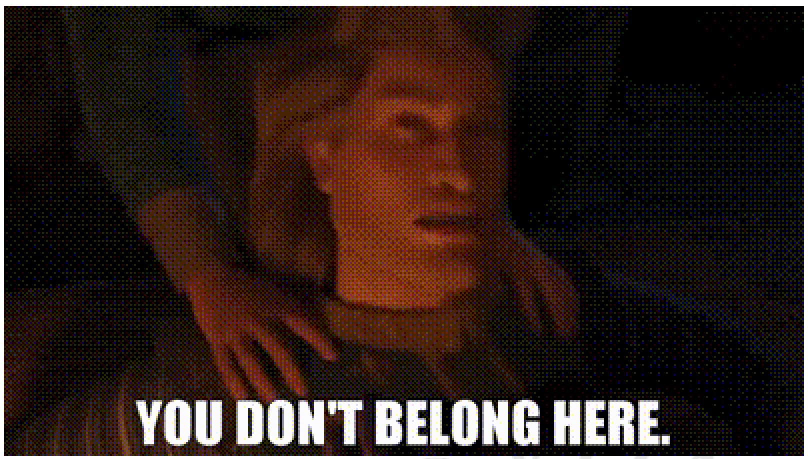
Authentication

"On the Internet, nobody knows you're a dog."

Peter Seiner, The New Yorker, 1993

DTU Compute
Department of Applied Mathematics and Computer Science

Authentication vs. Authorization



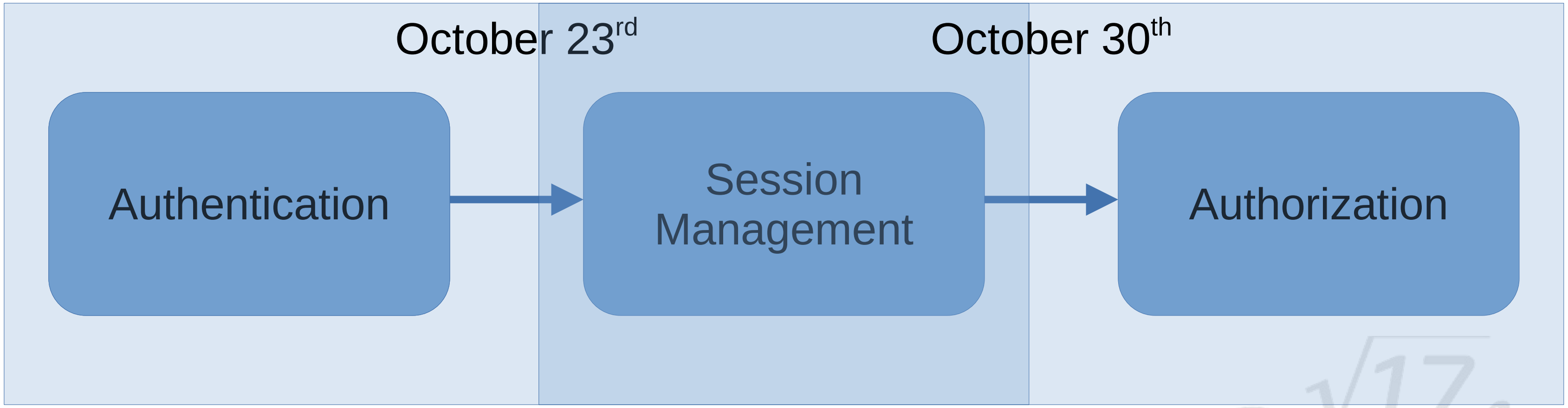
Authentication

“Who are you?”

Authorization

”Are you allowed to do that?”

Authentication, Session Management and Authorization



The purpose of authentication

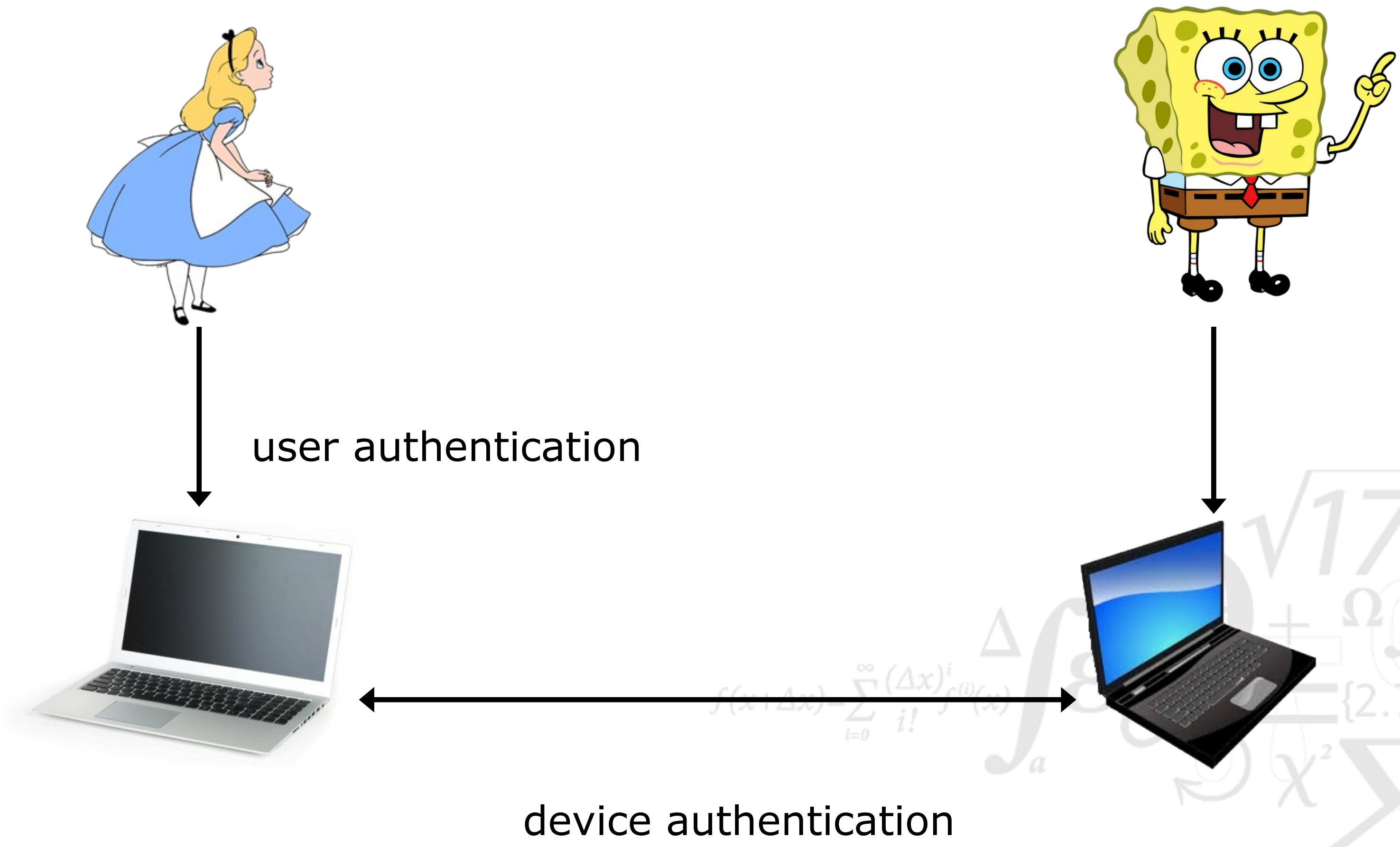
- Authentication is the process of verifying a claimed identity

- Allowing people to access the system
  - Ability to authenticate demonstrates authorization to access system

- Binding an identity to system entities
  - Authorizations are linked to identities (or roles that are assigned to ID)

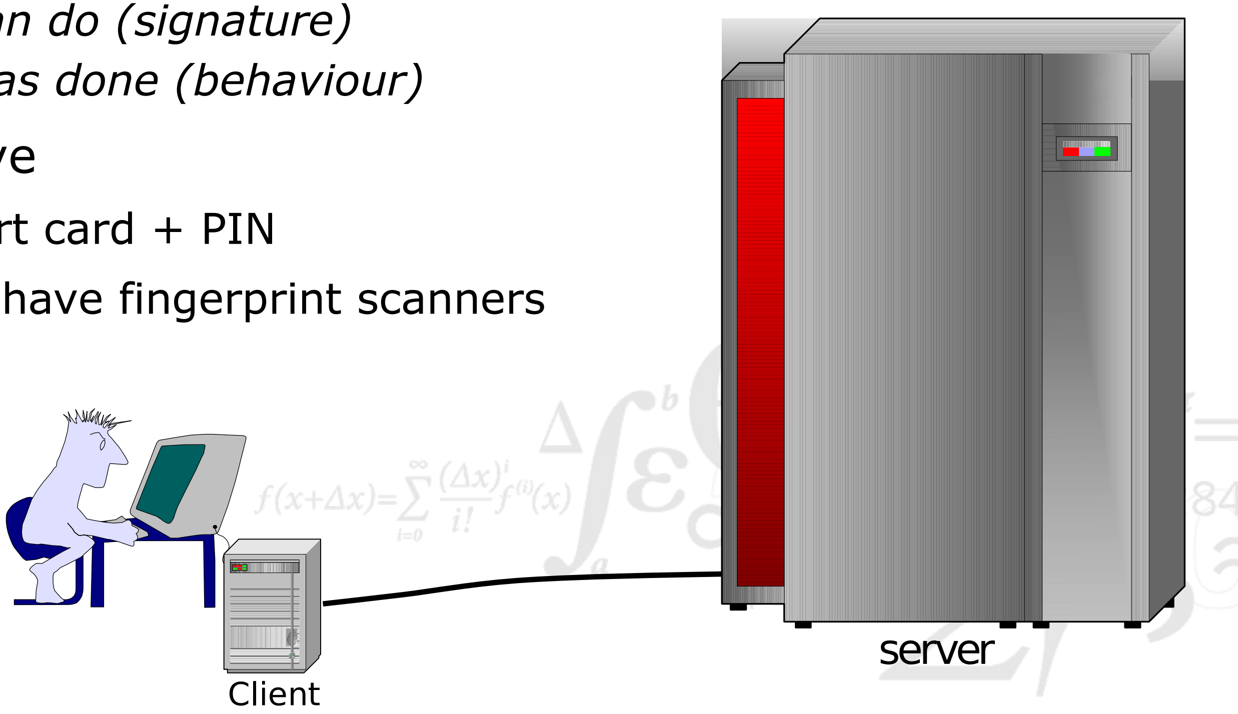
- Associating a real world identity (transitively) with system events
  - Accountability facilitated through recording ID of requesting entities

Different Types of Authentication

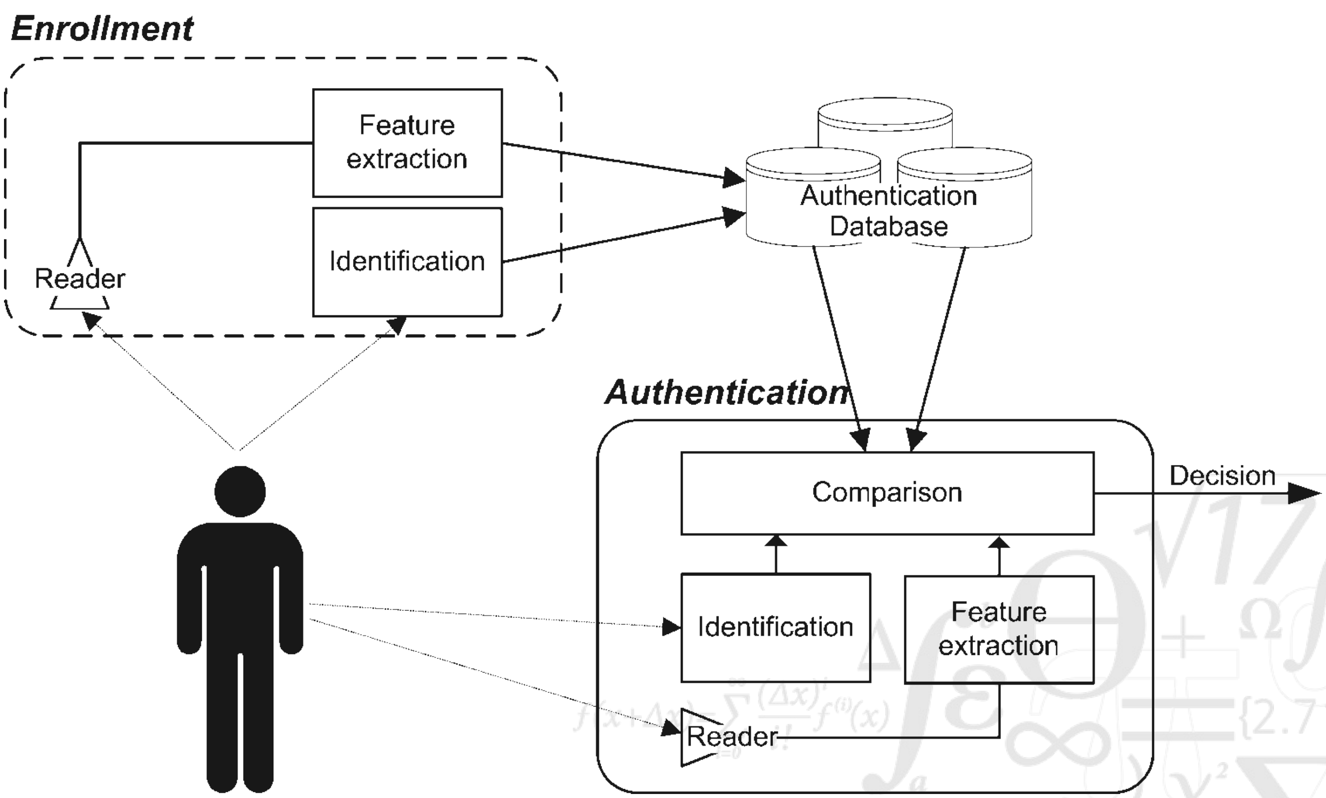


User Authentication

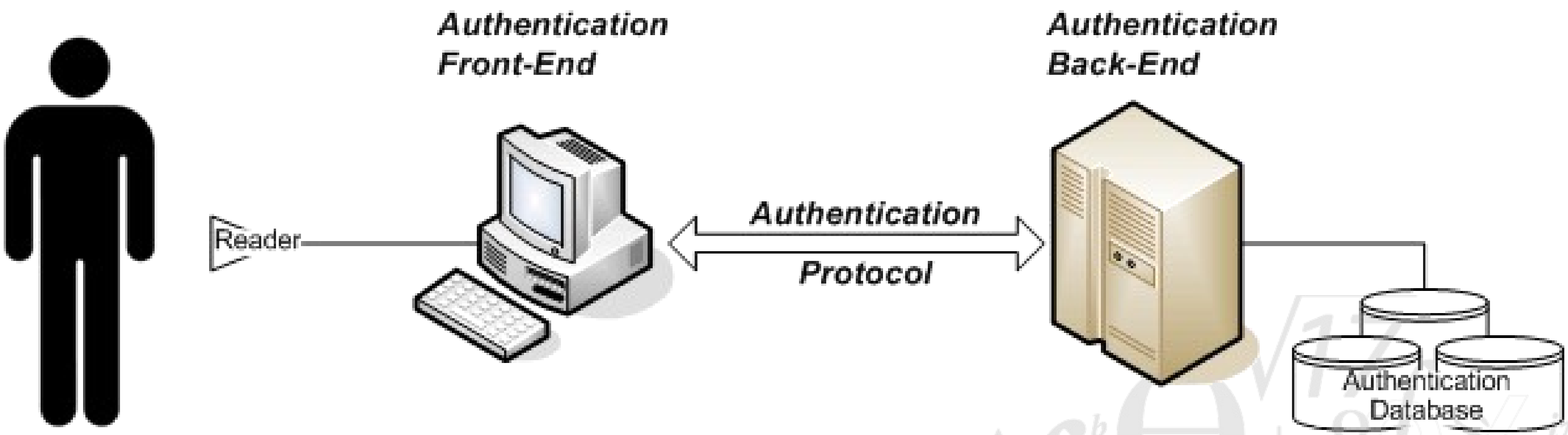
- Establish the identity of principals by means of:
  - something he knows (*shared secrets, e.g., password, PIN*)
  - something he possesses (*smart card, USB token, mobile phone*)
  - something he is (*fingerprint, face, voice, retina scan*)
    - *something he can do (signature)*
    - *Something he has done (behaviour)*
- Combinations of above
  - VISA cards has smart card + PIN
  - Many smart phones have fingerprint scanners



Authentication Mechanism



General Model of Authentication



Where can this go wrong?

Something you know – Password Authentication

- Most operating systems rely exclusively on passwords

  login: username

  Password: *********

- Password is checked by “login” program
  - Prompts for username
  - Prompts for password
  - Performs one-way function on password and salt (hash)
  - Compares digest with password and salt stored in password file

Password Authentication in OS

- Example – Unix /etc/passwd and /etc/shadow files

```
Username:Password:UserID:GroupID:Gecos:HomeDirectory:Shell
Username:Password:LastChanged:Minimum:Maximum:Warn:Inactive:Expire
```

  - Example

```
bill:5fg63fhD3d5gh:157:5:Bill Smith:/home/bill:/bin/sh

bill:$6$YTJ7JKnfsB4esnbS$5XvmYk2.GXVWhDo2TYGN2hCitD/
  wU9Kov.uZD8xsnleuf1r0ARX3qodIKiDsdoQA444b8IMPMOnUWDmVJVkeg1:19446:
  0:99999:7:::
```

- Example – Windows: passwords are stored in the Security Account Manager (SAM) database file in the form of a LM hash or NTLM hash

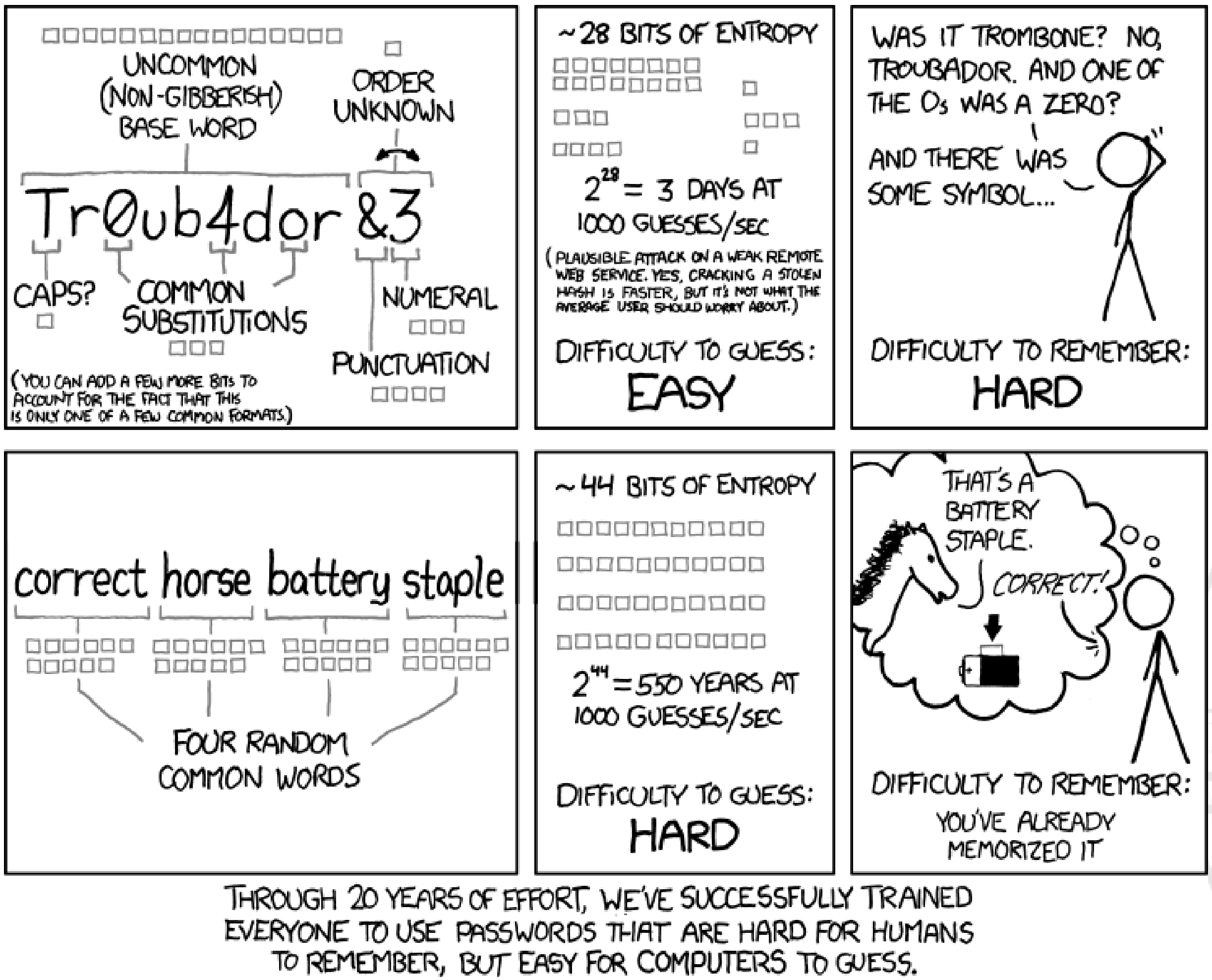
Passwords

- Single password is the most common method for authentication
  - Simple and "generally accepted"
    - *Everyone knows how they work*
  - Cheap to implement
    - *No additional hardware required*
- Password Security
  - Anyone who knows the password will be authenticated
  - Passwords must be difficult to guess
    - *Resistance to brute force attacks*
      - Passwords must be long (more than 12 characters)
      - Passwords must be complex (difficult to remember)
    - *Resistance to guessing-/dictionary attacks*
    - *Passwords should be unique (no reuse across websites)*
- Remembering many long complex passwords is hard for users

Passphrases

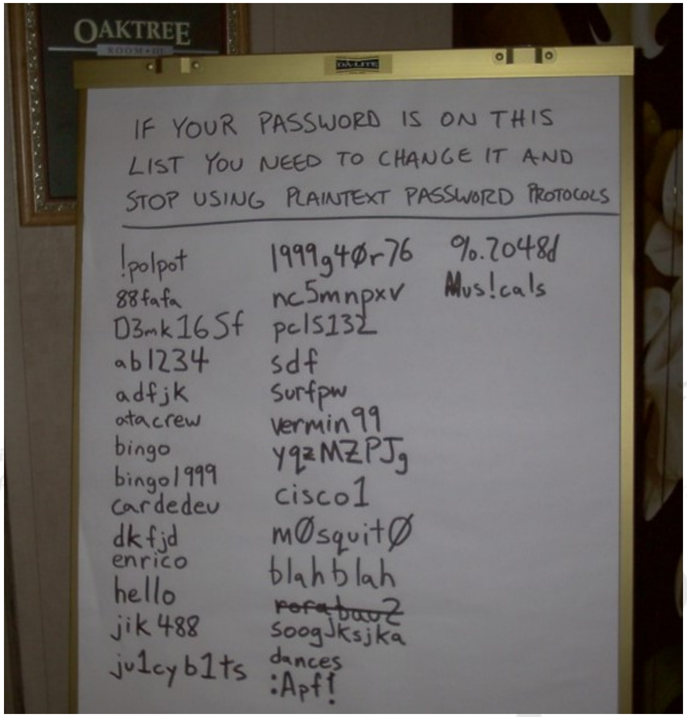
- Passphrases include several words
  - Technology is similar to password implementation
- Three major challenges
  - Usability vs. Security
    - *Random words (similar to passwords but with a larger alphabet)*
    - *Natural language sentences (obey syntax and grammar)*
      - Length improves security
      - Structure reduces entropy and thereby security
  - Passphrase retention
    - *Structure of passphrases makes them easier to remember*
    - *Common advice to construct and recall complex passwords*
  - Passphrase Entry
    - *Time consuming (typing more characters)*
    - *Error prone (getting all the characters exactly right)*
- Improve usability by accepting passphrases with a few small errors

Password Strength



Attacks on Password Entry

- Passwords can be read over the shoulder (shoulder surfing)
  - Computers, payment terminal design, pay phones, smartphones, …
  - Try to hide what you are typing
- Passwords may be compromised every time they are used over a network (packet sniffers)
  - Prevented by One Time Passwords (OTP)
- Password can be intercepted in transit between user and system
  - Trojan horse login screen
  - Trusted path between user and system is required
    - *ctrl-alt-del on Windows gives a genuine login screen*



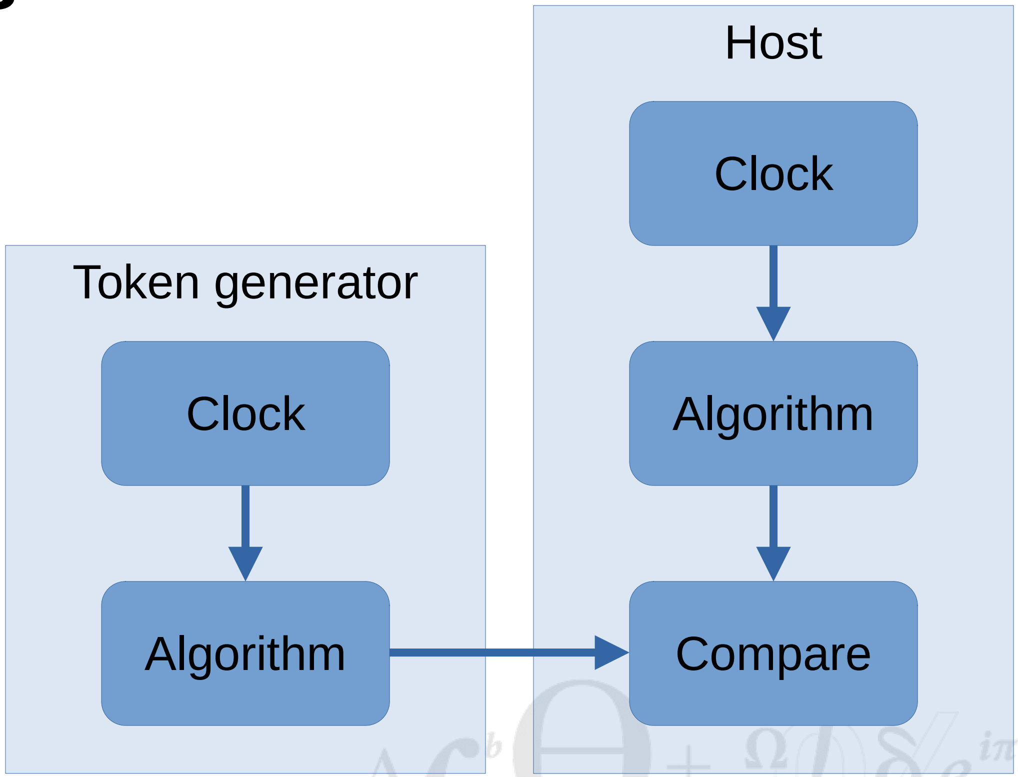
Something you have – Authentication Tokens

- Authentication tokens of the form One-Time Passwords (OTP) come mostly in two flavours:
  - Synchronised generators must produce the same sequence of random OTP both in the token and the servers (ex: time-based tokens)
  - Challenge-Response tokens: a challenge is sent by the host and the user can send a response based on this challenge

MitID token generator

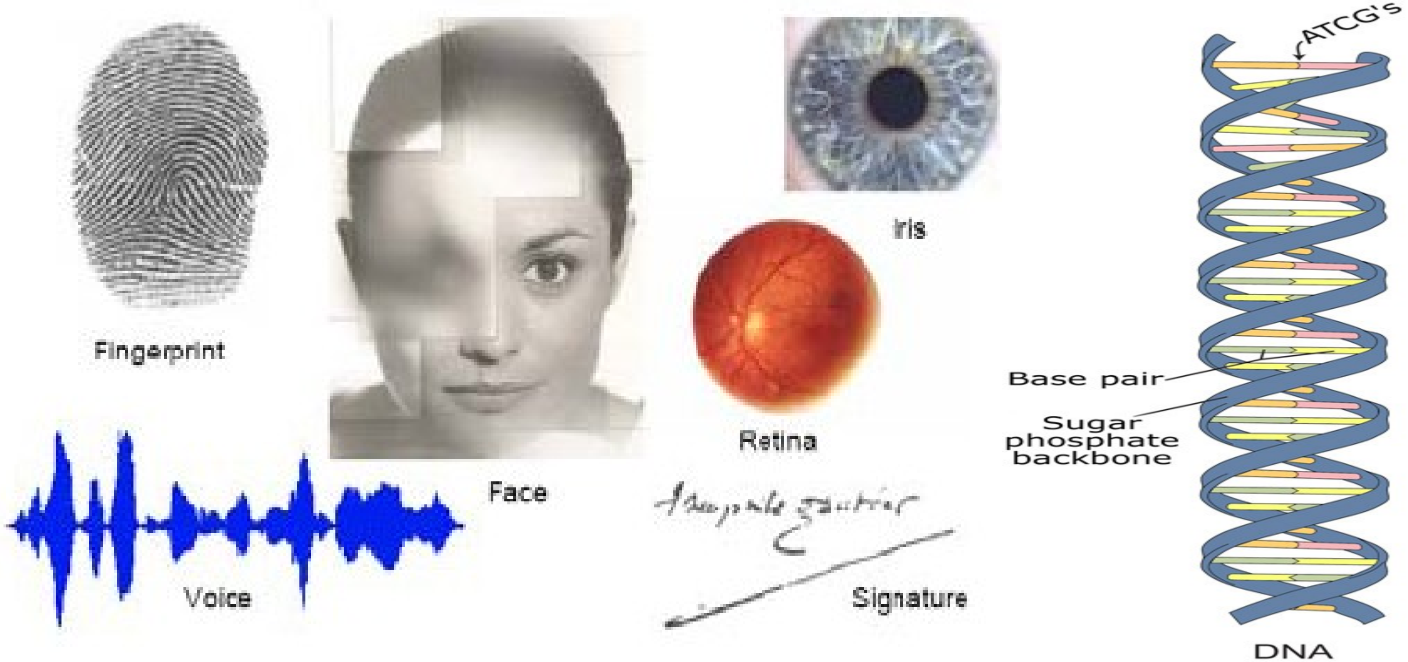
Synchronised tokens

- The algorithm is a pseudo-random generator
- Suceptible to time drift
- If seeds of the pseudo-random generators leak, attackers could generate the OTP: RSA SecureID hack



Something you are – Biometrics

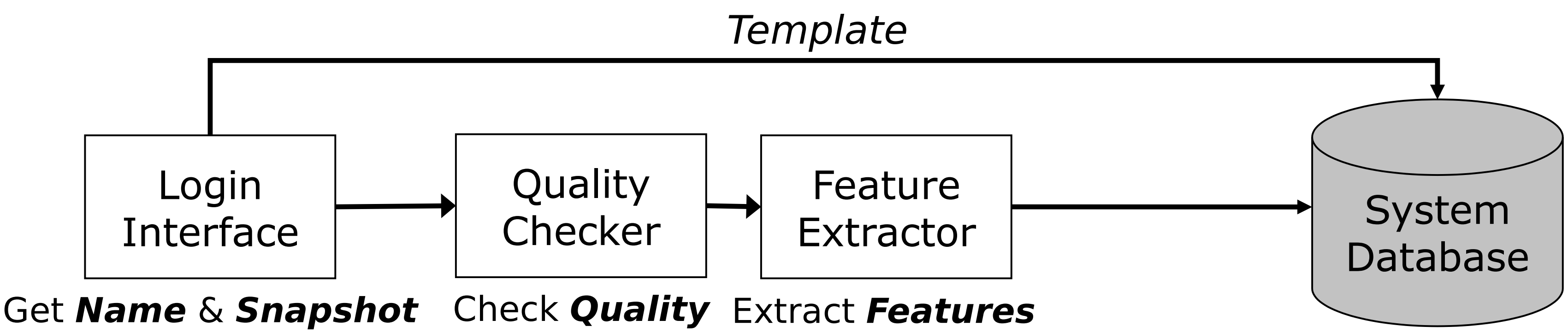
Biometrics identify people by measuring some aspect of individual anatomy or physiology (hand geometry or fingerprint), some deeply ingrained skill, or other behavioural characteristic (handwritten signature), or something that is a combination of the two (voice)



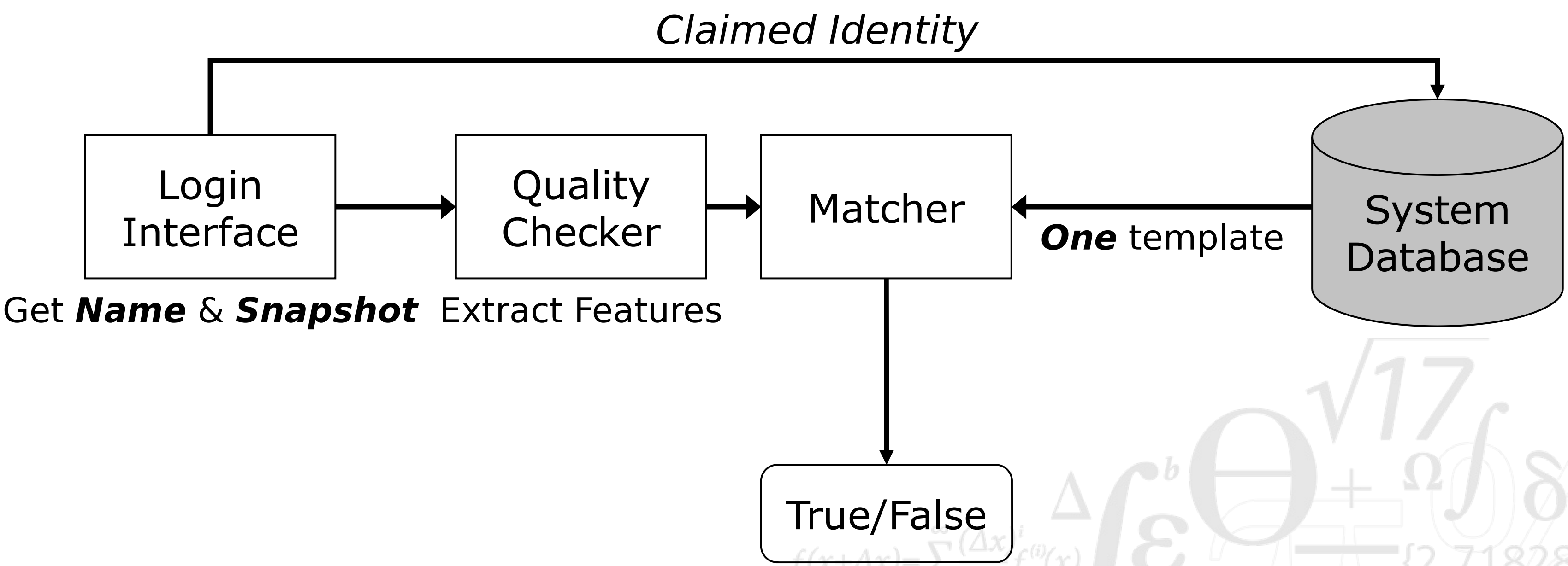
Biometric Authentication Systems

- Biometric systems have three types of operations
  - Enrollment (just like any other authentication system)
  - Verification (biometric authentication)
    - *match 1:1   one captured template to one stored template*
  - Identification
    - *match 1:N one captured template to N (or all) stored templates*

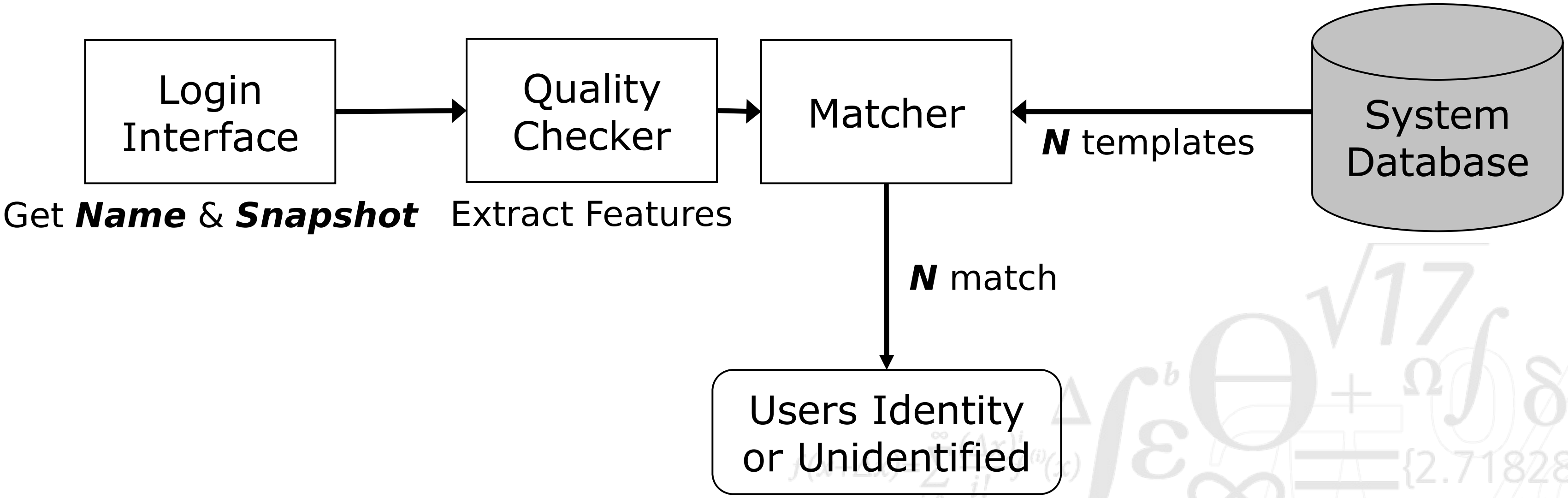
Enrollment in Biometric Systems



Verification in Biometric Systems



Identification in Biometric Systems

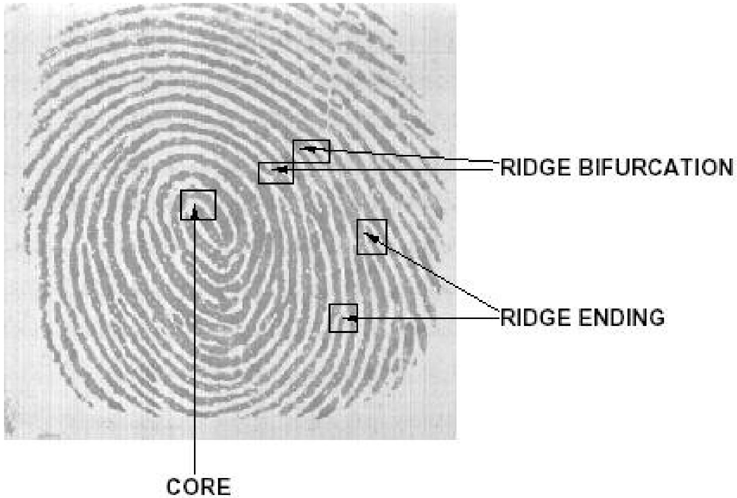


Handwritten Signatures

- Identifying handwriting is difficult, experts have an error rate of 6.5%, non experts have an error rate at 38%
- Problem with false accepts and rejects
  - false accepts result in fraud
  - false rejects result in insult (bad for business)
  - systems can be tuned to favour one over the other
- Optical systems are unreliable
- Signature tablets record shape, speed and dynamics of signature - more reliable

Fingerprints

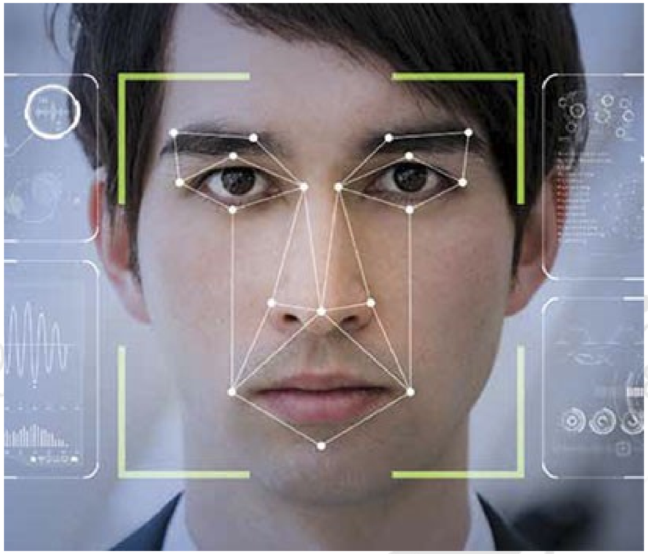
- Fingerprints have been used as signatures for centuries
- They are currently used to identify criminals (affects acceptability)
- Measures unique characteristics in a fingerprint (minutiae)
  - Crossover
  - Core
  - Bifurcations
  - Ridge ending
  - Island
  - Delta
  - Pore



- Error rate can be affected by scars, wear, etc.
  - Very old and very young have weak fingerprints
- Many systems defeated by Gummy Fingers
  - Requires liveness detection

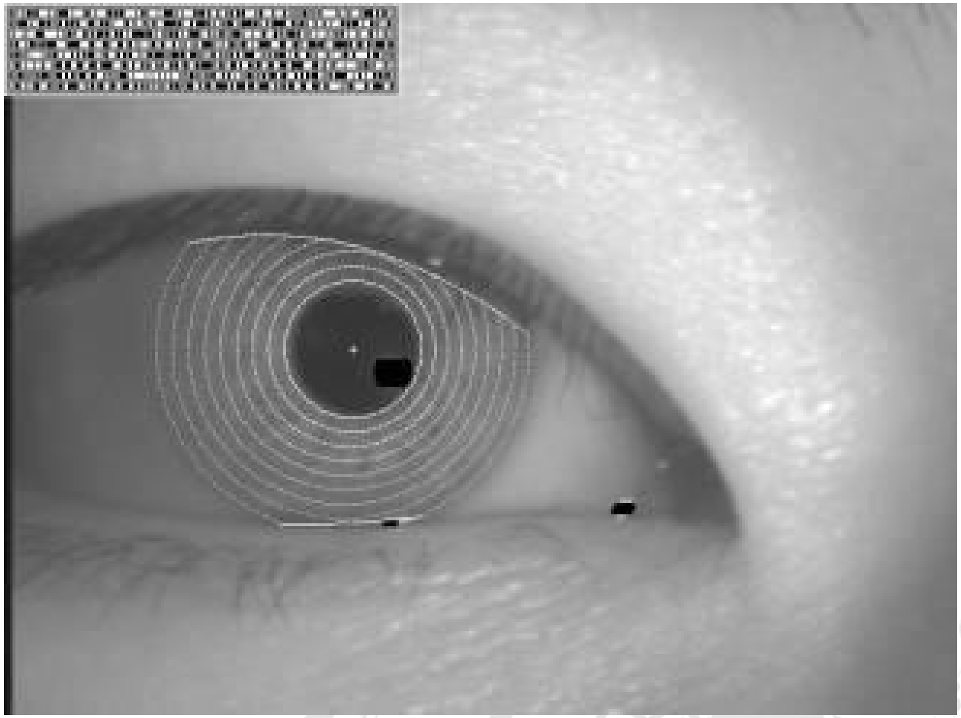
Face Recognition

- Face recognition is the oldest and most widespread form of identification
  - Manually used in photo ID (passport, …)
  - Increasingly used in smartphones
- Uses off-the-shelf camera to measure the following facial features:
  - Distance between the eyes
  - Distance between the eyes and nose ridge
  - Angle of a cheek
  - Slope of the nose
  - Facial Temperatures (with IR camera)
- Multiple cameras allows 3D models
  - Stereo vision (2 normal cameras)
  - 1 normal camera + 1 IR camera



Iris Scan

- Measures unique characteristics of the iris
  - Ridges (rings)
  - Furrows
  - Straitions (freckles)



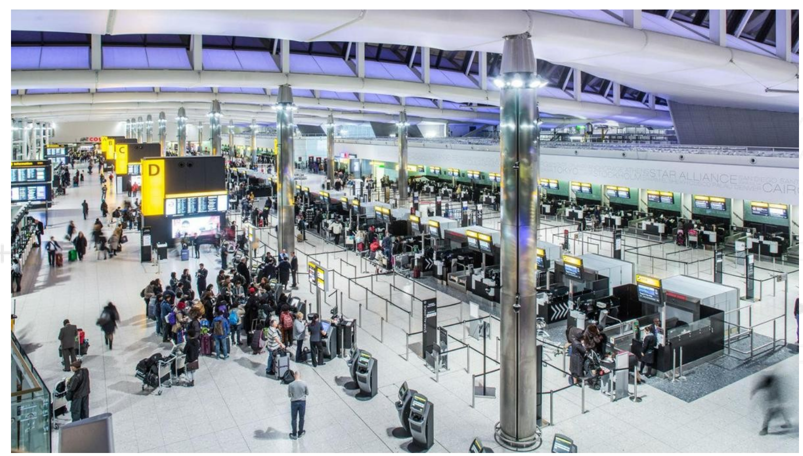
- Simple unattended systems can be defeated by photographs

Voice recognition

- Voice recognition identifies the speaker
  - Not to be confused with speech recognition (identifies what she says)
- Can be used for authentication over the phone
  - Include context to bind authentication to transaction: "Transfer 200 Kr to account 123456789"
- Systems exist with < 1% error rate
- Tape recorders distort the voice enough to prevent "replay attacks"
- Digital recorders may be used to attack voice recognition systems in the future

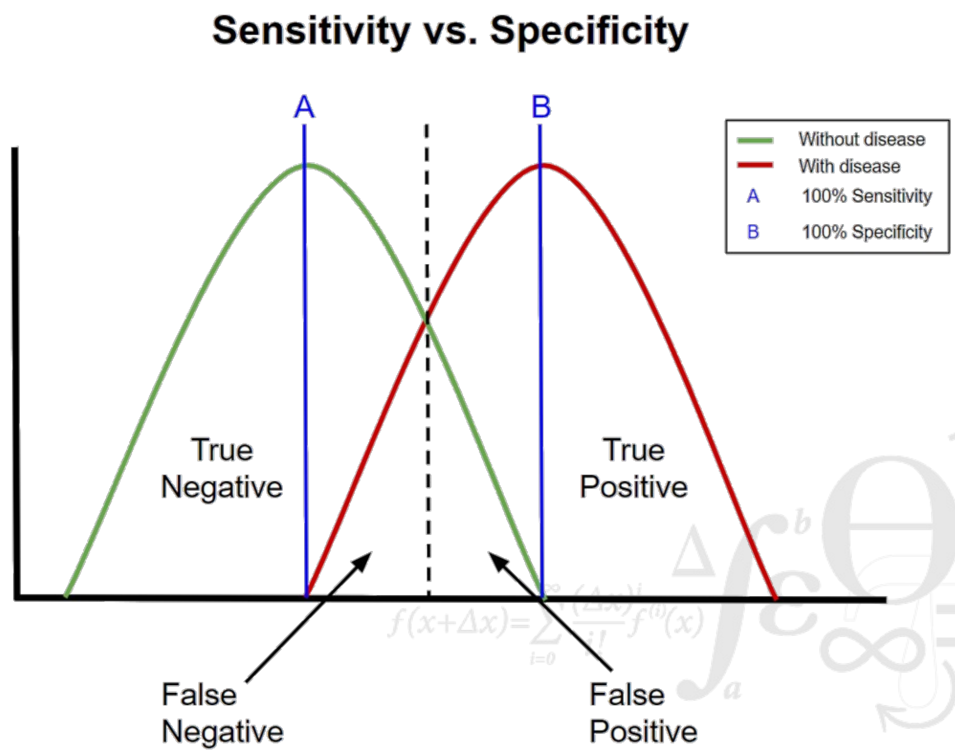
Biometric Authentication Summary

- Automated identification systems (scanners) have been developed
- Quality of system depends on accuracy (error rate)
- Error rate is often below 1%
  - What does error rate < 1% mean in practice?
    - *Heathrow Airport has ~128,000 passengers arriving every day (2021)*
    - *Around 5,333 passengers arrive every hour*
    - *Around 88 passengers arrive every minute*
    - *Around 1 error every minute for passengers arriving at Heathrow*
  - Reducing error rate by an order of magnitude makes little difference



Threshold Based Authentication Systems

- Comparison of presented features with stored template
  - Rarely an exact match, so verification is based on threshold



Authentication Mechanism Quality Metrics

- Threshold based mechanisms give four possible results

	Is the person claimed	Is **not** the person claimed
Test is positive (there is a match)	True Positive	False Positive
Test is negative (there is no match	False Negative	True Negative

$$\text{Sensitivity} = \frac{\text{True Positive}}{\text{True Positive + False Negative}}$$

$$\text{Specificity} = \frac{\text{True Negative}}{\text{False Positive + True Negative}}$$

$$\text{Accuracy} = \frac{\text{True Positive + True Negative}}{\text{True Positive + False Positive + False Negative + True Negative}}$$

$$\text{Prevalence} = \frac{\text{True Positive + False Negative}}{\text{True Positive + False Positive + False Negative + True Negative}}$$

Multi-Factor Authentication (MFA)

Multi-Factor Authentication combines authentication methods to ensure that a leaked password is not enough (and it should be the standard):

- Something you have: OTP passwords, SMS, email addresses
- Something you are: biometrics (but consider the risks)

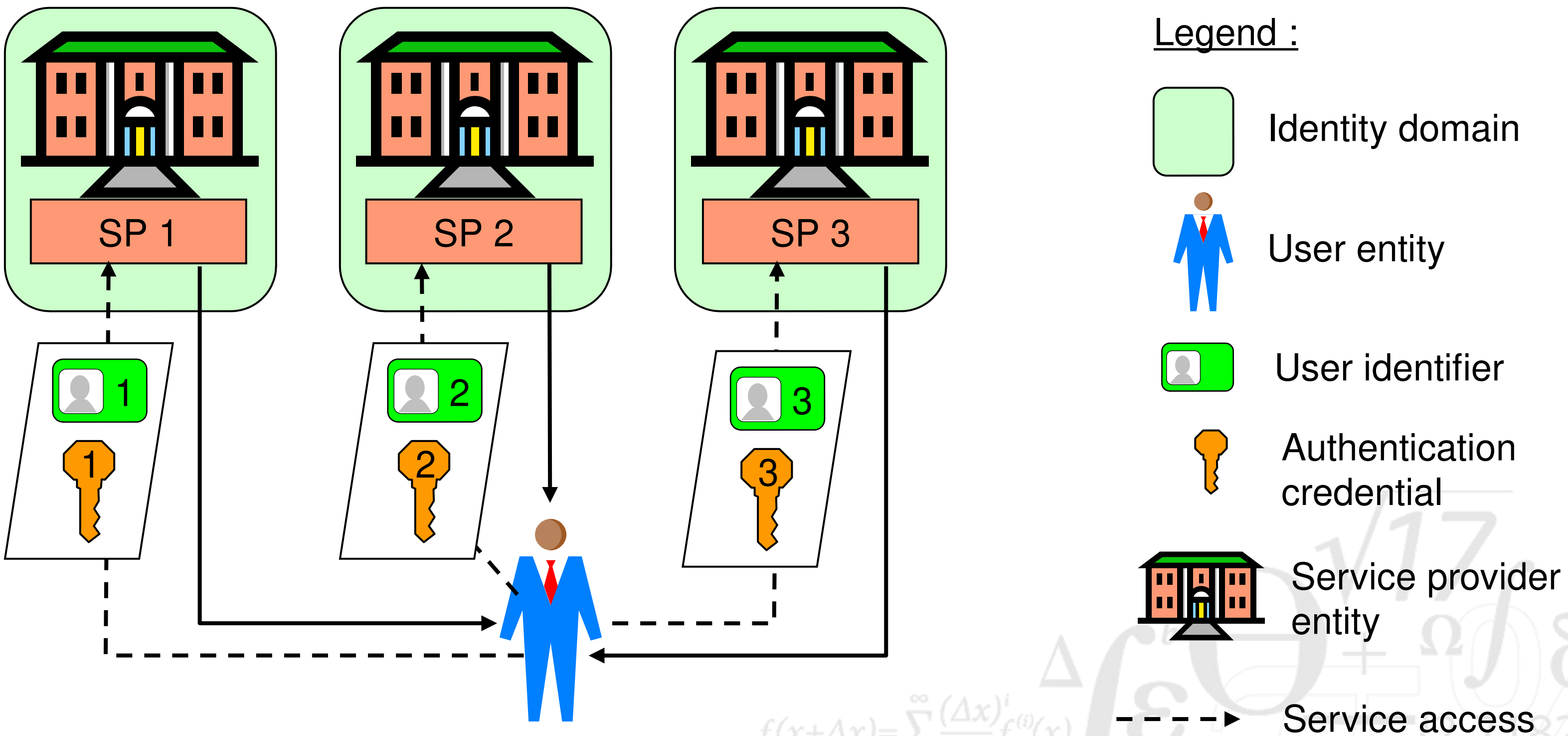
Break

Identity Management Models

- Authentication in distributed systems require an agreed model of identities and authenticatior

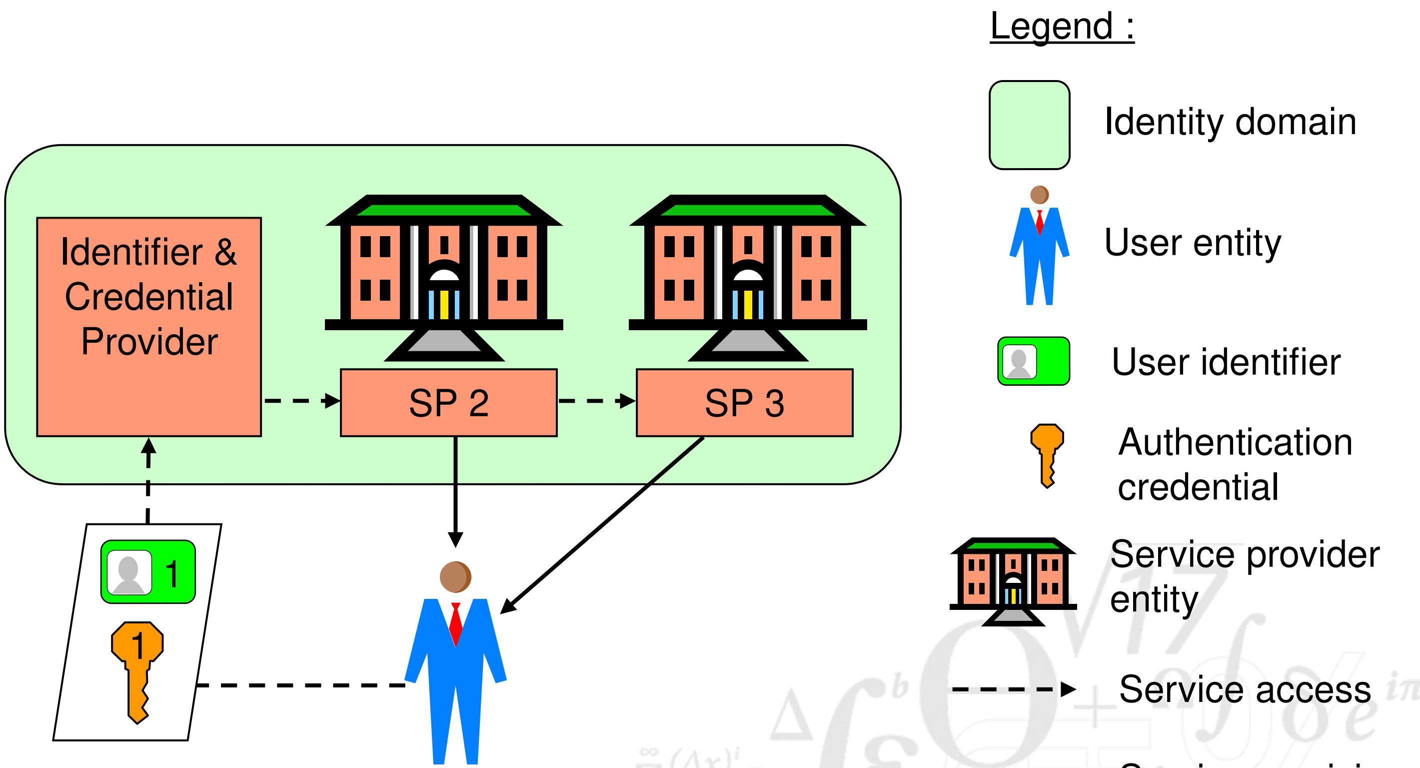
- Three fundamental models:
  - Identity Silos
  - Single Sign-On systems
  - Federated Identity models

Isolated User Identity Model



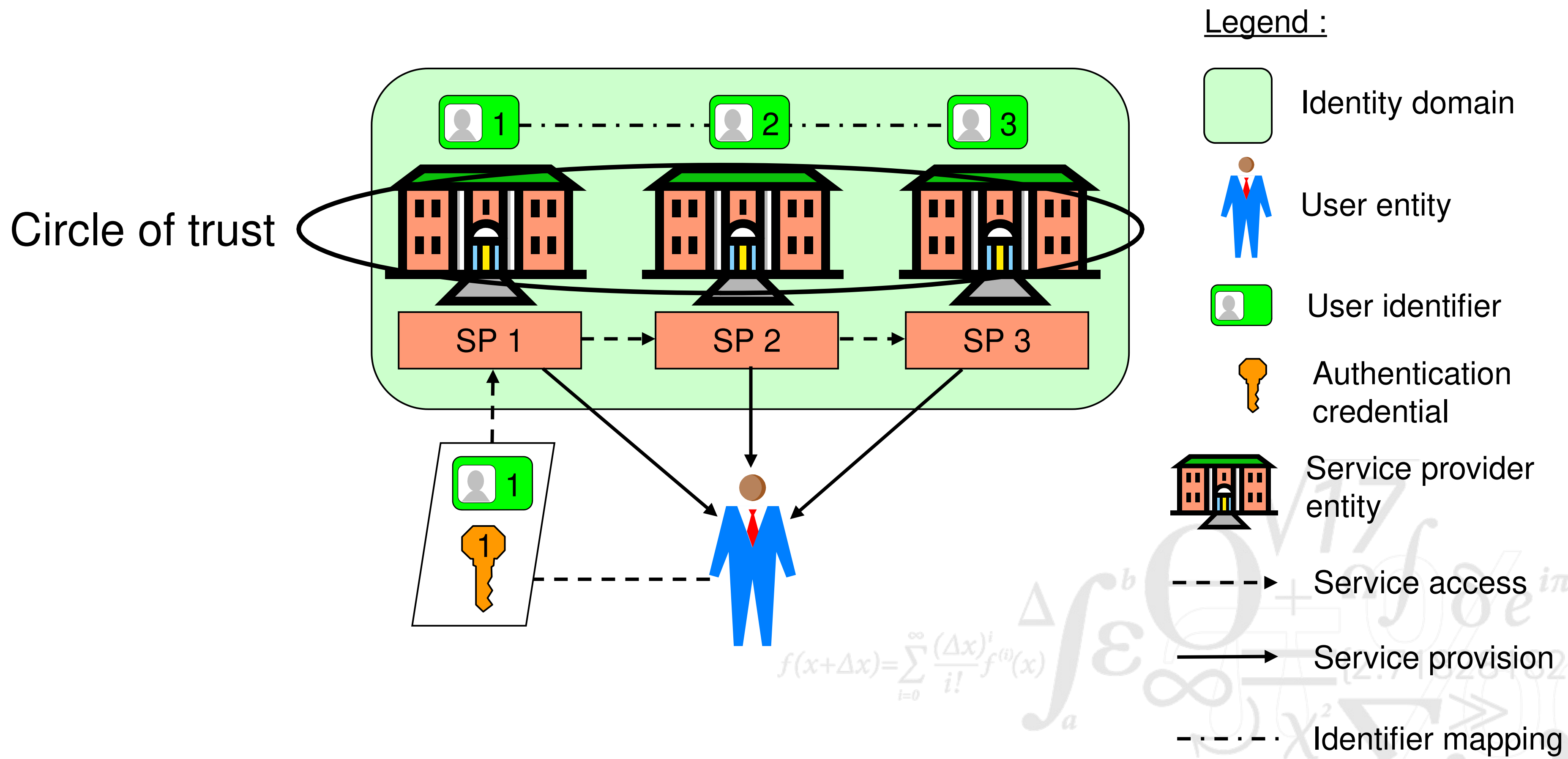
Examples: It's everywhere

Single Sign On (SSO) Systems



Examples: Kerberos, MitID

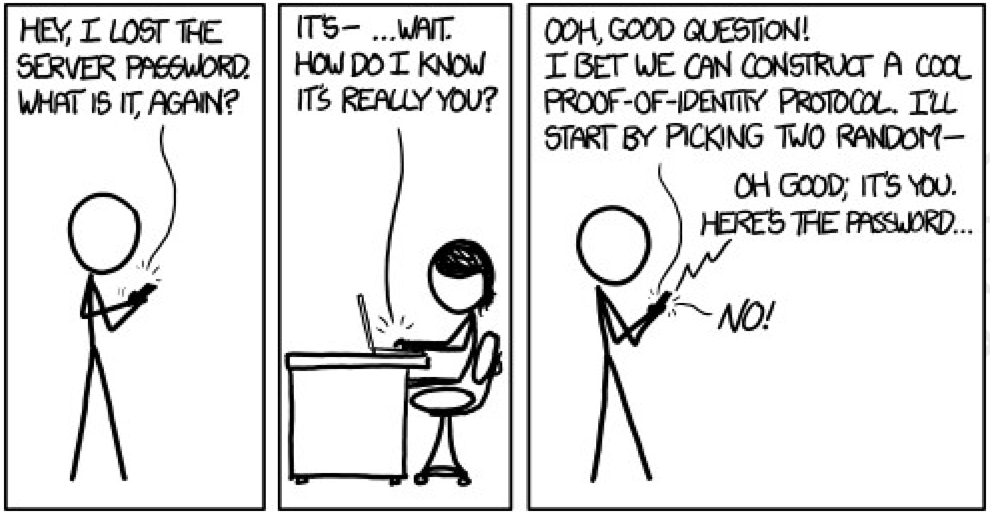
Federated Identity Model



Examples: SAML2.0, WS-Federation, Shibboleth, Eduroam

Authentication Protocols

- Authentication protocols extend authentication across networks
  - Authentication of remote users
  - Authentication of remote devices
  - Authentication of intention to authenticate
    - *Protection against relay attacks*
    - *Protection against replay attacks*

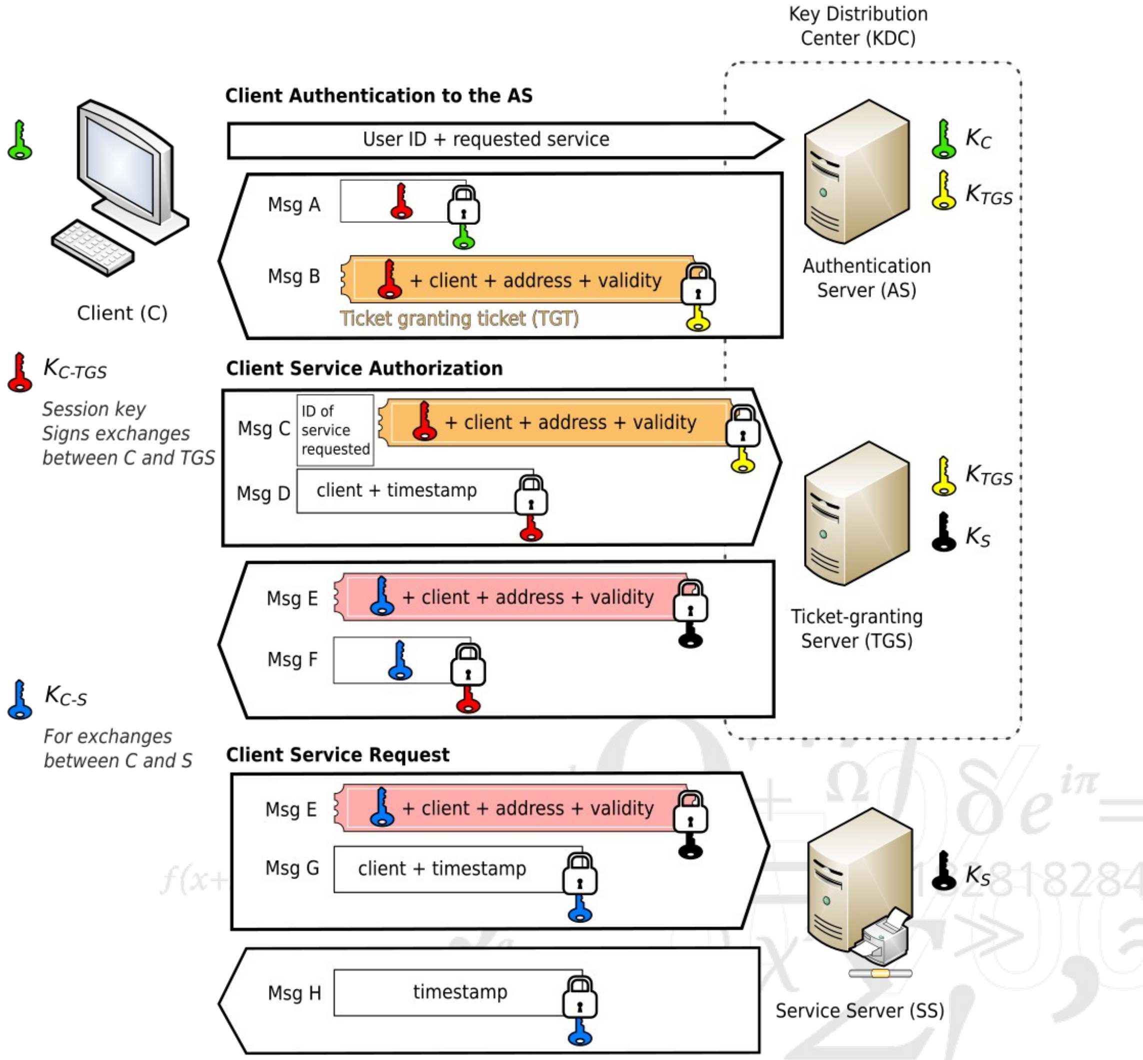


Kerberos

- Project Athena at MIT (mid to late 1980s)
- Hundreds of diskless workstations
  - Open terminal access, no physical security
  - Insecure network
- Few servers (programs, files, print, …)
  - Physically secure

- Kerberos designed to rely on symmetric cryptography
  - Resistent to quantum computers

Kerberos Protocol



Insecure Workstations

- What happens to tickets after a user has logged out?
  - An opponent could log on to the workstation and use the tickets
  - Could be explicitly destroyed when user logs out
  - Sniffer could be used to capture tickets, hacker may then login to the same workstation and use the tickets (replay session)

- This demonstrates common problems in authentication
  - In-memory caches of data and re-use of physical resources
  - Problem with secure revocation of authorisations
  - Ensuring freshness of authentication data

Limiting Ticket Lifespan

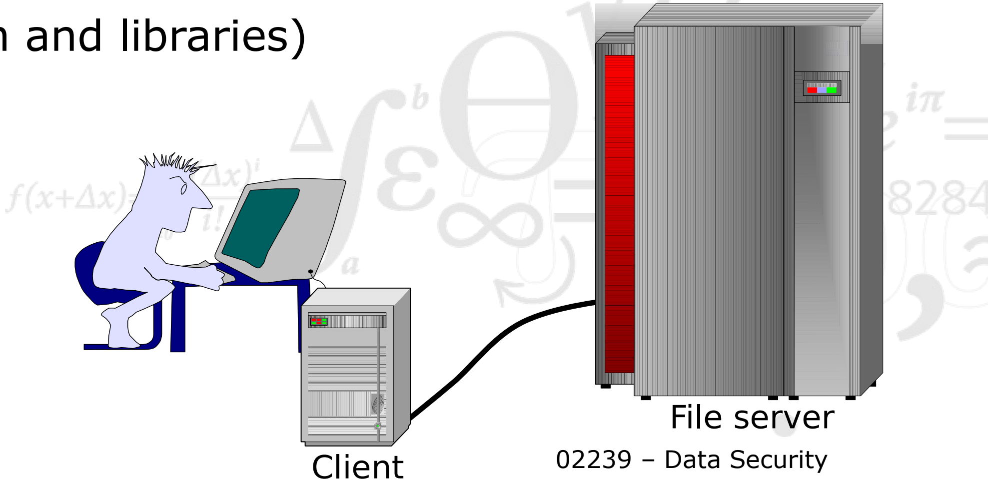
- KDC timestamps ticket when it is issued
- KDC includes lifespan along with timestamp
- Remaining problems:
  - Workstation clocks must be synchronized
  - What should the lifespan of a ticket be?

Verifying Tickets

- Authentication relies on the following tests:
  - Can the service decrypt the ticket?
  - Has the ticket expired?
  - Do the username and workstation address correspond?
- The tests prove:
  - The ticket came from KDC
  - The ticket is still valid
  - Failure proves that the ticket is false, success does not prove anything, tickets can be stolen and reused on the same workstation

User Authentication in Distributed Systems

- How can we prove that a remote user is who he claims to be?
- User authentication takes place remotely at the client
  - Needs federated identity management
- Communication channel must be secure (CIA)
  - Integrity is required
  - Confidentiality and Availability are required
- Device authentication needed to determine if client is trusted
  - Client applications (programs that handle authentication tokens)
  - Client system (operating system and libraries)
  - Client hardware



Session management

Authenticating is boring (think about if you had to enter your password every 5 minutes)!

Until now, we have seen how to authenticate, but many authentication system are stateless, let's see how we can make them stateful.

Sessions

- After a user signs in, the server needs to create a session for them
- The server needs to generate a secure session ID with a secure random generator
- Sessions usally needs to be invalidated: the server can set session lifetime
- Users need to store their session ID. In browsers, there are mainly 2 ways:
  - ***Cookies****:* a number of attributes are needed in order to make this secure (e.g. Http-Only but not only) and CSRF protections must be implemented
  - ***Web Storage API****:* sessions ID can be stored in localStorage or sessionStorage but this is vulnerable to XSS attacks amongst others

Threats against sessions

- **Cross-site Request Forgery** (CSRF): if session tokens are stored in cookies, sessions are vulnerable to CSRF. Attackers can steal the cookies and use it to make authenticated requests => CSRF can be mitigated via CSRF tokens

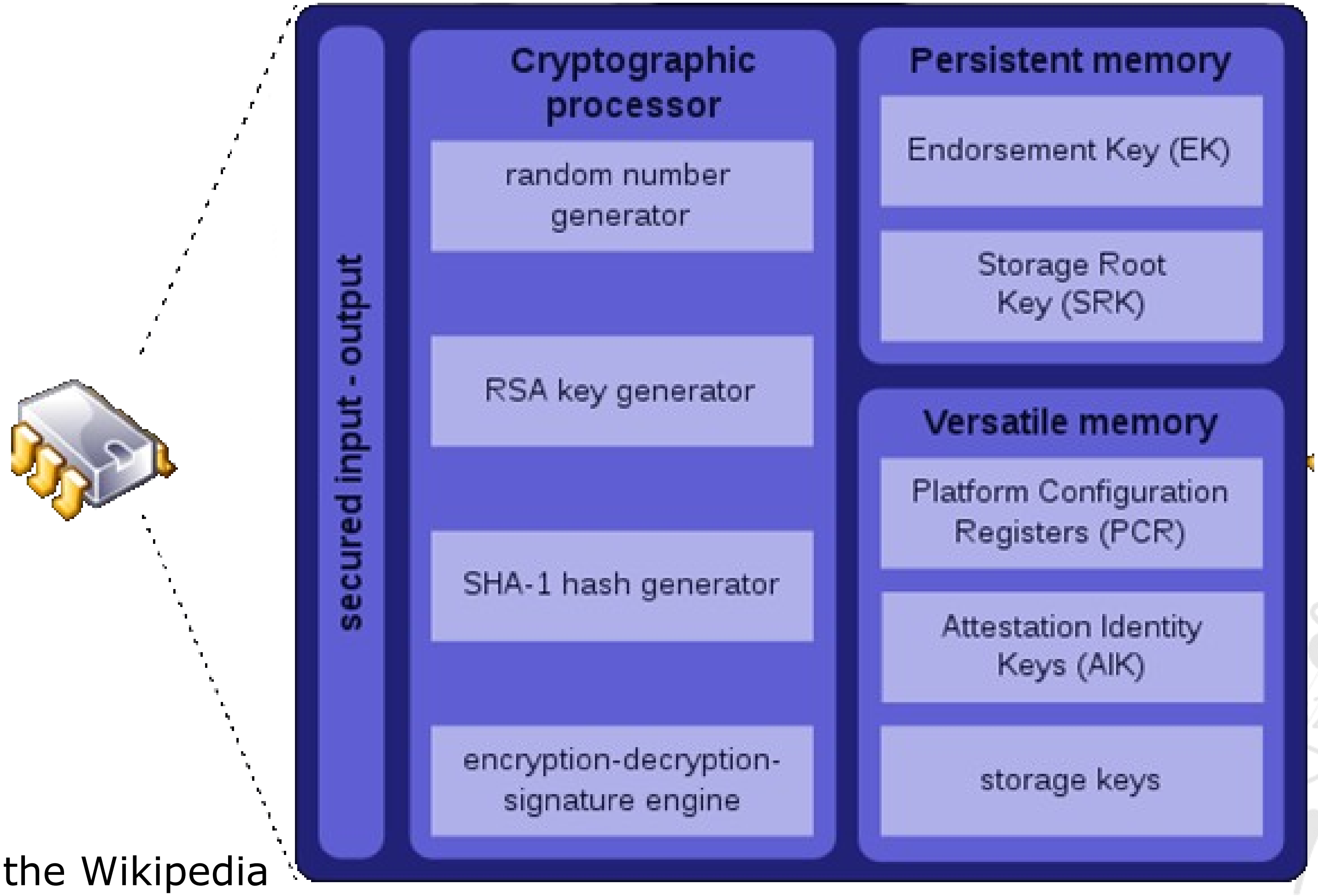
- **Session hijacking**: sessions can be stolen via cross-script scripting (XSS), man-in-the-middle (MITM), session sniffing, etc. => sessions can be invalidated if the user is in a unusual location

- **Session fixation attacks**: if the server reuse a previous session ID, attackers can make users sign in with attackers' session ID and gain a valid authenticated session => session ID should always be fresh

Trusted Computing Group

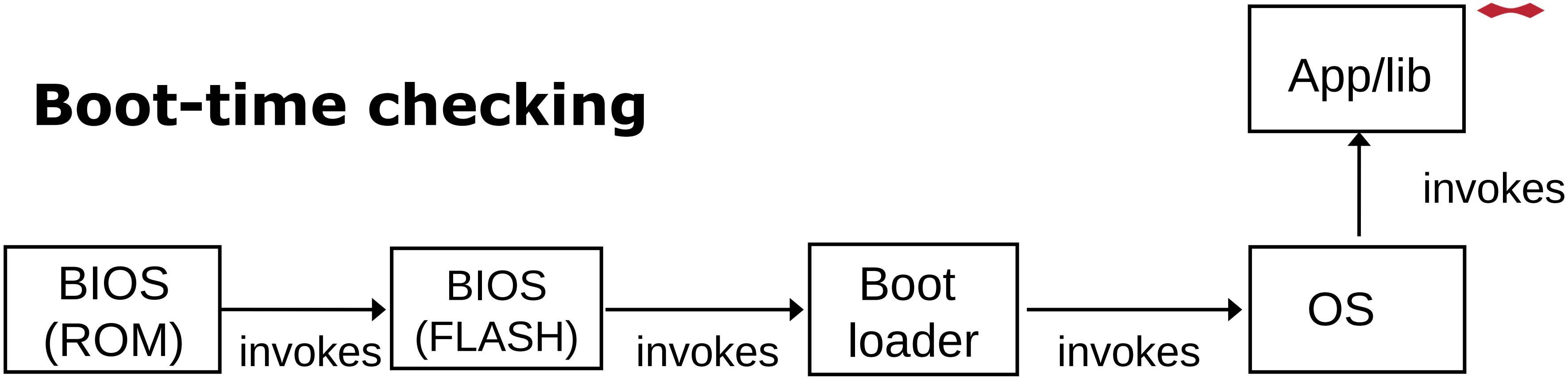
- Trusted Computing Group (TCG) defines the Trusted Platform Module (TPM) standard
  - TPM defines a special processor
    - *Tamper resistant hardware (environment for secure storage and processing)*
    - *Support for cryptographic operations*
- TPM provides the following functionalities
  - Protected capabilities
    - *Commands with exclusive access to shielded locations whose correct operation is necessary*
  - Shielded locations
    - *Domain where it is safe to access sensitive (shielded) data*
- TCG does not control the implementation
  - Vendors are free to differentiate the TPM implementation
  - TPM must still meet the protected capabilities and shielded locations requirements

TPM Architecture

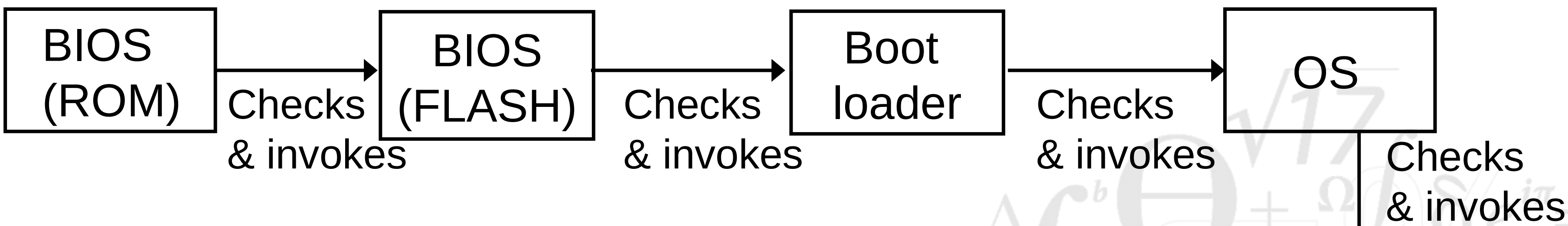


Picture from the Wikipedia

Boot-time checking



A well-defined sequence of software modules get executed at boot time.



Each element in the boot sequence checks the integrity of the next before invoking it

Need to know the verification process succeeded

Trusted boot or secure boot

Platform Configuration Registers (PCRs)

- PCRs are used to securely measure software (by computing hash) during boot (shielded location inside TPM)
- Each PCR can contain a SHA-1 hash value (20bytes)
  - At least 16 PCRs
- PCRs are reset to 0 at boot time
- Write to a PCR # n by extending it – hash extension

TPM_Extend(n,D): $\quad PCR[n] \leftarrow \text{SHA-1}(PCR[n] \,||\, D)$