Hash Functions and Data Integrity

Cryptographic hash functions play a role in data integrity and message authentication. A hash function is a function from strings of arbitrary finite bit length to strings of n bits for some fixed integer n. A hash function is necessarily many-to-one, but for cryptographic applications n will be in the range of 128 to 256 bits and should satisfy much stronger conditions than are required for typical hashing purposes. These conditions are classified by the difficulty of solving certain problems, as presented in the table below.

Preimage resistance:	Given $H(x)$ find y such that $H(y) = H(x)$.
Second preimage resistance:	Given x, find y such that $H(y) = H(x)$.
Collision resistance:	Find x, y such that $H(x) = H(y)$

Standard examples of hash functions are SHA-1 and MD5.

Message Authentication Codes

A message authentication code is a keyed hash function, such that the hash value depends on an input key. These have the principle application to the problem of data integrity and message authentication — the hash value of a file or data plus the public key of the originating party serve to verify that the data has not been alterred in transmission.

A message authentication code can be constructed from a block cipher, such as DES, using CBC mode. The protocol is given by the following steps.

Input: Message m, n-bit block cipher E, and secret MAC key K for E.

Algorithm:

1. Pad m if necessary and subdivide it into n-bit blocks $m_1, m_2, \ldots m_t$.

2. CBC processing: set $H_0 = 0 \ldots 0$, and compute $H_i = E_K(m_i \oplus H_{i-1})$.

Output: H_t.

N.B. the message m must be padding in a well-defined way. Typically this involves adding a tail of all 0's to form a complete block of length n. This has the disadvantage that trailing 0's of the message can not be distinguished from the padding. Alternatively the message can be padded with a 1 followed by all 0's, which can be reversed by chopping off all final zeros and the next 1, but implies that a message block of which is already a multiple of n-bits must adjoin an additional entire block of n-bits.

Digital Signatures

A digital signature is the digital analoque of a handwritten signature. The signature of a message is data dependent on some secret known only to the signer and on the content of the message. A digital signature must be verifiable without access to the signer's private key.

RSA Signature Scheme.

The RSA signature scheme is a signature scheme with *message recovery* — the signed message is recovered from the signature.

Key generation. This step is exactly as for RSA enciphering. The signer generates a public key (e, n) and guards a private key (d, n), where $n = pq$ is the product of two large primes.

Signature generation. Encode the message m in $\mathbb{Z}/n\mathbb{Z}$, and output the signature $s = m^d \in \mathbb{Z}/n\mathbb{Z}$, computed using the private key (d, n).

Verification. Compute $m = s^e \in \mathbb{Z}/n\mathbb{Z}$.

El Gamal Signature Scheme.

The El Gamal signature scheme requires an encoding of the message m as an element of $\mathbb{Z}/(p-1)\mathbb{Z}$.

Key generation. This step is exactly as for El Gamal enciphering. The signer generates a public key (p, a, c), where $c = a^x \bmod p$, and guards the private key (p, a, x), where a is a primitive element of $\mathbb{Z}/p\mathbb{Z}$ and x is an integer in the range $1 \leq x < p-1$ with $\mathrm{GCD}(x, p-1) = 1$.

Signature generation. The signer selects a random secret integer k in the range $1 \leq k < p-1$ with $\mathrm{GCD}(k, p-1) = 1$, and computes

$$r = a^k \bmod p \text{ and } s = l(m - rx) \bmod (p-1),$$

where $l = k^{-1} \bmod (p-1)$, and the signature (r, s) is output.

Note that r is well-defined in $\mathbb{Z}/p\mathbb{Z}$, but that to form s it is necessary to choose a minimal positive integer representative and reinterpret it $\bmod(p-1)$.

Verifcation. The signature is verified first that $1 \leq r \leq p-1$, or rejected. The values

$$v_1 = c^r r^s \bmod p, \text{ and } v_2 = a^m \bmod p,$$

are next computed, and the equality $v_1 = v_2$ is verified or the signature rejected.

Proof of equality. $v_1 = c^r a^{kl(m-xr)} = a^{xr} a^{m-xr} = a^m = v_2$.

Chaum's Blind Signature Scheme

Chaum's blind signature scheme is an RSA-based scheme, adapted for blind signatures. In the protocol below we assume that Bob has set up a public RSA key (e, n) with corresponding private key (d, n), so that Bob's RSA signature functions is $S_B(m) = m^d$.

1. *Initial setup:* Alice obtains Bob's public key (e, n) and chooses a random public session key k, such that $0 < k < n$ and $\mathrm{GCD}(k, n) = 1$.
2. *Blinding:* Alice computes $m^* = mk^e$, and sends m^* to Bob.
3. *Signing:* Bob computes $s^* = m^{*d}$, which he sends back to Alice.
4. *Unblinding:* Alice computes $s = k^{-1}s^*$, which equals $S_B(m) = m^d$.

As an application we mention a naive digital cash scheme. Suppose that Alice wants to withdraw a digital $100 from her account to be spent anonymously at a later date. She writes 1000 notes from the bank, each certifying its value to be $100, and blinds them, each with a separate session key. The bank asks for the session keys to 999 of these notes, verifies that each has the correct value, and blindly signs the last one, deducting $100 from her account, and returns the blinded signed $100 note to Alice for use as cash.

Ideal Properties of Digital Cash Schemes

We won't go into details of a particular digital cash protocol, but list the ideal properties which such a scheme should satisfy, as spelled out by Okomoto and Ohta in 1991 (Crypto'91).

1. Digital cash can be sent securely through an insecure channel.
2. Digital cash can not be copied or reused.
3. The spender remains anonymous under legitimate use of the protocol.
4. Spending does not require communication with a bank or external agency.
5. The cash is transferrable.
6. The cash can be subdivided.

There are several proposed digital cash schemes, which provide both partial and full solutions to these sets of conditions. Okamoto and Ohta provide a solution to all six of these conditions. Chaum has proposed a variety of schemes which give partial solutions to different subsets of the above, and Brands has a scheme which satisfies the first four properties. The complexity of the scheme is largely dependent on the number of these properties which it satisfies, so that the most complete scheme may not be the easiest to describe or to implement.

We note that anonymity, property three of this list, relies on an analogue blind signatures called restricted blind signatures, as in the naive example above. The naive example fails the above criteria, for instance, failing to ensure against multiple spending.