HỌC VIỆN KỸ THUẬT MẬT MÃ
KHOA CÔNG NGHỆ THÔNG TIN

BÀI TẬP MÔN HỌC CƠ SỞ AN TOÀN BẢO MẬT THÔNG TIN

VIẾT CHƯƠNG TRÌNH QUẢN LÝ DỮ LIỆU AN TOÀN DƯỚI DẠNG MẶT NẠ DỮ LIỆU TRÊN SQLite SỬ DỤNG NGÔN NGỮ RUST

Khoa: Công nghệ thông tin
Chuyên ngành: Kỹ thuật phần mềm nhúng và di động

Người hướng dẫn
TS.Nguyễn Đào Trường
Khoa Công nghệ thông tin - Học viện Kỹ thuật mật mã

Nhóm sinh viên thực hiện:
Phạm Thị Phương Anh CT040401
Phạm Văn Dũng CT040308
Lớp L01

Hà Nội - 2023

LỜI NÓI ĐẦU

Ngày nay với sự phát triển mạnh mẽ của công nghệ thông tin, đặt biệt là sự phát triển của mạng Internet, ngày càng có nhiều dữ liệu lớn được sản sinh và lưu trữ, bảo vệ quyền riêng tư và đảm bảo an toàn thông tin ngày càng trở nên cấp thiết hơn bao giờ hết. Trong thực tế nhiều dữ liệ quan trọng và nhạy cảm được lưu trữ trong các hệ thống máy tính, cơ sở dữ liệu, ứng dụng web và ứng dụng di động, tạo ra mối đe dọa đến quyền riêng tư của người dùng.

Để giải quyết vấn đề này, mặt nạ dữ liệu (Data Masking) đã trở thành một trong những kỹ thuật phổ biến để bảo vệ dữ liệu nhạy cảm và đảm bảo an toàn thông tin. Kỹ thuật này cho phép ẩn danh hoặc che giấu dữ liệu trong một tập dữ liệu, giúp đảm bảo những thông tin này không bị tiết lộ và được bảo vệ.

Với mong muốn tìm hiểu về Data Masking, nhóm thực hiện báo cáo với đề tài: "**Viết chưng trình quản lý dữ liệu an toàn dưới dạng mặt nạ dữ liệu trên SQLite sử dụng ngôn ngữ Rust**". Với mục tiêu trên, báo cáo bao gồm các chương và bố cục sau:

Chương 1: Tổng quan về an toàn và bảo mật thông tin
Chương 2: Xây dựng chương trình
Chương 3: Kết luận và đánh giá

Hà Nội, ngày 04 tháng 11 năm 2022

Nhóm sinh viên thực hiện

MỤC LỤC

DANH SÁCH HÌNH VẼ

DANH SÁCH BẢNG

DANH SÁCH KÝ HIỆU VÀ TỪ VIẾT TẮT

3DES Triple DES. 5

AES Advanced Encryption Standard. 5

DBMS Database Management System. 6

DES Data Encryption Standard. viii

DH Diffie-Hellman. 10, 11

ECC Elliptic Curve Cryptography. 10, 11

ECDH Elliptic-curve Diffie-Hellman. 11, 12, 13

RSA Rivest Shamir Adleman. 5, 10

DANH SÁCH THUẬT NGỮ

Advanced Encryption Standard là một tiêu chuẩn mã hóa đối xứng với độ dài khóa là 128, 192 hoặc 256 bit. vi, 5

băm là quá trình biến đổi một chiều từ một chuỗi với độ dài bất kỳ tạo ra chuỗi có độ dài cố định. vii, 9

ChaCha là một họ mã hóa đối xứng của tác giả Daniel J. Bernstein. vii, 5, 9

ChaCha20 là hệ mã dòng ChaCha với 20 vòng. vii

ChaCha20-Poly1305 là hệ mã dòng có kiến trúc mã hóa với dữ liệu liên kết dựa trên ChaCha20 và Poly1305. ii, iv, 6, 7

Data Encryption Standard là một tiêu chuẩn mã hóa đối xứng được FIPS chọn làm chuẩn chính thức vào năm 1976. vi, viii

Elgamal encryption là thuật toán mã hóa không đối xứng dựa trên trao đổi khóa Diffie-Hellman. 5

Elliptic Curve Cryptography Mật mã học trên đường cong elliptici. vi, 10

giải mã là phương pháp đưa thông tin đã được mã hóa về dạng thông tin ban đầu. vii, viii, 5, 8

hàm băm là thuật toán thực hiện quá trình băm. 9

khóa là một chuỗi bit cho phép mã hóa hoặc giải mã. vii, viii, 7, 8, 10, 11, 12, 14

mã hóa là phương pháp để biến thông tin (phim ảnh, văn bản, hình ảnh...) từ định dạng bình thường sang dạng thông tin không thể hiểu được nếu không có phương tiện giải mã. vii, viii, 5, 8, 10, 12, 14

mã hóa không đối xứng (mã hóa khóa công khai) là phương pháp mã hóa sử dụng 2 khóa cho việc mã hóa giải mã . vii, viii, 5, 10, 13

mã hóa đối xứng (mã hóa khóa bí mật) là phương pháp mã hóa mà khóa mã hóa và khóa giải mã là như nhau . vii, viii, 5

mặt nạ dữ liệu là kỹ thuật che giấu dữ liệu nhạy cảm để tránh thất thoát thông tin. 4, 5

Poly1305 là họ băm được thiết kế bởi Daniel J. Bernstein sử dụng cho mật mã. vii, 8

quarter (một phần tư) là một trong số 4 phần nào đó. iv, 7, 8

Rivest Shamir Adleman là một hệ thống mã hóa không đối xứng được sử dụng rộng rãi cho việc truyền tin. vi, 5

round function là một hàm thực hiện các phép biến đổi trên một chuỗi có độ dài cố định. iv, 7, 8

trao đổi khóa Diffie-Hellman là một phương pháp toán học để trao đổi khóa mật mã một cách bảo mật qua kênh công khai. vii

Triple DES là một thuật toán mã hóa đối xứng, áp dụng thuật toán mã Data Encryption Standard (DES) ba lần cho mỗi khối dữ liệu. vi, 5

Chương 1: GIỚI THIỆU TỔNG QUAN

1.1 Tổng quan về an toàn và bảo mật thông tin

1.1.1 Giới thiệu

Ngày nay với sự phát triển bùng nổ của công nghệ thông tin, hầu hết các thông tin của doanh nghiệp như chiến lược kinh doanh, các thông tin về khách hàng, nhà cung cấp, tài chính, mức lương nhân viên,…đều được lưu trữ trên hệ thống máy tính. Cùng với sự phát triển của doanh nghiệp là những đòi hỏi ngày càng cao của môi trường kinh doanh yêu cầu doanh nghiệp cần phải chia sẻ thông tin của mình cho nhiều đối tượng khác nhau qua Internet. Việc mất mát, rò rỉ thông tin có thể ảnh hưởng nghiêm trọng đến tài chính, danh tiếng của công ty và quan hệ với khách hàng.

Hệ thống thông tin là một hệ thống bao gồm các yếu tố có quan hệ với nhau cùng làm nhiệm vụ thu thập, xử lý, lưu trữ và phân phối thông tin và dữ liệu và cung cấp một cơ chế phản hồi để đạt được một mục tiêu định trước. Các thành phần của hệ thống bao gồm phần cứng, phần mềm, mạng truyền dữ liệu, dữ liệu và con người trong hệ thống thông tin.

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 doanh nghiệp. Vì vậy an toàn và bảo mật thông tin là nhiệm vụ rất nặng nề và khó đoán trước được, nhưng tựu trung lại gồm ba hướng chính:

- Bảo đảm an toàn thông tin tại máy chủ
- Bảo đảm an toàn cho phía máy trạm
- Bảo mật thông tin trên đường truyền

1.1.2 Các mối đe dọa và thiệt hại

Tấn công mạng là một trong những vấn đề quan trọng về an toàn và bảo mật thông tin. Đó là những nỗ lực của kẻ tấn công để truy cập vào các hệ thống mạng của một tổ chức hoặc cá nhân mà không có sự cho phép. Những tấn công mạng này có thể gây ra những thiệt hại nghiêm trọng cho các tổ chức hoặc cá nhân, bao gồm mất dữ liệu, mất tiền và thiệt hại đến danh tính.

Phần mềm độc hại: Phần mềm độc hại là một loại phần mềm được thiết kế để gây hại cho hệ thống máy tính hoặc để truy cập vào thông tin cá nhân. Các loại phần mềm độc hại bao gồm virus, phần mềm giám điệp, phần mềm mã độc và phần mềm ransonware.

Xâm nhập: Xâm nhập là quá trình xâm nhập vào hệ thống hoặc mạng của một tổ chức hoặc cá nhân mà không có sự cho phép. Các tấn công xâm nhập này có thể gây ra những thiệt hại nghiêm trọng cho các tổ chức hoặc cá nhân, bao gồm mất dữ liệu và thiệt hại đến danh tiếng.

Lừa đảo trực tuyến: Lừa đảo trực tuyến là một hoạt động gian lận trực tuyến được thực hiện bằng cách sử dụng các kỹ thuật gian lận để lừa đảo người dùng đưa ra thông tin cá nhân hoặc tiền bạc.

Rò rỉ dữ liệu: Rò rỉ dữ liệu là quá trình tiết lộ thông tin cá nhân hoặc thông tin nhạy cảm của một cá nhân hoặc tổ chức cho người không có quyền truy cập vào thông tin đó. Rò rỉ dữ liệu có thể gây ra những thiệt hại nghiêm trọng cho các tổ chức hoặc cá nhân.

Như vậy có thể rút ra 3 mối đe dọa chủ yếu đối với hệ thống:

- Phá hoại: Kẻ thù phá hỏng thiết bị phần cứng hoặc phần mềm hoạt động trên hệ thống.
- Sửa đổi: Tài sản của hệ thống bị sửa đổi trái phép. Điều này thường làm cho hệ thống không hoạt động đúng chức năng của nó. Ví dụ như thay đổi mật khẩu, quyền người dùng làm họ không thể truy cập vào hệ thống để làm việc.
- Can thiệp: Tài sản bị truy cập bởi những người không có thẩm quyền, các truyền thông thực hiện trên hệ thống bị ngăn chặn, sửa đổi. thống

Đe dọa đối với một hệ thống thông tin có thể đến từ nhiều nguồn khác nhau và được thực hiện bởi các đối tượng khác nhau. Có 3 loại đối tượng chính:

- Các đối tượng bên trong hệ thống (insider), các đối tượng này có quyền truy cập hợp lệ đối với hệ thống.
- Các đối tượng bên ngoài hệ thống (hacker) thường tấn công thông qua các đường kết nối với hệ thống như Internet.
- Các phần mềm độc hại chạy trên hệ thống.

1.1.3 Giải pháp điều khiển và kiểm soát

Thông thường, có 3 biện pháp ngăn chặn các mối đe dọa:

- Điều khiển thông qua phần mềm dựa trên các cơ chế an toàn bảo mật của hệ thống (hệ điều hành) hoặc các thuật toán mã học.
- Điều khiển thông qua phần cứng nhờ vào các cơ chế bảo mật, các thuật toán mã học được cứng hóa.
- Điều khiển thông qua chính sách của tổ chức. Tổ chức ban hành các quy định nhằm đảm bảo tính an toàn của hệ thống.

1.1.4 Mục tiêu và nguyên tắc chung

Mục tiêu

Ba mục tiêu cơ bản trong việc đảm bảo an toàn và bảo mật thông tin bao gồm: Tính bí mật, Tính toàn vẹn thông tin và Độ sẵn sàng của thông tin.

- Tính bảo mật (Confidentiality): Bảo mật thông tin nghĩa là chỉ những người, máy tính được cấp phép mới có quyền truy cập và sử dụng thông tin của doanh nghiệp, hay nói cách khác, bảo mật là tránh để rò rỉ thông tin ra bên ngoài hệ thống. Những tin tặc có vô vàn cách thức để đánh cắp thông tin với mục đích xấu như giám sát hệ thống mạng của doanh nghiệp, hay Social Engineering. Vì vậy, các doanh nghiệp cần cải tiến hệ thống bảo mật thông tin (sử dụng firewall hoặc ACL, yêu cầu người dung cung cấp credential…) để tránh những việc đáng tiếc xảy ra. Tính mật của thông tin được đại diện bởi quyền READ.
- Tính toàn vẹn (Integrity): Đảm bảo tính toàn vẹn của thông tin nghĩa là chỉ người có thẩm quyền mới được chỉnh sửa thông tin nhưng không làm thay đổi sự chính xác của dữ liệu. Một cách phổ biến nhất để tội phạm mạng thay đổi thông tin chính là xâm nhập vào các lỗ hổng bảo mật trong hệ thống của doanh nghiệp. Tính toàn vẹn của thông tin được đại diện bởi quyền MODIFY
- Tính khả dụng (Availability): Có nghĩa là hệ thống lưu trữ và xử lý thông tin luôn sẵn sàng để được truy xuất ở bất cứ thời điểm nào với mục đích tránh những rủi ro về phần cứng, phần mềm, hay thậm chí tránh được hình thức tấn công từ chối dịch vụ (DoS).

Nguyên tắc chung

Hai nguyên tắc của bảo mật thông tin:

- Việc thẩm định về bảo mật là khó, và cần tính tới tất cả tình huống tấn công có thể thực hiện.

- Tải sản được bảo vệ đến khi hết giá trị sử dụng hoặc hết ý nghĩa bí mật

1.2 Mặt nạ dữ liệu

Mặt nạ dữ liệu là một kỹ thuật tạo ra phiên bản dữ liệu giả nhưng giống thật của tổ chức nhằm bảo vệ thông tin nhạy cảm nhưng vẫn cung cấp các thông tin thay thế khi thông tin thực không yêu cầu. Thông tin sử dụng cho tập huấn, thử nghiệm hay kiểm thử là một vài ví dụ.

Mặt nạ dữ liệu thực hiện thay đổi giá trị của dữ liệu nhưng vẫn giữ nguyên định dạng. Mục đích của việc này là tạo ra phiên bản không thể giải mã hay dịch ngược. Có một số cách thay đổi dữ liệu như xáo trộn, hoán vị hay mã hóa.

Việc sử dụng mặt nạ dữ liệu trở nên cực kỳ quan trọng trong một số trường hợp nhờ một số lý do:

- Mặt nạ dữ liệu giải quyết một số mối đe dọa quan trọng: Thất thoát, đánh cắp dữ liệu, các mối đe dọa nội bộ hay giao tiếp không an toàn với bên thứ 3.
- Giảm thiểu rủi ro liên quan đến sử dụng đám mây.
- Biến dữ liệu trở nên vô dụng đối với kẻ tấn công trong khi vẫn giữ lại các thuộc tính cho một số chức năng.
- Cho phép chia sẻ dữ liệu giữa các người dùng được ủy quyền như người kiểm thử và người phát triển mà không cần sử dụng dữ liệu thực.

Một số loại mặt nạ dữ liệu bao gồm:

- Mặt nạ dữ liệu tĩnh: giúp tạo ra một phiên bản mới của cơ sở dữ liệu không chứa dữ liệu gốc. Thông thường, quá trình này bao gồm các bước: Sao lưu dữ liệu, xác định thông tin không cần thiết, thực hiện ”đeo”mặt nạ cho dữ liệu được sao lưu.

  Việc sử dụng mặt nạ dữ liệu tĩnh cung cấp dữ liệu chất lượng cao cho việc phát triển và kiểm thử mà không bị lộ thông tin thực tế. Tuy nhiên, quá trình sao lưu và tạo mới tốn nhiều thời gian tùy thuộc vào độ lớn dữ liệu.

- Mặt nạ dữ liệu động: chủ yêu sử dụng chức năng phân quyền người dùng cho từng cơ sở dữ liệu hay ứng dụng. Quá trình truy cập dữ liệu được thực hiện thông qua một máy chủ trung gian. Dựa vào quyền hạn của người truy cập mà máy chủ thực hiện thay đổi dữ liệu tương ứng.

  Mặt nạ dữ liệu động giải quyết được vấn đề thời gian thực cho dữ liệu và thêm vào một lớp bảo mật cho cơ sở dữ liệu. Tuy nhiên dữ liệu lấy ra chất lượng không cao như phương pháp tĩnh và phụ thuộc vào máy chủ trung gian.

Các kỹ thuật mặt nạ dữ liệu đa dạng và có thể áp dụng vào các tình huống sử dụng khác nhau.

- Mã hóa dữ liệu: khi dữ liệu được mã hóa, nó trở nên vô nghĩa trừ khi người đọc có khóa giải mã. Đây là một phương pháp mặt nạ kỹ thuật cao và phức tạp bởi nó yêu cầu các kỹ thuật mã hóa dữ liệu, quản lý và chia sẻ khóa mã hóa.

- Xáo trộn dữ liệu: các ký tự được xáo trộn ngẫu nhiên thay thế cho dữ liệu gốc. Đây là một phương pháp dễ thực hiện nhưng chỉ có thể áp dụng cho một số loại dữ liệu cụ thể và kém bảo mật.

- Thay thế dữ liệu: dữ liệu được thay thế bởi dữ liệu cố định khi truy cập bởi người dùng không được ủy quyền. Phương pháp này khiến dữ liệu trở nên kém hữu ích cho việc phát triển và kiểm thử.

- Làm giả dữ liệu: dữ liệu được thay thể bởi dữ liệu giả nhưng giống thật được tạo ra ngẫu nhiên.

Đối với kỹ thuật mã hóa dữ liệu, ngoài việc ẩn đi dữ liệu gốc, dữ liệu mã hóa có thể được khôi phục trực tiếp đối với người dùng có khóa giải mã. Vì vậy khóa giải mã cần được quản lý riêng.

1.3 Mã hóa dữ liệu

Mã hóa được coi là phương pháp bảo vệ hiệu quả giúp đảm bảo bảo mật và sự riêng tư của dữ liệu. Phương pháp này cung cấp tính bảo mật trong bộ ba mục tiêu Bảo mật-Toàn vẹn-Sẵn sàng. Nếu dữ liệu được mã hóa bị mất hay truy cập không được phép, nó vẫn luôn được bảo vệ. Vì vậy, nó có thể được sử dụng để truyền dữ liệu qua kênh công khai nay được dùng mã hóa dữ liệu trước khi lưu vào cơ sở dữ liệu.

Hai phương pháp mã hóa được sử dụng là mã hóa đối xứng và mã hóa không đối xứng:

- Mã hóa đối xứng là phương pháp mã hóa sử dụng chung một khóa bí mật. Một số thuật toán mã hóa bao gồm Advanced Encryption Standard (AES), Triple DES (3DES), ChaCha, ...

- Mã hóa không đối xứng sử dụng 2 khóa độc lập: khóa công khai và khóa bí mật. Khi dữ liệu được mã hóa bởi bởi khóa công khai, chỉ có khóa bí mật tương ứng mới có thể giải mã. Rivest Shamir Adleman (RSA) là thuật toán lâu đời và nổi tiếng nhất cho phương pháp mã hóa này. Ngoài ra các thuật toán khác như Elgamal encryption cũng có các ưu điểm riêng.

1.4 Cơ sở dữ liệu SQLite

SQLite là một Database Management System (DBMS) quan hệ tương tự như MySQL, ... Đặc điểm nổi bật của SQLite so với các DBMS khác là gọn, nhẹ, đơn giản, đặt biệt không cần mô hình server-client, không cần cài đặt, cấu hình hay khởi động nên không có khái niệm user, password hay quyền hạn trong SQLite Database. Dữ liệu cũng được lưu ở một file duy nhất.

SQLite thường không được sử dụng với các hệ thống lớn nhưng với những hệ thống ở quy mô vùa và nhỏ thì SQLite không thua các DBMS khác về chức năng hay tốc độ. Vì không cần cài đặt hay cấu hình nên SQLite được sử dụng nhiều trong việc phát triển, thử nghiệm … vì tránh được những rắc rối trong quá trình cài đặt.

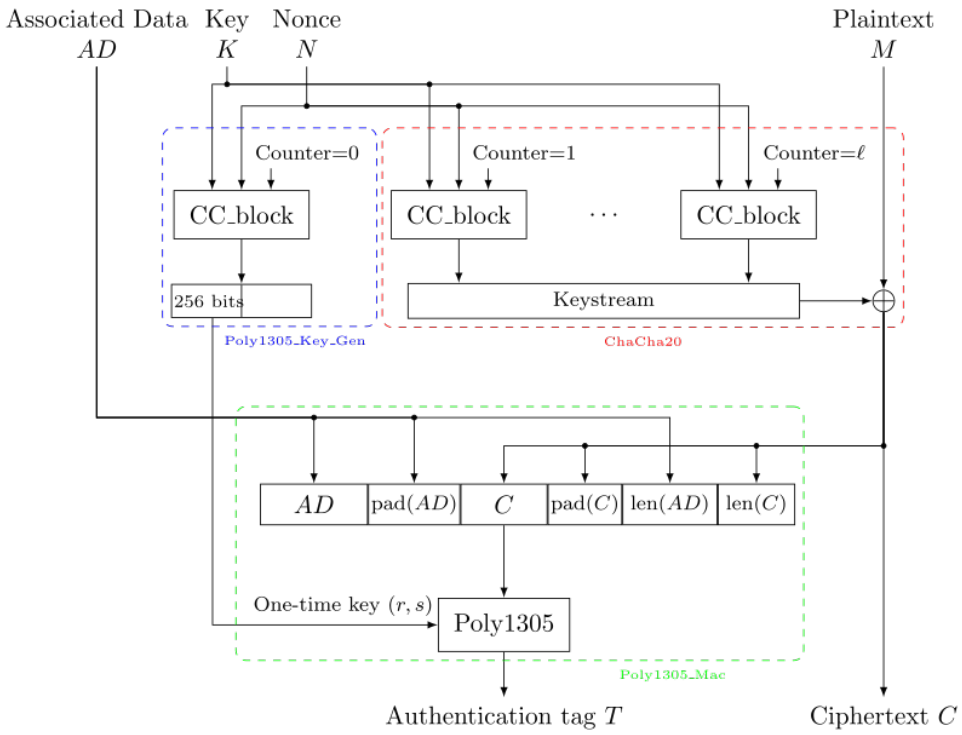
Do được lưu trữ trực tiếp trên một file duy nhất, nếu không quản lý tốt trong hệ thống file, dữ liệu có thể bị lộ một cách dễ dàng. Một cách khác giúp bảo vệ dữ liệu trong SQLite là áp dụng mã hóa dưới dạng mặt nạ dữ liệu cho các bảng bên trong cơ sở dữ liệu.

SQLite có một hệ thống kiểu dữ liệu đơn giản bao gồm 5 kiểu:

- NULL: giá trị NULL.
- Integer: giá trị số nguyên có độ dài 8, 16, 32 hay 64 bit tùy thuộc vào giá trị.
- Float: giá trị số thực 64 bit.
- Text: giá trị kiểu chuỗi Unicode.
- Blob: giá trị kiểu blob có thể chứa số lượng byte không biết trước. Đây là kiểu dữ liệu đa dụng nhất có thể lưu trữ toàn bộ các kiểu dữ liệu còn lại hoặc lưu trữ dữ liệu đã được mã hóa.

1.5 Hệ mã dòng có xác thực ChaCha20-Poly1305

ChaCha20-Poly1305 là hệ mã dòng có kiến trúc mã hóa với dữ liệu liên kết (Authenticated Encryption with Additional Data - AEAD) cung cấp tính bí mật và xác thực nguồn gốc dữ liệu truyền nhận trên kênh liên lạc. ChaCha20-Poly1305 có kiến trúc bao gồm hai thành phần chính là thuật toán mã dòng ChaCha20 và cơ chế xác thực Poly1305 của cùng tác giả là D.J. Berstein.



Hình 1.1: Sơ đồ khối hệ mã dòng có xác thực ChaCha20-Poly1305

1.5.1 Hệ mã dòng ChaCha

ChaCha có trạng thái khởi tạo rộng 64 byte (512bit) bao gồm lần lượt 16 byte cố đinh "$expand\ 32-byte\ k$", khóa độ dài 32 byte, 8 byte counter và 8 byte vector khởi tạo. Trạng thái này chia thành 16 khối 4 byte trên ma trận 4x4 để đưa vào quarter-round function.

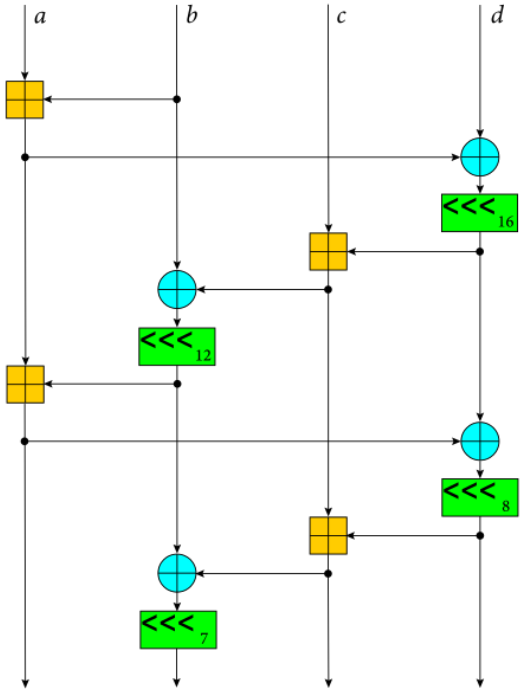
Bảng 1.1: Trạng thái khởi tạo của thuật toán ChaCha

”expa”	”nd 3”	”2-by”	”te k”
key[0..4]	key[4..8]	key[8..12]	key[12..16]
key[16..20]	key[20..24]	key[24..28]	key[28..32]
counter[0..4]	counter[4..8]	nonce[0..4]	nonce[4..8]

Từng bộ trong 4 bộ 4 byte lần lượt được đưa vào quarter-round function tạo lên một vòng. Đối với vòng lẻ, lần lượt 4 khối trên từng cột của ma trận trạng thái được đưa vào quarter-round function, đối vòng chẵn, lần lượt 4 khối trên 4 hàng của ma trận trạng thái được đưa vào quarter-round function.

Tên của từng thuật toán ChaCha ứng với số vòng được áp dụng như ChaCha8 thực hiện 8 vòng, ChaCha20 thực hiện 20 vòng.

Sau khi qua các vòng, ma trận trạng thái được sử dụng để cộng modulo-2 với tối đa 64 byte bản rõ tạo thành 64 byte bản mã. Nếu độ dài bản rõ nhiều hơn 64 byte, giá trị



Hình 1.2: Quarter-round function của thuật toán ChaCha

counter trong trạng thái khởi tạo được cộng thêm 1 và thực hiện qua các vòng để áp dụng cho các khối tối đa 64 byte tiếp theo.

Quá trình giải mã từ bản mã thành bản rõ được thực hiện giống quá trình mã hóa do việc biến đổi chỉ dựa trên phép cộng modulo-2.

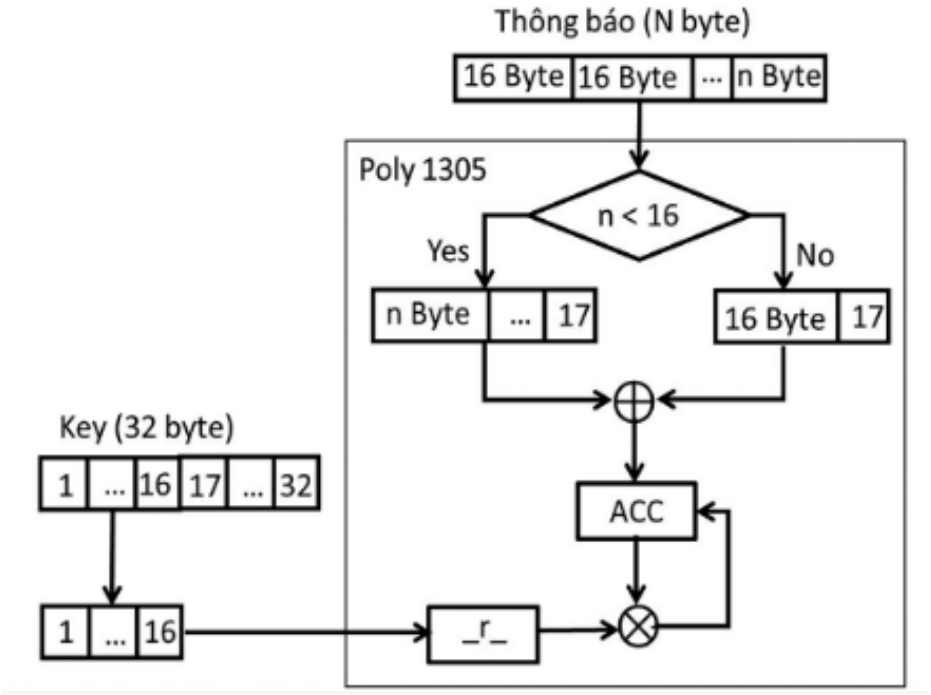
1.5.2 Cơ chế xác thực Poly1305

Poly1305 là cơ chế xác thực thông báo với đầu vào khóa 256 bit và thông báo có độ dài không cố định. Đầu ra là một thẻ xác thực độ dài 128 bit được sử dụng bởi bên nhận nhằm xác thực nguồn thông báo.

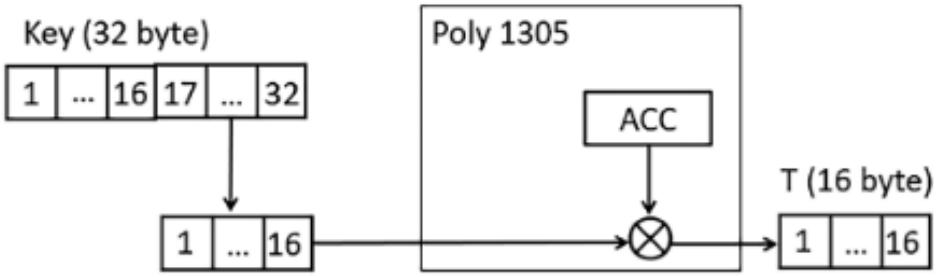
Khóa đầu vào được chia thành 2 phần gọi là r và s có độ dài 128 bit. Cặp (r, s) phải là duy nhất và không thể đoán cho mỗi lần gọi. r cần được xử lý bằng cách XOR với 0x0ffffffc0ffffffc0ffffffc0fffffff.

Thông báo đầu vào được chia thành các khối 16 byte, khối cuối cùng có thể ngắn hơn được thêm các bit 0. Các khối 16 byte này được thêm vào 1 byte có giá trị 0x01 thành 17 byte. Các phép tính được thực hiện trên các khối này với r trên trường Z () để tạo một bộ tích lũy ACC như trong hình 1.3.

Cuối cùng giá trị s được thêm vào bộ tích lũy và 128 bit được lấy ra làm thẻ xác thực. Xem hình 1.4.



Hình 1.3: Quá trình tính toán của cơ chế Poly1305



Hình 1.4: Tính toán thẻ xác thực trong Poly1305

1.6 Hàm băm BLAKE2s

BLAKE là hàm băm mật mã dựa trên mã dòng ChaCha của Daniel J. Bernstein nhưng một bản sao hoán vị của khối đầu vào được XOR với các hằng số vòng được thêm vào trước mỗi bước vòng của ChaCha.

ChaCha sử dụng ma trận trạng thái kích thước $4x4$ word. BLAKE liên tục kết hợp 8 word giá trị băm với 16 word giá trị thông báo, cắt bớt giá trị trả về của ChaCha để lấy giá trị băm tiếp theo. BLAKE-256 sử dụng word độ dài 32 bit và thu được 256 bit băm.

BLAKE2 được phát triển dựa trên BLAKE bằng cách tinh giản BLAKE và giảm số vòng từ 16 xuống 12 đối với BLAKE2b và từ 14 xuống 10 đối với BLAKE2s.

BLAKE2s là biến thể BLAKE2 được phát triển từ BLAKE-256, có độ dài đầu ra 256 bit.

1.7 Trao đổi khóa Diffie-Hellman và Elliptic-curve Diffie-Hellman

1.7.1 Trao đổi khóa Diffie-Hellman

Diffie-Hellman (DH) là một phương pháp trao đổi khóa được phát minh sớm nhất trong mật mã học. Phương pháp DH cho phép 2 hay nhiều bên thiết lập một khóa bí mật chung để mã hóa dữ liệu sử dụng trên kênh không an toàn mà không cần có sự thỏa thuận trước về khóa bí mật giữa hai bên. Khóa bí mật được tạo ra có thể sử dụng để mã hóa dữ liệu với phương pháp mã hóa không đối xứng.

Quá trình thiết lập khóa được mô tả như sau:

1. Hai bên sử dụng chung một nhóm cyclic hữu hạn G, một phần tử sinh g thuộc G và số nguyên số p.
2. Alice chọn số a ngẫu nhiên và gửi g^a mod p cho Bob.
3. Bob tương tự chọn số b ngẫu nhiên và gửi g^b mod p cho Alice.
4. Alice tính $(g^b)^a$ mod p.
5. Bob tính $(g^a)^b$ mod p.

Như vậy sau quá trình tính toán, hai bên có chung giá trị bí mật chung g^{ab} mod p. Và có thể được sử dụng để mã hóa hoặc tạo khóa mã hóa. Việc tính ra g^{ab} mod p từ các giá trị g, p, g^a, g^b được mô tả dưới dạng bài toán tính logarit rời rạc và được chứng minh là bài toán khó ngay cả đối với siêu máy tính.

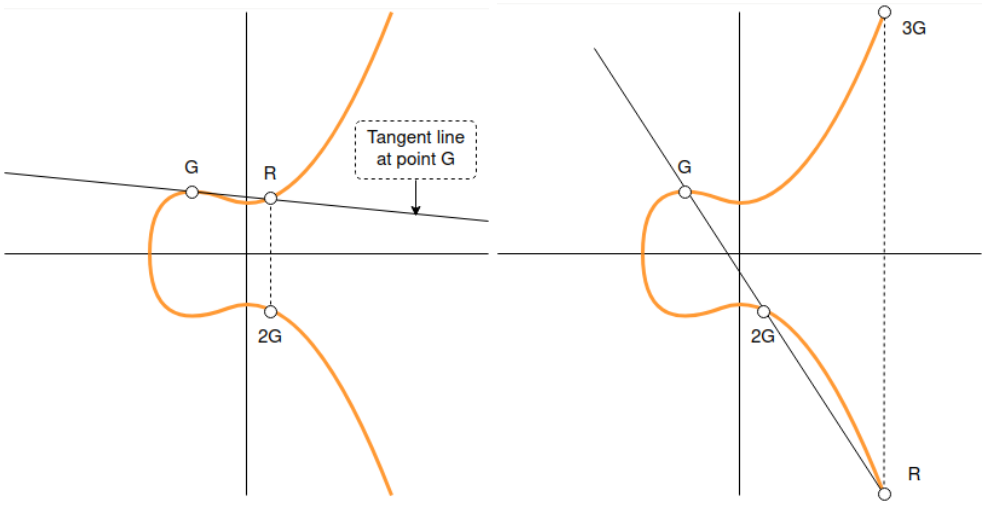
Ngoài chức năng trao đổi khóa, DH còn được sử dụng như một phần của hạ tầng khóa công khai. Khi đó, khóa công khai của A là (g^a mod p, g, p). Để mã hóa công khai cho A, B chọn số b ngẫu nhiên và gửi g^b mod p cũng thông điệp được mã hóa bởi khóa $(g^a)^b$ mod p. Do a được giữ bí mật, chỉ A có thể giải mã được thông điệp.

1.7.2 Mật mã đường cong Elliptic

Elliptic Curve Cryptography (ECC) là một trong những loại hiện đại, mạnh nhất hiện nay, cung cấp hiệu năng cao và an toàn hơn so với thế hệ đầu tiên RSA. Với độ dài 256 bit, khóa hệ mật trên đường cong elliptic có độ bảo mật tương đương khóa RSA độ dài 3248 bit.

Đường cong elliptic được biểu diễn dưới dạng $y^2 = x^3 + ax + b$. ECC hoạt động dựa trên dựa trên tính chất đường thẳng bất kỳ cắt đường cong elliptic tại nhiều nhất 3 điểm.

Nhân đôi điểm được sử dụng khi có 1 điểm A ban đầu. Đường tiếp tuyến với đường cong tại A sẽ cắt đường cong tại B. Đường thẳng song song với trục tung cắt đường cong tại điểm là kết quả của phép toán. Trong hình 1.5a ta có điểm ban đầu G và kết quả 2G.



(a) Nhân đôi điểm (b) Cộng hai điểm

Hình 1.5: Hai thao tác trên đường cong elliptic

Cộng 2 điểm được thực hiện với 2 điểm đầu vào. Đường thẳng đi qua 2 điểm cắt đường thẳng tại điểm thứ 3. Đường thẳng song song với trục tung đi qua điểm này cắt đường cong tại điểm thứ 2 là kết quả của phép toán. Hình 1.5b mô tả phép cộng 2 điểm G và 2G cho kết quả 3G.

Các tham số của ECC có dạng (p, a, b, G, n, h):

- a, b là tham số đường cong elliptic.
- Số nguyên tố p lớn giới hạn kích thước trường giá trị.
- Điểm khởi đầu G
- Số điểm n trên đường cong
- Số nhóm cyclic h của đường cong. Một số đường cong có 1 nhóm, một số khác có nhiều hơn 1 nhóm.

1.7.3 Trao đổi khóa Diffie-Hellman trên đường cong elliptic

Thay vì sử dụng phép lũy thừa như DH, Elliptic-curve Diffie-Hellman (ECDH) sử dụng phép nhân trong ECC để sinh khóa.

ECDH thực hiện 2 bước thỏa thuận khóa: tạo khóa và tính toán giá trị bí mật chung.

Tham số bí mật là số n ngẫu nhiên với $n \in 1..n_0$. Tham số công khai được tính $P = nG$ với n là tham số bí mật, G là điểm sinh, P là tham số công khai.

Giá trị bí mật được tính bằng $S = P_1 * n_2 = P_2 * n_1 = n_1 * n_2 * G = n_2 * n_1 * G$

Chương 2: THIẾT KẾ HỆ THỐNG

2.1 Lựa chọn công cụ

Rust là ngôn ngữ lập trình hệ thống mới nổi với nhiều ưu thế đảm bảo an toàn bộ nhớ, an toàn về luồng kèm theo các công cụ hỗ trợ mạnh mẽ, vì vậy nhóm sử dụng Rust làm ngôn ngữ lập trình chính.

Do lập trình giao diện trên Rust còn khó khăn, nhóm thực hiện phần mềm trên môi trường web tĩnh, các trang html được viết dưới dạng khung mẫu hỗ trợ bởi thư viện Tera.

Giao tiếp giữa cơ sở dữ liệu SQLite với chương trình được thực hiện qua thư viện sqlx. Sqlx là một thư viện hỗ trợ kết nối tới nhiều cơ sở dữ liệu khác nhau hỗ trợ cú pháp được phân tích ngay trong thời điểm biên dịch chương trình, giúp giảm thiểu lỗi khi lập trình.

Thuật toán X25519 được sử dụng cho trao đổi khóa ECDH do nó được coi là an toàn. Mã hóa đối xứng sử dụng thuật hệ mã ChaCha20Poly1305. Hàm băm BLAKE2s được sử dụng cho việc sinh khóa.

2.2 Thiết kế phần mềm

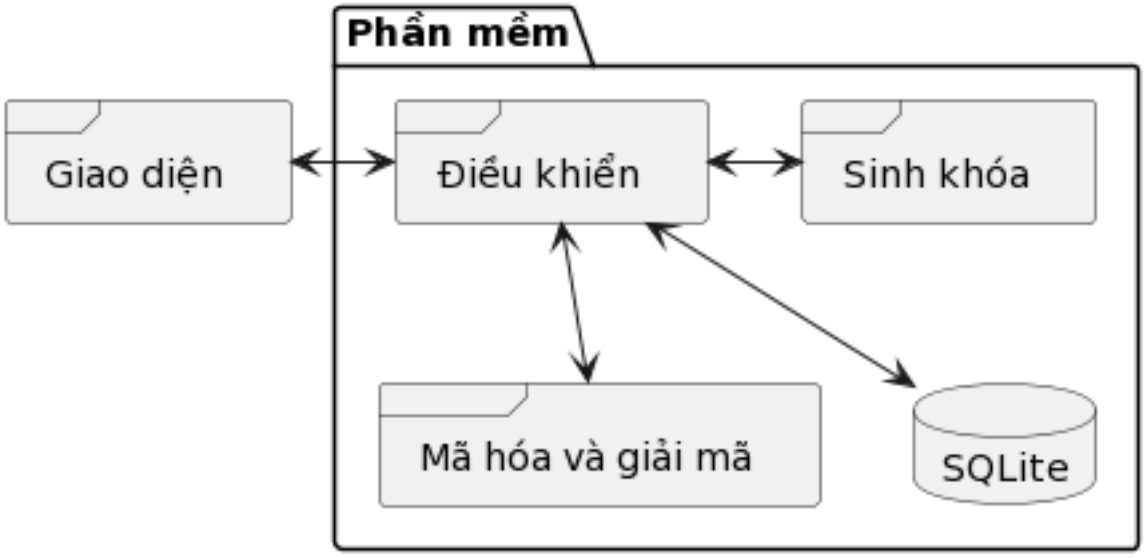
2.2.1 Thiết kế tổng thể

Các thành phần của phần mềm bao gồm:

1. Thành phần quản lý: nơi lưu trữ dữ liệu phân quyền, dữ liệu, điều khiển chương trình, thực hiện các thao tác mã hóa, sinh khóa gọi chung là phần mềm.

2. Thành phần mã hóa, giải mã: thực hiện mã hóa dữ liệu trên các ô khi lưu trữ và giải mã khi có yêu cầu từ người dùng hợp lệ. Dữ liệu được mã hóa sử dụng hệ thống mã hóa công khai, cho phép nhiều người dùng có thể đọc dữ liệu với khóa riêng và cho các người dùng khác có thể đọc được dữ liệu sau khi cập nhật mới.

3. Thành phần sinh khóa: thực hiện tạo khóa mã hóa cho người dùng. Mỗi người dùng có một cặp khóa công khai-bí mật, cặp khóa này được sinh bởi tên và mật khẩu nhập bởi người dùng và người dùng chịu trách nhiệm quản lý mật khẩu này.



Hình 2.1: Các thành phần của phần mềm

2.2.2 Thành phần sinh khóa

Tên và mật khẩu được gắn liền với mỗi người dùng và khóa giải mã, mã hóa được tạo ra từ 2 tham số này. Khóa bí mật s tính theo công thức $s = h(n + p)$ trong đó h là hàm băm BLAKE2s, n là tên người dùng, p là mật khẩu, + là phép nối chuỗi.

Khóa công khai k được tính theo thuật toán ECDH: $k = sG$.

2.2.3 Thành phần mã hóa - giải mã

Với mỗi đoạn dữ liệu cần mã hóa, kỹ thuật mã hóa không đối xứng được sử dụng dựa trên thuật toán X25519. Khi cần cập nhật dữ liệu, khóa bí mật tạm thời s_e được tạo ngẫu nhiên, đối với mỗi người dùng với khóa công khai k_x, bí mật chung Sh_x được tính bằng $Sh_x = s_e k_x$. Bí mật chung Sh_x sử dụng để tính khóa mã hóa Kc_x bằng cách đưa vào hàm băm h: $Kc_x = h(Sh_x) = h(s_e k_x)$. Khóa công khai tạm thời k_e được lưu cùng với dữ liệu mã hóa. Dữ liệu m cho từng người dùng x được mã hóa bởi khóa Kc_x qua hàm mã hóa C: $t_x = E(Kc_x, m)$. Các cặp khóa công khai, và dữ liệu mã hóa (k_x, t_x) được lưu kèm cùng với khóa công khai tạm thời k_e tạo thành bộ dữ liệu mã hóa: $T = (k_e, ((k_1, t_1), ..., (k_n, t_n)))$ với $t_x = E(h(s_e k_x), m) = E(h(s_x k_e), m)$

Quá trình giải mã được thực hiện bởi người dùng x có khóa bí mật s_x. Từ tập T, người dùng x lấy khóa công khai tạm thời k_e, và tìm thông điệp t_x thông qua khóa k_x của họ. Khóa giải mã Kc_x được tính $Kc_x = h(s_x ke)$ và được sử dụng để giải mã thông điệp m từ bản mã t_x: $m = D(h(s_e k_x), t_x$.

2.2.4 Thành phần quản lý

Thông tin tên và khóa công khai của người dùng được phần mềm lưu trữ trong bản smask_role:

Bảng 2.1: Bảng người dùng smask_role

Tên cột	Kiểu dữ liệu	Chú thích
smask_role	text	Tên người dùng
smask_key	blob	Khóa công khai

Các cột và bảng được có quyền truy cập bởi người dùng được lưu trong bảng smask_role_table và smask_role_column với smask_key là khóa công khai của người dùng có quyền hạn: |

Bảng 2.2: Bảng thông tin phân quyền cho bảng

Tên cột	Kiểu dữ liệu	Chú thích
smask_table	text	Tên bảng
smask_key	blob	Khóa công khai

Bảng 2.3: Bảng thông tin phân quyền cho cột

Tên cột	Kiểu dữ liệu	Chú thích
smask_column	text	Tên cột
smask_table	text	Tên bảng
smask_key	blob	Khóa công khai

Dữ liệu được lưu trong các cột có cấu trúc:

```
struct Cell (ephemeral_pubkey, Map<user_pubkey, cipher_text>)
```

Trong đó: ephemeral_pubkey là khóa công khai tạm thời, user_pubkey là khóa công khai của người dùng và cipher_text là dữ liệu m được mã hóa của ô đối với người dùng được chọn.

Chương 3: THỬ NGHIỆM - ĐÁNH GIÁ

3.1 Sản phẩm thực tế

3.2 Thử nghiệm chức năng

3.3 Thử nghiệm hiệu năng, tốc độ

3.4 Đánh giá hệ thống