**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**\*\*\*\*\*\*\*\*\*\***

**A logo of hands holding a book and a candle

AI-generated content may be incorrect.**

**BÁO CÁO ĐỒ ÁN CUỐI KỲ MÔN**

**MẬT MÃ HỌC**

**CHỦ ĐỀ**

**ỨNG DỤNG CỦA MẬT MÃ TRONG LƯU TRỮ MẬT KHẨU**

**TRONG TRÌNH DUYỆT FIREFOX TRÊN WINDOWS**

**GVHD: Trần Đắc Tốt**

**Nhóm thực hiện:**

**1. Nguyễn Thanh Nhật - 23162072**

**2. Lê Văn Ánh Thông - 23162097**

**3. Ngô Tuấn Phát - 23162075**

**4. Trương Xuân Nhật - 23162073**

*TP. HCM, tháng 4 năm 2025*

**MỤC LỤC**

[**I. MỞ ĐẦU** 4](#_Toc195992456)

[**1. Mục tiêu Nghiên Cứu** 4](#_Toc195992457)

[**2. Thời gian và phạm vi nghiên cứu** 4](#_Toc195992458)

[**3. Phương pháp nghiên cứu** 5](#_Toc195992459)

[1. Nghiên cứu lý thuyết 5](#_Toc195992460)

[2. Phân tích thực nghiệm 5](#_Toc195992461)

[3. Mô phỏng và biểu diễn 6](#_Toc195992462)

[4. Đánh giá và đề xuất 6](#_Toc195992463)

[**II. CÁC GIAI ĐOẠN THỰC HIỆN** 7](#_Toc195992464)

[**1. Tìm hiểu đề tài** 7](#_Toc195992465)

[1.1. Tổng quan 7](#_Toc195992466)

[1.2. Cơ sở lý thuyết 7](#_Toc195992467)

[1.3 Cơ sở thực tiễn 10](#_Toc195992468)

[**2. Cách thức lưu trữ mật khẩu trên Firefox (Windows)** 11](#_Toc195992469)

[2.1 Cấu trúc thư mục profile 11](#_Toc195992470)

[2.2 Cấu trúc file logins.json 12](#_Toc195992471)

[2.3 File key4.db 12](#_Toc195992472)

[2.4 Sơ đồ tổng quan quy trình lưu trữ 13](#_Toc195992473)

[**3. Các thành phần mật mã sử dụng trong Firefox** 13](#_Toc195992474)

[3.1. Tổng quan về các loại mã hóa trong Firefox 13](#_Toc195992475)

[3.2. Các loại mã hóa mà Firefox sử dụng 13](#_Toc195992476)

[3.3. Tổng kết quy trình mã hóa 17](#_Toc195992477)

[**4. Độ an toàn trong cơ chế lưu trữ mật khẩu của Firefox** 17](#_Toc195992478)

[4.1 Cơ chế bảo vệ mật khẩu của Firefox 17](#_Toc195992479)

[4.2 Tầm quan trọng của Mật khẩu chính (Primary Password) 18](#_Toc195992480)

[4.3 Công cụ giải mã mật khẩu Firefox 18](#_Toc195992481)

[**5. Lập trình triển khai** 18](#_Toc195992482)

[5.1. Giới thiệu về công cụ firefox\_decrypt 18](#_Toc195992483)

[5.2. Mô phỏng 19](#_Toc195992484)

[**KẾT LUẬN** 20](#_Toc195992485)

[**TÀI LIỆU THAM KHẢO** 21](#_Toc195992486)

# **I. MỞ ĐẦU**

## **1. Mục tiêu Nghiên Cứu**

Ngày nay, việc lưu trữ mật khẩu trình duyệt trở nên phổ biến khi người dùng truy cập các dịch vụ trực tuyến thường xuyên. Tuy nhiên, việc lưu trữ mật khẩu trên thiết bị cục bộ tiềm ẩn nhiều rủi ro nếu không được bảo vệ đúng cách. Để giải quyết vấn đề này, các trình duyệt hiện đại như Firefox đã tích hợp những kỹ thuật mật mã nhằm đảm bảo an toàn cho dữ liệu người dùng.

Firefox – trình duyệt mã nguồn mở do Mozilla phát triển – có một hệ thống quản lý mật khẩu tên là Firefox Lockwise (hiện nay đã tích hợp trực tiếp vào trình duyệt). Khi người dùng lưu mật khẩu, trình duyệt sẽ mã hóa dữ liệu và lưu trữ chúng trong thư mục cấu hình (profile) của người dùng. Việc này giúp đảm bảo rằng ngay cả khi người khác truy cập được vào máy tính, họ cũng không thể dễ dàng lấy được mật khẩu nếu không có khóa giải mã tương ứng.

Đề tài này sẽ đi sâu vào việc tìm hiểu cách Firefox lưu trữ mật khẩu, các thuật toán mật mã được sử dụng trong quá trình đó, và đánh giá độ an toàn của phương pháp này.

## **2. Thời gian và phạm vi nghiên cứu**

Về thời gian: Dự án được tiến hành thực hiện trong khoảng từ giữa tháng 03/2025 gần giữa tháng 05/2025.

Về phạm vi nghiên cứu: Liên quan đến vấn đề bảo mật thông tin tài khoản cá nhân của người dùng trên FireFox. Phân tích cơ chế mã hóa mật khẩu trong trình duyệt Firefox" tập trung vào việc tìm hiểu và đánh giá các phương pháp bảo mật mà Firefox sử dụng để bảo vệ thông tin đăng nhập của người dung.

## **3. Phương pháp nghiên cứu**

### **1. Nghiên cứu lý thuyết**

**- Thu thập tài liệu kỹ thuật:**  
 + Phân tích tài liệu chính thức từ Mozilla về Firefox, đặc biệt là về cơ chế lưu trữ và mã hóa mật khẩu.  
 + Tìm hiểu các thuật toán mật mã được sử dụng (AES, PBKDF2, TLS, NSS).

**- Tổng hợp và phân tích:**  
 + Hệ thống hóa các thông tin để hiểu rõ quy trình mã hóa/giải mã mật khẩu trong Firefox.  
 + So sánh với các trình duyệt khác (Chrome) để đánh giá ưu điểm và hạn chế.

### **2.** **Phân tích thực nghiệm**

**- Khám phá cấu trúc tệp tin:**  
 + Sử dụng công cụ SQLite để mở và phân tích tệp key4.db (nơi lưu trữ khóa mã hóa).  
 + Dùng trình soạn thảo văn bản xem nội dung logins.json (tệp chứa thông tin đăng nhập đã mã hóa).

**- Thử nghiệm với mật khẩu chính:**  
 + Thiết lập và gỡ bỏ mật khẩu chính trong Firefox để quan sát sự thay đổi trong cách lưu trữ và mã hóa dữ liệu.  
 + Ghi nhận thay đổi trong key4.db và logins.json khi có và không có mật khẩu chính.

**- Sử dụng công cụ mã hóa:**  
 + Áp dụng các công cụ mã hóa để thử giải mã dữ liệu trong logins.json bằng khóa từ key4.db, nhằm hiểu rõ quy trình giải mã.

### **3.** **Mô phỏng và biểu diễn**

**- Vẽ lưu đồ (flowchart):**  
 + Sử dụng draw.io hoặc Lucidchart để vẽ lưu đồ mô tả quy trình mã hóa và giải mã mật khẩu trong Firefox.  
 + Minh họa các bước: xác minh mật khẩu chính, giải mã khóa đối xứng, giải mã thông tin đăng nhập.

**- Mô phỏng quy trình:**  
 + Tạo mô hình mô phỏng quy trình mã hóa và giải mã để trình bày trong báo cáo, giúp người đọc dễ hiểu hơn.

### **4.** **Đánh giá và đề xuất**

**- Đánh giá bảo mật:**  
 + Phân tích mức độ an toàn của cơ chế mã hóa mật khẩu trong Firefox, đặc biệt khi không sử dụng mật khẩu chính.  
 + So sánh với các phương pháp bảo mật khác để đưa ra nhận định về hiệu quả của Firefox.

**- Đề xuất cải tiến:**  
 + Dựa trên phân tích và thử nghiệm, đề xuất cải tiến nhằm nâng cao bảo mật cho người dùng Firefox.  
 + Khuyến nghị người dùng thiết lập mật khẩu chính và sử dụng công cụ quản lý mật khẩu để bảo vệ thông tin cá nhân.

# **II. CÁC GIAI ĐOẠN THỰC HIỆN**

## **1. Tìm hiểu đề tài**

### **1.1. Tổng quan**

Nhóm nghiên cứu tiến hành phân tích và tra cứu nhiều tài liệu để đưa ra cái nhìn tổng quát về cơ chế bảo mật thông tin đăng nhập trong Firefox trên nền tảng Windows. Đề tài được phân rã thành các chủ đề nhỏ liên quan và nghiên cứu từ góc độ lý thuyết đến thực hành vận dụng, tập trung vào các mục tiêu chính sau:

- Tìm hiểu cơ chế bảo mật lưu trữ mật khẩu của Firefox

- Phân tích vai trò của mật mã và các thuật toán liên quan

- Thực nghiệm và phân tích cấu trúc dữ liệu

- So sánh với các trình duyệt khác

- Đề xuất giải pháp cải tiến

### **1.2. Cơ sở lý thuyết**

Nhóm nghiên cứu đã tiến hành tra cứu, tìm hiểu và phân tích nhiều tài liệu kỹ thuật nhằm xây dựng nền tảng lý thuyết cho đề tài "Ứng dụng của mật mã trong lưu trữ mật khẩu trong trình duyệt Firefox trên Windows". Các kiến thức nền tảng này không chỉ được trích xuất từ các tài liệu của Mozilla mà còn được tổng hợp từ các nguồn nghiên cứu độc lập liên quan đến mật mã học, mã hóa đối xứng, hàm dẫn xuất khóa, cũng như giao thức bảo mật và thư viện NSS.

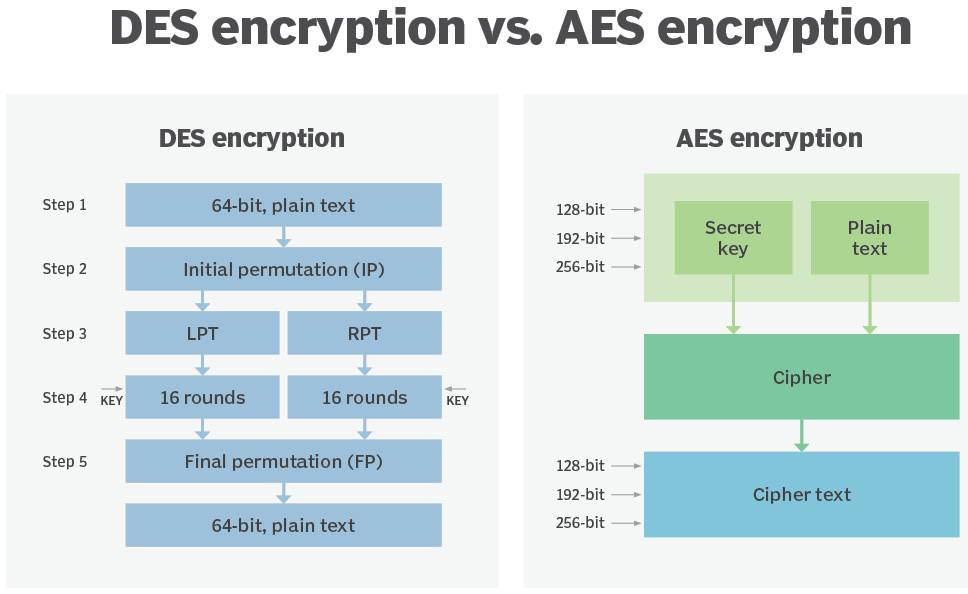
**- Mật mã học:**  
Nền tảng lý thuyết về mật mã học được nhóm nghiên cứu làm rõ, với mục tiêu bảo vệ dữ liệu bằng các phương pháp biến đổi thông tin (mã hóa) mà chỉ các bên được phép mới có thể giải mã. Những khái niệm này tạo cơ sở cho việc áp dụng các giải pháp bảo mật trong trình duyệt Firefox.

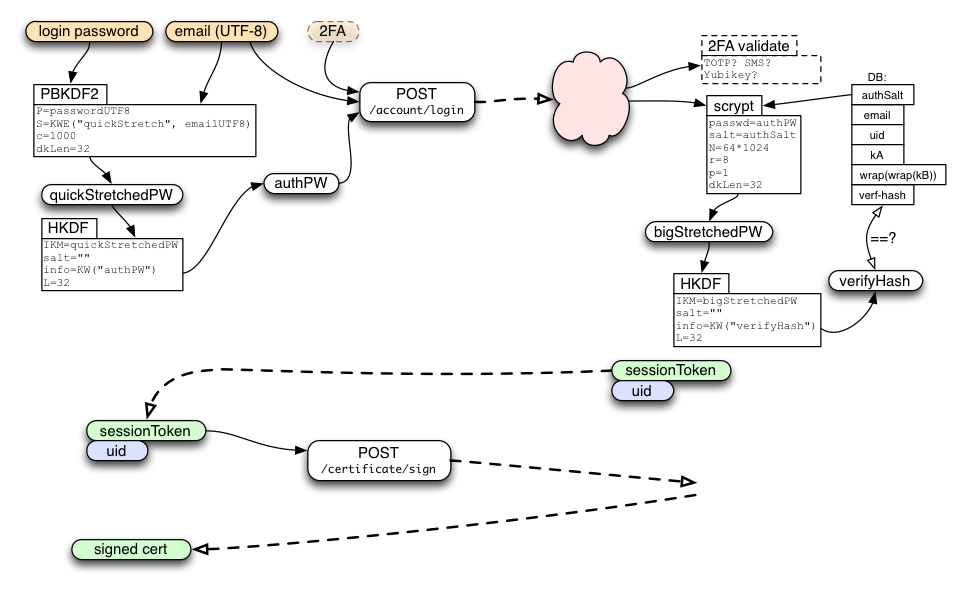
**- Mã hóa đối xứng:**  
Các thuật toán mã hóa đối xứng như AES (chủ yếu là AES-256 ở chế độ CBC với IV ngẫu nhiên) là trụ cột trong việc bảo vệ thông tin đăng nhập lưu trữ cục bộ. Nhóm nghiên cứu đã phân tích đặc điểm, ưu điểm và hạn chế của AES cũng như so sánh với các giải pháp cũ như 3DES để xác định mức độ an toàn thực tiễn của hệ thống lưu trữ.

**- Hàm dẫn xuất khóa (PBKDF2):**  
Nhóm nghiên cứu làm sáng tỏ vai trò của thuật toán PBKDF2 trong quá trình chuyển đổi mật khẩu chính (Primary Password) thành một khóa dẫn xuất mạnh mẽ. Khóa dẫn xuất này được dùng để mã hóa (bọc) khóa đối xứng trong file key4.db, qua đó đảm bảo rằng ngay cả khi file bị truy cập trái phép, dữ liệu vẫn được bảo vệ bởi lớp bảo mật phụ thuộc vào mật khẩu chính mà người dùng thiết lập.

**- Các tệp tin lưu trữ:**  
Nhóm nghiên cứu đã phân tích chi tiết cấu trúc của file logins.json, chứa thông tin đăng nhập đã được mã hóa, và file key4.db, lưu trữ khóa đối xứng. Qua đó, nhóm nhận thấy việc áp dụng cơ chế mã hóa và bảo vệ khóa thông qua mật khẩu chính cung cấp một lớp bảo vệ bổ sung cho dữ liệu người dùng.

**- TLS:**  
Trong bối cảnh bảo mật dữ liệu truyền tải, giao thức TLS được áp dụng để mã hóa các kết nối HTTPS, đảm bảo rằng dữ liệu người dùng không bị nghe lén trong quá trình truyền qua mạng. Nhóm nghiên cứu đã tổng hợp các tài liệu về TLS nhằm làm nổi bật vai trò của giao thức này trong hệ thống bảo mật tổng thể của Firefox.

**- NSS (Network Security Services):**  
NSS là bộ thư viện mã nguồn mở của Mozilla, cung cấp API cho việc triển khai các thuật toán mật mã (AES, PBKDF2, TLS, v.v.). Nhóm nghiên cứu đã tập trung vào cách thức NSS tích hợp vào Firefox trên Windows, đảm bảo việc thực hiện các thuật toán bảo mật một cách nhất quán và đáng tin cậy.



### **1.3 Cơ sở thực tiễn**

## **2. Cách thức lưu trữ mật khẩu trên Firefox (Windows)**

### **2.1 Cấu trúc thư mục profile**

Firefox lưu trữ thông tin người dùng trong thư mục "profile" có đường dẫn như sau (Windows):

**C:\Users\<username>\AppData\Roaming\Mozilla\Firefox\Profiles\<profile\_name>.default-release\**

Trong thư mục này, hai tệp quan trọng liên quan đến lưu trữ mật khẩu gồm:

|  |  |
| --- | --- |
| Tệp tin | Vai trò |
| **logins.json** | Chứa danh sách tài khoản và mật khẩu đã mã hóa. |
| **key4.db** | Lưu trữ khóa mã hóa (AES key) để giải mã mật khẩu trong logins.json. |

### **2.2 Cấu trúc file logins.json**

Đây là nơi chứa tất cả tài khoản đã lưu, mỗi mục bao gồm:

A screenshot of a computer program

AI-generated content may be incorrect.

Lưu ý: encryptedUsername và encryptedPassword là chuỗi đã mã hóa bằng thuật toán đối xứng AES.

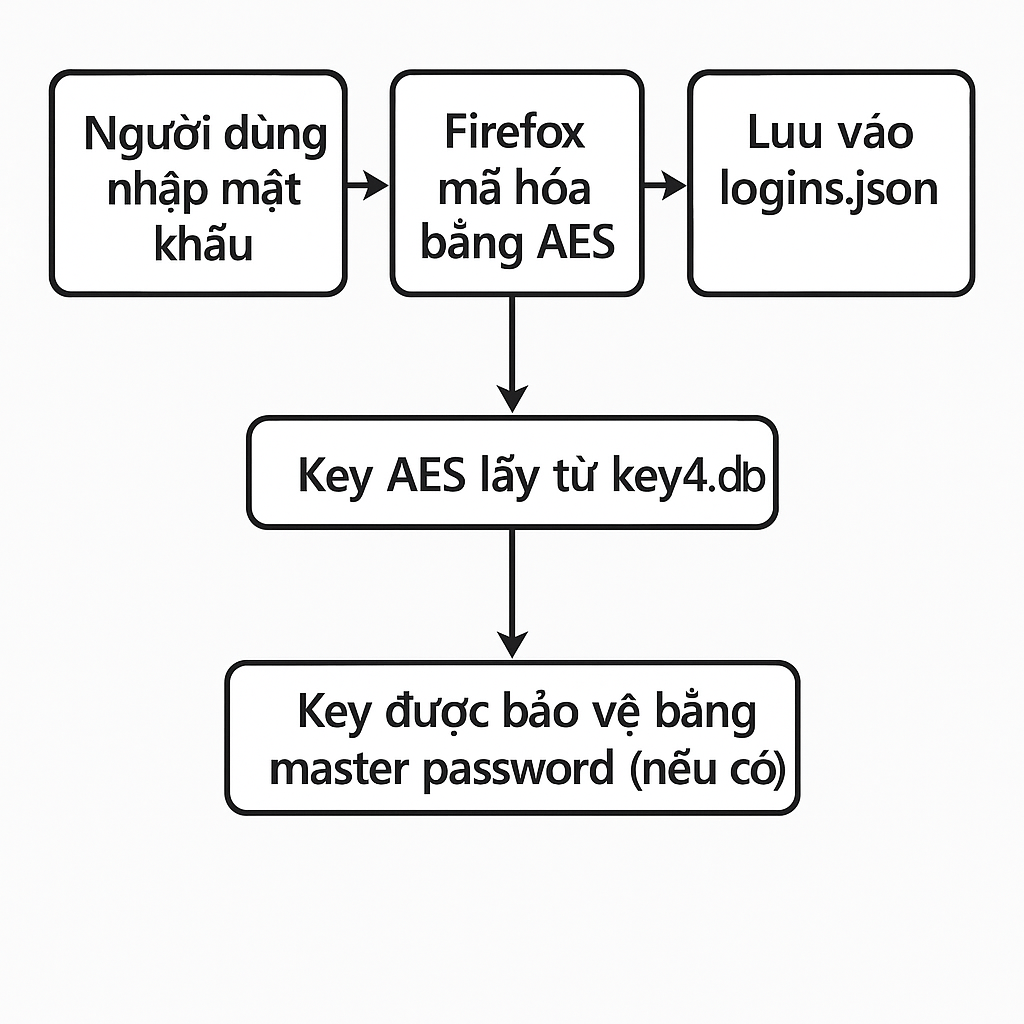
### **2.3 File key4.db**

- Là một cơ sở dữ liệu SQLite chứa key để giải mã mật khẩu.

- Nếu người dùng có đặt master password, key này được mã hóa và chỉ có thể truy cập sau khi nhập đúng mật khẩu chính.

- Nếu không có master password, Firefox vẫn tạo key nhưng sử dụng thông tin profile để bảo vệ ở mức thấp hơn.

### **2.4 Sơ đồ tổng quan quy trình lưu trữ**



## **3. Các thành phần mật mã sử dụng trong Firefox**

### **3.1. Tổng quan về các loại mã hóa trong Firefox**

Firefox là trình duyệt web mã nguồn mở của Mozilla, và để đảm bảo an toàn cho dữ liệu người dùng, nó sử dụng một hệ thống các thành phần mật mã hiện đại. Các thành phần này không chỉ bảo vệ dữ liệu đăng nhập lưu trữ cục bộ mà còn bảo đảm quá trình truyền tải dữ liệu qua Internet được mã hóa và an toàn khỏi các cuộc tấn công. Các thành phần chính bao gồm mã hóa đối xứng, thuật toán dẫn xuất khóa, bảo mật truyền tải qua TLS và thư viện NSS quản lý các thao tác mật mã.

### **3.2. Các loại mã hóa mà Firefox sử dụng**

**a) Mã hóa đối xứng (Symmetric Encryption)**

* **Nguyên tắc:**  
  Mã hóa đối xứng sử dụng một khóa duy nhất để thực hiện cả việc mã hóa và giải mã dữ liệu.
* **Ứng dụng trong Firefox:**
  + Dùng để mã hóa dữ liệu nhạy cảm, chẳng hạn như thông tin đăng nhập (username, password) trước khi lưu trữ cục bộ.
  + Thuật toán chủ đạo là AES, đặc biệt là phiên bản AES-256.
* **Chi tiết kỹ thuật:**
  + **AES-256:** Sử dụng khóa 256-bit mang lại độ bảo mật cao.
  + **Chế độ CBC (Cipher Block Chaining):** Mỗi khối dữ liệu được kết hợp với một Initialization Vector (IV) ngẫu nhiên. Cơ chế này giúp giảm nguy cơ lặp lại mẫu và làm cho ciphertext khác nhau mỗi khi cùng một dữ liệu được mã hóa với cùng một khóa.
* **Ưu điểm:**
  + Tốc độ mã hóa và giải mã nhanh, đồng thời được chuẩn hóa rộng rãi trên các hệ thống bảo mật hiện đại.

**b) Thuật toán dẫn xuất khóa (Key Derivation Function)**

* **Nguyên tắc:**  
  Chuyển đổi mật khẩu của người dùng—đặc biệt là mật khẩu chính (Master Password)—thành một khóa bảo mật thông qua việc áp dụng hàm băm mật mã kèm theo salt và số vòng lặp cao.
* **Ứng dụng cụ thể trong Firefox:**
  + Khi người dùng kích hoạt “mật khẩu chính”, Firefox sử dụng thuật toán PBKDF2 để chuyển đổi mật khẩu đó thành khóa dẫn xuất.
  + Khóa dẫn xuất được sử dụng để “bọc” (wrap) khóa đối xứng lưu trữ trong file key4.db.
* **Chi tiết kỹ thuật:**
  + **PBKDF2:** Kết hợp mật khẩu, salt ngẫu nhiên và số vòng lặp cao (ví dụ, hàng nghìn đến hàng chục nghìn vòng lặp) nhằm tăng độ phức tạp và làm chậm quá trình brute-force tấn công.
* **Ưu điểm:**
  + Số vòng lặp nhiều giúp tăng thời gian tấn công brute-force, từ đó cải thiện khả năng bảo vệ khóa đối xứng trước sự truy cập trái phép.

**c) Mã hóa trong quá trình truyền tải (Transport Layer Security - TLS)**

* **Mục đích:**  
  Bảo vệ dữ liệu khi giao tiếp qua Internet, đảm bảo dữ liệu giữa Firefox và các trang web được mã hóa end-to-end.
* **Ứng dụng trong Firefox:**
  + TLS đảm bảo các kết nối HTTPS an toàn, ngăn chặn các cuộc tấn công “man-in-the-middle”.
* **Chi tiết kỹ thuật:**
  + Sử dụng các bộ cipher hiện đại như AES hoặc ChaCha20, kèm theo hàm băm mật mã như SHA-256.
  + Xác thực các chứng chỉ số và xử lý kết nối bảo mật thông qua các giao thức mật mã đã chuẩn hóa.
* **Ưu điểm:**
  + Cung cấp bảo mật toàn diện cho dữ liệu trong quá trình truyền tải, đảm bảo thông tin không bị nghe lén hoặc thay đổi bởi bên thứ ba.

**d) Thư viện mã hóa NSS (Network Security Services)**

* **Vai trò tổng quát:**  
  NSS là thư viện mã nguồn mở do Mozilla phát triển, cung cấp bộ API toàn diện cho các chức năng mật mã trong Firefox.
* **Chức năng cung cấp:**
  + **Mã hóa đối xứng:** Hỗ trợ các thuật toán như AES, 3DES.
  + **Dẫn xuất khóa:** Thực hiện PBKDF2 nhằm tạo ra các khóa bảo mật từ mật khẩu người dùng.
  + **Bảo mật truyền tải:** Xử lý giao thức TLS, đảm bảo các kết nối HTTPS được bảo vệ.
  + **Xử lý chứng chỉ số và chữ ký số:** Dùng để xác thực danh tính của các trang web và đảm bảo tính toàn vẹn của dữ liệu.
* **Ưu điểm:**
  + Đồng nhất và tích hợp trên nhiều nền tảng như Windows, macOS, và Linux.
  + Cung cấp một nền tảng bảo mật mạnh mẽ, giúp giảm thiểu rủi ro bảo mật từ việc triển khai mã hóa không nhất quán.

**e) Các thành phần phụ khác: Legacy Encryption (3DES)**

* **Ứng dụng cũ của Firefox:**
  + Trong các phiên bản Firefox cũ, 3DES (Triple DES) được sử dụng để mã hóa dữ liệu đăng nhập.
* **Chi tiết kỹ thuật:**
  + 3DES chạy thuật toán DES ba lần liên tiếp nhằm tăng cường bảo mật so với thuật toán DES truyền thống.
* **Nhược điểm:**
  + So với AES-256, 3DES có tốc độ mã hóa và giải mã chậm hơn và mức độ bảo mật không cao, do đó đã dần được thay thế bởi AES trong các phiên bản hiện đại của Firefox.

### **3.3. Tổng kết quy trình mã hóa**

**A black background with white text

AI-generated content may be incorrect.**

Firefox kết hợp nhiều lớp mật mã để bảo vệ mật khẩu:

+ AES: mã hóa dữ liệu

+ PBKDF2 + SHA-1: sinh khóa từ mật khẩu

+ NSS: thực thi toàn bộ quá trình

***- Nếu người dùng không đặt master password, thì khóa AES trong key4.db không được bảo vệ, tạo lỗ hổng bảo mật tiềm tàng. 🡺 Brute force master password***

## **4. Độ an toàn trong cơ chế lưu trữ mật khẩu của Firefox**

Việc lưu trữ mật khẩu trong trình duyệt là một tiện ích quan trọng, nhưng cũng đặt ra nhiều thách thức về bảo mật. Dưới đây là phân tích về mức độ an toàn của cơ chế lưu trữ mật khẩu trong Firefox.

### **4.1 Cơ chế bảo vệ mật khẩu của Firefox**

Firefox quản lý thông tin đăng nhập người dùng qua hai tệp chính:

* **logins.json**: lưu trữ danh sách mục đăng nhập với các trường encryptedUsername và encryptedPassword đã được mã hóa.
* **key4.db**: CSDL SQLite chứa khóa bí mật dùng để mã hóa/giải mã dữ liệu trong logins.json.
* **Không có Primary Password**: Khóa trong key4.db không được mã hóa thêm, nên kẻ tấn công chỉ cần sao chép cả hai tệp vào một profile Firefox bất kỳ để truy xuất mật khẩu.
* **Có Primary Password**: Firefox sinh khóa từ mật khẩu chính (Primary Password) qua PBKDF2 (SHA‑1, số vòng lặp), rồi dùng khóa này để mã hóa nội dung key4.db. Điều này ngăn chặn truy cập trực tiếp nếu không biết mật khẩu chính.

### **4.2 Tầm quan trọng của Mật khẩu chính (Primary Password)**

* **Bảo vệ kép**: Khi được thiết lập, Primary Password cung cấp lớp bảo vệ bổ sung cho khóa mã hóa, giúp khoá cứng key4.db ngay cả khi profile bị đánh cắp.
* **Rủi ro brute‑force**: Nếu mật khẩu chính quá đơn giản, kẻ tấn công có thể tấn công brute‑force PBKDF2 với SHA‑1 và số vòng lặp mặc định của NSS trong thời gian khả thi.

### **4.3 Công cụ giải mã mật khẩu Firefox**

Một số công cụ mã nguồn mở như **firefox\_decrypt** được thiết kế để trích xuất và giải mã mật khẩu từ profile Mozilla:

* Với **profile không có** Primary Password, công cụ tự động đọc và hiển thị mật khẩu.
* Với **profile có** Primary Password, công cụ yêu cầu nhập đúng mật khẩu chính; nếu không, quá trình giải mã bị dừng.
* **Lưu ý**: firefox\_decrypt không thực hiện brute‑force hay tấn công từ điển, mà chỉ dùng mật khẩu chính đã biết để giải mã.

## **5. Lập trình triển khai**

### **5.1. Giới thiệu về công cụ firefox\_decrypt**

Firefox decrypt là một công cụ mã nguồn mở được viết bằng Python, cho phép trích xuất và giải mã mật khẩu từ hồ sơ người dùng Mozilla như Firefox, Thunderbird, SeaMonkey, v.v.

Công cụ này hoạt động dựa trên việc phân tích hai tệp logins.json và key4.db trong thư mục hồ sơ người dùng, sau đó tiến hành giải mã để trích xuất các thông tin như địa chỉ trang web, tên người dùng và mật khẩu.

* Nếu hồ sơ **không có Primary Password**, công cụ sẽ trực tiếp giải mã và hiển thị thông tin.
* Nếu hồ sơ **được bảo vệ bởi Primary Password**, công cụ sẽ yêu cầu nhập đúng mật khẩu để tiếp tục. Nếu mật khẩu không chính xác hoặc không được cung cấp, công cụ sẽ không thể truy cập dữ liệu.

**Lưu ý:** firefox\_decrypt không hỗ trợ brute-force hay bẻ khóa mật khẩu chính. Nó chỉ làm việc khi người dùng cung cấp đúng thông tin xác thực.

### **5.2. Mô phỏng**

# **KẾT LUẬN**

Trong bối cảnh chuyển đổi số và sự phát triển mạnh mẽ của trí tuệ nhân tạo (AI), việc bảo vệ thông tin cá nhân – đặc biệt là username và password – trở nên quan trọng hơn bao giờ hết. Firefox đã ứng dụng các thuật toán mật mã hiện đại để mã hóa và bảo vệ dữ liệu người dùng, nhưng mức độ an toàn thực tế lại phụ thuộc vào ý thức bảo mật của chính người sử dụng.

Nếu không thiết lập Primary Password, mọi cơ chế mã hóa của Firefox trở nên dễ bị vô hiệu hóa khi kẻ tấn công có quyền truy cập vật lý hoặc từ xa vào hệ thống. Ngược lại, việc sử dụng Primary Password với độ mạnh phù hợp sẽ tăng cường đáng kể khả năng chống lại các hành vi trích xuất và giải mã trái phép, kể cả khi sử dụng công cụ như firefox\_decrypt.

Trong thời đại mà AI có thể hỗ trợ tự động hóa các cuộc tấn công mạng thông minh và tinh vi hơn, vai trò của mật mã học không chỉ là một lớp phòng vệ kỹ thuật, mà còn là nền tảng then chốt để xây dựng hệ thống bảo mật vững chắc. Việc nghiên cứu và hiểu rõ các ứng dụng mật mã, như trong cơ chế lưu trữ mật khẩu của Firefox, mang lại giá trị thực tiễn lớn cho cả người dùng cá nhân lẫn các tổ chức.

Do đó, việc hiểu rõ cơ chế lưu trữ mật khẩu của Firefox và áp dụng đúng các biện pháp bảo vệ như thiết lập Primary Password là bước đi thiết thực và hiệu quả để giảm thiểu rủi ro bảo mật trong kỷ nguyên số.

# **TÀI LIỆU THAM KHẢO**

<https://github.com/unode/firefox_decrypt>

<https://pdfs.semanticscholar.org/a531/bfae3dc920c41dc9cc3b2d55528257823668.pdf>

<https://projekter.aau.dk/projekter/files/260127101/Master_Thesis_ICTE4___4SER_4_6.>

<https://www.google.com/url?q=https://curve.carleton.ca/system/files/etd/8e5c251c-54f8-43b9-a3c0-b847b47b989a/etd_pdf/b73446ef00838b10e69d04e9f2c9921a/mccarney-passwordmanagerscomparativeevaluationdesign.pdf&sa=D&source=apps-viewer-frontend&ust=1745158841299457&usg=AOvVaw2ZHcKJu9SW-vWU1h49s7TP&hl=vi>