**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC NÔNG LÂM TPHCM**

**KHOA CÔNG NGHỆ THÔNG TIN**



**Môn: An toàn và bảo mật hệ thống thông tin**

**GVHD: Th.S Phan Đình Long**

**Đề tài: Xây dựng ứng dụng mã hóa bằng Java Swing**

**Sinh viên thực hiện:**

Cao Thành Nam - 21130448

**Mục lục**

[**Lời mở đầu** 2](#_Toc183459086)

[**LỜI CÁM ƠN** 3](#_Toc183459087)

[**NHẬN XÉT** 4](#_Toc183459088)

[**I.** **Tổng quát** 5](#_Toc183459089)

[**1.** **Giới thiệu** 5](#_Toc183459090)

[**2.** **Ngôn ngữ** 5](#_Toc183459091)

[**3.** **Cơ sở thực tiễn** 5](#_Toc183459092)

[**4.** **Cơ sở lý thuyết** 5](#_Toc183459093)

[**II.** **Màn hình** 6](#_Toc183459094)

[**1.** **Mã hóa cơ bản** 6](#_Toc183459095)

[**2.** **Mã hóa đối xứng** 6](#_Toc183459096)

[**3.** **Mã hóa đối xứng sử dụng thư viện Bouncy Castle** 6](#_Toc183459097)

[**4.** **Mã hóa bất đối xứng** 7](#_Toc183459098)

[**5.** **Hàm băm** 7](#_Toc183459099)

[**6.** **Chữ ký điện tử** 7](#_Toc183459100)

[**III.** **Chức năng** 7](#_Toc183459101)

[**1.** **Mã hóa cơ bản** 7](#_Toc183459102)

[**2.** **Mã hóa đối xứng** 8](#_Toc183459103)

[**3.** **Mã hóa đối xứng sử dụng thư viện Bouncy Castle** 10](#_Toc183459104)

[**5.** **Hàm băm** 12](#_Toc183459105)

[**6.** **Chữ ký điện tử** 13](#_Toc183459106)

[**IV.** **Tổng kết** 14](#_Toc183459107)

# **Lời mở đầu**

Tài liệu này được thiết kế nhằm cung cấp một hướng dẫn chi tiết và dễ hiểu về cách sử dụng các tính năng và công cụ của hệ thống mã hóa hiện đại. Trong bối cảnh công nghệ ngày càng phát triển, bảo mật dữ liệu trở thành một yếu tố quan trọng không thể thiếu. Qua từng bước hướng dẫn, bạn sẽ hiểu rõ cách thức bảo vệ thông tin cá nhân và giao dịch trực tuyến một cách hiệu quả và an toàn nhất. Chúng tôi hy vọng tài liệu này sẽ là nguồn tài liệu hữu ích, giúp bạn khai thác tối đa khả năng bảo mật của hệ thống.

**LỜI CÁM ƠN**

Chúng em xin chân thành cảm ơn Thầy đã tận tình hướng dẫn và truyền đạt những kiến thức quý báu trong suốt khóa học. Nhờ sự chỉ dẫn tận tâm của Thầy, chúng em đã có cơ hội tiếp cận với những khái niệm và kỹ thuật mới mẻ, cũng như thực hành và phát triển các dự án thực tế. Khóa học này đã trang bị cho chúng em không chỉ kiến thức chuyên môn mà còn cả những kỹ năng quan trọng trong công việc và cuộc sống. Những bài học và kinh nghiệm quý báu mà chúng em đã nhận được sẽ là hành trang vững chắc trên con đường phát triển sự nghiệp trong lĩnh vực công nghệ thông tin. Song cũng không tránh khỏi những thiếu sót, mong thầy sẽ tận tình chỉ bảo thêm cho chúng em

**NHẬN XÉT**

Ngày 24 tháng 11 năm 2024

(**Ký tên**)

1. **Tổng quát**
2. **Giới thiệu**

* Đây là ứng dụng được viết bằng Java sử dụng Java Swing.
* Ứng dụng cho phép người dùng thực hiện các thuật toán mã hóa, hàm băm và chữ ký điện tử.
* Ứng dụng có đầy đủ các tính năng như tạo khóa, lưu khóa, tải khóa, mã hóa và giải mã dạng chữ, mã hóa và giải mã dạng file có thể đáp ứng đầy đủ các nhu cầu của người dùng.

1. **Ngôn ngữ**

* **Java**: 17.0.12 2024-07-16 LTS

1. **Cơ sở thực tiễn**

* Sự phát triển không gian số nhanh chóng làm cho thông tin cá nhân của mỗi người có nguy cơ bị rò rì ra bên ngoài. Ngoài ra các thông tin giả mảo tràn lan trên không gian mạng là một vấn đề nan giải, ngày nay khi tỷ lệ tội phạm không gian mạng và nạn nhân bị lừa mất tiền ngày càng nhiều chúng tôi cần áp dụng các lý thuyết về mã hóa đã học để áp dụng vào thực tiễn giúp các nguy cơ lừa đảo giảm đi.
* Chúng tôi đã đưa ra giải pháp xây dựng ứng dụng mã hóa nhằm giúp người dùng hiểu cách hoạt động của các thuật toán mã hóa và vai trò của nó trong thế giới ngày nay.

1. **Cơ sở lý thuyết**

* **Java** là một ngôn ngữ lập trình hướng đối tượng, được phát triển bởi Sun Microsystems và phát hành lần đầu vào năm 1995. Nó được thiết kế để có thể chạy trên nhiều nền tảng khác nhau mà không cần phải thay đổi mã nguồn (nguyên lý "write once, run anywhere"). Java nổi tiếng với sự an toàn, ổn định và khả năng quản lý bộ nhớ tự động (garbage collection). Java được sử dụng rộng rãi trong phát triển các ứng dụng web,
* mobile (Android), ứng dụng máy tính để bàn, và các hệ thống nhúng.
* **Java Swing** là một phần của bộ công cụ GUI (Graphical User Interface) thuộc Java Foundation Classes (JFC), được sử dụng để xây dựng các ứng dụng máy tính để bàn có giao diện đồ họa. Được phát triển như một sự thay thế cho AWT (Abstract Window Toolkit), Swing cung cấp nhiều tính năng nâng cao và các thành phần giao diện tùy chỉnh.

1. **Màn hình**
2. **Mã hóa cơ bản**

* Đây là màn hình dùng để thực hiện các thuật toán mã hóa cơ bản bao gồm: Affine, Caesar và Vigenere.
* Có các thành phần gồm: Nhập input, xuất ra output, nút mã hóa, nút giải mã, nút tạo khóa và các lựa chọn để lựa chọn thuật toán.

1. **Mã hóa đối xứng**

* Đây là màn hình dùng để thực hiện các thuật toán mã hóa đối xứng mà Java hỗ trợ bao gồm: AES, DES, 3-DES, Blowfish, RC2 và RC4.
* Các mode gồm: ECB, CBC, PCBC, OFC, CFB và CTR
* Các padding gồm: NoPadding, PKCS1Padding, PKCS5Padding, PKCS7Padding, SSL3Padding.
* Có các thành phần gồm: Nhập input, xuất ra output, nút mã hóa và giải mã dạng chữ, nút mã hóa và giải mã dạng file, nút tạo khóa, nút tải khóa, các lựa chọn để lựa chọn thuật toán, mode và padding.

1. **Mã hóa đối xứng sử dụng thư viện Bouncy Castle**

* Đây là màn hình dùng để thực hiện các thuật toán mã hóa đối xứng sử dụng thư viện bên ngoài bao gồm: ChaCha20, Salsa20, HC256 và HC128.
* Có các thành phần gồm: Nhập input, xuất ra output, nút mã hóa và giải mã dạng chữ, nút mã hóa và giải mã dạng
* file, nút tạo khóa, nút tải khóa, các lựa chọn để lựa chọn thuật toán.

1. **Mã hóa bất đối xứng**

* Đây là màn hình dùng để thực hiện thuật toán mã hóa bất đối xứng RSA.
* Có các thành phần gồm: Nhập input, xuất ra output, nút mã hóa và giải mã dạng chữ, nút tạo khóa, nút tải khóa.

1. **Hàm băm**

* Đây là màn hình dùng để thực hiện hàm băm bao gồm: MD5, SHA1, SHA256, SHA512, SHA3-256 và SHA3-512.
* Có các thành phần gồm: Nhập input, xuất ra output, nút băm chữ và băm file và nút tạo khóa.

1. **Chữ ký điện tử**

* Đây là màn hình dùng để thực hiện hàm băm bao gồm: MD5, SHA1, SHA256, SHA512, SHA3-256 và SHA3-512.
* Có các thành phần gồm: Nhập input, xuất ra output, nút băm chữ và băm file và nút tạo khóa.

1. **Chức năng**
2. **Mã hóa cơ bản**

* **Tạo khóa**
  + Đối với mã hóa Affine:
    - B1: Ấn nút “Tạo khóa”
    - B2: Nhập kích thước của khóa gồm 2 số nguyên dương a và b sao cho a và b là hai số nguyên tố cùng nhau nghĩa là ước chung lớn nhất bằng 1
  + Đối với mã hóa Caesar:
    - B1: Ấn nút “Tạo khóa”
    - B2: Nhập kích thước của khóa gồm 1 số nguyên dương a
  + Đối với mã hóa Vigenere:
    - B1: Ấn nút “Tạo khóa”
    - B2: Nhập khóa bất kỳ ở dạng chữ
* **Mã hóa:**
  + B1: Chọn thuật toán mã hóa
  + B2: Ấn nút “Tạo khóa”, đối với mỗi thuật toán khác nhau sẽ yêu cầu người dùng nhập các tham số khác nhau
  + B3: Nhập nội dung cần mã hóa vào ô “Mã hóa”
  + B4: Ấn nút “Mã hóa”
* **Giải mã:**
  + B1: Nhập nội dung vừa mã hóa ở trên vào ô “Mã hóa”
  + B2: Ấn nút “Giải mã”

1. **Mã hóa đối xứng**

* **Kích thước khóa:**
  + **AES:**
    - Key(bit): 128, 192, 256
    - IV(bit): 16
    - Hỗ trợ các mode: ECB, CBC, PCBC, OFB, CFB, CTR
    - Hỗ trợ các padding: PKCS5Padding, NoPadding(khi dùng CTR)
  + **DES**
    - Key(bit): 56
    - IV(bit): 8
    - Hỗ trợ các mode: ECB, CBC, PCBC, OFB, CFB, CTR
    - Hỗ trợ các padding: PKCS5Padding, NoPadding(khi dùng CTR)
  + **3-DES**
    - Key(bit): 112, 168
    - IV(bit): 8
    - Hỗ trợ các mode: ECB, CBC, PCBC, OFB, CFB, CTR
    - Hỗ trợ các padding: PKCS5Padding, NoPadding(khi dùng CTR)
  + **Blowfish**
    - Key(bit): từ 32 đến 448 và là số chia hết cho 8
    - IV(bit): 8
    - Hỗ trợ các mode: ECB, CBC, PCBC, OFB, CFB, CTR
    - Hỗ trợ các padding: PKCS5Padding, NoPadding(khi dùng CTR)
  + **RC2**
    - Key(bit): 40, 1024
    - IV(bit): 8
    - Hỗ trợ các mode: ECB, CBC, PCBC, OFB, CFB, CTR
    - Hỗ trợ các padding: NoPadding, PKCS5Padding, nếu không dùng mode thì hỗ trợ tất cả padding, nếu sử dụng mode CTR thì chỉ hỗ trợ NoPadding
  + **RC4**
    - Key(bit): 40, 1024
    - Hỗ trợ các mode: ECB
    - Hỗ trợ các padding: Nếu không dùng mode thì hỗ trợ tất cả padding, nếu dùng mode ECB thì hỗ trợ NoPadding
* **Tạo khóa:**
  + B1: Ấn nút “Tạo khóa”
  + B2: Chọn thư mục, nhập tên file
  + B3: Nhập kích thước key. Nếu không phải mode ECB thì cần nhập thêm kích thước IV
* **Tải khóa:**
  + B1: Ấn nút “Tải khóa”
  + B2: Chọn file đã lưu khóa. Nếu không phải mode ECB thì cần chọn thêm file lưu IV
* **Mã hóa chữ:**
  + B1: Chọn thuật toán, mode và padding mã hóa.
  + B2: Nếu đã có khóa thì ấn nút “Tải khóa”. Nếu chưa có khóa thì ấn “Tạo khóa”
  + B3: Nhập nội dung cần mã hóa vào ô “Mã hóa”
  + B4: Ấn nút “Mã hóa”
* **Giải mã chữ:**
  + B1: Chọn thuật toán, mode và padding mã hóa.
  + B2: Nếu đã có khóa thì ấn nút “Tải khóa”. Nếu chưa có khóa thì ấn “Tạo khóa”
  + B3: Nhập nội dung cần mã hóa vào ô “Mã hóa”
  + B4: Ấn nút “Giải mã”
* **Mã hóa file:**
  + B1: Chọn thuật toán, mode và padding mã hóa.
  + B2: Nếu đã có khóa thì ấn nút “Tải khóa”. Nếu chưa có khóa thì ấn “Tạo khóa”
  + B3: Kéo file cần mã hóa và thả vào ô “Mã hóa”
  + B4: Ấn nút “Mã hóa file”
* **Giải mã file:**
  + B1: Chọn thuật toán, mode và padding mã hóa.
  + B2: Nếu đã có khóa thì ấn nút “Tải khóa”. Nếu chưa có khóa thì ấn “Tạo khóa”
  + B3: Kéo file cần giải mã và thả vào ô “Mã hóa”
  + B4: Ấn nút “Giải mã file”

1. **Mã hóa đối xứng sử dụng thư viện Bouncy Castle**

* **Kích thước khóa:**
  + **ChaCha20**
    - Key(byte): 32, 16
    - IV(byte): 8
  + **Salsa20**
    - Key(byte): 32, 16
    - IV(byte): 8
  + **HC256**
    - Key(byte): 32, 16
    - IV(byte): 16
  + **HC128**
    - Key(byte): 32, 16
    - IV(byte): 16
* **Tạo khóa:**
  + B1: Ấn nút “Tạo khóa”
  + B2: Chọn thư mục, nhập tên file
  + B3: Nhập kích thước key
* **Tải khóa:**
  + B1: Ấn nút “Tải khóa”
  + B2: Chọn file đã lưu khóa
* **Mã hóa chữ:**
  + B1: Chọn thuật toán
  + B2: Nếu đã có khóa thì ấn nút “Tải khóa”. Nếu chưa có khóa thì ấn “Tạo khóa”
  + B3: Nhập nội dung cần mã hóa vào ô “Mã hóa”
  + B4: Ấn nút “Mã hóa”
* **Giải mã chữ:**
  + B1: Chọn thuật toán
  + B2: Nếu đã có khóa thì ấn nút “Tải khóa”. Nếu chưa có khóa thì ấn “Tạo khóa”
  + B3: Nhập nội dung cần mã hóa vào ô “Mã hóa”
  + B4: Ấn nút “Giải mã”
* **Mã hóa file:**
  + B1: Chọn thuật toán
  + B2: Nếu đã có khóa thì ấn nút “Tải khóa”. Nếu chưa có khóa thì ấn “Tạo khóa”
  + B3: Kéo file cần mã hóa và thả vào ô “Mã hóa”
  + B4: Ấn nút “Mã hóa file”
* **Giải mã file:**
  + B1: Chọn thuật toán
  + B2: Nếu đã có khóa thì ấn nút “Tải khóa”. Nếu chưa có khóa thì ấn “Tạo khóa”
  + B3: Kéo file cần giải mã và thả vào ô “Mã hóa”
  + B4: Ấn nút “Giải mã file”

1. **Mã hóa bất đối xứng RSA**

* **Kích thước:**
  + Key(bit): 1024, 2048, 3072, 4096
  + Tương ứng với mỗi kích thước khóa mà ta có thể mã hóa chiều dài văn bản khác nhau. Nếu kích thước khóa là:
    - 1024 bit: kích thước byte tối đa là 117 byte
    - 2048 bit: kích thước byte tối đa là 245 byte
    - 3072 bit: kích thước byte tối đa là 373 byte
    - 4096 bit: kích thước byte tối đa là 501 byte
* **Tạo khóa:**
  + B1: Ấn nút “Tạo khóa”
  + B2: Nhập kích thước key
  + B3: Ta cần lưu hai khóa public key và private key. Chọn thư mục, nhập tên file
* **Tải khóa:**
  + B1: Ấn nút “Tải khóa”
  + B2: Chọn file đã lưu public key và private key.
* **Mã hóa chữ:**
  + B1: Nếu đã có khóa thì ấn nút “Tải khóa”. Nếu chưa có khóa thì ấn “Tạo khóa”.
  + B2: Nhập nội dung cần mã hóa vào ô “Mã hóa”
  + B3: Ấn nút “Mã hóa”.
* **Giải mã chữ:**
  + B1: Nếu đã có khóa thì ấn nút “Tải khóa”. Nếu chưa có khóa thì ấn “Tạo khóa”
  + B2: Nhập nội dung cần mã hóa vào ô “Mã hóa”
  + B3: Ấn nút “Giải mã”

1. **Hàm băm**

Hàm băm (hash function) là một thuật toán dùng để chuyển đổi dữ liệu đầu vào (thường là chuỗi ký tự hoặc số) thành một chuỗi cố định có độ dài nhất định, gọi là giá trị băm (hash value). Dữ liệu đầu vào có thể có độ dài bất kỳ, nhưng kết quả của hàm băm luôn có độ dài cố định, phụ thuộc vào loại hàm băm. Tác dụng của hàm băm gồm:

* Mã hóa mật khẩu: Lưu trữ mật khẩu dưới dạng băm để tăng tính bảo mật.
* Chữ ký điện tử: Tạo một giá trị băm từ dữ liệu cần ký, sau đó mã hóa giá trị băm này để tạo chữ ký.
* Kiểm tra tính toàn vẹn dữ liệu: Đảm bảo dữ liệu không bị thay đổi nhờ so sánh giá trị băm.
* **Thuật toán:**
  + MD5
  + SHA1
  + SHA256
  + SHA512
  + SHA3-256
  + SHA3-512
* **Tạo khóa:**
  + B1: Chọn thuật toán
  + B2: Ấn nút “Tạo khóa”
* **Băm chữ:**
  + B1: Nhập nội dung vào ô “Mã hóa”
  + B2: Ấn nút “Băm chữ”
* **Băm file:**
  + B1: Kéo và thả file vào ô “Mã hóa”
  + B2: Ấn nút “Băm file”

1. **Chữ ký điện tử**

Chữ ký điện tử (digital signature) là một kỹ thuật mật mã cho phép xác thực tính hợp lệ và tính toàn vẹn của một tài liệu, tin nhắn, hoặc dữ liệu số. Chữ ký điện tử giúp đảm bảo rằng:

* Xác thực (Authentication): Người nhận có thể xác minh người gửi thực sự là ai. Chữ ký điện tử xác thực danh tính của người gửi, đảm bảo rằng dữ liệu đến từ đúng nguồn.
* Tính toàn vẹn (Integrity): Chữ ký điện tử bảo đảm dữ liệu không bị thay đổi trong quá trình truyền tải. Nếu dữ liệu bị thay đổi, chữ ký sẽ không khớp với dữ liệu gốc.
* Không thể phủ nhận (Non-repudiation): Người gửi không thể phủ nhận rằng họ đã ký và gửi tài liệu hoặc tin nhắn sau khi đã thực hiện chữ ký điện tử.
* **Kích thước:**
  + key(bit): 512, 1024, 2048, 3072
* **Tạo khóa:**
  + B1: Ấn nút “Tạo khóa”
  + B2: Nếu đã có khóa thì ấn nút “Tải khóa”. Nếu chưa có khóa thì ấn nút “Tạo khóa”
  + B3: Nhập kích thước khóa
  + B4: Ta cần lưu hai khóa public key và private key. Chọn thư mục, nhập tên file
* **Tải khóa:**
  + B1: Ấn nút “Tải khóa”
  + B2: Chọn file đã lưu public key và private key.
* **Ký chữ:**
  + B1: Nhập nội dung cần ký vào ô “Mã hóa”
  + B2: Ấn nút “Ký chữ”
* **Xác minh chữ ký chữ:**
  + B1: Nhập chữ ký cần xác minh vào ô “Mã hóa”
  + B2: Ấn nút “Xác minh chữ ký chữ”
* **Ký file:**
  + B1: Kéo file và thả vào ô “Mã hóa”
  + B2: Ấn nút “Ký file”
* **Xác minh chữ ký file:**
  + B1: Nhập chữ ký cần xác minh vào ô “Mã hóa”
  + B2: Ấn nút “Xác minh chữ ký file”

1. **Tổng kết**

Trong ứng dụng Java Swing xây dựng các thuật toán mã hóa, chúng tôi đã triển khai và tích hợp các thuật toán mã hóa phổ biến để minh họa cách thức chúng hoạt động trong thực tế. Ứng dụng này cho phép người dùng trải nghiệm trực quan quy trình mã hóa và giải mã dữ liệu, từ đó hiểu rõ hơn về các nguyên lý bảo mật thông tin.

* **Các tính năng chính của ứng dụng bao gồm:**

Mã hóa cơ bản: Affine, Caesar, Vigenere thể hiện tính đơn giản và cơ bản nhất để bắt đầu tìm hiểu sâu hơn về mã hóa đối xứng và bất đối xứng.

Mã hóa đối xứng: Sử dụng các thuật toán như AES, DES, 3-Des, Blowfish, RC2 và RC4 kết hợp với các mode và padding để mã hóa và giải mã dữ liệu nhanh chóng, phù hợp cho việc bảo mật dữ liệu trong các hệ thống yêu cầu tốc độ cao.

Mã hóa đối xứng sử dụng thư viên: như ChaCha20, Salsa20, HC256 và HC128 thực hiện các kỹ thuật sử dụng thư viện bên ngoài để có thêm khả năng thích ứng tốt đối với hệ thống yêu cầu bảo mật cao hơn.

Mã hóa bất đối xứng: Tích hợp thuật toán RSA để thực hiện mã hóa và giải mã dựa trên cặp khóa công khai và khóa riêng. Đây là phương thức mã hóa thường được dùng trong các hệ thống yêu cầu bảo mật cao và không cần chia sẻ khóa trước.

Chữ ký số: Sử dụng DSA cho phép tạo và xác minh chữ ký điện tử để đảm bảo tính toàn vẹn và xác thực của dữ liệu.

Hàm băm: Sử dụng các thuật toán băm như MD5, SHA1, SHA256, SHA512, SHA3-256, SHA3-512 để kiểm tra tính toàn vẹn của dữ liệu, giúp phát hiện bất kỳ sự thay đổi nào trong quá trình truyền tải.

* **Kết quả đạt được:**
  + Người dùng có thể dễ dàng thao tác với giao diện đồ họa thân thiện để mã hóa và giải mã dữ liệu theo nhu cầu.
  + Ứng dụng cung cấp môi trường thử nghiệm trực quan để hiểu rõ cách thức từng thuật toán hoạt động.
  + Việc xây dựng ứng dụng giúp củng cố kiến thức về bảo mật, đặc biệt là các kỹ thuật mã hóa đối xứng, bất đối xứng và băm.
* **Hướng phát triển:**

Trong tương lai, ứng dụng có thể được nâng cấp với các tính năng như:

* Tích hợp thêm nhiều thuật toán mã hóa mới như ECC, Hill,… nhằm mở rộng khả năng lựa chọn cho người dùng.
* Tăng cường tính năng quản lý khóa và kết hợp với cơ chế lưu trữ khóa an toàn.
* Mở rộng hỗ trợ cho các giao thức bảo mật hiện đại, chẳng hạn như SSL/TLS, để minh họa cách các thuật toán mã hóa được sử dụng trong truyền thông bảo mật trên mạng Internet.
* Với ứng dụng này, người dùng không chỉ có công cụ để mã hóa dữ liệu mà còn có cái nhìn toàn diện hơn về cách thức hoạt động của các thuật toán mã hóa trong thế giới thực.