Netzwerktechnologien
3 VO

Univ.-Prof. Dr. Helmut Hlavacs
helmut.hlavacs@univie.ac.at

Dr. Ivan Gojmerac
gojmerac@ftw.at

Bachelorstudium Medieninformatik
SS 2012

Kapitel 8 - Netzwerksicherheit

8.1 Sicherheitsanforderungen in Netzen

Vertraulichkeit: Nur der Sender und der korrekte Adressat sollen den Inhalt der Nachricht lesen können.

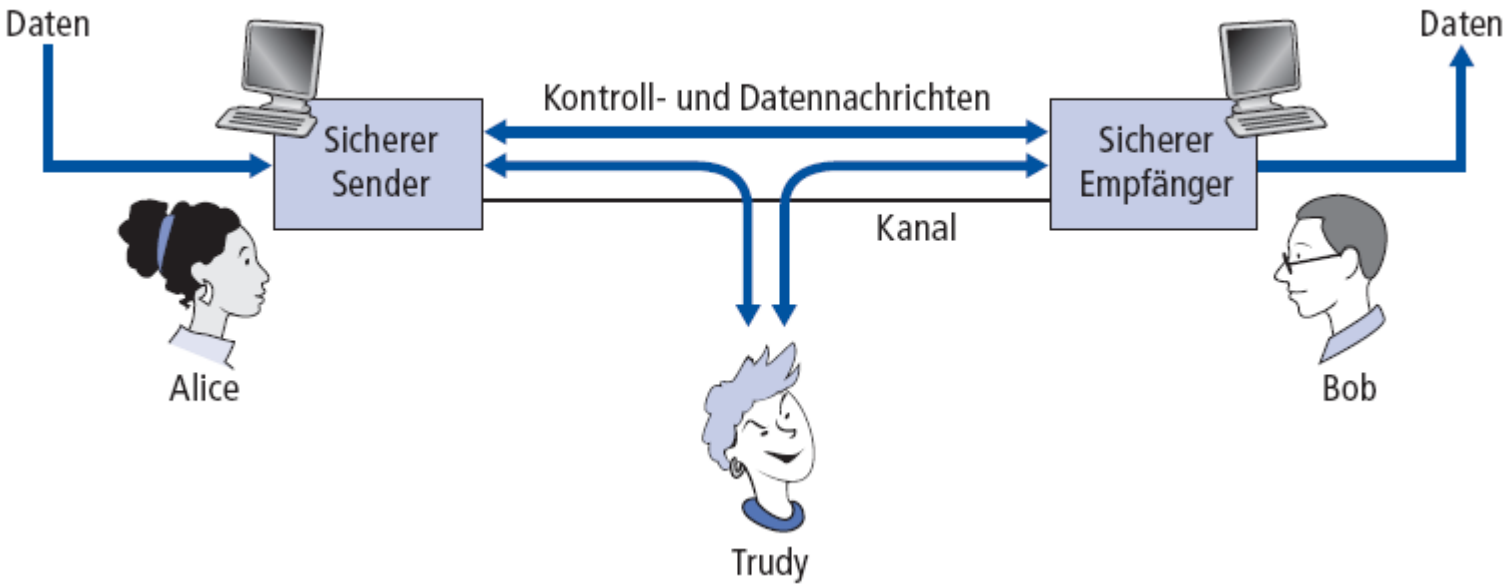
Authentifizierung: Sender und Empfänger wollen gegenseitig ihre Identität sicherstellen.

Nachrichtenintegrität: Sender und Empfänger wollen sicherstellen, dass die Nachricht nicht unbemerkt verändert wurde (während der Übertragung oder danach).

Zugriff und Verfügbarkeit: Dienste müssen für Benutzer zugreifbar und verfügbar sein.

8.1 Freunde und Feinde – Alice, Bob, Trudy

- Alice, Bob und Trudy sind „bekannte Gestalten“ in der Welt der Netzwerksicherheit
  - Alice und Bob möchten “sicher” kommunizieren
  - Trudy (ein Eindringling) kann Nachrichten abfangen, löschen, einfügen



8.1 Wer könnten Bob und Alice sonst noch sein?

- Webbrowser/-server für elektronische Transaktionen (z.B. Online-Einkäufe)
- Client und Server für Online-Banking
- DNS-Server
- Router, die Routingtabellen-Updates austauschen
- Weitere Beispiele?

8.1 Typen von Angriffen

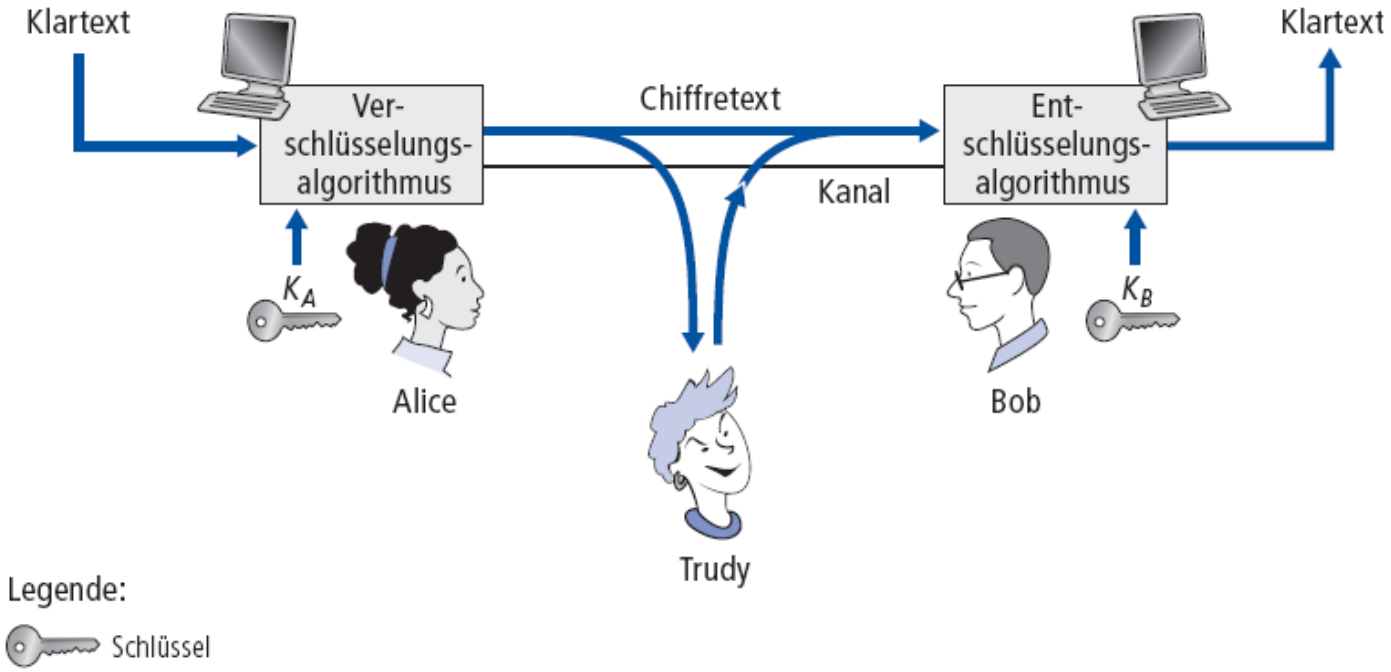
Q: Was können Angreifer tun?

A: Eine ganze Menge!

- **Lauschen:** Nachrichten mitlesen
- Aktiv Nachrichten in die Verbindung **einspeisen**
- **Fremde Identitäten annehmen und Quelladressen** (oder andere Felder im Paket) **fälschen**
- **Denial of Service:** verhindern, dass andere einen Dienst nutzen können (z.B. durch Überlasten von Ressourcen)
- Usw.

Kapitel 8 - Netzwerksicherheit

8.2 Terminologie der Kryptographie



Symmetrische Kryptographie: Sender- und Empfängerschlüssel sind *identisch*

Public-Key-Kryptographie: Schlüssel zur Verschlüsselung ist *öffentlich bekannt*, zur Entschlüsselung *geheim*

8.2 Kryptographie mit symmetrischen Schlüsseln

Ersetzungschiffre: eine Sache durch eine andere ersetzen

– Monoalphabetische Chiffre: einen Buchstaben durch einen anderen ersetzen

```
Klartext:   abcdefghijklmnopqrstuvwxyz
            ↓                       ↓
Chiffretext: mnbvcxzasdfghjklpoiuytrewq
```

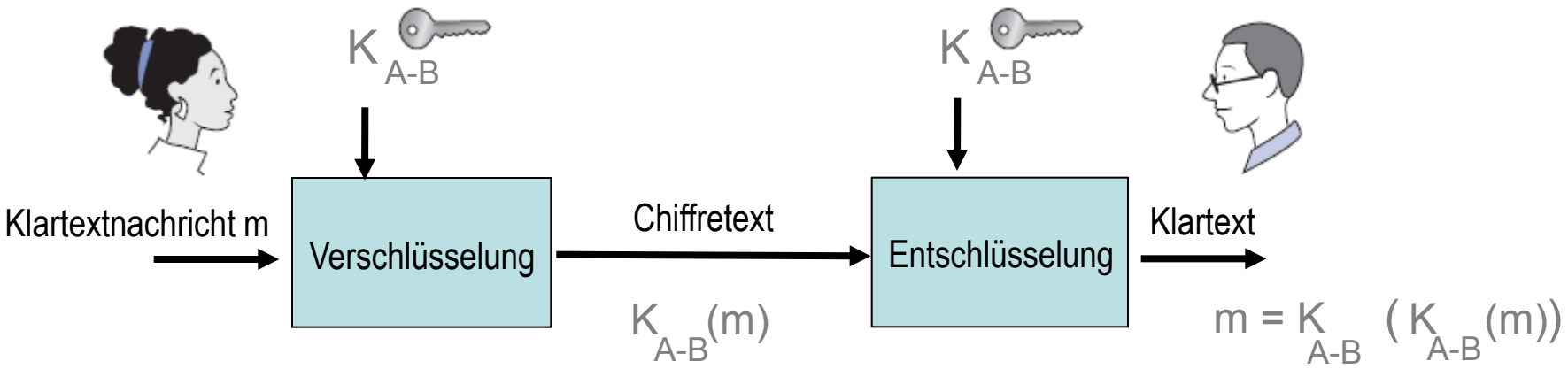
z.B.:

```
Klartext: bob, ich liebe dich. alice
Chiffretext: nkn, sba pscbc vsba. mgsbc
```

8.2 Kryptographie mit symmetrischen Schlüsseln

Kryptographie mit symmetrischen Schlüsseln: Bob and Alice kennen denselben (symmetrischen) Schlüssel **K**

- Der Schlüssel könnte zum Beispiel das Ersetzungsmuster der monoalphabetischen Chiffre sein



8.2 Symmetrische Kryptographie: DES Algorithmus

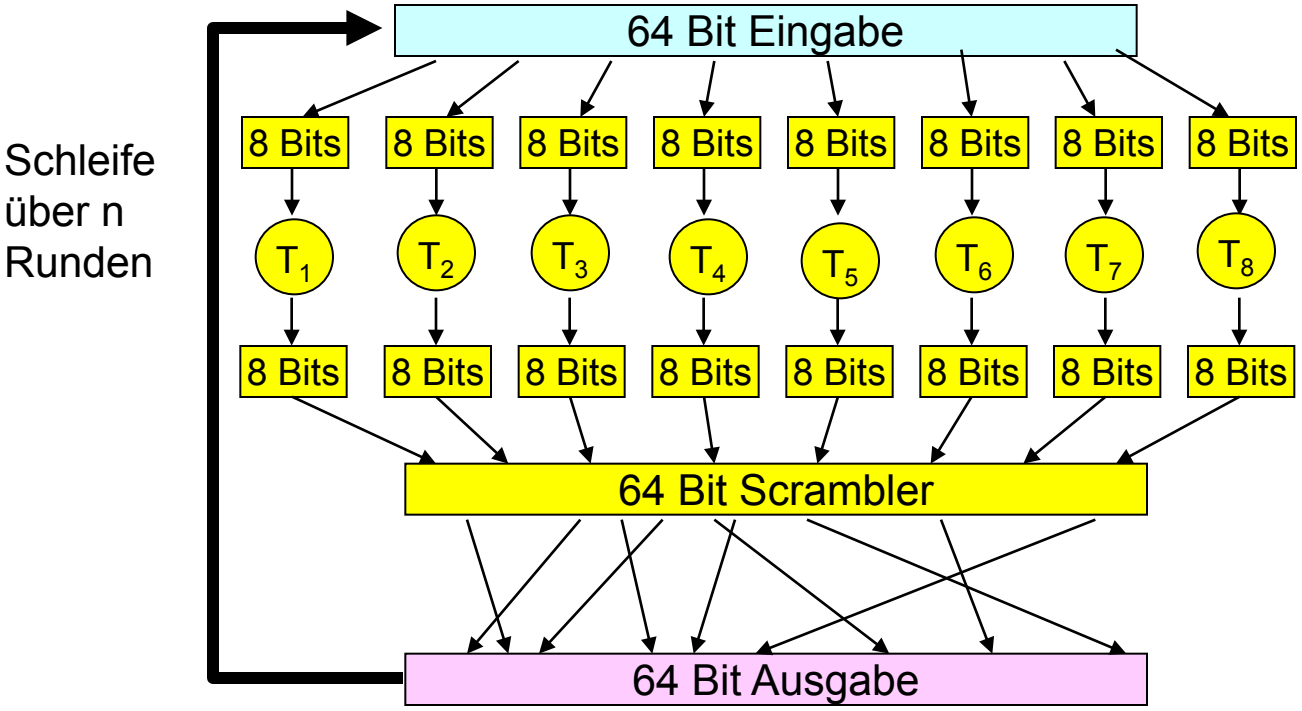
DES: Data Encryption Standard

- US-Verschlüsselungsstandard [NIST 1993]
- Symmetrische 56-Bit-Schlüssel, 64 Bit lange Klartext-Eingaben
- Problem: DES ist nicht ausreichend sicher → [RFC 4772]

8.2 AES: Advanced Encryption Standard

- Neuer symmetrischer NIST-Standard (Nov. 2001), der DES ersetzen soll.
- Verarbeitet Daten in 128-Bit-Blöcken
- 128, 192, oder 256 Bit lange Schlüssel
- Wenn Brute-Force-Entschlüsselung (alle Schlüssel ausprobieren) für DES eine Sekunde dauert, braucht sie für AES-128 149 Billionen Jahre

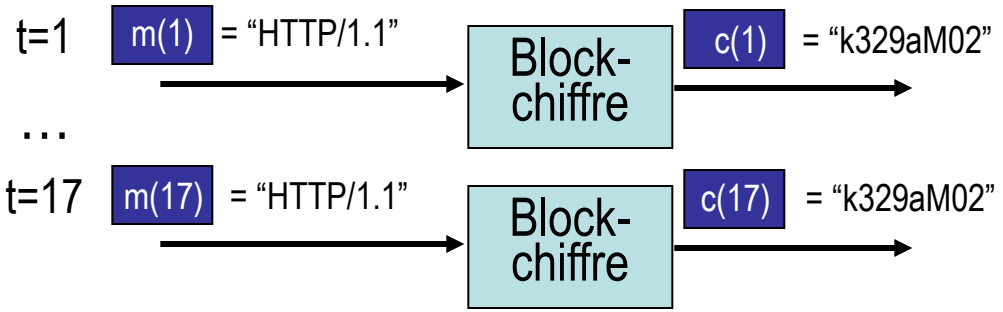
8.2 Blockchiffre



- Ein Durchlauf: ein Eingabebit beeinflusst acht Ausgabebits
- Mehrere Durchläufe: jedes Eingabebit hat Auswirkungen auf alle Ausgabebits
- Blockchiffren: DES, 3DES, AES

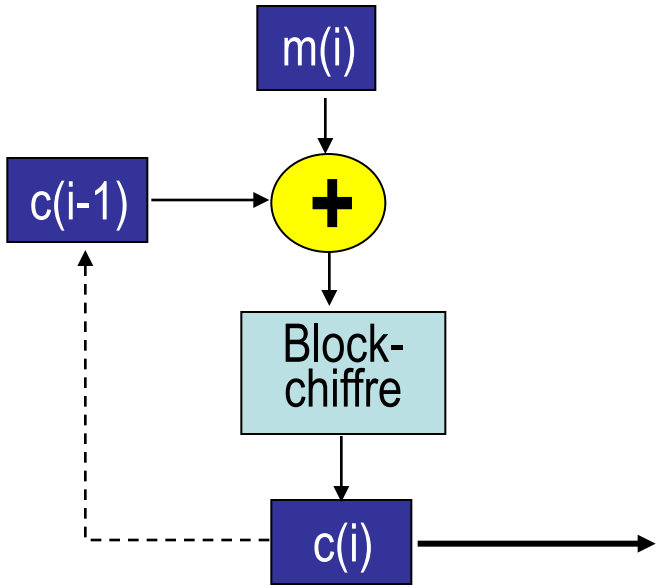
Cipher Block Chaining

Wenn ein Eingabeblock sich wiederholt, wird dieselbe Chiffre eine identische Ausgabe erzeugen.



Abhilfe:

- **Cipher Block Chaining:** XOR des i-ten Eingabeblocks m(i) mit dem vorangegangenen verschlüsselten Block c(i-1)
  - **Initialisierungsvektor** c(0) wird im Klartext an den Empfänger übertragen



8.2 Public Key Kryptographie

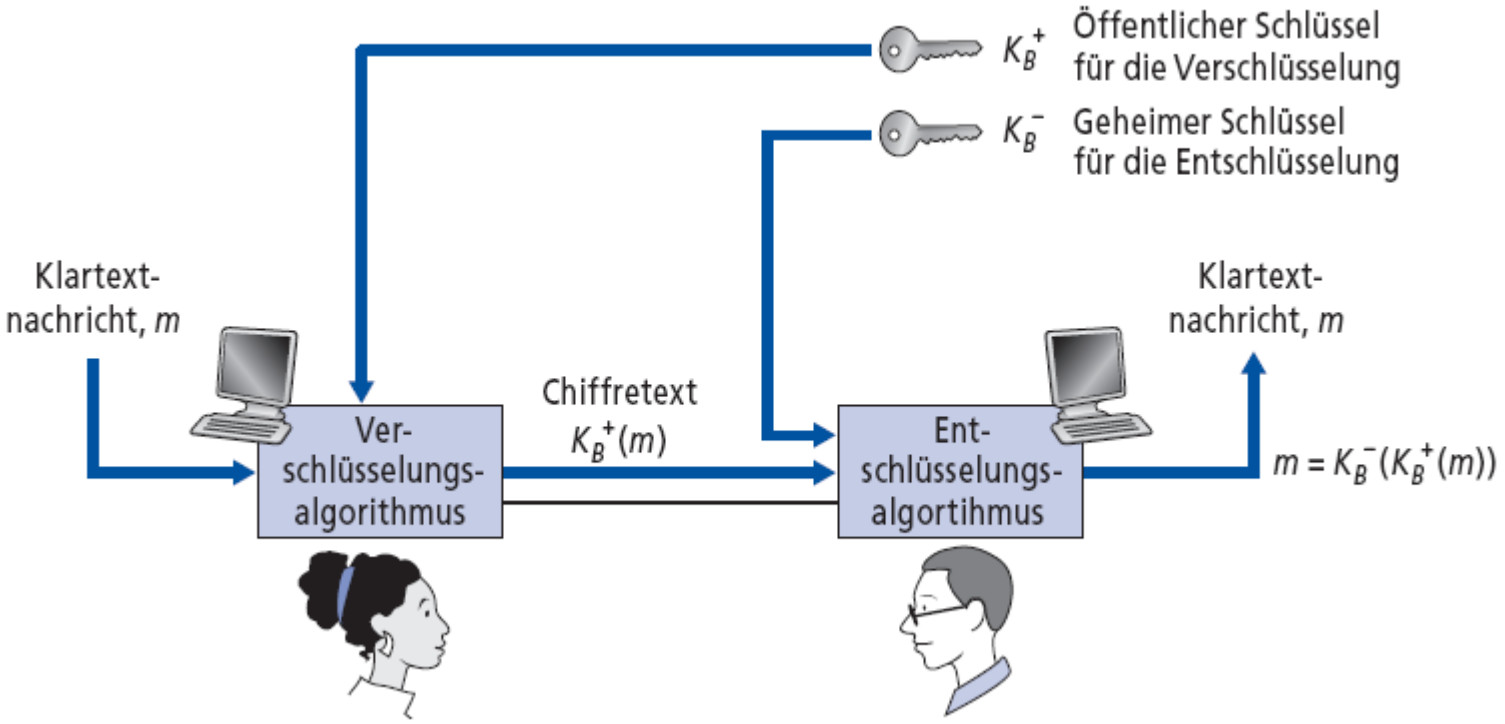
Symmetrische Kryptographie:

- erfordert, dass Sender und Empfänger über einen gemeinsamen Schlüssel verfügen
- Problem der symmetrischen Kryptographie: Wie kann man sich überhaupt auf einen Schlüssel einigen (vor allem dann, wenn man sich noch nie “getroffen” hat)?

Public-Key-Kryptographie

- radikal anderer Ansatz [RSA78]
- Sender, Empfänger kennen **keinen gemeinsamen** geheimen Schlüssel
- **öffentlicher** Verschlüsselungsschlüssel, den **alle** kennen
- **geheimen** Entschlüsselungsschlüssel kennt nur der Empfänger

8.2 Public Key Kryptographie



8.2 Public Key Algorithmen

Anforderungen:

1. benötigt $K_B^+( )$ und $K_B^-( )$, für die

$$K_B^-(K_B^+(m)) = m$$

2. gegeben den öffentl. Schlüssel K_B^+, soll es nicht möglich sein, den privaten Schlüssel K_B^- zu errechnen

RSA: Algorithmus von Rivest, Shamir, Adleman

8.2 RSA - Schlüsselgenerierung

1. Wähle zwei große Primzahlen *p, q*.
   (Wobei das Produkt von *p* und *q* z.B. 1024 Bit lang ist.)
2. Berechne **n** = *pq*, *z* = *(p-1)(q-1)*.

3. Wähle ein **e** *(*mit *e<n),* das keine Primfaktoren mit z gemeinsam hat. (*e, z* sind "relative Primzahlen").
4. Wähle **d**, so dass *ed-1* durch z ohne Rest teilbar ist
   (in anderen Worten: *ed* mod *z* = *1*).
5. **Öffentlicher** Schlüssel: Zahlenpaar *(***n, e***)*. **Privater** Schlüssel: Zahlenpaar *(***n, d***)*.

K_B^+ K_B^-

8.2 RSA – Ver- und Entschlüsselung

0. Gegeben: (n,e) und (n,d), berechnet wie oben

1. Um ein Bitmuster m zu verschlüsseln, berechne

   $c = m^e \bmod n$ (Rest beim Teilen von m^e durch n)

2. Zum Entschlüsseln des empfangenen Wertes c berechne

   $m = c^d \bmod n$ (Rest beim Teilen von c^d durch n)

$$m = \underbrace{(m^e \bmod n)}_{c}{}^d \bmod n$$

8.2 Beispiel zu RSA

Bob wählt *p=5, q=7*. Dann gilt: *n=35, z=(p-1)(q-1)=24*.
e=5 (*e* und *z* sind relative Primzahlen)
d=29 (*ed-1* ist ohne Rest teilbar durch z)

verschlüsseln:	Zeichen	m	m^e	$c = m^e \bmod n$
	I	12	1524832	17

entschlüsseln:	c	c^d	$m = c^d \bmod n$	Zeichen
	17	481968572106750915091411825223071697	12	I

8.2 RSA

Warum ist $m = (m^e \bmod n)^d \bmod n$

Resultat aus der Zahlentheorie: Wenn p, q Primzahlen sind und $n = pq$, dann gilt:

$$x^y \bmod n = x^{y \bmod (p-1)(q-1)} \bmod n$$

$$(m^e \bmod n)^d \bmod n = m^{ed} \bmod n$$

$$= m^{ed \bmod (p-1)(q-1)} \bmod n$$

$$= m^1 \bmod n$$

(weil wir ed so **gewählt** haben, dass es durch $(p-1)(q-1)$ mit Rest 1 teilbar ist)

$$= m$$

8.2 RSA

Eine wichtige Eigenschaft von RSA:

$$\underbrace{K_B^-(K_B^+(m))}_{\text{Erst öffentlicher Schlüssel angewendet, dann privater Schlüssel}} = m = \underbrace{K_B^+(K_B^-(m))}_{\text{Erst privater Schlüssel angewendet, dann öffentlicher Schlüssel}}$$

→ Identische Ergebnisse!