Sicherheit in Rechnernetzen

Mehrseitige Sicherheit in verteilten und durch verteilte Systeme

Folien zu den Vorlesungen:

Einführung in die Datensicherheit
Kryptographie

Andreas Pfitzmann
TU Dresden, Fakultät Informatik, D-01062 Dresden
Nöthnitzer Str. 46, Raum 3071
Tel.: 0351/ 463-38277, e-mail: pfitza@inf.tu-dresden.de, http://dud.inf.tu-dresden.de/

Vertiefungsrichtung Sicherheit / technischer Datenschutz

Lehrveranstaltung	*Lehrende(r)*	*SWS*
Einführung in die Datensicherheit	Pfitzmann	1/1
Kryptographie	Pfitzmann	2/2
Datensicherheit durch verteilte Systeme	Pfitzmann	1/1

(ab WS 2004/05 obige 3 LVS zusammen in Security and Cryptography I + II, je 2/2 SWS, 6 cr)

Kryptographie und -analyse	Franz	2
Kanalkodierung	Schönfeld	2/2
Steganographie und Multimedia-Forensik	Franz	2/1
Praktikum: Kryptographie und Datensicherheit	Clauß	/4
Lehrprojekt: Datenschutzfreundl. Technologien im Internet	Clauß, Köpsell	/2
Informatik und Gesellschaft	Pfitzmann	2
Hauptseminar: Technischer Datenschutz	Pfitzmann et.al.	2

Lehr- und Forschungsgebiete

- Mehrseitige Sicherheit, insbesondere Sicherheit durch verteilte Systeme
- Datenschutzfreundliche Technologien
- Kryptographie
- Steganographie
- Multimedia-Forensik
- Informations- und Kodierungstheorie

- Anonymer Webzugriff (Projekt: AN.ON, JAP)
- Identitätsmanagement (Projekte: PRIME, PrimeLife, FIDIS)
- SSONET und Nachfolgeaktivitäten
- Steganographie (Projekt: CRYSTAL)

Ziele von Lehre an Universitäten

Wissenschaft soll u.a. klären

Wie etwas ist.

Vor allem aber auch

Warum etwas so ist

oder

Wie es alternativ sein könnte

(und vielleicht auch sollte).

„**Ewige Wahrheiten**“ (d.h. Wissen mit großer Relevanzzeit) sollten an Universitäten mehr als 90% des Lehr- und Lernaufwands ausmachen.

Allgemeine Ausbildungsziele (nach Prioritäten)

1. Erziehung zu **Ehrlichkeit** und **realistischer Selbsteinschätzung**
2. Anregung zu realistischer **Fremdeinschätzung** von Personen, Firmen und Organisationen
3. **Sicherheits- und Datenschutzbedürfnisse** ermitteln
   - Realistische Schutzziele
   - Realistische Angreifermodelle / Vertrauensmodelle
4. **Validierung** und **Verifikation**, inkl. prinzipielle und praktische **Grenzen**
5. Sicherheits- und Datenschutz**mechanismen**
   - Kennen und verstehen sowie
   - Entwickeln können

Kurzum: ***Integre IT-Sicherheitsexpert(inn)en mit eigenem Urteil und Rückgrat.***

Allgemeine Ausbildungsziele — Wie erreichen ?

1. Erziehung zu **Ehrlichkeit** und **realistischer Selbsteinschätzung**

   Als Lehrende(e) die eigenen
   - **Stärken und Schwächen sowie**
   - **Grenzen thematisieren.**

   Mündliche Prüfung:
   - **Falsche Antworten deutlich negativer werten als „weiß nicht“.**
   - **„Kostenlose“ Möglichkeit, max. 25% des Stoffes jeder Lehrveranstaltung auszuklammern.**
   - **Angebot, mit dem Lieblingsthema zu beginnen.**
   - **Prüfen in die Tiefe bis zum Nichtwissen – sei es von Prüfer oder Prüfling.**

Allgemeine Ausbildungsziele

Wie erreichen ?

1. Erziehung zu **Ehrlichkeit** und **realistischer Selbsteinschätzung**
2. Anregung zu realistischer **Fremdeinschätzung** von Personen, Firmen und Organisationen

Fallbeispiele und Anekdoten aus erster Hand erzählen, diskutieren und auswerten.

Allgemeine Ausbildungsziele — Wie erreichen ?

1. Erziehung zu **Ehrlichkeit** und **realistischer Selbsteinschätzung**
2. Anregung zu realistischer **Fremdeinschätzung** von Personen, Firmen und Organisationen
3. **Sicherheits- und Datenschutzbedürfnisse** ermitteln
   - Realistische Schutzziele
   - Realistische Angreifermodelle / Vertrauensmodelle

Fallbeispiele (und Anekdoten) aus erster Hand erzählen, diskutieren und auswerten.

Szenarien erarbeiten und diskutieren lassen.

Allgemeine Ausbildungsziele — Wie erreichen ?

1. Erziehung zu **Ehrlichkeit** und **realistischer Selbsteinschätzung**
2. Anregung zu realistischer **Fremdeinschätzung** von Personen, Firmen und Organisationen
3. **Sicherheits- und Datenschutzbedürfnisse** ermitteln
   - Realistische Schutzziele
   - Realistische Angreifermodelle / Vertrauensmodelle
4. **Validierung** und **Verifikation**, inkl. prinzipielle und praktische **Grenzen**

Fallbeispiele erarbeiten und diskutieren.

Anekdoten!

Allgemeine Ausbildungsziele — Wie erreichen ?

1. Erziehung zu **Ehrlichkeit** und **realistischer Selbsteinschätzung**
2. Anregung zu realistischer **Fremdeinschätzung** von Personen, Firmen und Organisationen
3. **Sicherheits- und Datenschutzbedürfnisse** ermitteln
   - Realistische Schutzziele
   - Realistische Angreifermodelle / Vertrauensmodelle
4. **Validierung** und **Verifikation**, inkl. prinzipielle und praktische **Grenzen**
5. Sicherheits- und Datenschutz**mechanismen**
   - Kennen und verstehen sowie
   - Entwickeln können

Was in Übungen selbst erarbeitet werden kann, sollte nicht durch Vorlesungen vermittelt werden.

Ausbildungsziele: Angebote am Lehrstuhl

- **Wechselwirkungen** zwischen **IT-Systemen** und **Gesellschaft**, z.B. gegensätzliche Interessen der Beteiligten, Datenschutzprobleme, Verletzlichkeit ...
- **Grundsätzliche Sicherheitslücken** heutiger IT-Systeme verstehen
- Verstehen, was **Mehrseitige Sicherheit** bedeutet, wie sie beschrieben und erreicht werden kann
- Vertiefte Kenntnisse der wichtigen Tools für Sicherheit in verteilten Systemen: **Kryptographie** und **Steganographie**
- Vertiefte Kenntnisse in **fehlerfreier Übertragung und Wiedergabe**
- Grundkenntnisse in **Fehlertoleranz**
- Abwägungen bei der **Systemkonstruktion**: Aufwand vs. Leistung vs. Sicherheit
- Grundkenntnisse in den einschlägigen **gesetzlichen Regelungen**

Ausbildungsziele: Angebote an anderen Lehrstühlen

- Vertiefte Kenntnisse **Sicherheit in Betriebssystemen**
- **Verifikation** von Betriebssystemkernen
- Vertiefte Kenntnisse in **Fehlertoleranz**

Gliederung (1)

1 Einführung
- 1.1 Was sind Rechnernetze (verteilte offene Systeme)
- 1.2 Was bedeutet Sicherheit?
  - 1.2.1 Was ist zu schützen?
  - 1.2.2 Vor wem ist zu schützen?
  - 1.2.3 Wie und wodurch kann Sicherheit erreicht werden?
  - 1.2.4 Vorausschau auf Schutzmechanismen
  - 1.2.5 Angreifermodell
- 1.3 Was bedeutet Sicherheit in Rechnernetzen?

2 Sicherheit in einzelnen Rechnern und ihre Grenzen
- 2.1 Physische Sicherheitsannahmen
  - 2.1.1 Was kann man bestenfalls erwarten?
  - 2.1.2 Gestaltung von Schutzmaßnahmen
  - 2.1.3 Ein Negativbeispiel: Chipkarten
  - 2.1.4 Sinnvolle physische Sicherheitsannahmen
- 2.2 Schutz isolierter Rechner vor unautorisiertem Zugriff und Computerviren
  - 2.2.1 Identifikation
  - 2.2.2 Zugangskontrolle
  - 2.2.3 Zugriffskontrolle
  - 2.2.4 Beschränkung der Bedrohung „Computer-Viren“ auf die durch „transitive Trojanische Pferde“
  - 2.2.5 Restprobleme

Gliederung (2)

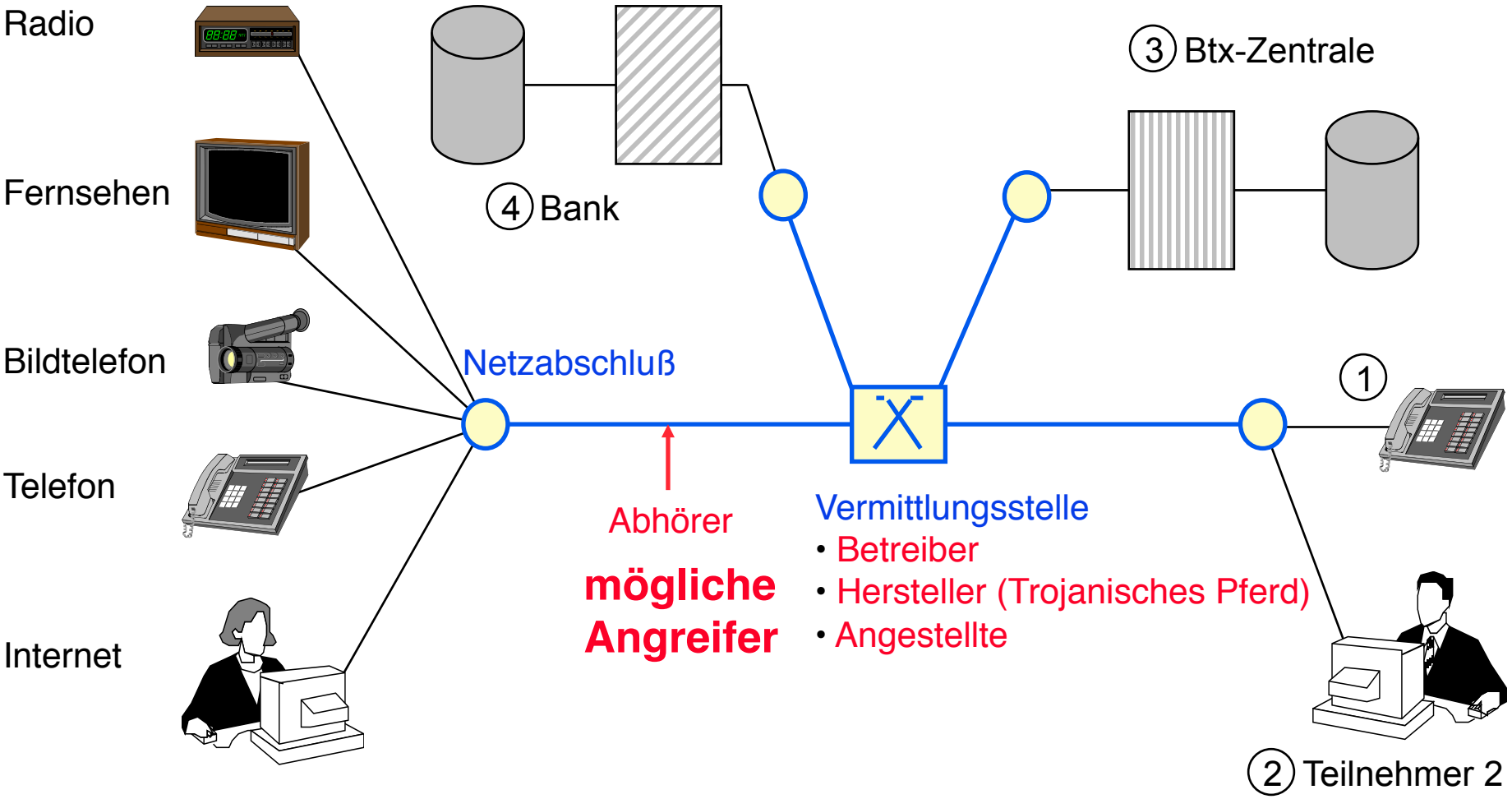
3 Kryptologische Grundlagen

4 Datenschutz garantierende Kommunikationsnetze

5 Digitale Zahlungssysteme und Credentials als Verallgemeinerung

6 Zusammenfassung und Ausblick

Ausschnitt eines Rechnernetzes



Bsp. ⑤ Patientenüberwachung, ⑥ Bewegtbildüberwachung während Operation

Warum reichen juristische Regelungen (für Rechtssicherheit und Datenschutz) nicht aus ?

Geschichte der Rechnernetze (1)

1833 erster **elektromagnetischer Telegraph**

1858 erste **Kabelverbindung zwischen Europa und Nordamerika**

1876 **Fernsprechen** über 8,5 km lange Versuchsstrecke

1881 erstes **Fernsprechortsnetz**

1900 Beginn der **drahtlosen Telegraphie**

1906 Einführung des **Selbstwählferndienstes** in Deutschland, realisiert durch Hebdrehwähler, d.h. erste vollautomatische Vermittlung durch Elektomechanik

1928 Fernsprechdienst Deutschland-USA eingeführt (über Funk)

1949 erster funktionierender **von-Neumann-Rechner**

1956 erstes **Transatlantikkabel für Fernsprechen**

1960 erster **Fernmeldesatellit**

1967 Beginn des Betriebes des **Datex-Netzes** durch die deutsche Bundespost, d.h. des ersten speziell für Rechnerkommunikation realisierten Kommunikationsnetzes (Rechnernetz erster Art). Die Übertragung erfolgt digital, die Vermittlung durch Rechner (Rechnernetz zweiter Art).

1977 Einführung des Elektronischen Wähl-Systems **(EWS)** für Fernsprechen durch die Deutsche Bundespost, d.h. erstmals Vermittlung durch Rechner (Rechnernetz zweiter Art) im Fernsprechnetz, aber weiterhin analoge Übertragung

Geschichte der Rechnernetze (2)

1981 erster persönlicher Rechner (PC) der Rechnerfamilie (**IBM PC**), die weite Verbreitung auch im privaten Bereich findet

1982 Investitionen in die **Übertragungssysteme** des Fernsprechnetzes erfolgen zunehmend in **digitale** Technik

1985 Investitionen in die Vermittlungssysteme des Fernsprechnetzes erfolgen zunehmend in rechnergesteuerte Technik, die nunmehr nicht mehr analoge, sondern **digitale Signale vermittelt** (in Deutschland 1998 abgeschlossen)

1988 Betriebsbeginn des **ISDN** (Integrated Services Digital Network)

1989 erster westentaschengroßer PC: **Atari Portfolio**; damit sind Rechner im engeren Sinne persönlich und mobil

1993 **zellulare Funknetze** werden Massendienst

1994 **www** Kommerzialisierung des Internet

2000 **WAP-fähige Handys** für 77 € ohne Vertragsbindung

2003 mit IEEE 802.11b finden **WLAN** (Wireless Local Area Network), mit Bluetooth **WPAN** (Wireless Personal Area Network) massenhafte Verbreitung

2005 **VoIP** (Voice over IP) wird Massendienst

Wichtige Begriffe

Rechner verbunden über **Kommunikationsnetz** = **Rechnernetz** (erster Art)

Prozeßrechner im **Kommunikationsnetz** = **Rechnernetz** (zweiter Art)

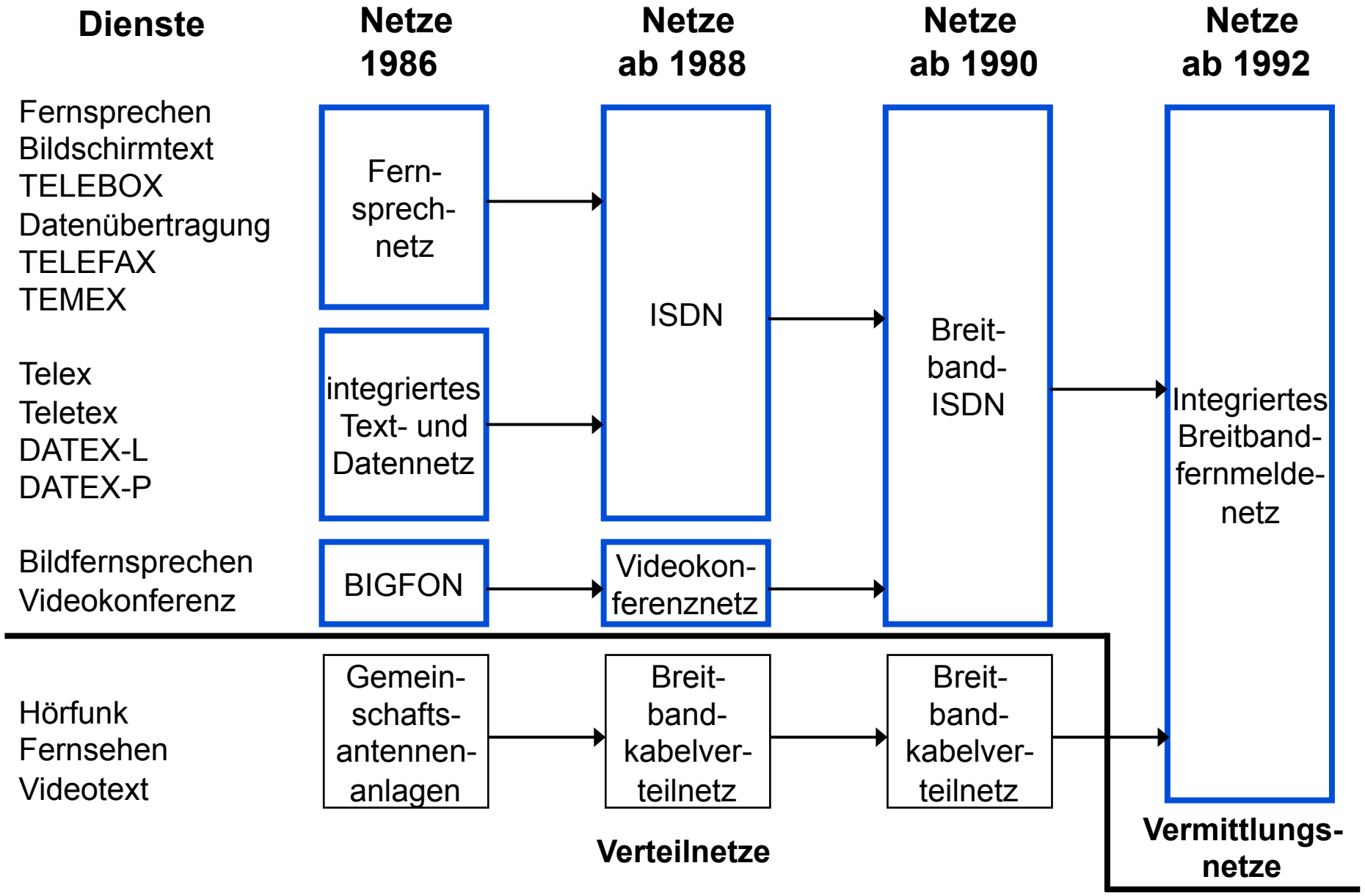
verteiltes System
räumlich
Kontroll- und Implementierungsstruktur

offenes System ≠ **öffentliches** System ≠ **Open Source** System

diensteintegrierendes System

digitales System

Entwicklung der leitungsgebundenen Kommunikationsnetze der Deutschen Bundespost



Bedrohungen und korrespondierende Schutzziele

Bedrohungen: Bsp.: medizinisches Informationssystem Schutzziele:

1) Informationsgewinn — Vertraulichkeit
   Rechnerhersteller erhält Krankengeschichten

2) Modifikation von Information
   unerkannt Dosierungsanweisungen ändern

3) Beeinträchtigung der Funktionalität
   erkennbar ausgefallen

≥ totale Korrektheit:
- Integrität ≅ partielle Korrektheit
- Verfügbarkeit für berechtigte Nutzer

keine Klassifikation, aber pragmatisch sinnvoll

Bsp.: Programm unbefugt modifiziert

1) nicht erkennbar, aber verhinderbar; nicht rückgängig zu machen
2)+3) nicht verhinderbar, aber erkennbar; rückgängig zu machen

Definitionen für die Schutzziele

Vertraulichkeit (confidentiality)

Informationen werden nur Berechtigten bekannt.

Integrität (integrity)

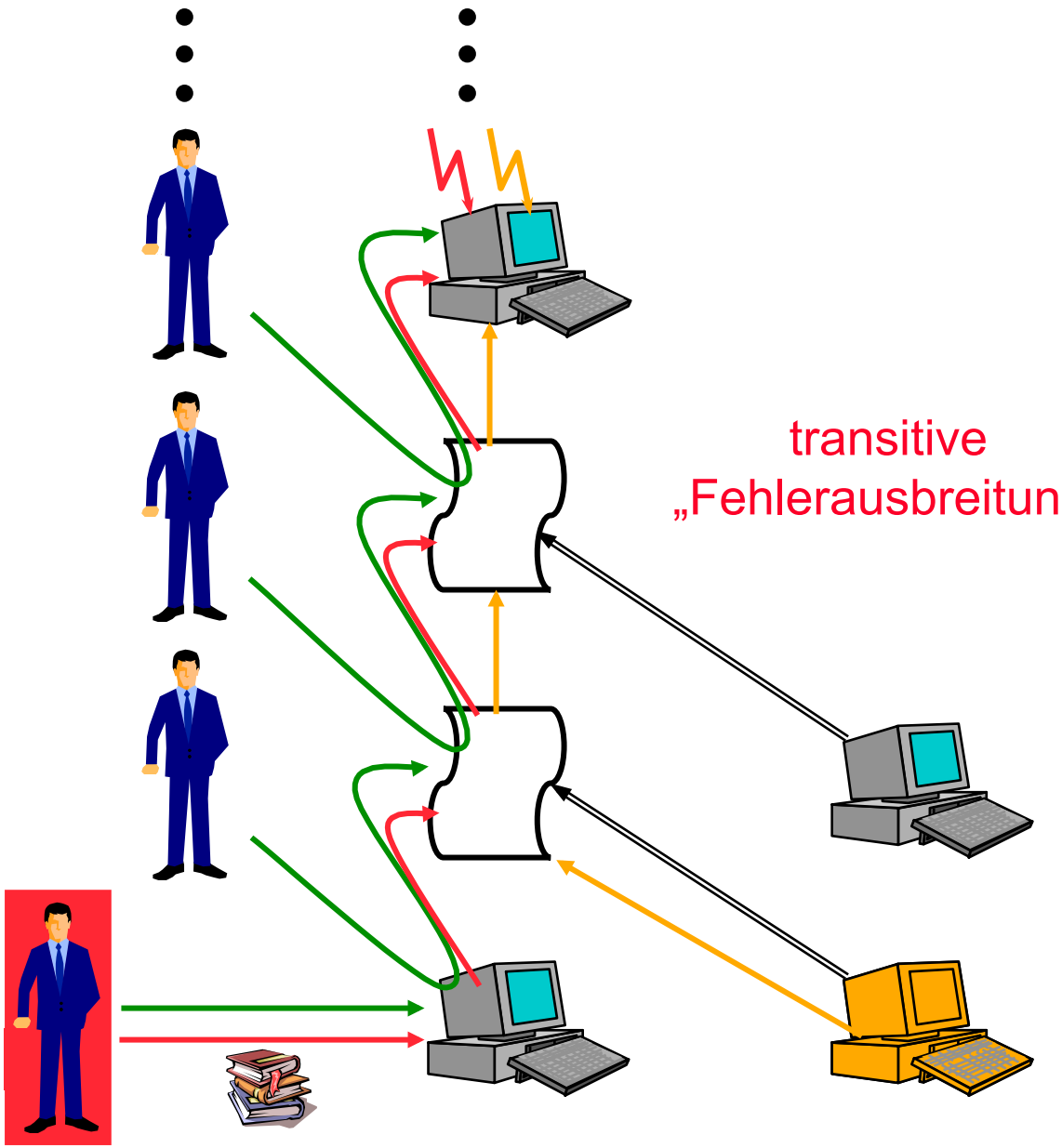
Informationen sind richtig, vollständig und aktuell oder aber dies ist erkennbar nicht der Fall.

Verfügbarkeit (availability)

Informationen sind dort und dann zugänglich, wo und wann sie von Berechtigten gebraucht werden.

- **subsumiert: Daten, Programme, Hardwarestrukturen**
- **es muss geklärt sein, wer in welcher Situation wozu berechtigt ist**
- **kann sich nur auf das Innere eines Systems beziehen**

Transitive Ausbreitung von Fehlern und Angriffen



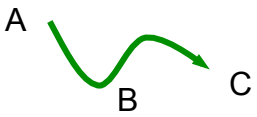
transitive
„Fehlerausbreitung“

Symbolerklärungen

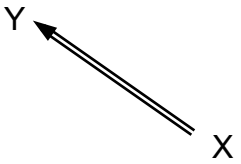
Rechner

Programm

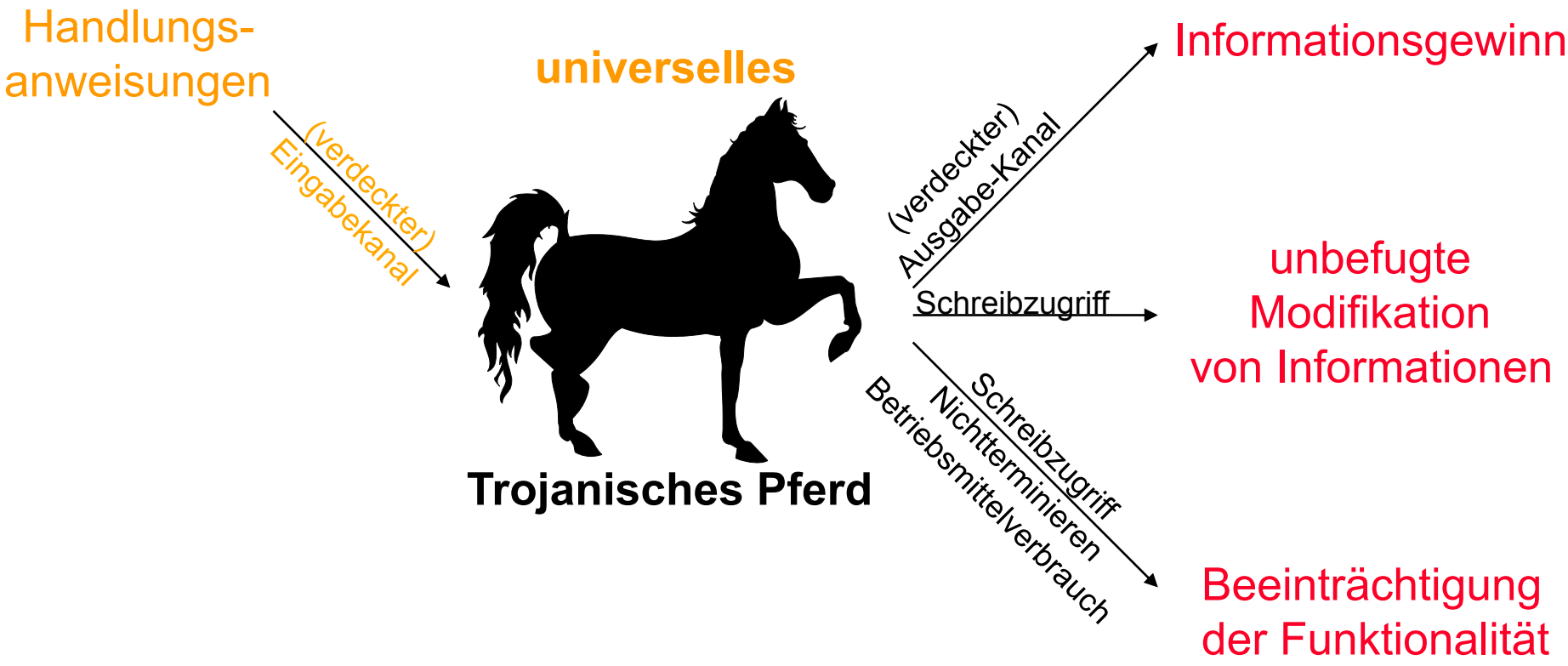
A benutzt B, um
C zu entwerfen



Maschine X führt
Programm Y aus



Universelles Trojanisches Pferd



Vor wem ist zu schützen ?

Naturgesetze und Naturgewalten

- Bauteile altern
- Überspannung (Blitzschlag, EMP)
- Spannungsausfall
- Überschwemmung (Sturmflut, Wasserrohrbruch)
- Temperaturänderungen ...

} Fehler-toleranz

Menschen

- Außenstehende
- Benutzer des Systems
- Betreiber des Systems
- Wartungsdienst
- Produzenten des Systems
- Entwerfer des Systems
- Produzenten der Entwurfs- und Produktionshilfsmittel
- Entwerfer der Entwurfs- und Produktionshilfsmittel
- Produzenten der Entwurfs- und Produktionshilfsmittel der Entwurfs- und Produktionshilfsmittel
- Entwerfer ... jeweils auch Benutzer, Betreiber, Wartungsdienst ... des verwendeten Systems

} Trojanisches Pferd
- universell
- transitiv

Welche Schutzmaßnahmen gegen welche Angreifer

Schutz bzgl. / Schutz vor	**Erwünschtes leisten**	**Unerwünschtes verhindern**
Entwerfer und Produzent der Entwurfs- und Produktionshilfsmittel	Zwischensprachen; Zwischenergebnisse, die unabhängig analysiert werden	
Entwerfer des Systems	wie oben + mehrere unabhängige Entwerfer	
Produzenten des Systems	unabhängige Analysen der Produkte	
Wartungsdienst	Kontrolle wie bei neuem Produkt, s. o.	
Betreiber des Systems		physischen Zugriff beschränken, logischen Zugriff beschränken und protokollieren
Benutzer des Systems	physischen und logischen Zugriff beschränken	
Außenstehende	physisch vom System, kryptographisch von den Daten fernhalten	

Welche Schutzmaßnahmen gegen welche Angreifer

Schutz bzgl. / Schutz vor	**Erwünschtes leisten**	**Unerwünschtes verhindern**
Entwerfer und Produzent der Entwurfs- und Produktionshilfsmittel	Zwischensprachen; Zwischenergebnisse, die unabhängig analysiert werden	
Entwerfer des Systems	wie oben + mehrere unabhängige Entwerfer	
Produzenten des Systems	unabhängige Analysen der Produkte	
Wartungsdienst	Kontrolle wie bei neuem Produkt, s. o.	
Betreiber des Systems		physischen Zugriff beschränken, logischen Zugriff beschränken und protokollieren
Benutzer des Systems	physischen und logischen Zugriff beschränken	
Außenstehende	physisch vom System, kryptographisch von den Daten fernhalten	

physische Verteilung und Redundanz

Unbeobachtbarkeit, Anonymität, Unverkettbarkeit:
Erfassungsmöglichkeit „unnötiger Daten“ vermeiden

Maximal berücksichtigte Stärke eines Angreifers

Angreifermodell

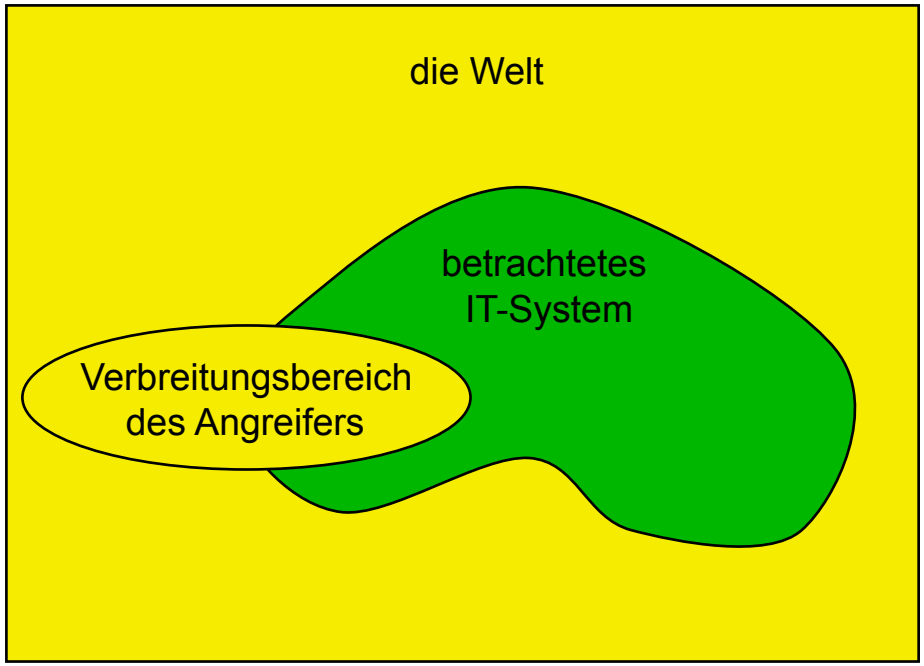
Schutz vor einem allmächtigen Angreifer ist unmöglich.

- Rollen des Angreifers (Außenstehender, Benutzer, Betreiber, Wartungsdienst, Produzent, Entwerfer …), *auch kombiniert*
- Verbreitung des Angreifers
- Verhalten des Angreifers
  - passiv / aktiv
  - beobachtend / verändernd (bzgl. seiner erlaubten Handlungen)
- dumm / intelligent
  - Rechenkapazität:
    - unbeschränkt: informationstheoretisch
    - beschränkt: komplexitätstheoretisch

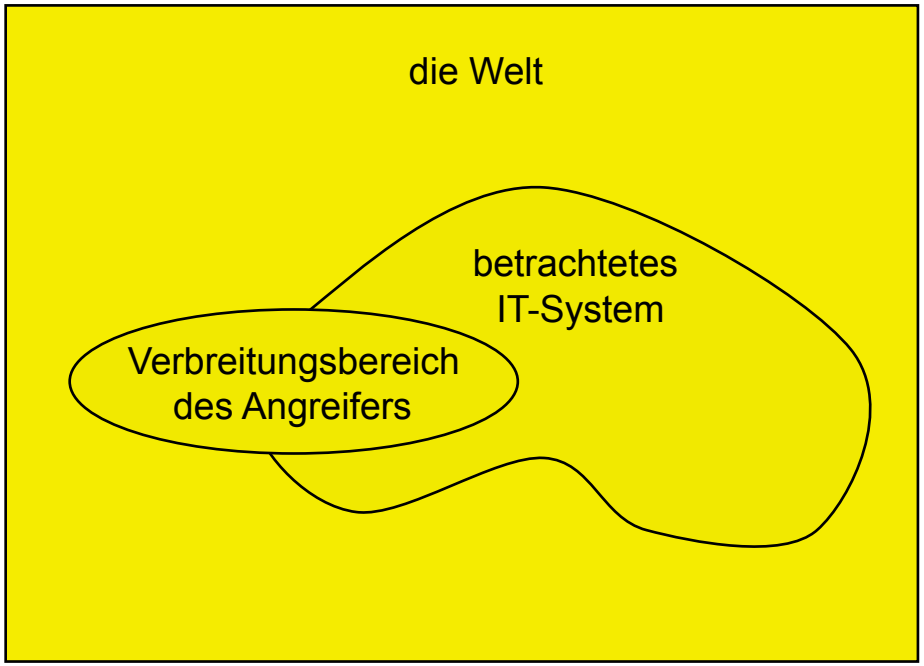
Geld

Zeit

Beobachtender vs. verändernder Angreifer



beobachtender Angreifer



verändernder Angreifer

nur erlaubtes Verhalten

auch verbotenes Verhalten

Stärke eines Angreifer(modell)s

Angreifer(modell) *A* ist stärker als Angreifer(modell) *B*, gdw. *A* in mindestens einer Hinsicht stärker ist als *B* und in keiner Hinsicht schwächer.

Stärker bedeutet:

- Menge der Rollen von *A* ⊃ Menge der Rollen von *B*,
- Verbreitung von *A* ⊃ Verbreitung von *B*,
- Verhalten des Angreifers
  - aktiv ist stärker als passiv
  - verändernd ist stärker als beobachtend
- intelligent ist stärker als dumm
  - Rechenkapazität: unbeschränkt ist stärker als beschränkt
- mehr Geld bedeutet stärker
- mehr Zeit bedeutet stärker

Definiert partielle Ordnung auf Angreifer(modelle)n.

Sicherheit in Rechnernetzen

Vertraulichkeit

- Nachrichteninhalte vertraulich — **Ende-zu-Ende-Verschlüsselung mit Konzelationssystem**
- **Ort**
  - Sender / Empfänger anonym — **Verfahren zum Schutz der Verkehrsdaten**

Integrität

- Fälschungen erkennen — **Authentikationssystem(e)**
- Empfänger kann Senden der Nachricht beweisen — **Nachrichten signieren**
- **Zeit**
  - Absender kann Senden beweis. — **Empfangsquittung**
- Nutzungsentgelte sichern — **während Diensterbringung mittels digitaler Zahlungssysteme**

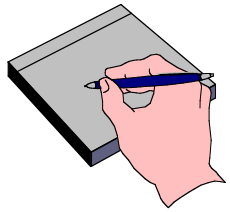
Verfügbarkeit

- Kommunikation ermöglichen — **Diversitäre Netze; faire Betriebsmittelaufteilung**

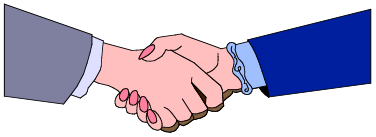
Mehrseitige Sicherheit

- Jeder Beteiligte hat eigene Sicherheitsinteressen.

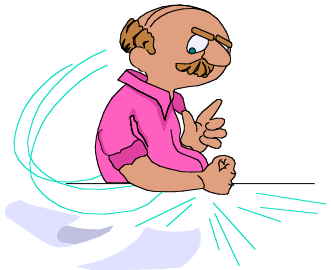
- Jeder Beteiligte kann seine Sicherheitsinteressen formulieren.



- Konflikte werden erkannt und Lösungen ausgehandelt.



- Jeder Beteiligte kann seine Sicherheitsinteressen in den ausgehandelten Lösungen durchsetzen.

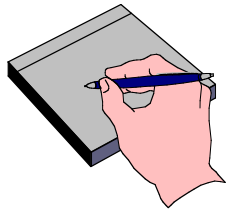


Sicherheit mit minimalen Annahmen über andere

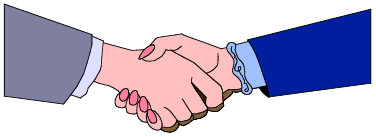
Mehrseitige Sicherheit (2. Version)

- Jeder Beteiligte hat eigene Interessen.

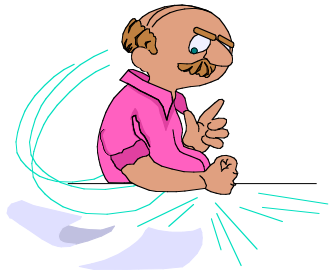
- Jeder Beteiligte kann seine Sicherheitsinteressen formulieren.



- Konflikte werden erkannt und Lösungen ausgehandelt.



- Jeder Beteiligte kann seine Sicherheitsinteressen in den ausgehandelten Lösungen durchsetzen.

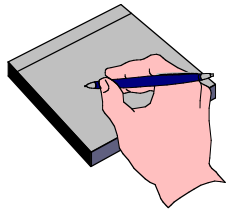


Sicherheit mit minimalen Annahmen über andere

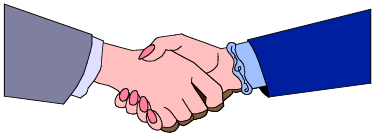
Mehrseitige Sicherheit (3. Version)

- Jeder Beteiligte hat eigene Interessen.

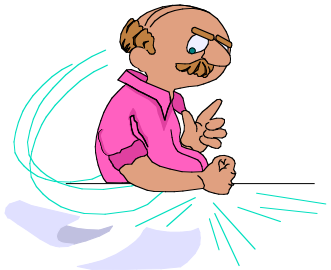
- Jeder Beteiligte kann seine Sicherheitsinteressen formulieren.



- Konflikte werden erkannt und Lösungen ausgehandelt.



- Jeder Beteiligte kann seine Sicherheitsinteressen in den ausgehandelten Lösungen durchsetzen. Grenzen der Durchsetzbarkeit betreffen alle Beteiligten in gleicher Weise.



Sicherheit mit minimalen Annahmen über andere

Schutzziele: Sortierung

	Inhalte	Umfeld
Unerwünschtes verhindern	Vertraulichkeit Verdecktheit	Anonymität Unbeobachtbarkeit
Erwünschtes leisten	Integrität	Zurechenbarkeit
	Verfügbarkeit	Erreichbarkeit Verbindlichkeit

Schutzziele: Definitionen

Vertraulichkeit: Geheimhaltung von Daten während der Übertragung. Niemand außer den Kommunikationspartnern kann den Inhalt der Kommunikation erkennen.

Verdecktheit: Versteckte Übertragung von vertraulichen Daten. Niemand außer den Kommunikationspartnern kann die Existenz einer vertraulichen Kommunikation erkennen.

Anonymität: Nutzer können Ressourcen und Dienste benutzen, ohne ihre Identität zu offenbaren. Selbst der Kommunikationspartner erfährt nicht die Identität.

Unbeobachtbarkeit: Nutzer können Ressourcen und Dienste benutzen, ohne daß andere dies beobachten können. Dritte können weder das Senden noch den Erhalt von Nachrichten beobachten.

Integrität: Modifikationen der kommunizierten Inhalte (Absender eingeschlossen) werden durch den Empfänger erkannt.

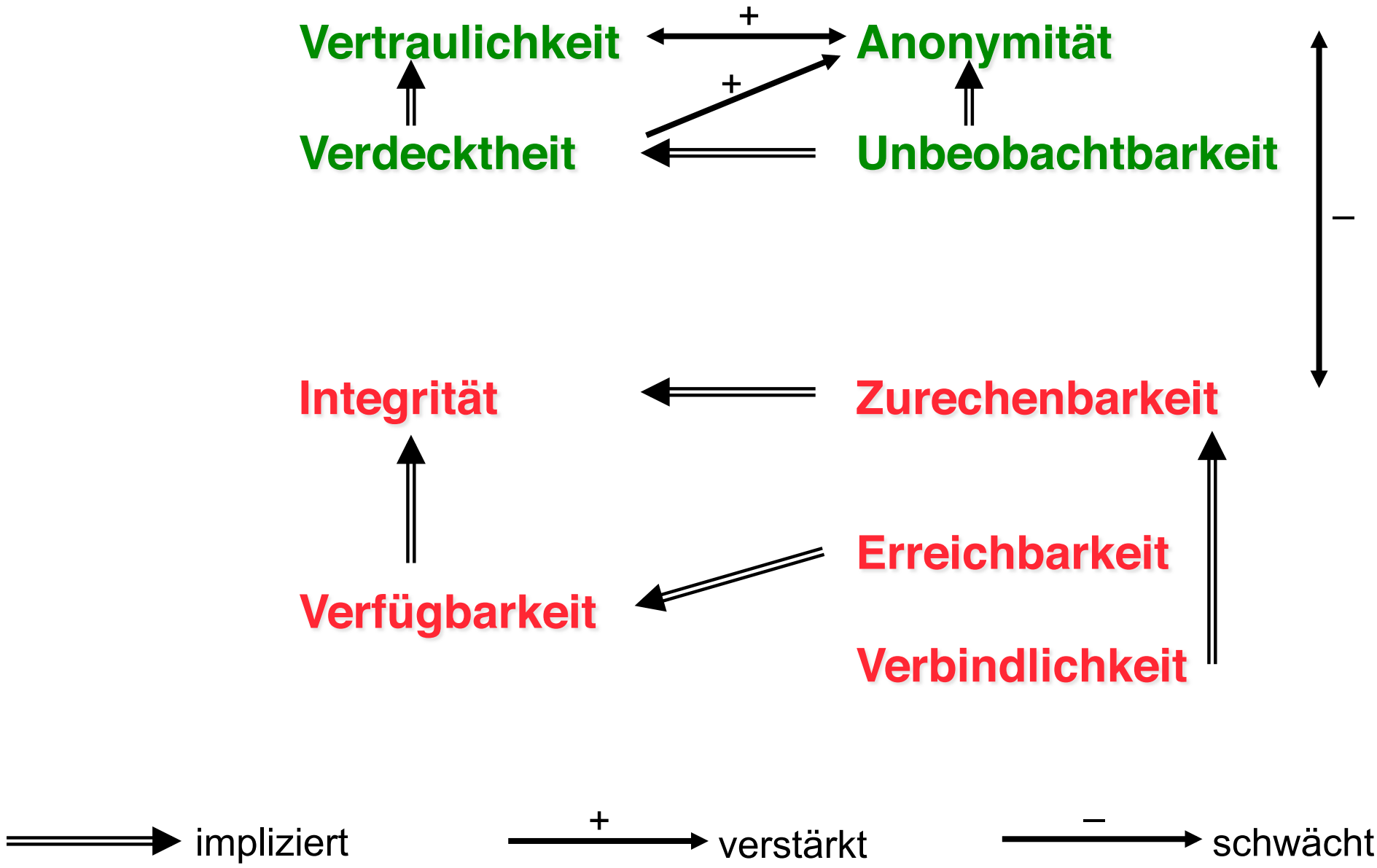
Zurechenbarkeit: Sendern bzw. Empfängern von Informationen kann das Senden bzw. der Empfang der Informationen bewiesen werden.

Verfügbarkeit: Nutzbarkeit von Diensten und Ressourcen, wenn ein Teilnehmer sie benutzen will.

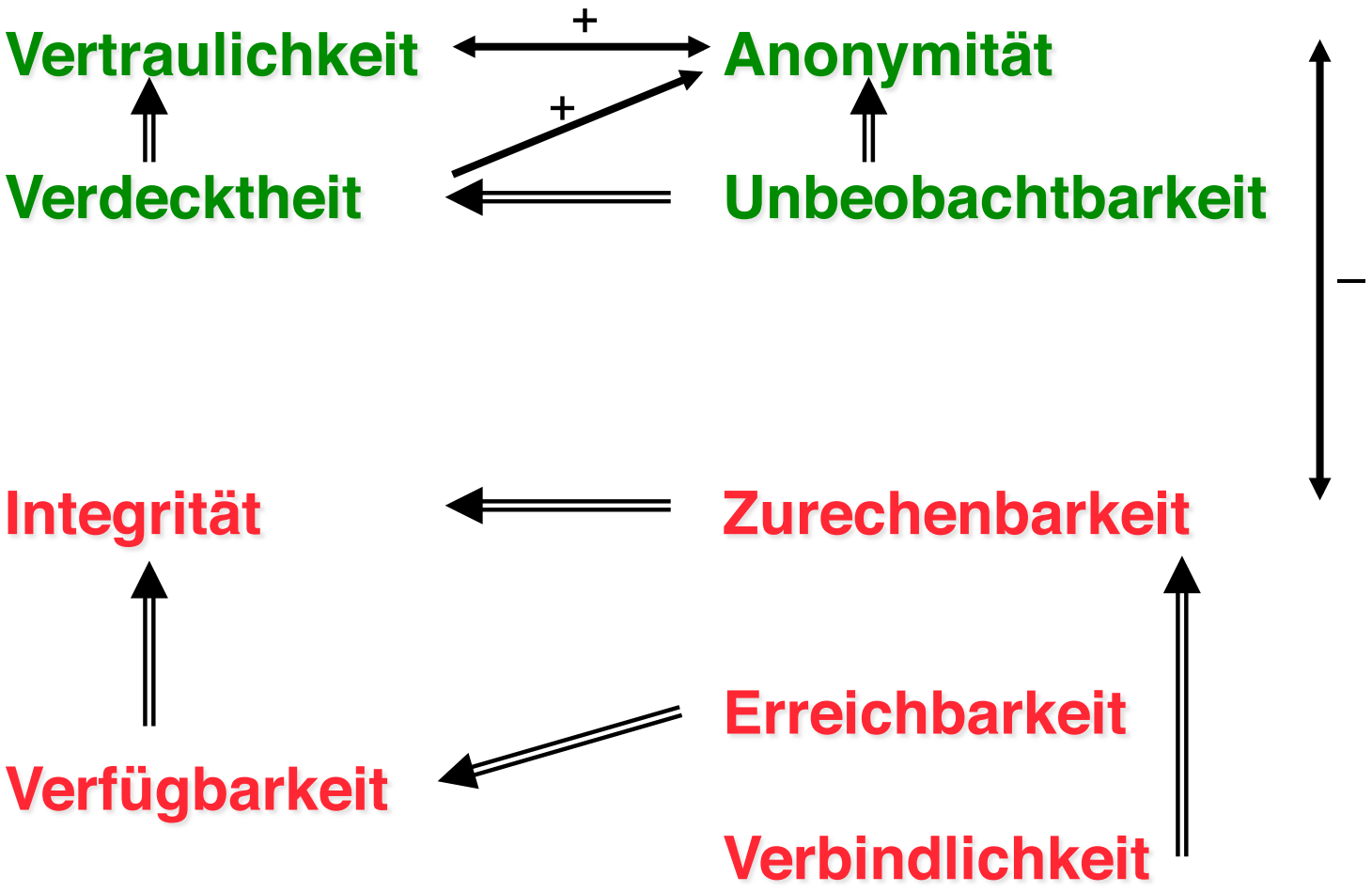
Erreichbarkeit: Zu einer Ressource (Nutzer oder Maschine) kann Kontakt aufgenommen werden, wenn gewünscht.

Verbindlichkeit: Ein Nutzer kann rechtlich belangt werden, um seine Verantwortlichkeiten innerhalb einer angemessenen Zeit zu erfüllen.

Wechselwirkungen zwischen Schutzzielen



Wechselwirkungen zwischen Schutzzielen



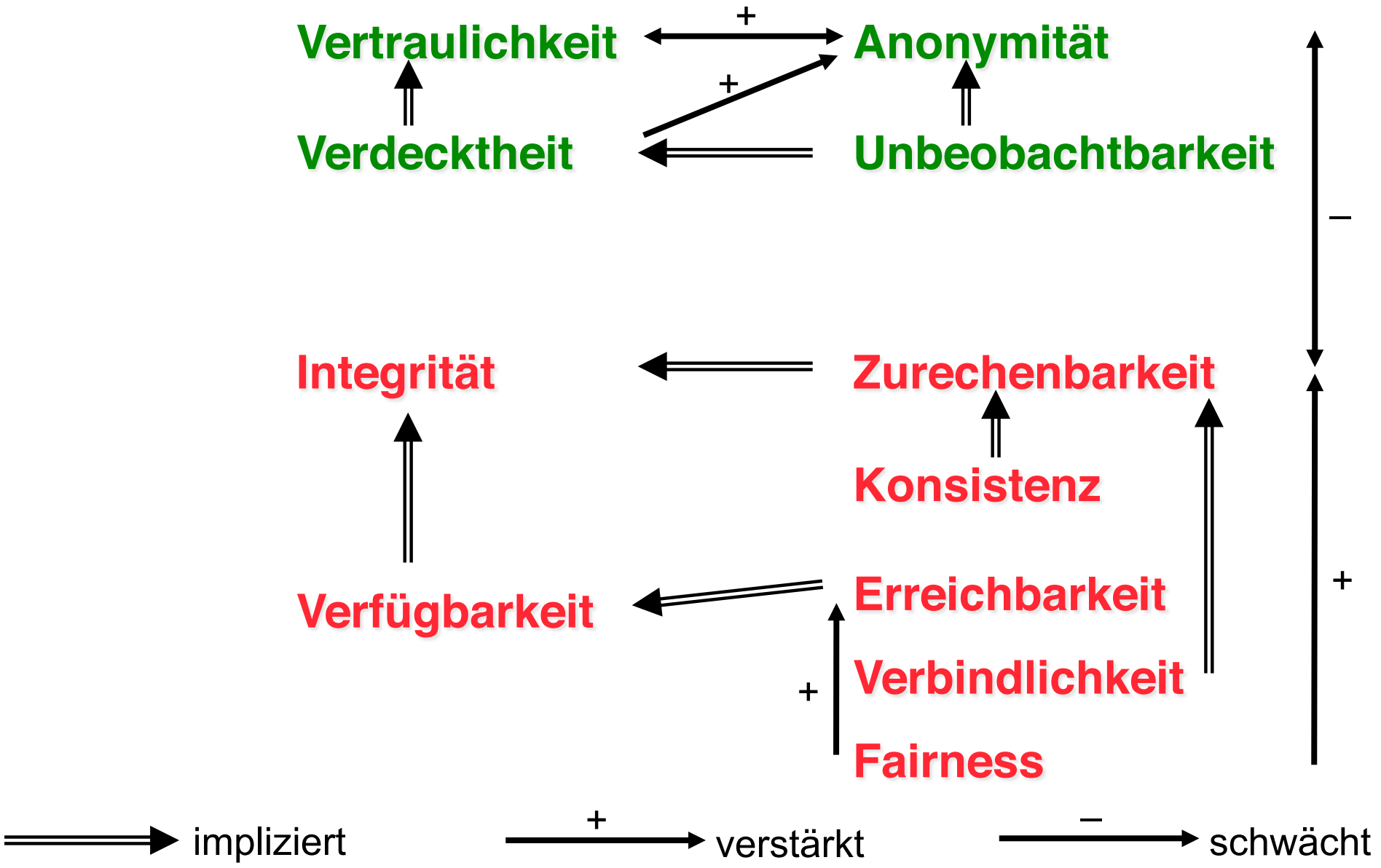
Transitive Hülle hinzufügen

⟹ impliziert

+→ verstärkt

–→ schwächt

Wechselwirkungen zwischen Schutzzielen, zwei zusätzliche



Physische Sicherheitsannahmen

Alle technischen Schutzmaßnahmen brauchen physische „Verankerung“ in einem Systemteil, auf den der Angreifer weder lesenden noch verändernden Zugriff hat.

Spektrum vom „Rechenzentrum X“ bis zur „Chipkarte Y“

Was kann man bestenfalls erwarten ?

Verfügbarkeit eines räumlich konzentrierten Systemteils ist gegen durchaus *vorstellbare* Angreifer nicht gewährleistbar

→ **physisch verteiltes System**

und hoffen, dass Angreifer nicht an vielen Orten gleichzeitig sein kann.

Verteilung erschwert **Vertraulichkeit** und **Integrität.**

Physische Maßnahmen bzgl. Vertraulichkeit und Integrität jedoch wirkungsvoller: Schutz gegen *alle* derzeit *vorstellbaren* Angreifer scheint erreichbar. Gelingt dies hinreichend, steht physischer Verteilung nichts im Wege.

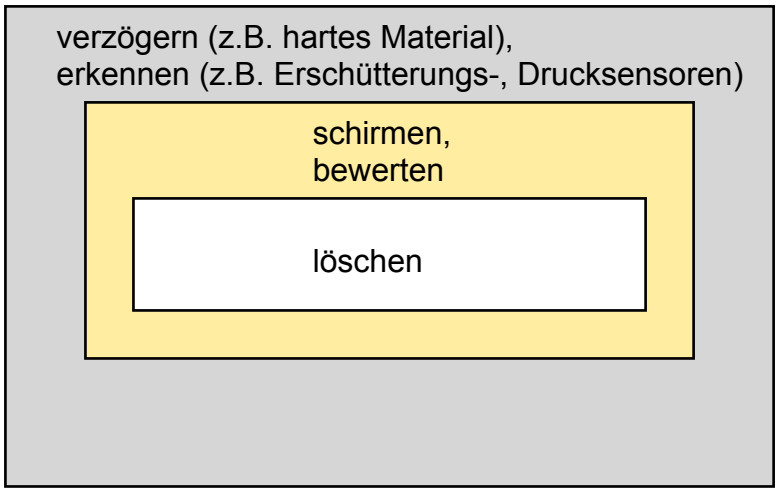
Unmanipulierbare Gehäuse

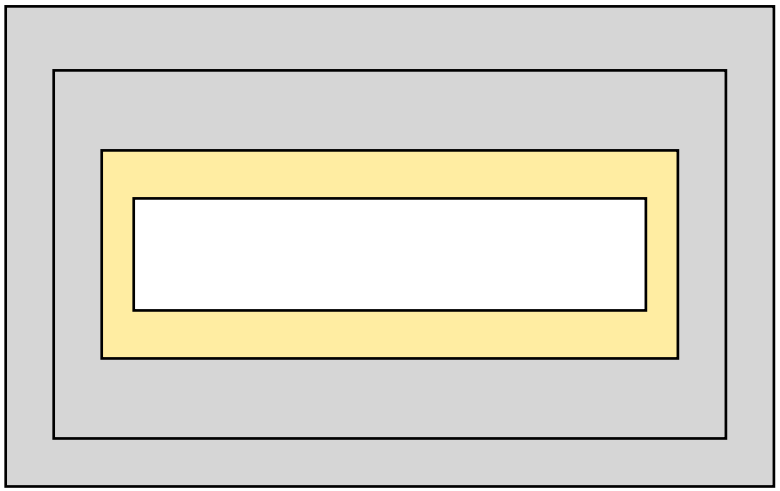
Eingriff: Erkennen
Bewerten

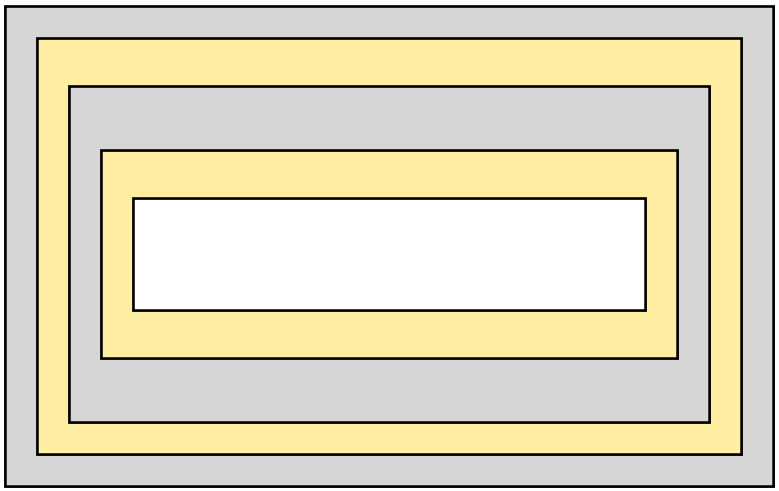
Angriff: Verzögern
Daten (etc.) löschen

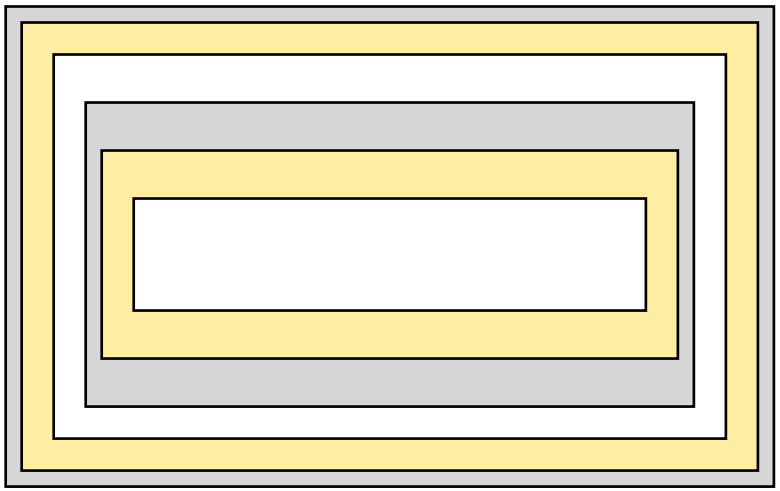
Möglichkeit: mehrere Schichten, Schirmung 

Schalenförmige Anordnung der fünf Grundfunktionen









Unmanipulierbare Gehäuse

Eingriff: Erkennen
Bewerten

Angriff: Verzögern
Daten (etc.) löschen

Möglichkeit: mehrere Schichten, Schirmung

Problem: Validierung ... Glaubwürdigkeit

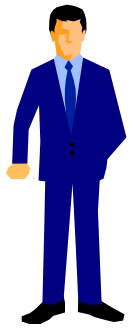
Negativ-Beispiel: Chipkarten

- kein Erkennen (u.a. Batterie fehlt)
- Schirmung schwierig (Karte dünn und biegbar)
- kein Löschen vorgesehen selbst bei Strom-versorgung

Goldene Regel

Übereinstimmung zwischen organisatorischen und informationstechnischen Strukturen

Identifikation von Menschen durch IT-Systeme



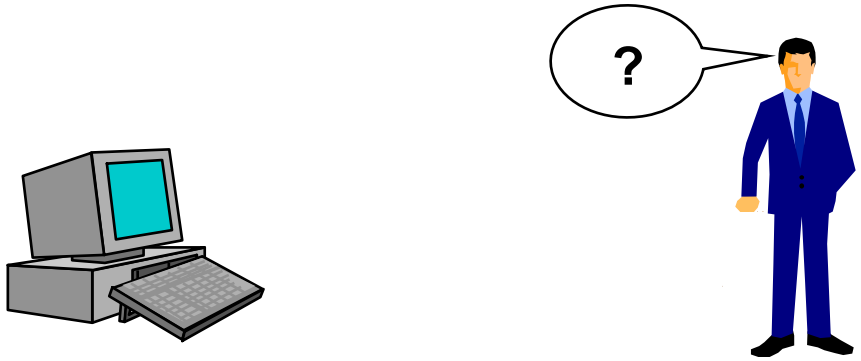
Was man *ist*
- Handgeometrie
- Fingerabdruck
- Aussehen
- eigenhändige Unterschrift
- Retina-Muster
- Stimme
- Tipp-Charakteristik

hat
- Papierdokument
- Metallschlüssel
- Magnetstreifenkarte
- Chipkarte
- Taschenrechner

weiß
- Passwort, Passphrase
- Antworten auf Fragen
- Rechnerergebnisse für Zahlen

→ **Ausweis**

Identifikation von IT-Systemen durch Menschen

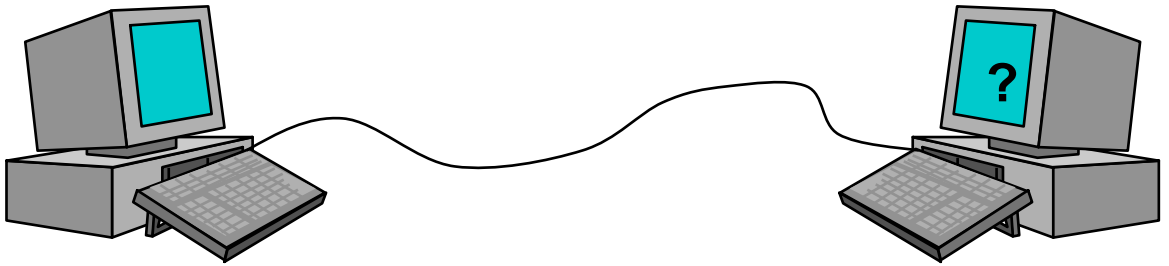


Was es *ist*
- Gehäuse
- Siegel, Hologramm
- Verschmutzung

weiß
- Passwort
- Antworten auf Fragen
- Rechnerergebnisse für Zahlen

Wo es *steht*

Identifikation von IT-Systemen durch IT-Systeme

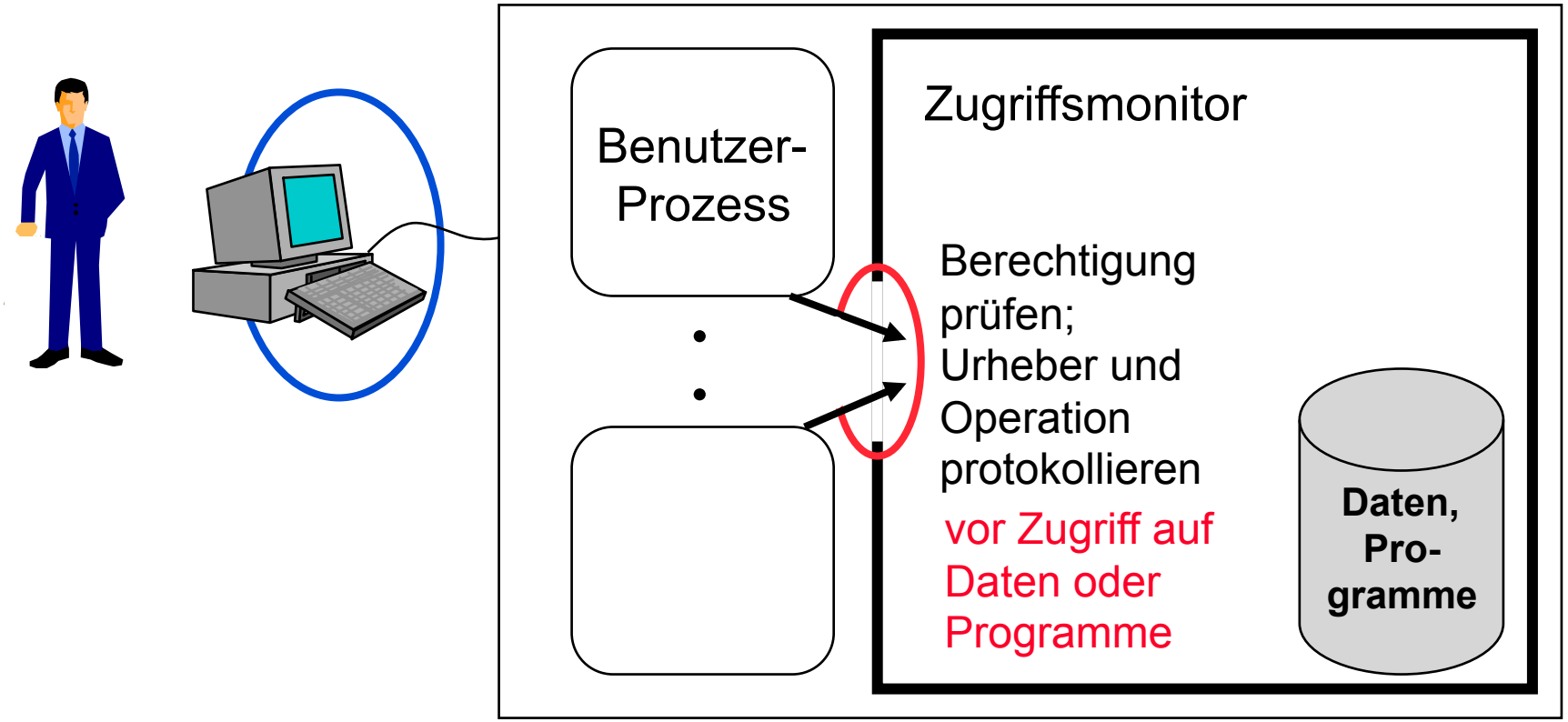


Was es *weiß*
- Passwort
- Antworten auf Fragen
- Rechnerergebnisse für Zahlen
- **Kryptographie**

Leitung *woher*

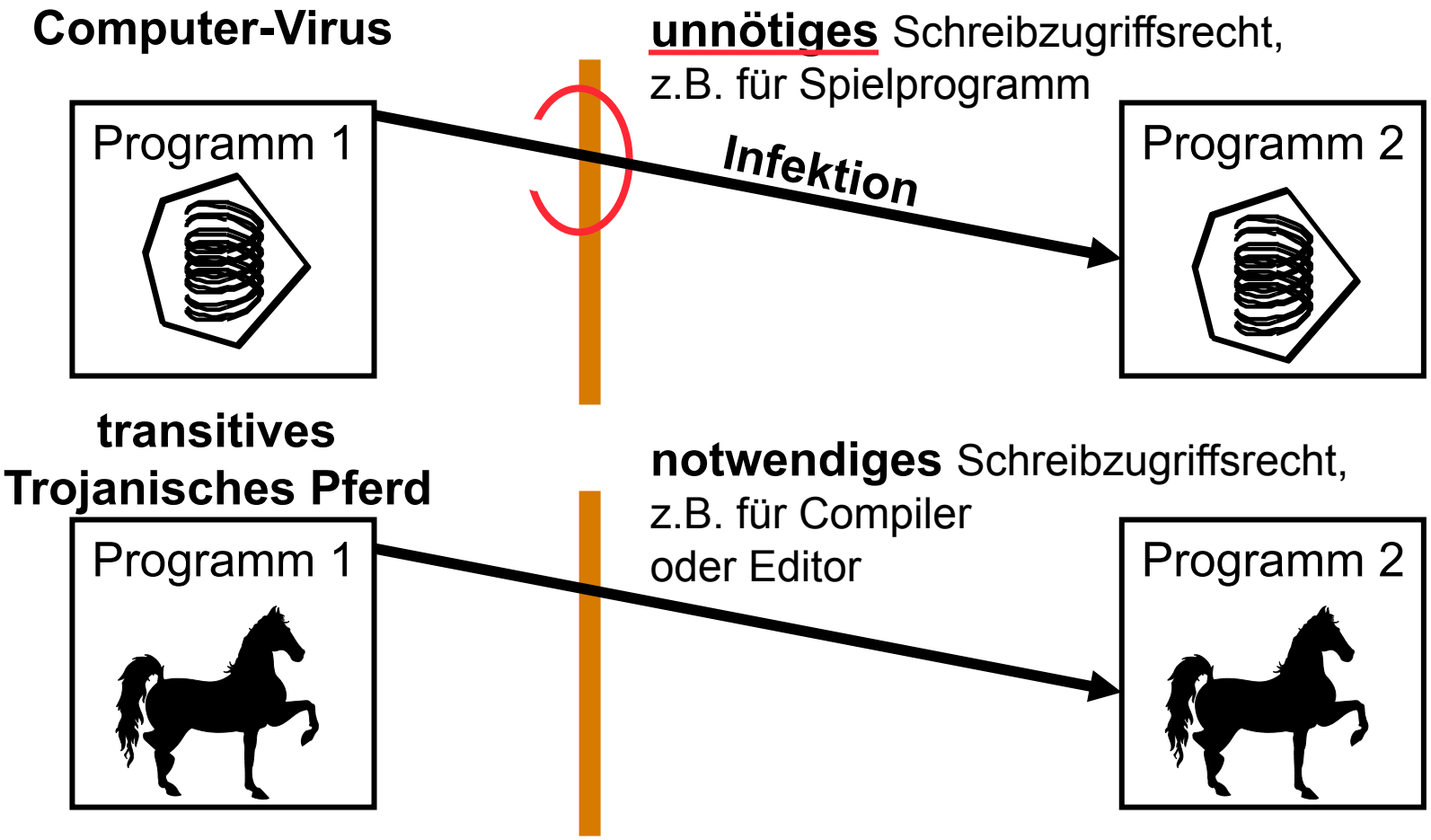
Zugangs- und Zugriffskontrolle

Zugangskontrolle nur mit berechtigten Partnern kommunizieren



Zugriffskontrolle Subjekt kann Operationen auf Objekt nur ausführen, wenn es ein Recht dazu hat.

Computer-Virus vs. Transitives Trojanisches Pferd



Zugriffskontrolle

Beschränkung der Angriffsausbreitung durch geringstmögliche Privilegierung:

Keine unnötigen Zugriffsrechte gewähren !

➡ Keine Computer-Viren, nur noch transitive trojanische Pferde !

Grundsätzliches zu Computer-Viren und Troj. Pferden

Andere Maßnahmen versagen:

1. Nicht entscheidbar, ob Programm ein Computer-Virus ist
   Beweis (ind.) Annahme decide (•)

```
program Gegenbeispiel
if decide (Gegenbeispiel) then  keine_Virusfkt
                                else Virusfkt
```

2. Nicht entscheidbar, ob Programm ein Trojanisches Pferd ist

Also: Besser zu vorsichtig!

3. Selbst bekannte Computer-Viren nicht wirksam erkennbar
   Selbstmodifikation → ~~Viren-Scanner~~

4. dito Trojanische Pferde

5. Schaden bzgl. Daten hinterher nicht ermittelbar
   Schadensfkt. könnte sich selbst modifizieren

Restprobleme

1. Genau spezifizieren, was IT-System tun und *unterlassen* soll. ?
2. *Totale Korrektheit* der Implementierung nachweisen. **heute** ?
3. Alle *verdeckten Kanäle* erkannt ? ?

Goldene Regel

IT-System so als *verteiltes* System entwerfen und realisieren, dass begrenzt viele angreifende Rechner keinen wesentlichen Schaden anrichten können.

Verteiltes System

Aspekte von Verteiltheit

räumliche Verteiltheit
verteilte Kontroll- und Implementierungsstruktur

verteiltes System:

keine Instanz hat globale Systemsicht

Sicherheit in verteilten Systemen

Vertrauenswürdige Endgeräte

vertrauenswürdig	nur für Benutzer auch für andere

Kommunikationsfähigkeit

Verfügbarkeit durch	Redundanz und Diversität

Kryptographie

Vertraulichkeit durch	Verschlüsselung
Integrität durch	MACs oder digitale Signaturen

Verfügbarkeit

Infrastruktur mit geringstmöglicher Entwurfskomplexität

Anschluß an vollständig diversitäre Netze

- unterschiedliche Frequenzbänder bei Funk
- unterschiedliche Leitungsführung bei leitungsgebundenen Netzen

Diversitätsengpässe vermeiden

- z.B. Funknetz benötigt gleiche OVSt,
- für alle Anschlußleitungen gibt es nur einen Übergangspunkt ins Fernnetz

Kryptologische Grundlagen

erreichbare Schutzziele:

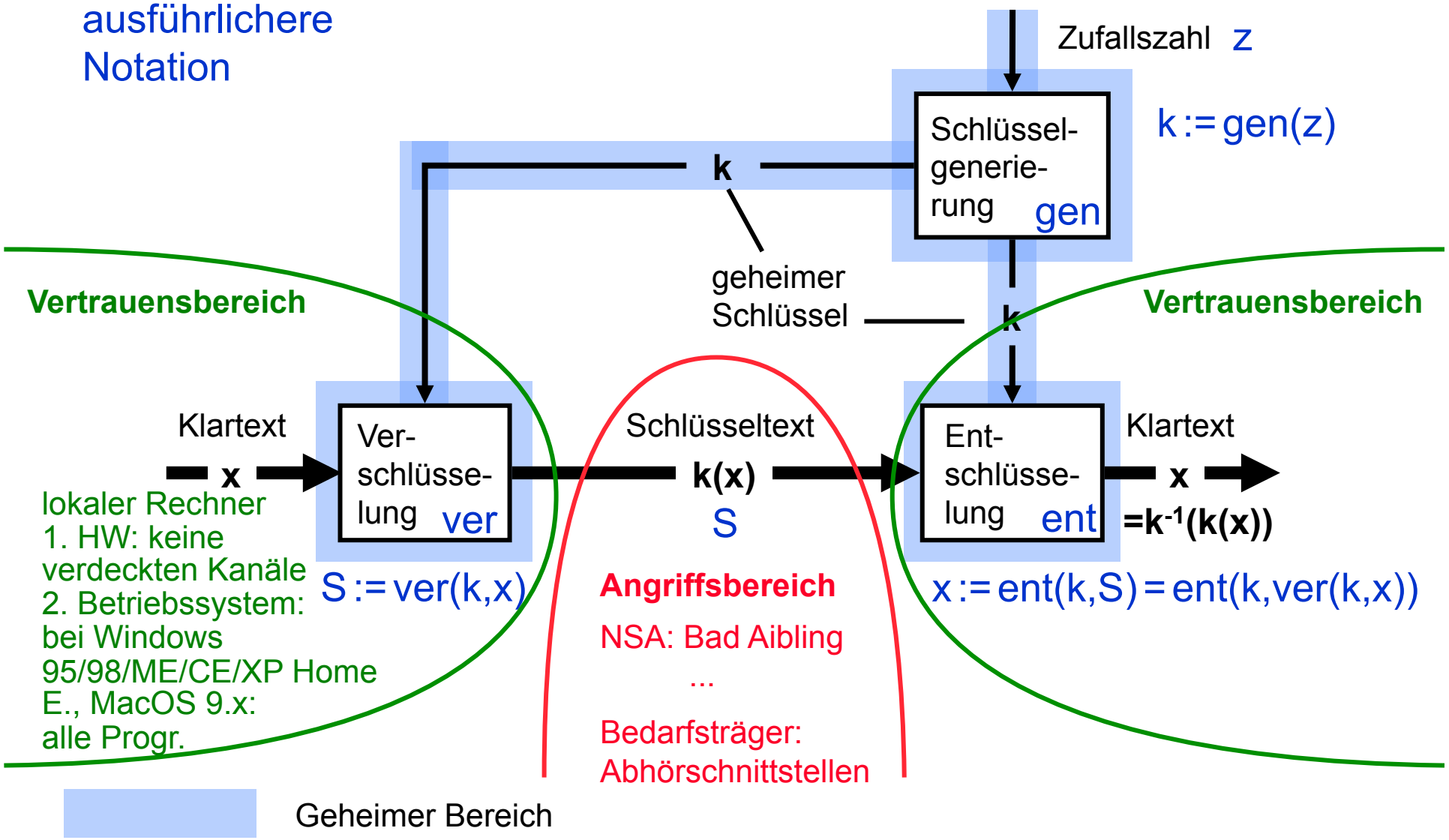
Vertraulichkeit, Konzelation genannt

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

durch Kryptographie unerreichbar:

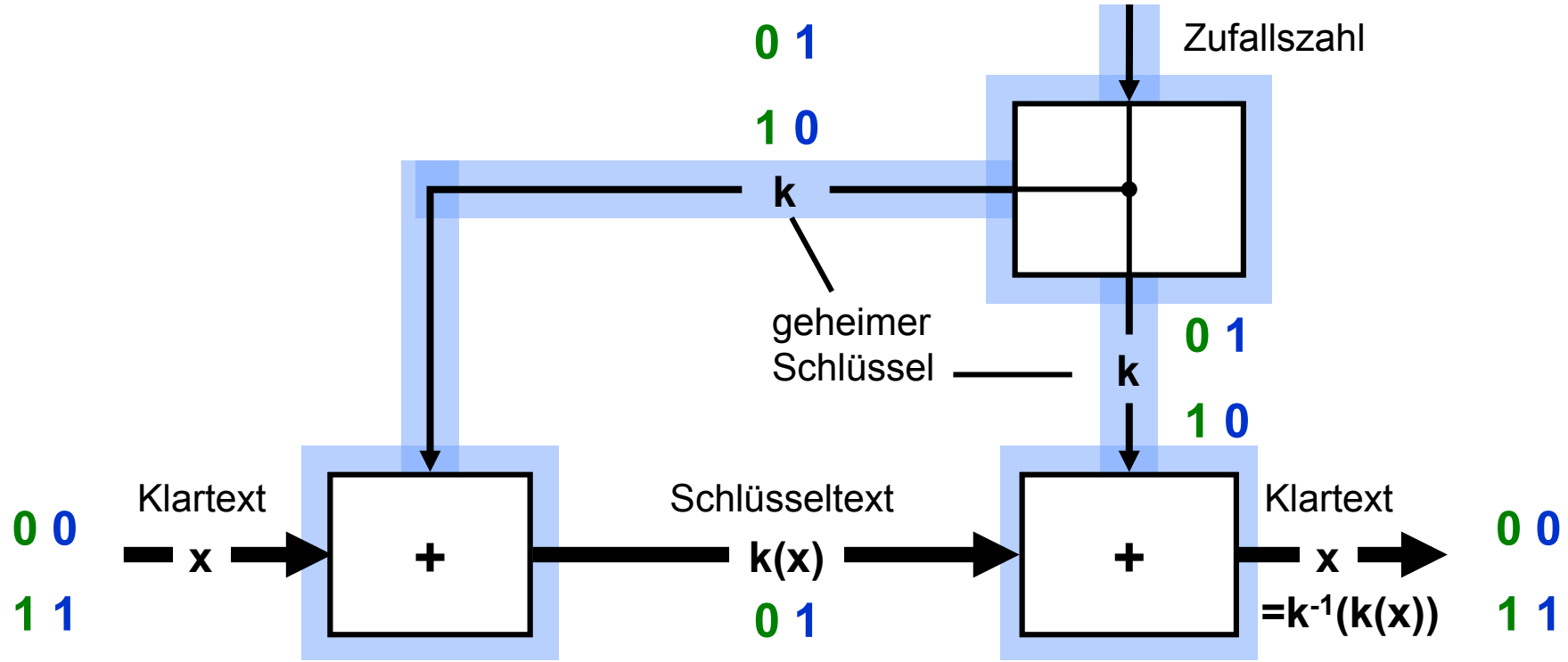
Verfügbarkeit – zumindest nicht gegen starke Angreifer

Symmetrisches Konzelationssystem



Undurchsichtiger Kasten mit Schloß; 2 gleiche Schlüssel

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

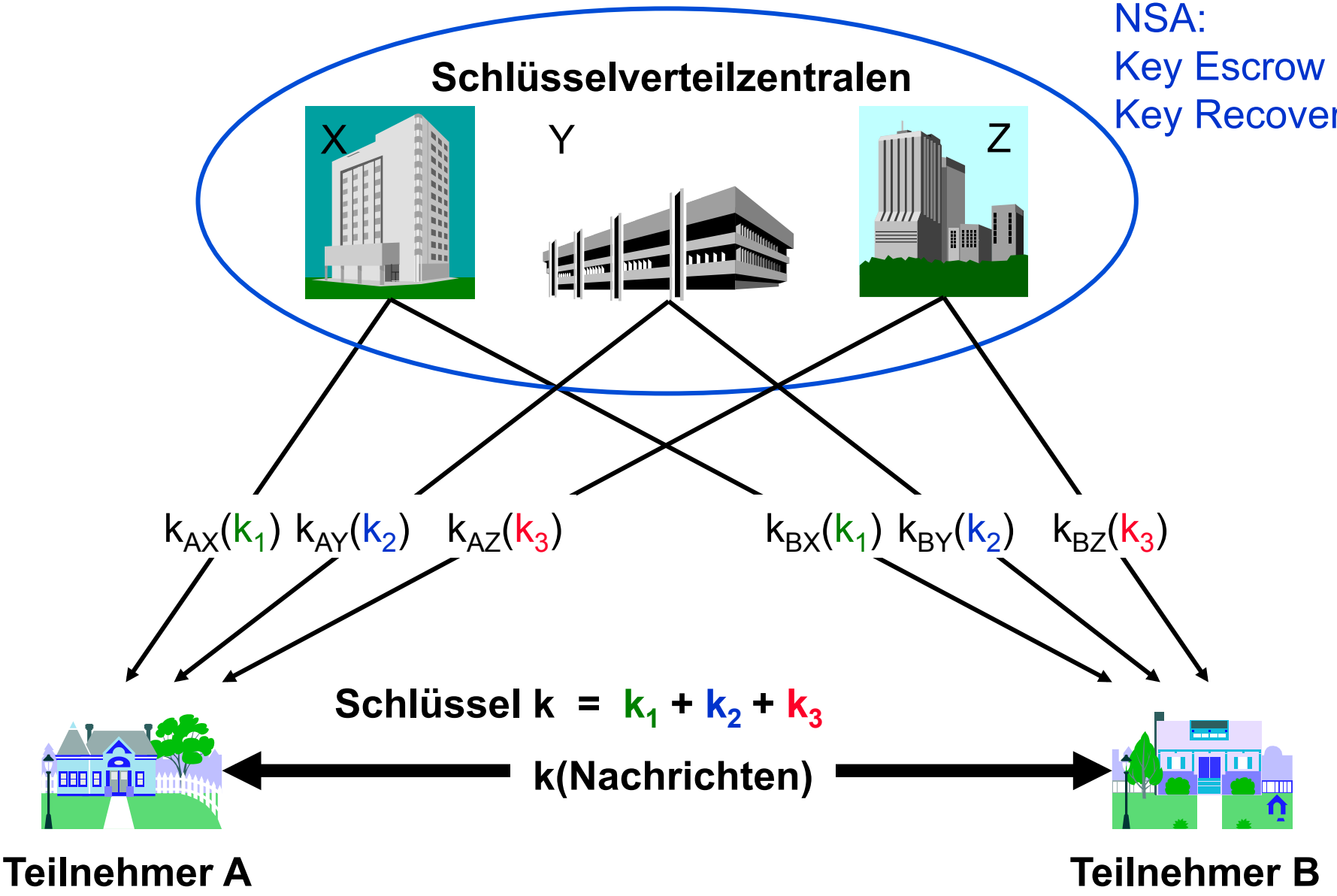


Geheimer Bereich

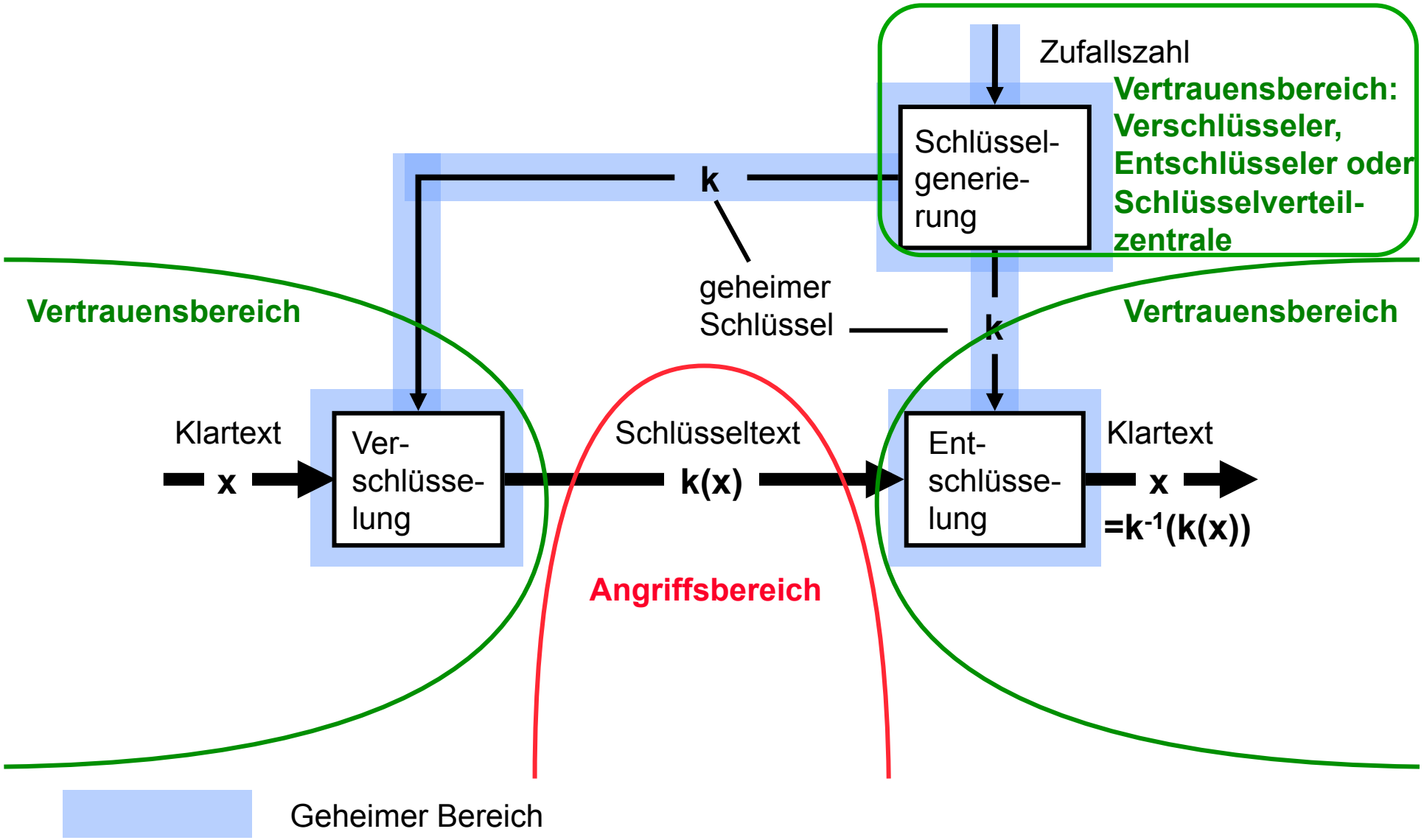
Undurchsichtiger Kasten mit Schloß; 2 gleiche Schlüssel

Schlüsselverteilung bei symmetrischem Kryptosystem

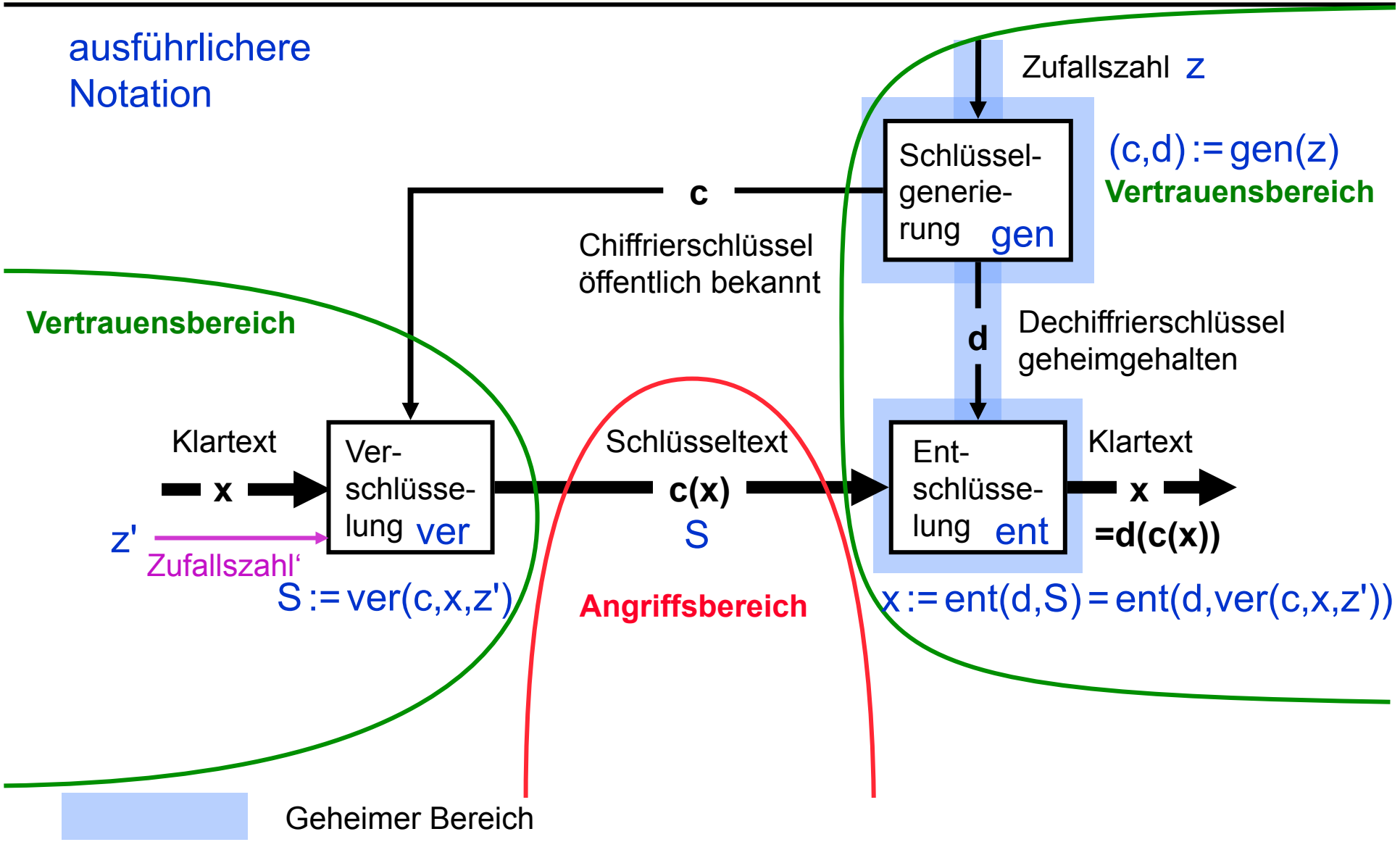
NSA:
Key Escrow
Key Recovery



Sym. Konz.system: Vertrauensbereich Schlüsselgenerierung

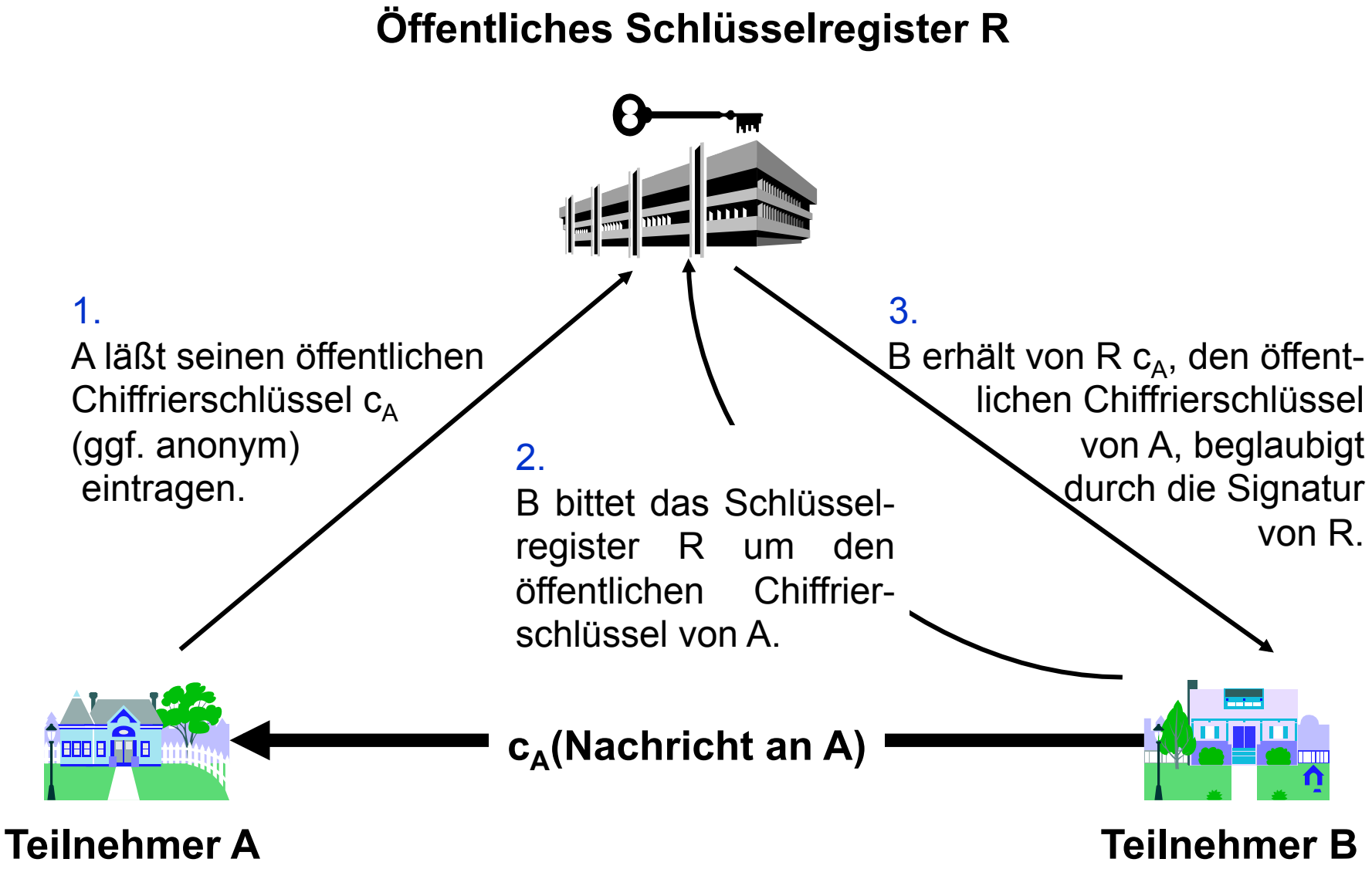


Asymmetrisches Konzelationssystem

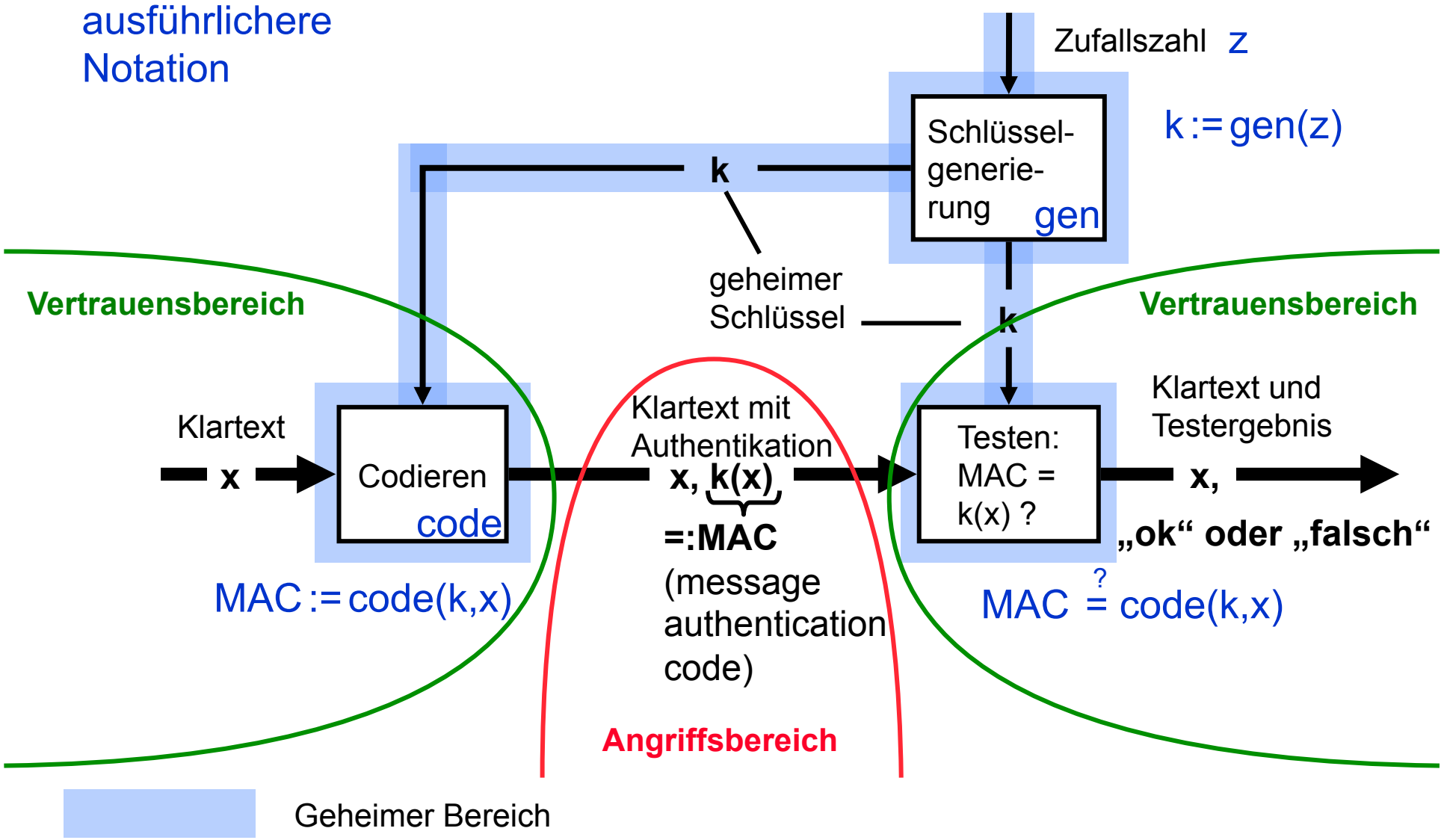


Undurchsichtiger Kasten mit Schnappschloß; 1 Schlüssel

Schlüsselverteilung bei asymmetrischem Konzelationssystem

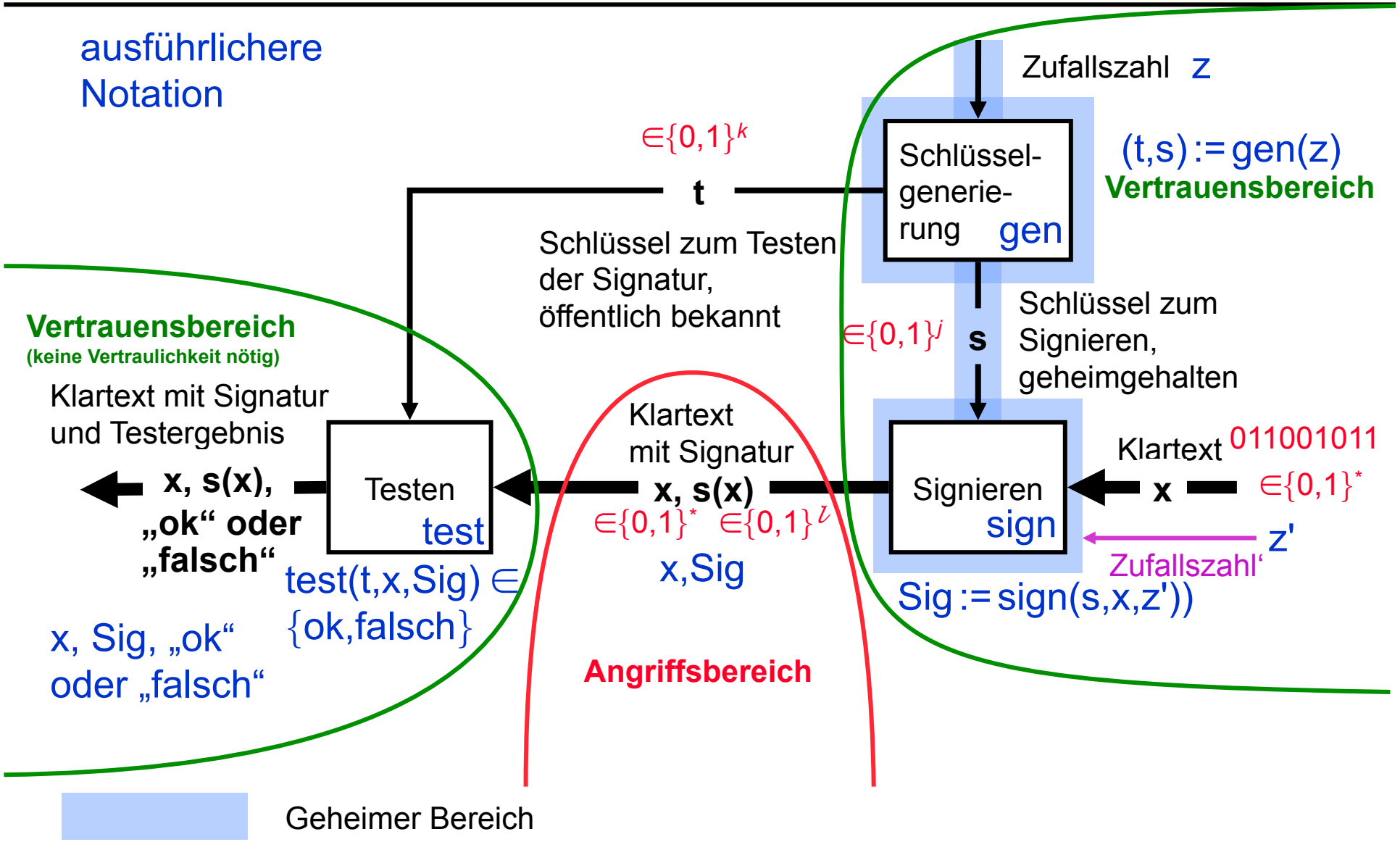


Symmetrisches Authentikationssystem



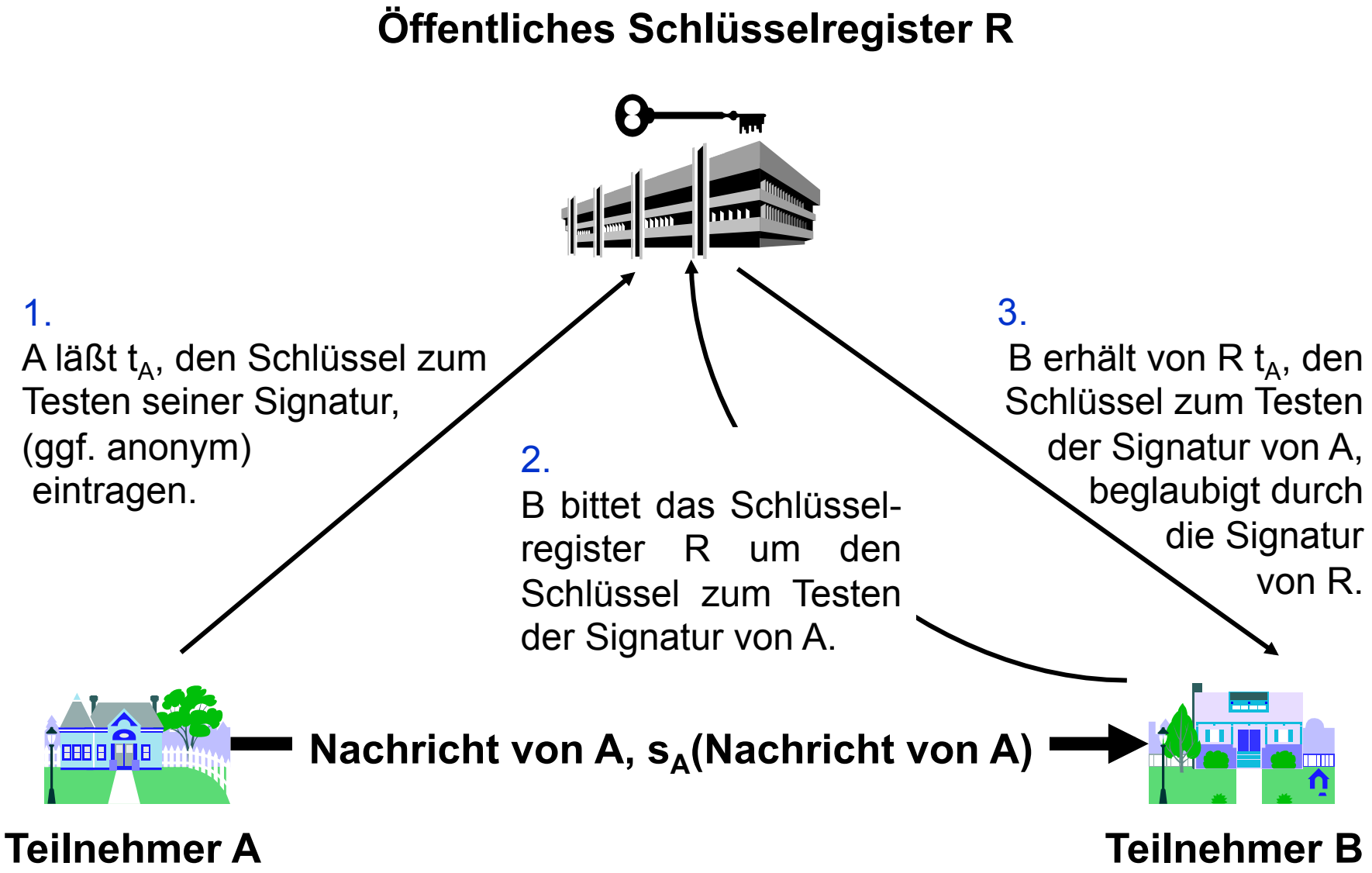
Glasvitrine mit Schloß; 2 gleiche Schlüssel

Digitales Signatursystem

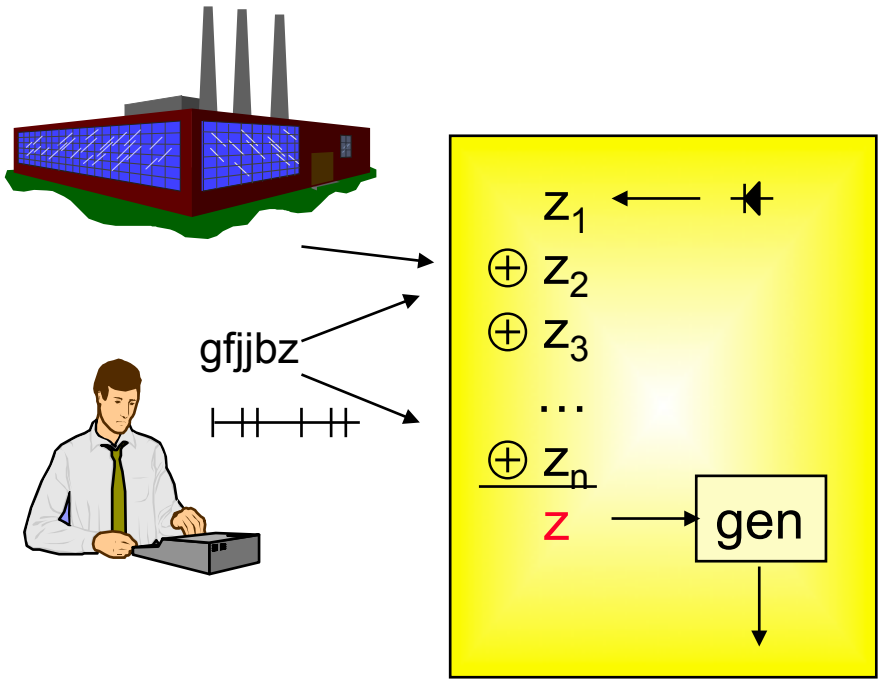


Glasvitrine mit Schloß; 1 Schlüssel

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

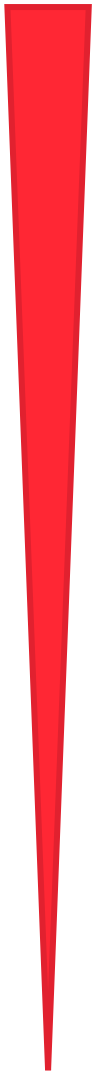
asymmetrische Systeme	je System *n*
symmetrische Systeme	$n \bullet (n\text{-}1)$

Wann Schlüssel generieren und austauschen?

Sicherheit des Schlüsselaustauschs begrenzt kryptographisch erreichbare Sicherheit:

Mehrere Ur-Schlüsselaustausche durchführen

Angriffsziel/ -erfolg



a) Schlüssel (total break)

b) zum Schlüssel äquivalentes Verfahren (universal break)

c) einzelne Nachrichten,

z.B. speziell für Authentikationssysteme

c1) eine gewählte Nachricht (selective break)

c2) irgendeine Nachricht (existential break)

Angriffstypen

Schwere

a) passiv

a1) reiner Schlüsseltext-Angriff (ciphertext-only attack)

a2) Klartext-Schlüsseltext-Angriff (known-plaintext attack)

b) aktiv

(je nach Kryptosystem; asym.: eins von beiden: b1 oder b2;
sym.: ggf. beides: auch b1 und b2)

b1) Signatursystem: Klartext → Schlüsseltext (Signatur)
(chosen-plaintext attack)

b2) Konzelationss.: Schlüsseltext → Klartext
(chosen-ciphertext attack)

Adaptivität

nicht adaptiv

adaptiv

Kriterium: Handlung		**Erlaubnis**
passiver Angreifer	≠	beobachtender Angreifer
aktiver Angreifer	≠	verändernder Angreifer

Grundsätzliches über „kryptographisch stark“

Falls keine informationstheoretische Sicherheit:

1) Verwendung von Schlüssel der festen Länge l:
   - Angreiferalgorithmus kann immer alle 2^l Schlüssel durchprobieren (bricht asym. Kryptosysteme und sym. bei Klartext-Schlüsseltext-Angriff).
   - erfordert exponentiell viele Operationen (ist also für $l > 100$ zu aufwendig).

   → das Beste, was der Kryptosystementwerfer erhoffen kann.

2) Komplexitätstheorie:
   - liefert hauptsächlich asymptotische Resultate
   - behandelt hauptsächlich „worst-case“-Komplexität

   → für Sicherheit unbrauchbar, ebenso „average-case“-Komplexität.

   Wunsch: Problem soll fast überall, d.h. bis auf einen verschwindenden Bruchteil der Fälle, schwer sein.
   - Sicherheitsparameter l (allgemeiner als Schlüssellänge; praktisch nützlich)
   - Wenn $\underbrace{l \to \infty}_{\text{langsam}}$, dann $\underbrace{\text{Brechwahrscheinlichkeit} \to 0}_{\text{schnell}}$.
   - Hoffnung:

Grundsätzliches über „kryptographisch stark“ (Forts.)

3) 2 Komplexitätsklassen:

   Ver-/Entschlüsseln: leicht = polynomiell in l
   Brechen: schwer = nicht polynomiell in l $\approx$ exponentiell in l
   Warum?

   a) Schwerer als exponentiell geht nicht, siehe 1).

   b) Abgeschlossen: Einsetzen von Polynomen in Polynome ergibt Polynome.

   c) Vernünftige Berechnungsmodelle (Turing-, RAM-Maschine) sind polynomiell äquivalent.

   Für die Praxis würde Polynom von hohem Grad für Laufzeit des Angreiferalgorithmus auf RAM-Maschine reichen.

4) Warum komplexitätstheoretische Annahmen ? z.B. Faktorisierung schwer

   Komplexitätstheorie kann bisher keine brauchbaren unteren Schranken beweisen. Kompakte, lang untersuchte Annahmen!

5) Was, wenn sich Annahme als falsch herausstellt?

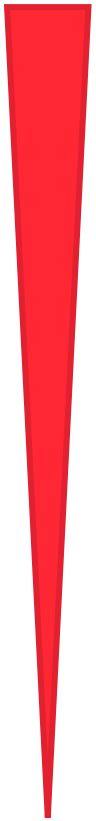
   a) Andere Annahmen treffen.

   b) Genauere Analyse, z.B. Berechnungsmodell genau fixieren und dann untersuchen, ob Polynom von genügend hohem Grad.

6) Beweisziel: Wenn der Angreiferalgorithmus das Kryptosystem brechen kann, dann kann er auch das als schwer angenommene Problem lösen.

Sicherheitsklassen kryptographischer Systeme

Sicherheit



1. informationstheoretisch sicher
2. kryptographisch stark
3. wohluntersucht
4. wenig untersucht
5. geheim gehalten

Überblick über kryptographische Systeme

Sicherheit \ Systemtyp		**Konzelation** sym. sym. Konzelationssystem	**Konzelation** asym. asym. Konzelationssystem	**Authentikation** sym. sym. Authentikationssystem	**Authentikation** asym. digitales Signatursystem
informationstheoretisch		Vernam-Chiffre (one-time pad)	1	Authentikationscodes	2
kryptographisch stark gegen...	**aktiver Angriff**	Pseudo-one-time-pad mit s^2-mod-n-Generator	3 CS **?**	4	GMR
	passiver Angriff	5	System mit mit s^2-mod-n-Generator	6	7
wohluntersucht	**Mathematik**	8	RSA	9	RSA
	Chaos	DES	10	DES	11

Hybride Kryptosysteme (1)

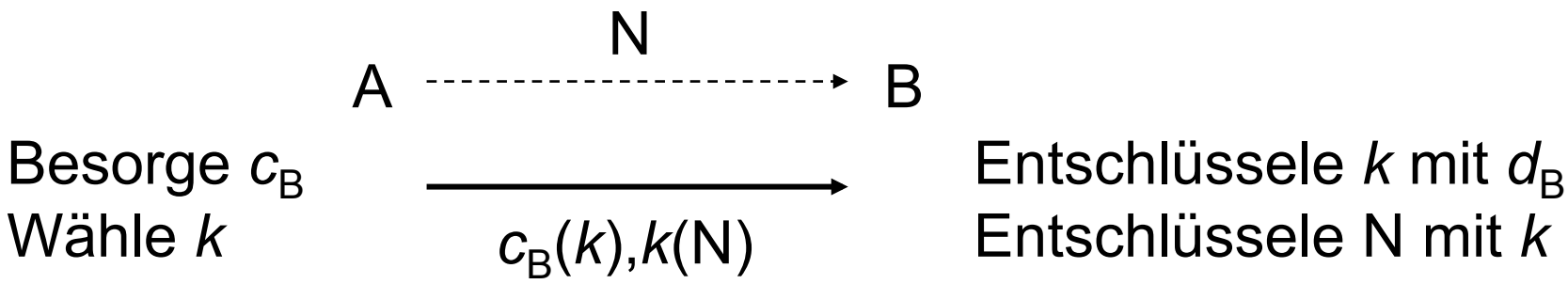
Kombiniere:

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

Wie?

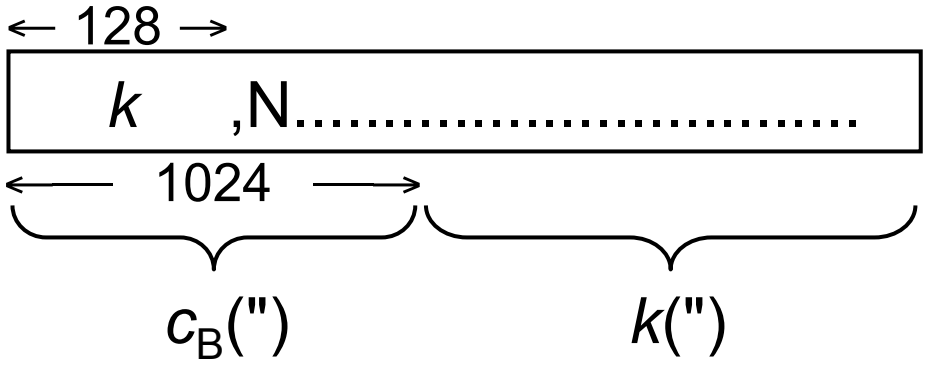
Asymmetrisches System nur, um Schlüssel für symmetrisches auszutauschen

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
Wähle *k*

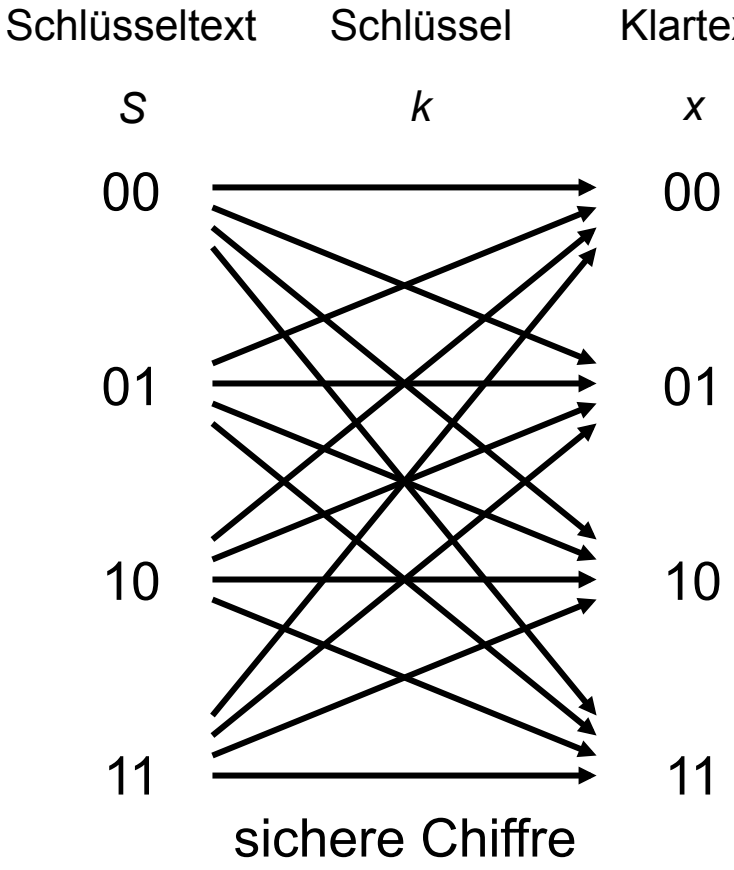
$N,k(N),c_B(B,k,s_A(B,k))$

MAC

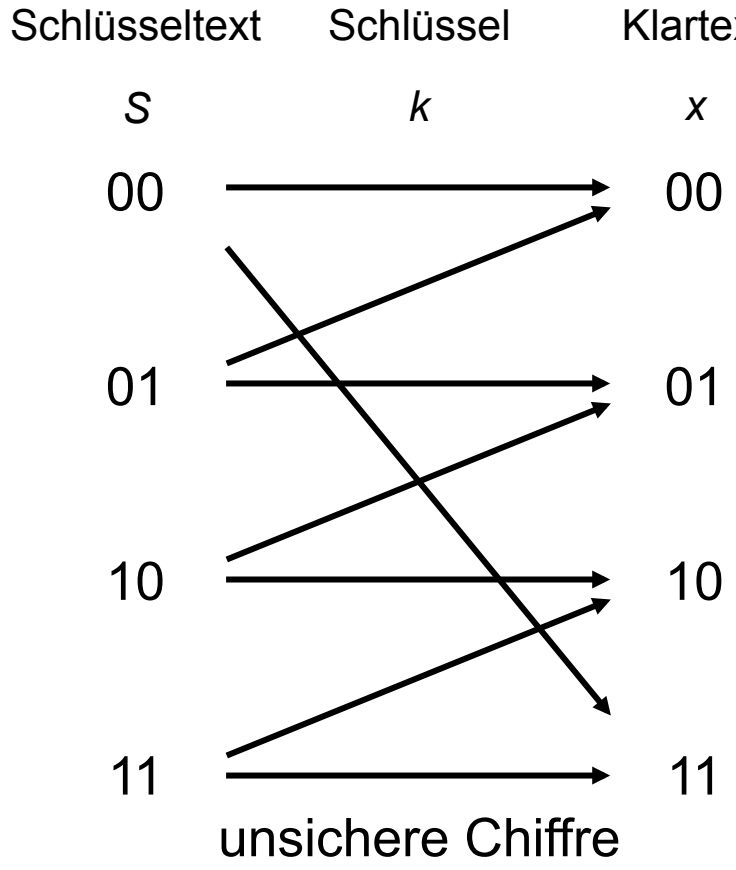
Besorge t_A
Entschlüssele $c_B(B,k,s_A(B,k))$
Teste B,*k* mit t_A
Teste N mit *k*

Informationstheoretisch sichere Konzelation (1)

„Hinter jedem Schlüsseltext *S* kann sich jeder Klartext *x* gleich gut verbergen“



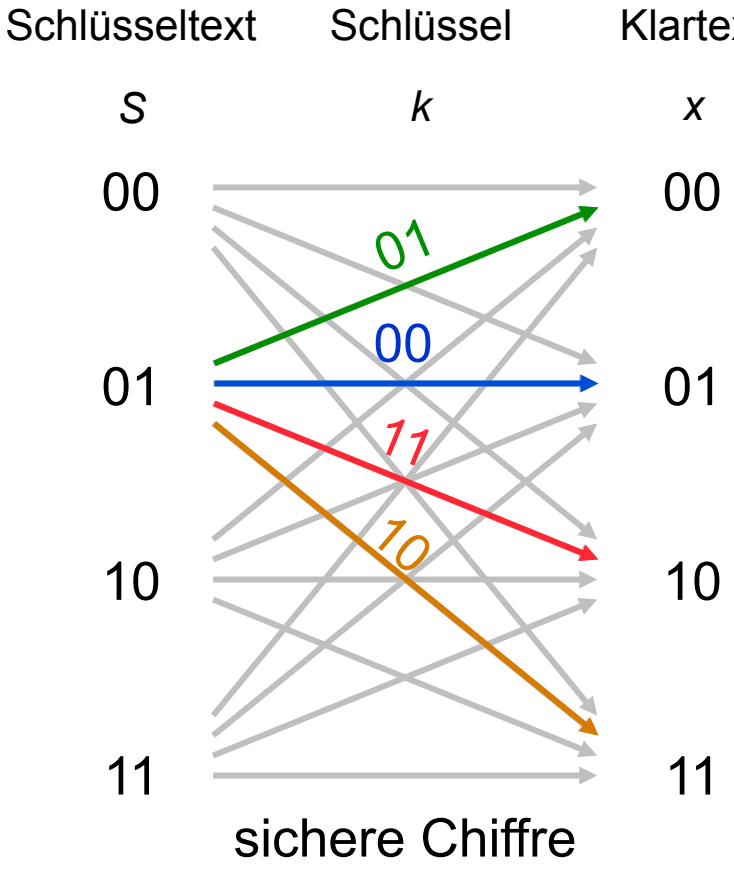
sichere Chiffre



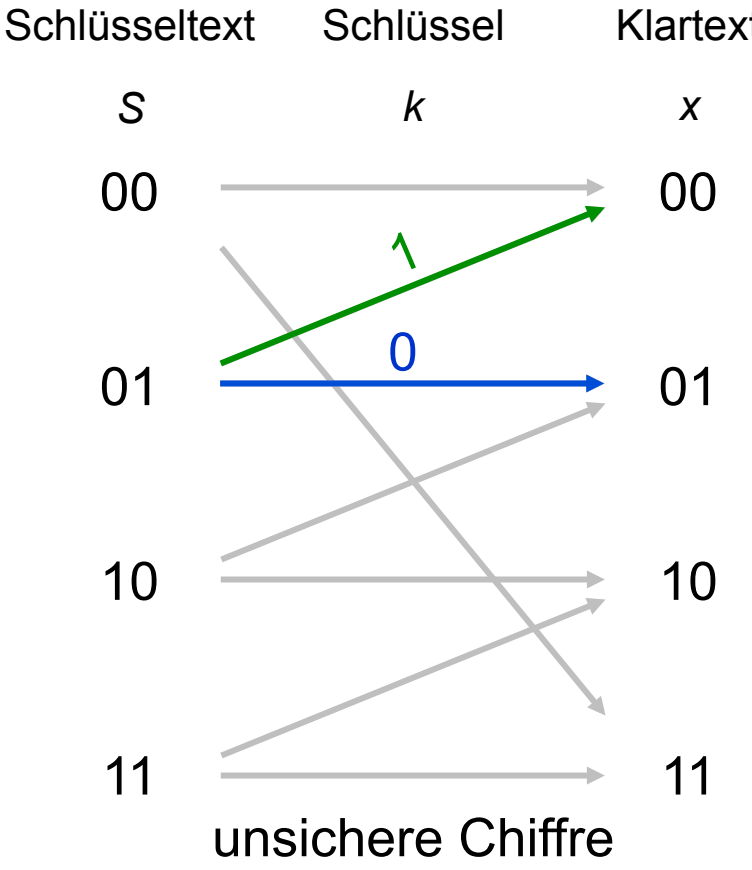
unsichere Chiffre

Informationstheoretisch sichere Konzelation (2)

„Hinter jedem Schlüsseltext *S* kann sich jeder Klartext *x* gleich gut verbergen“



sichere Chiffre



unsichere Chiffre

Bsp.: Vernam-Chiffre mod 2

x = 00 01 00 10

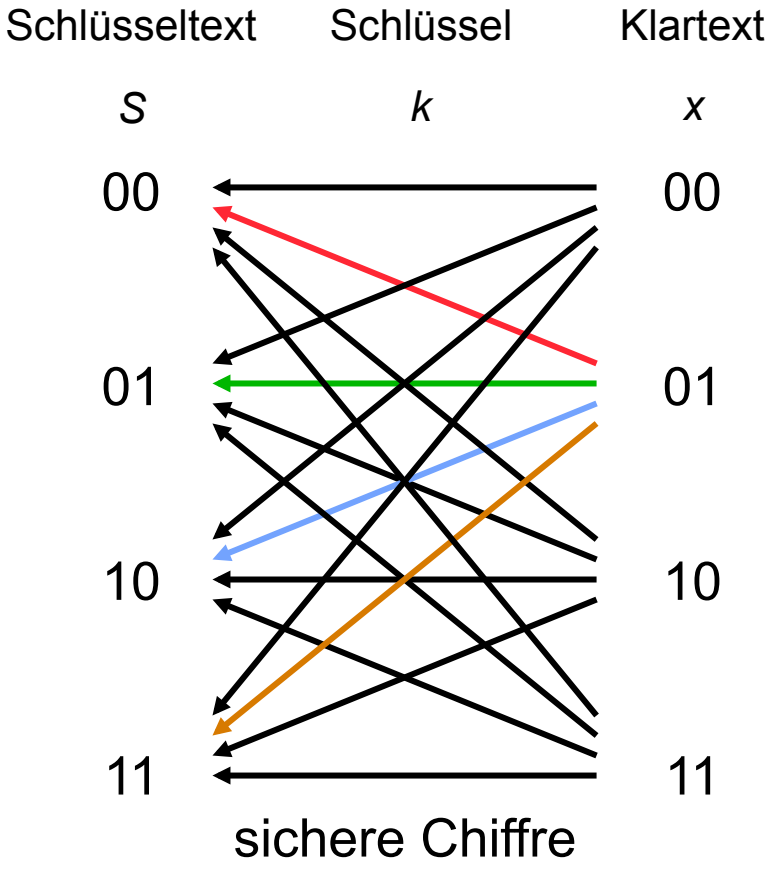
$\oplus$ k = 10 11 01 00

S = 10 10 01 10

Subtraktion von einem Schlüsselbit mod 4 von zwei Klartextbits

Informationstheoretisch sichere Konzelation (3)

Wie passen die verschiedenen Verteilungen zusammen?



Ungleich verteilte Klartexte

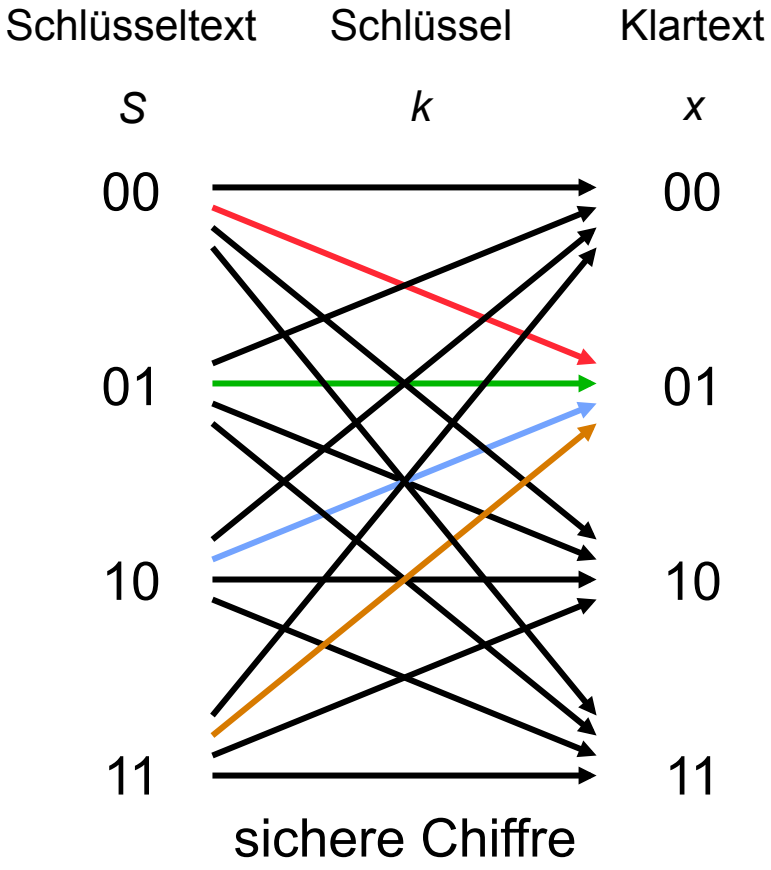
verschlüsselt mit

gleichverteilten Schlüsseln

ergibt gleichverteilte Schlüsseltexte.

Informationstheoretisch sichere Konzelation (4)

Wie passen die verschiedenen Verteilungen zusammen?



gleich-verteilt

gleichverteilt, aber *nicht* unabhängig von den Schlüsseltexten

ungleich verteilt

Gleichverteilte Schlüsseltexte entschlüsselt mit gleichverteilten Schlüsseln kann ungleich verteilte Klartexte dann und nur dann ergeben, wenn die Gleichverteilungen nicht unabhängig voneinander sind, d.h. die Schlüsseltexte wurden aus Klartexten und Schlüsseln berechnet.

Vernam-Chiffre (one-time pad)

Alle Zeichen sind Elemente einer Gruppe G.

Klartext, Schlüssel und Schlüsseltext sind Zeichenketten.

Zur Verschlüsselung einer Zeichenkette x der Länge n wird ein zufällig gewählter und vertraulich auszutauschender Schlüssel $k = (k_1, \ldots, k_n)$ verwendet.

Das i-te Klartextzeichen x_i wird verschlüsselt als

$$S_i := x_i + k_i$$

Entschlüsselt werden kann es durch

$$x_i := S_i - k_i.$$

Bewertung:
1. gegen adaptive Angriffe sicher;
2. einfach zu berechnen;
3. Schlüssel aber sehr lang

Für informationsth. Sicherheit *müssen* Schlüssel so lang sein

Sei $\mathcal{K}$ Schlüsselmenge, $\mathcal{X}$ Klartextmenge und $\mathcal{S}$ Menge der mindestens einmal auftretenden Schlüsseltexte.

$|\mathcal{S}| \geq |\mathcal{X}|$ damit eindeutig entschlüsselbar (k fest)

$|\mathcal{K}| \geq |\mathcal{S}|$ damit hinter jedem Schlüsseltext jeder Klartext stecken kann (x fest)

also $|\mathcal{K}| \geq |\mathcal{X}|$.

Falls Klartext geschickt codiert, folgt:

Schlüssel mindestens so lang wie Klartext.

Vorbereitung: Defs für informationstheoretische Sicherheit

Wie würden Sie

informationstheoretische Sicherheit

von Verschlüsselung definieren?

Schreiben Sie bitte mindestens

2 Definitionen

auf und argumentieren Sie für Ihre Definitionen!

Definitionen für informationstheoretische Sicherheit

1. Definition für informationstheoretische Sicherheit

(alle Schlüssel mit gleicher Wahrscheinlichkeit gewählt)

$$\forall S \in \mathcal{S}\ \exists\ const \in \mathbb{N}\ \forall x \in \mathcal{X}\colon\ |\{k \in \mathcal{K} \mid k(x) = S\}| = const. \qquad (1)$$

Die a-posteriori-Wahrscheinlichkeit eines Klartextes x, wenn der Angreifer den Schlüsseltext S gesehen hat, ist $W(x|S)$.

2. Definition

$$\forall S \in \mathcal{S}\ \forall x \in \mathcal{X}\colon\ W(x|S) = W(x). \qquad (2)$$

Beide Definitionen sind äquivalent:

Nach Bayes gilt:

$$W(x \mid S) = \frac{W(x) \bullet W(S \mid x)}{W(S)}$$

(2) ist also äquivalent zu

$$\forall S \in \mathcal{S}\ \forall x \in \mathcal{X}\colon\ W(S|x) = W(S). \qquad (3)$$

Wir zeigen, dass dies äquivalent ist zu

$$\forall S \in \mathcal{S}\ \exists\ const' \in \mathbb{R}\ \forall x \in \mathcal{X}\colon\ W(S|x) = const'. \qquad (4)$$

Beweis

(3)⇒(4) ist klar mit $const' := W(S)$.

Umgekehrt zeigen wir $const' = W(S)$:

$$W(S) = \sum_x W(x) \bullet \; W(S|x)$$

$$= \sum_x W(x) \bullet \; const'$$

$$= const' \bullet \sum_x W(x)$$

$$= const'.$$

(4) sieht (1) schon sehr ähnlich: Allgemein ist

$$\mathrm{W}(S|x) = \mathrm{W}(\{k \mid k(x) = S\}),$$

und wenn alle Schlüssel gleichwahrscheinlich sind,

$$\mathrm{W}(S|x) = |\{k \mid k(x) = S\}| \,/\, |\mathcal{K}|.$$

Dann ist (4) äquivalent (1) mit

$$const = const' \bullet |\mathcal{K}|.$$

Eine weitere Definition für informationstheoret. Sicherheit

Manchmal schlagen StudentInnen die folgende Definition vor:

$$\forall S \in \mathcal{S} \quad \forall x \in \mathcal{X}: \quad W(S) = W(S|x).$$

Dies ist *nicht* äquivalent, aber eine geringfügige Modifikation ist es:

3. Definition

$$\forall S \in \mathcal{S} \quad \forall x \in \mathcal{X} \text{ mit } W(x)>0: \quad W(S) = W(S|x).$$

Definitionen 2. und 3. sind äquivalent:

Zur Erinnerung: Bayes

$$\boxed{W(x \mid S) = \frac{W(x) \bullet W(S \mid x)}{W(S)}}$$

$$W(x|S) = W(x) \qquad \text{<==> (Bayes)}$$

$$\frac{W(x) \bullet W(S \mid x)}{W(S)} = W(x) \qquad \text{<==> (wenn } W(x) \neq 0\text{, durch } W(x) \text{ teilen)}$$

$$W(S|x) = W(S)$$

$W(S|x)$ wie vorgeschlagen unterstellt, dass x gesendet werden kann, d.h. $W(x)>0$.

Symmetrische Authentikationssysteme (1)

Schlüsselverteilung:

Wie symmetrische Konzelationssysteme

Einfaches Beispiel (Angreifersicht)

Authentisiert gesendet werden soll das Ergebnis eines Münzwurfs: Head (H) oder Tail (T)

		x, MAC			
		H,0	H,1	T,0	T,1
k	00	H	-	T	-
	01	H	-	-	T
	10	-	H	T	-
	11	-	H	-	T

Sicherheit: z.B. Angreifer will T senden.

a) blind : Erwischt mit Wahrscheinlichkeit 0,5

b) sehend : z.B. H,0 abgefangen $\Rightarrow$ $k \in \{00, 01\}$

Immer noch T,0 und T,1 mit Wahrscheinlichkeit 0,5

Symmetrische Authentikationssysteme (2)

Definition „Informationstheoretische Sicherheit“ mit Fehlerwahrscheinlichkeit ε:

$\forall x$, MAC (die Angreifer sieht)

$\forall y \neq x$ (das Angreifer statt x sendet)

$\forall$ MAC' (von denen Angreifer den besten für y aussucht)

$W(k(y) = \text{MAC}' \mid k(x) = \text{MAC}) \leq \varepsilon$

(Wahrscheinlichkeit, dass MAC' stimmt, wenn man nur die Schlüssel k betrachtet, die wegen (x,MAC) noch möglich sind.)

Verbesserung des Beispiels:

a) 2σ Schlüsselbits statt 2: $k = k_1 k_1^* \ldots k_\sigma k_\sigma^*$
MAC = $\text{MAC}_1,\ldots,\text{MAC}_\sigma$; MAC_i aus $k_i k_i^*$
$\Rightarrow$ Fehlerwahrscheinlichkeit $2^{-\sigma}$

b) l Nachrichtenbits:

$$x^{(1)}, \text{MAC}^{(1)} = \text{MAC}_1^{(1)}, \ldots, \text{MAC}_\sigma^{(1)}$$
$$\vdots \qquad \vdots \qquad\qquad\qquad \vdots$$
$$x^{(l)}, \text{MAC}^{(l)} = \text{MAC}_1^{(l)}, \ldots, \text{MAC}_\sigma^{(l)}$$

Symmetrische Authentikationssysteme (3)

Grenzen:

σ-bit-MAC $\Rightarrow$ Fehlerwahrscheinlichkeit $\geq 2^{-\sigma}$
(MAC raten)

σ-bit-Schlüssel $\Rightarrow$ Fehlerwahrscheinlichkeit $\geq 2^{-\sigma}$
(Schlüssel raten, MAC ausrechnen)

Noch klar: Für Fehlerwahrscheinlichkeit $2^{-\sigma}$ reichen σ-bit-Schlüssel nicht, denn $k(x)$ = MAC schließt viele k's aus.

Satz: Man braucht 2σ-bit-Schlüssel
(Für weitere Nachrichten reichen σ, wenn Empfänger auf Authentikations„fehler“ geeignet reagiert.)

Möglich zur Zeit: $\approx 4\sigma \cdot \log_2(\text{Länge}(x))$

(Wegman, Carter)

Viel kürzer als one-time pad.

Zu kryptographisch starken Systemen (1)

Mathematische Geheimnisse:

(zum Entschlüsseln, Signieren ...)

p, q, prim.

Öffentlicher Teil:

(zum Verschlüsseln, Testen, ...)

$n = p \cdot q$

p, q groß, z.Zt. $\approx \ell$ = 500 bis 2000 Bit
(Theorie : $\ell \to \infty$)

Oft noch Spezialeigenschaft

$p \equiv q \equiv 3 \bmod 4$

(die Bedeutung von „≡ ... mod“ ist:
$a \equiv b \bmod c$ gdw. c teilt a-b,
anders ausgedrückt: a und b lassen
bei Division durch c denselben Rest)

Zu kryptographisch starken Systemen (2)

Verwendung : s^2-mod-n-Generator,
GMR und viele andere,
z.B. nur wohluntersuchte Systeme wie RSA

(wichtige Alternative nur „diskreter Logarithmus“,
auch Zahlentheorie, ähnlich gut)

Nötig: 1. Faktorisieren schwer
2. p,q erzeugen leicht
3. Nachrichtenabhängige Dinge mit p, q
mit n allein geht nur die Umkehrung

Faktorisieren ? (1)

Klar : In NP ⇒ Schwierigkeiten z.Zt. nicht beweisbar

Komplexität z.Zt.

$$L(n) = e^{c \cdot \sqrt[3]{\ln(n) \cdot (\ln\ln(n))^2}} \approx e^{\sqrt[3]{l}}$$

, $c \approx 1{,}9$

„subexponentiell“

Praktisch bis 155 Dezimalstellen im Jahr 1999
174 Dezimalstellen im Jahr 2003
200 Dezimalstellen im Jahr 2005
232 Dezimalstellen im Jahr 2010 (www.crypto-world.com/FactorRecords.html)

(Merke : ∃ schnellere Algorithmen z.B. für $2^r \pm 1$, so was stört nicht.)

Annahme: Faktorisieren ist schwer

(Beachte : Wenn Angreifer z.B. jedes 1000-te *n* faktorisieren könnte, wäre das unakzeptabel.)

Faktorisieren ? (2)

∀ PPA $\mathcal{F}$ (probabilistischer polynomialer Algorithmus, der zu faktorisieren versucht)

∀ Polynome Q

∃ L ∀ $l \geq L$: (asymptotisch gilt:)

Wenn p, q zufällige Primzahlen der Länge l und $n = p \bullet q$:

$$W(\mathcal{F}(n) = (p, q)) \leq \frac{1}{Q(l)}$$

(Wahrscheinlichkeit, dass $\mathcal{F}$ wirklich faktorisiert, sinkt schneller als $\frac{1}{jedes\ Polynom}$.)

Vertrauenswürdig ??

Von allen am gründlichsten untersucht.

Primzahlensuche (1)

1. Gibt es genug ? (Auch für Faktorisierungsannahme wichtig)

$$\frac{\pi(x)}{x} \approx \frac{1}{\ln(x)}$$

$\pi(x)$ Anzahl der Primzahlen $\leq x$

„Primzahlsatz“

$\Rightarrow$ bis Länge ℓ mehr als jede ℓ-te.

Und $\approx$ jede 2. $\equiv 3 \bmod 4$ „Dirichletscher Primzahlsatz“

2. Suchprinzip :

repeat

Wähle Zufallszahl p ($\equiv 3 \bmod 4$)

teste ob p prim

until p prim

Primzahlensuche (2)

3. Primzahltests :

(Anmerkung: Faktorisierungsversuch zu langsam)

Probabilistisch; „Rabin-Miller“

Spezialfall $p \equiv 3 \bmod 4$:

p prim $\Rightarrow \forall\, a \not\equiv 0 \bmod p : \; a^{\frac{p-1}{2}} \equiv \pm 1 \pmod p$

p nicht prim $\Rightarrow$ für $\leq \frac{1}{4}$ der a´s : $a^{\frac{p-1}{2}} \equiv \pm 1 \pmod p$

$\Rightarrow$ Teste das für m verschiedene, unabhängig gewählte a´s,

Fehlerwahrscheinlichkeit $\leq \frac{1}{4^m}$

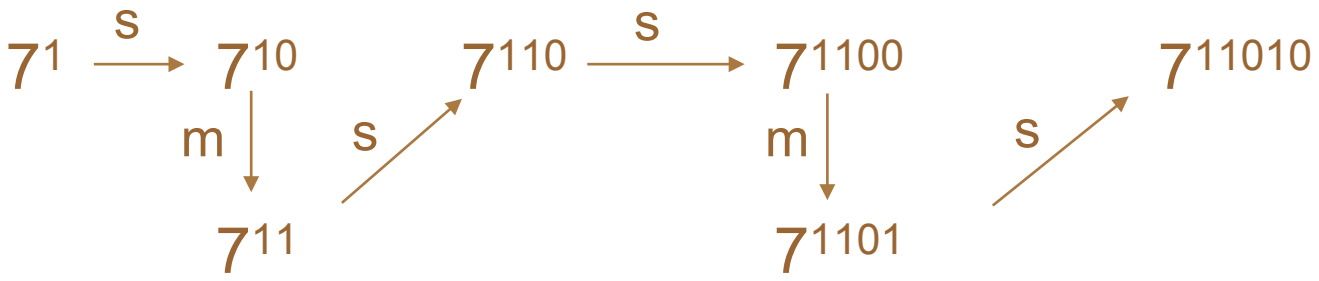
(stört im allgemeinen nicht)

Rechnen mit und ohne *p*,*q* (1)

Z_n : Restklassenring mod n $\hat{=}$ {0, ... , n-1}

- +, -, • schnell
- Exponentiation „schnell“ (square & multiply)

  Bsp: $7^{26} = 7^{(11010)_2}$; von links



- ggT schnell in Z (Euklidscher Algorithmus)

Rechnen mit und ohne *p*,*q* (2)

Z_n^* : Multiplikative Gruppe

$a \in Z_n^* \Leftrightarrow \text{ggT}(a,n) = 1$

- Invertierung schnell (erweiterter Euklidscher Algorithmus)
  Bestimmt zu a,n die Werte u,v mit

  $$a \cdot u + n \cdot v = 1$$

  Dann gilt: $u \equiv a^{-1} \bmod n$

  Bsp: $3^{-1} \bmod 11$?

$$= -11 + 4 \cdot 3$$

$$11 = 3 \cdot 3 + 2 \qquad = 1 \cdot 3 - 1 \cdot (11 - 3 \cdot 3)$$

$$3 = 1 \cdot 2 + 1 \longrightarrow 1 = 1 \cdot 3 - 1 \cdot 2$$

$$\Rightarrow 3^{-1} \equiv 4 \bmod 11$$

Rechnen mit und ohne *p*,*q* (3)

Elementzahl von Z_n^*

Die Eulersche Φ-Funktion ist definiert als

$$\Phi(n) := \left|\{a \in \{0,...,n\text{-}1\} \mid \text{ggT}(a,n)=1\}\right|,$$

wobei für beliebige ganze Zahlen $n \neq 0$ gilt: $\text{ggT}(0,n)=|n|$.

Es folgt sofort aus den beiden Definitionen, dass

$$|Z_n^*| = \Phi(n).$$

Speziell für $n = p \bullet q$, p,q prim und $p \neq q$ kann man $\Phi(n)$ leicht ausrechnen:

$$\Phi(n) = (p\text{-}1) \bullet (q\text{-}1)$$

ggT ≠ 1 haben nämlich 0, dann p, $2p$, …, $(q\text{-}1)p$ und q, $2q$, …, $(p\text{-}1)q$, und diese $1+(q\text{-}1)+(p\text{-}1) = p+q\text{-}1$ Zahlen sind für $p \neq q$ alle verschieden.

Rechnen mit und ohne *p*,*q* (4)

Zusammenhang $Z_n \leftrightarrow Z_p, Z_q$:

Chinesischer Restsatz

$$x \equiv y \bmod n \quad \Leftrightarrow \quad x \equiv y \bmod p \quad \wedge \quad x \equiv y \bmod q$$

denn

$$n|(x-y) \quad \Leftrightarrow \quad p|(x-y) \quad \wedge \quad q|(x-y)$$

$n = p \cdot q$, p,q prim, $p \neq q$

$\Rightarrow$ Um f(x) mod n zu berechnen, zunächst mod p, q einzeln berechnen

y_p := f(x) mod p

y_q := f(x) mod q

Rechnen mit und ohne *p*,*q* (5)

Zusammensetzen ?

Erweiterter Euklid : $u \cdot p + v \cdot q = 1$

$$y := (u \cdot p) \cdot y_q + (v \cdot q) \cdot y_p \quad \begin{cases} \equiv y_p \bmod p \\ \equiv y_q \bmod q \end{cases}$$

Denn :

	mod *p*	mod *q*
$u \cdot p$	0	1
$v \cdot q$	1	0
y	$0 \cdot y_q + 1 \cdot y_p$	$1 \cdot y_q + 0 \cdot y_p$
	$\equiv y_p$	$\equiv y_q$

CRA

Rechnen mit und ohne *p*,*q* (6)

Quadrate und Wurzeln

$QR_n := \{ x \in Z_n^* \mid \exists\, y \in Z_n^* : y^2 \equiv x \bmod n \}$

x : „quadratischer Rest“

y : „Wurzeln aus x“

$-y$ ist auch Wurzel

$(-1)^2 = 1$

Aber Vorsicht: z.B. mod 8

$1^2 \equiv 1 \quad 3^2 \equiv 1$

$7^2 \equiv 1 \quad 5^2 \equiv 1$

} 4 Wurzeln

QR_n multiplikative Gruppe:

$x_1, x_2 \in QR_n \Rightarrow x_1 \bullet x_2 \in QR_n : (y_1 y_2)^2 = y_1^2 y_2^2 = x_1 x_2$

$x_1^{-1} \in QR_n : (y_1^{-1})^2 = (y_1^2)^{-1} = x_1^{-1}$

Rechnen mit und ohne *p*,*q* (7)

Quadrate und Wurzeln mod *p*, prim:

Z_p Körper

$\Rightarrow$ Wie gewohnt $\leq$ 2 Wurzeln

$x \not\equiv 0$, $p \neq 2$: 0 oder 2 Wurzeln

$$\Rightarrow |QR_p| = \frac{p-1}{2}$$ (Quadrierfunktion 2 $\rightarrow$ 1)

x	0	1	2	...	$\frac{p-1}{2}$	$-\frac{p-1}{2}$	...	–2	–1	= p - 1
x^2	0	1	4	...			...	4	1	

Jacobi – Symbol $\left(\frac{x}{p}\right) := \begin{cases} 1 \text{ falls } x \in QR_p \\ -1 \text{ sonst} \end{cases}$ (für $x \in Z_p^*$)

Rechnen mit und ohne *p*,*q* (8)

Fortsetzung Quadrate und Wurzeln mod *p*, prim:

Euler Kriterium : $\left(\frac{x}{p}\right) \equiv x^{\frac{p-1}{2}} \bmod p$

(d.h. schneller Algorithmus zur Quadratprüfung)

Beweis mittels kleinem Fermatschen Satz:

$x^{p-1} \equiv 1 \mod p$

Wertebereich ok : $x^{\frac{p-1}{2}} \in \{\pm 1\}$, da $(x^{\frac{p-1}{2}})^2 \equiv 1$

x Quadrat : $\left(\frac{x}{p}\right) = 1 \Rightarrow x^{\frac{p-1}{2}} \equiv (y^2)^{\frac{p-1}{2}} \equiv y^{p-1} \equiv 1$

x kein Quadrat : Die $\frac{p-1}{2}$ Lösungen von $x^{\frac{p-1}{2}} \equiv 1$ sind die Quadrate. Also erfüllt kein Nicht-Quadrat die Gleichung.

Also: $x^{\frac{p-1}{2}} \equiv -1$.

Rechnen mit und ohne *p*,*q* (9)

Quadrate und Wurzeln mod $p \equiv 3 \bmod 4$

- Wurzelziehen leicht : Gegeben $x \in QR_p$

$$w := x^{\frac{p+1}{4}} \qquad \text{ist Wurzel}$$

Beweis : 1. $p \equiv 3 \bmod 4 \Rightarrow \frac{p+1}{4} \in \mathbb{N}$

$$2.\ w^2 = x^{\frac{p+1}{2}} = x^{\frac{p-1}{2}+1} = x^{\frac{p-1}{2}} \bullet x = 1 \bullet x$$

$$\Downarrow$$

Euler, $x \in QR_p$

Und : $w \in QR_p$ (Potenz von $x \in QR_p$) → Mehrfaches Wurzelziehen geht

- $\left(\frac{-1}{p}\right) \equiv (-1)^{\frac{p-1}{2}} = (-1)^{\frac{4r+2}{2}} = (-1)^{2r+1} = -1$

($p = 4r+3$)

$\Rightarrow -1 \notin QR_p$

$\Rightarrow$ Von Wurzeln $\pm w$: $-w \notin QR_p$ (sonst $-1 = (-w) \bullet w^{-1} \in QR_p$)

Rechnen mit und ohne *p*,*q* (10)

Quadrate und Wurzeln mod *n* mit *p*,*q*

(mögliche geheime Operationen)

- Quadrattest ist leicht ($n = p \cdot q$, p,q prim, $p \neq q$)

$$x \in \mathrm{QR}_n \quad \Leftrightarrow \quad x \in \mathrm{QR}_p \ \wedge \ x \in \mathrm{QR}_q$$

Chinesischer Restsatz

Beweis: „$\Rightarrow$“ $x \equiv w^2 \bmod n \quad \Rightarrow \quad x \equiv w^2 \bmod p \ \wedge \ x \equiv w^2 \bmod q$

„$\Leftarrow$“ $x \equiv w_p^2 \bmod p \ \wedge \ x \equiv w_q^2 \bmod q$

$w := \mathrm{CRA}(w_p, w_q)$

dann $w \equiv w_p \bmod p \ \wedge \ w \equiv w_q \bmod q$

mit Chinesischem Restsatz folgt aus:

$w^2 \equiv w_p^2 \equiv x \bmod p \ \wedge \ w^2 \equiv w_q^2 \equiv x \bmod q$

$w^2 \equiv x \bmod n$

Rechnen mit und ohne *p*,*q* (11)

Fortsetzung Quadrate und Wurzeln mod *n* mit *p*,*q*

$x \in QR_n \Rightarrow$ *x* hat genau 4 Wurzeln

(mod *p* und mod *q* : $\pm w_p, \pm w_q$.

Daher die 4 Kombinationen nach Chinesischem Restsatz)

- Wurzelziehen ist leicht ($p, q \equiv 3 \bmod 4$)

  Bestimme Wurzeln w_p, w_q mod *p*, *q*

$$w_p := x^{\frac{p+1}{4}} \qquad w_q := x^{\frac{q+1}{4}}$$

  kombinieren mit CRA

Rechnen mit und ohne *p*,*q* (12)

Fortsetzung Quadrate und Wurzeln mod *n* mit *p*,*q*

Jacobi Symbol $\left(\frac{x}{n}\right) := \left(\frac{x}{p}\right) \bullet \left(\frac{x}{q}\right)$

Also: $\left(\frac{x}{n}\right) = \begin{cases} +1 & \text{wenn } x \in QR_p \wedge x \in QR_q \vee \\ & \quad\; x \notin QR_p \wedge x \notin QR_q \\ -1 & \text{wenn „überkreuz“} \end{cases}$

Also: $x \in QR_n \Rightarrow \left(\frac{x}{n}\right) = 1$

$\not\Leftarrow$ gilt nicht

Rechnen mit und ohne *p*,*q* (13)

Fortsetzung Quadrate und Wurzeln mod *n* mit *p*,*q*

Jacobi – Symbol bestimmen ist leicht

z.B. $p \equiv q \equiv 3 \bmod 4$

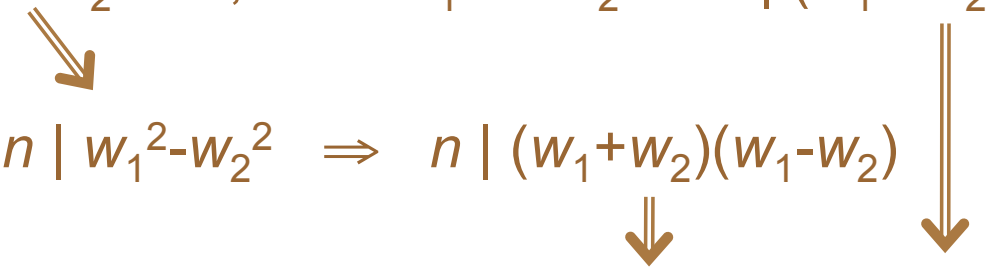
$$\left(\frac{-1}{n}\right) = \left(\frac{-1}{p}\right) \bullet \left(\frac{-1}{q}\right) = (-1) \bullet (-1) = 1$$

aber $-1 \notin QR_n$, da $\notin QR_{p,q}$

Rechnen mit und ohne *p*,*q* (14)

Quadrate und Wurzeln mod *n* ohne *p*,*q*

- Wurzelziehen schwer: beweisbar so schwer wie Faktorisieren
  - a) Wenn jemand 2 wesentlich verschiedene Wurzeln eines *x* mod *n* kennt, kann er definitiv *n* faktorisieren (d.h. $w_1^2 \equiv w_2^2 \equiv x$, aber $w_1 \not\equiv \pm w_2 \Rightarrow n \mid\!\!\not\mid (w_1 \pm w_2)$)



Beweis: $n \mid w_1^2 - w_2^2 \Rightarrow n \mid (w_1+w_2)(w_1-w_2)$

p in einen Faktor, *q* im anderen

$\Rightarrow$ ggT(w_1+w_2, *n*) ist *p* oder *q*

Rechnen mit und ohne *p*,*q* (15)

Fortsetzung Quadrate und Wurzeln mod *n* ohne *p*,*q*

b) Skizze von „Faktorisieren schwer ⇒ Wurzel schwer“

Beweis von „Faktorisieren leicht ⇐ Wurzel leicht“

Also Ann. : $\exists\ \mathcal{W} \in$ PPA: Wurzelziehalgorithmus

z.Z. : $\exists\ \mathcal{F} \in$ PPA: Faktorisierungsalgorithmus

Struktur

program $\mathcal{F}$
  subprogram $\mathcal{W}$
  [black box]
begin
...
call $\mathcal{W}$
...
call $\mathcal{W}$
...
end.

} polynomial oft

Rechnen mit und ohne *p*,*q* (16)

zu b)

$\mathcal{F}$: Eingabe n

repeat forever

wähle $w \in Z_n^*$ zufällig, setze $x := w^2$

$w' := \mathcal{W}(n,x)$

teste ob $w' \not\equiv \pm w$, wenn ja faktorisiere gemäß a) break

- Bestimmen des Jacobi-Symbols ist leicht

  (wenn *p* und *q* unbekannt: mittels quadratischem Reziprozitätsgesetz)

  Aber Anmerkung : Wenn $\left(\frac{x}{n}\right) = 1$, bestimmen, ob $x \in QR_n$, ist schwer

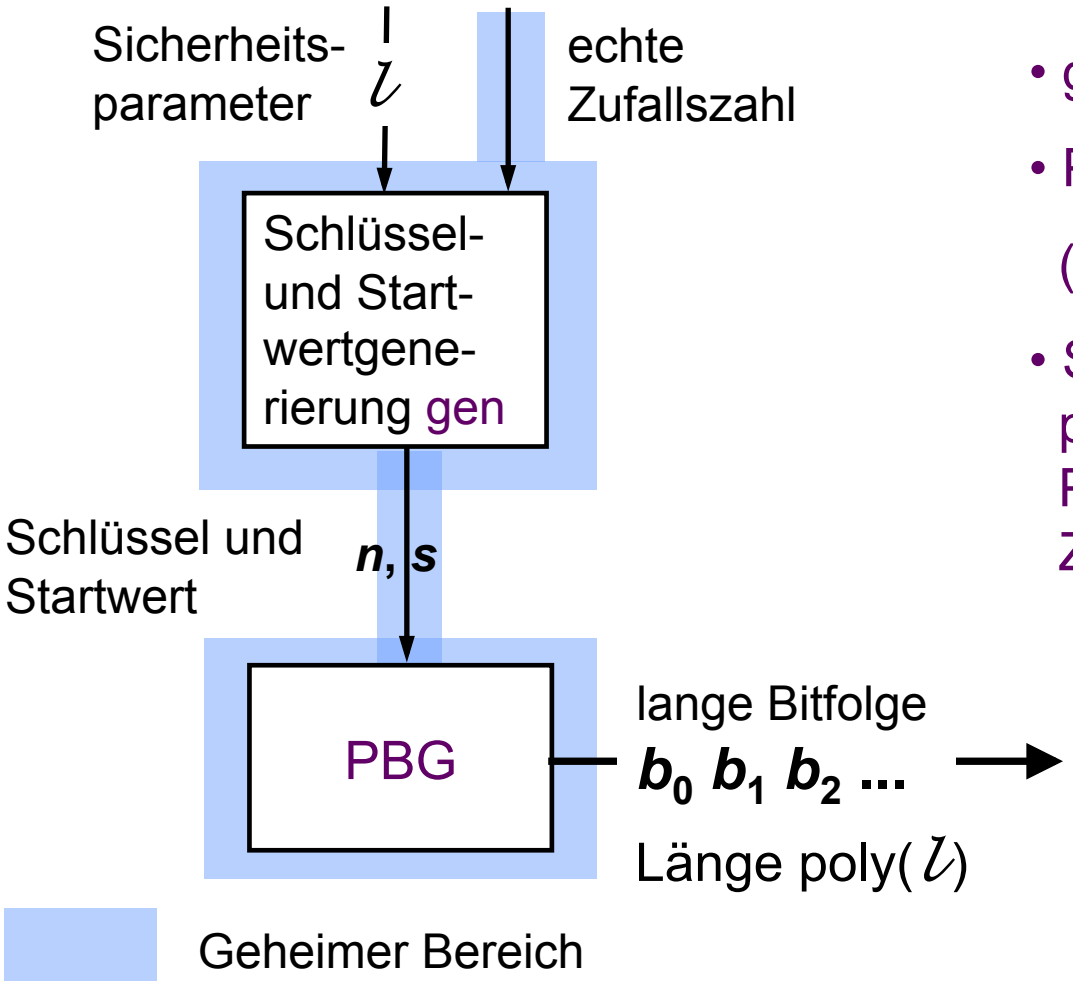
  (d.h. geht nicht wesentlich besser als raten)

QRA

Der s^2-mod-n-Pseudozufallsbitfolgengenerator (PBG)

Idee: kurzer Startwert (seed) → lange Bitfolge (soll zufällig sein aus Sicht von polynomialen Angreifern)

Schema:



Forderungen:

- gen und PBG sind effizient
- PBG ist deterministisch

  (⇒ Folge reproduzierbar)
- Sicher: Kein probabilistischer polynomialer Test kann PBG-Folgen von echten Zufallsfolgen unterscheiden

s^2-mod-n-Generator

Verfahren

- Schlüsselwert: p,q prim, groß, $\equiv 3 \bmod 4$
  $n = p \cdot q$
- Startwert: $s \in Z_n^*$
- PBG: $s_0 := s^2$
  $s_{i+1} := s_i^2$ $\qquad b_i := s_i \bmod 2$
  ... (letztes Bit)
  ...

Beispiel: $n = 3 \cdot 11 = 33$, $s = 2$

Index	0	1	2	3	4
s_i :	4	16	25	31	4
b_i :	0	0	1	1	0

$16^2 \bmod 33$
$= 8 \cdot 32 = 8 \cdot (-1) = 25$

$25^2 = (-8)^2 \equiv 64 \equiv 31$

$31^2 = (-2)^2 = 4$

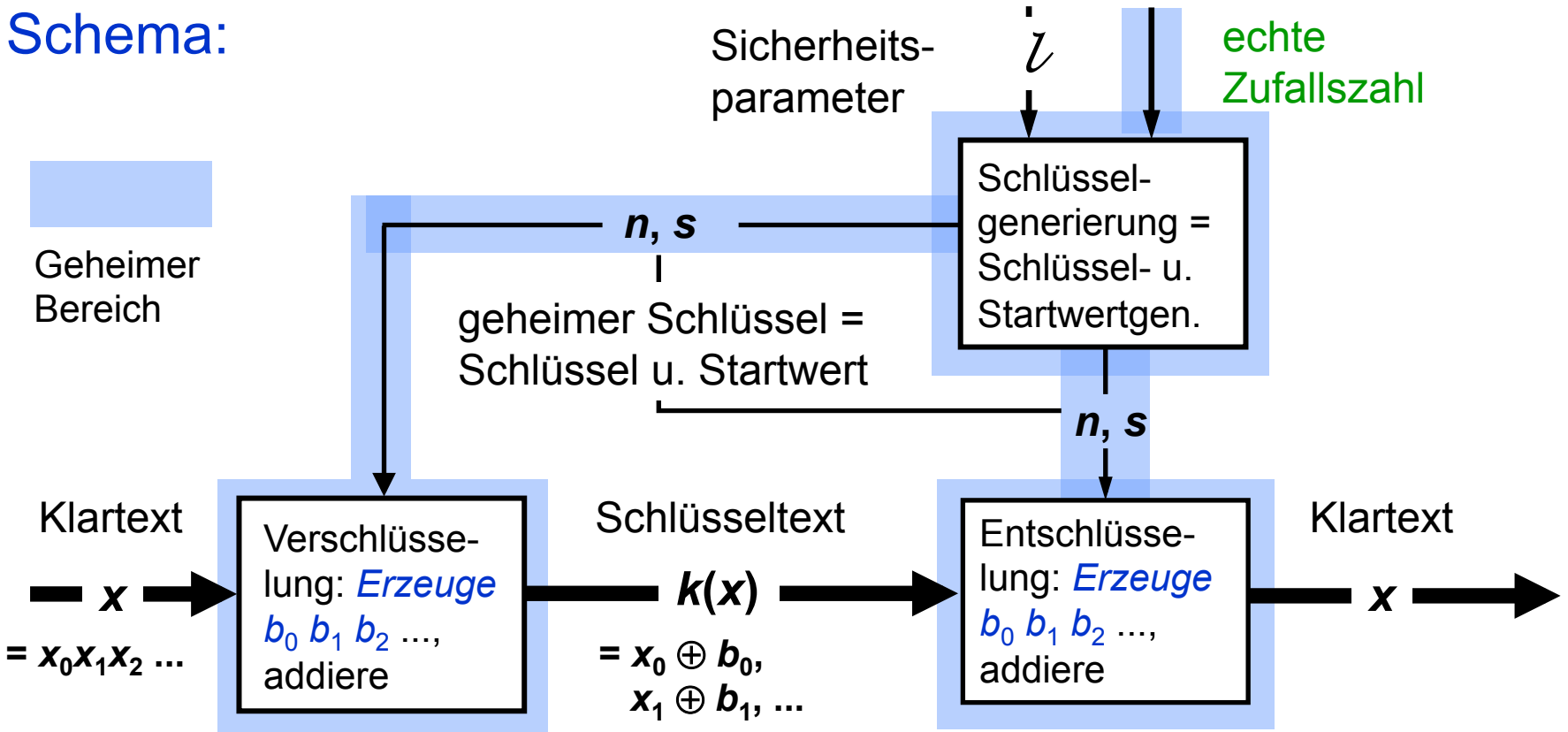
Anmerkung: Periodenlänge bei großen Zahlen kein Problem
(Blum / Blum / Shub 1983 / 86)

s^2-mod-n-Generator als symmetrisches Konzelationssystem

Zweck: Anwendung als symmetrisches Konzelationssystem:
„Pseudo-one-time-pad“

Vgl. one-time-pad: Addiere lange echte Zufallsbitfolge mit Klartext
Pseudo-one-time-pad: Addiere lange Pseudozufallsfolge mit Klartext

Schema:



s^2-mod-n-Gener. als symm. Konzelationssystem: Sicherheit

Idee:

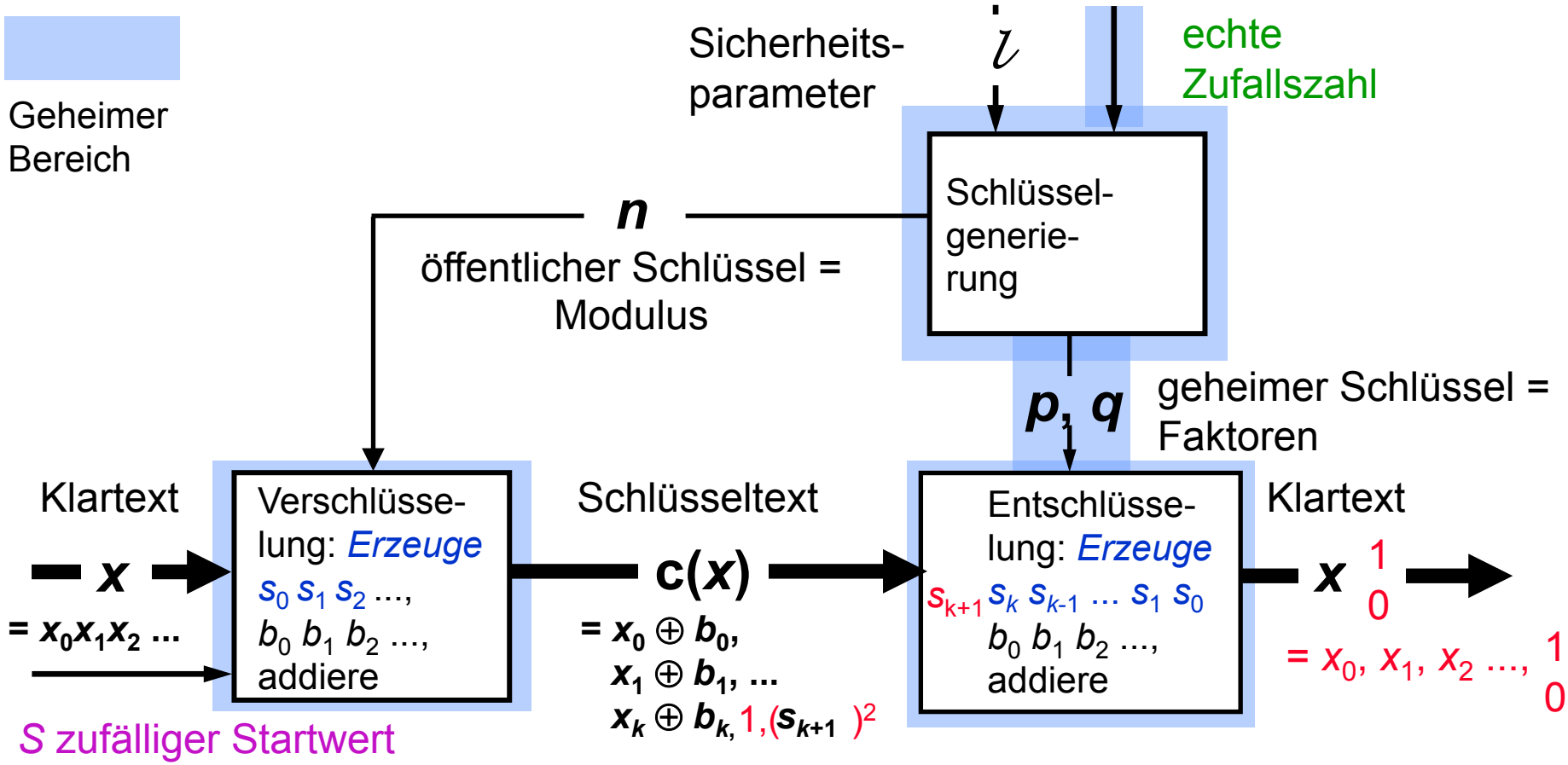
Wenn kein probabilistischer polynominaler Test Pseudozufallsfolgen von echten Zufallsfolgen unterscheiden kann, dann ist Pseudo-one-time-pad gegen polynomiale Angreifer so gut wie echtes one-time-pad.

(Sonst ist der Angreifer ein Test !)

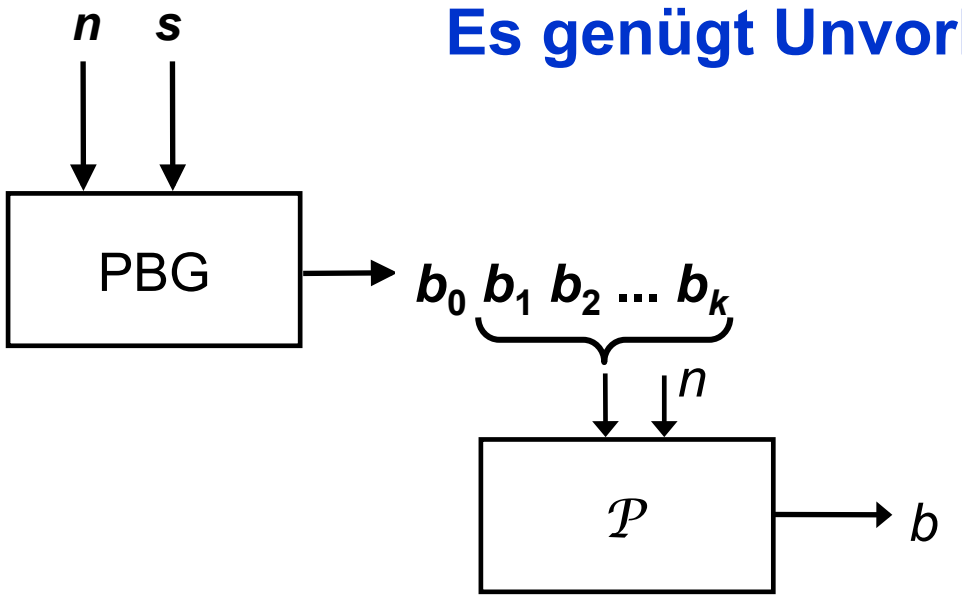
Konstruktion geht also mit jedem guten PBG

s^2-mod-n-Generator als asymm. Konzelationssystem

gewählter Schlüsseltext-Klartext-Angriff



Sicherheit des s^2-mod-n-Generators (1)



Es genügt Unvorhersagbarkeit nach links

s^2-mod-n-Generator ist kryptographisch stark: $\Leftrightarrow$

$\forall\ \mathcal{P}$ { Prädiktor für b_0 }

$\forall$ Konstanten δ, $0 < \delta < 1$ { Dichte der „schlechten“ n }

$\forall\ t \in \mathbb{N}$: { Grad des Polynoms }

sofern l (= $|n|$) genügend groß gilt: Für alle Schlüssel n bis auf höchstens δ-Anteil

$$W(b_0 = \mathcal{P}(n, b_1 b_2 \ldots b_k) \mid s \in Z_n^* \text{ zufällig}) < \frac{1}{2} + \frac{1}{l^t}$$

Sicherheit des s^2-mod-n-Generators (2)

Beweis : Durch Widerspruch zur QRA in 2 Schritten

Ann.: s^2-mod-n-Generator sei schwach,
d.h. es gibt Prädiktor $\mathcal{P}$, der b_0 zu $b_1\ b_2\ b_3$ ... mit ε-Vorteil rät.

1. Schritt: Transformiere $\mathcal{P}$ in $\mathcal{P}^*$, das mit ε-Vorteil das letzte Bit von s_0 zu gegebenem s_1 aus QR_n rät.

Gegeben s_1.

Bilde $b_1\ b_2\ b_3$ ... mit s^2-mod-n-Generator, wende $\mathcal{P}$ auf diese Folge an. $\mathcal{P}$ rät b_0 mit ε-Vorteil. Genau dies ist das Ergebnis von $\mathcal{P}^*$.

2. Schritt: Konstruiere aus $\mathcal{P}^*$ Verfahren $\mathcal{R}$, das mit ε-Vorteil rät, ob gegebenes s^* mit Jacobi-Symbol +1 ein Quadrat ist.

Gegeben s^*. Setze $s_1 := (s^*)^2$.

Wende $\mathcal{P}^*$ auf s_1 an. $\mathcal{P}^*$ rät letztes Bit von s_0 mit ε-Vorteil.

Dabei sind s^* und s_0 Wurzeln von s_1; $s_0 \in QR_n$.

Also $s^* \in QR_n \Leftrightarrow s^* = s_0$

Letztes Bit b^* von s^* und geratenes b_0 von s_0 genügt zum richtigen Raten, da

Sicherheit des s^2-mod-n-Generators (3)

1) Wenn $s^* = s_0$, dann $b^* = b_0$
2) zu zeigen: Wenn $s^* \neq s_0$, dann $b^* \neq b_0$

   Wenn $s^* \neq s_0$ gilt wegen gleichen Jacobi-Symbolen
   $s^* \equiv -s_0 \bmod n$

   also $s^* = n - s_0$ in Z

   n ist ungerade, also haben s^* und s_0 verschiedene letzte Bits

Das konstruierte $\mathcal{R}$ steht im Widerspruch zur QRA.

Anmerkungen:

1) Man kann $O(\log(\mathcal{l}))$ statt 1 Bit pro Quadrierschritt nehmen.
2) Es gibt einen komplizierteren Beweis, dass s^2-mod-n-Generator unter Faktorisierungsannahme sicher

Sicherheit von PBGs genauer (1)

Forderungen an PBG:

„Stärkste“ Forderung: PBG besteht *jeden* probabilistischen Test T polynomieller Laufzeit.

besteht = Folgen des PBG können von keinem probabilistischen Test polynomieller Laufzeit von echten Zufallsbitfolgen mit signifikanter Wahrscheinlichkeit unterschieden werden.

probabilistischer Test polynomieller Laufzeit = probabilistischer polynomiell zeitbeschränkter Algorithmus, der jeder Eingabe aus $\{0,1\}^*$ eine reelle Zahl aus [0,1] zuordnet. (Wert hängt im Allgemeinem von der Folge der Zufallsentscheidungen ab.)

Sei α_m der durchschnittliche (bzgl. Gleichverteilung) Wert, den T einer zufälligen *m*-Bit-Kette zuordnet.

Sicherheit von PBGs genauer (2)

Ein PBG besteht T gdw.

Für alle $t > 0$ liegt für genügend große l der Durchschnitt (über alle Startwerte der Länge l), den T der von PBG generierten poly(l)-Bit-Kette zuordnet, in $\alpha_{poly(l)} \pm 1/l^t$

Zu dieser „stärksten“ Forderung sind die 3 folgenden äquivalent (aber leichter beweisbar):

Für jede erzeugte endliche Anfangs-Bitkette, bei der ein beliebiges (das rechte, linke) Bit fehlt, kann jeder polynomiell zeitbeschränkte Algorithmus P (Prädiktor) das fehlender Bit „nur raten“.

Beweisidee für: Aus jeder dieser 3 Forderungen folgt die „stärkste“

Einfacher Teil: konstruiere Test aus Prädiktor

Schwieriger Teil: konstruiere Prädiktor aus Test

Sicherheit von PBGs genauer (3)

Bew. (indirekt): Konstruiere aus dem Test T den Prädiktor P.

Für ein $t>0$ und unendlich viele ι liegt der Durchschnitt (über alle Startwerte der Länge ι), den T der von PBG generierten poly(ι)-Bit-Kette zuordnet (z.B. oberhalb) $\alpha_{poly(\iota)} \pm 1/\iota^t$. T Bitkette aus 2 Teilen vorwerfen: $j+k$ = poly(ι)

echt zufällig

A={r_1 ... r_j r_{j+1} b_1 ... b_k} erhalten Werte näher bei $\alpha_{poly(\iota)}$

B={r_1 ... r_j b_0 b_1 ... b_k} erhalten Werte weiter weg,

von PBG generiert z.B. höher

Prädiktor für Bitkette b_1 ... b_k folgendermaßen:

T auf {r_1 ... r_j 0 b_1 ... b_k} schätze α^0

T auf {r_1 ... r_j 1 b_1 ... b_k} schätze α^1

Rate b_0 = 0 mit Wahrscheinlichkeit 1/2 + 1/2 (α^0- α^1)

(Genauer: L. Blum, M. Blum, M. Shub: A simple unpredictable Pseudo-Random Number Generator; SIAM J. Comput. 15/2 (May 1986) Seite 375f)

Zusammenfassung PBG und Einstieg GMR

Erinnerung:

s^2-mod-n-Generator sicher gegen passive Angreifer bei beliebiger Nachrichtenverteilung

⟶ Begründung für Pfeil: Zufallszahl‘ im Bild asymmetrische Konzelationssysteme

⟶ Begriff merken: indeterministische Verschlüsselung (probabilistic encryption)

Begriffe:

Einwegfunktion

Einwegpermutation

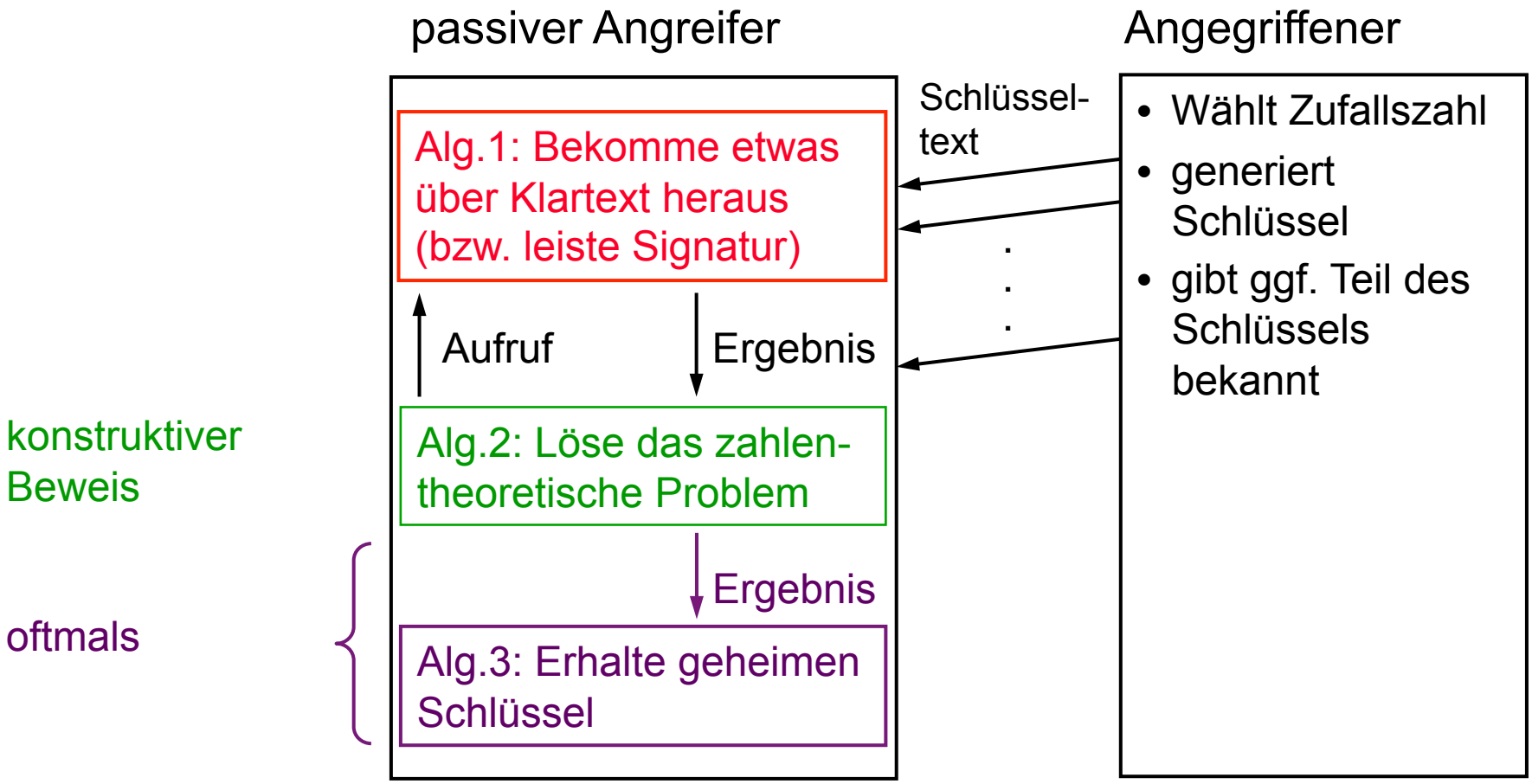
Einweg = so gut wie nirgends praktisch umkehrbar

ggf. mittels Geheimnis umkehrbar (trap door)

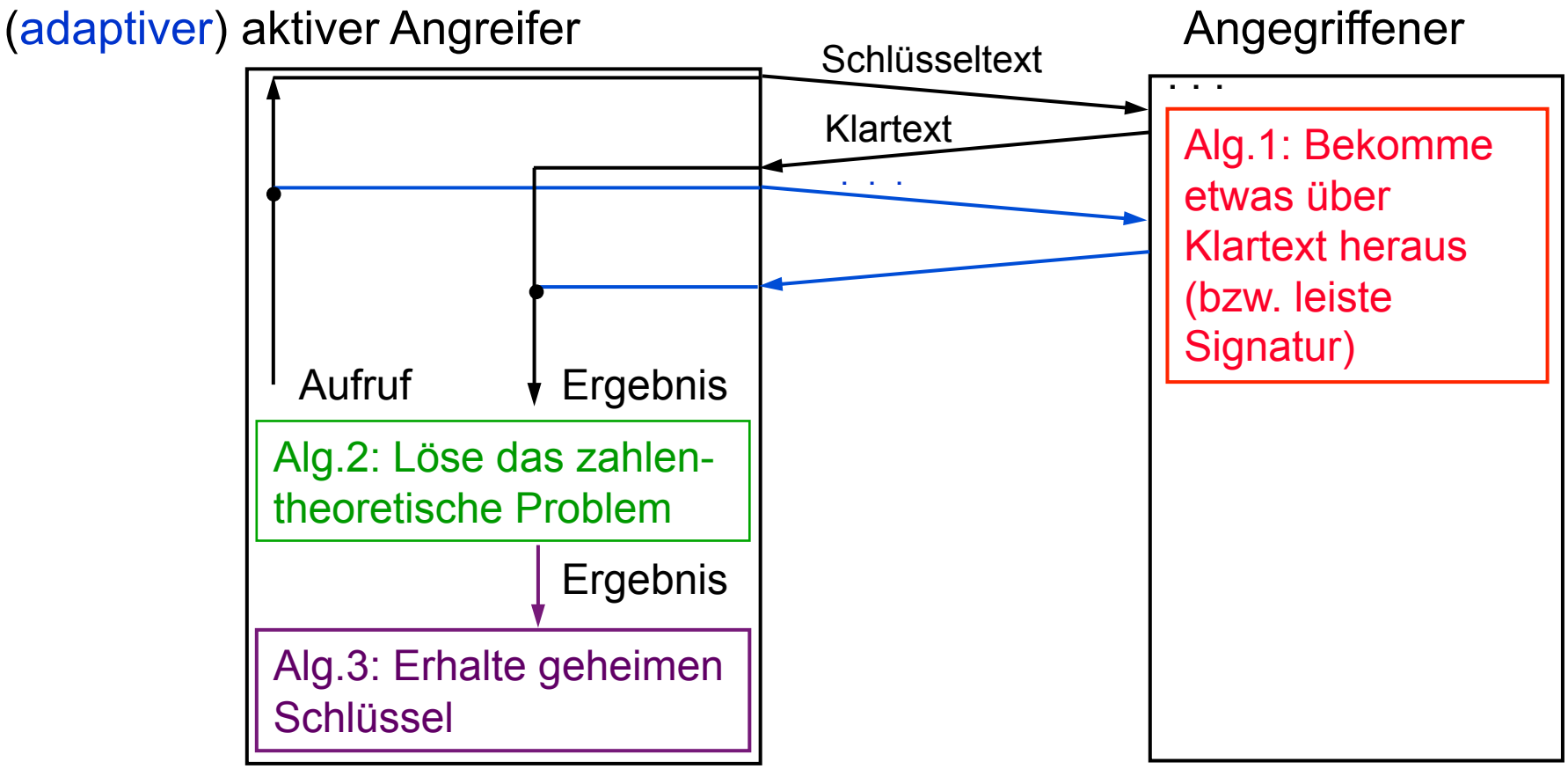
Einstieg:

aktiver Angriff auf s^2-mod-n-Generator als asymmetrisches Konzelationssystem

Schema von Sicherheitsbeweisen (1)



Schema von Sicherheitsbeweisen (2)



Beweisbar sichere Kryptosysteme gegen adaptive aktive Angriffe gibt es scheinbar nicht.

Konstruktiver Sicherheitsbeweis scheint ein Spiel mit dem Feuer zu sein.

Warum Trugschluss ?

Angreifer	Angegriffener
Alg.1: uniform für beliebige Schlüssel	Alg.1: nicht uniform: nur eigener Schlüssel
Alg.2: muss Uniformität erfordern	

GMR – Signatursystem

Shafi Goldwasser, Silvio Micali, Ronald Rivest:
A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks;
SIAM J. Comput. 17/2 (April 1988) 281 – 308

Kernideen

1) Abbildung einer zufällig gewählten, einmalig verwendeten Referenz $\mathcal{R}$.
2) Aus einer Menge kollisionsresistenter Permutationen (die mittels Geheimnis umkehrbar sind) wird jeder Nachricht *m* eine Permutation zugeordnet.

$$\mathcal{R} \underset{\mathcal{F}_{n,m}\ (\mathrm{Sig}_m^{\mathcal{R}})}{\overset{\mathcal{F}_{n,m}^{-1}\ (\mathcal{R})}{\rightleftarrows}} \mathrm{Sig}_m^{\mathcal{R}}$$

GMR – Signatursystem (1)

Konsequenz

„Variation von *m*“ (aktiver Angriff) bedeutet nun auch
„Variation von $\mathcal{R}$“ – einer zufällig gewählten Referenz, die dem Angreifer bei seiner Wahl von *m* unbekannt ist

Probleme

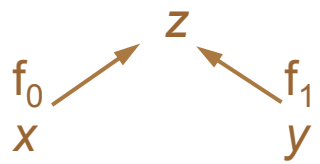
1) Sicherstellung der Originalität der zufällig gewählten Referenz
2) Konstruktion der kollisionsresistenten Permutationen (die mittels Geheimnis umkehrbar sind) in Abhängigkeit von den Nachrichten

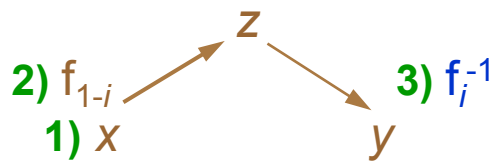
Lösung von Problem 2

Idee Wähle 2 kollisionsresistente Permutationen f_0, f_1 (die mittels Geheimnis umkehrbar sind) und setze $\mathcal{F}_{n,m}$ aus ihnen zusammen.
{Zur Vereinfachung f_0 statt $f_{n,0}$ und f_1 statt $f_{n,1}$}

Def. Zwei Permutationen f_0,f_1 heißen kollisionsresistent gdw.
bel. *x*,*y*,*z* zu finden mit $f_0(x) = f_1(y) = z$ ist schwierig

Bem. Beh. kollisionsresistent ⇒ einweg
Bew. (ind.): Sei f_i nicht einweg: 1) Wähle *x*; 2) $f_{1-i}(x) = z$; 3) $f_i^{-1}(z) = y$





GMR – Signatursystem (2)

Zusammensetzung:

Für $m = b_0 b_1 ... b_k$ $(b_0,...,b_k \in \{0,1\})$ sei

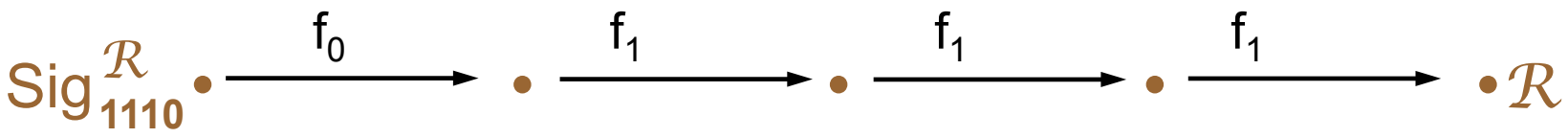
$$\mathcal{F}_{n,m} := f_{b_0} \circ f_{b_1} \circ ... \circ f_{b_k}$$

$$\mathcal{F}_{n,m}^{-1} := f_{b_k}^{-1} \circ ... \circ f_{b_1}^{-1} \circ f_{b_0}^{-1}$$

Signieren: $\mathcal{R} \xrightarrow{f_{b_0}^{-1}} f_{b_0}^{-1}(\mathcal{R}) \xrightarrow{f_{b_1}^{-1}} ... \xrightarrow{f_{b_k}^{-1}} f_{b_k}^{-1}(...(f_{b_0}^{-1}(\mathcal{R}))...) =: \mathrm{Sig}_m^{\mathcal{R}}$

Testen: $\mathrm{Sig}_m^{\mathcal{R}} \xrightarrow{f_{b_k}} f_{b_k}(\mathrm{Sig}_m^{\mathcal{R}}) \xrightarrow{f_{b_{k-1}}} ... \xrightarrow{f_{b_0}} f_{b_0}(...(f_{b_k}(\mathrm{Sig}_m^{\mathcal{R}}))...) = \mathcal{R}$?

Beispiel:



GMR – Signatursystem (3)

Problem: Zwischenergebnisse der Überprüfung sind gültige Signaturen für Anfangsabschnitte der Nachricht *m*

Idee: Nachrichten präfixfrei codieren

Def. Eine Abbildung $<\bullet>: M \rightarrow M$ heißt präfixfrei
gdw. $\forall\, m_1, m_2 \in M: \forall\, b \in \{0,1\}^+: <m_1>b \neq <m_2>$
$<\bullet>$ injektiv

Bsp. für präfixfreie Abbildung
$0 \rightarrow 00$; $1 \rightarrow 11$; Endekennung 10

Präfixfreie Abbildung sollte in beide Richtungen effizient berechnet werden können.

Faktorisieren schwierig (1)

Satz: Wenn Faktorisieren schwierig, dann existieren kollisionsresistente Permutationenpaare

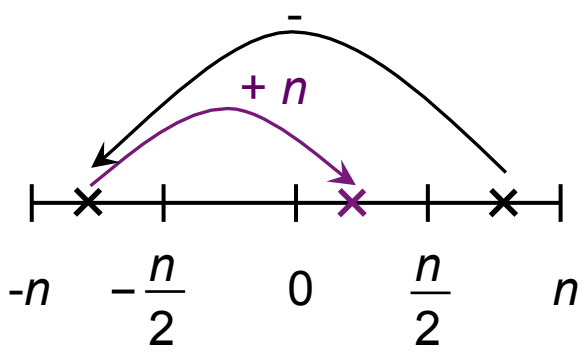
Beweis: Geheimnis: $p \bullet q = n$; $p \equiv_8 3$ und $q \equiv_8 7$ (Blum-Zahlen)

es gilt: $\left(\frac{-1}{n}\right) = 1$ $\quad -1 \notin QR_n$

$\left(\frac{2}{n}\right) = -1$

$$f_0(x) := \begin{cases} x^2 \bmod n \text{ , falls } < \frac{n}{2} \\ -x^2 \bmod n \text{ , sonst} \end{cases}$$

$$f_1(x) := \begin{cases} (2x)^2 \bmod n \text{, falls } < \frac{n}{2} \\ -(2x)^2 \bmod n \text{, sonst} \end{cases}$$



Definitionsbereich : $\{x \in Z_n^* \mid \left(\frac{x}{n}\right) = 1 \text{ , } 0 < x < \frac{n}{2} \}$

Faktorisieren schwierig (2)

zu zeigen : 1) Permutation = bijektive Abbildung mit Bild- = Urbildraum
2) Inverse mittels p,q leicht berechenbar
3) Falls es schnellen Kollisionsfinde-Algorithmus gibt, gibt es schnellen Faktorisierungsalgorithmus

$-1 \notin QR_n$

Da $(2y)^2 \in QR_n$ ist $x^2 \equiv_n -(2y)^2$ unerfüllbar

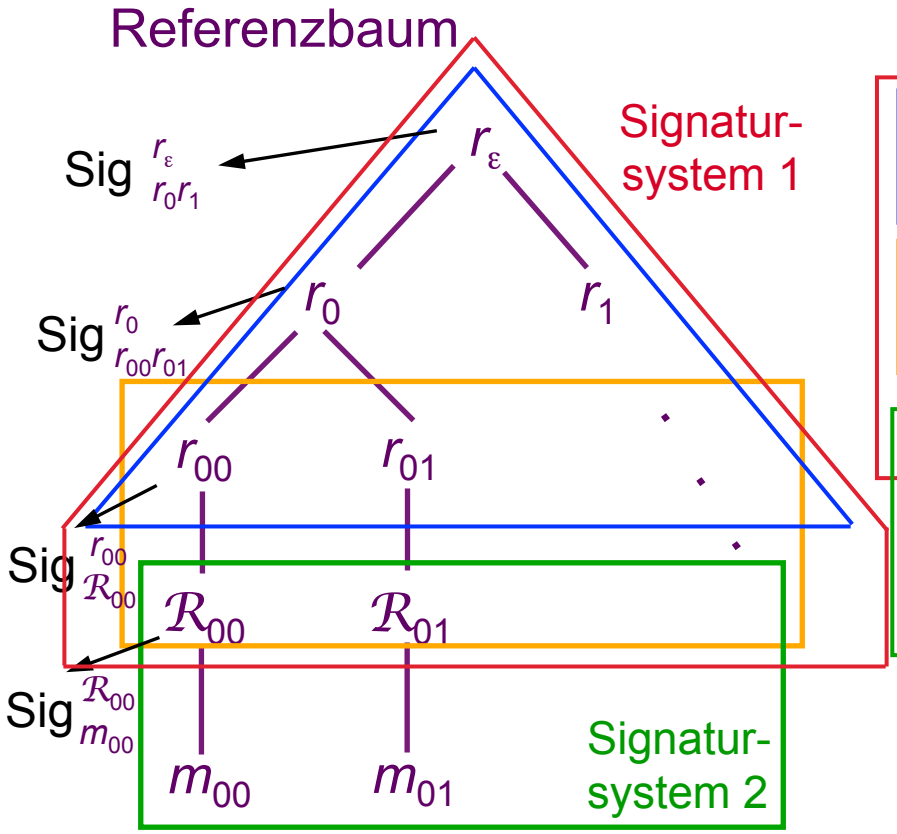
Also $x^2 \equiv_n (2y)^2 \Rightarrow (x+2y)(x-2y) \equiv_n 0$.

Da $\left(\frac{x}{n}\right) = 1$ und $\left(\frac{\pm 2y}{n}\right) = -1$ folgt

$x \not\equiv_n \pm 2y$

Also liefert ggT $(x \pm 2y, n)$ einen nichttrivialen Faktor von n, d.h. p oder q.

Lösung von Problem 1 (1)



Erst nach m_i
wird $\mathcal{R}_i$ bekannt.

Generieren (≈ Signieren)

Signatursystem 1
kein
aktiver Angriff

$$\text{Sig}\ {}^{r_j}_{r_{j0}r_{j1}} = \mathcal{F}^{-1}_{n,<r_{j0}r_{j1}>}(r_j)$$

$$\text{Sig}\ {}^{r_i}_{\mathcal{R}_i} = \mathcal{F}^{-1}_{n,<\mathcal{R}_i>}(r_i)$$

Referenzen $\mathcal{R}$;

$$\text{Sig}\ {}^{\mathcal{R}_i}_{m_i} = \mathcal{F}^{-1}_{n',<m_i>}(\mathcal{R}_i)$$

indeterministisches
Signatursystem 2

Überprüfen (≈ Testen)

$$\mathcal{F}_{n,<r_{j0}r_{j1}>}(\text{Sig}\ {}^{r_j}_{r_{j0}r_{j1}}) = r_j\ ?$$

$$\mathcal{F}_{n,<\mathcal{R}_i>}(\text{Sig}\ {}^{r_i}_{\mathcal{R}_i}) = r_i\ ?$$

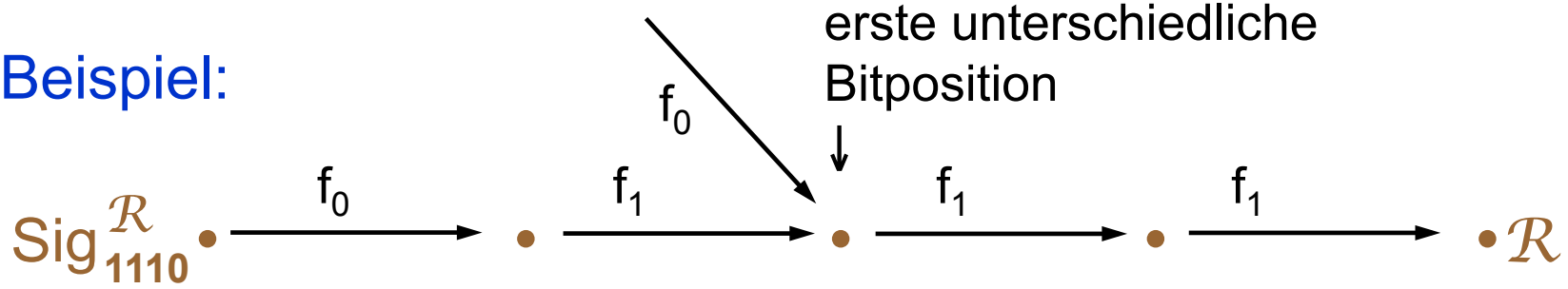
$$\mathcal{F}_{n',<m_i>}(\text{Sig}\ {}^{\mathcal{R}_i}_{m_i}) = \mathcal{R}_i\ ?$$

Lösung Problem 1 (2)

Beh. Sind die Permutationenpaare kollisionsresistent, kann vom adaptiven aktiven Angreifer nicht mal eine beliebige Nachricht mit GMR signiert werden.

Bew. Eine gefälschte Signatur führt entweder zu einer Kollision im Referenzbaum (Widerspruch) oder zu einer legalen zusätzlichen Signatur. Damit hat der Angreifer kollisionsresistente Permutation invertiert. Mit dieser Fähigkeit könnte er auch Kollisionen erzeugen (Widerspruch).

Beispiel:



Anmerkung

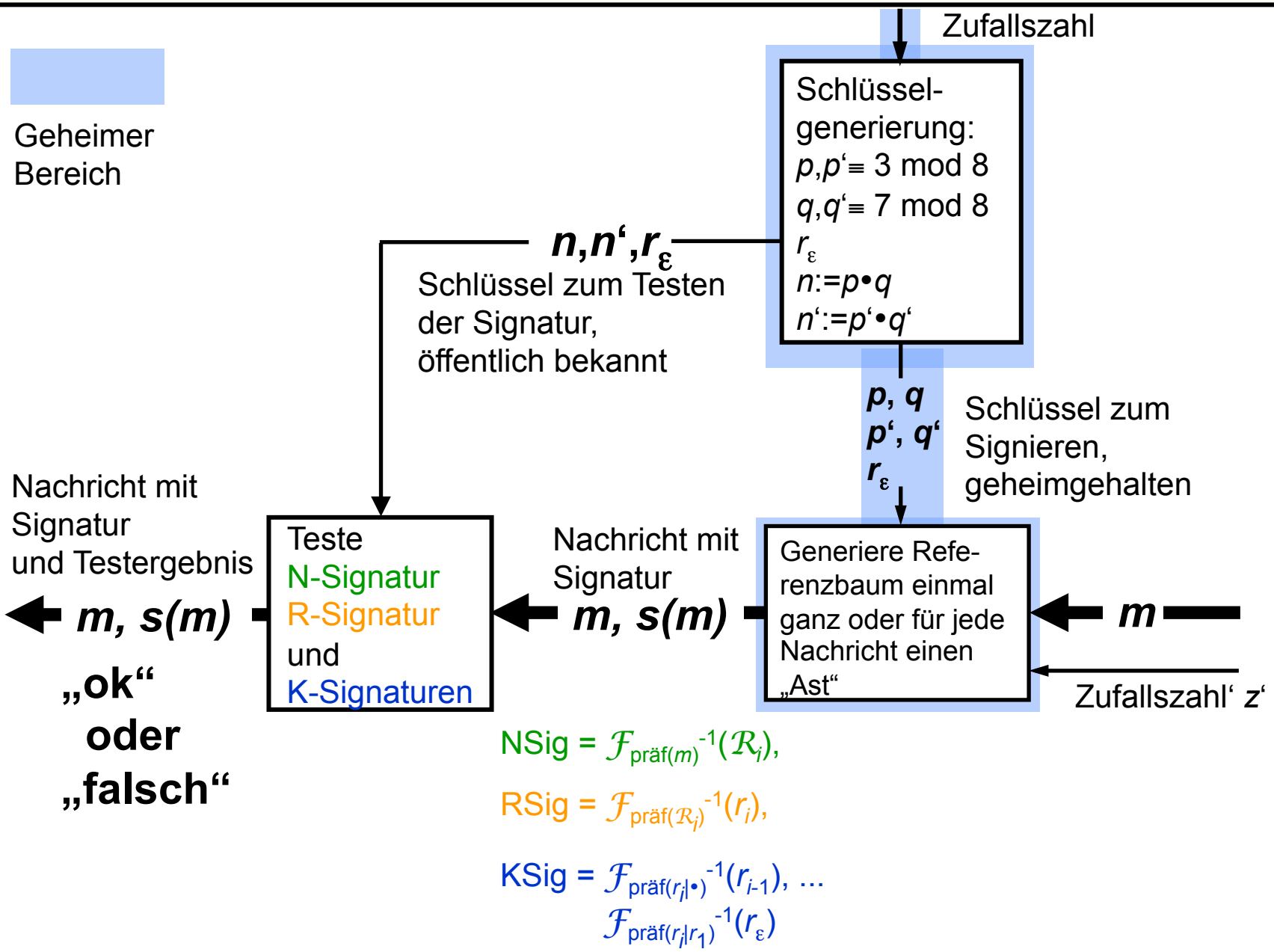
Das „Orakel“ (den Angegriffenen) wird man im Beweis dadurch los, dass der Angreifer den „halben“ Baum von unten oder (exklusiv) oben mit der gleichen Wahrscheinlichkeitsverteilung wie der Angegriffene erzeugen kann.

Lehre:

einmalig verwendete, zufällig gewählte Referenzen (vgl. one-time-pad) machen adaptiven aktiven Angriff wirkungslos

→ Pfeil erklärt (Zufallszahl z') im Bild Signatursystem

GMR Signatursystem



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 Z_n

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

Beweis (1)

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

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

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

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 $m \equiv_p 0$. Somit gilt Kongruenz für alle $m \in \mathbf{Z}_p$.

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 gemäß CRA auch bzgl. $p \cdot q = n$.

Daher gilt für alle $m \in Z_n$

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

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

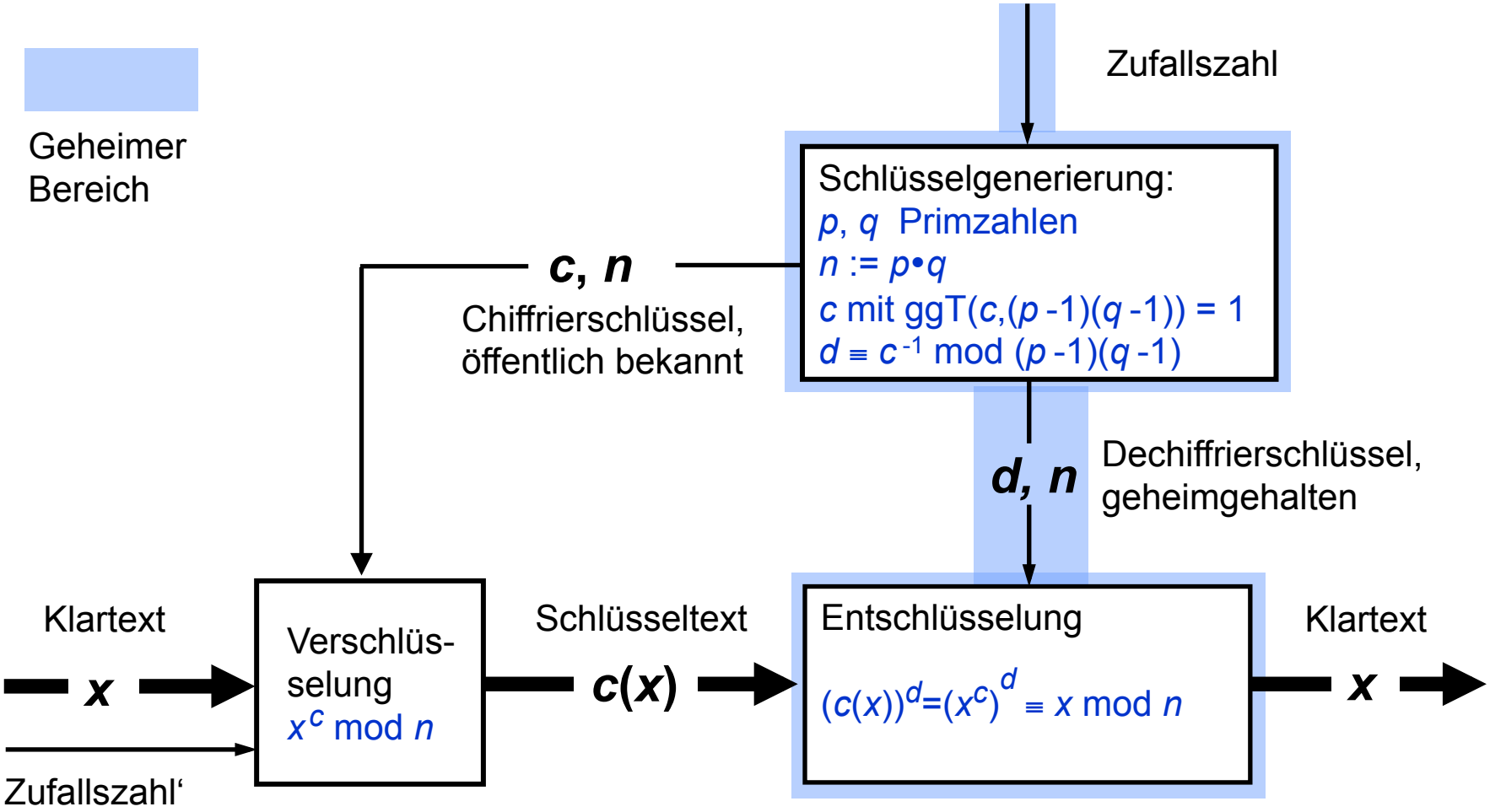
RSA als digitales Signatursystem

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

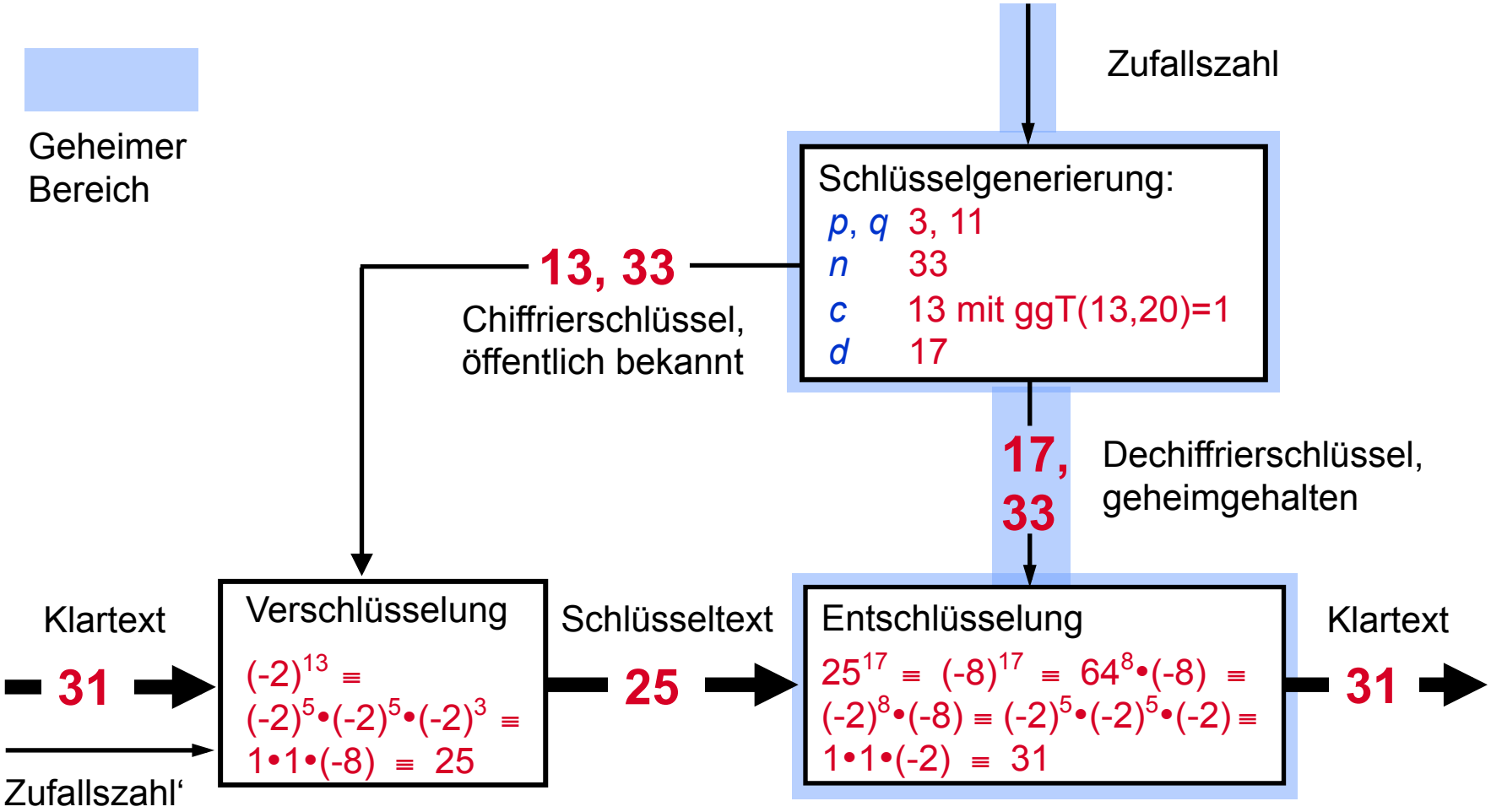
Signieren von m: $m^s \bmod n$

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

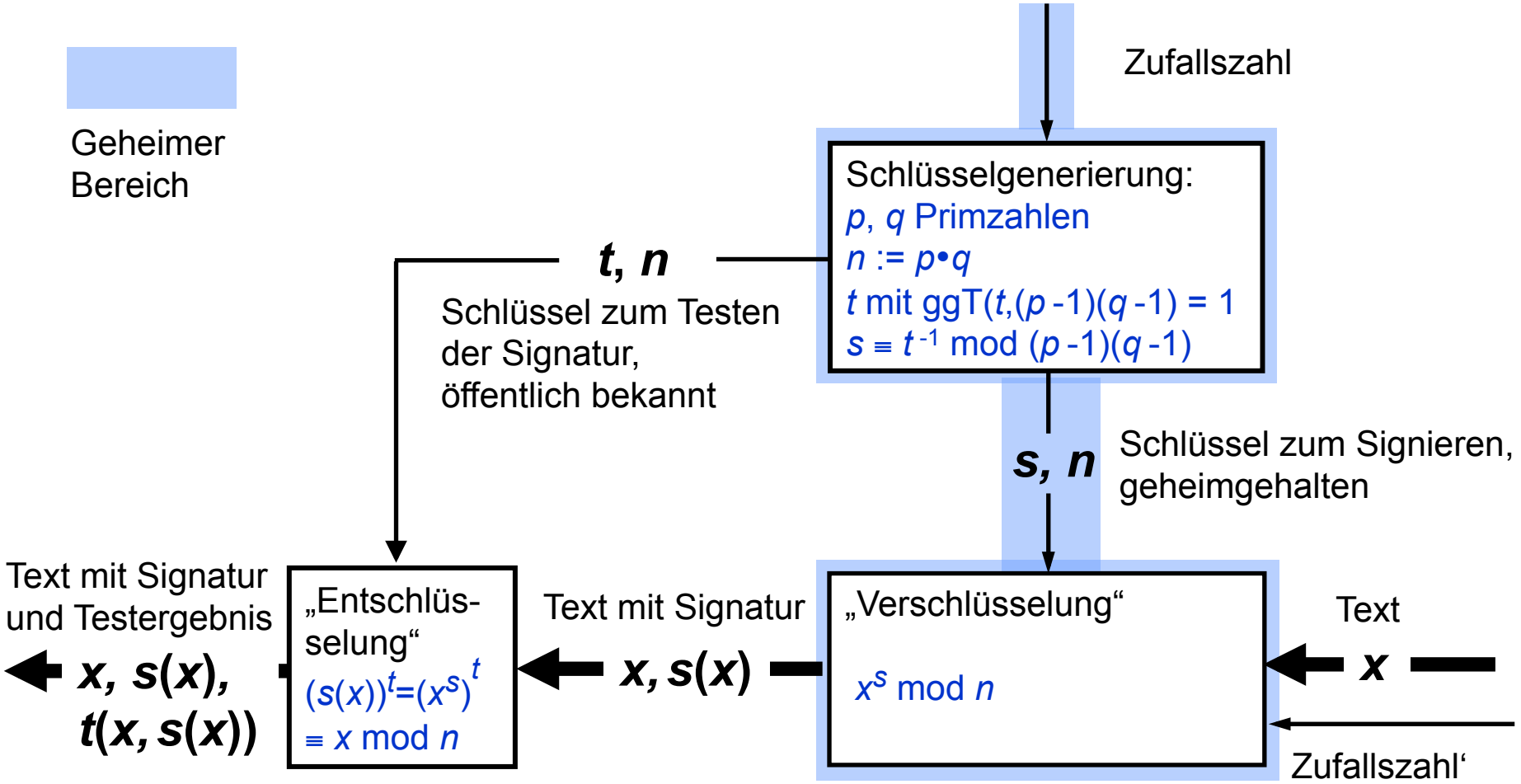
RSA als asymmetrisches Konzelationssystem: naiv



RSA als asymmetrisches Konzelationssystem: Beispiel



RSA als digitales Signatursystem: naiv



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

Angriff auf Konzelation mit RSA naiv: alternative Darstellung

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

abgehört

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

aus y selbst gebildet

entschlüsseln lassen

$$(u \bullet v)^d = u^d \bullet v^d$$

$$= x \bullet y$$

teile durch y, erhalte x

Angriff auf digitale Signatur mit RSA naiv

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

$$(x^s \bullet y)^t \equiv x \bullet y^t$$ gewählte Nachricht y

signieren lassen

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

teile durch y, erhalte x^s

Angriff auf dig. Signatur mit RSA naiv: alternative Darstellung

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

gewünschte Nachricht

$$(u \bullet v)^t = u^t \bullet v^t$$

gewählte Nachricht v

signieren lassen

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

$$= x^s \bullet v$$

teile durch v, erhalte x^s

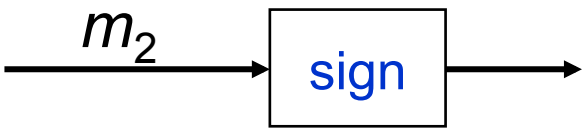
Hinführung zu den Davida-Angriffen

Einfache Version eines Davida-Angriffs: (auf RSA als Signatursystem)

1. Gegeben $Sig_1 = m_1^s$
   $Sig_2 = m_2^s$
   $\Rightarrow$ $Sig := Sig_1 \bullet Sig_2 = (m_1 \bullet m_2)^s$
   Neue Signatur erzeugt !
   (Passiver Angriff, dafür *m* nicht wählbar.)

2. Aktiv, gewünscht $Sig = m^s$
   Wähle m_1 beliebig; $m_2 := m \bullet m_1^{-1}$
   Lasse m_1, m_2 signieren.
   Weiter wie oben.

3. Aktiv, trickreicher (Moore) {siehe folgende Folie}
   „Blinding“ : Wähle *r* beliebig,
   $m_2 := m \bullet r^t$



$m_2^s = m^s \bullet r^{t \bullet s} = m^s \bullet r$

$\bullet r^{-1}$ $m^s = Sig$

Aktiver Angriff von Davida auf RSA

1.) asymmetrisches Konzelationssystem:

Entschlüsselung der gewählten Nachricht m^c

Angreifer	wählt Zufallszahl r, $0 < r < n$ bildet r^c mod n; dies ist gleichverteilt in [1, n-1] lässt Angegriffenen $r^c \bullet m^c \equiv_n prod$ entschlüsseln
Angegriffener	bildet $prod^d$ mod n
Angreifer	weiß, dass $prod^d \equiv_n (r^c \bullet m^c)^d \equiv_n r^{c \bullet d} \bullet m^{c \bullet d} \equiv_n r \bullet m$ teilt also $prod^d$ durch r und erhält so m.

Wenn das nicht geht: Faktorisiere n.

2.) digitales Signatursystem:

Signieren der gewählten Nachricht m.

Angreifer	wählt Zufallszahl r, $0 < r < n$ bildet r^t mod n; dies ist gleichverteilt in [1, n-1] läßt Angegriffenen $r^t \bullet m \equiv_n prod$ signieren
Angegriffener	bildet $prod^s$ mod n
Angreifer	weiß, dass $prod^s \equiv_n (r^t \bullet m)^s \equiv_n r^{t \bullet s} \bullet m^s \equiv_n r \bullet m^s$ teilt also $prod^s$ durch r und erhält so m^s.

Wenn das nicht geht: Faktorisiere n.

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

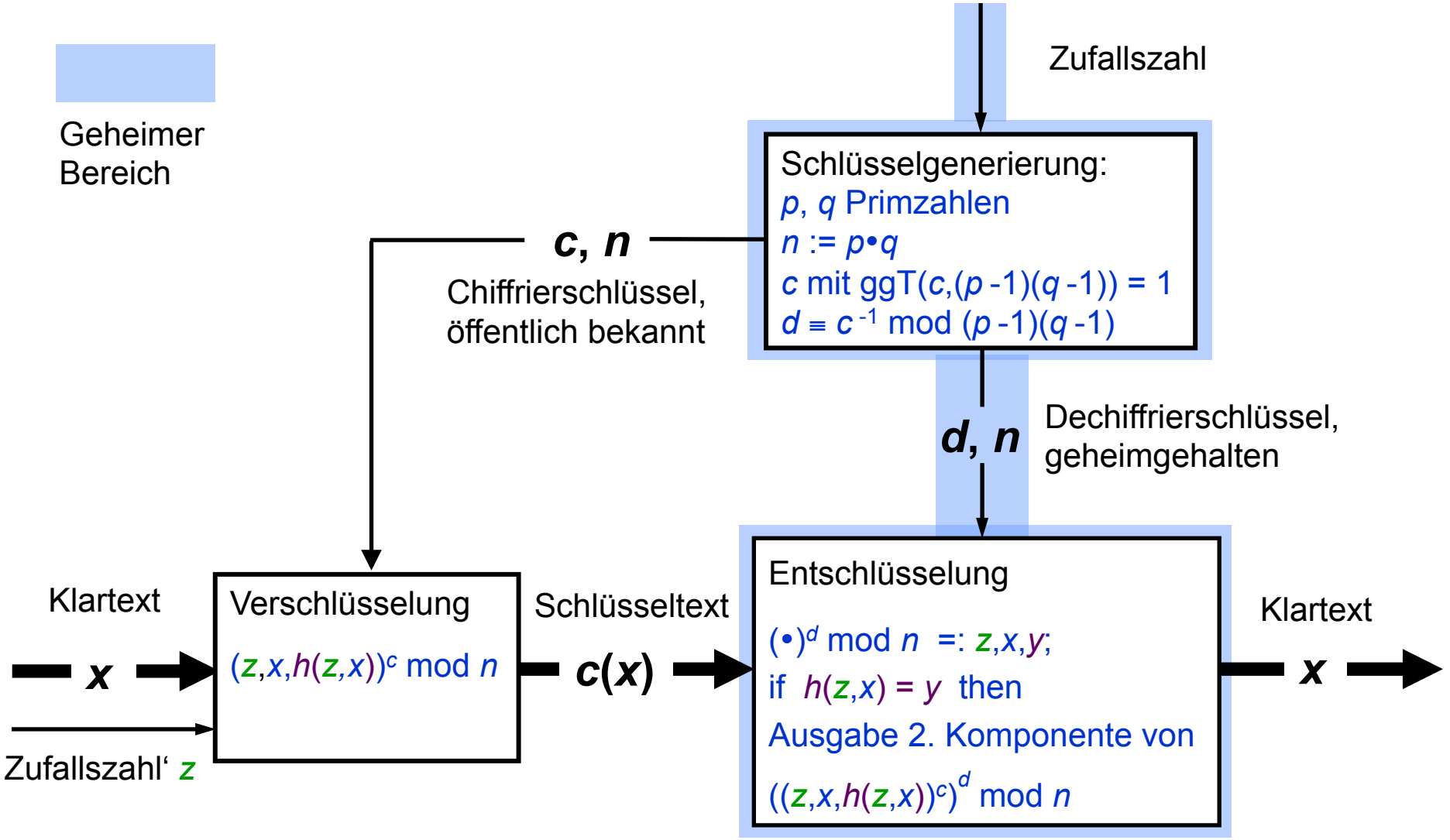
Vor dem Signieren wird auf die Nachricht h angewendet

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

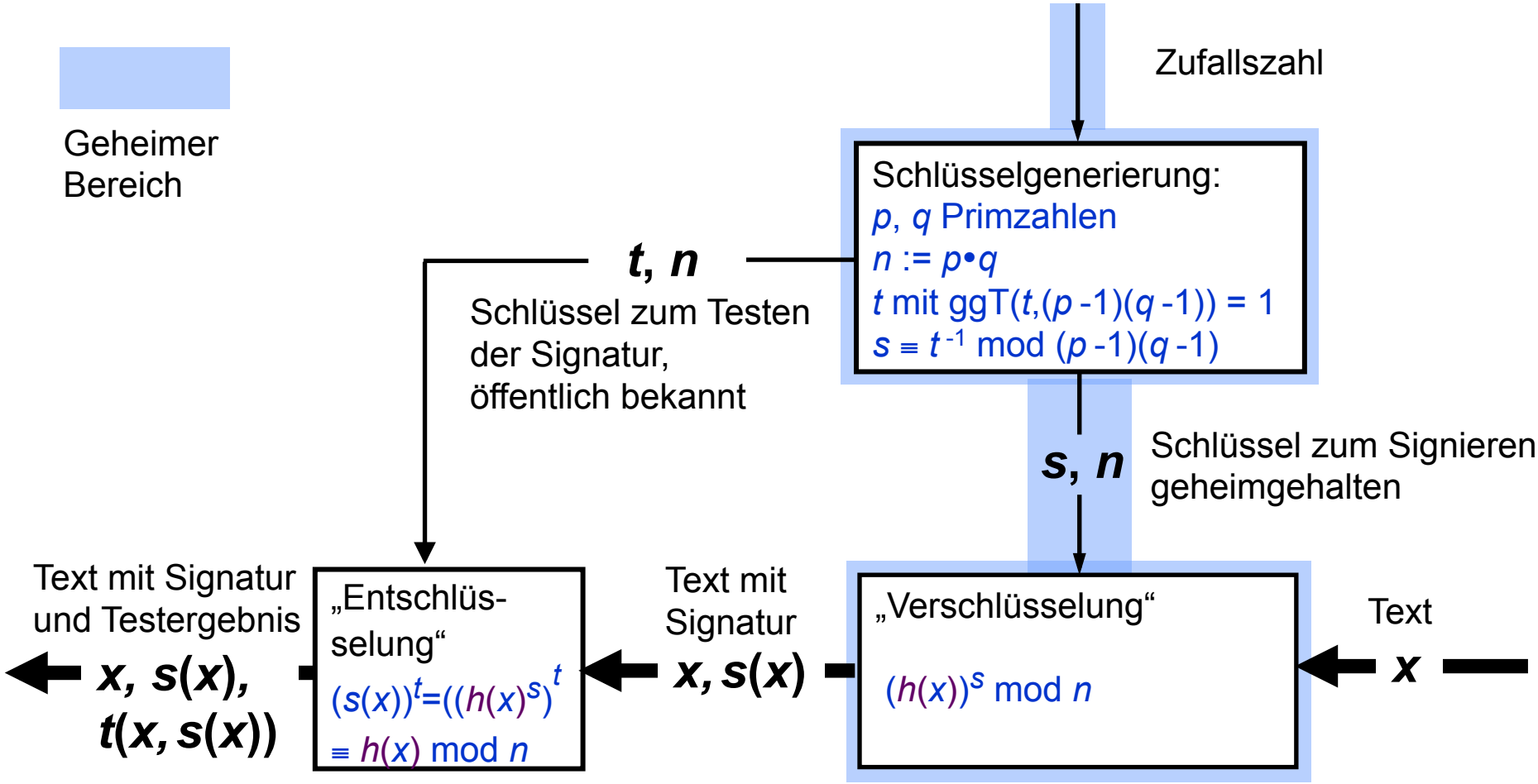
prüfe ob $h(m) = ((h(m))^s)^t \bmod n$

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

RSA als asymmetrisches Konzelationssystem



RSA als digitales Signatursystem



kollisionsresistente Hashfunktion *h*
- global bekannt -

Schnellere Berechnung der geheimen Operation

$y^d \equiv w$

mod *p*, *q* einzeln:

ein für allemal:

$$d_p := c^{-1} \bmod p\text{-}1 \quad \Rightarrow \quad (y^{d_p})^c \equiv y \bmod p$$

$$d_q := c^{-1} \bmod q\text{-}1 \quad \Rightarrow \quad (y^{d_q})^c \equiv y \bmod q$$

jedes mal:

Setze $w := \mathrm{CRA}(y^{d_p}, y^{d_q})$

Beweis:

$$\Rightarrow w^c \equiv \begin{cases} (y^{d_p})^c \equiv y \bmod p \\ (y^{d_q})^c \equiv y \bmod q \end{cases}$$

$$\Rightarrow w^c \equiv y \bmod n$$

Um wie viel schneller ?

Aufwand Exponentiation: $\approx l^3$

Aufwand 2 Exponentiationen halber Länge: $\approx 2 \cdot \left(\frac{l}{2}\right)^3 = \frac{l^3}{4}$

Aufwand CRA: 2 Multiplikationen $\approx 2 \cdot l^2$

1 Addition $\approx l$

Also: ≈ Faktor 4

unerheblich

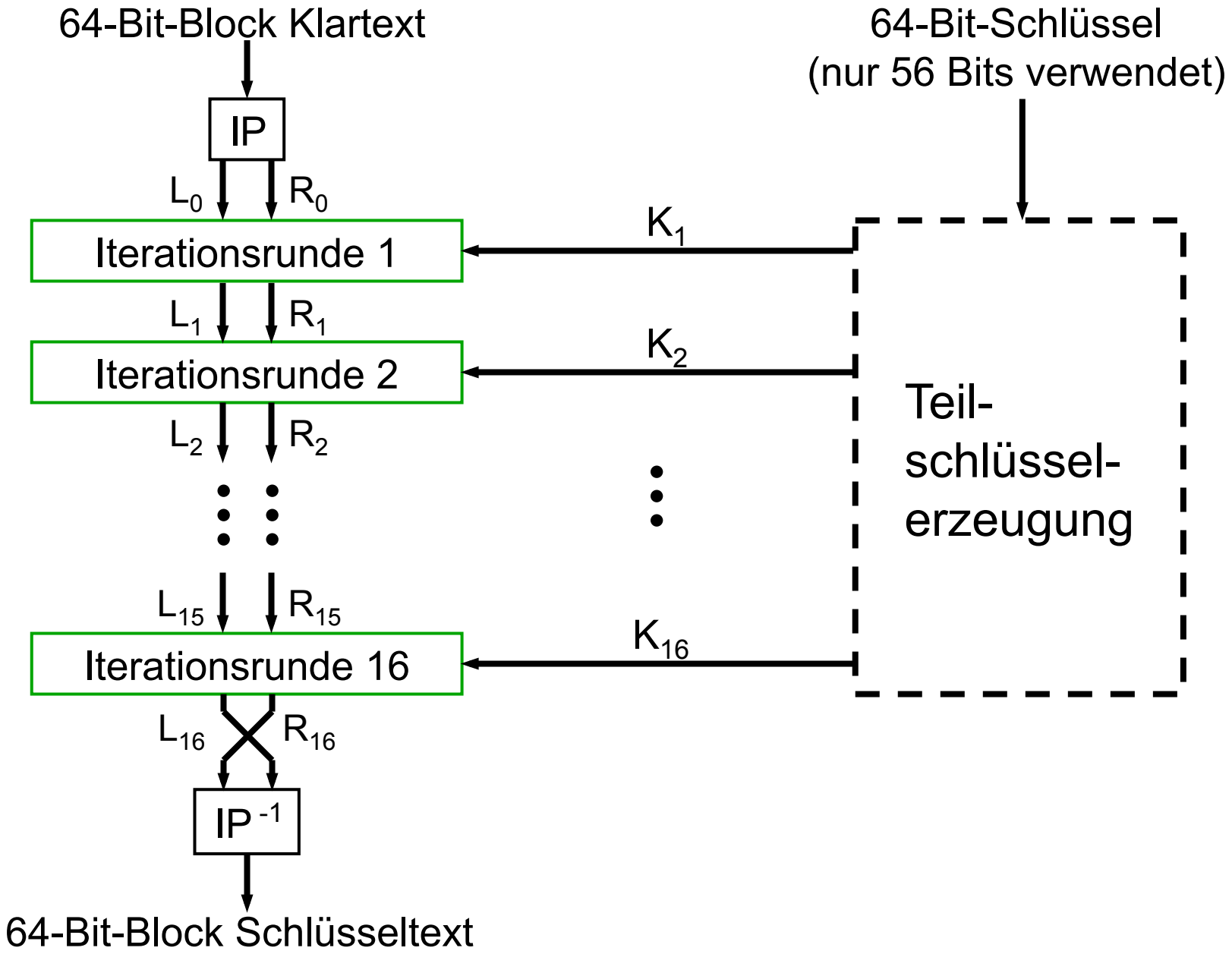
c-te Wurzeln sind eindeutig

Gezeigt : Jedes $y \in Z_n$ hat c-te Wurzel

$\Rightarrow$ Funktion $w \rightarrow w^c$ surjektiv

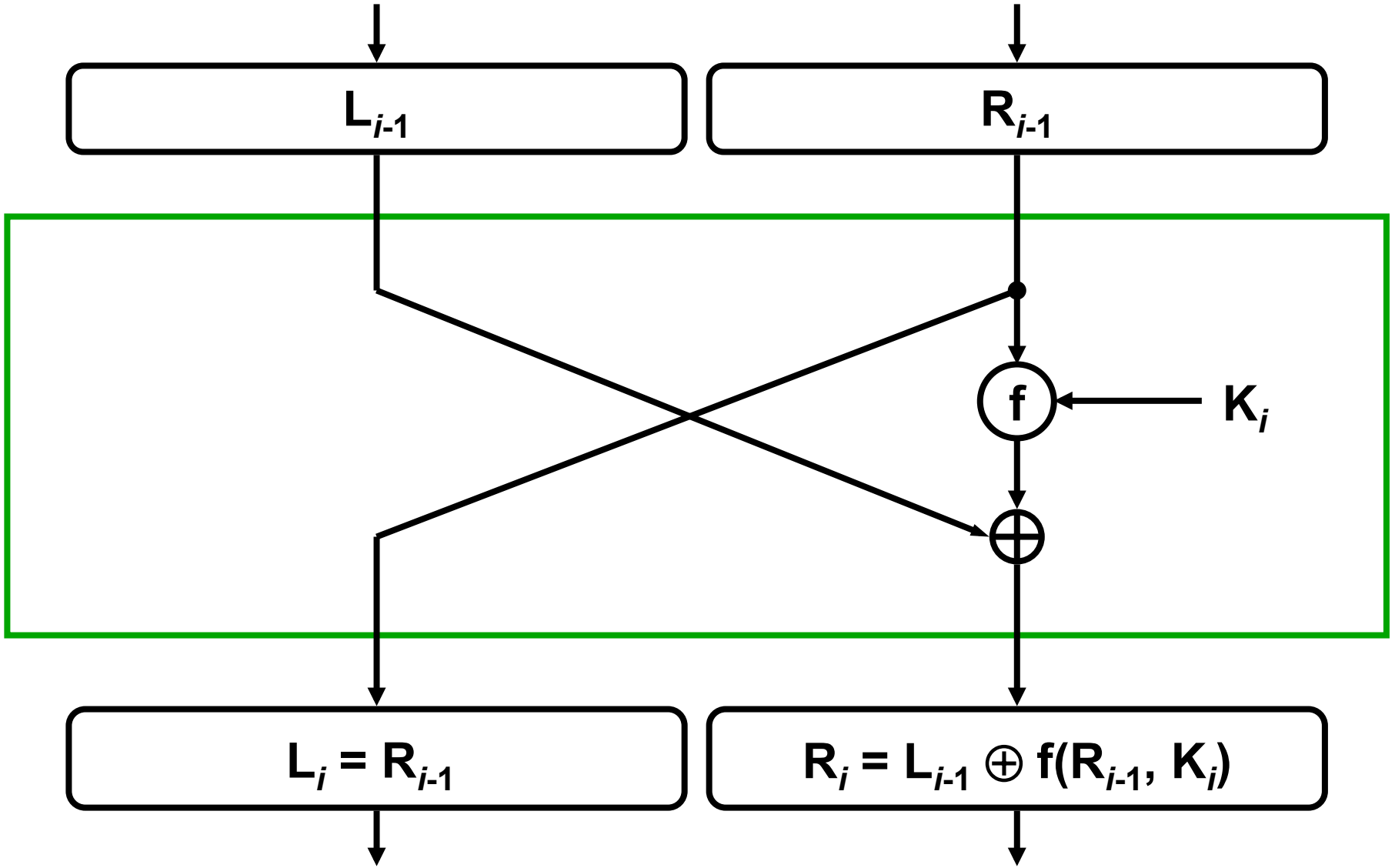
$\Rightarrow$ Auch injektiv.

Symmetrisches Kryptosystem DES

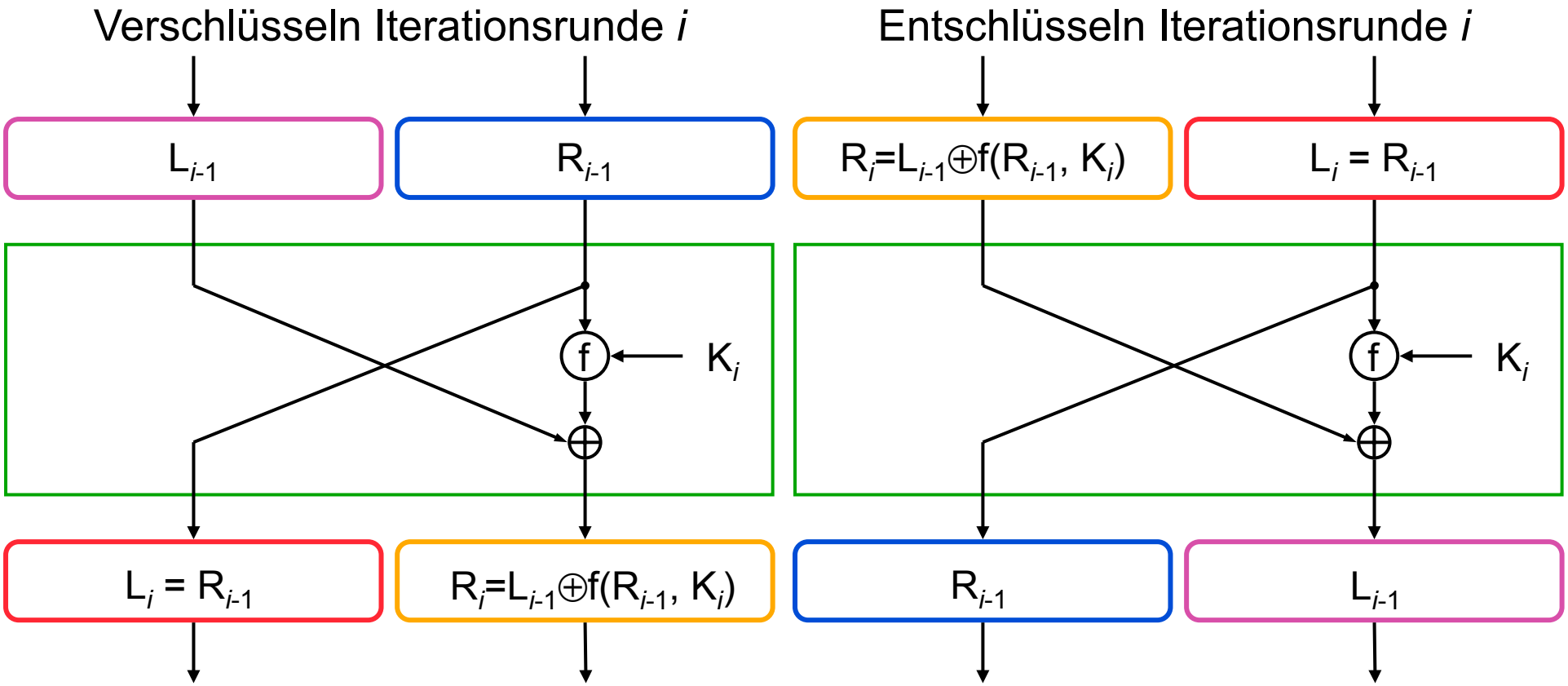


Eine Iterationsrunde

Feistel Chiffren



Entschlüsselungsprinzip



Entschlüsselungsprinzip

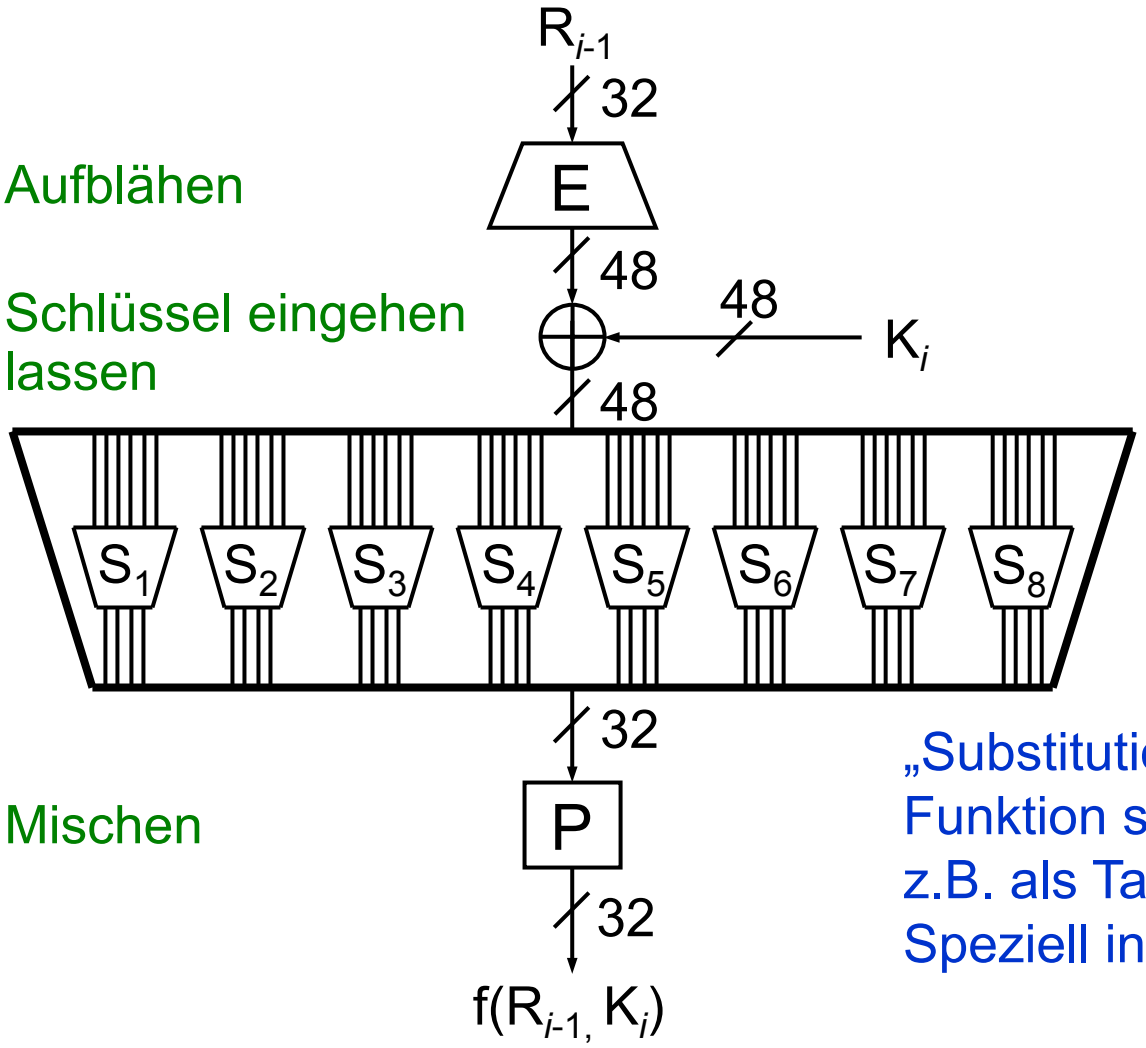
trivial

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

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

Ersetze R_{i-1} durch L_i

Verschlüsselungsfunktion f



Aufblähen

Schlüssel eingehen lassen

Mischen

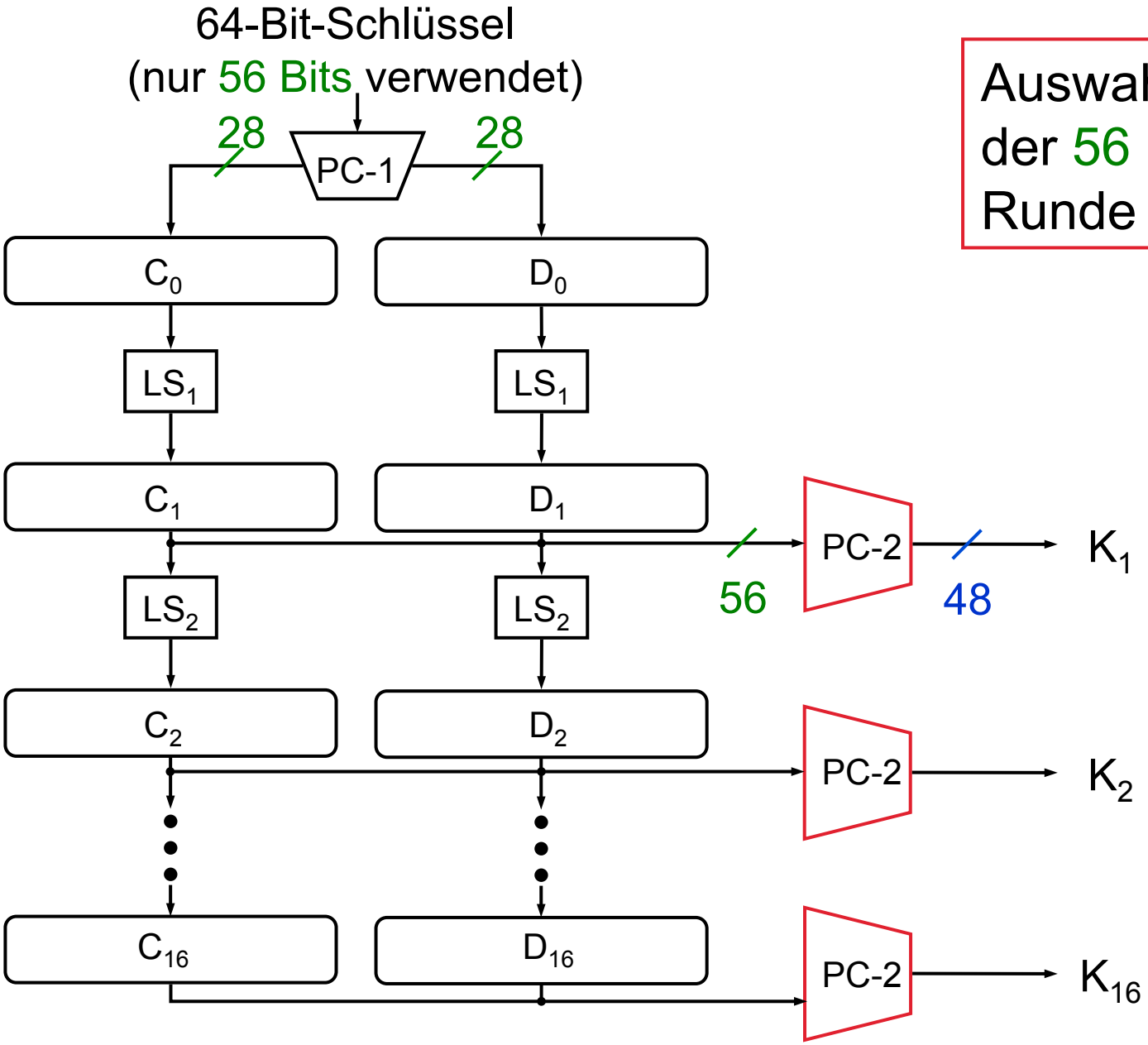
Nichtlinearität schaffen (Permutationen und ⊕ sind linear)

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

Begriffe

- Substitutions-Permutationsnetze
- Confusion - Diffusion

Teilschlüsselerzeugung

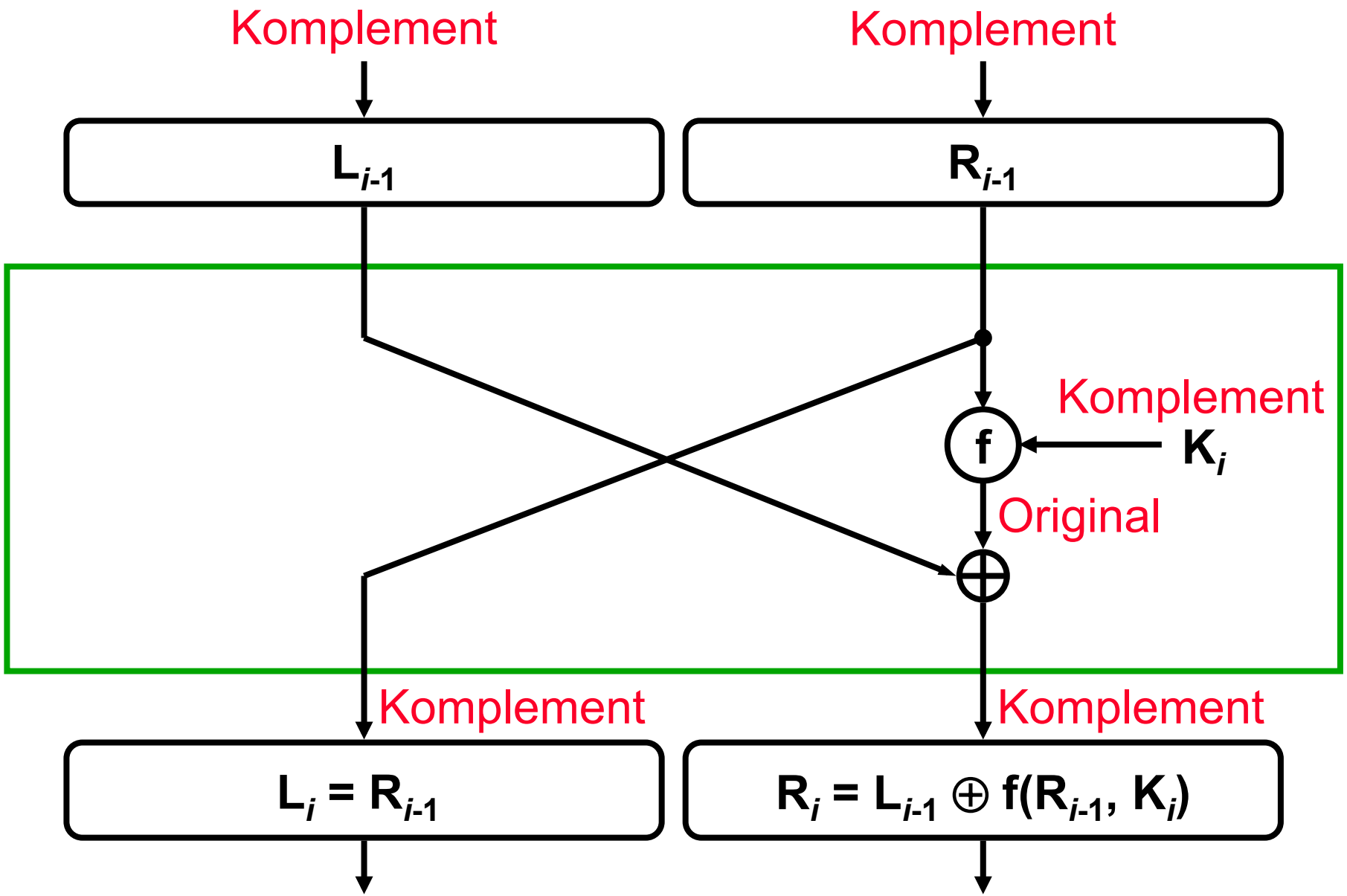


Auswahl von 48 der 56 Bits für jede Runde

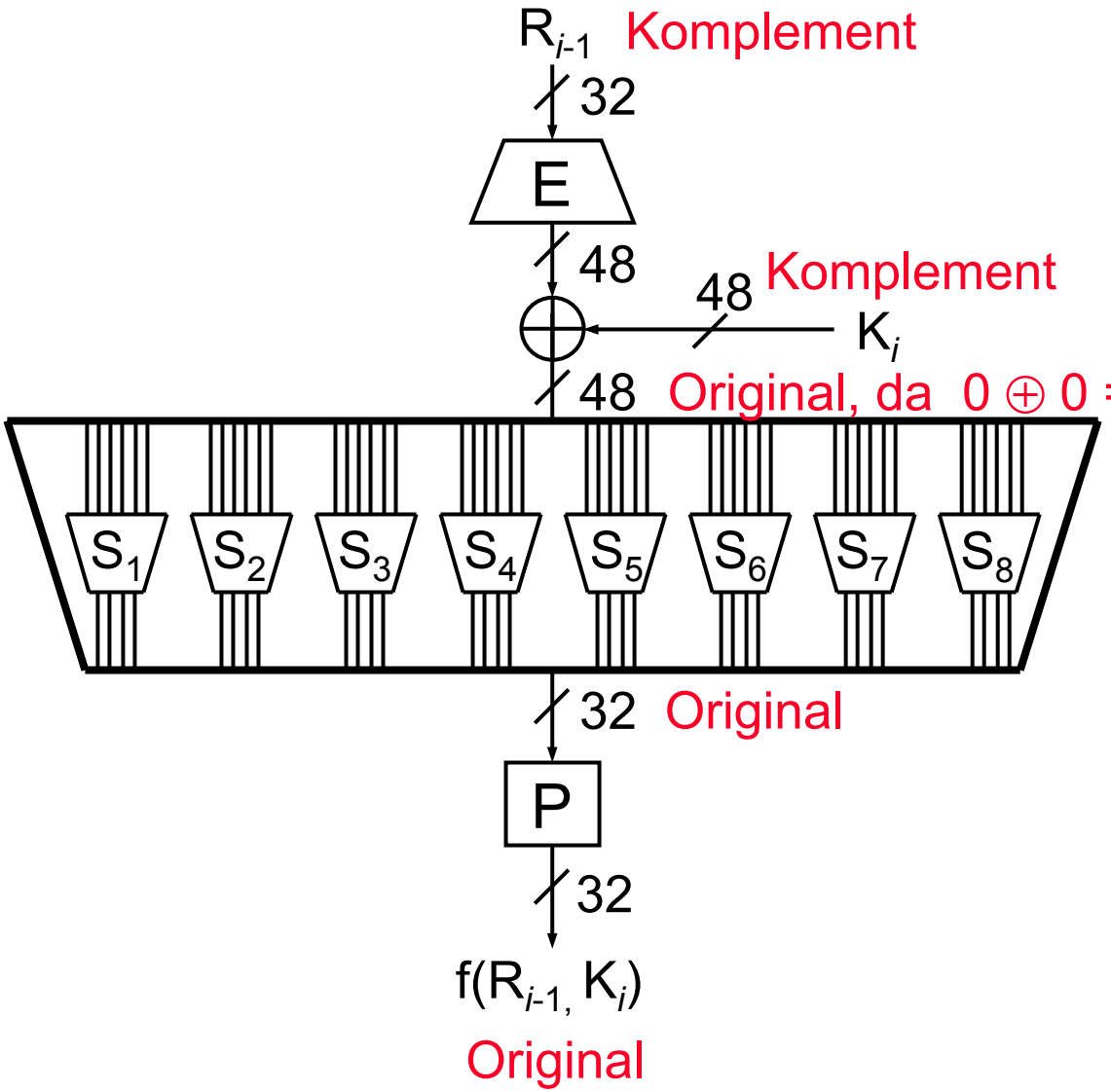
Komplementaritätseigenschaft von DES

$$\mathrm{DES}(\bar{k}, \bar{x}) = \overline{\mathrm{DES}(k, x)}$$

Eine Iterationsrunde



Verschlüsselungsfunktion f



Verallgemeinerung von DES

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

2.) variable Substitutionsboxen

3.) variable Permutationen

4.) variable Expansionspermutation

5.) variable Anzahl Iterationsrunden

Chiffren

Stromchiffre

synchron

selbstsynchronisierend

Blockchiffre

Betriebsarten:

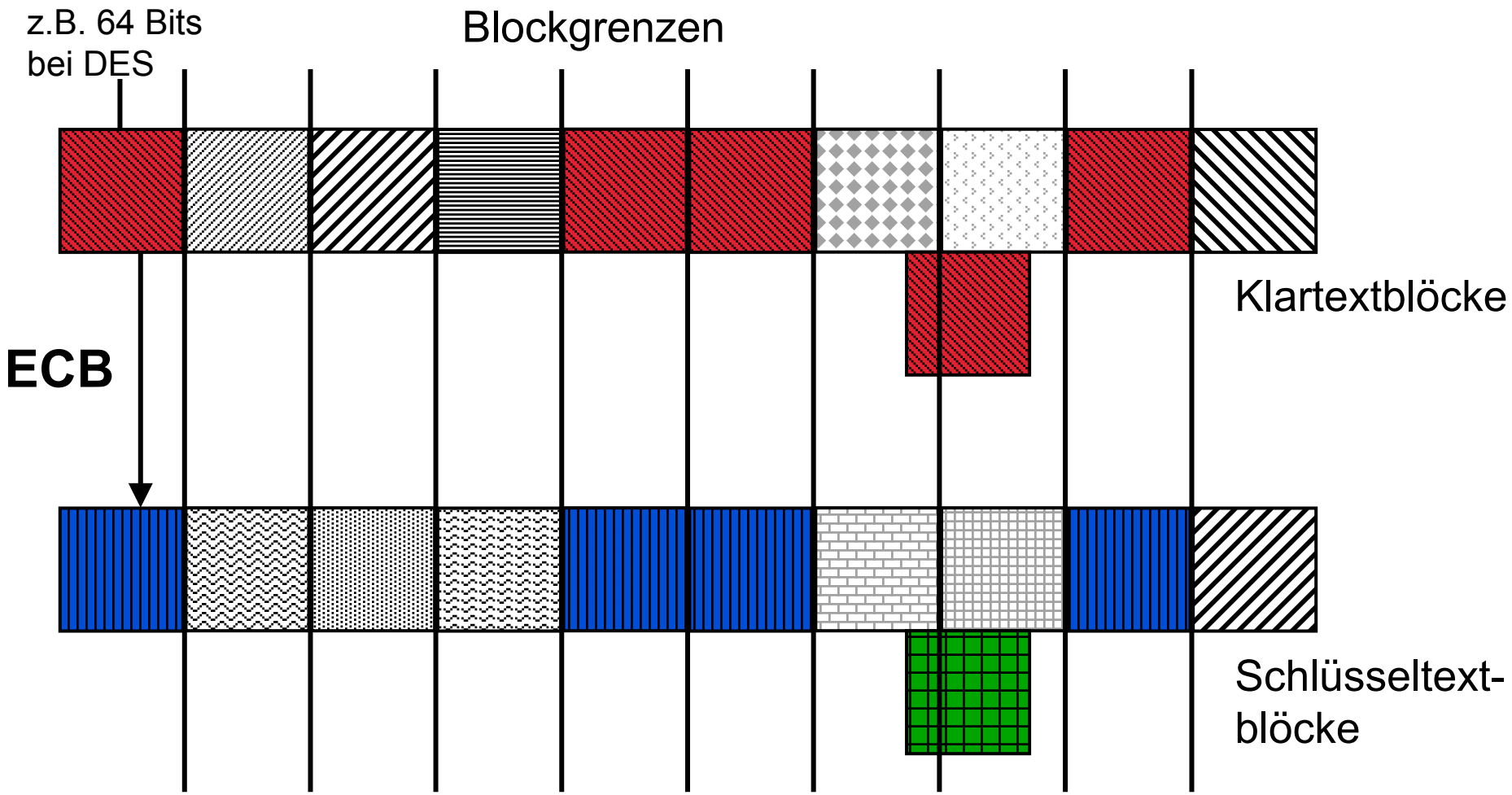
Einfachste: ECB (electronic codebook)

Jeder Block einzeln

Aber: Konzelation: Blockmuster erkennbar

Authentikation: Blöcke vertauschbar

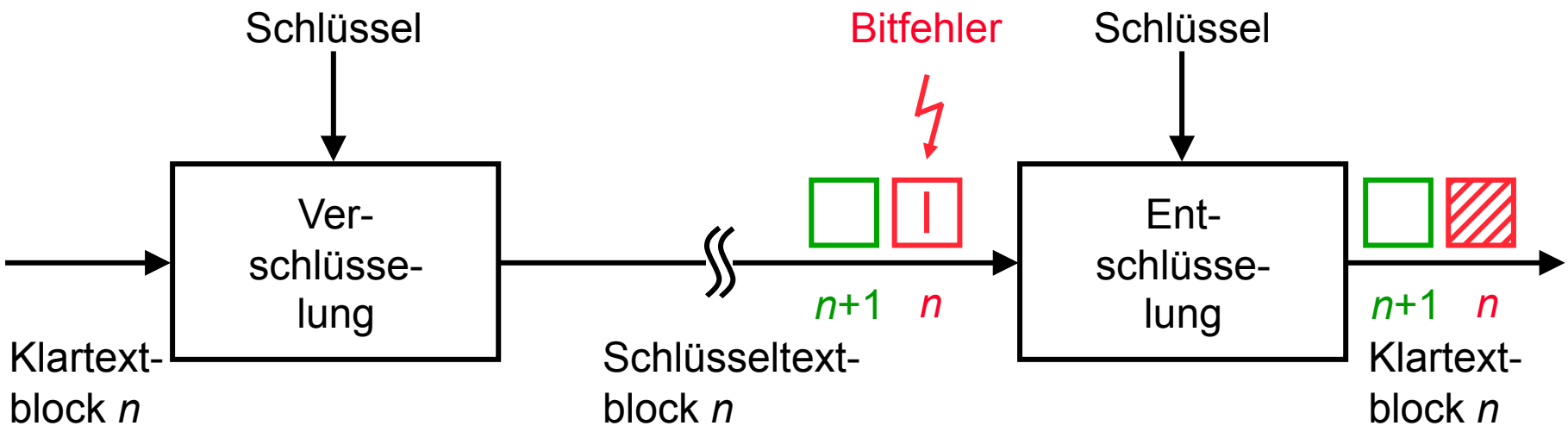
Hauptproblem von ECB



gleiche Klartextblöcke —ECB→ gleiche Schlüsseltextblöcke

Faxbeispiel (→ Kompression hilft)

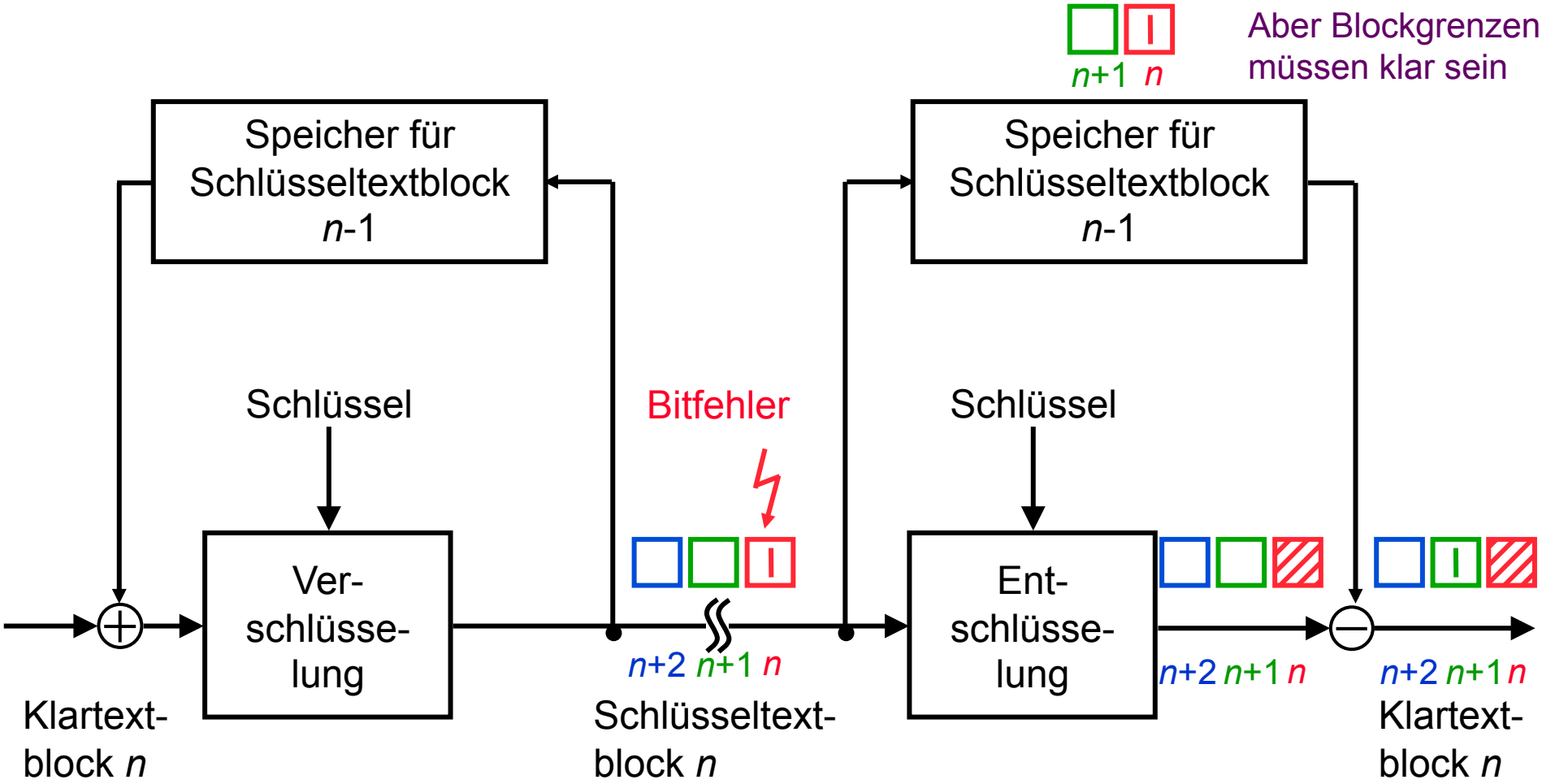
Elektronisches Codebuch (ECB)



Blockchiffre mit Blockverkettung (CBC)

Alle Linien führen der Blocklänge entsprechend viele Alphabetzeichen
⊕ Addition bezüglich passend gewähltem Modulus
⊖ Subtraktion bezüglich passend gewähltem Modulus

Bei Fehler auf Leitung: Resynchronisation nach 2 Blöcken
Aber Blockgrenzen müssen klar sein



selbstsynchronisierend

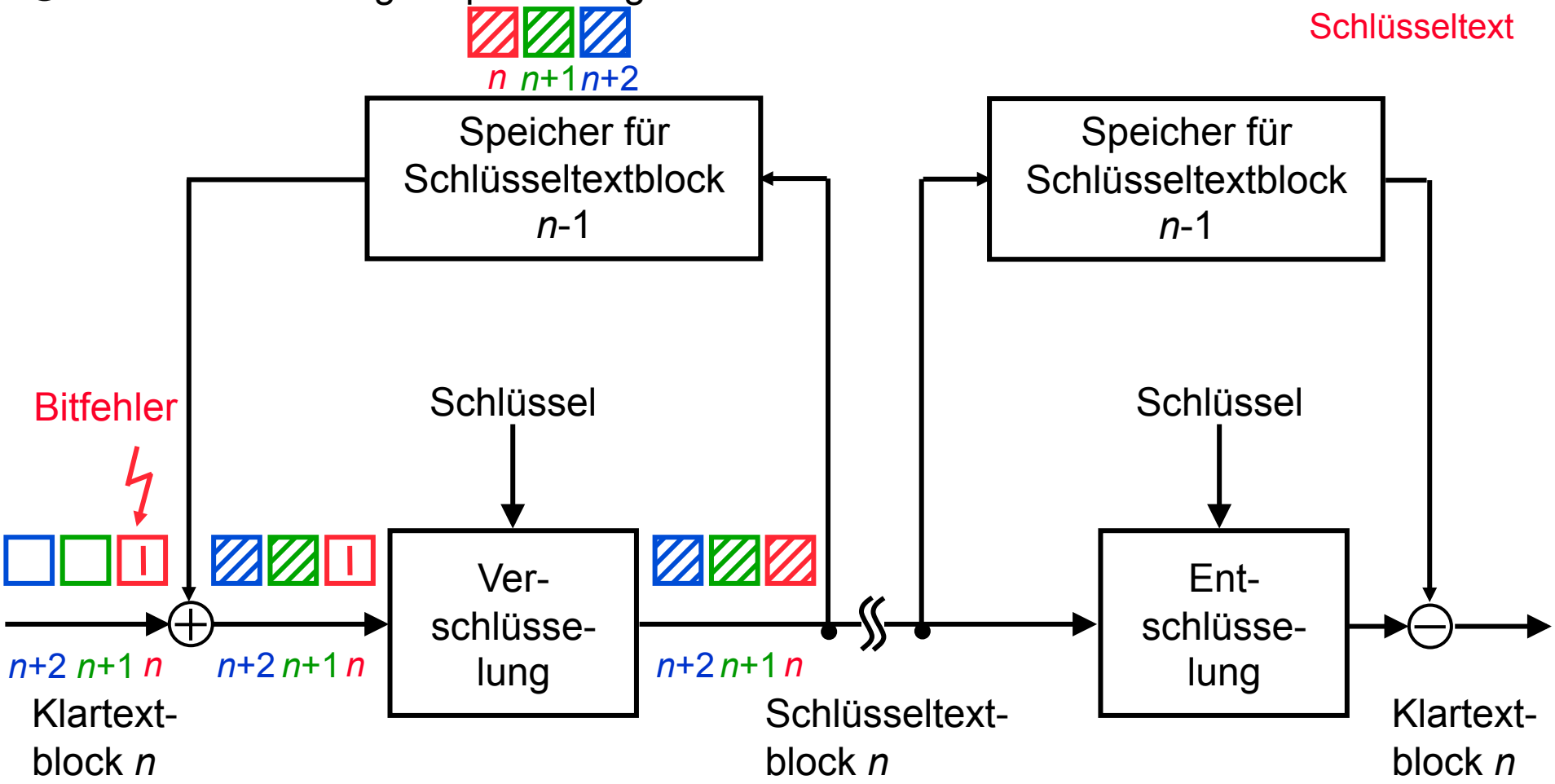
Blockchiffre mit Blockverkettung (CBC) (2)

Alle Linien führen der Blocklänge entsprechend viele Alphabetzeichen

⊕ Addition bezüglich passend gewähltem Modulus

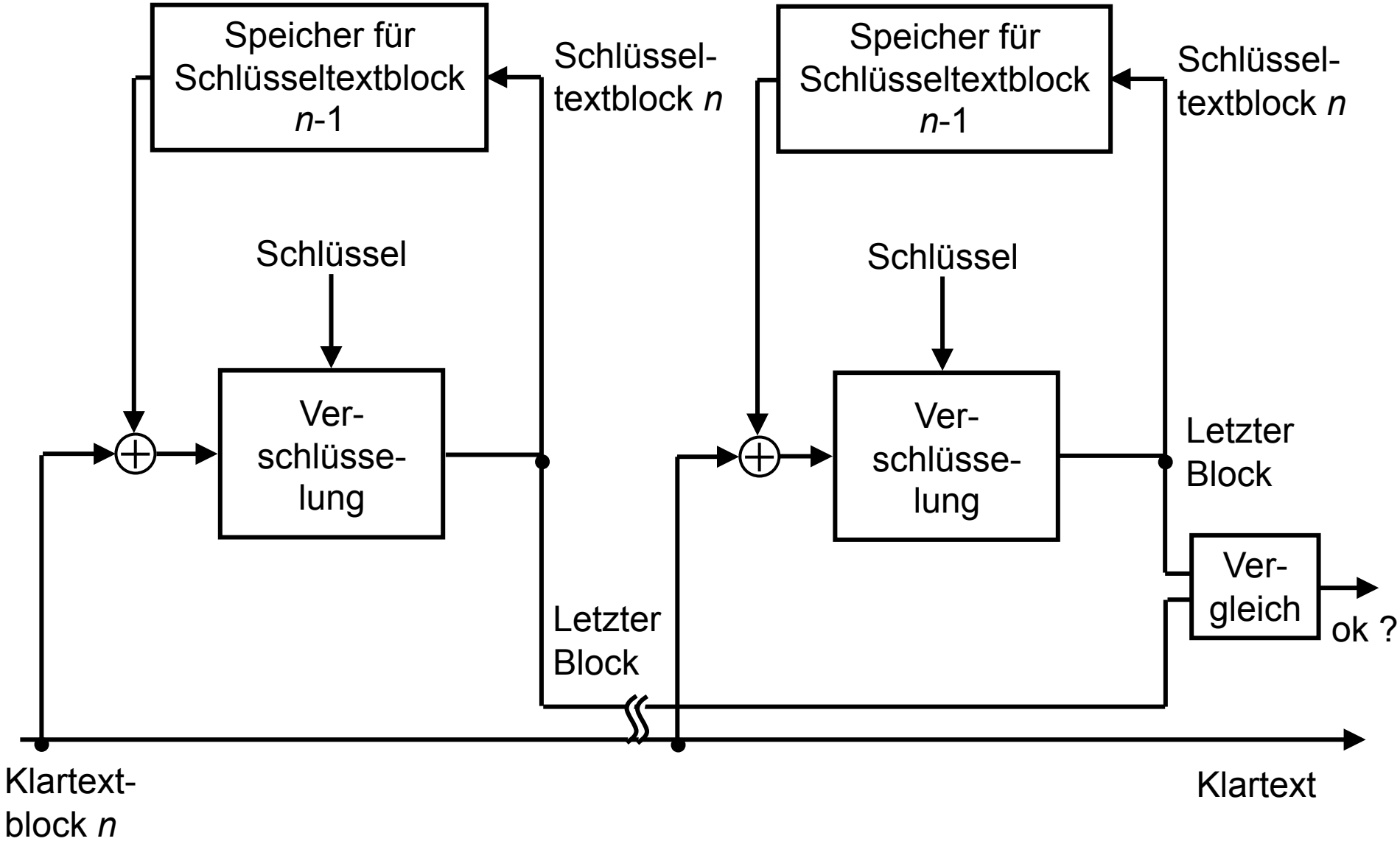
⊖ Subtraktion bezüglich passend gewähltem Modulus

1 geändertes Klartextbit ⇒ ab da anderer Schlüsseltext

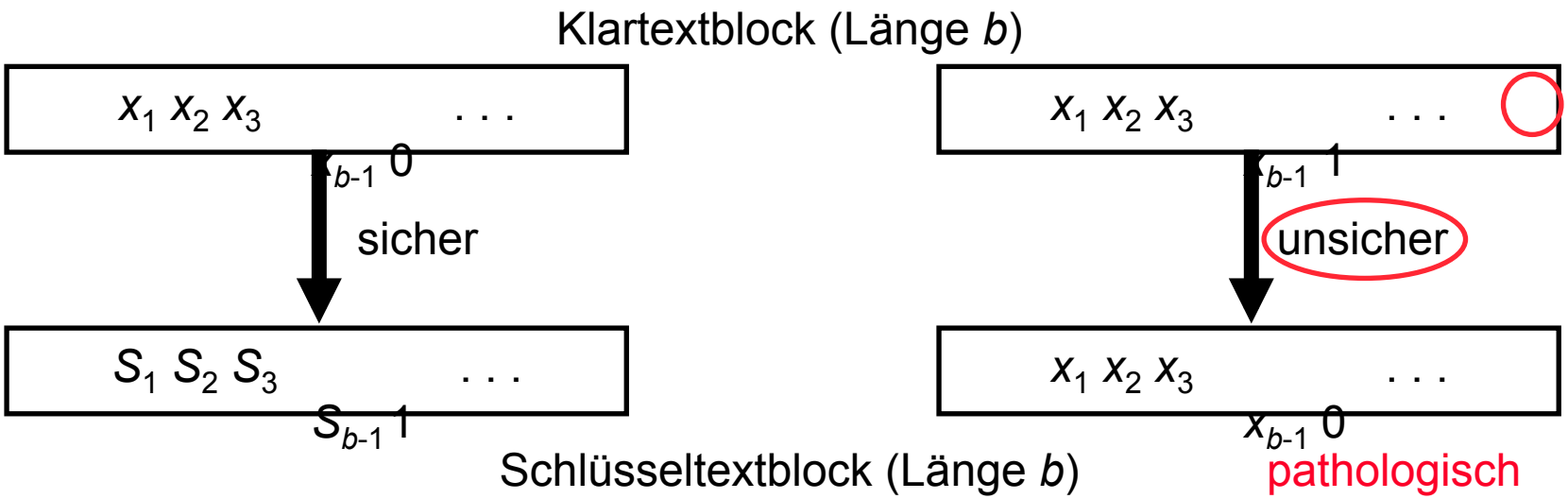


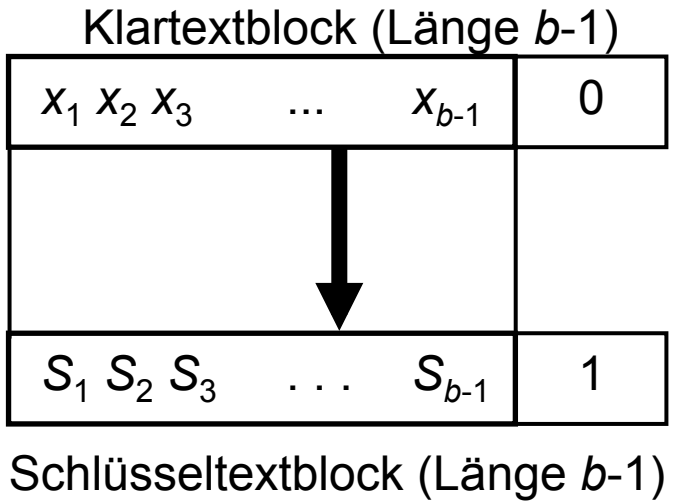
verwendbar zur Authentikation ⇒ letzten Block als MAC verwenden

CBC zur Authentikation



Pathologische Blockchiffre

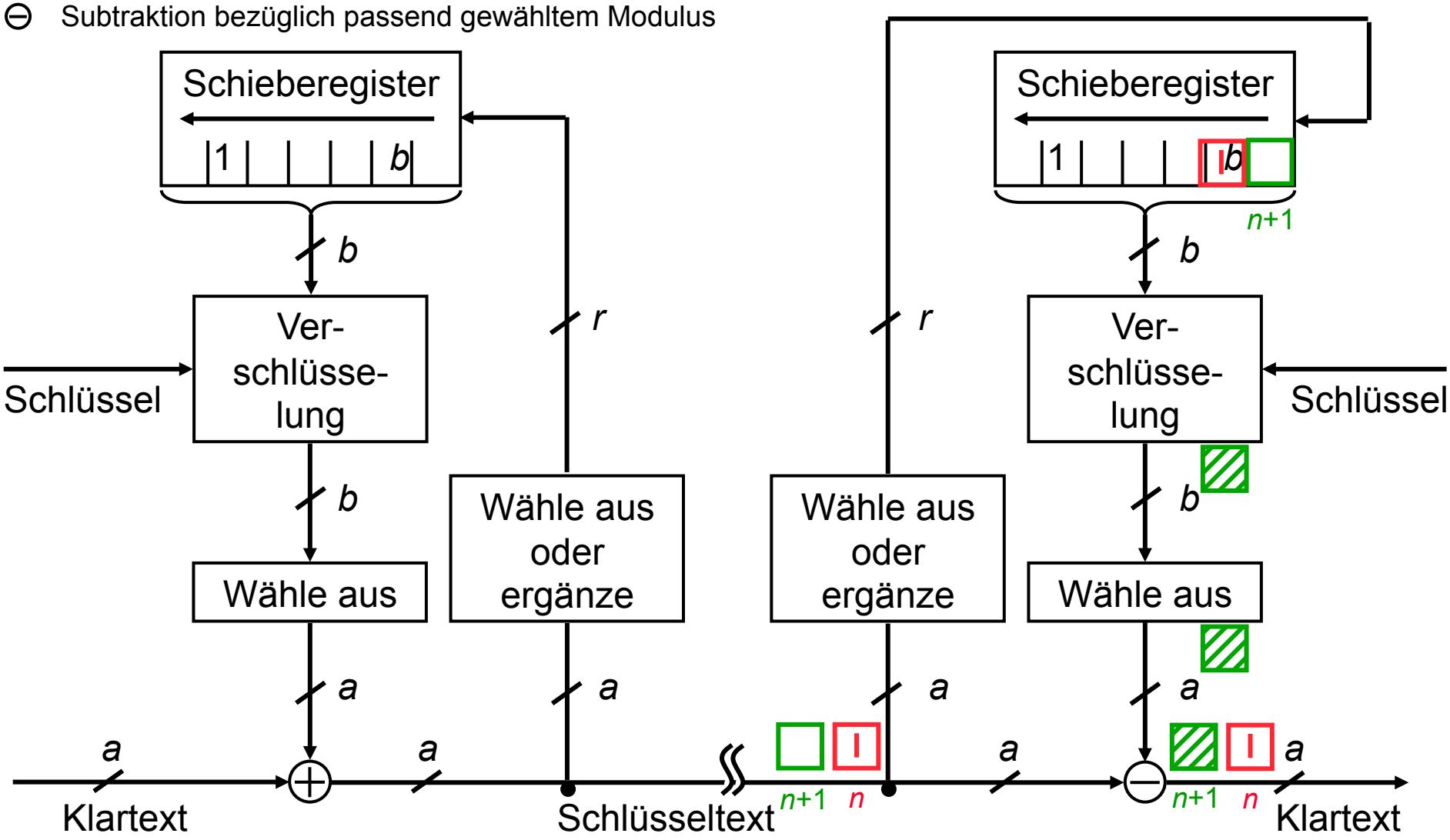




Schlüsseltextrückführung (CFB)

- b Blocklänge
- a Länge der Ausgabeeinheit, $a \leq b$
- r Länge der Rückkopplungseinheit, $r \leq b$
- $\oplus$ Addition bezüglich passend gewähltem Modulus
- $\ominus$ Subtraktion bezüglich passend gewähltem Modulus

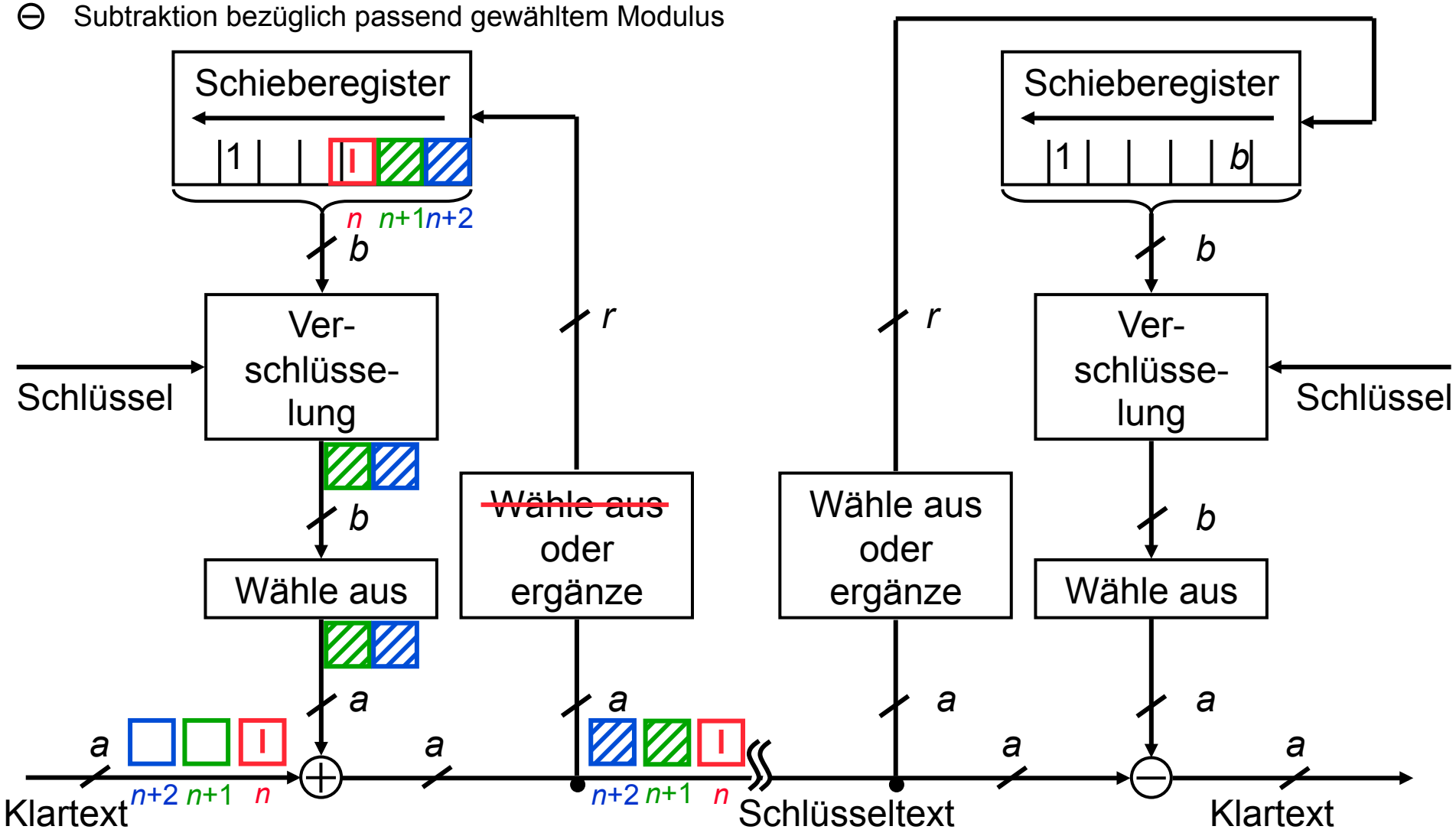
symmetrisch;
selbstsynchronisierend



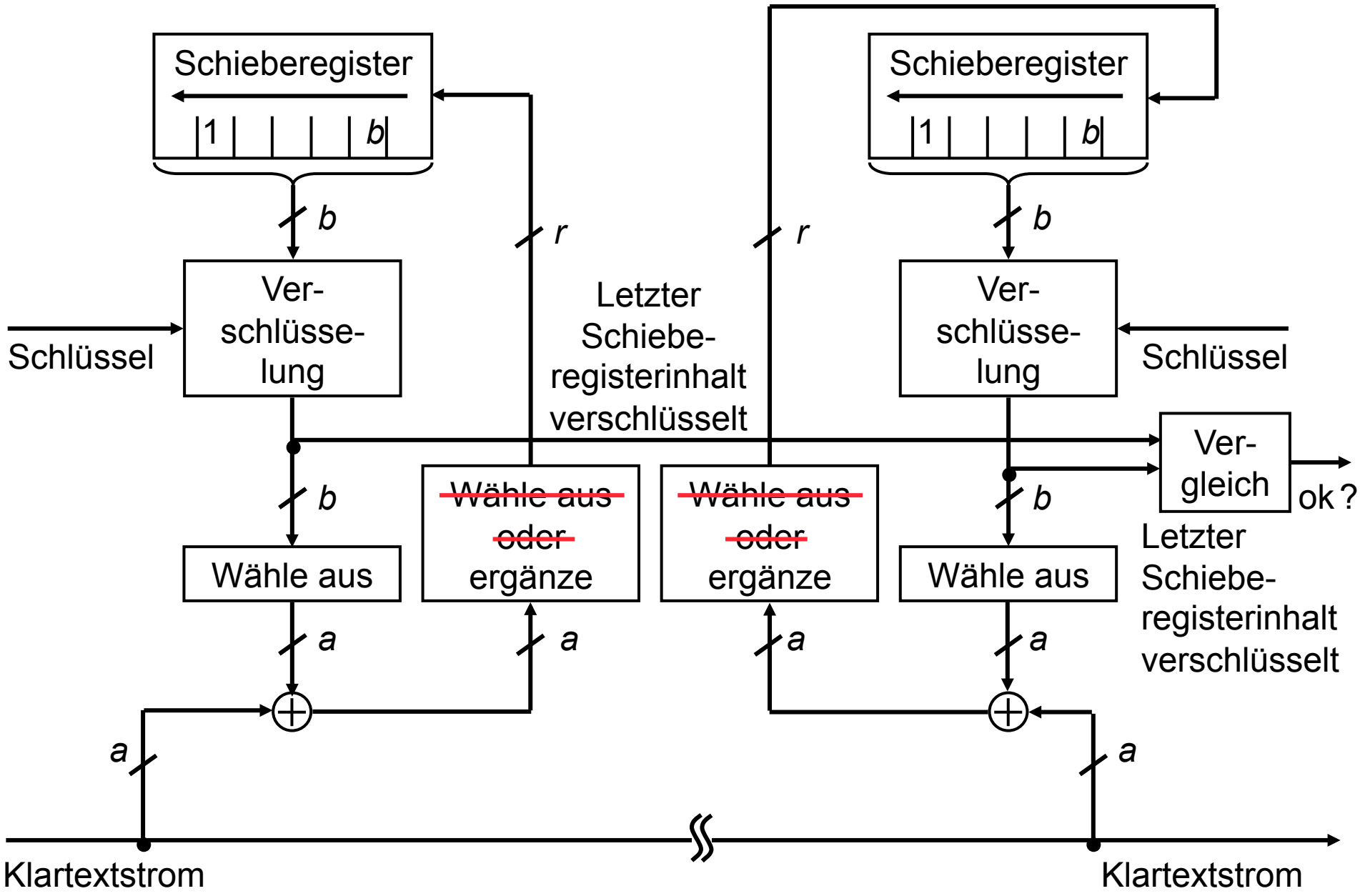
Schlüsseltextrückführung (CFB) (2)

b Blocklänge
a Länge der Ausgabeeinheit, $a \leq b$
r Länge der Rückkopplungseinheit, $r \leq b$
$\oplus$ Addition bezüglich passend gewähltem Modulus
$\ominus$ Subtraktion bezüglich passend gewähltem Modulus

symmetrisch;
selbst synchronisierend



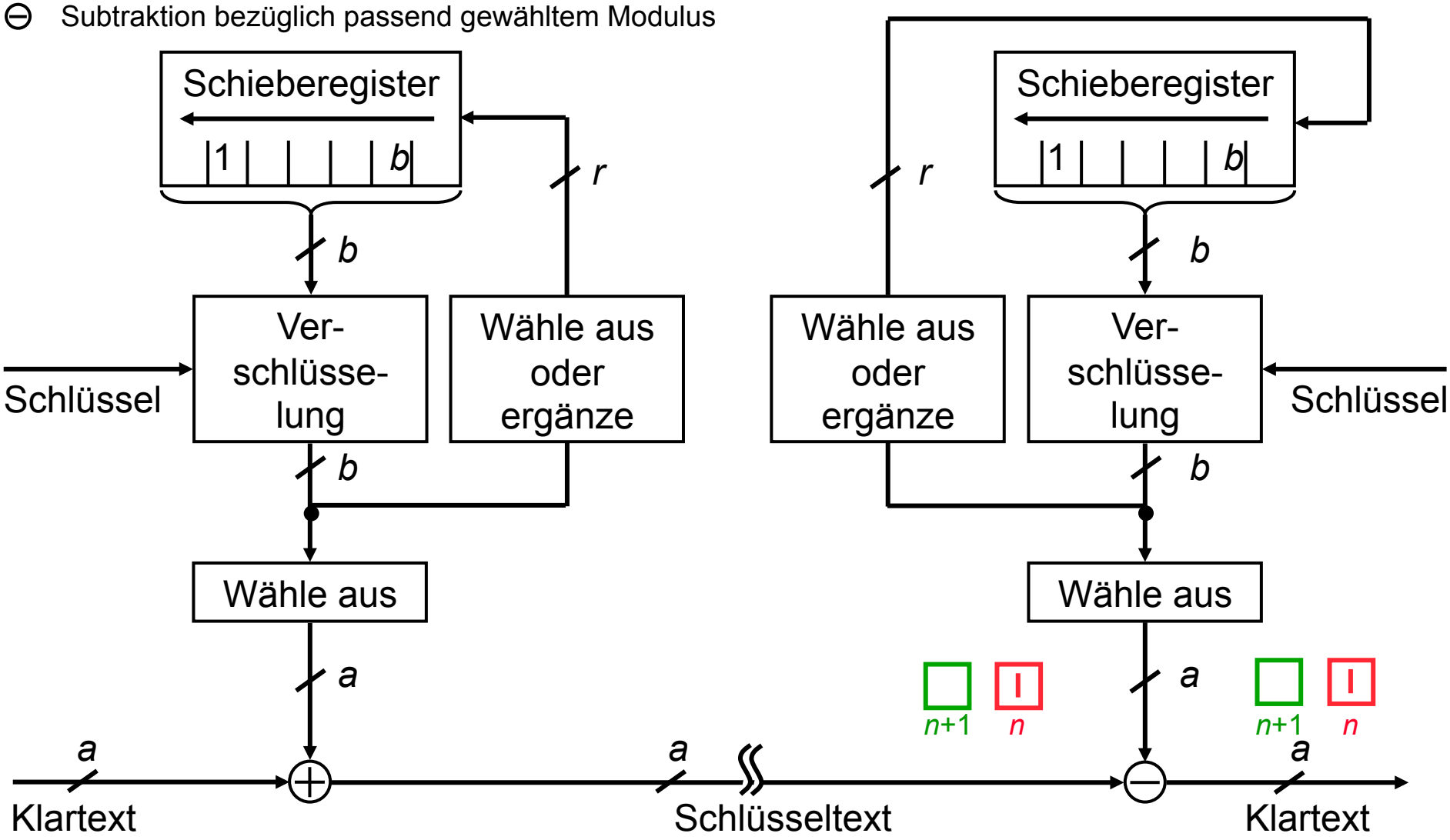
CFB Authentikation



Ergebnisrückführung (OFB)

- b Blocklänge
- a Länge der Ausgabeeinheit, $a \leq b$
- r Länge der Rückkopplungseinheit, $r \leq b$
- $\oplus$ Addition bezüglich passend gewähltem Modulus
- $\ominus$ Subtraktion bezüglich passend gewähltem Modulus

symmetrisch;
synchron
Pseudo-one-time-pad



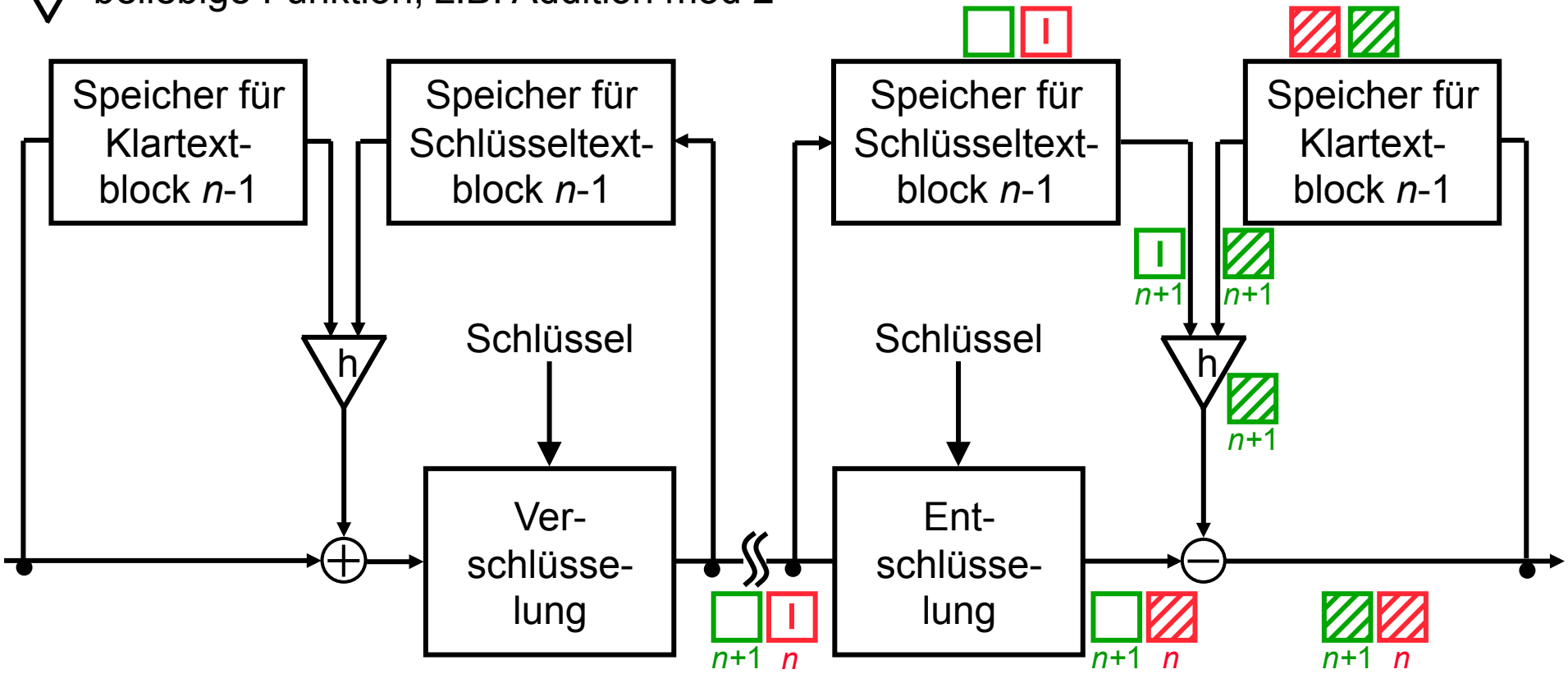
Blockverschlüsselung über Schlüssel- u. Klartext (PCBC)

Alle Linien führen der Blocklänge entsprechend viele Alphabetzeichen

⊕ Addition bezüglich passend gewähltem Modulus, z.B. 2

⊖ Subtraktion bezüglich passend gewähltem Modulus, z.B. 2

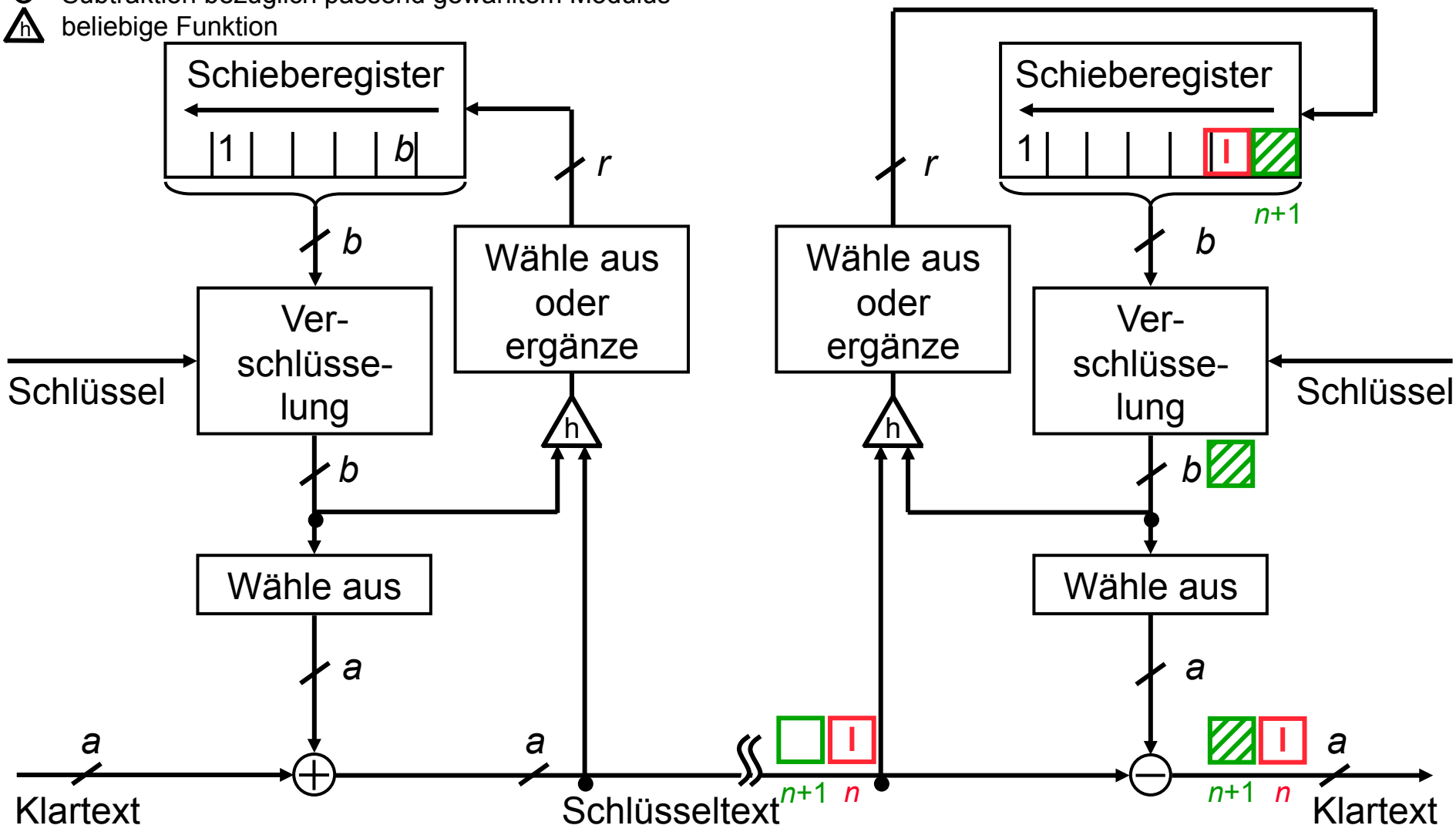
h beliebige Funktion, z.B. Addition mod $2^{\text{Blocklänge}}$



Schlüsseltext- u. Ergebnisrückführung (OCFB)

b Blocklänge
a Länge der Ausgabeeinheit, $a \le b$
r Länge der Rückkopplungseinheit, $r \le b$
⊕ Addition bezüglich passend gewähltem Modulus
⊖ Subtraktion bezüglich passend gewähltem Modulus
h beliebige Funktion

symmetrisch;
synchron



Eigenschaften der Betriebsarten

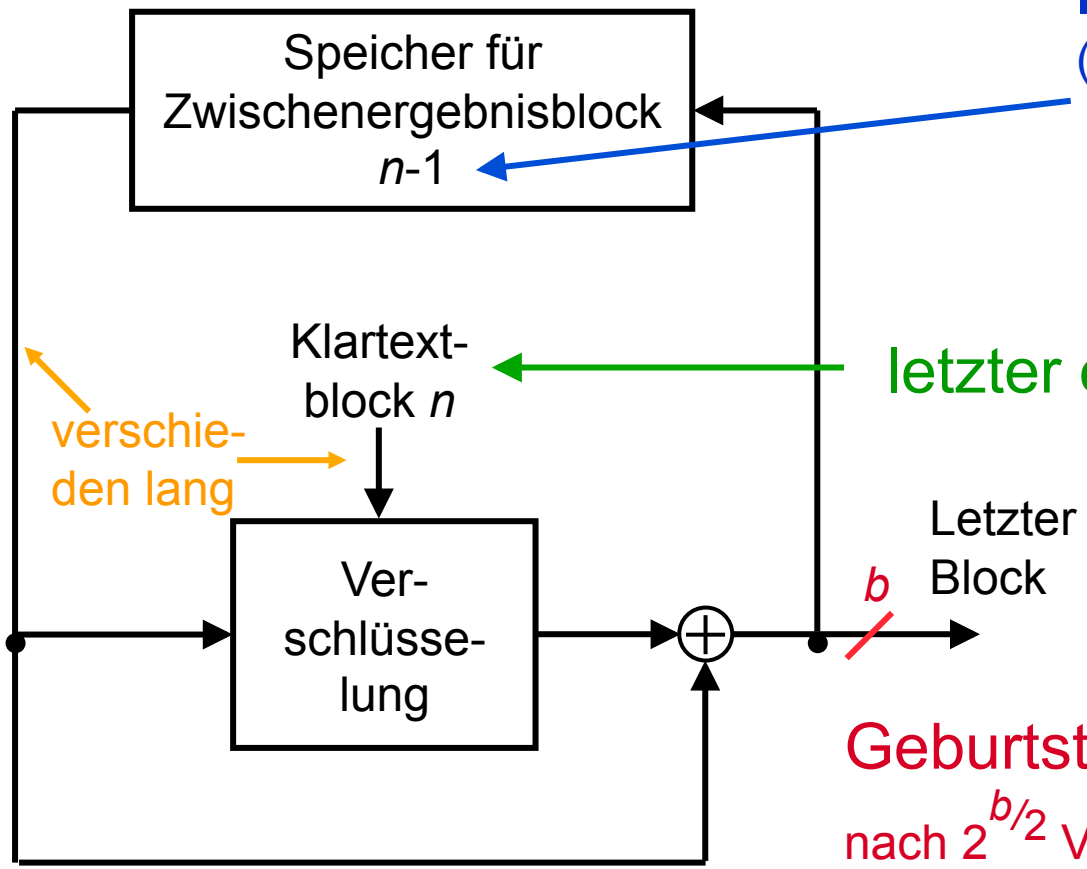
	ECB	CBC	PCBC	CFB	OFB	OCFB
Verwendung indeterministischer Blockchiffren	+ möglich			- nicht möglich		
Bei asymmetrischer Blockchiffre entsteht	+ asymmetrische Stromchiffre			- symmetrische Stromchiffre		
Länge der verschlüsselbaren Einheiten	- durch Blocklänge der Blockchiffre bestimmt			+ beliebig		
Fehlererweiterung	nur innerhalb eines Blockes	2 Blöcke	potentiell unbegrenzt	$1 + \lceil b/r \rceil$ Blöcke, wenn Fehler ganz rechts, sonst evtl. einer weniger	keine bei Verfälschung	potentiell unbegrenzt
auch zur Authentikation geeignet?	bei Redundanz innerhalb jedes Blockes: ja	bei deterministischer Blockchiffre: ja	ja, sogar Konzelation im selben Durchgang	bei deterministischer Blockchiffre: ja	bei geeigneter Redundanz: ja	ja, sogar Konzelation im selben Durchgang

Kollisionsresistente Hashfkt. aus determ. Blockchiffre

effizient !

beliebig nahezu

bewiesenermaßen nein, aber wohl untersucht



Initialisierungswert fest!
(sonst triviale Kollisionen: Zwischenergebnisblöcke und verkürzte Klartexte)

letzter enthält Länge in Bit

Geburtstagsparadox

nach $2^{b/2}$ Versuchen Kollision

Diffie-Hellman Schlüsselvereinbarung (1)

praktisch wichtig: Patent vor RSA abgelaufen → PGP ab Version 5

theoretisch wichtig: Steganographie mit öffentlichen Schlüsseln

beruht auf Schwierigkeit, **diskrete Logarithmen** zu ziehen

Sei ***p*** Primzahl, ***g*** ein Generator von Z_p^*

$$g^x = h \bmod p$$

x heißt ***diskreter Logarithmus*** von ***h*** zur Basis ***g*** modulo ***p***:

$$x = \log_g(h) \bmod p$$

Diskrete-Logarithmus-Annahme

Diskrete-Logarithmus-Annahme

∀ PPA $\mathcal{DL}$ (probabilistischer poly. Algorithmus, der diskrete Logarithmen zu ziehen versucht)

∀ Polynome Q

∃ L ∀ $\mathcal{l} \geq L$: (asymptotisch gilt:)

Wenn p zufällige Primzahl der Länge $\mathcal{l}$

danach g zufällig innerhalb der Generatoren von Z_p^*

x zufällig in Z_p^*

gewählt werden und $g^x = h \mod p$

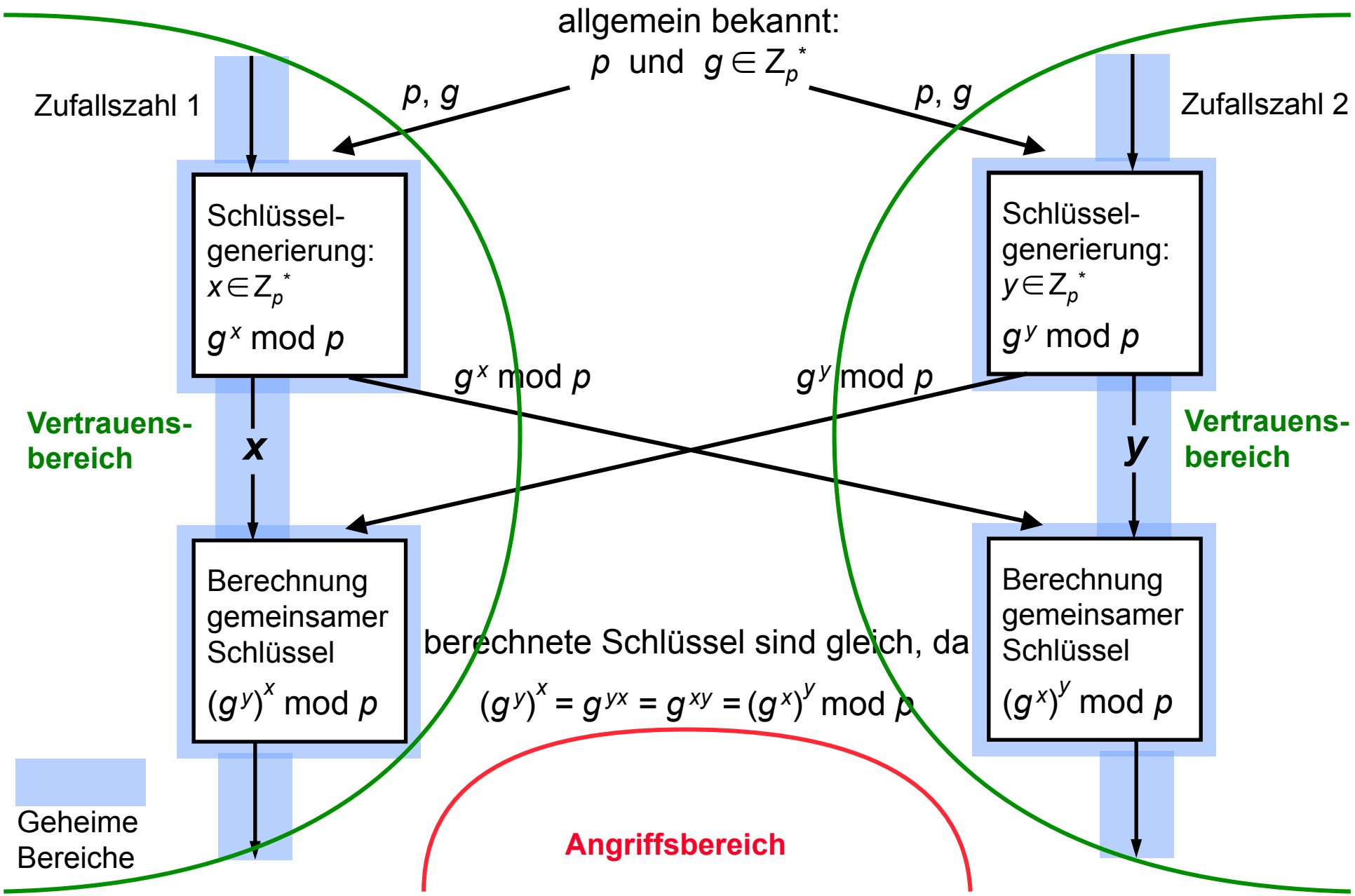
$$\mathcal{W}(\mathcal{DL}(p,g,h)=x) \leq \frac{1}{Q(\mathcal{l})}$$

(Wahrscheinlichkeit, dass $\mathcal{DL}$ wirklich den diskreten Logarithmus zieht, sinkt schneller als $\frac{1}{\text{jedes Polynom}}$)

Vertrauenswürdig ??

Praktisch genauso gut untersucht wie Faktorisierungsannahme

Diffie-Hellman Schlüsselvereinbarung (2)



Diffie-Hellman-Annahme

DH-Annahme:

Gegeben p, g, $g^x \bmod p$ und $g^y \bmod p$

Berechnen von $g^{xy} \bmod p$ ist schwierig

DH-Annahme ist stärker als Diskrete-Logarithmus-Annahme

- Diskrete Logs ziehen $\Rightarrow$ DH gebrochen
  Bestimme aus p, g, $g^x \bmod p$ und $g^y \bmod p$ entweder x oder y. Berechne $g^{xy} \bmod p$ wie einer der Partner der DH-Schlüsselvereinbarung.
- Bisher konnte nicht gezeigt werden:
  Aus p, g, $g^x \bmod p$, $g^y \bmod p$ und $g^{xy} \bmod p$ kann x oder y bestimmt werden.

Finden Generator in zyklischer Gruppe Z_p^*

Finden eines Generators einer zyklischen Gruppe Z_p^*

Faktorisiere $p-1 =: p_1^{e_1} \bullet p_2^{e_2} \bullet \ldots \bullet p_k^{e_k}$

1. Wähle zufälliges Element g in Z_p^*

2. Für i von 1 bis k:

   $$b := g^{\frac{p-1}{p_i}} \bmod p$$

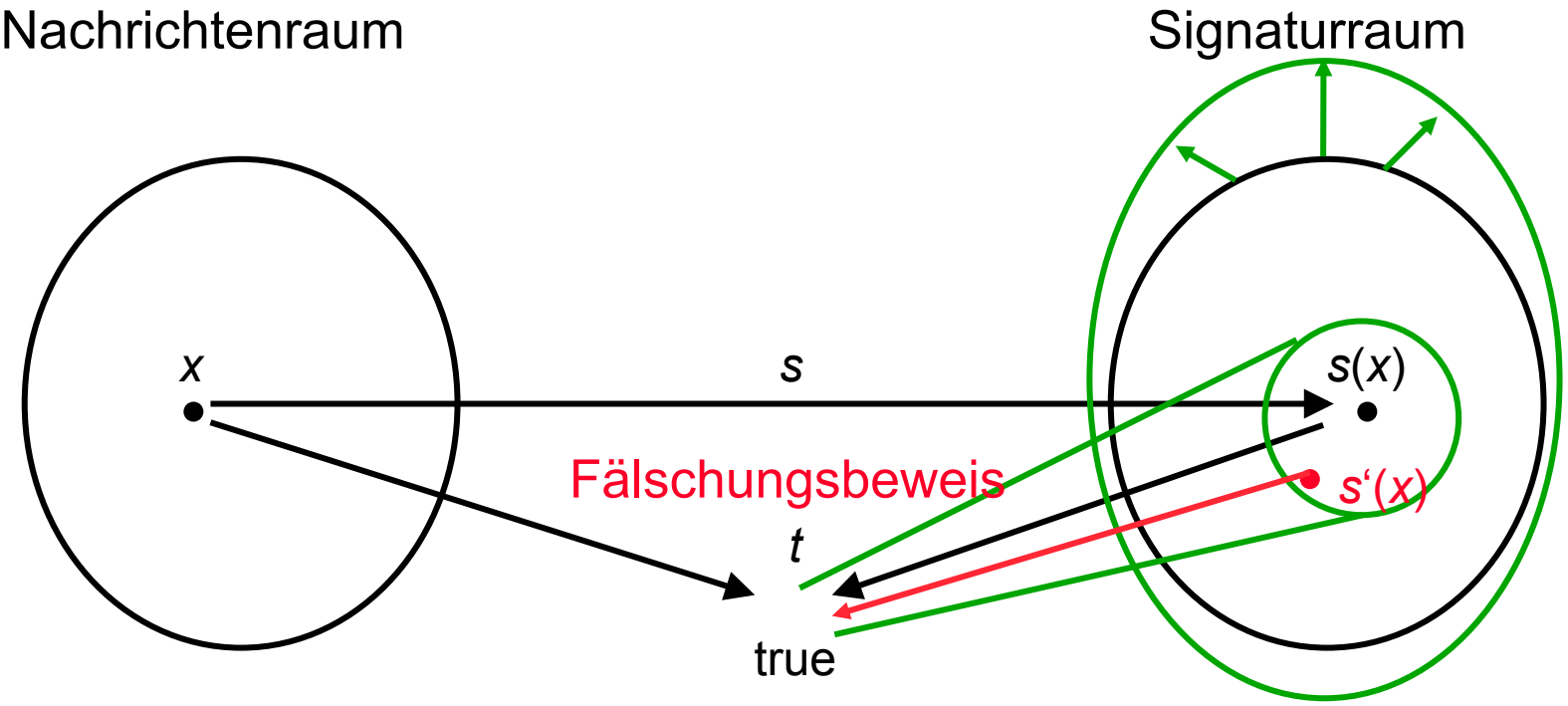
   Wenn $b=1$ gehe zu 1.

Digitales Signatursystem

Sicherheit auch „asymmetrisch“

üblich: unbedingt sicher für Empfänger
nur kryptographisch für Signierer

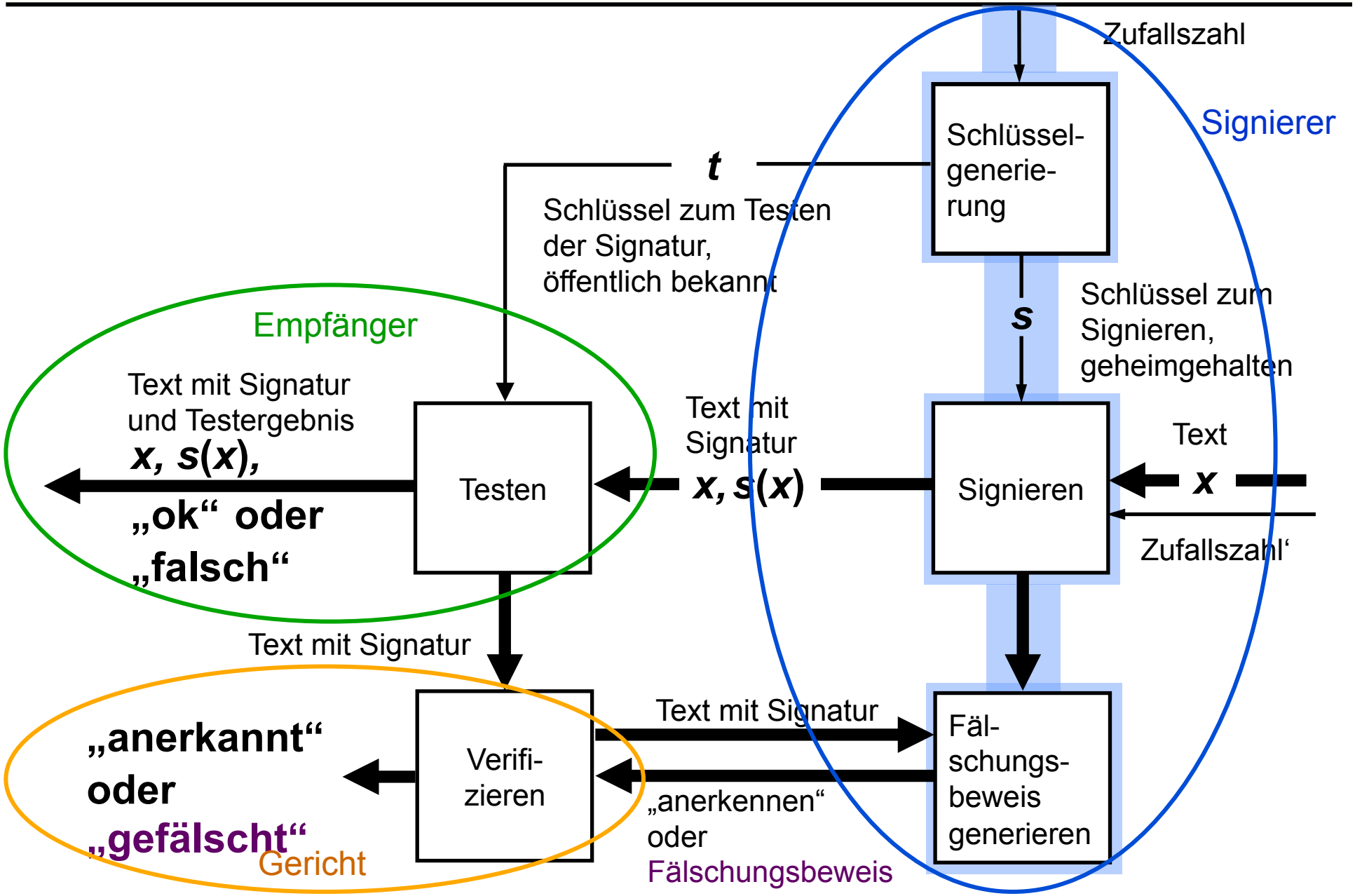
neu: Brechen unbedingt sicher für Signierer
beweisbar nur kryptographisch für Empfänger



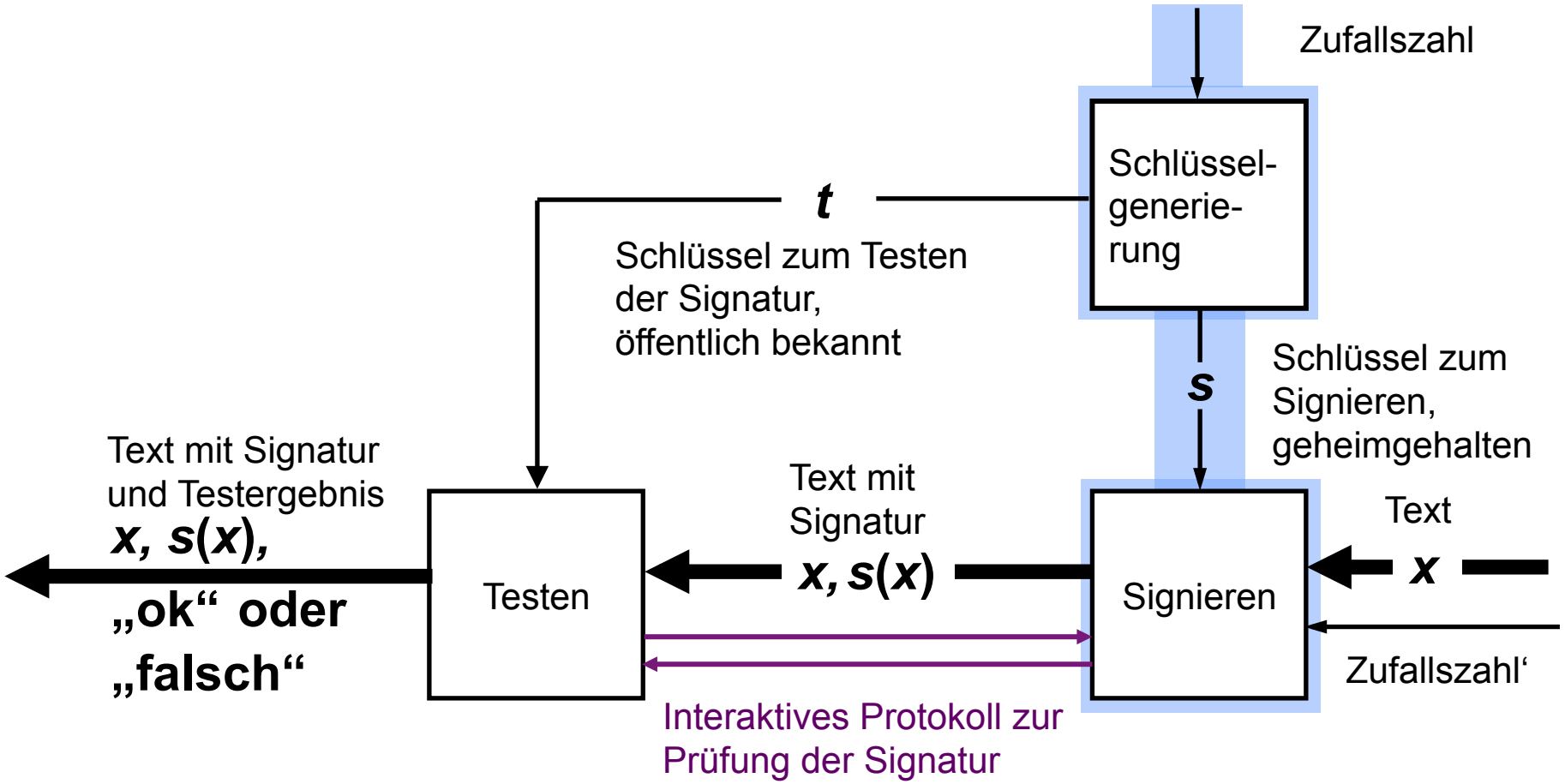
Risikoverteilung bei gefälschter Signatur:

1. Empfänger
2. Versicherung oder Systembetreiber
3. Signierer

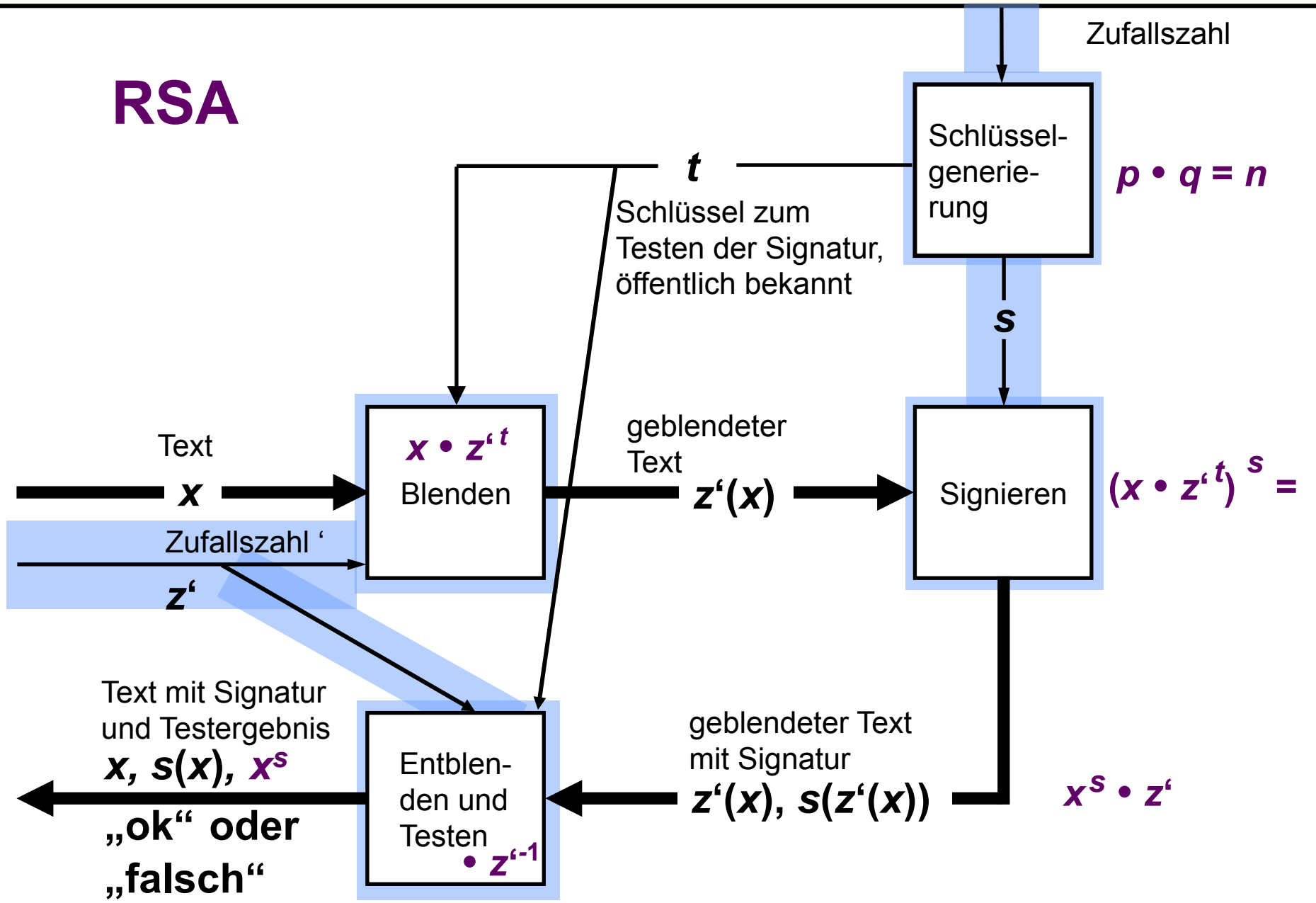
Fail-Stop-Signatursystem



Nicht herumzeigbare Signaturen (undeniable signatures)



Signatursystem zum blinden Leisten von Signaturen



Schwellwertschema (1)

Schwellwertschema:

Geheimnis $\mathcal{G}$
n Teile
k Teile: effiziente Rekonstruktion von $\mathcal{G}$
k-1 Teile: keine Information über $\mathcal{G}$

Realisierung: Polynominterpolation (Shamir, 1979)

Zerlegung des Geheimnisses:

Geheimnis $\mathcal{G}$ sei Element von Z_p, p sei Primzahl

Polynom $q(x)$ des Grades k-1:

$a_1, a_2, \ldots, a_{k-1}$ zufällig, gleichmäßig und unabhängig in Z_p

$q(x) := \mathcal{G} + a_1 x + a_2 x^2 + \ldots + a_{k-1} x^{k-1}$

n Teile $(i, q(i))$ mit $1 \leq i \leq n$, wobei $n < p$ gelten muss.

Schwellwertschema (2)

Rekonstruktion des Geheimnisses:

k Teile $(x_j, q(x_j))$ $(j = 1 \ldots k)$:

$$q(x) = \sum_{j=1}^{k} q(x_j) \prod_{m=1,\, m \neq j}^{k} \frac{(x - x_m)}{(x_j - x_m)} \mod p$$

Das Geheimnis $\mathcal{G}$ erhält man dann als $q(0)$.

Beweisskizze:

1. k-1 Teile $(j, q(j))$ liefern keine Information über $\mathcal{G}$, da es für jeden Wert von $\mathcal{G}$ immer noch genau ein Polynom vom Grad k-1 gibt.
2. richtigen Grad k-1; liefert für jedes Argument x_j den Wert $q(x_j)$ (denn Produkt liefert bei bei Einsetzen von x_j für x den Wert 1 und bei Einsetzen aller anderen x_i für x den Wert 0).

Schwellwertschema (3)

Polynominterpolation ist Homomorphismus bzgl. +

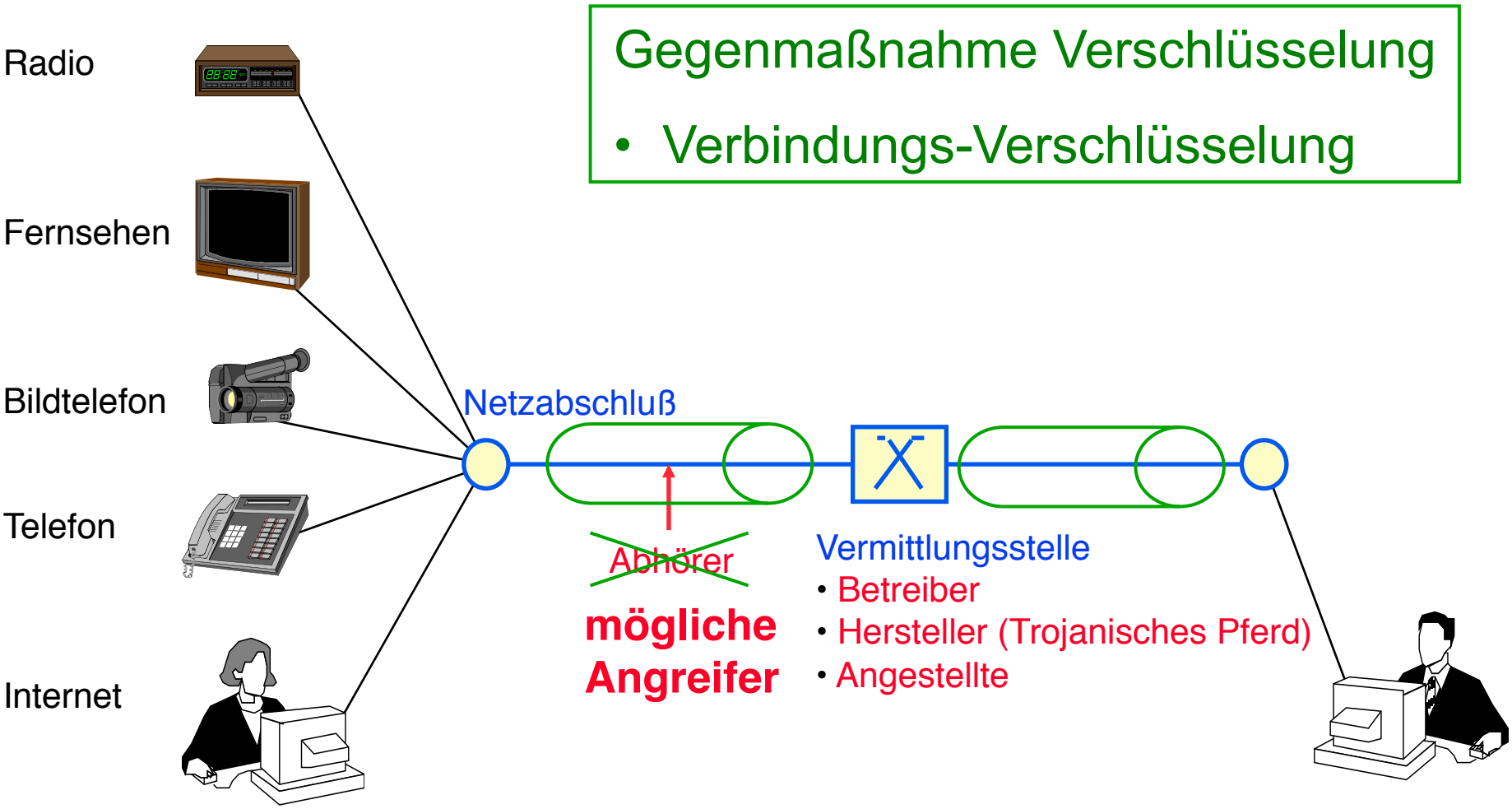
Addition der Teile ⇒ Addition der Geheimnisse

Share refreshing

1.) Wähle zufälliges Polynom q' für $\mathcal{G}' = 0$
2.) Verteile die n Teile $(i, q'(i))$
3.) Jeder addiert dies zu seinem Teil hinzu
→ „neues“ zufälliges Polynom mit „altem“ Geheimnis $\mathcal{G}$

- Wiederhole dies, so dass jeder mal das zufällige Polynom wählt
- Benutze *verifiable secret sharing*, damit jeder überprüfen kann, dass Polynome richtig gebildet.

Beobachtbarkeit von Benutzern in Vermittlungsnetzen



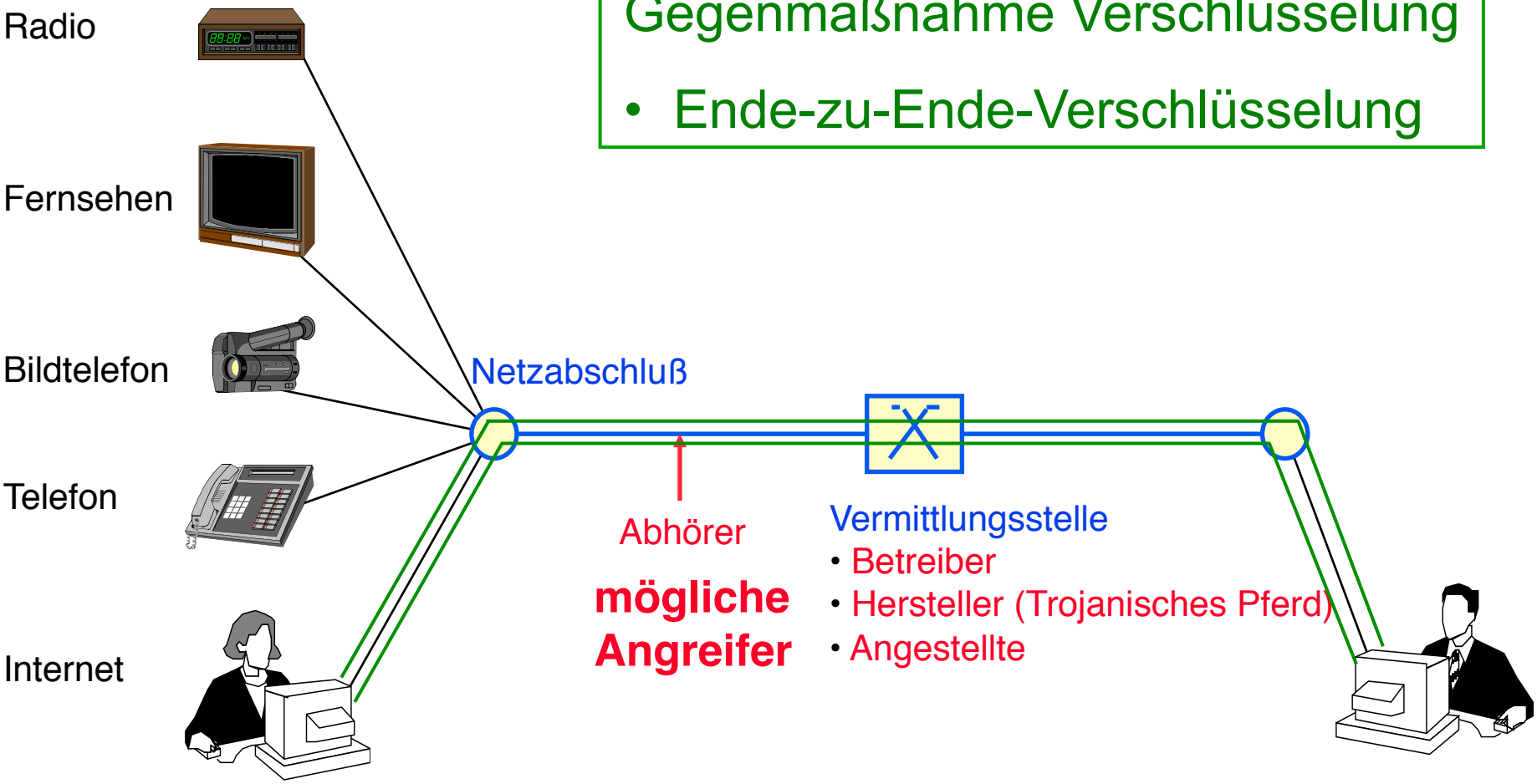
Gegenmaßnahme Verschlüsselung

- Verbindungs-Verschlüsselung

Beobachtbarkeit von Benutzern in Vermittlungsnetzen

Gegenmaßnahme Verschlüsselung

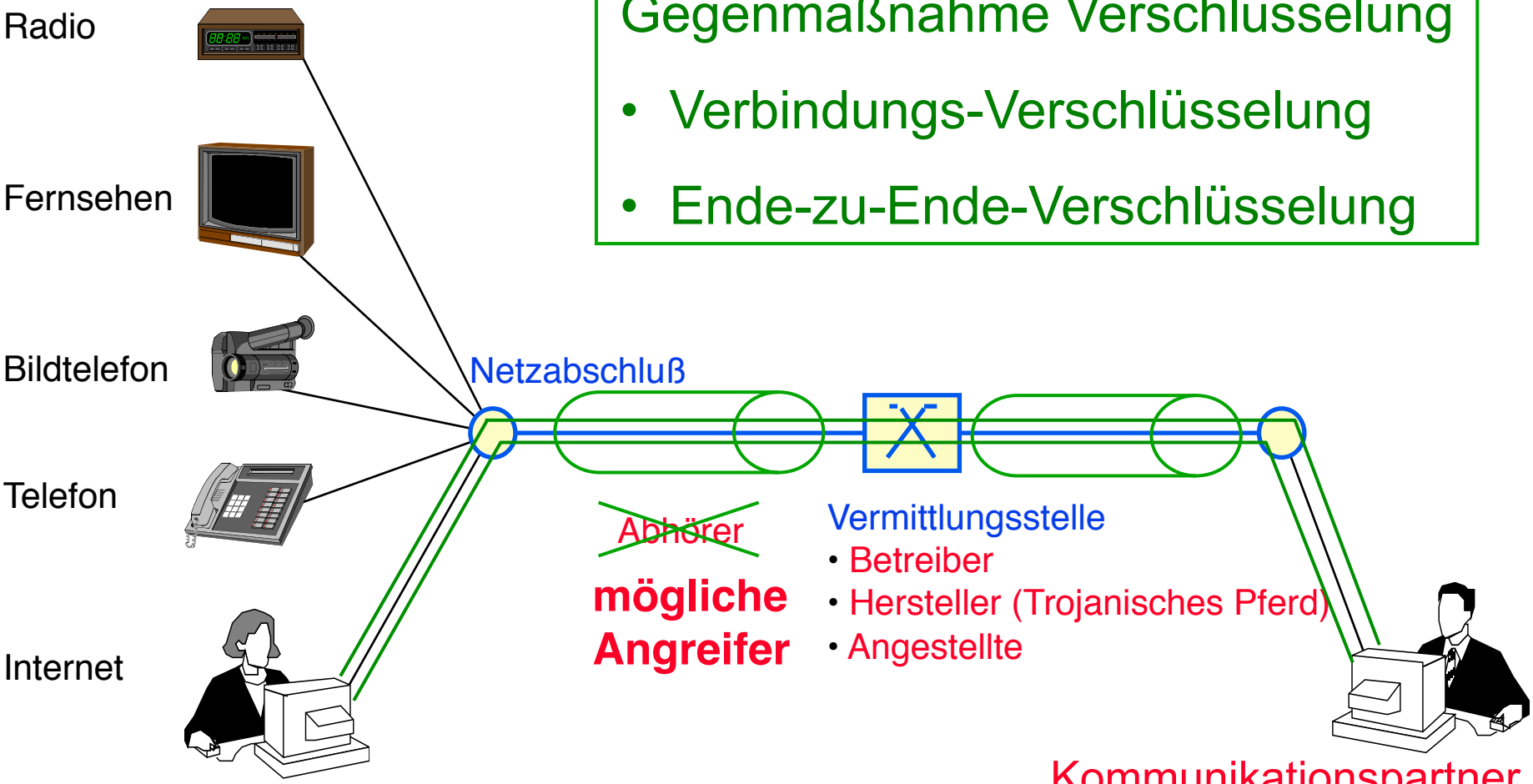
- Ende-zu-Ende-Verschlüsselung



Beobachtbarkeit von Benutzern in Vermittlungsnetzen

Gegenmaßnahme Verschlüsselung

- Verbindungs-Verschlüsselung
- Ende-zu-Ende-Verschlüsselung

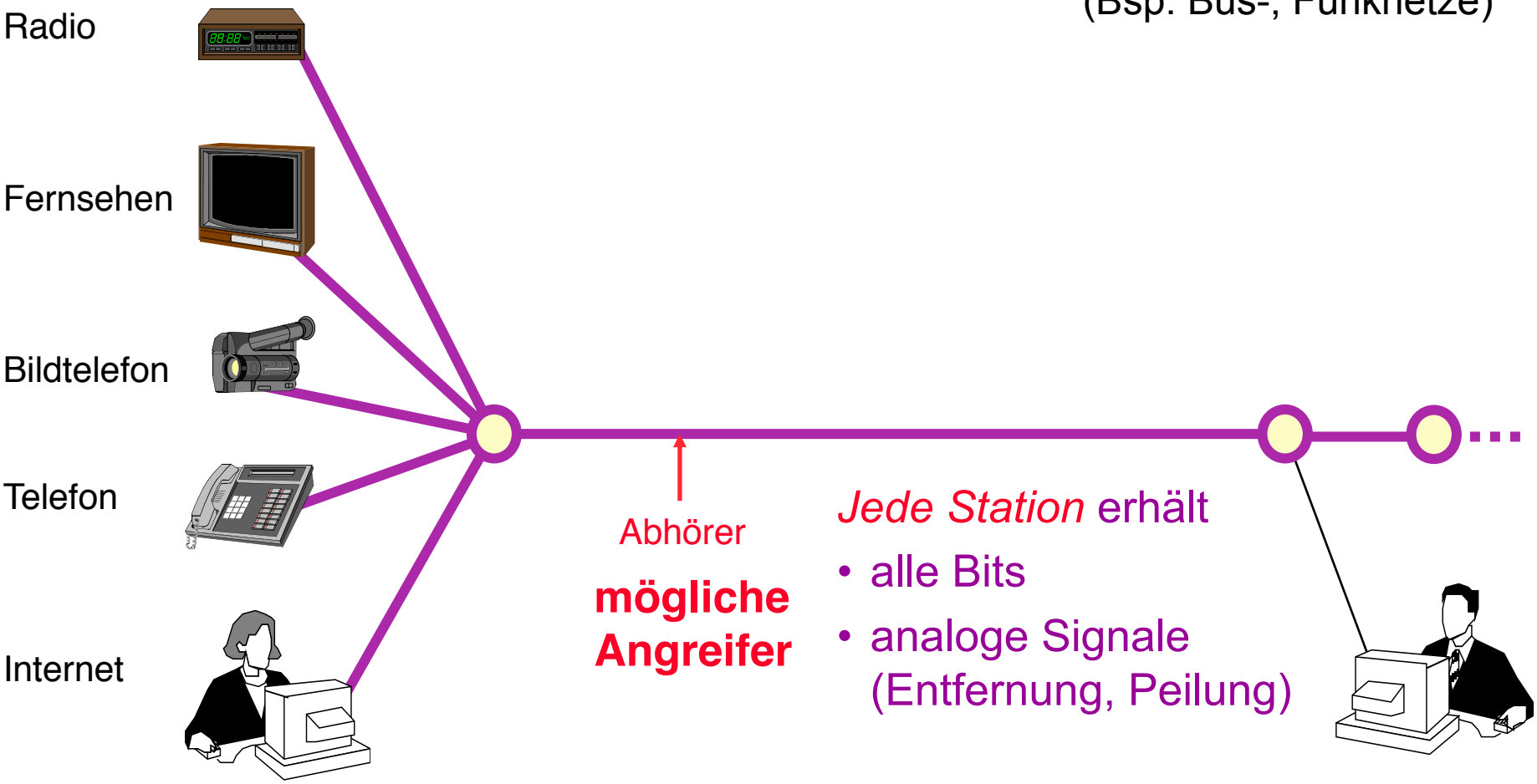


Problem: Verkehrsdaten
wer mit wem?
wann? wie lange?
wieviel Information?

Ziel: Verkehrsdaten (und damit auch Interessensdaten) dadurch „schützen“, dass sie nicht erfasst werden können.

Beobachtbarkeit von Benutzern in Broadcastnetzen

(Bsp. Bus-, Funknetze)



Realität oder Science Fiction?

Seit etwa 1990 Realität

Video-8 Kassette

5 G-Byte

= 3 * Volkszählung 1987

Speicherkosten < 25 EUR

100 Video-8 (oder in 2003: 2 Festplatten mit je 250 G-Byte für je < 280 EUR) speichern

alle Fernsprechverbindungen eines Jahres:

Wer mit wem ?

Wann ?

Wie lange ?

Von wo ?

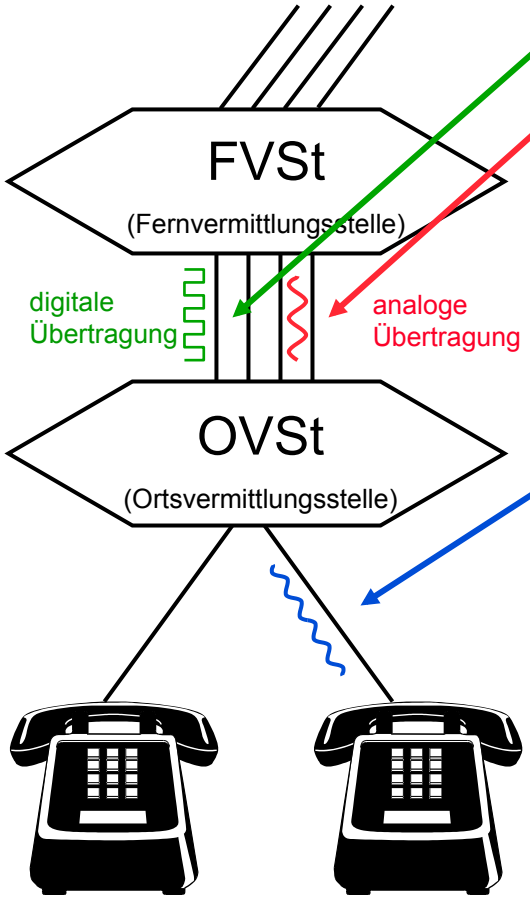
Auszug aus: 1984

With the development of television,
and the technical advance which
made it possible to receive and transmit
simultaneously on the same instrument,
private life came to an end.

George Orwell, 1948

Probleme bei Vermittlungsstellen

Durch differenzierte Gestaltung getrennter Vermittlungsstellen ungelöste Probleme:



+ Verschlüsselung
– Nutzdaten
– Verbindungsdaten, falls Sprechererkennung oder ⊂ Nutzdaten

Trojanisches Pferd v Zusatzgerät: wie unten

Abhören von Teilnehmeranschlussleitungen (Zerhacken der Signale ist aufwendig und ineffektiv, Verschlüsselung der analogen Signale nicht möglich):

– Nutzdaten (Gesprächsinhalte)
– Verbindungsdaten
  - Zielrufnummer
  - Sprechererkennung oder ⊂ Nutzdaten

Verfahren zum Schutz der Verkehrsdaten

Schutz außerhalb des Netzes

Öffentliche Anschlüsse

– Benutzung ist umständlich

Zeitlich entkoppelte Verarbeitung

– Kommunikationsformen mit Realzeitanforderungen

Lokale Auswahl

– Übertragungsleistung des Netzes

– Abrechnung von kostenpflichtigen Diensten

Schutz innerhalb des Netzes

Angreifer (-modell)

Fragen:

- wie weit verbreitet ? (Stationen, Leitungen)
- beobachtend / verändernd ?
- wie viel Rechenkapazität ? (informationstheoretisch, komplexitätstheoretisch)

Unbeobachtbarkeit eines Ereignisses E

Für Angreifer gilt für alle Beobachtungen B: $0 < P(E|B) < 1$

perfekt: $P(E) = P(E|B)$

Anonymität einer Instanz

Unverkettbarkeit von Ereignissen

gegebenenfalls Klasseneinteilung