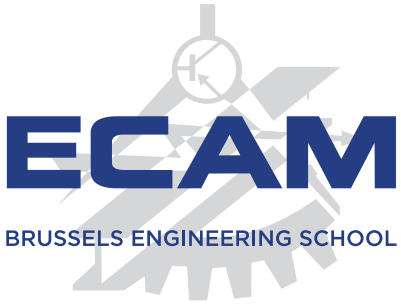


I5020 Computer Security

Session 7

Authentication and Access Control

Sébastien Combéfis

Fall 2019

User Authentication

User Authentication (1)

- Fundamental block of computer security is user authentication
  - Primary line of defence in computer security contexts
  - Basis for access control and user accountability
- Process of verifying identity claimed by or for a system entity

  Checking that the identity of a user is authentic
- Authentication process consists in two steps
  1. **Identification**: presenting an identifier to the security system
  2. **Verification**: proving relation between entity and identifier

(User, Password) Pair

- User identifiers stored on the system/server that is used

  Could be known by administrators and other users

- Association of an item of authentication information
  - One such item associated to each user identifier
  - Could be a secret password chosen by the user, for example
  - Only known by the user and the system

- Managing access permissions and activity audit

  Only if no one is able to guess the password

User Authentication (2)

- User provides a claimed identity to the system by identification

  User authentication establishes validity of the claim

- User authentication different from message authentication
  - Verifying that content of message not altered...
  - ...and that the source of the message is authentic

- Many different means of user authentication can be used

  Passwords, smart cards, or biometric information

Electronic User Authentication

- Establish confidence in user identities presented electronically

  An authenticated identity is then available to the system

- System manages what an authenticated individual can perform

  Controlling database transactions, system resources access, etc.

- Authentication and authorised functions on several places

  Typically across an open network, such as the internet

Electronic User Authentication Model

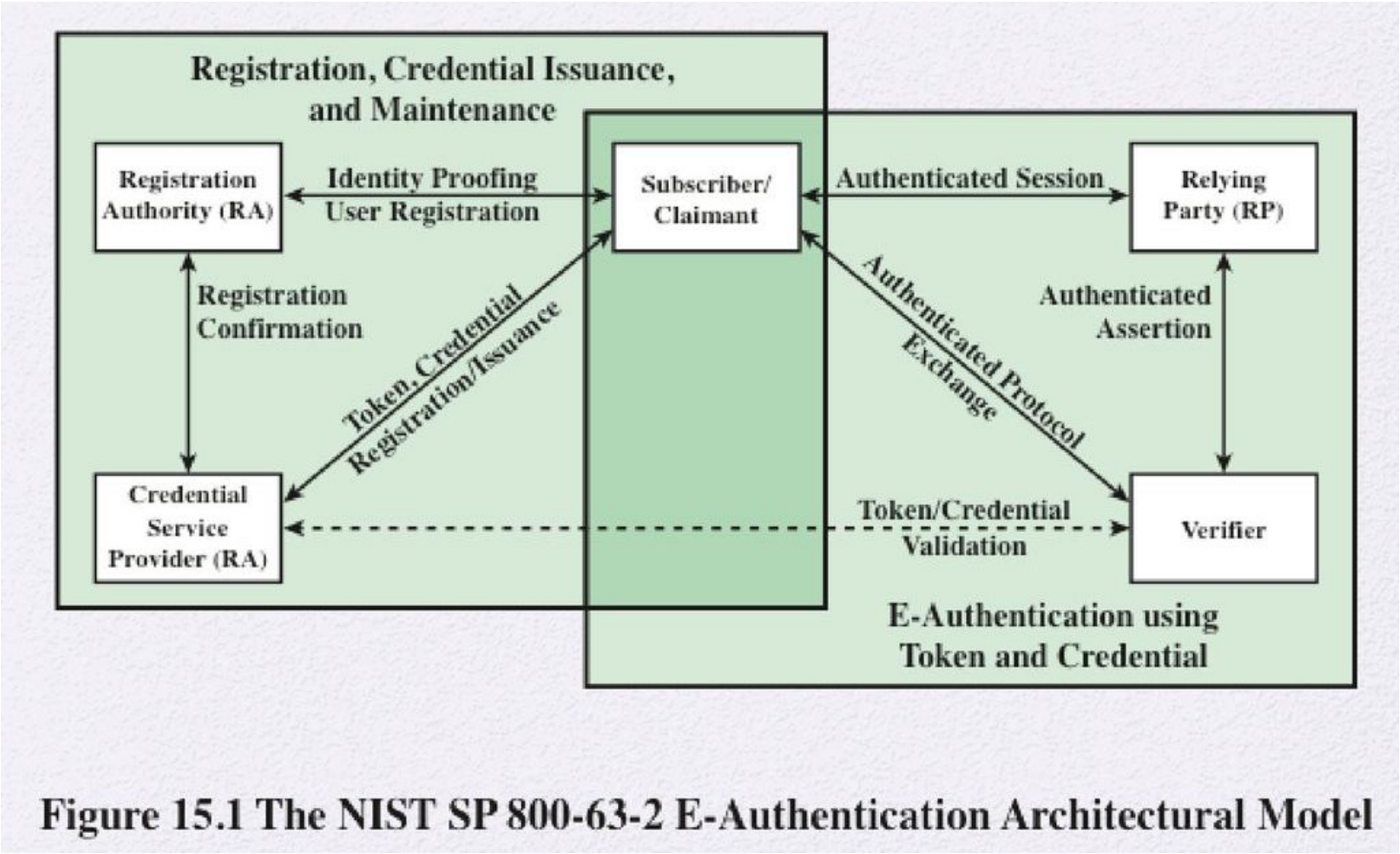


Figure 15.1 The NIST SP 800-63-2 E-Authentication Architectural Model

User Authentication Mean (1)

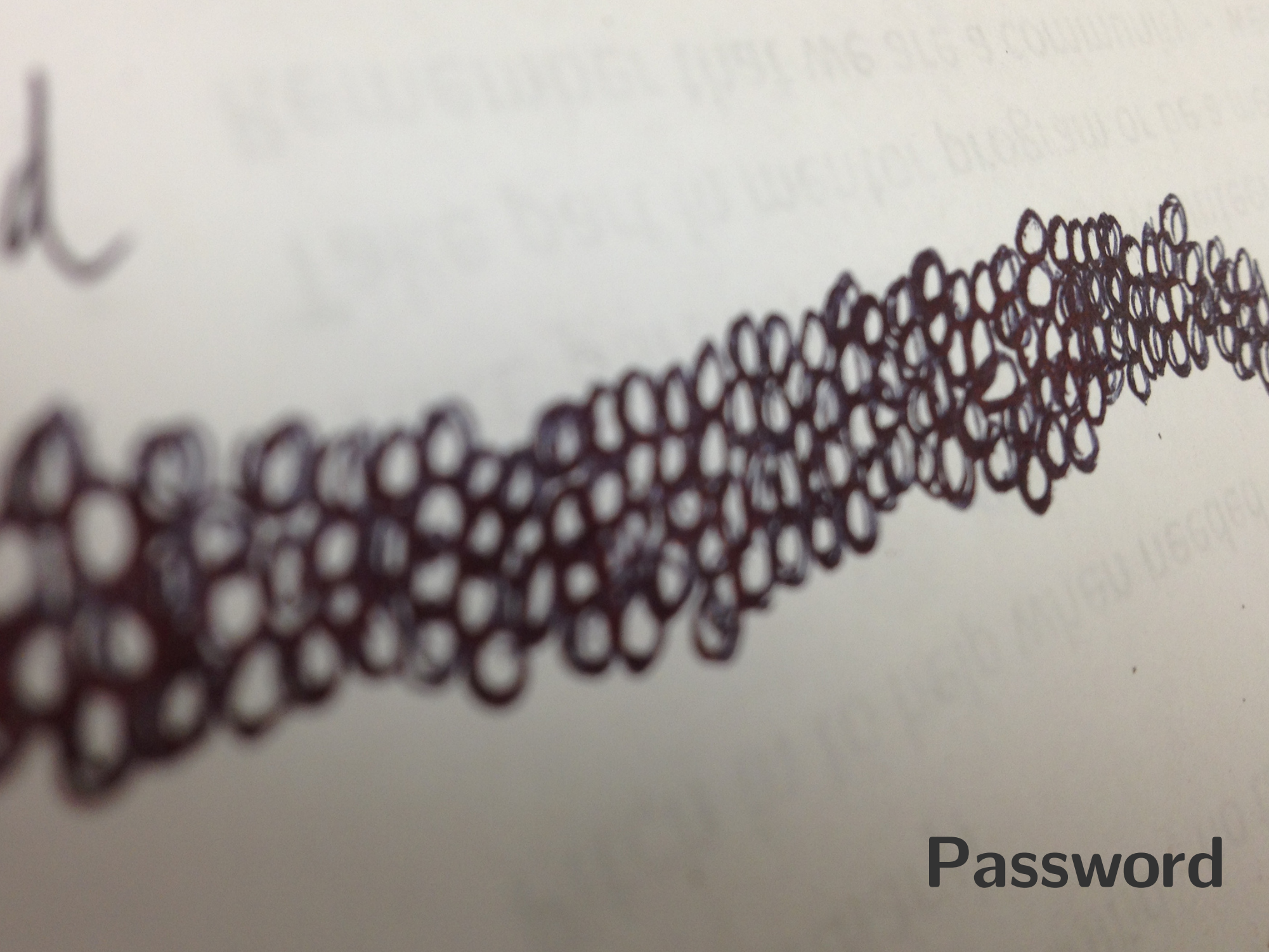
- Four general means to authenticate a user's identity
  - **Knowledge**: *password, PIN, answer to prearranged questions*
  - **Possession** (token): *keycard, smart card, physical key*
  - **Attribute** (static biometrics): *fingerprint, retina, face*
  - **Attribute** (dynamic biometrics): *voice pattern, handwriting characteristics, typing rhythm*
- Authentication means can be used alone or in combination

  Should be properly implemented and used for good security

User Authentication Mean (2)

- All user authentication means do have problems
  - A password can be stolen or guessed, or forgotten
  - A token can also be stolen or can be forged, or lost
  - Biometric authenticators suffer from false positives/negatives...
  - ...and from user acceptance, cost and convenience
- Choosing the right combination compromise

  Depending on the security level that is expected


Password

Password-Based Authentication

- Authentication with a username and the associated password

  Password compared with the one stored in the system

- Different kinds of associations for passwords
  - One password for each object to protect
  - Protecting a set of access rights with the same password
- One password should only be used for one access

  Not a good practice to use the same password for the same user

Identifier

- User identifier provides security in several ways, determining...
  - ...whether user authorised to gain access to system
  - ...privileges accorded to the user (admin, superuser, etc.)
  - ...access in a discretionary access control mode
- Identifier searched in the system database

  Before comparing password with the one stored in the system

Password Vulnerability (1)

- Using passwords is not secure at all
  - Could be easy to guess, can be exposed, sniffed, etc.
  - Can be illegally transfered to an unauthorised user!
- Only 10,000 possibilities with a four-digit pin code

  Only 5,000 attempts on average (only 5s if one test/ms)
- A password can be seen while it is exposed

  Shoulder surfing, network sniffing, keylogger, etc.

Password Vulnerability (2)

- Passwords are typically stored in an hashed form

  Theoretically impossible to reverse the function

- Several kinds of attacks against passwords and strategies
  - Offline dictionary attacks
  - Specific account and popular password attacks
  - Password guessing against a single user
  - Waiting for a user to log-in by workstation hijacking
  - Exploiting user mistakes such as password writing, sharing
  - Exploiting multiple use of the same password for a single user
  - Electronic monitoring by eavesdropping the network

Password Vulnerability (2)

- Several countermeasures can be deployed to secure a system
  - Protecting password file, setting up intrusion detection
  - Account lock mechanism, policy avoiding common passwords
  - Password policy for secrecy, length, character set, lifetime
  - Changing preconfigured and default passwords
  - Educating users so that they protect their passwords
  - Policy to avoid same password on different devices/websites
- Passwords are still used and that will not change soon

  Despite the many vulnerabilities they are subject to

Password Alternative

- Client-size hardware such as fingerprint/smart card reader

  Need for appropriate software to exploit this hardware

- User authentication with physical tokens

  Pretty expensive or inconvenient to carry around

- Rely on a single sign-on to access multiple services

  Dangerous as it creates a single point of security risk

- Automated password manager to remember and type them

  Poor support for roaming and synchronisation

Securing Password

- Passwords should be secretly stored in the system

  But it must be possible to check whether a password is correct

- The storage should be protected against theft

  Use disk encryption, physical protection, backups, etc.

- Passwords should not be stored in clear in the database

  Typically storing them in a secured hashed form

Brute-Force Attack

- Brute-force attack tries all the possible passwords

  The passwords space should be large enough

- Using botnets to make the attack legitimate

  Simulate multiple users trying to access the resource

- Require an access to the system and the possibility to connect

  Not always possible to do so, in particular for remote access

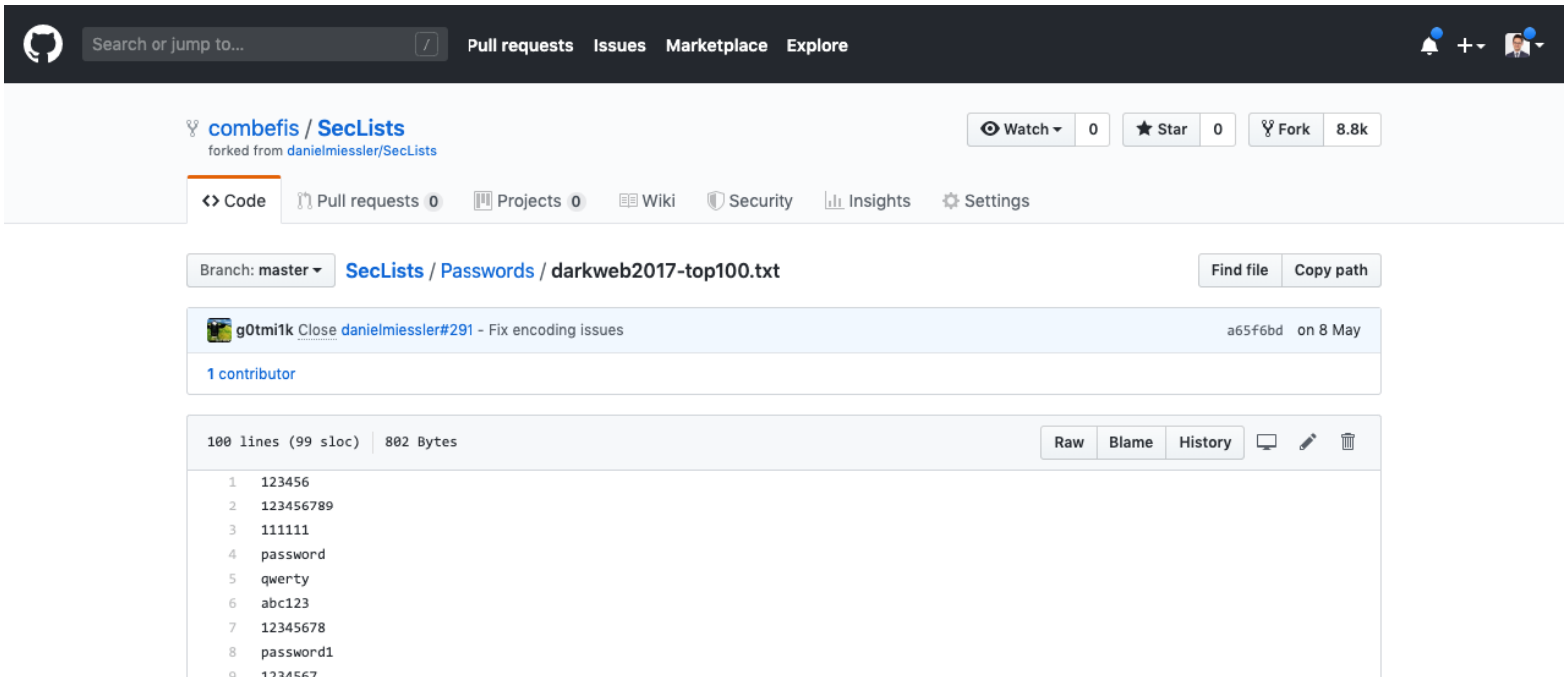
Dictionary Attack

- Try all the passwords from a dictionary

  It is an improvement of brute-force attack, with fewer trials

- Most people use common words as passwords

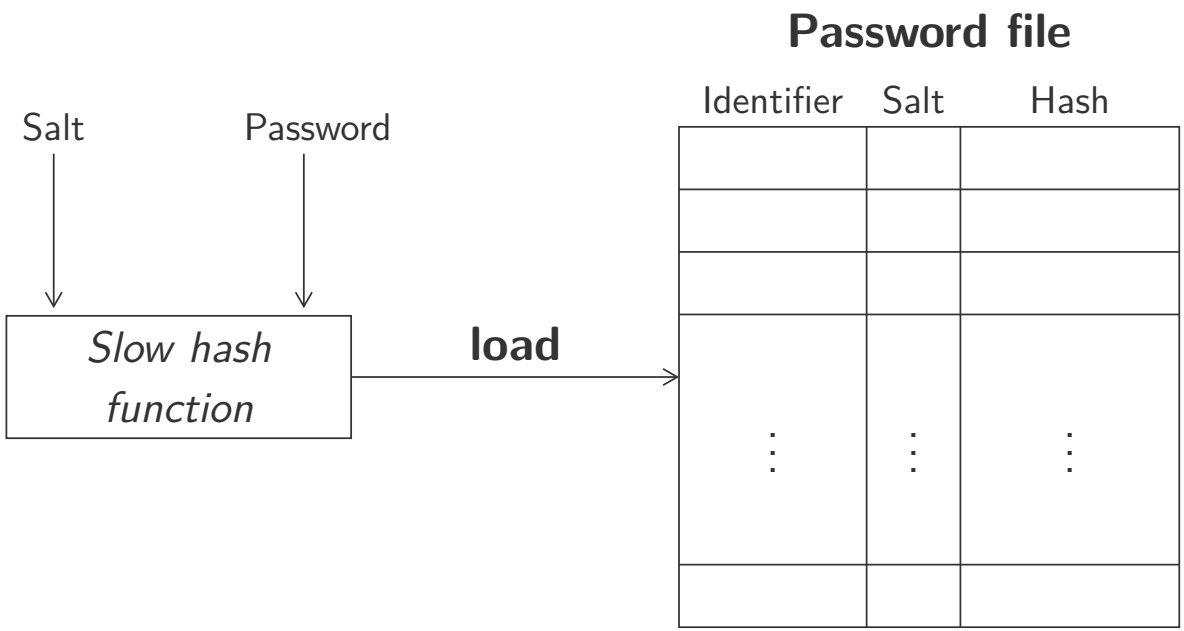
  Dictionary can be general or specialised for a particular target



Unix Password Scheme (1)

- Adding a new password in the system

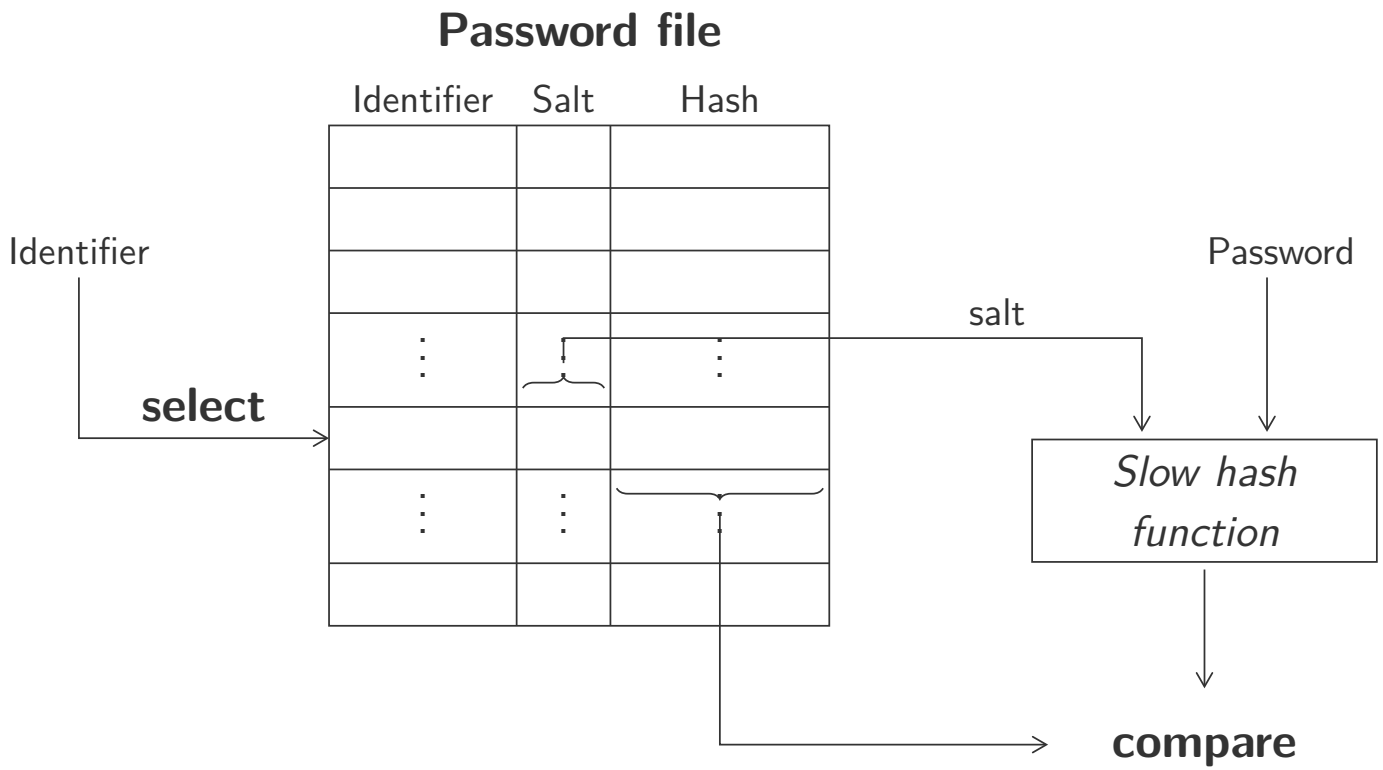
  The salt is chosen by the system and used to compute the hash



Unix Password Scheme (2)

- Checking a password for a given identifier

  The goal is to authenticate a user



Salt

- Using a salt when generating the password hashes

  Combination of the password with a fixed-length salt

- Using a salt servers three main purposes
  - Prevent duplicates passwords to be visible in the database
  - Increases the difficulty of offline dictionary attacks
  - More difficult to identify same password on several systems

Password File

- An opponent should be denied the access to the password file

  Must be stored securely and only accessible by privileged users

- Hashed passwords are often stored separately from identifiers

  Specific file referred to as shadow password file

- Passwords file is still vulnerable

  Unanticipated break-ins, protection accident, sniffing, etc.

Password Selection

- Complex for a user to select a password
  - Too short or too easy to guess if chosen by the user
  - Impossible to remember if effectively impossible to crack
- Four basic techniques to eliminate guessable passwords
  - Guidelines to educate users to choose hard-to-guess passwords
  - Computer-generated passwords are difficult to memorise
  - Reactive password checking to ask user to change it
  - Proactive password checker when the user is choosing it

Proactive Password Checking

- Simple system for rule enforcement about the passwords

  At least 8 chars, upper/lower, numeric digit, punctuation mark

- Password checker against a dictionary of "bad" passwords

  Need a lot of space to be stored (30 MB) and time to be searched

- Linux uses bloom filters to reject some passwords

  Password hash similar to hashes of passwords from a dictionary

Token

Token-Based Authentication

- Authenticating users thanks to tokens

  That is an object the user possesses and is unique to him/her

- Several types of cards can be used as tokens
  - **Embossed**: raised characters (old credit card)
  - **Magnetic stripe**: (bank card)
  - **Memory**: electronic memory (prepaid phone card)
  - **Smart**: electronic memory and processor (biometric ID card)

- Smart cards can be either with contact or contactless

  Electrical contacts exposed on surface or embedded radio antenna

Memory Card

- Memory card stores data but cannot process them
  - Magnetic stripes can be read and reprogrammed by a reader
  - Can be used alone for a physical access (hotel room)...
  - ...or with a PIN or password for user authentication
- Several drawbacks of using tokens
  - May require costly special reader to maintain (HW/SW)
  - Loosing the token prevent the user to gain access to the system
  - Inconvenient for a user for computer access

Smart Card (1)

- Different physical characteristics but all embed microprocessor

  Can look like cards, calculators, keys, etc.

- Can be equipped with user interface elements

  Can have keypads and displays for human-token interaction

- Electronic interface to communicate with the reader/writer
  - **Contact**: card inserted and direct connection
  - **Contactless**: reader close proximity, antenna communication

Smart Card (2)

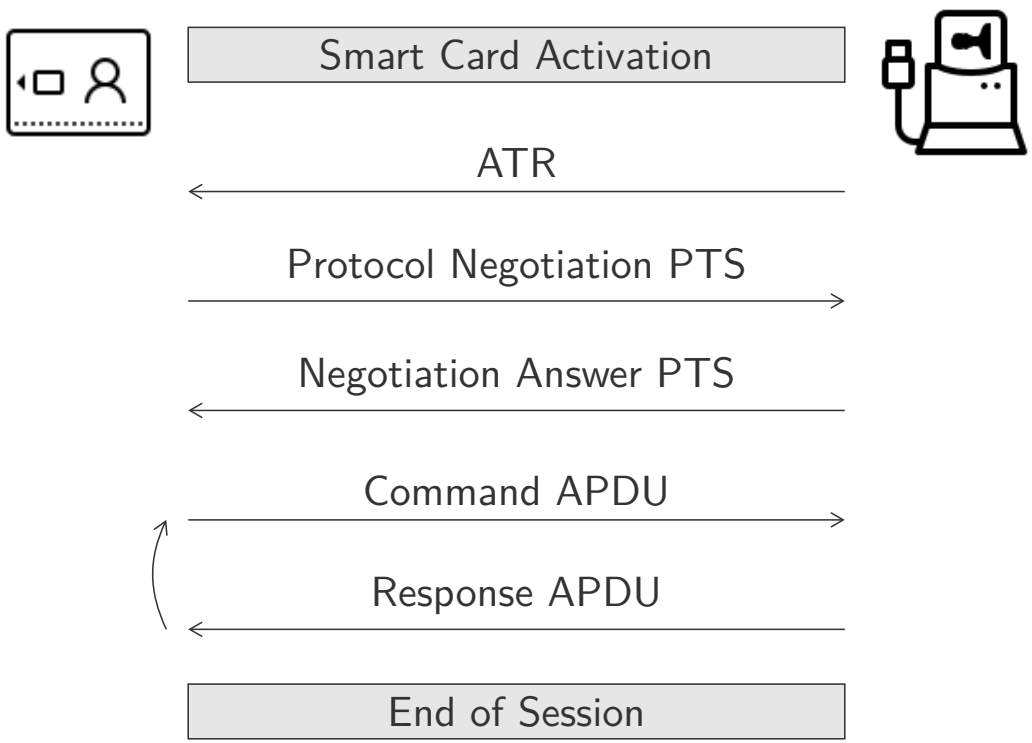
- Three categories of supported authentication protocol
  - **Static**: user authenticated to token, then token to computer
  - **Dynamic pwd generator**: generates unique password regularly
  - **Challenge-response**: computer generates challenge for token
- Three kinds of memory can be used on smart cards

  ROM (card nb/holder), EEPROM (protocol, phone time), RAM

Smart Card/Reader Exchange

- Exchanges between the smart card and the reader are simple

  Application data exchanged (APDU) depend on protocol



Electronic Identity Card

- Smart card can be used as national identity cards for citizens
  - Used to provide access to government and commercial services
  - Verified by national government as valid and authentic
- For example, the German eID card *neuer Personalausweis*
  - Personal data and unique document number (identifier)
  - Card access number (CAN): six-digit decimal random number
  - Machine readable zone (MRZ): can also be used as a password

eID Function (1)

- Three separate electronic functions, own protected dataset
  - **ePass**: digital representation of cardholder's identity
  - **eID**: identity record accessible to authorised service
  - **eSign**: store private key and certificate verifying it
- ePass function exclusively reserved for government

  Can only be used offline (e.g. passport control checkpoint)
- Offline or online access to eID function by inspection system

  Read identifying information and also biometric ones

eID Function (2)

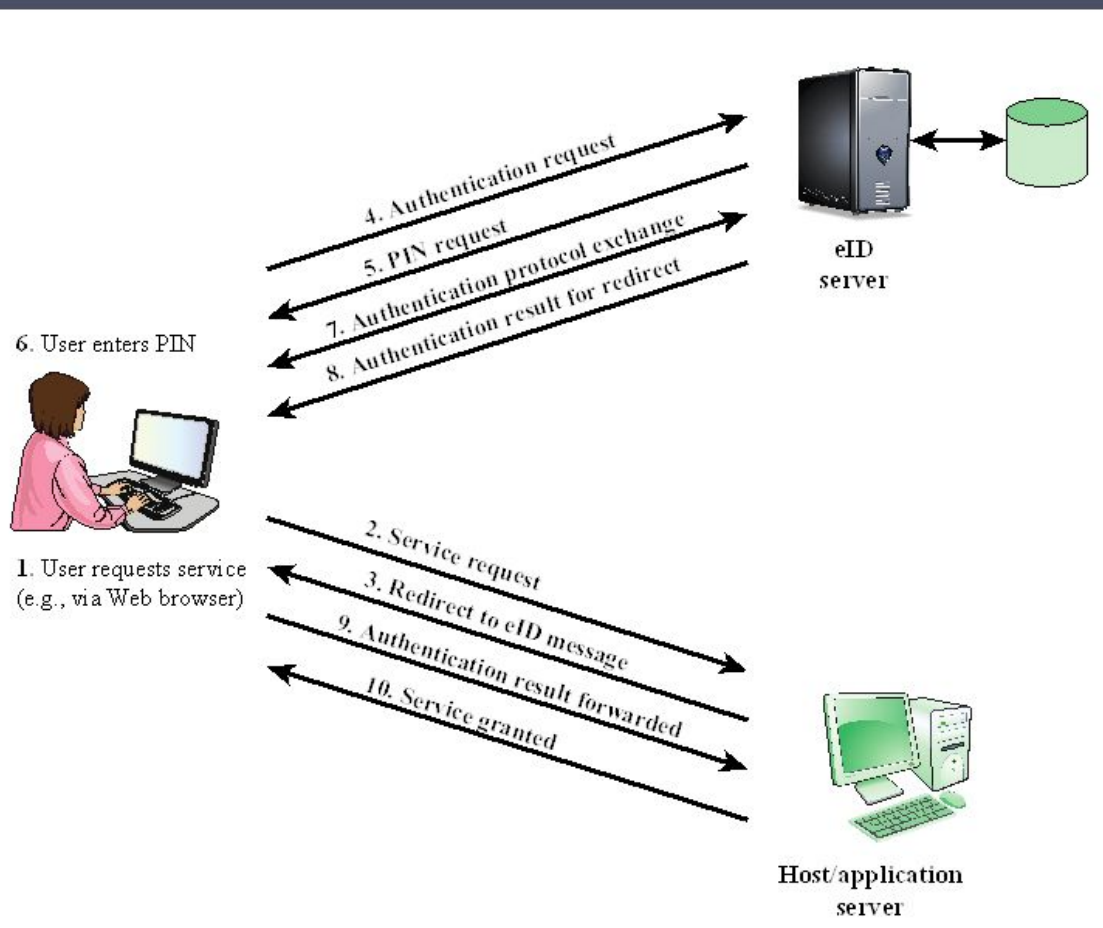


Figure 3.6 User Authentication with eID

PACE

- Password Authenticated Connection Establishment (PACE)

  Ensure that contactless RF chip not read without access control

- Using the eID PIN, CAN or MRZ as a PACE password

  Depending on whether it is an online or offline application

Biometric

Biometric Authentication

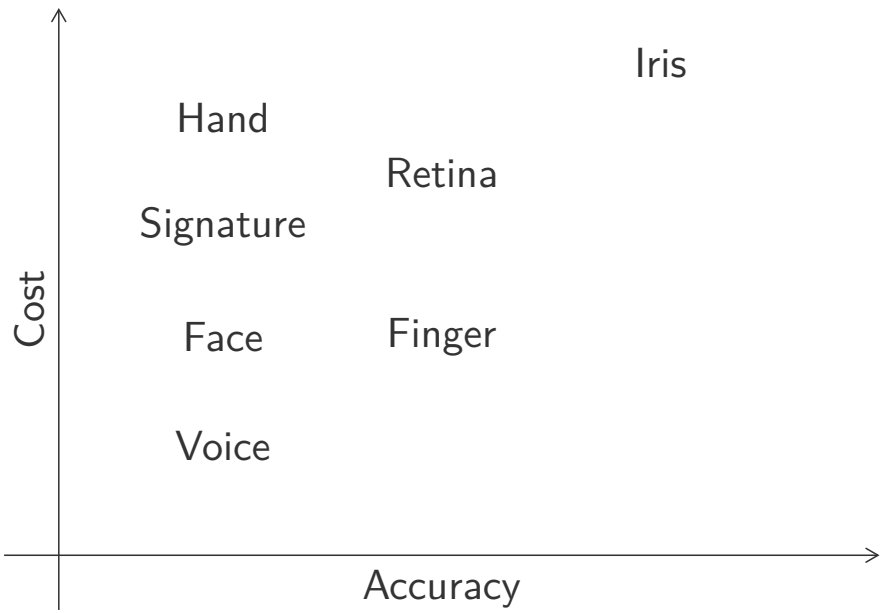
- Authenticating user based on unique physical characteristics
  - **Static**: fingerprint, hand geometry, facial characteristics, retinal and iris pattern
  - **Dynamic**: voiceprint and signature
- Biometric authentication based on pattern recognition

  Both technically more complex and expensive

Biometric Cost

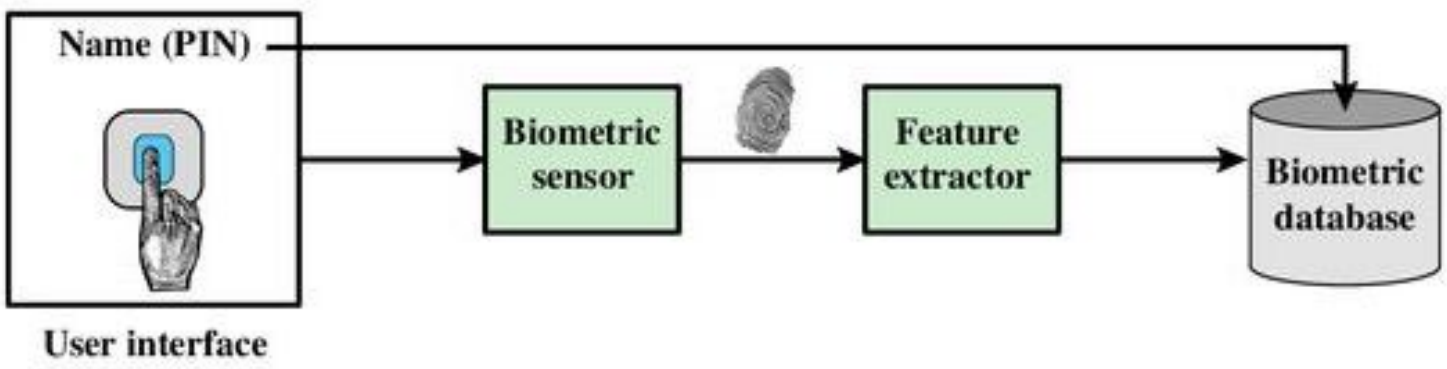
- Biometric authentication not mature enough yet

  As a standard for user authentication to computer systems



Generic Biometric System (1)

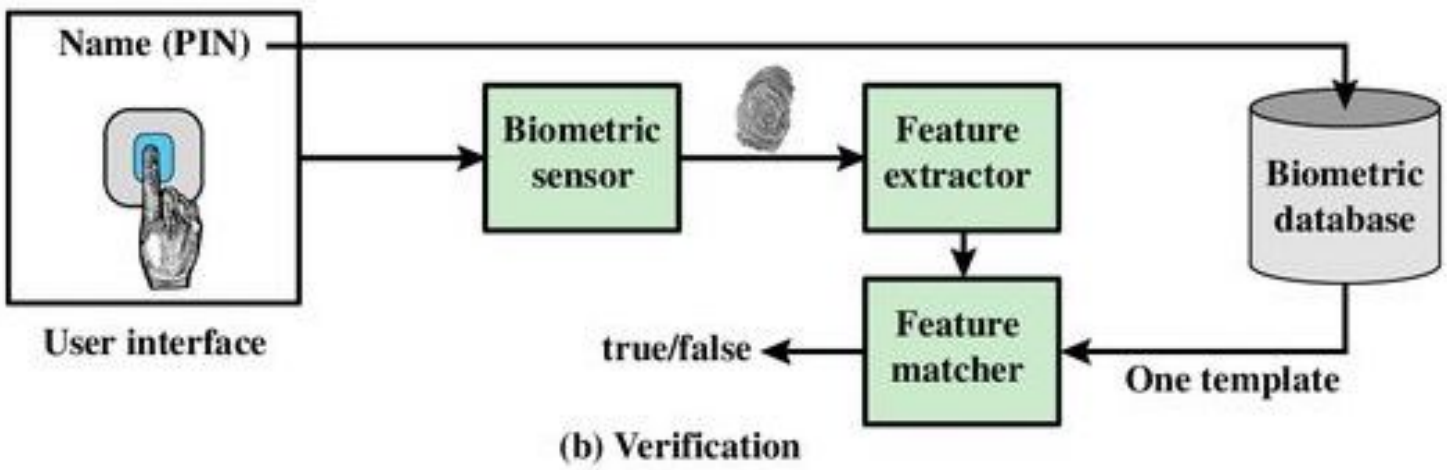
- Authorised users must first be enrolled in the system
  - Identifier, password or PIN and biometric characteristics sensed
  - Digitisation and features extraction



(a) Enrollment

Generic Biometric System (2)

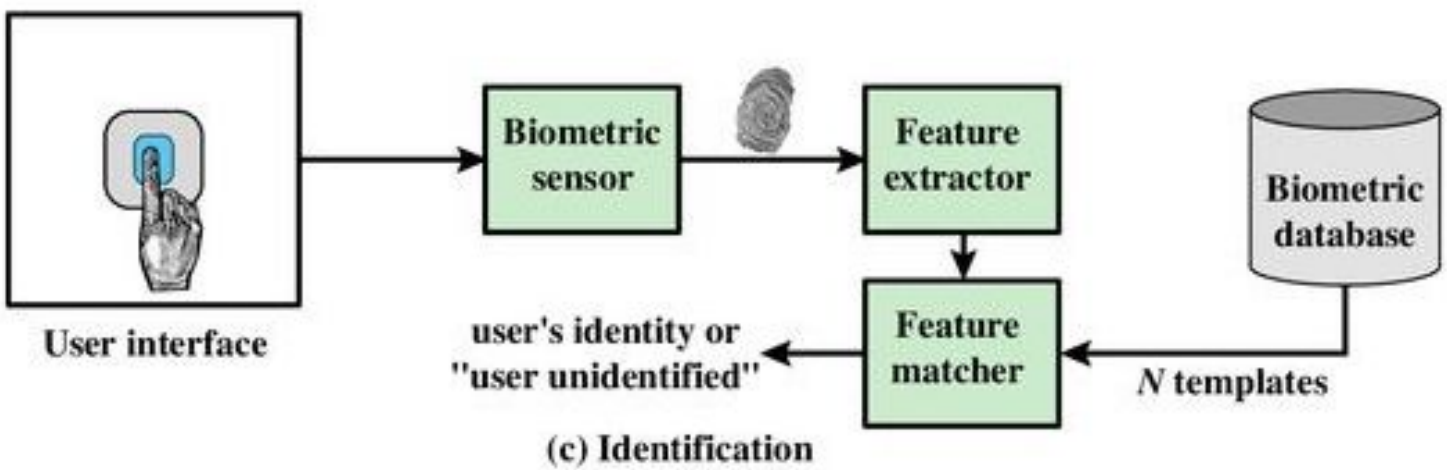
- Verification when user wants to log-in a system
  - Similar to using a token after having input the password/PIN
  - Comparison of the extracted features by sensor with database



Generic Biometric System (3)

- Possible to build an identification system

  Searching for a user with similar biometric information



Credits

- Hideya HAMANO, January 5, 2015, https://www.flickr.com/photos/mawari/16021501609.
- Jack Acecroft, March 4, 2013, https://www.flickr.com/photos/jackace/8663584323.
- Jonathan Molina, March 23, 2009, https://www.flickr.com/photos/knk/3379898651.
- Aleksi Aaltonen, June 1, 2010, https://www.flickr.com/photos/aleksiaaltonen/4659509151.