



BÀI GIẢNG AN TOÀN & BẢO MẬT THÔNG TIN CHƯƠNG 5- MÃ CÔNG KHAI VÀ CHỮ KÝ ĐIỆN TỬ

TS. NGUYỄN ĐÌNH DƯƠNG
BỘ MÔN KHMT - KHOA CÔNG NGHỆ THÔNG TIN

Email: duongnd@utc.edu.vn

Ngày 03/07/2022



Nội dung

Chữ ký điện tử

- 1.1 Mở đầu
- 1.2 Sơ đồ chữ kí RSA
- 1.3 Sơ đồ chữ kí ElGamal
- 1.4 Chuẩn chữ kí điện tử
- 1.5 Mô hình ứng dụng của chữ kí số

Hàm băm

- 2.1 Giới thiệu
- 2.2 Thuật toán MD5
- 2.3 Thuật toán SHA

Trao đổi



Nội dung

Chữ ký điện tử

- 1.1 Mở đầu
- 1.2 Sơ đồ chữ kí RSA
- 1.3 Sơ đồ chữ kí ElGamal
- 1.4 Chuẩn chữ kí điện tử
- 1.5 Mô hình ứng dụng của chữ kí số

Hàm băm

- 2.1 Giới thiệu
- 2.2 Thuật toán MD5
- 2.3 Thuật toán SHA

Trao đổi



1. Chữ ký điện tử

1. 1. Mở đầu

Ví dụ 1.1

lệnh chuyển tiền từ một người A tới người B thực hiện bởi một ứng dụng trên máy tính

- Về văn bản đây là một dạng séc đã được "máy tính hoá"
- Giao dịch ở dạng giấy tờ được thực hiện như sau:
 - Séc là một đối tượng xác định có tư cách giao dịch thương mại
 - Chữ ký trên séc xác nhận tính xác thực bởi vì chắc chắn chỉ có người kí hợp pháp mới có thể tạo được chữ kí này
 - Trong trường hợp bất hợp pháp thì sẽ có một bên thứ 3 tham gia vào để phán xét tính xác thực
 - Séc bị huỷ để nó không được sử dụng lại
 - Séc giấy không thể thay đổi được (hoặc hầu hết các kiểu thay đổi đều có thể dễ dàng phát hiện được)



1. Chữ ký điện tử

1. 1. Mở đầu

Xét mô hình giao dịch trên máy tính:

- Alice gửi cho ngân hàng của mình một thông báo ủy quyền ngân hàng chuyển 100\$ cho Bob
- Ngân hàng của Bob phải làm những việc sau:
 - Kiểm tra và chứng tỏ được thông báo này thực sự đến từ Alice
 - Phải biết chắc rằng toàn bộ thông báo này là của Alice và nó không bị chỉnh sửa
- Alice cũng muốn biết chắc rằng ngân hàng của mình không thể giả mạo những thông báo tương tự
- Cả hai đều muốn đảm bảo rằng thông báo đó là thông báo mới, không phải là một thông báo trước đó được sử dụng lại và nó không bị sửa đổi trong khi truyền.



1. Chữ ký điện tử

1. 1. Mở đầu

- **Chữ kí số** là một giao thức tạo ra một hiệu quả tương tự như chữ kí viết tay:
 - Là một dấu hiệu mà chỉ có người gửi mới có thể tạo ra nhưng những người khác có thể nhận ra được rằng nó là của người gửi
 - Sử dụng để xác nhận nội dung thông báo
- Chữ kí số phải thoả mãn điều kiện:
 - **Không thể giả mạo:** Nếu Alice kí thông báo M bằng chữ kí $S(M,A)$ thì không một ai có thể tạo được cặp $[M, S(M,A)]$
 - **Xác thực:** Nếu Bob nhận được cặp $[M, S(M,A)]$ thì có thể kiểm tra được rằng chữ kí có thực sự là của Alice hay không?
 - **Không thể thay đổi:** sau khi được phát M không thể bị thay đổi bởi S , Bob hoặc bởi một kẻ thu trộm nào đó
 - **Không thể sử dụng lại:** một thông báo trước đó đưa ra sẽ ngay lập tức bị Bob phát hiện



1. Chữ ký điện tử

1. 1. Mở đầu

- Một sơ đồ chữ ký số thường chứa hai thành phần:
thuật toán kí sig() + **thuật toán xác minh ver()**
- Alice kí một thông điệp x dùng thuật toán kí **bí mật** (an toàn) với khoá K để được $y = \text{sig}_K(x)$
- Bob có thể kiểm tra chữ ký y bằng thuật toán xác minh **công khai** $\text{ver}_K(x, y)$

$$\text{ver}_K(x, y) = \begin{cases} \text{True} & \text{nếu } y = \text{sig}(x) \\ \text{False} & \text{nếu } y \neq \text{sig}(x) \end{cases}$$

- Một số sơ đồ chữ ký thường gặp: **RSA, ElGamal**



1. Chữ ký điện tử

1. 2. Sơ đồ chữ kí RSA

- Dựa trên bài toán phân tích ra thừa số nguyên tố
- Đảo ngược hàm mã hoá và giải mã trong hệ mã hoá RSA
 - Cho $n = p \cdot q$, trong đó p, q là các số nguyên tố
 - Chọn a, b là các số nguyên $< n$ thoả mãn $ab \equiv 1 \pmod{\Phi(n)}$
 - Đặt $K = (n, p, q, a, b)$, trong đó (n, b) công khai, còn (p, q, a) là bí mật

$$y = \text{sig}_K(x) = x^a \pmod{n}, \quad \text{ver}_K(x, y) = \text{True} \Leftrightarrow x \equiv y^b \pmod{n}$$

* **Chú ý:** Thông thường, chữ kí số được kết hợp với hàm mã hoá công khai:

- Với bản rõ x cho trước, Alice sẽ tính toán chữ kí của mình $y = \text{sig}_A(x)$, sau đó mã hoá cả x, y sử dụng khoá công khai e_B của Bob $\rightarrow z = e_B(x, y)$
- Bob sẽ giải mã z với hàm giải mã d_B của mình để nhận được (x, y) . Sau đó dùng hàm xác minh công khai của Alice để kiểm tra $\text{ver}_A(x, y) = \text{True}$?

1. Chữ ký điện tử

1. 3. Sơ đồ chữ ký ElGamal

- Được đề xuất năm 1985
- NIST cải tiến và thiết kế riêng biệt cho mục đích chữ ký → thành chuẩn chữ ký điện tử - DSS (khác với RSA dùng cho cả mã hoá công khai và chữ ký)
- Cho p là số nguyên tố, α là căn nguyên thuỷ theo mod p
- Chọn số bí mật $a < p$ rồi tính $\beta = \alpha^a \pmod{p}$ → **công khai** (p, α, β)
- Với khoá $K = (p, \alpha, a, \beta)$ và bản rõ x , chọn ngẫu nhiên số $k < p - 1$:

$$(r, s) = \text{sig}_K(x, k), \quad \begin{cases} r = \alpha^k \pmod{p} \\ s = (x - a \cdot r) \cdot k^{-1} \pmod{p-1} \end{cases}$$
$$\text{ver}_K(x, r, s) = \text{True} \Leftrightarrow \beta^r \cdot r^s \equiv \alpha^x \pmod{p}$$

- Giải thích:** Nếu chữ ký đúng thì việc xác nhận thành công vì

$$\beta^r \cdot r^s \equiv \alpha^{ar} \alpha^{ks} \pmod{p} \equiv \alpha^x \pmod{p}, \text{ trong đó } ar + ks \equiv x \pmod{p-1}$$



Ví dụ 1.2

- Chọn $p = 467$, $\alpha = 2$, $a = 127$
- Tính $\beta = \alpha^a \pmod{p} = 2^{127} \pmod{467} = 132$
- A muốn kí lên bức điện $x = 100$, chọn ngẫu nhiên $k = 213 < p - 1$. Để ý rằng $\gcd(213, 466) = 1$ và $213^{-1} \pmod{466} = 431$
- Tính $\begin{cases} r = \alpha^k \pmod{p} = 2^{213} \pmod{467} = 29 \\ s = (x - a \cdot r) \cdot k^{-1} \pmod{p-1} = (100 - 127 \cdot 29) \cdot 431 \pmod{466} = 51 \end{cases}$
- Bất kì ai cũng có thể kiểm tra chữ kí này bằng cách:
$$\begin{cases} \beta^r \cdot r^s = 132^{213} \cdot 29^{51} \equiv 189 \pmod{467} \\ 2^{100} \equiv 189 \pmod{467} \end{cases}$$

- * **Chú ý:** Kẻ thứ ba C muốn giả mạo chữ kí của A trên x nhưng không biết số bí mật a
- Nếu C chọn một giá trị r và cố gắng tìm $s \rightarrow$ tính $\log_r \alpha^x \beta^{-r}$
 - Nếu C chọn một giá trị s và cố gắng tìm $r \rightarrow$ tính $\beta^r \cdot r^s \equiv \alpha^x \pmod{p}$



1. Chữ ký điện tử

1. 3. Sơ đồ chữ kí ElGamal

- C có thể giả mạo chữ kí bằng cách **sử dụng lại chữ kí trước đó**, tức là (r, s) (giá trị chữ kí của x) được C kí cho nhiều bức điện khác
- Cho h, i, j là các số nguyên, $0 \leq i, j, h \leq p - 2$ và $\gcd(hr - js, p - 1) = 1$.
- Tính

$$\begin{cases} r' = r^h \cdot \alpha^i \cdot \beta^j \pmod{p} \\ s' = s \cdot r' \cdot (hr - js)^{-1} \pmod{p-1} \\ x' = r' \cdot (hx + is) \cdot (hr - js)^{-1} \pmod{p-1} \end{cases}$$

- Có thể kiểm tra $\beta^{r'} \cdot (r')^{s'} \equiv \alpha^{x'} \pmod{p} \Rightarrow (r', s')$ là chữ kí đúng của x'



1. Chữ ký điện tử

1. 3. Sơ đồ chữ kí ElGamal

- Một sai lầm của Alice là sử dụng cùng giá trị k khi kí 2 bức điện khác nhau
- Cho (r, s_1) là chữ kí trên bức điện x_1 , (r, s_2) là chữ kí trên bức điện x_2
- Việc kiểm tra sẽ thực hiện:

$$\beta^r r^{s_1} \equiv \alpha^{x_1} \pmod{p}; \quad \beta^r r^{s_2} \equiv \alpha^{x_2} \pmod{p} \Rightarrow \alpha^{x_1 - x_2} \equiv r^{s_1 - s_2} \pmod{p}$$

- Đặt $r = \alpha^k \Rightarrow x_1 - x_2 \equiv k(s_1 - s_2) \pmod{p-1}$
- Đặt $d = \gcd(s_1 - s_2, p-1) \Rightarrow x_1 - x_2 \vdots d$
- Đặt $x' = \frac{x_1 - x_2}{d}, s' = \frac{s_1 - s_2}{d}, p' = \frac{p-1}{d}$, suy ra $x' \equiv ks' \pmod{p'}$
- Vì $\gcd(s', p') = 1$ nên tồn tại $\varepsilon = (s')^{-1} \pmod{p'}$
- Khi đó $k \equiv x'\varepsilon \pmod{p'} \equiv x'\varepsilon + ip' \pmod{p}$
- Với $0 \leq i \leq d-1$, có thể tìm được k nhờ hệ thức $r \equiv \alpha^k \pmod{p}$



1. Chữ ký điện tử

1. 4. Chuẩn chữ ký điện tử

- Năm 1991, NIST đưa ra thuật toán chữ ký điện tử (DSA-Digital Signature Algorithm) → chuẩn chữ ký điện tử (DSS-Digital Signature Standard)
- DSA là một biến thể của thuật toán ElGamal
- DSS sử dụng một khoá công khai để kiểm tra tính toàn vẹn của dữ liệu nhận được và đồng nhất với dữ liệu người gửi
- DSS cũng có thể sử dụng bởi bên thứ ba để xác định tính xác thực của chữ ký và dữ liệu trong nó
- Nói cách khác, một bức điện được kí đảm nhiệm chức năng như một văn bản hợp pháp (chẳng hạn như các bản hợp đồng) → cần thiết xác minh chữ kí sau rất nhiều năm bức điện được kí
- ElGamal không đảm bảo được điều này vì cần một giá trị lớn modulo p (ít nhất 512/1024 bit nhằm chống lại việc giả mạo trong tương lai)
- DSS đã sửa đổi ElGamal theo cách khéo léo: mỗi bức điện 160 bit sử dụng chữ kí 320 bit nhưng việc tính toán được thực hiện với số 512 bit p



Một số thay đổi trên DSA:

- $s = (x + \alpha \cdot r)k^{-1} \pmod{p-1}$
- $\alpha^x \cdot \beta^r \equiv r^s \pmod{p}$
- nếu $\gcd(x + \alpha \cdot r, p - 1) = 1$ thì tồn tại $s^{-1} \pmod{p-1} \Rightarrow \alpha^{xs^{-1}} \beta^{rs^{-1}} \equiv r \pmod{p}$

Sơ đồ thuật toán DSA:

- Cho p là số nguyên tố 512 bit, α là căn nguyên thuỷ, q là số nguyên tố 160 bit và q là ước của $p - 1$
- Chọn số bí mật $a < p$ rồi tính $\beta = \alpha^a \pmod{p} \rightarrow$ công khai (p, q, α, β)
- Với khoá $K = (p, \alpha, a, \beta)$ và bản rõ x , chọn ngẫu nhiên số $1 \leq k \leq q - 1$:

$$(r, s) = \text{sig}_K(x, k), \quad \begin{cases} r = (\alpha^k \pmod{p}) \pmod{q} \\ s = (x + a \cdot r) \cdot k^{-1} \pmod{q} \end{cases}$$

$$\text{ver}_K(x, r, s) = \text{True} \Leftrightarrow (\alpha^{u_1} \beta^{u_2} \pmod{p}) \pmod{q} \equiv r, \quad \begin{cases} u_1 = xs^{-1} \pmod{q} \\ u_2 = rs^{-1} \pmod{q} \end{cases}$$

1. Chữ ký điện tử

1. 4. Chuẩn chữ kí điện tử

Ví dụ 1.3

- Chọn $q = 101$ và $p = 78 \cdot q + 1 = 7879$, $\alpha = 170$ là căn nguyên thuỷ theo $\mod p$
- Chọn $a = 75 \Rightarrow \beta = \alpha^a \mod p = 4567$
- Alice muốn kí lên bức điện $x = 1234$, chọn ngẫu nhiên số $k = 50(k^{-1} \mod q = 99)$
- Tính
$$\begin{cases} r = (170^{50} \mod 7879) \mod 101 = 2518 \mod 101 = 94 \\ s = (1234 + 75 \cdot 94) \cdot 99 \mod 101 = 97 \end{cases}$$

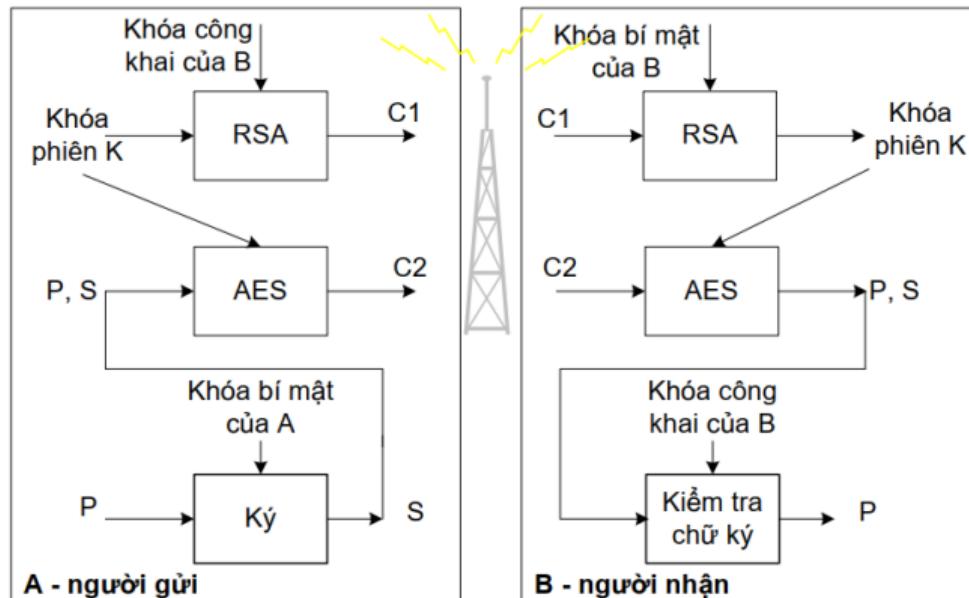
$\Rightarrow (94, 97)$ là chữ kí điện tử của $x = 1234$
- Xác thực:
$$\begin{cases} s^{-1} = 97^{-1} \mod 101 = 25 \\ u_1 = 1234 \cdot 25 \mod 101 = 45 \\ u_2 = 94 \cdot 25 \mod 101 = 27 \end{cases}$$

$\Rightarrow (170^{45}4567^{27} \mod 7879) \mod 101 = 2518 \mod 101 = 94$



1. Chữ ký điện tử

- Khác với chữ ký thông thường, chữ kí số là một thông tin ở dạng số hoá được tạo ra từ văn bản và **không phải** là một phần của văn bản
- Sau khi được tạo ra, chữ kí số sẽ được gửi đi cùng thông điệp





Nội dung

Chữ ký điện tử

- 1.1 Mở đầu
- 1.2 Sơ đồ chữ kí RSA
- 1.3 Sơ đồ chữ kí ElGamal
- 1.4 Chuẩn chữ kí điện tử
- 1.5 Mô hình ứng dụng của chữ kí số

Hàm băm

- 2.1 Giới thiệu
- 2.2 Thuật toán MD5
- 2.3 Thuật toán SHA

Trao đổi



2. Hàm băm

2. 1. Giới thiệu

- Các hệ chữ kí ở trên chỉ cho phép kí các bức điện ngắn, ví dụ trong DSS: bức điện 160 bit được kí với 320 bit → **bức điện lớn megabyte ???**
- Cách giải quyết đơn giản: chia bức điện lớn thành các đoạn nhỏ 160 bit rồi kí lên mỗi đoạn đó (giống mã hoá)
- **Nhược điểm:**
 - với thông điệp có kích thước a thì kích thước chữ kí là $2a$ (DSS) → dung lượng truyền đi rất lớn
 - với chữ kí an toàn thì tốc độ rất chậm vì dùng nhiều phép tính số học phức tạp (môđulo)
 - quá nhiều đoạn được kí → khi sắp xếp lại có thể bị xáo trộn hoặc mất mát → mất tính toàn vẹn
- Hàm băm (Hash function) có thể giải quyết các rắc rối trên



2. Hàm băm

2. 1. Giới thiệu

Định nghĩa 2.1

Hàm băm h sẽ lấy đầu vào là bức điện x có độ dài bất kì và sinh kết quả là một chuỗi có độ dài cố định, được gọi là "cốt" của bức điện (message digest) hay giá trị băm (hash value).

Ví dụ 2.1

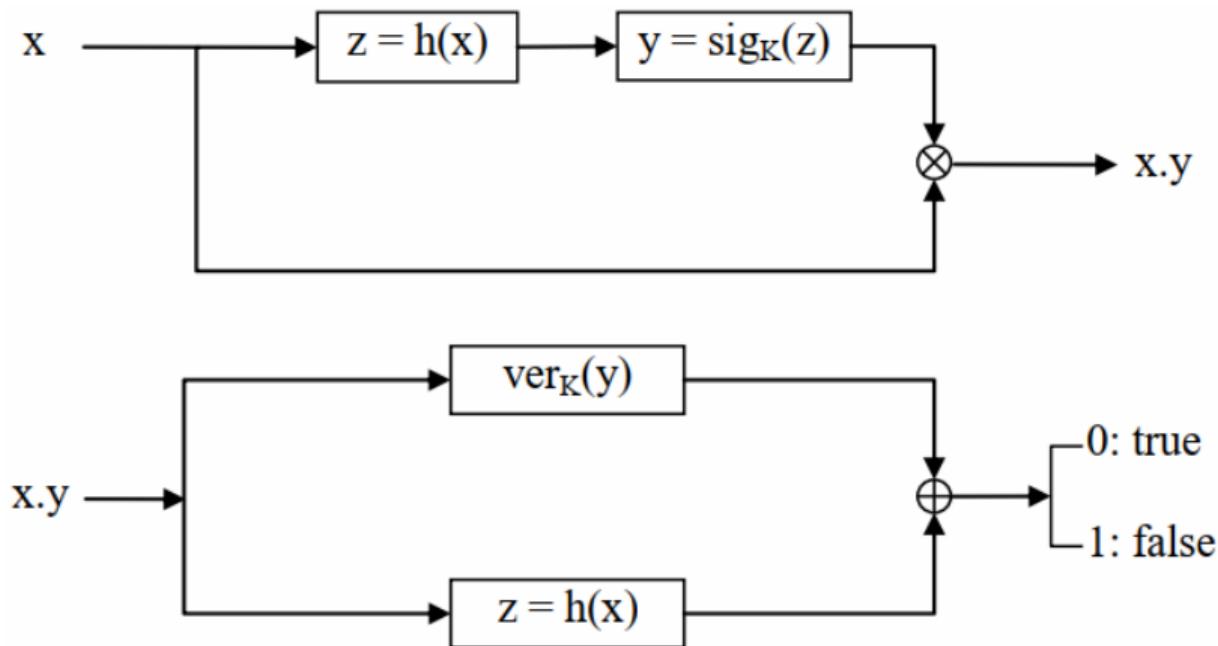
Alice muốn kí một bức điện x (độ dài bất kì).

- tính cốt của bức điện $z = h(x)$ (độ dài cố định)
- kí $y = \text{sig}_K(z)$
- gửi đi cặp (x, y)
- xác minh bằng việc tính lại $z = h(x)$ và $\text{ver}_K(z, y) = \text{True}$?



2. Hàm băm

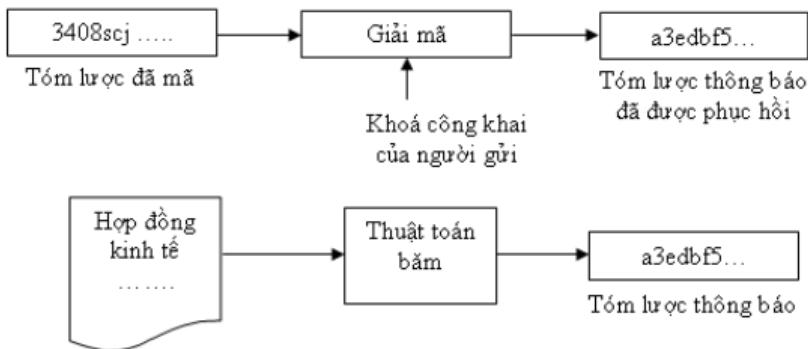
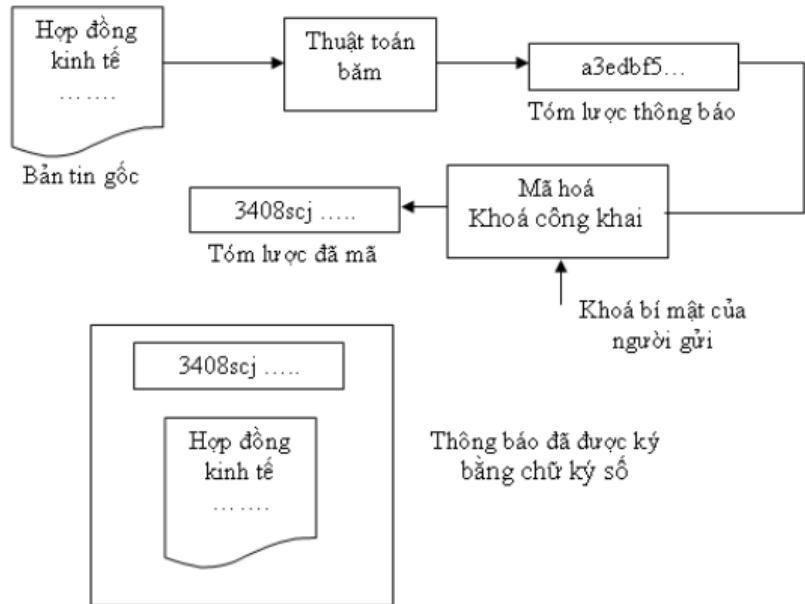
2. 1. Giới thiệu





2. Hàm băm

2. 1. Giới thiệu





2. Hàm băm

2. 1. Giới thiệu

Đặc trưng hàm băm

- Với thông điệp đầu vào x thu được bản băm $z = h(x)$ duy nhất
- Với 2 thông điệp $x \neq x' \Rightarrow h(x) \neq h(x')$
- h là hàm một chiều: với thông điệp x cho trước, dễ dàng tính được $z = h(x)$ nhưng rất khó tìm x nếu biết z



2. Hàm băm

2. 1. Giới thiệu

Nghịch lý ngày sinh nhật

- **Câu hỏi:** Trong lớp có ít nhất bao nhiêu sinh viên, để xác suất có ít nhất 2 sinh viên trùng ngày sinh nhật là lớn hơn 0.5 ?
- Gọi số sinh viên ít nhất trong lớp là $k \rightarrow$ xác suất q để không có 2 người nào trùng ngày sinh là $q = \frac{C_{365}^k}{365^k}$
- Xác suất p để có ít nhất 2 người trùng ngày sinh là: $p = 1 - q = 1 - \frac{C_{365}^k}{365^k}$
- $p > 0.5 \Rightarrow k > 22$ hay $k = 23$ ($p = 0.5073$) \rightarrow **nghịch lý ngày sinh nhật**

Xác suất để hai mẫu tin có cùng bản Hash là không nhỏ như chúng ta nghĩ!



2. Hàm băm

2. 1. Giới thiệu

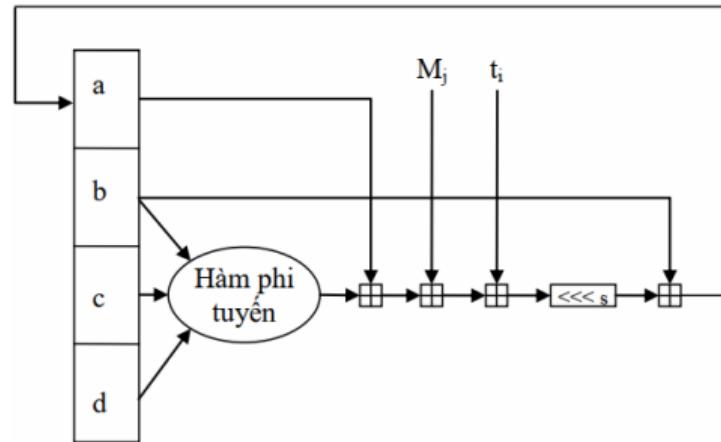
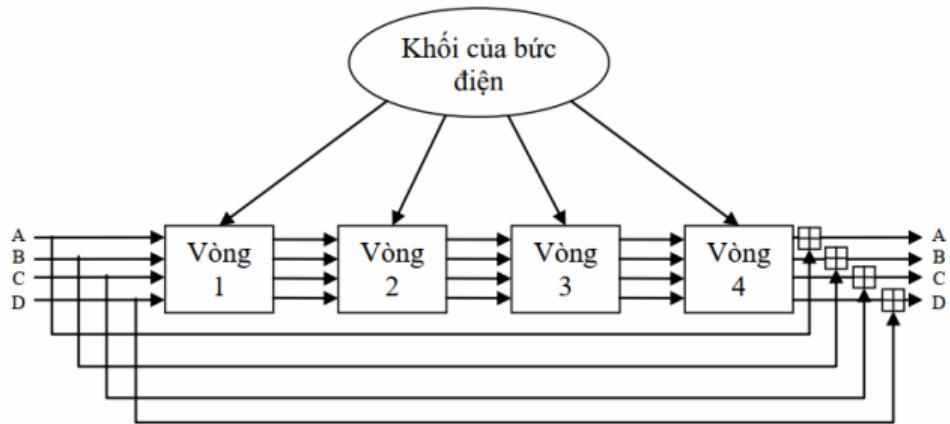
- **Tổng quát:** giả sử hàm băm có n giá trị băm khác nhau, nếu có k giá trị băm từ k thông tin khác nhau được chọn ngẫu nhiên thì điều kiện để xác xuất có ít nhất 2 giá trị băm trùng nhau lớn hơn ϵ là $k \approx 1.1774\sqrt{n}$
- Một hàm băm 40 bit sẽ là không an toàn vì chỉ cần thử 2^{20} phép thử đã có xác suất đúng độ là 50%. Tương tự với hàm băm 64/128 bit.
- **Birthday attack:**
 - Kẻ thám mã tạo ra $2^{m/2}$ biến thể của mẫu tin đúng, mà tất cả đều có bản chất ngữ nghĩa như nhau, với m ở đây là độ dài của bản mã hash
 - Kẻ thám mã cũng có thể tạo ra $2^{m/2}$ biến thể khác nhau của mẫu tin lừa dối, tức là có ngữ nghĩa ngược lại
 - Hai tập tin được so sánh với nhau để tìm cặp có cùng bản hash (xác suất lớn hơn hoặc bằng 0,5)
 - Người dùng ký vào mẫu tin đúng, sau đó bị thay thế bằng mẫu tin giả mà cũng có chữ ký đúng.
- Một số thuật toán băm nổi tiếng: **MD5 (Message Digest), SHA (Secure Hash**



2. Hàm băm

2. 2. Thuật toán MD5

- **Input:** khối 512 bit (chia thành 16 khối con 32 bit)
- **Ouput:** 4 khối 32 bit (= 128 bit)
- **A = 0x01234567; B = 0x89abcdef; C = 0xfedcba98; D = 0x76543210**

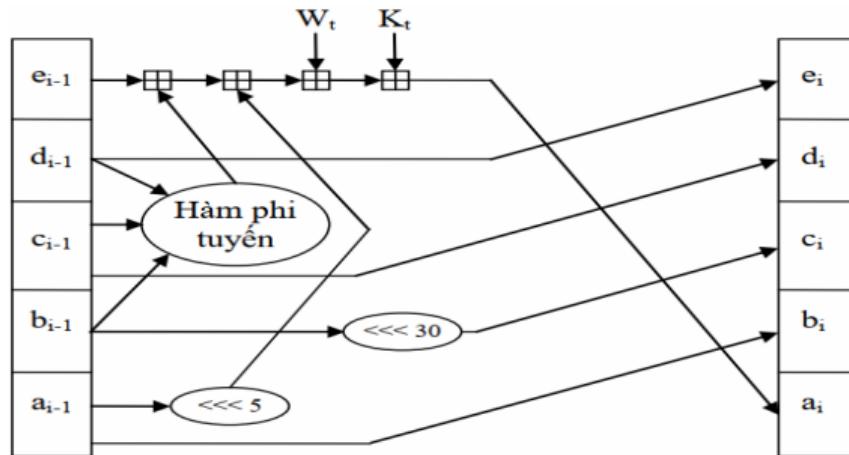




2. Hàm băm

2. 3. Thuật toán SHA

- NIST sử dụng cho chuẩn chữ kí DSS (tương tự MD5)
- **Input:** khối 512 bit (chia thành 16 khối con 32 bit)
- **Ouput:** 5 khối 32 bit (= 160 bit)
- $A = 0x67452301; B = 0xefcdab89; C = 0x98badcfe; D = 0x10325476; E = 0xc3d2e1f0$





① Sinh khoá DSA:

- Chọn số nguyên tố lớn $p = 2^L$, $L = 512/1024$ bit và là bội của 64, q là số nguyên tố 160 bit và là ước của $p - 1$
- Chọn $h < p - 1$ và $h^{\frac{p-1}{q}} \pmod{p} > 1 \rightarrow g = h^{\frac{p-1}{q}} \pmod{p}$
- Chọn khoá bí mật $x < q$ và tính khoá công khai $y = g^x \pmod{p}$

② Tạo chữ kí DSA: cho mẫu tin M

- Alice chọn khoá ngẫu nhiên $k < p$ (xoá sau khi dùng và không bao giờ dùng lại)
- Tính cặp giá trị $\begin{cases} r = (g^k \pmod{p}) \pmod{q} \\ s = (SHA(M) + xr) k^{-1} \pmod{q} \end{cases}$
- Gửi cặp (r, s) cùng với mẫu tin M cho Bob.

③ Xác minh chữ kí DSA:

- Bob cần tính

$$w = s^{-1} \pmod{q} \rightarrow \begin{cases} u_1 = SHA(M) \cdot w \pmod{q} \\ u_2 = r \cdot w \pmod{q} \end{cases} \rightarrow v = (g^{u_1} \cdot y^{u_2} \pmod{p}) \pmod{q}$$

- Nếu $v = r$ thì chữ kí đúng là của Alice



Ví dụ 2.2 (Chữ ký điện tử DSA)

① Sinh khoá DSA:

- Cho $p = 23, q = 11, h = 7 \rightarrow g = h^2 \pmod{23} = 3$
- Chọn khoá bí mật $x = 5$ và tính khoá công khai $y = 3^5 \pmod{23} = 13$

② Tạo chữ ký DSA: cho mẫu tin M có $H(M) = 9$

- Alice chọn khoá ngẫu nhiên $k = 6$
- Tính cặp giá trị $\begin{cases} r = (g^k \pmod{p}) \pmod{q} = (3^6 \pmod{23}) \pmod{11} = 5 \\ s = (H(M) + xr) k^{-1} \pmod{q} = (9 + 5 \cdot 5) \cdot 6^{-1} \pmod{11} = 2 \end{cases}$
- Alice gửi $(5, 2)$ cùng với mẫu tin M cho Bob.

③ Xác minh chữ ký DSA:

- Bob tính $w = s^{-1} \pmod{q} = 2^{-1} \pmod{11} = 6$
 $\Rightarrow \begin{cases} u_1 = H(M) \cdot w \pmod{q} = (9 \cdot 6) \pmod{11} = 10 \\ u_2 = r \cdot w \pmod{q} = (5 \cdot 6) \pmod{11} = 8 \end{cases}$
 $\Rightarrow v = (g^{u_1} \cdot y^{u_2} \pmod{p}) \pmod{q} = (3^{10} \cdot 13^8 \pmod{23}) \pmod{11} = 5$
- $v = r = 5 \rightarrow$ chữ ký đúng của Alice



Nội dung

Chữ ký điện tử

- 1.1 Mở đầu
- 1.2 Sơ đồ chữ kí RSA
- 1.3 Sơ đồ chữ kí ElGamal
- 1.4 Chuẩn chữ kí điện tử
- 1.5 Mô hình ứng dụng của chữ kí số

Hàm băm

- 2.1 Giới thiệu
- 2.2 Thuật toán MD5
- 2.3 Thuật toán SHA

Trao đổi



TRAO ĐỔI