CHƯƠNG 4
HÀM BĂM VÀ CHỮ KÝ SỐ

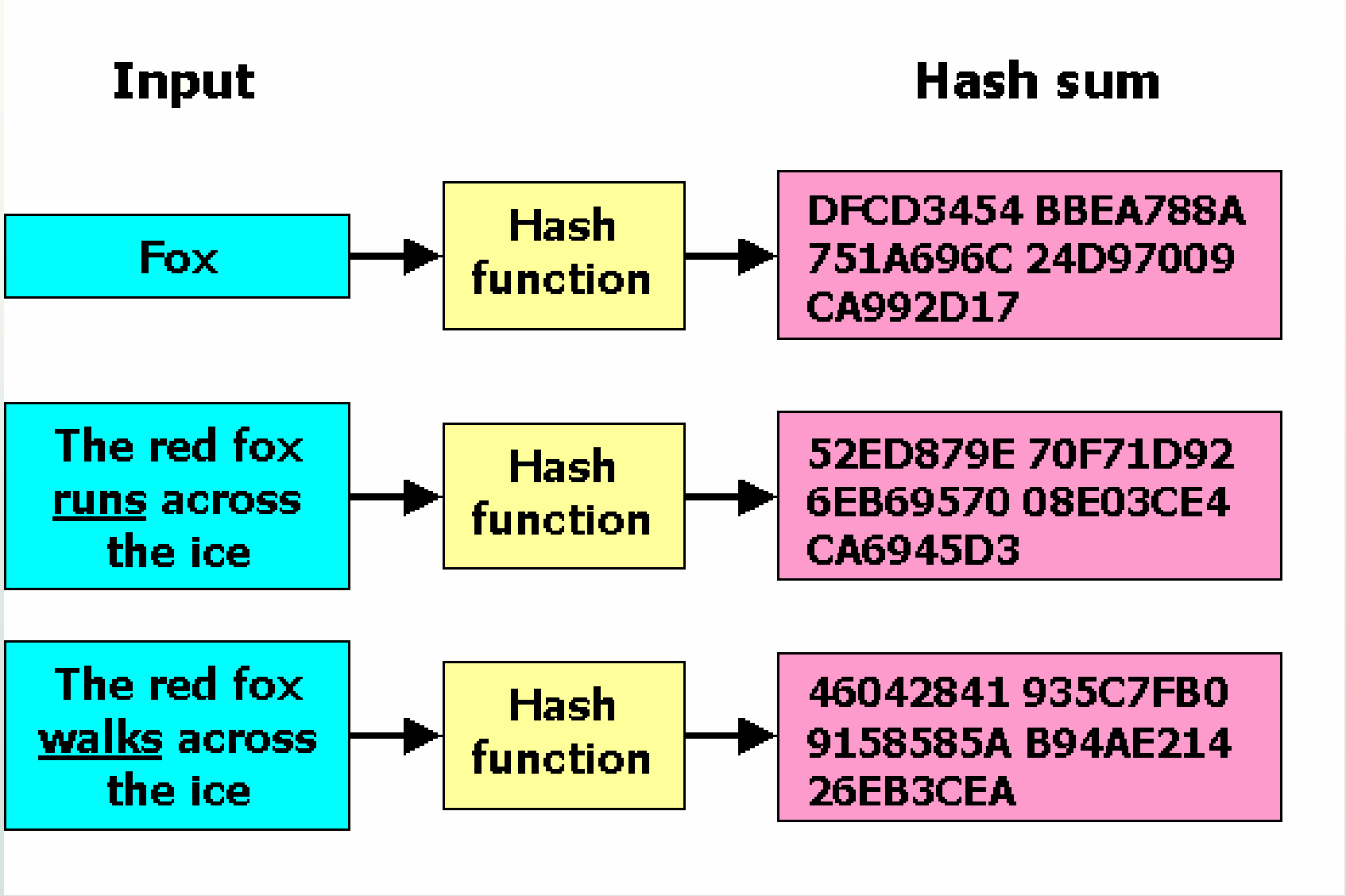
Hàm băm

- Các ứng dụng chú trọng mục tiêu **toàn vẹn**
  - Tài liệu được sử dụng giống tài liệu lưu trữ
  - Các thông điệp trao đổi trong một hệ thống an toàn không bị thay đổi/sửa chữa
- "Niêm phong" tài liệu/thông điệp
  - "Niêm phong" không bị sửa đổi/phá hủy → tài liệu/thông điệp toàn vẹn
  - "Niêm phong": băm (hash), tóm lược (message digest), đặc số kiểm tra (checksum)
  - Tạo ra "niêm phong": hàm băm

Hàm băm

- Mục tiêu: các hàm băm h tạo ra bản nhận dạng (fingerprint) cho một tập tin, thông điệp hay một khối dữ liệu truyền đi nhằm kiểm tra tính toàn vẹn
- Đặc điểm:
  - H có thể được áp dụng trên khối dữ liệu có độ dài bất kỳ
  - H tạo đầu ra có độ dài cố định
  - H(x) tính toán mọi x tương đối dễ dàng, tạo điều kiện cho việc cài đặt trên phần cứng lẫn phần mềm được thiết thực

Hàm băm


Input
Hash sum
Fox
Hash function
DFCD3454 BBEA788A 751A696C 24D97009 CA992D17
The red fox runs across the ice
Hash function
52ED879E 70F71D92 6EB69570 08E03CE4 CA6945D3
The red fox walks across the ice
Hash function
46042841 935C7FB0 9158585A B94AE214 26EB3CEA

Hàm băm

❖ Đặc điểm:

- Với bất kỳ giá trị băm h, không thể tính được x sao cho $H(x) = h$ hay ***H*** **được gọi là hàm một chiều**
- **Tính bền xung đột yếu (weak collision resistance)**: với bất kỳ giá trị x, không thể tính được $y \neq x$ sao cho $H(y) = H(x)$
- **Tính bền xung đột mạnh (strong collision resistance)**: không thể tính được một cặp (x, y) sao cho $H(x) = H(y)$

Hàm băm

- Hàm băm có khóa
  - $\boldsymbol{H : \Sigma^* \times K \rightarrow \Sigma^n}$
- Hàm băm không khóa
  - $\boldsymbol{H : \Sigma^* \rightarrow \Sigma^n}$

Kỹ thuật tạo hàm băm

- Dùng các hàm mã hóa
  - CBC
  - RMDP
  - DM
- Dùng các phép toán số học đồng dư
  - QCMDC
  - DP
- Dùng các hàm thiết kế đặc biệt
  - MD4/MD5
  - SHA/SHS

CBC - Chaining Block Cipher

- Mật mã đối xứng
  - Hàm mã hóa E
  - Khóa K
- Hàm băm
  - $m = m_1 m_2 \ldots m_n$
  - $h_i = E(K, m_i \; XOR \; h_{i-1})$
  - $h = h_n$

RMDP - Rabin, Matyas, Davise, Price

- Mật mã đối xứng
  - Hàm mã hóa E
  - Khóa là các khối của tin
- Hàm băm
  - $m = m_1 m_2 \dots m_n$
  - $h_0 = r\ (r\ ng\tilde{a}u\ nhi\hat{e}n)$
  - $h_i = E(m_i, h_{i-1})$
  - $h = h_n$

DM - Davies, Meyer

- Mật mã đối xứng
  - Hàm mã hóa E
  - Khóa là các khối của tin
- Hàm băm
  - $m = m_1 m_2 \ \ldots m_n$
  - $h_0 = r \ (r \ ng\tilde{a}u \ nhi\hat{e}n)$
  - $h_i = E(m_i, h_{i-1}) \ XOR \ h_{i-1}$
  - $h = h_n$

QCMDC

❖ QCMDC - Quadratic Congruential Manipulation Detection Code

- $m = m_1 m_2 \ldots m_n$ với m_i là khối n bit
- p là số nguyên tố sao cho $p \geq 2^{n-1}$
- Hàm băm:
  - $h_0 = r\ (r\ ngẫu\ nhiên)$
  - $h_i = (h_{i-1} + m_i)^2\ mod\ p$
  - $h = h_n$

DP - Davies, Price

- $m = m_1 m_2 \ldots m_n$ với m_i là khối n bit
- p là số nguyên tố sao cho $p \geq 2^r$
- Hàm băm:
  - $h_0 = 0$
  - $h_i = (h_{i-1}\ XOR\ m_i)^2\ mod\ p$
  - $h = h_n$

Giải thuật MD5

- Phát triển bởi Ron Rivest tại đại học MIT Input: thông điệp với độ dài bất kỳ
- Output: giá trị băm (message digest) 128 bit
- Giải thuật gồm 5 bước thao tác trên khối 512 bit

Giải thuật MD5 - Nguyên lý

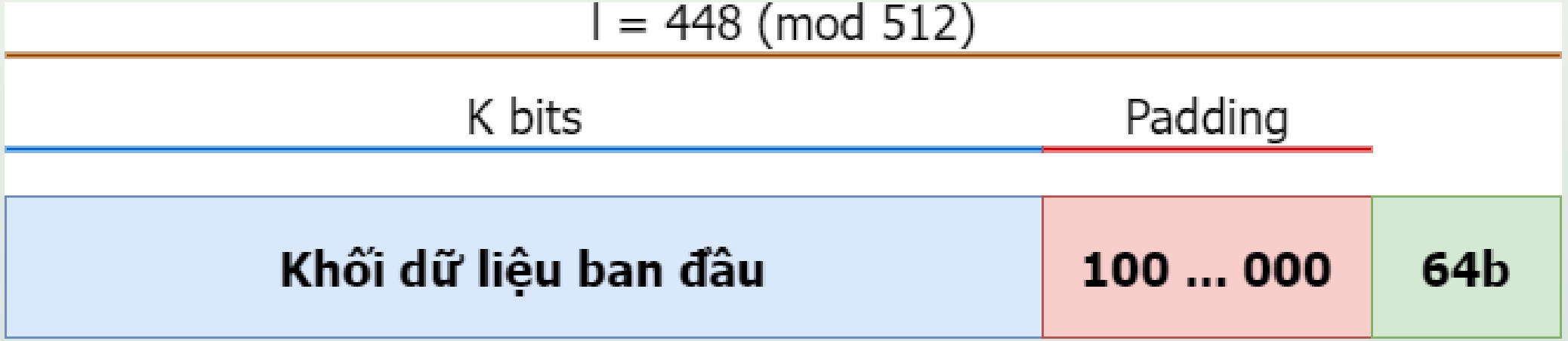
- Bước 1: **nhồi dữ liệu**
  - Nhồi thêm các bit sao cho dữ liệu có độ dài

    $l \equiv 448 \, mod \, 512$ hay

    $l = n * 512 + 448$ (n, l nguyên)
  - **Luôn thực hiện nhồi dữ liệu ngay cả khi dữ liệu ban đầu có độ dài mong muốn.**
    - Ví dụ, dữ liệu có độ dài 448 được nhồi thêm 512 bit để được độ dài 960 bit.

Giải thuật MD5 - Nguyên lý

- Bước 1: **nhồi dữ liệu**
  - Số lượng bit nhồi thêm nằm trong khoảng 1 đến 512
  - Các bit được nhồi gồm 1 bit "1" và các bit 0 theo sau



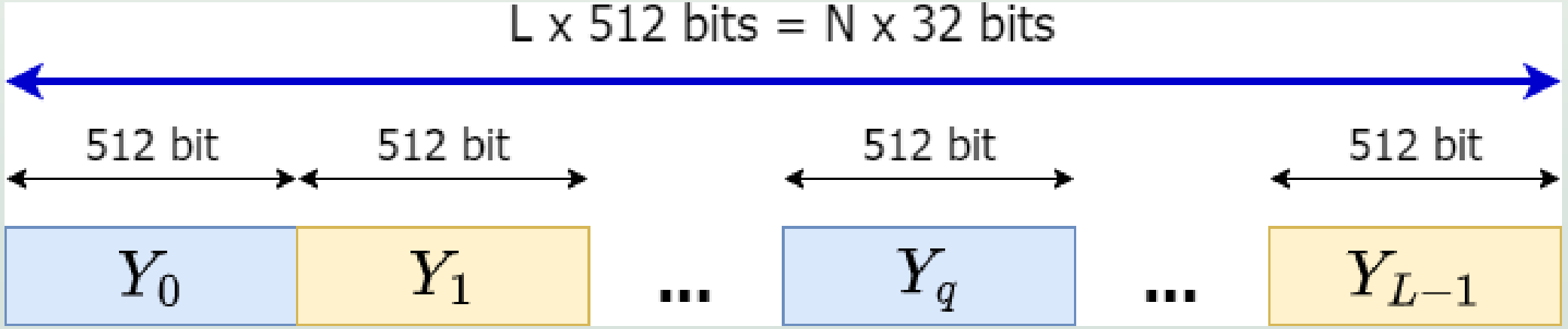
Giải thuật MD5 - Nguyên lý

- Bước 2: **thêm vào độ dài**
  - Độ dài của khối dữ liệu ban đầu được biểu diễn dưới dạng nhị phân 64-bit và được thêm vào cuối chuỗi nhị phân kết quả của bước 1
  - Nếu độ dài của khối dữ liệu ban đầu > 264, chỉ 64 bit thấp được sử dụng, nghĩa là giá trị được thêm vào bằng $K \ mod \ 264$

Giải thuật MD5 - Nguyên lý

- Kết quả có được từ 2 bước đầu là một khối dữ liệu có độ dài là bội số của 512. Khối dữ liệu được biểu diễn:
  - Bằng một dãy L khối 512-bit $Y_0, Y_1, \ldots, Y_{L-1}$
  - Bằng một dãy N từ (word) 32 bit $M_0, M_1, \ldots, M_{N-1}$

  Vậy $N = L \times 16$ $(32 \times 16 = 512)$



Giải thuật MD5 - Nguyên lý

- Bước 3: **khởi tạo bộ đệm MD (MD buffer)**
- Một bộ đệm 128-bit được dùng lưu trữ các giá trị băm trung gian và kết quả.
- Bộ đệm được biểu diễn bằng 4 thanh ghi 32-bit với các giá trị khởi tạo ở dạng little-endian (byte có trọng số nhỏ nhất trong từ nằm ở địa chỉ thấp nhất) như sau:

A = 67 45 23 01 B = EF CD AB 89

C = 98 BA DC FE D = 10 32 54 76

Giải thuật MD5 - Nguyên lý

- Bước 3: **khởi tạo bộ đệm MD (MD buffer)**
  - Các giá trị này tương đương với các từ 32 bit:
    - A = 01 23 45 67
    - B = 89 AB CD EF
    - C = FE DC BA 98
    - D = 76 54 32 10

Giải thuật MD5 - Nguyên lý

- Bước 4: **xử lý các khối dữ liệu 512 bit**
  - Trọng tâm của giải thuật là hàm nén gồm 4 "vòng" xử lý.
  - Các vòng có cấu trúc giống nhau nhưng sử dụng các hàm luận lý khác nhau gồm:
    - $F\ (X, Y, Z) = X \wedge Y \vee \neg X \wedge Z$
    - $G(X, Y, Z)\ = X \wedge Z \vee Y \wedge \neg Z$
    - $H(X, Y, Z) = X\ xor\ Y\ xor\ Z$
    - $I(X, Y, Z) = Y\ xor(X\ \vee\ \neg Z)$

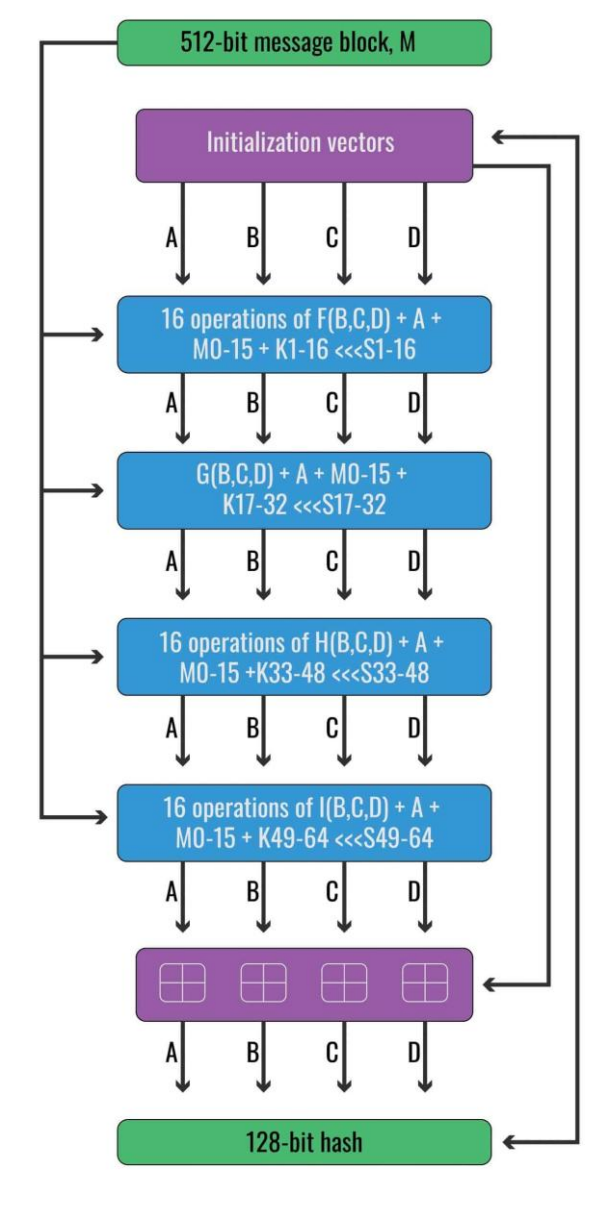
Giải thuật MD5 - Nguyên lý

- Mảng 64 phần tử được tính theo công thức:
  $T[i] = \lfloor 232 \times abs(\sin(i)) \rfloor$, i được tính theo radian
- Kết quả của 4 vòng được cộng (theo modulo 2^{32} với đầu vào CV_q để tạo CV_{q+1})

❖ Bước 5: **xuất kết quả**

- Sau khi xử lý hết L khối 512-bit, đầu ra của lần xử lý thứ L là giá trị băm 128 bits

Giải thuật MD5 - Nguyên lý



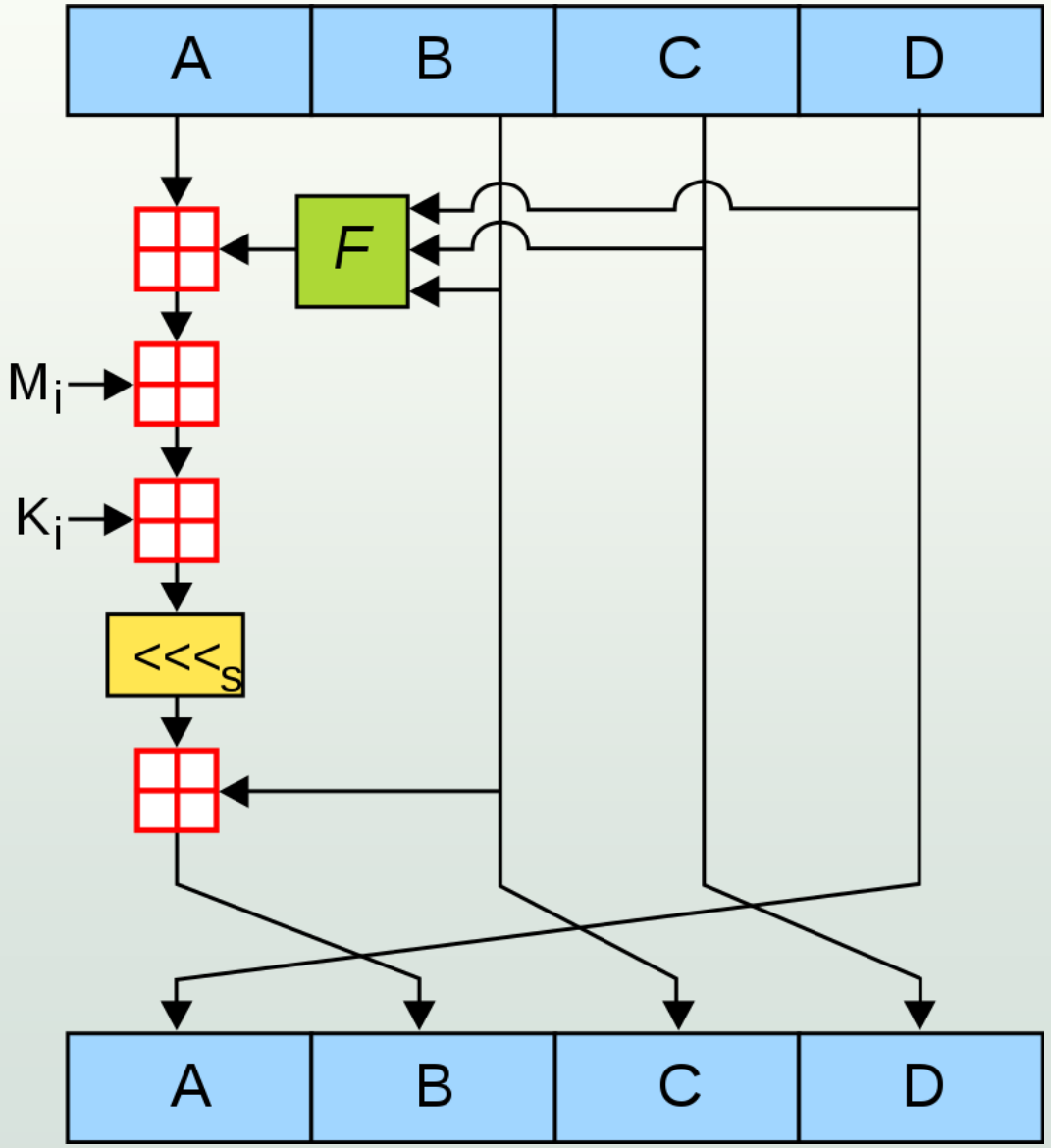
Giải thuật MD5 - Nguyên lý

T[1] = d76aa478	T[17] = f61e2562	T[33] = fffa3942	T[49] = f4292244
T[2] = e8c7b756	T[18] = c040b340	T[34] = 8771f681	T[50] = 432aff97
T[3] = 242070db	T[19] = 265e5a51	T[35] = 6d9d6122	T[51] = ab9423a7
T[4] = c1bdceee	T[20] = e9b6c7aa	T[36] = fde5380c	T[52] = fc93a039
T[5] = f57c0faf	T[21] = d62f105d	T[37] = a4beea44	T[53] = 655b59c3
T[6] = 4787c62a	T[22] = 2441453	T[38] = 4bdecfa9	T[54] = 8f0ccc92
T[7] = a8304613	T[23] = d8a1e681	T[39] = f6bb4b60	T[55] = ffeff47d
T[8] = fd469501	T[24] = e7d3fbc8	T[40] = bebfbc70	T[56] = 85845dd1
T[9] = 698098d8	T[25] = 21e1cde6	T[41] = 289b7ec6	T[57] = 6fa87e4f
T[10] = 8b44f7af	T[26] = c33707d6	T[42] = eaa127fa	T[58] = fe2ce6e0
T[11] = ffff5bb1	T[27] = f4d50d87	T[43] = d4ef3085	T[59] = a3014314
T[12] = 895cd7be	T[28] = 455a14ed	T[44] = 4881d05	T[60] = 4e0811a1
T[13] = 6b901122	T[29] = a9e3e905	T[45] = d9d4d039	T[61] = f7537e82
T[14] = fd987193	T[30] = fcefa3f8	T[46] = e6db99e5	T[62] = bd3af235
T[15] = a679438e	T[31] = 676f02d9	T[47] = 1fa27cf8	T[63] = 2ad7d2bb
T[16] = 49b40821	T[32] = 8d2a4c8a	T[48] = c4ac5665	T[64] = eb86d391

Giải thuật MD5 - Hàm nén

- Mỗi vòng thực hiện 16 bước, mỗi bước thực hiện các phép toán để cập nhật giá trị buffer ABCD, mỗi bước được mô tả như sau
  - $B \leftarrow B + \left((A + F(B, C, D) + X[k] + T[i\ ]) <<< s\right)$
    - A,B,C,D: các từ của thanh ghi
    - F: một trong các hàm F, G, H, I
    - $<<<\ s$: dịch vòng trái s bit
    - $M_i \sim X[k]$: từ 32-bit thứ k của khối dữ liệu 512 bit. k = 1 . . . 15
    - $K_i \sim T[i]$: giá trị thứ i trong bảng T
    - +: phép toán cộng modulo 232

Giải thuật MD5 - Hàm nén



Giải thuật SHA-1

- Secure Hash Algorithm (SHA) phát triển bởi National Institute of Standard and Technology (NIST)
- Đầu vào: thông điệp với độ dài tối đa 264 bit
- Đầu ra: giá trị băm (message digest) có độ dài 160 bit
- Giải thuật gồm 5 bước thao tác trên các khối 512 bit

Giải thuật SHA-1: Nguyên lý

- Bước 1: **nhồi thêm dữ liệu**
  - Thông điệp được nhồi thêm các bit sao cho độ dài

    $l \equiv 448 \, mod \, 512$ hay

    $l = n * 512 + 448$ (n, l nguyên)
  - Thông điệp luôn luôn được nhồi thêm dữ liệu
  - Số bit nhồi thêm nằm trong khoảng 1 đến 512
  - Phần dữ liệu nhồi thêm bao gồm một bit 1 và theo sau là các bit 0

Giải thuật SHA-1: Nguyên lý

❖ Bước 2: thêm vào độ dài

- Độ dài của khối dữ liệu ban đầu được biểu diễn dưới dạng nhị phân 64-bit và được thêm vào cuối chuỗi nhị phân kết quả của bước 1
- Độ dài được biểu diễn dưới dạng nhị phân 64-bit không dấu
- Kết quả có được từ 2 bước đầu là một khối dữ liệu có độ dài là bội số của 512.
- Khối dữ liệu được biểu diễn:
  - Bằng một dãy L khối 512 bit $Y_0, Y_1, \ldots, Y_{L-1}$
  - Bằng một dãy N từ (word) 32 bit $M_0, M_1, \ldots, M_{N-1}, N = L \times 16$

Giải thuật SHA-1: Nguyên lý

- **Bước 3: khởi tạo bộ đệm MD (MD buffer)**
  - Một bộ đệm 160 bit được dùng lưu trữ các giá
  - trị băm trung gian và kết quả.
  - Bộ đệm được biểu diễn bằng 5 thanh ghi 32 bit với các giá trị khởi tạo ở dạng big-endian (byte có trọng số lớn nhất trong từ nằm ở địa chỉ thấp nhất) như sau:
    - A = 01 23 45 67
    - B = 89 AB CD EF
    - C = FE DC BA 98
    - D = 76 54 32 10
    - E = C3 D2 E1 F0

Giải thuật SHA-1: Nguyên lý

❖ Bước 3: khởi tạo bộ đệm MD (MD buffer)

- ➢ Các giá trị này tương đương với các từ 32 bit sau:
  - A = 01 23 45 67
  - B = 89 AB CD EF
  - C = FE DC BA 98
  - D = 76 54 32 10
  - E = C3 D2 E1 F0

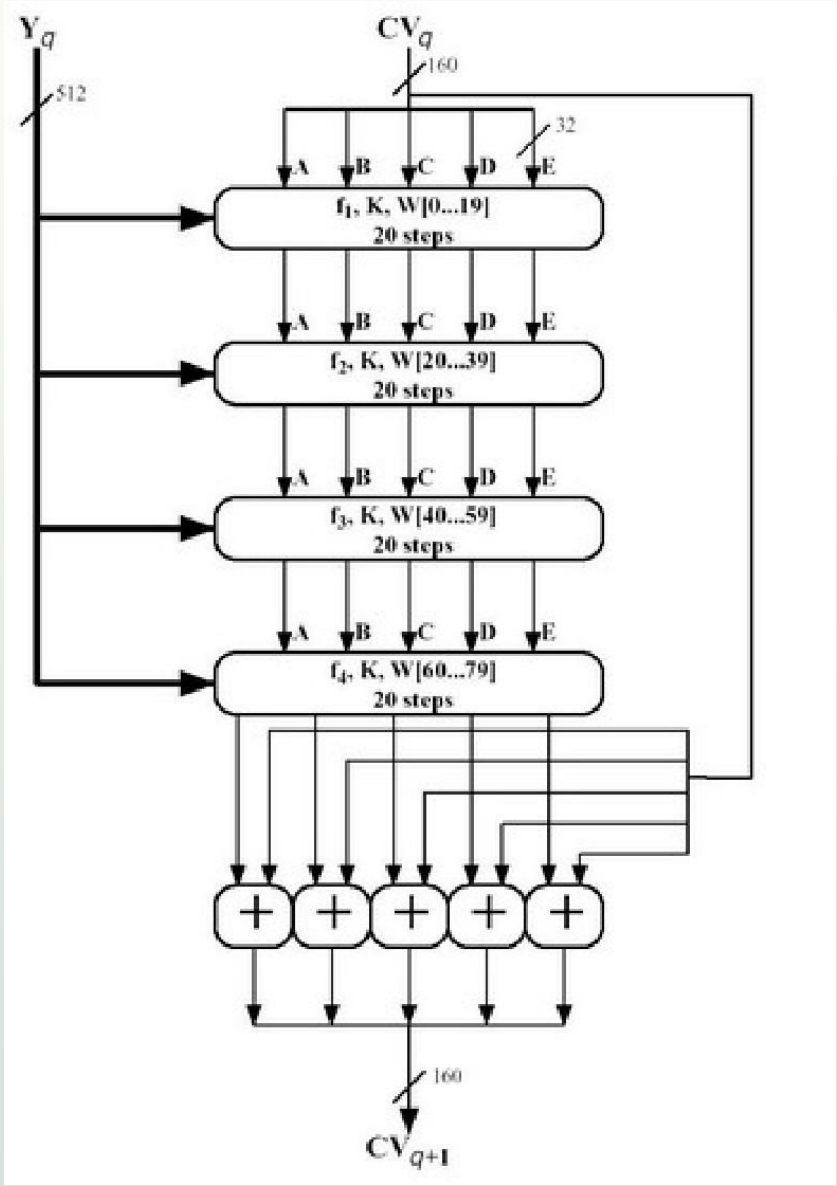
Giải thuật SHA-1: Nguyên lý

- **Bước 4: xử lý các khối dữ liệu 512 bit**
  - Gồm 4 vòng lặp thực hiện tất cả 80 bước
  - 4 vòng lặp có cấu trúc như sau, chỉ khác nhau ở các hàm logic f_1, f_2, f_3, f_4
  - Mỗi vòng có đầu vào gồm khối 512 bit và một bộ đệm 160 bit ABCDE.
  - Thao tác sẽ cập nhập giá trị bộ đệm

Giải thuật SHA-1: Nguyên lý

- Bước 4: xử lý các khối dữ liệu 512 bit
  - Mỗi bước sử dụng một hằng số $K_t(0-79)$
    - $K_t = 5A827999\ (0 \leq t \leq 19)$
    - $K_t = 6ED9EBA1(20 \leq t \leq 39)$
    - $K_t = 8F1BBCDC\ (40 \leq t \leq 59)$
    - $Kt = CA62C\ 1D6\ (60 \leq t \leq 79)$
  - Đầu ra của 4 vòng (bước 80) được … đầu ra của bước CV_q để tạo ra CV_{q+1}

Giải thuật SHA-1: Nguyên lý



Giải thuật SHA-1: Nguyên lý

❖ Bước 5: xuất kết quả

- ➢ Sau khi thao tác trên toàn bộ L khối. Kết quả của khối thứ L là bảng băm 160 bit
- ➢ Giải thuật được tóm tắt như sau
  - $CV_0 = IV$
  - $CV_{q+1} = SUM_{32}(CV_q , ABCDE_q)$
  - $MD = CV_L$

Giải thuật SHA-1: Nguyên lý

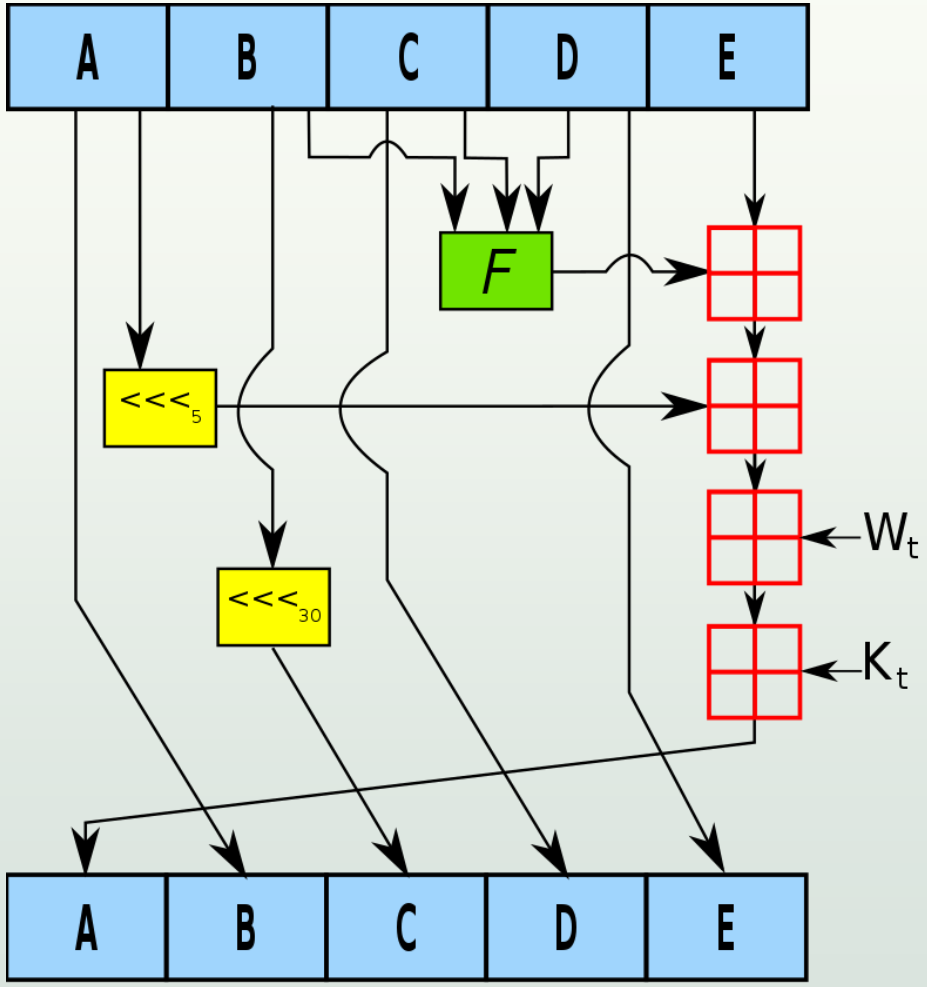
❖ Bước 5: xuất kết quả

- ➢ Với
  - IV = giá trị khởi tạo của bộ đệm ABCDE
  - $ABCDE_q$ = đầu ra của hàm nèn
  - L = số khối 512 bit của thông điệp
  - SUM_{32} = phép cộng modulo 2^{32} trên từng từ (32 bit) của đầu vào
  - MD = giá trị băm

Giải thuật SHA-1: Hàm nén

- Giải thuật thực hiện tất cả 80 bước, mỗi bước được mô tả như sau:
  - $A \leftarrow E + f(t, B, C, D) + S^5(A) + W_t + K_t$
  - $B \leftarrow A$
  - $C \leftarrow S^{30}(B)$
  - $D \leftarrow C$
  - $E \leftarrow D$
- Trong đó:
  - A, B, C, D, E = các từ trong bộ đệm
  - t = số thứ tự của bước
  - $f(t, B, C, D)$ = làm logic tại bước t
  - S_k = dịch vòng trái k bit
  - W_t = từ thứ t của khối dữ liệu
  - K_t = hằng số
  - + = phép cộng modulo 2^{32}

Giải thuật SHA-1: Hàm nén



Giải thuật SHA-1: Hàm nén

Bước	Hàm *f*	Giá trị
$0 \le t \le 19$	$f_1 = f(t, B, C, D)$	$(B \wedge C) \vee (\neg B \wedge D)$
$20 \le t \le 39$	$f_2 = f(t, B, C, D)$	$B \ xor \ C \ xor \ D$
$40 \le t \le 59$	$f_3 = f(t, B, C, D)$	$(B \wedge C) \vee (B \wedge D) \vee (C \wedge D)$
$60 \le t \le 79$	$f_4 = f(t, B, C, D)$	$B \ xor \ C \ xor \ D$

- Từ 16 từ (32 bit từ) khối dữ liệu đầu vào, mở rộngthành 80 từ Wt
  - Với $0 \le t \le 15$, giá trị W_t lấy trực tiếp từ khối dữ liệu
  - Với $t > 15$: $W_t = S^1(W_{t-16} \ xor \ W_{t-14} \ xor \ W_{t-8} \ xor \ W_{t-3})$

So sánh MD5 và SHA-1

- Khả năng chống lại tấn công brute-force:
  - Để tạo ra thông điệp có giá trị băm cho trước, cần 2^{128} thao tác với MD5 và 2^{160} với SHA-1
  - Để tìm 2 thông điệp có cùng giá trị băm, cần 2^{64} thao tác với MD5 và 2^{80} với SHA-1
- Khả năng chống lại thám mã (cryptanalysis): cả 2 đều có cấu trúc tốt

So sánh MD5 và SHA-1

- Tốc độ
  - Cả hai dựa trên phép toán 32 bit, thực hiện tốt trên các kiến trúc 32 bit
  - SHA-1 thực hiện nhiều hơn 16 bước và thao tác trên thanh ghi 160 bit nên tốt độ thực hiện chậm hơn
- Tính đơn giản: cả hai đều được mô tả đơn giản và dễ dàng cài đặt trên phần cứng và phần mềm

Hàm băm - Ứng dụng

- Key Stretching (tạo khóa bí mật từ mật khẩu)
- Integrity checking (kiểm tra tính toàn vẹn dữ liệu)
- HMAC - Hashed Message Authentication Code (mã chứng thực thông điệp sử dụng hàm băm)
- Chữ ký điện tử

Mật mã hóa dựa trên mật khẩu (PBE)

- Khóa của DES:
  - Chiều dài 56 bit (trong thực tế cài đặt cần 64 bit)
  - Phức tạp, khó nhớ

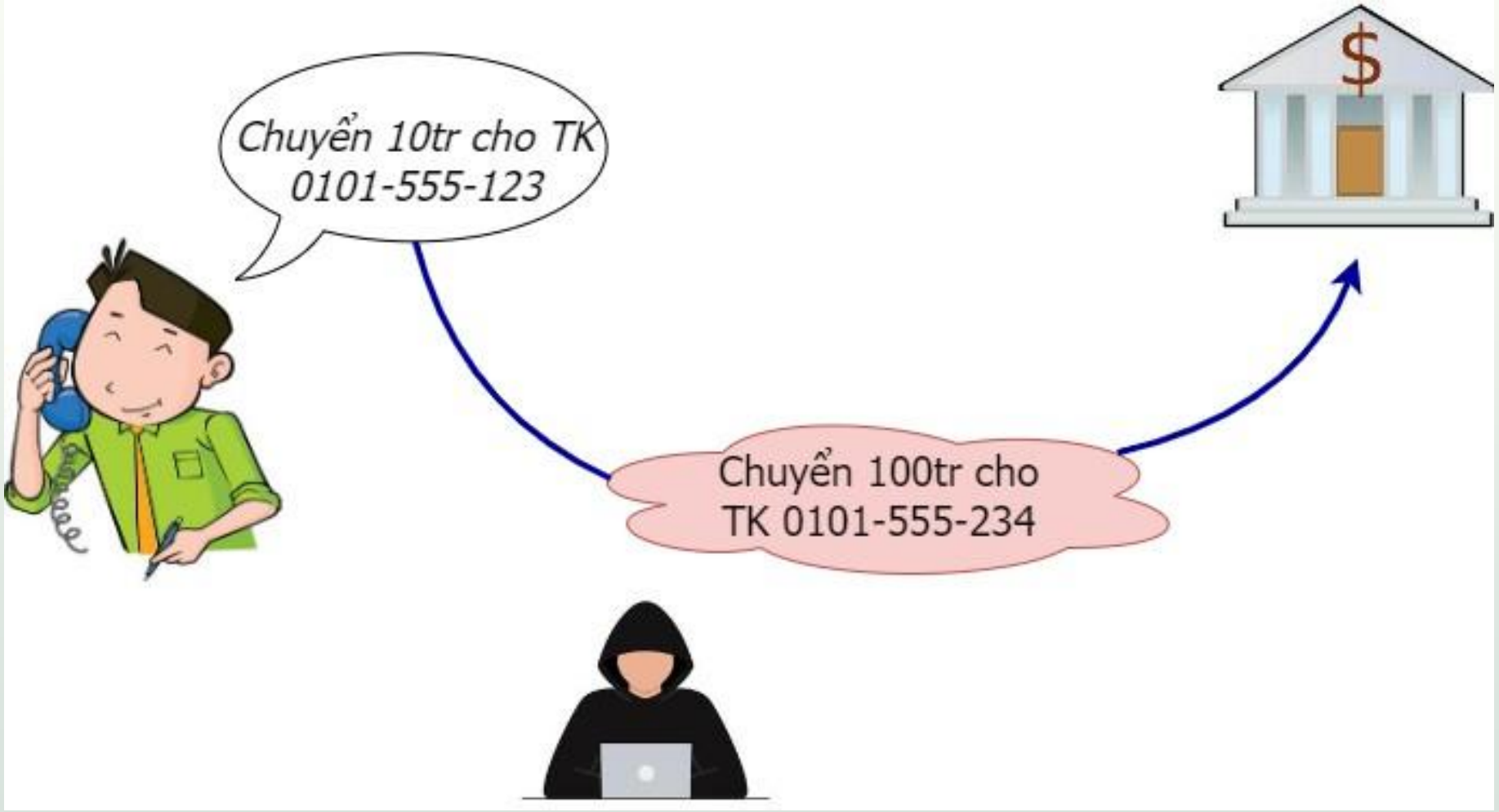
→ *Sử dụng mật khẩu (password)*

  - Chiều dài thay đổi, có thể không đúng bằng 64 bit (hay 8 ký tự)

→ *Mật mã hóa dựa trên mật khẩu*

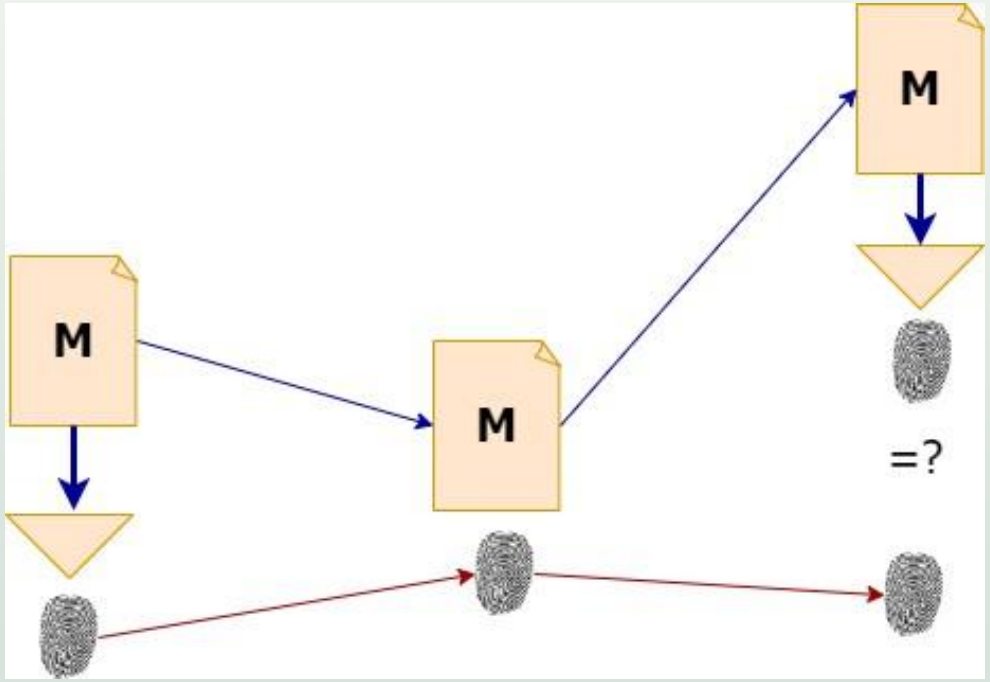
  - Băm mật khẩu có kích thước bất kỳ thành khóa có đúng 64 bit

Toàn vẹn dữ liệu (Integrity)

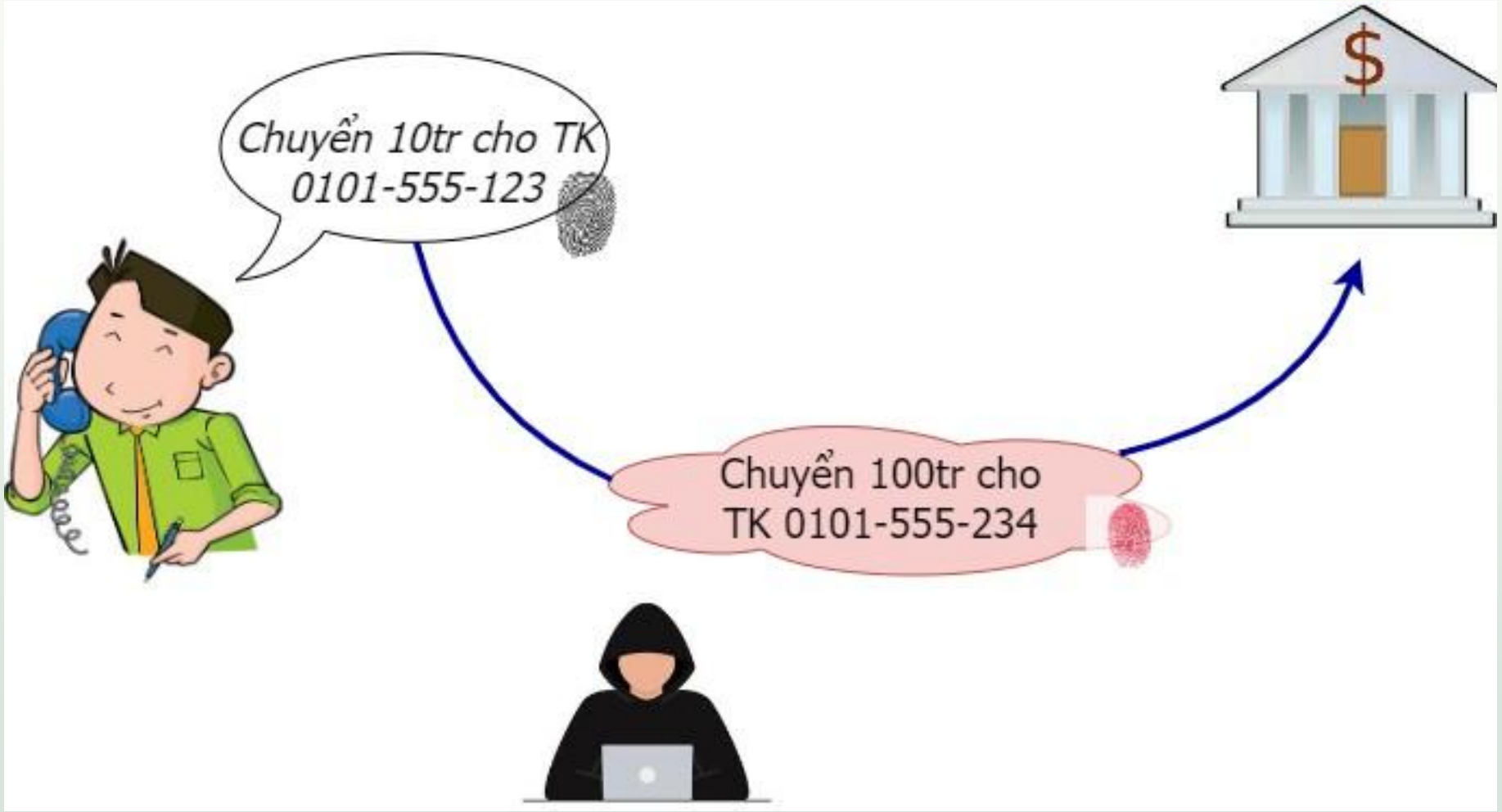


Toàn vẹn dữ liệu (Integrity)

- Gửi: gửi đính kèm theo thông điệp một bản băm của nó
- Nhận: băm thông điệp và so sánh với bản băm đi kèm

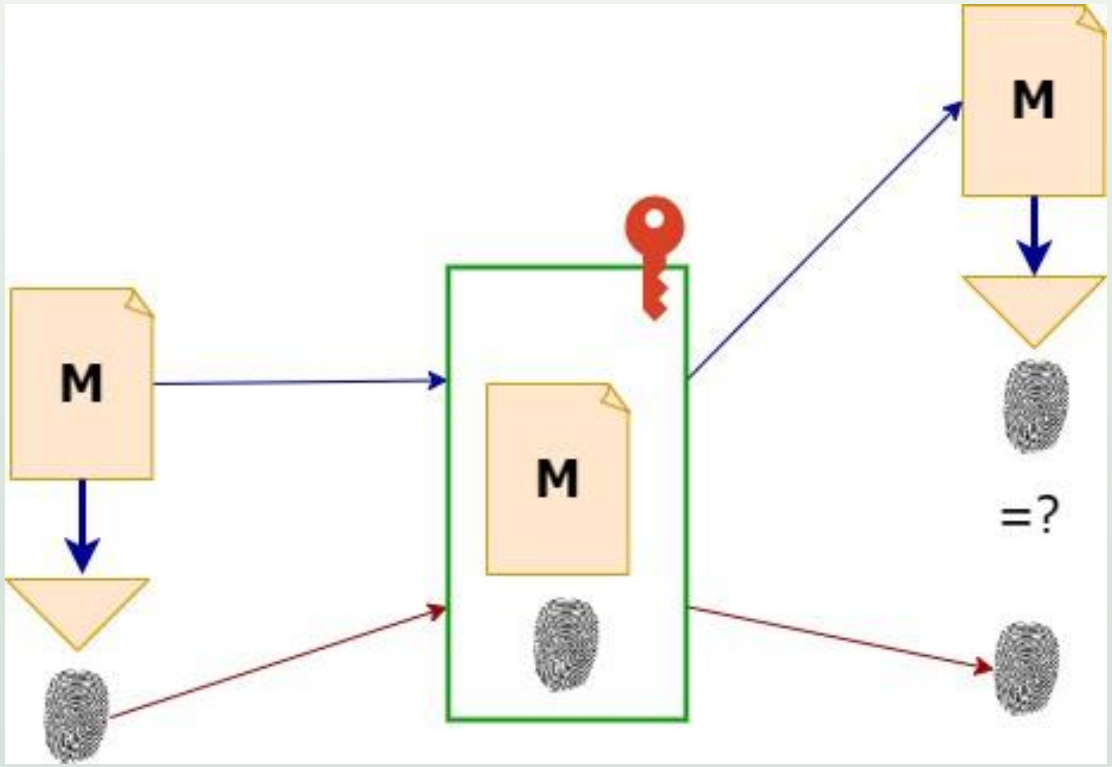


Toàn vẹn dữ liệu (Integrity)

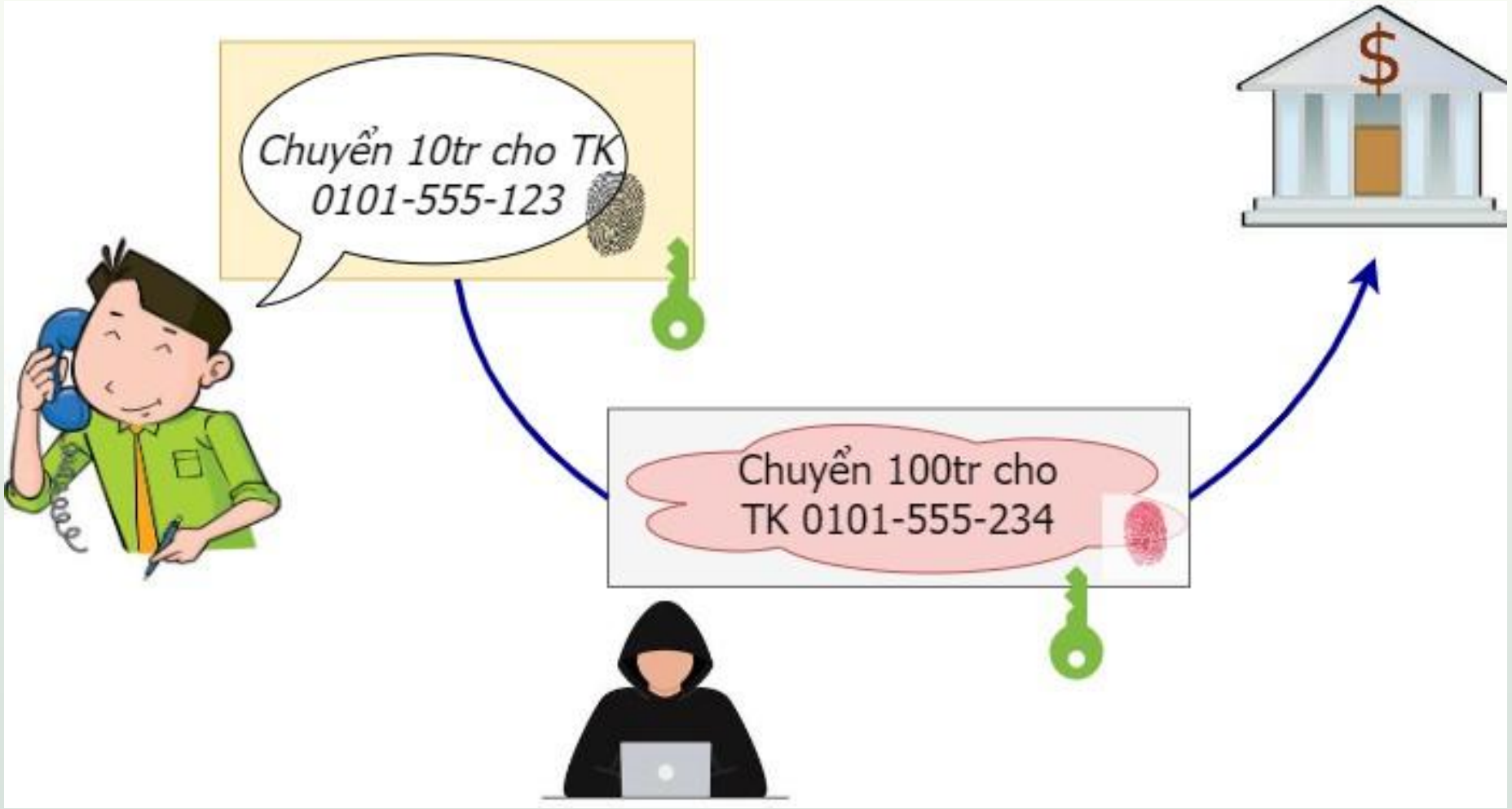


Toàn vẹn dữ liệu (Integrity)

- Gửi đính kèm theo thông điệp một bản băm của nó
- Mật mã hóa cả thông điệp và bản băm bằng khóa công khai của người nhận



Toàn vẹn dữ liệu (Integrity)



Mã chứng thực thông điệp

- **Chứng thực thông điệp** (message authentication) là một **cơ chế** hoặc **dịch vụ** được sử dụng để xác minh tính toàn vẹn của thông điệp.
  - đảm bảo dữ liệu nhận được chính xác như được gửi (không chỉnh sửa, thêm, xóa hoặc thay thế)
  - đảm bảo danh tính người gửi là hợp lệ
- Mã hóa khóa đối xứng cung cấp tính năng chứng thực giữa những người chia sẻ khóa bí mật.

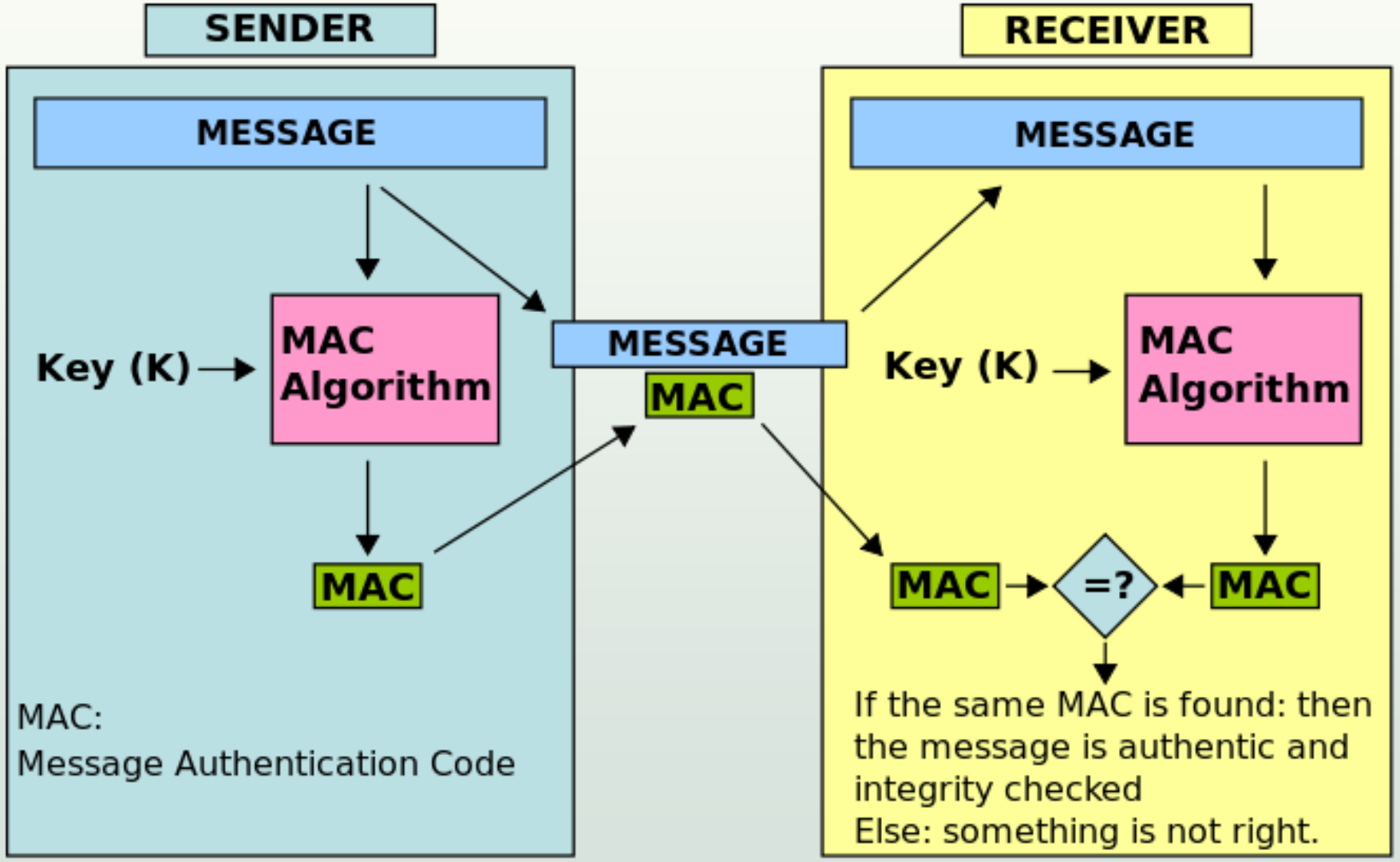
Mã chứng thực thông điệp

- **Mã chứng thực thông điệp** (Message Authentication Code **MAC**) là giải thuật yêu cầu sử dụng khóa bí mật
- **Đầu vào**: thông điệp chiều dài khác nhau và khóa bí mật; **Đầu ra**: mã chứng thực
- Người sở hữu khóa bí mật có thể tạo các mã chứng thực để xác minh tính toàn vẹn của thông điệp

Mã chứng thực thông điệp

- kết hợp hàm băm mật mã với một khóa bí mật
- sử dụng mã hóa đối xứng để tạo ra output có chiều dài cố định cho một input chiều dài thay đổi

Mã chứng thực thông điệp



Chữ ký điện tử

- Chữ ký điện tử (Electronic signature hay e-signature)
  - một biểu tượng điện tử được gắn vào tài liệu dưới dạng điện tử và được sử dụng bởi người ký để ký tên
  - được ứng dụng nhằm đảm bảo an toàn trong thương mai điện tử (e-commerce) và quản trị điện tử (e-governance)

Chữ ký điện tử

- Một số vấn đề cần đánh giá:
  - **Vấn đề ký một tài liệu**
    - với chữ ký thông thường thì nó là một phần vật lý của tài liệu
  - **Vấn đề kiểm tra**
    - chữ ký thông thường được kiểm tra bằng cách so sánh nó với các chữ ký xác thực khác

Chữ ký điện tử: Tính chất

- Tính chống thoái thác
- Tính toàn vẹn
- Xác thực

Chữ ký điện tử: Phân loại

- Gồm các nhóm chính
  - Digital Signature (chữ ký số): dựa vào mã hóa khóa công khai
  - E-sign: không sử dụng PKI mà dựa vào định danh và logs
  - Bio-metric signature (chữ ký sinh trắc): dựa vào đặc điểm cá nhân đặc biệt . . .

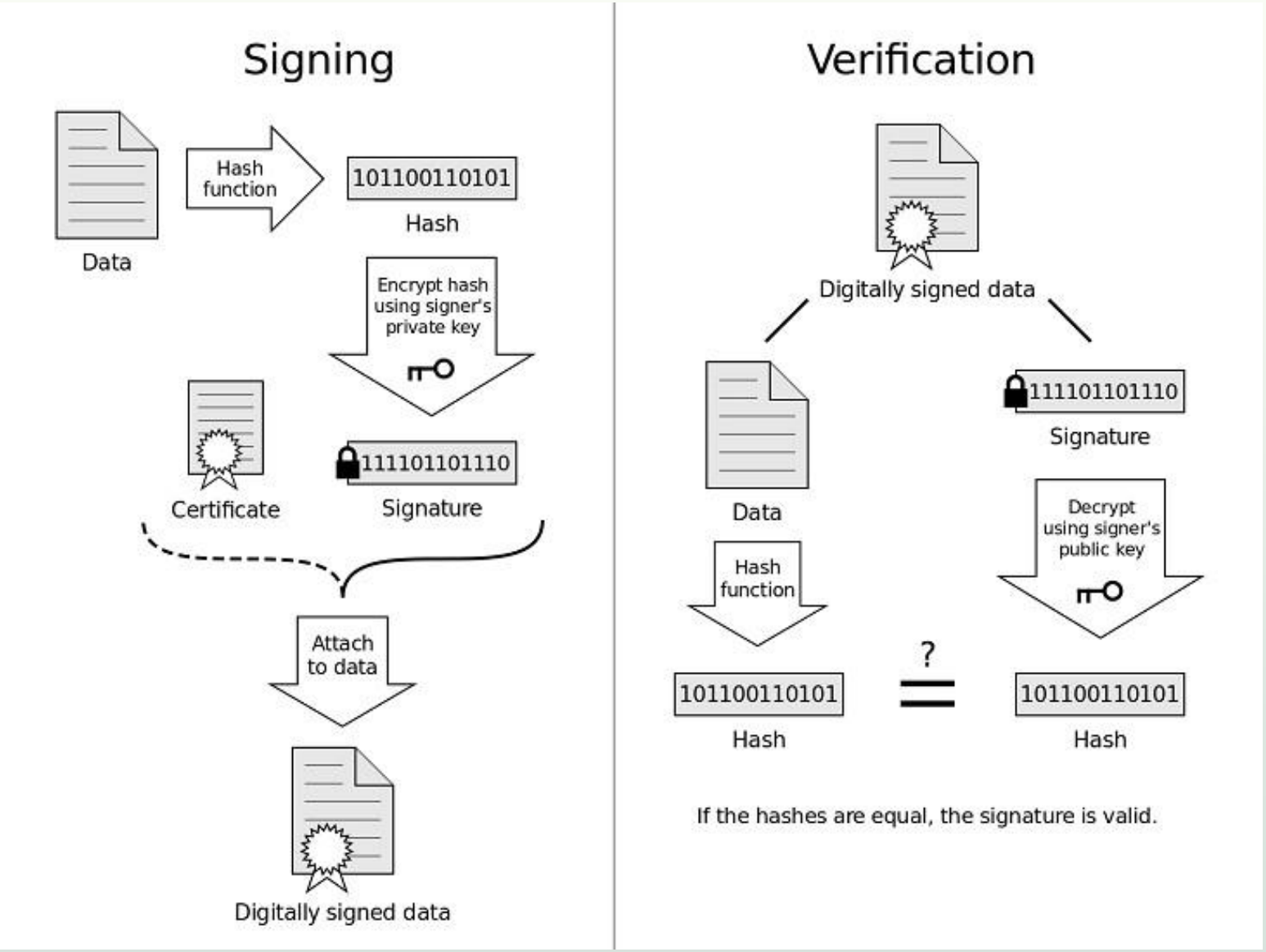
Chữ ký số

- là một dạng của chữ ký điện tử
- dựa trên công nghệ mã hóa khóa công khai cung cấp các dịch vụ: xác thực, toàn vẹn và chống thoái thác
- khóa công khai thường được phân phối thông qua chứng thực khóa công khai

Chữ ký số

- Sơ đồ chữ ký số: 03 giai đoạn chính
  - Tạo bộ khóa: KeyGeneration()
  - Tạo ra chữ ký: Sign()
  - Kiểm tra / xác minh công khai chữ ký: Verified()

Chữ ký số


Signing
Data
Hash function
101100110101
Hash
Encrypt hash using signer's private key
Certificate
111101101110
Signature
Attach to data
Digitally signed data
Verification
Digitally signed data
Data
111101101110
Signature
Decrypt using signer's public key
Hash function
?
101100110101
Hash
101100110101
Hash
If the hashes are equal, the signature is valid.

Chữ ký số

- Sự quan trọng của mô hình này dựa vào:
  - Chữ ký sinh ra được xác thực bằng cách sử dụng khóa công khai
  - Không người nào có thể sinh chữ ký hợp lệ mà không có khóa riêng tư (private key) đúng
- Khác với chữ ký thông thường:
  - thường không xuất hiện trong văn bản
  - kiểm tra chữ ký bằng giải thuật rất khó giả mạo

Chữ ký số RSA

- tương tự như hệ mật mã RSA, vai trò của 2 khóa bị thay đổi
- khóa riêng sử dụng để ký, khóa công khai của người gửi sử dụng để xác minh chữ ký

Chữ ký số RSA: sơ đồ

- Theo 3 giai đoạn:
  - **1. Tạo bộ khóa**: Server tạo ra các tham số liên quan từ đó tính được

$$K: PU = \{e, n\}; PR = \{d, n\}$$

  - **2. Thuật toán sinh chữ ký số**
    - Input: giá trị băm của thông điệp h(M)
    - Output: chữ ký dựa vào khóa riêng sử dụng mã hóa

$$S = h(M)^d \bmod n$$

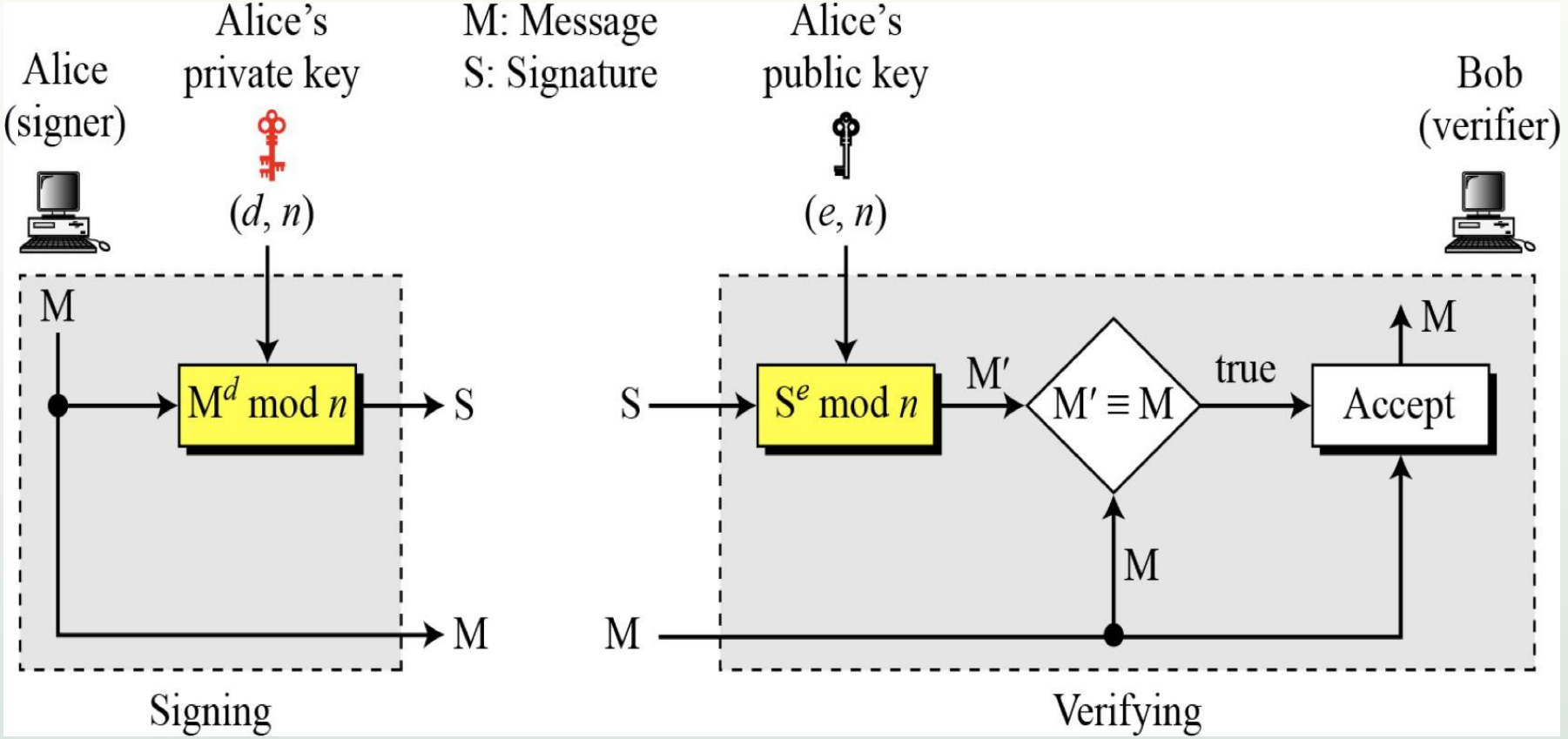
Chữ ký số RSA: sơ đồ

3. Thuật toán xác minh chữ ký

- Input: thông điệp gốc M + thông điệp đã ký S + PU của người gửi
- Output: kết quả xác minh "TRUE" hoặc "FALSE"

$$M' = S^e mod\ n \Rightarrow M' = h(M)? \Rightarrow True\ or\ False$$

Chữ ký số RSA: sơ đồ



Chữ ký số Elgamal: Sơ đồ

❖ Tạo bộ khóa:

- ➢ Bộ khóa $K = \{PU, PR\}$

  với $PU = \{p, \varepsilon, y\}$ và $PR = \{a\}$

❖ Tạo chữ ký trên văn bản M

- ➢ Tính $m = h(M), 0 \leq m \leq p - 1$
- ➢ Chọn số nguyên k sao cho:

  $1 \leq k \leq p - 1$ và $UCLN(k, p - 1) = 1$
- ➢ Tính khóa:

  $S1 = \varepsilon^k (mod\ p)$

  $S2 = (m - a * S1)k^{-1}\ mod\ (p - 1)$

  $\Rightarrow$ Chữ ký số gửi đi $(S1, S2)$

Chữ ký số Elgamal: Sơ đồ

➢ 3. Thuật toán xác minh chữ ký

$$v1 = \varepsilon^m \ mod \ p$$

$$v2 = \ y^{S1} S1^{S2} \ mod \ p$$

$$\Rightarrow \ v1 \ = \ v2?$$


Thank You