

Unit 4

LIKE



COMMENT



SHARE



SUBSCRIBE



1. Authentication requirement – Authentication function – MAC – Hash function – Security of hash function: HMAC, CMAC – SHA
2. Digital signature and authentication protocols – DSS – Schnorr Digital Signature Scheme – ElGamal cryptosystem
3. Entity Authentication: Biometrics, Passwords, Challenge Response protocols – Authentication applications – Kerberos ~~MUTUAL~~
4. TRUST: Key management and distribution – Symmetric key distribution using symmetric and asymmetric encryption – Distribution of public keys – X.509 Certificates.

LIKE



COMMENT



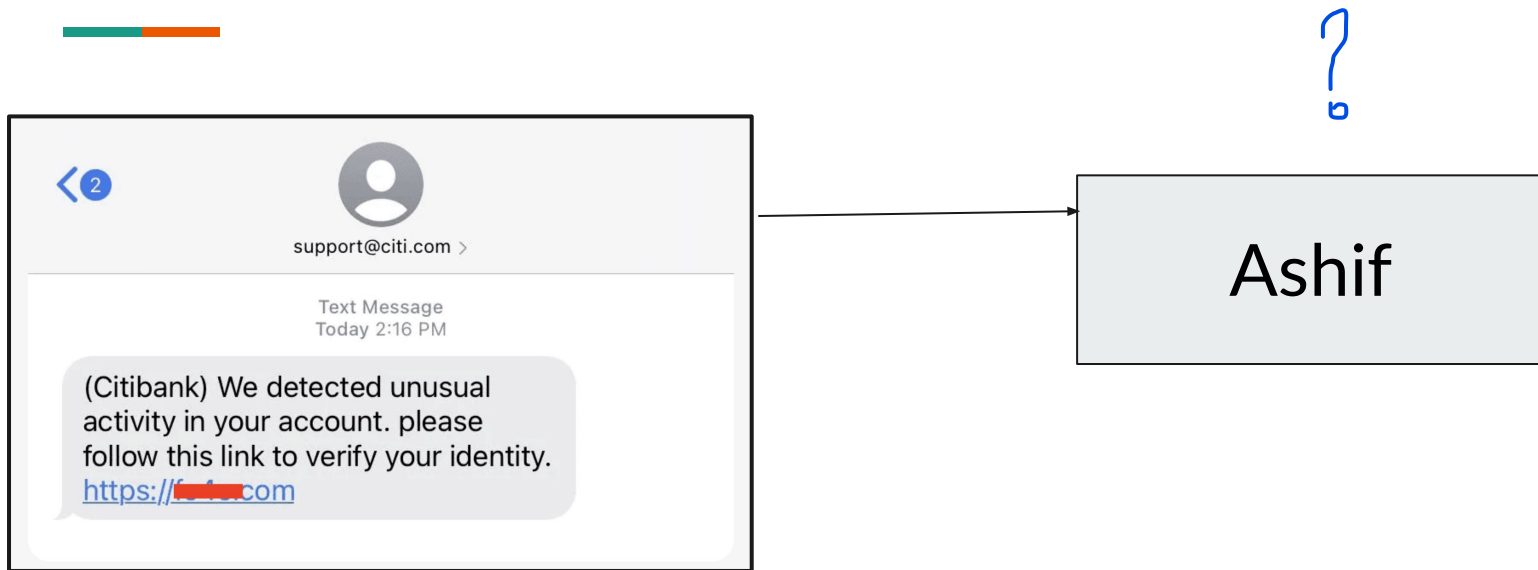
SHARE



SUBSCRIBE

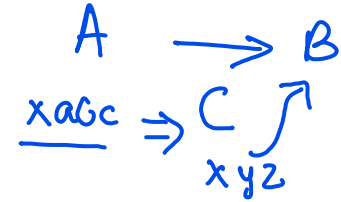
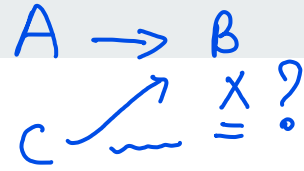


What is Message Authentication?



Need for Message Authentication

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. Also included are fraudulent acknowledgments of message receipt or nonreceipt by someone other than the message recipient.
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. In a connectionless application, an individual message (e.g., datagram) could be delayed or replayed.



Need for Digital Signature — obc, Birth?

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

Message Authentication Function

abc + 3456
~~~~~

- Any message authentication or digital signature mechanism has two levels of functionality.
- At the lower level, there must be some sort of function that produces an authenticator: a value to be used to authenticate a message.
- This lower-level function is then used as a primitive in a higher-level authentication protocol that enables a receiver to verify the authenticity of a message.

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

LIKE



COMMENT



SHARE

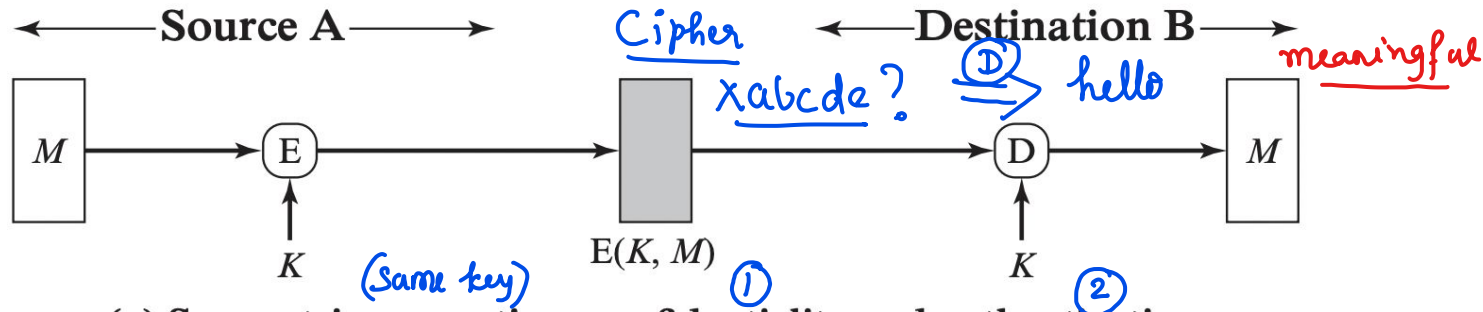


SUBSCRIBE

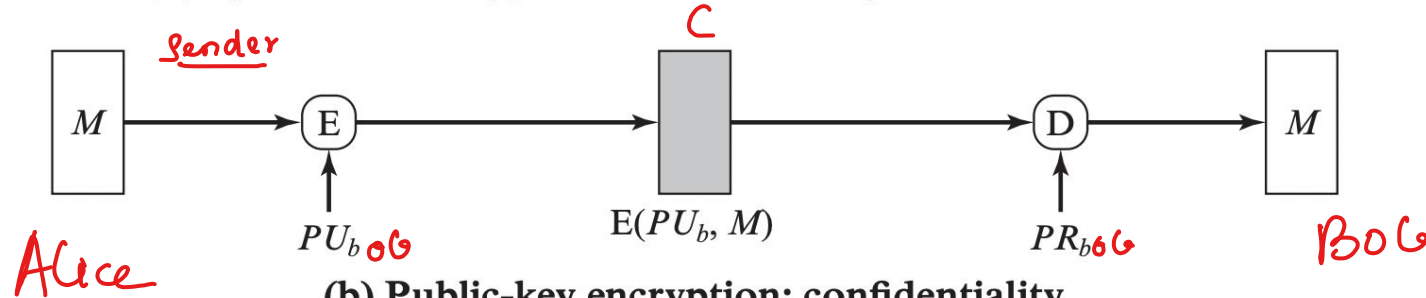


# 1. Message Encryption

$i j k l m \Rightarrow \textcircled{1} \Rightarrow \text{tweety}$  (CCS paper)  $A \rightarrow B$  (Exam)  
 $C \xrightarrow{\quad} B$



(a) Symmetric encryption: confidentiality and authentication



(b) Public-key encryption: confidentiality

$C \rightarrow P_{\text{Aman}} \Rightarrow X$  Aman  
 $A \rightarrow B$   
 Mohan  $\Rightarrow P_{\text{Bob}}$

# Key Points

- A message  $M$  transmitted from source  $A$  to destination  $B$  is encrypted using a secret key  $K$  shared by  $A$  and  $B$ . If no other party knows the key, then confidentiality is provided: No other party can recover the plaintext of the message.
- In addition,  $B$  is assured that the message was generated by  $A$ . Why? The message must have come from  $A$ , because  $A$  is the only other party that possesses  $K$  and therefore the only other party with the information necessary to construct ciphertext that can be decrypted with  $K$ .
- Furthermore, if  $M$  is recovered,  $B$  knows that none of the bits of  $M$  have been altered, because an opponent that does not know  $K$  would not know how to alter bits in the ciphertext to produce the desired changes in the plaintext.
- Suppose the message  $M$  can be any arbitrary bit pattern. In that case, there is no way to determine automatically, at the destination, whether an incoming message is the ciphertext of a legitimate message.
- Then the probability that any randomly chosen bit pattern, treated as ciphertext, will produce a legitimate plaintext message is very minimal.
- If the plaintext is, say, a binary object file or digitized X-rays, determination of properly formed and therefore authentic plaintext may be difficult. Thus, an opponent could achieve a certain level of disruption simply by issuing messages with random content purporting to come from a legitimate user.



# How to handle random noise?

Check sum (Code)

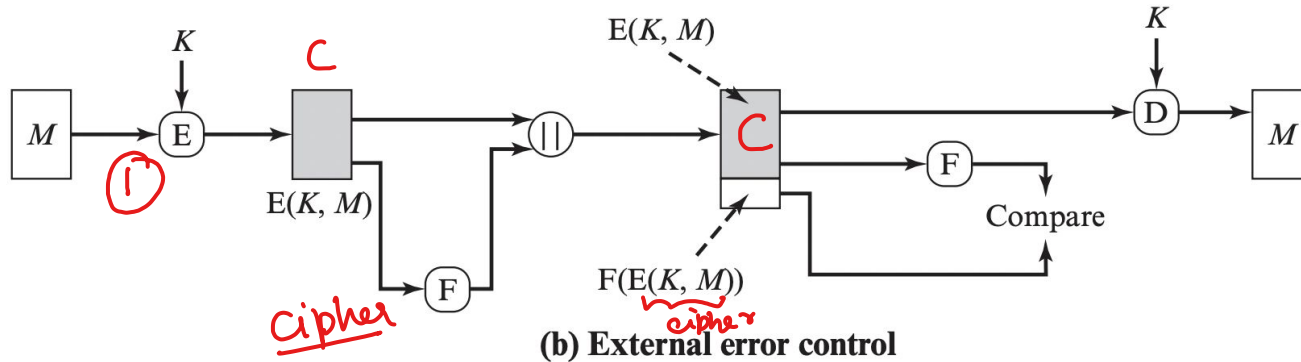
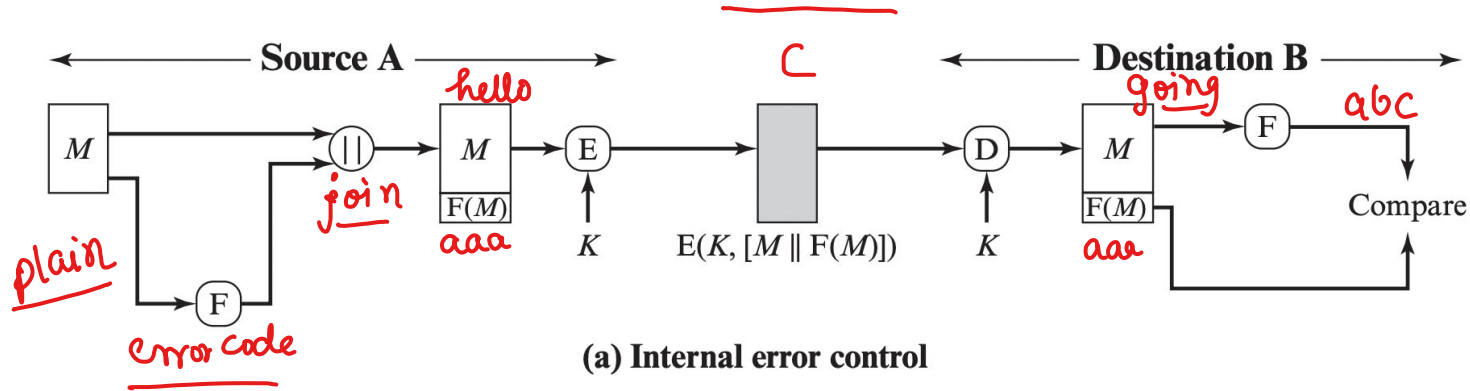
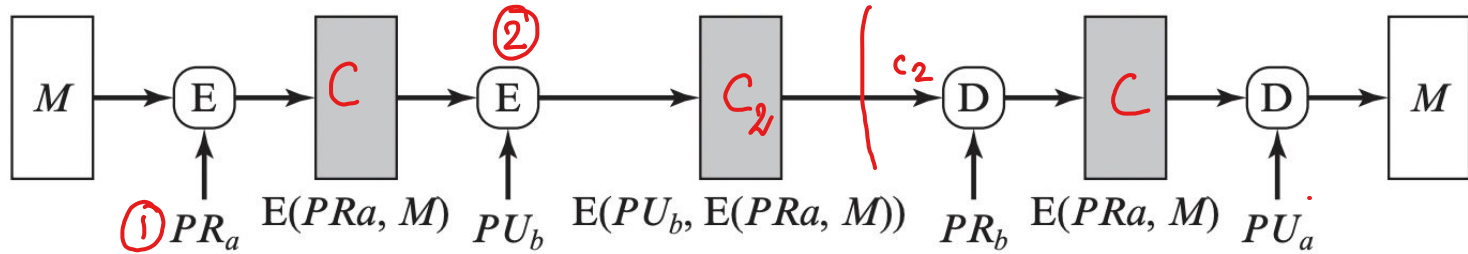
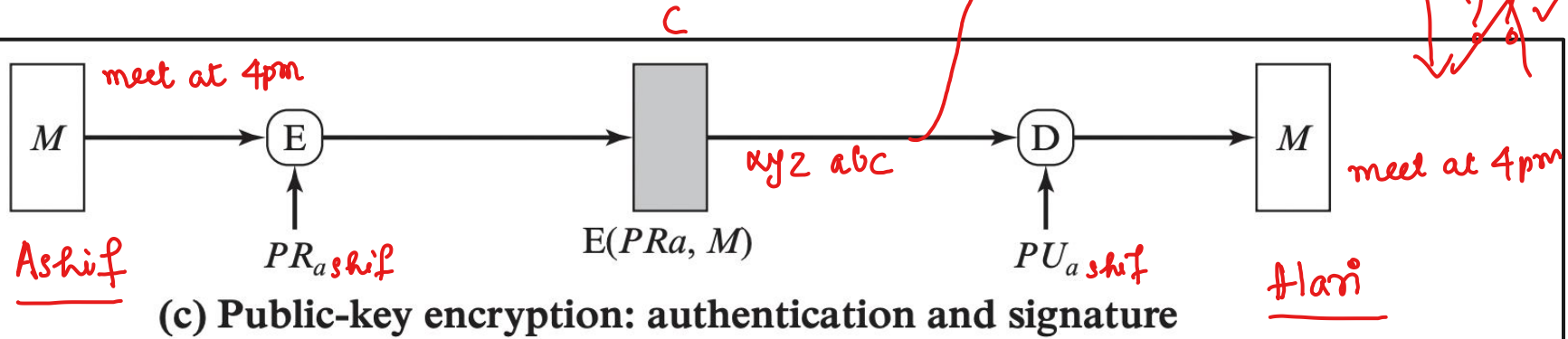


Figure 12.2 Internal and External Error Control

# Message Encryption




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

Handwritten notes at the bottom of the page:

$PR_a, PR_b \rightarrow PR_G$   
 $PR_G \rightarrow PU_a$

# Key Points

- 
- The straightforward use of public-key encryption (Figure 12.1b) provides confidentiality but not authentication. The source (A) uses the public key  $PU_b$  of the destination (B) to encrypt M. Because only B has the corresponding private key  $PR_b$ , only B can decrypt the message. This scheme provides no authentication, because any opponent could also use B's public key to encrypt a message and claim to be A.
  - To provide authentication, A uses its private key to encrypt the message, and B uses A's public key to decrypt.
  - 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 (Figure 12.1d). The disadvantage of this approach is that the public-key algorithm, which is complex, must be exercised four times rather than two in each communication.

LIKE



COMMENT



SHARE

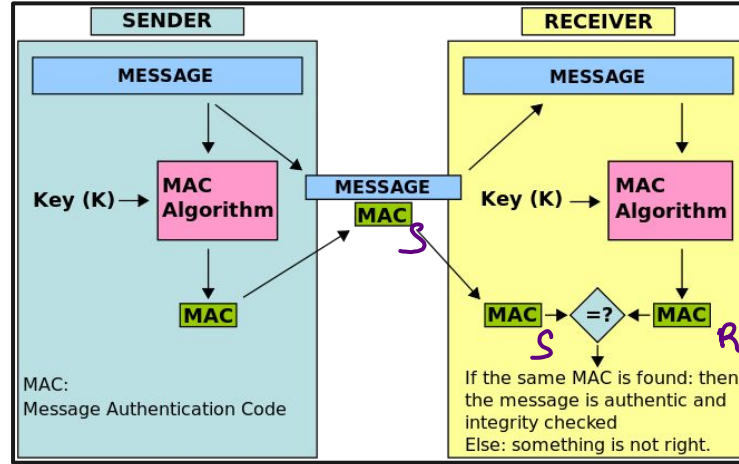


SUBSCRIBE



# Message Authentication Code

i/p (M, K)



A → B

hello ⇒ hello|abc ⇒

MAC → abc  
algo + key K

MAC

- A secret key is used to generate a small fixed-size block of data, known as a cryptographic checksum or MAC, that is appended to the original message.
- This technique assumes that two communicating parties, say A and B, share a common secret key K.

ⓧ

# Uses of MAC:

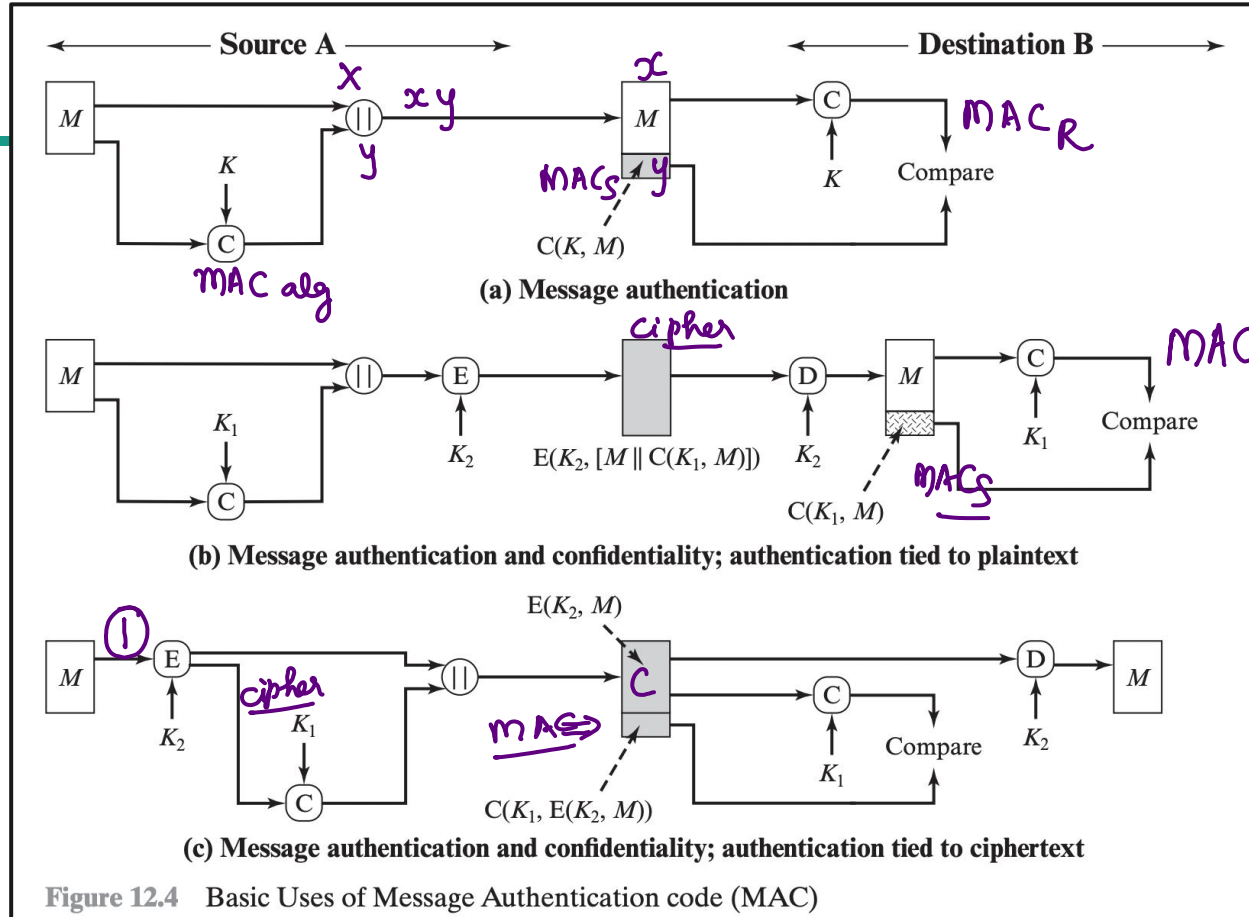


Figure 12.4 Basic Uses of Message Authentication code (MAC)

## Benefits of MAC:

1. 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 calculation of the 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.
2. 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.
3. If the message includes a sequence number (such as is used with HDLC, X.25, and TCP), then the receiver can be assured of the proper sequence because an attacker cannot successfully alter the sequence number.

# Applications of MAC when symmetric encryption does the same thing>>>>

1. In some cases, a single destination is responsible for verifying the authenticity of a broadcasted message, making it cost-effective and reliable. The message is sent in plain text with an attached authentication code, and the responsible system checks it. If an issue arises, other systems are alerted.
2. In situations where one side has a heavy message load and can't decrypt everything in real-time, authentication is done selectively by randomly checking messages.
3. Authenticating a computer program in plain text is useful. The program can run without constant decryption, saving processing power. Attaching a message authentication code allows integrity checks when needed.
4. Some applications prioritize message authentication over secrecy, like SNMPv3. Managed systems need to authenticate incoming SNMP messages, especially if they contain commands, even if the traffic itself doesn't need to be concealed.
5. Separating authentication and confidentiality functions provides flexibility in system design. You can authenticate at the application level and ensure confidentiality at a lower level like transport.
6. Users might want to protect messages beyond reception while still allowing processing. Encryption only protects messages during transit, not after decryption within the target system.



# Steps in MAC:

When A has a message to send to B, it calculates the MAC as a function of the message and the key:

$$\text{MAC} = C(K, M)$$

where

$M$  = 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

LIKE



COMMENT



SHARE



SUBSCRIBE

