NT219 - BTTL - LAB04

Nguyễn Hoàng Quý - 24521494

Ngày 17 tháng 11 năm 2025

Mục lục

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:

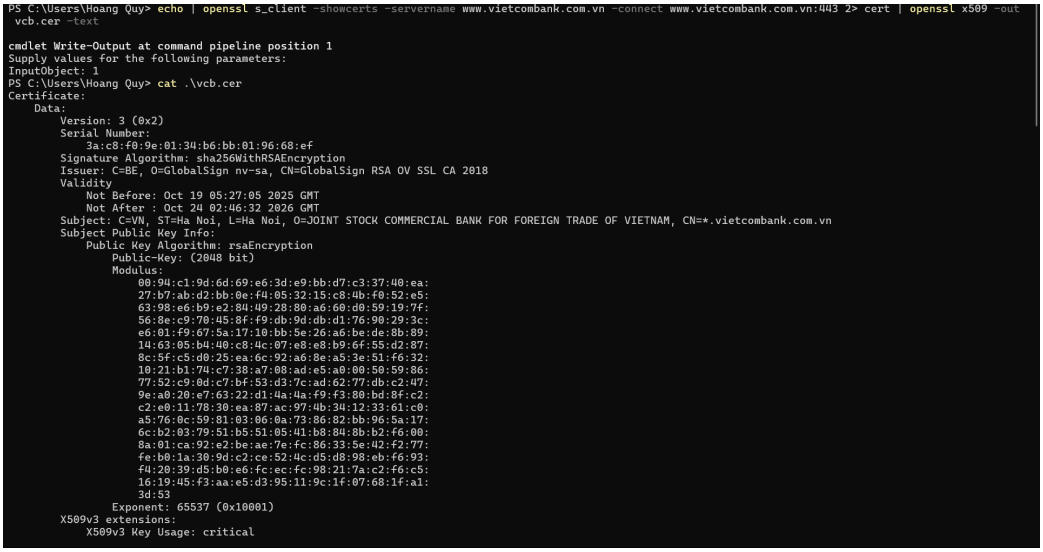
```
SubjectPublicKeyInfo ::= SEQUENCE {
    algorithm AlgorithmIdentifier,
    subjectPublicKey BIT STRING
}
```

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

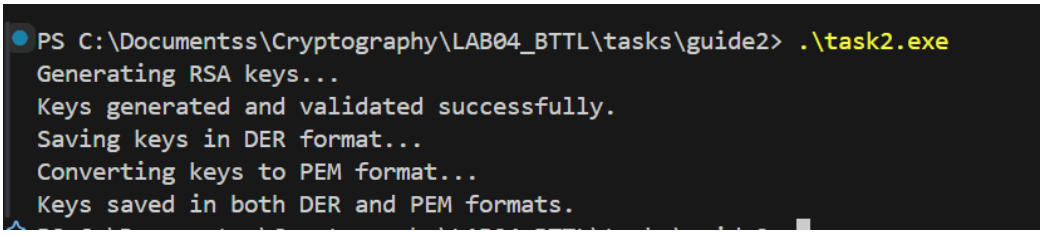


Hình 1: Kết quả chạy thử OPENSSL

2 Guide2_rsa_key_generation

2.1 Task2.1

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



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, ...)

Hình 3: Enter Caption

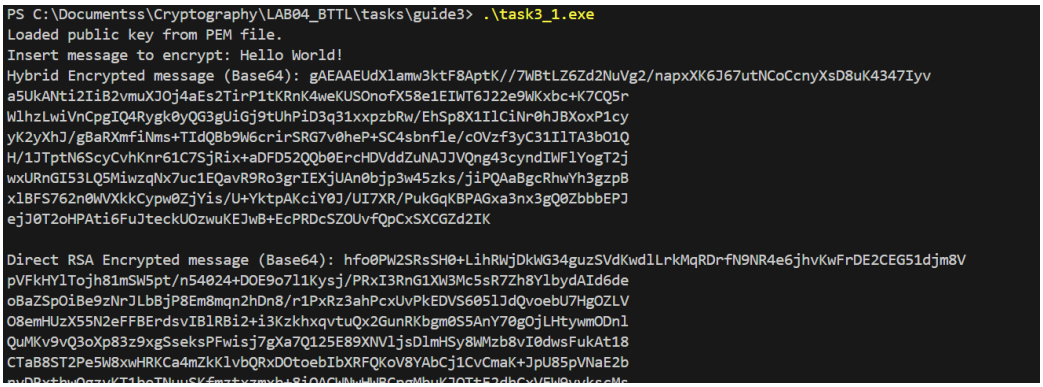
3 Guide3_rsa_encryption

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

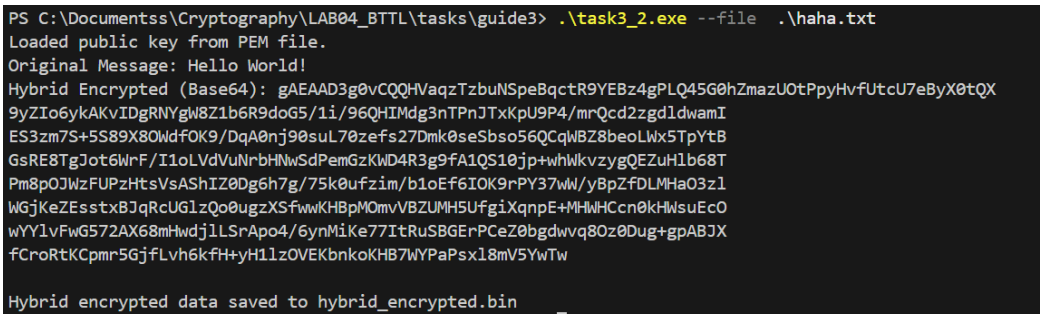
Target: Encryption use cin to get message from screen.



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)

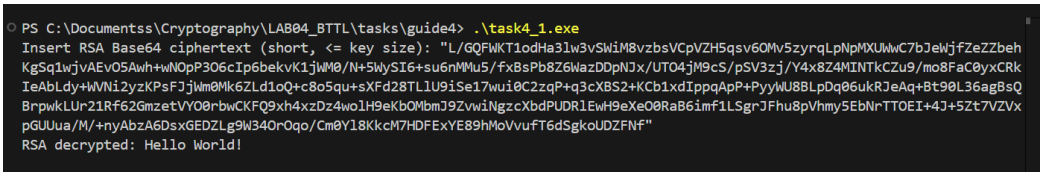


Hình 5: Kết quả thu được sau khi lấy plaintext từ file

4 Guide4_rsa_decryption

4.1 Task4.1

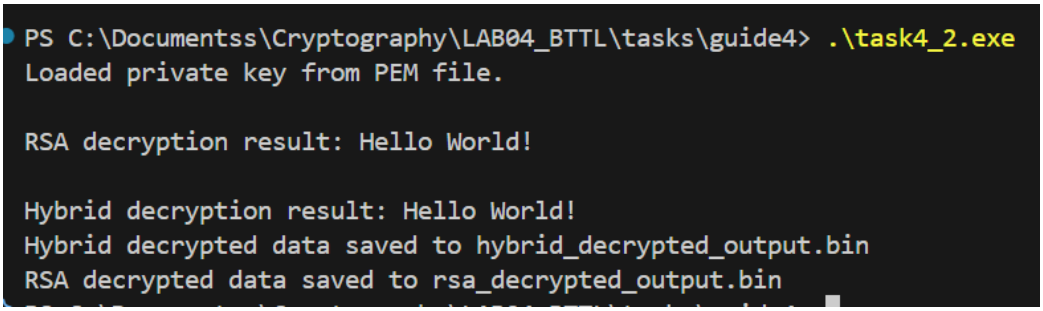
Target: Decryption use cin to get message from screen



Hình 6: Mã hóa ciphertext với cin input

4.2 Task4.2

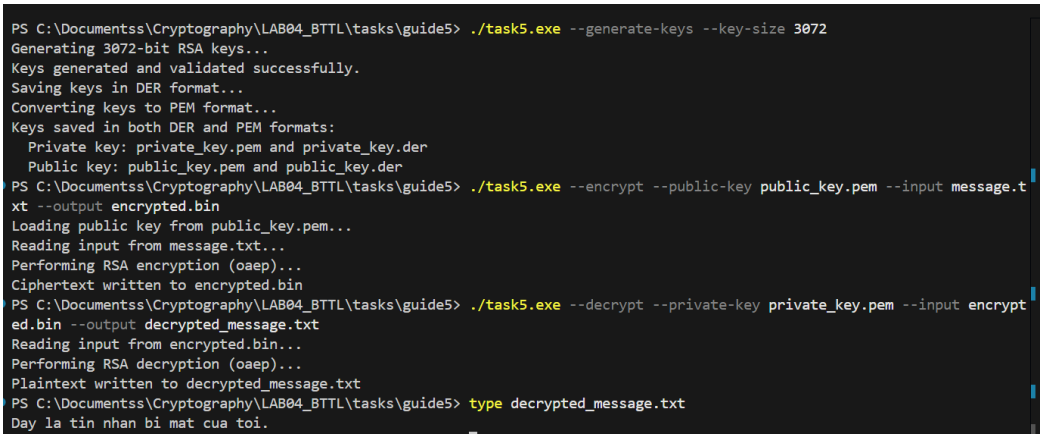
Target: Decryption (OAEP_SHA) use File/Source/Sink to get message (bin) from file and save output to file (bin)



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.

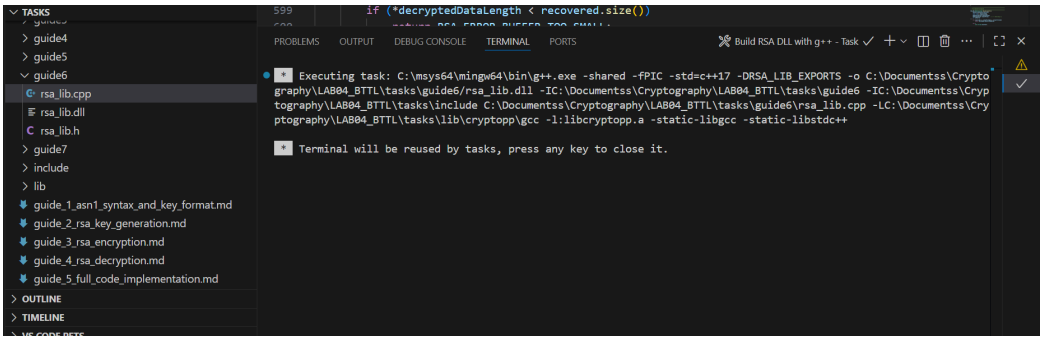


Hình 8: Kết quả chạy test CLI task5

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.

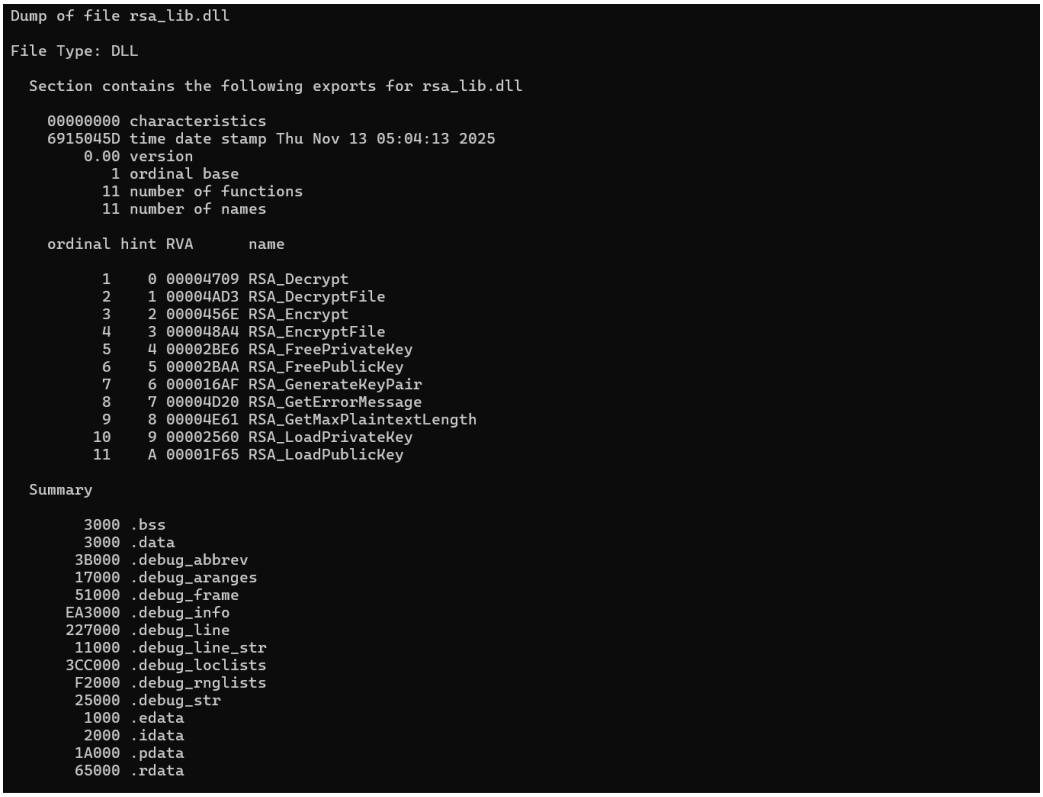
6.1 Windows



Hình 9: Kết quả build file DLL

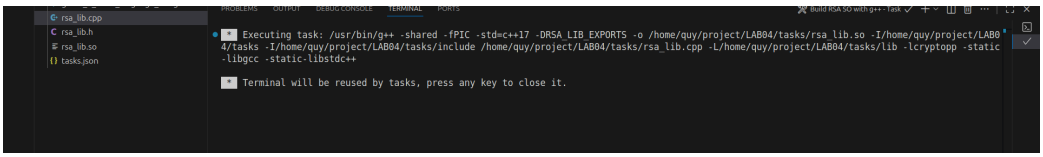
```
C:\Documentss\Cryptography\LAB04_BTTL\tasks\guide6>dumpbin /dependents rsa_lib.dll
Microsoft (R) COFF/PE Dumper Version 14.44.35214.0
Copyright (C) Microsoft Corporation.  All rights reserved.


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



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.

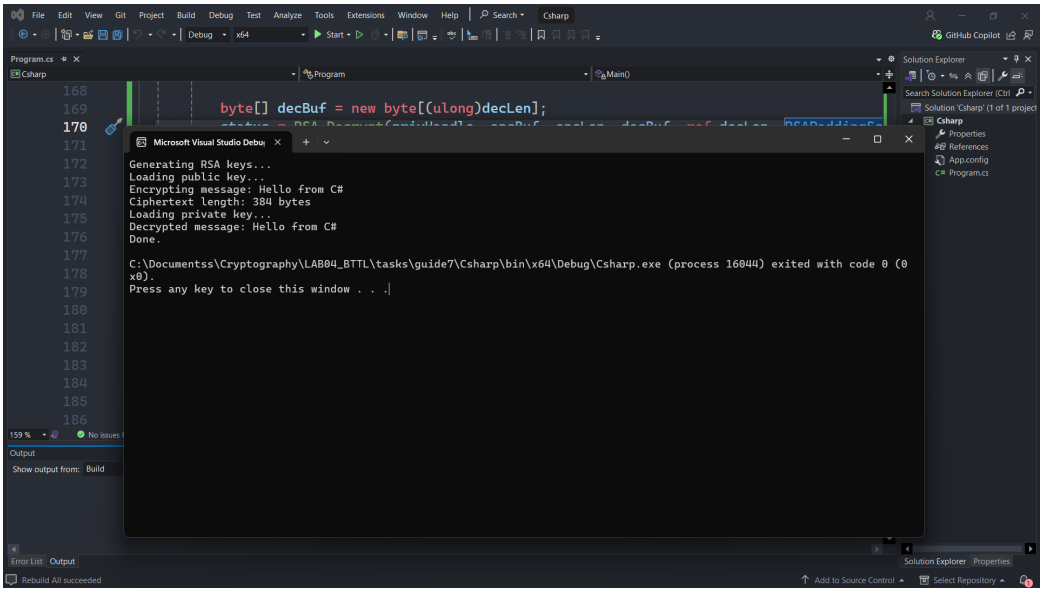
7.1 Windows

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.
```

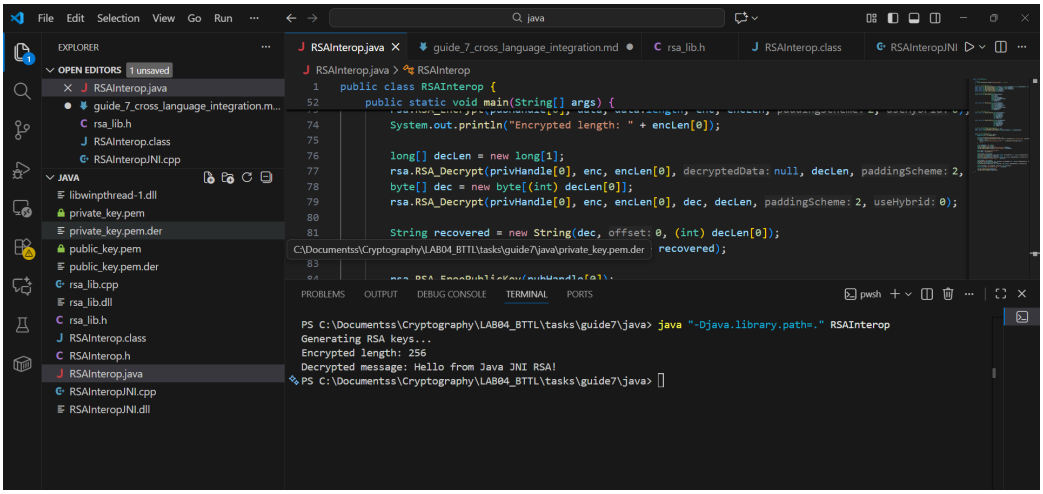
Hình 13: Kết quả chạy rsa_example.py

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

```
quy@quy-ASUS-TUF-Gaming-F15-FX507VV-FX507VV:~/project/LAB04/tasks$ python3 ./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 16: Kết quả chạy rsa_example.py trên Linux