**2021-2022 HK 1 Mật mã và An toàn dữ liệu**

**Nhật ký giảng dạy**

**Lịch trình giảng dạy dự kiến**

| **Tuần** | **Nội dung** | **Ghi chú** | **Kiến thức**  **chuẩn bị** |
| --- | --- | --- | --- |
| 1  14-15/9 | GV: Phó Đức Tài  -Giới thiệu môn học  -Bổ sung kiến thức về số học và trường hữu hạn GF(p)  -Giới thiệu SageMath | Bài tập cá nhân |  |
| 2  21-22/9 | GV: Võ Tùng Linh  - Định nghĩa hàm băm và các ví dụ đơn giản.  - Các yêu cầu an toàn đối với hàm băm.  - Một số tấn công cơ bản vào hàm băm. | Bài tập cá nhân |  |
| 3  28-29/9 | GV: Võ Tùng Linh, Phó Đức Tài  - Một số cấu trúc đơn giản để xây dựng hàm băm.  - Cấu trúc Merkle-Damgard để xây dựng hàm băm.  - Trường hữu hạn GF() & Thực hành SageMath | Bài tập cá nhân |  |
| 4 | GV: Võ Tùng Linh, Phó Đức Tài  - Hàm băm SHA-512.  - Thực hành SageMath | Bài tập cá nhân | Tính toán số học trong trường hữu hạn. |
| 5 | GV: Võ Tùng Linh, Phó Đức Tài  - Hàm băm SHA-512 (tiếp).  - Một số ứng dụng của hàm băm.  - Thực hành SageMath | **Bài tập nhóm số 1** | Tính toán số học trong trường hữu hạn. |
| 6 | GV: Võ Tùng Linh, Phó Đức Tài  -  - Thực hành SageMath | Bài tập cá nhân |  |
| 7 | GV: Võ Tùng Linh, Phó Đức Tài  -  - Thực hành SageMath | Bài tập cá nhân |  |
| 8 | GV: Võ Tùng Linh, Phó Đức Tài  -  - **Kiểm tra giữa kỳ** | Bài tập cá nhân |  |
| 9 | GV: Võ Tùng Linh, Phó Đức Tài  -  - Thực hành SageMath | **Bài tập nhóm số 2** |  |
| 10 | GV: Võ Tùng Linh, Phó Đức Tài  -  - Thực hành SageMath | Bài tập cá nhân |  |
| 11 | GV: Võ Tùng Linh, Phó Đức Tài  -  -Thực hành SageMath | Bài tập cá nhân |  |
| 12 | GV: Võ Tùng Linh, Phó Đức Tài  -  -QR code phần 1: Lý thuyết & thuật toán | Bài tập cá nhân |  |
| 13 | GV: Võ Tùng Linh, Phó Đức Tài  -QR code phần 2: Thực hành với SageMath | **Bài tập nhóm số 3** |  |
| 14 | GV: Võ Tùng Linh, Phó Đức Tài  -  -QR code phần 3: Bảo mật trong QR code | Bài tập cá nhân |  |
| 15 | GV: Võ Tùng Linh, Phó Đức Tài  - Giới thiệu nhanh về ECDSA  - Giới thiệu nhanh về blockchain  - Ôn tập cuối kỳ | **Ôn tập thi cuối kỳ** |  |

**Đăng ký danh sách nhóm, mỗi nhóm tối đa 3 sinh viên:**

Tên nhóm trưởng để lên đầu, hạn đăng ký cuối tuần thứ 2

| **Nhóm** | **Các thành viên trong nhóm** | **Ghi chú** |
| --- | --- | --- |
| 1 | **Phạm Minh Tiến**, Nguyễn Xuân Thành, Phạm Thị Diệu Linh |  |
| 2 | **Nguyễn Long Hải**, Phí Linh Chi, Đào Tấn Dũng |  |
| 3 | **Trần Thế Hùng,** Trần Hoàng Đức, Bùi Long Giang |  |
| 4 | **Đỗ Duy Đạo**, Nguyễn Khánh Hoà, Lê Huy |  |
| 5 | **Nguyễn Trang Nhung**, Nguyễn Minh Kiên, Lành Trường Giang |  |
| 6 | **Hoàng Trung Kiên**, Nguyễn Lê Bình |  |
| 7 | **Nguyễn Thị Hà**, Đặng Hà My |  |
| 8 | **Nguyễn Xuân Duẩn**, Trần Đức Thịnh, Phạm Anh Tuyến |  |
| 9 | **Nguyễn Thành Vinh**, Nguyễn Văn Thạo, Mai Vũ Vĩnh Tiến |  |
| 10 | **Nguyễn Đỗ Hoàng Quân**, Phạm Ngọc Hoàng Long, Nguyễn Trung Khánh |  |
| 11 | **Trần Quang Thụy**, Hà Quốc Việt, Đỗ Hữu Quang |  |
| 12 | **Phạm Vũ Anh Quân** |  |

**Ngày 14/9/2021 (Tuần 1, buổi 1): GV Phó Đức Tài**

Thuật toán chia Euclid (tìm ƯCLN của 2 số nguyên)

**Ví dụ:** Tìm ƯCLN của 2021 và 14

Bước 1: 2021 chia 14

2021 = 144 x 14 + 5

Do đó

ƯCLN(2021,14) =ƯCLN(144 x 14 + 5,14) = ƯCLN(5,14) = ƯCLN(14,5)

(Tính chất: ƯCLN(a+ nxb , b) = ƯCLN(a,b))

Bước 2: 14 chia 5

14 = 2 x 5 + 4

Do đó

ƯCLN(14,5) = ƯCLN(2 x 5 + 4,5) = ƯCLN(4,5) = ƯCLN(5,4)

Bước 3: 5 chia 4

5 = 1 x 4 + 1

Do đó

ƯCLN(5,4) = ƯCLN(1 x 4 + 1,4) = ƯCLN(1,4) = ƯCLN(4,1)

Bước 4: ƯCLN(4,1) =?

4 chia 1 dư 0, suy ra ƯCLN(4,1) = 1.

(Tính chất: Giá sử a và b là hai số nguyên dương:

Nếu a chia hết cho b, thì ƯCLN(a,b) = b)

Bài tập về nhà: Viết chương trình máy tính (Java / Python / C) thực hiện thuật toán chia Euclid để tìm ƯCLN.

Mai: Chữa BT, bằng phần mềm SageMath (miễn phí, mã nguồn mở, viết dựa trên Python, có thể nhúng được nhiều phần mềm khác vào). (Phần mềm có phí Maple/Matlab/… ?)

Đọc tiếp mod

-11 mod 7 = ?

-11 chia cho 7 được

-11 = (-2) x 7 + 3 (Cần 0 <= r < b)

**Bài tập 2:** Cộng và nhân modulo. Viết chương trình máy tính hoặc lấy ví dụ bằng số lớn cụ thể.

Cho buổi học ngày mai:

**Hướng dẫn cài SageMath:** Xem youtube video hoặc theo các bước: Video giới thiệu phần mềm SageMath:

* Cài đặt (Xem video clip:<https://www.youtube.com/watch?v=QpNJcpKfiwo>);
* Giới thiệu nhanh (Xem video clip: ttps://www.youtube.com/watch?v=9CG2YQxEIcs)

**Hướng dẫn cài Sage trên win:**

1. Download binary tại: <https://github.com/sagemath/sage-windows/releases>

2. Cài lên máy.

3. Trên desktop có 3 icon: Dùng "SageMath Notebook" (hai cái kia không có giao diện tốt).

4. Có thể xuất hiện:

To access the notebook, open this file in a browser:

file:///home/sage/.local/share/jupyter/runtime/nbserver-759-open.html

Or copy and paste one of these URLs:

<http://localhost:8888/?token=af1b5cd024f911c52aa8ca91ae38d6a2043c024d734ec44e>

thì mở chương trình web browser ra, dán địa chỉ

<http://localhost:8888/?token=af1b5cd024f911c52aa8ca91ae38d6a2043c024d734ec44e>

vào là ra giao diện jupyter.

Cài trên linux hay IOS: Xem hướng dẫn trong trang web của SageMath.

**Ngày 15/9/2021 (Tuần 1, buổi 2): GV Phó Đức Tài**

**Tiết 1: Giới thiệu phần mềm SageMath**

**Mở SageMath như thế nào?**

**Cách 1:** Tính toán như máy tính bỏ túi: http://sagecell.sagemath.org/

Nhập dòng lệnh vào trong cell (ô lệnh), thực hiện bằng cách bấm tổ hợp phím Shift-Enter hoặc click chuột Evaluate.

**Ví dụ:**

1. gcd(12,15) cho ra kết quả 3

b) xgcd(12,15) cho ra kết quả (3,-1,1)

xgcd(a,b) cho ra (d,m,n) với d=ƯCLN(a,b), m và n là 2 số nguyên thỏa mãn

d=am+bn.

Ví dụ: a) 3 = 12(-1) + 15(1).

1. xgcd(60,90)=(30,-1,1)
2. xgcd(1956,2021)=(1, 342, -331)

**Cách 2:** Sử dụng SageMath trên đám mây, Cocalc (<https://cocalc.com/>). Cần tạo tài khoản.

**Cách 3:** (Khuyên dùng) Cài SageMath lên PC. Mở SageMath Notebook

**Ví dụ:** Thuật toán Euclid

Input: 2 số nguyên a và b

Output: ƯCLN(a,b)

Nhắc lại: Trong SageMath có một số lệnh sau: Với a và n là hai số nguyên, n>0.

a//n và a%r cho thương và số dư của phép chia a cho n.

def định nghĩa (**de**finition a **f**unction) hàm trong SageMath (như Python)

Ký hiệu: != (khác), == (bằng) toán tử so sánh

def euclide(a,b):

r=a%b

print (a,b,r)

while r != 0:

a=b; b=r

r=a%b

print(a,b,r)

if r==0:

print('ƯCLN =',b)

Chẳng hạn chạy euclide(12,15) cho ta kết quả

12 15 12

15 12 3

12 3 0

ƯCLN = 3

**Tính toán gần đúng:**

numerical\_approx(20.0/14,digits=20) cho số 20.0/14 với 20 chữ số thập phân

1.4285714285714285714

numerical\_approx(pi,digits=1000) cho số Pi với 1000 chữ số thập phân

Xem các ví dụ ở trên trong file: EuclideanAlgorithm.ipynb (trong thư mục của lớp).

**Bài tập về nhà:** Chạy tính toán thực hành file SageMathGuideTour.ipynb cũng có trong thư mục này!

**Tiết 2: Trường hữu hạn (Chương 5)**

1. **Nhóm**

Ví dụ: Nghịch đảo

1. Phép toán +, e=0.

Nghịch đảo của a bằng -a

1. Phép toán x, e=1

Nghịch đảo của a bằng 1/a

G=R (tập số thực)

* Là một nhóm

X Không phải là một nhóm, vì a=0 thì không tồn tại nghịch đảo nhân.

Câu hỏi: Phép lũy thừa? (Kiểm tra: (a^b)^c = a^(b^c) ? SAI, chẳng hạn

(2^3)^5 << 2^(3^5).)

Lũy thừa: a^n = a.a. … .a (n lần)

X 2^3 = 2x2x2 = 8

* 2^3 = 2+2+2 = 6

Ví dụ khác: Z10={0,1,...,9}

+ nghĩa là + mod10 thì 2^3 = 6, 2^5=0, 6^3=8

cụ thể: 6^3 = 6 + 6 + 6 = 2 + 6 = 8

x nghĩa là x mod10 thì 2^3 = 8, 2^5=2 , 6^3=6

cụ thể: 6^3 = 6 x 6 x 6 = 6 x 6 = 6

2^2021 = 2 ???

(2^1=2, 2^2= 4, 2^3=8, 2^4=6, 2^5=2) Gợi ý: 2^(4k+r) = 2^r,

với r=0,1,2,3.

Suy ra: 2^2021 = 2^(4.k+1)=2^1=2.

**Nhóm xyclic:**

Ví dụ (**Z**,+) là một nhóm xyclic, với a (phần tử sinh) bằng 1 (hoặc -1)

(x^n = x+x+...+x)

1^5 = 1+1+ … + 1 (5 số 1) =5

2021 = 1^2021

-3 = 1^(-3) được hiểu thế nào?

Quy ước: Nếu m<0 thì 1^m = nghich đảo của 1^(-m)

1^(-3) = -(1^3) = -3

1. **Vành & miền nguyên**

R là một vành, ký hiệu 0 là đơn vị của phép +, 1 là đơn vị của phép nhân (nếu có).

R không có ước của 0 nghĩa là gì?

a x b = 0 thì hoặc a = 0 hoặc b =0

**Ví dụ:**

1. (**Z**,+,x) là một miền nguyên.
2. (**R**,+,x) là một miền nguyên.
3. (**Zn**,+,x)

**Zn** = {0,1,2,...,n-1}

+,x là các phép toán +,x modulo n.

a + b = (a+b) mod n

a x b = (axb) mod n

Chẳng hạn, n=6, Z6 = {0,1,2,3,4,5}

Phép cộng

| + | 0 | 1 | 2 | 3 | 4 | 5 |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1 | 2 | 3 | 4 | 5 | 0 |
| 2 | 2 | 3 | 4 | 5 | 0 | 1 |
| 3 | 3 | 4 | 5 | 0 | 1 | 2 |
| 4 | 4 | 5 | 0 | 1 | 2 | 3 |
| 5 | 5 | 0 | 1 | 2 | 3 | 4 |

Phép nhân

| X | 0 | 1 | 2 | 3 | 4 | 5 |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 2 | 3 | 4 | 5 |
| 2 | 0 | 2 | 4 | 0 | 2 | 4 |
| 3 | 0 | 3 | 0 | 3 | 0 | 3 |
| 4 | 0 | 4 | 2 | 0 | 4 | 2 |
| 5 | 0 | 5 | 4 | 3 | 2 | 1 |

Kết luận: Z6 không phải là một miền nguyên, vì nó có ước của 0 (2x3=0 nhưng cả 2 và 3 đều khác 0).

**BTVN:** Kiểm tra xem Z5 có phải là một miền nguyên hay không?

Chứng minh tổng quát, Zn là một miền nguyên khi và chỉ khi n là một số nguyên tố.

1. **Trường**

Miền nguyên, mọi phần tử khác 0 đều có nghich đảo của x

Phép chia?

a/b hiểu a x (nghịch đảo của b)

**BTVN:** Z5, viết các phần tử nghịch đảo của 1,2,3,4.

**Ngày 21/9/2021 (Tuần 2, buổi 1) - GV: Võ Tùng Linh**

Một số nội dung:

- Hàm băm: SHA-512. (Chương 11 của cuốn sách của Stallings).

- Lược đồ chữ ký số: RSA-PSS.

1. Định nghĩa và một vài ví dụ đơn giản

* Tính toàn vẹn dữ liệu: chỉ giúp phát hiện những thay đổi trái phép, không giúp che dấu dữ liệu.
* Định nghĩa hàm băm:
* Độ dài bit (bit length), các phép thao tác trên bit (OR, AND, XOR, dịch trái, dịch phải, dịch vòng,...): ????
* a\_0a\_1…..a\_n XOR b\_0b\_1….b\_n: thực hiện được?

Ví dụ:

1001 XOR 0101 = 1100? 1001 thuộc {0,1}^4

**Wikipedia: Thao tác theo bit, bitwise operators:**

* Tính chất nén: H : {0, 1}^\* → {0, 1}^n , x |--> H(x) = h\_0h\_1...h\_{n-1}.
* H(x): được gọi mã giá trị băm (hash value), mã băm (hash code), bản tóm lược thông điệp (message digest)
* Tiền ảnh (preimage): x, H(x) = y. Khi đó, x được gọi là **tiền ảnh** của giá trị băm y.
* Tính chất kháng tiền ảnh (tính một chiều) = preimage resistance = (one-way): cho trước mã băm y, bạn không thể tính được thông điệp x sao cho H(x) = y.
* Cho trước x, hàm băm H, dễ dàng tính được ngay y =H(x). Ngược lại, là khó.

**Ví dụ 1:** f(x) = x^2.

Cho x → f(x)

Cho trước y, tính được x = \sqrt{y}. Kết luận: f không thoả mãn tính chất kháng tiền ảnh.

* Trong khoá học, hàm băm = hàm băm mật mã.
* Lưu ý: tính duy nhất của giá trị băm là xét đối một hàm băm cụ thể, có nghĩa là với hàm H\_1 thì H\_1(x) là duy nhất. Nếu dùng hàm băm H\_2, thì ta có giá trị băm tương ứng H\_2(x), nói chung H\_1(x) \ne H\_2(x).
* \Sigma = tập các văn bản có thể có = {0,1}^\*
* \Delta = tập các bản tóm lược có thể được sử dụng = {0,1}^n.

**Ví dụ 2:** Xây dựng hàm băm H để băm các thông điệp M có độ dài L bit. Ta muốn tính giá trị băm H(M) có độ dài là n bit.

Thông điệp M được chia thành m khối, mỗi khối gồm có n-bit. Không phải lúc nào L cũng chia hết cho n, cho nên thông thường khối cuối cùng là có độ dài bit là < n. **Do đó khối cuối cùng thường phải được đệm (padding) thêm cho đủ thành n bit.**

| B\_1 | B\_2 | …. | B\_m |
| --- | --- | --- | --- |

bitlength(B\_1) = bitlength(B\_2) = … = bitlength(B\_n) = b.

L = 16, b = 6, ta có sẽ có 3 khối, khối thứ 3 chỉ có 4 bit.

00111, b > k

x\_0x\_1...x\_{k-1}00...0: đệm thêm n - k bit 0., k + n - k = n-bit.

Từ thông điệp M → xây dựng quy tắc để nhận được giá trị H(M)?

Ký hiệu C\_i là bit thứ i trong đầu ra của hàm băm, tức là

H(M) = C\_0C\_1....C\_{n-1}.

Ký hiệu b\_{ij} là bit thứ i trong khối B\_j của thông điệp M.

Từ đây ta tính:

**C\_i = b\_{i1} XOR b\_{i2} XOR … XOR b\_{im}, 0 <= i <= n-1.**

Nén: độ dài L → n << L.

Hiệu quả: dễ dàng tính được.

Suy ra: H là một hàm băm. Tuy nhiên, H không an toàn ??

**BTVN 1**: Xem lại ví dụ tôi vừa trình bày trong sách tham khảo, chú ý phần phân tích tính an toàn của hàm băm vừa xây dựng + ví dụ về cách hàm băm sử dụng phép dịch vòng trong sách.

**Ngày 22/9/2021 (Tuần 2, Buổi 2), GV: Võ Tùng Linh**

**1. Các yêu cầu an toàn đối với hàm băm**

* Tiền ảnh (preimage): thông điệp x, hàm băm H, y = H(x). Khi đó y được gọi giá trị băm (hash value) = mã băm (hash code) = bản tóm lược thông điệp (message digest); x được tiền ảnh của y.
* Tiền ảnh thứ 2: Giả sử bạn có x và giá trị băm y = H(x). Khi đấy, nếu tồn tại x’ \ne x: H(x) = H(x’) thì x’ được gọi là tiền ảnh thứ 2 (second preimage).
* Va chạm (collision): Nếu ta có x \ne x’ mà thoả mãn H(x) = H(x’) thì ta nói hàm băm H có một **va chạm** (xảy ra một va chạm).

+ Hàm băm H cần thoả mãn các yêu cầu sau:

* Kích thước đầu vào tuỳ ý: hàm băm phải áp dụng được trên các khối dữ liệu đầu vào có độ dài tuỳ ý.
* Đầu ra phải có kích thước cố định: Hàm băm phải tạo ra các giá trị băm có kích thước cố định, không phụ thuộc vào độ dài của thông điệp đầu vào.
* Tính hiệu quả: cho đầu vào x, việc tính giá trị băm H(x) phải là dễ dàng thực hiện được.
* Tính kháng tiền ảnh (peimage resistant) = tính chất một chiều (one-way property): cho trước giá trị băm y, bạn không thể tính được thông điệp đầu vào x thoả mãn H(x) = y.

H : {0,1}^\* → {0,1}^n

Không gian đầu vào: vô hạn

Không gian đầu ra: hữu hạn

* Tính kháng tiền ảnh thứ 2 (second preimage resistant) = Tính kháng va chạm yếu (weak collision resistant): Cho trước thông điệp x, việc tìm một thông điệp x’ khác x thoả mãn H(x) = H(x’) là không thể thực hiện được.
* Tính kháng va chạm (collision resistant): Việc tìm được một cặp thông điệp (x, x’) thoả mãn x khác x’ ,H(x) = H(x’) là không thể thực hiện được.

Tính kháng va chạm ⇒ tính kháng tiền ảnh thứ hai.

* Tính giả ngẫu nhiên (Pseudorandomness): Đầu ra của hàm băm H phải vượt qua được các tiêu chuẩn về tính ngẫu nhiên.

+ Hàm băm thoả mãn 5 yêu cầu an toàn đầu tiên thì được gọi là hàm băm yếu (weak hash function); hàm băm thoả mãn 6 yêu cầu đầu tiên thì được gọi là hàm băm mạnh (strong hash function).

+ Mối quan hệ giữa các tính chất an toàn của hàm băm:

* Tính kháng va chạm ⇒ tính kháng tiền ảnh thứ hai. Chiều ngược lại thì chưa hẳn đã đúng.
* Tồn tại những hàm băm kháng va chạm nhưng không kháng tiền ảnh, và ngược lại.
* Tồn tại những hàm băm kháng tiền ảnh thứ 2 nhưng không kháng tiền ảnh, và ngược lại.

+ Các kiểu tấn công cơ bản vào hàm băm: có 2 kiểu tấn công

* Tấn công vét cạn (brute-force attack): là những tấn công không phụ thuộc vào thuật toán cụ thể nào, mà chỉ phụ thuộc độ dài bit.
* Tấn công phân tích mã (cryptanalysis attack): là những tấn công khai thác điểm yếu của những thuật toán mật mã. Đây là kiểu tấn công nguy hiểm.

+ Tấn công vét cạn để tìm tiền ảnh hoặc tìm tiền ảnh thứ 2:

* Giả sử có y là một giá trị băm. Mục tiêu: tìm x sao cho H(x) = y? Tấn công = cách: thử tất cả các giá trị của x cho đến khi tìm được thông điệp có giá trị băm = y.

Lấy thông điệp x\_1, tính H(x\_1) = y\_1, rồi so sánh y\_1 với y. Nếu = ⇒ khẳng định x\_1 là tiền ảnh cần tìm. Nếu không bằng nhau thì tiếp tục thử một thông điệp x\_2 khác,... cho đến khi tìm được thông điệp như mong muốn.

Với hàm băm đầu ra m bit, #{0,1}^m = 2^m. Giá trị băm y sẽ năm trong số 2^m chuỗi bit trong {0,1}^m. Để tìm được x thoả mãn H(x) = y theo phương pháp vét cạn như trên thì số lần thử **trung bình sẽ 2^{m-1}**.

Để chống lại tấn công này thì chỉ cần chọn giá trị m sao cho việc thực hiện 2^{m-1} phép thử này là không thể được.

**BTVN 1**: Tìm hiểu về năng lực tính toán của các máy tính tốt nhất hiện nay (kể máy tính thông dụng lẫn các hệ thống siêu máy tính)?

* Tương tự như trên là tấn công vét cạn để tìm tiền ảnh thứ 2 (xem tài liệu). Đã có x, thủ lần lượt từng x’ \ne x, so sánh H(x) và H(x’). Nếu bằng nhau thì khẳng định là tìm một tiền ảnh thứ 2. Nếu ngược lại, thì tiếp tục với thông điệp x’’,...

+ Tấn công vét cạn để tìm va chạm:

* Lấy cặp thông điệp (x, x’) khác nhau, tính H(x), H(x’) rồi so sánh các giá trị này với nhau. Nếu H(x) = H(x’) thì khẳng định là đã tìm một va chạm. Nếu ngược lại, thì ta tiếp tục với cặp thông điệp khác, cho đến khi tìm được các thông điệp thoả mãn yêu cầu.
* Với hàm băm có đầu ra m bit, thì ta kỳ vọng sau khoảng 2^{m/2} lần thử sẽ tìm được va chạm như mong muốn.
* Tại sao lại có kết quả số lần thử là 2^{m/2}: ta có **Nghịch lý ngày sinh (birthday paradox)**. (Xem slide + tài liệu).

+ Ngưỡng 2^{m/2} được gọi là độ mạnh của hàm băm mà kháng lại các tấn công vét cạn. Do đó ta cần chọn m sao cho giá trị 2^{m/2} vượt qua khả năng vét cạn của bên tấn công.

**Ngày 28/9/2021 (Tuần 3, Buổi 1) GV: Võ Tùng Linh:**

1**. Cấu trúc Merkle - Damgard:**

* Hiện nay có 2 kiểu hàm băm chính:
  + Các hàm băm chuyên dụng: là những thuật toán được thiết kế đặc biệt để băm (xử lý) thông điệp: MD4, MD5, SHA-1, SHA-2, SHA-3,...
  + Hàm băm dựa trên mã khối: Sử dụng mã khối (block cipher) để xây dựng hàm băm: cấu trúc Matayas - Meyer - Oseas,
* Người ta đều sử dụng một cấu trúc lặp để xây dựng hàm băm.
* Cấu trúc lặp để xây dựng phần lớn hàm băm hiện nay được gọi là Cấu trúc Merkle-Damgard.

**Mô tả chi tiết cấu trúc Merkle-Damgard:**

Ký hiệu H là hàm băm cần xây dựng với cấu trúc Merkle-Damgard; M là thông điệp cần băm; ký hiệu H(M) là giá trị băm nhận được.

Việc xây dựng hàm băm H chính là thiết lập quá trình tính toán (cách xử lý) thông điệp M để nhận được H(M).

* L: số lượng khối thông điệp đầu vào. Tức là thông điệp M được chia thành L khối con có độ dài cố định là b, ký hiệu các khối con là Y\_0, Y\_1,..., Y\_{L-1}, |Y\_0| = |Y\_1| = … = |Y\_{L\_1}|. Do không phải khi nào |M| cũng là bội của b, nên nếu cần thiết thì khối cuối cùng Y\_{L\_1} sẽ được đệm thêm bit để đạt được độ dài bit là b.
* b: độ dài của các khối con (khối đầu vào).
* IV (initial value): Giá trị khởi tạo.
* CV (chaining variable): Biến chuỗi; các biến chuỗi có độ dài là n = chính là độ dài (bit) của đầu ra (giá trị băm). Thông thường, b > n.
* CV\_0 = IV; CV\_L = H(M): Giá trị băm.
* f: hàm nén (compress function).
* Hàm băm H sẽ sử dụng lặp đi lặp lại một hàm nén f.
* Hàm nén f nhận 2 đầu vào:
  + Thứ nhất là một đầu vào có độ dài n-bit, là giá trị biến chuỗi CV nhận được từ việc xử lý trước đấy.
  + Thứ hai là đầu vào có độ dài b-bit, là một trong số các khối thông điệp Y\_i, 0 <= i <= L - 1.
* Đầu ra của hàm nén f sẽ là giá trị biến chuỗi CV có độ dài n-bit.
* Quá trình tính toán lặp cụ thể là:
  + CV\_0 = IV
  + CV\_i = f(CV\_{i-1}, Y\_{i-1}), 1 <= i <= L.
  + CV\_L = H(M).
* Hàm nén f có thể là một thuật toán được thiết kế đặc biệt hoặc là hàm mã hoá của một mã khối nào đấy.

**Định lý (Merkle-Damgard):** Nếu hàm nén f là một hàm kháng va chạm thì hàm băm H được xây dựng như trên cũng là một hàm băm kháng va chạm.

**Bài toán đặt ra:** Ta đi xây dựng những hàm nén f tốt, tức là hàm f thoả mãn tính chất kháng va chạm.

Ngược lại, bên tấn công cũng sẽ đi khai thác những “điểm yếu” của hàm nén f ⇒ tấn công vào hàm băm. (tấn công thám mã).

**BTVN:** Đọc lại kỹ phần cấu trúc Merkle-Damgard trong tài liệu.

* Cấu trúc Merkle-Damgard không phải là cấu trúc duy nhất để xây dựng băm. Tồn tại những cấu trúc khác cũng an toàn để xây dựng hàm băm: Cấu trúc Sponge (SHA3);
* Hiện nay nhiều hàm băm được xây dựng với cấu trúc Merkle-Damgard: MD4 (Rivest đưa ra 90) - Rivest = R trong thuật toán RSA (S = Shamir, A = Adleman); MD5 (92), SHA**-0** (NIST 93) ⇒
* Đầu những năm 2000: nhóm nghiên cứu của giáo sư Wang xây dựng được va chạm đối với hàm băm SHA-1 ⇒ không là được xem là an toàn ⇒ không được chấp nhận là chuẩn hàm băm.
* **2002 NIST SHA-2 (SHA-512): Đối tượng mà ta sẽ tìm hiểu chi tiết về cấu trúc**.

------------------------------------------------------------------------------------------------------------

29/9/2021 (Thầy Tài)

Tiết 1: Nhắc lại trường hữu hạn GF(p)

**Thuật toán Euclid mở rộng** xgcd (xem wiki link bên dưới) https://vi.wikipedia.org/wiki/Gi%E1%BA%A3i\_thu%E1%BA%ADt\_Euclid\_m%E1%BB%9F\_r%E1%BB%99ng

xgcd

Input: 2 số nguyên a và b không đồng thời bằng 0

Output: Bộ 3 số nguyên (d,x,y), trong đó d=gcd(a,b) và ax+by=d

xgcd(a,b)=(d,x,y)

**Ứng dụng:** Tìm nghịch đảo của phép nhân trong GF(p) (p: nguyên tố)

Chẳng hạn muốn tìm nghịch đảo của a khác 0 trong GF(p), tức là a thuộc {1,2,...,p-1}. Rõ ràng gcd(a,p)=1.

Bước 1: Thực hiện xgcd(a,p) cho ra kết quả (1,m,n) (do gcd(a,p)=1). Nghĩa là

1 = a.m + p.n

Lấy modulo p hai vế, ta có: a.m = 1. Vì vậy a.m = 1.

Bước 2: Lấy b = m mod p thì b là nghịch đảo của a.

Ví dụ: Ng Xuân Thành’s case dd/mm = 26/1, a= dd mod 7 = 26 mod 7 =5, b =1

Câu 1: Z7

1. a+b = 5+1=6
2. a.b = 5.1=5
3. Nghịch đảo cộng của a là 2, của b là 6.

Nghịch đảo nhân của a là 3, của b là 1.

(Thanh thử: 5x1, 5x2, 5x3 … cho đến khi có kết quả = 1)

Cách 2: Dùng xgcd

Bước 1: xgcd(a,p) =xgcd(5,7) = (1,3,-2), m=3

Bước 2: m mod 7 = 3. Vậy 3 là nghịch đảo của a=5

Câu 2: Z10

c) Nghịch đảo x: của a không tồn tại, của 1 thì chính nó.

Câu hỏi thêm, nghịch đảo của 7

(thử 7x1, 7x2, … có 7x3 = 1 (mod 10))

BTVN: Làm lại câu hỏi trong Mini Test trên Zn với 2 trường hợp sau:

1. n=2027
2. n = số nguyên tố nhỏ nhất > dd^mm (dd/mm là sinh nhật của bạn)

Gợi ý:

Bước 1: xgcd(5,2027) = (1, 811, -2)

Bước 2: 811 mod 2027 = 811

Đáp số: 811

Thử lại: 5 x 811 mod 2027

b) Quang: dd/mm=12/12, nên m = next\_prime(12^12) cho 8916100448291

Tiết 2: GF(p^n) (NỢ sang Tiết 1 tuần sau)

GF(2^n) ????

GF(2^2): 4 phần tử.

Liệu {0,1,2,3} với phép toán +,x mod 4 có lập thành một trường hay không?

| A x B | 0 | 1 | 2 | 3 |
| --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 2 | 3 |
| 2 | 0 | 2 | 0 | 2 |
| 3 | 0 | 3 | 2 | 1 |

Nhận xét: Phần tử 2 không có nghịch đảo nhân

| 2 | 0 | 2 | 0 | 2 |
| --- | --- | --- | --- | --- |

(không có số 1 xuất hiện).

Vì vậy {0,1,2,3} với 2 phép toán ở trên không lập thành một trường.

**Tổng quát:** **Zn**={0,1,...,n-1}, với 2 phép toán cộng & nhân mod n, là một trường khi và chỉ khi n là một số nguyên tố.

(**Giải thích:** n=p nguyên tố. Hãy chứng minh, với mọi a thuộc Zp={0,1,...,p-1}, thì luôn tồn tại nghịch đảo nhân (tức là tồn tại b, sao cho a.b=1 mod p) !

Thật vậy, xét tập hợp A={a.1,a.2,...,a.(p-1)} có (p-1) phần tử, nhưng khi chia cho p thì:

* Không có phần tử nào chia hết cho p.
* Không có 2 số dư nào giống nhau, vì nếu ngược lại a.x = a.y (mod p) thì a.|x-y| chia hết cho p, dẫn đến vô lý.

Vì vậy tập hợp các số dư của các phần tử của A chia cho p thì chính bằng {1,2,...,p-1}. Vậy có b, a.b=1 (mod p).

n=m.k (hợp số, m và k>1).

Ta sẽ chỉ ra Zn không phải là một trường, bằng cách chứng minh m không có nghịch đảo nhân.

Giả sử m có nghịch đảo nhân là s, nghĩa là m.s=1 (mod n), suy ra

m.s-1 chia hết cho n (=m.k), suy ra m.s-1 chia hết cho m (vô lý) !!! )

GF(2^2) = GF(4) ????

Tập hợp: GF(2^n) tương ứng 1-1 với Tập hợp K các đa thức bậc <n, hệ số thuộc GF(2), tương ứng 1-1 với các xâu nhị phân độ dài n bits.

K = {a\_{n-1}x^{n-1}+ … + a1x + a0 | ai hoặc 0 hoặc 1}

|K| = 2^n

Chẳng hạn, n=2: K = {x+1,x,1,0} có 4 phần tử.

n=3, K={x^2+x+1,x^2+x,x^2+1,x^2,x+1,x,1,0} có 8 phần tử.

Xác định phép +, x trên các xâu n bits này như thế nào? (Gợi ý: Dựa vào phép toán đa thức trong Z2[x], tương tự như trong Zp thực hiện cộng và nhân mod p nguyên tố, thực hiện cộng và nhân mod một đa thức f bất khả qui bậc n.).

Đa thức bất khả qui? Là đa thức không phân tích được thành tích của 2 đa thức bậc nhỏ hơn, ví dụ x^2+1 bất khả qui trong Z[x], Q[x], R[x], nhưng không bất khả qui trong Z2[x], vì:

x^2 + 1 = (x+1)^2 (trong Z2)

**Ví dụ:** n=2, đa thức bất khả qui bậc n-1=1, chẳng hạn x+1 hoặc x.

n=3, đa thức bất khả qui bậc n-1=2, x^2+x+1.

Nếu x^2+x+1 = (x+a).(x+b) thì có hệ

a+b=1 (mod 2), a.b=1 (mod 1)

(VÔ NGHIỆM)

GF(2^2) có 4 phần tử {00,01,10,11}, tương ứng {0,1,x,x+1}

Phép cộng mod x^2+x+1

| f+g | 0(00) | 1(01) | x(10) | x+1(11) |
| --- | --- | --- | --- | --- |
| 0 (00) | 0 | 1 | x | x+1 |
| 1 (01) | 1 | 0 | x+1 | x |
| x(10) | x | x+1 | 0 | 1 |
| x+1(11) | x+1 | x | 1 | 0 |

Phép nhân mod x^2+x+1

| f+g | 0(00) | 1(01) | x(10) | x+1(11) |
| --- | --- | --- | --- | --- |
| 0 (00) | 0 | 0 | 0 | 0 |
| 1 (01) | 0 | 1 | x | x+1 |
| x(10) | 0 | x | x+1 | 1 |
| x+1(11) | 0 | x+1 | 1 | x |

Có nghịch đảo nhân đối với mọi phần tử khác 0 !!!

Bài kiểm tra 20 phút: GF(2^4) trường có 16 phần tử

Task1: Tìm một đa thức f bậc 4 bất khả qui trong Z2[x].

Task 2: Thực hiện 2 phép toán F+G và FxG mod f.

Với F và G xác định như sau:

Bạn có ngày sinh dd/mm thì

F = x^3 + (dd mod 2) x^2 + (mm mod 2) x + 1 và

G = x^2 + x + 1.

BTVN: Tìm được nghịch đảo nhân của F và G. (Tuần sau).

**Ngày 06/10/2021 (Tuần 4, Thứ 4), GV: Võ Tùng Linh**

**Hàm băm SHA-512**

* Thuộc họ hàm băm SHA-2, có độ dài đầu ra (giá trị băm) là 512 bit.
* Hàm băm SHA-2 được NIST ban hành thành chuẩn mật mã năm 2002, tên chuẩn là FIPS 180-2: SHA-256, SHA-384 và SHA-512.
* Cấu trúc là tương tự với hàm băm SHA-1, tuy nhiên có độ an toàn cao hơn nhiều.
* Ký hiệu thông điệp là M, độ dài là L, hàm băm H.

**Mô tả chi tiết về hàm băm SHA-512:**

* Hàm băm SHA-512 nhận đầu vào là các thông điệp có độ dài không vượt quá 2^128 bit.
* Độ dài đầu ra: 512 bit (2^10).
* Thông điệp đầu vào sẽ được xử lý theo từng khối 1024 bit (2^11): |M| = 2^20, số khối con sẽ là 2^20/2^11 = 2^9 khối con.
* Quá trình xử lý (băm) thông điệp gồm 5 bước chính, cụ thể như sau:

**Bước 1 - Đệm bit (Append padding bits):**

* Thông điệp được đệm thêm các bit sao cho sau khi đệm thì độ dài L của nó thoả mãn .
* Việc đệm thêm bit này phải luôn được thực hiện, kể cả khi độ dài thông điệp đã thoả mãn yêu cầu đồng dư ở trên. Như vậy, số lượng bit đệm thêm sẽ là từ 1 cho đến 1024 bit.
* Phần đệm gồm một bit 1 và theo sau là số các bit 0 cần thiết. Ví dụ đệm thêm 5 bit, thì phần đệm sẽ là 10000.

**Bước 2 - Đệm độ dài (Append length):**

* Một khối (chuỗi, xâu, string) gồm 128 bit sẽ được đệm thêm vào thông điệp nhận được sau Bước 1. Khối đệm thêm này thể hiện một số nguyên không dấu (unsigned integer) 128-bit, tức là độ dài ban đầu của thông điệp. (Ví dụ, Thông điệp M ban đầu có độ dài là L\_0 bit, do , nên L\_0 là số nguyên có tối đa 128 bit. Ta biểu diễn số nguyên L\_0 thành một xâu bit có độ dài không vượt quá 128. Khi đó xâu biểu diễn số nguyên L\_0 này chính là phần được đệm thêm vào thông điệp nhận được sau Bước 1.)

Như vậy, sau Bước 1 và Bước 2 ta nhận được thông điệp M có độ dài bit là bội của 1024. Và chia M thành các khối con, ký hiệu , mỗi khối con có độ dài bit là 1024; tổng độ dài của thông điệp M sẽ là N x 1024.

**Bước 3 - Khởi tạo bộ đệm giá trị băm (Initialize hash buffer):**

* Một bộ đệm 512-bit sẽ được sử dụng để lưu giữ các giá trị trung gian và cuối cùng của quá trình tính toán.
* Bộ đệm này được biểu diễn bằng 8 thanh ghi 64-bit, ký hiệu (a, b, c, d, e, f, g, h).
* 8 thanh ghi được khởi tạo bằng các số nguyên a, b, c, d, e, f, g, h, như trong tài liệu. (512 bit khởi tạo này chính là giá trị IV = H\_0.)

**Bước 4 - Xử lý thông điệp theo từng khối 1024-bit (128-byte):**

* Trọng tâm của hàm băm là hàm nén F gồm 80 vòng.
* Mỗi vòng nhận đầu vào là giá trị bộ đệm 512-bit, abcdefgh, và cập nhật giá trị của bộ đệm này.
* Tại thời điểm bắt đầu của vòng đầu tiên (Round 0), bộ đệm có giá trị là băm trung gian .
* Ở vòng thứ t (Round t) sẽ sử dụng một giá trị 64-bit, ký hiệu W\_t, nhận được từ khối thông điệp M\_i đang được xử lý. (*Ta sẽ trình bày việc nhận được W\_t từ M\_i trong phần sau.*)
* Ngoài ra, mỗi vòng sử dụng thêm một hằng số, ký hiệu K\_t. (Ta sẽ trình bày về K\_t ở phần sau: K\_t được tính thế nào, tại sao lại cần có K\_t???)
* Đầu ra của vòng thứ 80 (Round 79) được cộng với giá trị băm trung gian để cho ra giá trị băm trung gian . Trong đó, việc cộng được thực hiện một cách độc lập đối theo mỗi từ trong bộ đệm với các từ tương ứng trong giá trị trung gian sử dụng phép cộng theo modulo .

**BTVN:** Xem lại nội dung đã học, đọc kỹ các bước tính toán, đặc biệt là Bước 4. Xem trước nội dung cho tuần sau.

**Ngày 12/10/2021**

Tính toán thực hành SageMath: Lấy file trong thư mục của lớp SageMathHashFunctions….

Giới thiệu 2 nguồn (thư viện hàm băm & thư viện các hệ mã):

1. Thư viện Python "hashlib": <https://docs.python.org/3/library/hashlib.html>

Thư viện này bao gồm nhiều thuật toán băm và thông báo an toàn khác nhau. Bao gồm các thuật toán băm an toàn FIPS SHA1, SHA224, SHA256, SHA384 và SHA512 (được định nghĩa trong FIPS 180-2) cũng như thuật toán MD5 của RSA (được xác định trong Internet RFC 1321).

Các thuật ngữ “băm an toàn” (secure hash) và “phân tích thông điệp” (message digest) có thể hoán đổi cho nhau. Các thuật toán cũ hơn được gọi là phân tích thông điệp. Thuật ngữ hiện đại là băm an toàn.

Ví dụ: SHA255 băm thông điệp "Hello Vietnam" thành

b3cd875241a6fdbef567f2ef429f15f56a60679cacf2c47f6c7463f736c82fe0

1. Thư viện về các hệ mật mã có sẵn trong SageMath.

Nguồn:

https://doc.sagemath.org/html/en/reference/cryptography/index.html

Hãy tính toán thực hành một vài hệ mã cổ điển:

https://doc.sagemath.org/html/en/reference/cryptography/sage/crypto/classical.html

Mỗi hệ mã tính ít nhất 1 ví dụ khác trong HELP.

1. shift cipher (Mã dịch chuyển)

2. substitution cipher (Mã thay thế); see SubstitutionCryptosystem

3. transposition cipher (Mã chuyển vị); see TranspositionCryptosystem

4. affine cipher (Mã affine); see AffineCryptosystem

5. Hill or matrix cipher (Mã Hill); see HillCryptosystem

6. Vigenere cipher (Mã Vigenere); see VigenereCryptosystem

Kiểm tra 30 phút (7:25-7:55)

Gia hạn thời gian: đến 9:00.

**Ngày 13/10/2021 (Tuần 5, Thứ 4), GV: Võ Tùng Linh**

**Hàm băm SHA-512 (tiếp theo):**

Sau khi đã xử lý thông điệp cần băm (sau Bước 1 và Bước 2), các tính toán chính trong mô tả hàm băm SHA-512 có thể được viết dưới dạng công thức:

* IV = H\_0
* H\_i = SUM\_64(H\_{i-1}, abcdefgh\_i), 1 <= i <= N
* H(M) = H\_N

Trong đó:

* IV: là giá trị khởi tạo ban đầu cho bộ đệm abcdefgh (Bước 3).
* abcdeefgh\_i: đầu ra của vòng cuối cùng của hàm nén (hàm vòng) F khi xử lý một khối thông điệp 1024-bit.
* N: số khối thông điệp con được chia từ khối thông điệp ban đầu sau khi đã đệm.
* SUM\_64: Phép cộng theo modulo 2^64 thực hiện một cách độc lập trên từng cặp word tương ứng của các đầu vào.
* H(M): Giá trị băm mà ta cần tính.

**Chi tiết về Hàm nén (Hàm vòng) F:**

* Vòng thứ t có 3 đầu vào: bộ đệm 512-bit abcdefgh\_t, một word W\_t, và hằng số K\_t; Đầu ra sẽ là trạng thái mới cho bộ đệm 512-bit, abcdefgh\_{t+1}; 0 <= t <= 79.
* Công thức tính toán: xem trong tài liệu. (Ký tự a, b, c, d, e, f, g, h bên trái là thể hiện giá trị trạng thái mới, còn bên phải là giá trị ta đang xử lý.)
* ch(e, f, g) = (e AND f) ++ (NOT e AND g); hàm này được gọi là hàm điều kiện
* + : phép cộng modulo 2^64 bit.
* ++ : phép cộng XOR.
* Maj(a, b, c) = (a AND b) ++ (a AND c) ++ (b AND c).
* ROTR^n(x): Phép dịch vòng phải n-bit với đối số x. (x = 1001111, n = 2 ⇒ ROTR^2(x) = 1110011.)
* W\_t : một word 64-bit nhận được khối thông điệp M\_i với giá trị i: 1 <= i <= N.
* K\_t: hằng số 64-bit.

**Lược đồ thông điệp:** Từ 1 khối thông điệp M\_i 1024-bit, cho ta 80 word, mỗi word 64 bit:

* 16 word đầu tiên (W\_0, …, W\_15) nhận được trực từ khối thông điệp M\_i 1024-bit, 16 x 64 = 1024.
* 64 word tiếp theo nhận được qua công thức (xem tài liệu):
* SHR^n(x): Phép dịch phải đối số x đi n bit, đệm các bit 0 vào bên trái. (x = 1001111; n = 2 ⇒ SHR^2(x) = 0010011.)
* + : Phép cộng modulo 2^64.

**Các hằng số K\_t:** Mỗi K\_t sẽ là 64 bit đầu tiên trong biểu diễn phần thập phân của căn bậc 3 của 80 số nguyên tố đầu tiên. Việc đưa thêm hằng số K\_t chỉ là để tăng thêm “tính ngẫu nhiên” cho quá trình tính toán và đầu ra.

**BTVN:** Các bạn xem mô tả hàm băm SHA-512 dưới dạng thuật toán trong tài liệu, Hình 11.13 trang 363.

**Đánh giá độ an toàn của hàm băm SHA-512:** Cho đến nay hàm băm SHA-512 vẫn được xem là một trong những hàm băm an toàn nhất và được sử dụng rộng rãi trong thực tế. Hàm băm SHA-512 đảm bảo được các tính chất: kháng tiền ảnh, kháng va chạm, kháng tiền ảnh thứ 2, hiệu quả trong tính toán, đầu ra của nó đảm bảo được tính ngẫu nhiên.

**Ngày 19/10/2021 (Tuần 6, Thứ 3), GV: Võ Tùng Linh**

**Chữ ký số (Chữ ký điện tử, Digital Signatures)**

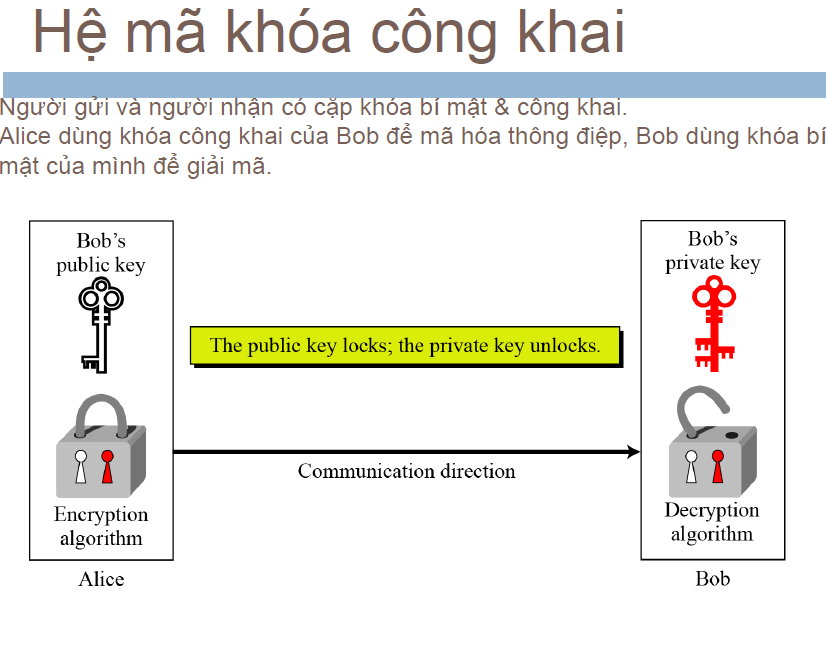
Tài liệu: Chương 13 trong sách của Stallings.

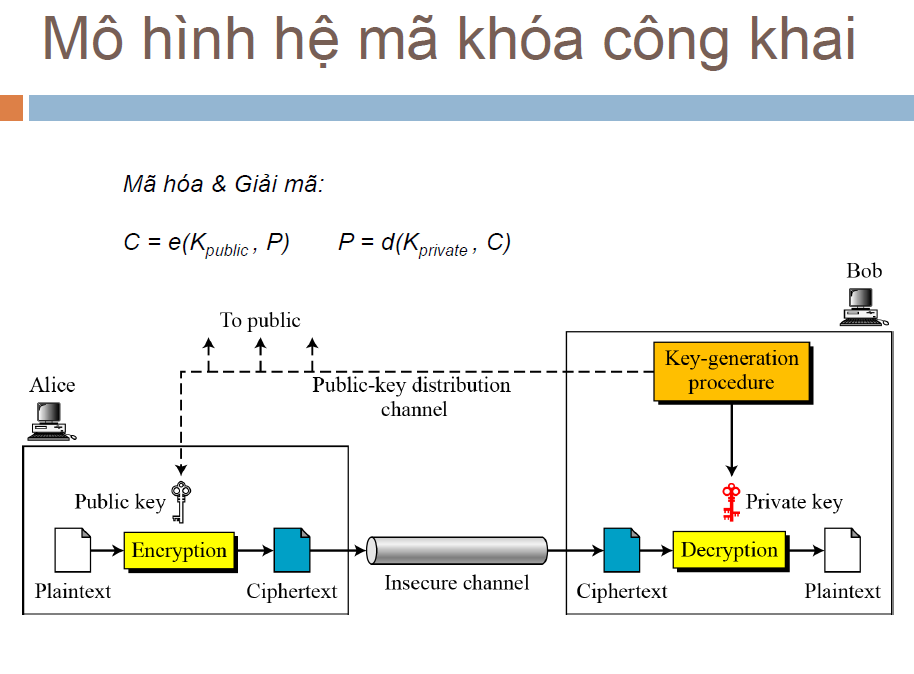
Lược đồ (Sơ đồ - scheme) chữ ký số là một giao thức, thuật toán hoặc là một quy trình cho đầu ra có hiệu quả tương tự như là chữ ký viết tay. “Đầu ra” = chữ ký số:

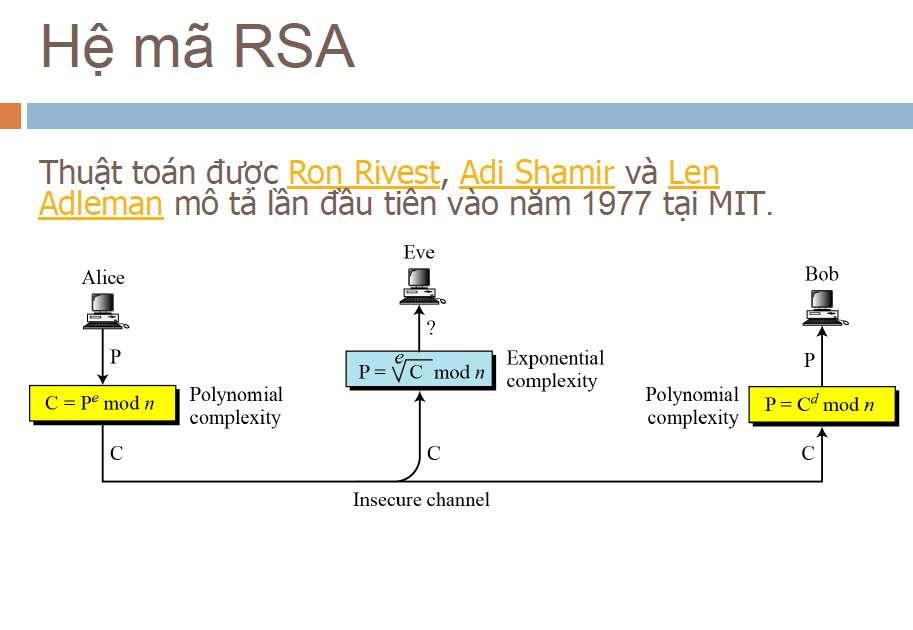
* Chữ ký số là dấu hiệu mà chỉ có người gửi mới có thể tạo ra được, nhưng những người khác đều có thể nhận biết được hoặc xác minh được hoặc kiểm tra được chữ ký đấy chính là của người gửi.
* Cũng như chữ ký tay, chữ ký số cũng được sử dụng để xác nhận thông báo.

Chữ ký số cần phải thoả mãn những yêu cầu gì:

* Không thể giả mạo được: Nếu người ký P ký một thông điệp M bằng chữ ký số S(M, P) thì cặp thông điệp, chữ ký số (M, S(M, P)) không thể bị giả mạo bởi bất kỳ một ai khác ngoài P.
* Xác thực:
* Không thể thay đổi:
* Không thể sử dụng lại:







Hàm Euler:

phi(N) số các số tự nhiên nguyên tố cùng nhau với số tự nhiên N và <N.

Ví dụ: Phi(6)=2, vì có đúng 1 và 5 (<6) nguyên tố cùng nhau với 6.

Mệnh đề: Nếu p và q là 2 số nguyên tố khác nhau/ Đặt N=p.q thì

phi(N) = (p-1)(q-1)

Chứng minh: Liệt kê các số < N mà không nguyên tố cùng nhau với N=p.q?

Nhận xét (a,N)=1 tương đương với (a,p.q)=1, tương đương với (a,p)=1 và (a,q)=1.

Do đó a<N, (a,N)=1 thì a thuộc P hợp Q, P={p,2p,3p,...,(q-1)p} và Q={q,2q,3q,...,(p-1)q},

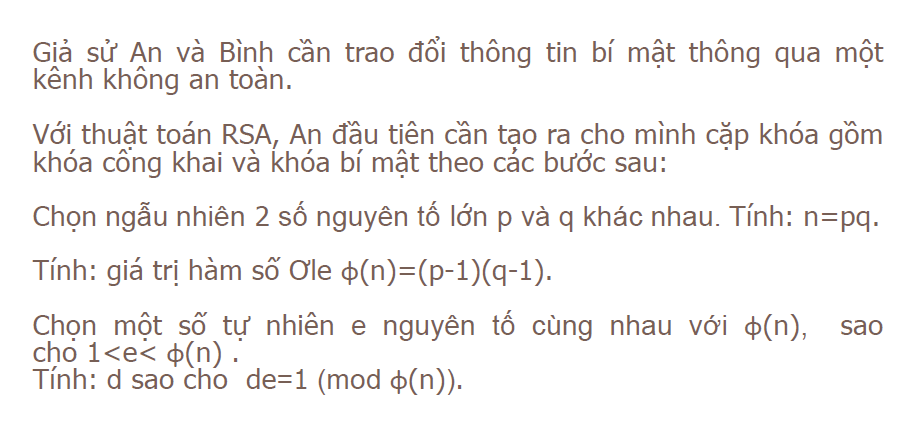
Tập các số tự nhiên nguyên tố cùng nhau với số tự nhiên N và <N bằng

X = {1,2,...,N-1} \ P U Q

|X| = N-1-((q-1)+(p-1))= pq -p -q +1 =(p-1).(q-1). ĐPCM.

Thuật toán RSA

* Bước tạo khóa: Cặp khóa



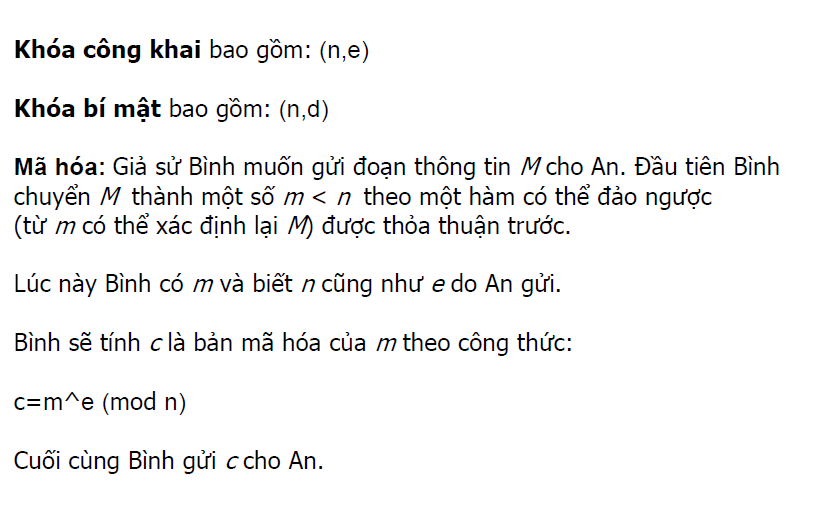
Ví dụ: An chọn 2 số nguyên tố 11 & 19.

n= 11.19 = 209

phi(n) = (11-1)(19-1) = 180

Chọn e (1<e<phi(n)) nguyên tố cùng nhau với phi(n), chẳng hạn e=23. Nghich đảo nhân d=47

(Trong SageMath: inverse\_mod(23,180))



An có cặp khóa (pk,sk):

* Khóa công khai pk=(n,e)=(209,23)
* Khóa bí mật sk=(n,d) = (209,47)

Bình muốn gửi thông điệp M (tương ứng với một số tự nhiên m < n=p.q=209).

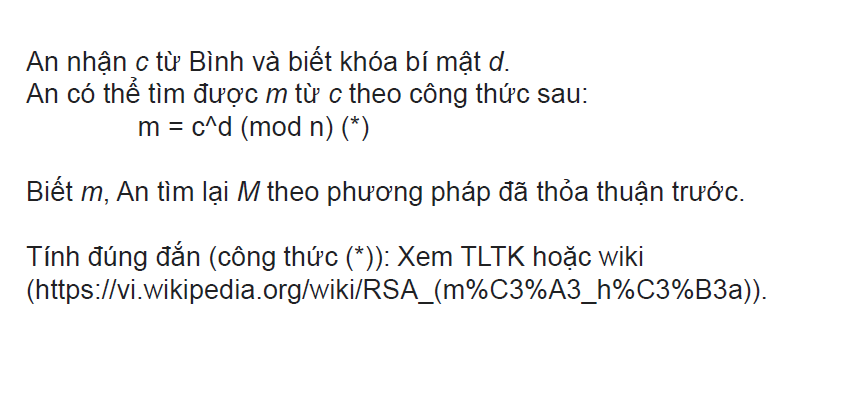
An công khai pk=(n,e)

Chẳng hạn m=100

Bình mã hóa: c=m^e (mod n) = 100^23 (mod 209) = 199

Bình gửi số cho An số c=199.

Tiếp theo bước giải mã:



Ví dụ: An tính c^d (mod n) = 199^47 (mod 209) = 100 = m

Chứng minh tính đúng đắn của thuật toán:

Tại sao có c^d (mod n) = m ????

Khi lấy mod n ta có:

c^d = (m^e)^d = m^(e.d) =m ????

Ta sẽ chứng minh m^(ed)-m chia hết cho p và cũng chia hết cho q

x=d.e = 1 (mod phi(n)), x-1 chia hết cho phi(n)

Chứng minh: m^x - m chia hết cho p

m^x -m = m(m^{x-1}-1) chia hết cho m^phi(n)-1 = m^(p-1)\*(q-1)-1

m^(p-1) =1 (mod p)

m^(q-1) =1 (mod q)

Suy ra m^((p-1)(q-1)) =1 (mod pq)

(m^{ab}-1 = (m^a)^b-1 = (m^a-1)\*....)

Kết quả số học (Định lý Fermat): a^p-a = 0 (mod p)

a^p-a = a(a^{p-1}-1) chia hết cho p

Nếu a không chia hết cho p thì a^{p-1} chia p dư 1

\*\*\*\*\*\*\*\*\*\*\*

Gán thông điệp M với một số m như thế nào?

**Ngày 26/10/2021 (Tuần 7, Thứ 3), GV: Võ Tùng Linh**

**Chữ ký số (tiếp)**

**Định nghĩa lược đồ chữ ký số:** Một *lược đồ chữ ký số (signature scheme)* gồm 3 thuật toán thời gian đa thức, xác suất (Gen, Sign, Vrfy) và một không gian thông điệp M, trong đó:

* Gen được gọi là **thuật toán sinh khóa (key generation algorithm)**. Đây là một thuật toán *được ngẫu nhiên hoá*, nhận đầu vào là một tham số an toàn k, và cho đầu ra là một cặp giá trị - gọi là cặp khoá (sk, pk), trong đó sk được gọi là *khoá bí mật (khoá ký, khoá riêng - private key)*, pk được gọi là *khoá công khai (khoá xác minh, khoá kiểm tra - public key)*.
* Sign được gọi **thuật toán ký (signing algorithm)**. Thuật toán này *có thể được ngẫu nhiên hoá*. Thuật toán nhận đầu vào là khoá bí mật sk và thông điệp cần ký m thuộc không gian thông điệp M, cho đầu ra là chữ ký số . (Lưu ý: Người ký sử dụng khoá bí mật của mình.)
* Vrfy được gọi là **thuật toán xác minh (kiểm tra) chữ ký (verification algorithm)**. Thuật toán này là một thuật toán *tất định*. Thuật toán nhận đầu vào là khoá công khai pk, thông điệp m, và chữ ký số . Thuật toán trả về kết luận chữ ký số là ***hợp lệ*** hoặc ***không hợp lệ***. (Lưu ý: Người xác minh sử dụng khoá công khai của người ký.) .
* Ta nói rằng cặp thông điệp, chữ ký (m, ) là hợp lệ (valid) nếu thuật toán kiểm tra cho kết luận là *hợp lệ*; nếu không thì cặp thông điệp, chữ ký là *không hợp lệ*.
* Một lược đồ chữ ký số **phải** có tính nhất quán, tức là nếu chữ ký được tạo ra một cách đúng đắn thì chữ ký số phải thoả mãn thuật toán xác minh.
* .

**Một số kiểu tấn công vào các lược đồ chữ ký số:**

Ký hiệu A là người ký, C là kẻ tấn công.

* Tấn công chỉ sử dụng khoá công khai (key-only attack): C chỉ biết khoá công khai của A.
* Tấn công có sử dụng thông điệp đã biết (Known message attack): C được phép truy cập vào một tập các thông điệp và chữ ký tương ứng.
* Tấn công sử dụng các thông điệp được lựa chọn trước (generic chosen message attack): Trước khi cố gắng tấn công chữ ký của A, thì C chọn một danh sách các thông điệp, độc lập với khoá công khai của A (C chọn các thông điệp trước khi biết khoá công khai của A). Sau đó thì C được cung cấp các chữ ký tương ứng với các thông điệp đã chọn.
* Tấn công sử dụng các thông điệp được lựa chọn sau (directed chosen message attack): Tương tự như tấn công trên, C được chọn một danh sách các thông điệp sau khi đã biết khoá công khai của A nhưng mà trước khi biết các chữ ký tương ứng. Sau đấy thì C cũng được cung cấp các chữ ký tương ứng.
* Tấn công sử dụng các thông điệp được lựa chọn một cách thích ứng (adaptive chosen message attack): C được phép sử dụng A như một bộ “tiên tri” (oracle). Tức là, C có thể yêu cầu A trả về chữ ký của những thông điệp mà phụ thuộc vào các cặp thông điệp-chữ ký đã có trước đấy.

**Các mức thành công khi tấn công vào một lược đồ chữ ký số:**

* Phá vỡ hoàn toàn (total break): C xác định được khoá bí mật của A.
* Giả mạo phổ quát (universal forgery): C tìm được một thuật toán hiệu quả mà có thể tạo ra được chữ ký số giả mạo chữ ký của A trên các thông điệp tuỳ ý.
* Giả mạo lựa chọn (selective forgery): C giả mạo được chữ ký của A trên một thông điệp cụ thể do chính C chọn.
* Giả mạo tồn tại (Existential forgery): C có thể giả mạo được chữ ký của A trên ít nhất một thông điệp. C không có quyền can thiệp vào thông điệp.

27/10/2021

Chữ ký số RSA (RSA-DSA)

Thực hành SageMath: RSA-DSA.ipynb trong thư mục môn học:

<https://drive.google.com/file/d/1hL_dVCPP4oP7orwb_xEjGMyDyXLJ9jE7/view?usp=sharing>

Số RSA: (n=pxq, với p và q nguyên tố, khác nhau).

<https://en.wikipedia.org/wiki/RSA_numbers>

RSA-100 = 1522605027922533360535618378132637429718068114961380688657908494580122963258952897654000350692006139

**Ví dụ: RSA-DSA**

|  | **Alice** | **Bob** |
| --- | --- | --- |
| **Bí mật** | pA=29, qA=71, dA=647, phi(nA)=1960 | pB=29, qB=37, phi(nB)=1008, dB=835 |
| **Công khai** | nA=2059, eA=103 | nB=29x37=1073, eB=571 |
|  | m=331  331^571 (mod 1073) =829 | Bob nhận được 829, tính:  829^571 (mod 1073) =331 |
| **Chữ ký** | m^dA = 331^647 (mod 2059) = 1235 | Bob nhận được (829,1235)  Xác thực chữ ký:  m1=1235^103 (mod 2059)=331  Tính như ví dụ 1:  m2= 829^571 (mod 1073) =331  Chữ ký là đúng nếu m1=m2 |

**Ngày 02/11/2021 (Tuần 8, Thứ 3), GV: Võ Tùng Linh**

Lược đồ CKS gồm bộ 3 thuật toán: sinh khoá, sinh chữ ký, xác minh chữ ký.

**Lược đồ chữ ký số ElGamal:**

* ElGamal đề xuất, 1984 hoặc 1985.
* Sử dụng cơ chế mật mã khoá công khai: khoá bí được sử dụng để ký và khoá công khai được sử dụng để xác minh chữ ký.
* Nhắc lại một kết quả từ lý thuyết số: giả sử q là một số nguyên tố, nếu là căn nguyên thuỷ bậc q, tức là là những phần tử khác nhau. Nói cách khác, là phần tử có cấp q-1, (mod q). . (Ví dụ, ta lấy trường hữu hạn F\_q, nhóm các phần tử khác 0, , .)
* Với mọi số nguyên m, nếu và chỉ nếu .
* Với mọi số nguyên i, j, nếu và chỉ nếu .

**Các tham số miền** (domain parameters): là các tham số công khai được thống nhất trước giữa các bên tham gia.

Các tham số miền của lược đồ chữ ký số ElGamal gồm có , trong đó là một số nguyên tố, là một phần tử có cấp . (F\_q = { x mod q}) , phép toán + và phép toán nhân theo modulo

Trong các lược đồ chữ ký số, thường sẽ cần sử dụng một hàm băm H. Hàm băm này cũng sẽ được thống nhất trước và công khai.

**Mô tả lược đồ chữ ký số ElGamal:**

**Thuật toán sinh khoá:** Người dùng A sinh cặp khoá bí mật/công khai như sau:

1. Sinh một số nguyên ngẫu nhiên thoả mãn .
2. Tính .
3. Khoá bí mật của A là giá trị , và khoá công khai tương ứng của A là giá trị .

**Thuật toán sinh chữ ký:** Để ký một thông điệp M, trước tiên A sử dụng hàm băm H đã thống nhất trước để tính giá trị băm . Ở đây yêu cầu hàm băm H phải thoả mãn điều kiện sao cho . Khi đấy, A thực hiện các tính toán sau: *(1. Băm thông điệp M: m=H(M) sao cho 1 <= m <= q-1)*

1. A chọn một số nguyên ngẫu nhiên thoả mãn và . (Giá trị K này còn được gọi là *giá trị bí mật tức thời*)
2. Tính giá trị .
3. Tính .
4. Tính .
5. Chữ ký số là cặp giá trị .

Thuật toán ký này là thuật toán *được ngẫu nhiên hoá*.

**Thuật toán xác minh chữ ký:** Sau khi B nhận được chữ ký số , trước tiên B cũng sẽ sử dụng hàm băm H để tính , sau đó thực hiện các tính toán sau:

1. Tính .
2. Tính .
3. Kiểm tra đẳng thức ? Nếu đúng thì chấp nhận chữ ký; ngược lại thì bác bỏ chữ ký.

**Tính đúng đắn của chữ ký:**

Ta có:

.

Như vậy biểu thức trên trở thành:

.

**BTVN: 1)** Đọc kỹ lại trong sách phần mô tả 3 thuật toán của Lược đồ chữ ký số ElGamal, đặc biệt là phải hiểu được những tính toán trong thuật toán cũng như tính đúng đắn của chữ ký.

**2)** Thực hiện ví dụ ở trang 425 trong sách bằng Sage.

**Ngày 09/11/2021 (Tuần 9, Thứ 3), GV: Võ Tùng Linh**

**Lược đồ chữ ký số DSA (DSA = Digital Signature Algorithm)**

**Các tham số miền:** và hàm băm , trong đó:

* là một số nguyên tố thoả mãn với
* là một ước nguyên tố của , độ dài bit của là , tức là , ở đây .
* mod p, trong đó là một số nguyên thoả mãn và . ( là phần tử có cấp bằng trong nhóm .) Với g được lấy như trên, hiển nhiên .
* là một hàm băm có độ dài đầu ra 256 hoặc 512 bit.

**Thuật toán sinh khoá:** Một người dùng sinh cặp khoá bí mật/công khai của mình như sau:

1. Sinh một số nguyên ngẫu nhiên hoặc giả ngẫu nhiên .
2. Tính .
3. Cặp khoá bí mật/công khai sẽ là . Cặp khoá này còn được gọi là cặp khoá bí mật/công khai dài hạn.

**Thuật toán sinh chữ ký (thuật toán ký):** Để ký một thông điệp , thì người ký thực hiện các bước sau:

1. Sinh một số nguyên ngẫu nhiên hoặc giả ngẫu nhiên . Giá trị được gọi là khoá bí mật tức thời của người ký.
2. Tính . ( có cấp nên )
3. Tính . ()
4. Chữ ký số là .

**Thuật toán xác minh chữ ký:** Nhận được thông điệp cùng với chữ ký , người xác minh (người kiểm tra) chữ ký thực hiện các bước sau:

1. Tính .
2. Tính .
3. Tính .
4. Tính .
5. Kiểm tra đẳng thức ? Nếu đúng thì chữ ký là hợp lệ; ngược lại thì chữ ký là không hợp lệ.

**Chứng minh tính đúng đắn của lược đồ:**

.

**BTVN:**

1. Viết chương trình thực thi lược đồ chữ ký số DSA bằng Sage.
2. Xem lại nội dung bài học trong sách.

10/11/2021

**DSA với SageMath:**

**Ví dụ 1:** (trang741-742 [Stallings]) Thực hiện DSA step-by-step để hiểu thuật toán trên lớp.

<https://drive.google.com/file/d/1jJFKDjotxf7dLZnJY7wg99lJaN-QaxXp/view?usp=sharing>

**Ví dụ 2:** (trang742-743 [Stallings]) Viết lại ví dụ 1 dưới dạng các hàm.

(Cùng nhau thực thi ví dụ 2!!!)

**Kiểm tra 30 phút:** Hoàn thiện file DSA-2 (<https://drive.google.com/file/d/1jJiIkRGhWJIJMZIZm-FIzEW5P0PsDu7O/view?usp=sharing>)

**Đáp án:**

def DSA\_generate\_domain\_parameters():

g=1;

while (1==g):

q=1

while (q<2^15):

q=random\_prime(2^16)

p=1;

while (not is\_prime(p)):

p=(2^48+randint(1,2^46)\*2)\*q+1

h=randint(2,p-2)

g=power\_mod(h,int((p-1)/q),p)

return(p,q,g)

para=DSA\_generate\_domain\_parameters();para

p=para[0]

q=para[1]

g=para[2]

def DSA\_generate\_keypair(p,q,g):

x=randint(2,q-1)

y=power\_mod(g,x,p)

return(x,y)

def DSA\_sign(p,q,g,x,H):

k=randint(2,q-1)

r=power\_mod(g,k,p)%q

kinv=inverse\_mod(k,q)

s=(kinv\*(H+x\*r))%q

return (r,s)

def DSA\_signcheck(p,q,g,y,r,s,H):

w=inverse\_mod(s,q)

u1=(H\*w)%q

u2=(r\*w)%q

v=((power\_mod(g,u1,p)\*power\_mod(y,u2,p))%p)%q

return(v==r)

Tính giá trị băm:

import hashlib

hhex=hashlib.sha256(b"Nguyen Van Binh 1234567810112021").hexdigest();hhex

H=Integer(hhex, base=16);H

Hàm chuyển từ cơ số 16 sang thập phân:

Integer(hhex, base=16)

**Ví dụ:** Thực hiện

hhex=hashlib.sha256(b"Nguyen Van Binh 1234567810112021").hexdigest();hhex

H=Integer(hhex, base=16);H

Cho kết quả:

15621663767330805691303543387125036124455649128583869415299347619097230409890

Xem đáp án: file DSA-3

Bài tập về nhà: Mã hóa + chữ ký số DSA như thế nào?

**Ngày 16/11/2021 (Tuần 10, Thứ 3), GV: Võ Tùng Linh**

**Lược đồ chữ ký số RSA-PSS**

Lược đồ chữ ký số RSA:

**Thuật toán sinh khoá:** Người ký sinh cặp khoá công khai, bí mật như sau:

* Sinh hai số nguyên tố lớn có cùng kích thước. (p, q phải được giữ bí mật)
* Tính và tính .
* Chọn ngẫu nhiên một số nguyên và .
* Tính giá trị và .
* Khoá công khai:
* Khoá bí mật: .

Những người tham gia cùng thống nhất sử dụng một hàm băm an toàn.

**Thuật toán ký:** Để ký một thông điệp , người ký sử dụng khoá bí mật của mình để thực hiện các bước tính:

* Tính giá trị băm và kiểm tra .
* Tính .
* Chữ ký số là .

**Thuật toán xác minh chữ ký:** Sau khi nhận được thông điệp và chữ ký , người xác minh sử dụng khoá công khai của người ký để thực hiện việc xác minh chữ ký như sau:

* Tính giá trị băm và kiểm tra .
* Tính và so sánh giá trị nhận được với . Nếu bằng nhau thì khẳng định chữ ký là hợp lệ; ngược lại thì khẳng định chữ ký là không hợp lệ.

**Tính đúng đắn của chữ ký:**

Ta có:

.

⇒ với là một số nguyên nào đấy.

.

(Các tính chất của hàm Euler).

**Lược đồ chữ ký số RSA-PSS**

***Hàm sinh mặt che (MGF - Mask Generation Function):***

* Tuỳ chọn: Hàm băm với đầu ra có độ dài là octet. có thể là hàm băm SHA-1, SHA-2, SHA-3,...

**Mô tả hàm MGF:**

* Input:
  + : là một xâu cần được che.
  + : độ dài theo octet của mặt che.
* Output:
  + : một xâu có độ dài là octet.

*Các bước thực hiện của hàm MGF:*

1. Khởi tạo các biến:

* (về tìm hiểu khái niệm hàm trần - ceil function??? Hàm làm tròn lên / Round up)

1. Tính toán các giá trị trung gian:

* **For** **to** :
  + Biểu diễn giá trị như là một xâu 32-bit, ký hiệu là .

1. Đưa ra kết quả:

* octet cao nhất của .

Cụ thể, ta có viết

.

Khi đó, đầu ra chính là octet đầu tiên của .

**BTVN:** Các bạn về xem trước phần mô tả hoạt động ký và xác minh chữ ký.

**17/11/2021**

Kiểm tra giữa kỳ

Bài tập nhóm (nộp 30/11, 11:59PM, 90%) + trình bày vào 1/12 (10%)

**Nội dung:**

**Nội dung 1:** (50%) Viết một bài luận “Tìm hiểu chữ ký điện tử ở Việt Nam”:

1. Quy chuẩn: Chẳng hạn QCVN 5 : 2016/BQP
2. Chữ ký điện tử thông dụng nhất được sử dụng ở Việt Nam?
3. Chữ ký điện tử ở ĐHQG HN đang sử dụng: VNU-office (<https://drive.google.com/drive/folders/1ZDQZMhXlf9mkO9YfAfukgrwH2Lh5kJwc>)
4. Ví dụ một văn bản được ký số của HUS: https://drive.google.com/file/d/1wnDS0\_UdiFAMtbyn465lxs0TmZvb3hq6/view?usp=sharing
5. Ở VN hiện nay có những công ty/doanh nghiệp nào được phép làm dịch vụ chữ ký điện tử?

(Viettel, VNPT, … / Công ty Misa https://esign.misa.vn/)

1. Start-up về chữ ký điện tử?

**Nội dung 2:** (30%)Viết chương trình SageMath thực thi thuật toán RSA-PSS.

**Nội dung 3:** (10%) Bài luận tìm hiểu về mã QR. Đưa ứng dụng mật mã vào mã QR. Kết hợp RSA vào QR code? Chẳng hạn RSA token là gì?

<https://laodong.vn/the-gioi/ma-qr-thong-tin-ca-nhan-co-duoc-an-toan-964210.ldo>

Làm cách nào để tăng cường bảo mật mã QR?

Video clip về QR code: <https://www.youtube.com/watch?v=gzlSjrL2ryg&t=33s>

(phụ đề dịch bởi Khoa T-C-T).

**Final project:** (Điểm thành phần cuối kỳ)

Đưa ứng dụng mật mã vào mã QR.

Start-up QR code bảo mật?

Crypto (mật mã) khác với Mã sửa sai (Error correcting codes)?

**Song ánh là gì?**

ĐN1: Ánh xạf: X → Y được gọi là một song ánh nếu f vừa là một đơn ánh và vừa là một toàn ánh.

ĐN2: Ánh xạf: X → Y được gọi là một song ánh nếu mọi y thuộc Y tồn tại duy nhất x thuộc X sao cho f(x)=y.

Hiểu nôm na, song ánh cho một tương ứng 1-1 giữa X và Y.

**Ví dụ:**

1. Ánh xạ **Z** ----> 2**Z**, biến n thành 2n, là một song ánh. **Z**: tập hợp các số nguyên, 2**Z** tập hợp các số nguyên chẵn.
2. Tồn tại nhiều song ánh (0,1) vào **R** (tập hợp các số thực).

**Crypto (mật mã):** Có “song ánh”, e (mã hóa, encrypt) & d (giải mã, decrypt)

e(P) = C, Alice gửi C cho Bob, Bob nhận được C, Bob tính d(C)=P (khi e,d là các ánh xạ ngược của nhau). P: postscript text (bản rõ), C: cipher text (bản mã)

**Mã sửa sai (Error correcting codes):**

**Câu hỏi:** QR code là một ví dụ điển hình về mã sửa sai. Có cần song ánh hay không?

QR code: f biến x thành mã QR của x (x: đường link trang web, văn bản, …)

<https://www.the-qrcode-generator.com/>

f có là đơn ánh? Có.

f có là toàn ánh??? (BTVN)

v: bản gốc

Người gửi (Alice) sẽ mã hóa v, bằng ánh xạ mã hóa (encode) c(v) = w.

Người gửi sẽ gửi w cho người nhận (Bob).

Trong quá trình gửi, do đường truyền, Bob không nhận được chính xác w, thay vào đó, nhận được w’.

Bob “sửa sai” w’ được w” (nói chung, thì w”=w), sử dụng hàm giải mã (decode) d, d(w)=v.

Ví dụ 1: (Mã sửa sai) Cổ điển nhất “Repeating code”.

Alice muốn gửi thông điệp LOVE cho Bob, dùng mã sửa sai “Repeating code” 3 lần:

v=”LOVE”

w=c(v)=”LLLOOOVVVEEE”

Giả sử đường truyền khá tốt, nhiều nhất chỉ có 1 lỗi, chẳng hạn, Bob nhận được

w’=”LLLOOEVVVEEE”

Bob sửa sai được:

w’=”LLLOOEVVVEEE” thành w”=”LLLOOOVVVEEE”.

Nhận xét: Không hiệu quả, có 2 lí do:

1. Tốn quá nhiều không gian lưu trữ;
2. Chỉ sửa sai được khi có nhiều nhất 1 lỗi.

Ví dụ 2: Mã Hamming (7,4), mã sửa sai không tầm thường xuất hiện đầu tiên

Mã 1 xâu 4 kí tự thành 7 kí tự.

Các xâu bit (0 hoặc 1)

abcd mã hóa thành abcdxyz

x = (a+b+c) mod 2

y = (a+b+d) mod 2

z = (b+c+d) mod 2

Xem video clip về mã Hamming: <https://www.youtube.com/watch?v=eixCGqdlGxQ>

Buổi sau: Hamming code, linear code

Cyclic code, Reed-Solomon code (dùng trong CD, DVD, Facebook, Google, …, QR code)

QR code.

**Ngày 23/11/2021 (Tuần 11, Thứ 3), GV: Võ Tùng Linh**

**Thuật toán ký số RSA-PSS (tiếp)**

***Hoạt động ký (Signing Operator):***

Trước khi tạo chữ ký RSA-PSS trên thông điệp , bước đầu tiên người ký sẽ tạo ra một bản tóm lược thông báo có độ dài cố định từ thông điệp , gọi là thông điệp đã được ghi mã, ký hiệu (Encoded Message). ( M ⇒ EM)

*Thuật toán ghi mã thông điệp (message encoding):*

**Tuỳ chọn:**

* : là một hàm băm có độ dài đầu ra octet.
* : hàm sinh mặt che được mô tả như ở phần trước.
* : độ dài tính theo octet của giá trị salt. Thông thường, .

**Input:**

* : Thông điệp cần được ghi mã để ký.
* : Một giá trị nhỏ hơn hoặc bằng độ dài theo bit của giá trị modulo .

**Output:**

* : thông điệp đã được ghi mã.

**Các tham số:**

* : độ dài tính theo octet của : .
* : xâu dạng hex , tức là một xâu gồm 64 bit 0.
* : xâu hex gồm octet theo sau octet có giá trị . ().
* : giá trị hex BC.

**Các bước thực hiện:** Quá trình ghi mã gồm các bước sau:

1. Tính giá trị băm của .
2. Sinh một xâu octet giả ngẫu nhiên và tính khối dữ liệu.
3. Tính giá trị băm của .
4. Tính khối dữ liệu .
5. Tính giá trị mặt che của : .
6. Tính .
7. Thiết lập bit ngoài cùng bên trái của octet ngoài cùng bên trái trong bằng . (*ví dụ minh hoạ, maskedDB =* ***O****OOOOOOO,* ***O =*** *000****11100***)
8. .

**Một vài lưu ý:**

* Việc sử dụng các xâu đệm và sẽ làm cho bên tấn công khó tìm được một thông điệp khác mà cho cùng một bản tóm lược thông điệp;
* Việc sử dụng giá trị sẽ đảm bảo khi ký hai lần trên cùng một thông điệp với cùng 1 khoá bí mật thì chữ ký sẽ khác nhau.

*Tạo chữ ký (forming the signature):* Ta đi tạo chữ ký số với khoá bí mật và khoá công khai tương ứng sẽ là .

* Ta xử lý như là một số nguyên, không âm, ký hiệu .
* Chữ ký được tính bởi

.

* Chuyển giá trị thành xâu có độ dài là octet, phụ thuộc vào độ dài của giá trị modulo . (Ví dụ, ).

***Hoạt động xác minh chữ ký (Signature Verification):***

*Thuật toán giải mã (Decryption):* Sau khi nhận được chữ ký số , người xác minh thực hiện:

* Biến đổi xâu thành số nguyên không âm .
* Giá trị sẽ được khôi phục qua việc giải mã bằng cách tính

.

* Sau đấy biểu diễn thành một xâu ký hiệu có độ dài octet, trong đấy chính là độ dài bit của giá trị modulo .

*Thuật toán xác minh ( Verification):*

**Tuỳ chọn:**

* : một hàm băm có độ dài đầu ra là octet.
* : hàm sinh mặt che.
* : độ dài tính theo octet của giá trị .

**Input:**

* : thông điệp cần xác minh.
* : xâu biểu diễn của chữ ký đã được giải mã có độ dài .
* : một giá trị nhỏ hơn độ dài theo bit của modulo .

**Output:**

* Khẳng định “***consistent***” hay là “***inconsistent***” (Khẳng định chữ ký “Hợp lệ” hoặc “Không hợp lệ”).

**Các tham số:**

* : xâu dạng hex , tức là một xâu gồm 64 bit 0.
* : xâu hex gồm octet theo sau octet có giá trị . ().

**Các bước thực hiện:** Để xác minh , người xác minh thực hiện như sau:

1. Tính giá trị băm của .
2. Nếu , thì đưa ra “inconsistent” (“Không hợp lệ”) và dừng.
3. Nếu octet ngoài cùng bên phải của không có giá trị hexa là , thì đưa ra “inconsistent” (“Không hợp lệ”) và dừng.
4. Lấy là xâu gồm octet ngoài cùng bên trái của , và ký hiệu là xâu gồm octet tiếp theo.
5. Nếu bit bên trái của octet ngoài cùng bên trái của không bằng 0, thì đưa ra “inconsistent” (“Không hợp lệ”) và dừng.
6. Tính .
7. Tính .
8. Thiết lập bit ngoài cùng bên trái của octet ngoài cùng bên trái của bằng .
9. Nếu octet của không bằng , thì đưa ra “inconsistent” (“Không hợp lệ”) và dừng.
10. Thiết lập là xâu gồm octet cuối cùng của .
11. Tính khối dữ liệu .
12. Tính giá trị băm của .
13. Nếu , thì đưa ra “consistent” (“Hợp lệ”); ngược lại thì đưa ra “inconsistent” (“Không hợp lệ”).