第四章 数据库安全性

Principles of Database Systems

数据库安全性

- 问题的提出：
  - 数据库的一大特点是数据可以共享
  - 数据共享必然带来数据库的安全性问题
  - 数据库系统中的数据共享不能是无条件的共享
- 例：

  军事秘密、国家机密、

  新产品实验数据、

  市场需求分析、市场营销策略、销售计划、

  客户档案、医疗档案、银行储蓄数据……

数据库安全性

第四章　数据库安全性

4.1 安全性概述

1. 定义

数据库的安全性是指保护数据库以防止不合法的使用所造成的数据泄露、更改或破坏。

2. 重要性

系统的安全保护措施是否有效是数据库系统的主要性能指标之一。

3. 计算机系统三类安全性问题

技术安全

管理安全

政策法律类

4.1.1 数据库的不安全因素

1）非授权用户对数据库的恶意存取和破坏

- 安全措施：用户身份鉴别、存取控制、和视图等

2）数据库中重要或敏感数据被泄露

- 安全措施：审计、日志、入侵检测等

3）安全环境的脆弱性，包括：硬件、OS、网络等。

- 可信计算机系统

4.1.2 安全标准简介

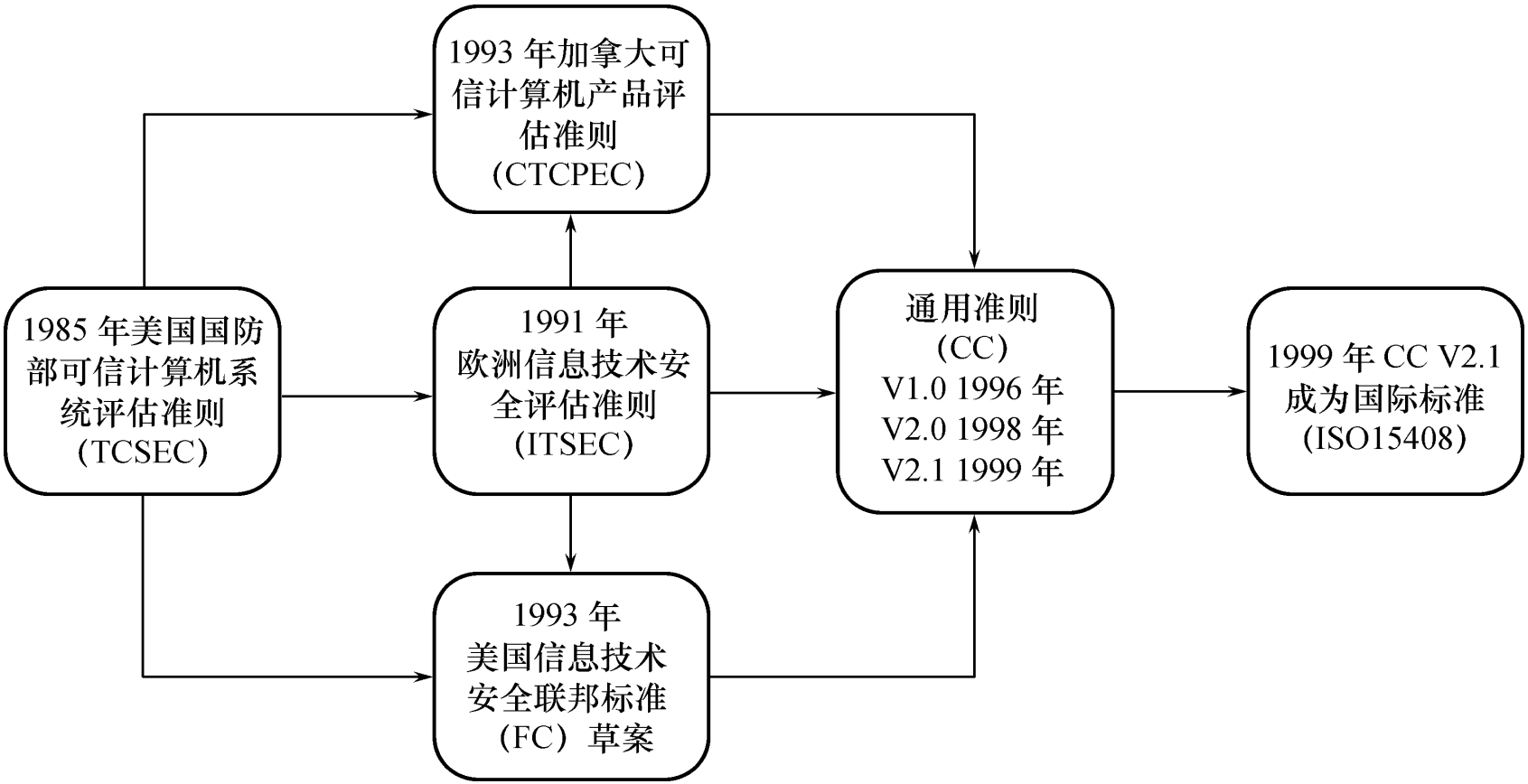
TCSEC：20世纪60年代后期，美国国防部（DOD）开始对计算机安全评估标准进行研究，并在1985年发布了《DoD可信计算机系统评估标准》（TCSEC，即桔皮书）。后来又颁布了《可信计算机系统评估标准关于可信数据库系统的解释》（TDI，即紫皮书）。

ITSEC：法、英、荷、德欧洲四国90年代初联合发布信息技术安全评估标准（ITSEC，欧洲白皮书），它提出了信息安全的机密性、完整性、可用性的安全属性。

CC：1996年，由六个国家（美、加、英、法、德、荷）联合提出了信息技术安全评价的通用标准（CC）。该标准提出了目前国际上公认的表述信息技术安全性的结构。

ISO 15408：1999年，CC 2.1版被ISO采纳为国际标准ISO 15408。

4.1.2 安全标准简介



信息安全标准的发展历史

4.1.2 安全标准简介

- **TCSEC/TDI安全级别划分**
  - TCSEC/TDI，从*四个方面*来描述安全性级别划分的指标：安全策略、责任、保证、文档。

安 全 级 别	定　　义
A1	验证设计（Verified Design）
B3	安全域（Security Domains），安全域TCB
B2	结构化保护（Structural Protection），安全策略模型
B1	标记安全保护（Labeled Security Protection），MAC
C2	受控的存取保护（Controlled Access Protection），个人注册
C1	自主安全保护（Discretionary Security Protection），DAC
D	最小保护（Minimal Protection）

按系统可靠或可信程度逐渐增高
各安全级别之间：偏序向下兼容

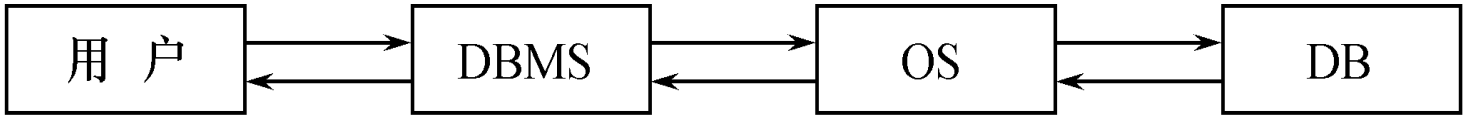
4.1.2 安全标准简介

- CC：提出国际公认的表述信息技术安全性的结构
  把信息产品的安全要求分为：安全功能要求、安全保证要求
  - CC评估保证级划分：

评估保证级	定　　义	TCSEC安全级别（近似相当）
EAL1	功能测试（functionally tested）	
EAL2	结构测试（structurally tested）	C1
EAL3	系统地测试和检查（methodically tested and checked）	C2
EAL4	系统地设计、测试和复查（methodically designed，tested， and reviewed）	B1
EAL5	半形式化设计和测试（semiformally designed and tested）	B2
EAL6	半形式化验证的设计和测试（semiformally verified design and tested）	B3
EAL7	形式化验证的设计和测试（formally verified design and tested）	A1

4.2 数据库安全控制技术

- 计算机系统中，安全措施是一级一级层层设置



计算机系统的安全模型

- 非法使用数据库的情况
  - 编写合法程序绕过DBMS及其授权机制
  - 直接或编写应用程序执行非授权操作
  - 通过多次合法查询数据库从中推导出一些保密数据

- 数据库安全性控制的常用方法：
  - 用户标识和鉴定、存取控制、视图、审计、密码存储

4.2.1 用户身份鉴别

- 用户标识与鉴别（Identification & Authentication）：
  - 系统提供的最外层安全保护措施；
  - 系统提供一定的方式让用户标识自己的名字和身份，系统进行核实，通过鉴定后才提供系统使用权。

常用鉴别方法：

- 口令：静态（易被窃取）、动态（一次一密）
- 生物特征识别：指纹、声音、照片等
- 智能卡
- 回答问题

4.2.2 存取控制

1. 什么是存取控制？

对于获得上机权的用户还要根据系统预先定义好的外模式（视图）或用户权限进行存取控制，保证用户只能存取他有权存取的数据。

2. 方法

—定义用户权限

—合法权限检查

3.常用存取控制方法

- 自主存取控制（Discretionary Access Control，DAC），C1级，灵活
- 强制存取控制（Mandatory Access Control，MAC），B1级，严格

4.2.2 存取控制

自主存取控制

自主存取控制（Discretionary Access Control，简称DAC）在TCSEC/TDI安全级别中处于C1级。是由用户或DBA定义存取权限的一种控制策略。

- 访问权限由两个要素组成：数据对象和操作类型
  - **数据对象**：模式、子模式、表、视图、索引、属性列
  - **操作类型**：create，select，update，insert，delete
- 系统通过控制数据对象的访问权限防止非授权访问
- SQL语言的数据访问控制命令

  授权命令GRANT

  收回授权REVOKE

4.2.4 授权与回收

发出GRANT：

- DBA
- 数据库对象创建者（即属主Owner）
- 拥有该权限的用户

接受权限的用户

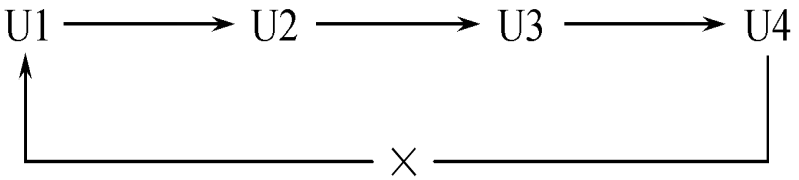
- 一个或多个具体用户
- PUBLIC（全体用户）

1.GRANT

- GRANT语句的一般格式：

```
GRANT <权限>[,<权限>]...
[ON <对象类型> <对象名>]
TO <用户>[,<用户>]...
[WITH GRANT OPTION];
```

- 语义：将对指定操作对象的指定操作权限授予指定的用户
- WITH GRANT OPTION子句:
  - 指定：可以再授予
  - 没有指定：不能传播
- 不允许循环授权



4.2.4 授权与回收

例1 把查询Student表权限授给用户U1。

GRANT SELECT ON TABLE Student TO U1;

例2 把对Student表和Course表的全部权限授予用户U2和U3。

GRANT ALL PRIVILIGES ON TABLE Student, Course TO U2, U3;

例3 把对表SC的查询权限授予所有用户。

GRANT SELECT ON TABLE SC TO PUBLIC;

例4 把查询Student表和修改学生学号的权限授给用户U4。

GRANT UPDATE(Sno), SELECT ON TABLE Student TO U4;

对属性列的授权时必须明确指出相应属性列名。

4.2.4 授权与回收

例5 把对表SC的INSERT权限授予U5用户，并允许他再将此权限授予其他用户。

GRANT INSERT ON TABLE SC TO U5 WITH GRANT OPTION;

执行此SQL语句后，U5不仅拥有了对表SC的INSERT权限，还可以传播此权限，即由U5用户发上述GRANT命令给其他用户。

如：U5可以将此权限授予U6：

GRANT INSERT ON TABLE SC TO U6 WITH GRANT OPTION;

同样，U6还可以将此权限授予U7：

GRANT INSERT ON TABLE SC TO U7;

因为U6未给U7传播的权限，因此U7不能再传播此权限。

4.2.4 授权与回收

2. REVOKE

- 授予的权限可以由DBA或其他授权者用REVOKE语句收回。
- REVOKE语句的一般格式为：

```
REVOKE <权限>[,<权限>]...
[ON <对象类型> <对象名>]
FROM <用户>[,<用户>]…[CASCADE | RESTRICT];
```

[例] 把用户U4修改学生学号的权限收回。

```
REVOKE UPDATE(Sno)
ON TABLE Student
FROM U4;
```

4.2.4 授权与回收

[例] 把用户U5对SC表的INSERT权限收回

```
REVOKE INSERT
ON TABLE SC
FROM U5 CASCADE ;
```

- 将用户U5的INSERT权限收回的时候必须级联（CASCADE）收回
- 系统只收回直接或间接从U5处获得的权限

自主存取控制的缺陷——用户可以自由地决定将数据地存取权限授予任何人、决定是否将“授权”的权限授予别人，而系统对此无法控制。

原因——仅通过对数据的存取权限来进行安全控制，而数据本身并无安全性标记。

4.2.4 授权与回收

3.创建数据库模式的权限

- DBA在创建用户时实现。
- CREATE USER语句格式：

CREATE USER <username>

[WITH] [DBA | RESOURCE | CONNECT];

拥有的权限	可否执行的操作			
	CREATE USER	CREATE SCHEMA	CREATE TABLE	登录数据库 执行数据查询和操纵
DBA	可以	可以	可以	可以
RESOURCE	不可以	不可以	可以	可以
CONNECT	不可以	不可以	不可以	可以，但必须拥有相应权限

权限与可执行的操作对照表

4.2.5 数据库角色

- 数据库角色：被命名的一组与数据库操作相关的权限
  - 角色是权限的集合；
  - 可以为一组具有相同权限的用户创建一个角色；
  - 简化授权的过程。

- **1. 角色的创建**

```
CREATE  ROLE  <角色名>
```

- **2. 给角色授权**

```
GRANT  <权限> [，<权限>] …
ON <对象类型>对象名
TO <角色> [，<角色>] …
```

4.2.5 数据库角色

3. 将一个角色授予其他的角色或用户

GRANT <角色1>[，<角色2>]…
TO <角色3>[，<用户1>]…
[WITH ADMIN OPTION]

4. 角色权限的收回

REVOKE <权限>[，<权限>]…
ON <对象类型> <对象名>
FROM <角色>[，<角色>]…

4.2.5 数据库角色

[例] 通过角色来实现将一组权限授予一个用户。

步骤如下：

1. 首先创建一个角色 R1

```
CREATE ROLE R1；
```

2. 然后使用GRANT语句，使角色R1拥有Student表的SELECT、UPDATE、INSERT权限

```
GRANT SELECT，UPDATE，INSERT
ON TABLE Student
TO R1；
```

4.2.5 数据库角色

3. 将这个角色授予王平，张明，赵玲。使他们具有角色R1所包含的全部权限

GRANT R1 TO 王平，张明，赵玲；

4. 可以一次性通过R1来回收王平的这3个权限

REVOKE R1 FROM 王平；

5. 角色的权限修改

GRANT DELETE ON TABLE Student TO R1；