

One-way ciphers

Used to protect on-line passwords.

Recall main frame days.

Purdy sparse polynomial:

$$f(x) = (x^n + a_{n-1}x^{n-1} + \dots + a_1x + a_0) \bmod p$$

Example: $n = 2^{24}$, $p \approx 2^{64}$.

It takes about $O(n^2(\log p)^2)$ steps to solve $f(x) = c$.

UNIX stores a password as $DES_{pass}(0)$ using the 64-character set a-zA-Z0-9./

A study found that 86% of passwords are easily guessed names or words in dictionaries. Therefore, a 12-bit “salt” is added to each password.

Authentication functions

They enable a recipient to determine that a message came from the alleged sender (although he might not be able to prove this to others) and that the message has not been tampered with (inserted, deleted, changed).

Single key encryption provides some authentication, if only the sender and receiver share a key K .

Public key encryption provides no authentication unless signatures are used; if they are, there is both authentication and non-repudiation.

Encryption also provides confidentiality, often not needed. For example:

1. A significant broadcast message: should you believe it?
2. A program someone sends you. Before you run it, how can you tell that it was not modified en route?

Another type of authentication function uses a secret key K to generate a small, fixed-sized data block, called a cryptographic checksum or message authentication code (MAC), that is appended to the message.

If only the sender and recipient know K , and the recipient verifies the checksum, then he knows the message comes from the alleged sender (because no one else could compute the checksum) and the message has not been altered.

A typical implementation of a cryptographic checksum uses K in DES in CBC mode with initial register value 0 to encipher the message. Only the final block of ciphertext is saved; it is the checksum.

In the case of the significant broadcast message mentioned above, it would be better to have a “key-less” algorithm generate one “checksum” and then encipher it once for each recipient to produce authenticity.

In this situation the “key-less checksum” is called a Message Digest or Hash Function. They are a common way to provide authentication without confidentiality.

Hash Functions

A *weak hash function* is a function $h = H(M)$ of fixed length (m bits) of a message M of any length such that:

1. Given M , it is easy to compute $H(M)$,
2. Given h , it is hard to compute any M for which $H(M) = h$, and
3. Given M , it is hard to find $M' \neq M$ for which $H(M') = H(M)$.

A *strong hash function* is a weak hash function which also satisfies:

4. It is hard to find *any* two messages $M' \neq M$ for which $H(M') = H(M)$.

Usually $H(M)$ is either enciphered or transmitted separately from M .

Property 3 provides the authentication.

Property 4 protects M against “birthday attacks”.

Most hash functions are built from a one-way function f which takes two values of length m bits and produces a value of length m bits. The whole message M is broken into blocks M_i of length m bits each. One computes $h_i = f(M_i, h_{i-1})$ with some standard initial value h_0 . The hash value (or digest) is the final h_i .

Some examples of hash functions are SNEFRU, N-Hash, MD4, MD5, and SHA.

We discuss MD5 and SHA. MD5 produces a 128-bit digest, while SHA's digest has 160 bits.