Auszüge aus:

Andreas Pfitzmann: Sicherheit in Rechnernetzen;

Mehrseitige Sicherheit in verteilten und durch verteilte Systeme

vollständig im Web unter

http://dud.inf.tu-dresden.de

Kryptologische Grundlagen

erreichbare Schutzziele:

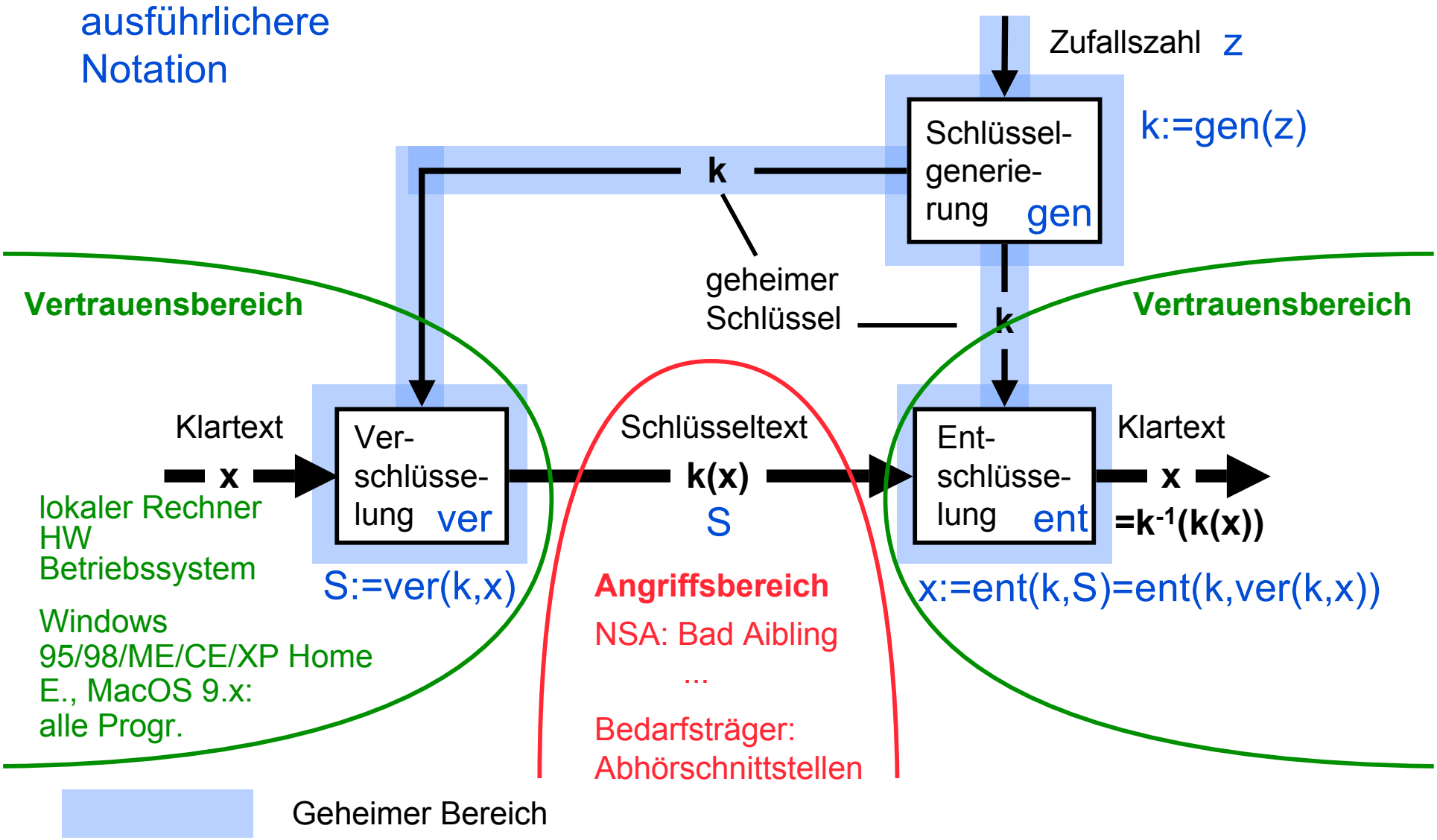
Vertraulichkeit, Konzelation genannt

Integrität (= keine *unerkannte* unbefugte Modifikation von Informationen), Authentikation genannt

durch Kryptographie unerreichbar:

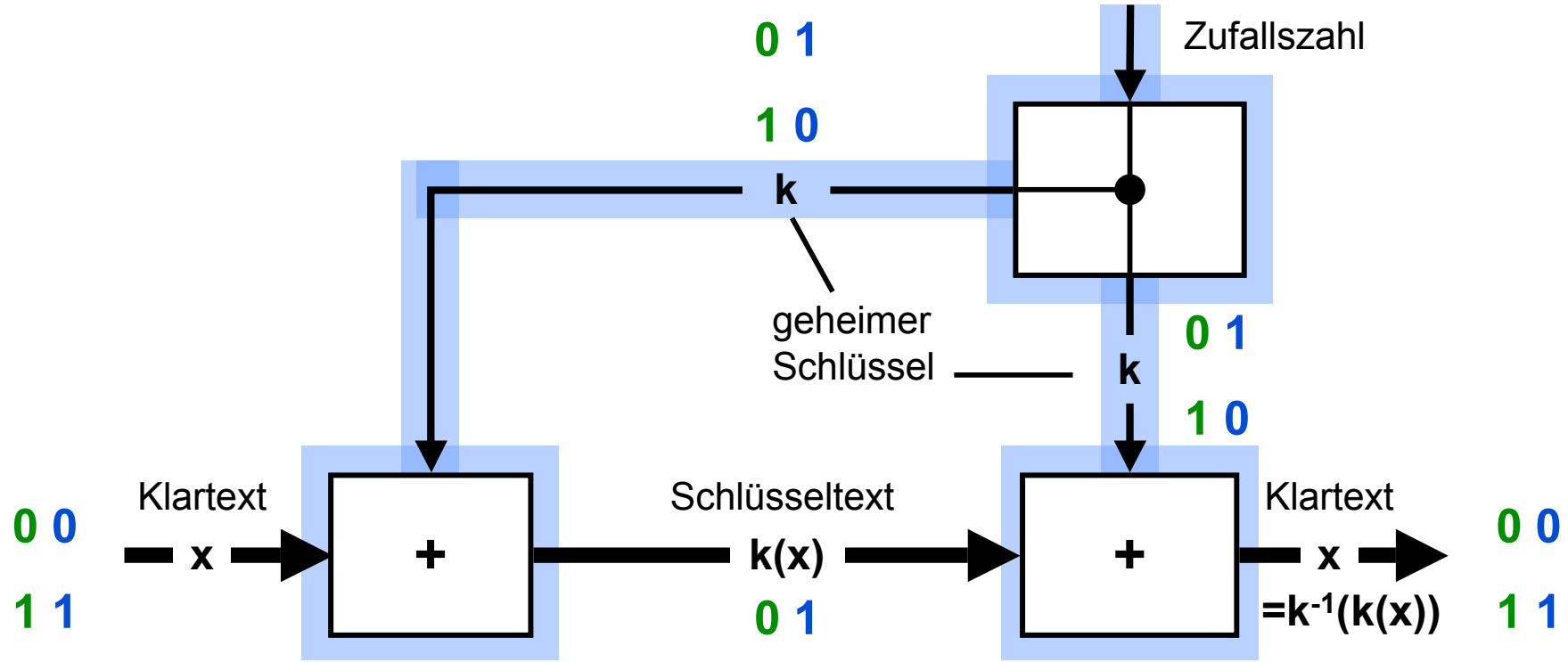
Verfügbarkeit – zumindest nicht gegen starke Angreifer

Symmetrisches Konzelationssystem



Undurchsichtiger Kasten mit Schloß; 2 gleiche Schlüssel

Bsp. Vernam-Chiffre (=one-time-pad)

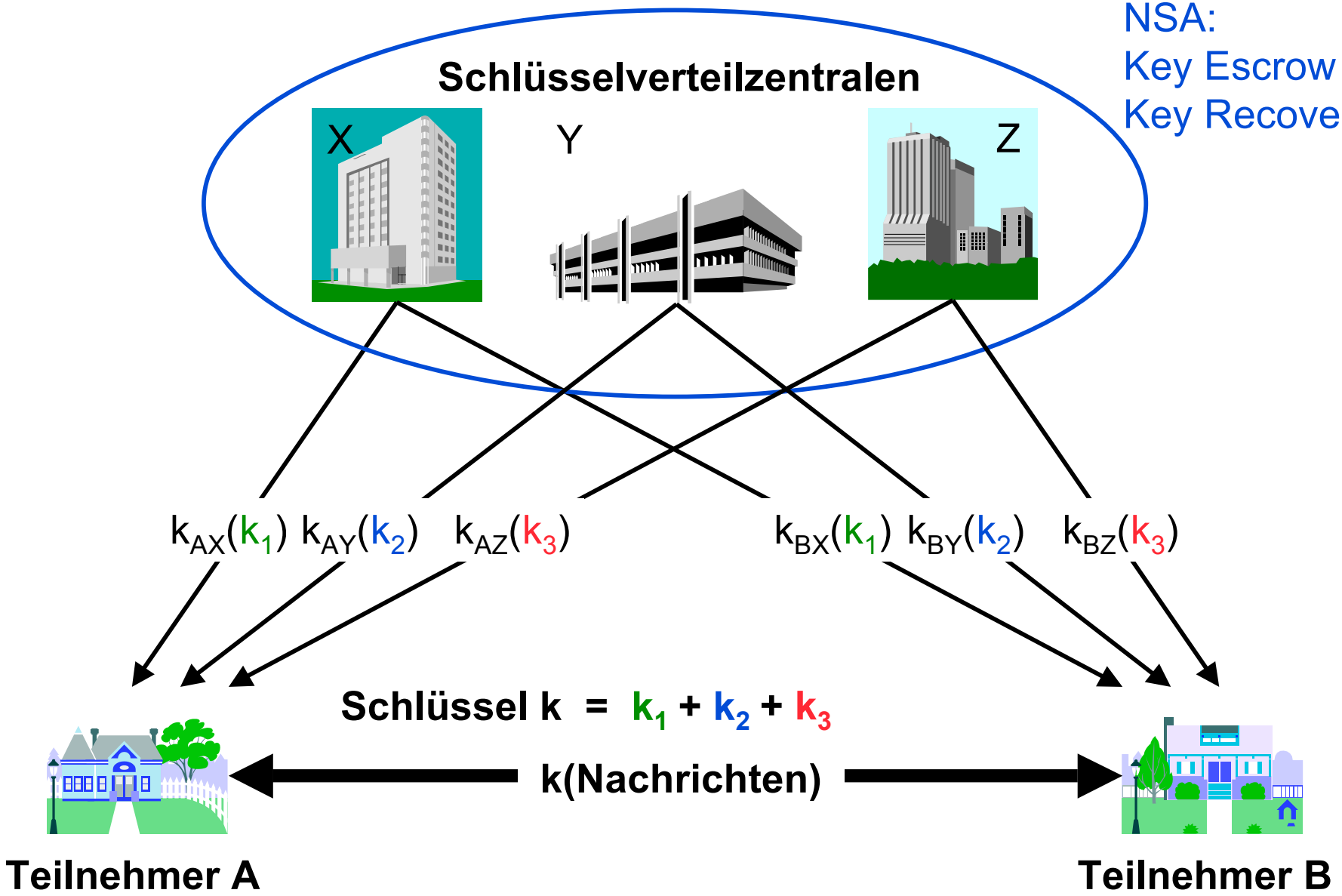


Geheimer Bereich

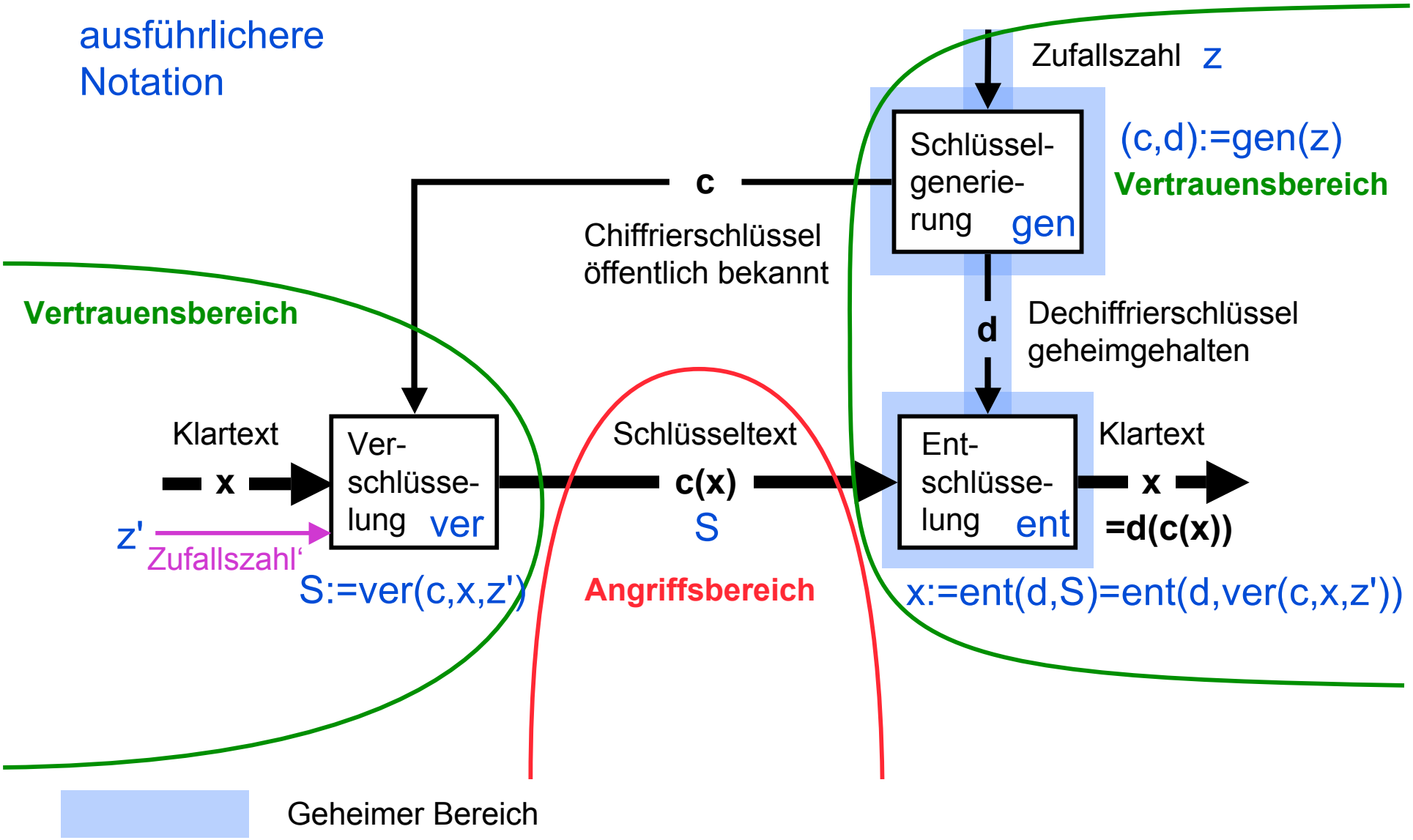
Undurchsichtiger Kasten mit Schloß; 2 gleiche Schlüssel

Schlüsselverteilung bei symmetrischem Kryptosystem

NSA:
Key Escrow
Key Recovery

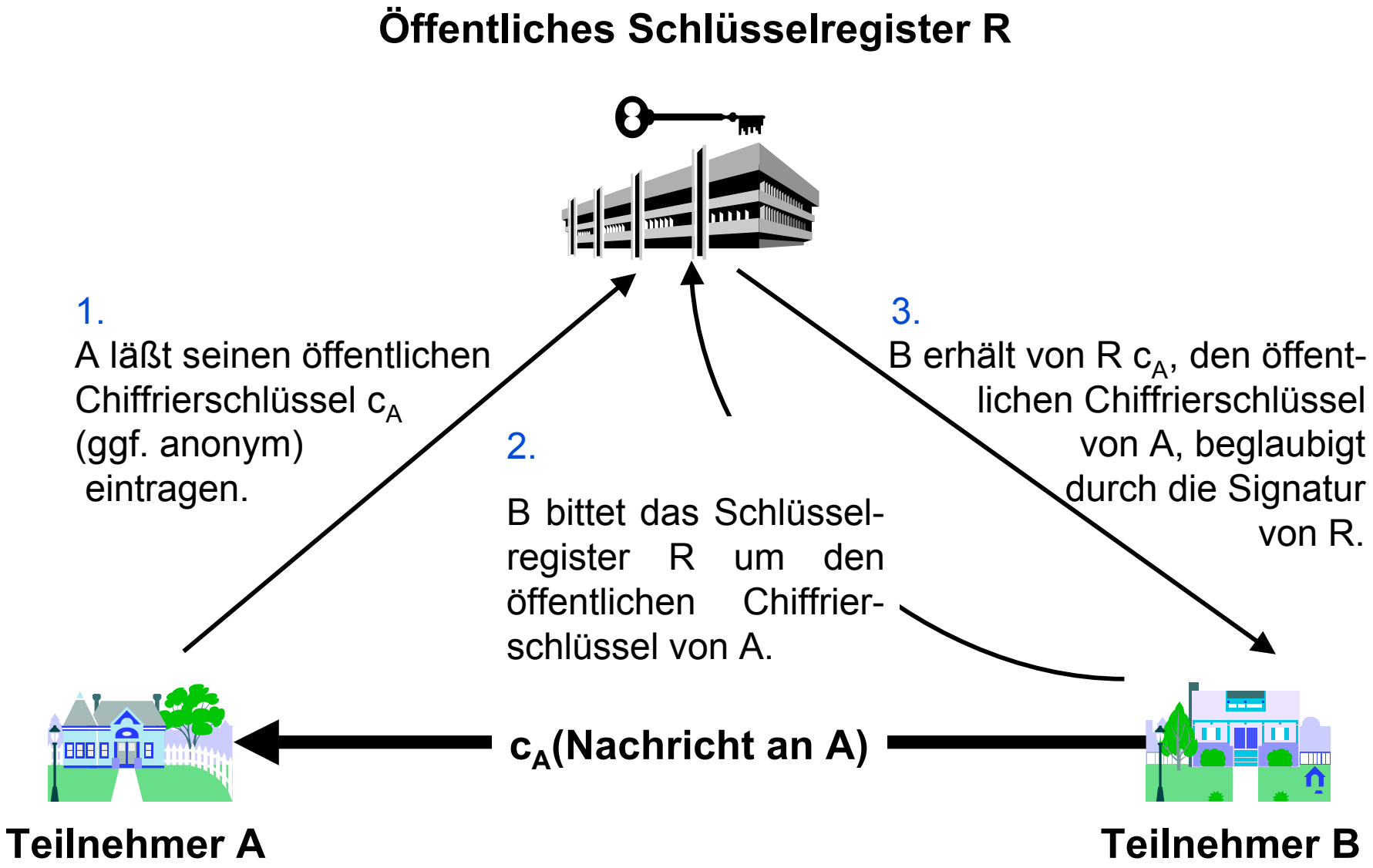


Asymmetrisches Konzelationssystem

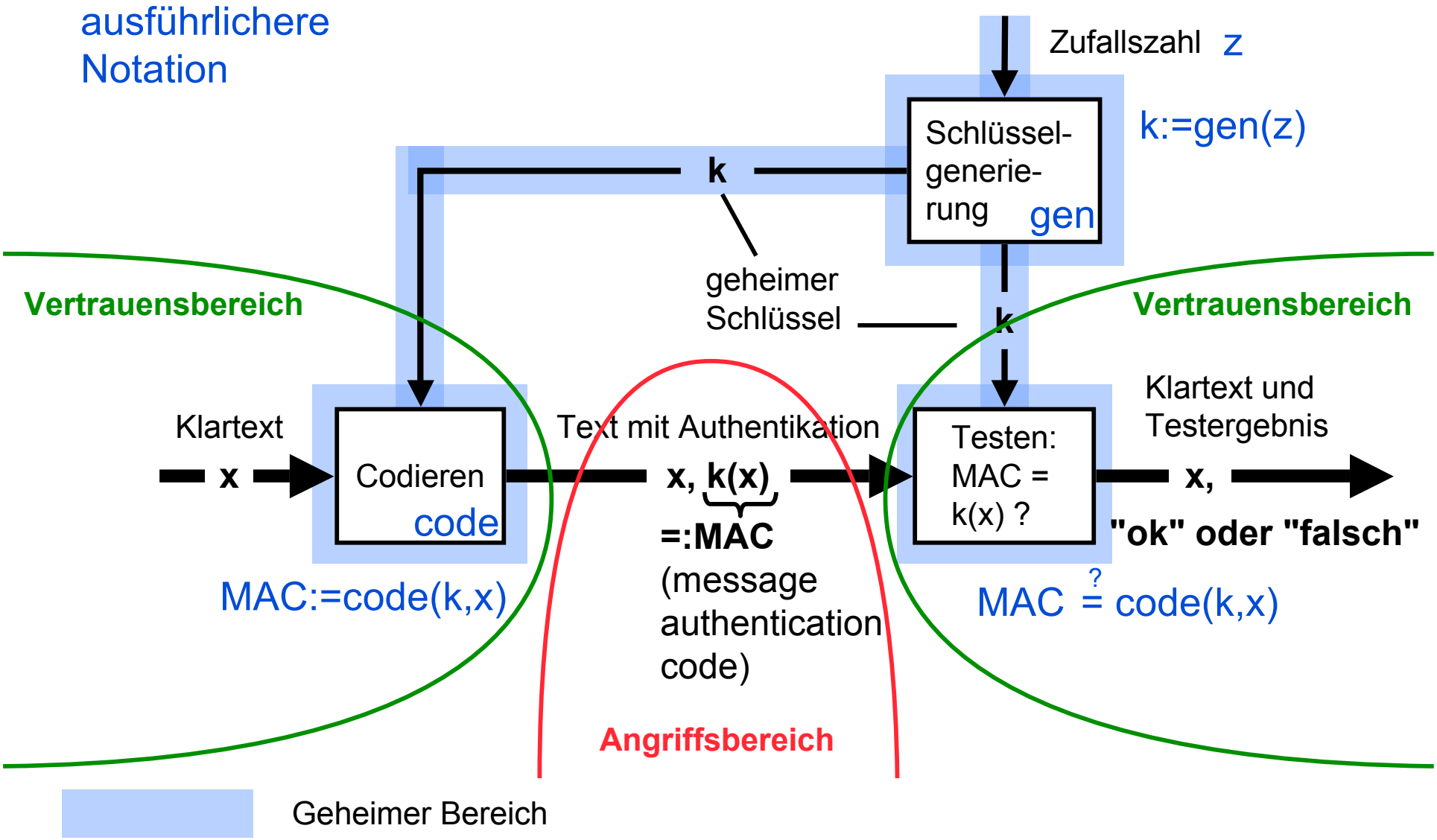


Undurchsichtiger Kasten mit Schnappschloß; 1 Schlüssel

Schlüsselverteilung bei asymmetrischem Konzelationssystem

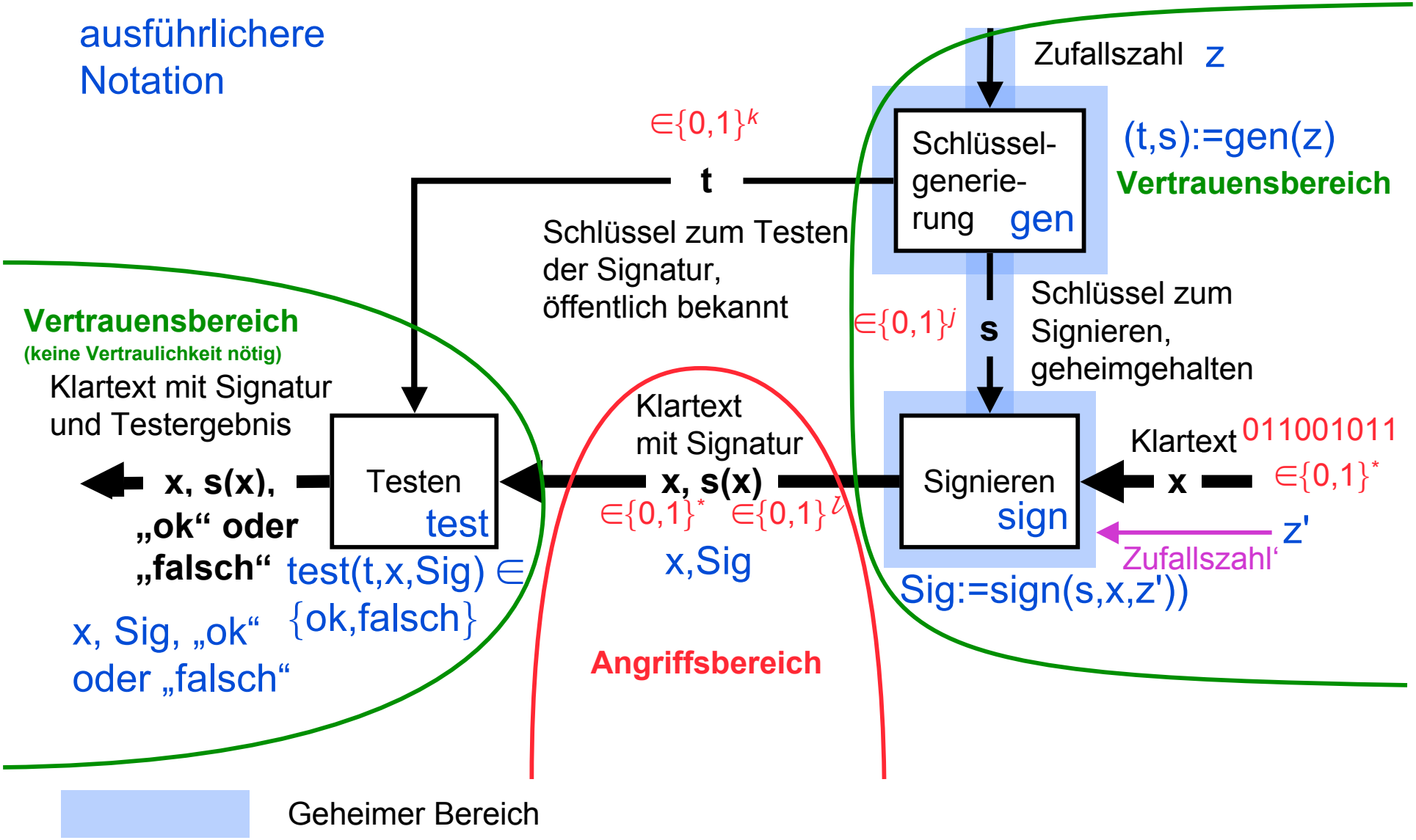


Symmetrisches Authentikationssystem



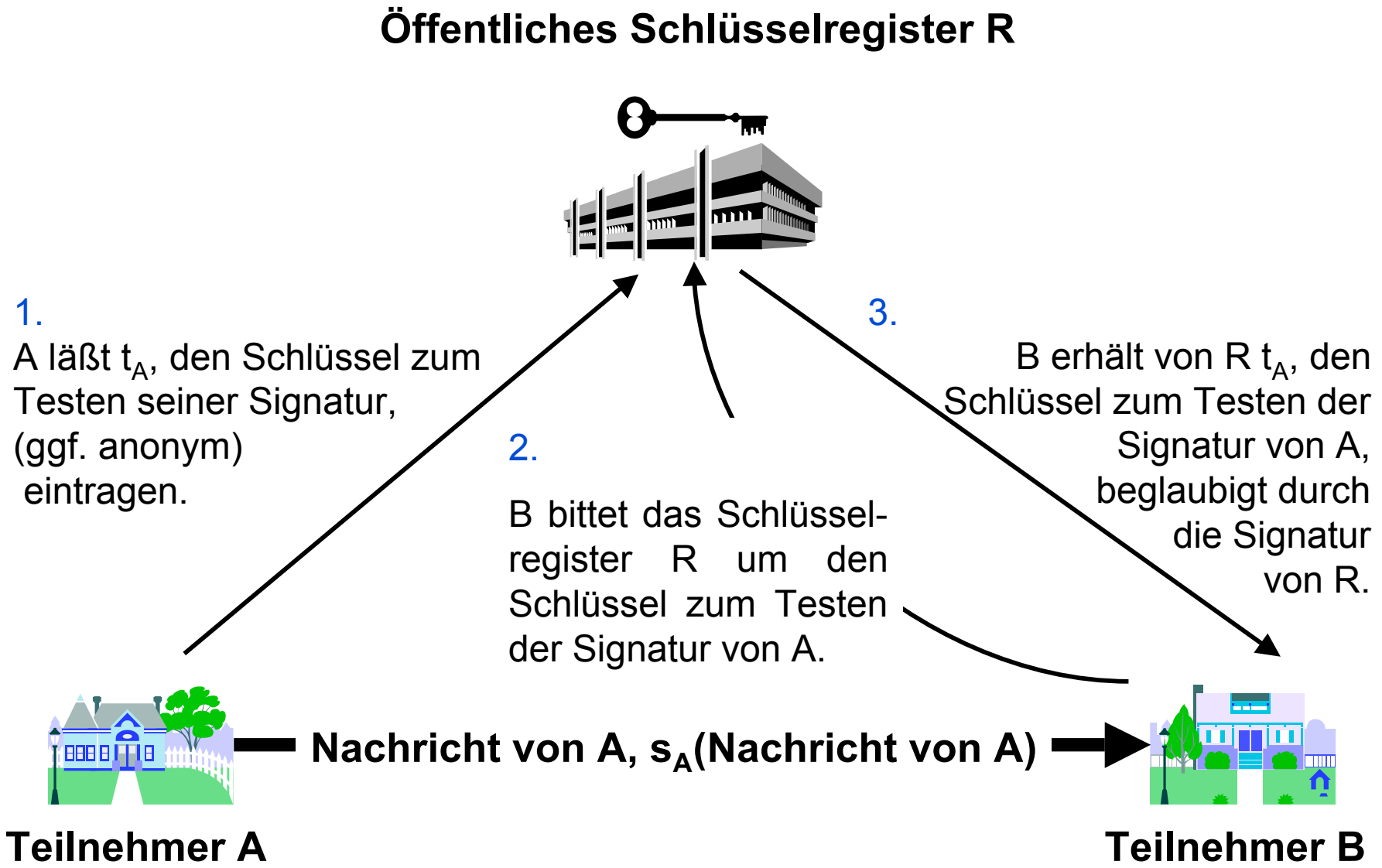
Glasvitrine mit Schloß; 2 gleiche Schlüssel

Digitales Signatursystem

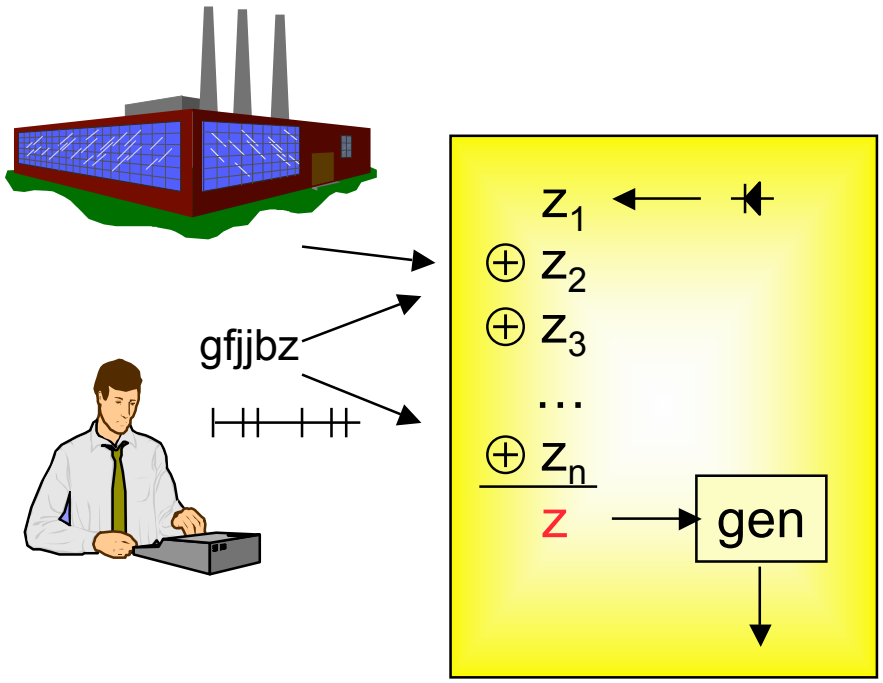


Glasvitrine mit Schloß; 1 Schlüssel

Schlüsselverteilung bei digitalem Signatursystem



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.

Anmerkungen zum Schlüsselaustausch

Wem werden Schlüssel zugeordnet?

1. einzelnen Teilnehmern — asymmetrische Systeme
2. Paarbeziehungen — symmetrische Systeme
3. Gruppen — –

Wie viele Schlüssel müssen ausgetauscht werden?

n Teilnehmer

asymmetrische Systeme: je System *n*

symmetrische Systeme: $n \cdot (n\text{-}1)$

Wann Schlüssel generieren und austauschen?

Sicherheit des Schlüsselaustauschs begrenzt kryptographisch erreichbare Sicherheit:

Mehrere Ur-Schlüsselaustausche durchführen

Hybride Kryptosysteme (1)

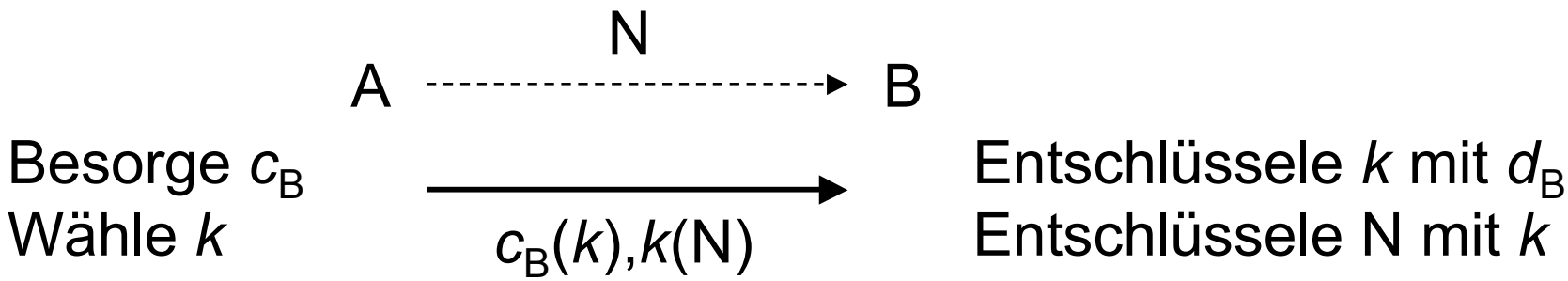
Kombiniere:

- von asymmetrischen: Einfache Schlüsselverteilung
- von symmetrischen: Effizienz (Faktor 100 bis 10000, SW und HW)

Wie?

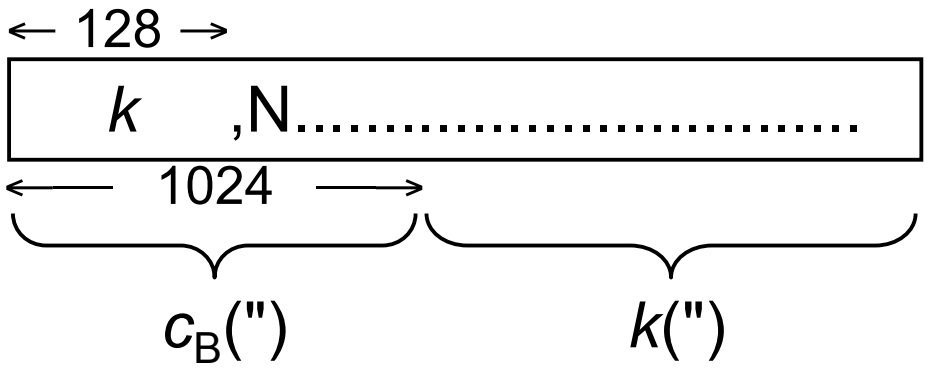
Asymmetrisches System nur, um Schlüssel für symmetrisches auszutauschen

Konzelation:



Hybride Kryptosysteme (2)

Noch effizienter: Teil von N in 1. Block

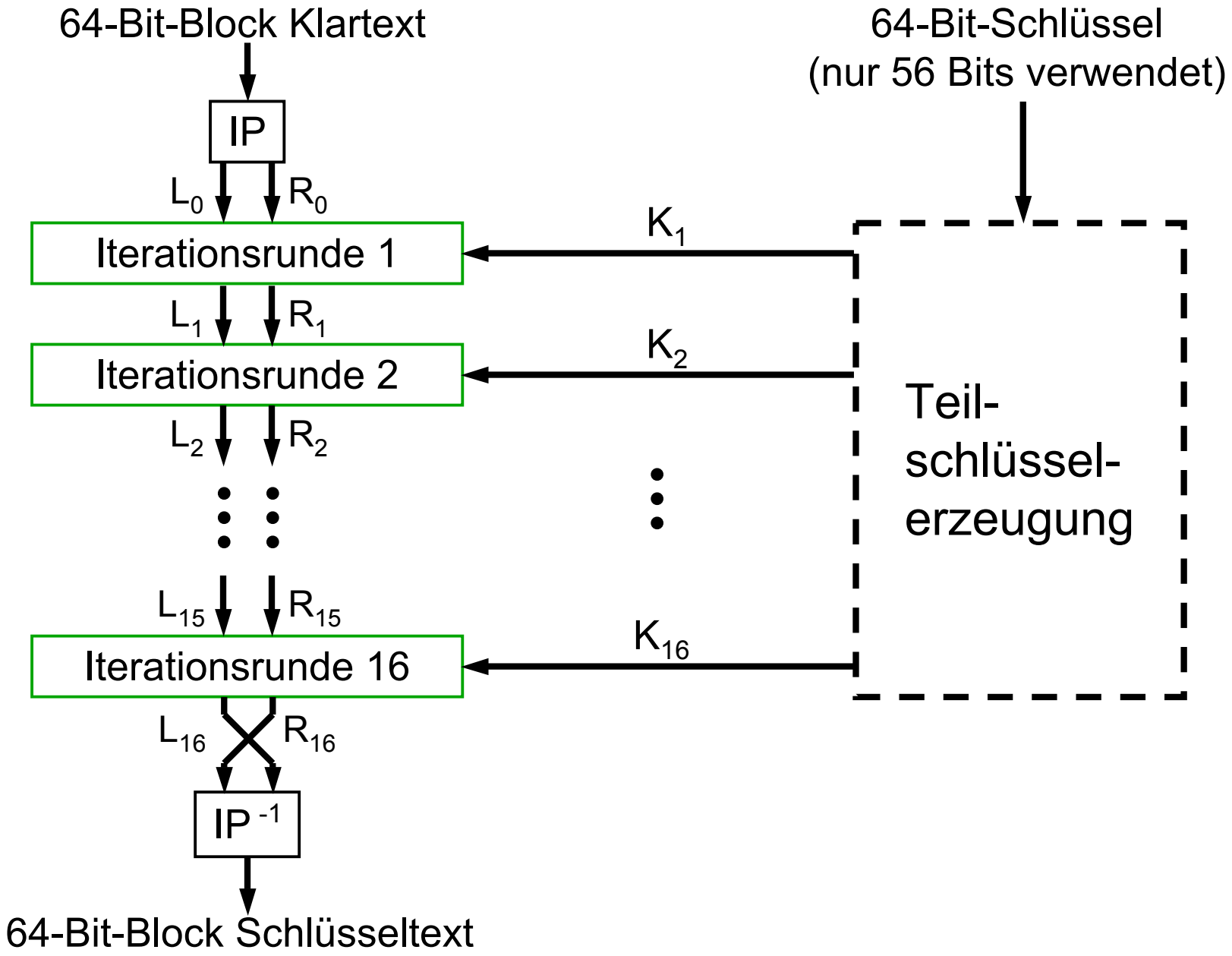


Wenn B auch k benutzen soll: $s_A(B,k)$ dazulegen

Authentikation: k authentisieren und geheimhalten

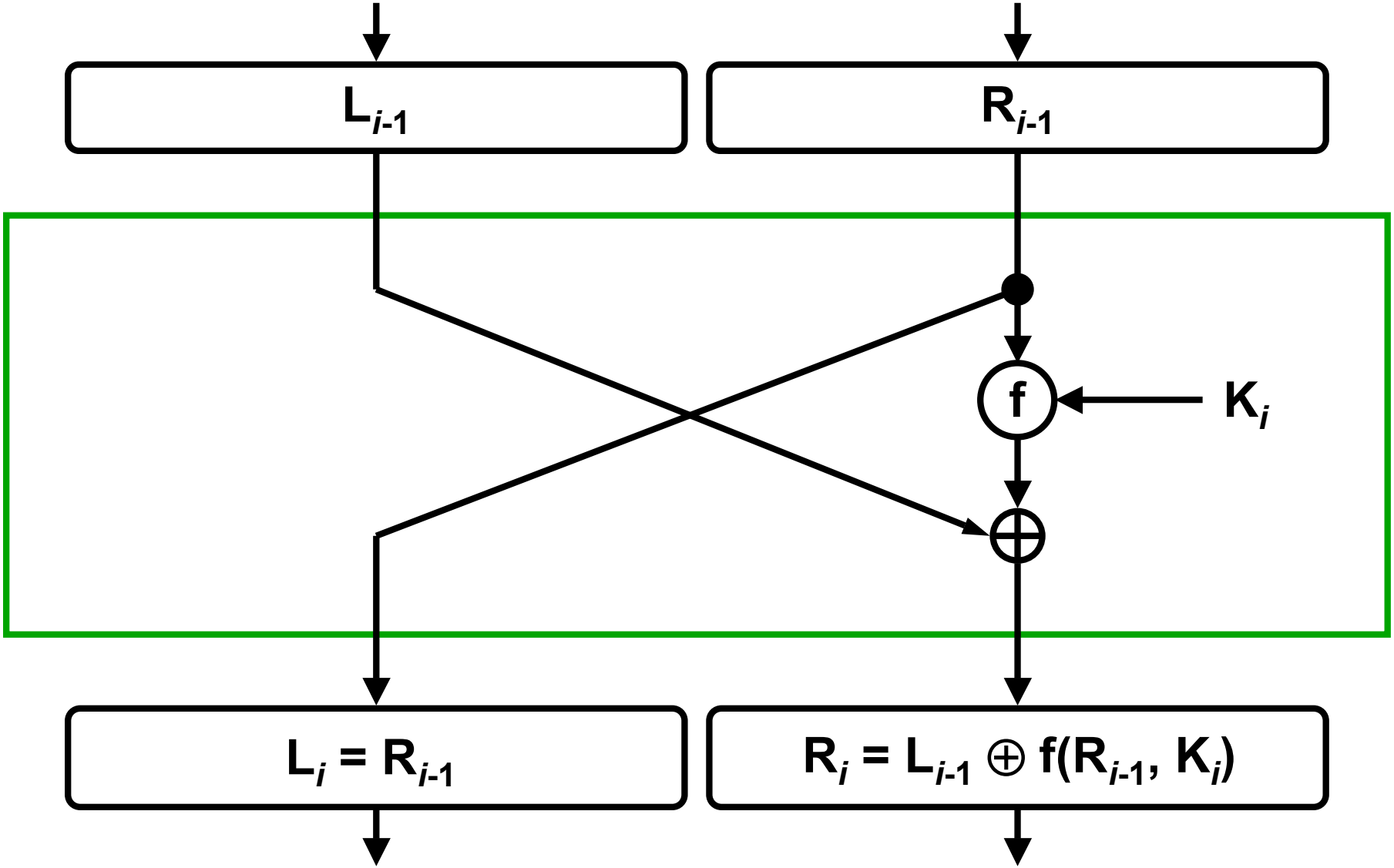
Besorge c_B		Besorge t_A
Wähle k	N,k(N),c_B(B,k,s_A(B,k))	Entschlüssele c_B(B,k,s_A(B,k))
	MAC	Teste B,k mit t_A
		Teste N mit k

Symmetrisches Kryptosystem DES

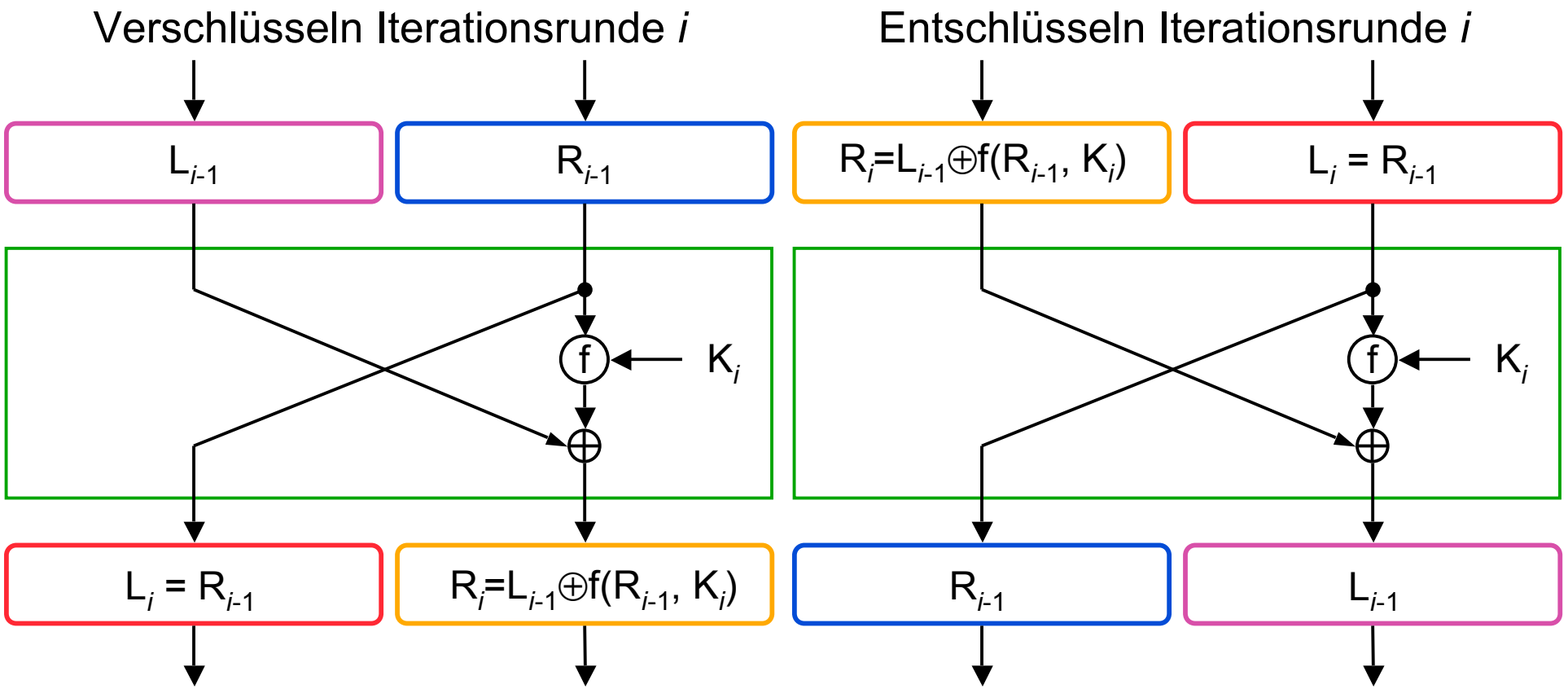


Eine Iterationsrunde

Feistel Chiffren



Entschlüsselungsprinzip



Entschlüsselungsprinzip

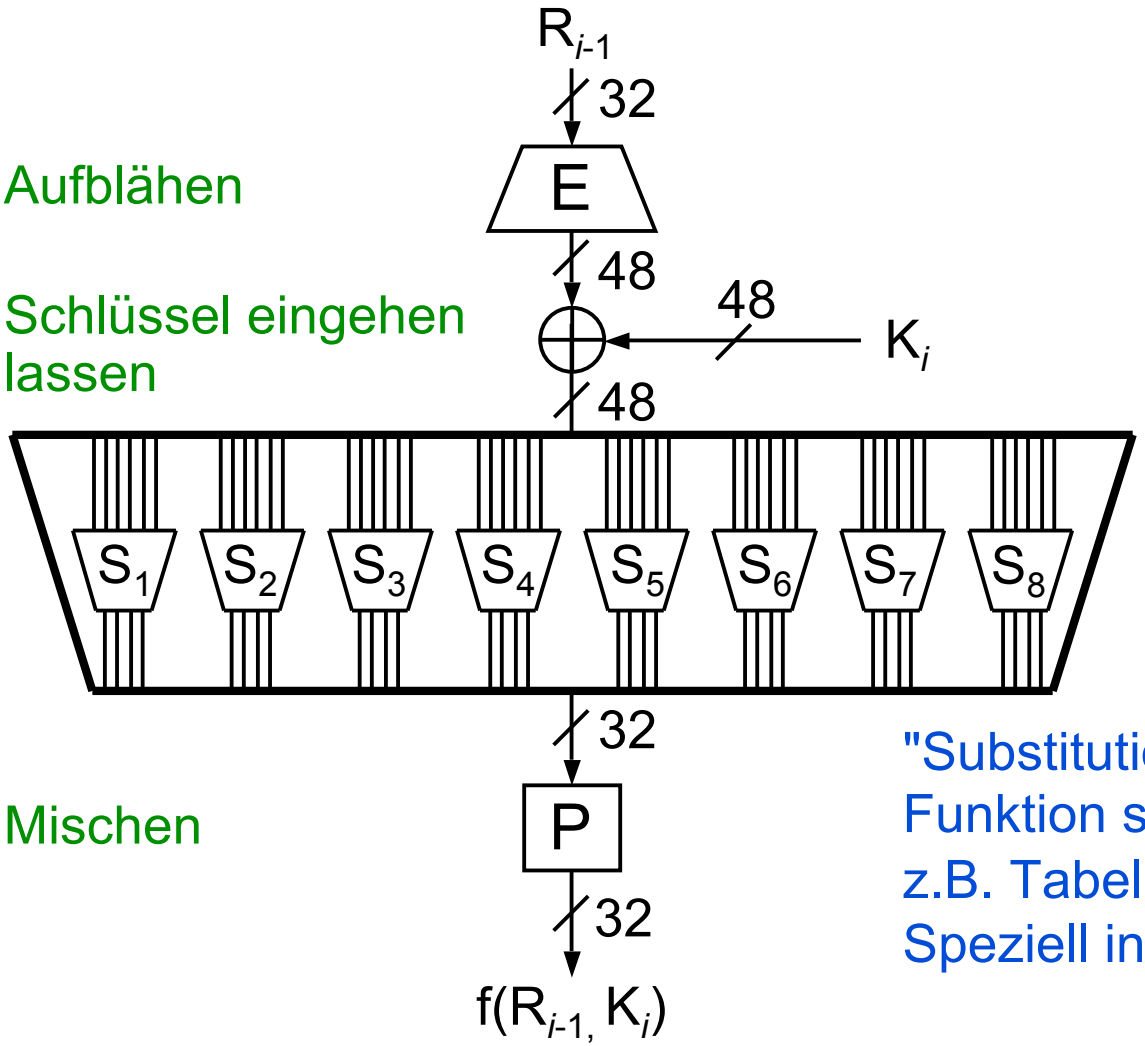
trivial

$L_{i-1} \oplus f(R_{i-1}, K_i) \oplus f(L_i, K_i) =$

$L_{i-1} \oplus f(L_i, K_i) \oplus f(L_i, K_i) = L_{i-1}$

Ersetze R_{i-1} durch L_i

Verschlüsselungsfunktion f



Aufblähen

Schlüssel eingehen lassen

Mischen

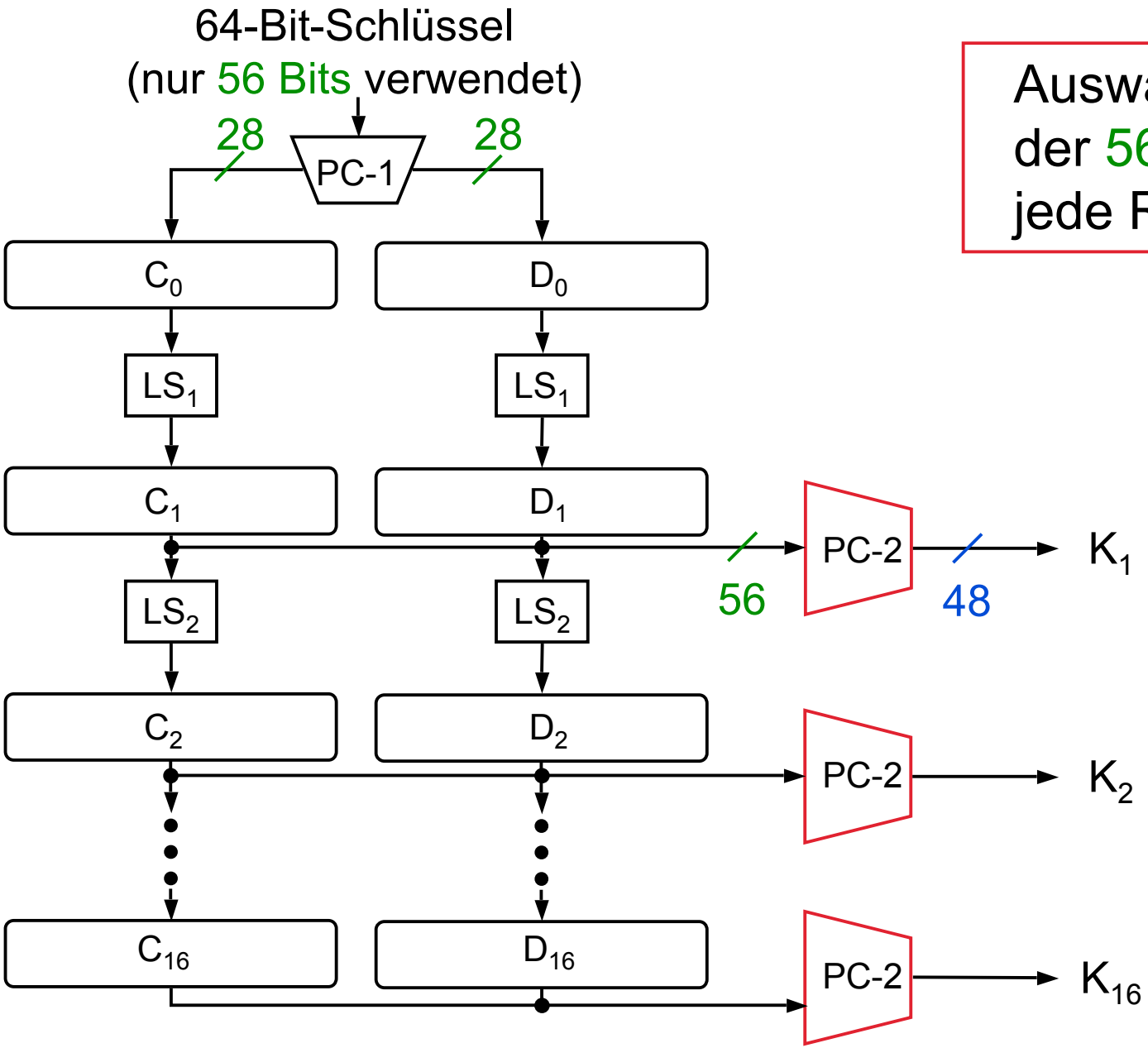
Nichtlinearität schaffen (Permutationen und ⊕ sind linear)

"Substitutionsbox" S kann beliebige Funktion $s : \{0,1\}^6 \rightarrow \{0,1\}^4$ aufnehmen, z.B. Tabelle.
Speziell in DES aber festgelegt.

Begriffe

- Substitutions-Permutationsnetze
- Confusion - Diffusion

Teilschlüsselerzeugung



Auswahl von 48 der 56 Bits für jede Runde

Verallgemeinerung von DES

1.) 56 $\Rightarrow$ 16 • 48 = 768 Schlüsselbits

2.) variable Substitutionsboxen

3.) variable Permutationen

4.) variable Expansionspermutation

5.) variable Anzahl Iterationsrunden

RSA - asymmetrisches Kryptosystem

R. Rivest, A. Shamir, L. Adleman: A Method for obtaining Digital Signatures and Public-Key Cryptosystems; Communications of the ACM 21/2 (Feb. 1978) 120-126.

Schlüsselgenerierung

1) Wähle zwei Primzahlen p und q zufällig sowie stochastisch unabhängig mit $|p| \approx |q| = l,\ p \neq q$
2) Berechne $n := p \cdot q$
3) Wähle c mit $3 \leq c < (p\text{-}1)(q\text{-}1)$ und $\mathrm{ggT}\left(c, \underbrace{(p\text{-}1)(q\text{-}1)}_{\Phi(n)}\right) = 1$
4) Berechne d mittels p, q, c als multiplikatives Inverses von c mod $\Phi(n)$

   $c \cdot d \equiv 1 \pmod{\Phi(n)}$
5) Veröffentliche c und n.

Ver-/Entschlüsselung

Exponentation mit c bzw. d in $\mathbb{Z}_n$

Beh.: $\forall m \in \mathbb{Z}_n$ gilt: $(m^c)^d \equiv m^{c \cdot d} \equiv (m^d)^c \equiv m \pmod{n}$

Beweis (1)

$$c \cdot d \equiv 1 \pmod{\Phi(n)} \Leftrightarrow$$
$$\exists k \in \mathbb{Z}: c \cdot d - 1 = k \cdot \Phi(n) \Leftrightarrow$$
$$\exists k \in \mathbb{Z}: c \cdot d = k \cdot \Phi(n) + 1$$

Also gilt $m^{c \cdot d} \equiv m^{k \cdot \Phi(n)+1} \pmod{n}$

Mittels des Fermatschen Satzes
$\forall m \in \mathbb{Z}_n^*: m^{\Phi(n)} \equiv 1 \pmod{n}$

folgt für alle zu p teilerfremden m

$$m^{p-1} \equiv 1 \pmod{p}$$

Da p-1 ein Teiler von $\Phi(n)$ ist, gilt

$$m^{k \cdot \Phi(n)+1} \equiv_p m^{k \cdot (p-1)(q-1)+1} \equiv_p m \cdot \underbrace{(\underbrace{m^{p-1}}_{1})^{k \cdot (q-1)}}_{1} \equiv_p m$$

Beweis (2)

Gilt trivialerweise für $m \equiv_p 0$

Entsprechende Argumentation für q ergibt

$$m^{k \cdot \Phi(n)+1} \equiv_q m$$

Da Kongruenz sowohl bzgl. p als auch q gilt, gilt sie auch

bzgl. $p \cdot q = n$

$$m^{c \cdot d} \equiv m^{k \cdot \Phi(n)+1} \equiv m \pmod{n}$$

Vorsicht:

Es gibt (bisher ?) **keinen** Beweis

RSA leicht zu brechen $\Rightarrow$ Faktorisierung leicht

Naiver unsicherer Einsatz von RSA

RSA als asymmetrisches Konzelationssystem

Codiere Nachricht (ggf. geblockt) als Zahl $m < n$.

Verschlüsselung von m: $m^c \bmod n$

Entschlüsselung von m^c: $(m^c)^d \bmod n = m$

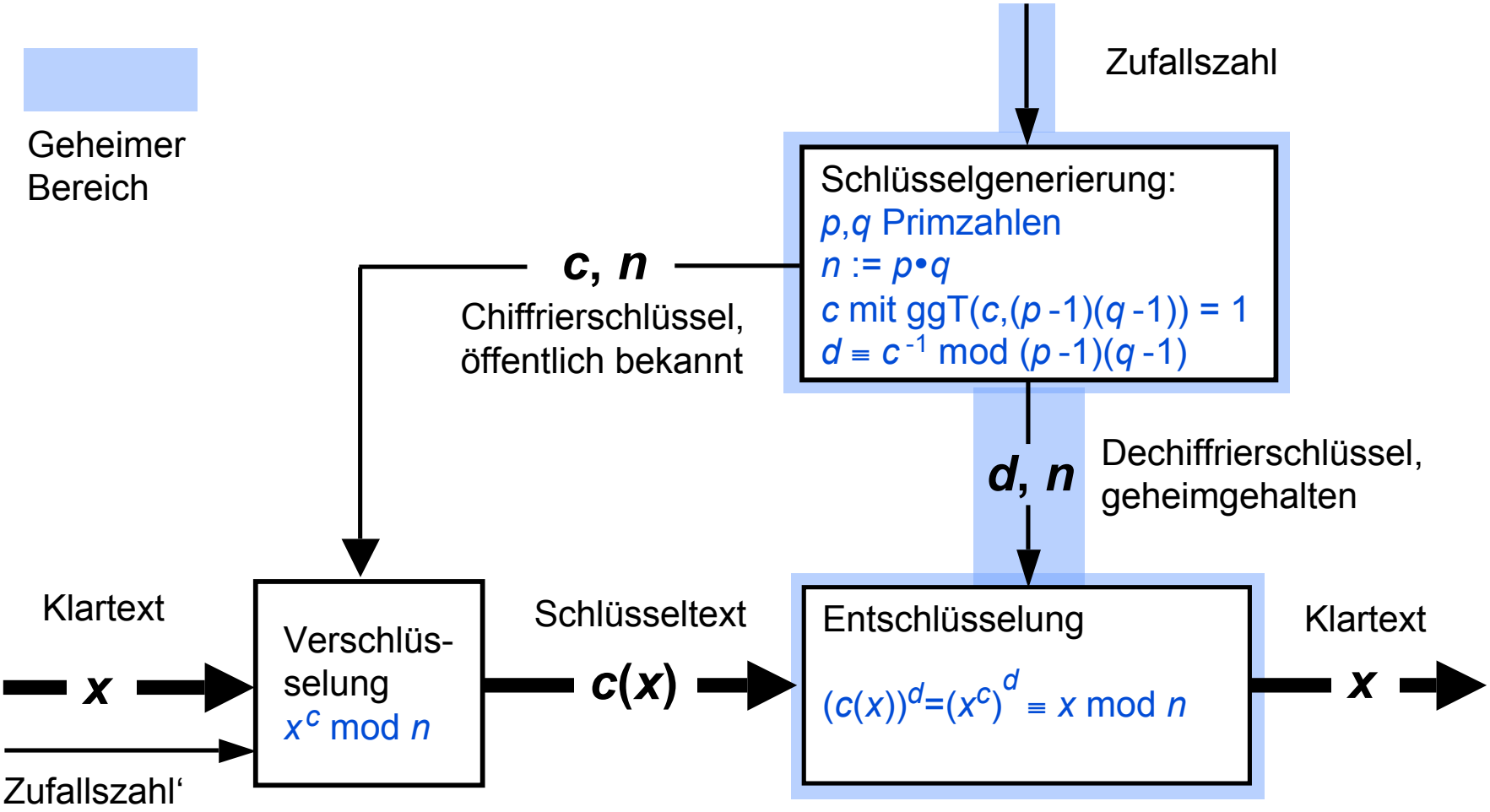
RSA als digitales Signatursystem

Umbenennung: $c \rightarrow t$, $d \rightarrow s$

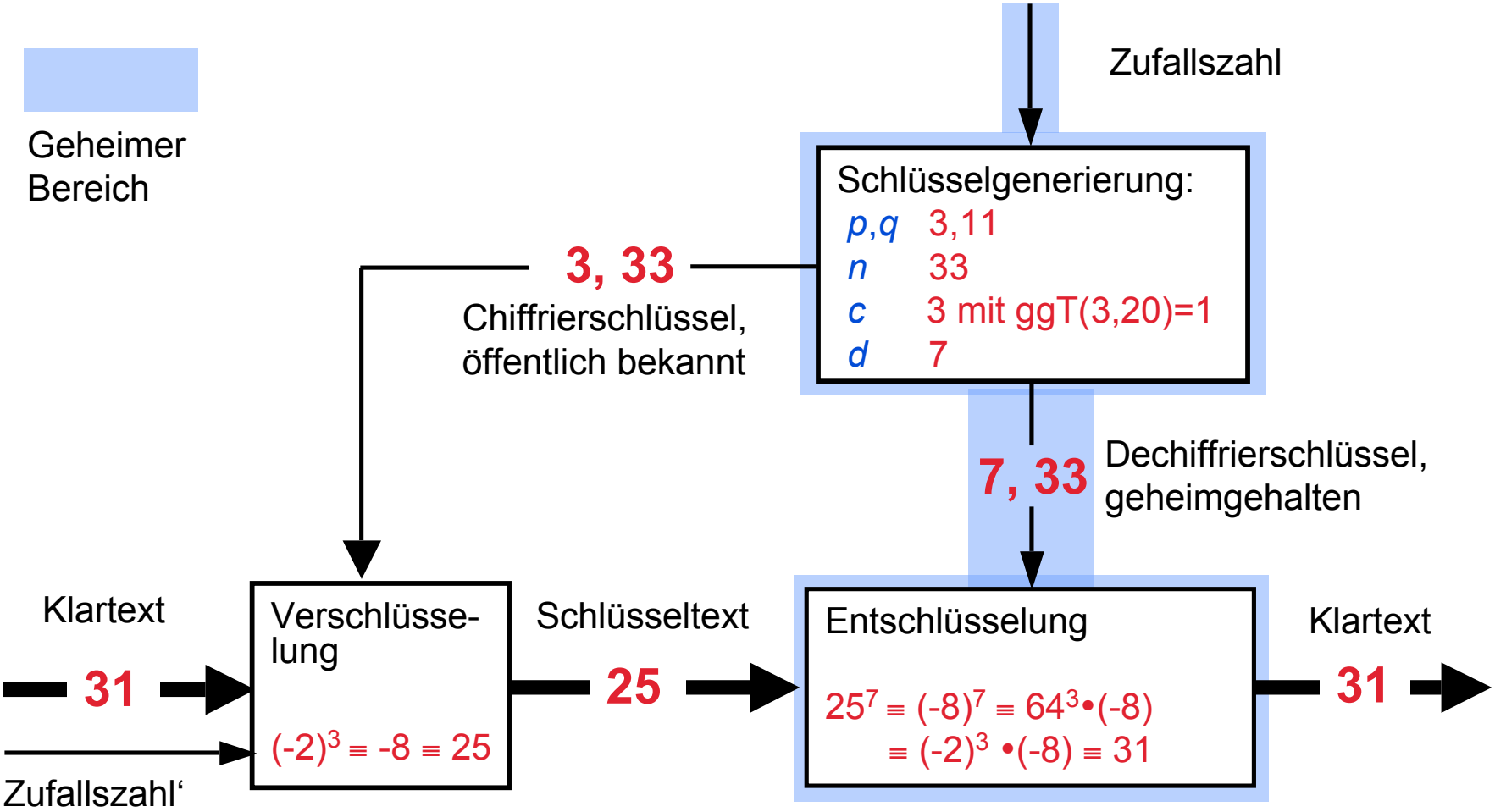
Signieren von m: $m^s \bmod n$

Testen von m, m^s: $(m^s)^t \bmod n = m$?

RSA als asymmetrisches Konzelationssystem: naiv



RSA als asymmetrisches Konzelationssystem: Beispiel



Angriff auf Konzelation mit RSA naiv

$$(x^c)^d \equiv x$$

Schlüsseltext abgehört

$$(x \bullet y)^c = x^c \bullet y^c$$

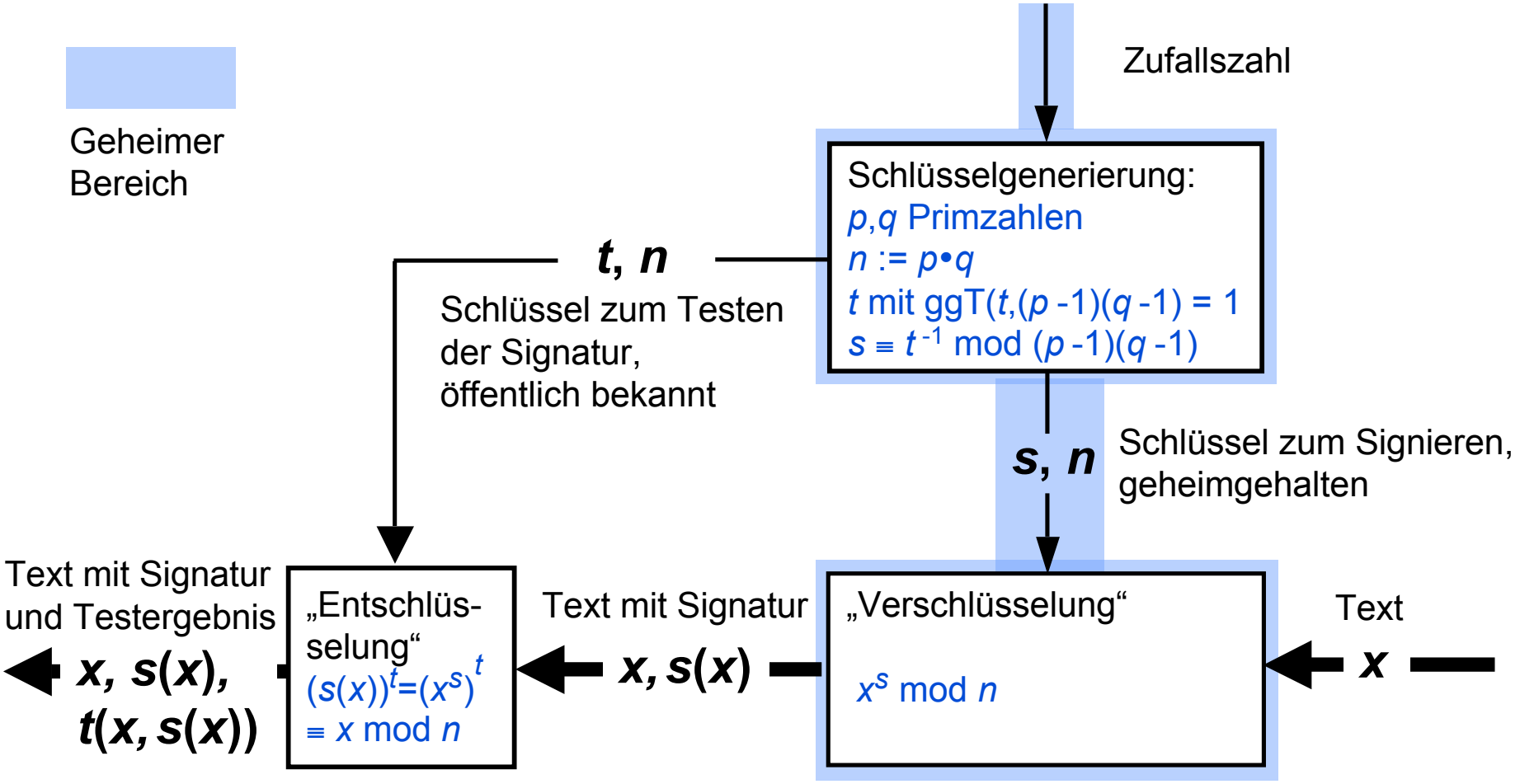
aus y selbst gebildet

entschlüsseln lassen

$$((x \bullet y)^c)^d \equiv x \bullet y$$

teile durch y, erhalte x

RSA als digitales Signatursystem: naiv



Angriff auf digitale Signatur mit RSA naiv

$$(x^s)^t \equiv x$$

gewünschte Nachricht

$$(x^s \cdot y)^t \equiv x \cdot y^t$$

gewählte Nachricht y

signieren lassen

$$((x^s \cdot y)^t)^s \equiv x^s \cdot y$$

teile durch y, erhalte x^s

Abwehr der Davida-Angriffe mittels kollisionsresist. Hashfkt.

h() : kollisionsresistente Hashfunktion

1.) asymmetrisches Konzelationssystem

Klartextnachrichten müssen Redundanzprädikat erfüllen

m, Redundanz $\Rightarrow$ prüfe ob $h(m)$ = Redundanz

2.) digitales Signatursystem

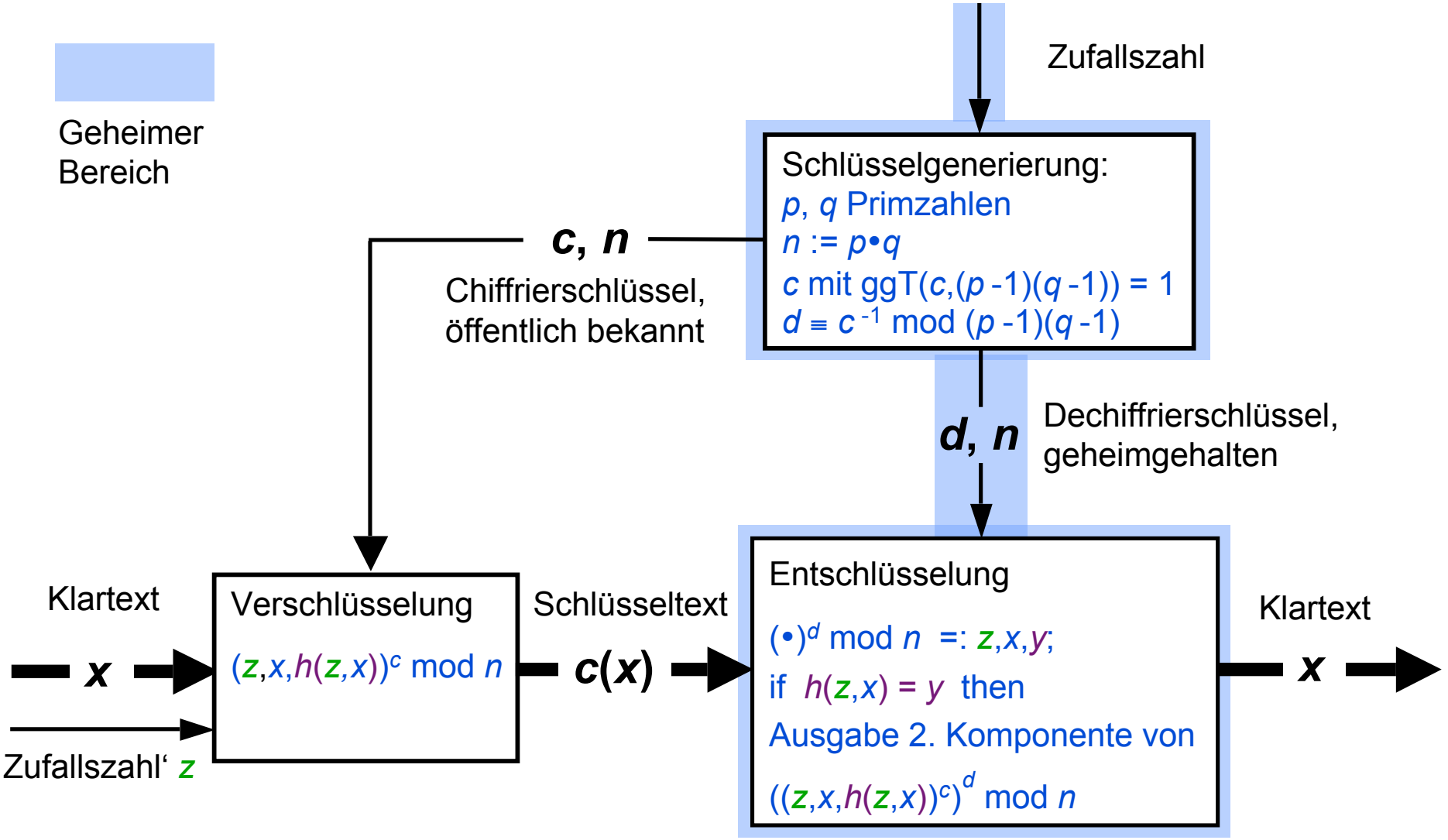
Vor dem Signieren wird auf die Nachricht h angewendet

Signatur zu m $= (h(m))^s \bmod n$

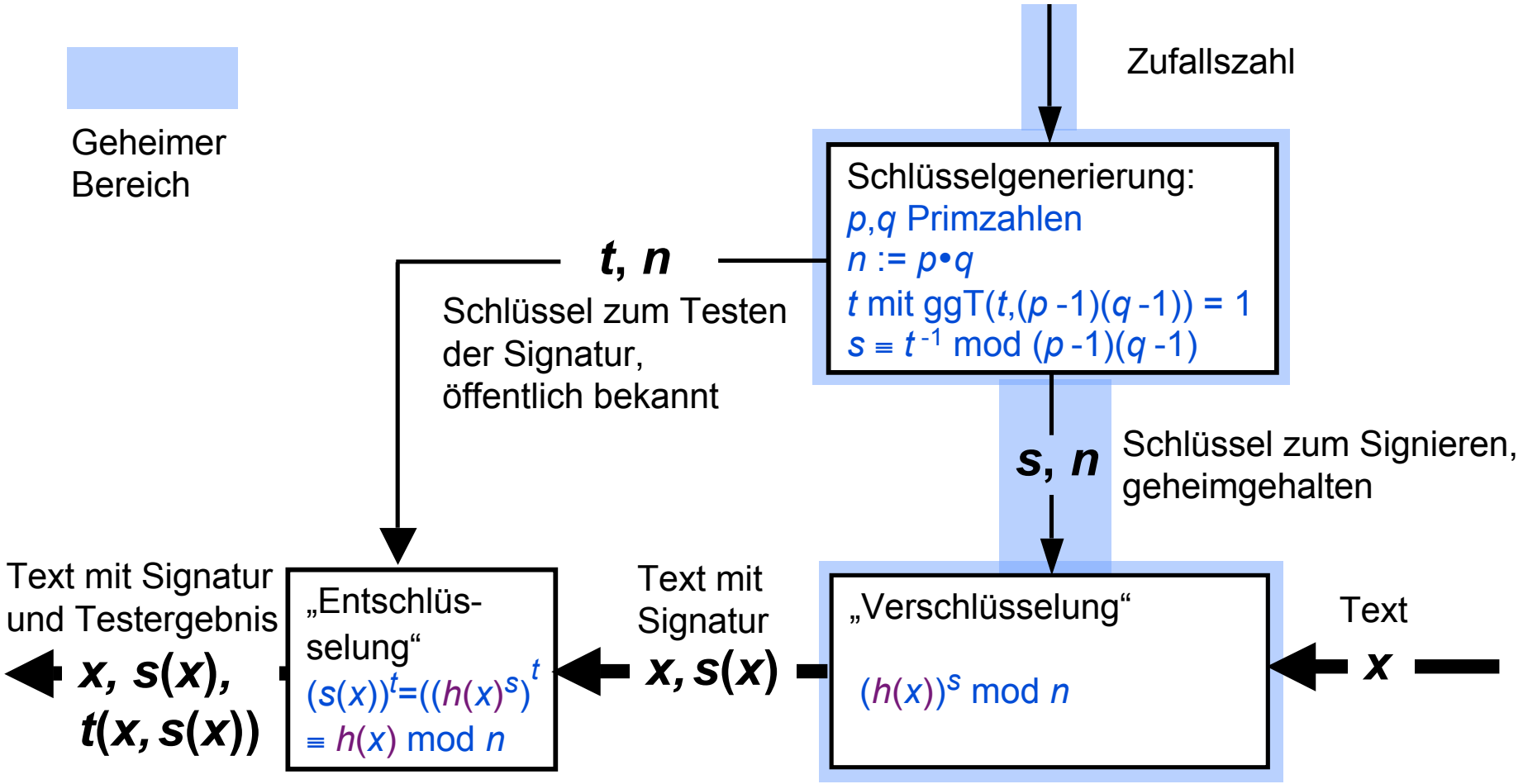
prüfe ob $h(m) = ((h(m))^s)^t \bmod n$

Vorsicht: Es gibt (bisher?) keinen Beweis für Sicherheit!

RSA als asymmetrisches Konzelationssystem



RSA als digitales Signatursystem



kollisionsresistente Hashfunktion *h*
- global bekannt -