

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

**KHOA CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO THỰC NGHIỆM**

Học phần: An ninh mạng

**Chủ đề:** **Chữ ký số RSA và ứng dụng trong xác thực tính toàn vẹn của dữ liệu**

Giảng viên hướng dẫn: TS. Phạm Văn Hiệp

Nhóm sinh viên thực hiện:

1. Trần Gia Hoàng Mã SV: 2021605995
2. Phạm Việt Hưng Mã SV: 2021603253
3. Phạm Đức Toản Mã SV: 2021606942
4. Trần Xuân Đức Mã SV: 2022600490
5. Đỗ Trọng Thích Mã SV: 2021604318

Mã lớp học phần: 202412IT6070001

Nhóm: 1

Hà Nội - Năm 2025

# MỤC LỤC

[MỤC LỤC 2](#_Toc186720065)

[DANH MỤC HÌNH ẢNH 5](#_Toc186720066)

[DANH MỤC BẢNG BIỂU 7](#_Toc186720067)

[LỜI CẢM ƠN 8](#_Toc186720068)

[LỜI MỞ ĐẦU 9](#_Toc186720069)

[CHƯƠNG 1 TỔNG QUAN VỀ ĐỀ TÀI 11](#_Toc186720070)

[1.1. Tổng quan về an ninh mạng 11](#_Toc186720071)

[1.2. Các kiến thức cơ sở 12](#_Toc186720072)

[1.2.1 Toán học cơ bản 12](#_Toc186720073)

[1.2.2 Thuật toán RSA 12](#_Toc186720074)

[1.2.3 Ngôn ngữ lập trình 12](#_Toc186720075)

[1.3. Nội dung nghiên cứu 12](#_Toc186720076)

[1.3.1. Lý do chọn đề tài 12](#_Toc186720077)

[1.3.2. Các nội dung nghiên cứu 13](#_Toc186720078)

[1.4. Giới thiệu về RSA 14](#_Toc186720079)

[CHƯƠNG 2 KẾT QUẢ NGHIÊN CỨU 16](#_Toc186720080)

[2.1. Nghiên cứu, tìm hiểu hệ mã hóa khóa công khai 16](#_Toc186720081)

[2.1.1. Khái niệm hệ mã hóa khóa công khai 16](#_Toc186720082)

[2.1.2. Các đặc điểm của hệ mã hóa khóa công khai 17](#_Toc186720083)

[2.1.3. Nguyên lý hoạt động của hệ mã hóa khóa công khai 17](#_Toc186720084)

[2.1.4. Thuật toán RSA trong hệ mã hóa khóa công khai 18](#_Toc186720085)

[2.1.5. Ưu điểm và nhược điểm của hệ mã hóa khóa công khai 19](#_Toc186720086)

[2.1.6. Ứng dụng của hệ mã hóa khóa công khai trong thực tiễn 19](#_Toc186720087)

[2.2. Chữ ký số và ứng dụng trong thực tế 20](#_Toc186720088)

[2.2.1. Định nghĩa và khái niệm về chữ ký số 20](#_Toc186720089)

[2.2.2. Cách thức hoạt động của chữ ký số 20](#_Toc186720090)

[2.2.3. Các tiêu chuẩn chữ ký số phổ biến 21](#_Toc186720091)

[2.2.4. Ưu điểm và nhược điểm của chữ ký số 22](#_Toc186720092)

[2.2.5. Ứng dụng của chữ ký số trong thực tế 23](#_Toc186720093)

[2.2.6. Chữ ký số và các thách thức hiện tại 25](#_Toc186720094)

[2.2.7. Vai trò của chữ ký số trong tương lai 25](#_Toc186720095)

[2.3. Nghiên cứu, tìm hiểu về chữ ký số rsa và ứng dụng trong xác thực tính toàn vẹn của dữ liệu 25](#_Toc186720096)

[2.3.1. Quá trình tạo khóa cho hệ mật mã RSA 27](#_Toc186720097)

[2.3.2. Quá trình mã hóa 27](#_Toc186720098)

[2.3.3. Quá trình giải mã 28](#_Toc186720099)

[2.3.4. Ví dụ 28](#_Toc186720100)

[2.3.5. Chuyển đổi văn bản rõ 29](#_Toc186720101)

[2.3.6. Tạo chữ ký vào văn bản 30](#_Toc186720102)

[2.4. Thiết kế chương trình và thuật toán 31](#_Toc186720103)

[2.4.1. Thiết kế kịch bản chương trình 31](#_Toc186720104)

[2.4.2. Giới thiệu ngôn ngữ lập trình sử dụng để cài đặt thuật toán 34](#_Toc186720105)

[2.4.3. Cài đặt thuật toán và giao diện cho đề tài 37](#_Toc186720106)

[CHƯƠNG 3 KẾT LUẬN VÀ BÀI HỌC KINH NGHIỆM 46](#_Toc186720107)

[3.1. Kiến thức kỹ năng đã học được trong quá trình thực hiện đề tài. 46](#_Toc186720108)

[3.1.1. Kiến thức chuyên môn 46](#_Toc186720109)

[3.1.2. Kỹ năng kỹ thuật 46](#_Toc186720110)

[3.1.3. Kinh nghiệm thực tế 47](#_Toc186720111)

[3.1.4. Trình bày và báo cáo 48](#_Toc186720112)

[3.2. Bài học kinh nghiệm 48](#_Toc186720113)

[3.3. Tính khả thi của chủ đề nghiên cứu, những thuận lợi, khó khăn 50](#_Toc186720114)

[3.3.1. Tính khả thi 50](#_Toc186720115)

[3.3.2. Những thuận lợi và khó khăn 52](#_Toc186720116)

[KẾT LUẬN 55](#_Toc186720117)

[TÀI LIỆU THAM KHẢO 57](#_Toc186720118)

# DANH MỤC HÌNH ẢNH

[Hình 1. 1 Các tác giả 14](#_Toc186716723)

[Hình 2. 1 Sơ đồ biểu diễn khóa công khai 18](#_Toc186716727)

[Hình 2. 2 Nguyên lý hoạt động của mã hóa khóa công khai 20](#_Toc186716728)

[Hình 2. 3 Thuật toán Java 40](#_Toc186716729)

[Hình 2. 4 Giao diện chương trình Java 41](#_Toc186716730)

[Hình 2. 5 Sinh khóa 41](#_Toc186716731)

[Hình 2. 6 Sinh khóa thành công 41](#_Toc186716732)

[Hình 2. 7 Mã hóa 41](#_Toc186716733)

[Hình 2. 8 Mã hóa thành công 42](#_Toc186716734)

[Hình 2. 9 Giải mã 42](#_Toc186716735)

[Hình 2. 10 Giải mã thành công 42](#_Toc186716736)

[Hình 2. 11 Giải mã thất bại 42](#_Toc186716737)

[Hình 2. 12 Giao diện chương trình C# 42](#_Toc186716738)

[Hình 2. 13 Chọn kích thước khóa 43](#_Toc186716739)

[Hình 2. 14 Tạo khóa thành công 43](#_Toc186716740)

[Hình 2. 15 Chọn tệp 44](#_Toc186716741)

[Hình 2. 16 Tạo chữ kí 44](#_Toc186716742)

[Hình 2. 17 Mã hóa chữ ký 45](#_Toc186716743)

[Hình 2. 18 Xác thực chữ ký 45](#_Toc186716744)

[Hình 2. 19 Xuất chữ ký ra tệp 46](#_Toc186716745)

# DANH MỤC BẢNG BIỂU

[Bảng 2. 1 So sánh kết quả giữa hai hàm băm 22](#_Toc186712652)

[Bảng 2. 2 Liệt kê chi tiết ưu và nhược điểm 24](#_Toc186712653)

[Bảng 2. 3 Tổng hợp lĩnh vực ứng dụng 26](#_Toc186712654)

LỜI CẢM ƠN

Lời đầu tiên, chúng em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến các thầy cô trong khoa Công nghệ Thông tin - Trường Đại học Công Nghiệp Hà Nội. Nhờ sự tận tâm và nhiệt huyết của các thầy cô trong suốt quá trình giảng dạy, chúng em đã tiếp thu được những kiến thức quý báu, không chỉ giúp ích cho quá trình học tập mà còn là hành trang quan trọng cho sự nghiệp sau này.

Đặc biệt, chúng em xin gửi lời tri ân sâu sắc đến thầy **Phạm Văn Hiệp**, người đã trực tiếp hướng dẫn, dìu dắt chúng em trong suốt quá trình thực hiện đồ án. Thầy không chỉ cung cấp cho chúng em những kiến thức chuyên môn cần thiết mà còn khơi dậy tinh thần học hỏi, sáng tạo, và khả năng giải quyết vấn đề. Sự hỗ trợ tận tình và những chỉ bảo quý báu của thầy đã giúp chúng em vượt qua nhiều khó khăn, thử thách để hoàn thành đồ án này.

Mặc dù đã nỗ lực hết sức, nhưng do còn hạn chế về kiến thức và kinh nghiệm, chúng em chắc chắn không thể tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự thông cảm và những ý kiến đóng góp quý báu từ các thầy cô và các bạn để chúng em có thể hoàn thiện đồ án này một cách tốt nhất. Chúng em xin trân trọng cảm ơn!

**Nhóm thực hiện**

Trần Gia Hoàng

Phạm Việt Hưng

Phạm Đức Toản

Trần Xuân Đức

Đỗ Trọng Thích

# LỜI MỞ ĐẦU

Trong thời đại công nghệ thông tin phát triển vượt bậc, an ninh mạng đã trở thành một lĩnh vực trọng yếu, quyết định sự an toàn và phát triển của các cá nhân, tổ chức và quốc gia. Trước sự gia tăng không ngừng của các cuộc tấn công mạng, việc đảm bảo tính bảo mật, toàn vẹn, và xác thực của dữ liệu trở thành nhu cầu cấp thiết.

Trong số các giải pháp bảo mật, chữ ký số RSA nổi bật như một phương pháp hiệu quả, được ứng dụng rộng rãi trong việc xác thực danh tính và đảm bảo tính toàn vẹn của dữ liệu. Chữ ký số không chỉ giúp ngăn chặn các hành vi giả mạo mà còn đóng vai trò quan trọng trong việc xây dựng lòng tin trong các giao dịch số, bảo vệ dữ liệu nhạy cảm trước những mối đe dọa an ninh mạng.

Nhận thấy tầm quan trọng của vấn đề này, nhóm chúng em đã chọn nghiên cứu đề tài **"Chữ ký số RSA và ứng dụng trong xác thực tính toàn vẹn của dữ liệu"**. Đề tài tập trung vào việc tìm hiểu nguyên lý hoạt động của thuật toán RSA, cách thức tạo và xác thực chữ ký số, đồng thời triển khai ứng dụng thực tiễn để minh họa khả năng bảo vệ dữ liệu của chữ ký số RSA trong môi trường thực tế.

**Chương 1: Tổng quan về đề tài** – Chương này giới thiệu tổng quan về an ninh mạng, các kiến thức cơ sở liên quan đến chữ ký số RSA, đồng thời trình bày lý do và nội dung nghiên cứu.

**Chương 2: Kết quả nghiên cứu** – Chương này nghiên cứu tìm hiểu khái niệm,đặc điểm,nguyên lí,thuật toán.ưu điểm và nhược điểm hệ mã khóa công khai,chữ kí số

Đặc biệt là chữ kí số RSA và ứng dụng.chương còn thiết kế chương trình và thuật toán

**Chương 3: Kết luận và bài học kinh nghiệm** – Chương này tóm tắt các kiến thức, kỹ năng đã học được, rút ra bài học kinh nghiệm và đánh giá tính khả thi của chủ đề nghiên cứu.

Chúng em hy vọng rằng báo cáo này không chỉ cung cấp cái nhìn rõ ràng về chữ ký số RSA mà còn góp phần nâng cao nhận thức về tầm quan trọng của việc bảo vệ tính toàn vẹn của dữ liệu trong bối cảnh các mối đe dọa mạng ngày càng gia tăng.

# CHƯƠNG 1 TỔNG QUAN VỀ ĐỀ TÀI

## Tổng quan về an ninh mạng

Trong kỷ nguyên số, mọi thứ đều được kết nối, từ chiếc điện thoại thông minh đến các hệ thống công nghiệp. Sự tiện lợi mà công nghệ mang lại đi kèm với những rủi ro tiềm ẩn về an ninh mạng. Các cuộc tấn công mạng ngày càng tinh vi và đa dạng, đe dọa không chỉ các tổ chức lớn mà còn cả cá nhân. Từ những vụ rò rỉ dữ liệu quy mô lớn làm chấn động toàn cầu đến các cuộc tấn công ransomware khiến các hệ thống tê liệt, an ninh mạng đã trở thành một vấn đề cấp bách.

Những kẻ tấn công không ngừng tìm kiếm những lỗ hổng trong hệ thống để đánh cắp dữ liệu, phá hoại hoạt động và tống tiền. Các hình thức tấn công cũng trở nên đa dạng hơn, từ các cuộc tấn công truyền thống như phần mềm độc hại, phishing đến các cuộc tấn công nhắm vào các thiết bị IoT, trí tuệ nhân tạo và blockchain.

Để đối phó với những thách thức này, các tổ chức cần xây dựng một chiến lược an ninh mạng toàn diện, kết hợp các giải pháp công nghệ hiện đại với việc nâng cao nhận thức của người dùng. Việc đầu tư vào các công cụ bảo mật như tường lửa, hệ thống phát hiện xâm nhập, và các giải pháp AI để phát hiện các mối đe dọa mới là vô cùng cần thiết. Bên cạnh đó, việc đào tạo nhân viên về an ninh mạng, xây dựng các quy trình bảo mật chặt chẽ và thường xuyên cập nhật các biện pháp phòng thủ cũng đóng vai trò quan trọng.

Tuy nhiên, an ninh mạng không chỉ là một vấn đề công nghệ. Đó còn là một cuộc chiến giữa trí tuệ của con người. Các hacker luôn tìm kiếm những cách thức mới để tấn công, trong khi các chuyên gia bảo mật phải không ngừng đổi mới để đối phó. Đây là một cuộc đua không có hồi kết, đòi hỏi sự sáng tạo, kiên trì và sự hợp tác giữa các tổ chức, chính phủ và cộng đồng.

Mặc dù đối mặt với nhiều thách thức, an ninh mạng cũng mang đến những cơ hội lớn. Việc phát triển các giải pháp bảo mật hiệu quả không chỉ giúp bảo vệ dữ liệu mà còn tạo ra những sản phẩm, dịch vụ mới, thúc đẩy sự phát triển của ngành công nghiệp bảo mật. Bên cạnh đó, việc nâng cao nhận thức về an ninh mạng cũng giúp mọi người trở thành những người sử dụng internet an toàn hơn.

Tóm lại, an ninh mạng là một vấn đề quan trọng ảnh hưởng đến mọi khía cạnh của cuộc sống số. Để xây dựng một thế giới số an toàn và đáng tin cậy, chúng ta cần chung tay nỗ lực, đầu tư vào công nghệ, nâng cao nhận thức và không ngừng học hỏi để đối phó với những thách thức mới.

## Các kiến thức cơ sở

### *1.2.1 Toán học cơ bản*

**Số nguyên tố**: Hai số nguyên tố lớn được sử dụng để tạo khóa trong RSA.

**Modulo**: Dùng trong mã hóa và giải mã để xử lý các phép toán với số lớn.

**Thuật toán Euclid mở rộng**: Tìm nghịch đảo modulo – một bước quan trọng để tính khóa bí mật trong RSA.

### *1.2.2 Thuật toán RSA*

RSA là hệ mã hóa khóa công khai sử dụng hai khóa: khóa công khai (public key) và khóa bí mật (private key).

### *1.2.3 Ngôn ngữ lập trình*

**Java**: Hỗ trợ nhiều thư viện mã hóa như java.security.

**C#**: Cung cấp các API mạnh mẽ qua System.Security.Cryptography.

## Nội dung nghiên cứu

### *Lý do chọn đề tài*

Trong kỷ nguyên số hiện nay, an ninh mạng đóng vai trò cực kỳ quan trọng trong việc bảo vệ thông tin và dữ liệu. Việc đảm bảo tính toàn vẹn và xác thực của dữ liệu là một thách thức lớn, đặc biệt khi dữ liệu được truyền tải và lưu trữ trên các hệ thống mạng mở và dễ bị tấn công. Chữ ký số RSA, một trong những thuật toán mật mã học bất đối xứng phổ biến nhất, cung cấp một giải pháp hiệu quả cho vấn đề này. Đề tài này được chọn vì:

Tính thời sự và cấp thiết: Với sự gia tăng của các hoạt động tội phạm mạng, việc bảo vệ tính toàn vẹn dữ liệu là một nhu cầu bức thiết. Chữ ký số RSA đóng vai trò quan trọng trong nhiều ứng dụng, từ giao dịch điện tử đến bảo mật hệ điều hành.

Tính ứng dụng thực tiễn: Chữ ký số RSA được ứng dụng rộng rãi trong thực tế, giúp người dùng có thể hiểu rõ hơn về cơ chế hoạt động và tầm quan trọng của nó trong bảo mật thông tin.

### *Các nội dung nghiên cứu*

Đề tài sẽ tập trung vào các nội dung chính sau:

**Giới thiệu về chữ ký số và tầm quan trọng của nó trong bảo mật thông tin:** Phần này sẽ định nghĩa chữ ký số, phân loại các loại chữ ký số, và nhấn mạnh vai trò của chữ ký số trong việc đảm bảo tính toàn vẹn, xác thực và không chối bỏ dữ liệu.

**Thuật toán chữ ký số RSA:** Phần này sẽ đi sâu vào chi tiết thuật toán RSA, bao gồm:Nguyên lý hoạt động: Giải thích cơ chế hoạt động của RSA dựa trên vấn đề khó tính toán phân tích thừa số nguyên tố lớn.Các bước tạo khóa: Mô tả chi tiết quá trình tạo cặp khóa công khai và khóa riêng.Các bước tạo và xác thực chữ ký: Giải thích chi tiết các bước tạo chữ ký số và quá trình xác thực chữ ký.Phân tích độ an toàn: Thảo luận về độ an toàn của thuật toán RSA và các tấn công tiềm tàng.

**Ứng dụng của chữ ký số RSA trong xác thực tính toàn vẹn dữ liệu:** Phần này sẽ tập trung vào việc minh họa cách thức sử dụng chữ ký số RSA để bảo vệ tính toàn vẹn của dữ liệu, bao gồm:

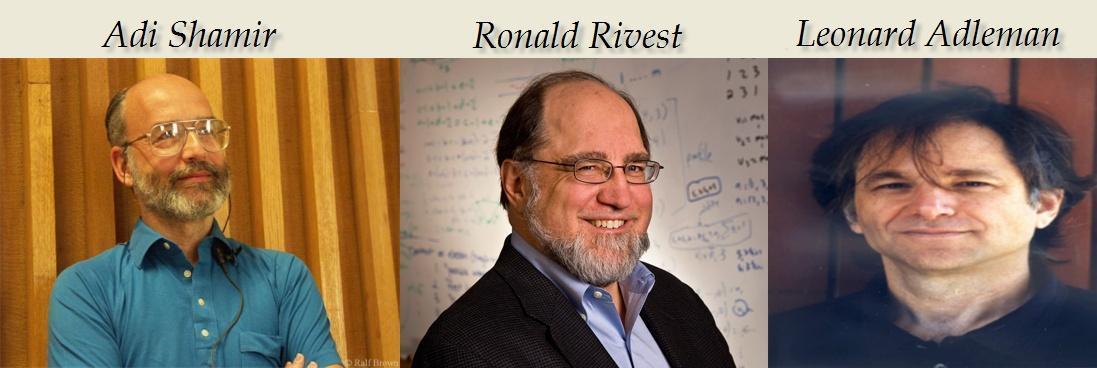
**Xác thực tính toàn vẹn của tập tin:** Mô tả cách sử dụng RSA để kiểm tra xem một tập tin có bị thay đổi hay không.Bảo mật giao tiếp: Giải thích cách RSA được sử dụng để bảo mật quá trình truyền dữ liệu giữa các hệ thống.Ví dụ cụ thể: Trình bày một số ví dụ minh họa cụ thể về ứng dụng của chữ ký số RSA trong các hệ thống thực tế.

**So sánh chữ ký số RSA với các thuật toán khác:** So sánh ưu điểm, nhược điểm của RSA so với các thuật toán chữ ký số khác như DSA, ECDSA, v.v. Đánh giá hiệu quả và tính khả thi của RSA trong các ứng dụng khác nhau.

**Thực trạng và xu hướng phát triển của chữ ký số RSA:** Phân tích thực trạng ứng dụng chữ ký số RSA hiện nay và dự báo xu hướng phát triển trong tương lai. Bao gồm cả việc thảo luận về các thách thức và giải pháp liên quan đến việc đảm bảo an toàn cho chữ ký số RSA.

## Giới thiệu về RSA

Tiêu chuẩn Rivest-Shamir-Adleman (RSA) - Giải thuật mã hóa khóa công khai RSA là một tiêu chuẩn được các tác giả Ronald Rivest, Adi Shamir và Leonard Adleman phát triển tại Học Viện Công nghệ Massachusetts (MIT) vào năm 1977, tên tiêu chuẩn được lấy từ 3 chữ cái đầu của tên 3 tác giả, hiện tiêu chuẩn được các tổ chức Viện Tiêu chuẩn Quốc gia Hoa Kỳ (American National Standards Institute – ANSI), Viện Kỹ nghệ Điện và Điện tử (Institute of Electrical and Electronics Engineers – IEEE) và Phòng thí nghiệm RSA công nhận (RSA Laboratories là một bộ phận của Tập đoàn EMC). Đây là thuật toán đầu tiên phù hợp với việc tạo ra [chữ ký điện tử](http://vi.wikipedia.org/wiki/Ch%E1%BB%AF_k%C3%BD_%C4%91i%E1%BB%87n_t%E1%BB%AD) đồng thời với việc [mã hóa](http://vi.wikipedia.org/wiki/M%C3%A3_h%C3%B3a). Nó đánh dấu một sự tiến bộ vượt bậc của lĩnh vực [mật mã học](http://vi.wikipedia.org/wiki/M%E1%BA%ADt_m%C3%A3_h%E1%BB%8Dc) trong việc sử dụng khóa công cộng. RSA đang được sử dụng phổ biến trong [thương mại điện tử](http://vi.wikipedia.org/wiki/Th%C6%B0%C6%A1ng_m%E1%BA%A1i_%C4%91i%E1%BB%87n_t%E1%BB%AD) và được cho là đảm bảo an toàn với điều kiện độ dài [khóa](http://vi.wikipedia.org/wiki/Kh%C3%B3a_%28m%E1%BA%ADt_m%C3%A3%29) đủ lớn.



Hình 1. 1 Các tác giả

Trước đó vài năm, Clifford Cox, một chuyên gia mã hóa người Anh đã phát triển riêng một biến thể của RSA. Tuy nhiên, Chính phủ Anh xem đây là vấn đề mật và đã không công bố. Khi Rivest, Shamir và Adleman công bố RSA trong ấn phẩm Scientific American tháng 9/1977, Cơ quan an ninh quốc gia Hoa Kỳ (NSA) không đồng ý về việc phổ biến rộng rãi RSA và ra lệnh cấm, tuy nhiên lệnh cấm này không có cơ sở pháp lý.

Thuật toán RSA được MIT đăng ký bằng sáng chế tại Hoa Kỳ vào năm [1983](http://vi.wikipedia.org/wiki/1983) (Số đăng ký 4,405,829). Bằng sáng chế này hết hạn vào ngày [21 tháng 9](http://vi.wikipedia.org/wiki/21_th%C3%A1ng_9) năm [2000](http://vi.wikipedia.org/wiki/2000). Tuy nhiên, do thuật toán đã được công bố trước khi có đăng ký bảo hộ nên sự bảo hộ hầu như không có giá trị bên ngoài Hoa Kỳ. Ngoài ra, nếu như công trình của Clifford Cocks đã được công bố trước đó thì bằng sáng chế RSA đã không thể được đăng ký.

Năm 1978, các tác giả đã công bố thuật toán trên Tạp chí của Hiệp hội Kỹ thuật Tính toán Hoa Kỳ (Communications of the Association for Computing Machinery – ACM). Hiện nay, có thể tham khảo đặc tả của RSA trên trang thông tin của Tập đoàn EMC.

# KẾT QUẢ NGHIÊN CỨU

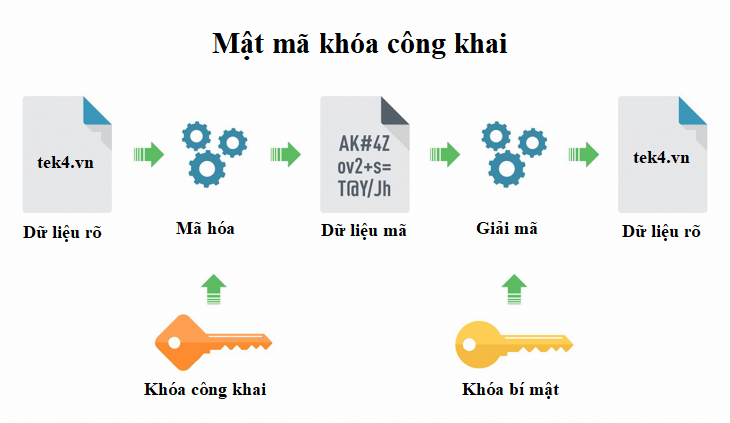
## Nghiên cứu, tìm hiểu hệ mã hóa khóa công khai

### *2.1.1. Khái niệm hệ mã hóa khóa công khai*

Hệ mã hóa khóa công khai (Public-Key Cryptography) là một công nghệ mã hóa trong đó sử dụng hai khóa khác nhau: khóa công khai và khóa bí mật. Khóa công khai được sử dụng để mã hóa dữ liệu hoặc xác minh chữ ký số, trong khi khóa bí mật được dùng để giải mã dữ liệu hoặc tạo chữ ký số.

**Khóa công khai (Public Key)**: Được công khai và chia sẻ rộng rãi. Dùng để mã hóa dữ liệu hoặc kiểm tra chữ ký số.

**Khóa bí mật (Private Key)**: Giữ kín bởi chủ sở hữu. Dùng để giải mã dữ liệu hoặc tạo chữ ký số.



Hình 2. 1 Sơ đồ biểu diễn khóa công khai

Điểm đặc biệt của hệ mã hóa này là khóa công khai có thể được chia sẻ công khai, nhưng khóa bí mật phải được giữ kín bởi chủ sở hữu. Sự kết hợp giữa hai khóa này tạo ra một hệ thống bảo mật mạnh mẽ, dựa trên các nguyên tắc toán học phức tạp.

### *2.1.2. Các đặc điểm của hệ mã hóa khóa công khai*

Hệ mã hóa khóa công khai có một số đặc điểm nổi bật:

**Tính bất đối xứng**: Hệ thống sử dụng hai khóa khác nhau cho hai quá trình mã hóa và giải mã, khác với hệ mã hóa đối xứng chỉ sử dụng một khóa duy nhất.

**Tính công khai**: Khóa công khai được chia sẻ tự do, giảm nguy cơ lộ thông tin nhạy cảm khi truyền khóa.

**Tính bảo mật cao**: Sự an toàn của hệ thống dựa trên các bài toán toán học phức tạp, như bài toán thừa số nguyên lớn hay logarit rời rạc.

**Ứng dụng đa dạng**: Được sử dụng trong xác minh danh tính, bảo vệ dữ liệu, và đảm bảo tính toàn vẹn của thông tin.

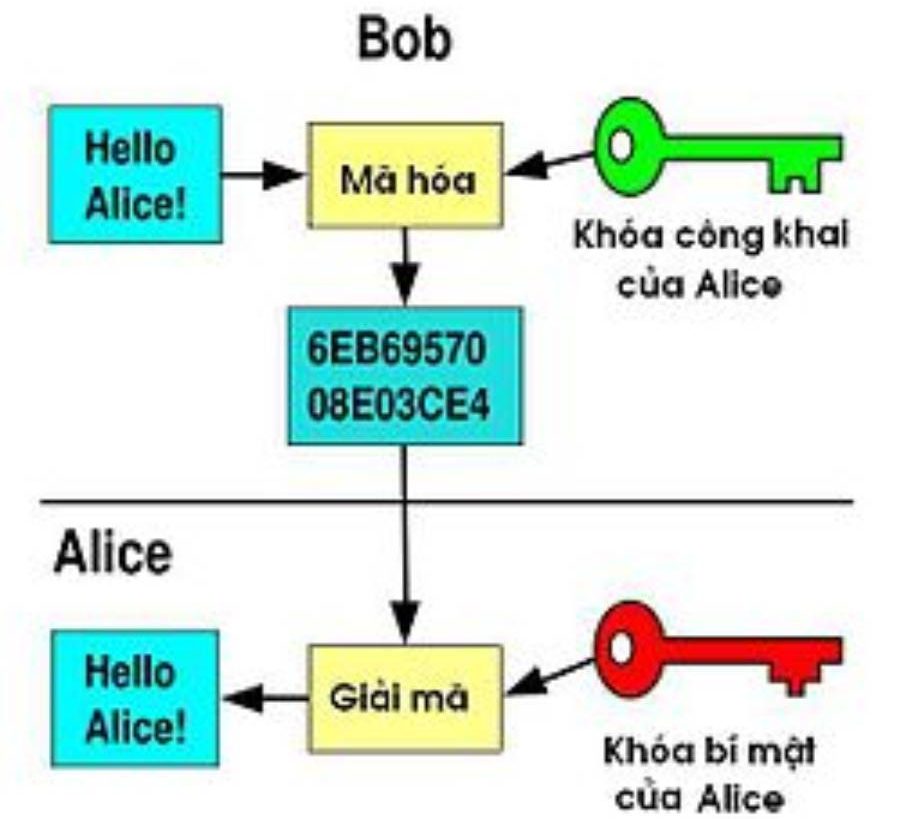
### *2.1.3. Nguyên lý hoạt động của hệ mã hóa khóa công khai*

Hệ mã hóa khóa công khai hoạt động dựa trên sự kết hợp của hai khóa toán học liên kết chặt chẽ với nhau. Cơ chế hoạt động bao gồm các bước:

**Sinh khóa**: Một cặp khóa được tạo, bao gồm khóa công khai và khóa bí mật. Khóa công khai được chia sẻ rộng rãi, trong khi khóa bí mật chỉ có chủ sở hữu nắm giữ.

**Mã hóa dữ liệu**: Thông điệp gốc được mã hóa bằng khóa công khai, biến thành một chuỗi dữ liệu mã hóa không thể đọc được.

**Giải mã dữ liệu**: Chỉ có khóa bí mật tương ứng mới có thể giải mã dữ liệu đã được mã hóa.



Hình 2. 2 Nguyên lý hoạt động của mã hóa khóa công khai

Ví dụ, khi xác minh danh tính, người gửi mã hóa thông điệp bằng khóa bí mật của họ. Người nhận dùng khóa công khai để giải mã, đảm bảo rằng thông điệp thực sự từ đúng người gửi.

### *2.1.4. Thuật toán RSA trong hệ mã hóa khóa công khai*

Thuật toán RSA (Rivest–Shamir–Adleman) là một trong những thuật toán mã hóa khóa công khai phổ biến nhất, được sử dụng để bảo mật và xác thực dữ liệu. RSA hoạt động dựa trên nguyên tắc sử dụng hai khóa riêng biệt: khóa công khai để mã hóa và khóa bí mật để giải mã.

Quá trình sinh khóa bắt đầu bằng việc chọn hai số nguyên tố lớn và tính toán các giá trị cần thiết để tạo ra cặp khóa công khai và bí mật. Thông điệp cần mã hóa sẽ được chuyển đổi thành bản mã bằng khóa công khai. Ngược lại, bản mã này chỉ có thể được giải mã bằng khóa bí mật, giúp đảm bảo dữ liệu được truyền tải an toàn và chỉ có người nhận dự định mới có thể đọc được thông điệp.

RSA được đánh giá cao nhờ vào độ an toàn dựa trên độ khó của việc phân tích thừa số nguyên tố của các số rất lớn, trở thành nền tảng cho nhiều ứng dụng bảo mật hiện đại như giao dịch trực tuyến và xác thực danh tính.

### *2.1.5.* ***Ưu điểm và nhược điểm của hệ mã hóa khóa công khai***

**Ưu điểm**:

Bảo mật cao: Khóa bí mật không cần truyền đi, giảm nguy cơ bị chặn.

Linh hoạt: Dễ dàng ứng dụng trong nhiều lĩnh vực như chữ ký số, mã hóa email, bảo vệ thông tin.

**Nhược điểm**:

Hiệu suất thấp: Các phép toán trên số lớn trong hệ mã hóa khóa công khai yêu cầu nhiều tài nguyên tính toán.

Khả năng bị phá vỡ: Nếu các số nguyên tố không đủ lớn hoặc bị lộ, tính bảo mật sẽ bị đe dọa.

### *2.1.6. Ứng dụng của hệ mã hóa khóa công khai trong thực tiễn*

Hệ mã hóa khóa công khai đã trở thành công cụ quan trọng trong bảo mật thông tin. Một số ứng dụng phổ biến:

**Chữ ký số**: Dùng để xác thực danh tính và đảm bảo tính toàn vẹn của dữ liệu trong các giao dịch điện tử.

**Bảo mật email**: Nội dung email được mã hóa bằng khóa công khai, chỉ người nhận có khóa bí mật mới có thể đọc được.

**Thương mại điện tử**: Đảm bảo an toàn trong giao dịch thanh toán trực tuyến, tránh bị đánh cắp thông tin.

**Chính phủ điện tử**: Bảo vệ thông tin nhạy cảm trong các dịch vụ công trực tuyến.

## Chữ ký số và ứng dụng trong thực tế

### *2.2.1. Định nghĩa và khái niệm về chữ ký số*

Chữ ký số là một dạng chữ ký điện tử sử dụng công nghệ mã hóa để xác nhận tính xác thực và toàn vẹn của thông tin trong giao dịch điện tử. Được tạo ra bằng cách sử dụng **khóa bí mật** của người ký, chữ ký số có thể được kiểm tra bằng **khóa công khai** tương ứng. Đây là một phần quan trọng của hệ thống mã hóa khóa công khai, đảm bảo rằng thông tin không bị thay đổi và thực sự đến từ nguồn đáng tin cậy.

Chữ ký số có ba đặc điểm chính:

**Tính xác thực**: Đảm bảo rằng dữ liệu đến từ đúng người ký.

**Tính toàn vẹn**: Bảo vệ dữ liệu không bị thay đổi trong quá trình truyền tải.

**Tính không thể phủ nhận**: Người ký không thể phủ nhận trách nhiệm đối với dữ liệu đã ký.

### *2.2.2.* ***Cách thức hoạt động của chữ ký số***

Chữ ký số hoạt động dựa trên thuật toán mã hóa khóa công khai. Cơ chế hoạt động bao gồm các bước cụ thể như sau:

**1.Tạo chữ ký số:**

Dữ liệu gốc được băm lại thành một **hàm băm (hash)** duy nhất bằng các thuật toán băm như SHA-256.Hàm băm này sau đó được mã hóa bằng **khóa bí mật** của người ký để tạo thành chữ ký số.

**2.Xác minh chữ ký số:**

Người nhận giải mã chữ ký số bằng **khóa công khai** của người ký để khôi phục hàm băm ban đầu.Đồng thời, người nhận tự băm lại dữ liệu gốc nhận được.So sánh hai hàm băm. Nếu giống nhau, chữ ký số hợp lệ và dữ liệu không bị thay đổi.

Bảng 2. 1 So sánh kết quả giữa hai hàm băm

|  |  |  |  |
| --- | --- | --- | --- |
| Bước | Hàm băm từ dữ liệu gốc | Hàm băm từ chữ ký số | Kết quả xác minh |
| Trước khi ký | 6F2C3A…. | - | - |
| Sau khi ký và giải mã | - | 6F2C3A…. | Trùng khớp |

Ví dụ, trong một giao dịch điện tử, người gửi sử dụng khóa bí mật để ký dữ liệu. Khi người nhận nhận được dữ liệu, họ sử dụng khóa công khai của người gửi để xác minh chữ ký và đảm bảo dữ liệu không bị thay đổi.

### *2.2.3. Các tiêu chuẩn chữ ký số phổ biến*

Chữ ký số được xây dựng dựa trên các tiêu chuẩn quốc tế và công nghệ tiên tiến để đảm bảo tính bảo mật và khả năng tương thích. Một số tiêu chuẩn phổ biến bao gồm:

**PKI (Public Key Infrastructure)**: Là hạ tầng khóa công khai hỗ trợ việc tạo, quản lý và phân phối khóa công khai và khóa bí mật. PKI là nền tảng cơ bản cho mọi hệ thống chữ ký số.

**X.509**: Một tiêu chuẩn quốc tế cho chứng chỉ số, giúp xác minh danh tính và đảm bảo an toàn trong giao dịch.

**RSA, ECDSA**: Các thuật toán phổ biến được sử dụng để tạo và xác minh chữ ký số, dựa trên các nguyên tắc toán học phức tạp.

### *2.2.4. Ưu điểm và nhược điểm của chữ ký số*

Chữ ký số có nhiều ưu điểm vượt trội nhưng cũng tồn tại một số hạn chế. Dưới đây là phân tích chi tiết:

**Ưu điểm**:

**An toàn và bảo mật cao**: Chữ ký số khó bị làm giả nhờ vào các thuật toán mã hóa mạnh mẽ. Khóa bí mật được giữ kín và không cần truyền đi, giúp giảm nguy cơ bị đánh cắp.

**Tiết kiệm thời gian và chi phí**: Loại bỏ nhu cầu sử dụng tài liệu giấy và chi phí gửi thư trong các giao dịch truyền thống.

**Phù hợp với pháp luật**: Chữ ký số được công nhận rộng rãi và có giá trị pháp lý ở nhiều quốc gia, đảm bảo tính pháp lý trong các giao dịch.

**Đảm bảo tính pháp lý**: Cung cấp bằng chứng xác thực danh tính và trách nhiệm của các bên tham gia.

**Nhược điểm**:

**Phụ thuộc vào cơ sở hạ tầng**: Việc triển khai và duy trì hệ thống PKI yêu cầu chi phí và quản lý phức tạp.

**Chi phí ban đầu cao**: Bao gồm chi phí phần mềm, chứng chỉ số và đào tạo nhân viên.

**Rủi ro quản lý khóa**: Nếu khóa bí mật bị mất hoặc lộ, toàn bộ hệ thống bảo mật sẽ bị đe dọa.

Bảng 2. 2 Liệt kê chi tiết ưu và nhược điểm

|  |  |
| --- | --- |
| Đặc điểm | Chi tiết |
| Ưu điểm | An toàn,bảo mật cao  Tiết kiệm thời gian và chi phí  Hợp pháp |
| Nhược điểm | Phụ thuộc vào cơ sở hạ tầng  Chi phí ban đầu cao  Quản lý khóa phức tạp |

### *2.2.5. Ứng dụng của chữ ký số trong thực tế*

Chữ ký số đã và đang được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau để đảm bảo an ninh thông tin và tính toàn vẹn dữ liệu. Một số ứng dụng nổi bật bao gồm:

**Thương mại điện tử**:

Đảm bảo tính bảo mật và xác thực trong các giao dịch mua bán trực tuyến.Sử dụng để chứng thực hóa đơn, hợp đồng và các tài liệu điện tử, giúp tăng niềm tin giữa các bên giao dịch.

**Chính phủ điện tử**:

Xác thực danh tính và bảo mật tài liệu trong các dịch vụ công trực tuyến như kê khai thuế, đăng ký kinh doanh và xử lý thủ tục hành chính.Đảm bảo tính toàn vẹn và bảo mật của các văn bản pháp lý và dữ liệu nhạy cảm.

**Ngân hàng và tài chính**:

Xác thực giao dịch trực tuyến và bảo vệ thông tin tài chính quan trọng.Ứng dụng trong hợp đồng vay vốn, bảo hiểm và các chứng từ tài chính khác.

**Y tế**:

Đảm bảo tính bảo mật và toàn vẹn của hồ sơ bệnh án điện tử.Hỗ trợ việc cấp phát thuốc và chia sẻ thông tin y tế an toàn giữa các cơ sở y tế.

**Doanh nghiệp và quản trị nội bộ**:

Chứng thực hợp đồng, email và các tài liệu nội bộ để đảm bảo tính minh bạch và hợp pháp trong quản lý doanh nghiệp.Hỗ trợ các quy trình phê duyệt tài liệu và hợp đồng từ xa.

**Học thuật**:

Chứng thực bằng cấp, chứng chỉ số và các tài liệu nghiên cứu học thuật.Đảm bảo tính bảo mật và toàn vẹn trong việc chia sẻ thông tin nghiên cứu giữa các tổ chức giáo dục.

Bảng 2. 3 Tổng hợp lĩnh vực ứng dụng

|  |  |  |
| --- | --- | --- |
| Lĩnh vực | Mục tiêu | Ví dụ cụ thể |
| Thương mại điện tử | Bảo mật giao dịch | Chứng thực hóa đơn điện tử |
| Chính phủ điện tử | Xác thực danh tính | Nộp thuế trực tuyến |
| Ngân hàng | Bảo mật tài chính | Xác thực giao dịch ngân hàng |
| Y tế | Bảo mật thông tin bệnh nhân | Hồ sơ bệnh án điện tử |
| Doanh nghiệp và quản trị nội bộ | Đảm bảo tính minh bạch và hợp pháp trong quản lý doanh nghiệp | Chứng thực hợp đồng, email và các tài liệu nội bộ |
| Học thuật | Bảo mật thông tin nghiên cứu | Chứng thực bằng cấp, chứng chỉ số và các tài liệu |

### *2.2.6. Chữ ký số và các thách thức hiện tại*

Mặc dù chữ ký số mang lại nhiều lợi ích, vẫn còn một số thách thức cần giải quyết để tối ưu hóa tính ứng dụng:

**Tấn công an ninh mạng**: Các phương pháp tấn công ngày càng tinh vi có thể làm giả chữ ký số nếu khóa bí mật bị lộ.

**Khả năng tương thích**: Không phải tất cả các hệ thống hoặc phần mềm đều hỗ trợ hoặc tương thích hoàn toàn với chữ ký số.

**Nhận thức của người dùng**: Nhiều cá nhân và tổ chức chưa hiểu rõ hoặc chưa nhận thức đầy đủ về lợi ích và cách sử dụng chữ ký số.

### *2.2.7. Vai trò của chữ ký số trong tương lai*

Chữ ký số sẽ tiếp tục đóng vai trò quan trọng trong việc đảm bảo an ninh thông tin và xác thực trong kỷ nguyên số. Một số xu hướng đáng chú ý bao gồm:

**Chữ ký số trên blockchain**: Kết hợp với công nghệ blockchain để tăng tính minh bạch, bảo mật và khả năng xác minh nguồn gốc dữ liệu.

**Chữ ký số di động**: Phát triển các giải pháp chữ ký số trên thiết bị di động, giúp người dùng thực hiện các giao dịch một cách tiện lợi mọi lúc, mọi nơi.

**Hỗ trợ trí tuệ nhân tạo (AI)**: Sử dụng AI để phát hiện các hành vi gian lận hoặc giả mạo chữ ký số, từ đó nâng cao tính bảo mật và hiệu quả.

## 2.**3. Nghiên cứu, tìm hiểu về chữ ký số rsa và ứng dụng trong xác thực tính toàn vẹn của dữ liệu**

*Định lý cơ sở:*

Định lý nhỏ của Fermat: Với p là một số nguyên tố khác 2 thì chia một số a lũy thừa p cho p sẽ có số dư chính bằng a:

Mở rộng ta có:

Với là số nguyên tố cùng nhau với m và nhỏ hơn m

Thuật toán RSA có 2 khóa:

Khóa công khai (Public Key)

Khóa bí mật (Private Key)

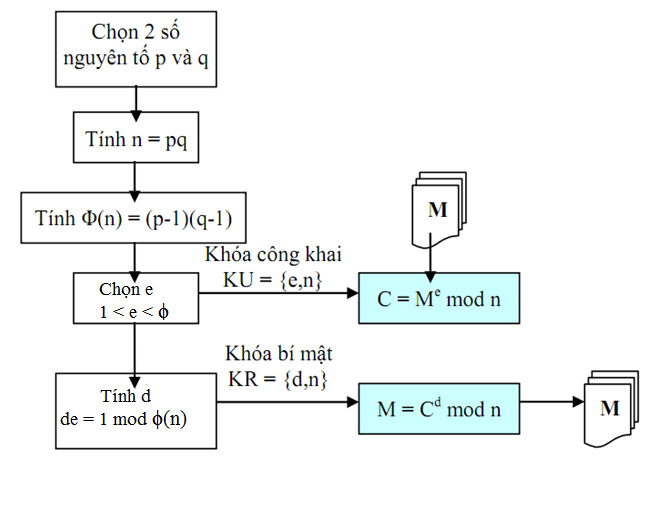
Mỗi khóa là những số cố định sử dụng trong quá trình mã hóa và giải mã.

Khóa công khai được công bố rộng rãi cho mọi người và được dùng để mã hóa.

Khóa bí mật được cá nhân giữ kín và dùng để giải mã

*Ví dụ:* Bod muốn gửi cho Alice một thông tin mật M mà Bod muốn duy nhất Alice có thể đọc được.

|  |  |  |  |
| --- | --- | --- | --- |
| Bod |  | | Alice |
| ( Mã hóa M bằng Public Key của A đc C) | | (Giải mã C bằng Private Key của A đc M ) | |



### *2.3.1. Quá trình tạo khóa cho hệ mật mã RSA*

Giả sử Alice và Bob cần trao đổi thông tin bí mật thông qua một kênh không an toàn (ví dụ như [Internet](http://vi.wikipedia.org/wiki/Internet)). Với thuật toán RSA, Alice đầu tiên cần tạo ra cho mình cặp khóa gồm khóa công khai và khóa bí mật theo 5 bước sau:

1. Chọn 2 số nguyên tố lớn khác nhau p, q thỏa mãn điều kiện
2. Tính tích của nó
3. Tính giá trị hàm Phi Euler của n:
4. Chọn số nguyên d, sao cho và
5. Tính giá trị e thỏa mãn điều kiện:

**Khóa công khai** bao gồm: n và e. **Khóa mật:** d còn p,q và A black background with a black square

Description automatically generated with medium confidencethường là xóa sau khi tính toán khóa.

### *2.3.2. Quá trình mã hóa*

Giả sử Bob muốn gửi đoạn thông tin M cho Alice. Đầu tiên Bob chuyển M thành một số m < n theo một hàm có thể đảo ngược (từ m có thể xác định lại M) được thỏa thuận trước. Quá trình này được mô tả ở phần Chuyển đổi văn bản rõ.

Lúc này Bob có m và biết n cũng như e do Alice gửi. Bob sẽ tính c là bản mã hóa của m theo công thức:

Hàm trên có thể tính dễ dàng sử dụng phương pháp tính hàm mũ (môđun) bằng phương pháp bình phương. Cuối cùng Bob gửi C cho Alice

### *2.3.3. Quá trình giải mã*

Alice nhận *c* từ Bob và khóa bí mật *d*. Alice có thể tìm được m từ *c* theo công thức sau:

Biết m, Alice tìm lại M theo phương pháp đã thỏa thuận trước. Quá trình giải mã hoạt động vì ta có:

Do và , theo [Định lý Fermat nhỏ](http://vi.wikipedia.org/w/index.php?title=%C4%90%E1%BB%8Bnh_l%C3%BD_Fermat_nh%E1%BB%8F&action=edit) nên:

Và

Do p và q là hai số nguyên tố cùng nhau, áp dụng định lý phần dư trung hoa, chúng ta có:

Hay

### *2.3.4. Ví dụ*

Sau đây là một ví dụ với những số cụ thể. Ở đây chúng ta sử dụng những số nhỏ để tiện tính toán còn trong thực tế phải dùng các số có giá trị đủ lớn.

Lấy:

p = 61 : số nguyên tố thứ nhất (giữ bí mật hoặc hủy sau khi tạo khóa)

q = 53 : số nguyên tố thứ hai (giữ bí mật hoặc hủy sau khi tạo khóa)

n = pq = 3233: môđun (công bố công khai)

e = 17 : số mũ công khai

d = 2753: số mũ bí mật

Khóa công khai là cặp (*e*, *n*). Khóa bí mật là *d*. Hàm mã hóa là:

encrypt(*m*) = *me* mod *n* = *m*17 mod 3233

với *m* là [văn bản rõ](http://vi.wikipedia.org/w/index.php?title=V%C4%83n_b%E1%BA%A3n_r%C3%B5&action=edit&redlink=1). Hàm giải mã là:

decrypt(*c*) = *cd* mod *n* = *c*2753 mod 3233

với *c* là [văn bản mã](http://vi.wikipedia.org/w/index.php?title=V%C4%83n_b%E1%BA%A3n_m%C3%A3&action=edit&redlink=1).

Để mã hóa văn bản có giá trị 123, ta thực hiện phép tính:

Encrypt (123) = 12317 mod 3233 = 855

Để giải mã văn bản có giá trị 855, ta thực hiện phép tính:

Decrypt (855) = 8552753 mod 3233 = 123

Cả hai phép tính trên đều có thể được thực hiện hiệu quả nhờ giải thuật bình phương và nhân

### *2.3.5. Chuyển đổi văn bản rõ*

Trước khi thực hiện mã hóa, ta phải thực hiện việc chuyển đổi văn bản rõ (chuyển đổi từ M sang m) sao cho không có giá trị nào của M tạo ra văn bản mã không an toàn.

Nếu không có quá trình này, RSA sẽ gặp phải một số vấn đề sau:

Nếu m = 0 hoặc m = 1 sẽ tạo ra các bản mã có giá trị là 0 và 1 tương ứng

Khi mã hóa với số mũ nhỏ (chẳng hạn e = 3) và m cũng có giá trị nhỏ, giá trị me cũng nhận giá trị nhỏ (so với n). Như vậy phép môđun không có tác dụng và có thể dễ dàng tìm được m bằng cách khai căn bậc e của c (bỏ qua môđun).

RSA là phương pháp mã hóa xác định (không có thành phần ngẫu nhiên) nên kẻ tấn công có thể thực hiện tấn công lựa chọn bản rõ bằng cách tạo ra một bảng tra giữa bản rõ và bản mã. Khi gặp một bản mã, kẻ tấn công sử dụng bảng tra để tìm ra bản rõ tương ứng.

Trên thực tế, ta thường gặp 2 vấn đề đầu khi gửi các bản tin ASCII ngắn với m là nhóm vài ký tự ASCII. Một đoạn tin chỉ có 1 ký tự NULL sẽ được gán giá trị m = 0 và cho ra bản mã là 0 bất kể giá trị của e và N. Tương tự, một ký tự ASCII khác, SOH, có giá trị 1 sẽ luôn cho ra bản mã là 1. Với các hệ thống dùng giá trị e nhỏ thì tất cả ký tự ASCII đều cho kết quả mã hóa không an toàn vì giá trị lớn nhất của m chỉ là 255 và 2553 nhỏ hơn giá trị n chấp nhận được. Những bản mã này sẽ dễ dàng bị phá mã.

Để tránh gặp phải những vấn đề trên, RSA trên thực tế thường bao gồm một hình thức chuyển đổi ngẫu nhiên hóa m trước khi mã hóa. Quá trình chuyển đổi này phải đảm bảo rằng m không rơi vào các giá trị không an toàn. Sau khi chuyển đổi, mỗi bản rõ khi mã hóa sẽ cho ra một trong số khả năng trong tập hợp bản mã. Điều này làm giảm tính khả thi của phương pháp tấn công lựa chọn bản rõ (một bản rõ sẽ có thê tương ứng với nhiều bản mã tùy thuộc vào cách chuyển đổi).

Một số tiêu chuẩn, chẳng hạn như PKCS, đã được thiết kế để chuyển đổi bản rõ trước khi mã hóa bằng RSA. Các phương pháp chuyển đổi này bổ xung thêm bit vào M. Các phương pháp chuyển đổi cần được thiết kế cẩn thận để tránh những dạng tấn công phức tạp tận dụng khả năng biết trước được cấu trúc của bản rõ. Phiên bản ban đầu của PKCS dùng một phương pháp đặc ứng (ad-hoc) mà về sau được biết là không an toàn trước tấn công lựa chọn bản rõ thích ứng (adaptive chosen ciphertext attack). Các phương pháp chuyển đổi hiện đại sử dụng các kỹ thuật như chuyển đổi mã hóa bất đối xứng tối ưu như (Optimal Asymmetric Encryption Padding – OAEP) để chống lại tấn công dạng này. Tiêu chuẩn PKCS còn được bổ xung các tính năng khác để đảm bảo an toàn cho chữ ký RSA (Probabilistic Signature Scheme for RSA-RSA-PSS).

### *2.3.6. Tạo chữ ký vào văn bản*

Thuật toán RSA còn được dùng để tạo chữ ký số cho văn bản.

Giả sử Alice muốn gửi cho Bob một văn bản có chữ ký của mình. Để làm việc này, Alice tạo ra một ***giá trị băm* (**hash value) của văn bản cần ký và tính giá trị mũ **d mod n** của nó (giống như khi Alice thực hiện giải mã). Giá trị cuối cùng chính là chữ ký điện tử của văn bản đang xét.

Khi Bob nhận được văn bản cùng với chữ ký điện tử, anh ta tính giá trị mũ **e mod n** của chữ ký đồng thời với việc tính giá trị băm của văn bản. Nếu 2 giá trị này như nhau thì Bob biết rằng người tạo ra chữ ký biết khóa bí mật của Alice và văn bản đã không bị thay đổi sau khi ký.

Cần chú ý rằng các phương pháp chuyển đổi bản rõ (như RSA-PSS) giữ vai trò quan trọng đối với quá trình mã hóa cũng như chữ ký điện tử và không được dùng chung cho đồng thời cho cả hai mục đích trên.

## 2.4. Thiết kế chương trình và thuật toán

### *2.4.1. Thiết kế kịch bản chương trình*

a. Mục tiêu của chương trình

Chương trình tập trung triển khai thuật toán RSA để thực hiện:

Sinh khóa RSA: Tạo khóa công khai và khóa bí mật để ký số và xác thực.

Ký số: Tạo chữ ký số từ dữ liệu/tài liệu đầu vào bằng khóa bí mật.

Xác thực chữ ký số: Sử dụng khóa công khai để kiểm tra tính hợp lệ của chữ ký số, đồng thời đảm bảo dữ liệu không bị thay đổi (tính toàn vẹn).

b. Chức năng chính của chương trình

Tạo cặp khóa: Người dùng khởi tạo cặp khóa RSA với độ dài tùy chọn (1024, 2048, hoặc 4096 bit).Khóa công khai và khóa bí mật sẽ được lưu dưới dạng file hoặc xuất ra màn hình.

Ký số: Người dùng cung cấp tài liệu cần ký (dạng văn bản, tệp tin, hoặc chuỗi dữ liệu).Hệ thống sinh ra hash của tài liệu và sử dụng khóa bí mật để tạo chữ ký số.Chữ ký số được lưu lại hoặc xuất ra màn hình.

Xác thực chữ ký số: Người dùng nhập tài liệu, chữ ký số và khóa công khai.Chương trình sẽ kiểm tra: Hash của tài liệu có khớp với hash từ chữ ký số hay không.Nếu hợp lệ, thông báo "Tài liệu toàn vẹn."Nếu không hợp lệ, thông báo "Tài liệu đã bị thay đổi."

c. Luồng hoạt động của chương trình

Dưới đây là luồng hoạt động mô tả từng bước:

1. Khởi động chương trình:

Chương trình hiển thị menu chức năng:

* + 1. Tạo cặp khóa RSA.
    2. Ký số tài liệu.
    3. Xác thực chữ ký số.
    4. Thoát chương trình.

1. Tạo cặp khóa:

Người dùng chọn tùy chọn “Tạo cặp khóa”.

Chương trình yêu cầu người dùng nhập độ dài khóa (1024, 2048, hoặc 4096 bit).

Sinh khóa công khai và bí mật.

Lưu trữ hai khóa dưới dạng file (PublicKey.txt, PrivateKey.txt).

1. Ký số tài liệu:

Người dùng chọn “Ký số tài liệu”.

Chương trình yêu cầu:

Tài liệu cần ký (nhập trực tiếp hoặc chọn file).

File chứa khóa bí mật (PrivateKey.txt).

Hệ thống thực hiện:

Tạo hash của tài liệu (SHA-256).

Mã hóa hash bằng khóa bí mật để tạo chữ ký số.

Kết quả:

Xuất chữ ký số ra file (Signature.txt).

1. Xác thực chữ ký số:

Người dùng chọn “Xác thực chữ ký số”.

Chương trình yêu cầu:

Tài liệu gốc.

Chữ ký số (Signature.txt).

File chứa khóa công khai (PublicKey.txt).

Hệ thống thực hiện:

Tạo hash từ tài liệu gốc.

Giải mã chữ ký số bằng khóa công khai để lấy hash.

So sánh hai hash:

* + - * Trùng khớp: Tài liệu toàn vẹn.
      * Không trùng khớp: Tài liệu bị thay đổi.

1. Kết thúc chương trình:

Người dùng chọn “Thoát chương trình” để đóng ứng dụng.

d. Giao diện chương trình

1. Command-line interface (CLI):

Menu các chức năng với các tùy chọn.

Nhập liệu và hiển thị kết quả trên terminal.

Ví dụ giao diện CLI:

================================

Hệ thống RSA - Chữ ký số

1. Tạo cặp khóa RSA

2. Ký số tài liệu

3. Xác thực chữ ký số

4. Thoát

================================

Nhập lựa chọn của bạn: \_

1. Giao diện đồ họa (GUI) - Tùy chọn bổ sung:

Một cửa sổ ứng dụng cơ bản với các button:

“Tạo khóa”

“Ký tài liệu”

“Xác thực chữ ký”

“Thoát”

### *2.4.2. Giới thiệu ngôn ngữ lập trình sử dụng để cài đặt thuật toán*

#### 2.4.2.1. Lý do lựa chọn ngôn ngữ lập trình

Hai ngôn ngữ lập trình được sử dụng để cài đặt thuật toán RSA trong bài toán này là **Java** và **C#**. Cả hai đều phù hợp cho các ứng dụng yêu cầu bảo mật cao nhờ vào:

**Hỗ trợ thư viện mạnh mẽ:** Java và C# đều cung cấp các thư viện mã hóa đã được tối ưu hóa, hỗ trợ đầy đủ các chức năng cần thiết như sinh khóa, ký số, và xác thực chữ ký.

**Khả năng phát triển đa nền tảng:** Java chạy trên mọi hệ điều hành thông qua JVM tích hợp tốt trong hệ sinh thái Windows và .NET Core giúp chạy trên Linux/MacOS.

**Dễ triển khai:** Cả hai ngôn ngữ đều có cấu trúc rõ ràng, dễ học và cung cấp tài liệu phong phú.

#### 2.4.2.2. Lựa chọn ngôn ngữ Java

**Ưu điểm:**

**Đa nền tảng:** Một ứng dụng Java có thể chạy trên bất kỳ hệ điều hành nào hỗ trợ JVM.

**Hệ thống thư viện mã hóa mạnh mẽ:** Java cung cấp gói java.security và javax.crypto, giúp triển khai dễ dàng các thuật toán mã hóa và chữ ký số.

**Bảo mật:** Quản lý bộ nhớ tự động và sandboxing giúp hạn chế lỗi bảo mật do tràn bộ nhớ.

**Công cụ và thư viện hỗ trợ trong Java:**

**KeyPairGenerator:** Để sinh cặp khóa RSA (khóa công khai và khóa bí mật).

**Signature:** Cung cấp chức năng ký và xác thực chữ ký số.

**MessageDigest:** Tạo hash bằng các thuật toán như SHA-256 để đảm bảo tính toàn vẹn của dữ liệu.

**Base64:** Để mã hóa và giải mã chuỗi khóa hoặc chữ ký số.

**Quy trình triển khai RSA bằng Java:**

**Bước 1:** Sinh cặp khóa bằng KeyPairGenerator.

**Bước 2:** Sử dụng Signature để ký dữ liệu bằng khóa bí mật.

**Bước 3:** Kiểm tra chữ ký bằng khóa công khai và đảm bảo tính toàn vẹn của dữ liệu.

**Mã minh họa (Tóm tắt):**

// Sinh cặp khóa RSA

KeyPairGenerator keyPairGen = KeyPairGenerator.getInstance("RSA");

keyPairGen.initialize(2048);

KeyPair pair = keyPairGen.generateKeyPair();

// Ký tài liệu

Signature signature = Signature.getInstance("SHA256withRSA");

signature.initSign(pair.getPrivate());

signature.update(data.getBytes());

byte[] digitalSignature = signature.sign();

// Xác thực chữ ký

signature.initVerify(pair.getPublic());

signature.update(data.getBytes());

boolean isVerified = signature.verify(digitalSignature);

#### 2.4.2.3. Lựa chọn ngôn ngữ C#

**Ưu điểm:**

**Tích hợp tốt với hệ sinh thái Windows:** Hỗ trợ lập trình GUI nhanh chóng qua WinForms hoặc WPF.

**Hệ thống mã hóa dễ sử dụng:** C# cung cấp các lớp như RSACryptoServiceProvider và SHA256Managed trong System.Security.Cryptography để thực hiện các tác vụ liên quan đến RSA.

**Hỗ trợ tài liệu phong phú:** Các API trong .NET Framework được ghi chú rõ ràng, dễ sử dụng.

**Công cụ và thư viện hỗ trợ trong C#:**

**RSACryptoServiceProvider:** Thực hiện sinh khóa, mã hóa, giải mã, và ký số.

**SHA256Managed:** Tạo hash của dữ liệu để đảm bảo tính toàn vẹn.

**FileStream:** Đọc/ghi dữ liệu từ các tệp chứa khóa, chữ ký, hoặc tài liệu.

**Quy trình triển khai RSA bằng C#:**

**Bước 1:** Sinh cặp khóa RSA bằng RSACryptoServiceProvider.

**Bước 2:** Sử dụng RSACryptoServiceProvider để ký dữ liệu bằng khóa bí mật.

**Bước 3:** Xác thực chữ ký bằng cách giải mã chữ ký số và so sánh hash.

**Mã minh họa (Tóm tắt):**

// Sinh cặp khóa RSA

using (RSACryptoServiceProvider rsa = new RSACryptoServiceProvider(2048))

{

string publicKey = rsa.ToXmlString(false); // Chỉ lấy khóa công khai

string privateKey = rsa.ToXmlString(true); // Lấy cả khóa công khai và bí mật

}

// Ký tài liệu

byte[] dataBytes = Encoding.UTF8.GetBytes(data);

byte[] hash = new SHA256Managed().ComputeHash(dataBytes);

byte[] signature = rsa.SignHash(hash, CryptoConfig.MapNameToOID("SHA256"));

// Xác thực chữ ký

bool isVerified = rsa.VerifyHash(hash, CryptoConfig.MapNameToOID("SHA256"), signature);

#### 2.4.2.4. So sánh giữa Java và C#:

| **Tiêu chí** | **Java** | **C#** |
| --- | --- | --- |
| **Đa nền tảng** | Tốt (chạy trên mọi hệ điều hành có JVM). | Hạn chế hơn, tối ưu trên Windows. |
| **Thư viện mã hóa** | Nhiều và đa dạng. | Dễ sử dụng, tích hợp tốt trong .NET. |
| **Giao diện người dùng** | Phải sử dụng JavaFX hoặc Swing. | Phát triển giao diện GUI dễ dàng với WinForms hoặc WPF. |
| **Hiệu năng** | Tốt cho xử lý dữ liệu lớn. | Hiệu năng cao khi chạy trên hệ sinh thái Windows. |

### 2.4.3. Cài đặt thuật toán và giao diện cho đề tài

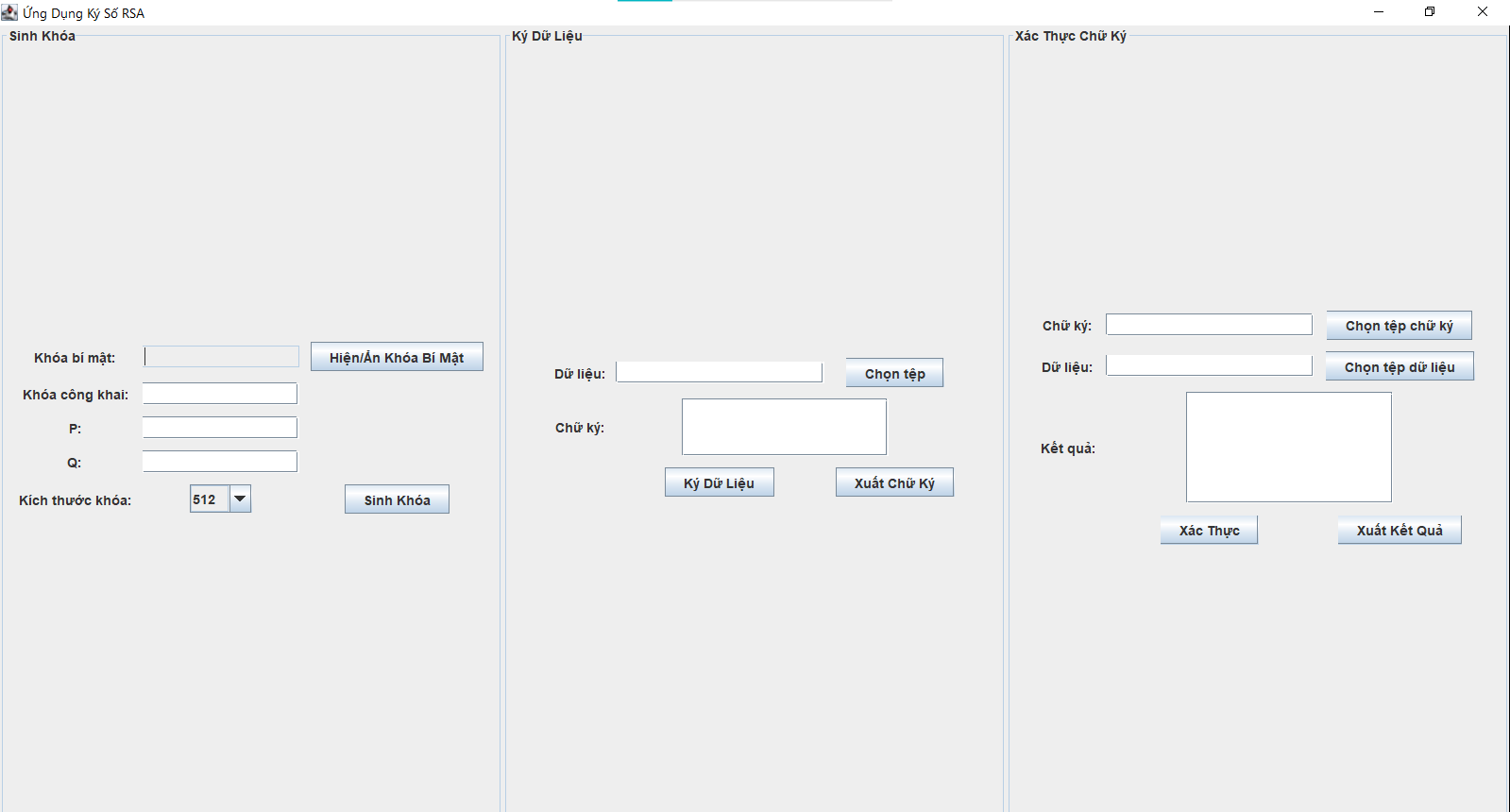
#### 2.4.3.1. Theo ngôn ngữ Java

**Cài đặt thuật toán:**



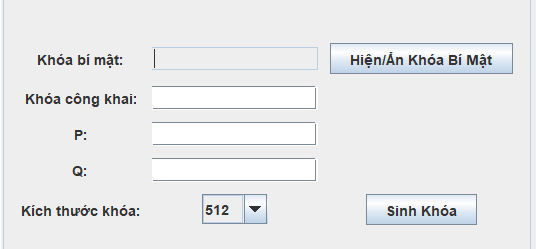
Hình 2. 3 Thuật toán Java

**Giao diện chương trình:**

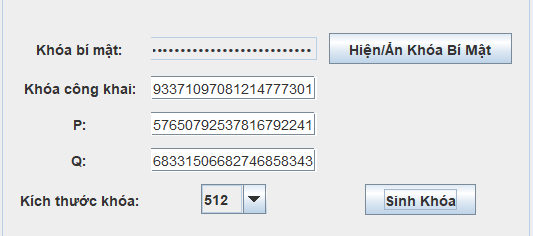


Hình 2. 4 Giao diện chương trình Java

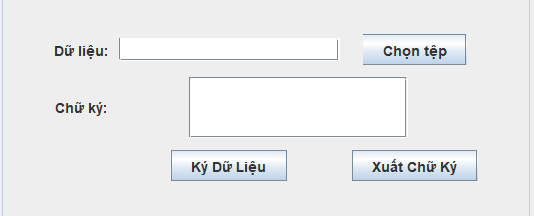
**Sinh khóa:**



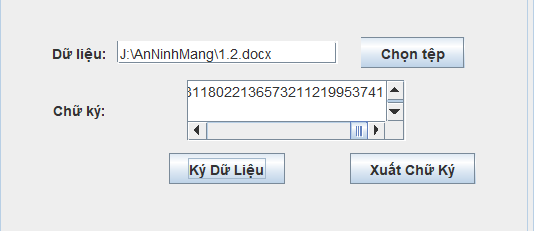
Hình 2. 5 Sinh khóa



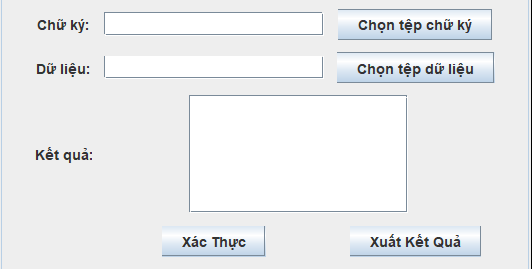
Hình 2. 6 Sinh khóa thành công

**Mã hóa:** ****

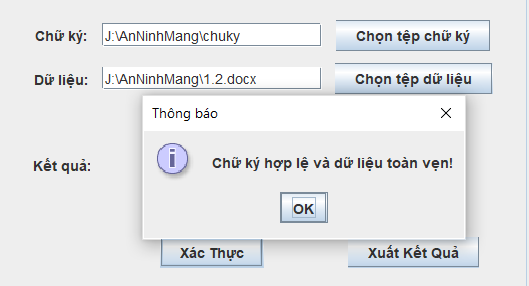
Hình 2. 7 Mã hóa

****

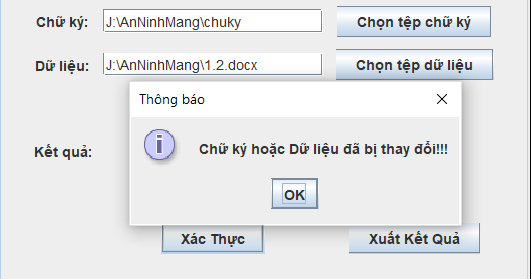
Hình 2. 8 Mã hóa thành công

**Giải mã**

Hình 2. 9 Giải mã

****

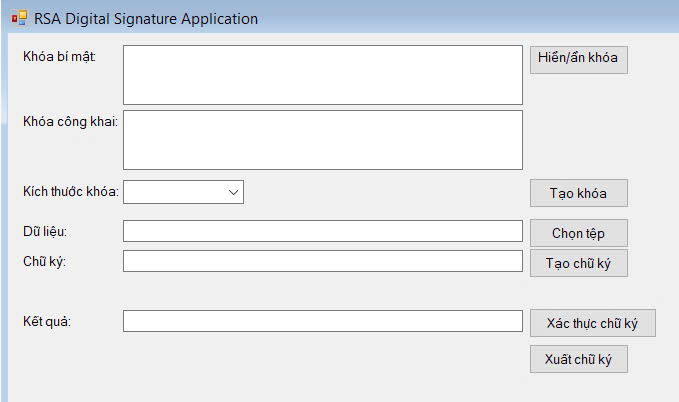
Hình 2. 10 Giải mã thành công



Hình 2. 11 Giải mã thất bại

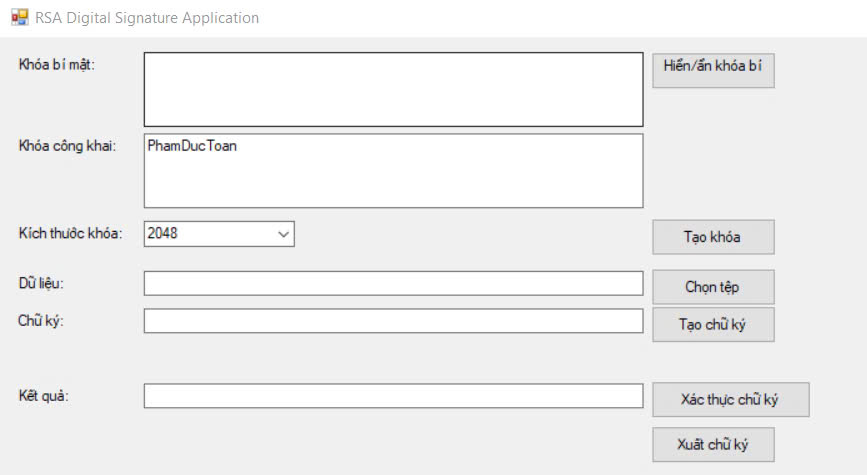
#### 2.4.3.2. Theo ngôn ngữ C#

**Giao diện chương trình:**

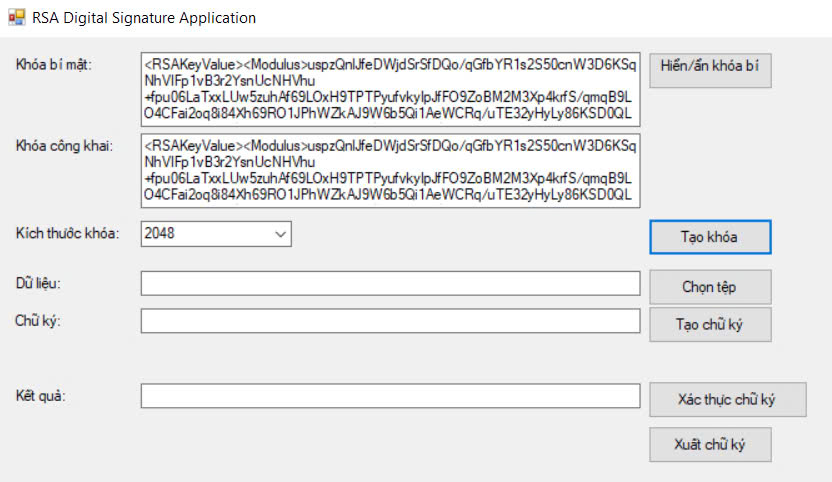


Hình 2. 12 Giao diện chương trình C#

**Chọn kích thước khóa và tạo khóa**

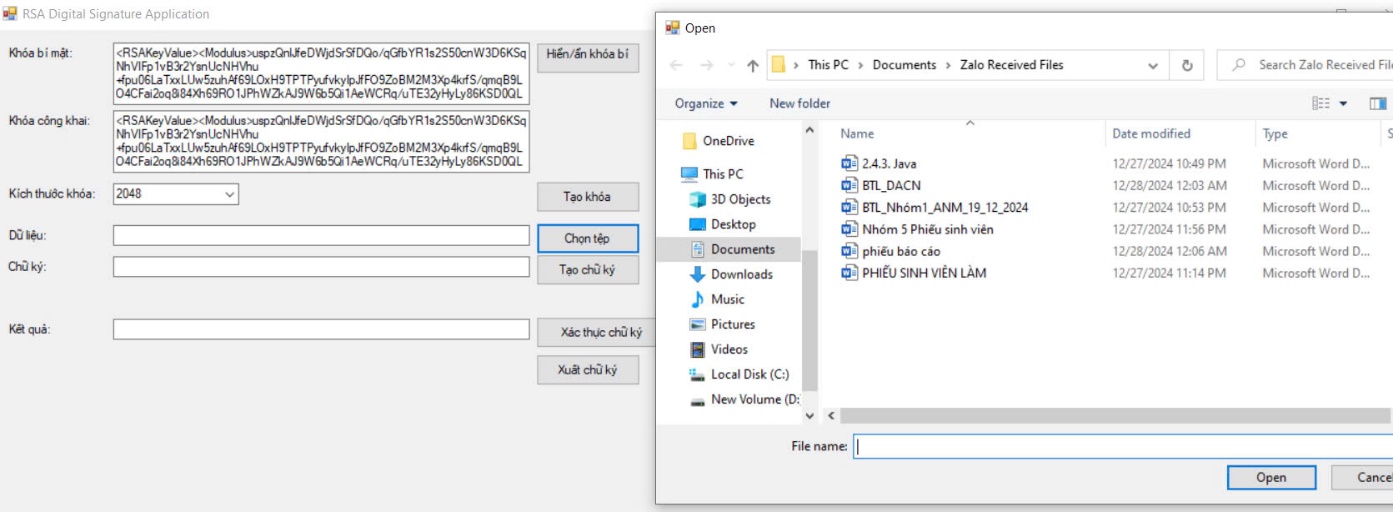


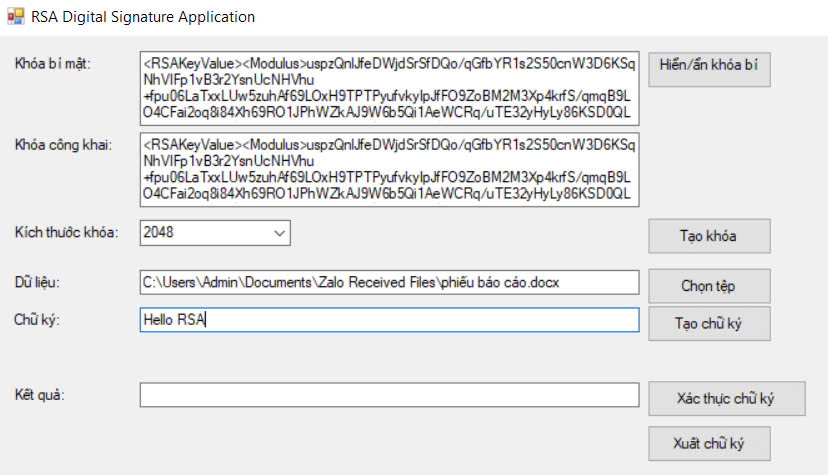
Hình 2. 13 Chọn kích thước khóa



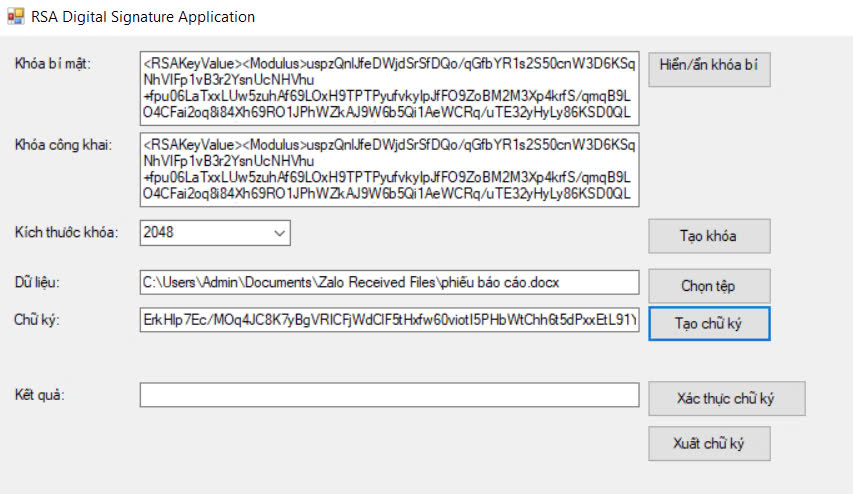
Hình 2. 14 Tạo khóa thành công

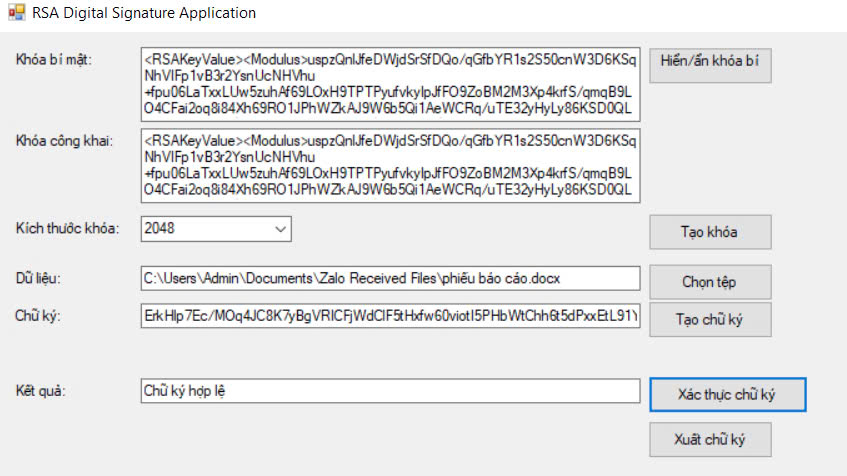
**Chọn dữ liệu từ tệp và tạo chữ kí**

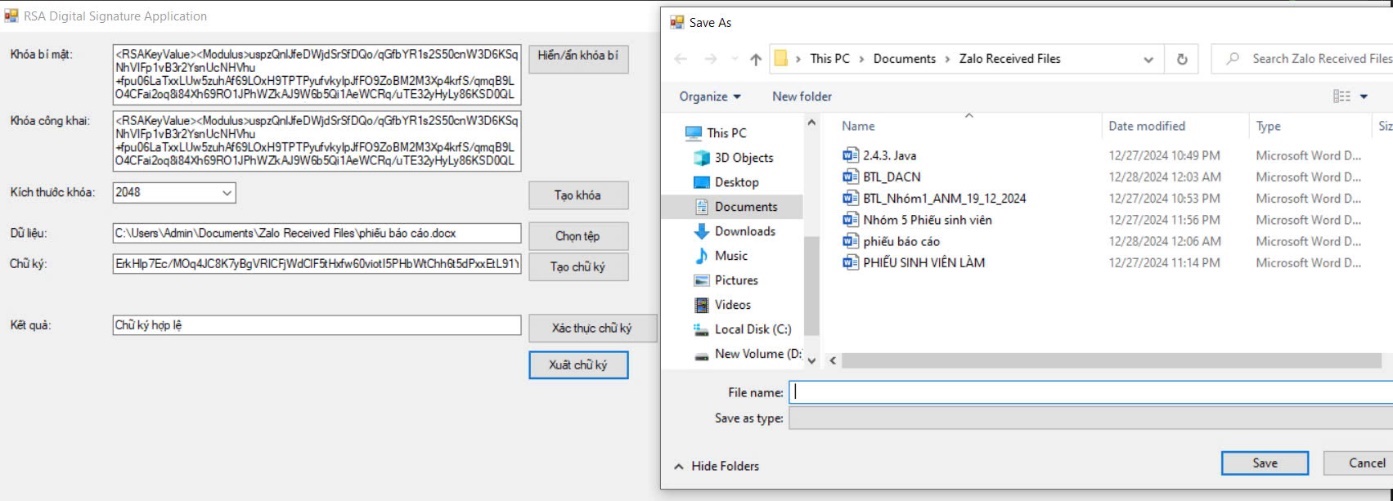
Hình 2. 15 Chọn tệp

****

Hình 2. 16 Tạo chữ kí

Hình 2. 17 Mã hóa chữ ký

Hình 2. 18 Xác thực chữ ký



Hình 2. 19 Xuất chữ ký ra tệp

# KẾT LUẬN VÀ BÀI HỌC KINH NGHIỆM

## Kiến thức kỹ năng đã học được trong quá trình thực hiện đề tài.

### *3.1.1. Kiến thức chuyên môn*

**Mật mã học và vai trò của chữ ký số:**Chữ ký số không chỉ đảm bảo tính toàn vẹn của dữ liệu mà còn xác thực danh tính người gửi. Nắm rõ 3 yếu tố chính:Tính toàn vẹn dữ liệu: Đảm bảo dữ liệu không bị thay đổi trong quá trình truyền tải.Xác thực danh tính: Người nhận có thể chắc chắn rằng dữ liệu đến từ đúng người gửi (dựa vào khóa công khai).Chống chối bỏ: Người gửi không thể phủ nhận việc họ đã ký dữ liệu.

**Nguyên lý RSA:Tạo khóa:**Phép toán cơ bản: Chọn 2 số nguyên tố lớn p, q để tính n=p×q. Chọn số e sao cho ước chung lớn nhất gcd(e,ϕ(n))=1 , sau đó tính d là nghịch đảo của e theo modulo ϕ(n)Khóa công khai: (e,n), khóa bí mật: (d,n).Ký dữ liệu: Băm dữ liệu bằng hàm băm (SHA-256), sau đó mã hóa băm bằng khóa bí mật (S=H(m)^d mod  n).Xác minh: Giải mã chữ ký bằng khóa công khai và so sánh với băm của dữ liệu gốc.

**Ứng dụng thực tế:**Sử dụng trong hệ thống ký tài liệu PDF, giao dịch ngân hàng, xác thực trong blockchain, hoặc ký hợp đồng điện tử.

### *3.1.2. Kỹ năng kỹ thuật*

**Lập trình C# với RSA**

**Sinh khóa và lưu trữ:**Sử dụng RSA.ExportParameters() để xuất khóa dưới dạng cấu trúc dữ liệu hoặc tệp XML:

**Tạo chữ ký số:**Tích hợp hàm băm SHA256 để tăng tính an toàn:

**Xác minh chữ ký:**Tải khóa công khai từ tệp và thực hiện xác minh:

**Xử lý lỗi và bảo mật:**Mã hóa khóa bí mật khi lưu trữ bằng AES hoặc bảo vệ bằng **Windows Data Protection API (DPAPI)**.

**Triển khai ứng dụng:**

**Giao diện người dùng (Windows Forms hoặc WPF):**

Một giao diện cơ bản có các chức năng:Nút "Tạo khóa" để sinh và lưu khóa RSA.Nút "Tạo chữ ký" để ký dữ liệu nhập từ người dùng.Nút "Xác minh chữ ký" để kiểm tra dữ liệu và chữ ký.

**Tích hợp lưu trữ:**

Lưu chữ ký số dưới dạng file .sig và kiểm tra trên các dữ liệu đã lưu.

**Kỹ năng mềm**

**Nghiên cứu tài liệu:**Tìm hiểu sâu về các thuật toán liên quan trong C# qua tài liệu:NET API Documentation: System.Security.Cryptography.Các bài báo khoa học về bảo mật RSA và các cải tiến.Dùng Zoom, Google Meet hoặc Zalo để trao đổi công việc và cập nhật tiến độ.

### *3.1.3. Kinh nghiệm thực tế*

**Thử nghiệm và đánh giá:**

**Hiệu suất:**Đánh giá tốc độ xử lý của RSA với các kích thước khóa khác nhau (1024, 2048, 4096 bit).Phân tích mức độ an toàn của chữ ký số khi sử dụng các hàm băm khác nhau (SHA-1, SHA-256).

**Bảo mật:**Mô phỏng các tình huống tấn công phổ biến:Tấn công brute force vào khóa.Tấn công replay (phát lại dữ liệu ký).

**Ứng dụng:**Triển khai thử nghiệm:Ký một tài liệu PDF và xác minh chữ ký.

Gửi dữ liệu qua socket hoặc API và xác thực chữ ký số tại phía nhận.

**Hạn chế và giải pháp:**

**Hạn chế:**RSA hiệu suất không cao với dữ liệu lớn,rủi ro nếu khóa bí mật bị lộ.

**Giải pháp:**Sử dụng RSA kết hợp mã hóa đối xứng (AES) cho dữ liệu lớn.Áp dụng các giao thức bảo mật bổ sung như TLS để bảo vệ khóa khi truyền tải.

### *3.1.4. Trình bày và báo cáo*

**Cấu trúc báo cáo:**Mở đầu: Giới thiệu RSA và chữ ký số.

Nội dung chính:Nguyên lý hoạt động của RSA,quy trình thực hiện (từ lý thuyết đến lập trình).

Kết quả:Mã nguồn minh họa.Ảnh chụp giao diện ứng dụng hoặc kết quả thử nghiệm.

Kết luận: Bài học kinh nghiệm và hướng phát triển.

**Minh họa thực tế:**Biểu đồ so sánh thời gian ký và xác minh với các kích thước khóa khác nhau.Sơ đồ quy trình hoạt động của hệ thống chữ ký số.

## Bài học kinh nghiệm

**Tầm quan trọng của mật mã trong thế giới số:**

**Bảo vệ thông tin:** Mật mã, đặc biệt là các thuật toán như RSA, đóng vai trò thiết yếu trong việc bảo vệ thông tin nhạy cảm khỏi những truy cập trái phép. Nó giúp chúng ta duy trì sự riêng tư và an toàn trong môi trường số.

**Xây dựng niềm tin:** Chữ ký số không chỉ bảo vệ dữ liệu mà còn tạo dựng niềm tin giữa các bên tham gia giao dịch trực tuyến. Việc xác thực nguồn gốc và tính toàn vẹn của dữ liệu là nền tảng cho các hoạt động thương mại điện tử, chính phủ điện tử và nhiều lĩnh vực khác.

**Hiểu rõ nguyên lý hoạt động của thuật toán mật mã:**

**Không đơn giản:** Các thuật toán mật mã như RSA dựa trên những nguyên lý toán học phức tạp. Việc hiểu rõ cách chúng hoạt động, đặc biệt là các khái niệm như khóa công khai, khóa bí mật, và hàm băm, là rất quan trọng để ứng dụng chúng một cách hiệu quả.

**Cập nhật kiến thức:** Lĩnh vực mật mã luôn phát triển. Việc theo dõi và cập nhật những tiến bộ mới là cần thiết để đảm bảo tính bảo mật của hệ thống.

**Ứng dụng thực tiễn và đa dạng của chữ ký số:**

**Không giới hạn:** Chữ ký số không chỉ giới hạn trong việc xác thực tài liệu. Nó có thể được ứng dụng rộng rãi trong nhiều lĩnh vực như: giao dịch tài chính, ký kết hợp đồng trực tuyến, xác thực phần mềm, bảo mật email, và nhiều hơn nữa.

**Giải pháp toàn diện:** Chữ ký số RSA, kết hợp với các công nghệ khác, tạo thành một giải pháp toàn diện để bảo vệ dữ liệu và xây dựng lòng tin trong thế giới số.

**Cân nhắc về hiệu năng và bảo mật:**

**Đánh đổi:** Việc sử dụng mật mã có thể tạo ra gánh nặng về hiệu năng cho hệ thống. Cần cân nhắc giữa mức độ bảo mật mong muốn và hiệu suất hoạt động để lựa chọn giải pháp phù hợp.

**Không có gì là tuyệt đối:** Dù RSA là một thuật toán mạnh mẽ, không có hệ thống nào là hoàn toàn bất khả xâm phạm. Việc thường xuyên đánh giá, kiểm tra và cập nhật hệ thống là rất quan trọng để đảm bảo an toàn.

**Tư duy hệ thống và toàn diện:**

**Không chỉ là thuật toán:** Việc triển khai chữ ký số đòi hỏi một cái nhìn hệ thống, bao gồm việc lựa chọn thuật toán phù hợp, quản lý khóa an toàn, xây dựng cơ chế xác thực, và đào tạo người dùng.

**Phối hợp nhiều yếu tố:** Bảo mật không chỉ là vấn đề công nghệ, mà còn liên quan đến con người và quy trình. Cần có sự phối hợp giữa nhiều yếu tố để đạt được hiệu quả tối ưu.

## Tính khả thi của chủ đề nghiên cứu, những thuận lợi, khó khăn

### *3.3.1. Tính khả thi*

#### 3.3.1.1. Tính khả thi về mặt kỹ thuật

**C# hỗ trợ mạnh mẽ cho RSA và mật mã học:**Ngôn ngữ C# và .NET Framework có thư viện tích hợp sẵn như System.Security.Cryptography, hỗ trợ đầy đủ cho việc triển khai RSA và các thuật toán liên quan (SHA-256, PKCS#1).Không cần phải xây dựng toàn bộ giải thuật từ đầu, giúp tập trung vào ứng dụng và tối ưu hóa.

**Tài liệu tham khảo phong phú:**Có nhiều tài liệu hướng dẫn lập trình C# liên quan đến mật mã học, chữ ký số, và cách sử dụng RSA.Các diễn đàn lập trình (Stack Overflow, Microsoft Docs) cung cấp giải pháp cho các vấn đề cụ thể.

**Hiệu suất của RSA trong C#:**Thư viện .NET được tối ưu hóa để xử lý các số lớn trong RSA, đảm bảo hiệu quả khi thực hiện các phép toán mã hóa, giải mã, và băm.

#### 3.3.1.2. Tính khả thi về mặt thực tế

**Ứng dụng đa dạng:**RSA và chữ ký số được sử dụng rộng rãi trong xác thực danh tính, bảo mật giao tiếp, và ký tài liệu.Kết quả của nghiên cứu này có thể được áp dụng ngay trong các hệ thống thực tế như:

Ký tài liệu PDF, email.

Giao dịch ngân hàng hoặc thanh toán trực tuyến.

Xác thực phần mềm hoặc hợp đồng điện tử.

**Khả năng triển khai:**Với giao diện người dùng cơ bản (Windows Forms hoặc WPF), ứng dụng chữ ký số có thể được triển khai như một công cụ độc lập hoặc tích hợp vào các hệ thống lớn hơn.

#### 3.3.1.3. Tính khả thi về mặt nghiên cứu

**Chủ đề có độ khó vừa phải:**RSA là một thuật toán mật mã kinh điển, dễ tiếp cận nhưng vẫn đủ phức tạp để làm nổi bật năng lực nghiên cứu và triển khai.

Kết hợp lý thuyết mật mã học với thực hành lập trình giúp người thực hiện cải thiện cả kiến thức và kỹ năng.

**Khả năng mở rộng:**Sau khi hoàn thành, chủ đề này có thể được mở rộng bằng các hướng tiếp cận nâng cao:Kết hợp RSA với các thuật toán khác (AES, ECC) để tăng hiệu suất.Ứng dụng trong hệ thống xác thực đa yếu tố (MFA).

Xây dựng hệ thống chữ ký số phân tán hoặc tích hợp blockchain.

#### 3.3.1.4. Rủi ro và giải pháp

**Rủi ro kỹ thuật:**Khó khăn trong việc xử lý các số lớn hoặc tối ưu hóa RSA.Giải pháp: Sử dụng thư viện mật mã học tích hợp sẵn trong .NET để giảm thiểu lỗi và tăng hiệu suất.

**Rủi ro thời gian:**RSA yêu cầu nhiều bước tính toán phức tạp, có thể mất thời gian hoàn thành.Giải pháp: Chia nhỏ công việc thành các giai đoạn cụ thể, tập trung vào các chức năng chính trước.

**Rủi ro bảo mật:**Việc lưu trữ khóa bí mật không an toàn có thể dẫn đến lộ dữ liệu.Giải pháp: Sử dụng các biện pháp bảo mật bổ sung, như mã hóa khóa bí mật khi lưu trữ (AES, DPAPI).

### *3.3.2. Những thuận lợi và khó khăn*

#### 3.3.2.1. Thuận lợi

**Thuận lợi về kỹ thuật**

**Hỗ trợ mạnh mẽ từ .NET Framework và C#:**Ngôn ngữ C# và .NET Framework cung cấp các thư viện sẵn có như System.Security.Cryptography, hỗ trợ đầy đủ cho việc triển khai RSA và các thuật toán băm như SHA-256.Các hàm xử lý số học lớn, mã hóa, và giải mã được tối ưu hóa, giảm công sức lập trình.

**Tài liệu tham khảo phong phú:**Nhiều tài liệu, ví dụ, và hướng dẫn lập trình RSA bằng C# có sẵn trên các nền tảng như Microsoft Docs, Stack Overflow, GitHub.RSA là thuật toán phổ biến, được nghiên cứu và triển khai rộng rãi, giúp việc học hỏi và áp dụng dễ dàng hơn.

**Thuận lợi về lý thuyết**

**Độ phổ biến của RSA:**RSA là một thuật toán kinh điển trong mật mã học, dễ dàng tìm hiểu thông qua sách vở, bài báo khoa học, và khóa học trực tuyến.Phương pháp sử dụng khóa công khai và khóa bí mật trong RSA rất trực quan và dễ hiểu.

**Tính ứng dụng cao:**Đề tài phù hợp với các nhu cầu bảo mật thực tế như ký tài liệu, xác thực dữ liệu, và bảo mật giao tiếp.

**Thuận lợi về thực hiện**

**Công cụ phát triển dễ sử dụng:**Visual Studio hoặc Visual Studio Code cung cấp môi trường lập trình thân thiện, tích hợp đầy đủ công cụ cần thiết để phát triển ứng dụng C#.

**Giao diện đồ họa đơn giản:**Việc xây dựng giao diện ứng dụng minh họa có thể thực hiện nhanh chóng với Windows Forms hoặc WPF.

#### 3.3.2.2. Khó khăn

**Khó khăn về lý thuyết**

**Kiến thức mật mã học phức tạp:**RSA yêu cầu hiểu biết sâu về số học mô-đun, số nguyên tố lớn, và các thuật toán băm. Việc nắm bắt các khái niệm này có thể khó khăn đối với những người mới bắt đầu.Phân biệt rõ ràng giữa các khái niệm như mã hóa, băm, và ký số có thể gây nhầm lẫn.

**Hiểu biết về bảo mật:**Để đảm bảo hệ thống chữ ký số an toàn, cần hiểu các rủi ro bảo mật như tấn công brute force, lộ khóa bí mật, và các lỗ hổng phổ biến.

**Khó khăn về kỹ thuật**

**Xử lý số lớn:**RSA hoạt động dựa trên các phép toán với số nguyên lớn, yêu cầu hiệu suất cao và tránh lỗi tràn số khi lập trình.

**Tích hợp và kiểm thử:**Việc tích hợp các bước (băm dữ liệu, ký số, xác minh chữ ký) cần được thực hiện đúng thứ tự và kiểm tra kỹ lưỡng để đảm bảo tính chính xác.Kiểm thử chữ ký số trên các dữ liệu lớn hoặc nhiều định dạng khác nhau (PDF, JSON, v.v.) có thể phức tạp.

**Khó khăn về quản lý**

**Thời gian thực hiện:**RSA và chữ ký số là chủ đề đòi hỏi nghiên cứu lý thuyết sâu và lập trình chi tiết, dễ dẫn đến việc không kịp tiến độ nếu không quản lý thời gian tốt.

**Khó khăn trong việc xử lý lỗi:**Lỗi trong tính toán mô-đun hoặc xử lý số lớn có thể khó phát hiện và sửa chữa.Các công cụ debug không trực quan khi làm việc với số lớn hoặc thuật toán mật mã.

# KẾT LUẬN

Chữ ký số RSA là một trong những giải pháp bảo mật quan trọng giúp đảm bảo tính toàn vẹn và xác thực của dữ liệu trong môi trường mạng. Qua đề tài **“Chữ Ký Số RSA và Ứng Dụng trong Xác Thực Tính Toàn Vẹn của Dữ Liệu”**, chúng em đã nghiên cứu chi tiết về cơ chế hoạt động của thuật toán RSA, từ cách tạo khóa, mã hóa, giải mã, cho đến ứng dụng thực tế trong chữ ký số.

Kết quả nghiên cứu không chỉ làm rõ lý thuyết về chữ ký số và thuật toán RSA mà còn minh họa cụ thể cách ứng dụng chúng để xác thực tính toàn vẹn của dữ liệu. Thông qua các bước thực nghiệm, chúng em đã thiết kế và triển khai thành công một chương trình sử dụng chữ ký số RSA, đồng thời kiểm chứng được khả năng bảo vệ dữ liệu khỏi các hành vi giả mạo hoặc xâm phạm.

Quá trình thực hiện đề tài đã mang lại cho chúng em nhiều kiến thức và kỹ năng quan trọng, bao gồm:

Hiểu rõ nguyên lý hoạt động và tầm quan trọng của các hệ thống mã hóa khóa công khai, đặc biệt là thuật toán RSA.

Áp dụng kiến thức về lập trình và an ninh mạng để triển khai một ứng dụng thực tế, từ đó rút ra các bài học kinh nghiệm về cách bảo vệ dữ liệu trong môi trường mạng.

Tăng cường khả năng phân tích và giải quyết vấn đề trong lĩnh vực an ninh mạng.

Mặc dù đạt được những kết quả tích cực, đề tài vẫn còn một số hạn chế, chẳng hạn như phạm vi ứng dụng chưa mở rộng sang các hệ thống phức tạp hoặc các bài toán thực tế lớn hơn. Trong tương lai, chúng em hy vọng sẽ tiếp tục phát triển và tối ưu hóa các ứng dụng liên quan đến chữ ký số, từ đó nâng cao hiệu quả trong việc đảm bảo an ninh dữ liệu.

Với sự phát triển không ngừng của công nghệ số, chữ ký số nói chung và RSA nói riêng sẽ tiếp tục đóng vai trò quan trọng trong việc bảo vệ thông tin và dữ liệu. Đề tài này không chỉ giúp chúng em hiểu rõ hơn về một công cụ bảo mật hữu ích mà còn góp phần nhỏ vào việc nâng cao nhận thức về an ninh mạng, đặc biệt trong bối cảnh các mối đe dọa mạng ngày càng gia tăng.

# TÀI LIỆU THAM KHẢO

[1]. Nguyễn Xuân Dũng, *Bảo mật thông tin – Mô Hình và ứng dụng*, NXB thống kê, 2009.

[2]. Bùi Doãn Khanh, Nguyễn Đình Thúc, *Mã hóa thông tin – Lý thuyết và ứng dụng*, NXB Lao động xã hội, 2011.

[3]. William Stallings**,** *Cryptography and Network Security Principles and Practices***,** Fourth Edition, Prentice Hall, 2005.