CHƯƠNG 9

HÀM BĂM VÀ TOÀN VẸN DỮ LIỆU
HASH FUNCTIONS AND DATA INTEGRITY

Nội dung

- Khái niệm và phân loại về hàm băm

Nội dung

- Khái niệm và phân loại về hàm băm
- Cách cơ bản xây dựng hàm băm

Nội dung

- Khái niệm và phân loại về hàm băm
- Cách cơ bản xây dựng hàm băm
- Hàm băm dùng khóa

Nội dung

- Khái niệm và phân loại về hàm băm
- Cách cơ bản xây dựng hàm băm
- Hàm băm dùng khóa
- Toàn vẹn dữ liệu và xác thực tin

Định nghĩa

Hàm băm là một hàm h ít nhất thỏa mãn hai tính chất sau:

(i) *h ánh xạ mọi giá trị đầu vào có kích thước bất kỳ cho kết quả là đoạn tin có độ dài bit cố định*

Định nghĩa

Hàm băm là một hàm h ít nhất thỏa mãn hai tính chất sau:

(i) *h ánh xạ mọi giá trị đầu vào có kích thước bất kỳ cho kết quả là đoạn tin có độ dài bit cố định*

(ii) *h đảm bảo dễ dàng tính toán*

Khái niệm hàm băm

- Mục đích ứng dụng cụ thể của hàm băm là đảm bảo tính toàn vẹn dữ liệu và xác thực tin

Khái niệm hàm băm

- Mục đích ứng dụng cụ thể của hàm băm là đảm bảo tính toàn vẹn dữ liệu và xác thực tin
- Giá trị của h ứng với mỗi đầu vào gọi là giá trị băm (hash value, hash code ...)

Khái niệm hàm băm

- Mục đích ứng dụng cụ thể của hàm băm là đảm bảo tính toàn vẹn dữ liệu và xác thực tin
- Giá trị của h ứng với mỗi đầu vào gọi là giá trị băm (hash value, hash code ...)
- Vì các giá trị băm có độ dài bit cố định trong khi số đầu vào là vô hạn nên h là ánh xạ kiểu $n - 1$. Điều đó nghĩa là sẽ có trường hợp hai giá trị đầu vào khác nhau mà có cùng giá trị băm. Trường hợp đó gọi là đụng độ của hàm băm.

Khái niệm hàm băm

- Mục đích ứng dụng cụ thể của hàm băm là đảm bảo tính toàn vẹn dữ liệu và xác thực tin
- Giá trị của h ứng với mỗi đầu vào gọi là giá trị băm (hash value, hash code ...)
- Vì các giá trị băm có độ dài bit cố định trong khi số đầu vào là vô hạn nên h là ánh xạ kiểu $n - 1$. Điều đó nghĩa là sẽ có trường hợp hai giá trị đầu vào khác nhau mà có cùng giá trị băm. Trường hợp đó gọi là đụng độ của hàm băm.
- Ý tưởng cơ bản của hàm băm mã hóa là sử dụng giá trị băm xem như đại diện cho một thực thể mà nó được xác định duy nhất bởi giá trị băm đó.

1. Không khả thi về mặt tính toán để tìm ra tạo ảnh x mà $h(x) = y$ khi biết trước y.

Tính chất cơ bản của hàm băm

1. Không khả thi về mặt tính toán để tìm ra tạo ảnh x mà $h(x) = y$ khi biết trước y.

2. Không khả thi để tìm ra tại ảnh thứ 2 x' mà $h(x') = h(x)$ với x cho trước.

Tính chất cơ bản của hàm băm

1. Không khả thi về mặt tính toán để tìm ra tạo ảnh x mà $h(x) = y$ khi biết trước y.

2. Không khả thi để tìm ra tại ảnh thứ 2 x' mà $h(x') = h(x)$ với x cho trước.

3. Không khả thi để tìm ra hai giá trị bất kỳ khác nhau x_1, x_2 mà $h(x_1) = h(x_2)$.

Tính chất cơ bản của hàm băm

1. Không khả thi về mặt tính toán để tìm ra tạo ảnh x mà $h(x) = y$ khi biết trước y.

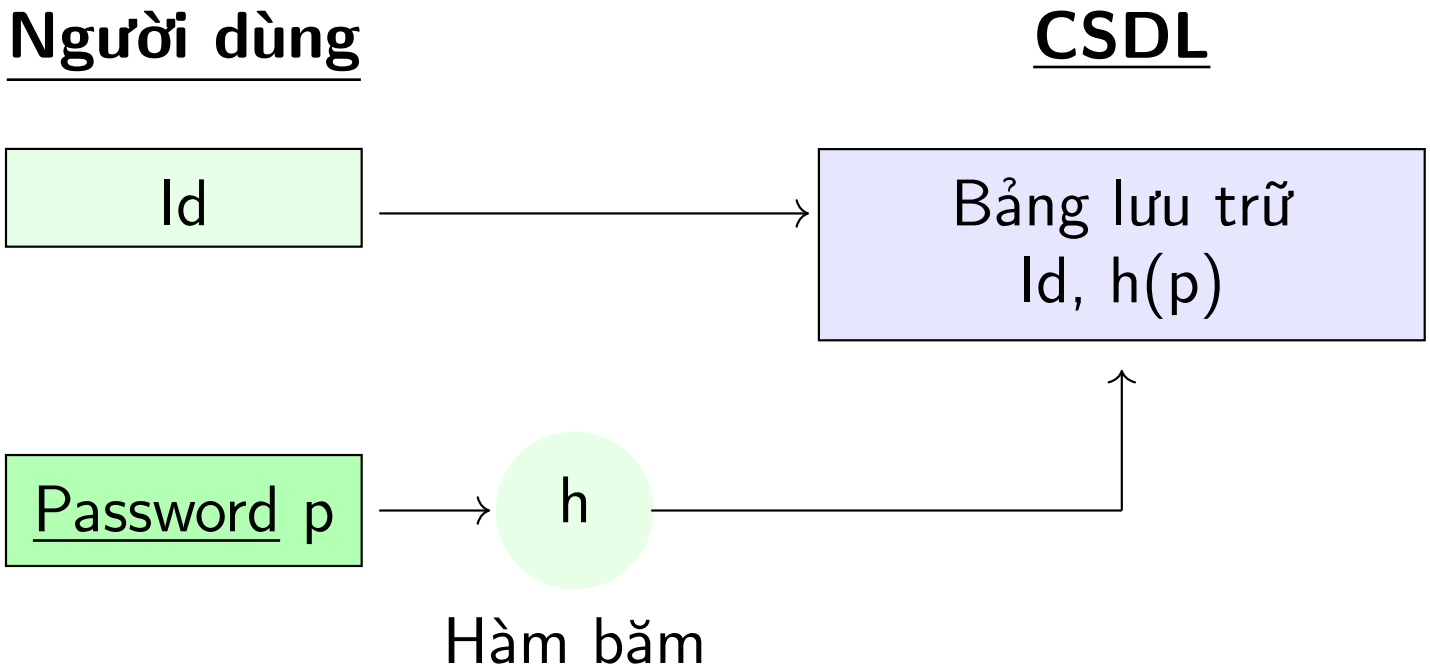
2. Không khả thi để tìm ra tại ảnh thứ 2 x' mà $h(x') = h(x)$ với x cho trước.

3. Không khả thi để tìm ra hai giá trị bất kỳ khác nhau x_1, x_2 mà $h(x_1) = h(x_2)$.

Chú ý: Về lý thuyết thì các khả năng trên đều có thể xảy ra vì kiểu ánh xạ $n \longrightarrow 1$ của hàm băm.

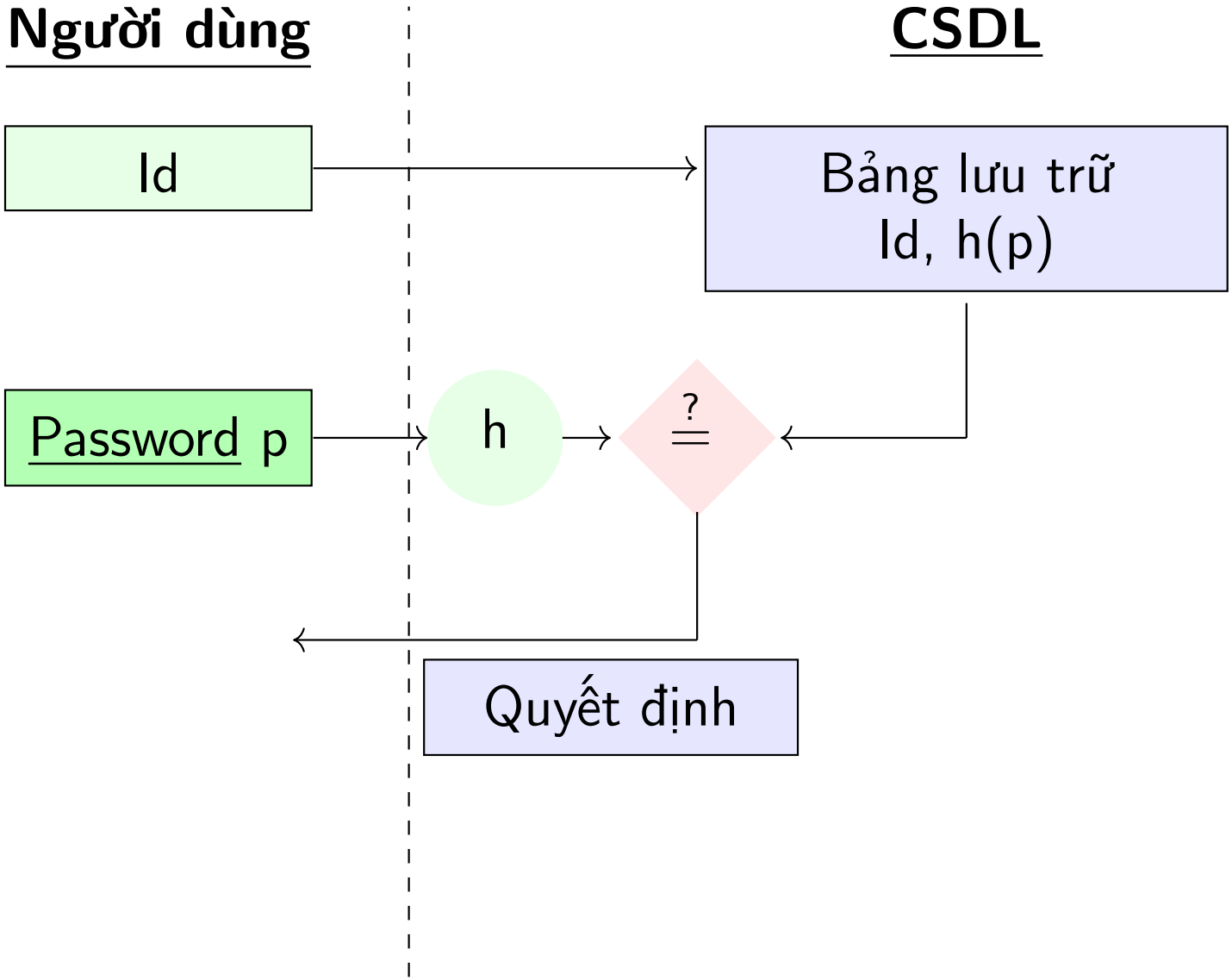
Một ứng dụng thực tế của hàm băm

- Lưu trữ thông tin đăng ký của người dùng



Một ứng dụng thực tế của hàm băm

- Xác thực khi đăng nhập



Kiểm tra sự thay đổi, chỉnh sửa (Modification Detection Codes - MDCs): Mục đích của MDC là cung cấp ảnh đại diện hoặc giá trị băm của một file dữ liệu và thỏa mãn tính chất dưới đây:

Kiểm tra sự thay đổi, chỉnh sửa (Modification Detection Codes - MDCs): Mục đích của MDC là cung cấp ảnh đại diện hoặc giá trị băm của một file dữ liệu và thỏa mãn tính chất dưới đây:

(i) Tính một chiều, nghĩa là khó có thể tìm được giá trị đầu vào để ảnh là giá trị băm cho trước. Hàm băm thỏa mãn tính chất này gọi là hàm băm một chiều (OWHF)

Kiểm tra sự thay đổi, chỉnh sửa (Modification Detection Codes - MDCs): Mục đích của MDC là cung cấp ảnh đại diện hoặc giá trị băm của một file dữ liệu và thỏa mãn tính chất dưới đây:

(i) Tính một chiều, nghĩa là khó có thể tìm được giá trị đầu vào để ảnh là giá trị băm cho trước. Hàm băm thỏa mãn tính chất này gọi là hàm băm một chiều (OWHF)

(ii) Chống đụng độ, nghĩa là khó có thể tìm được 2 giá trị đầu vào khác nhau mà có cùng giá trị băm.Hàm băm thỏa mãn tính chất này gọi là hàm băm kháng đụng độ (CRHF).

Kiểm tra sự thay đổi, chỉnh sửa (Modification Detection Codes - MDCs): Mục đích của MDC là cung cấp ảnh đại diện hoặc giá trị băm của một file dữ liệu và thỏa mãn tính chất dưới đây:

(i) Tính một chiều, nghĩa là khó có thể tìm được giá trị đầu vào để ảnh là giá trị băm cho trước. Hàm băm thỏa mãn tính chất này gọi là hàm băm một chiều (OWHF)

(ii) Chống đụng độ, nghĩa là khó có thể tìm được 2 giá trị đầu vào khác nhau mà có cùng giá trị băm.Hàm băm thỏa mãn tính chất này gọi là hàm băm kháng đụng độ (CRHF).

Xác thực tin hoặc thực thể (Message Authentication Codes - MACs): Mục đích của MAC là để có thể đảm bảo liên hệ giữa nguồn tin và tính toàn vẹn của nó mà không cần sử dụng thêm kỹ thuật khác nữa.

Định nghĩa

Một thuật toán MAC là họ các hàm h_k với tham số k là khóa va thỏa mãn các tính chất sau:

- *Dễ cài đặt tính toán, nghĩa là $h_k(x)$ dễ dàng tính được với giá trị đầu vào x và khóa k.*
- *Có tính nén dữ liệu, nghĩa là ánh xạ từ x có độ dài bit bất kỳ cho kết quả có độ dài bit cố định.*
- *Chống lại tính toán, nghĩa là cho trước các cặp $(x_i, h_k(x_i))$ (nếu có) thì không khả thi để tính được bất cứ cặp $(x, h_k(x))$ với bất cứ giá trị mới x.*

Với MDC:

- Tấn công vào các hàm OWHF, nghĩa là tìm x để $h(x) = y$ cho trước hoặc tìm x' mà $h(x) = h(x')$ với x biết trước.

Với MDC:

- Tấn công vào các hàm OWHF, nghĩa là tìm x để $h(x) = y$ cho trước hoặc tìm x' mà $h(x) = h(x')$ với x biết trước.

- Tấn công các hàm CRHF, nghĩa là tìm 2 giá trị $x_1 \neq x_2$ để $h(x_1) = h(x_2)$

Mục tiêu của kẻ địch với MDC và MAC

Với MDC:

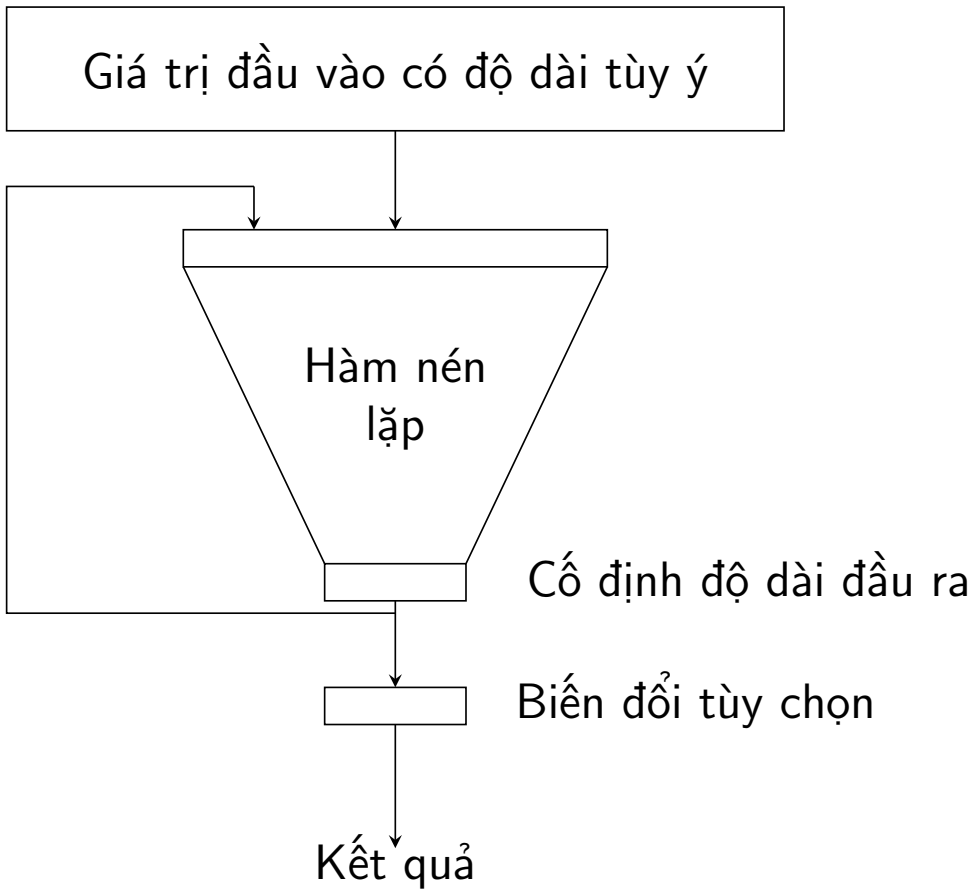
- Tấn công vào các hàm OWHF, nghĩa là tìm x để $h(x) = y$ cho trước hoặc tìm x' mà $h(x) = h(x')$ với x biết trước.

- Tấn công các hàm CRHF, nghĩa là tìm 2 giá trị $x_1 \neq x_2$ để $h(x_1) = h(x_2)$

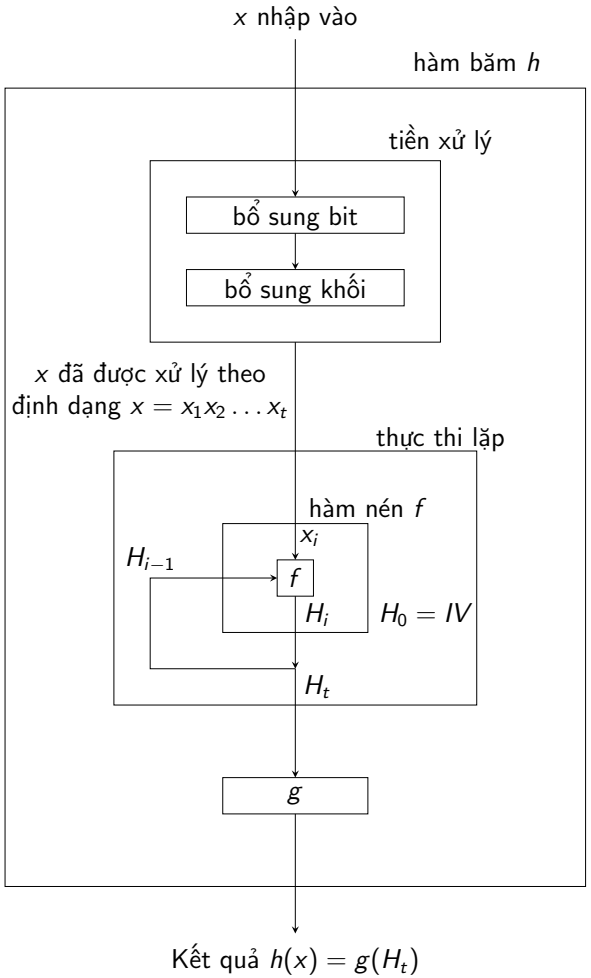
Với MAC:

- Tấn công vào MAC, nghĩa là tính cặp MAC mới $(x, h_k(x))$ khác với các cặp $(x_i, h_k(x_i))$ cho trước và khóa k không được biết.

Mô hình cô đọng chung



Mô hình chi tiết



Hàm băm MD5

- MD = Message Digest

Hàm băm MD5

- MD = Message Digest
- MD5: $\{0,1\}^* \longrightarrow \{0,1\}^{128}$

Qui ước ban đầu:

- Giả sử ta có đoạn tin đầu vào là M_0 có biểu diễn ở nhị phân gồm b bit như sau:

$$[\mathrm{m}_0, \mathrm{m}_1, \ldots, \mathrm{m}_{\mathrm{b}-1}]$$

Qui ước ban đầu:

- Giả sử ta có đoạn tin đầu vào là M_0 có biểu diễn ở nhị phân gồm b bit như sau:

$$[\mathrm{m}_0, \mathrm{m}_1, \ldots, \mathrm{m}_{\mathrm{b}-1}]$$

- Một "từ" trong thuật toán này được qui ước hiểu là gồm 4 bytes $\approx$ 32 bits khi biểu diễn nhị phân.

- **Bước 1**: Bổ sung thêm bit vào sau tin M_0 để được tin mới có độ dài là $\equiv 448 (mod\, 512)$.

- **Bước 1**: Bổ sung thêm bit vào sau tin M_0 để được tin mới có độ dài là $\equiv 448(mod\,512)$.
  - Ở đây viết $\mathrm{a} \equiv \mathrm{b}(\mathrm{mod}\,\mathrm{d})$ nghĩa là $\mathrm{a} - \mathrm{b}$ chia hết cho d.

- **Bước 1**: Bổ sung thêm bit vào sau tin M_0 để được tin mới có độ dài là $\equiv 448(mod\ 512)$.
  - Ở đây viết $\mathrm{a} \equiv \mathrm{b}(\mathrm{mod}\,\mathrm{d})$ nghĩa là $\mathrm{a} - \mathrm{b}$ chia hết cho d.
  - Nếu tin gốc M_0 có độ dài b mà $\mathrm{b} \equiv 448(\mathrm{mod}512)$ thì ta vẫn bổ sung thêm 512 bit nữa.

- **Bước 1**: Bổ sung thêm bit vào sau tin M_0 để được tin mới có độ dài là $\equiv 448(mod\ 512)$.
  - Ở đây viết $\mathrm{a} \equiv \mathrm{b}(\mathrm{mod}\,\mathrm{d})$ nghĩa là $\mathrm{a} - \mathrm{b}$ chia hết cho d.
  - Nếu tin gốc M_0 có độ dài b mà $\mathrm{b} \equiv 448(\mathrm{mod}512)$ thì ta vẫn bổ sung thêm 512 bit nữa.
  - Khi bổ sung thêm bit thì bit đầu tiên thêm vào là 1 và tiếp theo còn lại là 0.

- **Bước 1**: Bổ sung thêm bit vào sau tin M_0 để được tin mới có độ dài là $\equiv 448(mod\ 512)$.
  - Ở đây viết $\mathrm{a} \equiv \mathrm{b}(\mathrm{mod}\,\mathrm{d})$ nghĩa là $\mathrm{a} - \mathrm{b}$ chia hết cho d.
  - Nếu tin gốc M_0 có độ dài b mà $\mathrm{b} \equiv 448(\mathrm{mod}512)$ thì ta vẫn bổ sung thêm 512 bit nữa.
  - Khi bổ sung thêm bit thì bit đầu tiên thêm vào là 1 và tiếp theo còn lại là 0.
  - Khi bổ sung bit thì có ít nhất 1 bit và nhiều nhất là 512 bit được thêm vào tin gốc M_0.

- **Bước 2**: Biểu diễn d (là độ dài của tin gốc) ở dạng 64-bit.

- **Bước 2**: Biểu diễn d (là độ dài của tin gốc) ở dạng 64-bit.
  - Nếu $\mathrm{d} > 2^{64}$ (d được biểu diễn vượt quá 64 bit) thì ta biểu diễn d ở cấp nhỏ hơn.

- **Bước 2**: Biểu diễn d (là độ dài của tin gốc) ở dạng 64-bit.
  - Nếu $\mathrm{d} > 2^{64}$ (d được biểu diễn vượt quá 64 bit) thì ta biểu diễn d ở cấp nhỏ hơn.
  - Khi bổ sung bit ở bước 1 và 2 thì ta được tin mới có độ dài là bội của 512, cũng có nghĩa là được biểu diễn ở bội của 16 ký tự (1 ký tự gồm 32 bit như qui ước).

- **Bước 2**: Biểu diễn d (là độ dài của tin gốc) ở dạng 64-bit.
  - Nếu $\mathrm{d} > 2^{64}$ (d được biểu diễn vượt quá 64 bit) thì ta biểu diễn d ở cấp nhỏ hơn.
  - Khi bổ sung bit ở bước 1 và 2 thì ta được tin mới có độ dài là bội của 512, cũng có nghĩa là được biểu diễn ở bội của 16 ký tự (1 ký tự gồm 32 bit như qui ước).
  - Giả sử khi bổ sung xong ta được tin mới $\mathrm{M}[0, \ldots \mathrm{N}-1]$ có N ký tự ở đó N chia hết cho 16.

- **Bước 3.1**: Khởi tạo bộ đệm của MD:
  - A=0x67452301
  - B=0xefcdab89
  - C=0x98badcfe
  - D=0x1032476

- **Bước 3.2**: Khởi tạo hàm F(X,Y,Z) lần lượt là:
  - F1(X,Y,Z) = XY v not(X) Z
  - F2(X,Y,Z) = XZ v Y not(Z)
  - F3(X,Y,Z) = X xor Y xor Z
  - F4(X,Y,Z) = Y xor (X v not(Z))

- **Bước 3.2**: Khởi tạo hàm F(X,Y,Z) lần lượt là:
  - F1(X,Y,Z) = XY v not(X) Z
  - F2(X,Y,Z) = XZ v Y not(Z)
  - F3(X,Y,Z) = X xor Y xor Z
  - F4(X,Y,Z) = Y xor (X v not(Z))
- **Bước 3.3**: Tính các giá trị
  $\mathrm{K_i} = \mathrm{floor}[2^{32} * \mathrm{sin(i)}], \quad \mathrm{i} = 1, \ldots, 64.$

Tài liệu tham khảo

Handbook of Applied Crytography, chapter 9

http://cacr.uwaterloo.ca/hac/