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*

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 lí 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 luỹ 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.

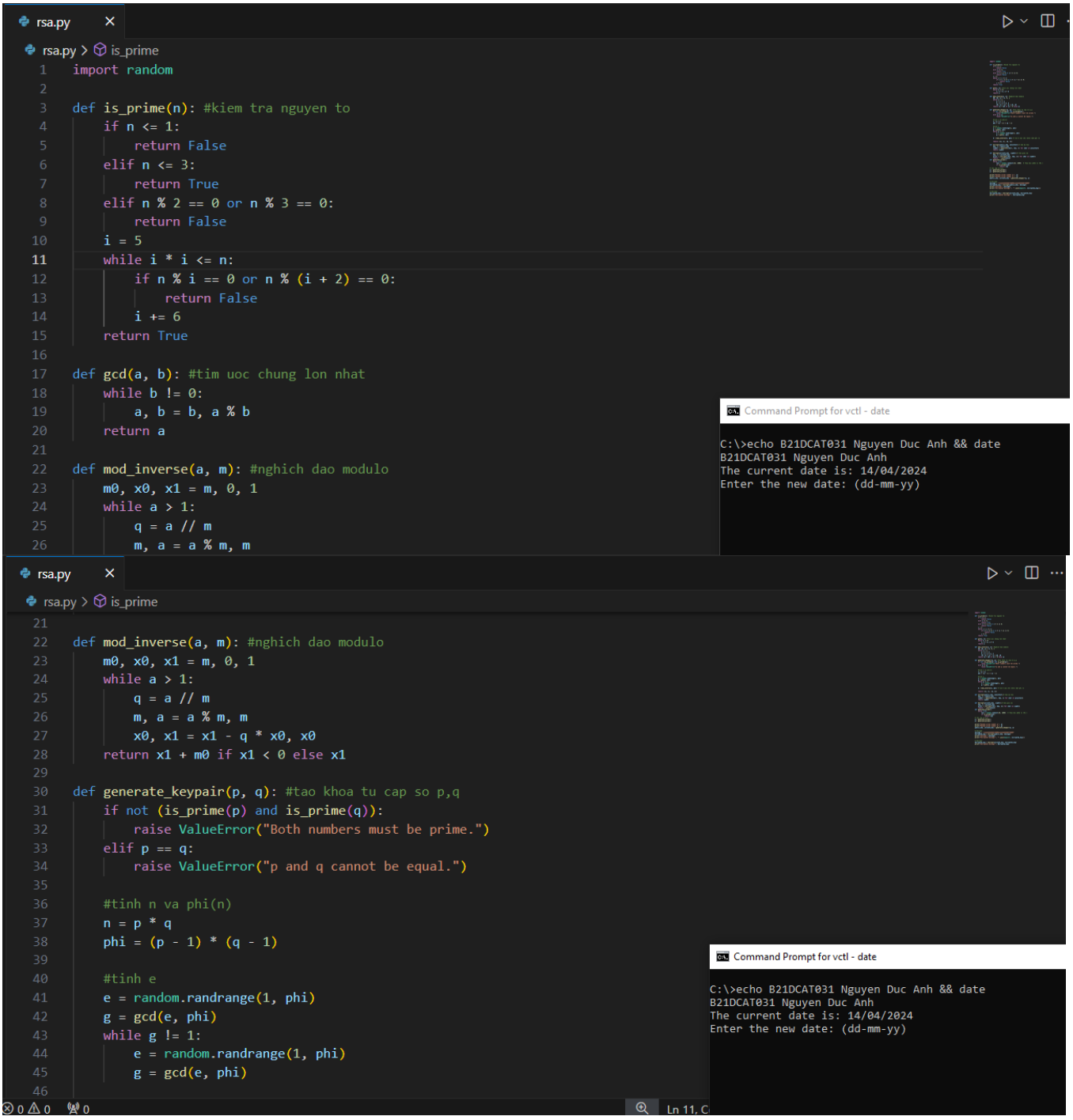
Quy trình mã hoá:

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

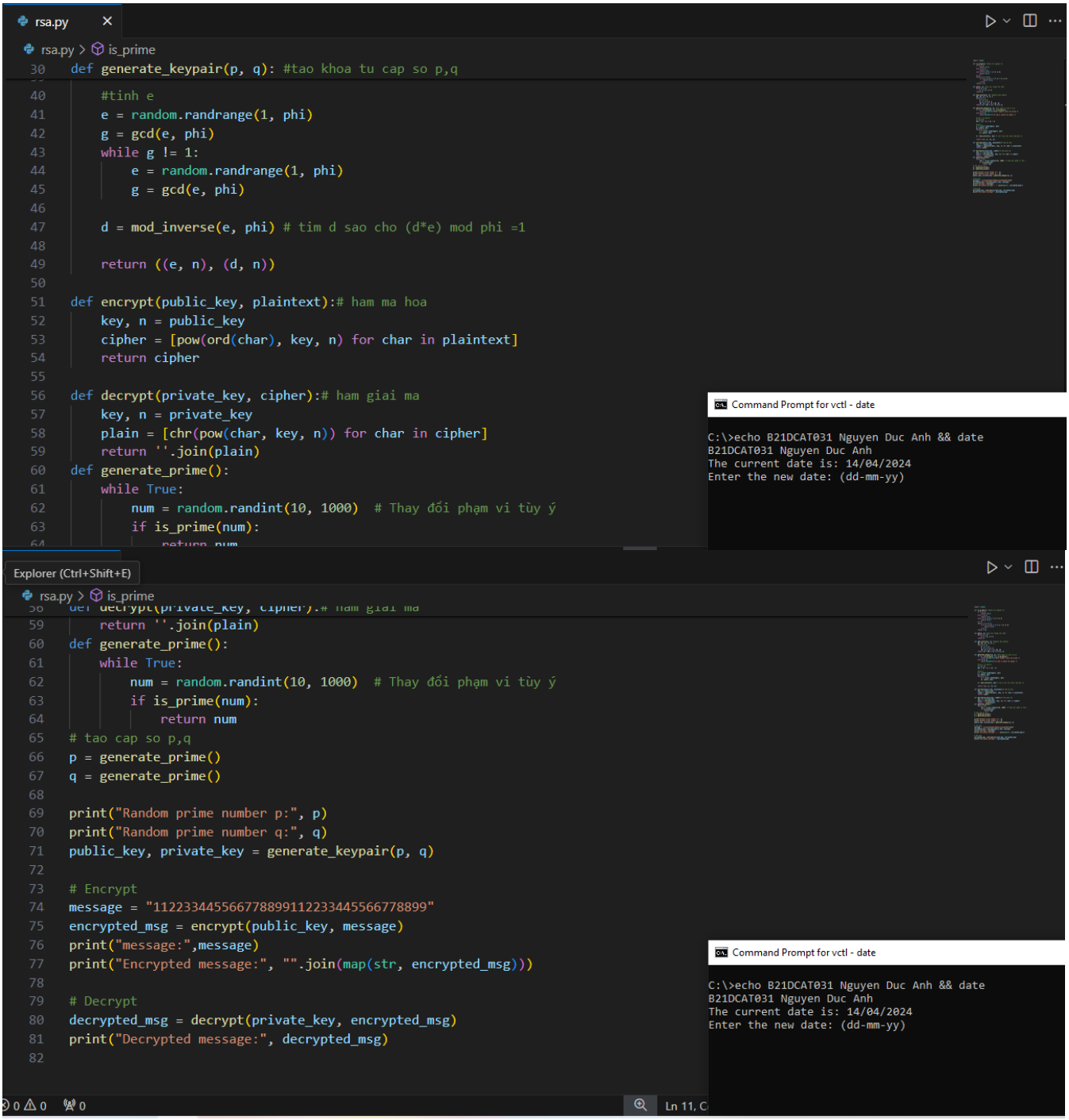
2.2 Chuẩn bị môi trường

- Môi trường lập trình theo mong muốn.

2.3 Các bước thực hiện


rsa.py
rsa.py > is_prime
import random
def is_prime(n): #kiem tra nguyen to
if n <= 1:
return False
elif n <= 3:
return True
elif n % 2 == 0 or n % 3 == 0:
return False
i = 5
while i * i <= n:
if n % i == 0 or n % (i + 2) == 0:
return False
i += 6
return True
def gcd(a, b): #tim uoc chung lon nhat
while b != 0:
a, b = b, a % b
return a
def mod_inverse(a, m): #nghich dao modulo
m0, x0, x1 = m, 0, 1
while a > 1:
q = a // m
m, a = a % m, m
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
rsa.py > is_prime
def mod_inverse(a, m): #nghich dao modulo
m0, x0, x1 = m, 0, 1
while a > 1:
q = a // m
m, a = a % m, m
x0, x1 = x1 - q * x0, x0
return x1 + m0 if x1 < 0 else x1
def generate_keypair(p, q): #tao khoa tu cap so p,q
if not (is_prime(p) and is_prime(q)):
raise ValueError("Both numbers must be prime.")
elif p == q:
raise ValueError("p and q cannot be equal.")
#tinh n va phi(n)
n = p * q
phi = (p - 1) * (q - 1)
#tinh e
e = random.randrange(1, phi)
g = gcd(e, phi)
while g != 1:
e = random.randrange(1, phi)
g = gcd(e, phi)
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)

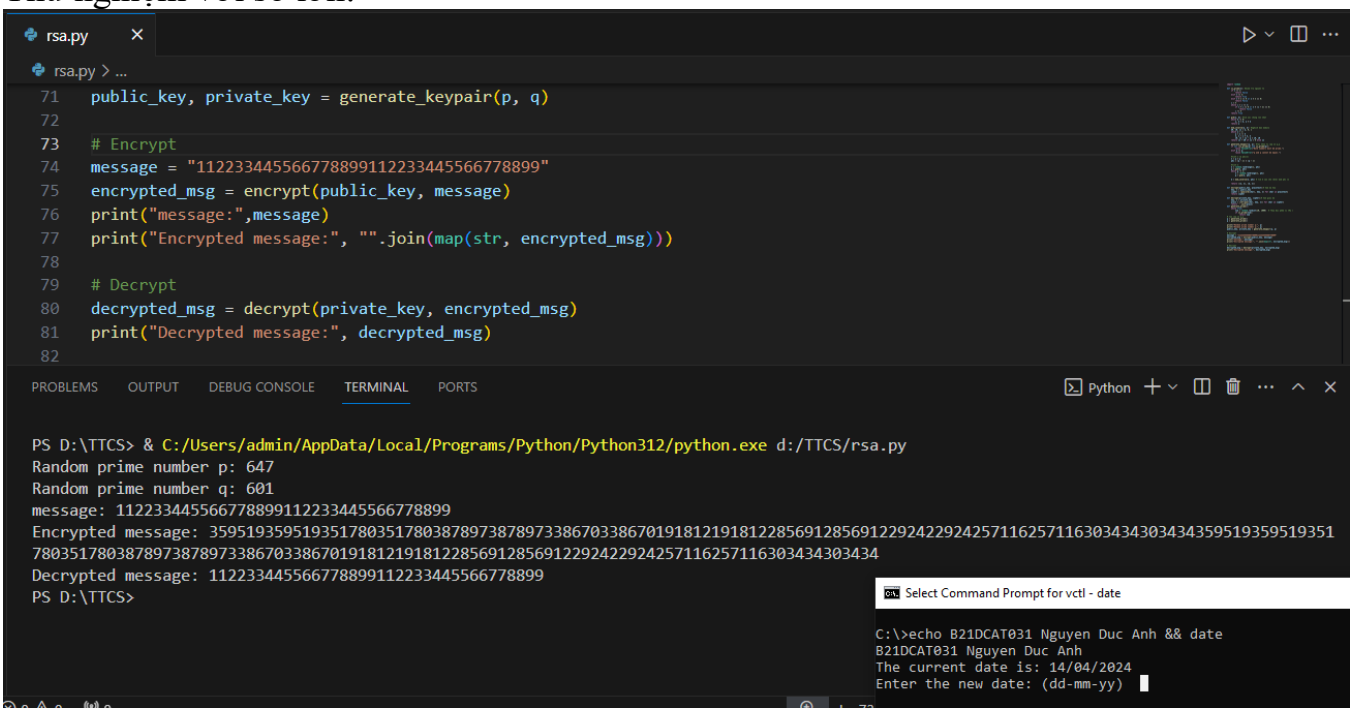


Ở đây ta sử dụng ngôn ngữ Python để triển khai thuật toán mã hóa RSA
- Tạo p,q là một cặp số nguyên tố ngẫu nhiên (tạm thời giới hạn trong phạm vi 10-1000)
- Các hàm được sử dụng:
    - hàm is_prime: kiểm tra nguyên tố
    - hàm gcd: tìm ước chung lớn nhất
    - hàm mod_inverse: tính nghịch đảo modulo
    - 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

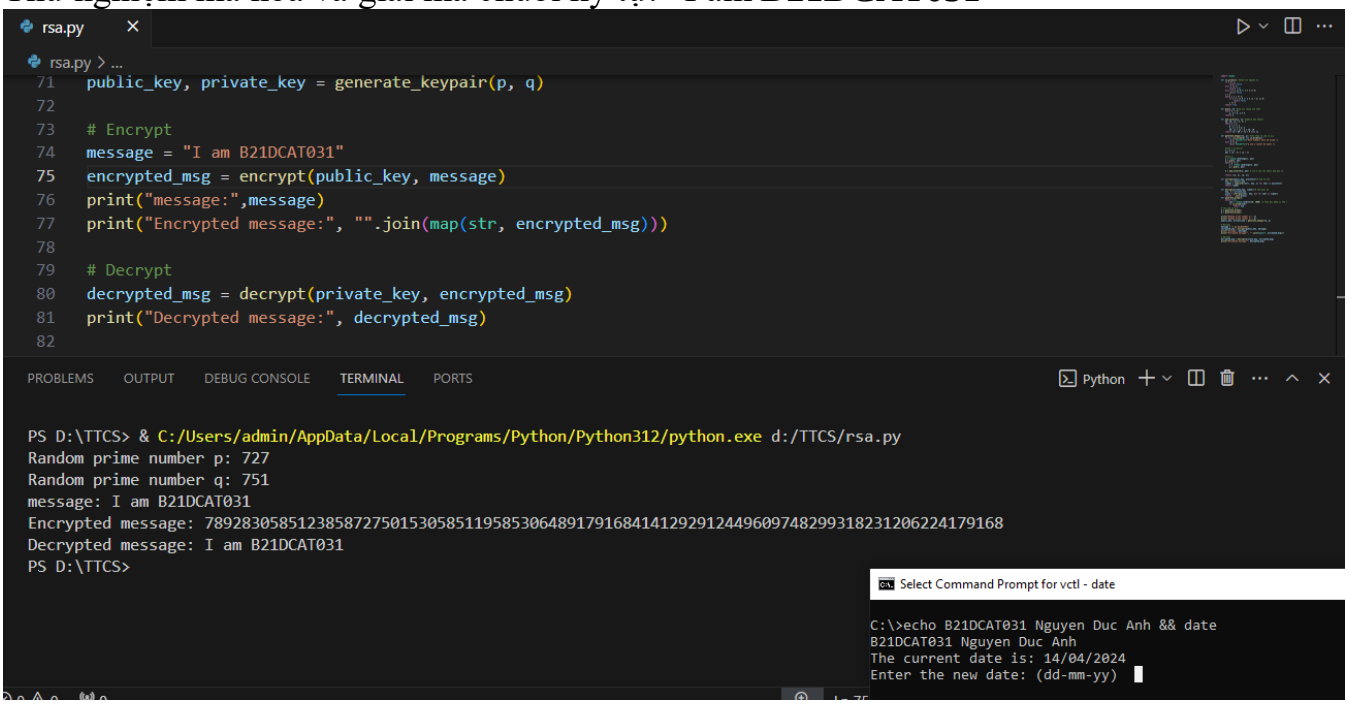
được chuyển thành dạng byte vì các thông tin được truyền thường ở dạng chuỗi ký tự

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

Thử nghiệm với số lớn:



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



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.