

HASH FUNCTION:

It is a one of the authentication function; it accepts a variable size message M as input and produces a fixed size output.

A hash value 'h' is generated by a function H of the form

$$h = H(M)$$

$M \rightarrow$ variable length message

$H(M) \rightarrow$ fixed length hash value.

The hash code is also referred as Message Digest (MD) or hash value.

The main difference between Hash Function and MAC is a hash code does not use a key but is a function only of the input message.

The hash value is appended to the message at the source at a time when the message is assumed or known to be correct.

The receiver authenticates that message by re-computing the hash value.

Hash functions are often used to determine whether or not data has changed.

Figure 11.1 depicts the general operation of a cryptographic hash function

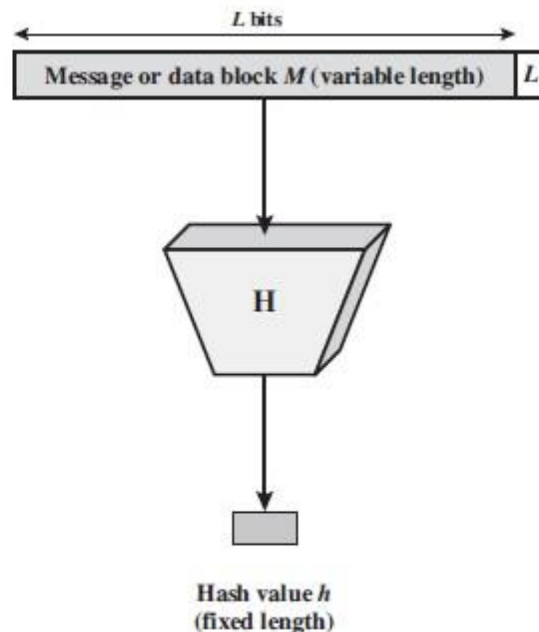


Figure 11.1 Black Diagram of Cryptographic Hash Function; $h = H(M)$

APPLICATIONS OF CRYPTOGRAPHIC HASH FUNCTIONS

It is used in a wide variety of security applications and Internet protocols

Message Authentication

Message authentication is a mechanism or service used to verify the integrity of a message. Message authentication assures that data received are exactly as sent (i.e., contain no modification, insertion, deletion, or replay)

When a hash function is used to provide message authentication, the hash function value is often referred to as a message digest.

Figure 11.2 illustrates a variety of ways in which a hash code can be used to provide message authentication, as follows.

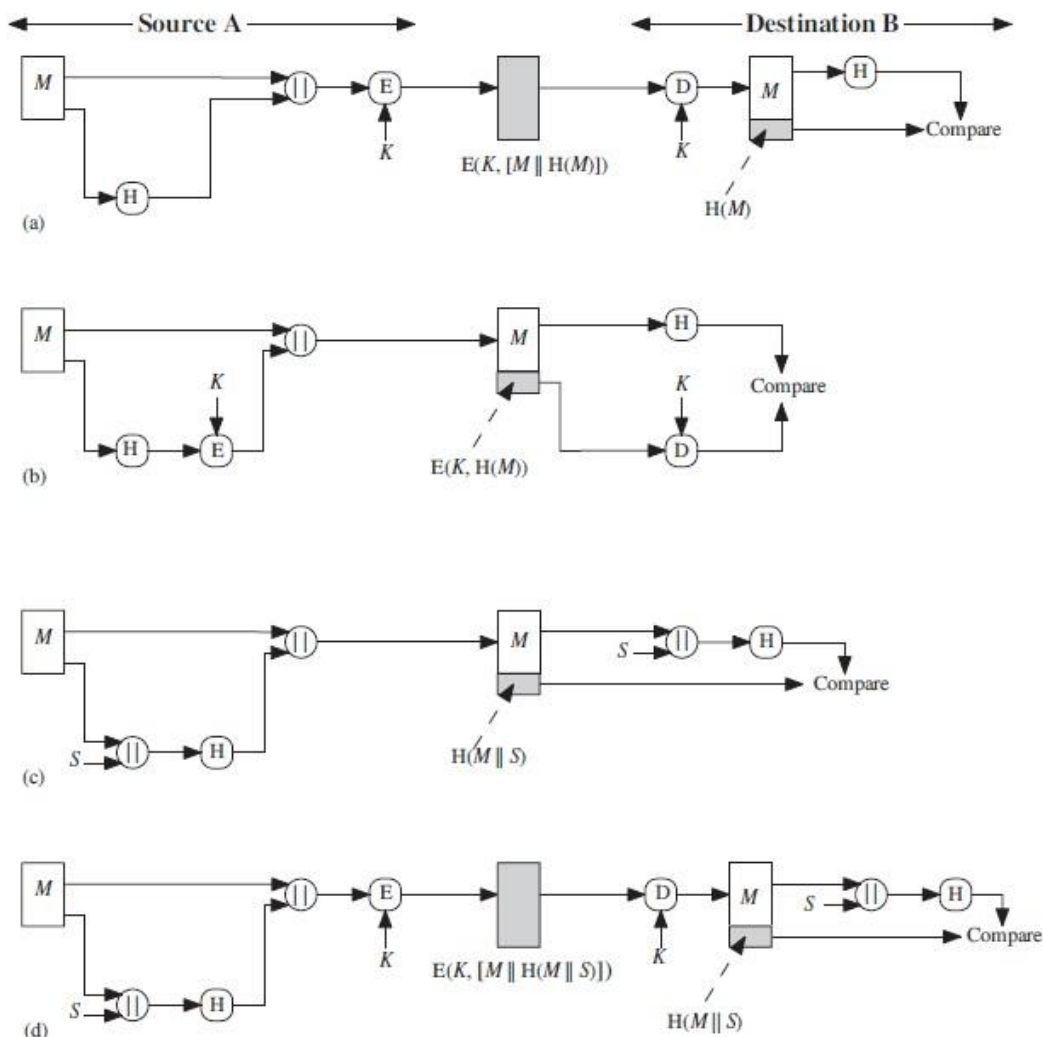


Figure 11.2 Simplified Examples of the Use of a Hash Function for Message Authentication

(a) The message plus concatenated hash code is encrypted using symmetric encryption. Because only A and B share the secret key, the message must have come from A and has not been altered.

The hash code provides the structure or redundancy required to achieve authentication. Because encryption is applied to the entire message plus hash code, confidentiality is also provided.

(b) Only the hash code is encrypted, using symmetric encryption. This reduces the processing burden for those applications that do not require confidentiality

(c) It is possible to use a hash function but no encryption for message authentication. The technique assumes that the two communicating parties share a common secret value S . A computes the hash value over the concatenation of M and S and appends the resulting hash value to M . Because B possesses, it can recompute the hash value to verify. Because the secret value itself is not sent, an opponent cannot modify an intercepted message and cannot generate a false message.

(d) Confidentiality can be added to the approach of method (c) by encrypting the entire message plus the hash code.

Digital Signatures

Another important application, which is similar to the message authentication application, is the digital signature.

The operation of the digital signature is similar to that of the MAC. In the case of the digital signature, the hash value of a message is encrypted with a user's private key. Anyone who knows the user's public key can verify the integrity of the message that is associated with the digital signature.

Figure 11.3 illustrates, in a simplified fashion, how a hash code is used to provide a digital signature.

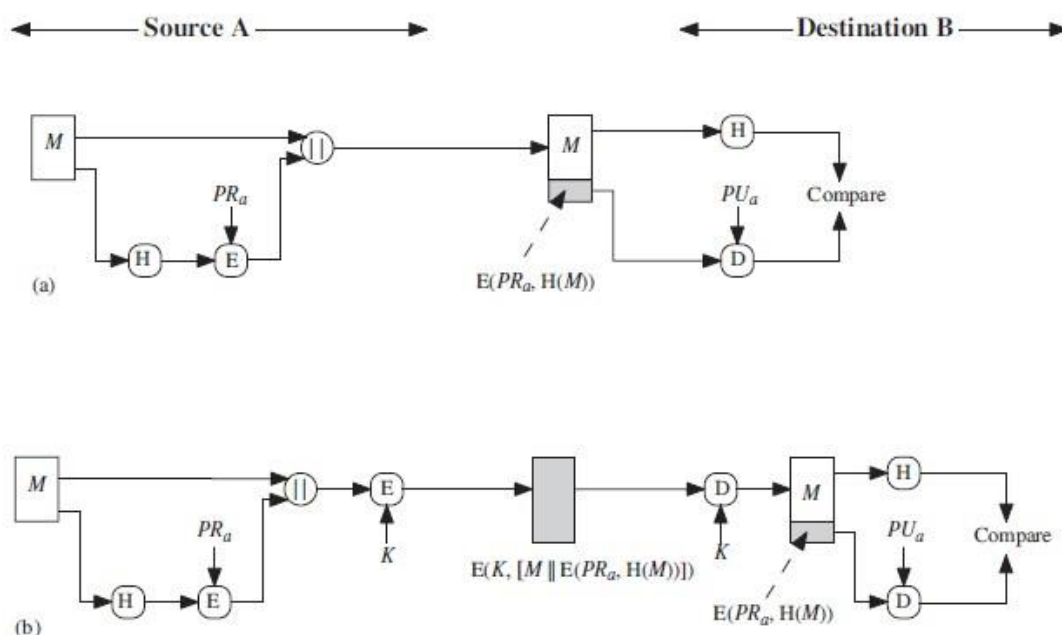


Figure 11.3 Simplified Examples of Digital Signatures

REQUIREMENTS& SECURITY FOR A HASH FUNCTION:

The purpose of a hash function is to produce a “fingerprint” of a file, message or other block of data. To be useful for message authentication, a hash function H must have the following properties:

H can be applied to a block of data of any size H

produces a fixed length output.

$H(x)$ is relatively easy to compute for any given x , making both hardware and software implementations practical.

One-way property: - for any given value h , it is computationally infeasible to find x such that $H(x)=h$. this sometimes referred to in the literature as the one way property.

Weak collision resistance:- for any given block x . it is computationally infeasible to find $y \neq x$ with $H(y)=H(x)$. this is referred as weak collision resistance.

Strong collision resistance:- it is computationally infeasible to find any pair (X,Y) such that $H(x)=H(y)$. this is referred as strong collision resistance.

Requirements for a Cryptographic Hash Function H

Requirement	Description
Variable input size	H can be applied to a block of data of any size.
Fixed output size	H produces a fixed-length output.
Efficiency	$H(x)$ is relatively easy to compute for any given x , making both hardware and software implementations practical.
Preimage resistant (one-way property)	For any given hash value h , it is computationally infeasible to find y such that $H(y) = h$.
Second preimage resistant (weak collision resistant)	For any given block x , it is computationally infeasible to find $y \neq x$ with $H(y) = H(x)$.
Collision resistant (strong collision resistant)	It is computationally infeasible to find any pair (x, y) such that $H(x) = H(y)$.
Pseudorandomness	Output of H meets standard tests for pseudorandomness.

A hash function that satisfies the first five properties in Table 11.1 is referred to as a weak hash function. If the sixth property, collision resistant, is also satisfied, then it is referred to as a strong hash function.

As with encryption algorithms, there are two categories of attacks on hash functions: brute-force attacks and cryptanalysis

Brute-Force Attacks

A brute-force attack does not depend on the specific algorithm but depends only on bit length. In the case of a hash function, a brute-force attack depends only on the bit length of the hash value. A cryptanalysis, in contrast, is an attack based on weaknesses in a particular cryptographic algorithm.

Cryptanalysis

As with encryption algorithms, cryptanalytic attacks on hash functions seek to exploit some property of the algorithm to perform some attack other than an exhaustive search. The way to measure the resistance of a hash algorithm to cryptanalysis is to compare its strength to the effort required for a brute-force attack.

That is, an ideal hash algorithm will require a cryptanalytic effort greater than or equal to the brute-force effort.