Week 6

Lecture 1

Un Keyed Cryptography: Hash Functions

Lecture 2

Message Authentication Codes or Keyed Hash Function
Additional Material on HMAC from textbook

Workshop 3: Workshop based on Lectures in Week5

Quiz 6

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

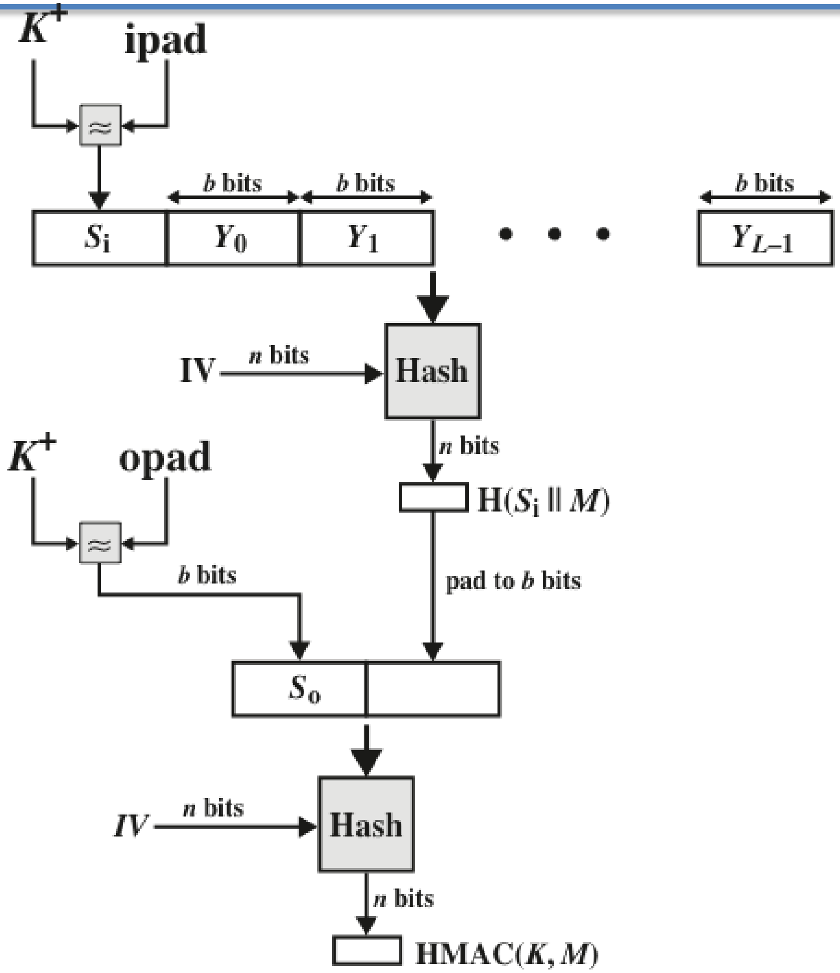


Figure 12.5 HMAC Structure

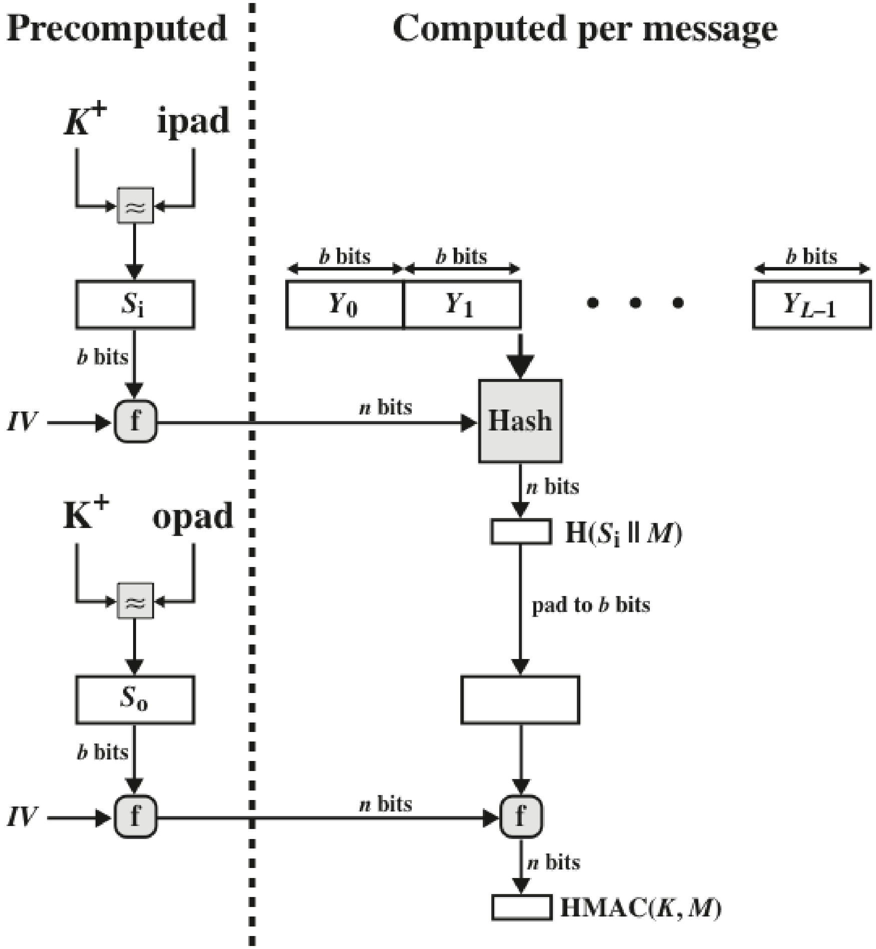


Figure 12.6 Efficient Implementation of HMAC

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

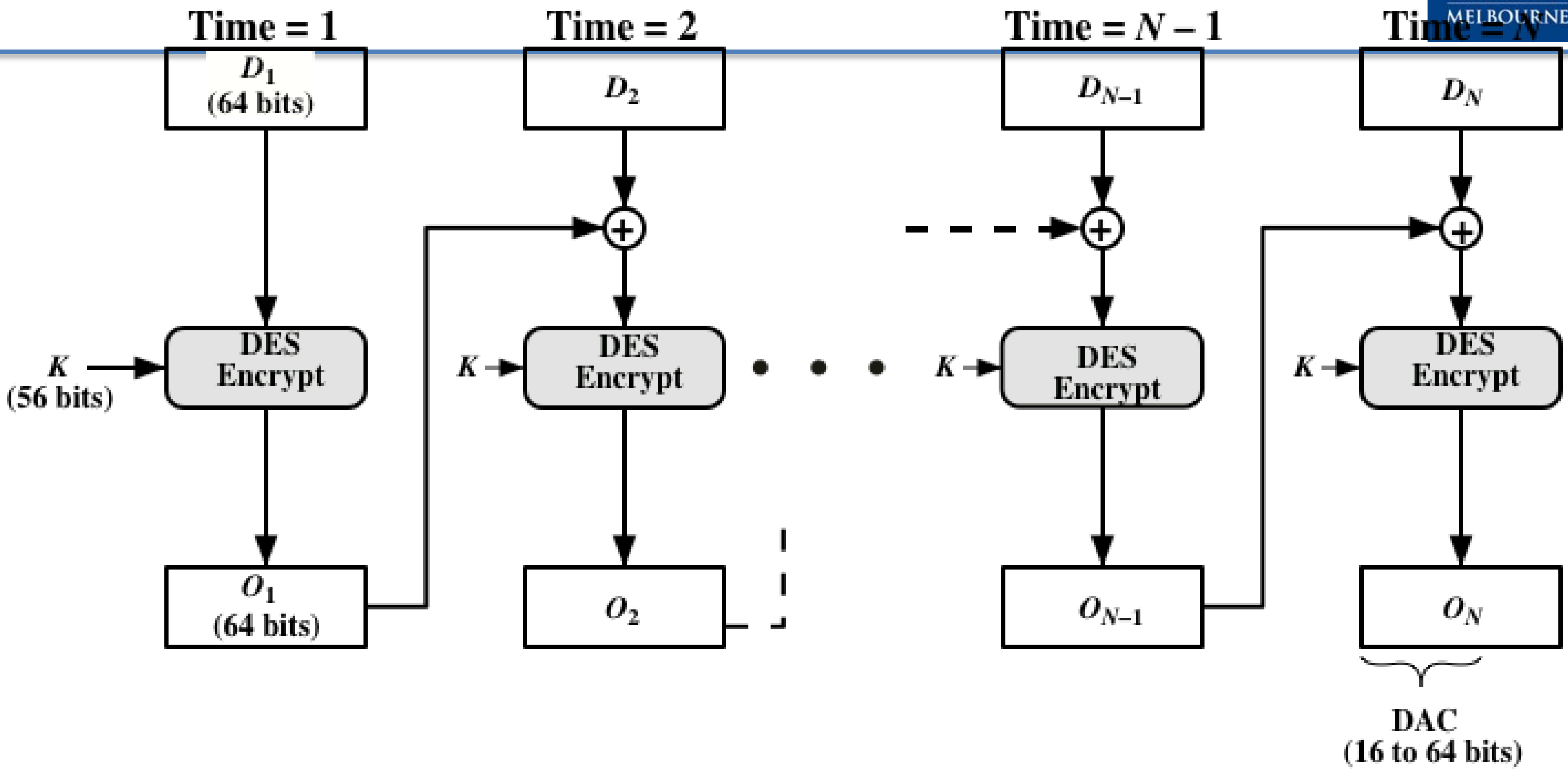
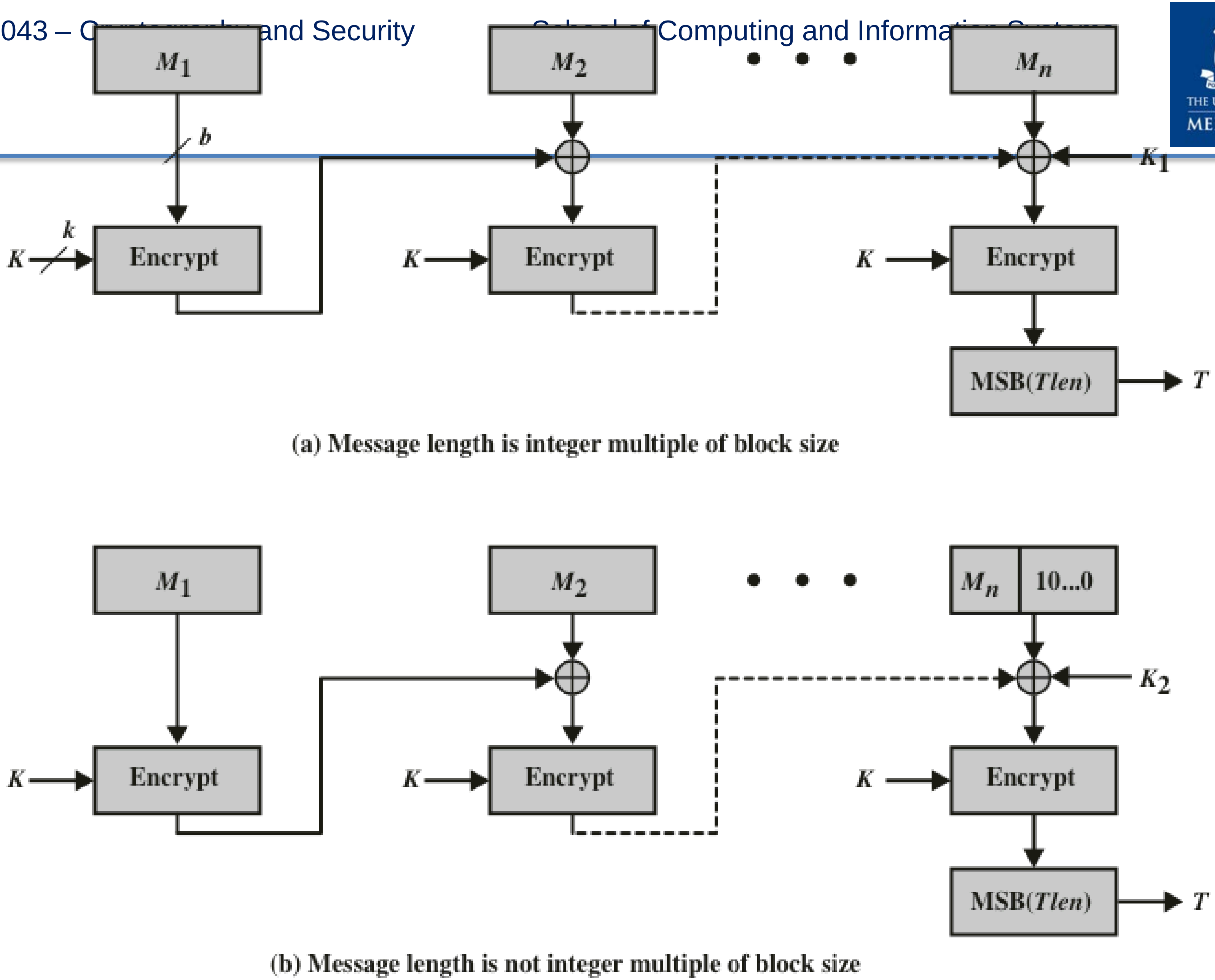


Figure 12.7 Data Authentication Algorithm (FIPS PUB 113)



(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)

Authenticated Encryption (AE)

- A term used to describe encryption systems that simultaneously protect confidentiality and authenticity of communications
- Approaches:
  - Hashing followed by encryption
  - Authentication followed by encryption
  - Encryption followed by authentication
  - Independently encrypt and authenticate
- Both decryption and verification are straightforward for each approach
- There are security vulnerabilities with all of these approaches

Counter with Cipher Block Chaining-Message Authentication Code (CCM)

- Was standardized by NIST specifically to support the security requirements of IEEE 802.11 WiFi wireless local area networks
- Variation of the encrypt-and-MAC approach to authenticated encryption
  - Defined in NIST SP 800-38C
- Key algorithmic ingredients:
  - AES encryption algorithm
  - CTR mode of operation
  - CMAC authentication algorithm
- Single key *K* is used for both encryption and MAC algorithms

The input to the CCM encryption process consists of three elements:

Data that will be both authenticated and encrypted

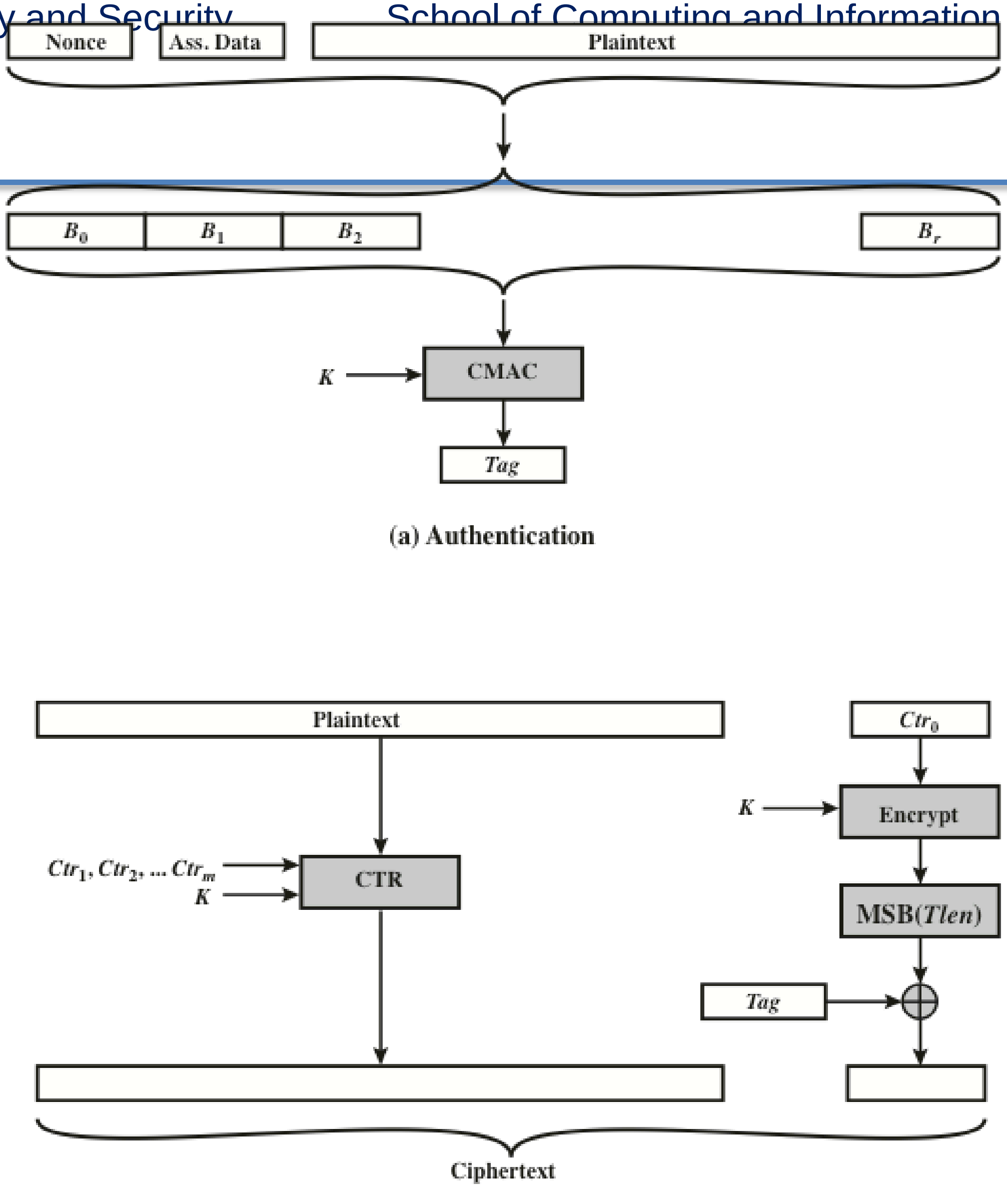
This is the plaintext message *P* of the data block

Associated data *A* that will be authenticated but not encrypted

An example is a protocol header that must be transmitted in the clear for proper protocol operation but which needs to be authenticated

A nonce *N* that is assigned to the payload and the associated data

This is a unique value that is different for every instance during the lifetime of a protocol association and is intended to prevent replay attacks and certain other types of attacks

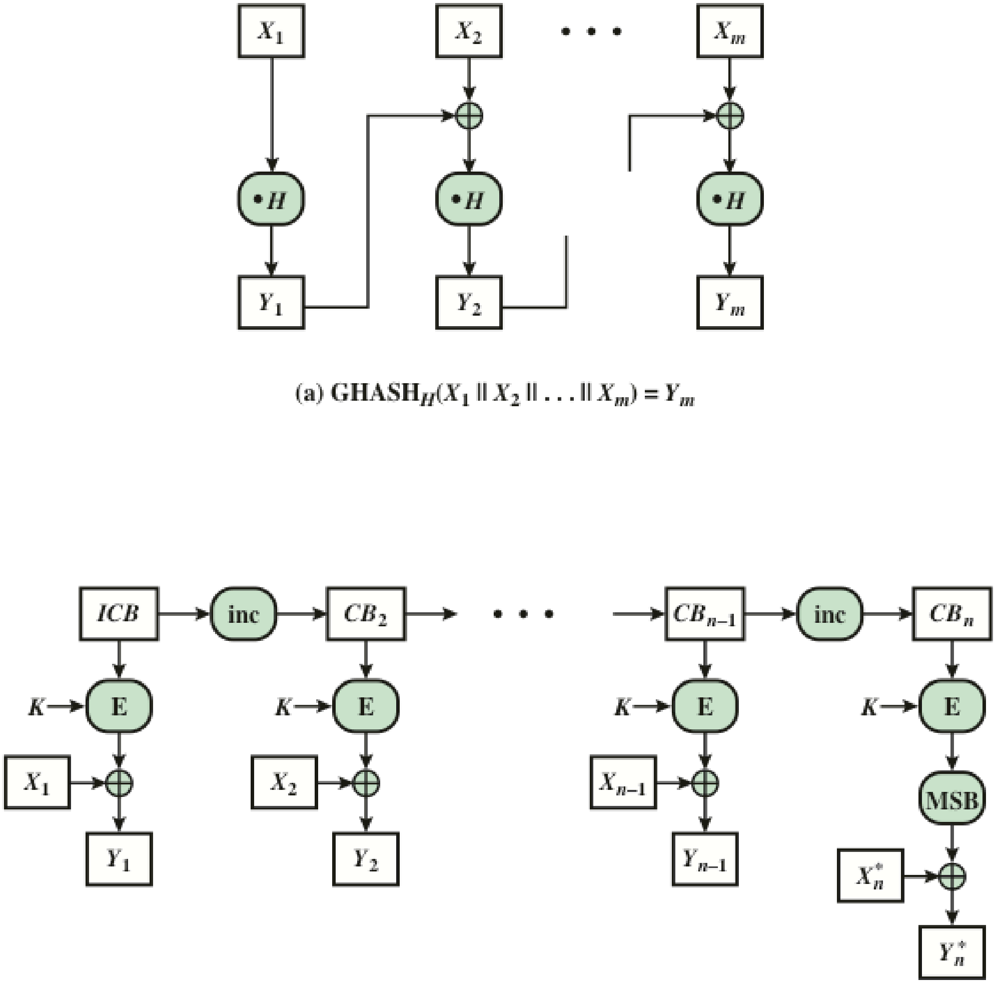


(b) Encryption

Figure 12.9 Counter with Cipher Block Chaining-Message Authentication Code (CCM)

Galois/Counter Mode (GCM)

- NIST standard SP 800-38D
- Designed to be parallelizable so that it can provide high throughput with low cost and low latency
  - Message is encrypted in variant of CTR mode
  - Resulting ciphertext is multiplied with key material and message length information over GF (2^{128}) to generate the authenticator tag
  - The standard also specifies a mode of operation that supplies the MAC only, known as GMAC
- Makes use of two functions:
  - GHASH - a keyed hash function
  - GCTR - CTR mode with the counters determined by simple increment by one operation



(a) $\mathrm{GHASH}_H(X_1 \| X_2 \| \ldots \| X_m) = Y_m$

(b) $\mathrm{GCTR}_K(ICB, X_1 \| X_2 \| \ldots \| X_n) = Y_1 \| Y_2 \| \ldots \| Y_n$

Figure 12.10 GCM Authentication and Encryption Functions

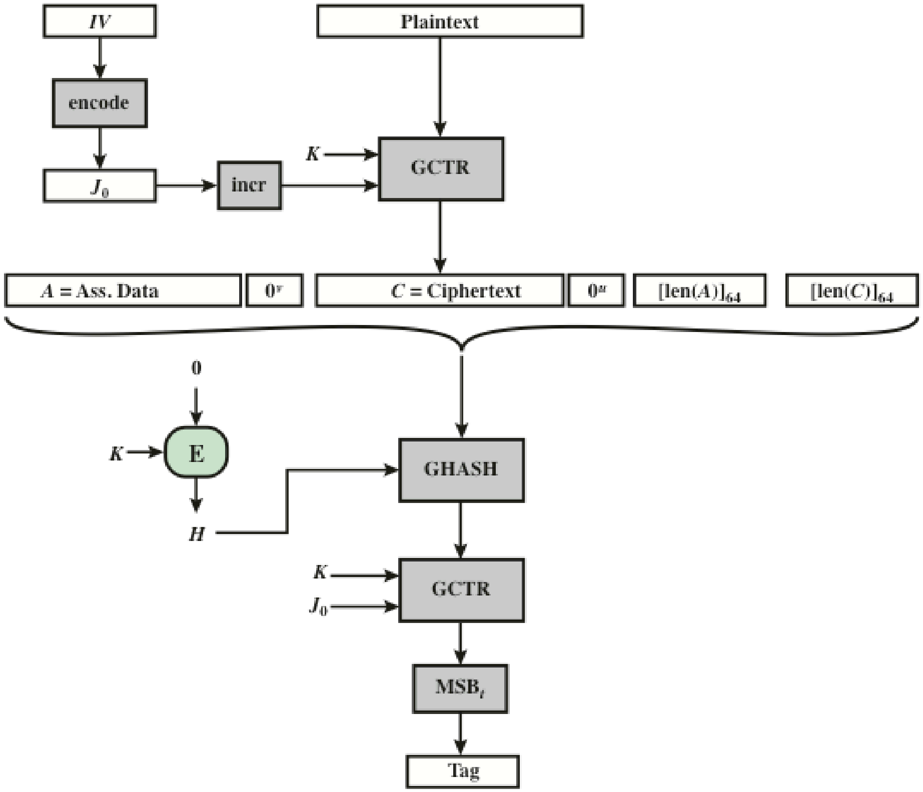


Figure 12.11 Galois Counter - Message Authentication Code (GCM)

Key Wrap (KW)

- Most recent block cipher mode of operation defined by NIST
  - Uses AES or triple DEA as the underlying encryption algorithm
- Purpose is to securely exchange a symmetric key to be shared by two parties, using a symmetric key already shared by those parties
  - The latter key is called a *key encryption key* (KEK)

– Robust in the sense that each bit of output can be expected to depend in a nontrivial fashion on each bit of input

– Only used for small amounts of plaintext

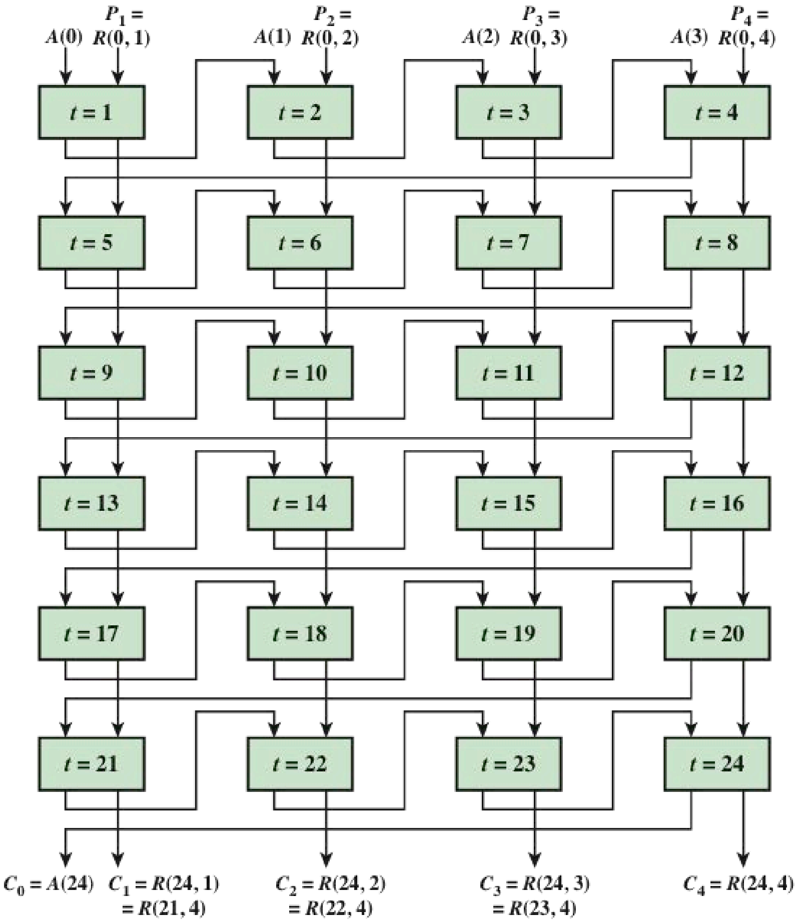


Figure 12.12 Key Wrapping Operation for 256-bit Key

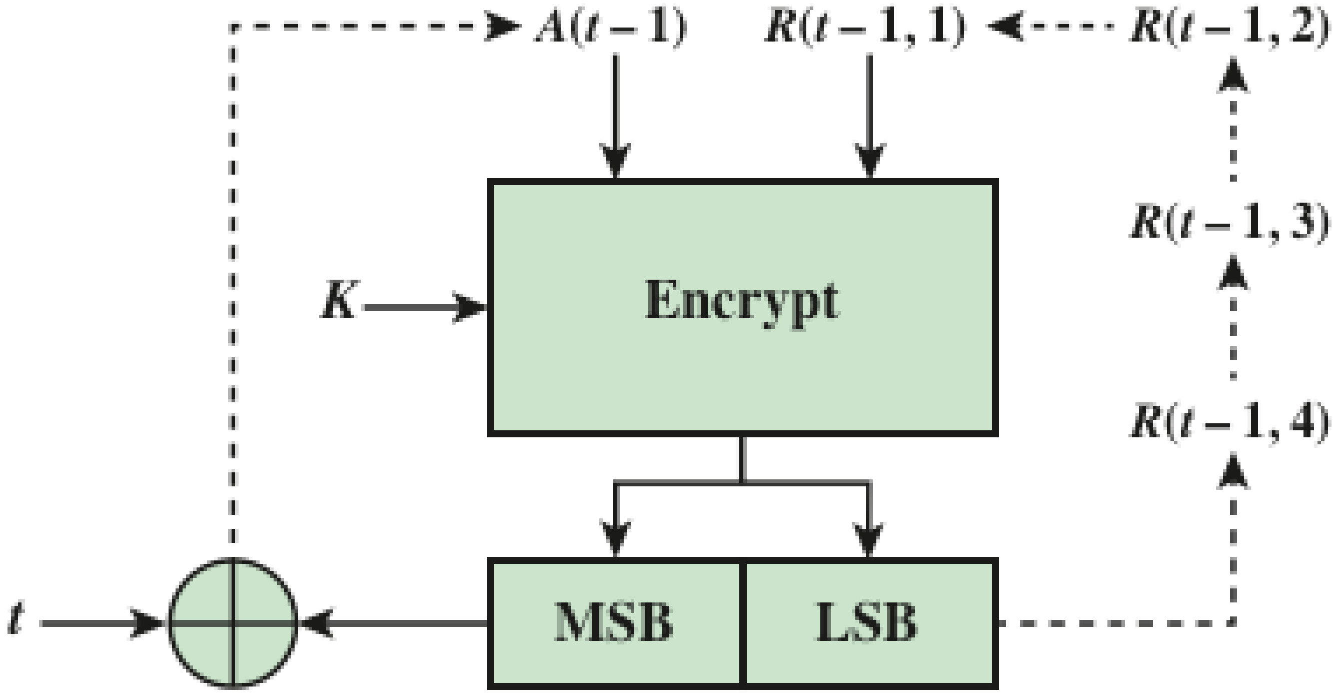
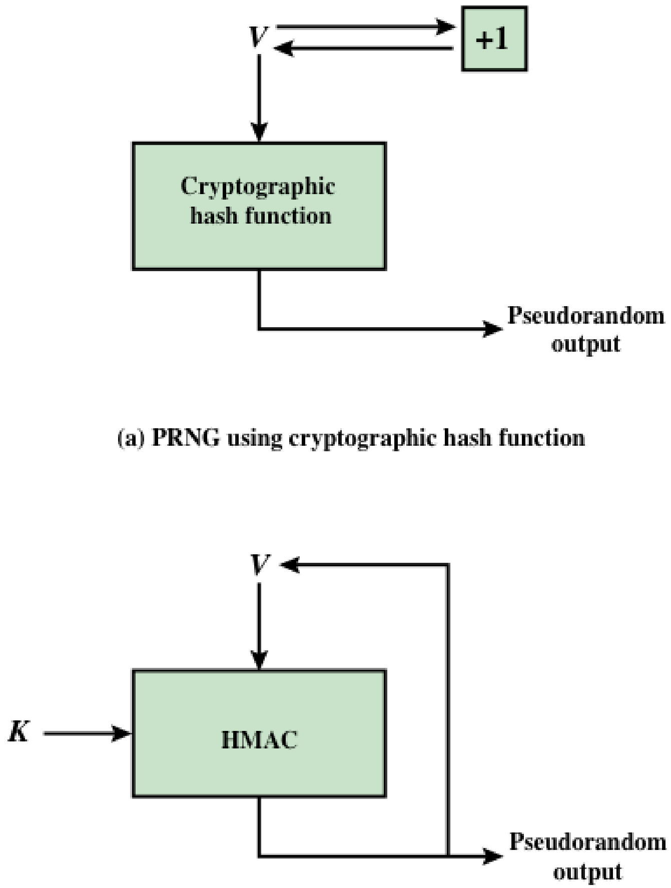


Figure 12.13 Key Wrapping Operation for 256-bit Key: stage t

Pseudorandom Number Generation Using Hash Functions and MACs

- Essential elements of any pseudorandom number generator (PRNG) are a seed value and a deterministic algorithm for generating a stream of pseudorandom bits
  - If the algorithm is used as a pseudorandom function (PRF) to produce a required value, the seed should only be known to the user of the PRF
  - If the algorithm is used to produce a stream encryption function, the seed has the role of a secret key that must be known to the sender and the receiver

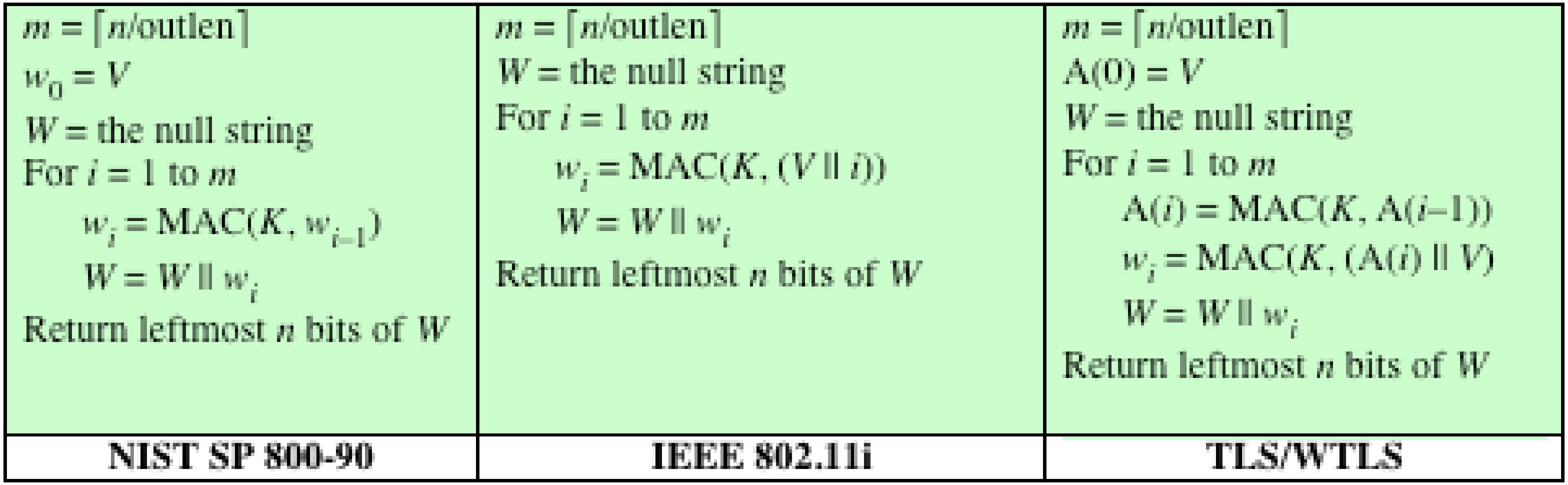
– A hash function or MAC produces apparently random output and can be used to build a PRNG



(a) PRNG using cryptographic hash function

(b) PRNG using HMAC

Figure 12.14 Basic Structure of Hash-Based PRNGs (SP 800-90)



$m = \lceil n/\text{outlen} \rceil$ $w_0 = V$ W = the null string For i = 1 to m $\quad w_i = \text{MAC}(K, w_{i-1})$ $\quad W = W \Vert w_i$ Return leftmost n bits of W	$m = \lceil n/\text{outlen} \rceil$ W = the null string For i = 1 to m $\quad w_i = \text{MAC}(K, (V \Vert i))$ $\quad W = W \Vert w_i$ Return leftmost n bits of W	$m = \lceil n/\text{outlen} \rceil$ $\text{A}(0) = V$ W = the null string For i = 1 to m $\quad \text{A}(i) = \text{MAC}(K, \text{A}(i-1))$ $\quad w_i = \text{MAC}(K, (\text{A}(i) \Vert V)$ $\quad W = W \Vert w_i$ Return leftmost n bits of W
NIST SP 800-90	**IEEE 802.11i**	**TLS/WTLS**

Figure 12.15 Three PRNGs Based on HMAC