CSEN1001

Computer and Network Security

Mervat AbuElkheir
Ahmad Helmy
Mohamed Abdelrazik

Lecture (9)

Message Authentication

Message Authentication

- ❑ Encryption protects secrecy of message
- ❑ Key management authenticates encryption keys

- ❑ **Message authentication** is concerned with:
  - protecting the integrity of a message
  - validating identity of originator
  - non-repudiation of origin (dispute resolution)

Security Requirements

- Disclosure
- Traffic analysis
- Masquerade
- Content modification
- Sequence modification
- Timing modification
- Source repudiation
- Destination repudiation

Message Authentication

- Virtually all authentication mechanisms rely on an **authenticator** that is sent with the message

- Three alternative functions used:
  - message encryption
  - message authentication codes (MAC)
  - hash functions

Message Encryption

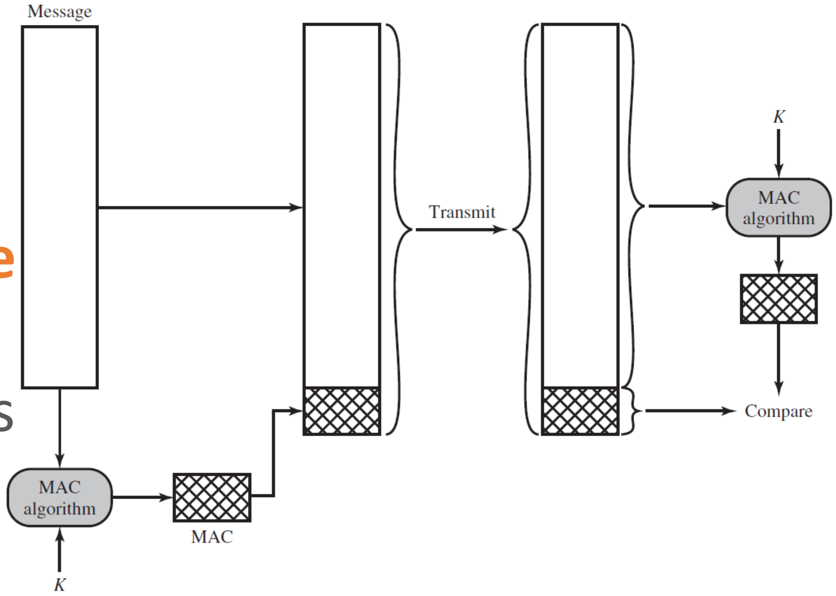
- Message encryption by itself also provides a measure of authentication
- If symmetric encryption is used then:
  - receiver knows sender must have created it
  - since only sender and receiver know key used
  - know content cannot have been altered
  - if message has suitable structure, redundancy or a checksum to detect any changes

Message Encryption

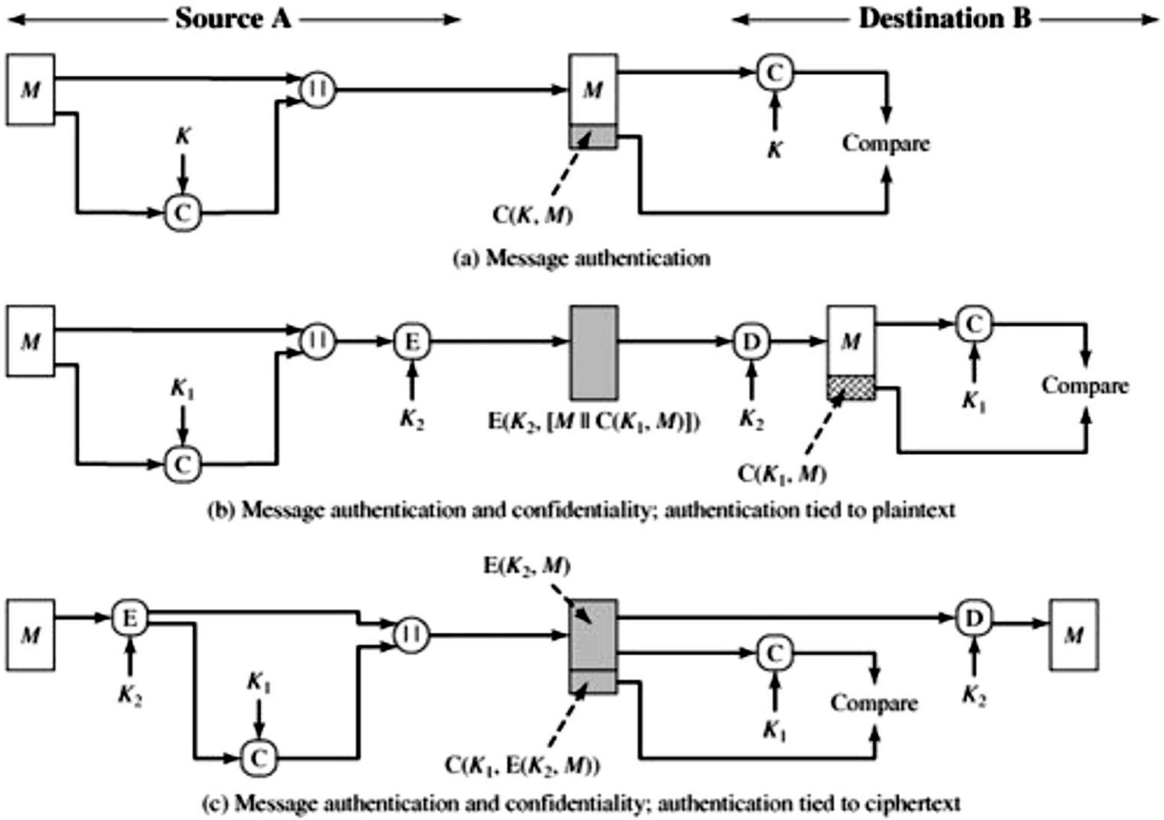
- If public-key encryption is used:
  - encryption provides no confidence of sender
  - since anyone potentially knows public-key
  - however if
    - sender **signs** message using their private-key
    - then encrypts with recipients public key
  - have both secrecy and authentication
  - But at cost of two public-key uses on message

Message Authentication Codes (MAC)

- Generated by an algorithm that creates a small fixed-sized block
  - depending on both message and key
  - like encryption though need not be reversible
- Appended to message as a **signature**
- Receiver performs same computation on message and checks it matches the MAC
- Provides assurance that message is unaltered and comes from sender



Message Authentication Codes (MAC)



Message Authentication Codes (MAC)

- MAC provides authentication
- Can also use encryption for secrecy
  - generally use separate keys for each
  - can compute MAC either before or after encryption
  - generally, MAC done before encryption is regarded as better
- Why use a MAC?
  - sometimes only authentication is needed (e.g. broadcast message)
  - sometimes need authentication to persist longer than the encryption (e.g. archival use, software code)
- Note that a MAC is not a digital signature

MAC Properties

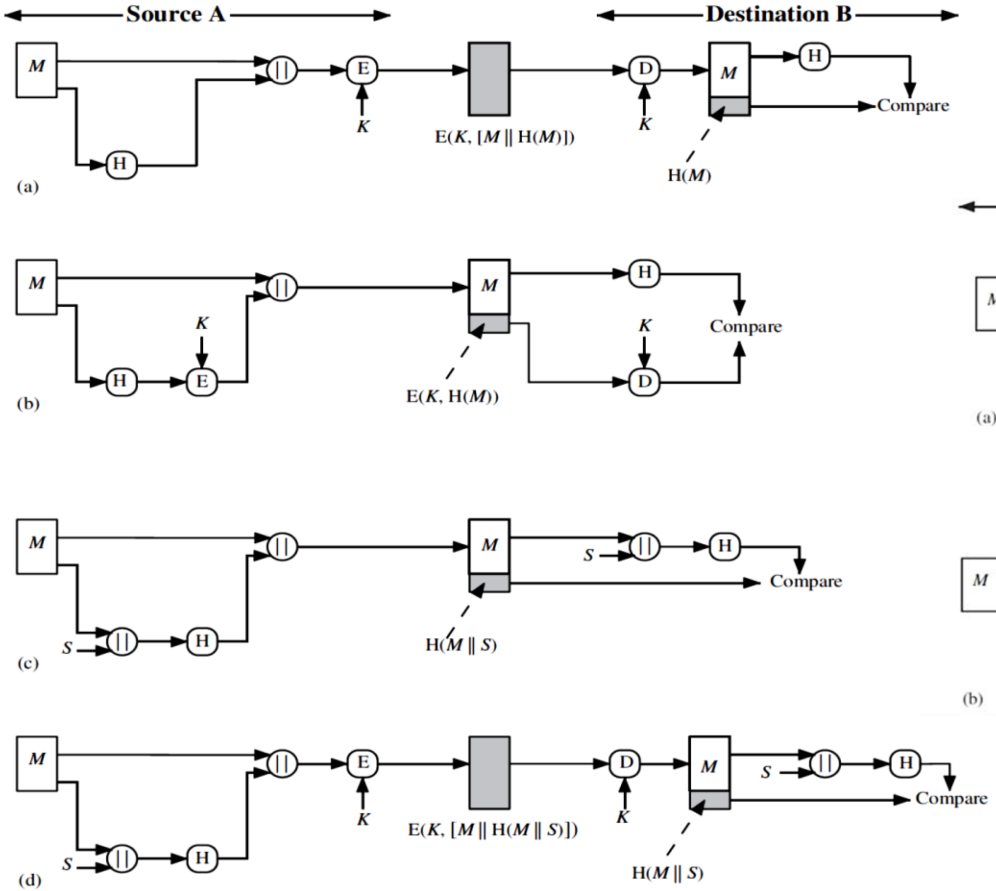
- A MAC is a cryptographic checksum
  - $MAC = C_K(M)$
  - condenses a variable-length message M
  - using a secret key K
  - to a fixed-sized authenticator
- Is a many-to-one function
  - potentially many messages have same MAC
  - but finding these needs to be very difficult

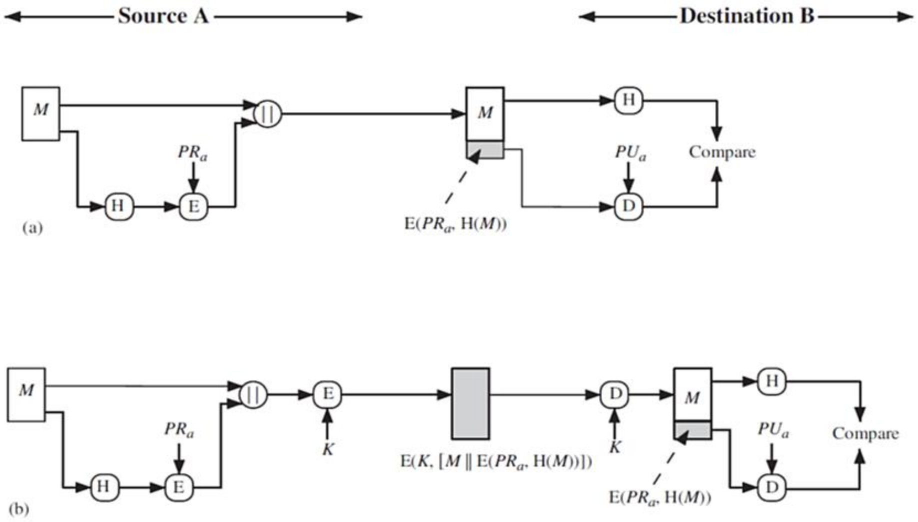
MAC Requirements

- Need the MAC to satisfy the following:
  1. knowing a message and MAC, is infeasible to find another message with same MAC
  2. MACs should be uniformly distributed
  3. MAC should depend equally on all bits of the message

Hash Functions

- Condenses arbitrary message to fixed size `h=H(M)`
- usually assume that the hash function is public and not keyed
  - cf. MAC which is keyed
- Hash used to detect changes to message
- Can use in various ways with message
- Most often to create a digital signature


Source A
Destination B
M
||
E
K
E(K, [M || H(M)])
D
H
Compare
H(M)
(a)
E(K, H(M))
(b)
S
H(M || S)
(c)
E(K, [M || H(M || S)])
(d)


Source A
Destination B
M
||
H
E
PR_a
PU_a
D
Compare
E(PR_a, H(M))
(a)
K
E(K, [M || E(PR_a, H(M))])
(b)

Requirements for Hash Functions

1. Can be applied to any sized message M
2. Produces fixed-length output h
3. Is easy to compute $h=H(M)$ for any message M
4. Given h, it is infeasible to find x s.t. $H(x)=h$
   - **one-way property**
5. Given x, it is infeasible to find y s.t. $H(y)=H(x)$
   - **weak collision resistance**
6. Is infeasible to find any x, y s.t. $H(y)=H(x)$
   - **strong collision resistance**

Simple Hash Functions

- There are several proposals for simple functions
- Based on XOR of message blocks
- Not secure since can manipulate any message and either not change hash or change hash also
- Need a stronger cryptographic function

Birthday Attack

- Might think a 64-bit hash is secure
- But by **Birthday Paradox** is not
- **Birthday attack** works thus:
  - opponent generates $2^{m/2}$ variations of a valid message all with essentially the same meaning
  - opponent also generates $2^{m/2}$ variations of a desired fraudulent message
  - two sets of messages are compared to find pair with same hash (probability > 0.5 by birthday paradox)
  - have user sign the valid message, then substitute the forgery which will have a valid signature
- Conclusion is that we need to use larger MAC/hash

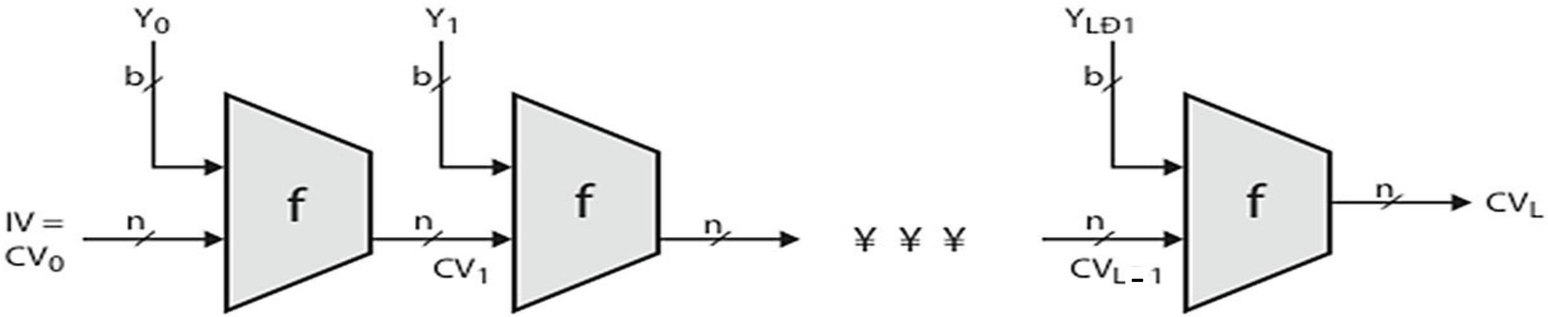
Dear Anthony,

{This letter is / I am writing} to introduce {you to / to you} {Mr. / --} Alfred {P. / --} Barton, the {new / newly appointed} {chief / senior} jewellery buyer for {our / the} Northern {European / Europe} {area / division}. He {will take / has taken} over {the / --} responsibility for {all / the whole of} our interests in {watches and jewellery / jewellery and watches} in the {area / region}. Please {afford / give} him {every / all the} help he {may need / needs} to {seek out / find} the most {modern / up to date} lines for the {top / high} end of the market. He is {empowered / authorized} to receive on our behalf {samples / specimens} of the {latest / newest} {watch and jewellery / jewellery and watch} products, {up / subject} to a {limit / maximum} of ten thousand dollars. He will {carry / hold} a signed copy of this {letter / document} as proof of identity. An order with his signature, which is {appended / attached} {authorizes / allows} you to charge the cost to this company at the {above / head office} address. We {fully / --} expect that our {level / volume} of orders will increase in the {following / next} year and {trust / hope} that the new appointment will {be / prove} {advantageous / an advantage} to both our companies.

Hash Function Security

- Like block ciphers have:
- **Brute-force** attacks exploiting
  - strong collision resistance hash have cost $2^{m/2}$
  - MACs with known message-MAC pairs
    - can either attack keyspace (cf. key search) or MAC
- **Cryptanalytic attacks** exploit structure
  - like block ciphers want brute-force attacks to be the best alternative
  - Exploiting properties of round functions of block ciphers

Hash Algorithm Structure



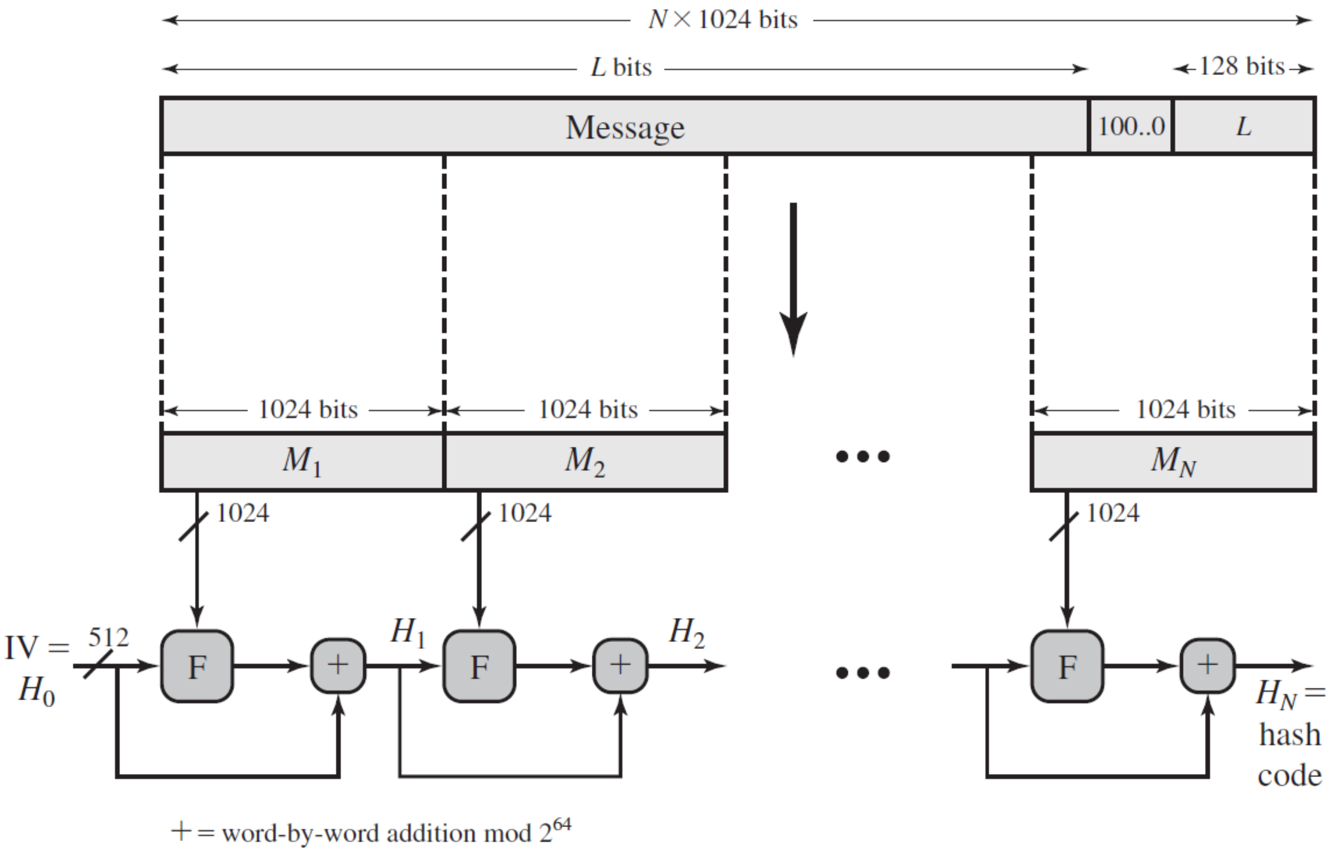
IV = Initial value
CVi = chaining variable
Yi = ith input block
f = compression algorithm

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

Secure Hash Algorithm (SHA)

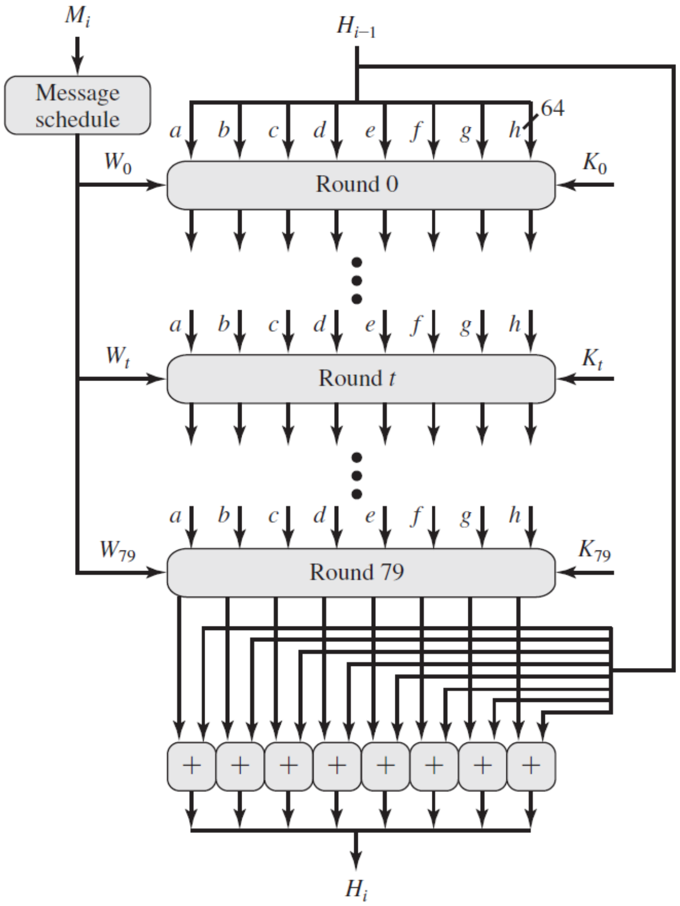
- SHA originally designed by NIST & NSA in 1993
- Was revised in 1995 as SHA-1
- US standard for use with DSA signature scheme
  - standard is FIPS 180-1 1995, also Internet RFC3174
- Based on design of the MD4 hash algorithm with key differences
- Produces 160-bit hash values
- Recent 2005 results on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

Secure Hash Algorithm (SHA)

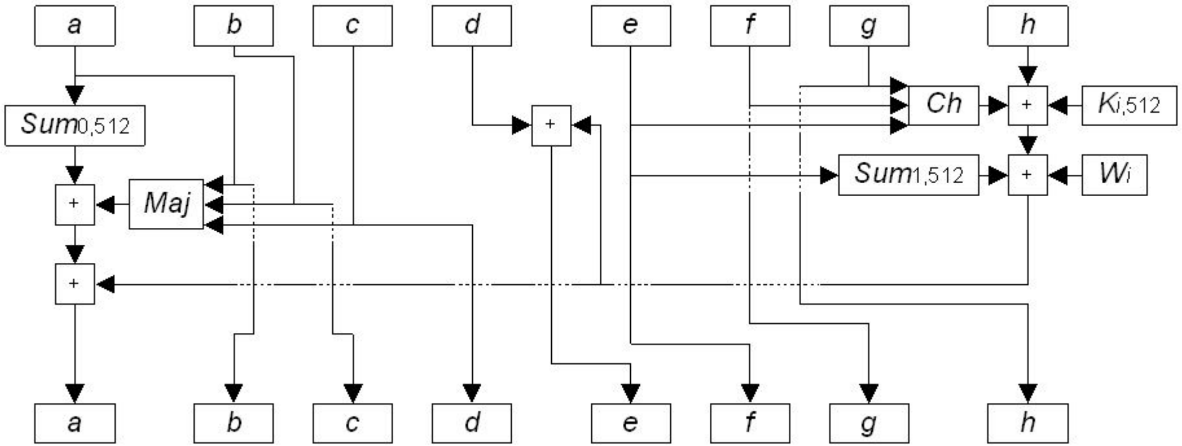


SHA-512 Compression Function

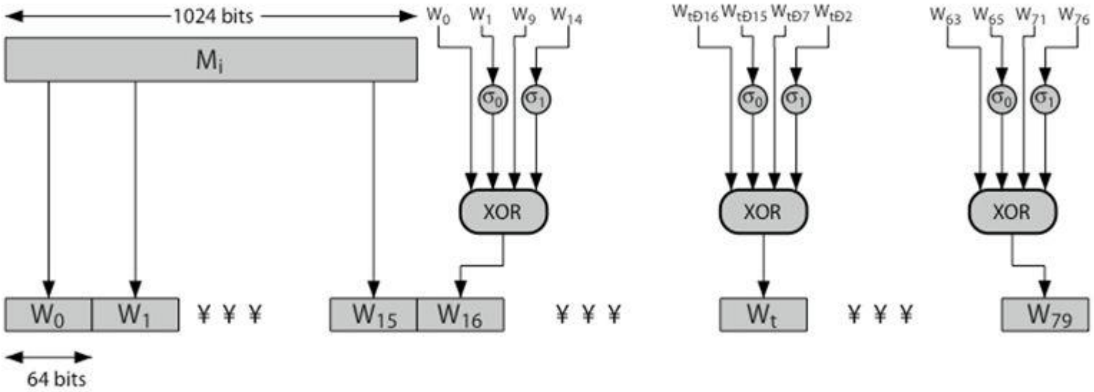
- Heart of the algorithm
- Processing message in 1024-bit blocks
- Consists of 80 rounds
  - updating a 512-bit buffer
  - using a 64-bit value W_t derived from the current message block
  - and a round constant K_i based on cube root of first 80 prime numbers



SHA-512 Round Function



SHA-512 Message Schedule



$$T_1 = h + \mathrm{Ch}(e, f, g) + \left(\sum\nolimits_1^{512} e\right) + W_t + K_t$$
$$T_2 = \left(\sum\nolimits_0^{512} a\right) + \mathrm{Maj}(a, b, c)$$
$$h = g$$
$$g = f$$
$$f = e$$
$$e = d + T_1$$
$$d = c$$
$$c = b$$
$$b = a$$
$$a = T_1 + T_2$$

where

t = step number; $0 \leq t \leq 79$

$\mathrm{Ch}(e, f, g)$ = (e AND f) $\oplus$ (NOT e AND g)
the conditional function: If e then f else g

$\mathrm{Maj}(a, b, c)$ = (a AND b) $\oplus$ (a AND c) $\oplus$ (b AND c)
the function is true only of the majority (two or three) of the arguments are true

$\left(\sum_0^{512} a\right) = \mathrm{ROTR}^{28}(a) \oplus \mathrm{ROTR}^{34}(a) \oplus \mathrm{ROTR}^{39}(a)$

$\left(\sum_1^{512} e\right) = \mathrm{ROTR}^{14}(e) \oplus \mathrm{ROTR}^{18}(e) \oplus \mathrm{ROTR}^{41}(e)$

$\mathrm{ROTR}^n(x)$ = circular right shift (rotation) of the 64-bit argument x by n bits

W_t = a 64-bit word derived from the current 512-bit input block

K_t = a 64-bit additive constant

$+$ = addition modulo 2^{64}

$$W_t = \sigma_1^{512}(W_{t-2}) + W_{t-7} + \sigma_0^{512}(W_{t-15}) + W_{t-16}$$

where

$\sigma_0^{512}(x) = \mathrm{ROTR}^1(x) \oplus \mathrm{ROTR}^8(x) \oplus \mathrm{SHR}^7(x)$

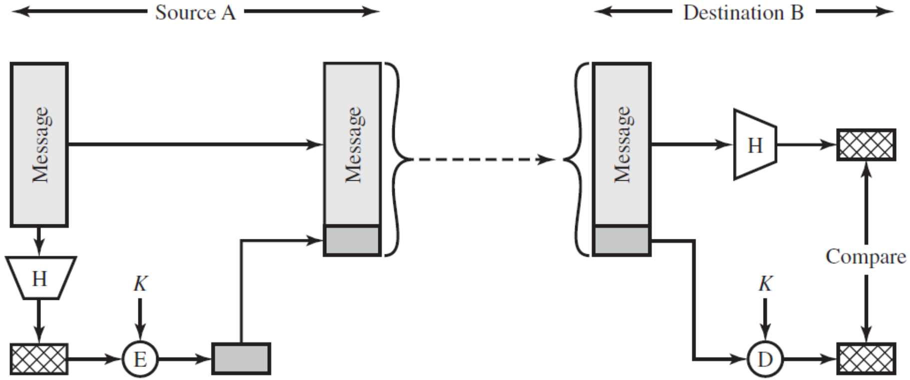
$\sigma_1^{512}(x) = \mathrm{ROTR}^{19}(x) \oplus \mathrm{ROTR}^{61}(x) \oplus \mathrm{SHR}^6(x)$

$\mathrm{ROTR}^n(x)$ = circular right shift (rotation) of the 64-bit argument x by n bits

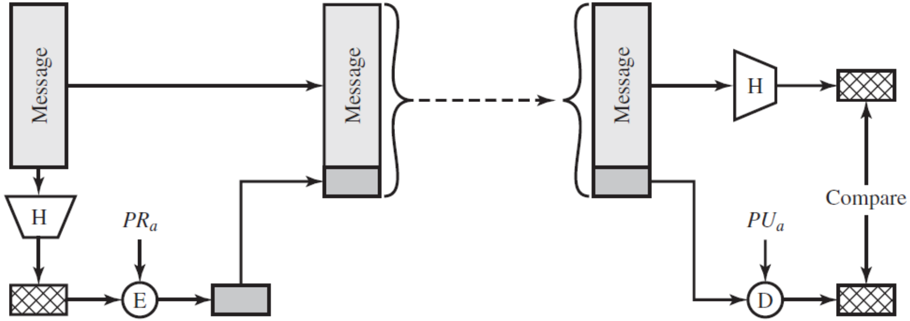
$\mathrm{SHR}^n(x)$ = left shift of the 64-bit argument x by n bits with padding by zeros on the right

$+$ = addition modulo 2^{64}

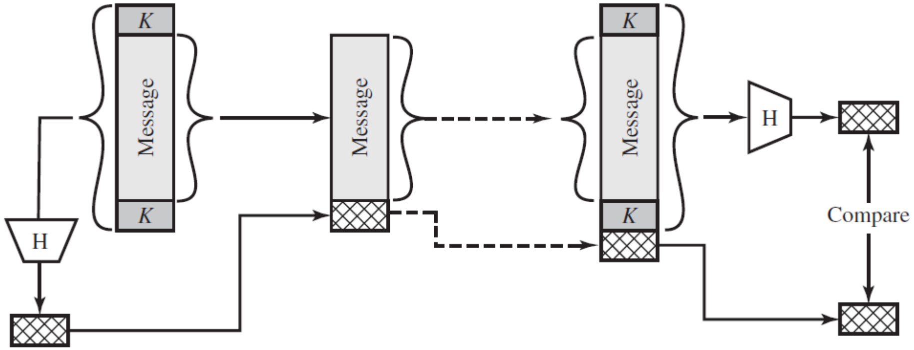
MACs Using Hash Functions



(a) Using symmetric encryption



(b) Using public-key encryption



(c) Using secret value

Problem

A security system requires three services: confidentiality, integrity, and non-repudiation. For each of the following diagrams, specify service(s) achieved and the type of integrity function used (hash, MAC, or neither).