Willkommen zur Vorlesung

Modellbasierte Softwaretechniken für sichere Systeme

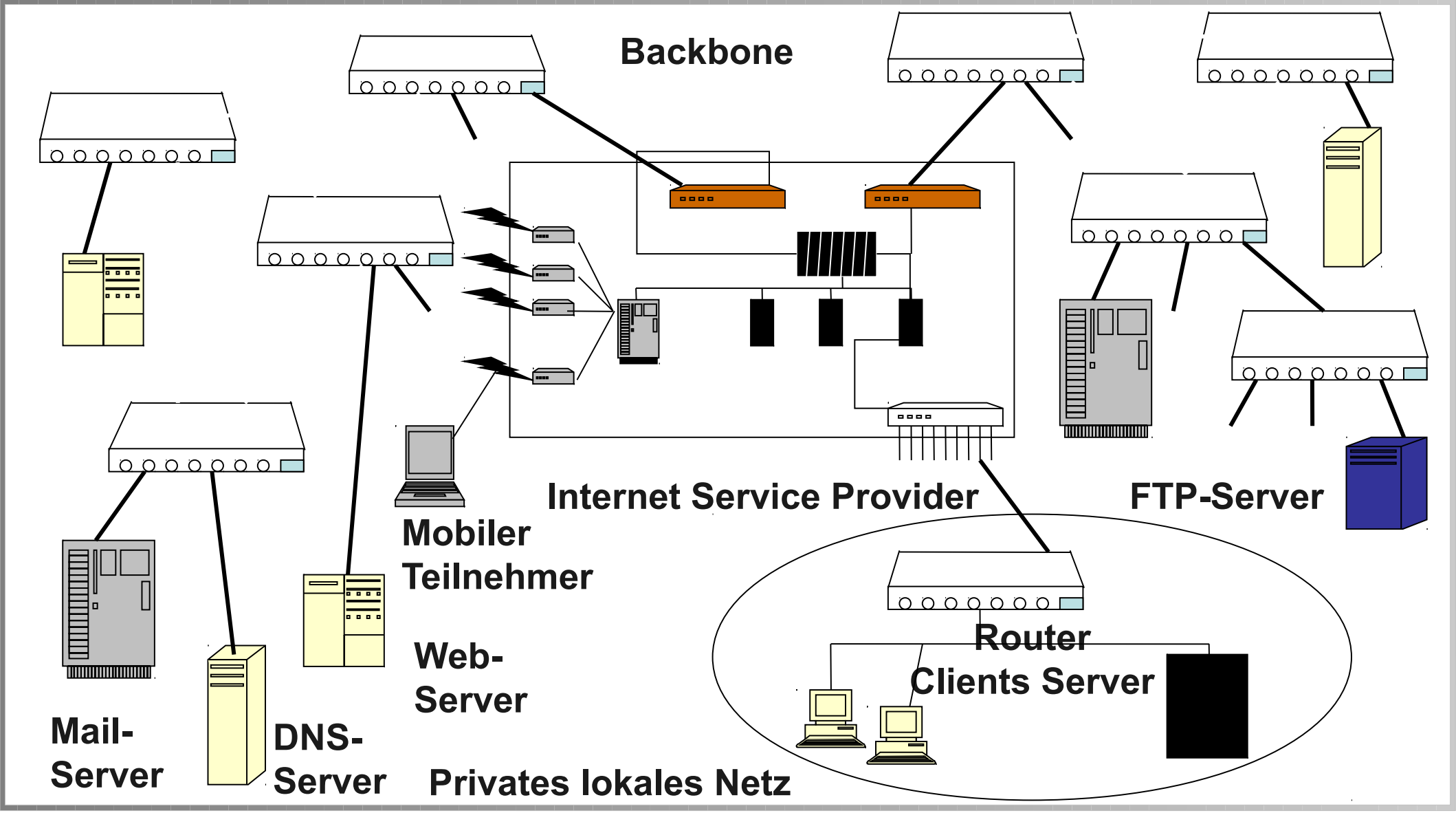
im Sommersemester 2012

Prof. Dr. Jan Jürjens

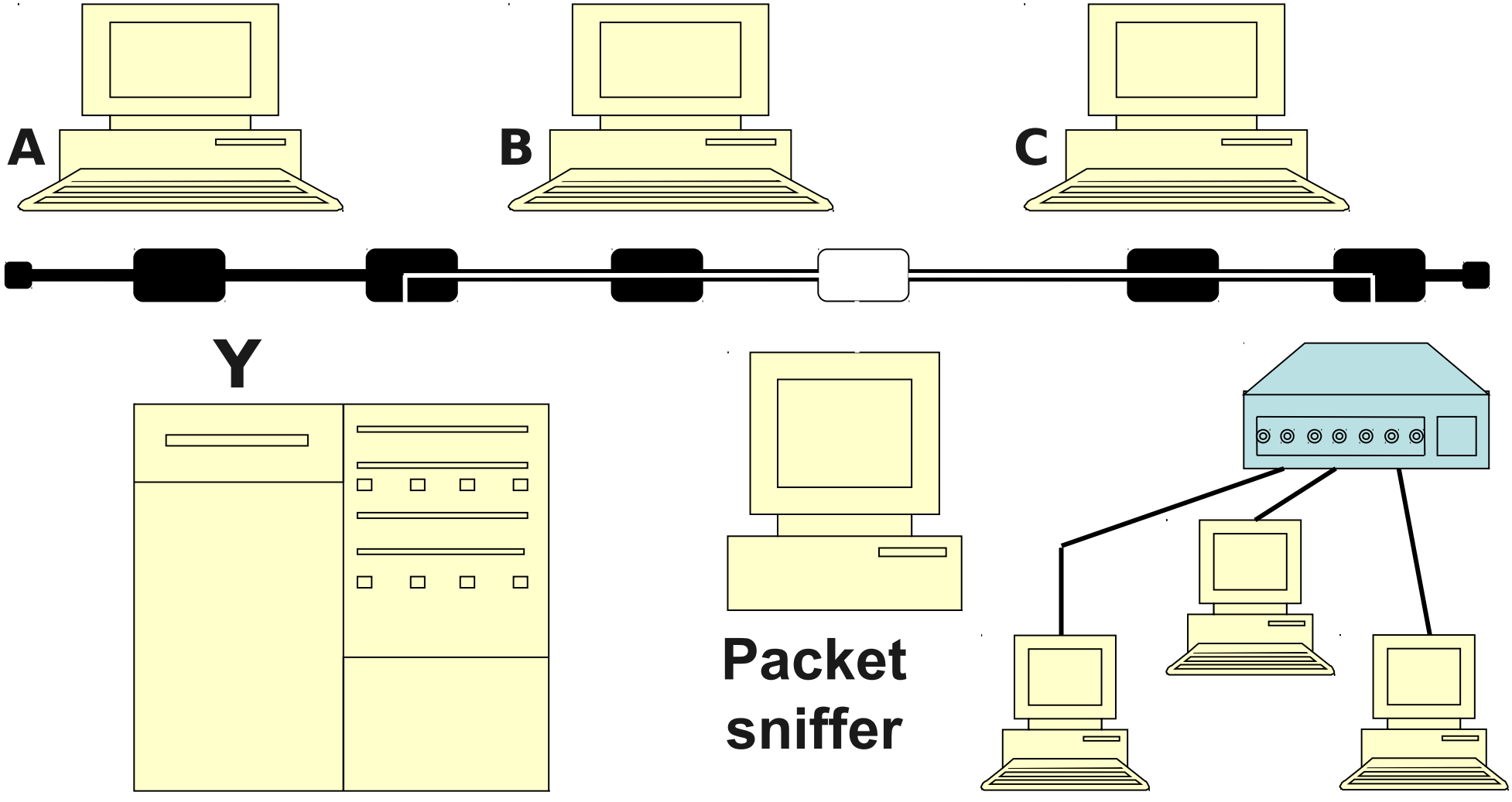
TU Dortmund, Fakultät Informatik, Lehrstuhl XIV

2. Netzwerksicherheit und Kryptographie

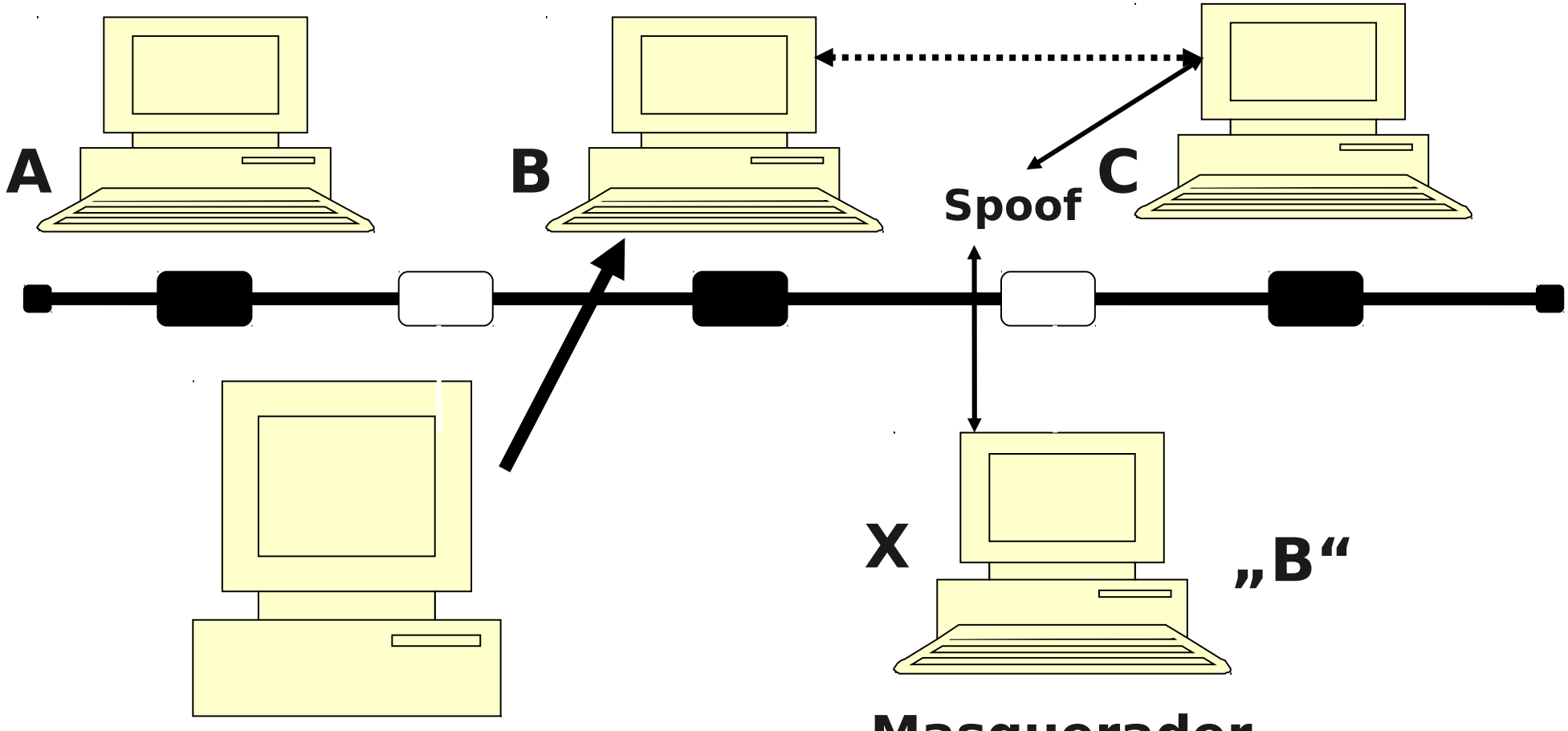
Das Internet



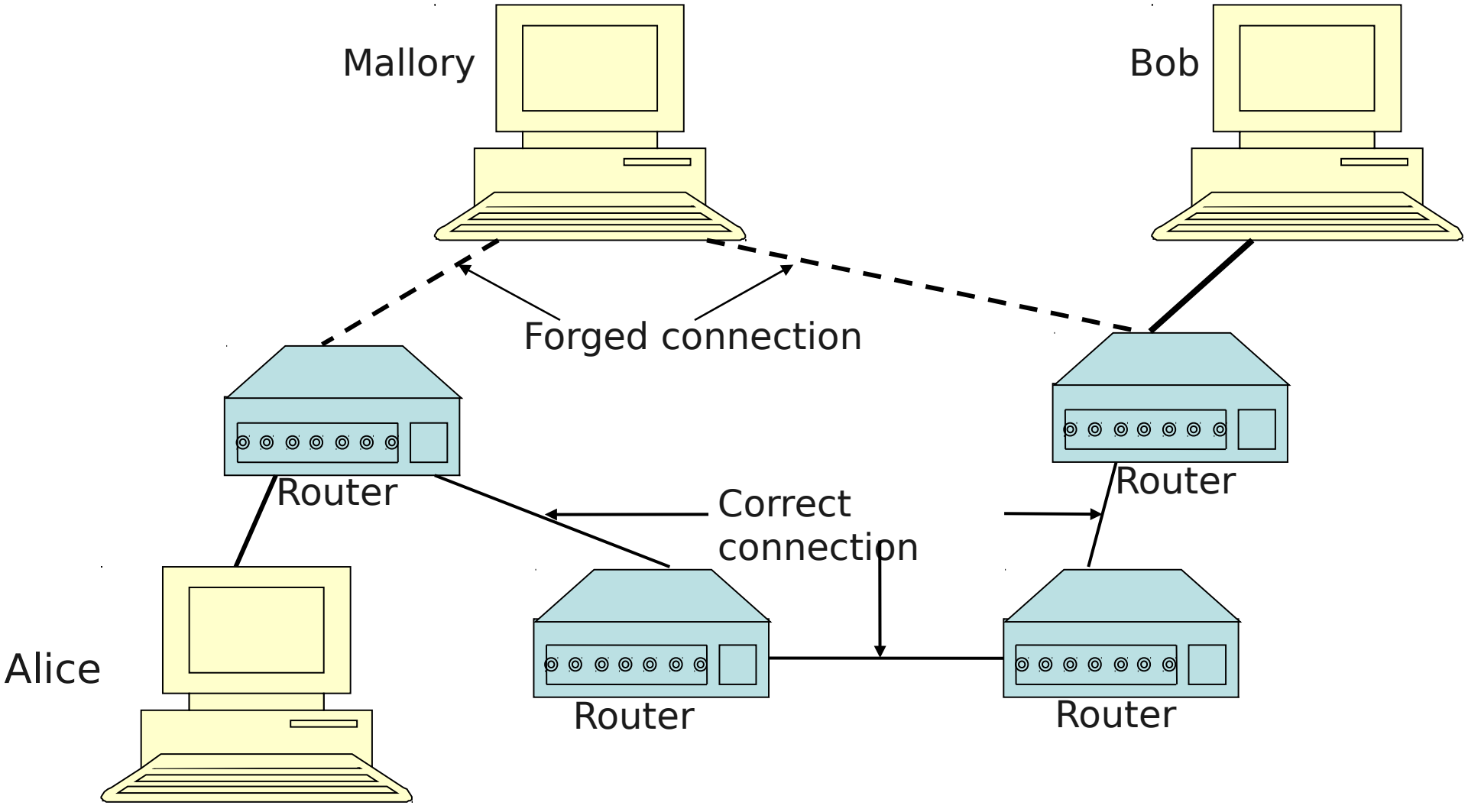
Internet-Angriffe: Abhören



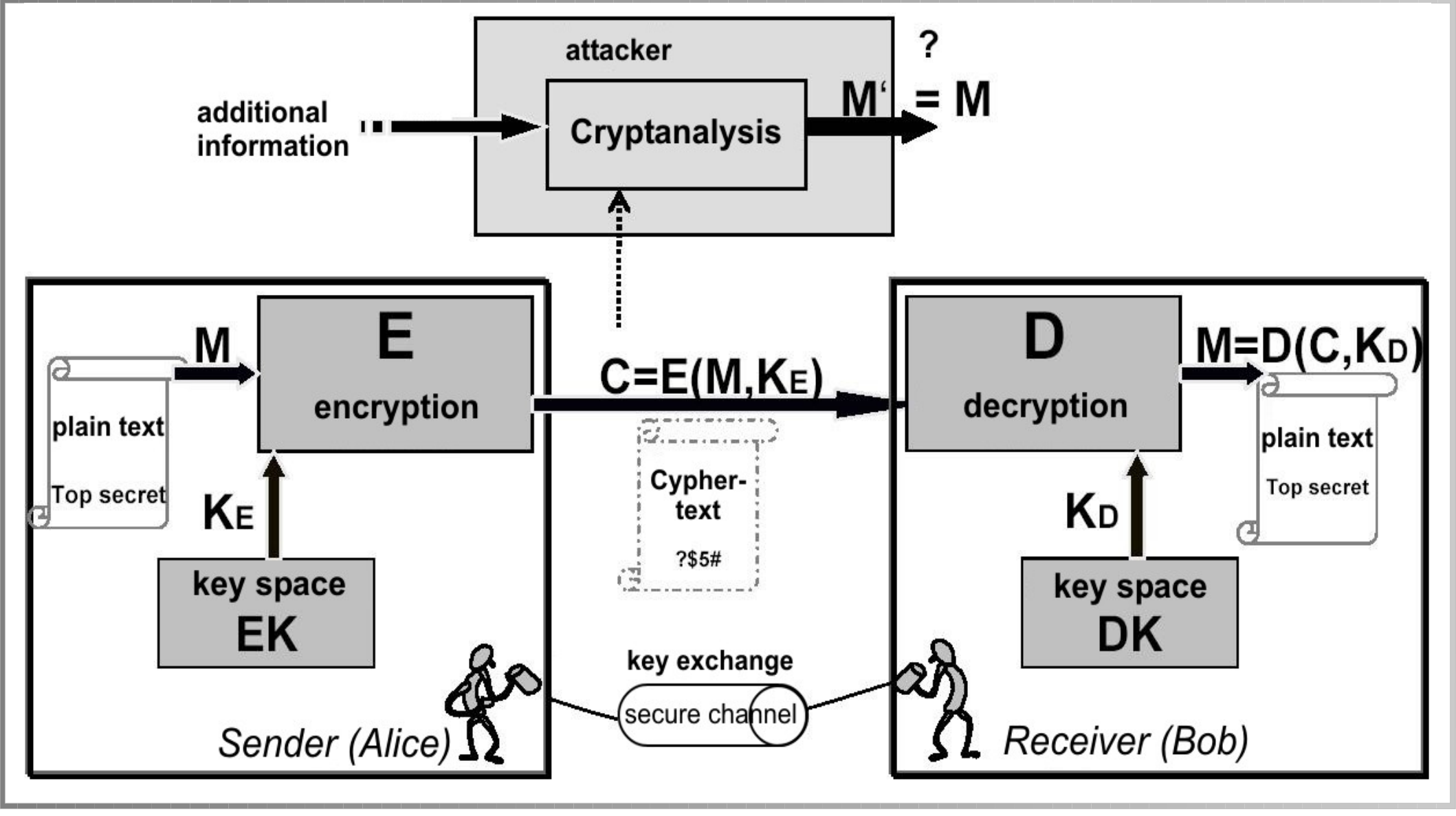
Internet-Angriffe II: Masquerading (Spoofing)



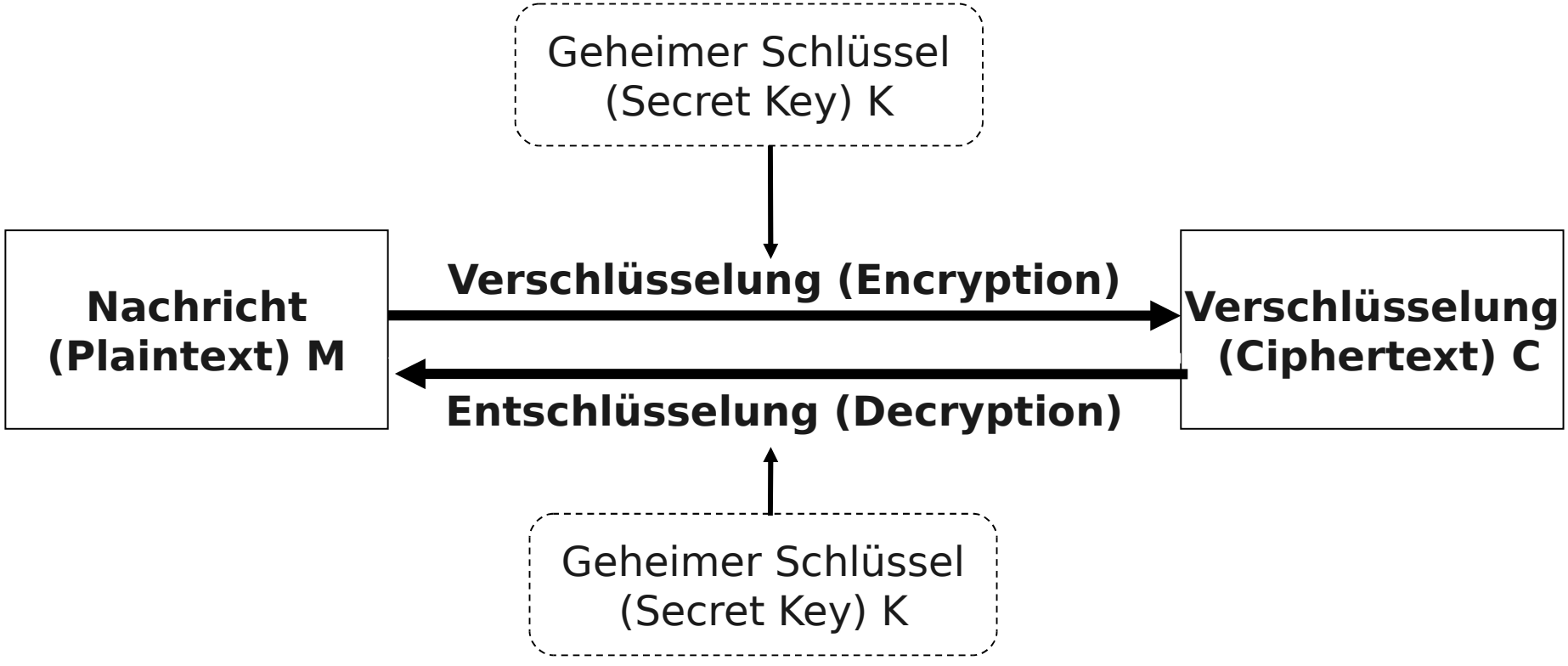
Internet-Angriffe III: „Man-in-the-Middle“



Abwehr: Kryptographie

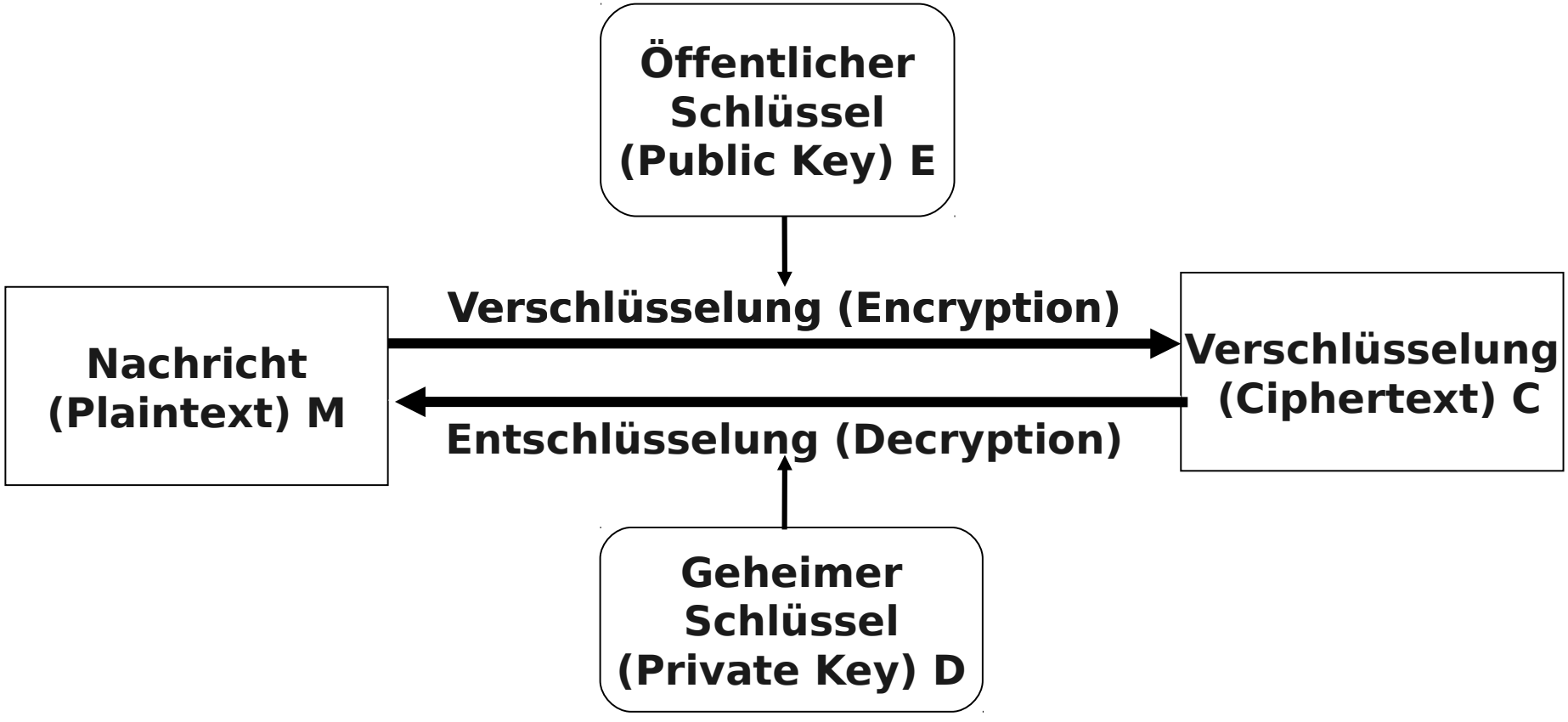


Symmetrische Verschlüsselung



Beziehungen: $C = K(M)$ und $M = K(C)$

Asymmetrische Verschlüsselung



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

Frage

a) Wenn man bedenkt, dass die Menge der möglichen zu verschlüsselnden Texte sehr klein sein kann (z.B. nur die Nachrichten “ja” oder “nein”), welches Problem ergibt sich bei einem deterministischen Public-Key-Verfahren ?

Kryptographische Algorithmen

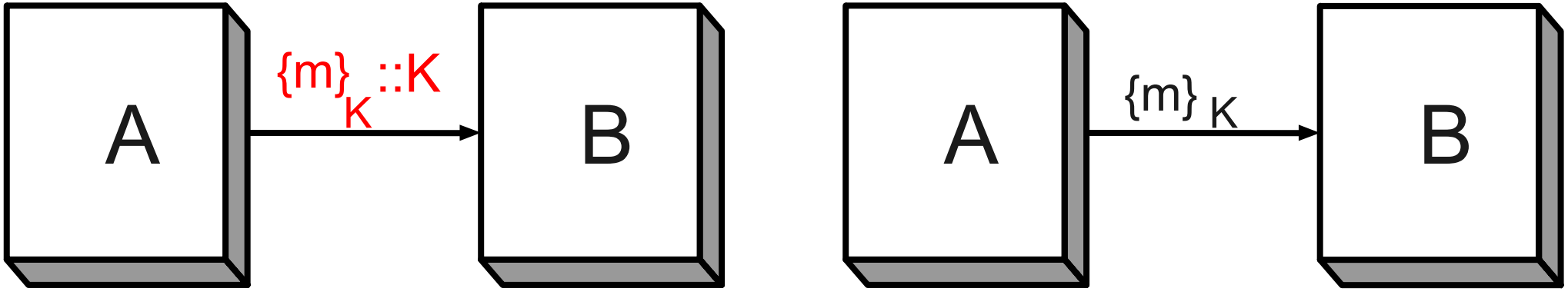
Symmetrisch:

- 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

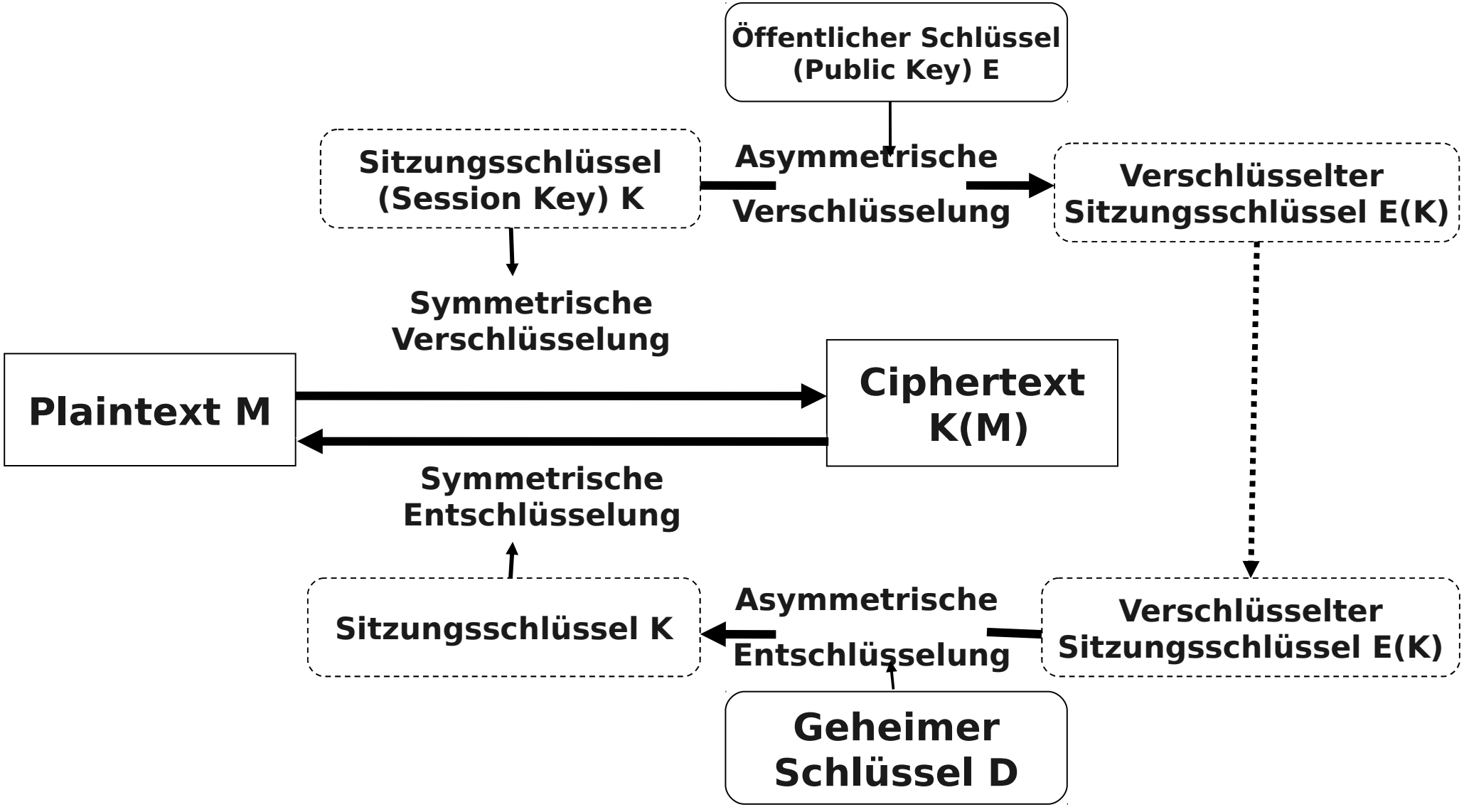


Gegen passiven Angreifer: Vertraulichkeit von *m*…

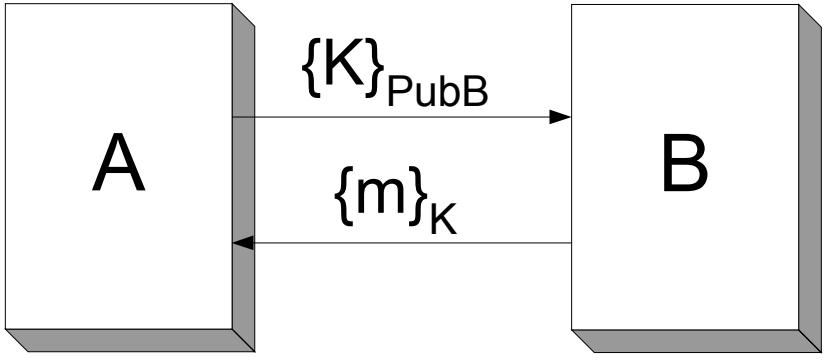
- bei Versenden von $\{m\}_K::K$ nicht bewahrt,
- bei Versenden von $\{m\}_K$ bewahrt (Annahme: Angreifer bekommt *K* nicht auf anderem Wege)

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

Hybride Verschlüsselung



Hybrid vs. Vertraulichkeit

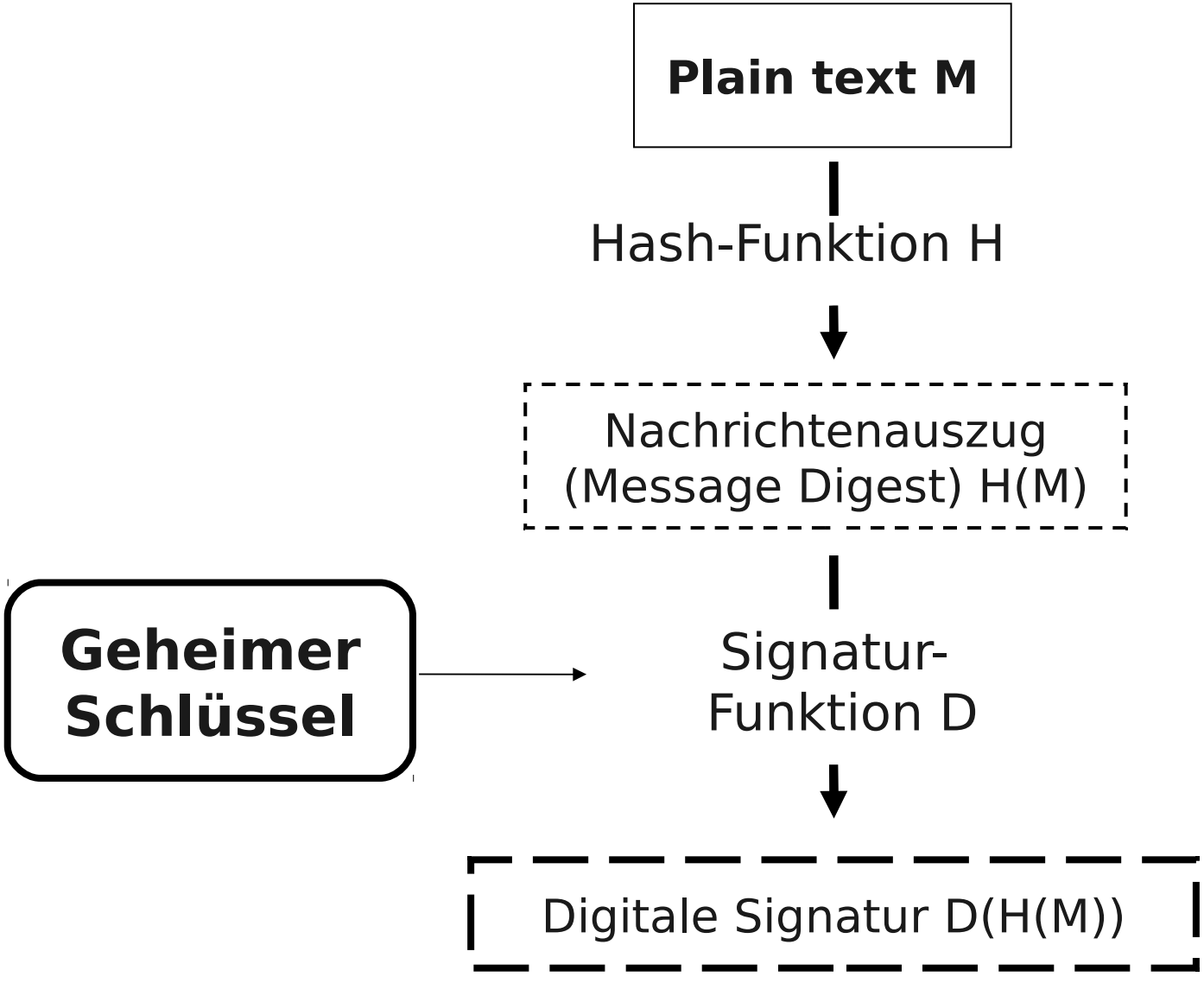


Für symmetrischen Schlüssel *K* und öffentlichen (asymm.) Schlüssel *PubB*:

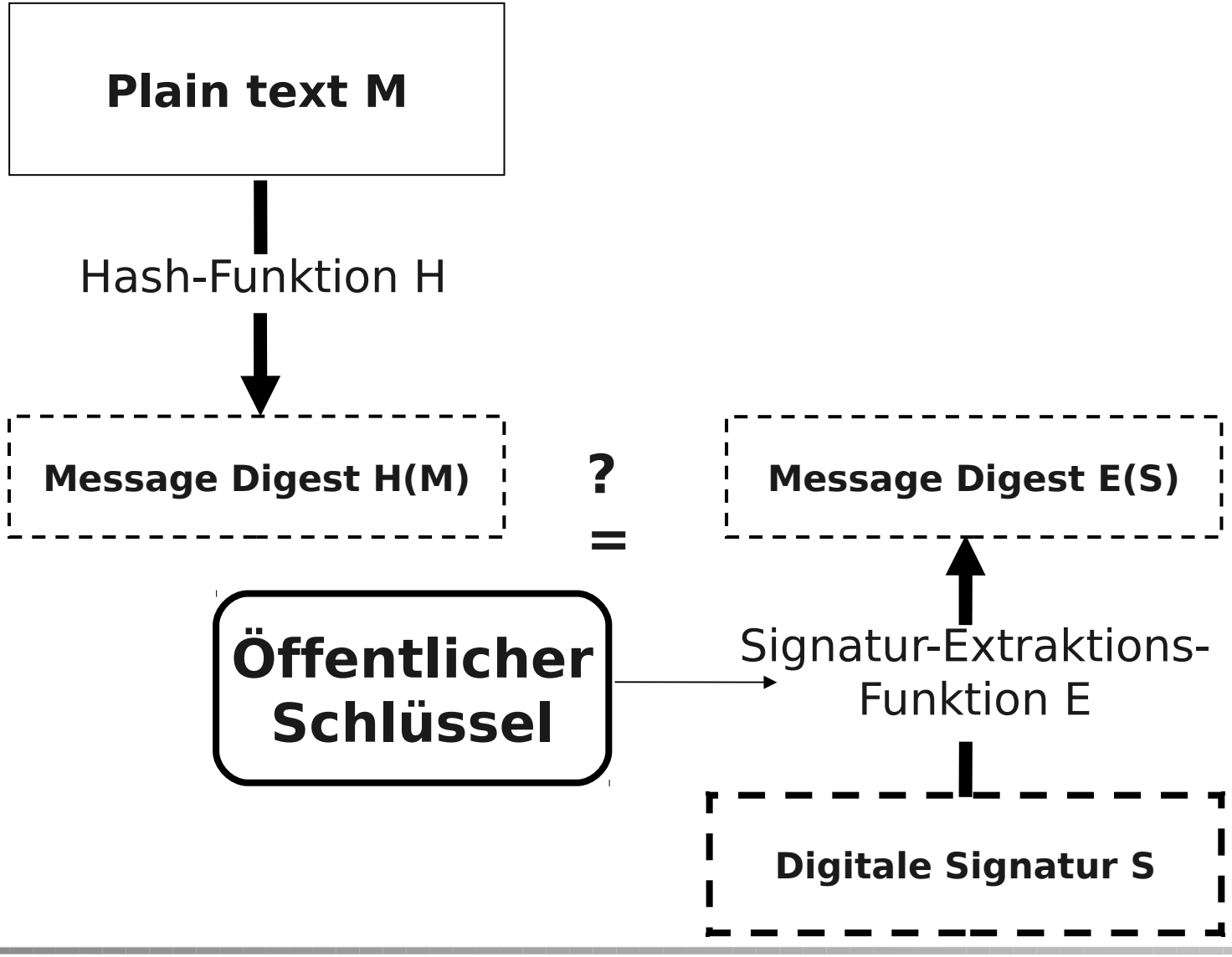
Vertraulichkeit von *m* nicht bewahrt gegen Angreifer, der Nachrichten löschen und einfügen kann.

Vertraulichkeit von *m* bewahrt gegen passiven Angreifer.

Erzeugung digitaler Signaturen



Verifikation digitaler Signaturen



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]

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

[Schneier: Angewandte Kryptographie]

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

Frage

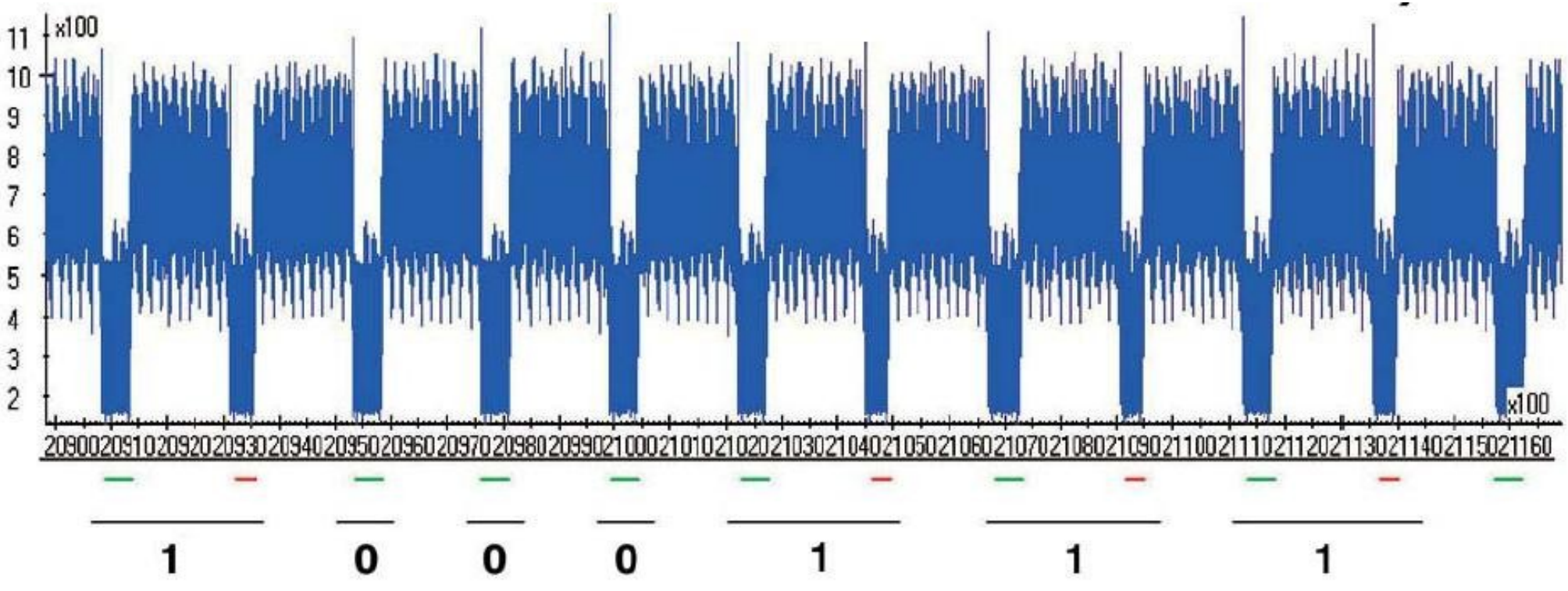
Der RSA-Signaturalgorithmus *D* hat die Homomorphie-Eigenschaft, dass:

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

für alle Nachrichten *M1*, *M2*. Wenn man kein Hash verwenden würde – wie könnte dann ein Angreifer 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) ?

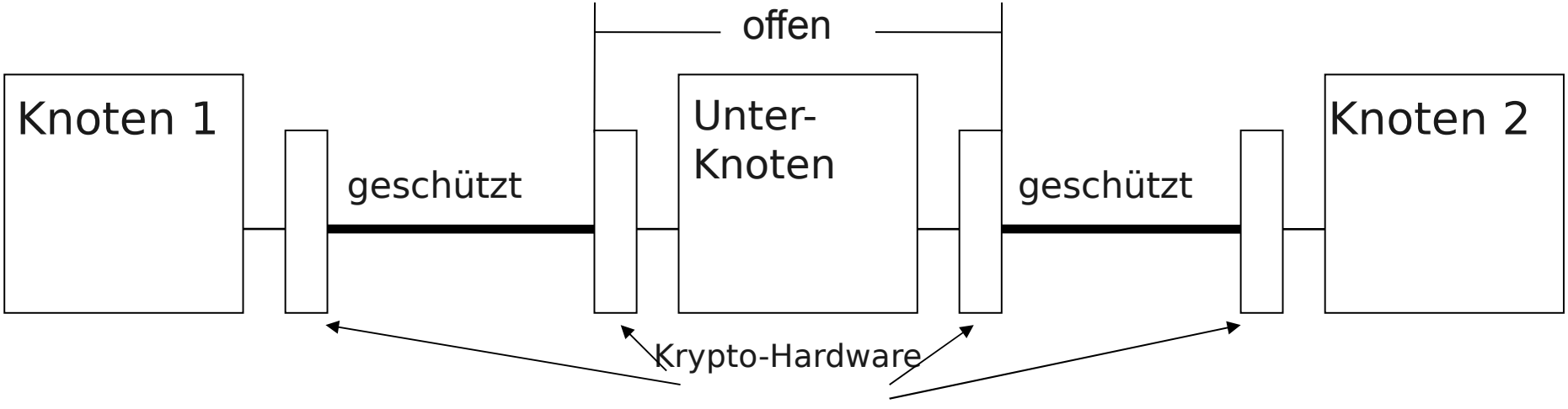
„Side-Channel“-Angriffe

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



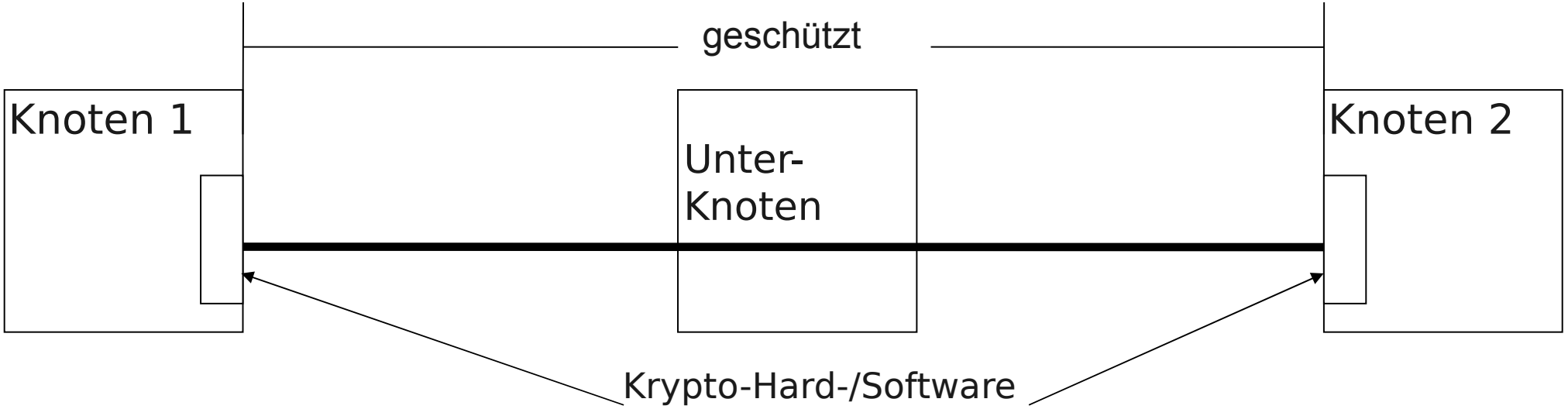
Reichweite der Verschlüsselung I

„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

- Lesson Learned
  - Angriffe auf Netzwerke
  - symmetrische und asymmetrische Kryptographie
  - Signaturen
  - Reichweite von Verschlüsselung