IV	Authentication requirements- Authentication functions- Message authentication codes- Hash functions- SHA -1, MD5, Security of Hash functions and MACs- Authentication protocols-Digital signatures-Digital signature standards.

Message Authentication

Security Requirements

1. Disclosure
2. Traffic analysis
3. Masquerade
4. Content modification
5. Sequence modification
6. Timing modification
7. Source repudiation
8. Destination repudiation

Security Requirements

1. Disclosure
2. Traffic analysis

Message Confidentiality

3. Masquerade
4. Content modification
5. Sequence modification
6. Timing modification
7. Source repudiation
8. Destination repudiation

Security Requirements

1. Disclosure
2. Traffic analysis
3. Masquerade
4. Content modification
5. Sequence modification
6. Timing modification
7. Source repudiation
8. Destination repudiation

Message Authentication

Security Requirements

1. Disclosure
2. Traffic analysis
3. Masquerade
4. Content modification
5. Sequence modification
6. Timing modification
7. Source repudiation
8. Destination repudiation

Digital Signature

Security Requirements

1. Disclosure
2. Traffic analysis

Message Confidentiality

3. Masquerade
4. Content modification
5. Sequence modification
6. Timing modification
7. Source repudiation
8. Destination repudiation

Message Authentication

Digital Signature

- **Authentication Requirements**

- In the context of communications across a network, the following attacks can be identified:

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

- **Traffic analysis:** Discovery of the pattern of traffic between parties. In a connection-oriented application, the frequency and duration of connections could be determined. In either a connection-oriented or connectionless environment, the number and length of messages between parties could be determined.

- **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. Also included are fraudulent acknowledgments of message receipt or non receipt by someone other than the message recipient

- **Content modification:** Changes to the contents of a message, including insertion, deletion, transposition, and modification.
- **Sequence modification:** Any modification to a sequence of messages between parties, including insertion, deletion, and reordering.
- **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. In a connectionless application, an individual message (e.g., datagram) could be delayed or replayed.
- **Source repudiation:** Denial of transmission of message by source.
- **Destination repudiation:** Denial of receipt of message by destination.

Message Authentication

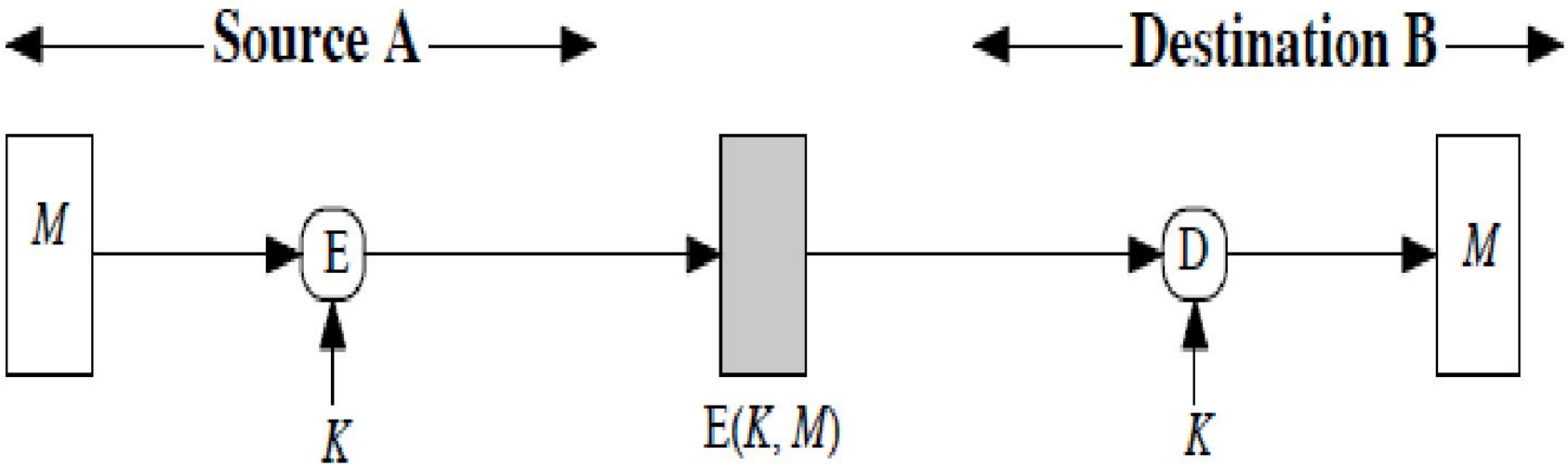
- A procedure to verify that received messages come from alleged source and have not been altered.
- It also verify sequencing and timelines.
- Digital Signature is an authentication technique
  - Also used to counter repudiation by the source

- 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 by (i.e., contain no modification, insertion, deletion, or replay) and that the purported identity of the sender is valid.

Authentication Functions

- Message encryption
- Message authentication code (MAC)
- Hash functions

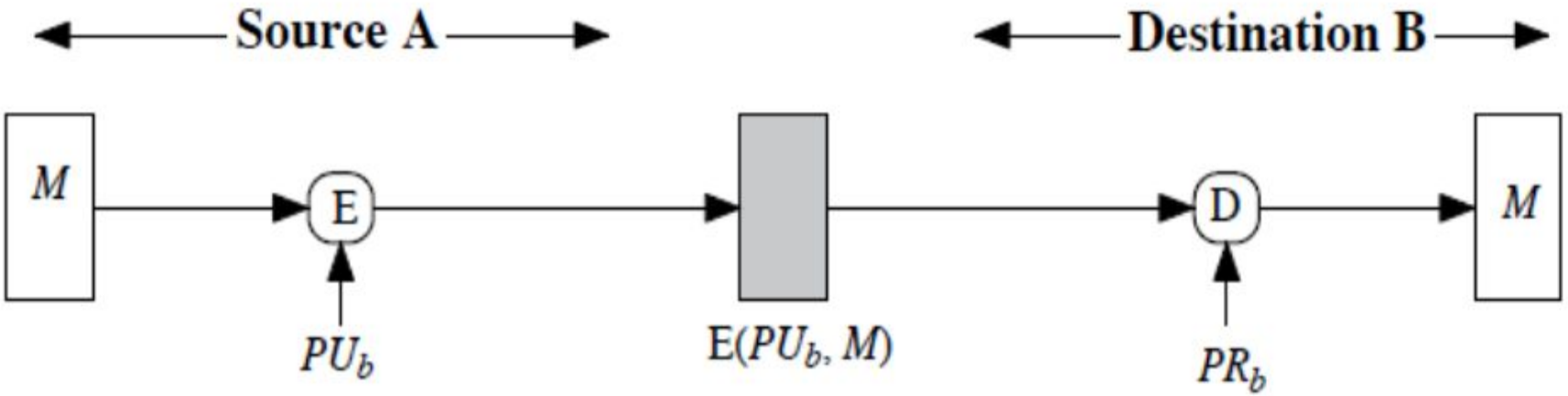
Message Encryption



(a) Symmetric encryption: confidentiality and authentication

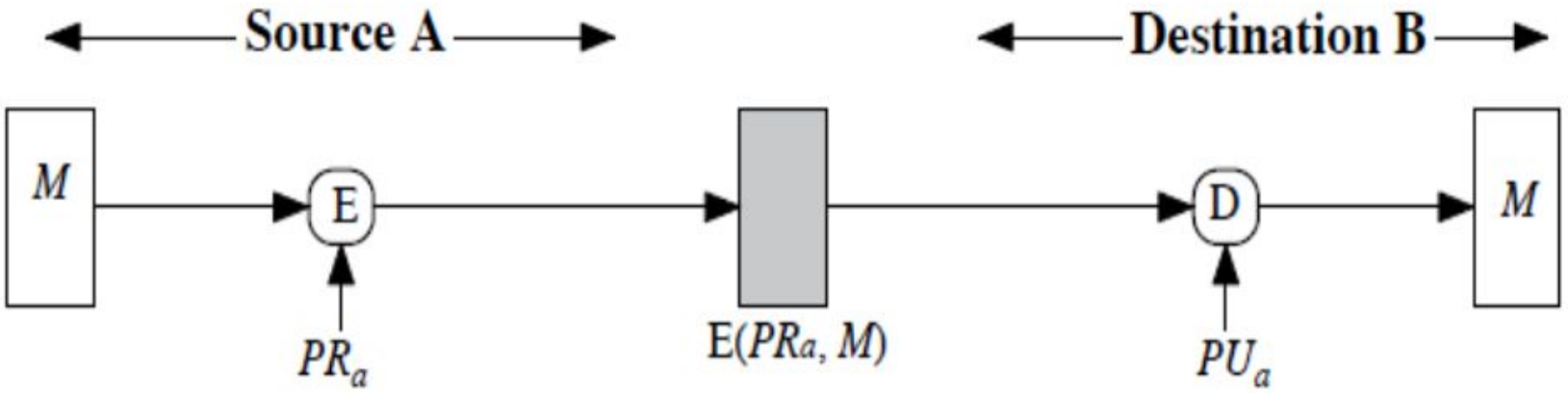
Consider a Message M is transmitted from source A to destination B.it is encrypted using key K, which is shared by A and B. if no one else knows the key confidentiality is provided.

Message Encryption



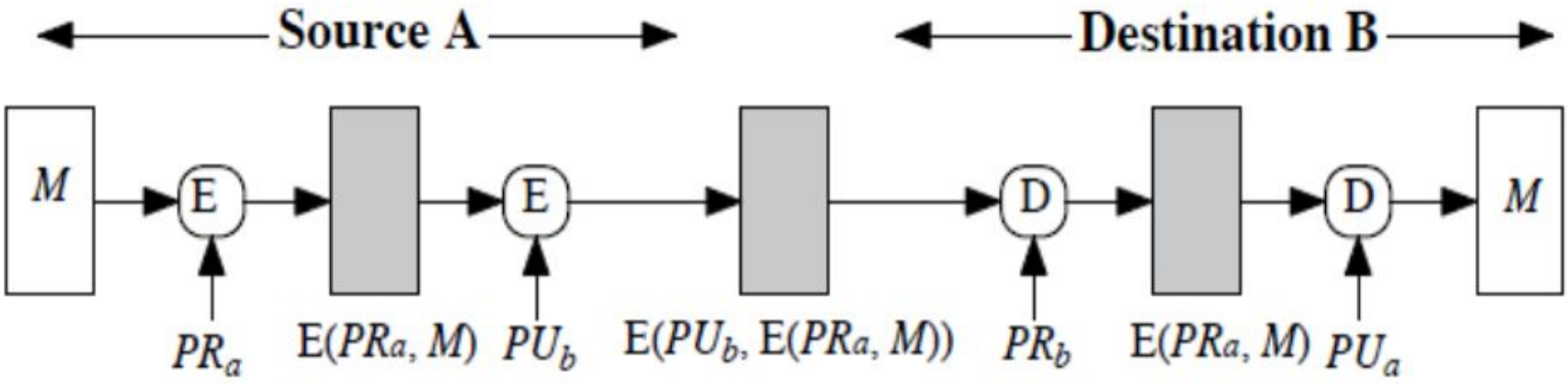
(b) Public-key encryption: confidentiality

Message Encryption



(c) Public-key encryption: authentication and signature

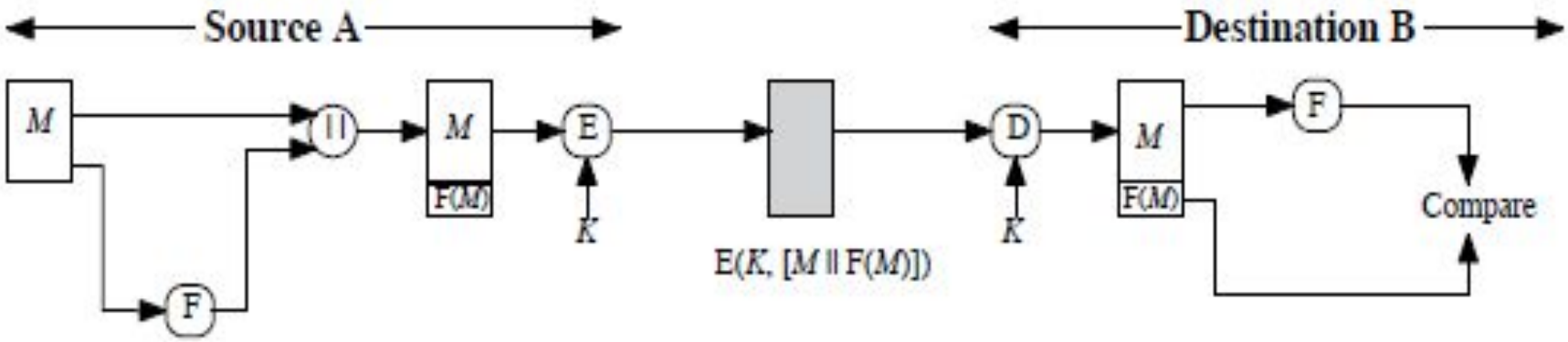
Message Encryption



(d) Public-key encryption: confidentiality, authentication, and signature

- To provide authentication, A uses its private key to encrypt the message and B uses' A's public key to decrypt the message. It ensures that the message has come from A because A is the only party that possess PRa and therefore the only party with the information necessary to construct cipher text that can be decrypted with PUa.
- To provide both confidentiality and authentication, A can encrypt M first using its private key, which provides the digital signature, and then using B's public key which provides confidentiality.

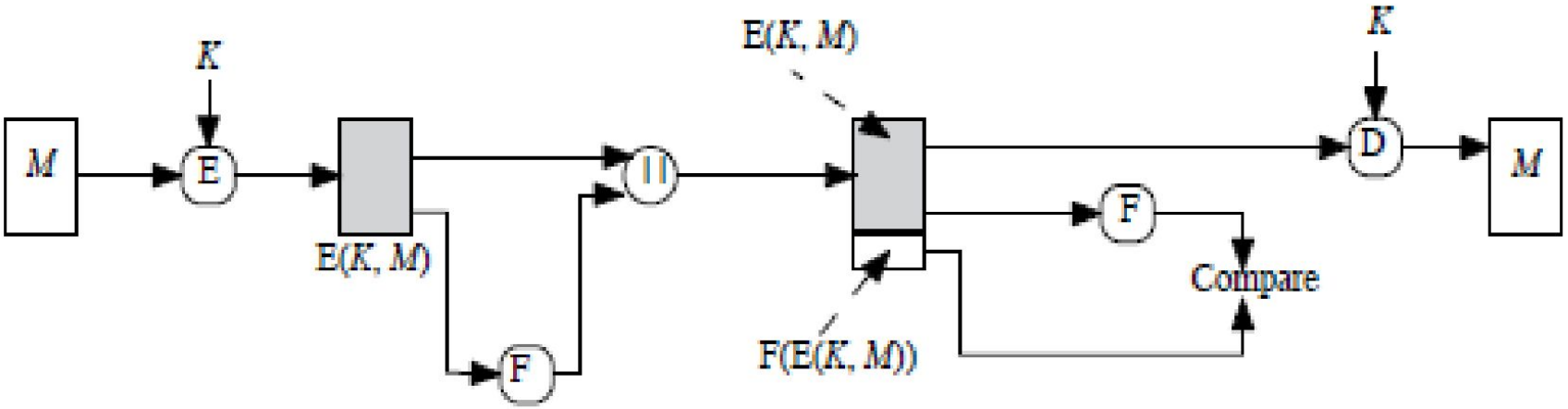
Message Encryption With Error Control



(a) Internal error control

- In order to ensure authentication, append an error-detecting code, also known as a frame check sequence (FCS) or checksum, to each message before encryption.
- A prepares a plaintext message *M* and then provides this as input to a function F that produces an FCS. The FCS is appended to *M* and the entire block is then encrypted.
- At the destination, B decrypts the incoming block and treats the results as a message with an appended FCS. B applies the same function F to attempt to reproduce the FCS. If the calculated FCS is equal to the incoming FCS, then the message is considered authentic.

Message Encryption With Error Control



(b) External error control

Message Authentication Code

Message Authentication Code (MAC)

- An authentication technique that involves the use of secret key to generate a small fixed-size block of data, known as cryptographic checksum or MAC that is appended to the message.
- When A has a message to send to B, it calculates the MAC as a function of the message using a shared secret key :

$$MAC = C_K(M).$$

M = input message ,C = MAC function ,K shared secret key

- MAC = message authentication code

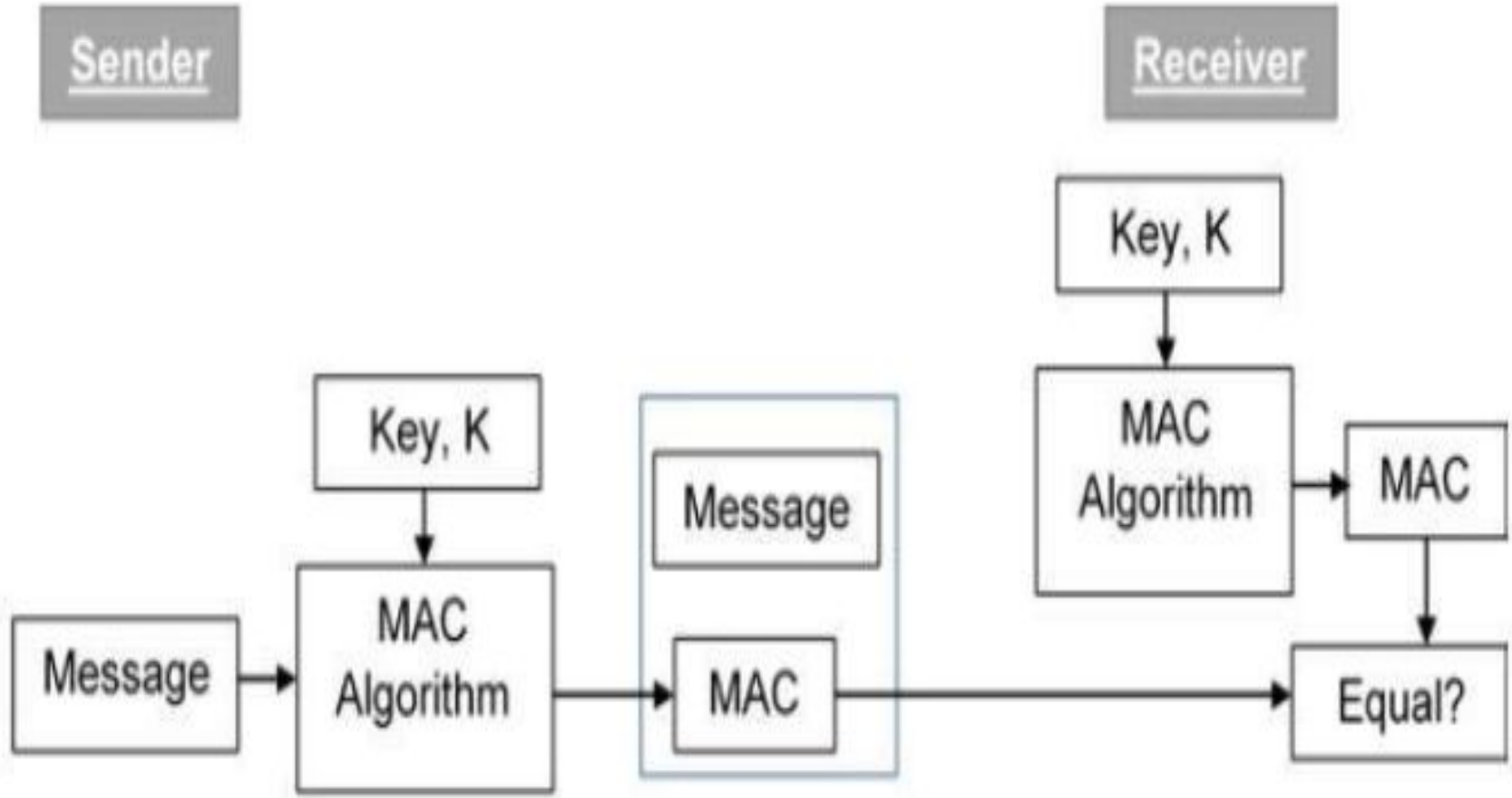
- The message plus MAC are transmitted to the intended recipient. The recipient performs the same calculation on the received message, using the same secret key to generate a new MAC. The received MAC is compared to the calculated MAC. If we assume that only the receiver and the sender know the identity of the secret key, and if the received MAC matches the calculated MAC then

- The receiver is assured that the message has not been altered. If an attacker alters the message but does not alter the MAC, then the receiver's MAC will differ from the received MAC.

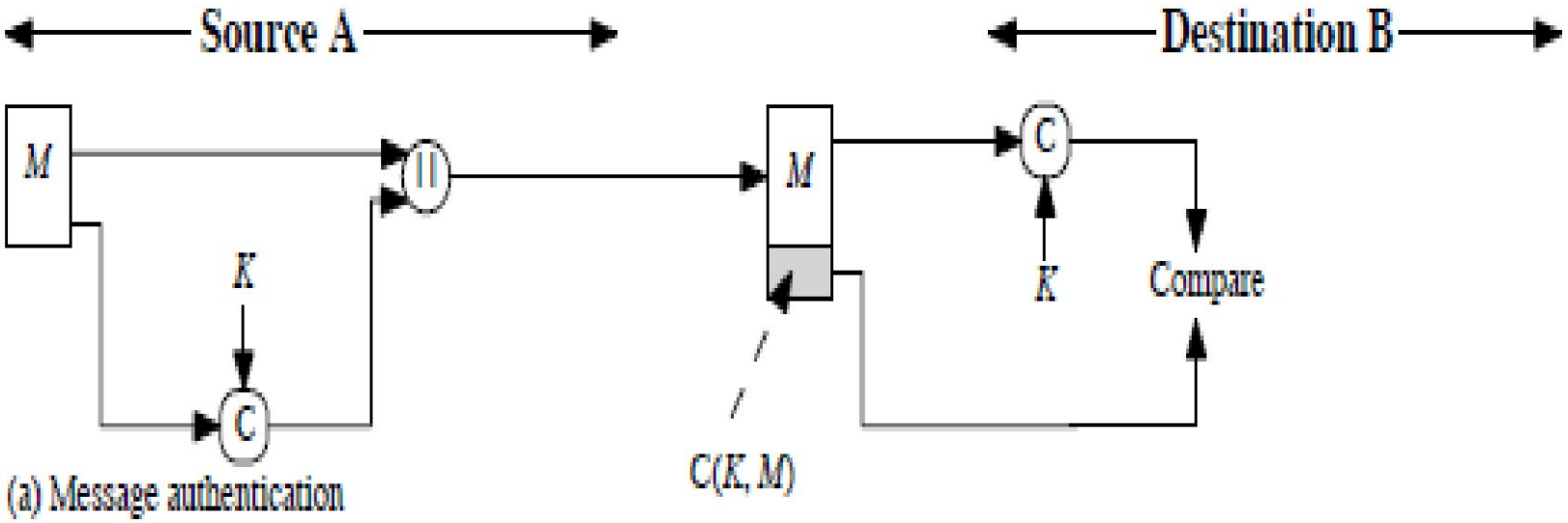
Because the attacker is assumed not to know the secret key, the attacker cannot alter the MAC to correspond to the alterations in the message.

- The receiver is assured that the message is from the alleged sender. Because no one else knows the secret key, no one else could prepare a message with a proper MAC.

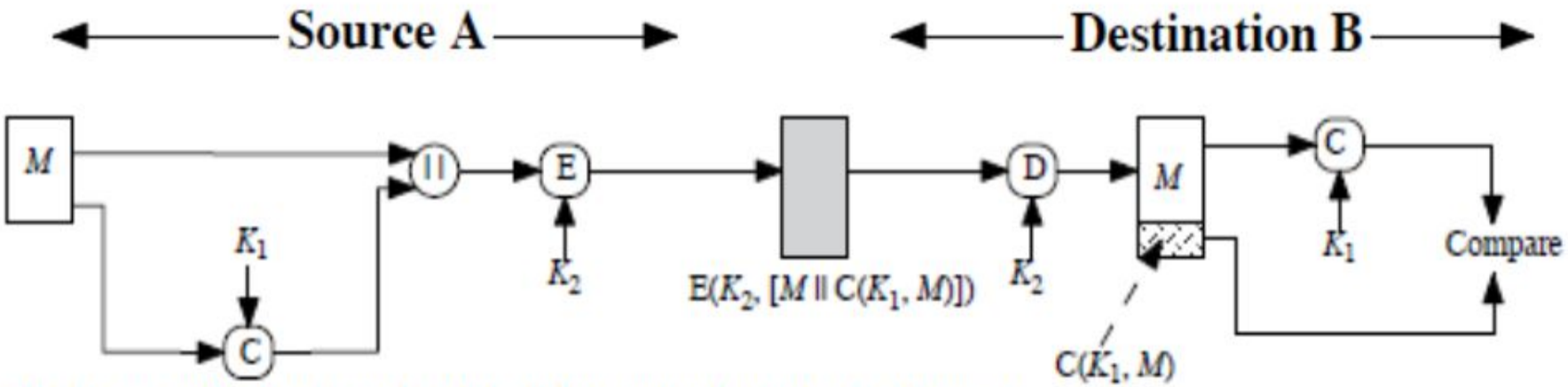
- If the message includes a sequence number, then the receiver can be assured of the proper sequence because an attacker cannot successfully alter the sequence number.


Sender
Receiver
Key, K
Key, K
MAC Algorithm
MAC
Message
Message
MAC Algorithm
MAC
Equal?

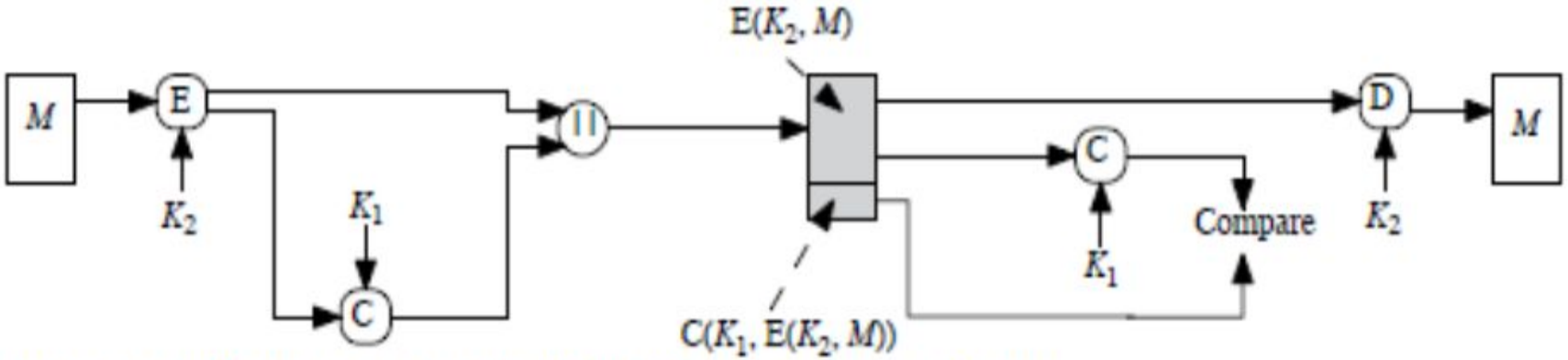
Message Authentication Code



Message Authentication Code



(b) Message authentication and confidentiality; authentication tied to plaintext



(c) Message authentication and confidentiality; authentication tied to ciphertext

- Let us now try to understand the entire process in detail −
- The sender uses some publicly known MAC algorithm, inputs the message and the secret key K and produces a MAC value.
- Similar to hash, MAC function also compresses an arbitrary long input into a fixed length output. The major difference between hash and MAC is that MAC uses secret key during the compression.
- The sender forwards the message along with the MAC. Here, we assume that the message is sent in the clear, as we are concerned of providing message origin authentication, not confidentiality. If confidentiality is required then the message needs encryption.

- In receipt of the message and the MAC, the receiver feeds the received message and the shared secret key K into the MAC algorithm and re-computes the MAC value.
- The receiver now checks equality of freshly computed MAC with the MAC received from the sender. If they match, then the receiver accepts the message and assures himself that the message has been sent by the intended sender.
- If the computed MAC does not match the MAC sent by the sender, the receiver cannot determine whether it is the message that has been altered or it is the origin that has been falsified. As a bottom-line, a receiver safely assumes that the message is not the genuine.

Message Authentication Code

- If the received MAC matches the calculated MAC, then the receiver is assured that.
  1. The message is not altered.
  2. The message is from alleged sender.
  3. The message is in proper sequence.

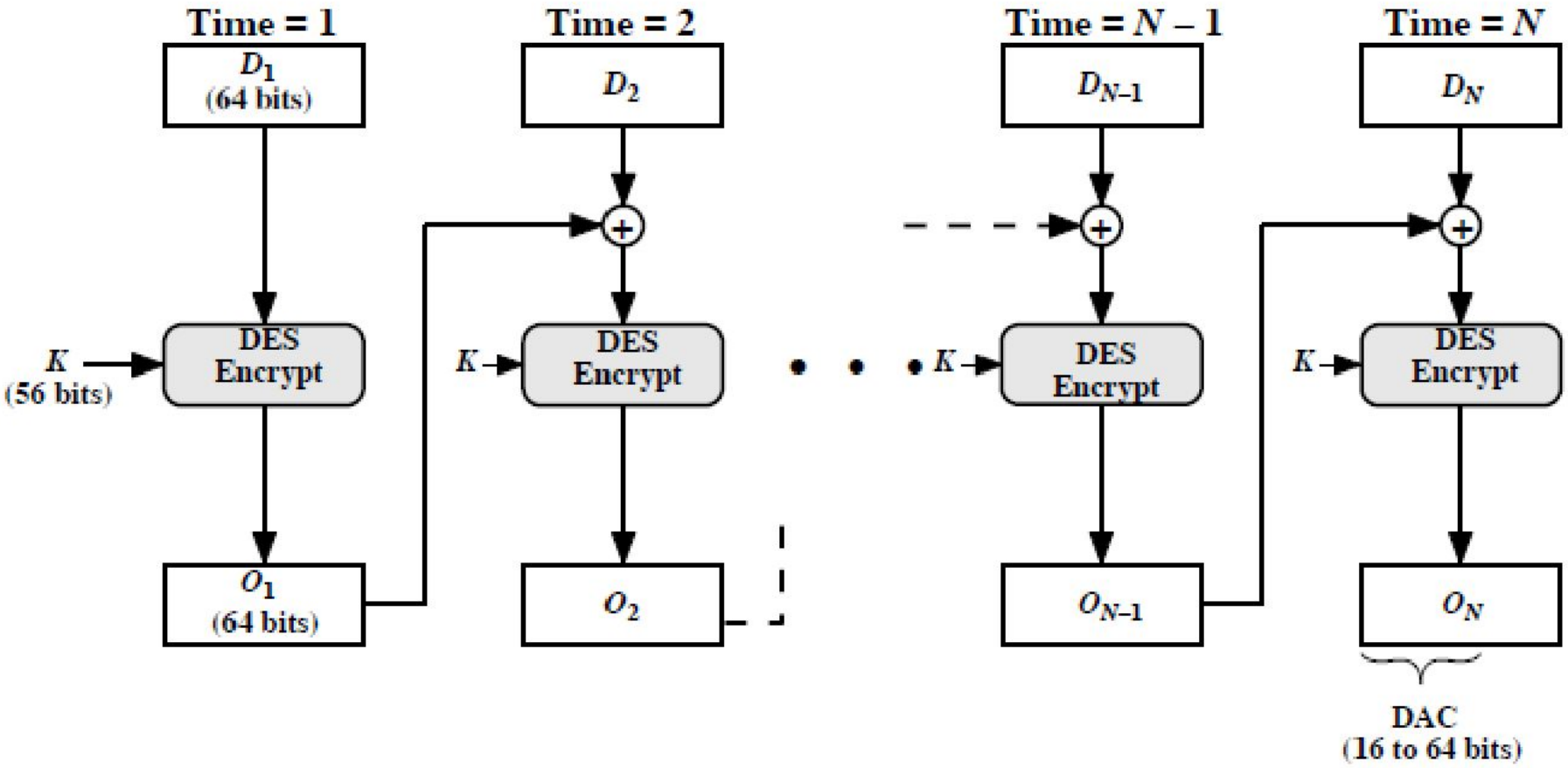
Message Authentication Code

- generated by an algorithm that creates a small fixed-sized block
  - depending on both message and some key
  - like encryption though need not be reversible
- is a many-to-one function
  - potentially many messages have same MAC
  - but finding these needs to be very difficult

Why MAC?

- Sometimes only authentication is needed.
- Sometimes need authentication to persist longer than the encryption (eg. Archival use).
- Sometimes messages are broadcasted to a number of destinations. Then only one destination can be made responsible for authenticity.

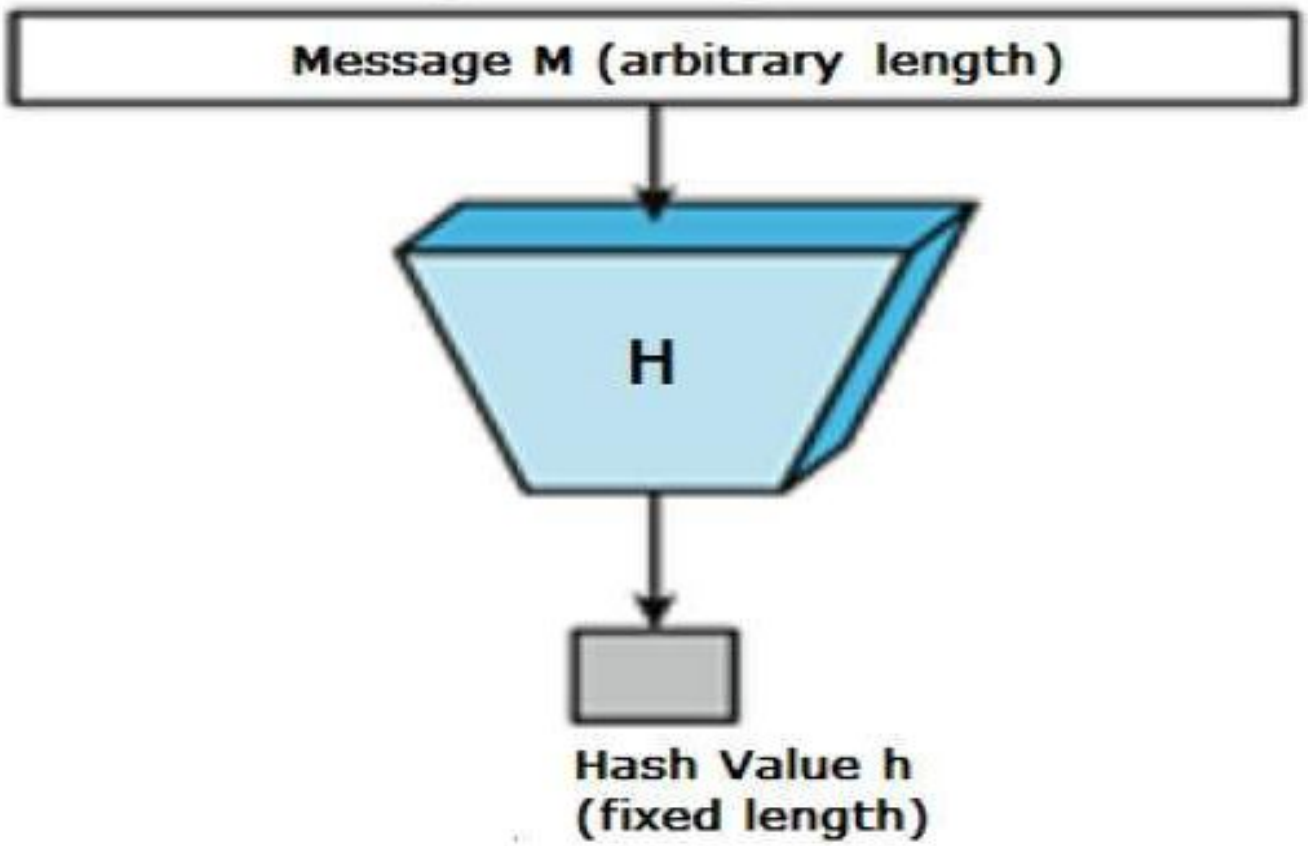
MAC based on DES



Hash Functions

Hash Functions

- Accepts variable size messages M as input and produces a fixed-size output, called hash code H(M).
- It doesn't use a key, unlike a MAC.
- The hash code is also called as a message digest or hash value.
- It is function of all bits of the message and provides an error-detection capability.


Message M (arbitrary length)
H
Hash Value h
(fixed length)

- A hash code does not use a key but is a function of the input message. It is also referred as hash value or message digests.
- The hash code is a function of all the bits of the message and provides an error detection capability.
- A change to any bits or bits in the message results in a change to the hash code.

Properties of Hash Functions

- No matter how big or small your input is, the output will always have a fixed 256-bits length

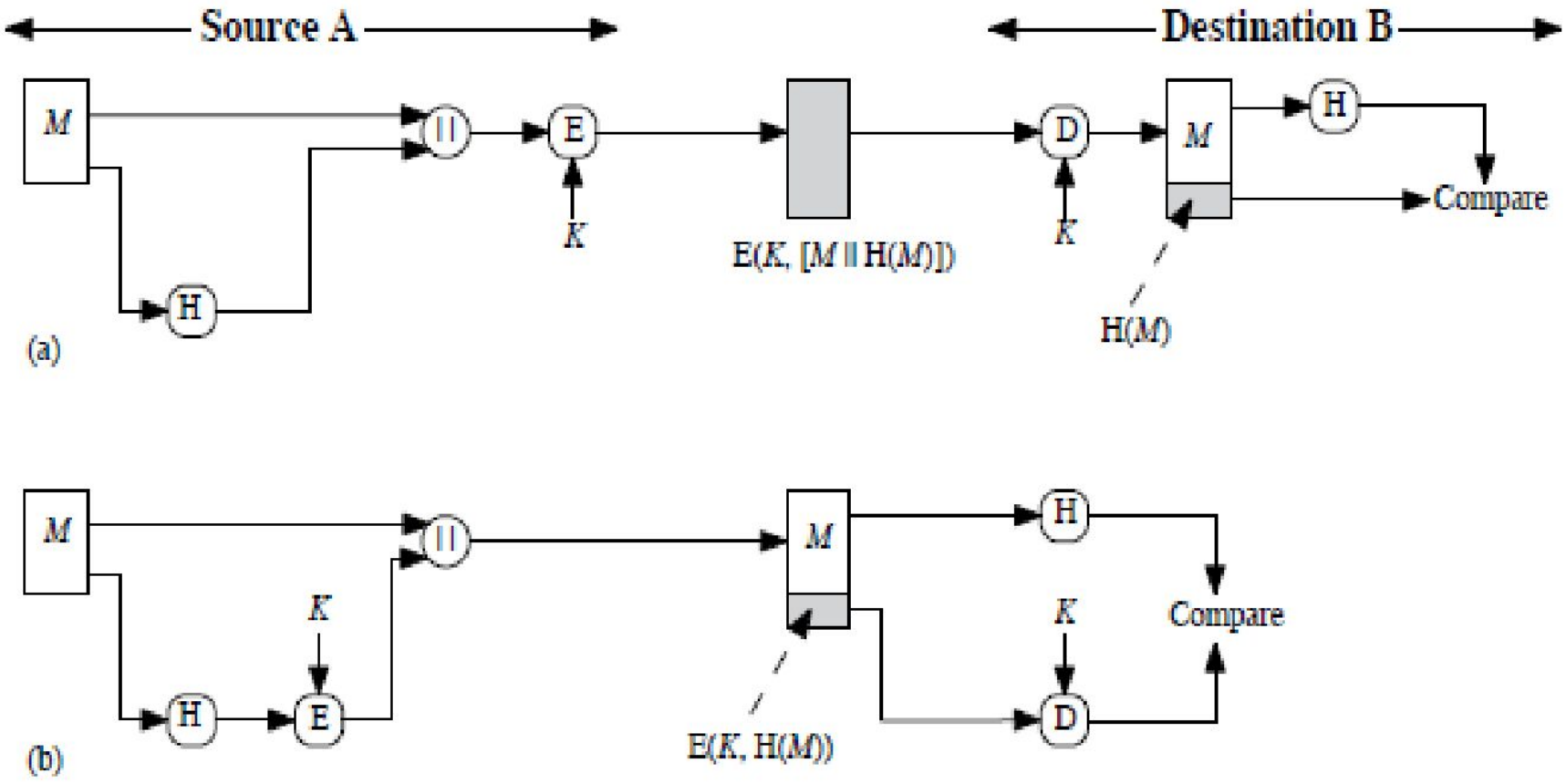
INPUT	HASH
Hi	3639EFCD08ABB273B1619E82E78C29A7DF02C1051B1820E99FC395DCAA3326B8
Welcome to blockgeeks. Glad to have you here.	53A53FC9E2A03F9B6E66D84BA701574CD9CF5F01FB498C41731881BCDC68A7C8

Properties of Hash Functions

- Avalanche Effect
  - A small change in your input, the changes that will be reflected in the hash will be huge

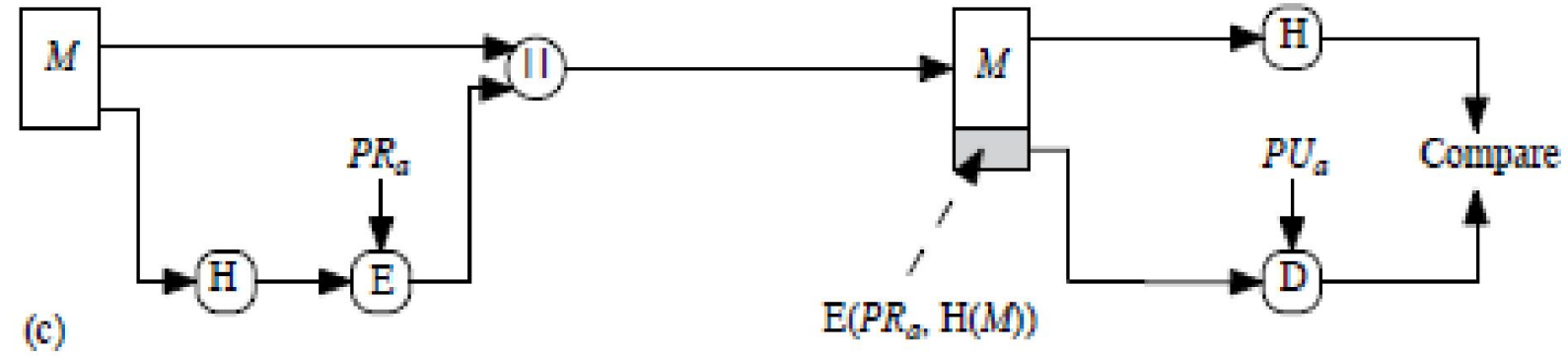
INPUT	HASH
This is a test	C7BE1ED902FB8DD4D48997C6452F5D7E509FBCDBE2808B16BCF4EDCE4C07D14E
this is a test	2E99758548972A8E8822AD47FA1017FF72F06F3FF6A016851F45C398732BC50C

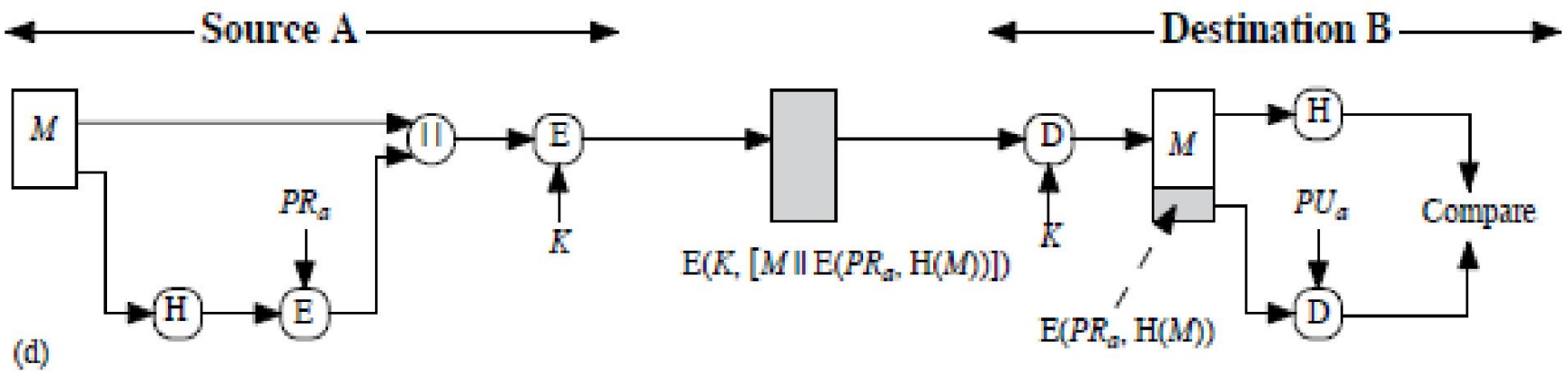
Basic Uses of Hash Functions



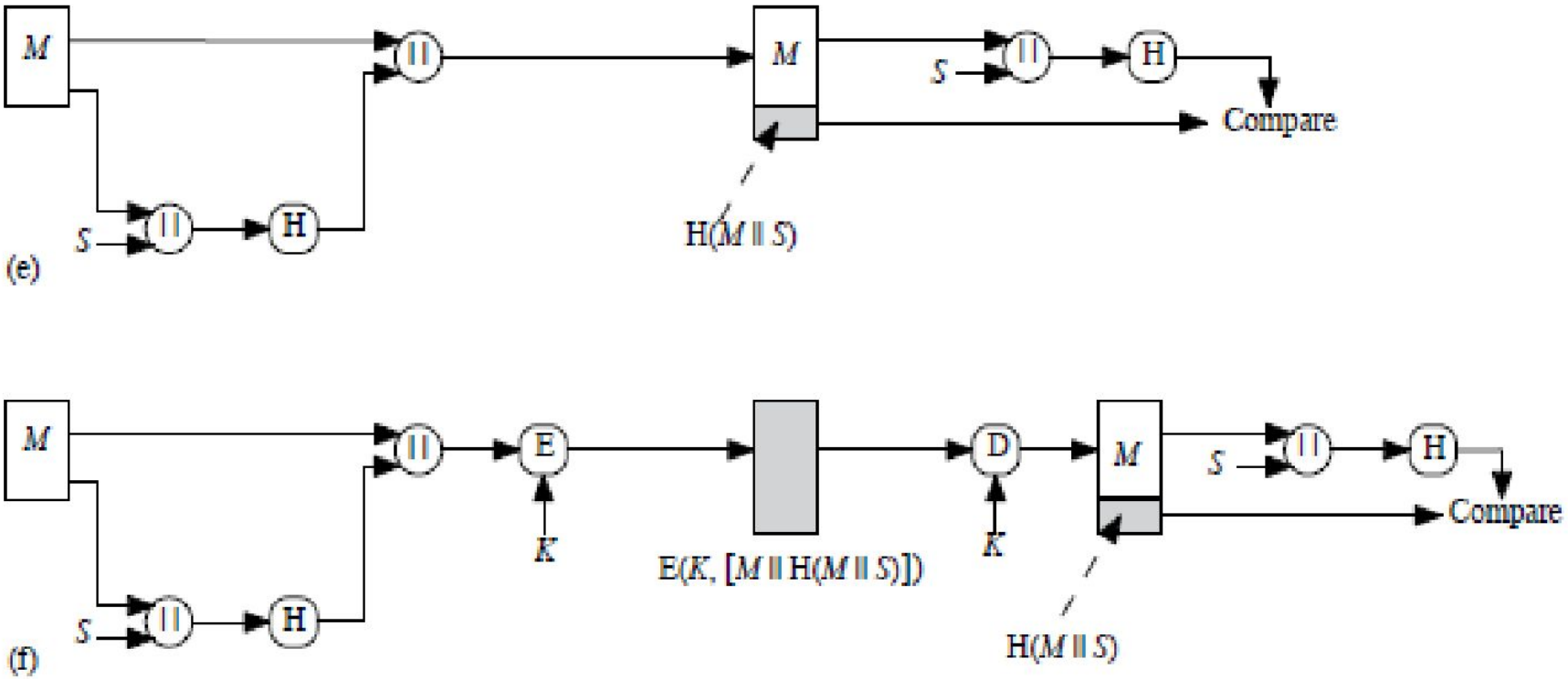
- The message plus concatenated hash code is encrypted using symmetric encryption.
- Only A and B share the secret key, the message must have come from A and has not been altered.
- The hash code provides the structure or redundancy required to achieve authentication. Because encryption is applied to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

Basic Uses of Hash Functions





Basic Uses of Hash Functions



- Popular Hash Functions
- Let us briefly see some popular hash functions −
- Message Digest (MD)
- MD5 was most popular and widely used hash function for quite some years.
- The MD family comprises of hash functions MD2, MD4, MD5 and MD6. It was adopted as Internet Standard RFC 1321. It is a 128-bit hash function.
- MD5 digests have been widely used in the software world to provide assurance about integrity of transferred file. For example, file servers often provide a pre-computed MD5 checksum for the files, so that a user can compare the checksum of the downloaded file to it.
- Secure Hash Function (SHA)
- Family of SHA comprise of four SHA algorithms; SHA-0, SHA-1, SHA-2, and SHA-3. Though from same family, there are structurally different.
- SHA-1 is the most widely used of the existing SHA hash functions. It is employed in several widely used applications and protocols including Secure Socket Layer (SSL) security.

Hash Functions

- Simple hash functions

$$C_i = b_{i1} \oplus b_{i2} \oplus b_{i3} \oplus \ldots \oplus b_{im}$$

Security of MACs

- Brute – Force Attacks
  - **Computation resistance**: Given one or more text-MAC pairs (x_i, $C_k(x_i)$), it is computationally infeasible to compute any text-MAC pair (x, $C_k(x)$), for any new input $x \neq x_i$.

- Certain properties of cryptographic hash functions impact the security of password storage.
- **Non-reversibility, or one-way function.** A good hash should make it very hard to reconstruct the original password from the output or hash.
- **Diffusion, or avalanche effect**. A change in just one bit of the original password should result in change to half the bits of its hash. In other words, when a password is changed slightly, the output of enciphered text should change significantly and unpredictably.
- **Determinism.** A given password must always generate the same hash value or enciphered text.
- **Collision resistance.** It should be hard to find two different passwords that hash to the same enciphered text.
- **Non-predictable.** The hash value should not be predictable from the password.

Security of Hash Functions

- Brute – Force Attacks
  - **One–way** : For any given code *h*, it is computationally infeasible to find *x* such that

    H(*x*) = *h*.
  - **Weak Collision resistance** : For any given block x, it is computationally infeasible to find $y \neq x$ with H(*y*) = H(*x*).
  - **Strong Collision resistance** : It is computationally infeasible to find any pair (*x,y*) such that H(*x*) = H(*y*)

Summary

- have considered:
  - message authentication using
    - message encryption
    - MACs
    - hash functions
  - general approach & security

Questions

List different types of attacks addressed by message authentication. (4)

Illustrate Needham and Schroedor protocol for mutual authentication. (4)