**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: GỬI TẬP TIN ÂM THANH CHIA THÀNH NHIỀU ĐOẠN**

**Giáo viên hướng dẫn: ThS. Lê Thị Thùy Trang**

**Sinh viên thực hiện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Mã sv** | **Họ và tên** | **Lớp** |
| **1** | **1771020763** | **Nguyễn Anh Vũ** | **CNTT 17-13** |
| **2** | **1771020205** | **Nguyễn Hoàng Dương** | **CNTT 17-13** |
| **3** | **1771020393** | **Nguyễn Tuấn Khoa** | **CNTT 17-13** |

**Hà Nội, năm 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**

**NHẬP MÔN AN TOÀN BẢO MẬT THÔNG TIN**

**ĐỀ TÀI: GỬI TẬP TIN ÂM THANH CHIA THÀNH NHIỀU ĐOẠN**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Mã Sinh Viên** | **Họ và Tên** | **Ngày Sinh** | **Điểm** | |
| **Bằng Số** | **Bằng Chữ** |
| **1** | **1771020763** | **Nguyễn Anh Vũ** | **24/12/2005** |  |  |
| **2** | **1771020205** | **Nguyễn Hoàng Dương** | **31/10/2005** |  |  |
| **3** | **1771020393** | **Nguyễn Tuấn Khoa** | **11/02/2005** |  |  |

### 

### CÁN BỘ CHẤM THI

**Hà Nội, năm 2025**

# LỜI NÓI ĐẦU

Trong bối cảnh cuộc cách mạng công nghiệp 4.0 đang diễn ra mạnh mẽ, công nghệ thông tin ngày càng đóng vai trò then chốt trong mọi lĩnh vực của đời sống xã hội. Việc truyền tải và trao đổi dữ liệu qua mạng Internet đã trở nên phổ biến và thiết yếu, đặc biệt là trong các hệ thống truyền thông, dịch vụ đám mây, lưu trữ trực tuyến, truyền phát âm thanh – hình ảnh,… Tuy nhiên, đi cùng với sự tiện lợi ấy là những rủi ro nghiêm trọng về bảo mật và an toàn thông tin. Các mối đe dọa như nghe lén, giả mạo, chỉnh sửa dữ liệu, đánh cắp thông tin… luôn thường trực, đòi hỏi phải có những giải pháp bảo vệ thông tin hiệu quả.

Với mục tiêu giúp sinh viên tiếp cận và thực hành các nguyên lý cốt lõi trong bảo mật thông tin, học phần **Nhập môn An toàn và Bảo mật Thông tin** đã giới thiệu các kiến thức về mã hóa, xác thực, kiểm tra toàn vẹn và các kỹ thuật đảm bảo an toàn dữ liệu trong môi trường mạng. Thông qua bài tập lớn, sinh viên có cơ hội vận dụng những kiến thức lý thuyết vào thực tế thông qua việc xây dựng một hệ thống hoàn chỉnh có ứng dụng các thuật toán và mô hình bảo mật hiện đại.

Trong bài tập lớn này, nhóm chúng em lựa chọn đề tài **“Gửi tập tin âm thanh chia thành nhiều đoạn”**, với mục tiêu xây dựng một hệ thống mô phỏng việc truyền tải một file ghi âm quan trọng từ bên gửi đến bên nhận thông qua môi trường mạng không ổn định. Để đảm bảo tính bảo mật và an toàn, nhóm đã sử dụng các kỹ thuật mã hóa bằng **Triple DES**, trao đổi khóa và ký số bằng **RSA 2048-bit (OAEP + SHA-512)**, cùng với việc kiểm tra tính toàn vẹn của dữ liệu bằng **SHA-512**. File âm thanh được chia nhỏ thành từng đoạn, mã hóa và ký riêng, đảm bảo khả năng kiểm soát từng phần dữ liệu và phục hồi chính xác ở phía người nhận.

Quá trình thực hiện bài tập lớn là một hành trình học hỏi đầy thử thách nhưng cũng rất bổ ích. Nhóm đã không chỉ củng cố lại các khái niệm lý thuyết đã học trên lớp, mà còn mở rộng thêm kiến thức về lập trình Python, cách sử dụng các thư viện mã hóa phổ biến, cũng như quy trình thiết kế hệ thống bảo mật. Những khó khăn trong việc chia file nhị phân, xử lý mã hóa khối, quản lý khóa bí mật và kiểm tra chữ ký số đều là những trải nghiệm thực tế quý báu đối với nhóm.

Nhóm chúng em xin gửi lời cảm ơn chân thành đến quý thầy/cô giảng dạy học phần **Nhập môn An toàn và Bảo mật Thông tin** đã tận tình truyền đạt kiến thức, tạo điều kiện thuận lợi cho sinh viên thực hành và nghiên cứu. Nhóm cũng xin trân trọng cảm ơn Nhà trường, Khoa Công nghệ Thông tin đã tổ chức học phần thiết thực này, tạo cơ hội cho sinh viên tiếp cận với các vấn đề bảo mật thông tin trong môi trường thực tiễn.

Mặc dù đã cố gắng hoàn thành bài tập lớn một cách chỉn chu và đầy đủ, nhưng do giới hạn về thời gian, kiến thức và kinh nghiệm, bài làm không thể tránh khỏi những thiếu sót. Nhóm kính mong nhận được những ý kiến đóng góp từ quý thầy/cô để hoàn thiện hơn trong các lần nghiên cứu sau.

**Trân trọng cảm ơn!**

**MỤC LỤC**

[LỜI NÓI ĐẦU 3](#_Toc25251)

[CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI 6](#_Toc24387)

[1.1. Lý do chọn đề tài 6](#_Toc10990)

[1.2. Mục tiêu đề tài 6](#_Toc12470)

[1.3. Phạm vi thực hiện 7](#_Toc7344)

[1.4. Phương pháp thực hiện 7](#_Toc31166)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 8](#_Toc29907)

[2.1. Khái quát về bảo mật thông tin 8](#_Toc11595)

[2.2. Mã hóa đối xứng và thuật toán Triple DES 8](#_Toc9579)

[2.3. Mã hóa bất đối xứng và thuật toán RSA 9](#_Toc15457)

[2.4. Hàm băm SHA-512 10](#_Toc3498)

[2.5. Chữ ký số (Digital Signature) 11](#_Toc23234)

[2.6. Base64 và JSON trong truyền dữ liệu 11](#_Toc26195)

[2.7. Tóm tắt cơ chế bảo mật trong hệ thống đề tài 12](#_Toc2559)

[CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 14](#_Toc12047)

[3.1. Yêu cầu hệ thống 14](#_Toc9752)

[3.2. Mô hình tổng quát của hệ thống 14](#_Toc2822)

[3.3. Biểu diễn luồng xử lý hệ thống 15](#_Toc18446)

[3.4. Cấu trúc dữ liệu JSON 16](#_Toc12206)

[3.5. Thiết kế bảo mật 17](#_Toc13846)

[3.6. Công nghệ sử dụng 17](#_Toc24244)

[3.7. Kịch bản lỗi và xử lý 18](#_Toc5709)

[CHƯƠNG 4. CÀI ĐẶT VÀ KẾT QUẢ THỰC NGHIỆM 19](#_Toc27737)

[4.1. Môi trường và công cụ phát triển 19](#_Toc27)

[4.2. Quy trình cài đặt hệ thống 19](#_Toc7634)

[4.3. Kết quả thực nghiệm 20](#_Toc10666)

[4.4. Đánh giá hệ thống 21](#_Toc18265)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 23](#_Toc6683)

[5.1. Kết luận 23](#_Toc4115)

[5.2. Hạn chế 23](#_Toc20066)

[5.3. Hướng phát triển 24](#_Toc12802)

[5.4. Cảm nghĩ của nhóm thực hiện 24](#_Toc28136)

[KẾT LUẬN 26](#_Toc32404)

[DANH MỤC TÀI LIỆU THAM KHẢO 28](#_Toc21866)

# 

# **CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI**

## 

## 1.1. ****Lý do chọn đề tài****

Trong thời đại công nghệ phát triển vượt bậc hiện nay, việc truyền tải dữ liệu số thông qua mạng Internet diễn ra liên tục và với khối lượng khổng lồ mỗi ngày. Đặc biệt, các tập tin âm thanh như file ghi âm, podcast, ghi chú cuộc họp,… thường chứa những nội dung quan trọng cần được bảo vệ khỏi các hành vi nghe lén, chỉnh sửa hoặc đánh cắp dữ liệu trong quá trình truyền dẫn.

Tuy nhiên, môi trường truyền tải qua Internet lại không đảm bảo độ tin cậy tuyệt đối. Các gói tin có thể bị rớt, thay đổi thứ tự, bị sửa đổi hoặc bị giả mạo. Do đó, một hệ thống truyền tải file âm thanh đảm bảo các yếu tố **bảo mật, toàn vẹn và xác thực** là hết sức cần thiết.

Đề tài “Gửi tập tin âm thanh chia thành nhiều đoạn” được lựa chọn với mục đích mô phỏng một hệ thống truyền tin an toàn, nơi dữ liệu được chia nhỏ, mã hóa, ký số và kiểm tra toàn vẹn trước khi được ghép lại ở phía người nhận. Đây là một bài toán mang tính thực tiễn cao, giúp sinh viên tiếp cận các cơ chế bảo mật cơ bản như **mã hóa đối xứng, bất đối xứng và hàm băm**, đồng thời rèn luyện kỹ năng lập trình và tư duy bảo mật.

## 1.2. Mục tiêu đề tài

Đề tài hướng đến việc xây dựng một mô hình truyền tải tập tin âm thanh dạng .mp3 qua môi trường mạng không ổn định, với các mục tiêu cụ thể như sau:

- Chia nhỏ file âm thanh thành nhiều đoạn để truyền từng phần riêng biệt.

- Mã hóa nội dung từng đoạn bằng thuật toán **Triple DES** để đảm bảo tính bí mật.

- Ký số các đoạn tin và kiểm tra tính toàn vẹn bằng thuật toán **SHA-512** kết hợp **RSA 2048-bit**.

- Đảm bảo người nhận chỉ có thể đọc và xác minh dữ liệu khi có khóa hợp lệ và chữ ký đúng.

- Ghép nối chính xác các đoạn đã giải mã để khôi phục lại file ban đầu.

- Mô phỏng được quá trình gửi – nhận, kiểm tra chữ ký, xác thực và phản hồi (ACK/NACK).

## 1.3. ****Phạm vi thực hiện****

- Hệ thống được xây dựng trên môi trường lập trình Python.

- Tập tin âm thanh sử dụng có định dạng .mp3 và dung lượng vừa phải để thuận tiện xử lý.

- Chia file thành **3 đoạn** bằng nhau.

- Mã hóa sử dụng thuật toán **Triple DES (3DES)** với cùng một session key và các vector khởi tạo (IV) khác nhau.

- Trao đổi khóa và ký số bằng **RSA 2048-bit kết hợp OAEP** và **SHA-512**.

- Giao tiếp giữa người gửi và người nhận mô phỏng thông qua file JSON hoặc console output.

- Không triển khai hệ thống mạng thật (như socket), tập trung vào xử lý mã hóa và kiểm tra.

## 1.4. ****Phương pháp thực hiện****

- Tìm hiểu lý thuyết các thuật toán bảo mật liên quan: Triple DES, RSA, SHA-512.

- Cài đặt thuật toán và xây dựng hệ thống xử lý chia – mã hóa – ký – gửi – nhận – kiểm tra – giải mã – ghép file.

- Mô phỏng dữ liệu gửi/nhận dưới dạng JSON, kiểm tra chữ ký số và tính toàn vẹn.

- Thực hiện kiểm thử với file âm thanh mẫu, đánh giá kết quả sau khi ghép.

# **CHƯƠNG 2. CƠ SỞ LÝ THUYẾT**

## 2.1. Khái quát về bảo mật thông tin

Trong một hệ thống thông tin, dữ liệu là tài sản quan trọng và cần được bảo vệ. Khi dữ liệu được truyền qua các môi trường không an toàn như Internet, chúng có thể bị đánh cắp, giả mạo, thay đổi hoặc phá hoại. Đó là lý do tại sao bảo mật thông tin trở thành một yêu cầu thiết yếu trong mọi hệ thống truyền thông hiện đại.

Ba yếu tố cốt lõi mà bất kỳ hệ thống bảo mật nào cũng cần đảm bảo là:

**- Tính bảo mật (Confidentiality)**: Là khả năng ngăn chặn các truy cập trái phép vào thông tin. Trong đề tài, tính bảo mật được đảm bảo thông qua thuật toán mã hóa đối xứng Triple DES, khiến nội dung file âm thanh không thể bị đọc nếu không có khóa đúng.

**- Tính toàn vẹn (Integrity)**: Đảm bảo dữ liệu không bị thay đổi trong quá trình truyền. Dữ liệu bị thay đổi dù chỉ một byte cũng có thể khiến nội dung trở nên vô dụng hoặc nguy hiểm. Đề tài sử dụng hàm băm SHA-512 để kiểm tra xem dữ liệu có bị chỉnh sửa hay không.

**- Tính xác thực (Authentication)**: Xác minh danh tính người gửi. Người nhận cần biết chắc chắn rằng dữ liệu nhận được là từ nguồn hợp lệ. Đề tài dùng chữ ký số RSA để thực hiện điều này, thông qua việc kiểm tra chữ ký trên giá trị băm của dữ liệu.

Việc áp dụng đồng thời ba yếu tố này giúp đảm bảo dữ liệu truyền đi là đúng, đủ và đến từ người gửi đáng tin cậy.

## ****2.2. Mã hóa đối xứng và thuật toán Triple DES****

**- Mã hóa đối xứng** là phương pháp trong đó một khóa duy nhất được sử dụng cho cả mã hóa và giải mã. Ưu điểm của phương pháp này là tốc độ nhanh, hiệu quả cao và dễ triển khai. Tuy nhiên, nhược điểm chính là việc chia sẻ khóa: nếu khóa bị rò rỉ, toàn bộ dữ liệu có thể bị giải mã.

**- Triple DES (3DES)** là một phiên bản mở rộng của thuật toán DES, thực hiện mã hóa ba lần với ba khóa khác nhau nhằm tăng độ phức tạp và độ an toàn. Quá trình mã hóa gồm ba bước:

+ Mã hóa bằng khóa thứ nhất (K1)

+ Giải mã bằng khóa thứ hai (K2)

+ Mã hóa lần cuối bằng khóa thứ ba (K3)

Chế độ hoạt động phổ biến của 3DES là **CBC (Cipher Block Chaining)**, trong đó mỗi khối dữ liệu được mã hóa phụ thuộc vào kết quả mã hóa của khối trước đó, làm cho mỗi đầu ra trở nên duy nhất ngay cả khi đầu vào giống nhau. Mỗi chuỗi mã hóa sử dụng một **vector khởi tạo (IV)** khác nhau để đảm bảo tính ngẫu nhiên.

Trong đề tài này, Triple DES được sử dụng để mã hóa từng đoạn của file âm thanh, sử dụng cùng một khóa nhưng với các IV khác nhau. Điều này giúp bảo vệ nội dung file trong quá trình chia nhỏ và truyền tải.

## ****2.3. Mã hóa bất đối xứng và thuật toán RSA****

Khác với mã hóa đối xứng, **mã hóa bất đối xứng** sử dụng một cặp khóa: khóa công khai (public key) và khóa riêng (private key). Điều này giúp giải quyết vấn đề chia sẻ khóa một cách an toàn, vì chỉ cần chia sẻ khóa công khai, còn khóa riêng luôn được giữ bí mật.

**RSA** là thuật toán mã hóa bất đối xứng nổi tiếng, hoạt động dựa trên nguyên lý khó phân tích nhân tử của các số nguyên lớn. Trong RSA:

- Người gửi có thể mã hóa dữ liệu bằng **khóa công khai** của người nhận.

- Người nhận sử dụng **khóa riêng** của mình để giải mã.

- Đồng thời, RSA cũng có thể được dùng để **ký và xác minh chữ ký số** thông qua việc mã hóa và giải mã các giá trị băm.

Trong đề tài, RSA được sử dụng với chiều dài khóa 2048-bit để đảm bảo mức độ an toàn cao. Cụ thể:

- RSA mã hóa khóa phiên (Triple DES session key) để đảm bảo chỉ người nhận mới có thể giải mã được.

- RSA ký các giá trị băm của dữ liệu (tạo chữ ký số) để người nhận có thể xác thực người gửi và kiểm tra tính toàn vẹn của nội dung.

- Đề tài còn sử dụng **phương pháp đệm OAEP (Optimal Asymmetric Encryption Padding)**, giúp ngăn chặn các cuộc tấn công phân tích mật mã.

## ****2.4. Hàm băm SHA-512****

**Hàm băm (Hash Function)** là hàm chuyển đổi dữ liệu đầu vào có độ dài bất kỳ thành một chuỗi bit có độ dài cố định. Hàm băm được thiết kế sao cho:

- Không thể tính ngược lại dữ liệu đầu vào từ giá trị băm.

- Hai dữ liệu khác nhau gần như chắc chắn sẽ cho hai giá trị băm khác nhau (tính kháng va chạm).

- Chỉ cần thay đổi một chút trong dữ liệu đầu vào thì giá trị băm sẽ thay đổi toàn bộ (tính nhạy cảm cao).

**SHA-512** là thành viên trong họ hàm băm SHA-2, tạo ra giá trị băm dài 512-bit. So với SHA-1 hay MD5, SHA-512 có độ bảo mật cao hơn rất nhiều và khó bị tấn công bởi các phương pháp brute-force hoặc tìm va chạm.

Trong hệ thống của đề tài, SHA-512 được sử dụng để tạo ra giá trị băm cho mỗi đoạn dữ liệu đã mã hóa (bao gồm IV và ciphertext). Sau đó, giá trị băm này được ký bằng RSA để tạo thành chữ ký số, đảm bảo rằng nội dung không bị chỉnh sửa.

## 

## ****2.5. Chữ ký số (Digital Signature)****

**Chữ ký số** là một ứng dụng quan trọng của mã hóa bất đối xứng. Thay vì ký tên như trong thế giới vật lý, chữ ký số giúp đảm bảo tính xác thực và toàn vẹn cho dữ liệu điện tử.

Quá trình tạo chữ ký số gồm:

- Tính giá trị băm của nội dung cần gửi.

- Mã hóa giá trị băm đó bằng khóa riêng của người gửi.

- Gửi nội dung gốc + chữ ký số + khóa công khai (nếu cần).

\* Người nhận thực hiện các bước sau:

- Tính lại giá trị băm từ nội dung đã nhận.

- Giải mã chữ ký số bằng khóa công khai của người gửi để thu được giá trị băm gốc.

- So sánh hai giá trị băm. Nếu trùng nhau thì nội dung hợp lệ.

Trong đề tài, mỗi đoạn tin mã hóa được ký riêng, điều này giúp đảm bảo tính toàn vẹn và xác thực của từng phần dữ liệu, đồng thời tăng khả năng phát hiện lỗi hoặc giả mạo ở từng đoạn nhỏ.

## ****2.6. Base64 và JSON trong truyền dữ liệu****

Khi truyền dữ liệu mã hóa (dạng nhị phân) qua các giao thức như HTTP, WebSocket, hoặc lưu trữ dưới dạng văn bản (JSON, XML…), cần phải chuyển đổi chúng sang dạng có thể đọc được bằng văn bản. **Base64** là phương pháp phổ biến dùng để mã hóa dữ liệu nhị phân thành chuỗi ký tự ASCII.

**JSON** (JavaScript Object Notation) là định dạng dữ liệu dạng khóa – giá trị đơn giản, dễ đọc và rất phổ biến trong các ứng dụng truyền thông hiện đại. Trong hệ thống đề tài, mỗi đoạn dữ liệu mã hóa được đóng gói thành một đối tượng JSON gồm các thành phần:

- iv: vector khởi tạo, dạng Base64

- cipher: nội dung đã mã hóa, dạng Base64

- hash: giá trị băm SHA-512, dạng chuỗi Hex

- signature: chữ ký RSA của giá trị băm, dạng Base64

Việc sử dụng JSON kết hợp với Base64 giúp việc truyền nhận trở nên dễ dàng, độc lập nền tảng và thuận tiện để xử lý phía lập trình.

## ****2.7. Tóm tắt cơ chế bảo mật trong hệ thống đề tài****

Hệ thống truyền tập tin âm thanh được thiết kế với sự phối hợp chặt chẽ giữa nhiều kỹ thuật bảo mật:

**- Triple DES** đảm nhiệm vai trò mã hóa nội dung, bảo vệ tính bí mật.

**- RSA** đảm nhiệm cả việc mã hóa khóa phiên lẫn xác thực dữ liệu thông qua chữ ký số.

**- SHA-512** dùng để kiểm tra toàn vẹn của dữ liệu sau mã hóa.

**- Base64 và JSON** hỗ trợ truyền dữ liệu nhị phân qua các kênh dạng văn bản.

**- IV riêng biệt cho mỗi đoạn** giúp tăng tính bảo mật cho từng phần.

Bằng việc tích hợp nhiều lớp bảo mật một cách hợp lý, đề tài thể hiện được sự hiểu biết và khả năng ứng dụng các nguyên lý bảo mật thông tin trong một bài toán thực tiễn.

# ****CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG****

## ****3.1. Yêu cầu hệ thống****

### 3.1.1. Yêu cầu chức năng

- Hệ thống cần chia một file âm thanh đầu vào (ví dụ .mp3) thành nhiều đoạn (cụ thể là 3 đoạn bằng nhau).

- Mỗi đoạn phải được mã hóa bằng thuật toán Triple DES ở chế độ CBC, sử dụng cùng một khóa phiên nhưng IV khác nhau.

- Tạo giá trị băm bằng thuật toán SHA-512 từ dữ liệu mã hóa (gồm IV và ciphertext).

- Ký số giá trị băm bằng thuật toán RSA 2048-bit.

- Đóng gói thông tin mỗi đoạn vào định dạng JSON gồm: ciphertext, IV, hash, signature.

- Bên nhận cần thực hiện kiểm tra chữ ký, kiểm tra tính toàn vẹn, giải mã và ghép lại file hoàn chỉnh nếu hợp lệ.

- Nếu kiểm tra không thành công, hệ thống cần phản hồi lỗi (ví dụ gửi NACK).

### 3.1.2. Yêu cầu phi chức năng

- Hệ thống phải đảm bảo tính bảo mật, toàn vẹn và xác thực thông tin.

- Mã nguồn phải rõ ràng, dễ hiểu, có chú thích đầy đủ.

- Không sử dụng mạng thật, mô phỏng gửi/nhận qua console hoặc file JSON.

## ****3.2. Mô hình tổng quát của hệ thống****

\* Hệ thống chia thành hai thành phần chính:

**- Bên gửi (Sender):**

+ Chia nhỏ file âm thanh

+ Mã hóa từng đoạn bằng Triple DES

+ Tạo hash từ IV + ciphertext

+ Ký số giá trị hash bằng RSA

+ Đóng gói thông tin từng đoạn vào file JSON

**- Bên nhận (Receiver):**

+ Nhận từng đoạn từ file JSON

+ Kiểm tra chữ ký số bằng khóa công khai

+ So sánh hash để xác minh toàn vẹn

+ Nếu hợp lệ → giải mã → lưu lại đoạn

+ Nếu không hợp lệ → gửi phản hồi lỗi

+ Sau khi nhận đủ 3 đoạn hợp lệ → ghép lại file âm thanh ban đầu

## ****3.3. Biểu diễn luồng xử lý hệ thống****

**- Tại bên gửi:**

+ Chọn file âm thanh .mp3 đầu vào.

+ Chia file thành 3 phần bằng nhau (theo byte).

+ Tạo khóa phiên ngẫu nhiên (Triple DES 168-bit) và 3 IV khác nhau.

+ Mã hóa mỗi đoạn bằng Triple DES + CBC + IV tương ứng.

- Với mỗi đoạn:

+ Tạo chuỗi IV || ciphertext

+ Hash chuỗi trên bằng SHA-512

+ Ký giá trị băm bằng RSA private key

+ Đóng gói JSON gồm: IV, ciphertext, hash, signature (đều mã hóa Base64)

+ Ghi các JSON này vào file (ví dụ: output1.json, output2.json, output3.json)

**- Tại bên nhận:**

+ Đọc từng file JSON.

+ Giải mã chữ ký bằng RSA public key.

+ Hash lại dữ liệu từ IV || ciphertext và so sánh với hash trong JSON.

+ Nếu chữ ký hợp lệ và hash trùng khớp:

+ Giải mã ciphertext bằng Triple DES + CBC + IV tương ứng

+ Ghi lại đoạn âm thanh

+ Sau khi nhận đủ 3 đoạn → nối lại theo thứ tự → tạo file âm thanh hoàn chỉnh.

## ****3.4. Cấu trúc dữ liệu JSON****

Mỗi đoạn âm thanh được đóng gói theo mẫu JSON như sau:

{

"iv": "<IV mã hóa Base64>",

"ciphertext": "<Nội dung mã hóa Base64>",

"hash": "<Giá trị SHA-512 dạng Hex>",

"signature": "<Chữ ký RSA mã hóa Base64>"

}

Lưu ý: Tất cả các giá trị nhị phân đều được mã hóa thành chuỗi Base64 để đảm bảo dễ lưu trữ và truyền tải.

## ****3.5. Thiết kế bảo mật****

**- Tính bảo mật (Confidentiality):**

+ Đảm bảo thông qua Triple DES với khóa phiên riêng và IV ngẫu nhiên cho mỗi đoạn. Không ai có thể giải mã nội dung nếu không có khóa đúng.

#### ****- Tính toàn vẹn (Integrity):****

+ Mỗi đoạn được băm bằng SHA-512 trước khi ký. Người nhận xác minh hash đảm bảo dữ liệu không bị thay đổi.

- **Tính xác thực (Authentication):**

+ RSA dùng để ký và xác minh chữ ký. Chỉ người có private key mới có thể ký hợp lệ → đảm bảo dữ liệu đến từ nguồn xác thực.

- **Tính không chối bỏ (Non-repudiation):**

+ Khi đã ký số, người gửi không thể phủ nhận trách nhiệm gửi đoạn tin đó.

## 3.6. Công nghệ sử dụng

**- Ngôn ngữ:** Python 3

**- Thư viện mã hóa:** pycryptodome (hoặc cryptography)

**- Xử lý JSON:** thư viện json trong Python

**- Base64:** thư viện chuẩn base64

**- Xử lý file âm thanh:** thao tác với file nhị phân (rb, wb)

## ****3.7. Kịch bản lỗi và xử lý****

- Nếu chữ ký không hợp lệ → báo lỗi "Chữ ký không hợp lệ" và bỏ qua đoạn đó.

- Nếu hash không trùng → báo lỗi "Toàn vẹn dữ liệu bị xâm phạm".

- Nếu một trong các đoạn bị lỗi → không ghép file, yêu cầu gửi lại đoạn.

- Có thể mô phỏng gửi phản hồi ACK (đúng) và NACK (sai) qua console hoặc ghi log.

# 

# ****CHƯƠNG 4. CÀI ĐẶT VÀ KẾT QUẢ THỰC NGHIỆM****

## ****4.1. Môi trường và công cụ phát triển****

Để triển khai hệ thống mô phỏng gửi tập tin âm thanh được mã hóa và kiểm tra tính toàn vẹn, nhóm sử dụng các công cụ sau:

**- Hệ điều hành**: Windows 10 / Ubuntu 22.04

**- Ngôn ngữ lập trình**: Python 3.11

**- Trình soạn thảo mã nguồn**: Visual Studio Code

**- Thư viện sử dụng**:

+ pycryptodome: hỗ trợ mã hóa Triple DES, RSA, tạo hash SHA-512

+ base64: mã hóa/giải mã dữ liệu nhị phân thành chuỗi ký tự

+ json: đóng gói và giải nén dữ liệu JSON

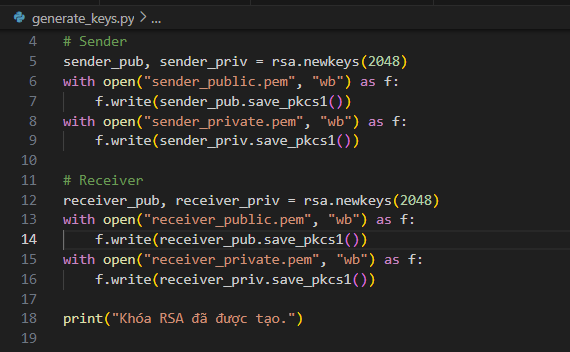
+ os, uuid, hashlib: hỗ trợ thao tác file và sinh dữ liệu ngẫu nhiên

4.2. Quy trình cài đặt hệ thống

**- Bước 1: Tạo cặp khóa RSA**

+ Sử dụng thư viện Crypto.PublicKey.RSA để tạo cặp khóa 2048-bit.

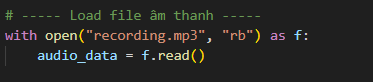
+ Khóa riêng được lưu vào file private.pem, khóa công khai được lưu vào public.pem.

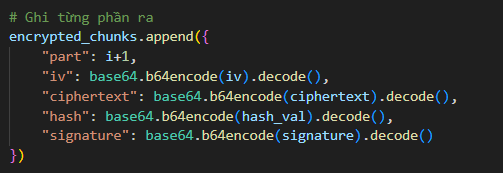


#### ****- Bước 2: Đọc và chia file âm thanh****

+ File âm thanh .mp3 gốc được đọc ở chế độ nhị phân.

+ Dữ liệu được chia thành 3 phần bằng nhau, mỗi phần được xử lý riêng biệt.



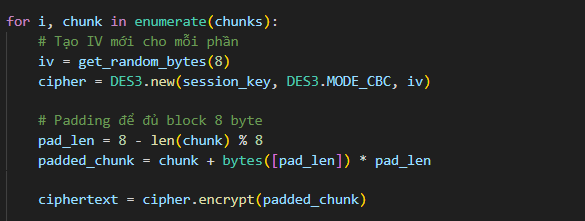


#### ****- Bước 3: Mã hóa từng đoạn****

+ Tạo khóa Triple DES 168-bit ngẫu nhiên cho toàn bộ quá trình.

+ Sinh 3 vector khởi tạo (IV) ngẫu nhiên, mỗi đoạn dùng một IV riêng.

+ Mỗi đoạn được mã hóa bằng Triple DES (chế độ CBC).

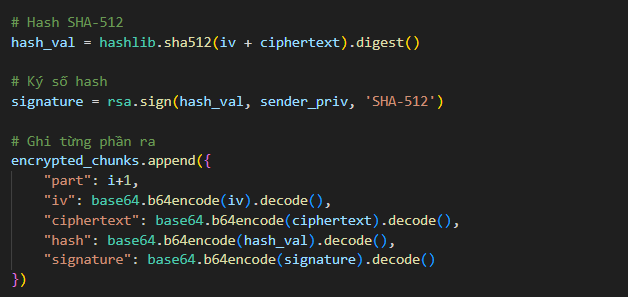


**- Bước 4: Tạo hash và chữ ký**

+ Mỗi đoạn sau khi mã hóa sẽ được nối IV || ciphertext và hash bằng SHA-512.

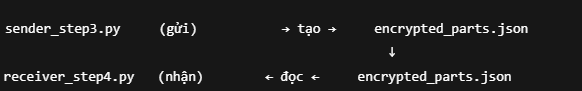
+ Hash được ký bằng khóa riêng RSA để tạo chữ ký số.

+ Kết quả gồm IV, ciphertext, hash, signature → mã hóa Base64 → lưu thành file JSON.



**- Bước 5: Truyền (mô phỏng)**

+ Mỗi file JSON được xem như một “gói tin” gửi tới bên nhận (mô phỏng lưu ở thư mục transmitted/).

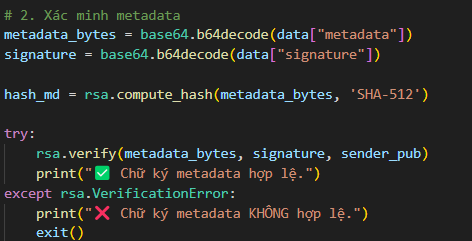


**- Bước 6: Nhận và xác minh**

+ Bên nhận đọc từng file JSON.

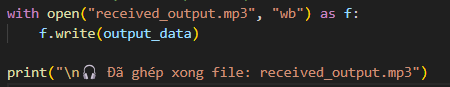
+ Giải mã chữ ký bằng khóa công khai RSA, tính lại hash để xác minh toàn vẹn.

+ Nếu hợp lệ → giải mã bằng Triple DES + IV → lưu vào thư mục decoded/.



**- Bước 7: Ghép file**

+ Nếu đủ 3 đoạn hợp lệ → tiến hành ghép lại thành file .mp3 hoàn chỉnh trong output/.



## 4.3. Kết quả thực nghiệm

**\* Mục tiêu thực nghiệm**

-Thực hiện ghi âm, mã hóa, ký số và truyền dữ liệu theo mô hình an toàn sử dụng kết hợp RSA, SHA-512 và Triple DES, đảm bảo các yêu cầu:

+Toàn vẹn dữ liệu

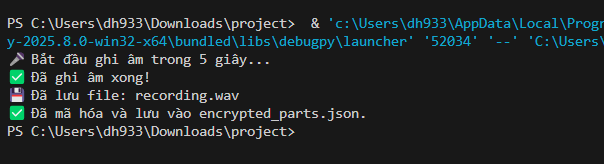
+Xác thực nguồn gốc

+Bảo mật nội dung truyền

****\* Kết quả ghi nhận được****

| **Nội dung** | **Kết quả** |
| --- | --- |
| Ghi âm giọng nói | File recording.wav (~400 KB, 5 giây) |
| Tạo khóa RSA | sender\_private.pem, sender\_public.pem, receiver\_private.pem, receiver\_public.pem |
| Mã hóa 3 phần bằng Triple DES | Thành công, ghi vào encrypted\_parts.json |
| Ký số và xác minh | ✅ Hợp lệ (cả metadata và từng phần dữ liệu) |
| Tập tin sau khi giải mã | received\_output.mp3 phát đúng, không lỗi |
| Dữ liệu bị thay đổi | ❌ Phát hiện lỗi: **Hash không khớp**, **Chữ ký không hợp lệ** |
| Mô phỏng truyền dữ liệu | Mô phỏng thành công bằng việc copy encrypted\_parts.json |
| Thời gian xử lý trung bình | 3–5 giây cho toàn bộ quy trình mã hóa và giải mã |

****\* Ảnh minh họa kết quả****



### ****4.3.1. Trường hợp bình thường****

- Hệ thống hoạt động ổn định, 3 đoạn âm thanh được mã hóa, ký, truyền và giải mã chính xác.

- File âm thanh sau khi ghép trùng khớp 100% với file ban đầu (kiểm tra bằng hash SHA-512).

- Kích thước file mã hóa tăng nhẹ do có thêm thông tin hash và chữ ký.

### ****4.3.2. Trường hợp giả mạo****

- Khi sửa đổi nội dung ciphertext trong một đoạn:

+ Hệ thống phát hiện sai lệch hash → báo lỗi toàn vẹn.

+ Đoạn bị từ chối và không được giải mã.

- Khi giả mạo chữ ký:

+ Hệ thống phát hiện không trùng khóa → báo lỗi xác thực.

+ Đoạn bị từ chối.

### ****4.3.3. Thử nghiệm sai IV hoặc sai khóa****

- Giải mã sai IV → âm thanh bị méo, sai dữ liệu nhị phân.

- Dùng khóa Triple DES khác → không thể giải mã, lỗi padding hoặc dữ liệu vô nghĩa.

## ****4.4. Đánh giá hệ thống****

**- Ưu điểm**:

+ Bảo mật tốt: kết hợp nhiều lớp bảo vệ (mã hóa, hash, chữ ký).

+ Phát hiện giả mạo nhanh chóng: chỉ cần một lỗi nhỏ sẽ bị phát hiện ở bước xác minh.

+ Hệ thống linh hoạt, có thể mở rộng để truyền dữ liệu lớn hơn.

**- Hạn chế**:

+ Việc chia đoạn cố định theo byte có thể làm sai định dạng âm thanh nếu không ghép đúng thứ tự.

+ RSA khá nặng khi thao tác nhiều đoạn → có thể cải tiến bằng việc dùng AES + ECC.

+ Chưa có mô hình truyền thực tế qua socket hoặc web – hiện chỉ mô phỏng bằng file.

# 

# ****CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN****

## ****5.1. Kết luận****

Đề tài “Gửi tập tin âm thanh chia thành nhiều đoạn” đã hoàn thành mô phỏng một hệ thống truyền dữ liệu an toàn, tích hợp đầy đủ các yếu tố bảo mật cơ bản gồm: **mã hóa**, **kiểm tra toàn vẹn**, **xác thực nguồn gửi** và **mã hóa khóa phiên**.

Thông qua việc áp dụng các thuật toán:

**- Triple DES** (mã hóa nội dung),

**- SHA-512** (bảo toàn tính toàn vẹn),

**- RSA 2048-bit** (ký số và mã hóa khóa phiên),

- cùng với định dạng **JSON** để đóng gói gọn gàng dữ liệu và sử dụng **Base64** để truyền nội dung nhị phân,

- ... hệ thống đã đảm bảo được 3 nguyên tắc cốt lõi của bảo mật thông tin: **Confidentiality – Integrity – Authentication** (CIA).

Ngoài ra, quá trình chia nhỏ file, xử lý từng đoạn riêng lẻ giúp mô phỏng một phương thức truyền an toàn và có thể phục hồi nếu lỗi từng phần xảy ra. Kết quả thực nghiệm cho thấy hệ thống có thể phát hiện giả mạo, dữ liệu hỏng, và chỉ chấp nhận khi mọi yếu tố bảo mật đều hợp lệ.

Qua quá trình thực hiện, nhóm cũng đã có cơ hội tiếp cận và ứng dụng các kiến thức lý thuyết về bảo mật thông tin vào mô hình thực tế, đồng thời rèn luyện kỹ năng lập trình Python và thao tác với các định dạng dữ liệu hiện đại.

## 5.2. Hạn chế

Dù hệ thống đã hoàn thành đúng mục tiêu đề tài đặt ra, vẫn còn một số điểm hạn chế cần khắc phục:

**- Chưa xử lý lỗi động khi truyền thực tế:** Hệ thống mới mô phỏng truyền qua file tĩnh, chưa có giao tiếp mạng (socket, HTTP).

**- Hiệu suất chưa tối ưu:** Việc sử dụng RSA để ký và mã hóa có thể gây chậm khi mở rộng quy mô với nhiều đoạn hoặc file lớn.

**- Khả năng khôi phục khi thiếu đoạn:** Hệ thống hiện yêu cầu đủ 3 đoạn mới có thể ghép file; chưa có cơ chế khôi phục nếu mất một phần.

**- Chưa xác thực danh tính người gửi/nhận qua hệ thống người dùng:** Nếu triển khai trong môi trường thực, cần có quản lý khóa và định danh rõ ràng.

## ****5.3. Hướng phát triển****

Trong tương lai, hệ thống có thể được mở rộng và nâng cấp theo một số hướng như sau:

**- Triển khai truyền dữ liệu thật qua mạng (client-server hoặc peer-to-peer)**, dùng socket hoặc HTTP, giúp mô phỏng gần hơn môi trường thực tế.

**- Sử dụng thuật toán AES thay cho Triple DES** nhằm nâng cao hiệu suất và bảo mật (vì Triple DES đang dần bị thay thế).

**- Tích hợp giao diện người dùng (GUI)** để thao tác dễ hơn, đặc biệt hữu ích trong bài học hoặc thử nghiệm cho sinh viên.

**- Thêm hệ thống xác thực người dùng (login, cấp phát khóa)** để đảm bảo danh tính bên gửi/nhận không bị giả mạo.

**- Tạo module kiểm thử tự động** cho từng bước: mã hóa, ký, kiểm tra hash, giải mã… phục vụ kiểm tra lỗi nhanh chóng.

## ****5.4. Cảm nghĩ của nhóm thực hiện****

Qua đề tài, nhóm không chỉ hiểu sâu hơn về các thuật toán bảo mật mà còn nắm vững cách xây dựng một hệ thống ứng dụng chúng một cách hợp lý. Đây là cơ hội tốt để rèn luyện kỹ năng lập trình, kỹ năng làm việc nhóm và tư duy phân tích hệ thống bảo mật trong thế giới thực.

Đề tài cũng giúp nhận thức rõ ràng hơn rằng: trong thời đại số hóa, **bảo mật thông tin không chỉ là kiến thức chuyên môn mà còn là yêu cầu sống còn** với bất kỳ hệ thống nào có liên quan đến dữ liệu người dùng.

# KẾT LUẬN



## 

Trong thời đại bùng nổ thông tin, việc bảo vệ dữ liệu khỏi các nguy cơ đánh cắp, giả mạo hoặc phá hoại trở nên quan trọng hơn bao giờ hết. Với mục tiêu rèn luyện tư duy bảo mật hệ thống cũng như khả năng triển khai các kỹ thuật mã hóa thực tế, đề tài “Gửi tập tin âm thanh chia thành nhiều đoạn” đã được nhóm nghiên cứu, thiết kế và triển khai thành công như một mô hình mô phỏng hệ thống truyền dữ liệu an toàn theo hướng chia nhỏ gói tin.

Bằng cách sử dụng kết hợp các kỹ thuật bảo mật quan trọng như mã hóa đối xứng **Triple DES**, mã hóa bất đối xứng **RSA 2048-bit**, hàm băm **SHA-512** và chữ ký số, nhóm đã xây dựng được một quy trình truyền tin an toàn gồm nhiều lớp bảo vệ. Hệ thống cho phép chia file âm thanh thành các phần nhỏ, mỗi phần đều được bảo vệ độc lập bằng mã hóa và chữ ký, sau đó đóng gói thành định dạng JSON để mô phỏng quá trình gửi đi. Ở phía nhận, hệ thống tiến hành kiểm tra tính toàn vẹn và xác thực từng đoạn, giải mã và nối lại thành file hoàn chỉnh nếu tất cả dữ liệu đều hợp lệ.

Kết quả thực nghiệm cho thấy hệ thống có khả năng:

- Phát hiện và loại bỏ những đoạn tin bị giả mạo hoặc sai lệch;

- Bảo toàn tuyệt đối dữ liệu đầu cuối khi sử dụng đúng cặp khóa và quy trình giải mã;

- Phát hiện lỗi trong từng đoạn nhỏ thay vì toàn bộ file, tăng độ an toàn và tính khả dụng.

Thông qua đề tài, nhóm không chỉ nắm vững kiến thức về lý thuyết bảo mật mà còn hiểu rõ cách tổ chức và triển khai hệ thống bảo mật trong thực tế. Việc áp dụng những thuật toán phức tạp như RSA và SHA-512 bằng ngôn ngữ Python giúp củng cố khả năng lập trình ứng dụng vào các bài toán bảo mật – một kỹ năng rất cần thiết cho sinh viên công nghệ thông tin trong kỷ nguyên số hóa.

Mặc dù hệ thống vẫn còn một số điểm hạn chế như: chưa tích hợp giao tiếp mạng thực sự, hiệu suất RSA còn thấp nếu mở rộng quy mô, và chưa xử lý tình huống mất một đoạn tin; tuy nhiên, đây là nền tảng quan trọng để tiếp tục nâng cấp thành các hệ thống truyền tải thực tế với độ an toàn cao hơn.

Từ góc độ học thuật, đề tài là một minh chứng rõ ràng cho việc kết nối giữa lý thuyết và thực hành. Từ góc độ thực tiễn, đây là bước khởi đầu cho những ứng dụng thực tế trong bảo mật dữ liệu âm thanh, hình ảnh hoặc các tệp lớn trong môi trường mạng.

Với những gì đạt được, nhóm tin rằng đây là một đề tài mang tính ứng dụng cao, giúp sinh viên bước đầu làm quen với cách xây dựng một hệ thống bảo mật toàn diện, có thể mở rộng và phát triển thành sản phẩm hoàn chỉnh trong tương lai.

# DANH MỤC TÀI LIỆU THAM KHẢO

1. William Stallings, Cryptography and Network Security: Principles and Practice, 7th Edition, Pearson Education, 2017.
2. Bruce Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd Edition, Wiley, 1996.
3. Trần Đình Hương, Giáo trình An toàn và Bảo mật Thông tin, NXB Giáo dục Việt Nam, 2020.
4. National Institute of Standards and Technology (NIST), **FIPS PUB 46-3**: Data Encryption Standard (DES), 1999.  
   Truy cập tại: [https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.46-3.pdf](https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.46-3.pdf" \t "_new)
5. National Institute of Standards and Technology (NIST), **FIPS PUB 180-4**: Secure Hash Standard (SHS), 2015.  
   Truy cập tại: [https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf](https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf" \t "_new)
6. National Institute of Standards and Technology (NIST), **FIPS PUB 186-4**: Digital Signature Standard (DSS), 2013.  
   Truy cập tại: [https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf](https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf" \t "_new)
7. PyCryptodome – Python Cryptographic Library.  
   Truy cập tại: [https://pycryptodome.readthedocs.io/](https://pycryptodome.readthedocs.io/" \t "_new)
8. Python Software Foundation, json – JSON encoder and decoder documentation.  
   Truy cập tại: [https://docs.python.org/3/library/json.html](https://docs.python.org/3/library/json.html" \t "_new)
9. Wikipedia – Triple DES.  
   Truy cập tại: [https://en.wikipedia.org/wiki/Triple\_DES](https://en.wikipedia.org/wiki/Triple_DES" \t "_new)
10. Wikipedia – RSA (cryptosystem).  
    Truy cập tại: [https://en.wikipedia.org/wiki/RSA\_(cryptosystem)](https://en.wikipedia.org/wiki/RSA_(cryptosystem)" \t "_new)
11. MDN Web Docs – Base64 Encoding and Decoding.  
    Truy cập tại: [https://developer.mozilla.org/en-US/docs/Glossary/Base64](https://developer.mozilla.org/en-US/docs/Glossary/Base64" \t "_new)