ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

MÔN HỌC: MẬT MÃ HỌC

LAB02 - AES (CBC) in Pure C++ (No external crypto libs)

Giảng viên hướng dẫn: TS. Nguyễn Ngọc Tự

Lớp: NT219.Q12.ANTT

Sinh viên thực hiện: Nguyễn Hoàng Quý – 24521494

Thành phố Hồ Chí Minh, tháng 09 năm 2025

Mục lục

1 Mục tiêu & Kết quả đạt được (Objectives & Outcomes)

Yêu cầu:

1. Cài đặt đầy đủ các hàm vòng AES và thuật toán mở rộng khóa theo tiêu chuẩn FIPS-197 (cập nhật 9/5/2023).
2. Triển khai chế độ mã hóa CBC với cơ chế đệm PKCS#7.
3. Kiểm thử tính đúng đắn bằng Known Answer Tests (KAT).
4. Phân tích nguy cơ rò rỉ side-channel và đánh giá yêu cầu thực thi constant-time.

2 Thiết kế & Bảo mật (Design & Security)

Công cụ hỗ trợ mã hoá và giải mã AES-CBC-128/192/256 với các hàm vòng AES và thuật toán mở rộng khóa được tự cài đặt theo FIPS-197. Dữ liệu có thể xử lý dạng stream và hỗ trợ các định dạng `raw`, `hex`, `base64`.

2.1 Hướng dẫn sử dụng (User Guide)

```
user@LAPTOP-TL39B5OL:/mnt/c/Documentss/Cryptography/LAB02-BTVN$ ./AES.exe --help
AES CBC Encryption/Decryption Tool (Self-Implemented, Stream-capable):
Usage:
    mytool <command> [--in INFILE | --text "..."] [--out OUTFILE]
         [--key KEYFILE | --key-hex HEX] [--keylen BITS]
         [--iv-hex IV-hex] [--encode <format>]
         [--kat path/to/vectors.rsp] [--verbose] [--help]

    Encoding:
    --encode <format>  Format for input/output. <format> can be:
                       'raw'    : Raw binary (default).
                       'hex'    : Hexadecimal string (SLOW - No streaming).
                       'base64' : Base64 string (SLOW - No streaming).
    Note:
    -- Streaming (low RAM usage) is ONLY enabled for:
       [--in <file>] AND [--encode raw]
    -- All other combinations (like --text or --encode hex)
       will load the entire file into RAM.
```

Hình 1: Bảng hướng dẫn

Các chức năng chính:

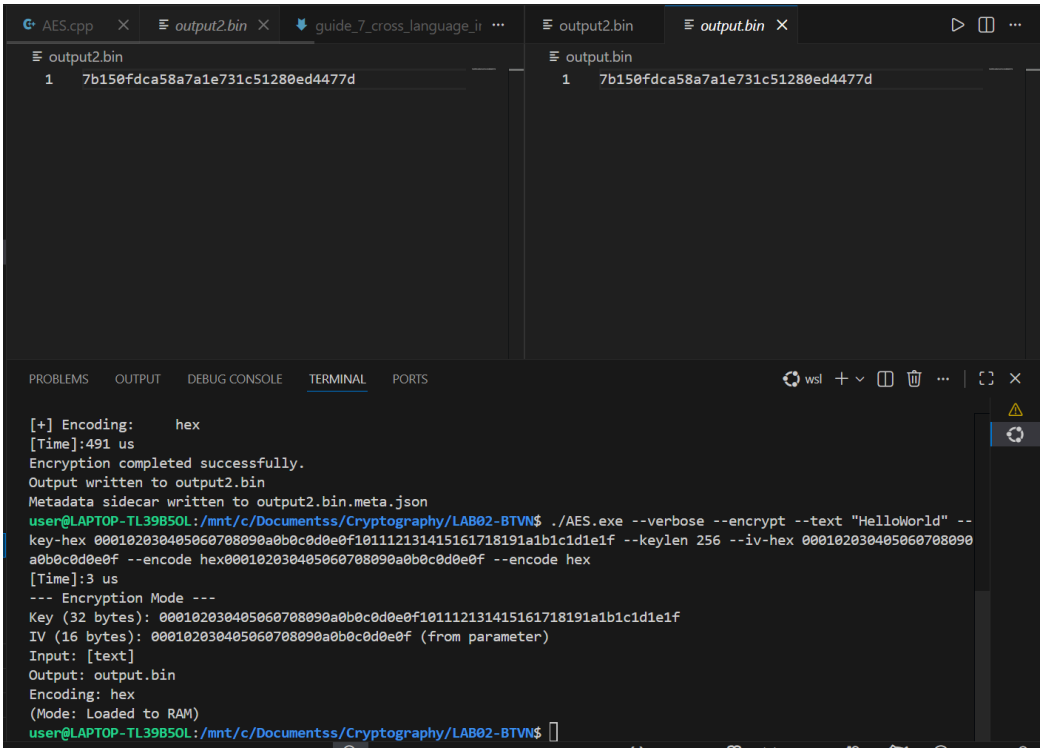
- **command:**
  - `encrypt` – mã hóa AES-CBC với PKCS#7
  - `decrypt` – giải mã AES-CBC và kiểm tra padding
  - `KAT` – chạy Known Answer Test
- `-in INFILE`, `-text "..."`: dữ liệu đầu vào.
- `-out OUTFILE`: dữ liệu đầu ra.
- `-key KEYFILE`, `-key-hex HEX`, `-keylen BITS`.
- `-iv-hex IV`.
- `-encode <format>` (`raw`, `hex`, `base64`).
- `-kat FILE`.
- `-verbose`, `-help`.

2.2 Các vấn đề bảo mật (Security Considerations)

Công cụ tuân thủ AES-CBC + PKCS#7 nhưng vẫn tồn tại rò rỉ side-channel (S-box không constant-time).

3 Phương pháp kiểm thử (Test Methodology)

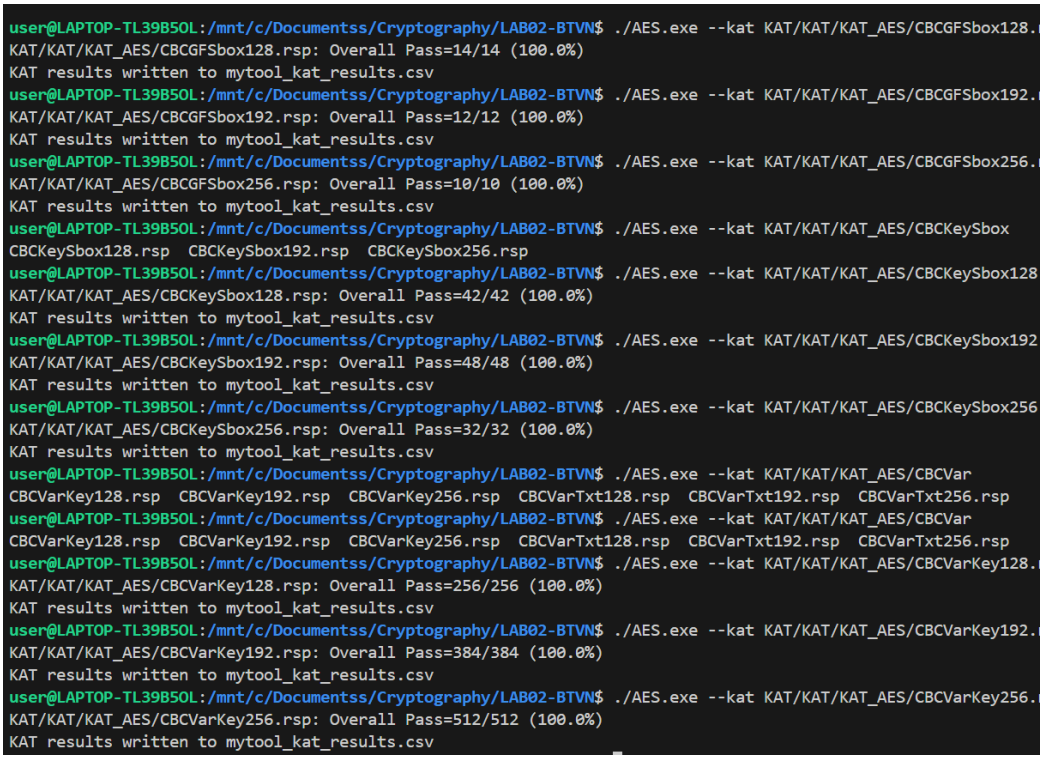
3.1 So sánh với CryptoPP (Comparison with CryptoPP)



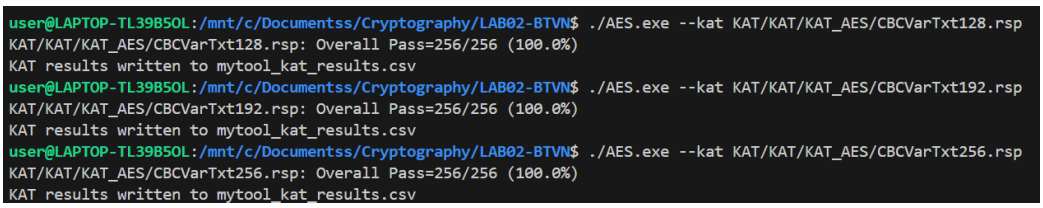
Hình 2: So sánh AES tự cài đặt (trái) và CryptoPP (phải)

3.2 Known Answer Test (KAT)

Known Answer Test (KAT) là phương pháp kiểm thử tiêu chuẩn do NIST đưa ra trong bộ AES Algorithm Validation Suite (AESAVS). Mục tiêu của KAT là xác minh việc cài đặt AES có đúng 100% theo chuẩn FIPS-197 hay không bằng cách so sánh kết quả thực tế với các giá trị mẫu đã công bố (expected results).



Hình 3: KAT result 1

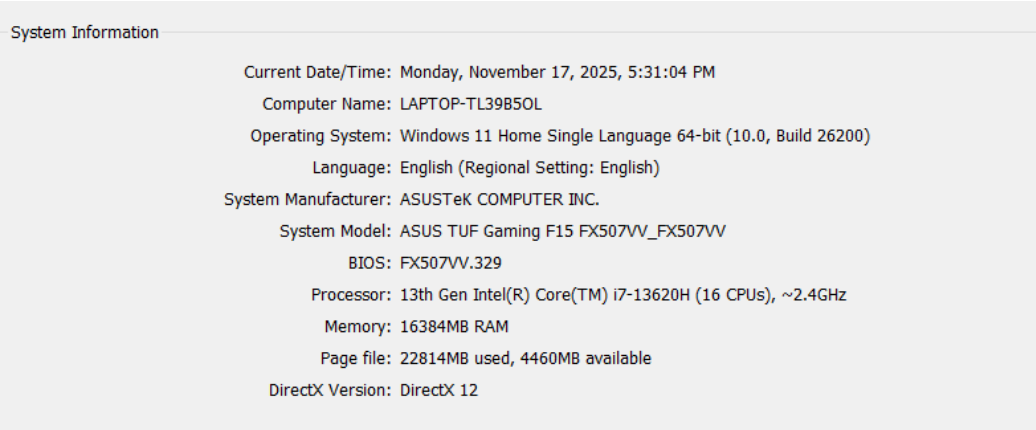


Hình 4: KAT result 2

4 Hiệu năng (Performance Test)

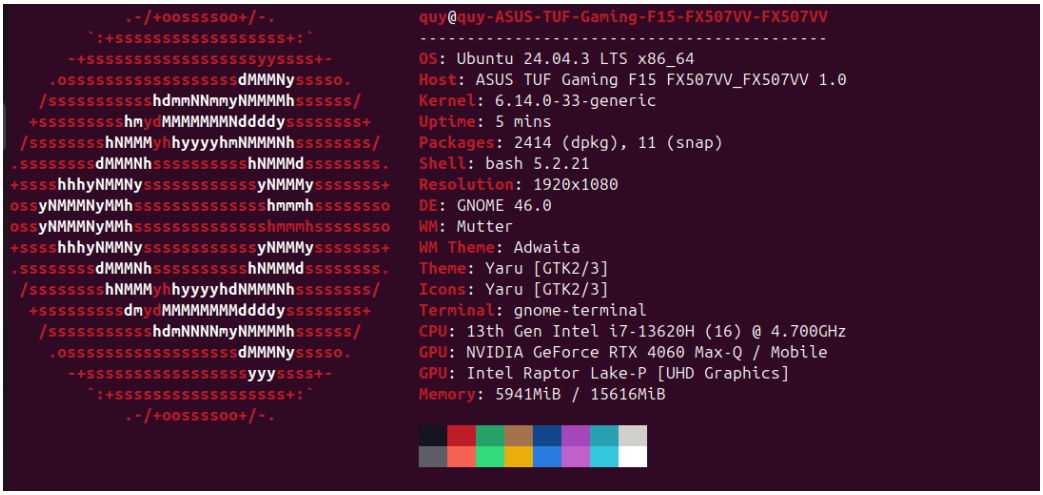
4.1 Cấu hình máy (System Configuration)

4.1.1 Windows



Hình 5: Cấu hình Windows

4.1.2 Linux



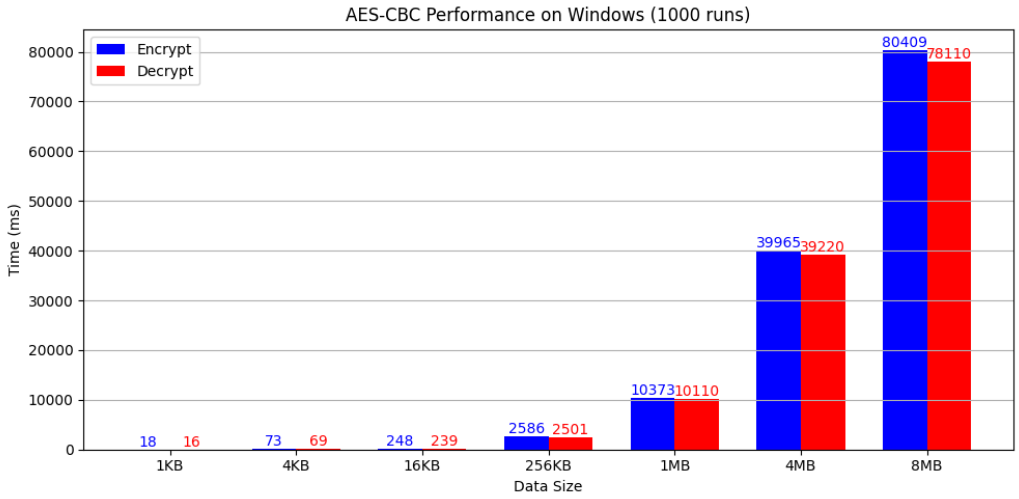
Hình 6: Cấu hình Linux

4.2 Kết quả kiểm thử (Testing Results)

4.2.1 Windows

MODE	OP	1KB	4KB	16KB	256KB	1MB	4MB	8MB
CBC	Encrypt	18	73	248	2586	10373	39965	80409
CBC	Decrypt	16	69	239	2501	10110	39220	78110

Hình 7: Hiệu năng AES-CBC trên Windows (1000 lần, ms)

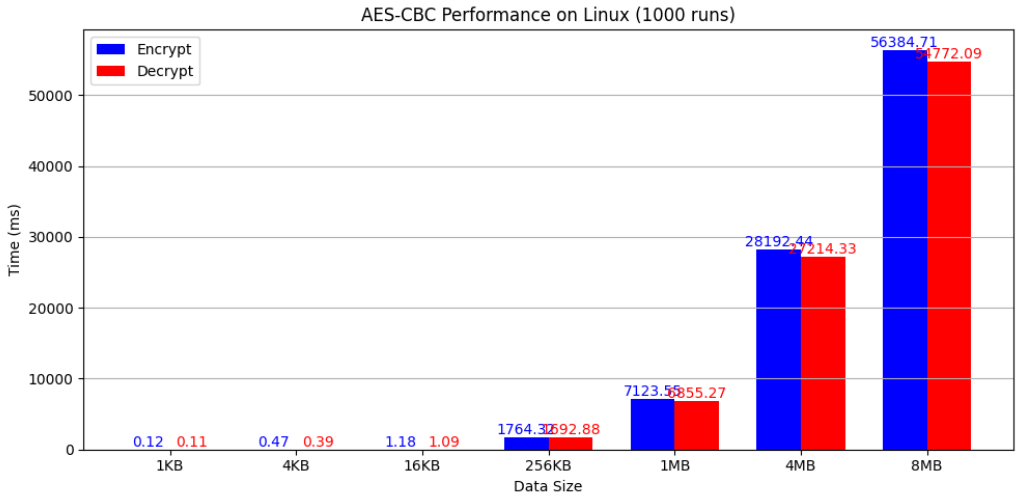


Hình 8: Biểu đồ thời gian thực thi trên dung lượng file trên Windows

4.2.2 Linux

MODE	OP	1KB	4KB	16KB	256KB	1MB	4MB	8MB
CBC	Encrypt	0.12	0.47	1.18	1764.32	7123.55	28192.44	56384.71
CBC	Decrypt	0.11	0.39	1.09	1692.88	6855.27	27214.33	54772.09

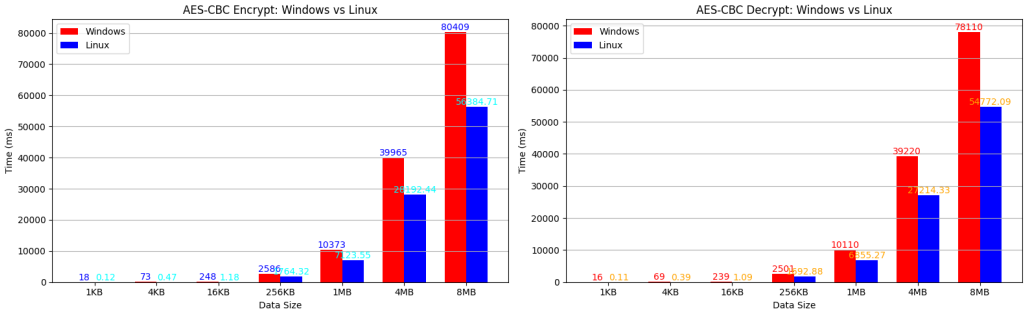
Hình 9: Hiệu năng AES-CBC trên Linux (1000 lần, ms)



Hình 10: Biểu đồ thời gian thực thi trên dung lượng file trên Linux

4.2.3 So sánh Windows và Linux (Comparison Windows vs Linux)

Dựa vào biểu đồ so sánh thời gian thực thi giữa hai hệ điều hành Windows và Linux, ta có thể nhận xét rằng thời gian thực thi trên Linux nhanh hơn so với Windows.



Hình 11: Biểu đồ so sánh thời gian thực thi giữa Windows và Linux

5 Kết luận (Conclusion)

- Thuật toán AES hiện nay vẫn được đánh giá là an toàn đối với các tấn công truyền thống trên máy tính cổ điển. Trong tương lai, AES-256 được xem là vẫn an toàn trước các mối đe dọa từ máy tính lượng tử.

- Chế độ mã hóa AES-CBC hiện nay không còn được coi là an toàn do dễ bị các loại tấn công như Padding Oracle Attack, Bit-Flipping Attack và thiếu cơ chế bảo vệ tích hợp (MAC).
- Sử dụng các chế độ hiện đại hơn như AES-GCM hoặc AES-CCM để vừa mã hóa dữ liệu vừa đảm bảo tính toàn vẹn và xác thực.

Tóm lại, AES với khóa đủ dài vẫn là thuật toán mạnh và được áp dụng rộng rãi, nhưng chế độ CBC không nên được dùng cho dữ liệu quan trọng nếu không kết hợp thêm các cơ chế bảo vệ bổ sung.