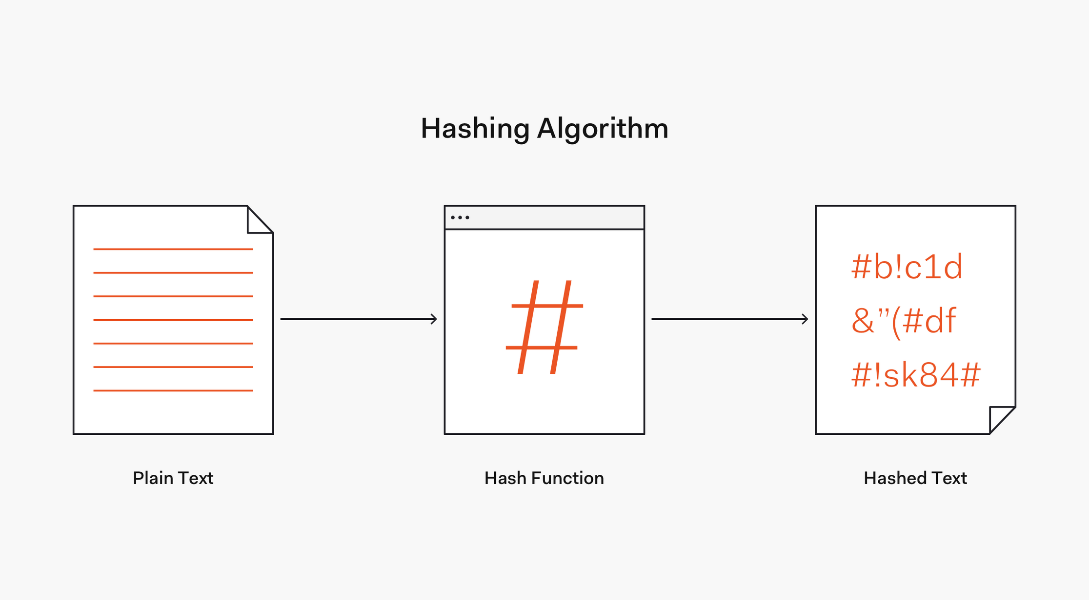
**Topic 1: What is Crypto Hash Function?**

**Hash Funtion là gì?**

Hàm băm mật mã học (tiếng Anh: Cryptographic hash function) là một hàm băm với một số tính chất bảo mật nhất định để phù hợp việc sử dụng trong nhiều ứng dụng bảo mật thông tin đa dạng, chẳng hạn như chứng thực (authentication) và kiểm tra tính nguyên vẹn của thông điệp (message integrity). Một hàm băm nhận đầu vào là một xâu ký tự dài (hay thông điệp) có độ dài tùy ý và tạo ra kết quả là một xâu ký tự có độ dài cố định, đôi khi được gọi là tóm tắt thông điệp (message digest) hoặc chữ ký số (digital fingerprint).



**Và để 1 hàm hash được coi là 1 hàm cryptographic hash, nó cần đạt ít nhất 5 tiêu chí sau đây:  
Deterministic(xác định)**

**Vd: https://tools.superdatascience.com/blockchain/hash**

Nó có nghĩa là khi bạn đưa 1 dữ liệu đầu vào đi qua hàm hash, bạn sẽ luôn luôn nhận về 1 chuổi đầu ra giống nhau



Điều này rất quan trọng vì nếu với mỗi lần chạy hàm hash ta nhận được 1 output khác nhau thì sẽ là bất khả thi để ta có thể theo dõi giá trị đầu vào

### Quick Computation(tính toán nhanh)

Hàm hash phải có khả năng trả về kết quả với tốc độ nhanh nhất có thể. Bởi vì trong thực tế, dữ liệu đầu vào không phải lúc nào cũng là một chuỗi string, 1 giá trị int mà có thể là 1 file ảnh rất lớn và ta cần chạy hàm hash cho mỗi dữ liệu ta đưa vào. Vì vậy nếu quá trình băm chậm sẽ dẫn đến hệ thống sẽ không thể vận hành tốt được.

**Small Changes In The Input Changes the Hash (tính nhạy cảm).**

Đảm bảo rằng bất kỳ một thay đổi nào, dù là nhỏ nhất trên dữ liệu đều sẽ gây ra sự thay đổi cực lớn trên giá trị băm và tạo ra giá trị băm hoàn toàn khác, và không hề có liên hệ gì với giá trị băm cũ (hiệu ứng tuyết lở- Avalanche effect).

**Pre-Image Resistance(Tính kháng tiền ảnh)**

Pre-Image Resistance hiểu đơn giản nghĩa là việc ta biết 1 hàm hash và 1 output, và ta khó có thể quay ngược lại xác định đầu vào là gì. Tuy nhiên, việc tìm ra input không phải là 1 điều bất khả thi.

**Collision Resistant(Chống va chạm)**

Collision Resistant có nghĩa là trong trường hợp ta có 2 input là A và B, với hàm hash A và hàm hash B tương ứng. Sẽ rất khó để giá trị output của hai hàm hash trên sẽ trùng nhau. Điều đó nghĩa là đối với hầu hết các khả năng, mỗi đầu vào sẽ có 1 hàm băm duy nhất của riêng nó.*Nhưng với ví dụ Birthday Paradox. Birthday Paradox là gì ? Nếu bạn gặp bất kì người lạ ngẫu nhiên nào trên đường phố thì khả năng cả hai có cùng ngày sinh là rất thấp. Tuy nhiên, nghịch lý ở đây là, nếu bạn tập hợp 20-30 người trong 1 phòng thì tỉ lệ hai người có cùng ngày sinh nhật sẽ tăng lên nhanh chóng. Trên thực tế thì có cơ hội 50-50 cho 2 người có cùng ngày sinh trong kịch bản này! Tại sao lại xảy ra nghịch lý như vậy, bởi vì đó là 1 quy tắc đơn giản trong xác suất diễn ra như sau. Giả sử bạn có N khả năng khác nhau của 1 sự kiện xảy ra, thì bạn sẽ cần căn bậc hai của N các vật được chọn ngẫu nhiên của chúng để có 50% xảy ra va chạm. Áp dụng lí thuyết trong trường hợp này, bạn có 365 khả năng khác nhau về ngày sinh nhật, vì vậy bạn chỉ cần căn bậc 2 của 365 tức là khoảng 23 người được chọn ngẫu nhiên để có 50% cơ hội cho 2 người có cùng ngày sinh nhật. Ứng dụng điều này trong hàm băm như thế nào? Giả sử bạn có 1 giá trị băm 128 bit thì có 2^128 khả năng khác nhau. Bằng cách sử dụng nghịch lý trên, bạn có 50% cơ hội xảy ra sự trùng lặp ở lần thứ 2^64.*Vì vậy, không có hàm băm nào có thể đảm bảo rằng việc giá trị output của nó sẽ không bị trùng với giá trị băm của 1 hàm bằng khác, tuy nhiên tốn rất nhiều thời gian để sự va chạm này xảy ra.

**Topic 2: Một số dạng hàm băm**

**MD5**

MD5 (Message-Digest algorithm 5) là một thuật toán băm (hash function) được thiết kế để tạo ra một giá trị hash (đôi khi được gọi là "checksum" hoặc "digest") từ một dữ liệu đầu vào có kích thước bất kỳ. Thuật toán này đã được phát triển bởi Ronald Rivest vào năm 1991 và đã trở thành một phần quan trọng của nhiều ứng dụng bảo mật và kiểm tra tính toàn vẹn.

Đặc điểm quan trọng của MD5:

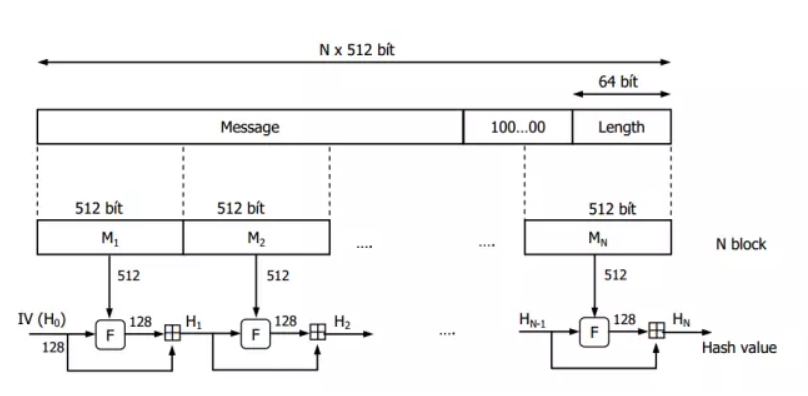
Độ dài hash: MD5 tạo ra một giá trị hash có độ dài cố định là 128-bit, hay 32 ký tự hệ thập lục phân (hexadecimal). Vì vậy, mỗi giá trị hash MD5 được biểu diễn bằng 32 ký tự hex.

Sử dụng phổ biến: MD5 đã được sử dụng rộng rãi trong các ứng dụng bảo mật và kiểm tra tính toàn vẹn. Nó thường được sử dụng để tạo ra một "bảng băm" (hash) của một tệp tin hoặc dữ liệu và sau đó kiểm tra xem tệp tin đã được tải xuống hoặc truyền tải có được thay đổi hay không.

Chuẩn Internet: MD5 đã được chuẩn hóa thông qua RFC 1321, là một tài liệu mô tả cụ thể về cách thuật toán hoạt động và các yếu tố an ninh liên quan.

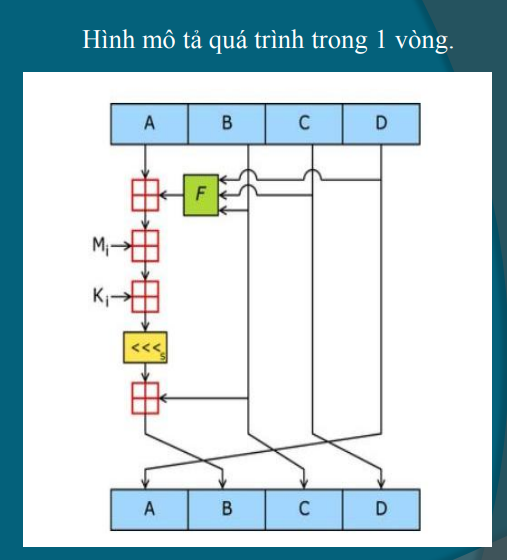
MD5Sum: Các chương trình tính toán hash MD5 thường được gọi là "MD5Sum". Các công cụ này nhận một đầu vào (ví dụ: một tệp tin) và tạo ra giá trị hash MD5 tương ứng với nó.

Tuy nhiên, quan điểm an ninh của MD5 đã bị đặt ra nhiều khi các lỗ hổng bảo mật được phát hiện, và nó không còn được coi là an toàn đối với các ứng dụng yêu cầu mức độ bảo mật cao. Do đó, nhiều hệ thống đã chuyển sang sử dụng các thuật toán băm mạnh mẽ hơn như SHA-256 hoặc SHA-3.

****

MD5 biến đổi một thông điệp có chiều dài bất kỳ thành một khối có kích thước cố định 128 bit. Thông điệp đưa vào sẽ được cắt thành các khối 512 bits. Thông điệp được đưa vào bộ đệm để chiều dài của nó sẽ chia hết cho 512.

Bộ đệm hoạt động như sau: MD5-Thuật toán: Trước tiên nó sẽ chèn bit 1 vào cuối thông điệp. Tiếp đó là hàng loạt bit zero cho tới khi chiều dài của nó nhỏ hơn bội số của 512 một khoảng 64 bit. Phần còn lại sẽ được lấp đầy bởi một số nguyên 64 bit biểu diễn chiều dài ban đầu của thông điệp.

****

Thuật toán chính của MD5 hoạt động trên một bộ 128 bit, được chia thành 4 từ 32 bit mỗi từ, và được ký hiệu là A, B, C, D. Các giá trị này được sử dụng như là các hằng số cố định trong quá trình xử lý. Quá trình xử lý diễn ra trên các khối 512 bit, và mỗi khối thông điệp được chia thành các từ 32 bit.

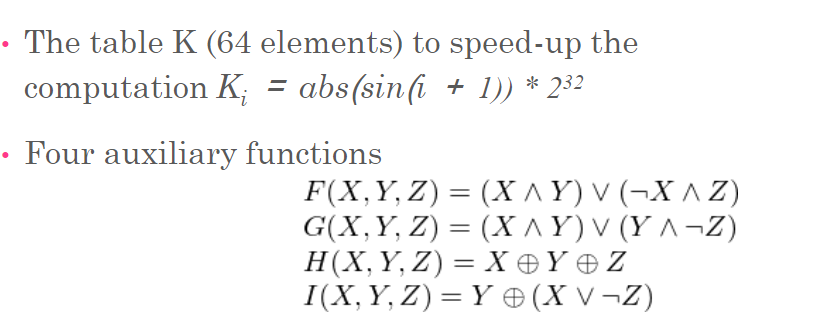
Bước 1: Khởi tạo giá trị.

Gán giá trị khởi tạo cho A, B, C, D.

Bước 2: Xử lý vòng.

Quá trình xử lý mỗi vòng gồm 16 quá trình tương tự nhau, dựa trên hàm một chiều F, phép cộng module và phép xoay trái.

Trong mỗi vòng, mỗi từ trong bộ 128-bit được cập nhật dựa trên các phép toán và giá trị từ vòng trước đó.



MD5 đã bị coi là không an toàn cho nhiều ứng dụng an ninh do nhiều lỗ hổng bảo mật, và nó đã được thay thế bởi các thuật toán băm mạnh mẽ hơn như SHA-2.

**SHA**

# Giải thuật SHA-1

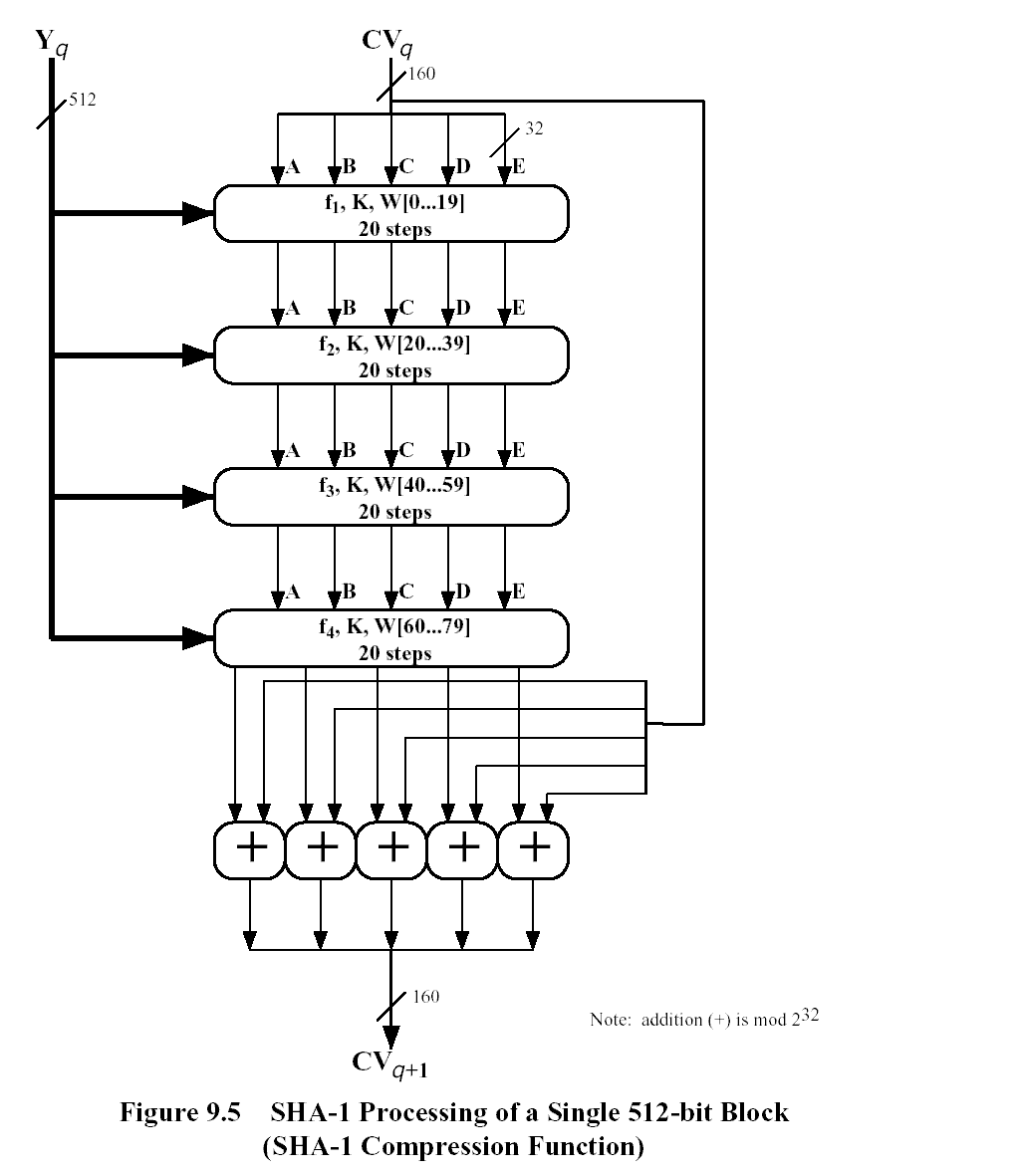
## Secure Hash Algorithm (SHA) phát triển bởi National Institute of Standard and Technology (NIST)

* + - Đầu vào: thông điệp với độ dài tối đa 264 bits
    - Đu ra: giá trị băm (message digest) có độ dài 160 bits
    - Giải thuật gồm 5 bước thao tác trên các khối 512 bits

Giải thuật SHA-1 – Nguyên lý

* + - Bước 1: **nhồi thêm dữ liệu**
      * Thông điệp được nhồi thêm các bits sao cho độ dài ***l = 448 mod 512*** hay ***l = n \* 512 + 448 (n,l nguyên)***
      * Thông điệp luôn luôn được nhồi thêm dữ liệu
      * Số bits nhồi thêm nằm trong khoảng 1 đến 512
      * Phần dữ liệu nhồi them bao gồm 1 bit l và theo sau là các bit 0
    - 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 như phần 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 *Y0, Y1,…, YL-1*
        + Bằng một dãy *N từ (word)* 32-bit *M0, M1, MN 1*. Vậy **N = L x 16**

# Giải thuật SHA-1 – Nguyên lý

* + - Bước 3: **khởi tạo bộ đệm MD** (MD buffer)
      * Một bộ đếm 160-bit được dung lưu trữ các giá trị băm trung gian và kết quả. Bộ đệm được biểu diễn bằng thanh ghi 32-bit với các giá trị khởi tạo ở dạng big-endian 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
      * 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
* Bước 4: **xử lý các khối dữ liệu 512-bit**
  + 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 các hàm logic f1, f2, f3, f4
  + Mỗi vòng có đầu vào gồm khối 512-bit hiện thời và một bộ đệm 160-bit ABCDE. Các thao tác sẽ cập nhật giá trị bộ đệm
  + Mỗi bước sử dụng một hằng số Kt (0 ≤ t ≤ 79)
    - Kt = 5A827999 (0 ≤ t ≤ 19)
    - Kt = 6ED9EBA1 (20 ≤ t ≤ 39)
    - Kt = 8F1BBCDC (40 ≤ t ≤ 59)
    - Kt = CA62C1D6 (60 ≤ t ≤ 79)
  + Đầu ra của 4 vòng (bước 80) được cộng với đầu ra của bước CVq để tạo ra CVq+1
    - Bước 5: **xuất kết quả**
      * bit Sau khi thao tác trên toàn bộ L blocks. Kết quả của khối thứ L là bảng băm 160-bit
      * Giải thuật được tóm tắt như sau:
        + CV0 = IV
        + CVq+1 = SUM32(CVq, ABCDEq)
        + MD = CVL
      * Với
        + IV = Giá trị khởi tạo của bộ đệm ABCDE
        + ABCDEq = đầu ra của hàm nén trên khối thứ q
        + L = số khối 512-bit của thông điệp
        + SUM32 = phép cộng module 232 trên từng (32 bits) 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 € E + f(t, B, C, D) + S5(A) + Wt +

Kt

* + B € A
  + C € S30(B)
  + D € C
  + E € 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
  + Sk = dịch vòng trái k bits
  + Wt = từ thứ t của khối dữ liệu
  + Kt = hằng số

● + = phép cộng module 232

* + - Các hàm f

|  |  |  |
| --- | --- | --- |
| Bước | Hàm | Giá trị |
| (0 ≤ t ≤ 19) | f1 = f(t,B,C,D) | (B ˄ C) ˅ ( ̚ B ˄ D) |
| (20 ≤ t ≤ 39) | f2 = f(t,B,C,D) | B xor C xor D |
| (40 ≤ t ≤ 59) | f3 = f(t,B,C,D) | (B ˄ C) ˅ (B ˄ D) ˅ (C ˄ D) |
| (60 ≤ t ≤ 79) | f4 = f(t,B,C,D) | B xor C xor D |

* Với 0 ≤ t ≤ 15, giá trị Wt lấy trực tiếp từ khối dữ liệu
* Với t > 15 : Wt = S1(Wt-16 xor Wt-14 xor Wt-8 xor Wt-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 2128 thao tác với MD5 và 2160 với SHA-1
    - Để tạo ra thông điệp có cùng giá trị băm, cần 264 thao tác với MD5 và 280 với SHA-1
  + Khả năng chống lại thám mã (cryptanalysis): cả 2 đều có cấu trúc tốt
  + 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

# 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