Lecture 6

HASH FUNCTIONS

UTAS
Sultanate of Oman
February 2023

CSSY2201 : Introduction to Cryptography

جامعة التقنية والعلوم التطبيقية
University of Technology and Applied Sciences
Salalah صلالة

Plan

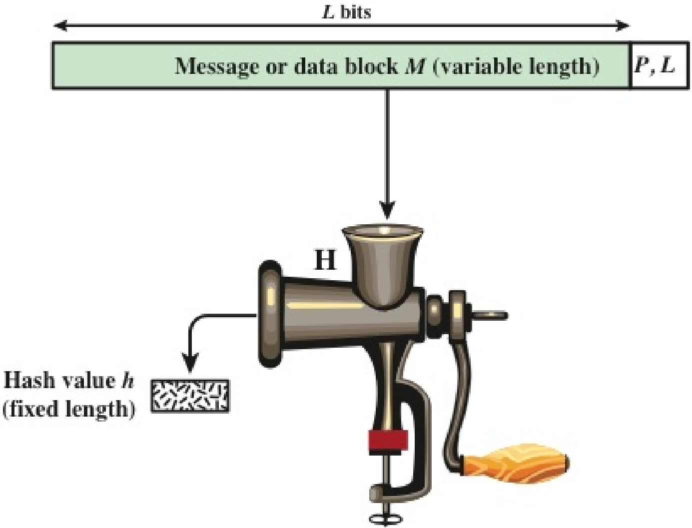
جامعة التقنية
والعلوم التطبيقية
University of Technology
and Applied Sciences
Salalah صلالة

Hash functions

- Hash function accepts variable-length input and outputs a digest or hash of fixed length
- $h = H(M)$
- Its main purpose is integrity checking
- A cryptographic hash function is an algorithm that is mathematically difficult to :
  - Find an entry that gives a well-specified digest (Property : one-way function)
  - find two entries that give the same digest (Property : Collision-free)

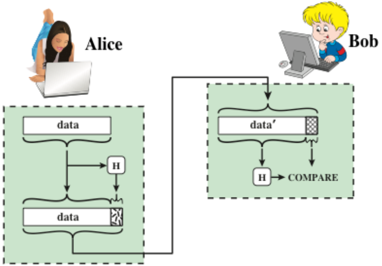
جامعة التقنية والعلوم التطبيقية
University of Technology and Applied Sciences
Salalah صلالة

Cryptographic Hash function h=H(M)

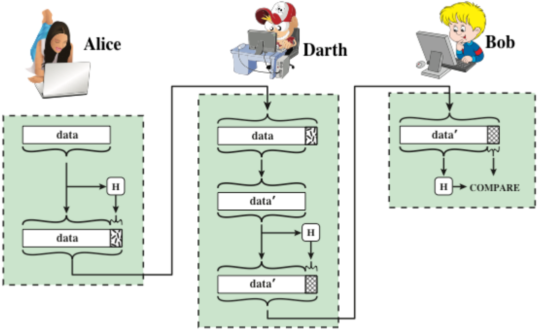


the output of a hash function from a message is named digest

- The digest of the message is its fingerprint
- Much smaller than original post
- easy to calculate
- cannot find message from digest
- Changing the message automatically changes the digest



(a) Use of hash function to check data integrity

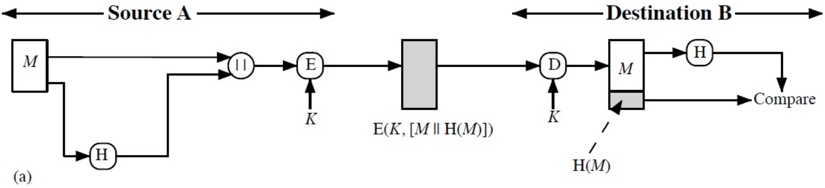


(b) Man-in-the-middle attack

جامعة التقنية
والعلوم التطبيقية
University of Technology
and Applied Sciences
Salalah صلالة

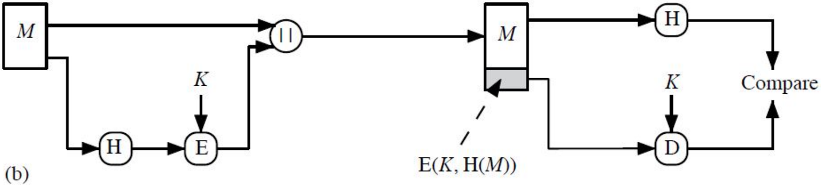
Message Authentication

- Verify message integrity
  - Ensure received data is exactly as sent
  - Ensure sender identity is valid
- Example 1 : Encrypt the message and its digest with a symmetric cryptosystem



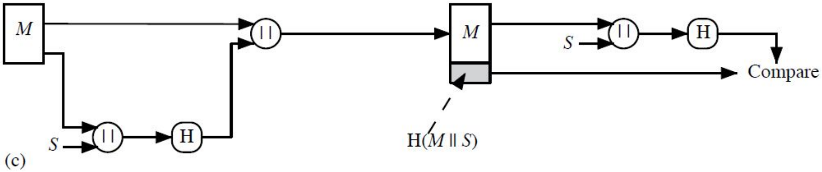
Message Authentication

- Example 2 : Encrypt only the message digest
- it reduces the complexity of calculation if confidentiality is not requested



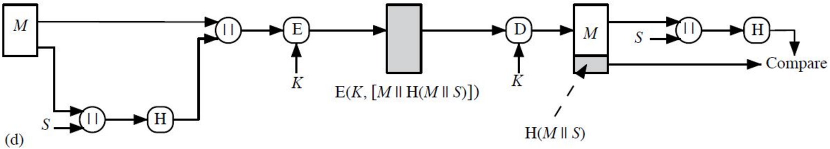
Message Authentication

- Example 3 : A shared secret is hashed
- No need for encryption



Message Authentication

- Example 4 : A shared secret combined with confidentiality



Other Uses of Hash Functions

- Used to create password files
  - When a user types a password, the hash of the password is compared to the saved hash for verification
  - This approach is used by the majority of operating systems
- used to detect intrusions and viruses
  - save the H(f) of each file to disk
  - the antivirus can later check if the file has been altered or not by recalculating its digest H(f)
  - An intruder will try to change F without changing H(f) : very difficult !
- can be used to build PRNG pseudo-random sequence generators
  - generate keystreams, secret keys

Requirements of a hash function

- Variable length entry
- Fixed length output
- Efficiency : given x, it is easy to generate the digest H(x) in s/w or h/w
- One-way function (Pre-image resistant) : For a given digest h, it is impossible to find y such that $H(y) = h$
- No broad sense collision (Second pre-image resistant : weak collision resistant) : For any given x, it is impossible to find $y \neq x$ such that $H(y) = H(x)$
- No collision in the strict sense (collision resistant : Strong collision resistant) : it is impossible to find a pair (x, y) such that $H(x) = H(y)$
- Random criterion : The output of H must be random according to the standard tests (NIST : 16 tests of the random criterion)

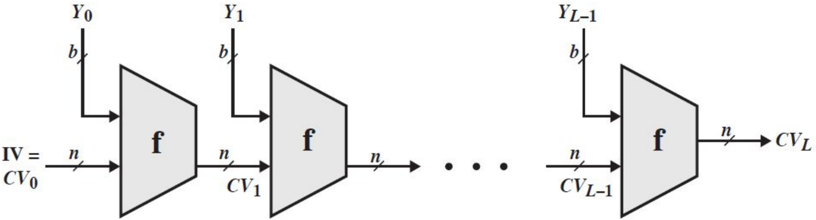
NB : "impossible" = "mathematically or by calculation difficult"

Requirements of a hash function

	Preimage Resistant	Second Preimage Resistant	Collision Resistant
Hash + digital signature	yes	yes	yes*
Intrusion detection and virus detection		yes	
Hash + symmetric encryption			
One-way password file	yes		
MAC	yes	yes	yes*

* Resistance required if attacker is able to mount a chosen message attack

General structure of a hash function



IV	=	Initial value
CV_i	=	chaining variable
Y_i	=	ith input block
f	=	compression algorithm
L	=	number of input blocks
n	=	length of hash code
b	=	length of input block

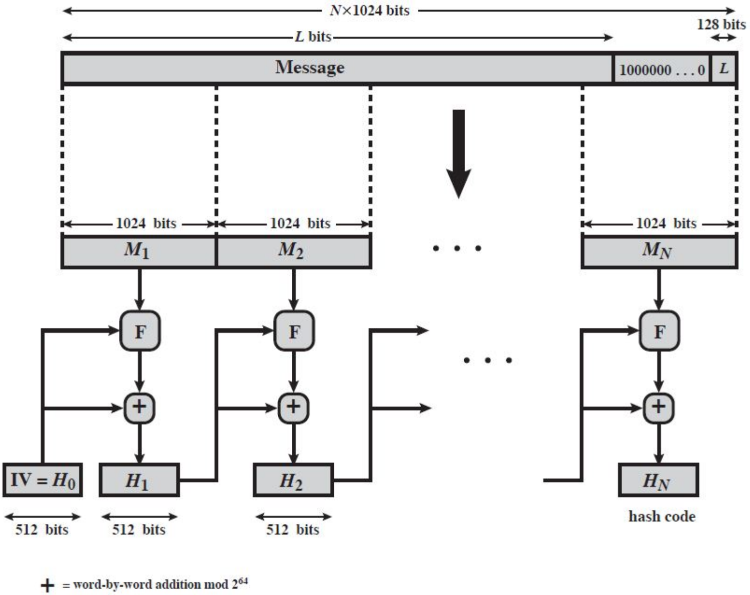
Secure Hash Algorithm (SHA)

- SHA was designed by "National Institute of Standards and Technology (NIST)" and published as "federal information processing standard" (FIPS 180) in 1993
- was revised in 1995 as SHA-1
- Based on MD4 hash function
- Produces a 160-bit size digest
- In 2002 NIST produced a revised version of the standard to define 3 more SHAs with lengths 256, 384, and 512 Known as SHA-2

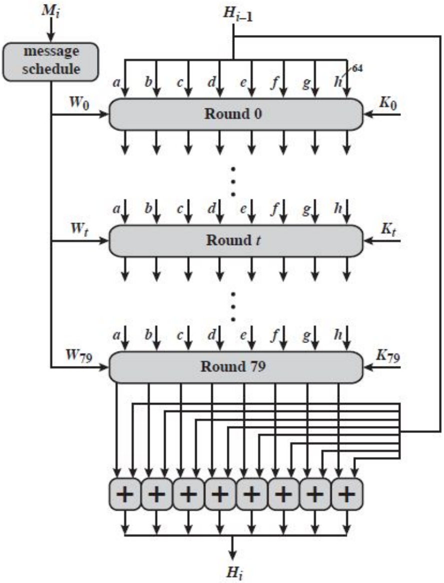
Comparaison of SHA versions

	SHA-1	SHA-224	SHA-256	SHA-384	SHA-512
Message Digest Size	160	224	256	384	512
Message Size	$< 2^{64}$	$< 2^{64}$	$< 2^{64}$	$< 2^{128}$	$< 2^{128}$
Block Size	512	512	512	1024	1024
Word Size	32	32	32	64	64
Number of Steps	80	64	64	80	80

SHA-512

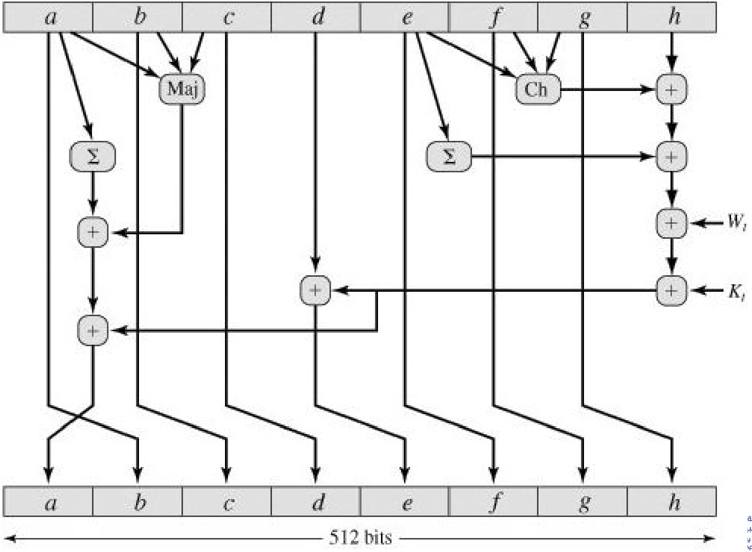

N×1024 bits
L bits
128 bits
Message
1000000 . . . 0
L
1024 bits
1024 bits
1024 bits
M1
M2
MN
F
F
F
+
+
+
IV = H0
H1
H2
HN
512 bits
512 bits
512 bits
hash code
+ = word-by-word addition mod 2^64

SHA-512 : Processing of a 1024-Bit block

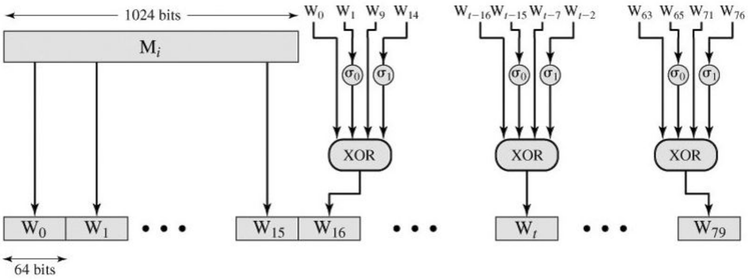


جامعة التقنية والعلوم التطبيقية
University of Technology and Applied Sciences
Salalah صلالة

SHA-512 : buffers update



SHA-512 : Processing of the message M_i

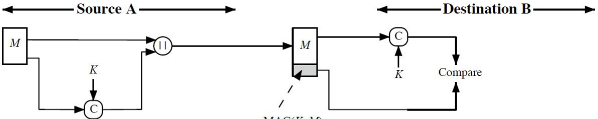


Message authentication techniques

1. Hash functions : a function that accepts n variable-length messages as input and outputs a fixed-length digest. The digest is the authenticator of the message (already seen)
2. The encryption of the message : the ciphertext of the message constitutes its authenticator : Authenticated encryption
3. The MAC (Message authentication code) : a function of the message and a secret key that produce a fixed length output MAC which constitutes the authenticator of the message
4. HMAC
5. CMAC

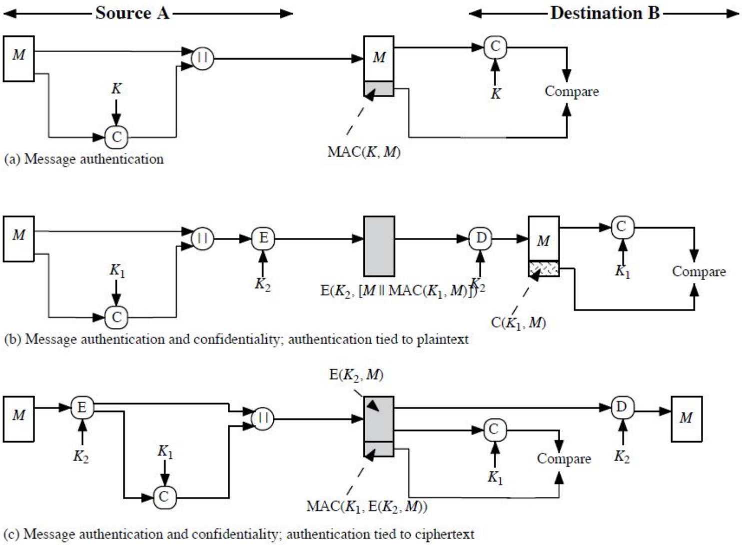
MAC

- Also known as keyed hash function
- used when two entities sharing the same key to authenticate the information exchanged between them
- Takes as input a secret key K and a block of data M and produces a MAC=C(K,M)
- the MAC is associated with the message when it is sent
- If the integrity of the message needs to be checked, the MAC function is applied to the message and the result is compared to the associated MAC (received)
- a hacker who wants to modify the message will be unable to modify the MAC without knowing the secret key.
- MAC is not a digital signature

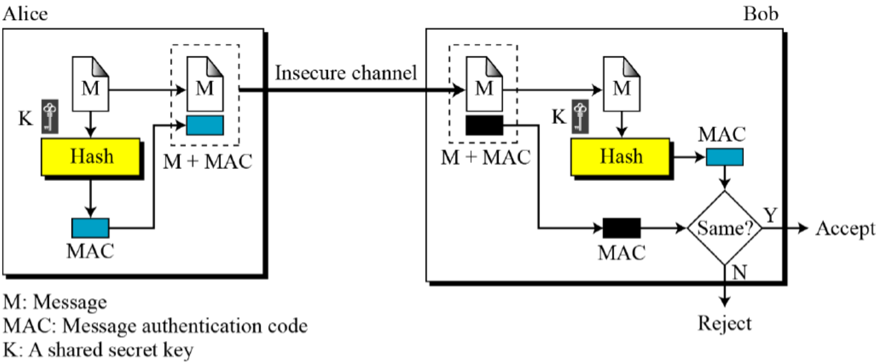


(a) Message authentication

Basic use of MAC



Keyed hash= MAC



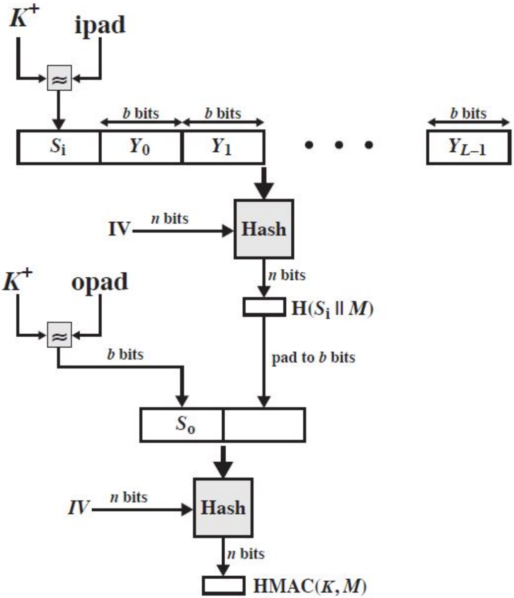
HMAC

- specified as Internet standard RFC2104
- uses hash function on the message :

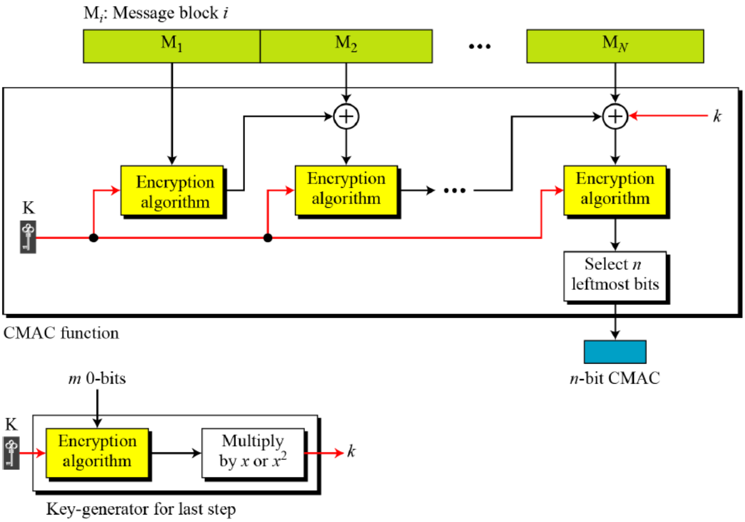
$$HMACK(M) = Hash[(K^+ \oplus opad)||Hash[(K^+ \oplus ipad)||M)]]$$

  - where K^+ is the key padded out to block size
  - opad, ipad are specified padding constants
- any hash function can be used eg. MD5, SHA-1, SHA-2, RIPEMD-160, Whirlpool

HMAC


K^+
ipad
b bits
b bits
b bits
S_i
Y_0
Y_1
Y_{L-1}
IV
n bits
Hash
n bits
K^+
opad
$H(S_i \| M)$
b bits
pad to b bits
S_o
IV
n bits
Hash
n bits
HMAC(K, M)

CMAC



M_i: Message block i
M_1
M_2
M_N
K
Encryption algorithm
Encryption algorithm
Encryption algorithm
k
Select n leftmost bits
CMAC function
n-bit CMAC
m 0-bits
K
Encryption algorithm
Multiply by x or x^2
k
Key-generator for last step

Authenticated encryption

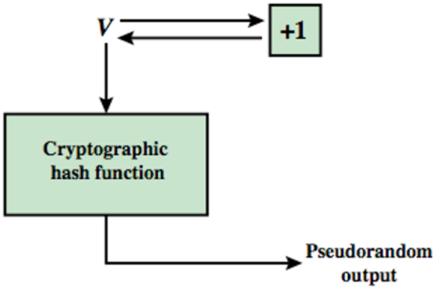
- Protect privacy and provide authentication at the same time
- Different approaches :
  - Hash-then-encrypt : $E(K, (M||H(M))$
  - MAC-then-encrypt : $E(K2, (M||MAC(K1, M))$
  - Encrypt-then-MAC : $C = E(K2, M), T = MAC(K1, C)$
  - Encrypt-and-MAC : $C = E(K2, M), T = MAC(K1, M)$
- decryption and verification is easy

PRNG

- essential elements of PRNG are
  - seed value
  - deterministic algorithm
- seed must be known only as needed
- can base PRNG on
  - encryption algorithm,
  - hash function or
  - MAC (NIST SP 800-90)

PRNG from Hash function

- hash PRNG from SP800-90 and ISO18031
  - take seed V
  - repeatedly add 1
  - hash V
  - use n-bits of hash as random value
- secure if good hash used



PRNG using a MAC

- MAC PRNGs in SP800-90, IEEE 802.11i, TLS
  - use key
  - input based on last hash in various ways