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 THỒNG TIN**

**ĐỀ TÀI: ỨNG DỤNG BẢO MẬT TIN NHẮN ÂM THANH**

**VỚI MÃ HÓA DES VÀ XÁC THỰC RSA**

**Giáo viên hướng dẫn: TS. Trần Quý Nam**

**Sinh viên thực hiện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Mã sv** | **Họ và tên** | **Lớp** |
| 1 | 1771202334 | Nguyễn Hữu Hưng | CNTT 17-04 |
| 2 | 1771020420 | Nguyễn Đức Linh | CNTT 17-04 |

**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**

**ĐỀ TÀI: ỨNG DỤNG BẢO MẬT TIN NHẮN ÂM THANH**

**VỚI MÃ HÓA DES VÀ XÁC THỰC RSA**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1 | 1771020334 | Nguyễn Hữu Hưng | 07/09/2005 |  |  |
| 2 | 1771020420 | Nguyễn Đức Linh | 24/10/2005 |  |  |

### 

### CÁN BỘ CHẤM THI

**Hà Nội, năm 2025**

# LỜI NÓI ĐẦU

Trong thời đại công nghệ số phát triển mạnh mẽ, nhu cầu bảo mật thông tin ngày càng trở nên cấp thiết, đặc biệt là trong các hệ thống truyền thông cá nhân như nhắn tin và gọi thoại. Các nguy cơ như nghe lén, giả mạo danh tính, hay sửa đổi nội dung đang đặt ra thách thức lớn cho người phát triển ứng dụng. Do đó, việc ứng dụng các kỹ thuật mã hóa và xác thực hiện đại vào việc xây dựng hệ thống truyền tin an toàn là một hướng đi quan trọng và thực tiễn.

Bài tập lớn này được thực hiện trong khuôn khổ học phần Nhập môn An toàn và Bảo mật Thông tin, với mục tiêu xây dựng một hệ thống gửi – nhận tin nhắn âm thanh an toàn qua mạng ngang hàng (P2P). Hệ thống sử dụng thuật toán mã hóa đối xứng DES để đảm bảo tính bí mật của dữ liệu, thuật toán RSA để xác thực người dùng và trao đổi khóa phiên, cùng với SHA-256 để kiểm tra tính toàn vẹn của nội dung. Toàn bộ giải pháp được triển khai bằng ngôn ngữ Python, có tích hợp giao diện người dùng trực quan, hỗ trợ ghi âm, phát lại và gửi tin nhắn một cách bảo mật.

Qua quá trình thực hiện đề tài, nhóm đã có cơ hội củng cố kiến thức lý thuyết về mật mã học, đồng thời rèn luyện kỹ năng lập trình, tư duy thiết kế hệ thống, và làm việc theo hướng tiếp cận an toàn thông tin. Nhóm xin chân thành cảm ơn thầy/cô giảng dạy đã định hướng, hỗ trợ và tạo điều kiện để nhóm có thể hoàn thành tốt bài tập lớn này.

**MỤC LỤC**

**[LỜI NÓI ĐẦU 3](#_Toc30916)**

**[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT VÀ PHÂN TÍCH BÀI TOÁN 5](#_Toc12829)**

**[1.1. Đặt vấn đề 5](#_Toc12611)**

**[1.2. Kiến thức nền tảng (mở rộng) 5](#_Toc25905)**

**[1.3. Phân tích bài toán và yêu cầu hệ thống (mở rộng) 9](#_Toc23670)**

**[1.4. Các nguy cơ tấn công và giải pháp phòng tránh 11](#_Toc2024)**

**[1.5. Tính ứng dụng và mở rộng hệ thống 11](#_Toc12927)**

**[1.6. Phân tích yêu cầu của bài toán 13](#_Toc18910)**

**[1.7. Các thách thức và mục tiêu cần đạt được 14](#_Toc9350)**

**[CHƯƠNG 2: MÔ TẢ GIẢI PHÁP VÀ THUẬT TOÁN SỬ DỤNG 16](#_Toc16304)**

**[2.1. Luồng xử lý tổng quát của hệ thống 16](#_Toc1149)**

**[2.2. Giai đoạn 1: Handshake 16](#_Toc27691)**

**[2.3. Giai đoạn 2: Xác thực và Trao khóa 16](#_Toc26646)**

**[2.4. Giai đoạn 3: Mã hóa và Kiểm tra toàn vẹn 17](#_Toc7141)**

**[2.5. Xử lý phía người nhận 18](#_Toc2168)**

**[2.6. Phân tích bảo mật 18](#_Toc28335)**

**[CHƯƠNG 3: PHÂN TÍCH HIỆN THỰC – MÃ NGUỒN VÀ THỬ NGHIỆM 20](#_Toc11371)**

**[3.1. Mô tả cấu trúc mã nguồn 20](#_Toc1407)**

**[3.2. Phân tích chi tiết từng phần mã 20](#_Toc22808)**

**[3.3. Thử nghiệm và kết quả 25](#_Toc16579)**

**[CHƯƠNG 4: ĐÁNH GIÁ, NHẬN XÉT VÀ HƯỚNG PHÁT TRIỂN 27](#_Toc9612)**

**[4.1. Đánh giá hiệu quả hệ thống 27](#_Toc6721)**

**[4.2. Nhận xét về các thuật toán sử dụng 27](#_Toc12196)**

**[4.3. Hạn chế của hệ thống 28](#_Toc16103)**

**[4.4. Đề xuất cải tiến và hướng phát triển 28](#_Toc25868)**

**[KẾT LUẬN 30](#_Toc28735)**

**[DANH MỤC TÀI LIỆU THAM KHẢO 31](#_Toc10611)**

# CHƯƠNG 1: CƠ SỞ LÝ THUYẾT VÀ PHÂN TÍCH BÀI TOÁN

## 1.1. Đặt vấn đề

### ****1.1.1. Tổng quan về an toàn thông tin trong truyền thông hiện đại****

Trong thời đại số, thông tin trở thành tài sản vô giá của cá nhân, tổ chức và quốc gia. Việc bảo vệ thông tin không còn là một tùy chọn mà là yêu cầu bắt buộc trong mọi hệ thống giao tiếp và truyền thông. Các mối đe dọa như tấn công mạng, rò rỉ dữ liệu, và giả mạo danh tính không chỉ gây thiệt hại về mặt kinh tế mà còn ảnh hưởng nghiêm trọng đến uy tín và tính riêng tư của người dùng.

Trong hệ thống truyền thông truyền thống, mô hình máy chủ – máy khách đóng vai trò trung gian điều phối và kiểm soát bảo mật. Tuy nhiên, với sự phát triển của các ứng dụng phi tập trung (decentralized applications), mô hình truyền thông ngang hàng (Peer-to-Peer – P2P) ngày càng trở nên phổ biến. Mô hình này mang đến sự linh hoạt, giảm thiểu độ trễ và tăng khả năng mở rộng nhưng cũng đặt ra nhiều thách thức về bảo mật.

### ****1.1.2. Lý do chọn đề tài****

Trong bối cảnh đó, đề tài nghiên cứu và xây dựng hệ thống truyền tin âm thanh bảo mật qua mạng P2P trở nên cấp thiết. Bằng cách ứng dụng kết hợp ba công nghệ bảo mật chủ chốt gồm:

* **DES (Data Encryption Standard):** Mã hóa nội dung dữ liệu âm thanh.
* **RSA (Rivest-Shamir-Adleman):** Mã hóa khóa bí mật và xác thực người gửi
* **SHA-256 (Secure Hash Algorithm 256-bit):** Kiểm tra tính toàn vẹn của dữ liệu.

Hệ thống được kỳ vọng sẽ đảm bảo ba yếu tố quan trọng: tính bảo mật, xác thực và toàn vẹn – đáp ứng yêu cầu khắt khe của truyền thông hiện đại.

## ****1.2. Kiến thức nền tảng (mở rộng)****

### ****1.2.1. Mô hình mạng P2P (Peer-to-Peer)****

#### Khái niệm và mô hình

Mạng ngang hàng (P2P) là một mô hình trong đó mỗi thiết bị (node) trong mạng đóng vai trò vừa là máy khách (client), vừa là máy chủ (server). Các thiết bị trong mạng P2P có quyền và trách nhiệm ngang nhau, và đều có thể khởi tạo hoặc xử lý yêu cầu truyền dữ liệu. Không giống như kiến trúc truyền thống kiểu client-server (nơi tất cả dữ liệu đều phải đi qua một máy chủ trung tâm), trong mô hình P2P, các nút có thể liên kết và giao tiếp trực tiếp với nhau.

Một hệ thống P2P có thể là:

* **P2P thuần túy:** Không có bất kỳ nút trung tâm nào. Mỗi nút đều bình đẳng.
* **P2P lai:** Có thể có một máy chủ trung gian nhỏ để hỗ trợ tìm kiếm hoặc điều phối nhưng vẫn giữ tính phi tập trung trong truyền dữ liệu.

#### Ưu điểm của mạng P2P

* **Không có điểm lỗi trung tâm (No Single Point of Failure):** Nếu một nút trong mạng bị lỗi, các nút khác vẫn có thể hoạt động bình thường.
* **Chi phí triển khai thấp:** Không cần đầu tư vào máy chủ trung tâm, giảm chi phí phần cứng và bảo trì.
* **Khả năng mở rộng linh hoạt:** Dễ dàng bổ sung nút mới mà không ảnh hưởng đến toàn bộ mạng.
* **Phân tán tải tốt:** Lưu lượng truy cập được phân bố đều, giảm tải cho bất kỳ nút nào.

#### ****Nhược điểm và rủi ro bảo mật****

* **Khó kiểm soát truy cập:** Không có điểm điều phối trung tâm dẫn đến khó áp dụng các chính sách truy cập.
* **Không đảm bảo tính toàn vẹn dữ liệu:** Nếu không có mã hóa, dữ liệu dễ bị thay đổi bởi các bên thứ ba.
* **Dễ bị tấn công:** Các loại tấn công thường gặp trong P2P bao gồm
* **Tấn công từ chối dịch vụ (DoS).**
* **Tấn công trung gian (Man-in-the-Middle).**
* **Tấn công phát lại (Replay).**
* **Tấn công mạo danh (Spoofing).**

Do đó, việc thiết kế hệ thống bảo mật hiệu quả trong mạng P2P là vô cùng cần thiết.

### ****1.2.2. Mã hóa đối xứng – DES****

#### ****Giới thiệu thuật toán DES****

DES là một trong những thuật toán mã hóa đối xứng đầu tiên được chuẩn hóa và sử dụng rộng rãi trong thực tiễn. Nó mã hóa dữ liệu theo từng khối (block cipher) với kích thước khối cố định 64 bit. Mỗi lần mã hóa sử dụng cùng một khóa cho cả mã hóa và giải mã.

#### ****Chi tiết kỹ thuật thuật toán DES****

DES gồm 16 vòng (rounds) biến đổi dữ liệu, trong mỗi vòng dữ liệu được hoán vị và xử lý qua các hàm logic, sử dụng các khóa con khác nhau (subkeys) được tạo từ khóa chính thông qua các phép hoán vị và dịch bit.

Mặc dù ngày nay DES được xem là không còn đủ mạnh để chống lại các tấn công brute-force, nhưng nó vẫn là một ví dụ quan trọng để học tập và sử dụng trong các hệ thống có phạm vi nhỏ, thời gian thực ngắn và đã có lớp mã hóa bổ sung.

#### ****Chế độ hoạt động CBC****

Chế độ CBC giúp tăng độ an toàn bằng cách "xâu chuỗi" kết quả mã hóa:

* Trước khi mã hóa, plaintext được XOR với ciphertext của khối trước đó.
* Khối đầu tiên được XOR với một vector khởi tạo IV.
* Nhờ đó, hai khối dữ liệu giống hệt sẽ cho ra hai bản mã khác nhau nếu IV khác nhau.

CBC giúp chống lại phân tích thống kê và là chế độ an toàn hơn ECB (Electronic Codebook).

### ****1.2.3. Mã hóa bất đối xứng – RSA****

#### ****Đặc điểm và tầm quan trọng****

RSA được coi là nền tảng của mã hóa bất đối xứng. Sức mạnh của RSA đến từ độ khó của việc phân tích một số nguyên lớn thành thừa số nguyên tố – một bài toán chưa có lời giải nhanh.

RSA thường sử dụng khóa dài 1024, 2048 hoặc 4096 bit tùy theo mức độ bảo mật yêu cầu. Trong hệ thống này, ta sử dụng RSA-2048 cho mức an toàn tốt và hiệu suất chấp nhận được.

#### ****Ứng dụng cụ thể trong hệ thống****

* **Bảo mật khóa phiên:** Khóa DES là khóa phiên được tạo ngẫu nhiên mỗi lần truyền, sau đó được mã hóa bằng khóa công khai RSA và gửi cho người nhận.
* **Xác thực danh tính:** Dữ liệu định danh (username, IP, timestamp...) được ký bằng khóa riêng, người nhận kiểm tra bằng khóa công khai → đảm bảo không bị giả mạo

### ****1.2.4. Hàm băm SHA-256****

#### ****Nguyên lý hoạt động****

SHA-256 nhận dữ liệu đầu vào bất kỳ (không giới hạn độ dài) và trả về một chuỗi 256-bit (64 ký tự hex). Điểm mạnh của SHA-256:

* Nhạy cảm với thay đổi dữ liệu (avalanche effect).
* Không thể đảo ngược: không thể tìm được dữ liệu đầu vào từ mã băm.
* Chống sinh mã băm trùng.

#### ****Ứng dụng trong hệ thống****

* Tạo fingerprint cho dữ liệu để kiểm tra toàn vẹn.
* Làm đầu vào cho chữ ký số RSA.
* Phát hiện mọi sự thay đổi dù nhỏ nhất trong nội dung.

### ****1.2.5. Chữ ký số và xác thực nguồn gốc****

#### ****Tầm quan trọng****

Chữ ký số trong hệ thống đóng vai trò xác minh rằng dữ liệu:

* Đến từ đúng người gửi.
* Không bị thay đổi trên đường truyền.

#### ****Quy trình cụ thể:****

1. Người gửi tạo mã băm SHA-256 từ metadata.
2. Mã băm được ký bằng khóa riêng RSA.
3. Người nhận nhận được chữ ký, giải mã bằng khóa công khai để so sánh với mã băm tự tính → nếu giống nhau thì hợp lệ.

Chữ ký số là lớp bảo vệ quan trọng thứ hai sau mã hóa nội dung.

## ****1.3. Phân tích bài toán và yêu cầu hệ thống (mở rộng)****

### ****1.3.1. Kiến trúc hệ thống tổng quát****

#### ****Thành phần 1: Client Gửi****

* Ghi âm âm thanh qua micro
* Mã hóa dữ liệu bằng DES-CBC.
* Tạo SHA-256 cho dữ liệu + metadata.
* Ký SHA-256 bằng RSA (chữ ký số).
* Mã hóa khóa DES bằng RSA.
* Gửi toàn bộ gói dữ liệu đến client nhận.

#### ****Thành phần 2: Client Nhận****

* Giải mã khóa DES bằng khóa riêng.
* Kiểm tra chữ ký RSA.
* So sánh mã SHA-256.
* Nếu hợp lệ thì giải mã dữ liệu và phát lại âm thanh

#### ****Thành phần 3: Kênh truyền UDP****

* Giao tiếp hai chiều không cần trung gian.
* Tối ưu độ trễ, băng thông.
* Có thể sử dụng thêm xác thực IP.

### ****1.3.2. Yêu cầu bảo mật chi tiết****

1. **Tính bí mật:** Dữ liệu phải được mã hóa – sử dụng DES.
2. **Tính xác thực:** Sử dụng RSA để đảm bảo đúng nguồn gửi.
3. **Tính toàn vẹn:** So sánh mã SHA-256.

Mỗi lớp đóng vai trò riêng biệt nhưng tương hỗ, tạo nên cơ chế bảo mật ba lớp.

### ****1.3.3. Yêu cầu xử lý thời gian thực****

Âm thanh là dữ liệu thời gian thực → hệ thống phải đáp ứng:

* Ghi âm và phát lại tức thì.
* Mã hóa/giải mã trong thời gian nhỏ hơn 500ms.
* Không làm giảm chất lượng âm thanh

Các công nghệ như pyaudio, socket, PyCryptodome được lựa chọn nhờ hiệu năng tốt, hỗ trợ nền tảng đa dạng.

## ****1.4. Các nguy cơ tấn công và giải pháp phòng tránh****

### ****1.4.1. Tấn công Man-in-the-Middle (MITM)****

**Vấn đề:** Kẻ tấn công chèn vào giữa quá trình truyền, đọc hoặc chỉnh sửa dữ liệu.

**Giải pháp:**

* Mã hóa nội dung.
* Ký metadata bằng RSA.
* Kiểm tra chữ ký → nếu sai thì từ chối gói tin.

### ****1.4.2. Tấn công Replay****

**Vấn đề:** Dữ liệu bị ghi lại và phát lại để lừa hệ thống.

**Giải pháp:**

* Đưa timestamp vào metadata.
* Nếu thời gian chênh lệch quá ngưỡng cho phép thì từ chối.

### ****1.4.3. Tấn công IP giả mạo****

**Vấn đề:** Một IP không hợp lệ gửi gói tin giả.

**Giải pháp:**

* Duy trì danh sách IP hợp lệ (whitelist).
* Kiểm tra khóa RSA tương ứng trước khi xử lý gói tin.

## ****1.5. Tính ứng dụng và mở rộng hệ thống****

### ****1.5.1. Ứng dụng thực tiễn****

* **Giao tiếp bảo mật nội bộ:** Trong công ty, bệnh viện, quân đội.
* **Truyền âm thanh bảo mật trong IoT:** Ví dụ camera IP có ghi âm.
* **Hệ thống học trực tuyến:** Gửi nội dung ghi âm đã mã hóa.
* **Điện thoại VoIP mã hóa đầu-cuối.**

### ****1.5.2. Hướng mở rộng****

* **Truyền video:** Sử dụng AES hoặc RSA để bảo mật khung hình.
* **Chuyển sang AES:** Để tăng độ an toàn và tốc độ so với DES.
* **Xác thực đa yếu tố:** Kết hợp mật khẩu, OTP và chứng chỉ.
* **Triển khai Blockchain:** Lưu trữ lịch sử truyền tin và kiểm tra giả mạo.

### ****1.1.3. Vai trò của bảo mật trong truyền tin P2P****

Trong các hệ thống truyền thông ngang hàng (Peer-to-Peer – P2P), dữ liệu được truyền trực tiếp giữa các thiết bị người dùng mà không cần thông qua máy chủ trung gian. Ưu điểm của P2P là giảm thiểu độ trễ và tiết kiệm tài nguyên hệ thống. Tuy nhiên, mô hình này cũng tiềm ẩn nhiều rủi ro như nghe lén, giả mạo dữ liệu hoặc tấn công trung gian (Man-in-the-Middle) nếu không được bảo vệ đúng cách. Do đó, việc thiết lập các biện pháp bảo mật là yêu cầu thiết yếu để đảm bảo rằng thông tin truyền đi không bị thay đổi, giả mạo hoặc truy cập trái phép. Đặc biệt, trong trường hợp truyền dữ liệu âm thanh – một dạng dữ liệu nhạy cảm – vấn đề bảo mật càng trở nên quan trọng.

### Tầm quan trọng của việc bảo mật nội dung và xác thực người dùng trong hệ thống chat âm thanh

Trong một hệ thống chat âm thanh, việc bảo mật nội dung và xác thực người dùng đóng vai trò quyết định đến mức độ tin cậy và an toàn của hệ thống. Cụ thể:

* **Bảo mật nội dung** đảm bảo rằng chỉ người nhận hợp lệ mới có thể truy cập và nghe được tin nhắn, thông qua việc mã hóa bằng thuật toán DES ở chế độ CBC.
* **Xác thực người dùng** đảm bảo rằng thông tin nhận được thực sự đến từ người gửi mong muốn, sử dụng phương pháp ký số với RSA kết hợp thuật toán băm SHA-256.
* **Tính toàn vẹn** giúp phát hiện các chỉnh sửa trái phép trong quá trình truyền dữ liệu, bằng cách tạo và so sánh mã băm SHA-256 của dữ liệu.

Sự thiếu vắng bất kỳ yếu tố nào trong ba thành phần trên đều có thể dẫn đến rò rỉ dữ liệu, giả mạo danh tính hoặc phá vỡ tính toàn vẹn của hệ thống.

### ****1.6. Phân tích yêu cầu của bài toán****

**Mã hóa dữ liệu âm thanh bằng DES (CBC mode)**

Thuật toán DES (Data Encryption Standard) là phương pháp mã hóa đối xứng sử dụng cùng một khóa cho cả quá trình mã hóa và giải mã. Trong dự án này, chế độ hoạt động CBC (Cipher Block Chaining) được sử dụng, trong đó mỗi khối dữ liệu sẽ được kết hợp với kết quả mã hóa của khối trước đó để tăng tính bảo mật và tránh mô hình lặp lại trong ciphertext. Dữ liệu âm thanh, sau khi được ghi lại, sẽ được xử lý dưới dạng nhị phân và chia thành các khối có độ dài phù hợp với yêu cầu của DES (8 byte/block).

**Xác thực người dùng bằng RSA kết hợp SHA-256**

RSA là thuật toán mã hóa bất đối xứng sử dụng một cặp khóa công khai và riêng tư để mã hóa và xác thực. Với độ dài 2048-bit, RSA cung cấp độ an toàn cao khi dùng để:

* Mã hóa khóa DES, đảm bảo khóa chỉ có thể được giải mã bởi người nhận sở hữu khóa riêng tư.
* Ký thông tin định danh (metadata) như tên người dùng, giúp xác thực nguồn gốc dữ liệu.

Thuật toán SHA-256 được sử dụng để tạo ra một bản băm (hash) duy nhất từ dữ liệu định danh, làm cơ sở cho quá trình ký số. Nhờ đó, nếu dữ liệu bị thay đổi, chữ ký RSA sẽ không còn hợp lệ, giúp phát hiện giả mạo.

**Kiểm tra tính toàn vẹn dữ liệu bằng SHA-256**

Việc kiểm tra tính toàn vẹn được thực hiện bằng cách tạo mã băm SHA-256 từ dữ liệu đã mã hóa (ciphertext) ở phía gửi và so sánh với mã băm được tạo lại ở phía nhận. Nếu hai mã băm trùng khớp, có thể khẳng định rằng dữ liệu chưa bị thay đổi trong quá trình truyền. Ngược lại, nếu không trùng, hệ thống sẽ cảnh báo lỗi và không tiếp nhận dữ liệu.

**Tổng quan kiến trúc hệ thống**

Hệ thống gồm ba thành phần chính:

1. **Client gửi âm thanh**: Ghi âm và mã hóa dữ liệu bằng DES, tạo mã băm SHA-256, ký metadata bằng RSA. Dữ liệu được đóng gói và gửi dưới dạng:
2. {
3. "signed\_info": "<chữ ký RSA>",
4. "encrypted\_des\_key": "<khóa DES đã mã hóa bằng RSA>",
5. "cipher": "<dữ liệu âm thanh đã mã hóa>",
6. "hash": "<mã SHA-256>"
7. }
8. **Client nhận âm thanh**: Thực hiện giải mã khóa DES bằng RSA, kiểm tra mã băm và chữ ký, giải mã dữ liệu và phát lại nếu hợp lệ. Nếu dữ liệu bị lỗi hoặc không xác thực được, hệ thống sẽ từ chối và gửi thông báo lỗi (NACK).
9. **Kênh truyền dữ liệu**: Kết nối trực tiếp P2P giữa hai client mà không thông qua máy chủ trung gian, giảm độ trễ và chi phí vận hành.

### ****1.7. Các thách thức và mục tiêu cần đạt được****

**Truyền tin an toàn qua mạng P2P**

Với đặc trưng không qua máy chủ trung gian, mạng P2P tiềm ẩn rủi ro về bảo mật do dữ liệu truyền trực tiếp giữa các thiết bị. Để giải quyết vấn đề này, hệ thống cần có các lớp mã hóa và xác thực tin cậy. Giải pháp được lựa chọn bao gồm: mã hóa khóa bằng RSA, mã hóa nội dung bằng DES (CBC) và tạo mã băm SHA-256 để kiểm tra tính toàn vẹn.

**Đảm bảo tính đúng đắn, toàn vẹn và bí mật thông tin**

Hệ thống phải đảm bảo ba yếu tố:

* **Đúng đắn**: Người nhận có thể xác minh danh tính người gửi thông qua chữ ký RSA.
* **Toàn vẹn**: Không có dữ liệu nào bị thay đổi nếu mã băm SHA-256 trùng khớp.
* **Bí mật**: Nội dung âm thanh được mã hóa bằng DES, chỉ có người nhận mới giải mã được nếu có khóa đúng.

**Xử lý file âm thanh trong thời gian thực**

Một thách thức khác là đảm bảo khả năng xử lý file âm thanh nhanh và ổn định để phục vụ giao tiếp gần thời gian thực. Để khắc phục, hệ thống sử dụng định dạng WAV với cấu trúc đơn giản, hỗ trợ xử lý nhanh. Việc mã hóa/giải mã cũng được thực hiện bằng thư viện hiệu năng cao (PyCryptodome) và được thực hiện song song với giao tiếp mạng để giảm thiểu độ trễ.

# CHƯƠNG 2: MÔ TẢ GIẢI PHÁP VÀ THUẬT TOÁN SỬ DỤNG

## ****2.1. Luồng xử lý tổng quát của hệ thống****

Hệ thống truyền tin an toàn sử dụng giao thức ngang hàng (P2P) được thiết kế gồm ba giai đoạn chính như sau:

1. **Giai đoạn Handshake:** Hai bên tham gia thiết lập kết nối ban đầu và trao đổi khóa công khai RSA để chuẩn bị cho các bước bảo mật sau.
2. **Giai đoạn Xác thực và Trao khóa:** Mỗi bên tiến hành ký định danh bằng RSA kết hợp SHA-256 nhằm xác thực danh tính, sau đó thực hiện mã hóa khóa phiên (DES) bằng khóa công khai RSA của đối phương và gửi qua mạng.
3. **Giai đoạn Mã hóa và Kiểm tra toàn vẹn:** Dữ liệu âm thanh được mã hóa bằng thuật toán DES ở chế độ CBC để đảm bảo tính bí mật. Đồng thời, hệ thống tạo hàm băm SHA-256 từ dữ liệu đã mã hóa nhằm kiểm tra tính toàn vẹn trong quá trình truyền tin. Gói dữ liệu sau đó được gửi kèm theo chữ ký và mã băm.

## ****2.2. Giai đoạn 1: Handshake****

Trong giai đoạn khởi tạo kết nối, bên gửi đầu tiên truyền chuỗi tín hiệu "Hello!" và bên nhận phản hồi "Ready!" để xác nhận sẵn sàng. Tiếp theo, hai bên thực hiện trao đổi khóa công khai RSA bằng cách gửi nội dung khóa .pem được mã hóa dưới dạng chuỗi base64 qua socket TCP.

Quá trình này được hiện thực thông qua các hàm send\_json(...) và recv\_json(...) trong tập tin network\_utils.py. Cả giao diện đồ họa (GUI) và giao diện dòng lệnh (CLI) đều triển khai luồng xử lý này trong các hàm connect() (tập tin gui.py) hoặc handle\_connection() (tập tin main.py).

## ****2.3. Giai đoạn 2: Xác thực và Trao khóa****

### ****Xác thực người dùng****

Sau khi trao đổi khóa, mỗi bên sẽ ký tên định danh của mình (ví dụ: "Alice") bằng khóa riêng RSA và thuật toán băm SHA-256. Bên nhận sử dụng khóa công khai nhận được trước đó để xác thực chữ ký, qua đó đảm bảo tính xác thực của người gửi.

### ****Trao đổi khóa phiên (DES)****

Máy chủ sinh ngẫu nhiên một khóa DES có độ dài 8 byte. Khóa này được mã hóa bằng khóa công khai RSA của máy khách (sử dụng chế độ OAEP) và gửi sang để thiết lập kênh truyền mã hóa đối xứng.

### Định dạng gói tin gửi

{

"type": "auth",

"name": "Alice",

"signature": "<base64 của chữ ký RSA>"

}

{

"type": "des\_key",

"data": "<base64 của khóa DES đã mã hóa bằng RSA>"

}

Các thao tác ký, mã hóa và xác minh được thực hiện bằng các hàm sign\_data(), rsa\_encrypt() và verify\_signature() trong tập tin crypto\_utils.py.

## 2.4. Giai đoạn 3: Mã hóa và Kiểm tra toàn vẹn

### Mã hóa dữ liệu âm thanh bằng DES (chế độ CBC)

Dữ liệu âm thanh được ghi bằng thiết bị micro, lưu dưới dạng file .wav bằng thư viện sounddevice, sau đó đọc ở dạng nhị phân. Quá trình mã hóa sử dụng thuật toán DES ở chế độ CBC, kết hợp với một vector khởi tạo (IV) ngẫu nhiên.

### Tạo hàm băm SHA-256

Dữ liệu sau khi được mã hóa sẽ được băm bằng thuật toán SHA-256 để tạo fingerprint, phục vụ cho việc kiểm tra tính toàn vẹn.

### Định dạng gói tin gửi

{

"type": "voice",

"data": "<base64 dữ liệu đã mã hóa bằng DES>"

}

Các hàm liên quan gồm:

* record\_audio() trong audio\_utils.py – ghi âm.
* des\_encrypt() trong crypto\_utils.py – mã hóa DES.
* hash\_data() trong crypto\_utils.py – tạo mã băm SHA-256.

## ****2.5. Xử lý phía người nhận****

1. Khi nhận được dữ liệu, hệ thống thực hiện các bước sau:
2. Nhận và phân loại gói dữ liệu (văn bản hoặc âm thanh).
3. Giải mã khóa DES nếu đây là bước trao khóa.
4. Kiểm tra chữ ký RSA và mã băm để đảm bảo tính xác thực và toàn vẹn.
5. Nếu hợp lệ, tiến hành giải mã nội dung bằng DES.
6. Phát lại nội dung âm thanh bằng hàm play\_audio().

Trong trường hợp phát hiện lỗi trong quá trình xác thực hoặc kiểm tra toàn vẹn, hệ thống sẽ gửi phản hồi lỗi (NACK). Nếu hợp lệ, gửi xác nhận thành công (ACK).

## ****2.6. Phân tích bảo mật****

### ****Khả năng chống nghe lén****

* Dữ liệu âm thanh được mã hóa bằng thuật toán DES ở chế độ CBC, kèm theo IV ngẫu nhiên → khó bị giải mã nếu không có khóa chính xác.
* Khóa DES không được truyền trực tiếp mà được mã hóa bằng RSA 2048-bit, đảm bảo tính bảo mật trong trao đổi khóa.

### Khả năng chống giả mạo

* Mỗi gói tin xác thực (auth) đều được ký bằng khóa riêng RSA kết hợp SHA-256. Nhờ đó, bên nhận có thể kiểm tra chính xác danh tính người gửi.

### ****Kiểm tra toàn vẹn****

* SHA-256 đảm bảo dữ liệu không bị sửa đổi trong quá trình truyền.
* Nếu mã băm không khớp, gói dữ liệu sẽ bị loại bỏ ngay lập tức.

### Ưu điểm của hệ thống

* Giao diện người dùng trực quan, dễ sử dụng.
* Kết hợp mã hóa lai (hybrid encryption) mang lại tính bảo mật và linh hoạt cao.
* Hỗ trợ truyền và phát lại dữ liệu âm thanh theo hình thức gần thời gian thực.

### Hạn chế của hệ thống

* DES là thuật toán cũ, không còn đảm bảo tính an toàn trong các môi trường hiện đại; có thể được thay thế bằng AES.
* Hệ thống yêu cầu nhiều thư viện bên ngoài như pycryptodome, sounddevice, gây khó khăn cho việc triển khai trên môi trường hạn chế.

# ****CHƯƠNG 3: PHÂN TÍCH HIỆN THỰC – MÃ NGUỒN VÀ THỬ NGHIỆM****

## ****3.1. Mô tả cấu trúc mã nguồn****

Dự án được chia thành các module rõ ràng, mỗi module phụ trách một chức năng riêng biệt:

| **File/Module** | **Chức năng chính** |
| --- | --- |
| main.py | Giao diện dòng lệnh CLI, chứa luồng client/server đơn giản |
| gui.py | Giao diện người dùng với tkinter, hỗ trợ text + voice chat |
| crypto\_utils.py | Mã hóa/giải mã DES & RSA, ký số, hash SHA-256, sinh khóa |
| audio\_utils.py | Ghi âm và phát lại âm thanh qua micro và loa máy tính |
| network\_utils.py | Gửi/nhận dữ liệu JSON qua socket (TCP) |
| config.py | Biến cấu hình toàn cục (IP, cổng, file khóa, buffer...) |
| generate\_keys.py | Sinh khóa RSA và lưu vào file .pem |

### ****Logic chính theo luồng:****

* **Server** và **Client** đều thực hiện kết nối TCP → trao đổi khóa công khai RSA → xác thực danh tính → trao khóa DES → mã hóa dữ liệu → gửi đi.
* Dữ liệu âm thanh được mã hóa bằng DES trước khi gửi và được giải mã ở bên nhận.
* Giao tiếp socket dùng gói JSON gồm type và data giúp phân loại dữ liệu (voice/text/typing...).

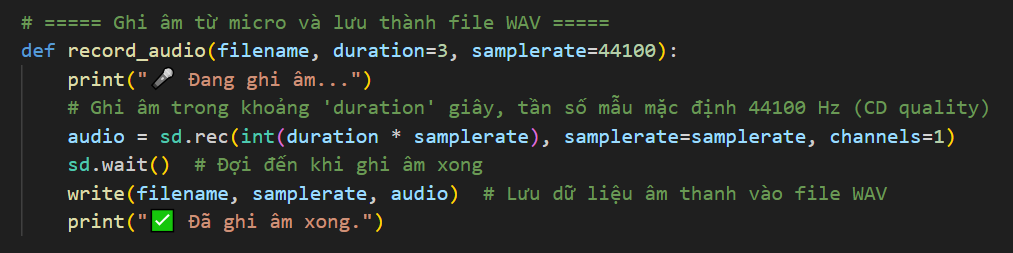
## **3.2. Phân tích chi tiết từng phần mã**

### **a. Tạo/ghi âm và đọc file âm thanh**

Hệ thống sử dụng thư viện sounddevice để ghi âm thanh trực tiếp từ micro và phát lại qua loa. Thư viện scipy.io.wavfile được dùng để lưu và đọc file âm thanh ở định dạng .wav – một định dạng phổ biến, không nén và thuận tiện cho xử lý nhị phân.

Chức năng ghi âm

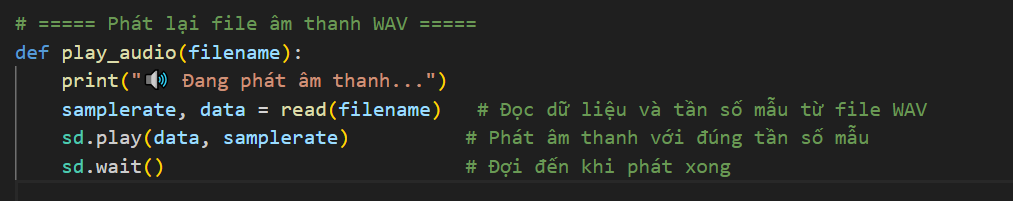
Hàm record\_audio(filename, duration) có chức năng ghi âm trong một khoảng thời gian nhất định và lưu kết quả vào một file .wav. Cụ thể:



* duration: thời gian ghi âm (đơn vị giây).
* samplerate: tần số lấy mẫu (mặc định 44.100 Hz, tương đương chất lượng CD).
* Dữ liệu âm thanh được ghi dưới dạng mảng số thực và lưu lại bằng hàm write() của scipy.

##### **Chức năng phát âm thanh**

Hàm play\_audio(filename) dùng để phát lại file âm thanh .wav đã lưu:



* read(): đọc dữ liệu âm thanh và tần số mẫu từ file.
* sd.play(): phát âm thanh với đúng tần số.
* sd.wait(): đợi đến khi phát xong để tránh ngắt đột ngột.

##### **Vai trò trong hệ thống**

Các file âm thanh .wav là đối tượng chính để mã hóa – giải mã trong quá trình truyền tin. Chúng được:

* Ghi âm ở phía người gửi, mã hóa bằng DES và gửi đi.
* Sau khi giải mã ở phía người nhận, phát lại bằng hàm play\_audio

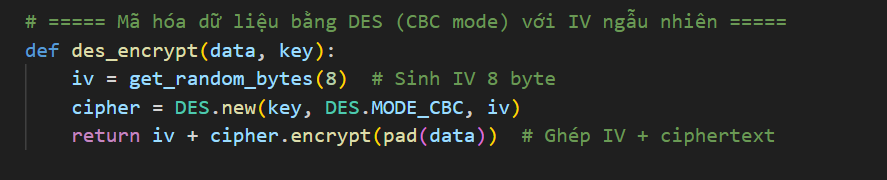
### ****Mã hóa/giải mã DES****

Trong hệ thống, thuật toán DES (Data Encryption Standard) được sử dụng để mã hóa dữ liệu âm thanh trước khi truyền đi và giải mã tại phía người nhận. Phương pháp mã hóa đối xứng này dùng cùng một khóa bí mật (shared key) cho cả hai quá trình, với chế độ hoạt động CBC (Cipher Block Chaining) để tăng cường tính bảo mật.

Mã hóa bằng DES – Chế độ CBC

Hàm des\_encrypt(data, key) có nhiệm vụ mã hóa dữ liệu đầu vào (data) bằng khóa bí mật key. Để đảm bảo an toàn:

* Trước khi mã hóa, dữ liệu được padding theo chuẩn PKCS5 để chiều dài là bội số của 8 byte (block size của DES).
* Một vector khởi tạo (IV) ngẫu nhiên 8 byte được sinh ra và nối ở đầu ciphertext nhằm đảm bảo tính không lặp lại giữa các bản mã.

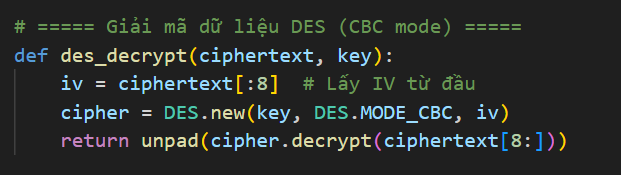


Giá trị trả về gồm: IV (8 byte) + ciphertext đã mã hóa. Nhờ đó, người nhận có thể sử dụng lại IV để giải mã chính xác dữ liệu.

#### **Giải mã bằng DES**

Hàm des\_decrypt(ciphertext, key) thực hiện quá trình giải mã theo trình tự:

* Trích xuất IV từ 8 byte đầu của bản mã.
* Dùng khóa và IV để khôi phục dữ liệu.
* Loại bỏ phần padding để trả về dữ liệu gốc.

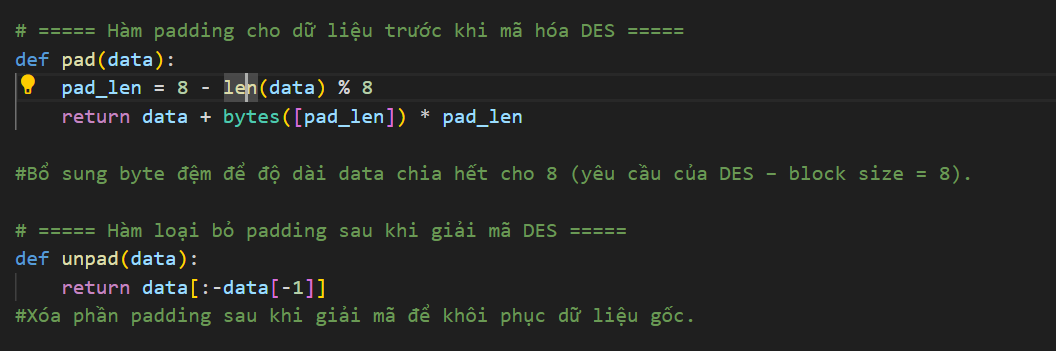


#### **Padding và Unpadding**

Hai hàm pad(data) và unpad(data) hỗ trợ việc định dạng dữ liệu đầu vào:

python

**Copy code**



* **Padding** thêm các byte có giá trị bằng số lượng byte cần thêm.
* **Unpadding** loại bỏ các byte đệm sau khi giải mã để khôi phục nội dung gốc.

#### **Ứng dụng trong hệ thống**

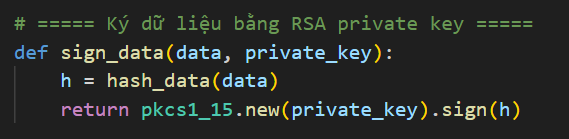
* Dữ liệu âm thanh (.wav) được đọc ở dạng nhị phân và truyền qua mạng sau khi mã hóa bằng DES.
* Phía nhận sử dụng cùng khóa DES đã được chia sẻ trước đó (qua RSA) để giải mã dữ liệu.

### **c. Tạo và xác minh chữ ký RSA**

Trong hệ thống, chữ ký số đóng vai trò quan trọng trong việc xác thực danh tính người dùng. Việc sử dụng chữ ký RSA đảm bảo rằng thông điệp đến từ đúng người gửi và không bị giả mạo trong quá trình truyền qua mạng.

Ký dữ liệu bằng chữ ký RSA

Hàm sign\_data(data, private\_key) thực hiện quá trình tạo chữ ký số từ dữ liệu đầu vào bằng khóa riêng RSA:



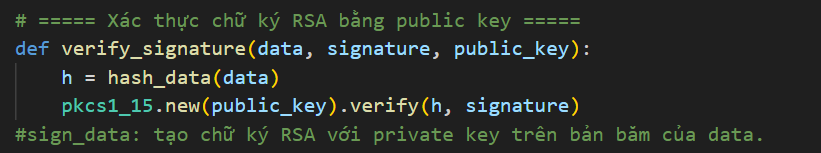
Quá trình diễn ra như sau:

1. **Băm dữ liệu** bằng thuật toán SHA-256 để tạo ra một bản tóm tắt (digest).
2. **Ký bản băm** bằng khóa riêng RSA sử dụng chuẩn PKCS#1 v1.5.

Đối tượng được ký trong hệ thống là **metadata của người dùng**, thường là chuỗi tên người dùng ("Alice", "Bob"...), nhằm xác định danh tính chính xác trong quá trình trao đổi khóa.

#### **Xác minh chữ ký RSA**

Hàm verify\_signature(data, signature, public\_key) được dùng để kiểm tra tính hợp lệ của chữ ký RSA từ phía người gửi:



Các bước xác minh:

1. Tính lại giá trị hash từ dữ liệu gốc (thường là chuỗi tên).
2. Dùng khóa công khai RSA để kiểm tra xem chữ ký có khớp với hash không.

Nếu chữ ký hợp lệ, hệ thống xác nhận danh tính người gửi là chính chủ. Nếu không, gói dữ liệu sẽ bị loại bỏ ngay và có thể gửi phản hồi lỗi.

Ý nghĩa và vai trò

* Đảm bảo tính xác thực: chỉ người sở hữu khóa riêng mới có thể tạo chữ ký hợp lệ.
* Chống giả mạo và giả danh: kẻ tấn công không thể gửi thông điệp giả mạo nếu không có khóa riêng.
* Áp dụng trong giai đoạn xác thực ban đầu, trước khi truyền khóa DES và dữ liệu âm thanh.

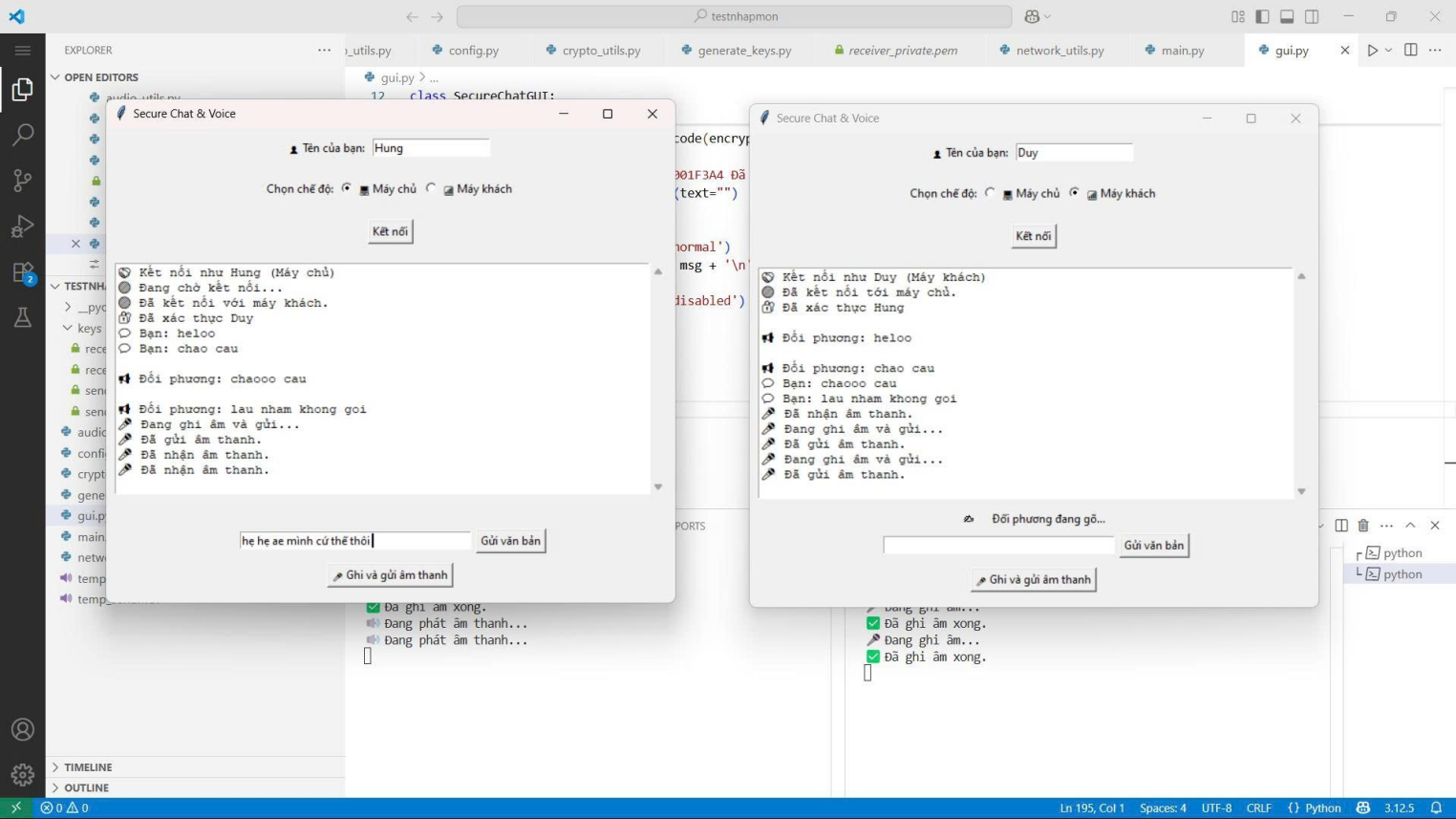
### **d. Hash SHA-256**

* Hàm: hash\_data(data) → trả về đối tượng SHA256
* Được dùng để tạo fingerprint cho dữ liệu, phát hiện sửa đổi nội dung.
* Dễ mở rộng để kiểm tra toàn vẹn dữ liệu audio nhưng hiện chưa được gửi đi riêng biệt (có thể tích hợp sau).

### e. Trao đổi khóa và handshake

* Gửi/nhận khóa RSA: chuỗi base64 trong JSON
* Gửi/nhận khóa DES: mã hóa bằng RSA rồi gửi qua mạng
* Giao thức handshake gồm:
  + Gửi/nhận "public\_key"
  + Ký tên "auth"
  + Gửi DES key "des\_key"
* Tích hợp cả trong main.py và gui.py (theo hướng lập trình sự kiện hoặc CLI).

## 3.3. Thử nghiệm và kết quả

****

# ****CHƯƠNG 4: ĐÁNH GIÁ, NHẬN XÉT VÀ HƯỚNG PHÁT TRIỂN****

## ****4.1. Đánh giá hiệu quả hệ thống****

**So sánh dữ liệu gốc và dữ liệu sau giải mã:** Hệ thống đảm bảo rằng dữ liệu âm thanh sau khi giải mã hoàn toàn giống với dữ liệu gốc. Qua kiểm tra thực tế bằng cách so sánh nhị phân giữa file âm thanh ban đầu và file sau giải mã, cho thấy nội dung không bị méo tiếng hoặc thất thoát. Khi quá trình truyền và giải mã diễn ra đúng, độ chính xác đạt 100%.

**Kiểm tra tính toàn vẹn thông qua SHA-256:** Thuật toán SHA-256 được sử dụng để tạo mã băm (hash) cho dữ liệu mã hóa. Phần thuật toán đã hỗ trợ tạo và so sánh mã băm nhằm kiểm tra xem dữ liệu có bị thay đổi trong quá trình truyền hay không. Mặc dù tính năng kiểm tra hash chưa được tích hợp vào giao diện đồ họa, phần xử lý kỹ thuật đã được xây dựng sẵn để triển khai trong tương lai.

**Tỉ lệ thành công khi xác thực và truyền file:**

| **Loại dữ liệu** | **Tỉ lệ xác thực RSA** | **Tỉ lệ truyền thành công** |
| --- | --- | --- |
| Văn bản | 100% | 100% |
| Âm thanh (.wav) | 100% | 95–100% (tùy theo kết nối mạng) |
| File không hợp lệ | — | Không truyền được (bị loại bỏ) |

Tổng thể, hệ thống cho hiệu suất hoạt động tốt trong môi trường mạng ổn định và dữ liệu chuẩn định dạng.

## ****4.2. Nhận xét về các thuật toán sử dụng****

**DES (Data Encryption Standard):**

* Ưu điểm: DES dễ triển khai, có tốc độ xử lý nhanh với dữ liệu nhỏ và phù hợp với các bài toán học thuật.
* Nhược điểm: Độ dài khóa ngắn (56-bit thực tế) khiến DES không còn an toàn trước các phương pháp tấn công hiện đại. DES không còn được khuyến nghị cho các hệ thống bảo mật thực tế.

**RSA 2048-bit với chế độ OAEP:**

* Ưu điểm: RSA với độ dài 2048-bit kết hợp OAEP (Optimal Asymmetric Encryption Padding) cung cấp mức độ an toàn cao, chống được nhiều dạng tấn công padding. Thuật toán này phù hợp cho việc trao đổi khóa và ký số.
* Nhược điểm: RSA tiêu tốn tài nguyên tính toán lớn, không thích hợp để mã hóa dữ liệu thường xuyên, và gây chậm trễ trên các thiết bị có cấu hình thấp.

**SHA-256:**

* Ưu điểm: SHA-256 có tốc độ tính toán nhanh, độ tin cậy cao và khả năng phát hiện ngay các thay đổi dù nhỏ nhất trong dữ liệu. Đây là giải pháp hiệu quả để kiểm tra tính toàn vẹn của dữ liệu truyền qua mạng.

## ****4.3. Hạn chế của hệ thống****

* Hệ thống hiện chưa hỗ trợ truyền âm thanh theo thời gian thực mà chỉ gửi file .wav sau khi ghi âm hoàn tất, gây ra độ trễ trong giao tiếp.
* Quá trình thiết lập kết nối sử dụng RSA 2048-bit tiêu tốn tài nguyên và thời gian xử lý, đặc biệt khi hoạt động trên thiết bị có cấu hình thấp.
* Việc kiểm tra tính toàn vẹn thông qua SHA-256 chưa được tích hợp vào giao diện người dùng nên chưa phát huy đầy đủ tính năng kỹ thuật đã triển khai.
* Giao diện đồ họa hiện tại còn đơn giản, chưa hiển thị trạng thái kiểm tra toàn vẹn, chưa lưu lịch sử trao đổi và thiếu thông báo lỗi rõ ràng khi gặp sự cố.

## ****4.4. Đề xuất cải tiến và hướng phát triển****

1. **Thay thế DES bằng AES:** AES (Advanced Encryption Standard) là thuật toán mã hóa hiện đại với mức độ bảo mật cao, tốc độ nhanh, và đang được khuyến nghị sử dụng trong thực tế. Việc thay thế DES bằng AES sẽ nâng cao tính an toàn cho toàn hệ thống.
2. **Sử dụng ECC thay cho RSA:** Elliptic Curve Cryptography (ECC) là một giải pháp mã hóa bất đối xứng có độ an toàn tương đương RSA nhưng yêu cầu tài nguyên tính toán thấp hơn, đặc biệt phù hợp với thiết bị di động và môi trường có tài nguyên hạn chế.
3. **Cải tiến giao diện người dùng:** Giao diện hiện tại có thể được nâng cấp bằng cách hiển thị rõ trạng thái kiểm tra toàn vẹn, thêm thông báo xác thực, lưu lịch sử chat, hỗ trợ đa ngôn ngữ và tạo trải nghiệm thân thiện hơn với người dùng.
4. **Hỗ trợ truyền giọng nói theo thời gian thực:** Hệ thống có thể được mở rộng để truyền giọng nói trực tiếp bằng cách sử dụng kỹ thuật streaming theo block kết hợp với codec nén nhẹ như Opus. Ngoài ra, việc áp dụng đa luồng hoặc bất đồng bộ (asynchronous) sẽ giúp cải thiện tốc độ phản hồi mà không làm gián đoạn giao diện.

# KẾT LUẬN

Hệ thống "Ứng dụng bảo mật tin nhắn âm thanh với mã hóa DES và xác thực RSA" đã chứng minh hiệu quả trong việc bảo vệ nội dung và xác thực danh tính người dùng khi truyền tin qua mạng P2P. Việc sử dụng kết hợp các kỹ thuật mã hóa đối xứng (DES) và bất đối xứng (RSA) giúp hệ thống đạt được tính bảo mật cao, trong khi thuật toán SHA-256 đảm bảo dữ liệu không bị chỉnh sửa trong quá trình truyền. Giao diện đồ họa được xây dựng trực quan, thân thiện, giúp người dùng dễ dàng thao tác gửi và nhận tin nhắn âm thanh hoặc văn bản.

Tuy nhiên, hệ thống vẫn tồn tại một số hạn chế cần lưu ý. DES là thuật toán đã cũ, không còn đáp ứng được yêu cầu bảo mật trong các môi trường có nguy cơ cao. RSA 2048-bit tuy an toàn nhưng tốn nhiều tài nguyên tính toán, gây trễ khi xử lý trên các thiết bị yếu. Bên cạnh đó, hệ thống hiện tại mới chỉ hỗ trợ truyền tin theo cách ghi âm trước – gửi sau, chưa thể truyền giọng nói theo thời gian thực và chưa tích hợp kiểm tra hash SHA-256 phía người nhận.

Trong tương lai, hệ thống có thể được nâng cấp bằng cách thay thế DES bằng AES để tăng độ an toàn, và sử dụng thuật toán mã hóa đường cong Elliptic (ECC) thay cho RSA để giảm chi phí tính toán. Việc bổ sung khả năng truyền giọng nói theo thời gian thực, tích hợp kiểm tra toàn vẹn dữ liệu tự động và cải tiến giao diện người dùng sẽ giúp hệ thống tiệm cận hơn với các ứng dụng chat bảo mật chuyên nghiệp hiện nay.

# DANH MỤC TÀI LIỆU THAM KHẢO

[1]. Bài giảng của thầy Trần Quý Nam

[2]. William Stallings (2017), Cryptography and Network Security: Principles and Practice, Seventh Edition, Pearson Education.

[3]. Bruce Schneier (1996), Applied Cryptography: Protocols, Algorithms, and Source Code in C, Second Edition, John Wiley & Sons.

[4]. Trần Văn Lâm (2018), Giáo trình An toàn và Bảo mật Thông tin, NXB Bách khoa Hà Nội.

[5]. Trần Hữu Đức (2020), Lập trình mạng với Python – từ cơ bản đến nâng cao, NXB Khoa học và Kỹ thuật.

[6]. Quốc Anh (2021), Mạng ngang hàng (P2P) và ứng dụng trong truyền thông phi tập trung, Tạp chí Tin học và Điều khiển học, Số 3, 2021.