6.2 Hash-Funktionen

Ein besonders wichtiger Spezialfall der Einweg-Funktionen sind die Hash-Funktionen, auch „Message Digest“ oder „kryptographische Prüfsumme“ genannt.

Definition 1. Sei Σ ein Alphabet und $n \in \mathbb{N}$ eine feste natürliche Zahl ≥ 1. Dann heißt eine Einweg-Funktion

$$h\colon \Sigma^* \longrightarrow \Sigma^n$$

schwache Hash-Funktion über Σ.

Zeichenketten *beliebiger* Länge werden dabei also auf Zeichenketten vorgegebener *fester* Länge abgebildet. Da Σ^* unendlich ist, ist gemeint, dass die Einschränkung von h auf Σ^r für alle genügend großen r Einweg-Funktion ist.

Definition 2. Eine Einweg-Funktion $f : M \longrightarrow N$ heißt **kollisionsfrei**, wenn es nicht effizient möglich ist, $x_1, x_2 \in M$ zu finden mit $x_1 \neq x_2$, aber $f(x_1) = f(x_2)$.

Man könnte das auch als „praktisch injektiv“ bezeichnen; injektive Einweg-Funktionen sind natürlich kollisionsfrei. Ist $\#M > \#N$, so kann f nicht injektiv, könnte aber durchaus kollisionsfrei sein.

Definition 3. Eine **(starke) Hash-Funktion** ist eine kollisionsfreie schwache Hash-Funktion.

Für die praktische Anwendung (meistens mit $\Sigma = \mathbb{F}_2$) soll die Länge n der Hashwerte klein sein. Da die Umkehrbarkeit aber nicht effizient sein darf, will man kryptographische Sicherheit erreichen, muss n andererseits auch genügend groß sein. Geht man davon aus, dass die Hash-Funktion statistisch zufällig aussehende, gleichverteilte Werte liefert, so muss man bei einer schwachen Hash-Funktion also sicher vor Exhaustion (vollständiger Suche) sein. Das bedeutet, dass $n = 80$ als Untergrenze gerade nicht mehr ausreichend ist, man also besser 128-Bit-Hashwerte verwenden sollte. Das ist gerade die Länge bei den bekannten Verfahren MD2, MD4, MD5.

Bei praktisch allen Anwendungen ist aber auch die Kollisionsfreiheit wichtig. Hier ist das Geburtstagsphänomen, siehe I.2.6, zu berücksichtigen: Um Kollisionen mit hinlänglicher Sicherheit unauffindbar zu machen, ist etwa die doppelte Bitlänge nötig. Hashwerte von 160 Bit sind also gerade nicht mehr lang genug. Die noch gültigen Standard-Hashverfahren SHA-1 und RIPEMD verwenden genau diese Länge, sollten also schleunigst ausgemustert werden. Im Kontext der AES-Standardisierung wurde das Hash-Verfahren SHA-2 mit mindestens 256-Bit-Hashwerten eingeführt,

das dann auch passenderweise als SHA-256 usw. bezeichnet wird [siehe `http://csrc.nist.gov/publications/`]. Für die MDx-Verfahren wurden tatsächlich schon vor einiger Zeit systematisch Kollisionen gefunden [Dobbertin 1996ff.], für SHA-1 unlängst (2005).

Anwendungen: (Starke) Hash-Funktionen verwendet man

- bei der digitalen Signatur. Eine lange Nachricht direkt mit dem privaten Schlüssel zu verschlüsseln würde bei der Langsamkeit der asymmetrischen Verfahren zu lange dauern. Daher signiert man einen Hashwert der Nachricht. Damit das sicher ist, braucht man eine kollisionsfreie Hash-Funktion. Sonst nämlich könnte ein Angreifer sich auf folgende Weise ein beliebiges Dokument a von Alice signieren lassen: Er fertigt ein unverdächtiges Dokument b an, das Alice gerne unterschreibt. Von beiden Dokumenten stellt er $m = 2^k$ Varianten $a_1, \ldots, a_m$ und $b_1, \ldots, b_m$ her, indem er an k verschiedenen Stellen jeweils ein Leerzeichen einfügt oder nicht. Falls er eine Kollision findet, etwa $h(a_i) = h(b_j)$, lässt er b_j von Alice signieren und hat dann eine gültige Signatur für a_i.
- für die Umwandlung einer langen, für einen Menschen merkbaren Passphrase in einen (schwer merk- und eingebbaren) n-Bit-Schlüssel für eine symmetrische Chiffre.