Fachhochschule
Dortmund
University of Applied Sciences and Arts

Informationssicherheit – SoSe 2023
Kryptographische Verfahren: Hashing & MAC

Prof. Dr. Holger Schmidt
holger.schmidt004[at]fh-dortmund.de

Fachhochschule Dortmund
Fachbereich Informatik
Professur für IT-Sicherheit, Informatik

Themen & Lernziele

Themen

- Grundlagen Hashing und MAC
- Grundlegende Angriffsarten und Kryptoanalyse
- Merkle-Damgård-Konstruktion
- SHA-2 Familie
- SHA-3 Familie

Lernziele

Die Studierenden sind in der Lage,

- Grundlagen der Themen Hashing und MAC zu differenzieren und zu erklären.
- SHA-2 und SHA-3 zu erklären und anzuwenden.

Kryptographie
Kryptographische Hashfunktionen

Dieser Abschnitt basiert auf Schmeh, 2016, Teil 2 – Kapitel 16. Aufgrund der Präsentation als Folien und Notizen sind die Texte der Quelle typischerweise paraphrasiert.

Kryptographische Hashfunktionen I

- Schutzziel: **Integrität**
- H ist die **Hashfunktion**, h der **Hashwert**, m das zu hashende Dokument
- Hashfunktion: $H(m) = h$
- Anwendung z. B. im Kontext digitaler Signaturen

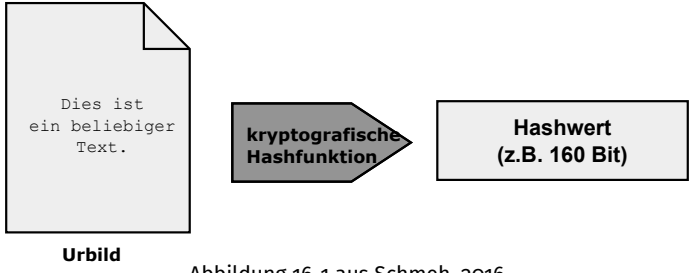


Abbildung 16-1 aus Schmeh, 2016

Kryptographische Hashfunktionen II

- **Nichtkryptographische Hashfunktionen**:
  - 4-stellige natürliche Zahl ergänzt um eine weitere Stelle als Hash
  - z. B. Quersumme der 4 Stellen als Hashwert
  - 10.000 Urbilder gegenüber 10 Hashwerten führt dazu, dass mehreren Urbildern der gleiche Hashwert zugeordnet wird (**Kollision**)
- **Kryptographische Hashfunktionen** sind so konstruiert, dass Angreifer **keine Kollisionen** verursachen können.

Kryptographische Hashfunktionen III

Entwurfsziele:

- Jeder Hashwert ist **gleichwahrscheinlich**.
- **Chaos**, d. h. kippt ein Bit des Klartextes, dann kippen durchschnittlich die Hälfte aller Bits des Hashwerts.
- **Surjektivität**, d. h. alle Hashwerte können erzeugt werden.
- **Effiziente** Berechenbarkeit
- **Kompression**

Kryptographische Hashfunktionen IV

Angriffe:

- **Urbild-Angriff**: Finde zu einem vorgegebenen Hashwert h_v eine Nachricht m die denselben Hashwert (d. h. $h_v = H(m)$) ergibt.
- **Zweites-Urbild-Angriff**: Finde zu einer vorgegebenen Nachricht m eine andere Nachricht m', sodass sich für beide Nachrichten der gleiche Hashwert (d. h. $H(m) = H(m')$) ergibt.
- **Kollisionsangriff**: Finde zwei beliebige verschiedene Nachrichten m, m' mit gleichem Hashwert (d. h. $H(m) = H(m')$).
- Hashfunktionen, die gegen Urbild-Angriffe resistent sind, sind auch gegen Zweites-Urbild-Angriffe resistent. Resistenz gegen Kollisionsangriffe ist dadurch jedoch nicht garantiert.

Kryptographie
Message Authentication Code (MAC)

Dieser Abschnitt basiert auf Schmeh, 2016, Teil 2 – Kapitel 18.1. Aufgrund der Präsentation als Folien und Notizen sind die Texte der Quelle typischerweise paraphrasiert.

Message Authentication Code (MAC)

- Schutzziele: **Integrität** und **Authentizität**
- **Schlüsselabhängige Hashfunktion**
- M ist die **MAC Funktion**, v der **MAC-Wert**, m das zu schützende Dokument, k der Schlüssel
- MAC Wert berechnen: $MAC(m, k) = v$
- Basierend auf Hashfunktion (**HMAC**) gem. ISO/IEC 9797-2/3 bzw. Chiffre (**CMAC**) gem. ISO/IEC 9797-1

Kryptographie
SHA-2 Familie

Dieser Abschnitt basiert auf Schmeh, 2016, Teil 2 – Kapitel 16.3. Aufgrund der Präsentation als Folien und Notizen sind die Texte der Quelle typischerweise paraphrasiert.

SHA (Secure Hash)-2 Familie

- **Kryptographische Hashfunktionen**
- Vom NIST im Jahr 2000 erstmals veröffentlicht (FIPS-180-4, 2015)
- Spezifiziert in RFC 6234 (https://www.rfc-editor.org/rfc/rfc6234, aufgerufen am 28. Juni 2023)
- Große **praktische Bedeutung: TLS, IPSec, PGP, S/MIME**
- **Urbilder** $< 2^{64}$ **Bit**: SHA-224 m. Hashwertlänge 224 Bit, SHA-256 m. Hashwertlänge 256 Bit
- **Urbilder** $< 2^{128}$ **Bit**: SHA-384 m. Hashwertlänge 384 Bit, SHA-512 m. Hashwertlänge 512 Bit, SHA-512/224 m. Hashwertlänge 224 Bit, SHA-512/256 m. Hashwertlänge 256 Bit

SHA-2 ist Merkle-Damgård Konstruktion

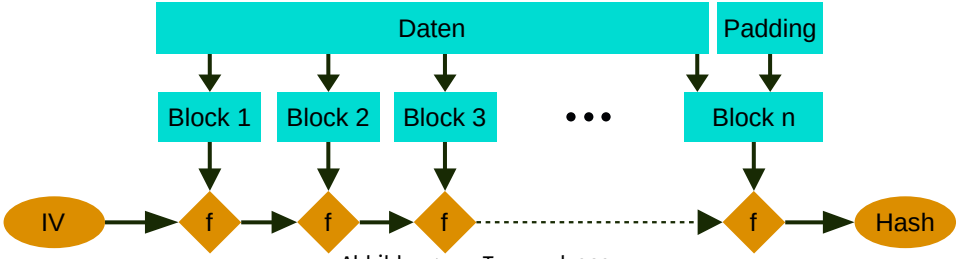
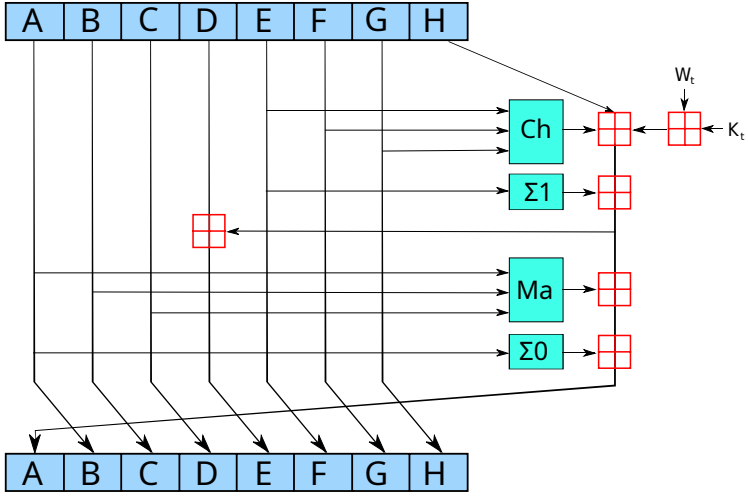


Abbildung aus Tremmel, 2021

- SHA-224, SHA-256: 512 Bit Blocklänge, interne Verarbeitung von 32 Bit Wörtern
- SHA-384, SHA-512 Varianten: 1024 Bit Blocklänge, interne Verarbeitung von 64 Bit Wörtern

SHA-224, SHA-256 Rundenfunktion



SHA-224, SHA-256 Rundenfunktion, https://en.wikipedia.org/wiki/SHA-2, aufgerufen am 28. Juni 2023, lizensiert unter CC BY 3.0[1]

[1] https://creativecommons.org/licenses/by-sa/3.0/

SHA-224, SHA-256 Rundenfunktion: Operationen und Konstanten

Fachhochschule
Dortmund
University of Applied Sciences and Arts

- $Ch(E, F, G) = (E \wedge F) \boxplus_{32} (\neg E \wedge G)$
- $Ma(A, B, C) = (A \wedge B) \boxplus_{32} (A \wedge C) \boxplus_{32} (B \wedge C)$
- $\Sigma_0(A) = (A >>> 2) \boxplus_{32} (A >>> 13) \boxplus_{32} (A >>> 22)$
- $\Sigma_1(E) = (E >>> 6) \boxplus_{32} (E >>> 11) \boxplus_{32} (E >>> 25)$

SHA-2 – Kryptoanalyse

- SHA-2 Familie nicht anfällig für SHA-1[2] Geburtstagsangriff
- Urbild-Angriffe (Khovratovich, Rechberger & Savelieva, 2012) und Kollisionsangriffe (Lamberger & Mendel, 2011) für Runden-reduzierte Varianten
- **Length extension attacks** (Duong & Rizzo, 2009)

[2] https://tools.ietf.org/html/rfc3174, aufgerufen am 28. Juni 2023

Kryptographie
SHA-3 Familie

Dieser Abschnitt basiert auf Schmeh, 2016, Teil 2 – Kapitel 17.3. Aufgrund der Präsentation als Folien und Notizen sind die Texte der Quelle typischerweise paraphrasiert.

SHA-3 (Keccak)-Algorithmus

- **Kryptographische Hashfunktion**
- Ende der 2010er von Guido Bertoni, Joan Daemen, Michaël Peeters und Gilles Van Assche im Rahmen eines NIST-Wettbewerbs entwickelt
- Vom NIST standardisiert (FIPS-202, 2015)
- Keccak/SHA-3-Familie mit 224, 256, 384, 512 Bits
- Noch selten in der Praxis eingesetzt

SHA-3 (Keccak)-Algorithmus – Status I

Status von Keccak ist $(5 \times 5 \times 64)$-Bits Array

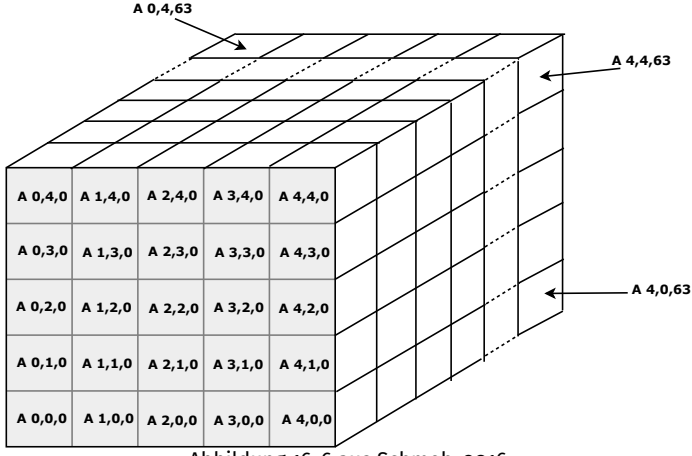


Abbildung 16-6 aus Schmeh, 2016

SHA-3 (Keccak)-Algorithmus – Status II

- Statusgröße von 1600 Bits
- Senkrechter 5-Bit Block heißt **Spalte**, ein waagerechter **Reihe**.
- Ein nach hinten gehender 64-Bits Block heißt **Wort**.
- Die Differenz zwischen Status und Blocklänge heißt **Kapazität**.
- Je größer die Kapazität, desto (mutmaßlich) sicherer ist Keccak.
- Die Länge des Hashwerts beträgt die Hälfte der Kapazität.
- Soll der Hashwert 224 Bits haben, dann muss die Kapazität 448 Bits betragen.
- Daraus folgt die Blocklänge von 1152 Bits, d. h. das Urbild wird in Blöcke dieser Länge zerlegt (ggfs. mit Padding).

SHA-3 (Keccak)-Algorithmus – Keccak-Schwamm

Fachhochschule Dortmund
University of Applied Sciences and Arts

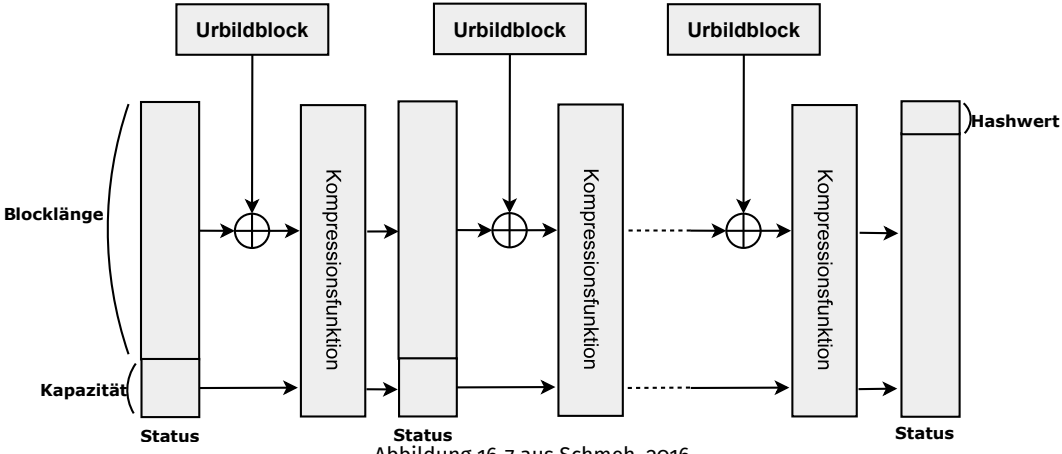


Abbildung 16-7 aus Schmeh, 2016

SHA-3 (Keccak)-Algorithmus – Kompressionsfunktion

- **Kompressionsfunktion** mit 24 Runden mit jeweils 5 Teilrunden:
  - Θ Berechnung von Paritätsbits der Spalten und XOR-Verknüpfung mit benachbarten Spalten.
  - ρ Rotation der Wörter des Status.
  - π Permutation der Wörter des Status.
  - χ XOR-Verknüpfung der Bits einer Reihe.
  - ι XOR-Verknüpfung des Status mit einer Konstante (welche sich in jeder Runde ändert)
- Runden 1-3 und 5 linear (Diffusion), Runde 4 nicht linear (Konfusion)

Fallstricke und Anwendungshinweise

- SHA-1 (und erst recht nicht noch ältere Verfahren wie z. B. MD5, spezifiziert in RFC 1321[3]) nicht verwenden
- SHA-2 bzw. SHA-3 224er Variante nicht verwenden, siehe BSI TR-02102-1[4]

[3] https://www.rfc-editor.org/rfc/rfc1321, 28. Juni 2023

[4] https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102.pdf, 28. Juni 2023

Zusammenfassung

Zusammenfassung

- Grundlagen Hashing und MAC definiert
- SHA-2 und SHA-3 vorgestellt

Weiterführende Literatur

Weiterführende Literatur

- *Kryptografie – Verfahren - Protokolle - Infrastrukturen*, Kapitel 16, 17.3 und 18.1 von Schmeh (2016)
- *IT-Sicherheit – Konzepte - Verfahren - Protokolle*, Kapitel 8.1 von Eckert (2023)

Literatur I

- Duong, T., & Rizzo, J. (2009, September). Flickr's API Signature Forgery Vulnerability. https://archive.org/details/pdfy-HaRR7XMfTOb_RrUo (siehe S. 17).
- Eckert, C. (2023). *IT-Sicherheit: Konzepte - Verfahren - Protokolle* (11. Aufl.). De Gruyter Oldenbourg. (Siehe S. 28).
- FIPS-180-4. (2015). Federal Information Processing Standards Publication (FIPS 180-4). Secure Hash Standard (SHS). https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf (siehe S. 13).
- FIPS-202. (2015). Federal Information Processing Standards Publication (FIPS 202). SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf (siehe S. 19).

Literatur II

- Khovratovich, D., Rechberger, C., & Savelieva, A. (2012).Bicliques for Preimages: Attacks on Skein-512 and the SHA-2 Family. *FSE*, *7549*, 244–263. https://doi.org/10.1007/978-3-642-34047-5_15 (siehe S. 17).
- Lamberger, M., & Mendel, F. (2011). Higher-Order Differential Attack on Reduced SHA-256. https://eprint.iacr.org/2011/037 (siehe S. 17).
- Schmeh, K. (2016). *Kryptografie – Verfahren - Protokolle - Infrastrukturen* (6. Aufl.). dpunkt.verlag. (Siehe S. 5, 6, 10, 12, 18, 20, 22, 28).
- Tremmel, S. (2021).Deterministisches Chaos. *c't Magazin für Computer und Technik*, (7), 64–67 (siehe S. 14).