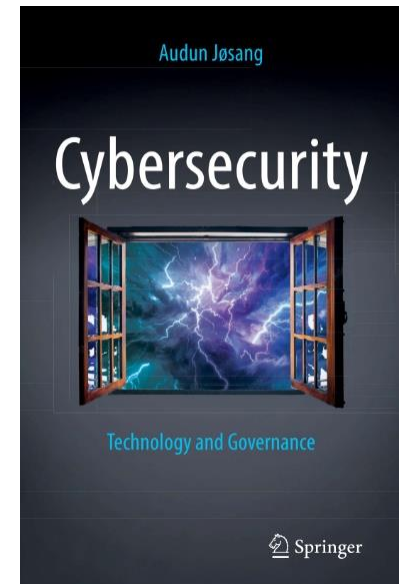


Chapter 8: User Authentication

Cybersecurity: Technology and Governance

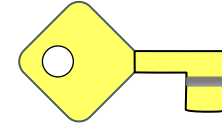
Audun Jøsang,

1st Edition, Springer Nature, 2024.



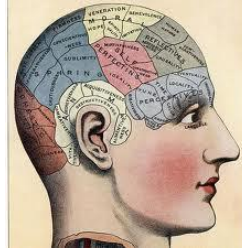
Presentation Overview

- Authenticator categories:



- Something you know

- Passwords



- Something you have

- Devices and apps (e.g. passkeys, tokens)
 - Secondary channels (e.g. SMS and email)



- Something you are/do

- Biometrics

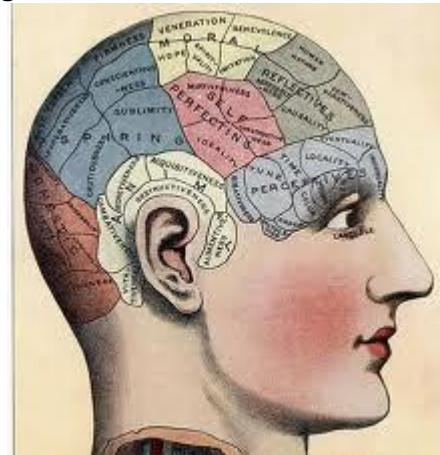


- E-authentication frameworks

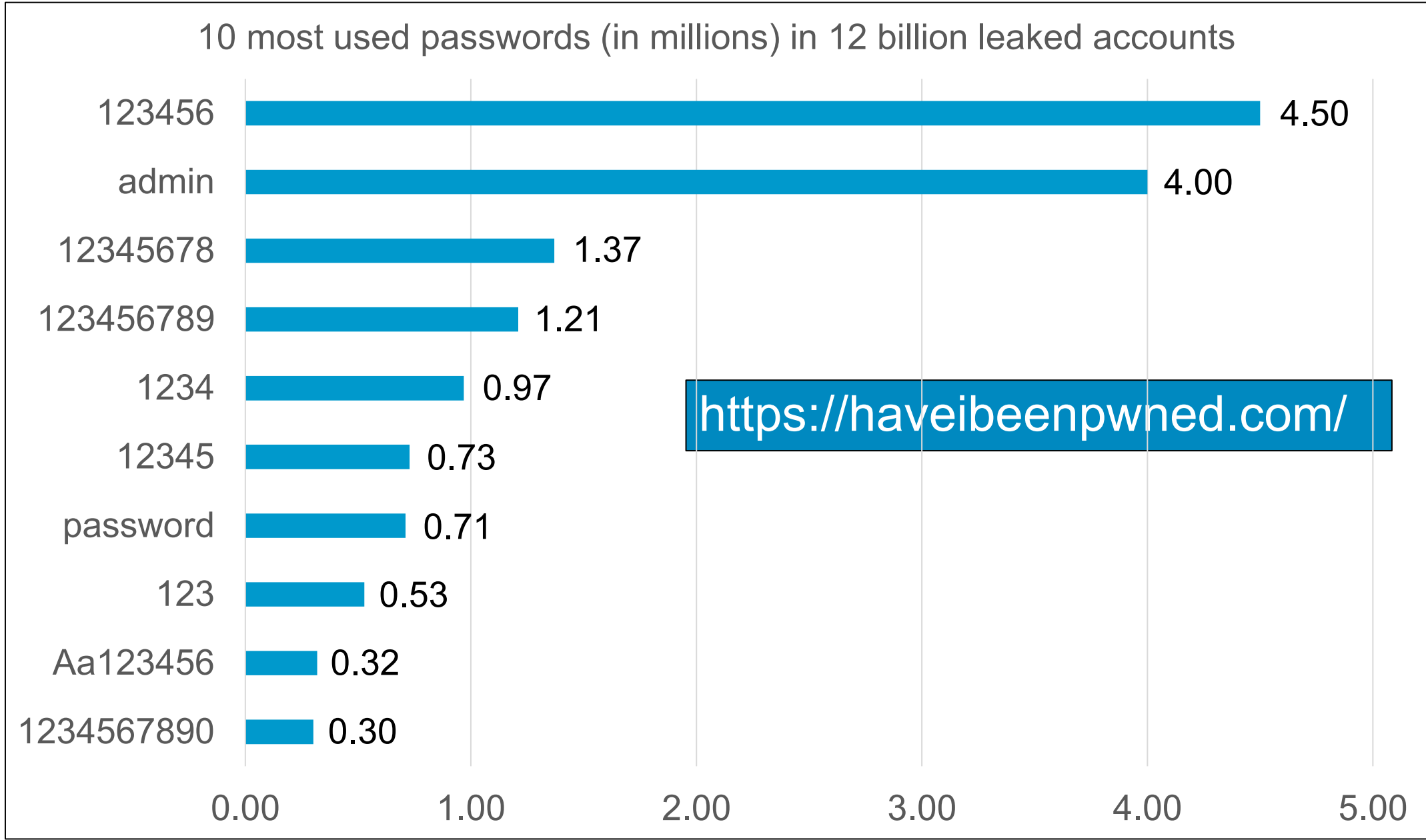


Knowledge-based authenticators

Something you know: Passwords



- Passwords are the simplest and most common authenticator
 - Something the user knows, or keeps in a password manager
- Problems:
 - Easy to share (intentionally or not)
 - Easy to forget (can become a burden for the help desk)
 - Often easy to guess (with powerful cracking tools)
 - Can be written down (which is both good and not good)
 - If written down, then "what you know" is "know where to find it"
 - Often remains in the computer memory and cache, can be stolen by malware



Advice on secure password management

- Password length at least 13 characters
- Use multiple categories of characters
 - Upper/lower case, numbers, special characters
- Do not use common words (from dictionary), name, date, year, or sequential numbers
- Changing passwords
 - Normally rarely (e.g. every 2 years) or never.
 - Immediately upon suspicion of compromise or if account exposed (haveibeenpwned.com/)
- Never reuse passwords
 - or only between accounts with very low sensitivity
- Store passwords securely
 - in your head
 - on paper, if you keep the paper safe
 - in an (online) password manager
 - always encrypted in online digital devices
 - can be stored in plain text in an offline digital device

Storing passwords in systems/networks

- Linux/Unix: `/etc/shadow` –The file stores the password. Previous version saved it in `/etc/passwd`
 - Text
 - Must have root access to read/modify
- Windows: Passwords are stored locally in the Security Account Manager (SAM) database `\\windows\\system32\\config\\sam`
 - NTLM format. Only admin has access.
 - Must have admin access to read/modify
- Storing passwords in network environments
 - AD (Active Directory) in Windows
 - ENTRA (online version of AD in Microsoft Azure)
 - LDAP app (Lightweight Directory Access Protocol) in Linux



Active Directory



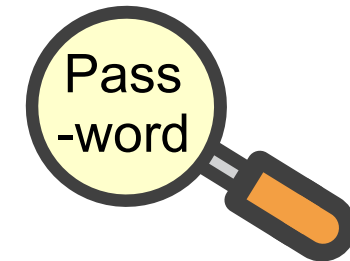
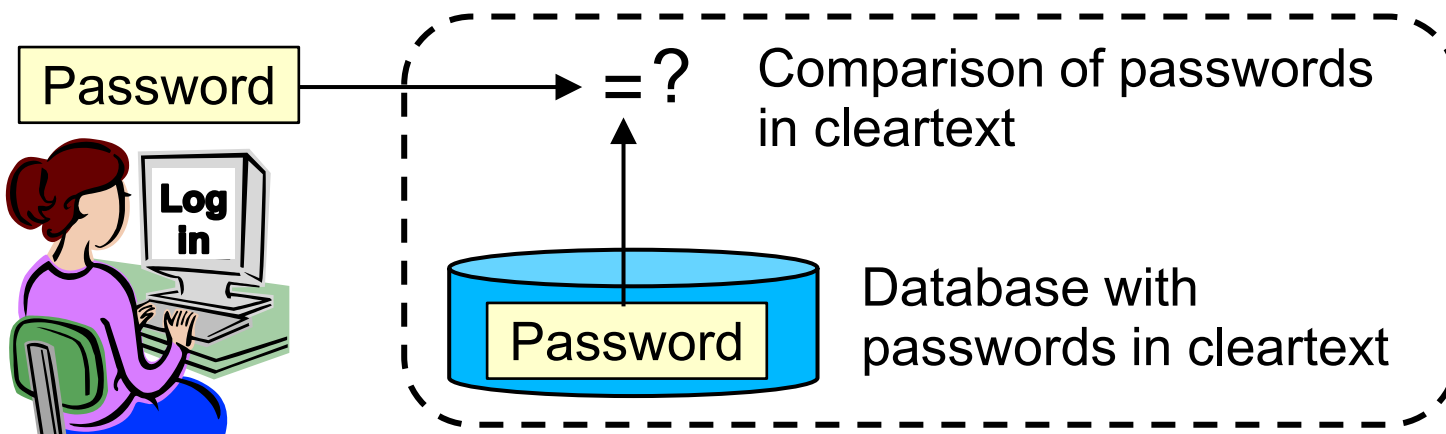
Microsoft
Entra ID



OpenLDAP™
<http://www.OpenLDAP.org>

Database of passwords in clear text: Weak security

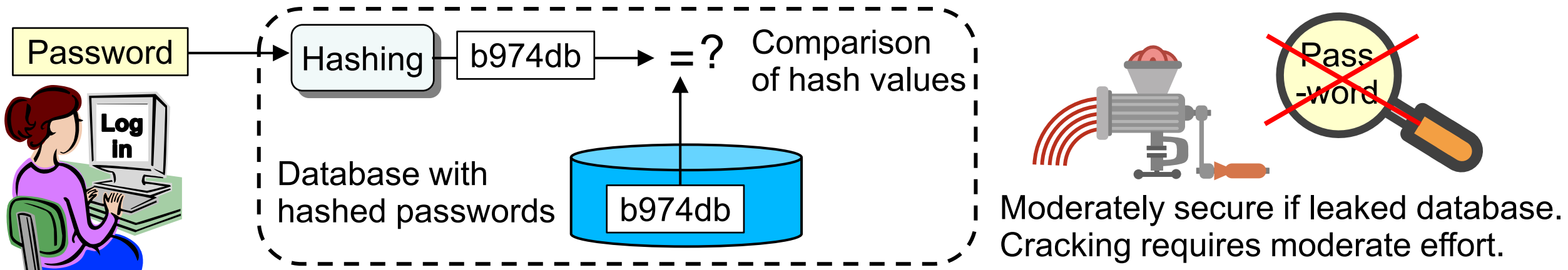
- The Plain Text Password Database is a simple solution for authentication servers, where passwords received from the user are compared to stored passwords.
- But it is also an uncertain solution.
- It is not uncommon for password databases to be stolen/leaked, and if that happens, all passwords are directly accessible



Insecure in case of a leaked database.
All passwords are directly accessible.

Database of hashed passwords: Moderate security

- Database of hashed passwords means that passwords received from users must be hashed before they are compared to the stored hashed password.
- In case of stolen/leaked database, passwords are not directly available in plaintext.
- This provides moderate security, but powerful cracking tools can be used to recover relatively weak passwords in the database.



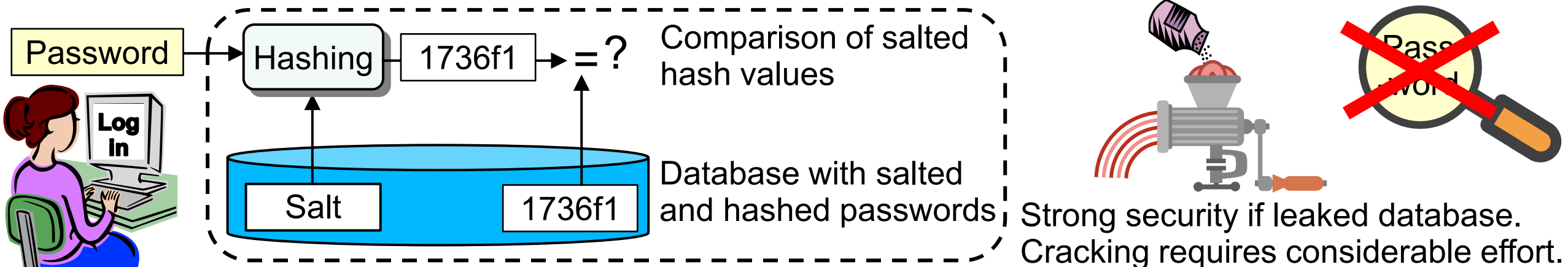
Password cracking of stolen password databases



- Brute Force
 - Try all possible character combinations to find the correct hash
- Intelligent search to find the right hash
 - Username
 - Names of friends/relatives
 - Telephone number
 - Dates of birth
 - Dictionary attack
 - Try all words from a dictionary
 - Precalculated hash tables and rainbow tables

Database of salted hashed passwords: Strong security

- Database of salted hashed passwords means that passwords are appended with a random number (salt) before hashing and storing. The salt is also stored.
- Passwords received from the user must be hashed together with the salt before it is compared to the stored salted hashed password.
- In case of stolen/leaked database, heavy computing is required to crack passwords.
- Provides strong security, and is considered standard practice for password databases.



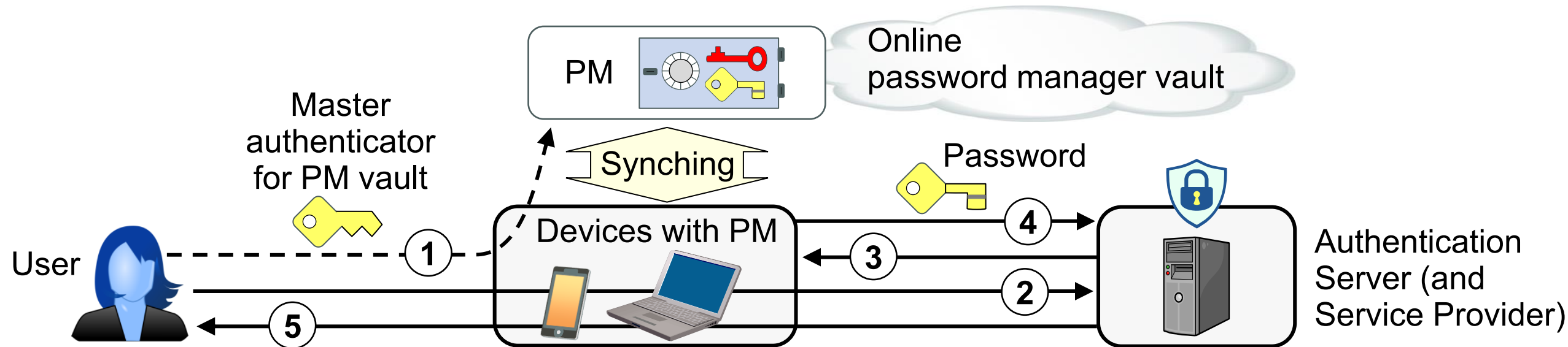
Password salting: Defense against password cracking



- Append random data (salt) to the user's password before hashing it
 - In Linux: a randomly selected integer from 0 to 4095.
 - Different salt for each user
 - Different salt yields different hashes for similar passwords
- Prevents users with identical passwords from having the same password hash value stored in the database
- Makes it virtually impossible for attackers to use hash tables and rainbow tables
 - Attackers would have to generate a hash table for each possible salt value
 - Passwords over 8 characters give table size greater than Petabytes

Password Managers

- **Built into the OS.** For instance, on Apple devices, Keychain is a built-in one, so users don't need to download or install any separate app.
- **Built into browser.** Firefox and Chrome have built-in password banks that allow users to save passwords for various websites, generating strong passwords.
- **Third-party password managers.** Dedicated software combined with online storage, which requires users to install an app or a browser extension.

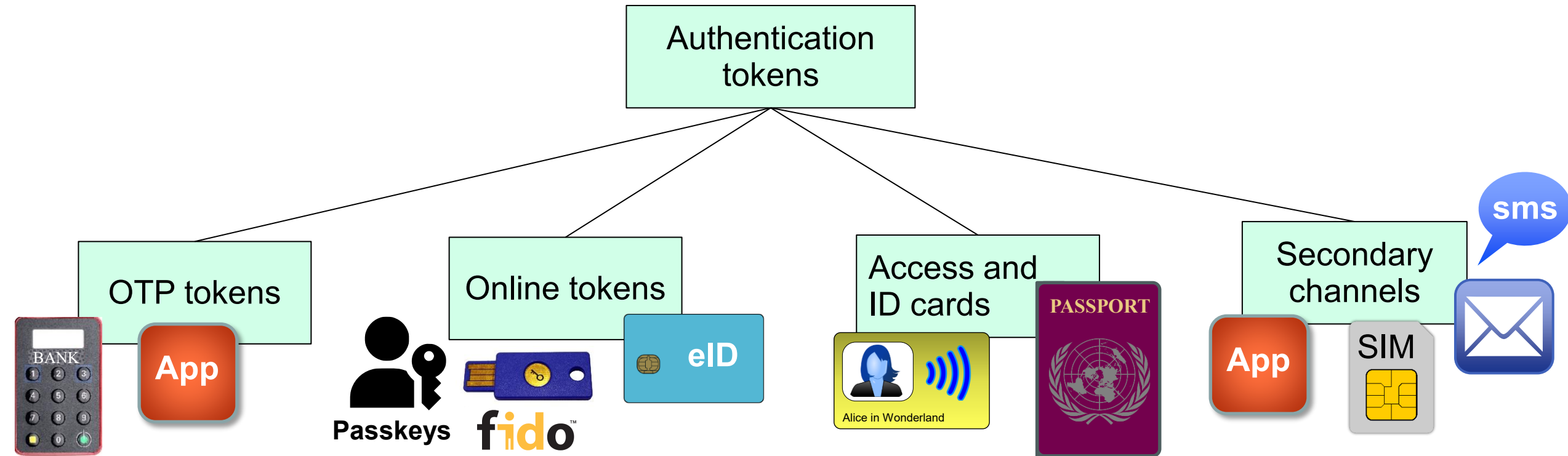


Ownership-based authentication

Something you have: Devices



Types of authentication devices



OTP tokens (One-Time-Password):



- The token contains a clock
- Displays time-dependent code (one-time code) on the screen
 - User copies code to terminal to log in
- Each code is valid for a specific time window
- Codes from close neighbor time windows are accepted
- The clocks must always be synchronized
 - Needs battery power
- Example: RSA SecureID

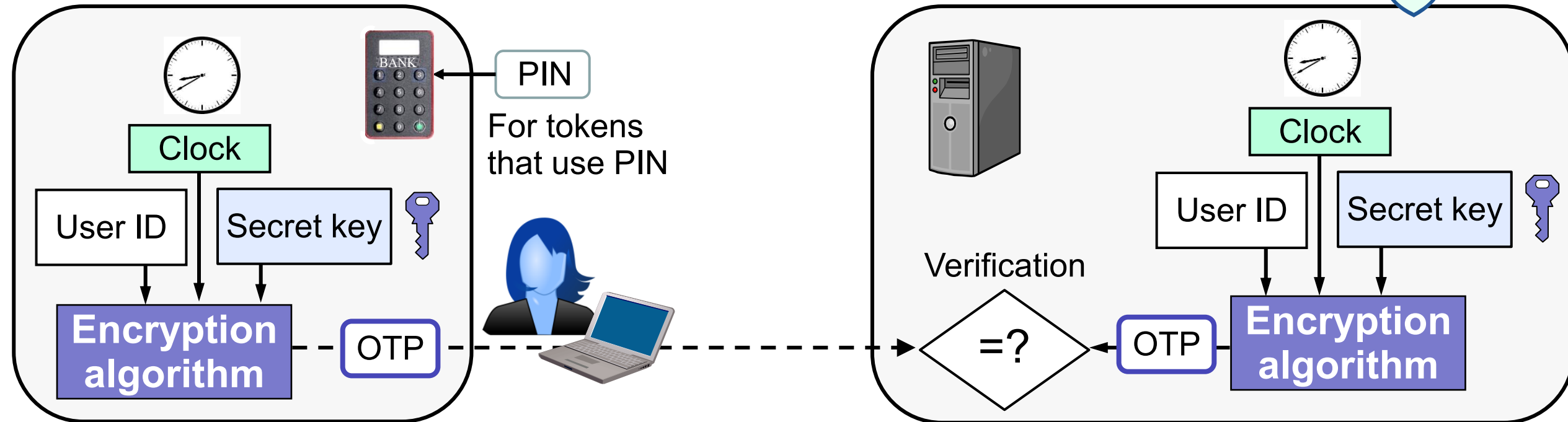
Synchronous OTP chip, with clock (and PIN)

e.g. RSA SecureID



User's token

Authentication server



Time-based OTP code tokens:



SafeID OTP token
with PIN



ActiveID OTP token
with PIN



OTP token with PIN



Feitan OTP token
without PIN



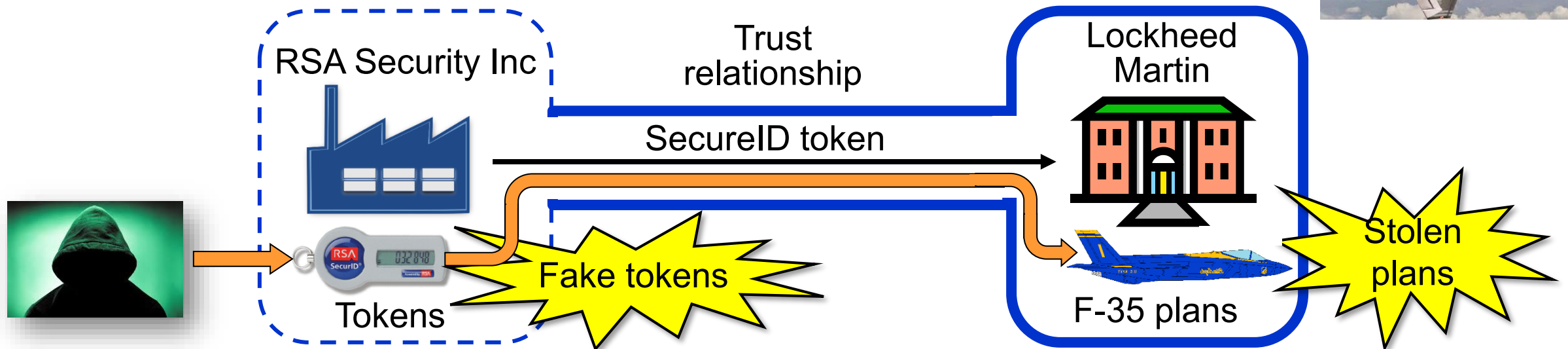
RSA SecurID
without PIN



OTP token without PIN

Supply chain attack: Hacking OTP token

- RSA Security Inc. was hacked in 2007, the hackers stole secret info to design SecureID chips
- The hackers created fake SecureID chips for ID theft
- Businesses using RSA SecureID were vulnerable
- Lockheed Martin used RSA SecureID
- Chinese attackers accessed Lockheed Martin employee accounts
 - Stole plans for F-35 fighter jets



Phishing-resistant authentication



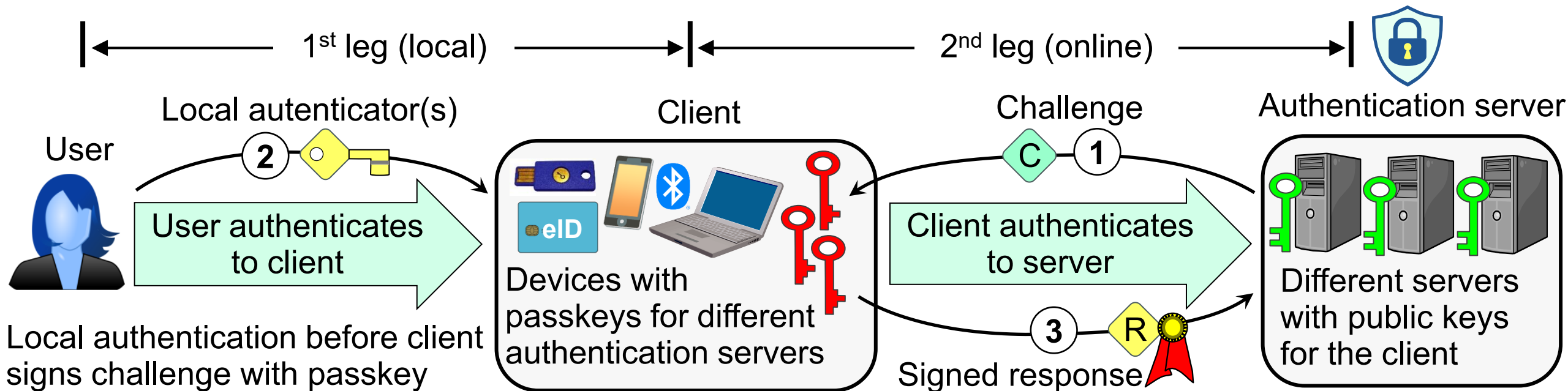
- Phishing attacks exploit users' failure to recognize fake websites.
 - ID spoofing happens if fake websites get hold of authenticators
- Phishing resistance prevents authenticator(s) from being exploited by fake websites.
- Domain name integrated into authenticators, and client checks correct domain name
 - Domain names are integrated as part of private passkeys (authenticators) stored on the client
 - Client blocks authentication to a fake website when the domain name is not part of the passkey
 - Passkeys create complexity for account recovery and synchronization between devices
- Authentication schemes that are not phishing-resistant require the user to understand domain names, know and check domain names of websites
 - Many users do not understand domain names, and/or ignore domain names for real websites
 - Domain names can be misleading, leaving even experts confused
- Common authentication tokens and apps are not phishing-resistant.
 - Requires the user to understand domain names and not be scammed by fake websites.
- People ignore how extremely important it is to understand domain names.

Authentication with Passkeys, FIDO, and WebAuthn



Passkeys

- Client/token authenticates itself on the user's behalf to an authentication server.
- Passkeys can be stored:
 - embedded in client (smartphone or laptop), in browser, OS or hardware
 - in a token (USB or smart card) connected to the client platform physically, or via NF/Bluetooth
 - online in a password manager/wallet
- The client authenticates the user locally before making online authentication.
- Phishing-resistant, because passkeys contain domain name of authentication server.



Passkey authentication



- A passkey is a cryptographic private key.
- A passkey is in many ways like a password.
- Password banks support passkeys.
- The user client/chip stores separate passkeys for different authentication servers.
- Passkeys will probably most often be stored embedded in the client, rather than in a separate chip.
- Each passkey contains the domain name for a specific authentication server.
- The client requires the authentication server's domain name to match the passkey.
- The user cannot override the client to sign the challenge from fake websites.
- Mobile platform can support passkey authentication via laptop with Bluetooth connection.

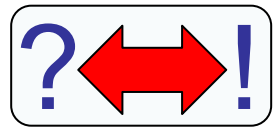


Benefits and Challenges of Passkeys/FIDO/WebAuthn

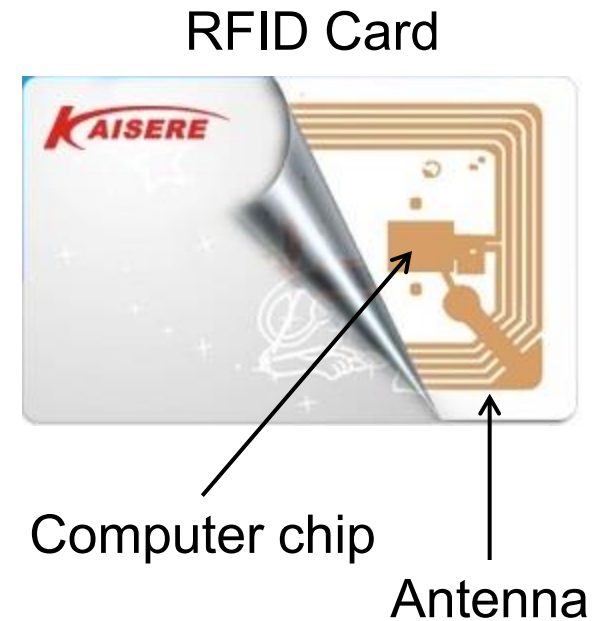


- Benefits
 - Passkeys/FIDO/WebAuthn provides phishing-resistant authentication
 - Private keys stored in platform/device are specific to the IdP's domain name, which means that the platform/device cannot find the passkey for fake websites.
 - The user does not have to deal with passwords
- Challenges
 - It can be technically complicated to synchronize (transfer) passkeys to a new platform/device when replacing or losing a platform/device.
 - Passkey synchronization solutions are based on cloud storage, such as:
 - Own password manager
 - Passkey manager controlled by the manufacturer for platform/device
 - Ultimately, accessing passkeys in the cloud requires traditional authentication with passwords or other authenticators, which is not phishing-resistant.
 - Passkey schemes are difficult to understand for ordinary users.

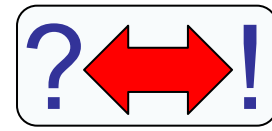
Access cards, ID cards and passports



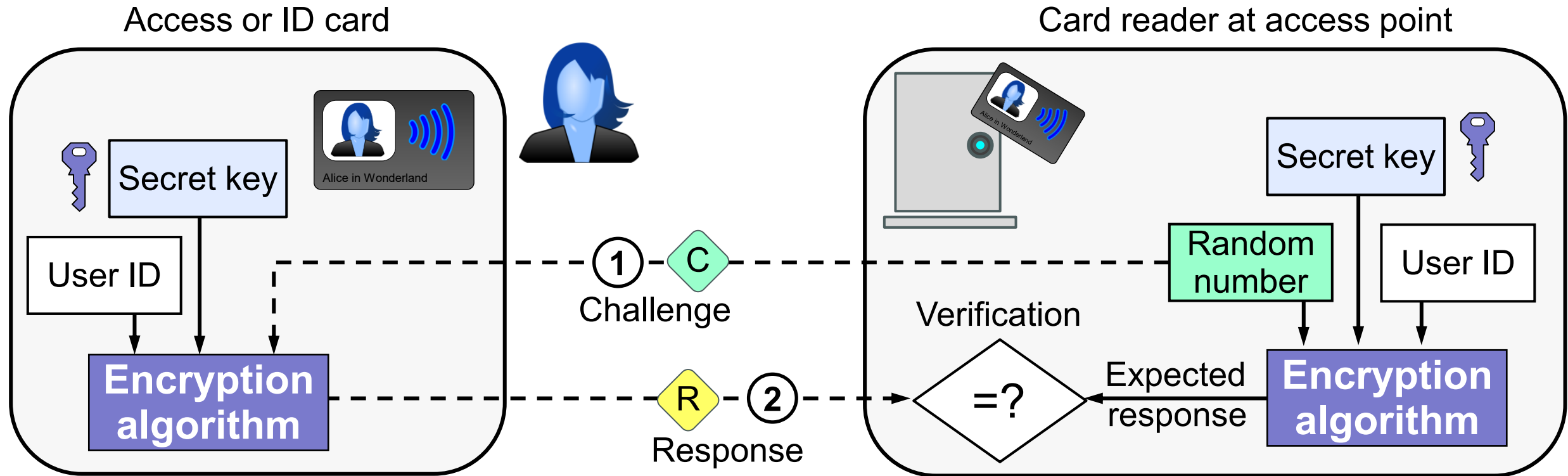
- Contactless cards, also called RFID (Radio Frequency Id), consist of a computer chip and an antenna.
 - No need for physical contact with reader
 - communicate via radio signals
 - Power generated by magnetic field from reader
 - When out of the reader's reach, the card gets no power and remains idlebattery-powered RFID tags are also used
- Suitable for use in hot, cold, dirty and damp environments



Access cards, ID cards and passports

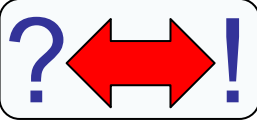


- Example access card with symmetric key
- National ID card and passport use asymmetric key

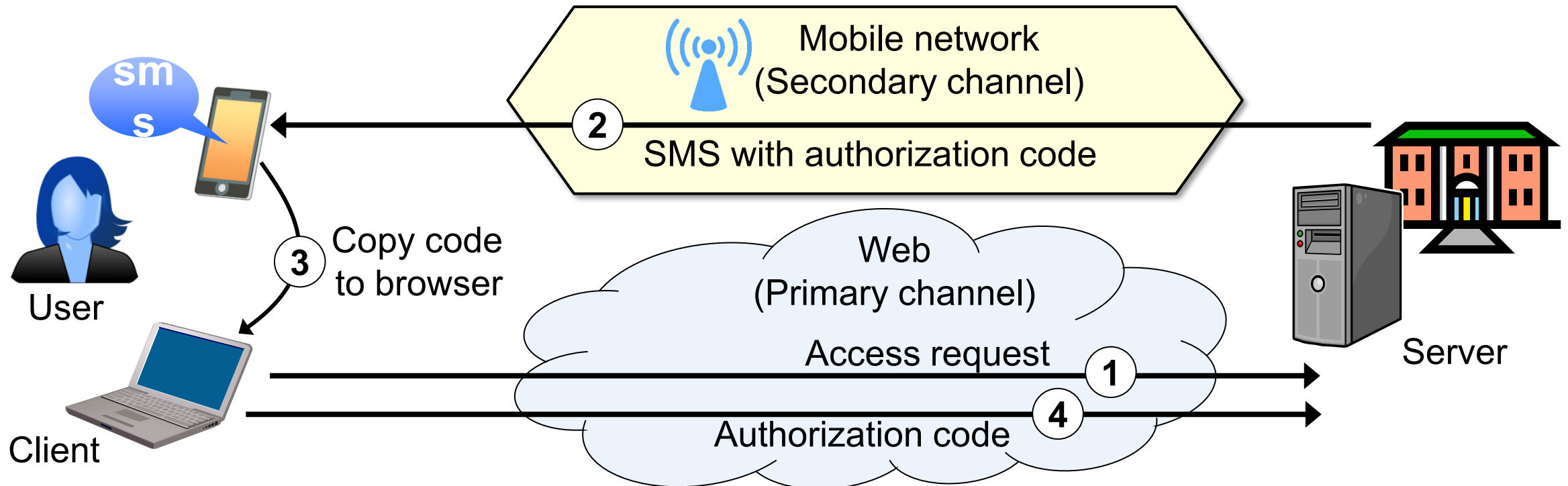


Secondary channel authentication

- If web communication is considered a primary channel, authentication can take place via other secondary channels:
- Example secondary are, for example:
 - SMS
 - App
 - email
 - SIM
- Some secondary channels are more secure than others.
 - SMS is considered relatively weak, because SMS can be stolen relatively easily
 - Apps use security function on a mobile phone as a secondary channel



Secondary channel authentication



Inherence-based authentication

Biometrics



Something you are

Something you do

Biometers intro



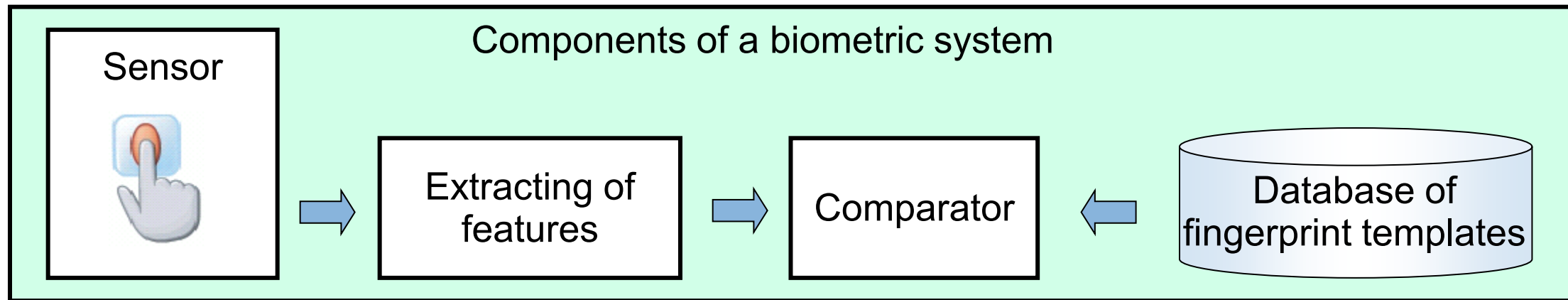
- What is it?
 - Automated methods of verifying or recognizing a person based on physiological characteristics.
- Biometric modalities, examples:
 - fingerprint
 - Facial recognition
 - eye retina/iris scan
 - Hand geometry
 - Hand signature
 - Voice/speech characteristics
 - Keystroke dynamics

Biometric modality requirements



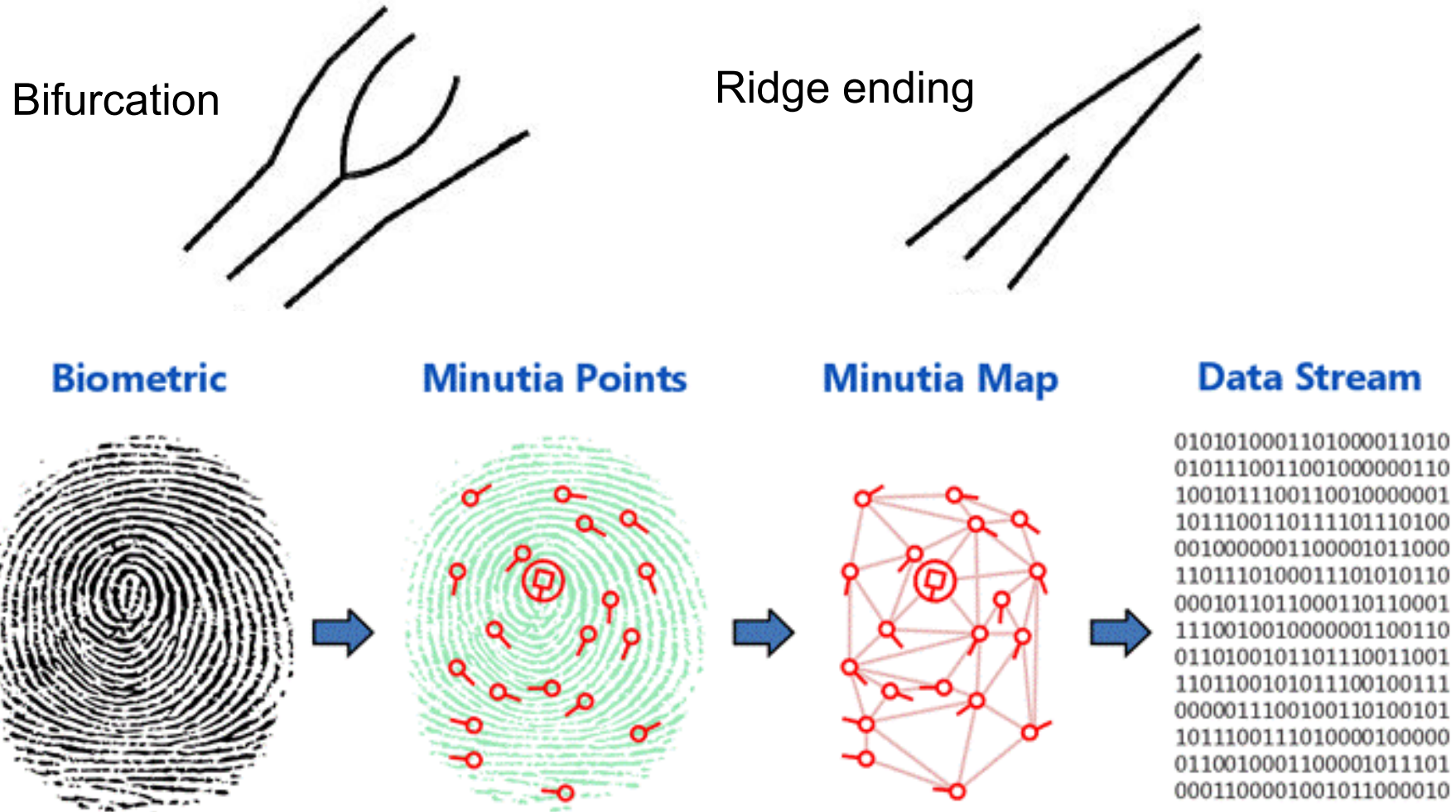
- **Universality:**
 - Each person should have the modality
- **Distinctiveness:**
 - Different people should be sufficiently different with regard to the characteristics of the modality;
- **Permanence:**
 - The characteristic should remain unchanged (with respect to criteria for comparison) over time;
- **Measurability:**
 - The characteristic should be easy to obtain and measure in a quantitative manner.
- **Accuracy:**
 - Accuracy of a biometric system, expressed as EER (Equal Error Rate).
- **Performance:**
 - the speed of the analysis, the resources required to achieve the desired speed;
- **Acceptance:**
 - the extent to which people are willing to accept the use of a particular biometric modality
- **Protection from presentation attacks:**
 - Difficulty of fooling the biometric system

Biometrics: System components



Extracting features

Example: Extract minutia from fingerprints

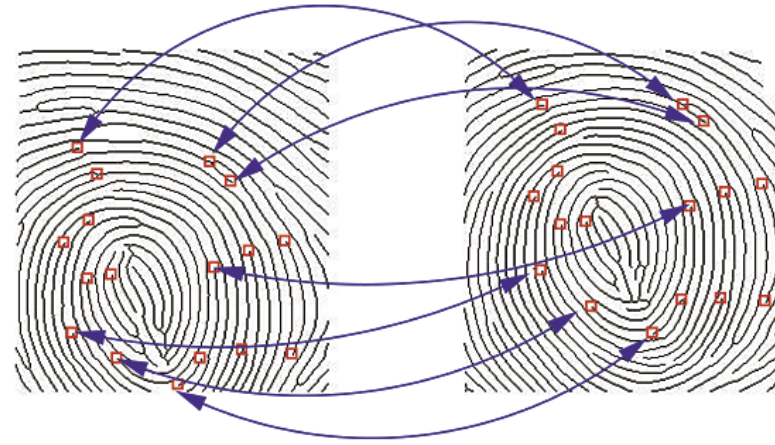


Biometrics: Operation modes



- **Enrolment:**
 - Analogue collection of users' biometric characteristics.
 - Processing of collected sample to generate a template of users' characteristics that is stored for later use.
 - database of biometric templates
- **Identification (1:N, one-to-many)**
 - collection of a new biometric sample from the candidate-person(s)
 - Search the database of saved templates for a match based solely on collected sample(s).
- **Authentication (1:1, one-to-one):**
 - Obtaining a new biometric sample from the user for authentication
 - Comparison of new trial with the user's saved template

Comparison of samples:



- Characteristics from collected samples are compared with characteristics from stored template
- Often based on classification with machine learning
- The comparison gives a score of S
 - The greater the equality, the higher the score S
- Threshold T determines whether the test is a match
 - *match (presumed correct person) is when $S \geq T$*
 - *non-match (presumed wrong person) is when $S < T$*

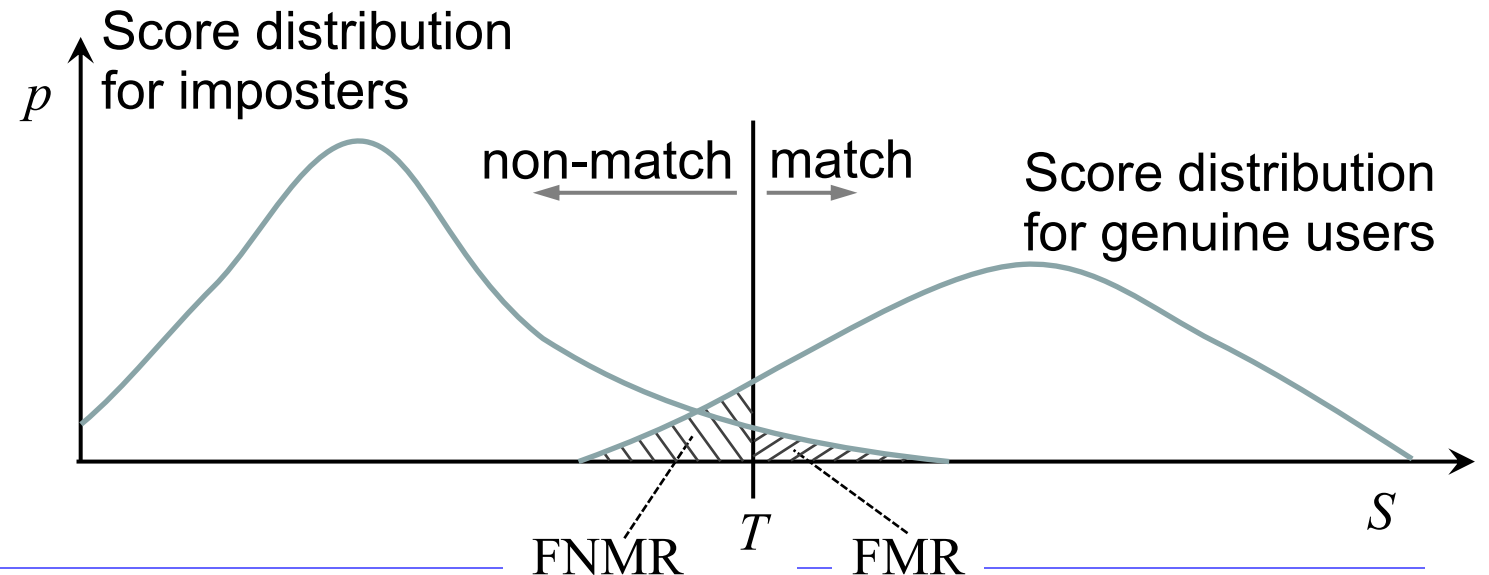
Comparison of samples



- True positive
 - *Match of genuine sample → genuine user is accepted*
- True negative
 - *Non-match of imposter sample → imposter is rejected*
- False positive
 - *Match of imposter sample → imposter is accepted*
- False negative
 - *Non-match of genuine sample → genuine user is rejected*
- False-match-rate and False-Non-match-rate
 - $\text{FMR} = (\# \text{ match of imposter samples}) / (\text{total } \# \text{ imposter samples})$
 - $\text{FNMR} = (\# \text{ non-match genuine samples}) / (\text{total } \# \text{ genuine samples})$
- Threshold T determines balance between FMR and FNMR

Accuracy: Distinguish between genuine users and imposters

- Plot the distributions of scores S from a sample of genuine and non-genuine users
- Threshold T determines relative magnitude of FMR and FNMR
 - Low T provides large FMR, thus low security, but small FNMR and thus good usability
 - High T gives small FMR, thus high security, but large FNMR and thus poor usability
- «Rate» (equivalent to percentage) is the relative size of the shaded areas
- When T is adjusted so that $FMR = FNMR$, the rate is called EER (Equal Error Rate)
- Small EER for high accuracy



- Small overlap between curves results in small EER
- All biometric systems have more or less overlapping curves

Biometrics Spoofing: Presentation Attacks

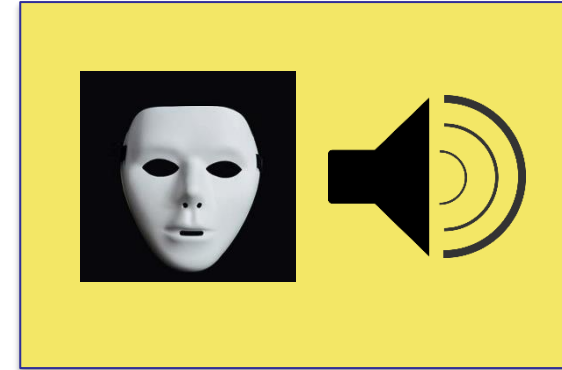
- Relatively easy to fool a biometric system
 - Terminology: Presentation attacks



Fake finger



Fake face



Deepfake audio/image/video

- Biometric authentication on smartphones is insecure
- A lot of research is being done on PAD (Presentation Attack Detection) to make biometrics more secure
- An alternative is to check the environment for obtaining biometric samples

Safety of using biometrics



- It may be appropriate to set requirements for personal safety when using biometric systems, as the user is physically involved.
- In USA in 2023, it was reported that victims have been drugged to unlock their smartphones with facial recognition to access their bank accounts.
- In all cases, these are body-based biometric modalities, where a body part, such as a face or finger, is used for authentication.
- Behavior-based biometrics, such as voice recognition and keyboard dynamics, require the user to be conscious, so it's difficult to force a biometric sample.

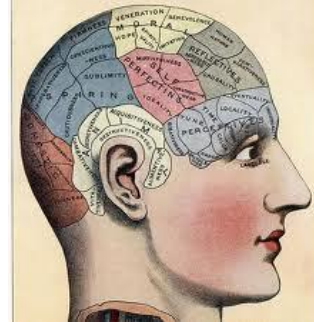


Car thieves chopped off part of the driver's left index finger to start S-Class Mercedes Benz equipped with fingerprint key.

Malaysia, March 2005

(NST picture by Mohd Said Samad)

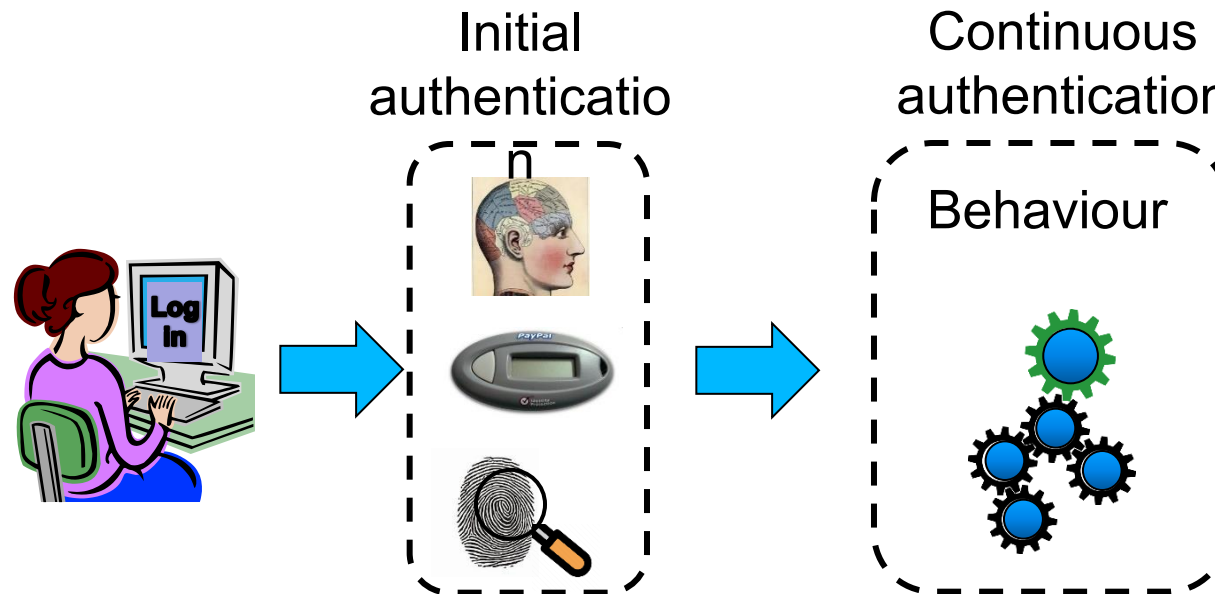
Multi-factor authentication



- Multi-factor authentication combines two or more techniques to provide stronger authentication.
- Two-factor authentication is often based on something the user knows (1st factor) plus something the user has (2nd factor).
- Examples
 - password and token
 - password and secondary channel (e.g. SMS)

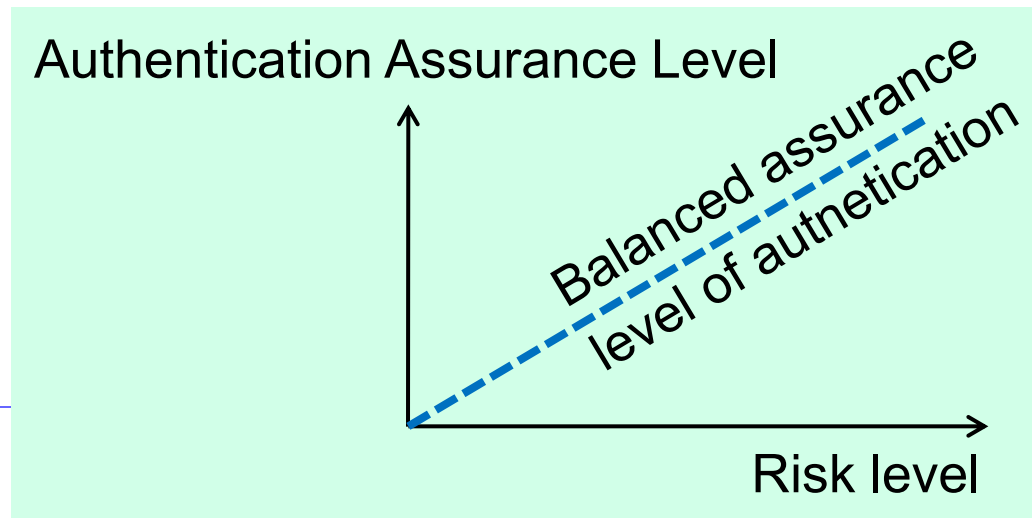
Continuous authentication

- Continuous authentication is comparing actual behavior with expected behavior during interaction with the system.
- If the similarity falls below a given threshold value, the user will be logged out, or alternatively may have their authorization reduced.
- Regular authentication first, then continuous authentication



Authentication Assurance Level

- Assurance level of authentication = robustness of authentication
- Services with different sensitivity and risk levels set different requirements for authentication
 - Serious consequences and risks in the event of authentication failures require strong authentication with a high level of guarantee
- Strong authentication costs more than weak authentication
 - Unnecessarily strong authentication is a waste of money
- The assurance level for authentication should therefore be balanced



E-authentication framework

- Trust in identity is a requirement for secure e-government
- Strong authentication provides confidence in identity.
- Authentication depends on technology, policy, standards, practices, awareness, and regulation.
- Consistent authentication frameworks provide a basis for the effectiveness of e-government.



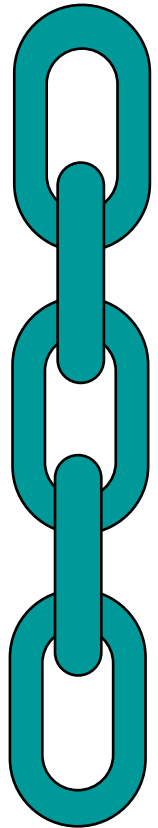
E-authentication framework, with different levels of assurance

Frameworks for e-Authentication		Authentication assurance levels			
SP 800-63-4 Digital Identity Guidelines NIST, USA 2024			Assurance Level 1	Assurance Level 2	Assurance Level 3
eIDAS-2 Electronic identification and trust services EU 2024			Low	Substantial	High
e-Pramaan: Framework for e-Authentication India, 2014			1FA	2FA	MFA
IS 29115 Entity authentication assurance framework ISO/IEC 2013		Low (1)	Medium (2)	High (3)	Very High (4)

- "Level of Assurance" (LoA), or "Authentication Assurance Level (AAL).

Categories of authentication requirements

The assurance level of authentication is a function of 3 categories of requirements



a) ID proofing and provisioning

Methods for identity proofing, and for provisioning of authenticators

Requirements for identity proofing and correct registration:

- birth certificate, passport, biometrics

Requirements for provisioning of authenticators:

- delivery and recovery by mail, physical or online

b) Authentication method

Technology and combination of methods for user authentication.

Requirements for authentication methods:

- password length and quality
- cryptographic strength and PKI
- multi-factor authentication
- phishing resistance

c) Federation security

Trust levels and security in the interaction between IdP and SP.

Requirements for certification of IdP (authentication servers) and ID portals, and legal agreements between stakeholders.

eIDAS

electronic IDentification, Authentication and trust Services

- eIDAS is the EU's regulation on e-authentication and trust services for e-government.
- "Trust service" is EU jargon for PKI certification services.
- eIDAS specifies three Level of Assurance (LoA).

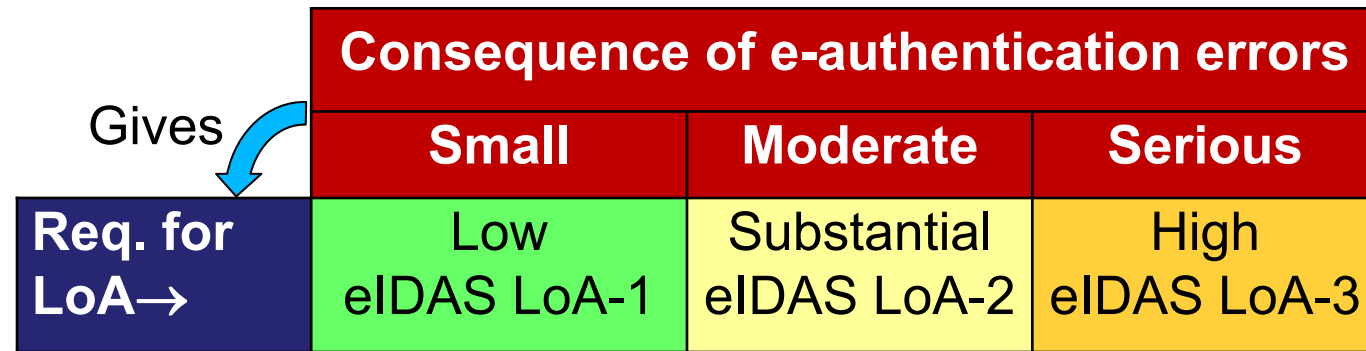
Low Assurance eIDAS LoA-1	Substantial Assurance eIDAS LoA-2	High Assurance eIDAS LoA-3
Limited degree of confidence in the claimed or asserted identity of a person	Substantial degree of confidence in the claimed or asserted identity of a person	High degree of confidence in the claimed or asserted identity of a person



EU symbol for qualified trust services

Adequate assurance level of authentication based on risk

Table to determine the adequate level of authentication guarantee for an eGovernment service (eIDAS)



Consequence of e-authentication errors		
Small	Moderate	Serious
Low eIDAS LoA-1	Substantial eIDAS LoA-2	High eIDAS LoA-3

Example risk matrix for eIDAS

E-authentication error means that an attacker is able to spoof and steal other people's identities

End of presentation

