## Q.1 Why Message Authentication is required?

**Ans** In the context of communications across a network, the following attacks can be identified.

1**. Disclosure**: Release of message contents to any person or process not possessing the appropriate cryptographic key.

2. **Traffic analysis:** Discovery of the pattern of traffic between parties. In a connection-oriented application, the frequency and duration of connections could be determined.

3**. Masquerade:** Insertion of messages into the network from a fraudulent source. This includes the creation of messages by an opponent that are purported to come from an authorized entity.

4. **Content modification:** Changes to the contents of a message, including insertion, deletion, transposition, and modification.

5. **Sequence modification**: Any modification to a sequence of messages between parties, including insertion, deletion, and reordering.

6. **Timing modification:** Delay or replay of messages. In a connection-oriented application, an entire session or sequence of messages could be a replay of some previous valid session, or individual messages in the sequence could be delayed or replayed.

7. **Source repudiation**: Denial of transmission of message by source.

8**. Destination repudiation**: Denial of receipt of message by destination.

## Q.2 Explain some message authentication functions?

**Ans.** Authentication Functions that may be used for produce an authenticator. These may be grouped into three classes.

• **Message encryption:** The ciphertext of the entire message serves as its authenticator

• **Message authentication code (MAC):** A function of the message and a secret key that produces a fixed-length value that serves as the authenticator

• **Hash function:** A function that maps a message of any length into a fixed length hash value, which serves as the authenticator.

## Q.3 What do you mean by MAC. Discuss working of MAC with suitable block diagram.

**Ans.** In cryptography, a message authentication Code (MAC) is a short piece of information used to authenticate a message and to provide integrity and authenticity (valid) assurances (free form doubt) on the message.

- Integrity assurance detects accidental and internal message changes.
- Authenticity assurances affirm (swear) the messages origin

- MAC, also known as a cryptographic checksum.
- It is an alternative authentication technique involves the use of a secret key to generate a small fixed size block of data, that is appended to the message.
- A MAC or cryptographic checksum, is generated by a function C of the form

**MAC=C (K, M)**

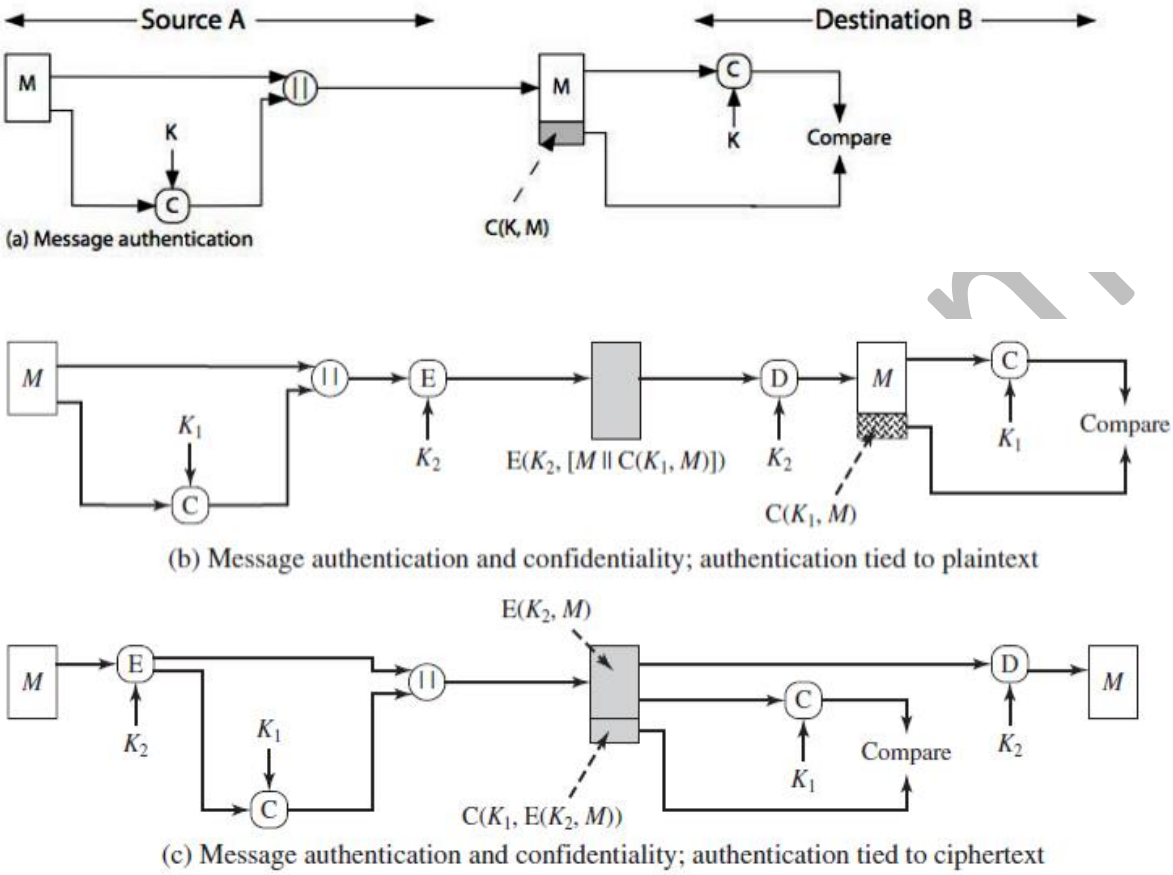
M→Input Message
C→MAC function
K→ Shared Secret Key of Communication parties (A & B).
MAC→ Message Authentication Code.

- When A has a message to send to B, The "message + MAC" are transmitted to the intended (aimed) recipient (B).
- The recipient performs the same calculation on the received message, using the same secret key, to generate a new MAC.
- The received MAC is compared to the calculated MAC.

**NOTE: A MAC function is similar to encryption; one difference is that the MAC algorithm need not be reversible, as it must for decryption.**

- MAC function is a many-to-one, potentially many messages have same MAC. That is, if message M is 100 bit message, and 10 bit MAC then there are $2^{100}$ messages and $2^{10}$ MAC are available.
- There four $2^{100}/2^{10} = 2^{90}$ different message, 5 bit key used then $2^5$=32 different mapping form the set of messages to the set of MAC values.

- In general n-bit MAC is used, then there are $2^n$ possible MAC's
  **N→possible messages with N>>$2^n$ With K-bits key, there $2^K$ possible keys.**




**Q.3 Discuss some situation where MAC can be used.**

**Ans. MAC can be used in various situation:**

1. There are a number of applications in which the same message is broadcast to a number of destinations.

2. Another possible scenario is an exchange in which one side has a heavy load and cannot afford the time to decrypt all incoming messages. Authentication is carried out on a selective basis, messages being chosen at random for checking.

3. For some applications, it may not be of concern to keep messages secret, but it is important to authenticate messages.

## Q. 4 . What is Hash Function?

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

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

**h=H (M)**

M→ variable length message
H(M)→ 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.

## Q.5 Explain Birthday atack?

**Ans**. Birthday attacks are a class of brute force techniques used in an attempt to solve a class of cryptographic hash function problem. These methods takes advantage of functions which, when supplied with a random input, return one of k equally likely values.

Suppose that a 64 bit hash code is use, if an encryption hash code C is transmitted with the corresponding unencrypted message M then an opponent would need to find an $M^1$ such that $H(M^1)=H(M)$ to substitute another message and fool the receiver. On average, the opponent would have to try about $2^{63}$ messages to find one that matches the hash code of the intercepted message.

## Q.6 Name all the hash algorithms.

**Ans. Hash Algorithms :**

1. Message Digest:MD5
2. Secure Hash Algorithm: SHA-1 (from MD4)
3. RIPEMD-160
4. HMAC

**Q.7 Explain the use of hash function for message authentication.**

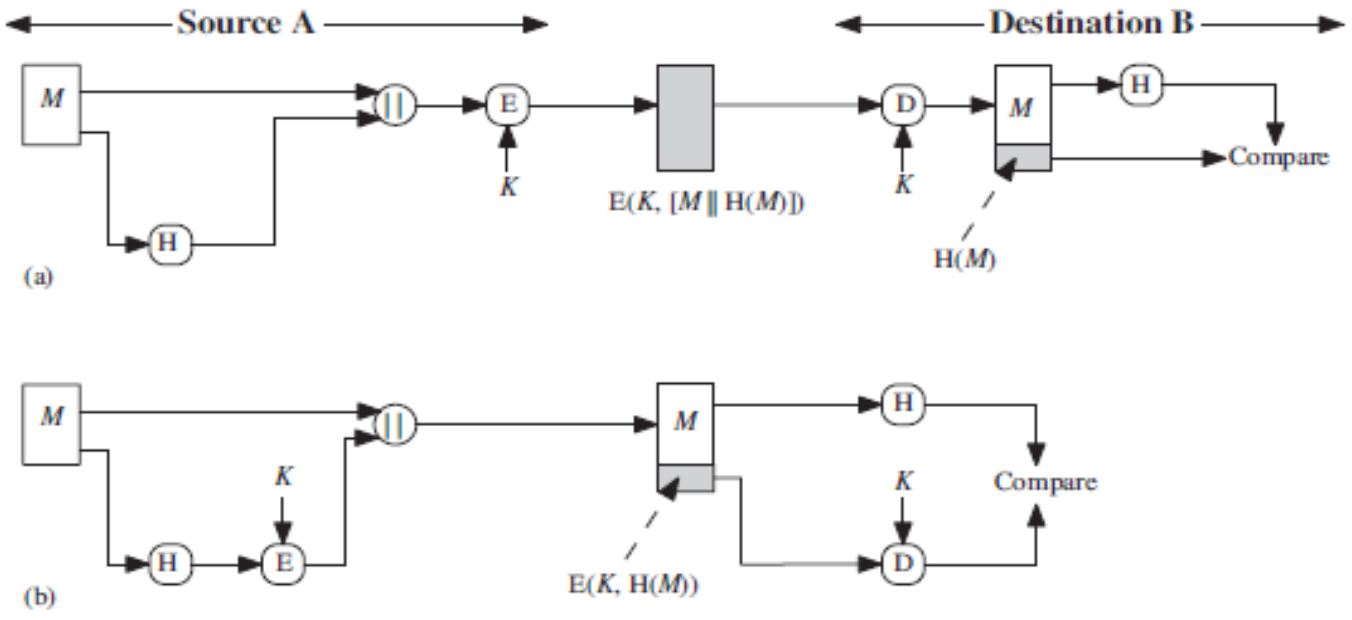
**Ans.**

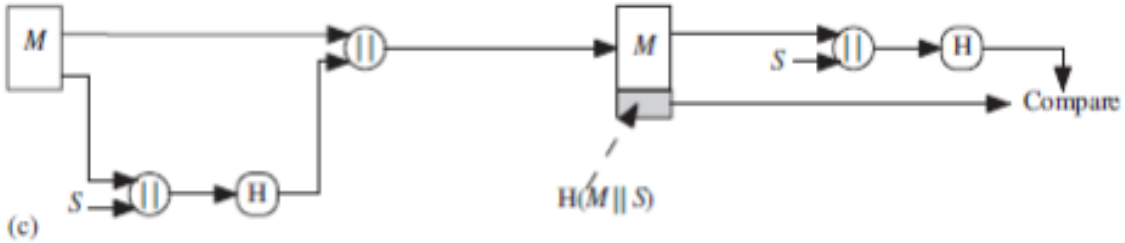
**a.** The message plus concatenated hash code is encrypted using symmetric encryption. Because only A and B share the secret key, the message must have come from A and has not been altered. The hash code provides the structure or redundancy required to achieve authentication. Because encryption is applied to the entire message plus hash code, confidentiality is also provided.

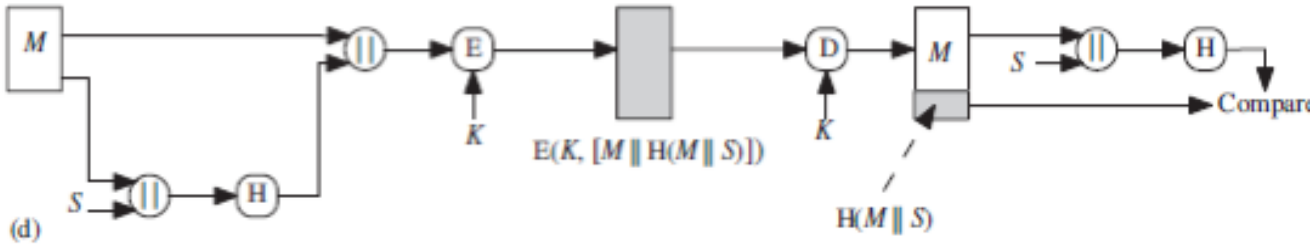
**b.** Only the hash code is encrypted, using symmetric encryption. This reduces the processing burden for those applications that do not require confidentiality

**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 recomputed 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.






**Q.8 What are the requirements of hash function?**

**Ans.** 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:

1. H van 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. 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.
5. Weak collision resistance:- for any given block x. it is computationally infeasible to find y≠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 referred as strong collision resistance

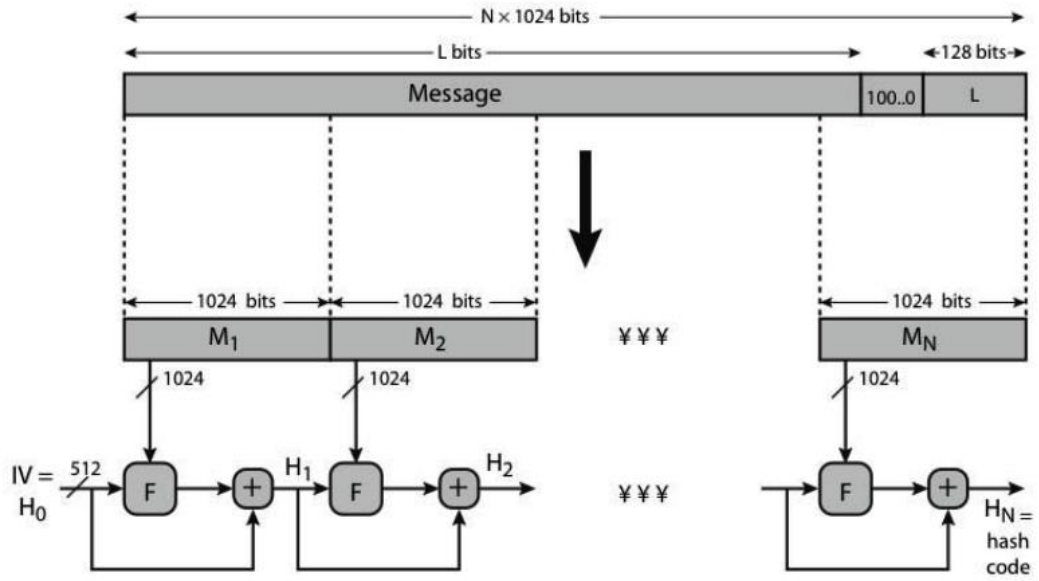
**Q.9 Discuss MD-5 Algorithm with all required steps with suitable block diagram?**

**Ans.** It was developed to avoid brute force & crypt-analytic attacks. MD5 was the most widely used secure hash algorithm.

MD5 logic:

Input:-This algorithm takes as input a message of arbitrary length.
Output:- produce a 128 bit message digest.

The input is processed in 512 bit blocks.




**Algorithm processing Steps:**

Step1: Append Padding Bits
Step 2: Append Length
Step 3: Initialize MD Buffer
Step 4: Process Message in 512 bit (16-Word) Blocks
Step 5: Output

Step-1: **Appending Padding Bits.** The original message is "padded" (extended) so that its length (in bits) is congruent to 448, modulo 512. The padding rules are:

- The original message is always padded with one bit "1" first.
- Then zero or more bits "0" are padded to bring the length of the message up to 64 bits fewer than a multiple of 512.

Step-2: **Appending Length**. 64 bits are appended to the end of the padded message to indicate the length of the original message in bytes. The rules of appending length are:

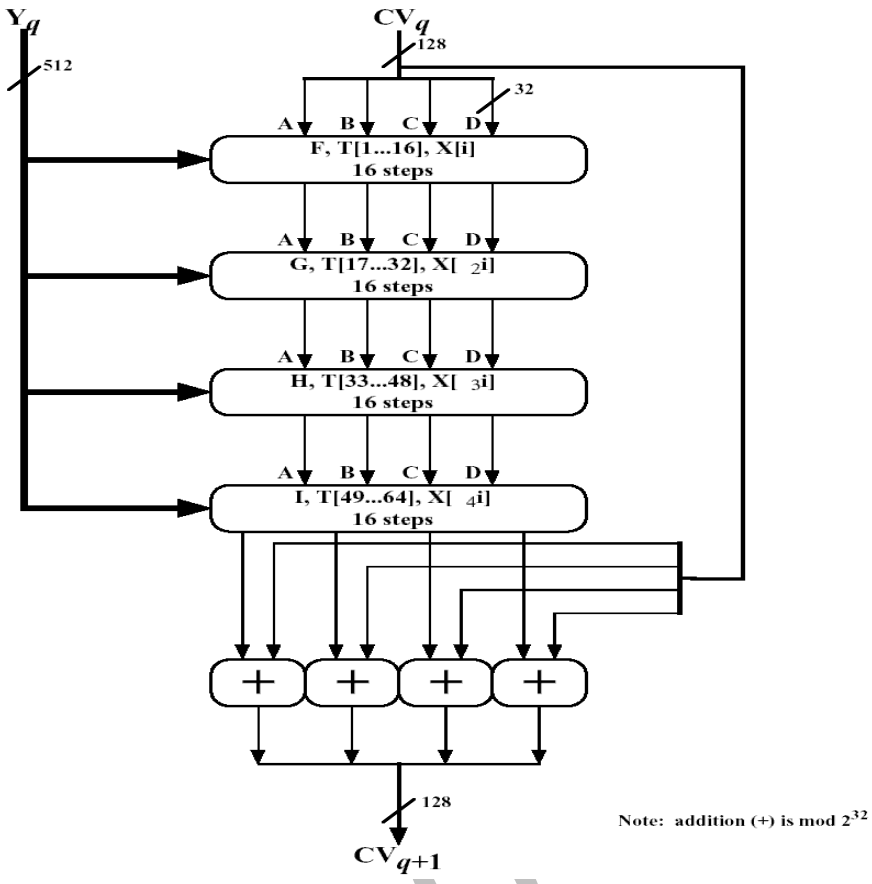
- The length of the original message in bytes is converted to its binary format of 64 bits. If overflow happens, only the low-order 64 bits are used.
- Break the 64-bit length into 2 words (32 bits each).
- The low-order word is appended first and followed by the high-order word.

Step-3: **Initializing MD Buffer**. A 128 bit buffer is used to hold intermediate and final results of the ash function. The buffer can be represented as four 32 bit registers (A, B, C, D). these registers are initialize to the following 32 bit integers (hexadecimal values)

- Word A is initialized to: 0x67452301.
- Word B is initialized to: 0xEFCDAB89.
- Word C is initialized to: 0x98BADCFE.
- Word D is initialized to: 0x10325476.

Step-4: Processing Message in 512-bit Blocks. This is the main step of MD 5 algorithm, which loops through the padded and appended message in blocks of 512 bits each. For each input block, 4 rounds of operations are performed with 16 operations in each round. The four rounds have a similar structure, but each uses a different primitive logical function, referred to as F,G,H and I in the specification.

Step-5: **output:** After all L 512 bit blocks have been processed, the output form the Lth stage is the 128 bit message digest.




**Q.10 Discuss SHA- 512 with all required steps, round function & block diagram.**

**Ans.**The **Secure Hash Algorithm** is a family of cryptographic hash functions developed by the NIST (National Institute of Standards & Technology).

Purpose: Authentication, not encryption.

SHA-1 logic:

- The algorithm takes a message with maximum of length of less than 264 bits.
- Produce output is 160 bits message digest.
- The input is processed 512 bits block.

Processed Steps:

**Algorithm processing Steps:**

Step1: Append Padding Bits

Step 2: Append Length

Step 3: Initialize MD Buffer

Step 4: Process Message in 512 bit (16-Word) Blocks
Step 5: Output

Step-1: **Appending Padding Bits.** The original message is "padded" (extended) so that its length (in bits) is congruent to 448, modulo 512. The padding rules are:

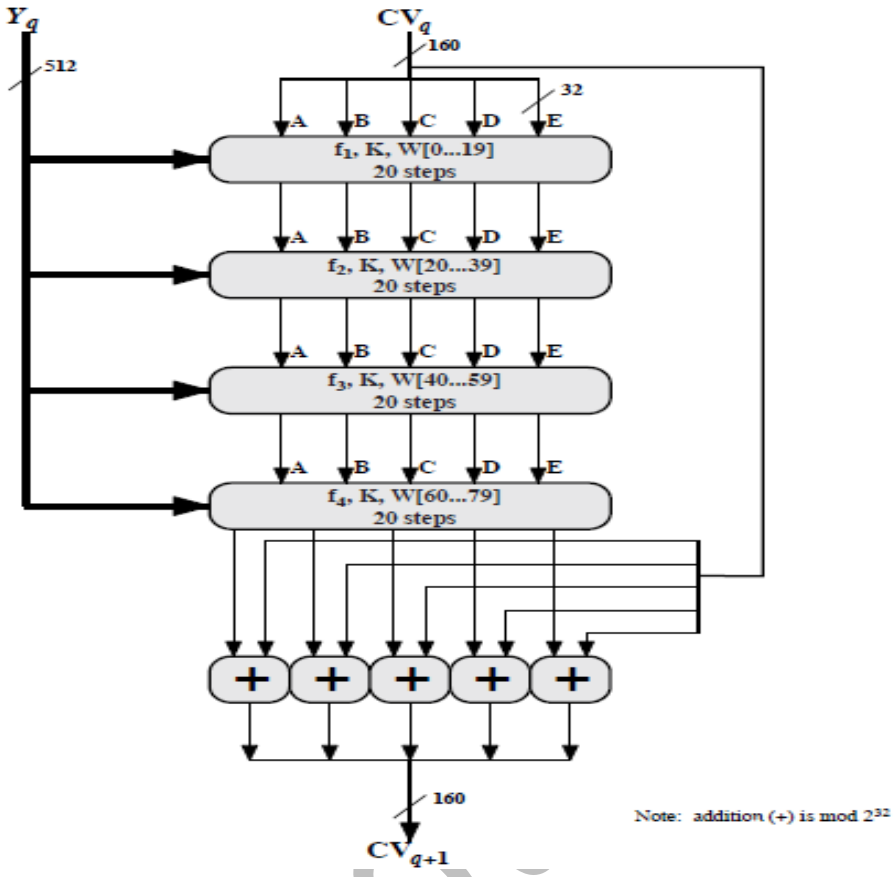
- The original message is always padded with one bit "1" first.
- Then zero or more bits "0" are padded to bring the length of the message up to 64 bits fewer than a multiple of 512.

Step-2: append length: a block of 64 bits is appended to the message. This block is treated as unsigned 64 bit integers (most significant byte first) and contains the length of the original message.

Step-3: Initialize MD buffer: 160 bit buffer is used to hold intermediate and final results of the hash function. This buffer can be represented as five 32 bit registers (A, B,C,D,E). the register are initialized to the following 32 bit integers

- Word A is initialized to: 0x67452301.
- Word B is initialized to: 0xEFCDAB89.
- Word C is initialized to: 0x98BADCFE.
- Word D is initialized to: 0x10325476.
- Word E is initialized to: 0xC3D2E1F0

Step 4: Process Message in 512 bits: this algorithm consist 4 rounds of 20 steps each. Four rounds have similar structures, but each uses a different primitive logical function, we refer it as f1, f2, f3 and f4. Each round takes input the current 512 bit blocks being processed ( $Y_q$ ) and the 160 bit buffer value a ABCDE and updates the contents of the buffer. Each round also make use of an additive constant $K_t$ where $0 \leq t \leq 79$ indicates one of the 80 steps across four rounds.
The output of $4^{th}$ round added to the input to the $1^{st}$ round (CVq) to produce CVq+1.




Step-5: output: after all L 512 bits block have been processed, the output from the Lth stage is the 160 bit message digest.

**The behavior of SHA-1can be summarized as:**

CV0=IV

**CVq+1=SUM32 (CVq, ABCDEq)**

*MD=CVL*

IV→ initialize value of the ABCDE buffer define in step-3

ABCDEQ→ output of last round of qth message block.

L→number of block (512 bit) in message

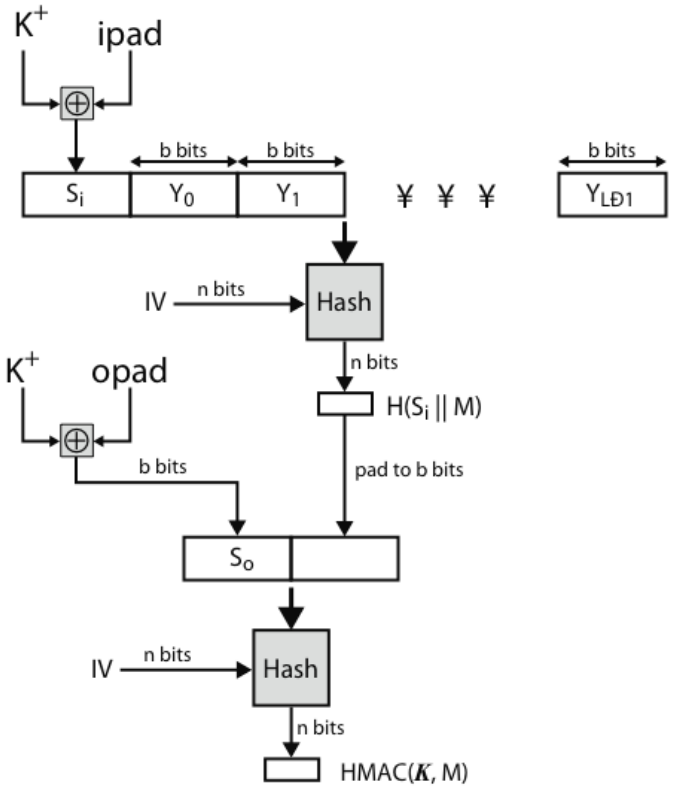
SUM32→addition modulo $2^{32}$

MD→ final message Digest Value.

## Q.11 Discuss HMAC in detail.

**Ans.** HMAC is a special construction for calculating MAC (message authentication code) involving a cryptographic hash function in combination with a secret cryptographic key.

- MAC is used to simultaneously verify both the data integrity and the authentication of message.
- Hash function such as MD5 or SHA-1 may be used in calculation of an HMAC, the resulting MAC algorithm is termed HMAC-MD5 or HMAC_SHA-1.
- The strength of HMAC depends upon the underlying hash function, the size of its hash output and the size of the quality of the key.



**Q.12. Compare the MD5 and SHA algorithm.**
**Ans.**

| | **MD5** | **SHA-1** |
|---|---|---|
| **Digest Length** | 128 bits | 160 bits |
| **Basic unit of Processing** | 512 bits | 512 bits |
| **Number of Steps** | 64 (4 rounds of 16) | 80(4 rounds of 20) |
| **Maximum Message Size** | ∞ | 264-1 bits |
| **Primitive logical functions** | 4 | 4 |
| **Additive constants used** | 64 | 4 |
| **Endianness** | Little endian | Big endian |

**Q.13 Explain Digital Signature. Discuss signing & verifying process of Digital Signature**

**Ans.** A digital signature or digital signature scheme is a mathematical scheme for demonstration the authenticity of digital message or document.

- Means, a digital signature is an authentication mechanism that enables the creator of a message to attach a code that act as a signature.
- This signature is formed by taking the hash of the message and encrypting the message with the creator's private key. The signature guarantees the source and integrity of the message.
- The digital signature standard (DSS) is an NIST standard that uses the secure hash algorithm (SHA).

**Where it used:**

Message authentication protects two parties who exchange message from any third party. But it does not protect the two parties against each other.

Example: Suppose that john sends an authenticated message to marry, using any authentication scheme (symmetric or public key cryptography). There are two disputes (clash or fight or arguments) that could arise.

1. Mary may forge a different message & claim that it came from John. Means, Mary would simply have to create a message & append an authentication code using the key, which John and Mary share.

2. John can deny sending the message Because it is possible for mary to forge that john did n fact send the message.

**Properties of Digital Signature:**

• It must verify the author and the date and time of the signature.
• It must authenticate the contents at the time of the signature.
• It must be verifiable by third parties, to resolve disputes.

**Digital Signature Requirements**

1. The signature must be a bit pattern that depends on the message being signed.
2. The signature must use some information unique to the sender to prevent both forgery and denial.
3. It must be relatively easy to produce the digital signature.
4. It must be relatively easy to recognize and verify the digital signature.
5. It must be computationally infeasible to forge a digital signature, either by constructing a new message for an existing digital signature or by constructing a fraudulent digital signature for a given message.
6. It must be practical to retain a copy of the digital signature in storage.

**Approaches for Digital Signature:**

1. Direct Digital Signature
2. Arbitrated Digital Signature

**Direct Digital Signature**:

- The term **direct digital signature** refers to a digital signature scheme that involves only the communicating parties (source, destination).
- The validly of scheme depends on the security of the sender's private key.
- The sender later wishes to deny sending a particular message by claiming the private key was lost or stolen or some other reason.
- There is chance in stole the private key of a sender at some time T.

**Arbitrated Digital Signature:**

- In this every signed message from a sender X to a receiver Y goes first to an arbiter A, who subjects the message and its signature to a number of tests to check it origin and content. The message is then dated and sent to Y.
- This process is an indication that has been verified to the satisfaction of the arbiter.

- By this process, it solves the direct Digital signature problem.

Sender X,
Arbiter A,
Receiver Y,

- X→ construct message M and compute hash value H(M) then X transmitted "M+ Digital Signature" to A.
  Signature consists→ identity "IDx of X +hash value" of all encrypted using $K_{XA}$ (it is common shared key between Sender X and Arbiter A).

- A→ A decrypts the signature & checks the hash value to validate the message. Then transmit it to Y by encryption it with $K_{AY}$ (it is common shared key between Arbiter A and Receiver Y). the message include IDx and M & time Spam.
- 
- Y→ Decrypt it by using $K_{AY}$
  X→A: $M||E_{KXA}[IDx||H(M)]$
  A→Y: $E_{KAY}[IDx||M||\ E_{KXA}[IDx||H(M)]||T]$

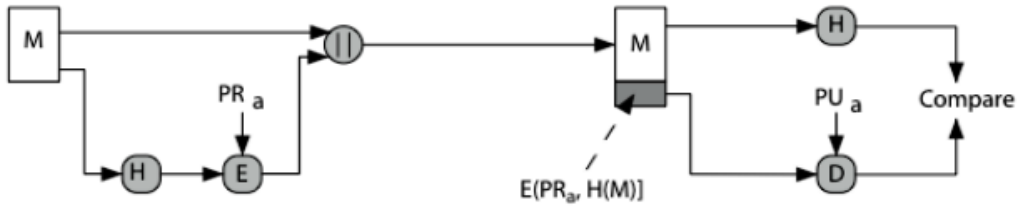
**Q.14 Discuss the algorithm (DSA) in detail with suitable steps.**

**Ans.** The National Institute of Standards and Technology (NIST) has published Federal Information Processing Standard FIPS 186, known as the Digital Signature Standard (DSS). The DSS makes use of the Secure Hash Algorithm (SHA) presents a new digital signature technique, the Digital Signature Algorithm (DSA).

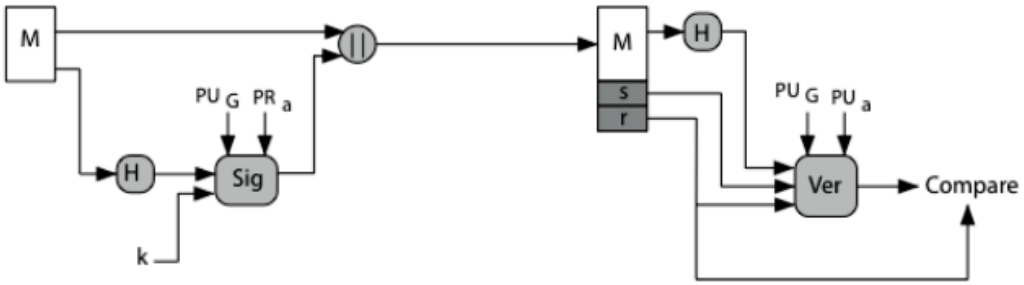
**The DSS Approach:**

The DSS uses an algorithm that is designed to provide only the digital signature function. Unlike RSA, it cannot be used for encryption or key exchange. Nevertheless, it is a public-key technique.

The DSS approach also makes use of a hash function. The hash code is provided as input to a signature function along with a random number generated for this particular signature. The signature function also depends on the sender's private key (*PRa*) and a set of parameters known to a group of communicating principals. We can consider this set to constitute a global public key ($PU_G$).




(a) RSA Approach




(b) DSS Approach

**Q.15 What are the properties of hash functions? Explain what characteristics are needed in a secure Hash Function. How MAC is differs from hash function and Digital signature.**

**Ans.** Properties of hash functions, consider a hash function H

1. Performance: Easy to compute H(m)
2. One-way property: Given H(m) but not m, it's computationally infeasible to find m
3. Weak collision resistance (free): Given H(m), it's computationally infeasible to find m' such that **H(m') = H(m).**
4. Strong collision resistance (free): Computationally infeasible to find m1, m2 such that **H(m1) = H(m2)**

**Characteristics are needed in a secure Hash Function are:**

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 $y \neq x$ with H(y) = H(x).
6. It is computationally infeasible to find any pair (x, y) such that H(x) = H(y).

| Security Goal | Hash | MAC | Digital Signature |
|---|---|---|---|
| Integrity | Yes | Yes | Yes |
| Authenticity | No | Yes | Yes |
| Non-repudiation | No | No | Yes |
| Key used | Normally no keys | Symmetric | *Asymmetric |