

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA AN TOÀN THÔNG TIN

BỘ MÔN THỰC TẬP CƠ SỞ



BÀI 16:
LẬP TRÌNH THUẬT
TOÁN MẬT MÃ HỌC

Giảng viên : Nguyễn Ngọc Điệp

Sinh viên : Nguyễn Đức Anh

Mã sinh viên : B21DCAT031

Hệ : Đại học chính quy

Hà Nội, 4/2024

Table of Contents

1. Mục đích	3
2. Nội dung thực hành	3
2.1 Tìm hiểu lý thuyết.....	3
2.2 Chuẩn bị môi trường	3
2.3 Các bước thực hiện	3
3. Kết quả đạt được.....	7

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.

2. Nội dung thực hành

2.1 Tìm hiểu lý thuyết

a. Lập trình với số lớn

Việc lập trình với những số có độ dài hàng nghìn bit là rất khó khăn. Thay vì tự viết hàm tính toán các số lớn, sinh viên sử dụng class BigInteger có sẵn trong Java chuyên để xử lý các số lớn.

Class BigInteger cũng cung cấp những phép toán cơ bản như cộng add(), trừ subtract(), nhân multiply(), chia divide(), giúp việc tính toán các phép toán cơ bản dễ dàng hơn.

Ngoài ra, BigInteger còn cung cấp hàm lũy thừa lấy phần dư modpow() hay hàm nghịch đảo modulo modInverse() giúp việc lập trình mã hoá và giải mã RSA dễ dàng hơn.

b. Giải thuật mật mã khóa công khai RSA

Thuật toán mã hoá RSA là thuật toán mã hoá khóa công khai được sử dụng rộng rãi để truyền dữ liệu an toàn

Thuật toán mã hoá RSA được phát triển bởi Rivest, Shamir, Adleman. Quy trình mã hoá của RSA được công khai năm 1977.

Độ an toàn của RSA liên hệ chặt chẽ với độ khó của bài toán phân tích nhân tử của một số rất lớn thành hai thừa số nguyên tố. Hiện nay vẫn chưa có siêu máy tính nào có thể giải bài toán này với thời gian chấp nhận được, nhưng trong tương lai với máy tính lượng tử có thể sẽ khả thi.

Quy trình mã hoá:

- Chọn hai số nguyên tố lớn p và q và tính $N = p \cdot q$. Cần chọn p và q sao cho $M < 2^{(i-1)} < N < 2^i$
- Tính $\Phi(n) = (p - 1)(q - 1)$
- Tìm một số e sao cho: e và $\Phi(n)$ là 2 số nguyên tố cùng nhau và $0 < e < \Phi(n)$
- Tìm một số d sao cho: $e \cdot d \bmod \Phi(n) = 1$ (hay: $d = e^{-1} \bmod \Phi(n)$)
- Chọn khóa công khai $K1$ là cặp (N, e) , khóa riêng $K2$ là cặp (N, d) .
- Mã hoá $C = M^e \bmod N$, hoặc $C = M^d \bmod N$ nếu mã hoá chứng thực.
- Giải mã $M = C^d \bmod N$, hoặc $M = C^e \bmod N$ nếu chứng thực.

2.2 Chuẩn bị môi trường

- Môi trường lập trình theo mong muốn.

2.3 Các bước thực hiện

```
rsa.py X
rsa.py > is_prime
1 import random
2
3 def is_prime(n): #kiem tra nguyen to
4     if n <= 1:
5         return False
6     elif n <= 3:
7         return True
8     elif n % 2 == 0 or n % 3 == 0:
9         return False
10    i = 5
11    while i * i <= n:
12        if n % i == 0 or n % (i + 2) == 0:
13            return False
14        i += 6
15    return True
16
17 def gcd(a, b): #tim uoc chung lon nhat
18     while b != 0:
19         a, b = b, a % b
20     return a
21
22 def mod_inverse(a, m): #nghich dao modulo
23     m0, x0, x1 = m, 0, 1
24     while a > 1:
25         q = a // m
26         m, a = a % m, m
27         x0, x1 = x1 - q * x0, x0
28     return x1 + m0 if x1 < 0 else x1
29
30 def generate_keypair(p, q): #tao khoa tu cap so p,q
31     if not (is_prime(p) and is_prime(q)):
32         raise ValueError("Both numbers must be prime.")
33     elif p == q:
34         raise ValueError("p and q cannot be equal.")
35
36     #tinh n va phi(n)
37     n = p * q
38     phi = (p - 1) * (q - 1)
39
40     #tinh e
41     e = random.randrange(1, phi)
42     g = gcd(e, phi)
43     while g != 1:
44         e = random.randrange(1, phi)
45         g = gcd(e, phi)
46
Command Prompt for vctl - date
C:\>echo B21DCAT031 Nguyen Duc Anh && date
B21DCAT031 Nguyen Duc Anh
The current date is: 14/04/2024
Enter the new date: (dd-mm-yy)
```

```
rsa.py X
rsa.py > is_prime
21
22 def mod_inverse(a, m): #nghich dao modulo
23     m0, x0, x1 = m, 0, 1
24     while a > 1:
25         q = a // m
26         m, a = a % m, m
27         x0, x1 = x1 - q * x0, x0
28     return x1 + m0 if x1 < 0 else x1
29
30 def generate_keypair(p, q): #tao khoa tu cap so p,q
31     if not (is_prime(p) and is_prime(q)):
32         raise ValueError("Both numbers must be prime.")
33     elif p == q:
34         raise ValueError("p and q cannot be equal.")
35
36     #tinh n va phi(n)
37     n = p * q
38     phi = (p - 1) * (q - 1)
39
40     #tinh e
41     e = random.randrange(1, phi)
42     g = gcd(e, phi)
43     while g != 1:
44         e = random.randrange(1, phi)
45         g = gcd(e, phi)
46
Command Prompt for vctl - date
C:\>echo B21DCAT031 Nguyen Duc Anh && date
B21DCAT031 Nguyen Duc Anh
The current date is: 14/04/2024
Enter the new date: (dd-mm-yy)
```

The screenshot shows a Python IDE with a file named `rsa.py`. The code implements the RSA algorithm. The first part defines `is_prime`, `gcd`, `mod_inverse`, and `generate_keypair` functions. The second part defines `encrypt` and `decrypt` functions. The third part defines a `generate_prime` function. The fourth part generates random prime numbers `p` and `q`, prints them, and generates a keypair. The fifth part encrypts a message and prints the encrypted message. The sixth part decrypts the message and prints the decrypted message.

```
rsa.py > is_prime
30 def generate_keypair(p, q): #tao khoa tu cap so p,q
40     #tinh e
41     e = random.randrange(1, phi)
42     g = gcd(e, phi)
43     while g != 1:
44         e = random.randrange(1, phi)
45         g = gcd(e, phi)
46
47     d = mod_inverse(e, phi) # tim d sao cho (d*e) mod phi =1
48
49     return ((e, n), (d, n))
50
51 def encrypt(public_key, plaintext):# ham ma hoa
52     key, n = public_key
53     cipher = [pow(ord(char), key, n) for char in plaintext]
54     return cipher
55
56 def decrypt(private_key, cipher):# ham giai ma
57     key, n = private_key
58     plain = [chr(pow(char, key, n)) for char in cipher]
59     return ''.join(plain)
60 def generate_prime():
61     while True:
62         num = random.randint(10, 1000) # Thay đổi phạm vi tùy ý
63         if is_prime(num):
64             return num
65
66 # tao cap so p,q
67 p = generate_prime()
68 q = generate_prime()
69
70 print("Random prime number p:", p)
71 print("Random prime number q:", q)
72 public_key, private_key = generate_keypair(p, q)
73
74 # Encrypt
75 message = "112233445566778899112233445566778899"
76 encrypted_msg = encrypt(public_key, message)
77 print("message:",message)
78 print("Encrypted message:", "".join(map(str, encrypted_msg)))
79
80 # Decrypt
81 decrypted_msg = decrypt(private_key, encrypted_msg)
82 print("Decrypted message:", decrypted_msg)
```

Command Prompt for vctl - date

```
C:\>echo B21DCAT031 Nguyen Duc Anh && date
B21DCAT031 Nguyen Duc Anh
The current date is: 14/04/2024
Enter the new date: (dd-mm-yy)
```

Ở đây ta sử dụng ngôn ngữ Python để triển khai thuật toán mã hóa RSA

- Tạo `p, q` là một cặp số nguyên tố ngẫu nhiên (tạm thời giới hạn trong phạm vi 10-1000)
- Các hàm được sử dụng:

- hàm `is_prime`: kiểm tra nguyên tố
- hàm `gcd`: tìm ước chung lớn nhất
- hàm `mod_inverse`: tính nghịch đảo modulo
- hàm `generate_keypair`: tạo cặp khóa công khai và khóa bí mật từ cặp số `p, q`
- hàm `encrypt`: hàm thực hiện mã hóa thông tin. Cần chú ý thông tin mã hóa cần phải

được chuyển thành dạng byte vì các thông tin được truyền thường ở dạng chuỗi ký tự

- hàm decrypt: hàm thực hiện giải mã thông tin
- hàm generate_prime: hàm tạo số nguyên tố ngẫu nhiên

Thử nghiệm với số lớn:

```
rsa.py x
rsa.py > ...
71 public_key, private_key = generate_keypair(p, q)
72
73 # Encrypt
74 message = "112233445566778899112233445566778899"
75 encrypted_msg = encrypt(public_key, message)
76 print("message:", message)
77 print("Encrypted message:", "".join(map(str, encrypted_msg)))
78
79 # Decrypt
80 decrypted_msg = decrypt(private_key, encrypted_msg)
81 print("Decrypted message:", decrypted_msg)
82
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\TTCS> & C:/Users/admin/AppData/Local/Programs/Python/Python312/python.exe d:/TTCS/rsa.py

Random prime number p: 647
Random prime number q: 601
message: 112233445566778899112233445566778899
Encrypted message: 35951935951935178035178038789738789733867033867019181219181228569128569122924229242571162571163034343034343595193595193517803517803878973878973386703386701918121918122856912856912292422924257116257116303434303434
Decrypted message: 112233445566778899112233445566778899
PS D:\TTCS>

Select Command Prompt for vctl - date

C:\>echo B21DCAT031 Nguyen Duc Anh && date
B21DCAT031 Nguyen Duc Anh
The current date is: 14/04/2024
Enter the new date: (dd-mm-yy)

Thử nghiệm mã hóa và giải mã chuỗi ký tự: “I am B21DCAT031”

```
rsa.py x
rsa.py > ...
71 public_key, private_key = generate_keypair(p, q)
72
73 # Encrypt
74 message = "I am B21DCAT031"
75 encrypted_msg = encrypt(public_key, message)
76 print("message:", message)
77 print("Encrypted message:", "".join(map(str, encrypted_msg)))
78
79 # Decrypt
80 decrypted_msg = decrypt(private_key, encrypted_msg)
81 print("Decrypted message:", decrypted_msg)
82
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\TTCS> & C:/Users/admin/AppData/Local/Programs/Python/Python312/python.exe d:/TTCS/rsa.py

Random prime number p: 727
Random prime number q: 751
message: I am B21DCAT031
Encrypted message: 78928305851238587275015305851195853064891791684141292912449609748299318231206224179168
Decrypted message: I am B21DCAT031
PS D:\TTCS>

Select Command Prompt for vctl - date

C:\>echo B21DCAT031 Nguyen Duc Anh && date
B21DCAT031 Nguyen Duc Anh
The current date is: 14/04/2024
Enter the new date: (dd-mm-yy)

3. Kết quả đạt được

- Lập trình thành công thuật toán mã hóa RSA sử dụng ngôn ngữ Python.
- Đáp ứng yêu cầu mã hóa/giải mã hoạt động tốt với phạm vi số lớn.