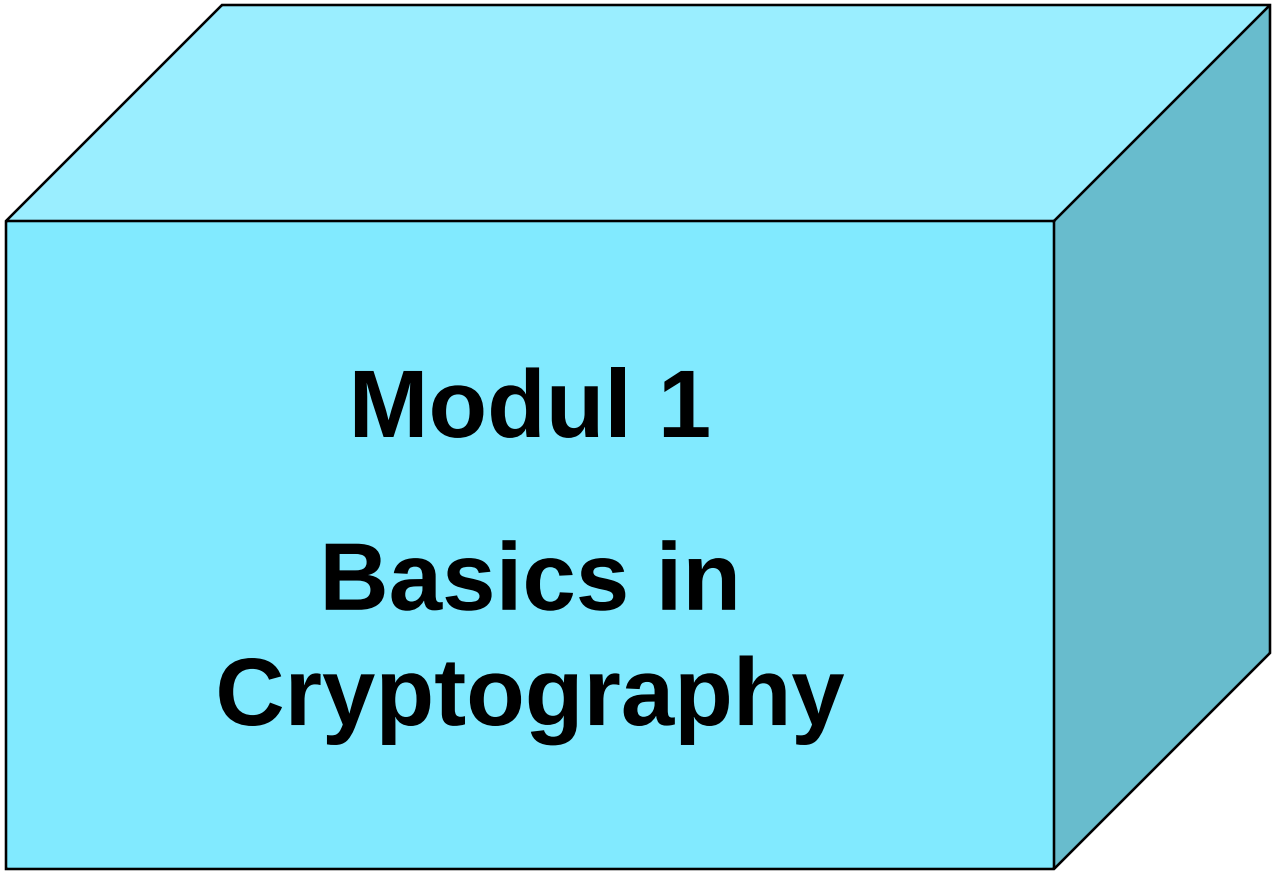


© M. Leischner

Objectives of Cryptography

- **Privacy:**
  Assure confidentiality of information
- **Integrity:**
  Assure retention of information, i.e. no unauthorized modification
- **Authentication:**
  Identify for certain who is communicating with you resp. verify the origin
- **Accountability:**
  Assure identification who did what and when
- **No-repudiation:**
  The ability to provide proof of the origin or delivery of data

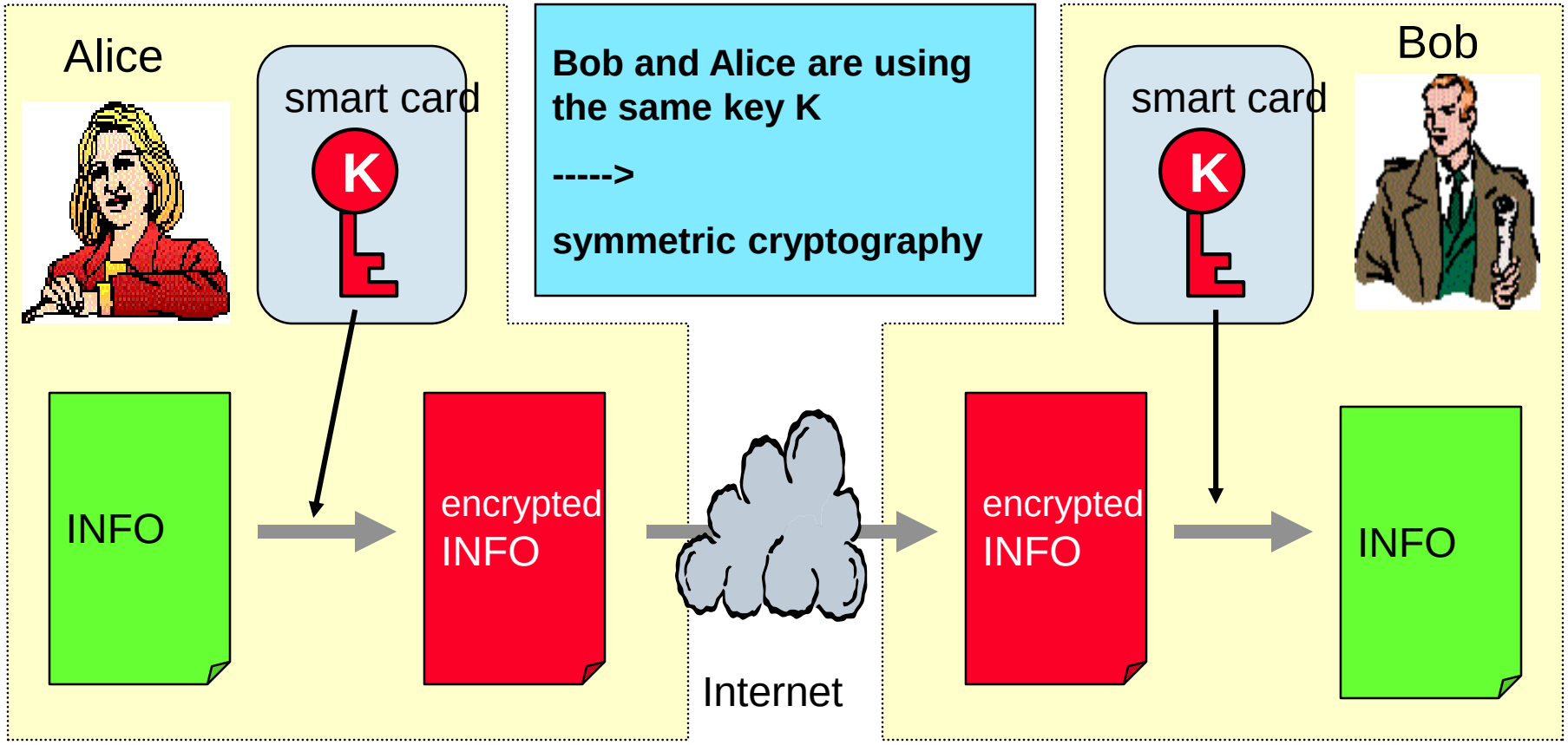
© M. Leischner

One Time Pad

- Random Numbers + linking by XOR
- pros: provable safety
- cons: Key exchange, there are no random numbers
- Application case: TAN-method in the context of Home banking

© M. Leischner

Symmetric Cryptography



Symmetric block encryption - symmetric cryptography

Let K be a Key and ***enc*** and ***dec*** a encryption function such that

$$dec = enc^{-1}$$

then

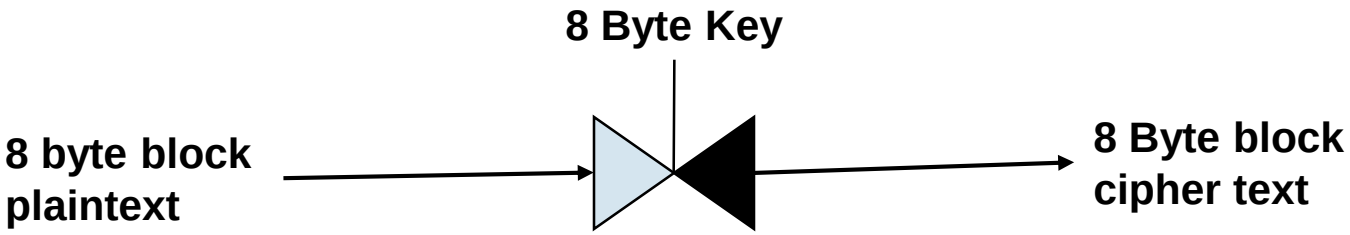
plaintext = ***dec***(K, ***enc*** (K, plaintext))

key space (set of all keys) has ? elements
(this implies a key length up to ? bit)

→ there are approximately infinite methods to construct a symmetric block encryption algorithm

© M. Leischner

The symmetric block encryption algorithm DES



- Currently DES is the most important symmetric block encryption algorithm.
- To be honest: only 56 bit of the DES 8 Byte key is used.
- A saver variant of DES is Triple-DES (112 bit key).
- In the future the DES algorithm will be substituted by the recent AES (Advanced Encryption Standard) algorithm.
- DES is a very fast algorithm and it can easily be hardware-implemented in a smart card (crypto processor unit)
- Other block encryption algorithms:
  IDEA (used by PGP, 128 key length), Blowfish, Twofish, RC2, RC5, RC6 ….

© M. Leischner

DES and Triple-DES

- **DES key**

$$k = (k_1, \ldots, k_{56}) \in \{0,1\}^{56}$$

- **DES encryption**

$$E^1{}_k : \{0,1\}^{64} \rightarrow \{0,1\}^{64}$$

- **DES decryption**

$$D^1{}_k : \{0,1\}^{64} \rightarrow \{0,1\}^{64}$$

- **formula**

$$E^1{}_{(k_1,\ldots,k_{56})} = D^1{}_{(k_{56},\ldots,k_1)}$$

- **Triple DES encryption**

$$E^3{}_k := E^1{}_{k_l} \circ D^1{}_{kr} \circ E^1{}_{kl}$$

$$k = (kl, kr)$$

- **Triple DES decryption**

$$D^3{}_k := D^1{}_{k_l} \circ E^1{}_{kr} \circ D^1{}_{kl}$$

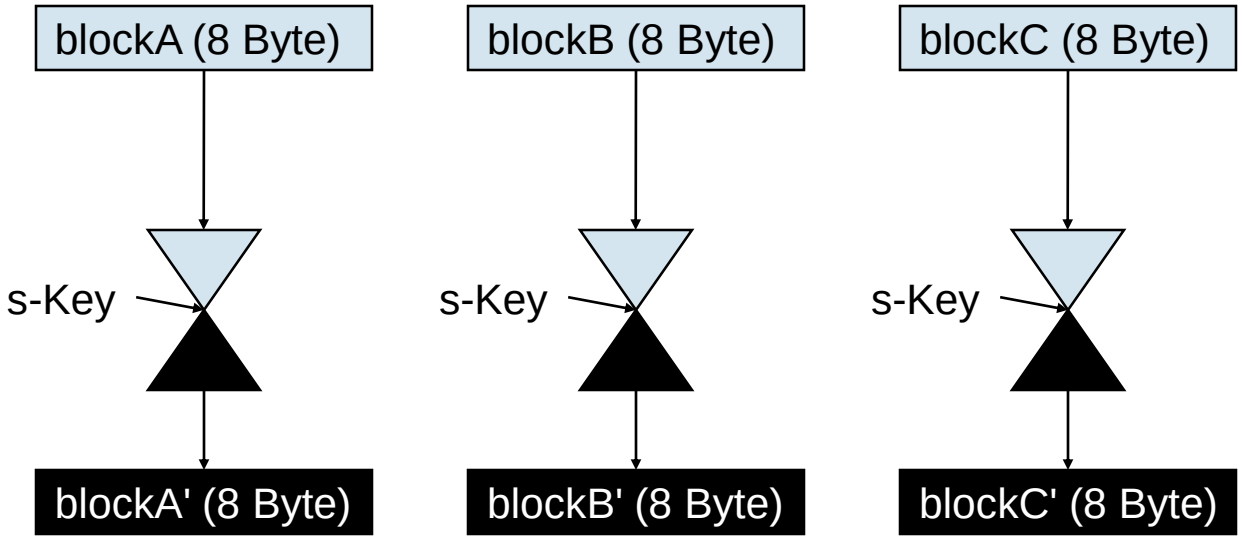
© M. Leischner

AES (Advanced Encryption Standard)

- in October 2000, the NIST (National Institute of Standards and Technology) announced the approval of a new secret key cipher standard chosen among 15 candidates
- block and key length of 128, 192 or 256
- Very high encryption speed (200 MBit/sec using a 1GB-PC)
- Very efficiently to implement (even on a 8-Bit smart card (math. calculation is based on the Galois field (→ 8 bit numbers).
- Universally applicable: One way hash, MAC, pseudo random number generator
- AES can also be used by smart cards

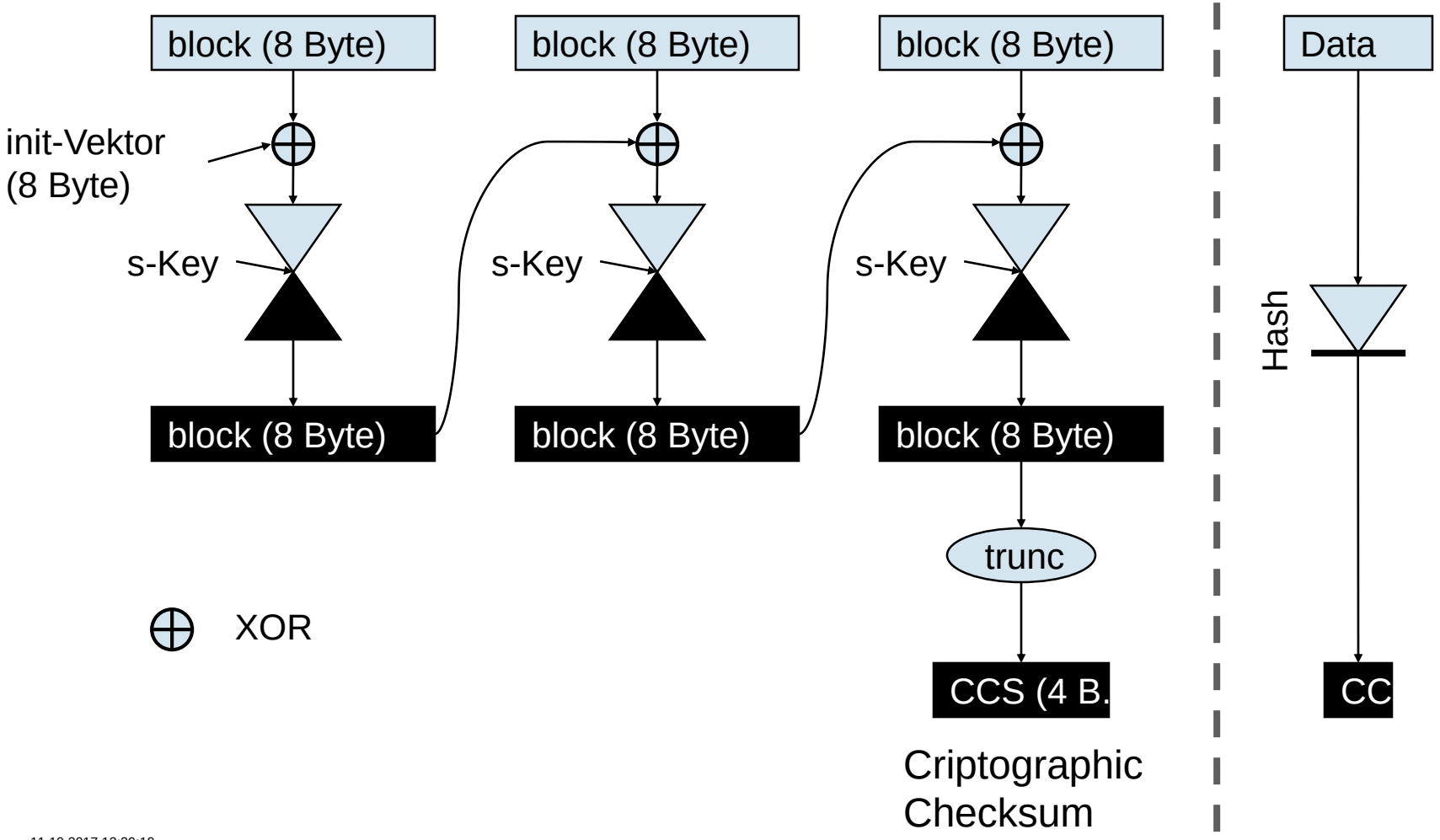
© M. Leischner

Operating Mode Electronic Code Book (ECB)



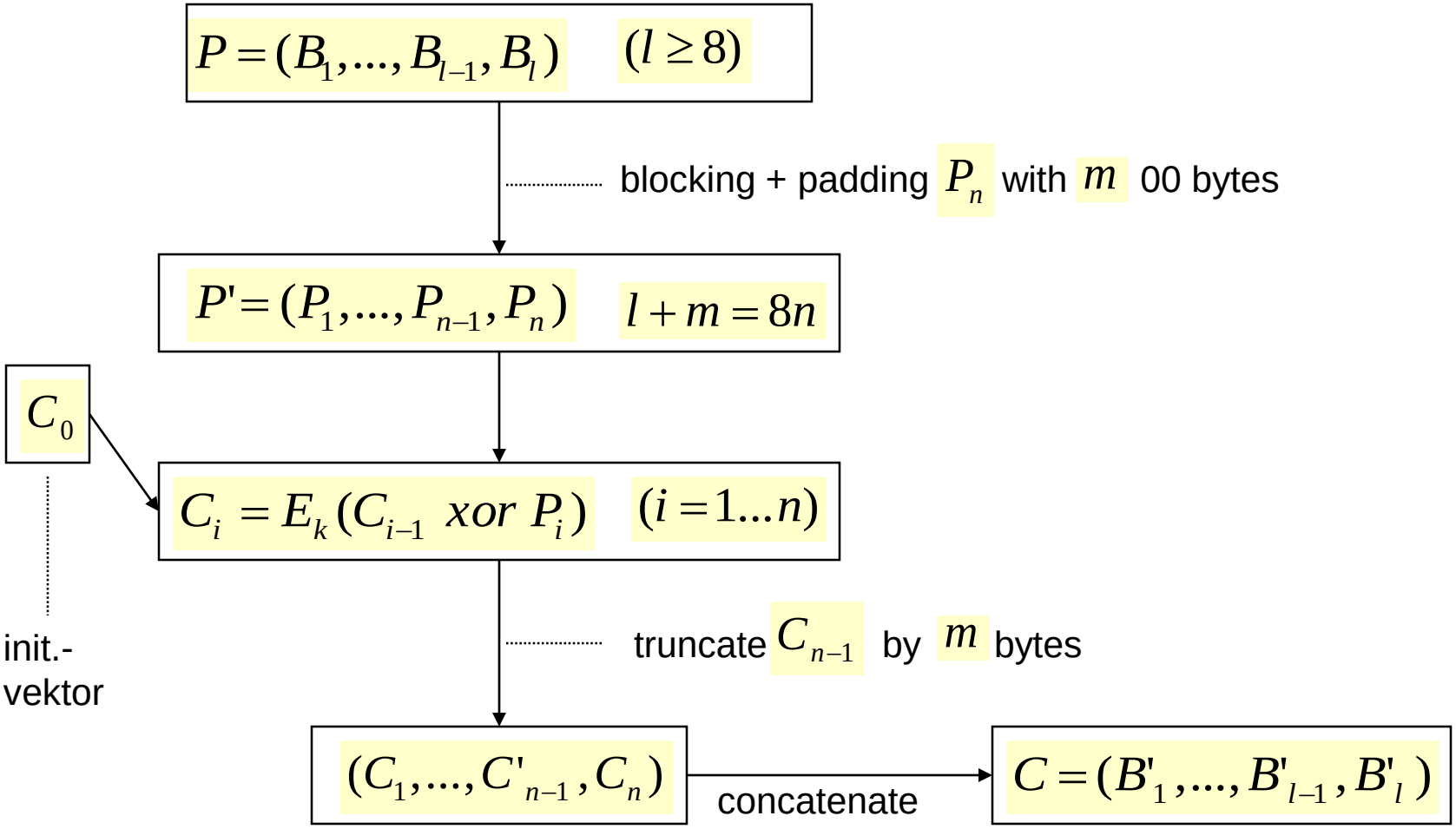
- **very simple**
- **It is not very safe:**
  If block A equals block B then also the cipher blocks A' and B' are equal.

Operating Mode Cipher Block Chain (CBC) and Hashing



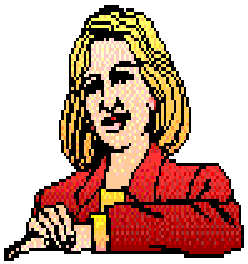
© M. Leischner

cipher block chaining (used by the BasicCard)

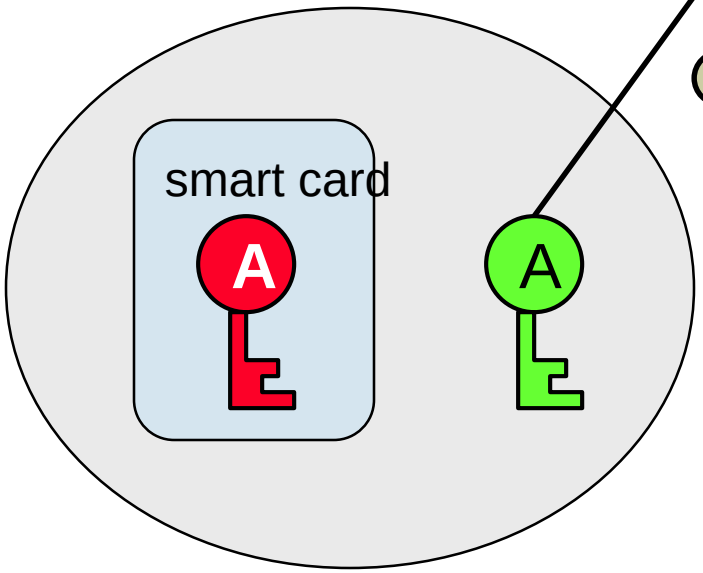


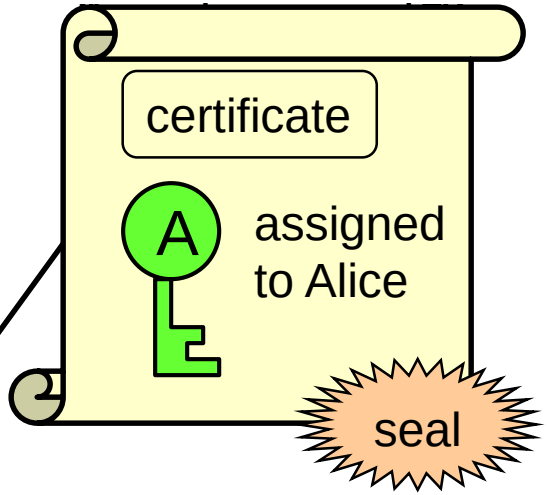
Public Key Cryptography

Public-Key-Kryptographie



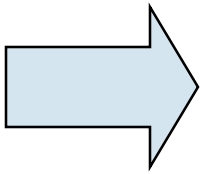
Alice





Alice has two keys

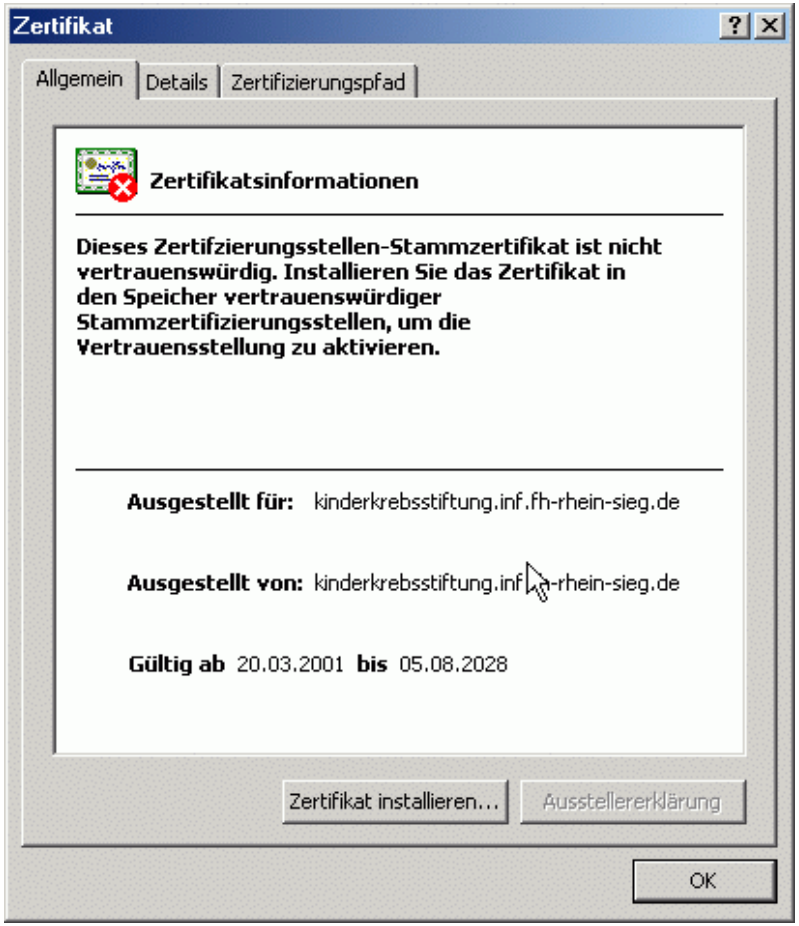
- **a secret key (e.g. on a smart card)**
- **a public key (not secret, can be stored in a public directory or on her homepage**



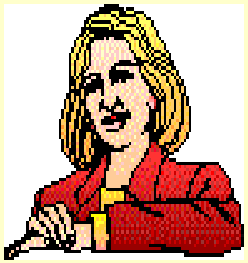
To encrypt ***and*** decrypt information, you need ***both*** keys

Certificate

- Tuple (owner, public key, SignatureCA(owner, public key))
- Additional data: Angabe des eingesetzten Signaturverfahrens, Beginn und Ende der Gültigkeit, Seriennummer, Attribute, Namen der ausstellenden Certification Authority, Einschränkungen der Nutzung des Signaturschlüssels
- Time Stamp =SignatureCA (hash value, time)
- Standard: X509v3



Public Key Cryptography - The Concept of Confidentiality



Alice

smart card

A

public keys
(to be used by everyone)

A B

smart card

B

Bob

B

INFO

INFO

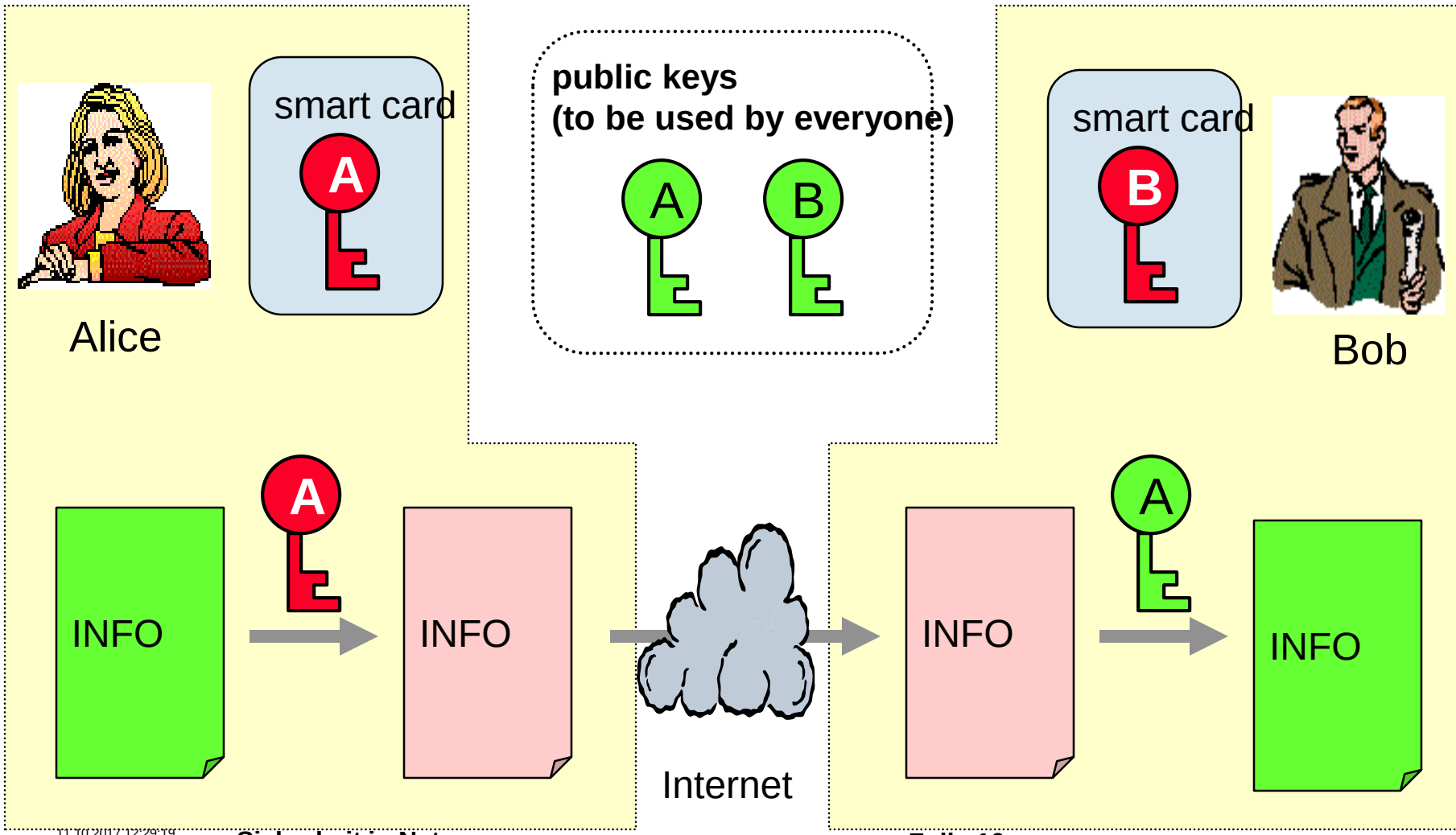
Internet

INFO

B

INFO

Public Key Cryptography - The Concept of Digital Signatures



Public Key Algorithms

RSA

- based on prime numbers
- typical key length: 1024 bit

Elliptic functions

- based on elliptic functions and finite fields
- typical key length: 160 bit

Pro and cons of public key cryptography:

+ allows sophisticated key management

+ very high safety

- slow algorithms

© M. Leischner

RSA

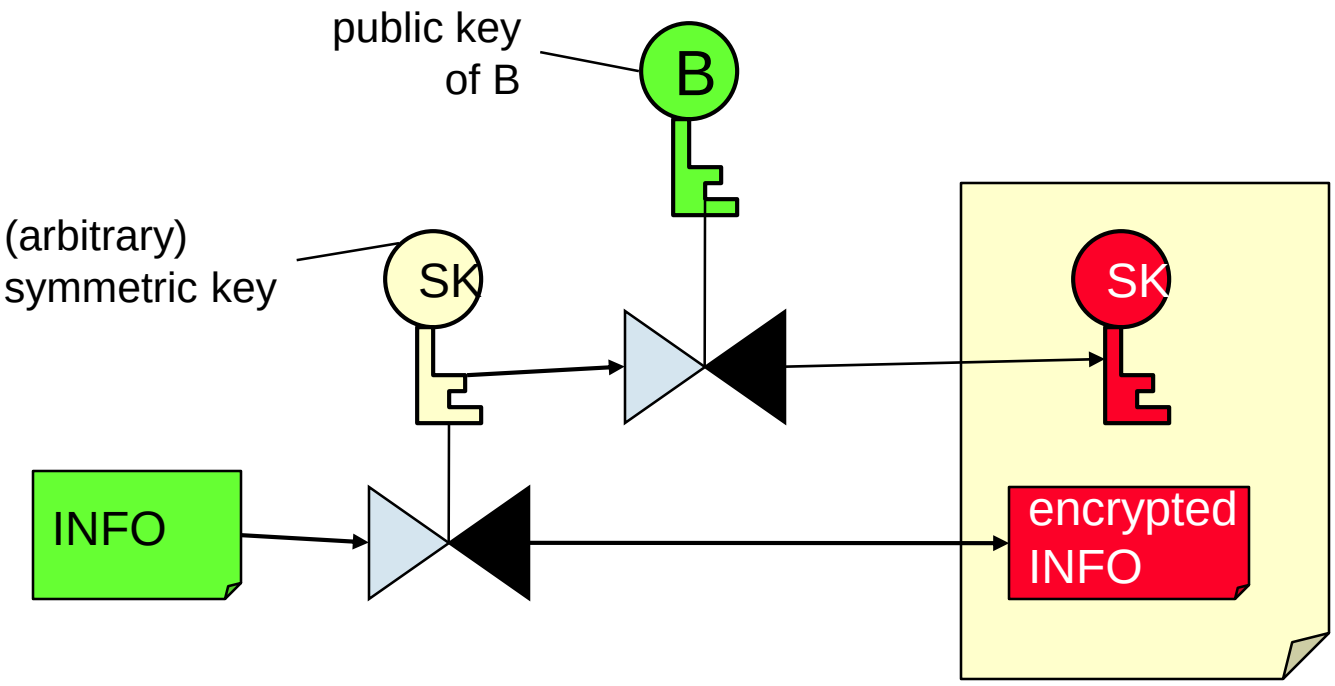
- **Method**
  public key = (e,n)
  private key = (d,n), where $d \equiv e^{-1} \bmod \varphi(n)$ with $\varphi(n) = (p-1) * (q-1)$
  encryption: plain text M (<n) → cipher text $C = M^e \bmod n$
  decryption: cipher text C → plain text $M = C^d \bmod n$
- **no patent (since autumn 2000)**
- **Convention:**
  pk: public key, encryption key
  sk: signature key, secret key, private key, decryption key

© M. Leischner

Hybrid Encryption

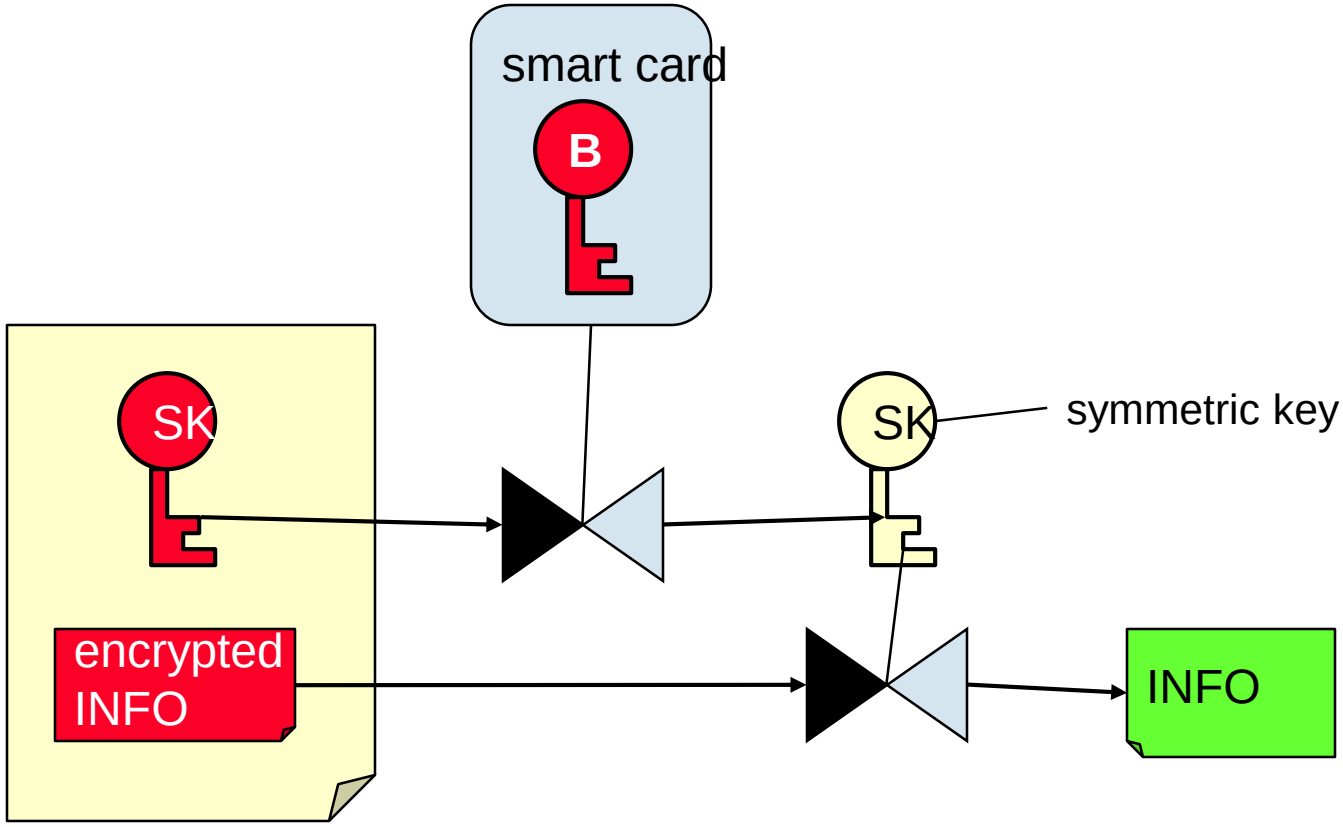
The Concept of Hybrid Encryption

Scenario: A (Alice) wants to encrypt a document INFO for B (Bob)



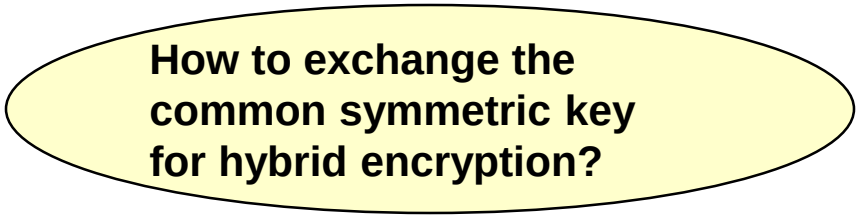
The Concept of Hybrid Decryption

Scenario: B (Bob) wants to read the document sent by A (Alice)



The Problem of Key Management

The problem of key management:

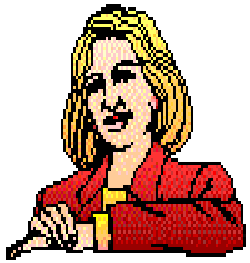


Two Solutions:

- **Solution 1: By sending an encrypted symmetric key**
- **Solution 2: Using the Diffie-Hellman key exchange algorithm**
  --> next slide

© M. Leischner

Diffie-Hellman Key Exchange Algorithm



Alice

smart card

A

public keys
(to be used by everyone)

A B

smart card

B

Bob

Diffie-Hellman key exchange algorithm

Diffie-Hellman key exchange algorithm

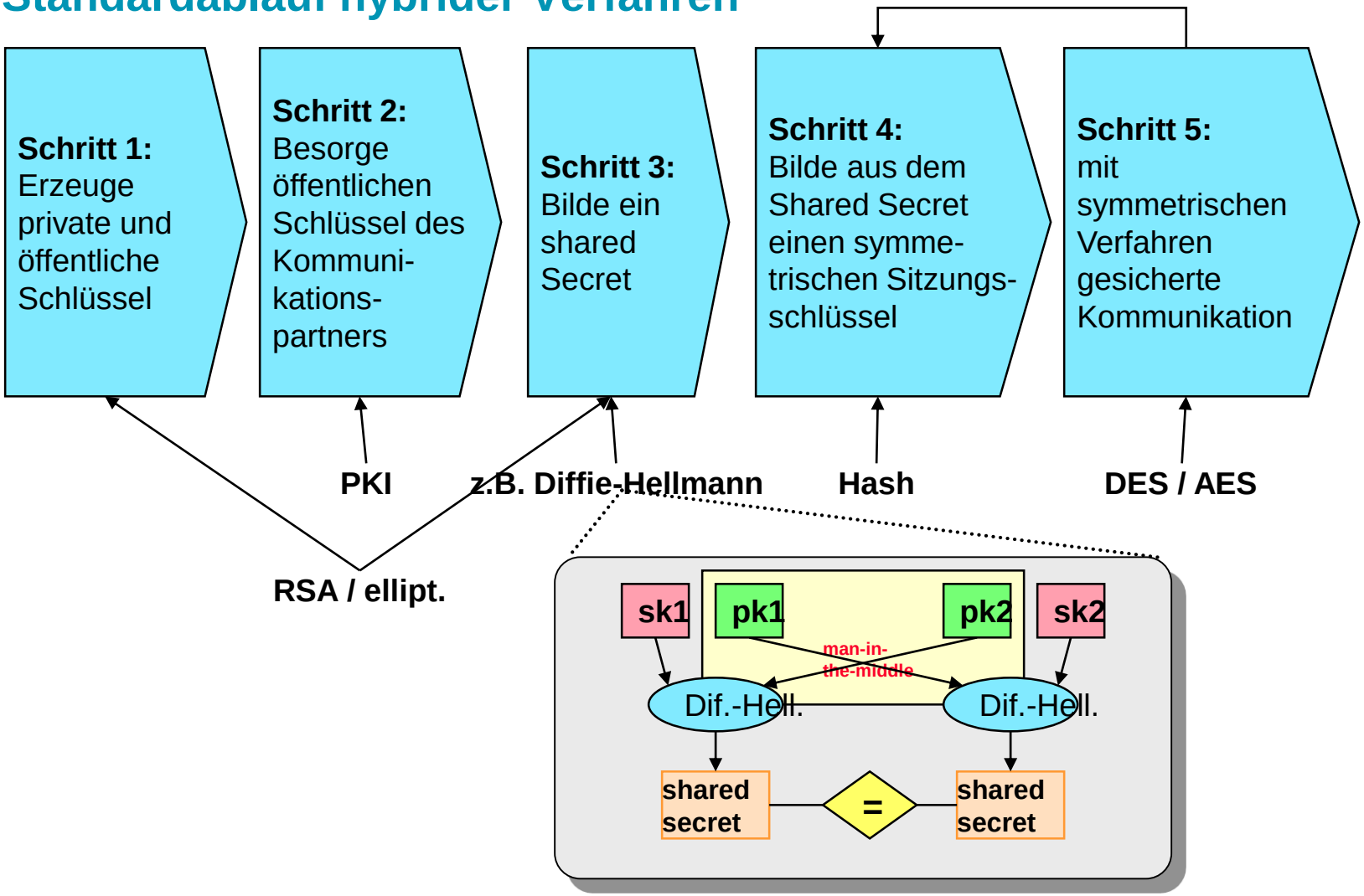
this derived key is often called "a shared secret"

SK

These keys are equal !

SK

Standardablauf hybrider Verfahren



© M. Leischner

hash function (one way function)

- plain text of arbitrary length is mapped to a hash value of fixed length
- It is difficult to find any (x, y) with H(x) = H(y) (strong collision resistant)
- Given x it is difficult to find a y such that H(x) = H(y) (weak collision resistant)
- MD5: 128 Bit hash value (used by PGP, not secure)
- SHA-1: 160 Bit hash value (attacked 2005 , not secure)
- SHA-2: set of hash functions (similar to SHA 1, secure)
- SHA-3: set of hash functions (alternative, dissimilar to SHA 1/2)
- RIPE-MD: 160 Bit hash value (is supposed to be very save)
- Message Authentication Code (MAC) ist eine schlüsselabhängige Einweg-Hashfunktion (→ Integrität + Authentizität der Nachricht), Realisierung durch DES oder AES im CBC-Mode

© M. Leischner

digital signature

- **Method:**
  Hash the document D , encrypt the hash value H using the secret key S
  → the signed document = (D, encS(H))
- **Standards for digital signetures:**
  - **DSA (Digital Signature Algorithm, diskreter Algorithmus, 1024-Bit Schlüssellänge,**
  - **NIST-Standard (National Institute of Standards and Technology)),**
  - **DSS (Digital Signature Standard, Nachfolger DSA, digitalen Beglaubigungsstandard der US-Regierung), RSA (einfacher und beliebter)**

© M. Leischner

Erzeugung von sicheren Zufallszahlen

- Zufallszahlen werden zur Realisierung sicherer Protokolle benötigt
- Die Erzeugung von *echten* Zufallszahlen ist schwierig und mit algorithmischen Verfahren unmöglich:

  John von Neumann:
  Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin.
- jeder „Zufallszahlen“-Generator in einem Computer ist zwangsläufig periodisch und damit nicht zufällig
- Definition:
  Ein Pseudozufallsbitgenerator ist ein deterministischer Algorithmus, der als Eingabe eine echt zufällige Bitfolge (seed) erhält und daraus eine (längere) Bitfolge erzeugt, die den *Eindruck der Zufälligkeit* erweckt.
- Die Güte eines Pseudozufallsbitgenerator kann mathematisch gefasst werden → kryptographisch sicherer Pseudozufallsbitgenerator

© M. Leischner