**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC NGOẠI NGỮ – TIN HỌC THÀNH PHỐ HỒ CHÍ MINH**

**CÔNG NGHỆ THÔNG TIN**

A yellow sign with red text

Description automatically generated

**MÔN HỌC: LẬP TRÌNH MẠNG NÂNG CAO**

**ĐỀ TÀI: Đảm Bảo Tính Toàn Vẹn Dữ Liệu Bằng Chữ Ký Số**

**GIẢNG VIÊN HƯỚNG DẪN**: Phan Gia Lượng

**Thành viên nhóm:**

Nguyễn Phi – MSSV: 21DH113322

Lý Tấn Minh – MSSV: 21DH111106

Nguyễn Đức Tuấn – MSSV: 21DH114261

TP.HCM, ngày…tháng 07 năm 2024

**Lời nói đầu**

Trong thời đại công nghệ thông tin hiện nay, hệ thống Client-Server đã trở thành một phần không thể thiếu trong các ứng dụng mạng. Với sự phát triển nhanh chóng của Internet và nhu cầu ngày càng tăng về bảo mật dữ liệu, việc xây dựng các hệ thống mạng an toàn và hiệu quả đang trở thành một thách thức lớn. Đề tài "Xây dựng hệ thống Client-Server với mã hóa BASE64(SHA256withRSA)" nhằm mục đích nghiên cứu và triển khai một hệ thống mạng an toàn, giúp đảm bảo tính toàn vẹn và xác thực của dữ liệu truyền giữa client và server.

Hệ thống này không chỉ cung cấp các giải pháp bảo mật thông qua việc mã hóa và ký số dữ liệu mà còn ứng dụng Remote Method Invocation (RMI) để tạo ra một môi trường giao tiếp linh hoạt và tiện lợi. Với cơ chế xác thực tin nhắn, hệ thống có thể xác định chính xác nguồn gốc của dữ liệu, từ đó nâng cao độ tin cậy và an toàn cho các giao dịch trực tuyến.

Báo cáo này sẽ trình bày chi tiết về quá trình xây dựng và triển khai hệ thống Client-Server. Từ việc thiết kế kiến trúc hệ thống, triển khai các kỹ thuật mã hóa và xác thực, cho đến kiểm thử và đánh giá kết quả, tất cả sẽ được mô tả một cách chi tiết và khoa học. Hy vọng rằng, qua đề tài này, người đọc sẽ có cái nhìn rõ nét hơn về các phương pháp bảo mật trong hệ thống mạng và áp dụng được những kiến thức này vào thực tiễn.

Em xin chân thành cảm ơn thầy Lượng đã tận tình hướng dẫn và hỗ trợ trong suốt quá trình thực hiện báo cáo. Em cũng xin cảm ơn các thầy cô và các bạn đã đóng góp ý kiến và giúp đỡ để em hoàn thành đề tài này.

**Lời Cảm Ơn**

Trước hết, chúng em xin bày tỏ lòng biết ơn sâu sắc đến thầy Phan Gia Lượng – người đã hướng dẫn của chúng em, người đã dành nhiều thời gian và công sức để hỗ trợ, hướng dẫn và chỉ bảo chúng em trong suốt quá trình thực hiện đề tài. Sự tận tình và kiến thức sâu rộng của thầy đã giúp chúng em hiểu rõ hơn về các khái niệm phức tạp cũng như để triển khai Client-Server. Thầy không chỉ truyền đạt kiến thức mà còn khơi dậy niềm đam mê nghiên cứu và tinh thần ham học hỏi trong chúng em.

Chúng em cũng xin gửi lời cảm ơn đến các thành viên trong nhóm, những người đã làm việc không ngừng nghỉ và luôn sẵn sàng giúp đỡ nhau trong suốt quá trình làm việc. Sự hợp tác và tinh thần làm việc nhóm của mọi người đã giúp chúng ta vượt qua nhiều khó khăn và hoàn thành đề tài này một cách xuất sắc. Mỗi người trong nhóm đều đóng góp những ý tưởng và nỗ lực quý báu, tạo nên một tác phẩm chung đầy ý nghĩa.

Bên cạnh đó, chúng em xin gửi lời cảm ơn đến các bạn học cùng lớp, những người đã chia sẻ tài liệu, ý kiến và kinh nghiệm quý báu trong quá trình học tập và nghiên cứu. Sự giúp đỡ và ủng hộ từ các bạn đã tiếp thêm động lực để chúng em hoàn thành đề tài này

Chúng em cũng không quên cảm ơn gia đình và bạn bè, những người luôn động viên, cổ vũ và tạo điều kiện thuận lợi để chúng em có thể tập trung vào việc học tập và nghiên cứu. Sự yêu thương và ủng hộ từ mọi người là nguồn động lực lớn lao giúp chúng em vượt qua mọi thử thách.

Một lần nữa, chúng em xin chân thành cảm ơn tất cả mọi người. Hy vọng rằng đề tài này sẽ đóng góp một phần nhỏ vào kho tàng kiến thức về công nghệ điện toán đám mây và giúp ích cho những người quan tâm đến lĩnh vực này trong tương lai.

Trân trọng!

**MỤC LỤC**

[**Chương I: Giới Thiệu** 5](#_Toc172967267)

[**1. Lý do chọn đề tài** 6](#_Toc172967268)

[**2. Phạm vi, giới hạn của đề tài** 7](#_Toc172967269)

[**3. Mục tiêu đạt được** 7](#_Toc172967270)

[**Chương II: Cơ Sở Lý Thuyết** 9](#_Toc172967271)

[**1. Hệ thống Client-Server** 9](#_Toc172967272)

[**1.1. Khái Niệm** 9](#_Toc172967273)

[**1.2. Mã hóa BASE64** 9](#_Toc172967274)

[**1.3. SHA256** 10](#_Toc172967275)

[**1.4.** **RSA** 11](#_Toc172967276)

[**1.5.** **Remote Method Invocation (RMI)** 11](#_Toc172967277)

[**2. Lợi ích** 12](#_Toc172967278)

[**3. Các mô hình dịch vụ** 12](#_Toc172967279)

[**3.1. Mô hình Client-Server cơ bản** 12](#_Toc172967280)

[**3.2. Mô hình 3-tier (ba tầng)** 13](#_Toc172967281)

[**3.3. Mô hình N-tier (đa tầng)** 13](#_Toc172967282)

[**3.5. Mô hình Microservices** 13](#_Toc172967283)

[**4. Kiến Trúc Hệ Thống** 14](#_Toc172967284)

[**4.1 Sơ đồ tổng quan của hệ thống.** 14](#_Toc172967285)

[**4.2 Mô tả chi tiết các thành phần.** 15](#_Toc172967286)

[**4.3 Cơ chế mã hóa.** 16](#_Toc172967287)

[**4.4 Hoạt động của hệ thống** 16](#_Toc172967288)

[**Kết Luận** 17](#_Toc172967289)

[**CHƯƠNG III: Triển Khai** 18](#_Toc172967290)

[**1. Các file java thuộc project** 18](#_Toc172967291)

[**2. Bắt đầu chạy chương trình** 30](#_Toc172967292)

[**TÀI LIỆU THAM KHẢO** 37](#_Toc172967293)

**Danh mục hình ảnh**

[Hình 1. Hệ thống Client-Server 9](#_Toc172967369)

[Hình 2. Mã hóa Base64 10](#_Toc172967370)

[Hình 3. SHA256 10](#_Toc172967371)

[Hình 4. RSA 11](#_Toc172967372)

[Hình 5. Remote Method Invocation (RMI) 12](#_Toc172967373)

[Hình 6. Mô hình client-server 13](#_Toc172967374)

[Hình 7. Mô hình Microservices 14](#_Toc172967375)

[*Hình 8. Sơ đồ tổng quan của hệ thống.* 15](#_Toc172967376)

[Hình 9. Sơ đồ SHA256withRSA 16](#_Toc172967377)

[Hình 10. Project dự án 18](#_Toc172967378)

[Hình 11. Database project 19](#_Toc172967379)

[Hình 12. Server.java 20](#_Toc172967380)

[Hình 13. Service.java ( interface ) 21](#_Toc172967381)

[Hình 14. ServiceImpl.java (1) 21](#_Toc172967382)

[Hình 15. ServiceImpl (2) 22](#_Toc172967383)

[Hình 16. ServiceImpl (3) 23](#_Toc172967384)

[Hình 17. ServiceImpl (4) 24](#_Toc172967385)

[Hình 18. ServiceImpl (5) 25](#_Toc172967386)

[Hình 19. User.java 26](#_Toc172967387)

[Hình 20. ConnectionDatabase.java 27](#_Toc172967388)

[Hình 21. Client.java (1) 28](#_Toc172967389)

[Hình 22. Client.java (2) 29](#_Toc172967390)

[Hình 23. Client.java (3) 30](#_Toc172967391)

[Hình 24. Server.java running 31](#_Toc172967392)

[Hình 25. Client.java running 32](#_Toc172967393)

[Hình 26. Xem danh sách user 33](#_Toc172967394)

[Hình 27. Thêm user 33](#_Toc172967395)

[Hình 28. Cập nhập user 34](#_Toc172967396)

[Hình 29. Server nhận phản hồi 35](#_Toc172967397)

[Hình 30. Xóa user 36](#_Toc172967398)

# **Chương I: Giới Thiệu**

## **1. Lý do chọn đề tài**

Trong bối cảnh công nghệ thông tin phát triển mạnh mẽ, các hệ thống Client-Server đóng vai trò quan trọng trong việc cung cấp các dịch vụ trực tuyến, từ những ứng dụng web cơ bản cho đến các hệ thống phức tạp trong doanh nghiệp. Một trong những thách thức lớn nhất đối với các hệ thống này là đảm bảo an toàn và bảo mật thông tin trong quá trình trao đổi dữ liệu giữa Client và Server. Việc sử dụng các kỹ thuật mã hóa và xác thực hiện đại không chỉ giúp bảo vệ dữ liệu mà còn đảm bảo tính toàn vẹn và nguồn gốc của thông tin.

\*Việc bảo mật thông tin và xác thực nguồn gốc dữ liệu đang trở thành một trong những yếu tố then chốt trong các hệ thống mạng hiện đại. Các cuộc tấn công mạng ngày càng tinh vi và phức tạp, đòi hỏi các giải pháp bảo mật mạnh mẽ hơn. Đề tài "Xây dựng hệ thống Client-Server với mã hóa **BASE64(SHA256withRSA)" được chọn vì:**

\* Tính cấp thiết: Bảo mật thông tin là một trong những yêu cầu hàng đầu trong các hệ thống mạng, đặc biệt là trong các ứng dụng đòi hỏi độ tin cậy cao.

\* Ứng dụng thực tiễn: Hệ thống này có thể được áp dụng trong nhiều lĩnh vực khác nhau như thương mại điện tử, ngân hàng trực tuyến, và các hệ thống quản lý thông tin doanh nghiệp.

\* Khả năng mở rộng và phát triển: Đề tài cung cấp một nền tảng cơ bản cho việc nghiên cứu và phát triển các giải pháp bảo mật nâng cao hơn trong tương lai.

\* Nâng cao kiến thức và kỹ năng cá nhân: Thực hiện đề tài này giúp tôi củng cố và nâng cao kiến thức về lập trình mạng, các kỹ thuật mã hóa và bảo mật, cũng như các công nghệ như Remote Method Invocation (RMI).

\* Đáp ứng yêu cầu của khóa học: Đề tài phù hợp với nội dung và mục tiêu của môn học lập trình mạng nâng cao, giúp tôi hoàn thành tốt yêu cầu của khóa học và đạt được các mục tiêu học tập đã đề ra.

\* Thách thức và học hỏi: Đề tài này đặt ra nhiều thách thức kỹ thuật, giúp tôi học hỏi và giải quyết các vấn đề phức tạp trong việc xây dựng hệ thống an toàn và hiệu quả.

\* Xu hướng công nghệ: Việc nghiên cứu và ứng dụng các công nghệ bảo mật tiên tiến như SHA256withRSA phù hợp với xu hướng phát triển của ngành công nghệ thông tin hiện nay, giúp tôi cập nhật và bắt kịp với những tiến bộ mới nhất trong lĩnh vực này.

\* Đóng góp cho cộng đồng: Hệ thống được xây dựng có thể làm cơ sở cho các nghiên cứu và phát triển tiếp theo, góp phần nâng cao chất lượng và an toàn của các hệ thống mạng trong tương lai.

## **2. Phạm vi, giới hạn của đề tài**

Phạm vi nghiên cứu của đề tài bao gồm:

Thiết kế và triển khai hệ thống Client-Server: Xây dựng một hệ thống trong đó Client gửi tin nhắn được mã hóa và ký số bằng BASE64(SHA256withRSA) đến Server.

Xác thực tin nhắn: Server sẽ kiểm tra tính toàn vẹn và nguồn gốc của tin nhắn từ Client.

Gọi RMI để lấy thông tin từ cơ sở dữ liệu: Sau khi xác thực thành công, Client sẽ gọi RMI từ Server để lấy thông tin toàn bộ user trong cơ sở dữ liệu.

Giới hạn của đề tài:

Cơ sở dữ liệu đơn giản: Hệ thống chỉ tập trung vào việc lấy thông tin user từ cơ sở dữ liệu, không bao gồm các chức năng quản lý dữ liệu phức tạp.

Môi trường thử nghiệm: Hệ thống được triển khai và kiểm thử trong môi trường mạng nội bộ, chưa bao gồm các yếu tố phức tạp từ mạng internet.

## **3. Mục tiêu đạt được**

Xây dựng hệ thống Client-Server an toàn: Thiết kế và triển khai hệ thống có khả năng bảo mật thông tin và xác thực nguồn gốc dữ liệu.

Ứng dụng kỹ thuật mã hóa và ký số: Sử dụng BASE64(SHA256withRSA) để đảm bảo tính toàn vẹn và bảo mật của tin nhắn.

Thực hiện RMI: Tích hợp Remote Method Invocation để tạo ra một hệ thống giao tiếp linh hoạt và hiệu quả giữa Client và Server.

Kiểm thử và đánh giá hệ thống: Đảm bảo hệ thống hoạt động ổn định, an toàn và đạt được các yêu cầu đặt ra.

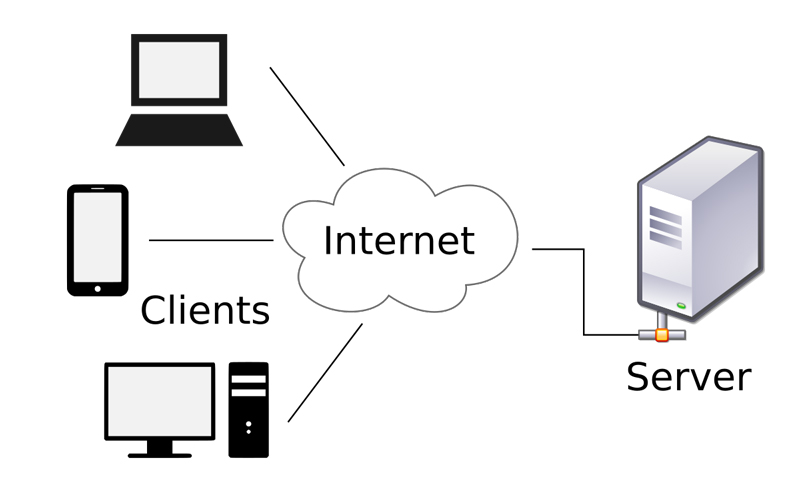
Việc hoàn thành đề tài này không chỉ giúp tôi củng cố kiến thức về lập trình mạng và bảo mật thông tin mà còn mở ra nhiều hướng nghiên cứu và ứng dụng mới trong tương lai.

# **Chương II: Cơ Sở Lý Thuyết**

## **1. Hệ thống Client-Server**

### **1.1. Khái Niệm**

- Hệ thống Client-Server là mô hình kiến trúc phân tán trong đó các client (người dùng) gửi yêu cầu dịch vụ và các server (máy chủ) cung cấp dịch vụ. Các client và server có thể hoạt động trên các máy tính khác nhau và giao tiếp với nhau qua mạng.



Hình 1. Hệ thống Client-Server

### **1.2. Mã hóa BASE64**

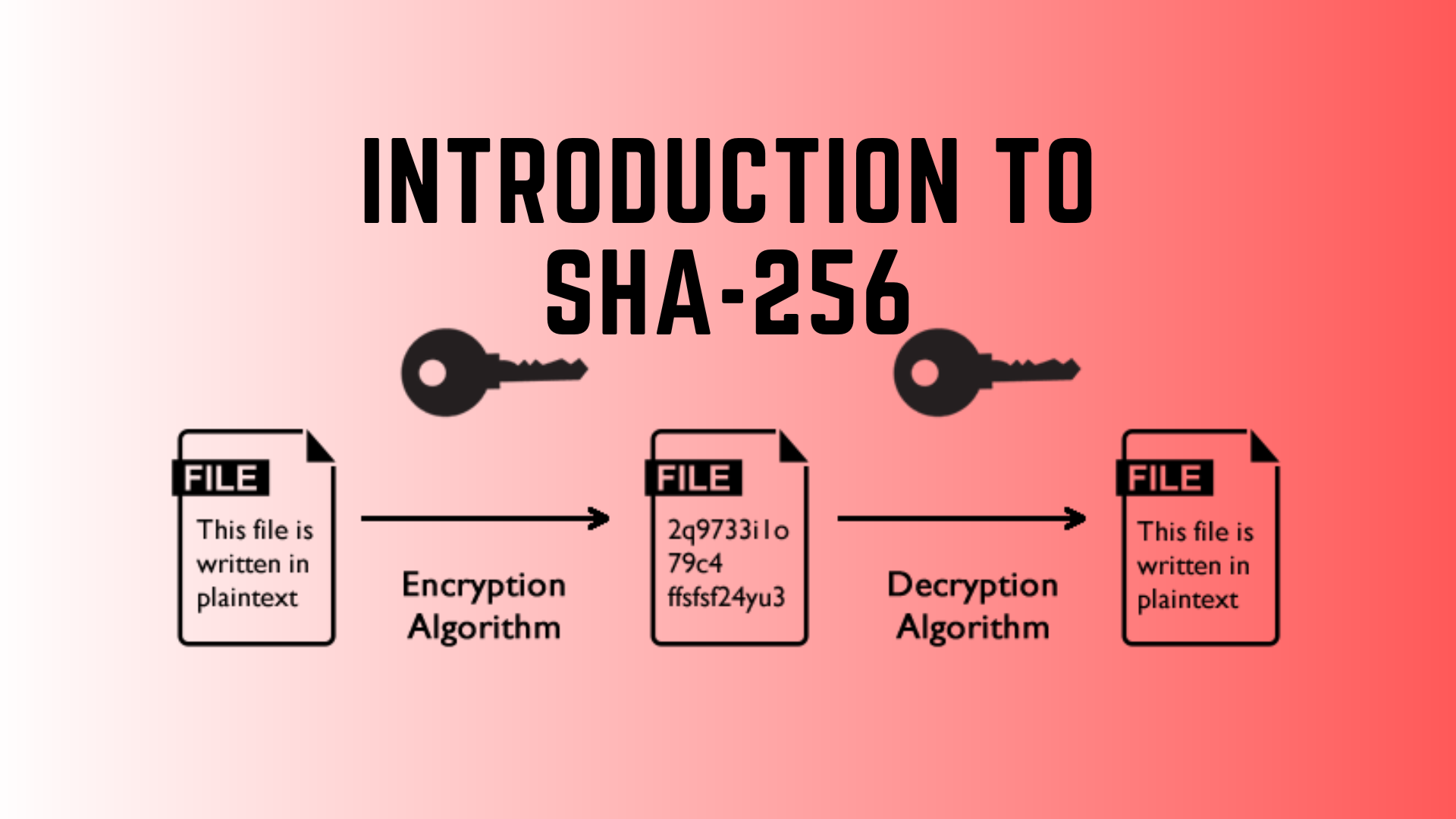
- BASE64 là một phương pháp mã hóa dữ liệu bằng cách chuyển đổi dữ liệu nhị phân thành chuỗi ký tự ASCII. Mã hóa BASE64 thường được sử dụng để mã hóa dữ liệu khi truyền qua mạng hoặc lưu trữ.



Hình 2. Mã hóa Base64

### **1.3. SHA256**

- SHA256 là một thuật toán băm mật mã thuộc họ SHA-2, tạo ra một giá trị băm có độ dài 256 bit. SHA256 được sử dụng để đảm bảo tính toàn vẹn của dữ liệu bằng cách tạo ra một mã băm duy nhất từ dữ liệu gốc.



Hình 3. SHA256

### **1.4.** **RSA**

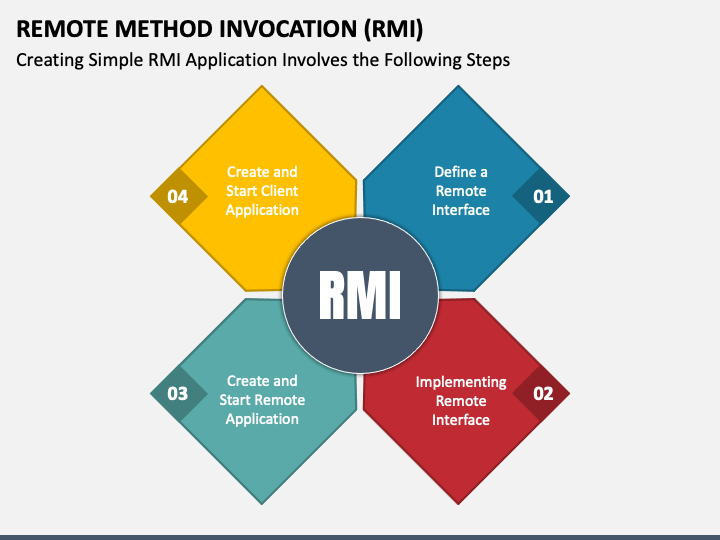
- RSA là một thuật toán mật mã không đối xứng, sử dụng cặp khóa công khai và khóa bí mật để mã hóa và giải mã dữ liệu. RSA thường được sử dụng trong các hệ thống bảo mật để mã hóa dữ liệu và xác thực nguồn gốc dữ liệu.



Hình 4. RSA

### **1.5.** **Remote Method Invocation (RMI)**

- RMI là một cơ chế trong Java cho phép một đối tượng trên máy tính này gọi phương thức của một đối tượng trên máy tính khác. RMI được sử dụng để tạo ra các ứng dụng phân tán, nơi các đối tượng trên các máy tính khác nhau có thể giao tiếp với nhau.



Hình 5. Remote Method Invocation (RMI)

## **2. Lợi ích**

**Tổng quan về lợi ích**

-Bảo mật thông tin

Mã hóa và xác thực dữ liệu giúp bảo vệ thông tin khỏi các cuộc tấn công mạng, đảm bảo tính toàn vẹn và bảo mật của dữ liệu.

-Phân tán và mở rộng

Mô hình Client-Server cho phép hệ thống được phân tán và mở rộng dễ dàng, đáp ứng được các yêu cầu về tài nguyên và hiệu suất khi số lượng người dùng tăng lên.

-Quản lý và bảo trì dễ dàng

Việc quản lý và bảo trì hệ thống Client-Server trở nên dễ dàng hơn do các chức năng được phân chia rõ ràng giữa client và server.

-Khả năng tương thích và tích hợp

Mô hình Client-Server cho phép tích hợp với các hệ thống và dịch vụ khác một cách linh hoạt, hỗ trợ nhiều giao thức và công nghệ khác nhau.

## **3. Các mô hình dịch vụ**

### **3.1. Mô hình Client-Server cơ bản**

Mô hình này, client gửi yêu cầu dịch vụ tới server, server xử lý yêu cầu và gửi phản hồi về cho client. Đây là mô hình phổ biến nhất và được sử dụng rộng rãi trong nhiều ứng dụng mạng.

Ảnh có chứa văn bản, biểu đồ, ảnh chụp màn hình, hàng

Mô tả được tạo tự động

Hình 6. Mô hình client-server

### **3.2. Mô hình 3-tier (ba tầng)**

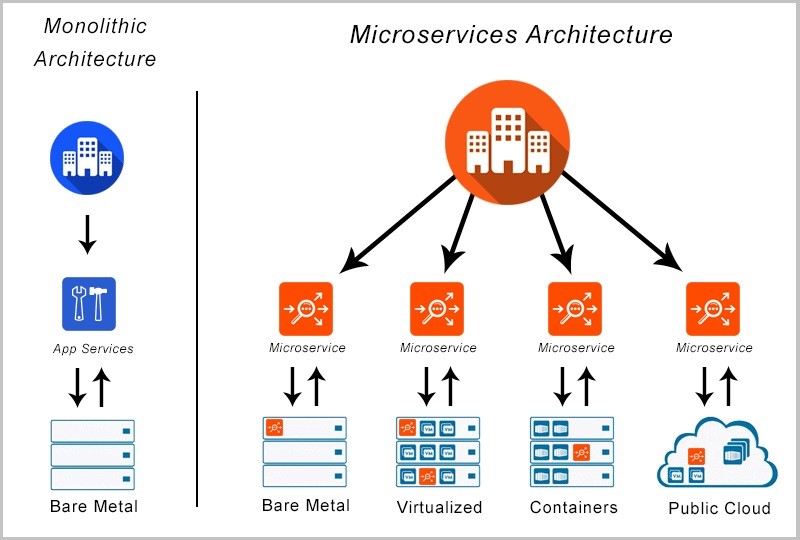
- Mô hình 3-tier phân chia ứng dụng thành ba tầng: tầng giao diện người dùng (UI), tầng logic nghiệp vụ (BLL), và tầng dữ liệu (DAL). Tầng UI xử lý giao diện và tương tác người dùng, tầng BLL xử lý logic nghiệp vụ, và tầng DAL quản lý truy cập dữ liệu.

### **3.3. Mô hình N-tier (đa tầng)**

- Mô hình N-tier là sự mở rộng của mô hình 3-tier, phân chia ứng dụng thành nhiều tầng hơn để tăng cường khả năng mở rộng và quản lý. Mỗi tầng trong mô hình N-tier có thể đảm nhận một vai trò cụ thể và giao tiếp với các tầng khác thông qua các giao thức định sẵn.

### **3.5. Mô hình Microservices**

- Microservices là một kiến trúc trong đó ứng dụng được chia thành các dịch vụ nhỏ, độc lập, mỗi dịch vụ thực hiện một chức năng cụ thể. Các dịch vụ này giao tiếp với nhau qua các API, giúp hệ thống dễ dàng mở rộng và bảo trì.

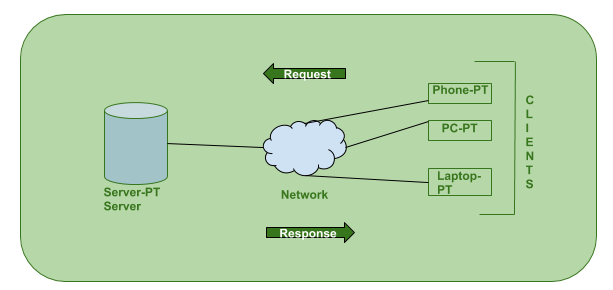


Hình 7. Mô hình Microservices

## **4. Kiến Trúc Hệ Thống**

### **4.1 Sơ đồ tổng quan của hệ thống.**

- Sơ đồ trên mô tả tổng quan kiến trúc của hệ thống Client-Server với các thành phần chính gồm Client, Server, và cơ chế giao tiếp giữa chúng qua mạng. Hệ thống sử dụng các kỹ thuật mã hóa và xác thực để đảm bảo an toàn thông tin.



*Hình 8. Sơ đồ tổng quan của hệ thống.*

### **4.2 Mô tả chi tiết các thành phần.**

**\* Client**

Client là ứng dụng hoặc thiết bị gửi yêu cầu dịch vụ đến Server. Trong hệ thống này, Client thực hiện các chức năng sau:

Tạo và gửi tin nhắn: Client tạo một tin nhắn và sử dụng RSA để ký số (SHA256withRSA) và mã hóa bằng BASE64 trước khi gửi đến Server.

Nhận và xử lý phản hồi từ Server: Client nhận phản hồi từ Server và thực hiện các hành động dựa trên kết quả xác thực (ví dụ: gọi RMI nếu tin nhắn được xác thực).

Gọi RMI: Sau khi nhận được phản hồi "VERIFIED; HUFLIT", Client sẽ gọi phương thức RMI từ Server để lấy thông tin toàn bộ user trong cơ sở dữ liệu.

**\* Server**

Server là ứng dụng hoặc thiết bị cung cấp dịch vụ cho Client. Trong hệ thống này, Server thực hiện các chức năng sau:

Nhận và xác thực tin nhắn: Server nhận tin nhắn từ Client, giải mã BASE64, và xác thực chữ ký số bằng cách sử dụng khóa công khai RSA.

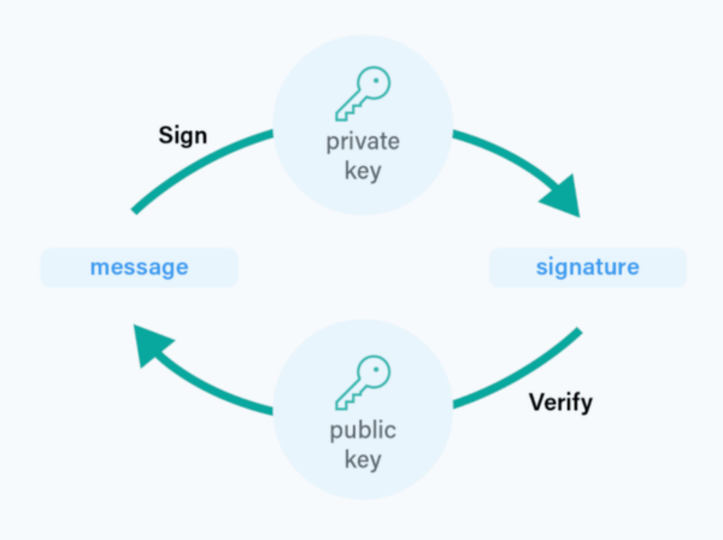
Gửi phản hồi cho Client: Nếu tin nhắn được xác thực thành công, Server gửi phản hồi "VERIFIED; HUFLIT" về cho Client. Nếu không, Server gửi "VERIFICATION\_FAILED".

Thực hiện RMI: Server cung cấp các phương thức RMI để Client có thể gọi và lấy thông tin từ cơ sở dữ liệu.

### **4.3 Cơ chế mã hóa.**

SHA256withRSA: Đây là phương pháp ký số sử dụng thuật toán SHA256 để tạo ra giá trị băm của tin nhắn và sau đó ký số bằng RSA. Kết quả là một chữ ký số đảm bảo tính toàn vẹn và xác thực nguồn gốc của tin nhắn.

BASE64: Mã hóa BASE64 được sử dụng để chuyển đổi dữ liệu nhị phân thành chuỗi ký tự ASCII, giúp dễ dàng truyền dữ liệu qua mạng.



Hình 9. Sơ đồ SHA256withRSA

**Xác thực**

Client ký số tin nhắn: Client sử dụng khóa bí mật RSA để ký số tin nhắn và tạo ra một chữ ký số.

Server xác thực chữ ký số: Server sử dụng khóa công khai RSA để kiểm tra chữ ký số của tin nhắn. Nếu chữ ký hợp lệ, tin nhắn được xác thực là từ Client hợp lệ.

### **4.4 Hoạt động của hệ thống**

1. Client tạo tin nhắn: Client tạo một tin nhắn và sử dụng SHA256withRSA để ký số, sau đó mã hóa kết quả bằng BASE64.

2. Client gửi tin nhắn đến Server: Client gửi tin nhắn đã mã hóa đến Server.

3. Server nhận và xác thực tin nhắn: Server nhận tin nhắn, giải mã BASE64 và sử dụng khóa công khai RSA để xác thực chữ ký số.

4. Server phản hồi Client: Nếu tin nhắn hợp lệ, Server gửi "VERIFIED; HUFLIT" về Client. Nếu không, Server gửi "VERIFICATION\_FAILED".

5. Client gọi RMI: Nếu nhận được "VERIFIED; HUFLIT", Client gọi phương thức RMI từ Server để lấy thông tin người dùng từ cơ sở dữ liệu.

## **Kết Luận**

Kiến trúc hệ thống Client-Server với mã hóa và xác thực tin nhắn bằng BASE64(SHA256withRSA) kết hợp với cơ chế RMI tạo ra một hệ thống bảo mật, hiệu quả và dễ mở rộng. Việc áp dụng các kỹ thuật này không chỉ đảm bảo an toàn thông tin mà còn nâng cao tính linh hoạt và khả năng quản lý của hệ thống.

# **CHƯƠNG III: Triển Khai**

## **1. Các file java thuộc project**

A screenshot of a computer program

Description automatically generated

Hình 10. Project dự án

- Database project sử dụng

A screenshot of a computer

Description automatically generated

Hình 11. Database project

A screenshot of a computer

Description automatically generated

Hình 12. Server.java

A screenshot of a computer program

Description automatically generated

Hình 13. Service.java ( interface )

A screenshot of a computer

Description automatically generated

Hình 14. ServiceImpl.java (1)

A screenshot of a computer

Description automatically generated

Hình 15. ServiceImpl (2)

A screenshot of a computer program

Description automatically generated

Hình 16. ServiceImpl (3)

A screenshot of a computer program

Description automatically generated

Hình 17. ServiceImpl (4)

A screenshot of a computer program

Description automatically generated

Hình 18. ServiceImpl (5)

A screenshot of a computer program

Description automatically generated

Hình 19. User.java

A computer screen shot of a program

Description automatically generated

Hình 20. ConnectionDatabase.java

A screenshot of a computer program

Description automatically generated

Hình 21. Client.java (1)

A screenshot of a computer program

Description automatically generated

Hình 22. Client.java (2)

A screenshot of a computer

Description automatically generated

Hình 23. Client.java (3)

## **2. Bắt đầu chạy chương trình**

- Đầu tiên ta chạy Server.java để khởi chạy RMI Server

A screenshot of a computer

Description automatically generated

Hình 24. Server.java running

- Sau đó ta chạy Client.java để verify với server. Sau khi Client nhận được lệnh “VERIFIED HUFLIT” thì server sẽ trả về thông tin của tất cả user trong database

A screenshot of a computer program

Description automatically generated

Hình 25. Client.java running

A screenshot of a computer program

Description automatically generated

Hình 26. Xem danh sách user

A screenshot of a computer code

Description automatically generated

Hình 27. Thêm user

A screenshot of a computer program

Description automatically generated

Hình 28. Cập nhập user

- Server nhận phản hồi sau khi sửa user trong database

A screenshot of a computer program

Description automatically generated

Hình 29. Server nhận phản hồi

A screenshot of a computer

Description automatically generated

Hình 30. Xóa user

# **TÀI LIỆU THAM KHẢO**

[1] *W3schools.com*. W3Schools Online Web Tutorials. (n.d.). https://www.w3schools.com/java/default.asp

[2] *Huflit / lab · GITLAB*. GitLab. (n.d.).

https://gitlab.com/huflit0/lab