IT security

Monday 25th February
Course day #3

Theme A (iii)
User authentication
Case: EU's digital passports

Niels Christian Juul (ncjuul@ruc.dk)
Niels Jørgensen (nielsj@ruc.dk)

Literature

Stallings & Brown

- Chapter 2 (2.4-2.7)
- Chapter 3 (3.1-3.6 + 3.8-3.9)
  - p 102-104 omit or read extensively only
  - p 117 and Figures 3.13 (b), (c) and (d) omit or ext. only
- Chapter 23.2

Additional (mandatory) literature:

- Hoepman et al. Crossing Borders
  - p 6-7 and upper half of p 8 may be skipped

Case: EU's digital passports

Application of
- biometry
- digital signatures
- two of the four main technologies
  for user-authentication

Main lessons
- which digital signatures/certificates to trust?
  can we trust?
- privacy aspect

Four themes (A-D)

A. Computer security technology and principles (Part One in Stallings & Brown)	11th Feb 18th Feb 25th Feb
B. Software and system security (Part Two)	11th March (#5)
C. Management issues (Part Three)	4th March (#4) 18th March 25th March
D. Network security (Part Five)	1st April 8th April

Topic:
- User authentication

Case:
- EUs digital passports

Exam questions for theme A (see list on moodle)

Q1: In the NotPetya attack, how was encryption used?

Q2: Following the NotPetya attack, what were the management lessons?

Q3: “Hybrid encryption model” of ransomware

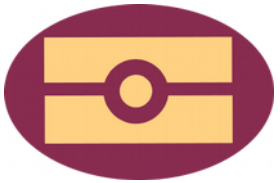
Q4: Taxonomy of ransomware

Q5: Advantages and disadvantages of biometric authentification

Q6: Privacy issues of digital passports. “Basic access control” and “Extended access control” in EUs standard.

(Perhaps there should be a question about passwords)

Plan for today



User authentication: passwords (Chapter 3) (NJ)

Digital signatures and digital certificates (Figure 2.7-2.8) (Thomas)

User authentication: tokens (NJ) (Chapter 3)

Main advantages and disadvantages of biometric authentication? (Stallings & Brown) (Daniel)

EU’s digital passports (NJ)

What is user authentication?

“In essence, identification is the means by which a user provides a claimed identity to the system;
user authentication is the means of establishing the validity of the claim.”
(Stallings & Brown p86)

Estalishing (verifying) a claimed identity

Claimed identity

Establishing identity (verifying)

User authentication

Three major methods for user authentication (in Chapter 3)

(1) Password-based
(2) Token-based
(3) Biometrics-based

We can consider asymmetric encryption a fourth method (from Chapter 2)
• sometimes used with (2) and (3)

Each method is using different *means* of authentication

Exercise

RUC logon
(username, password)

RUC's buildings
(id-card, pin-code)

EU's digital passports
(the passport)

For each user authentication system:
What means of authentication is used?

RUC logon

RUC logon
(username, password)

Gives access to
- digital services at RUC (email, moodle, ..)
- and other universities

What means of authentication is used?
- *something the individual knows (a password)*

RUC’s buildings

RUC’s buildings
(id-card, pin-code)

What does the system give access to?

- Some of RUC’s buildings

What means of authentication is used?

- *something the individual knows (a pincode)*
- *something the individual possesses (id-card)*

EU’s digital passports

EU’s digital passports
(the passport)

What does the system give access to?

- the passport holder is allowed entry to a country

What means of authentication is used?

- *something the user is (static biometry): face, height, age,..*

Comparison

RUC logon
(username, password)

username+password
is only way to log on
(with some exceptions)

RUC's buildings
(id-card, pin-code)

there may be other ways
to access buildings
(eg., windows)

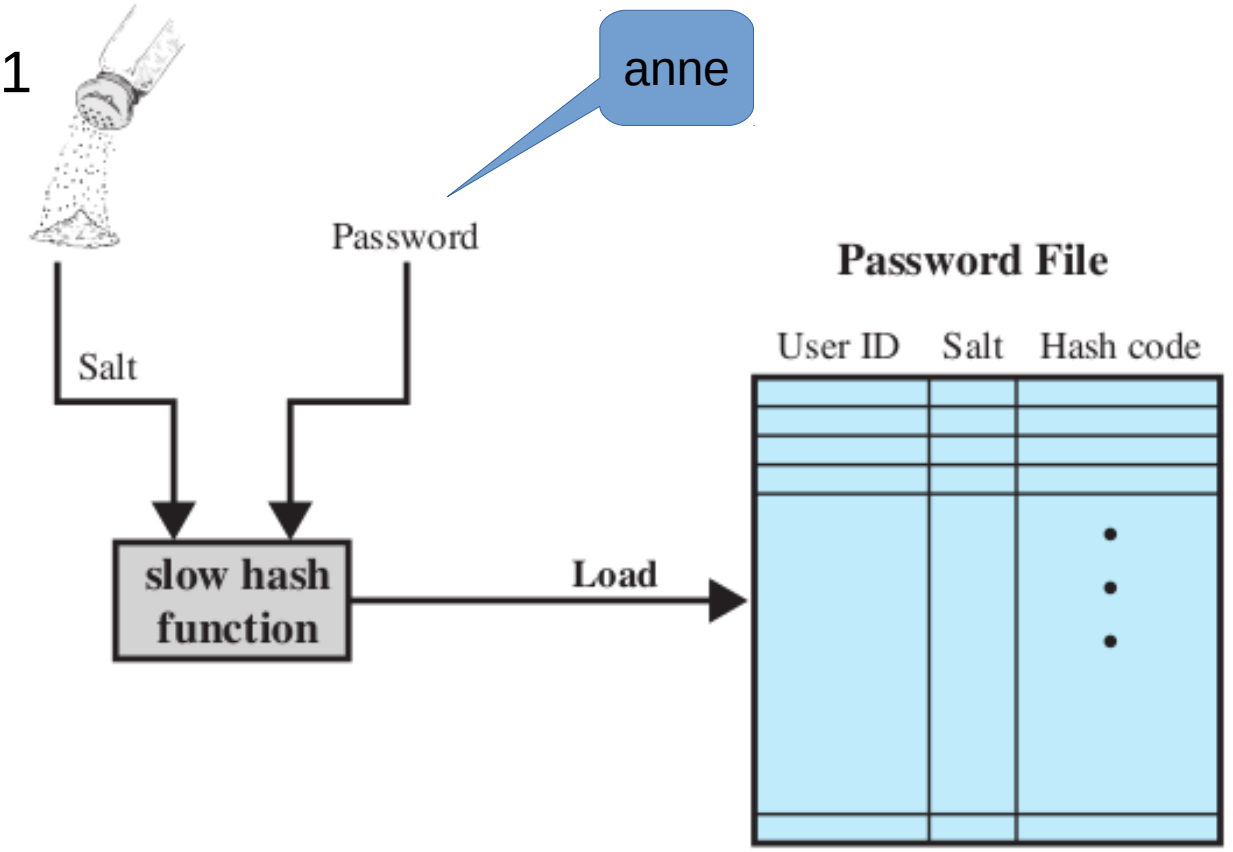
EU's digital passports
(the passport)

there may be other ways
to enter a country
(a border crossing
without password gates)

- used in many ways
- automated vs. manned passport gates

Technology #1: Password-based user-authentication

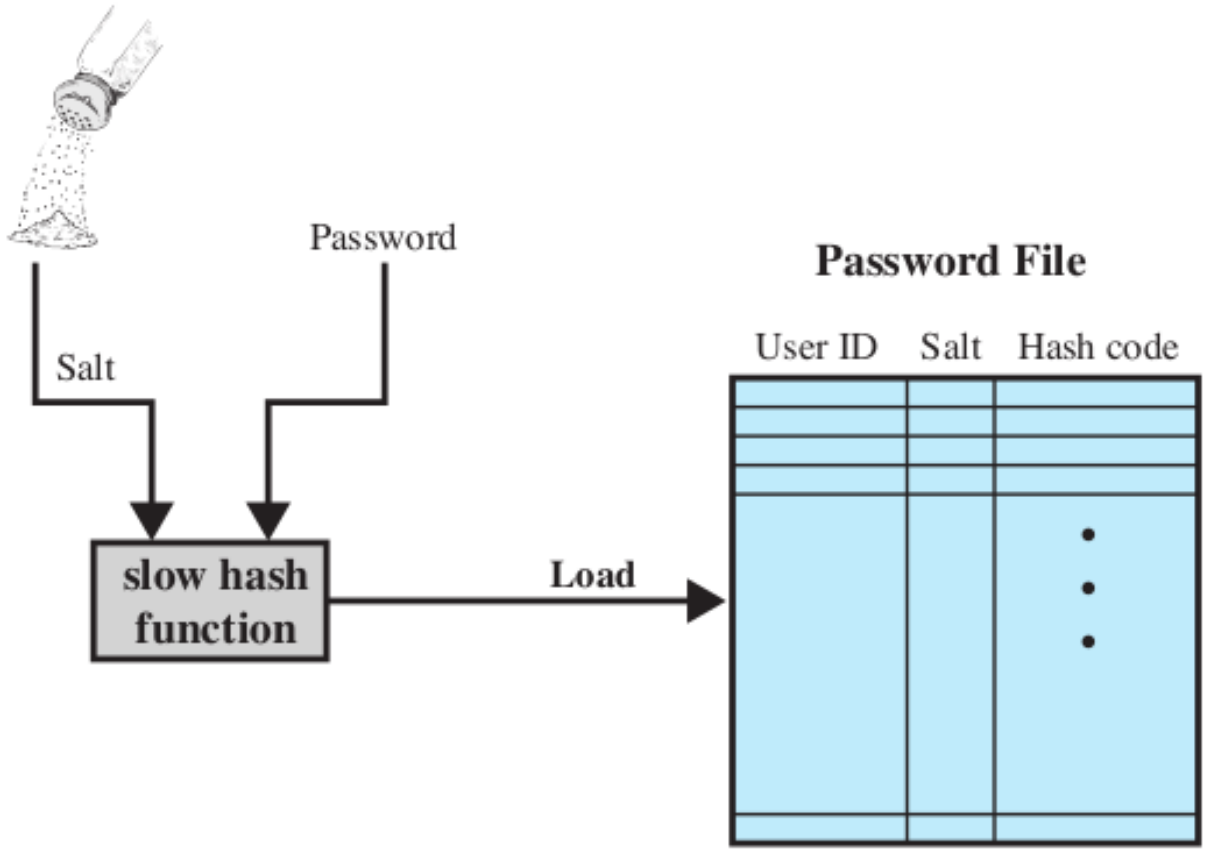
011011011101



A Password file on Unix and Mac contains

- user ID (say, ncjuul)
- a unique salt, say with 12 bits: 011011011101 (better 48 or 128 bits)
- the hashvalue of salt+password, say: sha-512 “011011011101anne”
- hashvalue is 3e1aef..

Exercise

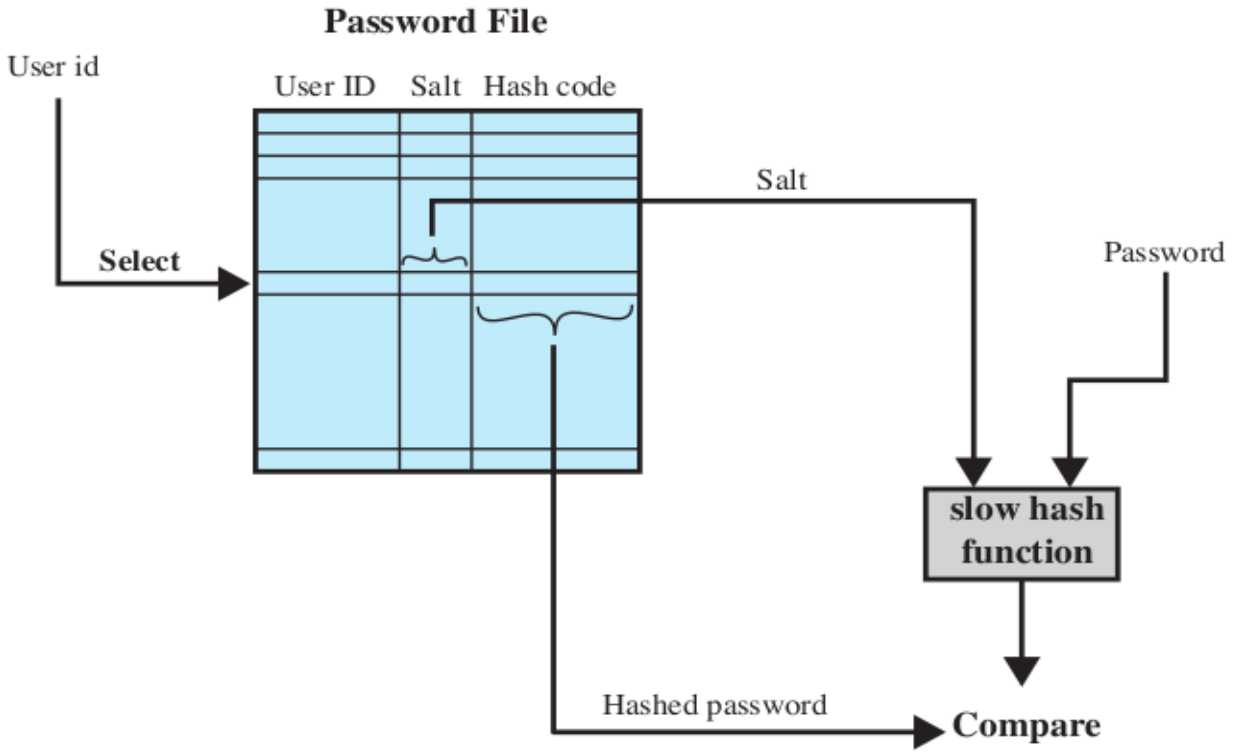


Salt:
- why is the salt stored in clear? (not hashed or encrypted)

Hash code:
- why is the salt+password hashed? (not encrypted)

Hash algorithm:
- why should we use something slower than sha 512?

Exercise solution



Salt:

- why is the salt stored in clear? (not hashed or encrypted)
- *the salt is needed when verifying the password*

Hash code:

- why is the salt+password hashed? (not encrypted)
- *encryption would introduce a risk: theft of the key*

Hash algorithm:

- why should we use something slower than sha 512?
- *to slow down attacks (not so slow as to annoy legitimate use)*

Passwords attack #1: dictionary attack on a single account

"Offline attack" = attacker has copy of password file

Attack algorithm in Java pseudo-code:

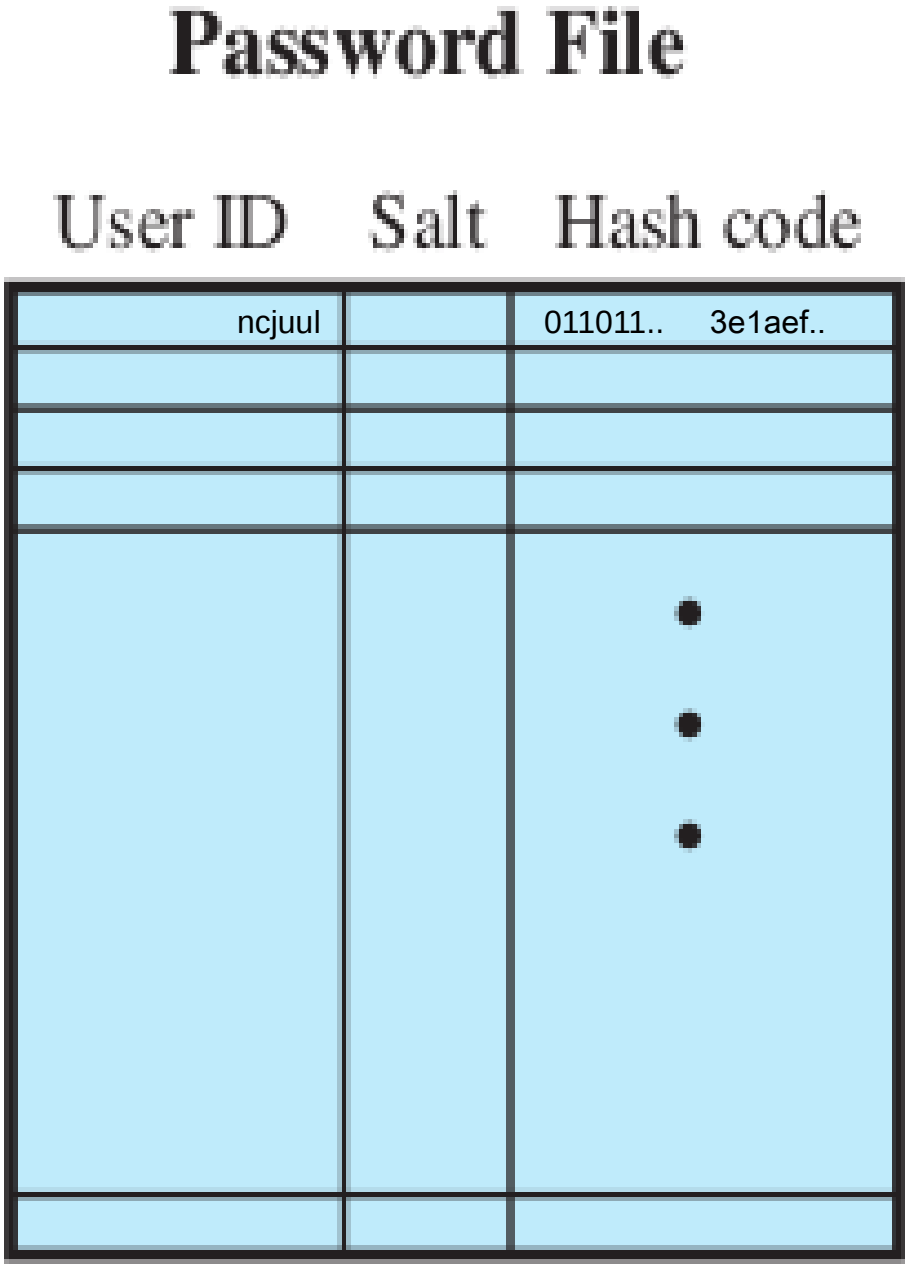
```
do {
   pwguess := passwordguess();
   while ( hash(salt_ncjuul+pwguess)
             != hashcode_ncjuul     )
}
// now pwguess is the password
```

The function passwordguess()
first select words from a "dictionary"

- pizza, peace (ordinary words)
- niels, peter, anne (words that are na

Then uses combinations, permutation

- P1zza123 (capitalization, i -> 1, ..)
- say, a total of 10^9 combinations

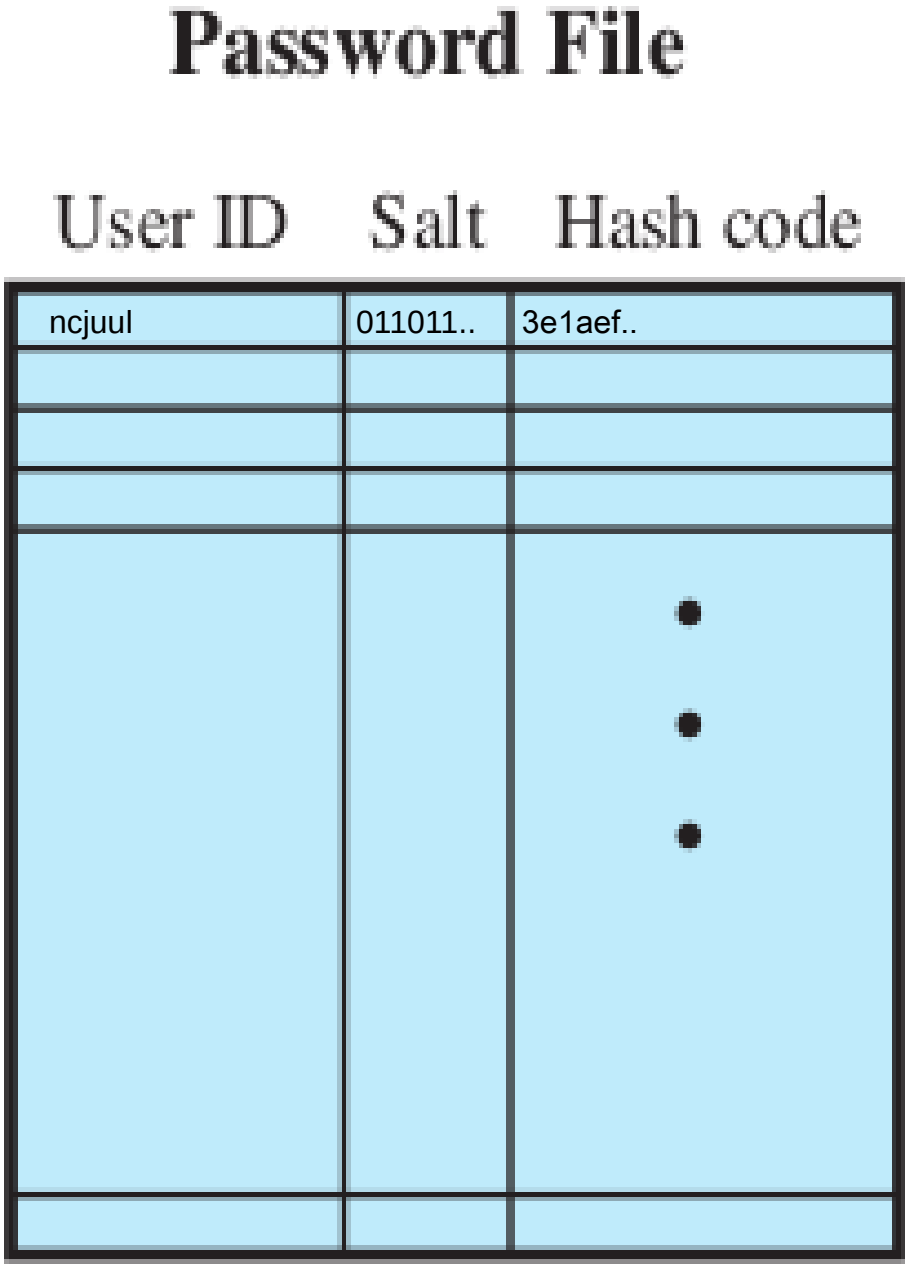


Passwords attack #1: dictionary attack on a single account

Exercise:

What are the futre countermeasures against this attack? Note the attack beacuse more powerful with the increasing cpu power

- sha-512(sha-512(.. (password+salt)..))

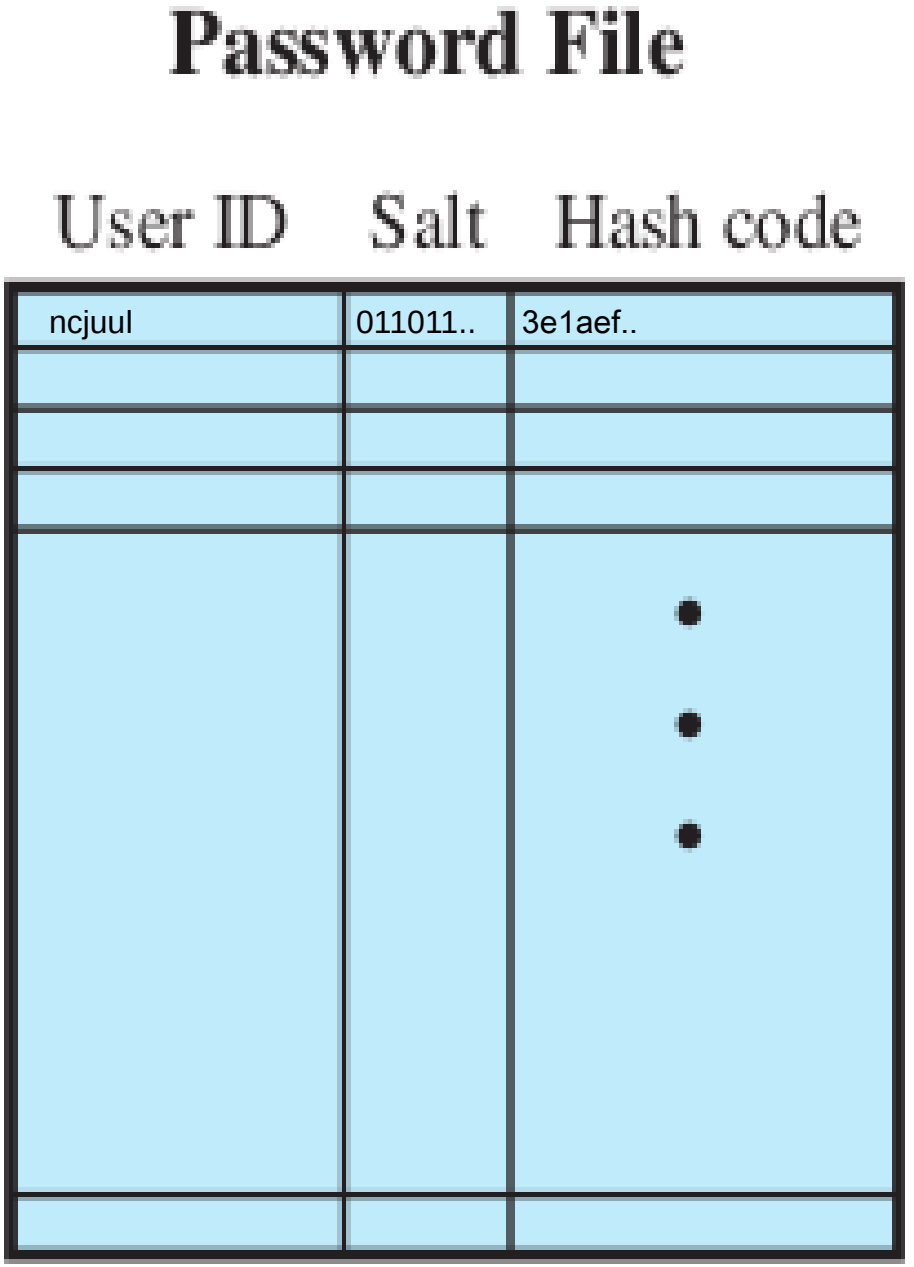


Passwords attack #1: dictionary attack on a single account

Exercise solution:

What are the futre countermeasures against this attack? Note the attack beacuse more powerful with the increasing cpu power

Slowing down the hash function even further.



Password attack #2: rainbow table

Attack *some* account
Attack algorithm in Java pseudo-code:

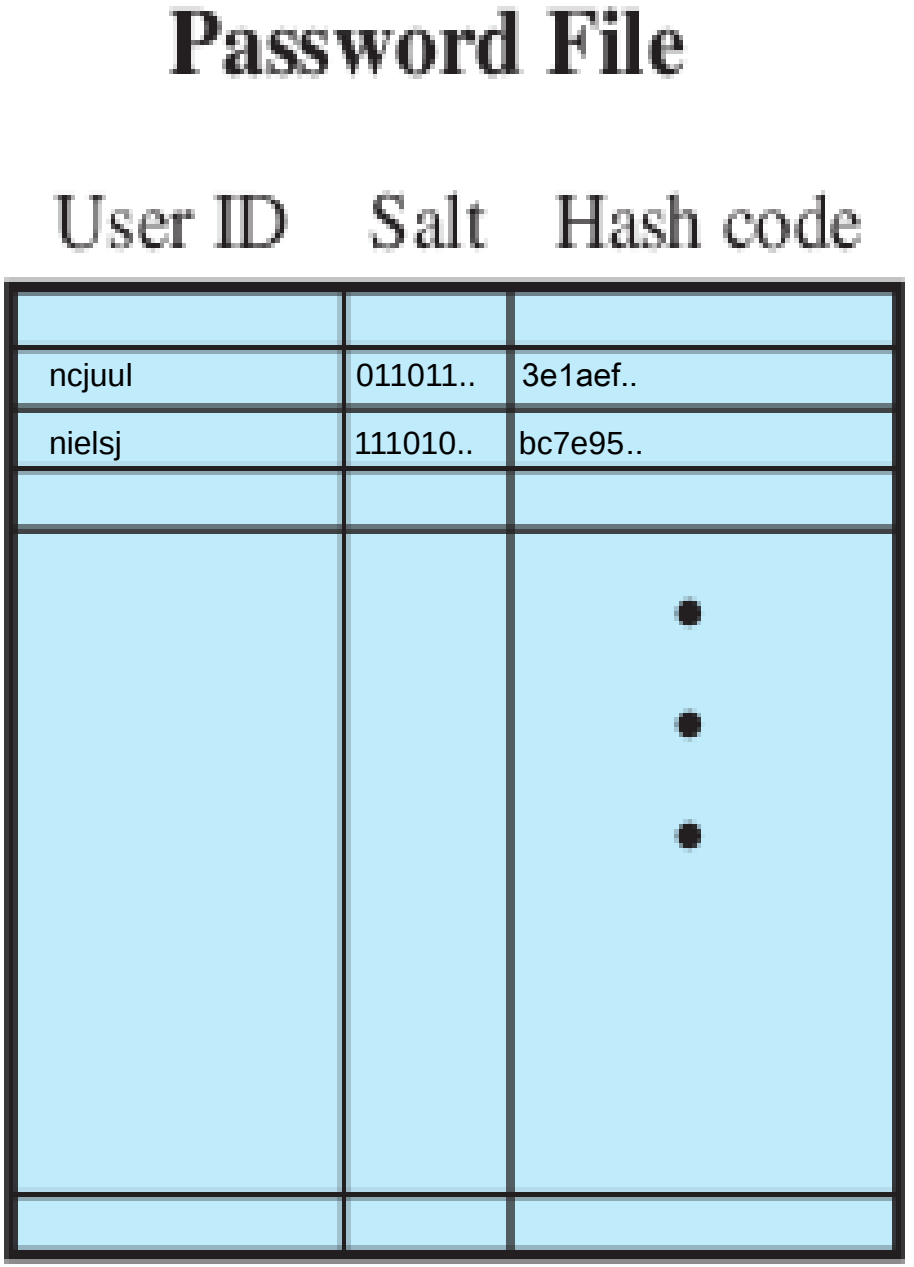
```
do {
   hcguess := hashcodeguess();
   user := select_user();
   while (hashcode_user) != hcguess
}
// now user has hashcode hcguess
```

The function hashcodeguess()
selects hashcodes from a “ranbow table”.

When a match is found, look up in the rainbow table, to find password + salt.

Exercise:

- *what does the rainbow table contain?*
- *how many entries (rows) in the table? (12 bit salt and 10^6 passw-guesses)*
- *countermeasures in future arms race?*



Exercise solution

An entry in a rainbow table contains a passwordguess, a salt, and their hash value.

A total of $10^6 * 2^{12} = 4*10^9$ entries.
- that's 100 GB if one entry is 25B.

2^{12} entries for each password-guess
- one entry for each salt

10^6 password guesses

Passwordguess	*Salt*	*Hashvalue*
pizza	0000 0000 0000 0000 0000 0001 0000 0000 0010 ..	hashvalue hashvalue hashvalue ..
peter		
P1zza123		

Exercise solution (continued)

The future arms race:

- hardware development means larger storage capacity
- but size of salt can be increased

10^6 password guesses

2^{12} entries for each password-guess

- one entry for each salt

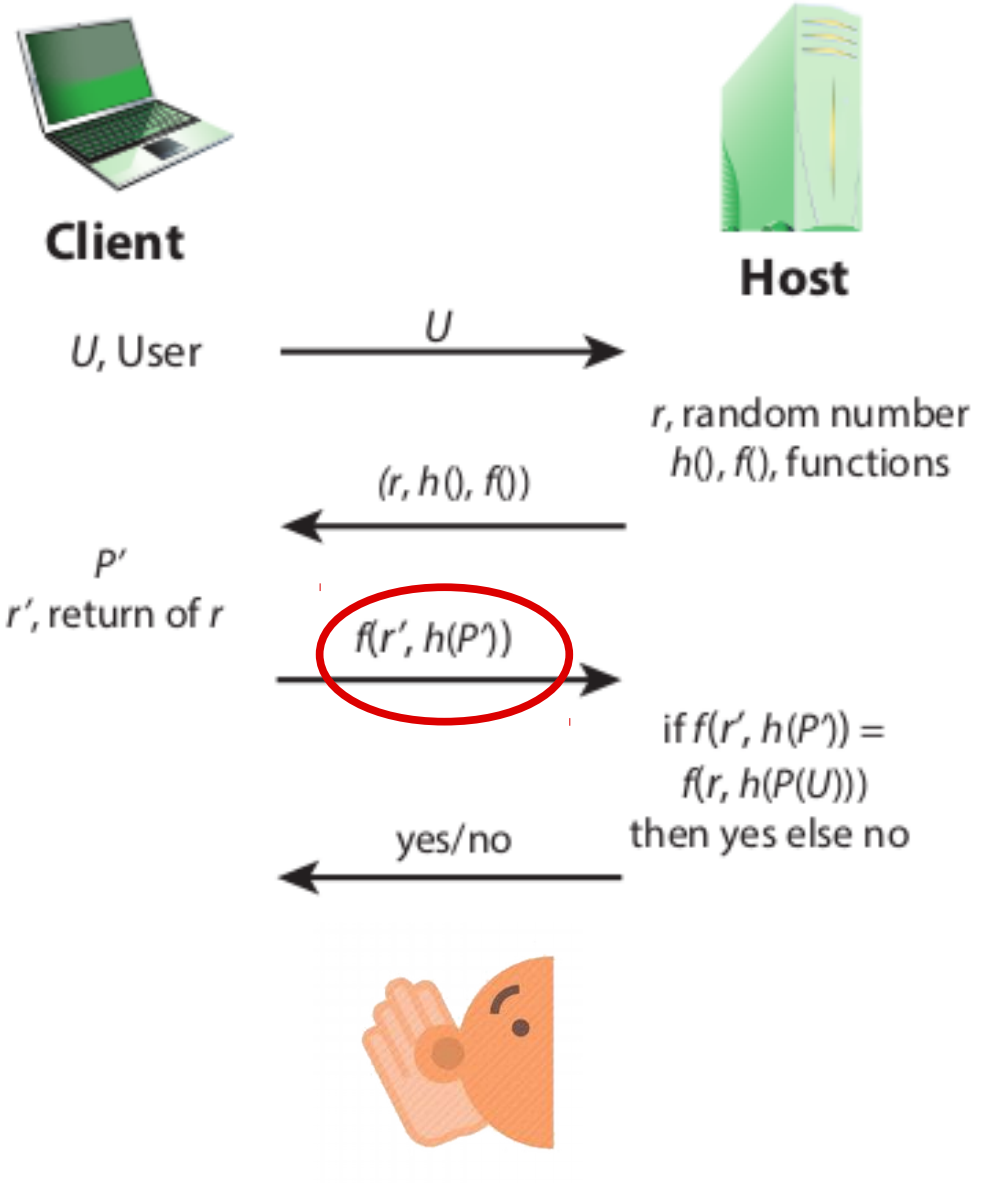
Passwordguess	*Salt*	*Hashvalue*
pizza	0000 0000 0000 0000 0000 0001 0000 0000 0010 ..	hashvalue hashvalue hashvalue ..
peter		
P1zza123		

Password attack #3: eavesdropping (remote logon)

Do not send password in the clear.

(1) Use encryption
(2) Use a challenge-response protocol.

Password attack #3: challenge-response protocol for passwords



P' is the password typed.

h(): the hash function of the password file, eg., sha 512 (but likely a slowed down function)

- standard password hashing takes place on the client side

f(): another hash function, say, sha 512 (ordinary, not slow)

- protects the hashed password

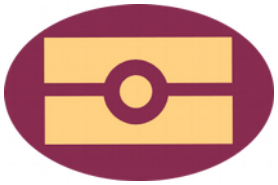
r: a random number

- protects against replay by the eavesdropper

h() and f() are the same at every logon

Plan for today

User authentication: passwords (Chapter 3) (NJ)



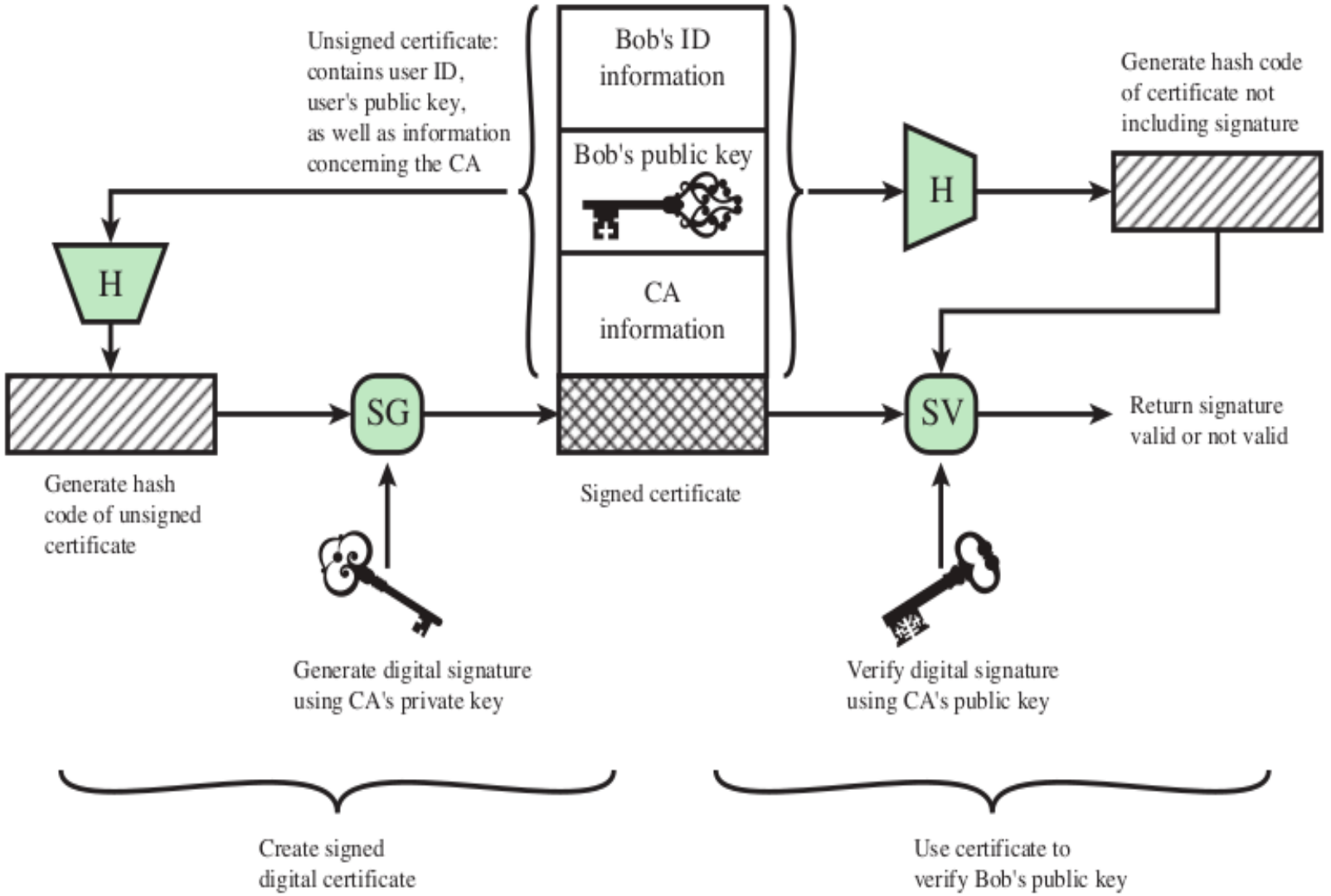
Digital signatures and digital certificates (Figure 2.7-2.8) (Thomas)

User authentication: tokens (NJ) (Chapter 3)

Main advantages and disadvantages of biometric authentication? (Stallings & Brown) (Daniel)

EU’s digital passports (NJ)

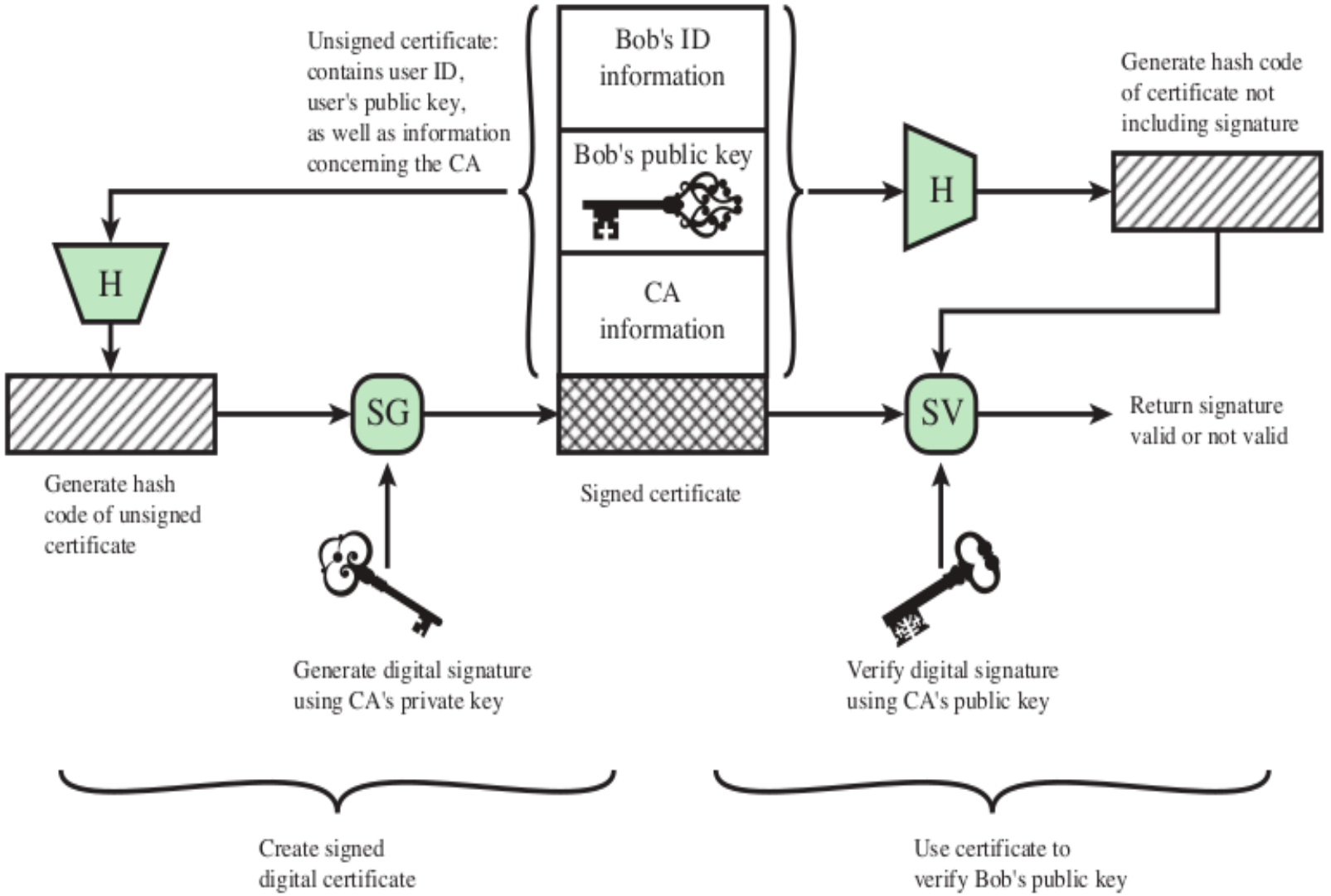
Digital certificates



Exercise:

Is a digital certificate a digitally signed message? (in the sense of Figure 2.7)

Digital certificates



Exercise solution:

Is a digital certificate a digitally signed message? (in the sense of Figure 2.7)

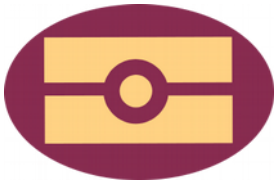
Yes.

- *the message M is the unsigned certificate*
- *the signature S is the signature on the certificate*

Plan for today

User authentication: passwords (Chapter 3) (NJ)

Digital signatures and digital certificates (Figure 2.7-2.8) (Thomas)



User authentication: tokens (NJ) (Chapter 3)

Main advantages and disadvantages of biometric authentication? (Stallings & Brown) (Daniel)

EU’s digital passports (NJ)

Tokens: PKI-based challenge-response protocol

Suppose in the future, we want to use a token to strengthen the logon process at RUC:

1. present ID
2. present password
3. present token, using a card reader attached to user's PC

- thus the user proves possession of the token

How would we design the token? Assuming

- PKI-based
- challenge-response protocol

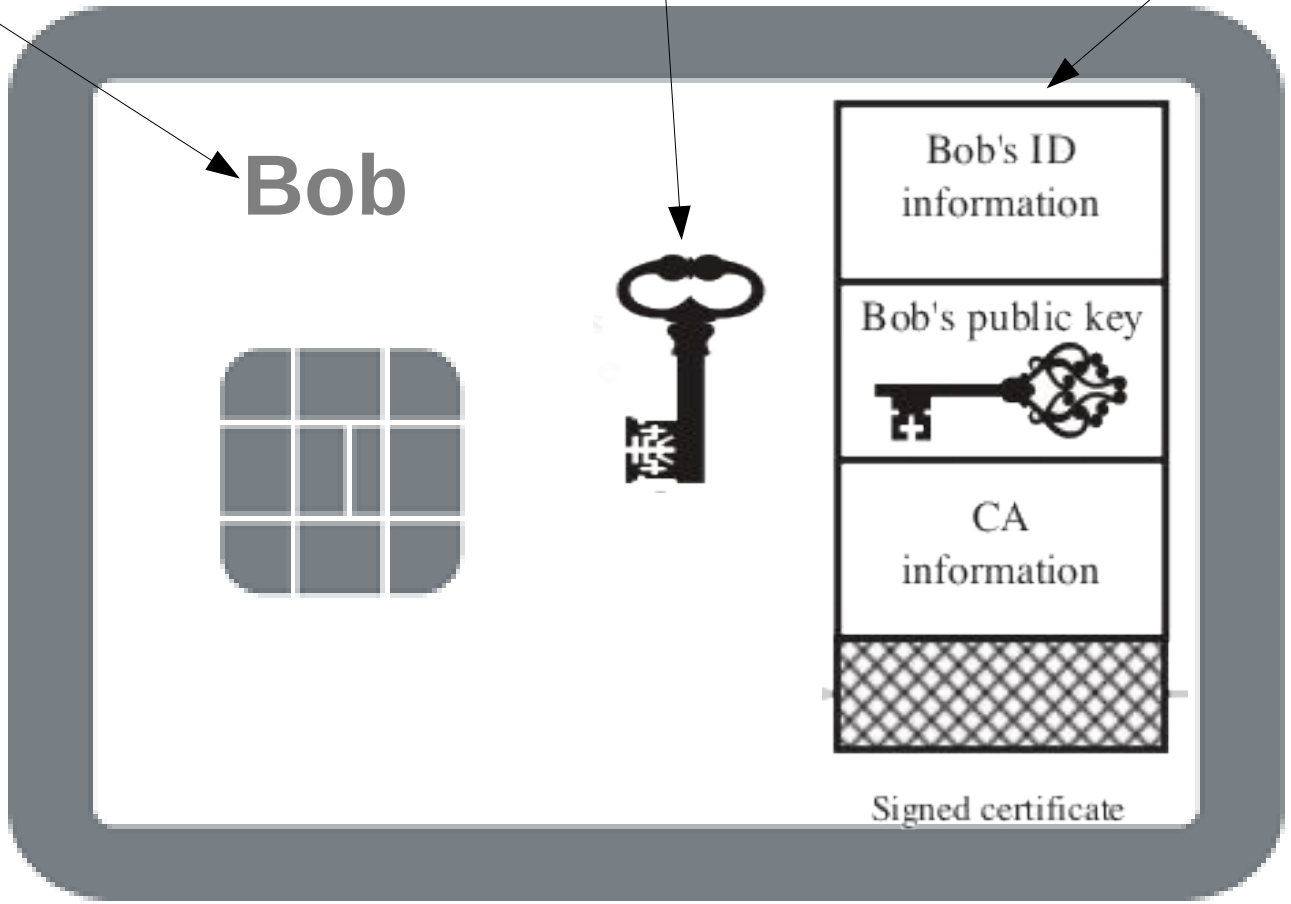
Design: Token for RUC logon

Bob’s name printed on the outside

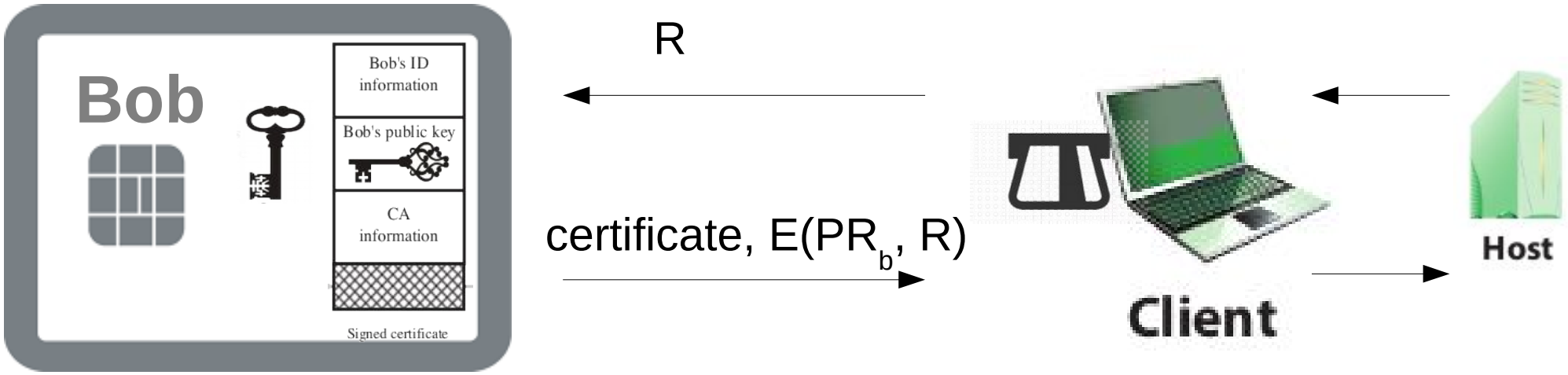
Bob’s private key

Certificate signed by RUC

- Bobs public key PU_b
- Information about the issuer of the certificate issuer (RUC)
- .. (Chpt. 23.2)



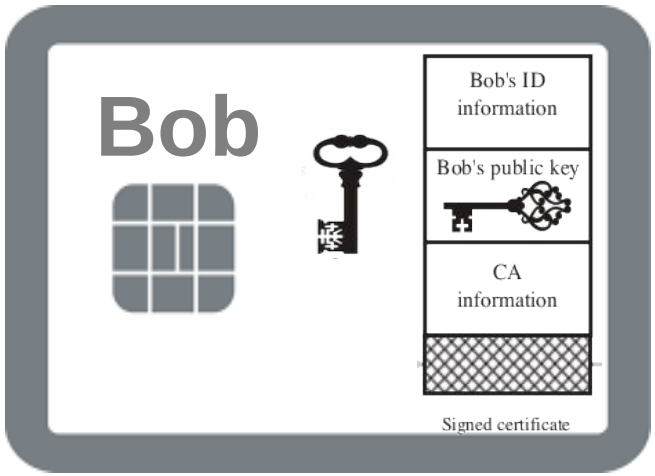
Challenge-response protocol



R is the challenge
- a random number
- prevents replay (if card would always encrypt the same challenge)

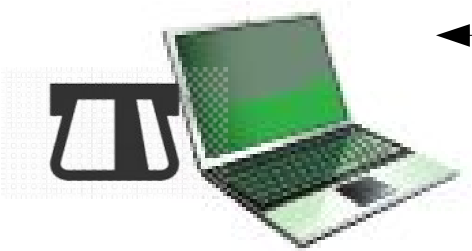
Bob proves he is in possession of the card
- only the card contains PR_b

Exercise



R

certificate, $E(PR_b, R)$



Client

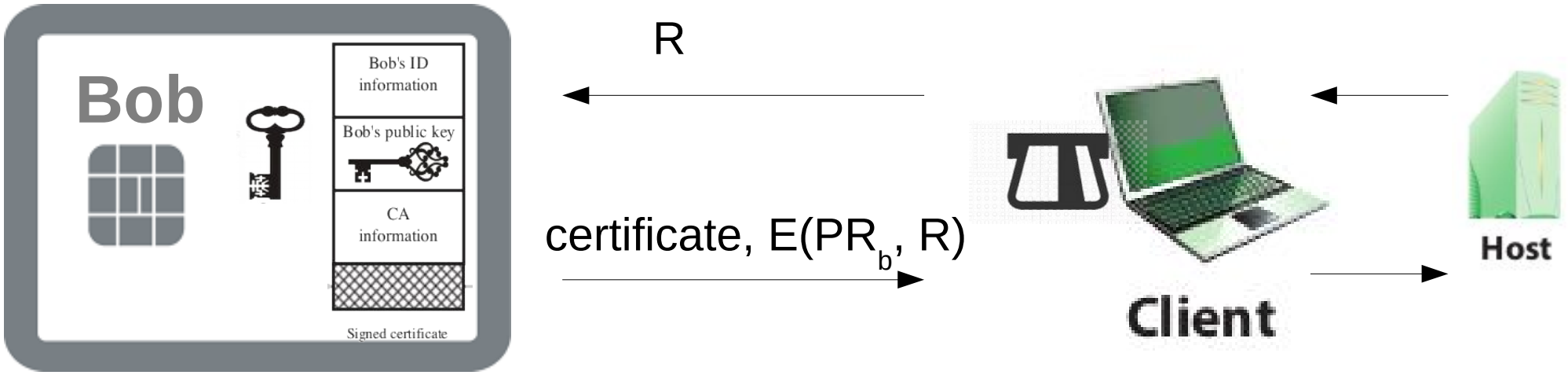
Host

How can the host verify that the response originates from Bob's card?

R is the challenge
- a random number
- prevents replay
- (if card would always encrypt the same challenge)

Bob proves he is in possession of the card
- only the card contains PR_b

Exercise solution



R is the challenge
- a random number
- prevents replay
- (if card would always encrypt the same challenge)

Bob proves he is in possession of the card
- only the card contains PR_b

How can the host verify that the response originates from Bob's card?

1. Decrypt $E(PR_b, R)$ using Bob's public key PU_b, checking the result is R.

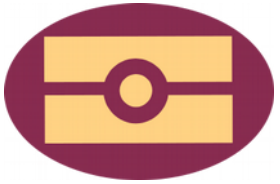
2. Verifying that PU_b is actually Bob's public key, according to RUC. This is by verifying the certificate. That is, verifying the signature on the certificate against PU_{RUC}.

Plan for today

User authentication: passwords (Chapter 3) (NJ)

Digital signatures and digital certificates (Figure 2.7-2.8) (Thomas)

User authentication: tokens (NJ) (Chapter 3)



Main advantages and disadvantages of biometric authentication? (Stallings & Brown) (Daniel)

EU’s digital passports (NJ)

Figure 3.10

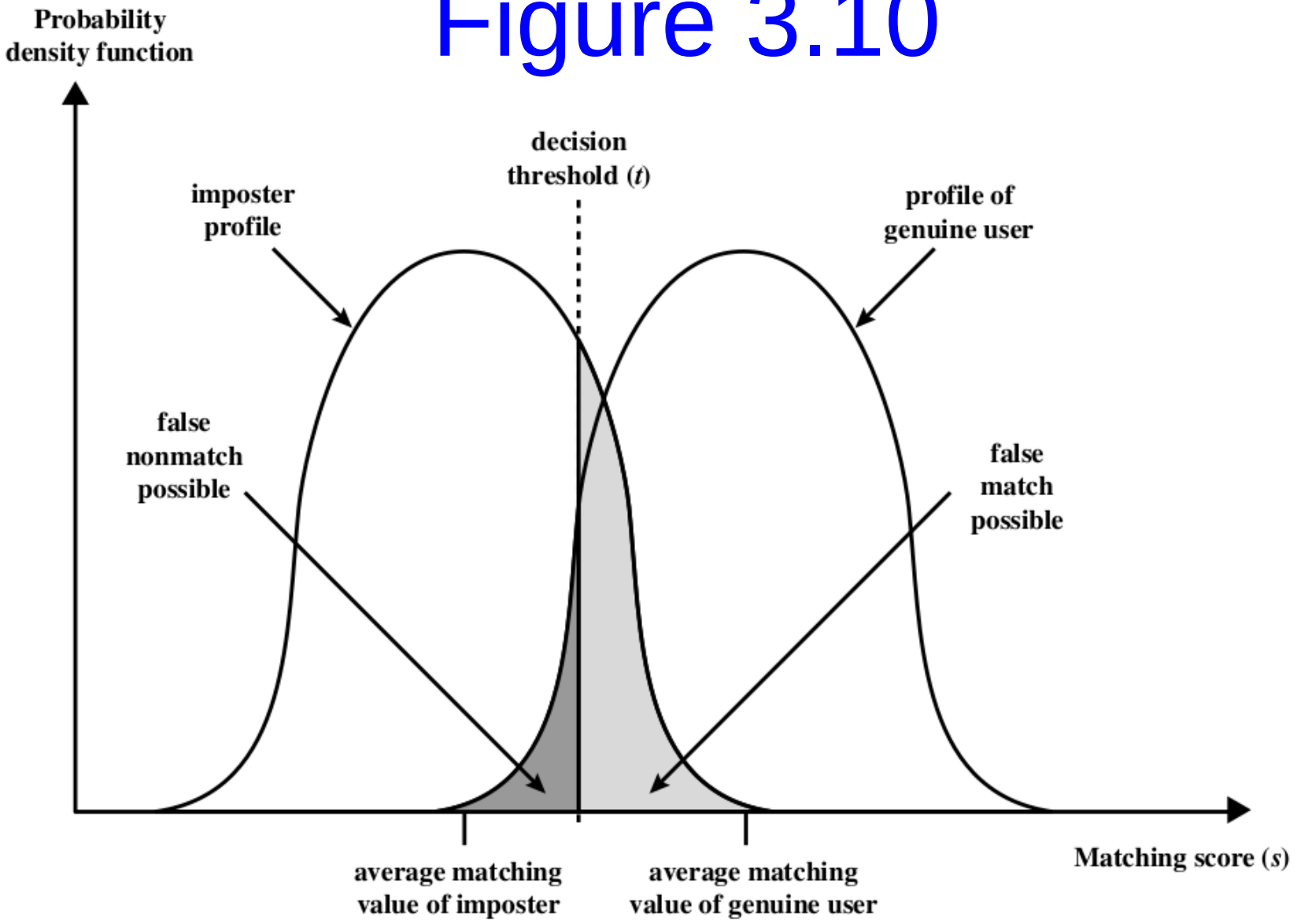


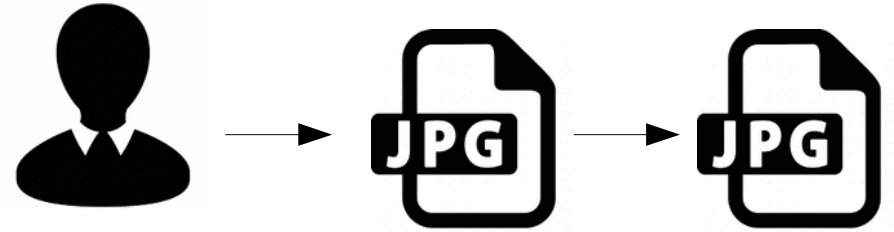
Figure shows matching against a single stored value

- say a fingerprint of a genuine user
- matched against genuine user (to the right)
- matched against imposer (to the left)

Biometric authentication

On a digital passport, in what form would biometric data be stored?

- fingerprints?
  - stored as a single number
  - to fingerprints compared numerically



- photo of face?
  - in EU's digital passports, stored as a JPG-file
  - two photos compared by facial recognition systems
  - or a person (customs officer) visually compares person's face with foto in passport (chip of paper)

Biometric authentication - advantages and disadvantages

Advantages

- may be more user-friendly
- iris and DNA very accurate

Disadvantages

- may be more costly
- face biometrics very inaccurate

- but may be only technology available for passport

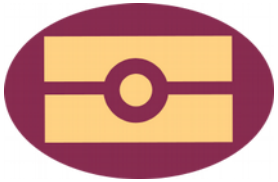
Plan for today

User authentication: passwords (Chapter 3) (NJ)

Digital signatures and digital certificates (Figure 2.7-2.8) (Thomas)

User authentication: tokens (NJ) (Chapter 3)

Main advantages and disadvantages of biometric authentication? (Stallings & Brown) (Daniel)



EU’s digital passports (NJ)

Digital passports

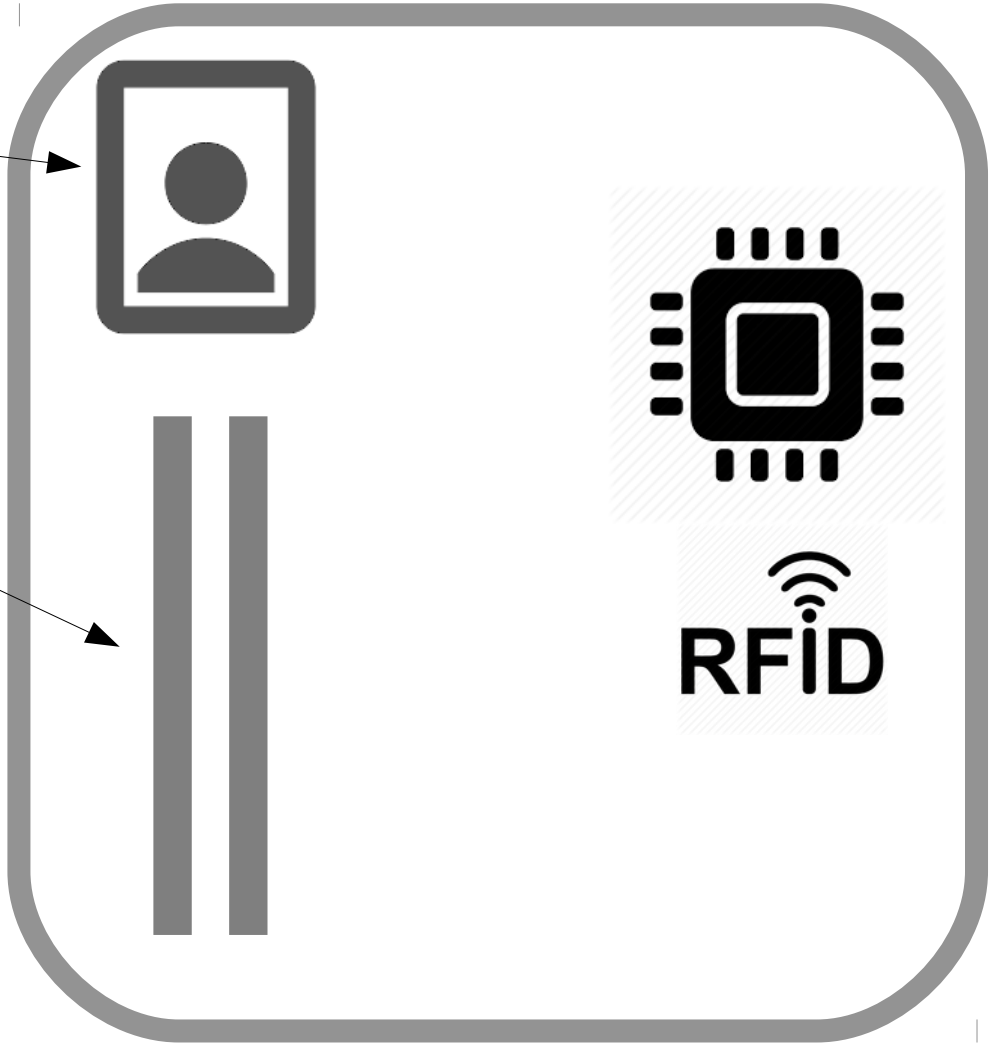
EU passports contain digital fingerprints
- in Denmark from 2012 onwards
- however, fingerprints not used

Governments want a method for establish authenticity of passport holder
- fingerprints preferred over photos
- prevent “look-alike” from using passport of other person
- faster, automated checks (?)

An EU passport (page 4)

Photo

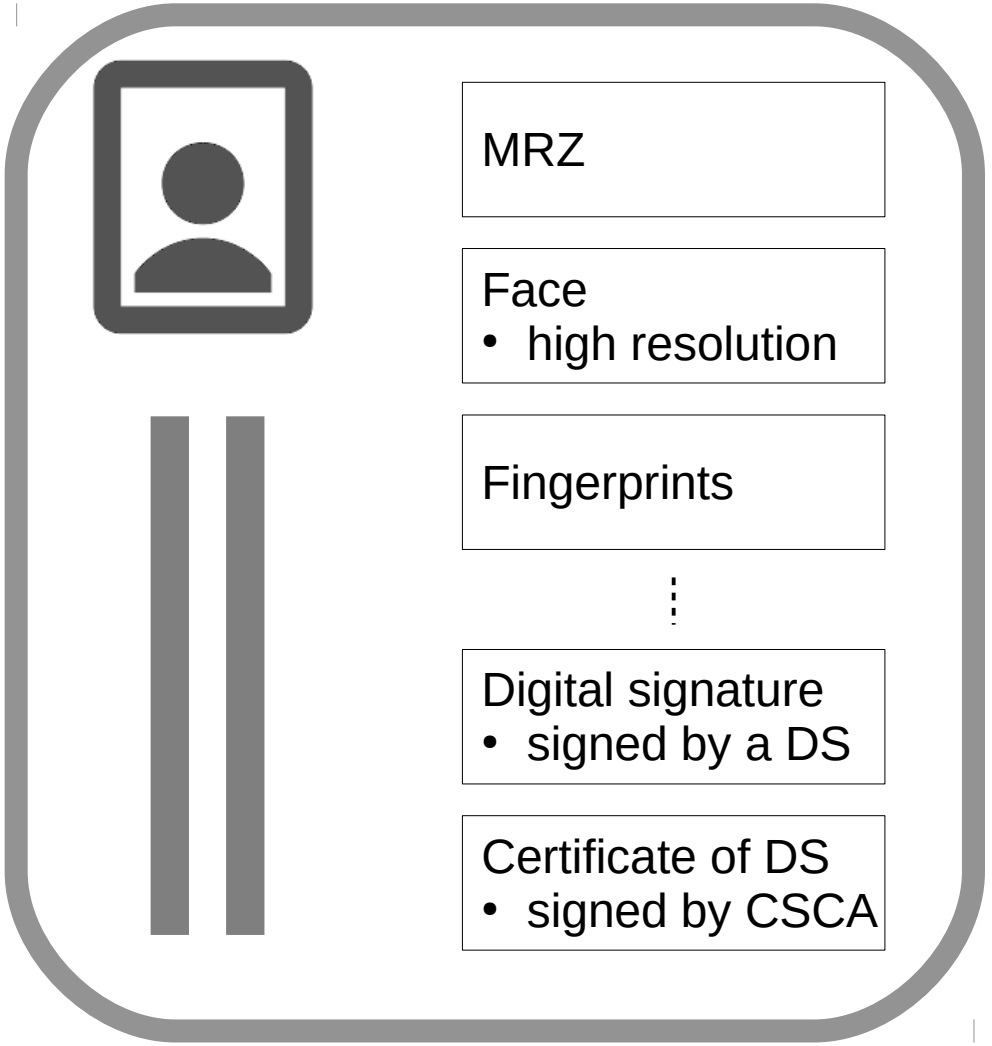
MRZ
- visible
- machine-readable
- name
- passport#
- ..



Chip
- processor
- storage

RFID
- contactless i/o
- ~10 cm distance

What digital data is in a EU passport?



Data is stored on chip
- customized format
- not a certificate

The signature is made by a Document Signer (DS)
- in a given country, there may be many DSs (Berlin, Hamburg, ...)

There is a certificate of the signer/DS
- issued by CSCA = Country Signing Certificate Authority of the country

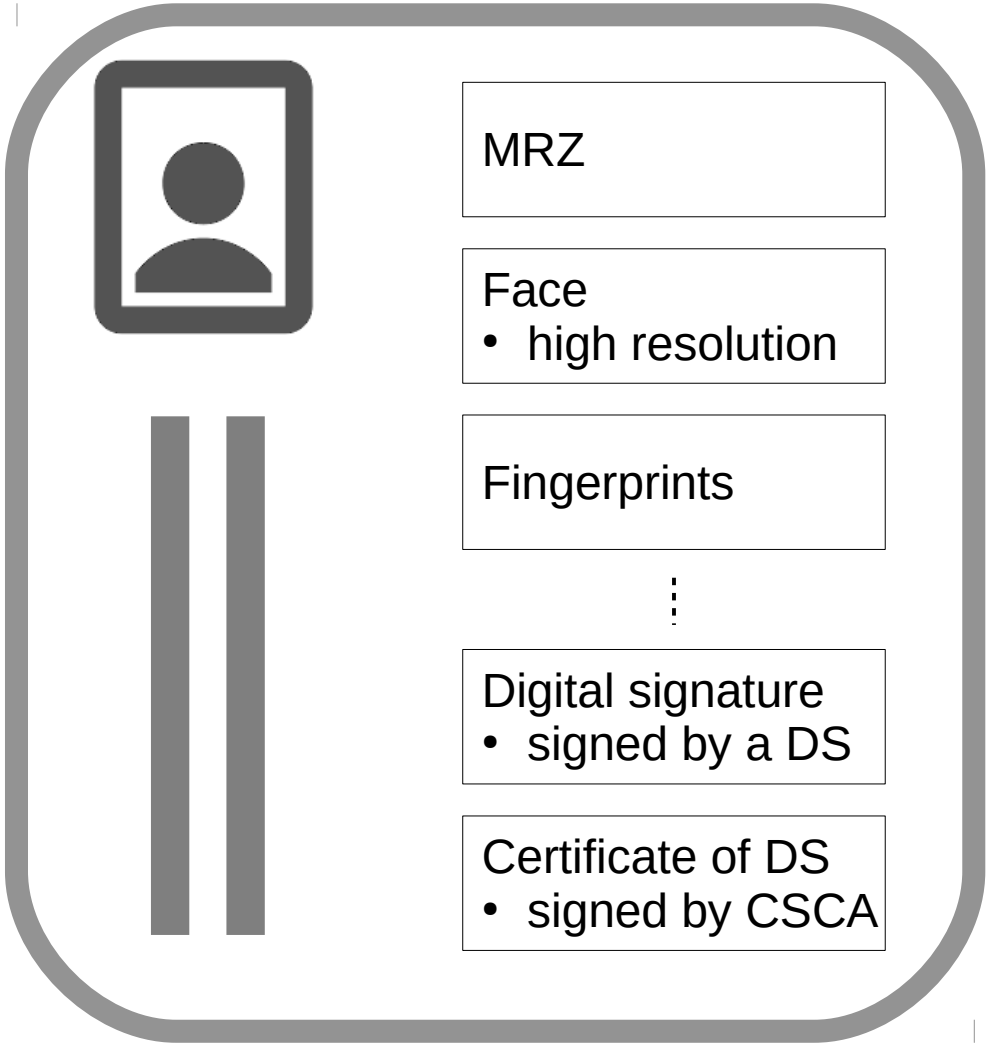
Aims and security goals (Hoepman, p 2)

1. A passport reader should verify itself first, so that only “trusted” parties get to read the information stored in the chip.
- Extended Access Control (EAC): readers must be authenticated
- As far as I understand, EAC must be implemented before fingerprints are read by passport readers

2. No identifying information should be released without the consent of the passport holder.
- Basic Access Control (BAC): release only to parties that have read the MRZ of the passport

3. The receiver of the information should be able to establish the integrity and authenticity of the data.

3. The receiver of the information should be able to establish the integrity and authenticity of the data



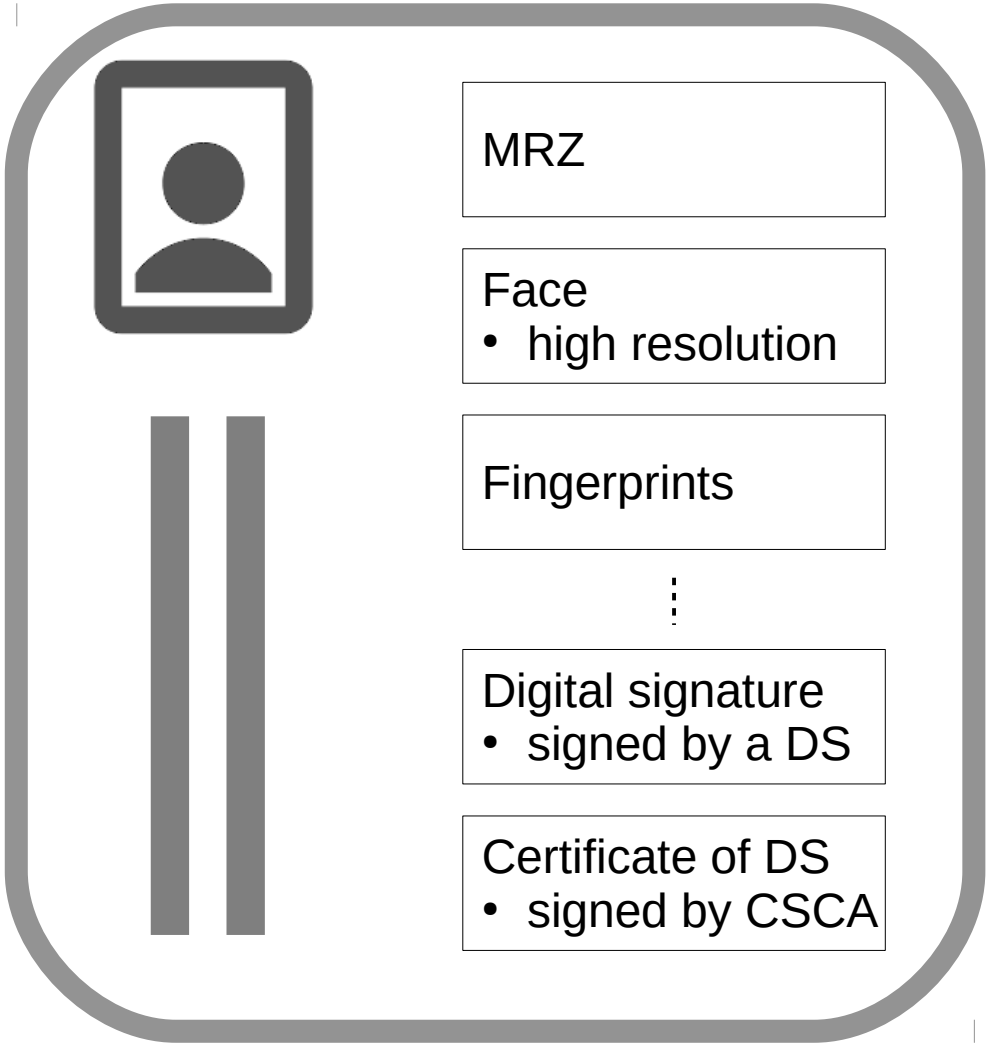
Passport reader reads all data (eventually)

How can it trust the data?

Recall that the data on the chip can be written by anyone

- (A) the Face data can be falsified
- (B) the DS signature can be falsified
- (C) the DS certificate can be falsified
- ..

3. The receiver of the information should be able to establish the integrity and authenticity of the data



Passport reader reads all data (eventually)

Then,

1. Verifies digital signature, using certificate of DS (in particular PU_{DS}) (establishes A and B)

2. Verifies certificate of DS using PU_{CSCA} (establishes C)

Therefore, all terminals (say, in Kastrup airport) must have public keys of all CSCAs of all 28 member states of the EU

Finally, the passport holders face is checked against the digital photo.

3. The receiver of the information should be able to establish the integrity and authenticity of the data

Conclusion

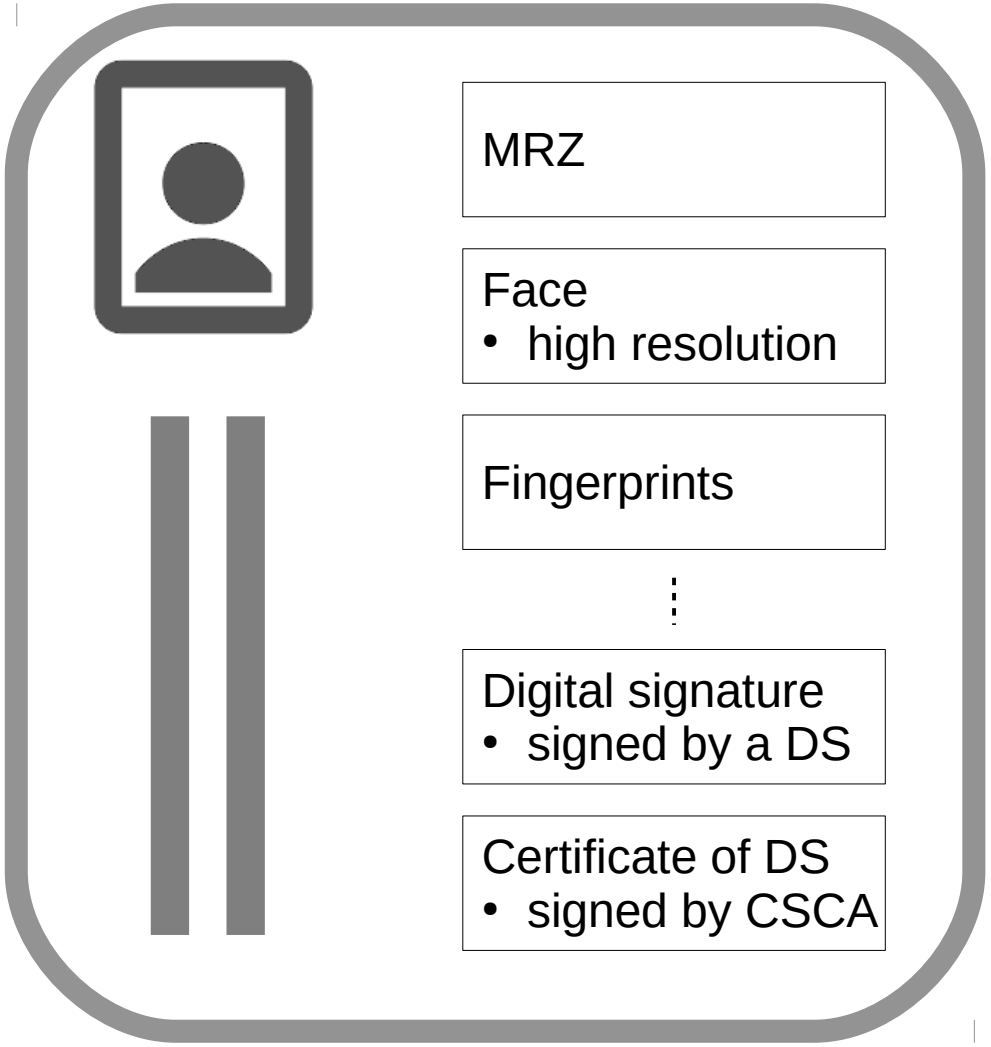
This goal is satisfied.

The goal requires a hierarchy:

Each country has a single CSCA (Country Signing Certificate Authority) which produces certificates of many DSs (Document Signers) in the country

A DS is a government institution, say a municipality, that issues passports.

Passive authentication vs. active authentication



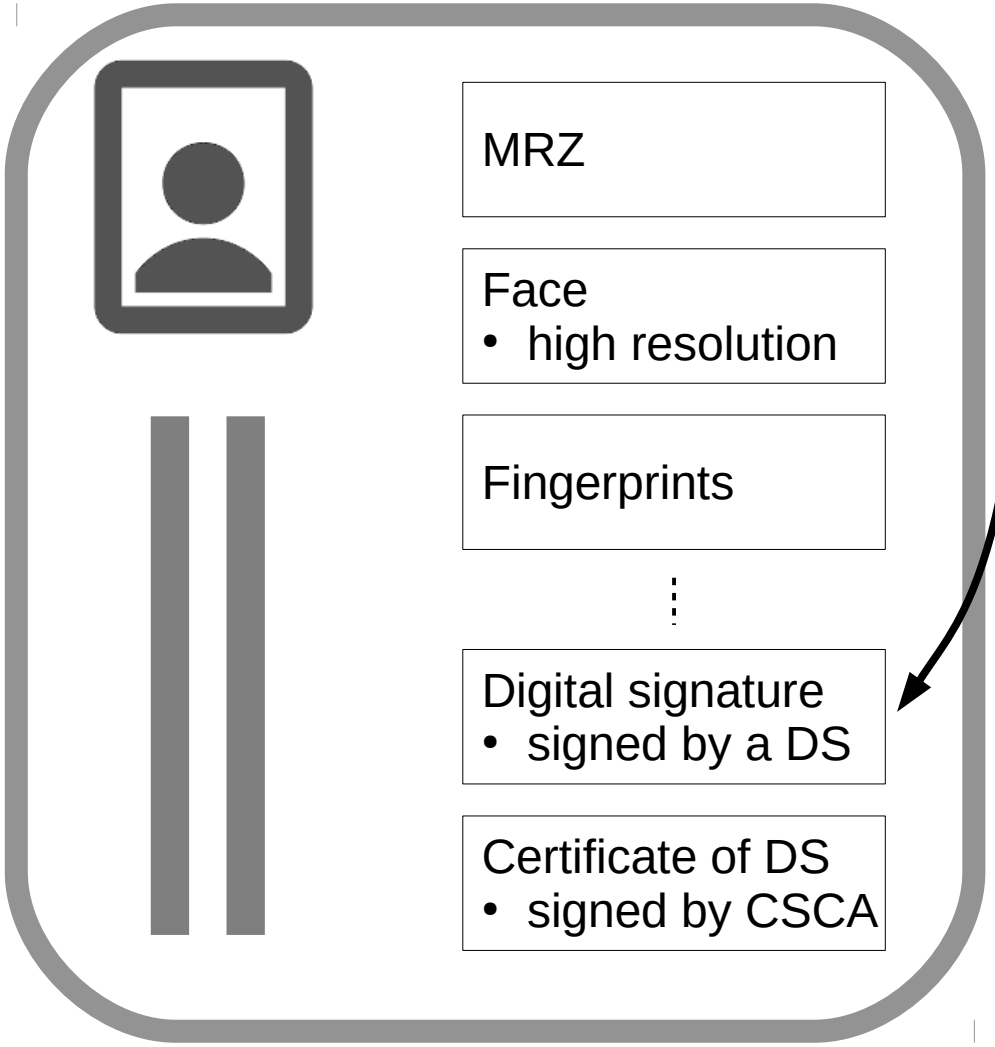
Passive authentication (Hoepman)

- refers to establishing integrity + authenticity (as described sofar)

Active authentication (Hoepman) refers to a stronger form of securing the passport's digital data

- the passport chip contains a private key
- I believe the point is that the private key can be written only once (when the passport is produced)

Technical notes to Hoepman



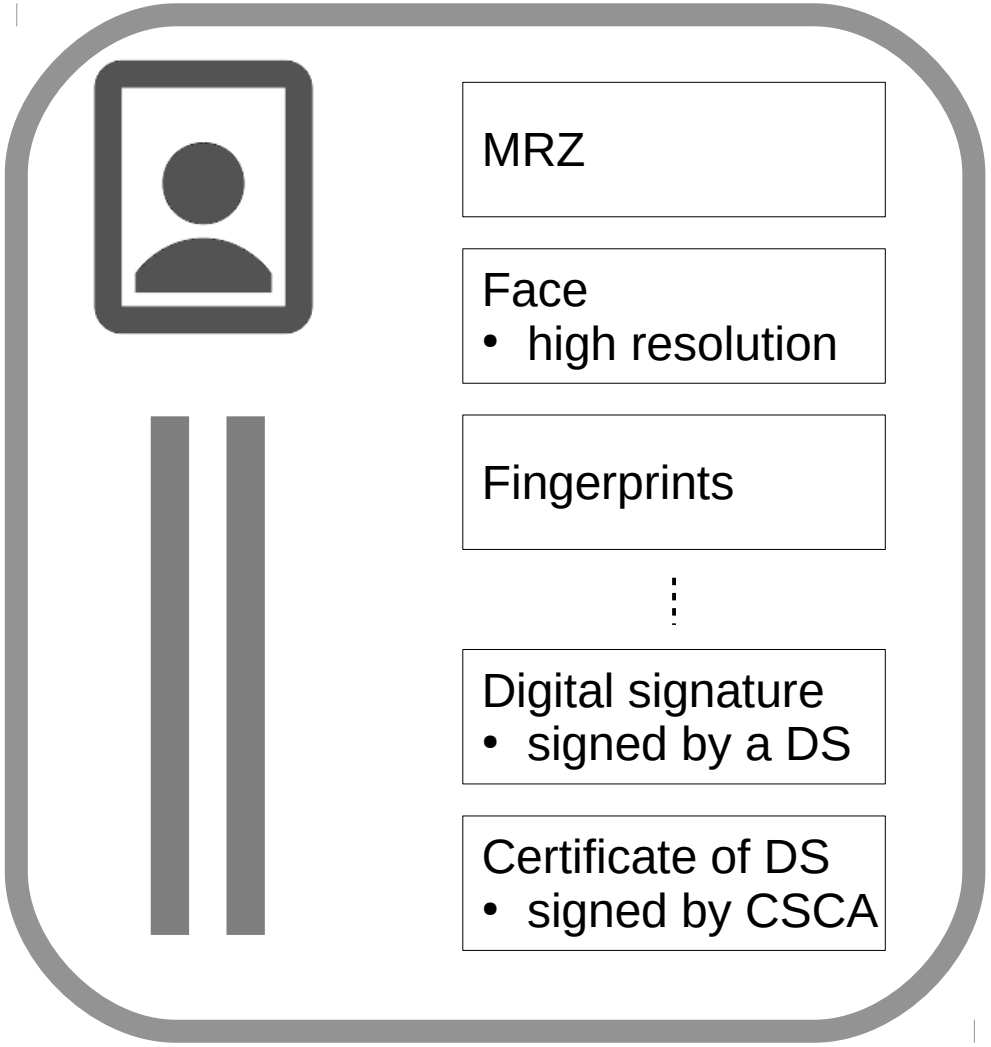
The digital signature (signed by a DS) is called SO_D by Hoepman.

Hoepman writes that the SO_D is signed "by the issuing country".

- actually, the signer is one DS (Document Signer) in the country, out of many DSs in the country.

2. No identifying information should be released without the consent of the passport holder.

- *Basic Access Control:* release only to parties that have read the MRZ of the passport



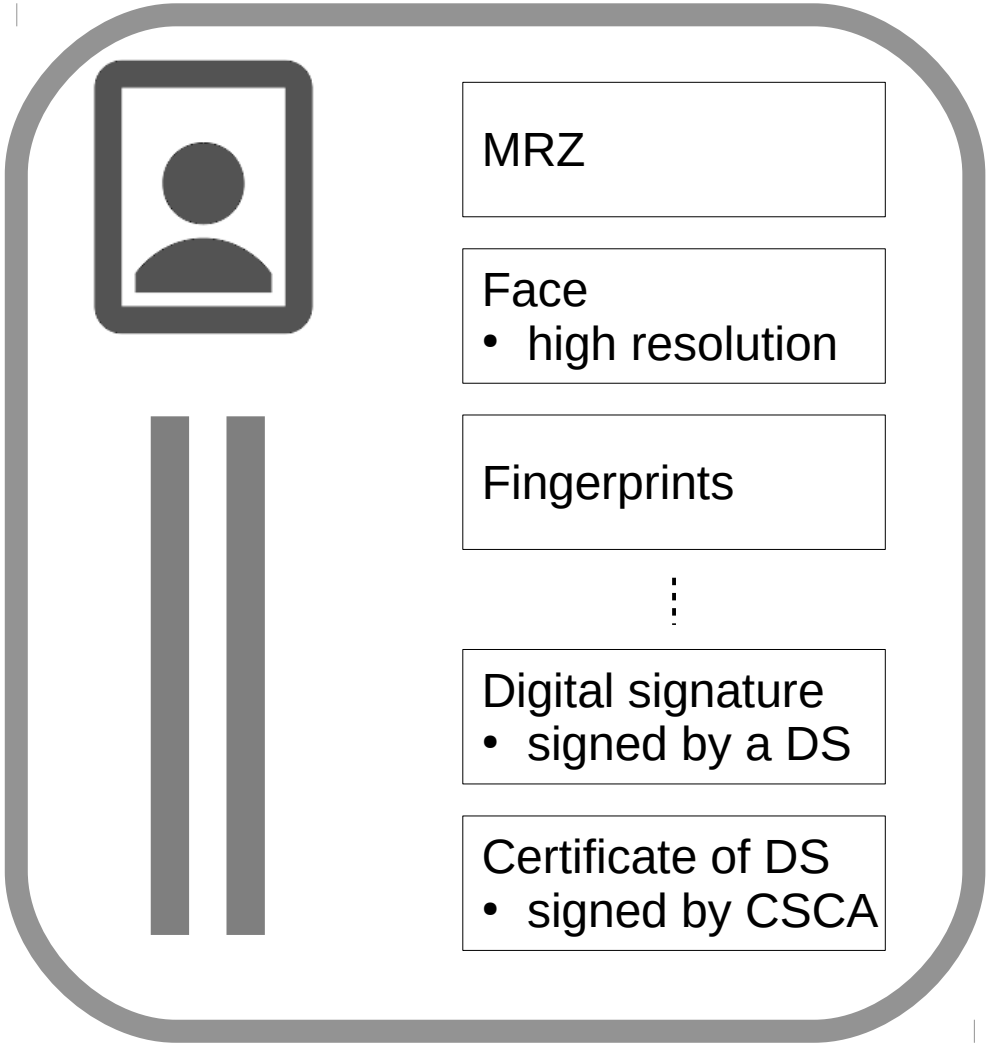
Before the chip on the passport releases data to the passport reader, the passport reader must prove that it has read the MZR

MZR = name + passport#
+ date of birth
+ expiry date

The passport reader computes an access key ($k_{IDF/ICC}$) based on the MZR

Hoepman's criticism:
- not enough different values of the key
- "not enough entropy in the MZR"
- less than 2^{80} values

2. No identifying information should be released without the consent of the passport holder.

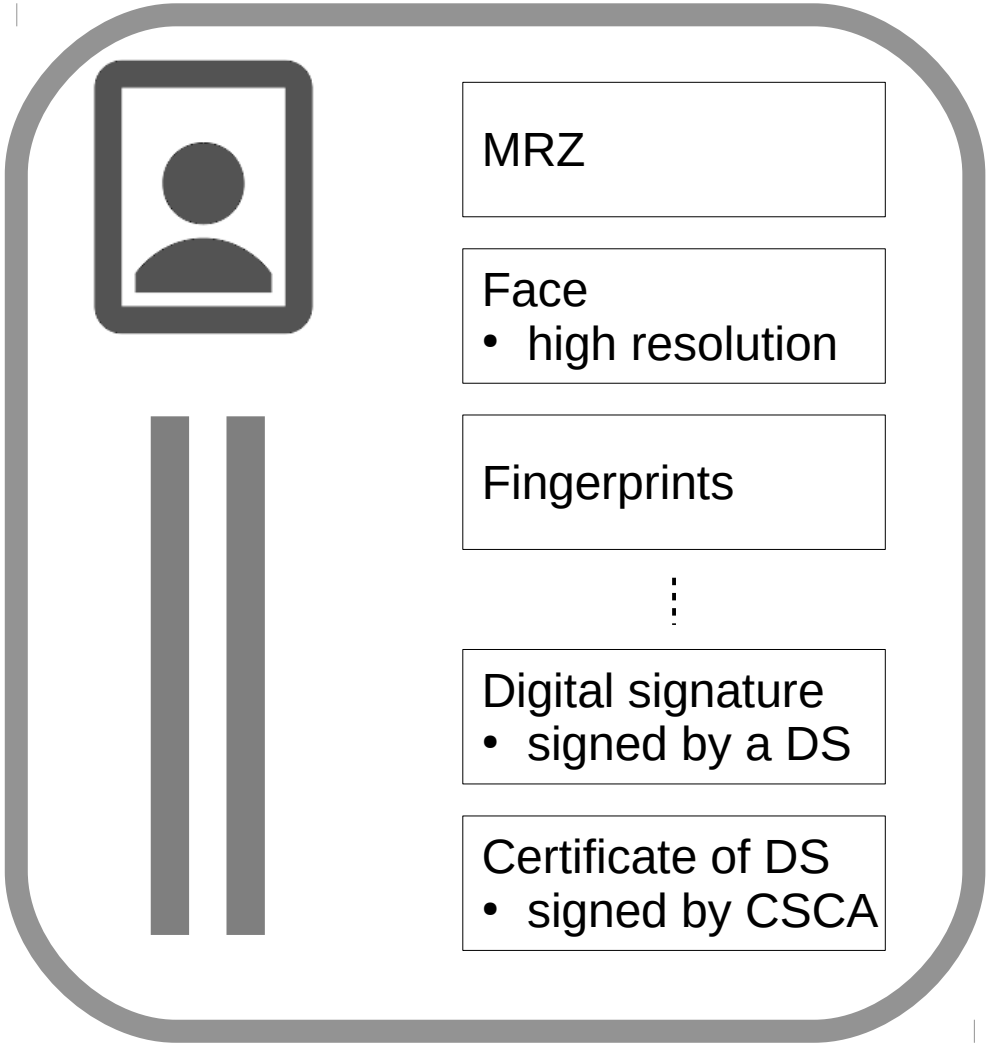


MZR = name + passport#
+ date of birth
+ expiry date

Date of birth
- ~ 2^{15} (100 years * 365 days)
- In fact only $2^{10,8}$ ~ 1.800 options guessing age within a 5 years margin

Hoepman concludes the total number of different MZRs is 2^{70} or even smaller.

2. No identifying information should be released without the consent of the passport holder.



Conclusion:

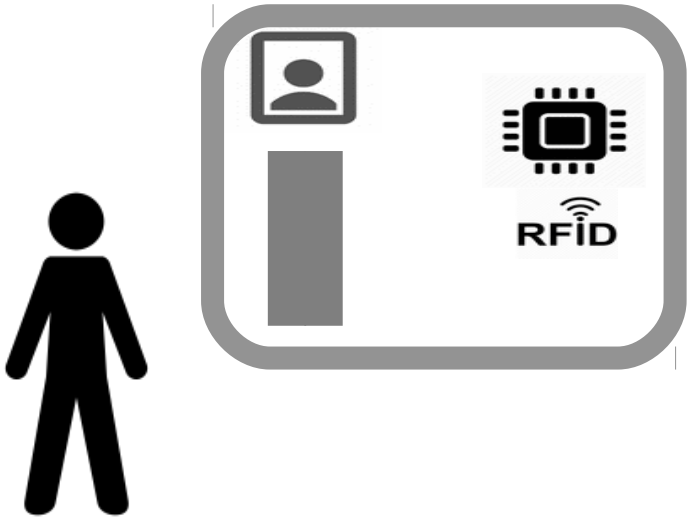
It may be possible for an eavesdropper to pretend to have read the MRZ, and so trick the passport to release its data.

However, the trick requires trying, say 2^{70} different key values.

(The underlying RFID protocol may be used in a way that makes it possible to identify a passport).

1. A passport reader should verify itself first, so that only “trusted” parties get to read the information stored in the chip

- Extended Access Control: readers must be authenticated



Extended Access Control

Root certificate (Germany)

Public key trusted by passport

Private key

Document Verifier (DV) Certificate

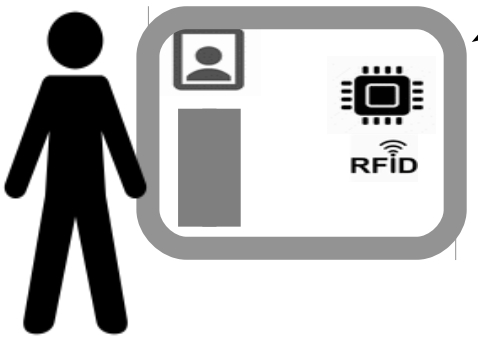
- Danish
- Germany-signed

Private key

Inspection System (IS) Certificate

Private key

IS proves possession of it's private key to passport chip

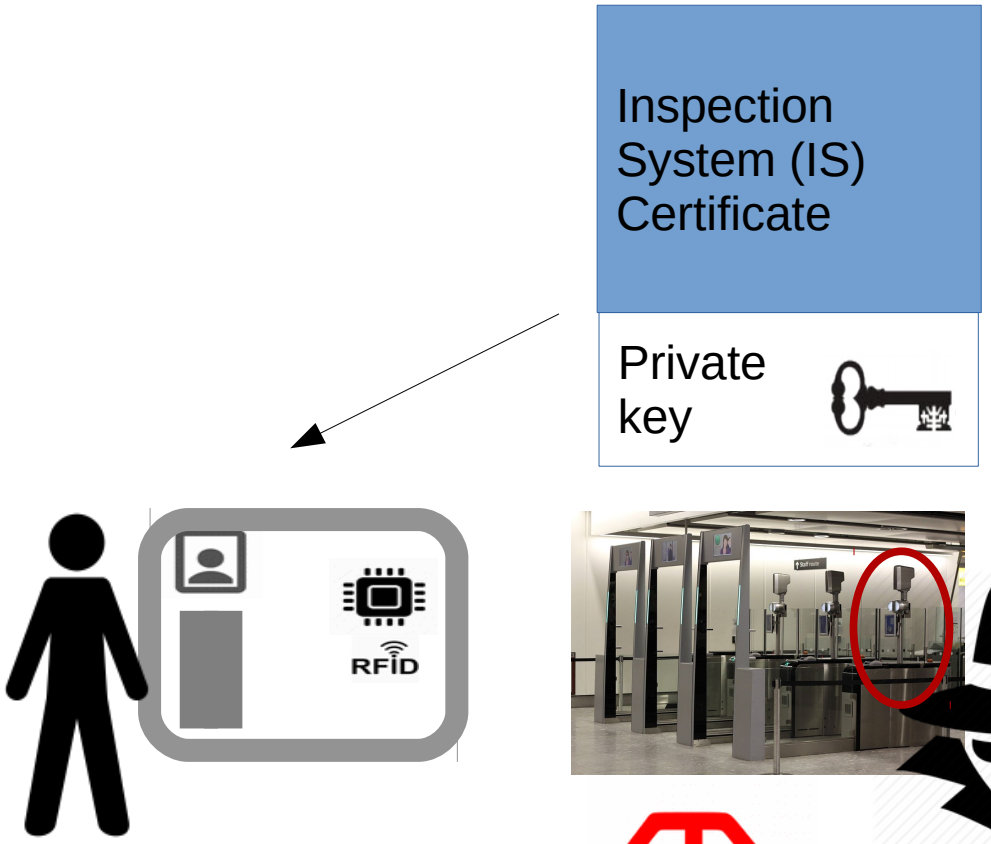


Issuing country (Germany)

German citizen

Inspecting country (DK)

What if an inspection system is stolen?



IS stolen
- now sits in a taxi
- and tries to read passport data

A certificate contains expiry date
- IS certificates are short-lived (eg., 3 months)

Passport chip checks that time is not past expiry date of IS certificate
say, expiry date is March 24th

How does the passport chip know the time?
- does not have a clock
- updates time each time it reads an IS certificate
- protects frequent travellers
- not others
- suppose on October 24th, last time the person travelled was Jan 24th