BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC ĐÀ NẴNG

NGUYỄN CỬU THỊ ÁNH MAI

NGHIÊN CỨU GIẢI PHÁP BẢO MẬT CƠ SỞ DỮ LIỆU SQL SERVER BẰNG PHƯƠNG PHÁP MÃ HÓA

Chuyên ngành: Khoa học máy tính
Mã số: 60.48.01

TÓM TẮT LUẬN VĂN THẠC SĨ KỸ THUẬT

Người hướng dẫn khoa học: PGS.TSKH. TRẦN QUỐC CHIẾN

ĐÀ NẴNG, 2010

Công trình được hoàn thành tại
ĐẠI HỌC ĐÀ NẴNG

Người hướng dẫn khoa học: PGS.TSKH. TRẦN QUỐC CHIẾN

Phản biện 1: TS. NGUYỄN THANH BÌNH

Phản biện 2: TS. TRƯƠNG CÔNG TUẤN

Luận văn được bảo vệ tại Hội đồng chấm Luận văn tốt nghiệp thạc sĩ kỹ thuật họp tại Đại học Đà Nẵng vào ngày 15 tháng 10 năm 2010.

**** Có thể tìm hiểu luận văn tại***

- Trung tâm Thông tin - Học liệu, Đại học Đà Nẵng
- Trung tâm Học liệu, Đại học Đà Nẵng

MỞ ĐẦU

1. Lý do chọn đề tài

Thông tin luôn là một tài sản vô giá của doanh nghiệp và cần được bảo vệ bằng mọi giá. Tuy nhiên, với những đòi hỏi ngày càng gắt gao của môi trường kinh doanh yêu cầu doanh nghiệp phải năng động chia sẻ thông tin của mình cho nhiều đối tượng khác nhau qua Internet hay Intranet, việc bảo vệ thông tin trở nên ngày càng quan trọng và khó khăn hơn bao giờ hết.

Hầu hết các doanh nghiệp ngày nay đều sử dụng các hệ quản trị cơ sở dữ liệu (CSDL) để lưu trữ tập trung tất cả các thông tin quý giá của mình. Hệ thống này sẽ là tiêu điểm tấn công của những kẻ xấu. Ở mức độ nhẹ, các tấn công sẽ làm hệ thống CSDL bị hỏng hóc, hoạt động không ổn định, mất mát dữ liệu làm cho các giao dịch hàng ngày của doanh nghiệp bị đình trệ. Nghiêm trọng hơn, các thông tin sống còn của doanh nghiệp bị tiết lộ (như chiến lược kinh doanh, các thông tin về khách hàng, nhà cung cấp, tài chánh, mức lương nhân viên,…) và được đem bán cho các doanh nghiệp đối thủ. Có thể nói là thiệt hại của việc thông tin bị rò rỉ là vô cùng kinh khủng. Đó sẽ là một đòn chí mạng đối với uy tín của doanh nghiệp đối với khách hàng và các đối tác.

Vì vậy vấn đề bảo mật CSDL trở nên cấp bách và rất cần thiết cho tất cả mọi người và nhất là đối với các cơ quan lưu trữ những dữ liệu quan trọng. Một trong những cách bảo mật CSDL là sử dụng phương pháp mã hóa. Đây cũng là lý do tôi chọn đề tài: "*Nghiên cứu giải pháp bảo mật cơ sở dữ liệu SQL Server bằng phương pháp mã hóa*".

2. Mục tiêu và nhiệm vụ

- Nghiên cứu, tìm hiểu hệ quản trị CSDL SQL Server 2008 nhằm tìm ra các giải pháp bảo mật của hệ quản trị CSDL để giải quyết ba vấn đề cơ bản là tính bí mật, tính toàn vẹn của dữ liệu và tính sẵn sàng của hệ thống dữ liệu.
- Nghiên cứu các tính năng mã hóa trong SQL Server 2008.
- Dựa trên các tính năng này xây dựng chương trình thực hiện chức năng mã hóa CSDL, xây dựng cơ chế phân quyền đối với dữ liệu đã được mã hóa, thực hiện các thao tác điều khiển dữ liệu.
- Tính năng mã hóa trong SQL Server 2008 chỉ có thể mã hóa từng cột. Điều này khiến cho việc muốn bảo mật thông tin của một đối tượng nào đó thì bắt buộc phải mã hóa toàn bộ các cột lưu trữ dữ liệu của tất cả các đối tượng trong bảng. Để khắc phục khuyết điểm này tôi có ý tưởng xây dựng chương trình thực hiện mã hóa dòng dữ liệu của bảng. Chỉ những dòng dữ liệu cần được che dấu sẽ được thực hiện bằng cách mã hóa, dữ liệu còn lại vẫn hiển thị bình thường.

3. Đối tượng và phạm vi nghiên cứu

Đối tượng trọng tâm của đề tài là mô hình mã hóa với những nội dung cụ thể của nó gồm: các kỹ thuật mã hóa và phương hướng mã hóa cho CSDL lưu trong SQL Server 2008.

Phạm vi nghiên cứu của đề tài bao gồm nghiên cứu lý thuyết và xây dựng chương trình mã hóa CSDL SQL Server 2008. Giới hạn

nghiên cứu ứng dụng trong phạm vi của hệ quản trị CSDL SQL Server 2008.

4. Phương pháp nghiên cứu

Với các mục tiêu trên tôi chọn phương pháp nghiên cứu lý thuyết kết hợp thực nghiệm. Đề tài dự định tiến hành theo các bước sau:

- Nghiên cứu lý thuyết về kỹ thuật mã hóa trong SQL SERVER 2008.
- Nghiên cứu phương pháp xây dựng tầng mã hóa sử dụng cơ chế có sẵn trong CSDL SQL SERVER 2008.
- Nghiên cứu các thuật toán mã hóa được sử dụng trong các kỹ thuật mã hóa của SQL Server 2008.
- Nghiên cứu giải pháp mã hóa dữ liệu ở mức ứng dụng, giải pháp này xử lý mã hóa dữ liệu trước khi truyền dữ liệu vào CSDL SQL SERVER 2008.
- Cài đặt chương trình mã hóa dòng dữ liệu bằng ngôn ngữ lập trình Java.

5. Ý nghĩa khoa học và thực tiễn của đề tài

Các kỹ thuật mã hóa của SQL Server 2008 tạo nên một mô hình tầng mã hóa. Mô hình này truy xuất dữ liệu từ bảng ảo và lưu dữ liệu mã hóa vào bảng gốc.

Ngoài cách sử dụng cơ chế có sẵn trong SQL Server 2008, mô hình tầng mã hóa này còn được thực hiện bằng cách mã hóa dữ liệu bởi ứng dụng trước khi lưu dữ liệu vào CSDL SQL Server 2008.

Việc mã hóa dữ liệu trong CSDL là một giải pháp của tương lai. Đến một lúc nào đó chúng ta sẽ không còn lưu dữ liệu tại một máy cố định mà hướng đến việc lưu tất cả dữ liệu trên mạng Internet, việc mất mát và để lộ thông tin là điều không tránh khỏi. Vì vậy chỉ một giải pháp đó là mã hóa chúng để bất cứ ai cũng không đọc được thông tin này.

Các nghiên cứu của luận văn góp phần chuyển tải thông tin về các kỹ thuật mã hóa dữ liệu đến người xây dựng ứng dụng quản lý CSDL. Giúp cải thiện tư duy bảo mật dữ liệu của bản thân, vận dụng có hiệu quả cách thức đảm bảo an toàn dữ liệu.

6. Bố cục luận văn

Luận văn được bố cục trong ba chương.

Chương 1: Mã hóa dữ liệu trong SQL SERVER 2008.

Trình bày các tính năng mã hóa trong SQL Server 2008. Mỗi tính năng có cách thực hiện và ưu nhược điểm riêng. Ngoài ra trong chương này còn trình bày về mô hình tầng mã hóa trong SQL Server 2008.

Chương 2: Thuật toán mã hóa dữ liệu trong SQL SERVER 2008.

Mô tả các thuật toán mã hóa dữ liệu được sử dụng trong SQL Server 2008.

Chương 3: Ứng dụng mã hóa dòng dữ liệu với java.

Trình bày lý thuyết mã hóa của Java và ý tưởng về mã hóa dòng dữ liệu được lưu trữ trong SQL Server 2008.

CHƯƠNG 1

MÃ HÓA DỮ LIỆU TRONG SQL SERVER 2008

1.1. Các khái niệm cơ bản của mã hóa dữ liệu trong SQL Server

Trong bối cảnh bảo mật dữ liệu, quá trình mã hóa được sử dụng để chuyển đổi hoặc mã hóa dữ liệu gốc thành dữ liệu không thể

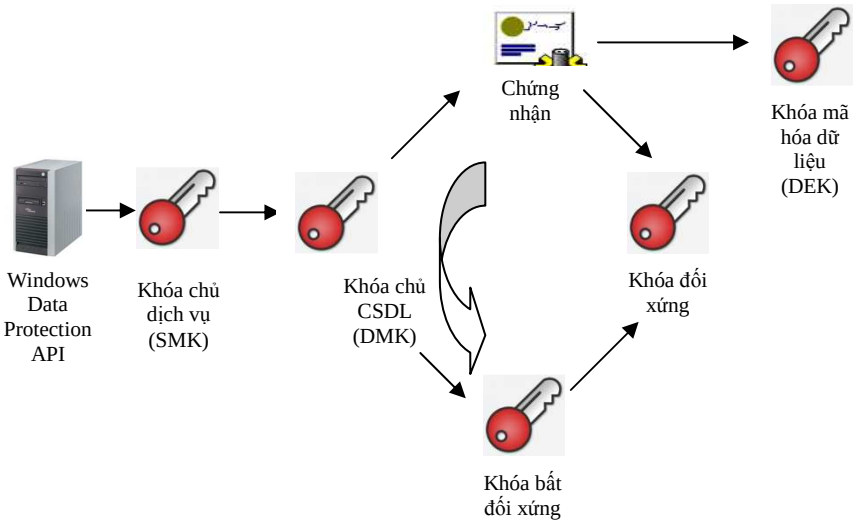
đọc được gọi là văn bản mã sau đó giải mã nó trở lại thành một định dạng có thể đọc được gọi là văn bản rõ.

Phần này trình bày về các khóa được sử dụng để thực hiện chức năng mã hóa trong SQL Server và mối quan hệ giữa các khóa với nhau trong hệ thống khóa cấp bậc. Các khóa này phải đảm bảo rằng chúng cung cấp một mức độ bảo vệ nhất quán, lâu dài đồng thời luôn luôn an toàn với quá trình sao lưu dữ liệu.

1.1.1. Khóa

Thành phần chính của mã hóa là khóa. Mỗi khóa chứa thuật toán, trình tự thực hiện chức năng mã hóa khác nhau do SQL Server cung cấp để mã hóa và giải mã dữ liệu.

1.1.2. Hệ thống phân cấp khóa mã hóa



Hình 1.1. Hệ thống khóa phân cấp.

Hệ thống phân cấp này cung cấp một cơ sở bảo mật cao cho dữ liệu nhạy cảm. Tại phía trên cùng của hệ thống phân cấp này là khóa chủ dịch vụ SMK, khóa này thực hiện bảo vệ khóa chủ CSDL DMK tại mỗi ứng dụng CSDL trong SQL Server. Khóa chủ CSDL DMK được sử dụng để mã hóa các khóa riêng như khóa bất đối xứng và chứng nhận trong CSDL. Khóa bất đối xứng, chứng nhận được sử dụng để bảo vệ khóa riêng tư khác, đó là khóa đối xứng và dữ liệu chứa trong CSDL. Các khóa đối xứng trong CSDL được sử dụng để bảo vệ các khóa đối xứng khác cũng như dữ liệu trong CSDL.

1.1.2.1. Khóa chủ dịch SMK

1.1.2.2. Khóa chủ cơ sở dữ liệu DMK

1.1.2.3. Khóa bất đối xứng

1.1.2.4. Chứng nhận

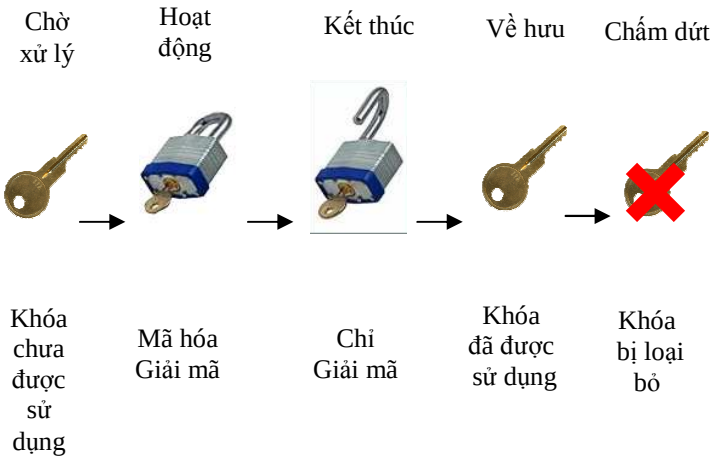
1.1.2.5. Khóa đối xứng

1.1.2.6. Khóa mã hóa dữ liệu DEK

1.1.2.7. Mật khẩu

1.1.3. Bảo vệ khóa

Khóa mã hóa và mật khẩu bảo vệ khóa đảm bảo tính năng bảo mật dữ liệu nhạy cảm. Thường xuyên bảo vệ các khóa và mật khẩu làm giảm sự xuất hiện của việc phá hủy dữ liệu mã hóa thông qua sự theo dõi các giá trị mã hóa của tin tặc. Sự bảo vệ này được điều khiển thông qua một vòng đời của mỗi khóa, minh họa trong hình 1.2 sau đây.



Hình 1.2. Vòng đời của khóa

1.1.4. Sao lưu khóa

1.2. Các thuật toán mã hóa dữ liệu được sử dụng trong SQL Server

1.2.1. Thuật toán đối xứng

1.2.1.1. Thuật toán DES (Data Encryption Standard)

1.2.1.2. Thuật toán AES (Advanced Encryption Standard)

1.2.2. Thuật toán bất đối xứng RSA

1.2.3. Thuật toán băm

1.3. Mã hóa cột

1.3.1. Cơ bản mã hóa cột

Ở cấp độ ô, mức độ tốt nhất của mã hóa, mỗi ô chứa dữ liệu mã hóa được bảo vệ bởi một khóa cụ thể do người dùng thực hiện mã hóa. Giải mã được thực hiện thông qua việc sử dụng cùng một khóa hoặc một khóa công khai tùy thuộc vào phương pháp mã hóa được áp dụng.

Ngoài ra mã hóa cột là mã hóa tất cả các ô trong một cột duy nhất cùng với khóa và cho phép giải mã với khóa này sau đó cấp quyền cho các thành viên với vai trò của CSDL.

1.3.2. Ưu và nhược điểm của mã hóa cột

1.3.2.1. Ưu điểm

- Mã hóa cột cung cấp mã hóa ở mức độ tốt hơn nhiều so với mã hóa tập tin sao lưu dữ liệu. Nó cung cấp phương tiện để mã hóa một cột duy nhất trong bảng từ một cột khác.
- An toàn - yếu tố dữ liệu được mã hóa duy trì ở trạng thái đó cho đến khi nó giải mã.
- Người sử dụng - người sử dụng có thể được cấp quyền truy cập vào khóa mã hóa và giải mã dữ liệu.

1.3.2.2. Nhược điểm

- Hạn chế kiểu dữ liệu - thực hiện mã hóa cột yêu cầu sửa đổi kiểu dữ liệu. Tất cả các dữ liệu mã hóa phải được lưu trữ với kiểu dữ liệu varbinary.
- Trong quá trình quét bảng dữ liệu, các giá trị bị mã hóa một cách gượng ép. Khóa chính và chỉ mục sau khi mã hóa không còn sử dụng được.
- Tổng chi phí xử lý – các xử lý cho quá trình mã hóa và giải mã tốn chi phí cao.

1.3.3. Mã hóa một khối lượng lớn dữ liệu

1.3.4. Các bước thực hiện mã hóa cột

1.3.4.1. Xác định thuật toán mã hóa

Có nhiều thuật toán có sẵn được chọn để mã hóa dữ liệu. Tùy thuộc vào người sử dụng mà chọn thuật toán phù hợp. Đối với mã hóa dữ liệu sử dụng thuật toán bất đối xứng dựa trên một thuật toán phức tạp và cung cấp một mức độ bảo vệ rất cao. Còn về mã hóa đối xứng thì sức mạnh của mã hóa này phụ thuộc vào độ dài của các khóa được sử dụng. Các khóa có kích thước dài hơn cung cấp một cấp độ bảo mật cao hơn nhưng đi kèm với một chi phí xử lý cao hơn. Các thuật toán mã hóa đối xứng nói chung ít phức tạp và do đó yếu hơn so với mã hóa bất đối xứng nhưng kết quả xử lý nhanh hơn.

1.3.4.2. Thực hiện hệ thống khóa cấp bậc

Theo hệ thống phân cấp khóa mã hóa nêu trên, để mã hóa dữ liệu bằng khóa đối xứng, các bước thực hiện như sau:

Bước đầu tiên sẽ tạo ra một khóa DMK. Điều này được thực hiện bằng cách sử dụng câu lệnh CREATE MASTER KEY.

Bước tiếp theo là tạo ra một chứng nhận, chứng nhận này được bảo vệ bởi khóa chủ CSDL quan trọng. Tất cả các chứng nhận được tự tạo ra trong SQL Server.

Khóa cuối cùng trong hệ thống cấp bậc là khóa đối xứng sẽ được sử dụng để mã hóa các dữ liệu nhạy cảm. Việc lựa chọn một khóa đối xứng dựa trên các thuật toán mạnh và nhanh. Khóa đối xứng được tạo ra thông qua việc thực hiện các câu lệnh CREATE SYMMETRIC KEY.

1.3.4.3. Thay đổi cấu trúc dữ liệu

Mã hóa cột đòi hỏi giá trị mã hóa được lưu trữ trong một cột với kiểu dữ liệu là varbinary. Để tiến hành mã hóa cột, tại bước này

phải tạo một cột dữ liệu mới với kiểu dữ liệu varbinary. Cột này để lưu dữ liệu mã hóa.

1.3.4.4. Mã hóa cột

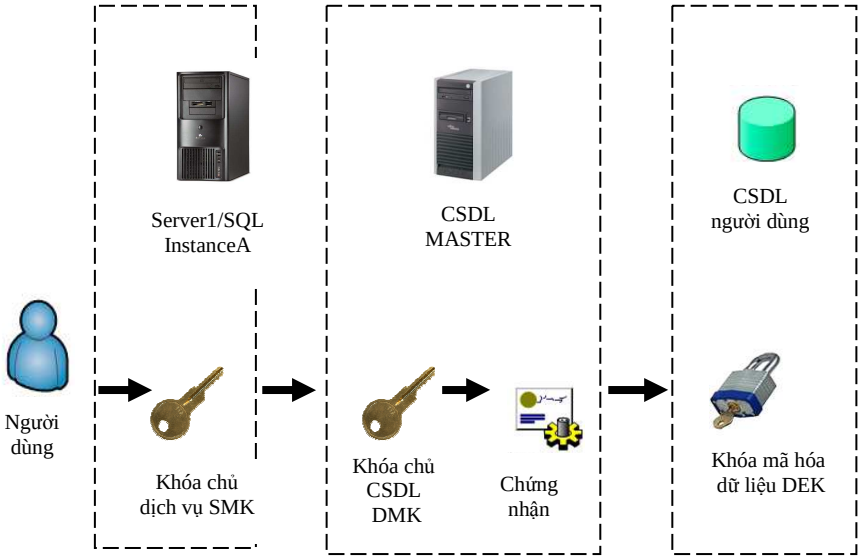
Sau khi đã tạo mới cột lưu dữ liệu mã hóa, sử dụng một trong bốn phương thức: EncryptByAsymKey, EncryptByCert, EncryptByKey và EncryptByPassphrase để mã hóa dữ liệu cột và lưu dữ liệu mã hóa vào cột mới tạo ra với kiểu dữ liệu là varbinary.

1.4. Mã hóa dữ liệu trong suốt TDE

1.4.1. Cách làm việc của TDE

Mục đích cụ thể của TDE là để bảo vệ dữ liệu bằng cách mã hóa các tập tin vật lý của CSDL, chứ không phải là mã hóa dữ liệu. Những tập tin vật lý bao gồm các tập tin CSDL (.mdf), các tập tin giao dịch log (. ldf) và tập tin sao lưu (.bak).

Việc bảo vệ các tập tin CSDL được thực hiện thông qua một hệ thống khóa phân cấp tồn tại bên ngoài CSDL trong đó đã được TDE kích hoạt. Trong hình 1.3 dưới đây sẽ minh họa hệ thống khóa cấp bậc và vị trí yêu cầu của mỗi khóa.



Hình 1.3. Hệ thống khóa cấp bậc

1.4.2. Ưu và nhược điểm của TDE

1.4.2.1. Ưu điểm

1.4.2.2. Nhược điểm

1.4.3. Các bước thực hiện TDE

1.4.3.1. Cân nhắc trước khi thực hiện TDE

1.4.3.2. Các bước thực hiện TDE

- Sao lưu trước khi mã hóa.
- Thực hiện mã hóa TDE
  - Bước 1: tạo khóa DMK cho CSDL Master bằng cách sử dụng câu lệnh CREATE MASTER KEY.
  - Bước 2: tạo ra một chứng nhận, khóa này được bảo vệ bởi khóa DMK của CSDL Master. Lúc này nên thực hiện sao lưu khóa chứng nhận với khóa riêng của nó.

  - Bước 3: tạo khóa mã hóa dữ liệu DEK, khóa này được sử dụng để thực hiện chức năng mã hóa cho các tập tin vật lý của CSDL này.
  - Bước 4: Thiết lập quá trình mã hóa TDE bằng cách thực hiện lệnh ALTER DATABASE với đối số SET ENCRYPTION ON.

1.4.4. Kiểm nghiệm TDE

1.5. Mã hóa một chiều

Mã hóa dữ liệu một chiều rất đơn giản. Giá trị được mã hóa và lưu trữ trong bảng dữ liệu. Tuy nhiên không giống như mã hóa cột, khóa không được tạo ra và dữ liệu luôn được duy trì ở trạng thái bảo vệ. Không xảy ra quá trình giải mã với phương thức mã hóa một chiều.

1.5.1. Cách thức hoạt động của mã hóa một chiều

Trong SQL Server, mã hóa một chiều được hoàn thành thông qua sử dụng phương thức Hashbytes. Phương thức sử dụng một thuật toán để tạo nên giá trị băm. Không giống như mã hóa cột, nó tạo ra một giá trị băm duy nhất mỗi lần mã hóa dữ liệu. Phương thức HashBytes trả về giá trị băm.

1.5.2. Ưu và nhược điểm của mã hóa một chiều

1.5.2.1. Ưu điểm

1.5.2.2. Nhược điểm

1.5.3. Các lỗ hổng trong mã hóa một chiều

1.5.3.1. Lỗ hổng tấn công từ điển

Cuộc tấn công từ điển là cuộc tấn công mà trong đó một danh sách các giá trị băm được tạo ra và so với các giá trị băm lưu trữ

trong bảng dữ liệu mục tiêu. Phương pháp này thường sử dụng để cố gắng làm lộ mật khẩu được bảo vệ bằng cách sử dụng mã hóa một chiều.

Một cuộc tấn công từ điển lợi dụng tính chất cố hữu của mã hóa một chiều bằng cách thực hiện cùng một hành động được sử dụng khi người dùng tìm kiếm dữ liệu mã hóa một chiều nhưng trên một quy mô lớn hơn.

Nếu người quản trị CSDL thêm vào một loạt các ký tự trước khi nó được mã hóa, kết quả giá trị băm sẽ khác hơn kết quả giá trị băm mã hóa trên và sẽ tăng số kết hợp ký tự có thể yêu cầu một cuộc tấn công tích cực.

1.5.3.2. Lỗ hổng tấn công bảng cầu vồng

Nhân vật chính trong tấn công này là bảng cầu vồng. Bảng cầu vồng bao gồm một loạt các hàng đang nắm giữ dữ liệu của hai cột. Cột đầu tiên chứa các giá trị dữ liệu gốc đang tìm kiếm. Cột thứ hai chứa một giá trị băm kết thúc của một chuỗi giảm. Một chuỗi giảm là kết quả của việc đưa giá trị dữ liệu gốc vào trong cột đầu tiên của bảng cầu vồng và tạo ra một giá trị băm ban đầu, sau đó, một phần của giá trị băm ban đầu tạo ra một giá trị băm khác. Quá trình này tiếp tục lặp lại một số lần cho đến khi một giá trị băm kết thúc được hình thành.

1.5.4. Giảm tính dễ tổn thương bằng cách ướp muối dữ liệu gốc

"Muối" trong mã hóa làm gia tăng tính bảo mật. Một giá trị băm của mã hóa một chiều dễ bị tổn thương bởi tấn công từ điển và bảng cầu vồng. Nhưng thêm "muối" vào dữ liệu gốc trước khi nó được mã hóa, kết quả tạo nên một giá trị băm rất đàn hồi đối với các

cuộc tấn công. "Ướp muối" vào làm cho dữ liệu gốc phức tạp hơn và phá vỡ dự kiến mô hình được dự đoán của kẻ tấn công.

Các bước thực hiện mã hóa một chiều:

- B1. Sao lưu CSDL trước khi thực hiện mã hóa dữ liệu một chiều.
- B2. Tạo cột băm với kiểu dữ liệu varbinary để lưu trữ các giá trị băm của dữ liệu cần mã hóa.
- B3. "Ướp muối" dữ liệu gốc trước khi băm và sau đó sử dụng phương pháp HashBytes mã hóa dữ liệu một chiều.
- B4. Kiểm tra và xác minh kiến trúc mã hóa một chiều.

Để biết được quá trình mã hóa có thành công hay không? Có thể thực thi câu lệnh Select để lọc dữ liệu, kết quả tùy thuộc vào cột mã hóa mới tạo ra.

- B5. Xóa cột lưu trữ dữ liệu gốc đã được mã hóa một chiều.

Nên chắc chắn rằng quá trình mã hóa đã thành công, bây giờ có thể loại bỏ cột chứa thông tin nhạy cảm.

1.6. Tầng mã hóa

Mô hình tầng mã hóa giải quyết vấn đề mã hóa ở mức ứng dụng. Giải pháp này xử lý mã hóa dữ liệu trước khi truyền dữ liệu vào CSDL. Những vấn đề về quản lý khóa và quyền truy cập được hỗ trợ bởi ứng dụng. Truy vấn dữ liệu đến CSDL sẽ trả về dữ liệu ở dạng mã hóa và dữ liệu này sẽ được giải mã bởi ứng dụng.

Một giải pháp bảo mật CSDL tối ưu cần hỗ trợ các yếu tố chính sau:

- Hỗ trợ mã hóa tại các mức dữ liệu cấp bảng, cột, hàng.
- Hỗ trợ chính sách an ninh phân quyền truy cập đến mức dữ liệu cột.

- Cơ chế mã hóa không ảnh hưởng đến các ứng dụng hiện tại.

1.6.1. Xây dựng tầng CSDL trung gian

Trong mô hình này, một CSDL trung gian được xây dựng giữa ứng dụng và CSDL gốc. CSDL trung gian này có vai trò mã hóa dữ liệu trước khi cập nhật vào CSDL gốc, đồng thời giải mã dữ liệu trước khi cung cấp cho ứng dụng. CSDL trung gian đồng thời cung cấp thêm các chức năng quản lý khóa, xác thực người dùng và cấp phép truy cập.

1.6.2. Tầng mã hóa với SQL Server

Để xây dựng tầng mã hóa trong SQL Server 2008, thực hiện dựa trên cơ chế sau:

- Các hàm Stored Procedure trong CSDL cho chức năng mã hóa và giải mã.
- Sử dụng cơ chế View trong CSDL tạo các bảng ảo, thay thế các bảng thật đã được mã hóa.
- Cơ chế "instead of" trigger được sử dụng nhằm tự động hóa quá trình mã hóa từ View đến bảng gốc.

Trong mô hình này, dữ liệu trong các bảng gốc sẽ được mã hóa, tên của bảng gốc được thay đổi. Một bảng ảo (View) được tạo ra mang tên của bảng gốc, ứng dụng sẽ truy cập đến bảng ảo này.

1.7. Kết luận

CHƯƠNG 2
THUẬT TOÁN MÃ HÓA DỮ LIỆU

2.1. Thuật toán DES

DES là thuật toán mã hóa một khối dữ liệu 64 bit. Dữ liệu đầu vào là khối bản rõ 64 bit và dữ liệu đầu ra là một khối bản mã 64 bit. Cả mã hóa và giải mã sử dụng cùng một thuật toán và khóa.

2.1.1. Các bước thuật toán DES

DES mã hóa một xâu bit x có độ dài 64 bit bằng một khóa 54 bit. Bản mã nhận được cũng là một xâu bit có độ dài 64 bit.

Thuật toán tiến hành theo 3 giai đoạn:

- Bước 1: với bản rõ cho trước x, một xâu bit x_0 sẽ được xây dựng bằng cách hoán vị các bit của x theo phép hoán vị cố định ban đầu IP, x_0 được viết: x_0= IP(X) = L_0R_0, trong đó L_0 gồm 32 bit đầu và R_0 là 32 bit cuối.
- Bước 2 : sau đó tính toán 16 lần lặp theo một hàm xác định, sẽ tính L_iR_i, $1 \le i \le 16$ theo quy tắc sau:
  - $L_i = R_{i-1}$
  - $R_i = L_{i-1} \oplus F(R_{i-1},K_i)$

Trong đó $\oplus$ kí hiệu phép hoặc loại trừ của hai xâu bit. F là một hàm sẽ được mô tả tại mục 3.1.4. Các khóa $K_1,K_2, \ldots ,K_{16}$ là các xâu bit có độ dài 48 bit được tạo ra theo thuật toán tạo khóa của DES.

- Bước 3: áp dụng phép hoán vị ngược IP^{-1} cho xâu bit $R_{16}L_{16}$, thu được bản mã y. Tức là $y = IP^{-1}(R_{16}L_{16})$. Trong trường hợp này phải chú ý thứ tự đã đảo của L_{16} và R_{16}.

2.1.2. Hoán vị khởi đầu IP

2.1.3. Tính khóa

2.1.4. Hàm F và hộp S

2.1.4.1. Hoán vị mở rộng (Hộp E)

2.1.4.2. Hộp S

2.1.4.3. Hộp hoán vị P

2.1.5. Giải mã

2.1.6. Nhận xét

2.2. Thuật toán TRIPLE DES

Tripple DES hay còn gọi là 3DES thực ra là mã hóa cùng 1 thông tin qua 3 lần mã hóa DES với 3 khóa khác nhau. Do đó chiều dài khóa sẽ lớn hơn và an toàn hơn so với DES.

2.3. Thuật toán AES

2.3.1. Giới thiệu

2.3.2. Các khái niệm và ký hiệu

2.3.3. Các hàm, ký hiệu và các tham số của thuật toán

2.3.4. Thuật toán

Độ dài của input, output và các trạng thái (state) của chuẩn mã hóa cao cấp AES là 128 bit tương ứng với giá trị của Nb = 4 (là số lượng các word 32-bit và cũng là số cột của mỗi trạng thái). Khóa của AES có độ dài là 128, 192 hoặc 256 bit tương ứng với các giá trị của Nk là 4, 6, hoặc 8 và cũng là số cột của khóa mã hóa.

Cả quá trình mã hóa và giải mã AES sử dụng một hàm lặp kết hợp của bốn hàm biến đổi sau: 1) biến đổi thay thế byte sử dụng một bảng thế (S-box), 2) dịch các hàng của mảng trạng thái với số lần

dịch của mỗi hàng là khác nhau, 3) kết hợp dữ liệu của mỗi cột trong mảng trạng thái và 4) cộng một khóa RoundKey vào trạng thái.

2.3.4.1. Mã hóa

Có thể thấy tất cả các vòng đều thực hiện công việc giống nhau dựa trên 4 hàm (theo thứ tự) SubBytes(), ShiftRows(), MixColumns() và AddRoundKey() trừ vòng cuối cùng bỏ qua việc thực hiện hàm MixColumns().

2.3.4.2. Giải mã

Thuật toán giải mã khá giống với thuật toán mã hóa về mặt cấu trúc nhưng 4 hàm cơ bản sử dụng là các hàm ngược của các hàm trong thuật toán giải mã.

2.3.5. Nhận xét

2.4. Thuật toán RSA

Thuật toán mã hóa bất đối xứng RSA sử dụng một loạt các phương pháp tính toán dựa trên cặp khóa tư nhân và khóa công khai thay vì phương thức khối/dòng được định nghĩa trong mã hóa đối xứng.

Thuật toán bất đối xứng, nói chung, mạnh hơn thuật toán đối xứng, nhưng chúng đòi hỏi tốn nhiều bộ nhớ.

Đoạn tin được mã hóa từng khối, với mỗi khối có giá trị nhỏ hơn N (N được miêu tả ở B1). Cho khối văn bản rõ M (M<N) và khối bảo mật C, việc mã hóa và giải mã gồm các bước sau:

- B1: A chọn 2 số nguyên tố ngẫu nhiên p và q. N là tích 2 số p và q. N là khóa công cộng. A gởi cho B giá trị N.
- B2: A chọn một số nguyên tố khác e, e cũng là một phần của khóa công cộng sao cho ed=1(mod (p-1)(q-1)). A gởi cho B e còn giữ d làm khóa bí mật cho mình.

- B3: Sau khi nhận các giá trị N và e, B bắt đầu mã hóa dữ liệu theo công thức: $C=M^e(\text{mod } N)$ và gởi kết quả này cho A.
- B4: Sau khi nhận dữ liệu đã bị mã hóa C, A sử dụng khóa bí mật d và tiến hành giải mã theo công thức sau:
  - $M=C^d(\text{mod } N)=(M^e)^d \text{mod } N=M^{de} \text{mod } N$.

2.5. Thuật toán MD5

Đầu vào của hàm băm là những khối 512-bit, được chia cho 16 khối con 32-bit. Đầu ra của thuật toán là một thiết lập của 4 khối 32-bit để tạo thành một hàm băm 128-bit duy nhất.

Đầu tiên, chia bức điện thành các khối 512-bit, mỗi khối 512-bit lại được chia ra 16 khối 32-bit đi vào bốn vòng lặp của MD5. Giả sử đặt a, b, c và d thay cho A, B, C và D đối với khối 512-bit đầu tiên của bức điện. Bốn vòng lặp trong MD5 đều có cấu trúc giống nhau. Mỗi vòng thực hiện 16 lần biến đổi: thực hiện với một hàm phi tuyến của 3 trong 4 giá trị a, b, c và d; sau đó nó cộng kết quả đến giá trị thứ 4, tiếp đó cộng với một trong 16 khối con 32-bit Mj và một hằng số ti. Sau đó, nó dịch trái một lượng bit thay đổi và cộng kết quả vào 1 trong 4 giá trị a, b, c hay d. Kết quả cuối cùng là một giá trị mới được thay thế 1 trong 4 giá trị a, b, c hay d.

2.6. Thuật toán SHA

Cũng giống với MD5, bức điện được cộng thêm một bit 1 và các bit 0 ở cuối bức điện để bức điện có thể chia hết cho 512 bit. SHA sử dụng 5 thanh ghi dịch A, B, C, D, E.

Bức điện được chia ra thành nhiều khối 512-bit. Ta cũng đặt là a, b, c, d và e thay cho A, B, C, D và E đối với khối 512-bit đầu tiên của bức điện. SHA có bốn vòng lặp chính với mỗi vòng thực hiện 20

lần biến đổi: bao gồm thực hiện với một hàm phi tuyến của 3 trong 5 giá trị a, b, c, d và e sau đó cũng được cộng và dịch như trong MD5.

2.7. Kết luận

CHƯƠNG 3
ỨNG DỤNG MÃ HÓA DÒNG DỮ LIỆU VỚI JAVA

3.1. Lý thuyết mã hóa trong Java

3.1.1. JCA và JCE

JCA (Java Cryptography Architecture). JCA quy định các mẫu thiết kế cụ thể và kiến trúc mở rộng xác định các khái niệm và các thuật toán mã hóa. JCA thiết kế để phân loại các khái niệm mã hóa từ hiện thực. Các khái niệm này được gói gọn bởi lớp trong các gói java.security và javax.crypto.

JCE (Java Cryptography Extension): các lớp mã hóa được phân chia thành 2 nhóm. Nhóm đầu tiên bao gồm gói java.security. Nhóm thứ 2 là JCE. JCE là phần mở rộng của JCA. Nó bao gồm nhà cung cấp mã hóa khác gọi là SUNJCE.

JCA và JCE cung cấp một tập các lớp và giao diện mã hóa. Trong JCA và JCE, bộ sưu tập của các lớp được gọi là providers. JCA và JCE có một số cơ chế đơn giản cho phép mọi người thêm providers và lựa chọn providers cụ thể.

3.1.2. Key Management

Quản lý khóa là thách thức lớn nhất đối với những người muốn phát triển ứng dụng mã hóa. Phần này trình bày các khái niệm quản lý khóa đại diện bởi các lớp và giao diện.

3.1.2.1. Key

3.1.2.2. Key Genertors

3.1.3. Key Translators

3.1.3.1. SecretKeySpec

3.1.3.2. SecretKeyFactory

3.1.4. Cipher

3.1.4.1. Hình thành Cipher

3.1.4.2. Khởi tạo Cipher

3.1.4.3. Mã hóa và giải mã dữ liệu với Cipher

3.1.5. Các bước mã hóa dữ liệu

3.1.5.1. Mã hóa đối xứng

3.1.5.2. Mã hóa bất đối xứng

3.1.5.3. Mã hóa một chiều

3.2. Thiết kế thuật toán cho chương trình

Bài toán: Với các kỹ thuật mã hóa của SQL Server được nêu trên, tôi nhận thấy rằng hai kỹ thuật mã hóa cột và mã hóa một chiều chỉ mã hóa cột không mã hóa được bản ghi hay hàng trong một bảng. Nếu muốn mã hóa một hay nhiều bản ghi trong một bảng của CSDL thì SQL Server 2008 không làm được. Khắc phục điểm yếu này, sau đây là chương trình mã hóa từng dòng dữ liệu hay còn gọi là mã hóa từng bản ghi trong bảng. Chương trình phân cấp chức quyền cho nhân viên dựa trên chức vụ và từng phòng ban khác nhau. Mỗi trưởng phòng có quyền mã hóa hay giải mã dữ liệu của một hay tất cả nhân viên trong phòng. Và tương tự mỗi nhân viên chỉ được quyền mã hóa hay giải mã dữ liệu của riêng mình. Người có chức vụ lớn

nhất Hiệu Trưởng hoặc admin mới được quyền mã hóa hay giải mã dữ liệu của tất cả nhân viên trong trường.

3.2.1. Thuật toán đăng nhập

3.2.2. Thuật toán phân quyền quản lý toàn trường

3.2.3. Thuật toán phân quyền quản lý phòng

3.2.4. Thuật toán phân quyền nhân viên

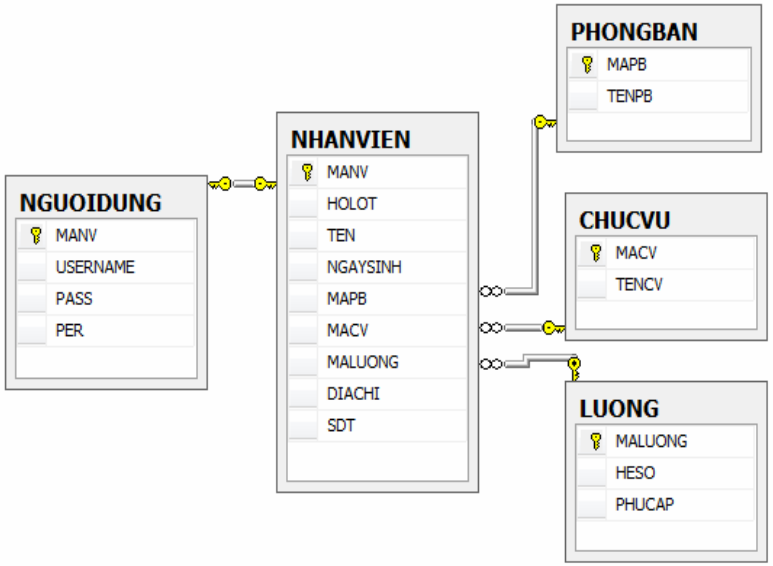
3.2.5. Thuật toán mã hóa dữ liệu một trường

3.2.6. Thuật toán giải mã dữ liệu một trường

3.2.7. Thuật toán mã hóa dữ liệu một bản ghi

3.2.8. Thuật toán giải mã dữ liệu đã được mã hóa của bản ghi

3.3. Thiết kế cơ sở dữ liệu



3.4. Kết luận

KẾT LUẬN

Việc mã hóa dữ liệu để đảm bảo an toàn thông tin ngày càng được sử dụng rộng rãi. Ngay trong hệ quản trị CSDL SQL Server 2008 cũng đã tích hợp công cụ này góp phần gia tăng mức độ an toàn của dữ liệu được lưu trữ bên trong. Tuy nhiên muốn phát huy được ưu điểm của các kỹ thuật mã hóa này, phải sử dụng kết hợp các kỹ thuật mã hóa đó với mô hình tầng mã hóa được nêu trên.

SQL Server 2008 đã sử dụng các thuật toán mã hóa: DES, Triple DES, AES, RSA, MD5, SHA … để mã hóa dữ liệu. Mỗi thuật toán có những ưu nhược điểm riêng. Trong luận văn, tôi đã mô tả khá chi tiết nội dung từng thuật toán. Hiểu được mục đích và quá trình tạo nên dữ liệu mã hóa của các thuật toán này sẽ giúp ích cho việc lựa chọn thuật toán phù hợp với chương trình.

Trong thời gian qua, tôi đã nỗ lực tìm hiểu, phân tích, nghiên cứu để thực hiện đề tài luận văn "*Nghiên cứu giải pháp bảo mật cơ sở dữ liệu SQL Server bằng phương pháp mã hóa*", với sự nhiệt tình giúp đỡ, chỉ bảo của các thầy giáo và bạn bè đồng nghiệp. Tôi xin tóm tắt những kết quả đạt được, những hạn chế và đề xuất một số định hướng phát triển cho đề tài như sau:

1. Những kết quả đạt được

Luận văn đã tổng hợp và giới thiệu các kỹ thuật mã hóa của hệ quản trị CSDL SQL Server 2008 cũng như định hướng phát triển ứng dụng sử dụng kỹ thuật mã hóa thông qua mô hình tầng mã hóa.

Công tác nghiên cứu đề tài đã cung cấp cho tôi rất nhiều kiến thức quí giá về cách thức mã hóa dữ liệu.

Nhìn chung đề tài đã đạt được các mục tiêu như đề ra ban đầu.

2. Những hạn chế

Hiện nay các thuật toán mã hóa rất đa dạng nhưng luận văn vẫn chưa tổng hợp, giới thiệu được.

Việc ứng dụng các kiến thức nghiên cứu được vào hệ thống quản lý dữ liệu tiến hành chưa tốt.

3. Hướng phát triển

Với những mặt còn tồn tại, định hướng sẽ cố gắng khắc phục trong thời gian đến. Tiếp tục hoàn thiện phần nội dung nghiên cứu. Tập trung vận dụng lý thuyết nghiên cứu vào trong thực tế.

Tiếp tục nghiên cứu chuyên sâu vào các thuật toán mã hóa và công cụ mã hóa của ngôn ngữ lập trình Java, nhất là các kiến thức phục vụ cho yêu cầu phát triển hệ thống thông tin quản lý.

Nghiên cứu các kỹ thuật mã hóa dữ liệu của hệ quản trị CSDL Oracle. Từng bước so sánh, bổ sung kiến thức mã hóa dữ liệu cho bản thân.