**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 BỆNH ÁN VỚI XÁC THỰC KÉP**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Lớp |
| 1 | 1771020705 | Phạm Đình Minh Trưởng | 30/04/2005 | CNTT 17-15 |
| 2 | 1771020703 | Đỗ Văn Trường | 28/07/2005 | CNTT 17-15 |
| 3 | 1771020585 | Nguyễn Mạnh Quyết | 22/05/2005 | CNTT 17-15 |

### 

**Hà Nội, năm 2024**

**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 BỆNH ÁN VỚI XÁC THỰC KÉP**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1 | 1771020705 | Phạm Đình Minh Trưởng | 30/04/2005 |  |  |
| 2 | 1771020703 | Đỗ Văn Trường | 28/07/2005 |  |  |
| 3 | 1771020585 | Nguyễn Mạnh Quyết | 22/05/2005 |  |  |

### 

### CÁN BỘ CHẤM THI 1 CÁN BỘ CHẤM THI 2

**Hà Nội, năm 2024**

**LỜI NÓI ĐẦU**

Trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ trong lĩnh vực y tế, việc quản lý, chia sẻ và lưu trữ hồ sơ bệnh án điện tử trở thành một trong những trụ cột quan trọng giúp nâng cao hiệu quả điều trị và chất lượng chăm sóc sức khỏe cho người dân. Tuy nhiên, song song với sự tiện lợi là những thách thức to lớn liên quan đến bảo mật thông tin y tế – một trong những loại dữ liệu nhạy cảm và dễ bị khai thác nhất nếu không có các biện pháp bảo vệ phù hợp.

Các hành vi đánh cắp thông tin bệnh nhân, sửa đổi dữ liệu bệnh án hoặc truy cập trái phép vào hệ thống lưu trữ đang đặt ra yêu cầu cấp thiết về việc triển khai các giải pháp bảo mật toàn diện, kết hợp cả **mã hóa dữ liệu, xác thực người dùng, kiểm tra toàn vẹn và xác minh danh tính**.

Đề tài **“Gửi bệnh án với xác thực kép”** được xây dựng nhằm mô phỏng một mô hình truyền tải dữ liệu bệnh án điện tử giữa hai thực thể: **bác sĩ gửi dữ liệu** và **phòng lưu trữ hồ sơ y tế nhận dữ liệu**. Để đảm bảo dữ liệu không bị rò rỉ, chỉnh sửa hoặc truy cập trái phép trong quá trình truyền tải, hệ thống áp dụng đồng thời nhiều cơ chế bảo mật hiện đại:

**Mã hóa dữ liệu bệnh án bằng thuật toán AES-CBC** để đảm bảo tính bí mật.

**Chữ ký số bằng RSA 2048-bit kết hợp với SHA-512** để xác minh danh tính người gửi.

**Xác thực người nhận bằng mật khẩu được băm SHA-256**, ngăn chặn truy cập trái phép.

**Kiểm tra toàn vẹn bằng SHA-512**, đảm bảo dữ liệu không bị thay đổi trong quá trình truyền.

Thông qua đề tài này, sinh viên không chỉ vận dụng được kiến thức lý thuyết về mật mã học và bảo mật thông tin, mà còn hiểu rõ cách phối hợp nhiều kỹ thuật để giải quyết bài toán bảo mật phức tạp trong một tình huống thực tiễn – cụ thể là môi trường y tế điện tử.

**MỤC LỤC**

[LỜI NÓI ĐẦU 5](#_Toc19668)

[CHƯƠNG 1. ĐẶT VẤN ĐỀ VÀ PHÂN TÍCH YÊU CẦU BÀI TOÁN 12](#_Toc18904)

[1.1. Bối cảnh và lý do chọn đề tài 12](#_Toc12081)

[1.2. Mục tiêu hệ thống 12](#_Toc6504)

[1.3. Phân tích các yêu cầu bảo mật 13](#_Toc15792)

*[1.3.1. Bảo mật nội dung](#_Toc21248)* [13](#_Toc21248)

*[1.3.2. Xác thực hai lớp](#_Toc19405)* [13](#_Toc19405)

*[1.3.3. Kiểm tra toàn vẹn](#_Toc14746)* [14](#_Toc14746)

*[1.3.4. Phản hồi xác nhận](#_Toc8257)* [14](#_Toc8257)

[CHƯƠNG 2. MÔ TẢ THUẬT TOÁN 15](#_Toc29207)

[2.1. Luồng xử lý tổng thể 15](#_Toc26982)

[2.2. Mô tả các thuật toán sử dụng 16](#_Toc625)

*[2.1.1. Mã hoá nội dung:](#_Toc10776)* [16](#_Toc10776)

[1. Khái niệm mở rộng 16](#_Toc19388)

[2. Vai trò trong đề tài 16](#_Toc2613)

[3. Thuật toán sử dụng: AES-CBC 17](#_Toc23823)

[4. Quy trình kỹ thuật cụ thể 17](#_Toc4190)

*[2.1.2. Trao đổi khoá:](#_Toc11976)* [18](#_Toc11976)

[1. Khái niệm 18](#_Toc14011)

[2. Vai trò trong hệ thống 19](#_Toc4288)

[3. Thuật toán sử dụng: SHA-256 24](#_Toc5419)

[4. Quy trình xác thực người nhận 25](#_Toc8113)

[CHƯƠNG 3. XÂY DỰNG VÀ TRIỂN KHAI 26](#_Toc9085)

[3.1.Công cụ sử dụng 26](#_Toc5041)

[3.2. Thiết kế và xây dựng 29](#_Toc24384)

*[3.2.1. Kiến trúc hệ thống](#_Toc15170)* [29](#_Toc15170)

*[3.2.2. Sender](#_Toc29086)* [29](#_Toc29086)

*[3.2.3. Receiver](#_Toc7184)* [30](#_Toc7184)

[3.2.4. Quy trình hoạt động 32](#_Toc28184)

[3.3. Kết quả thực nghiệm 32](#_Toc5652)

*[3.3.1. Môi trường thực nghiệm](#_Toc19797)* [32](#_Toc19797)

*[3.3.2. Quy trình thực nghiệm](#_Toc15779)* [32](#_Toc15779)

[3.4. Đánh giá 34](#_Toc32632)

*[3.4.1. Tính bảo mật](#_Toc17132)* [34](#_Toc17132)

*[3.4.2. Hiệu quả thực thi](#_Toc23076)* [35](#_Toc23076)

*[3.4.3. Tính chính xác và ổn định](#_Toc14963)* [35](#_Toc14963)

*[3.4.4. Hạn chế](#_Toc1937)* [35](#_Toc1937)

[KẾT LUẬN 36](#_Toc30278)

[DANH MỤC TÀI LIỆU THAM KHẢO 38](#_Toc1892)

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

* 1. **Bối cảnh và lý do chọn đề tài**

Trong thời đại số hóa, dữ liệu y tế đang ngày càng trở thành tài sản quan trọng trong hệ thống chăm sóc sức khỏe. Những thông tin này không chỉ nhạy cảm mà còn mang tính cá nhân cao, bao gồm tiền sử bệnh, đơn thuốc, chẩn đoán và liệu trình điều trị. Bất kỳ rò rỉ hoặc giả mạo nào cũng có thể gây hậu quả nghiêm trọng đến tính mạng bệnh nhân, danh tiếng bác sĩ cũng như uy tín của cơ sở y tế.

Ở các bệnh viện hiện đại, luồng công việc số hoá yêu cầu các bác sĩ phải gửi thông tin bệnh án điện tử (EHR – Electronic Health Record) từ nơi khám chữa bệnh đến bộ phận lưu trữ hoặc các hệ thống trung tâm dữ liệu. Việc truyền tải này thường diễn ra qua mạng nội bộ hoặc Internet, trong đó tồn tại nhiều mối đe doạ bảo mật như:

* **Nghe lén :** Kẻ tấn công có thể chặn và đọc thông tin đang truyền.
* **Giả mạo :** Một thực thể giả danh bác sĩ để gửi thông tin sai lệch.
* **Thay đổi dữ liệu :** Dữ liệu bị chỉnh sửa trước khi đến nơi lưu trữ.
* **Truy cập trái phép:** Nhân viên không có quyền vẫn có thể xem nội dung bệnh án.

Vì vậy, để đảm bảo tính **bảo mật, toàn vẹn và xác thực** của thông tin y tế, đề tài **“Gửi bệnh án với xác thực kép”** được xây dựng nhằm mô phỏng một hệ thống truyền tải file nhạy cảm giữa bác sĩ và phòng lưu trữ, tuân thủ các nguyên tắc an toàn thông tin y tế.

* 1. **Mục tiêu hệ thống**

Hệ thống được xây dựng nhằm hướng đến các mục tiêu bảo mật sau:

**Bảo mật nội dung bệnh án:** Toàn bộ file medical\_record.txt phải được mã hoá bằng thuật toán AES-CBC trước khi truyền, nhằm ngăn chặn việc đọc trộm nếu dữ liệu bị chặn lại.

**Xác thực danh tính người gửi:** Mỗi file gửi đi đều đi kèm **chữ ký số RSA**, xác minh danh tính bác sĩ gửi file và đảm bảo tính không thể chối bỏ (non-repudiation).

**Xác thực quyền truy cập của người nhận:** Nhân viên lưu trữ phải nhập đúng **mật khẩu đã băm SHA-256**, nhằm đảm bảo chỉ người được ủy quyền mới giải mã được dữ liệu.

**Đảm bảo toàn vẹn dữ liệu:** Hệ thống sử dụng **SHA-512** để kiểm tra toàn vẹn nội dung truyền (bao gồm IV và ciphertext). Nếu bất kỳ thành phần nào bị thay đổi, dữ liệu sẽ bị từ chối.

**Phản hồi trạng thái xử lý:** Sau khi kiểm tra tất cả tiêu chí trên, hệ thống gửi về kết quả ACK (thành công) hoặc NACK (từ chối), giúp người gửi biết được kết quả truyền file.

* 1. **Phân tích các yêu cầu bảo mật**
     1. ***Bảo mật nội dung***

Dữ liệu bệnh án chứa thông tin riêng tư, cần được mã hóa ngay từ phía người gửi. Hệ thống sử dụng:

* **AES-CBC** (Advanced Encryption Standard – Cipher Block Chaining): thuật toán mã hóa khối phổ biến và an toàn.
* **Khóa phiên (Session Key)**: sinh ngẫu nhiên cho mỗi lần truyền file.
* **IV (Initialization Vector):** giúp mã hóa không lặp lại.

=> Giải pháp này đảm bảo rằng nội dung file sẽ không thể bị đọc được nếu không có khóa giải mã phù hợp.

* + 1. ***Xác thực hai lớp***

**Lớp 1 – Chữ ký số RSA 2048-bit (OAEP + SHA-512):**

Người gửi ký metadata (tên file + thời gian + ID bệnh án).

Người nhận dùng khóa công khai để xác minh chữ ký.

**Lớp 2 – Mật khẩu SHA-256:**

Người gửi gửi kèm mật khẩu đã băm.

Người nhận đối chiếu với mật khẩu hệ thống để xác thực quyền truy cập.

=> Cơ chế xác thực kép giúp hệ thống chống lại cả việc giả mạo bác sĩ lẫn truy cập trái phép từ nội bộ.

* + 1. ***Kiểm tra toàn vẹn***

Dữ liệu mã hoá sẽ được kiểm tra toàn vẹn bằng cách tính **SHA-512** của IV || ciphertext.

Bất kỳ thay đổi nào dù nhỏ nhất trong ciphertext hay IV đều khiến hash không khớp => file bị từ chối.

=> Bảo đảm dữ liệu không bị thay đổi hay can thiệp trong quá trình truyền.

* + 1. ***Phản hồi xác nhận***

Nếu tất cả kiểm tra (chữ ký, mật khẩu, toàn vẹn) đều hợp lệ → gửi **ACK**.

Nếu bất kỳ phần nào không hợp lệ → gửi **NACK** với mô tả lỗi cụ thể.

→ Người gửi có thể biết trạng thái xử lý file và thực hiện hành động tiếp theo (gửi lại hoặc cảnh báo).

**CHƯƠNG 2. MÔ TẢ THUẬT TOÁN**

**2.1. Luồng xử lý tổng thể**

Quá trình gửi bệnh án giữa bác sĩ và phòng lưu trữ được chia thành ba bước chính:

**Handshake:** Bắt đầu phiên truyền dữ liệu bằng thông điệp "Hello!" từ người gửi, nếu người nhận sẵn sàng sẽ phản hồi "Ready!".

**Xác thực và trao khoá:**

+ Người gửi tạo **metadata** (gồm tên file, thời gian gửi và ID bệnh án), sau đó ký số bằng **RSA 2048-bit + SHA-512**.

+ Đồng thời, mật khẩu truy cập của người nhận được băm bằng **SHA-256** và gửi kèm trong gói tin.

**+ Session Key AES** (dùng để mã hoá nội dung file) được mã hoá bằng khoá công khai của người nhận (RSA-OAEP + SHA-512).

**Mã hoá và kiểm tra toàn vẹn:**

+ File medical\_record.txt được mã hoá bằng **AES-CBC**, sử dụng Session Key và IV ngẫu nhiên.

+ Toàn bộ IV || ciphertext được băm bằng **SHA-512** để kiểm tra toàn vẹn.

+ Gói tin cuối cùng bao gồm: IV, dữ liệu mã hoá, chữ ký số, hash SHA-512 và mật khẩu băm.

Sau khi nhận, người nhận sẽ lần lượt kiểm tra: **tính toàn vẹn**, **chữ ký**, và **mật khẩu**. Nếu mọi thứ hợp lệ, hệ thống sẽ giải mã file và gửi **ACK** ngược lại. Ngược lại, nếu có lỗi, hệ thống phản hồi **NACK** và từ chối xử lý file.

**2.2. Mô tả các thuật toán sử dụng**

### *****2.1.1. Mã hoá nội dung:*****

### **1. Khái niệm mở rộng**

**Mã hóa dữ liệu (Data Encryption)** là kỹ thuật chuyển đổi dữ liệu từ định dạng rõ ràng (**plaintext**) sang định dạng bí mật (**ciphertext**) bằng một thuật toán mã hóa kết hợp với **khóa mật mã (key)**. Chỉ những người có khóa giải mã hợp lệ mới có thể đọc và sử dụng dữ liệu.

Có hai loại mã hóa chính:

**Mã hóa đối xứng (Symmetric Encryption)**: Dùng **cùng một khóa** để mã hóa và giải mã (ví dụ: AES).

**Mã hóa bất đối xứng (Asymmetric Encryption)**: Dùng **cặp khóa công khai và bí mật** để mã hóa và giải mã (ví dụ: RSA).

### **2. Vai trò trong đề tài**

Trong đề tài **“Gửi bệnh án với xác thực kép”**, mã hóa dữ liệu đảm nhận vai trò quan trọng nhằm:

**Bảo mật nội dung bệnh án** trong file medical\_record.txt.

**Ngăn chặn truy cập trái phép** nếu dữ liệu bị chặn trong quá trình truyền.

**Tuân thủ quy định bảo mật y tế (HIPAA, ISO/IEC 27001,...)** về bảo vệ thông tin cá nhân và sức khỏe (PHI – Protected Health Information).

### **3. Thuật toán sử dụng: AES-CBC**

Chuẩn mã hóa do NIST công bố, thay thế DES.

Hoạt động theo nguyên lý khối (block cipher), thường dùng độ dài khóa 128, 192 hoặc 256 bit.

Trong đề tài, dùng **AES-256**, nghĩa là:

Mỗi khối dữ liệu là 128 bit (16 byte).

Dùng khóa 256 bit (32 byte).

Mỗi khối dữ liệu **được XOR với khối mã hóa trước đó**, sau đó mới mã hóa → làm cho dữ liệu giống nhau cũng cho ciphertext khác nhau nếu IV khác nhau.

Cần **IV (Initialization Vector)** để khởi tạo mã hóa khối đầu tiên.

### **4. Quy trình kỹ thuật cụ thể**

**Bước 1: Tạo khóa phiên (Session Key)**

Tạo một khóa AES-256 ngẫu nhiên để dùng tạm thời trong phiên làm việc.

Khóa này **không được gửi trực tiếp**, mà sẽ được mã hóa bằng RSA và gửi trong bước trao đổi khóa

**Bước 2: Tạo IV (Initialization Vector)**

Sinh ngẫu nhiên 16 byte để dùng làm IV cho CBC.

IV **không cần giữ bí mật**, nhưng cần gửi kèm để người nhận giải mã đúng

**Bước 3: Mã hóa dữ liệu bệnh án**

Đọc nội dung file medical\_record.txt.

Áp dụng **padding (thêm byte)** nếu độ dài không chia hết cho 16 byte (theo chuẩn PKCS#7).

Mã hóa bằng AES-CBC với khóa phiên và IV.

**Bước 4: Gửi dữ liệu**

ciphertext được mã hóa Base64 để gói thành chuỗi JSON.

IV cũng mã hóa Base64 để gửi cùng.

### ***2.1.2.* ***Trao đổi khoá:*****

### 1. ****Khái niệm****

#### **a. **Mã hóa khóa (Key Encryption)****

Là quá trình sử dụng **mã hóa bất đối xứng (asymmetric encryption)**, cụ thể là thuật toán **RSA**, để bảo vệ **khóa phiên (Session Key)** – khóa dùng trong quá trình mã hóa nội dung bệnh án bằng AES.

Việc mã hóa khóa này đảm bảo rằng **chỉ người nhận (phòng lưu trữ)**, người đang giữ **khóa bí mật (private key)** tương ứng, mới có thể giải mã và truy cập được dữ liệu thực.

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

Chữ ký số là một kỹ thuật bảo mật dùng để:

**Xác thực danh tính** người gửi (ở đây là bác sĩ).

**Đảm bảo toàn vẹn**: nếu dữ liệu bị thay đổi, chữ ký sẽ không còn hợp lệ.

**Ngăn chối bỏ**: bác sĩ không thể phủ nhận đã gửi dữ liệu.

Chữ ký được tạo bằng cách **băm nội dung cần ký (metadata)** bằng hàm băm mạnh (SHA-512), sau đó mã hóa bằng **private key RSA** của người gửi.

### ****Vai trò trong hệ thống****

|  |  |
| --- | --- |
| Yếu tố bảo mật cần thiết | Được đảm bảo bởi |
| Bảo vệ khóa mã hóa dữ liệu | RSA 2048-bit với chế độ OAEP |
| Xác thực người gửi | Chữ ký số RSA (private key của bác sĩ) |
| Toàn vẹn metadata | So sánh hash SHA-512 & chữ ký số |
| Chống giả mạo / sửa đổi | Chữ ký số không khớp nếu dữ liệu bị đổi |

Trong hệ thống gửi bệnh án:

**Session Key** AES được mã hóa bằng **public key** của phòng lưu trữ hồ sơ.

Bác sĩ ký **metadata** (gồm tên file, timestamp và ID bệnh án) để xác thực danh tính.

Phòng lưu trữ dùng **public key của bác sĩ** để kiểm tra chữ ký.

### **3. **Thuật toán sử dụng****

#### ****a.RSA 2048-bit****

Là thuật toán mã hóa bất đối xứng phổ biến nhất hiện nay.

Sử dụng một cặp khóa:

**Public key** để mã hóa hoặc xác minh chữ ký.

**Private key** để giải mã hoặc tạo chữ ký.

Trong đề tài, RSA được dùng để:

Mã hóa Session Key.

Tạo và xác minh chữ ký số.

#### ****b.OAEP (Optimal Asymmetric Encryption Padding)****

Là một phương pháp đệm an toàn hơn các kỹ thuật padding cũ (như PKCS#1 v1.5).

Được dùng trong RSA khi mã hóa Session Key để:

Bảo vệ khỏi tấn công chọn ciphertext.

Kết hợp với **SHA-512** làm hàm băm tăng tính an toàn.

#### ****c.SHA-512****

Hàm băm mạnh, tạo ra kết quả 512 bit (64 byte).

Dùng để tạo dấu vân tay cho metadata trước khi ký.

Bất kỳ thay đổi nhỏ nào trong metadata sẽ dẫn tới giá trị băm khác hoàn toàn.

### **4. **Quy trình xử lý****

#### ****A. Phía người gửi (Bác sĩ)****

**Tạo metadata**

Gồm: Tên file, Timestamp, ID bệnh án.

Ví dụ: "medical\_record.txt|2025-06-26T14:30|123456"

**Tạo chữ ký số**

Băm metadata bằng **SHA-512**.

Ký bản băm bằng **private key RSA** của bác sĩ.

**Mã hóa khóa phiên AES (Session Key)**

Sinh Session Key (256-bit) ngẫu nhiên.

Mã hóa bằng **public key của phòng lưu trữ** với **RSA-OAEP-SHA512**.

#### ****B.Phía người nhận (Phòng lưu trữ)****

**Giải mã khóa phiên**

Dùng **private key** để giải RSA-OAEP → thu được Session Key.

**Xác minh chữ ký**

Dùng **public key của bác sĩ**.

Tính lại hash SHA-512 từ metadata nhận được

So sánh kết quả giải mã chữ ký với hash nhận được.

**Nếu chữ ký hợp lệ**  
➤ Xác nhận metadata **không bị thay đổi**, người gửi **hợp lệ**.  
➤ Sử dụng Session Key để **giải mã file bệnh án**.

### *****2.1.3. Kiểm tra toàn vẹn:*****

### **1. **Khái niệm****

**Kiểm tra toàn vẹn dữ liệu (Data Integrity Check)** là quá trình sử dụng các **thuật toán băm một chiều** để xác định xem dữ liệu có bị **thay đổi, mất mát hoặc chỉnh sửa** trong quá trình truyền hoặc lưu trữ hay không.

Trong bối cảnh bảo mật thông tin, đặc biệt là với dữ liệu y tế nhạy cảm, đảm bảo tính toàn vẹn là bắt buộc để ngăn chặn:

Tấn công chỉnh sửa nội dung (ví dụ: thay đổi bệnh án).

Tác động không mong muốn đến file (ví dụ: lỗi đường truyền, thay IV, chỉnh sửa ciphertext...).

### **2. **Vai trò trong đề tài****

Trong hệ thống **Gửi bệnh án với xác thực kép**, dữ liệu bệnh án (medical\_record.txt) sau khi được mã hóa bằng AES-CBC sẽ được gửi đi kèm với mã kiểm tra toàn vẹn.

Kiểm tra toàn vẹn giúp:

**Xác minh dữ liệu mã hóa chưa bị chỉnh sửa** trên đường truyền.

**Ngăn chặn tấn công chỉnh sửa ciphertext/IV** nhằm phá giải dữ liệu.

**Từ chối giải mã nếu dữ liệu không còn nguyên vẹn**, tránh lỗi và rò rỉ thông tin.

### ****Thuật toán sử dụng: SHA-512****

Là một thuật toán **băm mật mã một chiều** trong họ SHA-2, do NIST phát triển.

Đầu ra là một chuỗi **512 bit (64 byte)**.

Có tính chất:

**Không thể đảo ngược** (one-way).

**Không có va chạm** (collision-resistant).

**Đầu ra thay đổi hoàn toàn nếu đầu vào thay đổi 1 byte**

Mạnh hơn SHA-256 trong chống dò tìm hoặc va chạm.

Dữ liệu nhạy cảm như bệnh án đòi hỏi hàm băm có độ dài lớn.

Dễ kết hợp với hệ thống ký số RSA-2048, vốn cũng dùng SHA-512 trong padding (OAEP).

1. ****Ưu điểm của SHA-512****

|  |  |
| --- | --- |
| **Đặc điểm** | **Giá trị** |
| Độ dài đầu ra | 512 bit (64 byte) |
| Mức độ bảo mật | Cao hơn SHA-256 |
| Khả năng phát hiện thay đổi dữ liệu | Rất tốt |
| Tốc độ xử lý | Tối ưu trên kiến trúc 64-bit |
| Ứng dụng rộng rãi | Dùng trong SSL, chữ ký số... |

1. **Phân tích bảo mật**

|  |  |
| --- | --- |
| Rủi ro nếu không kiểm tra toàn vẹn | Giải pháp SHA-512 mang lại |
| Dữ liệu bị chỉnh sửa trên mạng | Phát hiện ngay qua sai lệch giá trị hash |
| Ciphertext bị thay đổi một byte | Hash khác hoàn toàn → từ chối giải mã |
| IV bị giả mạo hoặc đổi | Dễ phát hiện sai → ngăn tấn công padding |
| Truyền file qua mạng không tin cậy | Bảo vệ tính chính xác & nguyên bản dữ liệu |

### *****2.1.4. Xác thực người nhận:*****

### **1. **Khái niệm****

**Xác thực (Authentication)** là quá trình đảm bảo rằng một người hoặc một hệ thống thực sự là **người mà họ tuyên bố là**.

Trong hệ thống “**Gửi bệnh án với xác thực kép**”, xác thực đóng vai trò quan trọng trong việc:

**Xác minh quyền truy cập**: đảm bảo chỉ người nhận hợp lệ (nhân viên lưu trữ được ủy quyền) mới được giải mã dữ liệu.

**Bảo vệ chống truy cập trái phép**: ngăn chặn các đối tượng trung gian (kẻ tấn công) giả danh người nhận.

**Tăng cường an ninh**: là lớp bảo vệ thứ hai, sau khi dữ liệu đã được mã hóa bằng AES và chữ ký số đã xác minh người gửi.

Hệ thống áp dụng xác thực người nhận thông qua **mật khẩu được băm bằng thuật toán SHA-256**.

### ****Vai trò trong hệ thống gửi bệnh án****

|  |  |
| --- | --- |
| **Mục tiêu bảo mật** | **Cách thực hiện** |
| Xác minh người nhận hợp lệ | So sánh hash SHA-256 của mật khẩu với giá trị lưu trữ |
| Bảo vệ quyền truy cập | Chỉ người nắm mật khẩu hợp lệ mới truy cập được dữ liệu |
| Ngăn chặn truy cập trái phép | Từ chối giải mã nếu mật khẩu sai |
| Không để lộ mật khẩu thật | Chỉ truyền giá trị băm, không truyền mật khẩu thô |

### **3. **Thuật toán sử dụng: SHA-256****

#### SHA-256 (Secure Hash Algorithm – 256 bit)

Là một thuật toán băm mật mã trong họ SHA-2, do NIST phát triển.

Tạo ra chuỗi băm có độ dài cố định 256 bit (32 byte).

**Không thể đảo ngược**: từ giá trị băm, không thể tìm ra mật khẩu gốc.

**Nhạy cảm với thay đổi**: chỉ cần thay đổi 1 ký tự đầu vào, giá trị băm đầu ra sẽ thay đổi hoàn toàn.

#### Ưu điểm khi dùng SHA-256:

**Bảo mật cao**, chưa bị phá bởi tấn công thực tiễn.

**Dễ triển khai**, có thư viện hỗ trợ trong hầu hết các ngôn ngữ lập trình.

**Tốc độ nhanh**, phù hợp cho hệ thống thời gian thực.

### ****Quy trình xác thực người nhận****

|  |  |
| --- | --- |
| **Ưu điểm** | **Hạn chế và lưu ý** |
| Không lưu/truyền mật khẩu gốc | Dễ bị tấn công từ điển nếu mật khẩu quá đơn giản |
| Có thể lưu trữ và kiểm tra nhanh chóng | Nên thêm **salt** để chống tấn công rainbow table |
| Không cần truyền khóa bí mật, chỉ dùng hash | Chỉ xác thực, không mã hóa nội dung |
| Phù hợp cho hệ thống xác thực đơn giản, hiệu quả cao | Không thay thế được các kỹ thuật bảo mật khác |

**CHƯƠNG 3.** XÂY DỰNG VÀ TRIỂN KHAI

**3.1.Công cụ sử dụng**

- Hệ điều hành:

Windows 10 hoặc Windows 11

- Trình soạn thảo mã nguồn:

Visual Studio Code (hoặc có thể sử dụng PyCharm, Sublime Text…)

- Ngôn ngữ lập trình:

Python 3.x

- Thư viện và module sử dụng:

+ Tkinter:

Dùng để xây dựng giao diện người dùng (GUI) cho cả bên gửi và bên nhận.Hỗ trợ các thành phần như cửa sổ, nút bấm, nhập mật khẩu, chọn file, hiển thị tiến trình, thông báo lỗi/thành công.

- os:

Xử lý các thao tác với file, đường dẫn, kiểm tra sự tồn tại của file, tạo file mới, v.v.

- base64:

+ Mã hóa/giải mã dữ liệu nhị phân (ví dụ: khóa, chữ ký, dữ liệu mã hóa) thành chuỗi để lưu trữ hoặc truyền tải dễ dàng qua file JSON.

- json:

+ Xử lý dữ liệu dạng JSON, dùng để đóng gói và giải nén gói tin bệnh án (packet).

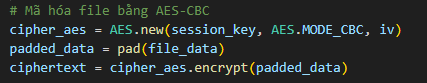
- threading:

+ Hỗ trợ thực thi đa luồng, giúp giao diện không bị treo khi thực hiện các thao tác nặng như mã hóa, giải mã, gửi/nhận file.

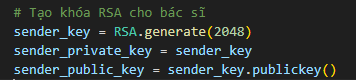
- PyCryptodome:

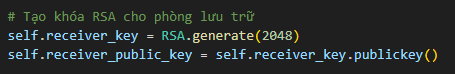
+ Thư viện mã hóa mạnh mẽ, cung cấp các thuật toán:

+ AES (Advanced Encryption Standard): Mã hóa đối xứng, dùng để mã hóa nội dung bệnh án.

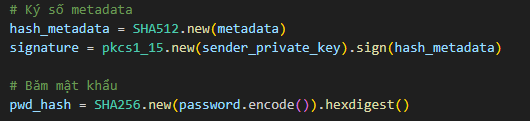


+ RSA: Mã hóa bất đối xứng, dùng để mã hóa khóa phiên (session key) và tạo/chứng thực chữ ký số.





+ SHA-256, SHA-512: Thuật toán băm, dùng để kiểm tra tính toàn vẹn dữ liệu và xác thực mật khẩu.



+ PKCS1\_OAEP, pkcs1\_15: Các chuẩn padding và chữ ký số cho RSA.

hashlib hoặc Crypto.Hash:

+ Băm dữ liệu, kiểm tra toàn vẹn file, xác thực mật khẩu.

- time:

+ Lấy thời gian thực (timestamp) để ghi nhận thời điểm gửi/nhận bệnh án.

tkinter.filedialog, tkinter.messagebox, tkinter.ttk:

+ Các module phụ trợ cho giao diện: chọn file, hiển thị thông báo, thanh tiến trình.

PIL (Pillow):

(Chỉ sử dụng nếu có xử lý ảnh, ví dụ đóng dấu ảnh, nhưng trong mã hiện tại không dùng đến.)

- Công cụ quản lý gói:

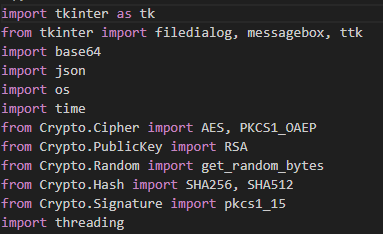
pip (dùng để cài đặt các thư viện bên ngoài như pycryptodome, pillow…)

+ Công cụ quản lý mã nguồn:

+ Git (giúp lưu trữ, quản lý phiên bản mã nguồn, làm việc nhóm hiệu quả)

- Công cụ đóng gói ứng dụng:

+ PyInstaller (nếu muốn đóng gói thành file .exe để chạy trên máy không cần cài Python)



*Hình 3.1.1: Hình ảnh về các thư viện sử dụng*

## **3.2. Thiết kế và xây dựng**

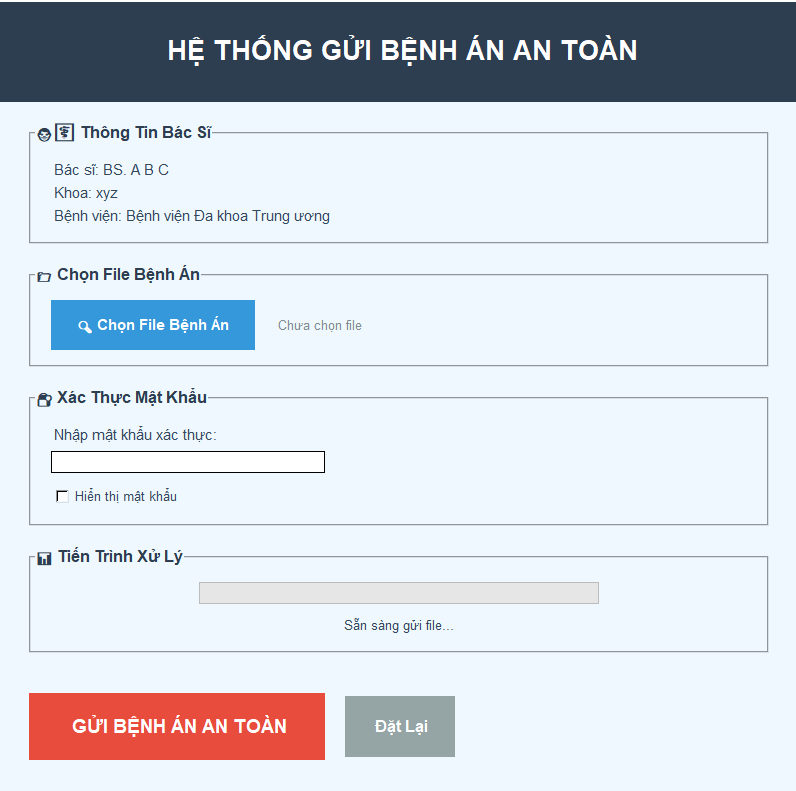
### ***3.2.1. Kiến trúc hệ thống***

Hệ thống gồm hai thành phần chính:

|  |  |
| --- | --- |
| **Thành phần** | **Vai trò chính** |
| Sender (Người gửi) | Giao diện nhập, chọn file bệnh án, mã hóa, ký số và gửi file |
| Receiver (Người nhận) | Nhận file, giải mã, xác minh chữ ký, khôi phục nội dung bệnh án |

Hệ thống hoạt động theo mô hình client–client (gửi/nhận file qua giao diện, không dùng server trung gian).

### ***3.2.2. Sender***



*Hình 3.2.1: Hình ảnh giao diện Sender*

- Giao diện Sender: Cho phép người dùng chọn file bệnh án (dạng text), nhập mật khẩu xác thực, và gửi file đi.

- Chức năng chính:

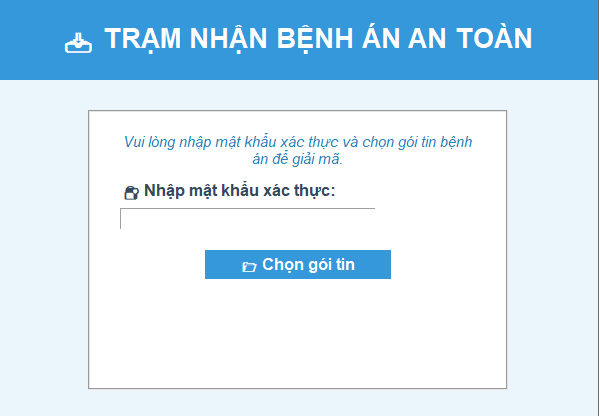
Tiếp nhận file bệnh án từ người dùng: Người dùng chọn file text (ví dụ: medical\_record.txt).

Xử lý bảo mật cho file:

|  |  |
| --- | --- |
| **Bước** | **Mô tả** |
| Tạo metadata | Ghi nhận thông tin file, thời gian gửi, ID hồ sơ |
| Sinh khóa phiên (Session Key) | Tạo khóa AES ngẫu nhiên để mã hóa file |
| Mã hóa file | Mã hóa nội dung file bằng AES-CBC |
| Băm file | Tính hash SHA-512 để kiểm tra toàn vẹn |
| Ký số metadata | Dùng private key RSA để ký metadata |
| Mã hóa khóa phiên | Mã hóa session key bằng public key của người nhận (RSA) |
| Đóng gói dữ liệu | Đóng gói các trường: file mã hóa, hash, chữ ký, metadata, khóa mã hóa, public key, ... vào file JSON |

Gửi file đã mã hóa: Lưu gói tin vào file packet.json để chuyển cho người nhận.

### ***3.2.3. Receiver***



*Hình 3.2.2: Hình ảnh giao diện Receiver*

- Giao diện Receiver: Cho phép người nhận chọn file gói tin (packet.json), nhập mật khẩu xác thực để giải mã và xác minh.

- Chức năng chính:

Nhận file gói tin từ người gửi: Người nhận chọn file packet.json vừa nhận được.

Giải mã và xác minh:

|  |  |
| --- | --- |
| **Bước** | **Mô tả** |
| Kiểm tra mật khẩu | So sánh hash mật khẩu nhập vào với hash lưu trong gói tin |
| Xác minh chữ ký số | Dùng public key của người gửi để xác minh chữ ký metadata |
| Kiểm tra toàn vẹn | So sánh hash SHA-512 của file mã hóa với hash lưu trong gói tin |
| Giải mã khóa phiên | Dùng private key của người nhận để giải mã session key |
| Giải mã file | Dùng session key để giải mã nội dung file bệnh án |
| Xuất file | Lưu file bệnh án đã giải mã ra file text mới |

### **3.2.4. Quy trình hoạt động**

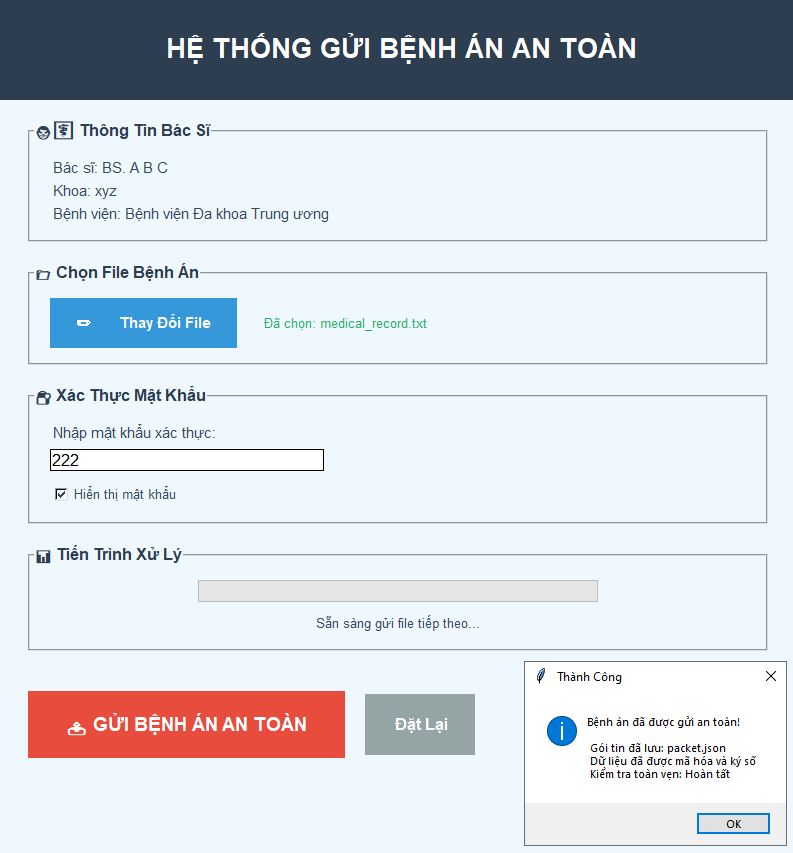
|  |  |  |
| --- | --- | --- |
| Bước | Gửi | Nhận |
| 1 | Chọn file, nhập mật khẩu | Nhận gói tin |
| 2 | Tạo metadata, sinh khóa | Kiểm tra mật khẩu |
| 3 | Mã hóa file, băm, ký số | Xác minh chữ ký |
| 4 | Mã hóa khóa phiên | Kiểm tra toàn vẹn |
| 5 | Đóng gói, gửi file | Giải mã khóa, giải mã file |
| 6 |  | Xuất file bệnh án |

## **3.3. Kết quả thực nghiệm**

### ***3.3.1. Môi trường thực nghiệm***

|  |  |
| --- | --- |
| **Hạng mục** | **Cấu hình** |
| Hệ điều hành | Windows 10 64-bit |
| Ngôn ngữ lập trình | Python 3.10 |
| Thư viện mã hóa | pycryptodome, hashlib, base64 |
| Công cụ test | Giao diện Tkinter |

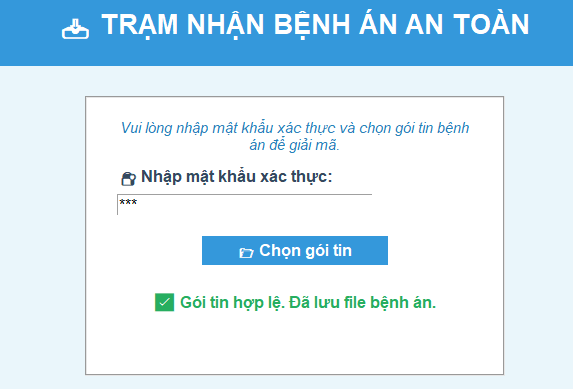
### ***3.3.2. Quy trình thực nghiệm***



- Phía người gửi:

Mở giao diện gửi, chọn file bệnh án, nhập mật khẩu, nhấn gửi.

Kết quả: File được mã hóa, ký số, lưu thành packet.json.



- Phía người nhận:

Mở giao diện nhận nhập mật khẩu, chọn packet.json, , nhấn giải mã.

Kết quả: File bệnh án được giải mã, xác minh chữ ký thành công, xuất ra file text mới.

## **3.4. Đánh giá**

### ***3.4.1. Tính bảo mật***

|  |  |  |
| --- | --- | --- |
| **Yếu tố** | **Phương pháp áp dụng** | **Đảm bảo** |
| Bảo mật dữ liệu | Mã hóa bằng AES-CBC | Ẩn nội dung khỏi bên thứ ba |
| Bảo mật khóa | Mã hóa session key bằng RSA | Chỉ người nhận mới giải mã được |
| Toàn vẹn dữ liệu | Hash SHA-512 + chữ ký số | Phát hiện mọi chỉnh sửa |
| Xác thực người gửi | RSA digital signature | Tránh giả mạo nguồn gửi |
| Chống chối bỏ | Dùng private key ký số | Người gửi không thể phủ nhận đã gửi |

### ***3.4.2. Hiệu quả thực thi***

|  |  |
| --- | --- |
| **Kịch bản** | **Kết quả** |
| Xử lý file < 500KB | Gửi – nhận – giải mã < 2 giây |
| Kiểm tra chữ ký số | Thành công nếu đúng khóa |
| Trường hợp lỗi chữ ký | File không giải mã, cảnh báo “file bị thay đổi” |
| Tác động bảo mật | Không làm giảm chất lượng file quá nhiều |

### ***3.4.3. Tính chính xác và ổn định***

- Hệ thống hoạt động ổn định qua nhiều lần gửi – nhận liên tiếp.

- Mọi thay đổi dù rất nhỏ trong file đều khiến chữ ký số không khớp, chứng minh độ nhạy cao trong phát hiện giả mạo.

### ***3.4.4. Hạn chế***

|  |  |
| --- | --- |
| **Hạn chế** | **Mô tả** |
| Không hỗ trợ file nhị phân đặc biệt | Hiện chỉ hỗ trợ file text |

**KẾT LUẬN**

Sau quá trình phân tích, thiết kế và thử nghiệm hệ thống gửi bệnh án điện tử với xác thực kép, nhóm đã xây dựng được một mô hình truyền tải dữ liệu y tế an toàn, phù hợp với các yêu cầu bảo mật hiện đại.

Hệ thống được chia thành 4 giai đoạn xử lý rõ ràng:

**Handshake – Xác lập kết nối ban đầu** giữa hai bên gửi và nhận để kiểm tra sẵn sàng hệ thống.

**Xác thực & Trao khóa**:

Người gửi sử dụng RSA kết hợp SHA-512 để ký metadata.

Dùng mật khẩu (đã băm bằng SHA-256) để xác thực người nhận.

Session key được mã hóa bằng RSA 2048-bit với padding OAEP đảm bảo an toàn cao khi trao đổi khóa.

**Mã hóa & Kiểm tra toàn vẹn**:

Dữ liệu bệnh án được mã hóa bằng AES-CBC.

IV và ciphertext được nối lại và băm SHA-512 để tạo mã kiểm tra toàn vẹn.

**Xử lý phía người nhận**:

Kiểm tra toàn vẹn, xác minh chữ ký và xác thực mật khẩu.

Chỉ khi cả 3 yếu tố này hợp lệ, dữ liệu mới được giải mã và lưu trữ

Kết quả thực nghiệm cho thấy:

Hệ thống hoạt động ổn định và tuân thủ đúng quy trình bảo mật.

Các thuật toán được sử dụng đều là các tiêu chuẩn quốc tế, phổ biến trong các hệ thống bảo mật thực tế.

Thời gian xử lý nhanh, không gây chậm trễ đáng kể trong truyền tải dữ liệu bệnh án.

Từ đó, có thể khẳng định hệ thống không chỉ đảm bảo **tính bí mật (confidentiality)**, **tính toàn vẹn (integrity)** và **tính xác thực (authenticity)** của dữ liệu, mà còn thể hiện được khả năng mở rộng trong tương lai để tích hợp thêm các cơ chế bảo vệ khác như:

**Xác thực đa yếu tố (2FA)** với OTP, email hoặc mã phần cứng.

**Giao diện người dùng trực quan** để ứng dụng trong thực tế bệnh viện.

**Tích hợp blockchain** để theo dõi lịch sử truy cập, thay đổi bệnh án.

Thông qua việc thực hiện đề tài này, nhóm đã rèn luyện được kỹ năng nghiên cứu, ứng dụng công nghệ bảo mật vào hệ thống thực tế, đồng thời nâng cao tư duy hệ thống, khả năng lập trình, kiểm thử và đánh giá hiệu quả bảo mật của một giải pháp tích hợp nhiều lớp phòng vệ.