1. **Kiến trúc của hàm băm**

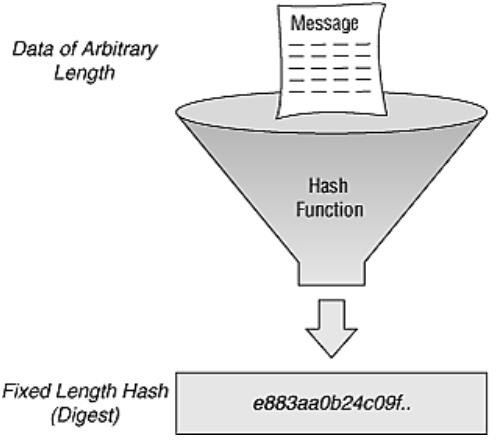
**1.Giới thiệu về hàm băm**

* Nếu bạn mua một chiếc điện thoại mới và lớp bọc co lại của nó bị rách hoặc hư hỏng, bạn có thể biết ngay rằng ai đó đã mở, sử dụng, thay thế hoặc làm hỏng điện thoại. Hàm băm mật mã trong mã hóa cũng giống như vậy nhưng đối với dữ liệu thay vì đối tượng vật lý. Theo cách tương tự, băm giống như đặt màng bọc thu nhỏ ảo lên một phần mềm, ứng dụng hoặc dữ liệu để thông báo cho người dùng nếu nó đã được sửa đổi theo bất kỳ cách nào.
* Nhưng băm là gì? Hashing, hoặc [thuật toán băm](https://www.thesslstore.com/blog/difference-encryption-hashing-salting/) , là một quá trình một chiều chuyển đổi dữ liệu đầu vào của bạn ở bất kỳ kích thước nào thành dữ liệu được nén có độ dài cố định. Trung tâm của quá trình là nơi bạn sẽ tìm thấy hàm băm. Về cơ bản, bạn có thể lấy một câu ngắn hoặc toàn bộ luồng dữ liệu, chạy nó thông qua một hàm băm và kết thúc với một chuỗi dữ liệu có độ dài cụ thể. Đó là một cách để ẩn dữ liệu ban đầu của bạn để làm cho việc thiết kế ngược trở nên khó khăn nhất có thể.
* Theo nghĩa kỹ thuật hơn, đó là một kỹ thuật sử dụng một phép toán học để thu nhỏ một lượng ngẫu nhiên dữ liệu đầu vào (được gọi là khóa băm) thành một chuỗi bit có độ dài cố định theo cách quá phi thực tế để đảo ngược với các máy tính hiện đại. Vì vậy, định nghĩa của hàm băm sẽ là thứ lấy dữ liệu đầu vào và sử dụng nó để tạo ra giá trị đầu ra có độ dài cố định, duy nhất và hầu như không thể thay đổi (cho tất cả các ý định và mục đích thực tế).

**2.Định nghĩa tổng quát của hàm băm**

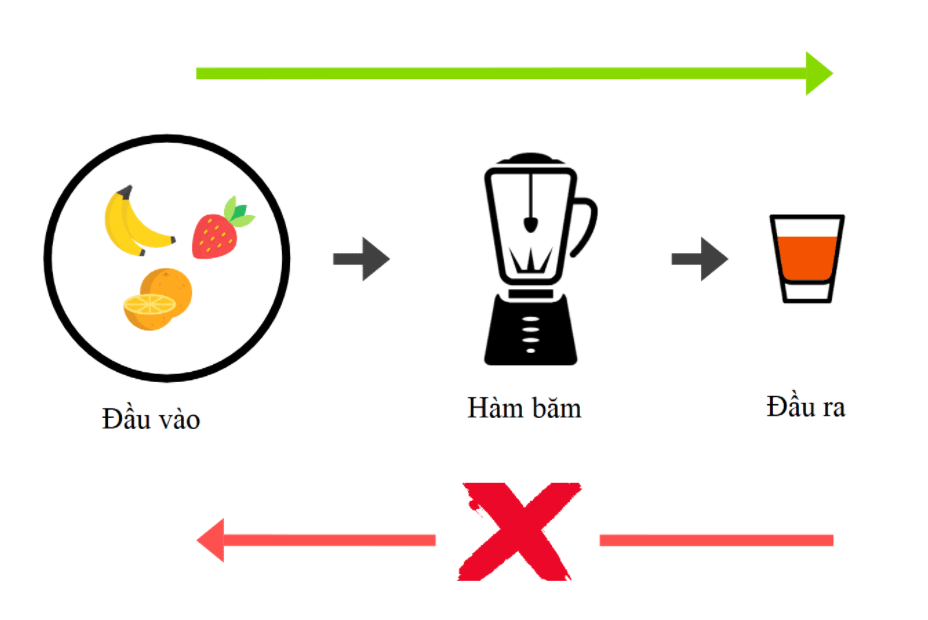
Hàm băm (hash function) là một hàm toán học h có tối thiểu 2 thuộc tính:

* Nén (Compression): h là một ánh xạ từ chuỗi đầu vào x có chiều dài bất kỳ sang một chuỗi đầu ra h(x) có chiều dài cố định n bit.
* Dễ tính toán (Ease of computation): cho trước hàm h và đầu vào x, việc tính toán h(x) là dễ dàng.[1]



**3 .Tính chất cơ bản của hàm băm**

* **Tính một chiều:** không thể suy ra dữ liệu ban đầu từ kết quả, điều này tương tự như việc bạn không thể chỉ dựa vào một dấu vân tay lạ mà suy ra ai là chủ của nó được.
* **Tính duy nhất**: xác suất để có một vụ va chạm (hash collision), tức là hai thông điệp khác nhau có cùng một kết quả hash là cực kỳ nhỏ

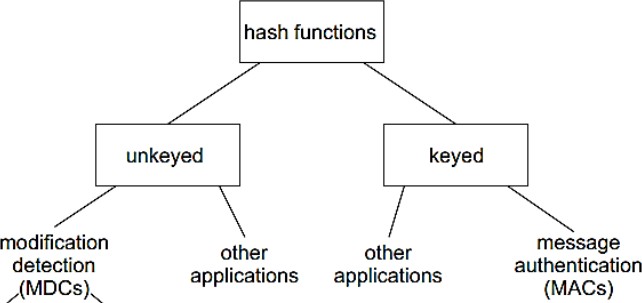


**4.Ngoài ra, với mục đích đảm bảo an toàn cho dữ liệu, các hàm băm mật mã phải có khả năng chịu được tất cả các loại tấn công mã hóa đã biết. Trong lý thuyết mật mã, mức độ an toàn của hàm băm mật mã đã được xác định bằng các thuộc tính sau:**

* Tính kháng tiền ảnh thứ nhất. Tính chất yêu cầu rằng với một giá trị băm h bất kỳ, sẽ khó tìm thấy bất kỳ thông điệp m nào sao cho h = hash (m). Khái niệm này có liên quan đến tính chất một chiều của hàm băm.
* Tính kháng tiền ảnh thứ hai. Với đầu vào m1, sẽ khó tìm được đầu vào m2 khác sao cho hash(m1) = hash (m2).
* Tính kháng va chạm. Rất khó để tìm thấy hai thông điệp khác nhau m1 và m2 sao cho hash (m1) = hash (m2). Một giá trị như vậy được gọi là va chạm của hàm băm mật mã.

**5.Phân loại hàm băm theo khóa sử dụng**

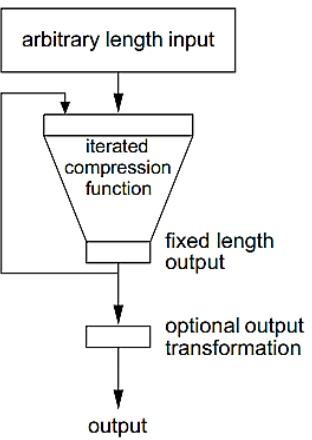
* Hàm băm không khóa (unkeyed): đầu vào chỉ là thông điệp
* Hàm băm có khóa (keyed): đầu vào gồm thông điệp và khóa



Theo chức năng, có thể chia các hàm băm thành 2 loại chính:

* Mã phát hiện sửa đổi (MDC - Modification Detection Code): MDC thường được sử dụng để tạo chuỗi đại diện cho thông điệp và dùng kết hợp với các kỹ thuật khác (như chữ ký số) để đảm bảo tính toàn vẹn của thông điệp. MDC thuộc loại hàm băm không khóa. MDC gồm 2 loại nhỏ:
* Hàm băm một chiều (OWHF - One-way hash functions): Với hàm băm một chiều, việc tính giá trị băm là dễ dàng, nhưng việc khôi phục thông điệp từ giá trị băm là rất khó khăn;
* Hàm băm chống đụng độ (CRHF - Collision resistant hash functions): Với hàm băm chống đụng độ, sẽ là rất khó để tìm được 2 thông điệp khác nhau nhưng có cùng giá trị băm.
* Mã xác thực thông điệp (MAC - Message Authentication Code): MAC cũng được dùng để đảm bảo tính toàn vẹn của thông điệp mà không cần một kỹ thuật bổ sung nào khác. MAC là loại hàm băm có khóa như đã đề cập ở trên, với đầu vào là thông điệp và một khóa bí mật [1].

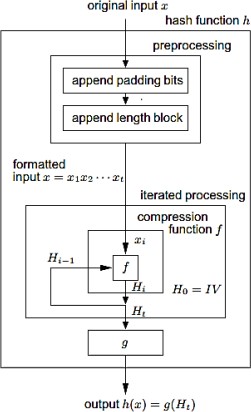
**6.Mô hình xử lí dữ liệu**



Thông điệp đầu vào với độ dài tùy ý (arbitrary length input) đi qua hàm nén lặp nhiều vòng 11 (iterated compression function) để tạo chuỗi đầu ra có kích thước cố định (fixed length output). Chuỗi này đi qua một khâu chuyển đổi định dạng tùy chọn (optional output transformation) để tạo ra chuỗi băm kết quả (output)

**7.Mô hình chi tiết xử lí dữ liệu của hàm băm**

Quá trình xử lý gồm 3 bước chính: (1) tiền xử lý (preprocessing), (2) xử lý lặp (iterated processing) và (3) chuyển đổi định dạng. Trong bước tiền xử lý, thông điệp đầu vào x trước hết được nối đuôi thêm một số bit và kích thước khối, sau đó chia thành các khối có kích thước xác định. Kết quả của bước này là t khối dữ liệu có cùng kích thước có dạng x = x1 x2…xt làm đầu vào cho bước 2. Trong bước 2, từng khối dữ liệu xi được xử lý thông qua hàm nén f để tạo đầu ra là Hi. Kết quả của bước 2 là chuỗi đầu ra Ht và Ht được chuyển đổi định dạng bởi hàm g để tạo chuỗi giá trị băm hết quả h(x).



**Kết luận:**

Thông qua phần 1 kiến trúc của hàm băm ta đã hiểu được định nghĩa của hàm băm , các tính chất , phân loại hàm băm dựa trên các yếu tố khác nhau và môn hình xử lí dữ liệu của nó

**Nguồn :**

* https://sectigostore.com/blog/hash-function-in- cryptography-how-does-it-work/
* Bài giảng cơ sở an toàn thông tin – Hoàng Xuân Dậu
* Wikipedia https://vi.wikipedia.org/wiki/H%C3%A0m\_b%C4%83m
* https://cryptoviet.com/hash-la-gi