

NT219 - BTTL - LAB04

Nguyễn Hoàng Quý - 24521494

Ngày 17 tháng 11 năm 2025

Mục lục

1	Guide1_asn1_syntax_and_key_format	3
1.1	Khái niệm ASN.1	3
1.2	Chức năng của ASN.1	3
1.3	Các quy tắc mã hóa	3
1.4	Ứng dụng trong mật mã học	3
1.5	Ví dụ thực tế	4
1.6	Công cụ hỗ trợ	4
1.7	OPENSSL	4
2	Guide2_rsa_key_generation	5
2.1	Task2.1	5
2.2	Task2.2	5
3	Guide3_rsa_encryption	5
3.1	Task3.1	6
3.2	Task3.2	6
4	Guide4_rsa_decryption	7
4.1	Task4.1	7
4.2	Task4.2	7
5	Guide5_full_code_implementation	7
6	Guide6_library_export	8
6.1	Windows	8
6.2	Linux	10

7	Guide7_cross_language_integration	10
7.1	Windows	11
7.1.1	Python	11
7.1.2	Csharp	11
7.1.3	Java	12
7.2	Linux	12
7.2.1	Python	12

1 Guide1_asn1_syntax_and_key_format

1.1 Khái niệm ASN.1

ASN.1 (Abstract Syntax Notation One) là ngôn ngữ mô tả cấu trúc dữ liệu trừu tượng, dùng để định nghĩa cách biểu diễn và trao đổi dữ liệu giữa các hệ thống khác nhau — đặc biệt trong mật mã học, chứng chỉ số, và các giao thức bảo mật như SSL/TLS, X.509, hay PKCS.

1.2 Chức năng của ASN.1

ASN.1 cho phép biểu diễn các đối tượng phức tạp (như khóa, chứng chỉ, hoặc thông điệp) theo một chuẩn chung, giúp đảm bảo khả năng tương thích giữa các phần mềm hoặc thiết bị của những nhà cung cấp khác nhau.

1.3 Các quy tắc mã hóa

ASN.1 đi kèm với nhiều quy tắc mã hóa, phổ biến nhất là:

- **BER (Basic Encoding Rules):** Dạng cơ bản, linh hoạt, có thể biểu diễn cùng một giá trị theo nhiều cách khác nhau.
- **DER (Distinguished Encoding Rules):** Phiên bản ràng buộc hơn của BER, mỗi giá trị chỉ có một biểu diễn duy nhất — thường được sử dụng trong chữ ký số và chứng chỉ X.509.
- **CER (Canonical Encoding Rules):** Tương tự DER, nhưng được dùng cho dữ liệu có độ dài lớn.

1.4 Ứng dụng trong mật mã học

Trong mật mã học, ASN.1 được sử dụng để định nghĩa:

- Cấu trúc khóa công khai và khóa bí mật (Public/Private Key Structure)
- Định dạng chứng chỉ X.509
- Thông tin gói dữ liệu trong chuẩn PKCS#7, PKCS#8, PKCS#12

1.5 Ví dụ thực tế

Ví dụ, trong chứng chỉ X.509, phần khóa công khai (Public Key) của người dùng được biểu diễn bằng ASN.1, sau đó mã hóa theo DER trước khi lưu trong file có phần mở rộng .cer hoặc .pem. Một đoạn ASN.1 mẫu có thể trông như sau:

```
SubjectPublicKeyInfo ::= SEQUENCE {  
    algorithm AlgorithmIdentifier,  
    subjectPublicKey BIT STRING  
}
```

1.6 Công cụ hỗ trợ

Các thư viện như OpenSSL hoặc Crypto++ hỗ trợ phân tích, sinh, và xử lý dữ liệu ASN.1. Chúng cho phép:

- Chuyển đổi giữa định dạng PEM và DER
- Trích xuất hoặc kiểm tra thành phần trong chứng chỉ
- Sinh khóa và chứng chỉ theo chuẩn X.509

1.7 OPENSSL

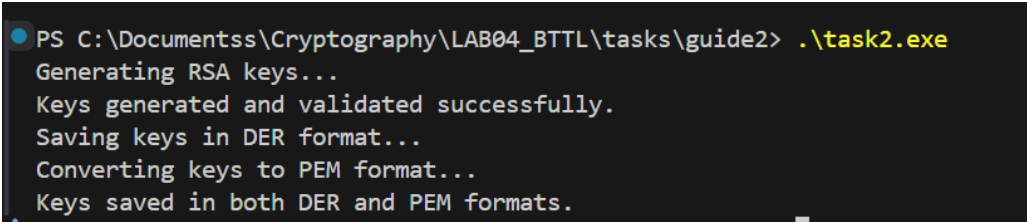
```
PS C:\Users\Hoang Quy> echo | openssl s_client -showcerts -servername www.vietcombank.com.vn -connect www.vietcombank.com.vn:443 2> cert | openssl x509 -out  
vcb.cer -text  
  
cmdlet Write-Output at command pipeline position 1  
Supply values for the following parameters:  
InputObject: 1  
PS C:\Users\Hoang Quy> cat .\vcb.cer  
Certificate:  
Data:  
    Version: 3 (0x2)  
    Serial Number:  
        3a:c8:f0:9e:01:34:b6:bb:01:96:68:ef  
    Signature Algorithm: sha256WithRSAEncryption  
    Issuer: C=BE, O=GlobalSign nv-sa, CN=GlobalSign RSA OV SSL CA 2018  
    Validity  
        Not Before: Oct 19 05:27:05 2025 GMT  
        Not After : Oct 24 02:46:32 2026 GMT  
    Subject: C=VN, ST=Ha Noi, L=Ha Noi, O=JOINT STOCK COMMERCIAL BANK FOR FOREIGN TRADE OF VIETNAM, CN=*.vietcombank.com.vn  
    Subject Public Key Info:  
        Public Key Algorithm: rsaEncryption  
        Public-Key: (2048 bit)  
        Modulus:  
            00:94:c1:9d:6d:69:e6:3d:e9:bb:d7:c3:37:08:ea:  
            27:07:ab:d2:bb:0e:f4:05:32:15:c8:4b:f0:52:e5:  
            63:98:e6:b9:e2:84:49:28:80:a6:60:d0:59:19:7f:  
            56:8e:c9:70:45:8f:f9:db:9d:db:d1:76:90:29:3c:  
            e6:01:f9:07:5a:17:10:bb:5e:26:a6:be:de:8b:89:  
            14:63:05:bd:00:c6:4c:07:e8:e8:b9:6f:55:d2:87:  
            8c:5f:c5:d0:25:ea:6c:92:a6:8e:a5:3e:51:f6:32:  
            10:21:b1:74:c7:38:a7:08:ad:e5:a0:00:50:59:86:  
            77:52:c9:0d:c7:bf:53:d3:7c:ad:62:77:db:c2:47:  
            9e:a0:20:e7:63:22:d1:4a:4a:f9:f3:80:bd:8f:c2:  
            c2:e0:11:70:30:aa:87:ae:97:4b:34:12:33:d1:c0:  
            a5:76:0c:59:81:03:06:0a:73:86:82:bb:96:5a:17:  
            6c:b2:03:79:51:b5:51:05:41:b8:84:8b:b2:f6:00:  
            8a:01:ca:92:e2:be:ae:7e:fc:86:33:5e:42:f2:77:  
            fe:b0:1a:30:9d:c2:ce:52:4e:d5:d8:98:eb:f6:93:  
            f4:20:39:d5:b0:e6:fc:ec:fc:98:21:7a:c2:f6:c5:  
            16:19:45:f3:aa:e5:d3:95:11:9c:1f:07:68:1f:a1:  
            3d:53  
        Exponent: 65537 (0x10001)  
    X509v3 extensions:  
        X509v3 Key Usage: critical
```

Hình 1: Kết quả chạy thử OPENSSL

2 Guide2_rsa_key_generation

2.1 Task2.1

Target: Run all demo codes, include full and part codes

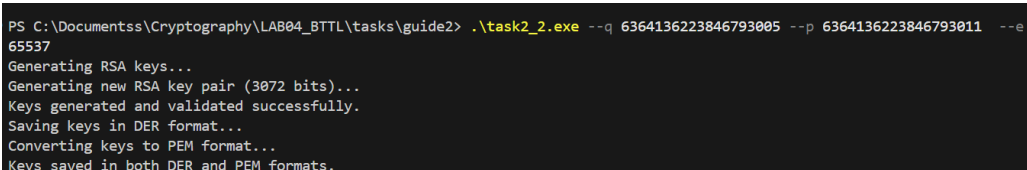


```
PS C:\Documentss\Cryptography\LAB04_BTTL\tasks\guide2> .\task2.exe
Generating RSA keys...
Keys generated and validated successfully.
Saving keys in DER format...
Converting keys to PEM format...
Keys saved in both DER and PEM formats.
```

Hình 2: Kết quả chạy full code task2

2.2 Task2.2

Target: Task 2.2 Revise code to set any public key (p, q, n, e, d, ...)



```
PS C:\Documentss\Cryptography\LAB04_BTTL\tasks\guide2> .\task2_2.exe --q 6364136223846793005 --p 6364136223846793011 --e 65537
Generating RSA keys...
Generating new RSA key pair (3072 bits)...
Keys generated and validated successfully.
Saving keys in DER format...
Converting keys to PEM format...
Keys saved in both DER and PEM formats.
```

Hình 3: Enter Caption

3 Guide3_rsa_encryption

RSA không thể mã hóa dữ liệu có độ dài vượt quá độ dài khóa của chính nó. Ví dụ, với khóa RSA 3072-bit, kích thước dữ liệu tối đa có thể mã hóa chỉ vào khoảng 384 byte ($3072 / 8$), chưa tính phần chiếm dụng cho padding. Đối với các thông điệp có kích thước lớn, RSA thường được sử dụng để mã hóa một khóa đối xứng ngẫu nhiên, sau đó khóa này được dùng để mã hóa toàn bộ nội dung thông điệp bằng thuật toán đối xứng như AES. Cách tiếp cận này được gọi là mã hóa lai (hybrid encryption), trong đó RSA đảm nhiệm việc bảo mật khóa đối xứng nhỏ, còn AES đảm bảo hiệu suất và khả năng xử lý dữ liệu lớn. Phương pháp này kết hợp tính bảo mật của mã hóa bất

đối xứng với tốc độ và hiệu quả của mã hóa đối xứng, và được sử dụng phổ biến trong các hệ thống mật mã hiện đại.

3.1 Task3.1

Target: Encryption use cin to get message from screen.

```
PS C:\Documentss\Cryptography\LAB04_BTTL\tasks\guide3> .\task3_1.exe
Loaded public key from PEM file.
Insert message to encrypt: Hello World!
Hybrid Encrypted message (Base64): gAEAAEUdX1amw3ktF8AptK//7WbTLZ6Zd2NuVg2/napxXK6J67utNCocnyXsD8uK4347Iyv
a5UkANti2IiB2vmuXJOj4aEs2TirP1tKRnK4weKUSOnofX58e1EIWT6J22e9WkXbc+K7CQ5r
WlhZLwiVnCPgIQ4RykyQG3UiGj9tUhPiD3q31xxpzbRw/EhSp8X1ILCiNr0hJBXoxP1cy
yK2yXhJ/gBaRXmfiNms+TidQBb9W6crirSRG7v0heP+SC4sbnf1e/cOVzf3yC31I1TA3b01Q
H/1J1TptN6SycVvhKnr61C7SjRix+aDFD52Q0b0ErchHDVddZuNAJJVQng43cyndIWF1YogT2j
wxURnGI53LQ5MiWzqNx7uc1EQavR9Ro3grIEXjUAn0bjp3w45zks/jiPQAaBgcRhwVh3gzpB
x1BFS762n0WVXkKcypw0ZjYis/U+yktpAKciY0J/UI7XR/PukGqKBAGxa3nx3gQ0ZbbEPJ
ejJ0T2oHPAti6FuJteckU0zWuKEJwB+EcPRDcSZOUvfQpCxSXC6Zd2IK

Direct RSA Encrypted message (Base64): hfo0PW2SRsSH0+LihRwjDkklG34guzSVdKwd1LrkMqDRfN9NR4e6jhvKwFrDE2CEG51djm8V
pVfKHylTojh81mSW5pt/n54024+DOE9o711Kysj/PRxI3RnG1XW3Mc5sR7Zh8YlbydAId6de
oBaZSp0iBe9zNrJLb8jP8Em8mqn2hDn8/r1PxRz3ahPcxUvPkEDVS6051JdQvoebU7HgOZLV
O8emHuzX55N2eFFBEndsvIB1RB12+i3KzkxhqvtuQx2GunRKbgm0S5AnY70gOjLHtywmODn1
QuMKv9vQ30xp83z9xgSseksPFwisj7gXa7Q125E89XNV1jsDlmHSy8Wmzb8vI0dwsFukAt18
CTaB8S2PeW8xHRKCa4mZkK1vbQRxD0toebIbXRFQKoV8YAbCj1CvCmaK+JpU85pVNaE2b
ruDPRfhwQzWvKt1beTNUuSKfnczvxwmbu8iQAQubhH8RGoemhuyK7QTf52dbGvYF8uRwKp6M5
```

Hình 4: Kết quả mã hóa với message nhập bằng cin

3.2 Task3.2

Target: Encryption (OAEP_SHA) use File/Source/Sink to get message (bin) from file and save output to file (bin/base64)

```
PS C:\Documentss\Cryptography\LAB04_BTTL\tasks\guide3> .\task3_2.exe --file .\haha.txt
Loaded public key from PEM file.
Original Message: Hello World!
Hybrid Encrypted (Base64): gAEAAD3g0vCQQHVaqtZtZbuNSpeBqctR9YEBz4gPLQ45G0hZmazUotPpyHvfUtcU7eByX0tQX
9yZIo6ykAKViDRNYgW8Z1b6R9doG5/1i/96QHIMdg3nTPnJTxKpU9P4/mrQcd2zgdldwamI
ES3zm7S+SS89X80WdfOK9/DqA0nj90suL70zefs27Dmk0seSbs056QCqWbZ8beoLWx5TpYtB
GsRE8TgJot6Wrf/I1oLVdVuNrbHNwSdPemGzKwD4R3g9fA1QS10jp+whlwkvygQEZuH1b68T
Pm8pOJWzFUPzHtsVsASHIZ0Dg6h7g/75k0ufzim/b1oEf6IOK9rPY37wW/yBpZFDLMHa03z1
WgJkZeSstxBJqRcUG1zQo0ugzXSfwKHBpM0mvVBZUMH5UfgiXqnpE+MHwHCcn0kHwsuEcO
wYY1vFwG572AX68mHwdj1LSrApo4/6ynMiKe77ItRuSBGERPCeZ0bgdwwq80z0Dug+gpABJX
fCroRtKCPmr5GjfLvH6kfH+yH1lz0VEKbnkoKHB7WYPaPsx18mV5YwTw

Hybrid encrypted data saved to hybrid_encrypted.bin
```

Hình 5: Kết quả thu được sau khi lấy plaintext từ file

4 Guide4_rsa_decryption

4.1 Task4.1

Target: Decryption use cin to get message from screen

```
PS C:\Documentss\Cryptography\LAB04_BTTL\tasks\guide4> .\task4_1.exe
Insert RSA Base64 ciphertext (short, <= key size): "L/GQFWKT1odHa3lw3vSwiM8vzbsVCpVZH5qsv6OMv5zyrqLpNpMXUWwC7bJewJfZeZZbeh
KgSq1w3vAEv05AwH+wNOpP306cIp6bekvK1jWm0/N+5MySI6+su6nMMu5/fx8sPb8Z6WazDDpNJx/UTO4jM9cS/pSV3zj/Y4x8Z4MINTkCZu9/mo8FaC0yxCRk
IeAbLdy+WNi2yzKPsFjJwM0Mk6ZLd1oQ+c8o5qu+sXfd28TL1U9iSe17wui0C2zqP+q3cXBS2+KCb1xdIppqApP+PyYwU8BLpDq06ukRJeAq+8t90L36ag8sQ
BrpwkLur21Rf62GmzetVY08rbwCKFQ9xh4xzDz4wo1H9eKbQMbmJ9ZvwiNgzcXbdPUDR1EwH9eXe00Ra861mf1LSgrJFhu8pVhmy5EbNrTTOEI+4J+5Zt7VZVx
pGUUua/M/+nyAbzA6DsXGEDZLg9W340rOqo/Cm0Y18KkcM7HDFExYE89hMoVvufT6dSgkoUDZFNf"
RSA decrypted: Hello World!
```

Hình 6: Mã hóa ciphertext với cin input

4.2 Task4.2

Target: Decryption (OAEP_SHA) use File/Source/Sink to get message (bin) from file and save output to file (bin)

```
PS C:\Documentss\Cryptography\LAB04_BTTL\tasks\guide4> .\task4_2.exe
Loaded private key from PEM file.

RSA decryption result: Hello World!

Hybrid decryption result: Hello World!
Hybrid decrypted data saved to hybrid_decrypted_output.bin
RSA decrypted data saved to rsa_decrypted_output.bin
```

Hình 7: Kết quả mã hóa được lưu vào file bin

5 Guide5_full_code_implementation

Target: create a complete RSA encryption/decryption application using the Crypto++ library with command-line arguments. This comprehensive implementation combines all the concepts covered in the previous guides into a single, versatile command-line tool that can generate keys, encrypt, and decrypt data.

```

PS C:\Documentss\Cryptography\LAB04_BTTL\tasks\guide5> ./task5.exe --generate-keys --key-size 3072
Generating 3072-bit RSA keys...
Keys generated and validated successfully.
Saving keys in DER format...
Converting keys to PEM format...
Keys saved in both DER and PEM formats:
  Private key: private_key.pem and private_key.der
  Public key: public_key.pem and public_key.der
PS C:\Documentss\Cryptography\LAB04_BTTL\tasks\guide5> ./task5.exe --encrypt --public-key public_key.pem --input message.txt --output encrypted.bin
Loading public key from public_key.pem...
Reading input from message.txt...
Performing RSA encryption (oaep)...
Ciphertext written to encrypted.bin
PS C:\Documentss\Cryptography\LAB04_BTTL\tasks\guide5> ./task5.exe --decrypt --private-key private_key.pem --input encrypted.bin --output decrypted_message.txt
Reading input from encrypted.bin...
Performing RSA decryption (oaep)...
Plaintext written to decrypted_message.txt
PS C:\Documentss\Cryptography\LAB04_BTTL\tasks\guide5> type decrypted_message.txt
Day la tin nhan bi mat cua toi.

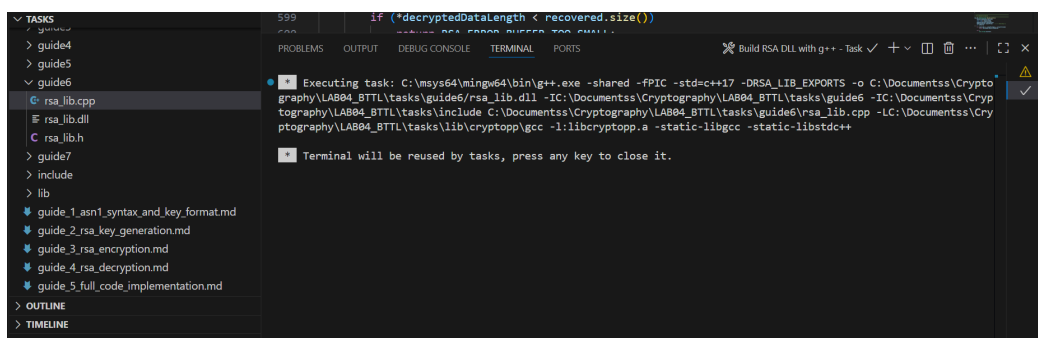
```

Hình 8: Kết quả chạy test CLI task5

6 Guide6_library_export

Target: export the RSA functionality developed in previous guides to a shared library (DLL on Windows or SO on Linux/macOS). Creating a shared library allows the RSA functionality to be used by multiple applications and programming languages, promoting code reuse and modularity.

6.1 Windows



Hình 9: Kết quả build file DLL


```

C:\Documentss\Cryptography\LAB04_BTTL\tasks\guide6>dumpbin /dependents rsa_lib.dll
Microsoft (R) COFF/PE Dumper Version 14.44.35214.0
Copyright (C) Microsoft Corporation. All rights reserved.

Dump of file rsa_lib.dll

File Type: DLL

Image has the following dependencies:

  ADVAPI32.dll
  KERNEL32.dll
  msvcrt.dll
  libwinpthread-1.dll

Summary
  3000 .bss
  3000 .data
  3B000 .debug_abbrev
  17000 .debug_aranges
  51000 .debug_frame
  EA3000 .debug_info
  227000 .debug_line
  11000 .debug_line_str
  3CC000 .debug_loclists
  F2000 .debug_rnglists
  25000 .debug_str
  1000 .edata
  2000 .idata
  1A000 .pdata
  65000 .rdata
  B000 .reloc
  226000 .text
  1000 .tls
  2E000 .xdata

```

Hình 10: Dumpin dependents

```
Dump of file rsa_lib.dll
File Type: DLL

Section contains the following exports for rsa_lib.dll

00000000 characteristics
6915045D time date stamp Thu Nov 13 05:04:13 2025
0.00 version
1 ordinal base
11 number of functions
11 number of names

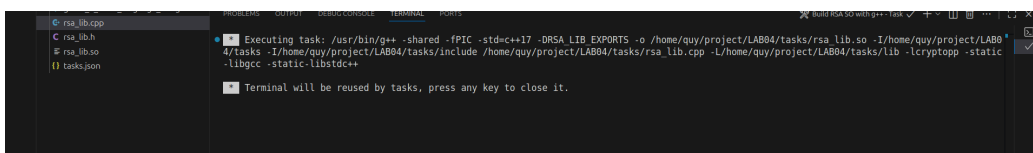
ordinal hint RVA      name
1 0 00004709 RSA_Decrypt
2 1 00004AD3 RSA_DecryptFile
3 2 0000456E RSA_Encrypt
4 3 000048A4 RSA_EncryptFile
5 4 00002BE6 RSA_FreePrivateKey
6 5 00002BAA RSA_FreePublicKey
7 6 000016AF RSA_GenerateKeyPair
8 7 00004D20 RSA_GetErrorMessage
9 8 00004E61 RSA_GetMaxPlaintextLength
10 9 00002560 RSA_LoadPrivateKey
11 A 00001F65 RSA_LoadPublicKey

Summary

3000 .bss
3000 .data
3B000 .debug_abbrev
17000 .debug_aranges
51000 .debug_frame
EA3000 .debug_info
227000 .debug_line
11000 .debug_line_str
3CC000 .debug_loclists
F2000 .debug_rnglists
25000 .debug_str
1000 .edata
2000 .idata
1A000 .pdata
65000 .rdata
```

Hình 11: Dumpin Export

6.2 Linux



Hình 12: Kết quả build file SO

7 Guide7_cross_language_integration

Target: use the RSA shared library created in Guide 6 from Python, Cs, and Java. Cross-language integration allows you to leverage the performance and security of the C++ implementation while working in your preferred programming language.

7.1 Windows

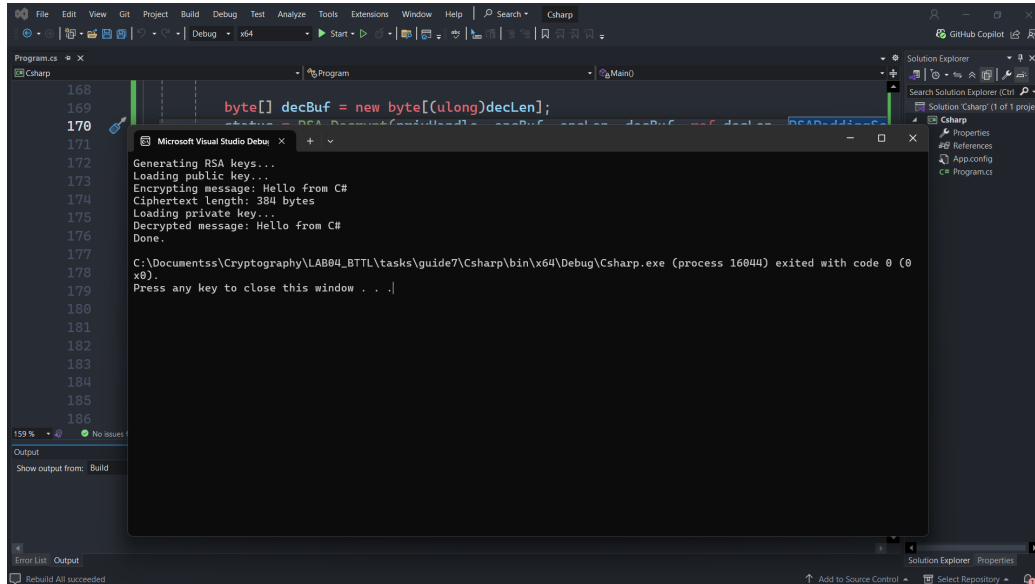
7.1.1 Python

```
PS C:\Documentss\Cryptography\LAB04_BTTL\tasks\guide7\python> python .\rsa_example.py
Generating RSA keys...
Loading public key...
Encrypting message: Hello from Python!
Loading private key...
Decrypting message...
Decrypted message: Hello from Python!

File encryption example:
Encrypting file...
File encrypted successfully.
Decrypting file...
File decrypted successfully.
Decrypted file content: This is a test file for RSA encryption.
Done.
```

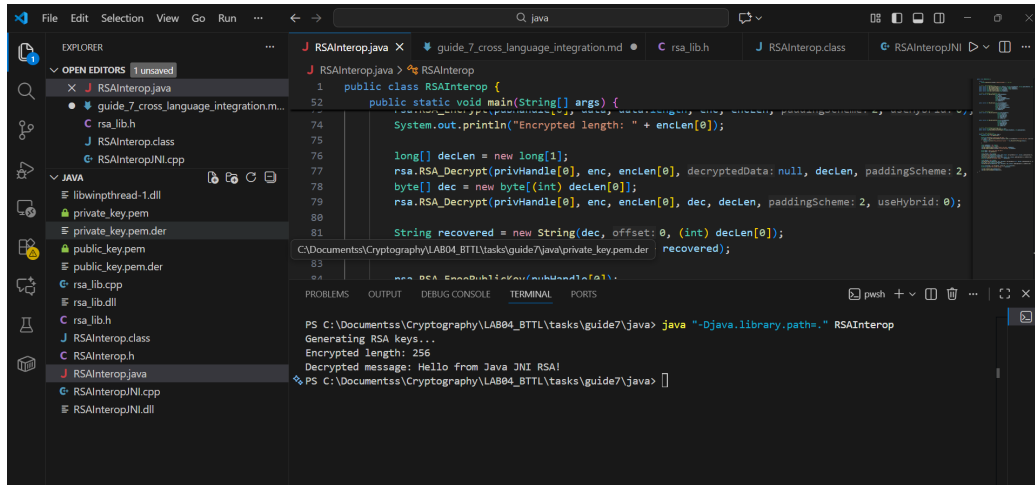
Hình 13: Kết quả chạy rsa_example.py

7.1.2 Csharp



Hình 14: Kết quả chạy RSA trên CSharp

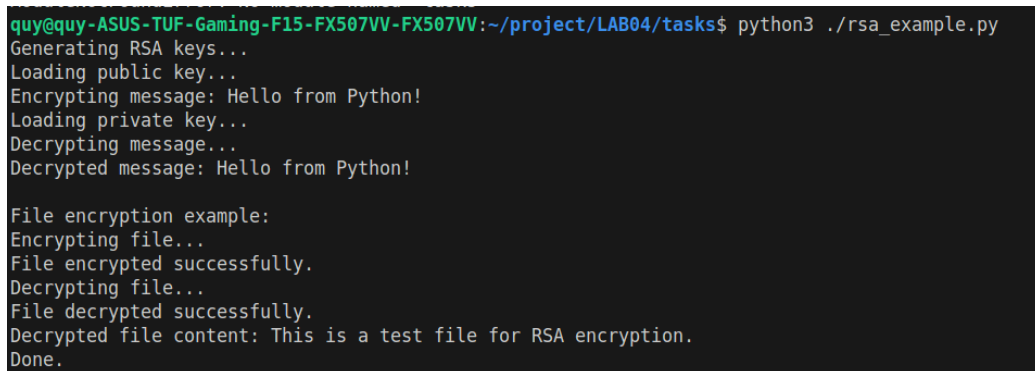
7.1.3 Java



Hình 15: Kết quả chạy RSA trên Java

7.2 Linux

7.2.1 Python



Hình 16: Kết quả chạy rsa_example.py trên Linux