# CS 465
# Computer Security

MAC: Message Authentication Code

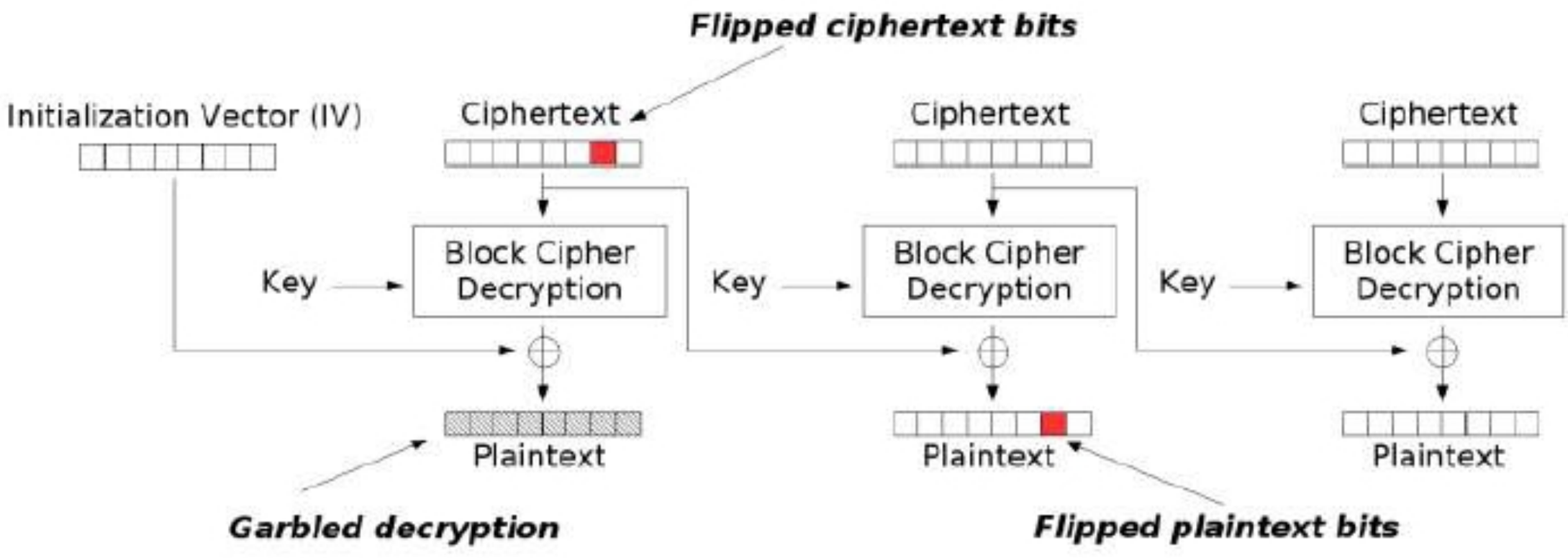
# What Assurances are Provided by Symmetric Encryption?

Assume CTR or CBC mode

- Authentication?
- Confidentiality?
- Integrity?
- Non-repudiation?

# Bit Flipping Attacks (Block Cipher)

## Modification attacks on CBC




Modification attack on CBC

# Bit Flipping Attacks (Stream Cipher)

- Plaintext:
  - ACCT_NO:123-45-6789 ADD:100
- Ciphertext:
  - 15b1206b7efa68b9 89 c87357507e3a27a138ca dc b2a1bb f8 eebee5

# Goals of Message Authentication

- Assure that the message has not been altered
- Assure the source of the message is authentic

# Message Authentication: Ciphertext vs. Plaintext

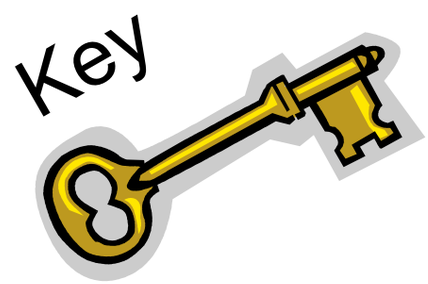
- Authentication of encrypted messages
  - Include an error-detection code in plaintext message
  - Attach a key-based error-detection code to an encrypted message
  - Attach a TAG – remember the newer AEAD modes
- Authentication of plaintext messages
  - Authentication without confidentiality
  - Attach a key-based error-detection code to plaintext message

# Message Authentication Code (MAC)

Dear BYU,

Thank you so much for an awesome computer security course.

Sincerely,
Emma




MAC Algorithm

This message really is from me and hasn't been modified

# Message Authentication Code (MAC)

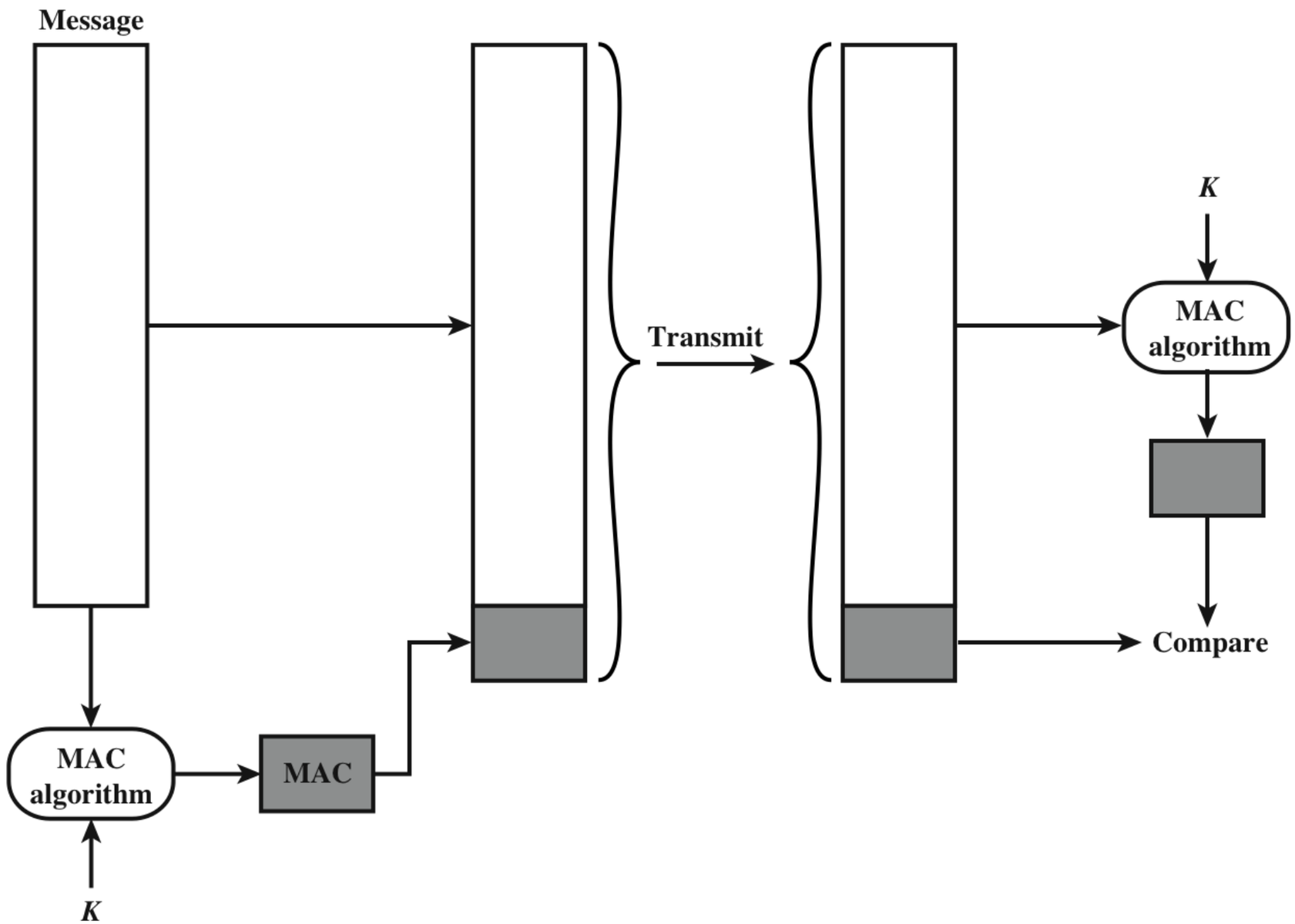



**Figure 3.1 Message Authentication Using a Message Authentication Code (MAC)**

Source: Network Security Essentials (Stallings)

# Three Ways to Implement a MAC

1. CBC-MAC

- Use CBC mode and a block cipher — fixed length messages only

- OMAC — for variable length messages

Source: Network Security Essentials (Stallings)

# OMAC1 (also called CMAC)

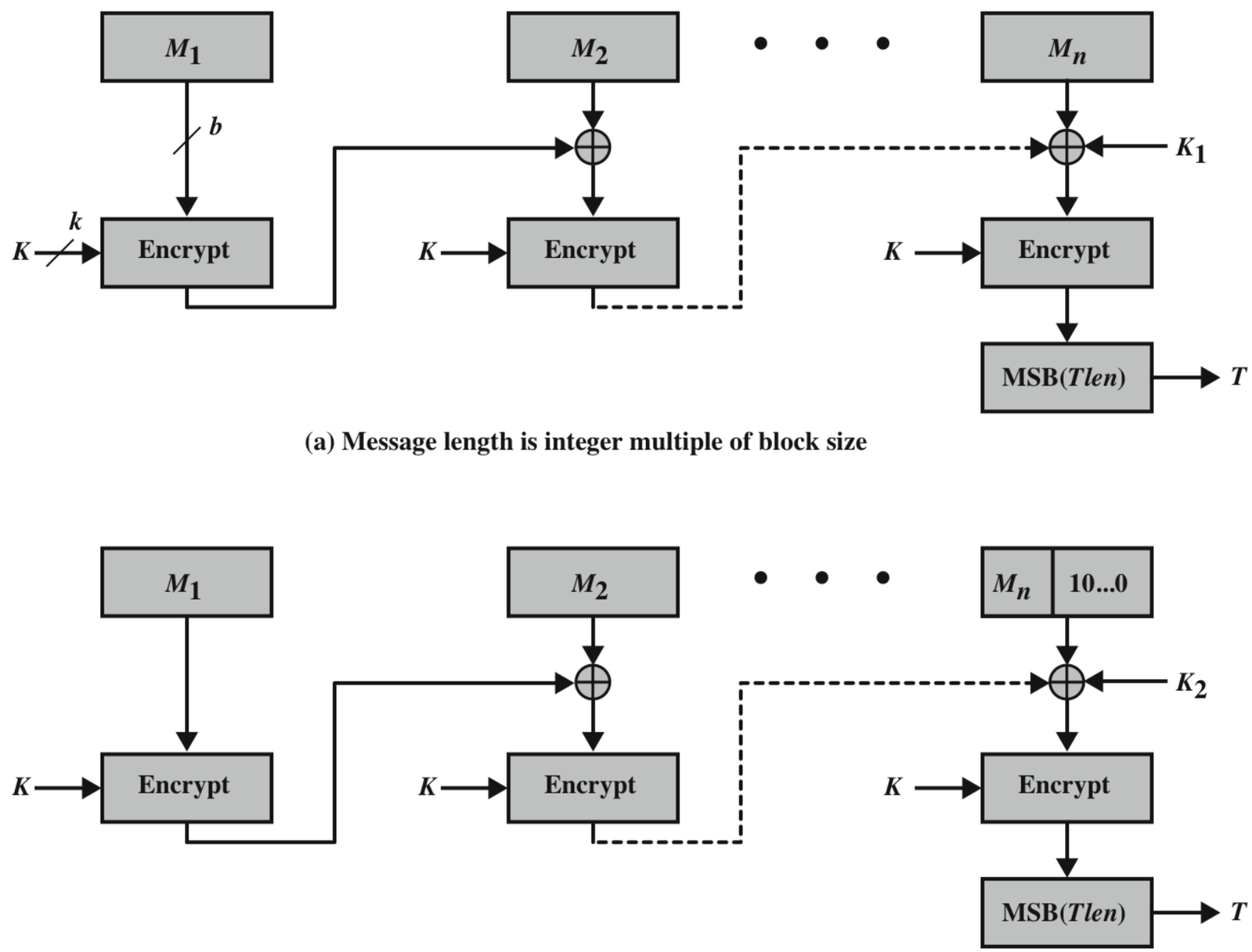
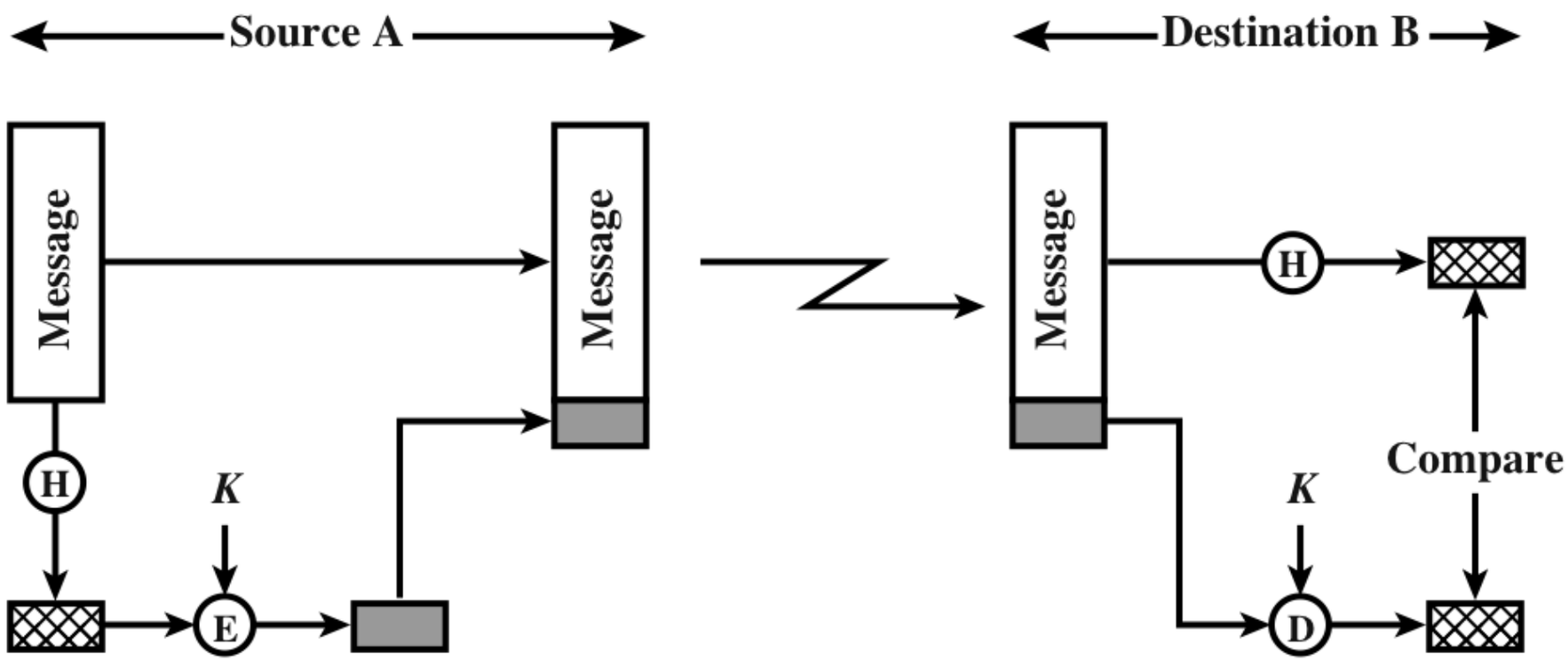



**Figure 12.12 Cipher-Based Message Authentication Code (CMAC)**

# Three Ways to Implement a MAC

2. Hash the message and encrypt the digest



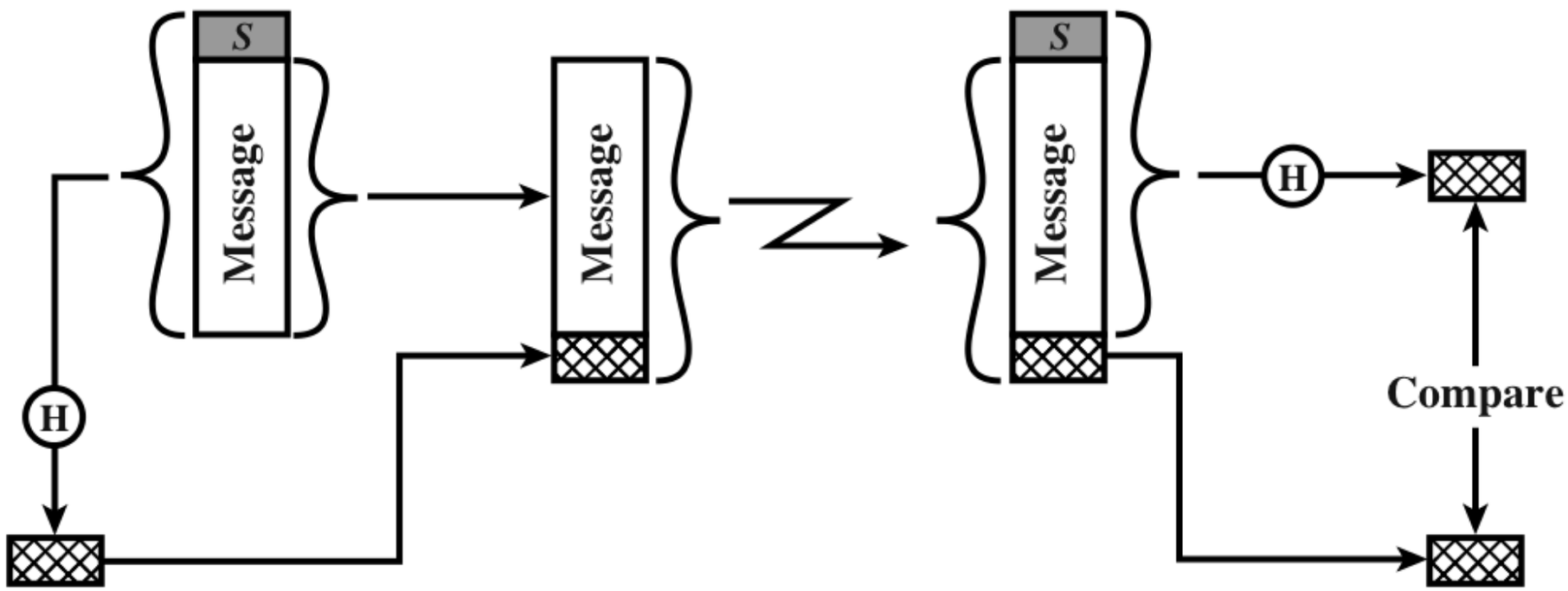

(a) Using conventional encryption

Source: Network Security Essentials (Stallings)

# Three Ways to Implement a MAC

3. Hash the message along with a shared key

- MAC generated using hashing is known as an HMAC

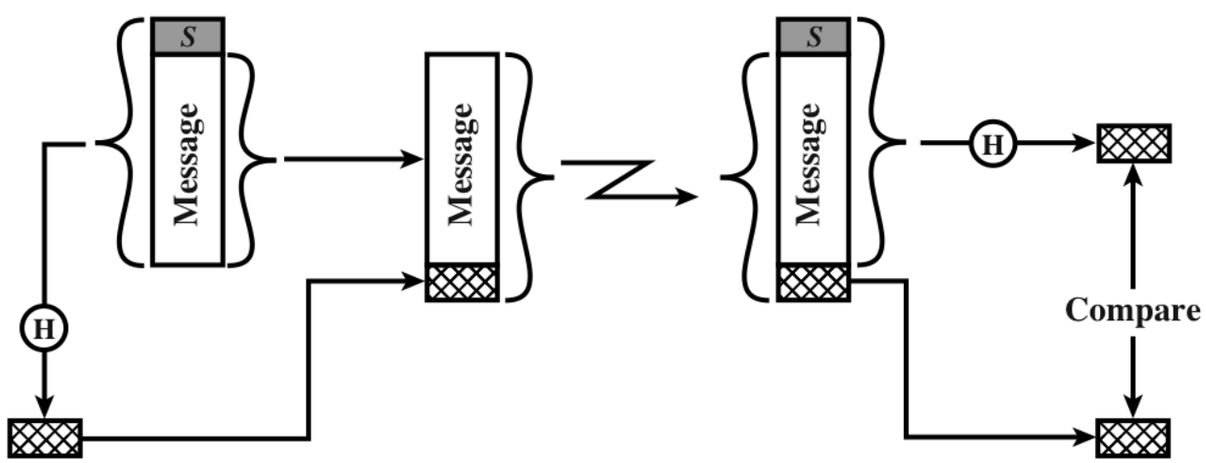



(c) Using secret value

Source: Network Security Essentials (Stallings)

# Design Flaw!

- Cryptographers recommend against this kind of HMAC using modern hash functions
- Vulnerable to a message extension attack
- An example of an implementation weaknesses in the algorithm


(c) Using secret value

# Iterative Hash Function

- Popular hash functions (MD5, SHA1, SHA2) use an iterative implementation technique known as the Merkle-Damgård construction
- SHA-3 uses a sponge construction

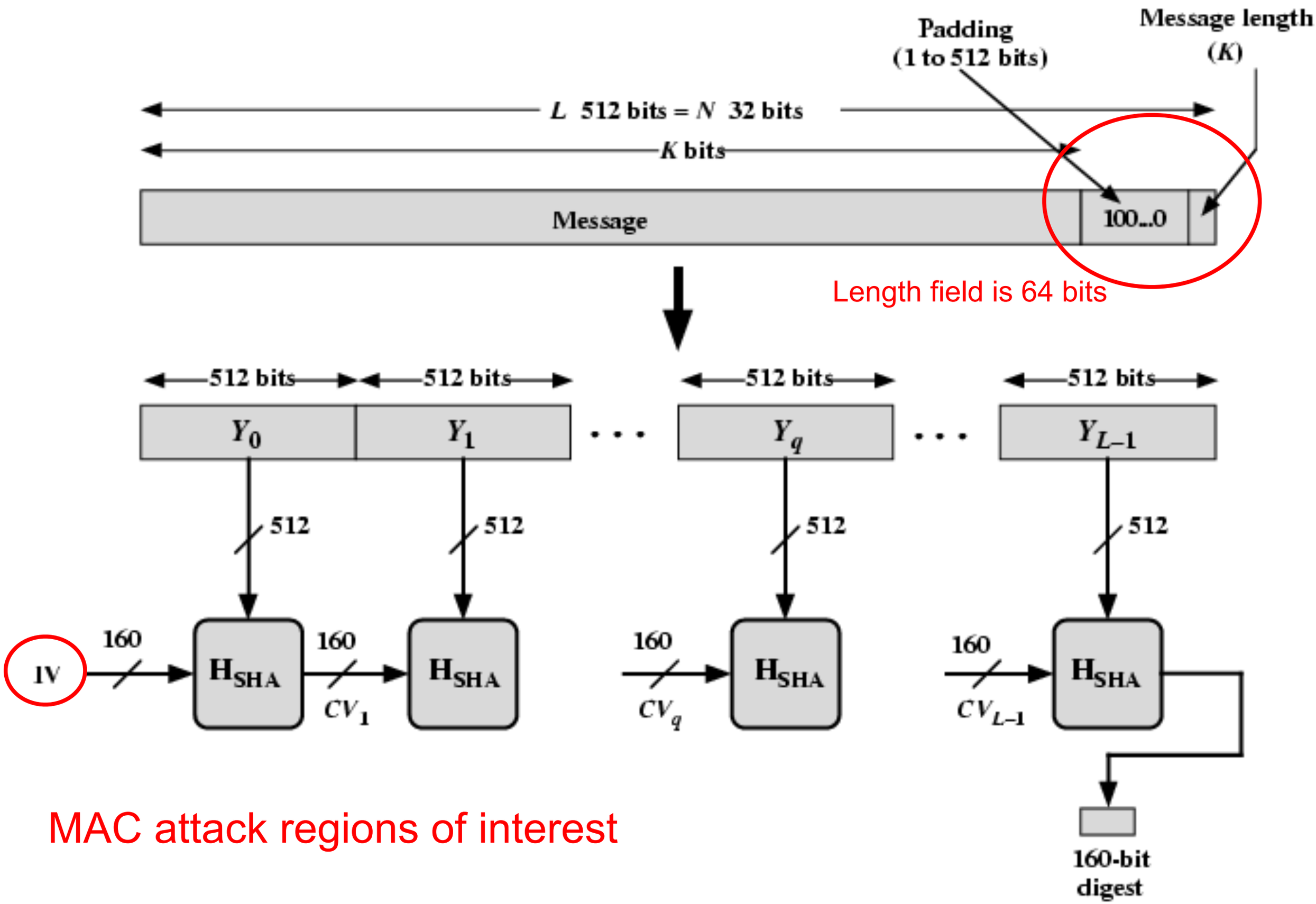



**Figure 3.4 Message Digest Generation Using SHA-1**

# Alice and Bob

- Alice and Bob share a key K
- Alice sends message M1 to Bob such that Bob knows it came from Alice
  - Alice computes H(K || M1) = mac1
  - Alice sends M1 and mac1 to Bob
- Bob verifies the message
  - Bob computes H(K || M1) = mac2 and compares it to mac1. If they match, the message came from Alice.
  - Or did it????

# Message Extension Attack

- Mallory can intercept a plaintext message and a mac.
- Mallory “extends” the message – adds new material to the end of the message
- She modifies the mac without knowing the key. She needs to know the length of the key.
- She replaces the message and mac with the extended message and new mac and forwards it along
- Bob receives the modified message and mac, and it passes his verification step. He believes is came from Alice!
- See Project 3, resources on that page

# HMAC

- Because of the message extension attack vulnerability, the government standard HMAC algorithm guards against this threat
  - FIPS 198
  - RFC 2104

$$\mathrm{HMAC}(K, m) = H\Big((K' \oplus opad) \| H\big((K' \oplus ipad) \| m\big)\Big)$$

- K' = H(K) if K is larger than the block size, otherwise K
- opad = 0x5c5c5c…5c5c, one-block-long constant
- ipad = 0x363636…3636, one-block-long constant
- IV is fixed, as with SHA-2 and other hash functions

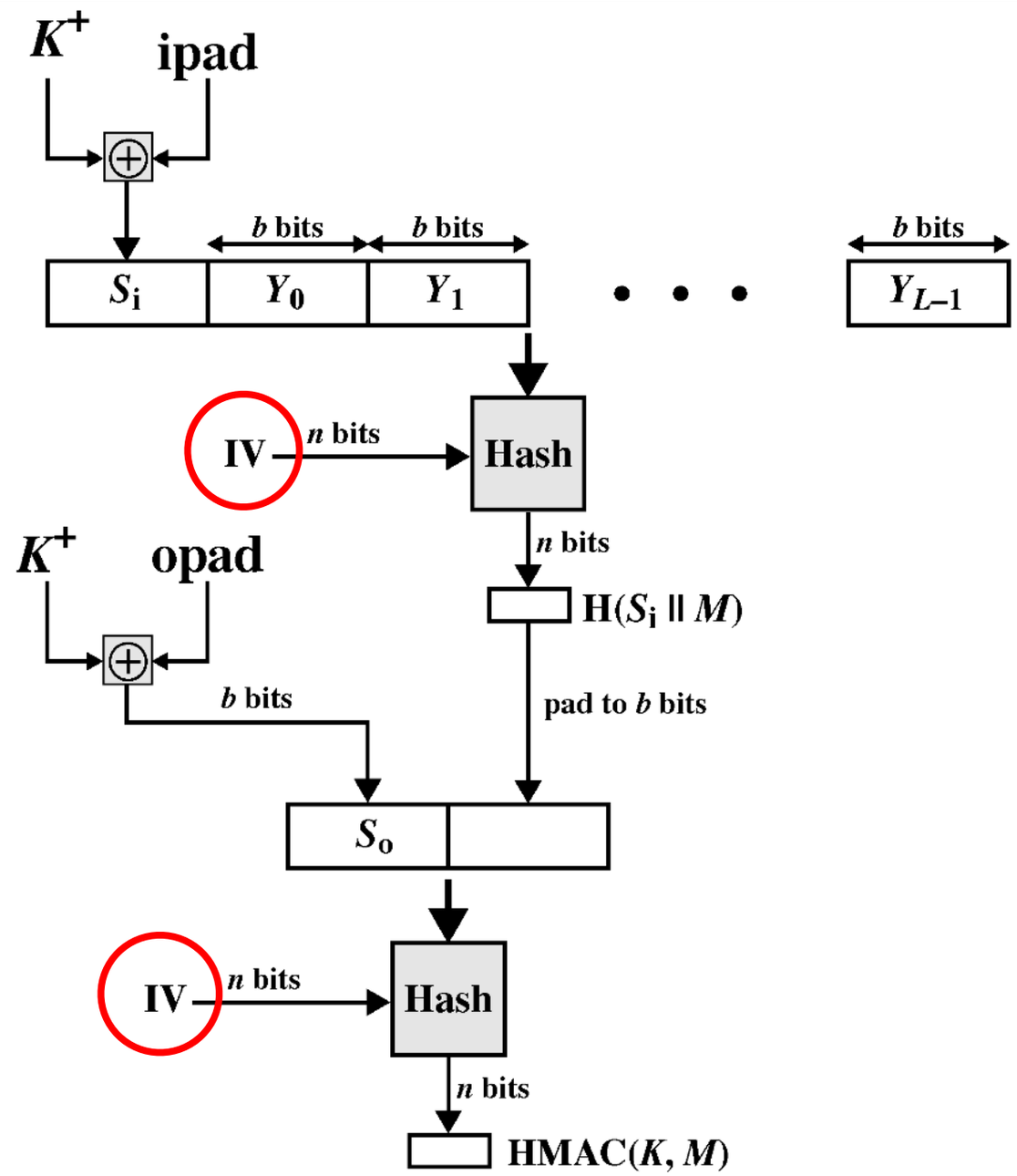



**Figure 3.6 HMAC Structure**

# Recommendation

- If you need just a MAC, use HMAC
- If you need encryption and a MAC, use AEAD
- See https://blog.cryptographyengineering.com/2013/02/15/why-i-hate-cbc-mac/