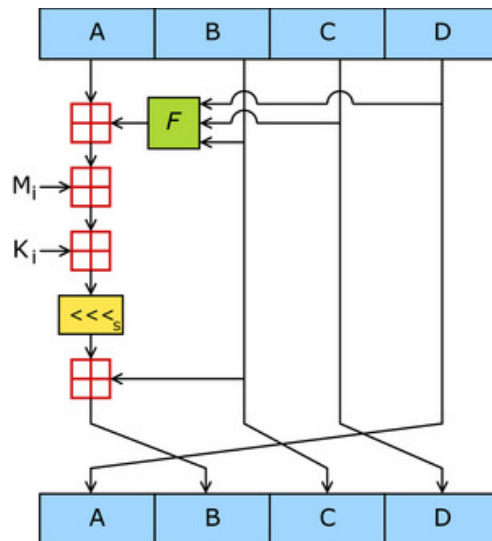




Handout Kryptographie

Hashing vs. blinde Signatur



Luca Costa
HTW Klasse TETv05

Ausgabedatum: 20.02.2008

Abgabedatum: 25.06.2008

Inhaltsverzeichnis

1 Hashing	3
1.1 Was bedeutet Hashing	3
1.2 Beispiele mit der md5-Hash-Funktion	3
1.3 Kriterien für eine gute Hash-Funktion	4
1.4 Vorteile und Nachteile	4
2 Blinde Signatur	5
2.1 Was ist eine blinde Signatur.....	5
2.2 Beispiel mit konkreten Werten.....	5
3 Vergleich	8
4 Quellenverzeichnis	9
4.1 Quellen aus dem Internet.....	9
4.2 Quellen aus den Büchern.....	9
5 Anhang	9
5.1 Abbildungen.....	9
5.2 Tabellen.....	9
6 Literatur	9

1 Hashing

1.1 Was bedeutet Hashing

Hashing, oder auch Hash-Funktion genannt, ist ein Algorithmus, der grosse Quellmengen (wie z. B. Zeichenketten) in kürzere Datenmengen umwandelt, so dass eine kürzere Zeichenkette entsteht. Diese Kette heisst Hashcode. Der Vorteil ist, dass solche Hashcodes nahezu eindeutig sind, also jede beliebige Quellmenge hat einen eindeutigen Hashcode. Wichtig ist auch, dass die Hash-Funktionen Einwegs sind. Man kann nicht aus einem Hashcode die ursprüngliche Quellmenge rekonstruieren, weil beim Hashing Informationen verloren gehen.

Mit Hashcodes werden normalerweise Binärdokumente gekennzeichnet, um die Authentizität beim Empfänger prüfbar zu machen. Eine andere Einsatzmöglichkeit ist das Aufbewahren von wichtigen Informationen, wie z. B. Passwörter. In viele Online-Foren werden die Benutzerdaten mit einer Hash-Funktion konvertiert und dann in der Datenbank gespeichert, somit kann man diese persönlichen Informationen vor Hacker-Attacken besser schützen.

Um die Funktionsweise einer Hashing-Funktion besser zu verstehen gibt es in nachfolgendem Kapitel ein praktisches Beispiel.

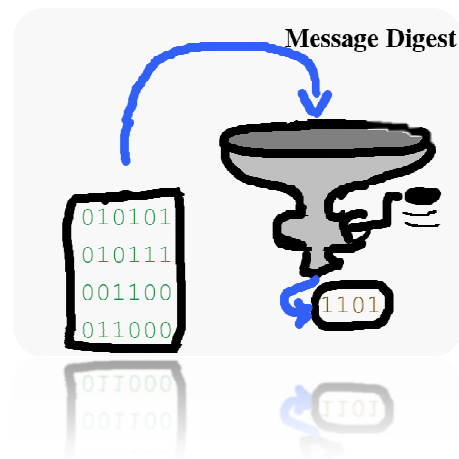


Abbildung 1: Prinzip einer Hash-Funktion

1.2 Beispiele mit der md5-Hash-Funktion

Eine sehr verbreitete Hash-Funktion im Internet ist md5 (Message-Digest-Algorithm 5). Wie schon der Algorithmus-Name sagt, MD5 ist die 5. Version und wurde von Ronald L. Rivest im Jahr 1991 entwickelt.

Nehmen wir an, dass wir die Nachricht „HTW Chur“ mit md5 berechnen möchten. Der Quelltext der PHP-Script sieht so aus:

```
$hashcode = md5 („HTW Chur“);
echo $hashcode; // Ausgabe = „e070ed1d56a4552733e72afe85058b19“
```

Also, der eindeutige md5-Hashcode von „HTW Chur“ ist „e070ed1d56a4552733e72afe85058b19“. Der erzeugte Hashwert ist immer 128 Bits lang, das heisst, dass auch einen langen Satz in 128 Bits umgewandelt wird, zum Beispiel:

```
$hashcode = md5 („To be, or not to be: that is the question“);
echo $hashcode; // Ausgabe = „eaf606c87569b2f97e230e792049833e“
```

Die Funktionsweise dieser Algorithmus ist eigentlich nicht so kompliziert. Der 128 Bits lange Hashwert ist in 4 Blöcke unterteilt, somit hat jeder Block eine Länge von 32 Bits. Solche Blöcke

werden dann mit verschiedenen vorbestimmten Konstanten initialisiert und mit einer Modulo-2-Funktion zusammengezählt. Dieses Verfahren wird 16 Mal durchgeführt. Als Ergebnis wird ein 128-Bit-Wert geliefert, die MD5-Summe. Genauere Informationen über den md5-Algorithmus sind auf Wikipedia [1] zu finden.

1.3 Kriterien für eine gute Hash-Funktion

Es gibt prinzipiell 4 Kriterien, um eine Hash-Funktion zu beurteilen:

- Datenreduktion: der Speicherbedarf des Hash-Wertes soll deutlich kleiner sein als die ursprüngliche Nachricht.
- Chaotisch, der Hash-Wert soll möglichst chaotisch sein, also man sollte keine Struktur im Hash-Wert finden.
- Surjektiv [2], alle Ergebnisse sollen möglich sein, also jede Nachricht muss einen eindeutigen Hash-Wert haben.
- Effizienz, die Hash-Funktion soll möglichst schnell arbeiten und in kurze Zeit einen Hash-Wert liefern.

Ein anderer wichtiger Aspekt, der auch eine grosse Rolle an die Qualität eines Hash Algorithmus spielt, sind die Kollisionen oder auch „sicheres Hashing“ genannt.

1.4 Vorteile und Nachteile

Ein wichtiger Vorteil ist, dass man mit md5 Dateien indizieren kann. Zum Beispiel bei den neuen Betriebssystemen, wie Mac OS X und Windows Vista, wurde auch eine solche Dateiindizierung eingeführt. Alle relevanten Dateien (wie z. B. Dokumente) werden mit ihrer Hash-Funktion in einer Datenbank indiziert, somit kann man sie schneller suchen und finden.

Ein Nachteil könnte sein, dass alle Quellinformationen nicht einen eindeutigen Hashcode haben, somit tritt man auf den sogenannten Kollisionen auf. Eine Lösung wäre die Hashcodelänge zu vergrössern, somit hätte man mehrere Kombinationen und sehr wenige Kollisionen.

2 Blinde Signatur

2.1 Was ist eine blinde Signatur

Mit blinder Signatur (engl. „blind signature“) geht es darum, ein Dokument zu unterschreiben, ohne den Inhalt des Dokuments zu sehen. Die Unterschrift auf das unterschriebene Dokument muss natürlich gültig sein. Auf den ersten Blick könnte eine blinde Signatur selten gebraucht werden, aber in der Praxis gibt es verschiedene Anwendungen, die eine solche Signatur benutzen, wie z.B. die elektronische Wahl.

Nehmen wir an, dass der nächste US-Präsident mit einer elektronischen Wahl gewählt wird, so können mehrere Millionen Bürger ein Formular im Internet ausfüllen und sofort senden. Sie können dann gleichzeitig die aktuellen Resultate der Wahl im Fernsehen nachschauen. Elektronische Wahlen haben zum Vorteil, dass die Wahlergebnisse schnell und in der Regel zuverlässig vorhergesagt werden können. Ein riesiger Nachteil ist, dass ein Bürger mehrmals abstimmen könnte und somit würden die Wahlergebnisse verfälscht. Es gibt aber eine Lösung; sie heisst blinde Signatur. Die elektronische Abstimmung muss von einer wahlberechtigten Person geschickt werden und sie muss von der zuständigen Angestellten validiert werden (Die Validierung könnte auch von einem elektronischen System durchgeführt werden). Das Problem ist, dass der Angestellte (oder elektronisches System), die die Abstimmung validieren muss, den Inhalt nicht sehen darf, weil es ja anonym ist. Somit muss man die Abstimmung mit einer blinden Signatur validieren.

Um die Funktionsweise besser zu verstehen gibt es in nachfolgendem Kapitel ein praktisches Beispiel.

2.2 Beispiel mit konkreten Werten

Zuerst muss man zwei Primzahlen wählen $p = 11$ und $q = 13$. Mit p und q kann man den Modul n berechnen, und zwar $n = pq = 11 \cdot 13 = 143$.

Als nächstes berechnet man $\varphi(n) = (p - 1)(q - 1) = \varphi(91) = (11 - 1)(13 - 1) = 120$.

Anschliessend wird eine zu $\varphi(n)$ teilerfremde Zahl e gewählt, die den öffentlichen Schlüssel bildet. In diesem Beispiel ist $e = 7$.

Jetzt muss man den privaten Schlüssel d berechnen, diesen kann man mit Hilfe der Vielfachsummandarstellung berechnen, und zwar $ggT(p, q) = ggT(11, 13) = 1$. Mit Hilfe der diophantische Gleichung, der grösste gemeinsame Teiler vom Modul n und von dem öffentlichen Schlüssel, kann man den privaten Schlüssel d berechnen:

$$ggT(\varphi(n), e) = ggT(120, 7) = u \cdot 120 + v \cdot 7 = 1.$$

$$\begin{aligned} 120u + 7v &= 1 \\ 7v &= 1 - 120u \\ v &= \frac{1 - 120u}{7} = \frac{1 - 119u - u}{7} = -17u + \frac{1 - u}{5} \end{aligned}$$

$$\begin{aligned} 1 - u &= 7k_1 \\ -u &= 7k_1 - 1 \end{aligned}$$

$$u = 1 - 7k_1$$

k_1 ist frei wählbar, also z. B. $k_1 = 1$, somit ergibt sich

$$\begin{aligned} u &= 1 - 7 = -6 \\ v &= -17 \cdot -6 + 1 = 103 \\ d &= 103 \end{aligned}$$

Nach dieser Berechnung kann man die Primzahlen p und q vernichten, weil sie für die Verschlüsselung nicht mehr nötig sind. Mit der Modulo-Funktion $\text{mod } n$ kann man den öffentlichen Schlüssel e und den privaten Schlüssel d überprüfen [3]:

$$ed \text{ mod } \varphi(n) = 1$$

$$7 \cdot 103 \text{ mod } 120 \equiv 721 \text{ mod } 120 \equiv 1 \text{ mod } 120 = 1$$

Bisher haben wir folgende Werte berechnet:

- $n = 143$ (Modul)
- $e = 7$ (Öffentlicher Schlüssel)
- $d = 103$ (Privater Schlüssel)

Um das Verfahren besser zu verstehen nehmen wir wieder das Beispiel der elektronischen Wahlen. Es gibt zwei Entitäten in diesem Verfahren: A (Bürger) und B (Validator).

A soll jetzt seine Wahl von B blind signieren lassen, somit schickt A seine Nachricht $m = 7$ an B . (Die Nachricht trifft seine Wahl, also $m = 7$ könnte z. B. Hillary Clinton sein). Bevor die Datenübermittlung startet, schauen wir noch schnell was A und B kennen:

- A kennt seine Wahl $m = 7$ (d. h. Clinton), das Modul $n = 143$, den öffentlichen Schlüssel $e = 7$ von B und einen Blendungsfaktor $r = 2$ (Der Blendungsfaktor ist frei wählbar aber er muss teilerfremd zum Modul n sein).
- B kennt den öffentlichen Schlüssel $e = 7$, seinen privaten Schlüssel $d = 103$ und das Modul $n = 143$.

Der Prozess der blinden Signatur ist in fünf Schritte strukturiert:

1. A erzeugt mit Hilfe der Blendungsfaktor $r = 2$ eine geblendete Nachricht x und schickt sie an B . $x = mr^e \text{ mod } n = 7 \cdot 2^7 \text{ mod } 143 \equiv 896 \text{ mod } 143 \equiv 38 \text{ mod } 143 = 38$. Somit wäre die geblendete Nachricht $x = 38$.
2. B unterschreibt x mit seinem privaten Schlüssel d : $y = x^d \text{ mod } n = 38^{103} \text{ mod } 143 \equiv 14^{51} \cdot 38 \text{ mod } 143 \equiv 27^{17} \cdot 38 \text{ mod } 143 \equiv 14^8 \cdot 27 \cdot 38 \text{ mod } 143 \equiv 27 \cdot 27 \cdot 38 \text{ mod } 143 \equiv 27702 \text{ mod } 143 \equiv 103 \text{ mod } 143 = 103$.
3. B sendet das Ergebnis $y = 103$ zurück an A .
4. Bei 1. wählte A einen Blendungsfaktor $r = 2$, um die nachricht $m = 7$ für B nicht lesbar zu machen. Um die Unterschriebene Nachricht $y = 103$ wieder im Klartext zu sehen, muss A das inverse Element von r also r^{-1} berechnen:

$$\begin{aligned} r \cdot r^{-1} \text{ mod } n &= 1 \\ 2 \cdot r^{-1} \text{ mod } 143 &= 1 \\ 2 \cdot r^{-1} - 1 &= 0 \text{ mod } 143 \\ 2 \cdot r^{-1} - 1 &= 143k \\ 2r^{-1} &= 143k + 1 \\ r^{-1} &= \frac{143k + 1}{2} = \frac{142k + k + 1}{2} = 71k + \frac{k + 1}{2} \end{aligned}$$

$$k + 1 = 2k_1$$

$$k = 2k_1 - 1$$

k_1 ist frei wählbar, z. B. $k_1 = 1$

$$k = 2 - 1 = 1$$

$$r^{-1} = 71 + 1 = 72$$

5. A kann jetzt mit Hilfe von r^{-1} die unterschriebene Nachricht bekommen

$$yr^{-1} \bmod n = 103 \cdot 72 \bmod 143 \equiv 7416 \bmod 143 \equiv 123 \bmod 143 = 123$$

Das elektronische Abstimmungssystem (Validator) hat die Nachricht m visiert, ohne den Inhalt zu wissen, also die Nachricht wurde blind signiert. Wichtig ist dass der Blendungsfaktor r nur im Besitz des Bürgers bleibt, sonst konnte man natürlich den Inhalt der Mitteilung ohne Probleme lesen.

Bei diesen kleinen Zahlen die im Beispiel verwendet wurden, kann man noch relativ einfach durch probieren von der geblendeten Nachricht auf den Klartext zu kommen. Aber in der Praxis werden solche Zahlen viel grösser gewählt und somit ist es fast unmöglich eine geblendete Nachricht im Klartext zu rekonstruieren.

3 Vergleich

	Hashing	Blinde Signatur
Einsatzgebiet	Prüfsumme (CRC) für Dateiübertragungen, Indizierung Dateien, Passwörter abspeichern	Elektronische Wahlen, elektronische Münzen, verschiedene Dateien blind signieren
Struktur	Einwegfunktion	Öffentlicher und privater Schlüssel
Algorithmus/-en	MD5 (128bit), SHA (160bit), Tiger (192 bit), HAVAL (128-256 bit), LM-Hash (128 bit)	RSA (Variabel Länge, normal 1024 bit)
Datenreduktion	128 bis 256 bit	Keine Datenreduktion
Sicherheit	Nur mit viel Zeitaufwand knackbar	Nur mit viel Zeitaufwand knackbar
Sicherheit - Chaotisch	Keine Struktur erkennbar	Ursprüngliche Nachricht ist gut versteckt
Surjektivität / Kollisionen	Mit MD5 und sha-1 gibt's praktisch keine Kollisionen	Keine, die Wahrscheinlichkeit, dass man eine Kollision findet, ist sehr klein
Vorteile	Effizient	Sicher
Nachteile	ev. Kollisionen	Etwas langsam

Tabelle 1: Vergleich Hashing - Blinde Signatur

4 Quellenverzeichnis

4.1 Quellen aus dem Internet

- [Md5a08] MD5-Algorithmus
[http://de.wikipedia.org/wiki/Message-Digest_Algorithm_5, 24.04.2008]
- [Md5e08] MD5-Encrypter
[<http://md5.xpzone.de/>, 24.04.2008]
- [Rafa08] Blinde Signaturen von Rafael Pfister, HTW Chur
- [Info08] Hashing, Informatikjahr 2006
[<http://www-i1.informatik.rwth-aachen.de/~algorithmus/algo34.php>, 24.04.2008]

4.2 Quellen aus den Büchern

- [Albr06] Albrecht Beutelspacher, Moderne Verfahren der Kryptographie, 6. Auflage, Seiten 33 - 35

5 Anhang

5.1 Abbildungen

- Abbildung 1: Prinzip einer Hash-Funktion 3

5.2 Tabellen

- Tabelle 1: Vergleich Hashing - Blinde Signatur 8

6 Literatur

- [1] Message-Digest Algorithm 5, http://de.wikipedia.org/wiki/Message-Digest_Algorithm_5
- [2] Surjektive Funktionen, <http://de.wikipedia.org/wiki/Surjektiv>
- [3] Albrecht Beutelspacher, Jörg Schwenk, Klaus-Dieter Wolfenstetter. *Moderne Verfahren der Kryptographie*. 6. Auflage. RSA-Algorithmus, Seiten 17-19