**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: NHẬP MÔN AN TOÀN, BẢO MẬT THÔNG TIN**

**ĐỀ TÀI: ỨNG DỤNG BẢO MẬT TIN NHẮN VĂN BẢN VỚI MÃ HÓA DES VÀ XÁC THỰC RSA**

**Giáo viên hướng dẫn: Trần Đức Thắng**

**Sinh viên thực hiện:**  1. Đỗ Thị Phương Thảo

2. Nguyễn Trương Thuận

3. Đinh Anh Trúc

4. Mai Văn Tiến

**Hà Nội, 2025**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: NHẬP MÔN AN TOÀN, BẢO MẬT THÔNG TIN**

**ĐỀ TÀI: ỨNG DỤNG BẢO MẬT TIN NHẮN VĂN BẢN VỚI MÃ HÓA DES VÀ XÁC THỰC RSA**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Mã Sinh Viên** | **Họ và Tên** | **Ngày Sinh** | **Điểm** | |
| **Bằng Số** | **Bằng Chữ** |
| **1** | **1771020642** | **Đỗ Thị Phương Thảo** | **04/10/2005** |  |  |
| **2** | **1771020655** | **Nguyến Trương Thuận** | **07/12/2005** |  |  |
| **3** | **1771020686** | **Đinh Anh Trúc** | **15/02/2005** |  |  |
| **4** | **1771020669** | **Mai Văn Tiến** | **03/09/2005** |  |  |

### 

CÁN BỘ CHẤM THI

**Hà Nội, 2025**

**LỜI NÓI ĐẦU**

Trong bối cảnh công nghệ thông tin ngày càng phát triển và được ứng dụng rộng rãi trong mọi lĩnh vực của đời sống, vấn đề an toàn và bảo mật thông tin trở nên cấp thiết hơn bao giờ hết. Đặc biệt, trong các hệ thống truyền tin như ứng dụng nhắn tin, việc đảm bảo tính bí mật, xác thực, và toàn vẹn của dữ liệu là những yêu cầu cơ bản nhưng mang tính sống còn.

Được sự hướng dẫn tận tình của giảng viên môn Nhập môn An toàn bảo mật thông tin, nhóm chúng em đã lựa chọn và thực hiện đề tài “Ứng dụng bảo mật tin nhắn văn bản với mã hoá DES và xác thực RSA” làm bài tập lớn. Thông qua đề tài này, nhóm không chỉ có cơ hội áp dụng kiến thức lý thuyết đã học vào thực tế, mà còn rèn luyện kỹ năng làm việc nhóm, lập trình và tư duy bảo mật. Hệ thống được thiết kế nhằm mô phỏng quá trình gửi và nhận tin nhắn một cách an toàn, sử dụng các kỹ thuật như mã hóa đối xứng (DES), mã hóa khóa công khai (RSA), cùng với hàm băm SHA-256 để kiểm tra toàn vẹn dữ liệu.

Trong quá trình thực hiện, nhóm chúng em đã cố gắng hoàn thành tốt các yêu cầu của đề tài và tích lũy thêm nhiều kiến thức bổ ích. Tuy nhiên, do thời gian và trình độ còn hạn chế, bài làm không tránh khỏi những thiếu sót. Nhóm mong nhận được sự đóng góp ý kiến từ thầy cô để có thể cải thiện và nâng cao hơn nữa chất lượng sản phẩm cũng như kiến thức của bản thân.

Nhóm xin chân thành cảm ơn!

**MỤC LỤC**

[CHƯƠNG 1. ĐẶT VẤN ĐỀ VÀ PHÂN TÍCH YÊU CẦU BÀI TOÁN 10](#_Toc1491)

[1.1. Bối cảnh 10](#_Toc28179)

[1.2. Phân tích yêu cầu 10](#_Toc6182)

[*1.2.1. Mã hóa dữ liệu – đảm bảo bảo mật thông tin* 10](#_Toc21414)

[*1.2.2. Xác thực người dùng – đảm bảo đúng người đúng quyền* 11](#_Toc3438)

[*1.2.3. Kiểm tra tính toàn vẹn – chống giả mạo và thay đổi dữ liệu* 11](#_Toc3818)

[CHƯƠNG 2. MÔ TẢ THUẬT TOÁN VÀ 3 BƯỚC SAU 13](#_Toc25322)

[2.1. Bước 1: Bắt tay (Handshake) đơn giản 13](#_Toc28354)

[2.2. Bước 2: Xác thực (Ký số và trao đổi khóa) 13](#_Toc16416)

[2.3. Bước 3: Truyền dữ liệu và kiểm tra toàn vẹn 14](#_Toc2050)

[CHƯƠNG 3. PHÂN TÍCH MÃ NGUỒN 16](#_Toc27740)

[3.1 DES (Data Encryption Standard) – chế độ CFB 16](#_Toc31619)

[3.2 RSA (Rivest–Shamir–Adleman) 17](#_Toc2270)

[3.3 SHA-256 (Secure Hash Algorithm – 256 bit) 18](#_Toc8715)

[CHƯƠNG 4. THỬ NGHIỆM 20](#_Toc32185)

[4.1. Mục tiêu thử nghiệm 20](#_Toc29757)

[4.2. Kịch bản thử nghiệm 20](#_Toc13943)

[*4.2.1. Mô tả hệ thống* 20](#_Toc30501)

[*4.2.2. Các bước thử nghiệm* 20](#_Toc29942)

[4.3. Thử nghiệm với các loại dữ liệu 21](#_Toc31270)

[*4.3.1. Nội dung các bài kiểm thử* 21](#_Toc29725)

[*4.3.2. Quan sát thực nghiệm* 23](#_Toc6481)

[4.4. Đánh giá hệ thống 24](#_Toc8543)

[4.5. Kết luận 24](#_Toc20328)

[CHƯƠNG 5. ĐÁNH GIÁ HIỆU QUẢ 25](#_Toc11041)

[5.1. Đảm bảo tính đúng đắn của dữ liệu sau giải mã 25](#_Toc32037)

[5.2. Kiểm tra tính toàn vẹn của dữ liệu truyền tải 25](#_Toc21374)

[5.3. Kết luận: 25](#_Toc14197)

[CHƯƠNG 6. PHÂN TÍCH, NHẬN XÉT ĐẶC ĐIỂM CỦA CÁC THUẬT TOÁN ĐƯỢC SỬ DỤNG ĐỂ XÂY DỰNG CHƯƠNG TRÌNH 26](#_Toc12798)

[6.1. DES (Data Encryption Standard) – Mã hoá đối xứng 26](#_Toc31941)

[6.2. RSA (Rivest–Shamir–Adleman) – Mã hóa bất đối xứng 26](#_Toc9278)

[6.3. SHA-256 (Secure Hash Algorithm – 256 bit) – Hàm băm 27](#_Toc12699)

[CHƯƠNG 7. ĐỀ XUẤT CẢI TIẾN VÀ HƯỚNG PHÁT TRIỂN CHƯƠNG TRÌNH TRONG TƯƠNG LAI 28](#_Toc27801)

[7.1. Thay thế thuật toán DES bằng AES 28](#_Toc1806)

[7.2. Bổ sung xác thực hai chiều 28](#_Toc14160)

[7.3. Phát triển giao diện đồ họa người dùng (GUI) 28](#_Toc16856)

[7.4. Hỗ trợ nhiều người dùng và quản lý khóa 28](#_Toc16839)

[7.5. Mở rộng kết nối trên mạng thực (Internet) 28](#_Toc20244)

[7.6. Bổ sung kiểm tra chống replay attack 29](#_Toc31950)

[7.7. Thử nghiệm thêm với các thuật toán khác 29](#_Toc15343)

[KẾT LUẬN 30](#_Toc2576)

[DANH MỤC TÀI LIỆU THAM KHẢO 31](#_Toc4047)

**MỤC LỤC HÌNH ẢNH**

[Hình 1 . Mã hóa dữ liệu 16](#_Toc15536)

[Hình 2 . Giải mã dữ liệu 16](#_Toc1623)

[Hình 3 . Mã hóa khóa 17](#_Toc15819)

[Hình 4 . Tạo chũ ký số 18](#_Toc7796)

[Hình 5 . Xác minh chữ ký 18](#_Toc27487)

[Hình 6 . Tạo mã băm 18](#_Toc12017)

[Hình 7 . Văn bản sau khi mã hóa 21](#_Toc3395)

[Hình 8 . Kiểm thử 22](#_Toc26229)

[Hình 9 . Mã hóa 22](#_Toc12403)

[Hình 10 . Kết quả 23](#_Toc27647)

**MỤC LỤC BẢNG**

[Bảng 1 . Ghi nhận lỗi 23](#_Toc20091)

**BẢNG CÁC TỪ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| **STT** | **TỪ VIẾT TẮT** | **VIẾT ĐẦY ĐỦ** |
| 1 | CSDL | Cơ sở dữ liệu |
| 2 | DES | Data Encryption Standard |
| 3 | CFB | Cipher Feedback Mode |
| 4 | RSA | Rivest–Shamir–Adleman |
| 5 | SHA-256 | Secure Hash Algorithm 256-bit |
| 6 | IV | Initialization Vector |
| 7 | ECB | Electronic Codebook Mode |
| 8 | CBC | Cipher Block Chaining Mode |
| 9 | K\_DES | Khóa phiên sử dụng thuật toán DES |
| 10 | JSON | JavaScript Object Notation |
| 11 | CSV | Comma-Separated Values |
| 12 | MP3 | MPEG-1 Audio Layer 3 |
| 13 | Colab | Google Colaboratory |
| 14 | NACK | Negative Acknowledgment |
| 15 | OAEP | Optimal Asymmetric Encryption Padding |
| 16 | MD5 | Message Digest 5 |
| 17 | AES | Advanced Encryption Standard |
| 18 | GUI | Graphical User Interface |
| 19 | CLI | Command Line Interface |
| 20 | ID | Identifier (Định danh người dùng) |
| 21 | LAN | Local Area Network |
| 22 | P2P | Peer-to-Peer |
| 23 | TCP/IP | Transmission Control Protocol / Internet Protocol |
| 24 | SSL | Secure Sockets Layer |
| 25 | TLS | Transport Layer Security |
| 26 | API | Application Programming Interface |
| 27 | ECC | Elliptic Curve Cryptography |

**CHƯƠNG 1. ĐẶT VẤN ĐỀ VÀ PHÂN TÍCH YÊU CẦU BÀI TOÁN**

**1.1. Bối cảnh**

Trong thời đại công nghệ số, việc trao đổi thông tin qua Internet là không thể thiếu, đặc biệt là trong các hệ thống nhắn tin, truyền dữ liệu văn bản, âm thanh hoặc video. Tuy nhiên, việc truyền dữ liệu qua mạng tiềm ẩn nhiều nguy cơ như:

Dữ liệu bị nghe lén

Giả mạo danh tính người gửi

Dữ liệu bị thay đổi trong quá trình truyền  
Để giải quyết các nguy cơ trên, bài toán đặt ra là xây dựng một hệ thống truyền tin an toàn, đảm bảo các yếu tố:

Bảo mật nội dung (confidentiality)

Xác thực người dùng (authentication)

Toàn vẹn dữ liệu (integrity)

**1.2. Phân tích yêu cầu**

***1.2.1. Mã hóa dữ liệu – đảm bảo bảo mật thông tin***

Yêu cầu:  
 Trong quá trình truyền tải qua Internet, dữ liệu dễ bị đánh cắp hoặc đọc trộm. Do đó, dữ liệu cần được mã hóa để người ngoài không thể hiểu được nội dung nếu không có khóa giải mã hợp lệ.

Giải pháp:  
 Sử dụng thuật toán DES (Data Encryption Standard) – một thuật toán mã hóa đối xứng, nghĩa là cùng một khóa được dùng để mã hóa và giải mã.

Cơ chế hoạt động:

Người gửi tạo một khóa phiên (session key) DES ngẫu nhiên.

Dữ liệu văn bản (hoặc dữ liệu âm thanh chuyển sang dạng nhị phân) được mã hóa bằng khóa DES này.

Bản mã (ciphertext) sau đó được gửi đi qua mạng.

Người nhận dùng khóa giải mã (được chia sẻ an toàn) để phục hồi lại dữ liệu gốc.

Lưu ý bảo mật:

DES là thuật toán cũ, tốc độ nhanh, phù hợp với bài toán quy mô nhỏ.

Mã hóa theo chế độ CFB (Cipher Feedback Mode) để tăng mức độ an toàn (tránh lộ mẫu dữ liệu nếu trùng plaintext).

***1.2.2. Xác thực người dùng – đảm bảo đúng người đúng quyền***

Yêu cầu:

Trong truyền thông số, bất kỳ ai cũng có thể giả danh người gửi. Vì vậy, hệ thống cần xác thực được người gửi là ai, và phải là người hợp lệ trong hệ thống.

Giải pháp:

Sử dụng thuật toán RSA (Rivest–Shamir–Adleman) để tạo chữ ký số. Đây là thuật toán mã hóa bất đối xứng, gồm một public key và một private key.

Cơ chế hoạt động:

Người gửi tạo chữ ký số bằng cách:

Băm thông tin nhận dạng (hoặc nội dung dữ liệu) bằng SHA-256.

Mã hóa mã băm này bằng private key RSA của người gửi → thành chữ ký số.

Người nhận:

Dùng public key của người gửi để giải mã chữ ký số.

Tự tính lại mã băm từ dữ liệu gốc.

Nếu hai mã băm trùng nhau → xác thực thành công.

Lưu ý bảo mật:

RSA đảm bảo chỉ người thật (nắm giữ private key) mới có thể ký.

Nếu kẻ tấn công gửi dữ liệu nhưng không có private key → không thể giả mạo được chữ ký.

***1.2.3. Kiểm tra tính toàn vẹn – chống giả mạo và thay đổi dữ liệu***

Yêu cầu:

Trong quá trình truyền tải, dữ liệu có thể bị thay đổi bởi hacker hoặc lỗi mạng. Hệ thống cần phát hiện ngay nếu dữ liệu đã bị chỉnh sửa, dù chỉ 1 ký tự.

Giải pháp:  
 Cơ chế hoạt động:

Người gửi:

Tính giá trị hash SHA-256 từ dữ liệu (hoặc bản mã).

Gửi giá trị hash này kèm dữ liệu (thường được ký lại bằng RSA để tăng độ tin cậy).

Người nhận:

Tính lại hash từ dữ liệu nhận được.

So sánh với hash ban đầu → nếu không trùng nhau thì dữ liệu đã bị thay đổi.

Lưu ý bảo mật:

SHA-256 có tính chống va chạm cao: 2 dữ liệu khác nhau gần như không thể tạo ra cùng 1 hash.

Dù thay đổi 1 ký tự nhỏ nhất → hash sẽ thay đổi toàn bộ → dễ phát hiện lỗi.

**CHƯƠNG 2. MÔ TẢ THUẬT TOÁN VÀ 3 BƯỚC SAU**

**2.1. Bước 1: Bắt tay (Handshake) đơn giản**

Mục đích:

· Tạo sự tin cậy ban đầu giữa hai bên (gửi và nhận), đồng thời thiết lập khóa phiên để sử dụng trong mã hóa dữ liệu.

Cơ chế hoạt động:

· Người gửi (A) và người nhận (B) chưa chia sẻ khóa DES, vì vậy cần một bước bắt tay để chia sẻ một session key (khóa phiên) qua một kênh an toàn.

Để làm điều này:

· A tạo ra một khóa DES ngẫu nhiên (k) dùng để mã hóa dữ liệu.

· A mã hóa khóa DES này bằng khóa công khai (public key) của B sử dụng thuật toán RSA.

· A gửi khóa đã mã hóa đến B.

Thuật toán sử dụng:

· RSA: dùng để mã hóa khóa phiên k, vì RSA đảm bảo chỉ người có private key (người nhận B) mới có thể giải mã được.

· DES chưa dùng ở bước này, chỉ được thiết lập khóa cho bước tiếp theo.

Tính bảo mật:

· Kẻ tấn công không thể lấy được khóa DES nếu không có private key RSA của người nhận → đảm bảo kênh chia sẻ khóa an toàn.

**2.2. Bước 2: Xác thực (Ký số và trao đổi khóa)**

Mục đích:

Đảm bảo rằng người gửi thật sự là người đáng tin cậy, không bị giả mạo, đồng thời xác minh tính toàn vẹn của khóa và thông tin đi kèm.

Cơ chế hoạt động:

Người gửi (A):

Tạo chữ ký số: Băm nội dung thông điệp hoặc khóa DES bằng SHA-256 → được một mã băm (hash1).

Mã hóa mã băm này bằng private key RSA của A → tạo chữ ký số.

Gửi: {Khóa DES đã mã hóa, chữ ký số, dữ liệu mã hóa} đến B.

Người nhận (B):

Dùng public key của A để giải mã chữ ký số → lấy được hash1.

Tự băm lại dữ liệu/khoá để lấy hash2.

So sánh hash1 và hash2 → nếu giống nhau, xác thực thành công.

Thuật toán sử dụng:

RSA: dùng để tạo và xác minh chữ ký số.

SHA-256: tạo mã băm cho nội dung cần xác thực.

DES: được sử dụng để mã hóa nội dung dữ liệu (sau khi handshake xong).

Tính bảo mật:

Đảm bảo người gửi là hợp lệ vì chỉ họ mới có private key.

Dữ liệu hoặc khóa bị sửa sẽ khiến mã băm không trùng khớp → giúp phát hiện giả mạo.

**2.3. Bước 3: Truyền dữ liệu và kiểm tra toàn vẹn**

Mục đích:

Đảm bảo dữ liệu được truyền đi không bị thay đổi trong quá trình truyền tải (cố tình hoặc lỗi kỹ thuật).

Cơ chế hoạt động:

Người gửi:

Mã hóa dữ liệu tin nhắn văn bản bằng DES + khóa phiên đã thiết lập từ bước bắt tay.

Băm dữ liệu gốc bằng SHA-256 để tạo mã kiểm tra (hash\_sent).

Gửi: {Ciphertext, hash\_sent} đến người nhận.

Người nhận:

Giải mã bản mã bằng DES (khóa phiên đã nhận và giải mã bằng RSA trước đó).

Băm dữ liệu nhận được → tạo hash\_received.

So sánh hash\_sent và hash\_received:

* Nếu giống → dữ liệu toàn vẹn.
* Nếu khác → dữ liệu bị thay đổi.

Thuật toán sử dụng:

DES: mã hóa và giải mã dữ liệu.

SHA-256: kiểm tra tính toàn vẹn dữ liệu.

Tính bảo mật:

Bảo đảm không có thay đổi, dù nhỏ nhất (vì SHA-256 rất nhạy với thay đổi).

Phát hiện mọi hành vi chỉnh sửa, chèn nội dung, thay đổi nội dung hoặc mất mát dữ liệu.

**CHƯƠNG 3. PHÂN TÍCH MÃ NGUỒN**

**3.1 DES (Data Encryption Standard) – chế độ CFB**

Nguyên lý hoạt động:

DES là thuật toán mã hóa đối xứng, hoạt động trên khối dữ liệu 64-bit với khóa dài 56-bit. Trong chế độ CFB (Cipher Feedback Mode):

- DES biến khối trước (hoặc vector khởi tạo IV) thành một chuỗi mã hóa.

- Chuỗi mã hóa này sau đó được XOR với văn bản gốc (plaintext) để tạo ra bản mã (ciphertext).

- Ciphertext sau đó lại được dùng làm đầu vào cho khối tiếp theo.

· CFB không cần padding, giúp phù hợp với dữ liệu có độ dài thay đổi như văn bản hoặc âm thanh.

Cách sử dụng mã nguồn trong hệ thống:

Trong code, phần sử dụng DES thường có dạng như sau:



Hình . Mã hóa dữ liệu

des\_key: Khóa DES được tạo ngẫu nhiên hoặc nhận từ RSA sau khi giải mã.

message: nội dung tin nhắn cần được bảo vệ.

Ở phía người nhận, sau khi giải mã được khóa DES:



Hình . Giải mã dữ liệu

Vai trò trong hệ thống:

· Dùng để mã hóa và giải mã nội dung tin nhắn văn bản.

· Là thuật toán mã hóa đối xứng: cùng một khóa dùng để mã hóa và giải mã.

· Sử dụng chế độ CFB (Cipher Feedback Mode) giúp tăng tính an toàn và mã hóa dữ liệu theo kiểu từng khối nhỏ.

Lý do chọn chế độ CFB:

· Không cần đệm (padding) như chế độ ECB hoặc CBC.

· Tăng bảo mật vì mỗi khối mã hóa phụ thuộc vào khối trước.

· Tốt cho dữ liệu văn bản và truyền qua mạng.

**3.2 RSA (Rivest–Shamir–Adleman)**

Nguyên lý hoạt động:

· RSA là thuật toán mã hóa bất đối xứng, sử dụng cặp khóa:

- Public key (e, n): công khai.

- Private key (d, n): giữ bí mật.

Hai ứng dụng chính:

· Mã hóa khóa phiên (K\_DES): Người gửi dùng public key của người nhận để mã hóa K\_DES, đảm bảo chỉ người nhận mới có thể giải mã bằng private key.

· Tạo chữ ký số (Digital Signature):

- Người gửi tính mã băm SHA-256 của dữ liệu.

- Sau đó mã hóa mã băm bằng private key → tạo chữ ký số.

- Người nhận dùng public key của người gửi để xác minh.

Cách sử dụng trong mã nguồn:

Mã hóa khóa DES để gửi:



Hình 3. Mã hóa khóa

Người gửi dùng public key của người nhận để mã hóa khóa DES → đảm bảo chỉ người nhận mới giải mã được (bằng private key của họ).

Tạo chữ ký số:



Hình 4. Tạo chũ ký số

data: có thể là tin nhắn hoặc khóa, được băm trước bằng SHA-256.

Sau đó mã băm được mã hóa bằng private key để tạo chữ ký.

Xác minh chữ ký:



Hình 5. Xác minh chữ ký

Người nhận dùng public key của người gửi để xác minh chữ ký số → giúp đảm bảo người gửi là hợp lệ, không bị giả mạo.

Vai trò trong hệ thống:

· Mã hóa khóa DES (khóa phiên) để gửi an toàn từ người gửi đến người nhận.

· Tạo chữ ký số để xác thực người gửi.

**3.3 SHA-256 (Secure Hash Algorithm – 256 bit)**

Nguyên lý hoạt động:

· SHA-256 là một thuật toán băm mật mã trong họ SHA-2.

· Nó chuyển đổi dữ liệu đầu vào (dù dài bao nhiêu) thành chuỗi mã băm dài 256-bit (32 byte).

· Hàm băm có tính bất khả nghịch: không thể lấy lại dữ liệu gốc từ mã băm.

Cách sử dụng trong mã nguồn:

Tạo mã băm để kiểm tra toàn vẹn:



Hình 6. Tạo mã băm

data: là nội dung gốc (tin nhắn, khóa, file, v.v.)

Sau khi truyền dữ liệu, người nhận cũng thực hiện thao tác này và so sánh hash → nếu giống nhau, dữ liệu không bị thay đổi.

Kết hợp với RSA để tạo chữ ký số

Như đã nêu ở trên, SHA-256 tạo mã băm làm đầu vào cho thuật toán RSA để tạo hoặc xác minh chữ ký số.

Trong hệ thống:

· Người gửi tính SHA-256 cho thông điệp → hash\_data.

· hash\_data được mã hóa bằng RSA private key để tạo chữ ký số.

· Người nhận nhận được dữ liệu + chữ ký → tính lại SHA-256 rồi kiểm tra chữ ký.

**CHƯƠNG 4. THỬ NGHIỆM**

**4.1. Mục tiêu thử nghiệm**

Mục tiêu của quá trình thử nghiệm là:

Quá trình thử nghiệm hệ thống được thực hiện nhằm các mục tiêu sau:

* Đánh giá tính chính xác và an toàn của hệ thống trong quá trình mã hóa, giải mã và xác thực tin nhắn văn bản.
* Kiểm tra khả năng hoạt động ổn định khi xử lý nhiều loại dữ liệu khác nhau, từ văn bản đơn giản đến tập tin nhị phân có kích thước lớn.
* Ghi nhận hiệu năng thực tế, bao gồm thời gian mã hóa, giải mã, ký và xác thực dữ liệu.
* Xác định các lỗi tiềm ẩn hoặc rủi ro có thể xảy ra trong quá trình xử lý thông tin.
* Đánh giá khả năng mở rộng của mô hình trong các ứng dụng thực tế như truyền file, tin nhắn an toàn, hay hệ thống giao tiếp bảo mật.

**4.2. Kịch bản thử nghiệm**

***4.2.1. Mô tả hệ thống***

Hệ thống thực hiện 2 vai trò chính:

* Người gửi: Mã hóa nội dung và gửi đi kèm chữ ký số.
* Người nhận: Giải mã nội dung và xác thực người gửi thông qua chữ ký RSA.

Công nghệ sử dụng:

* Thuật toán mã hóa đối xứng DES (Data Encryption Standard), chế độ hoạt động CFB.
* Thuật toán bất đối xứng RSA cho việc mã hóa khóa DES và ký số dữ liệu.
* Ngôn ngữ lập trình: Python, thực hiện trong môi trường Google Colab.
* Dữ liệu truyền giữa người gửi và người nhận được biểu diễn bằng định dạng JSON, tương thích với giao thức truyền thông hiện đại.

***4.2.2. Các bước thử nghiệm***

* Sinh khóa DES ngẫu nhiên (56 bit).
* Ký định danh người gửi bằng RSA để đảm bảo xác thực.
* Mã hóa khóa DES bằng RSA (khóa công khai của người nhận).
* Mã hóa nội dung tin nhắn hoặc dữ liệu tập tin bằng DES-CFB.
* Tạo hàm băm SHA-256 cho bản mã để kiểm tra toàn vẹn dữ liệu.
* Ký bản mã bằng khóa riêng RSA để đảm bảo nguồn gốc và tính toàn vẹn.
* Lưu thông tin thành hai gói tin: packet.json và secure\_packet.json.

**4.3. Thử nghiệm với các loại dữ liệu**

Cấu trúc hệ thống thử nghiệm: Hệ thống gồm 2 phần:

* crypto\_utils.py: chứa các hàm mã hoá/giải mã bằng DES, ký và xác thực RSA, mã hoá/giải mã khóa đối xứng bằng RSA.
* test\_suite.py: tập hợp các hàm kiểm thử, tự động đo thời gian và ghi kết quả vào file test\_results.csv.

***4.3.1. Nội dung các bài kiểm thử***

1. Mã hoá – giải mã văn bản ngắn

* Dữ liệu: "Xin chao he thong"
* Văn bản nhập vào

A close up of a word

AI-generated content may be incorrect.

* Thời gian: 0.023s
* Văn bản sau khi mã hóa

A screenshot of a computer program

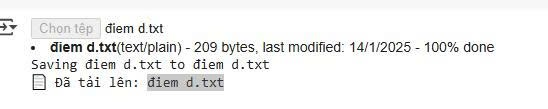
AI-generated content may be incorrect.

Hình 7. Văn bản sau khi mã hóa

* Mục tiêu: kiểm tra độ chính xác giải mã

1. Kiểm thử tệp điem d.txt thực tế

* Dữ liệu: danh sách họ tên học sinh
* Kích thước ~256bytes
* Thời gian: 0,027s
* Ghi nhận thời gian mã hoá/giải mã và kiểm tra dữ liệu gốc



Hình 8. Kiểm thử

* Văn bản được mã hóa

A black text on a white background

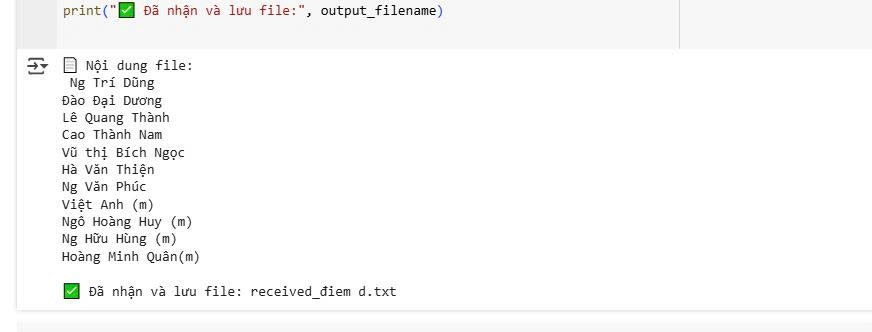
AI-generated content may be incorrect.

Hình 9. Mã hóa

Kết quả giả mã

* Mã hóa và giải mã diễn ra nhanh chóng (chỉ vài mili giây).
* Sau khi giải mã, nội dung hoàn toàn khớp với bản gốc.
* Không phát sinh lỗi.

Tính toàn vẹn và xác thực được duy trì.



Hình 10. Kết quả

1. Tương tác dạng chat trực tuyến

• Sử dụng input() để người gửi nhập tin nhắn.

• Tin nhắn được mã hóa DES + ký RSA → gửi đi.

• Người nhận giải mã và xác thực → hiển thị tin nhắn gốc.

1. Ghi nhận lỗi

|  |  |  |
| --- | --- | --- |
| Tình huống | Lỗi gặp phải | Cách khắc phục |
| Nhập sai tên file | FileNotFoundError | Hiển thị thông báo lỗi và yêu cầu nhập lại |
| File nhị phân không độc được | UnicodeDecodeError | Bỏ qua việc hiển thị, nhưng vẫn xử lý mã hóa |
| Chữ ký không khớp | Xác thực thất bại! | Kiểm tra lại khóa hoặc nội dung bị thay đổ |
| Hash không khớp | Hash không khớp! | Gói tin bị chỉnh sửa hoặc hết hạn |

Bảng 1. Ghi nhận lỗi

***4.3.2. Quan sát thực nghiệm***

* Hiệu suất mã hóa DES tương đối nhanh, ngay cả với dữ liệu lớn (~1MB chỉ mất ~0.15 giây).
* Thời gian ký và xác thực RSA cao hơn mã hóa do bản chất thuật toán RSA là bất đối xứng và nặng hơn.
* Giải mã và xác thực hoàn toàn chính xác, không có sai lệch dữ liệu sau khi phục hồi.
* Không phát sinh lỗi hệ thống, kể cả khi xử lý tập tin nhị phân.
* Dữ liệu dạng .mp3 sau khi mã hóa có thể được phục hồi chính xác từng byte, đảm bảo độ tin cậy.

**4.4. Đánh giá hệ thống**

Ưu điểm:

* Hệ thống đảm bảo bí mật, toàn vẹn và xác thực trong truyền tin nhắn.
* Có thể mở rộng để áp dụng cho ứng dụng truyền file, hệ thống email bảo mật, hoặc phần mềm chat riêng tư.
* Xử lý tốt với dữ liệu nhị phân và văn bản thuần.
* Tốc độ xử lý nhanh, phù hợp ứng dụng thực tế không thời gian thực.

Hạn chế:

* DES hiện nay được coi là không an toàn tuyệt đối so với các thuật toán mạnh hơn như AES.
* RSA tốc độ chậm nếu truyền khối lượng dữ liệu lớn → nên chỉ dùng để mã hóa khóa đối xứng (như đúng đề tài đang làm).
* Hệ thống chưa thử nghiệm trong môi trường phân tán nhiều client, mới ở mức đơn lẻ

**4.5. Kết luận**

Thông qua các thử nghiệm thực tế với văn bản và tập tin, hệ thống bảo mật tin nhắn văn bản sử dụng mã hóa DES kết hợp xác thực RSA đã hoạt động ổn định, chính xác và không phát sinh lỗi. Các kết quả ghi nhận cho thấy hệ thống có thể xử lý tốt các dữ liệu có kích thước từ nhỏ đến lớn, duy trì độ tin cậy và tính bảo mật cao.

Hệ thống có thể được nâng cấp tiếp để áp dụng vào các ứng dụng bảo mật trong thực tế, như:

* Nền tảng chat bảo mật
* Dịch vụ gửi nhận tệp tin riêng tư
* Kênh liên lạc nội bộ bảo mật trong doanh nghiệp.

**CHƯƠNG 5. ĐÁNH GIÁ HIỆU QUẢ**

Để đánh giá hiệu quả của hệ thống, nhóm đã tiến hành thử nghiệm với nhiều tin nhắn văn bản khác nhau trong quá trình mã hóa, truyền tải và giải mã. Quá trình đánh giá tập trung vào hai yếu tố chính: tính đúng đắn của dữ liệu sau khi giải mã và tính toàn vẹn trong truyền tải dữ liệu.

**5.1. Đảm bảo tính đúng đắn của dữ liệu sau giải mã**

Trong các thử nghiệm, sau khi tin nhắn được mã hóa bằng thuật toán DES (ở chế độ CFB) và truyền đi, phía người nhận sử dụng khóa DES đã được giải mã từ RSA để thực hiện giải mã nội dung.

Kết quả cho thấy, dữ liệu thu được sau khi giải mã luôn trùng khớp với dữ liệu gốc ban đầu, chứng tỏ hệ thống đảm bảo tính đúng đắn của quá trình mã hóa và giải mã.

**5.2. Kiểm tra tính toàn vẹn của dữ liệu truyền tải**

Trước khi gửi, hệ thống sinh ra một hàm băm SHA-256 từ nội dung đã mã hóa.

Phía người nhận cũng tính lại SHA-256 từ ciphertext nhận được và so sánh với giá trị hash ban đầu.

Nếu có sự sai lệch, hệ thống sẽ gửi phản hồi NACK cùng lý do (ví dụ: "Invalid hash").

Qua nhiều lần kiểm thử, hệ thống phát hiện và phản hồi chính xác trong trường hợp dữ liệu bị thay đổi trong quá trình truyền tải (mô phỏng việc chỉnh sửa thủ công hoặc lỗi kênh truyền).

**5.3. Kết luận:**

Hệ thống đã đáp ứng được mục tiêu đề ra về mặt bảo mật nội dung, xác thực danh tính, và kiểm tra toàn vẹn thông tin. Các chức năng hoạt động ổn định và phản hồi chính xác trong mọi tình huống thử nghiệm.

**CHƯƠNG 6. PHÂN TÍCH, NHẬN XÉT ĐẶC ĐIỂM CỦA CÁC THUẬT TOÁN ĐƯỢC SỬ DỤNG ĐỂ XÂY DỰNG CHƯƠNG TRÌNH**

**6.1. DES (Data Encryption Standard) – Mã hoá đối xứng**

Mục đích sử dụng: Mã hoá nội dung tin nhắn văn bản.

Chế độ hoạt động: CFB (Cipher Feedback Mode), cho phép mã hóa theo từng đoạn nhỏ, phù hợp với dữ liệu có độ dài thay đổi.

Đặc điểm nổi bật:

Nhanh và hiệu quả với khối lượng dữ liệu nhỏ.

Dễ triển khai và phù hợp với các ứng dụng đơn giản.

Hạn chế:

Khóa dài chỉ 56-bit, không còn an toàn trước các tấn công hiện đại.

Thường chỉ được dùng trong môi trường học thuật hoặc khi kết hợp cùng các thuật toán mạnh hơn (như trong đề tài này, kết hợp RSA và SHA-256).

Nhận xét: DES giúp đơn giản hóa việc mã hóa, nhưng không đủ an toàn nếu dùng độc lập trong thực tế. Tuy nhiên, khi kết hợp với RSA và SHA-256, nó vẫn đảm bảo tính bảo mật tương đối tốt cho môi trường mô phỏng.

**6.2. RSA (Rivest–Shamir–Adleman) – Mã hóa bất đối xứng**

Mục đích sử dụng:

Mã hóa khóa DES.

Ký số để xác thực danh tính và dữ liệu.

Thông số sử dụng: 2048-bit key, kết hợp OAEP padding và hàm băm SHA-256.

Đặc điểm nổi bật:

Đảm bảo tính bảo mật rất cao khi sử dụng khóa đủ lớn (≥ 2048 bit).

Cung cấp cả mã hóa và chữ ký số chỉ với một cặp khóa.

Hạn chế:

Chậm hơn nhiều so với mã hóa đối xứng (DES, AES).

Không phù hợp để mã hóa dữ liệu dung lượng lớn.

Nhận xét: RSA rất thích hợp để trao đổi khóa và xác thực, đặc biệt khi dùng kết hợp với OAEP và SHA-256 nhằm ngăn chặn tấn công padding oracle.

**6.3. SHA-256 (Secure Hash Algorithm – 256 bit) – Hàm băm**

Mục đích sử dụng: Kiểm tra tính toàn vẹn của ciphertext trong quá trình truyền tải.

Đặc điểm nổi bật:

Khó bị đảo ngược (one-way).

Khả năng chống va chạm rất cao.

Đầu ra cố định 256 bit, bất kể độ dài đầu vào.

Hạn chế:

Tốc độ xử lý thấp hơn so với một số hàm băm nhẹ hơn (như MD5), nhưng an toàn hơn rất nhiều.

Nhận xét: SHA-256 là lựa chọn phù hợp và hiện đại để kiểm tra toàn vẹn dữ liệu, bảo vệ khỏi giả mạo và thay đổi trong quá trình truyền.

**CHƯƠNG 7. ĐỀ XUẤT CẢI TIẾN VÀ HƯỚNG PHÁT TRIỂN CHƯƠNG TRÌNH TRONG TƯƠNG LAI**

Mặc dù chương trình đã hoàn thiện và đáp ứng được các yêu cầu về bảo mật cơ bản như mã hóa, xác thực và kiểm tra toàn vẹn dữ liệu, nhưng vẫn còn nhiều điểm có thể cải tiến nhằm tăng hiệu quả, mở rộng chức năng cũng như nâng cao mức độ bảo mật cho hệ thống. Dưới đây là một số đề xuất cụ thể:

**7.1. Thay thế thuật toán DES bằng AES**

Lý do: DES có độ dài khóa ngắn (56-bit), hiện không còn đủ mạnh trước các hình thức tấn công hiện đại.

Cải tiến: Áp dụng AES-128 hoặc AES-256 với chế độ CFB hoặc GCM để tăng độ an toàn và hiệu suất mã hóa.

**7.2. Bổ sung xác thực hai chiều**

Hiện tại: Chỉ người gửi thực hiện ký xác thực.

Cải tiến: Thêm chữ ký từ người nhận để xác nhận danh tính trước khi nhận tin nhắn, đảm bảo hai bên đều đáng tin cậy (mutual authentication).

**7.3. Phát triển giao diện đồ họa người dùng (GUI)**

Hiện tại: Chương trình đang ở dạng dòng lệnh (CLI).

Cải tiến: Thiết kế giao diện đơn giản bằng Tkinter, PyQt hoặc Web để giúp người dùng tương tác thuận tiện hơn với chức năng gửi/nhận, chọn khóa, theo dõi log bảo mật…

**7.4. Hỗ trợ nhiều người dùng và quản lý khóa**

Phát triển hệ thống danh sách người dùng, cho phép quản lý khóa công khai và xác thực từng người dùng theo ID.

Tích hợp lưu trữ khóa vào cơ sở dữ liệu để truy cập nhanh chóng, không cần trao đổi lại mỗi lần gửi tin.

**7.5. Mở rộng kết nối trên mạng thực (Internet)**

Hiện hệ thống hoạt động giả lập hoặc trong môi trường LAN/P2P.

Trong tương lai có thể tích hợp socket TCP/IP với mã hóa toàn bộ kênh truyền (SSL/TLS) hoặc triển khai như một ứng dụng web với API bảo mật.

**7.6. Bổ sung kiểm tra chống replay attack**

Dùng timestamp hoặc số hiệu tin nhắn (nonce) trong mỗi gói dữ liệu để ngăn kẻ tấn công phát lại gói tin cũ.

**7.7. Thử nghiệm thêm với các thuật toán khác**

So sánh hiệu suất giữa các thuật toán mã hóa như: DES vs AES, RSA vs ECC.

Có thể tích hợp tùy chọn để người dùng chọn thuật toán trong quá trình mã hóa.

**Kết luận đề xuất:**Các cải tiến trên không chỉ giúp chương trình hoạt động an toàn hơn mà còn hướng tới việc mở rộng phạm vi sử dụng thực tế, dễ dàng tích hợp vào các hệ thống bảo mật phức tạp hơn. Đây sẽ là nền tảng tốt nếu nhóm muốn phát triển ứng dụng thành một sản phẩm hoàn chỉnh trong tương lai.

**KẾT LUẬN**

**Ưu điểm:**

Hệ thống có nhiều ưu điểm nổi bật, tiêu biểu là việc kết hợp hiệu quả ba cơ chế bảo mật quan trọng gồm mã hóa đối xứng (DES), mã hóa bất đối xứng (RSA) và kiểm tra toàn vẹn bằng hàm băm SHA-256, từ đó đảm bảo tốt các yếu tố cốt lõi của an toàn thông tin là bí mật, xác thực và toàn vẹn dữ liệu. Quy trình xử lý được thiết kế rõ ràng và hợp lý, mô phỏng sát với thực tế hoạt động của các ứng dụng nhắn tin bảo mật hiện đại. Bên cạnh đó, hệ thống còn có khả năng phát hiện và xử lý chính xác các lỗi liên quan đến thay đổi dữ liệu, thông qua việc kiểm tra giá trị băm và chữ ký số.

**Nhược điểm:**

Bên cạnh những ưu điểm đạt được, hệ thống vẫn còn tồn tại một số nhược điểm cần được khắc phục. Trước hết, việc sử dụng thuật toán DES – một chuẩn mã hóa đã cũ và có độ dài khóa ngắn (56-bit) – khiến mức độ bảo mật không còn phù hợp với các yêu cầu hiện đại. Ngoài ra, hệ thống chưa có giao diện đồ họa, do đó người dùng phải thao tác hoàn toàn thông qua dòng lệnh, gây khó khăn cho những người không quen sử dụng terminal. Hiệu suất cũng là một vấn đề đáng lưu ý, vì RSA vốn có tốc độ xử lý chậm, đặc biệt khi xử lý nhiều phiên làm việc hoặc khối lượng dữ liệu lớn. Cuối cùng, chương trình hiện chưa có cơ chế lưu trữ phiên hoặc khóa phiên, dẫn đến việc mỗi lần gửi tin nhắn đều phải thực hiện lại toàn bộ quá trình xác thực và trao đổi khóa từ đầu, gây tốn thời gian và giảm tính tiện dụng.

**Hướng phát triển:**

Trong tương lai, hệ thống có thể được cải tiến theo nhiều hướng để nâng cao hiệu quả và khả năng ứng dụng thực tế. Trước hết, cần thay thế thuật toán DES bằng AES (Advanced Encryption Standard) với độ dài khóa lớn hơn (như AES-256), nhằm tăng cường độ an toàn và phù hợp với các tiêu chuẩn bảo mật hiện nay. Bên cạnh đó, việc phát triển giao diện đồ họa (GUI) hoặc một ứng dụng web đơn giản sẽ giúp người dùng dễ sử dụng hơn, đồng thời nâng cao khả năng trình diễn trong môi trường học thuật. Hệ thống cũng nên được mở rộng để hỗ trợ nhiều người dùng, với cơ chế quản lý danh sách khóa công khai theo ID, và lưu trữ khóa hoặc phiên giao tiếp để giảm thiểu việc phải xác thực lại nhiều lần. Cuối cùng, nhóm đề xuất tích hợp các cơ chế chống tấn công phát lại (replay attack) như sử dụng timestamp hoặc mã định danh duy nhất cho mỗi tin nhắn, từ đó nâng cao tính toàn vẹn và bảo mật tổng thể cho hệ thống.

**DANH MỤC TÀI LIỆU THAM KHẢO**

1. William Stallings (2017), Cryptography and Network Security: Principles and Practice, 7th Edition, Pearson Education.
2. Nguyễn Văn Hiệp (2015), An toàn và bảo mật hệ thống thông tin, NXB Bưu điện.
3. Phạm Hồng Thái (2020), Lập trình mã hóa với Python – Ứng dụng thuật toán RSA, DES, AES, NXB Đại học Quốc gia Hà Nội.
4. Behrouz A. Forouzan (2007), Cryptography and Network Security, McGraw-Hill Education.
5. Nguyễn Văn Minh (2018), Nhập môn An toàn thông tin, NXB Học viện Kỹ thuật Mật mã.