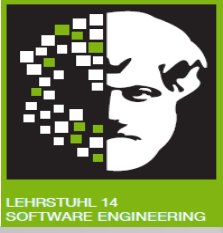


Vorlesung

Methodische Grundlagen des Software-Engineering

im Sommersemester 2014

Prof. Dr. Jan Jürjens

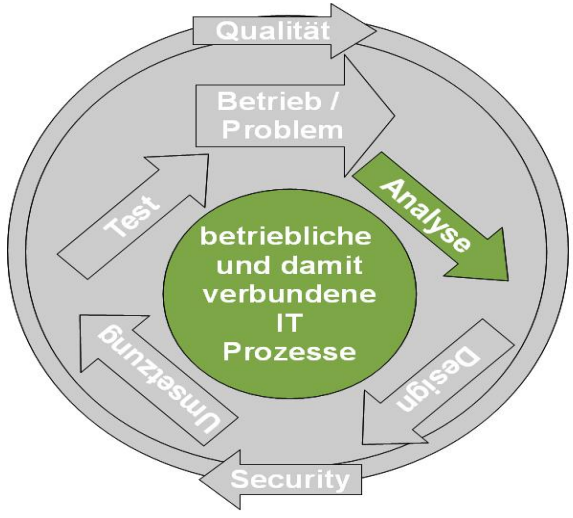
TU Dortmund, Fakultät Informatik, Lehrstuhl XIV

3.1 Hintergrund IT-Sicherheit

v. 25.06.2014

Einordnung
3.1 Hintergrund IT-Sicherheit

- Geschäftsprozessmodellierung
- Process-Mining
- Modellbasierte Entwicklung sicherer Software
  - Einführung: Software Security
  - Hintergrund IT-Sicherheit
  - Wiederholung: Modellbasierte Software Entwicklung
  - Modellbasierte Sicherheit mit UML
  - Sichere Architekturen
  - Kryptographische Protokolle
  - Protokollanalyse
  - Biometrische Authentisierung
  - Biometrische Authentisierung: Analyse
  - Elektronische Geldbörsen
  - Clouds
  - Elektronische Signatur
  - Bankarchitektur



Literatur:
[Jür05] Jan Jürjens: **Secure systems development with UML**, Springer-Verlag 2005.
Unibibliothek (e-Book):
http://www.ub.tu-dortmund.de/katalog/titel/1361890
Papier-Version:
http://www.ub.tu-dortmund.de/katalog/titel/1091324

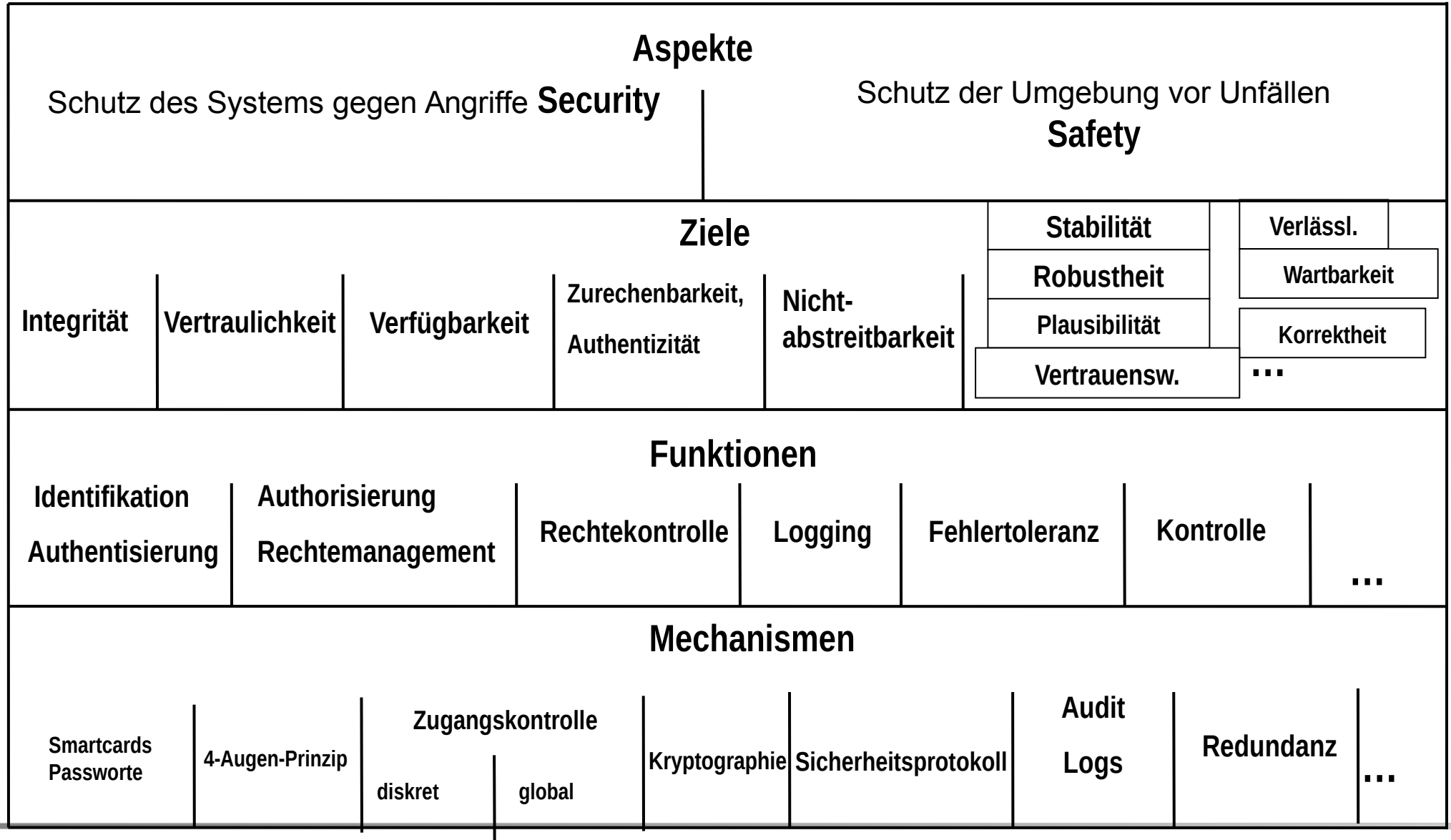
- **Letzter Abschnitt:** Modellbasierte Sicherheit
  - Probleme sicherer Systeme
  - Modellbasierte Entwicklung
- **Dieser Abschnitt:** IT-Sicherheit
  - Sicherheitsanforderungen
  - Angriffe
  - Sicherheitsmechanismen

Überblick
Hintergrund IT-Sicherheit

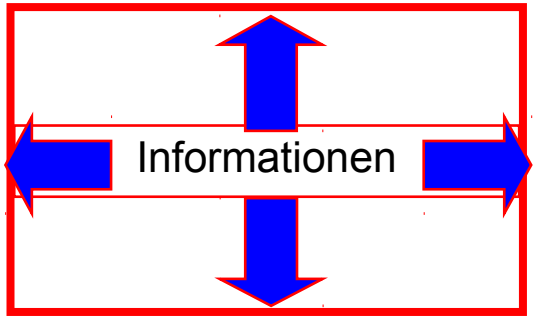
- Sicherheits-Anforderungen und Risiken
- Kryptographie

Sicherheitsanforderungen

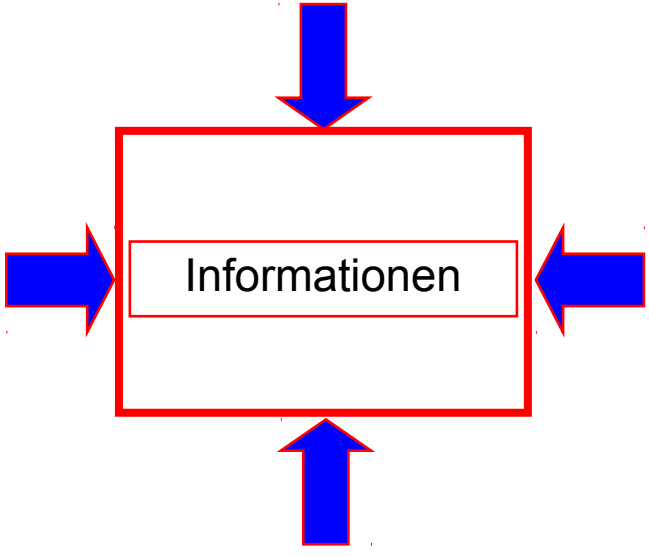
[Weck 2003]



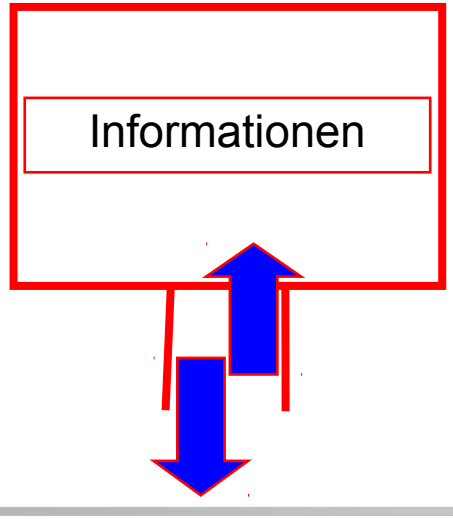
Vertraulichkeit



Integrität



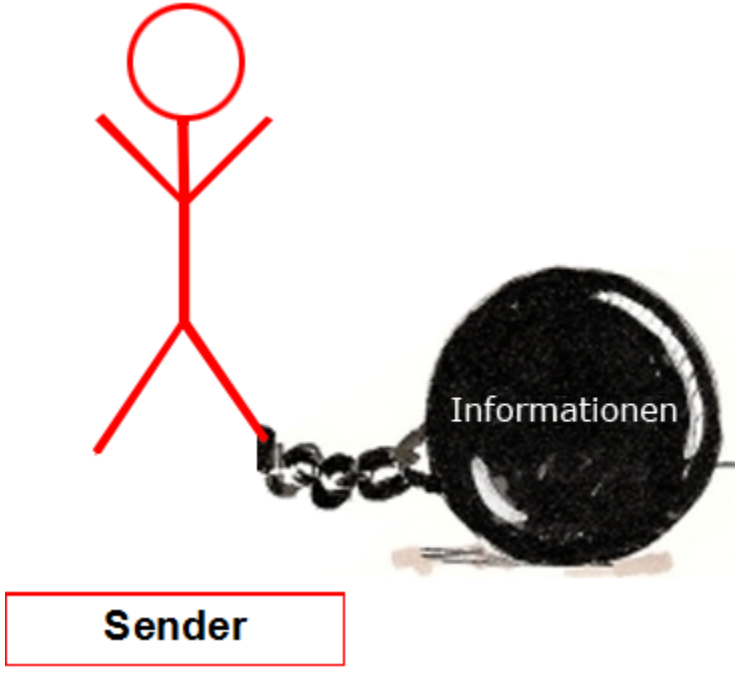
Verfügbarkeit



Sicherheitsanforderungen II

Authentizität

Nichtabstreitbarkeit



Es gibt noch weitere: Anonymität von Benutzern, Nicht-Duplizierbarkeit von elektronischem Geld, …

Frage

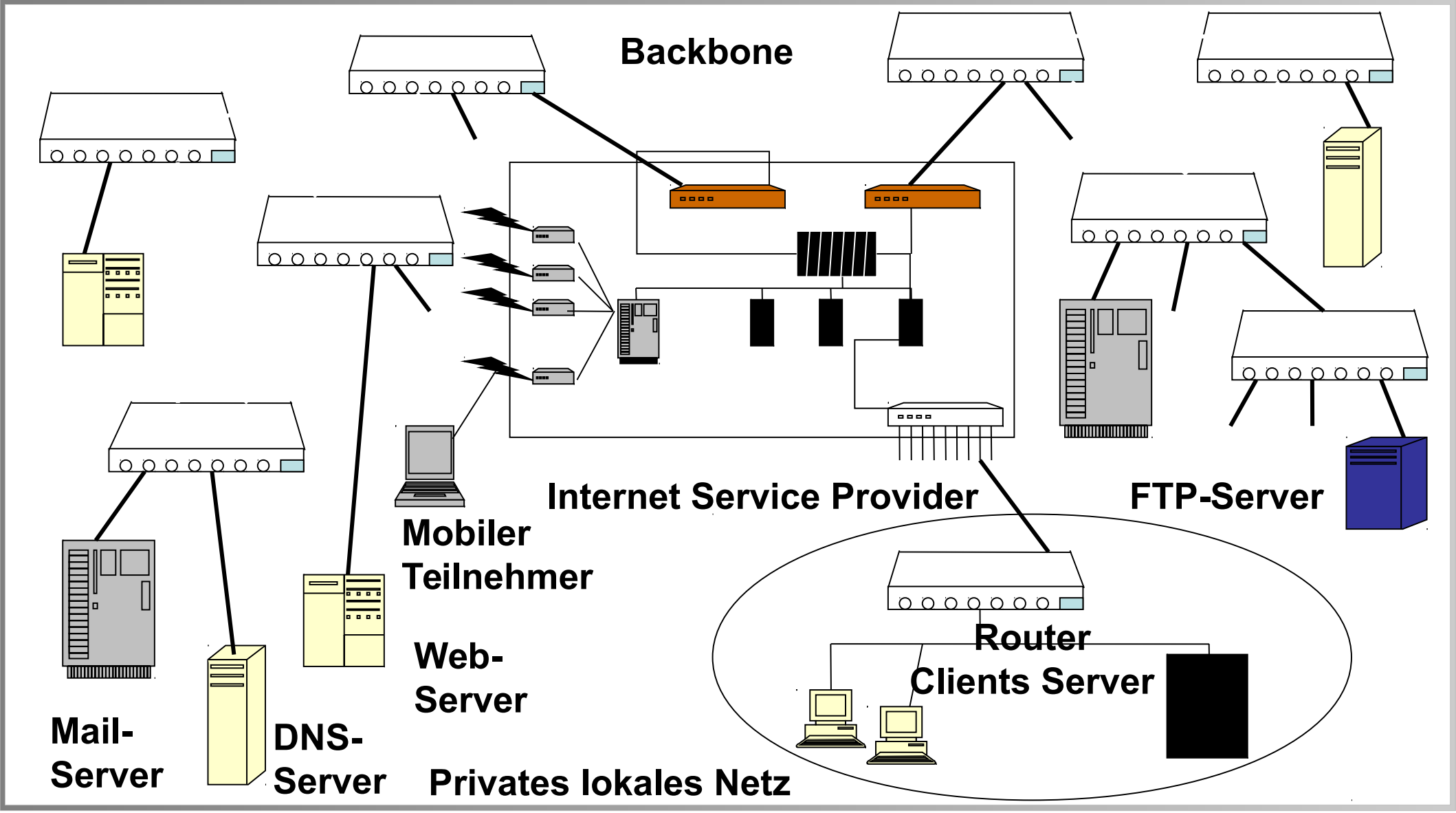
Gebe zwei der genannten Sicherheitsanforderungen an, die sich gegenseitig ausschliessen.

Frage

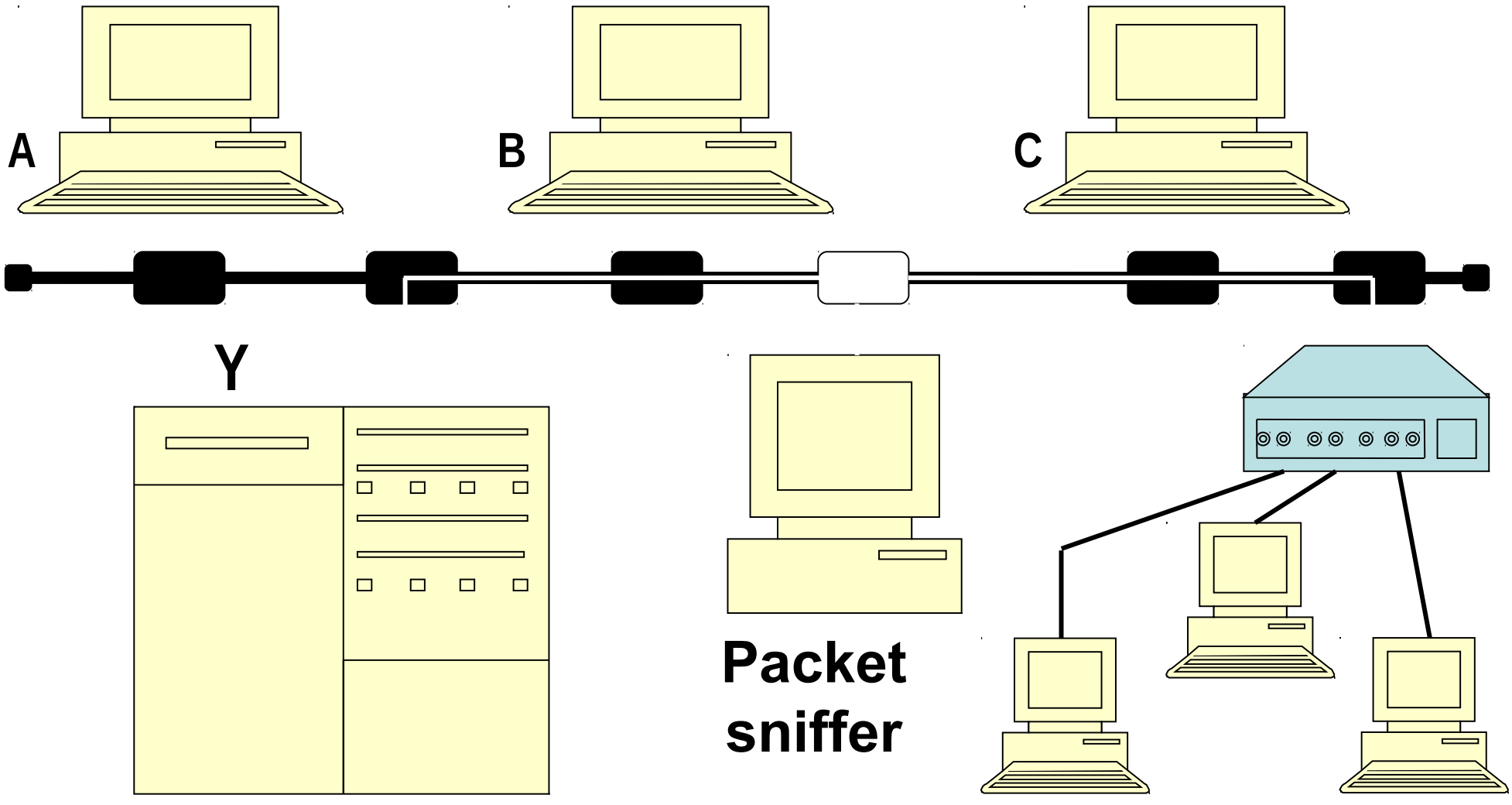
Gebe zwei der genannten Sicherheitsanforderungen an, die sich gegenseitig ausschliessen.

- Zum Beispiel Nicht-Abstreitbarkeit und Anonymität.

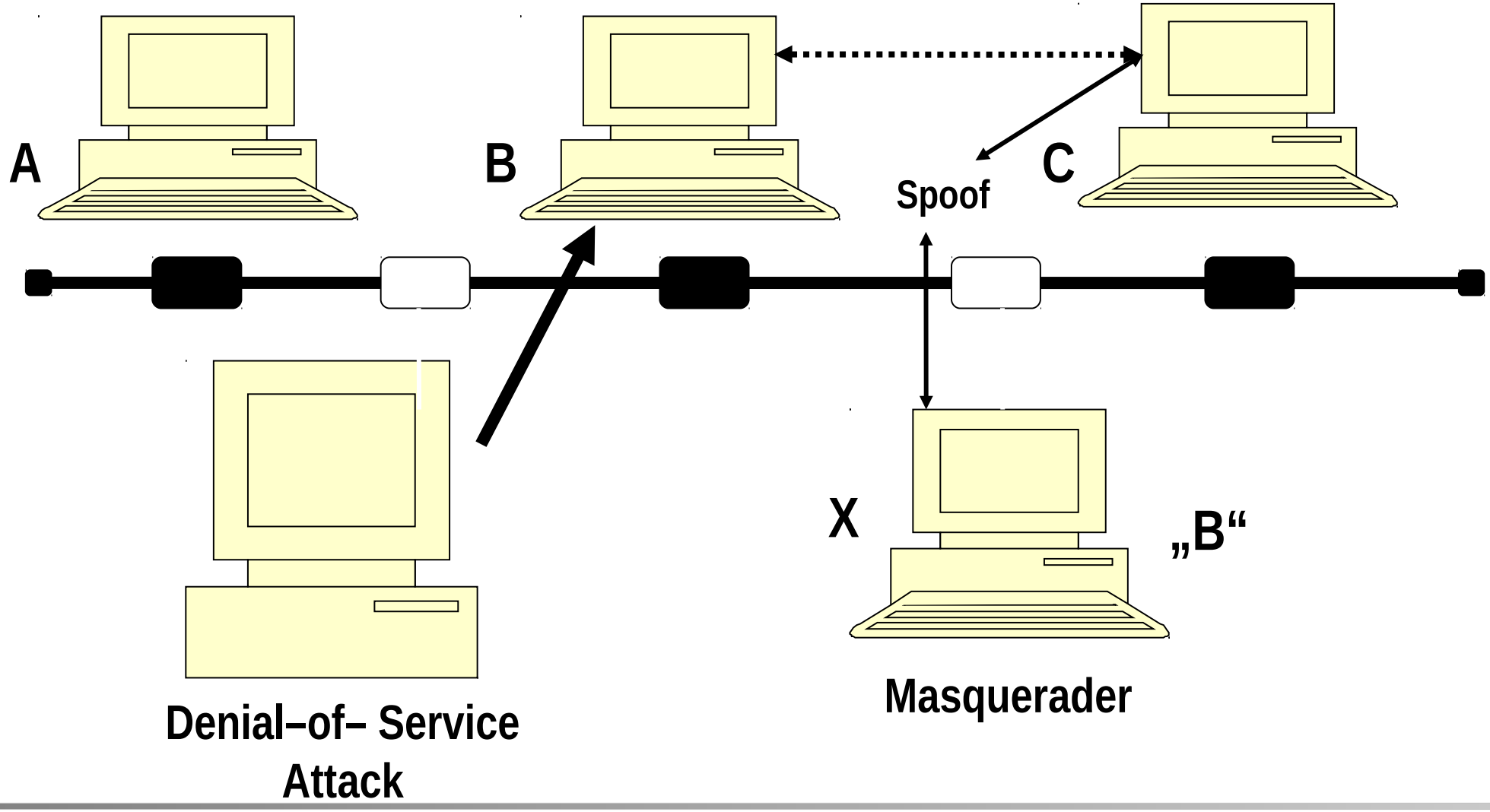
Das Internet



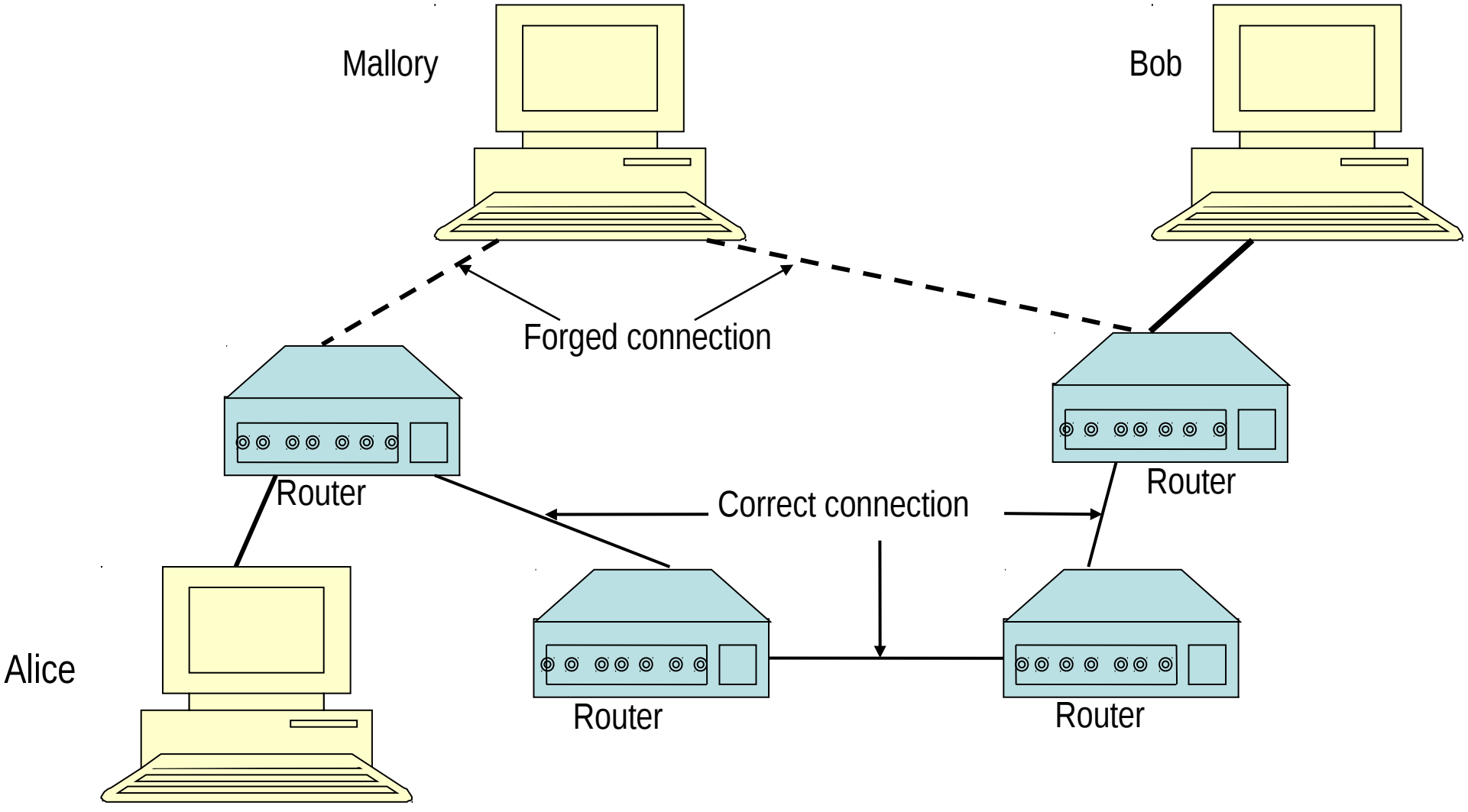
Internet-Angriffe: Abhören



Internet-Angriffe II: Masquerading (Spoofing)

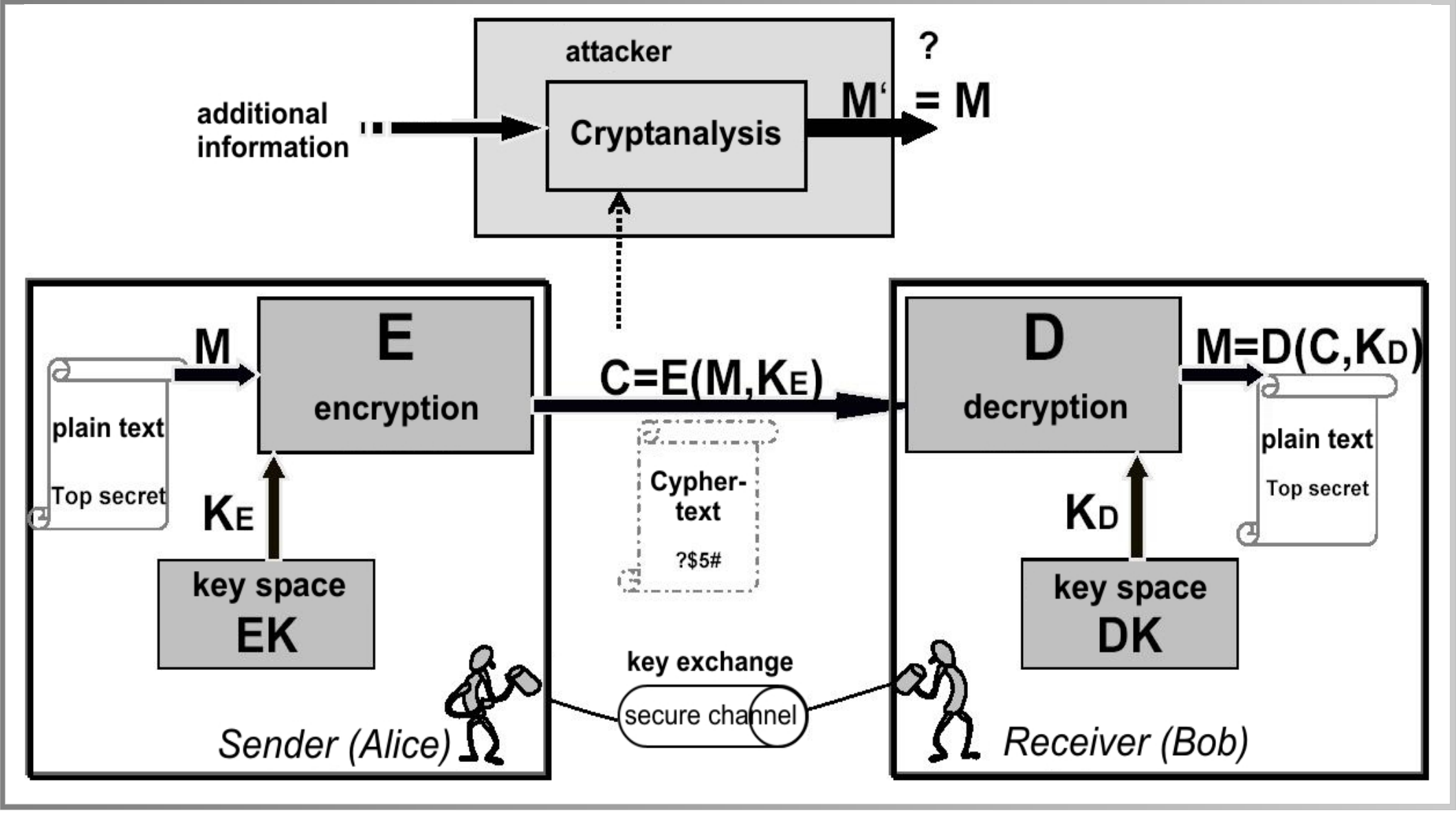


Internet-Angriffe III: „Man-in-the-Middle“

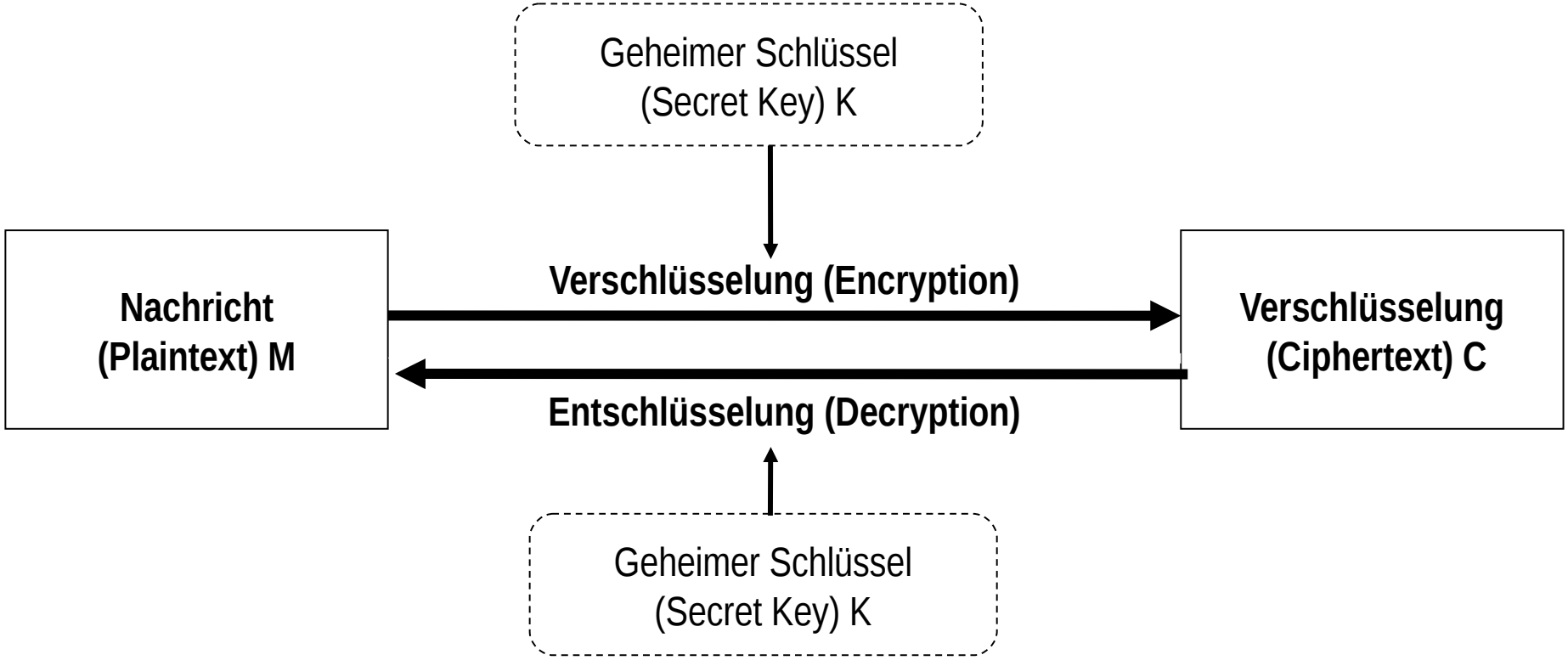


- Sicherheits-Anforderungen und -Risiken
- Kryptographie

Abwehr: Kryptographie



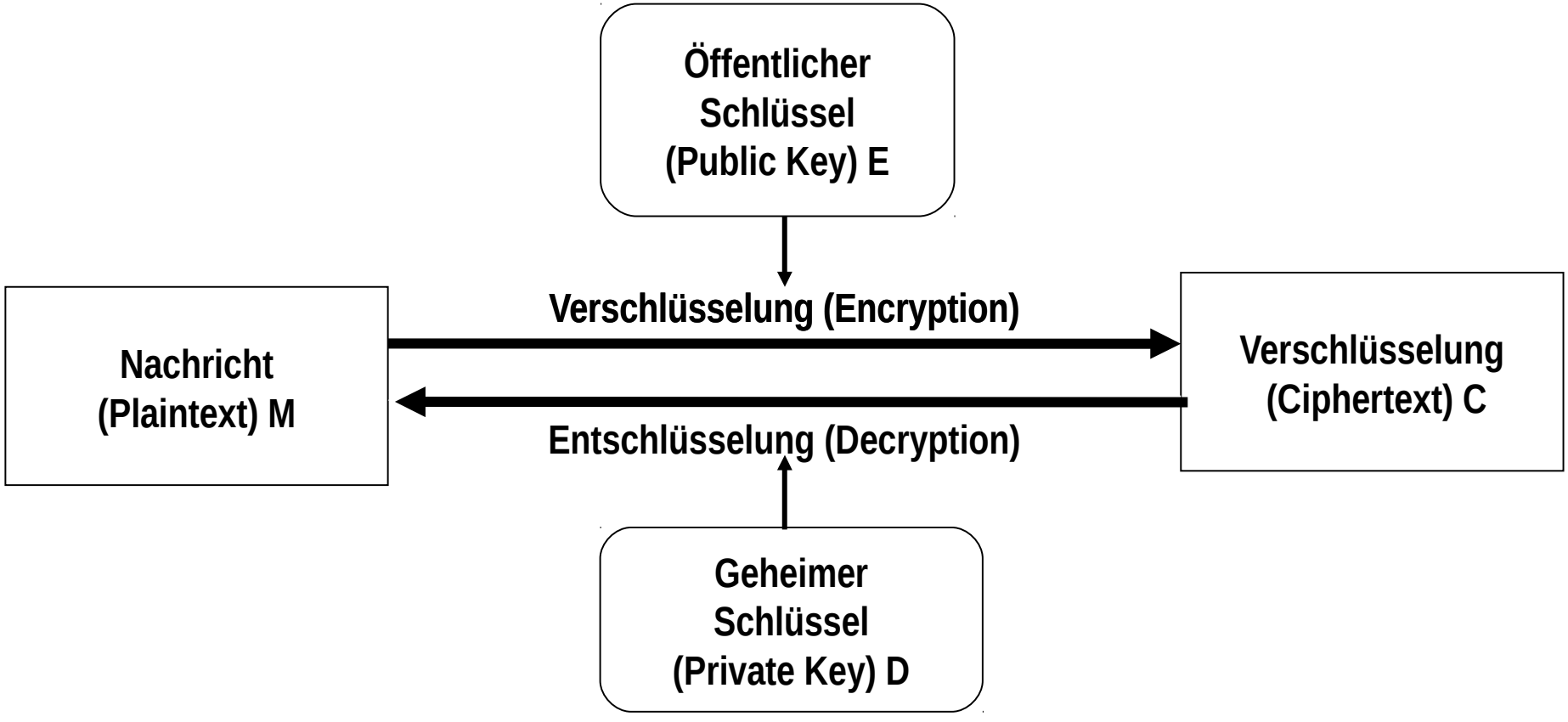
Symmetrische Verschlüsselung



Beziehungen:

$$C = K(M) \text{ und } M = K(C)$$

Asymmetrische Verschlüsselung



Beziehungen: $C = E(M)$ und $M = D(C)$

Frage

Menge möglicher zu verschlüsselnder Texte kann **sehr klein** sein (z.B. nur Nachrichten “ja” oder “nein”).

Welches Problem ergibt sich bei einem deterministischen **Public-Key-Verfahren** ?

Frage

Menge möglicher zu verschlüsselnder Texte kann **sehr klein** sein (z.B. nur Nachrichten “ja” oder “nein”).

Welches Problem ergibt sich bei einem deterministischen **Public-Key-Verfahren** ?

- Man kann alle möglichen Plaintexte verschlüsseln und mit dem vorliegenden Cyphertext vergleichen.

Kryptographische Algorithmen

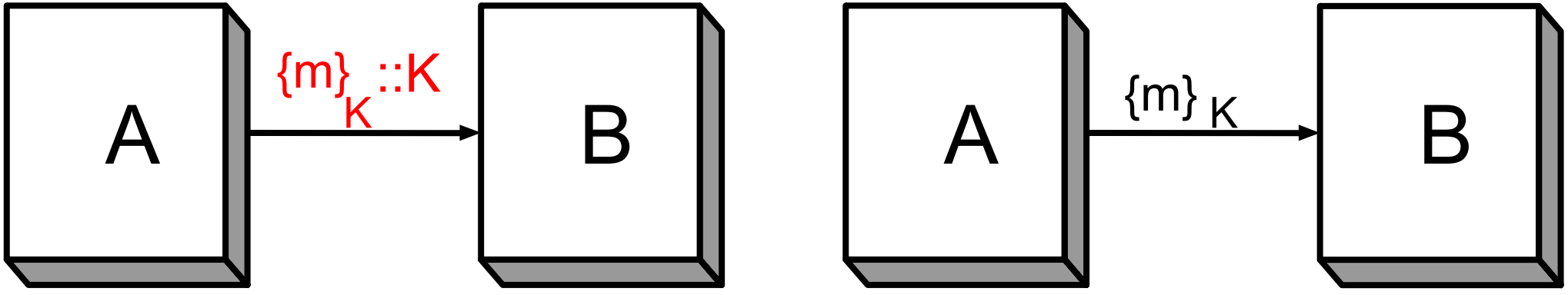
Symmetrisch:

- Digital Encryption Standard (DES), 3DES
- Advanced Encryption Standard (AES): Ryndael 2001

Asymmetrisch:

- RSA (Rivest/Shamir/Adleman): Integer-Faktorisierung
- ElGamal: diskreter Logarithmus
- Diffie-Hellman: Sitzungsschlüssel generieren

Symmetrische Verschlüsselung vs. Vertraulichkeit

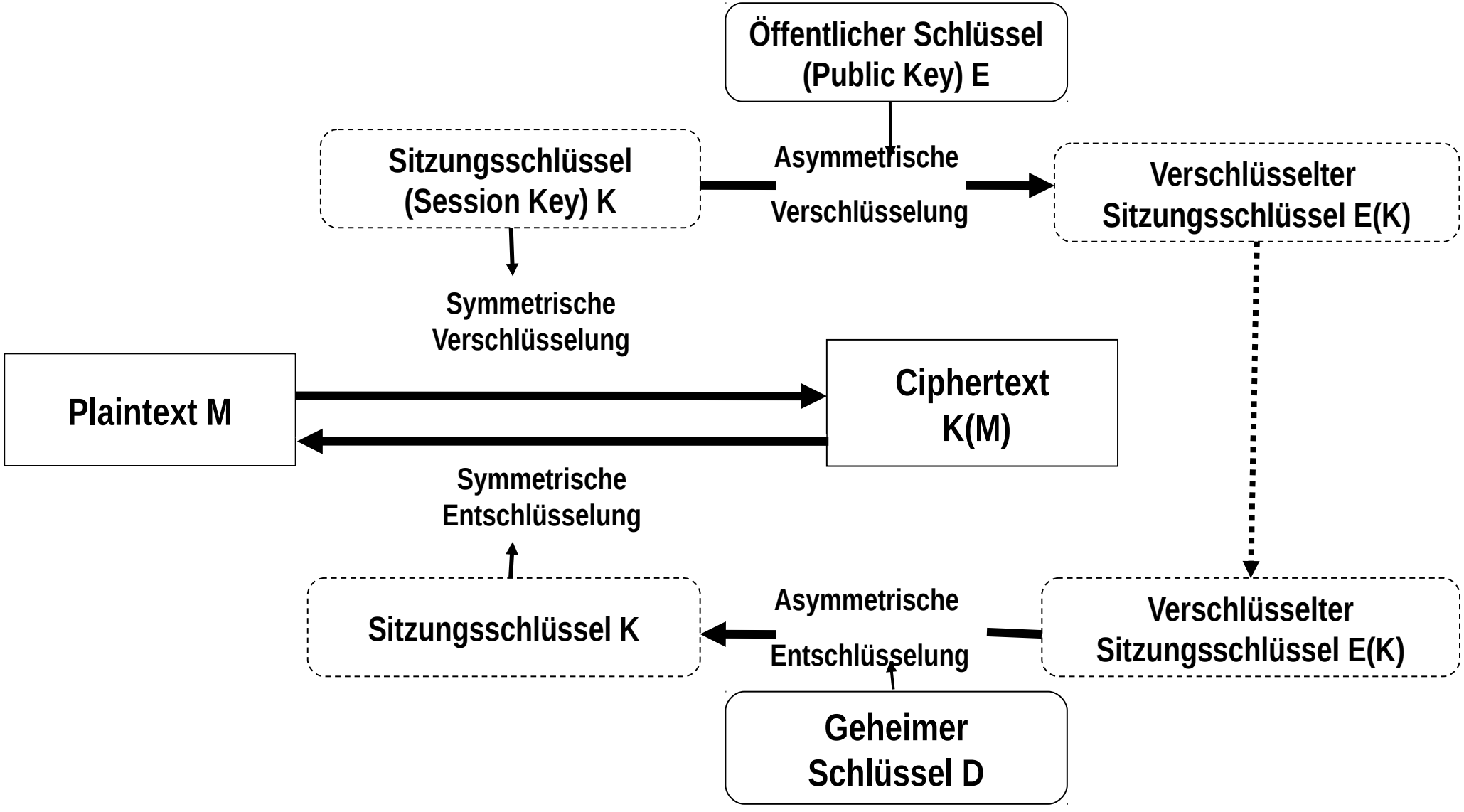


Gegen passiven Angreifer: **Vertraulichkeit** von *m*…

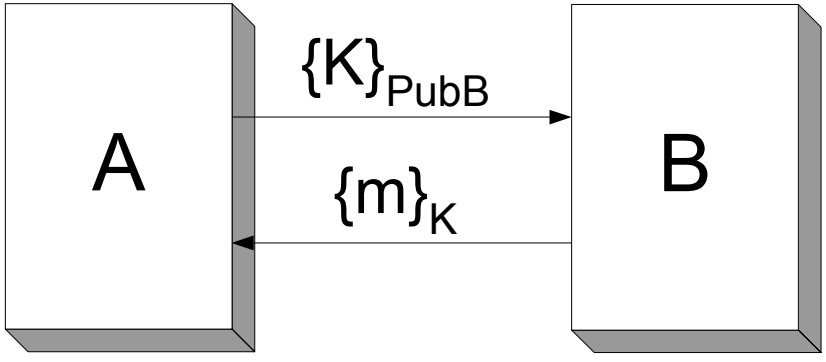
- bei Versenden von $\{m\}_K::K$ nicht bewahrt,
- bei Versenden von $\{m\}_K$ bewahrt (Annahme: Angreifer bekommt *K* nicht auf anderem Wege)

(wobei :: Konkatenation, $\{m\}_K$ **Verschlüsselung** von *m* mit symmetrischem Schluessel *K*).

Hybride Verschlüsselung



Hybrid vs. Vertraulichkeit

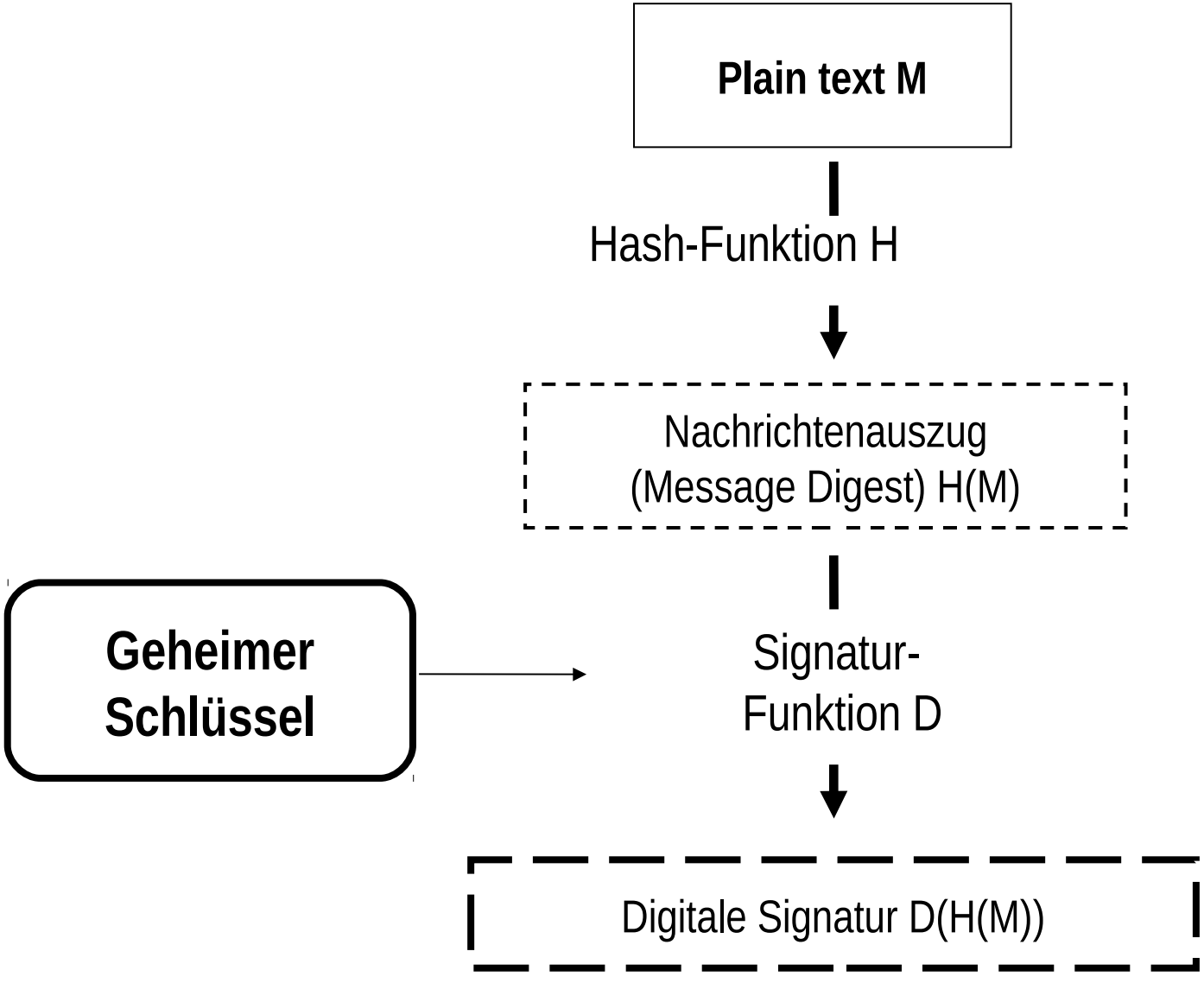


Für symmetrischen Schlüssel *K* und öffentlichen (asymm.) Schlüssel *PubB*:

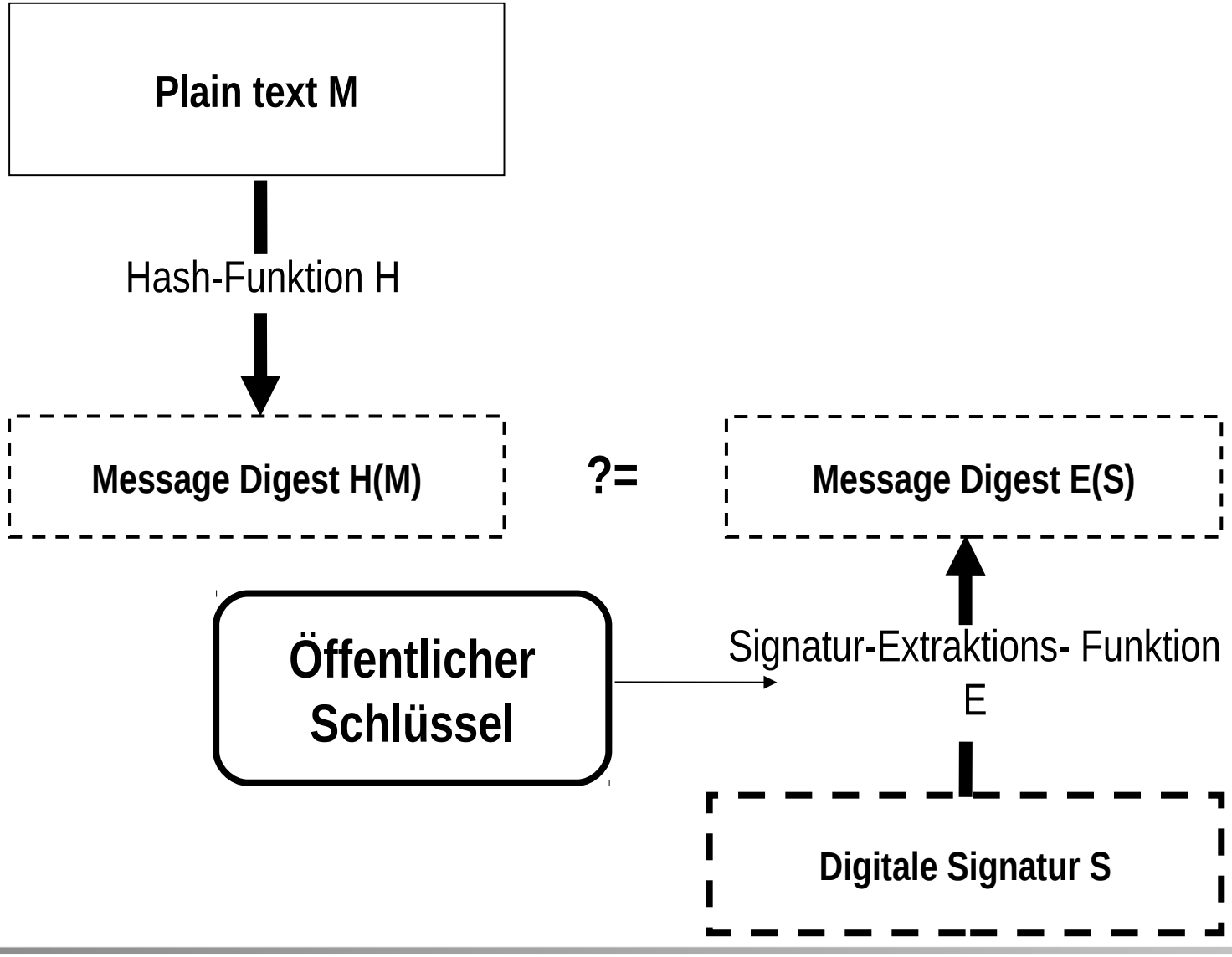
Vertraulichkeit von *m* nicht bewahrt gegen Angreifer, der Nachrichten löschen und einfügen kann.

Vertraulichkeit von *m* bewahrt gegen passiven Angreifer.

Erzeugung digitaler Signaturen



Verifikation digitaler Signaturen



Frage

RSA-Signaturalgorithmus *D* hat die

Homomorphie-Eigenschaft, dass:

D(M1::M2)=D(M1)::D(M2)

für alle **Nachrichten** *M1*, *M2*.

Wie könnte Angreifer bei direkter Anwendung des Signaturalgorithmus (ohne Verwendung der Hashfunktion) den Geldbetrag in der **Signatur** *D* (“Ich schulde Dir 10 EUR.”) auf 100 EUR erhöhen, ohne den Algorithmus brechen zu müssen (wenn Zeichenketten Konkatenationen von Zeichen sind) ?

Frage

RSA-Signaturalgorithmus *D* hat die

Homomorphie-Eigenschaft, dass:

D(M1::M2)=D(M1)::D(M2)

für alle **Nachrichten** *M1*, *M2*.

Wie könnte Angreifer bei direkter Anwendung des Signaturalgorithmus (ohne Verwendung der Hashfunktion) den Geldbetrag in der **Signatur** *D* (“Ich schulde Dir 10 EUR.”) auf 100 EUR erhöhen, ohne den Algorithmus brechen zu müssen (wenn Zeichenketten Konkatenationen von Zeichen sind) ?

Angreifer bekommt:

- *D* (“Ich schulde Dir 10 EUR.”) = *D* (“Ich schulde Dir 1”)::D(“0”)::D(“ EUR.”)

Und kann damit erstellen:

- *D* (“Ich schulde Dir 100 EUR.”) = *D* (“Ich schulde Dir 1”)::D(“0”)::D(“0”)::D(“ EUR.”)

Brute-Force-Angriffe

Kosten ($)	40	56	64	80	112	128
100.000	2 s	35 h	1 J	70.000 J	10^{14} J	10^{19} J
1.000.000	0,2 s	3,5 h	37 T	7.000 J	10^{13} J	10^{18} J
10 Mio	20 ms	21 min	4 T	700 J	10^{12} J	10^{17} J
100 Mio	2 ms	2 min	9 h	70 J	10^{11} J	10^{16} J
1 Mrd	0,2 ms	13 s	1 h	7 J	10^{10} J	10^{15} J
10 Mrd	20 µs	1 s	5,4 min	245 T	10^{9} J	10^{14} J
100 Mrd	2 µs	0,1 s	32 s	24 T	10^{8} J	10^{13} J
10^{12}	0,2 µs	10 ms	3 s	2,4 T	10^{7} J	10^{12} J
10^{13}	20 ns	1 ms	0,3 s	6 h	10^{6} J	10^{11} J

NSA ???

Schlüssellänge in Bit

[Schneier: Angewandte Kryptographie]

(A-)Symmetrische Schlüssellängen

Vergleichbare Sicherheit von symmetrischen und asymmetrischen Schlüssellängen

Schlüssellänge (in Bits)

Symmetrisch	**Asymmetrisch**
56	384
64	512
80	768
112	1792
128	2304

[Schneier: Angewandte Kryptographie]

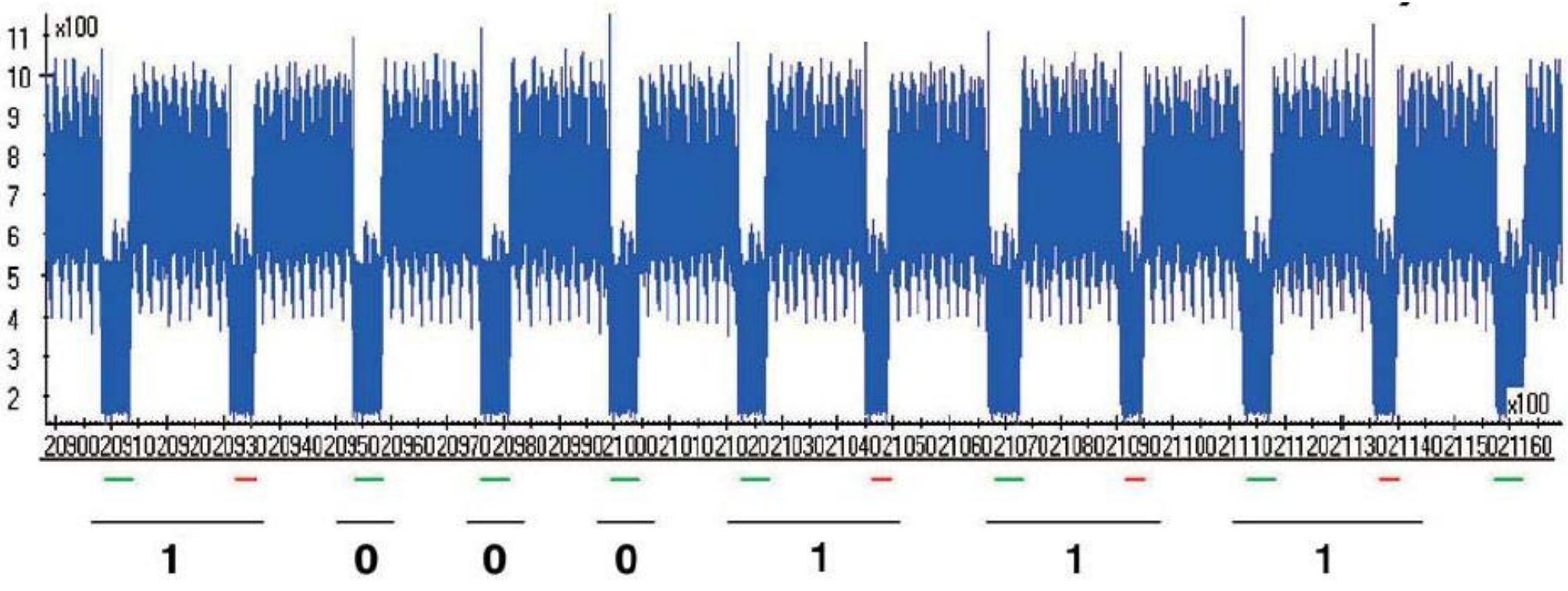
Schlüssellängen

Informationsart	Lebensdauer	Bits (min.)
militärtaktische Informationen	Min. / Stunden	56 – 64
Produktankündigungen, Firmen-zusammenschlüsse, Zinssätze	Tage / Wochen	64
langfristige Geschäftsplanungen	mehrere Jahre	64
Wirtschaftsgeheimnisse (Coca Cola)	Jahrzehnte	112
geheime Daten zur Wasserstoffbombe	über 40 Jahre	128
Identität von Spionen	über 50 Jahre	128
personenbezogene Daten	über 50 Jahre	128
Geheimdiplomatie	über 65 Jahre	> 128
Daten der US-Volkszählung	100 Jahre	> 128

[Schneier: Angewandte Kryptographie]

„Side-Channel“-Angriffe

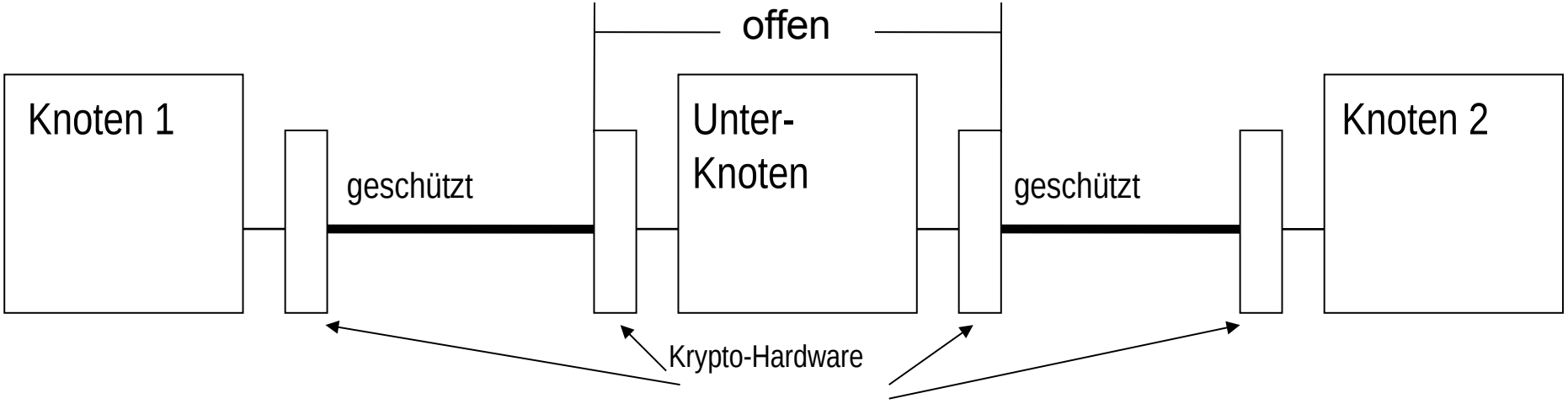
Vertrauliche **kryptographische Daten** rekonstruieren
(z.B. externen Stromverbrauch von Smartcard beobachten)



Reichweite der Verschlüsselung I

„Link encryption“

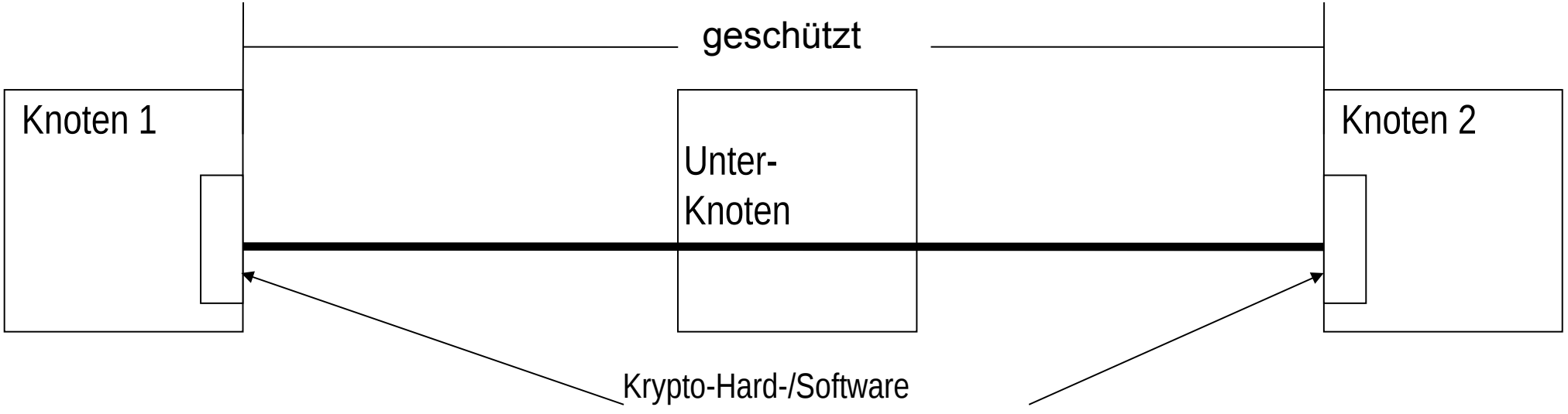
- Nur direkte Knotenverbindungen verschlüsselt.
- Einfache Konstruktion.
- Unterstützt durch Controller-Hardware, transparent für Software.
- Daten in Netzwerkknoten als Plaintext.



Reichweite der Verschlüsselung II

„End-to-End-Encryption“

- Daten durchgehend verschlüsselt.
- Komplexer zu implementieren.
- Intransparent für Software, separate Behandlung von Adressen und Daten



Cryptool

Freies Programmpaket zum **„Erfahren“ von Kryptographie** (www.cryptool.de; B. Esslinger (Deutsche Bank)).

Kryptoverfahren anwenden und analysieren.

Fast alle **State-of-the-Art** Kryptofunktionen.

- **klassische** Verfahren (Cäsar,…) und Analysen (Entropie, gleitende Häufigkeit,…)
- **Moderne** (a-)symmetrische Verfahren (3DES, AES, RSA,…), Analysen
- Signaturen, Zufallszahlen, Hash, MACs,…

Zusammenfassung

- Sicherheitsanforderungen
- Angriffe auf Netzwerke
- Symmetrische und asymmetrische Kryptographie
- Signaturen
- Reichweite von Verschlüsselung