上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

SJTU

数 据 库 技 术

郭 捷

(guojie@sjtu.edu.cn)

饮 水 思 源 • 爱 国 荣 校

第七章 数据库安全

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

目 录

数据库安全性—问题的提出

- 问题的提出：
  - 数据库的一大特点是数据可以共享；
  - 但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

数据库的不安全因素

- 数据库的安全性，是指保护数据库以防止不合法使用所造成的数据泄漏、更改或破坏。
- 产生威胁的因素：
  - 非授权用户对数据库的恶意存取和破坏；
  - 数据库中重要或敏感的数据被泄露；
  - 安全环境的脆弱性；

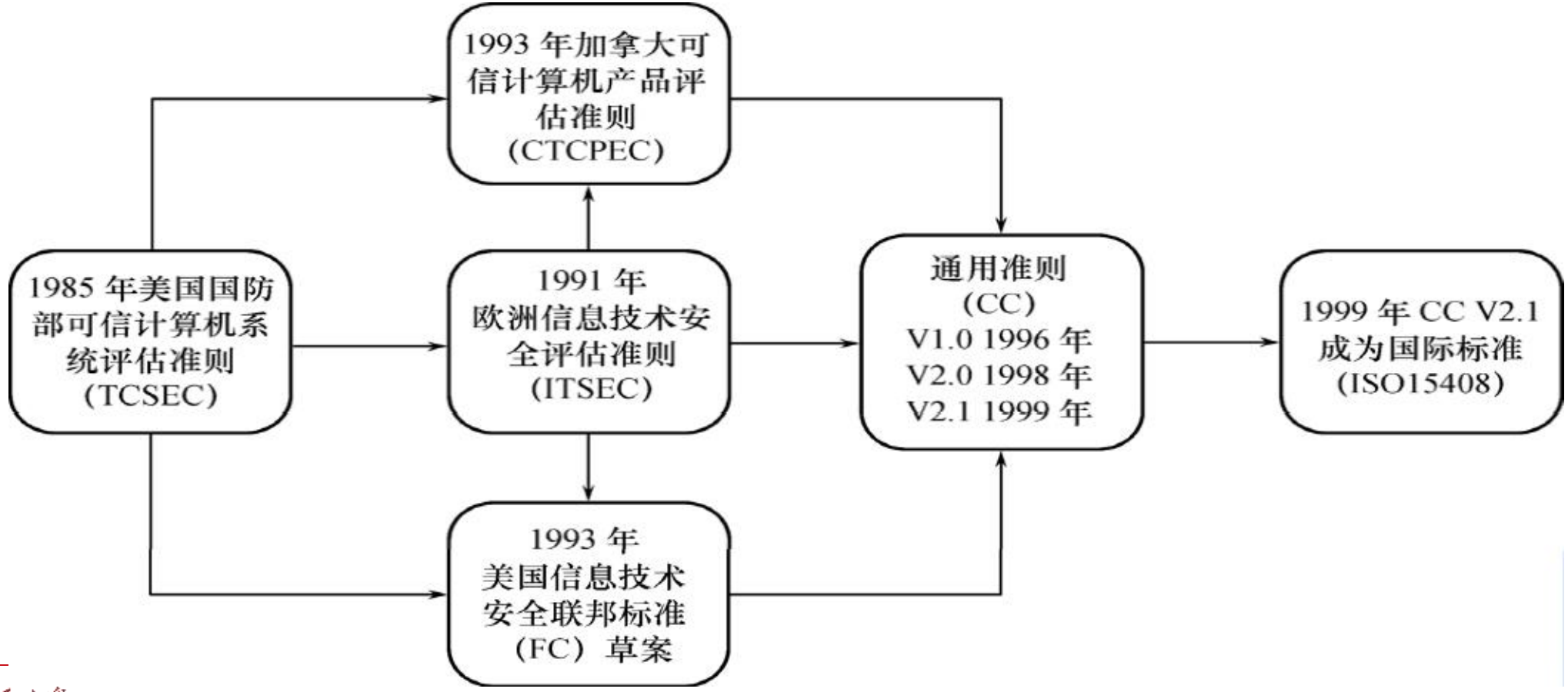
计算机系统的三类安全性问题

- 计算机系统安全性：为计算机系统建立和采取的各种安全保护措施，以保护计算机系统中的硬件、软件及数据，防止其因偶然或恶意的原因使系统遭到破坏，数据遭到更改或泄露等。

- 三类计算机系统安全性问题：
  - 技术安全类
  - 管理安全类
  - 政策法律类

可信计算机系统评测标准

- 为降低进而消除对系统的安全攻击，各国引用或制定了一系列安全标准：



可信计算机系统评测标准

- 1985年美国国防部（DoD）正式颁布《 DoD可信计算机系统评估标准》（简称TCSEC或DoD85）
  - TCSEC又称桔皮书
  - TCSEC标准的目的：
    - 提供一种标准，使用户可以对其计算机系统内敏感信息安全操作的可信程度做评估。
    - 给计算机行业的制造商提供一种可循的指导规则，使其产品能够更好地满足敏感应用的安全需求。

可信计算机系统评测标准

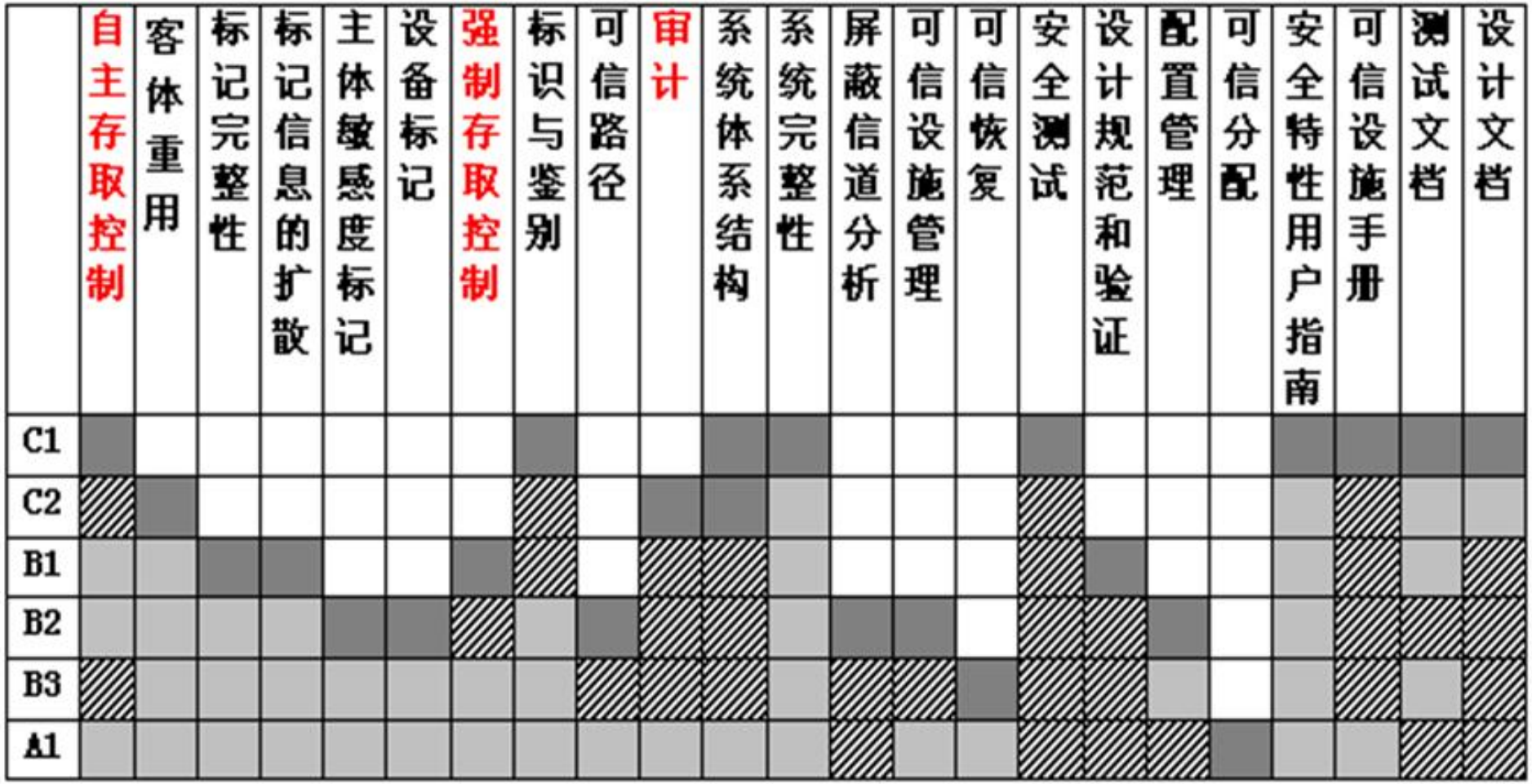
- 1991年4月美国NCSC（国家计算机安全中心）颁布了《可信计算机系统评估标准关于可信数据库系统的解释》（TCSEC/ Trusted Database Interpretation 简称TCSEC/TDI）
  - TDI又称紫皮书，它将TCSEC扩展到数据库管理系统。
  - TDI中定义了数据库管理系统的设计与实现中需满足和用以进行安全性级别评估的标准。

TCSEC/TDI安全级别划分

类别	级别	名称	主要特征
D	D	最小保护	没有安全保护， 如ms-dos
C	C1	自主安全保护	实现自主存取控制DAC，具有识别与授权的责任，如早期UNIX系统
	C2	受控存储控制	安全产品的最低档，提供受控的存取保护，将C1的DAC进一步细化，实施审计和资源隔离，如windows 2000 和Oracle 7
B	B1	标识安全保护	对系统数据加以标记，实施强制存取控制MAC和审计，如Oracle公司的Trusted Oracle 7，Sybase公司的Secure SQL Server version 11.0.6，Informix公司的Incorporated INFORMIX Secure 5.0
	B2	结构化保护	除满足B1要求外，要实行强制性的控制并进行严格的保护，如操作系统Trusted Xenix系统。
	B3	安全域	提供可信设备的管理和恢复,即使计算机崩溃也不会泄露系统信息。如Honeywell Federal Systems XTS-200
A	A	验证设计	形式化的最高级描述和验证

TCSEC/TDI安全级别划分

不同安全级别对安全指标的支持情况



□ 表示该级不提供对该指标的支持；■ 表示该级新增的对该指标的支持；▒ 表示该级对该指标的支持与相邻低一级的等级一样；▨ 表示该级对该指标的支持较下一级有所增加或改动。

可信计算机系统评测标准

CC标准

- 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：规范产品和系统的安全行为；
  - 安全保证要求：解决如何正确有效地实施这些功能。

可信计算机系统评测标准

CC评估保证级划分

评估保证级	定义	TCSEC安全级别（近似）
EAL 1	功能测试 (functionally tested)	
EAL 2	结构测试 (structurally tested)	C1
EAL 3	系统地测试和检查 (methodically tested and checked)	C2
EAL 4	系统地设计、测试和复查 (methodically designed, tested and reviewed)	B1
EAL 5	半形式化设计和测试(semiformally designed and tested)	B2
EAL 6	半形式化验证的设计和测试(semiformally verified design and tested)	B3
EAL 7	形式化验证的设计和测试(formally verified design and tested)	A1

数据库安全的定义

- **最具有代表性的数据库安全定义：**

By C. P. Pfleeger ——《Security in Computing – Database Security. PTR，1997》

(1) 物理数据库的完整性：数据库中的数据不被各种自然的或物理的问题而破坏，如电力问题或设备故障等。

(2) 逻辑数据库的完整性：对数据库结构的保护，如对其中一个字段的修改不应该破坏其他字段。

(3) 元素安全性：存储在数据库中的每个元素都是正确的。

(4) 可审计性：可以追踪存取和修改数据库元素的用户。

数据库安全的定义

- **最具有代表性的数据库安全定义：**

By C. P. Pfleeger ——《Security in Computing – Database Security. PTR，1997》

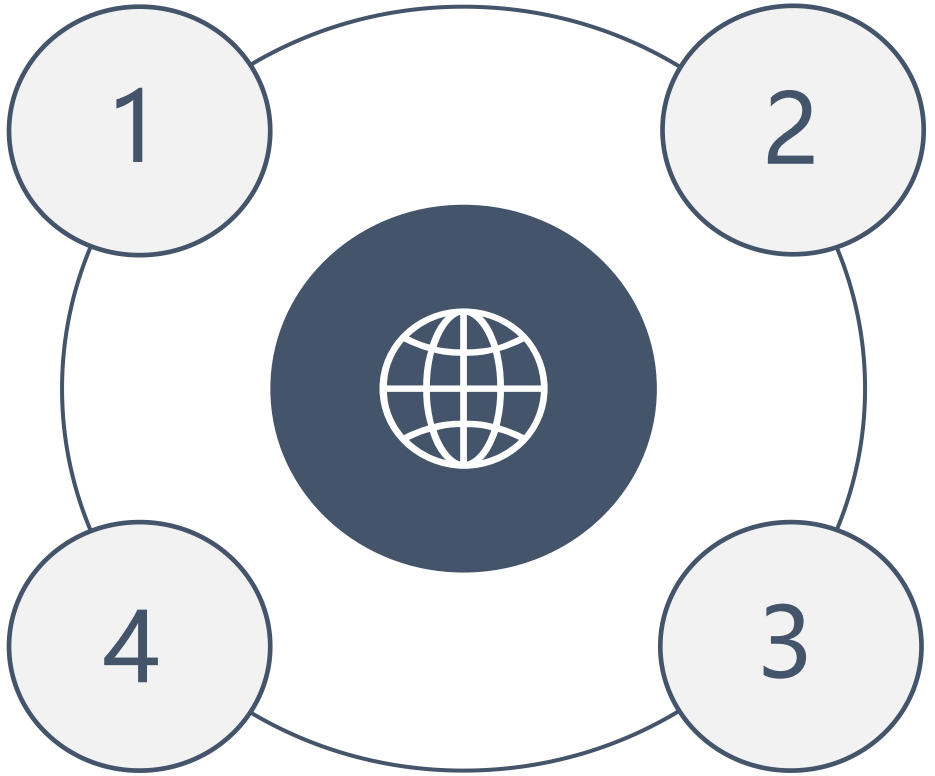
(5) 访问控制：确保只有授权的用户才能访问数据库，不同的用户被限制不同的访问方式。

(6) 身份验证：不管是审计追踪或者是对某一数据库的访问都要经过严格的身份验证。

(7) 可用性：对授权的用户应该随时可进行应有的数据库访问。

数据库安全的定义

- 我国在《计算机信息系统安全保护等级划分准则》中对数据库安全的定义：



保密性

保护数据库中的数据不被泄露和未授权的获取

完整性

保护数据库中的数据不被破坏和删除

一致性

确保数据库中的数据满足实体完整性、参照完整性和用户定义完整性要求

可用性

确保数据库中的数据不因人为的和自然的原因对授权用户不可用

第七章 数据库安全

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

目录

数据库安全性控制概述

▣非法使用数据库的情况

◆用户编写一段合法的程序绕过DBMS及其授权机制，通过操作系统直接存取、修改或备份数据库中的数据；

◆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；

◆破坏安全性的行为可能是无意的，故意的，恶意的。

数据库的安全需求

防止非法数据访问

- 数据库安全最关键的需求之一，仅允许授权的合法用户访问数据库；

数据的分级保护

- 依据数据敏感级别进行多级保护，严格控制对敏感数据的访问请求；

防止推断性攻击

- 防止用户通过授权访问的低安全级数据，推导出敏感数据；

数据库的安全需求

数据库的完整性

- 数据库完整性是指数据库内容的正确性、有效性和一致性。防止更改数据内容的非授权访问，以及病毒、蓄意破坏，或是系统级错误及物理故障（物理完整性）等，由DBMS通过访问控制以及备份、恢复机制等完成保护工作；

数据的操作完整性

- 在并行事务的模式下，保持数据的逻辑一致性，通常采用并行管理器和加锁机制完成；

数据库的安全需求

数据的语义完整性

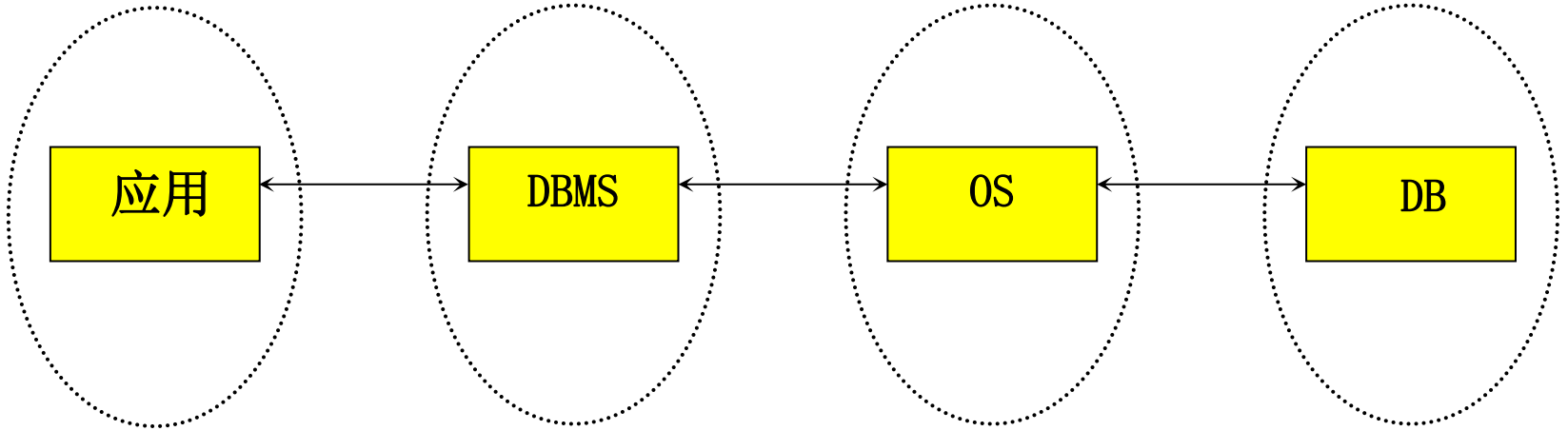
- 确保对数据在允许范围内修改，以保持数据的一致完整性；

审计功能

- 提供数据的物理完整性，并记录下对数据的所有存取访问，根据结果进行分析和追踪；

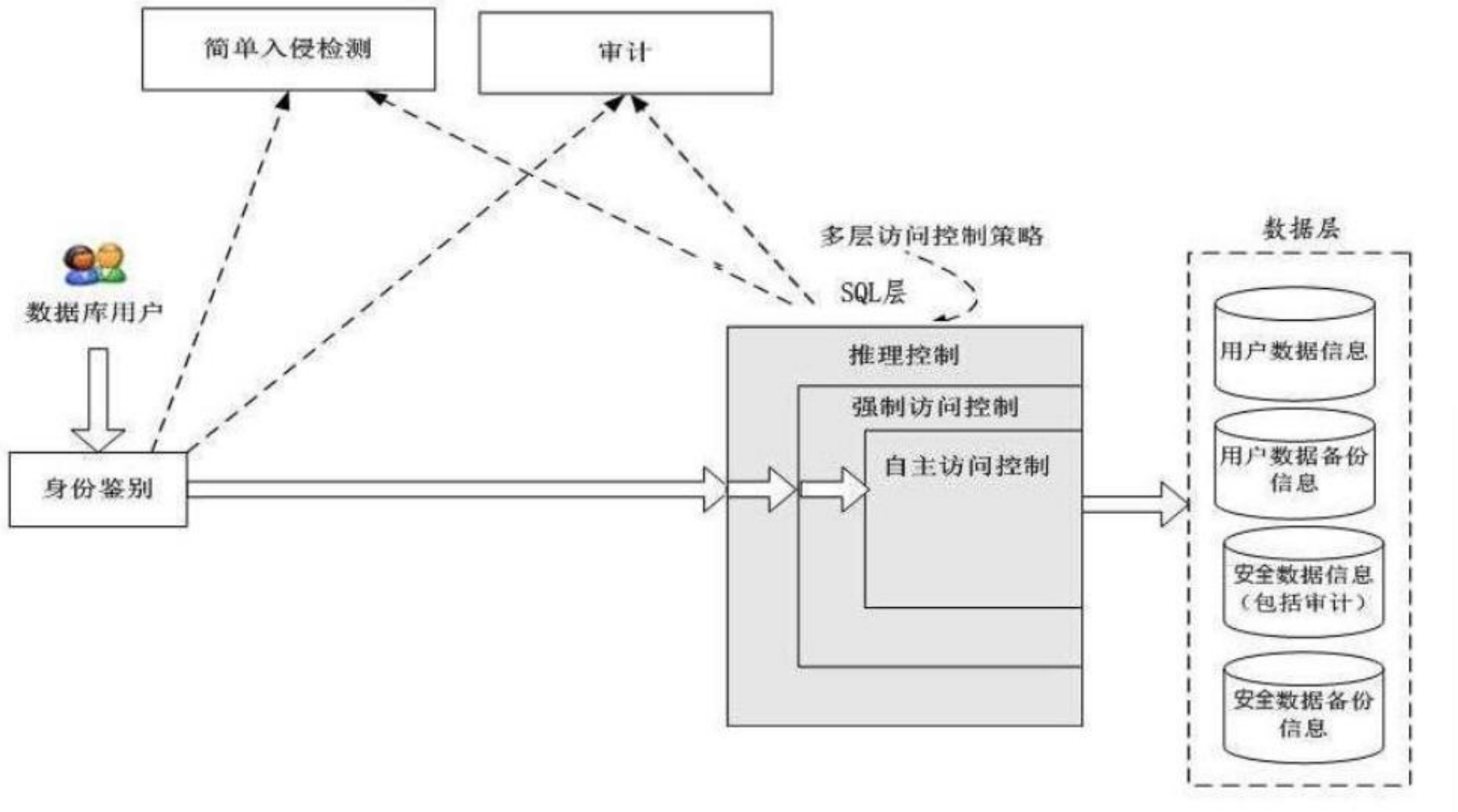
数据库安全性控制的层次

安全性控制层次



方法：	用户标识和鉴定	存取控制 审计 视图	操作系统安全保护	密码存储

数据库管理系统安全性控制模型



用户标识与鉴别

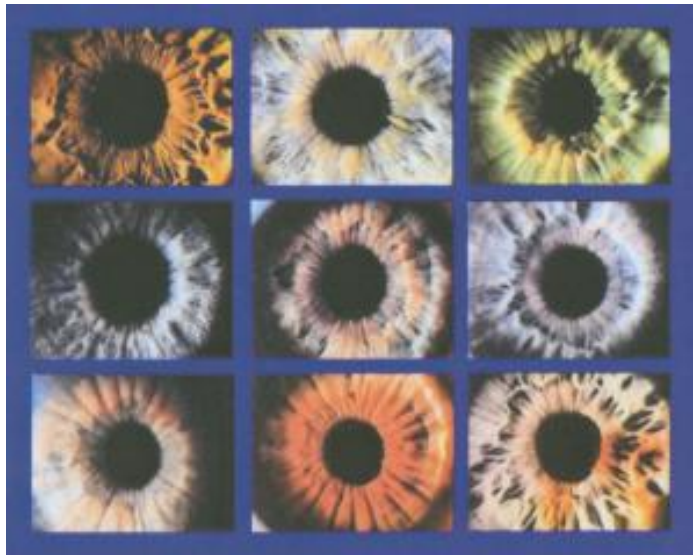
▣ 用户标识与鉴别（Identification & Authentication）

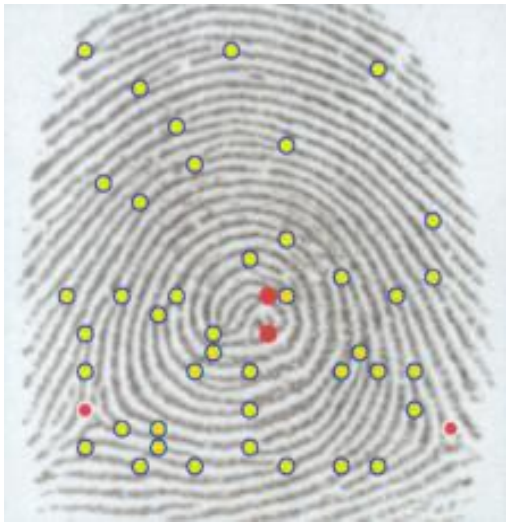
- ◈ 系统提供的最外层安全保护措施
- ◈ 主要方法：
  - ✓ 静态口令鉴别：用户自己设定，口令静态不变
  - ✓ 动态口令鉴别：口令动态变化，一次一密
  - ✓ 生物特征鉴别：生物特征进行认证
  - ✓ 智能卡鉴别：不可复制的硬件，内置基层电路芯片，具有硬件加密功能

用户标识与鉴别

生物认证和生物模板安全

- 生物认证（指纹识别、人脸识别、虹膜识别、声纹识别、笔迹识别等）
- 图像哈希生物认证算法；





存取控制

存取控制机制的组成：

定义用户存取权限

- 用户对某一数据对象的操作权力，称为权限；
- DBMS提供适当的语言来定义用户权限，存放在数据字典中，称作安全规则或授权规则

合法存取权限检查

- 用户发出存取数据库操作请求；
- DBMS查找数据字典，进行合法权限检查；

用户权限定义和合法权检查机制一起组成了DBMS的存取控制子系统。

存取控制

▣常用存取控制方法：

◈ 自主存取控制（Discretionary Access Control ，简称DAC）

用户对不同的数据对象有不同的存取权限，不同的用户对同一对象也有不同的权限，用户还可将其拥有的存取权限转授给其他用户。

- C2级
- 灵活

• 强制存取控制（Mandatory Access Control，简称 MAC）

每一个数据对象被标以一定的密级，每一个用户也被授予某一个级别的许可证。对任意一个对象，只有具有合法许可证的用户才可以存取。

- B1级
- 严格

自主存取控制

- 通过SQL 的GRANT语句和REVOKE语句实现。
- 用户权限组成：
  - 数据对象
  - 操作类型
- 定义用户存取权限：定义用户可以在哪些数据库对象上进行哪些类型的操作。
- 定义存取权限称为授权。

自主存取控制

关系数据库系统中的存取控制对象和存取权限

对象类型	对象	操作类型
数据库模式	模式	CREATE SCHEMA
	基本表	CREATE TABLE, ALTER TABLE
	视图	CREATE VIEW
	索引	CREATE INDEX
数据	基本表和视图	SELECT, INSERT, UPDATE, DELETE, REFERENCES, ALL PRIVILEGES
	属性列	SELECT, INSERT, UPDATE, REFERENCES, ALL PRIVILEGES

授权与回收

一、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];
```

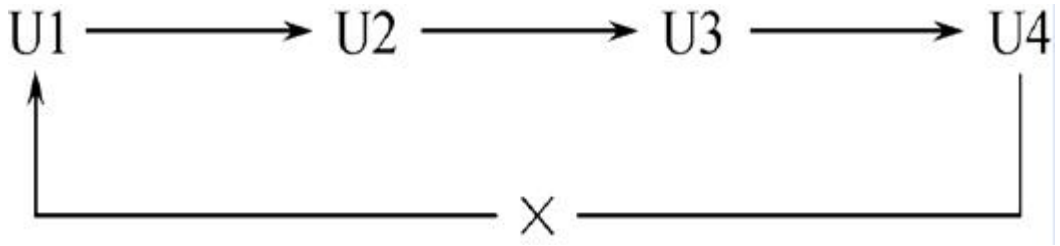
- 语义：将对指定操作对象的指定操作权限授予指定的用户。

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）。

授权与回收

- 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：



授权与回收

▣ 例题：

[例1] 把查询Student表的权限授给用户U1。

```
GRANT SELECT
ON TABLE Student
TO U1;
```

授权与回收

- 例题：

[例2] 把对Student表和Course表的全部操作权限授予用户U2和U3

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

授权与回收

■ 例题：

[例3] 把对表SC的查询权限授予所有用户。

```
GRANT SELECT
ON TABLE SC
TO PUBLIC;
```

授权与回收

▣ 例题：

[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

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;
```

授权与回收

▣ 传播权限：

◈执行例5后，U5不仅拥有了对表SC的INSERT权限，还可传播此权限

[例6] GRANT INSERT ON TABLE SC TO U6

WITH GRANT OPTION;

同样，U6还可以将此权限授予U7：

[例7] GRANT INSERT ON TABLE SC TO U7;

但U7不能再传播此权限。

授权与回收

- 下表是执行了[例1]到[例7]的语句后，学生-课程数据库中的用户权限定义表

授权用户名	被授权用户名	数据库对象名	允许操作类型	能否转授权
DBA	U1	关系Student	SELECT	不能
DBA	U2	关系Student	ALL	不能
DBA	U2	关系Course	ALL	不能
DBA	U3	关系Student	ALL	不能
DBA	U3	关系Course	ALL	不能
DBA	PUBLIC	关系SC	SELECT	不能
DBA	U4	关系Student	SELECT	不能
DBA	U4	属性列Student.Sno	UPDATE	不能
DBA	U5	关系SC	INSERT	能
U5	U6	关系SC	INSERT	能
U6	U7	关系SC	INSERT	不能

授权与回收

二、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>]...;

授权与回收

[例8] 把用户U4修改学生学号的权限收回。

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

授权与回收

［例9］收回所有用户对表SC的查询权限。

```
REVOKE SELECT

ON TABLE SC

FROM PUBLIC;
```

授权与回收

[例10] 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，不然系统将拒绝执行该命令。

◆系统只收回直接或间接从U5处获得的权限。

授权与回收

- 执行[例8]到[例10]的语句后，学生-课程数据库中的用户权限定义表：

授权用户名	被授权用户名	数据库对象名	允许操作类型	能否转授权
DBA	U1	关系Student	SELECT	不能
DBA	U2	关系Student	ALL	不能
DBA	U2	关系Course	ALL	不能
DBA	U3	关系Student	ALL	不能
DBA	U3	关系Course	ALL	不能
DBA	U4	关系Student	SELECT	不能

授权与回收

小结：SQL灵活的授权机制

- DBA：拥有所有对象的所有权限
  - 不同的权限授予不同的用户。
- 用户：拥有自己建立的对象的全部的操作权限
  - GRANT：授予其他用户。
- 被授权的用户
  - “继续授权”许可：再授予。
- 所有授予出去的权力在必要时又都可用REVOKE语句收回

授权与回收

三、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]
```

授权与回收

- 权限与可执行的操作对照表：

拥有的权限	可否执行的操作			
	CREATE USER	CREATE SCHEMA	CREATE TABLE	登陆数据库，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，须有相应权限

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。

数据库角色

一、角色的创建

CREATE ROLE <角色名>

二、给角色授权

GRANT <权限>[，<权限>]…

ON <对象类型>对象名

TO <角色>[，<角色>]…

数据库角色

三、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]

四、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>]…

数据库角色

[例11] 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;
```

数据库角色

［例11］通过角色来实现将一组权限授予一个用户。步骤如下：

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；
```

数据库角色

[例12] 角色的权限修改。

```
GRANT DELETE
ON TABLE Student
TO R1;
```

[例13] 角色的权限修改。

```
REVOKE SELECT
ON TABLE Student
FROM R1;
```

自主存取控制方法

检查存取权限

- 对于获得上机权后又进一步发出存取数据库操作的用户
  - DBMS查找数据字典，根据其存取权限对操作的合法性进行检查
  - 若用户的操作请求超出了定义的权限，系统将拒绝执行此操作

自主存取控制方法

授权粒度

授权粒度是指可以定义的数据对象的范围

- 它是衡量授权机制是否灵活的一个重要指标。
- 授权定义中数据对象的粒度越细，即可以定义的数据对象的范围越小，授权子系统就越灵活。

自主存取控制方法

- 关系数据库中授权的数据对象粒度：
  - 数据库
  - 表
  - 属性列
  - 行
- 能否提供与数据值有关的授权反映了授权子系统精巧程度
- 授权定义中数据对象的粒度越细，即可以定义的数据对象的范围越小，授权子系统就越灵活。

自主存取控制方法

- 缺点：
  - 可能存在数据的“无意泄露”
  - 原因：这种机制仅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。
  - 解决：对系统控制下的所有主客体实施强制存取控制策略。

例: 只有财务人员有权访问职工工资表EMP-Salary

```
CREATE TABLE Salary-copy
AS SELECT Eno, Name, Salary
FROM EMP-Salary;
Grant SELECT
ON TABLE Salary-copy
TO PUBLIC;
```

强制存取控制方法

什么是强制存取控制？

- 强制存取控制(MAC)是指系统为保证更高程度的安全性，按照TDI/TCSEC标准中安全策略的要求，所采取的强制存取检查手段。
- MAC不是用户能直接感知或进行控制的。
- MAC适用于对数据有严格而固定密级分类的部门
  - 军事部门
  - 政府部门

强制存取控制方法

主体与客体

- 在MAC中，DBMS所管理的全部实体被分为主体和客体两大类
- 主体是系统中的活动实体
  - DBMS所管理的实际用户
  - 代表用户的各进程
- 客体是系统中的被动实体，是受主体操纵的
  - 文件
  - 基表
  - 索引
  - 视图

强制存取控制方法

敏感度标记

- 对于主体和客体，DBMS为它们每个实例（值）指派一个敏感度标记（Label）
- 敏感度标记分成若干级别：
  - 绝密（Top Secret，TS）
  - 机密（Secret，S）
  - 可信（Confidential，C）
  - 公开（Public，P）

强制存取控制方法

- 主体的敏感度标记称为许可证级别（Clearance Level）；
- 客体的敏感度标记称为密级（Classification Level）；
- MAC机制就是通过对比主体的Label和客体的Label，最终确定主体是否能够存取客体；

强制存取控制方法

强制存取控制规则

- 当某一用户（或某一主体）以标记label注册入系统时，系统要求他对任何客体的存取必须遵循下面两条规则：

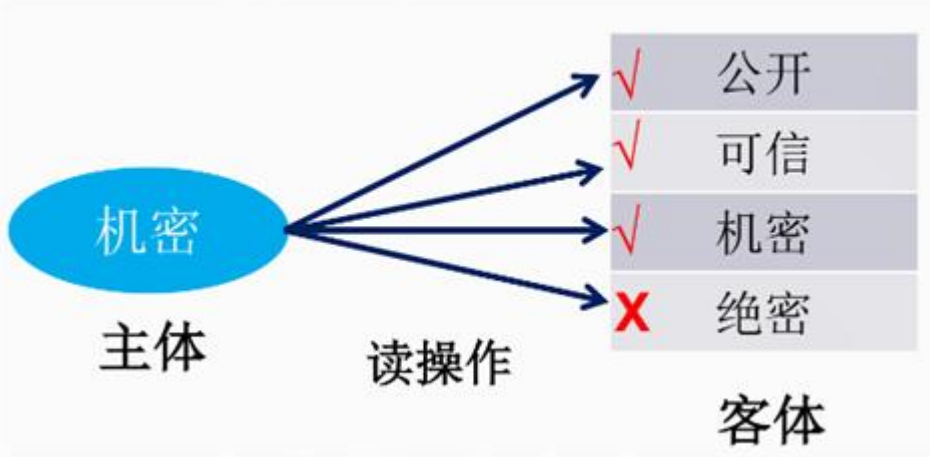
（1）仅当主体的许可证级别大于或等于客体的密级时，该主体才能读取相应的客体；

（2）仅当主体的许可证级别小于或等于客体的密级时，该主体才能写相应的客体。

强制存取控制方法

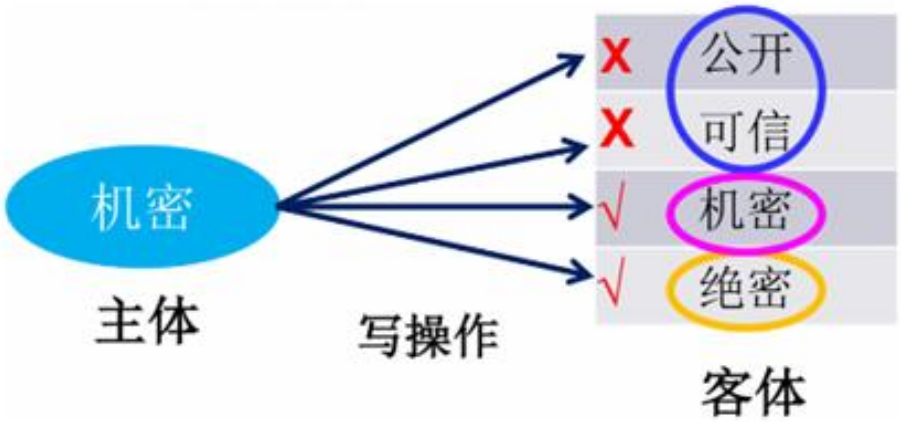
▣规则2解读：

- ◈主体的许可证级别 <= 客体的密级，主体能写客体；
- ◈用户可为写入的数据对象赋予高于自己的许可证级别的密级
- ◈一旦数据被写入，该用户自己也不能再读该数据对象了。



▣规则的共同点：

- ◈禁止拥有高许可证级别的主体更新低密级的数据对象



强制存取控制方法

强制存取控制的特点：

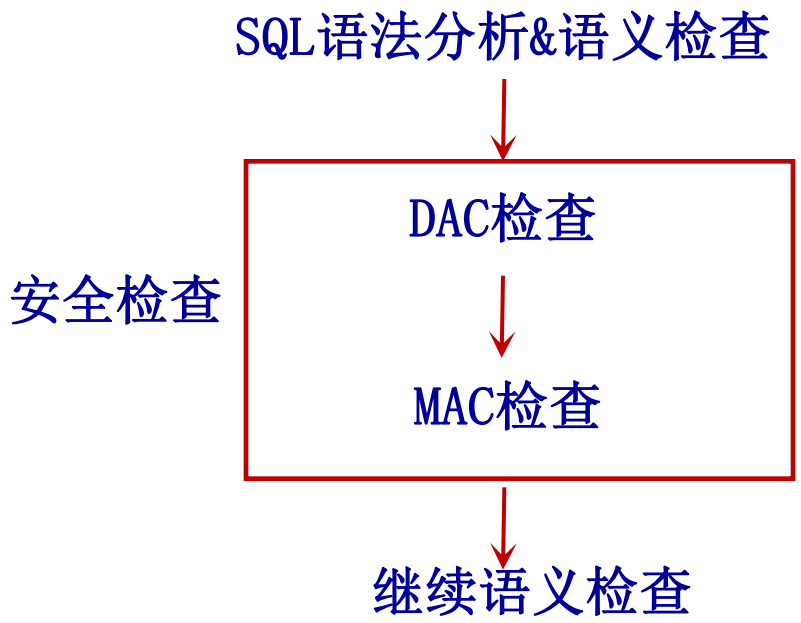
- MAC是对数据本身进行密级标记
- 无论数据如何复制，标记与数据是一个不可分的整体
- 只有符合密级标记要求的用户才可以操纵数据
- 从而提供了更高级别的安全性

MAC与DAC

- DAC与MAC共同构成DBMS的安全机制
- 实现MAC时要首先实现DAC
  - 原因：较高安全性级别提供的安全保护要包含较低级别的所有保护。

MAC与DAC

- DAC + MAC安全检查示意图：



- 先进行DAC检查，通过DAC检查的数据对象再由系统进行MAC检查，只有通过MAC检查的数据对象方可存取。

第七章 数据库安全

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

目 录

视图机制

- 视图机制把要保密的数据对无权存取这些数据的用户隐藏起来，对数据提供一定程度的安全保护。
  - 视图机制更主要的功能在于提供数据独立性，其安全保护功能太不精细，往往远不能达到应用系统的要求；

视图机制

- 间接实现了支持存取谓词的用户权限定义；

- 视图机制与授权机制配合使用：先用视图机制屏蔽掉一部分保密数据，再在视图上进一步定义存取权限。

视图机制

［例14］建立计算机系学生的视图，把对该视图的SELECT权限授于王平，把该视图上的所有操作权限授于张明。

◆先建立计算机系学生的视图CS_Student。

```
CREATE VIEW CS_Student
AS
SELECT *
FROM Student
WHERE Sdept='CS';
```

视图机制

[例14] 建立计算机系学生的视图，把对该视图的SELECT权限授于王平，把该视图上的所有操作权限授于张明。

- 在视图上进一步定义存取权限：

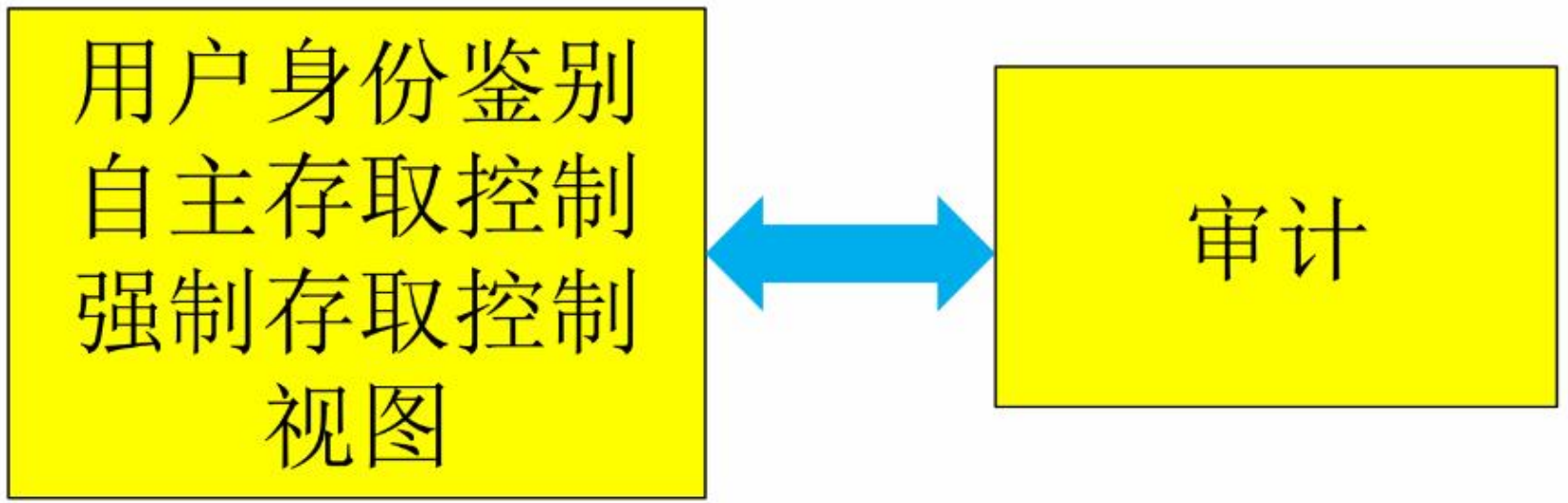
```
GRANT SELECT
ON CS_Student
TO 王平;

GRANT ALL PRIVILIGES
ON CS_Student
TO 张明;
```

第七章 数据库安全

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

目录

数据库安全性控制措施



审计

什么是审计？

- 启用一个专用的审计日志（Audit Log）

  将用户对数据库的所有操作记录在上面；

- DBA可以利用审计日志中的追踪信息

  找出非法存取数据的人、时间和内容；

- C2以上安全级别的DBMS必须具有审计功能；

可审计事件

- 服务器事件：
  - 审计数据库服务器发生的事件，包括数据库服务器的启动、停止、数据库服务器配置文件的重新加载；
- 系统权限：
  - 对系统拥有的结构或模式对象进行操作的审计，要求该操作权限是通过系统权限获得；

可审计事件

- 语句事件：
  - 对SQL语句，如DDL、DML、DQL、以及DCL语句的审计；
- 模式对象事件：
  - 对特定模式对象上进行的SELECT或DML操作的审计，模式对象包括表、视图、存储过程、函数等，模式对象不包括依附于表的索引、约束、触发器、分区表等；

审计功能

- 基本功能，提供多种审计查阅方式：基本的、可选的、有限的等等；
- 提供多套审计规则，审计规则一般在数据库初始化时设定，方便审计员管理；
- 提供审计分析和报表功能；
- 审计日志管理功能，包括为防止审计员误删除审计记录，审计日志必须先转储后删除；对转储的审计记录文件提供完整性和保密性保护；只允许审计员查阅和转储审计记录，不允许任何用户新增和修改审计记录；
- 系统提供查询审计设置及审计记录信息的专门视图；

审计的分类

◆用户级审计

- 针对自己创建的数据库表或视图进行审计。
- 记录所有用户对这些表或视图的一切成功和（或）不成功的访问要求，以及各种类型的SQL操作。

◆系统级审计

- DBA设置。
- 监测成功或失败的登录要求。
- 监测GRANT和REVOKE操作以及其他数据库级权限下的操作。

审计

AUDIT语句：设置审计功能

[例15] 对修改SC表结构或修改SC表数据的操作进行审计。

```
AUDIT ALTER, UPDATE
ON SC;
```

NOAUDIT语句：取消审计功能

[例16] 取消对SC表的一切审计。

```
NOAUDIT ALTER, UPDATE
ON SC;
```

审计

- 审计功能的可选性：
  - 审计很费时间和空间；
  - DBA可以根据应用对安全性的要求，灵活地打开或关闭审计功能。

第七章 数据库安全

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

目 录

数据库加密系统的要求

- 与通信加密相比，其信息保存时间长，不可能采用一次一密的方法进行加密；
- 实际加密后，存储空间不应明显增大；
- 加密和解密速度要快，尤其是解密速度要快，使用户感觉不到解密带来系统性能的变化；

数据库加密系统的要求

- 对数据库的加密不应影响系统原有功能，应保持对数据库操作（如查询、检索、修改、更新）的灵活性和简便性；
- 加密后的数据库仍能允许用户以不同的粒度对之进行访问；
- 灵活的密钥管理机制，加解密密钥存储安全，使用方便可靠；

数据库加密的实现机制

- 可考虑在三个不同层次实现对数据库数据的加密，这三个层次分别是OS、DBMS内核层和DBMS外层；

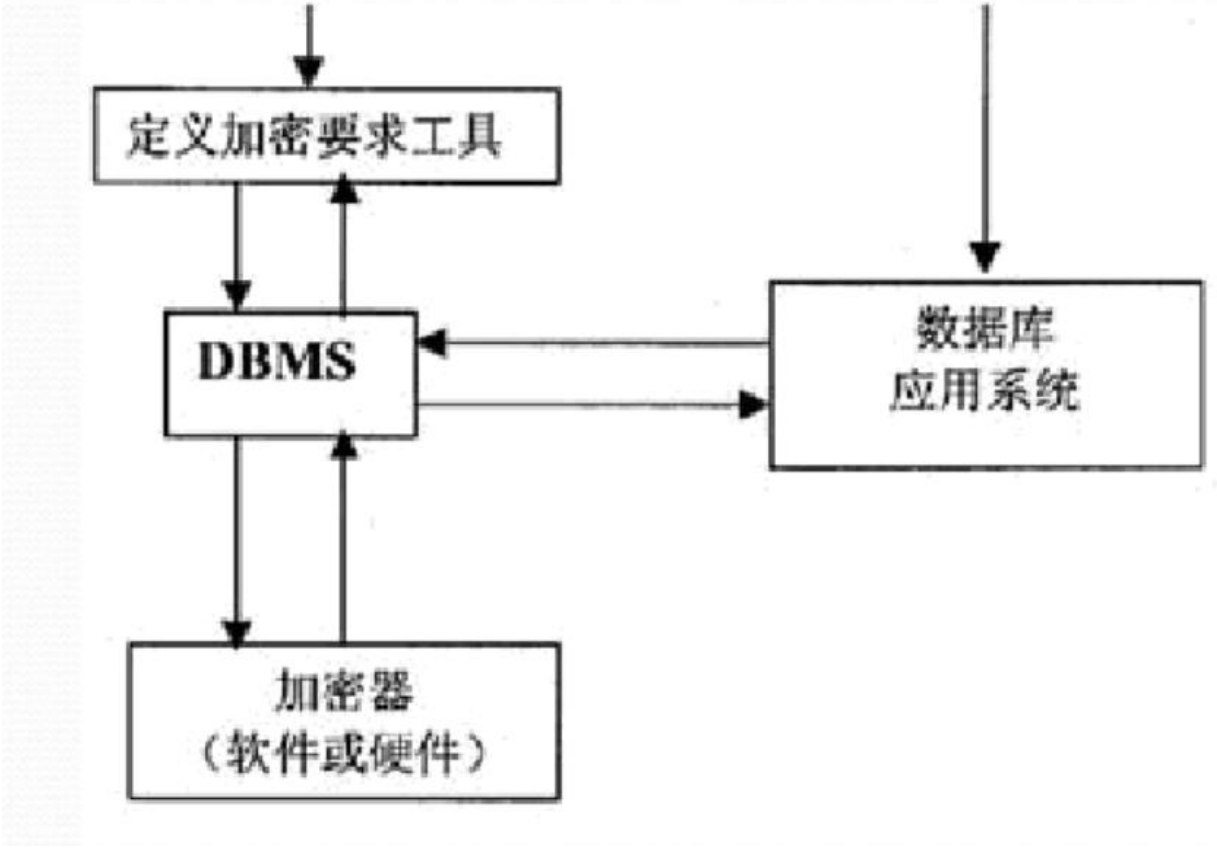
（1）OS层加密

在OS层无法辨认数据库文件中的数据关系，从而无法产生合理的密钥，对密钥合理的管理和使用也很难；大型数据库很难实现在OS层对数据文件进行加密。

数据库加密的实现机制

（2）DBMS内核层加密

在DBMS内核层实现加密，是指数据在物理存/取之前完成加/脱密工作，DBMS和加密器（硬件或软件）之间的接口需要DBMS开发商支持；



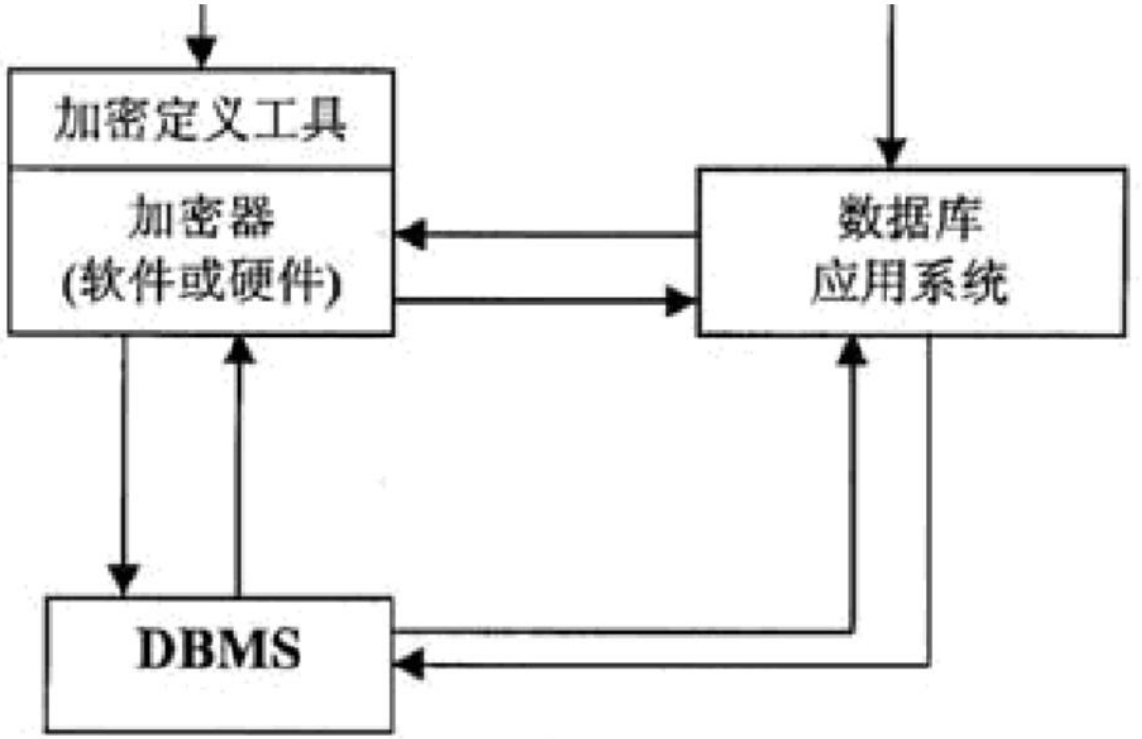
DBMS内核层加密

- 优点：
  - 加密功能强；
  - 加密功能集成为DBMS功能，实现加密与DBMS无缝耦合；
  - 对数据库应用来说，库内加密完全透明，不需任何改动直接使用；
- 缺点：
  - 对系统性能影响比较大，DBMS除了完成正常功能，还要加解密运算，加重了数据库服务器的负载；
  - 密钥管理风险大，加密密钥与数据库数据一同保存在服务器中，安全性依赖于DBMS的访问控制机制；
  - 加密功能依赖于数据库厂商支持，DBMS一般只提供有限的加密算法与强度可供选择，自主性受限；

数据库加密的实现机制

（3）DBMS外层加密

将数据库加密系统做成DBMS的一个外层工具，加解密过程发生在DBMS之外，DBMS管理的是密文。加解密过程可在客户端实现，也可由专门的加密服务器或硬件完成；



DBMS外层加密

- 优点：
  - 加解密在客户端或专门的加密服务器实现，减少了DBMS设计复杂度与运行负担
  - 加密密钥与加密数据分开保存，加密密钥保存在加密服务器中，甚至是硬件中，提高了安全性
  - 客户端与服务器配合，可以实现端到端网上密文传输；
- 缺点：
  - 加密后数据库功能受到一些限制，例如加密后的数据无法正常索引，数据加密破坏原有关系数据的完整性和一致性；

数据库加密的粒度

- 表级加密：
  - 加密对象是数据库文件，类似于操作系统文件加密的方法；
  - 数据的共享，通过用户对整个数据库文件进行解密来实现，即使用户只需要查看或修改某一记录，也需要将整个数据库文件解密，不仅增加了系统的时空开销，也无法控制用户对未授权信息的访问；
- 属性级加密：
  - 又称为“域加密”，以表中的列为单位进行加密，一般来说，属性的个数少于记录的条数，而且需要的密钥数相对较少，适合于只有少数属性需要加密的场合；

数据库加密的粒度

- 记录级加密：
  - 一般而言，数据库系统中每条记录所包含的信息具有一定的封闭性，它独立完整存储了一个实体的数据，因此基于记录的加密技术最常用，每条记录在各自密钥作用下加密成密文信息；
  - 查找记录时，可以通过将需要查找的值加密成密文后进行；
  - 缺点是在解密一个记录数据时，无法实现对在这个记录中不需要的数据项不解密；

数据库加密的粒度

- 数据项加密：
  - 数据项加密是以记录中每个字段的值为单位进行加密，数据项是数据库中最小的加密粒度；
  - 优点：系统的安全性与灵活性最高，实现技术也最为复杂，不同数据项使用不同密钥，相同明文生成不同密文，抗攻击能力得到提高；
  - 缺点：需要引入大量的密钥，一般要周密设计自动生成密钥的算法，密钥管理的复杂度大大增加，系统效率受到影响；

密码学的发展阶段

- 1949年之前

  古典密码，密码学仅为艺术

- 1949～1975年

  常规加密，密码学成为科学

- 1976年以后

  密码学的新方向——公钥密码学

经典加密技术

- 替代:明文的字母由其它字母或数字或符号代替；
  - ✓ 单一字母替代法：反映原来字母表频率，易被攻破；
  - ✓ 多字母加密密码：对明文多个字母加密，或使用多个密文字母表；
- 置换:通过执行对明文字母的置换，取得一种类型完全不同的映射；
- 转子机：通过多个加密阶段的组合，能使密码分析变得极为困难，对置换和替代都适合；

恺撒密码替换

破译以下密文：

wuhdwb lpsrvvleoh

TREATY IMPOSSIBLE

加密算法： Ci=E(Pi)=Pi+K，k在1~25取值

字母表：（密码本）

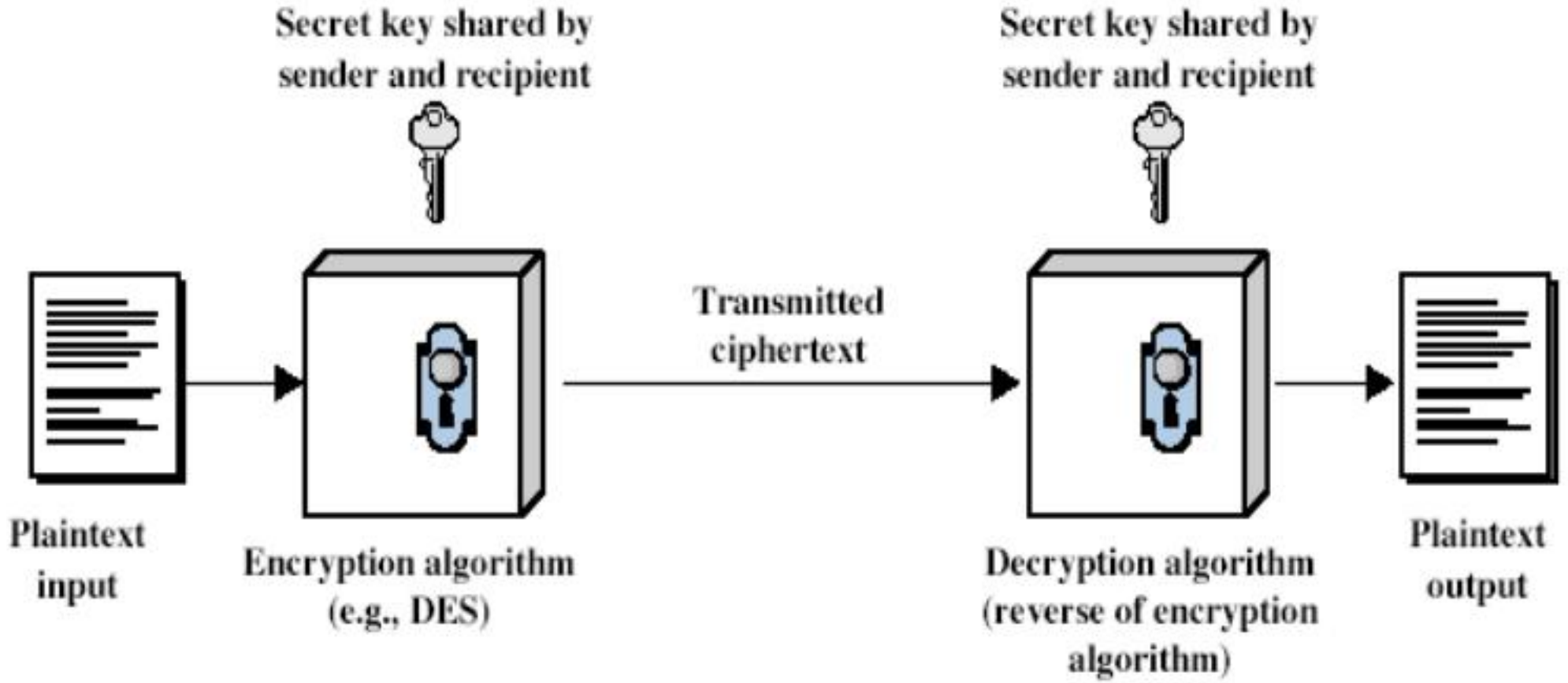
ABCDEFGHIJKLMNOPQRSTUVWXYZ
defghijklmnopqrstuvwxyzabc

置换技术

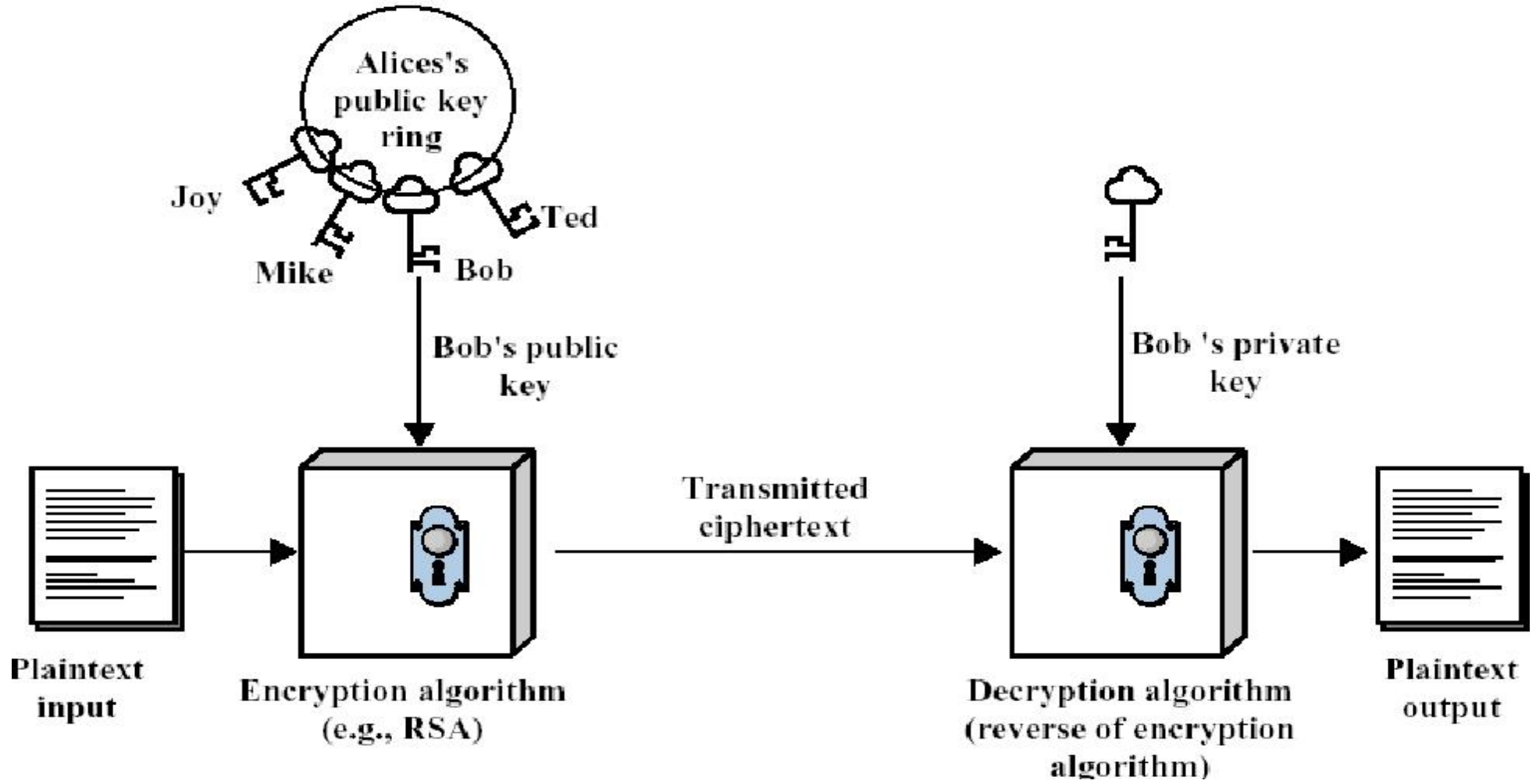
纯置换易于识别，因为它具有与原明文相同的字母频率；多次置换，可较大改观置换密码的安全性能。

密钥：	4	3	1	2	5	6	7
明文：	a	t	t	a	c	k	p
	o	s	t	p	o	n	e
	d	u	n	t	i	l	t
	w	o	a	m	x	y	z
密文：	TTNA	APTM	TSUO	AODW	COIX	KNLY	PETZ

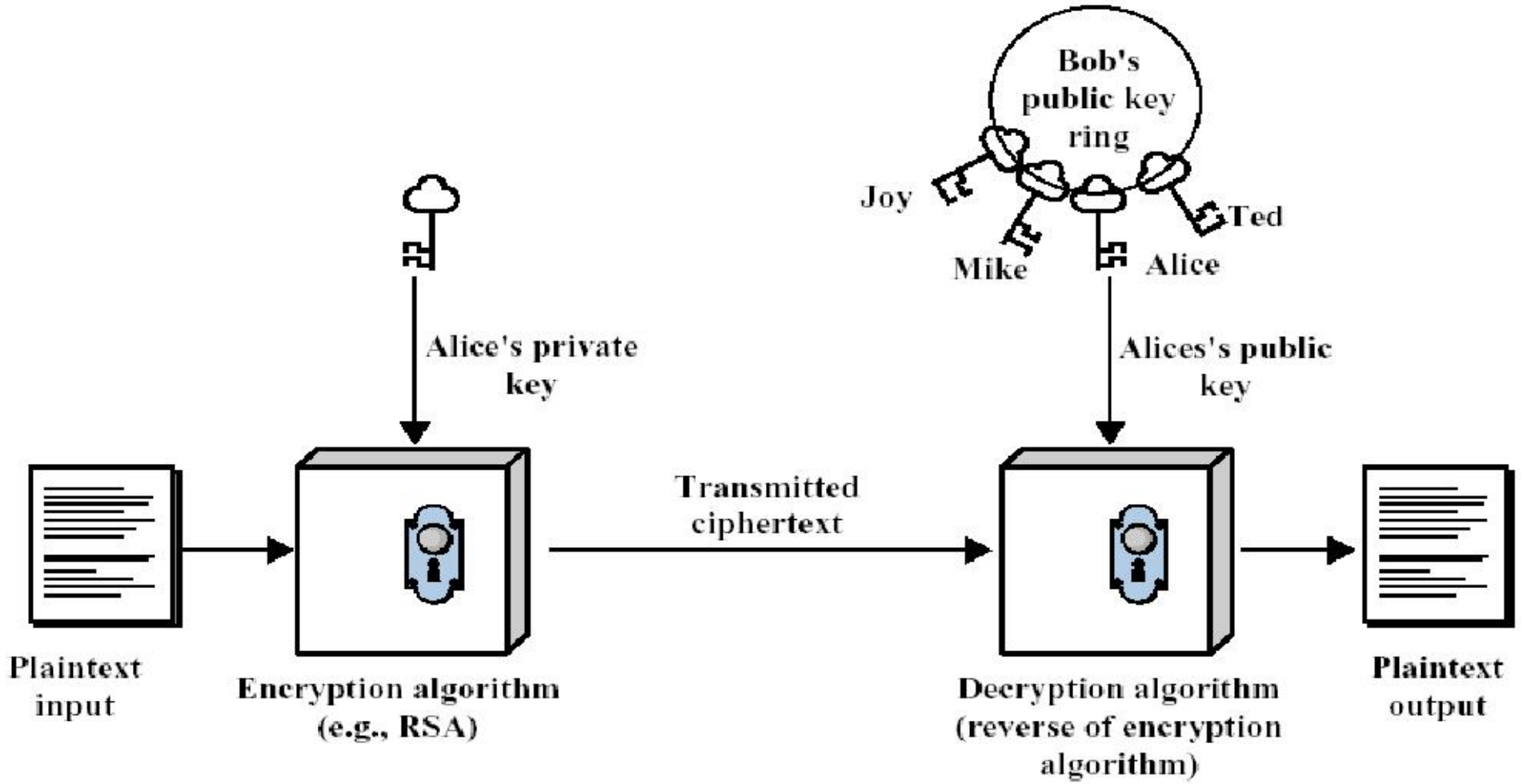
对称密码体制基本原理


Secret key shared by sender and recipient
Secret key shared by sender and recipient
Transmitted ciphertext
Plaintext input
Encryption algorithm (e.g., DES)
Decryption algorithm (reverse of encryption algorithm)
Plaintext output

基于公开密钥的加密过程



基于公开密钥的鉴别过程



数据库密钥管理

- 数据库的密钥管理，一般有集中密钥管理和多级密钥管理；
- 集中密钥管理
  - 设立密钥管理中心，负责产生密钥并对数据加密，形成一张密钥表；
  - 用户访问数据库时，密钥管理机构审核用户标识和用户密钥，并找出或计算出相应的数据密钥；
  - 便于用户使用和管理，但密钥一般由数据库管理人员控制，权限过于集中；

数据库密钥管理

多级密钥管理

- 加密粒度为数据项的三级密钥管理体制中，整个系统使用一个主密钥MK、每个表上的表密钥TK、以及各个数据项上的数据密钥等三级密钥结构；
- 表密钥被主密钥加密后，以密文的形式保存在数据字典中；
- 数据元素密钥由表密钥及数据项所在行列，通过某种函数自动生成，一般不需保存；
- 在多级密钥体制中，主密钥是加密子系统的关键，多级密钥管理体制的安全性，很大程度依赖于主密钥的安全性；

数据库加密

存储加密

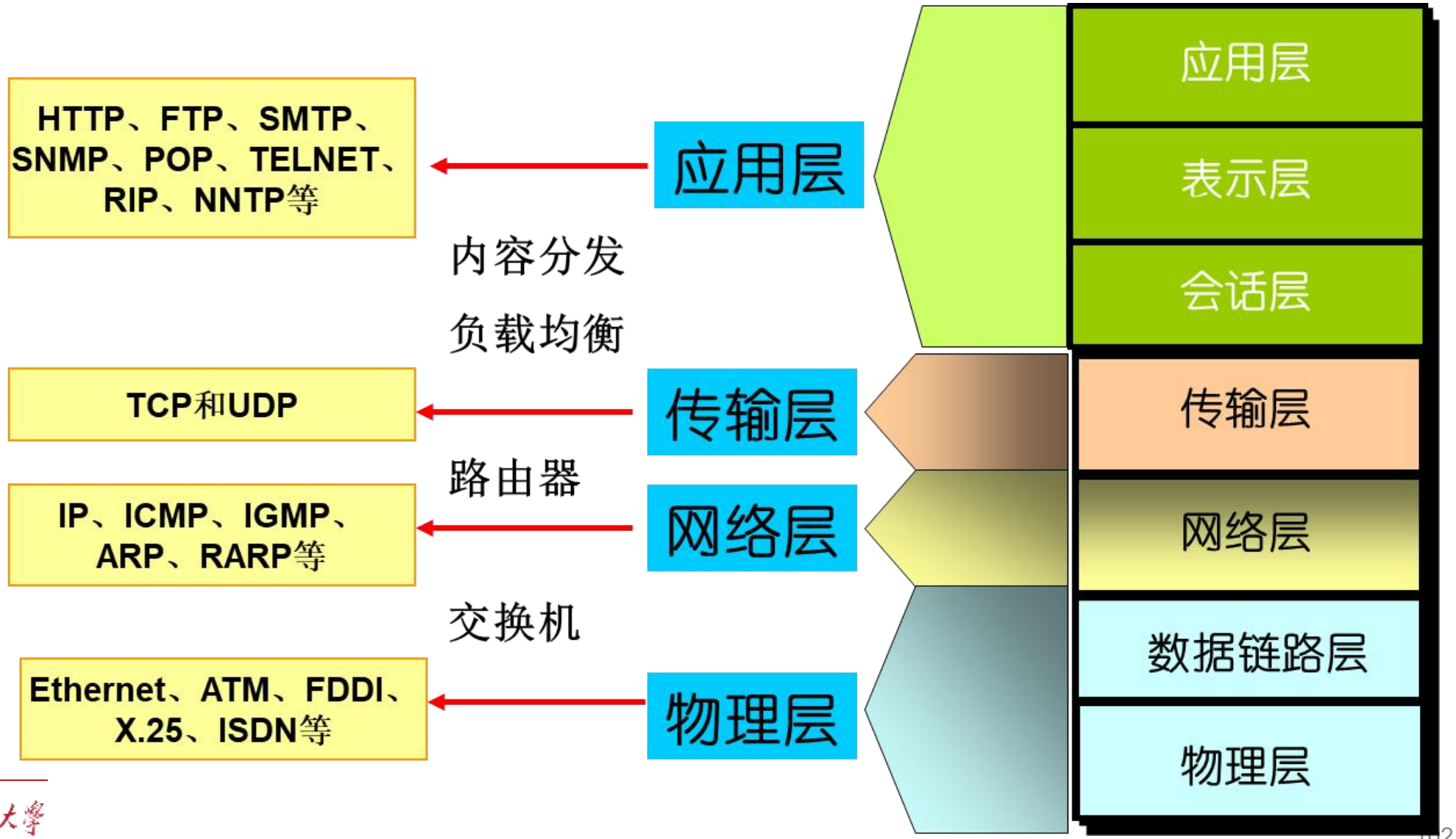
- 一般提供透明和非透明两种存储加密方式；
- 透明存储加密
  - ✓ 内核级加密保护方式，对用户完全透明；
  - ✓ 数据写到磁盘时对数据进行加密，授权用户读取数据时解密；
  - ✓ 应用程序不需要修改，只需在创建表语句中说明需加密字段；
  - ✓ 性能较好，安全完备性较高；
- 非透明存储加密，是通过多个加密函数实现；

数据库加密

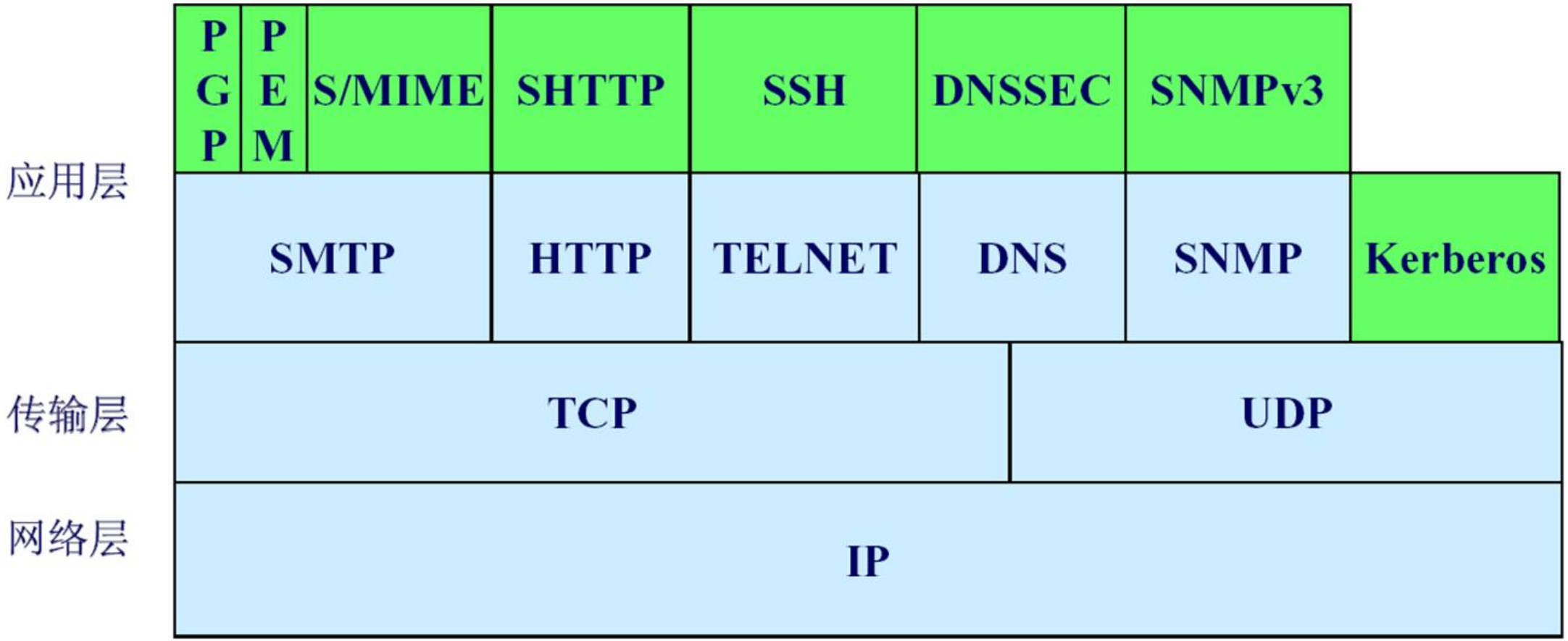
传输加密

- 帮助数据库用户和服务器之间进行安全数据交换；
- **链路加密**
  - 对传输数据在链路层加密，
  - 传输信息由报头和报文两部分组成，
  - 报头和报文均加密；
- **端到端加密**
  - 对传输数据在发送端加密，接收端解密；
  - 只加密报文，不加密报头；
  - 所需密码设备数量相对较少，但易被监听者获得敏感信息；

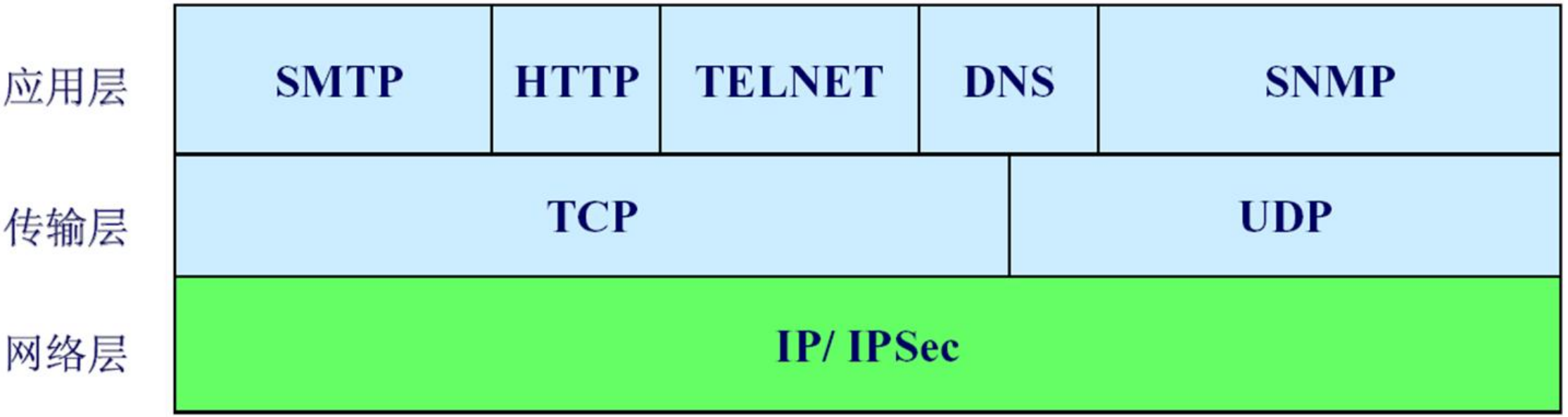
TCP/IP参考模型


HTTP、FTP、SMTP、
SNMP、POP、TELNET、
RIP、NNTP等
应用层
内容分发
负载均衡
TCP和UDP
传输层
路由器
IP、ICMP、IGMP、
ARP、RARP等
网络层
交换机
Ethernet、ATM、FDDI、
X.25、ISDN等
物理层
应用层
表示层
会话层
传输层
网络层
数据链路层
物理层

应用层安全


应用层
传输层
网络层
PGP
PEM
S/MIME
SHTTP
SSH
DNSSEC
SNMPv3
SMTP
HTTP
TELNET
DNS
SNMP
Kerberos
TCP
UDP
IP

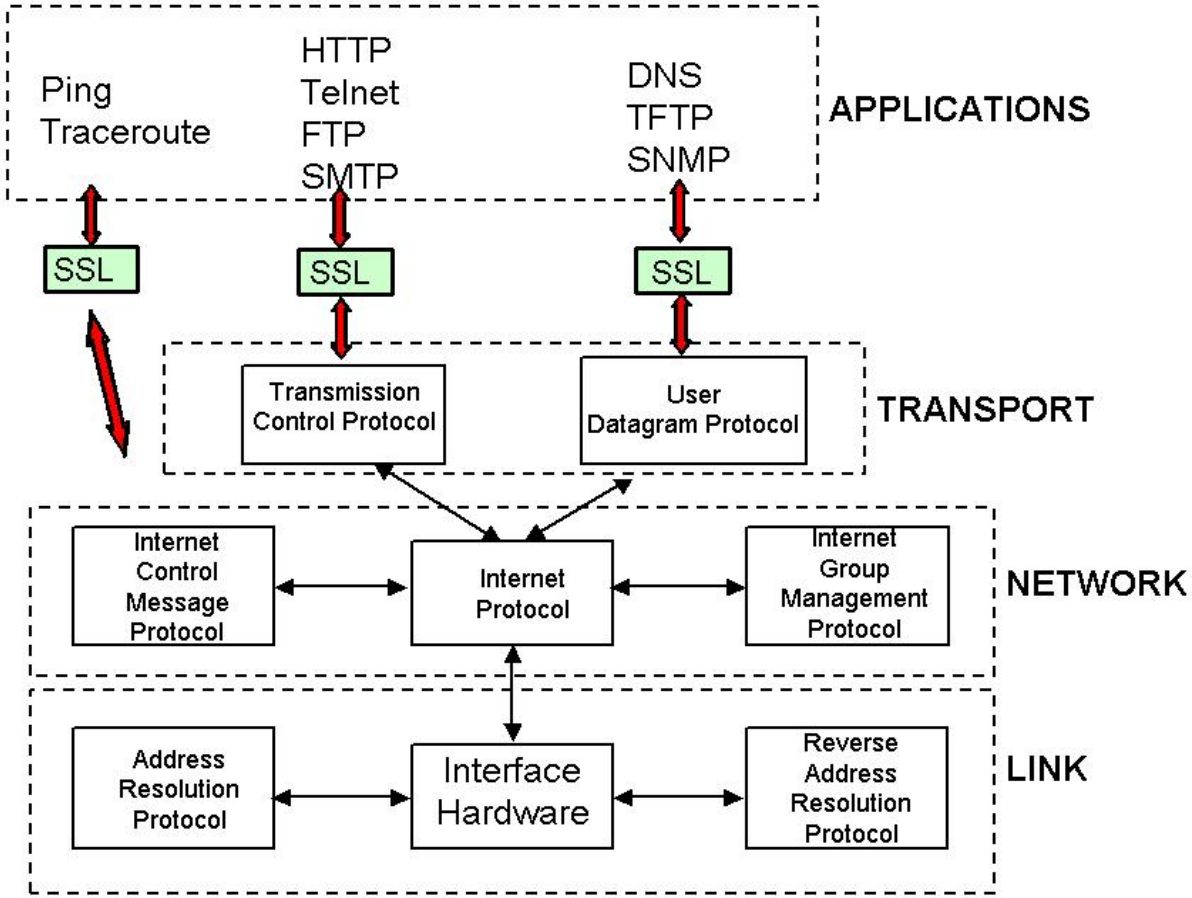
网络层安全



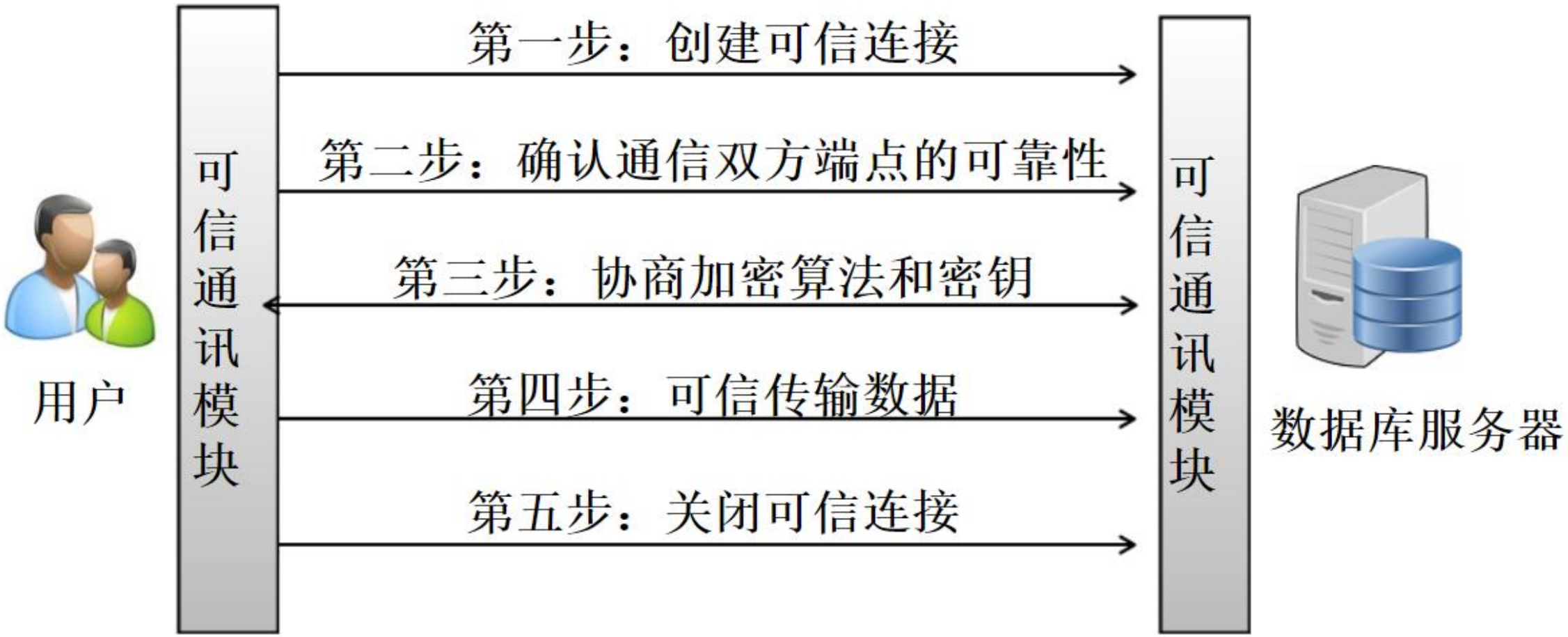
安全套接层协议SSL

- SSL（Secure Sockets Layer）被设计用来使用TCP提供一个可靠的端到端安全服务，为两个通讯个体之间提供保密性和完整性（身份鉴别）。

- 在Browser和Web Server之间提供敏感信息传输通道
  - Social Security Number (SSN)
  - Credit Card, etc
- 提供访问控制
  - Open
  - Closed



数据库管理系统可信传输示意图



SSL的实现思路

基于安全套接层协议SSL传输方案的实现思路：

（1）确认通信双方端点的可靠性

- 采用基于数字证书的服务器和客户端认证方式
- 通信时均首先向对方提供己方证书，然后使用本地的CA 信任列表和证书撤销列表对接收到的对方证书进行验证

（2）协商加密算法和密钥

- 确认双方端点的可靠性后，通信双方协商本次会话的加密算法与密钥

（3）可信数据传输

- 业务数据在被发送之前将被用某一组特定的密钥进行加密和消息摘要计算，以密文形式在网络上传输
- 当业务数据被接收的时候，需用相同一组特定的密钥进行解密和摘要计算

数据库加密总结

- 数据加密功能通常也作为可选特征，允许用户自由选择
  - 数据加密与解密是比较费时的操作
  - 数据加密与解密程序会占用大量系统资源
  - 应该只对高度机密的数据加密

第六章 数据库安全

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

目 录

一、推理控制

- 处理强制存取控制未解决的问题
- 避免用户利用能够访问的数据推知更高密级的数据
- 常用方法：
  - 基于函数依赖的推理控制
  - 基于敏感关联的推理控制

统计数据库安全性

统计数据库的特点：

- 允许用户查询聚集类型的信息（例如合计、平均值等）
- 不允许查询单个记录信息

统计数据库中特殊的安全性问题

- 隐蔽的信息通道
- 从合法的查询中推导出不合法的信息

统计数据库安全性

例1：下面两个查询都是合法的：

1．本公司共有多少女高级程序员？

2．本公司女高级程序员的工资总额是多少？

如果第一个查询的结果是“1”，

那么第二个查询的结果显然就是这个程序员的工资数。

规则1：任何查询至少要涉及N(N足够大)个以上的记录

统计数据库安全性

例2：用户A发出下面两个合法查询：

1．用户A和其他N个程序员的工资总额是多少？

2．用户B和其他N个程序员的工资总额是多少？

若第一个查询的结果是X，第二个查询的结果是Y，

由于用户A知道自己的工资是Z，

那么他可以计算出用户B的工资=Y-(X-Z)。

原因：两个查询之间有很多重复的数据项

规则2：任意两个查询的相交数据项不能超过M个

统计数据库安全性

可以证明，在上述两条规定下，如果想获知用户B的工资额

A至少需要进行1+(N-2)/M次查询

规则3：任一用户的查询次数不能超过1+(N-2)/M

注：如果两个用户合作查询就可以使这一规定失效

统计数据库安全性

数据库安全机制的设计目标：

试图破坏安全的人所花费的代价 >>

得到的利益

二、隐蔽信道

▣ 处理强制存取控制未解决的问题

例：利用未被强制存取控制SQL执行后反馈的信息进行间接信息传递

- ◈ insert语句对unique属性列写入重复值，系统会报错，操作失败
- ◈针对unique约束列，高级用户（发送者）先向该列插入数据，低级用户（接收者）向该列插入相同数据；
- ◈若插入失败，表明发送者已向该列插入数据，二者约定发送者传输信息位为0；
- ◈若插入成功，表明发送者未向该列插入数据，二者约定发送者传输信息位为1；
- ◈高级用户按事先约定方式，向低级用户传输信息，导致敏感信息泄露；

三、数据隐私

- 描述个人控制其不愿他人知道或他人不便知道的个人数据的能力；
- 研究范围很广：
  - 数据收集
  - 数据存储
  - 数据处理
  - 数据发布

数据库安全

- 数据库安全机制不是相互独立的，而是彼此依赖相互支持的；
- 用户身份认证是数据库安全的第一道防线，存取控制的正确性依赖于安全的用户身份认证机制；
- 存取控制是数据库安全最基本，也是最核心的措施；
- 数据库加密在带来更高安全性的同时，必然带来运行效率和可用性的降低，折中结果是部分敏感信息加密，需要推理控制和隐私保护手段的有效配合；
- 目前还没有可行的方法，彻底解决合法用户在通过身份认证后滥用特权的问题，审计追踪不仅是保证数据库安全的重要措施，也是任何一个安全系统不可缺少的一道防线；

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

THANK YOU !

饮水思源 爱国荣校