ĐẠI NAM
UNIVERSITY

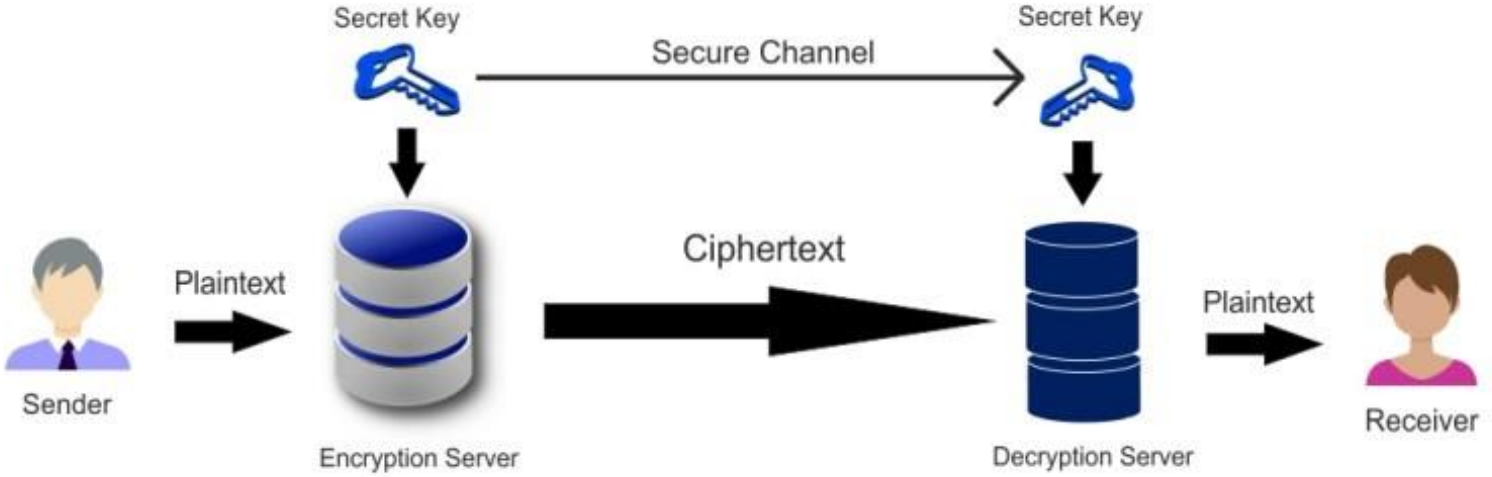
Bài 6

HÀM BĂM (HASH FUNCTION)

Giảng viên: TS. Trần Quý Nam
(namtq@dainam.edu.vn)

ÔN TẬP

- **Nhắc lại kiến thức bài trước**



AES Algorithm Working

NỘI DUNG

- Giới thiệu về hàm băm
- Các thuật toán MD5, SHA-512
- Ứng dụng của hàm băm

GIỚI THIỆU

- Hàm băm là gì?
- Nó hoạt động như thế nào?
- Hàm băm có đặc điểm gì?
- Ứng dụng của hàm băm?

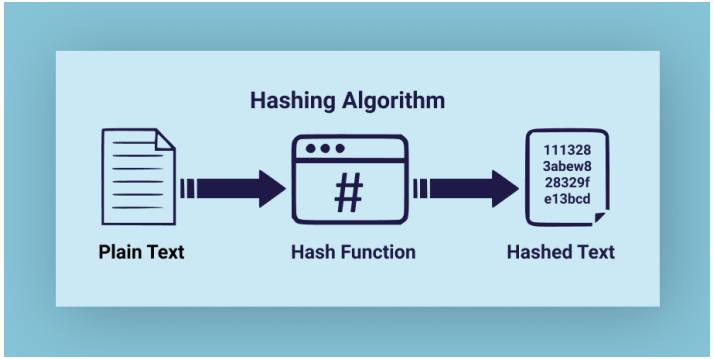
HÀM BĂM (HASH FUNCTION)

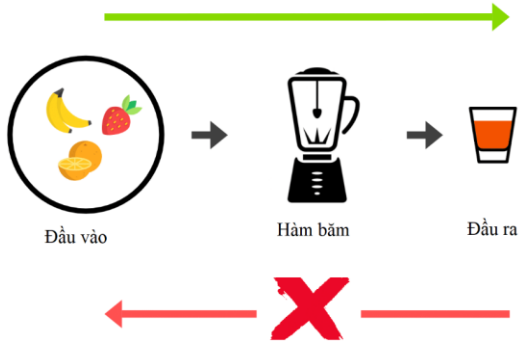
- **Hàm băm** là các thuật toán không sử dụng khóa để mã hóa, thực hiện băm thông điệp được đưa vào theo một thuật toán ***h*** một chiều nào đó, rồi đưa ra một bản băm (là văn bản đại diện) có kích thước cố định.
- Người nhận không biết được nội dung hay độ dài ban đầu của thông điệp đã được băm bằng hàm băm.
- Giá trị của hàm băm là duy nhất, và **không thể suy ngược** lại được nội dung thông điệp từ giá trị băm này.

HÀM BĂM (HASH FUNCTION)

- Thuật toán **mã hóa một chiều** (*Keyless Algorithm*)
- Chuyển đổi dữ liệu đầu vào có độ dài bất kỳ thành một giá trị đầu ra có độ dài cố định.
- Sử dụng những thuật toán, công thức toán học để biến thành đầu ra tiêu chuẩn có độ dài nhất định.

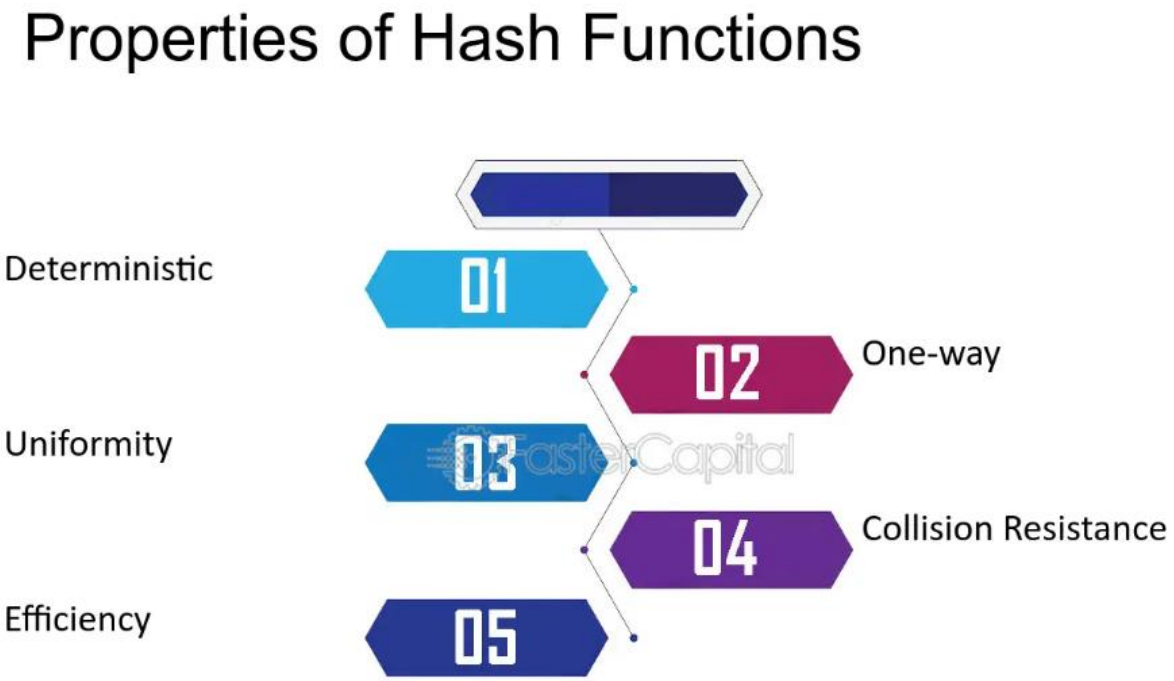
Ví dụ: MD5, SHA-1, SHA-256





HÀM BĂM (HASH FUNCTION)

Đặc điểm quan trọng của Hàm Băm

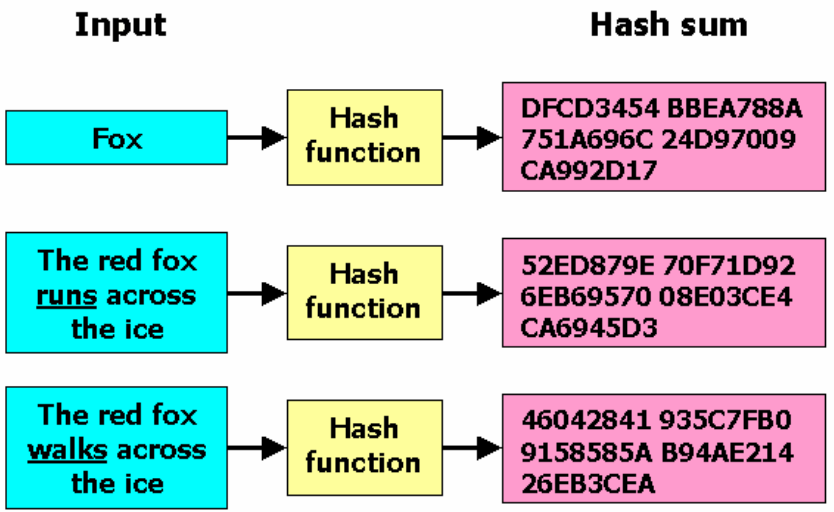


HÀM BĂM (HASH FUNCTION)

- Với thông điệp đầu vào *x* thu được bản băm ***z = h(x)*** là duy nhất.
- Nếu dữ liệu trong thông điệp *x* thay đổi để thành thông điệp *x'* thì ***h(x') ≠ h(x)*** => Hai thông điệp hoàn toàn khác nhau thì giá trị hàm băm cũng khác nhau.
- Nội dung của thông điệp gốc không thể bị suy ra từ giá trị hàm băm => Với thông điệp *x* thì dễ dàng tính được *z = h(x)*, nhưng lại không thể (thực chất là khó) suy ngược lại được *x* nếu chỉ biết giá trị hàm băm *h*

ĐẶC ĐIỂM HÀM BĂM

- **Xác định:** Cùng một đầu vào luôn tạo ra cùng một giá trị băm.
- **Đơn hướng:** Không thể tính ngược
- **Tính toàn vẹn:** Thay đổi nhỏ trong dữ liệu vào tạo giá trị băm khác.
- **Hiệu quả:** xử lý nhanh ngay cả với lượng dữ liệu lớn.
- **Không trùng lặp:** Hai đầu vào khác nhau giá trị băm khác nhau.



https://www.tools4noobs.com/online_tools/hash/

HÀM BĂM (HASH FUNCTION)

Hạn chế hàm băm:

- **Va chạm (Collisions):** Hiếm gặp, việc hai dữ liệu khác nhau tạo ra cùng một giá trị băm vẫn có thể xảy ra "collision".
- **Không phục hồi được dữ liệu gốc(One way- Đơn hướng).**
- **Vẫn có khả năng bị mất an toàn.**

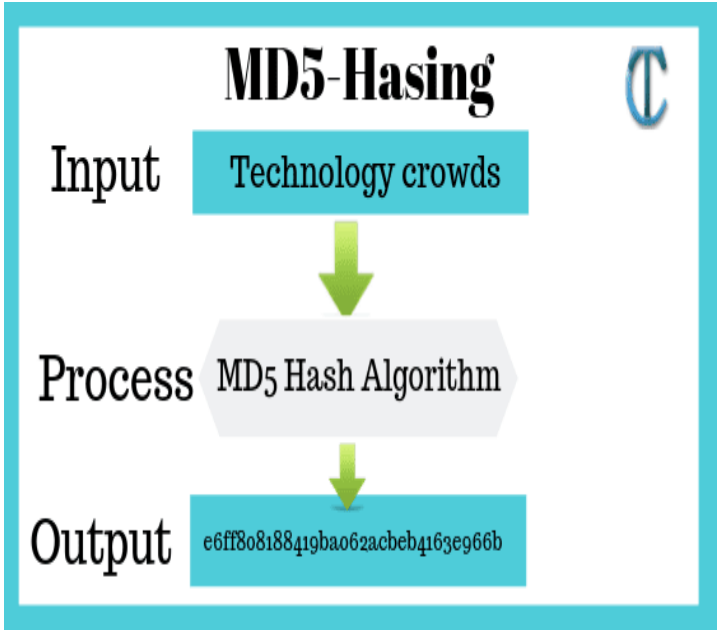
THUẬT TOÁN BĂM PHỔ BIẾN

- MD2, MD4, MD5, **MD5** (Message Digest Algorithm)
- **SHA** cho các bản băm đầu ra có kích thước cố định: 128 bit với dòng MD, 160 bit với SHA1

Algorithm	Message Size (bits)	Block Size (bits)	Word Size (bits)	Message Digest Size (bits)
SHA-1	$< 2^{64}$	512	32	160
SHA-224	$< 2^{64}$	512	32	224
SHA-256	$< 2^{64}$	512	32	256
SHA-384	$< 2^{128}$	1024	64	384
SHA-512	$< 2^{128}$	1024	64	512
SHA-512/224	$< 2^{128}$	1024	64	224
SHA-512/256	$< 2^{128}$	1024	64	256

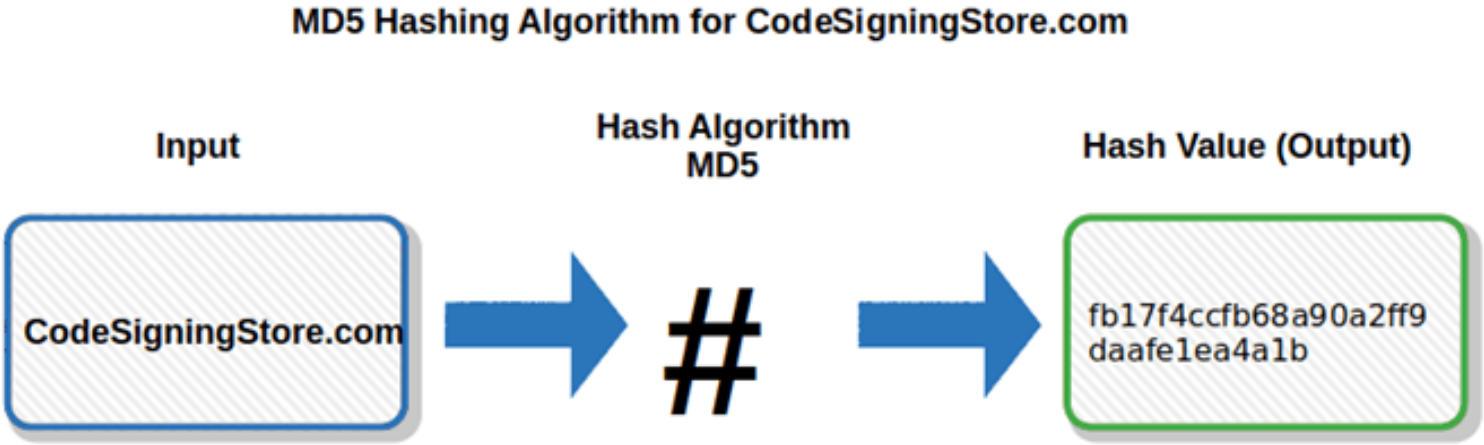
Hàm băm MD5

- MD5 (Message Digest 5) được phát minh bởi Ron Rivest, người cũng xây dựng RSA.
- MD5 được phát triển lên từ MD4 và trước đó là MD2, do MD2 và MD4 không còn được xem là an toàn.
- Kích thước giá trị băm của MD5 là 128 bít, coi như là an toàn (theo nghĩa không tìm được 2 thông điệp có cùng giá trị băm).
- Năm 1994: một phương pháp tấn công MD5 đã được tìm thấy. Tuy vậy ngày nay MD5 vẫn còn được sử dụng phổ biến.



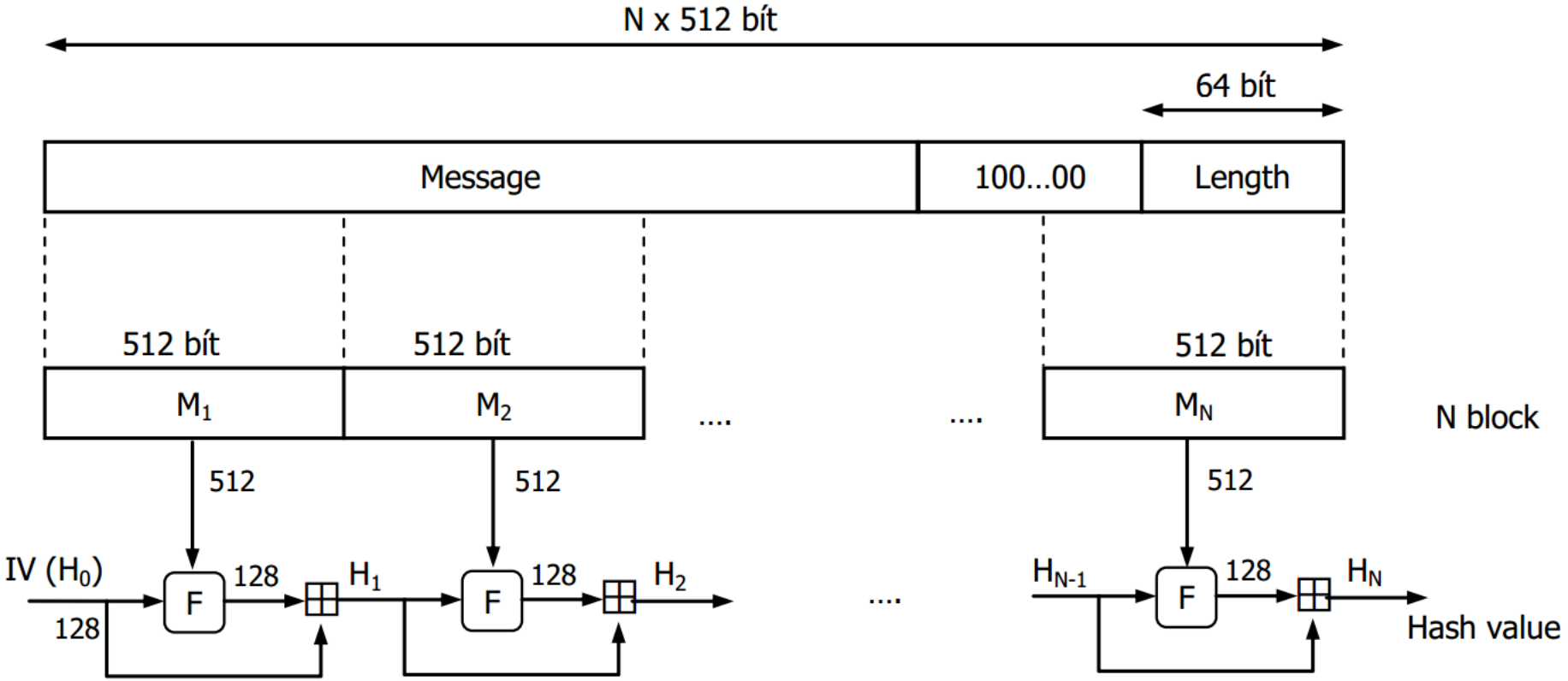
THUẬT TOÁN BĂM MD5

- **MD5 (Message Digest Algorithm 5):** phổ biến nhất, tạo ra giá trị băm dài 128-bit.



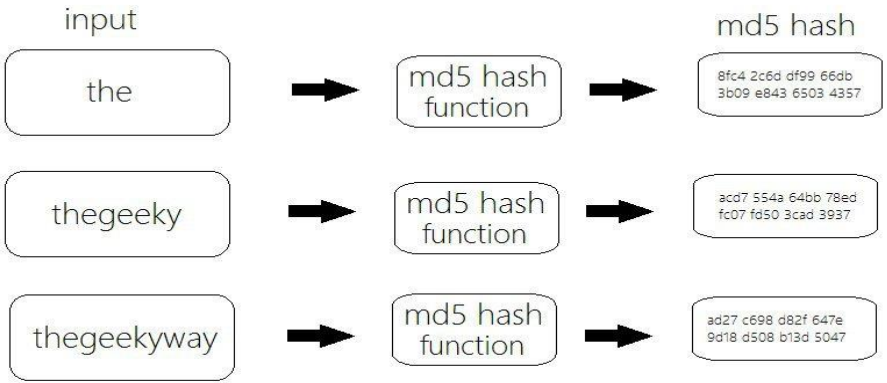
https://www.tools4noobs.com/online_tools/hash/

THUẬT TOÁN BĂM MD5



THUẬT TOÁN BĂM MD5

- Trước tiên thông điệp được thêm dãy bit padding 100….00. Sau đó thông điệp được biểu diễn bằng 64 bít.
- Chiều dài của dãy bít padding được chọn sao cho cuối cùng thông điệp có thể chia thành N block 512 bít M1, M2, … , Mn.



THUẬT TOÁN BĂM MD5

Quá trình tính giá trị băm của thông điệp là quá trình lũy tiến. Trước tiên block M_1 kết hợp với giá trị khởi tạo H_0 thông qua hàm F để tính giá trị hash H_1. Sau đó block M_2 được kết hợp với H_1 để cho ra giá trị hash là H_2. Block M_3 kết hợp với H_2 cho ra giá trị H_3. Cứ như vậy cho đến block M_N thì ta có giá trị băm của toàn bộ thông điệp là H_N.

H_0 là một dãy 128 bít được chia thành 4 từ 32 bít, ký hiệu 4 từ 32 bít trên là abcd. a, b, c, d là các hằng số như sau (viết dưới dạng thập lục phân):

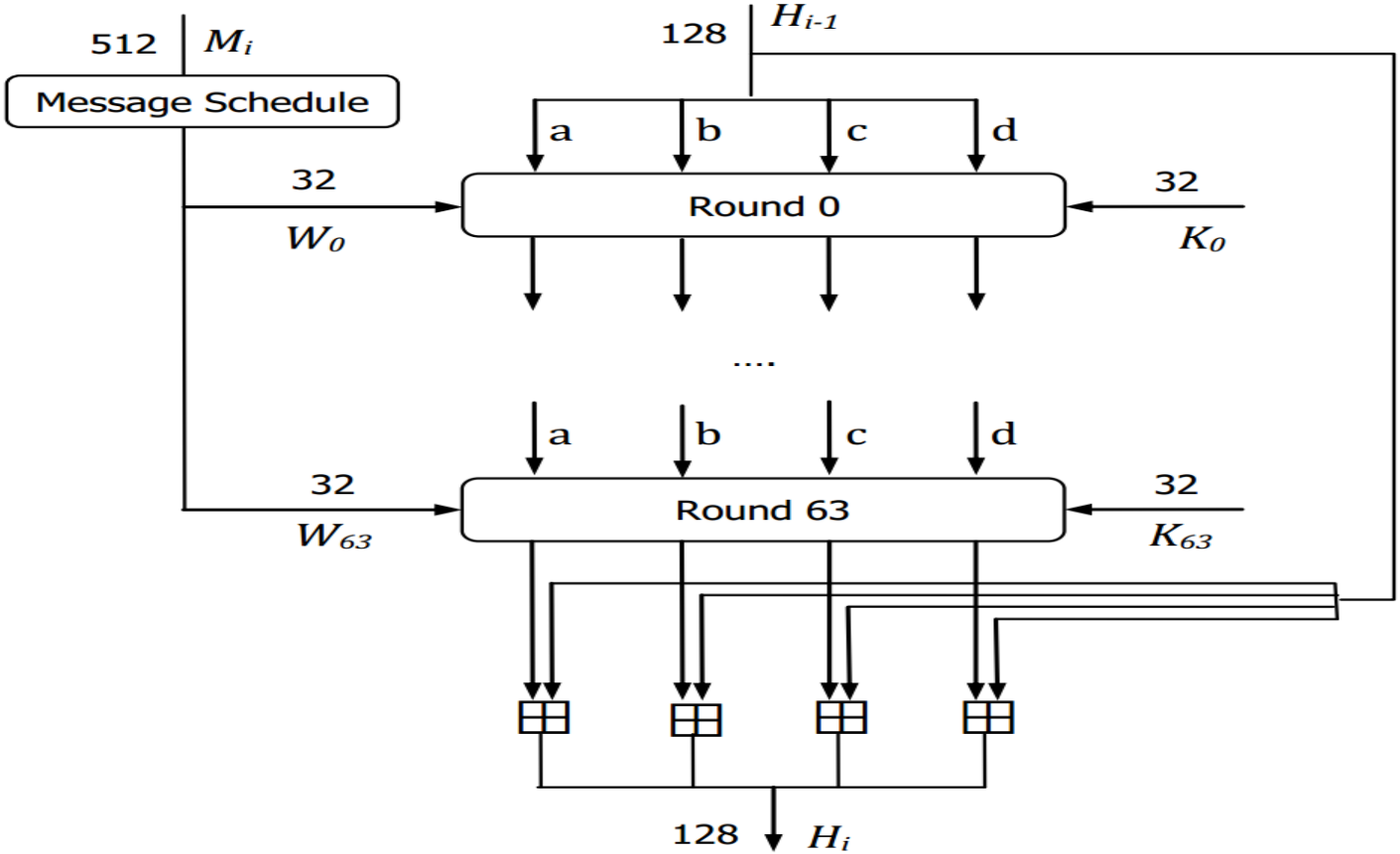
a = 01234567

b = 89abcdef

c = fedbca98

d = 76543210

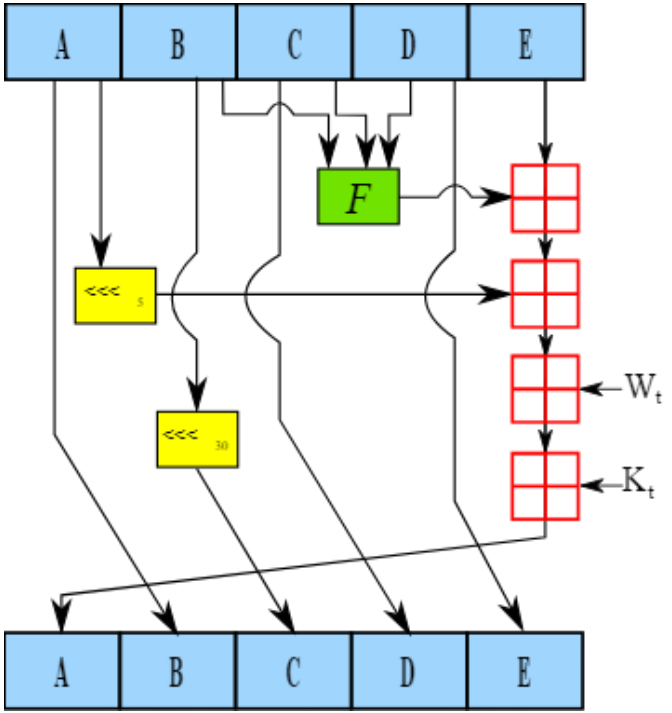
THUẬT TOÁN BĂM MD5 (HÀM F)



HÀM BĂM SHA

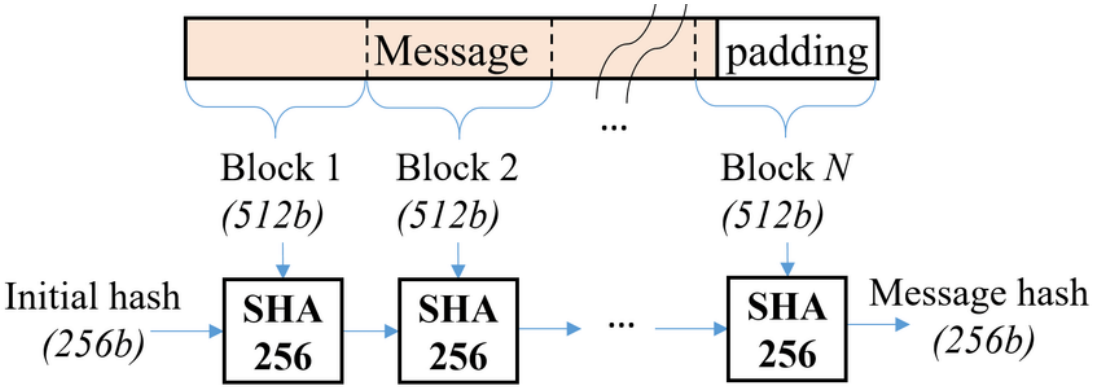
Các thuật toán phổ biến:

- **SHA-1 (Secure Hash Algorithm)**: SHA-1 tạo ra bản tóm tắt có kích thước 160 bit (20 byte).
- Giá trị Hash này được gọi là Message Digest, bao gồm một chuỗi các số thập lục phân dài 40 chữ số.
- Không còn an toàn: Đã bị phá vỡ bởi tấn công collision (năm 2017).



THUẬT TOÁN SHA

- **SHA-2 (SHA-224, SHA-256, SHA-384 và SHA-512):** Là một tập hợp các hàm băm mật mạnh mẽ hơn SHA-1(NSA), được xuất năm 2001.
- Bảo mật cao hơn SHA-1: Chống lại collision attack và brute-force tốt hơn.
- Được sử dụng rộng rãi trong bảo mật và tiền mã hóa.



THUẬT TOÁN SHA

So sánh tham số giữa các thuật toán

Bảng So sánh các tham số SHA (Đơn vị tính là bits)					
	SHA-1	SHA-224	SHA-256	SHA-384	SHA-512
Mã băm	160	224	256	384	512
Input M	$< 2^{64}$	$< 2^{64}$	$< 2^{64}$	$< 2^{128}$	$< 2^{128}$
block size	512	512	512	1024	1024
word size	32	32	32	64	64
Số lần lặp	80	64	64	80	80

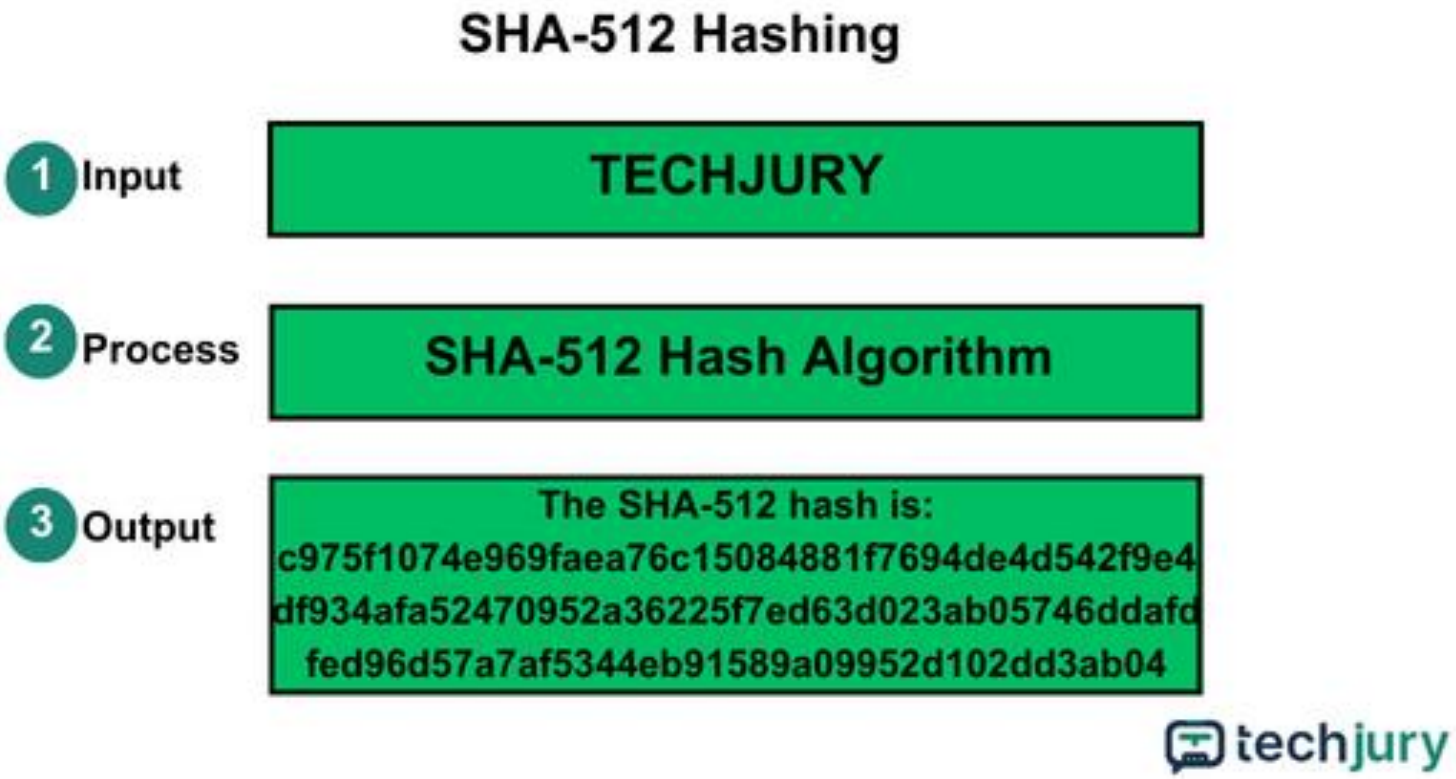
THUẬT TOÁN SHA 512

THỬ NGHIỆM ONLINE:

https://hash.online-convert.com/vi/sha512-generator

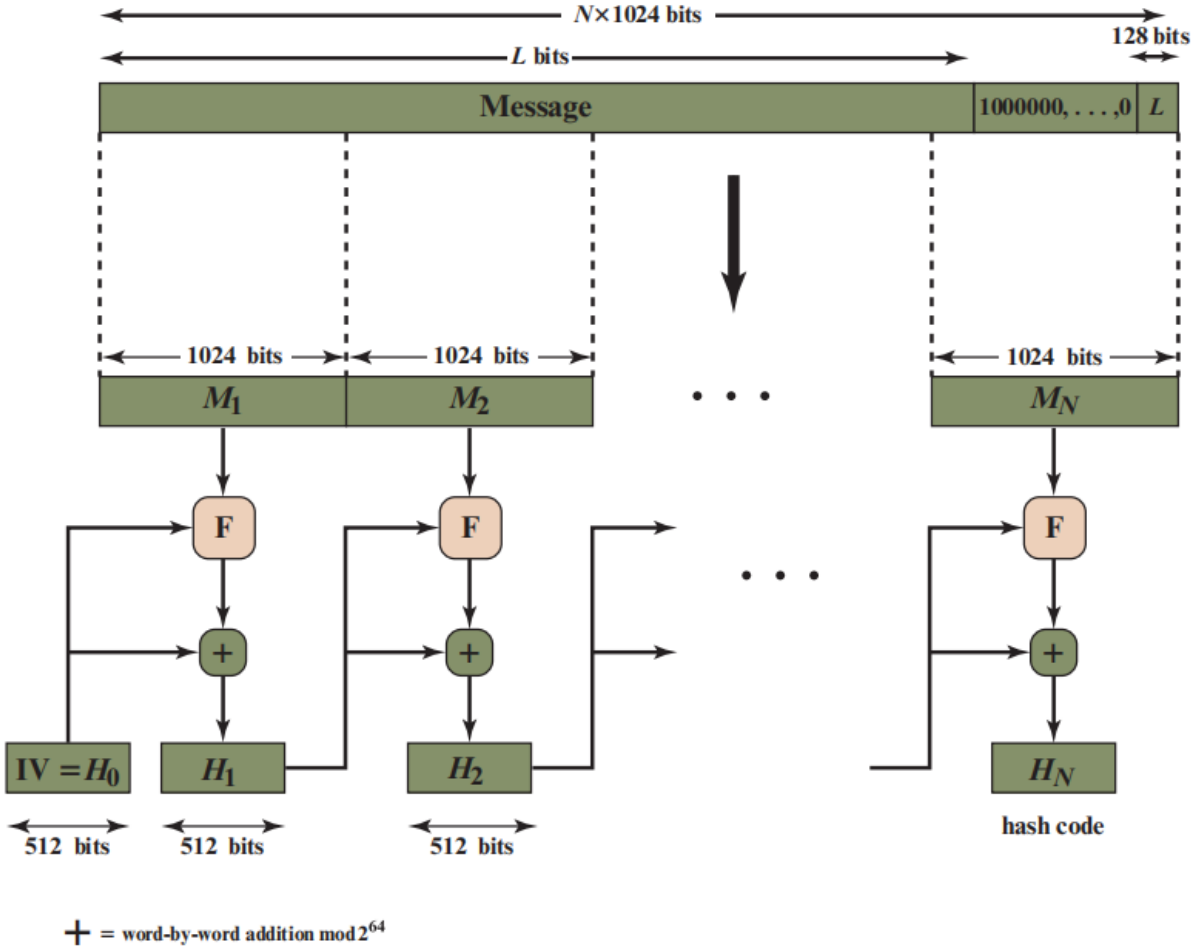
- Nhập message
- Nhập file

THUẬT TOÁN SHA 512



ĐẶ ĐIỂM THUẬT TOÁN SHA 512

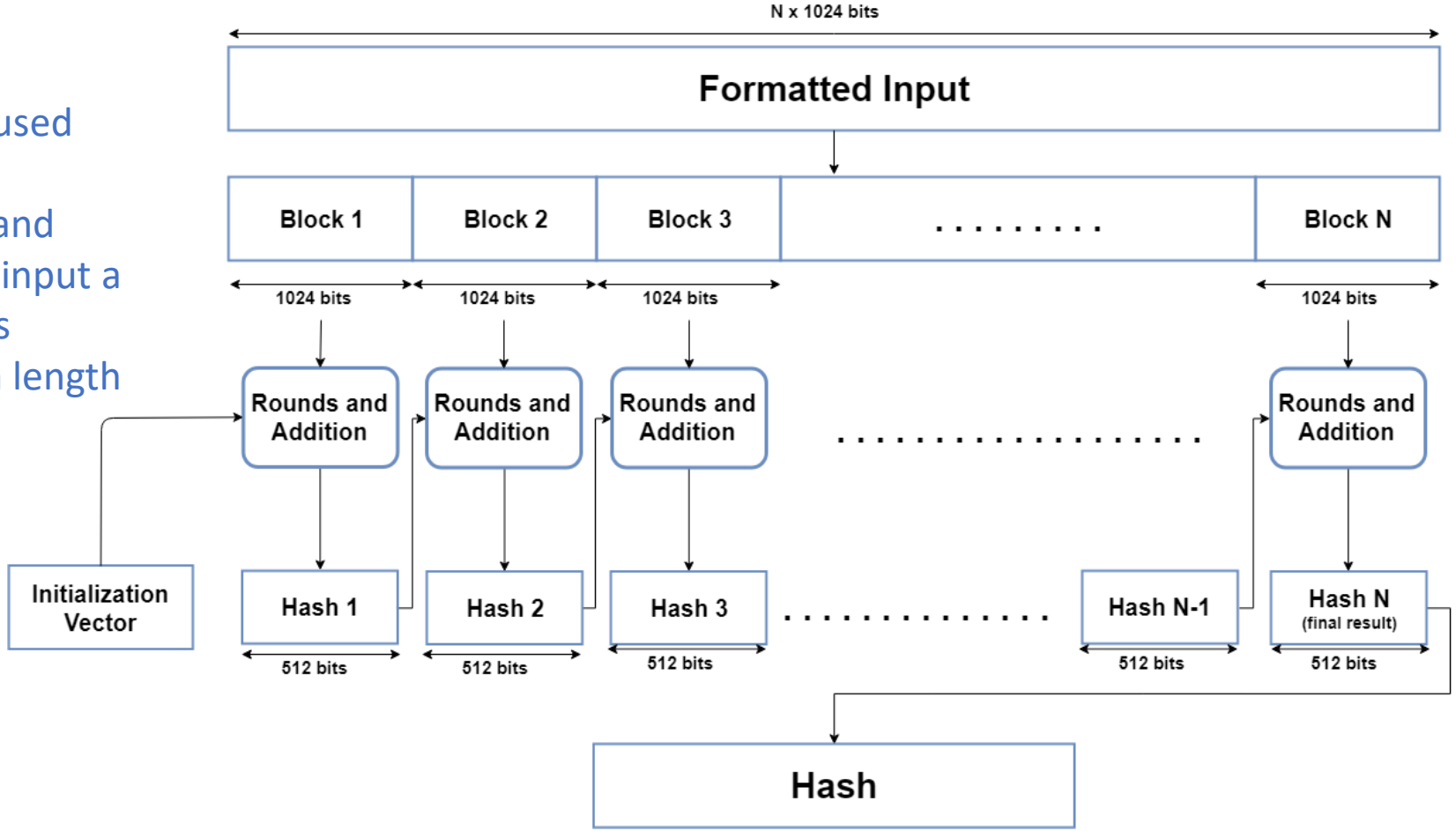
- Robustness and Resistance to various cryptographic attacks: SHA-512 produces a fixed-size 512-bit hash value, providing a vast number of possible output combinations, which enhances its resistance to collision attacks.
- Logical and bitwise operations: Solutions such as modular addition and bitwise rotation ensure the unpredictability and irreversibility of the hash function.
- Iterative Structure: SHA-512 employs an iterative process with multiple rounds of processing, each involving a set of specific mathematical functions.



Mô tả:

- **Input:** Message with a maximum length of less than 2128 bits, 1024-bit blocks.
- **Output:** 512-bit message digest (64 byte).

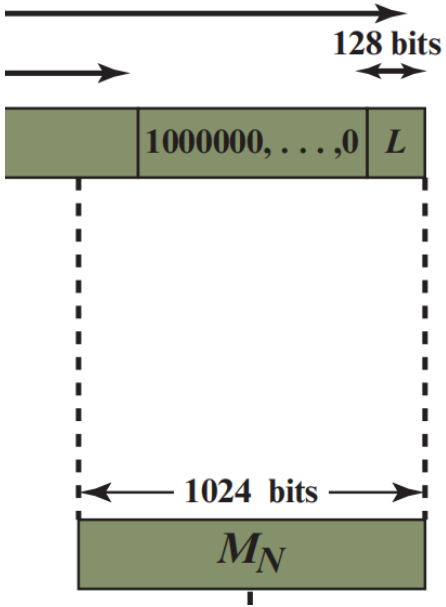
SHA-512 used 1024 bits "blocks", and accept as input a 2^128 bits maximum length string.

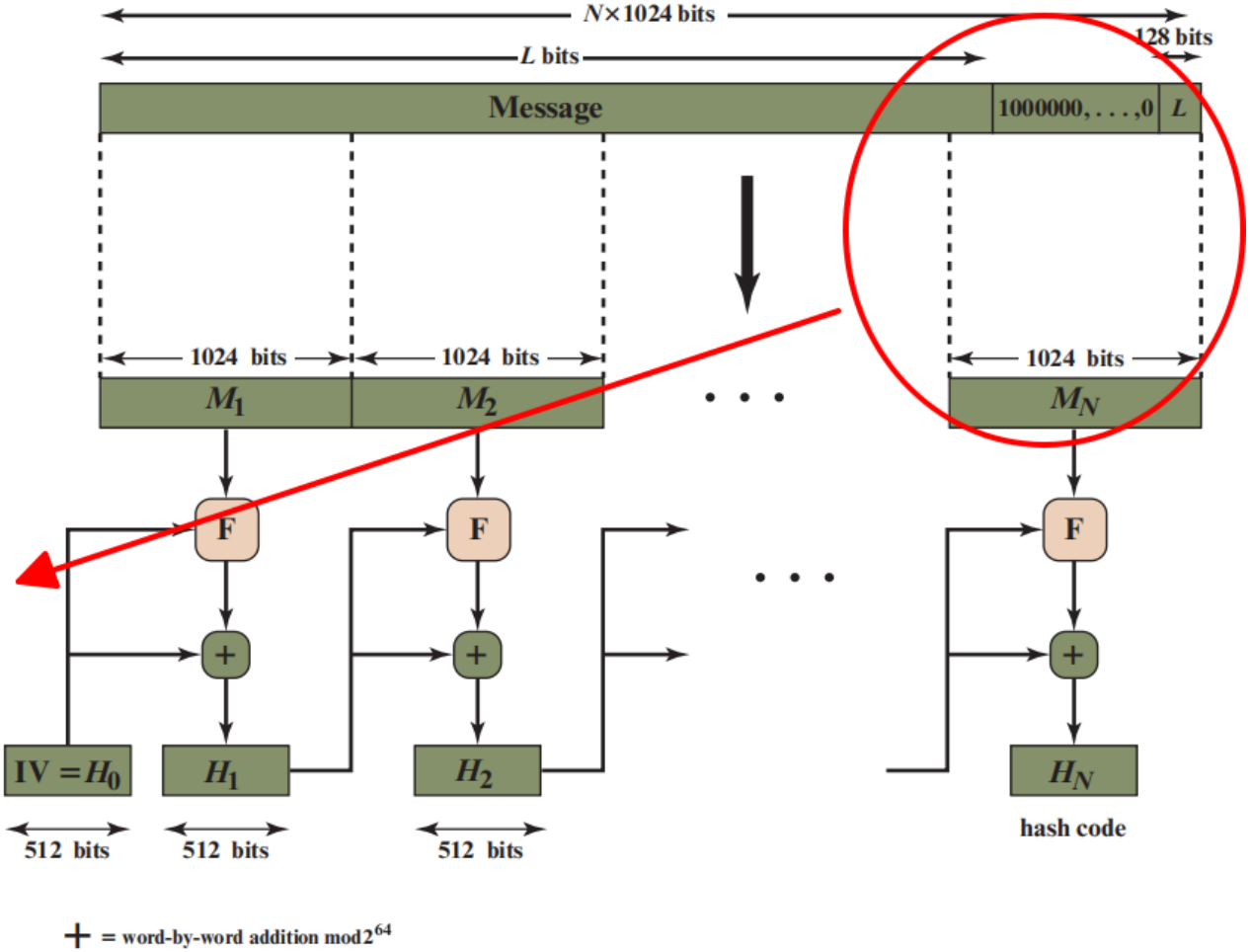


THUẬT TOÁN SHA 512

- **Padding:** The input message is padded with a single "1" bit followed by a variable number of "0" bits until the length of the padded message is congruent to 896 modulo 1024 bits. This ensures that the padded message length is a multiple of 1024 bits.
- **Length Appending:** The length of the original message (in bits) is then appended as a 128-bit big-endian integer, making the total length of the padded message a multiple of 1024 bits.
- **Hash Value Initialization:** Eight 64-bit hash values (H0 to H7) are initialized with predefined constants. These constants are derived from the square roots of the first eight prime numbers.

Step 1: Append padding bit





THUẬT TOÁN SHA 512

- **Step 2:** Append length, a block of 128 bits is appended to the message

- **Step 3:** Initialize hash buffer, a 512-bit buffer (a, b, c, d, e, f, g, h).

```
a = 6A09E667F3BCC908    e = 510E527FADE682D1
b = BB67AE8584CAA73B    f = 9B05688C2B3E6C1F
c = 3C6EF372FE94F82B    g = 1F83D9ABFB41BD6B
d = A54FF53A5F1D36F1    h = 5BE0CD19137E2179
```

THUẬT TOÁN SHA 512

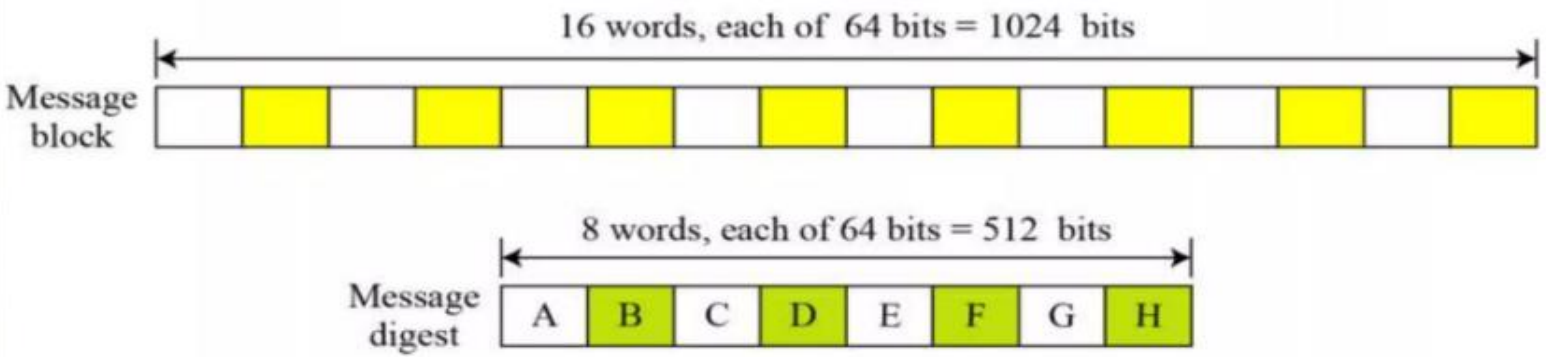
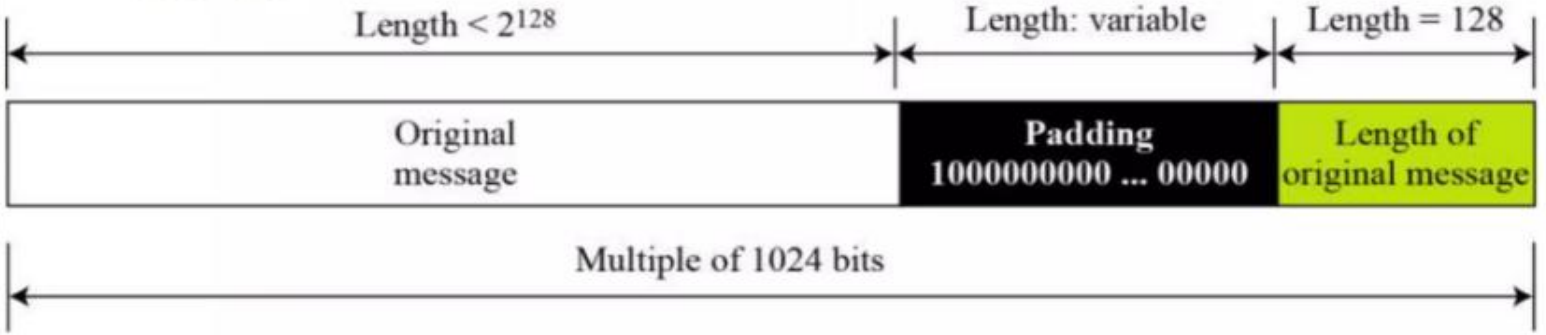


Figure: A message block and the digest as words



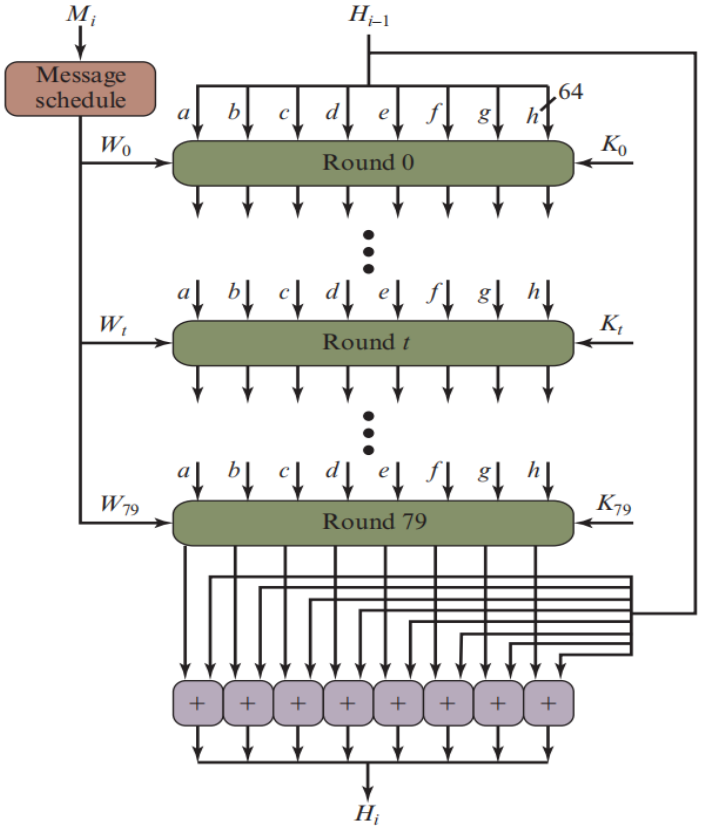
THUẬT TOÁN SHA 512

- **Step 4:** Process message in 1024-bit (128-byte) blocks, 80 rounds.

W_t: 64 bits from M_i

K_t: 64 bits, căn bậc 3 của 80 số nguyên tố ngẫu nhiên đầu tiên (*bảng 80 số tại next slide*).

Kết quả vòng 80 được cộng modulo với 2^{64}



80 HẰNG SỐ

Calculation of constants

For example,

The 8th prime is 19, with the square root $(19)^{1/2}$= 4.35889894354 Converting this number to binary with only 64 bits in the fraction part, we get,

$(100.0101\ 1011\ 1110\ \ldots 1001)_2 \rightarrow (4.5BE0CD19137E2179)_{16}$

The fraction part : $(5BE0CD19137E2179)_{16}$

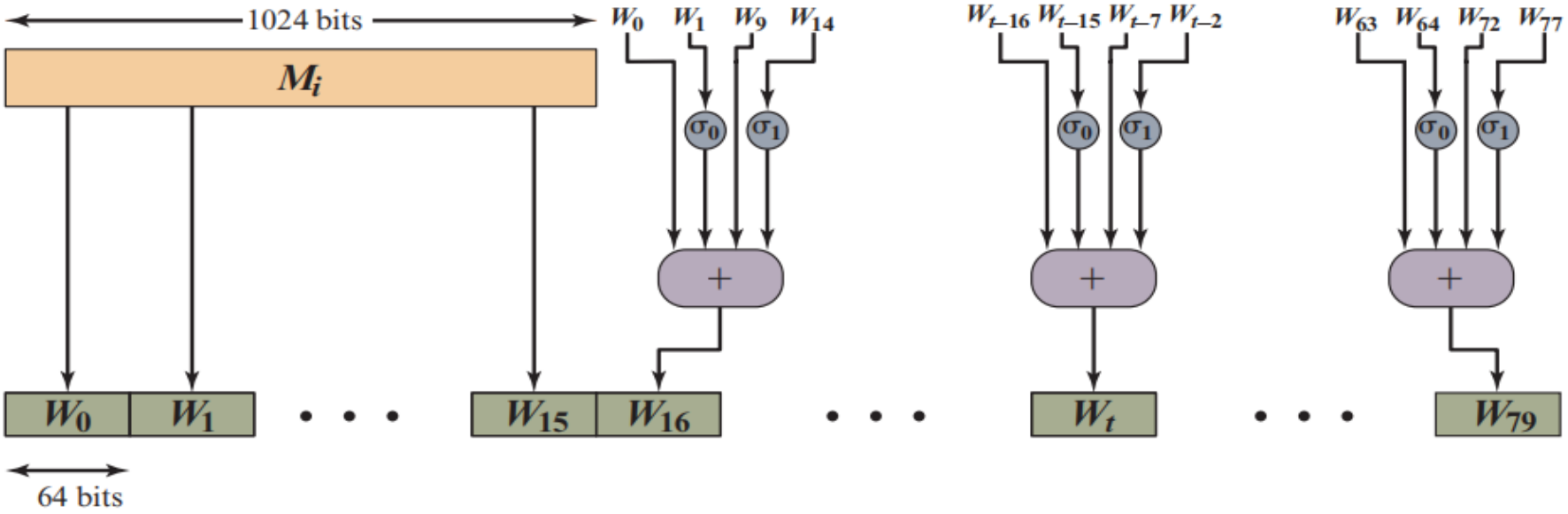
The 80th prime is 409, with the cubic root $(409)^{1/3}$ = 7.42291412044. Converting this number to binary with only 64 bits in the fraction part, we get

$(111.0110\ 1100\ 0100\ 0100\ \ldots\ 0111)_2 \rightarrow (7.6C44198C4A475817)_{16}$

The fraction part: $(6C44198C4A475817)_{16}$

428a2f98d728ae22	7137449123ef65cd	b5c0fbcfec4d3b2f	e9b5dba58189dbbc
3956c25bf348b538	59f111f1b605d019	923f82a4af194f9b	ab1c5ed5da6d8118
d807aa98a3030242	12835b0145706fbe	243185be4ee4b28c	550c7dc3d5ffb4e2
72be5d74f27b896f	80deb1fe3b1696b1	9bdc06a725c71235	c19bf174cf692694
e49b69c19ef14ad2	efbe4786384f25e3	0fc19dc68b8cd5b5	240ca1cc77ac9c65
2de92c6f592b0275	4a7484aa6ea6e483	5cb0a9dcbd41fbd4	76f988da831153b5
983e5152ee66dfab	a831c66d2db43210	b00327c898fb213f	bf597fc7beef0ee4
c6e00bf33da88fc2	d5a79147930aa725	06ca6351e003826f	142929670a0e6e70
27b70a8546d22ffc	2e1b21385c26c926	4d2c6dfc5ac42aed	53380d139d95b3df
650a73548baf63de	766a0abb3c77b2a8	81c2c92e47edaee6	92722c851482353b
a2bfe8a14cf10364	a81a664bbc423001	c24b8b70d0f89791	c76c51a30654be30
d192e819d6ef5218	d69906245565a910	f40e35855771202a	106aa07032bbd1b8
19a4c116b8d2d0c8	1e376c085141ab53	2748774cdf8eeb99	34b0bcb5e19b48a8
391c0cb3c5c95a63	4ed8aa4ae3418acb	5b9cca4f7763e373	682e6ff3d6b2b8a3
748f82ee5defb2fc	78a5636f43172f60	84c87814a1f0ab72	8cc702081a6439ec
90befffa23631e28	a4506cebde82bde9	bef9a3f7b2c67915	c67178f2e372532b
ca273eceea26619c	d186b8c721c0c207	eada7dd6cde0eb1e	f57d4f7fee6ed178
06f067aa72176fba	0a637dc5a2c898a6	113f9804bef90dae	1b710b35131c471b
28db77f523047d84	32caab7b40c72493	3c9ebe0a15c9bebc	431d67c49c100d4c
4cc5d4becb3e42b6	597f299cfc657e2a	5fcb6fab3ad6faec	6c44198c4a475817

Phương pháp sinh W_i



Dãy từ mở rộng W_i được sinh ra từ khối dữ liệu đầu vào M (đã được nạp vào bộ đệm 1024 bit và chia thành 16 từ đầu tiên W0,W1,...,W15). Sau đó, các từ từ W16 đến W79 được sinh ra theo công thức cụ thể.

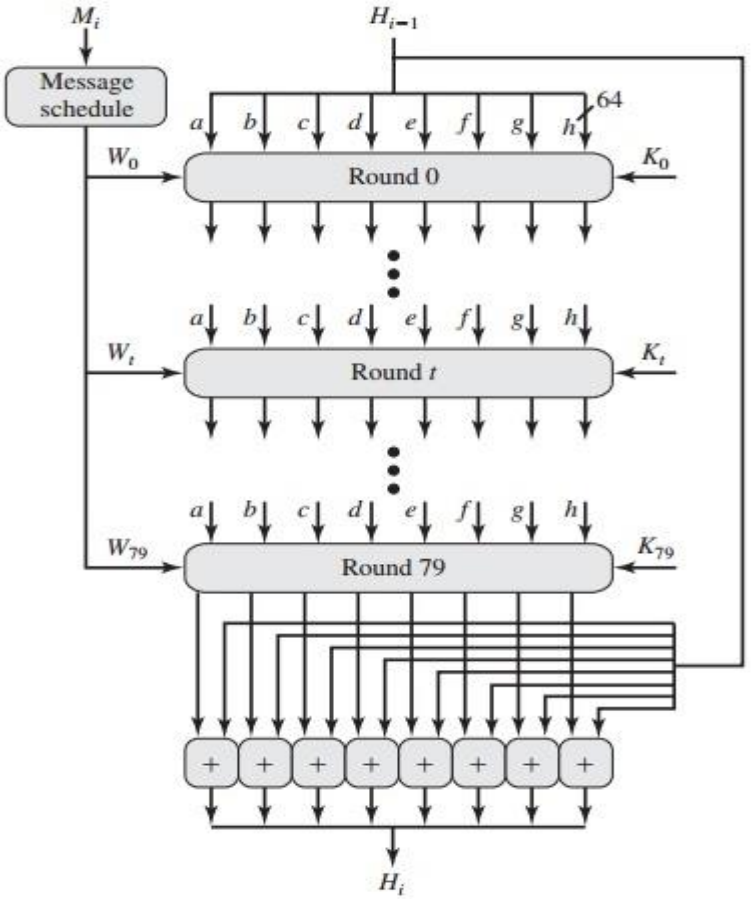


Figure 11.9 SHA-512 Processing of a Single 1024-Bit Block

The output of the eightieth round is added to the input to the first round (H_{i-1}) to produce H_i. The addition is done independently for each of the eight words in the buffer with each of the corresponding words in H_{i-1}, using addition modulo 2^{64}.

THUẬT TOÁN SHA 512

- **Message Parsing: The padded message is divided into 1024-bit (128-byte) blocks.**
- **Block Processing: Each block is processed through a series of 80 rounds. In each round, the following operations are performed:**
  - ✓ **Bitwise operations (AND, XOR, NOT) are applied to the current *hash values* and the *message block*.**
  - ✓ **The hash values are updated using *modular* addition and bitwise rotations.**

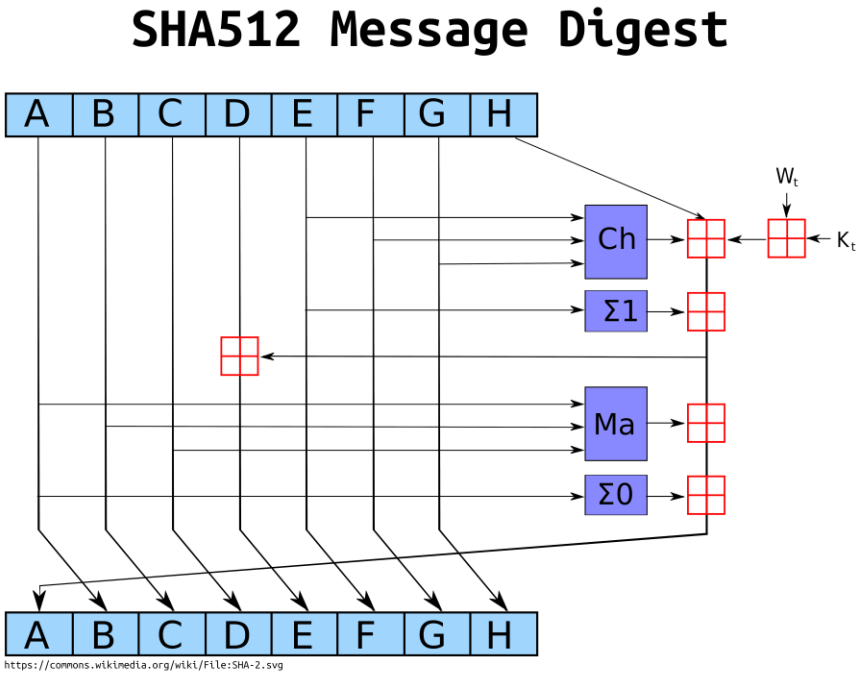
THUẬT TOÁN SHA 512

Phép cộng Modular

Trong **SHA-512**, các thanh ghi có kích thước **64-bit**, nên cộng theo:

result = (a+b) mode 2^{64}

- Kết quả luôn có **64-bit (8 bytes)**.
- **SHA-512 sử dụng 64-bit, còn SHA-256 sử dụng 32-bit.**



$Ch(E,F,G) = (E \wedge F) \oplus (\neg E \wedge G)$

$Ma(A,B,C) = (A \wedge B) \oplus (A \wedge C) \oplus (B \wedge C)$

$\Sigma_0(A) = (A \ggg 28) \oplus (A \ggg 34) \oplus (A \ggg 39)$

$\Sigma_1(E) = (E \ggg 14) \oplus (E \ggg 18) \oplus (E \ggg 41)$

SHA-512 Round Function

Majority Function

$$(A_j \text{ AND } B_j) \oplus (B_j \text{ AND } C_j) \oplus (C_j \text{ AND } A_j)$$

Conditional Function

$$(E_j \text{ AND } F_j) \oplus (\text{NOT } E_j \text{ AND } G_j)$$

Rotate Functions

$$\text{Rotate (A): } \text{RotR}_{28}(A) \oplus \text{RotR}_{34}(A) \oplus \text{RotR}_{29}(A)$$

$$\text{Rotate (E): } \text{RotR}_{28}(E) \oplus \text{RotR}_{34}(E) \oplus \text{RotR}_{29}(E)$$

THUẬT TOÁN SHA 512

THUẬT TOÁN SHA-512

- **Step 5:** Output, 512-bit message digest.

$$H_0 = \text{IV}$$
$$H_i = \text{SUM}_{64}(H_{i-1}, \text{abcdefgh}_i)$$
$$MD = H_N$$

IV	Giá trị khởi tạo của abcdef
N	Số block của thông tin
SUM_{64}	Phép cộng modulo 2^{64} vòng thứ 80
MD	Kết quả là giá trị tóm tắt

THUẬT TOÁN SHA 512

- The message block is mixed with the hash values using bitwise operations and modular addition.
- Hash Value Update: After processing each block, the resulting hash values are added to the previous hash values using modular addition.
- Output: The final hash value is obtained by concatenating the eight 64-bit hash values (H0 to H7) in big-endian order, resulting in a 512-bit (64-byte) hash value.

THUẬT TOÁN SHA 512

Padding SHA512: Khối = 1024 bit = 128 bytes, mã băm SHA512 = 512 bit = 64 bytes

- Băm chuỗi 100 ký tự (100 bytes)
- Tách thành 1 khối để băm
- Khối 1 = 100 bytes data + độn thêm 12 bytes (cho đủ 112 bytes = 896 bits) + 16 bytes (chiều dài dữ liệu gốc)
- 100 bytes data + bit 1 + 95 bit 0 + 0000 0000 0000 0000 0000 0011 0010 0000
- 100 bytes data + 8000 0000 0000 0000 0000 0000 0000 0000 (hệ thập lục phân) + 0000 0000 0000 0000 0000 0011 0010 0000

THUẬT TOÁN SHA 512

1. Chia nhỏ dữ liệu để xử lý

- **Dữ liệu đầu vào**: 100 bytes (800 bits).
- Vì **100 bytes < 128 bytes**, chỉ cần **1 khối** để xử lý

2. Padding (đệm dữ liệu):

Quy tắc padding SHA-512

- **Thêm 1 bit 1** ngay sau dữ liệu.
- **Thêm các bit 0** sao cho tổng số bit đạt **896-bit** (112 bytes).

→ **Ở đây, cần thêm 12 bytes dữ liệu 0**.

- **Thêm 16 bytes (128-bit) cuối cùng** để biểu diễn độ dài dữ liệu ban đầu.

THUẬT TOÁN SHA 512

3. Cách padding

Cấu trúc dữ liệu sau padding

- **100 bytes dữ liệu**.
- **Thêm bit 1**.
- **Thêm 95 bit 0** → (tổng cộng 12 bytes padding để đạt 896 bits).
- **Thêm 16 bytes cuối cùng chứa độ dài dữ liệu gốc (800 bits)**.

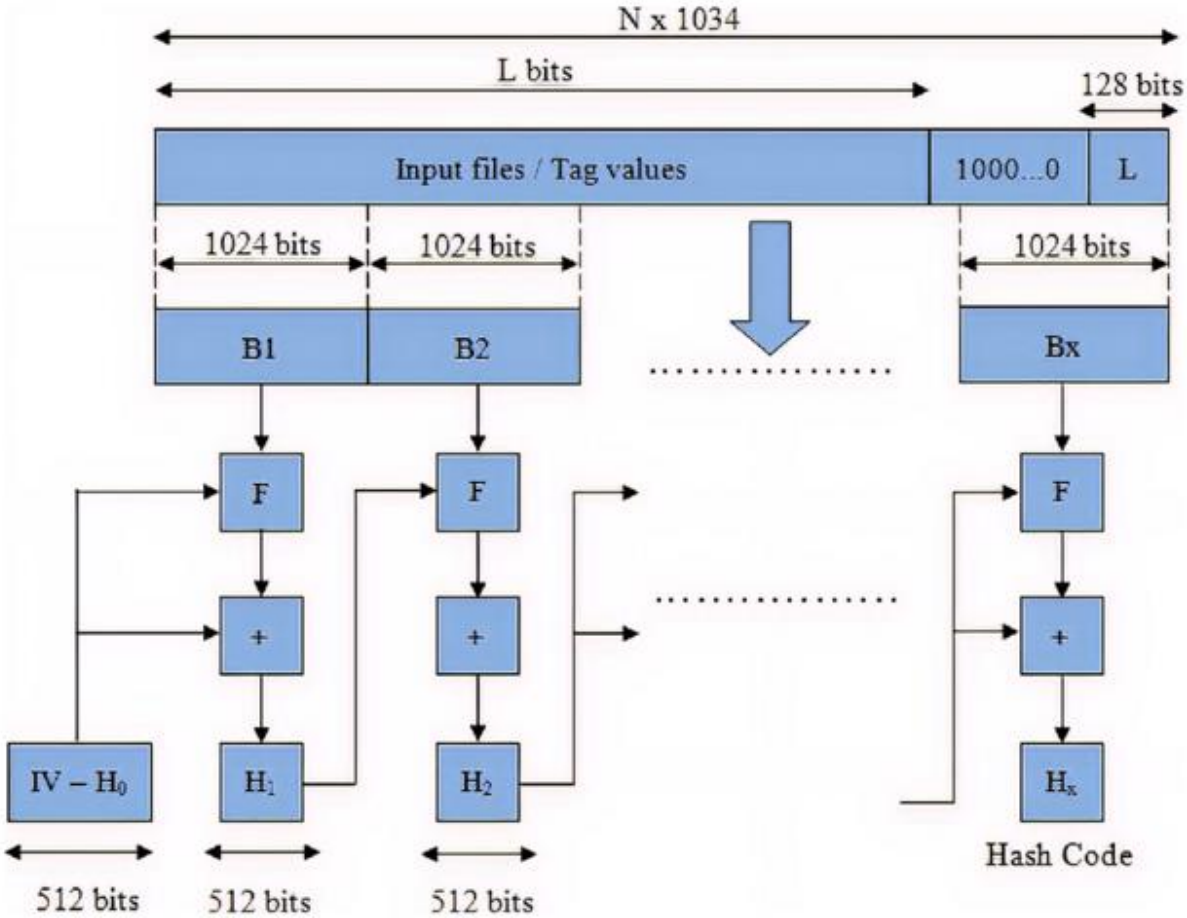
Biểu diễn số 800 trong 128-bit

- 800 trong hệ nhị phân

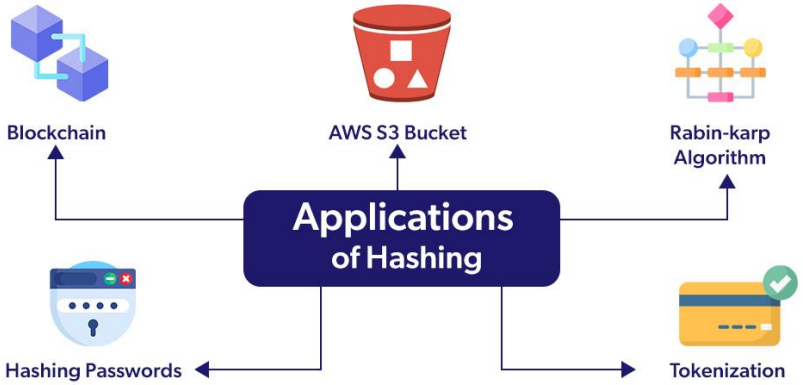
$$1100100000_2$$

- Biểu diễn trong 128-bit (16 bytes):

0000000000000000 …(3) 0000001100100000


N x 1034
L bits
128 bits
Input files / Tag values
1000...0
L
1024 bits
1024 bits
1024 bits
B1
B2
Bx
F
F
F
+
+
+
IV – H0
H1
H2
Hx
Hash Code
512 bits
512 bits
512 bits

ỨNG DỤNG

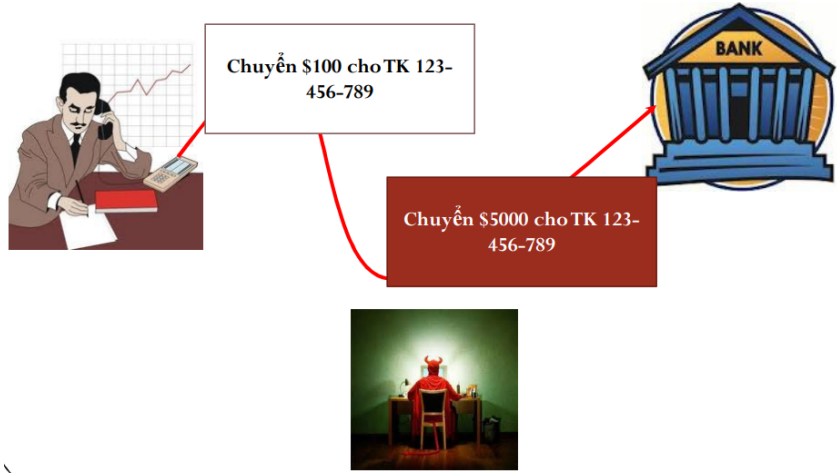


- **Bảo mật mật khẩu.**
- **Chữ ký số và chứng thực.**
- **Blockchain và tiền mã hóa.**
- **Kiểm tra tính toàn vẹn của tệp:** Kiểm tra tính toàn vẹn của tệp tin khi tải xuống từ internet
- **Hệ thống quản lý dữ liệu và cơ sở dữ liệu:** giúp truy vấn dữ liệu nhanh trong lập trình và cơ sở dữ liệu.

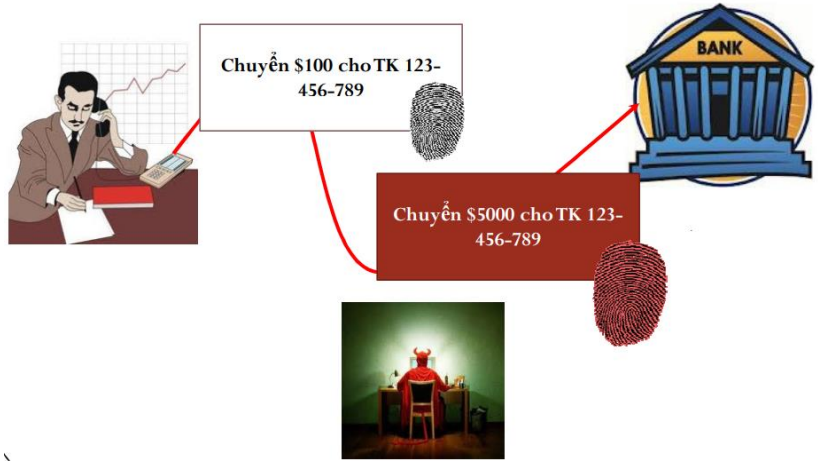
ỨNG DỤNG

- **Ứng dụng kiểm tra tính toàn vẹn**

Toàn vẹn dữ liệu (Integrity)

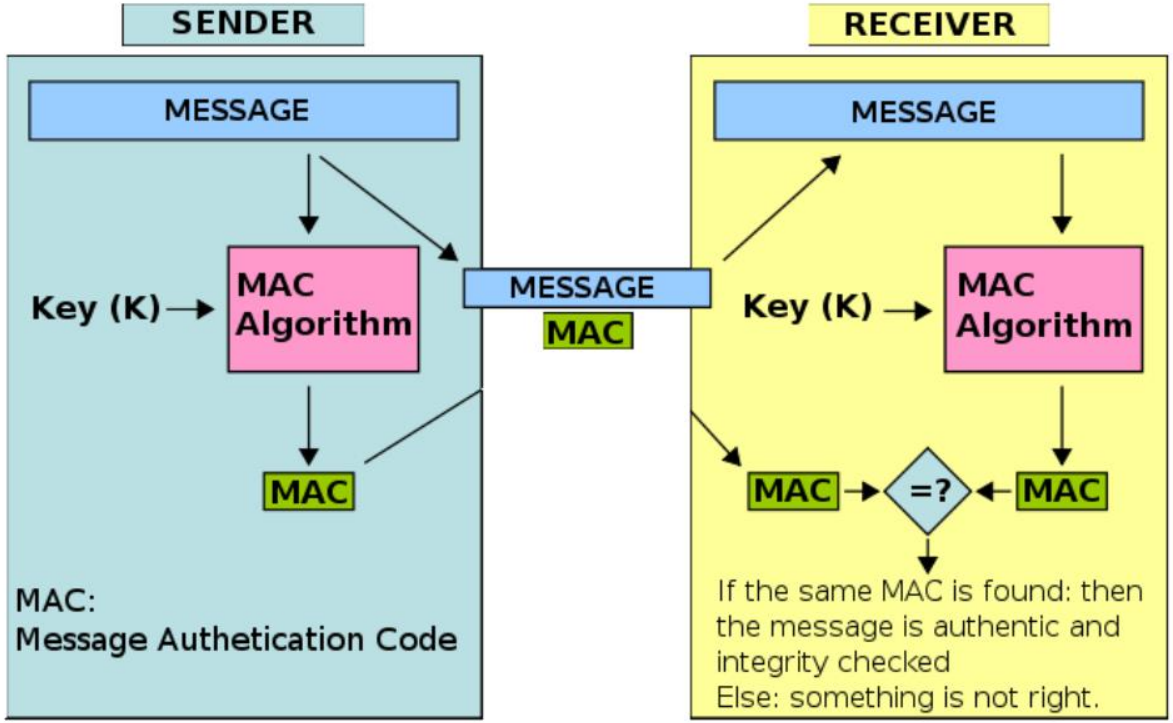


Toàn vẹn dữ liệu (Integrity)



ỨNG DỤNG

- **Fingerprint**
- **Face recognition**
- **Transaction**



SUMMARY

- Hàm băm
- Thuật toán SHA
- Ứng dụng

LUYỆN TẬP

Mã nguồn luyện tập thuật toán SHA

- Code in C++:
  - https://github.com/pr0f3ss/SHA
- Using Lib in Pyhton:
  - https://hashing.ssojet.com/sha-512-in-python/
  - https://hashing.ssojet.com/sha-512-in-python/
  - https://github.com/keanemind/python-sha-256
- Using Lin in Java:
  - https://www.geeksforgeeks.org/sha-512-hash-in-java/

LUYỆN TẬP

- **Bài 1:** Lập trình Python băm dữ liệu nhập qua message bằng MD5, SHA-512
- **Bài 2:** Lập trình Python kiểm tra tính toàn vẹn của file bằng MD5, SHA-512.

ĐẠI NAM
UNIVERSITY