Kryptographische Hash-Funktionen

C77A | 90409E65 | 60C056544D48F82949

Terve - God Dag - Hej

Teil I

(Dietmar Bremser)

Agenda

- Einführung und Motivation
- Theoretische Betrachtungen
  - Definition und Anforderung einer kryptographischen Hash-Funktion
  - Exkurs: Statistik und das „Geburtstagsparadox“
  - Verifikation der Anforderungen
- Iterated Hash Functions
  - Merkle-Damgård
  - SHA1 und MD5

Definition?

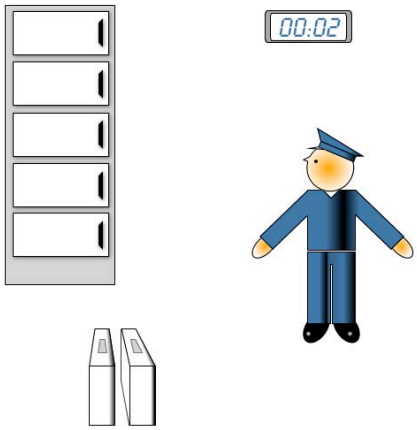
- **„Hashfunktion“** := „Einwege-Hash-Algorith-mus: eine mathematische Formel, die einen Textblock in einen einmaligen Block verschlüsselter Daten fester Länge umwandelt“
  [„PKI e-security implementieren“, S. 546, RSA-Press/MITP, 2001]

- **„Hash-Verfahren“** := „ist ein Speicherungs- und Suchverfahren, bei dem die Adressen von Datensätzen aus den zugehörigen Schlüsseln errechnet werden“
  [Duden der Informatik, S. 278, 2001]

Nutzen?

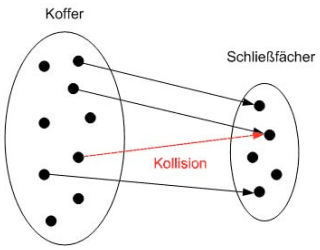
- Was hat das mit Kryptographie zu tun?
  - Verschlüsselung schützt in einem gewissen Sinn vor der Manipulation der Botschaft auf dem Übertragungsweg
  - Hash-Funktionen ermöglichen *zusätzlich* die Verifikation möglicher Manipulationen auf dem Übertragungsweg, sofern das „Kondensat“ der Funktion an einem separaten, „sicheren“ Ort gelagert wird

Beispiel



- Funktion?
  (Quersumme der Uhrzeit) mod 5

Das Problem



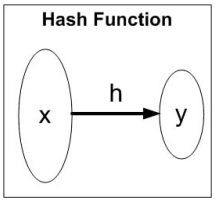
- **Wie bildet** man **ohne Kollision *n* Elemente auf *m*** ab, wenn ***n* potentiell unendlich** und ***m* beschränkt** ist?
- Oder anders: **wie bildet** man **ohne Kollision** potentiell **unendlich lange Zeichenketten auf Zeichenketten mit fixierter Länge** ab?

Definition! (1)

- Eine Hash-Funktion *H* ist $H : \Sigma^* \rightarrow \Sigma^n$, mit
  - einem Alphabet Σ
  - Worten über dem Alphabet
    unendlich viele variierender Länge:
    $$\Sigma^* = \{x_1 \dots x_n, x_i \in \Sigma, n \geq 0, n << \infty\} = \bigcup_{n \geq 0} \Sigma^n$$
    endlich viele fixer Länge: $\Sigma^n = \{x_1 \dots x_n \ : x_i \in \Sigma\}$
- Sei nun
  - *h* := *H*, eine kryptographische Hash-Funktion
    $$h : X \rightarrow Y$$
  - X := Σ^*, die Menge aller möglichen Botschaften
  - Y := Σ^n, die Menge aller möglichen Funktionsergebnisse

Definition! (2)

- Dann ist
  - $x \in X$ eine „message“
  - $y \in Y$ ein „finger print“
    bzw. „message digest“
  - (x, y) ein gültiges Paar, sofern $y = h(x)$



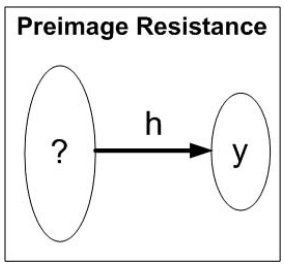
- Funktion *h* ist ‚many-to-one‘ (n:1) und damit nicht injektiv, hat Eigenschaft der
  - Kompression
  - einfachen Berechenbarkeit

Anforderungen

- Hash-Funktionen unterliegen den folgenden Anforderungen:
  1. „preimage resistance“
  2. „2nd preimage resistance“
  3. „collision resistance“
- Achtung: wir merken bald, was an der Definition der RSA-Labs nicht stimmt!

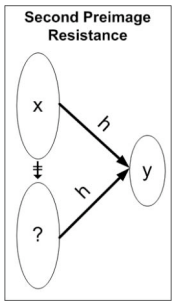
„preimage resistance“

- auch „Einwegfunktion“, „Urbildresistenz“
- gegeben:
  - Bild $y \in Y$ („message digest“)
  - Hash-Funktion *h*



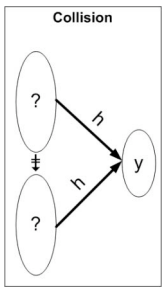
- gesucht: Urbild $x \in X$ („message itself“)
- **Anforderung**: **Urbild** soll **nicht auffindbar** bzw. kalkulierbar sein, d.h. die Funktion ist nicht „(berechenbar) umkehrbar“

„2nd preimage resistance“



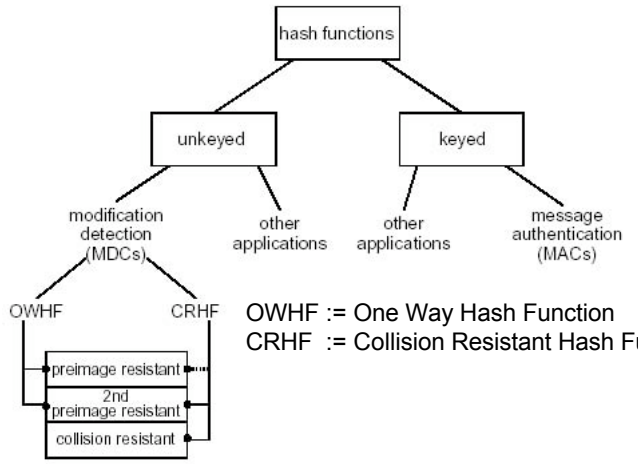
- „Resistenz gegen Zweites Urbild“, „schwache Kollisionsresistenz“
- gegeben:
  - Urbild $x \in X$ („message itself“)
  - Hash-Funktion *h*, damit auch $y \in Y$ („message digest“)
- gesucht: zweites Urbild $x' \in X$
- **Anforderung**: **kein zweites Urbild** $x' \in X$ mit gleichem Bild $y = h(x)$ darf **auffindbar** sein, also $x' \neq x$ mit $h(x') = h(x)$ soll nicht auffindbar sein

„collision resistance“



- „starke Kollisionsresistenz“
- gegeben: Hash-Funktion *h*
- gesucht: zwei Urbilder $x, x' \in X$ („messages itself“)
- **Anforderung**: **keine zwei Urbilder** $x, x' \in X$ mit gleichem Bild $y \in Y$ sollen auffindbar sein, also $x' \neq x$ mit $h(x') = h(x)$ soll nicht auffindbar sein

Klassifizierung



OWHF := One Way Hash Function
CRHF := Collision Resistant Hash Function

„unkeyed hash-function“

- benötigt nur einen „Eingabeparameter“, die Botschaft
  - Empfänger einer Botschaft und eines korrespondierenden „message digest“ muss verwendete Hash-Funktion bekannt sein
  - „message digest“ kann nicht über einen unsicheren Kanal übertragen werden, sondern muss an einem sicheren Ort separiert sein

„keyed hash-function“ (1)

- Formal:
  - ist ein Tupel (X, Y, K, H), auch „Hash-Familie“ mit
    - X, Y wie gewohnt
    - K als einer endlichen Schlüsselmenge
    - jeder Schlüssel $k \in K$ indiziert eine bestimmte Hash-Funktion $h_k \in H$
    - die Menge der verfügbaren Hash-Funktionen H ist ebenso endlich wie die Menge aller Schlüssel K
- Abbildungsvorschrift: $h_k : X \to Y$

„keyed hash-function“ (2)

- benötigt zwei „Eingabeparameter“, nämlich eine Botschaft und einen Schlüssel
  - Schlüssel bestimmt die zu verwendende Hash-Funktion eindeutig
  - Empfänger einer Botschaft und eines korrespondierenden „message digest“ muss nur der verwendeten Schlüssel bekannt sein
  - „message digest“ kann über einen unsicheren Kanal übertragen werden, da der Schlüssel geheim ist

Exkurs: Statistik (1)

- Wissen:
  - eine Menge von Ereignissen Ω
  - Wahrscheinlichkeit des Eintreten eines Ereignisses ω bei Gleichverteilung
    $$P(\omega) = \frac{1}{n} \qquad n = |\Omega|,\ \omega \in \Omega$$
  - Berechnung des gegenteiligen Ereignisses
    $P(\overline{\omega}) = 1 - P(\omega)$ oder $P(\overline{\omega}) = 1 - \frac{1}{n}$
  - Berechnung zweier unabhängiger Ereignisse ω_1, ω_2 :
    $P(\omega_1 \cap \omega_2) = P(\omega_1) \cdot P(\omega_2)$ folglich tritt bei *r* Versuchen

    Ereignis überhaupt nicht ein: $P(\overline{\omega}) = \left(1 - \frac{1}{n}\right)^r$

    Ereignis mindestens einmal ein: $P(\omega) = 1 - \left(1 - \frac{1}{n}\right)^r$

Exkurs: Statistik (2)

- Wieviele Versuche für eine bestimmte (fixe) Wahrscheinlichkeit?
  - Abschätzung: $P(\omega) \geq 1 - \left(1 - \frac{1}{n}\right)^r \Rightarrow \left(1 - \frac{1}{n}\right)^r \geq 1 - P(\omega)$
  - Logarithmus Naturalis ist Umkehrfunktion der Euler-Funktion
    $$r \cdot \ln\left(1 - \frac{1}{n}\right) \geq \ln\left(1 - P(\omega)\right) \Rightarrow r \geq \frac{\ln\left(1 - P(\omega)\right)}{\ln\left(1 - \frac{1}{n}\right)}$$
- Anwendung: „richtiges Geburtsdatum erraten“
  - bei r=**55** Leuten (Versuchen) beträgt die Wahrscheinlichkeit eines Treffers **14%**
  - für eine **50%**-ige Wahrscheinlichkeit eines Treffers bedarf es 253 Leute (Versuche)

Exkurs: Geburtstagsproblem (1)

- Frage: mit welcher Wahrscheinlichkeit *W* haben mindenstens zwei Menschen in einem Raum das gleiche Geburtsdatum?
  - Versuche sind unabhängig
  - Zwei Ereignisse, von denen eines eine Wiederholung des anderen ist (*Kollision*!)
  - Herleitung
    - 1. Ereignis keine Wiederholung: $W = \frac{N}{N} = 1$
    - 2. Ereignis keine Wiederholung: $W = \frac{N-1}{N}$
    - 3. Ereignis keine Wiederholung: $W = \frac{N-2}{N}$ (...)
    - *r*-tes Ereignis keine Wiederholung: $W = \frac{N-r+1}{N}$

Exkurs: Geburtstagsproblem (2)

- Schlussfolgerung:
  - die Wahrscheinlichkeit, eine Kollision unter r unabhängigen Ereignissen zu finden, ist gegeben mit $C(N,r) = 1 - P(N,r)$ wobei

$$P(N,r) = \frac{N \cdot (N-1) \cdot \ldots \cdot (N-r+1)}{N^r} = \left(1 - \frac{1}{N}\right) \cdot \ldots \cdot \left(1 - \frac{(r-1)}{N}\right) = \prod_{i=1}^{r-1} \left(1 - \frac{i}{N}\right)$$

- Anwendung: ein Alphabet $\Sigma := \{0,1\}$
  - Kollisionswahrscheinlichkeit für zehn 32-Bit-Zeichenketten $\approx 1.04774 \cdot 10^{-8}$
  - Kollisionswahrscheinlichkeit für 40 Acht-Bit-Zeichenketten ≈ 0.96

Exkurs: Geburtstagsproblem (3)

- Frage: wieviele Versuche für eine bestimmte (fixe) Wahrscheinlichkeit?
  - Gesetz: $1 - x \leq e^{-x},\ x \in \mathbb{R}$
  - Einsetzen: $\prod_{i=1}^{r-1}\left(1-\frac{i}{N}\right) \leq \prod_{i=1}^{r-1} e^{\frac{-i}{N}}$
  - Potenzgesetze: $\prod_{i=1}^{r-1} e^{\frac{-i}{N}} = e^{-\sum_{i=1}^{r-1}\frac{i}{N}} \Rightarrow \prod_{i=1}^{r-1}\left(1-\frac{i}{N}\right) \leq e^{-\sum_{i=1}^{r-1}\frac{i}{N}}$
  - Summenformel: $\sum_{k=1}^{n} k = \frac{n(n+1)}{2} \Rightarrow -\sum_{i=1}^{r-1}\frac{i}{N} = -\frac{r(r-1)}{2N}$
  - es folgt: $P(N,r) = \prod_{i=1}^{r-1}\left(1-\frac{i}{N}\right) \leq e^{-\frac{r(r-1)}{2N}}$

    $\Rightarrow 1 - e^{-\frac{r(r-1)}{2N}} \leq 1 - P(N,r)$
  - die erste Abschätzung: $C(N,r) \geq 1 - e^{-\frac{r(r-1)}{2N}}$

Exkurs: Geburtstagsproblem (4)

- Logarithmus Naturalis als Umkehrfunktion der Euler-Funktion $C(N,r) \geq 1 - e^{-\frac{r(r-1)}{2N}} \Rightarrow 1 - C(N,r) \leq e^{-\frac{r(r-1)}{2N}}$

  $$\Rightarrow \ln(1 - C(N,r)) \leq -\frac{r(r-1)}{2N} \cdot \underbrace{\ln e}_{=1}$$
- Rechenregel $-\ln x = \ln\left(\frac{1}{x}\right)$ nutzen, so dass

  $$r^2 - r \leq 2N \ln\left(\frac{1}{1 - C(N,r)}\right)$$
- weglassen des Terms *-r* und Fixieren der Wahrscheinlichkeit *C(N,r)* = 0.5 ergibt

  $$r \leq \sqrt{2N \ln\left(\frac{1}{1-0,5}\right)} = 1,18 \cdot \sqrt{N}$$

  fertig!

Exkurs: Geburtstagsproblem (5)

- *„the trick is to keep breathing“*
- Anwendung für Kollisionen:
  - Achtung: 50%-ige Wahrscheinlichkeit!
  - es bedarf nur 23 Leuten, um zwei gleiche Geburtstage zu erraten (nachrechnen)
  - es bedarf $1.237.320 \approx 1,18 \cdot \sqrt{2^{40}}$ Zeichenketten, die 40 Bit lang sind, um zwei gleiche Zeichenketten zu finden
  - aber: die Wahrscheinlichkeit, nur eine 40-Bit-Zeichenkette von insgesamt 1.237.320 zu erraten, beträgt $P(\varepsilon) = 1 - \left(1 - \frac{1}{2^{40}}\right)^{1.237.320} = 1,125 \cdot 10^{-6}$

„Random Oracle Model“

- mathematisches Modell einer idealen Hash-Funktion
  - die Hash-Funktion wird **zufällig** aus einer Menge $F^{X,Y}$ durch das Orakel ausgewählt
  - $F^{X,Y}$ kennzeichnet die Menge aller Funktionen, die von *X* auf *Y* abbilden.
  - verfügt über ein Gedächtnis
    - für jede Botschaft *x*, für die schon einmal ein „finger print“ *y* errechnet wurde, wird immer der gleiche „finger print“ zurückgegeben

Verifikation der Anforderungen

- Konstruktion eines „Las Vegas“- bzw. (ϵ, q) -Algorithmus, um
  - die drei Anforderungen
    1. „preimage resistance“
    2. „2nd preimage resistance“
    3. „collision“
  - bezüglich einer gegebenen Treffer-Wahrscheinlichkeit ϵ und einer
  - Menge an zufällig ausgewählten Eingaben *q* zu verifizieren
  - stellt auf „average case success probability“ ab

„preimage resistance“

Algorithm 4.1: FINDPREIMAGE(h, y, q)

choose any $\mathcal{X}_0 \subseteq \mathcal{X}, |\mathcal{X}_0| = q$
for each $x \in \mathcal{X}_0$
do { **if** $h(x) = y$ **then return** (x)
return (failure)

- prüfe, ob für eine zufällig gewählte Menge an **Botschaften der Anzahl** *q*, einem **gegebenen „message digest“** sowie einer **Hash-Funktion** ein Urbild gefunden werden kann
- Treffer-Wahrscheinlichkeit:

$$P(\varepsilon) = 1 - \left(1 - \tfrac{1}{n}\right)^q, \quad q = |X_0|, n = |Y|, \varepsilon = E_1 \vee E_2 \vee \ldots \vee E_q$$

„2nd preimage resistance“

Algorithm 4.2: FINDSECONDPREIMAGE(h, x, q)

$y \leftarrow h(x)$
choose $\mathcal{X}_0 \subseteq \mathcal{X} \backslash \{x\}, |\mathcal{X}_0| = q - 1$
for each $x_0 \in \mathcal{X}_0$
  do { **if** $h(x_0) = y$
    then return (x_0)
return (failure)

- prüfe, ob für eine zufällig gewählte Menge an **Botschaften der Anzahl** *q*, einer **gegebenen Botschaft** und einer **Hash-Funktion** ein **zweites Urbild** gefunden werden kann
- Treffer-Wahrscheinlichkeit:

$$P(\varepsilon) = 1 - \left(1 - \frac{1}{n}\right)^{q-1}, n = |Y|, \; X_0 \subseteq X \setminus \{x\} \Rightarrow |X_0| \leq |X| - 1 = q - 1$$

„collision resistance“

Algorithm 4.3: FINDCOLLISION(h, q)

choose $\mathcal{X}_0 \subseteq \mathcal{X} \backslash \{x\}, |\mathcal{X}_0| = q$
for each $x \in \mathcal{X}_0$
  do $y_x \leftarrow h(x)$
if $y_x = y_{x'}$ for some $x' \neq x$
  then return (x, x')
  else return (failure)

- prüfe, ob für eine zufällig gewählte Menge an **Botschaften der Anzahl** *q* und einer gegebenen **Hash-Funktion zwei identische Urbilder** gefunden werden können
- Treffer-Wahrscheinlichkeit:

$$C(N, r) = P(\varepsilon) = 1 - \prod_{i=1}^{q-1} \left(1 - \frac{i}{N}\right)$$

Reduktion

- Reduktion der drei vorgenannten Anforderungen untereinander
  - „collision resistance“ impliziert „2nd preimage resistance“:
    wird von Stinson und A. Menezes, P. van Oorschot, S. Vanstone bejaht
  - „collision resistance“ impliziert „preimage resistance“:
    wird von Stinson teilweise bejaht, von A. Menezes, P. van Oorschot, S. Vanstone verneint!

Reduktion (1)

- „collision resistance“ impliziert „2nd preimage resistance“
  - klar, weil
    - die Auffindbarkeit zweier gegebener, aber unbekannter Urbilder bei gegebener Hash-Funktion wird auf eine Situation reduziert, dass ein Urbild gegeben ist und ein zweites gesucht wird
    - Wahrscheinlichkeit für Kollision (Geburtstagsproblem) ist höher als die eines Treffers und „collision resistance bedeutet eine Wahrscheinlichkeit von 0
  - die Suche muss entweder über eine sehr große Menge oder über mehreren kleinen Mengen ausgeführt werden

Reduktion (2)

- „collision resistance“ impliziert keine „preimage resistance“
  - Stinsons „Fehler“: „Berechenbarkeit“?
  - Beweis durch Widerspruch (A. Menezes, P. van Oorschot, S. Vanstone)

$$h(x) = \begin{cases} 1 \circ x & \text{wenn } x \text{ hat die Bitlänge } n \\ 0 \circ g(x) & \text{sonst} \end{cases}$$

    - » ist „collision resistant“ und „2nd preimage resistant“
  - ferner impliziert „2nd preimage resistance“ keine „preimage resistance“

Zwischenstand

- **Angriffe** auf Hash-Funktionen sind **„brute force“**
- **Wahrscheinlichkeit des Erfolges** eines Angriffes ist **nicht auszuschließen**, weshalb auch keine „echten Einwege-Funktionen“ existieren
- **Sicherheit** ist bis dato nur durch **Kostenbetrachtung** (hoher Aufwand, geringer Ertrag) gegeben, indem die **Möglichkeit** einer Kollision **außerhalb der** gegenwärtigen Grenzen von **Berechenbarkeit** zu suchen ist

Literatur und Referenzen

1. A. Menezes, P. van Oorschot, S. Vanstone, *Handbook of Applied Cryptography*, 1996 (http://www.cacr.math.uwaterloo.ca/hac/)
2. Andrew Nash, William Duane, Celia Joseph, Derek Brink, *PKI - esecurity implementieren*. mitp-Verlag, RSA-Press, 2002.
3. Carlton R. Davis, *IPSec - Tunneling im Internet*, mitp-Verlag, RSAPress, 2002.
4. Jerry Luitwieler, *Kryptographie*, 1998. (http://home.t-online.de/home/jerry.luitwieler/krypto5.htm).
5. Klaus Pommerening, Marita Sergl, *Kryptographische Basistechniken*, Vorlesung Datenschutz und Datensicherheit, 1999. (http://www.staff.uni-mainz.de/pommeren/DSVorlesung/KryptoBasis/Hash.html)
6. Klaus Pommerening, Marita Sergl, *Das Geburtstagsphänomen*, Vorlesung Kryptologie, 2004. (http://www.staff.uni-mainz.de/pommeren/Kryptologie/Klassisch/2_Polyalph/GebPhaen.pdf)
7. Stinson, *Cryptography - Theory and Practice*, Chapman & Hall/CRC, 2002.
8. Wolfgang M. Ruppert, *Kryptographische Hashfunktionen*, Kryptographie-Vorlesung an der Universität Erlangen, 2000. Skript: (http://www.mi.uni-erlangen.de/~ruppert/WS0001/).
9. Neal Stephenson, *Cryptonomicon*, Wilhelm Goldmann Verlag, München, 2001.