CoNe Freiburg

IIF INSTITUT FÜR INFORMATIK FREIBURG

Systeme II

7. Sicherheit

Christian Schindelhauer
Technische Fakultät
Rechnernetze und Telematik
Albert-Ludwigs-Universität Freiburg
(Version 14.07.2014)

UNI FREIBURG

RSA

7

- Kodierung
  - Teile Nachricht in Blöcke der Größe 2^{2k} auf
  - Interpretiere Block M als Zahl $0 \leq M < 2^{2k}$
  - Chiffre: $P(M) = M^e \bmod n$
- Dekodierung
  - $S(C) = C^d \bmod n$
- Korrektheit gilt nach dem kleinen Satz von Fermat
  - Für Primzahl p und von p teilerfremde Zahl a gilt:

$$a^p \equiv a \pmod p$$

$2^7 \equiv 2 \bmod p$

$2^7 \bmod 7$

$= 2^7 \bmod 7$

$= 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \bmod$

$4 \cdot 4 \cdot 4 \cdot 2$

16 8

2

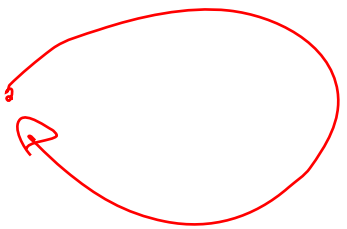
16 mod 7

= 2

$$a^{p} \equiv a \mod p$$

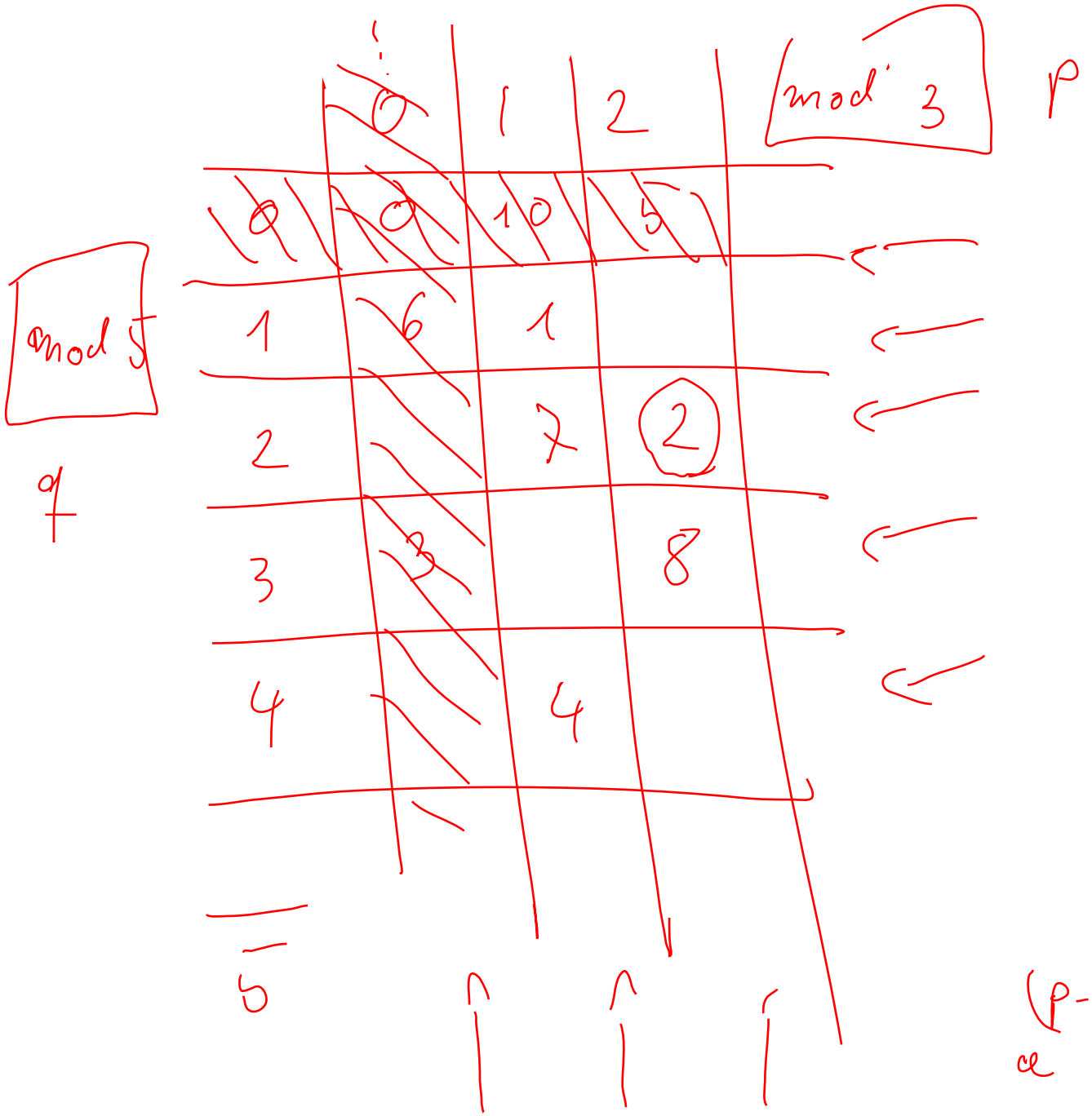
$$a^{p-1} \equiv a^{0} \mod p$$

$$a^{7} \cdot a^{g} \equiv a^{7+g \bmod p-1} \mod p$$



$$\left(a^{7}\right)^{g} \equiv a^{7 \cdot g \bmod p-1} \mod p$$

Chinesischer Restsatz



$0, \dots, 14$

$$a^{p-1} \equiv a^{0} \bmod p$$

$$a^{q-1} \equiv a^{0} \bmod q$$

$$a^{(p-1)(q-1)} \equiv a^{0} \bmod p \cdot q$$

$$\varphi(p \cdot q) = (p-1)(q-1)$$

$$a^{e \cdot d} \equiv a^{e \cdot d \bmod \varphi(n)} \pmod{n}$$

$$e \cdot d \equiv 1$$

$$a^{e \cdot d} \equiv a^{1} \pmod{n}$$

RSA Beispiel

- Bob wählt p=5, q=7
  - n=35, z=24
  - e = 5
  - d= 29
    - e d = 1 mod 24
- Verschlüsselung von 8-Bit-Nachrichten

Bit pattern	m	m	$c=m^e \bmod n$
00001000	12	248 832	17

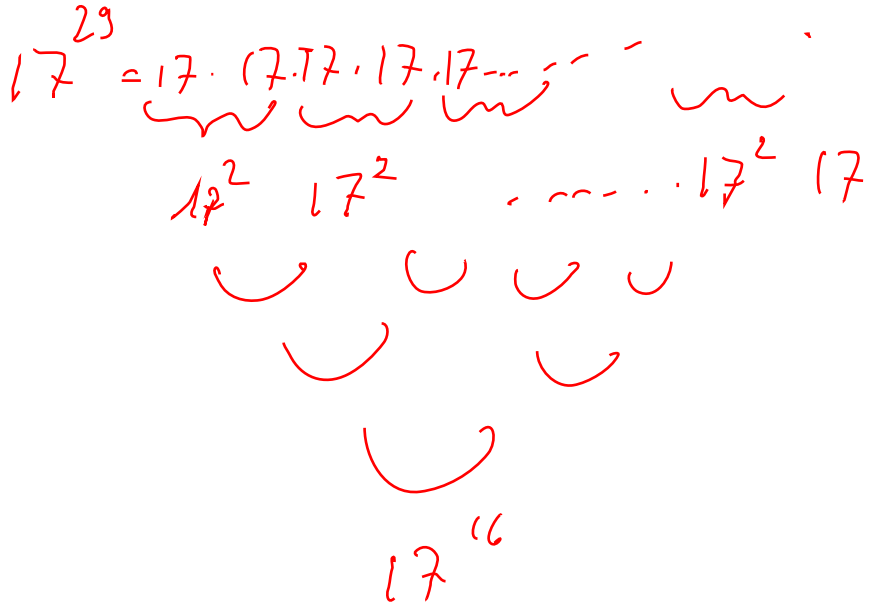
- Entschlüsselung

c	$c^d = 17^{29}$	$m=c^d \bmod n$
17	481968572106750915091411825223071697	12

CoNe Freiburg

RSA Beispiel

- Berechnung von $17^{29} \bmod 35$
- $29 = 11101_2$
  - $17^{29} = 17 \cdot (17^{14})^2 \bmod 35$
  - $17^{14} = (17^7)^2 \bmod 35$
  - $17^7 = 17 \cdot (17^3)^2 \bmod 35$
  - $17^3 = 17 \cdot (17)^2 \bmod 35$
- Einsetzen:
  - $17^3 = 4913 = 13 \bmod 35$
  - $17^7 = 17 \cdot 13^2 = 2873 = 3 \bmod 35$
  - $17^{14} = 3^2 = 9 \bmod 35$
  - $17^{29} = 17 \cdot 9^2 = 1377 = 12 \bmod 35$



CoNe Freiburg

Nachrichtenintegrität

- Erlaubt den Komunikationspartnern die Korrektheit und Authentizität der Nachricht zu überprüfen
  - Inhalt ist unverändert
  - Urheber ist korrekt
  - Nachricht ist keine Wiederholung
  - Reihenfolge der Nachrichten ist korrekt
- Message Digests

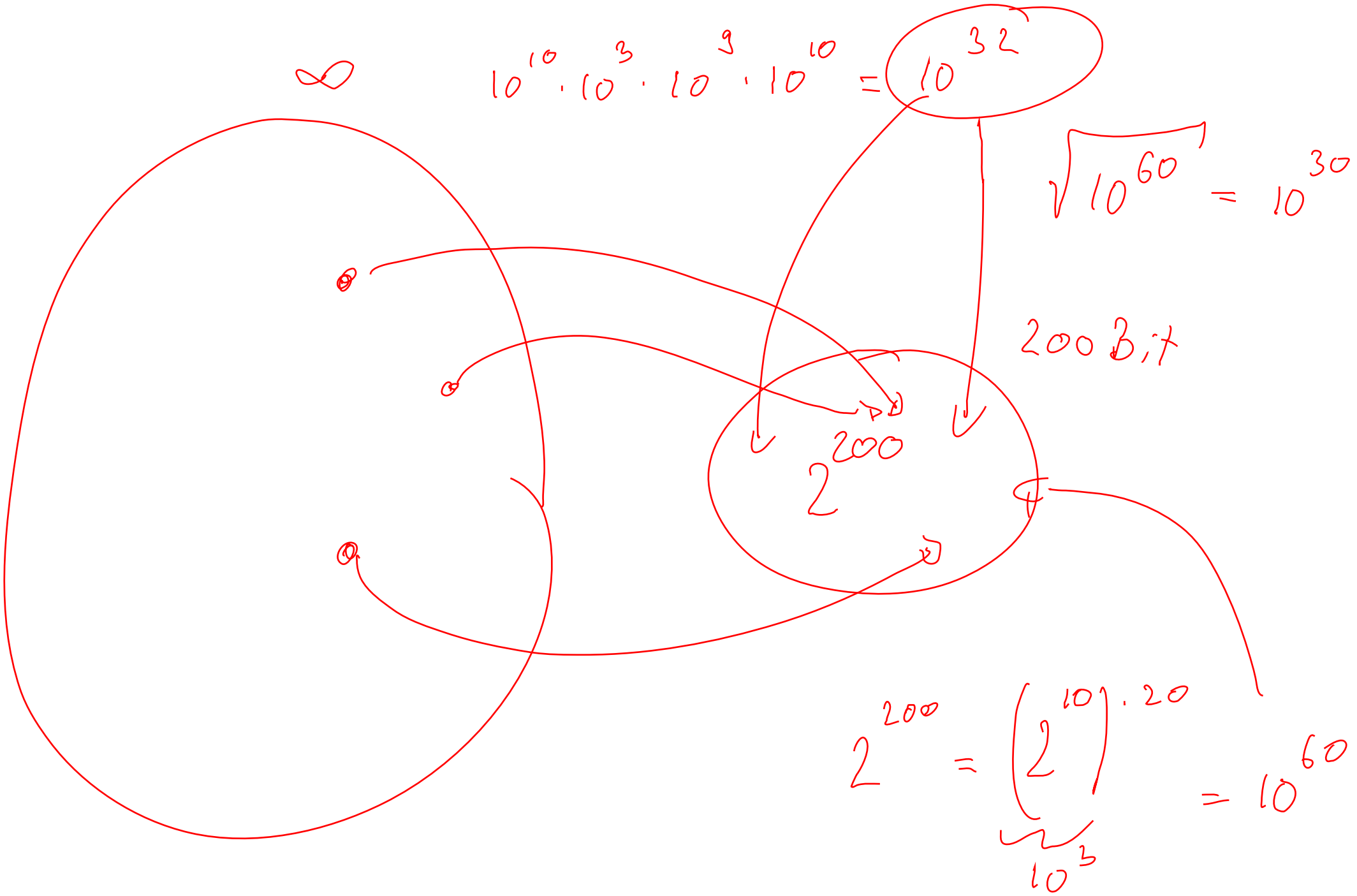
CoNe Freiburg

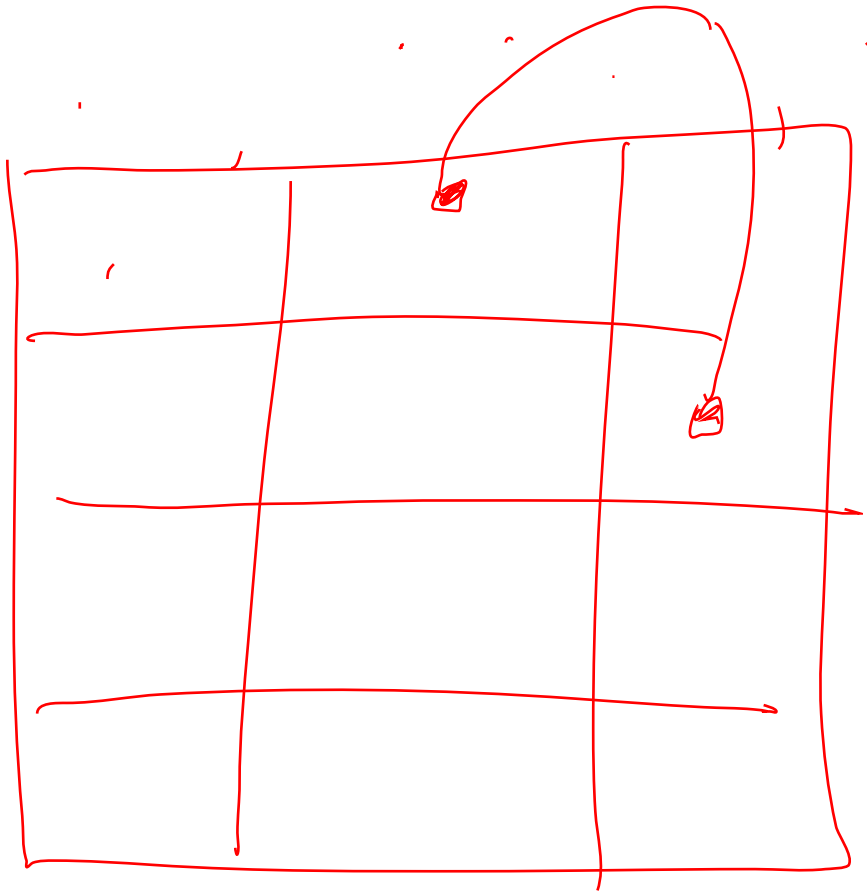
Kryptographische Hash-Funktion

- z.B. SHA-1, SHA-2, MD5
- Ein kryptographische Hash-Funktion h bildet einen Text auf einen Code fester Länge ab, so dass
  - h(text) = code
  - es unmöglich ist einen anderen Text zu finden mit:
    - h(text‘) = h(text) und text ≠ text‘
- Mögliche Lösung:
  - Verwendung einer symmetrischen Kodierung

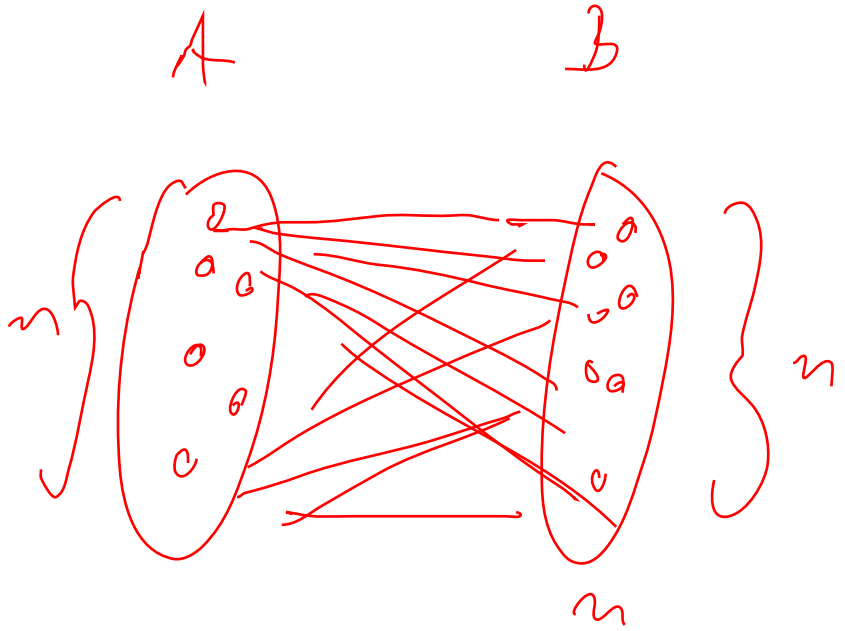
m mod p

UNI FREIBURG


∞
$10^{10} \cdot 10^{3} \cdot 10^{9} \cdot 10^{10} = 10^{32}$
$\sqrt{10^{60}} = 10^{30}$
200 Bit
2^{200}
$2^{200} = \left(2^{10}\right)^{20} = 10^{60}$
10^3



$\frac{1}{365}$



$$\frac{n}{365} + \frac{n}{365} \cdots \frac{n^2}{365}$$

$$n^2 \simeq 365$$

$$19 \simeq n \simeq \sqrt{365}$$

CoNe Freiburg

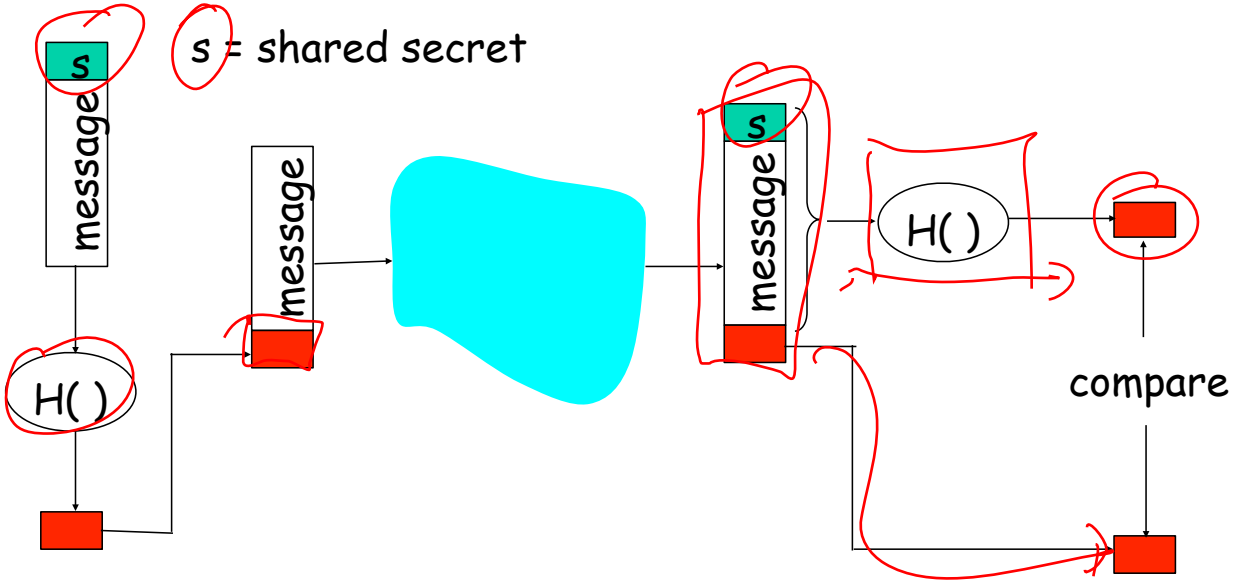
Kryptographische Hash-Algorithmen

- MD5 ist sehr verbreitet (RFC 1321)
  - berechnet 128-bit Nachricht
  - unsicher
- SHA-1 auch gebräuchlich
  - US standard [NIST, FIPS PUB 180-1]
  - 160-bit Message Digest
  - nicht mehr als sicher angesehen
- SHA-2
  - SHA-256/224
  - SHA-512/384
  - bis jetzt (2012) als sicher angesehen
- SHA-3
  - 2011 veröffentlicht

UNI FREIBURG

CoNe Freiburg

Message Authentication Code (MAC)



- Authentifiziert Absender
- Überprüft Nachrichtenintegrität
- Keine Verschlüsselung
- “keyed hash”
- Notation: MDm = H(s || m) ; sende m || MDm

a || b = 0110 || 1010 = 01101010

Computer Networking: A Top Down Approach
Jim Kurose, Keith Ross, Addison-Wesley, April 2009.

CoNe Freiburg

HMAC (Keyed-Hash Message Authentication Code)

- Populärer MAC-Standard
- Sicher gegen Anhängen von Nachrichten

$$HMAC_K(N) = H\Big((K \oplus opad) \,||\, H\big((K \oplus ipad) \,||\, N\big)\Big)$$

  - Nachricht N
  - geheimer Schlüssel K
  - Konstante opad und ipad
- Erhöht Sicherheit gegen angreifbare Hash-Codes
  - wird in TLS und IPsec verwendet

UNI FREIBURG

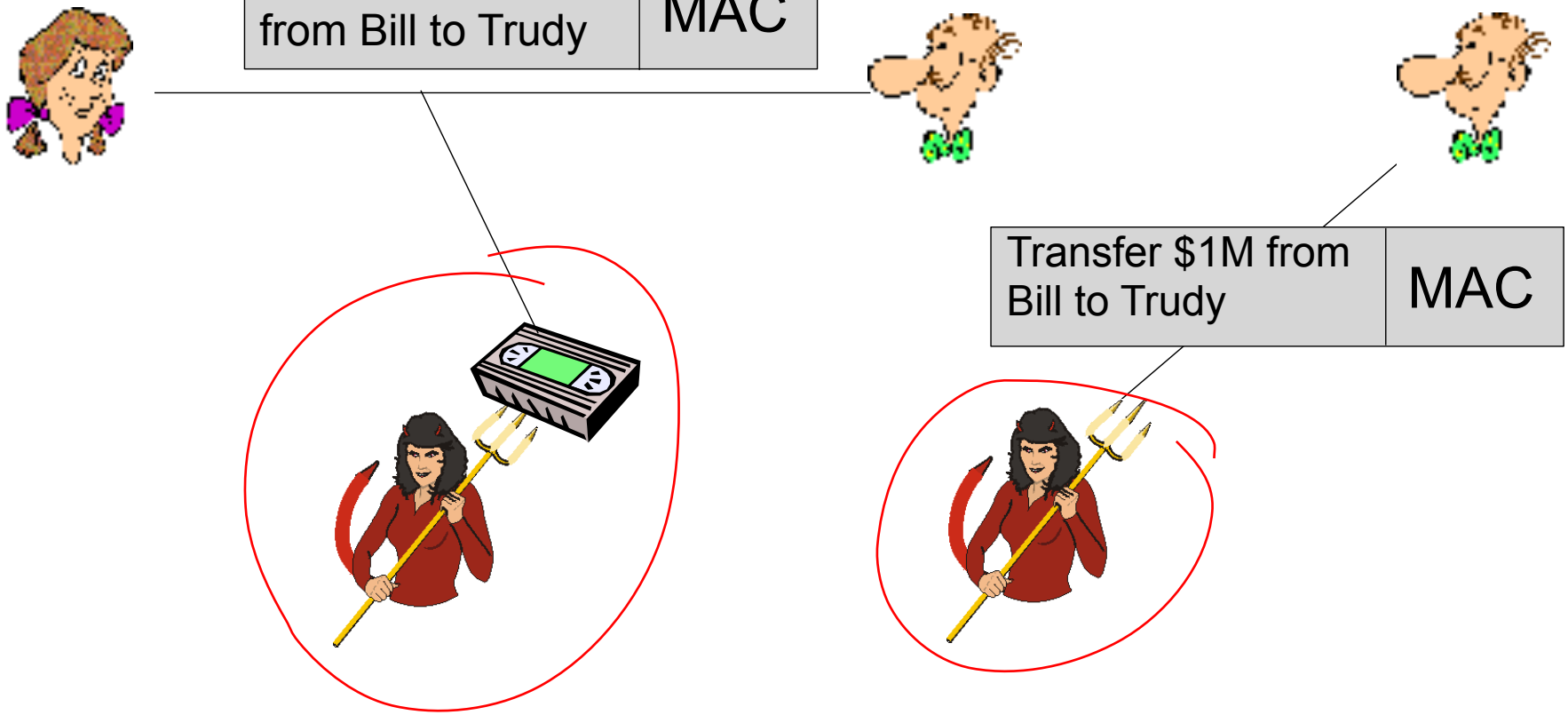
CoNe Freiburg

Authentifizierung der Endpunkte

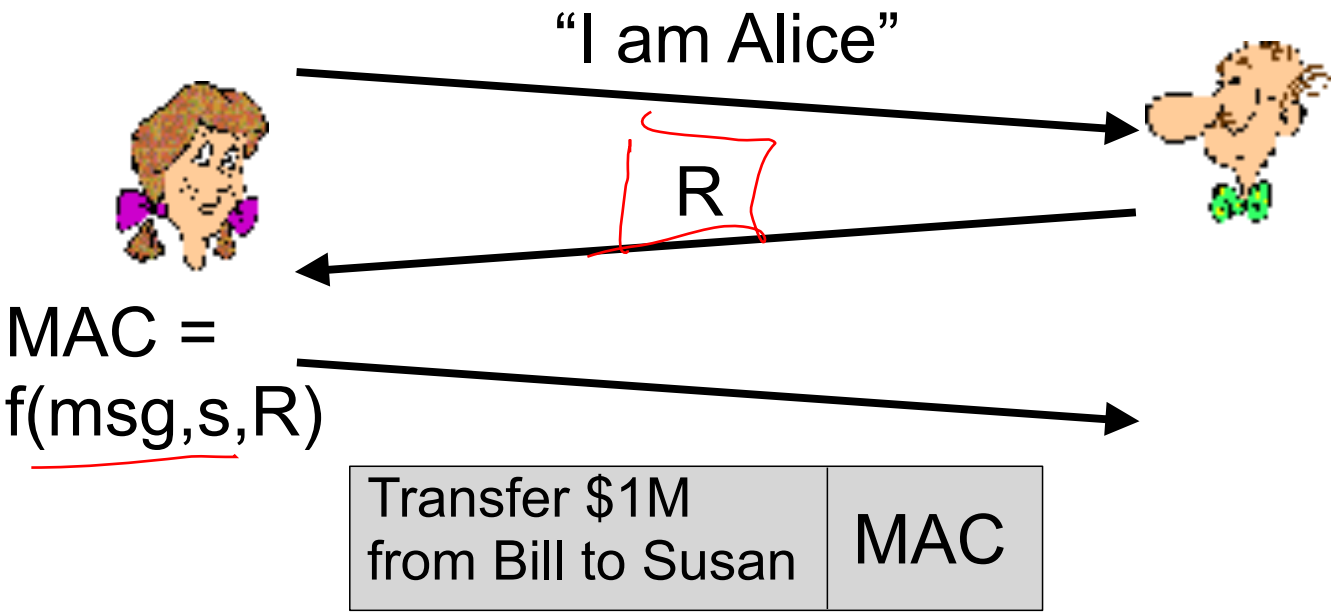
- Versicherung, dass der Kommunikationspartner korrekt ist
- Angenommen Alice und Bob haben ein gemeinsames Geheimnis, dann gibt MAC eine Authentifizeirung der Endpunkte
  - (end-point authentication)
  - Wir wissen, dass Alice die Nachricht erzeugt hat
  - Aber hat sie sie auch abgesendet?

Playback-Attacke

MAC = f(msg,s)



Verteidigung gegen die Playback-Attacke: nonce (use only once)



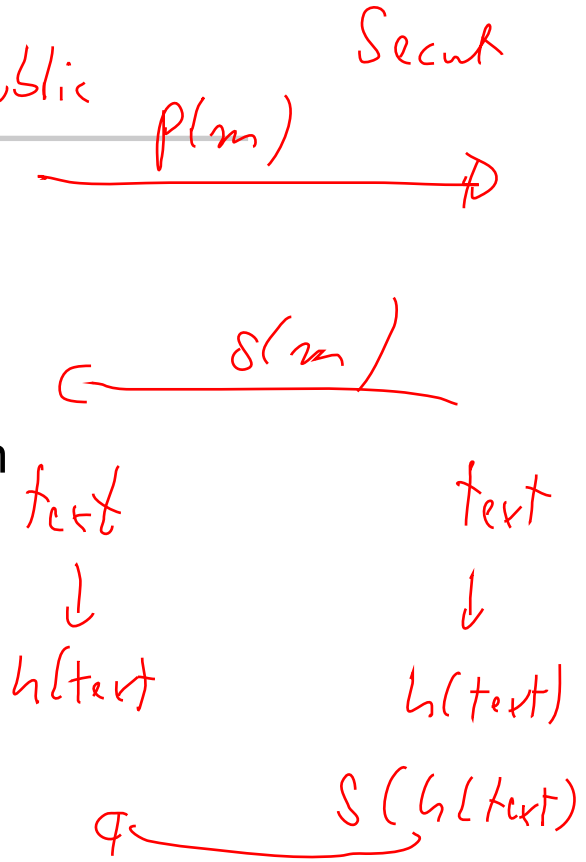
CoNe Freiburg

Digitale Unterschriften

- Kryptographischer Algorithmus analog zu handgeschriebenen Unterschriften
  - nur sicherer
- Absender (Bob) unterschreibt digital das Dokument
  - bestätigt seine Urheberschaft
- Ziel ist ähnlich wie MAC
  - aber mit Hilfe von Public-Key-Kryptographie
  - verifizierbar, nicht fälschbar:
    - Empfänger (Alice) kann anderen beweisen, dass Bob und sonst niemand das Dokument unterschrieben hat

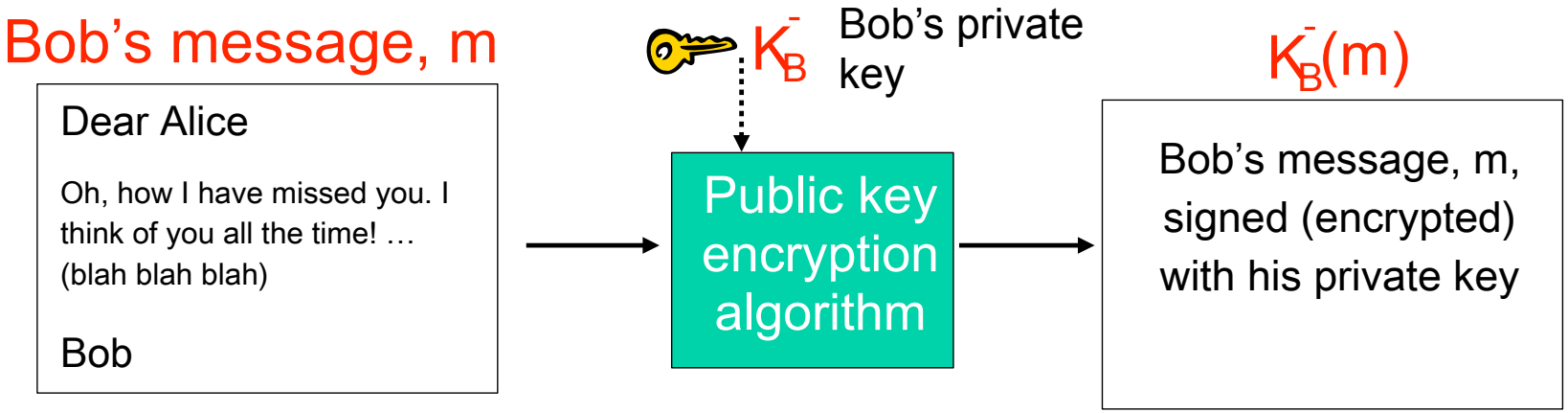
UNI FREIBURG

Elektronische Unterschriften



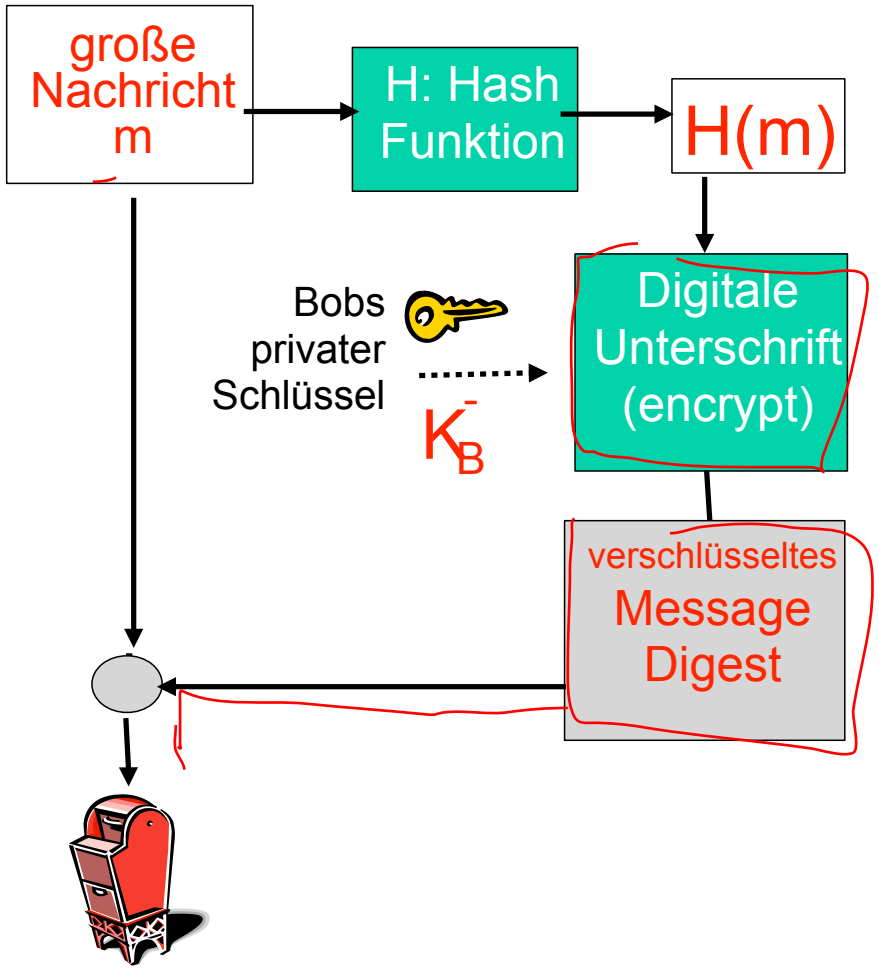
- Digitale Signaturen
  - Unterzeichner besitzt einen geheimen Schlüssel
  - Dokument wird mit geheimen Schlüssel unterschrieben
  - und kann mit einem öffentlichen Schlüssel verifiziert werden
  - Öffentlicher Schlüssel ist allen bekannt
- Beispiel eines Signaturschemas
  - m: Nachricht
  - Unterzeichner
    - berechnet h(text) mit kryptographischer Hashfunktion
    - und veröffentlicht m und signatur = g(privat, h(text)), für die Entschlüsselungsfunktion g
  - Kontrolleur
    - berechnet h(text)
    - und überprüft f(offen, signatur) = h(text), für die asymmetrische Verschlüsselungsfunktion g

Digitale Unterschrift

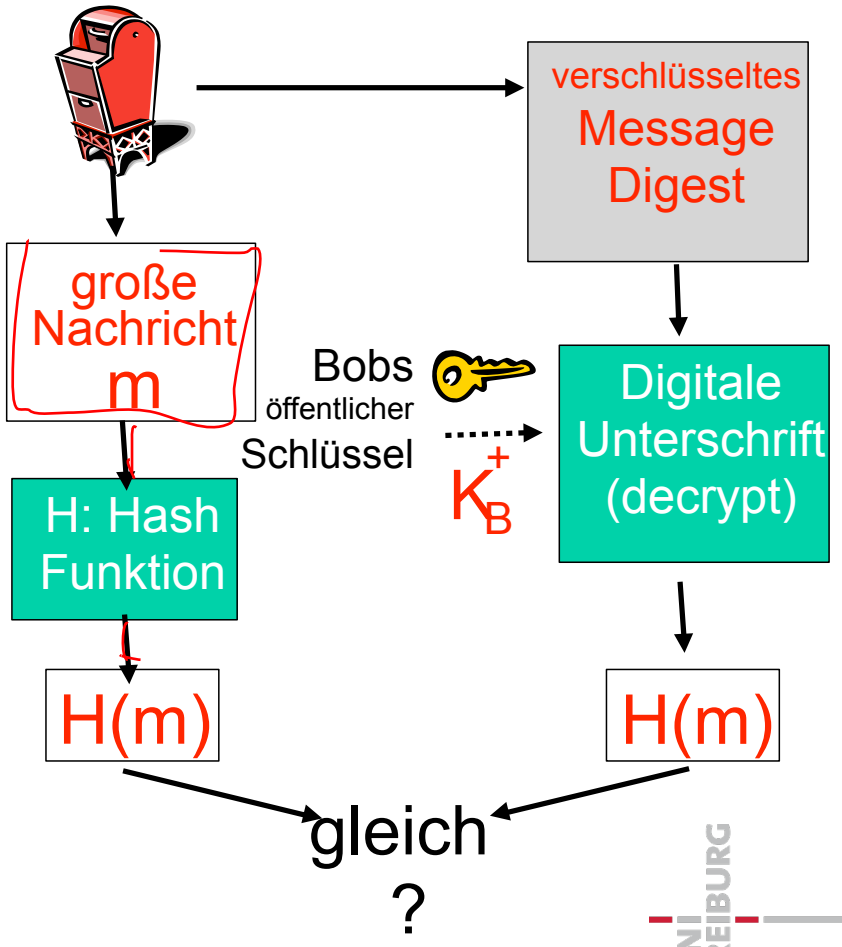


Digitale signature = signiertes Message Digest

Bob sendet eine digital unterschriebene Nachricht



Alice überprüft die Unterschrift und die Korrektheit der Nachricht



CoNe Freiburg

Digitale Unterschrift

- Angenommen Alice erhält
  - die Nachricht m
  - mit digitaler Unterschrift $K^-_B(m)$
- Alice überprüft m
  - mit den öffentlichen Schlüssel von Bob
  - Ist $K^+_B(K^-_B(m)) = m$?
- Falls $K^+_B(K^-_B(m)) = m$
  - dann hat jemand Bobs geheimen Schlüssel
- Alice verifiziert daher, dass
  - Bob hat m unterschrieben
  - Niemand anders hat m unterschrieben
  - Bob hat m und nicht m‘≠m unterschrieben
- Unleugbarkeit
  - Alice kann mit m und der Unterschrift vor Gericht gehen und und beweisen, dass Bob m unterschrieben hat

UNI FREIBURG

CoNe Freiburg

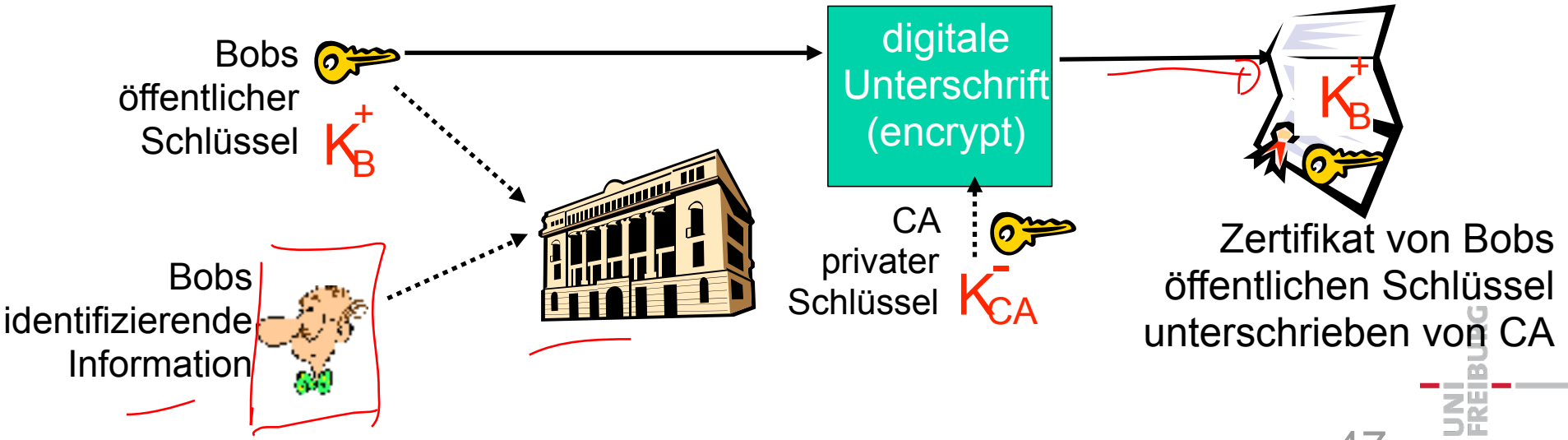
Public-key Zertifizierung

- Motivation: Trudy spielt Bob einen Pizza-Streich
- Trudy bestellt per e-mail order:
  - „Liebe Pizzeria, schick mir bitte vier Pepperoni-Pizza. vielen Dank Bob“
- Trudy unterschreibt mit ihrem privaten Schlüssel
- Trudy sendet die Bestellung zur Pizzeria
- Trudy sendet der Pizzeria ihren öffentlichen Schlüssel
  - behauptet aber er gehöre Bob
- Die Pizzeria überprüft die Unterschrift
  - Bob mag gar keine Pepperoni

UNI FREIBURG

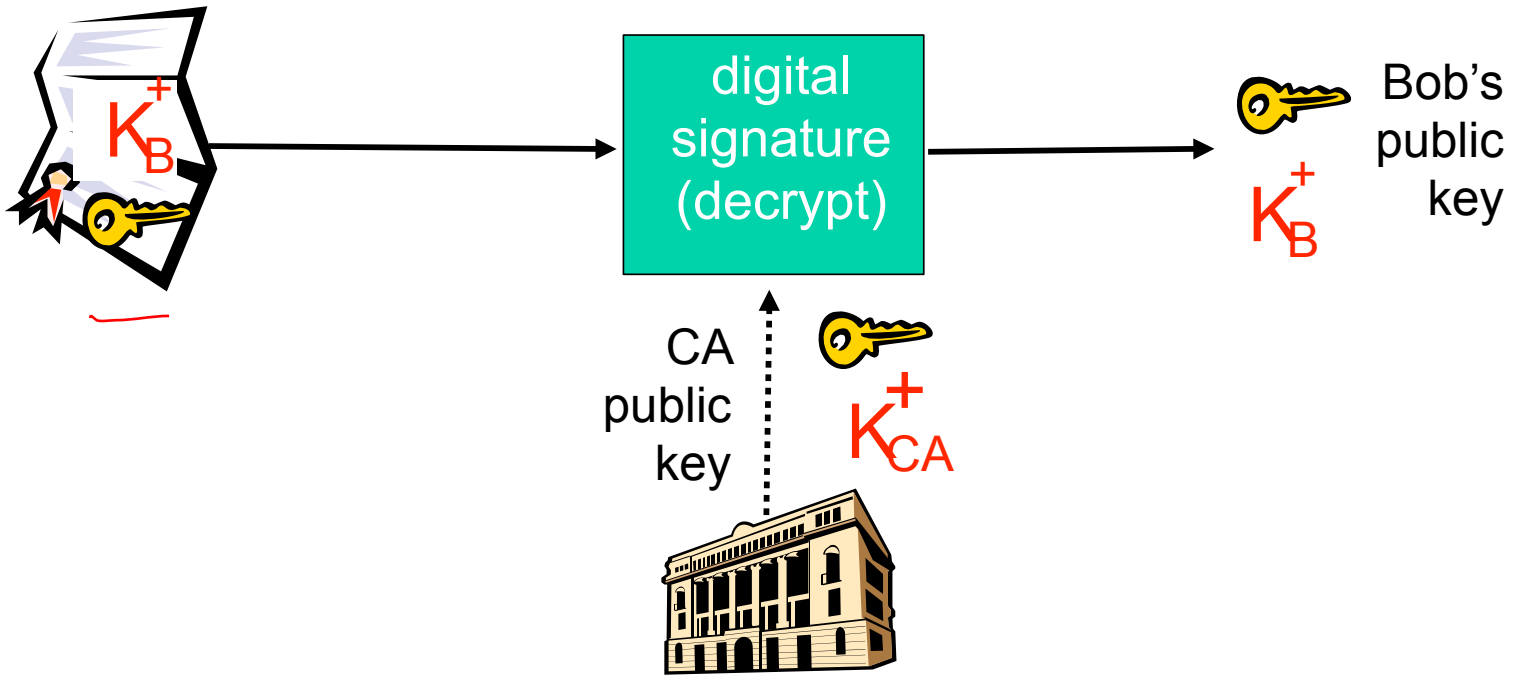
Zertifizierungsstelle
Certification Authorities (CA)

- Zertifizierungsstelle (Certification authority – CA): verknüpft öffentlichen Schlüssel mit der Entität (Person, Service, Router) E
- E registriert seinen öffentlichen Schlüssel mit CA
  - E „beweist seine Identität“ der Zertifizierungsstelle
  - CA erzeugt eine Zertifizierungsverknüpfung von E mit seinem öffentlichen Schlüssel
  - Zertifikat mit E‘s öffentlichen Schlüssel wird von der CA digital unterschriben:
    - „Das ist der öffentliche Schlüssel von E“

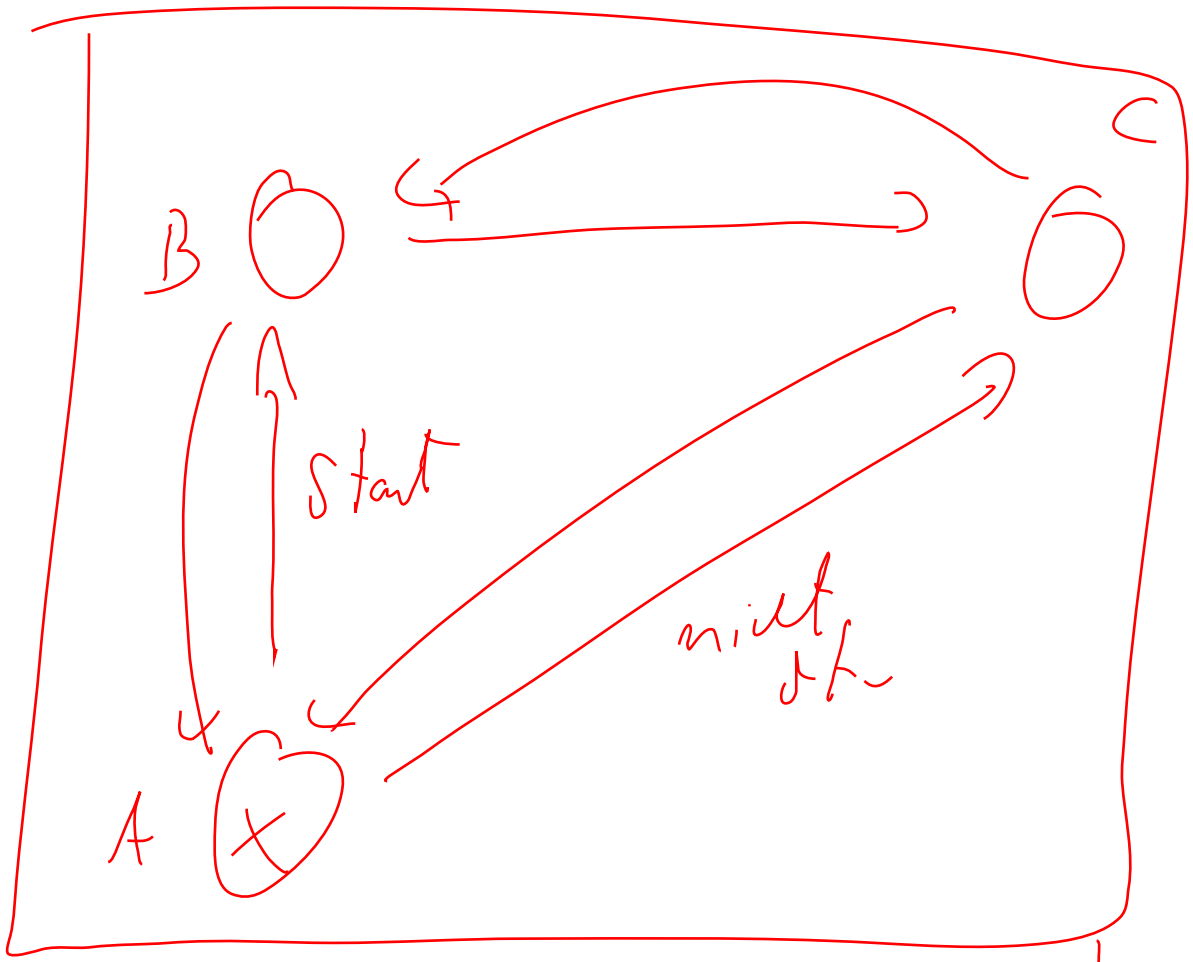


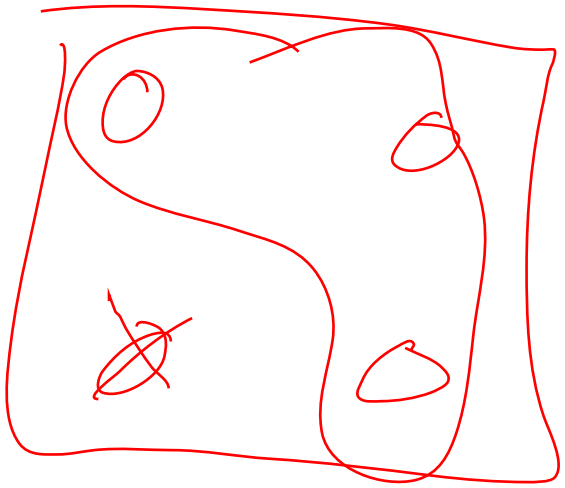
Zertifizierungsstelle
Certification Authorities (CA)

- Wenn Alice Bobs öffentlichen Schlüssel möchte
  - erhält Bobs Zertifikat
  - wendet CA's öffentlichen Schlüssel auf Bobs Zertifikat an
    - Alice erhält Bobs öffentlichen Schlüssel



Byzantinische Generäle





(A → B, „Start")

CoNe Freiburg

Zertifikate

- Hauptstandard X.509 (RFC 2459)
- Zertifikat enthält
  - Name des Ausstellers (Issuer name)
  - Name der Entität, Adresse, Domain-Name, etc.
  - Öffentlicher Schlüssel der Entität
  - Digitale Unterschrift (unterschrieben mit dem geheimen Schlüssel des Ausstellers)
- Public-Key Infrastruktur (PKI)
  - Zertifikate und Zertifizierungsstellen

UNI FREIBURG

CoNe Freiburg

SSL: Secure Sockets Layer

- Weit verbreitetes Sicherheitsprotokoll
  - Unterstützt durch alle Browser und Web-Server
  - https
  - Jährlich Transaktionen im Wert von Zigmilliarden Euro über SSL
- 1993 entworfen von Netscape
- Aktueller Name
  - TLS: transport layer security, RFC 2246
- Gewährleistet
  - Vertraulichkeit (Confidentiality)
  - Nachrichtenintegrität (Integrity)
  - Authentifizierung

UNI FREIBURG

CoNe Freiburg

SSL: Secure Sockets Layer

- Ursprüngliche Motivation
  - Web E-Commerce Transaktionen
  - Verschlüsselung (Credit-Karte)
- Web-server Authentifizierung
  - Optional Client Authentifizierung
- Kleinstmöglicher Aufwand für Einsteiger
- In allen TCP Anwendungen verfügbar
  - Secure socket interface

UNI FREIBURG

CoNe Freiburg

Zumeist: Symmetrische Codes

- DES – Data Encryption Standard: Block
- 3DES – Triple strength: Block
- RC2 – Rivest Cipher 2: Block
- RC4 – Rivest Cipher 4: Stream

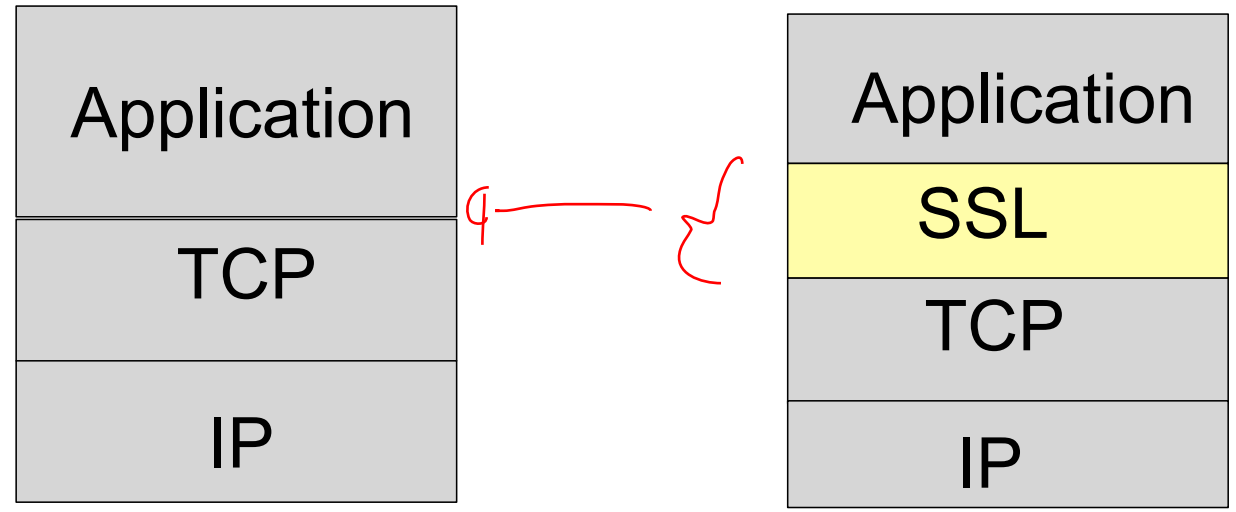
- Auch Public-Key-Verschlüsselung
  - RSA

CoNe Freiburg

SSL Cipher Suite

- Cipher Suite
  - Public-key Algorithmus
  - Symmetrische Verschlüsselungsalgorithmus
  - MAC Algorithmus
- SSL unterstützt mehrere Kodierungsverfahren
- Verbindungsvereinbarung (Negotiation)
  - Client und Server einigen sich auf ein Kodierungsverfahren
- Client bietet eine Auswahl an
  - Server wählt davon eines

SSL and TCP/IP



- SSL stellt eine Programm-Interface für Anwendungen zur Verfügung
- C and Java SSL Bibliotheken/Klassen verfügbar

CoNe Freiburg

SSL: Handshake (1)

- Ziel
  - Server Authentifizierung
  - Verbindungsvereinbarung:
    - Einigung auf gemeinsames kryptographische Verfahren
  - Schlüsselaustausch
  - Client Authentifizierung (optional)

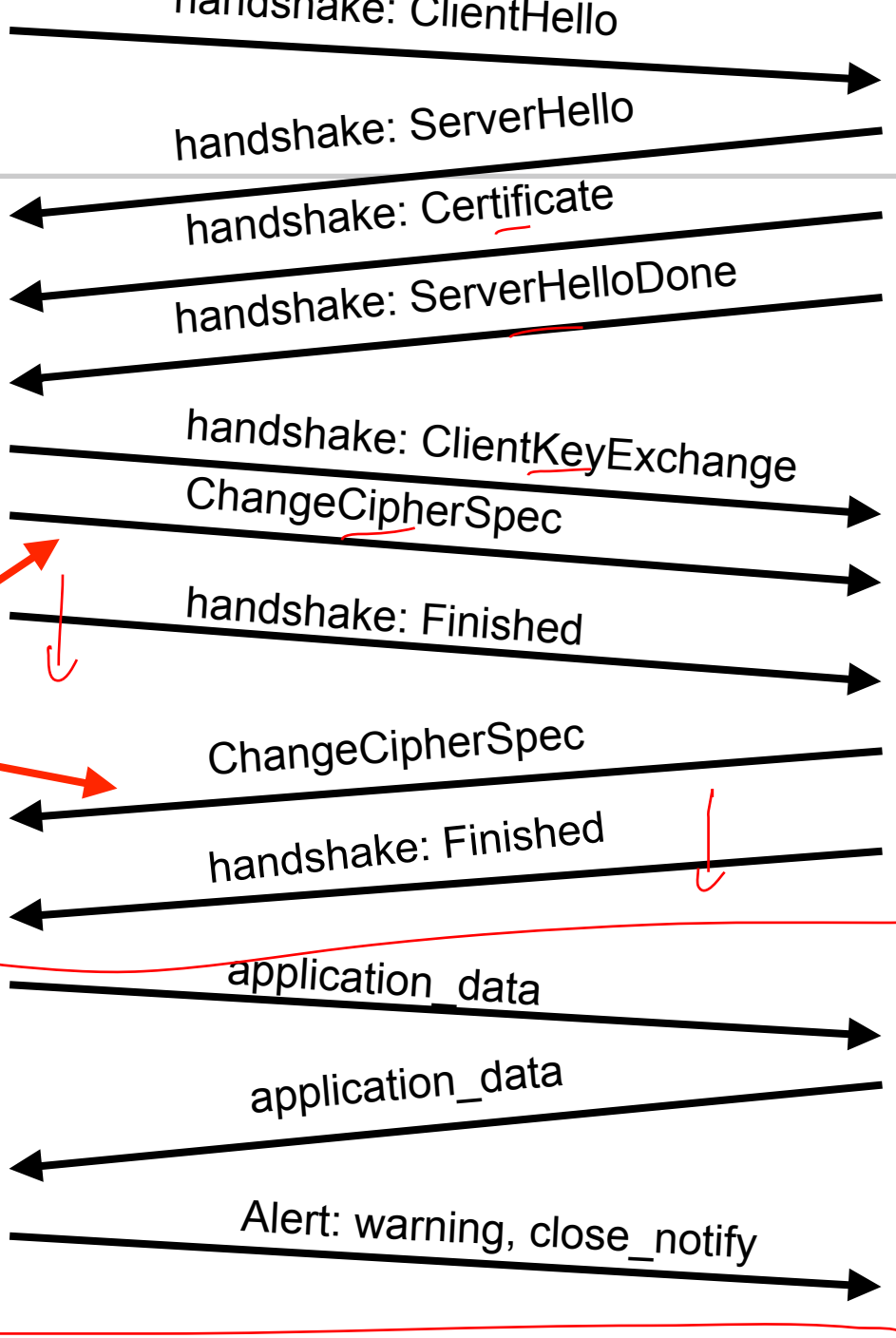
CoNe Freiburg

SSL: Handshake (2)

- Client sendet
  - Liste unterstützter Krypto-Algorithmen
  - Client nonce (salt)
- Server
  - wählt Algorithmen von der Liste
  - sendet zurück: Wahl + Zertifikat + Server Nonce
- Client
  - verifiziert Zertifikat
  - extrahiert Servers öffentlichen Schlüssel
  - erzeugt pre_master_secret verschlüsselt mit Servers öffentlichen Schlüssel
  - sendet pre_master_secret zum Server
- Client und Server
  - berechnen unabhängig die Verschlüsselungs- und MAC-Schlüssel aus pre_master_secret und Nonces
- Client sendet ein MAC von allen Handshake-Nachrichten
- Server sendet ein MAC von allen Handshake-Nachrichten

UNI FREIBURG

SSL Verbindung



Ab hier ist alles verschlüselt

TCP Fin folgt

CoNe Freiburg

Schlüsselherstellung

- Client Nonce, Server Nonce und pre-master secret werden in Pseudozufallsgenerator gegeben
  - Ausgabe: Master Secret
- Master Secret und neue Nonces werden in anderen Pseudozufallsgenerator mit Ausgabe: “key block”
- Key block:
  - Client MAC key
  - Server MAC key
  - Client encryption key
  - Server encryption key
  - Client initialization vector (IV)
  - Server initialization vector (IV)

UNI FREIBURG