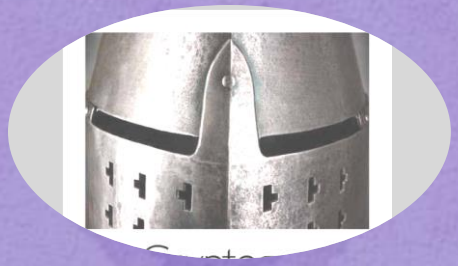


Chapter 12

Message Authentication Codes

Message Authentication

- Message Authentication is a procedure to verify that received messages come from the alleged source and have not been altered
  - User Authentication and Message Integrity
- Message Authentication may also verify sequencing and timeliness
- This module provides an overview of message authentication techniques and the various approaches to implementing them

Possible Attacks:

- 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 Requirements

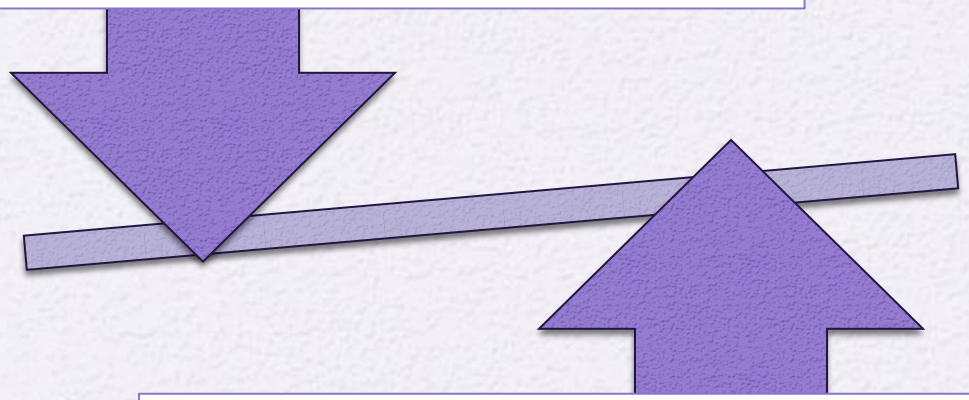
- Measures to deal with disclosure and traffic analysis attacks are in the realm of confidentiality
- Measures to deal with masquerade, content modification, sequence modification and timing modification attacks are in the realm of message authentication
- Measures to deal with source and destination repudiation issues are in the realm of digital signatures

Message Authentication Functions

- Two levels of

Lower level

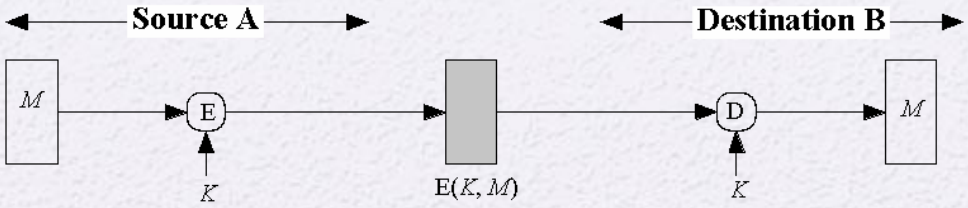
- There must be some sort of function that produces an authenticator



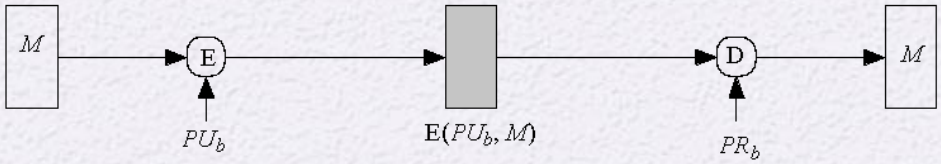
Higher-level

- Uses the lower-level function as a primitive in an authentication protocol that enables a receiver to verify the authenticity of a message

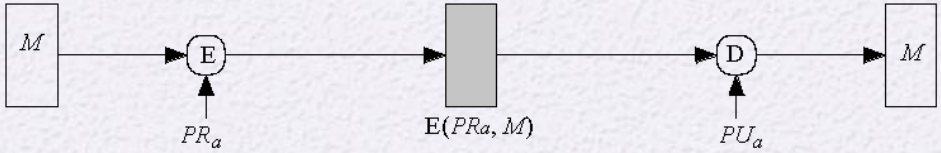
- Hash function
  - A function that maps a message of any length into a fixed-length hash value which serves as the authenticator
- Message encryption
  - The ciphertext of the entire message serves as its authenticator
- Message authentication code (MAC)
  - A function of the message and a secret key that produces a fixed-length value that serves as the authenticator



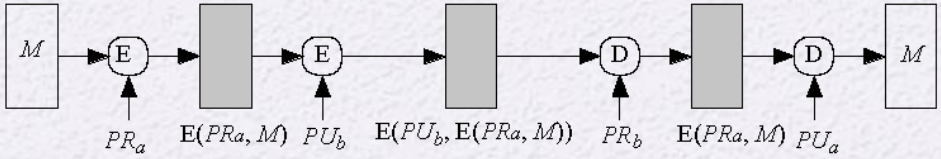
(a) Symmetric encryption: confidentiality and authentication



(b) Public-key encryption: confidentiality



(c) Public-key encryption: authentication and signature



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

Figure 12.1 Basic Uses of Message Encryption

Message Authentication with Encryption

- Message authentication provided by the encryption schemes (a) and (c) in the previous slide are only possible if the received message has an observable pattern
- Without an observable pattern in the message, encryption alone cannot provide message authentication
  - Binary and Multimedia files

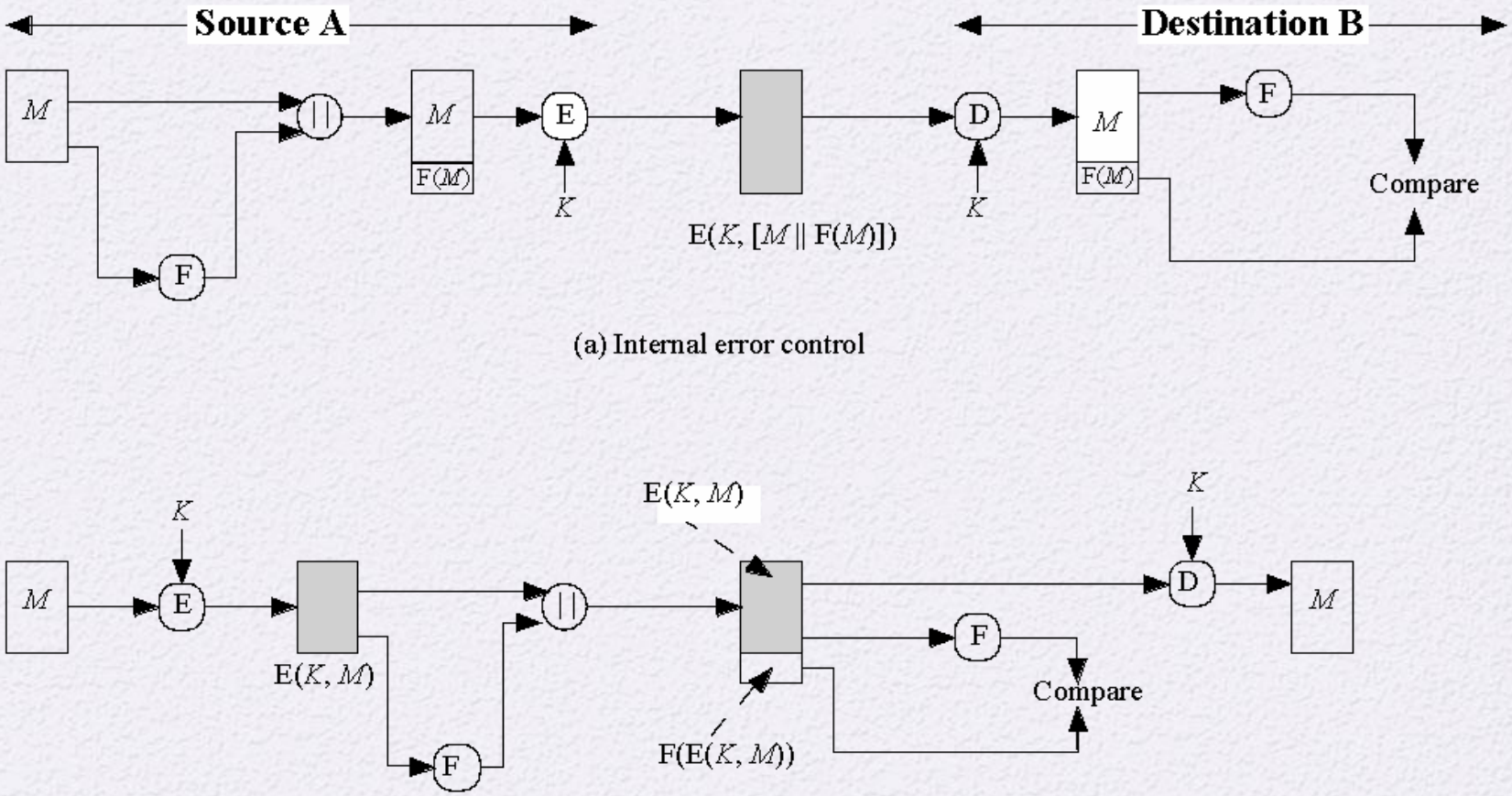


Figure 12.2 Internal and External Error Control

Message Authentication with Hash Function

- Message authentication can be provided by hash functions, using the hash value as an authenticator.
- The authenticator used in schemes (a) and (b) in the previous slide is the frame check sequence (FCS) used for error control in network communications
- While hash values are useful for message authentication, they need to be protected by encryption to prevent a man-in-the-middle attack
  - For this reason scheme (b) in the previous slide is not useful

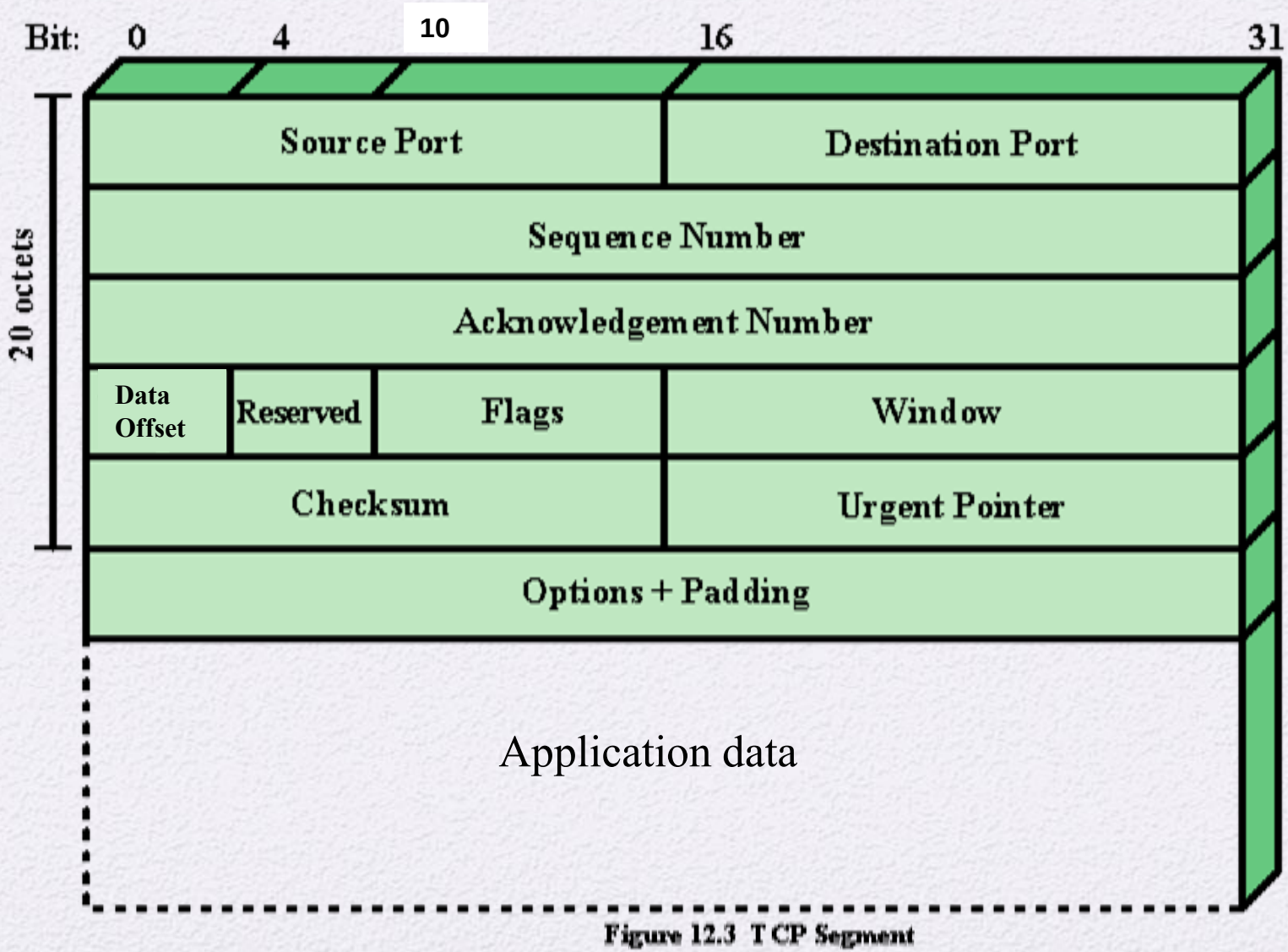


Figure 12.3 TCP Segment

Public-Key Encryption

- The straightforward use of public-key encryption provides **confidentiality** but not authentication
- To provide both confidentiality and authentication, User A can encrypt Message *M* first using its private key which provides the digital signature, and then using User B's public key, which provides confidentiality
- Disadvantage is that the public-key algorithm must be exercised four times rather than two in each communication

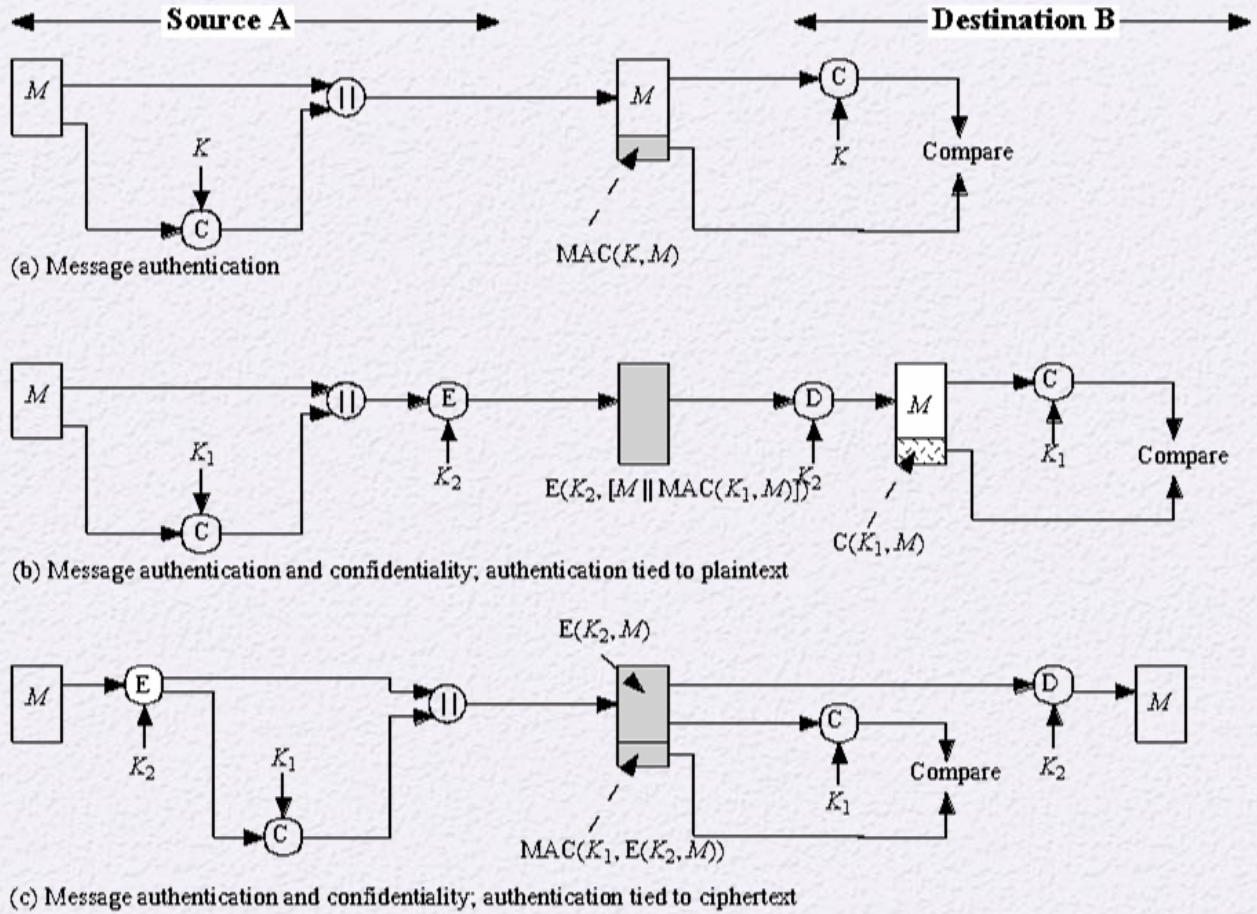


Figure 12.4 Basic Uses of Message Authentication Code (MAC)

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 MAC(*y*) = MAC(*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

Msg Authentication → Hash functions
→ Encryptions
→ MAC

$h = H(M)$

✓ (M, h) → $H(M) \stackrel{?}{=} h$ → (M, h) man in the middle attack

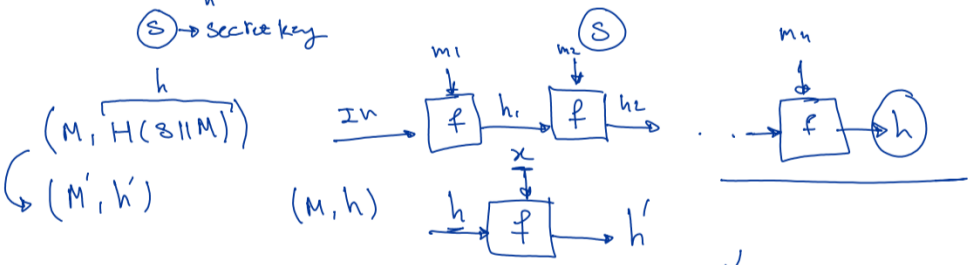
✗ (M', h') →

$(M, \underbrace{H(S \| M)}_{h})$ → $H(S \| M) \stackrel{?}{=} h$ → (M, h) length extension attack

S → secret key

$(M, \overbrace{H(S \| M)}^{h})$

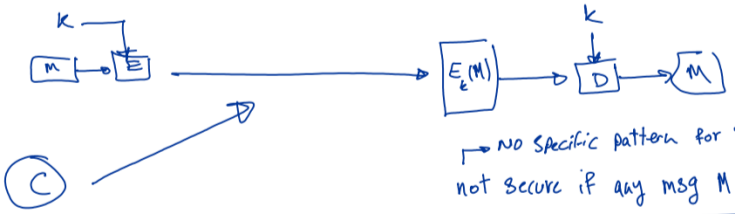
(M', h')



$f(h, x) = h'$

$H(S \| M) = h$

$H(\overbrace{S \| M \| \text{padding} \| x}^{M'}) = h'$



→ no specific pattern for M.

not secure if any msg M is a legitimate msg.

?

$E_K \left(\begin{array}{|c|} \hline M \\ \hline F(M) \\ \hline \end{array} \right)$ →

It is secure but not practical for applications that we only need authentication

Encryption & decryption are heavy computation compare to MAC

MACs Based on Hash Functions: HMAC

- There has been increased interest in developing a MAC derived from a cryptographic hash function
- Motivations:
  - Cryptographic hash functions such as MD5 and SHA generally execute faster in software than symmetric block ciphers such as DES
  - Library code for cryptographic hash functions is widely available
- HMAC has been chosen as the mandatory-to-implement MAC for IP security
- Has also been issued as a NIST standard (FIPS 198)

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

HMAC: Parameters

- H – embedded hash function
- IV – Initial value input to hash function
- K^+ - key value padded with zeroes
- b – block size of embedded hash function
- n – length of hash code
- k – length of key
- ipad = 0011 0110 (36) repeated b/8 times
- opad = 0101 1100 (5C) repeated b/8 times

Adv:

(M, t)

The adv needs to know the key K.

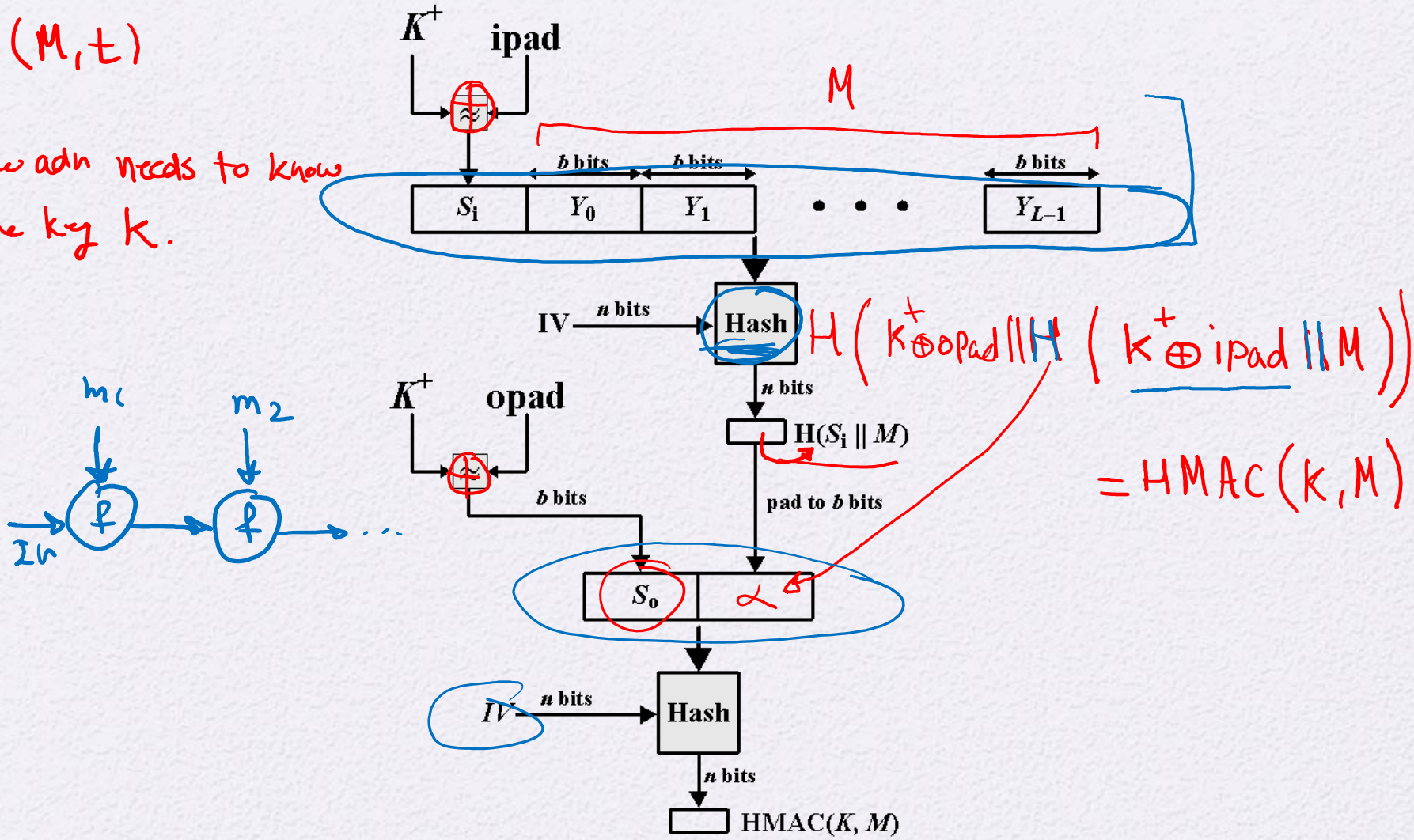


Figure 12.5 HMAC Structure

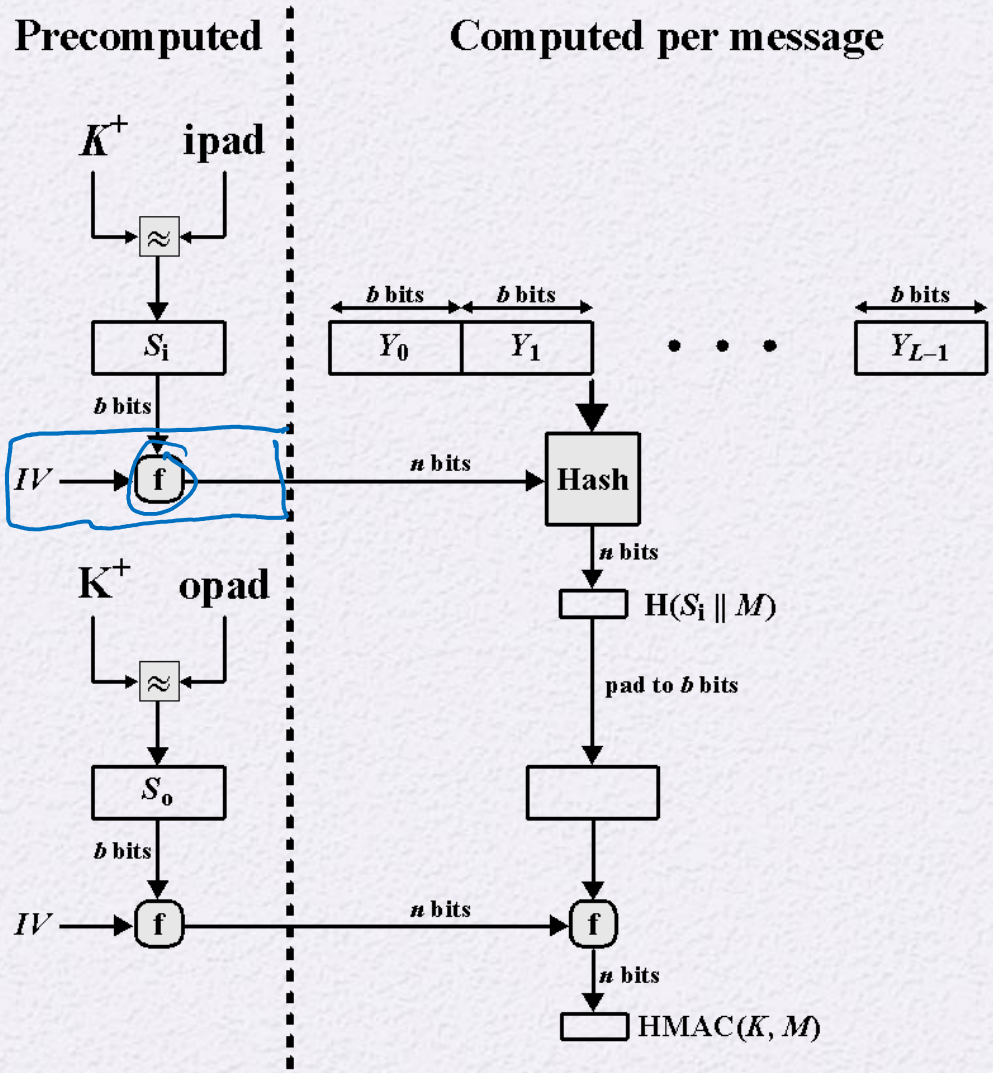


Figure 12.6 Efficient Implementation of HMAC

Security of HMAC

- Depends in some way on the cryptographic strength of the underlying hash function
- Appeal of HMAC is that its designers have been able to prove an exact relationship between the strength of the embedded hash function and the strength of HMAC
- Generally expressed in terms of the probability of successful forgery with a given amount of time spent by the forger and a given number of message-tag pairs created with the same key

Summary

- List and explain the possible attacks that are relevant to message authentication
- Define the term *message authentication code*

- List and explain the requirements for a message authentication code
- Present an overview of HMAC