**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA AN TOÀN THÔNG TIN**

****

**MÔN HỌC THỰC TẬP CƠ SỞ**

**BÀI THỰC HÀNH SỐ 16**

**Lập trình thuật toán mật mã học**

**Giảng viên hướng dẫn :** Đỗ Xuân Chợ

**Sinh viên thực hiện      :** Hoàng Trung Kiên

**Lớp                                :**  D20CQAT02-B

**Mã sinh viên                    :** B20DCAT098

 Hà nội – 2/2023

**I. Tìm hiểu lý thuyết**

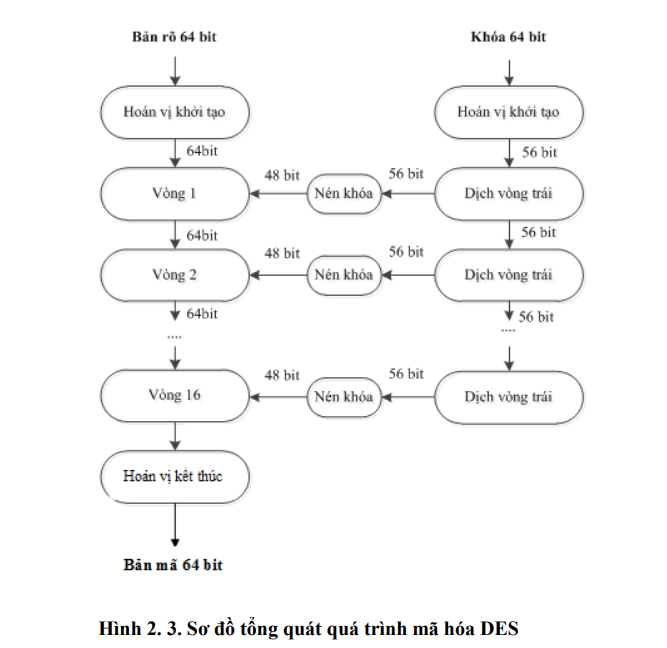
1. Tìm hiểu về thuật toán mã hóa đối xứng

a, Thuật toán DES:

- DES được phát triển tại IBM vào đầu những năm 1970 và được thừa nhận là chuẩn mã hóa tại Mỹ (NSA) vào năm 1976. Sau đó DES được sử dụng rộng rãi trong những năm 70 và 80. Mã hóa DES có các tính chất sau:

- Là mã thuộc hệ mã Feistel gồm 16 vòng, ngoài ra DES có thêm một hoán vị khởi tạo trước khi vào vòng 1 và một hoán vị kết thúc sau vòng 16.

- Kích thước của khối là 64 bit. biểu diễn theo mã ASCII thì mã DES sẽ mã hóa làm 3 lần, mỗi lần 8 chữ cái (64 bit). Kích thước khóa là 64 bit (thực chất chỉ có 56 bit được sử dụng để mã hóa); Mỗi vòng của DES dùng khóa con có kích thước 48 bit được trích ra từ khóa chính.

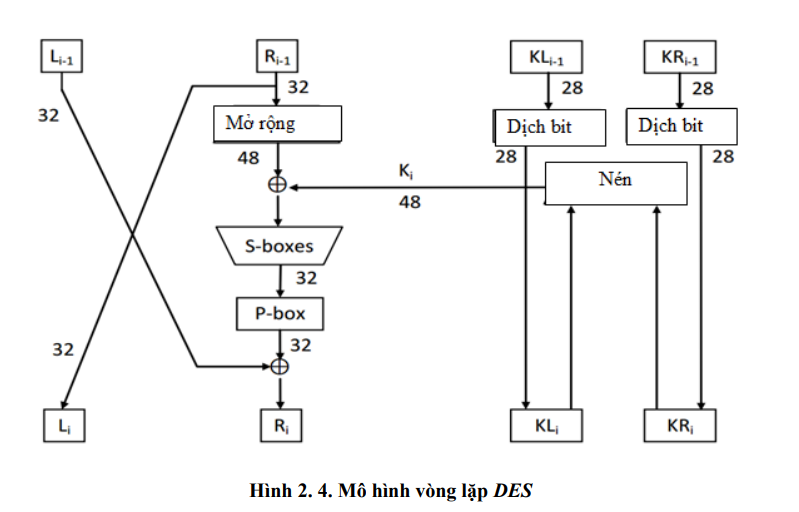


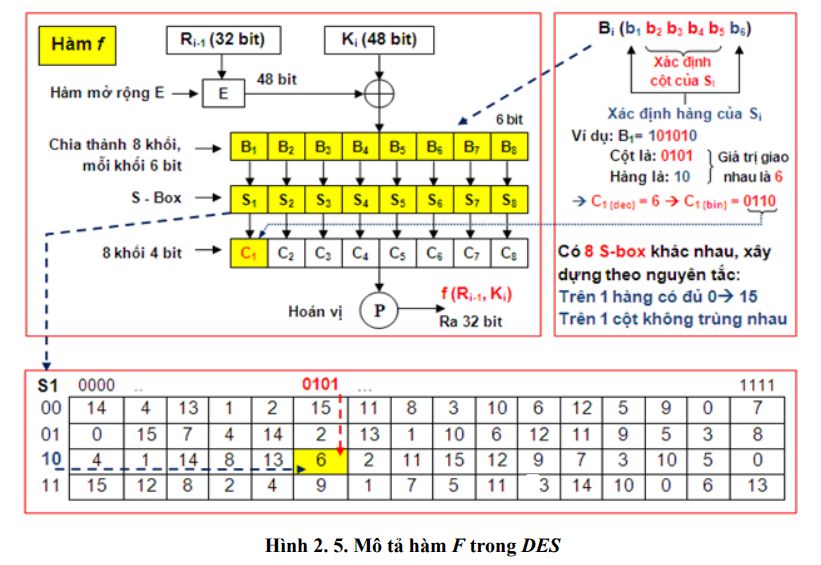
-Sơ đồ mã hóa DES gồm 3 phần, phần thứ nhất là các hoán vị khởi tạo và hoán vị kết thúc. Phần thứ hai là các vòng Feistel, phần thứ ba là thuật toán sinh khóa con.

-Hoán vị khởi tạo và hoán vị kết thúc:

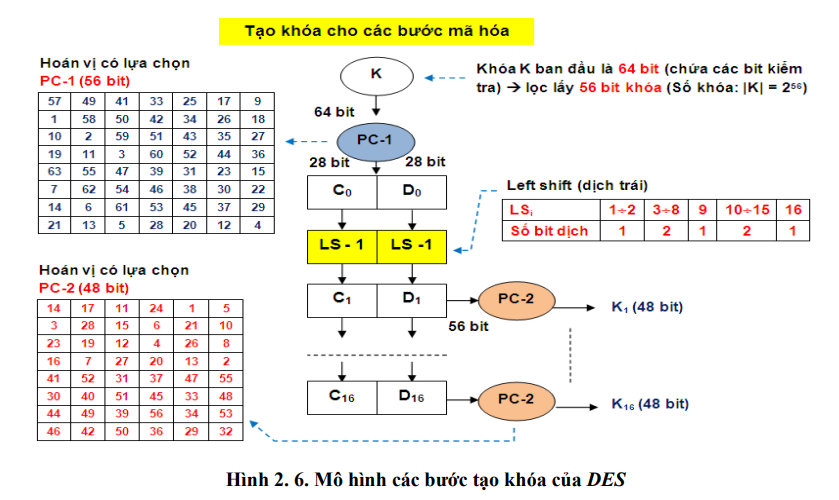
+Hoán vị kết thúc chính là hoán vị nghịch đảo của hoán vị khởi tạo. Đối với knownplaintext hay chosen-plaintext attack, hoán vị khởi tạo và hoán vị kết thúc không có ý nghĩa bảo mật, sự tồn tại của hai hoán vị trên được nhận định là do yếu tố lịch sử.

-Các vòng của DES:





-Thuật toán sinh khóa con



b, Thuật toán AES:

-Vào năm 1997 nhận thấy nguy cơ mã hóa DES là kích thước khóa ngắn có thể bị phá vỡ, do đó Viện chuẩn quốc gia Hoa Kỳ US NIST đã kêu gọi xây dựng một phương pháp mã hóa mới. Đến tháng 10/2000 thuật toán có tên là Rijndael được chọn làm mật mã nâng cao và xuất bản là chuẩn FIPS PUB 197. Vào 11/2001 và được đổi tên thành Andvanced Encryption Standard (AES). Như đã trình bày ở trên, thuật toán mã hóa DES được thực hiện dựa trên mã hóa khối Feistel. Tuy nhiên, cơ chế mã hóa của AES lại có điểm khác với DES là sử dụng mạng thay thế hoàn vị (SPN - Substitution-Permutation Network).

-Thuật toán AES Thuật toán AES khá phức tạp, được mô tả khái quát gồm 3 bước như sau:

+Vòng khởi tạo chỉ gồm phép AddRoundKey

+Vòng lặp gồm 4 phép biển đổi lần lượt: SubBytes, ShiftRows, MixColumns, AddRoundKey. +Vòng cuối gồm các phép biến đổi giống vòng lặp và không có phép MixColumns.

-Đặc tả chi tiết thuật toán mã hóa AES Cấu trúc sơ đồ mã hóa được thể hiện trên Hình 2.9 gồm:

+Vòng khởi tạo, Nr-1 vòng lặp và vòng kết thúc. Trong đó vòng khởi tạo chỉ có phép biến đổi AddRounKey,

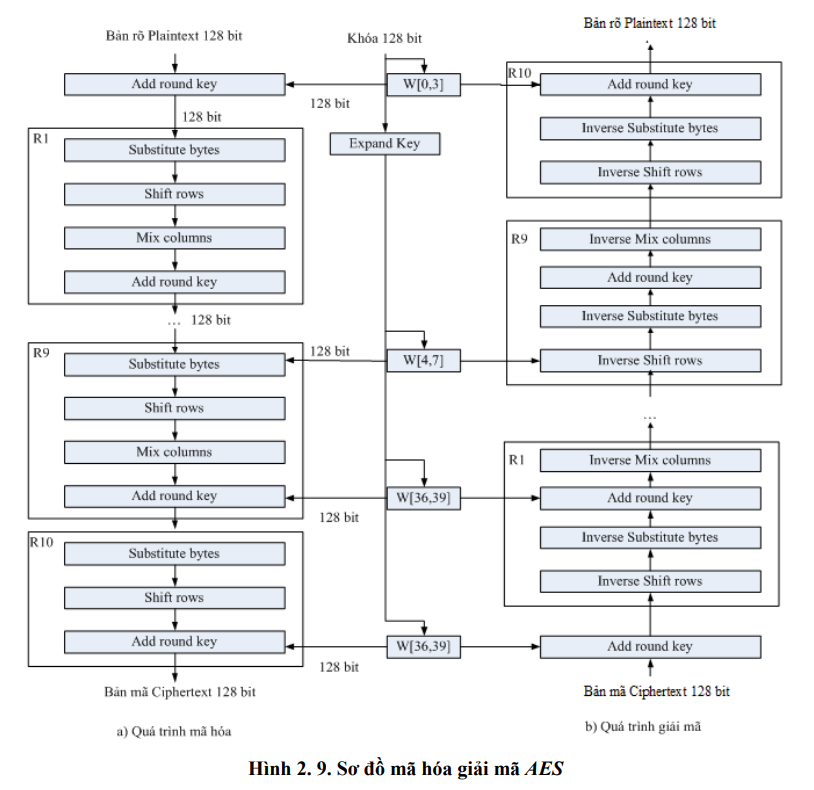
+Vòng lặp gồm 4 phép biến đổi chính: SubBytes, ShiftRows, MixColumns, AddRounKey; vòng kết thúc khác với vòng lặp chính ở chỗ không có phép MixColumns.

Riêng đối với cấu trúc giải mã trong AES gồm 2 chế độ giải mã:

+ Ở cấu trúc giải mã ngược, gồm vòng khởi tạo, Nr-1 vòng lặp và vòng kết thúc. Trong đó vòng khởi tạo chỉ có phép biến đổi AddRounKey, vòng lặp gồm lần lượt 4 phép biến đổi chính: InvShiftRows, InvSubBytes, AddRounKey, InvMixColumns; vòng kết thúc khác với vòng lặp chính ở chỗ không có phép InvMixColumns.

+ Ngược lại với cấu trúc giải mã ngược là cấu trúc giải mã xuôi, việc ngược lại thể hiện ở điểm: trong cấu trúc giải mã xuôi việc sắp xếp các phép biến đổi ngược giống hệt với cấu trúc mã hóa, cụ thể bao gồm: vòng khởi tạo, Nr-1 vòng lặp và vòng kết thúc.

+ Trong đó vòng khởi là phép AddRounKey; ở vòng lặp thứ tự các phép biến đổi ngược lần lượt là: InvSubBytes, InvShiftRows, InvMixColumns, AddRounKey; vòng kết thúc giống vòng lặp nhưng được lược bỏ phép InvMixColumns



ƯU ĐIỂM VÀ NHƯỢC ĐIỂM CỦA MÃ HÓA ĐỐI XỨNG Như vậy, trên đây đã trình bày chi tiết về một số thuật toán mã hóa đối xứng phổ biến hiện nay.

Từ quá trình nghiên cứu về các cơ chế hoạt động của các thuật toán mã hóa này, có thể rút ra một số đánh giá về ưu điểm và nhược điểm của mã hóa đối xứng như sau:

- Ưu điểm:

• Đơn giản, tức là các yêu cầu về phần cứng không phức tạp;

• Thời gian tính toán nhanh (tốc độ mã hóa và giải mã nhanh);

• Có tính hiệu quả, thông thường hệ số mở rộng bản tin bằng 1 (số bit đầu ra mã hóa bằng với số bit đầu vào); Bài giảng Mật mã học cơ sở Chương 2 86

• Dễ sử dụng cho các dịch vụ nhạy cảm với độ trễ và các dịch vụ di động.

- Nhược điểm:

• Yêu cầu phải dùng kênh an toàn để truyền khóa: chi phí cao và việc thiết lập kênh an toàn là khó khăn;

• Việc tạo khóa và giữ bí mật khóa rất phức tạp;

• Số lượng khóa cần tạo ra và lưu trữ lớn;

• Khó khăn trong việc xây dựng các dịch vụ an toàn khác như các dịch vụ đảm bảo tính toàn vẹn dữ liệu, các dịch vụ xác thực và chữ ký số.

2. Thuật toán mã hóa dòng đối xứng

-THUẬT TOÁN MÃ HÓA RSA Thuật toán mã hóa RSA là một thuật toán mã hóa khóa công khai. Thuật toán RSA được xây dựng bởi các tác giả Ron Rivest, Adi Shamir và Len Adleman tại học viện MIT vào năm 1977, và ngày nay đang được sử dụng rộng rãi. Về mặt tổng quát thuật toán RSA là một phương pháp mã hóa theo khối. Trong đó thông điệp M và bản mã C là các số nguyên từ 0 đến 2i với i số bit của khối. Kích thước thường dùng của i là 1024 bit. Thuật toán RSA sử dụng hàm một chiều là vấn đề phân tích một số thành thừa số nguyên tố và bài toán Logarith rời rạc.

-Quá trình mã hóa:

Để thực hiện mã hóa và giải mã, thuật toán RSA dùng phép lũy thừa modulo của lý thuyết số. Các bước thực hiện như sau:

+ Chọn hai số nguyên tố lớn p và q và tính N = pq. C

ần chọn p và q sao cho: M < 2i-1 < N < 2i .

+Tính Φ(n) = (p - 1)(q - 1)

+ Tìm một số e sao cho: {e và Φ(n) là 2 số cùng nhau và 0 < e < Φ(n)}

+ Tìm một số d sao cho: e.d ≡ 1 mod Φ(n) (hay: d = e-1 mod Φ(n)) (d là nghịch đảo của e trong phép modulo Φ(n))

+ Hủy bỏ n, p và q. Chọn khóa công khai KU là cặp (e, N), khóa riêng KR là cặp (d, N)

+ Việc mã hóa thực hiện theo công thức:

Theo phương án 1, mã hóa: C = E(M, KU) = Me mod N

Theo phương án 2, mã hóa chứng thực: C = E(M, KR) = Md mod N

-Quá trình giải mã:

Việc giải mã thực hiện theo công thức:

+ Theo phương án 1, giải mã: M = D(C, KR) = Cd mod N (3.7)

+ Theo phương án 2, mã hóa chứng thực: M = D(C, KU) = Ce mod N (3.8) Thông điệp M có kích thước i-1 bit, bản mã C có kích thước i bit.

-Độ phức tạp trong tính toán RSA: Có hai vấn đề về độ phức tạp tính toán trong phương pháp mã hóa RSA. Đó là các phép tính sinh khóa và các phép tính mã hóa/giải mã.

- Phép tính mã hóa và giải mã: Phép tính mã hóa và giải mã dùng phép lũy thừa modulo.

Để an toàn, thường phải chọn N, e, d, M lớn.

Tuy nhiên số học modulo có một tính chất sau: ( ) ( mod ) ( mod ) mod a b a n b n n × ≡ × [ ] Có thể sử dụng tính chất này để đơn giản phép tính lũy thừa modulo thông qua một phương pháp gọi là “bình phương liên tiếp”.

Ưu điểm:

• Kích thước khóa lớn, độ an toàn cao;

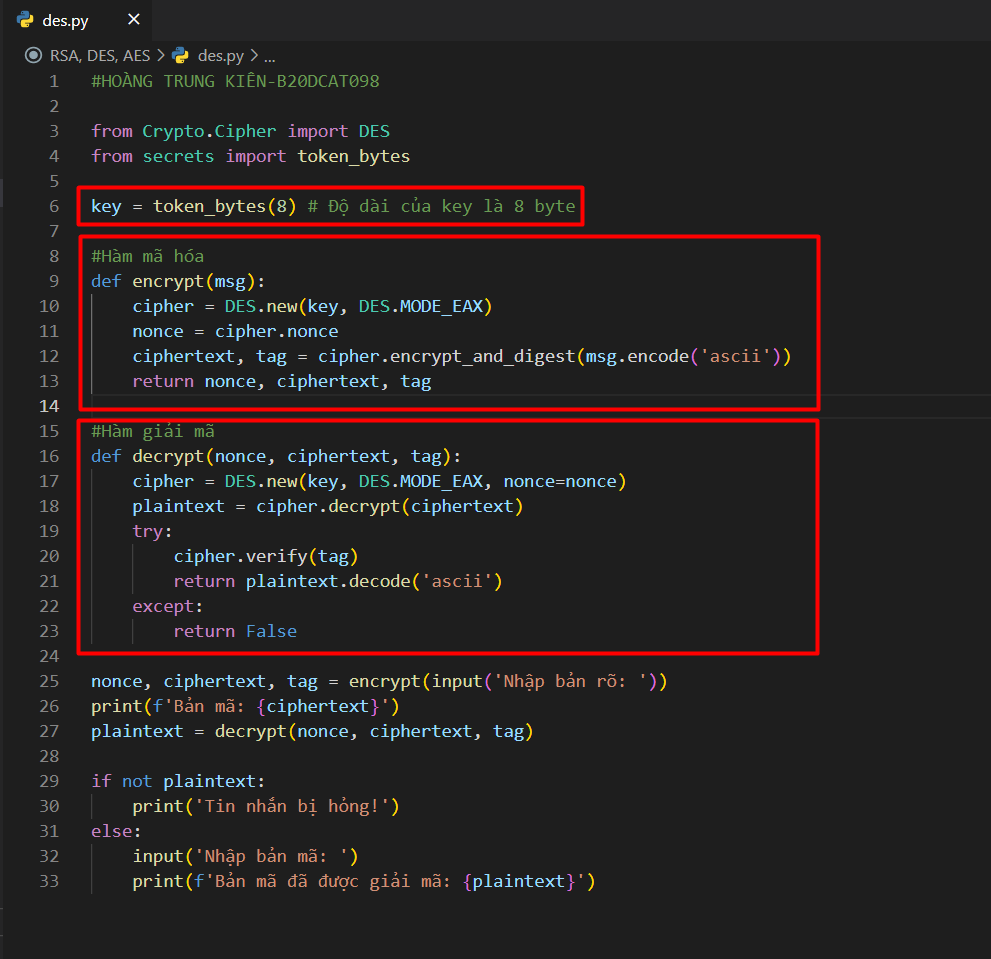
• Việc quản lý và phân phối khóa dễ dàng hơn so với mã hóa đối xứng.

Nhược điểm: • Tốc độ mã hóa, giải mã chậm hơn so với mã hóa đối xứng.

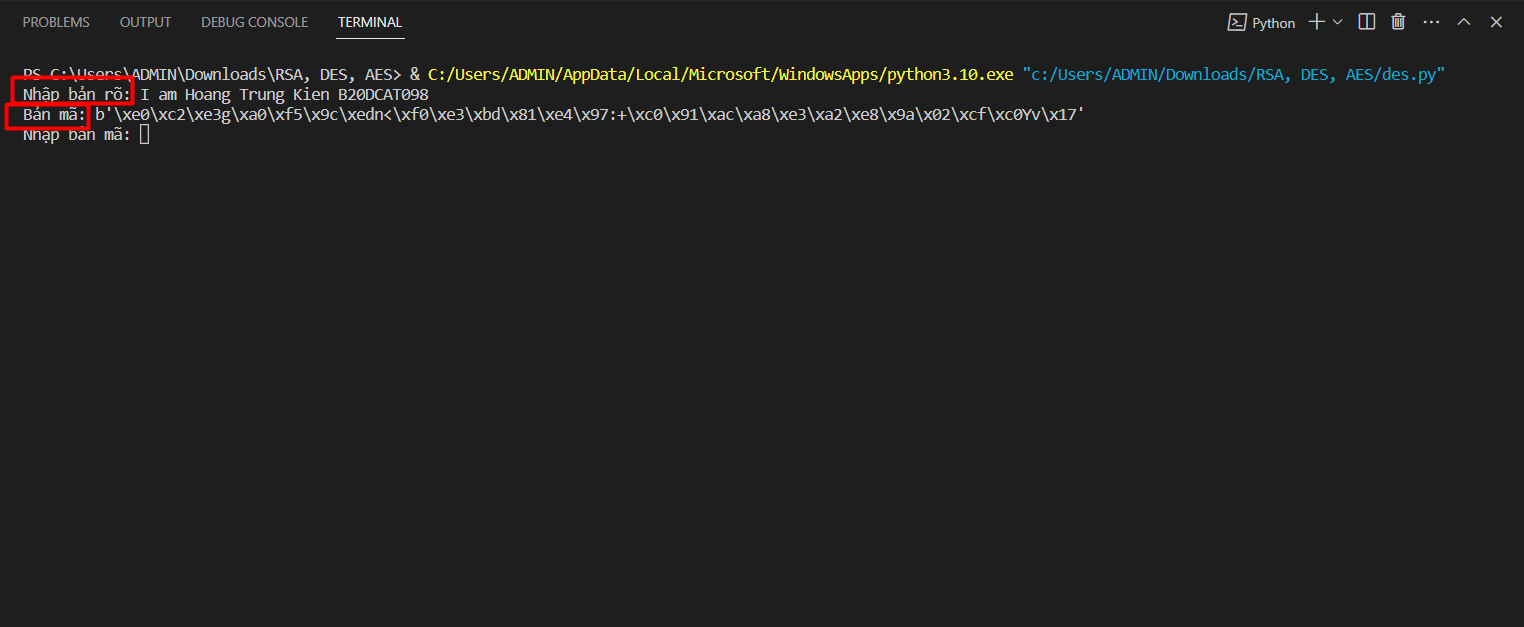
II. Thực hành

1. Thuật toán DES

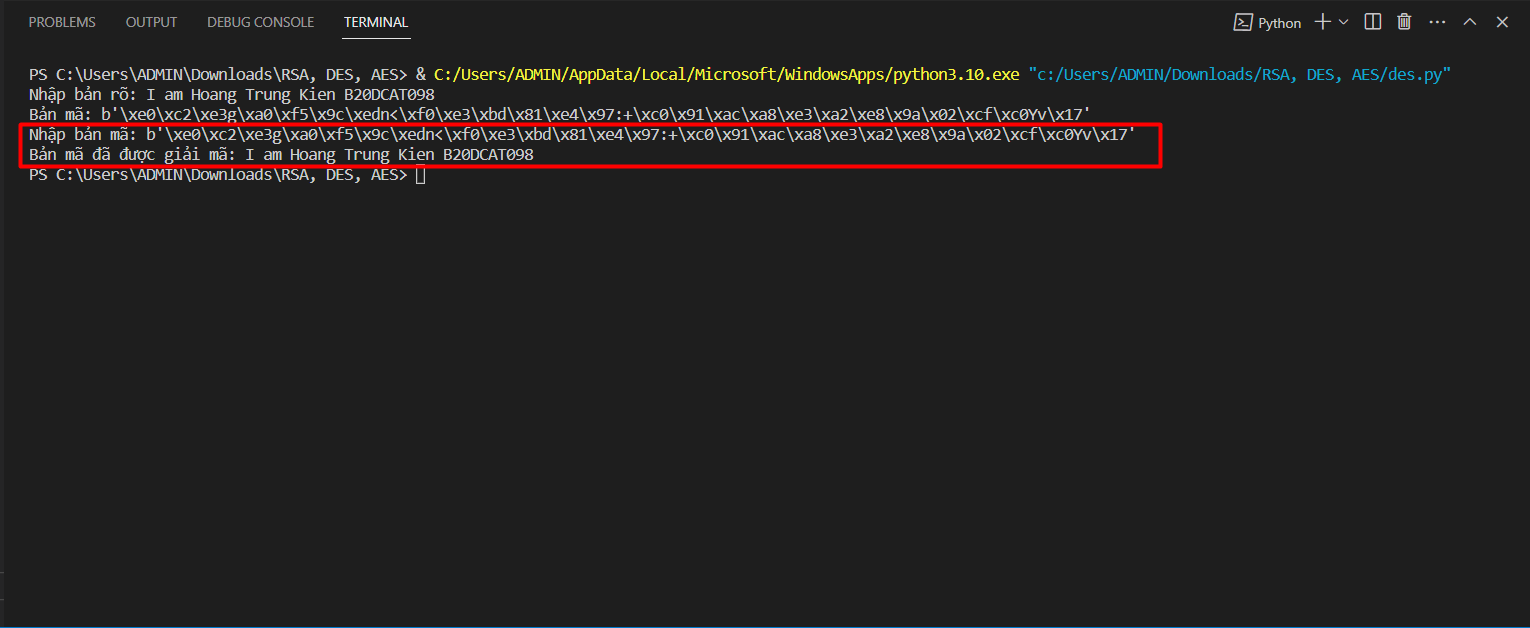
-Sử dụng thư viện Crypto được hỗ trợ trong ngôn ngữ python



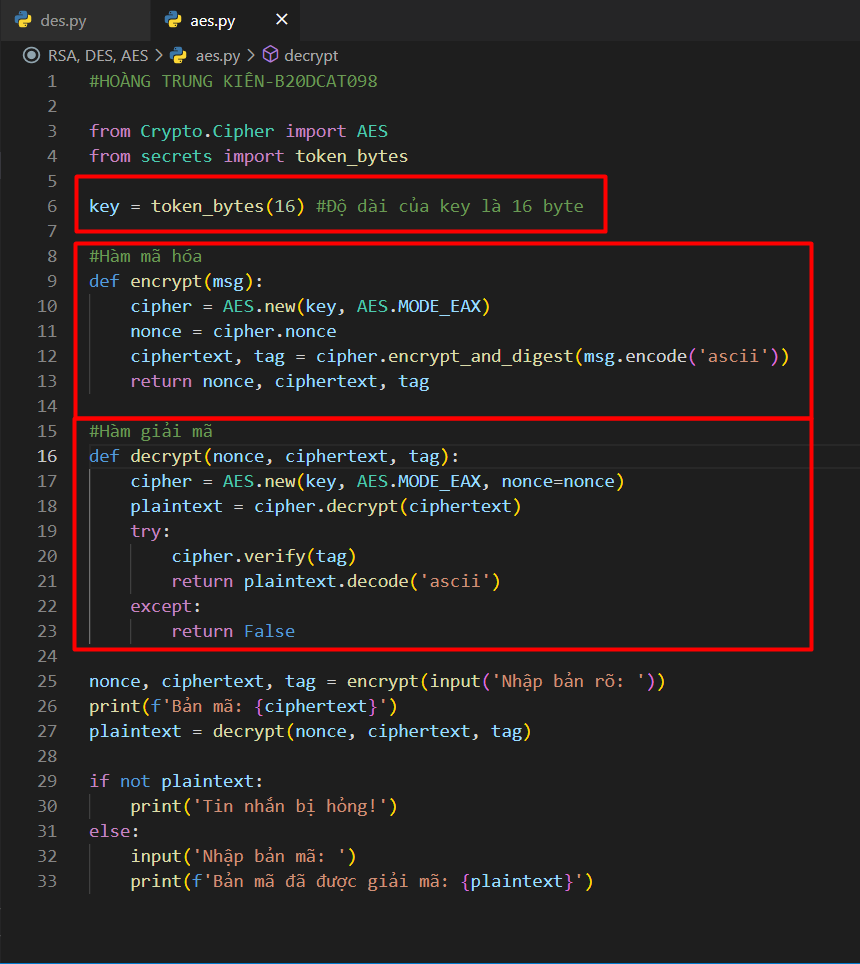
Nhập bản rõ là: I am Hoang Trung Kien B20DCAT098 và bên dưới là bản rõ đã được mã hóa



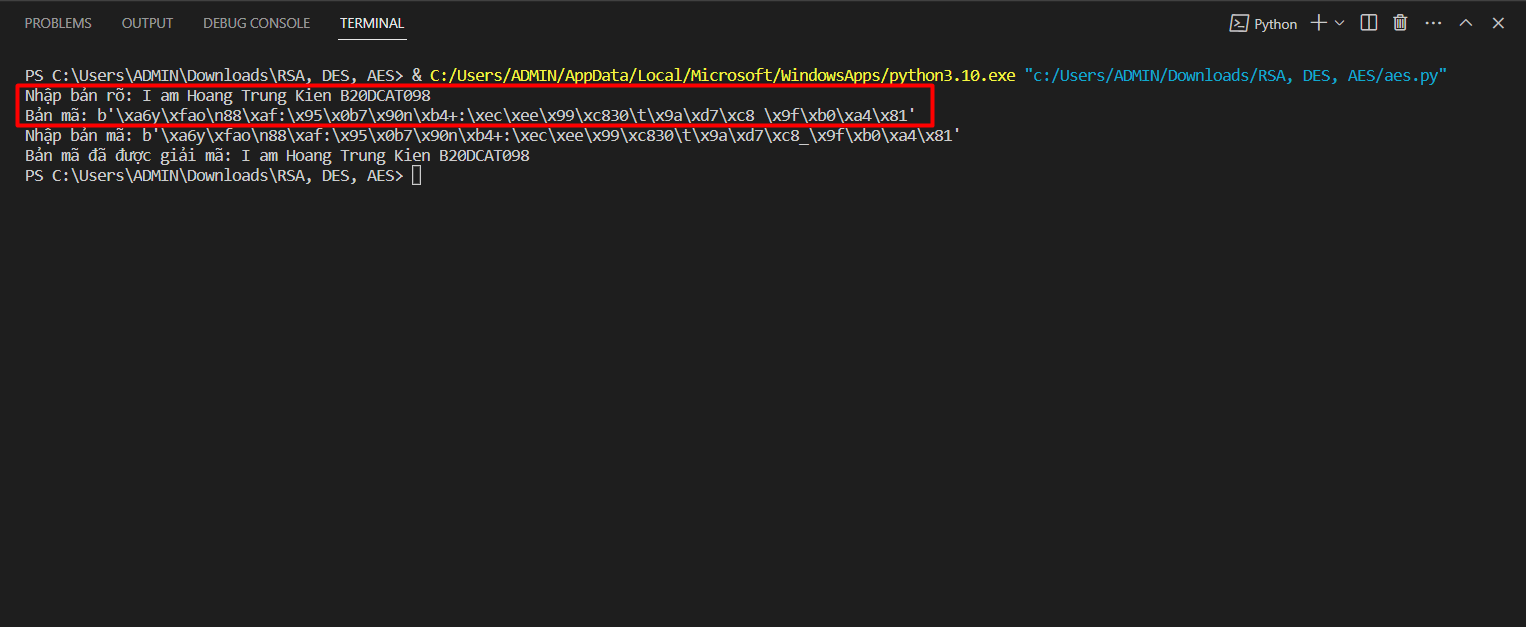
Ta nhập lại bản mã và bên dưới là bản rõ đã được giải mã từ bản mã



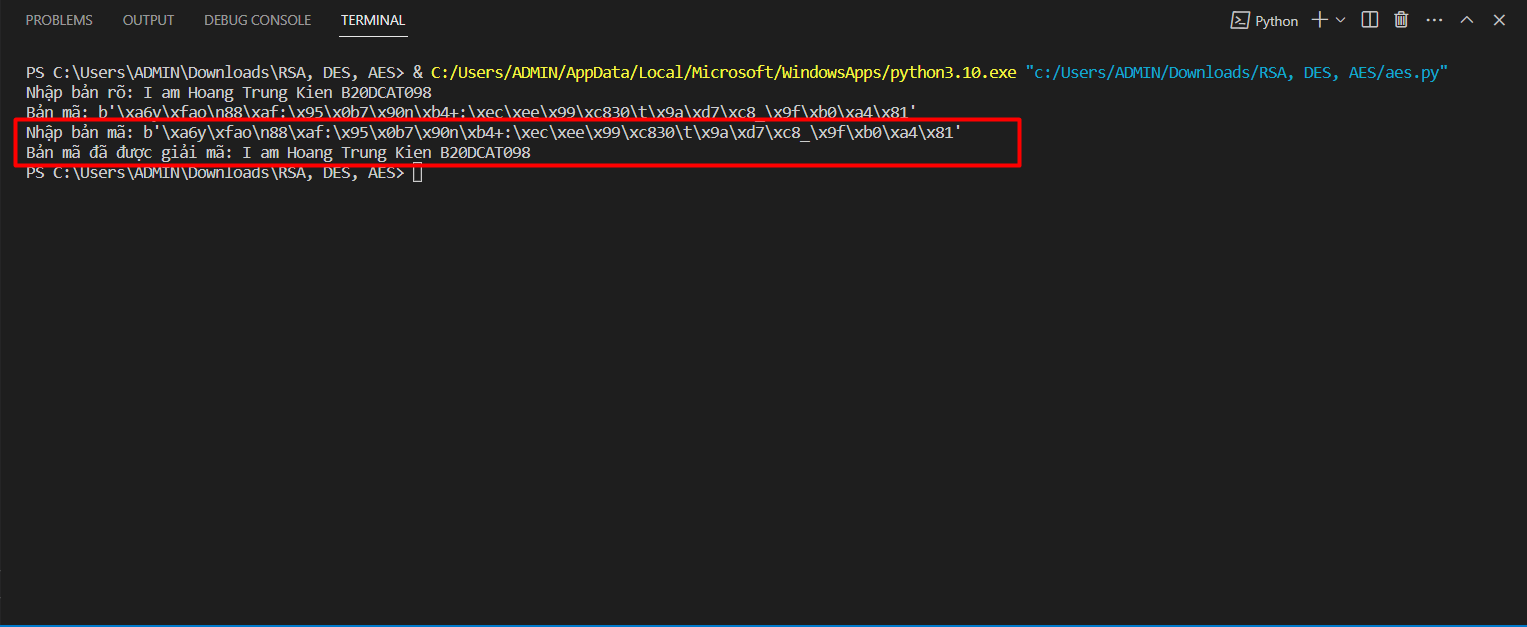
2. Thuật toán AES



Nhập bản rõ là: I am Hoang Trung Kien B20DCAT098 và bên dưới là bản rõ đã được mã hóa

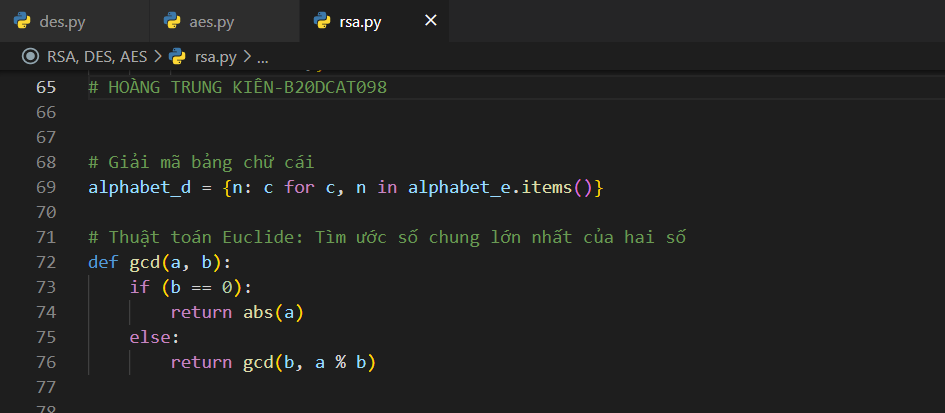


Ta nhập lại bản mã và bên dưới là bản rõ đã được giải mã từ bản mã

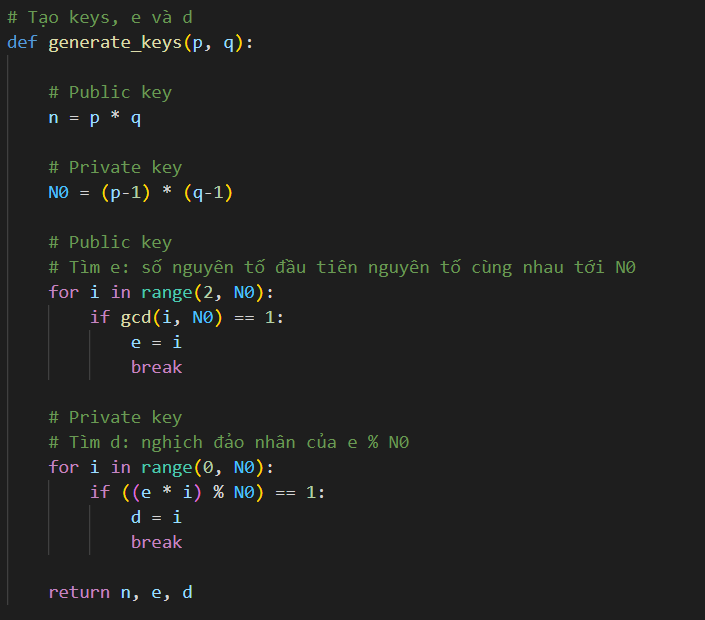


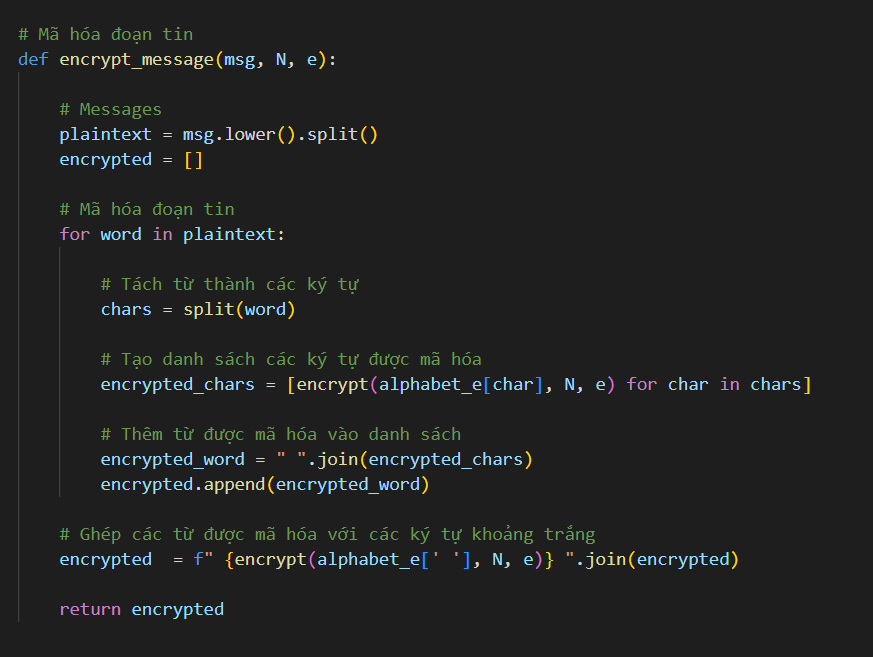
3. Thuật toán RSA

Hàm tìm ước số chung lớn nhất của 2 số

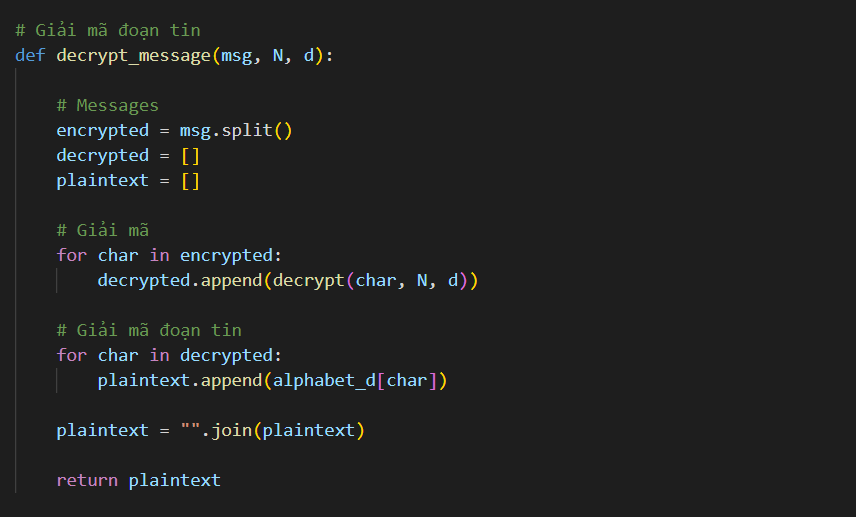


Hàm tạo khóa, cặp số nguyên tố

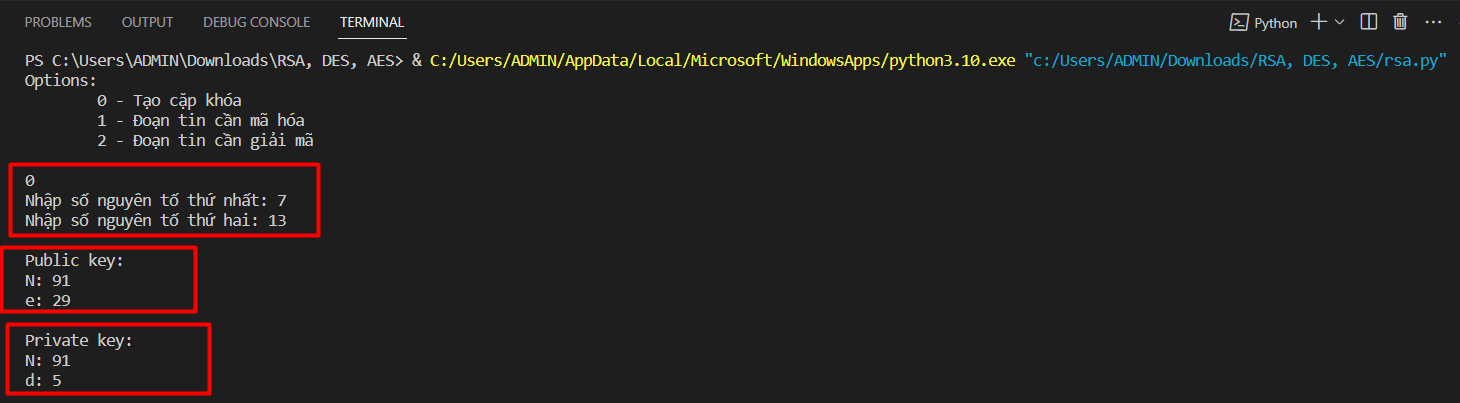


Hàm mã hóa 

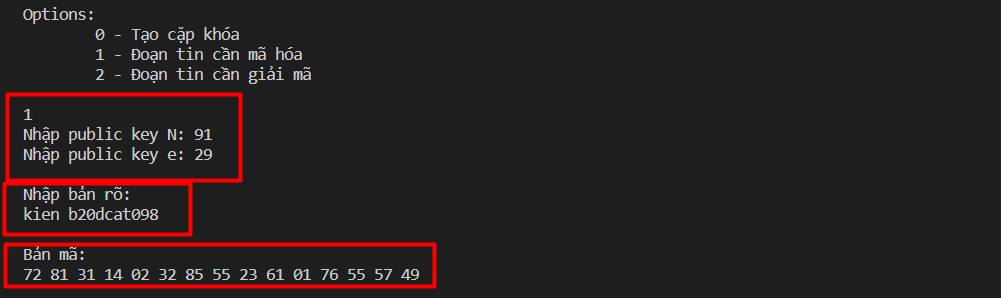
Hàm giải mã



Chọn 2 số nguyên tố và tạo cặp khóa



Nhập đoạn tin cần mã hóa: nhập public key (N, e)

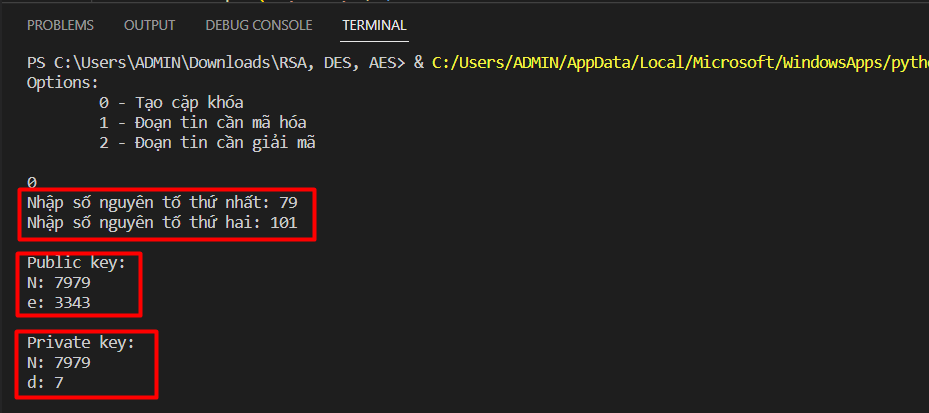


Nhập bản mã: nhập private key (N, d) => ta thu lại được bản rõ

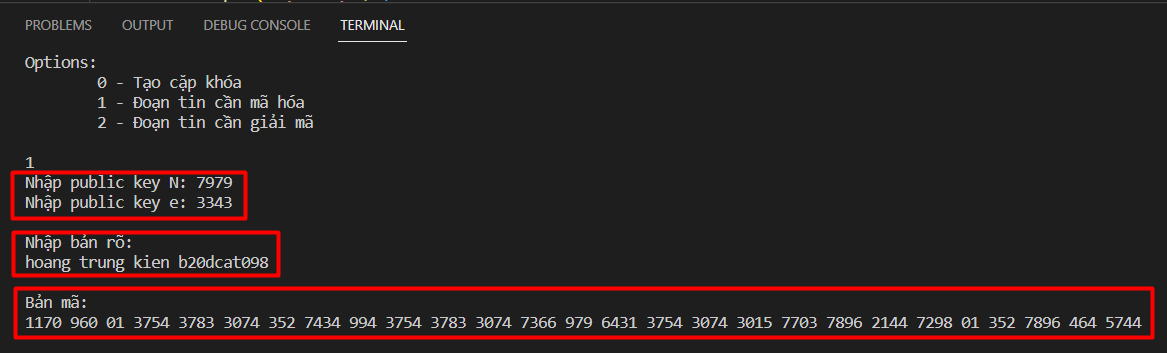


-Trường hợp 2:

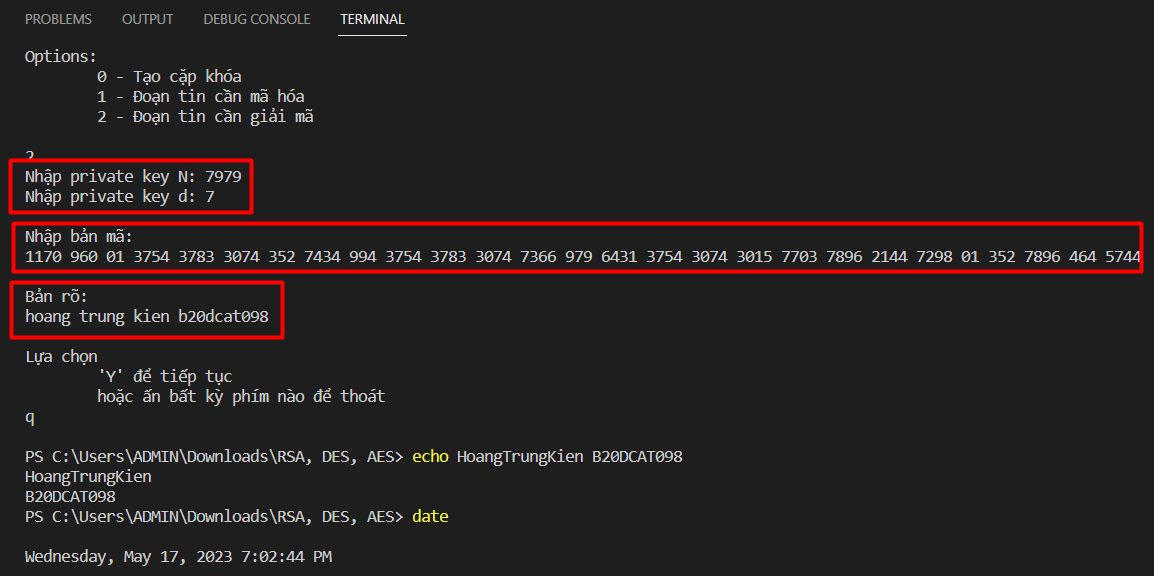
Chọn 2 số nguyên tố và tạo cặp khóa



Ta nhập public key và bản rõ => thu được bản mã



Ta nhập private key và bản mã => thu được bản rõ



**-Trường hợp 3:**

