UNIT - III

Hash Functions and Digital signatures :-

UNIT-III

HASH FUNCTIONS AND DIGITAL SIGNATURE

Authentication requirements :-

* Disclosure
* Traffic analysis
* Masquerade
* Content modification
* Sequence "
* Timing "
* Source repudiation
* Destination "

Message Authentication Function :-

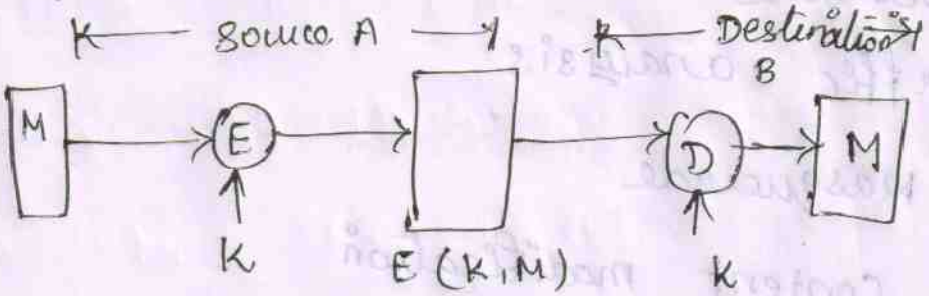
* Hash function
* Message encryption
* Message authentication code (MAC).

Message Encryption :-

Symmetric message encryption :-

* Encryption provides authentication
* Receiver know sender must have created it.

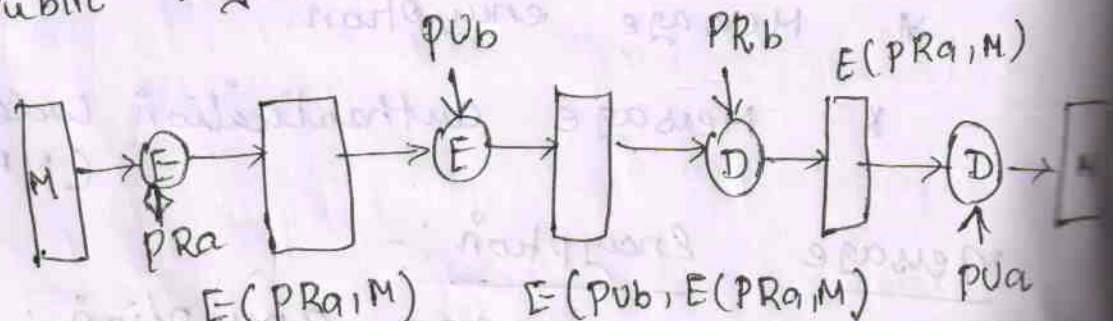
* know content cannot be altered.
* If msg has suitable structure, redundancy (or) a checksum to detect any changes.



confidentiality & authentication.

Public-key Message Encryption :-

* Encryption provides no confidence of sender. * Sender signs msg using their private key. then encrypts with recipients public key.



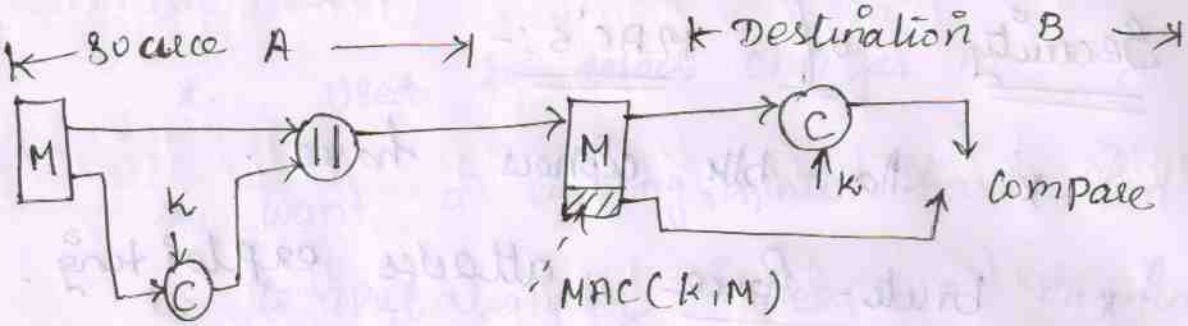
confidentiality, authentication & signature.

Message Authentication code (MAC):-

* A small fixed-sized blk of data.

* MAC = C(K, M)

* appended to msg when sent.



Why use a MAC?-

* Sometimes only authentication is needed.

* Sometimes need authentication to persist longer than the encryption.

Ex: archival use.

* MAC is not a digital signature.

Properties:-

* a MAC is a cryptographic checksum

$$MAC = C_K(M)$$

* Many-to-one fun.

Requirements:-

* Knowing a msg & MAC, is infeasible to find another msg with same MAC.

* MAC should be uniformly distributed
* MAC should depend equally on all bits of the msg.

Security of MAC's:-

* Na blk ciphers have:
* brute-force attacks exploiting.
  > Strong collision resistance hash have cost $2^{m/2}$.
  (i) 128 bit hash looks vulnerable, 160-bits better.
  > MAC's with known msg-MAC pairs
  (i) Can either attack keyspace (or) MAC. At least 128-bit MAC is needed for security.

* Cryptanalytic attacks exploit structure like blk ciphers want brute-force attacks to be the best alternative.
* More variety of MACs so hard to generalize abt cryptanalysis.

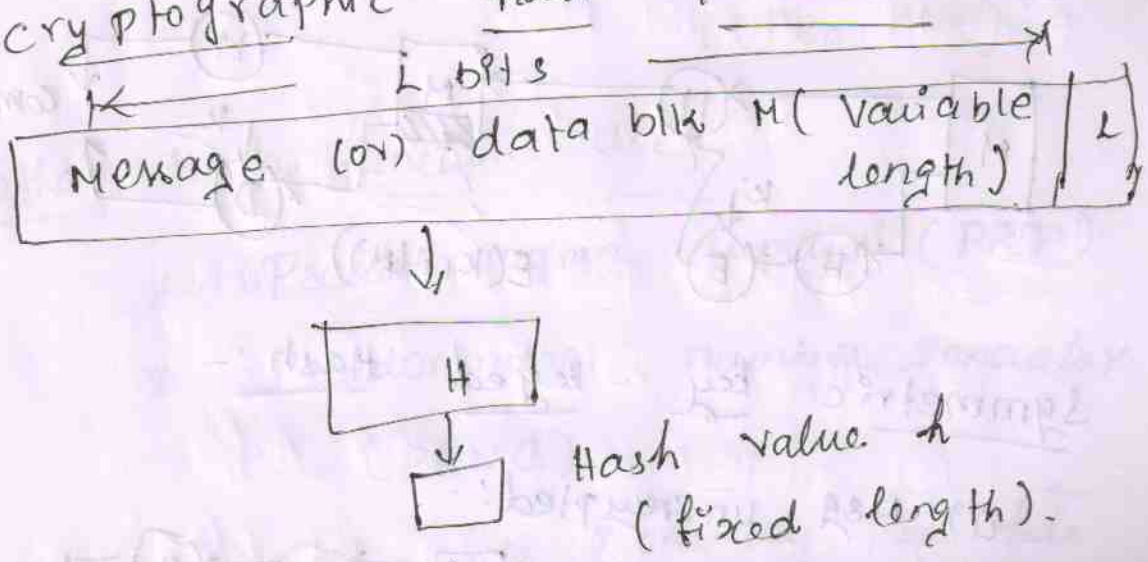
Hash Functions

* Condenses arbitrary msg to fixed size

$$h = H(M)$$

* Assume hash fun is public.
* Used to detect changes to msg.
* Want a cryptographic hash fun.
  - ↳ computationally infeasible to find data mapping to specific hash. (one-way property).
  - ↳ computationally infeasible to find two data to same hash - (collision-free property)

cryptographic hash function :-



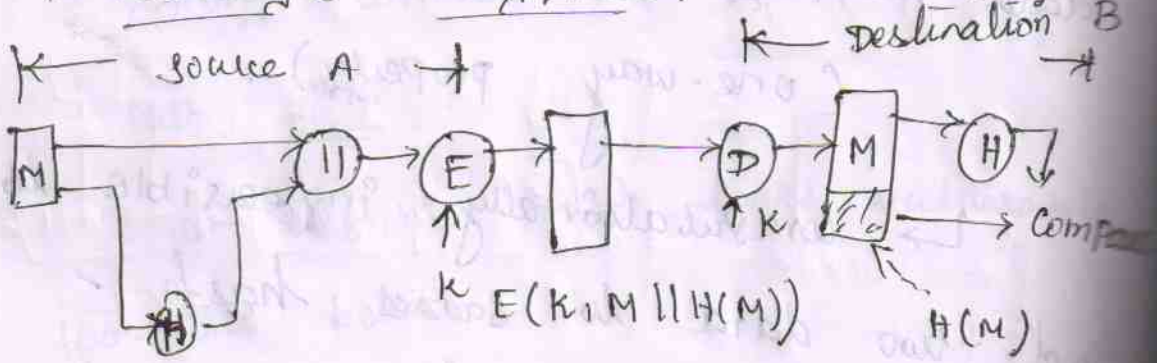
Hash function Uses :-

* Msg Integrity check (MIC)
* Msg Authentication code (MAC)
* Digital signature (non-repudiation)

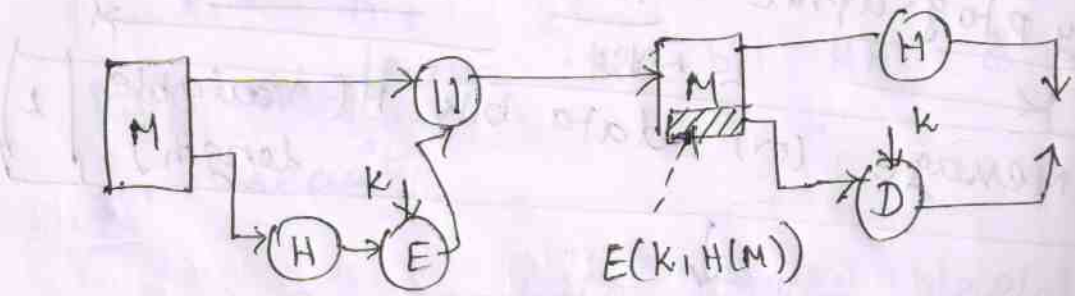
Hash functions & Message Authentication

Symmetric key Un-keyed Hash :-

a) Message encrypted.

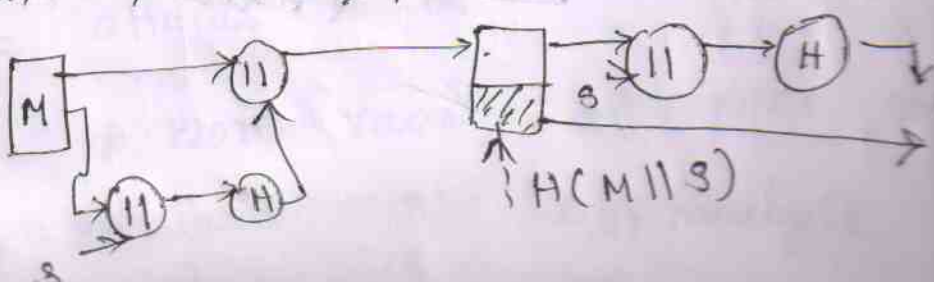


b) Message unencrypted :-

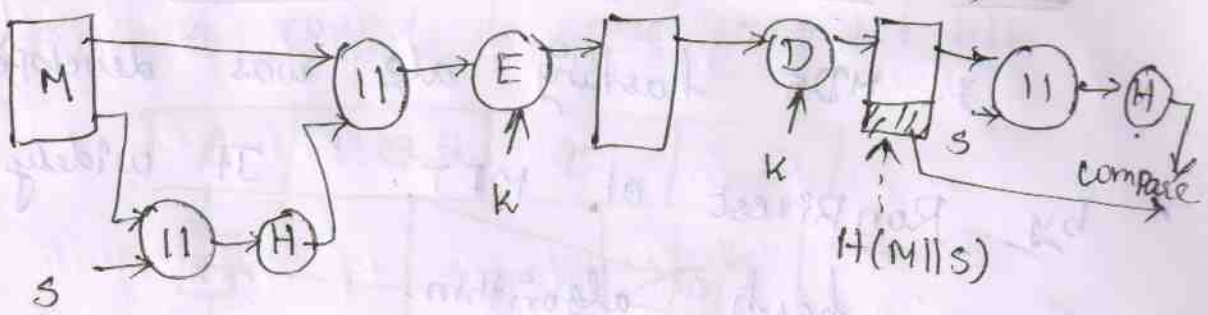


Symmetric key - keyed Hash :-

a) Msg unencrypted :-

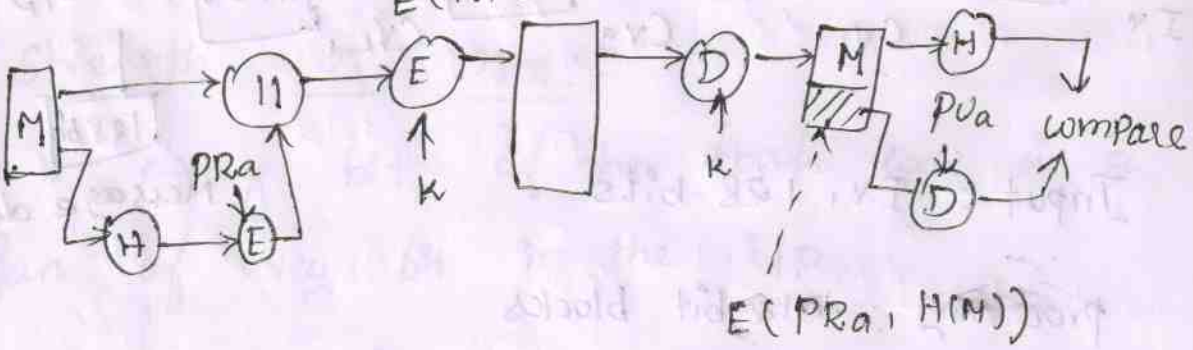


b) Msg Encrypted :- $E(k, [M \| H(M \| S)])$ (8)



Hash Functions & Digital signatures - PKCS:-

$E(K, [M \| E(PRa, H(M))])$



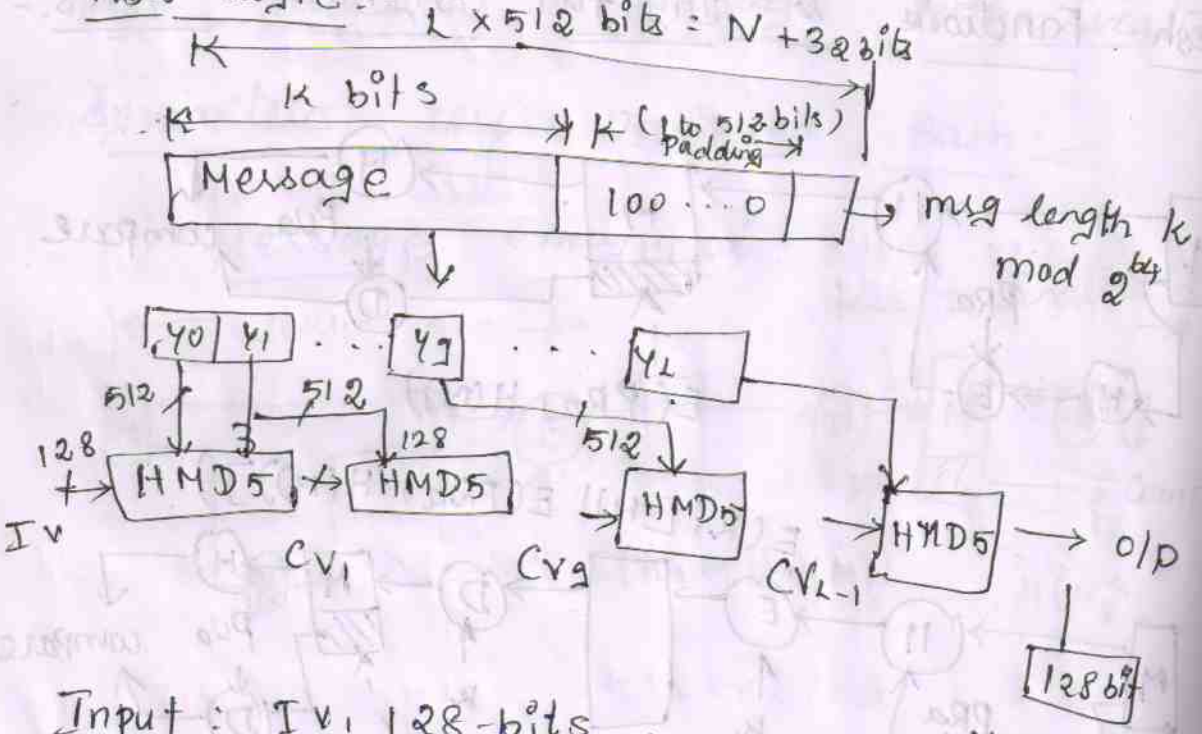
Other Hash function uses :-

* Pseudorandom function (PRF)
* Pseudorandom number Generator (PRNG).
* To create a one-way Pwd file
* For intrusion detection & virus detection.

MD5 - [Message Digest Algorithm] :-

*. MD5 hashing alg was developed by Ron Rivest at MIT. It widely used secure hash algorithm.

MD5 Logic :-



Input : IV, 128-bits

processing : 512 bit blocks

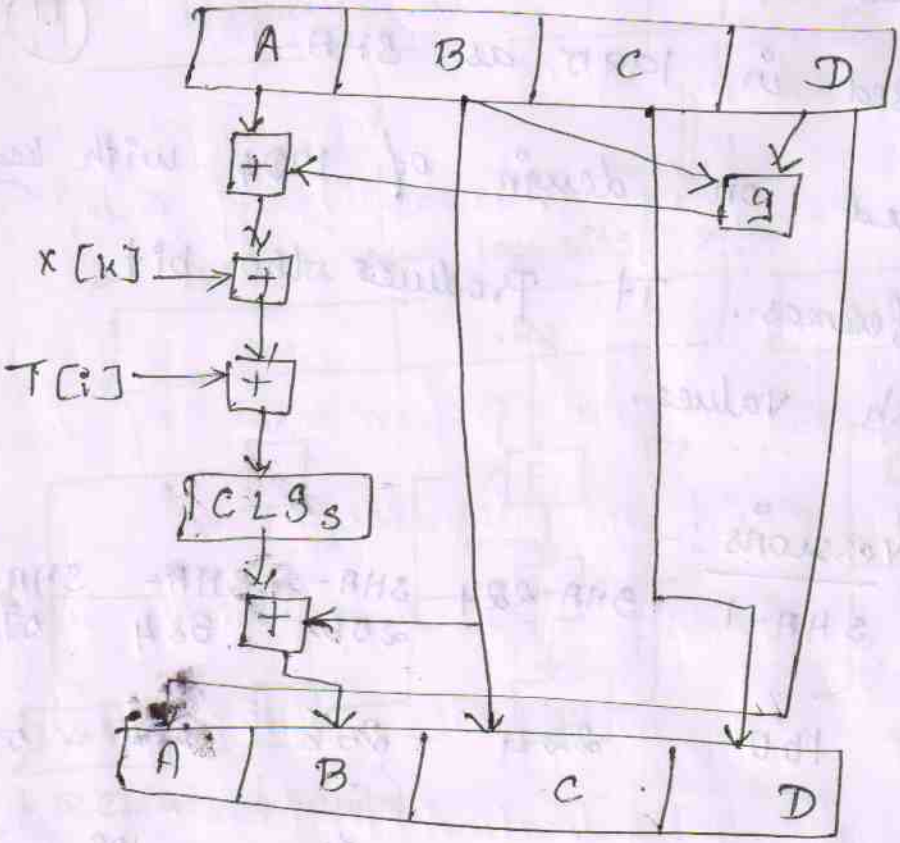
output : 128 bit msg digest

Steps :

1. Append padding bits
2. Append length
3. Initialize the MD Buffer
4. processing 512 Bit Block.

MD5 Compression function :-

* A round to one 512 bit blk



Strength of MD5 :-

* Every bit of the hash code is a fun of every bit in the i/p.

* when 2 msg choosen at random will not have the same hash code.

Attacks on MD5 :-

* Differential cypt analysis.

* Pseudo collision