**Bài tập 1: Cho bảng nhân viên**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| MãNV | Tên | Chức vụ | Đơn vị | Tuổi | Giới tính | Lương |
| 01 | Minh | NV | HVKTMM | 40 | F | 3000 |
| 02 | Quân | NV | TTCNTT | 50 | M | 3000 |
| 03 | Hùng | TB | TTCNTT | 51 | M | 5600 |
| 04 | Dũng | KT | BCY | 49 | M | 5000 |
| 05 | Hưng | KT | BCY | 60 | F | 4500 |
| 06 | Giang | PB | BCY | 38 | M | 3200 |

**C = ( Đơn vị = “BCY”) ^ (Chuc vu = “KT”) ^ (Giới tính = “M”)**

1. Thống kê trên có nhạy cảm không?
2. Thực hiện tấn công trình theo dõi. Xác định lương thông qua Sum. (k =2)
3. Thực hiện tấn công hệ phương trình tuyến tính

Thống kê trên là nhạy cảm. Vì Count (C ) =1 -> câu truy vấn này bị cấm.

**Kiểu 1: Công thức**

|  |
| --- |
| Tính T = A ^ thỏa mãn k< |T| < N-k  Tính gián tiếp Count (C ):  Count (C ) = Count ( A^B) = Count (A) – Count( A ^ )  Count(C ) = Count (A) – Count (T) |

Ta có A = (Đơn vị = “BCY”) và B = (Chucvu = “KT”) ^ (Gioitinh = “M”)

Tính T = A ^ thỏa mãn k< |T| < N-k

Count (A) = {04,05,06} = 3

Count(A ^ ) = Count(T) = {05,06} =2

Count (C ) = Count ( A^B) = Count (A) – Count( A ^ ) = Count (A) – Count (T)

= 3 -2 =1

Sum(Luong,C) = Sum(Luong,A) – Sum(Luong, A^ )

= 12700 – 7700 = 5000

**Kiểu 2: Công thức**

|  |
| --- |
| Chọn T thỏa mã: k <|T| , || < N –k  Q(D) = Q(All) = Q(T) + Q()  Q( C) = Q(C v T\_) + Q(C v T) – Q(D) |

Chọn T = (Tuoi<50) -> Count (T) = 3 > 2 và Count () = 3 < 4

Count(D) = Count(All) = Count (T) + Count( = 6

Count (C v T) = {01,04,06} = 3

Count (C v = {02,03,04,05} = 4

Count( C) = Count(C v T\_) + Count(C v T) – Count(D)

= 3 + 4 – 6 =1

Sum(Luong,C) = Sum(Luong,C v T) + Sum(Luong, C v ) – Sum (Luong,D)

= 11200 + 18100 – 24300 = 5000

**Hệ phương trình tuyến tính**

**C = ( Đơn vị = “BCY”) ^ (Chuc vu = “KT”) ^ (Giới tính = “M”)**

Tính q1 = Count(Donvi= “BCY”)

q2 = Count(Donvi= “BCY”, Tuoi # 49)

Count(q1) = x4 + x5 + x6

Count (q2) = x5 + x6

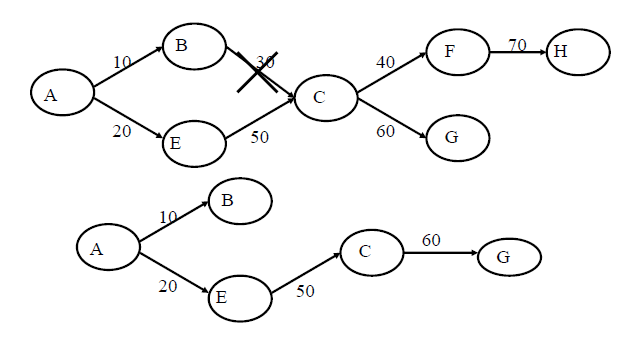
q3 = count(c ) = ( Đơn vị = “BCY”) ^ (Chuc vu = “KT”) ^ (Giới tính = “M”)

= q1 – q2 = 1

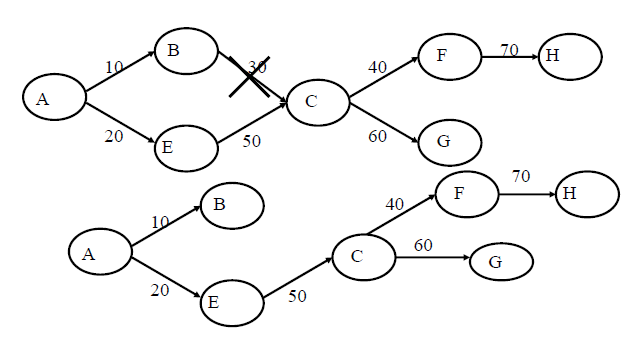
Sum(Luong,c) = Sum(Luong,q1) – Sum(luong,q2) = 12700 – 7700 = 5000

**Bài tập 2**

**Thu hồi đệ quy (recursive revocation) :** khi người dùng A thu hồi quyền truy xuất trên bảng của một người B thì tất cả các quyền mà B đã gán cho người khác đều bị thu hồi.

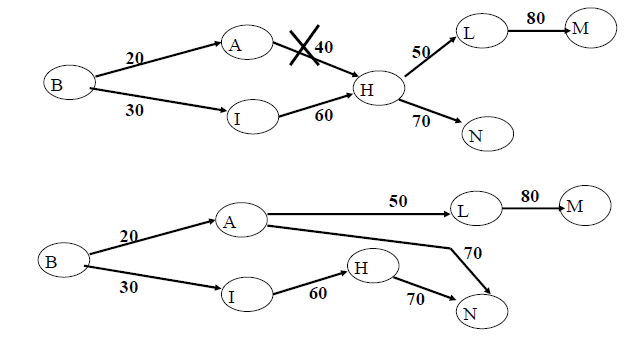
****

**Một biến thể của thu hồi đệ quy:** là không dựa vào nhãn thời gian, mục đích là để tránh thu hồi quyền dây chuyền.

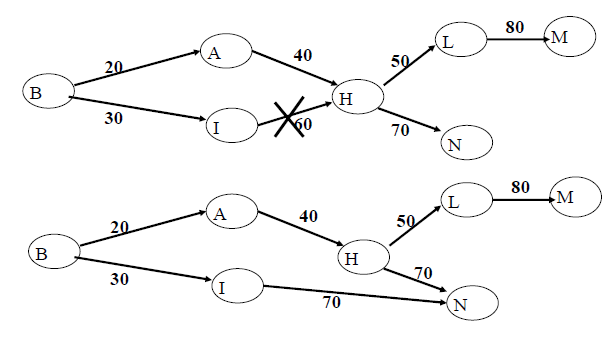


Khi đó, nếu C bị B thu hồi quyền và C lại có quyền tương đương do người khác cấp (mặc dù sau đó) thì quyền mà C cấp cho người khác vẫn được giữ.

**Thu hồi không xếp chồng (**noncascading revoke): Khi A thu hồi quyền truy xuất trên B thì tất cả quyền truy xuất mà B đã cấp cho chủ thể khác được thay bằng A đã cấp cho những chủ thể này.



Với những quyền H đã được cấp bởi C ≠ A, đến lượt B cấp cho người khác(N) thì B vẫn là người cấp các quyền này.



Với quyền mà H cấp cho L, sau khi thu hồi quyền, không được thay I như là người cấp vì quyền này được cấp trước khi I cấp quyền cho H.

**Bài tập 3**

1. Kiến trúc khóa toàn vẹn Integrity Lock là gì?

2. Cho bảng dữ liệu sau:

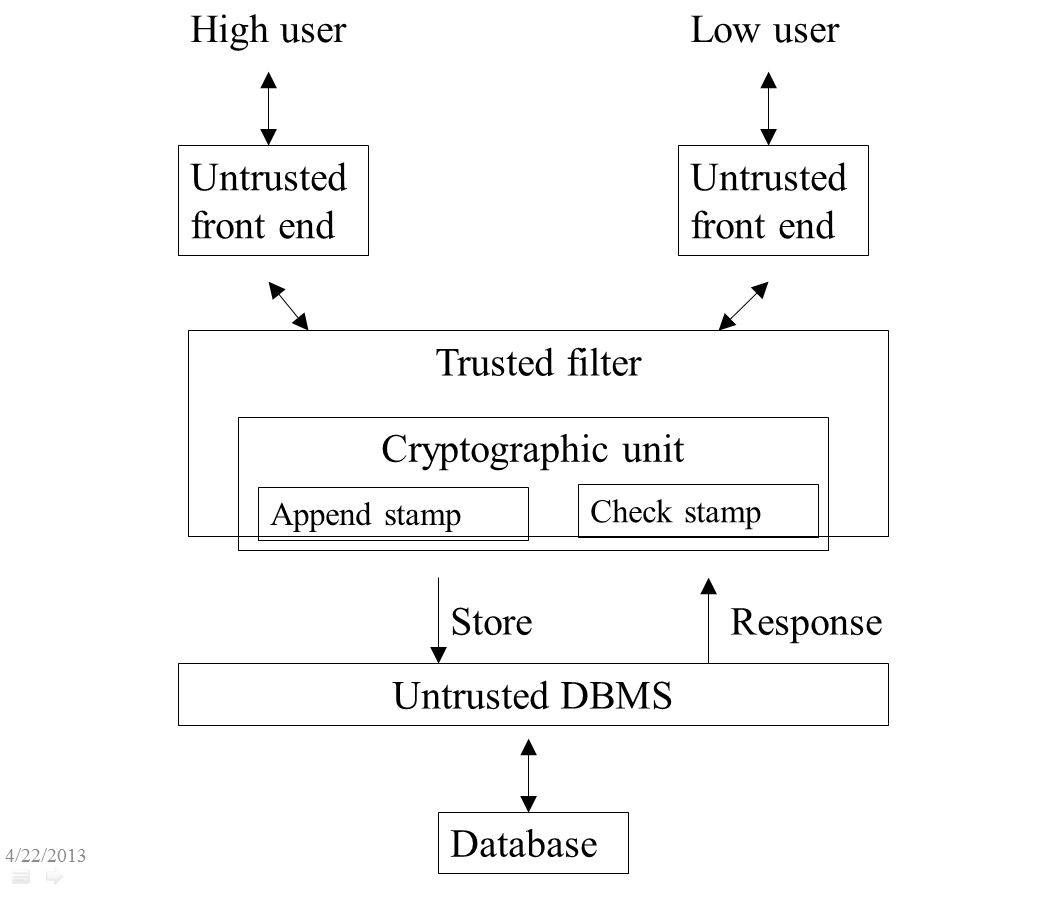
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **HoTen** | **Label** | **Tuoi** | **Label** | **Nghề nghiệp** | **Label** | **Thu nhập** | **Label** |
| Bình | C | 45 | C | Cảnh sát | S | 10000 | S |
| Trung | S | 51 | C | Giáo viên | C | 44000 | S |
| Hiếu | TS | 33 | C | Bộ đội | TS | 74500 | TS |

- Sử dụng kiến trúc Integrity Lock hãy gắn tem cho cơ sở dữ liệu trên, với 4 khóa bí mậ*t k1, k2, k3, k4*, mỗi khóa cho một cột.

- Một người dùngcó nhãn *S*, thì có thể đọc được những gì trong bảng trên sau khi đã gắn tem.

- Trình bày tính toàn vẹn khi các người dùng truy xuất dữ liệu vào bảng trên.

**Kiến trúc khóa toàn vẹn:**

****

Sử dụng cơ chế gán tem dùng mật mã khóa bí mật k1,k2,k3,k4 mỗi khóa cho một cột:

*Stamp = Ek(Dữ liệu + độ nhậy cảm user)*

Mã hoá (dữ liệu + độ nhạy cảm user) bằng một khoá bí mật K, tạo ra *tem,* và lưu vào trong CSDL cùng với mục dữ liệu đó (gắn với mục dữ liệu).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **HoTen** | **Stamp** | **Tuoi** | **Stamp** | **Nghề nghiệp** | **Stamp** | **Thu nhập** | **Stamp** |
| Bình(C) | Ek1(Bình+C) | 45(C ) | Ek2(45+C) | Cảnhsát(S) | Ek3(cảnh sat+ S) | 10000(S) | Ek4(10000+ S) |
| Trung(S) | Ek1(Trung+S) | 51(C ) | Ek2(51+C) | Giáoviên(C) | Ek3(giáo viên+ C) | 44000(S) | Ek4(44000+ S) |
| Hiếu(TS) | Ek1(Hiếu+TS) | 33(C ) | Ek2(33+C) | Bộ đội(TS) | Ek3(bộ đội+TS) | 74500(TS) | Ek4(74000+ S) |

**Một người dùngcó nhãn *S*, thì có thể đọc được những gì trong bảng trên sau khi đã gắn tem?**

Một người dùng có nhãn S thì đọc được đối tượng có mức nhạy cảm S và C.

**Kiểm tra tính toàn vẹn khi các người dùng truy xuất dữ liệu vào bảng trên:**

Khi người dùng lấy dữ liệu từ Database , TFE check stamp để kiểm tra tính toàn vẹn bằng cách giải mã tem và so sánh với dữ liệu nhận được. Nếu không trùn thì dữ liệu đã bị thay đổi.

**Bài tập 4: Thiết kế ATCSDL**

Học viện ktmm có nhiều phòng khoa. Mỗi phòng khoa có nhiều cán bộ. Học viện có nhiều phân viện nghiên cứu, mỗi phân viện tham gia nhiều đề tài, mội đề tài có thể do nhiều phân viện kết hợp nghiên cứu. Mỗi đề tài do một cán bộ làm chủ nhiệm đề tài, một cán bộ có thể làm chủ nhiệm của nhiều đề tài cùng một lúc. Mỗi một đề tài cần nhiều tài liệu, một tài liệu có thể được dùng để nghiên cứu trong nhiều đề tài.

* Thiết kế CSDL thực hiện cho bài toán trên
* Viết các câu lệnh SQL để thực hiện
* Thực hiện thiết kế ATCSDL

HVKTMM

Phòng khoa

1 n

1 n

Cán bộ

n

n

n 1

Phân viện nghiên cứu

Đề tài

n

n

Tài liệu

Phòng Khoa(mã phòngkhoa, tên phòngkhoa)

Cán bộ(mã CB, tên CB, địa chỉ, mã phânviện, mã đềtài)

Đề tài ( mã đềtài, tên đềtài, mã tàiliệu)

Phân viện( mã phânviện, tên phânviện, mã đềtài)

Tài liệu( mã tàiliêu, tên tàiliệu)

create database hvktmm

create table phongkhoa

(

maphongkhoa char(5) primary key,

tenphongkhoa varchar(50) not null,

)

create table canbo

(

macanbo char(5) primary key,

ten char(10) not null,

diachi varchar(50) not null,

maphanvien char(5) not null,

madetai char(5) not null

)

create table detai

(

madetai char(5) Primary key,

tendetai char(50) not null,

matailieu char(50) not null,

)

create table phanvien

(

maphanvien char(5) Primary key,

tenphanvien char(50) not null,

madetai char(50) not null,

)

create table tailieu

(

matailieu char(5) primary key,

tentailieu varchar(50) not null,

)

alter table canbo add constraint fk\_maphanvien foreign key (maphanvien) references phanvien(maphanvien)

alter table canbo add constraint fk\_madetai foreign key (madetai) references detai(madetai)

alter table detai add constraint fk\_matailieu foreign key (matailieu) references tailieu(matailieu)

alter table phanvien add constraint fk\_madetai foreign key (madetai) references detai(madetai)

**Thực hiện các thiết kế an toàn:**

* Xem xét các yêu cầu an toàn bảo đảm tính sẵn sàng, bí mật và toàn vẹn cho hệ thống
* Xem xét xem cần bảo vệ bảng dữ liệu nào, những bảng nào có chứa dữ liệu nhạy cảm. Ví dụ như bảng điểm có trường điểm thi.
* Lựa chọn hệ quản trị csdl phù hợp có tính bảo mật cao như Oracle hay SQL
* Lựa chọn dùng các phương pháp để đảm bảo an toàn như:

+Dùng chính sách cấp quyền DAC, MAC

+Dùng cơ chế sao lưu phục hồi dữ liệu

+Dùng cơ chế mã hoá dữ liệu