TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP HCM

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO MÔN: BẢO MẬT MẠNG MÁY TÍNH VÀ HỆ THỐNG

Đề Tài: NGHIÊN CỨU THUẬT TOÁN BẢO MẬT DỮ LIỆU CHO HỆ THỐNG IOT

Giảng viên bộ môn: **ThS. Phạm Trọng Huynh**

Lớp: **10_ĐH_CNPM1**

Khóa: **2021 - 2025**

Sinh viên thực hiện:

STT	Mã số sinh viên	Họ và tên
1	1050080002	Nguyễn Hoàng Anh
2	1050080014	Nguyễn Thái Hoành

TP. Hồ Chí Minh, tháng 5 năm 2025

TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP HCM

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO MÔN: BẢO MẬT MẠNG MÁY TÍNH VÀ HỆ THỐNG

Đề Tài: NGHIÊN CỨU THUẬT TOÁN BẢO MẬT DỮ LIỆU CHO HỆ THỐNG IOT

Giảng viên bộ môn: **ThS. Phạm Trọng Huynh**

Lớp: **10_ĐH_CNPM1**

Khóa: **2021 - 2025**

Sinh viên thực hiện:

STT	Mã số sinh viên	Họ và tên
1	1050080002	Nguyễn Hoàng Anh
2	1050080014	Nguyễn Thái Hoành

TP. Hồ Chí Minh, tháng 5 năm 2025

LỜI CẢM ƠN

Trong quá trình thực hiện đề tài nghiên cứu thuật toán đầy thách thức và thú vị này, chúng em đã nhận được sự hỗ trợ và hướng dẫn vô cùng quý báu từ nhiều cá nhân và tổ chức. Chúng em xin bày tỏ lòng biết ơn sâu sắc đến tất cả mọi người đã góp phần vào thành công của dự án này.

Trước hết, chúng em xin gửi lời cảm ơn chân thành đến thầy Phạm Trọng Huynh, người đã tận tâm hướng dẫn, truyền đạt kiến thức và kinh nghiệm quý báu cho chúng em. Những lời khuyên, sự động viên và định hướng của thầy đã giúp chúng em vượt qua những khó khăn, thử thách trong quá trình nghiên cứu và phát triển thuật toán.

Mặc dù đã cố gắng hết sức, chúng em nhận thức rằng báo cáo này không thể tránh khỏi những sai sót và điểm cần cải thiện. Chúng em rất mong nhận được những ý kiến đóng góp và phản hồi từ quý thầy cô, các chuyên gia và bạn đọc để báo cáo được hoàn thiện hơn, đáng tin cậy và hiệu quả hơn trong tương lai.

Một lần nữa, chúng em xin chân thành cảm ơn tất cả những người đã góp phần vào thành công của đề tài nghiên cứu này.

NHẬN XÉT CỦA GIẢNG VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Điểm

MỤC LỤC

MỤC LỤC HÌNH ẢNH

PHẦN 1: TỔNG QUAN VỀ HỆ THỐNG IOT VÀ BẢO MẬT DỮ LIỆU

1.1. Giới thiệu tổng quan về hệ thống IoT

1.1.1. Khái niệm và đặc điểm của IoT

Khái niệm:

- IoT (Internet vạn vật) là một mạng lưới các thiết bị vật lý ("vật thể") được nhúng với cảm biến, phần mềm và các công nghệ khác nhằm mục đích kết nối và trao đổi dữ liệu với các thiết bị và hệ thống khác qua internet1 hoặc các mạng truyền thông khác
- Các thiết bị IoT có thể là bất cứ thứ gì, từ đồ gia dụng đơn giản đến các công cụ công nghiệp phức tạp.

Đặc điểm:

- Kết nối: Các thiết bị IoT có khả năng kết nối với nhau và với internet, cho phép trao đổi dữ liệu và tương tác từ xa.
- Cảm biến: Nhiều thiết bị IoT được trang bị cảm biến để thu thập dữ liệu về môi trường xung quanh hoặc trạng thái của chính thiết bị.
- Xử lý dữ liệu: Các thiết bị IoT có khả năng xử lý dữ liệu thu thập được, từ đơn giản đến phức tạp, trước khi truyền đến các hệ thống khác.
- Tự động hóa: Nhiều hệ thống IoT có khả năng tự động hóa các quy trình và hoạt động dựa trên dữ liệu thu thập được.
- Quy mô lớn: Hệ thống IoT có thể bao gồm hàng triệu, thậm chí hàng tỷ thiết bị, tạo ra một lượng dữ liệu khổng lồ.
- Tính phân tán: Các thiết bị IoT thường được phân tán rộng rãi, đòi hỏi các giải pháp bảo mật và quản lý dữ liệu hiệu quả.

1.1.2. Kiến trúc và các thành phần của hệ thống IoT

Kiến trúc:

- Lớp thiết bị: Bao gồm các thiết bị vật lý được trang bị cảm biến, bộ xử lý và khả năng kết nối mạng.
- Lớp mạng: Cung cấp kết nối giữa các thiết bị IoT và các hệ thống khác, bao gồm các giao thức truyền thông như Wi-Fi, Bluetooth, Zigbee, v.v.
- Lớp nền tảng: Cung cấp các dịch vụ quản lý thiết bị, lưu trữ và xử lý dữ liệu, và các ứng dụng IoT.
- Lớp ứng dụng: Cung cấp các giao diện người dùng và các ứng dụng cho phép người dùng tương tác với hệ thống IoT.

Các thành phần:

- Thiết bị IoT: Các thiết bị vật lý được nhúng với cảm biến và khả năng kết nối mạng.
- Cảm biến: Thu thập dữ liệu về môi trường xung quanh hoặc trạng thái của thiết bị.
- Bộ xử lý: Xử lý dữ liệu thu thập được từ cảm biến.
- Giao thức truyền thông: Cho phép các thiết bị IoT kết nối và trao đổi dữ liệu.
- Nền tảng IoT: Cung cấp các dịch vụ quản lý thiết bị, lưu trữ và xử lý dữ liệu.
- Ứng dụng IoT: Cho phép người dùng tương tác với hệ thống IoT.

1.1.3. Các ứng dụng của IoT trong thực tiễn

- Nhà thông minh: Tự động hóa các thiết bị gia dụng, hệ thống chiếu sáng, hệ thống an ninh,...
- Thành phố thông minh: Quản lý giao thông, chiếu sáng công cộng, thu gom rác thải,...
- Y tế: Theo dõi sức khỏe bệnh nhân từ xa, quản lý thiết bị y tế,...
- Nông nghiệp thông minh: Giám sát điều kiện môi trường, quản lý tưới tiêu,...
- Công nghiệp: Tự động hóa quy trình sản xuất, giám sát thiết bị,...
- Giao thông vận tải: Theo dõi vị trí phương tiện, quản lý giao thông,...
- Bán lẻ: Theo dõi hàng tồn kho, quản lý chuỗi cung ứng,...

1.2. Tổng quan về bảo mật dữ liệu trong hệ thống IoT

1.2.1. Tầm quan trọng của bảo mật dữ liệu trong IoT

Bảo vệ thông tin cá nhân:

- Các thiết bị IoT thu thập một lượng lớn dữ liệu cá nhân, bao gồm thông tin về vị trí, thói quen sinh hoạt, sức khỏe, v.v.
- Việc bảo vệ dữ liệu này là rất quan trọng để đảm bảo quyền riêng tư của người dùng.

Bảo vệ tài sản:

- Các hệ thống IoT được sử dụng trong nhiều lĩnh vực quan trọng như công nghiệp, giao thông vận tải, năng lượng, v.v.
- Việc bảo vệ dữ liệu trong các hệ thống này là rất quan trọng để đảm bảo an toàn cho tài sản và hoạt động của các tổ chức.

Ngăn chặn các cuộc tấn công mạng:

- Các thiết bị IoT thường có khả năng bảo mật yếu, khiến chúng trở thành mục tiêu dễ dàng cho các cuộc tấn công mạng.
- Các cuộc tấn công này có thể gây ra hậu quả nghiêm trọng, chẳng hạn như đánh cắp dữ liệu, làm gián đoạn dịch vụ, hoặc gây thiệt hại vật chất.

Xây dựng lòng tin của người dùng:

- Người dùng sẽ không tin tưởng và sử dụng các hệ thống IoT nếu họ không cảm thấy dữ liệu của mình được bảo vệ an toàn.
- Việc đảm bảo an ninh bảo mật là yếu tố then chốt để thúc đẩy sự phát triển của IoT.

1.2.2. Các thách thức và yêu cầu về bảo mật dữ liệu trong IoT

Thách thức:

- Tính phân tán: Các thiết bị IoT thường được phân tán rộng rãi, khiến việc quản lý và bảo mật trở nên khó khăn.
- Tài nguyên hạn chế: Nhiều thiết bị IoT có tài nguyên tính toán và năng lượng hạn chế, khiến việc triển khai các giải pháp bảo mật phức tạp trở nên khó khăn.
- Tính đa dạng: Các hệ thống IoT bao gồm nhiều loại thiết bị và giao thức khác nhau, khiến việc xây dựng các giải pháp bảo mật chung trở nên khó khăn.

- Tính liên tục: Các hệ thống IoT thường hoạt động liên tục, khiến việc cập nhật và vá lỗi bảo mật trở nên khó khăn.

Yêu cầu:

- Tính bảo mật: Dữ liệu phải được bảo vệ khỏi các truy cập trái phép.
- Tính toàn vẹn: Dữ liệu phải được đảm bảo không bị thay đổi hoặc giả mạo.
- Tính sẵn sàng: Dữ liệu phải được cung cấp cho người dùng khi cần thiết.
- Tính riêng tư: Dữ liệu cá nhân phải được bảo vệ khỏi bị tiết lộ.
- Khả năng mở rộng: Các giải pháp bảo mật phải có khả năng mở rộng để đáp ứng sự phát triển của hệ thống IoT.

1.2.3. Các phương pháp bảo mật dữ liệu truyền thống và hiện đại

Phương pháp truyền thống:

- Mật khẩu: Sử dụng mật khẩu để xác thực người dùng.
- Tường lửa: Ngăn chặn các truy cập trái phép vào mạng.
- Hệ thống phát hiện xâm nhập (IDS): Phát hiện các hoạt động đáng ngờ trên mạng.
- Mã hóa: Mã hóa dữ liệu để bảo vệ khỏi bị đọc trái phép.

Phương pháp hiện đại:

- Xác thực đa yếu tố (MFA): Yêu cầu người dùng cung cấp nhiều yếu tố xác thực.
- Blockchain: Sử dụng công nghệ blockchain để đảm bảo tính toàn vẹn của dữ liệu.
- Học máy (Machine learning): Sử dụng học máy để phát hiện các cuộc tấn công mạng.
- Bảo mật dựa trên hành vi (Behavioral-based security): Phân tích hành vi của người dùng và thiết bị để phát hiện các hoạt động bất thường.
- Bảo mật dựa trên danh tiếng (Reputation-based security): Sử dụng thông tin về danh tiếng của người dùng và thiết bị để đánh giá rủi ro.

PHẦN 2: HÌNH THỨC TẤN CÔNG VÀ PHƯƠNG PHÁP BẢO MẬT DỮ LIỆU TRONG HỆ THỐNG IOT

2.1. Phân loại các hình thức tấn công dữ liệu trong IoT

2.1.1. Tấn công vật lý (Physical attacks)

Khái niệm:

- Tấn công vật lý là các hình thức tấn công nhắm vào phần cứng của thiết bị IoT.
- Kẻ tấn công có thể truy cập trực tiếp vào thiết bị để can thiệp, thay đổi hoặc đánh cắp dữ liệu.

Các hình thức tấn công:

- Giả mạo thiết bị (Device spoofing): Thay thế thiết bị IoT bằng một thiết bị giả mạo để thu thập hoặc thay đổi dữ liệu.
- Tấn công xâm nhập phần cứng (Hardware tampering): Can thiệp vào phần cứng của thiết bị để đánh cắp dữ liệu hoặc cài đặt mã độc.
- Tấn công side-channel (Side-channel attacks): Phân tích các thông tin phụ phát ra từ thiết bị (như điện năng tiêu thụ, thời gian xử lý) để suy luận ra dữ liệu nhạy cảm.
- Tấn công tamper (Tamper attacks): Cố gắng can thiệp vật lý vào thiết bị để thay đổi chức năng hoặc đánh cắp dữ liệu.

2.1.2. Tấn công mạng (Network attacks)

Khái niệm:

- Tấn công mạng là các hình thức tấn công nhắm vào hệ thống mạng kết nối các thiết bị IoT.
- Kẻ tấn công có thể khai thác các lỗ hổng trong giao thức truyền thông hoặc cấu hình mạng để xâm nhập và đánh cắp dữ liệu.

Các hình thức tấn công:

- Tấn công từ chối dịch vụ (DoS/DDoS): Kẻ tấn công làm quá tải hệ thống mạng bằng cách gửi một lượng lớn yêu cầu, khiến hệ thống không thể hoạt động bình thường.
- Tấn công giả mạo (Spoofing): Kẻ tấn công giả mạo địa chỉ IP hoặc MAC để truy cập trái phép vào hệ thống mạng.

- Tấn công nghe lén (Eavesdropping): Kẻ tấn công chặn và ghi lại các gói tin truyền trên mạng để đánh cắp dữ liệu.
- Tấn công trung gian (Man-in-the-middle): Kẻ tấn công chèn vào giữa hai thiết bị đang giao tiếp để đánh cắp hoặc thay đổi dữ liệu.
- Tấn công replay: Kẻ tấn công chặn 1 luồng truyền tin và truyền lại nó để làm giả 1 hành động đã xảy ra.

2.1.3. Tấn công phần mềm (Software attacks)

Khái niệm:

- Tấn công phần mềm là các hình thức tấn công nhắm vào phần mềm của thiết bị IoT hoặc hệ thống IoT.
- Kẻ tấn công có thể khai thác các lỗ hổng trong phần mềm để cài đặt mã độc hoặc đánh cắp dữ liệu.

Các hình thức tấn công:

- Tấn công tiêm nhiễm mã độc (Malware injection): Kẻ tấn công cài đặt mã độc vào thiết bị IoT để kiểm soát thiết bị hoặc đánh cắp dữ liệu.
- Tấn công SQL injection: Kẻ tấn công tiêm mã SQL độc hại vào ứng dụng web để truy cập trái phép vào cơ sở dữ liệu.
- Tấn công cross-site scripting (XSS): Kẻ tấn công tiêm mã JavaScript độc hại vào trang web để đánh cắp thông tin người dùng.
- Tấn công buffer overflow: Kẻ tấn công khai thác lỗ hổng trong phần mềm để ghi đè dữ liệu lên bộ nhớ, gây ra lỗi hoặc thực thi mã độc.

2.2. Phân tích chi tiết các hình thức tấn công phổ biến

2.2.1. Tấn công từ chối dịch vụ (DoS/DDoS)

Khái niệm:

- Tấn công DoS (Denial of Service) và DDoS (Distributed Denial of Service) là các hình thức tấn công làm gián đoạn dịch vụ của một hệ thống bằng cách làm quá tải tài nguyên của hệ thống đó.
- Trong IoT, các thiết bị có thể bị tấn công để gửi lượng lớn lưu lượng truy cập đến một máy chủ hoặc thiết bị khác, khiến hệ thống không thể hoạt động bình thường.

Cách thức hoạt động:

- Trong tấn công DoS, kẻ tấn công sử dụng một máy tính để tấn công mục tiêu.
- Kẻ tấn công xây dựng một botnet bằng cách lây nhiễm mã độc vào nhiều máy tính, từ đó điều khiển botnet để gửi một lượng lớn lưu lượng truy cập đến máy chủ mục tiêu. Lưu lượng truy cập quá lớn làm quá tải tài nguyên của máy chủ bị tấng công, khiến máy chủ đó không thể xử lý các yêu cầu hợp lệ.

Hậu quả:

- Gián đoạn dịch vụ, khiến người dùng không thể truy cập vào hệ thống.
- Làm chậm hoặc tê liệt hệ thống, gây thiệt hại về kinh tế và uy tín.
- Tạo điều kiện cho các cuộc tấn công khác.

Ví dụ:

- Tấn công vào các thiết bị camera IP để tạo ra lưu lượng truy cập lớn, làm gián đoạn dịch vụ giám sát.
- Tấn công vào các thiết bị điều khiển công nghiệp để làm gián đoạn quá trình sản xuất.
- Tấn công vào các thiết bị nhà thông minh.

Các loại phổ biến:

- Volume-based attacks: Loại tấn công này làm tràn ngập băng thông mạng bằng lưu lượng lớn.
- Protocol attacks: Loại tấn công này khai thác các lỗ hổng trong giao thức mạng.
- Application attacks: Loại tấn công này nhắm vào các ứng dụng web cụ thể.

2.2.2. Tấn công giả mạo (Spoofing)

Khái niệm:

- Tấn công giả mạo là hình thức tấn công mà kẻ tấn công giả mạo danh tính của một thiết bị hoặc người dùng khác để truy cập trái phép vào hệ thống.
- Trong IoT, kẻ tấn công có thể giả mạo địa chỉ IP, MAC hoặc các thông tin khác để đánh lừa hệ thống.

Cách thức hoạt động:

- Kẻ tấn công sử dụng các công cụ để giả mạo thông tin nhận dạng.
- Hệ thống mục tiêu tin rằng kẻ tấn công là một thiết bị hoặc người dùng hợp lệ.

Hậu quả:

- Truy cập trái phép vào hệ thống, đánh cắp hoặc thay đổi dữ liệu.
- Gây ra các hành động trái phép, chẳng hạn như điều khiển thiết bị từ xa.
- Làm gián đoạn hoạt động của hệ thống.

Ví dụ:

- Giả mạo địa chỉ IP của một cảm biến để gửi dữ liệu giả mạo đến hệ thống điều khiển.
- Giả mạo thiết bị điều khiển để điều khiển các thiết bị khác trong mạng.

Các loại phổ biến:

- IP spoofing: Giả mạo địa chỉ IP nguồn để ẩn danh hoặc để thực hiện các cuộc tấn công.
- ARP spoofing: Giả mạo các gói tin ARP để đánh cắp dữ liệu hoặc để thực hiện các cuộc tấn công trung gian.
- DNS spoofing: Giả mạo các bản ghi DNS để chuyển hướng người dùng đến các trang web độc hại.

2.2.3. Tấn công nghe lén (Eavesdropping)

Khái niệm:

- Tấn công nghe lén là hình thức tấn công mà kẻ tấn công chặn và ghi lại các thông tin truyền qua mạng.
- Trong IoT, kẻ tấn công có thể nghe lén các thông tin nhạy cảm như mật khẩu, dữ liệu cá nhân hoặc dữ liệu điều khiển.

Cách thức hoạt động:

- Kẻ tấn công sử dụng các công cụ để chặn và ghi lại lưu lượng mạng.
- Kẻ tấn công phân tích dữ liệu ghi lại để tìm kiếm thông tin nhạy cảm.

Hậu quả:

- Đánh cắp thông tin nhạy cảm, gây thiệt hại về tài chính và uy tín.
- Lộ lọt thông tin bí mật, gây ảnh hưởng đến an ninh quốc gia.
- Cho phép kẻ tấn công thực hiện các cuộc tấn công khác.

Ví dụ:

- Nghe lén các thông tin truyền từ cảm biến đến hệ thống điều khiển.
- Nghe lén các thông tin đăng nhập vào hệ thống quản lý IoT.

Các loại phổ biến:

- Passive eavesdropping: Kẻ tấn công chỉ nghe lén lưu lượng mạng mà không can thiệp vào.
- Active eavesdropping: Kẻ tấn công can thiệp vào lưu lượng mạng để thay đổi hoặc chèn dữ liệu.

2.2.4. Tấn công trung gian (Man-in-the-middle)

Khái niệm:

- Tấn công trung gian là hình thức tấn công mà kẻ tấn công chèn vào giữa hai thiết bị đang giao tiếp để đánh cắp hoặc thay đổi dữ liệu.
- Trong IoT, kẻ tấn công có thể chèn vào giữa thiết bị IoT và hệ thống điều khiển để đánh cắp hoặc thay đổi dữ liệu điều khiển.

Cách thức hoạt động:

- Kẻ tấn công đánh lừa hai thiết bị để chúng tin rằng kẻ tấn công là thiết bị còn lại.
- Kẻ tấn công chặn và thay đổi dữ liệu truyền giữa hai thiết bị.

Hậu quả:

- Đánh cắp thông tin nhạy cảm, thay đổi dữ liệu điều khiển.
- Gây ra các hành động trái phép, chẳng hạn như điều khiển thiết bị từ xa.
- Làm gián đoạn hoạt động của hệ thống.

Ví dụ:

- Chèn vào giữa thiết bị điều khiển và thiết bị chấp hành để thay đổi lệnh điều khiển.
- Chèn vào giữa cảm biến và hệ thống điều khiển để thay đổi dữ liệu cảm biến.

Các loại phổ biến:

- ARP poisoning: Kẻ tấn công gửi các gói tin ARP giả mạo để đánh lừa các thiết bị trên mạng.
- DNS spoofing: Kẻ tấn công giả mạo các bản ghi DNS để chuyển hướng người dùng đến các trang web độc hại.
- HTTPS spoofing: Kẻ tấn công giả mạo chứng chỉ HTTPS để đánh cắp dữ liệu từ các trang web được mã hóa.

2.2.5. Tấn công tiêm nhiễm mã độc (Malware injection)

Khái niệm:

- Tấn công tiêm nhiễm mã độc là hình thức tấn công mà kẻ tấn công cài đặt mã độc vào thiết bị IoT để kiểm soát thiết bị hoặc đánh cắp dữ liệu.
- Trong IoT, các thiết bị có thể bị nhiễm mã độc thông qua các lỗ hổng phần mềm hoặc phần cứng.

Cách thức hoạt động:

- Kẻ tấn công khai thác các lỗ hổng để cài đặt mã độc.
- Mã độc kiểm soát thiết bị hoặc đánh cắp dữ liệu.

Hậu quả:

- Kiểm soát thiết bị từ xa, thực hiện các hành động trái phép.
- Đánh cắp dữ liệu nhạy cảm, gây thiệt hại về tài chính và uy tín.
- Làm gián đoạn hoạt động của hệ thống.

Ví dụ:

- Cài đặt mã độc vào camera IP để theo dõi người dùng.
- Cài đặt mã độc vào thiết bị điều khiển công nghiệp để làm gián đoạn quá trình sản xuất.

Các loại phổ biến:

- Virus: Mã độc lây lan bằng cách gắn vào các tệp thực thi.
- Worms: Mã độc lây lan qua mạng mà không cần sự can thiệp của người dùng.
- Trojan horses: Mã độc ẩn trong các chương trình hợp pháp.
- Ransomware: Mã độc mã hóa dữ liệu và đòi tiền chuộc.
- Spyware: Mã độc thu thập thông tin người dùng mà không có sự đồng ý của họ.

2.3. Đánh giá mức độ nguy hiểm của các hình thức tấn công

2.3.1. Mức độ thấp

Tuy các hình thức nghe lén thụ động có thể được xem xét ở mức độ nguy hiểm thấp nhất nhưng thực chất, không có hình thức tấn công nào được xếp ở mức độ nguy hiểm thấp một cách tuyệt đối trong bối cảnh IoT.

Tấn công nghe lén thụ động (Passive Eavesdropping)

- Không gây gián đoạn dịch vụ: Hoạt động của hệ thống IoT không bị ảnh hưởng trực tiếp.
- Khó phát hiện: Kẻ tấn công không để lại dấu vết rõ ràng trên mạng.
- Nguy cơ tiềm ẩn: Dữ liệu thu thập được có thể được sử dụng cho các cuộc tấn công phức tạp hơn trong tương lai (ví dụ: thu thập thông tin đăng nhập, phân tích giao thức). Tuy nhiên, nguy cơ trực tiếp là thấp.

2.3.2. Mức độ trung bình

Tấn công vật lý (Physical attacks):

- Tấn công side-channel (Side-channel attacks): Khai thác thông tin rò rỉ từ hoạt động thiết bị để suy luận dữ liệu nhạy cảm, khó thực hiện nhưng có thể tiết lộ thông tin quan trọng.
- Tấn công tamper (Tamper attacks): Cố gắng can thiệp vật lý để thu thập thông tin hoặc thay đổi chức năng, mức độ ảnh hưởng phụ thuộc vào thành công của việc can thiệp.

Tấn công mạng (Network attacks):

- Tấn công giả mạo (Spoofing): Mạo danh thiết bị/dịch vụ để đánh lừa hệ thống, có thể dẫn đến truy cập trái phép hoặc dữ liệu sai lệch.
- Tấn công nghe lén (Eavesdropping): Thu thập thông tin truyền qua mạng, nguy cơ tiềm ẩn cho các tấn công khác dù không trực tiếp gây gián đoạn.
- Tấn công replay: Sử dụng lại các gói tin đã ghi lại để thực hiện lại hành động trái phép, khai thác cơ chế xác thực yếu.

Tấn công phần mềm (Software attacks):

- Tấn công cross-site scripting (XSS): Tiêm mã độc vào trang web hiển thị cho người dùng IoT, có thể đánh cắp thông tin hoặc thực hiện hành động thay mặt họ.

2.3.3. Mức độ cao

Tấn công vật lý (Physical attacks):

- Giả mạo thiết bị (Device spoofing): Thay thế thiết bị hợp lệ bằng thiết bị giả mạo, cho phép kiểm soát, gửi dữ liệu sai lệch hoặc tấn công các thiết bị khác.
- Tấn công xâm nhập phần cứng (Hardware tampering): Can thiệp trực tiếp vào phần cứng để trích xuất khóa, cài backdoor hoặc vô hiệu hóa bảo mật, dẫn đến kiểm soát hoàn toàn.

Tấn công mạng (Network attacks):

- Tấn công từ chối dịch vụ (DoS/DDoS): Làm gián đoạn hoạt động của thiết bị/hệ thống, gây tê liệt dịch vụ và ảnh hưởng đến tính sẵn sàng.

- Tấn công trung gian (Man-in-the-middle): Chặn và có khả năng sửa đổi dữ liệu truyền, đánh cắp thông tin nhạy cảm hoặc thay đổi lệnh điều khiển.

Tấn công phần mềm (Software attacks):

- Tấn công tiêm nhiễm mã độc (Malware injection): Cài đặt phần mềm độc hại để kiểm soát thiết bị, đánh cắp dữ liệu hoặc thực hiện các hành động nguy hiểm khác.
- Tấn công SQL injection: Khai thác lỗ hổng trong ứng dụng sử dụng cơ sở dữ liệu SQL để truy cập, sửa đổi hoặc xóa dữ liệu trái phép.
- Tấn công buffer overflow: Khai thác lỗi tràn bộ đệm để thực thi mã độc hoặc gây ra lỗi hệ thống, dẫn đến kiểm soát hoặc gián đoạn.

2.4. Các phương pháp bảo mật dữ liệu ở lớp thiết bị

2.4.1. Sử dụng các cơ chế xác thực mạnh

Khái niệm:

- Xác thực mạnh là quá trình xác minh danh tính của người dùng hoặc thiết bị bằng cách sử dụng nhiều yếu tố xác thực.
- Trong IoT, xác thực mạnh giúp ngăn chặn các truy cập trái phép vào thiết bị và dữ liệu.

Các phương pháp xác thực mạnh:

- Xác thực đa yếu tố (MFA): Yêu cầu người dùng hoặc thiết bị cung cấp nhiều yếu tố xác thực, chẳng hạn như mật khẩu, mã OTP, hoặc dữ liệu sinh trắc học.
- Chứng chỉ số: Sử dụng chứng chỉ số để xác thực danh tính của thiết bị và mã hóa dữ liệu.
- Xác thực dựa trên phần cứng: Sử dụng các module bảo mật phần cứng (HSM) để lưu trữ và quản lý khóa mã hóa.

Ví dụ:

- Sử dụng MFA để truy cập vào thiết bị điều khiển công nghiệp.
- Sử dụng chứng chỉ số để xác thực danh tính của cảm biến trong hệ thống giám sát môi trường.

2.4.2. Mã hóa dữ liệu trên thiết bị

Khái niệm:

- Mã hóa dữ liệu là quá trình chuyển đổi dữ liệu thành một định dạng không thể đọc được nếu không có khóa giải mã.
- Trong IoT, mã hóa dữ liệu trên thiết bị giúp bảo vệ dữ liệu khỏi bị đánh cắp hoặc truy cập trái phép.

Các phương pháp mã hóa dữ liệu:

- Mã hóa đối xứng: Sử dụng cùng một khóa để mã hóa và giải mã dữ liệu.
- Mã hóa bất đối xứng: Sử dụng một cặp khóa1 công khai và khóa riêng tư để mã hóa và giải mã dữ liệu.
- Mã hóa dữ liệu trong bộ nhớ: Mã hóa dữ liệu khi nó được lưu trữ trong bộ nhớ của thiết bị.

Ví dụ:

- Mã hóa dữ liệu cảm biến trước khi truyền đến hệ thống điều khiển.
- Mã hóa dữ liệu cấu hình của thiết bị để ngăn chặn các thay đổi trái phép.

2.4.3. Cập nhật phần mềm và firmware thường xuyên

Khái niệm:

- Cập nhật phần mềm và firmware là quá trình cài đặt các bản vá lỗi và bản cập nhật bảo mật cho thiết bị IoT.
- Việc cập nhật thường xuyên giúp vá các lỗ hổng bảo mật và ngăn chặn các cuộc tấn công.

Các phương pháp cập nhật phần mềm và firmware:

- Cập nhật tự động: Cho phép thiết bị tự động tải xuống và cài đặt các bản cập nhật.
- Cập nhật từ xa: Cho phép quản trị viên cập nhật phần mềm và firmware của thiết bị từ xa.
- Cập nhật an toàn: Đảm bảo rằng các bản cập nhật được xác thực và không bị giả mạo.

Ví dụ:

- Cập nhật firmware cho camera IP để vá các lỗ hổng bảo mật.

- Cập nhật phần mềm cho thiết bị điều khiển công nghiệp để cải thiện hiệu suất và bảo mật.

2.5. Các phương pháp bảo mật dữ liệu ở lớp mạng

2.5.1. Sử dụng tường lửa và hệ thống phát hiện xâm nhập (IDS/IPS)

Tường lửa (Firewall):

- Khái niệm: Tường lửa là một hệ thống bảo mật mạng giám sát và kiểm soát lưu lượng mạng dựa trên các quy tắc bảo mật được xác định trước.
- Cách thức hoạt động: Tường lửa lọc lưu lượng mạng dựa trên địa chỉ IP nguồn và đích, cổng, giao thức và các tiêu chí khác.
- Vai trò: Tường lửa giúp ngăn chặn các truy cập trái phép vào mạng IoT, bảo vệ các thiết bị IoT khỏi các cuộc tấn công từ bên ngoài.

Hệ thống phát hiện xâm nhập (IDS) và Hệ thống ngăn chặn xâm nhập (IPS):

- Khái niệm: IDS và IPS là các hệ thống bảo mật mạng giám sát lưu lượng mạng để phát hiện các hoạt động đáng ngờ hoặc độc hại.
- Cách thức hoạt động: IDS phát hiện các hoạt động đáng ngờ và cảnh báo cho quản trị viên, trong khi IPS có thể tự động ngăn chặn các hoạt động này.
- Vai trò: IDS/IPS giúp phát hiện và ngăn chặn các cuộc tấn công mạng nhằm vào hệ thống IoT, bảo vệ dữ liệu và thiết bị.

2.5.2. Triển khai mạng riêng ảo (VPN)

Khái niệm:

- VPN (Virtual Private Network) là một mạng riêng ảo được xây dựng trên một mạng công cộng, chẳng hạn như Internet.
- VPN mã hóa lưu lượng mạng, tạo ra một đường hầm an toàn để truyền dữ liệu.

Cách thức hoạt động:

- VPN thiết lập một kết nối an toàn giữa thiết bị IoT và máy chủ VPN.
- Tất cả lưu lượng mạng giữa thiết bị IoT và máy chủ VPN đều được mã hóa.

Vai trò:

- VPN giúp bảo vệ dữ liệu truyền qua mạng công cộng, ngăn chặn các cuộc tấn công nghe lén và trung gian.
- VPN cho phép truy cập từ xa an toàn vào mạng IoT.

2.5.3. Sử dụng các giao thức bảo mật (TLS/SSL, IPSec)

TLS/SSL (Transport Layer Security/Secure Sockets Layer):

- Khái niệm: TLS/SSL là các giao thức bảo mật được sử dụng để mã hóa lưu lượng mạng ở lớp giao vận.
- Cách thức hoạt động: TLS/SSL thiết lập một kết nối an toàn giữa hai thiết bị và mã hóa dữ liệu truyền qua kết nối đó.
- Vai trò: TLS/SSL giúp bảo vệ dữ liệu truyền qua Internet, chẳng hạn như dữ liệu từ cảm biến hoặc dữ liệu điều khiển.

IPSec (Internet Protocol Security):

- Khái niệm: IPSec là một bộ giao thức bảo mật được sử dụng để bảo vệ lưu lượng mạng ở lớp mạng.
- Cách thức hoạt động: IPSec mã hóa và xác thực các gói tin IP.
- Vai trò: IPSec giúp bảo vệ dữ liệu truyền qua mạng IP, chẳng hạn như dữ liệu từ thiết bị IoT đến máy chủ.

2.6. Các phương pháp bảo mật dữ liệu ở lớp ứng dụng

2.6.1. Kiểm soát truy cập dựa trên vai trò (RBAC)

Khái niệm:

- RBAC (Role-Based Access Control) là một phương pháp kiểm soát truy cập dựa trên vai trò của người dùng trong hệ thống.
- Mỗi vai trò được gán một tập hợp các quyền truy cập vào các tài nguyên của hệ thống.

Cách thức hoạt động:

- Người dùng được gán cho một hoặc nhiều vai trò.
- Hệ thống kiểm tra vai trò của người dùng để xác định quyền truy cập của họ.
- Người dùng chỉ có thể truy cập vào các tài nguyên mà vai trò của họ cho phép.

Vai trò:

- RBAC giúp quản lý quyền truy cập vào các ứng dụng và dịch vụ IoT một cách hiệu quả.
- RBAC giúp ngăn chặn các truy cập trái phép và bảo vệ dữ liệu nhạy cảm.

Ví dụ:

- Trong hệ thống nhà thông minh, người dùng có thể được gán các vai trò như "chủ nhà", "khách", hoặc "người quản lý" với các quyền truy cập khác nhau.
- Trong hệ thống công nghiệp, người dùng có thể được gán các vai trò như "kỹ sư", "nhân viên vận hành", hoặc "quản lý" với các quyền truy cập khác nhau.

2.6.2. Mã hóa dữ liệu đầu cuối (End-to-end encryption)

Khái niệm:

- Mã hóa dữ liệu đầu cuối là một phương pháp mã hóa dữ liệu sao cho chỉ người gửi và người nhận mới có thể đọc được dữ liệu.
- Dữ liệu được mã hóa trên thiết bị của người gửi và chỉ được giải mã trên thiết bị của người nhận.

Cách thức hoạt động:

- Người gửi và người nhận sử dụng một cặp khóa công khai và khóa riêng tư để mã hóa và giải mã dữ liệu.
- Dữ liệu được mã hóa bằng khóa công khai của người nhận và chỉ có thể được giải mã bằng khóa riêng tư của người nhận.

Vai trò:

- Mã hóa dữ liệu đầu cuối giúp bảo vệ dữ liệu nhạy cảm khỏi bị đánh cắp hoặc truy cập trái phép.
- Mã hóa dữ liệu đầu cuối giúp đảm bảo tính riêng tư của người dùng.

Ví dụ:

- Mã hóa dữ liệu từ cảm biến sức khỏe trước khi truyền đến ứng dụng y tế.
- Mã hóa dữ liệu từ camera an ninh trước khi lưu trữ trên đám mây.

2.6.3. Sử dụng các kỹ thuật phân quyền (Authorization)

Khái niệm:

- Phân quyền là quá trình xác định quyền truy cập của người dùng hoặc thiết bị vào các tài nguyên của hệ thống.
- Phân quyền khác với xác thực (authentication), xác thực là quá trình xác minh danh tính của người dùng hoặc thiết bị.

Các kỹ thuật phân quyền:

- OAuth: Một giao thức phân quyền phổ biến cho phép các ứng dụng của bên thứ ba truy cập vào các tài nguyên của người dùng trên một máy chủ khác.
- OpenID Connect: Một giao thức xác thực và phân quyền dựa trên OAuth 2.0.
- JSON Web Tokens (JWT): Một tiêu chuẩn mở để tạo ra các mã thông báo truy cập (access tokens) được mã hóa.

Vai trò:

- Phân quyền giúp kiểm soát quyền truy cập vào các ứng dụng và dịch vụ IoT.
- Phân quyền giúp đảm bảo rằng chỉ những người dùng hoặc thiết bị được ủy quyền mới có thể truy cập vào các tài nguyên nhạy cảm.

Ví dụ:

- Sử dụng OAuth để cho phép ứng dụng của bên thứ ba truy cập vào dữ liệu từ thiết bị nhà thông minh.
- Sử dụng JWT để xác thực và phân quyền truy cập vào các API IoT.

2.7. Nghiên cứu các thuật toán bảo mật dữ liệu tiên tiến cho hệ thống IoT

2.7.1. Thuật toán mã hóa khóa công khai (Public-key cryptography)

Khái niệm:

- Mã hóa khóa công khai là một hệ thống mã hóa sử dụng một cặp khóa: khóa công khai và khóa riêng tư.
- Khóa công khai được chia sẻ rộng rãi, trong khi khóa riêng tư được giữ bí mật.
- Dữ liệu được mã hóa bằng khóa công khai và chỉ có thể được giải mã bằng khóa riêng tư tương ứng.

Cấu trúc:

- Tạo khóa (Key Generation):
  - Tạo ra một cặp khóa liên quan về mặt toán học:
    - Khóa công khai (Public Key): Được chia sẻ rộng rãi và dùng để mã hóa dữ liệu hoặc xác minh chữ ký số.
    - Khóa riêng tư (Private Key): Được giữ bí mật bởi chủ sở hữu và dùng để giải mã dữ liệu đã được mã hóa bằng khóa công khai tương ứng hoặc tạo chữ ký số.
- Mã hóa (Encryption):
  - Một thuật toán (hàm toán học) sử dụng khóa công khai của người nhận để chuyển đổi dữ liệu rõ (plaintext) thành dữ liệu đã mã hóa (ciphertext).
  - Chỉ người nắm giữ khóa riêng tư tương ứng mới có thể giải mã ciphertext trở lại thành plaintext một cách hiệu quả.
- Giải mã (Decryption): Một thuật toán sử dụng khóa riêng tư để chuyển đổi ciphertext trở lại thành plaintext ban đầu.

Các thuật toán cụ thể: RSA, ECC (Elliptic Curve Cryptography), DSA (Digital Signature Algorithm). Mỗi thuật toán có cấu trúc toán học riêng cho việc tạo khóa, mã hóa và giải mã.

Cách thức hoạt động:

- Người gửi sử dụng khóa công khai của người nhận để mã hóa dữ liệu.
- Người nhận sử dụng khóa riêng tư của mình để giải mã dữ liệu.
- Các thuật toán phổ biến: RSA, ECC (Elliptic Curve Cryptography).

Vai trò:

- Xác thực thiết bị: Xác thực danh tính của các thiết bị IoT.
- Mã hóa dữ liệu: Bảo vệ dữ liệu nhạy cảm truyền qua mạng IoT.
- Quản lý khóa: Phân phối và quản lý khóa mã hóa.

Ưu điểm:

- Bảo mật cao: Khó bị phá vỡ.
- Quản lý khóa linh hoạt.

Nhược điểm:

- Tốc độ xử lý chậm hơn mã hóa đối xứng.
- Yêu cầu tài nguyên tính toán lớn hơn.

Ngôn ngữ lập trình:

- C/C++: Thường được sử dụng cho các thư viện mã hóa hiệu suất cao, đặc biệt quan trọng trên các thiết bị IoT có tài nguyên hạn chế. Các thư viện như OpenSSL, mbed TLS được viết bằng C/C++ và cung cấp nhiều thuật toán khóa công khai (RSA, ECC, DSA, etc.).
- Python: Phổ biến cho các ứng dụng phía máy chủ, công cụ phát triển và nghiên cứu bảo mật. Các thư viện như PyCryptodome, cryptography cung cấp các triển khai dễ sử dụng của các thuật toán khóa công khai.
- Java: Được sử dụng trong các hệ thống doanh nghiệp và nền tảng Android. Java Cryptography Architecture (JCA) cung cấp framework cho việc sử dụng các thuật toán mã hóa.
- Go: Ngày càng phổ biến cho các ứng dụng mạng và backend, có các thư viện mã hóa tích hợp và bên thứ ba tốt.
- JavaScript: Sử dụng trong các ứng dụng web và một số nền tảng IoT (Node.js). Các thư viện như crypto (trong Node.js) và các thư viện bên thứ ba cung cấp khả năng mã hóa khóa công khai.

Ứng dụng trong IoT:

- Xác thực thiết bị: Xác thực danh tính của các thiết bị khi chúng kết nối với nhau hoặc với máy chủ đám mây.
- Mã hóa dữ liệu truyền: Bảo vệ dữ liệu nhạy cảm (ví dụ: dữ liệu cảm biến, thông tin điều khiển) trong quá trình truyền tải giữa thiết bị và máy chủ hoặc giữa các thiết bị.
- Quản lý khóa: Thiết lập các kênh an toàn để trao đổi và quản lý khóa mã hóa.
- Cập nhật firmware an toàn: Đảm bảo tính toàn vẹn và xác thực của các bản cập nhật phần mềm.

Ví dụ: Giao tiếp an toàn giữa cảm biến nhiệt độ và máy chủ đám mây:

- Máy chủ đám mây tạo ra một cặp khóa: khóa công khai (public key) Ks_pub và khóa riêng tư (private key) Ks_priv.
- Máy chủ đám mây gửi khóa công khai Ks_pub đến cảm biến nhiệt độ (thông qua kênh không an toàn).
- Khi cảm biến nhiệt độ muốn gửi dữ liệu nhiệt độ (ví dụ: giá trị 25°C) đến máy chủ, nó sẽ mã hóa dữ liệu này bằng khóa công khai Ks_pub của máy chủ. Dữ liệu đã mã hóa trở thành một chuỗi ký tự vô nghĩa (ví dụ: "a1b2c3d4e5f6").
- Cảm biến gửi dữ liệu đã mã hóa đến máy chủ đám mây.
- Máy chủ đám mây, vì là người duy nhất giữ khóa riêng tư Ks_priv tương ứng, có thể giải mã dữ liệu đã mã hóa để khôi phục lại giá trị nhiệt độ ban đầu là 25°C.

Ý nghĩa: Ngay cả khi kẻ tấn công chặn được dữ liệu đã mã hóa, chúng không thể đọc được nội dung vì không có khóa riêng tư.

2.7.2. Thuật toán chữ ký số (Digital signature)

Khái niệm:

- Chữ ký số là một kỹ thuật mã hóa được sử dụng để xác thực tính xác thực và tính toàn vẹn của dữ liệu.
- Chữ ký số được tạo ra bằng cách sử dụng khóa riêng tư của người gửi và được xác minh bằng khóa công khai của người gửi.

Cấu trúc:

- Tạo khóa (Key Generation): Tương tự như mã hóa khóa công khai, tạo ra cặp khóa riêng tư (để ký) và khóa công khai (để xác minh).
- Ký (Signing):
  - Một thuật toán sử dụng khóa riêng tư của người gửi để tạo ra một đoạn mã ngắn (chữ ký số) dựa trên nội dung của dữ liệu (thường là một bản tóm tắt - hash - của dữ liệu).
  - Chữ ký số này được gắn kèm hoặc truyền cùng với dữ liệu.
- Xác minh (Verification):
  - Một thuật toán sử dụng khóa công khai của người gửi để kiểm tra tính hợp lệ của chữ ký số đối với dữ liệu đã nhận.
  - Nếu chữ ký hợp lệ, nó chứng minh được hai điều:
    - Tính xác thực (Authenticity): Dữ liệu thực sự đến từ người sở hữu khóa riêng tư tương ứng.
    - Tính toàn vẹn (Integrity): Dữ liệu không bị thay đổi kể từ khi được ký.

Các thuật toán cụ thể (ví dụ): RSA-PSS, ECDSA (Elliptic Curve Digital Signature Algorithm), DSA.

Cách thức hoạt động:

- Người gửi tạo chữ ký số bằng cách sử dụng khóa riêng tư của mình.
- Người nhận xác minh chữ ký số bằng cách sử dụng khóa công khai của người gửi.

Vai trò:

- Xác thực nguồn gốc dữ liệu: Xác định nguồn gốc của dữ liệu IoT.
- Đảm bảo tính toàn vẹn dữ liệu: Ngăn chặn việc thay đổi dữ liệu trái phép.
- Chống từ chối (non-repudiation): Ngăn chặn người gửi phủ nhận việc gửi dữ liệu.

Ưu điểm:

- Tính xác thực cao.
- Tính toàn vẹn dữ liệu.

Nhược điểm:

- Yêu cầu quản lý khóa cẩn thận.

Ngôn ngữ lập trình: Tương tự như mã hóa khóa công khai, các ngôn ngữ C/C++, Python, Java, Go và JavaScript đều được sử dụng để triển khai chữ ký số thông qua các thư viện mã hóa tương ứng (ví dụ: OpenSSL, PyCryptodome, JCA, các thư viện Go crypto, Node.js crypto).

Ứng dụng trong IoT:

- Xác thực nguồn gốc dữ liệu: Chứng minh rằng dữ liệu đến từ một nguồn đáng tin cậy (ví dụ: cảm biến chính hãng, máy chủ ủy quyền).
- Đảm bảo tính toàn vẹn dữ liệu: Phát hiện nếu dữ liệu đã bị thay đổi trong quá trình truyền tải hoặc lưu trữ.
- Cập nhật firmware an toàn: Xác minh tính xác thực và toàn vẹn của các bản cập nhật phần mềm.
- Giao dịch an toàn giữa các thiết bị: Xác thực các lệnh và dữ liệu trao đổi giữa các thiết bị IoT trong mạng ngang hàng.

Ví dụ: Đảm bảo tính xác thực của bản cập nhật firmware cho thiết bị camera an ninh:

- Nhà sản xuất camera an ninh tạo ra một cặp khóa: khóa riêng tư để ký (signing private key) Kn_priv và khóa công khai để xác minh (verification public key) Kn_pub.
- Khóa công khai Kn_pub được tích hợp sẵn vào firmware gốc của camera hoặc được cung cấp cho người dùng.
- Khi nhà sản xuất phát hành bản cập nhật firmware mới, họ sẽ sử dụng khóa riêng tư Kn_priv để tạo ra một chữ ký số cho bản cập nhật đó. Chữ ký số là một đoạn mã nhỏ được tạo ra dựa trên nội dung của bản cập nhật.
- Bản cập nhật firmware và chữ ký số đi kèm được gửi đến camera an ninh.
- Camera an ninh sử dụng khóa công khai Kn_pub đã được lưu trữ để xác minh chữ ký số của bản cập nhật. Nếu chữ ký số hợp lệ (chứng minh được tạo ra bởi khóa riêng tư tương ứng và nội dung bản cập nhật không bị thay đổi), camera sẽ tiến hành cài đặt bản cập nhật. Nếu chữ ký không hợp lệ, camera sẽ từ chối cài đặt.

Ý nghĩa: Người dùng có thể chắc chắn rằng bản cập nhật firmware đến từ nhà sản xuất chính thức và không bị can thiệp bởi kẻ xấu.

2.7.3. Thuật toán blockchain

Khái niệm:

- Blockchain (chuỗi khối) là một sổ cái kỹ thuật số phân tán, bất biến và công khai (hoặc riêng tư có kiểm soát), được tạo thành từ các khối (blocks) dữ liệu được liên kết với nhau theo trình tự thời gian bằng mã hóa.
- Vì là một sổ cái phân tán, bất biến, nên blockchain thường được sử dụng để ghi lại các giao dịch. Dữ liệu được lưu trữ trong các khối và được liên kết với nhau bằng các hàm băm.
- Thuật toán Blockchain thực chất là tập hợp các quy tắc và giao thức chi phối cách thức hoạt động của một mạng blockchain. Nó bao gồm nhiều khía cạnh, nhưng trọng tâm chính là cách các giao dịch được xác minh, đóng gói thành khối, liên kết với chuỗi hiện có và đạt được sự đồng thuận.

Cấu trúc:

- Khối (Block):
  - Dữ liệu (Data): Chứa các giao dịch hoặc thông tin cần được ghi lại.
  - Mã băm của khối trước (Previous Hash): Một chuỗi ký tự duy nhất là kết quả của việc băm (hàm băm mã hóa) nội dung của khối trước đó trong chuỗi. Liên kết các khối lại với nhau theo trình tự thời gian.
  - Mã băm của khối hiện tại (Current Hash): Một chuỗi ký tự duy nhất được tạo ra bằng cách băm nội dung của khối hiện tại (bao gồm dữ liệu, mã băm của khối trước và các thông tin khác như nonce). Mã băm này dùng để xác định khối này và được tham chiếu bởi khối tiếp theo.
  - Dấu thời gian (Timestamp): Ghi lại thời điểm khối được tạo.
  - Nonce: Một số ngẫu nhiên hoặc bán ngẫu nhiên được sử dụng trong cơ chế đồng thuận Proof-of-Work để tìm ra một mã băm hợp lệ cho khối.
- Chuỗi (Chain): Một chuỗi các khối được liên kết với nhau theo trình tự thời gian, với mỗi khối chứa mã băm của khối trước đó.
- Mạng lưới phân tán (Distributed Network): Dữ liệu blockchain được lưu trữ và duy trì bởi nhiều nút mạng độc lập.

- Cơ chế đồng thuận (Consensus Mechanism): Các quy tắc mà các nút mạng tuân theo để đạt được sự đồng thuận về tính hợp lệ của các giao dịch và việc thêm khối mới vào chuỗi (ví dụ: Proof-of-Work, Proof-of-Stake).

Các thuật toán cụ thể: SHA-256, Keccak-256, RIPEMD-160, Blake2b, ECDSA, EdDSA, RSA, DSA, Proof-of-Work (PoW), Proof-of-Stake (PoS), Proof-of-Authority (PoA), Delegated Proof-of-Stake (DPoS), Proof-of-Elapsed-Time (PoET), Paxos, Raft, Merkle Tree, Thuật toán quản lý UTXO, Thuật toán quản lý tài khoản, EVM (Ethereum Virtual Machine)...

Cách thức hoạt động:

- Các giao dịch được nhóm lại thành các khối.
- Các khối được liên kết với nhau bằng các hàm băm.
- Dữ liệu trong blockchain là bất biến, không thể bị thay đổi.

Vai trò:

- Bảo mật dữ liệu: Đảm bảo tính toàn vẹn và tính xác thực của dữ liệu IoT.
- Quản lý thiết bị: Theo dõi và quản lý các thiết bị IoT.
- Chia sẻ dữ liệu: Chia sẻ dữ liệu IoT một cách an toàn và minh bạch.

Ưu điểm:

- Tính bảo mật cao.
- Tính minh bạch.
- Tính phân tán.

Nhược điểm:

- Tốc độ xử lý chậm.
- Yêu cầu tài nguyên tính toán lớn.

Ngôn ngữ lập trình:

- Go: Rất phổ biến để xây dựng các nền tảng blockchain hiệu suất cao (ví dụ: Hyperledger Fabric, Go-Ethereum).
- C++: Được sử dụng cho một số blockchain nền tảng (ví dụ: Bitcoin, EOS) vì hiệu suất và khả năng kiểm soát bộ nhớ.

- Java: Sử dụng trong một số nền tảng blockchain doanh nghiệp (ví dụ: Hyperledger Fabric).
- Python: Thường được sử dụng cho các proof-of-concept, công cụ phát triển và tương tác với blockchain (ví dụ: Web3.py cho Ethereum).
- JavaScript (Node.js): Sử dụng cho các ứng dụng web tương tác với blockchain và một số blockchain nền tảng.
- Solidity: Ngôn ngữ đặc biệt được thiết kế để viết smart contract trên nền tảng Ethereum.

Ứng dụng trong IoT:

- Quản lý danh tính thiết bị: Tạo một hệ thống danh tính phi tập trung và an toàn cho các thiết bị IoT.
- Chia sẻ dữ liệu an toàn và minh bạch: Cho phép các thiết bị và tổ chức chia sẻ dữ liệu một cách đáng tin cậy mà không cần bên trung gian.
- Theo dõi chuỗi cung ứng: Ghi lại lịch sử và trạng thái của hàng hóa trong suốt quá trình vận chuyển.
- Quản lý giao dịch vi mô (Microtransactions): Hỗ trợ các giao dịch nhỏ giữa các thiết bị (ví dụ: thanh toán cho việc sử dụng tài nguyên).
- Bảo mật cập nhật phần mềm: Đảm bảo tính toàn vẹn và nguồn gốc của các bản cập nhật.

Ví dụ: Theo dõi lịch sử nhiệt độ của lô hàng dược phẩm nhạy cảm trong quá trình vận chuyển:

- Mỗi khi cảm biến nhiệt độ trên thùng hàng dược phẩm ghi nhận một giá trị nhiệt độ mới, thông tin này (thời gian, vị trí, nhiệt độ) được thêm vào một "khối" dữ liệu.
- Khối này được mã hóa và liên kết với khối dữ liệu trước đó bằng một "mã băm" (hash). Mã băm đảm bảo rằng nếu bất kỳ dữ liệu nào trong khối trước bị thay đổi, mã băm của khối hiện tại cũng sẽ thay đổi.
- Khối mới này được phân tán và lưu trữ trên nhiều thiết bị (ví dụ: máy chủ của các bên liên quan trong chuỗi cung ứng).
- Bất kỳ ai có quyền truy cập vào blockchain đều có thể xem lịch sử nhiệt độ của lô hàng một cách minh bạch và không thể giả mạo. Việc thay đổi bất kỳ dữ liệu nào

trong một khối sẽ làm cho các khối sau đó trở nên không hợp lệ do mã băm không khớp.

Ý nghĩa: Đảm bảo tính toàn vẹn và không thể chối bỏ của dữ liệu theo dõi, tăng cường độ tin cậy trong chuỗi cung ứng.

2.7.4. Thuật toán học máy (Machine learning) trong bảo mật IoT

Khái niệm:

- Học máy là một lĩnh vực của trí tuệ nhân tạo cho phép máy tính học hỏi từ dữ liệu mà không cần được lập trình rõ ràng.
- Học máy có thể được sử dụng để phát hiện các cuộc tấn công mạng và các hoạt động bất thường trong hệ thống IoT.

Cấu trúc:

- Thu thập dữ liệu (Data Collection): Thu thập các dữ liệu liên quan đến hoạt động của thiết bị IoT, lưu lượng mạng, nhật ký hệ thống, v.v. (có thể là dữ liệu bình thường và dữ liệu tấn công đã biết).
- Tiền xử lý dữ liệu (Data Preprocessing): Làm sạch, chuyển đổi và chuẩn hóa dữ liệu để phù hợp với thuật toán học máy.
- Lựa chọn đặc trưng (Feature Selection/Engineering): Xác định và trích xuất các đặc trưng quan trọng từ dữ liệu có khả năng phân biệt giữa hành vi bình thường và bất thường/tấn công.
- Xây dựng mô hình (Model Building): Chọn và huấn luyện một thuật toán học máy (ví dụ: phân loại, hồi quy, gom cụm, phát hiện dị thường) trên dữ liệu đã được chuẩn bị.
- Đánh giá mô hình (Model Evaluation): Đánh giá hiệu suất của mô hình trên dữ liệu kiểm tra để đảm bảo khả năng dự đoán chính xác.
- Triển khai mô hình (Model Deployment): Tích hợp mô hình đã huấn luyện vào hệ thống bảo mật IoT để thực hiện việc phát hiện hoặc dự đoán trong thời gian thực.
- Giám sát và cập nhật mô hình (Model Monitoring and Update): Theo dõi hiệu suất của mô hình theo thời gian và cập nhật/huấn luyện lại mô hình khi cần thiết để duy trì độ chính xác và khả năng thích ứng với các mối đe dọa mới.

Các thuật toán cụ thể (ví dụ):

- Phân loại (Classification): Logistic Regression, Support Vector Machines (SVM), Decision Trees, Random Forests, Neural Networks (để phân loại lưu lượng là bình thường hay tấn công).
- Phát hiện dị thường (Anomaly Detection): Isolation Forest, One-Class SVM, Autoencoders (để xác định các hành vi khác biệt so với mẫu bình thường).
- Gom cụm (Clustering): K-Means, DBSCAN (để nhóm các hành vi tương tự và xác định các cụm bất thường).

Cách thức hoạt động:

- Huấn luyện mô hình học máy bằng dữ liệu lịch sử.
- Sử dụng mô hình để phát hiện các cuộc tấn công mạng và các hoạt động bất thường.

Vai trò:

- Phát hiện xâm nhập: Phát hiện các cuộc tấn công mạng trong thời gian thực.
- Phân tích hành vi: Phát hiện các hoạt động bất thường của thiết bị IoT.
- Dự đoán các cuộc tấn công: Dự đoán các cuộc tấn công mạng trong tương lai.

Ưu điểm:

- Khả năng phát hiện các cuộc tấn công mới.
- Khả năng tự động hóa.

Nhược điểm:

- Yêu cầu dữ liệu lớn để huấn luyện mô hình.
- Khả năng bị đánh lừa bởi các cuộc tấn công đối nghịch.

Ngôn ngữ lập trình:

- Python: Là ngôn ngữ thống trị trong lĩnh vực học máy do có các thư viện mạnh mẽ như TensorFlow, PyTorch, scikit-learn, Keras.
- Java: Được sử dụng trong một số nền tảng học máy (ví dụ: Weka, Deeplearning4j).
- C++: Sử dụng cho các triển khai học máy hiệu suất cao, đặc biệt trên các thiết bị nhúng có tài nguyên hạn chế (thường là các framework được tối ưu hóa).

Ứng dụng trong IoT:

- Phát hiện xâm nhập (Intrusion Detection): Phân tích lưu lượng mạng và hành vi thiết bị để phát hiện các hoạt động đáng ngờ hoặc tấn công.
- Phân tích hành vi bất thường (Anomaly Detection): Xác định các mẫu hành vi khác biệt so với hoạt động bình thường của thiết bị, có thể chỉ ra sự xâm nhập hoặc lỗi.
- Dự đoán lỗi thiết bị: Phân tích dữ liệu cảm biến để dự đoán các lỗi phần cứng tiềm ẩn.
- Quản lý truy cập thích ứng: Điều chỉnh quyền truy cập dựa trên hành vi và ngữ cảnh của người dùng hoặc thiết bị.
- Phân tích nhật ký bảo mật: Tự động hóa việc phân tích lượng lớn nhật ký để phát hiện các mối đe dọa tiềm ẩn.

Ví dụ: Phát hiện hành vi bất thường của thiết bị đèn thông minh bị tấn công:

- Hệ thống thu thập dữ liệu về hoạt động bình thường của đèn thông minh trong một khoảng thời gian (ví dụ: thời gian bật/tắt, mức độ sáng, tần suất kết nối mạng).
- Một thuật toán học máy (ví dụ: thuật toán phát hiện dị thường dựa trên thống kê hoặc mạng nơ-ron) được huấn luyện trên dữ liệu này để xây dựng một "hồ sơ" về hành vi bình thường của đèn.
- Trong quá trình hoạt động thực tế, hệ thống liên tục theo dõi hành vi của đèn.
- Nếu thuật toán phát hiện thấy hành vi nào đó khác biệt đáng kể so với hồ sơ bình thường (ví dụ: đèn liên tục gửi yêu cầu mạng đến một địa chỉ lạ vào giữa đêm), nó có thể đưa ra cảnh báo về một cuộc tấn công tiềm ẩn (ví dụ: đèn đã bị botnet kiểm soát).

Ý nghĩa: Cho phép phát hiện các mối đe dọa mới và phức tạp dựa trên hành vi, thay vì chỉ dựa vào các mẫu tấn công đã biết.

PHẦN 3: XÂY DỰNG THUẬT TOÁN BẢO MẬT DỮ LIỆU

3.1. Lựa chọn thuật toán bảo mật dữ liệu

3.1.1. Phân tích yêu cầu bảo mật

Trong bối cảnh an toàn thông tin ngày càng trở nên quan trọng, việc lựa chọn các thuật toán bảo mật dữ liệu đóng vai trò then chốt trong việc đảm bảo tính toàn vẹn, tính bảo mật và khả năng xác thực của thông tin. Yêu cầu bảo mật bao gồm: đảm bảo dữ liệu không bị lộ (confidentiality), không bị thay đổi (integrity), và người nhận xác minh được nguồn gốc dữ liệu (authenticity).

Với yêu cầu trên, dự án này sẽ được xây dựng dựa vào hai thuật toán là Playfair và RSA.

3.1.2. Phân tích thuật toán

Playfair: Thuật toán chủ yếu bao gồm ba bước: [4]

- Bước 1: Chuyển đổi văn bản thuần túy thành dạng ghép (tức là thành cặp gồm hai chữ cái).
  - Khi tách văn bản thành từng cặp, nếu các chữ cái trong một cặp giống nhau thì chèn ký tự phụ x.
  - Cuối cùng nếu chỉ còn lại một chữ cái thì không có cặp nào, ta có thể chèn chữ cái phụ x
  - Xóa khoảng trắng khỏi chuỗi cũng như các ký tự đặc biệt ngoài 25 chữ cái.
- Bước 2: Tạo Ma trận Khóa Mã hóa.
  - Ma trận khóa mã hóa là một lưới 5×5 chữ cái đóng vai trò là khóa để mã hóa văn bản thuần túy.
  - Mỗi chữ cái trong số 25 chữ cái phải là duy nhất và một chữ cái trong bảng chữ cái (thường là chữ J) sẽ bị loại khỏi bảng (vì bảng chỉ có thể chứa 25 chữ cái).
  - Quy tắc:

1. Nếu khóa có các chữ cái lặp lại thì hãy bỏ chúng đi.
2. Điền khóa ký tự (đã bỏ lặp) và các vị trí còn lại bằng các chữ cái chưa sử dụng vào Ma trận 5X5.

- Bước 3: Mã hóa văn bản thuần túy bằng Ma trận khóa mã hóa và nhận được văn bản mã hóa.

- Quy tắc:
  - Nếu cả hai chữ cái không cùng một cột và không cùng một hàng thì hãy vẽ một hình chữ nhật tưởng tượng và lấy các chữ cái ở góc nằm ngang đối diện của hình chữ nhật.
  - Nếu cả hai chữ cái đều nằm trong cùng một cột: Lấy chữ cái bên dưới mỗi chữ cái (quay lại đầu nếu ở dưới cùng).
  - Nếu cả hai chữ cái nằm trên cùng một hàng: Lấy chữ cái ở bên phải của mỗi chữ cái (quay lại chữ cái bên trái nhất nếu ở vị trí bên phải nhất)

RSA: Thuật toán chủ yếu bao gồm ba giai đoạn: [6]

- Tạo khóa: Khóa công khai và khóa riêng tư
  - Chọn hai số nguyên tố lớn, giả sử là p và q. Các số nguyên tố này phải được giữ bí mật.
  - Tính tích các số nguyên tố, tích này là một phần của khóa công khai cũng như khóa riêng tư:

$$n = p * q$$

  - Tính hàm Euler Totient Φ(n):

$$aΦ(n) = Φ(p * q) = Φ(p) * Φ(q) = (p - 1) * (q - 1)$$

  - Chọn số mũ mã hóa e sao cho e phải đồng nguyên tố với Φ(n):

$$1 < e < Φ(n) \quad \text{và} \quad gcd(e, Φ(n)) = 1$$

  - Tính số mũ giải mã d, sao cho d là nghịch đảo nhân môđun của e mod Φ(n):

$$(d * e) \equiv 1 \bmod Φ(n)$$

  - Kết quả cuối cùng là khóa công khai = (n, e) và khóa riêng tư = (n, d).

- Mã hóa:
  - Bước 1: Thông điệp A được chuyển đổi thành dạng số bằng ASCII và các chương trình mã hóa khác.
  - Bước 2: Sử dụng khóa công khai (n, e) để mã hóa thông điệp và lấy văn bản mã hóa (a) bằng công thức:

$$a = M^e \bmod n$$

- Giải mã:

- Để giải mã văn bản mã hóa a , hãy sử dụng khóa riêng (n, d) và lấy dữ liệu gốc (A) bằng công thức:

$$A = C^d \bmod n$$

3.2. Thiết kế và xây dựng mô hình mô phỏng

Công cụ: Python với thư viện tkinter cho giao diện đồ họa, thư viện base64 và rsa cho mã hóa RSA.

Môi trường: Visual Studio Code, Python 3.x, hệ điều hành Windows.

Kịch bản tấn công:

- Giả sử người dùng ác ý cố gắng đọc trộm thông tin chưa được mã hóa.
- Thử đoán khóa mã hóa thủ công đối với Playfair.

Kịch bản phòng thủ:

- Mã hóa văn bản bằng Playfair và RSA trước khi truyền.
- Kiểm tra tính hợp lệ khóa và ngăn người dùng sử dụng giá trị không an toàn.

3.3. Triển khai thuật toán bảo mật dữ liệu trong mô hình mô phỏng

Các thuật toán được triển khai bằng Python và tích hợp vào giao diện người dùng với tkinter. Người dùng có thể chọn thuật toán, nhập văn bản, nhập khóa và xem kết quả mã hóa/giải mã.

Thực hiện mô phỏng bằng cách: Chạy giao diện chính với lựa chọn giữa Playfair và RSA. Giao diện riêng cho mỗi thuật toán hỗ trợ nhập khóa, văn bản và thực hiện mã hóa/giải mã.

Kết quả mô phỏng:

- Hệ thống hiển thị kết quả rõ ràng, hiển thị thời gian mã hóa và giải mã giúp người dùng đánh giá hiệu suất.
- Playfair mã hóa nhanh hơn nhưng bảo mật thấp. RSA bảo mật cao nhưng chậm hơn.

3.4. Đánh giá hiệu quả của thuật toán bảo mật dữ liệu

Đo lường các chỉ số bảo mật (Security metrics):

- Tỷ lệ phát hiện tấn công (Detection rate): Không đo cụ thể vì đây là mô phỏng, nhưng mã hóa giúp ẩn thông tin khỏi người dùng trái phép.
- Tỷ lệ cảnh báo sai: Không có trong mô hình.
- Thời gian phát hiện tấn công: Không áp dụng trực tiếp.
- Khả năng chịu đựng tấn công: RSA rất mạnh trước tấn công brute-force.
- Mức độ tiêu thụ tài nguyên: Playfair nhẹ, RSA nặng hơn do tính toán lớn.

Đánh giá khả năng chống lại các hình thức tấn công:

- Playfair dễ bị phân tích tần suất, phù hợp mục đích học thuật.
- RSA chống lại tấn công brute-force, thích hợp trong thực tế truyền thông tin nhạy cảm.

3.5. So sánh kết quả với các thuật toán bảo mật dữ liệu khác

So sánh hiệu quả bảo mật:

- Playfair là thuật toán mã hóa đối xứng đơn giản, dễ hiểu và dễ cài đặt, nhưng dễ bị tấn công, bảo mật thấp.
- RSA là thuật toán mã hóa bất đối xứng, bảo mật cao nhờ vào tính toán số học phức tạp, thường dùng trong trao đổi khóa và xác thực.
- Về độ an toàn, RSA vượt trội do khó giải mã nếu không có khóa riêng, trong khi Playfair dễ bị phân tích tần suất.

So sánh thời gian, chi phí triển khai:

- Thời gian: Playfair nhanh hơn rõ rệt.
- Chi phí triển khai: Playfair gần như không yêu cầu tài nguyên, RSA yêu cầu tính toán lớn hơn.

3.6. Ưu nhược điểm của thuật toán đề xuất

Ưu điểm:

- Playfair dễ hiểu, dùng tốt trong dạy học.
- RSA bảo mật cao, thực tế hơn.

Nhược điểm:

- Playfair bảo mật thấp.
- RSA thực hiện chậm và yêu cầu tính toán lớn.

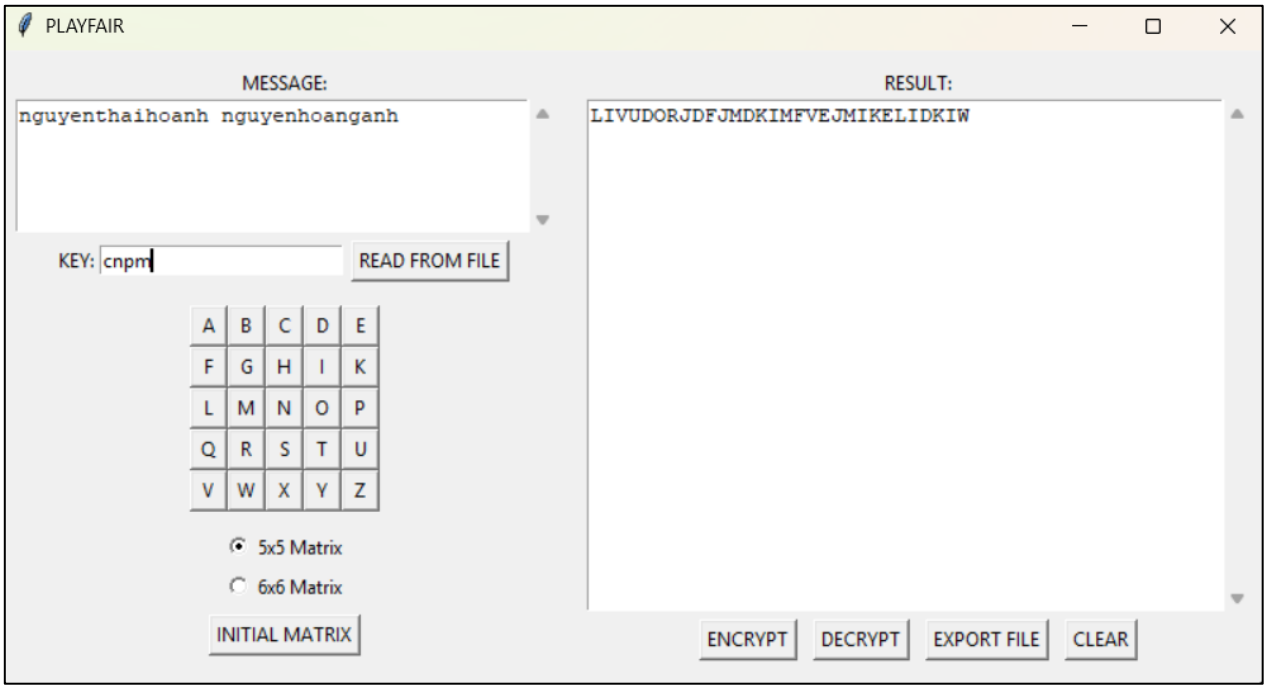
3.7 Giao diện phần mềm

3.7.1 Giao diện chính:

Hình 3.1: Giao diện chính

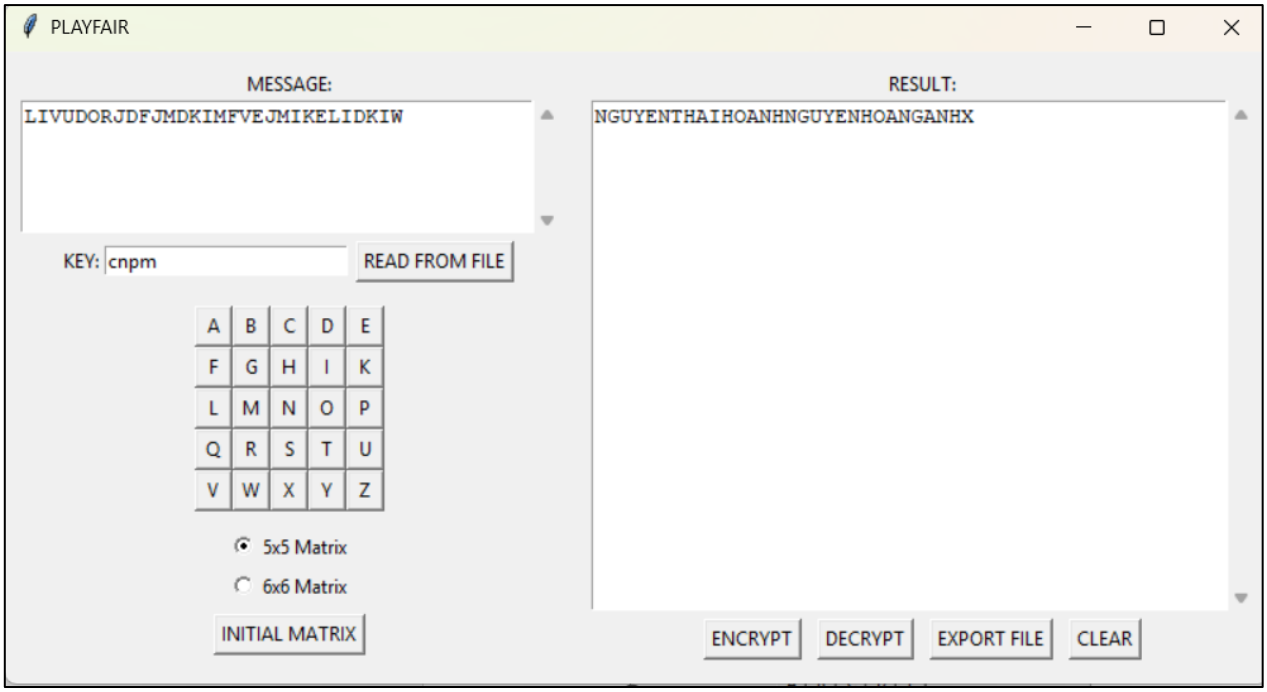
3.7.2 Giao diện Playfair:

Encrypt:



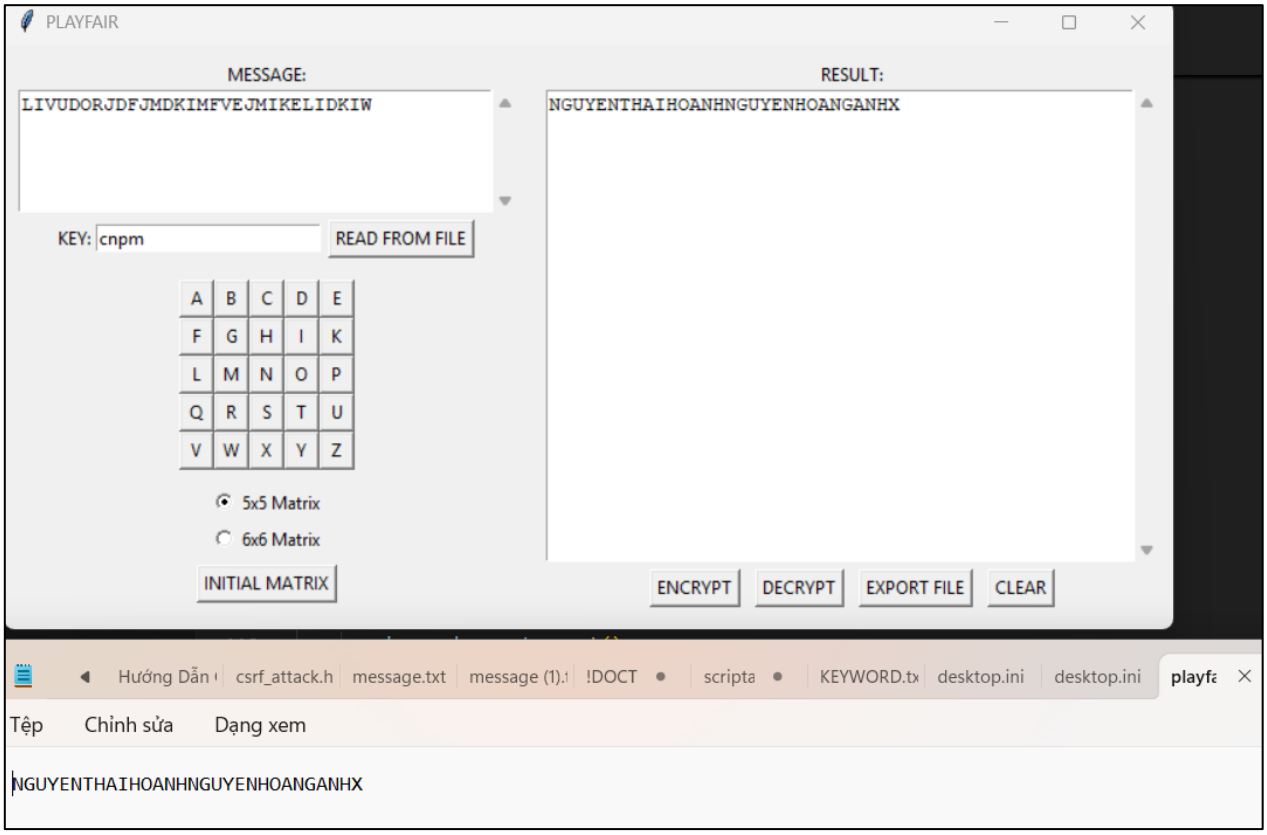
Hình 3.2: Giao diện Playfair Encrypt

Decrypt:



Hình 3.3: Giao diện Playfair Decrypt

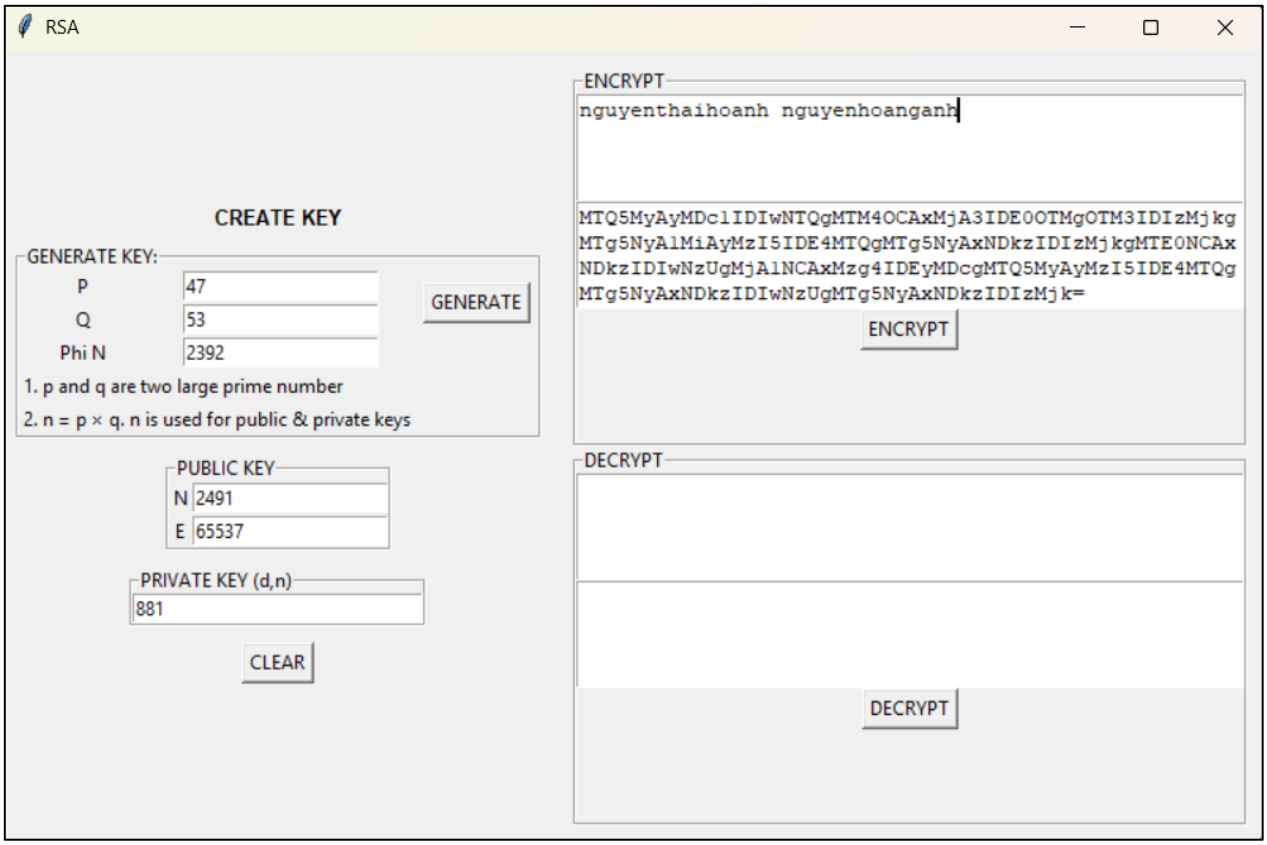
Xuất file :



Hình 3.4: Giao diện Playfair xuất file

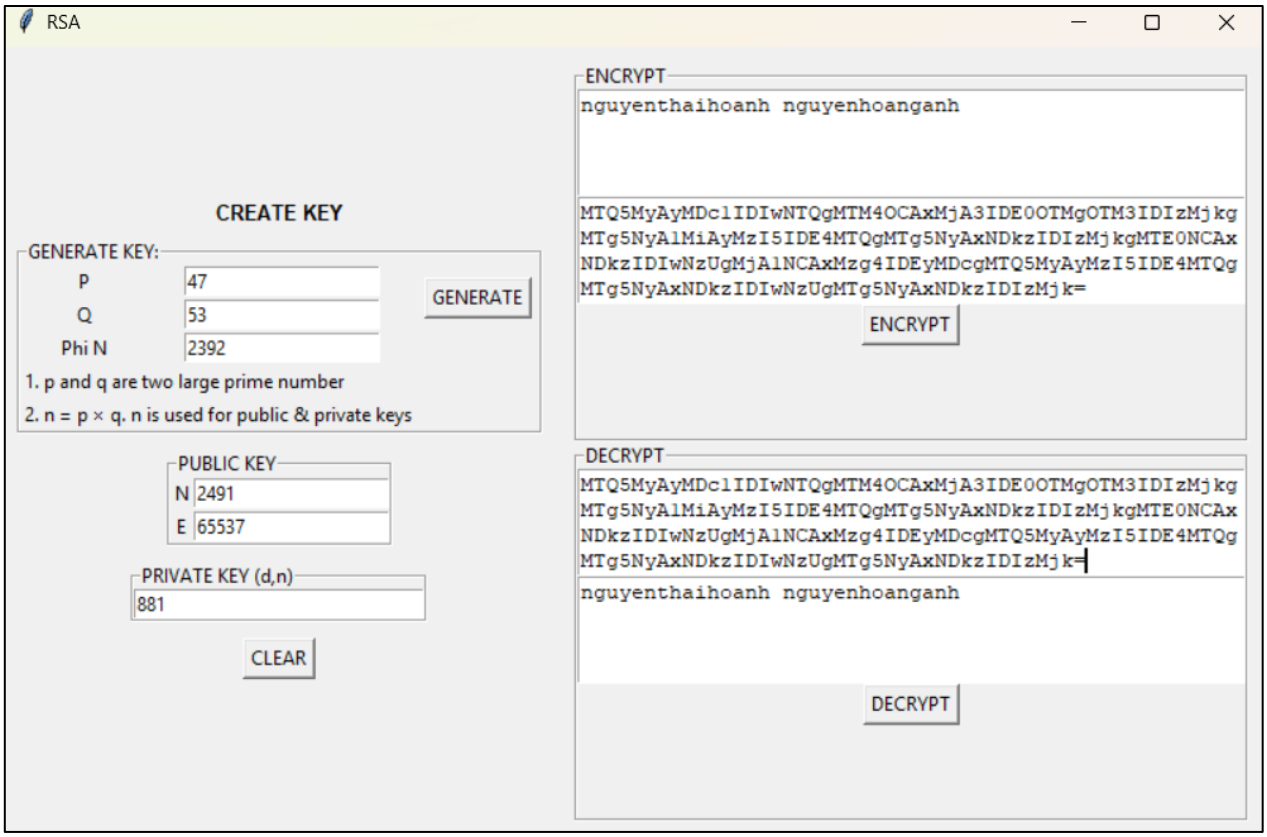
3.7.3 Giao diện RSA:

Encrypt:



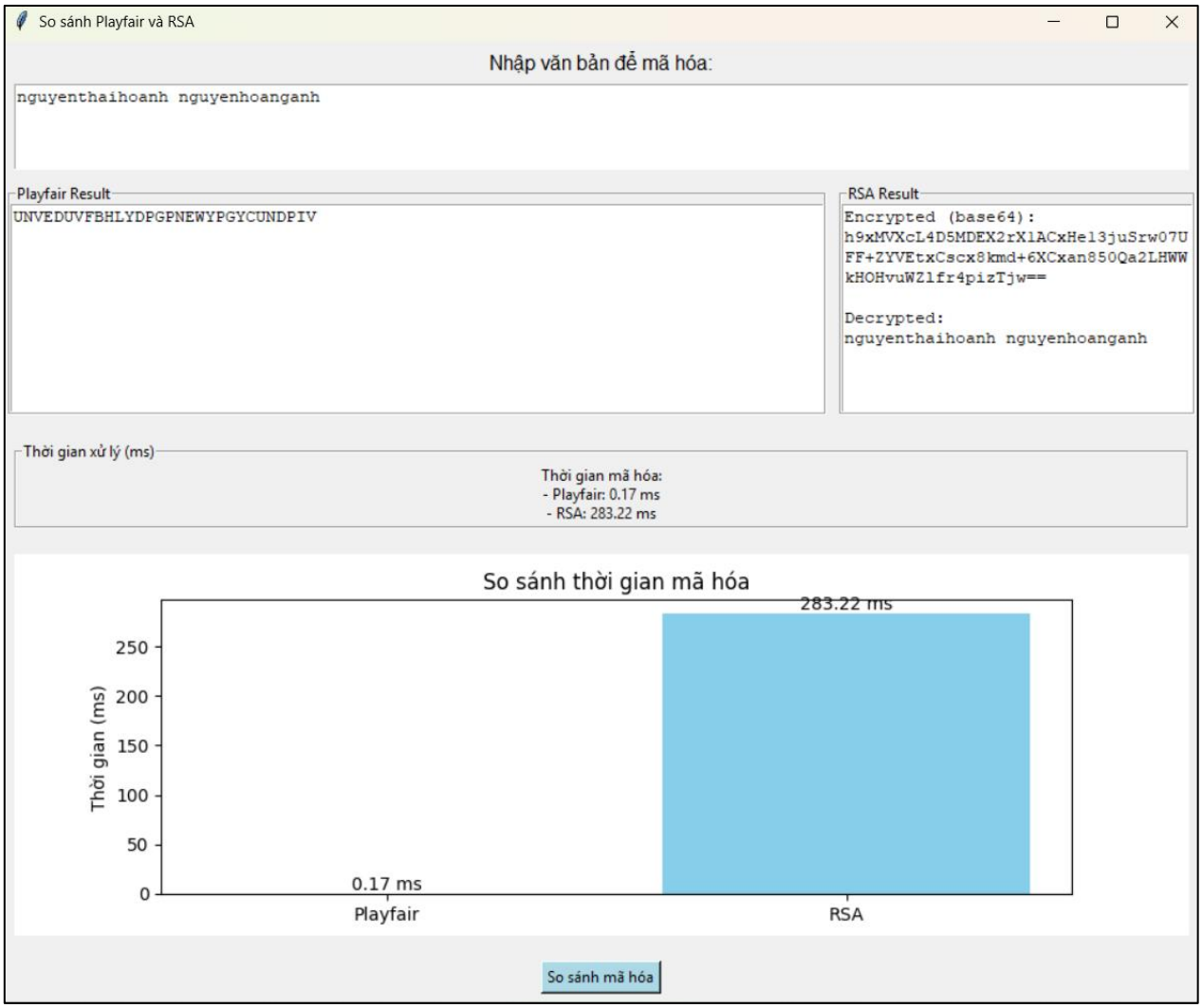
Hình 3.5: Giao diện RSA Encrypt

Decrypt:



Hình 3.6: Giao diện RSA Decrypt

3.7.4 Giao diện so sánh Playfair và RSA:



Hình 3.7: Giao diện so sánh Playfair và RSA

PHẦN 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1. Tóm tắt kết quả đạt được

Đề tài chúng em đã xây dựng được một mô hình mô phỏng trực quan hai thuật toán bảo mật Playfair và RSA với các chức năng:

- Giao diện GUI cho phép mã hóa/giải mã bằng Playfair và RSA.
- So sánh hiệu suất giữa hai thuật toán, hiển thị thời gian xử lý qua biểu đồ.
- Xuất kết quả dưới dạng PDF, bao gồm dữ liệu và biểu đồ.

Đánh giá chung về khả năng chống lại các cuộc tấn công, RSA vượt trội về bảo mật so với Playfair và Playfair có tốc độ nhanh hơn RSA.

Đề tài này đã góp phần giúp chúng em hiểu rõ hơn về mã hóa đối xứng và bất đối xứng. Là nền tảng để phát triển mô phỏng nâng cao hơn như kết hợp thuật toán lai hoặc tích hợp mạng.

4.2 Hướng phát triển:

Đề tài chúng em đã hoàn thành các mục tiêu cơ bản ban đầu, vẫn còn nhiều hướng phát triển tiềm năng có thể giúp mở rộng tính ứng dụng, tăng cường hiệu quả và nâng cao độ thực tiễn của mô hình mô phỏng. Cụ thể:

- Mở rộng mô phỏng nhiều thuật toán mã hóa khác:
- AES (Advanced Encryption Standard): thuật toán mã hóa đối xứng hiện đại, được sử dụng rộng rãi trong thực tế.
- ElGamal hoặc ECC (Elliptic Curve Cryptography): thuật toán mã hóa bất đối xứng với hiệu suất cao, độ bảo mật lớn.
- Xây dựng hệ thống web hoặc ứng dụng đa nền tảng
- Chuyển mô hình từ ứng dụng desktop sang ứng dụng web/mobile sẽ giúp:
    - Người dùng dễ dàng truy cập qua trình duyệt hoặc điện thoại mà không cần cài đặt.
    - Tăng tính tương tác, dễ tích hợp với cơ sở dữ liệu hoặc các API bên ngoài.
    - Phù hợp với xu hướng hiện đại hóa và học tập trực tuyến.

1. Tích hợp tính năng phân tích độ mạnh của khóa
    - Cho phép người dùng tạo khóa ngẫu nhiên hoặc tự chọn, sau đó phân tích độ mạnh của khóa dựa trên độ dài, độ phức tạp, tần suất ký tự…
    - Cảnh báo khi khóa yếu và đề xuất khóa mạnh hơn
- Mô phỏng tấn công thực tế
    - Tấn công vét cạn (brute-force attack)
    - Tấn công phân tích tần suất
    - Tấn công trung gian (man-in-the-middle)
    - Qua đó giúp người học nhận thức rõ về rủi ro bảo mật và lý do cần mã hóa.
- Tích hợp cơ sở dữ liệu để lưu trữ lịch sử mã hóa
    - Thời gian thao tác
    - Thuật toán đã dùng
    - Khóa được chọn
    - Kết quả mã hóa/giải mã
- Kết hợp công nghệ trí tuệ nhân tạo (AI)
    - Gợi ý thuật toán phù hợp tùy theo loại dữ liệu, mức độ bảo mật mong muốn.
    - Phân tích log hệ thống để phát hiện rủi ro bảo mật bất thường.
- Tích hợp tính năng ký số và xác thực
    - Chữ ký số
    - Kiểm tra toàn vẹn dữ liệu (hashing: MD5, SHA-256)
- Cuối cùng, nếu kết hợp được với các mô hình trên vào mô hình hiện tại sẽ giúp mô hình tiếp cận được nhiều đối tượng người dùng hơn, đặc biệt trong bối cảnh toàn cầu hóa và hội nhập quốc tế , còn tạo ra nền tảng vững chắc cho việc phát triển các phần mềm bảo mật thực tế trong tương lai, đồng thời đóng góp tích cực vào công cuộc giáo dục, nghiên cứu và ứng dụng công nghệ an toàn thông tin trong thời đại số.

4.3. Kết luận

Việc mô phỏng thuật toán bảo mật đóng vai trò quan trọng trong giáo dục và nghiên cứu. Thông qua đề tài này, chúng em có thể hình dung rõ quy trình, đặc điểm và cách vận hành của từng thuật toán.

Từ đó hiểu sâu hơn về an toàn thông tin và áp dụng trong các hệ thống thực tế.

Tài Liệu Tham Khảo

[1] Công nghệ & ngân hàng số, "*Bảo mật truyền thông IoT qua mạng di động trong hệ thống ngân hàng số*", Tạp chí Ngân Hàng, 2022
https://tapchinganhang.gov.vn/bao-mat-truyen-thong-iot-qua-mang-di-dong-trong-he-thong-ngan-hang-so-9471.html

[2] Jianxing Zhu & Lina Huo & Mohd Dilshad Ansari & Mohammad Asif Ikbal, "*Research on Data Security Detection Algorithm in IoT Based on K-means*", ResearchGate, 2021
https://www.researchgate.net/publication/355538969_Research_on_Data_Security_Detection_Algorithm_in_IoT_Based_on_K-means

[3] Lê Phương, "*Mật mã Playfair | An toàn thông tin | Đại học Thủy Lợi*", Đại học Thủy Lợi, 2024
https://docx.com.vn/tai-lieu/mat-ma-playfair-an-toan-thong-tin-dai-hoc-thuy-loi-76430

[4] Karan-Munjani, "*What is Playfair Cipher Encryption Algorithm*", DEV, 2021
https://dev.to/karanmunjani/what-is-playfair-cipher-encryption-algorithm-4npk

[5] ThS. Trần Công Mậu, "*Thuật toán mã hóa RSA và thực nghiệm*", Trường Đại học Hà Tĩnh, 2023
https://ent.htu.edu.vn/tin-t%E1%BB%A9c/352-thu%E1%BA%ADt-to%C3%A1n-m%C3%A3-h%C3%B3a-rsa-v%C3%A0-th%E1%BB%B1c-nghi%E1%BB%87m

[6] GeeksFofGeeks, "*RSA Algorithm in Cryptography*", GeeksFofGeeks, 2025
https://www.geeksforgeeks.org/rsa-algorithm-cryptography/

[7] Rakesh Mohanty et al., "*A Survey on IoT Security: Challenges and Solutions*", 2020
https://ieeexplore.ieee.org/document/9052607 *(22/05/2025: Link die 404)*

[8] Cisco, "*IoT Security: What Is IoT Security?*", Cisco, 2023
https://www.cisco.com/c/en/us/solutions/internet-of-things/iot-security.html

[9] Vietnamnet, "*An toàn thông tin trong thời đại IoT*", 2022
https://vietnamnet.vn/an-toan-thong-tin-trong-thoi-dai-iot-2032054.html

[10] Tạp chí An toàn Thông tin, "*Tổng quan về mã hóa trong bảo mật dữ liệu*", 2023
https://antoanthongtin.vn/chuyen-muc/nghien-cuu-trao-doi/tong-quan-ve-ma-hoa-trong-bao-mat-du-lieu.html