TECHNISCHE UNIVERSITÄT DARMSTADT

COMPUTERSYSTEMSICHERHEIT

Thema 3: Symmetrische Kryptographie

Prof. Sebastian Faust

THEMENÜBERSICHT

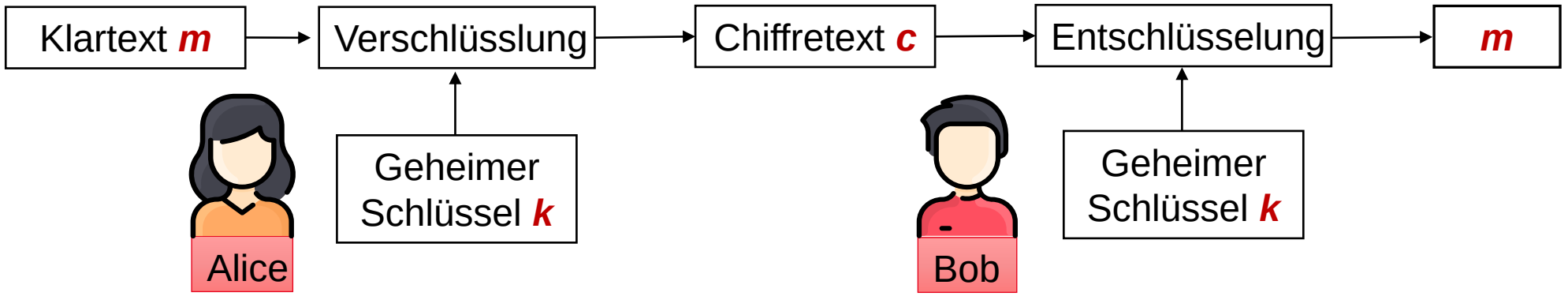
TECHNISCHE UNIVERSITÄT DARMSTADT

- Motivation Symmetrische Kryptographie
- Definition symmetrische Chiffre
- One-Time-Pad
- Blockchiffren
- Modes of Operation
- Kryptographische Hashfunktionen
- Message Authentication Codes (MACs)
- Authenticated Encryption

SYMMETRISCHE KRYPTOGRAPHIE

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Symmetrisch Kryptographie:** Es gibt nur **einen** Schlüssel für alle Algorithmen

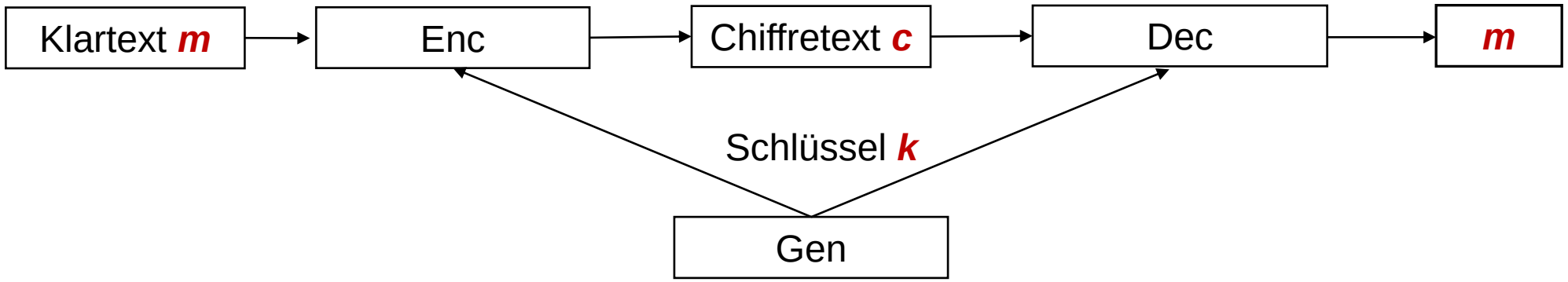


- Einigung auf geheimen Schlüssel:
  - Physikalisches Treffen
  - Kryptographisches Protokoll (später mehr)

DEFINITION SYMMETRISCHER CHIFFREN

FUNKTIONALE DEFINITION

- **Funktionale Definition:** Beschreibt Input/Output-Verhalten der Algorithmen
- **Algorithmen:** (Gen, Enc, Dec)



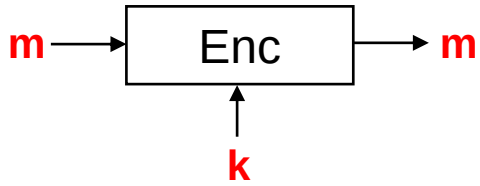
- **Korrektheit:** Die Entschlüsselung eines gültigen Chiffretexts resultiert in die original verschlüsselte Nachricht

  ➔ $Dec(k, Enc(k, m)) = m$ für alle Nachrichten m und Schlüssel $k \leftarrow Gen$

- **Effizienz:** Verschlüsselung und Entschlüsselung sind effizient (> 1GB/s)

SICHERHEITSDEFINITION

- **Ziel des Angreifers:** Was ist ein erfolgreicher Angriff?
  - *Naive Option:* Angreifer lernt den Schlüssel **k** nicht



  - *In der Kryptographie:* Angreifer lernt „**nichts Neues**“ über **m**

- **Angreifermodell:** Was kann der Angreifer tun und sehen?
  - Angreifer lernt nur Chiffretexte (*known ciphertext attack*), z.B. durch Abhören des Kanals
  - Angreifer lernt Paare von Klartexten/Chiffretexten (*known plaintext/ciphertext attack*), z.B. bestimmter Teil der verschlüsselten Nachricht kann bekannt sein
  - Angreifer wählt Klartexte und lernt zugehörige Chiffretexte (*chosen plainext attack*), z.B. Angreifer Nutzer davon überzeugen Nachrichten seiner Wahl zu verschlüsseln

SICHERHEITSSPIEL (IND-CPA)

TECHNISCHE UNIVERSITÄT DARMSTADT

Sicherheit wird in der Kryptographie durch ein **Spiel** zwischen **Angreifer** und „**Nutzer**“ definiert

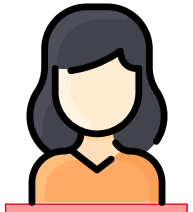
Angreifer

Nachricht m

Chiffretext $c = Enc(k, m)$

⋮

Chosen Plaintext Angriff



Nutzer

Erzeuge Schlüssel

$k \leftarrow Gen$

SICHERHEITSSPIEL (IND-CPA)

Sicherheit wird in der Kryptographie durch ein **Spiel** zwischen **Angreifer** und „**Nutzer**“ definiert

Angreifer

Gibt Bit b'

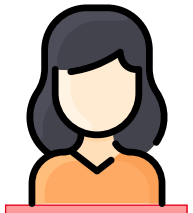
Nachricht m

Chiffretext $c = Enc(k, m)$

⋮

Nachricht m_0, m_1, $|m_0| = |m_1|$

Chiffretext $c^* = Enc(k, m_b)$



Nutzer

Erzeuge Schlüssel $k \leftarrow Gen$

Wähle zufälliges Bit b

Angreifer gewinnt das Sicherheitsspiel wenn $b = b'$

Sicherheit: Symmetrische Chiffre is **IND-CPA sicher**, falls alle „**effizienten**“ Angreifer das Sicherheitsspiel maximal mit Wahrscheinlichkeit $\approx \frac{1}{2}$ gewinnen können

IMPLIKATIONEN

TECHNISCHE UNIVERSITÄT DARMSTADT

Falls Verschlüsselungsverfahren IND-CPA sicher, dann:

➔ Das Beste was der Angreifer tun kann ist b raten

➔ Angreifer kann Chiffretext c^* im wesentlichen ignorieren

➔ Chiffretext c^* gibt keine zusätzlichen Informationen über verschlüsselte Nachricht

WIE ZEIGEN WIR (UN)SICHERHEIT?

TECHNISCHE UNIVERSITÄT DARMSTADT

Welche Annahmen machen wir über die Strategie des Angreifers? **Keine**, außer dass sie effizient sind!

Was bedeutet „**effizient**“?

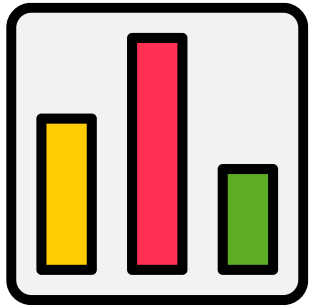
- **Nicht effizient:** hat Laufzeit exponentiell in der Schlüssellänge, bricht Faktorisierung,…
- **Effizient:** siehe „Einführung in die Kryptographie“

Unsicheres Verfahren: Konstruiere effizienten Angreifer, der mit Wahrscheinlichkeit **> ½** das Sicherheitsspiel gewinnt

Sicheres Verfahren: Zeige, dass alle effizienten Angreifer das Sicherheitsspiel mit Wahrscheinlichkeit **< ½** gewinnen

BRUTE FORCE ANGRIFF

Betrachte einen Brute Force Angriff wo der Angreifer alle Schlüssel ausprobiert, um das Verschlüsselungsverfahren anzugreifen. Bricht dieser Angreifer die IND-CPA Sicherheitseigenschaft?



- Ja, der Angreifer kann m_0 von m_1 unterscheiden.
- Nein, der Angreifer lernt nichts von c^* über m_b.
- Nein, der Angreifer ist nicht effizient.

pingo.coactum.de
Nr.: 643569

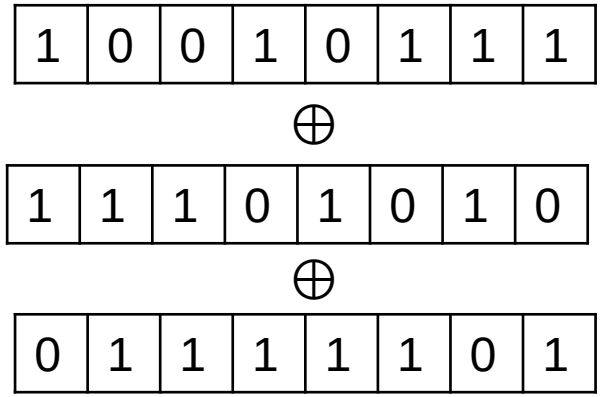
ONE-TIME PAD VERSCHLÜSSELUNG

WIEDERHOLUNG: XOR

- Bit XOR Operationen

$x \oplus 0 = x$
$x \oplus x = 0$
$x \oplus y = y \oplus x$
$(x \oplus y) \oplus z = x \oplus (y \oplus z)$
$(x \oplus y) \oplus x = y$

Erweiterung auf Bitstrings



ONE-TIME PAD

Wird häufig auch Vernam Chiffre genannt nach Gilbert Vernam

One-Time Pad zur Verschlüsselung von Bitstrings der Länge n

- **Gen:** Ausgabe zufälliger Schlüssel $k \xleftarrow{R} \{0,1\}^n$
- **Enc:** Für $m \in M$: Ausgabe $\mathbf{Enc}(k,m) = k \oplus m$.
- **Dec:** Für $c \in C$: Ausgabe $\mathbf{Dec}(k,c) = k \oplus c$.

Gilbert Vernam
(1890-1960)

Beispiel:

m =	1	0	0	1	0	1	1	1
	⊕							
k =	1	1	1	0	1	0	1	0
	=							
c =	0	1	1	1	1	1	0	1

Korrektheit

Für jedes k und m gilt:

$$\mathbf{Dec}\big(k, \mathbf{Enc}(k,m)\big) = \mathbf{Dec}(k, m \oplus k) = k \oplus m \oplus k = m.$$

SICHERHEIT: ONE-TIME PAD

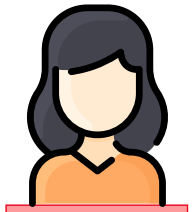
Beschränktes Sicherheitsspiel: Angreifer erhält keinen Zugriff auf Chiffretexte (wichtig: siehe nächste Slide)

Angreifer

Gibt Bit b'

Nachricht m_0, m_1

Chiffretext $c^* = k \oplus m_b$



Nutzer

Erzeuge Schlüssel $k \leftarrow \{0,1\}^n$

Wähle zufälliges Bit b

Angreifer gewinnt das Sicherheitsspiel wenn $b = b'$

Perfekte Sicherheit: c^*gibt keine Information über m_b preis

Heißer Draht

Dieser Artikel erläutert die Kommunikationsverbindung, für das Geschicklichkeitsspiel siehe Heißer Draht (Spiel), für die Samstagabendshow siehe Der heiße Draht, für die Zeitung siehe Der heisse Draht.

Als **Heißer Draht** (engl. *hotline*) oder **Rotes Telefon** wurde eine ständige Fernschreiberverbindung zwischen der Sowjetunion und den Vereinigten Staaten während der Zeit des Kalten Krieges bezeichnet.

Inhaltsverzeichnis [Verbergen]

1 Einrichtung
2 Erster Einsatz
3 Reaktivierung
4 Sonstiges
5 Literatur
6 Einzelnachweise

Einrichtung [Bearbeiten | Quelltext bearbeiten]

Die Verbindung wurde aufgrund der Erfahrungen aus der Kubakrise (14. bis 28. Oktober 1962) eingerichtet. Sie läuft über London, Kopenhagen, Stockholm, Helsinki und wurde am 30. August 1963[1] eröffnet. Parallel dazu kam es zu einer Funkverbindung über Tanger. 1966 folgte eine Verbindung der USA mit Frankreich, 1967 mit Großbritannien. Die Verbindungen sollen die Möglichkeit schaffen, eine Friedensgefährdung durch Irrtümer, Missverständnisse oder Verzögerungen im Kommunikationsweg zu verhindern.

Dass diese Gefahr weiterhin bestand, zeigt ein Vorfall vom 26. September 1983: Damals meldete die sowjetische Raketenabwehr fälschlich einen Angriff von US-Interkontinentalraketen auf die Sowjetunion (siehe den Artikel über Stanislaw Jewgrafowitsch Petrow).

Ein Rotes Telefon aus der Zeit von Jimmy Carter, das aber nie Bestandteil des heißen Drahtes, sondern wohl nur des amerikanischen *Defense Red Switch Networks* war.

Beide Seiten strebten eine höchstmögliche Sicherheit gegen Abhören oder Verfälschen der übermittelten Nachrichten an. Es kam die Verschlüsselungstechnik *One-Time-Pad* zum Einsatz – eine der wenigen bekannten Anwendungen des Verfahrens, das zwar absolute Sicherheit bietet, in der Praxis aufgrund des aufwendigen Schlüsselaustausches aber nur schwer durchführbar ist.

SCHLÜSSEL NUR EINMAL VERWENDEN

TECHNISCHE UNIVERSITÄT DARMSTADT

One-Time Pad ist unsicher bei Wiederverwendung des gleichen Schlüssels

$$m_0 \oplus k = c_0$$

$$\oplus$$

$$m_1 \oplus k = c_1$$

$$c_0 \oplus c_1 = m_0 \oplus m_1$$

Lernen des XORs kann nützliche Informationen preisgeben:

- z.B. welche Bits von m_0 und m_1 gleich sind
- Wenn m_0 bekannt ist dann ist auch m_1 bekannt (und vice versa)

➔ **Moral:** Zur Verschlüsselung jeder Nachricht muss ein neuer zufälliger Schlüssel gewählt werden

NACHTEILE ONE-TIME PAD

TECHNISCHE UNIVERSITÄT DARMSTADT

1. Schlüssel ist **so lang** wie Nachricht

➔ Für große Mengen von Daten müssen lange zufällige Schlüssel gespeichert und ausgetauscht werden

➔ Gute Zufälligkeit zu erzeugen, ist sehr aufwendig

2. Schlüssel **kann nur einmal benutzt** werden:

➔Kann etwas über Klartexte preisgeben

3. Sicherheit im **beschränkten** Angreifermodell

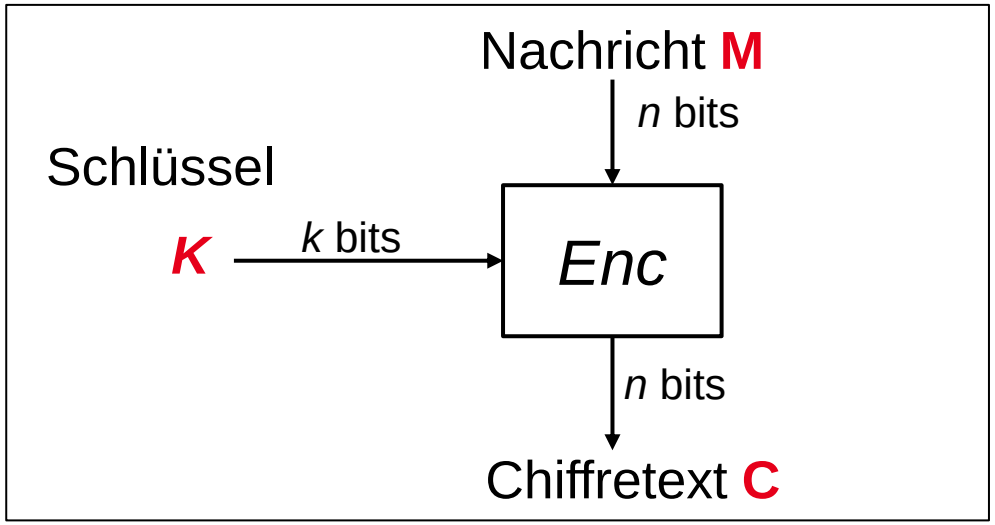
➔ Chiffretext-Only Angriffe

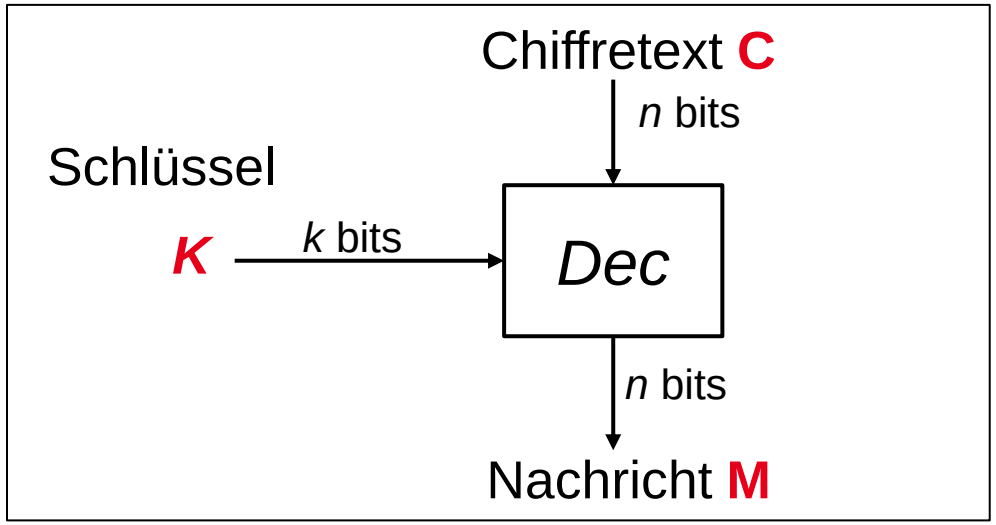
BLOCKCHIFFREN

BLOCKCHIFFRE

TECHNISCHE UNIVERSITÄT DARMSTADT

Verschlüsselung/Entschlüsselung von Nachrichten/Chiffretextblöcken mit **fixer Länge**





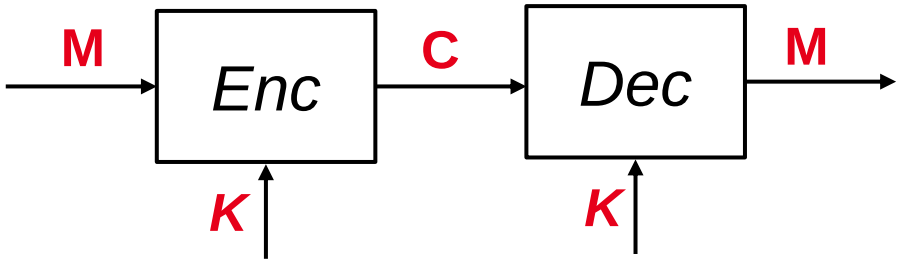
- **Blocklänge n = |m| = |c|**: häufig 64 – 128 Bits
- **Schlüssellänge k**: häufig 128 – 256 Bits

Wesentlicher Unterschied zu One-Time-Pad: Schlüssel kann wiederverwendet werden!

EIGENSCHAFTEN VON BLOCKCHIFFREN

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Korrektheit:** Für jeder Nachricht **M** und jeden Schlüssel **K**

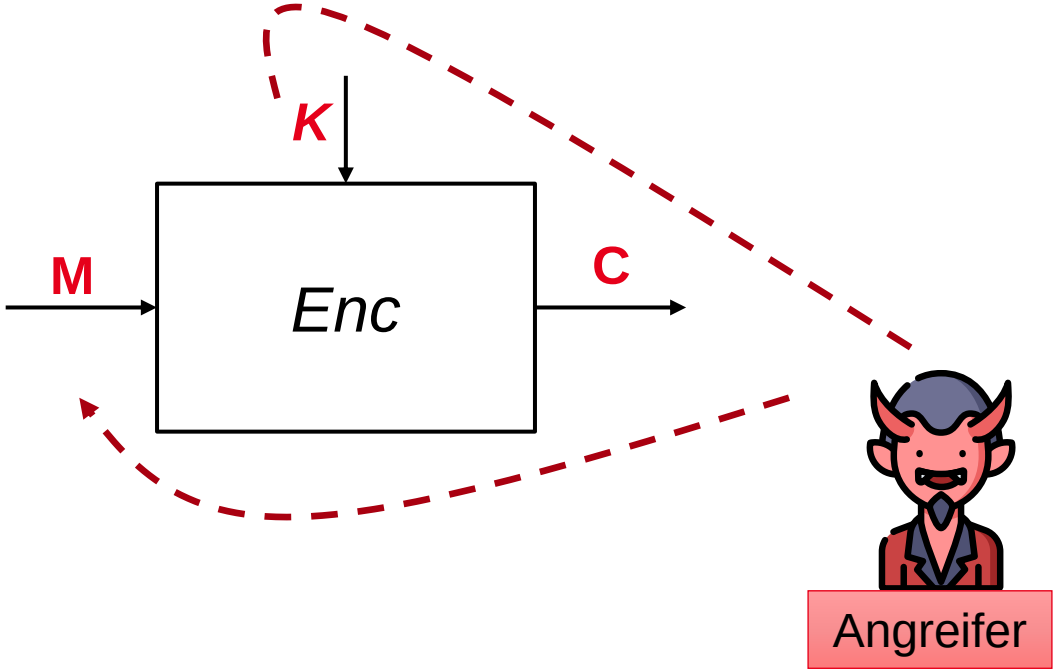


- **Effizienz:**
  - **Enc(.)** und **Dec(.)** sollte in Mikrosekunden berechenbar sein: XOR, bit-shifting, Lookup Tabellen, …
  - Häufig Hardware Support auf modernen CPUs
- **Sicherheit:**
  - Was soll ein Angreifer nicht können?

SICHERHEIT VON BLOCKCHIFFREN

TECHNISCHE UNIVERSITÄT DARMSTADT

Was soll ein Angreifer nicht können?



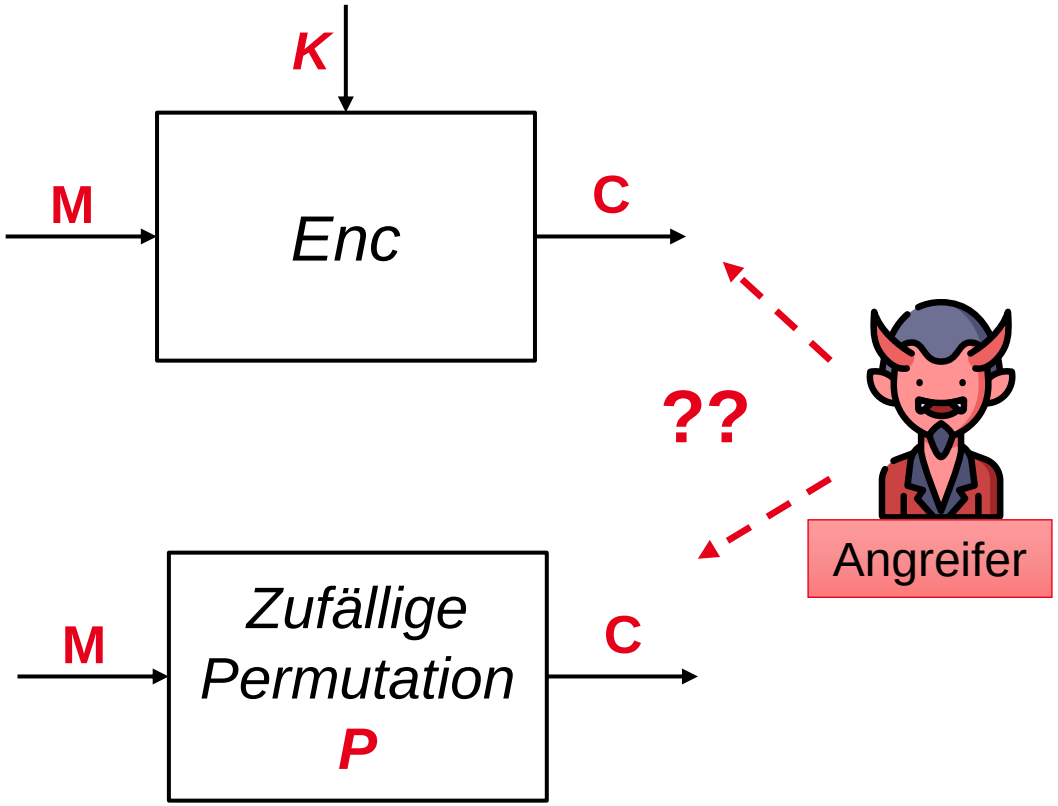
Aus **C** kann der Angreifer nicht…

- **M** lernen
- **K** lernen

SICHERHEIT VON BLOCKCHIFFREN

TECHNISCHE UNIVERSITÄT DARMSTADT

Was soll ein Angreifer nicht können?



Aus **C** kann der Angreifer nicht…

- **M** lernen
- **K** lernen

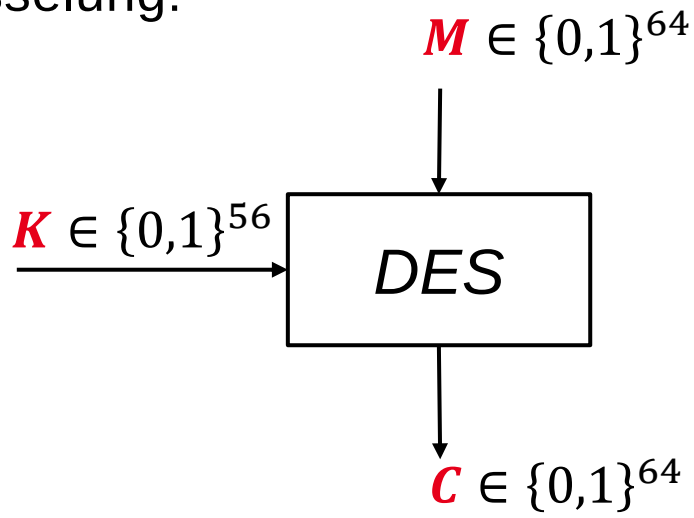
Formal: Angreifer kann nicht zwischen **Enc(.)** und **P(.)** unterscheiden.

DATA ENCRYPTION STANDARD (DES)

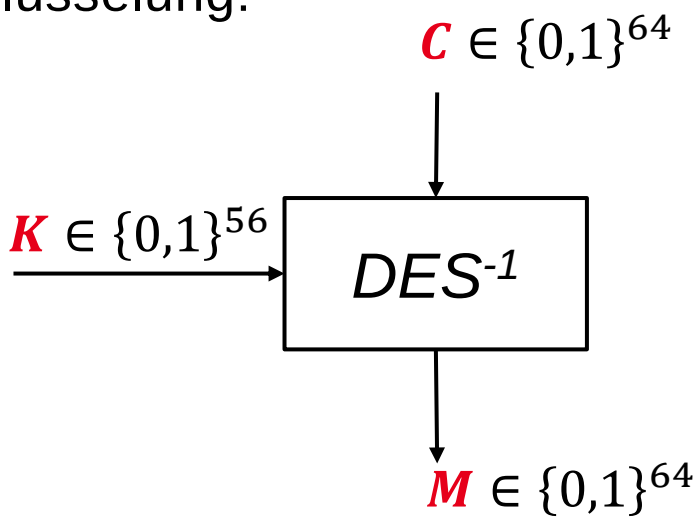
TECHNISCHE UNIVERSITÄT DARMSTADT

Entwickelt von IBM im Auftrag des NIST Ende der 1970er

Verschlüsselung:



Entschlüsselung:

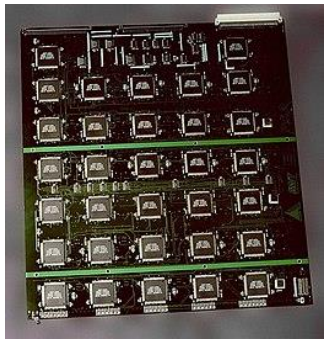


- Blocklänge **n = 64** Bits
- Schlüssellänge **k = 56** Bits

DES: ANGRIFFE

Theoretische Angriffe: Differenzielle Kryptoanalyse und **Lineare Kryptoanalyse**

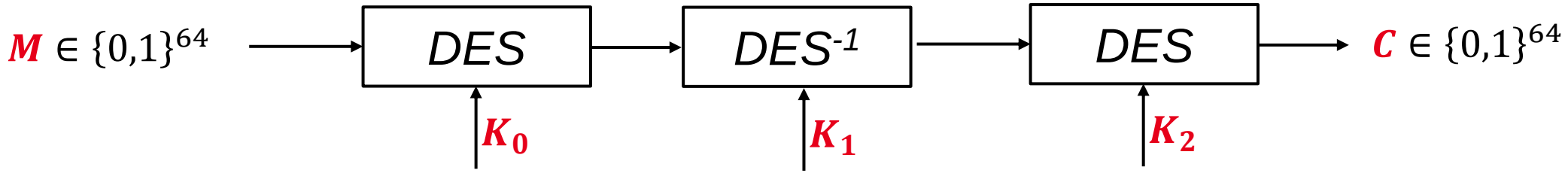
- DES bietet Schutz gegen Differenzielle Kryptoanalyse



Hauptschwachpunkt: Kurzer Schlüssel (nur 56 Bits)

- Brute-Force Angriff ist möglich
- DES Cracker: Bricht DES in wenigen Tagen

Triple DES: Sicherheitsniveau ist 2 x 56 Bits (wegen Time-Memory-Trade-Off)



Quellen: Wikipedia

ADVANCED ENCRYPTION STANDARD (AES)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Wettbewerb für **AES** wurde im **Januar 1997** von US **National Institute of Standards and Technology** (NIST) angekündigt.
- **15** Chiffren wurden eingereicht
- **5** Finalisten: **MARS, RC6, Rijndael, Serpent**, und **Twofish**
- **2. October, 2000**: **Rijandel** wurde als Gewinner gewählt.
- **26. November, 2001**: **AES** ist ein offizieller Standard geworden.
- Autoren: **Vincent Rijmen**, **Joan Daemen** (aus KU Leuven in Belgien)
- **Schlüssel-Größe**: **128**, **192** or **256** bit, **Block-Größe**: **128** Bits (weitere werden von Rijndael unterstützt)
- **Gilt als ungebrochen**: In den USA Zulassung für höchste Geheimhaltungsstufe

Vincent Rijmen (1970)

Joan Daemen (1965)

BRUTE FORCE ANGRIFFE

TECHNISCHE UNIVERSITÄT DARMSTADT

Wie schwer ist ein Brute-Force-Angriff auf einen **128** Bit Schlüssel?

- Probiere 2^{128} Schlüssel aus

Wie groß ist 2^{128}?

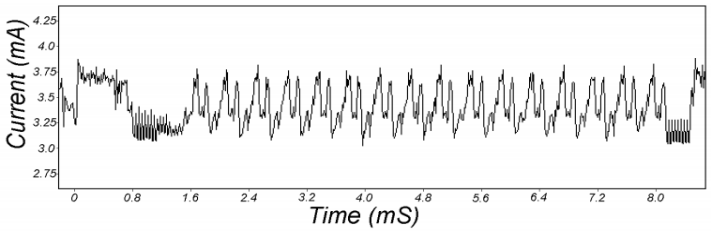
- $2^{128} = 2^{10*12.8} \approx 10^{3*13} = 10^{39}$

Angenommen wir haben Zugriff auf eine extrem leistungsstarke Hardware, die 10^9 (1 Milliarden) Schlüssel pro **Nanosekunde** testen kann

- D.h.: 10^{18} Schlüssel pro Sekunde
- D.h.: Wir benötigen $10^{39}/10^{18} = 10^{21}$ Sekunden
- **Das sind ca. 30 Billionen Jahre**

Takeaway: Brute-Force Angriffe auf moderne Chiffren mit Schlüsseln ≥ 128 Bits unmöglich

ANGRIFFE IN DER PRAXIS

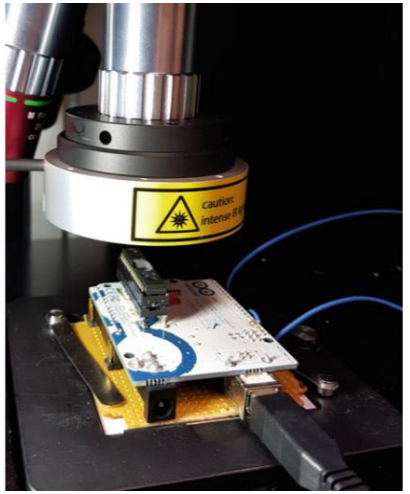


1. Seiten-Kanal-Angriffe

- **Beobachte** das Gerät bei Ent-/Verschlüsselung
  - Messe die **Zeit**, die für kryptographische Operationen benötigt wird.
  - Messe den **Stromverbrauch**

2. Fehlerangriffe

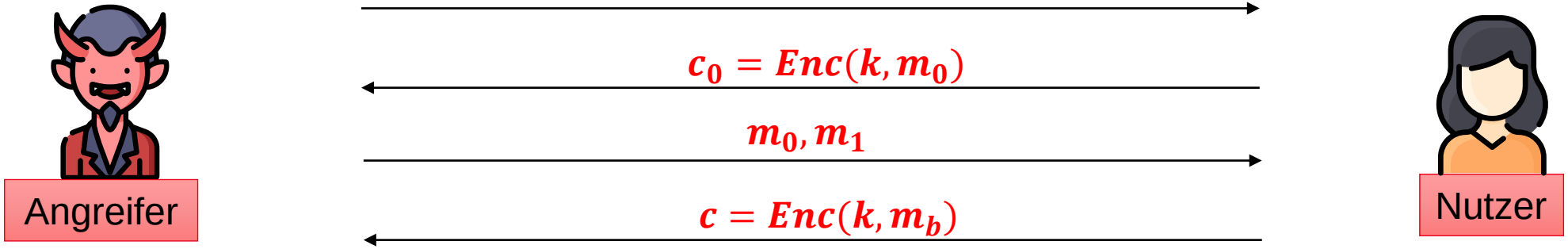
- **Füge Fehler** (z.B. mit einem Laserstrahl) in die Berechnung ein
- Beobachte die **Veränderungen** im Verhalten der **Ein-/Ausgabe**
- z.B. Berechnugsfehler in den letzten paar Runden von DES erlauben es, den Schlüssel **K** zu berechnen



BLOCKCHIFFREN IM EINSATZ

Blockchiffren besitzen zwei Probleme bei direktem Einsatz als Verschlüsselung:

- Nicht IND-CPA sicher:



Gib **0** aus falls $c = c_0$; ansonsten **1**

Merke: Deterministische Verschlüsselung kann nicht IND-CPA sicher sein

- Nicht möglich Nachrichten beliebiger Länge zu entschlüsseln

MODES OF OPERATION

MODES OF OPERATION

TECHNISCHE UNIVERSITÄT DARMSTADT

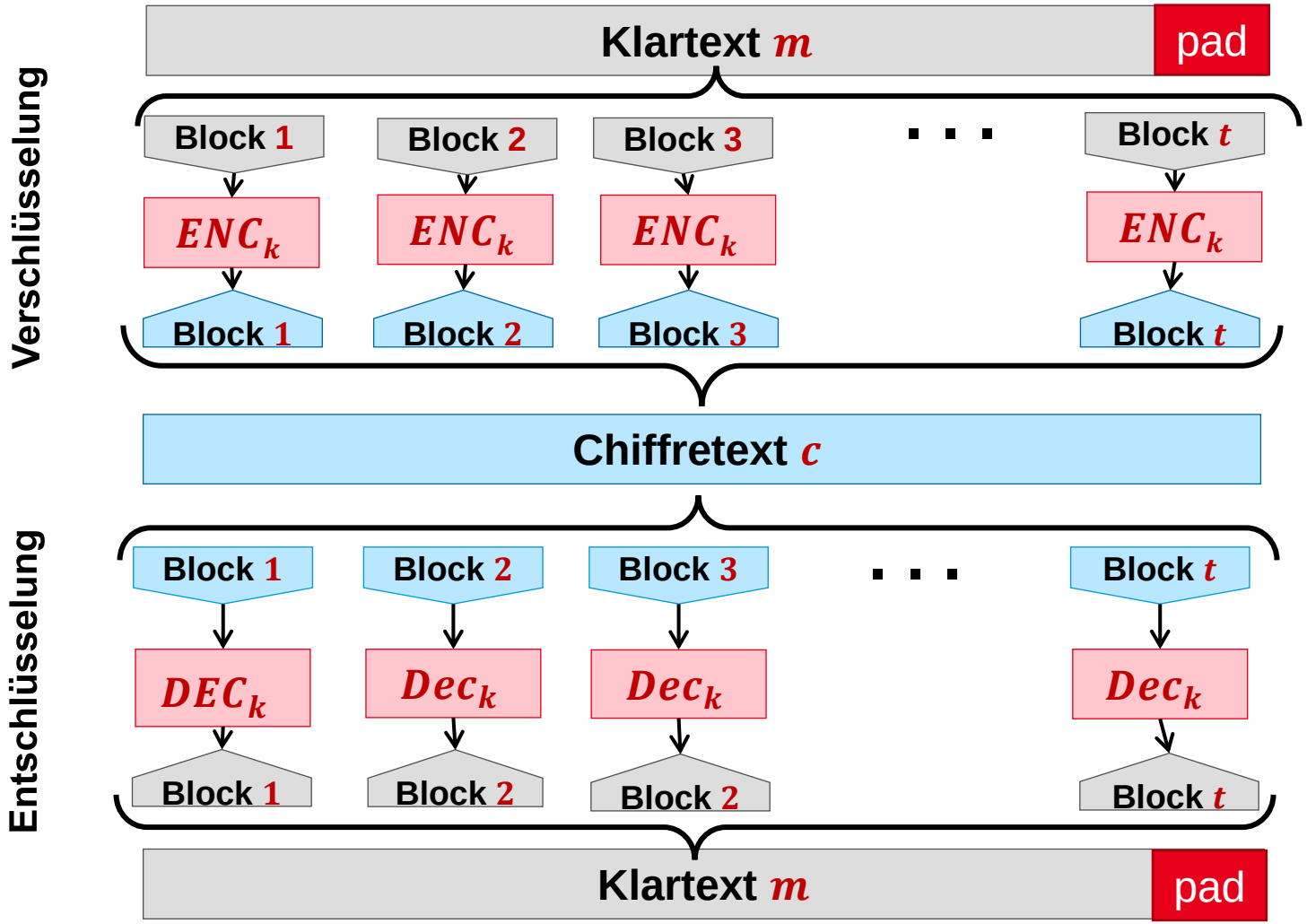
Ziel: Verschlüsselung von Nachrichten mit **beliebiger Länge**

- Ohne Vergrößerung des Chiffretexts (im Vergleich zum Klartext)
- Blockchiffren **können nicht direkt für die Verschlüsselung benutzt werden**.
- Sie werden immer in bestimmten "**Modes of Operations**" benutzt

1. **Electronic Codebook (ECB)** Modus ← **nicht sicher,**
2. **Cipher-Block Chaining (CBC)** Modus,
3. **Counter (CTR)** Modus,

. . .

ELECTRONIC CODE BOOK (ECB) MODUS



Wenn **|m|** kein Vielfaches der **Blocklänge**, dann muss der Klartext um ein Padding „**pad**“ ergänzt werden

SCHWÄCHE DES ECB MODUS

TECHNISCHE UNIVERSITÄT DARMSTADT

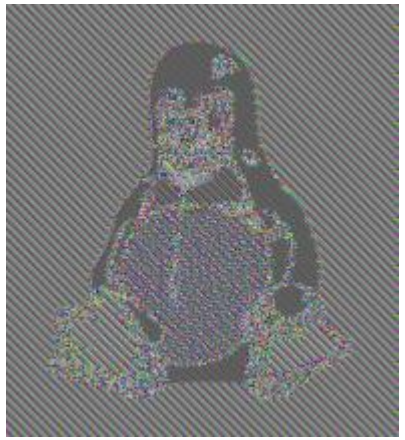
ECB Modus ist nicht sicher, da **deterministisch**!

$$M_1 = M_2 \rightarrow C_1 = C_2$$

Bilddatei Format

W	W	W
W	B	B
W	B	B
W	Y	Y
B	W	…

ECB



Quelle: Wikipedia

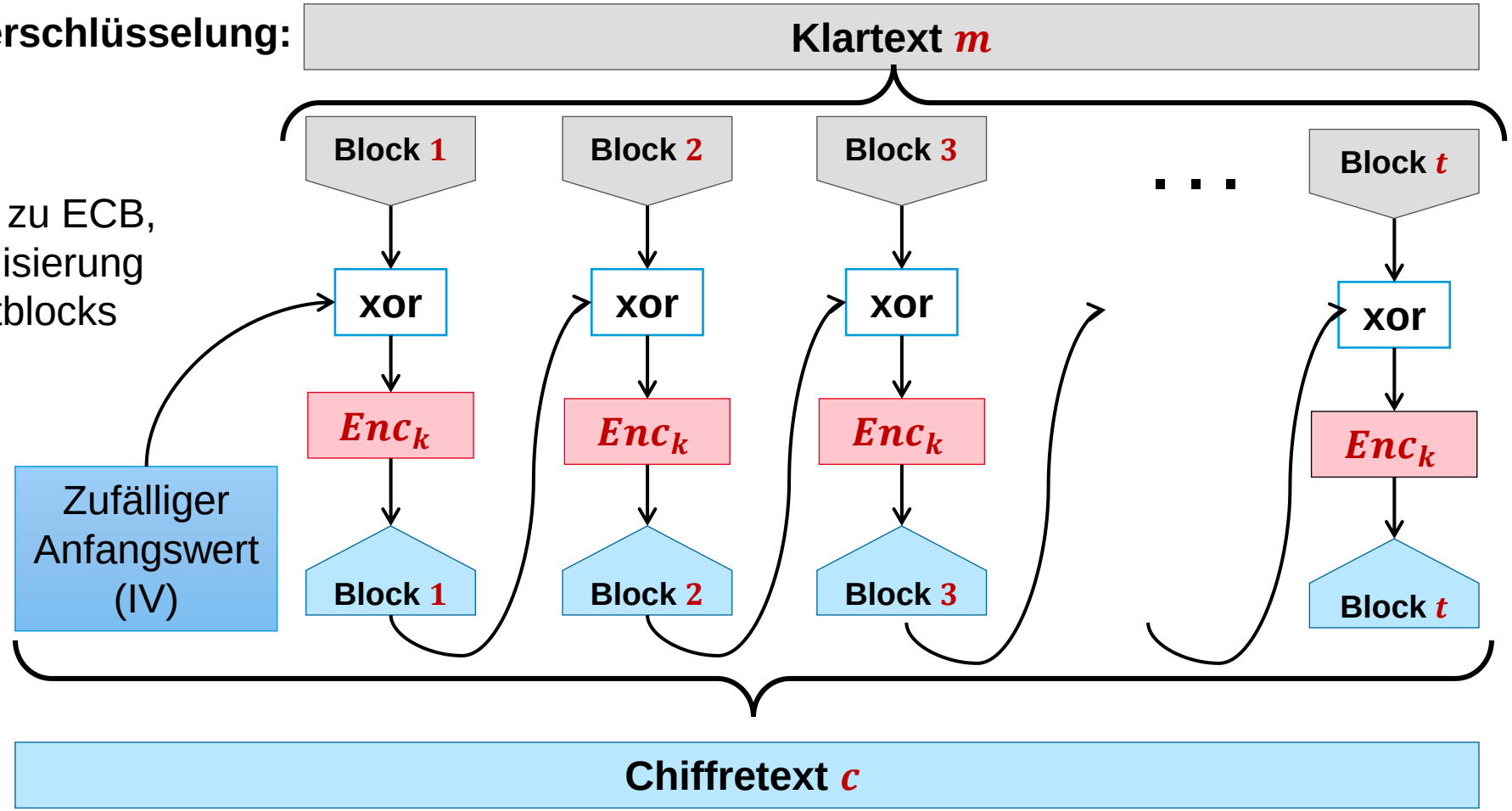
Verschlüsselte Bilddatei

E(k,W)	E(k,W)	E(k,W)
E(k,W)	E(k,B)	E(k,B)
E(k,W)	E(k,B)	E(k,B)
E(k,W)	E(k,Y)	E(k,Y)
E(k,B)	E(k,W)	…

CIPHER BLOCK CHAINING (CBC) MODUS

TECHNISCHE UNIVERSITÄT DARMSTADT

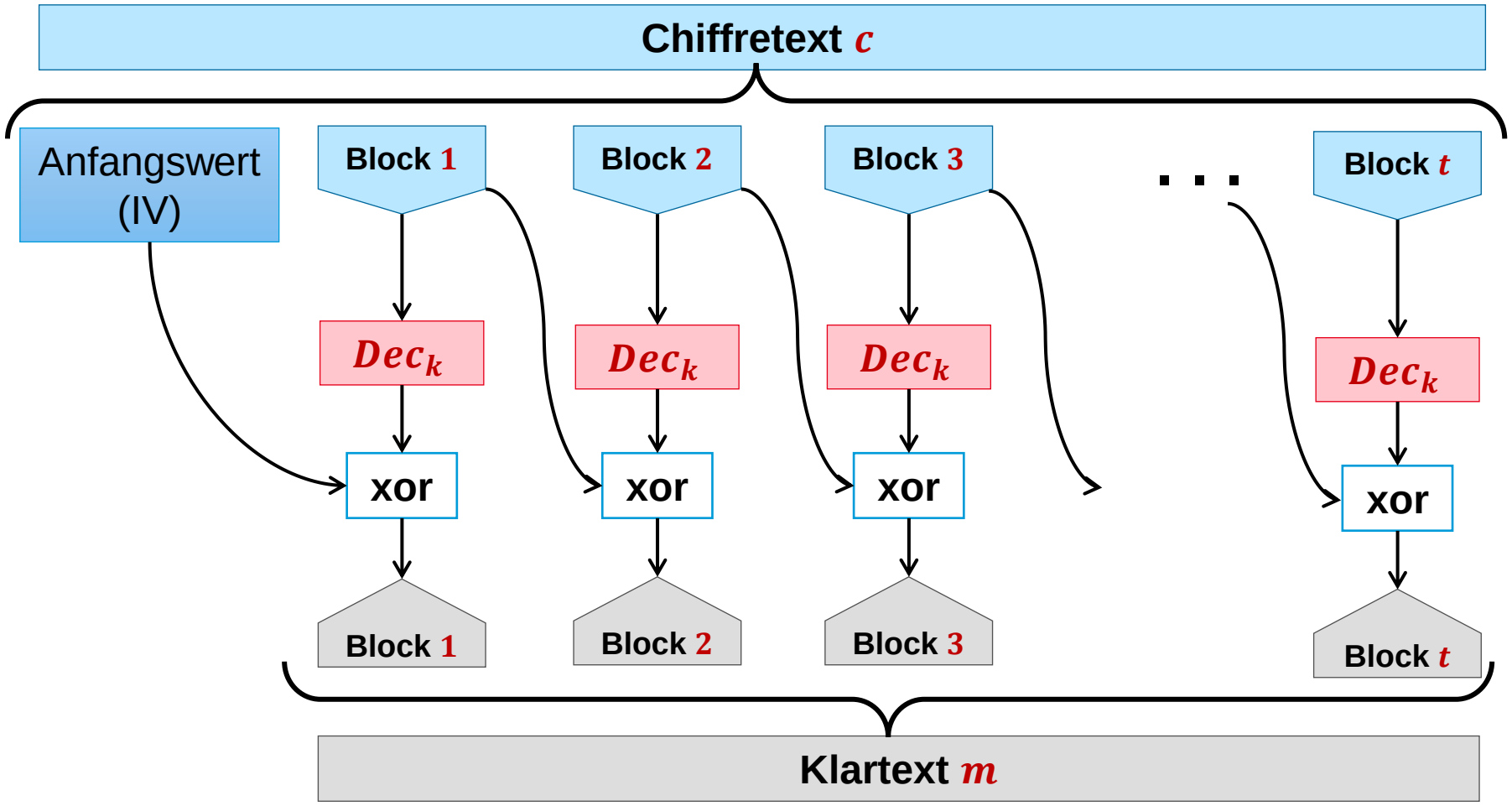
Idee: Ähnlich zu ECB, aber Randomisierung jedes Klartextblocks



CIPHER BLOCK CHAINING (CBC) MODUS

TECHNISCHE UNIVERSITÄT DARMSTADT

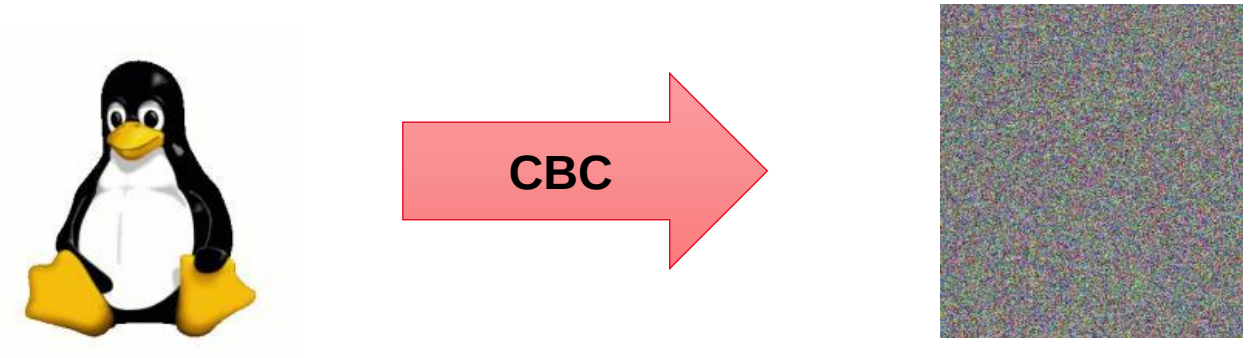
Entschlüsselung:



CBC MODUS: SICHERHEIT

Was passiert, wenn zweimal die gleiche Nachricht verschlüsselt wird?

$M_1 = M_2$ ➔ $C_1 \neq C_2$ (wegen zufälliger IV)



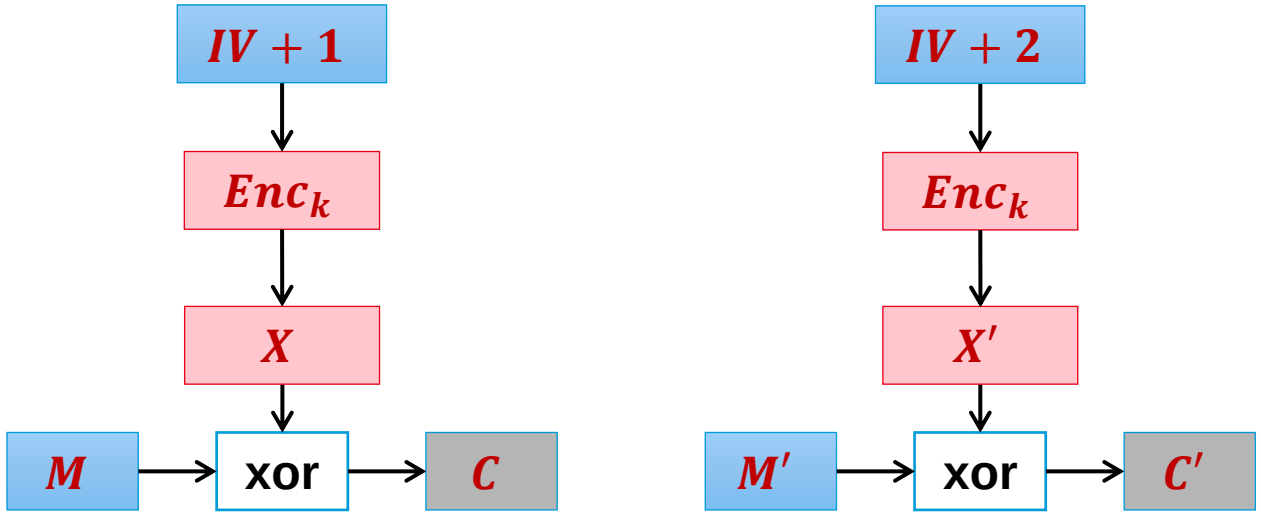
CBC ist **sicher**, wenn richtig verwendet!

Achtung: Probleme mit Padding sind häufig in der Praxis!

COUNTER MODUS (CTR)

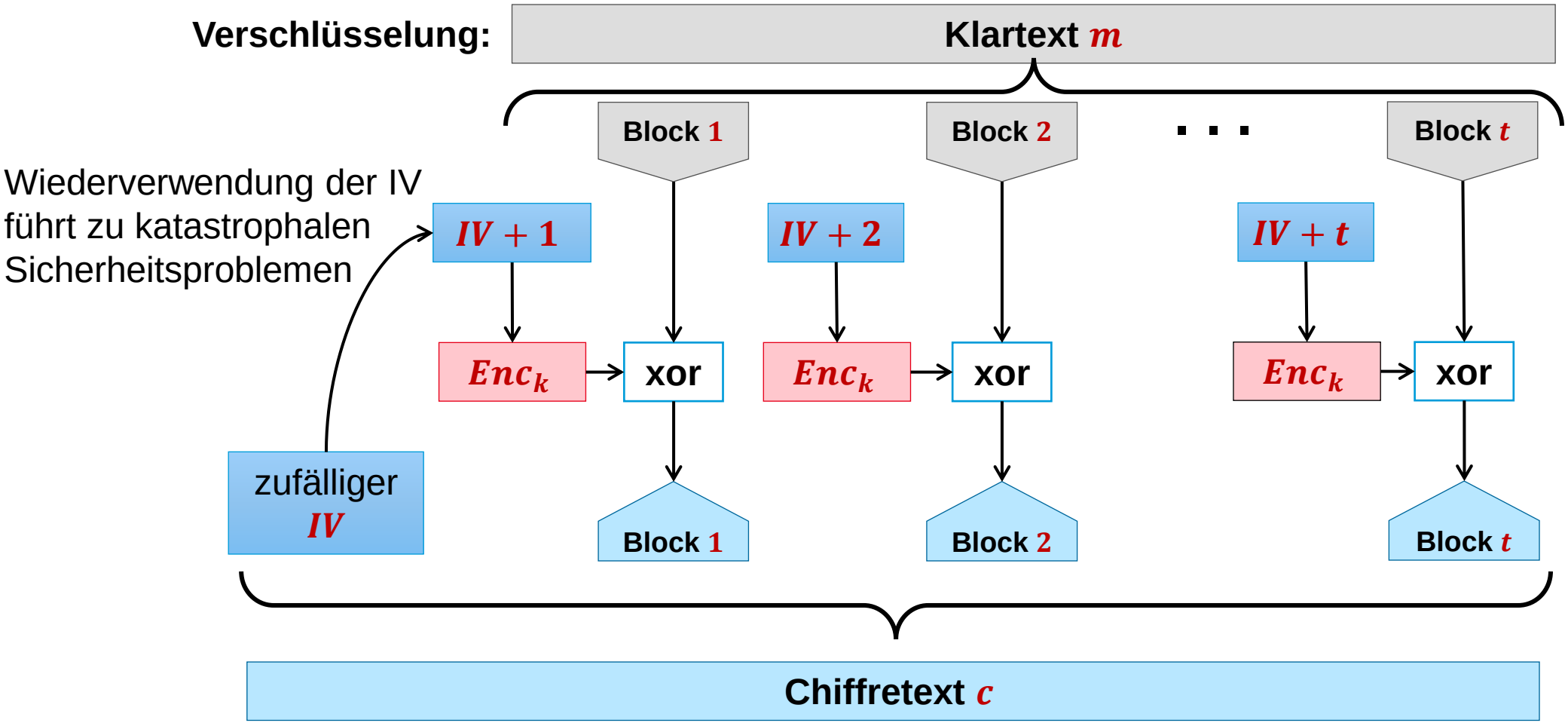
Wiederholung: One-Time-Pad

- Wenn Schlüssel **X** zufällig und nicht wiederverwendet wird dann sicher



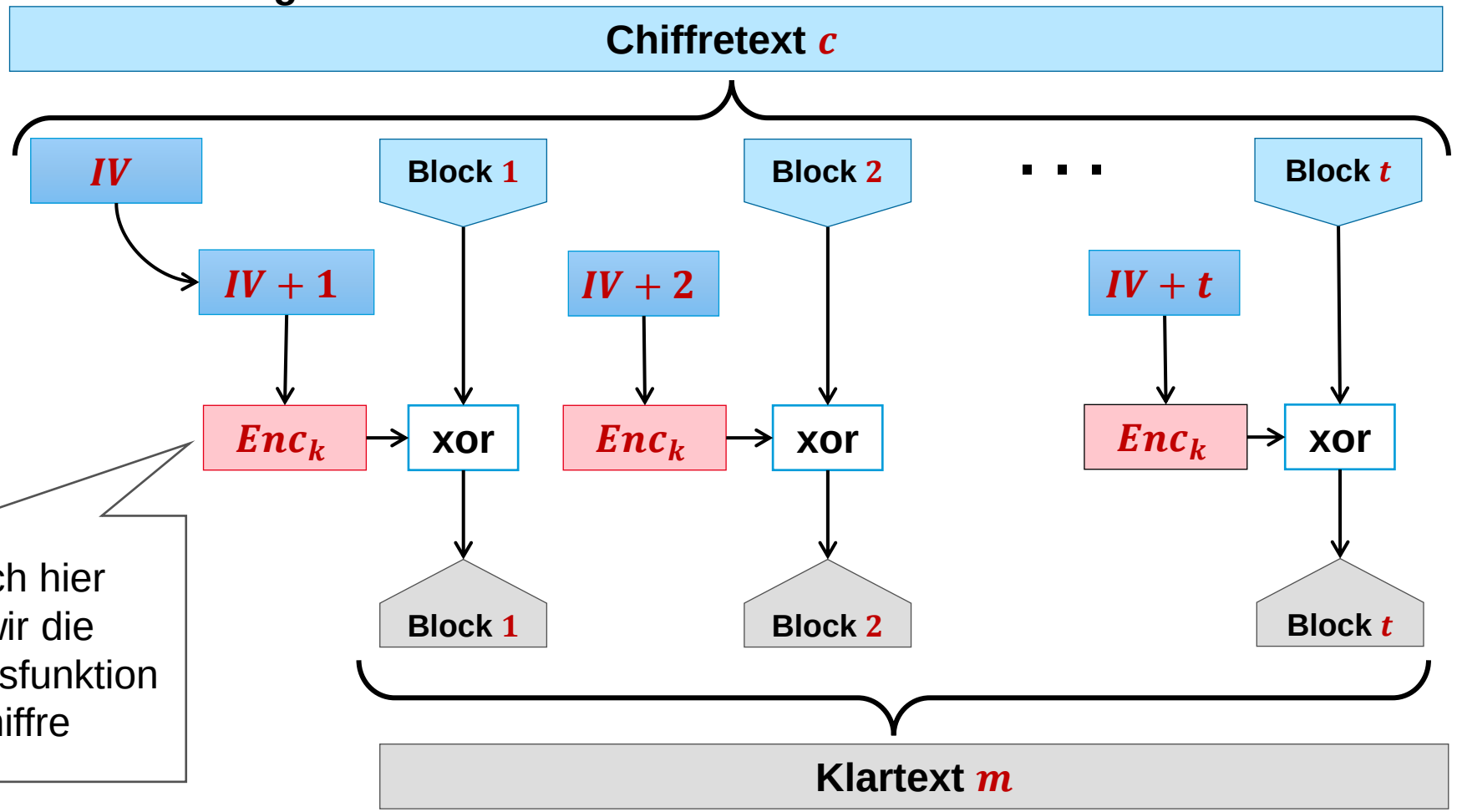
Idee: Nutze Ausgabe der Blockchiffre als One-Time-Pad

COUNTER MODUS (CTR)



COUNTER MODUS (CTR)

Entschlüsselung:



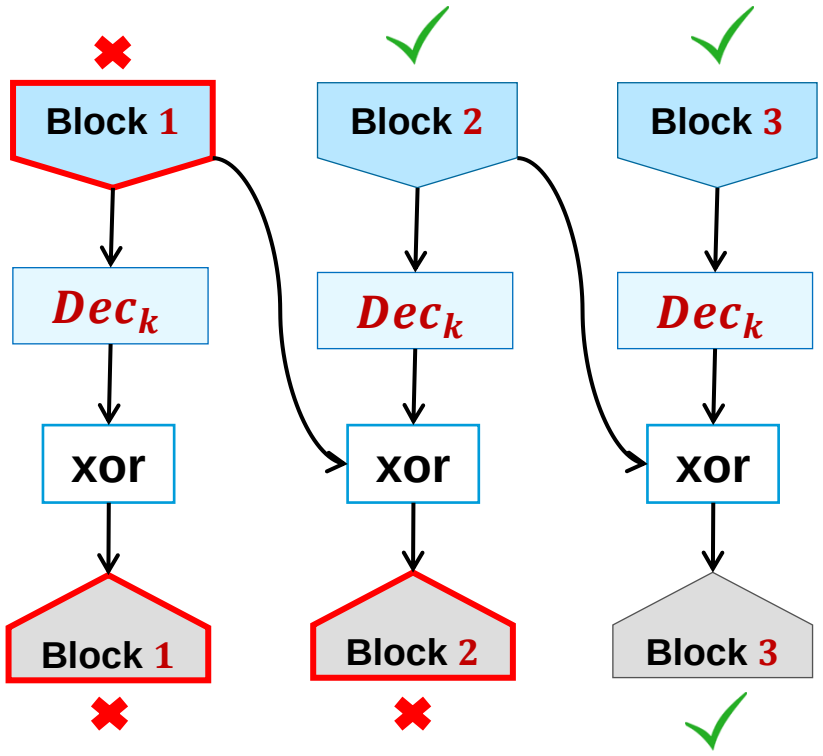
Beachte: Auch hier verwenden wir die Verschlüsselungsfunktion der Blockchiffre

FEHLERRESISTENZ

TECHNISCHE UNIVERSITÄT DARMSTADT

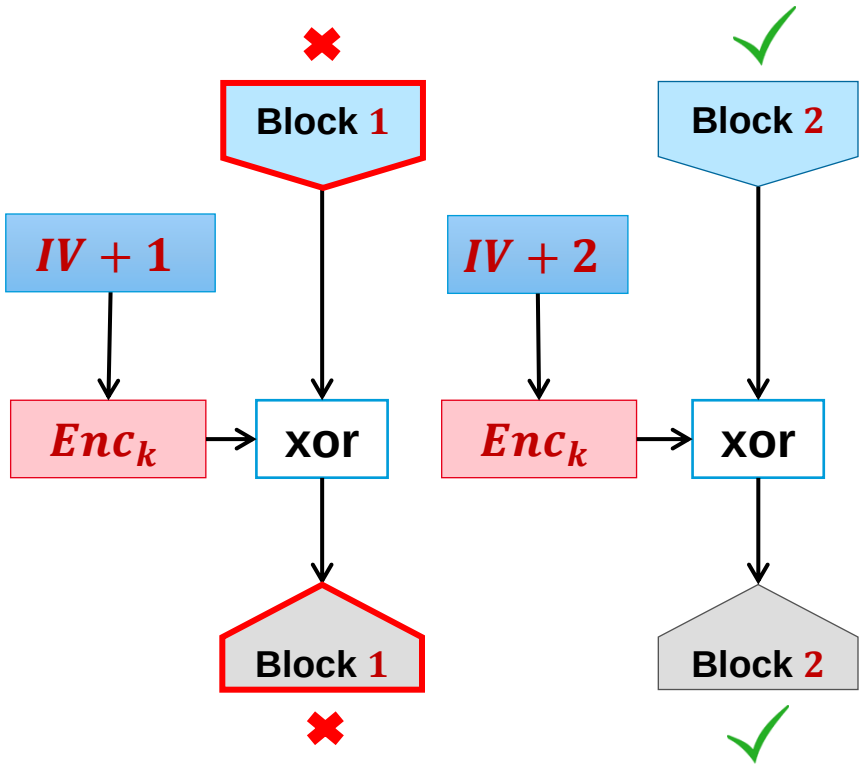
CBC Modus

- Fehler im c_1 betrifft nur Blöcke m_1 und m_2



CRT Modus

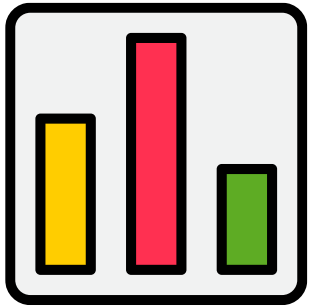
- Fehler im c_1 betrifft nur Blöcke m_1



PARALLELISERBARKEIT

TECHNISCHE UNIVERSITÄT DARMSTADT

Können im CBC-Modus oder CTR-Modus Blöcke parallel verschlüsselt werden?



- CBC ist parallelisierbar.
- CBC ist NICHT parallelisierbar.
- CTR ist parallelisierbar.
- CTR ist NICHT parallelisierbar.

pingo.coactum.de
Nr.: 877368

VERGLEICH DER MODI

TECHNISCHE UNIVERSITÄT DARMSTADT

CBC Modus

Fehlerresistenz

- Fehler verbreiten sich nicht über mehr als zwei Blöcke

Parallelisierung

- Verschlüsselung: Nein
- Entschlüsselung: Ja

Sicherheit

- Ja, aber Achtung bei Wahl des Paddings

CTR Modus

Fehlerresistenz

- Fehler wirken sich nur auf den aktuellen Block aus

Parallelisierung

- Verschlüsselung: Ja
- Entschlüsselung: Ja

Sicherheit

- Ja, aber Achtung auf Zufälligkeit der IV

ÜBERSICHT BETRIEBSMODI

TECHNISCHE UNIVERSITÄT DARMSTADT

Nicht authentifiziert

- **Electronic Code Book (ECB)**
- **Cipher Block Chaining (CBC)**
- **Counter Mode (CTR)**
- Cipher Feedback Mode (CFB)
- Ouput Feedback Mode (OFB)

Authentifiziert

- Galois Counter Mode
- Counter Mode with CBC-MAC (CCM)

TECHNISCHE UNIVERSITÄT DARMSTADT

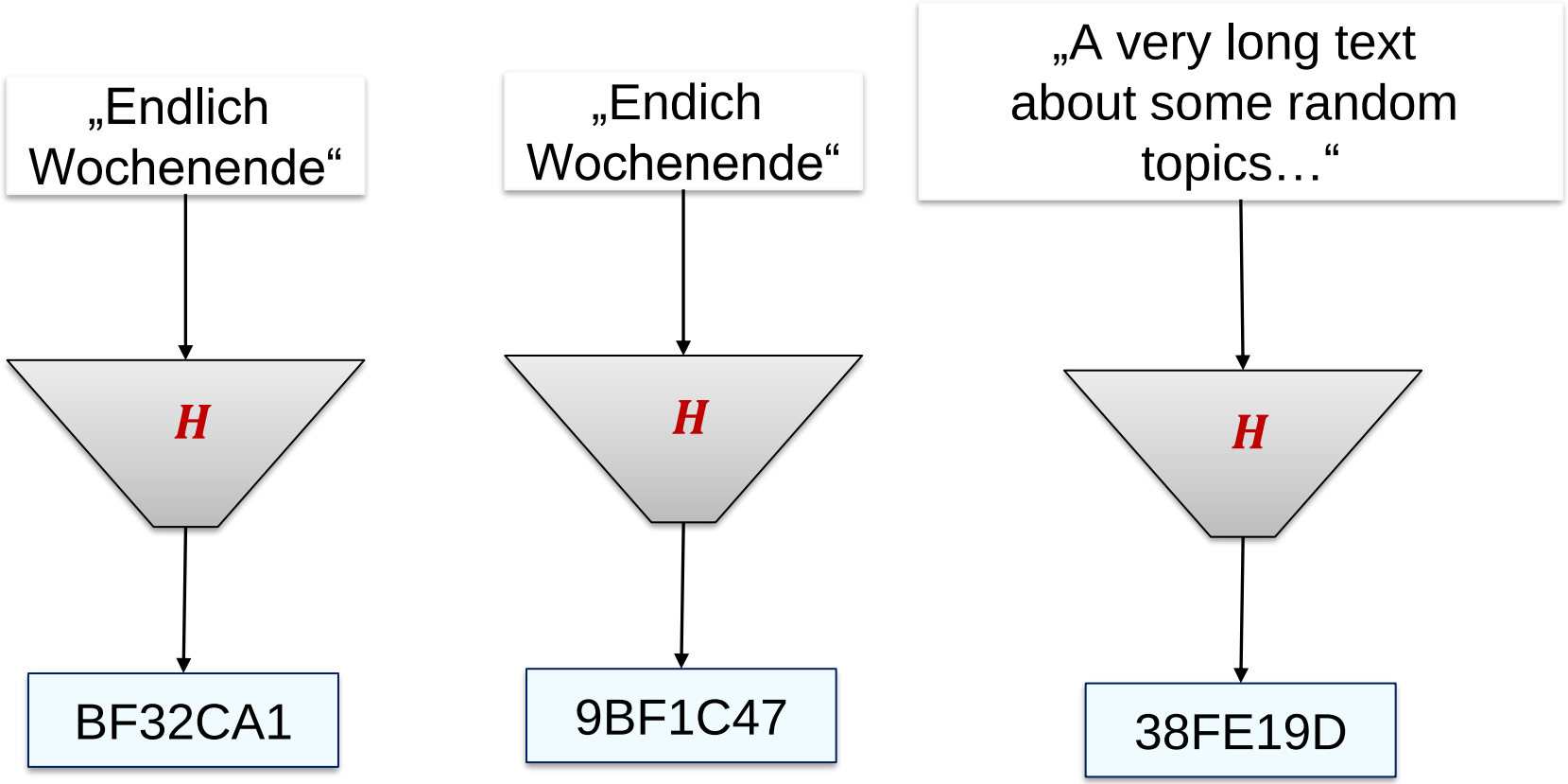
KRYPTOGRAPHISCHE HASHFUNKTIONEN

HASHFUNKTIONEN

TECHNISCHE UNIVERSITÄT DARMSTADT

Eingabe von Nachricht **beliebiger Länge**

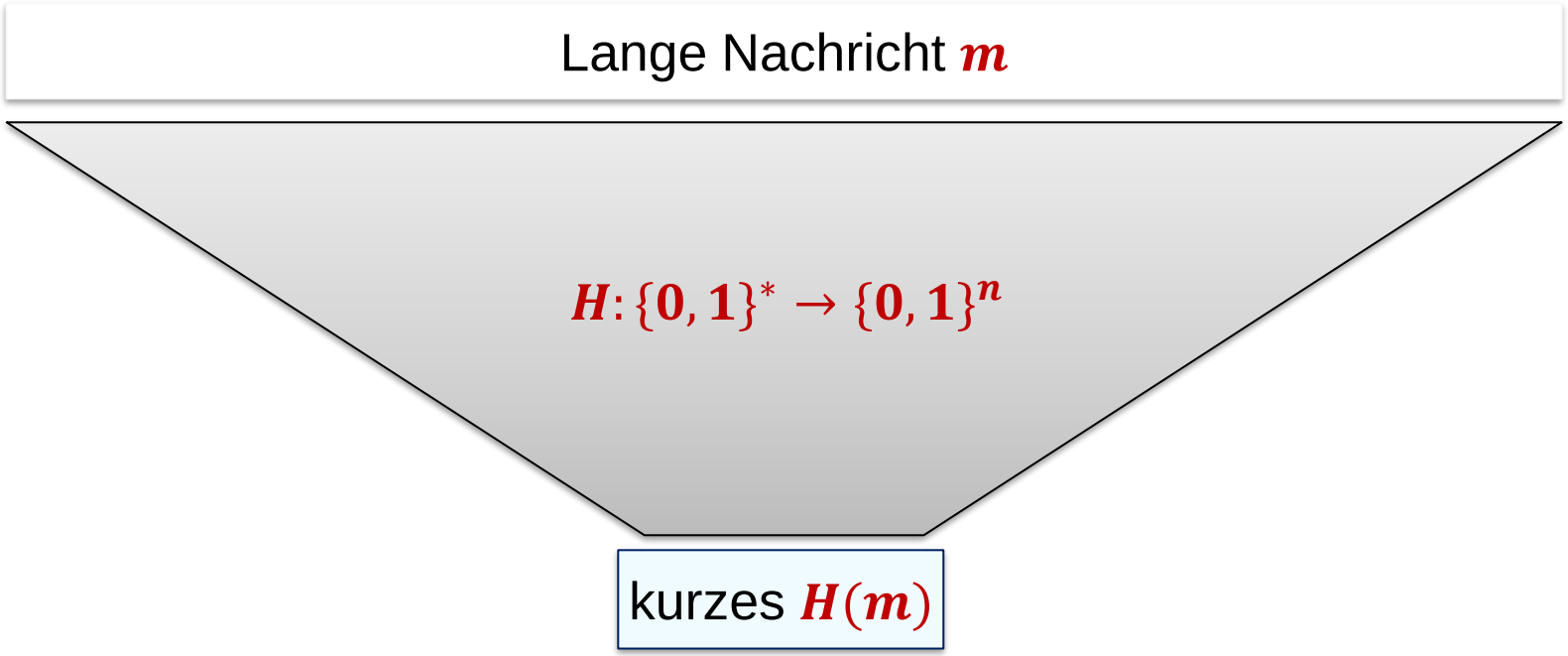
Ausgabe mit **fixer Länge** („message digest“)



HASHFUNKTION: DEFINITION

TECHNISCHE UNIVERSITÄT DARMSTADT

$H: \{0,1\}^* \rightarrow \{0,1\}^n$ bildet von großer Definitionsmenge auf kleinen Bildbereich ab

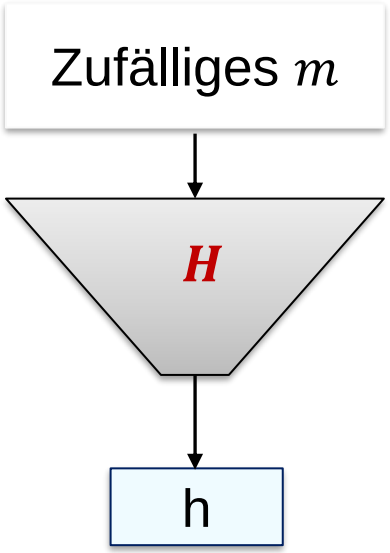


SICHERHEITSDEFINITIONEN

TECHNISCHE UNIVERSITÄT DARMSTADT

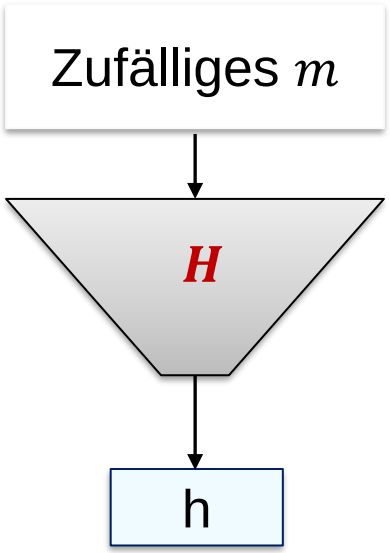
Es existieren 3 wichtige Sicherheitsdefinitionen für Hashfunktionen

Preimage resistance



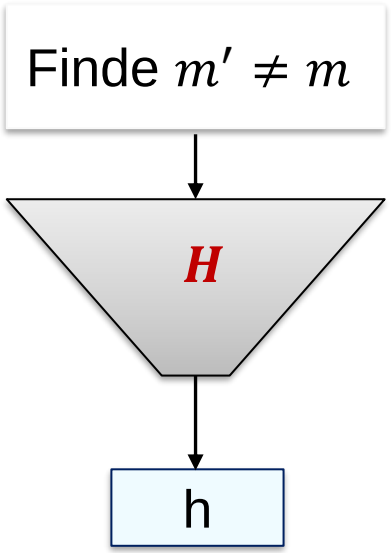
Gegeben h ist es schwer **m‘** zu finden, so dass **H(m‘) = h**

Second Preimage resistance



Gegeben **h**, ist es schwer m' $\neq m$ zu finden, so dass **H(m‘) = h**

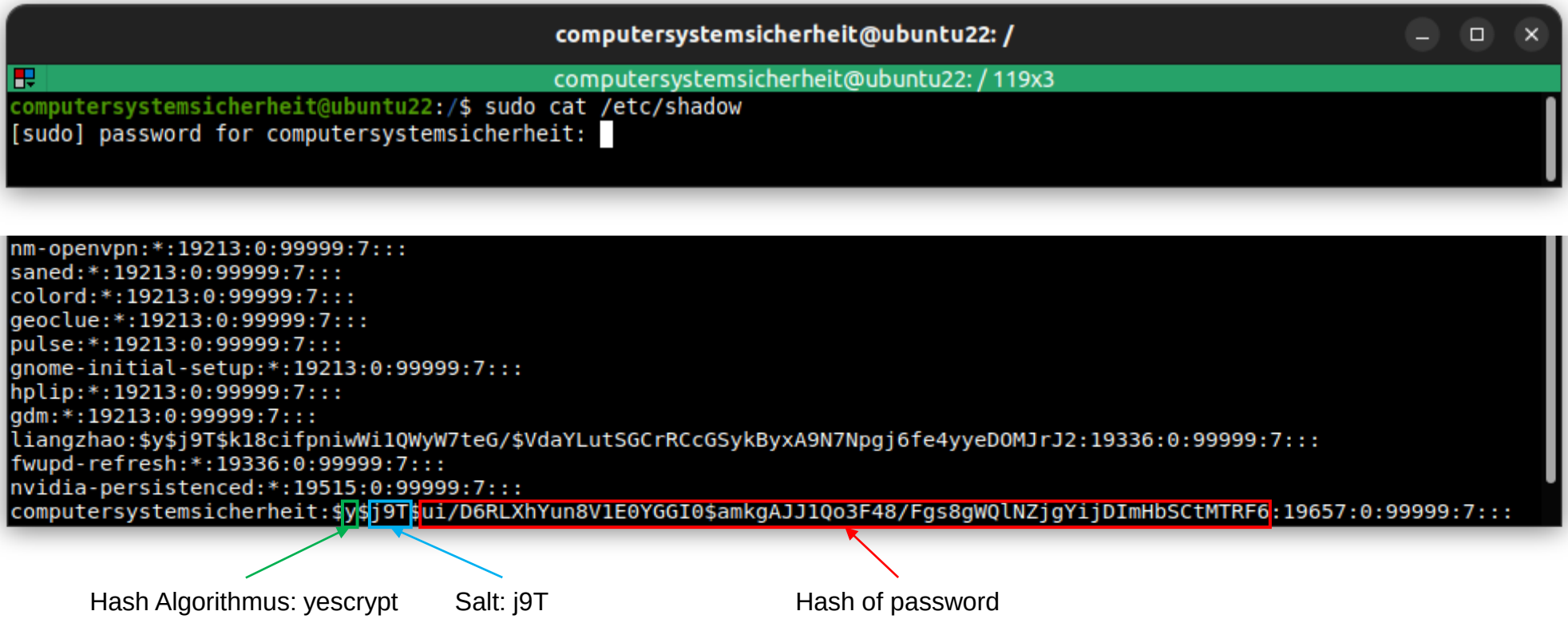
Collision resistance



Es gilt: $h := H(m') = H(m)$

BEISPIELANWENDUNG: PASSWORD HASHING

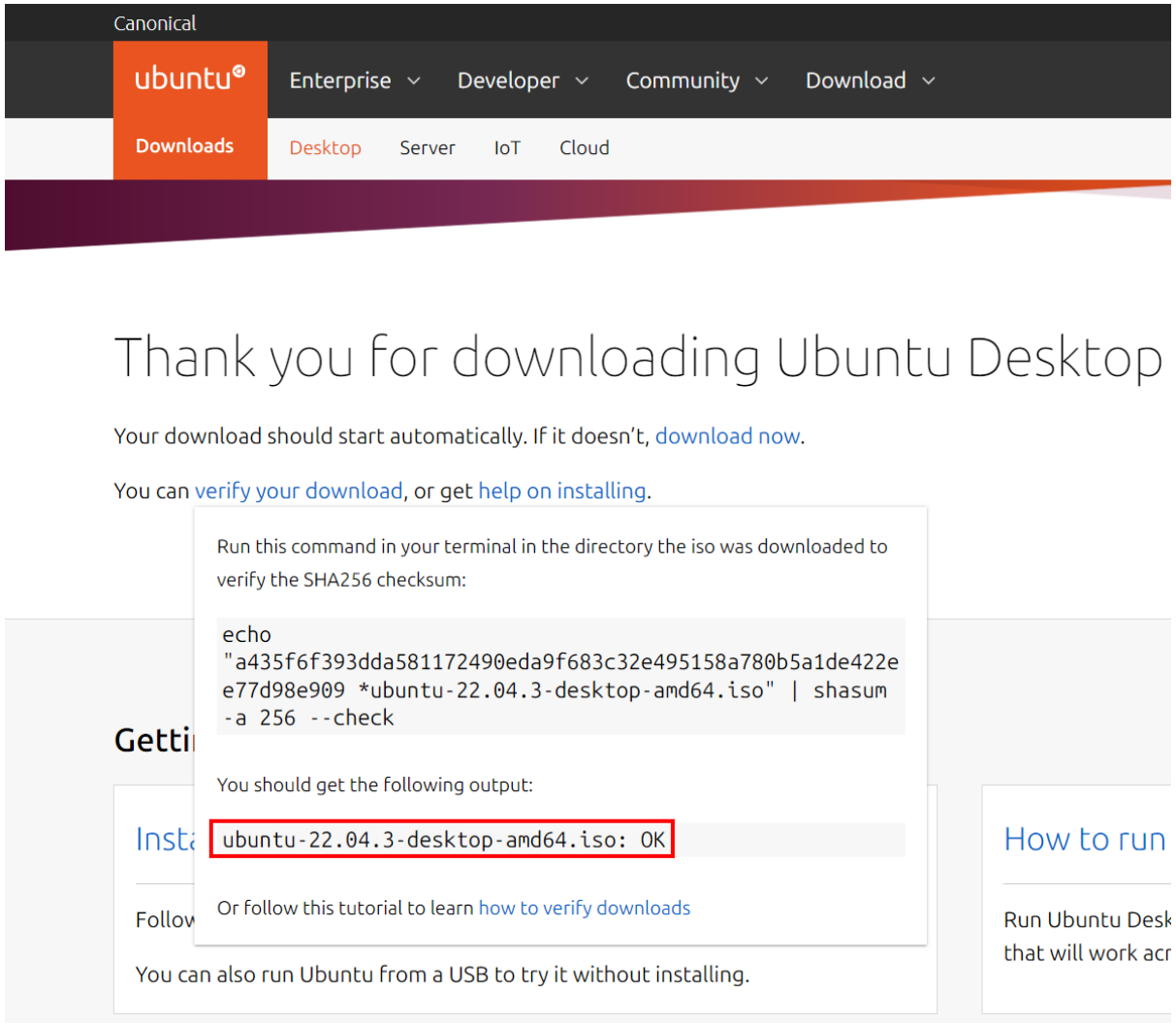
TECHNISCHE UNIVERSITÄT DARMSTADT

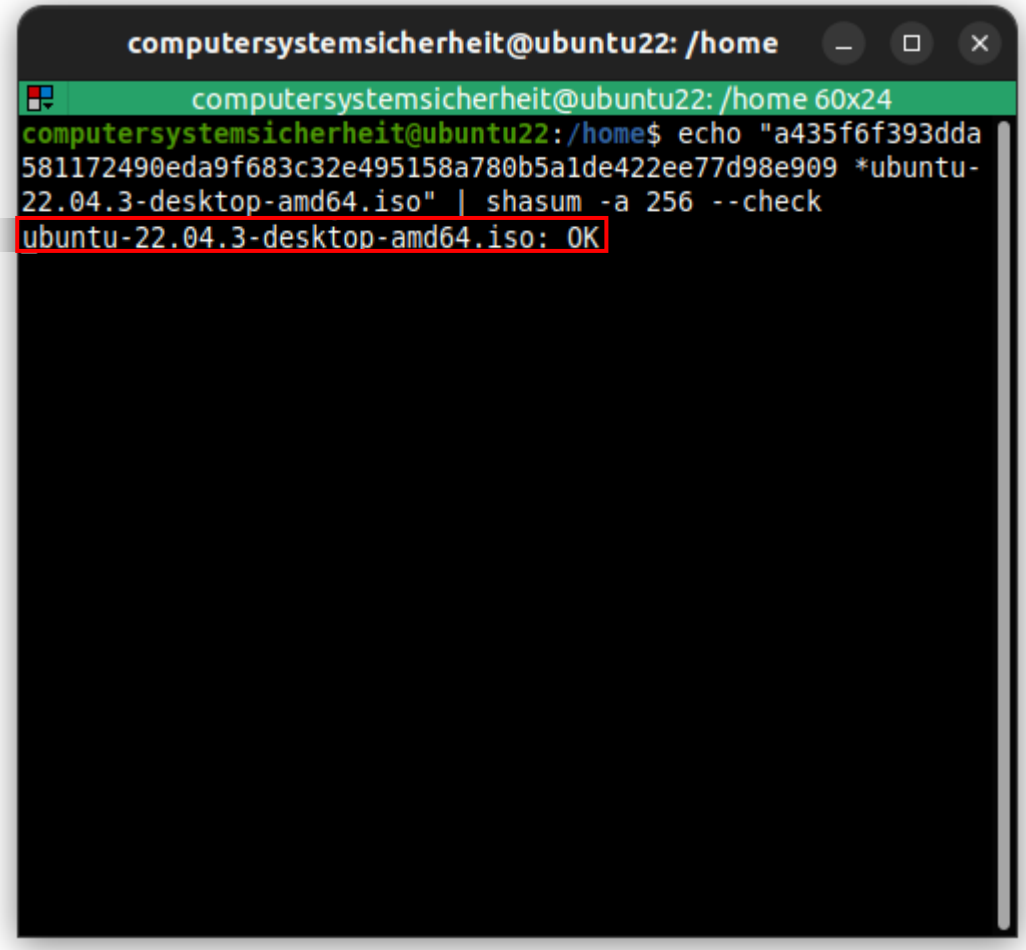


Später mehr zu Passwords und eingesetzten Hashfunktionen

BEISPIELANWENDUNG: DATENINTEGRITÄT

TECHNISCHE UNIVERSITÄT DARMSTADT





BEISPIELE FÜR HASHFUNKTIONEN

TECHNISCHE UNIVERSITÄT DARMSTADT

Name	|Ausgabe|	Kommentar
MD5	128 Bits	• Gilt als “gebrochen” (*) (Kollisionen) • Nicht mehr für neue Anwendungen geeignet
SHA-1	160 Bits	• Gilt als “gebrochen” seit 02/2017 (Kollisionen) • Nicht mehr für neue Anwendungen geeignet
SHA-256	256 Bits	Keine nichttrivialen Angriffe
SHA-3	224 Bits, 256 Bits, 384 Bits, 512 Bits	• Gewinner der SHA-3 Competition • Gute Wahl für neue Anwendungen
BLAKE3	256 Bits	• Effizienter als die obige Hash-Funktion • Hochgradig parallelisierbar

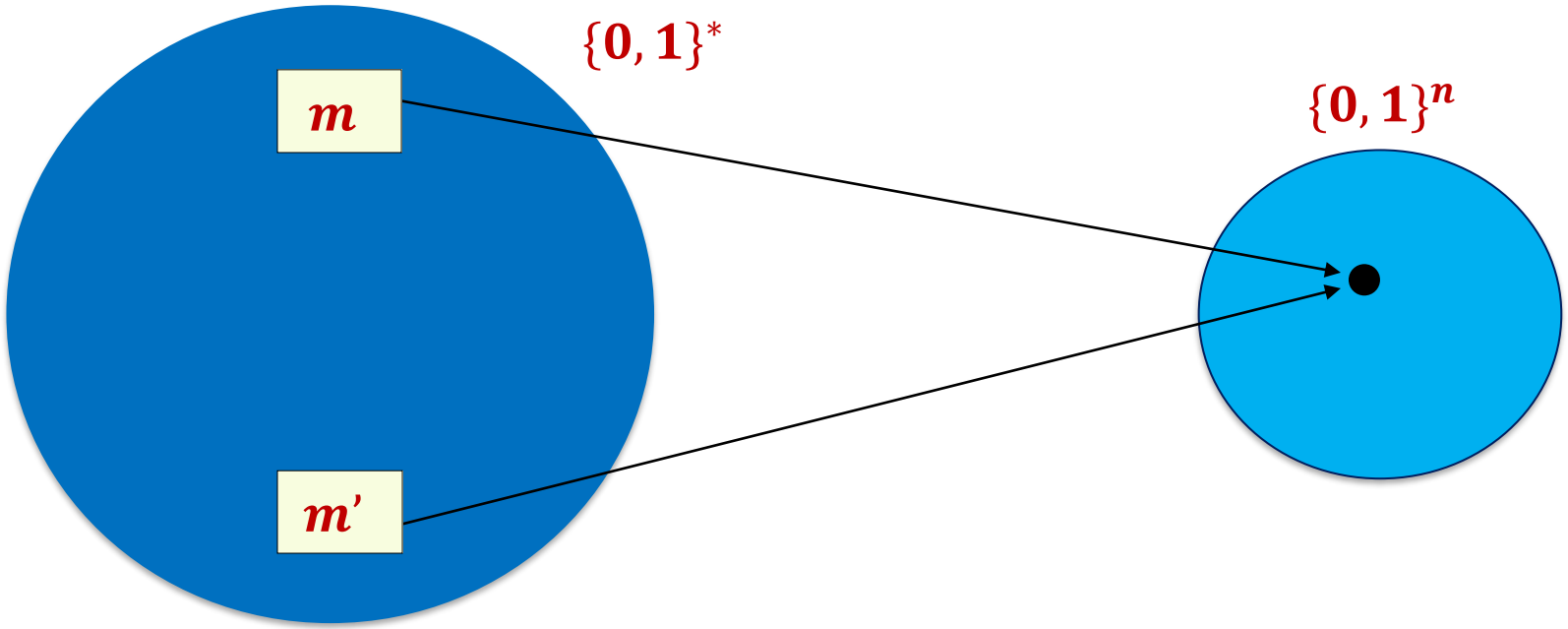
https://en.wikipedia.org/wiki/Hash_function_security_summary
https://github.com/BLAKE3-team/BLAKE3

HASHFUNKTIONEN: KOLLISIONEN

TECHNISCHE UNIVERSITÄT DARMSTADT

$H: \{0,1\}^* \rightarrow \{0,1\}^n$ bildet von großer Definitionsmenge auf kleinen Bildbereich ab

Problem: Definitionsmenge ist größer als der Bildbereich ➔ Kollisionen existieren immer



KOLLISIONSANGRIFFE

TECHNISCHE UNIVERSITÄT DARMSTADT

1. Generischer Brute-Force Angriff: Geburtstagsparadoxon
   - |H(x)| = n Bits
   - Nach ca. $\sqrt{2^n} = 2^{\frac{n}{2}}$: Wahrscheinlichkeit eine Kollision zu finden, ist $\geq \frac{1}{2}$

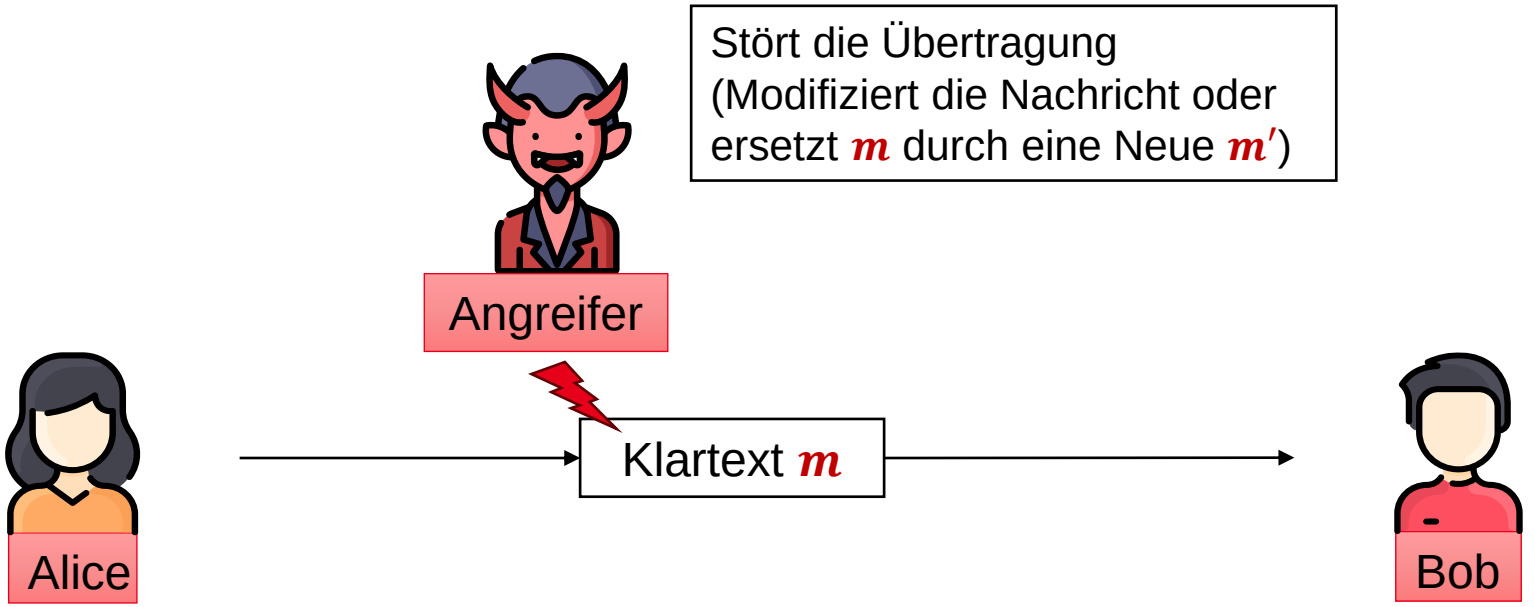
     ➔ Für 128-bit Sicherheit muss |H(x)| = 256 Bits

2. Kryptanalyse
   - MD5 Kollision in $2^{24.1}$ Schritten (Stevens 2009): wenige Sekunden auf modernem PC
   - SHA1 (Stevens et al 2017): Zwei kollidierende PDF Dokumente berechnet

MESSSAGE AUTHENTICATION CODES

MESSAGE AUTHENTICATION

- **Integrität?**



Sichere Kommunikation muss **Vertraulichkeit** **und** **Integrität** der Nachricht gleichzeitig garantieren

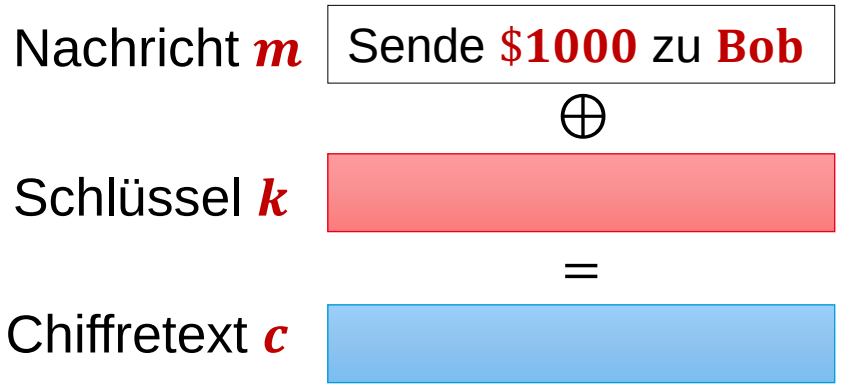
MESSAGE AUTHENTICATION

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Garantiert Verschlüsselung Integrität?**

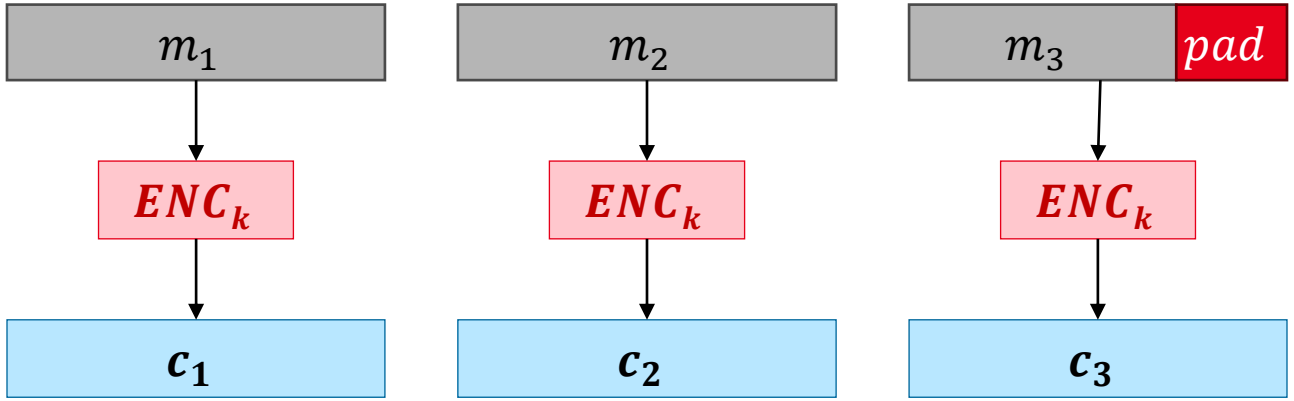
Im Allgemeinen ist diese Aussage nicht korrekt!

Beispiel: one-time pad.



Falls die **Angreifer** m und c kennt, kann sie k berechnen und einen Chiffretext für eine beliebige Nachricht erstellen.

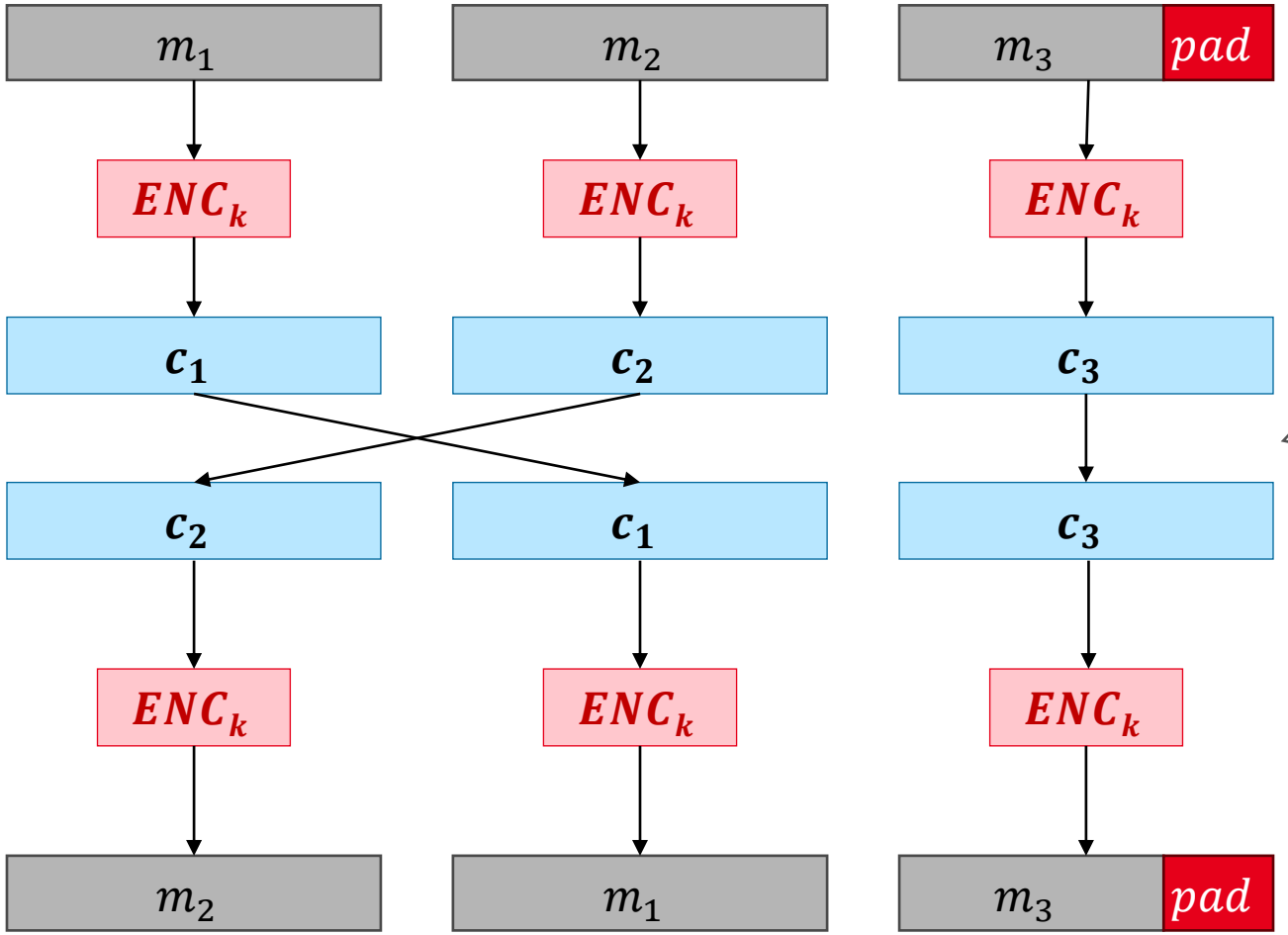
ERINNERUNG: ECB MODE VERSCHLÜSSELUNG



Kann ein Angreifer **sinnvolle** Änderungen am Klartext machen, **ohne k zu kennen**?

SCHWÄCHE DES ECB MODES

Angreifer



Angreifer kann sinnvolle Veränderungen am Klartext vornehmen

–

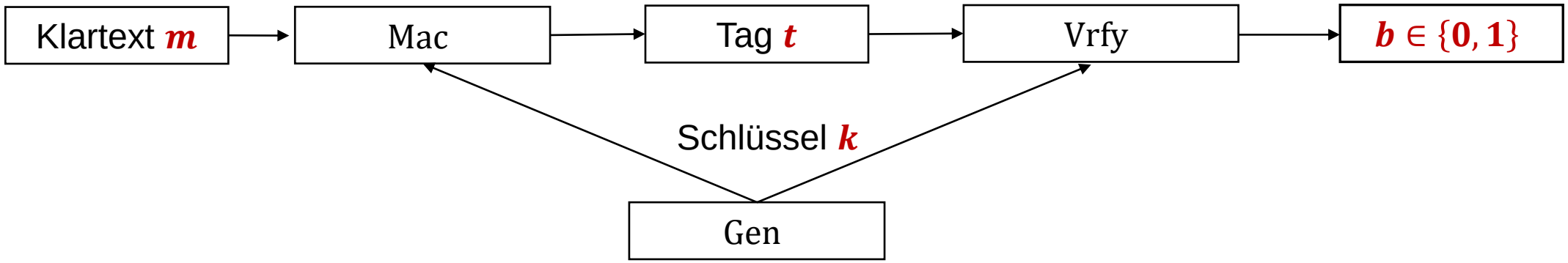
Empfänger kann das nicht erkennen

WAS WIRD BENÖTIGT?

TECHNISCHE UNIVERSITÄT DARMSTADT

Message Authentication Code (MACs)

- **Algorithmen:** $(\mathrm{Gen}, \mathrm{Mac}, \mathrm{Vrfy})$

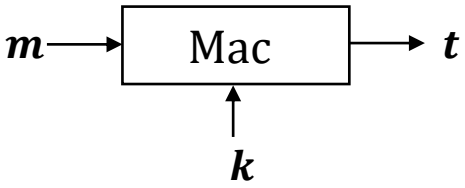


- **Korrektheit:** Die Verifizierung eines valider Tag t für m und k resultiert in $b = 1$

  ➔ $\mathrm{Vrfy}_k(m, \mathrm{Mac}_k(m)) = 1$ für alle Nachrichten m und Schlüssel $k \leftarrow \mathrm{Gen}$

- **Effizienz:** Authenfizierung und Verifizierung sind effizient (> 10GB/s)

SICHERHEITSDEFINITION

- Ziel des Angreifers: Was ist ein erfolgreicher Angriff?
  - Naive Option: Angreifer lernt den Schlüssel $\boldsymbol{k}$ nicht 

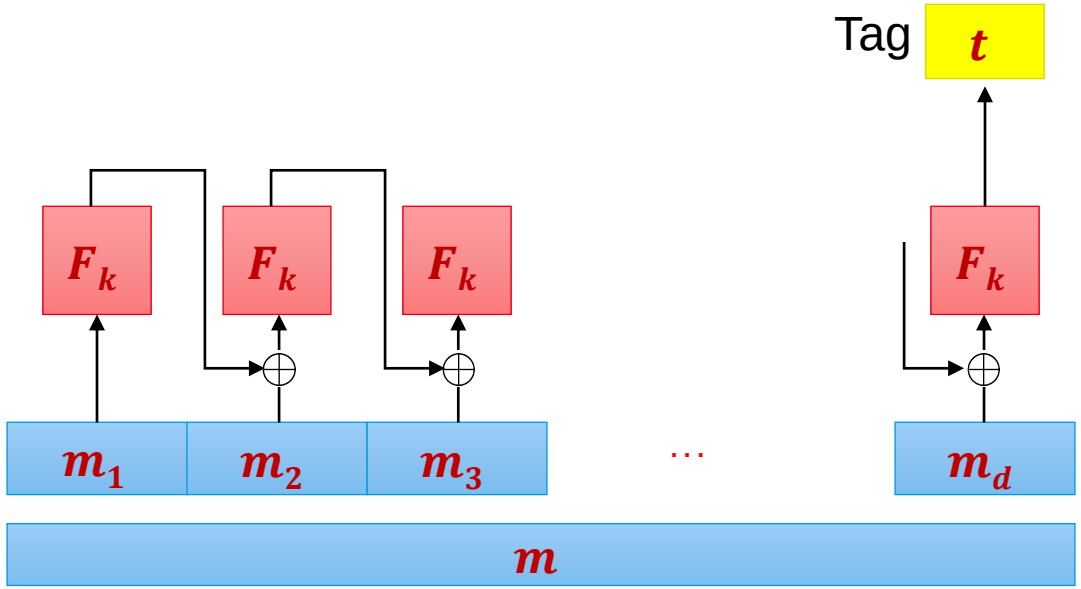
  - In der Kryptographie: Angreifer kann keine validen Tag $\boldsymbol{t}$ für $\boldsymbol{m}$ erzeugen

- Arte von MACs?
  - Informationstheoretisch sichere MACs ➔ **nicht effizient** für Authenfizierung von vielen Nachrichten
  - Komplexitätstheoretisch sichere MACs
    a. Für Nachrichten fixer Länge
    b. Für Nachrichten mit beliebiger Länge: z.B. **CBC-MAC** und **HMAC** (siehe nächste Slide)

CBC-MAC

Sei $F_K: \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$ eine Blockchiffre

$\mathbf{CBC-Mac}_k(m_1, \ldots, m_d)$ ist definiert wie folgt:



$\mathbf{Vrfy}_k(m, t)$: berechne $\tilde{t}$ und gib **1** aus falls $t = \tilde{t}$; ansonsten gib **0** aus

Wichtig: CBC-MAC ist nur sicher für Nachrichten der gleichen Länge

HASH-UND-MAC

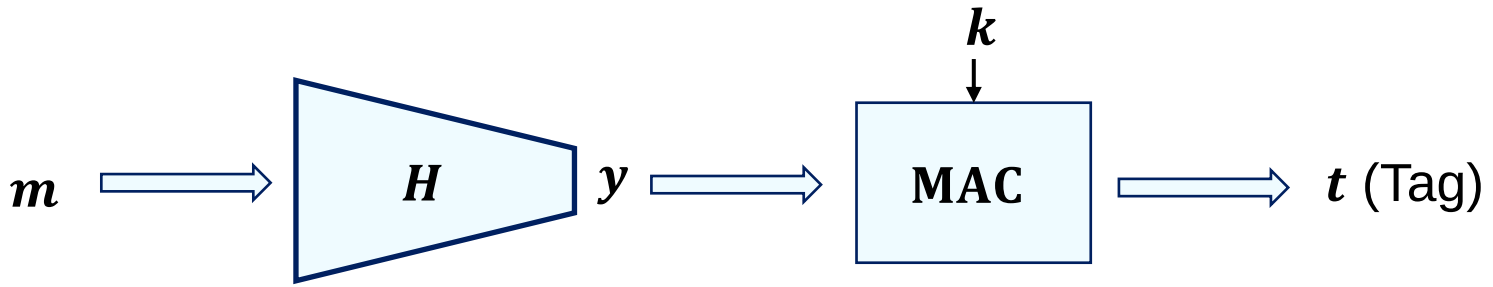
CBC-MAC ist nur für fixe Nachrichtenlänge

↓

Für beliebige Nachrichtenlänge?

Konstruktionsidee:

1. Berechne $\boldsymbol{y = H(m)}$ der langen Nachricht $\boldsymbol{m}$ mit Hilfe von Domain-Extension für Hashfunktionen
2. Berechne $\mathbf{MAC}_{\boldsymbol{k}}(\boldsymbol{y})$ mit Hilfe von MAC für fixe Nachrichtenlänge



Frage: Ist es möglich einen sicheren MAC **nur** basierend auf Hashfunktionen zu konstruieren?

HMAC

Trivialer Ansatz: $\mathbf{MAC}_k(y) = H(k||y)$

Ist diese Konstruktion sicher?
➔ Nur für bestimmte Hash Funktionen

- Sicher für manche Hash-Funktionen, bspw. SHA-3

- Unsicher für andere, bspw. SHA-256
  - Arbeitet mit fixen Eingabelängen und iterativem Hashing
  - Trivialer Ansatz ist anfällig für „length extension attacks“

Besser: Nutze HMAC (RFC 2104)

$$\mathbf{MAC}_k(y) = H(k \oplus op||H(k \oplus ip||m))$$

- op: Outer Padding
- op: Inner Padding

KRYPTOPRIMITIVEN

TECHNISCHE UNIVERSITÄT DARMSTADT

Folgende Kryptoprimitiven betrachten wir in der Vorlesung:

	Symmetrische Kryptoverfahren	**Asymmetrische Kryptoverfahren**
Vertraulichkeit	• One-Time Pad • Block-Chiffren & Modes of Operation	• RSA Verschlüsselung • ElGamal Verschlüsselung
Integrität & Authentizität	• MACs	• Digitale Signaturen

- Hash Funktionen
- Key Agreement
- Passwörter

Authentifizierte Verschlüsselung

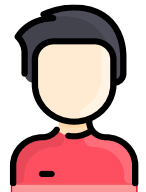
TECHNISCHE UNIVERSITÄT DARMSTADT

AUTHENTIFIZIERTE VERSCHLÜSSELUNG

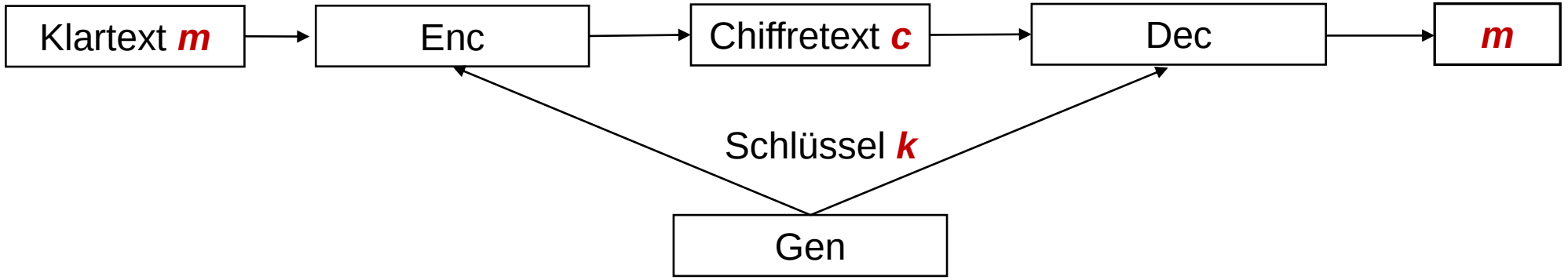
AUTHENTIFIZIERTE VERSCHLÜSSELUNG

TECHNISCHE UNIVERSITÄT DARMSTADT

Alice



Bob

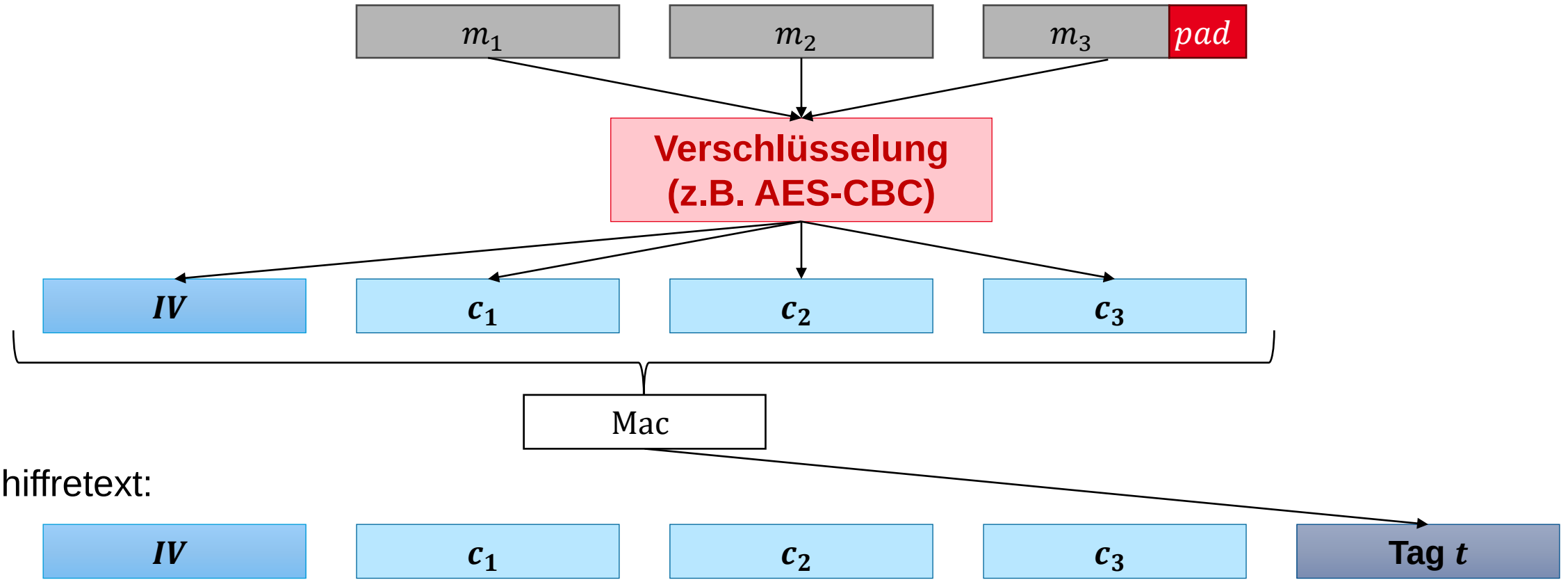


Angreifer

Angreifer sollte nicht …

- … „nützliche **Informationen**“ über den Klartext **lernen**
- … „irgendwelche“ **Änderungen** am Klartext **vornehmen** können

“ENCRYPT-THEN-MAC”



“MAC-THEN-ENCRYPT”?

TECHNISCHE UNIVERSITÄT DARMSTADT

Encrypt-then-MAC:

- Beweisbar sicher

MAC-then-Encrypt:

- Konkrete Angriffe bekannt
  - Lucky 13
  - BEAST
- Nutzen die Tatsache aus, dass auch dann entschlüsselt wird, falls Chiffretext verändert wurde
- Bei Encrypt-then-MAC: Nur authentische Chiffretexte werden entschlüsselt!

ÜBERSICHT BETRIEBSMODI

TECHNISCHE UNIVERSITÄT DARMSTADT

Nicht authentifiziert

- **Electronic Code Book (ECB)**
- **Cipher Block Chaining (CBC)**
- **Counter Mode (CTR)**
- Cipher Feedback Mode (CFB)
- Ouput Feedback Mode (OFB)

Authentifiziert

- Galois Counter Mode
- **Counter Mode with CBC-MAC (CCM)**

THEMENÜBERSICHT

TECHNISCHE UNIVERSITÄT DARMSTADT

Folgende Themen haben wir behandelt:

- Definition von symmetrischen Chiffren
- Sicherheitsbegriff für symmetrische Chiffren (IND-CPA)
- OTP Verschlüsselung
- Blockchiffren (DES, AES)
- Modes of Operation
- Hashfunktionen
- MACs
- Authentifizierte Verschlüsselung