



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

# SECURITY

## Integrität

May 31, 2024

Marc Stöttinger



Without integrity, encryption is meaningless.

Bruce Schneier

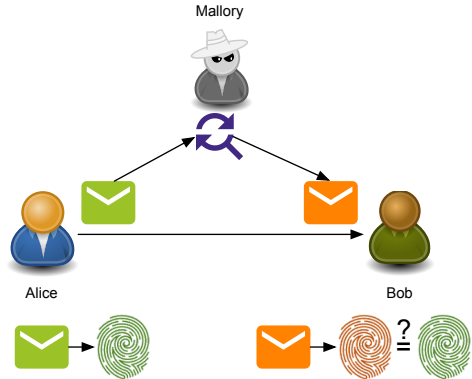
# MOTIVATION INTEGRITÄT

→ **Bedrohung:**

Mallory verändert die Nachricht

→ **Ziel:**

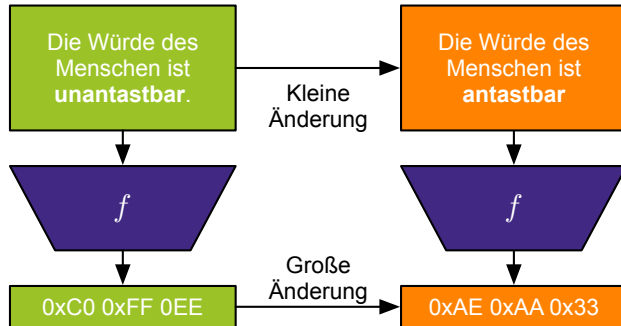
Eindeutiger Fingerabdruck mit dem unerlaubte Änderungen an der Nachricht erkannt werden können



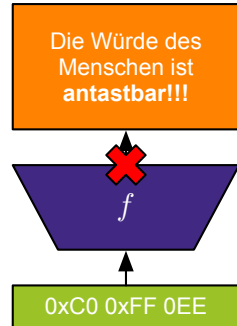
# INTEGRITÄT DURCH EINWEGFUNKTIONEN

Eine Einwegfunktion  $f$  bildet einen **beliebig langen** Wert auf einen **nicht-invertierbaren** Wert fixer Länge ab

## Änderungen in Eingaben von Einwegfunktionen



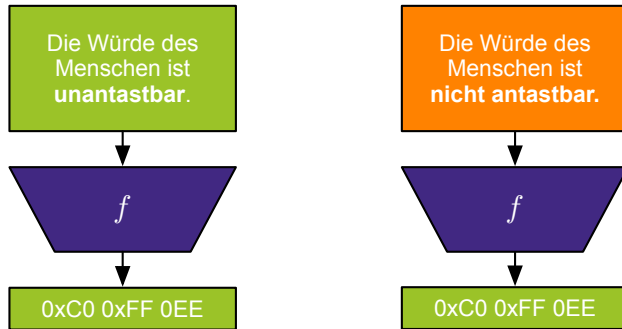
## Einwegeigenschaft



# KOLLISIONSRESISTENZ VON EINWEGFUNKTIONEN

Kollisionen nicht vermeidbar, da die Länge der Nachricht reduziert wird

→ Möglicher Angriff: Ausprobieren von Nachrichten bis eine Kollision gefunden wird



## Hashfunktionen

Hashfunktionen erweitern die Einwegfunktionen um Kollisionsresistenz

# EIGENSCHAFTEN VON HASHFUNKTIONEN $H$

Eine Hashfunktion  $H$  bildet eine **beliebig lange Nachricht  $m$**  auf einen **Hashwert (Digest) fixer Länge  $H(m)$**  und besitzt die folgenden Eigenschaften:

1. **Einwegeigenschaft:** (Preimage resistance)
  - Die Funktion  $H(m)$  muss effizient berechenbar sein
  - Es darf nicht möglich sein, die Funktion  $H$  zu invertieren, d.h. vom Hashwert auf ein Urbild  $m$  zu schließen
2. **Schwache Kollisionsresistenz:** (second pre-image resistance)
  - Es darf nicht möglich sein, zu  $m$  ein anderes  $m'$  zu finden mit  $m \neq m'$  und  $H(m) = H(m')$
3. **Starke Kollisionsresistenz:** (collision resistance)
  - Es darf nicht möglich sein, zwei beliebige  $m$  und  $m'$  zu finden mit  $m \neq m'$  und  $H(m) = H(m')$

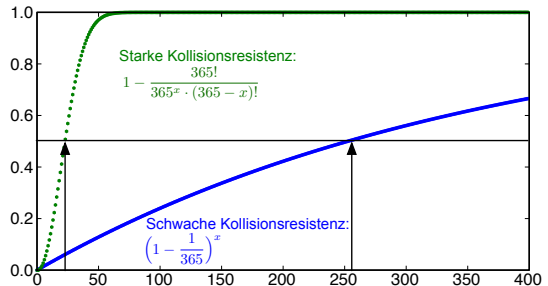
# SCHWACHE VS. STARKE KOLLISIONSRESISTENZ GEBURTSTAGSPARADOX (1/2)

- **Beispiel:** Geburtstag am gleichen Tag im Jahr
- **Schwache Kollisionsresistenz:**
  - Wie viele Personen müssen im Raum sein, damit mit  $\geq 50\%$  Wahrscheinlichkeit eine Person am gleichen Tag **wie Sie** Geburtstag hat?
- **Starke Kollisionsresistenz:**
  - Wie viele Personen müssen im Raum sein, damit mit  $\geq 50\%$  Wahrscheinlichkeit **zwei beliebige** Personen im Raum am gleichen Tag Geburtstag haben?

## SCHWACHE VS. STARKE KOLLISIONSRESISTENZ GEBURTSTAGSPARADOX (2/2)

→ Schwache Kollisionsresistenz  
(Bestimmter Tag): 253

→ Starke Kollisionsresistenz  
(Beliebiger Tag): 26



Quelle: <https://de.wikipedia.org/wiki/Geburtstagsparadoxon>



# WAHL DER HASHWERTLÄNGE

- Wie lang muss ein Hashwert sein, um starke Kollisionsresistenz zu besitzen?
- **Wurzel** als obere Schranke der starken Kollisionsresistenz
  - Bei Elementen mit  $x$ -bit Länge sind bei einer Menge von  $\sqrt{2^x} = 2^{\frac{x}{2}}$  zufällig gewählten Werten mit  $\leq 50\%$  Wahrscheinlichkeit zwei Werte gleich
  - **Beispiel Geburtstagsparadox**: 365 Tage im Jahr,  $\sqrt{365} = 19,10 \leq 26$
- Um  **$x$ -bit Berechnungssicherheit** zu erhalten, muss die **Ausgabelänge** einer **Hashfunktion  $2x$ -bit** sein:
  - 128-bit Sicherheit: 256-bit Ausgabelänge Hashfunktion
  - 256-bit Sicherheit: 512-bit Ausgabelänge Hashfunktion

# ÜBERSICHT VON HASH ALGORITHMEN

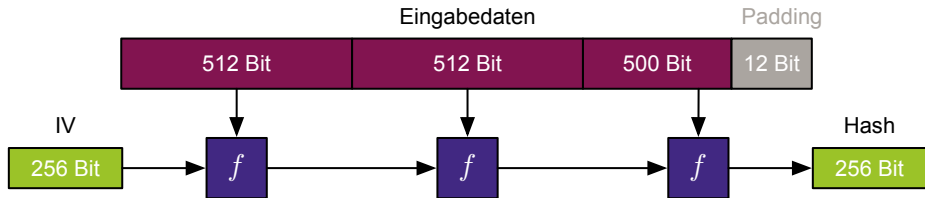
- Basierend auf den bisher diskutierten Anforderungen von Hash-Funktionen gibt es zwei generelle Typen von Hashfunktionen:
  - Dedizierte Hashfunktionen
  - Hashfunktionen basierend auf Blockchiffren
- Um von einer beliebigen Eingabelänge auf eine fixe Ausgabe zu kommen, haben sich folgend Konstruktionsprinzipien etabliert:
  - Merkle-Damgård (Kollisionaresistenz: Hälfte der Ausgabelänge)
  - Sponge (Kollisionaresistenz: Minimum (Hälfte der Ausgabelänge, Hälfte der Kapazität))
- Die Konstruktion einer Hashfunktion beruht auf der speziellen Kompressionsfunktion und einem der beiden Konstruktionsprinzipien

# HASHFUNKTIONEN UND SICHERHEIT

Hashfunktion	Digest Länge [Bits]	Sicherheit
MD5	128	Schwache Kollisionsresistenz theoretisch gebrochen ( $2^{123}$ [SA09]) Starke Kollisionsresistenz praktisch gebrochen (~35 Minuten Berechnung)
RIPEMD	128/160/256/320	RIPEMD-128 Starke Kollisionsresistenz praktisch gebrochen RIPEMD-160/256/320 Sicher (Angriffe gegen Versionen mit reduzierten Runden)
SHA1	160	Schwache Kollisionsresistenz theoretisch gebrochen ( $2^{159,3}$ [KK12]) Starke Kollisionsresistenz praktisch gebrochen (110 GPU Jahre Berechnung)
SHA2	224/256/384/512	Angriffe gegen Versionen mit reduzierten Runden
SHA3	224/256/384/512	Angriffe gegen Versionen mit reduzierten Runden

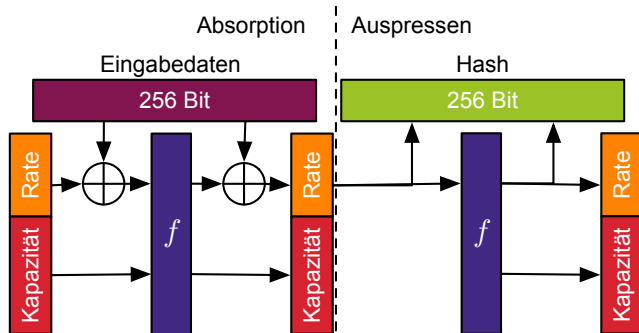
## STANDARD HASH ALGORITHM 2 (SHA2)

- Standardisiert vom US NIST im Jahr 2002 als Nachfolger des SHA-1
- Kommt in den Varianten SHA2-224, SHA2-256, SHA2-384 und SHA2-512 und basiert auf einer Merkle-Damgård Konstruktion
  - Zahl am Ende ist die Bitlänge des Hashwertes
  - SHA224/256 bzw. SHA384/512 nutzen die gleiche Funktion mit unterschiedlicher Ausgabelänge
- SHA2-256 nutzt eine Kompressionsfunktion  $f$ , die 512-Bit Eingabedatenblöcke mit einem 256-Bit internen Zwischenstand verarbeitet:



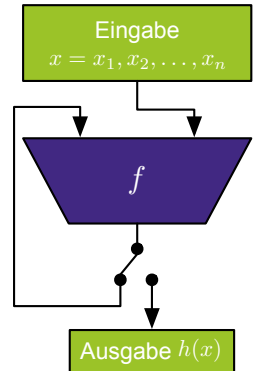
# STANDARD HASH ALGORITHM 3 (SHA3)

- Standardisiert vom US NIST im Jahr 2015 als Nachfolger des SHA2 und basiert auf einer Spong-Konstruktion
- Varianten SHA3-224, SHA3-256, SHA3-384, SHA3-512 und SHAKE128, SHAKE256
- mit der Variante SHAKE können beliebige Länge Digest generiert werden



# HASHFUNKTIONEN BASIEREND AUF BLOCKCHIFFREN BETRIEBSMODUS

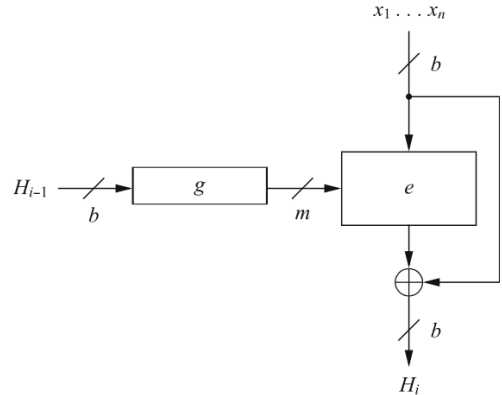
- Hashfunktionen, welche Blockchiffren als Permutationsfunktion benutzen, verwenden eine Merkle-Damgård-Konstruktion
- Die Kollisionsresistenz entspricht in den meisten Fällen der Hälfte der Blockgröße
- Im Fall von AES-128/192/256 ist die Kollisionsresistenz der Konstruktion immer 64-Bit wegen der Blockgröße.



## MATYAS-MEYER-OSEAS HASH

- Die Rückkopplung des vorherigen Hashwerts  $H_{i-1}$  geschieht über den Schlüsseleingang, die Länge muss über die Abbildungsfunktion  $g$  angepasst werden
- Funktionsvorschrift:  

$$H_i = \text{Enc}_{g(H_{i-1})}(x_i) \oplus x_i$$
- $b$  entspricht der Blockgröße der Chiffre und  $m$  der Schlüssellänge

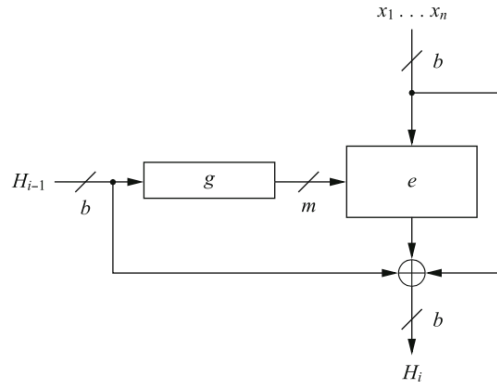


Quelle: Christoph Paar, Jan Pelzl: Kryptografie verständlich, 2016, Springer

# MIYAGUCHI-PRENEEL HASH

- Die Miyaguchi-Preneel Hash-Konstruktion verknüpft noch die Message mit Digest als Ergänzung zur Matyas-Meyer-Oseas Hash-Konstruktion
- Funktionsvorschrift:  

$$H_i = \text{Enc}_{g(H_{i-1})}(x_i) \oplus x_i \oplus H_{i-1}$$
- $b$  entspricht der Blockgröße der Chiffre und  $m$  der Schlüssellänge



Quelle: Christoph Paar, Jan Pelzl: Kryptografie verständlich, 2016, Springer



# DAVIS-MEYER HASH

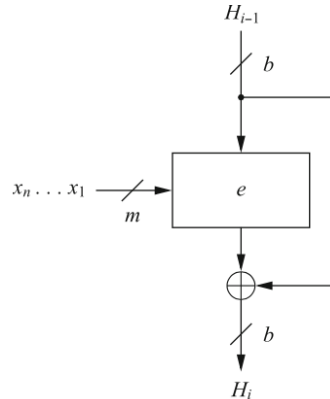
→ Die Davis-Meyer Hash-Konstruktion nutzt den Schlüsseleingang der Chiffre für die Texteingabe

→ Rückkopplung des vorherigen Digest über Plaintexteingang der Chiffre

→ Keine Abbildungsfunktion  $g$  wird benötigt.

→ Funktionsvorschrift:

$$H_i = \text{Enc}_{x_i}(H_{i-1}) \oplus H_{i-1}$$



Quelle: Christoph Paar, Jan Pelzl: Kryptografie verständlich, 2016, Springer

# ZUSAMMENFASSUNG

- Einsatzzwecke und Eigenschaften von kryptographischen Hashfunktionen
- Unterschied zwischen schwacher und starker Kollisionsresistenz
- Existierende kryptographische Hashfunktionen kennen (dedizierte und auf Blockchiffren basierende Hashfunktionen)
- Beurteilung ob eine Funktion die Eigenschaften der kryptographischer Hashfunktionen erfüllt