**Thành viên nhóm: Hà Xuân Lộc**

**Báo cáo**

**Mã hóa tài khoản cho một trang web bán hàng**

**Chương I:Tổng quan về mã hóa tài khoản**

**1.Khái niệm.**

Mã hóa tài khoản là quá trình chuyển đổi thông tin tài khoản người dùng thành dạng mã hóa, thông qua việc sử dụng một thuật toán mã hóa và một khóa.

Thông tin tài khoản được mã hóa trước khi lưu trữ trong cơ sở dữ liệu hoặc gửi qua mạng, và chỉ có thể được giải mã bằng cách sử dụng khóa và thuật toán tương ứng.

**2.Mục đích.**

Bảo vệ thông tin cá nhân: Mã hóa tài khoản giúp ngăn chặn việc truy cập trái phép vào thông tin cá nhân của người dùng, như tên người dùng và mật khẩu.

Ngăn chặn tấn công từ điển: Mã hóa tài khoản đảm bảo rằng mật khẩu không được lưu trữ dưới dạng văn bản thô trong cơ sở dữ liệu, ngăn chặn các tấn công từ điển nơi kẻ tấn công sử dụng danh sách mật khẩu phổ biến để tấn công vào tài khoản người dùng.

Tăng tính bảo mật: Mã hóa tài khoản là một biện pháp bảo mật quan trọng để ngăn chặn việc đánh cắp thông tin tài khoản và giảm nguy cơ bị xâm phạm an ninh trang web.

**3.Tại sao phải mã hóa tài khoản.**

Mã hóa tài khoản là một phần quan trọng trong việc bảo vệ thông tin cá nhân của người dùng và đảm bảo tính bảo mật của hệ thống. Dưới đây là một số lý do chính tại sao chúng ta cần mã hóa tài khoản trên một trang web:

Bảo mật mật khẩu: Mật khẩu là một yếu tố quan trọng trong việc xác thực người dùng và bảo vệ tài khoản của họ. Mã hóa tài khoản đảm bảo rằng mật khẩu không được lưu trữ dưới dạng văn bản thô trong cơ sở dữ liệu. Điều này ngăn chặn người có quyền truy cập vào cơ sở dữ liệu đọc hoặc sử dụng trực tiếp mật khẩu.

Ngăn chặn tấn công từ điển: Mã hóa tài khoản giúp ngăn chặn tấn công từ điển, nơi kẻ tấn công sử dụng danh sách mật khẩu phổ biến để tấn công vào tài khoản người dùng. Mật khẩu được mã hóa trước khi lưu trữ, do đó ngay cả khi dữ liệu bị đánh cắp, kẻ tấn công sẽ gặp khó khăn trong việc tìm ra mật khẩu gốc.

Bảo vệ thông tin cá nhân: Mã hóa tài khoản đảm bảo rằng thông tin cá nhân của người dùng, như tên người dùng, địa chỉ email và thông tin cá nhân khác, không thể đọc hoặc sử dụng trực tiếp. Ngay cả khi dữ liệu bị đánh cắp, thông tin này vẫn được bảo vệ bởi mã hóa.

Tuân thủ quy định bảo mật: Mã hóa tài khoản giúp trang web tuân thủ các quy định bảo mật và quy tắc an ninh. Nhiều quy định bảo mật, chẳng hạn như các tiêu chuẩn PCI DSS (Payment Card Industry Data Security Standard), yêu cầu việc mã hóa thông tin nhạy cảm như mật khẩu để đảm bảo tính bảo mật của người dùng.

Xây dựng niềm tin từ người dùng: Mã hóa tài khoản là một biện pháp bảo mật quan trọng để xây dựng niềm tin từ người dùng. Khi người dùng biết rằng thông tin cá nhân của họ được bảo vệ một cách an toàn, họ cảm thấy tin tưởng

**4.Hiệu quả.**

Bảo vệ dữ liệu: Mã hóa tài khoản giúp bảo vệ thông tin tài khoản khỏi việc lộ ra ngoài trong trường hợp dữ liệu bị đánh cắp hoặc bị truy cập trái phép.

Tăng độ tin cậy: Mã hóa tài khoản giúp tăng độ tin cậy của người dùng đối với trang web, bởi vì họ biết rằng thông tin cá nhân của họ được bảo vệ một cách an toàn.

Tuân thủ quy định bảo mật: Mã hóa tài khoản cũng giúp trang web tuân thủ các quy định về bảo mật và yêu.

**Chương II:Mã hóa tài khoản bằng thuật toán băm(hashing)**

**1.Khái niệm**

Thuật toán băm (hashing) là một quá trình biến đổi dữ liệu đầu vào thành một chuỗi đại diện có độ dài cố định, gọi là giá trị băm (hash value) hoặc mã băm (hash code). Mục đích chính của thuật toán băm là tạo ra một đại diện duy nhất cho dữ liệu đầu vào, sao cho rất khó khăn để tìm ra hai dữ liệu có cùng giá trị băm.

Một số khái niệm cơ bản liên quan đến thuật toán băm bao gồm:

Đầu vào (Input): Đây là dữ liệu ban đầu mà bạn muốn mã hóa bằng thuật toán băm. Đầu vào có thể là bất kỳ kiểu dữ liệu nào, từ chuỗi ký tự đến dữ liệu nhị phân hoặc tệp tin.

Giá trị băm (Hash value): Đây là chuỗi đại diện duy nhất có độ dài cố định mà thuật toán băm tạo ra từ dữ liệu đầu vào. Giá trị băm thường được biểu diễn dưới dạng một chuỗi hex hoặc một chuỗi nhị phân.

Mã băm (Hash code): Mã băm là cách khác để chỉ giá trị băm. Nó thường được sử dụng để tham khảo đến giá trị băm trong ngữ cảnh lập trình.

Hàm băm (Hash function): Đây là thuật toán được sử dụng để thực hiện quá trình băm. Hàm băm nhận đầu vào và tạo ra giá trị băm tương ứng. Nó phải có các đặc điểm như tính duy nhất, tính khó khăn để xung đột, và tính nhanh.

Xung đột (Collision): Xung đột xảy ra khi hai dữ liệu khác nhau tạo ra cùng một giá trị băm. Một hàm băm tốt sẽ giảm thiểu khả năng xảy ra xung đột.

Tính bảo mật (Security): Tính bảo mật của một hàm băm đánh giá khả năng của nó để ngăn chặn việc tìm ra đầu vào từ giá trị băm. Một hàm băm bảo mật phải đảm bảo tính khó khăn trong việc tìm ra đầu vào từ giá trị băm.

Thuật toán băm được sử dụng rộng rãi trong nhiều lĩnh vực, bao gồm bảo mật mật khẩu, xác thực dữ liệu, kiểm tra tính toàn vẹn và tìm kiếm dữ liệu. Các thuật toán băm phổ biến bao gồm MD5, SHA-1, SHA-256 và bcrypt.

**2.Ưu và nhược diểm của thuật toán băm(hashing)**

Thuật toán băm (hashing) có những ưu điểm và nhược điểm sau đây:

**Ưu điểm của thuật toán băm:**

Tính nhanh: Thuật toán băm thường rất nhanh và hiệu quả trong việc tính toán giá trị băm cho dữ liệu đầu vào.

Độ dài cố định: Giá trị băm có độ dài cố định, không phụ thuộc vào độ dài của dữ liệu đầu vào. Điều này giúp tiết kiệm không gian lưu trữ và dễ dàng trong việc so sánh các giá trị băm.

Bảo mật một chiều: Việc mã hóa bằng hàm băm là một chiều, tức là khó khăn để phục hồi dữ liệu gốc từ giá trị băm. Điều này bảo vệ tính bảo mật của thông tin, vì người khác không thể dễ dàng tìm ra dữ liệu gốc chỉ từ giá trị băm.

Xử lý lỗi thấp: Thuật toán băm thường có tỷ lệ xung đột (collision) thấp, tức là khó xảy ra trường hợp hai dữ liệu khác nhau tạo ra cùng một giá trị băm.

**Nhược điểm của thuật toán băm:**

Không thể phục hồi dữ liệu: Do tính chất một chiều của hàm băm, không thể phục hồi dữ liệu gốc từ giá trị băm. Điều này có thể gây khó khăn trong một số trường hợp, ví dụ như khi bạn cần phục hồi thông tin ban đầu từ giá trị băm.

Xung đột: Mặc dù khả năng xung đột trong thuật toán băm là rất thấp, nhưng không thể tránh khỏi trường hợp hai dữ liệu khác nhau tạo ra cùng một giá trị băm. Xung đột có thể gây ra các vấn đề về tính toàn vẹn và bảo mật dữ liệu.

Tấn công từ điển: Một cuộc tấn công từ điển có thể được thực hiện bằng cách tính toán giá trị băm cho các giá trị đầu vào tiềm năng và so sánh chúng với giá trị băm đã biết. Điều này có thể cho phép tấn công viên tìm ra giá trị gốc tương ứng với giá trị băm.

Sự phụ thuộc vào chất lượng hàm băm: Hiệu suất và tính bảo mật của thuật toán băm phụ thuộc vào chất lượng của hàm băm được sử dụng. Một hàm băm yếu hoặc bị tấn công có thể làm giảm tính bảo mật của thuật toán băm.

Để đảm bảo tính bảo mật cao và giảm thiểu các vấn đề tiềm ẩn, các thuật toán băm cần được lựa chọn và sử dụng cẩn thận, kết hợp với các biện pháp bổ sung như salt (muối) và pepper (ớt) để tăng cường tính bảo mật.

**3.Áp dụng thuật toán băm(hashing)**

Các bước thực hiện của thuật toán băm (hashing) bao gồm:

Chọn thuật toán băm: Lựa chọn một thuật toán băm phù hợp cho nhu cầu cụ thể. Các thuật toán phổ biến bao gồm MD5, SHA-1, SHA-256 và bcrypt.

Chuẩn bị dữ liệu đầu vào: Chuẩn bị dữ liệu mà bạn muốn băm. Đầu vào có thể là một chuỗi ký tự, dữ liệu nhị phân hoặc tệp tin.

Tạo hàm băm: Tạo một hàm băm từ thuật toán băm đã chọn. Hàm băm sẽ được sử dụng để biến đổi dữ liệu đầu vào thành giá trị băm tương ứng.

Mã hóa dữ liệu đầu vào: Áp dụng hàm băm cho dữ liệu đầu vào. Hàm băm sẽ thực hiện các phép biến đổi trên dữ liệu để tạo ra giá trị băm.

Nhận giá trị băm: Nhận giá trị băm sau khi áp dụng hàm băm lên dữ liệu đầu vào. Giá trị băm sẽ là đại diện duy nhất của dữ liệu đầu vào.

Sử dụng giá trị băm: Giá trị băm có thể được sử dụng để kiểm tra tính toàn vẹn của dữ liệu, so sánh dữ liệu, xác thực hay mã hóa mật khẩu.

Đối với một số thuật toán băm, như bcrypt, còn có các bước bổ sung như thêm "salt" (muối) để tăng cường tính bảo mật. Salt là một chuỗi ngẫu nhiên được thêm vào dữ liệu đầu vào trước khi áp dụng hàm băm. Việc sử dụng salt giúp ngăn chặn các cuộc tấn công từ điển và tạo ra giá trị băm khác nhau cho cùng một dữ liệu đầu vào.

**Chương III:Tổng kết**

**1.Tổng kết về thuật toán băm(hashing)**

Thuật toán băm (hashing) là một quá trình biến đổi dữ liệu đầu vào thành một chuỗi đại diện duy nhất có độ dài cố định gọi là giá trị băm. Mục đích chính của thuật toán băm là tạo ra một mã đại diện cho dữ liệu mà rất khó khăn để tìm ra dữ liệu gốc từ mã này.

Thuật toán băm có nhiều ưu điểm, bao gồm tính nhanh, độ dài cố định của giá trị băm, tính bảo mật một chiều và khả năng xử lý lỗi thấp. Nó được sử dụng rộng rãi trong các lĩnh vực như bảo mật mật khẩu, xác thực dữ liệu, kiểm tra tính toàn vẹn và tìm kiếm dữ liệu.

Tuy nhiên, thuật toán băm cũng có nhược điểm. Đầu tiên, nó là một quá trình một chiều, không thể phục hồi dữ liệu gốc từ giá trị băm. Thứ hai, mặc dù xác suất xung đột là rất thấp, nhưng không thể tránh khỏi trường hợp hai dữ liệu khác nhau tạo ra cùng một giá trị băm. Cuối cùng, hiệu suất và tính bảo mật của thuật toán băm phụ thuộc vào chất lượng của hàm băm được sử dụng, và một hàm băm yếu có thể làm giảm tính bảo mật của thuật toán băm.

Để đảm bảo tính bảo mật và giảm thiểu các vấn đề tiềm ẩn, lựa chọn và sử dụng thuật toán băm cần được thực hiện cẩn thận, kết hợp với các biện pháp bổ sung như sử dụng salt và pepper để tăng cường tính bảo mật.