INF3510
Information Security
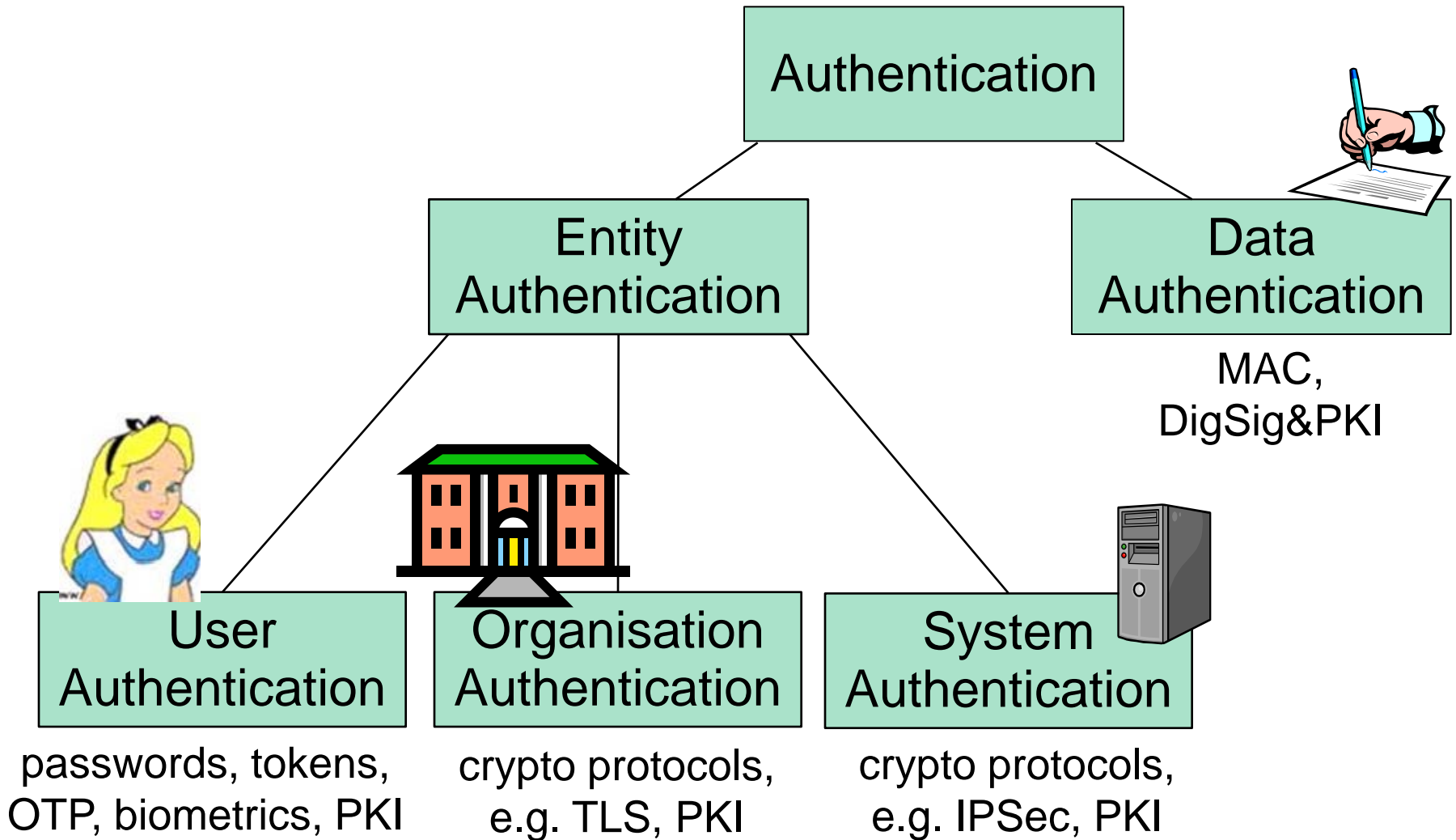
Lecture 8: User Authentication

University of Oslo

Spring 2017

Outline

- Concepts related to authentication
  - Identity and authentication steps
- User Authentication
  - Knowledge-Based Authentication
    - Passwords
  - Ownership-Based Authentication
    - Tokens
  - Inherence-Based Authentication
    - Biometrics
- Authentication frameworks for e-Government

Taxonomy of Authentication

Authentication
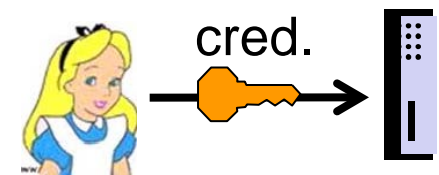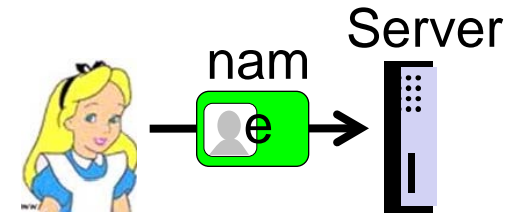
Entity Authentication

Data Authentication

MAC, DigSig&PKI

User Authentication

Organisation Authentication

System Authentication

passwords, tokens, OTP, biometrics, PKI

crypto protocols, e.g. TLS, PKI

crypto protocols, e.g. IPSec, PKI

Steps of User Authentication

Config-uration phase

1. **Registration**
   – User visits ID-provider, with pre-authentication credents.

2. **Provisioning**
   – ID-provider registers unique name and issues credential
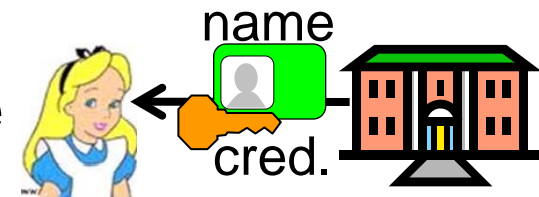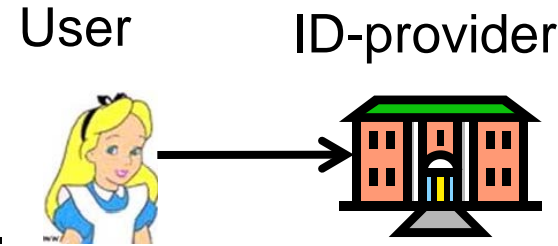
Operation phase

3. **Self-identification**
   – User presents the unique name to claim an identity

4. **Verification of identity**
   – Proof of identity based on credentials

User authentication credentials

- A credential is the 'thing' used for authentication.
  - May also be referred to as a "token" or "authenticator"
  - e.g. reusable passwords, PIN, biometrics, smart cards, certificates, cryptographic keys, OTP hardware tokens.
- Credential categories:
  1. Knowledge-Based (Something you know): Passwords
  2. Ownership-Based (Something you have): Tokens
  3. Inherence-Based (Something you are/do): Biometrics
     - physiological biometric characteristics
     - behavioural biometric characteristics
- Combinations, called multi-factor authentication

Knowledge-Based Authentication

Something you know: Passwords
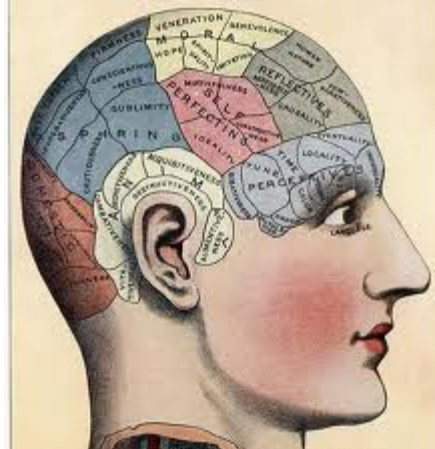
Authentication: Static passwords

`123456`

- Passwords are a simple and most-often-used authenticator.
  - Something the user knows
- Problems:
  - Easy to share (intentionally or not)
  - Easy to forget
  - Often easy to guess (weak passwords)
  - Can be written down (both god and bad)
    - If written down, then "what you know" is "where to find it"
  - Often remains in computer memory and cache

RockYou Hack

- 32 million cleartext passwords stolen from RockYou database in 2009
- Posted on the Internet
- Contains accounts and passwords for websites
    - MySpace, Yahoo, Hotmail
- Analyzed by Imperva.com
    - 1% use 123456
    - 20% use password from set of 5000 different passwords

MOST POPULAR PASSWORDS

Nearly one million RockYou users chose these passwords to protect their accounts.

1.	123456	17.	michael
2.	12345	18.	ashley
3.	123456789	19.	654321
4.	password	20.	qwerty
5.	iloveyou	21.	iloveu
6.	princess	22.	michelle
7.	rockyou	23.	111111
8.	1234567	24.	0
9.	12345678	25.	tigger
10.	abc123	26.	password1
11.	nicole	27.	sunshine
12.	daniel	28.	chocolate
13.	babygirl	29.	anthony
14.	monkey	30.	angel
15.	jessica	31.	FRIENDS
16.	lovely	32.	soccer

Source: Imperva

Secure password strategies

- Passwords length $\geq$ 13 characters
- Use $\geq$ 3 categories of characters
  - L-case, U-case, numbers, special characters
- Do not use ordinary words (names, dictionary wds.)
- Change typically every 3 − 13 months
- OK to reuse between low-sensitivity accounts
- Do not reuse between high-sensitivity accounts
- Store passwords securely
  - In brain memory
  - On paper, adequately protected
  - In cleartext on offline digital device, adequately protected
  - Encrypted on online digital device

Strategies for strong passwords

- ## User education and policies
    - Not necessarily with strict enforcement
- ## Proactive password checking
    - User selects a potential password which is tested
    - Weak passwords are not accepted
- ## Reactive password checking
    - SysAdmin periodically runs password cracking tool (also used by attackers) to detect weak passwords that must be replaced.
- ## Computer-generated passwords
    - Random passwords are strong but difficult to remember
    - FIPS PUB 181 http://www.itl.nist.gov/fipspubs/fip181.htm specifies automated pronounceable password generator
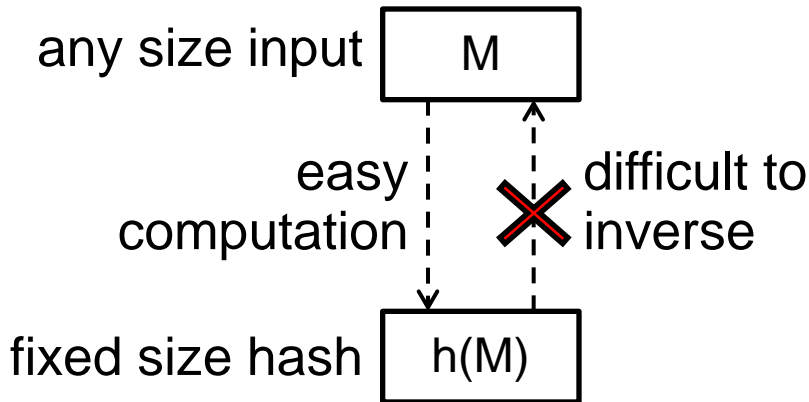
Password storage in OS

- /etc/shadow is the file where modern Linux/Unix stores it passwords
  - Earlier version stored it in /etc/passwd
  - Need root access to modify it
- \windows\system32\config\sam it the file Windows systems normally stores it passwords
  - Undocumented binary format
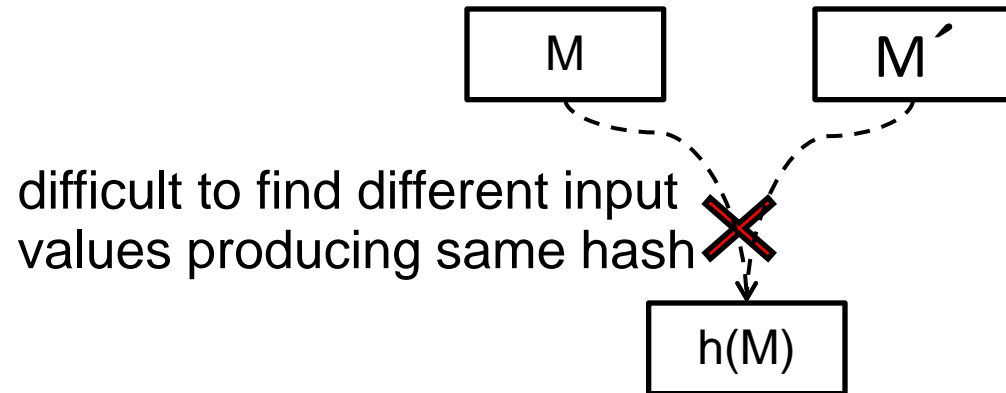
Prevent exposure of password file

- The computer verifies user passwords against stored values in the password file
- Password file must be available to OS
  - This file need protection from users and applications
  - Avoid offline dictionary attacks
- Protection measures
  - Access control (only accessible by OS kernel)
  - Hashing or Encryption
- In case a password file gets stolen, then hashing/encryption can provide protection.

Hash functions

One-way function

Collision free

any size input [M]

easy computation ⋮ ✗ difficult to inverse

fixed size hash [h(M)]

[M] [M′]

difficult to find different input values producing same hash ✗

[h(M)]

- *A hash function is easy to compute but hard to invert.*

- Passwords can be stored as hash values.
- Authentication function first computes hash of received password, then compares against stored hash value

Cracking passwords

- Bruce Force
  - Trying all possible combinations
- Intelligent search
  - User name
  - Name of friends/relatives
  - Phone number
  - Birth dates
  - Dictionary attack
    - Try all words from an dictionary
    - Precomputed hashes: Rainbow tables

Hash table and rainbow table attacks

- Attackers can compute and store hash values for all possible passwords up to a certain length
- A list of password hashes is a **hash table**
- A compressed hash table is a **rainbow table**
- Comparing and finding matches between hashed passwords and hash/rainbow table is used to determine cleartext passwords.

Password salting: Defence against password cracking
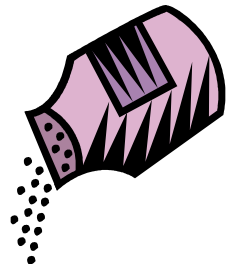
- Prepend or append random data (salt) to a user's password before hashing
    - In Unix: a randomly chosen integer from 0 to 4095.
    - Different salt for each user
    - Produces different hashes for equal passwords
    - Prevents that users with identical passwords get the same password hash value
    - Increases the amount of work required for hash table attacks and rainbow table attacks

Methods of storing passwords on server

- Cleartext password (low security)
  - Password: 123456,
  - Stored on server: 123456
- Hashed password (moderately security)
  - Password: 123456
  - Stored on server: e.g. SHA1-hash of password: 7c4a8d09ca3762af61e59520943dc26494f8941b
- Salted hashed password (good security)
  - Password: 123456
  - Stored on server: Salt + Salted hash
    e.g. "salt": f8b97abc30b72e54
    eg. SHA1-hash of password + salt
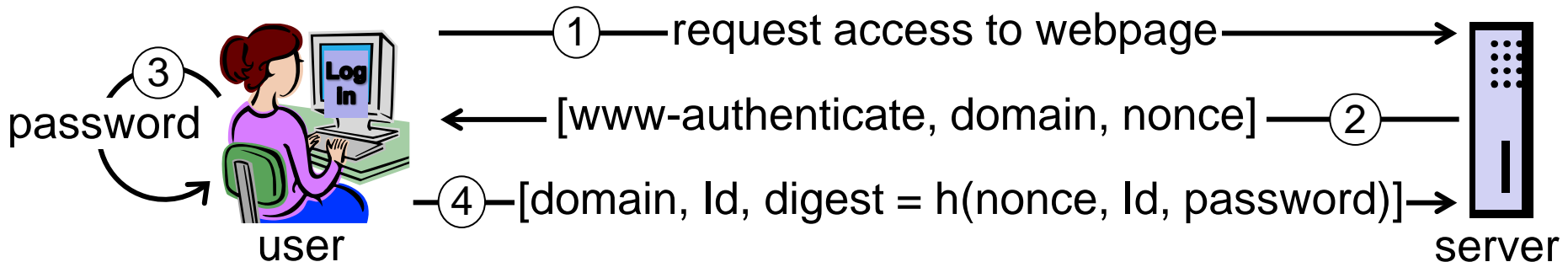   1736f11fae29189749a8a54f45e25fb693c3959d

Problems with using passwords in the clear

- A password sent "in clear" can be captured during transmission, so an attacker may reuse it.
- An attacker setting up a fake server can get the password from the user
  - E.g. phishing attack.
- Solutions to these problems include:
  - Encrypted communication channel
  - One-time passwords (token-based authentication)
  - Challenge-response protocols

HTTP Digest Authentication
A simple challenge-response protocol (rarely used)

- A simple challenge response protocol specified in RFC 2069
- Server sends:
  - WWW-Authenticate = Digest
  - realm="service domain"
  - nonce="some random number"
- User types Id and password in browser window
- Browser produces a password digest from nonce, Id and password using a 1-way hash function
- Browser sends Id and digest to server that validates digest

① request access to webpage →

← [www-authenticate, domain, nonce] — ②

③ password

④ [domain, Id, digest = h(nonce, Id, password)] →

user

server

Ownership-Based Authentication

Something you have: Tokens

Taxonomy of Authentication Tokens

Authentication Tokens

Synchronised Tokens

Challenge-Response Tokens

Clock-Based Tokens

Counter-Based Tokens
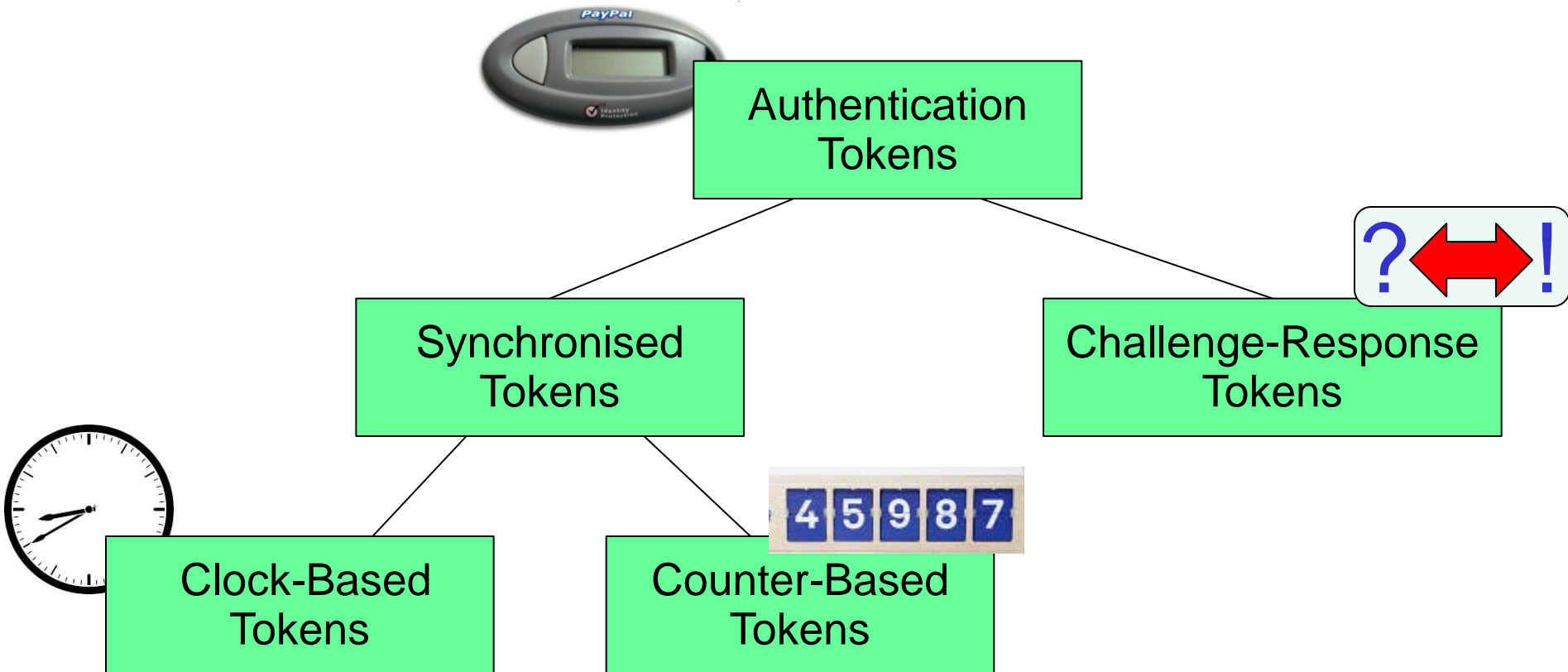
Synchronised OTP (One-Time-Password) Generator

- Using a password only once significantly strengthens the strength of user authentication.
- Synchronized password generators produce the same sequence of random passwords both in the token and at the host system.
  - OTP is 'something you have' because generated by token
- There are two general methods:
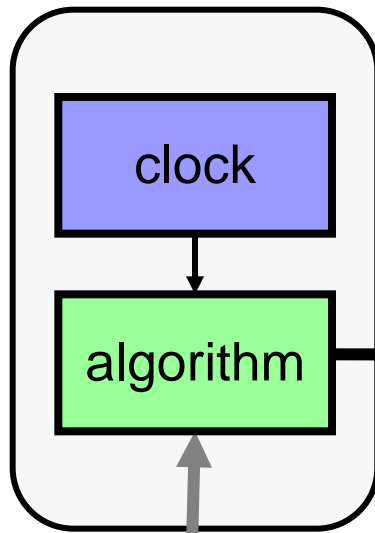 - Clock-based tokens
  - Counter-based tokens
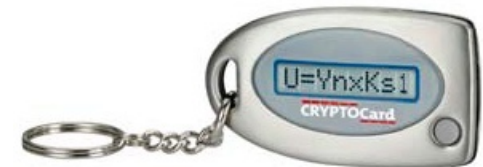
Clock-based OTP Tokens: Operation

- Token displays time-dependent code on display
  - User copies code from token to terminal to log in
- Possession of the token is necessary to know the correct value for the current time
- Each code computed for specific time window
- Codes from adjacent time windows are accepted
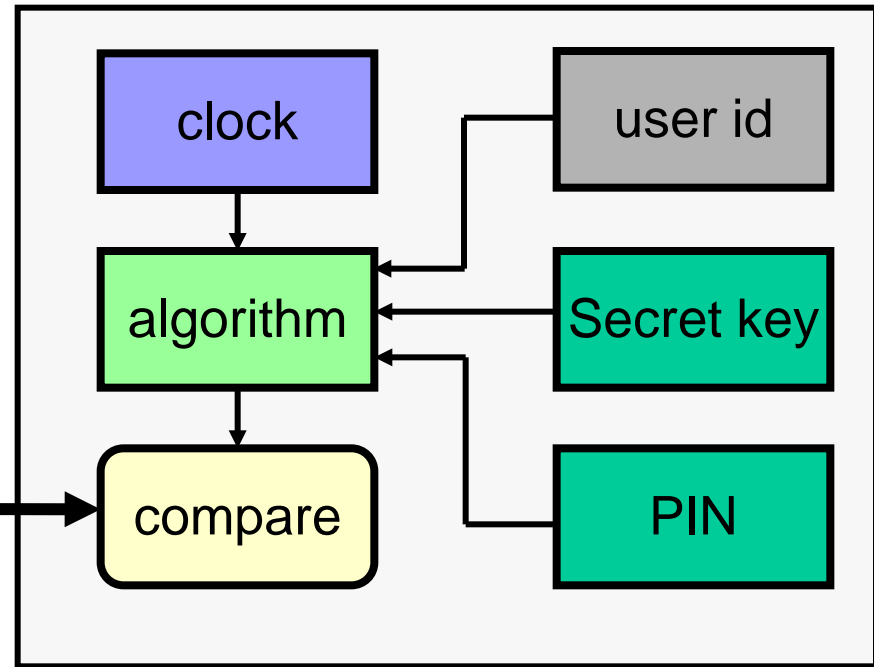- Clocks must be synchronised
- Example: BankID and SecurID

Clock-based OTP Token Operation with (optional) input arguments

USER'S
TOKEN

HOST

clock

algorithm

User Id + PIN / Secret key

clock

algorithm

compare

user id

Secret key

PIN

Clock-based OTP Tokens:

SafeID OTP
token with PIN

ActiveID OTP token
with PIN

BankID OTP token
with PIN

Feitan OTP
token witout PIN

RSA SecurID
without PIN

BankID OTP token
without PIN

Hacking OTP Tokens

- RSA was hacked in 2007.

- Secret key for OTP tokens stolen

- Hackers could generate OTP and spoof users

- Companies using RSA SecureID were vulnerable

- Lockheed Martin used RSA SecureID

- Chinese attackers spoofed Lockheed Martin staff
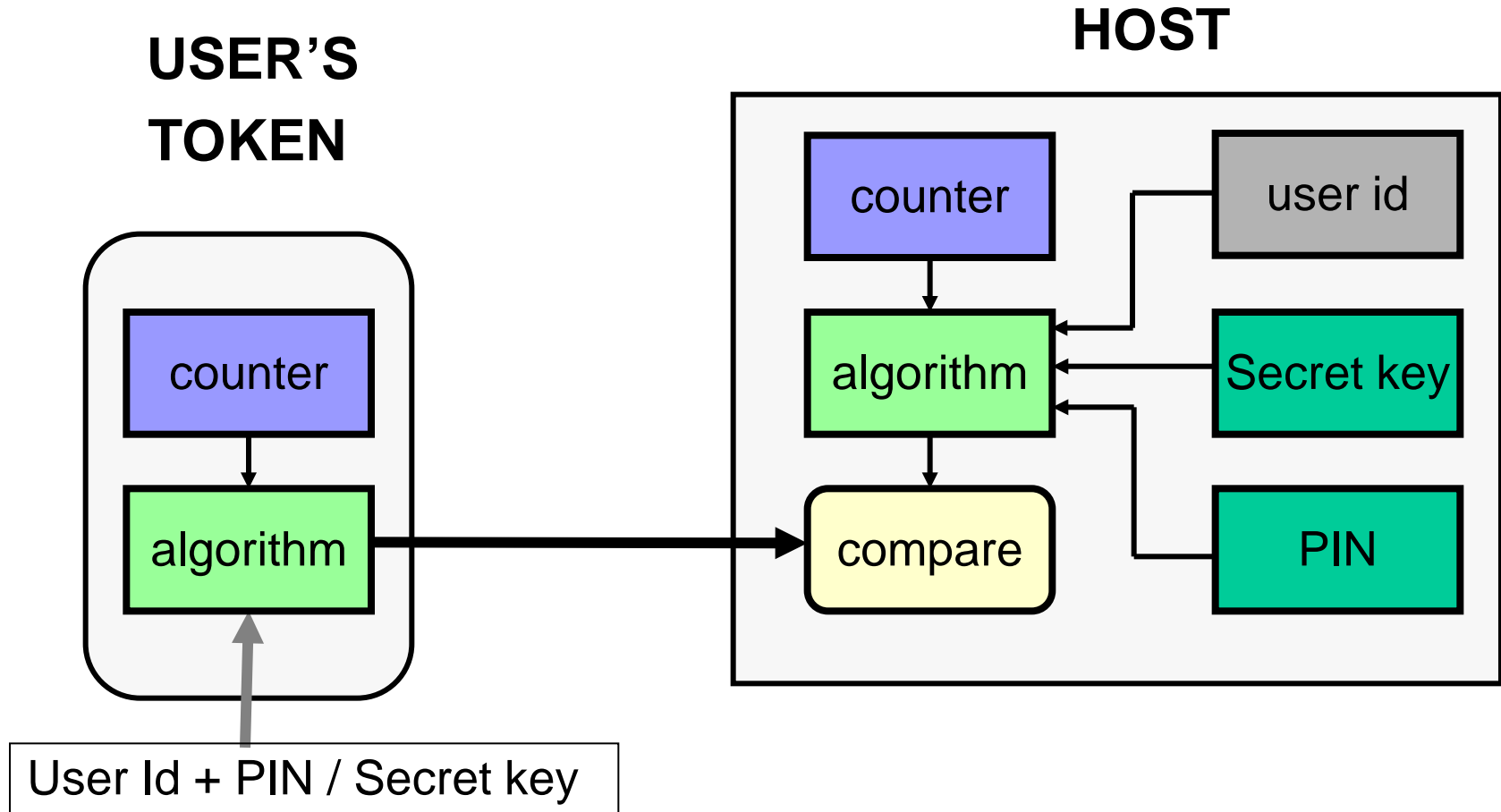 - Stole plans for F-35 fighter jet

China's J-31

U.S F-35

Counter-based OTP Tokens: Overview

- Counter-based tokens generate a 'password' result value as a function of an internal counter and other internal data, without external inputs.

- HOTP is a HMAC-Based One-Time Password Algorithm described in RFC 4226 (Dec 2005)
http://www.rfc-archive.org/getrfc.php?rfc=4226

  – Tokens that do not support any numeric input

  – The value displayed on the token is designed to be easily read and entered by the user.

Reflex 530

Counter-based OTP Token Operation with (optional) input arguments

USER'S TOKEN

HOST

- counter
- algorithm

- counter
- user id
- algorithm
- Secret key
- compare
- PIN

User Id + PIN / Secret key

Challenge Response Based Tokens for User Authentication:

- A challenge is sent in response to access request
  - A legitimate user can respond to the challenge by performing a task which requires use of information only available to the user (and possibly the host)
- User sends the response to the host
  - Access is approved if response is as expected by host.
- Advantage: Since the challenge will be different each time, the response will be too – the dialogue can not be captured and used at a later time
- Could use symmetric or asymmetric crypto

Token-based User authentication
Challenge Response Systems

HOST

TOKEN

key

Id/PIN/key

PIN → encrypt

challenge

generator → encrypt

display → compare

response
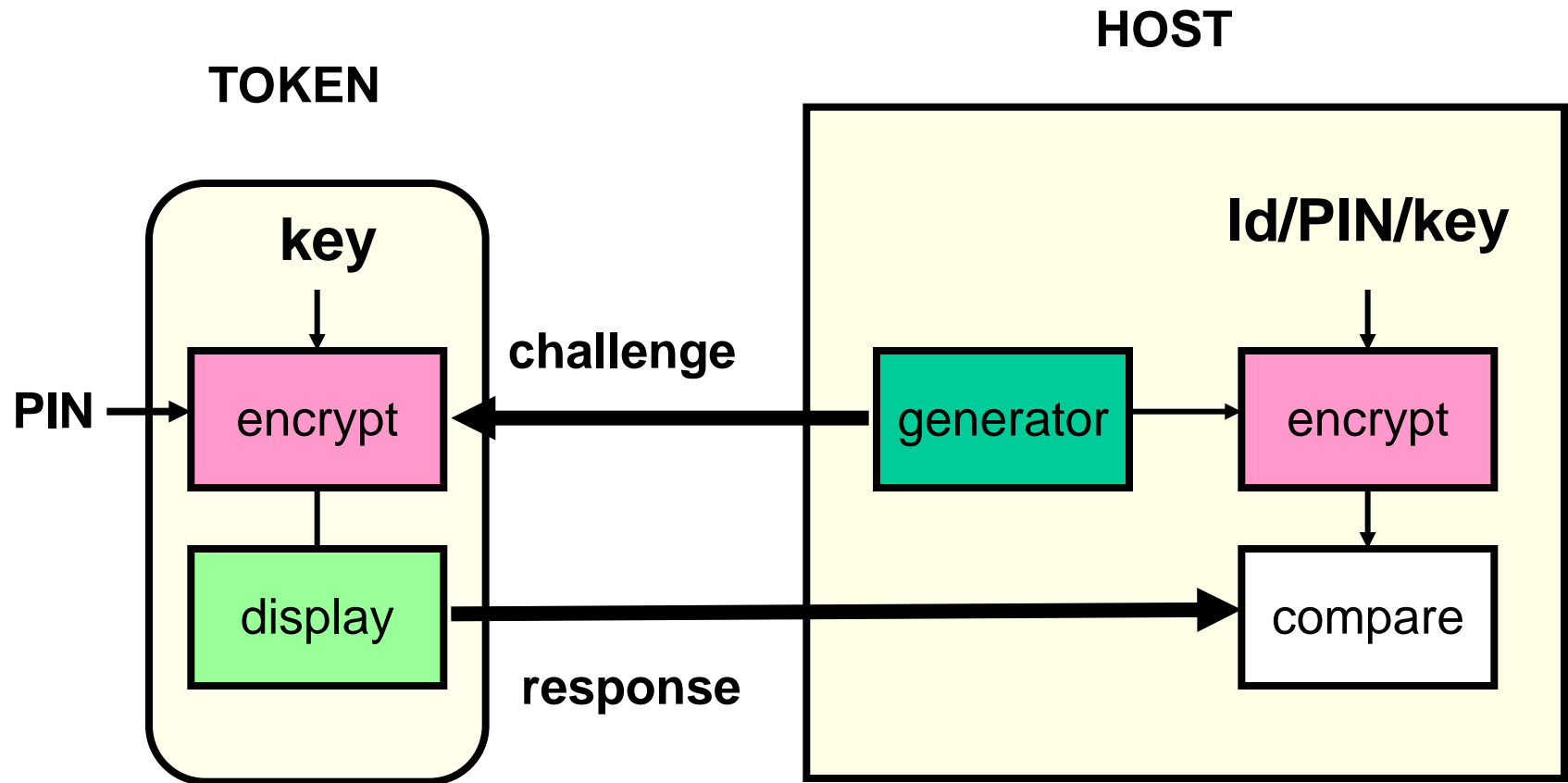
Symmetric algorithm case

Contactless Cards: Overview

- Contactless cards, also called RFID (Radio Frequency Id) cards, consists of a chip and an antenna.
  - No need to be in physical contact with the reader.
  - Uses radio signals to communicate
  - Powered by magnetic field from reader
  - When not within the range of a reader it is not powered and remains inactive.
  - Battery powered RFID tags also exist

- Suitable for use in hot, dirty, damp, cold, foggy environments

RFID Card

KAISERE

Computer chip

Antenna
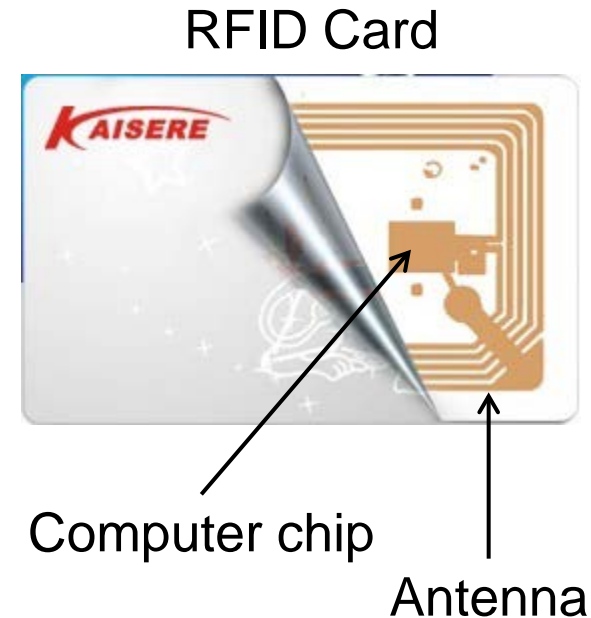
Inherence-Based Authentication

Biometrics

Something you are

Something you do

Biometrics: Overview

- ## What is it?
  - Automated methods of verifying or recognizing a person based upon a physiological characteristics.

- ## Biometric modalities, examples:
  - fingerprint
  - facial recognition
  - eye retina/iris scanning
  - hand geometry
  - written signature
  - voice print
  - keystroke dynamics

Biometrics: Requirements

- **Universality**:
Each person should have the characteristic;

- **Distinctiveness**:
Any two persons should be sufficiently different in terms of the characteristic;

- **Permanence**:
The characteristic should be sufficiently invariant (with respect to the matching criterion) over a period of time;

- **Collectability**:
The characteristic should be measurable quantitatively.

Biometrics: Practical considerations

- ## Accuracy:
  - The correctness of a biometric system, expressed as ERR (Equal Error Rate), where a low ERR is desirable.

- ## Performance:
  - the achievable speed of analysis,
  - the resources required to achieve the desired speed,

- ## Acceptability:
  - the extent to which people are willing to accept the use of a particular biometric identifier (characteristic)

- ## Circumvention resistance:
  - The difficulty of fooling the biometric system

- ## Safety:
  - Whether the biometric system is safe to use

Biometrics Safety

- Biometric authentication can be safety risk
  - Attackers might want to "steal" body parts
  - Subjects can be put under duress to produce biometric authenticator
- Necessary to consider the physical environment where biometric authentication takes place.

Car thieves chopped off part of the driver's left index finger to start S-Class Mercedes Benz equipped with fingerprint key. Malaysia, March 2005
(NST picture by Mohd Said Samad)

Biometrics: Modes of operation

- Enrolment:
  - analog capture of the user's biometric attribute.
  - processing of this captured data to develop a template of the user's attribute which is stored for later use.

- Identification (1:N, one-to-many)
  - capture of a new biometric sample.
  - search the database of stored templates for a match based solely on the biometric.

- Verification of claimed identity (1:1, one-to-one):
  - capture of a new biometric sample.
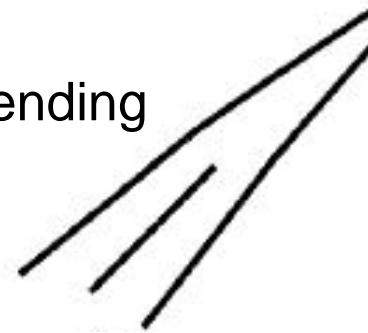 - comparison of the new sample with that of the user's stored template.

Extracting biometric features
Example fingerprints: Extracting minutia

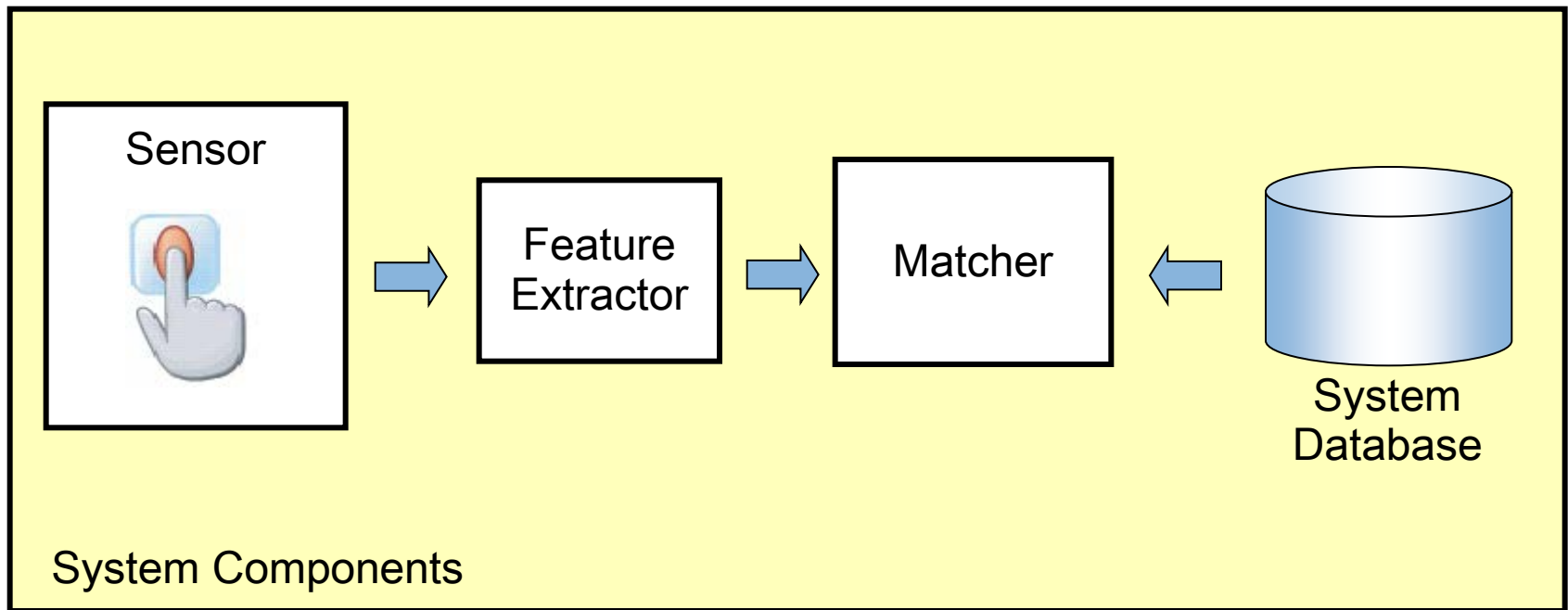Bifurcation
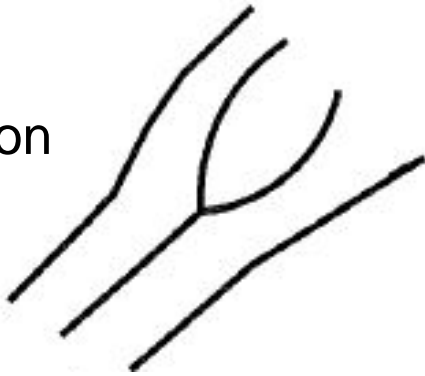
Ridge ending

Biometric

Minutia Points
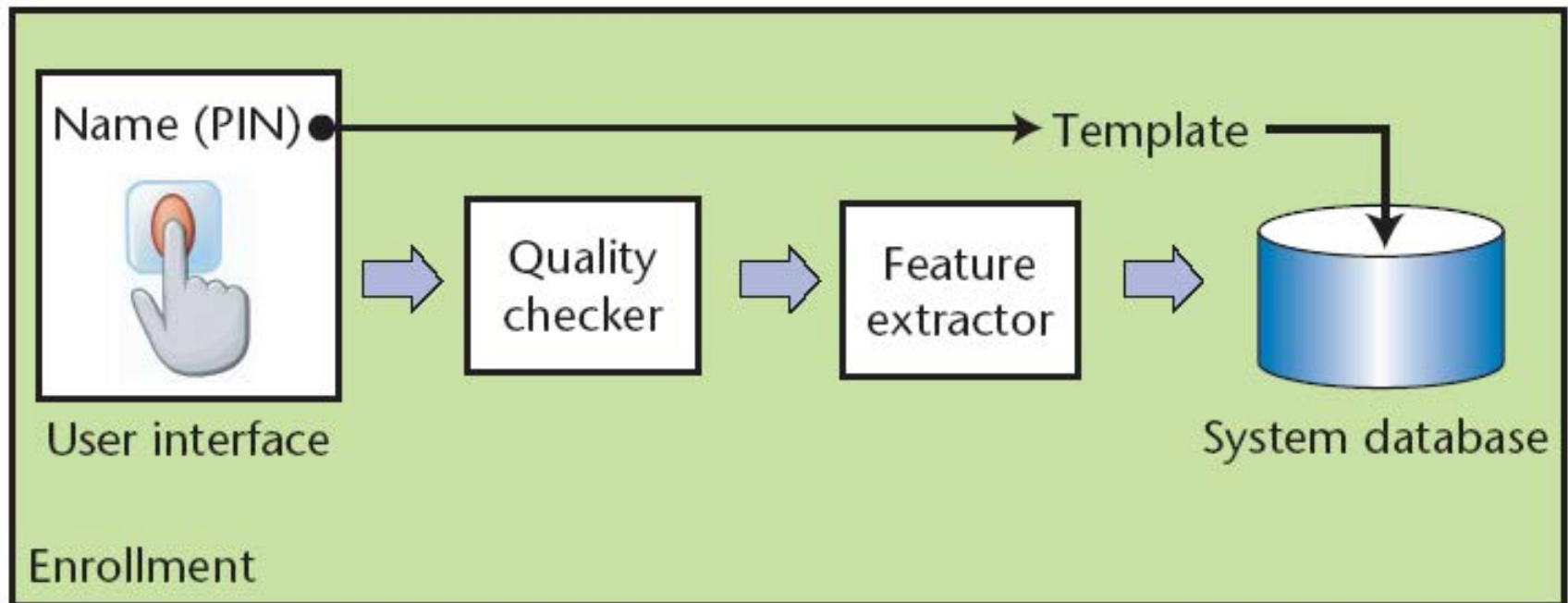
Minutia Map

Data Stream

```
0101010001101000011010
0101110011001000000110
1001011100110010000001
1011100110111101110100
0010000001100001011000
1101110100011101010110
0001011011000110110001
1110010010000001100110
0110100101101110011001
1101100101011100100111
0000011100100110100101
1011100111010000100000
0110010001100001011101
0001100001001011000010
```

Biometrics:
System components

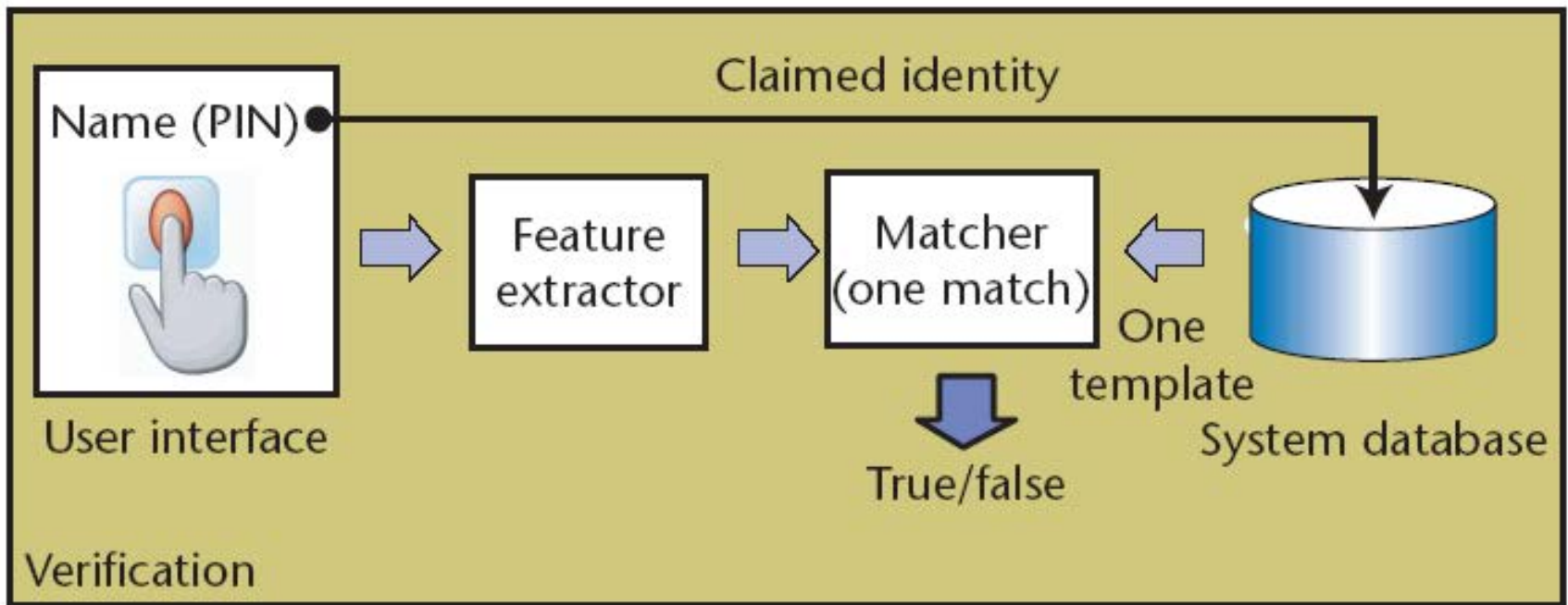Sensor → Feature Extractor → Matcher ← System Database

System Components

Biometrics: Enrolment

Name (PIN) → Template
User interface → Quality checker → Feature extractor → System database

Enrollment

Biometric Recognition: Security and Privacy Concerns

Biometrics: Verification

Claimed identity

Name (PIN) → Feature extractor → Matcher (one match) → True/false

One template

User interface

System database

Verification

Biometric Recognition: Security and Privacy Concerns

Biometrics: Identification

Feature extractor → Matcher (*N* matches) ← System database

User interface

Identification

N templates

User's identity or "user unidentified"

Biometric Recognition: Security and Privacy Concerns
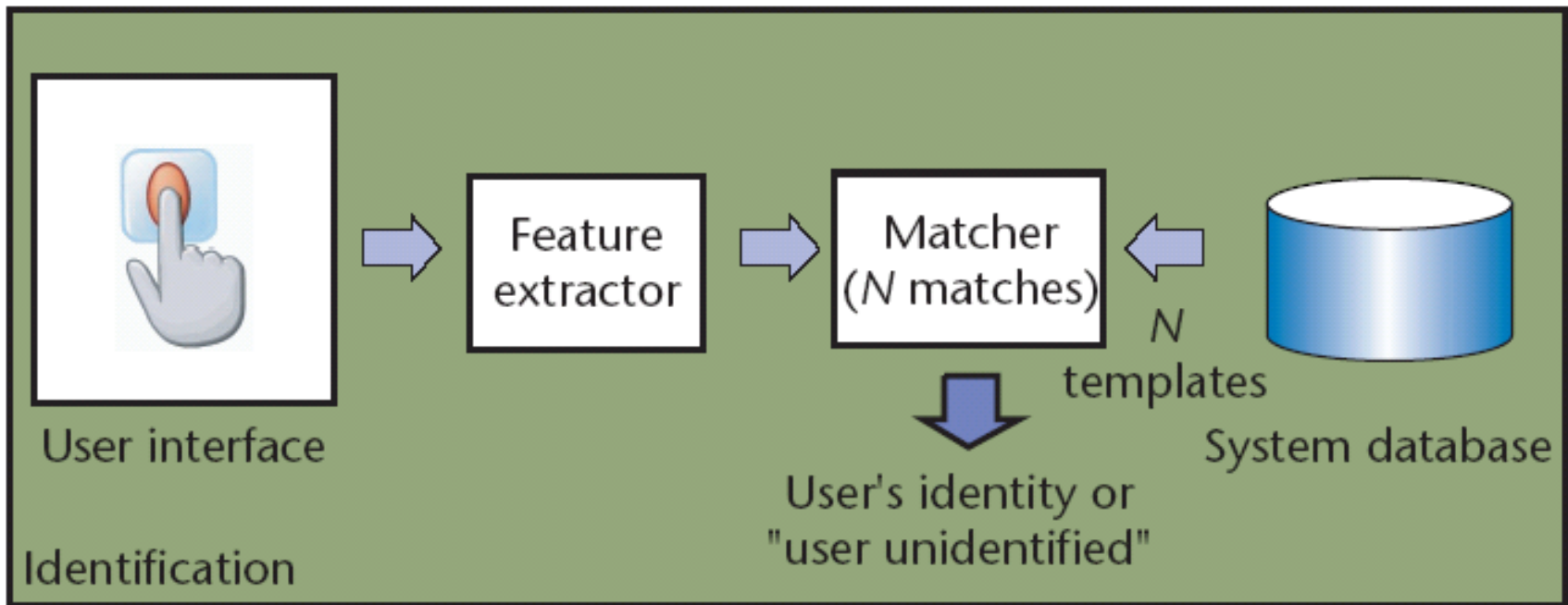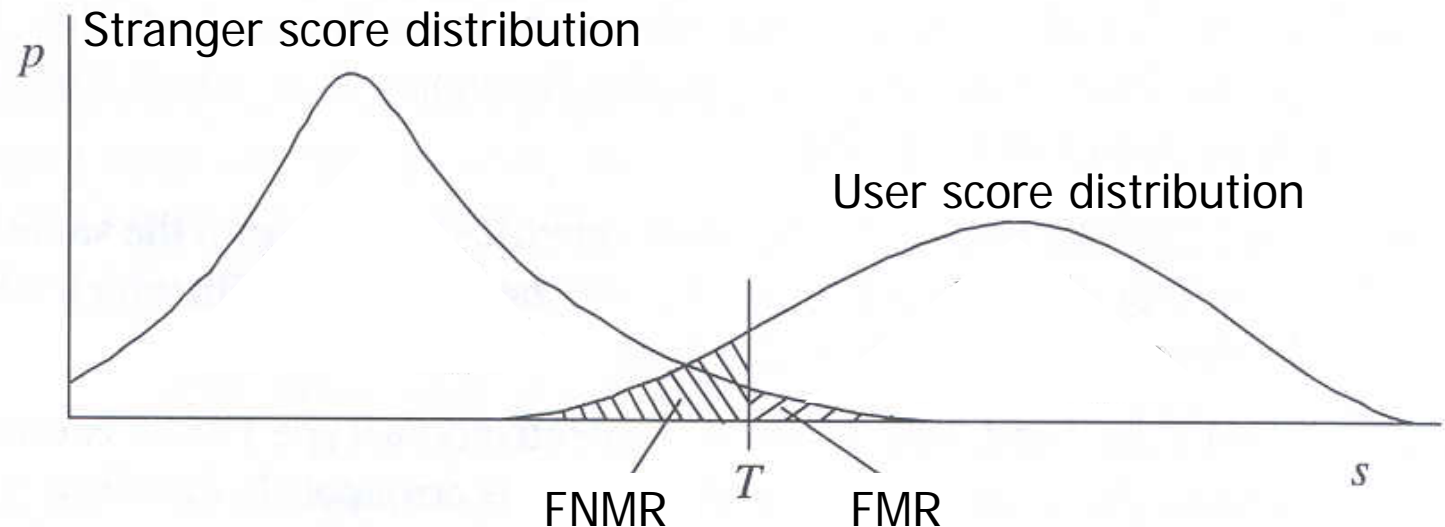
Evaluating Biometrics:

- Features from captured sample are compared against those of the stored template sample
- Score *s* is derived from the comparison.
  - Better match leads to higher score.
- The system decision is tuned by threshold *T*:
  - System gives a match (same person) when the sample comparison generates a score *s* where s ≥ *T*
  - System gives non-match (different person) when the sample comparison generates a score *s* where *s < T*

Matching algorithm characteristics

- True positive
  - User's sample matches → User is accepted
- True negative
  - Stranger's sample does not match → Stranger is rejected
- False positives
  - Stranger's sample matches → Stranger is accepted
- False negatives
  - User's sample does not match → User is rejected
- False Match Rate and False Non-Match Rate

  FMR = (# matching stranger samples) / (total # stranger samples)

  FNMR = (# non-matching user samples) / (total # user samples)

- T determines tradeoff between FMR and FNMR

Evaluating Biometrics: System Errors

- Comparing biometric samples produces score *s*
- Acceptance threshold *T* determines FMR and FNMR
  - If *T* is set low to make the system more tolerant to input variations and noise, then FMR increases.
  - On the other hand, if *T* is set high to make the system more secure, then FNMR increases accordingly.
- EER (Equal Error Rate) is the rate when FMR = FNMR.
- Low EER is good.

Spoofing Biometrics: Presentation Attacks

- It is relatively simple to trick a biometric system
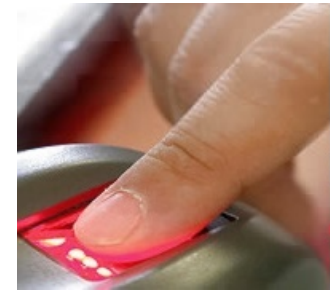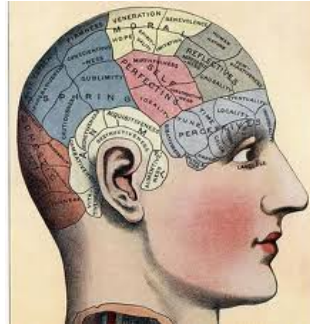 - Terminology: *Presentation Attacks*

False finger

False face

- Biometric authentication on smartphones is insecure
- PAD (Presentation Attack Detection) is the subject of intensive research, to make biometrics more secure
- Alternative solution is to capture biometrics in controlled environments

Authentication: Multi-factor

- Multi-factor authentication aims to combine two or more authentication techniques in order to provide stronger authentication assurance.

- Two-factor authentication is typically based on something a user knows (factor one) plus something the user has (factor two).
  - Usually this involves combining the use of a password and a token
  - Example: BankID OTP token with PIN + static password

Authentication Assurance

- Authentication assurance = robustness of authentication
- Resources have different sensitivity levels
  - High sensitivity gives high risk in case of authentication failure
- Authentication has a cost
  - Unnecessary authentication assurance is a waste of money
- Authentication assurance should balance resource sensitivity
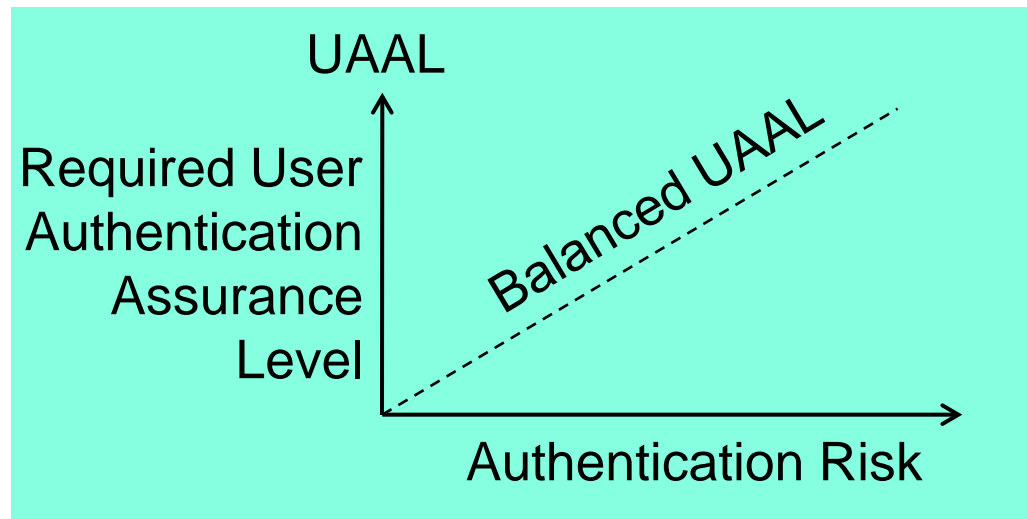
e-Authentication Frameworks for e-Gov.

- Trust in identity is a requirement for e-Government
- Authentication assurance produces identity trust.
- Authentication depends on technology, policy, standards, practice, awareness and regulation.
- Consistent frameworks allow cross-national and cross-organisational schemes that enable convenience, efficiency and cost savings.

Alignment of e-Authentication Frameworks

Authentication Framework	User Authentication Assurance Levels				
OMB / NIST USA 2004 / 2011	Little or no assurance (1)		Some (2)	High (3)	Very High (4)
RAU / FAD Norway 2008	Little or no assurance (1)		Low (2)	Moderate (3)	High (4)
STORK QAA EU 2009	No or minimal (1)		Low (2)	Substantial (3)	High (4)
NeAF Australia 2009	None (0)	Minimal (1)	Low (2)	Moderate (3)	High (4)
e-Pramaan India 2012	None (0)	Minimal (1)	Minor (2)	Significant (3)	Substantial (4)
ISO 29115 ISO/IEC 2013	Low (Little or no) (1)		Medium (2)	High (3)	Very High (4)

UAAL: User Authentication Assurance Level

- UAAL is determined by the weakest of three links:
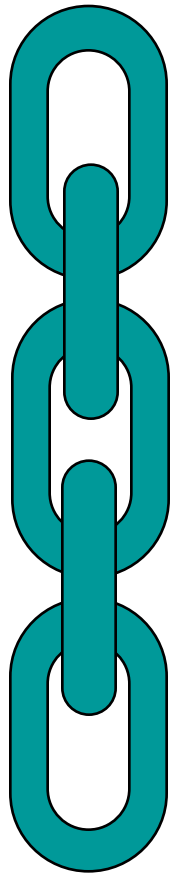
User Identity
Registration Assurance
(UIRA) requirements

Requirements for correct registration:
- Pre-authentication credentials, e.g.
  - birth certificate
  - biometrics

User Credential
Management Assurance
(UCMA) requirements

Requirements for secure handling of credentials:
- Creation
- Distribution
- Storage

User Authentication
Method Strength
(UAMS) requirements

Requirements for mechanism strength:
- Password length and quality
- Cryptographic algorithm strength
- Tamper resistance of token
- Multiple-factor methods

UAAL: User Authentication Assurance Levels

No Assurance	Minimal Assurance	Low Assurance	Moderate Assurance	High Assurance
Level 0	Level 1	Level 2	Level 3	Level 4
No registration of identity required	Minimal confidence in the identity assertion	Low confidence in the identity assertion	Moderate confidence in the identity assertion	High confidence in the identity assertion

Example taken from Australian NeAF 2009

Risk Analysis for Authentication

Determining the appropriate UAAL for an application
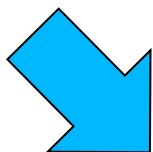
		Impact of e-Authentication failure				
		Insignificant	**Minor**	**Moderate**	**Major**	**Severe**
Likelihood	**Almost Certain**	None (0)	Low (2)	Moderate (3)	High (4)	High (4)
	Likely	None (0)	Low (2)	Moderate (3)	High (4)	High (4)
	Possible	None (0)	Minimal (1)	Low (2)	Moderate (3)	High (4)
	Unlikely	None (0)	Minimal (1)	Low (2)	Moderate (3)	Moderate (3)
	Rare	None (0)	Minimal (1)	Low (2)	Moderate (3)	Moderate (3)

Example: NeAF Australia

RAU Norway
Rammeverk for Autentisering og Uavviselighet (Framework for Authentication and Non-Repudiation)

RAU Level 1: Little or no authentication assurance

Alternative requirements:
- Online self-registration and self-chosen password
- Pre-authentication by providing person number

RAU Level 2: Low authentication assurance

Alternative requirements:

- Fixed password provisioned in person or by mail to user's address in national person register

- OPT calculator without PIN, provisioned in person or by mail to address in national person reg.

- List of OTP (one-time passwords) provisioned in person or by mail to address in national pers. reg.

RAU Level 3: Moderate authentication assurance

Alternative requirements:

- OTP calculator with PIN provisioned separately in person or by mail to address in national pers. reg.
- SMS-based authentication, where enrolment of mobile phone is based on code provisioned in person or by mail to address in national pers. reg.
- Personal public-key certificate with gov. PKI
- List of OTP (one-time passwords) combined with static password and username provisioned in person or by mail to address in national pers. reg.

Example: MinID

RAU Level 4: High authentication assurance

Alternative requirements:

- Two-factor, where at least one must be dynamic, and at least one is provisioned in person (the other by mail to address in national pers. reg. Also requires logging and auditing by third party.

  Examples: Buypass, Confides, BankID

End of lecture