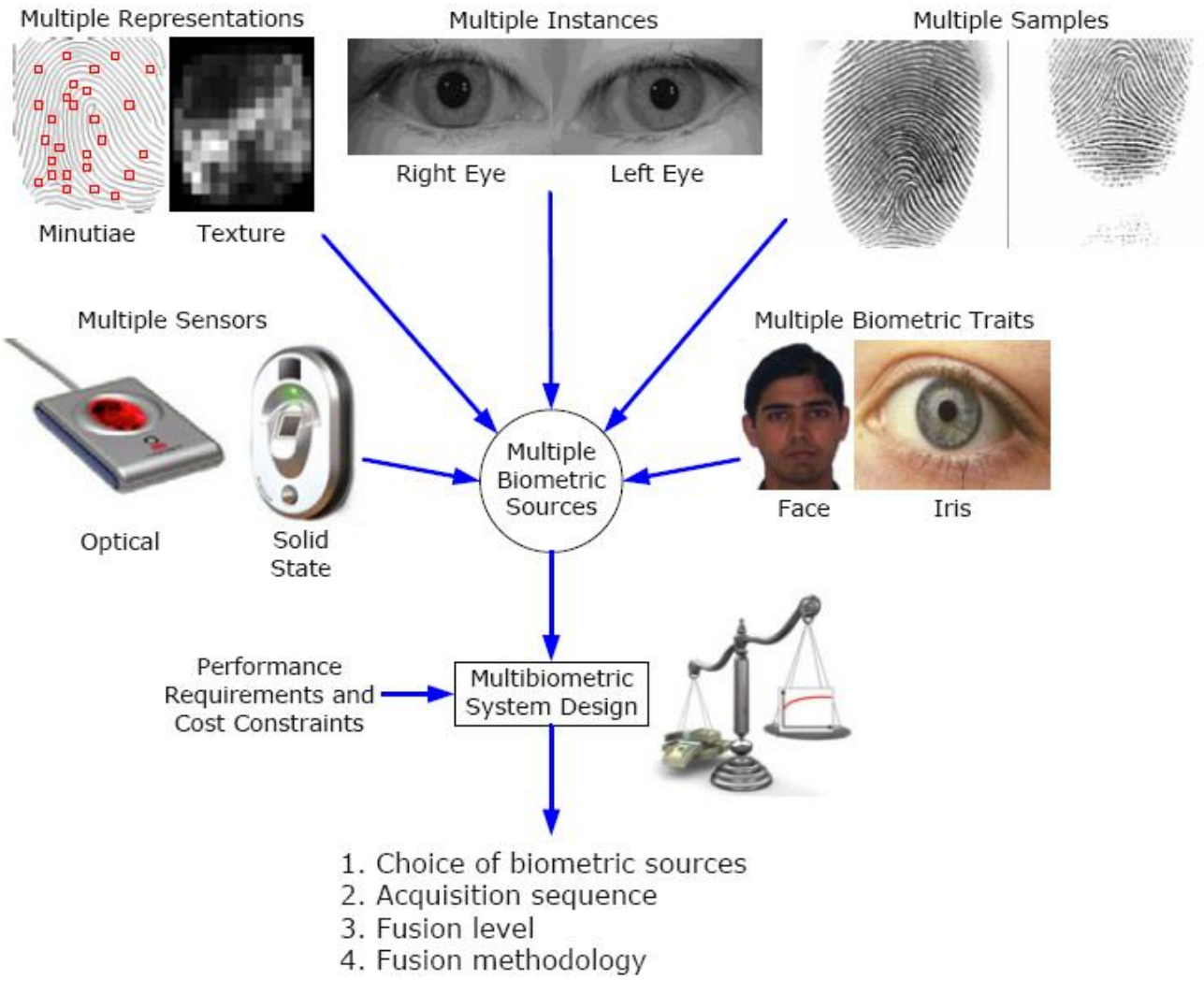
Attacks on the template database

- Gain unauthorized access/Deny access to legitimate users
- Leakage: Stored biometric templates available to adversaries
  - Password-based authentication: Hashed,minor problem
  - Biometrics based: Major problem
    - Biometrics not always secret
    - Physical link user/biometric trait

Attacks on the template database: Leakage

- Obtain biometric & biographic info about large number of users
- Reverse engineer template: Physical spoof
- Replay attack
- Compromised biometric traits: Not possible to replace
- Undermines privacy

Multibiometrics



Multibiometrics: Why?

- More unique (than single)
- Compensate noise, imprecision, inherent drift
- Redundancy
- Fault-tolerance
- Flexibility
- Increase resistance to spoofing
- But: Expensive – Tradeoff cost/benefits

Multi-modal systems

Use two or more different biometric features

AND or OR requirements for each feature

AND increases accuracy and thus protects against false acceptance

OR opens more options and thus protects against too much false rejection

OR is necessary in order to accommodate for physical handicaps

Multiple methods

Use of two or three of the basic categories (what you “know”, “hold” and “are”).

Thus use of something you know or hold in addition to biometrics (or just something you know and something you hold)

Examples:

PIN + card

Fingerprints + card with fingerprint template

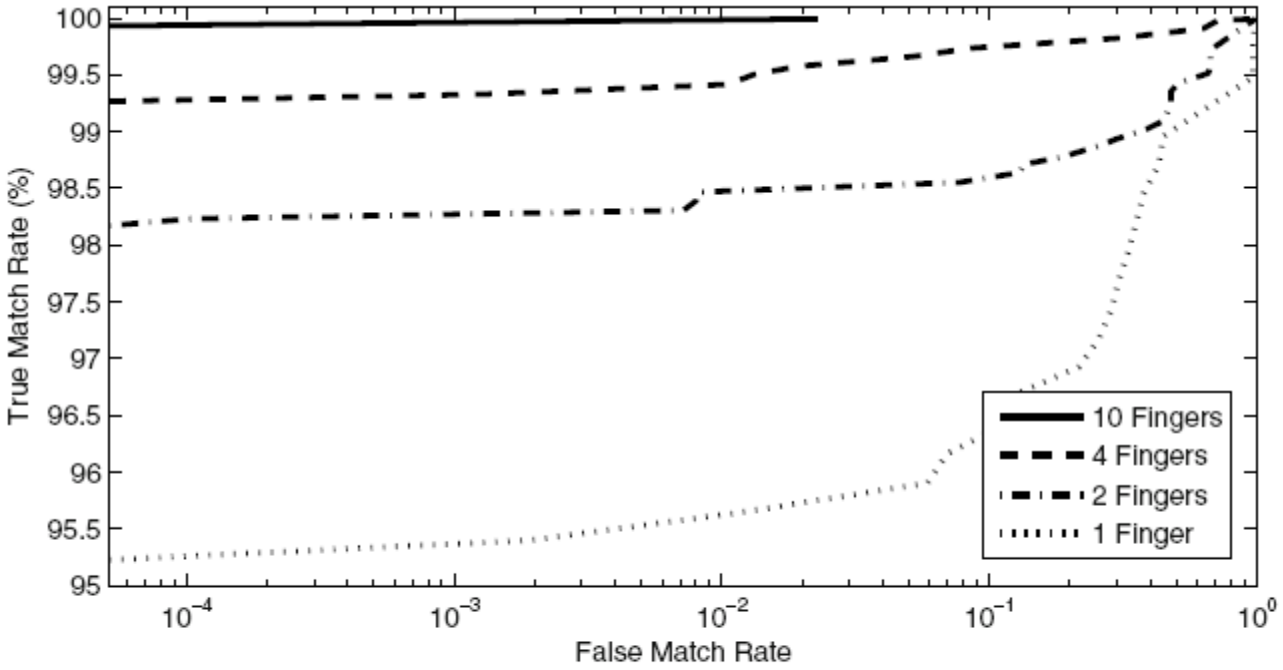
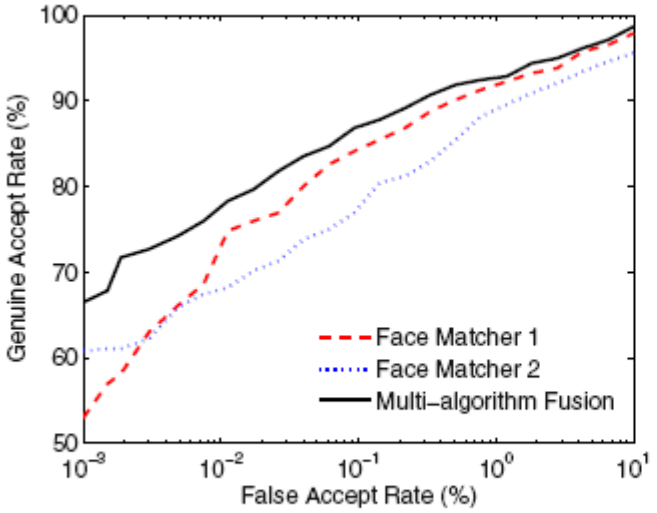
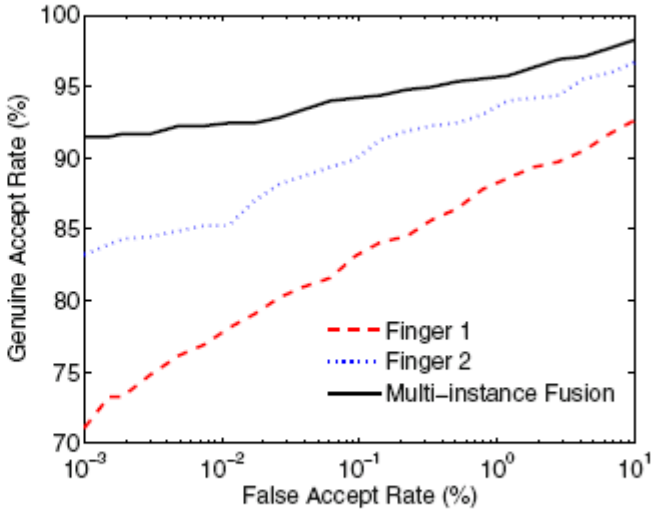


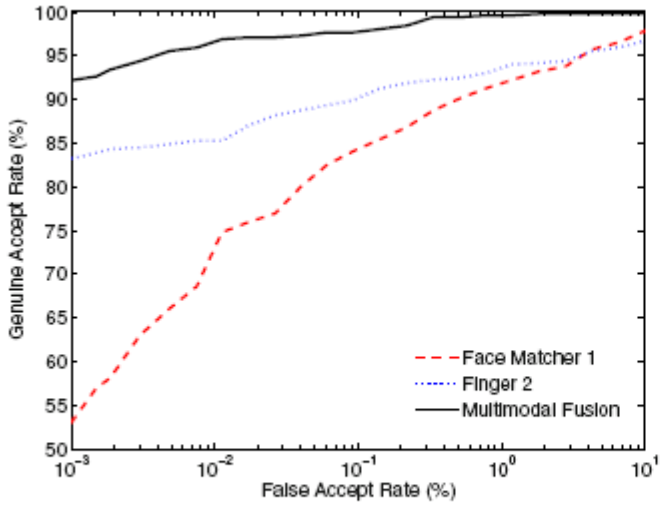
Fig. 6.7 Accuracy improvement in a multi-instance fingerprint verification system. This example extracted from FpVTE 2003 [37] shows that fusion of evidence from the multiple fingers leads to significant improvement in the accuracy.


Genuine Accept Rate (%)
False Accept Rate (%)
Face Matcher 1
Face Matcher 2
Multi–algorithm Fusion

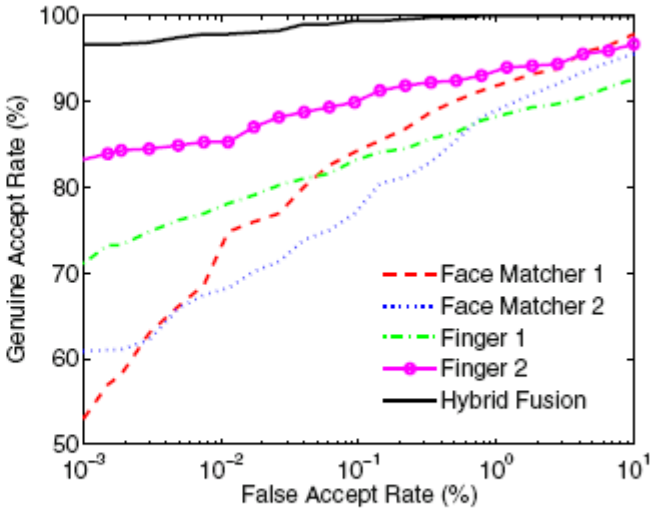
(a)


Genuine Accept Rate (%)
False Accept Rate (%)
Finger 1
Finger 2
Multi–instance Fusion

(b)


Genuine Accept Rate (%)
False Accept Rate (%)
Face Matcher 1
Finger 2
Multimodal Fusion

(c)


Genuine Accept Rate (%)
False Accept Rate (%)
Face Matcher 1
Face Matcher 2
Finger 1
Finger 2
Hybrid Fusion

(d)

GunVault Speedvault Biometric Pistol Safe SVB500

A unique design that really works! It is a safe that will stop kids and honest adults from getting the gun while keeping it ready to use if needed, but it is not designed to stop a determined attack.

”… they use a person’s fingerprint to open the safe”

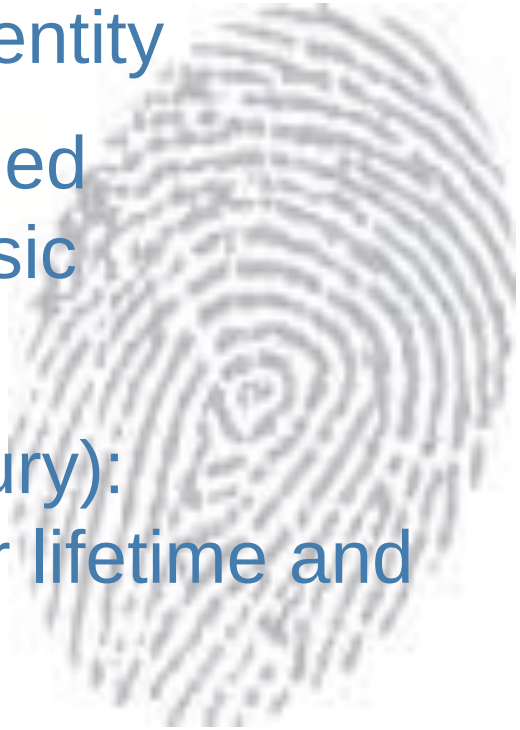
”Since no two people have the same fingerprint pattern, the system is a hundred percent effective”

Fingerprints - history

Already in ancient times fingerprints were used to denote authorship or identity

In 1823 a Czech physician classified fingerprint patterns into nine basic types

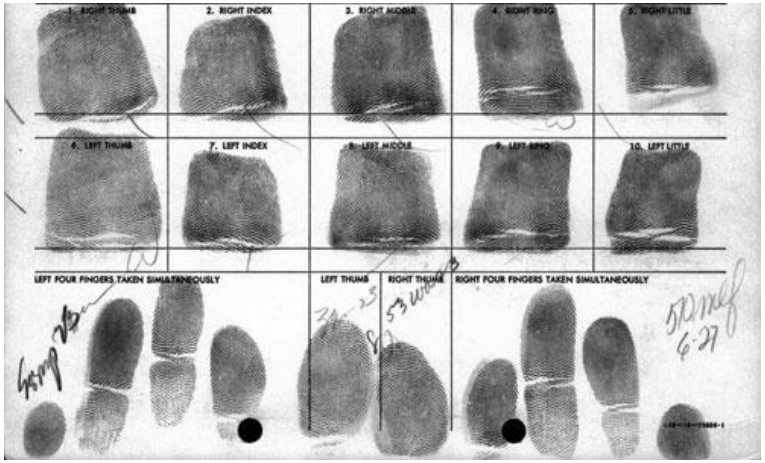
Sir Francis Galton (late 19th century): Fingerprints do not change over lifetime and that no two fingerprints are exactly alike



Fingerprints - history

In 1901 fingerprints were introduced for criminal identification in England and Wales

The first fingerprint scanners were introduced more than 30 years ago



AFIS installation at Michigan State Police facility. This system was first installed in 1989; the database has 3.*2 million tenprint cards and performs 700,000 searches each year*

Example: Fingerprints

Known and used with formal classification since 19th century.

Scanners were introduced more than 30 years ago.

Cheap readers that are easy to handle

High uniqueness

Fairly easy to make copies

Fingerprints - characteristics

Papillary lines
- ridges
- valleys

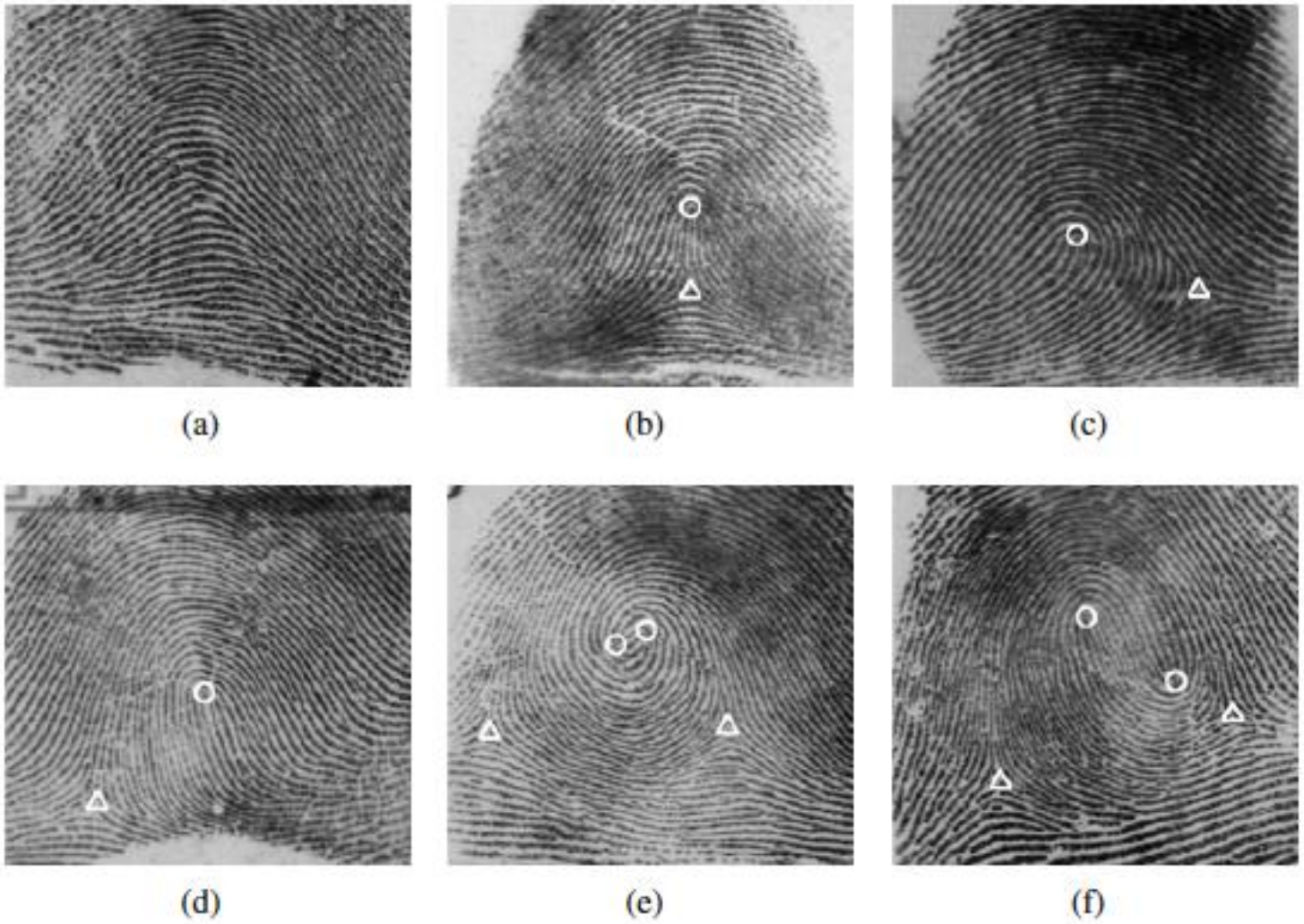


Fig. 2.8 Major fingerprint pattern types. (a) Plain arch, (b) tented arch, (c) left loop, (d) right loop, (e) whorl, and (f) twin loop. A loop is denoted by a circle and a delta is denoted by a triangle. Loop- and whorl-type of fingerprints are found most commonly; about 65% of fingerprints belong to loop-type, and 24% are whorl-type [52]. Twin loop, arch and tented arch account for approximately 4%, 4% and 3% of the fingerprints, respectively.

3 levels of fingerprint features

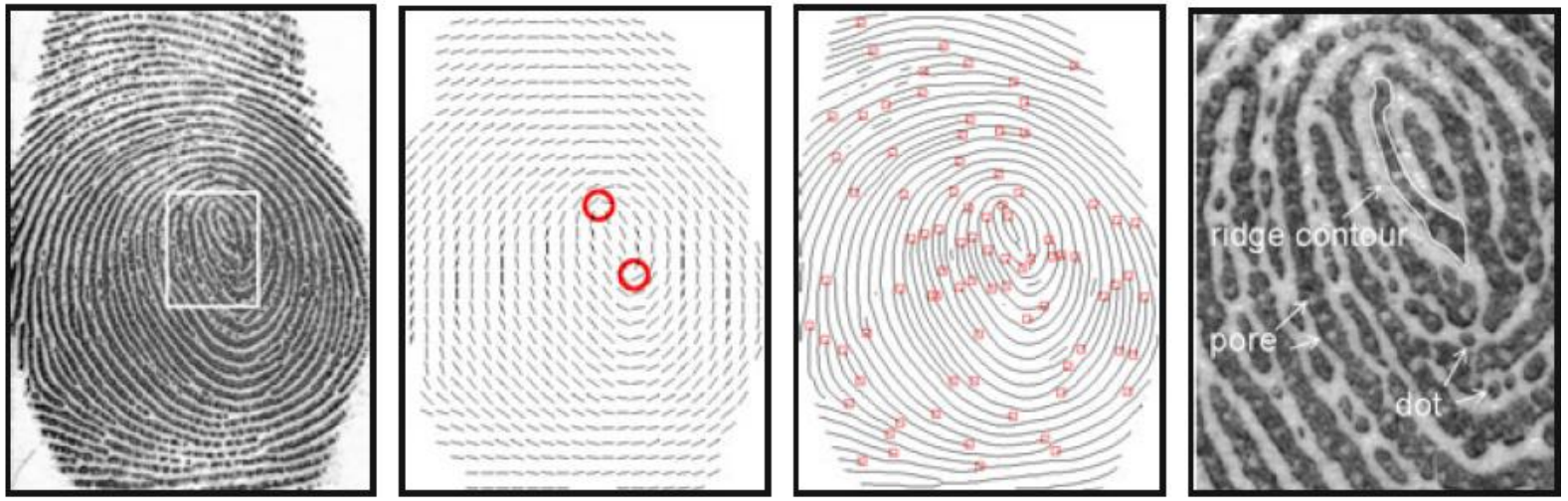


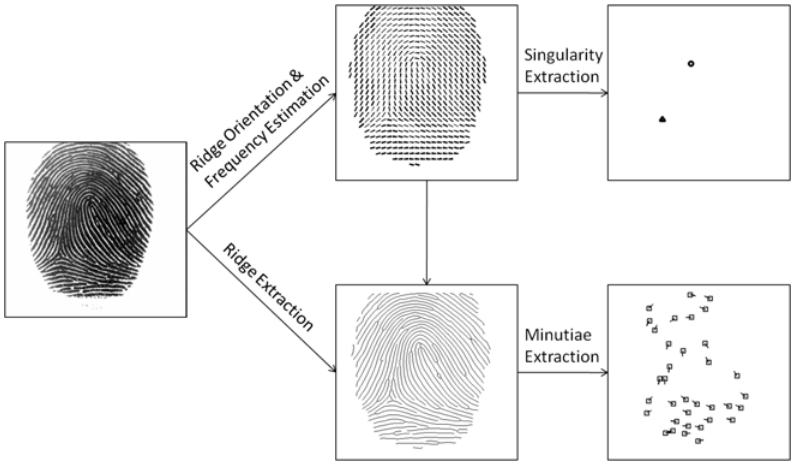
Fig. 2.5 Features at three different levels in a fingerprint. (a) Grayscale image (NIST SD30, A067_11), (b) Level 1 feature (orientation field or ridge flow and singular points), (c) Level 2 feature (ridge skeleton), and (d) Level 3 features (ridge contour, pore, and dot).

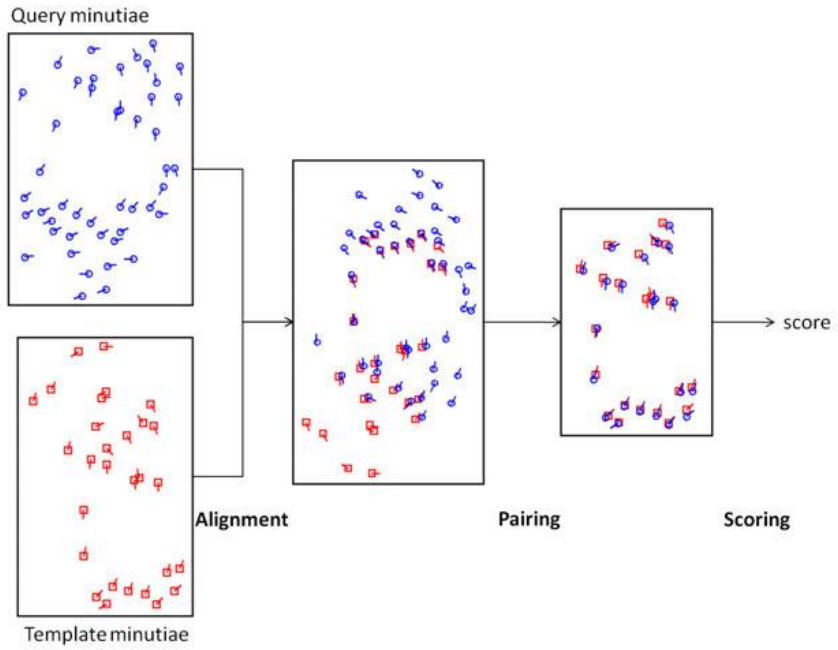
Fingerprints - characteristics

Pattern types
- arches
- loops
- whorls

Core and delta points

Minutiae points





Fingerprints -scanners

Optical scanner

Solid-state scanner
(capacitive sensors)

Ultrasound scanner

Fingerprints – scanners

- Good accuracy
- Used for both identification and verification
- Low cost
- Problem when skin is too dry or too wet
- Problem with dirt

Fingerprints - scanners

Touch (area) sensor

- Quickly becomes dirty
- Problem with latent prints
- Rotation problems
- Area vs cost

Sweep

- Reduced cost
- No dirt or latent prints
- Longer learning time
- Reconstruction of the image is time consuming

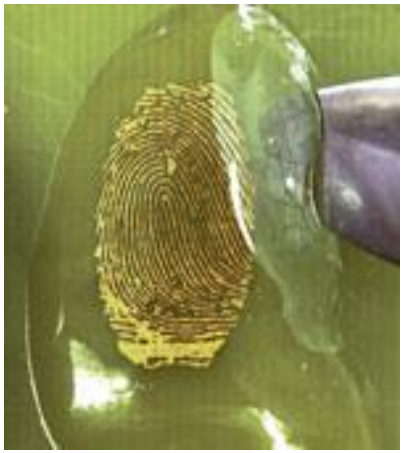
Fingerprints - attacks

Making a user cooperate using force or drugs

Using latent fingerprints

Artificial fingerprint

Gummy fingers



Making an Artificial Finger directly from a Live Finger

How to make a mold

Put the plastic into hot water to soften it.

Press a live finger against it.

The mold

It takes around 10 minutes.

Making an Artificial Finger directly from a Live Finger

Preparation of material

- A liquid in which immersed gelatin at 50 wt.% .



Add boiling water (30cc) to solid gelatin (30g) in a bottle and mix up them.

It takes around 20 minutes.

Making an Artificial Finger directly from a Live Finger

How to make a gummy finger

Pour the liquid
into the mold.

Put it into
a refrigerator to cool.

The gummy finger

It takes around 10 minutes.

Gummy fingers results

Real fingerprints	User 1	User 2	User 3
Reader 1	98%	100%	94%
Reader 2	100%	100%	100%
Reader 3	98%	34%	88%

Gummy fingerprint copies	User 1	User 2	User 3
Reader 1	98%	92%	100%
Reader 2	98%	100%	96%
Reader 3	92%	12%	82%

Fingerprint - liveness 1

Skin deformation

Pores

Perspiration

Fingerprint - liveness 2

Temperature

Optical properties

Pulse

Blood pressure

Electric resistance

Detection under epidermis

Example: Iris

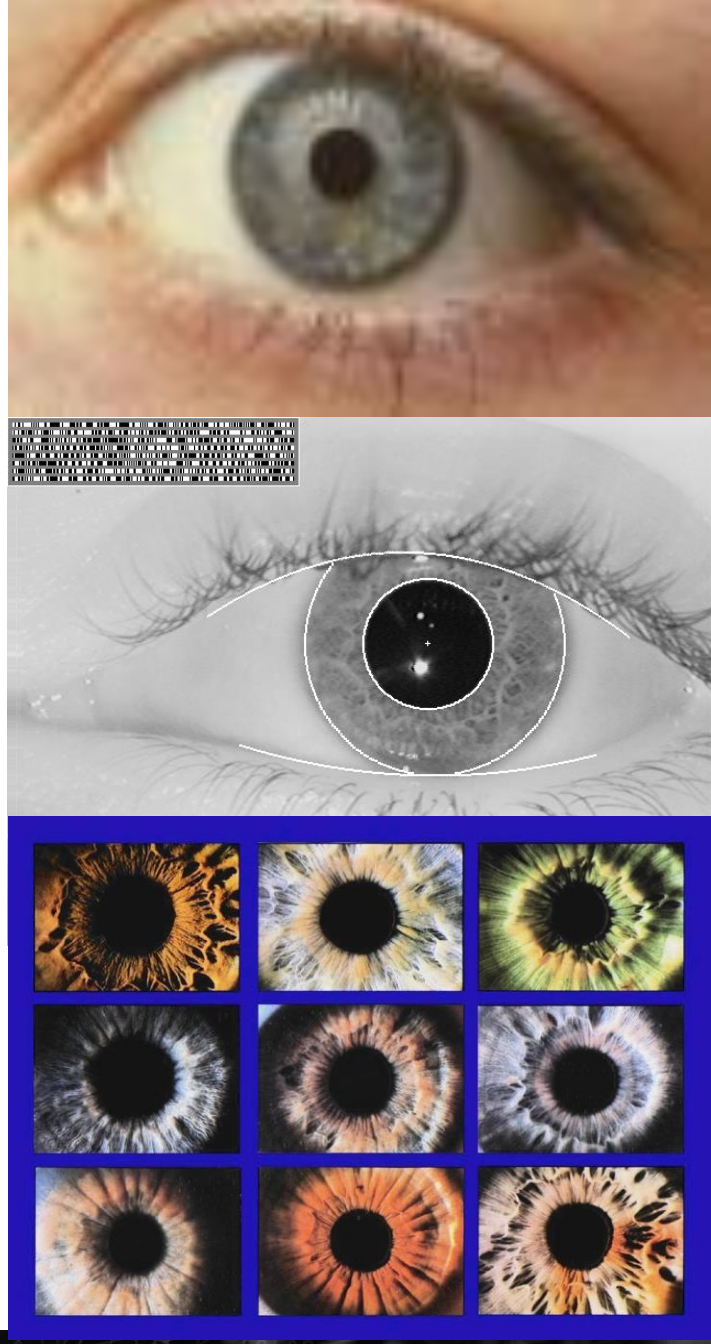
Can be captured from a distance

Monochrome camera with visible and near infra red light

Unique, two eyes and distinguish twins

Liveness detection

Experienced as intrusive



Disadvantages?

”Why the news on iris-recognition in cash machines started an ailien invasion”

Iris – or actually the rich texture from images of iris

The mesh consists of characteristics such as striations, rings, furrows, etc, giving the iris a unique pattern

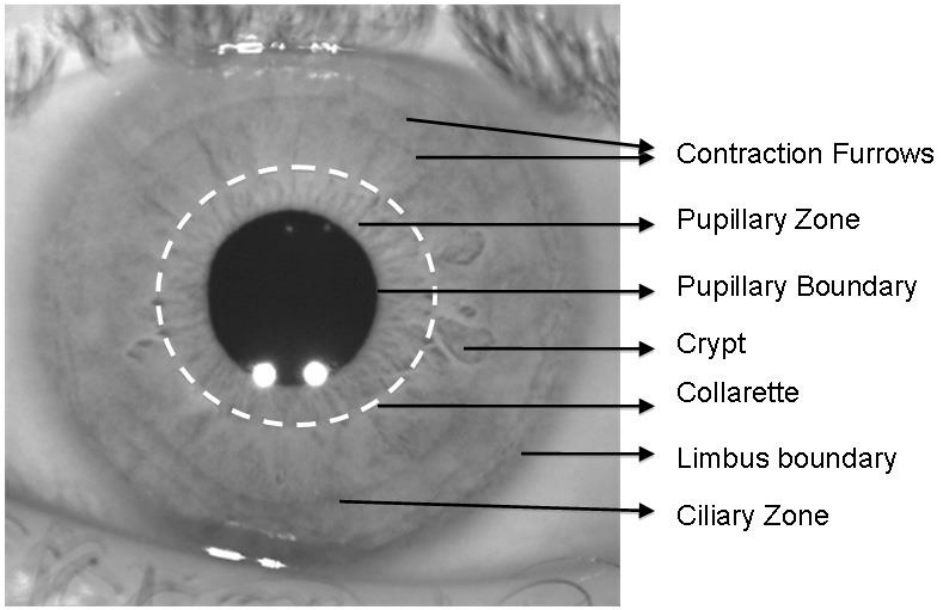
Don’t change with age

Can be captured from up to one meter

Ocular region of the human face

Iris

- Increased use since 1993
- Algorithm patent 1994
  - by Dr. John Daugman
  - used in all iris scanning systems today
- Works even with glasses and contact lenses
- Liveness is checked by using light to change the size of the pupil



NIR image

Iris

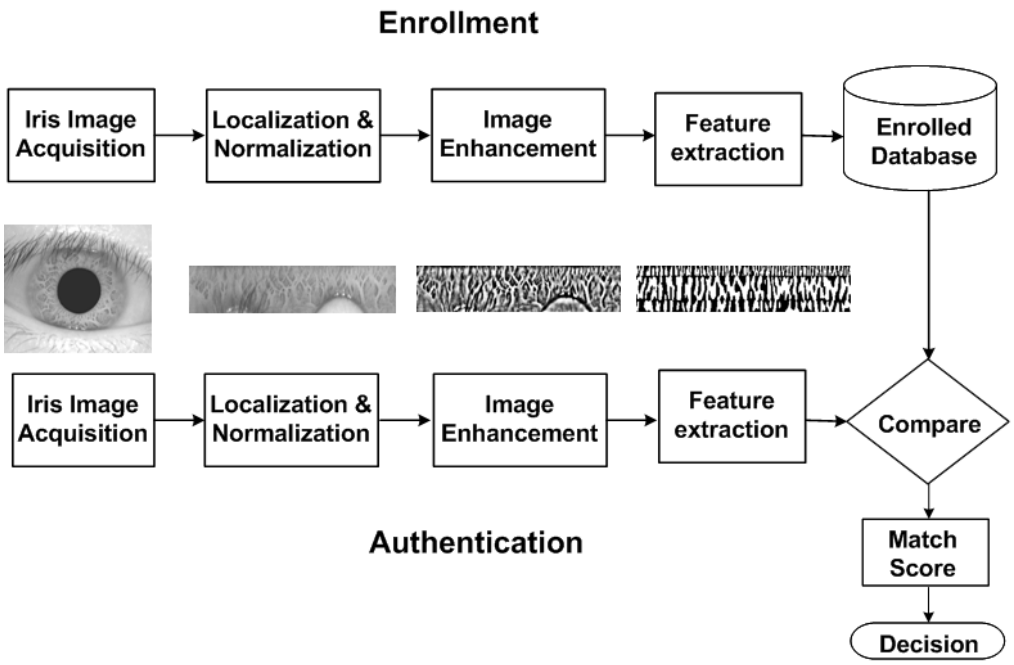
Very accurate, giving low FAR

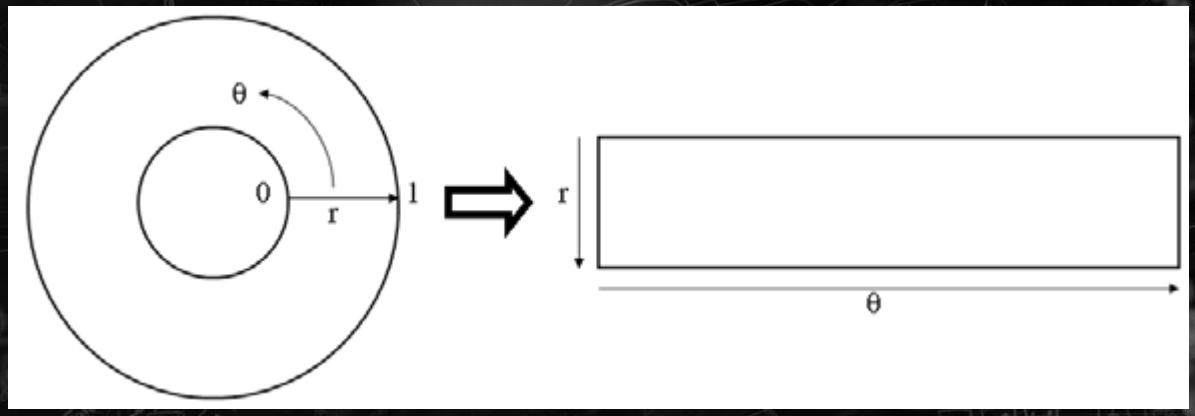
Used for identification and verification

High costs

May suffer from poor lighting and reflections

No human iris experts





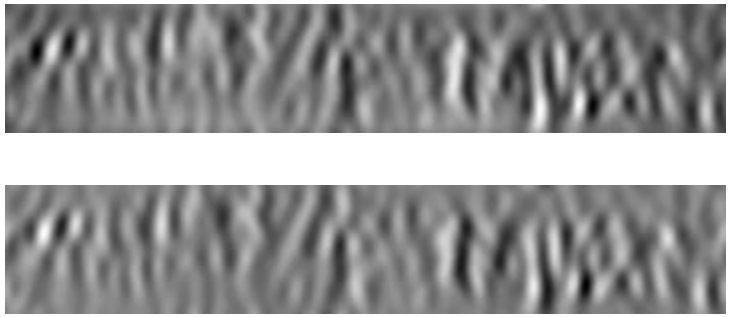
$I(x(r,\theta),y(r,\theta)) \rightarrow I(r,\theta)$

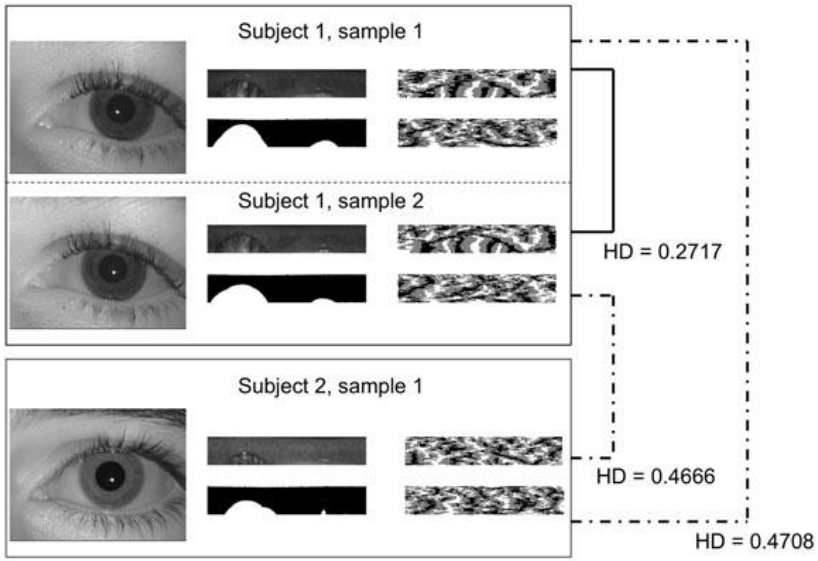
with

$x(r,\theta) = (1-r)xp(\theta)+rxl(\theta)$

and

$y(r,\theta) = (1-r)yp(\theta)+ryl(\theta)$





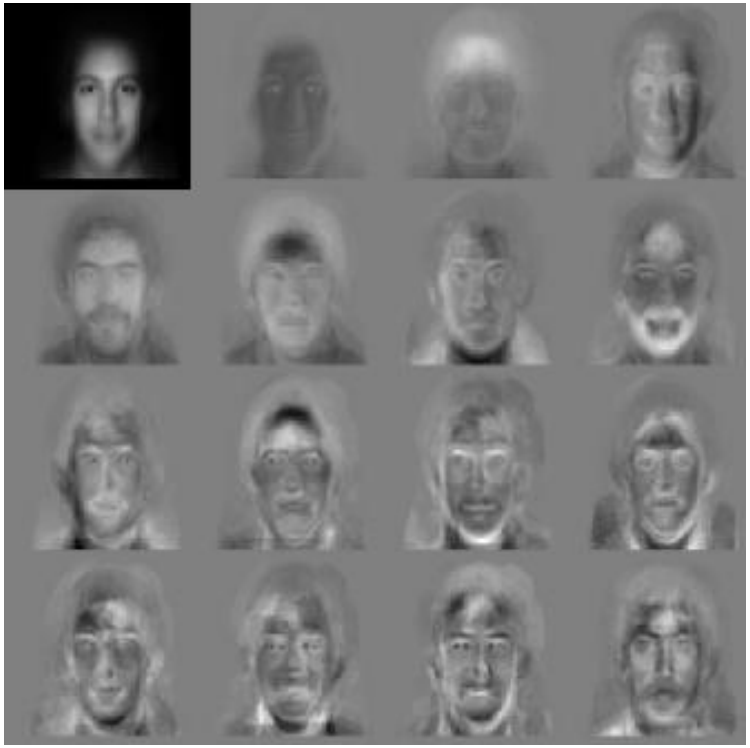
Iris - attacks

Contact lens with image

Porcelain eye

Photo of an eye

Example: Face



A face image can be acquired using a normal, off-the-shelf camera

Easy to accept by the public

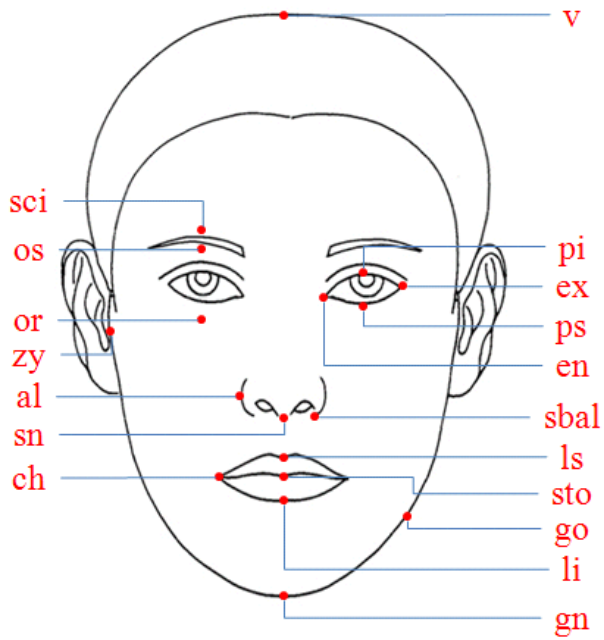
Cost is rather low

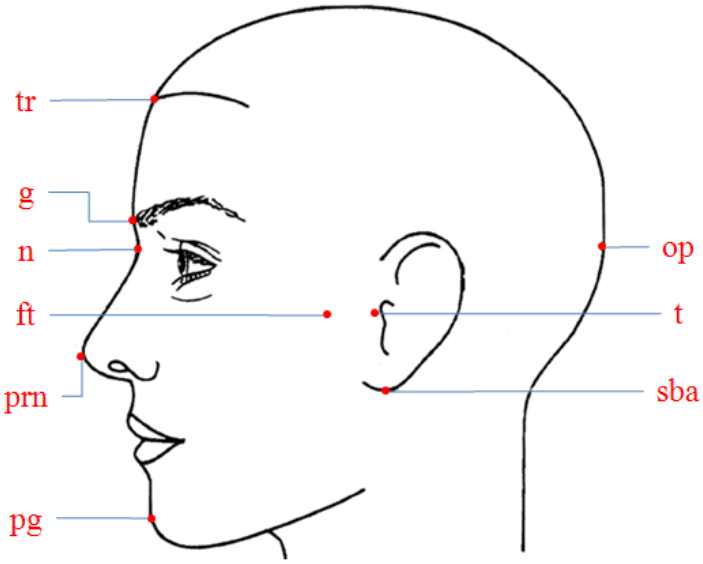
Huge problems with permanence and accuracy

Facial features

Gross facial characteristics, eg general geometry of the face and global skin

Localized face information eg structure of face components or their relations



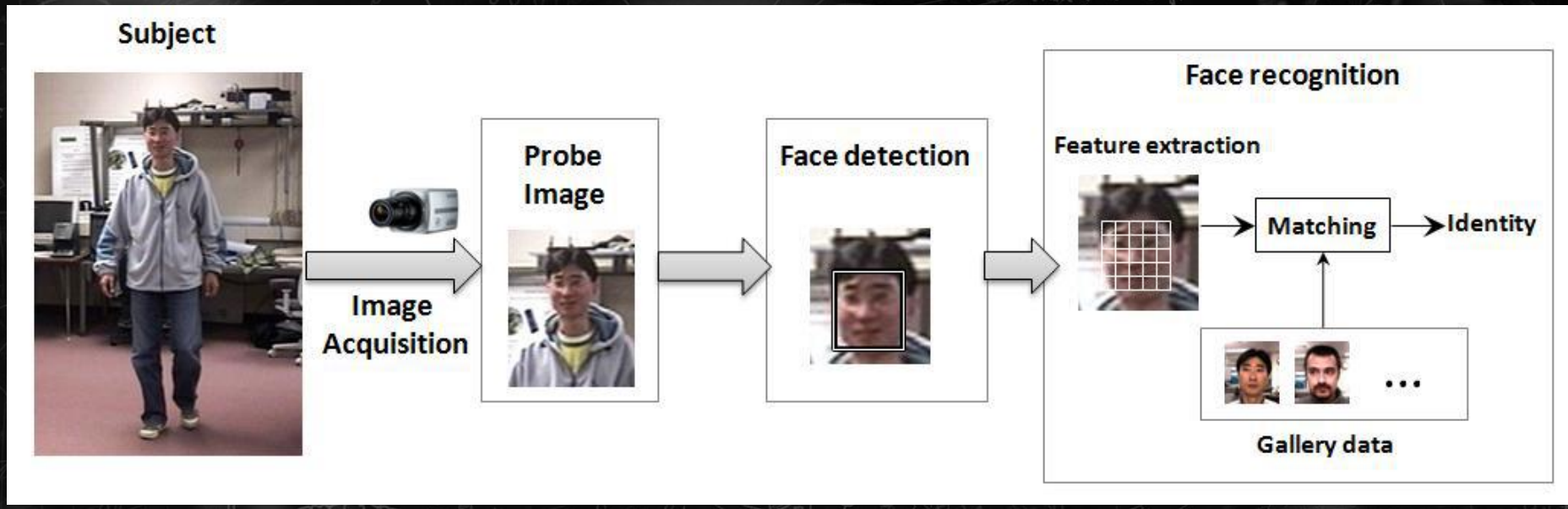


Face recognition algorithms

Global or feature-based approach

Feature-based
- standard points only
- not (too) sensitive to variation in position

Global
- process the entire face
- more accurate
- sensitive to variation in position and scale


Subject
Image Acquisition
Probe Image
Face detection
Face recognition
Feature extraction
Matching
Identity
Gallery data


VISIBLE
950 nm
1050 nm
1150 nm
1250 nm
1350 nm
1450 nm
1550 nm

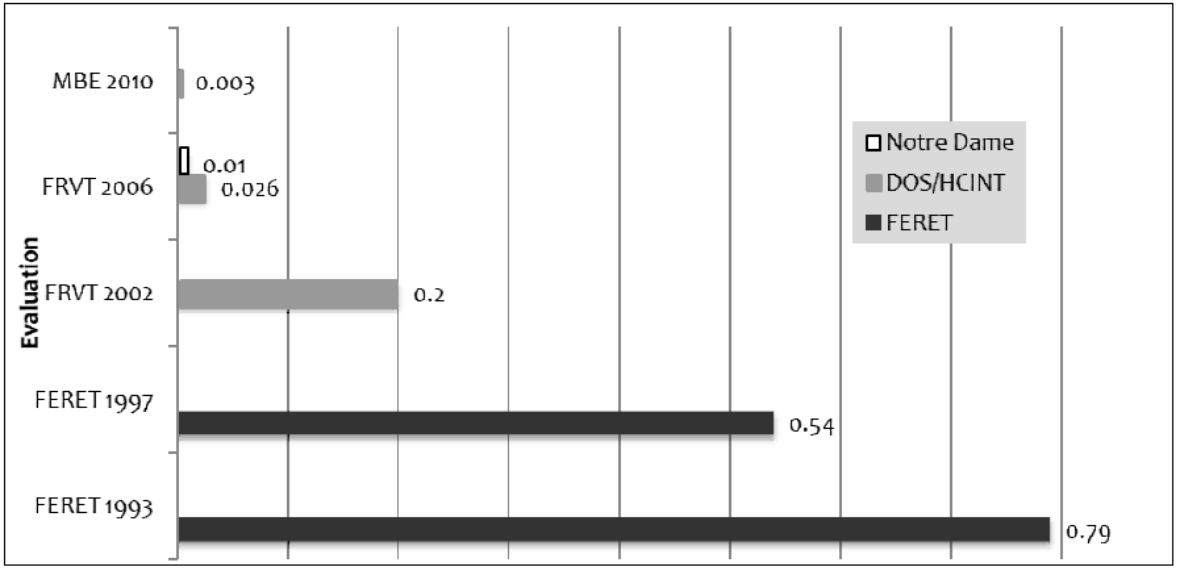
Face - attacks

Photo

Using low uniqueness

Masks or plastic surgery

False Reject Rate at a fixed False Accept Rate in the verification mode



Example: Hand geometry

Usually two views are taken, a top view and a side view.

The system is often bulky.

The hand geometry can change due to age and health conditions.

Example: Voice

Speaker recognition uses a microphone to record the voice.

Text dependent or text independent

Your voice can vary with age, illness and emotions.

Interesting with the increasing use of mobile phones.

Voice

Text dependent or text independent

Dependent
- The text is decided by the system
- Fixed or random
- Cooperation needed

Independent
- Any text can be used
- No cooperation needed
- Much harder

Voice - attacks

Recordings

Computer generated voice

”Tokens”?

”Token” is normally used for any authentication device with processing capacity

Smart cards are a variant

RFID devices (Radio-frequency identification) (ePassports have them!)

Phones with SIM-cards are another example

(Ross Anderson, Security Engineering chapter 16)

Attacking what?

Authentication tokens contain personal keys, which should not be easy to reveal

- Loss can be crucial to owner, if the attacker is another person, but usually further use can be blocked

Even more important are **system keys**!!!

- System keys may protect data proving payment for services
- System keys may enable fabrication of false tokens

Hardware attacks

Studying the equipment

- electro-magnetic signals
- power variations
- time to perform operations

Manipulating the equipment

- probing
- varying power
- inducing errors and stopping operations

Emission, examples

Electromagnetic emissions occur whenever you use an electronic device

Power consumption in the equipment can be measured

Sounds from keyboards can be recorded and analysed

Eavesdropping on tokens

Emissions from processing is usually too weak to intercept without going beyond the cover layer. See probing.

Power for smart cards can easily be eavesdropped at the reader

Power consumption can reveal what processing that goes on, including branches taken after testing internal data

Timing attacks

Speeding up calculations often includes dropping unnecessary steps

Typical example is not doing all the steps when a key bit is zero

Analysis of time to encrypt can directly reveal number of zero bits in key

Combined with power analysis, every key bit can be found

Defence against timing attacks

Do not optimise calculation times

- Multiply with zero and add to total sum
- Branch on values, but always do the same number of steps in both branches

If necessary (no division with zero etc.), insert dummy calculations

Defence against power analysis

Remove timing attacks first

Insert random steps

Defence against eavesdropping

Use sufficient shielding around processors

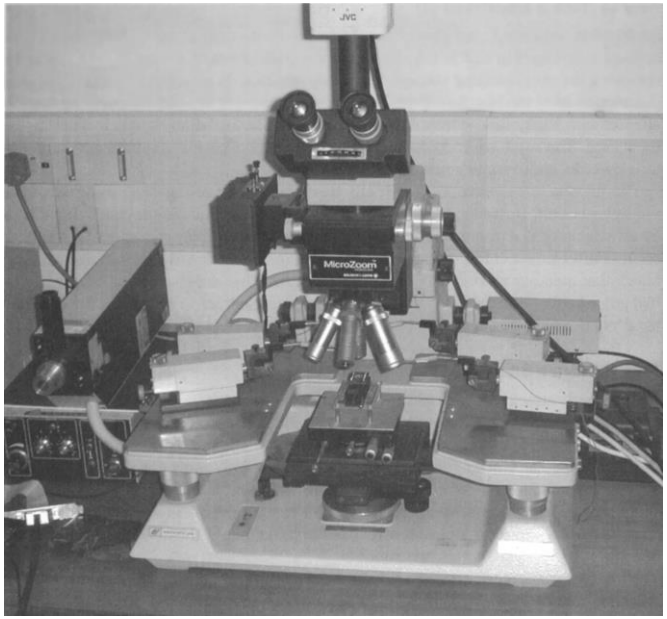
Avoid sending sensitive data, like keys, on internal buses

Probing

Direct contact with the electronics makes direct reading possible

See the literature (Anderson) for details

Also consider remanence! (It can make defences like power removal and erasures futile.)



Defence against probing

Use sufficient shielding around processors

Hardened and shatter-prone epoxy with meshes etc. makes removal of coatings much more difficult and expensive

Avoid sending sensitive data, like keys, on internal buses

Consider internal encryption

Remove power and erase sensitive data, when an attack is detected

Power manipulation

Preventing check data from being written may disable protective checks

Introduction of errors in the processing flow may alter the actual instruction sequence in ways that reveal sensitive data

- Checks can be skipped
- Limits for what can be output may be cancelled

Defence against power manipulation

When writing check data, always check that it is indeed written before proceeding with the calculations

Hide which step the processor executes in the processing flow (see power analysis)

Inducing errors

Carefully designed erroneous inputs can trigger unwanted events

- Similar to using security holes and badly designed protocols in general

Errors can be injected in stored data via particle beams, light on partly revealed surfaces etc.

- manipulate instruction flow
- change control limits
- alter key bits in ways that make analysis possible

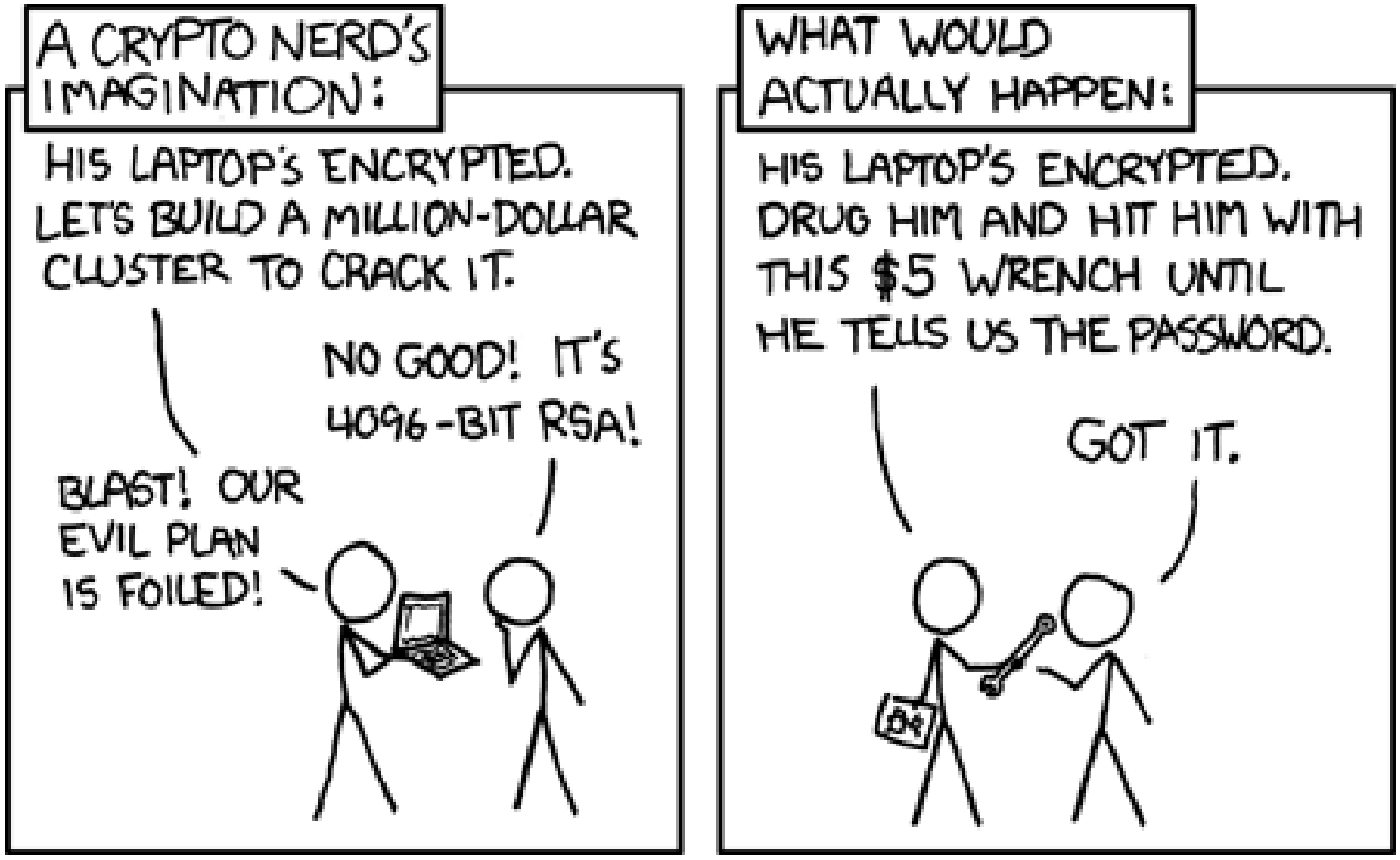
Defence against induced errors

Use error detection for stored values, and check before use

Check outputs for consistency, if possible

Check inputs and block everything except meaningful, correctly designed sets

Questions?


A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.

Linköping University

expanding reality

www.liu.se