JAYOTI VIDYAPEETH WOMEN'S UNIVERSITY, JAIPUR
Government of Rajasthan established
Through ACT No. 17 of 2008 as per UGC ACT 1956
NAAC Accredited University

Faculty of Education and methodology

Department of Science and Technology

Faculty Name- Jv'n Narendra Kumar Chahar (Assistant Professor)

Program- B.Tech 8thSemester

Course Name– Cryptography and Network Security

Session no.: 27

Session Name-Hash Functions

Academic Day starts with –

- Greeting with saying **'Namaste'** by joining Hands together following by 2-3 Minutes Happy session, Celebrating birthday of any student of respective class and **National Anthem**.

Lecture starts with- quotations' answer writing

Review of previous Session **– MAC**

Topic to be discussed today- Today We will discuss about **Hash Functions**

Lesson deliverance (ICT, Diagrams & Live Example)-

- Diagrams

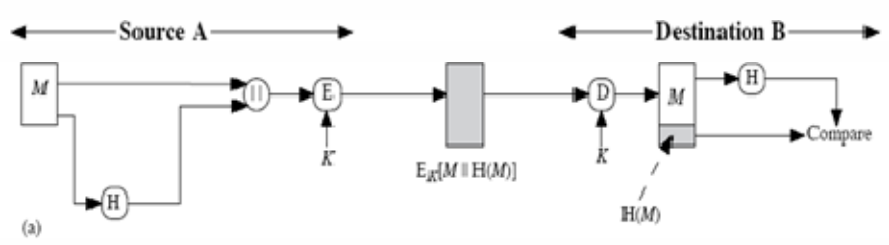
Introduction & Brief Discussion about the Topic**– Hash Functions**

Hash Functions

A variation on the message authentication code is the one-way hash function. As with MAC, a hash function accepts a variable size message M as input and produces a fixed-size output, referred to as hash code H(M). Unlike a MAC, a hash code does not use a key but is a function only of the input message. The hash code is also referred to as a message digest or hash value.

There are varieties of ways in which a hash code can be used to provide message authentication, as follows:

The message plus the hash code is encrypted using symmetric encryption. This is identical to that of internal error control strategy. Because encryption is applied to the entire message plus the hash code, confidentiality is also provided.



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

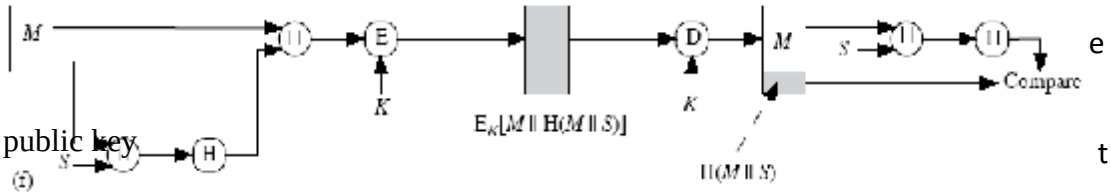


Figure 11.5 Basic Uses of Hash Function (page 2 of 2)

This technique uses a hash function, but no encryption for message authentication. This technique assumes that the two communicating parties share a common secret value ‘S’. The source computes the hash value over the concatenation of M and S and appends the resulting hash value to M.

Confidentiality can be added to the previous approach by encrypting the entire message plus the hash code.

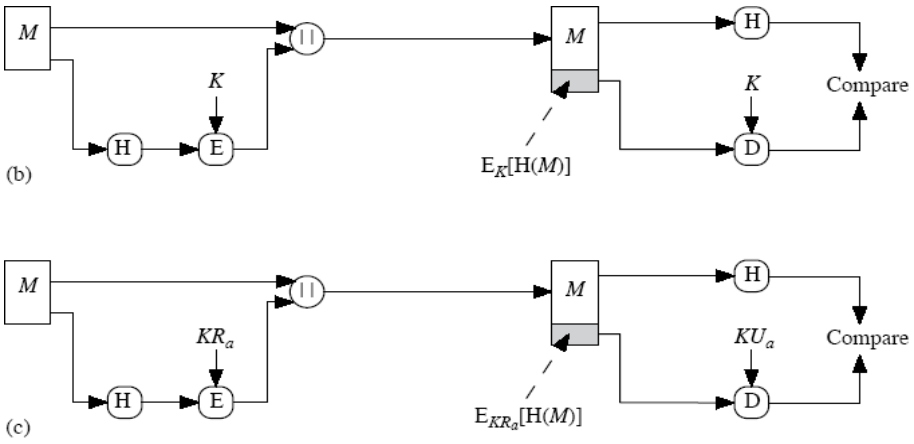


Figure 11.5 Basic Uses of Hash Function (page 1 of 2)

A hash value h is generated by a function H of the form h = H(M)

Where M is a variable-length message and H(M) is the fixed-length hash value. 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.

Requirements for a Hash Function

1. H can be applied to a block of data of any size.
2. H produces a fixed-length output.
3. H(x) is relatively easy to compute for any given x, making both hardware and software implementations practical.
4. For any given value h, it is computationally infeasible to find x such that H(x) = h. This is sometimes referred to in the literature as the one-way property.
5. For any given block x, it is computationally infeasible to find y x such that H(y) = H(x). This is sometimes referred to as weak collision resistance.
6. It is computationally infeasible to find any pair (x, y) such that H(x) = H(y). This is sometimes referred to as strong collision resistance.

The first three properties are requirements for the practical application of a hash function to message authentication. The fourth property, the one-way property, states that it is easy to generate a code given a message but virtually impossible to generate a message given a code.

The fifth property guarantees that an alternative message hashing to the same value as a given message cannot be found. This prevents forgery when an encrypted hash code is used. The sixth property refers to how resistant the hash function is to a type of attack known as the birthday attack, which we examine shortly.

Simple Hash Functions

All hash functions operate using the following general principles. The input (message, file, etc.) is viewed as a sequence of n-bit blocks. The input is processed one block at a time in an iterative fashion to produce an n-bit hash function.

One of the simplest hash functions is the bit-by-bit exclusive-OR (XOR) of every block. This can be expressed as follows:

Ci = bi1 ⊕ bi1 ⊕ ... ⊕ bim Where

Ci = ith bit of the hash code, $1 \leq i \leq n$

m = number of n-bit blocks in the input bij = ith bit in jth block= XOR operation

Thus, the probability that a data error will result in an unchanged hash value is 2n. With more predictably formatted data, the function is less effective. For example, in most normal text files, the high-order bit of each octet is always zero. So if a 128-bit hash value is used, instead of an effectiveness of 2128, the hash function on this type of data has an effectiveness of 2112.

A simple way to improve matters is to perform a one-bit circular shift, or rotation, on the hash value after each block is processed. The procedure can be summarized as follows:

Initially set the n-bit hash value to zero.

Process each successive n-bit block of data as follows:

Rotate the current hash value to the left by one bit. b. XOR the block into the hash value.

Reference-

1. **Book:** William Stallings, “Cryptography & Network Security”, Pearson Education, 4th Edition 2006.

QUESTIONS: -

Q1. What are the hash functions? Explain.

Q2. What are the requirements for the hash functions?

Next, we will discuss more about Birthday Attacks.

.

- Academic Day ends with-
  National song ‘Vande Mataram’