

# Cryptography and Network Security Chapter 12

Fourth Edition  
by William Stallings

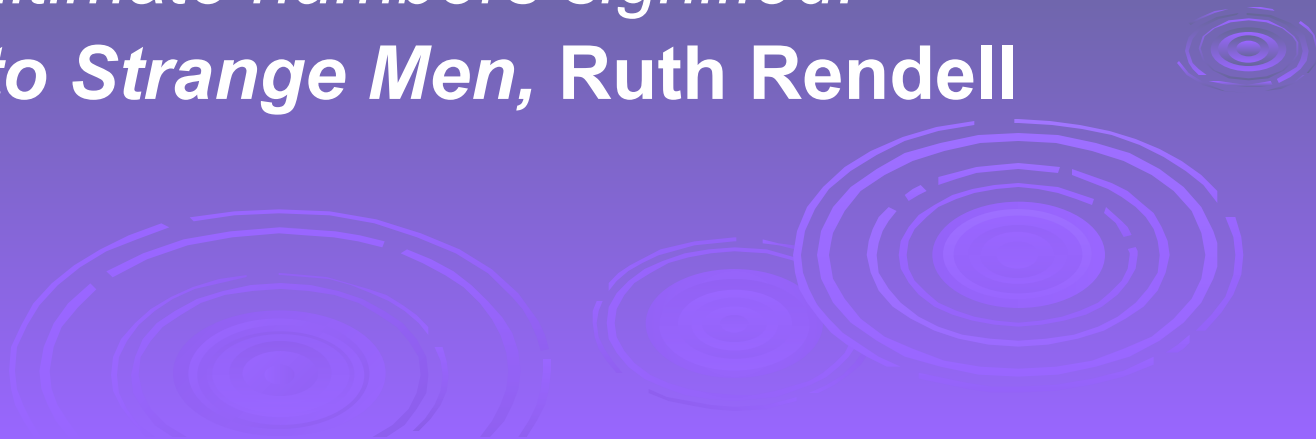
Lecture slides by Lawrie Brown

The background of the slide features several sets of concentric circles in a lighter shade of purple, resembling ripples in water. These circles are positioned in the lower right and bottom center areas of the slide.

# Chapter 12 – Hash and MAC Algorithms

*Each of the messages, like each one he had ever read of Stern's commands, began with a number and ended with a number or row of numbers. No efforts on the part of Mungo or any of his experts had been able to break Stern's code, nor was there any clue as to what the preliminary number and those ultimate numbers signified.*

**—Talking to Strange Men, Ruth Rendell**



# Hash and MAC Algorithms

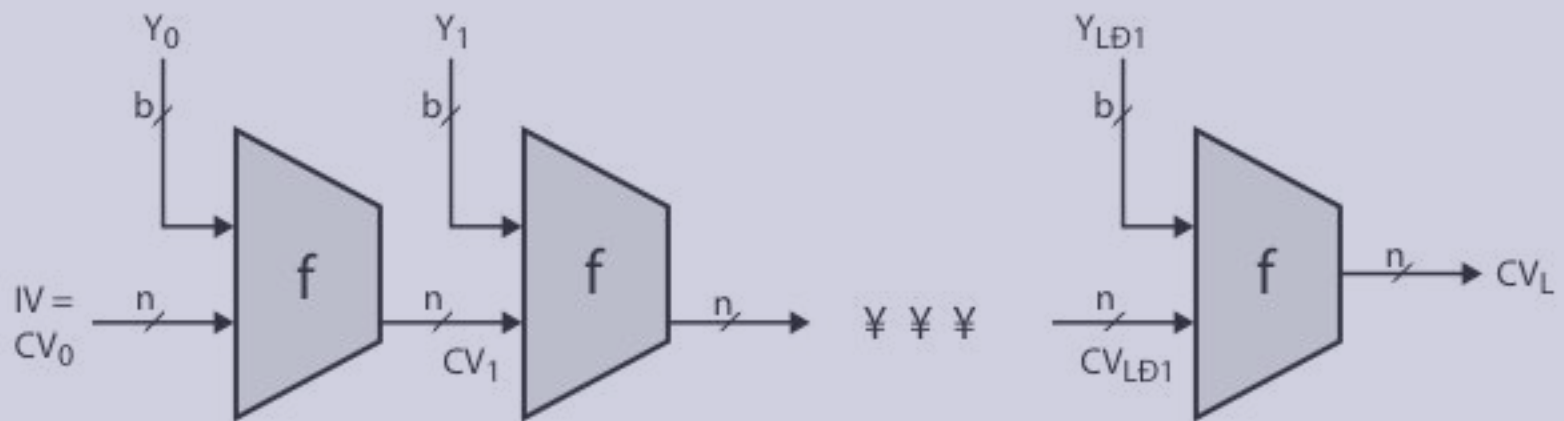
## □ Hash Functions

- condense arbitrary size message to fixed size
- by processing message in blocks
- through some compression function
- either custom or block cipher based

## □ Message Authentication Code (MAC)

- fixed sized authenticator for some message
- to provide authentication for message
- by using block cipher mode or hash function

# Hash Algorithm Structure



$IV$  = Initial value  
 $CV_i$  = chaining variable  
 $Y_i$  =  $i$ th input block  
 $f$  = compression algorithm

$L$  = number of input blocks  
 $n$  = length of hash code  
 $b$  = length of input block

# Secure Hash Algorithm

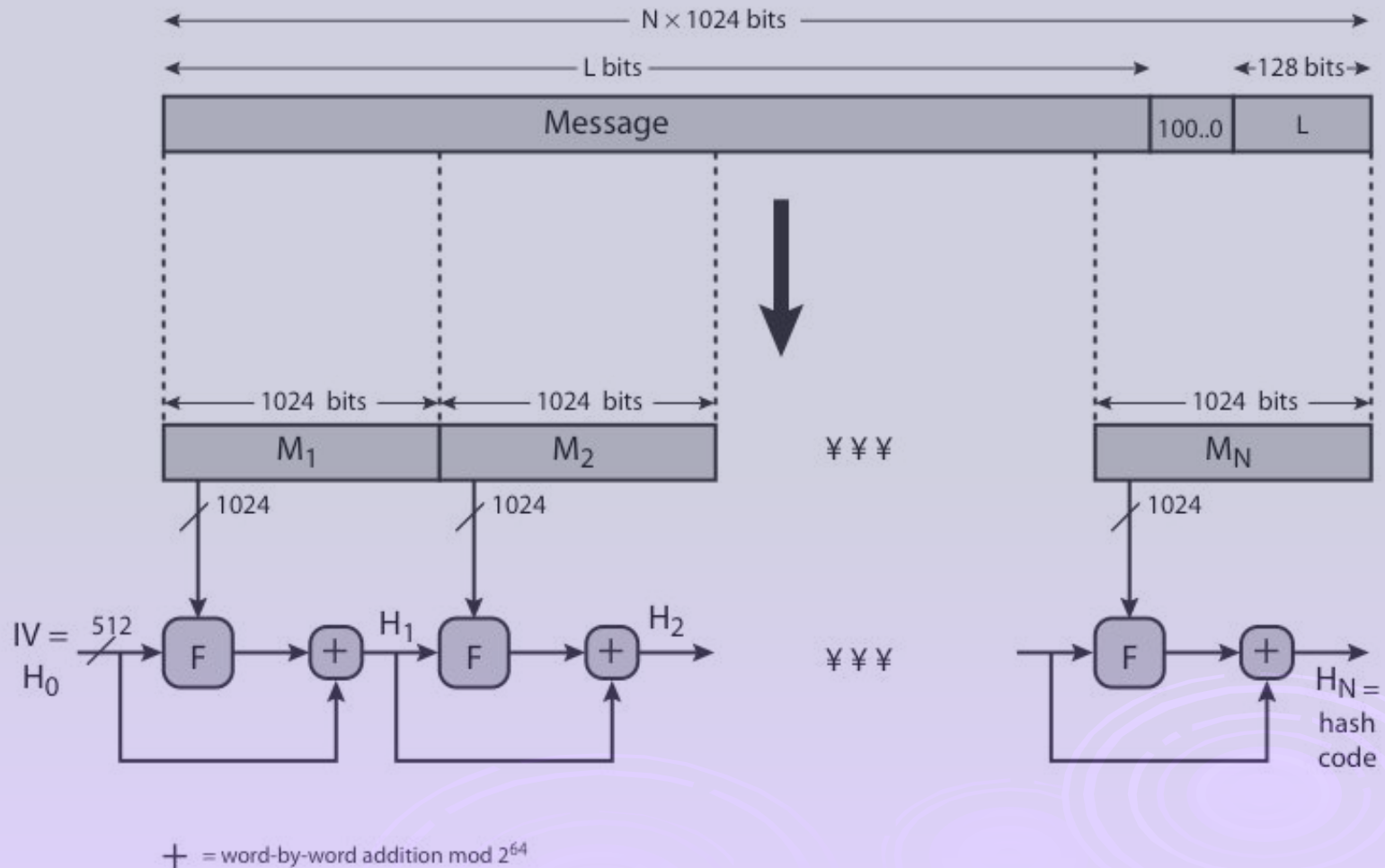
- ❑ SHA originally designed by NIST & NSA in 1993
- ❑ was revised in 1995 as SHA-1
- ❑ US standard for use with DSA signature scheme
  - standard is FIPS 180-1 1995, also Internet RFC3174
  - nb. the algorithm is SHA, the standard is SHS
- ❑ based on design of MD4 with key differences
- ❑ produces 160-bit hash values
- ❑ recent 2005 results on security of SHA-1 have raised concerns on its use in future applications

# Revised Secure Hash Standard

- ❑ NIST issued revision FIPS 180-2 in 2002
- ❑ adds 3 additional versions of SHA
  - SHA-256, SHA-384, SHA-512
- ❑ designed for compatibility with increased security provided by the AES cipher
- ❑ structure & detail is similar to SHA-1
- ❑ hence analysis should be similar
- ❑ but security levels are rather higher

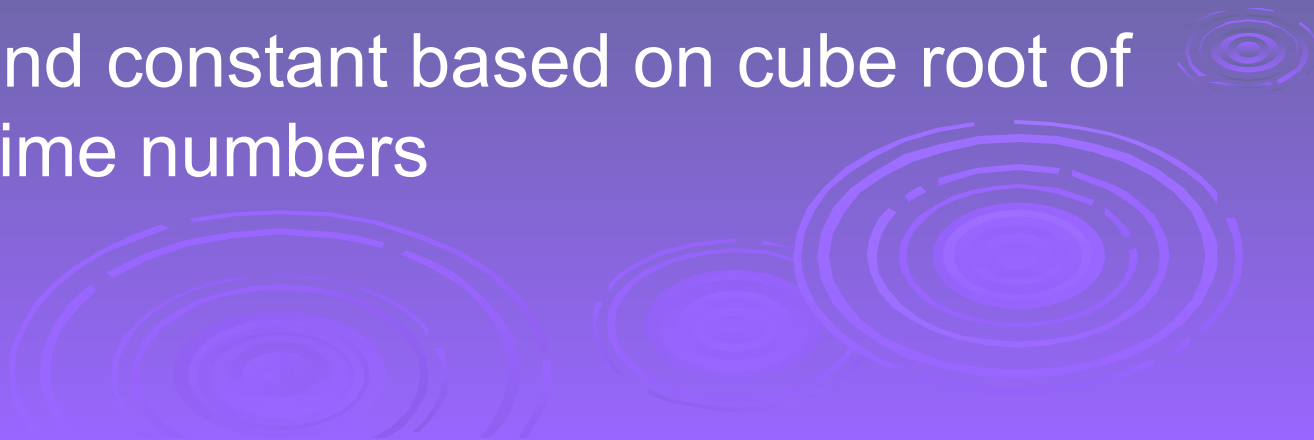


# SHA-512 Overview



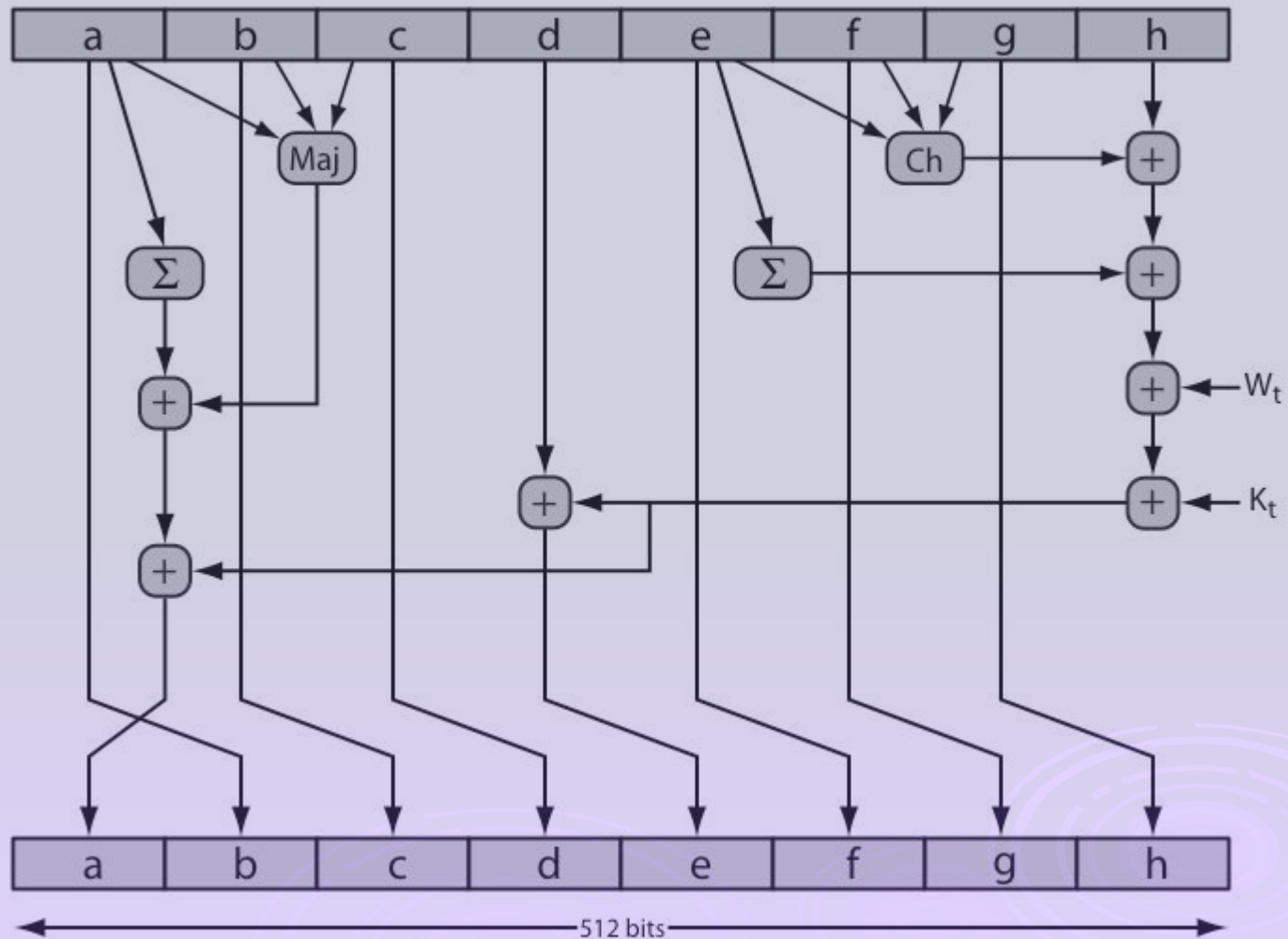
# SHA-512 Compression Function

- heart of the algorithm
- processing message in 1024-bit blocks
- consists of 80 rounds
  - updating a 512-bit buffer
  - using a 64-bit value  $W_t$  derived from the current message block
  - and a round constant based on cube root of first 80 prime numbers

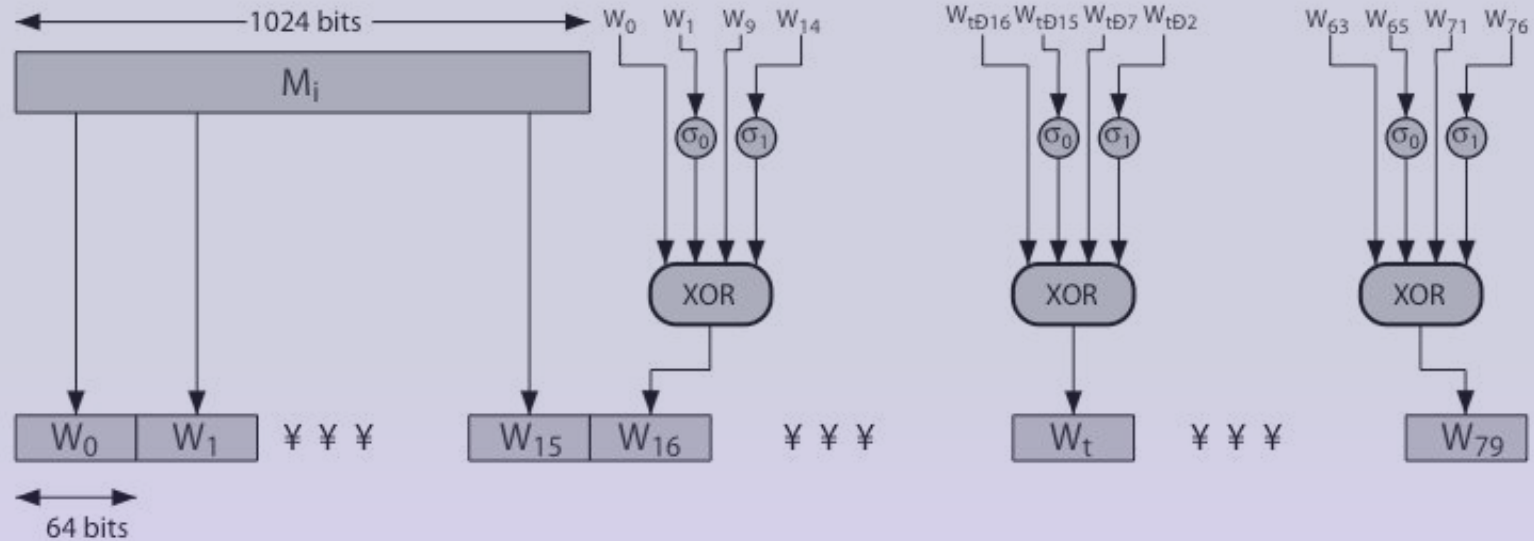




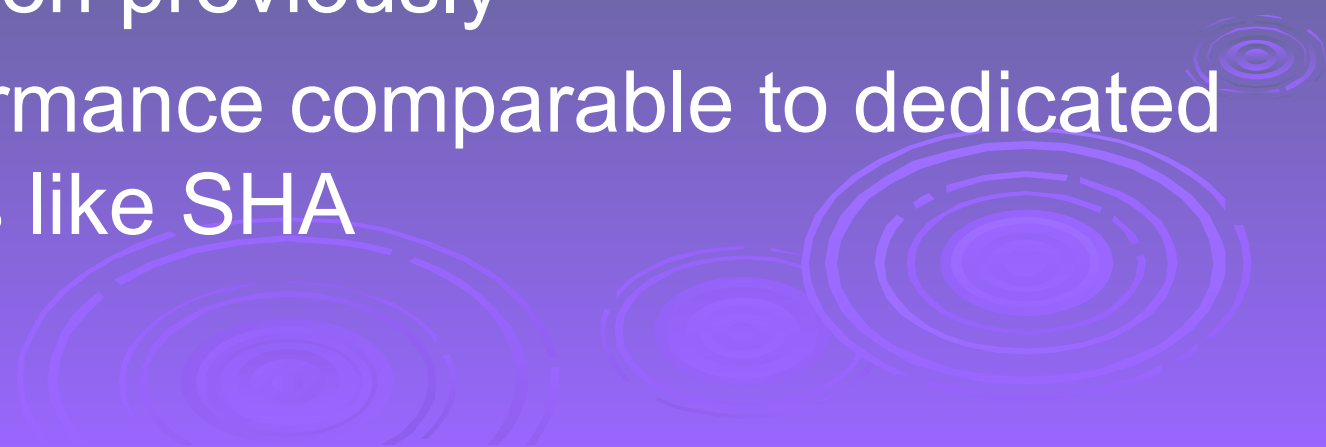
# SHA-512 Round Function



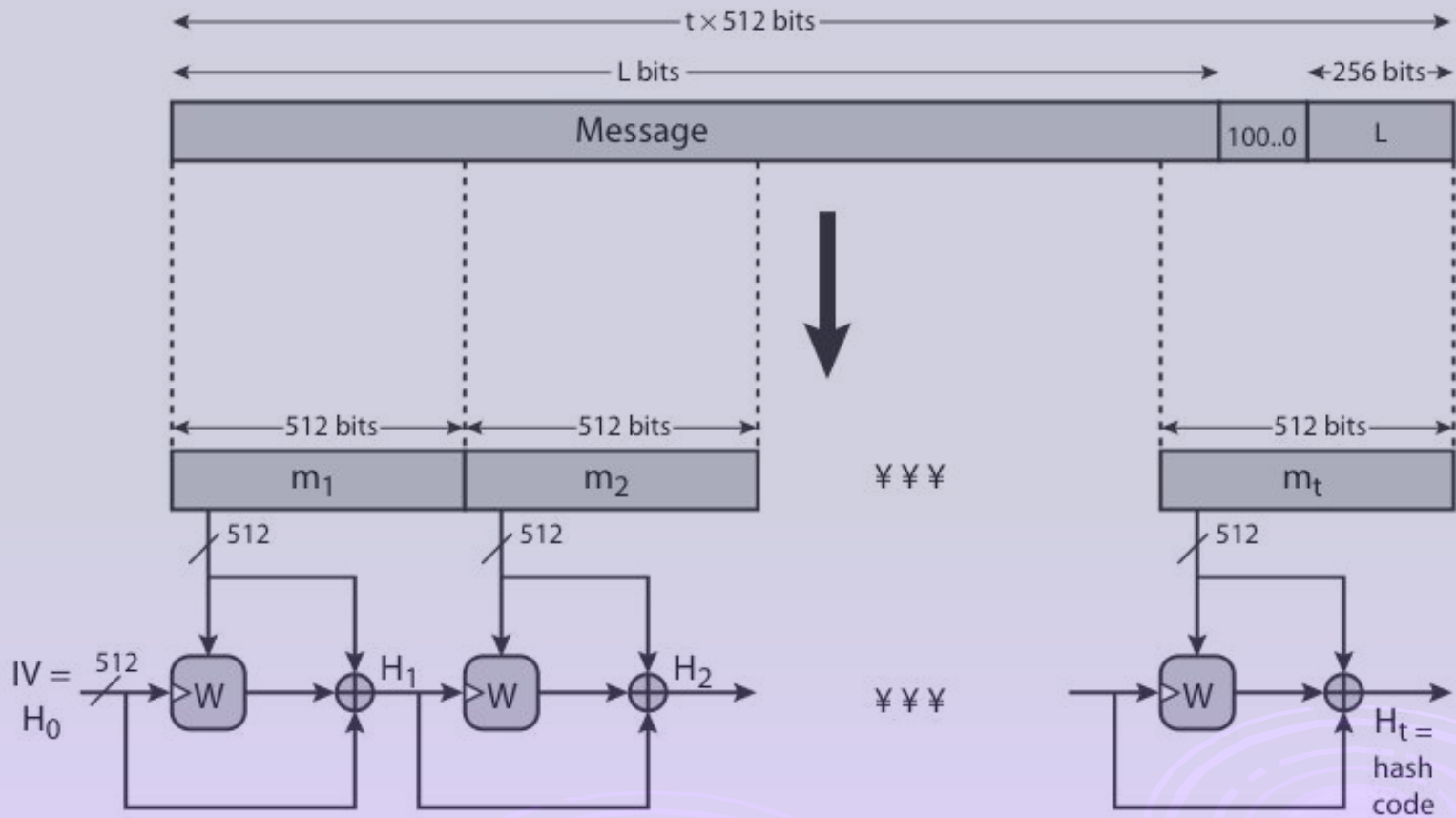
# SHA-512 Round Function



# Whirlpool

- now examine the Whirlpool hash function
  - endorsed by European NESSIE project
  - uses modified AES internals as compression function
  - addressing concerns on use of block ciphers seen previously
  - with performance comparable to dedicated algorithms like SHA
- 

# Whirlpool Overview

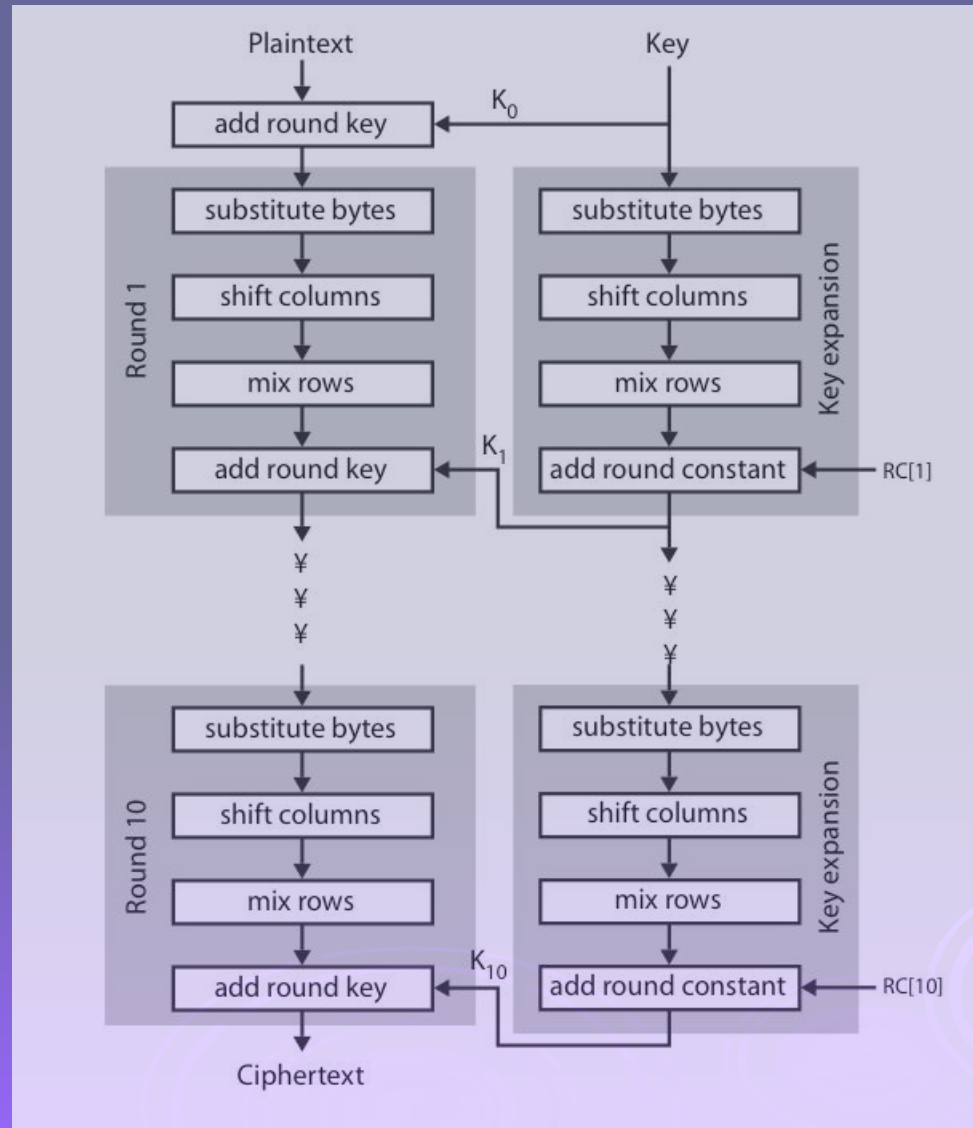


Note: triangular hatch marks key input

# Whirlpool Block Cipher W

- ❑ designed specifically for hash function use
- ❑ with security and efficiency of AES
- ❑ but with 512-bit block size and hence hash
- ❑ similar structure & functions as AES but
  - input is mapped row wise
  - has 10 rounds
  - a different primitive polynomial for  $GF(2^8)$
  - uses different S-box design & values

# Whirlpool Block Cipher W



# Whirlpool Performance & Security

- Whirlpool is a very new proposal
- hence little experience with use
- but many AES findings should apply
- does seem to need more h/w than SHA, but with better resulting performance



# Keyed Hash Functions as MACs

- want a MAC based on a hash function
  - because hash functions are generally faster
  - code for crypto hash functions widely available

- hash includes a key along with message

- original proposal:

$\text{KeyedHash} = \text{Hash}(\text{Key} | \text{Message})$

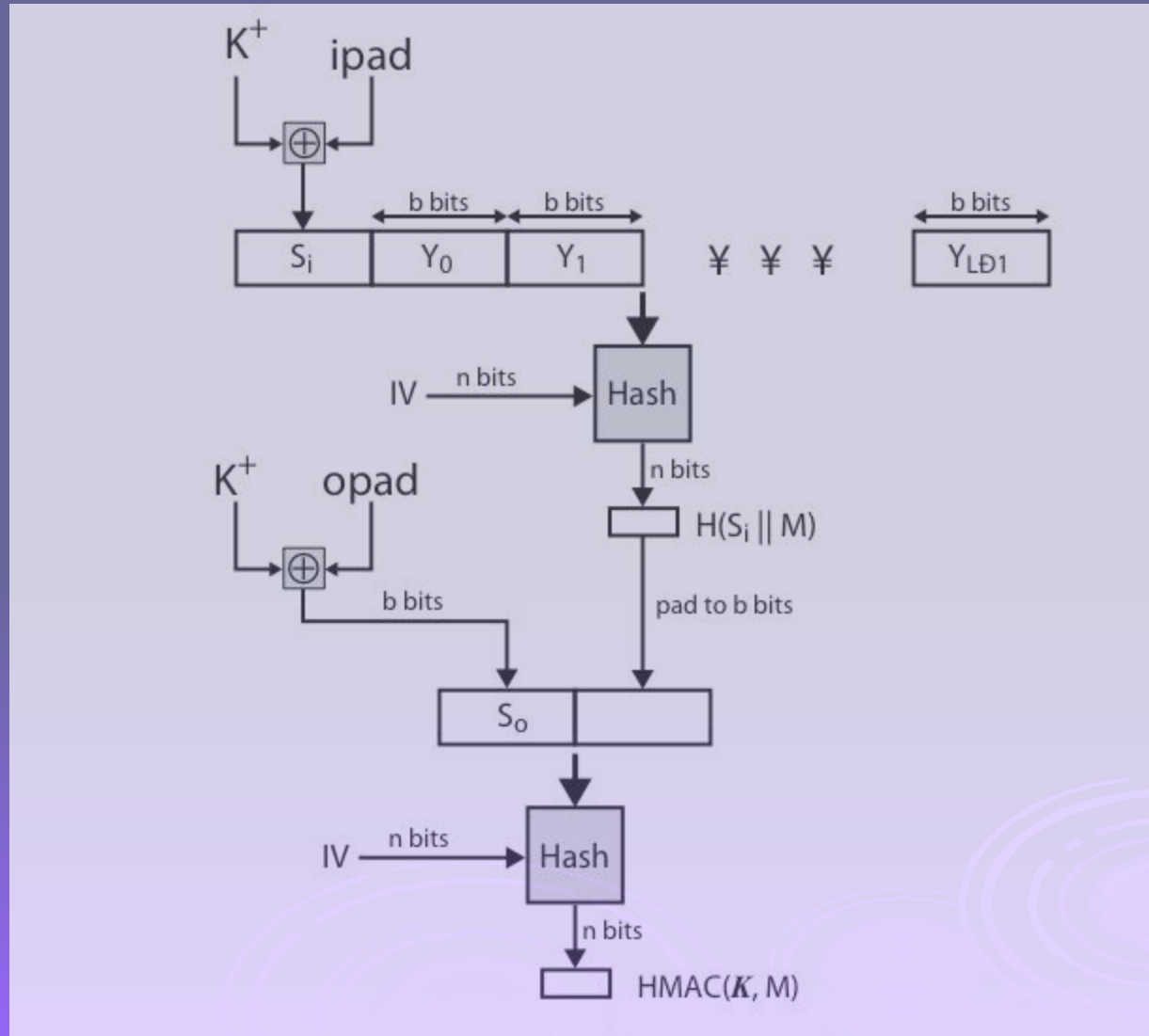
- some weaknesses were found with this
- eventually led to development of HMAC



# HMAC

- specified as Internet standard RFC2104
- uses hash function on the message:
$$\text{HMAC}_K = \text{Hash}[(K^+ \text{ XOR opad}) \parallel \text{Hash}[(K^+ \text{ XOR ipad}) \parallel M]]$$
- where  $K^+$  is the key padded out to size
- and opad, ipad are specified padding constants
- overhead is just 3 more hash calculations than the message needs alone
- any hash function can be used
  - eg. MD5, SHA-1, RIPEMD-160, Whirlpool

# HMAC Overview



# HMAC Security

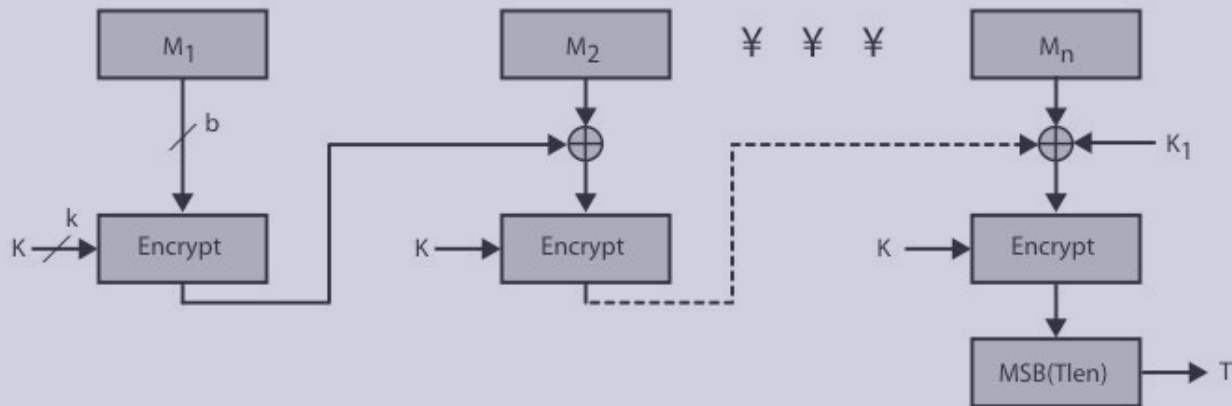
- proved security of HMAC relates to that of the underlying hash algorithm
- attacking HMAC requires either:
  - brute force attack on key used
  - birthday attack (but since keyed would need to observe a very large number of messages)
- choose hash function used based on speed verses security constraints



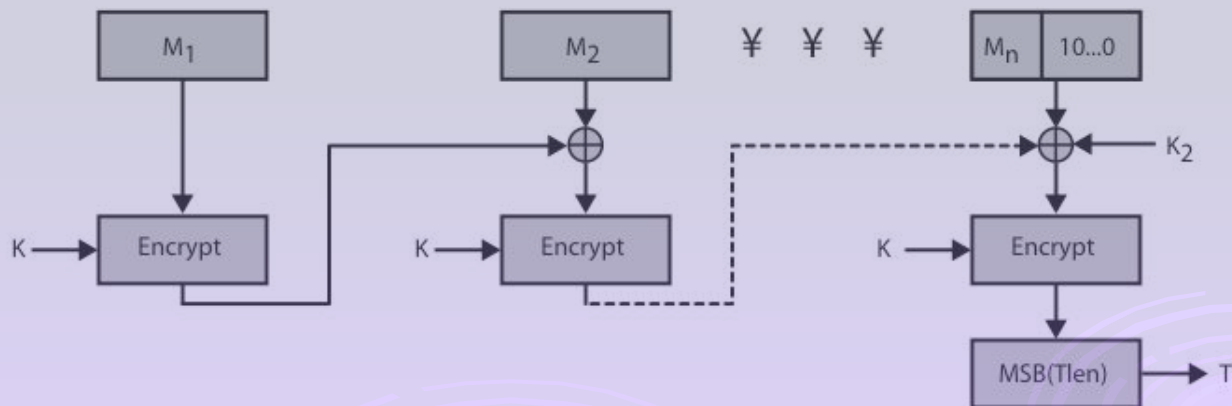
# CMAC

- ❑ previously saw the DAA (CBC-MAC)
- ❑ widely used in govt & industry
- ❑ but has message size limitation
- ❑ can overcome using 2 keys & padding
- ❑ thus forming the Cipher-based Message Authentication Code (CMAC)
- ❑ adopted by NIST SP800-38B

# CMAC Overview



(a) Message length is integer multiple of block size



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

Figure 12.12 Cipher-Based Message Authentication Code (CMAC)

# Summary

- have considered:
  - some current hash algorithms
    - SHA-512 & Whirlpool
  - HMAC authentication using hash function
  - CMAC authentication using a block cipher

