

Applications of cryptography

BUITA lecture 10 (ITS 3)
March 22nd, 2018

Niels Jorgensen
(nielsj@ruc.dk)

Learnings goals

Access control

- Know about authentication with passwords as well as with hashing, digital signatures and certificates.

GDPR, privacy

- Know about the new EU General Data Protection Regulation (GDPR).
- Be able to choose and use appropriate methods and techniques for privacy in context of specific it projects.

Usable Security

- Understand the challenges related to usable security.

Agenda



1. Technology: Access control with passwords and hashing

2. Technology: Access control with digital signatures & digital certificates

3. Law: EU's General Data Protection Regulation (including pseudonymization)

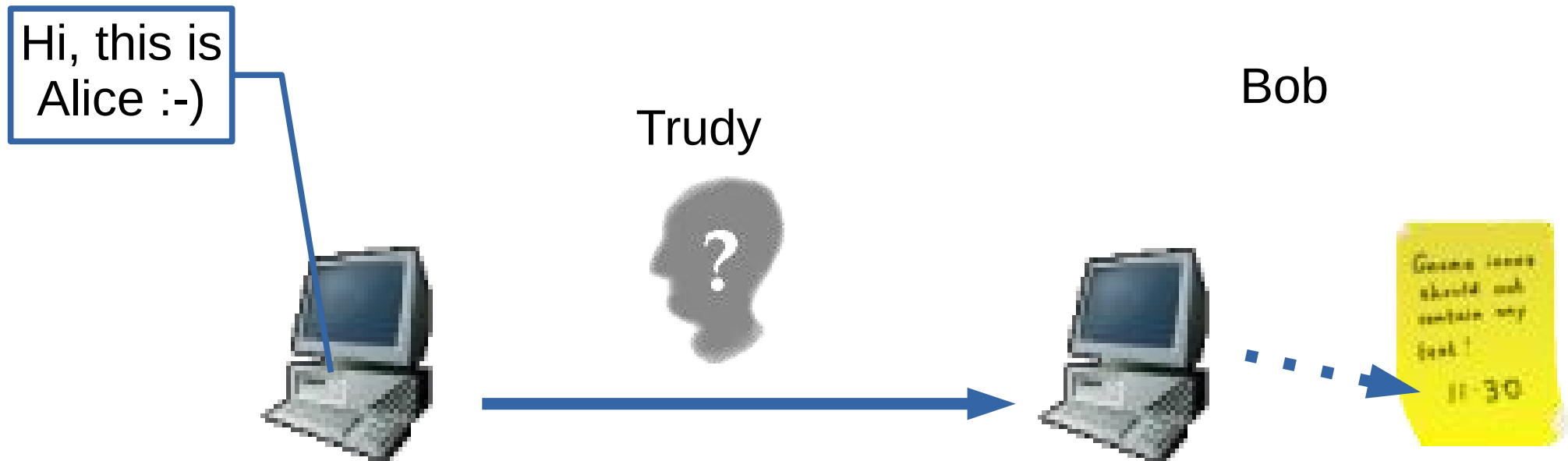
4. Usable Security (Why Johnny Can't Encrypt)

Access control: protection of stored data

Only legitimate users have access to data

Additional protection of stored data:

- encryption
- anonymization
- pseudonimization



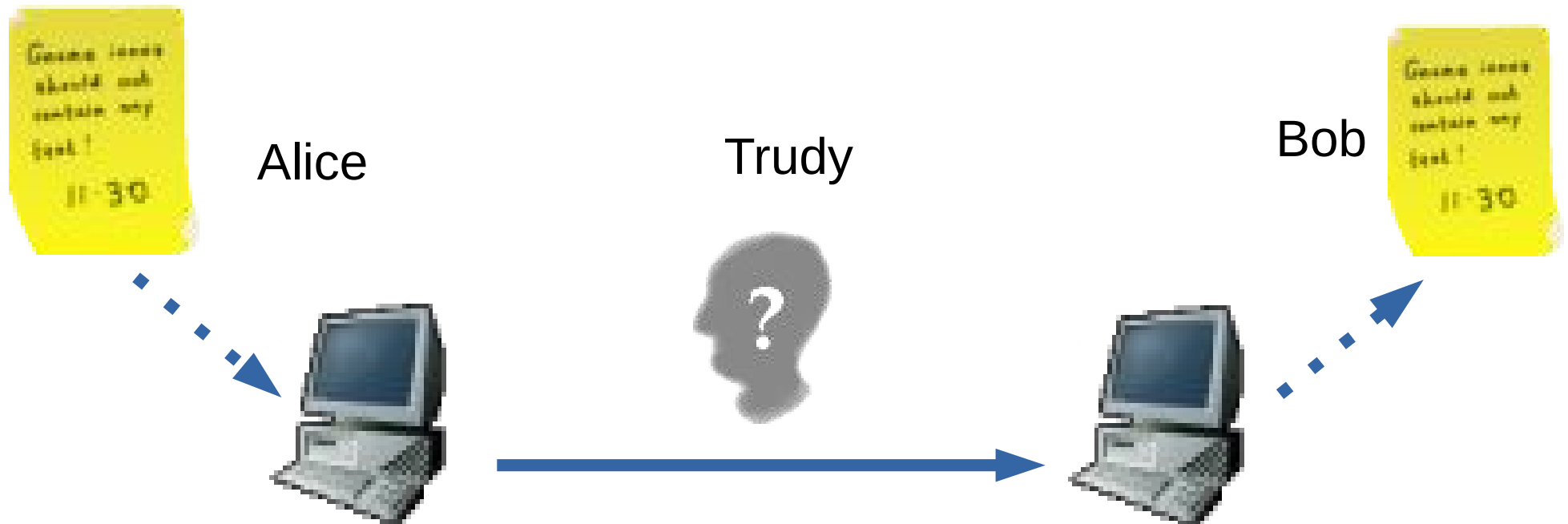
Data transfer - recap

We assume network is unprotected

- risk that Trudy reads or modifies Alices message to Bob

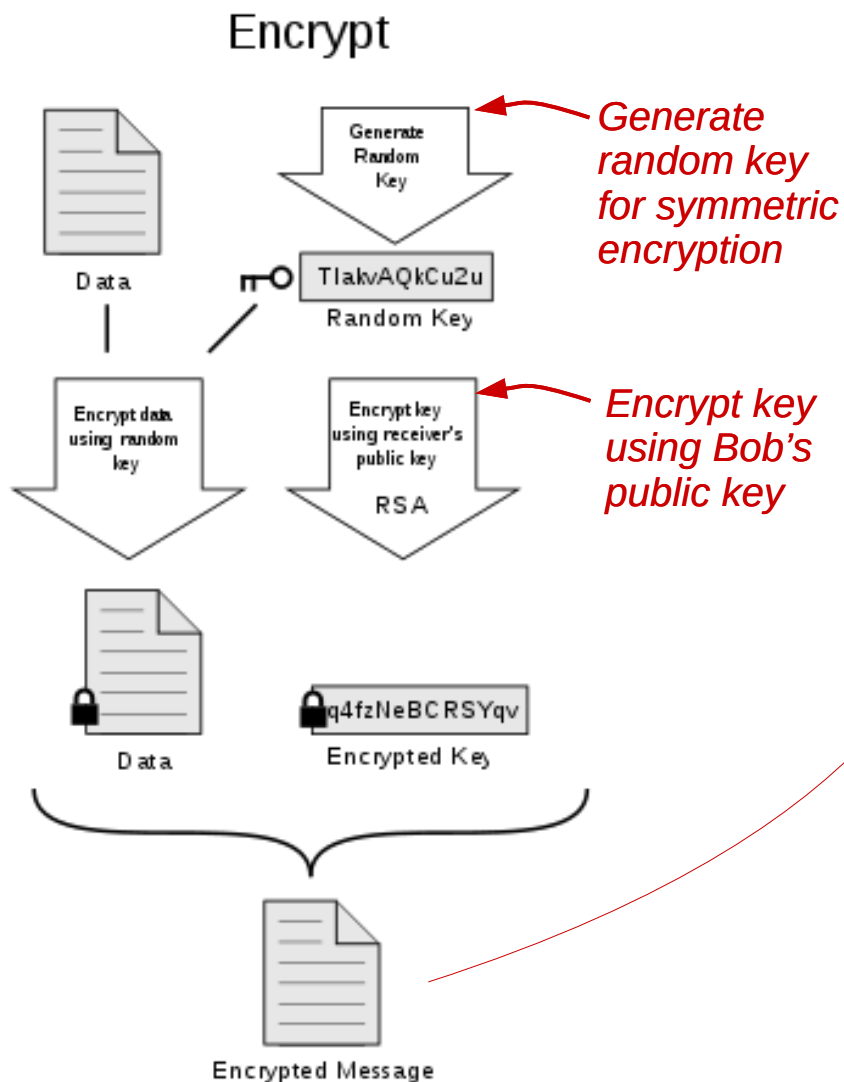
Any sensitive data must be encrypted when in transit

- symmetric encryption for the data itself
- asymmetric encryption used for exchange of symmetric key



Encryption of transmitted data (PGP)

Alice encrypts



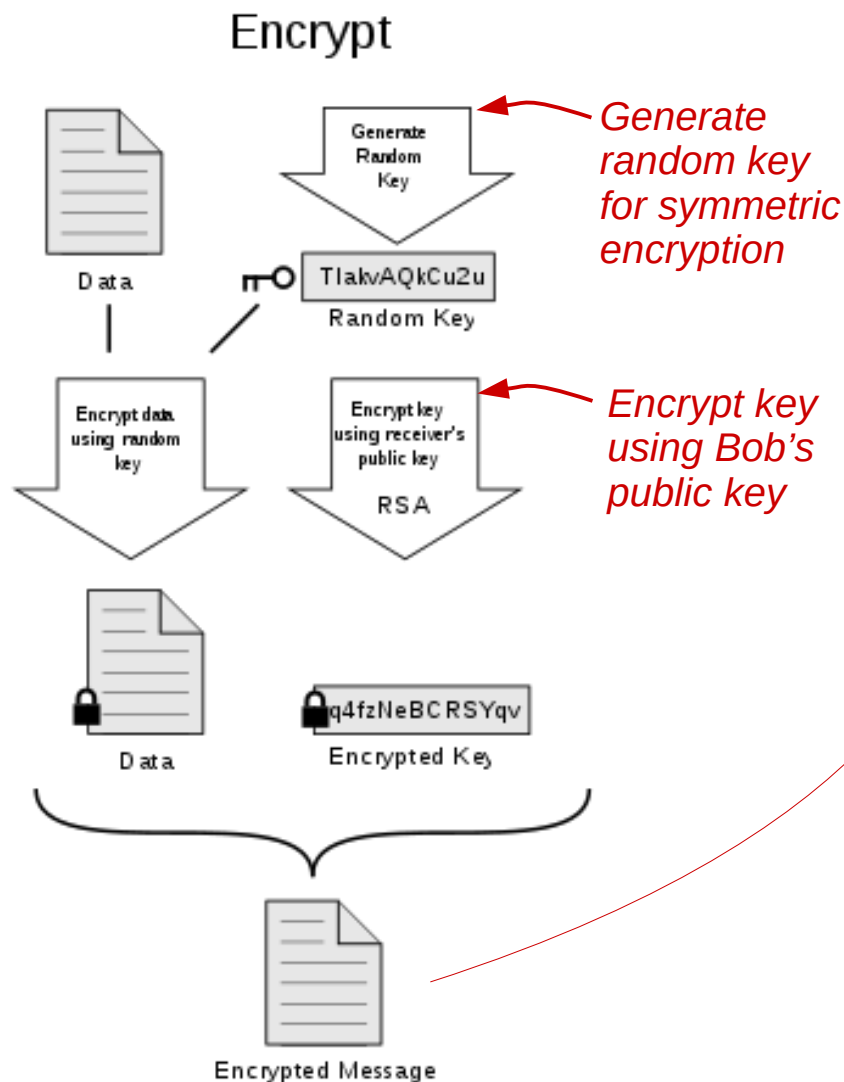
Exercise: How does Bob decrypt?

Decrypt

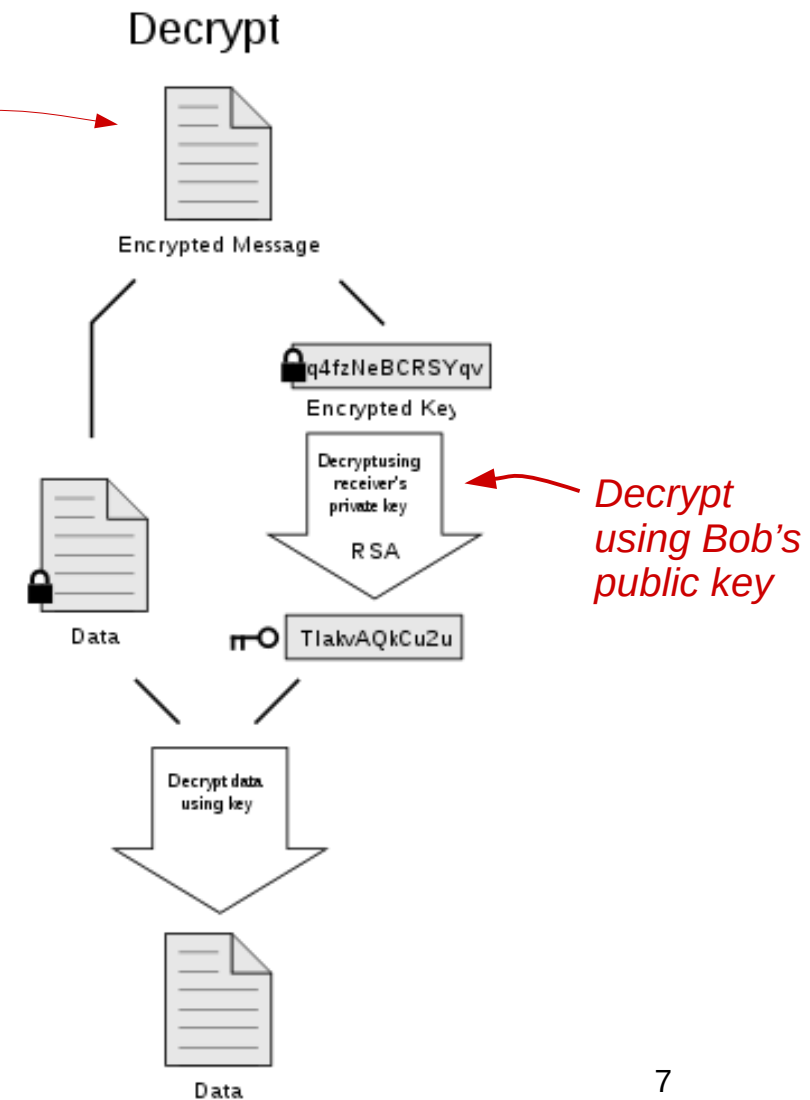


Answer: Encryption of transmitted data

Alice encrypts



Exercise: How does Bob decrypt?



Agenda



1. Technology: Access control with passwords and hashing

2. Law: EU's General Data Protection Regulation (including pseudonymization)

3. Technology: Access control with digital signatures & digital certificates

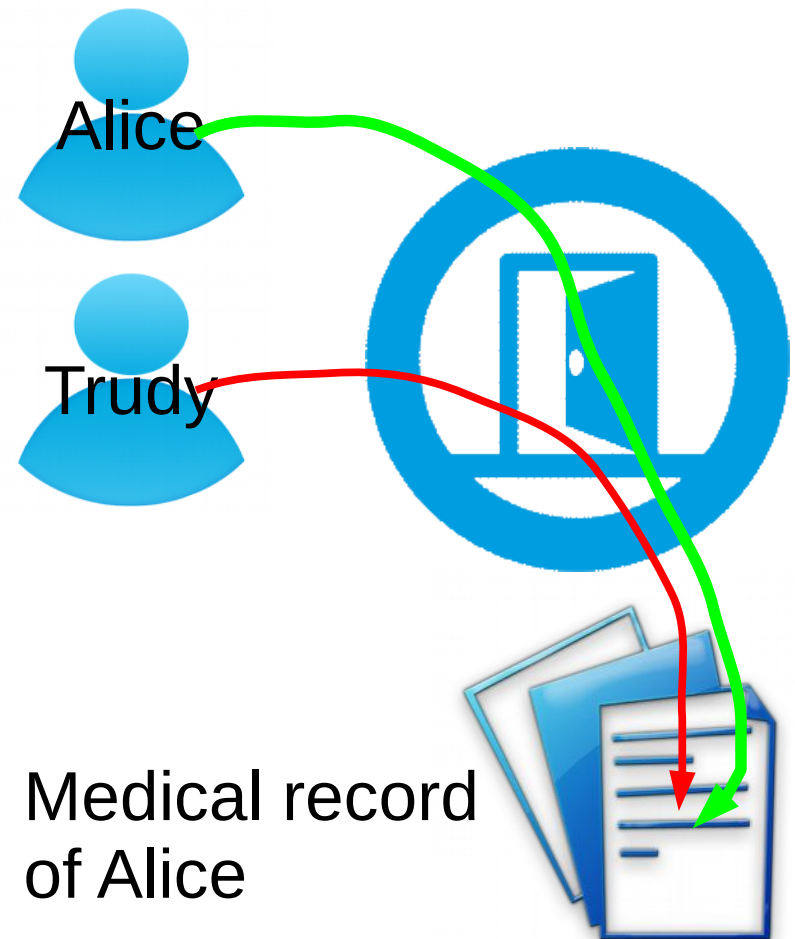
4. Usable Security (Why Johnny Can't Encrypt)

Access control using passwords

Suppose you work for a company that provides an internet-based service for employees or citizens

- sundhed.dk (“health.dk”)
- ..

How do you design access control?



Design of access control

Strong passwords

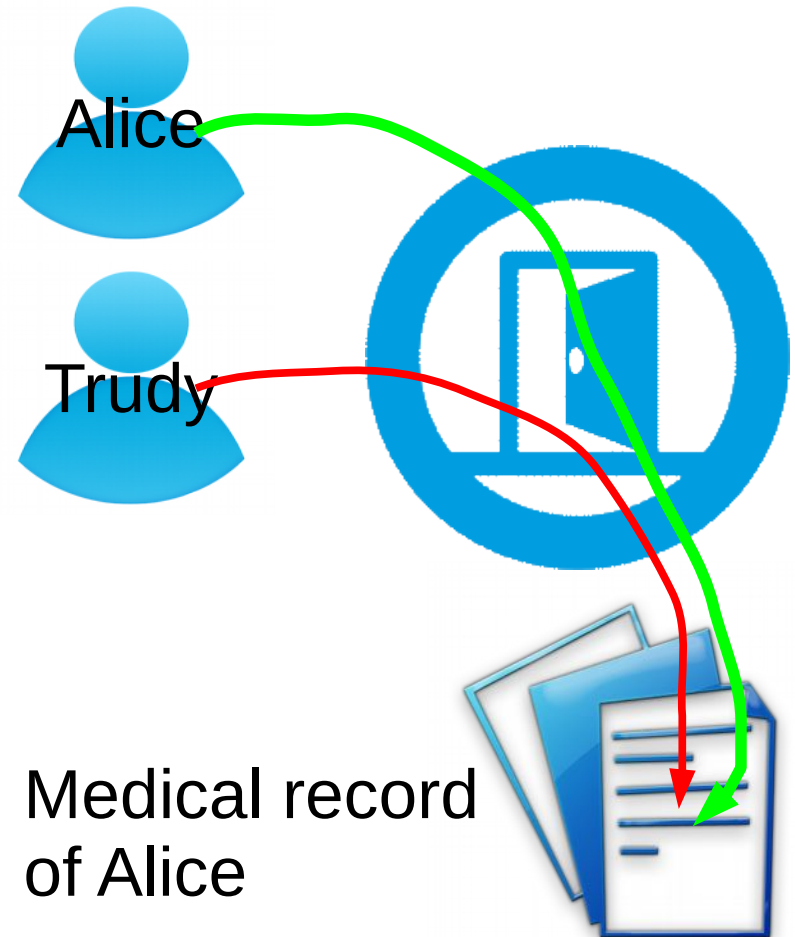
- protect against password guessing

All passwords stored in table

- compare at login

How to protect password table?

- hashing
- salting



Identification, authentication

Identification

- the user's *assertion* of who he or she is (eg., a username)

Authentication

- *proving* the identity of the user

Authentication by passwords

- “one factor-authentication”
- something you know (remember)
- password is provided by user and compared to a stored version

Password dilemmas:

- strength: strong vs. easy to remember
- storage: access for comparison vs. accessible for all

Strong passwords

Strong passwords protect against password guessing

Password guessing method #1: try all passwords

Exercise:

*Suppose a password consists of two characters,
and the alphabet has the letters: A, B and C.*

What is the total number of possible values?

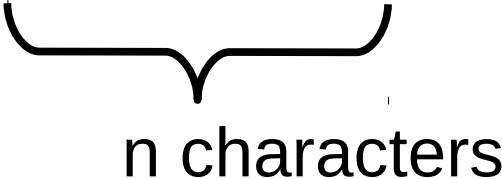
Answer: The total number of passwords

Suppose a password consists of two characters,
and the alphabet has the letters: A, B and C.

Then the total number of possible values is: $3 * 3 = 3^2 = 9$.

- AA, AB, AC, BA, BB, BC, CA, CB, CC

General formula:

- size of alphabet: S
 - number of characters: n
 - total number of passwords: $S * S * .. * S = S^n$
- 

Password guessing method #2: dictionary attack

Guess words in dictionaries, namelists, ..

- “alice”, “password”, “university”, “mail”, ..

Therefore passwords must have digits, in addition to letters.

Conclusion:

Strong (as at RUC):

- at least eight characters
- include lower case letters, upper case letters, digits (0-9)
- a123456A

Very strong

- at least ten characters
- include punctuation characters
- at least two of each type of character
- ab123456.,AB

Strong passwords: comparison with DES

Even strong passwords may not be protected against brute-force attacks

Therefore other measures are required

- an upper limit on the number of guesses
- ..

Exercise:

If we assume that a DES key (56 bit of information) is vulnerable to a brute force attack, is a strong password with eight characters also vulnerable? (assuming no other measures)

$$2^{56} = 72,057,594,037,927,936$$

4 billions, so a total of 72 million billions (Da: 72 millioner milliarder)

Hint: use <http://www.calculator.net/exponent-calculator.html>

Answer to exercise

No, eight characters do not provide protection against a brute force attack.

Eight characters, each a digit or a lower or upper case letter

One character:

- 10 digits + 26 lower case letters + 26 upper case letters
= 62 different values

Eight characters

- $62^8 = 218,340,105,584,896$ (two hundred thousand billions)
- That's much less than the DES key space of 2^{56}

Strong passwords - summary so far

Strong passwords are

- necessary to prevent password guessing (dictionary attacks)

But strong passwords are:

- difficult to remember
- and do not protect against brute-force search

So additional security is required

- blocking brute-force search
- password policies (eg. changing passwords frequently)
- for even better security, *two factor-authentication* should be considered (as in NemID)
 - something you *know* (normal password)
 - + something you *posses* (NemID card)
- *also the stored password file must be protected*

Stolen password files

Password file (naive)

User	Password
Alice	a123456A
Bob	b456890B



Used to verify a user's password at login

Password files must be protected

- system administrators should not know the passwords
- if intruders gain access, passwords should still be protected

Dropbox 2016

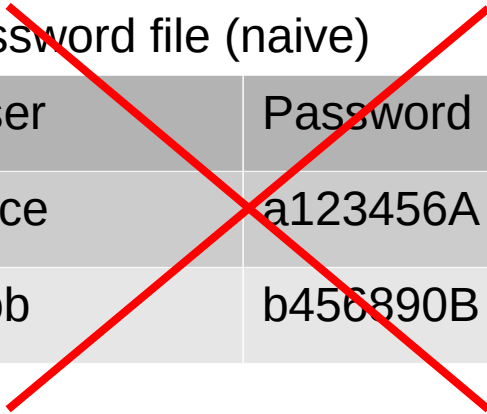
- password file with 68 mill. users stolen

Twitter 2016

- password file with 32 mill. users stolen

Passwords protected by hashing

Password file (naive)



User	Password
Alice	a123456A
Bob	b456890B



Password file with hashed passwords

User	Hashed passwd.
Alice	3X€!BXY7
Bob	Y4KUI??X

Hash function:

- a123456A -> 3X€!BXY7
- B456890B -> Y4KUI??X

Requirements:

- “Practically impossible” for Trudy, with file, to infer passwords
 - no method significantly better than brute force
 - so resembles requirements for encryption
- But we don't need keys / decryption (we can hash at every login)
- Also we don't want keys (they could be stolen)

ICO paper: Encryption and hashing

Encryption, the brute-force argument about infeasibility:

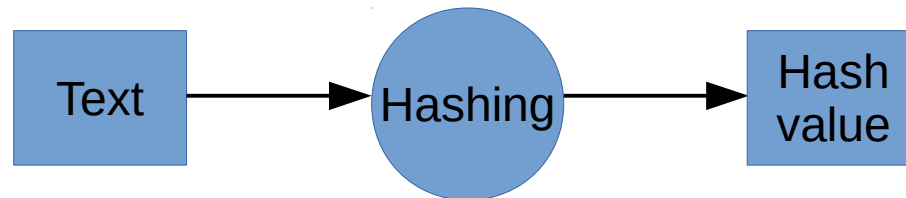
“Whilst it is possible to attempt decryption without the key (by trying all possible keys in turn), in practical terms it will take such a long time to find the right key (ie. many million years) that it becomes effectively impossible” (p4)

Hashing (p12)

“Hashing is a technique that generates a fixed length value summarising a file or message contents.”

Also, for a good hash function, it is not possible to guess the password that hashes to a given value.

Cryptographic hash functions



A hash value is a “fingerprint” of a text

- small
- unique for the text

Full set of requirements for cryptographic hash functions

- variable input length
 - fixed output length (ie., 128 or 256 bit)
 - reasonable speed (about the same as symm. encryption)
 - pre-image resistance (“one-way function”)
 - second pre-image resistance
 - collision resistance
- Easy* {
- Difficult* {

Exercise: Cryptographic hash functions

Propose a hash algorithm for passwords

in practice,
this would
be too short!

- must meet the three easy criteria
- fixed output: two characters (2 bytes, 16 bits)
- hash Alice and Bob's passwords

Show how to break it (find pre-image of any hash)

Easy

- variable input length
- fixed output length (ie., 128 or 256 bit)
- reasonable speed (about the same as symm. encryption)

Difficult

- pre-image resistance (“one-way function”)
- second pre-image resistance
- collision resistance

Answer: Cryptographic hash functions

Propose a hash algorithm for passwords
must meet the three easy criteria

- fixed output: two characters (2 bytes, 16 bits)
- hash Alice and Bobs passwords

Show how to break it (find pre-image of any hash)

Proposal: hash value is first character and last character

- a123456A -> aA
- b456890B -> bB

How to break this method:

- given a hash value, say 2x
- a pre-image is 2x, or 2ax, or 2aax, etc.

Defining a cryptographic hash function is possible

- but difficult

A file with hashed passwords of many users is stolen

Passwords threat model A:

Attacker wants to find the password of a *specific user*, such as a system administrator.

Time-consuming (days or years).
Perhaps no guarantee for the attacker.

Protection

- use a very strong password
- 10-12 characters, with digits, lower/upper case

A file with hashed passwords of many users is stolen

Passwords threat model B:

Attacker wants to find the password of *some user*, not a particular user.

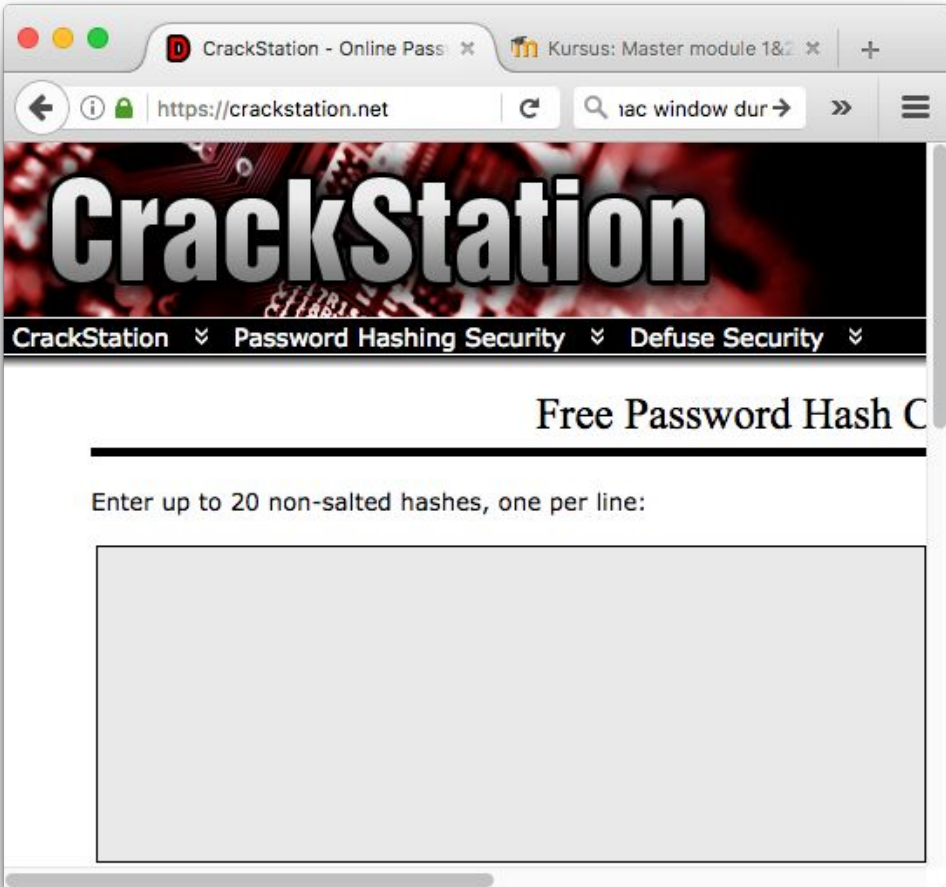
Password files with strong passwords are vulnerable to this type of attack, which may use a pre-computed “rainbow table”.

- table will find poor and some strong passwords (but not very strong p.)
- defend by using a “password salt”.

Rainbow tables are online

crackstation.net

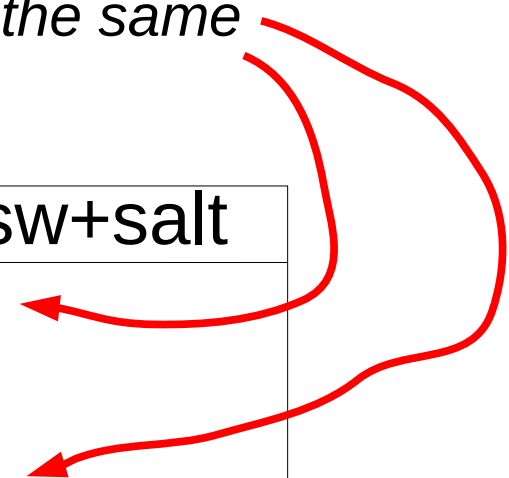
- pre-computed hash values
- hash algorithms: SHA-1, SHA-512
- of many, many passwords



Salt: protection against pre-computed password tables

Password file with salts
(data in parenthesis is not in file)

*different hash value
even though password
is the same*



user	(passw.)	salt	hashed passw+salt
alice	(aaaaaa)	ab	2d5d3e44
bob	(qwerty)	2d	2d46e346
peter	(aaaaaa)	3f	e30f4d27

Now a precomputed password table is useless.
Attacker must now do brute-force search on each
password + salt

Note: salt can be read from password file,
and is used by the program that authenticates the user. 27

Agenda

1. Technology: Access control with passwords and hashing



2. Technology: Access control with digital signatures & digital certificates

3. Law: EU's General Data Protection Regulation (including pseudonymization)

4. Usable Security (Why Johnny Can't Encrypt)

Cryptographic hashing (continued)

Hashing also useful with transmitted data

- for protecting the integrity of a message
- “message has not been changed”

e-banking:

- “Hello e-bank, I’m Alice, I wish to transfer 1,000 USD to Bob”
- (not to Trudy)

File download, integrity checking:

- A website with cryptographic algorithms in Java:
- “crypto-137.zip” has message digest (SHA-1):
ed737ef6 a242b67f d0d8d5c1 c13e0670 37eb5301

Method:

- Alice computes message digest (hash value)
- Alice sends message + message digest
- Bob computes message digest, if same as received, message must be OK
- (note importance of “second pre-image resistance”)

Message digest (produced by hashing)

From the creation of PGP (Pretty Good Privacy), a public-key encryption software package for the protection of electronic mail. Since PGP was published, documentation as the year in June of 1993, it has spread organically all over the world, and has since become the de facto worldwide standard for encryption of e-mail, sending numerous arbitrary e-mails along the way. For these years I was the target of a national investigation by the FBI Criminal Service, who surmised that there were broken, when PGP spread outside the US. That investigation was closed without indictment in January 1996.

Computers were developed in secret back in World War II mostly to break codes. Ordinary people did not have access to computers because they were rare in number and too expensive. Some people speculated that there would never be a need for more than half a dozen computers in the country, and assumed that ordinary people would never have a need for computers. Some of the government's attitude toward cryptography today were formed in that period, and because the old attitudes toward computers were. Why would ordinary people need to have access to good cryptography?

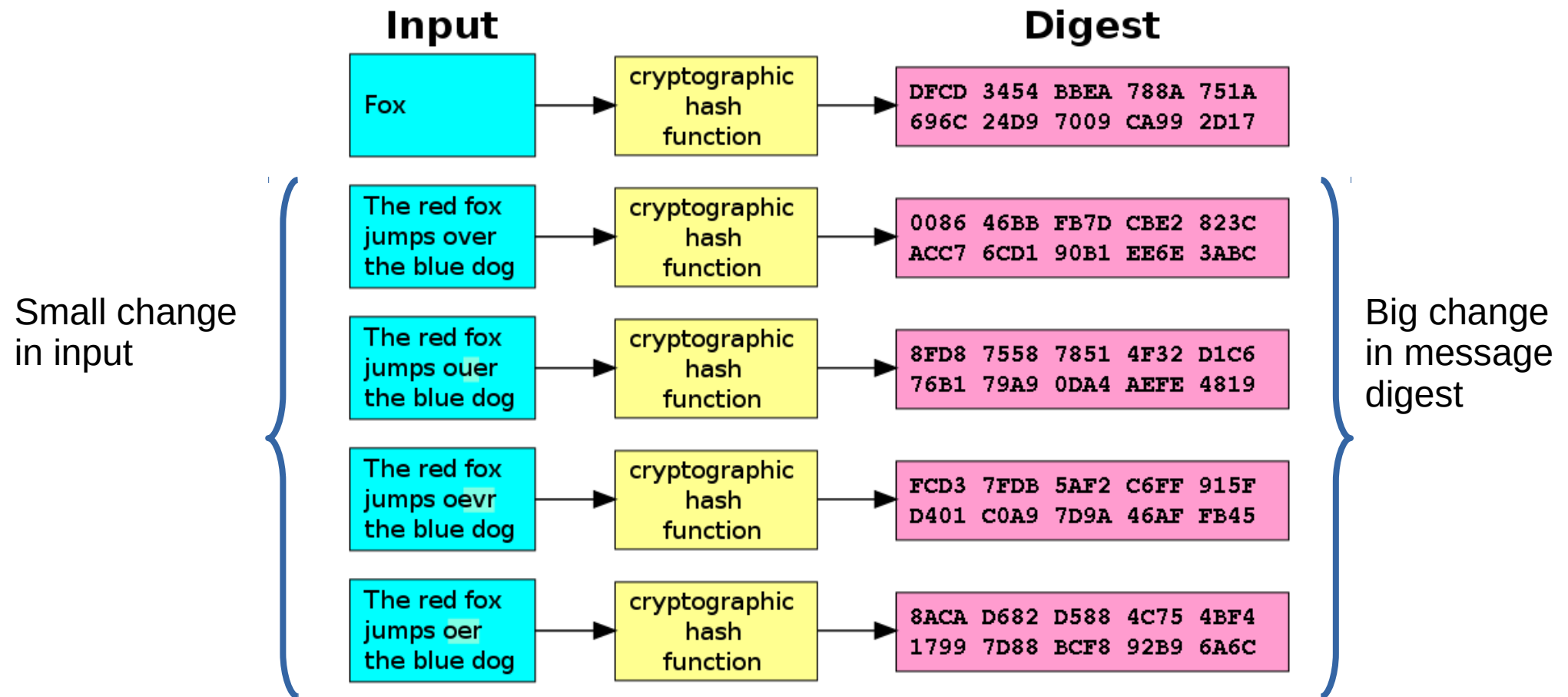
Hash

Message
Digest

Exercise:

If there is a small transmission error, where, say, only one bit or character is changed, would that always be detected by the receiver? (when the receiver verifies the message digest). Or, could there be a transmission error that changed *both message and message digest*, so that the changes cancelled each other out?

Hint: “Avalanche” effect of cryptographic hash functions



Message digest (produced by hashing)

From the creation of PGP (Pretty Good Privacy), a public-key encryption software package for the protection of electronic mail. Since PGP was published, documentation is the same in June of 1993, it has spread organically all over the world, and has since become the de facto worldwide standard for encrypted e-mail, running on numerous arbitrary systems along the way. For these years I was the target of a national investigation by the FBI Criminal Service, who surmised that there were broken, when PGP spread outside the US. That investigation was closed without indictment in January 1996.

Computers were developed in secret back in World War II mostly to break codes. Ordinary people did not have access to computers because they were rare in number and too expensive. Some people speculated that there would never be a need for more than half a dozen computers in the country, and assumed that ordinary people would never have a need for computers. Some of the government's attitude toward cryptography today were formed in that period, and because the old attitudes toward computers. Why would ordinary people need to have access to good cryptography?

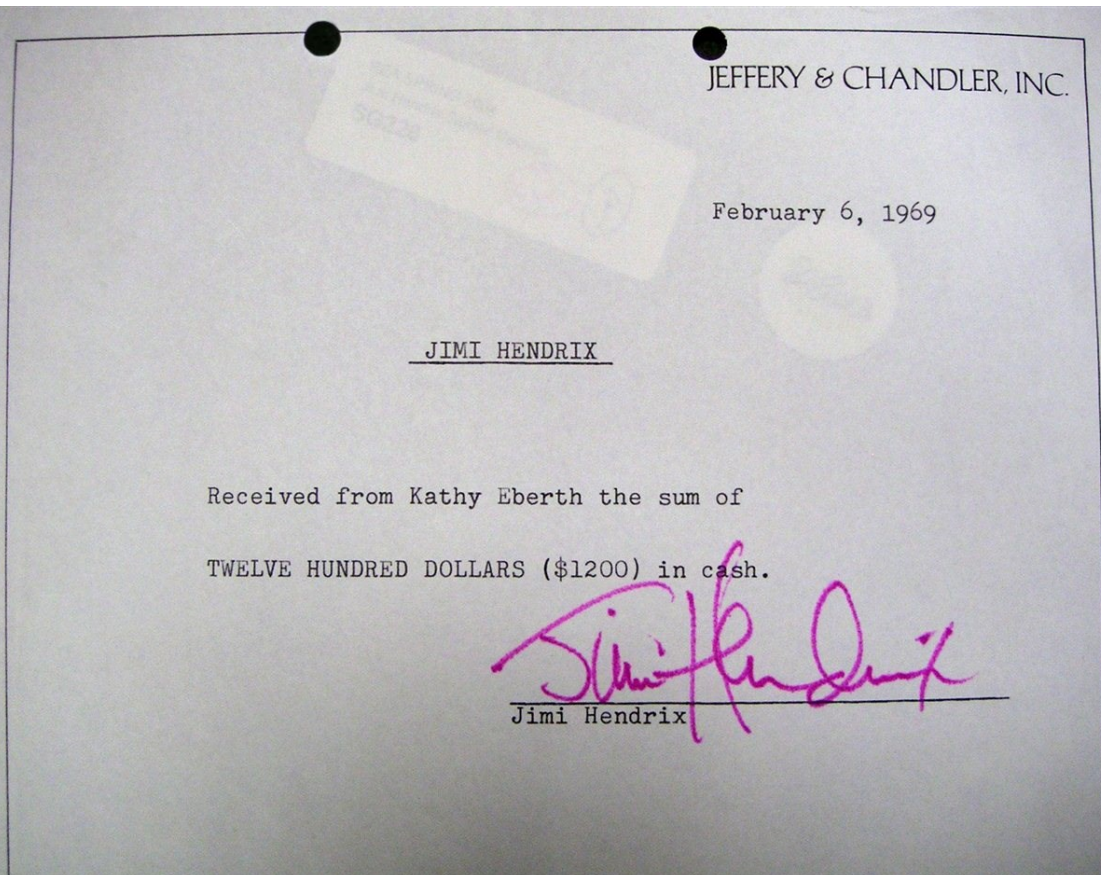
Hash

Message
Digest

Answer:

No, it is extremely unlikely that a transmission error would change the message and not be detected. It could not happen in practice. This is because both the message and the message digest should have been changed, and in a corresponding manner, and this would require changing a very large portion of the message digest (approximately half of the bits in the message digest).

Digital signatures resemble paper-based signatures



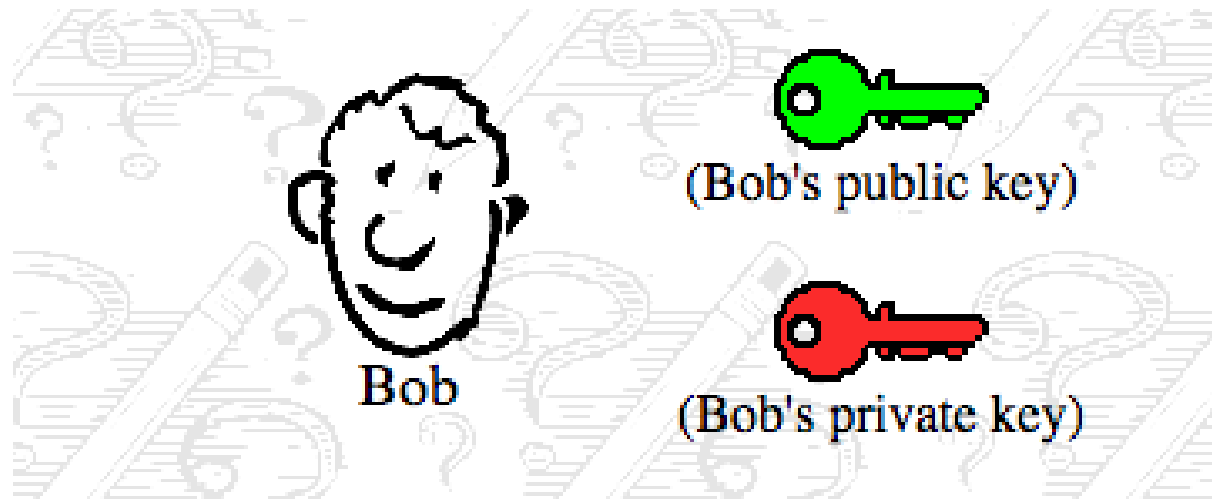
Properties of digital and physical signatures

- originality of document is protected
- authenticity of signer is protected
- links document and signer

Digital signatures

- created using signer's private key
- confirmed using signer's public key

Public key pair



(Thanks to David Youd for images and a very pedagogical explanation)

Encryption, transmission, decryption



"Hey Bob, how about lunch at Taco Bell. I hear they have free refills!"

Encrypt with
Public Key

HNFmsEm6Un
BejhhyCGKOK
JUxhiygSBCEiC
0QYIh/Hn3xgiK
BcyLK1UcYiY
lxx2lCFHDC/A



HNFmsEm6Un
BejhhyCGKOK
JUxhiygSBCEiC
0QYIh/Hn3xgiK
BcyLK1UcYiY
lxx2lCFHDC/A

Decrypt with
Private Key

"Hey Bob, how about lunch at Taco Bell. I hear they have free refills!"

(Please note: in practice, asymmetric encryption/decryption is not for large messages, because asymmetric encryption/decryption is time-consuming).

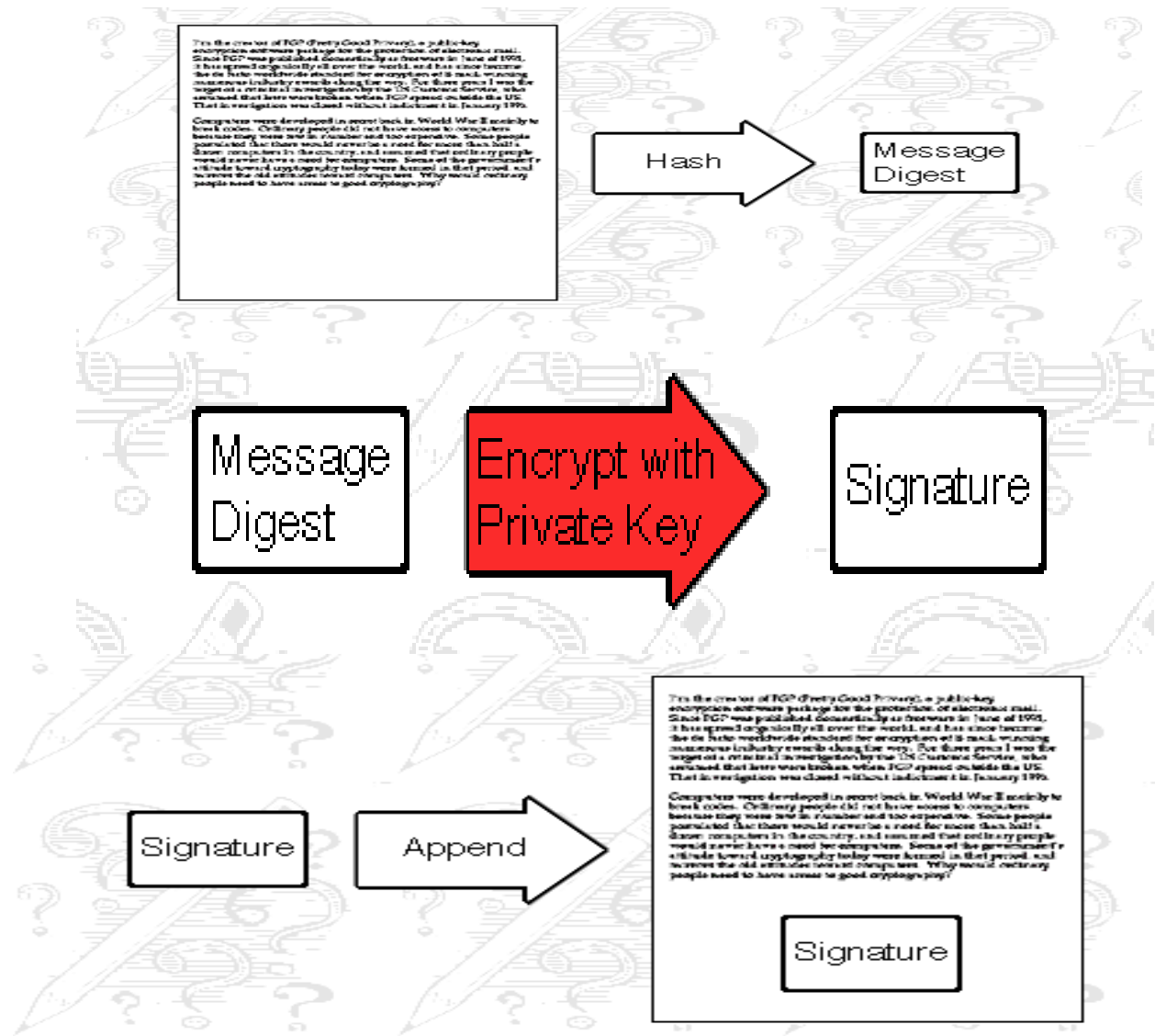
Digital signature produced by sender

Bob (signer, sender)

1. Compute message digest

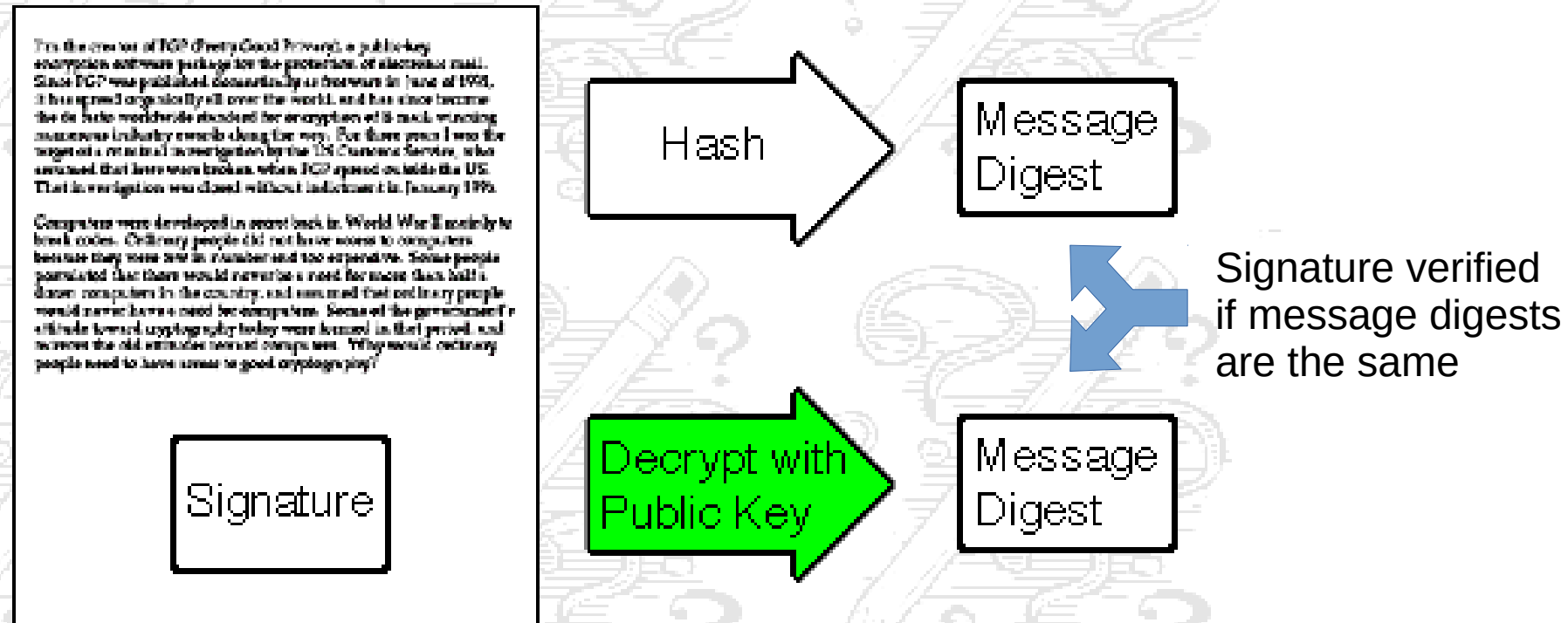
2. Compute signature with private key

3. Append signature to document, then send.



Receiver's verification of signature

Pat (receiver)



Receiver decrypts with sender's public key

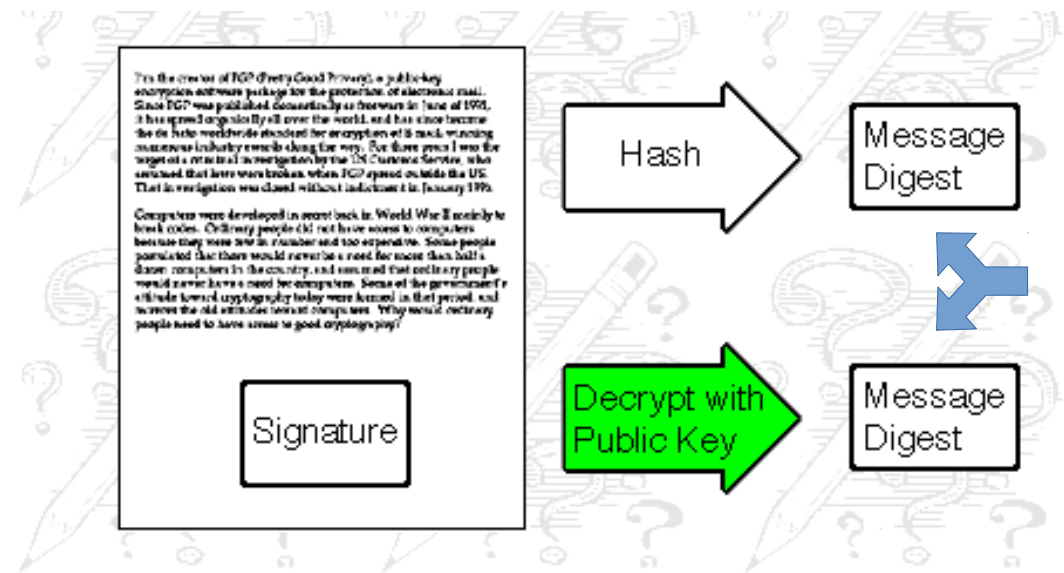
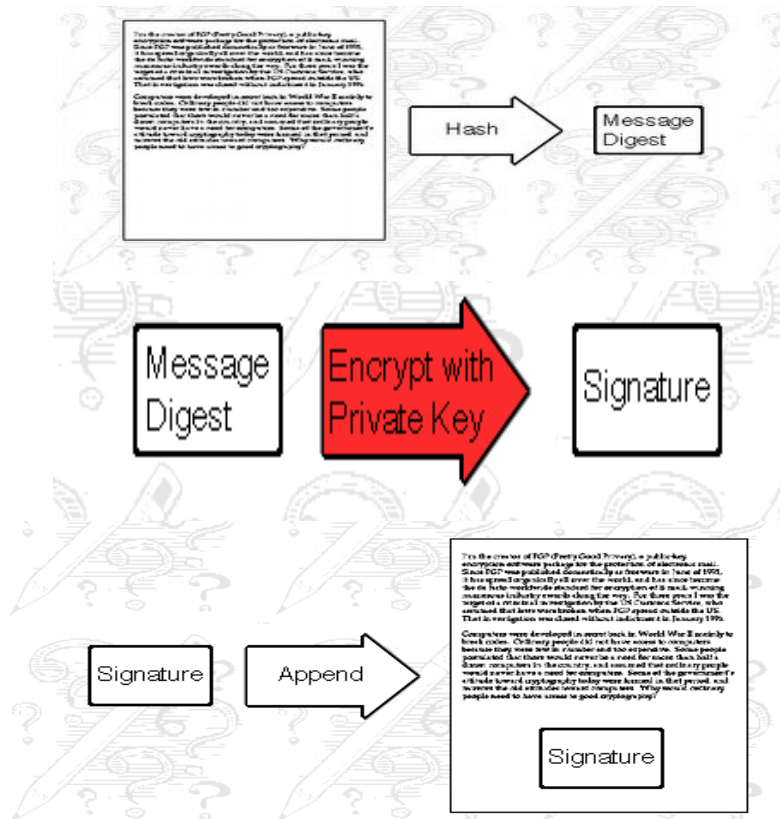
- Proved that message was signed with sender's private key
- Because no other key permits decryption with this public key

Exercise:

~~Bob (signer, sender)~~

Trudy is changing document..

Pat (receiver)



Suppose Trudy intercepts the communication

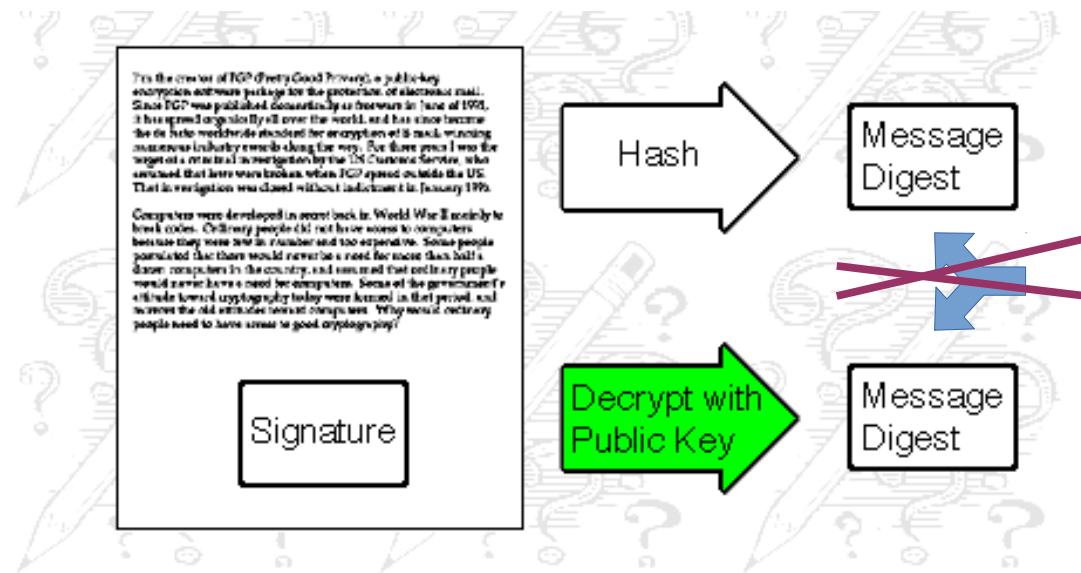
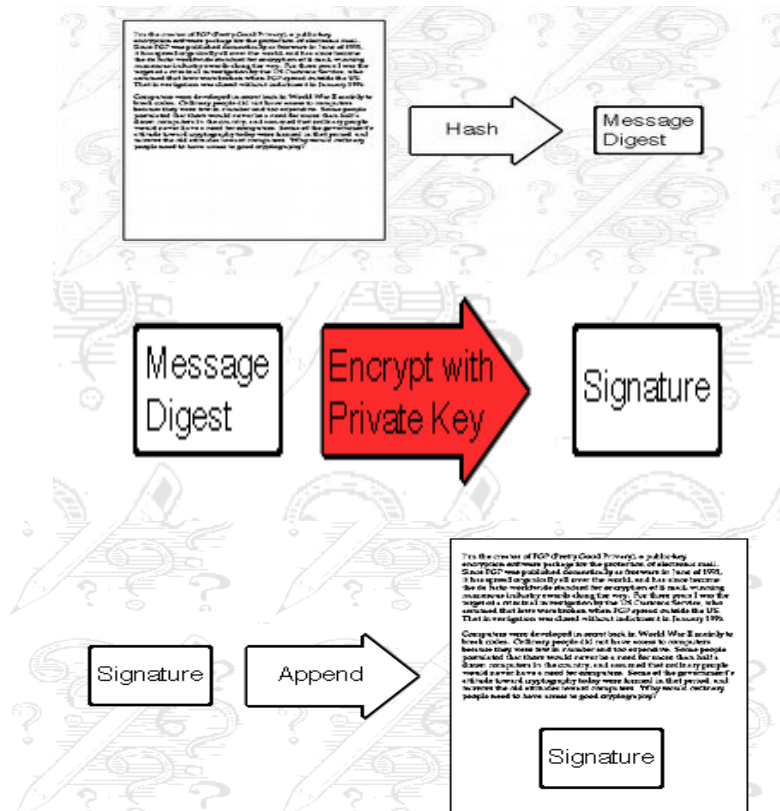
- changes the document
- computes a new message digest
- tries to produce a signature, using Trudy's own private key

How would Trudy's changes be revealed by the receiver?

Solution to exercise

Bob (signer, sender)

Pat (receiver)

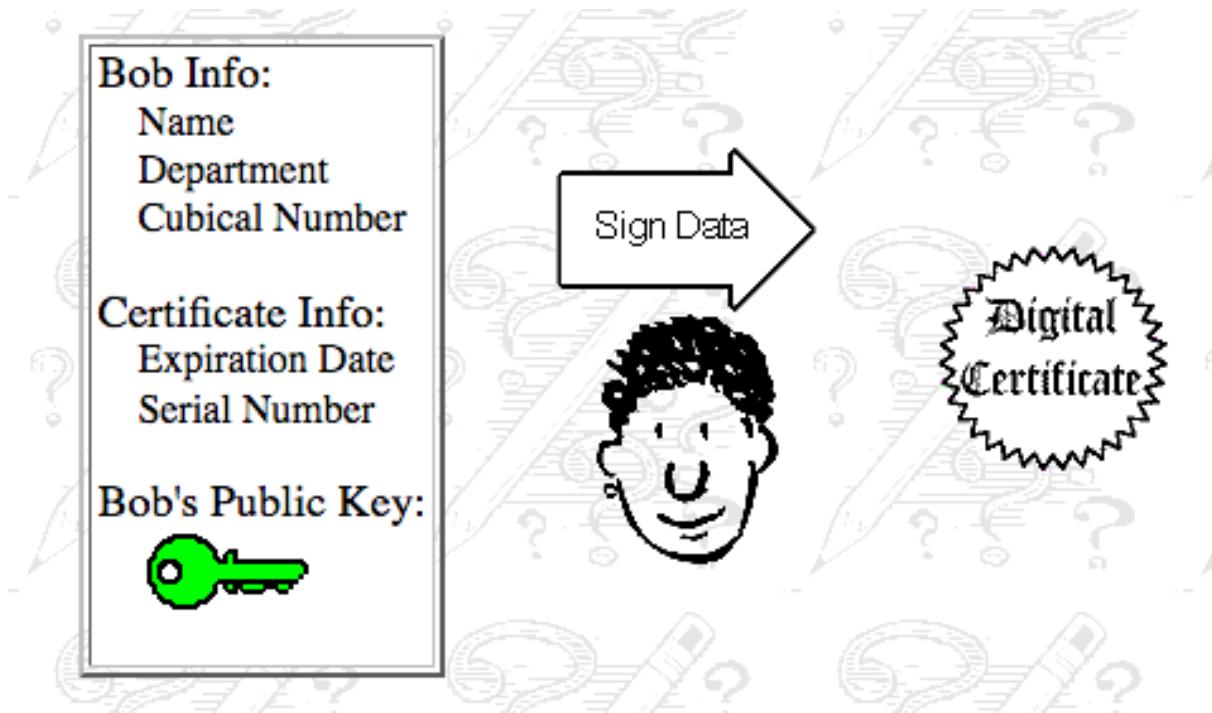


How would Trudy's changes be revealed by the receiver?

Even if Trudy intercepts the communication, Trudy can't compute the correct signature, because Trudy does not possess Bob's private key.

The changes will be revealed because the two message digests will be different.

Digital certificate



Purpose:

- to guarantee “this public key belongs to Bob”

Mechanism

- certificate is itself a digitally signed document
- signed with a trusted person’s private key
- certificate mainly contains: ID (“Bob”) + public key + signature

NemID



NemID is a PKI-based system but with central storage of private keys.

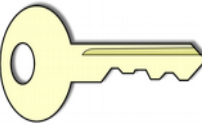
Advantages

- Much easier for the user
 - no installation of a signature file
- More secure for un-protected users (the majority)
 - the signature file can't be stolen from the user's computer

Disadvantages

- The user must trust Nets
 - disgruntled employees at Nets could steal the private key (if they could circumvent security at Nets)
- Danish government could demand access to the private keys
- A foreign government could demand access
 - Nets was sold in 2014 to a consortium of Danish and foreign companies (17 billion DKK)

Agenda

1. Technology: Access control with passwords and hashing
2. Technology: Access control with digital signatures & digital certificates
-  3. Law: EU's General Data Protection Regulation (including pseudonymization)
4. Usable Security (Why Johnny Can't Encrypt)

EU's General Data Protection Regulation (GDPR)

GDPR timeline

- passed April, 2016
- applies in all EU member states from May, 2018

GDPR contents

- focus on privacy, ie., of health-related data
- requirements, fines
- companies/organizations must have a data protection policy
- but requirements are general/vague

Denmark, before GDPR

- “Persondataloven”
- Compliance with ISO 27.000 (a security standard)
 - applies to all government institutions + suppliers

Other EU member states

- require or recommend Privacy Impact Assessments

ISO 27.000 in Denmark

The Office of the Auditor General (Rigsrevisionen)

2015 analysis of MedCom

- health care network (medcom.dk)
- transmits health care data between hospitals etc.

Criticism

- “a threat to the lives and health of patients”
- lack of a contingency plan (“beredskabsplan”)
 - what to do in case of non-availability of some services
- lack of guidelines about access control
 - data accessible to administrators at suppliers

Previous reports have criticized..

- .. the police, tax, defence, statistics and many other institutions

GDPR “considerations”

Considerations 1-173

- motivation, background (“bemærkninger”)
- focus: privacy; “assessing”; “appropriate”

Consideration 83:

[..] evaluate the risks inherent in the processing and implement measures to mitigate those risks, such as encryption. Those measures should ensure an appropriate level of security, including confidentiality, taking into account the state of the art and the costs of implementation in relation to the risks and the nature of the personal data to be protected. In assessing data security risk, consideration should be given to the risks that are presented by personal data processing, such as [..] unauthorised disclosure of, or access to, personal data transmitted, stored or otherwise processed [..]

(My emphasis)

GDPR articles

Articles 1-99

- the actual regulation
- applies in EU member states similarly as a national law

Article 5: Principles relating to processing of personal data

- (a) lawfulness, fairness and transparency
- (b) purpose limitation
- (c) data minimisation
- (d) accuracy
- (e) storage limitation: kept in a form which permits identification of data subjects for no longer than is necessary
- (f) integrity and confidentiality

Accountability: demonstrate that system complies with this article.

GDPR (continued)

Article 32: Security of processing

- 1 Taking into account the state of the art, the costs of implementation and [...] purposes of processing as well as the risk [...] appropriate [...] measures
 - (a) the pseudonymisation and encryption of personal data;
 - (b) [...] confidentiality, integrity, availability and resilience [...]
 - (c) [...] restore [...] in the event of a physical or technical incident;
 - (d) [...] regularly testing, assessing and evaluating the effectiveness of [...] measures for ensuring the security of the processing.
3. Adherence to an approved code of conduct [...] may be used as an element by which to demonstrate compliance with the requirements set out in paragraph 1 of this Article.

GDPR (continued)

Article 6: Principles relating to processing of personal data

Processing requires

- (a) consensus, or
- (b)-(e) necessity

Otherwise (??), take into account

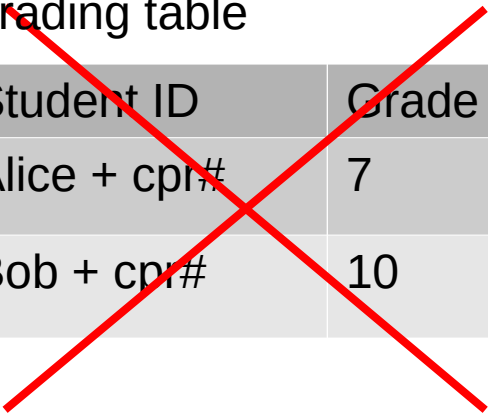
..

- (c) the nature of the personal data
- (d) the possible consequences of the intended further processing for data subjects;
- (e) the existence of appropriate safeguards, which may include encryption or pseudonymisation.

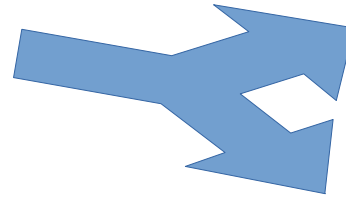
Pseudonymisation (UK) = pseudonymization (US)

Pseudonymization

~~Grading table~~



Student ID	Grade
Alice + cpr#	7
Bob + cpr#	10



Grading table with pseudonyms

Student study#	Grade
309235	7
763915	10

Pseudonym table

Student ID	Student study#
Alice + cpr#	309235
Bob + cpr#	763915

- Use these pseudonyms when
- transmitting btw. adm/teacher
 - publishing grades?
 - in storage?

Pseudonym must be reasonably protected against analysis

- avoid phone#
- because analysis of phone# -> name

Pseudonymization \neq anonymization

- if students were anonymous, the data is useless

Exercise (pseudonymization)

1. In your project, identify the data that is the most sensitive
 - personal data
 - business secrets
2. Suggest a way that pseudonymization can increase the protection of this data. Alternatively, if pseudonymization is not relevant, then argue why this is the case.

Agenda

1. Technology: Access control with passwords and hashing
2. Technology: Access control with digital signatures & digital certificates
3. Law: EU's General Data Protection Regulation (including pseudonymization)



4. Usable Security (Why Johnny Can't Encrypt)

Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0

PGP

= Pretty Good Privacy

A tool for encryption and signing mails, files, and other entities

Developed 1991+

- by Phil Zimmermann
- to promote privacy
- in opposition to US encryption regulation

PGP software is used in Enigmail

- an extension of Mozilla Thunderbird



If privacy is outlawed, only outlaws
will have privacy.

— *Phil Zimmermann* —

AZ QUOTES

Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0

“PGP 5.0 is not usable enough to provide effective security for most computer users” (p1)

PGP 5.0's user interface does not come even reasonably close to achieving our usability standard - it does not make public key encryption of electronic mail manageable for average computer users” (p2)

Usable security

The general concept of usability, applied to security

“Definition: Security software is usable if the people who are expected to use it:

- 1. are reliably made aware of the security tasks they need to perform;*
- 2. are able to figure out how to successfully perform those tasks;*
- 3. don't make dangerous errors; and*
- 4. are sufficiently comfortable with the interface to continue using it.”*

(Whitten & Tygar p2)

Usability

Usability was a dominant paradigm in the late 1990s-2000s

Compare to..

.. user-friendliness

- usability focuses more on the user's task (getting the job done)

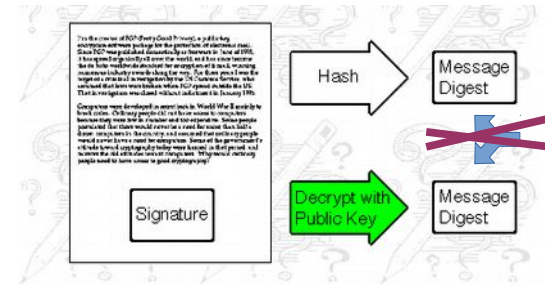
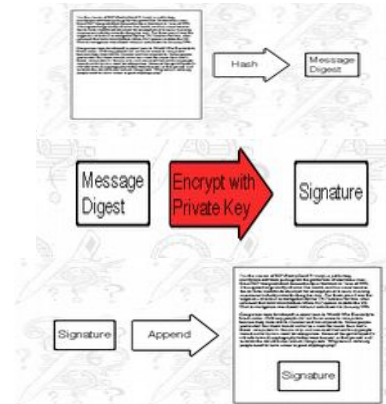
.. user experience

- usability focuses less on the user's emotions

Challenges related to usable security

1. The unmotivated user property
2. The abstraction property
3. The lack of feedback property
4. The barn door property
5. The weakest link property

(Whitten & Tygar p3)



Whitten & Tygar report from two tests

Test #1: Cognitive walkthrough

- W. & T. reviewed the user interface themselves
- image they are novice users
- understand what to do?

Test #2: User test

- asked 12 test persons to use PGP
- test persons should image they were campaign coordinators
- sending an itinerary (schedule) using signing and encryption

Exercise

Summarize Whitten & Tygar's critique of the visual metaphors
(part of the cognitive walkthrough)

Solution to exercise

Summarize Whitten & Tygar's critique of the visual metaphors (part of the cognitive walkthrough)



(Whitten & Tygar, Figure 1, p5)

Critique

1. private and public keys

- W&T would like to see a visual distinction

2. the icon for signing (the quill pen)

- W&T would like to see a visual indication that the private key is used

3. signature verification

- W&T would like a visual indication that the signature is confirmed

Exam questions (examples)

Your project

- what data is sensitive (in particular personal data)
 - why / not ?
- what data is encrypted? pseudonymized?
 - why / not ?
- other measures than encryption / pseudonymization?
 - such as delete data when no longer needed
- usable security
 - definition, goals: are users prevented from *making dangerous errors*?
 - challenges: is the *unmotivated user property* relevant?

Security technology in general

- explain basic properties of
 - symmetric encryption, asymmetric encryption
 - strong passwords, hashing
 - digital signatures, digital certificates
- algorithms such as AES/Rijndael that are considered secure
 - in what sense are they secure?