

**CTT201**  
**An toàn và Bảo mật Dữ liệu trong HTTP**  
 Chương 3: Mã hóa Cơ sở dữ liệu

TS. Phạm Thị Bạch Huệ  
 Ths. Lương Vĩ Minh – Ths. Hoàng Anh Tú

Khoa Công nghệ thông tin – Đại học Khoa học tự nhiên

@ 2014 – University of Science – Vietnam National University HCMC

1

Nội dung

1. Giới thiệu
2. Các mức độ mã hóa
3. Nhận xét về giải pháp mã hóa
4. Các vấn đề liên quan đến giải pháp mã hóa
5. Mô hình lưu trữ dữ liệu mã hóa
6. Hiện thực trên 1 DBMS

Copyright © University of Science – HCM-VNU 2014

2

Chương 3: Mã hóa Cơ sở dữ liệu

**GIỚI THIỆU**

Copyright © University of Science – HCM-VNU 2014

3

Giới thiệu

The diagram shows four interconnected components of database security:

- Identification:** Định danh người dùng (User Identification) - Represented by a smartphone icon.
- Authentication:** Xác thực người dùng (User Authentication) - Represented by a login screen icon.
- Access Control:** Điều khiển truy cập (Access Control) - Represented by a server icon.
- Auditing:** Giám sát hoạt động (Activity Monitoring) - Represented by a magnifying glass over a document icon.

Data Encryption  
Mã hóa dữ liệu

Copyright © University of Science – HCM-VNU 2014

4

**Mã hóa dữ liệu**

- Mã hóa là phương pháp che giấu dữ liệu, biến dữ liệu sang dạng mã không có ý nghĩa đối với kẻ tấn công
- Đây là rào cản cuối cùng, khi mà kẻ tấn công vượt qua được các cơ chế bảo vệ dữ liệu khác.

Copyright © University of Science – HCM-VNU 2014

5

**Mã hóa dữ liệu**

*Sniffer*

- Mã hóa dữ liệu có thể được thực hiện ở cả hai thời điểm:

Copyright © University of Science – HCM-VNU 2014

6

**Mã hóa dữ liệu trên đường truyền**

- SSL (Secure Socket Layer) – Netscape
- PCT (Private Communication Technology) – Microsoft
- TLS (Transport Layer Security) - IETF (Internet Engineering Task Force)

Việc bảo vệ thông tin trên đường truyền rất cần thiết.  
Tuy nhiên, hầu hết các cuộc tấn công này cập thông tin xuất hiện tại điểm cuối cùng dữ liệu khi lưu trữ.

Copyright © University of Science – HCM-VNU 2014

7

**Các phương pháp mã hóa**

- Các phương pháp mã hóa hiện có:
  - Phương pháp Mã hóa đối xứng
  - Phương pháp Mã hóa bất đối xứng
  - Phương pháp Mã hóa lai
  - Phương pháp Hàm băm mật mã

Copyright © University of Science – HCM-VNU 2014

8

**Phương pháp Mã hóa đối xứng**

Source: <http://msdn.microsoft.com/en-us/library/ff650720.aspx>

- **Symmetric Cryptography**
- Chỉ sử dụng 1 mã khóa (**Shared Secret Key**) để mã và giải mã dữ liệu
- Thuật toán đơn giản, độ dài khóa ngắn → tốc độ xử lý nhanh, phù hợp cho bảo mật lượng lớn dữ liệu
- Khó khăn trong việc phân phối khóa → Cần có hệ thống quản lý khóa (phát sinh khóa, phân phối khóa, sao lưu khóa, tái phát sinh khóa và quản lý vòng đời của khóa)
- Không cung cấp khả năng chống sự thoái thác trách nhiệm; không chứng minh được ai là người thật sự đã gửi dữ liệu

Copyright © University of Science – HCM-VNU 2014

9

**Phương pháp Mã hóa đối xứng**

- **Một số thuật toán phổ biến:**
  - Block Cipher:
    - Data Encryption Standard (DES).
    - Triple Data Encryption Standard (3DES).
    - Advanced Encryption Standard (AES - Rijndael).
    - BlowFish, TwoFish, Serpent
  - Stream Cipher:
    - RC4

Copyright © University of Science – HCM-VNU 2014

10

**Phương pháp Mã hóa bất đối xứng**

Source: <http://msdn.microsoft.com/en-us/library/ff650720.aspx>

- **Asymmetric Cryptography (Public-key Cryptography)**
- Sử dụng 1 cặp khóa: **Public key** để mã và **Private key** giải mã dữ liệu.
- Giải quyết được vấn đề Trao đổi khóa
- Thuật toán phức tạp → tốc độ xử lý chậm, nhưng an toàn
- Chỉ phù hợp mã hóa dữ liệu ít

Copyright © University of Science – HCM-VNU 2014

11

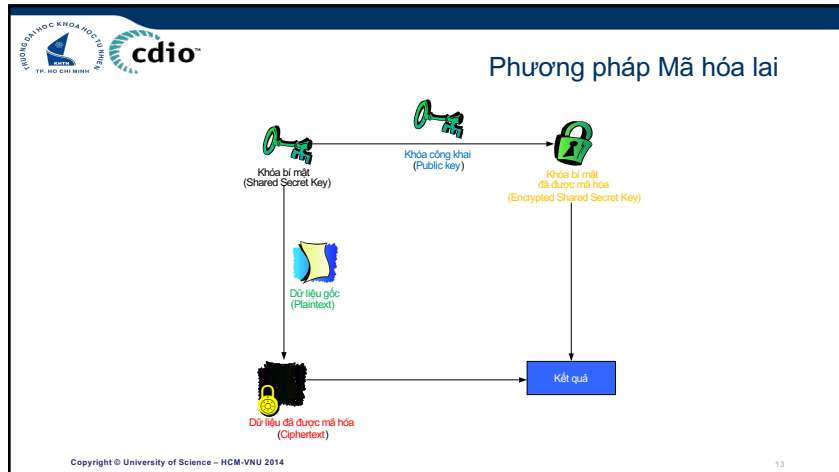
**Phương pháp Mã hóa bất đối xứng**

- **Một số thuật toán phổ biến:**
  - Diffie-Hellman key exchange
  - Rivest-Shamir-Adleman (RSA)
  - Digital Signature Algorithms (DSA)
  - ElGamal
  - Elliptic Curve Cryptography (ECC)
  - Paillier cryptosystem

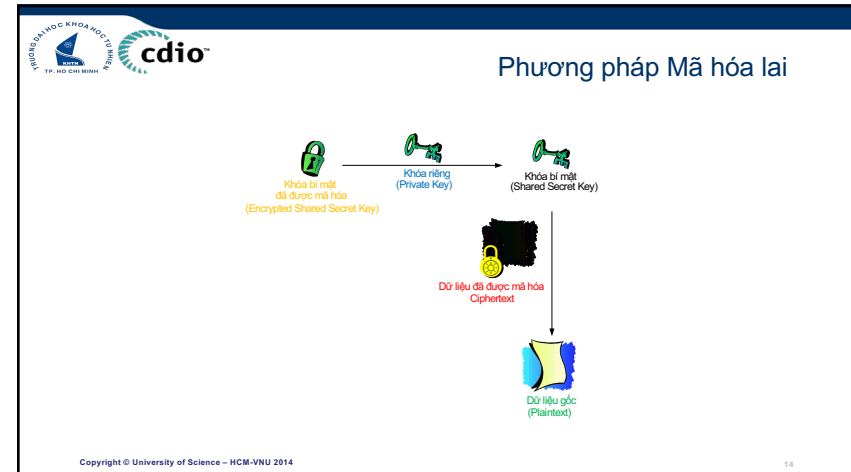
Public key exchange

Copyright © University of Science – HCM-VNU 2014

12



13



14

**Phương pháp Mã hóa lai**

- Kết hợp Phương pháp mã hóa đối xứng và Phương pháp mã hóa bất đối xứng
- Tận dụng được:
  - Ưu điểm về **tốc độ** của phương pháp mã hóa đối xứng
  - Tính **an toàn** của phương pháp mã hóa bất đối xứng

Copyright © University of Science – HCM-VNU 2014

15

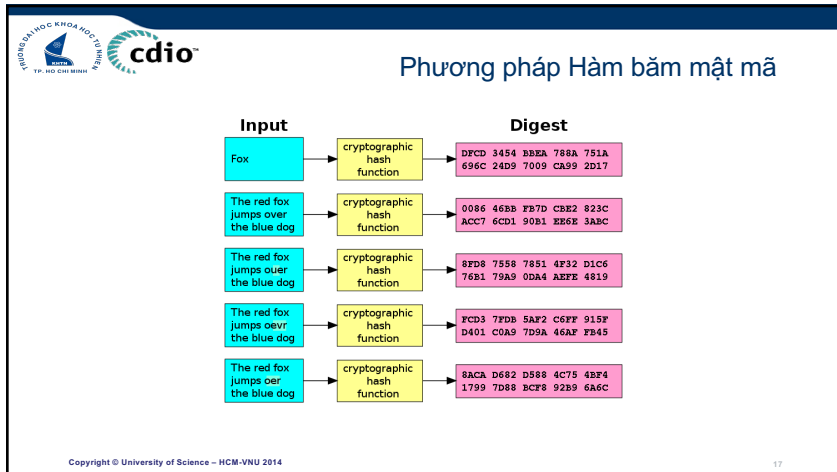
**Phương pháp Hàm băm mật mã**

- **Cryptographic Hash Function**
  - Hàm băm mật mã không khôi phục được dữ liệu sau khi băm
  - Kết quả của hàm băm cho ra một giá trị có chiều dài cố định (theo từng thuật giải)
  - Khả năng “đụng độ” của hàm băm mật mã với dữ liệu đầu vào là rất hiếm
- Hàm băm mật mã được sử dụng để xác thực dữ liệu
- Một số hàm băm mật mã phổ biến:
  - MD5
  - SHA-0, **SHA-1**, SHA-224, SHA-256, SHA-512

The diagram shows a laptop icon with an arrow pointing down to a box labeled 'Hash', which then points down to the hexadecimal string '952d2c56...'. Above the 'Hash' box is the label 'Hạt nhân'.

Copyright © University of Science – HCM-VNU 2014

16



17

**Chương 3: Mã hóa Cơ sở dữ liệu**

## CÁC MỨC ĐỘ MÃ HÓA

Copyright © University of Science – HCM-VNU 2014

18

**Các mức độ mã hóa dữ liệu**

- Mã hóa mức ứng dụng (Application Level)
- Mã hóa mức lưu trữ (Storage Level)
- Mã hóa mức CSDL (Database Level)

Copyright © University of Science – HCM-VNU 2014


19

**Mã hóa mức ứng dụng (Application Level)**

- Việc **mã hóa / giải mã** dữ liệu được thực hiện ngay trong mã lệnh chương trình ở mức ứng dụng (application), liên quan đến các thao tác xử lý trên dữ liệu cần bảo vệ, chọn lựa ĐVDL mã hóa
- Phù hợp đối với các ứng dụng thực hiện các công việc xử lý, cấp quyền, thao tác, ... trên dữ liệu bí mật ở mức ứng dụng
- Tận dụng được thư viện hỗ trợ mã hóa: **JCE** (Java-based application) hoặc **MS-CAPI** (Microsoft-based application)

Copyright © University of Science – HCM-VNU 2014

20




### Mã hóa mức ứng dụng (Application Level)

- **Bảo vệ liệu khỏi các nguy cơ:**
  - Thiết bị lưu trữ bị đánh cắp
  - Chống được tấn công dữ liệu ở mức lưu trữ
  - Truy cập dữ liệu bí mật từ người quản trị dữ liệu.
- **Hạn chế:**
  - Cơ sở dữ liệu không có khả năng dùng cho các ứng dụng khác
  - Phải sử dụng mô hình mã / giải mã dữ liệu tương thích hoặc thay đổi mã chương trình khi chia sẻ dữ liệu

Copyright © University of Science – HCM-VNU 2014

21




### Mã hóa mức lưu trữ (Storage Level)

- Mã hóa / giải mã **tập tin** lưu trữ toàn bộ dữ liệu, CSDL với 1 mã khóa duy nhất
- Được thực hiện ở cấp Hệ điều hành
- Phù hợp cho việc bảo vệ dữ liệu sao lưu, dữ liệu off-line
- Bảo vệ được dữ liệu khi thiết bị lưu trữ bị đánh cắp hoặc bị tấn công ở mức lưu trữ
- Thực tế đã có nhiều nhà cung cấp xây dựng các chức năng phần mềm đáp ứng nhu cầu này

Copyright © University of Science – HCM-VNU 2014

22




### Mã hóa mức lưu trữ (Storage Level)

- **Hạn chế:**
  - Không lựa chọn được dữ liệu cần bảo vệ
  - Không thể phân quyền trên đơn vị dữ liệu nhỏ hơn (như bảng, dòng, cột)
  - Không bảo vệ được dữ liệu khỏi những tấn công mức ứng dụng hoặc mức cơ sở dữ liệu
  - Không ngăn chặn được Quản trị hệ thống truy cập đến tập tin
  - Không ngăn chặn được việc truy cập đến tập tin dữ liệu đã được mã hóa khi mất quyền quản trị hệ thống của HĐH
  - Gây ra vấn đề về hiệu năng khi đọc và ghi dữ liệu từ cơ sở dữ liệu

Copyright © University of Science – HCM-VNU 2014

23




### Mã hóa mức Cơ sở dữ liệu (Database Level)

- Việc mã hóa / giải mã dữ liệu được thực hiện ở cấp HQT CSDL
- Được đảm nhận thông qua việc dùng thủ tục hoặc trigger
- Đơn vị dữ liệu có thể được chọn để mã hóa là: từng giá trị tại từng thuộc tính, từng dòng, từng cột, từng bảng, toàn bộ cơ sở dữ liệu ....
- Dễ dàng chia sẻ dữ liệu mã hóa giữa các chương trình ứng dụng khác nhau
- Chống được các kiểu tấn công như: đánh cắp thiết bị lưu trữ, tấn công mức cơ sở dữ liệu (ví dụ SQL injection), người quản trị truy cập dữ liệu bất hợp pháp

Copyright © University of Science – HCM-VNU 2014

24


 **Mã hóa mức Cơ sở dữ liệu (Database Level)**

Các cấp độ mã hóa cơ sở dữ liệu:

- Attribute value (cấp độ giá trị thuộc tính):** tất cả thuộc tính của bộ dữ liệu đều được mã hóa và từng giá trị thuộc tính của bộ dữ liệu được mã hóa riêng biệt
- Record/Row level (cấp độ bộ/dòng):** từng dòng trong bảng được mã hóa riêng lẻ. Tuy nhiên, mã hóa dữ liệu cấp độ dòng đôi khi mã hóa luôn những thuộc tính không cần thiết phải che giấu.
- Column/Attribute level (cấp độ cột/thuộc tính):** chỉ những thuộc tính nhạy cảm mới được mã hóa.
- Page/Block level (cấp độ trang/khối):** toàn bộ các dòng dữ liệu trong một trang được mã hóa một lần. Số lượng bộ dữ liệu trong trang phụ thuộc vào kích thước trang và kích thước bộ dữ liệu.

Copyright © University of Science – HCM-VNU 2014

25


 **Mã hóa mức Cơ sở dữ liệu (Database Level)**

**Hạn chế:**

- Khi **thay đổi kiểu hay kích thước** của trường dữ liệu liên quan → Tiến hành thay đổi thủ tục/ trigger mã hóa/ giải mã dữ liệu cho phù hợp
- Làm chậm hệ thống** đáng kể khi không có kinh nghiệm (do tốn nhiều thời gian cho các lần khởi động thuật toán mã hóa → Chỉ nên mã hóa dữ liệu nhạy cảm)
- Không an toàn** với tấn công ở mức Ứng dụng

Copyright © University of Science – HCM-VNU 2014


26

 **Chương 3: Mã hóa Cơ sở dữ liệu**

**NHẬN XÉT VỀ GIẢI PHÁP MÃ HÓA CSDL**

Copyright © University of Science – HCM-VNU 2014

27

 **Ví dụ - Mã hóa CSDL**


Order	First Name	Last Name	Address	ZIP	Email	Card
4667	Alma	MASKED	MASKED	MASKED	*****@****.m	XXXX-XXXX-XXXX-8094
3427	Mary	MASKED	MASKED	MASKED	*****@****.m	XXXX-XXXX-XXXX-9980
3672	Diana	MASKED	MASKED	MASKED	*****@****.m	XXXX-XXXX-XXXX-5710
5356	Mona	MASKED	MASKED	MASKED	*****@****.m	XXXX-XXXX-XXXX-5426
1289	Reggy	MASKED	MASKED	MASKED	*****@****.m	XXXX-XXXX-XXXX-5933
6768	Patric	MASKED	MASKED	MASKED	*****@****.m	XXXX-XXXX-XXXX-5861
3237	Hugh	MASKED	MASKED	MASKED	*****@****.m	XXXX-XXXX-XXXX-5262
9034	Allen	MASKED	MASKED			
1223	Kathy	MASKED	MASKED			
1233	John	MASKED	MASKED			
2314	Kelley	MASKED	MASKED			
4930	Lou	MASKED	MASKED			
5803	Malinda	MASKED	MASKED			
4687	Carmen	MASKED	MASKED			
4327	Karen	MASKED	MASKED			
2356	Charles	MASKED	MASKED			
7932	Clarice	MASKED	MASKED			
1332	Bella	MASKED	MASKED			
1845	Dale	MASKED	MASKED			
1234	Justin	MASKED	MASKED			

Results | Messages

	Customer_id	Customer_Name	Credit_card_number_encrypt
1	25665	msgsltp4	0x0043D06201CD98408F1283AE54FDC21010000008F2462D...
2	74112	MSSQLTps2	0x0043D06201CD98408F1283AE54FDC2101000000C3D8929...
3	74113	MSSQLTps3	0x0043D06201CD98408F1283AE54FDC2101000000FDC3D...
4	74114	MSSQLTps4	0x0043D06201CD98408F1283AE54FDC2101000000EC1891...
5	74115	MSSQLTps5	0x0043D06201CD98408F1283AE54FDC2101000000DDCA5B...

Copyright © University of Science – HCM-VNU 2014

28

 **cdio™**


## Nhận xét giải pháp mã hóa CSDL

**Ưu điểm:**

- Mã hóa cơ sở dữ liệu có thể **che giấu dữ liệu** khỏi những kẻ xâm nhập, thậm chí cả DBA nếu họ không được phép truy cập dữ liệu
- Mã hóa cơ sở dữ liệu là phương pháp bảo vệ dữ liệu rất hiệu quả đối với những **tấn công mức lưu trữ**. Những kẻ tấn công có được dữ liệu nhưng không thể hiểu được dữ liệu

Copyright © University of Science – HCM-VNU 2014

29

 **cdio™**


## Nhận xét giải pháp mã hóa CSDL

**Khuyết điểm:**

- Mã hóa CSDL **làm tăng lượng xử lý** khi truy cập dữ liệu, tăng dung lượng lưu trữ dữ liệu
- Mã hóa CSDL làm HQT CSDL **không thể thực thi** các phương thức truy cập dữ liệu cơ bản
- Mã hóa **cần có chính sách quản lý khóa** thích hợp
- Mã khóa là thành phần quan trọng nhất
  - Mất khóa → Bị lộ dữ liệu
  - Mất khóa → Dữ liệu không được giải mã

Copyright © University of Science – HCM-VNU 2014

30

 **cdio™**

## Nhận xét giải pháp mã hóa CSDL


Thuật toán	100 bytes x 100.000 lần mã hóa	120 bytes x 83.333 lần mã hóa	16 KB x 625 thao tác mã hóa
AES (16B)	365 ms	334 ms	194 ms
DES (8B)	327 ms	354 ms	229 ms
Blowfish (8B)	5280 ms	4409 ms	170 ms

Linux, 2.8 Ghz PIV, 1Gbyte RAM + thư viện OpenSSL

Source: Bodo Jans, Sheng-Ming Chen, Einar Mjølhus, Gene Tsudik2, and Yonghua Wu, A Framework for Efficient Storage Security in SIGMOD, 2004, LNCS 2992, pp. 147–164, 2004

Copyright © University of Science – HCM-VNU 2014

31

 **cdio™**

## Kết luận


Mặc dù có nhiều lý do để phải dùng **giải pháp mã hóa**, nhưng mã hóa không phải là giải pháp hoàn toàn tốt.

- Mã hóa **không thể đảm nhận công việc điều khiển truy cập**. Mã hóa chỉ nhằm giấu nội dung dữ liệu.
- Việc mã hóa **không được làm ảnh hưởng** đến kết quả của việc điều khiển truy cập.  
*Ví dụ: A có quyền SELECT trên bảng NHANVIEN thì khi mã hóa xong A không bị ngăn cản dữ liệu mà A được phép xem*
- Điều không mong muốn: DBA có thể truy cập đến toàn bộ dữ liệu → Mã hóa CSDL. **Mã hóa toàn bộ CSDL không phải là giải pháp tốt.**

Copyright © University of Science – HCM-VNU 2014

32






Chương 3: Mã hóa Cơ sở dữ liệu

## CÁC VẤN ĐỀ LIÊN QUAN ĐẾN GIẢI PHÁP MÃ HÓA CSDL

Copyright © University of Science – HCM-VNU 2014

33



## Vấn đề mã hóa trên khóa chính, khóa ngoại và ràng buộc toàn vẹn

- Nếu dữ liệu **khóa chính** chứa dữ liệu nhạy cảm → cần mã hóa
- Giải pháp mã hóa:** Mã hóa dữ liệu ở tất cả các dòng tại các cột tham gia làm khóa chính của bảng dữ liệu
  - Sử dụng **cùng** 1 mã khóa + **cùng** 1 vector khởi tạo (IV)
  - Sử dụng mỗi dòng một mã khóa **khác nhau**
  - Sử dụng **cùng** 1 mã khóa + **khác** vector khởi tạo (IV)
- Vấn đề nào cần lưu ý khi mã hóa dữ liệu trên Khóa chính ?

Copyright © University of Science – HCM-VNU 2014

34




## Vấn đề mã hóa trên khóa chính, khóa ngoại và ràng buộc toàn vẹn

- Vi phạm **ràng buộc khóa chính** (khi mã khóa khác nhau hoặc cùng mã khóa nhưng IV khác nhau)
  - Hủy ràng buộc khóa chính + tự cài đặt thủ tục kiểm tra
- Vi phạm **ràng buộc khóa ngoại**
  - Mã hóa cùng mã khóa và IV với giá trị tham chiếu ở khóa chính đã được mã hóa
- Vi phạm **ràng buộc toàn vẹn khác** trên khóa chính, khóa ngoại (nếu hệ thống có sẵn dữ liệu)
  - Hủy tất cả ràng buộc, tiến hành mã hóa và tạo lại ràng buộc
- Không thực hiện được** ràng buộc toàn vẹn hiện có (do đặc tính của dữ liệu)
  - Tự cài đặt lại bằng hàm / thủ tục / trigger

Copyright © University of Science – HCM-VNU 2014

35




## Vấn đề chỉ mục trên dữ liệu mã hóa

- Mục tiêu của lập chỉ mục trong CSDL → Tăng tốc độ tìm kiếm
- Nếu cần mã hóa trên dữ liệu nhạy cảm có chỉ mục, cần giải quyết 2 trường hợp sau:
  - Lập chỉ mục cho các cột dữ liệu **đã được** mã hóa
  - Lập chỉ mục **trước khi** mã hóa dữ liệu
- Các HQT CSDL **không khuyến khích** lập chỉ mục trên dữ liệu mã hóa vì trong nhiều trường hợp, tìm kiếm trên cột dữ liệu đã được mã hóa sẽ yêu cầu HQT phải duyệt qua toàn bộ bảng để xác định phần tử cần tìm. Lúc đó, vai trò của chỉ mục trở nên vô nghĩa

Copyright © University of Science – HCM-VNU 2014

36



## Vấn đề chỉ mục trên dữ liệu mã hóa

### Lập chỉ mục cho các cột dữ liệu đã được mã hóa


- Giải pháp này phù hợp với việc thực hiện tìm kiếm với phép so sánh bằng.
- Dữ liệu đã bị mã hóa thì thứ tự chỉ mục không còn liên quan về mặt ngữ nghĩa → có khả năng làm hạn chế hoặc chậm quá trình tìm kiếm

### Lập chỉ mục trước khi mã hóa dữ liệu

- Một chỉ mục trung gian sẽ được tạo dựa trên plaintext, sau đó sẽ mã hóa dữ liệu.
- Sau đó, chỉ mục sẽ được cập nhật lại và nó tham chiếu đến giá trị đã được mã hóa của dữ liệu. Lúc này bản thân của chỉ mục lại không có thứ tự nhưng dữ liệu sau khi giải mã thì có thứ tự.
- Với cách làm này thì khi tìm kiếm ta vẫn phải duyệt tuần tự qua các phần tử của chỉ mục.

Copyright © University of Science – HCM-VNU 2014

37




## Vấn đề tìm kiếm trên dữ liệu mã hóa

- Vấn đề **tìm kiếm chính xác** → sử dụng **cùng 1 khóa và IV** khi mã hóa tất cả giá trị trên cột dữ liệu cần tìm kiếm này
- Vấn đề **tìm kiếm gần đúng** (like, >, <, ...) → Thông thường phải duyệt toàn bộ bảng nếu như không có cơ chế hỗ trợ tìm kiếm nhanh dùng chỉ mục
- Giải pháp chung:** Áp dụng **hàm băm mật mã** trên một phần của dữ liệu nhạy cảm và lưu cùng dòng nhưng trên một cột khác

Copyright © University of Science – HCM-VNU 2014

38




## Vấn đề tìm kiếm trên dữ liệu mã hóa

**Ví dụ:**

- Có bảng dữ liệu lưu lại thông tin của khách hàng, trong đó có trường **địa chỉ Email** đã được mã hóa vì EMAIL là thông tin nhạy cảm
- Vậy để có thể tìm kiếm trên trường Email, sẽ tạo thêm 1 cột nữa lưu lại **giá trị băm** của 4 ký tự đầu của địa chỉ email đó
- Cách giải quyết này cũng có thể dùng cho việc tìm kiếm chính xác, với điều kiện là **biết trước điều kiện tìm kiếm thường được thành lập trên những tiêu chí nào** (4 ký tự đầu hay 5 ký tự cuối, ...)
- Với cách tiếp cận này, **luôn phải tạo ra thêm 1 trường mới** để phục vụ cho mỗi một nhu cầu tìm kiếm

Copyright © University of Science – HCM-VNU 2014

39




## Vấn đề quản lý khóa

- Mã khóa sử dụng trong quá trình mã hóa dữ liệu trên đường truyền dữ liệu → Không cần phải lưu trữ
- Do dữ liệu trong CSDL có thời gian sống dài và cố định hơn → Mã khóa sử dụng trong quá trình mã hóa CSDL cần phải:
  - Tạo ra và truyền khóa cho người dùng được phép
  - Lưu trữ các khóa cho lần truy cập sau
- Việc quản lý khóa mã hóa phải đảm bảo:
  - Những người dùng **không** có quyền thì **không** được “thấy” dữ liệu nhạy cảm đang được bảo vệ
  - Dữ liệu sẽ được mã hóa cho từng người nhận **khác nhau**, với các quyền hạn **khác nhau**
  - Các mã khóa phải được đảm bảo **an toàn**

Copyright © University of Science – HCM-VNU 2014

40




## Quản lý khóa

1. Lưu khóa trong cơ sở dữ liệu
2. Quản lý khóa bởi ứng dụng
3. Tính toán ra khóa
4. Quản lý khóa dùng phương pháp mã hóa lại

Copyright © University of Science – HCM-VNU 2014

41




## 1. Lưu khóa trong cơ sở dữ liệu

- Mã khóa được lưu trong CSDL nhằm thuận tiện cho việc **sao lưu** và **phục hồi** dữ liệu, vì khóa được bảo trì cùng với dữ liệu

Copyright © University of Science – HCM-VNU 2014

42




## 1. Lưu khóa trong cơ sở dữ liệu

**Một số điều lưu ý khi lưu khóa trong CSDL:**

1. Bảng lưu trữ khóa phải **được che giấu và bảo vệ** chặt chẽ
2. Sử dụng các cơ chế điều khiển truy cập để **hạn chế truy cập** vào bảng lưu trữ khóa
3. Tên bảng và tên các thuộc tính của bảng lưu trữ khóa **không nên đặt rõ ràng**
4. **Không nên tạo ràng buộc** giữa bảng lưu trữ khóa và các bảng khác để tránh sự suy diễn
5. Dữ liệu lưu trữ trong bảng Lưu khóa **cũng phải được mã hóa** với các hàm mã và giải mã tự xây dựng
6. **Duy trì việc giám sát truy cập** vào bảng này và định kỳ kiểm tra
7. Vấn phải **lưu ý về rủi ro đối với việc can thiệp và thay đổi** khóa của DBA

Copyright © University of Science – HCM-VNU 2014

43



## 2. Quản lý khóa bởi ứng dụng

1. Lưu trữ khóa trong tập tin của Application Server
2. Tập tin lưu khóa phải được **mã hóa bằng một Master Key**
3. Đảm bảo khóa **chỉ được gửi đến các chương trình liên quan**
4. Đảm bảo DBA cũng không lấy được khóa
5. Luôn đảm bảo các khóa này **phải giải mã được dữ liệu** → cần có chiến lược để đảm bảo các khóa quản lý bởi ứng dụng được **lưu trữ** và **sao lưu** an toàn
6. Lưu ý **rủi ro mất khóa** khi ứng dụng gặp sự cố

Copyright © University of Science – HCM-VNU 2014

44

**3. Tính toán ra khóa**

- Một cách hiệu quả để quản lý khóa là **không lưu trữ khóa một cách thật sự**. Các khóa có thể được tính toán ra một cách gián tiếp bởi một hàm dựa trên một giải thuật bảo mật
- Các hàm, thủ tục trong cơ sở dữ liệu **nên được che giấu** để ngăn chặn những người dùng có quyền thực thi tất cả thủ tục biết được thuật toán bảo mật
- Sự an toàn của các khóa trong cơ sở dữ liệu liên quan mật thiết đến **sự an toàn của các thuật toán tạo khóa**. Nếu mã chương trình bị phân tích, thuật toán bị lộ, các khóa sẽ bị lộ

Copyright © University of Science – HCM-VNU 2014

45

**3. Tính toán ra khóa**

**Ví dụ:**

- Có thể **BIT\_XOR** khóa chính với **tên lược đồ** và **tên thực thể** của database server.
- Điều này có thể mã hóa dữ liệu và đảm bảo rằng nó được bảo vệ cho thể hiện và lược đồ cơ sở dữ liệu này, do đó, không cho phép việc xuất sang một cơ sở dữ liệu khác hay một lược đồ khác.

Copyright © University of Science – HCM-VNU 2014

46

**4. Quản lý khóa dùng phương pháp mã hóa lai**

- Sử dụng Mã hóa đối xứng để mã / giải mã dữ liệu ở một cột dữ liệu nhạy cảm bằng một **Mã khóa bí mật**
- Sử dụng **mã khóa khác nhau** để mã / giải mã dữ liệu ở các cột không có quan hệ với nhau
- Mỗi người dùng có 1 cặp khóa bất đối xứng (**Public/Private**). Private key được bảo vệ bằng **mật khẩu** của người dùng (được xem như là **passphrase**)
- Khi người dùng được cấp quyền truy cập dữ liệu mã hóa → **Mã khóa bí mật** được mã hóa bằng **Public key** của người dùng và được lưu trữ công khai trong CSDL (**Mã khóa bí mật bị mã**)
- Mã khóa bí mật bị mã** này chỉ được giải mã bằng **Private key** của người dùng đã được cấp quyền truy cập dữ liệu để có được **Mã khóa bí mật** để giải mã dữ liệu

Copyright © University of Science – HCM-VNU 2014

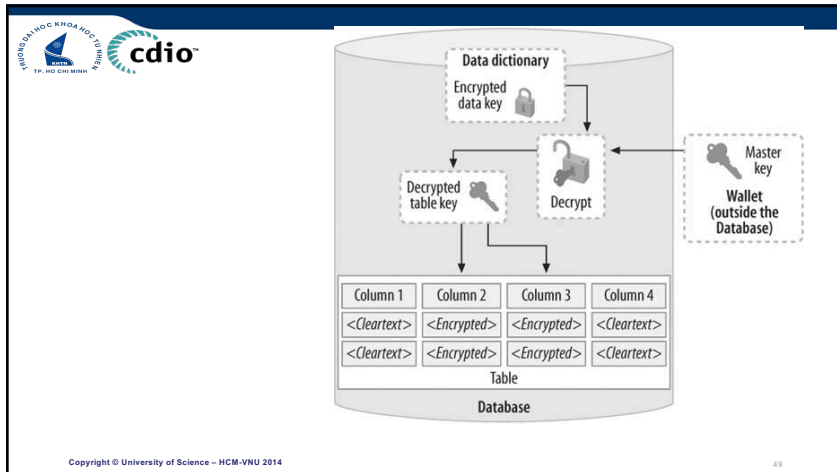
47

**Quản lý khóa**

Khi người chủ của private key quên passphrase thì coi như dữ liệu trở nên vô nghĩa, không có cách nào giải mã dữ liệu

Copyright © University of Science – HCM-VNU 2014

48



49

Chương 3: Mã hóa Cơ sở dữ liệu

## HIỆN THỰC TRÊN 1 DBMS

66

- Đọc tài liệu tham khảo.

67

### DBMS\_CRYPTO

- Chứa các hàm/ thủ tục để mã và giải mã.
- Có thể làm việc trên các kiểu dữ liệu phổ biến của Oracle, gồm RAW và LOB (hình ảnh, âm thanh).
- Hỗ trợ BLOB và CLOB, với tập ký tự khác nhau.
- Gồm các thuật toán:
  - Data Encryption Standard (DES), Triple DES (3DES, 2-key and 3-key)
  - Advanced Encryption Standard (AES)
  - MD5, MD4, and SHA-1 cryptographic hashes
  - MD5 and SHA-1 Message Authentication Code (MAC)
- Block Cipher modifier:
  - Padding options: có PKCS (Public Key Cryptographic Standard) #5
  - Four block cipher chaining modes: có Cipher Block Chaining (CBC).

68

DBMS_CRYPTO & DBMS_OBFUSCATION_TOOLKIT		
Package Feature	DBMS_CRYPTO	DBMS_OBFUSCATION_TOOLKIT
Cryptographic algorithms	DES, 3DES, AES, RC4, 3DES_2KEY	DES, 3DES
Padding forms	PKCSS, zeroes	none supported
Block cipher chaining modes	CBC, CFB, ECB, OFB	CBC
Cryptographic hash algorithms	MD5, SHA-1, MD4	MD5
Keyed hash (MAC) algorithms	HMAC_MD5, HMAC_SH1	none supported
Cryptographic pseudo-random number generator	RAW, NUMBER, BINARY_INTEGER	RAW, VARCHAR2
Database types	RAW, CLOB, BLOB	RAW, VARCHAR2

Copyright © University of Science – HCM-VNU 2014

69

Kiểu dữ liệu của DBMS_CRYPTO	
Type	Description
BLOB	A source or destination binary LOB
CLOB	A source or destination character LOB (excluding NCLOB)
PLS_INTEGER	Specifies a cryptographic algorithm type (used with BLOB, CLOB, and RAW datatypes)
RAW	A source or destination RAW buffer

Copyright © University of Science – HCM-VNU 2014

70

Các hàm băm của DBMS_CRYPTO	
Name	Description
DBMS_CRYPTO.MD4	Produces a 128-bit hash, or message digest of the input message
DBMS_CRYPTO.MD5	Also produces a 128-bit hash, but is more complex than MD4
DBMS_CRYPTO.SHA1	Secure Hash Algorithm (SHA). Produces a 160-bit hash.

Copyright © University of Science – HCM-VNU 2014

71


- Để tính giá trị băm có chiều dài cố định của 1 chuỗi có chiều dài bất kỳ
- Đây là những hàm băm một chiều
- Input là RAW hoặc LOB
- Chiều dài giá trị băm an toàn tối thiểu là 128 bit
- Dùng giá trị băm để kiểm tra dữ liệu có bị chỉnh sửa

DBMS_CRYPTO: Các hàm MAC	
Name	Description
HMAC_MD5	Same as MD5 hash function, except it requires a secret key to verify the hash value.
HMAC_SH1	Same as SHA hash function, except it requires a secret key to verify the hash value.

Copyright © University of Science – HCM-VNU 2014

72

- Cũng là những hàm băm một chiều, nhưng cần có khoá
- Dùng để xác thực file giữa 2 người dùng, hoặc bởi cùng người dùng phòng virus



## DBMS\_CRYPTO: Các hàm mã

Name	Description
ENCRYPT_DES	Data Encryption Standard. Block cipher. Uses key length of 56 bits.
ENCRYPT_3DES_2KEY	Data Encryption Standard. Block cipher. Operates on a block 3 times with 2 keys. Effective key length of 112 bits.
ENCRYPT_3DES	Data Encryption Standard. Block cipher. Operates on a block 3 times.
ENCRYPT_AES128	Advanced Encryption Standard. Block cipher. Uses 128-bit key size.
ENCRYPT_AES192	Advanced Encryption Standard. Block cipher. Uses 192-bit key size.
ENCRYPT_AES256	Advanced Encryption Standard. Block cipher. Uses 256-bit key size.
ENCRYPT_RC4	Stream cipher. Uses a secret, randomly generated key unique to each session.

Copyright © University of Science – HCM-VNU 2014

73



## DBMS\_CRYPTO: Các hàm mã

Name	Description
DES_CBC_PKCS5	ENCRYPT_DES + CHAIN_CBC + PAD_PKCS5
DES3_CBC_PKCS5	ENCRYPT_3DES + CHAIN_CBC + PAD_PKCS5

Copyright © University of Science – HCM-VNU 2014

74



## DBMS\_CRYPTO: Các hàm mã

Name	Description
CHAIN_ECB	Electronic Codebook. Encrypts each plaintext block independently.
CHAIN_CBC	Cipher Block Chaining. Plaintext is XORed with the previous ciphertext block before it is encrypted.
CHAIN_CFB	Cipher-Feedback. Enables encrypting units of data smaller than the block size.
CHAIN_OFB	Output-Feedback. Enables running a block cipher as a synchronous stream cipher. Similar to CFB, except that $n$ bits of the previous output block are moved into the right-most positions of the data queue waiting to be encrypted.

Copyright © University of Science – HCM-VNU 2014

75

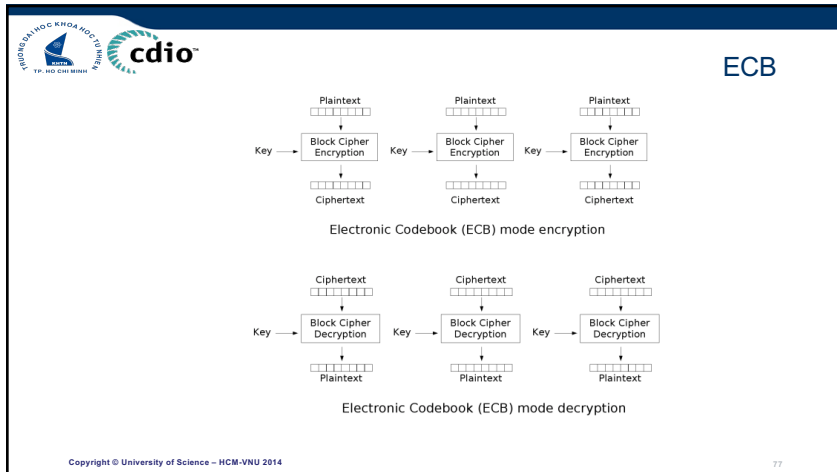


## DBMS\_CRYPTO: Các hàm mã

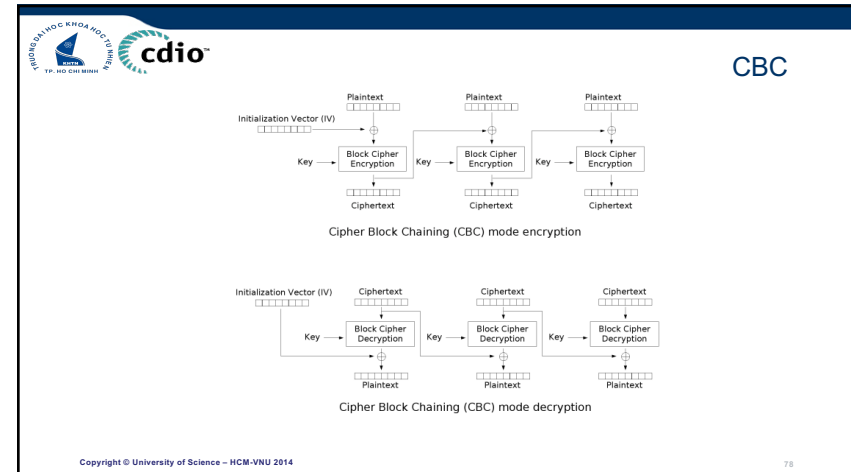
Name	Description
PAD_PKCS5	Provides padding which complies with the PKCS #5: Password-Based Cryptography Standard
PAD_NONE	Provides option to specify no padding. Caller must ensure that blocksize is correct, else the package returns an error.
PAD_ZERO	Provides padding consisting of zeroes.

Copyright © University of Science – HCM-VNU 2014

76



77



78

Thảo luận

# Câu hỏi

TS. Phạm Thị Bạch Huệ - [ptbhue@fit.hcmus.edu.vn](mailto:ptbhue@fit.hcmus.edu.vn)  
 Ths. Lương Vĩ Minh - [lvminh@fit.hcmus.edu.vn](mailto:lvminh@fit.hcmus.edu.vn)

 **cdio™**

Copyright © University of Science – HCM-VNU 2014

79