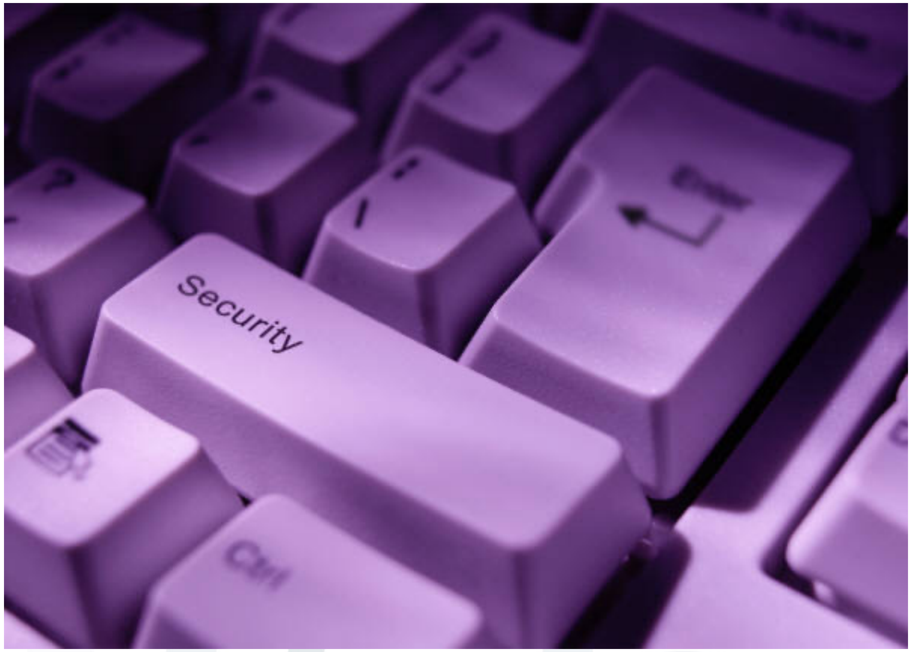
Lecture 4

Cryptography



Mobile Business II (SS 2021)

Prof. Dr. Kai Rannenberg

Chair of Mobile Business & Multilateral Security
Goethe University Frankfurt a. M.

Agenda

- Introduction
- Symmetric Cryptosystems
- Public Key Cryptography

Cryptographic Systems

- Intention
  - Confidentiality (secrecy of messages): encryption systems
  - Integrity (protection from undetected manipulation) and accountability: authentication systems and digital signature systems
- Key distribution
  - Symmetric: Both partners have the same key.
  - Asymmetric: Different (but related) keys for encryption and decryption
- In practice mostly hybrid systems

- Introduction
- Symmetric Cryptosystems
  - General Concept
  - Caesar Cipher
  - AES
  - Advantages and Problems
- Public Key Cryptography

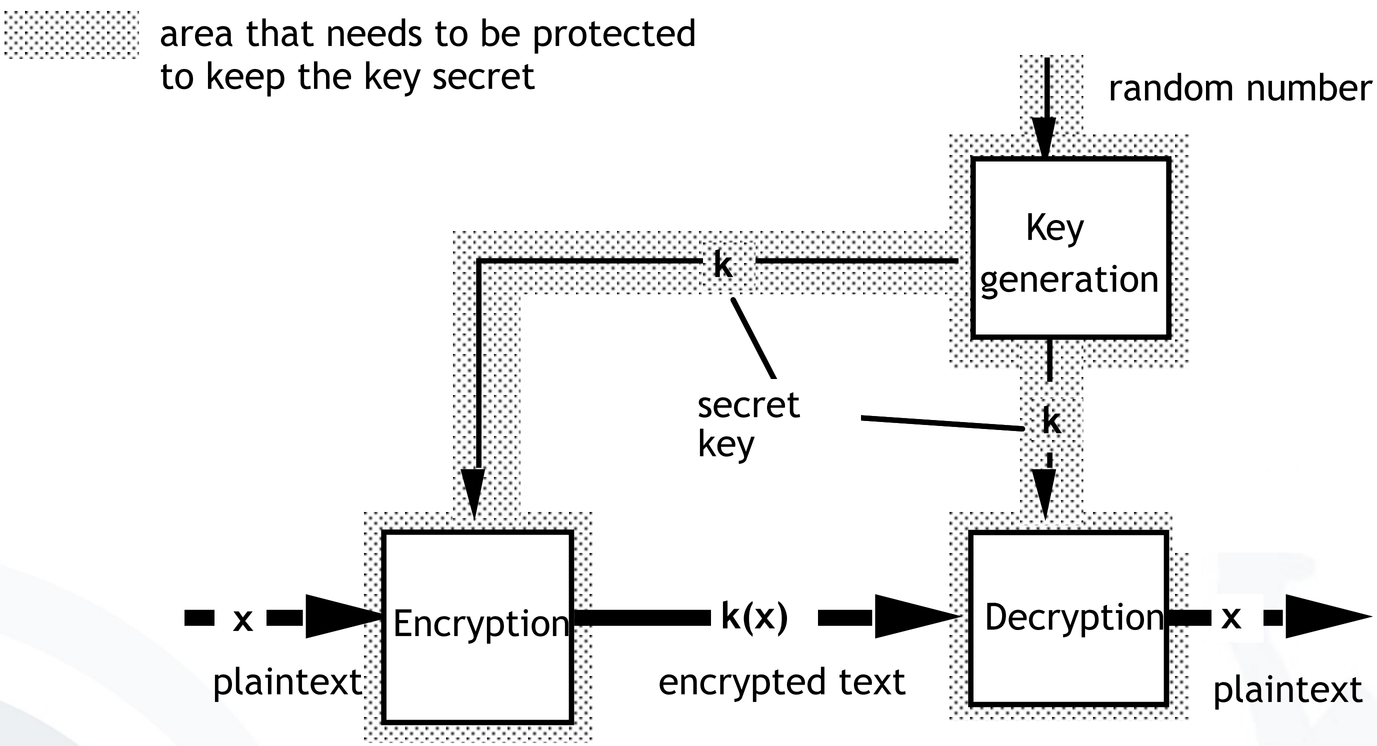
Symmetric Encryption Systems

- Typical applications
  - confidential storage of user data
  - transfer of data between 2 users who negotiate a key via a secure channel
  - end-to-end channel encryption
- Examples
  - **Vernam-Code** (one-time pad, Gilbert Vernam)
    - key length = length of the plaintext (information theoretically secure)
  - **DES: Data Encryption Standard**
    - key length 56 bit → 2^{56} different keys
  - **AES: Advanced Encryption Standard** (Rijndael, [NIST])
    - 3 alternatives for key lengths: 128, 192 and 256 bit

Agenda

- Introduction
- Symmetric Cryptosystems
  - General Concept
  - Caesar Cipher
  - AES
  - Advantages and Problems
- Public Key Cryptography

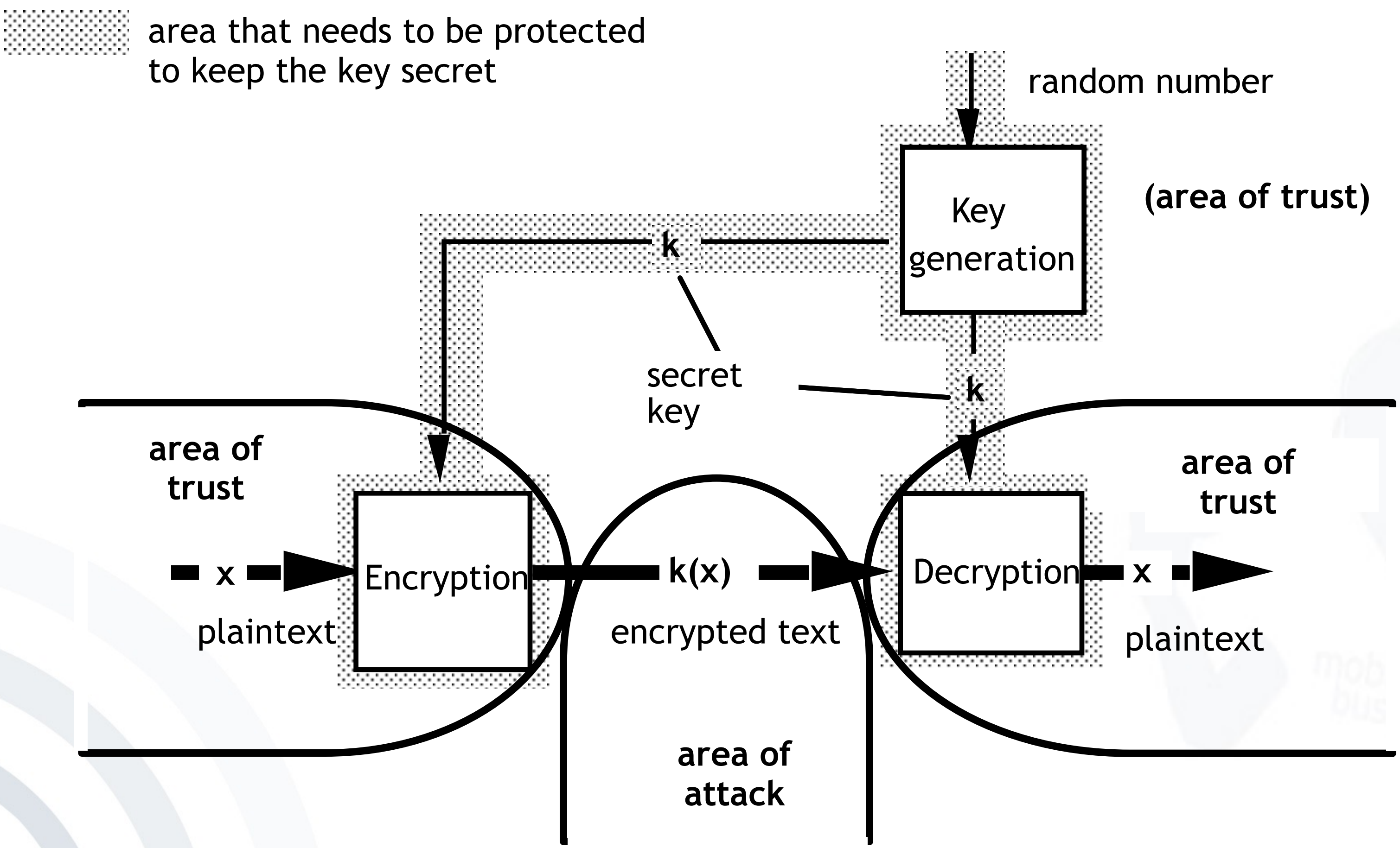
Symmetric Encryption Systems (1)



black box with lock, two equal keys

[based on Federrath and Pfitzmann 1997]

Symmetric Encryption Systems (2)



[based on Federrath and Pfitzmann 1997]

Symmetric Encryption Systems (3)

- **Keys have to be kept secret (*secret key* crypto system).**
- It must not be possible to derive the plaintext or the used keys from the encrypted text (ideally encrypted text is not distinguishable from a numerical random sequence).
- Each key shall be equally probable.
- In principle each system with limited key length is breakable by testing all possible keys.
- **Publication of encoding and decoding functions (algorithms) is considered as good style and is trust-building.**
- **Security of cryptosystems should base on the strength of chosen key lengths.**

Agenda

- Introduction
- Symmetric Cryptosystems
  - General Concept
  - Caesar Cipher
  - AES
  - Advantages and Problems
- Public Key Cryptography

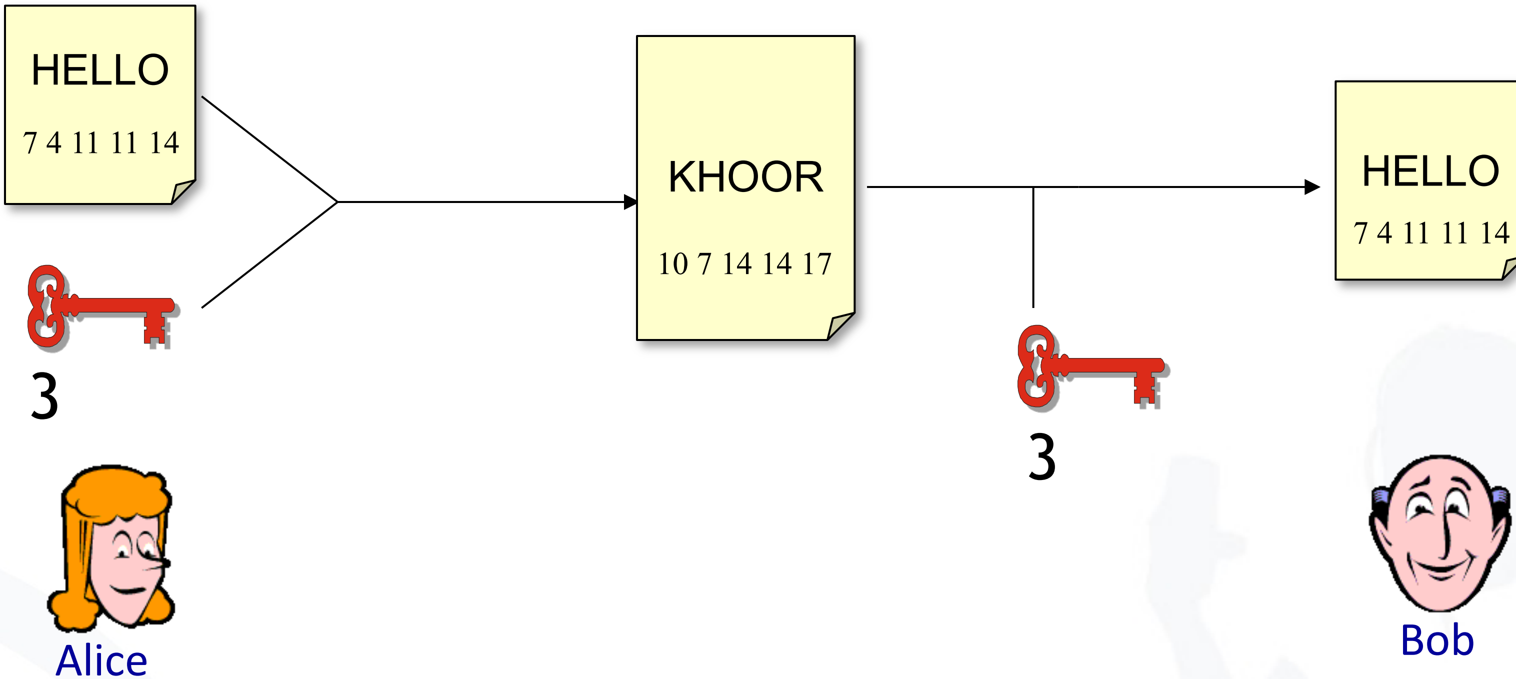
Caesar Cipher

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

- We assign a number for every character.
- This enables us to calculate with letters as if they were numbers.

Caesar Cipher: Example



Assessment of Caesar Cipher

- Very simple form of encryption.
- The encryption and decryption algorithms are very easy and fast to compute.
- It uses a very limited key space (n=26)
- Therefore, the encryption is very easy and fast to compromise.

Agenda

- Introduction
- Symmetric Cryptosystems
  - General Concept
  - Caesar Cipher
  - AES
  - Advantages and Problems
- Public Key Cryptography

Advanced Encryption Standard

- The Data Encryption Standard (DES) was designed to encipher sensitive but not classified data.
- The standard has been issued in 1977.
- In 1998, a design for a computer system and software that could break any DES-enciphered message within a few days was published.
- By 1999, it was clear that the DES no longer provided the same level of security it had 10 years earlier, and the search was on for a new, stronger cipher.
- The successor is called Advanced Encryption Standard (AES).
- AES has been approved for Secret or even Top Secret information by the NSA.

[Bishop 2005]

Agenda

- Introduction
- Symmetric Cryptosystems
  - General Concept
  - Caesar Cipher
  - AES
  - Advantages and Problems
- Public Key Cryptography

Symmetric Encryption

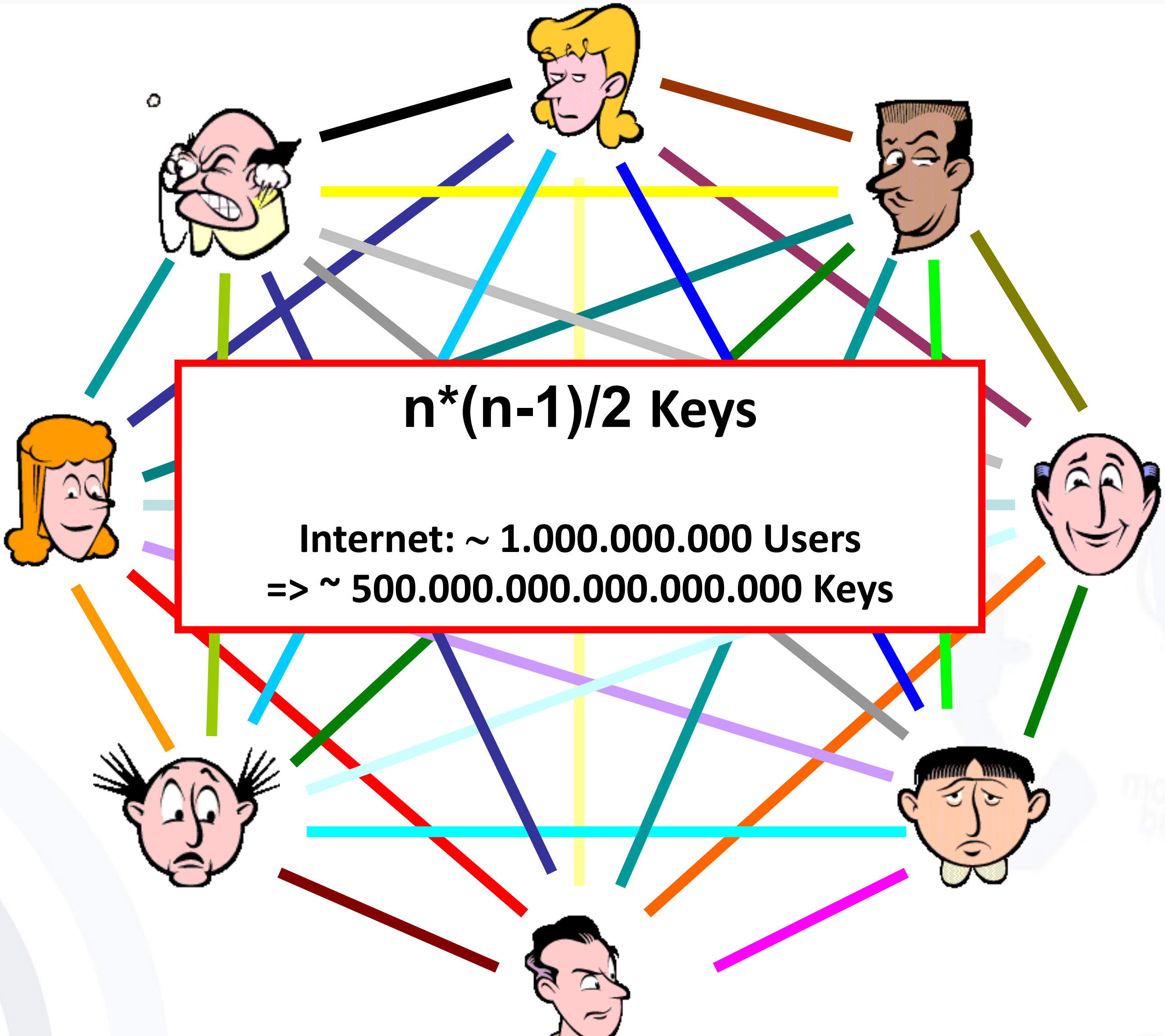
Advantage: Algorithms are very fast

Algorithm	Performance*
RC6	78 ms
SERPENT	95 ms
IDEA	170 ms
MARS	80 ms
TWOFISH	100 ms
DES-ede	250 ms
RIJNDEAL (AES)	65 ms

* Encryption of 1 MB on a Pentium 2.8 GHz, using the FlexiProvider Java)

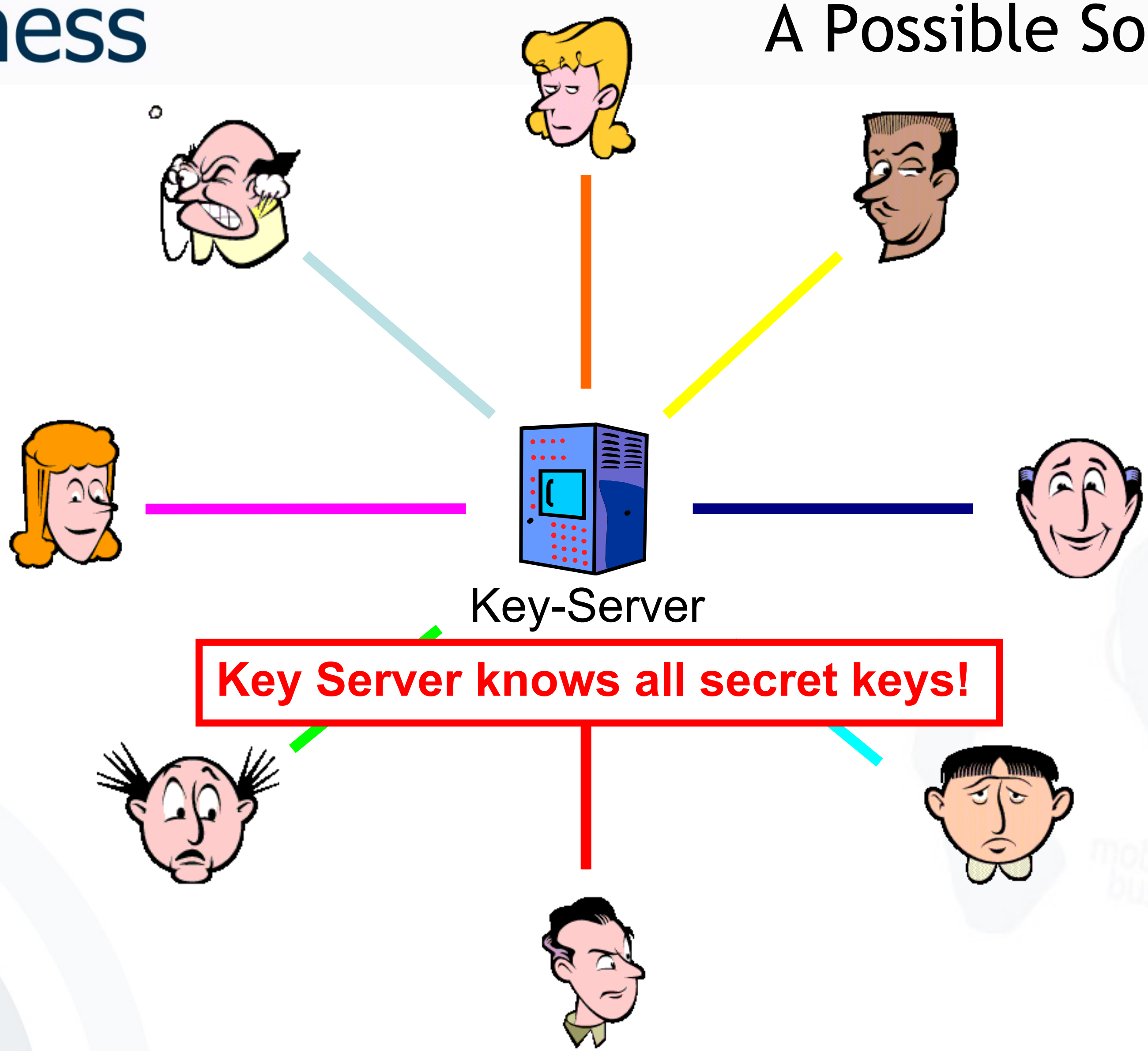
[J. Buchmann: Lecture Public Key Infrastrukturen, FG Theoretische Informatik, TU-Darmstadt]

Problems of Symmetric Cryptosystems: Key Exchange



[J. Buchmann: Lecture Public Key Infrastrukturen, FG Theoretische Informatik, TU-Darmstadt]

Symmetric Encryption: A Possible Solution



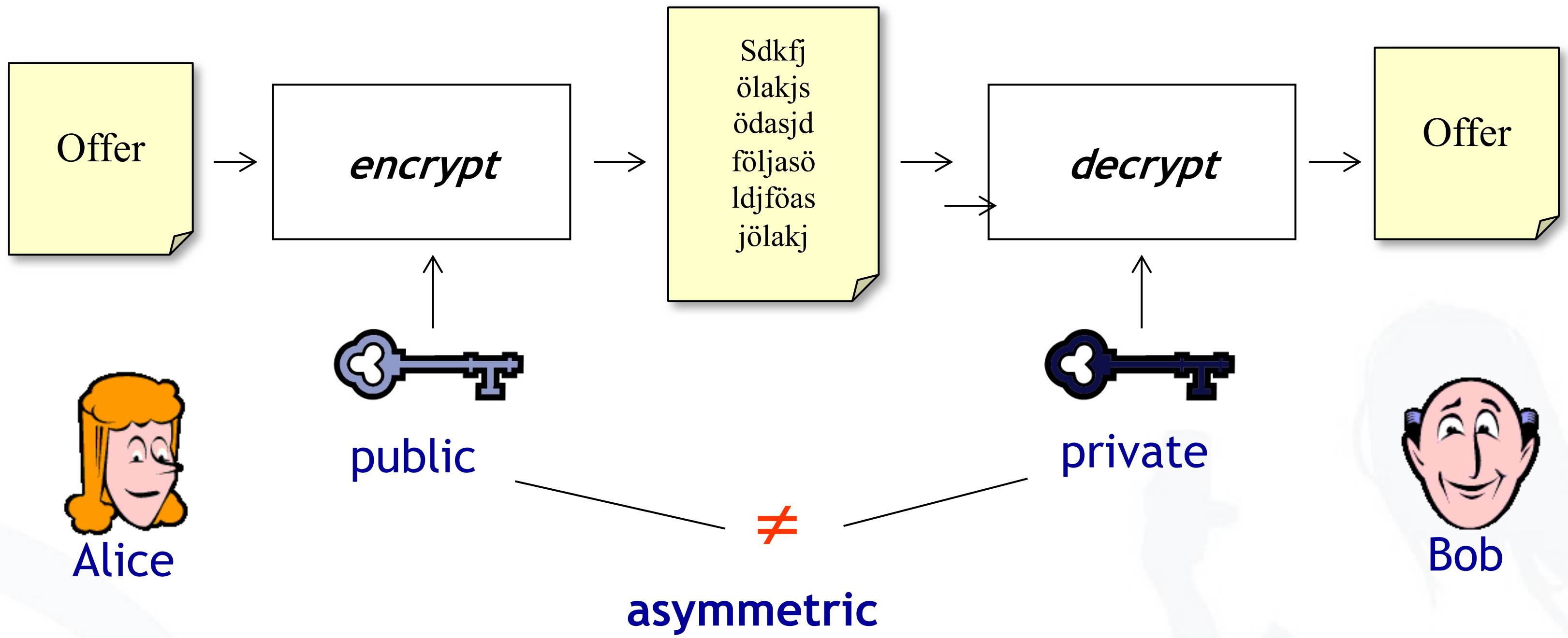
[J. Buchmann: Lecture Public Key Infrastrukturen, FG Theoretische Informatik, TU-Darmstadt]

mobile business

Agenda

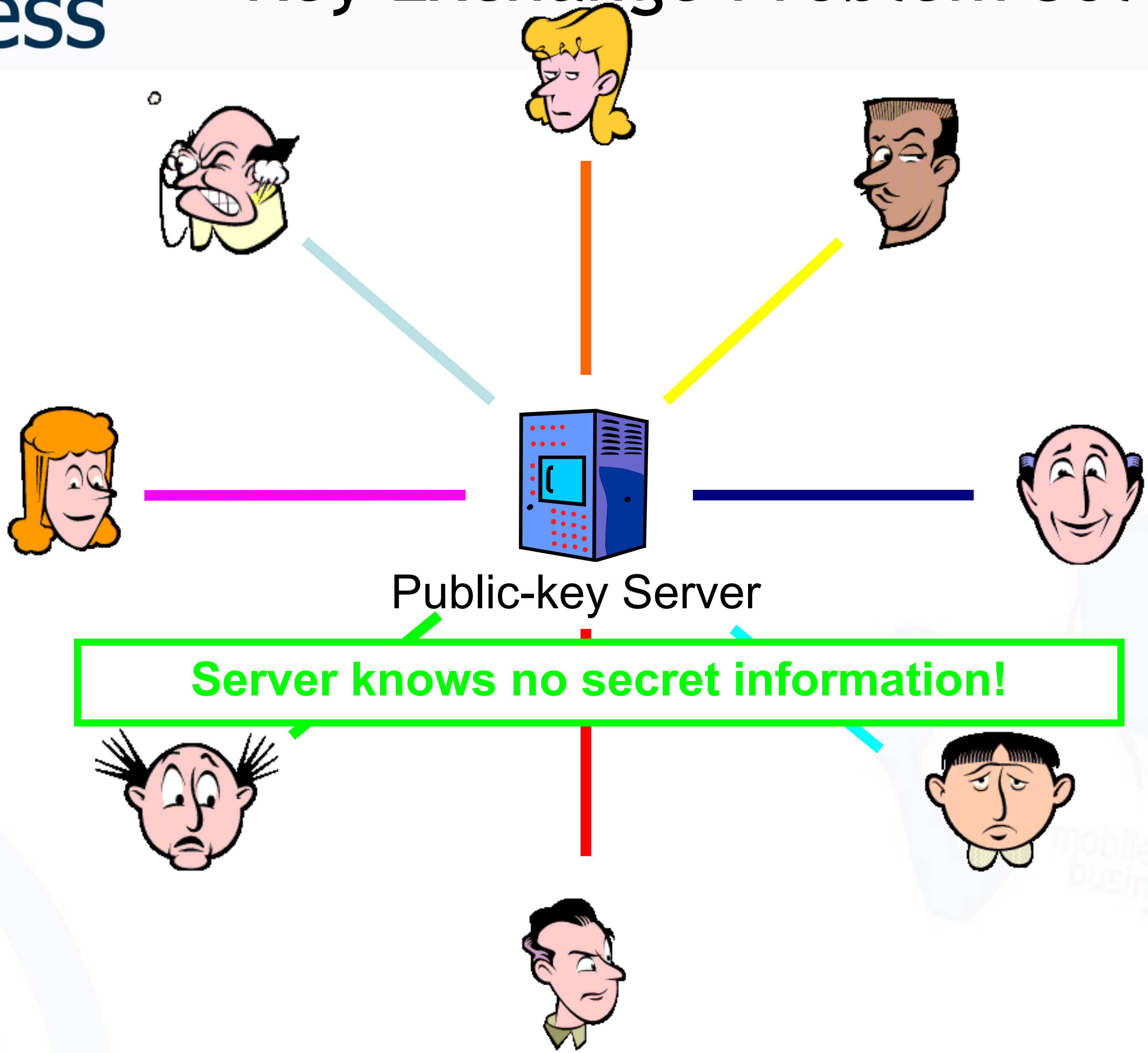
- Introduction
- Symmetric Cryptosystems
- Public Key Cryptography
  - General Concept
  - Algorithms
  - Hybrid Systems
  - Digital Signature
  - Key Management
  - Example: PGP

Public Key Encryption



[J. Buchmann: Lecture Public Key Infrastrukturen, FG Theoretische Informatik, TU-Darmstadt]

Key Exchange Problem Solved!



[J. Buchmann: Lecture Public Key Infrastrukturen, FG Theoretische Informatik, TU-Darmstadt]

Agenda

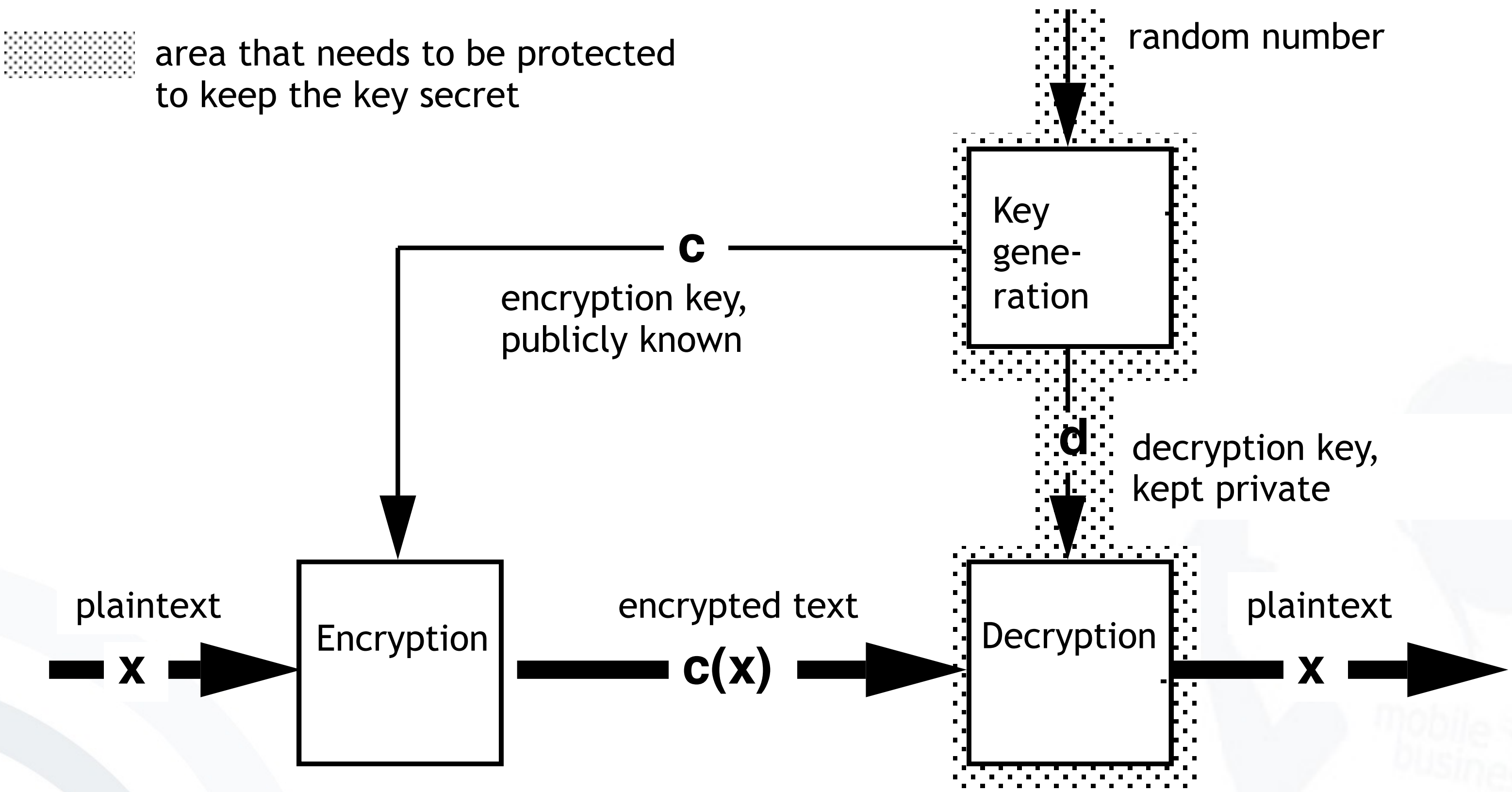
- Introduction
- Symmetric Cryptosystems
- Public Key Cryptography
  - General Concept
  - Algorithms
  - Hybrid Systems
  - Digital Signature
  - Key Management
  - Example: PGP

Concept of Asymmetric Encryption Systems

- Use of 'corresponding' key pairs instead of one key:
  - **Public key** is **solely** for encryption.
  - Encrypted text can only be decrypted with the corresponding **private (undisclosed) key**.
- Deriving the private key from the public key is hard (practically impossible).
- The public key can be distributed freely, even via insecure ways (e.g. directory *(public key* crypto system)).

- Messages are encrypted via the public key of the addressee.
- Only the addressee possesses the private key for decoding (and has to manage the relation between the private and the public key).

Asymmetric Encryption Systems



box with slot, access to messages only with a key

[based on Federrath and Pfitzmann 1997]

Agenda

- Introduction
- Symmetric Cryptosystems
- Public Key Cryptography
  - General Concept
  - Algorithms
  - Hybrid Systems
  - Digital Signature
  - Key Management
  - Example: PGP

Asymmetric Encryption Systems: Examples

- RSA
  - Rivest, Shamir, Adleman, 1978
  - is based on the assumption that the factorization of the product of two (big) prime numbers (p*q) is "difficult" (product is basis for the keys)
  - key lengths typically 1024 bit, today rather 2048

[Rivest et al., 1978]

- Diffie-Hellman
  - Diffie, Hellman, 1976, first patented algorithm with public keys
  - allows the exchange of a secret key
  - is based on the "difficulty" of calculating discrete logarithms in a finite field

[Diffie, Hellman, 1976]

mobile business

Agenda

- Introduction
- Symmetric Cryptosystems
- Public Key Cryptography
  - General Concept
  - Algorithms
  - Hybrid Systems
  - Digital Signature
  - Key Management
  - Example: PGP

Performance of Public Key Algorithms

Algorithm	Performance*	Performance compared to Symmetric encryption (AES)
RSA (1024 bits)	6.6 s	Factor 100 slower
RSA (2048 bits)	11.8 s	Factor 180 slower

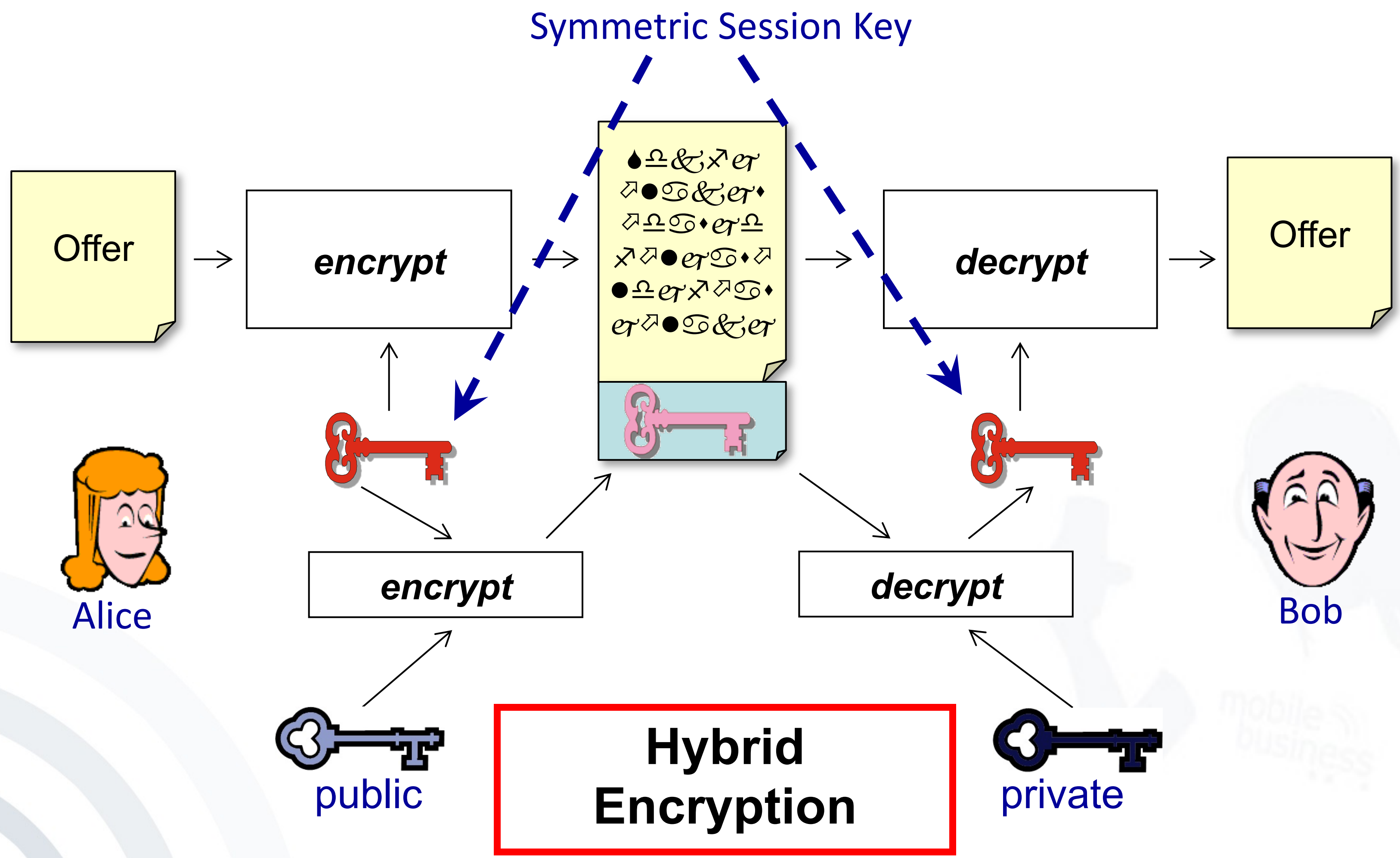
Disadvantage: Complex operations with very big numbers

⇒ Algorithms are very slow.

* Encryption of 1 MB on a Pentium 2.8 GHz, using the FlexiProvider (Java)

[J. Buchmann: Lecture Public Key Infrastrukturen, FG Theoretische Informatik, TU-Darmstadt]

Solution: Hybrid Systems

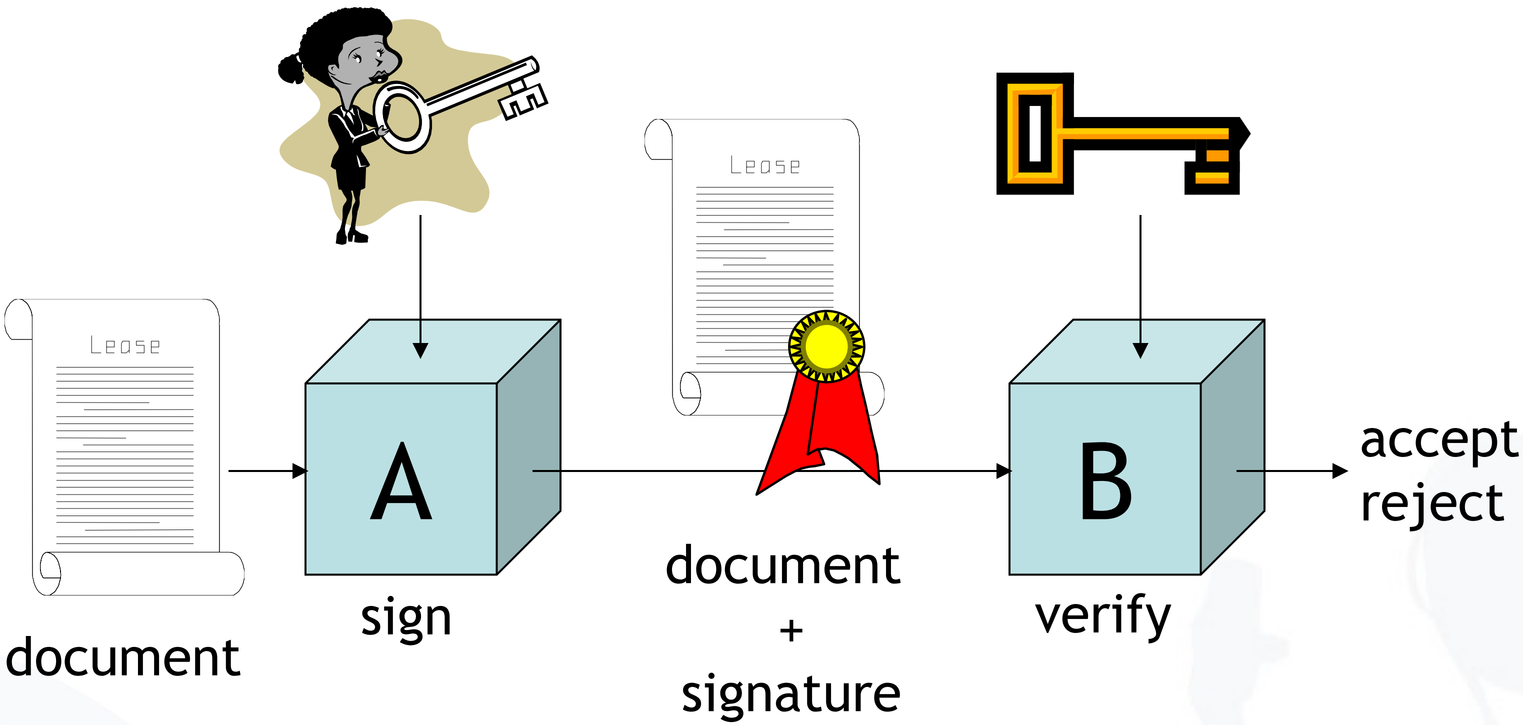


[based on: J. Buchmann 2005: Lecture Public Key Infrastrukturen, FG Theoretische Informatik, TU-Darmstadt]

Agenda

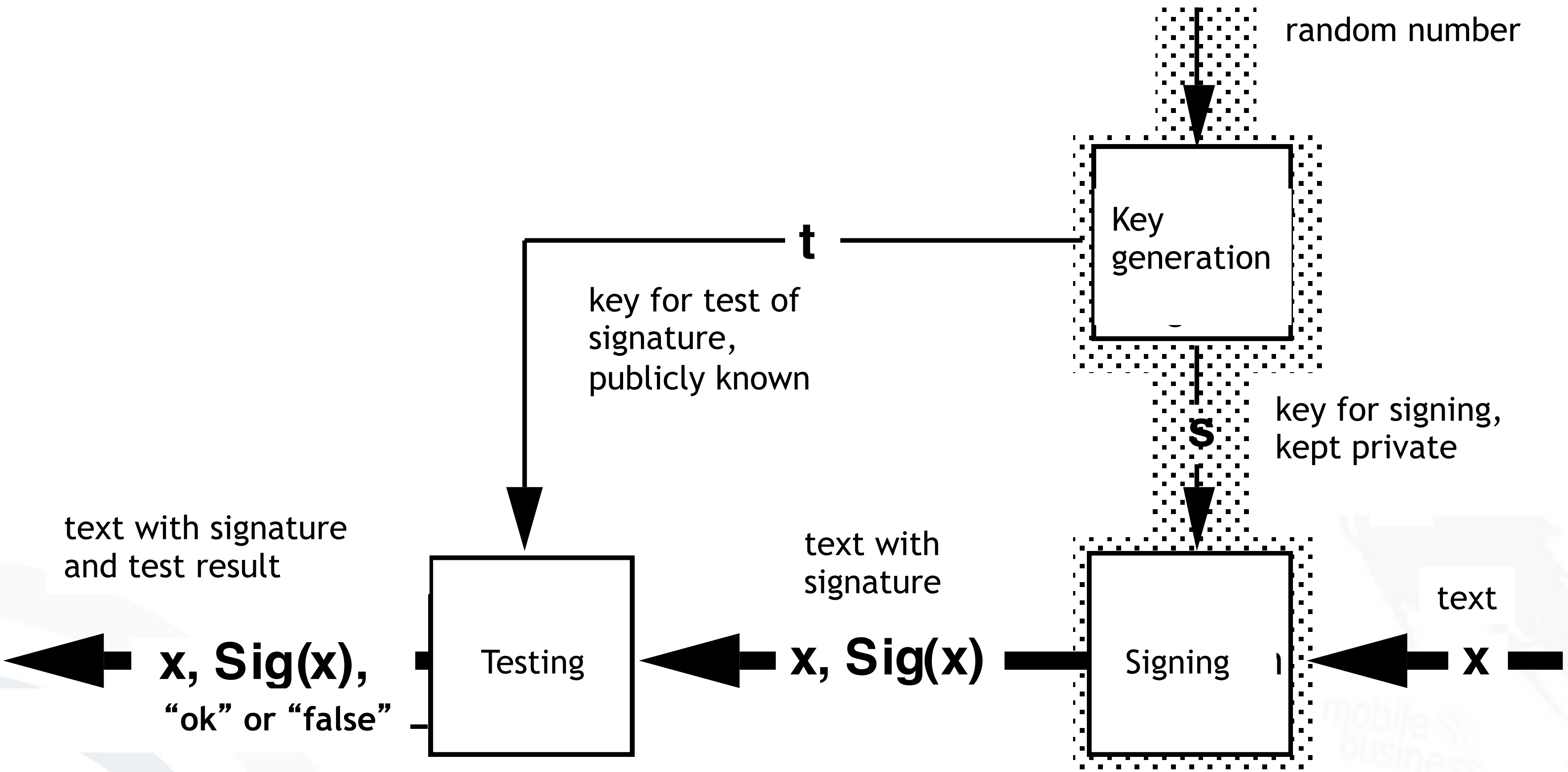
- Introduction
- Symmetric Cryptosystems
- Public Key Cryptography
  - General Concept
  - Algorithms
  - Hybrid Systems
  - Digital Signature
  - Key Management
  - Example: PGP

Digital Signatures



- Protect the authenticity and integrity of documents signed by A
- B has to get an authentic copy of A's public key.

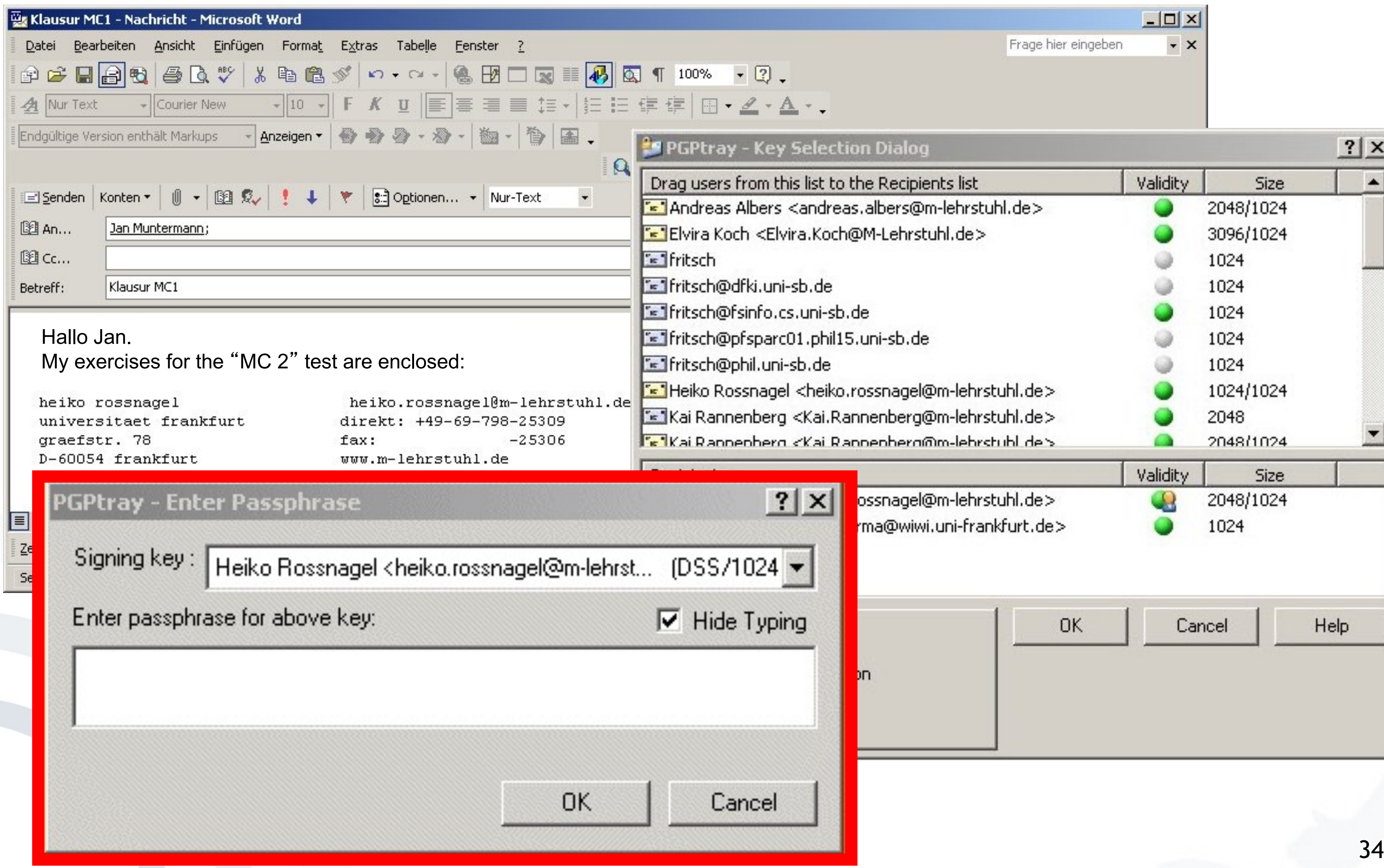
Asymmetric Signature System



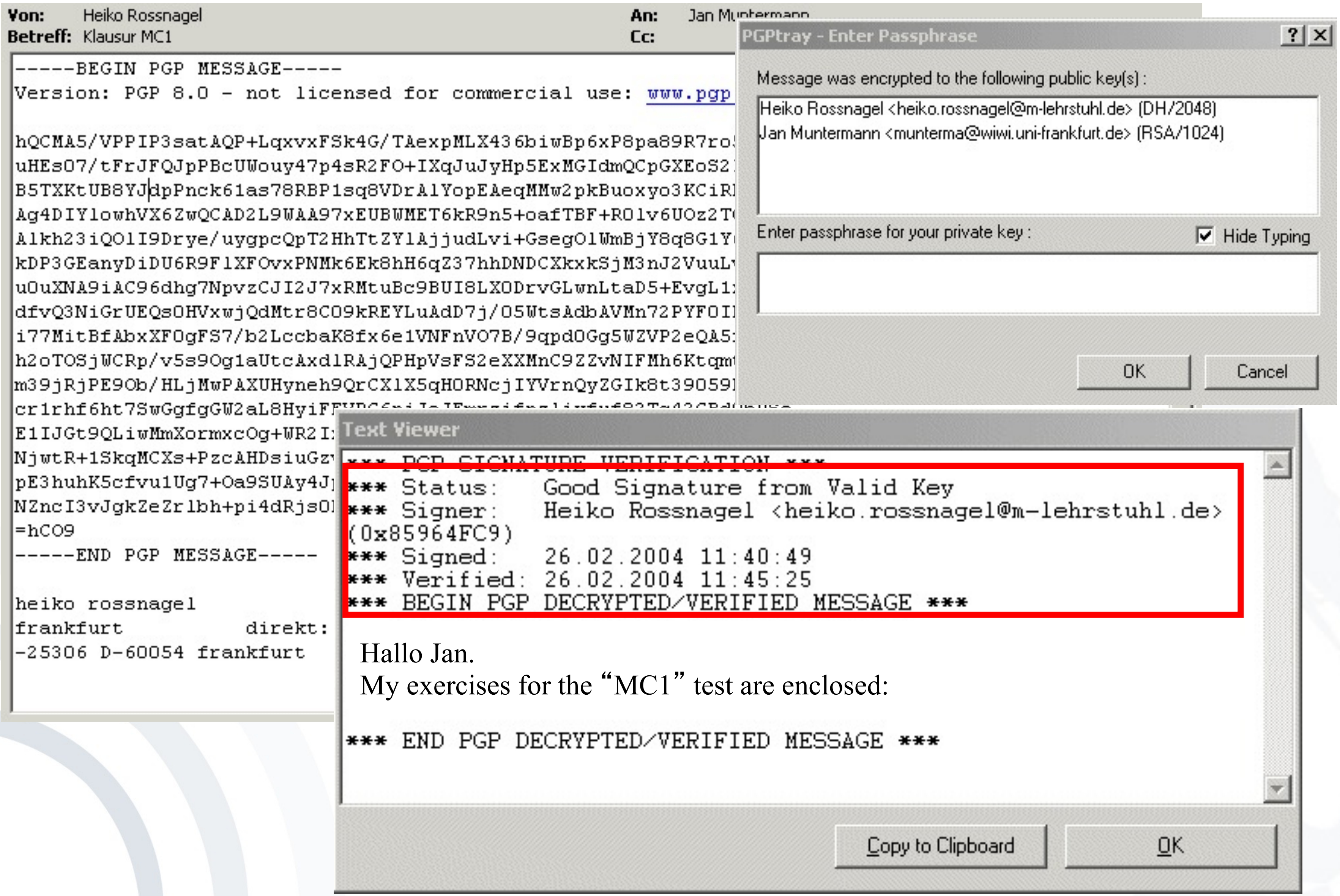
➲ locked glass show-case; just one key to put something in

[Federrath and Pfitzmann 1997]

Example PGP: Encrypt and Sign a Message



Example PGP: Decrypt and Check a Message



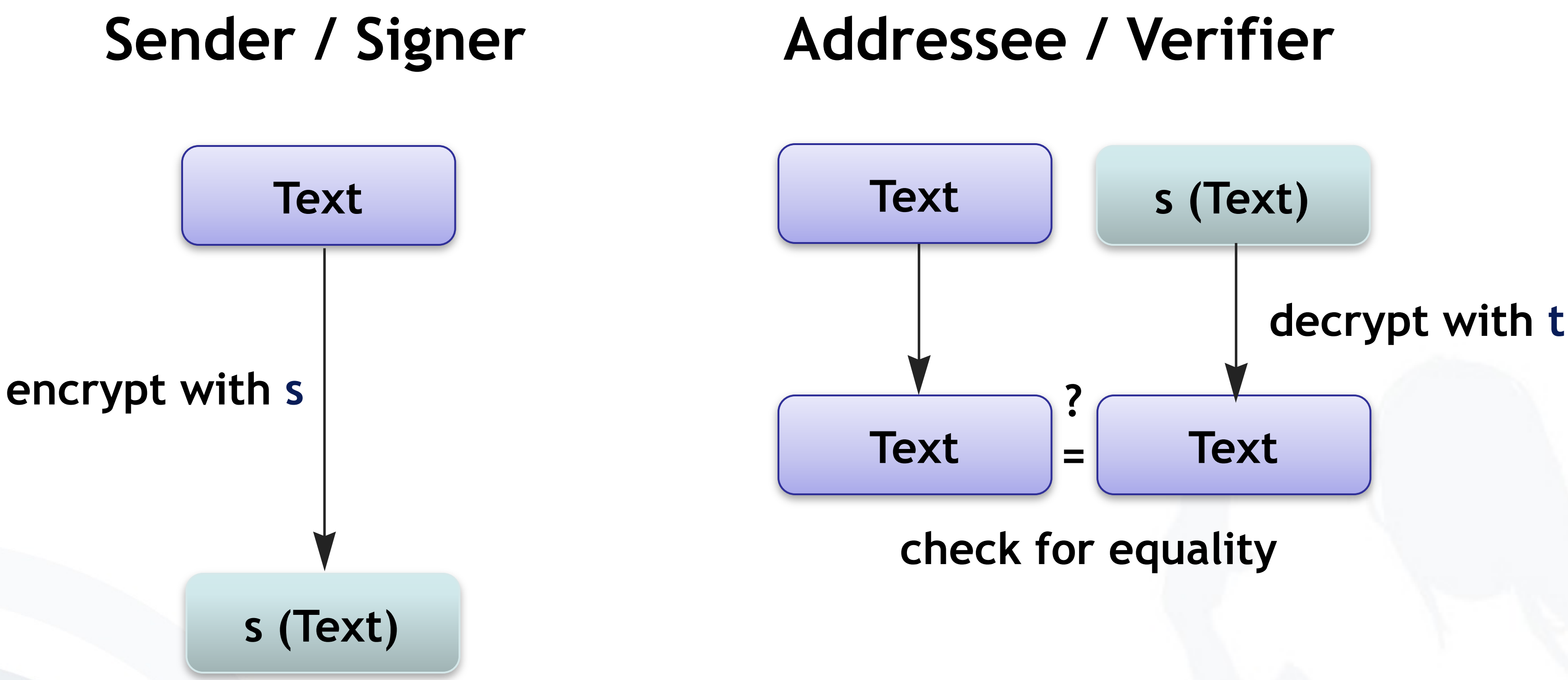
Asymmetric Signature Systems: Examples

- RSA: **R**ivest, **S**hamir, **A**dleman
  - Asymmetric encryption system which also can be used as a signature system via “inverted use”,
  - Message encrypted with the private key (= signing key) gives the signature,
  - Decoding with the public key (=testing key) has to produce the message.

[Rivest et al. 1978]

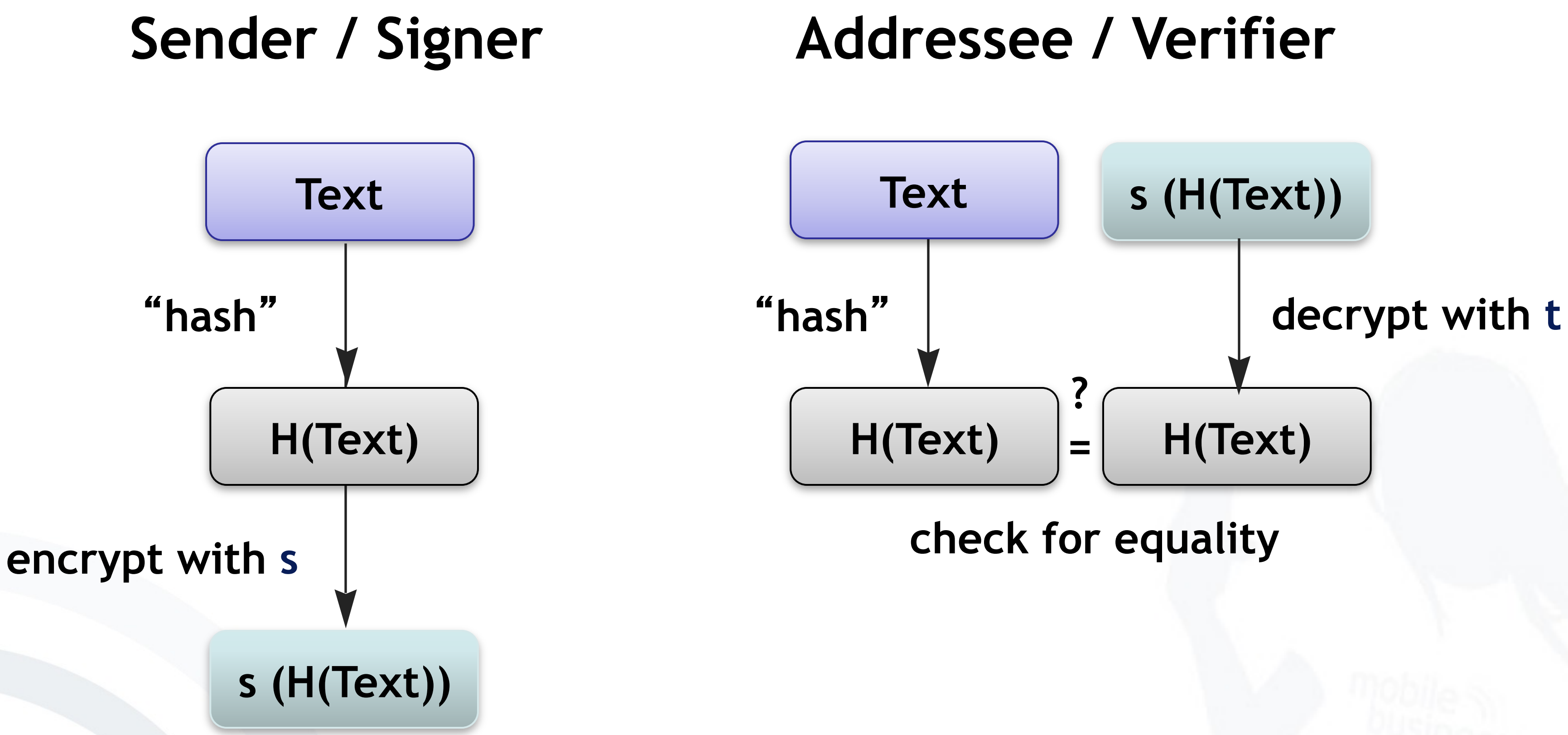
- DSA: Digital Signature Algorithm
  - Determined in the Digital Signature Standard of the NIST (USA),
  - Based on discrete logarithms (Schnorr, ElGamal),
  - Key length is set to 1024 bit.

Asymmetric Signature System (Simplified Example RSA)



- ➲ Signing key s only with the sender, test key t public
- ➲ Example is often mistakenly generalized.

Asymmetric Signature System (Example RSA)



- ➲ Signing key s only with the sender, test key t public
- ➲ Example is often mistakenly generalized.

Hash Functions

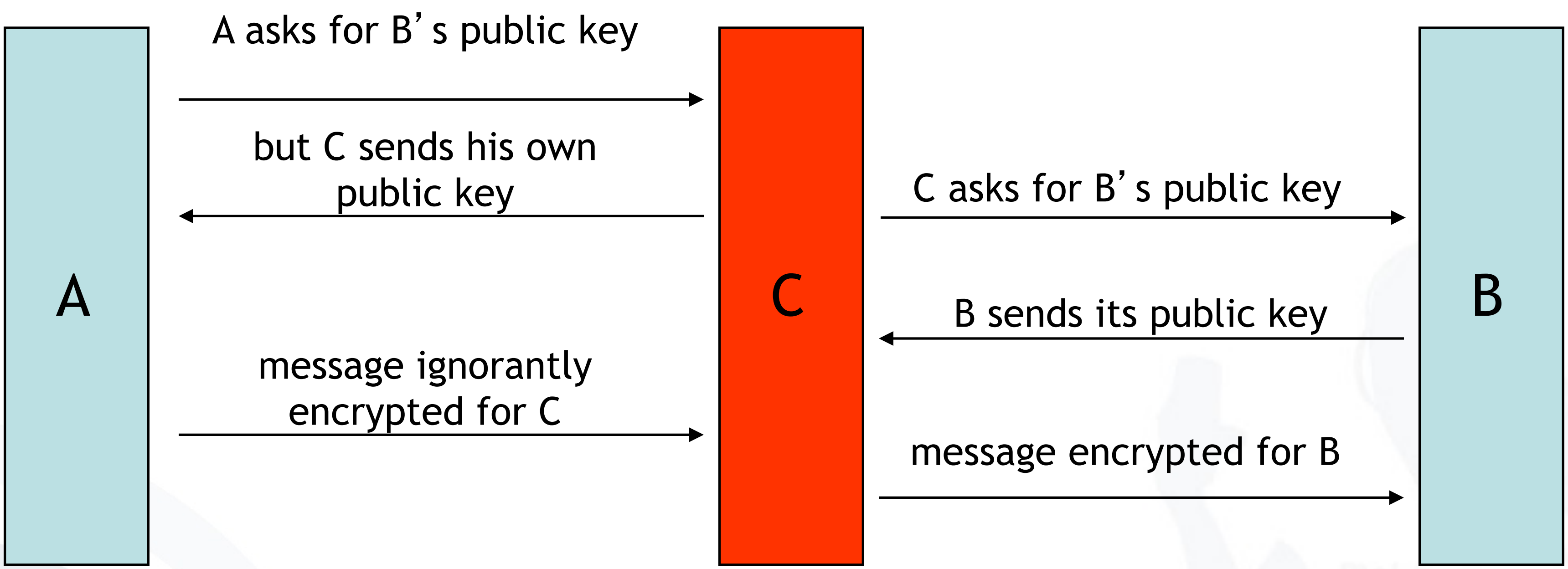
- **General** hash functions *(H(s))*
  - Transformation of an input string *s* into an output string *h* **of fixed length** which is called hash value.
  - Example: mod 10 in the decimal system
- **Cryptographic** hash functions
  - Generally require further characteristics
    - *H*(*s*) is easily to compute for each *s*.
    - *H*(*s*) must be difficult to invert: In terms of figures it is difficult to compute *s* from *h*.
    - Virtual collision freedom: In terms of figures it is difficult to create collisions H(s1) = H(s2).
  - Examples: SHA-1, MD5, MD4

Agenda

- Introduction
- Symmetric Cryptosystems
- Public Key Cryptography
  - General Concept
  - Algorithms
  - Hybrid Systems
  - Digital Signature
  - Key Management
  - Example: PGP

Attacks on Public Key Distribution

"Man in the middle attack"



➲ Keys are certified: a 3rd person/institution confirms (with its digital signature) the affiliation of the public key to a person.

Certification of Public Keys (1)

- **B** can freely distribute his own public key.
- But: Everybody (e.g. **C**) could distribute a public key and claim that this one belongs to **B**.
- If **A** uses this key to send a message to **B**, **C** will be able to read this message!
- Thus:
  How can **A** decide if a public key was really created and distributed by **B** without asking **B** directly?

➲ Keys get **certified**, i.e. a third person/institution confirms with its (digital) signature the **affiliation of a public key to entity B**.

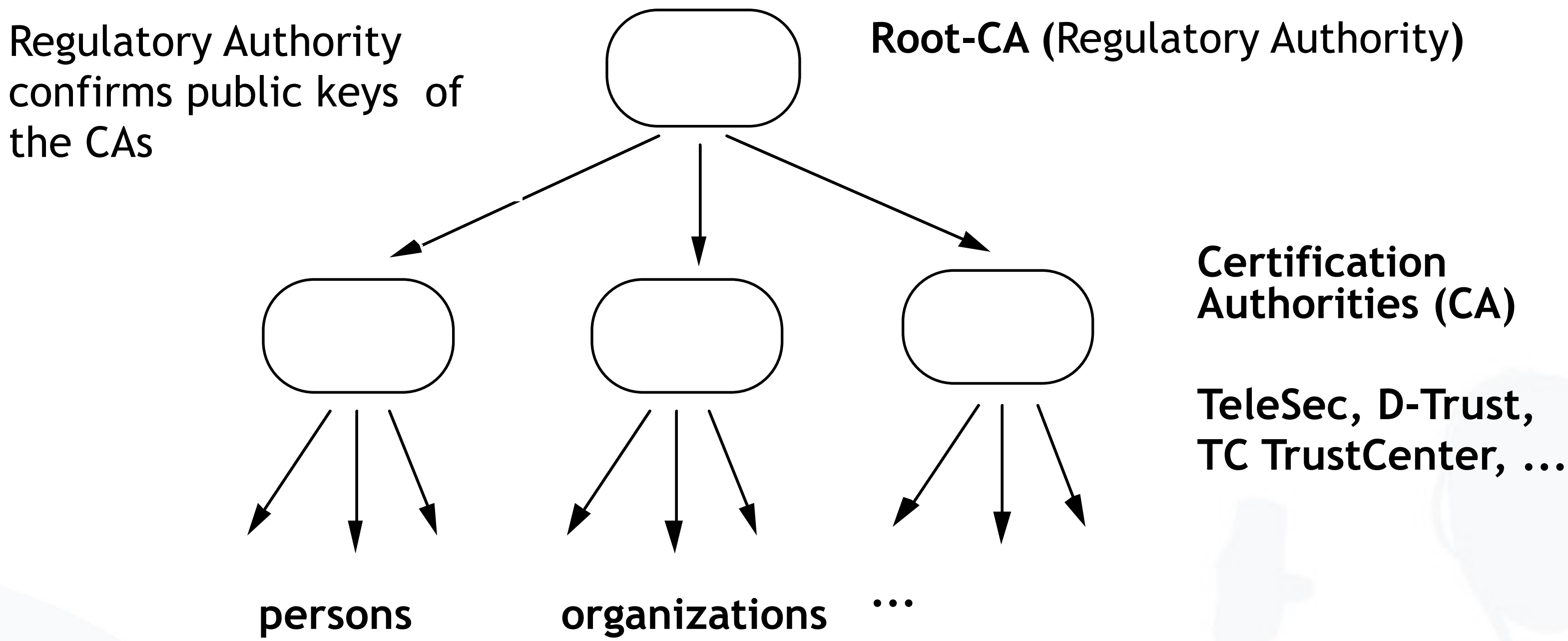
➲ Public Key Infrastructures (PKIs)

Certification of Public Keys (2)

Three types of organization for certification systems (PKIs?):

- Central certification authority (CA)
  - A single CA, keys often integrated in checking software
  - Example: older versions of Netscape (CA = Verisign)
- Hierarchical certification system
  - CAs which in turn are certified by “higher” CA
  - Examples: PEM, Teletrust, infrastructure according to Signature Law
- Web of Trust
  - Each owner of a key may serve as a CA
  - Users have to assess certificates on their own
  - Example: PGP (but with hierarchical overlay system)

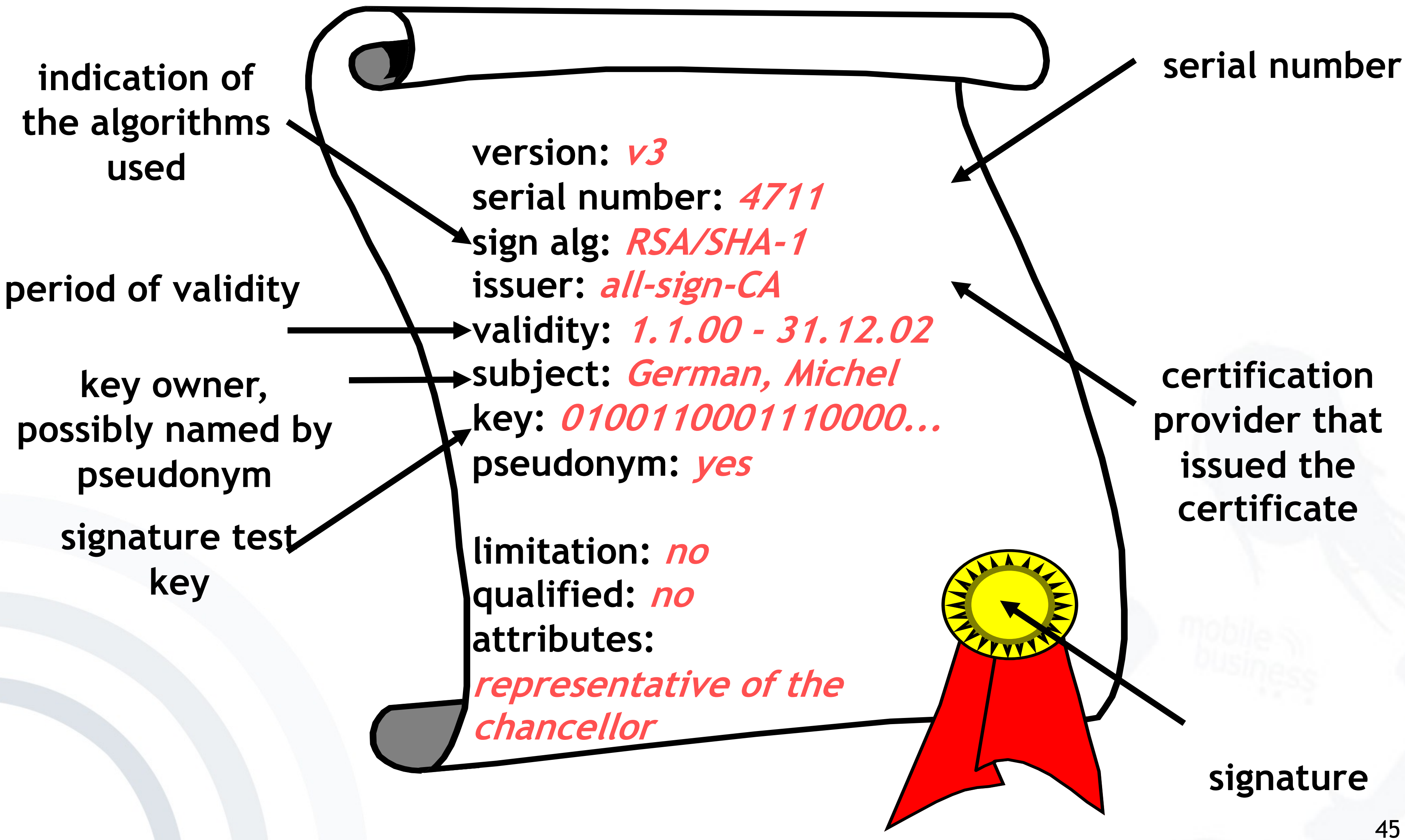
Hierarchical Certification of Public Keys
(Example: German Signature Law)



- The actual checking of the identity of the key owner takes place at so called Registration Authorities
(e.g. notaries, bank branches, T-Points, ...)
- Security of the infrastructure depends on the reliability of the CAs.

Content of a Key Certificate

(according to German Signature Law and Regulation)



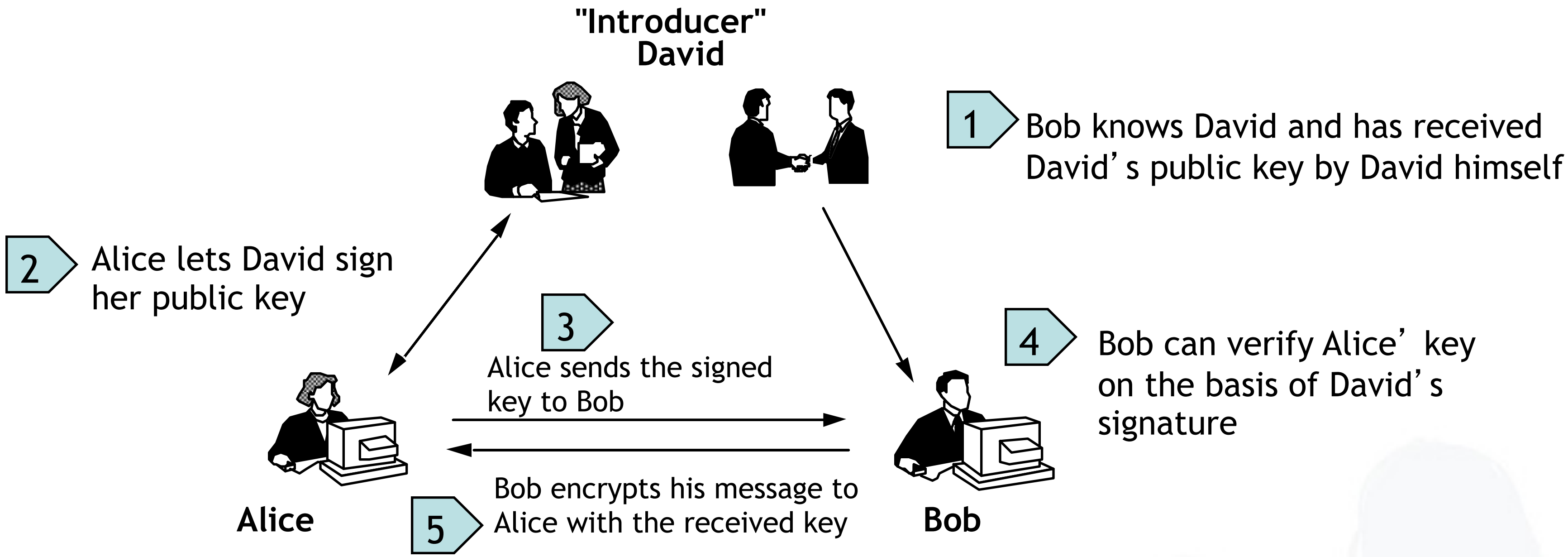
Tasks of a Certification Authority
(according to German Signature Law and Regulation)

- Reliable identification of persons who apply for a certificate
- Information on necessary methods for fraud resistant creation of a signature
- Provision for secure storage of the private key
  - at least Smartcard (protected by PIN)
- Publication of the certificate (if wanted)
- Barring of certificates
- If necessary issuing of time stamps
  - for a fraud resistant proof that an electronic document has been at hand at a specific time

Requirements to an Accredited CA
(according to German Signature Law and Regulation)

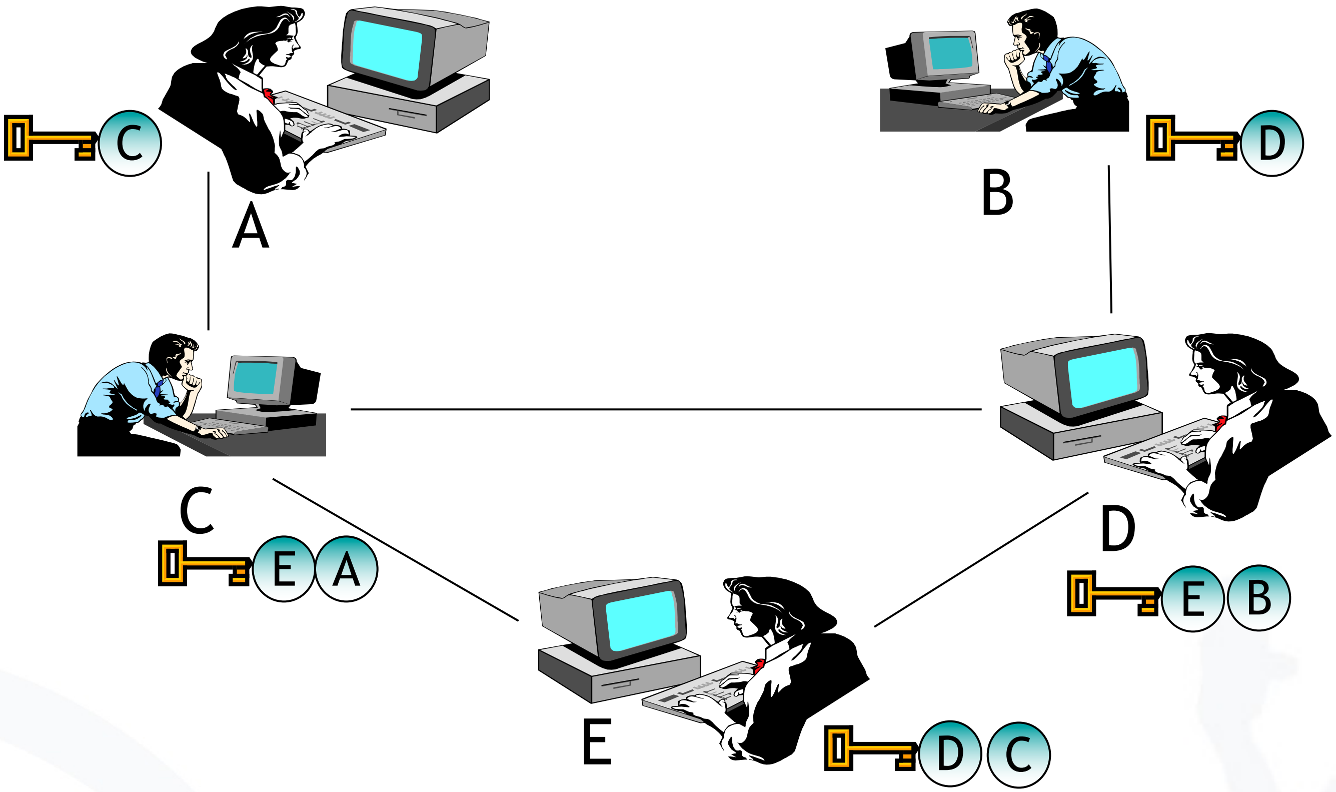
- Checking of the following items by certain confirmation centers (BSI, TÜVIT, ...)
  - Concept of operational security
  - Reliability of the executives and of the employees as well as of their know-how
  - Financial power for continuous operation
  - Exclusive usage of licensed technical components according to SigG and SigV
  - Security requirements as to operating premises and their access controls
- Possibly license of the regulation authority

Web of Trust



- Each user can act as a "CA".
- Mapping of the social process of creation of trust.
- Keys are "certified" through several signatures.
- Expansion is possible by public key servers and (hierarchical) CAs.

Web of Trust Example



Web of Trust:

- Certification of the public keys mutually by users
- Level of the mutual trust is adjustable.

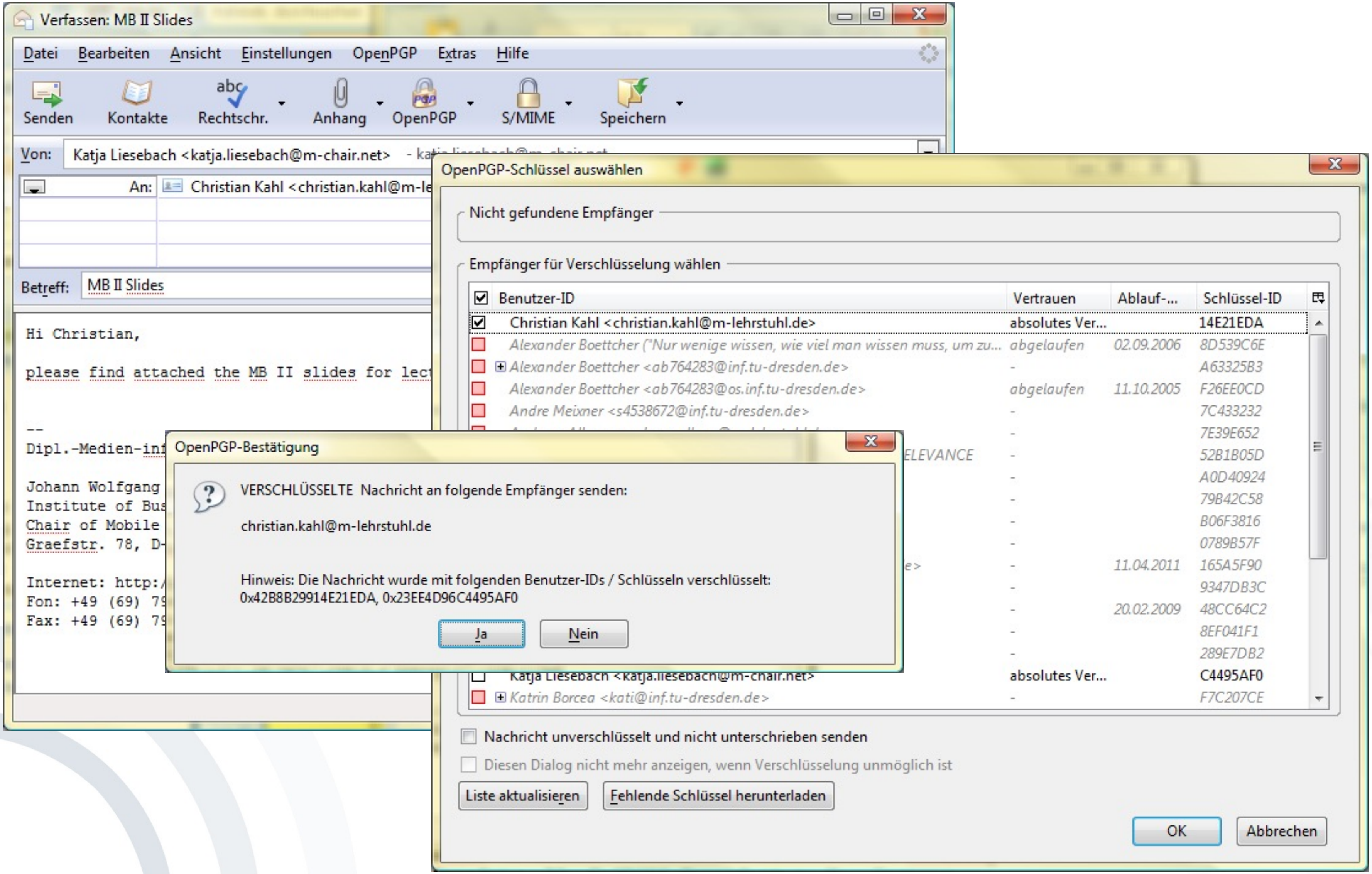
Agenda

- Introduction
- Symmetric Cryptosystems
- Public Key Cryptography
  - General Concept
  - Algorithms
  - Hybrid Systems
  - Digital Signature
  - Key Management
  - Example: PGP

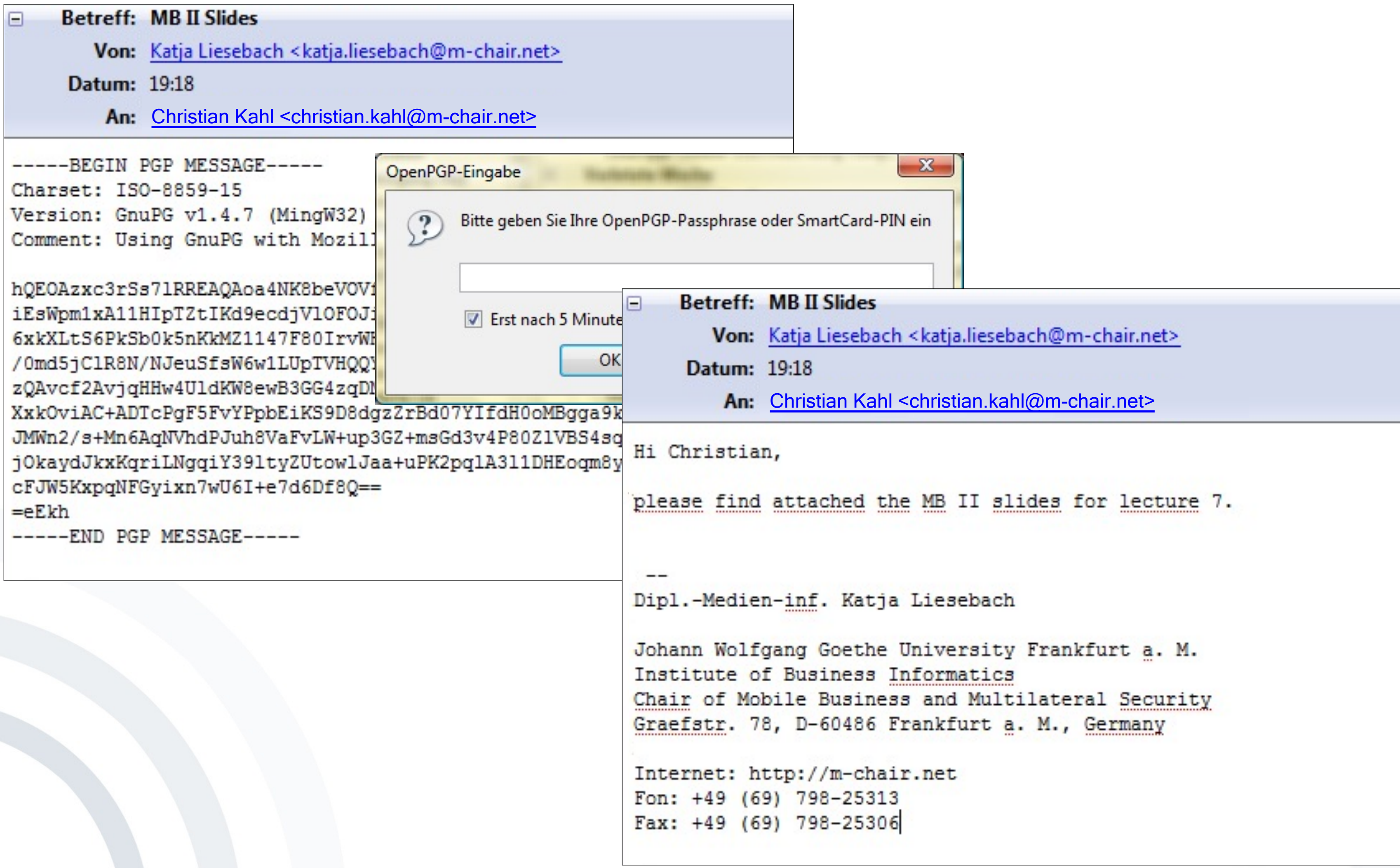
Protection of Email
Example PGP

- PGP = Pretty Good Privacy
  - De facto-Standard for freely accessible e-mail encryption systems on the Internet
  - First implementation by Phil Zimmermann
  - Long trial against Phil Zimmermann because of suspicion of violation of export clauses
  - In U.S., free version in cooperation with MIT (agreement with RSA because of the patent)
  - Meanwhile commercialized: www.pgp.com
  - Gnu Privacy Guard (GPG): non-commercial Open Source variant (OpenPGP, RFC2440)

OpenPGP: Encrypt Message



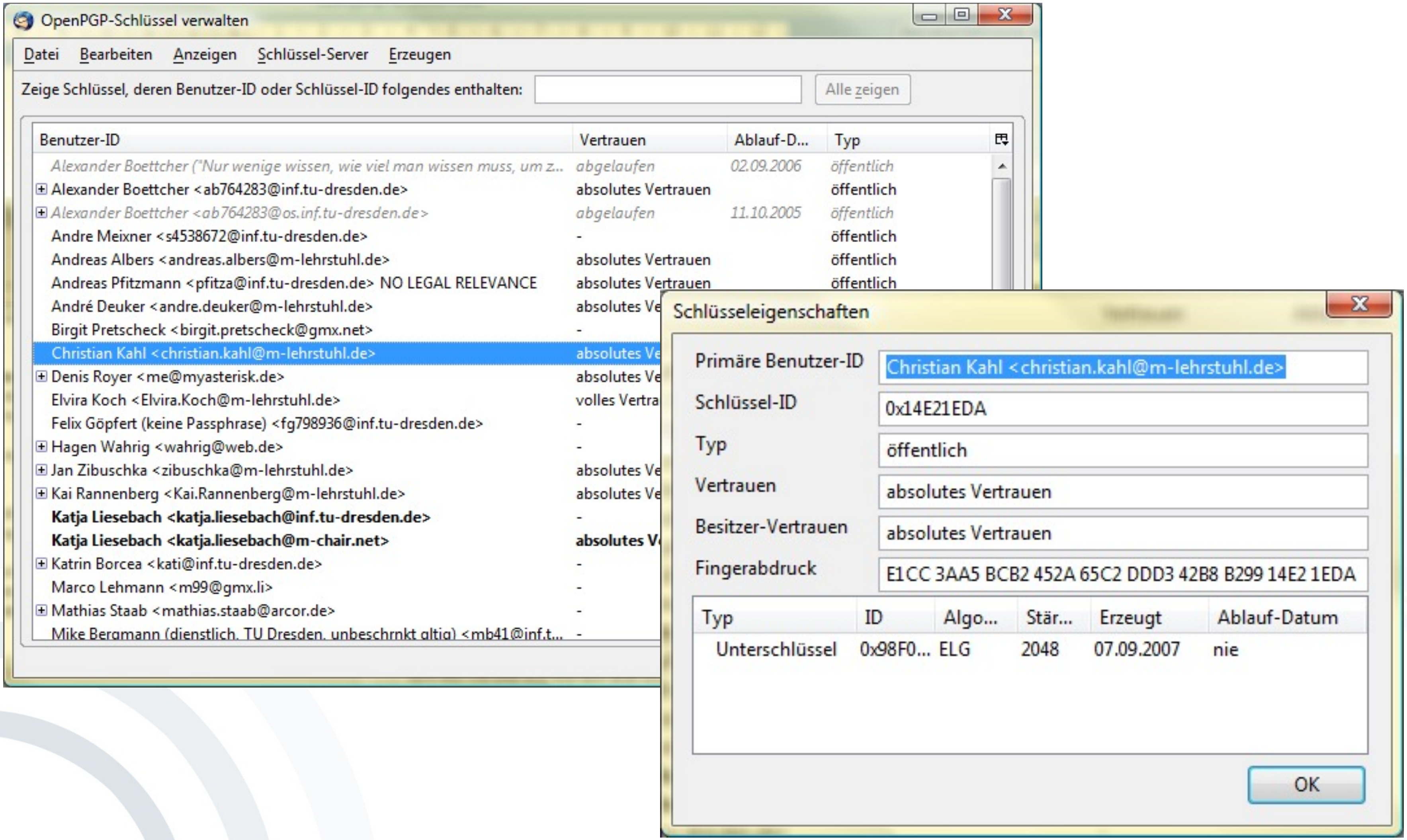
OpenPGP: Decrypt Message



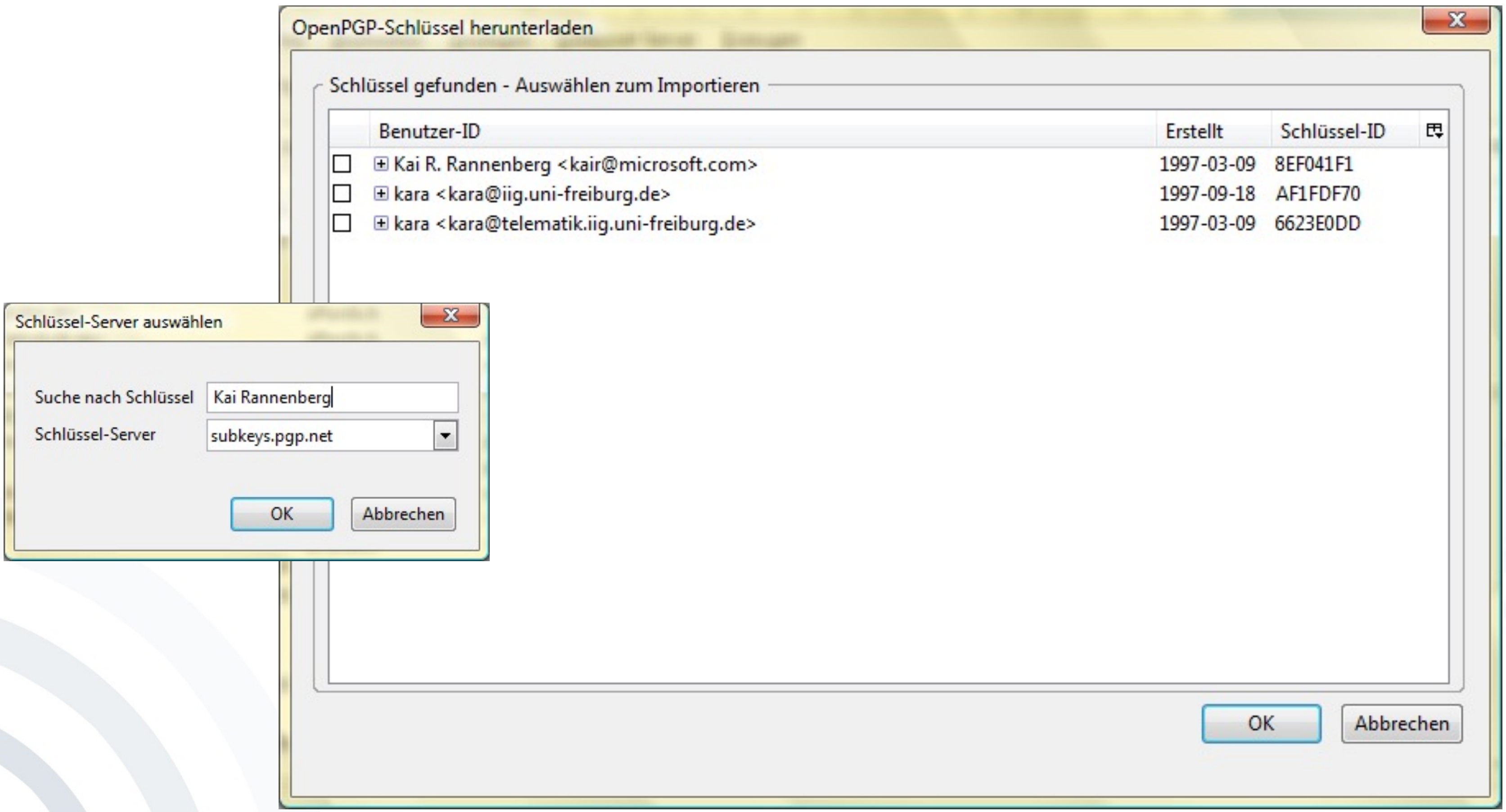
PGP-Certification of Keys

- Certification of public keys by users: “Web of Trust”
- Differentiation between ‘validity’ and ‘trust’
  - ‘Trust’ :
    trust that a person / an institution signs keys only if their authenticity has really been checked
  - ‘Validity’ :
    A key is valid for me if it has been signed by a person / an institution I trust (ideally by myself).
- Support through key-servers:
  - Collection of keys
  - Allocation of ‘validity’ and ‘trust’ remains task of the users
- Path server:
  Finding certification paths between keys

OpenPGP: Key Management

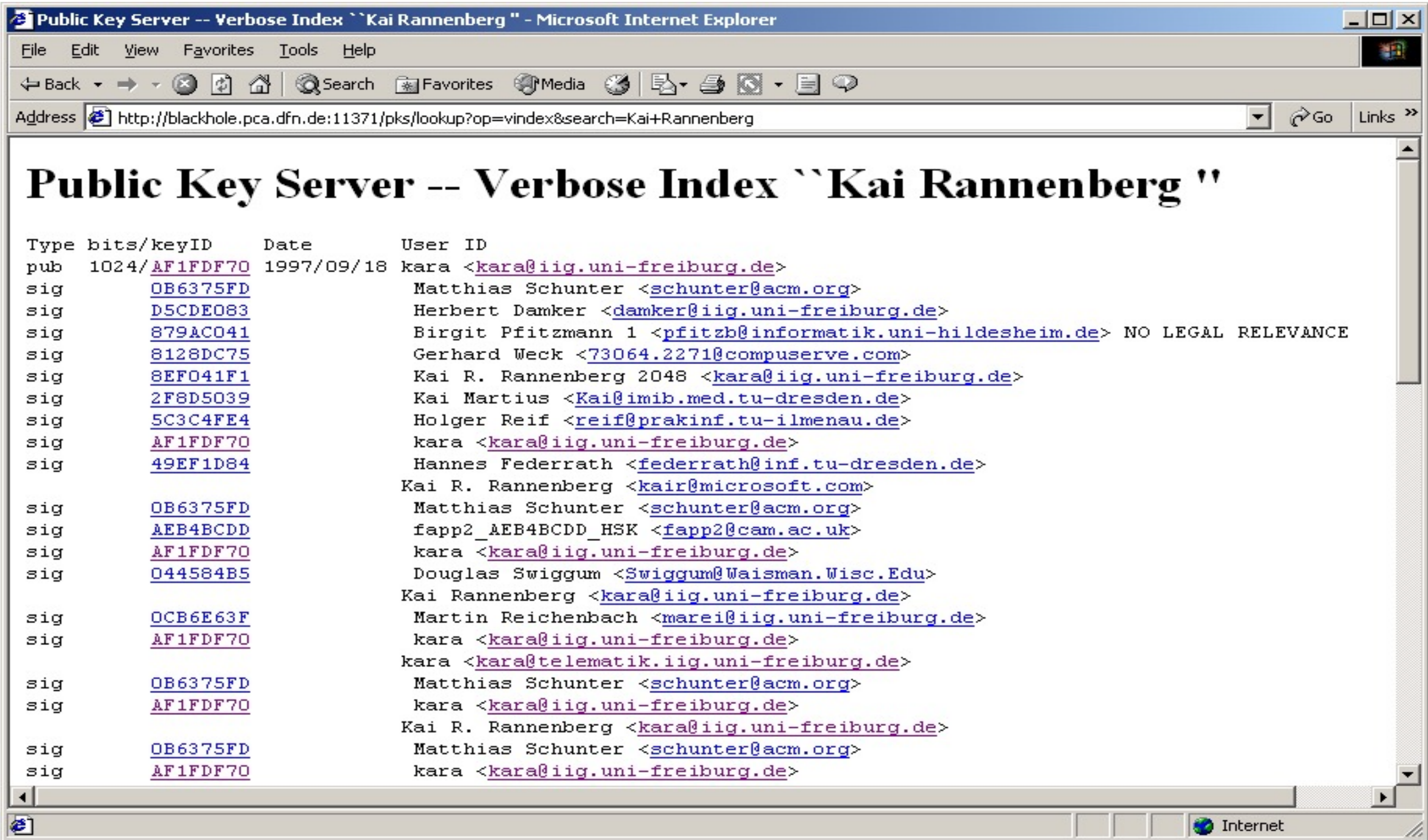


Key-Server



OpenPGP-Schlüssel herunterladen
Schlüssel gefunden - Auswählen zum Importieren
Benutzer-ID
Erstellt
Schlüssel-ID
Kai R. Rannenberg <kair@microsoft.com>
1997-03-09
8EF041F1
kara <kara@iig.uni-freiburg.de>
1997-09-18
AF1FDF70
kara <kara@telematik.iig.uni-freiburg.de>
1997-03-09
6623E0DD
OK
Abbrechen
Schlüssel-Server auswählen
Suche nach Schlüssel
Kai Rannenberg
Schlüssel-Server
subkeys.pgp.net
OK
Abbrechen

PGP: Public Key Catalogs



Public Key Server -- Verbose Index ``Kai Rannenberg '' - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites Media

Address http://blackhole.pca.dfn.de:11371/pks/lookup?op=vindex&search=Kai+Rannenberg Go Links

Public Key Server -- Verbose Index ``Kai Rannenberg ''

```
Type bits/keyID     Date       User ID
pub  1024/AF1FDF70 1997/09/18 kara <kara@iig.uni-freiburg.de>
sig       0B6375FD             Matthias Schunter <schunter@acm.org>
sig       D5CDE083             Herbert Damker <damker@iig.uni-freiburg.de>
sig       879AC041             Birgit Pfitzmann 1 <pfitzb@informatik.uni-hildesheim.de> NO LEGAL RELEVANCE
sig       8128DC75             Gerhard Weck <73064.2271@compuserve.com>
sig       8EF041F1             Kai R. Rannenberg 2048 <kara@iig.uni-freiburg.de>
sig       2F8D5039             Kai Martius <Kai@imib.med.tu-dresden.de>
sig       5C3C4FE4             Holger Reif <reif@prakinf.tu-ilmenau.de>
sig       AF1FDF70             kara <kara@iig.uni-freiburg.de>
sig       49EF1D84             Hannes Federrath <federrath@inf.tu-dresden.de>
                              Kai R. Rannenberg <kair@microsoft.com>
sig       0B6375FD             Matthias Schunter <schunter@acm.org>
sig       AEB4BCDD             fapp2_AEB4BCDD_HSK <fapp2@cam.ac.uk>
sig       AF1FDF70             kara <kara@iig.uni-freiburg.de>
sig       044584B5             Douglas Swiggum <Swiggum@Waisman.Wisc.Edu>
                              Kai Rannenberg <kara@iig.uni-freiburg.de>
sig       0CB6E63F             Martin Reichenbach <marei@iig.uni-freiburg.de>
sig       AF1FDF70             kara <kara@iig.uni-freiburg.de>
                              kara <kara@telematik.iig.uni-freiburg.de>
sig       0B6375FD             Matthias Schunter <schunter@acm.org>
sig       AF1FDF70             kara <kara@iig.uni-freiburg.de>
                              Kai R. Rannenberg <kara@iig.uni-freiburg.de>
sig       0B6375FD             Matthias Schunter <schunter@acm.org>
sig       AF1FDF70             kara <kara@iig.uni-freiburg.de>
```

Internet

- Network of public-key servers:
  - www.cam.ac.uk.pgp.net/pgpnet/email-key-server-info.html
  - http://pgp.mit.edu/

PGP: Practical Attacks and Weaknesses

- Brute-Force-Attacks on the pass phrase
  - PGPCrack for conventionally encrypted files
- Trojan horses, changed PGP-Code
  - e.g. predictable random numbers, encryption with an additional key
- Attacks on the computer of the user
  - Not physically deleted files
  - Paged memory
  - Keyboard monitoring
- Analysis of electromagnetic radiation
- Non-technical attacks
- Confusion of users [Whitten, Tygar 1999]

Remark

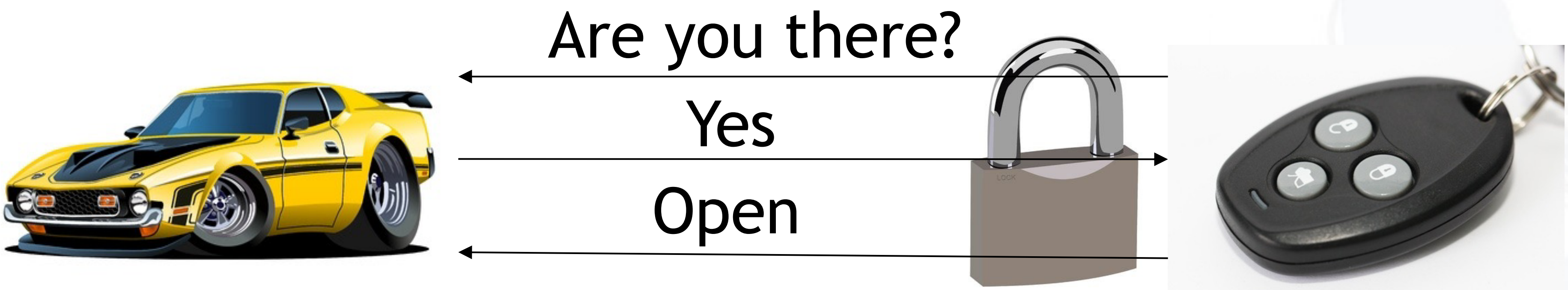
“Anybody who asserts that a problem is readily solved by encryption, understands neither encryption nor the problem.”

(Roger Needham / Butler Lampson)

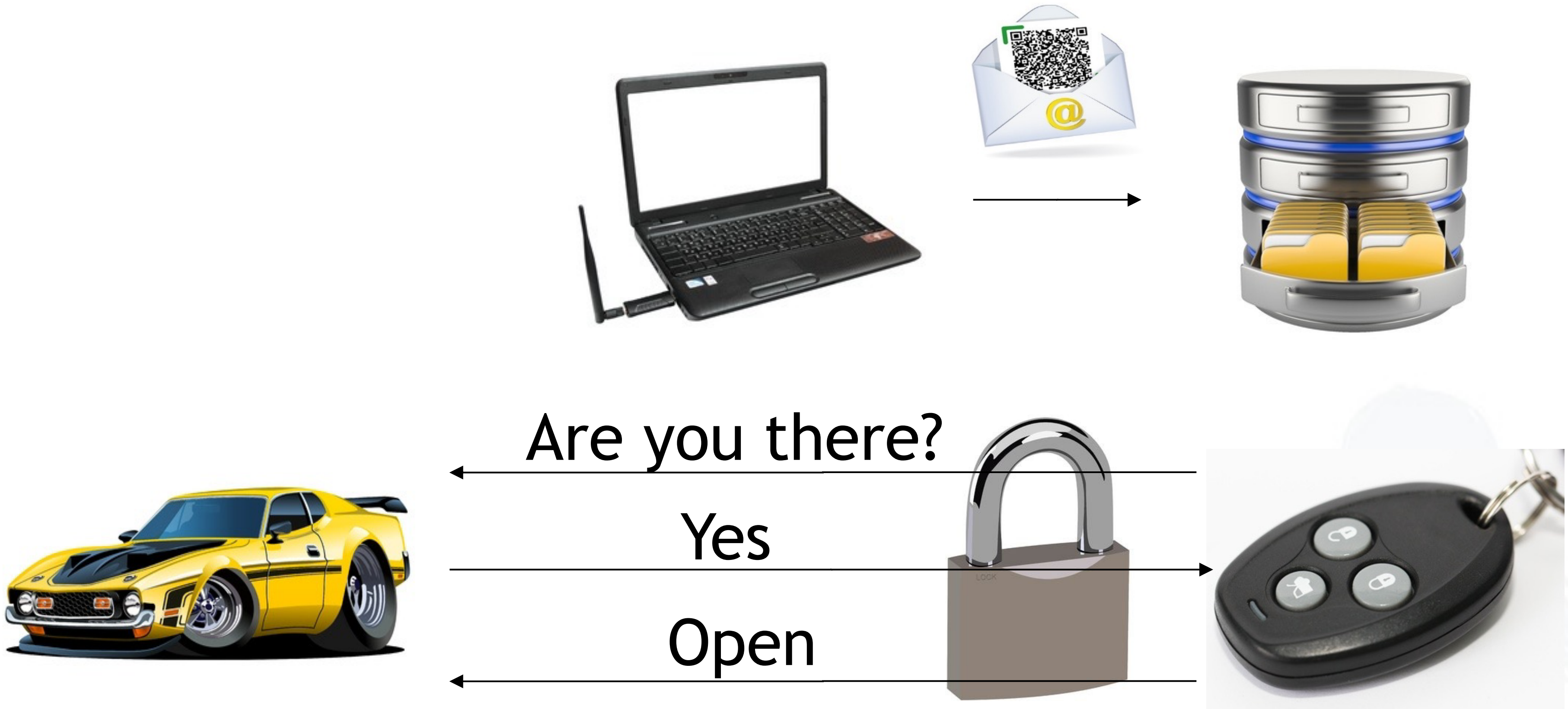
[Marshall Symposium 1998] [Randell 2004]

Example: Keyless Entry System

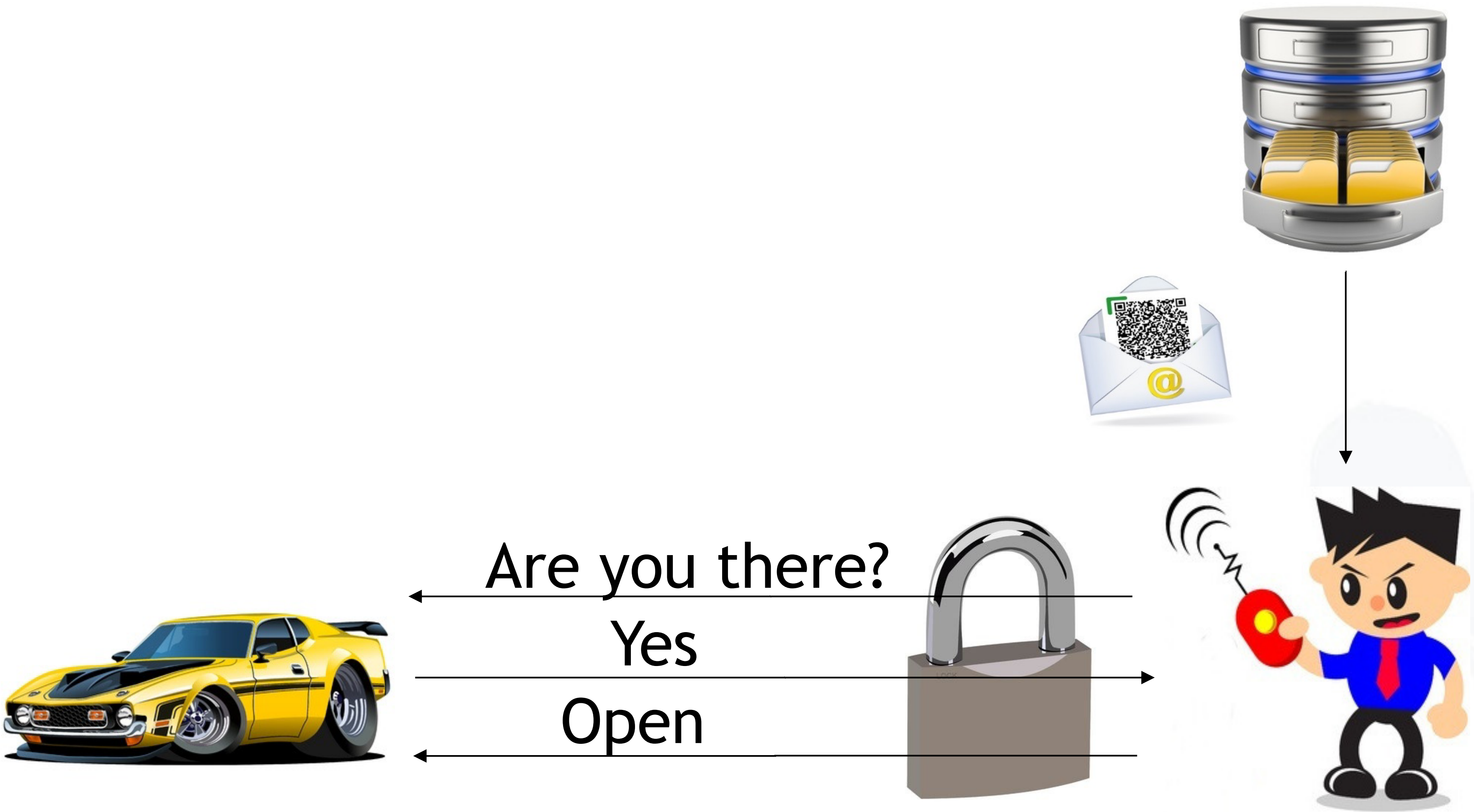
- Solution: Protect communication with crypto?
- e.g. symmetric cryptography + hash/signature



Replay Attack: Eavesdrop

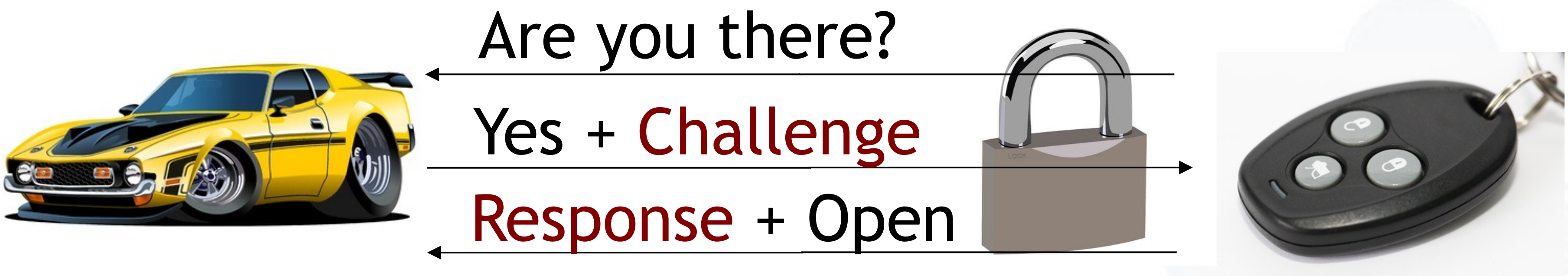


Replay Attack: Replay



Replay Attack: Solution

- e.g. Challenge-Response helps



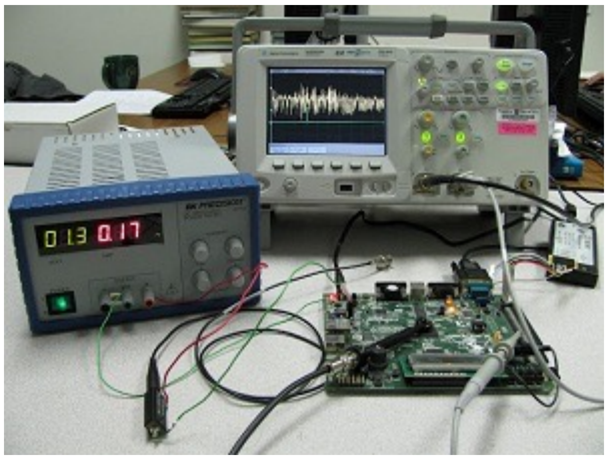
Side-Channel Attacks I

- A secure cryptoalgorithm does not imply that the implementation is also secure

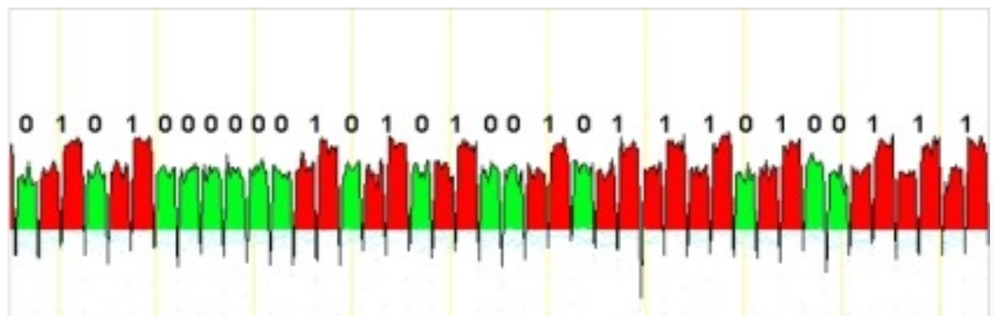
Source: Eran Tromer

Side-Channel Attacks II

- Side-Channels: Time, Power, Noise, Radiation, ...



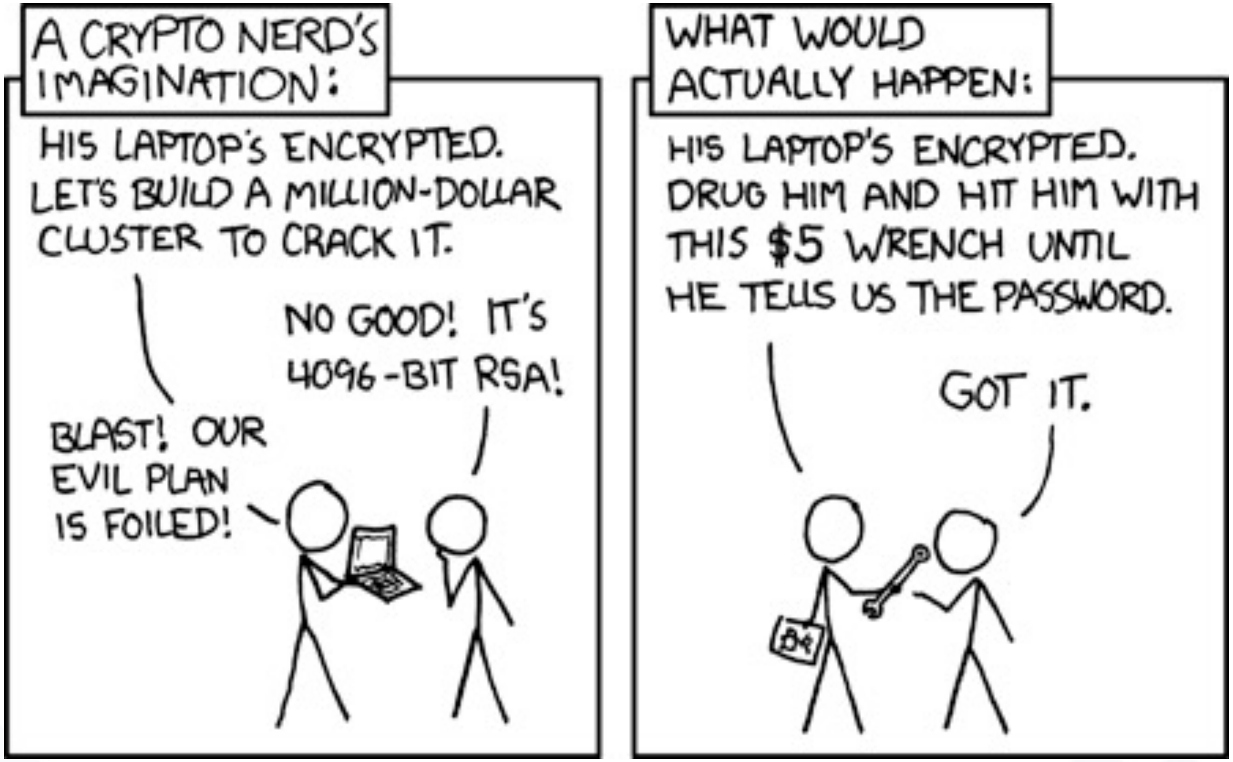
Source: CESCA



Source: Gilbert Goodwill

- Other data (side-channel) leaks information
- Conclusion on processed bits possible

The Human Element



Source: https://xkcd.com/538/

Human Element: Behavior and Passwords

1. Florencio, D. & Herley, C., 2007. A large-scale study of web password habits. *Proceedings of the 16th international conference on World Wide Web - WWW '07*, p.657. Available at: http://portal.acm.org/citation.cfm?doid=1242572.1242661.
2. Florêncio, D., Herley, C. & Coskun, B., 2007. Do strong web passwords accomplish anything? *Proceedings of the 2nd USENIX workshop on Hot topics in security (HOTSEC'07)*, p.10. Available at: http://portal.acm.org/citation.cfm?id=1361419.1361429.
3. Norberg, P.A., Horne, D.R. & Horne, D.A., 2007. The Privacy Paradox: Personal Information Disclosure Intentions versus Behaviors. *Journal of Consumer Affairs*, 41(1), pp.100–126.

Literature

- Bishop, M. (2005)
  Introduction to Computer Security, Addison Wesley, Boston, pp. 97-116.
- Cremers, Cas, et al. "Distance hijacking attacks on distance bounding protocols." 2012 IEEE Symposium on Security and Privacy. IEEE, 2012.
- Diffie, W. and Hellman, M. E. (1976)
  New Directions in Cryptography, *IEEE Transactions on Information Theory* (22:6),
  pp. 644-654.
- Federrath, H. and Pfitzmann, A. (1997)
  Bausteine zur Realisierung mehrseitiger Sicherheit, in: G. Müller and A. Pfitzmann (Eds.): *Mehrseitige Sicherheit in der Kommunikationstechnik,* Boston, Addison Wesley, pp. 83-104.
- The Marshall Symposium: Address Roger Needham, May 29, 1998, Rackham School of Graduate Studies, University of Michigan web.archive.org/web/20081201182254/http:/www.si.umich.edu/marshall/docs/p201.htm, accessed 2015-04-15.
- Randell, B. (2004) *Brief Encounters*; Pp. 229-235 in: Andrew Herbert, Karen Spärck Jones: Computer Systems: Theory, Technology, and Applications; New York, Springer 2004
- Rivest, R. L.; Shamir, A. and Adleman, L. (1978)
  A Method for Obtaining Digital Signatures and Public Key Cryptosystems, *Communications of the ACM* (21:2), pp. 120-126.
- Whitten, A. and Tygar, J. (1999) *Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0.* In: Proceedings of the 9th USENIX Security Symposium, August 1999, www.gaudior.net/alma/johnny.pdf