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

Lab 3 — RSA-OAEP with Crypto++

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. **Generate & store RSA keys safely**: Xây dựng chức năng tạo và lưu trữ cặp khóa RSA theo chuẩn PEM, đảm bảo an toàn, hỗ trợ độ dài khóa linh hoạt (2048–4096 bits), đồng thời tạo metadata đi kèm để phục vụ việc quản lý khóa.

2. **Implement OAEP (SHA–256) encryption/decryption**: Cài đặt thuật toán mã hóa và giải mã theo chuẩn RSA-OAEP sử dụng hàm băm SHA-256, bao gồm xử lý padding, kiểm tra tính hợp lệ và hỗ trợ nhiều định dạng đầu vào/đầu ra (raw, hex, base64).

3. **Handle chunking for large inputs**: Triển khai cơ chế chia nhỏ dữ liệu khi mã hóa, bao gồm chế độ mã hóa Hybrid (RSA + AES-GCM) khi kích thước đầu vào vượt khả năng mã hóa trực tiếp của RSA-OAEP. Hệ thống đảm bảo toàn vẹn dữ liệu và tự động sinh metadata giải mã.

4. **Measure performance vs key size**: Đo đạc và phân tích hiệu năng (thời gian tạo khóa, mã hóa, giải mã) theo các độ dài khóa khác nhau; từ đó đánh giá mức độ đánh đổi giữa tính bảo mật và tốc độ xử lý của hệ thống.

Kết quả đạt được: Hệ thống hoàn thiện toàn bộ các tính năng trên, cung cấp giao diện dòng lệnh (CLI) trực quan, hỗ trợ nhiều chế độ hoạt động, và đảm bảo tính an toàn – toàn vẹn cho quá trình mã hóa/giải mã. Các thử nghiệm hiệu năng chứng minh hệ thống hoạt động ổn định với nhiều kích thước khóa khác nhau.

2 Thiết kế & Bảo mật (Design & Security)

2.1 Hướng dẫn sử dụng (User Guide)

Các chức năng chính:

- **command:**
  - `-genkey` – Tạo cặp khóa RSA theo chiều dài yêu cầu (mặc định: 3072 bits).

  - `-encrypt` – Mã hóa dữ liệu bằng RSA-OAEP-SHA256. Tự động chuyển sang Hybrid Encryption (RSA + AES-GCM) nếu kích thước dữ liệu vượt giới hạn RSA.
  - `-decrypt` – Giải mã dữ liệu theo định dạng đã mã hóa (RSA hoặc Hybrid).

- **Input/Output:**
  - `-in INFILE`: Đường dẫn file đầu vào.
  - `-text "..."`: Nhập dữ liệu dạng chuỗi trực tiếp.
  - `-out OUTFILE`: File đầu ra (mặc định: `output.bin`).

- **Key Options:**
  - `-pub PUBFILE`: Đường dẫn Public Key (mặc định: `public_key.pem`).
  - `-priv PRIVFILE`: Đường dẫn Private Key (mặc định: `private_key.pem`).
  - `-bits BITS`: Độ dài khóa RSA khi tạo khóa.

- **Encoding:**
  - `-encode <format>`: Định dạng mã hóa đầu ra, hỗ trợ:
    * `hex`
    * `base64`

    (Lưu ý: Giải mã tự phát hiện định dạng nếu không phải dạng raw.)

- **Chế độ hiển thị:**
  - `-verbose`: Hiển thị chi tiết quá trình.
  - `-help`: Hiển thị thông tin hướng dẫn.

Manual Key Components (dùng với `-genkey` tùy chỉnh)

Các tham số thủ công (dành cho người dùng nâng cao):

- `-n HEX`: Modulus (n).
- `-e HEX`: Public exponent (e).
- `-d HEX`: Private exponent (d).
- `-p HEX`: Prime 1 (p).
- `-q HEX`: Prime 2 (q).

Cơ chế Hybrid Encryption

Công cụ tự động chuyển sang Hybrid Encryption khi kích thước dữ liệu vượt quá khả năng mã hóa trực tiếp của RSA-OAEP.

- Sinh khóa AES-256.
- Mã hóa dữ liệu bằng AES-GCM.
- Mã hóa AES key bằng RSA-OAEP-SHA256.
- Tất cả các giá trị (nonce, tag, ciphertext, RSA-wrapped key) được ghi vào file đầu ra.
- Tạo file `.meta.json` chứa metadata (phiên bản, kiểu mã hóa, kích thước khóa...).

Ví dụ sử dụng

- **Tạo khóa 3072-bit:**

  ```
  mytool --genkey --bits 3072
  ```

- **Mã hóa file:**

  ```
  mytool --encrypt --in data.bin --out data.bin
  ```

- **Giải mã file:**

  ```
  mytool --decrypt --in data.bin --out data.bin
  ```

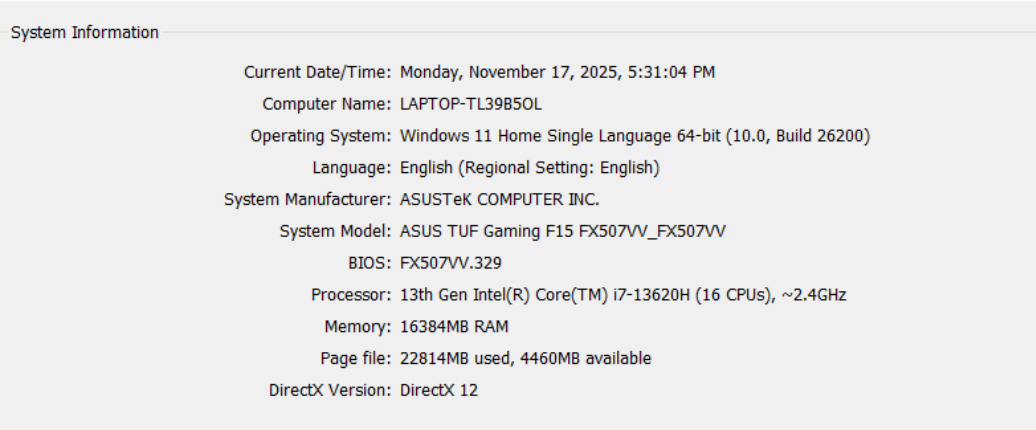
- **Mã hóa từ text, xuất base64:**

  ```
  mytool --encrypt --text "Hello world" --encode base64
  ```

3 Kiểm thử hiệu năng (Performance Test)

3.1 Cấu hình máy (System Configuration)

3.1.1 Windows



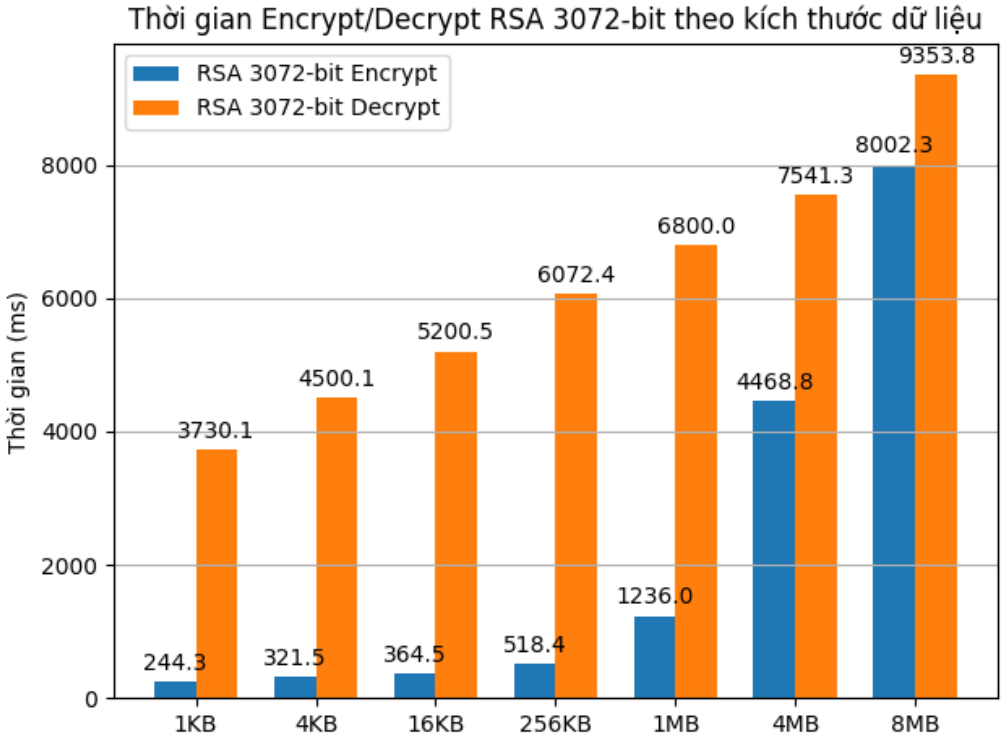
Hình 1: Cấu hình máy windows

3.2 Kết quả kiểm thử (Testing Results)

3.2.1 Windows 3072

Bảng 1: Thời gian mã hóa và giải mã RSA 3072-bit theo kích thước dữ liệu (đơn vị: ms)

Mode	**1KB**	**4KB**	**16KB**	**256KB**	**1MB**	**4MB**	**8MB**
RSA 3072 Encrypt	244.3	321.5	364.5	518.4	1236.0	4468.8	8002.3
RSA 3072 Decrypt	3730.1	4500.1	5200.5	6072.4	6800.0	7541.3	9353.8

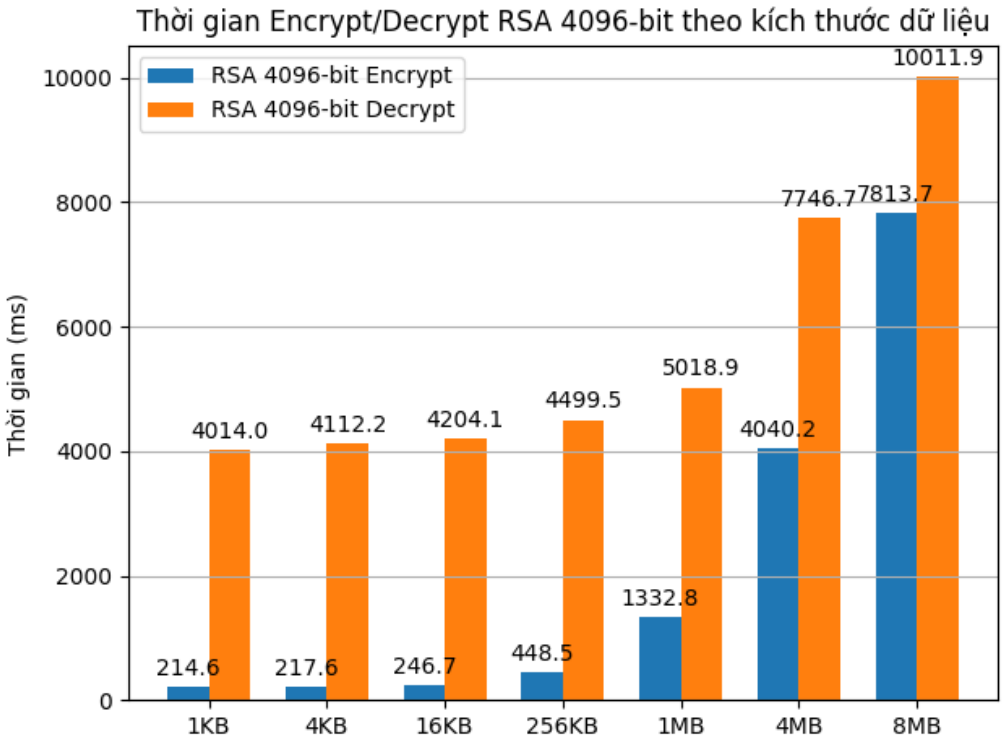


Hình 2: So sánh thời gian thực thi của keysize 3072 bits

3.2.2 Windows 4096

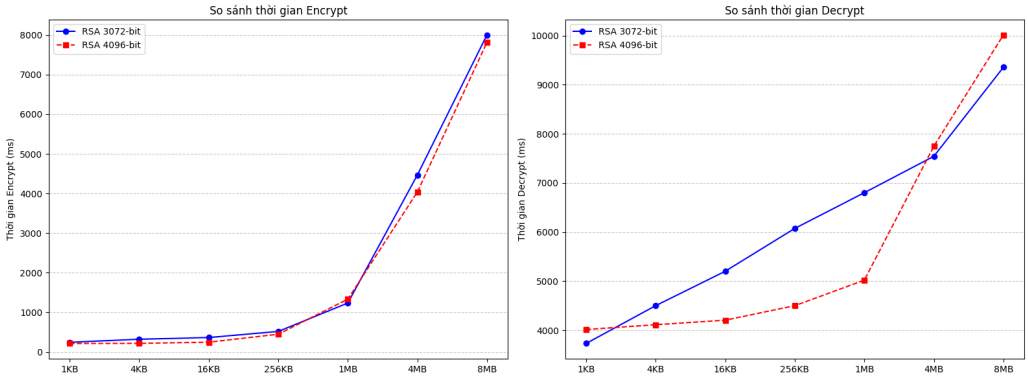
Bảng 2: Thời gian mã hóa và giải mã RSA 4096-bit theo kích thước dữ liệu (đơn vị: ms)

Mode	**1KB**	**4KB**	**16KB**	**256KB**	**1MB**	**4MB**	**8MB**
RSA 4096 Encrypt	214.6	217.6	246.7	448.5	1332.8	4040.2	7813.7
RSA 4096 Decrypt	4014.0	4112.2	4204.1	4499.5	5018.9	7746.7	10011.9



Hình 3: So sánh thời gian thực thi của keysize 4096 bits

3.2.3 So sánh 3072 bits và 4096 bits



Hình 4: So sánh thời gian giữa 3072 bits và 4096 bits

4 Kết luận (Conclusion)

4.1 Về tính an toàn của RSA (Security Analysis)

- **Độ dài khóa (Key Size):** Với năng lực tính toán của máy tính cổ điển hiện nay, độ dài khóa RSA **3072-bit** được NIST khuyến nghị là mức an toàn tiêu chuẩn (tương đương 128-bit security strength) dùng cho tương lai gần, và **4096-bit** là mức an toàn rất cao dùng cho lưu trữ dài hạn. Việc cài đặt hỗ trợ cả hai độ dài này đảm bảo hệ thống chống lại được các tấn công phân tích thừa số nguyên tố (factorization attacks) hiện hành.
- **Cơ chế Padding (OAEP):** Việc bắt buộc sử dụng OAEP (Optimal Asymmetric Encryption Padding) kết hợp với hàm băm SHA-256 là tối quan trọng. Nó giúp hệ thống đạt được tính an toàn ngữ nghĩa (semantic security), chống lại các tấn công bản mã lựa chọn (CCA) và khắc phục các điểm yếu tất định của RSA trơn (textbook RSA).
- **Hybrid Encryption:** Với các tập tin kích thước lớn, việc kết hợp AES-GCM đảm bảo tính toàn vẹn dữ liệu (integrity) và xác thực (authentication), điều mà RSA đơn thuần khó thực hiện hiệu quả.

4.2 Nhận xét và Phân tích kết quả (Analysis)

Dựa trên biểu đồ so sánh thời gian thực thi giữa RSA 3072-bit và RSA 4096-bit (Hình 4), ta có những nhận xét sau:

- **Đối với quá trình Mã hóa (Encrypt):** Đường biểu diễn thời gian của hai độ dài khóa gần như bám sát nhau và tăng tuyến tính theo kích thước dữ liệu.
  - Tại mức dữ liệu lớn nhất (8MB), thời gian xử lý của cả hai đều dao động quanh mức 8000ms.
  - *Kết luận:* Việc tăng độ dài khóa từ 3072 lên 4096 bit **không gây ảnh hưởng đáng kể** đến tốc độ mã hóa. Điều này phù hợp với lý thuyết do số mũ công khai (e) thường nhỏ và cố định.
- **Đối với quá trình Giải mã (Decrypt):** Sự khác biệt về hiệu năng thể hiện rõ rệt hơn so với quá trình mã hóa.
  - Tại kích thước nhỏ (1KB), RSA 4096-bit chậm hơn so với 3072-bit (khoảng 4000ms so với 3700ms).

– Khi kích thước file tăng lên mức cực đại trong bài test (8MB), đường biểu diễn của RSA 4096-bit (nét đứt màu đỏ) vọt lên mức hơn 10.000ms, cao hơn đáng kể so với RSA 3072-bit (khoảng 9.300ms).

– *Kết luận:* Độ dài khóa càng lớn, chi phí tính toán cho việc giải mã càng cao. Do đó, RSA 4096-bit có sự **đánh đổi lớn về hiệu năng giải mã** để đổi lấy độ bảo mật cao hơn.