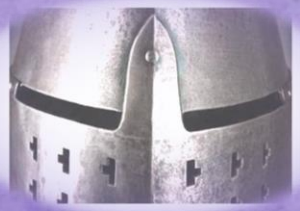


Message Authentication Codes

Book by William Stallings

Message Authentication Requirements

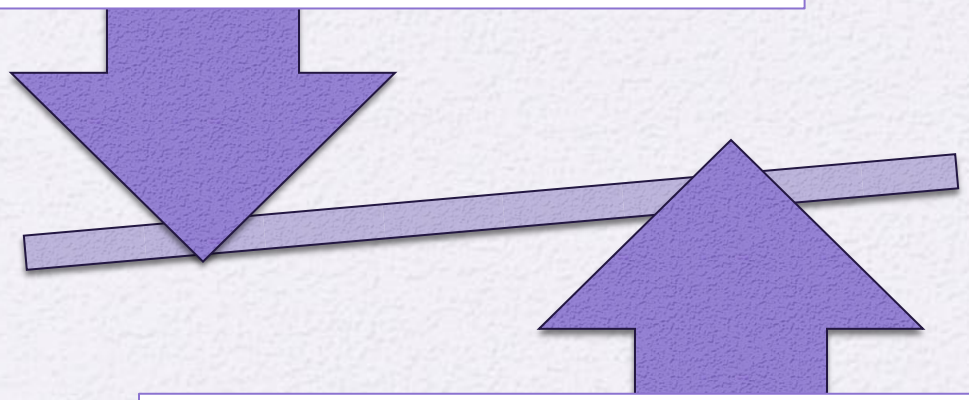
- Disclosure
  - Release of message contents to any person or process not possessing the appropriate cryptographic key
- Traffic analysis
  - Discovery of the pattern of traffic between parties
- Masquerade
  - Insertion of messages into the network from a fraudulent source
- Content modification
  - 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
- Source repudiation
  - Denial of transmission of message by source
- Destination repudiation
  - Denial of receipt of message by destination

Message Authentication Functions

- Two levels of functionality:

Lower level

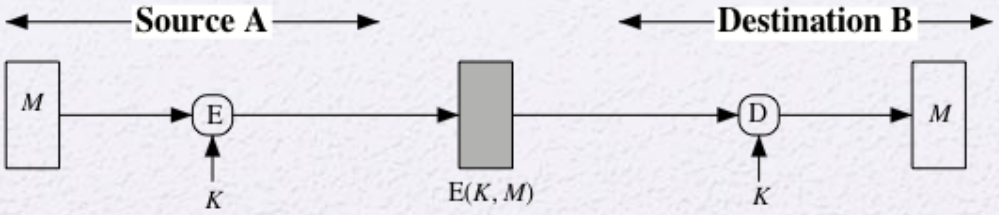
- There must be some sort of function that produces an authenticator



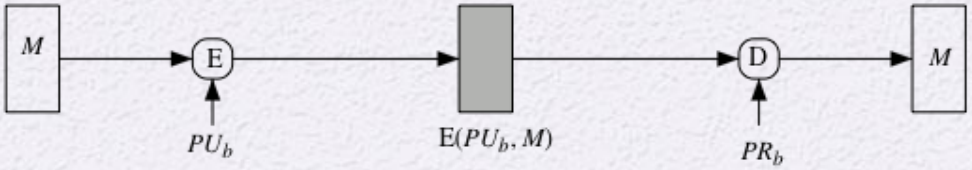
Higher-level

- Uses the lower-level function as a primitive in an 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



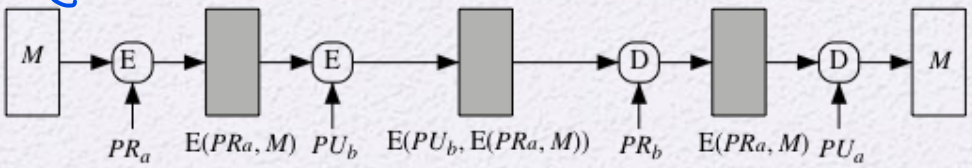
(a) Symmetric encryption: confidentiality and authentication



(b) Public-key encryption: confidentiality

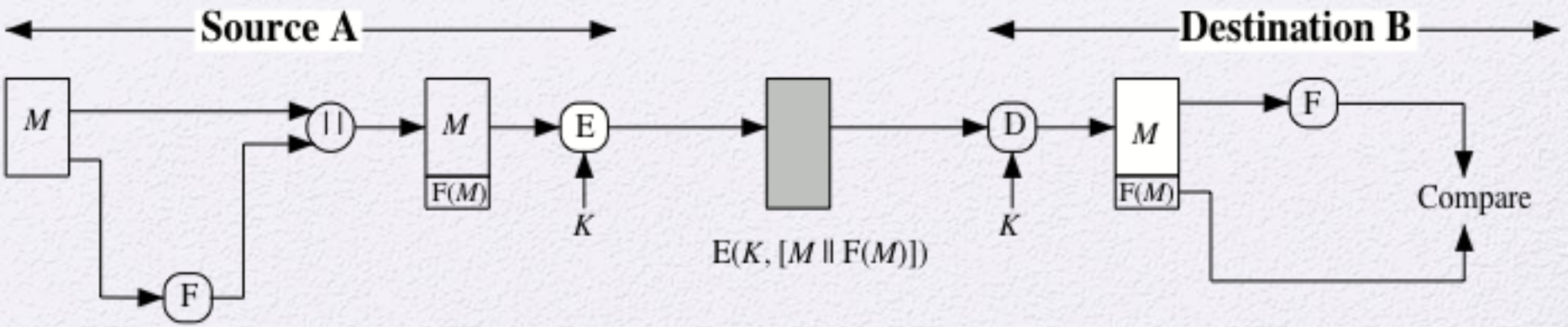
(c) Public-key encryption: authentication and signature

pehla authenticate karo and then confidential 6e aevu prove karo.

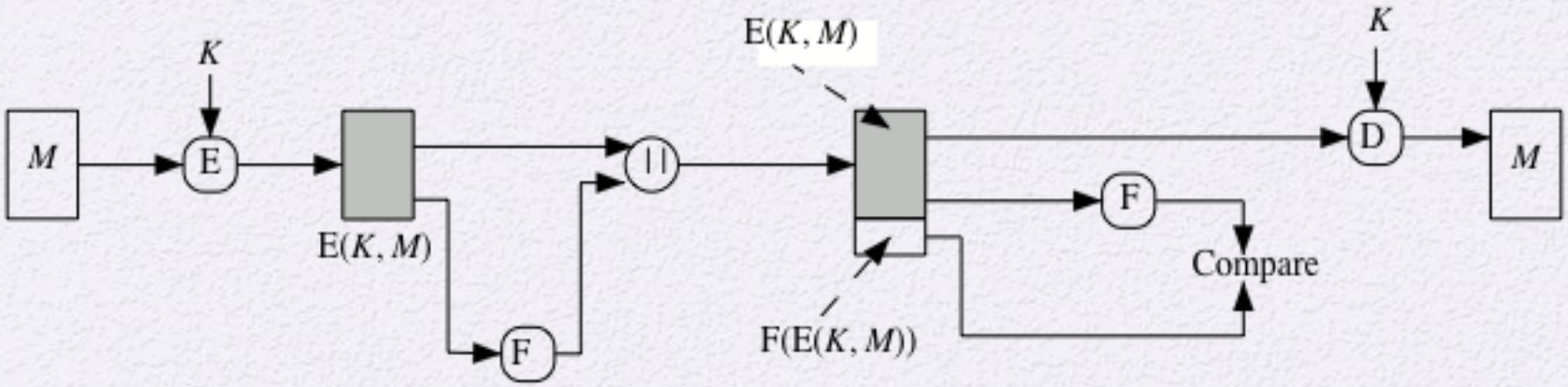


(d) Public-key encryption: confidentiality, authentication, and signature

Figure 12.1 Basic Uses of Message Encryption



(a) Internal error control



(b) External error control

Figure 12.2 Internal and External Error Control

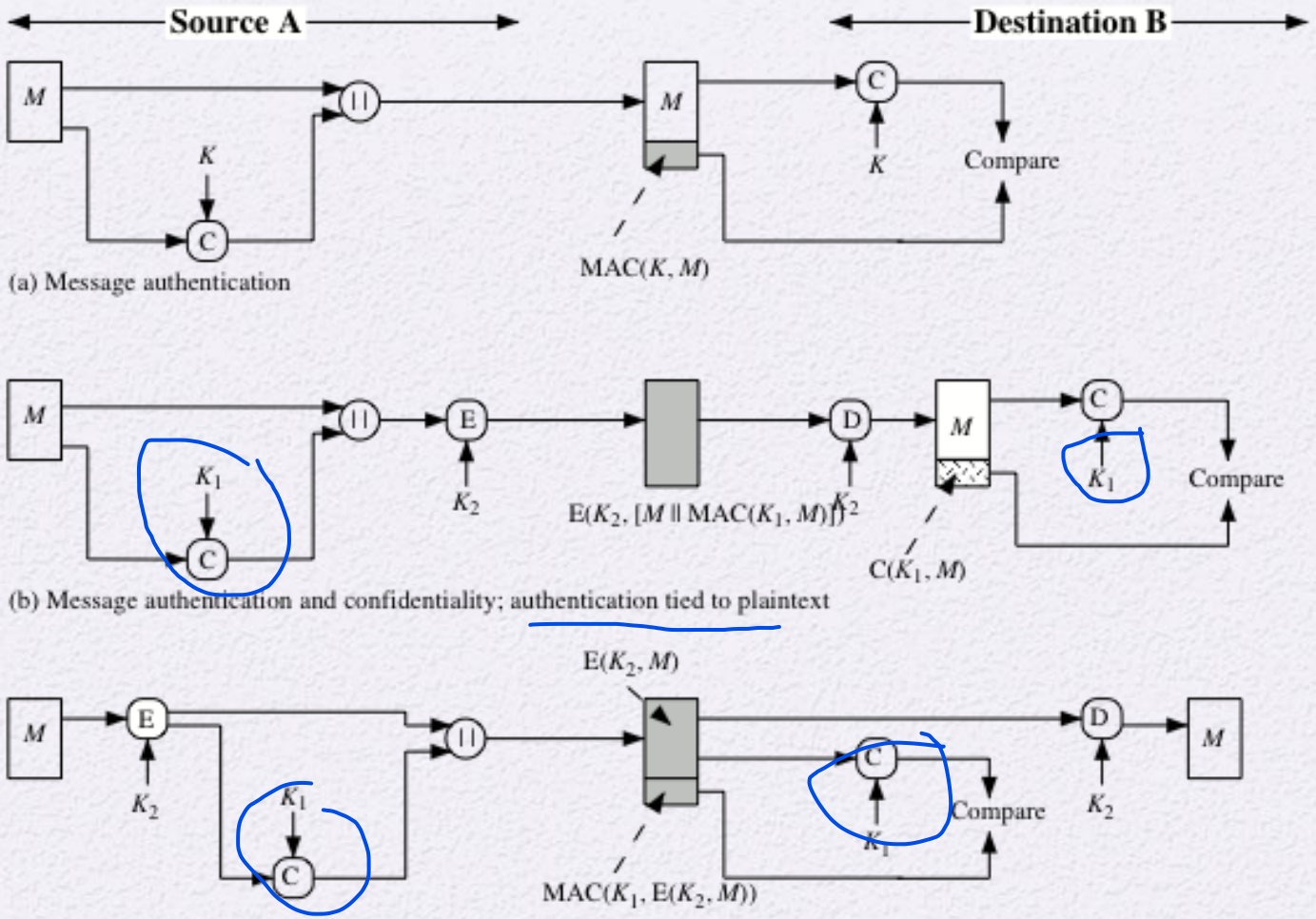


Figure 12.4 Basic Uses of Message Authentication Code (MAC)

Requirements for MACs

Taking into account the types of attacks, the MAC needs to satisfy the following:

The first requirement deals with message replacement attacks, in which an opponent is able to construct a new message to match a given MAC, even though the opponent does not know and does not learn the key

The second requirement deals with the need to thwart a brute-force attack based on chosen plaintext

The final requirement dictates that the authentication algorithm should not be weaker with respect to certain parts or bits of the message than others

Brute-Force Attack

- Requires known message-tag pairs
  - A brute-force method of finding a collision is to pick a random bit string y and check if $H(y) = H(x)$

Two lines of attack:

- Attack the key space
  - If an attacker can determine the MAC key then it is possible to generate a valid MAC value for any input x
- Attack the MAC value
  - Objective is to generate a valid tag for a given message or to find a message that matches a given tag

Cryptanalysis

- Cryptanalytic attacks seek to exploit some property of the algorithm to perform some attack other than an exhaustive search
- An ideal MAC algorithm will require a cryptanalytic effort greater than or equal to the brute-force effort
- There is much more variety in the structure of MACs than in hash functions, so it is difficult to generalize about the cryptanalysis of MACs

HMAC Design Objectives

RFC 2104 lists the following objectives for HMAC:

- To use, without modifications, available hash functions
- To allow for easy replaceability of the embedded hash function in case faster or more secure hash functions are found or required
- To preserve the original performance of the hash function without incurring a significant degradation
- To use and handle keys in a simple way
- To have a well understood cryptographic analysis of the strength of the authentication mechanism based on reasonable assumptions about the embedded hash function

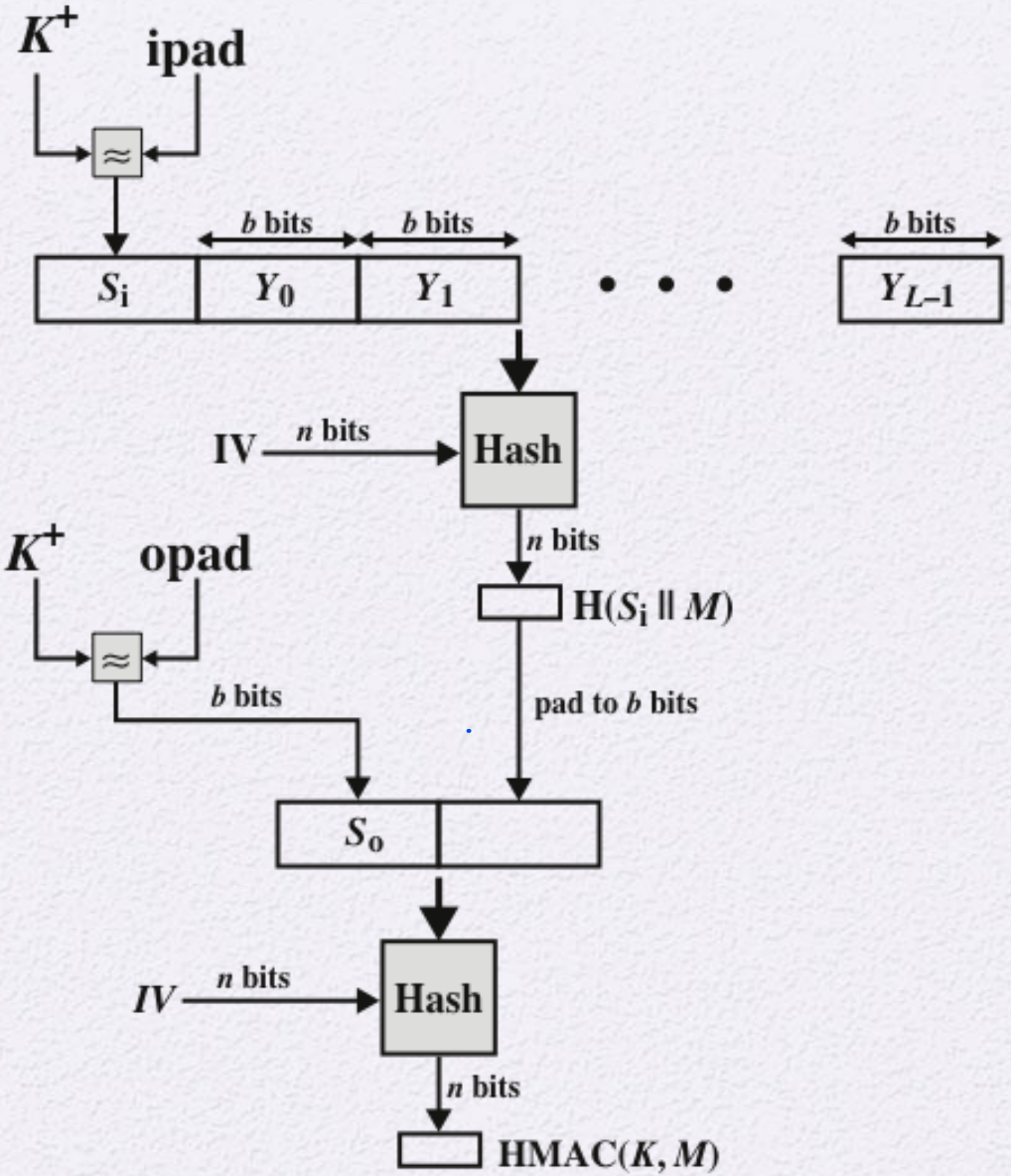


Figure 12.5 HMAC Structure

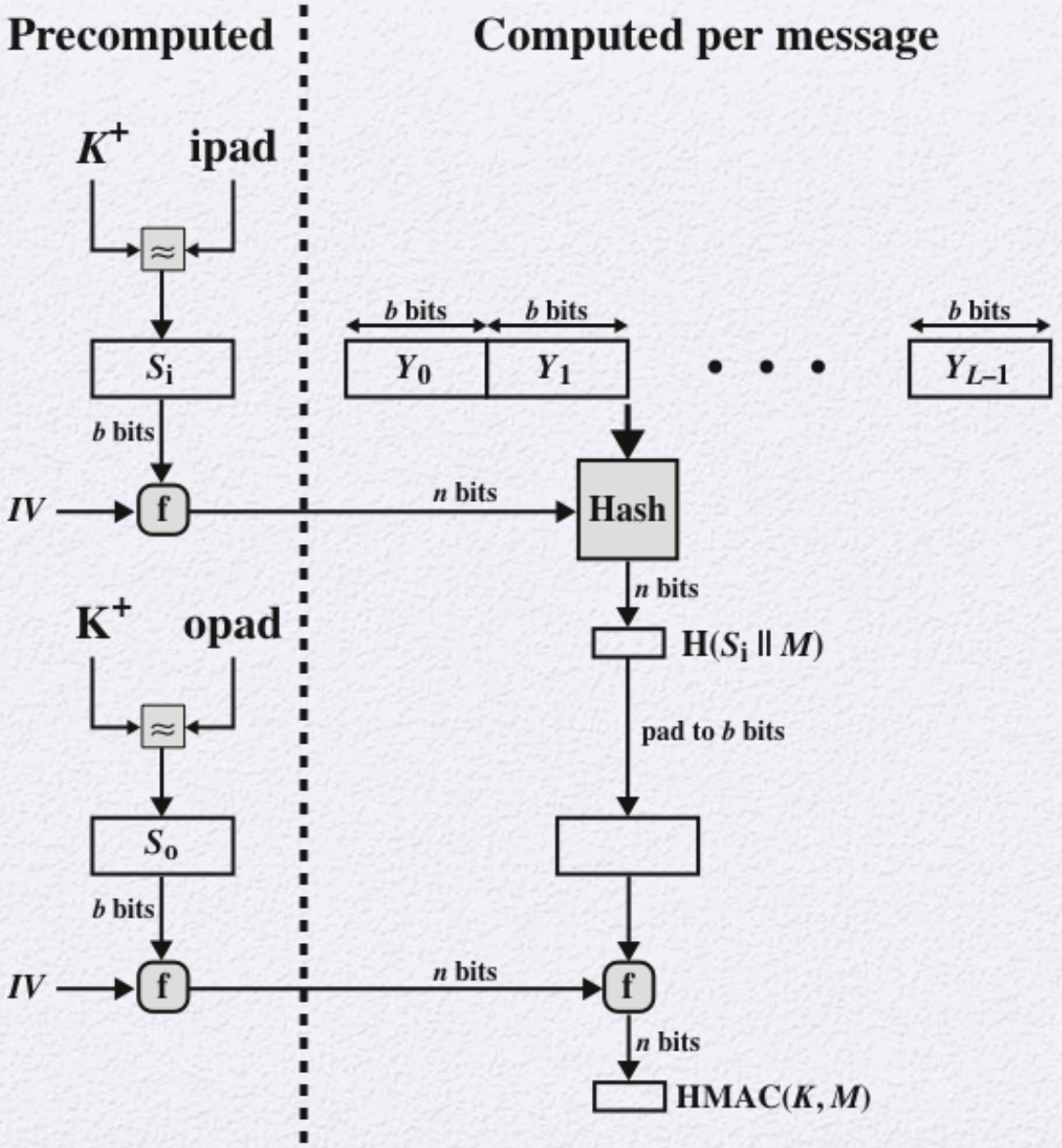
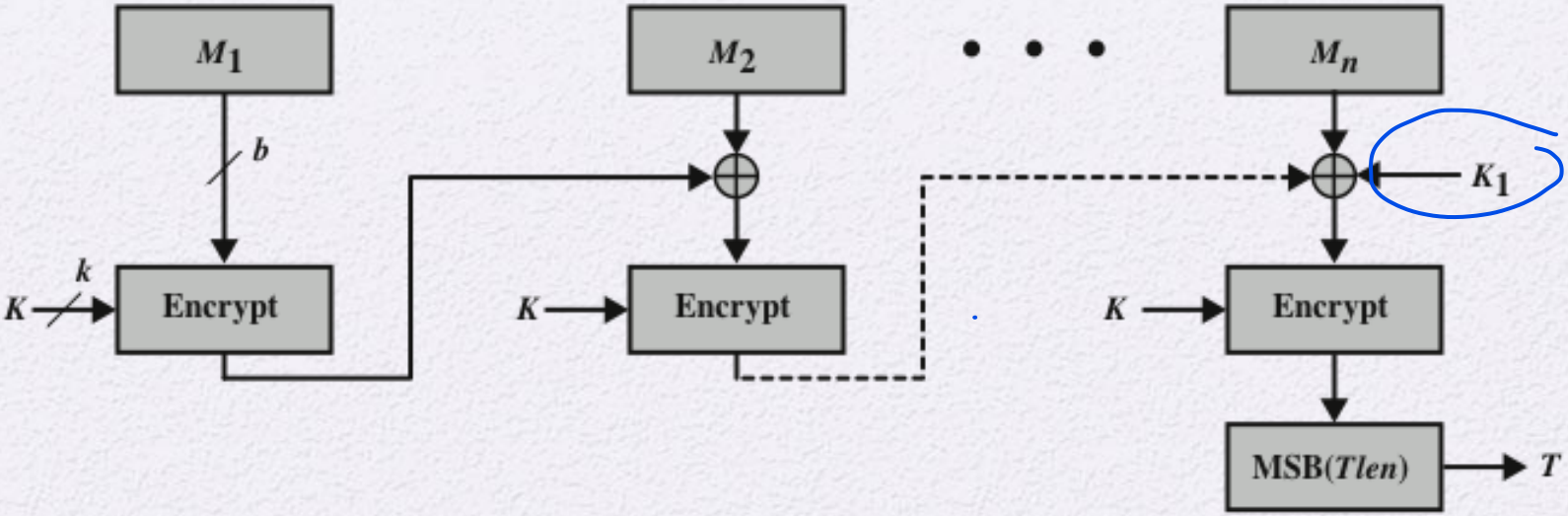
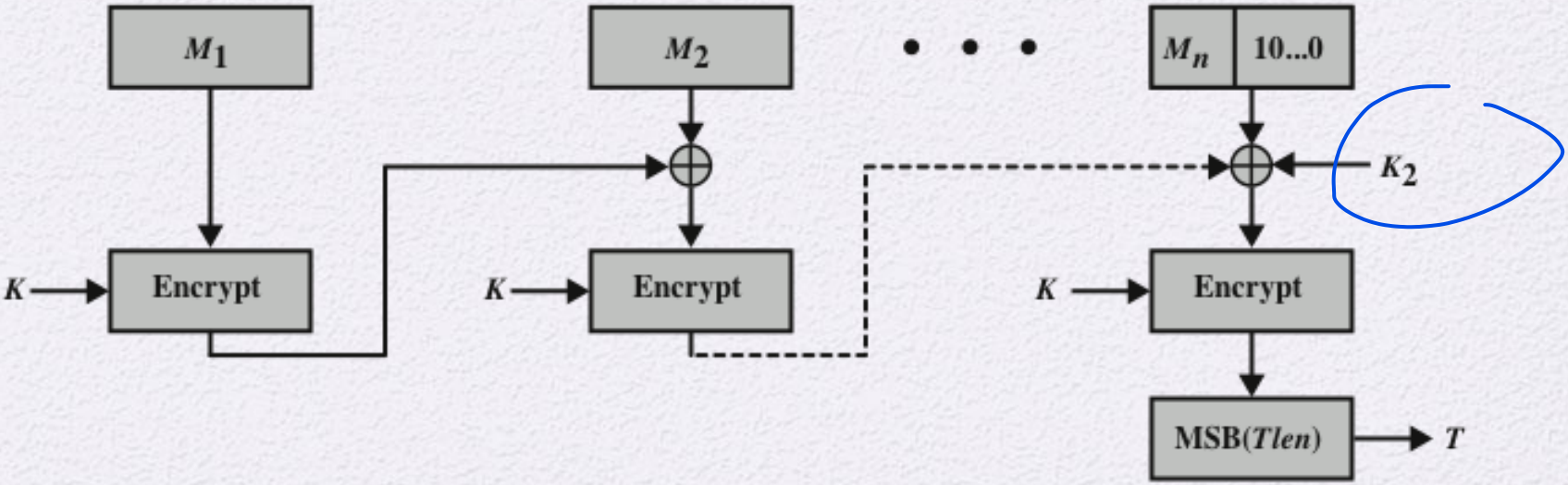


Figure 12.6 Efficient Implementation of HMAC



(a) Message length is integer multiple of block size



(b) Message length is not integer multiple of block size

Figure 12.8 Cipher-Based Message Authentication Code (CMAC)