DOP: / /2023 **DOS: / /2023**

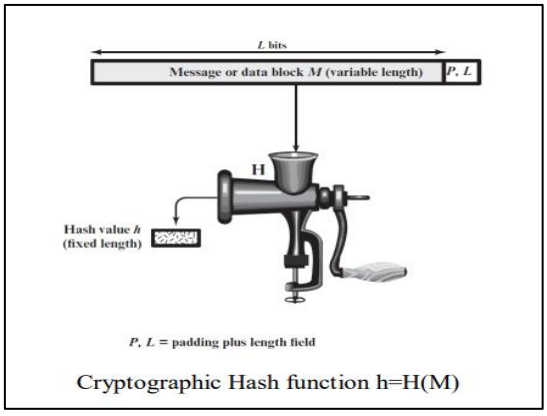
Experiment No: 05

Aim: Cryptographic Hash Functions and Applications (HMAC): to understand the need, design and applications of collision resistant hash functions.

Theory:

◆Cryptographic Hash Functions:

- A hash function H accepts a variable-length block of data M as input and produces a fixed-size hash value h = H(M).
- A "good" hash function has the property that the results of applying the function to a large set of inputs will produce outputs that are evenly distributed and apparently random. In general terms, the principal object of a hash function is data integrity. A change to any bit or bits in M results, with high probability, in a change to the hash value.
- The kind of hash function needed for security applications is referred to as a cryptographic hash function.
- A cryptographic hash function is an algorithm for which it is computationally infeasible (because no attack is significantly more efficient than brute force) to find either
- A data object that maps to a pre-specified hash result (the one-way property) or two data objects that map to the same hash result (the collision-free property).
- Because of these characteristics, hash functions are often used to determine whether or not data has changed.



Cryptographic Hash function h=H(M)

The above figure depicts the general operation of a cryptographic hash function.

Typically, the input is padded out to an integer multiple of some fixed length (e.g., 1024 bits), and the padding includes the value of the length of the original message in bits.

The length field is a security measure to increase the difficulty for an attacker to produce an alternative message with the same hash value.

HMAC Algorithm:

- H = embedded hash function (e.g., MD5, SHA-1, RIPEMD-160
- IV = initial value input to hash function
- M = message input to HMAC (including the padding specified in the embedded hash function)
- Yi _ i th block of M, 0 ≤i ≤(L – 1)
- L _ number of blocks in M
- b _ number of bits in a block
- n _ length of hash code produced by embedded hash function
- K _ secret key; recommended length is ≥n; if key length is greater than b, the key is input to the hash function to produce an n-bit key
- K+ _ K padded with zeros on the left so that the result is b bits in length
- ipad _ 00110110 (36 in hexadecimal) repeated b/8 times
- opad _ 01011100 (5C in hexadecimal) repeated b/8 time

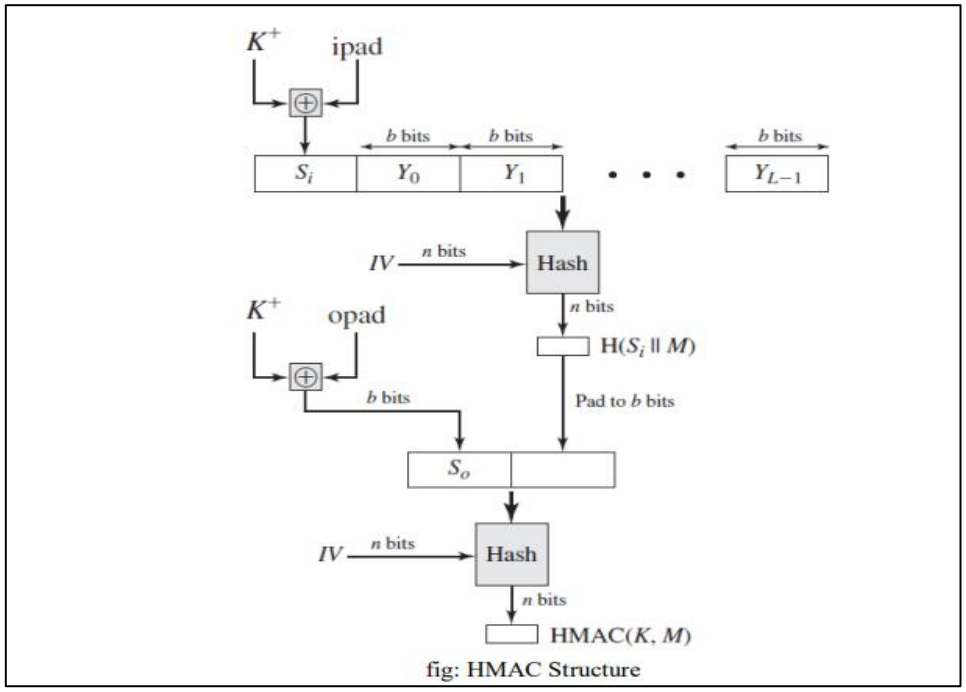


fig: HMAC Structure

Applications of Cryptographic Hash Functions:

The most versatile cryptographic algorithm is the cryptographic hash function. It is used in a wide variety of security applications and Internet protocols. The following are various applications where it is employed.

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., there is 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.

The essence of the use of a hash function for message integrity is as follows.

- The sender computes a hash value as a function of the bits in the message and transmits both the hash value and the message.
- The receiver performs the same hash calculation on the message bits and compares this value with the incoming hash value.
- If there is a mismatch, the receiver knows that the message (or possibly the hash value) has been altered
- The hash value must be transmitted in a secure fashion. That is, the hash value must be protected so that if an adversary alters or replaces the message, it is not feasible for adversary to also alter the hash value to fool the receiver.

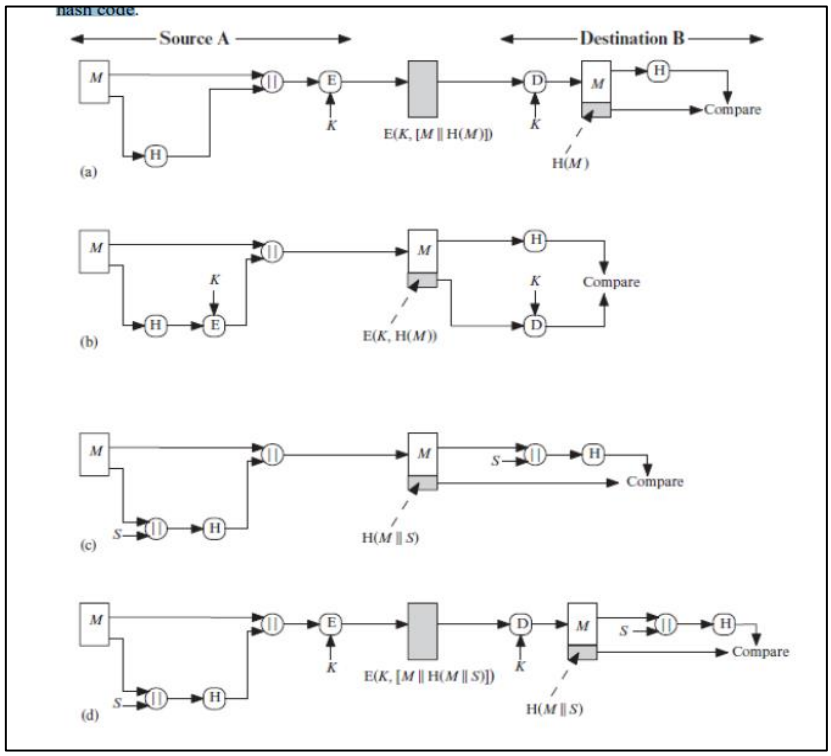
The following are a variety of ways in which a hash code can be used to provide message authentication.

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 S, 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.

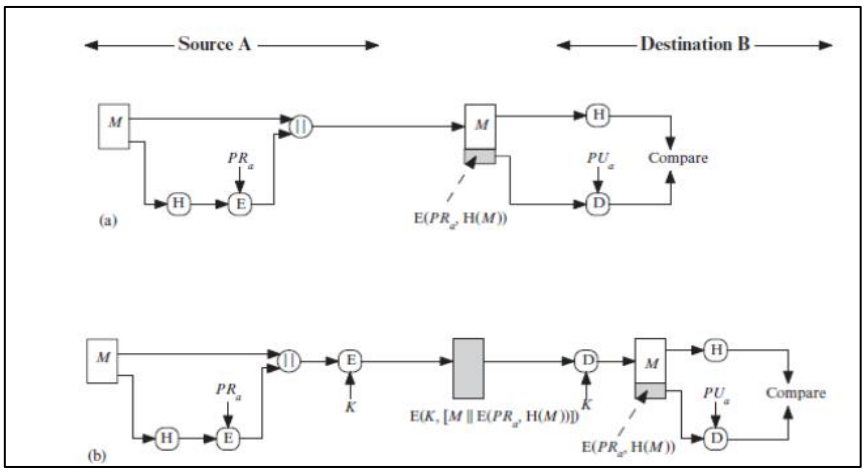


Digital Signatures:

- 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.
- In this case, an attacker who wishes to alter the message would need to know the user's private key.

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

a. The hash code is encrypted, using public-key encryption with the sender's private key. As with Figure b, this provides authentication. It also provides a digital signature, because only the sender could have produced the encrypted hash code. In fact, this is the essence of the digital signature technique.
b. If confidentiality as well as a digital signature is desired, then the message plus the private-key encrypted hash code can be encrypted using a symmetric secret key. This is a common technique.



Other Applications:

- Hash functions are commonly used to create a one-way password file.
- Hash functions can be used for intrusion detection and virus detection
- A cryptographic hash function can be used to construct a pseudorandom function (PRF) or a pseudorandom number generator (PRNG)

Two-Simple Hash Functions:

To get the understanding of security considerations involved in cryptographic hash functions, we present two simple, insecure hash functions in this section.

All hash functions operate using the following general principles.

a. The input (message, file, etc.) is viewed as a sequence of n -bit blocks.
b. 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:

$$Ci = bi1 \oplus bi2 \oplus \ldots \oplus bim$$

Where

Ci = i th bit of the hash code, 1 … i … n

m = number of n-bit blocks in the input

bij = i th bit in j th bloc

$k \oplus$ = XOR operation

- This operation produces a simple parity bit for each bit position and is known as a longitudinal redundancy check.
- It is reasonably effective for random data as a data integrity check. Each n-bit hash value is equally likely.
- Thus, the probability that a data error will result in an unchanged hash value is 2-n.
- 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 2-128, the hash function on this type of data has an effectiveness of 2-112.
- 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. 1. Initially set the n-bit hash value to zero. 2. Process each successive n-bit block of data as follows: a. Rotate the current hash value to the left by one bit. b. XOR the block into the hash value.
- This has the effect of “randomizing” the input more completely and overcoming any regularities that appear in the input.
- Although the second procedure provides a good measure of data integrity, it is virtually useless for data security when an encrypted hash code is used with a plaintext message.
- Although a simple XOR or rotated XOR (RXOR) is insufficient if only the hash code is encrypted, you may still feel that such a simple function could be useful when the message together with the hash code is encrypted.

Conclusion: Hence successfully studied the need, design and application of collusion resistant hash function of Cryptographic Hash Functions and Applications (HMAC).