6.2 Hash Functions

Hash functions are the most important special cases of one-way functions. They are also known as "message digests" or "cryptographic check sums".

Definition 1 Let Σ be an alphabet and $n \in \mathbb{N}$ be a fixed integer ≥ 1. A one-way function

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

is called **weak hash function** over Σ.

It maps character strings of *arbitrary* lengths to character strings of a given *fixed* length. (Since Σ^* is infinite we interpret the one-way property as: the restriction of h to Σ^r is one-way for all sufficiently large r.

Definition 2 A one-way function $f\colon M \longrightarrow N$ is called **collision free** if there is no efficient way to find $x_1, x_2 \in M$ with $x_1 \neq x_2$, but $f(x_1) = f(x_2)$.

This is a kind of "virtual injectivity". Needless to say that true injective one-way functions are collision free. If $\#M > \#N$, then f cannot be injective, but nevertheless could be collision free.

Definition 3 A **(strong) hash function** is a collision free weak hash function.

For practical applications (mostly with $\Sigma = \mathbb{F}_2$) the length n of the hash values should be as small as possible. On the other hand to exclude efficient invertibility, and thus to get cryptographic security, n must be sufficiently large. We want a weak hash function to deliver uniformly distributed values that look statistically random, and to be safe from an exhaustion attack as illustrated in Figure 6.1. Inserting m blanks at will we generate 2^m different—but optically indistinguishable—versions of a text document. If m is large enough, with high probability one of these versions will have the given hash value.

row 1	(add blank)	$\rightarrow$ 2 different versions
$\vdots$	$\vdots$	
row i	(add blank)	$\rightarrow$ 2 different versions
	$\vdots$	$\vdots$
row m	(add blank)	$\rightarrow$ 2 different versions

Figure 6.1: An exhaustion attack: How to fake a document to have a given hash value by generating 2^m different versions

As a consequence $n = 80$ is just too weak as a lower bound, we'd better use 128-bit hashes. This is the hash length of the well-known but outdated functions MD2, MD4, MD5.

But virtually all applications even need collision free hashes. Remember the birthday paradox, see I.2.6: To exclude collisions with sufficient certainty we need about twice the bitlength than for the one-way property. So hash values of 160 bits are just below the limit. The former standard hash functions SHA-1 and RIPEMD use exactly this length. Their use is strongly discouraged. In the context of AES the hash function SHA-2 with at least 256-bit values was specified, conveniently also denoted as SHA-256 etc. [see `http://csrc.nist.gov/publications/`]. The new standard SHA-3 is valid since 2015.

In fact for the MDx functions there is a systematic way to find collisions [Dobbertin 1996ff.], also SHA-1 collisions are known (2005).

document a		document b	
row 1	(add blank)	row 1	(add blank)
$\vdots$	$\vdots$	$\vdots$	$\vdots$
row m	(add blank)	row m	(add blank)

Figure 6.2: A collision attack: How to fake a document to have the same hash value as another document

Applications

(Strong) hash functions are in use for

- digital signatures: To sign a long message with the private key would take much time due to the slowness of asymmetric ciphers. The standard procedure is to sign a hash of the message.

  For security we need a collision free hash function. Otherwise the attacker could get a valid signature for an arbitrary document a without stealing Alice's private key: He produces an innocently looking document b that Alice is glad to sign. Then he fabricates $q = 2^m$ variants $a_1, \ldots, a_q$ and $b_1, \ldots, b_q$ of both documents, for example by inserting spaces at m different positions. If he finds a collision $h(a_i) = h(b_j)$, he lets Alice sign b_j, getting a valid signature for a_i too.

- transforming a long, but memorizable passphrase ("Never change a working % password 24 because you'll ? forget the nEW one+") into an n-bit key (`BA8C0C8C1C65364F` in hexadecimal notation) for a symmetric cipher.