

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN 1



BÀI THỰC HÀNH 16
THỰC TẬP CƠ SỞ

Họ và tên : Đinh Quang Hiếu

Mã sinh viên: B19DCAT065

Giảng viên giảng dạy: Hoàng Xuân Dậu

HÀ NỘI, THÁNG 5/2022

Bài 16: Tìm hiểu và cài đặt mã hóa bất đối xứng RSA

1.1. Mục đích

Sinh viên tìm hiểu một giải thuật mã hóa phổ biến và lập trình được chương trình mã hóa và giải mã sử dụng ngôn ngữ lập trình phổ biến như C/C++/Python/Java, đáp ứng chạy được với số lớn

1.2 Nội dung thực hành

1.2.1 Tìm hiểu lý thuyết

- Tìm hiểu về lập trình số lớn với các phép toán cơ bản
- Tìm hiểu về giải thuật mật mã khóa công khai RSA

Hệ mã RSA được giới thiệu vào năm 1977 bởi 3 nhà khoa học Ron Rivest, Adi Shamir, Len Adleman. Đây là một trong những hệ mã được sử dụng phổ biến nhất hiện nay, ứng dụng cho truyền dữ liệu an toàn qua internet, email. RSA còn là nền tảng mật mã cho các giao thức SSL/TLS, SET, SSH, PGP, ... RSA cũng được ứng dụng trong chữ ký số Digital Signature.

Breaking RSA là dạng bài thường xuyên gặp phải trong các cuộc thi CTF dưới nhiều hình thức. Bài viết này tôi sẽ mô tả ngắn gọn về hệ mã này, liệt kê các điểm yếu có thể khai thác của nó, cũng như các công cụ toán học, các tool cần thiết giúp giải quyết các bài RSA hay gặp khi chơi CTF.

RSA Cryptosystem

RSA thuộc nhóm hệ mã khóa công khai, dựa vào độ khó của bài toán phân tích 1 số ra thừa số nguyên tố (factoring problem). Để tạo cặp khóa Public key và Private key, Alice cần:

- Chọn 2 số nguyên tố lớn p, q với $p \neq q$
- Tính $n = pq$
- Tính giá trị hàm số Euler $\phi(n) = (p-1)(q-1)$
- Chọn 1 số e sao cho $1 < e < \phi(n)$ và $\gcd(e, \phi(n)) = 1$
- Tính $d = e^{-1} \pmod{\phi(n)}$, số d thỏa mãn $ed \equiv 1 \pmod{\phi(n)}$

Public Key gồm:

- n – module.
- e – số mũ mã hóa.

Private Key gồm:

- n – module.

- d – số mũ giải mã.

Khi Bob muốn gửi một tin nhắn M cho Alice, Bob chuyển M thành một số $m < n$ theo 1 cách thỏa thuận trước. Bob sẽ tính ra bản mã c từ bản rõ m theo công thức:

$$c = m^e \pmod{n}$$

Để giải mã, Alice dùng Private Key của mình để tính ngược lại:

$$m = c^d \pmod{n}$$

Quá trình giải mã có thể thu lại được m ban đầu là do:

$$c^d \equiv (m^e)^d \equiv m^{ed} \pmod{n} \equiv m \pmod{n} \text{ hay } m = c^d \pmod{n}$$

Dấu \equiv cuối cùng là tôi đã áp dụng định lý Euler. Chi tiết hơn về thiết kế hệ mã cũng như ví dụ có thể đọc ở đây [RSA – Wikipedia](#)

Độ mạnh của hệ mã RSA dựa trên việc bạn cần phân tích được n ra thừa số nguyên tố để tính d nếu muốn phá mã, và đến nay chưa có giải thuật nào hiệu quả trong thời gian đa thức giúp ta phân tích thừa số nguyên tố đối với các số lớn.

Hệ mã RSA nếu được thiết kế một cách đúng đắn với việc chọn các tham số n, p, q, e hợp lý thì sẽ rất an toàn, thế nhưng trong các bài CTF, các tham số này thường được chọn theo một cách nào đó khiến cho hệ mã yếu đi và dễ bị tấn công.

1.2.2 Chuẩn bị môi trường

- **Môi trường lập trình theo mong muốn.**

1.2.3 Các bước thực hiện và kết quả cần đạt

a) Các bước thực hiện

- Lập trình thư viện số lớn với các phép toán cơ bản để sử dụng trong giải thuật mã hóa/giải mã RSA
- Thử nghiệm chứng minh thư viện hoạt động tốt với các ví dụ phép toán cho số lớn

```
23
24     BigInteger bigInteger1 = new BigInteger(val: "12345456546546678909876543214534534534");
25     BigInteger bigInteger2 = new BigInteger(val: "214145124546546546345345124123112");
26     BigInteger add = bigInteger1.add(bigInteger2);
27     System.out.println("add : " + add);
28     BigInteger subtract = bigInteger1.subtract(bigInteger2);
29     System.out.println("subtract : " + subtract);
30     BigInteger multiply = bigInteger1.multiply(bigInteger2);
31     System.out.println("multiply : " + multiply);
32     BigInteger divide = bigInteger1.divide(bigInteger2);
33     System.out.println("divide : " + divide);
34 }
35
36
```

```
PS E:\HOC\nam3-ki2\Thực Tập Cơ Sở RSA> e.; cd 'e:\HOC\nam3-ki2\Thực Tập Cơ Sở RSA'; & 'C:\Program Files\Java\jdk-16.0.2\bin\java.exe'
ocket,server=n,suspend=y,address=localhost:56921' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\dinh\AppData\Roaming\Code\
ce251e3c42a9cb56acb852f\redhat.java\jdt_ws\RSA_73b77a81\bin' 'makhoaRSA'
add :12345670691671225456422888559658657646
subtract :12345242401422132363330197869410411422
multiply :2643719329744216965515315980755574926667253438342927983154287831549808
divide :57649
```

- Lập trình giải thuật mã hóa và giải mã
 - + Sử dụng thư viện BigInteger để triển khai mã hóa RSA

```
import java.io.UnsupportedEncodingException;
import java.math.BigInteger;
import java.util.Scanner;

public class makhoaRSA {
    Run | Debug
    public static void main(String[] args) throws Unsu
```

```
PS E:\HOC\nam3-ki2\Thực Tập Cơ Sở RSA> e.; cd 'e:\HOC\nam3-ki2\Thực Tập Cơ Sở RSA'; & 'C:\Program Files\Java\jdk-16.0.2\bin\java.exe'
ocket,server=n,suspend=y,address=localhost:56921' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\dinh\AppData\Roaming\Code\
ce251e3c42a9cb56acb852f\redhat.java\jdt_ws\RSA_73b77a81\bin' 'makhoaRSA'
add :12345670691671225456422888559658657646
subtract :12345242401422132363330197869410411422
multiply :2643719329744216965515315980755574926667253438342927983154287831549808
divide :57649
```

- + Khởi báo các giá trị của mã hóa RSA

```
import java.math.BigInteger;
import java.util.Random;

public class RSA {
    public static final int VERSION = 2048;
    public static final BigInteger E = new BigInteger("3");
    private BigInteger p;
    private BigInteger q;
    private BigInteger n;
    private BigInteger phiN;
    private BigInteger d;
```

```
PS E:\HOC\nam3-ki2\Thực Tập Cơ Sở RSA> e.; cd 'e:\HOC\nam3-ki2\Thực Tập Cơ Sở RSA'; & 'C:\Program Files\Java\jdk-16.0.2\bin\java.exe'
ocket,server=n,suspend=y,address=localhost:56921' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\dinh\AppData\Roaming\Code\
ce251e3c42a9cb56acb852f\redhat.java\jdt_ws\RSA_73b77a81\bin' 'makhoaRSA'
add :12345670691671225456422888559658657646
subtract :12345242401422132363330197869410411422
multiply :2643719329744216965515315980755574926667253438342927983154287831549808
divide :57649
```

- + Khởi tạo các giá trị của mã hóa RSA

```

public void init() {
    p = BigInteger.probablePrime(VERSION / 2, new Random());
    q = BigInteger.probablePrime(VERSION / 2, new Random());
    n = p.multiply(q);
    phiN = (p.subtract(BigInteger.ONE)).multiply(q.subtract(BigInteger.ONE));
    d = E.modInverse(phiN);
}

public BigInteger encrypt(BigInteger message, BigInteger partnerN) {
    return message.modPow(E, partnerN);
}

public BigInteger decrypt(BigInteger cipher) {
    return cipher.modPow(d, n);
}

public BigInteger getN() {
    return n;
}

```

Command Prompt

Microsoft Windows [Version 10.0.22000.675]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dinh>echo dinhquanghieu-b19dcat065
dinhquanghieu-b19dcat065

C:\Users\dinh>

- Mã hóa 6 số:

```

11 RSA q = new RSA();
12 p.init();
13 q.init();
14 BigInteger message = new BigInteger(string.getBytes());
15 BigInteger cipher = p.encrypt(message, q.getN());
16 BigInteger decrypted = q.decrypt(cipher);
17 System.out.println(x: "Code ma hoa");
18 System.out.println(cipher);
19 System.out.println(x: "thong tin giai ma");
20 String result = new String(decrypted.toByteArray());
21 System.out.println(result);
22 scanner.close();

```

Command Prompt

Microsoft Windows [Version 10.0.22000.675]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dinh>echo dinhquanghieu-b19dcat065
dinhquanghieu-b19dcat065

C:\Users\dinh>

PROBLEMS OUTPUT **TERMINAL** DEBUG CONSOLE

ce251e3c42a9c56ac852f\redhat.java\jdk_ws\RSA_73b77a81\bin' 'maKhoaRSA'

Nhap du lieu muon ma hoa va giai ma

123456

Code ma hoa

102415131661437490440182885641258589369226542937783979548834456167123526789777267197020730623925156384410307624457463526420766381193446193477580959266474311790692917
179685934472982214727252558953843184703709855876907269093403002716453630871449732357824474672947020197689277290721213065245318216729173977338044462156108181483409
05513738270241487593221738840128293890686801688275551287011456399771334814657612866128414816431749446756381341144286417701198099705394890168993944907590882687389948
7771510789363184770236408528509578315222779005247447638753076164628806482853156681888893203613021781336757525009523023051

thong tin giai ma

123456

PS E:\HOC\nam3-ki2\Thuc Tap Co So\RSA>

- Mã hóa với số cực lớn

```

12 p.init();
13 q.init();
14 BigInteger message = new BigInteger(string.getBytes());
15 BigInteger cipher = p.encrypt(message, q.getN());
16 BigInteger decrypted = q.decrypt(cipher);
17 System.out.println(x: "Code ma hoa");
18 System.out.println(cipher);
19 System.out.println(x: "thong tin giai ma");
20 String result = new String(decrypted.toByteArray());
21 System.out.println(result);
22 scanner.close();

```

Command Prompt

Microsoft Windows [Version 10.0.22000.675]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dinh>echo dinhquanghieu-b19dcat065
dinhquanghieu-b19dcat065

C:\Users\dinh>

PROBLEMS OUTPUT **TERMINAL** DEBUG CONSOLE

PS E:\HOC\nam3-ki2\Thuc Tap Co So\RSA> e; cd 'e:\HOC\nam3-ki2\Thuc Tap Co So\RSA'; & 'C:\Program Files\Java\jdk-16.0.2\bin\java.exe' '-agentlib:jdwp=transport=dt_s
ocket,server=n,suspend=y,address=localhost:56824' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\dinh\AppData\Roaming\Code\User\workspacestorage\668e7c01
ce251e3c42a9c56ac852f\redhat.java\jdk_ws\RSA_73b77a81\bin' 'maKhoaRSA'

Nhap du lieu muon ma hoa va giai ma

9324793274239847239847238976482734698273468273

Code ma hoa

35226411741646241708721643522166272545950158025005440433036505347456503483481364624812790472545778870907443287806054451187040991145463760174291807877225844515066288
9709219160890025785492284965926262923362318963990089319608362607098406250940087553379579188647358370759248148569844650283395533251780669124702245780944104548647971842
92021381634218331719049326357486737153078777550982615324794032678887719510652925537181851732455740001712800260520348989722678078722028286846216526609443486937572519
3272246694420878205752943100299558120503566142212094298222100325377427873935984468181096602747845094397059306777923297484

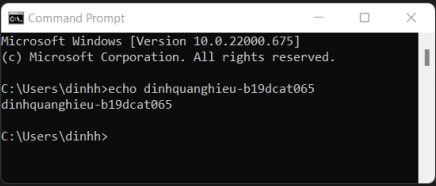
thong tin giai ma

9324793274239847239847238976482734698273468273

PS E:\HOC\nam3-ki2\Thuc Tap Co So\RSA>

- Mã hóa xâu kí tự dài và có nhiều kí tự đặc biệt như *,#...

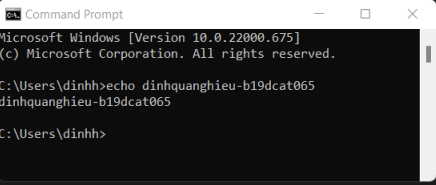
```
11 RSA q = new RSA();
12 p.init();
13 q.init();
14 BigInteger message = new BigInteger(string.getBytes());
15 BigInteger cipher = p.encrypt(message, q.getN());
16 BigInteger decrypted = q.decrypt(cipher);
17 System.out.println(x: "Code ma hoa");
18 System.out.println(cipher);
19 System.out.println(x: "thong tin giai ma");
20 String result = new String(decrypted.toByteArray());
21 System.out.println(result);
22 scanner.close();
```



```
PS E:\HOC\nam3-ki2\Thuc Tap Co So RSA> e;; cd 'e:\HOC\nam3-ki2\Thuc Tap Co So RSA'; & 'C:\Program Files\Java\jdk-16.0.2\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:56839' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\dinh\AppData\Roaming\Code\User\workspaceStorage\668e7c01ce251e3c42a9cbc56acb852f\redhat.java\jdt_ws\RSA_73b77a81\bin' 'makhoaRSA'
Nhap du lieu muon ma hoa va giai ma
sodifhsdihfsdi//uyigu##sdfdsdfdeifmsdLfmsdfmsd
Code ma hoa
2323425605044204935296187367714603364376478017183180398431462737183453166382406859326977316440849655230563192924192004166117536023231853754011250639384714252564389
029413822324170456474981685726122596096242960786268722678638466831900893113963756127600033691942664385807721320316612996784948631177110604630412659597336128359968547
256829476946837251804781409326951446709066763671063369130263488059340408616848435869140508499335133091279835283925047176915359621949736275172294018164585763073929450
39505811944916051472871930047161137613536367744519065170132242023409062886144828258737029782321535031918808200234863192685
thong tin giai ma
sodifhsdihfsdi//uyigu##sdfdsdfdeifmsdLfmsdfmsd
PS E:\HOC\nam3-ki2\Thuc Tap Co So RSA> |
```

- Mã hóa xâu kí tự “I am B19DCAT065”

```
9 String string = scanner.nextLine();
10 RSA p = new RSA();
11 RSA q = new RSA();
12 p.init();
13 q.init();
14 BigInteger message = new BigInteger(string.getBytes());
15 BigInteger cipher = p.encrypt(message, q.getN());
16 BigInteger decrypted = q.decrypt(cipher);
17 System.out.println(x: "Code ma hoa");
18 System.out.println(cipher);
19 System.out.println(x: "thong tin giai ma");
20 String result = new String(decrypted.toByteArray());
21 System.out.println(result);
22 scanner.close();
```



```
PS E:\HOC\nam3-ki2\Thuc Tap Co So RSA> e;; cd 'e:\HOC\nam3-ki2\Thuc Tap Co So RSA'; & 'C:\Program Files\Java\jdk-16.0.2\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:56858' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\dinh\AppData\Roaming\Code\User\workspaceStorage\668e7c01ce251e3c42a9cbc56acb852f\redhat.java\jdt_ws\RSA_73b77a81\bin' 'makhoaRSA'
Nhap du lieu muon ma hoa va giai ma
I am B19DCAT065
Code ma hoa
209478586594247051318045947672447948570080847870665477500612563986687466688591718538456122983571237237087884818021581140098712806350527927157424225370861438430104603
237298877103021381840239151518319580964363294435325143640140285292495271083645324943558034306929070348375124142532797610194504601960277442978491852355142396230083344
2095150670816468029540461762853067029132827032553930804698425198978336541904910444005195177378056423859894573072044365946277678902625673095396498151733391361067250446
1530869703153260166358061552692056687056324262587778350923528703088147936951864014275629532918896238836491494387638438217
thong tin giai ma
I am B19DCAT065
```