



Kryptographie: Einführung

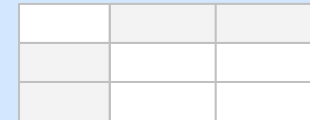
Symmetrische und asymmetrische Systeme
Verschlüsselung und Authentikation
Schlüsselverteilung und Schlüssellängen



Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Kriterien zur Einteilung von Kryptosystemen

- Kryptographische Basisbausteine
 - Konzelationssysteme
 - Authentikationssysteme
 - Hashfunktionen
 - Pseudozufallszahlengeneratoren
- Schlüsselbeziehung Sender–Empfänger
 - Symmetrische Systeme
 - Asymmetrische Systeme
- Alphabet, auf dem die Chiffre operiert
 - Blockchiffre: Operiert auf Blöcken von Zeichen
 - Stromchiffren: Operiert auf einzelnen Zeichen
- Längentreue
- Erreichbare Sicherheit



Anwendungsfall x Schlüsselbeziehung

	Konzelation (Verschlüsselung)	Authentikation
symmetrische	<i>One-time-pad, DES, Triple-DES, AES, IDEA, A5/1 (GSM), A5/2 (GSM) ...</i> <div> <div>GnuPG/PGP</div> <div>WPA2</div> <div>IPSec</div> <div>SSL/TLS</div> </div>	<i>Symmetrische Authentikationscodes, CCM, A3 (GSM), ...</i> <div> <div>SecurID</div> <div>WPA2</div> <div>IPSec</div> <div>SSL/TLS</div> </div>
asymmetrische	<i>RSA, ElGamal, McEliece, ...</i> <div> <div>GnuPG/PGP</div> <div>HBCI</div> <div>SSL/TLS</div> </div>	<i>RSA, ElGamal, DSA, GMR, ...</i> <div> <div>GnuPG/PGP</div> <div>HBCI</div> <div>SSL/TLS</div> </div>

Algorithmus

Anwendung

Erreichbare Sicherheit

- **Sicherheit**

- (informations) theoretisch sicher
- kryptographisch stark (beweisbar)
 - gegen aktive Angriffe
 - gegen passive Angriffe
- wohluntersucht (praktisch sicher)
 - Chaos
 - Zahlentheorie
- geheim gehaltene

komplexitäts-
theoretisch
sicher

- **Kerckhoffs-Prinzip**

- Die Sicherheit eines kryptographischen Verfahrens soll von der Geheimhaltung des kryptographischen Schlüssels abhängen.
 - Geht zurück auf
Auguste Kerckhoffs: La Cryptographie militaire, 1883

Angriffsarten

- Ciphertext Only Attack
 - Angreifer kennt nur Schlüsselttext
- Known Plaintext Attack
 - Angreifer kenn Klartext-Schlüsselttext-Paare
- [Adaptively] Chosen Plaintext (Ciphertext) Attack
 - Adaptively:
 - Angreifer kann in Abhängigkeit vorheriger gewählter Nachrichten neue Nachrichten wählen
 - Non-adaptively:
 - Angreifer muss alle Nachrichten zu Beginn wählen, kann also nicht abhängig vom Verschlüsselungsergebnis, weitere Nachrichten wählen

Angriffsarten

- | | | | | |
|-----------|---|------------------------------|--|---------------------------|
| | | Authentikations-
systeme: | | Konzelations-
systeme: |
| • Brechen | = | Fälschen | | Entschlüsseln |

- Vollständiges Brechen: Finden des Schlüssels
- Universelles Brechen: Finden eines zum Schlüssel äquivalenten Verfahrens
- Nachrichtenbezogenes Brechen: Brechen für einzelne Nachrichten, ohne den Schlüssel selbst in Erfahrung zu bringen
 - selektives Brechen: für eine bestimmte Nachricht
 - existenzielles Brechen: für irgendeine Nachricht

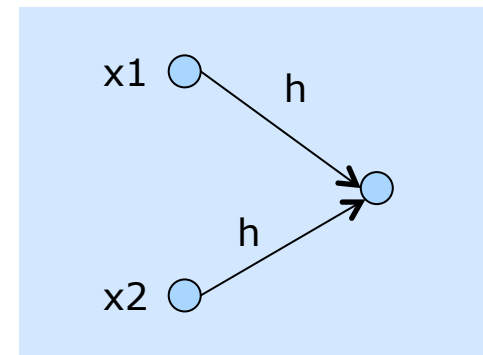
- Aufwand/Kosten:
 - Einmalige Kosten, jeder Schlüssel effizient knackbar
 - Jeder Angriff verursacht Kosten beim Angreifer

Hashfunktionen

- Abbildung $h: X \rightarrow Y$
 - Einwegfunktion (auch: Falltürfunktion)
 - Umkehrfunktion nicht effizient berechenbar
- Hashfunktionen sind verkürzend:
 - Beliebige lange Inputs werden auf Output bestimmter Länge abgebildet, z.B. SHA1: 160 Bit (10 Blöcke zu je 16 Bit in Hex)

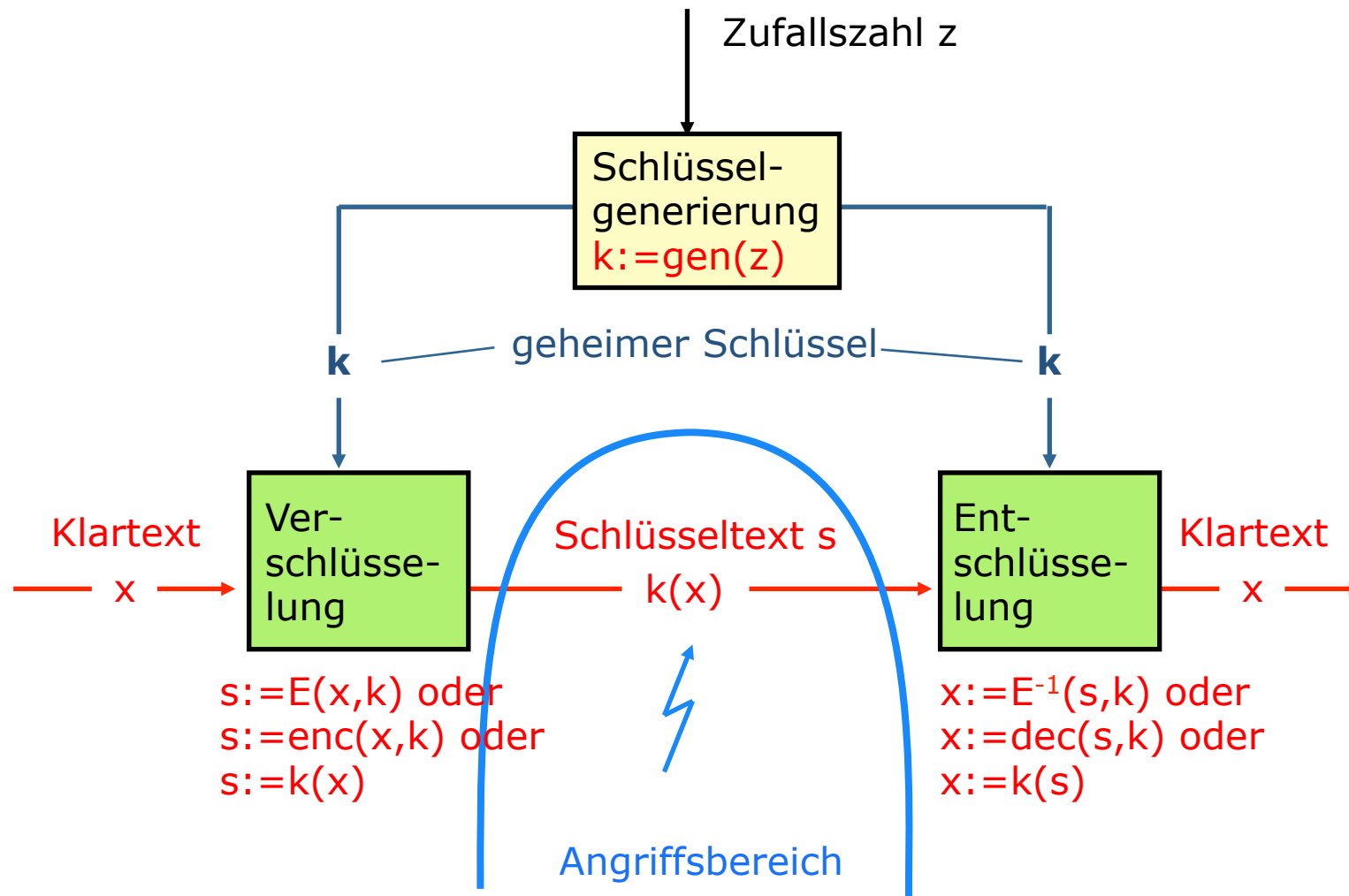
17EC 1A95 14E4 F581 7C68 2AC1 0939 D2CA 9879 FFBF

- Kollision:
 - $h(x1) = h(x2)$ mit $x1 \neq x2$



- Kryptographische Hashfunktionen sind kollisionsresistent:
 - nicht mit vertretbarem Aufwand möglich, eine Kollision gezielt herbeizuführen, z.B. Finden eines $x2$ zu einem gegebenen $h(x1)$

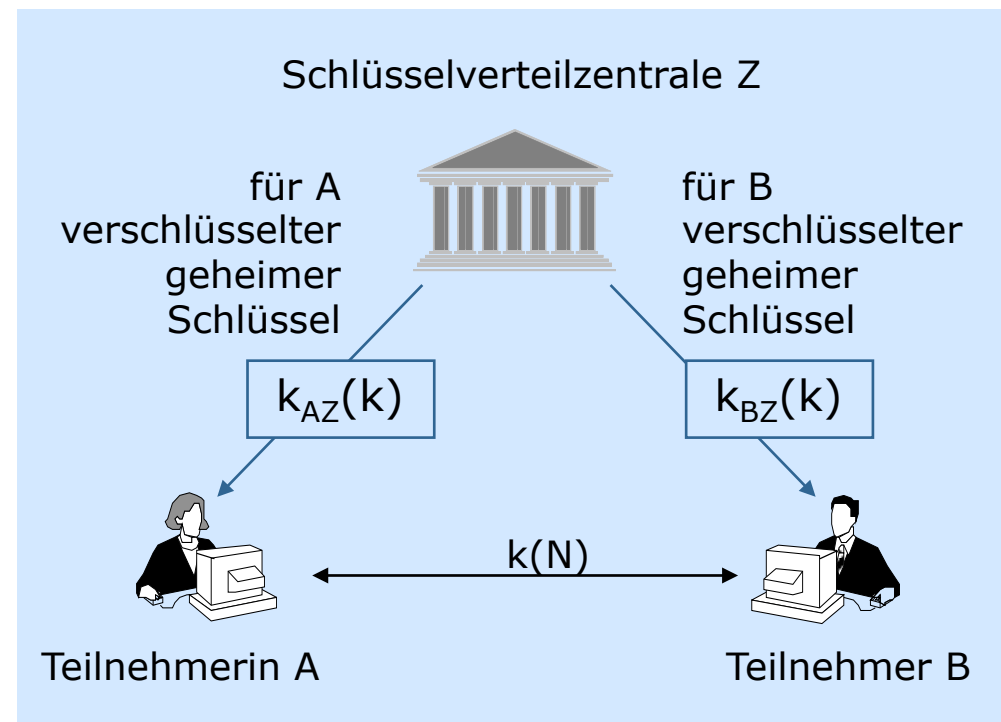
Symmetrische Verschlüsselung



»Undurchsichtiger Kasten mit Schloss. Es gibt zwei gleiche Schlüssel.«

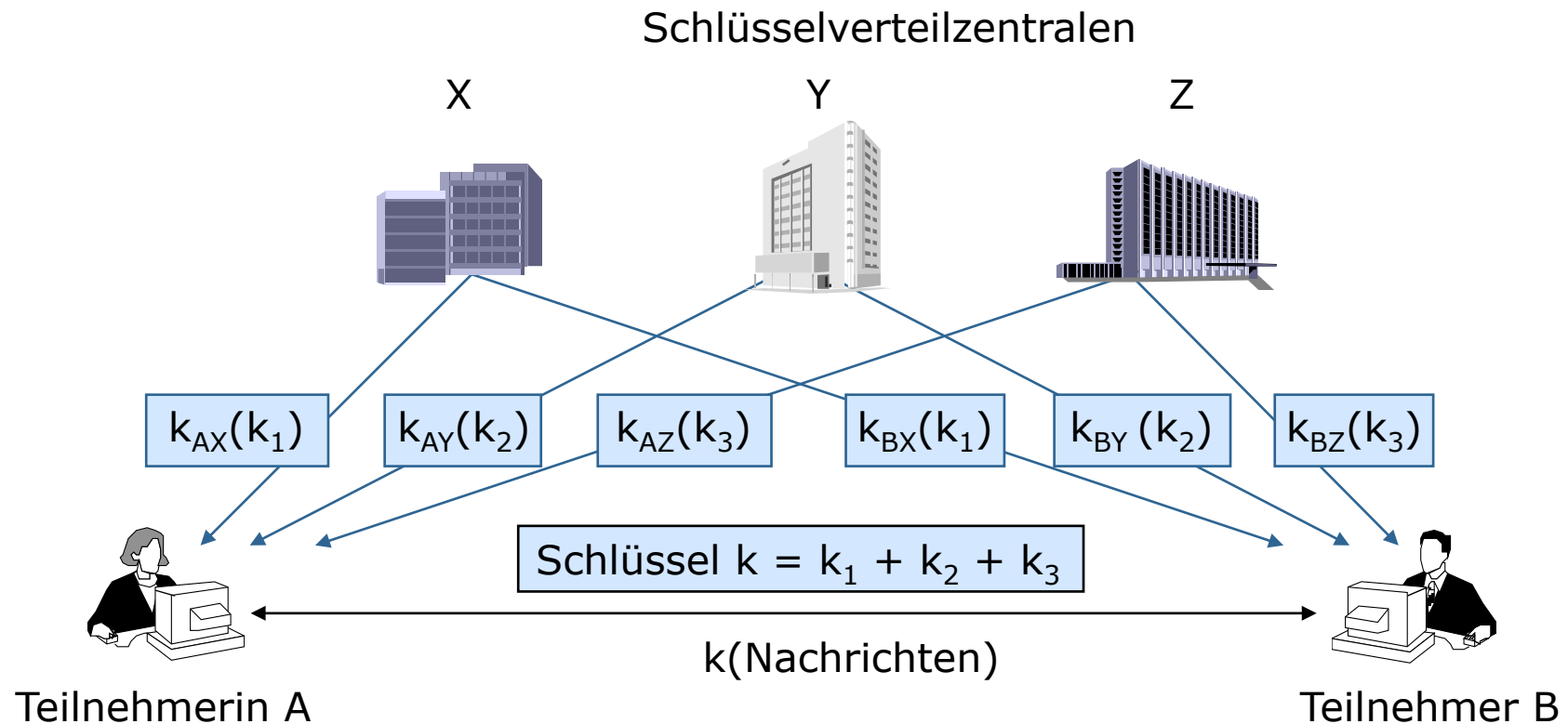
Schlüsselverteilung für symmetrische Systeme

- Schlüsselaustausch:
 - A und B tauschen zunächst (offline) jeweils symmetrischen Schlüssel mit Z aus:
 - K_{AZ} und K_{BZ}
 - Z generiert auf Anforderung einen symmetrischen Kommunikationsschlüssel k und verschlüsselt diesen für A und B:
 - $K_{AZ}(k) \rightarrow A$
 - $K_{BZ}(k) \rightarrow B$
 - A und B entschlüsseln k
- Kommunikation:
 - Sender verschlüsselt Nachricht N mit k :
 - $k(N)$

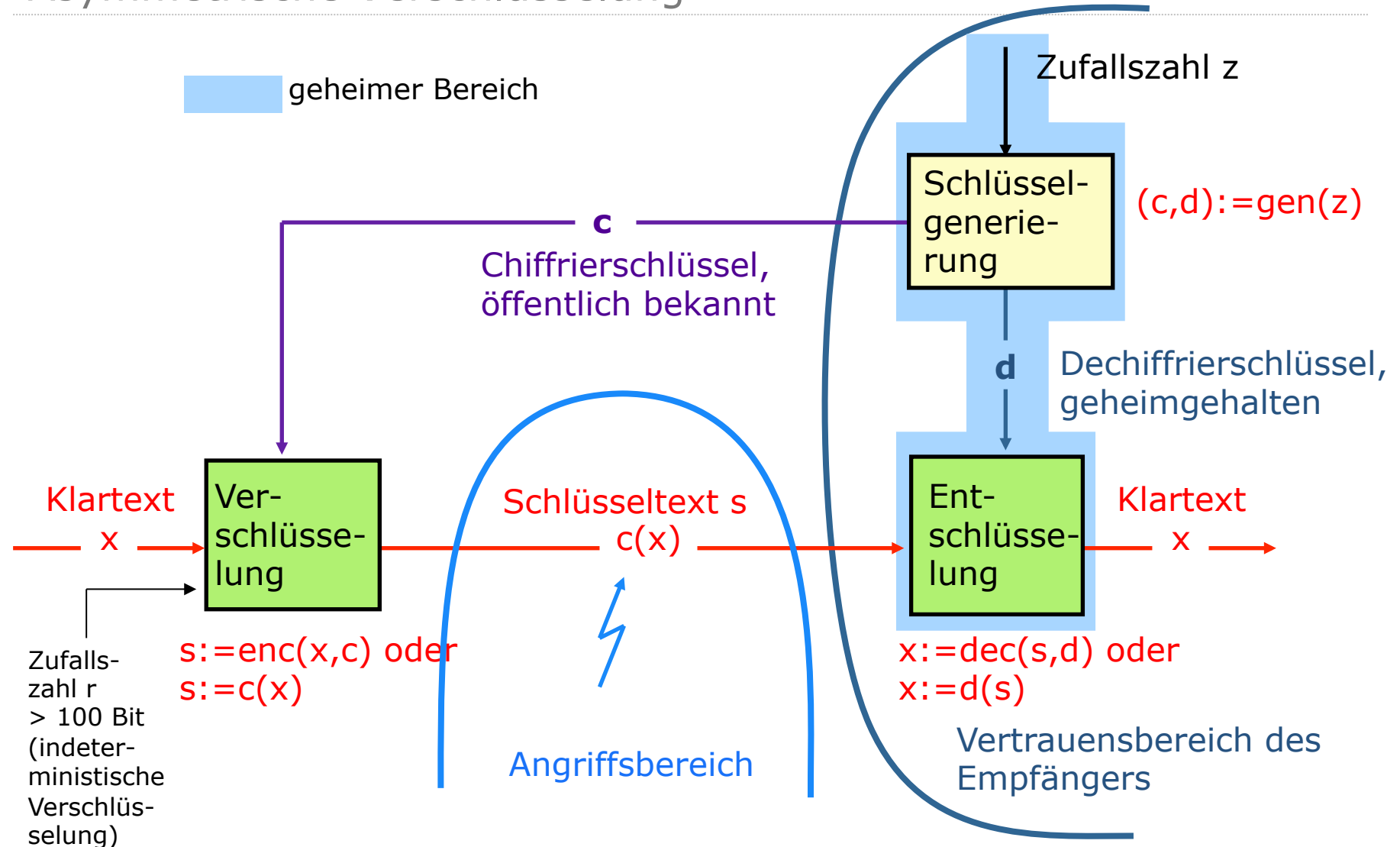


Dezentralisierte Variante

- Dezentralisierte Schlüsselverteilung ist möglich
- Ziel: Alle beteiligten Schlüsselverteilzentralen müssen zusammen arbeiten, damit sie den Kommunikationsschlüssel k erfahren
- Überlagerung der Teilschlüssel z.B. mit XOR-Verknüpfung

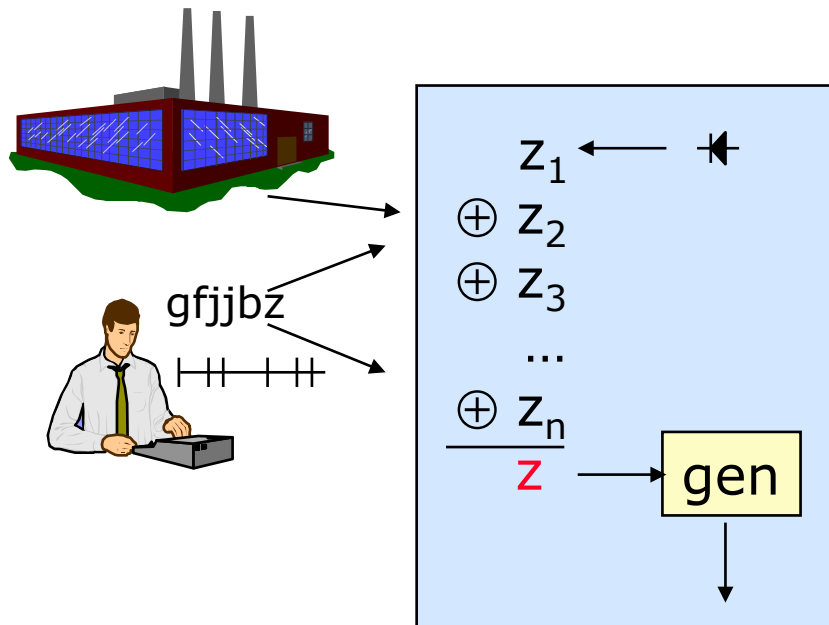


Asymmetrische Verschlüsselung



»Kasten mit Schnappschloss. Es gibt nur einen Schlüssel.«

Schlüsselgenerierung

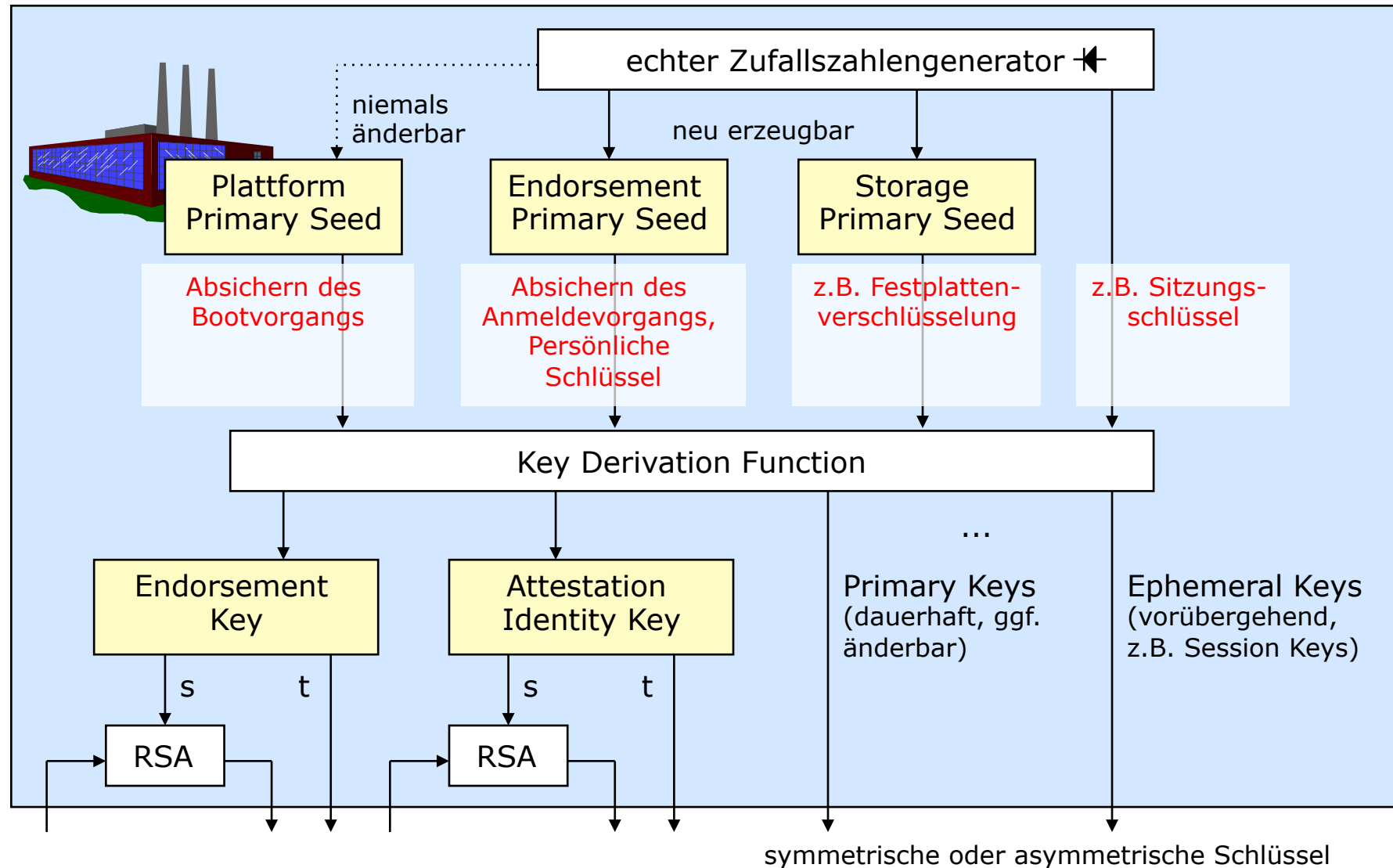


Erzeugung einer Zufallszahl z für die Schlüsselgenerierung:

XOR aus

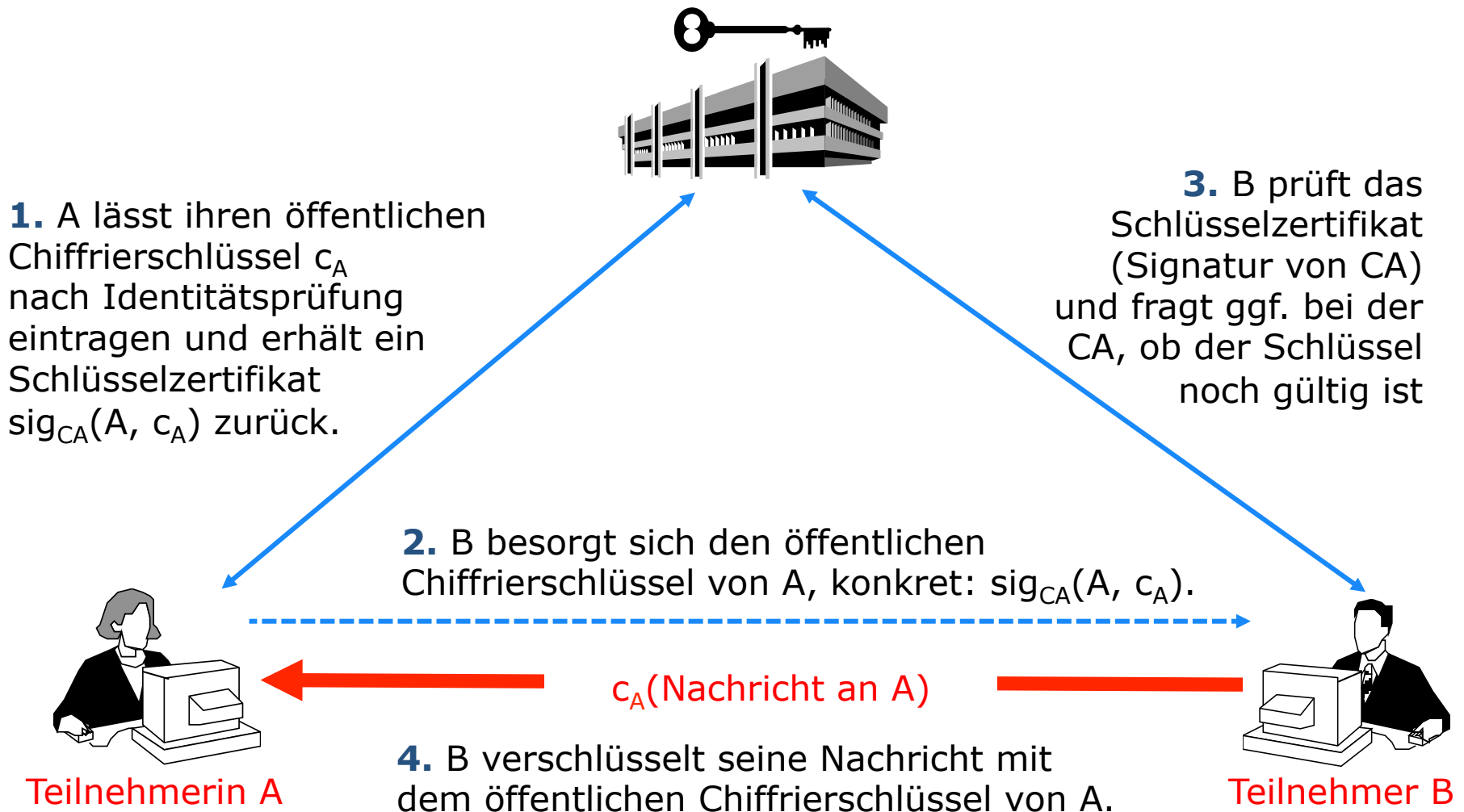
- z_1 , einer im Gerät erzeugten,
- z_2 , einer vom Hersteller gelieferten,
- z_3 , einer vom Benutzer gelieferten,
- z_n , einer aus Zeitabständen errechneten.

Beispiel: Trusted Platform Module (TPM) 2.0



Zertifizierung des öffentlichen Schlüssels

Zertifizierungsstelle (Certification Authority) CA

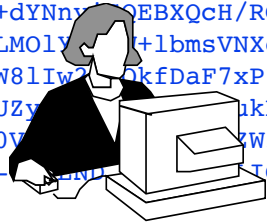


Maskerade-Angriff 1/2

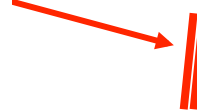
Alice hat Schlüsselpaar generiert und will ihn veröffentlichen.

Alice <alice@abc.de>

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
mQGIBDQyJk0RBADVPjcdvmyOtqsZBt6z4/5M9MYDB
i+dYNny1QEBXQcH/RGe2i30LRvRk4asX++JSTylku
8LM0LY...+lbmsVNxeQsdbSAUfd3d9bI/+fGwQcz
6W81Iw2...QkfDaF7xPI7oVZUY1I7cqEfTvic003bgL
sUZy...Kj01066wVmqlnXcbi2XUebka
L0V...W59gf5I0eUBevSmydIaliH9Pm
-----END PGP PUBLIC KEY BLOCK-----
```



C_{Alice}



Angreifer

- hält C_{Alice} zurück (blockiert Verteilung)
- generiert selbst ein Schlüsselpaar $C_{\text{Mask}}, d_{\text{Mask}}$ unter falschem Namen
- schickt C_{Mask} an Bert

C_{Mask}



Bert besitzt jetzt nicht authentischen Schlüssel von Alice.

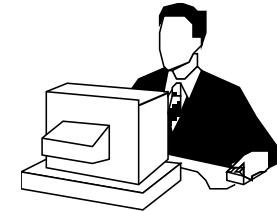
Alice <alice@abc.de>

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
OTUAoLncfli6Yit0Kqgp/N9h37uopJHbiQCVAw
xBBPLRdmlP22ij0dARxbJL07u7XOrnyV3b4
l4ydps/ruj9yaY62BwQNMEoGjAnZGA5t3MMO
7ZLpldmFYVYVPL4xRfOJ+MF5ifb8PX+DA1
CwMBAgAKCRDhQCBhSe8dhOYAJSEI
u64hbO2wuFQlwwqlyb+JAD8DBRA0
-----END PGP PUBLIC KEY BLOCK-----
```



Maskerade-Angriff 2/2

Bert will Alice eine Nachricht N schicken.



$c_{\text{Mask}}(N)$

Angreifer:

- Weiterleitung verhindern
- entschlüsseln von $c_{\text{Mask}}(N)$ mit d_{Mask}
- verschlüsseln von N mit c_{Alice}

$c_{\text{Alice}}(N)$

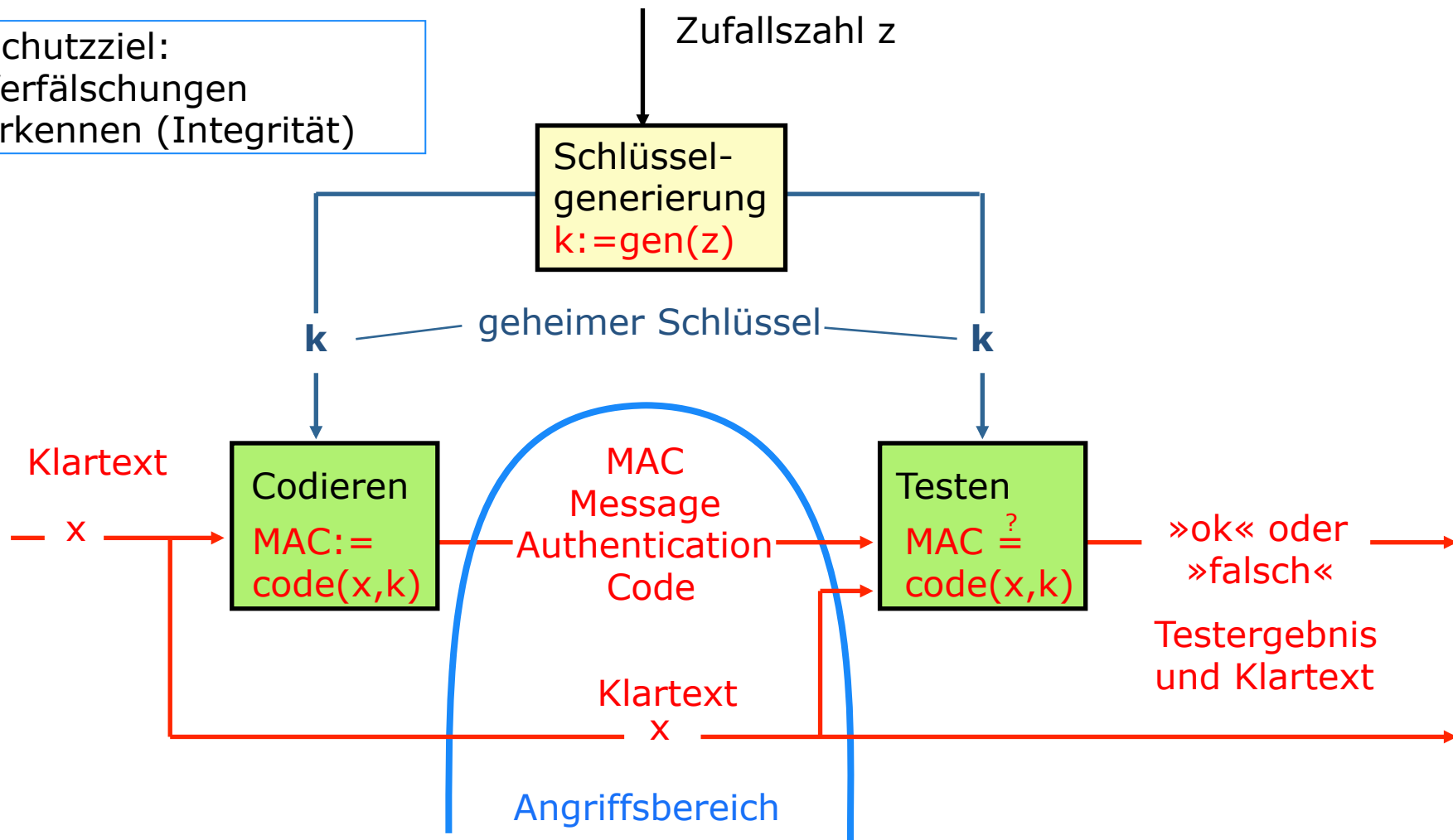
Alice erhält die Nachricht N .
 N ist verschlüsselt mit ihrem öffentlichen Schlüssel.



Ohne die Gewissheit über die Echtheit eines öffentlichen Schlüssels funktioniert keine sichere asymmetrische Kryptographie. Deshalb: Schlüsselzertifizierung

Symmetrische Authentikation

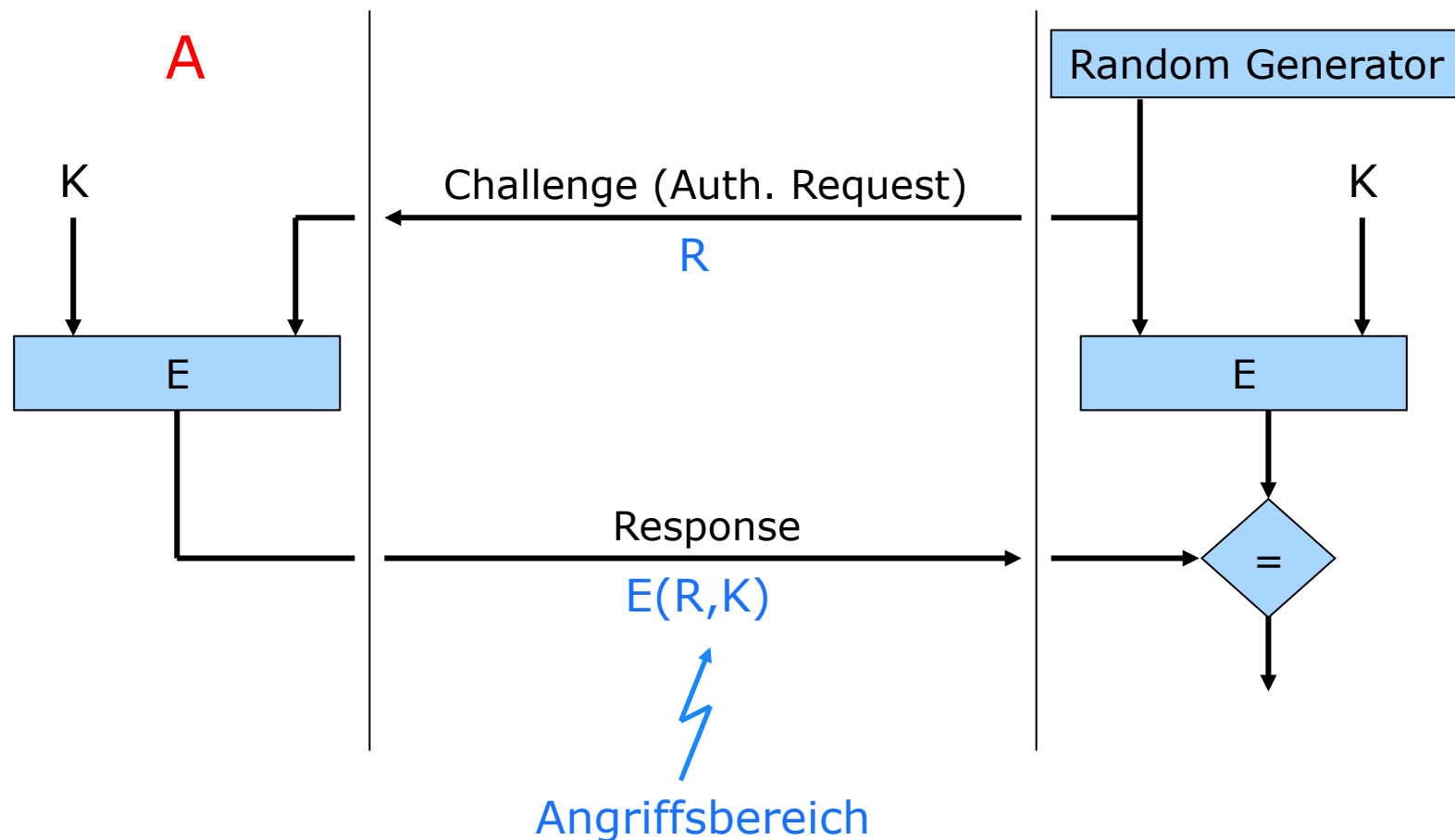
Schutzziel:
Verfälschungen
erkennen (Integrität)



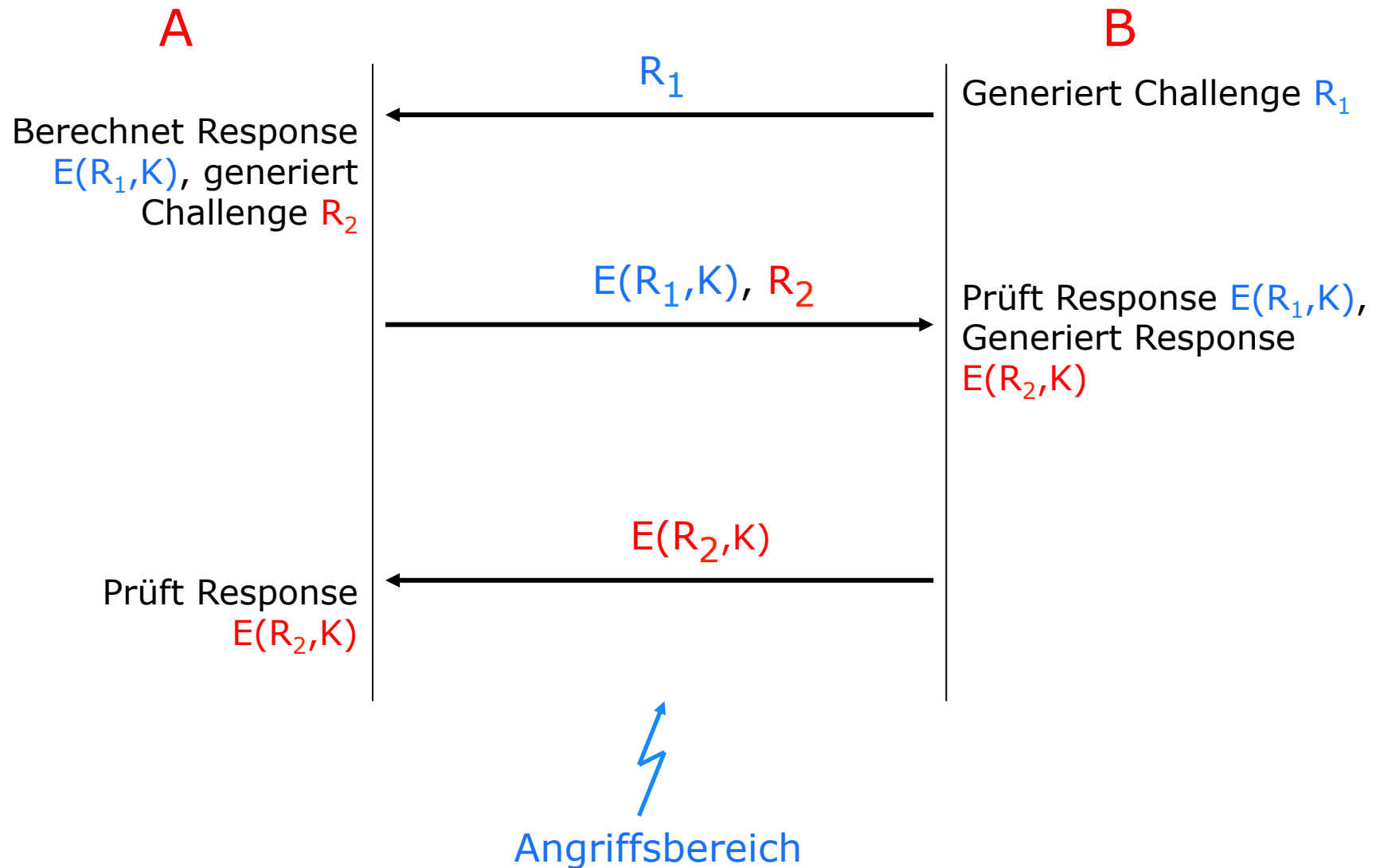
»Glasvitrine mit Schloss. Es gibt zwei gleiche Schlüssel.«

Challenge-Response-Authentikation

- Frage-Antwort-Verfahren
 - meist basierend auf symmetrischem Authentikationssystem
 - A soll sich vor B authentisieren

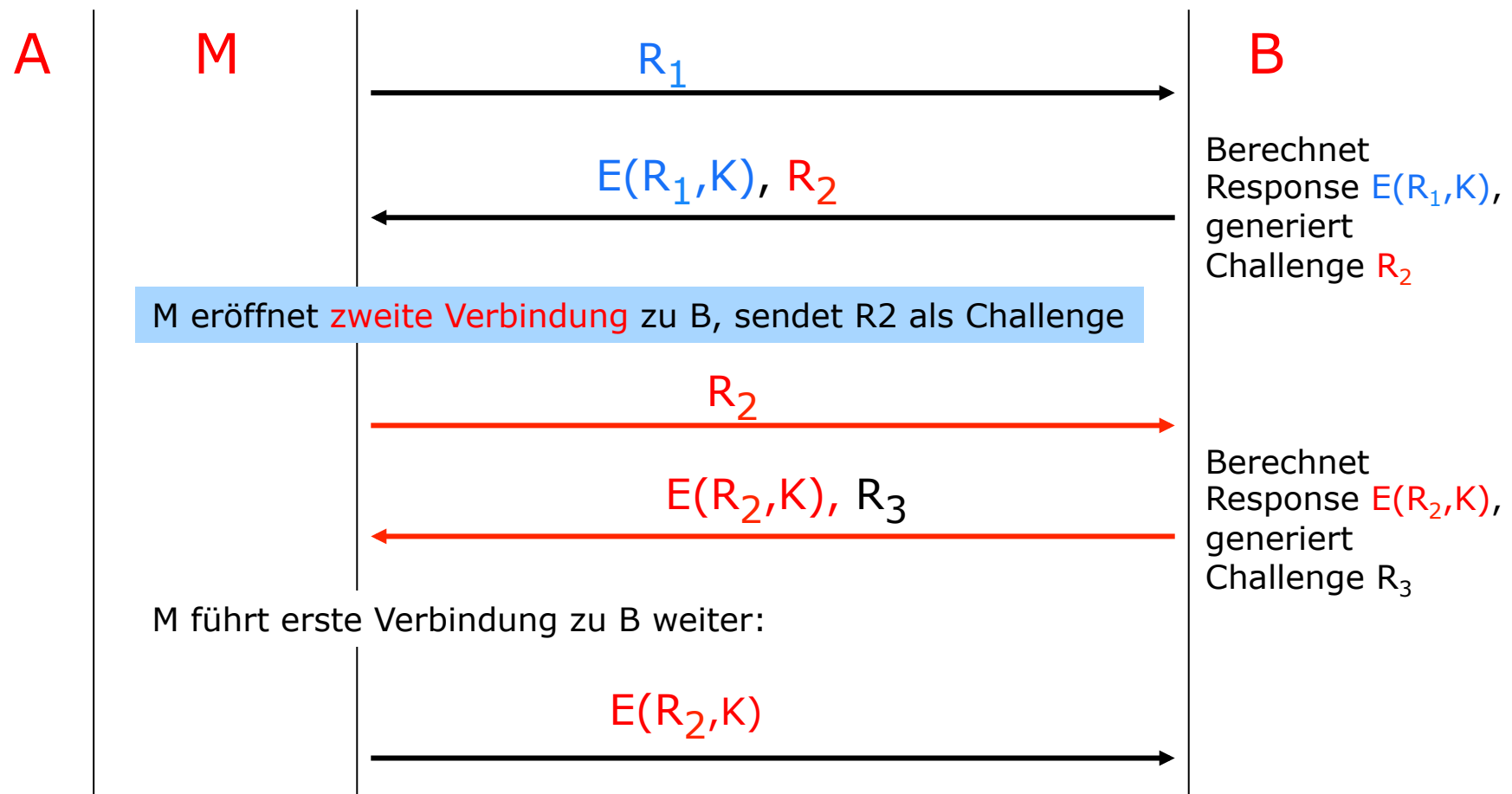


Gegenseitige Authentikation



Gegenseitige Authentikation

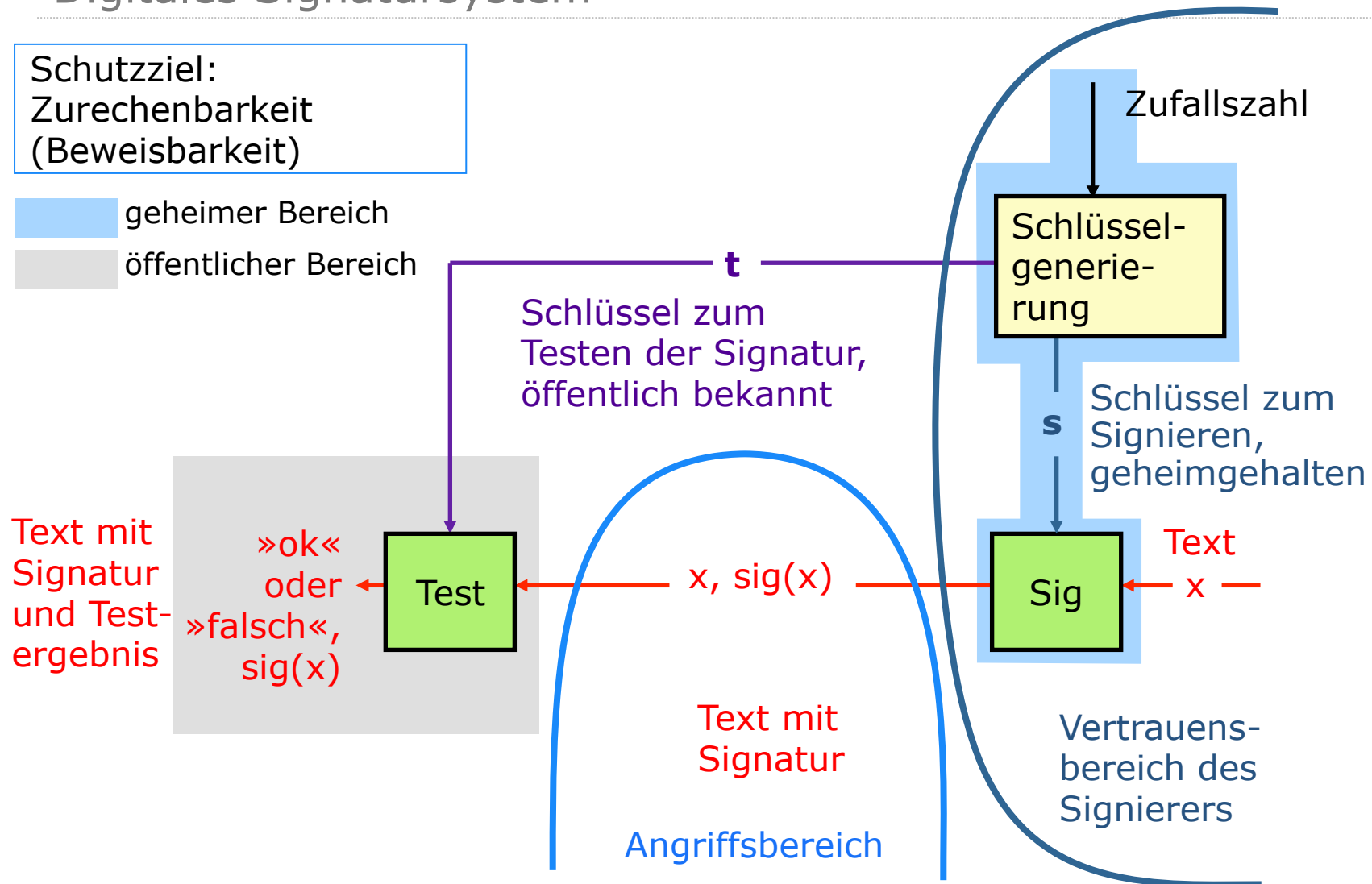
- Aktiver Angriff auf gegenseitige Authentikation auf der Basis symmetrischer Kryptosysteme
 - Angreifer **M** maskiert sich als **A**, kennt **K** *nicht*



Digitales Signatursystem

Schutzziel:
Zurechenbarkeit
(Beweisbarkeit)

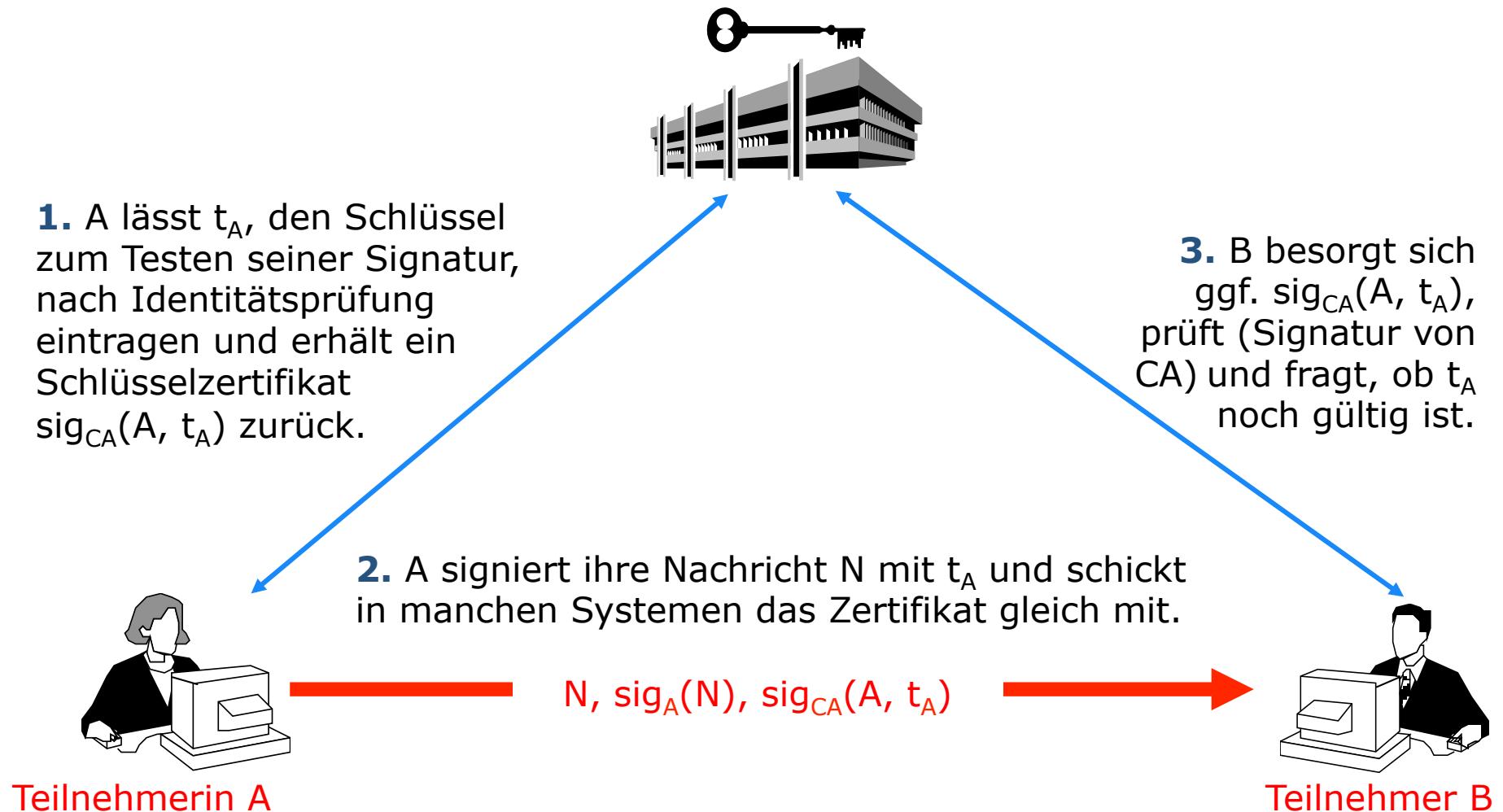
geheimer Bereich
öffentlicher Bereich



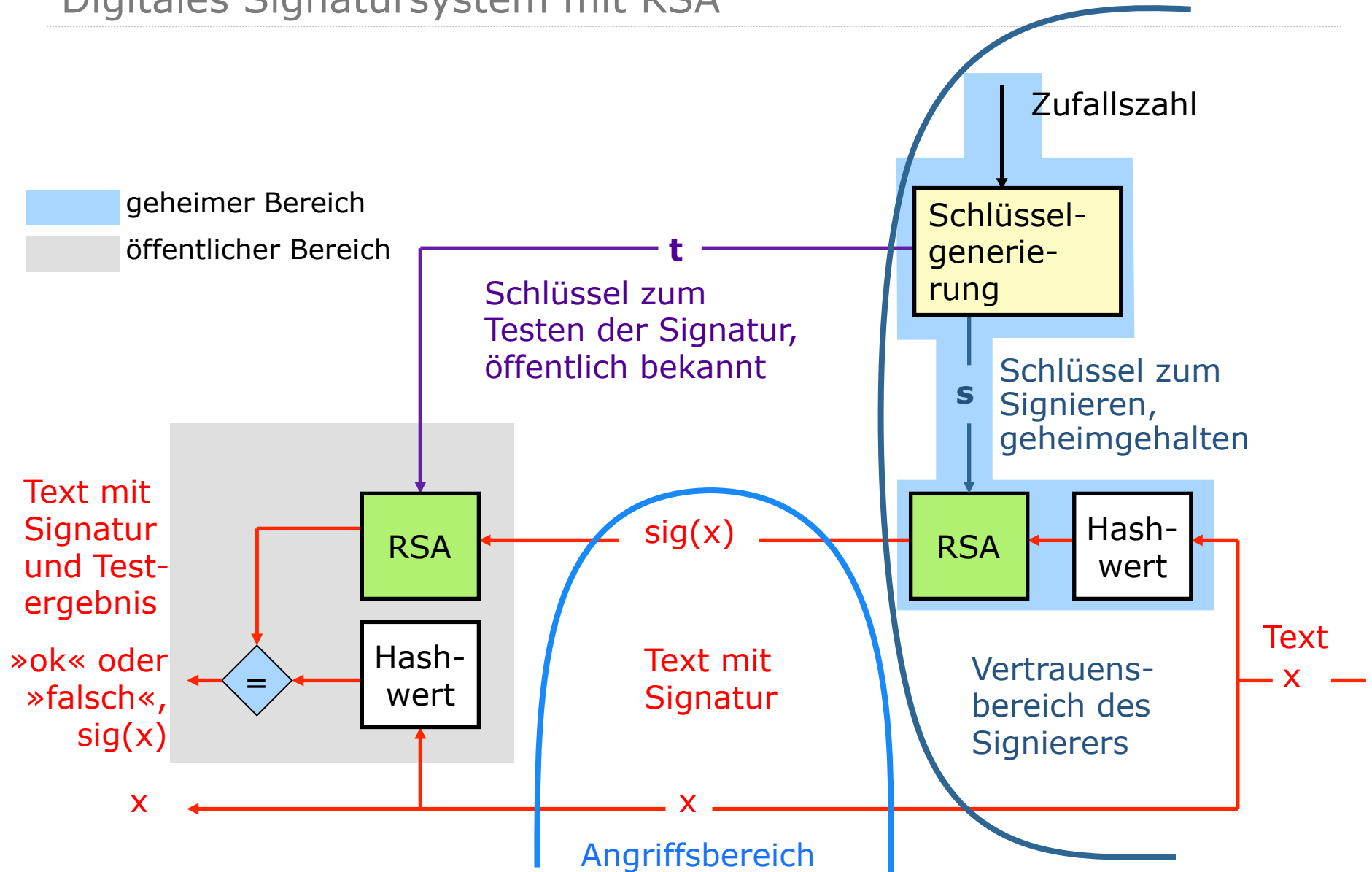
»Glasvitrine mit Schloss. Es gibt nur einen Schlüssel.«

Zertifizierung des öffentlichen Testschlüssels

Zertifizierungsstelle (Certification Authority) CA



Digitales Signatursystem mit RSA



Schlüssellängen

- **Beispielrechnung:**
 - 56 Bit (DES) sind heute unsicher.
 - 56 Bit Schlüssellänge → 2^{56} mögliche Schlüssel (ca. $7 \cdot 10^{16}$)
 - Ausprobieren eines Schlüssels dauere 1 Nanosekunde (10^{-9} s)
 - Ausprobieren aller Schlüssel dauert dann:
 $2^{56} \cdot 10^{-9} \text{ s} = 72057594 \text{ s} = 2,28 \text{ Jahre}$
- **Symmetrische Systeme:**
 - Vergrößerung des Schlüssels um 1 Bit bedeutet Verdoppelung des Schlüsselraumes
 - Schlüssellängen: 128–256-Bit auf »absehbare Zeit« sicher
 - jeder Schlüssel aus Sicht des Angreifers gleichwahrscheinlich
- **Asymmetrische Systeme:**
 - meist Vergrößerung des Zahlenbereichs nötig, da nur bestimmte Zahlen (z.B. Primzahlen) Schlüssel sein können
 - Schlüssellängen: 2048-4096 Bit, elliptische Kurven: ca. 250 Bit

Welche Schlüssellängen und Kryptoalgorithmen sind sicher?

Jährlicher Algorithmenkatalog nach § 17 (1) SigG des Bundesamts für die Sicherheit in der Informationstechnik (BSI)

Tabelle 3: Geeignete Schlüssellängen für DSA

Parameter \ Zeitraum	bis Ende 2015	bis Ende 2021
p	2048	2048
q	224	256

Tabelle 8: Nicht mehr geeignete RSA-Schlüssellängen

Modullänge n	geeignet bis
768	Ende 2000
1024	Ende März 2008*
1280	Ende 2008
1536	Ende 2009
1728	Ende 2010

* Januar – März 2008: Übergangsfrist

Bundesnetzagentur für Elektrizität, Gas, Telekommunikation, Post und Eisenbahnen

Bekanntmachung zur elektronischen Signatur nach dem Signaturgesetz und der Signaturverordnung

(Übersicht über geeignete Algorithmen)

Vom 15. 12. 2014

Die Bundesnetzagentur für Elektrizität, Gas, Telekommunikation, Post und Eisenbahnen als zuständige Behörde gemäß § 3 Signaturgesetz (SigG) vom 16. Mai 2001 (BGBl. I S. 876), zuletzt geändert durch Artikel 4 des Gesetzes vom 17. Juli 2009 (BGBl. I S. 2091), veröffentlicht gemäß Anlage 1 Abschnitt 1 Nr. 2 Signaturverordnung (SigV) vom 16. November 2001 (BGBl. I S. 3074), zuletzt geändert durch die Verordnung vom 15. November 2010 (BGBl. I S. 1542), im Bundesanzeiger eine Übersicht über die Algorithmen und zugehörigen Parameter, die zur Erzeugung von Signaturschlüsseln, zum Hashen zu signierender Daten oder zur Erzeugung und Prüfung qualifizierter elektronischer Signaturen als geeignet anzusehen sind, sowie den Zeitpunkt, bis zu dem die Eignung jeweils gilt.

Geeignete Algorithmen zur Erfüllung der Anforderungen nach § 17 Abs. 1 bis 3 SigG vom 16. Mai 2001 in Verbindung mit Anlage 1 Abschnitt I Nr. 2 SigV vom 16. November 2001

Vorbemerkung: Wie in den Vorjahren werden im Folgenden geeignete Algorithmen und Schlüssellängen für den Zeitraum der kommenden sieben Jahre anstatt des in der SigV vorgesehenen Mindestzeitraums von sechs Jahren aufgeführt. Das heißt konkret, dass geeignete Algorithmen und Schlüssellängen bis Ende 2021 statt bis Ende 2020 aufgeführt sind. Im Allgemeinen sind solche längerfristigen Prognosen schwer möglich. Die vorliegende Übersicht über geeignete Algorithmen unterscheidet sich von der zuletzt veröffentlichten Übersicht vom 20. Februar 2014 (BAnz AT 20.02.2013 B4) im Wesentlichen in folgenden Punkten:

1. Die Eignung von Nyberg-Rueppel-Signaturen wird nicht über das Jahr 2020 hinaus verlängert. Dies hat keine Sicherheitsgründe, sondern dient der Vereinfachung der Pflege des Algorithmenkatalogs. Nachdem die Streichung von Nyberg-Rueppel-Signaturen in den letzten beiden Versionen der vorliegenden Bekanntmachung angekündigt und in den entsprechenden Expertenanhörungen diskutiert wurde, sind bei den zuständigen Stellen im Bundesamt für Sicherheit in der Informationstechnik und in der Bundesnetzagentur keine Einsprüche gegen die Streichung dieses Verfahrens eingegangen. Es wird daher davon ausgegangen, dass es keine praktische Verwendung findet im Bereich der qualifizierten elektronischen Signatur.
2. Wie bereits im vorigen Algorithmenkatalog angekündigt wurde, wird die Eignung von Zufallsgeneratoren, die entsprechend der Funktionalitätsklassen nach [31] zertifiziert wurden, von wenigen Ausnahmefällen abgesehen nicht über das Jahr 2020 hinaus verlängert.

<https://www.bsi.bund.de/Algorithmenkatalog>

Vollständiges Durchsuchen (brute-force, exhaustive search)

- Angriff über Supercomputer und künftig Quantencomputer
 - betrifft nur komplexitätstheoretisch sichere Systeme
- Schutz gegen Supercomputer
 - Schlüssel ausreichend lang wählen
- Schutz gegen Quantencomputer
 - symmetrisch: Schlüssellänge verdoppeln auf mind. 256 Bit
 - asymmetrisch: [post-quantum cryptography]

	Key lengths	Complexity		
		Super Computer	Quantum Computer	
Symm.	128 Bit	2^{127}	2^{64}	Grover, 1996
	256 Bit	2^{255}	2^{128}	
Asymm.	1024 Bit	$\approx 2^{90}$	$\approx 2^{25}$	Shor, 1994
	2048 Bit	$\approx 2^{117}$	$\approx 2^{28}$	

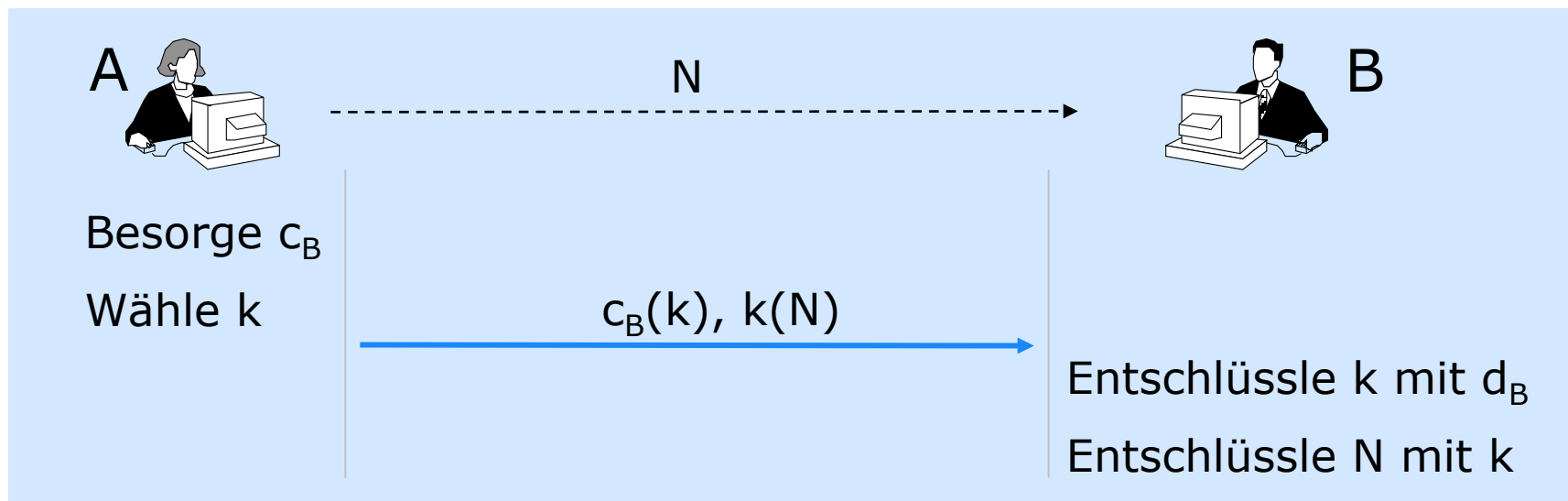
nach: Bernstein, Buchmann, Dahmen: Post Quantum Cryptography. Springer, 2009

Vergleich: symmetrische-asymmetrische Systeme

- Wieviele Schlüssel müssen bei n Teilnehmern ausgetauscht werden?
 - symmetrische Systeme:
 - asymmetrische Systeme:
- Typische Schlüssellängen: (bei vergleichbarem Sicherheitsniveau)
 - symmetrische Systeme: 128–256 Bit
 - asymmetrische Systeme: 2048–4096 Bit
Elliptische Kurven: ca. 250 Bit
- Performance:
 - symmetrische Systeme ver- bzw. entschlüsseln etwa um den Faktor 100–10.000 schneller
- Asymmetrische Systeme: Geringere Effizienz und größere Schlüssellängen werden aufgewogen durch den stark vereinfachten Schlüsselaustausch

Hybride Kryptosysteme

- **Kombiniere**
 - einfachen Schlüsselaustausch der asymmetrischen Systeme
 - hohe Verschlüsselungsleistung der symmetrischen Systeme
- **Verfahren**
 - Asymmetrisches Kryptosystem wird zum Austausch eines symmetrischen Sitzungsschlüssels k (session key) **verwendet**.
 - Eigentliche Nachricht N wird mit k verschlüsselt.
- Nur sinnvoll, wenn N deutlich länger als wenige Bit ist.



Pretty Good Privacy (PGP) und Gnu Privacy Guard (GnuPG)

-----BEGIN PGP MESSAGE-----

Version: GnuPG v1.4.8 (Darwin)

hQIOA2ThYngSETJKEAgA4I9+HpuDVc95Sip7xgHXQXooxpEQZb7xaAV84XhSly48
wDDhe4Dk9kwqKlYZJgw5Df/pW9MzhJHilQ9jgU90AE5t5IkmcB+EJSor1YqxUjMZ
Q7baKGRNBQhVNP/+5i3K2GuuXVeYaccYfGvF4mSANremmbHeH0L9j6cSrGCsqQqa
b4ASOc+2ov6UU5PRJX+gXzkUkm2Gz/HPpkPfr2QS90CjpdQnyNpHCHAPmMKSIZU
WjuTZfGOOGtvYpMqCFn3cv+6zeCpPXGNDk0W/VYNQ877Irykn3XLuKrAULQYkFwU
Vaml6/s2jlufdPLuTF9g3i0xeQuJnv5pKv0DcTxwPR0mLBOKJUS6DDUq1lY6rviO
l7km72jIz83wl6PZAfWmzj4IyZq8ktPcZ/fdnrZ50FE34Vfwzvh0lbRqdeFY6GPW
f6Y3FnF9DJUkm1kYuAp65X6E19fapJdAnTvjb2WV9XWrmPypJIcF5kTXL8vOLCtu
yZ6R+PS0q6c=
=x491

-----END PGP MESSAGE-----

<http://www.pgpi.net>
<http://www.gnupg.org>

-----BEGIN PGP SIGNED MESSAGE-----

Hash: SHA1

Deutlicher jedoch nähert sich das Präludium g-moll der Toccata mit einem zwischen
rahmende Pfeiler gestellten, ausgedehnten improvisatorischen Mittelteil, in dessen
figurativer Sequenzierung Bach mit einer über eine Dezime chromatisch absteigenden
Skala die elementare Farbigkeit der enharmonischen Umdeutungen entdeckte.

-----BEGIN PGP SIGNATURE-----

Version: GnuPG v1.4.8 (Darwin)

iEYEARECAAYFAkj9yQACgkQ4UAgYUnvHYSahQCfaWrrH1l9s4tXeFToa6aQPryw
TX4AoL7l7WQHXPzxVG6SX9fSOAskCzn
=Ebit

-----END PGP SIGNATURE-----


Key Recovery und Key Escrow

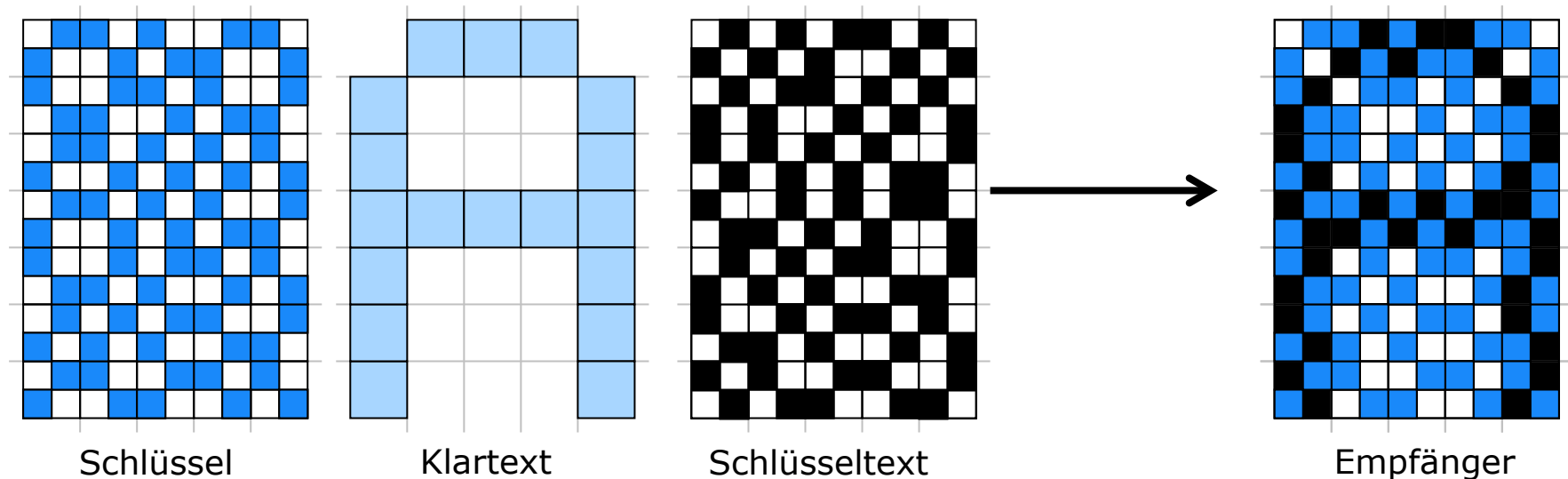
- **Key Recovery**
 - Hinterlegung des Entschlüsselungsschlüssels zum Zweck der Entschlüsselbarkeit bei Schlüsselverlust.
 - Schwellwertschema: Schlüssel wird in $n+k$ Teile zerlegt. Zur Rekonstruktion werden wenigstens n Teile benötigt.
- **Key Escrow**
 - Hinterlegung des Entschlüsselungsschlüssels zum Zweck der Strafverfolgung.
 - so dass alle Nachrichten ab einem bestimmten Zeitpunkt entschlüsselt werden können
 - so dass Nachrichten auch rückwirkend entschlüsselt werden können
- **Beachte**
 - Signaturschlüssel müssen nie hinterlegt werden, da eine Signatur stets testbar bleibt.
 - Bei Verlust des Signierschlüssels: neuen erzeugen.

Key Recovery

	Schutz der Kommunikation	Langfristige Speicherung
Verschlüsselung	Key Recovery	Key Recovery
Authentifikation	für Funktion unnötig, aber	sinnvoll
	asymmetrisch (dig. Signatur)	
	zusätzliches Sicherheitsrisiko	

Visuelle Kryptographie

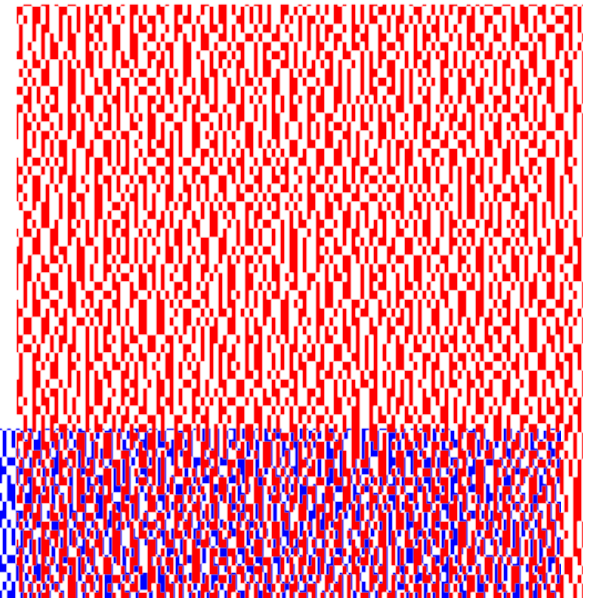
- Symmetrisches Verfahren
 - Symmetrischer Schlüssel: Sender und Empfänger erzeugen sich Zufallsmuster aus zwei »Basismustern«: 
- Visuelle Botschaft:
 - Sender verwendet negiertes Muster für schwarze Bildpunkte
 - Für »weiße« Bildpunkte: keine Veränderung



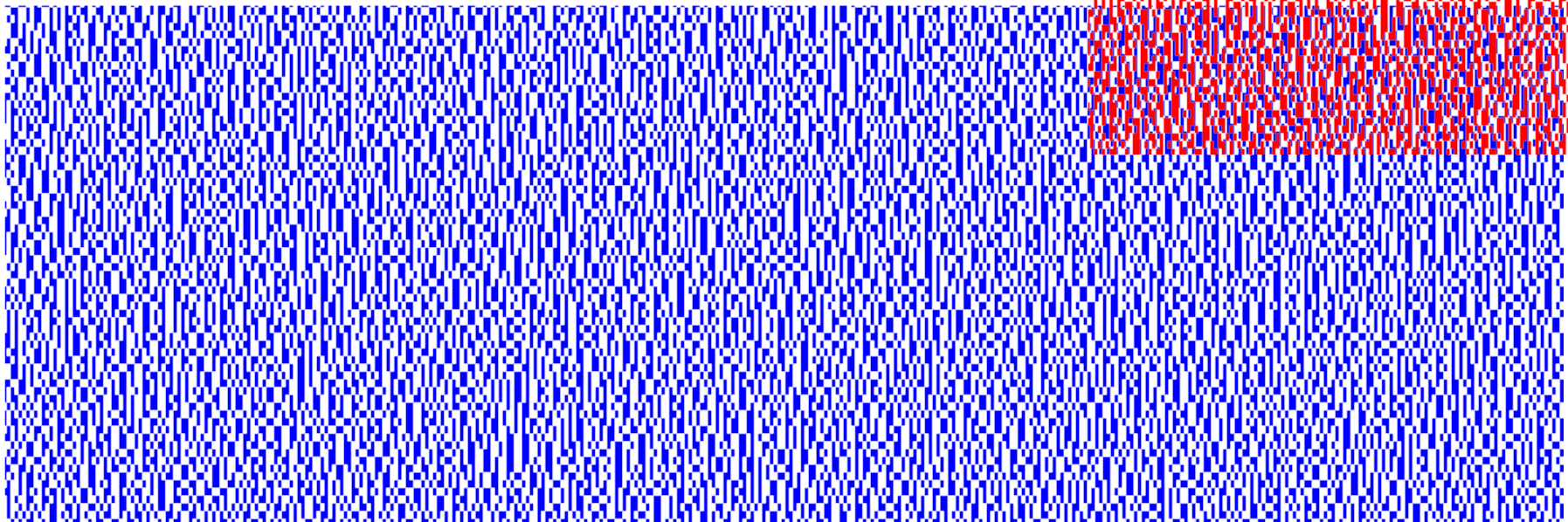
Visuelle Kryptographie: Demo



Schlüssel



Botschaft

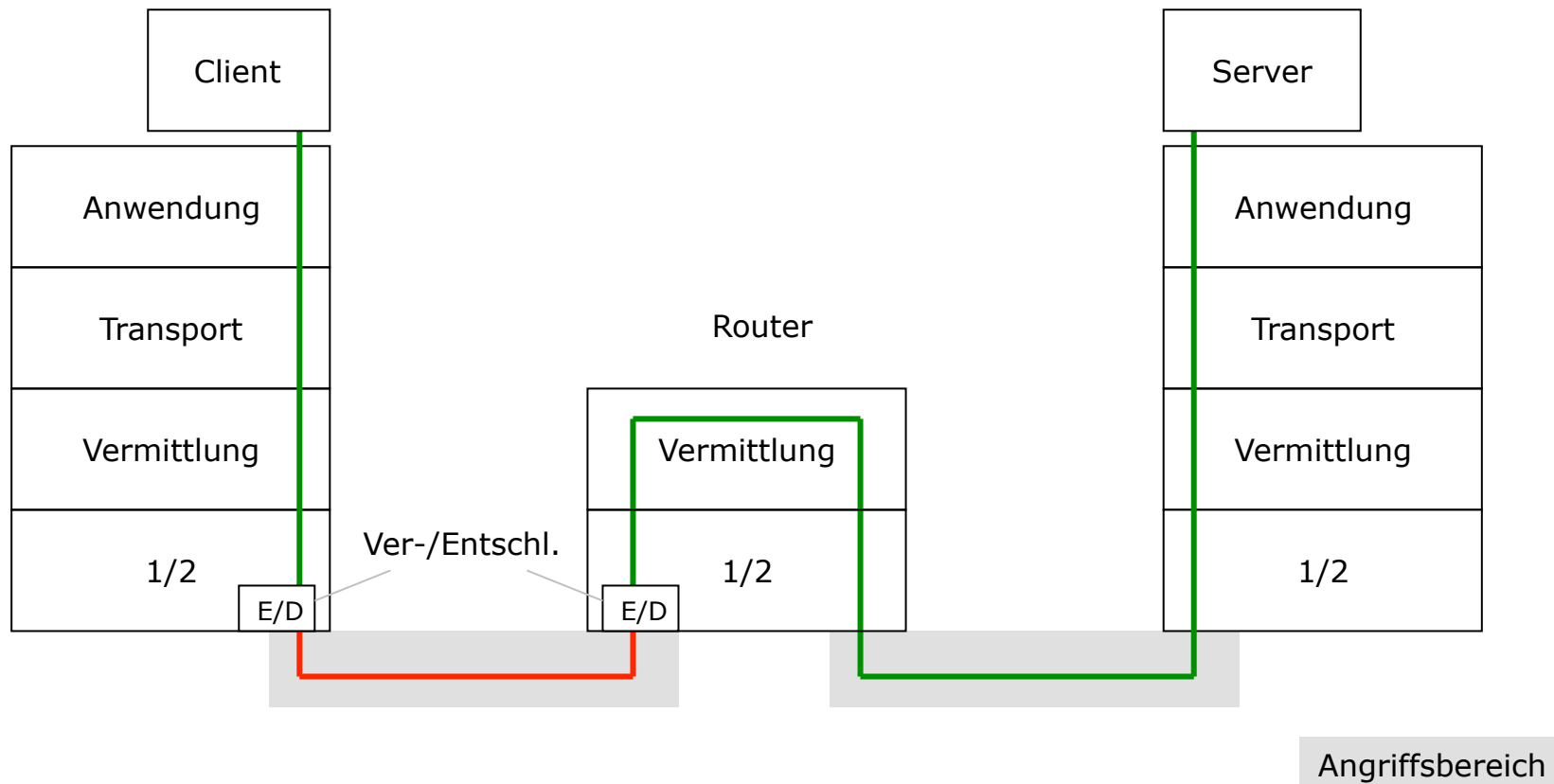


Sicherheitsfunktionen nach Schichten geordnet

Kommunikations- schicht im OSI- Referenzmodell	Sicherheitsfunktion
Anwendungsschicht	Pretty Good Privacy (PGP), S/MIME (Secure Multipurpose Internet Mail Extensions), Secure Shell (SSH)
Transportschicht	Secure Sockets Layer/Transport Layer Security (SSL/TLS)
Vermittlungsschicht	Authentication Header (AH) zur Integritätssicherung von Datagrammen, Encapsulated Security Payload (ESP) zur Verschlüsselung von Datagrammen
Schichten 1/2	Challenge Handshake Protocol (CHAP, Passwort), Encrypt Control Protocol (ECP), WiFi Protected Access (WPA) 2

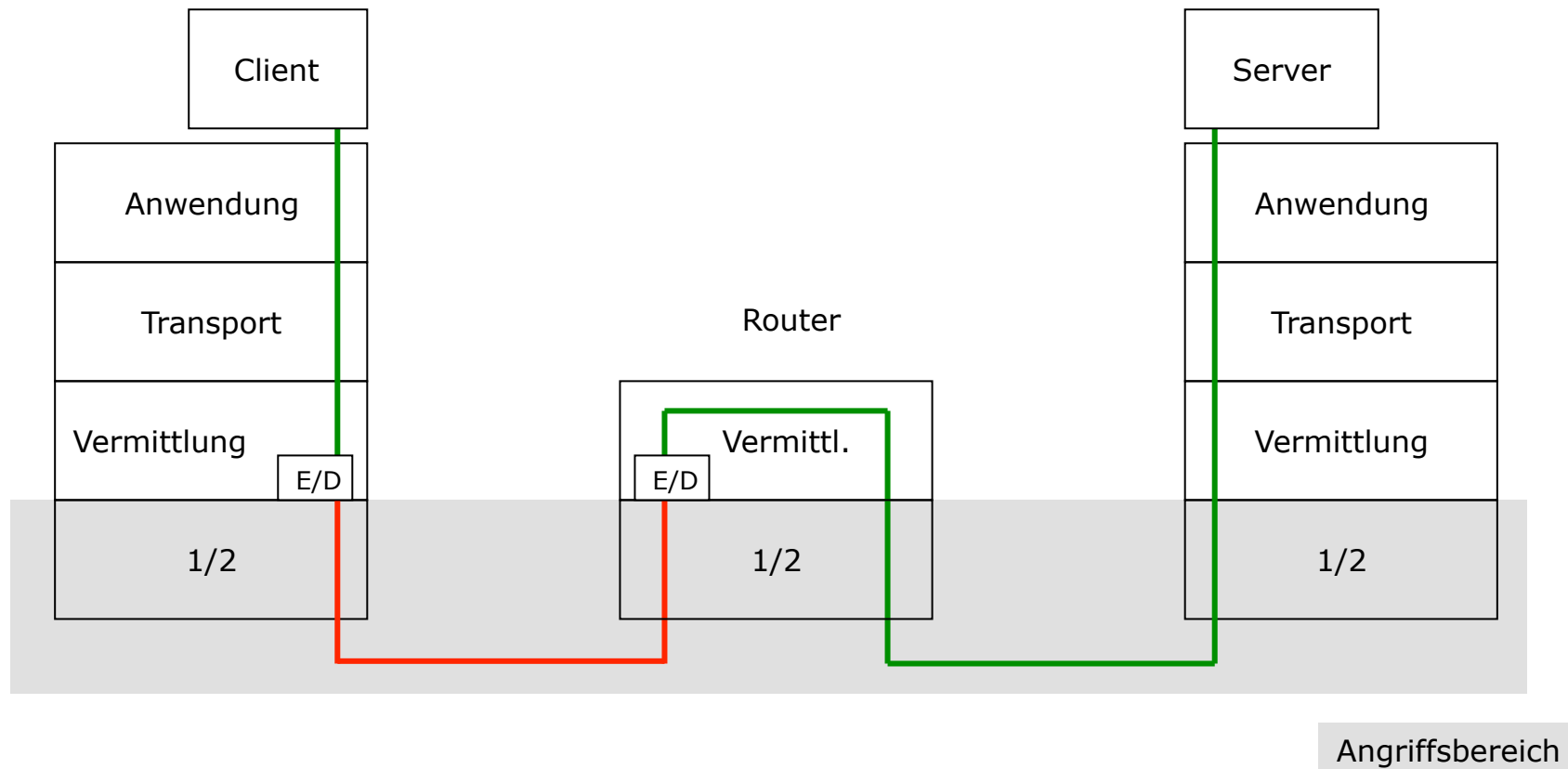
Verschlüsselung in Schicht 1/2

- Verschlüsselung nur bis zum nächsten Router (Verbindungsverschlüsselung)
 - Nicht alle Teilstrecken müssen verschlüsselt sein
 - Wenig Kontrolle durch den Endnutzer



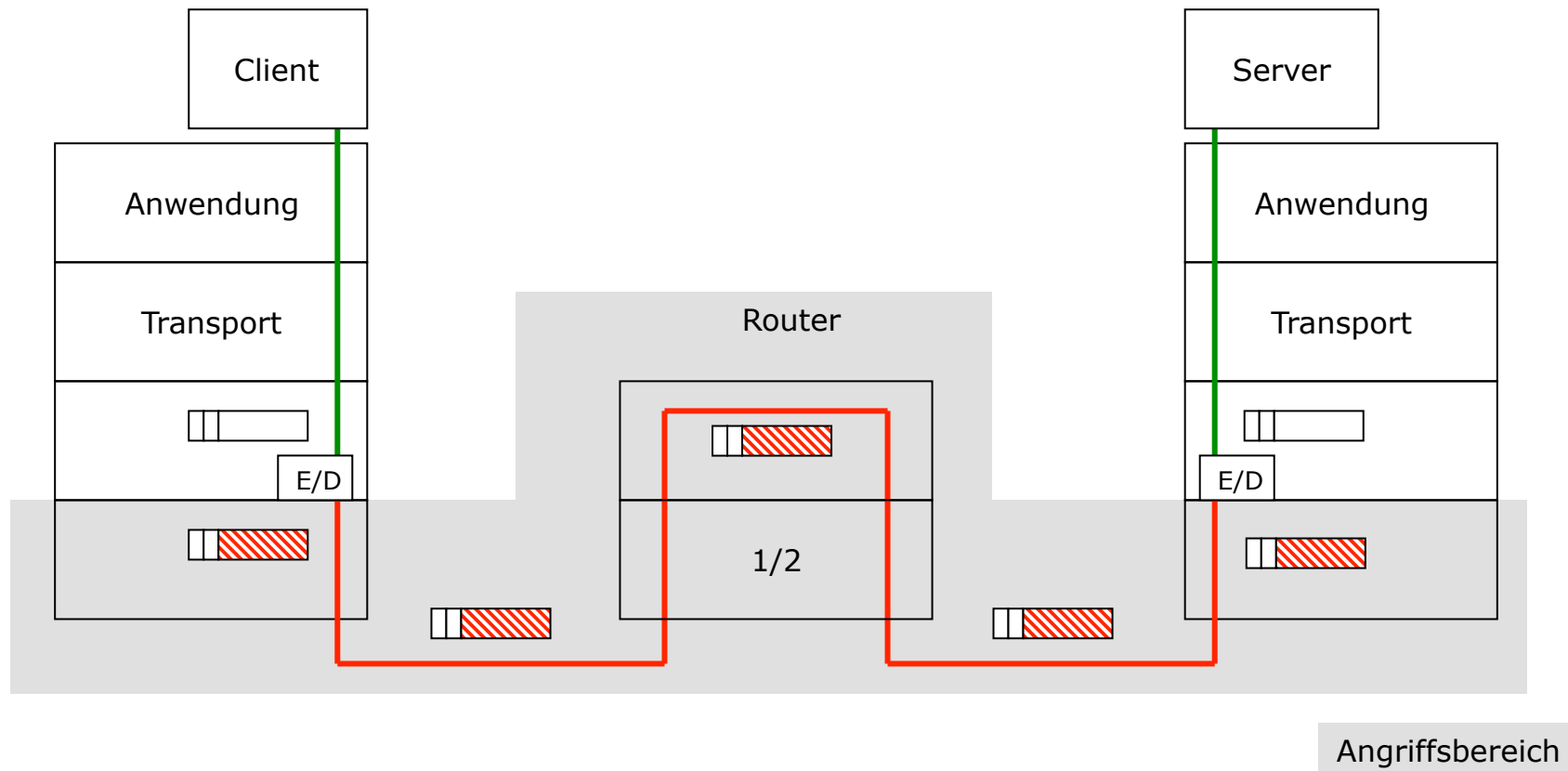
Verschlüsselung in Vermittlungsschicht: IPSec

- Transportmodus
 - Verbindungs- und Ende-zu-Ende-Verschlüsselung möglich



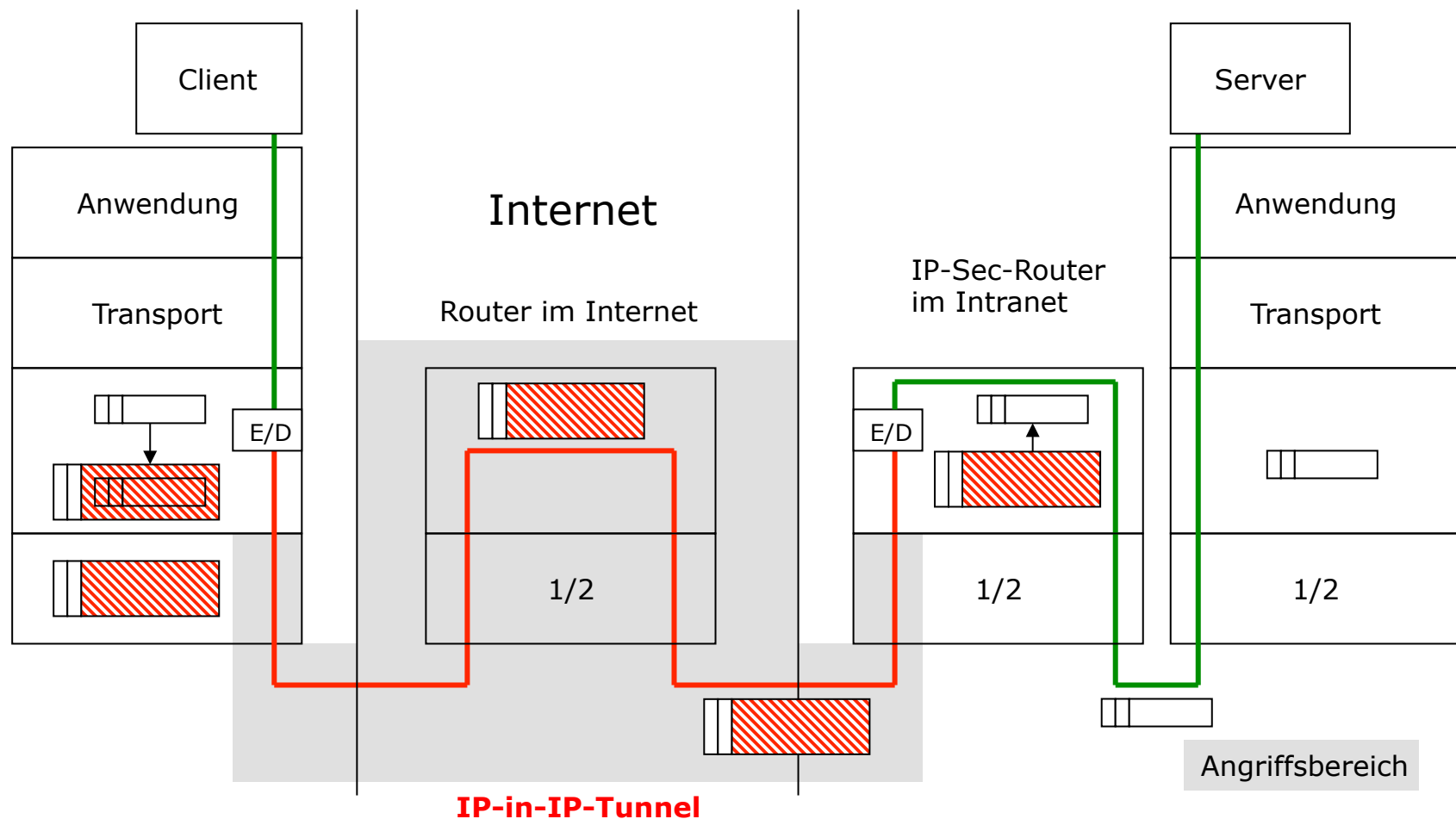
Verschlüsselung in Vermittlungsschicht: IPSec

- Transportmodus
 - Verbindungs- und Ende-zu-Ende-Verschlüsselung möglich



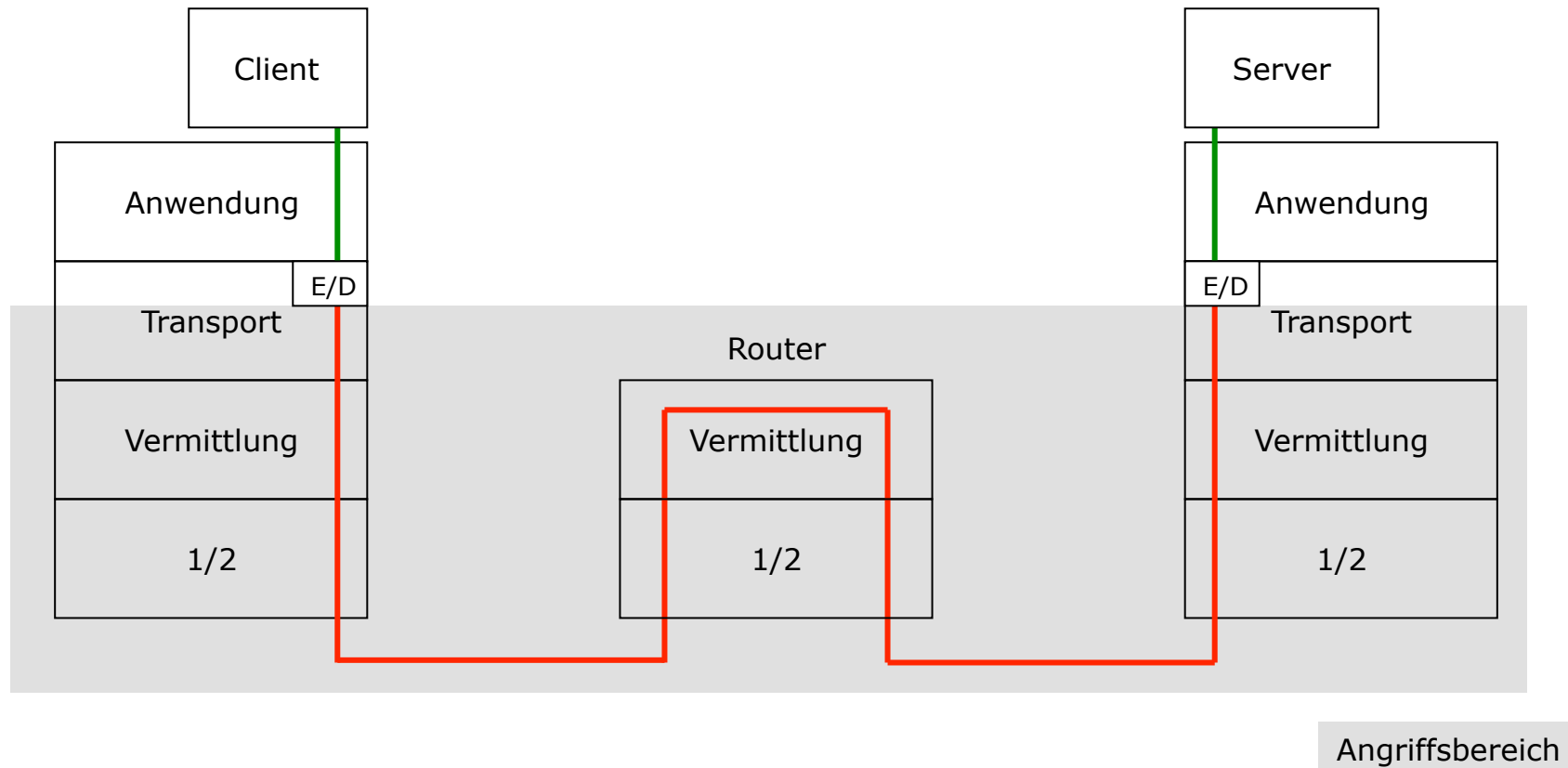
Verschlüsselung in Vermittlungsschicht: IPSec

- Tunnelmodus
 - Momentane Hauptanwendung: Virtuelles Privates Netz



Verschlüsselung in Transportschicht: SSL/TLS

- **Anwendung:**
 - Verschlüsselung von TCP-Verbindungen
 - von Netscape entwickelt
 - in jeden modernen Browser integriert

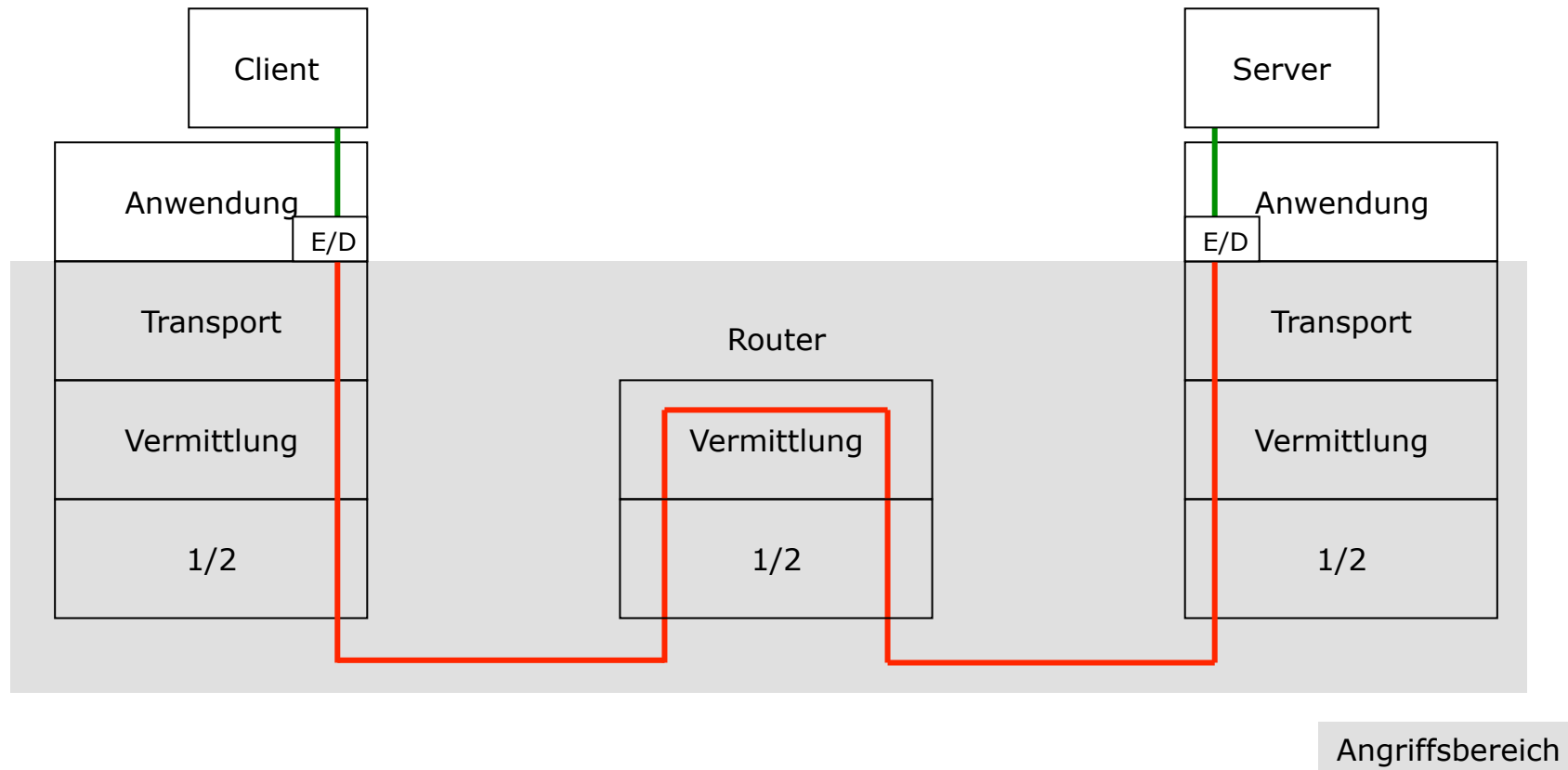


Vergleich SSL – IPSec

	SSL	IPSec
Komplexität	hoch	gering
Anwendungsnahe	hoch	gering
Für VPNs geeignet?	nein	ja
Für paketorientierte Dienste geeignet?	nein	ja
Für verbindungsorientierte Dienste geeignet?	ja	ja

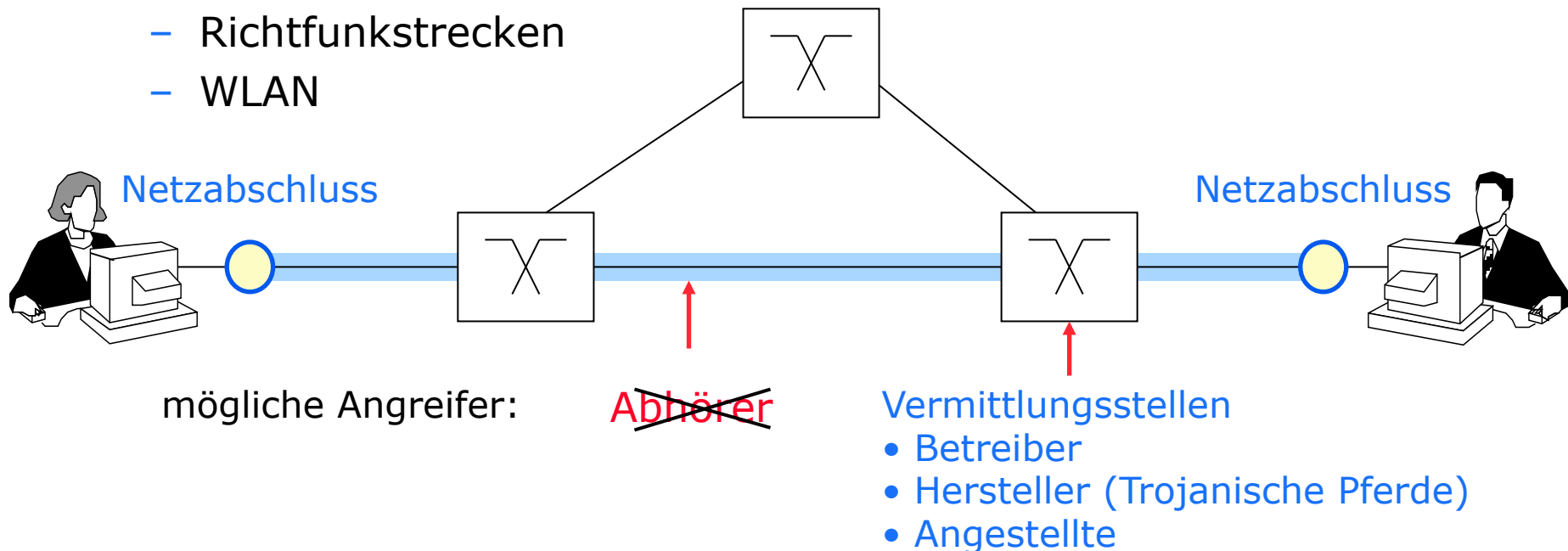
Verschlüsselung in Anwendungsschicht

- Ende-zu-Ende-Verschlüsselung zwischen Client und Server



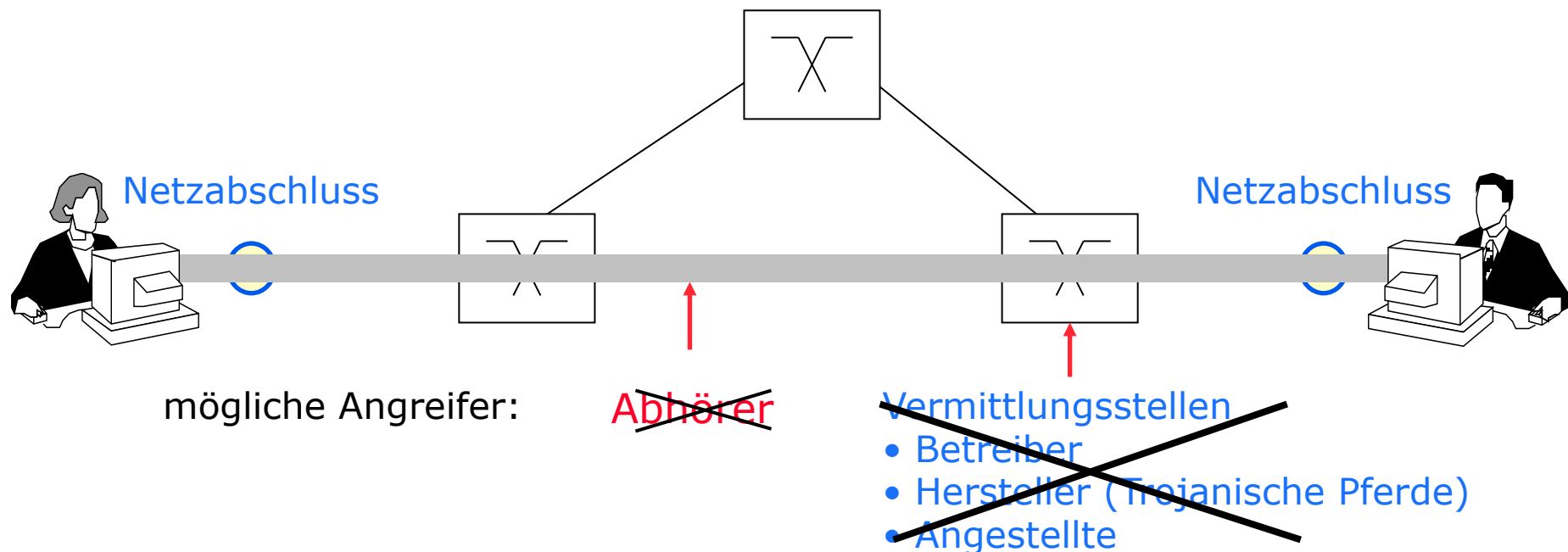
Verbindungsverschlüsselung

- Verbindungsverschlüsselung: (meist symmetrische Verschlüsselung)
 - zwischen Netzabschluss und Vermittlungsstelle
 - zwischen Vermittlungsstelle und Vermittlungsstelle
- In Vermittlungsstelle liegt Klartext vor
- Anwendungsgebiete:
 - Virtuelle Private Netze (VPN)
 - Leitungsverschlüsselung in Telekommunikationsnetzen
 - Richtfunkstrecken
 - WLAN



Ende-zu-Ende-Verschlüsselung

- Ende-zu-Ende-Verschlüsselung der Inhalte
 - von Endgerät zu Endgerät
- Anwendungsgebiete:
 - E-Mail-Verschlüsselung mit PGP oder S/MIME
 - Secure Sockets Layer (SSL)
- Adressierungsinformation kann nicht verschlüsselt werden



Verbindungs- und Ende-zu-Ende-Verschlüsselung

- Kombination von Verbindungs- und Ende-zu-Ende-Verschlüsselung
 - Ende-zu-Ende-Verschlüsselung allein schützt *nicht* die Adressierungsdaten vor **Außenstehenden**
 - zusätzliche Verbindungsverschlüsselung sinnvoll
- Restproblem Verkehrsdaten:
 - **Netzbetreiber** kann weiterhin feststellen, wer mit wem, wann, wie lange, wo, wieviel Information ausgetauscht hat

