

**HỌC VIỆN NGÂN HÀNG**

**KHOA HỆ THỐNG THÔNG TIN QUẢN LÝ**

---🙞🙜🕮🙞🙜---



**BÁO CÁO BÀI TẬP LỚN**

***Học phần: Công nghệ ngân hàng***

*Đề tài*:

**TÌM HIỂU VỀ CHUẨN CHỮ KÝ SỐ RSA (TCVN 7635:2007 VỀ KỸ THUẬT MẬT MÃ - CHỮ KÝ SỐ) SỬ DỤNG TRONG GIAO DỊCH ĐIỆN TỬ VÀ TRIỂN KHAI CÀI ĐẶT THỬ NGHIỆM**

Giảng viên hướng dẫn : ThS. Vũ Duy Hiến

Mã lớp học phần : 221IS36A01

Nhóm sinh viên thực hiện : Nhóm 1

Trần Thị Hải Yến – 22A4040002

Bùi Thị Bích Hằng – 22A4040026

Nguyễn Thị Minh Nguyệt – 22A4040031

Mai Thị Huyền – 22A4040008

Hà Nội – 2022

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

THÔNG TIN CHUNG

**Tên đề tài:** Tìm hiểu về chuẩn chữ ký số RSA (TCVN 7635:2007 về Kỹ thuật mật mã - Chữ ký số) sử dụng trong giao dịch điện tử và triển khai cài đặt thử nghiệm

**Danh sách thành viên nhóm 1:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S**  **T**  **T** | **Thành viên** | **Mã sinh viên** | **Nội dung  công việc** | **Đóng góp (%)** |
| 1 | Trần Thị Hải Yến | 22A4040002 | 2.2, 2.3, Kết luận  Tổng hợp, chỉnh sửa báo cáo. | 25 % |
| 2 | Nguyễn Thị Minh Nguyệt | 22A4040031 | Chương 3 | 25 % |
| 3 | Bùi Thị Bích Hằng | 22A4040026 | 1.2  Tạo bài trình bày. | 25 % |
| 4 | Mai Thị Huyền | 22A4040008 | Mở đầu, 1.1, 2.1  Tạo bài trình bày. | 25 % |

**Giảng viên hướng dẫn:** ThS. Vũ Duy Hiến.

LỜI CAM ĐOAN

Nhóm 01 xin cam đoan kết quả đạt được trong báo cáo là sản phẩm nghiên cứu, tìm hiểu của riêng của nhóm. Trong toàn bộ nội dung của báo cáo, những điều được trình bày hoặc là của nhóm hoặc là được tổng hợp từ nhiều nguồn tài liệu. Tất cả các tài liệu tham khảo đều có xuất xứ rõ ràng và được trích dẫn hợp pháp.

Chúng em xin hoàn chịu trách nhiệm và chịu mọi hình thức kỷ luật theo quy định cho lời cam đoan của mình.

NHÓM SINH VIÊN THỰC HIỆN

LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành tới thầy Vũ Duy Hiến - Giảng viên Khoa Hệ thống thông tin quản lý, Học viện Ngân hàng. Trong quá trình học tập và thực hiện bài báo cáo “Tìm hiểu về chuẩn chữ ký số RSA (TCVN 7635:2007 về Kỹ thuật mật mã - Chữ ký số) sử dụng trong giao dịch điện tử và triển khai cài đặt thử nghiệm”, thầy đã luôn tạo điều kiện, giúp đỡ để chúng em hoàn thành được bài báo cáo này.

Nhóm đã cố gắng hoàn thiện bài báo cáo với tất cả sự nỗ lực và cố gắng của cả nhóm. Tuy nhiên, do còn thiếu nhiều kinh nghiệm, chắc chắn bài báo cáo sẽ không tránh khỏi thiếu sót. Vì vậy, chúng em rất mong nhận được sự quan tâm, những ý kiến đóng góp của thầy để bài báo cáo của chúng em có thể hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

MỤC LỤC

[NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN i](#_Toc123074244)

[THÔNG TIN CHUNG ii](#_Toc123074245)

[LỜI CAM ĐOAN iii](#_Toc123074246)

[LỜI CẢM ƠN iv](#_Toc123074247)

[MỤC LỤC v](#_Toc123074248)

[DANH MỤC HÌNH ẢNH vii](#_Toc123074249)

[DANH MỤC BẢNG BIỂU viii](#_Toc123074250)

[LỜI MỞ ĐẦU 1](#_Toc123074251)

[CHƯƠNG 1. TỔNG QUAN VỀ CHỮ KÝ SỐ RSA 2](#_Toc123074252)

[1.1. Khái niệm 2](#_Toc123074253)

[1.2. Ứng dụng của chữ ký số tại Việt Nam 5](#_Toc123074254)

[1.2.1. Ứng dụng chữ ký số chuyên dùng trong cơ quan Nhà nước 6](#_Toc123074255)

[1.2.2. Ứng dụng chữ ký số trong cơ quan tổ chức 9](#_Toc123074256)

[1.2.3. Ứng dụng chữ ký số trong các lĩnh vực 10](#_Toc123074257)

[CHƯƠNG 2. TIÊU CHUẨN CHỮ KÝ SỐ RSA 14](#_Toc123074258)

[2.1. Các quy định về chữ ký số RSA 14](#_Toc123074259)

[2.1.1. Giải thích từ ngữ 14](#_Toc123074260)

[2.1.2. Các yêu cầu chung về cặp khóa RSA 16](#_Toc123074261)

[2.1.3. Quy định chi tiết về các khóa RSA 17](#_Toc123074262)

[2.2. Giới thiệu về Tiêu chuẩn Quốc gia Việt Nam về chữ ký số (TCVN 7635:2007 Các kỹ thuật mật mã - Chữ ký số) 17](#_Toc123074263)

[2.2.1. Thuật toán chữ ký số RSA-PSS 19](#_Toc123074264)

[2.2.2. Hàm băm SHA-256 27](#_Toc123074265)

[2.2.3. Bộ tạo số giả ngẫu nhiên dùng AES-128 29](#_Toc123074266)

[2.2.4. Lý do cho việc sử dụng 3 thuật toán trong tiêu chuẩn 30](#_Toc123074267)

[2.3. So sánh chữ ký số RSA trong lý thuyết và tiêu chuẩn 31](#_Toc123074268)

[CHƯƠNG 3. TRIỂN KHAI CÀI ĐẶT THỬ NGHIỆM 33](#_Toc123074269)

[3.1. Thiết lập môi trường và kịch bản thử nghiệm 33](#_Toc123074270)

[3.1.1. Thiết lập môi trường 33](#_Toc123074271)

[3.1.2. Kịch bản thử nghiệm 34](#_Toc123074272)

[3.2. Cài đặt thử nghiệm 35](#_Toc123074273)

[3.2.1. Mô tả các tệp 35](#_Toc123074274)

[3.2.2. Các hàm chính được sử dụng 35](#_Toc123074275)

[3.2.3. Xây dựng giao diện chương trình 36](#_Toc123074276)

[3.2.4. Khởi tạo khóa 38](#_Toc123074277)

[3.2.5. Tạo chữ ký 40](#_Toc123074278)

[3.2.6. Xác thực chữ ký 42](#_Toc123074279)

[3.2.7. Giao diện chương trình 42](#_Toc123074280)

[3.3. Kết quả thử nghiệm 46](#_Toc123074281)

[KẾT LUẬN 47](#_Toc123074282)

[TÀI LIỆU THAM KHẢO 48](#_Toc123074283)

DANH MỤC HÌNH ẢNH

[Hình 1.1 Quy trình sinh khóa 4](#_Toc123056201)

[Hình 1.2 Quy trình tạo chữ ký số RSA 4](#_Toc123056202)

[Hình 1.3 Quy trình thẩm định chữ ký số RSA 5](#_Toc123056203)

[Hình 1.4 Số lượng chứng thư số đang hoạt động theo từng CA công cộng tính đến 31/12/2022 Nguồn: (Trung tâm Chứng thực điện tử quốc gia - Bộ Thông tin và Truyền thông, 2021) 6](#_Toc123056204)

[Hình 2.1 Mô tả quá trình tạo và kiểm tra chữ ký số 18](#_Toc123056205)

[Hình 2.2 Mã hóa EMSA-PSS 21](#_Toc123056206)

[Hình 2.3 Kiểm tra EMSA-PSS 23](#_Toc123056207)

[Hình 3.1 Số lượng tải xuống hàng ngày của gói RSA – Tổng thể 33](#_Toc123056208)

[Hình 3.2 Code giao diện form tạo khóa 36](#_Toc123056209)

[Hình 3.3 Code giao diện form tạo chữ ký số 37](#_Toc123056210)

[Hình 3.4 Code giao diện form xác minh chữ ký số 38](#_Toc123056211)

[Hình 3.5 Code khởi tạo khóa công khai và khóa bí mật 38](#_Toc123056212)

[Hình 3.6 Code lưu khóa được tạo 39](#_Toc123056213)

[Hình 3.7 Lưu khóa công khai với định dạng .pem 39](#_Toc123056214)

[Hình 3.8 Lưu khóa bí mật với định dạng .pem 40](#_Toc123056215)

[Hình 3.9 Code băm thông điệp đầu vào 40](#_Toc123056216)

[Hình 3.10 Code tạo chữ ký số 41](#_Toc123056217)

[Hình 3.11 Code xác thực chữ ký số 42](#_Toc123056218)

[Hình 3.12 Giao diện tạo khóa 43](#_Toc123056219)

[Hình 3.13 Giao diện tạo chữ ký số 44](#_Toc123056220)

[Hình 3.14 Giao diện xác minh chữ ký số khi chữ ký hợp lệ 45](#_Toc123056221)

[Hình 3.15 Giao diện xác minh chữ ký số khi chữ ký không hợp lệ 45](#_Toc123056222)

DANH MỤC BẢNG BIỂU

[Bảng 1.1 Danh sách 5 Bộ, cơ quan ngang Bộ, cơ quan thuộc Chính phủ có số lượng chứng thư số đang hoạt động nhiều nhất năm 2020 (Nguồn: Ban Cơ yếu Chính phủ) 7](#_Toc123056223)

[Bảng 1.2 Đối tượng, phạm vi sử dụng chữ ký số trong các Bộ, cơ quan ngang Bộ, cơ quan thuộc Chính Phủ tính đến 31/12/2020 (Nguồn NEAC) 8](#_Toc123056224)

[Bảng 2.1 Các ký hiệu 15](#_Toc123056225)

[Bảng 2.2 Quy định kỹ thuật 16](#_Toc123056226)

[Bảng 2.3 So sánh chữ ký số RSA trong lý thuyết và tiêu chuẩn 31](#_Toc123056227)

[Bảng 3.1 Dữ liệu thử nghiệm 35](#_Toc123056228)

[Bảng 3.2 Mô tả các tập tin để xây dựng chương trình 35](#_Toc123056229)

LỜI MỞ ĐẦU

Ngày nay trong mọi hoạt động của con người thông tin đóng một vai trò quan trọng không thể thiếu. Xã hội càng phát triển nhu cầu trao đổi thông tin giữa các thành phần trong xã hội ngày càng lớn. Mạng máy tính ra đời đã mang lại cho con người rất nhiều lợi ích trong việc trao đổi và xử lý thông tin một cách nhanh chóng và chính xác. Chính từ những thuận lợi này đã đặt ra cho chúng ta một câu hỏi, liệu thông tin đi từ nơi gửi đến nơi nhận có đảm bảo tuyệt đối an toàn, ai có thể đảm bảm thông tin của ta không bị truy cập bất hợp pháp. Thông tin được lưu giữ, truyền dẫn, cùng sử dụng trên mạng lưới thông tin công cộng có thể bị nghe trộm, chiếm đoạt, xuyên tạc hoặc phá huỷ dẫn đến sự tổn thất không thể lường được. Đặc biệt là đối với những số liệu của hệ thống ngân hàng, hệ thống thương mại, cơ quan quản lý của chính phủ hoặc thuộc lĩnh vực quân sự được lưu giữ và truyền dẫn trên mạng.

Các phương thức tấn công thông qua mạng ngày càng tinh vi, phức tạp có thể dẫn đến mất mát thông tin, thậm chí có thể làm sụp đổ hoàn toàn hệ thống thông tin của tổ chức. Để giải quyết vấn đề bảo mật thông tin, một trong những ứng dụng quan trọng đã được ra đời, đó là chữ ký số. Với đặc điểm là đơn giản cho người sử dụng mà vẫn đảm bảo được tính bảo mật, kỹ thuật sử dụng chữ ký số là một trong những kỹ thuật được sử dụng phổ biến, đa dạng trong hầu hết các lĩnh vực, nhất là Tài chính, Ngân hàng. Vì lý do đó, nhóm đã quyết định nghiên cứu về “Chữ ký số RSA” và chuẩn chữ ký số RSA (TCVN 7635:2007 về Kỹ thuật mật mã - Chữ ký số).

# TỔNG QUAN VỀ CHỮ KÝ SỐ RSA

## Khái niệm

Theo Khoản 6, Điều 3, Nghị định 130/2018/NĐ-CP quy định về Luật giao dịch điện tử về chữ ký số và Dịch vụ chứng thực chữ ký số:

“Chữ ký số là một dạng chữ ký điện tử được tạo ra bằng sự biến đổi một thông điệp dữ liệu sử dụng hệ thống mật mã không đối xứng, theo đó, người có được thông điệp dữ liệu ban đầu và khóa công khai của người ký có thể xác định được chính xác.” (Cục Quản lý mật mã dân sự và Kiểm định sản phẩm mật mã, 2016)

Vậy có thể hiểu, chữ ký số RSA là loại chữ ký số sử dụng hệ mã hóa RSA để tăng độ an toàn và truyền đạt dữ liệu dạng số hóa bảo mật hơn.

* Hệ mã hóa RSA:

Trong mật mã học, RSA là một thuật toán mật mã hóa khóa công khai. Đây là thuật toán đầu tiên phù hợp với việc tạo ra chữ ký điện tử đồng thời với việc mã hóa. Nó đánh dấu một sự tiến bộ vượt bậc của lĩnh vực mật mã học trong việc sử dụng khóa công khai (public key).

* Hàm băm (Hash Funtion):

Là hàm toán học chuyẻn đổi thông điệp (message) có độ dài bất kỳ (hữu hạn) thành một dãy bit có độ dài cố định (tùy thuộc vào thuật toán băm). Dãy bit này được gọi là thông điệp rút gọn (message disgest) hay giá trị băm (hash value), đại diện cho thông điệp ban đầu. Hàm băm có nhiều loại khác nhau, như: hàm băm SHA-1, SHA-256, .... Trong bài nghiên cứu này, nhóm sử dụng hàm băm SHA-256.

SHA-256 được sử dụng để tính giá trị băm của một thông điệp có độ dài là , với . Thuật toán sử dụng một chuỗi 64 từ (32 bit) được tạo ra từ một khối thông điệp đầu vào, 8 biến làm việc cho mỗi từ 32 bit, giá trị băm trung gian gồm 8 từ (32 bit), kết quả cuối cùng của SHA-256 là 256 bit mã băm hay còn gọi là bản tóm lược thông điệp.

* Lược đồ chữ ký số RSA: độ an toàn của lược đồ chữ ký RSA dựa vào độ an toàn của hệ mã RSA. Lược đồ bao gồm cả chữ ký số kèm theo bản rõ và tự khôi phục thông điệp từ chữ ký số.
* Thuật toán sinh khóa cho lược đồ chữ ký RSA
* Thuật toán sinh chữ ký RSA
* Thuật toán chứng thực chữ ký RSA

Dựa trên công nghệ mã hóa trên, Chữ ký số RSA gồm một cặp khóa được mã hóa không đối xứng như một ***khóa công khai (Public Key)*** và ***khóa bí mật (Private Key)****.* Trong đó:

* Khóa bí mật được dùng để tạo ra chữ ký số RSA
* Khóa công khai được dụng trong việc kiểm tra chữ ký số và xác thực người dùng. Nó được tạo ra bởi khóa bí mật tương ứng trong mỗi cặp khóa.

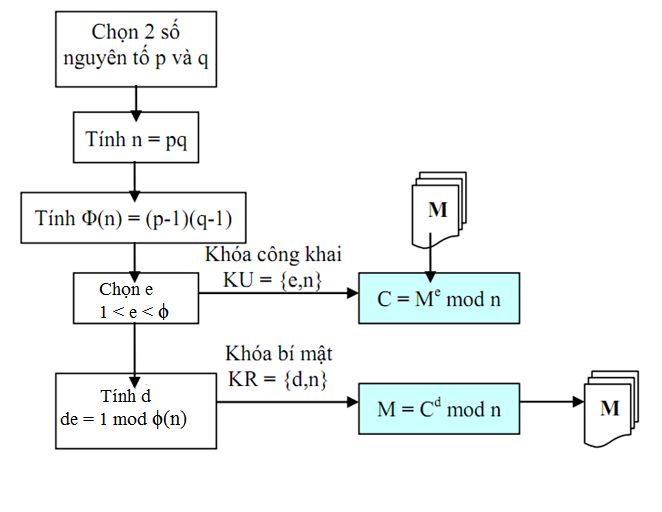
*Ví dụ minh họa sử dụng cặp khóa :*

Pepper muốn gửi thư mật cho Lisa và mong muốn chỉ có Lisa đọc được bức thư đó. Để bức thư được an toàn, trước đó Lisa đã gửi cho Pepper một cái hộp rỗng có khóa đã mở sẵn (Khóa công khai) và Lisa đã giữ chìa khóa. Pepper nhận được cái hộp liền bỏ thư mật vào trong và đóng khóa lại( khóa không cần chìa để đóng). Tiếp theo, Pepper gửi lại cái hộp cho Lisa, Lisa có thể dùng chìa khóa đang giữ để mở hộp và đọc thư.

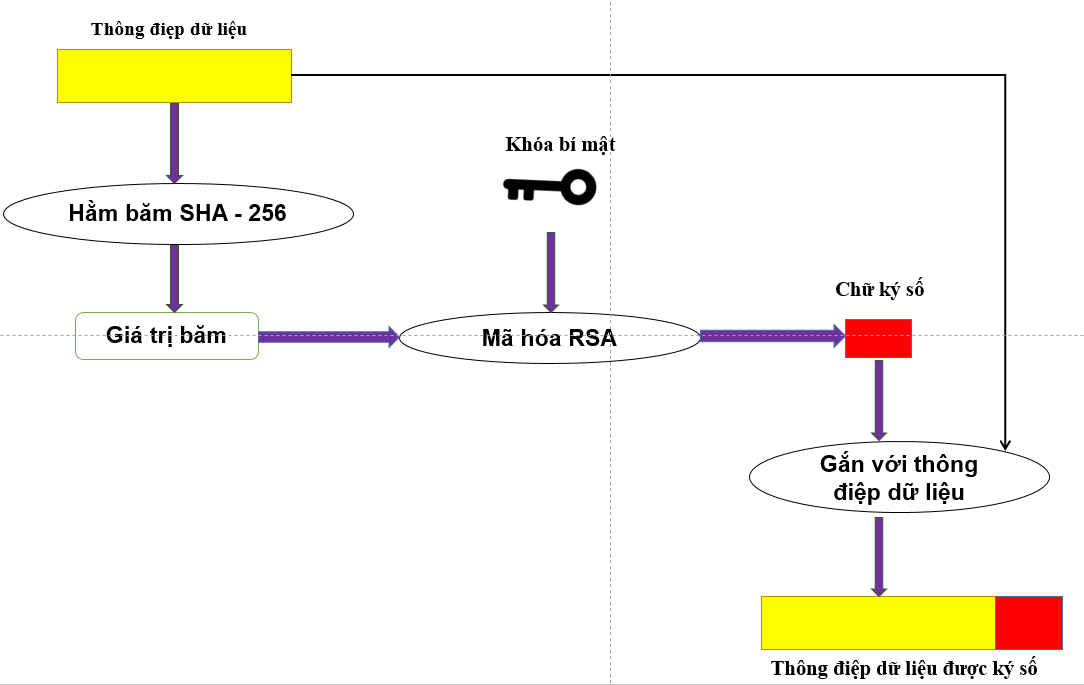
*Trong đó,*

* Cái hộp và khóa mở đóng vai trò khóa công khai; chìa khóa đóng vai trò khóa bí mật.
* Sau khi bỏ thư mật vô hộp và khóa lại, Pepper không thể mở lại hay sửa thông tin trong thư. Chỉ Lisa có thể mở được cái hộp đó.

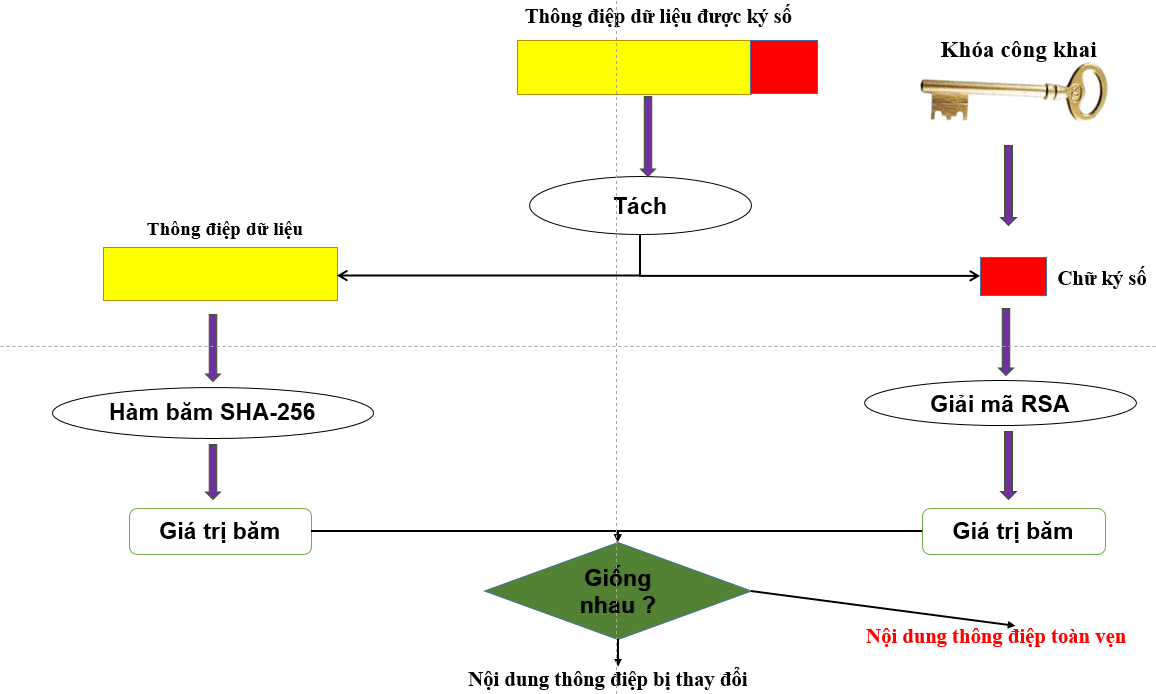
Từ những thuyết và định nghĩa trên, ta có các sơ đồ miêu tả cụ thể 3 quá trình của chữ ký số RSA bao gồm: *Quy trình sinh khóa****,*** *Quy trình tạo chữ ký số RSA, Quy trình thẩm định chữ ký số RSA.*

******

Hình 1.1 Quy trình sinh khóa



Hình 1.2 Quy trình tạo chữ ký số RSA



Hình 1.3 Quy trình thẩm định chữ ký số RSA

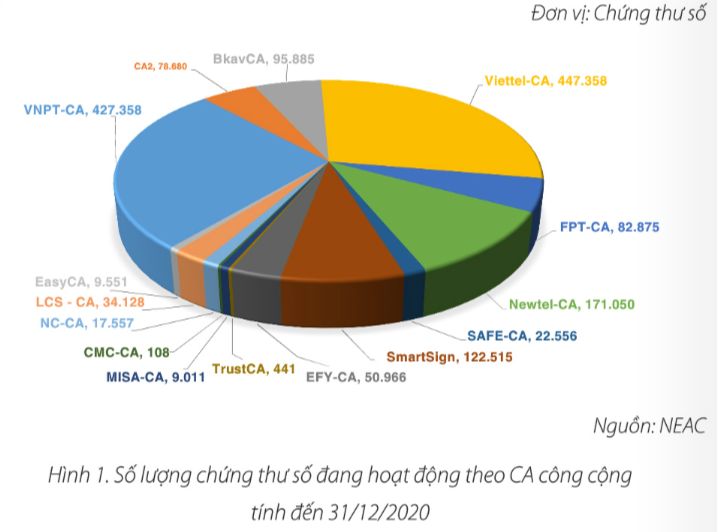
## Ứng dụng của chữ ký số tại Việt Nam

Chữ ký số đã và đang đóng vai trò quan trọng trong công cuộc chuyển đổi số. Với cấu tạo dựa trên công nghệ RSA đảm bảo tính chính xác và bảo mật, chữ ký số mang lại những chức năng tiện lợi, phù hợp với sự thay đổi và phát triển của thời đại.

Phạm vi ứng dụng của chữ ký rất rộng và bao gồm nhiều lĩnh vực như ký số trong thư điện tử cho phép khách hàng xác định người gửi, sử dụng chữ ký số thực hiện việc đăng ký các văn bản xác nhận khi đầu tư chứng khoán trực tuyến, bán hàng trực tuyến, thanh toán trực tuyến, chuyển tiền trực tuyến; Ký số trong hợp đồng kinh tế mà không cần gặp nhau trực tiếp, ký trong khai nộp thuế...

Danh sách các đơn vị cung cấp chữ ký số công cộng hiện nay như sau:

|  |  |  |
| --- | --- | --- |
| 1. VNPT-CA | 1. SAFE-CA | 1. NC-CA |
| 1. CA2 | 1. SMARTSIGN | 1. LCS-CA |
| 1. Bkav-CA | 1. EFY-CA | 1. EASYCA |
| 1. Viettel-CA | 1. TrustCA | 1. FASTCA |
| 1. FPT-CA | 1. MISA-CA | 1. I-CA |
| 1. Newtel-CA | 1. CMC-CA |  |



Hình 1.4 Số lượng chứng thư số đang hoạt động theo từng CA công cộng tính đến 31/12/2022 Nguồn: (Trung tâm Chứng thực điện tử quốc gia - Bộ Thông tin và Truyền thông, 2021)

### Ứng dụng chữ ký số chuyên dùng trong cơ quan Nhà nước

Thực hiện các quy định của pháp luật trong giao dịch điện tử và chữ ký số, trong thời gian qua, các cơ quan nhà nước các cấp đã chủ động tích cực triển khai chữ ký số chuyên dùng Chính phủ trong công tác quản lý, điều hành, trao đổi văn bản điện tử, cung cấp dịch vụ công trực tuyến và đã đạt được những kết quả quan trọng đảm bảo an toàn các giao dịch điện tử, tạo môi trường hiện đại, tiết kiệm thời gian và chi phí; nâng cao hiệu quả công việc, tăng tính công khai, minh bạch trong quản lý, điều hành; góp phần tích cực trong việc đẩy mạnh ứng dụng công nghệ thông tin, cải cách hành chính và phát triển Chính phủ điện tử.

Từ năm 2019, 100% Bộ, cơ quan ngang Bộ, cơ quan thuộc Chính phủ đã hoàn thành việc kết nối liên thông trên Trục liên thông văn bản quốc gia và đã ứng dụng chữ ký số hiệu quả để trao đổi văn bản điện tử. Một số Bộ, ngành tập trung đẩy mạnh phát triển cung cấp dịch vụ công trực tuyến ứng dụng chữ ký số hiệu quả như thuế, hải quản, bảo hiểm xã hội, kho bạc,…

Phần mềm ký số được sử dụng chủ yếu là phần mềm do Ban cơ yếu Chính phủ cung cấp cụ thể: phần mềm VGCA Sign Service; phần mềm ký số văn bản điện tử định dạng (vSignPDF), bộ công cụ tích hợp chữ ký số và xác thực chữ ký số vào các hệ thống quản lý văn bản và điều hành, dịch vụ công trực tuyến, một cửa điện tử của các bộ, ngành, địa phương, bộ công cụ tích hợp chữ ký số trên các thiết bị di động.

Năm 2020, các bộ, cơ quan ngang bộ, cơ quan thuộc Chính phủ được cấp 32520 chứng thư số, trong đó có 8656 chứng thư số cơ quan, tổ chức và 23864 chứng thư số cho cá nhân.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên đơn vị | Chứng thư số đang hoạt động | Chứng thư số đang hoạt động của công chức, viên chức | Tổng |
| 1 | Bộ Công an | 17643 | 15264 | 32907 |
| 2 | Bộ Tài chính | 3454 | 28275 | 31729 |
| 3 | Bảo hiểm xã hội Việt Nam | 1806 | 5850 | 7656 |
| 4 | Bộ Kế hoạch và Đầu tư | 751 | 5284 | 6035 |
| 5 | Bộ Tư pháp | 821 | 2483 | 3304 |

Bảng 1.1 Danh sách 5 Bộ, cơ quan ngang Bộ, cơ quan thuộc Chính phủ có số lượng chứng thư số đang hoạt động nhiều nhất năm 2020 (Nguồn: Ban Cơ yếu Chính phủ)

Thông tin, số liệu về đối tượng, phạm vi, mục đích sử dụng chữ ký số trong các Bộ, cơ quan ngang Bộ, cơ quan thuộc Chính phủ được thể hiện trong bảng sau:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Nội dung | Số lượng/Tổng số | | Tỉ lệ (%) |
| 1 | **Đối tượng sử dụng chữ ký số** | | | |
| 1.1 | Lãnh đạo cấp Bộ | 24/30 | | 80 |
| 1.2 | Lãnh đạo cấp Vụ | 25/30 | | 80,33 |
| 1.3 | Lãnh đạo cấp Phòng | 12/30 | | 40 |
| 1.4 | Văn thư/cơ quan ngang Bộ | 23/30 | | 76,67 |
| 1.5 | Kế toán trưởng/Phụ trách kế toán | 19/30 | | 63,33 |
| 1.6 | Tất cả cán bộ | 7/30 | | 23,33 |
| 1.7 | Khác | 8/30 | | 26,67 |
| 2 | **Phạm vi, mục đích sử dụng chữ ký số** | | | |
| 2.1 | Sử dụng hệ thống trục liên thông văn bản quốc gia | 30/30 | 100 | |
| 2.2 | Ký email | 7/30 | 23,33 | |
| 2.3 | Ký các văn bản ban hành nội bộ | 25/30 | 83,33 | |
| 2.4 | Ký các văn bản trao đổi với các cơ quan hành chính khác | 28/30 | 93,33 | |
| 2.5 | Báo cáo nhiệm vụ Chính phủ giao | 22/30 | 73,33 | |
| 2.6 | Sử dụng hệ thống dịch vụ công quốc gia | 16/30 | 53,33 | |
| 2.7 | Sử dụng dịch vụ công Kho bạc nhà nước | 21/30 | 70 | |
| 2.8 | Ký xử lý và trả kết quả trên công DVCTT | 9/30 | 30 | |
| 2.9 | Sử dụng trong hoạt động đấu thầu | 17/30 | 56,67 | |
| 2.10 | Kê khai thuế | 23/30 | 76,67 | |
| 2.11 | Kê khai bảo hiểm xã hội | 20/30 | 86,67 | |
| 2.12 | Kê khai hải quan điện tử | 6/30 | 20 | |
| 2.13 | Khác (\*) | 5/30 | 16,67 | |

Bảng 1.2 Đối tượng, phạm vi sử dụng chữ ký số trong các Bộ, cơ quan ngang Bộ, cơ quan thuộc Chính Phủ tính đến 31/12/2020 (Nguồn NEAC)

(\*) Phạm vi khác:

Bộ Công an: Cấp căn cước công dân điện tử.

Bộ Quốc phòng: Phần mềm quản lý công văn và hồ sơ công việc của Bộ Quốc phòng, phần mềm quản lý ngành Xe-Máy/Cục Xe-máy/Tổng cục kỹ thuật; Phần mềm quản lý công văn của Bộ Tư Lệnh Cảnh sát biển Việt Nam.

Bộ Tư pháp: Phần mềm quản lý văn bản và điều hành; Biểu mẫu thống kê ngành Tư pháp; Công tác báo cáo thu, chi hành án dân sự.

Bộ y tế: Ký các văn bản và dịch vụ công trực tuyến.

Bảo hiểm xã hội Việt Nam: Các hệ thống nghiệp vụ Ngành có ứng dụng chữ ký số: Thu và quản lý sổ - thẻ; Xét duyệt chính sách; Phần mềm kế toán tập trung; Tiếp nhận và quản lý hồ sơ; Thẩm định quyết toán; Lưu trữ hồ sơ điện tử; Quản lý đầu tư quỹ; Hệ thống thu, nộp, chi trả BHXH điện tử; Giám định BHYT.

### Ứng dụng chữ ký số trong cơ quan tổ chức

Tính đến 2020, Bộ thông tin và Truyền thông đã cấp giấy chứng nhận đăng ký hoạt động cho 6 tổ chức cung cấp dịch vụ chứng thực chữ ký số chuyên dùng của cơ quan tổ chức, bao gồm:

* Ngân hàng Nhà nước;
* Ngân hàng Thương mại Cổ phần Kỹ thương Việt Nam;
* Ngân hàng Nông nghiệp và Phát triển nông thôn Việt Nam;
* Ngân hàng Thương mại Cổ phần Đông Nam Á;
* Ngân hàng Thương mại Cổ phần Á Châu;
* Bảo hiểm xã hội Việt Nam.

#### Ứng dụng chữ ký số chuyên dùng tại Ngân hàng Thương mại Cổ phần Đông Nam Á – SeABank

Đối tượng sử dụng: Nhân sự trong nội bộ SeABank, các hệ thống, ứng dụng phục vụ cho các hoạt động nội bộ của SeABank và những đơn vị nội bộ SeABank.

Chữ ký số chuyên dùng của SeABank không sử dụng trong các giao dịch, chỉ sử dụng trong các văn bản chính sách ban hành trong nội bộ SeABank, cụ thể:

* Ký các văn bản, tài liệu có giá trị sử dụng trong hoạt động nội bộ của SeABank.
* Xác thực email trong SeABank.
* Xác thực đa thành phần nhằm đảm bảo an toàn với các hệ thống máy chủ, ứng dụng quan trọng của SeABank.
* Mã hóa, bảo vệ thông tin truyền tải qua mạng.

Tần suất sử dụng khoản 5200 lần ký trong tháng.

Các văn bản sử dụng chữ ký số chuyên dùng: tờ trình, báo cáo, tài liệu, chính sách, file dữ liệu phục vụ các hoạt động nghiệp vụ nội bộ SeABank. (Trung tâm Chứng thực điện tử quốc gia - Bộ Thông tin và Truyền thông, 2021)

#### Ứng dụng chữ ký số chuyên dùng tại Ngân hàng Nông nghiệp và Phát triển nông thôn Việt Nam – Agribank

Đối tượng sử dụng: Cán bộ, nhân viên Ngân hàng Agribank.

Ngân hàng Agribank ứng dụng chữ ký số để xác thực vào hệ thống và ký giao dịch chuyển tiền, tiền gửi nội bộ.

Tần suất sử dụng chữ ký số chuyên dùng để ký xác thực, giao dịch khoảng 420.000 lần trong ngày.

Hiệu quả sử dụng chữ ký số chuyên dùng tại Agribank: chống sửa đổi giao dịch sau khi đã ký, tăng cường an toàn trong xác thực đăng nhập hệ thống công nghệ thông tin. (Trung tâm Chứng thực điện tử quốc gia - Bộ Thông tin và Truyền thông, 2021)

#### Ứng dụng chữ ký số chuyên dùng tại Ngân hàng Nhà nước Việt Nam

Ngân hàng Nhà nước Việt Nam cung cấp các dịch vụ công trực tuyến cho các doanh nghiệp, cụ thể là các tổ chức tín dụng.

Đối tượng sử dụng: là những người thuộc Ngân hàng Nhà nước Việt Nam, các tổ chức tín dụng, Kho bạc Nhà nước, công ty tài chính và gần 1200 Quỹ tín dụng nhân dân.

Phạm vi sử dụng: Trong các ứng dụng nghiệp vụ của Ngân hàng Nhà nước Việt Nam như hệ thống thanh toán điện tử liên ngân hàng, hệ thống đấu thầu và nghiệp vụ thị trường mở, hệ thống quản lý văn bản và điều hành tác nghiệp (eDOC), hệ thống báo cáo mới và kho dữ liệu, hệ thống cung cấp dịch vụ công trực tuyến, hệ thống thư điện tử và hệ thống báo cáo bảo hiểm tiền gửi,… (Trung tâm Chứng thực điện tử quốc gia - Bộ Thông tin và Truyền thông, 2021)

#### Ứng dụng chữ ký số chuyên dùng tại Ngân hàng TMCP Kỹ thương Việt Nam Techcombank

Tại Techcombank, đối tượng sử dụng chữ ký số là cán bộ, nhân viên trong nội bộ hệ thống ngân hàng. Các loại giao dịch yêu cầu sử dụng chữ ký số chuyên dùng: Ký văn bản điện tử nội bộ Techcombank với tần suất sử dụng là trên 3000 lượt ký số mỗi tháng. (Trung tâm Chứng thực điện tử quốc gia - Bộ Thông tin và Truyền thông, 2021)

### Ứng dụng chữ ký số trong các lĩnh vực

#### Ứng dụng chữ ký số trong lĩnh vực thuế

Tính đến 31/12/2020, Tổng Cục Thuế - Bộ Tài Chính đang cung cấp 153 dịch vụ công trực tuyến (DVCTT) sử dụng chữ ký số để xác thực, trong đó có 18 DVCTT ở mức độ 3 và 135 DVCTT ở mức độ 4. Một vài dịch vụ công trực tuyến tiêu biểu theo Báo cáo của NEAC, 2021 gồm:

* Dịch vụ khai thuế điện tử: Số lượng doanh nghiệp sử dụng dịch vụ khai thuế qua mạng 50314 doanh nghiệp so với cùng kỳ năm 2019.
* Nộp thuế qua mạng: Tính đến 31/12/2020, số lượng doanh nghiệp đăng ký tham gia sử dụng dịch vụ nộp thuế với cơ quan thuế tăng 43149 so với cùng kỳ 2019.
* Hóa đơn điện tử.
* Dịch vụ hoàn thuế…

Triển khai ứng dụng chữ ký số trong các DVCTT để xác thực tại Tổng cục thuế giúp cho giao dịch điện tử giữa người nộp thuế với cơ quan thuế được đảm bảo an toàn, dữ liệu được toàn vẹn, xác thực. Đồng thời việc xác thực chữ ký số giúp doanh nghiệp cắt giảm nhiều thủ tục hành chính, giảm thời gian đi lại đến các cơ quan thuế trực tiếp nộp hồ sơ và giúp đẩy mạnh công tác cải cách thủ tục hành chính trong cơ quan thuế.

#### Ứng dụng chữ ký số trong lĩnh vực hải quan

Năm 2020, Tổng cục Hải quan cung cấp 238 DVCTT trong đó có 202 DVCTT ứng dụng chữ ký số để xác thực (chiếm 85% tổng số thủ tục hành chính) và 196 DVCTT mức độ 4 ứng dụng chữ ký số để xác thực. (Trung tâm Chứng thực điện tử quốc gia - Bộ Thông tin và Truyền thông, 2021)

Ứng dụng chữ ký số trong dịch vụ công trực tuyến của Tổng cục Hải quan đã góp phần đảm bảo tính pháp lý của các giao dịch điện tử giữa các cơ quan hải quan và các doanh nghiệp tham gia hoạt động xuất nhập khẩu; đảm bảo bảo mật an toàn, chống chối bỏ, giảm thiểu những thiệt hại về kinh tế; giảm thiểu thời gian thông quan hàng hóa, các chi phí phát sinh khi làm thủ tục xuất nhập khẩu, doanh nghiệp không cần tiếp xúc với các cơ quan nhà nước và giảm thiểu nhân sự trong công tác xác minh, kiểm tra các doanh nghiệp khai báo hải quan điện tử.

#### Ứng dụng chữ ký số trong lĩnh vực bảo hiểm xã hội

Trong những năm qua, ngành bảo hiểm xã hộ đã có nhiều đổi mới trong công tác cải cách thủ tục hành chính và ứng dụng công nghệ thông tin vào các hoạt động nghiệp vụ với mục tiêu cung cấp nhiều dịch vụ, tiện ích cho người tham gia và hưởng chế độ chính sách bảo hiểm xã hội, bảo hiểm y tế.

Năm 2020, Bảo hiểm y tế Việt Nam ban hành văn bản số 3968/BHXH-CNTT triển khai bổ sung 09 DVCTT mức độ 4, theo đó Bảo hiểm Xã hội Việt Nam đã hoàn thành cung cấp 100% các dịch vụ công mức độ 4 cho các thủ tục hành chính của ngành, cụ thể 30 DVCTT có ứng dụng chữ ký số mức 4 dành cho doanh nghiệp, trong đó 14 DVCTT dành cho doanh nghiệp/cơ quan và 16 DVCTT dành cho cá nhân. (Trung tâm Chứng thực điện tử quốc gia - Bộ Thông tin và Truyền thông, 2021)

Triển khai DVCTT ứng dụng chữ ký số đã góp phần đơn giản hóa quy trình gửi, nhận hồ sơ; bảo mật danh tính doanh nghiệp, cá nhân giao dịch; linh hoạt trong cách nộp hồ sơ giao dịch điện tử; hỗ trợ cơ quan tổ chức thực hiện giao dịch hồ sơ bảo hiểm xã hội với cơ quan bảo hiểm được thuận tiện và tiết kiệm được thời gian, chi phí trong quá trình thực hiện giao dịch.

#### Ứng dụng chữ ký số trong lĩnh vực chứng khoán

Hiện nay, Ủy Ban chứng khoán Nhà nước đang sử dụng 154 chứng thư số chuyên dùng do Ban Cơ yếu Chính phủ cấp, trong đó 2 chứng thư số của cơ quan, tổ chức và 143 chứng thư số của công chức, viên chức và người lao động. Chứng thư số chuyên dùng được sử dụng để thực hiện ký số và mã hóa các tập tin báo cáo trước khi gửi lên hệ thống, đảm bảo an toàn bảo mật trên đường truyền trong một số hệ thống nội bộ ngành chứng khoán.

Các hệ thống phần mềm dùng cho các đối tượng bên người để nhận báo cáo đều được sử dụng chữ ký số công cộng để ký số các tập tin như văn bản, báo cáo, tài liệu,… trước khi gửi lên hệ thống đảm bảo tính pháp lý của văn bản.

Chữ ký số công cộng cũng được sử dụng trong các DVCTT của Ủy ban Chứng khoán Nhà nước. Tính đến 31/12/2020, Ủy ban Chứng khoán Nhà nước đã cấp 165 DVCTT trong đó có 132 DVCTT yêu cầu tổ chức, doanh nghiệp khi nộp hồ sơ phải sử dụng chữ ký số để xác thực trước khi gửi lên hệ thống. (Trung tâm Chứng thực điện tử quốc gia - Bộ Thông tin và Truyền thông, 2021)

Việc ứng dụng chữ ký số trong các DVCTT đảm bảo tính xác thực của hồ sơ do các tổ chức gửi lên hệ thống, công tác lưu trữ hồ sơ, dữ liệu khoa học hiểu quả, nhanh chóng trong công tác tra cứu dữ liệu.

#### Ứng dụng chữ ký số trong các hoạt động nghiệp vụ của Kho bạc Nhà nước

Ứng dụng chữ ký số trong các 7 DVCTT:

* Thủ tục kiểm soát thanh toán các khoản chi thường xuyên, chi sự nghiệp có tính chất thường xuyên, chỉ chương trình mục tiêu quốc gia, chương trình mục tiêu sử dụng kinh phí sự nghiệp.
* Thủ tục kiểm soát thanh toán vốn đầu tư nguồn vốn ngân sách.
* Thủ tục kiểm soát, thanh toán chi phí quản lý dự án đầu tư của các dự án sử dụng vốn ngân sách kho bạc Nhà nước.
* Thủ tục kiểm soát chi vốn nước ngoài qua Kho bạc Nhà nước.
* Thủ tục hạch toán vốn ODA, vốn vay ưu đãi vào ngân sách nhà nước.
* Thủ tục đăng ký sử dụng tài khoản, bổ sung tài khoản và thay đổi mẫu dấu, mẫu chữ ký của đơn vị tại Kho bạc Nhà nước.
* Thủ tục hoàn tất tài khoản của đơn vị giao dịch mở tại kho bạc Nhà nước.

Dịch vụ công trực tuyến của Kho bạc Nhà nước sử dụng chữ ký số đã tạo thuận tiên, tăng tiện ích, giảm thời gian giao dịch cũng như tiện lợi trong quá trình giao dịch. (Trung tâm Chứng thực điện tử quốc gia - Bộ Thông tin và Truyền thông, 2021)

#### Ứng dụng chữ ký số trong hoạt động của hệ thống mạng đấu thầu quốc gia

Hệ thống mạng đấu thầu quốc gia cung cấp các dịch vụ công liên quan đến hoạt động đấu thầu sử dụng chữ ký số công cộng và chữ ký số chuyên dùng do Bộ Kế hoạch và Đầu tư xây dựng và cấp.

Theo số liệu của Cục quản lý đấu thầu – Bộ kế hoạch và Đầu tư cung cấp, số lượng người sử dụng trên hệ thống mạng đấu thầu quốc gia năm 2020 cụ thể như sau:

* Chứng thư số bên mời thầu: 6412 tăng 39 chứng thư số so với năm 2019
* Chứng thư số nhà thầu: 18825 tăng 1402 chứng thư số so với năm 2019

Việc ứng dụng chữ ký số vào hệ thống mạng đấu thầu quốc gia đảm bảo các giao dịch chính xác, toàn viện dữ liệu. Đầu thầy qua mạng góp phần nâng cao hiệu quả của các công tác đấu thầu, tạo môi trường cạnh tranh công bằng, minh bạch, tiết kiệm thời gian, chi phí, giảm thủ tục hành chính.

# TIÊU CHUẨN CHỮ KÝ SỐ RSA

## Các quy định về chữ ký số RSA

### Giải thích từ ngữ

#### Mật mã

Là những quy tắc, quy ước riêng dùng để thay đổi hình thức biểu hiện thông tin nhằm bảo đảm bí mật, xác thực, toàn vẹn của nội dung thông tin.

#### Mật mã dân sự

Là kỹ thuật mật mã và sản phẩm mật mã được sử dụng để bảo mật hoặc xác thực đối với thông tin không thuộc phạm vi bí mật nhà nước.

#### Kỹ thuật mật mã

Là phương pháp, phương tiện có ứng dụng mật mã để bảo vệ thông tin.

#### Mã hóa

Phép biến đổi (khả nghịch) dữ liệu bởi thuật toán mật mã để tạo ra bản mã, tức là che giấu nội dung thông tin của dữ liệu.

#### Giải mã

Phép toán ngược với phép mã hóa tương ứng.

#### Mật mã phi đối xứng

Hệ thống dựa trên kỹ thuật mật mã phi đối xứng, trong đó phép biến đổi công khai được sử dụng để mã hóa, phép biến đổi bí mật được sử dụng để giải mã.

#### Kỹ thuật mật mã phi đối xứng

Kỹ thuật mật mã phi đối xứng sử dụng hai phép biến đổi liên quan đến nhau, phép biến đổi công khai (được xác định bởi khóa công khai) và phép biến đổi bí mật (được xác định bởi khóa riêng). Cả hai phép biến đổi này có tính chất là cho biết phép biến đổi công khai, về mặt tính toán không thể có khả năng xác định được phép biến đổi bí mật.

#### Chữ ký số

Một chuỗi số, kết quả của phép biến đổi mật mã trên thông điệp dữ liệu nhằm cung cấp một phương tiện để kiểm tra tính xác thực của nguồn gốc thông điệp dữ liệu, tính toàn vẹn của dữ liệu và tính không thể chối bỏ của người đã ký.

#### Các ký hiệu

|  |  |
| --- | --- |
| *n* | Modulo RSA |
| *d* | Số mũ bí mật RSA |
| *e* | Số mũ công khai RSA |
| *(n,e)* | Khóa công khai RSA của người ký |
| *C* | Bản mã được biểu diễn ở dạng chuỗi octet |
| *M* | Thông điệp (văn bản), chuỗi octet |
| *N* | là ký hiệu độ dài theo bit của số nguyên tố q |
| *L* | là ký hiệu độ dài theo bit của số nguyên tố p |
| *||* | Toán tử nối hai chuỗi |
| *PKCS* | Tiêu chuẩn mật mã khóa công khai (Puplic Key Cryptography Standard) do Phòng thí nghiệm RSA (Mỹ) ban hành. |
| *PSS* | Lược đồ ký xác suất (Probabilistic Signature Scheme) |
| *RSA* | Tên của hệ mã do ba nhà toán học Rivest, Shamir và Adleman RSA sáng tạo ra |
| *RSAVP* | Phép toán cơ sở phục vụ cho kiểm tra chữ ký RSA |
| *RSASP* | Phép toán ký RSA cơ sở |
| *RSASSA* | Lược đồ ký RSA kèm phụ lục (RSA Signature Scheme with Appendix) |
| *SHA* | Thuật toán băm an toàn (Secure Hash Algorithm). |
| *Word* | Từ (32 bit) |
| *ECDSA* | Thuật toán chữ ký số dựa trên đường cong Elliptic |

Bảng 2.1 Các ký hiệu

#### Quy định kỹ thuật

|  |  |
| --- | --- |
| **Quá trình chữ ký số** | **Sử dụng** |
| Tạo chữ ký số | Độ an toàn ≥ 112 bit:  DSA: |*p*| ≥ 2048 và |*q*| ≥ 224  RSA: |*n|* ≥ 2048  EC: |*n*| ≥ 224 |
| Xác thực chữ ký số | Độ an toàn ≥ 112 bit:  DSA: |*p*| ≥ 2048 và |*q*| ≥ 224  RSA: |*n|* ≥ 2048  EC: |*n*| ≥ 224 |

Bảng 2.2 Quy định kỹ thuật

### Các yêu cầu chung về cặp khóa RSA

1. Cặp khóa RSA dùng để ký thì không được dùng cho mục đích khác (chẳng hạn dùng lại để mã thông điệp);
2. Hai số nguyên tố **p**, **q** và số mũ bí mật **d** cần phải được giữ bí mật tránh việc bị truy cập bất hợp pháp, làm lộ hoặc sửa đổi. Modulo **n** và số mũ công khai **e** phải được công bố công khai;
3. Mỗi người sử dụng cần có Modulo n riêng;
4. Độ dài của Modulo **n** (nlen) không được nhỏ hơn 2048 bit và nên được thay đổi theo thời gian như sau.

|  |  |  |
| --- | --- | --- |
| **Năm (y)** | **độ an toàn** | ***nlen* tối thiểu** |
|  | 112 | 2048 |
| 128 | 3072 |

Trong đó, độ mạnh về an toàn (security\_strength) là một số nguyên biểu thị lượng tính toán cần thiết đề phá hệ mã. Vì các phương pháp phá hệ mã thường xuyên được hoàn thiện nên cần phải định kỳ 3 đến 5 năm một lần xem xét lại nlen tối thiểu (có thể tham khảo chi tiết yêu cầu này trong tài liệu NIST Special Publication 800-57: Recommendation for Key Management - Part1: General, January 2016).

* Phiên bản áp dụng: Áp dụng phiên bản 2.1 của tiêu chuẩn RSA Cryptography Standard PKCS #1 v2.1.
* Áp dụng lược đồ RSAES-OAEP để mã hóa và RSASSA-PSS để ký.

### Quy định chi tiết về các khóa RSA

* Số mũ công khai **e** cần phải được chọn với các ràng buộc sau:

1. Số mũ công khai **e** cần được chọn trước khi tạo số mũ bí mật **d**;
2. Số mũ công khai **e** cần phải là số nguyên dương lẻ sao cho:   
   Với nlen là độ dài của modulo **n** theo bit.

***Chú ý:***  **e** có thể là giá trị bất kỳ mà thỏa mãn ràng buộc 1(b); **p** và **q** sẽ được chọn (trong mục 2) sao cho **e** là nguyên tố cùng nhau với cả (**p** - 1) và (**q** - 1).

* Hai số nguyên tố **p** và **q** được tạo ngẫu nhiên và giữ bí mật cần phải được chọn với các ràng buộc sau:

1. (**p** - 1) và (**q** - 1) cần phải là nguyên tố cùng nhau với số mũ công khai e;
2. Mỗi một trong bốn số (**p** + 1), (**p** - 1) và (**q** + 1), (**q** - 1) cần phải có các nhân tử nguyên tố lớn hơn 2security-strength+20;
3. Nhân tử nguyên tố bí mật p, q cần phải được chọn ngẫu nhiên từ các số nguyên tố thỏa mãn ≤ q < p ≤ ;
4. |**p - q**| > .

* Số mũ bí mật **d** cần phải được lựa chọn sau khi tạo **p** và **q** với các ràng buộc:

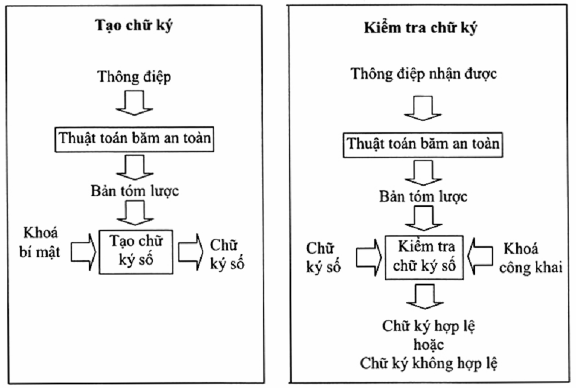
1. Số mũ **d** cần phải lớn hơn

(Chi tiết về hàm tạo các tham số RSA có thể tham khảo trong tài liệu FIPS 186-4: Digital Signature Standard).

## Giới thiệu về Tiêu chuẩn Quốc gia Việt Nam về chữ ký số (TCVN 7635:2007 Các kỹ thuật mật mã - Chữ ký số)

Ban hành từ quyết định số 1470/QĐ-BKHCN ngày 14/8/2007 của Bộ trưởng Bộ Khoa học - Công nghệ, TCVN 7635:2007 “Các kỹ thuật mật mã - Chữ ký số” là tiêu chuẩn quốc gia đầu tiên trong lĩnh vực Kỹ thuật mật mã do Tiểu Ban kỹ thuật tiêu chuẩn TCVN/JTC 1/SC 27 “Các kỹ thuật mật mã” biên soạn theo đề nghị của Ban Cơ yếu Chính phủ. Tiêu chuẩn này áp dụng cho các chữ ký số trong hoạt động giao dịch điện tử của mọi tổ chức, công dân Việt Nam và tổ chức, công dân nước ngoài có quan hệ kinh tế - xã hội với tổ chức, công dân Việt Nam và được viện dẫn theo Tiêu chuẩn Mã hóa tiên tiến của Mỹ FIPS 197: Advanced Encryption Standard (26/11/2001).

Tiểu Ban kỹ thuật TCVN/JTC 1/SC 27 đã thống nhất một số nguyên tắc sau: không nên đưa ra quá nhiều phương án trong Tiêu chuẩn bởi điều này có thể gây khó khăn cho người dùng trong khi lựa chọn;  ưu tiên lựa chọn Tiêu chuẩn được nhiều nước trên thế giới sử dụng vào thời điểm đó, nhằm tạo điều kiện thuận lợi trong việc tuân thủ yêu cầu về hội nhập khu vực và quốc tế và cũng thể hiện được tính an toàn cao qua việc trải nghiệm thực tiễn. Bởi vậy, Tiêu chuẩn quy định 3 thành phần cần thiết cho lược đồ chữ ký số: Thành phần thứ nhất là thuật toán chữ ký số RSA-PSS (theo RSA Cryptography Standard – 2002). Thành phần thứ hai là thuật toán hàm băm SHA-256 (theo FIPS PUB 180-2: Secure Hash Standard – 2001) và thành phần cuối cùng là thuật toán tạo số giả ngẫu nhiên dùng AES-128 (theo FIPS PUB 197: Advanced Encryption Standard – 2001). (Tiểu Ban kỹ thuật tiêu chuẩn TCVN/JTC 1/SC 27 “Các kỹ thuật mật mã”, 2007)



Hình 2.1 Mô tả quá trình tạo và kiểm tra chữ ký số

### Thuật toán chữ ký số RSA-PSS

RSA-PSS là viết tắt của RSASSA-PSS (RSA Signature Scheme with Appendix - Probabilistic Signature Scheme hay Lược đồ chữ ký RSA có phụ lục - Lược đồ chữ ký xác suất). Ở thuật toán chữ ký số RSA-PSS, các thao tác tạo chữ ký số áp dụng bước định dạng một thông điệp trước khi nó được chuyển thành một biểu diễn thông điệp ở dạng số nguyên. Phép toán cơ sở RSASP được áp dụng vào biểu diễn thông điệp này để tạo ra chữ ký số. Đảo ngược quá trình này, các thao tác kiểm tra chữ ký áp dụng một phép toán cơ sở RSAVP vào một chữ ký để khôi phục một biểu diễn thông điệp, sau đó nó được chuyển thành thông điệp đã được định dạng ở dạng chuỗi octet. Thao tác kiểm tra được áp dụng vào thông điệp ban đầu và thông điệp đã được định dạng để xác định xem chúng có tương ứng với nhau hay không.

Thuật toán sử dụng dạng chuỗi octet tương đương với 8 bit (theo chuẩn ISO) thay cho ký hiệu byte nhằm tránh sự nhầm lẫn do thuật ngữ “byte” trong quá khứ mang nhiều giá trị: 6 bits, 8 bits, 9 bits, …

#### Các hàm hỗ trợ thuật toán

##### Các hàm cơ sở chuyển đổi dữ liệu

* Hàm cơ sở chuyển đổi dữ liệu số nguyên sang dạng chuỗi octet

**I2OSP *(x, xLen)***

|  |  |  |
| --- | --- | --- |
| Chức năng | Chuyển số nguyên không âm x thành một chuỗi octet có độ dài *xLen* | |
| Đầu vào | x | Số nguyên không âm cần chuyển đổi |
| Đầu ra | X | Chuỗi octet tương ứng có độ dài xLen |
| Thông báo lỗi | “số nguyên quá lớn” | |

Các bước:

* 1. Nếu *x* ≥ 256*xLen*, cho ra thông báo lỗi “số nguyên quá lớn” và dừng
  2. Viết số nguyên *x* duy nhất gồm *xLen* chữ số với cơ số 256:   
     *x* = *xxLen-1*·256*xLen-1* + *xxLen-2*·256*xLen-2 + … + x1*·256 + *x*0Với 0 ≤ *xi* < 256 (chú ý rằng một hay nhiều chữ số đầu sẽ bằng 0 nếu x nhỏ hơn 256*xLen-1).*
  3. Cho octet *Xi* giá trị nguyên *xxLen-i* với 1 ≤ *i* ≤ *xLen*.   
     Cho ra chuỗi octet *X* = *X1 X2*… *XxLen*
* Hàm cơ sở chuyển đổi từ dạng chuỗi octet về dạng số nguyên

**OS2IP (X)**

|  |  |  |
| --- | --- | --- |
| Chức năng | Chuyển chuỗi octet thành một số nguyên không âm | |
| Đầu vào | X | Chuỗi octet cần chuyển đổi |
| Đầu ra | x | Số nguyên không âm tương ứng |

Các bước:

* 1. Cho *X1X2…XxLen* là các octet của X từ octet đầu tiên với octet cuối cùng, *xxLen-i* là giá trị nguyên của octet Xi với 1 ≤ i ≤ *xLen*;
  2. Cho x = *xxLen-1*·256*xLen-1* + *xxLen-2*·256*xLen-2 + … + x1*·256 + *x*0;
  3. Xuất ra x.

##### Các hàm định dạng chữ ký kèm phụ lục theo PSS (EMSA-PSS)

* Hàm tạo mặt nạ MGF dựa vào hàm băm

**MGF*(mgfSeed, maskLen)***

Trong đó:

|  |  |  |
| --- | --- | --- |
| Lựa chọn | *h* | Hàm băm (độ dài đầu ra của nó theo octet là *hLen)* |
| Đầu vào | *mgfSeed* | Mầm được dùng để tạo mặt nạ, là chuỗi octet |
| *maskLen* | Độ dài chủ ý theo octet của mặt nạ, nhiều nhất là *232hLen* |
| Đầu ra | *mask* | Mặt nạ, là chuỗi octet có độ dài *maskLen* |
| Thông báo lỗi | “mặt nạ quá dài” | |

Các bước:

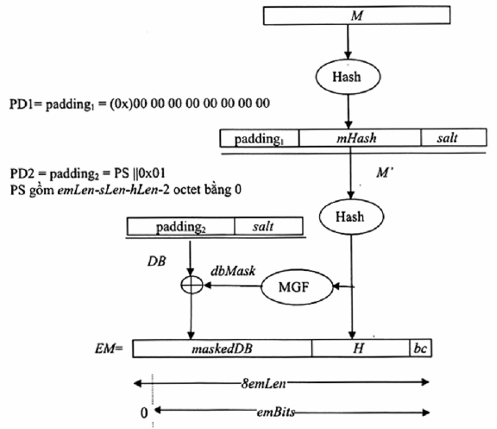
1. Nếu *maskLen* > 232 *hLen*, cho ra thông báo lỗi “mặt nạ quá dài” và dừng;
2. Lấy *T* là chuỗi octet rỗng;
3. Với *counter* chạy từ 0 tới ⎡*maskLen/hLen*⎤ -1, thực hiện các bước
4. Chuyển *counter* thành một chuỗi octet C có độ dài 4 octet;   
   C = I2OSP (*counter,* 4)
5. Nối hàm *mgfSeed* với C, tính hàm băm của chuỗi này. Sau đó nối chuỗi octet *T* với giá trị băm vừa thu được.   
   T = T || *h*(*mgfSeed* || C)
6. Xuất ra *maskLen* octet đầu tiên của T như là chuỗi octet *mask*.

* Mã hóa EMSA-PSS

**EMSA-PSS-ENCODE *(M, emBits)***

Trong đó:

|  |  |  |
| --- | --- | --- |
| Lựa chọn | *h* | Hàm băm (độ dài đầu ra của nó theo octet là hLen) |
| *MGF* | Hàm tạo mặt nạ |
| *sLen* | Đô dài chủ định của phần phụ thêm theo octet |
| Đầu vào | *M* | Văn bản để mã hóa, là một chuỗi octet |
| emBits độ dài tối đa theo bit của số nguyên OS2IP (EM), ít nhất bằng 8hLen + 8sLen + 9 | |
| Đầu ra | *EM* | Văn bản đã được mã, đó là chuỗi octet có độ dài emLen = ⎡emBits/8⎤ |
| Thông báo lỗi: “lỗi định dạng”; “văn bản quá dài” | | |



Hình 2.2 Mã hóa EMSA-PSS

Các bước:

1. Nếu độ dài của *M* lớn hơn giới hạn đầu vào cho hàm băm (264-1 đối với SHA-256 thì cho ra thông báo lỗi “văn bản quá dài” và dừng;
2. Lấy *mHash = h(M)*, đó là một chuỗi octet dài *hLen*;
3. Nếu *emLen < hLen* + *sLen* + 2, cho ra thông báo “lỗi định dạng” và dừng;
4. Tạo ra chuỗi octet ngẫu nhiên salt có độ dài *sLen*; nếu *sLen* = 0 thì salt không có;
5. Đặt *M’* = (0x)00 00 00 00 00 00 00 00 || *mHash* ||*salt*;
6. Lấy *H = Hash(M’)*, đó là một chuỗi octet dài *hLen*;
7. Lấy PS là một chuỗi octet bằng 0 dài *emLen* - *hLen* - *sLen* - 2;
8. Lấy *DB = PS* || 0x01 || *salt*; *DB* là một chuỗi octet dài *emLen-hLen* - 1;
9. Lấy *maskedDB* = *DB* ⊕ *dbMask*;
10. Đặt *8emLen-emBits* bit đầu tiên bên trái của octet đầu tiên bên trái trong *maskedDB* bằng 0;
11. Lấy *EM = maskedDB* ||*H* || *0xbc*;
12. Xuất ra *EM*.

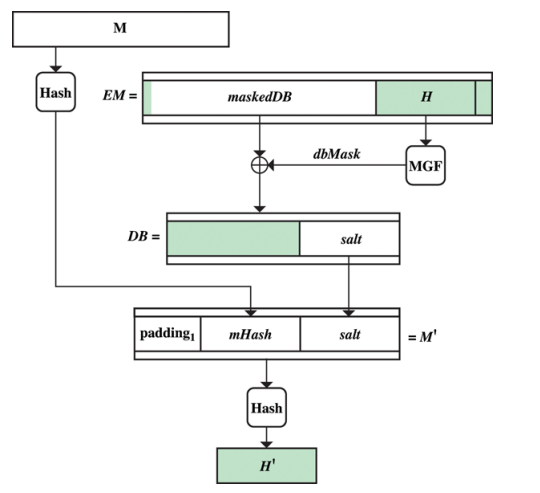
Ta có công thức để tính EM:

**((PD2||*r)* ⊕ MGF(*h(*PD1||*h(*M*)*||*r*))) || *h(*PD1||*h(M)||r)) || 0xbc***

* Kiểm tra EMSA-PSS-VERIFY

**EMSA-PSS-VERIFY*(M, EM, emBits)***

|  |  |  |
| --- | --- | --- |
| Lựa chọn | *h* | Hàm băm (độ dài đầu ra của nó theo octet là *hLen)* |
| *MGF* | Hàm tạo mặt nạ |
| *sLen* | Đô dài dự kiến của phần thêm theo octet |
| Đầu vào | *M* | Thông điệp cần kiểm tra chữ ký, là chuỗi octet |
| *EM* | Thông điệp đã được định dạng, là chuỗi octet có độ dài *emLen* = ⎡*emBits/8*⎤ |
| *emBits* Độ dài tối đa theo bit của số nguyên OS2IP (EM), tối thiểu là *8hLen + 8sLen + 9* | |
| Đầu ra | “phù hợp” hoặc “không phù hợp”. | |



Hình 2.3 Kiểm tra EMSA-PSS

Các bước:

1. Nếu độ dài của *M* lớn hơn giới hạn đầu vào của hàm băm (264 - 1 octet đối với SHA-256, thì đưa ra thông báo “không phù hợp” và dừng;
2. Đặt *mHash = h(M)*, là chuỗi octet có độ dài *hLen*;
3. Nếu *emBits* < *8hLen+8sLen+9*, đưa ra thông báo “không phù hợp” và dừng;
4. Nếu octet đầu tiên bên phải của EM không chứa giá trị bc, đưa ra thông báo “không phù hợp” và dừng;
5. Đặt *maskedDB* là *emLen-hLen-1* octet đầu tiên bên trái của *EM*, và *H* là *hLen* octet tiếp theo;
6. Nếu *8emLen-emBits* bit đầu tiên bên trái của octet đầu tiên bên trái trong *maskedDB* không phải tất cả bằng 0, đưa ra thông báo “không phù hợp” và dừng;
7. Đặt *dbMask = MGF*(*H, emLen-hLen-1*);
8. Đặt *DB = maskedDB* ⊕ *dbMask*;
9. Thiết lập *8emLen-emBits* bit đầu tiên bên trái của *DB* bằng 0;
10. Nếu *emLen-hLen-sLen-2* octet đầu tiên bên trái của *DB* không phải bằng 0 hoặc nếu octet tại vị trí thử *emLen-hLen-sLen-1* không bằng 0x01, đưa ra thông báo “không phù hợp” và dừng;
11. Đặt *salt* bằng *sLen* octet cuối cùng của DB;
12. Đặt M’ = 00 00 00 00 00 00 00 00 || *mHash* || *salt*   
    M’ là chuỗi octet có độ dài *8+hLen+sLen* với 8 octet bằng 0 khởi đầu;
13. Đặt *H’* *= h(M’)*, là chuỗi octet có độ dài *hLen*;
14. Nếu *H = H’* thì đưa ra thông báo “phù hợp”. Ngược lại, đưa ra thông báo “không phù hợp”.

##### Các phép toán mật mã cơ sở

* Phép toán cơ sở RSASP

**RSASP *(K, m)***

|  |  |  |
| --- | --- | --- |
| Đầu vào | *K* | Khóa bí mật RSA, với K có một trong hai dạng sau:  - Cặp (*n, d*);  - Bộ năm *(p, q, dP, dQ, qlnv)*; |
| *m* | Biểu diễn của thông điệp, dưới dạng số nguyên giữa 0 và *n*-1 |
| Đầu ra | s | Biểu diễn của chữ ký, là số nguyên giữa 0 và *n*-1 |
| Thông báo lỗi | “biểu diễn thông điệp ở ngoài miền hợp lệ” | |
| Giả thiết | K là một khóa bí mật RSA hợp lệ | |

Các bước:

* 1. Nếu biểu diễn của thông điệp *m* không nằm giữa 0 và *n*-1, cho ra thông báo lỗi “biểu diễn thông điệp ở ngoài miền hợp lệ” và dừng lại;
  2. Biểu diễn của chữ ký được tính như sau:

1. Nếu dạng thứ nhất (*n, d*) của *K* được sử dụng thì *s* = *md* mod *n*;
2. Nếu dạng thứ hai (*p, q, dP, dQ, qlnv*) của K được sử dụng thì tiến hành như sau:
3. Lấy *s1* = *mdP* mod *p* và *s2* = m*dQ* mod *q*
4. Đặt *h* = (*s1* - *s2*)·*qlnv* mod *p*
5. Đặt *s* = *s2* + *q·h*
6. Xuất ra *s.*

* Phép toán cơ sở RSAVP

**RSAVP** ((n, e), s)

|  |  |  |
| --- | --- | --- |
| Đầu vào | *(n, e)* | Khóa công khai RSA |
| *s* | Biểu diễn của chữ ký, là số nguyên giữa 0 và *n*-1 |
| Đầu ra | m | Biểu diễn của thông điệp, là số nguyên giữa 0 và *n*-1 |
| Thông báo lỗi | “biểu diễn chữ ký ở ngoài miền hợp lệ” | |
| Giả thiết | Khóa công khai RSA (*n, e*) là hợp lệ | |

Các bước:

* 1. Nếu biểu diễn của chữ ký s không nằm giữa 0 và *n*-1, cho ra “biểu diễn chữ ký ở ngoài miền hợp lệ” và dừng lại;
  2. Đặt *m* = *se* mod *n*;
  3. Xuất ra *m.*

#### Thao tác tạo chữ ký

**RSASSA-PSS-SIGN*(K, M)***

Trong đó:

|  |  |  |
| --- | --- | --- |
| Đầu vào | *K* | khóa bí mật RSA của người ký |
| *M* | thông điệp sẽ được ký, là một chuỗi octet |
| Đầu ra | *S* | chữ ký, chuỗi octet có độ dài *k*, với*k*là độ dài của modulo RSA theo octet |
| Thông báo lỗi | “văn bản quá dài”, “lỗi định dạng” | |

Các bước:

* 1. M*ã hóa EMSA-PSS*: Áp dụng thao tác EMSA-PSS-ENCODE (được giới thiệu ở phần sau) vào văn bản *M* để tạo ra thông điệp được định dạng *EM* có độ dài ⎡(*modBits*-1)/8⎤ octet sao cho độ dài bit của số nguyên OS2IP *(EM)* nhiều nhất là *modBits*-1, với *modBits* là độ dài theo bit của số n (mođun RSA):

*EM* = EMSA-PSS-ENCODE *(M*, *modBits-1)*

Chú ý rằng độ dài octet của EM sẽ bằng k - 1 nếu *modBits-1* chia hết cho 8 và bằng k nếu *modBits-1* không chia hết cho 8. Nếu hàm EMSA-PSS-ENCODE cho ra thông báo lỗi “văn bản quá dài” thì RSASSA-PSS-SIGN cũng cho ra thông báo lỗi “văn bản quá dài” và dừng lại. Nếu EMSA-PSS-ENCODE cho ra thông báo “lỗi định dạng” thì RSASSA-PSS-SIGN cho ra thông báo “lỗi định dạng” và dừng lại.

1. *Chữ ký RSA*:
2. Chuyển thông điệp đã được định dạng (chuỗi octet) *EM* thành biểu diễn thông điệp ở dạng số nguyên *m*;

*m = OS2IP (EM)*

1. Áp dụng phép toán cơ sở RSASP với K là khóa bí mật RSA và biểu diễn thông điệp *m* để tạo ra biểu diễn chữ ký là số nguyên *s*:

*s = RSASP(K, m);*

1. Chuyển chữ ký s dạng số nguyên thành chữ ký S dạng chuỗi octet có độ dài k:

S = I2OSP (*s, k*)

1. X*uất ra chữ ký S*.

#### Thao tác kiểm tra chữ ký

**RSASSA-PSS-VERIFY***((n, e), M, S)*

|  |  |  |
| --- | --- | --- |
| Đầu vào | *(n, e)* | khóa công khai RSA của người ký |
| *M* | thông điệp mà chữ ký của nó cần được kiểm tra, là chuỗi octet |
| *S* | chữ ký được kiểm tra, chuỗi octet có độ dài *k*, với *k*là độ dài theo octet của số *n*, modulo RSA |
| Đầu ra | “chữ ký hợp lệ” hoặc “chữ ký không hợp lệ” | |

Các bước:

1. *Kiểm tra độ dài*: Nếu độ dài của chữ ký *S* không là *k* octet, cho ra thông báo lỗi “chữ ký không hợp lệ” và dừng;
2. *Kiểm tra chữ ký RSA*:
3. Chuyển chữ ký S thành biểu diễn chữ ký ở dạng số nguyên s;

*s =* OS2IP *(S)*

1. Áp dụng phép toán cơ sở RSAVP với khóa công khai RSA là (*n, e*) và biểu diễn chữ ký s để tạo ra m là số nguyên biểu diễn thông điệp;

*m* = RSAVP *((n, e), s)*

1. Chuyển biểu diễn thông điệp m thành thông điệp đã được định dạng *EM* có độ dài *emLen* =⎡(*modBits*-1/8⎤ octet, trong đó *modBits* là độ dài theo bit của số n (mođun RSA):

*EM = I2OSP (m,* *emLen)*

Chú ý rằng *emLen* sẽ bằng *k-1* nếu *modBits-1* chia hết cho 8 và bằng k nếu *modBits-1* không chia hết cho 8. Nếu I2OSP cho ra thông báo lỗi “số nguyên quá lớn” thì RSASSA-PSS-VERIFY cho ra thông báo lỗi “chữ ký không hợp lệ” và dừng lại.

1. *Kiểm tra EMSA-PSS:* Áp dụng thao tác kiểm tra EMSA-PSS-VERIFY vào thông điệp M và thông điệp đã được định dạng EM để xác định xem chúng có tương ứng với nhau hay không;

*Result* = EMSA-PSS-VERIFY (*M, EM,* *modBits-1)*

1. Nếu kết quả (Result) là “phù hợp” thì cho ra “chữ ký hợp lệ”. Ngược lại sẽ cho ra “chữ ký không hợp lệ”.

### Hàm băm SHA-256

SHA-256 được sử dụng để tính giá trị băm của một thông điệp có độ dài là *l*, với *0 ≤ l < 264*. Thuật toán sử dụng một chuỗi 64 từ (32 bit) được tạo ra từ một khối thông điệp đầu vào, 8 biến làm việc cho mỗi từ 32 bit, giá trị băm trung gian gồm 8 từ (32 bit), kết quả cuối cùng của hàm băm SHA-256 là 256 bit mã băm hay còn gọi là bản tóm lược thông điệp.

Các từ tạo tạo ra từ khối thông điệp đầu vào được ký hiệu là *W0,W1,…,W63,* tám biến làm việc được ký hiệu là *a, b, c, d, e, f, g* và *h*. Các từ của kết quả băm được ký hiệu là , chúng được gán các giá trị băm khởi đầu, *H(0)*, và sẽ được thay thế bởi các giá trị băm trung gian (sau khi mỗi khối thông điệp được xử lý), *H(i)*, và cuối cùng là giá trị băm, *H(N)*.

Các bước:

1. *Tiền xử lý:* Tiền xử lý được thực hiện trước khi bắt đầu tính toán giá trị băm. Bước tiền xử lý được chia làm 3 bước nhỏ: bổ sung thông điệp, chia thông điệp đã được bổ sung thành các khối, và thiết lập các giá trị băm khởi đầu H(0).
2. *Bổ sung thông điệp:* Giả sử thông điệp M có độ dài là *l* bit, bổ sung bit “1” vào cuối thông điệp, tiếp theo là k bit 0, với k thỏa mãn *l*+1+k = 448 mod 512. Cuối cùng bổ sung một khối 64 bit để lưu giá trị *l* (độ dài thật của thông điệp). Như vậy thông điệp sau khi đã được bổ sung có độ dài là bội của 512 bit.
3. *Chia thông điệp thành khối sau khi đã được bổ sung:* Thông điệp sau khi đã được bổ sung được chia thành N khối 512 bit, *M(1), M(2),…, M(N)*. Mỗi khối thông điệp gồm 16 từ (32 bit). 32 bit đầu tiên của khối thông điệp thứ *i* là , 32 bit tiếp theo là  và 32 bit cuối cùng của khối thông điệp thứ *i* là .
4. *Thiết lập các giá trị băm khởi đầu H(0):* Các giá trị băm khởi đầu gồm 8 từ (32 bit):
5. *Tính toán giá trị băm SHA-256:* Việc tính toán giá trị băm SHA-256 sử dụng các hàm và hằng được định nghĩa trong phần 6.2.1. Phép tính “+” được xem là cộng môđun 232. Sau khi hoàn thành bước tiền xử lý, mỗi khối thông điệp, M(1), M(2),…, M(N) tuần tự được xử lý theo các bước dưới đây với i từ 0 tới n:
6. Tính các từ Wt từ khối thông điệp:

|  |  |
| --- | --- |
| Wt = C:\Users\trany\AppData\Local\Microsoft\Windows\Clipboard\HistoryData\{3FD28760-A31A-41B2-9E9D-D8F4F63EE413}\{680B33C1-B334-488E-B130-0025FE64C45C}\ResourceMap\{AC7429C2-841F-4FED-9312-ABD0FAB4DE85} | *với 0 ≤ t ≤ 15* |
| C:\Users\trany\AppData\Local\Microsoft\Windows\Clipboard\HistoryData\{3FD28760-A31A-41B2-9E9D-D8F4F63EE413}\{680B33C1-B334-488E-B130-0025FE64C45C}\ResourceMap\{08D2DEC7-1904-4421-B366-8ADF6D3AD610} | *với 16 ≤ t ≤ 63* |

1. Khởi gán tám biến làm việc a, b, c, d, e, f, g và h bằng các giá trị băm thứ (i - 1)

*; ; ; ;*

*; ; ;*

1. Với t từ 0 đến 63, tính:

C:\Users\trany\AppData\Local\Microsoft\Windows\Clipboard\HistoryData\{3FD28760-A31A-41B2-9E9D-D8F4F63EE413}\{A07E98A8-AFD3-47B9-8045-D0918AF604DB}\ResourceMap\{A0C7AAD6-6E34-47F6-AA84-CFC535E47FF0};

C:\Users\trany\AppData\Local\Microsoft\Windows\Clipboard\HistoryData\{3FD28760-A31A-41B2-9E9D-D8F4F63EE413}\{A07E98A8-AFD3-47B9-8045-D0918AF604DB}\ResourceMap\{EE6A8E75-E180-4A1E-AADE-A2D19A5CF890};

1. Tính giá trị băm trung gian thứ i, H(i)

; ; ; ;

; ; ; 

Sau khi xử lý N lần (tương ứng với N khối thông điệp), kết quả đầu ra hàm băm SHA-256 của thông điệp M là:

||||||||||||||

### Bộ tạo số giả ngẫu nhiên dùng AES-128

Thuật toán tạo số giả ngẫu nhiên dùng để tạo ra các số giả ngẫu nhiên (độc lập, đồng xác suất) làm tham số trong lược đồ chữ ký số.

#### Một số ký hiệu

|  |  |
| --- | --- |
| AESK(M) | Hàm mã hóa AES-128. Thực hiện việc mã hóa khối thông điệp M (128 bit) bởi khóa K (128 bit). Trả về 128 bít dữ liệu đã mã của M (Chi tiết về hàm AES-128 được nêu trong tài liệu FIPS 197 “Advanced Encryption Standard”) |
| DTj | Giá trị 128 bit, là ngày tháng/thời gian (date/time) của hệ thống |
| XOR | Phép toán XOR bit |
| ⎡x⎤ | Số nguyên bé nhất lớn hơn hay bằng x, ví dụ: ⎡6⎤ = ⎡5.1⎤ = 6 |
| ← | Phép gán giá trị; ví dụ: a ← b có nghĩa là gán b cho a |

#### Thuật toán

Đầu vào:

|  |  |
| --- | --- |
| L | Số bit cần tạo ngẫu nhiên |
| V0 | 128 bit ngẫu nhiên, lựa chọn bởi người dùng |
| DTj | 128 bit là ngày tháng/thời gian của hệ thống |
| K | 128 bit khóa cho AES-128 |

Đầu ra: Số giả ngẫu nhiên *p* có *L* bit

Các bước:

1. p = null
2. Với j từ 1 đến ⎡L/128⎤, thực hiện các bước sau:
   1. Ij = AESK(DTj)
   2. xj = AESK(Ij XOR Vj-1)
   3. Vj = AESK(Ij XOR xj)
   4. p ← p || xj
3. p ← Lấy L bit bên trái của p

### Lý do cho việc sử dụng 3 thuật toán trong tiêu chuẩn

Nguyên nhân khiến Tiểu Ban kỹ thuật TCVN/JTC 1/SC 27 lựa chọn 3 thuật toán trong tiêu chuẩn:

* Về lựa chọn thuật toán ký RSA-PSS. Trên thế giới các thuật toán chữ ký phổ biến nhất được sử dụng cho đến nay là RSA, DSA và thuật toán dựa trên đường cong Elliptic. Tuy nhiên, thuật toán ký DSA thời điểm xây dựng tiêu chuẩn chưa được chứng minh là có tính an toàn chứng minh được (provable secure). Ngoài ra, vào năm 2001, một lỗi quan trọng trong cách cài đặt của thuật toán này đã được đề cập trong bài báo của Bleichenbacher. Thuật toán ký dùng đường cong Eliptic (ECDSA) được nhiều nước khuyến cáo sử dụng, tuy nhiên, ECDSA chưa thể có được trải nghiệm thực tế nhiều bằng hệ chữ ký RSA. Thuật toán ký RSA-PSS có các ưu thế sau: Về lí thuyết, RSA-PSS thuộc lược đồ an toàn chứng minh được. Về độ an toàn thực tế: Do được ngẫu nhiên hóa cao nên phần lớn các tấn công vào lược đồ RSA-PSS khó thành công. RSA-PSS được nhiều tổ chức nghiên cứu ban hành tiêu chuẩn mật mã đánh giá cao.
* Về lựa chọn thuật toán hàm băm SHA-256: Các thuật toán hàm băm phổ biến nhất được dùng đến nay là các hàm băm thuộc họ MDx và SHA-x. Phần lớn các hàm băm thuộc họ MDx đều bị phá. Đối với hàm băm thuộc họ SHA-x, trước đây, trong các tiêu chuẩn mật mã thường sử dụng thuật toán hàm băm SHA-1 (có độ dài của giá trị băm là 160 bit) và được coi là an toàn. Nhưng tháng 3/2005, nhóm 3 nhà mật mã học người Trung Quốc đã thông báo phá được thuật toán này. Chính vì vậy mà trong Tiêu chuẩn CKS Việt Nam đã phân tích lựa chọn thuật toán SHA-256.

(Tạp chí An toàn thông tin, 2007)

## So sánh chữ ký số RSA trong lý thuyết và tiêu chuẩn

|  |  |  |
| --- | --- | --- |
|  | **Chữ ký số RSA** | **TCVN 7635:2007** |
| **Thao tác ký** | 3 bước:   * Băm thông điệp * Sử dụng khóa bí mật d, tính chữ ký * Xuất ra chữ ký S | 3 bước:   * Mã hóa EMSA-PSS * Sử dụng khóa bí mật d, tính chữ ký S * Xuất ra chữ ký S |
| **Thao tác kiểm tra chữ ký** | 3 bước:   * Sử dụng khóa công khai của người gửi e, tính giá trị * So sánh kết quả băm thông điệp M với giá trị m , đưa ra kết quả của * Nếu thì chữ ký hợp lệ. Ngược lại sẽ cho ra “chữ ký không hợp lệ”. | 3 bước:   * Sử dụng khóa công khai, tính giá trị m và chuyển về dạng EM: *EM = I2OSP (m,* *emLen)* * Kiểm tra EMSA-PSS: Từ EM xuất ra H, từ thông điệp M tiến hành băm , tính rồi băm , sau đó so sánh H với H' để đưa ra kết quả kiểm tra. * Nếu kết quả kiểm tra EMSA-PSS là “phù hợp” thì cho ra “chữ ký hợp lệ”. Ngược lại sẽ cho ra “chữ ký không hợp lệ”. |

Bảng 2.3 So sánh chữ ký số RSA trong lý thuyết và tiêu chuẩn

Vì lý do bảo mật, chữ ký số RSA không thể được sử dụng ở dạng “đơn giản”, cần một số loại tiền xử lý cho thông điệp truyền vào. Vai trò của việc mã hóa EMSA-PSS giải thích lý do có sự khác nhau giữa chữ ký số RSA trong lý thuyết và tiêu chuẩn:

* RSA-PSS băm thông điệp đầu vào và ghép với một muối - *salt* (một số ngẫu nhiên) và phần đệm – *padding* sau đó chạy hàm băm. Hàm băm H này được sử dụng cho đầu ra của việc mã hóa thông điệp. Sau đó, một mặt nạ của H được tính với độ dài bằng mô đun RSA trừ đi độ dài của H. Mặt nạ này sau đó được XOR với muối (và một số phần đệm bằng 0) và đầu ra sẽ được gọi là maskedDB. Sau đó, maskedDB được thêm vào H cùng với giá trị *0xbc* để tạo ra đầu vào cho hàm ký RSA.
* Tất cả các lược đồ chữ ký số được tiêu chuẩn hóa dựa trên RSA đều liên quan đến việc nối thêm một hoặc nhiều hằng số (ví dụ: padding1 và padding2) trong quá trình hình thành bản thông điệp tóm lược (message digest). Mục tiêu là làm cho đối thủ khó tìm thấy một thông điệp khác ánh xạ tới cùng một thông điệp tóm lược hoặc tìm hai thông điệp ánh xạ tới cùng một thông điệp tóm lược.
* RSA-PSS cũng kết hợp một số giả ngẫu nhiên, cụ thể là muối (salt). Vì muối thay đổi sau mỗi lần sử dụng, nên việc ký hai lần vào cùng một thông điệp bằng cùng một khóa riêng tư sẽ mang lại hai chữ ký khác nhau.

Thuật toán RSA-PSS đã được chứng minh tính an toàn khi tránh được cuộc tấn công Bleichenbacher *(Bleichenbacher Attack)*, tấn công dựa trên lỗi tính toán *(Fault-based Attack)*, … RSA Laboratories đề xuất là lược đồ an toàn nhất trong số các lược đồ RSA hiện nay. (Stallings, 2013)

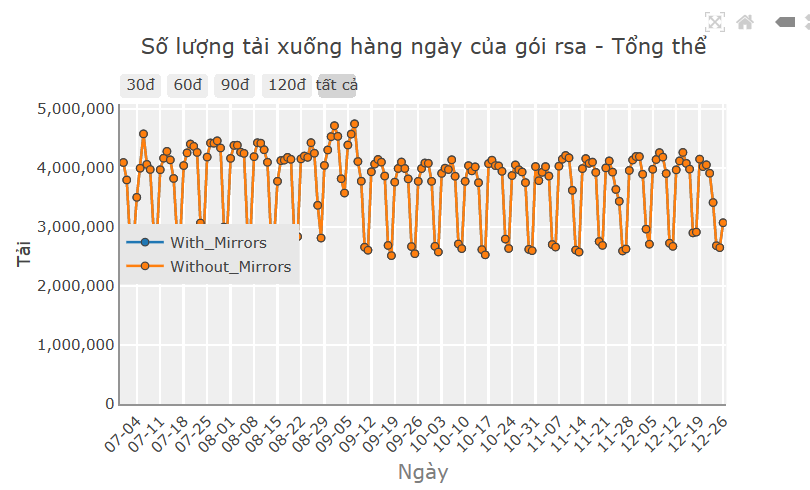
# TRIỂN KHAI CÀI ĐẶT THỬ NGHIỆM

## Thiết lập môi trường và kịch bản thử nghiệm

### Thiết lập môi trường

Khi triển khai thử nghiệm chữ ký điện tử bằng thuật toán RSA nhóm lựa chọn:

* Thiết lập và cài đặt trên máy tính cá nhân với hệ điều hành Window 10
* Ngôn ngữ lập trình chương trình: Python 3.11.0
* Trình soạn thảo mã nguồn: Visual Studio Code
* Thư viện sử dụng:

1. *Rsa* : Thư viện dùng để thực hiện các chức năng như tạo khóa, tạo chữ ký và xác minh chữ ký số theo tiêu chuẩn mật mã công khai PKCS (Public-Key Cryptography Standard). Ban đầu được viết bởi Sybren Stüvel, Marloes de Boer và Ivo Tamboer. Nó được cải thiện bởi Sybren Stüvel, Barry Mead, Yesudeep Mangalapilly, và những người khác. Thư viện này đã được cấp phép Apache phiên bản 2.0. (Stüvel, 2020)   
   Theo trang PyPi Stats, tính đến ngày 27/12/2022 số lượt tải của tháng trước lên đến 108,678,365 lượt tải, đặc biệt số lượt tải xuống vào ngày 26/12/2022 là 3.070.271 lượt tải. (PyPI Stats, 2022)   
   

Hình 3.1 Số lượng tải xuống hàng ngày của gói RSA – Tổng thể

1. *Tkinter*: Thư viện dùng để xây dựng giao diện chương trình có hỗ trợ hầu hết các nền tảng (Windows, Mac, Unix). Tkinter là thư viện GUI tiêu chuẩn cho Python. Tkinter trong Python cung cấp một cách nhanh chóng và dễ dàng để tạo các ứng dụng GUI. Tkinter cung cấp giao diện hướng đối tượng cho bộ công cụ Tk GUI. (Python Software Foundation, n.d.)
2. *Pyinstaller*: Thư viện để đóng gói chương trình được viết bởi David Cortesi, dựa trên cấu trúc của Giovanni Bajo & William Caban, dựa trên hướng dẫn sử dụng của Gordon McMillan. PyInstaller hỗ trợ Python 3.7 trở lên và đóng gói chính xác nhiều gói Python chính chẳng hạn như numpy, matplotlib, PyQt, wxPython… (Cortesi, n.d.)

### Kịch bản thử nghiệm

Với các thiết lập trên nhóm tiến hành xây dựng kịch bản thử nghiệm như sau:

***Thao tác tạo khóa***

Tạo khóa công khai và bí mật với kích thước khóa là 2048 và được lưu dưới dạng nhị phân với phần mở rộng là .pem: PrivateKey.pem, PublicKey.pem

***Thao tác tạo chữ ký số***

Tạo chữ ký số và lưu chữ ký số dưới dạng nhị phân với phần mở rộng là .sig. Trong đó:

* Thông điệp đầu vào: Chọn 1 tệp bất kỳ có định dạng .docx/.txt/.mp4/.jpg/.xlsx
* Hàm băm: Tích chọn hàm băm SHA-256 hoặc SHA-512
* Khóa bí mật: Chọn khóa bí mật được tạo và lưu ở thao tác tạo khóa

***Thao tác xác thực chữ ký số***

* Thông điệp đầu vào: Chọn tệp thông điệp đầu vào là tệp cần được kiểm tra xem hợp lệ không
* Chọn hàm băm: Chọn hàm băm SHA-256 hoặc SHA-512
* Chữ ký số: Chọn tệp chữ ký số được lưu ở bước tạo chữ ký số
* Chọn khóa công khai: Chọn tệp khóa công khai được tạo và lưu ở thao tác tạo khóa.

Dữ liệu thử nghiệm:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Tạo chữ ký số | Xác minh chữ ký số ***(Chữ ký hợp lệ)*** | Xác minh chữ ký số ***(Chữ ký không hợp lệ)*** |
| Khóa | PrivateKey.pem | PublicKey.pem | PublicKey.pem |
| Thông điệp đầu vào | Promise.docx | Promise.docx | Promise-edit.docx |
| tkb3.xlsx | tkb3.xlsx | tkb3-edit.xlsx |
| test.mp4 | test.mp4 | Test-edit.mp4 |
| utonabi.jpg | utonabi.jpg | Utonabi-edit.jpg |
| Hàm băm | SHA-256 | SHA-256 | SHA-256 |
| Chữ ký số |  | test\_sign1.sig | test\_sign1.sig |

Bảng 3.1 Dữ liệu thử nghiệm

## Cài đặt thử nghiệm

URL của mã nguồn: https://github.com/minhnguyet531/RSA\_Project

### Mô tả các tệp

|  |  |
| --- | --- |
| Common.py | Lưu các hàm dùng chung cho chương trình |
| Design.py | Lưu các thiết kế form chương trình |
| GenerateKeys.py | Lưu hàm tạo public\_key, private\_key và các hàm liên quan đến khóa công khai/ khóa bí mật |
| hashMess.py | Lưu hàm dùng để băm thông điệp đầu vào |
| paths.py | Lưu các đường dẫn chính của thư mục |
| sign.py | Lưu hàm tạo chữ ký số |
| verify.py | Lưu hàm xác minh chữ ký số |

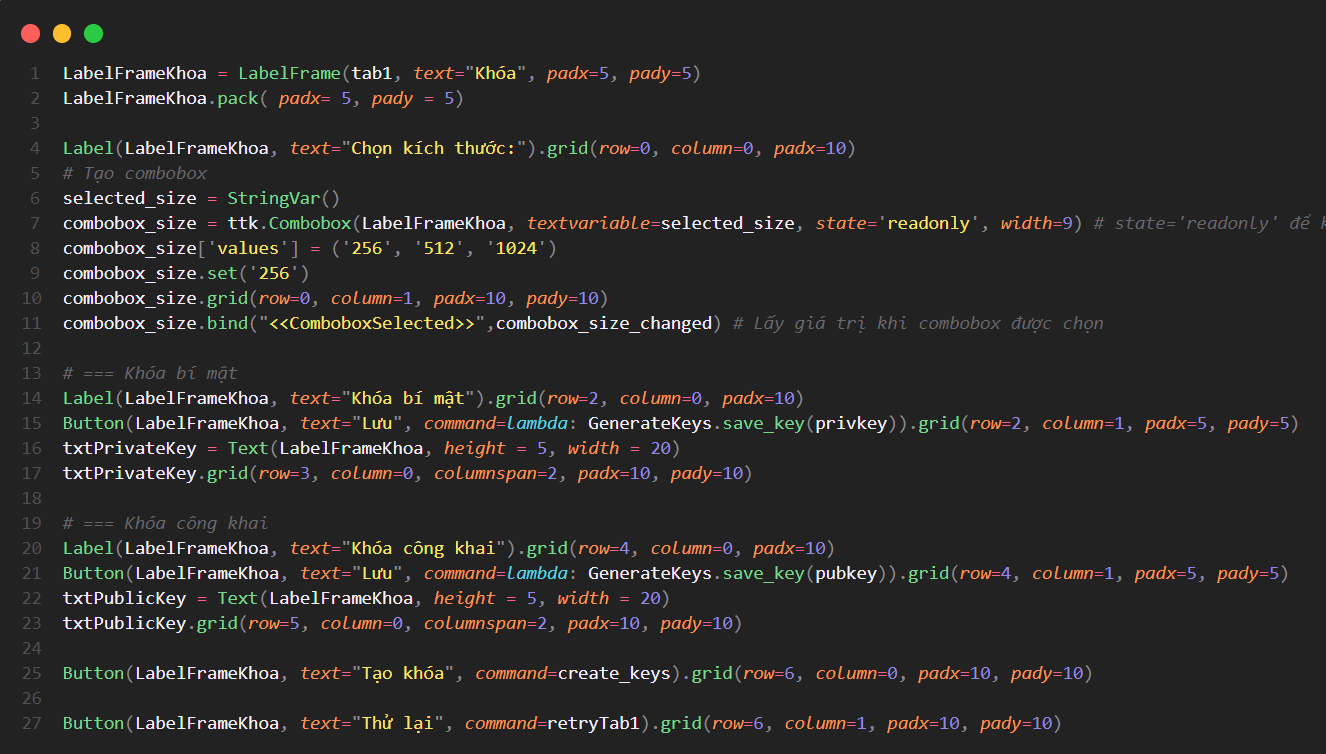
Bảng 3.2 Mô tả các tệp để xây dựng chương trình

### Các hàm chính được sử dụng

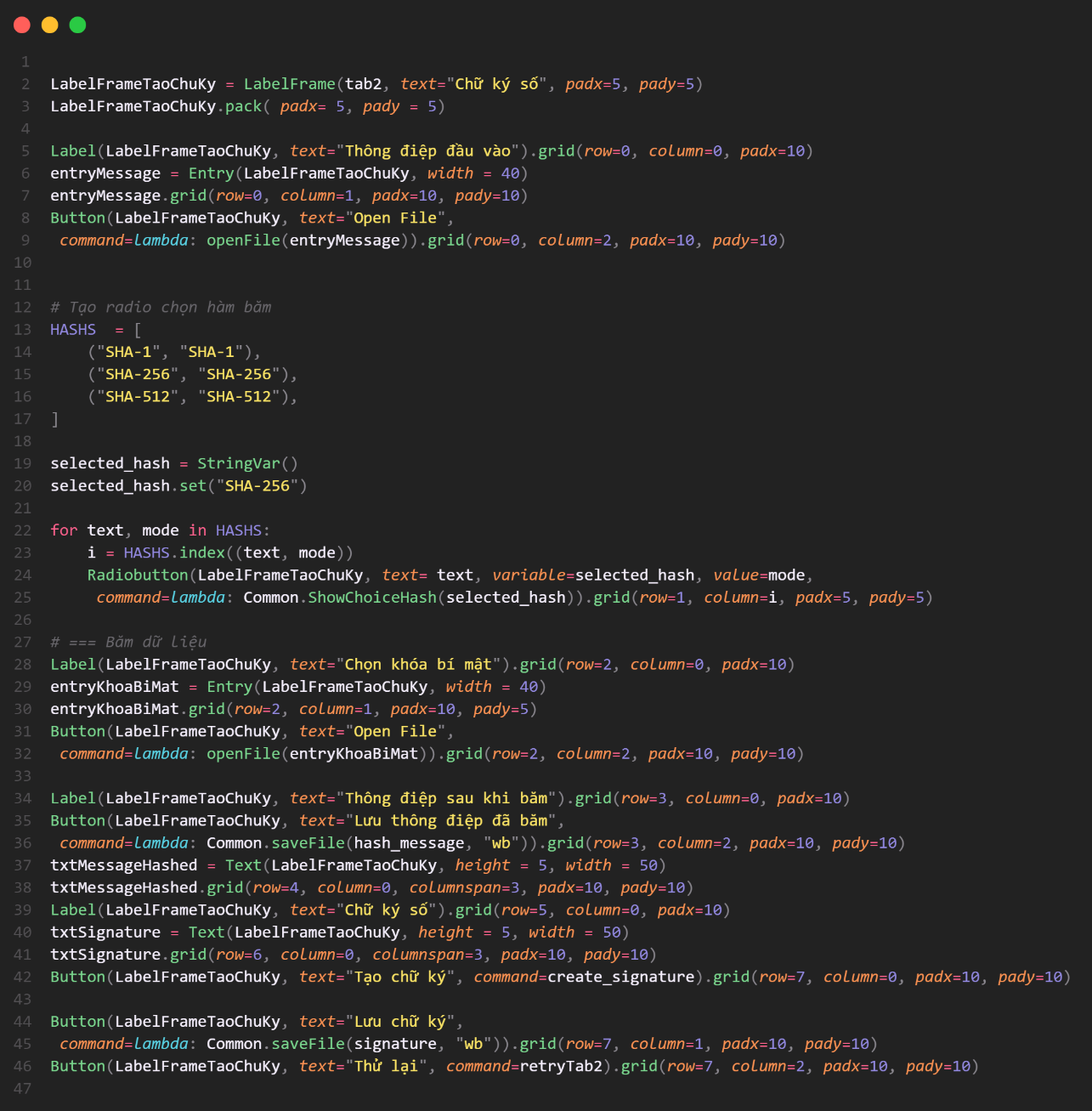
* (pubkey, privkey) = rsa.newkeys(size) => Tạo khóa công khai và khóa bí mật dựa vào thư viện RSA
* rsa.PublicKey.load\_pkcs1() => Load khóa công khai
* rsa.PrivateKey.load\_pkcs1() => Load khóa bí mật
* rsa.compute\_hash(input\_mess, hashType) => Hàm băm
* rsa.sign\_hash(hash\_msg, privkey, hash\_method) => Tạo chữ ký
* rsa.verify(message, signature, pubkey) => Xác minh chữ ký

### Xây dựng giao diện chương trình

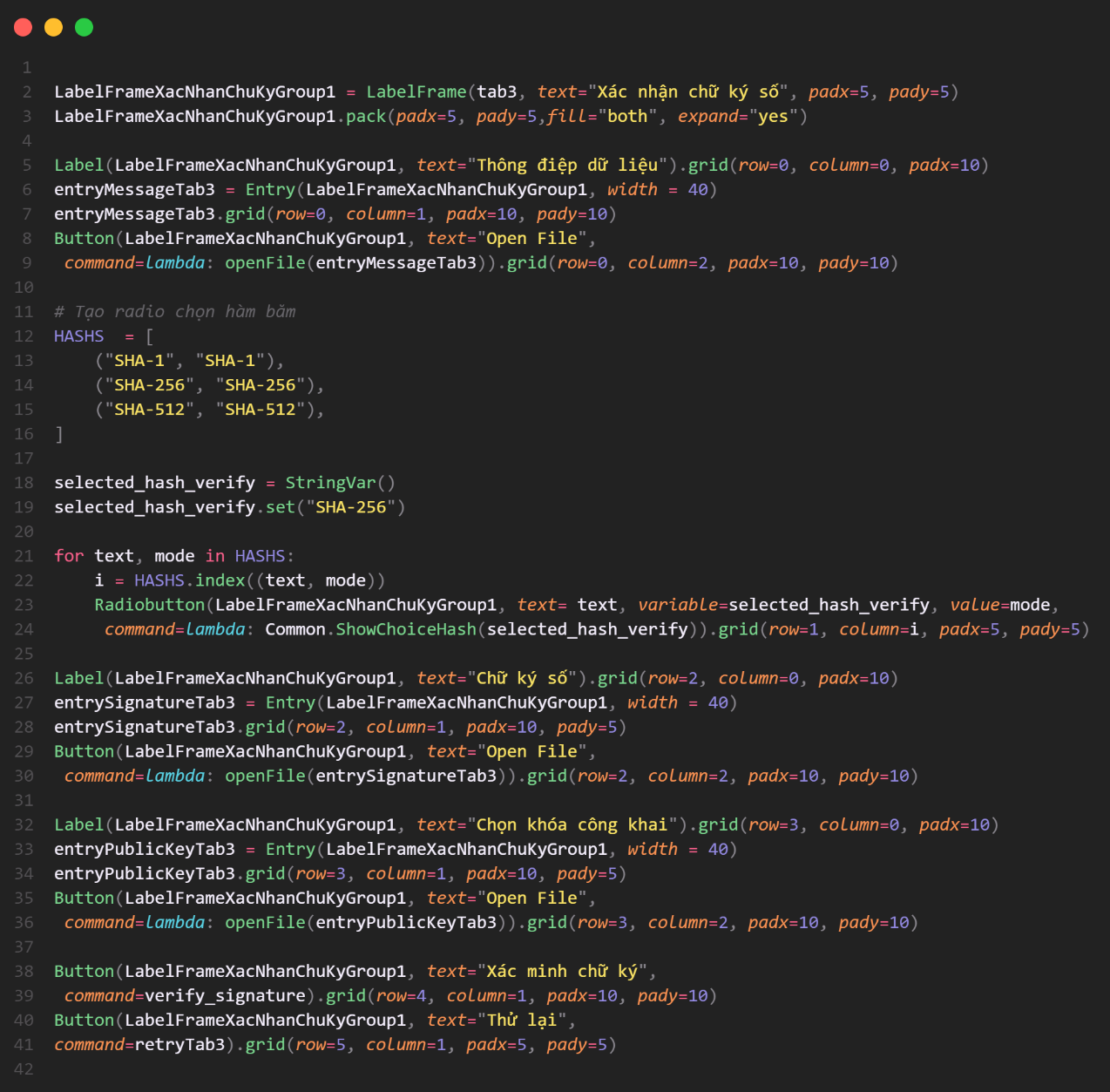
Nhóm đã xây dựng giao diện chương trình bằng đoạn mã dưới đây.

****

Hình 3.2 Code giao diện form tạo khóa

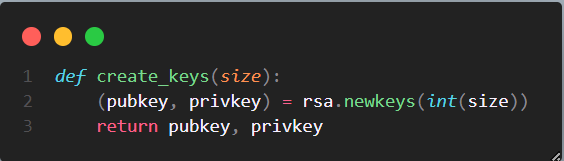
****

Hình 3.3 Code giao diện form tạo chữ ký số

****

Hình 3.4 Code giao diện form xác minh chữ ký số

### Khởi tạo khóa

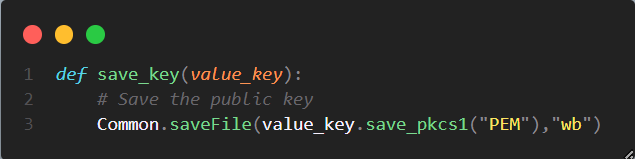


Hình 3.5 Code khởi tạo khóa công khai và khóa bí mật

Trong đó:

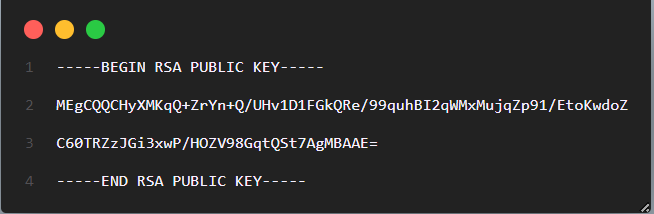
* size: Kích thước của khóa cần khởi tạo (2048/…)
* pubkey: Khóa công khai
* privkey: Khóa bí mật
* rsa.newkeys(): hàm tạo khóa công khai và khóa riêng tư với tham số đầu vào là kích thước khóa n, đồng thời trả về dưới dạng (rsa.PublicKey, rsa.PrivateKey)

Cả khóa pubkey và privkey đều được lưu dưới dạng PEM theo tiêu chuẩn PKCS#1.



Hình 3.6 Code lưu khóa được tạo

Sau khi lưu khóa thành công thì khóa có dạng như sau:

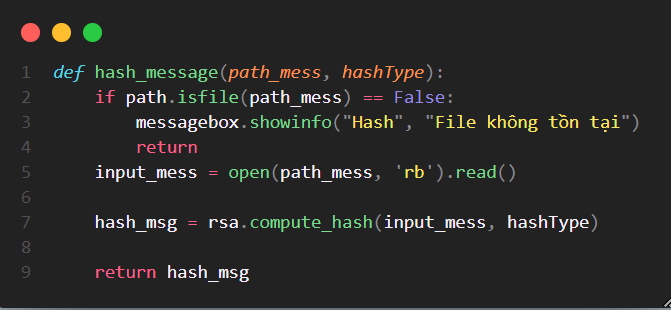


Hình 3.7 Lưu khóa công khai với định dạng .pem



Hình 3.8 Lưu khóa bí mật với định dạng .pem

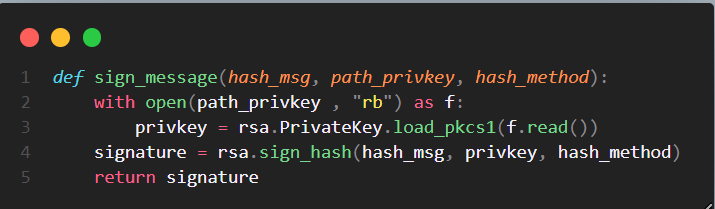
### Tạo chữ ký



Hình 3.9 Code băm thông điệp đầu vào

Trong đó:

* Hàm hash\_message: thực hiện băm thông điệp đầu vào và trả về kết quả đã được băm của thông điệp
* path\_mess: Đường dẫn lưu thư mục của tệp thông điệp đầu vào
* hashType: Loại hàm băm sử dụng (SHA256, SHA512)
* input\_mess: Biến lưu thông điệp đầu vào cần được băm
* rsa.compute\_hash(): Hàm để băm thông điệp của thư viện rsa, đầu ra là kết quả thông điệp đã được băm



Hình 3.10 Code tạo chữ ký số

Trong đó:

* sign\_message: Hàm tạo chữ ký số và trả về chữ ký số sau khi được tạo
* hash\_msg: Thông điệp đã được băm
* path\_privkey: Đường dẫn lưu thư mục của tệp chứa khóa bí mật
* hash\_method: Phương thức băm (SHA-256, SHA-512…)
* rsa.PrivateKey.load\_pkcs1(): Load khóa bí mật từ tệp lưu khóa bí mật
* rsa.sign\_hash: Hàm để tạo chữ ký số của thư viện rsa với thông điệp đã băm làm thông số đầu vào sau đó ký mã băm bằng khóa riêng tư theo phương thức băm tương ứng
* signature: Chữ ký số được tạo

### Xác thực chữ ký



Hình 3.11 Code xác thực chữ ký số

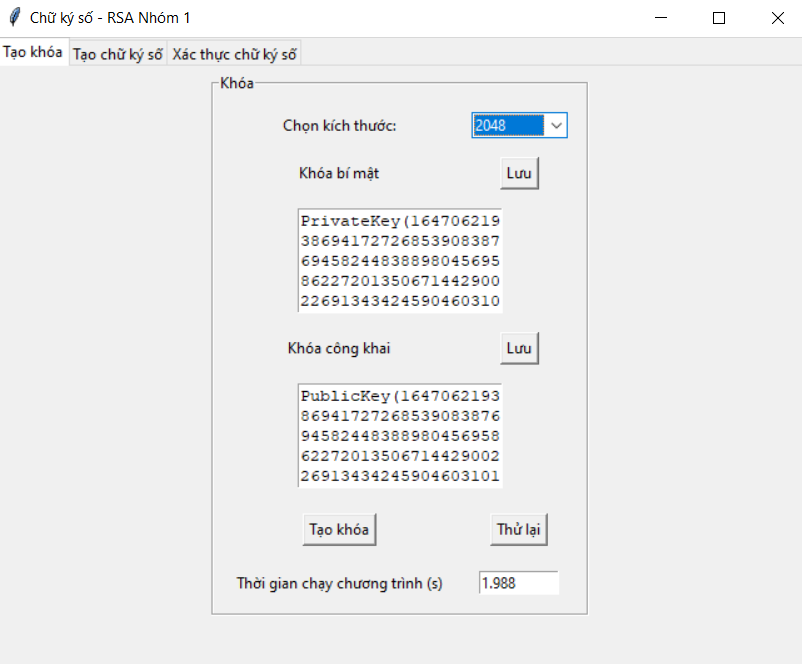
Một số biến và hàm sử dụng:

* verify\_signature: Hàm xác thực chữ ký số và trả về kết quả xác thực True nếu hợp lệ và False nếu không hợp lệ
* path\_message: Đường dẫn đến thư mục chứa tệp thông điệp đầu vào
* path\_signature: Đường dẫn đến thư mục chứa tệp chữ ký số
* path\_pubkey: Đường dẫn đến thư mục chứa tệp khóa công khai
* rsa.verify: Xác minh chữ ký số của thư viện rsa và trả về tên của hàm băm đã sử dụng (SHA-256/SHA512).

### Giao diện chương trình

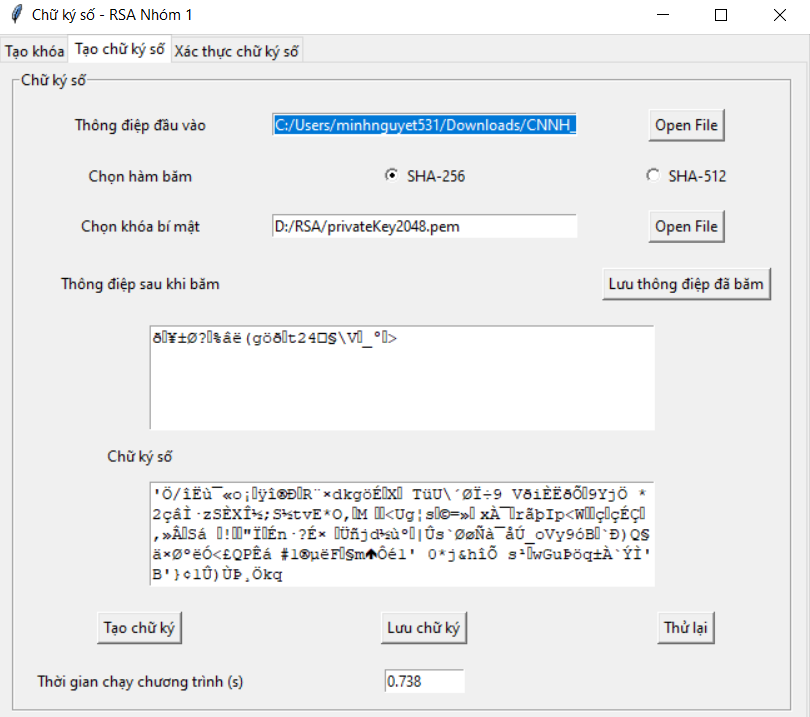
Chương trình gồm 3 giao diện:

* Tạo khóa,
* Tạo chữ ký số
* Xác thực chữ ký số.



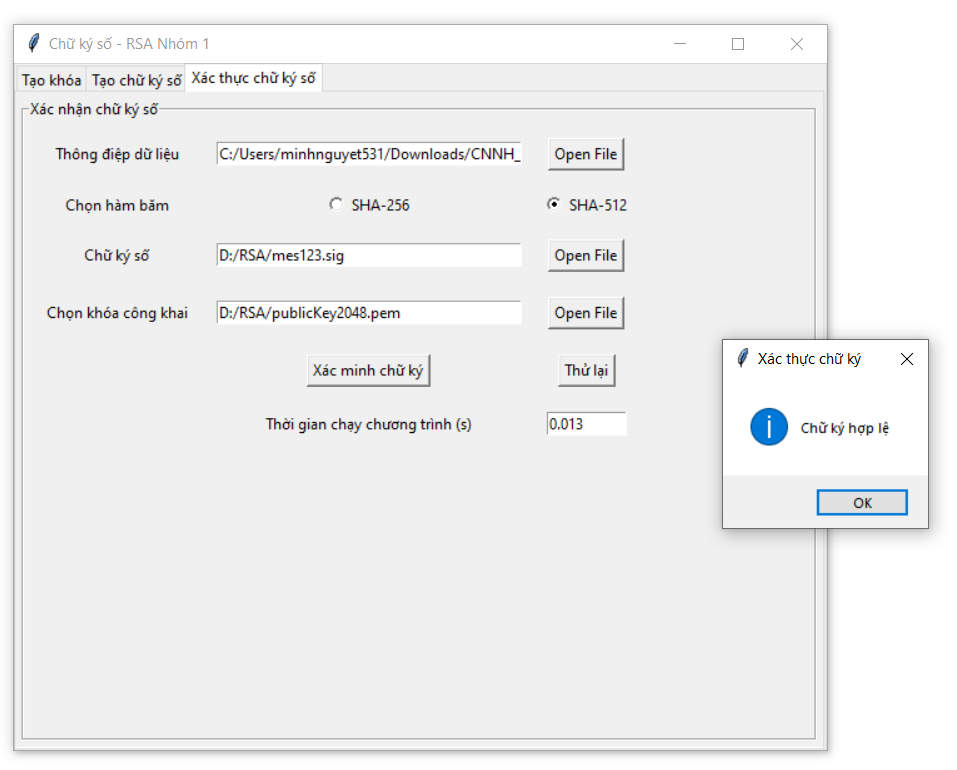
Hình 3.12 Giao diện tạo khóa

Các chức năng cơ bản trên giao diện bao gồm chức năng tạo khóa được tạo tương ứng với kích thước đầu vào được chọn tại combobox, chức năng lưu khóa và chức năng thử lại thao tác tạo khóa.

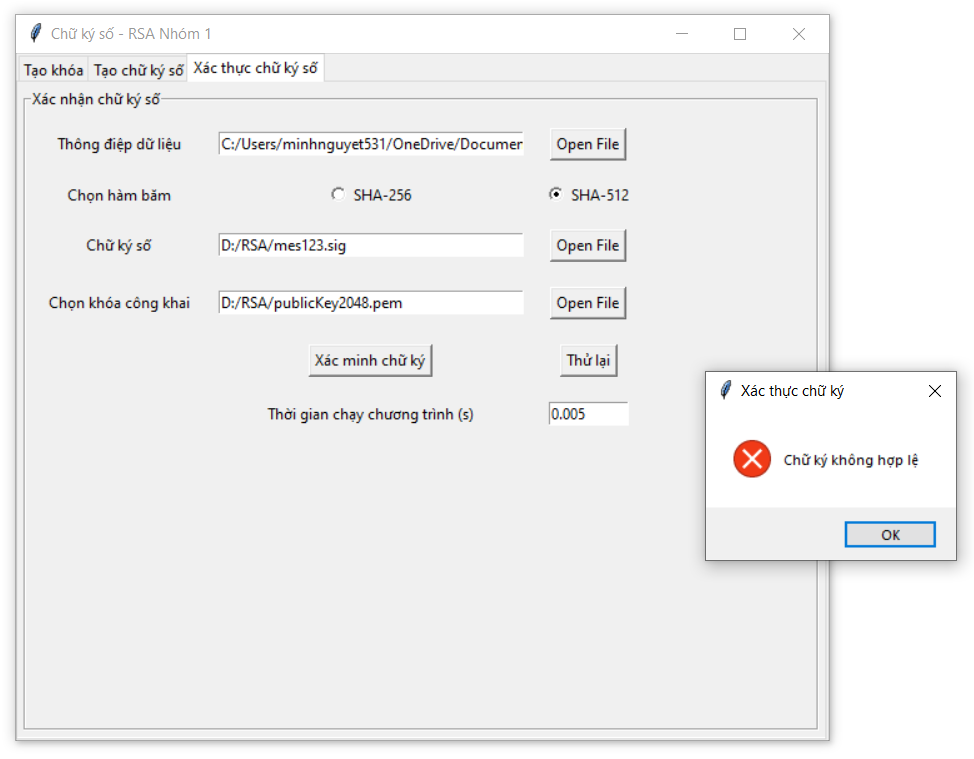


Hình 3.13 Giao diện tạo chữ ký số

Tại giao diện tạo chữ ký số bao gồm các chức năng như tạo chữ ký số với thông điệp đầu vào, khóa bí mật được chọn từ file và phương thức băm được chọn từ các nút radio button. Đồng thời có các chức năng như lưu chữ ký số, thông điệp đã băm và thử tại thao tác tạo chữ ký số.



Hình 3.14 Giao diện xác minh chữ ký số khi chữ ký hợp lệ

******

Hình 3.15 Giao diện xác minh chữ ký số khi chữ ký không hợp lệ

Tại giao diện xác minh chữ ký số thì chức năng xác minh chữ ký số được xử lý với các thông số đầu vào như thông điệp dữ liệu, phương thức băm, chữ ký số và khóa công khai được lựa chọn tương ứng. Nếu chữ ký hợp lệ sẽ hiển thị popup thông báo “Chữ ký hợp lệ” ngược lại nếu chữ ký không hợp lệ sẽ hiển thị popup thông báo “Chữ ký không hợp lệ”.

## Kết quả thử nghiệm

Với thư viện rsa nhóm sử dụng đã đem lại kết quả khả quan với việc tạo khóa, tạo chữ ký số cũng như xác thực chữ ký số. Đối với thao tác tạo chữ ký số tùy thuộc vào kích thước lựa chọn thì thời gian thực thi khác nhau nhưng nhìn chung với kích thước nhóm thử nghiệm là 2048 thì thời gian thực thi trung bình rơi vào từ 1.5 giây đến 3 giây. Còn thao tác tạo chữ ký số và xác thực chữ ký số thời gian thực thi khá nhanh đa số thời gian thực thi chưa đến 1 giây. Đồng thời nhóm chưa can thiệp được sâu vào các luồng xử lý logic của thư viện để dễ dàng tùy chỉnh theo như mong muốn

Ngoài việc chạy chương trình dựa trên mã nguồn thì nhóm cũng đã đóng gói thành công chương trình để dễ dàng cài đặt và sử dụng trên các máy tính sử dụng hệ điều hành windown khác.

URL chương trình đã được đóng gói: https://github.com/minhnguyet531/RSA\_Project/blob/main/dist/DesignRSA.exe

KẾT LUẬN

Chữ ký số ngày càng được sử dụng phổ biến trong các giao dịch điện tử, từ dân sự, kinh doanh, tài chính – ngân hàng, bảo hiểm, v.v : ủy quyền thanh toán ngân hàng (chuyển tiền), trao đổi các tài liệu điện tử đã ký, ký các giao dịch trong các hệ thống blockchain công khai (chuyển tiền, mã thông báo hoặc các tài sản kỹ thuật số khác), để ký hợp đồng kỹ thuật số và trong nhiều tình huống khác. Dựa trên kết quả nghiên cứu khoa học công nghệ, các tiến bộ kỹ thuật, cơ sở lựa chọn từ các tiêu chuẩn quốc tế, tiêu chuẩn khu vực và tiêu chuẩn nước ngoài, năm 2007, Bộ Khoa học và Công nghệ quyết định xây dựng Tiêu chuẩn quốc gia Việt Nam về Chữ ký số (ký hiệu TCVN 7635:2007 “Các kỹ thuật mật mã - Chữ ký số”). Tiêu chuẩn đã quy định 3 thành phần cần thiết cho lược đồ chữ ký số bao gồm thuật toán RSA-PSS, hàm băm SHA-256 và bộ tạo số giả ngẫu nhiên dùng giải thuật AES-128. Việc xây dựng tiêu chuẩn chung giúp cho 17 đơn vị cung cấp chữ ký số công cộng cũng như các Bộ, Ban, ngành, cơ quan tổ chức tránh được khó khăn khi phải lựa chọn quá nhiều phương án chữ ký số hiện nay.

Ứng dụng những kiến thức đã học, sử dụng ngôn ngữ lập trình Python 3.11.0 kèm với các thư viện chính được thiết lập là RSA 4.9 và Tkinter, nhóm đã triển khai cài đặt thử nghiệm chữ ký điện tử bằng thuật toán RSA. Người dùng có thể tạo, lưu khóa bí mật/công khai, băm và lưu giá trị băm thông điệp cũng như ký và xác thực chữ ký.

Bài báo cáo đã giúp nhóm hiểu hơn về một trong những cơ sở kỹ thuật mật mã được ứng dụng trong các công nghệ ngân hàng. Do sự hạn chế về thời gian, bài báo cáo vẫn còn một số hạn chế nên rất mong nhận được sự đóng góp và nhận xét từ thầy để bài báo cáo được hoàn thiện hơn.

TÀI LIỆU THAM KHẢO

Cortesi, D., n.d. *PyInstaller Manual.* [Online]   
Available at: https://pyinstaller.org/en/stable/  
[Accessed 2022].

Cục Quản lý mật mã dân sự và Kiểm định sản phẩm mật mã, 2016. *QCVN 5 : 2016/BQP - QUY CHUẨN KỸ THUẬT QUỐC GIA VỀ CHỮ KÝ SỐ SỬ DỤNG TRONG LĨNH VỰC NGÂN HÀNG.* [Online]   
Available at: https://luattrongtay.vn/ViewFullText/Id/569550c7-5f86-4507-adbe-25af0f0550d5

Nhiên, V. T., 2007. *Khảo sát các chữ ký số dựa trên hệ RSA, nghiên cứu lược đồ ký RSA-PSS và những chuẩn hóa.* [Online]   
Available at: https://123docz.net/document/2595973-khao-sat-cac-chu-ki-so-dua-tren-he-rsa-nghien-cuu-luoc-do-chu-ki-rsa-pss-va-nhung-chuan-hoa.htm

PyPI Stats, 2022. *PyPI Download Stats.* [Online]   
Available at: https://pypistats.org/packages/rsa

Python Software Foundation, n.d. *tkinter — Python interface to Tcl/Tk.* [Online]   
Available at: https://docs.python.org/3/library/tkinter.html  
[Accessed 2022].

Roseman, M., n.d. *TkDocs Tutorial.* [Online]   
Available at: https://tkdocs.com/tutorial/index.html  
[Accessed 2022].

Stallings, W., 2013. Digital Signature Algorithms. *Cryptologia.*

Stüvel, S. A., 2020. *Python-RSA.* [Online]   
Available at: https://stuvel.eu/software/rsa/

Stüvel, S. A., 2021. *Welcome to Python-RSA’s documentation!.* [Online]   
Available at: https://stuvel.eu/python-rsa-doc/

Tạp chí An toàn thông tin, 2007. *Tiêu chuẩn Quốc gia Việt Nam về chữ ký số Nguồn antoanthongtin.vn.* [Online]   
Available at: https://antoanthongtin.vn/ca-cqnn/tieu-chuan-quoc-gia-viet-nam-ve-chu-ky-so-100043

Tiểu Ban kỹ thuật tiêu chuẩn TCVN/JTC 1/SC 27 “Các kỹ thuật mật mã”, 2007. *TCVN 7635:2007 “Các kỹ thuật mật mã - Chữ ký số”.* s.l.:s.n.

Trung tâm Chứng thực điện tử quốc gia - Bộ Thông tin và Truyền thông, 2021. *Báo cáo tình hình phát triển và ứng dụng chữ ký số tại Việt Nam năm 2021,* s.l.: s.n.