Trần Minh Tiến mssv:1821050398

# I . Mở rộng

# CÁC THUẬT TOÁN TẠO HÀM BĂM

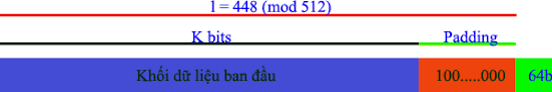
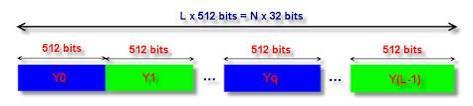
## Hàm băm MD5

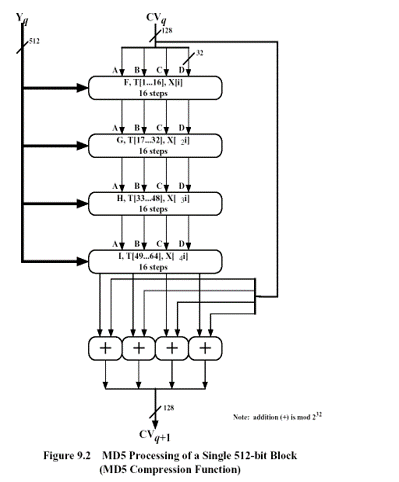
* Phát triển bởi Ron Rivest tại đại học MIT
* Input: thông điệp với độ dài bất kỳ
* Output: giá trị băm (message digest) 128 bits
* Giải thuật gồm 5 bước thao tác trên khối 512 bits

Ảnh có chứa văn bản, tờ báo, tài liệu

Mô tả được tạo tự động

## GIẢI THUẬT MD5 - NGUYÊN LÝ

* Bước 1: Nhồi dữ liệu
  + Nhồi thêm các bits sao cho dữ liệu có độ dài ***l ≡ 448 mod 512*** hay ***l = n \* 512 + 448 (n,l nguyên)***
  + Luôn thực hiện nhồi dữ liệu ngay cả khi dữ liệu ban đầu c ó độ d ài mong muốn. Ví dụ, dữ liệu c ó độ d ài 448 được nhồi thêm 512 bits để được độ d ài 960 bits.
  + Số lượng bit nhồi thêm nằm trong khoảng 1 đến 512
  + Các bit được nhồi gồm 1 bit “ 1 ” v à c ác bit 0 theo sau.
* 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 bits thấp được sử dụng, nghĩa là giá trị được thêm vào bằng **K mod 264**
  + 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 Y0, Y1,…, YL-1
    - Bằng một dãy N từ (word) 32-bit M0, M1, MN-1. Vậy **N = L x 16** (32 x 16 = 512)
* 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
  + 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
* Bước 4: Xử lý các khối dữ liệu 512- bit
  + Trọng tâm của giải thuật là hàm nén (compression function) gồm 4 “vòng” xử lý. Các vòng này có cấu trúc giống nhau nhưng sử dụng các hàm luận lý khác nhau gồm F, G, H và I
    - F(X,Y,Z) = X ˄ Y ˅ ̚ X ˄ Z
    - G(X,Y,Z) = X ˄ Z ˅ Y ˄ ̚ Z
    - H(X,Y,Z) = X xor Y xor Z
    - I(X,Y,Z) = Y xor (X ˅ ̚ Z)
  + Mảng 64 phần tử được tính theo công thức: T[i] = 232 x abs(sin(i)), i được tính theo radian.
  + Kết quả của 4 vòng được cộng (theo modulo 232 với đầu vào CVq để tạo CVq+1

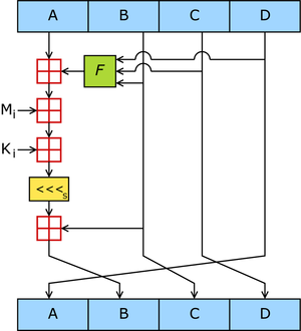


* Các giá trị trong bảng T

Ảnh có chứa bàn

Mô tả được tạo tự động

* + 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 được tóm tắt như sau:
    - CV0 = IV
    - CVq+1 = SUM32[CVq,RFI(Yq,RFH(Yq,RFG(Yq,RFF(Yq,CVq))))]
    - MD = CVL-1
    - Với các tham số
    - IV: bộ đệm gồm 4 thanh ghi ABCD
    - Yq: khối dữ liệu thứ q gồm 512 bits
    - L: số khối 512-bit sau khi nhồi dữ liệu
    - CVq: đầu ra của khối thứ q sau khi áp dụng hàm nén
    - RFx: hàm luận lý sử dụng trong các “vòng” (F,G,H,I)
    - MD: message digest – giá trị băm
    - SUM32: cộng modulo 232
  + 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
    - A <- B + ((A + F(B,C,D) + X[k] + T[i]) <<< s)
    - A,B,C,D: các từ của thanh ghi
    - F: một trong các hàm F,G,H,I
    - <<< s : dịch vòng trái s bits
    - Mi ~ X[k]: từ 32-bit thứ k của khối dữ liệu 512 bits.k=1..15
    - Ki ~ T[i]: giá trị thứ i trong bảng T.
    - +: phép toán cộng modulo 232



## Giới thiệu thuật toán hàm băm SHA-1

### Giới thiệu hàm băm SHA-1

Năm 1990, Ron Rivest đã sáng tạo ra hàm băm MD4. Sau đó năm 1992, ông cải tiến MD4 và phát triển một hàm băm khác: MD5. Năm 1993, Cơ quan An ninh Quốc gia Hoa Kỳ/Cục An ninh Trung ương (NSA) đã công bố,một hàm băm rất giống với MD5 được gọi là SHA. Vào năm 1995, sau việc khắc phục những lỗ hổng kỹ thuật, NSA đã thay đổi SHA trở thành một hàm băm mật mã khác gọi là SHA-1.

SHA-1 (Sercue Hash Algorithm) là thuật toán cũng được xây dựng trên thuật toán MD4, đang được sử dụng rộng rãi. Thuật toán SHA-1 tạo ra chuỗi mã băm có chiều dài cố định 160 bit từ chuỗi bit dữ liệu đầu vào x có chiều dài tùy ý.

### Thuật toán SHA-1

I**nput**: thông điệp với độ dài tối đa 2^64 bits

**Output**: thông điệp rút gọn (message digest) có độ dài 160 bits

Giải thuật gồm 5 bước trên khối 512 bits

***Bước 1: Nhồi dữ liệu***

* Thông điệp được nhồi thêm các bit sao cho độ dài L mod 512 luôn đồng dư là 448.
* Thông điệp luôn luôn được nhồi thêm các bit.
* Số bit nhồi thêm phải nằm trong khoảng 1-512.
* Phần thêm vào cuối dữ liệu gồm 1 bit 1 và theo sau là các bit 0.

***Bước 2: Thêm độ dài****:*

* Độ dài khối dữ liệu ban đầu sẽ được biểu diễn dưới dạng nhị phân 64 bit và được thêm cuối chuỗi nhị phân mà ta thu được ở bước 1.
* Độ dài được biểu diễn dưới dạng nhị phân 64 bit không dấu
* Kết quả thu được từ 2 bước là một khối dữ liệu có độ dài là bội số của 512. ( Với cứ 512 bit là một khối dữ liệu)

***Bước 3: Khởi tạo bộ đệm MD ( MD buffer)***

Một bộ đệm 160 bit được dùng để lưu trữ các giá trị băm trung gian và kết quả. Bộđệm được biểu diễn bằng 5 thanh ghi 32-bit với các giá trị khởi tạo ở dạng big-endian (buyte có trọng số lớn nhất trong từ nằm ở địa chỉ thấp nhất) và có 2 bộ đệm. 5 thanh ghi của bộ đệm đầu tiên được đánh đặt tên là A, B,C,D,E và tương tự cho bộ đệm thứ 2 là. Có giá trị như sau ( Theo dạng Hex):

=67452301

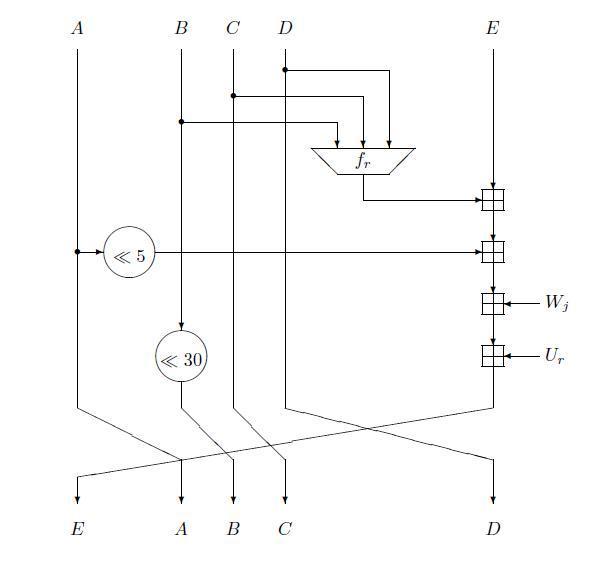
=EFCDAB89

= 98BADCFE

= 10325476

= C3D2E1F0

***Bước 4: Xử lý các khối dữ liệu 512 bit***



***Thuật toán SHA-1***

* Trọng tâm của giải thuật bao gồm 4 vòng lặp thực hiện tất cả 80 bước.
* 4 vòng lặp có cấu trúc như nhau, chỉ khác nhau ở hàm logic .

|  |  |  |
| --- | --- | --- |
| Bước | Hàm | Gía trị |
| (0≤ t ≤ 19) | =F(B, C,D) | (B AND C) OR ( (NOT B) AND D) |
| (20≤ t ≤ 39) | =F(B, C,D) | B XOR C XOR D |
| (40≤ t ≤ 59) | =F(B, C,D) | (B AND C) OR (B AND D) OR (C AND D) |
| (60≤ t ≤ 79) | =F(B, C,D) | B XOR C XOR D |

* Mỗi vòng có đầu vào gồm khối 512-bit hiện thời và một bộ điệm 160 bit A, C, B, D, E. Các thao tác sẽ cập nhật giá trị bộ đệm.
* Chia khối dữ liệu đã nhồi thêm (cuối bước 2) thành 16 nhóm (mỗi nhóm gồm 32 bit) và đặt theo thứ tự là: ,,....
* Mở rộng từ 16 nhóm 32bit lên đến 80 nhóm 32 bit bằng vòng lặp

For 16 to 79 let

= ( XOR XOR XOR )

* Gán A=, B=, C=, D=, E=.
* Mỗi vòng lặp sử dụng theo công thức chung với 1 hằng số = (0≤ t ≤ 79) như sau:

For t= 0 to 79 do

TEMP= (A)+(B,C,D)+ E + +

E=D; D=C; C= (B); B=A; A= TEMP

Với:

= 5A827999 (0≤ t ≤ 19)

= 6ED9EBA1(20≤ t ≤ 39)

= 8F1BBCDC(40≤ t ≤ 59)

= CA62C1D6(60≤ t ≤ 79) .

* Đầu ra của 4 vòng (bước 80) được cộng với giá trị của bộ đệm để tạo ra 1 chuỗi kết quả dài 160 bit.

= + A

= + E  
**Bước 5: Xuất kết quả**

* Sau khi thao tác trên toàn bộ N khối dữ liệu (blocks). Kết quả của khối thứ N là chuỗi băm 160 bit.

H=

Một số loại hàm băm khác

*RIPEMD-160*

RIPEMD (viết tắt của RACE Integrity Primitives Evaluation Message Digest) là họ hàm băm được phát triển tại Leuven, Bỉ, bởi ba nhà mật mã học Hans Dobbertin, Antoon Bosselaers và Bart Preneel của nhóm nghiên cứu COSIC thuộc đại học Katholieke Universiteit Leuven.  RIPEMD lần đầu tiên được công bố vào năm 1996 dựa trên các nguyên tắc thiết kế được sử dụng trong MD4. RIPEMD-160 tạo ra một bản tóm tắt gồm 160 bit (20 byte). RIPEMD có hiệu năng tương tự như SHA-1 nhưng ít được phổ biến hơn. Và cho đến nay RIPEMD-160 chưa bị phá vỡ.

*Whirlpool*

Whirlpool là một hàm băm mật mã được thiết kế bởi Vincent Rijmen và Paulo S. L. M. Barreto. Nó được mô tả đầu tiên vào năm 2000. Whirlpool là một trong hai hàm băm đã được phê duyệt trong dự án NESSIE - một dự án của Liên minh Châu Âu nhằm xác định tiêu chuẩn an toàn của mật mã nguyên thủy. Whirlpool cũng đã được công bố trong tiêu chuẩn quốc tế ISO / IEC 10118-3. Whirlpool dựa trên phiên bản sửa đổi đáng kể của Tiêu chuẩn mã hóa nâng cao (AES). Whirlpool tạo ra một bản tóm tắt có độ dài 512 bit (64 byte) của dữ liệu.

*SHA-2*

SHA-2 là một tập hợp các hàm băm mật mã được thiết kế bởi Cơ quan an ninh quốc gia Hoa Kỳ (NSA), được xuất bản lần đầu tiên vào năm 2001. Chúng được xây dựng bằng cấu trúc Merkle–Damgård, chức năng nén một chiều của nó được xây dựng bằng cấu trúc Davies–Meyer từ một hệ mật mã khối chuyên dụng.

SHA-2 về thực chất bao gồm hai thuật toán băm: SHA-256 và SHA-512. SHA-224 là một biến thể của SHA-256 với các giá trị khởi tạo và đầu ra bị cắt bỏ khác nhau. SHA-384 và SHA-512/224 và SHA-512/256 ít được biết đến là tất cả các biến thể của SHA-512. SHA-512 an toàn hơn SHA-256 và thường nhanh hơn SHA-256 trên các máy 64 bit như AMD64.

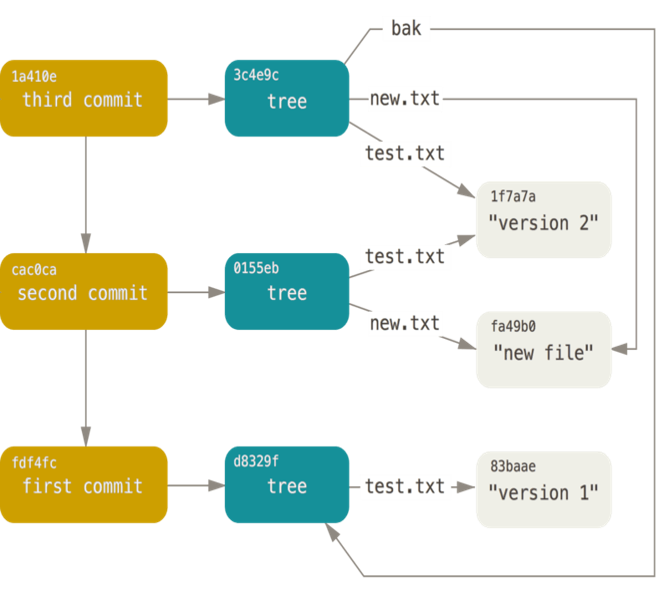
Do có nhiều phiên bản thuật toán khác nhau do đó kích thước đầu ra của họ SHA-2 cũng khác nhau tùy theo thuật toán. Phần mở rộng của tên phía sau tiền tố “SHA” chính là độ dài của thông điệp băm đầu ra. Ví dụ với SHA-224 thì kích thước đầu ra là 224 bit (28 byte), SHA-256 tạo ra 32 byte, SHA-384 tạo ra 48 byte và cuối cùng là SHA- 512 tạo ra 64 byte. Và chúng ta có thể đã biết rằng Bitcoin sử dụng hàm băm SHA-256 là một phiên bản trong họ SHA-2 này.

*SHA-3*

SHA-3 được NIST phát hành vào ngày 5 tháng 8 năm 2015. Đây có lẽ là tiêu chuẩn hàm băm mới nhất cho đến hiện nay. SHA-3 là một tập con của họ nguyên thủy mật mã rộng hơn là Keccak. Thuật toán Keccak được đưa ra bởi Guido Bertoni, Joan Daemen, Michael Peeters và Gilles Van Assche. Keccak dựa trên cấu trúc bọt biển (sponge). Cấu trúc này cũng có thể được sử dụng để xây dựng các nguyên thủy mã hóa khác như các hệ mật mã dòng. SHA-3 cũng có các kích cỡ đầu ra tương tự như SHA-2 bao gồm: 224, 256, 384 và 512 bit.

# ỨNG DỤNG

## Hashing trong định danh tệp hoặc dữ liệu

Giá trị băm có thể được sử dung như một phương tiện để định danh tập tin một cách đáng tin cậy. Một số hệ thống quản lý mã nguồn, như Git, Mercurial hay Monotone, sử dụng giá trị sha1sum của nội dung tệp, cây thư mục, thông tin thư mục gốc, v.v. để định danh chúng.

Giá trị băm cũng được sử dụng để xác định các tệp trên các mạng chia sẻ tệp ngang hang nhằm cung cấp đầy đủ thông tin để định vị nguồn gốc của tệp, xác minh nội dung tệp tải xuống

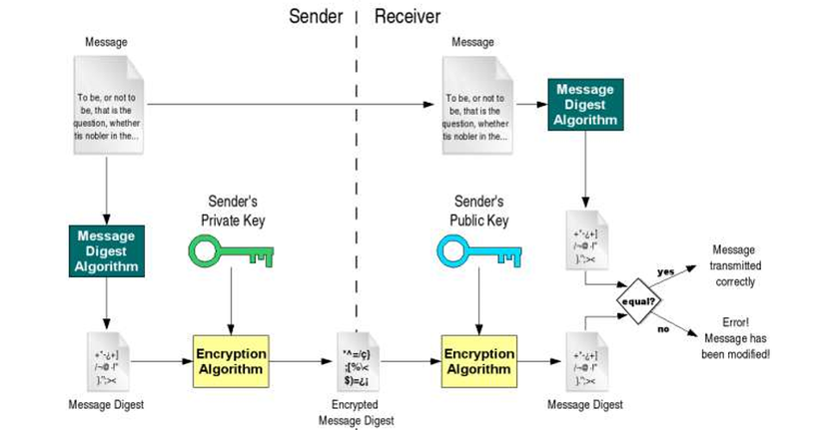
## Hashing trong xác minh tính toàn vẹn của thông điệp hoặc tập tinẢnh có chứa văn bản Mô tả được tạo tự động

Khi download một phần mềm hoặc tệp tin nào đó trên một số trang web, ta được cung cấp kèm theo các mã băm MD5 hoặc SHA.  
Khi đó sau khi tải về tệp tin, ta có thể tính và so sánh giá trị băm của tệp tin tải về với giá trị băm được cung cấp trên web, nếu có sự sai khác tức là tệp tin tải về đã bị sửa đổi.

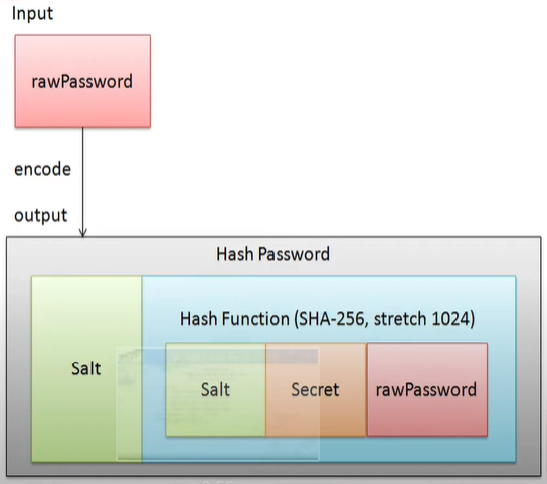
## Hashing trong tạo và xác nhận chữ ký

Hầu như tất cả các lược đồ chữ ký số đều yêu cầu tính toán bản tóm lược của thông điệp bằng các hàm băm mật mã.

Cho phép việc tính toán và tạo chữ ký được thực hiện trên một khối dữ liệu có kích thước tương đối nhỏ và cố định thay vì trên toàn bộ văn bản dài.

Tính chất toàn vẹn thông điệp của hàm băm mật mã được sử dụng để tạo các lược đồ chữ ký an toàn và hiệu quả.

## Hashing trong xác minh mật khẩu

Có thể lưu trữ mật khẩu bằng giá trị băm của nó để tang tính bảo mật. Để kiểm tra, password được user đưa vào sẽ được hash và so sánh với giá trị băm đã được lưu.

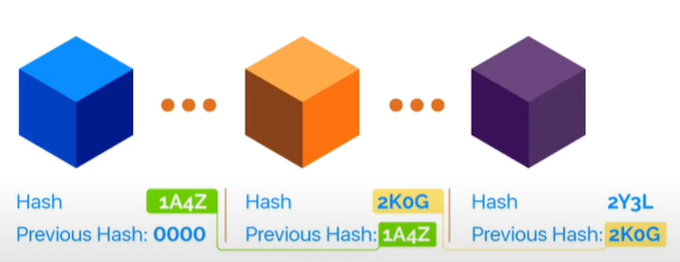
Đề phòng chống tấn công bằng brute force, ta có thể tang thời gian kiểm tra bằng cách sử dụng key stretching.

Ngoài ra ta có thể sử dụng them salt để tránh trường hợp: 2 password giống nhau có kết quả lưu trữ giống nhau.

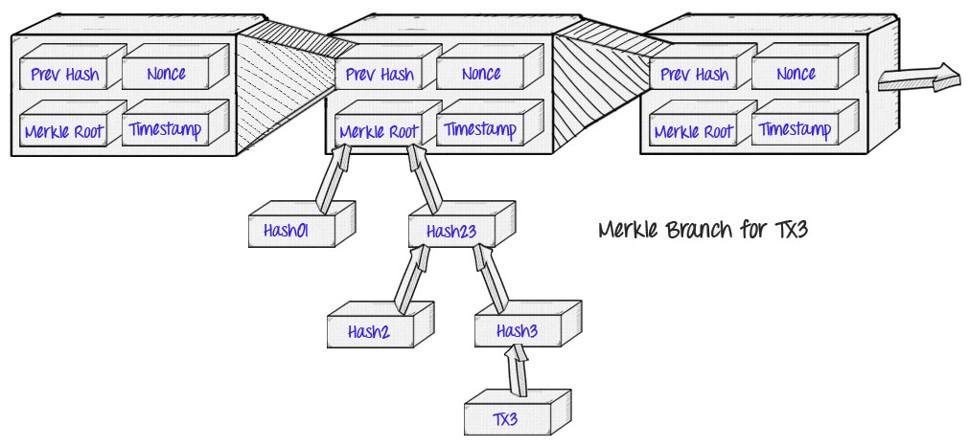
## Hashing trong blockchain

Với blockchain, ngoài việc có thể trỏ tới block trước đó, mỗi block còn có thể lưu giá trị digest của khối được trỏ tới. Thông qua việc kiểm tra giá trị hash có thể nhận dạng khối được trỏ tới có bị thay đổi hay không.

Cấu trúc blockchain cho phép chỉ cần lưu giá trị của con trỏ tới khối cuối cùng, đồng thời vẫn kiểm soát được nội dung của các khối còn lại không bị thay đổi.



Ví dụ, một hacker nào đó muốn thay đổi nội dung của dữ liệu của Blockchain ở một khối k trong danh sách; vì nội dung khối k này bị thay đổi, con trỏ hash của khối k+1 sẽ không còn đúng.  
Để không bị phát hiện, hacker phải tiếp tục thay đổi nội dung dữ liệu của khối k+1, nhưng chuyện này sẽ dẫn đến chuyện giá trị con trỏ hash ở khối k+2 không còn chính xác; tiếp tục như vậy sẽ dẫn đến việc thay đổi nội dung ở khối cuối cùng. Nhưng chúng ta lưu trữ giá trị của khối này, nên sẽ phát hiện ra sự thay đổi đó.



Kết luận

Có thể khẳng định rằng các hàm băm là một phần không thể thiếu trong các ứng dụng, sản phẩm mật mã. Tuy nhiên, với những tiến bộ của khoa học công nghệ tính an toàn (tồn tại va chạm) của nhiều hàm băm đã bị phá vỡ (MD5, SHA-1…). Các tổ chức về mật mã khác nhau trên thế giới cũng đã đưa ra các khuyến nghị về việc chọn thuật toán hàm băm an toàn.

Table

Description automatically generated

*ECRYPT-CSA Recommendations (2018)*

Vì vậy, khi sử dụng các sản phẩm mật mã dân sự, tùy thuộc vào yêu cầu bảo mật, an toàn người sử dụng cần lựa chọn những hàm băm phù hợp tránh tạo các lỗ hổng mất an toàn an ninh.

V

Tài liệu tham khảo : môn học các hệ thống thông tin thông minh

<https://www.google.com/search?q=m%E1%BB%9F+r%E1%BB%99ng+h%C3%A0m+b%C4%83m&oq=m%E1%BB%9F+r%E1%BB%99ng+h%C3%A0m+b%C4%83m+&aqs=chrome..69i57.4465j0j7&sourceid=chrome&ie=UTF-8>

<https://vi.wikipedia.org/wiki/H%C3%A0m_b%C4%83m>

<https://text.123docz.net/document/4612281-nghien-cuu-xay-dung-mot-lop-ham-bam-mo-rong-moi-va-kha-nang-ung-dung.htm>

https://nacis.gov.vn/nghien-cuu-trao-doi/-/view-content/253014/ham-bam