Network Security:

1) Hash Functions:-

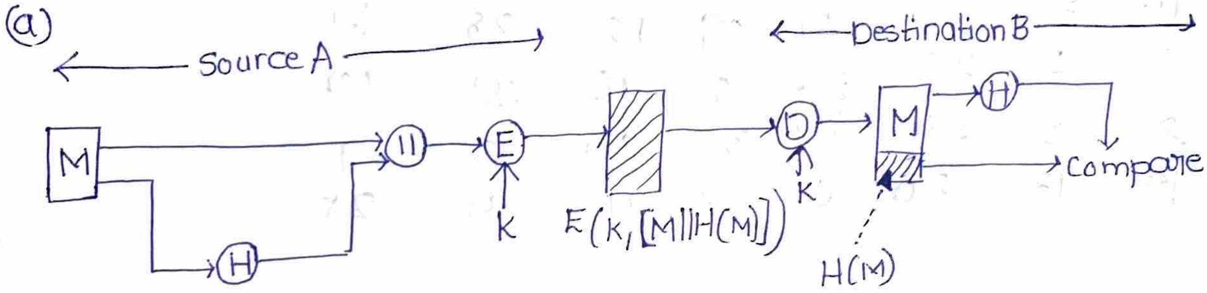
→ A variation on the message authentication code is the one way hash function.

→ As with MAC, a hash function accepts a variable size message as input and produces a fixed-size output, reffered to as Hash code H(M).

→ Unlike a MAC, a hash code does not ~~require~~ use ~~any~~ a key but is the function only of the input message.

→ The hash code is also reffered as message digest (or) has values.

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

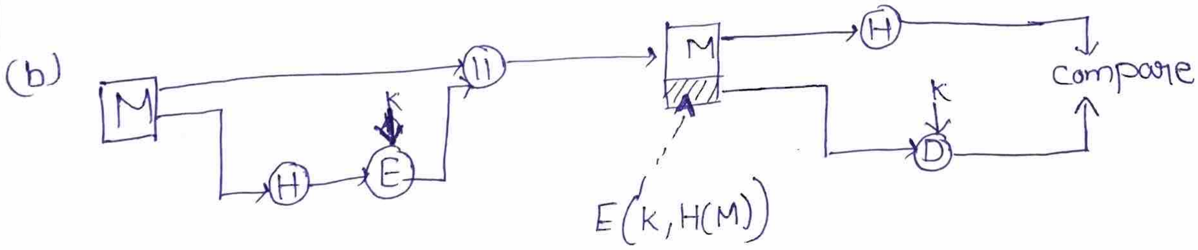


→ The message plus hash code is encrypted using symmetric encryption.

→ Since only A and B share the key, the message must have come from A and has not been altered.

→ This is identical to internal error strategy.

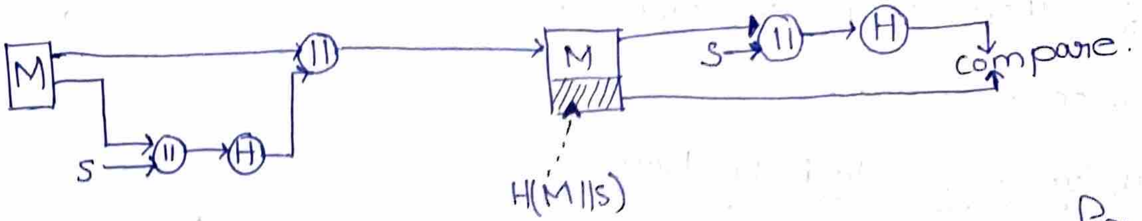
→ Confidentiality is also provided.



→ Only hashe code is encrypted, using symmetric functio encryption

→ This reduces the processing burden for those applications not requiring confidentiality.

(c)

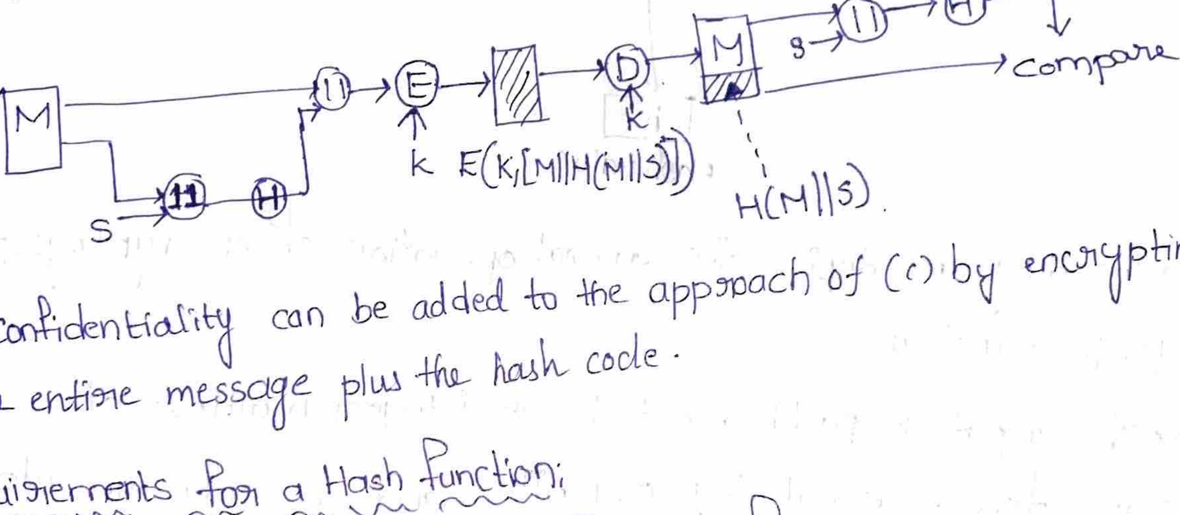


→ Shows the use of Hash function but no encryption for message authentication.

→ This technique assumes that the two communicating parties share a common secret value S.

→ In this technique, the secret value itself is not sent, an opponent cannot modify an intercepted message and cannot create a false message.

(d)



→ Confidentiality can be added to the approach of (c) by encrypti the entire message plus the hash code.

Requirements for a Hash function:

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

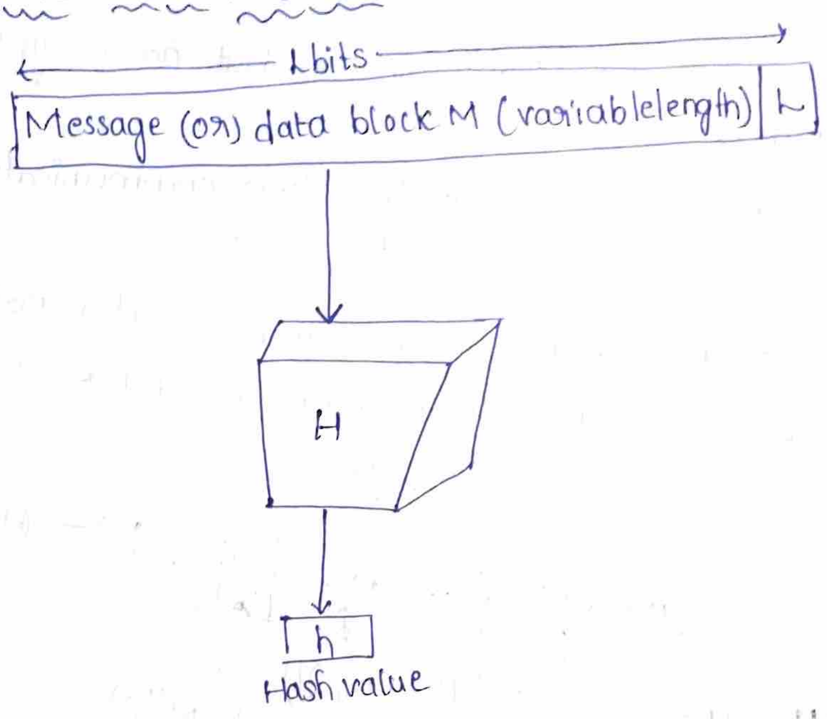
2. H(x) is relatively easy to compute for any given x, making both h/w and s/w implementations practical.

3. For any given value h, it is computationally infeasible to find x such that H(x)=h. Sometimes it reffered to one way property.

4. For any given block x, it is computation infeasible to find y such that H(y) = H(x). Sometimes it is reffered as weak collision resistance.

5. It is computationally infeasible to find pair (x, y) such that H(x) = H(y). Sometimes it is reffered as strong Collision resistance.

Cryptographic Hash function:-



→ the diagram depicts the general operation of a cryptographic hash function.

→ The input is padded out to an integer multiple of some fixed length (eg. 1024 bits) and padding includes the value of ~~(original)~~ length of original message in bits.

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

Features of Hash function:

- → One-way function
- → Deterministic
- → Fixed size output.
- → Collision resistance.

Advantages

→ Data

→ Messa

→ Passwo

→ Fast C

Disadvan

→ Collisi

→ Rainb

→ Limi

→ Hash

Applic

→ Dig

→ Pa

→ Fi

Advantages:

→ Data Integrity

→ Message Authentication

→ Password Storage

→ Fast Computation.

Disadvantages:

→ Collision attacks.

→ Rainbow table attacks.

→ Limited input size.

→ Hash function weaknes.

Applications:

→ Digital Signature

→ Pashword Hashing.

→ File Integrity verification.

MD-5 (Message Digest -5):-

* It is Fast and produces 128 Bit message digest.

working of MD5:-

1. Padding

original message + Padding (Adding extra Bits)

So that total length is 64 Bit less than exact multiple of 512.

Ex:- original msg = 1000 Bits + (?)

512×1 = 512 bits $^{-64}$ × (less than 1000)

512×2 = 1024 bits $^{-64}$ × (less than 1000)

512×3 = 1536 bits ✓

so, 1536 − 64 = 1472.

∴ so Add 472 bits.

1000 bits + 472 bits ⇒ 1472 bits.

2. Appending:-

Append the original length before padding.

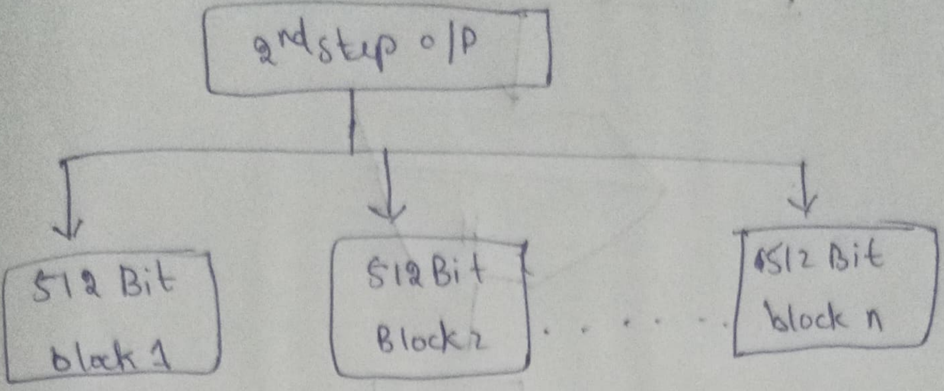
calculate length mod 64

most of cases, 64 bits is obtained as answer.

(∴ append 64 Bits)

so, it again becomes multiple of 512.

3 Dividing:- (each 512 bits)



4 Intialising:- (4 chaining variables).

each = 32 Bits

Ⓐ, Ⓑ, Ⓒ, Ⓓ → values predefined.

5 Processing:- (512 Bit Blocks).

1 Copy ④ Chaining variables into Some Corresponding variables

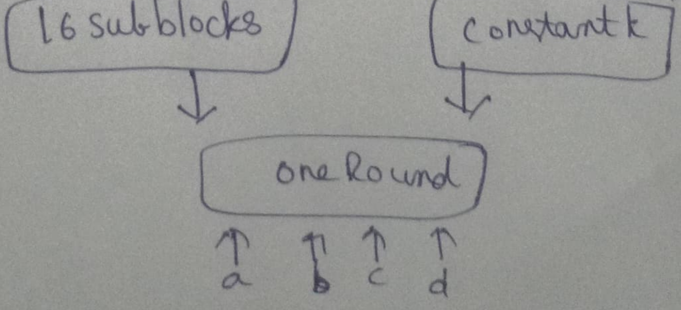
A = a, B = b, C = c, D = d.

2 Divide 512 Bits into 16 - 32 Bit Blocks

16 Blocks → 32 Bits size

3 four rounds

16 subblocks and a Constant (k).



$a = b + ((a + process\ P(b, c, d) + m[i] + T[k]))$