



经典教材《计算机操作系统》**最新版**

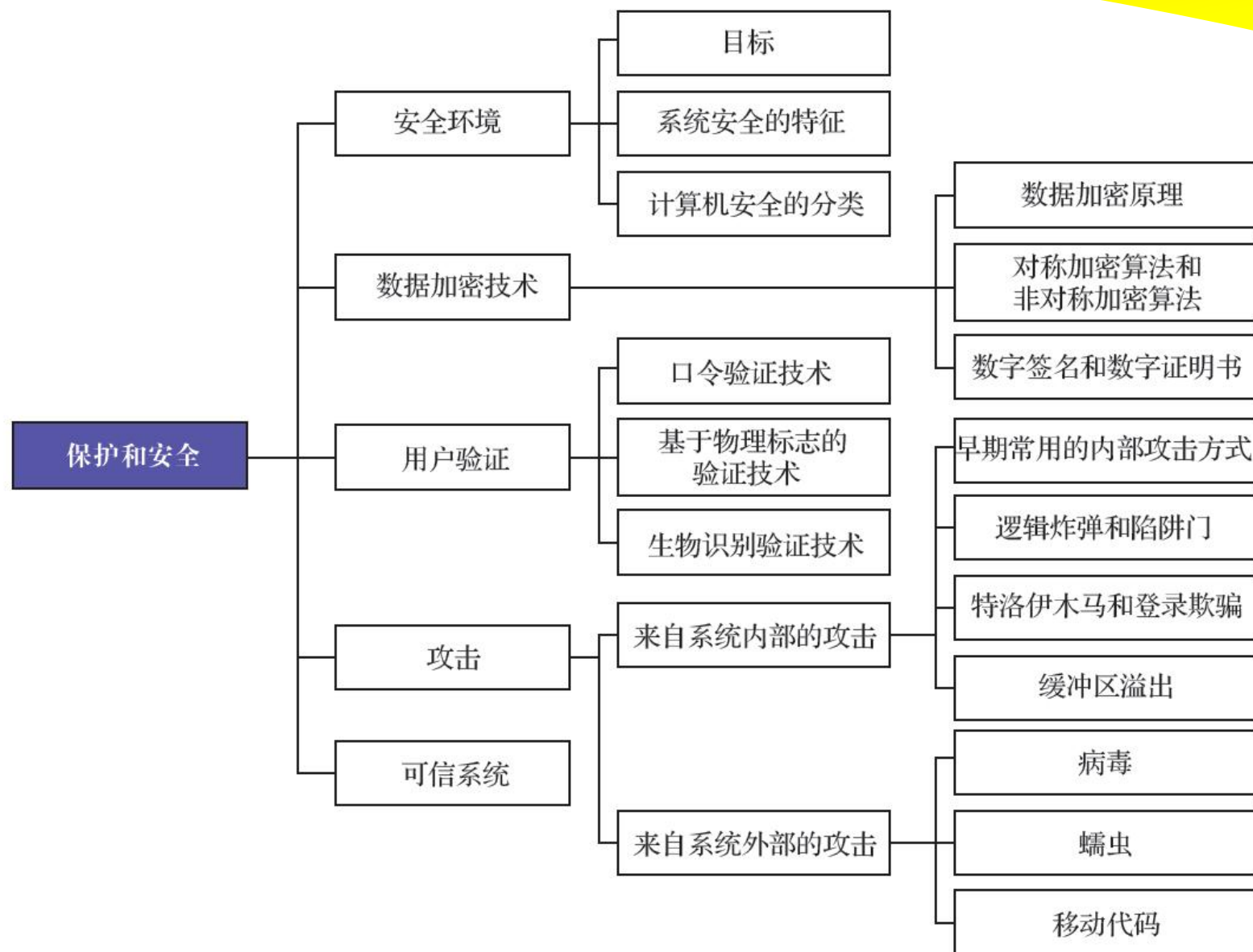
第12章 保护和安全

主讲教师：李灵慧





第1章	操作系统引论
第2章	进程的描述与控制
第3章	处理机调度与死锁
第4章	进程同步
第5章	存储器管理
第6章	虚拟存储器
第7章	输入/输出系统
第8章	文件管理
第9章	磁盘存储器管理
第10章	多处理机操作系统
第11章	虚拟化和云计算
第12章	保护和安全





内容导航:



12.1 安全环境



12.2 数据加密技术



12.3 用户验证



12.4 来自系统内部的攻击



12.5 来自系统外部的攻击



12.6 可信系统

第12章 保护和安全



保护可被定义为：

- 能够对攻击、入侵和损害系统的行为进行防御或监视的设施

安全可被定义为：

- 对系统完整性和数据安全性的可信度衡量

实现“安全环境”是目标，而保护是为了实现该目标所采取的方法和措施

12.1 实现“安全环境”的主要目标



数据机密性

- 是指将机密的数据置于保密状态，仅允许被授权用户访问系统中的信息，以避免数据暴露



数据完整性

- 是指未经授权的用户，不能擅自篡改系统中所保存的数据



系统可用性

- 是指保证计算机中的资源可供授权用户随时访问，而系统不会拒绝服务



面对的三个威胁

- 攻击者通过各种方式窃取系统中的机密信息以使其暴露
- 攻击者擅自修改系统中所保存的数据以使其被破坏（即实现数据篡改）
- 攻击者采用多种方法来扰乱系统以使其瘫痪而拒绝提供服务



信息技术安全评价通用准则，简称CC，作为了国际标准。

- 1983年，美国国防部颁布了历史上第一个计算机安全评价标准，最核心文件是TCSEC，因它是橙色封皮，简称“橙皮书”。



计算机安全的分类：

- TCSEC将计算机系统的安全程度划分为4类：D、C、B、A。
- 共分为7个等级：D、C₁、C₂、B₁、B₂、B₃、A₁。
 - ① D类，最低安全类别，又称为安全保护欠缺级。凡是无法达到另外3类标准要求的，都被归为D类。MS-DOS属于D类。
 - ② C₁级。C类是仅高于D类的安全类别。C类分为两级：C₁和C₂。C₁级要求OS使用保护模式和用户登录验证，并赋予用户自主访问控制权，即允许用户指定其他用户对自己文件的使用权限。大部分的UNIX系统属于C₁级。

- ③ C_2 级，称为受控存取控制级。它是在 C_1 级的基础上，增加了一个个体层访问控制。当前广泛使用的安全软件大多属于 C_2 级。
- ④ B_1 级，具有 C_2 级的全部安全属性。在B类系统中，会为每个可控用户和对象贴上一个安全标注。
- ⑤ B_2 级，具有 B_1 级的全部安全属性。 B_2 级要求系统必须采用自上而下的结构化设计方法，并能够对设计方法进行检验，对可能存在的隐蔽信道进行安全分析。
- ⑥ B_3 级，具有 B_2 级的全部安全属性。在 B_3 级系统中必须包含用户和组的访问控制表、足够的安全审计和灾难恢复能力。
- ⑦ A_1 级，要求系统具有强制存取控制和形式化模型技术的应用，能证明模型是正确的，并须说明有关实现方法是与保护模型一致的。



内容导航:



12.1 安全环境



12.2 数据加密技术



12.3 用户验证



12.4 来自系统内部的攻击



12.5 来自系统外部的攻击

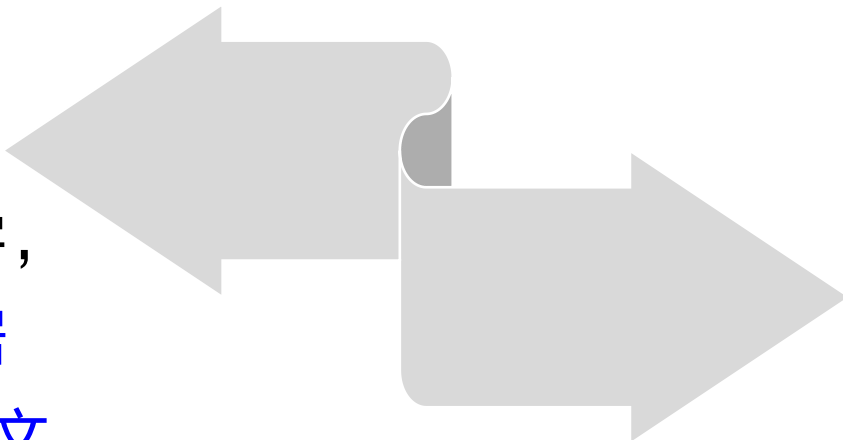


12.6 可信系统

第12章 保护和安全



加密是一种密写科学，
用于把系统中的数据
(称为明文)转换为密文。
使攻击者即使截获到被
加密的数据，也无法了
解数据的内容，从而有
效地保护了系统中信息
的安全性



数据加密技术包括：

- 数据加密
- 数据解密
- 数字签名
- 签名识别
- 数字证明
- ...

数据加密模型由4部分组成



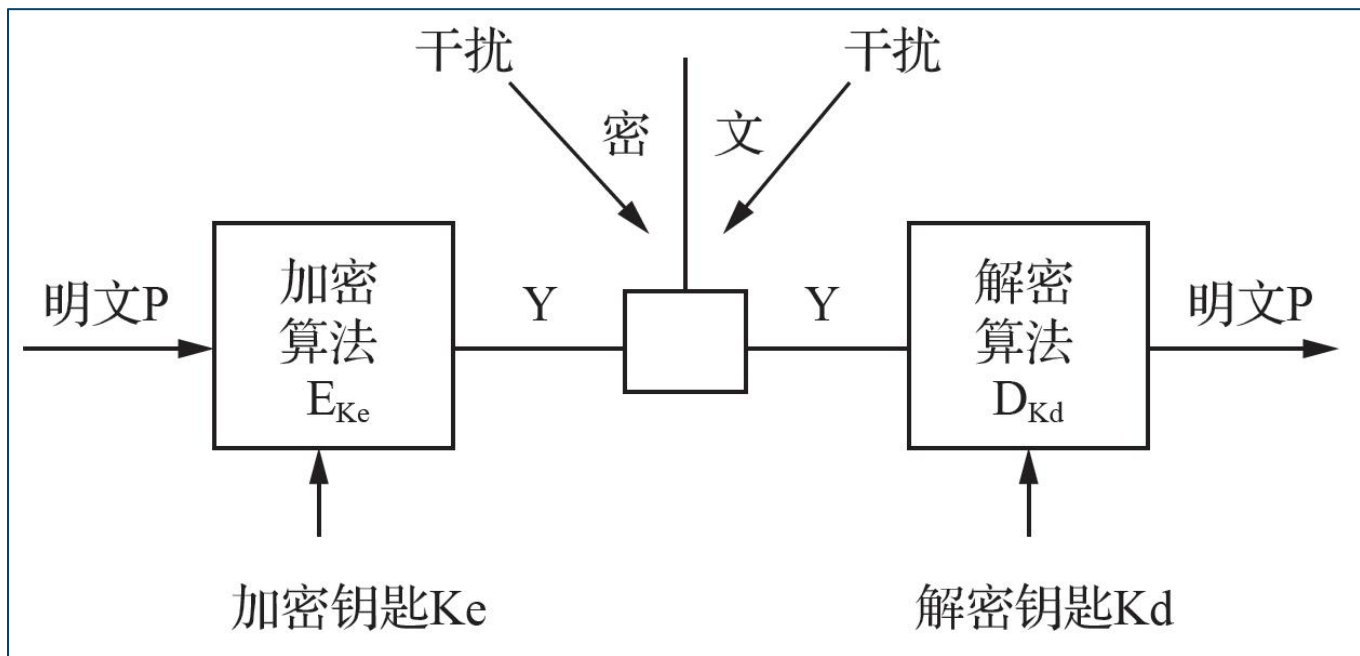
① **明文**：被加密的文本，称为明文P。


① **密文**：加密后的文本，称为密文Y。


① **加密（解密）算法**
EKe (DKd)：用于实现从明文（密文）到密文（明文）转换的公式、规则或程序。

① **密钥K**：加密和解密算法中的关键参数。

- ❑ 设计密码技术称为密码编码
- ❑ 将破译密码技术称为密码分析
- ❑ 密码编码和密码分析合起来称为**密码学**



 **易位法**是指按照一定的规则，重新安排明文中的比特或字符的顺序以形成密文，而字符本身却保持不变。

 按易位单位的不同，易位法又可分为**比特易位**和**字符易位**两种

- 比特易位法简单易行，并可用硬件实现，主要用于数字通信中
- 字符易位法利用密钥对明文进行易位后形成密文

M	E	G	A	B	U	C	K
7	4	7	1	2	8	3	6
p	l	e	a	s	e	t	r
a	n	s	f	e	r	o	n
e	m	i	l	l	i	o	n
d	o	l	l	a	r	s	t
o	m	y	s	w	i	s	s
b	a	n	k	a	c	c	o
u	n	t	s	i	x	t	w
o	t	w	o	a	b	c	d

明文

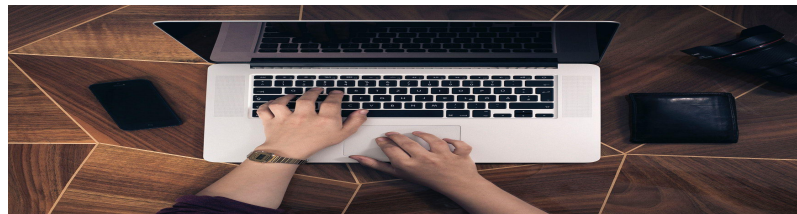
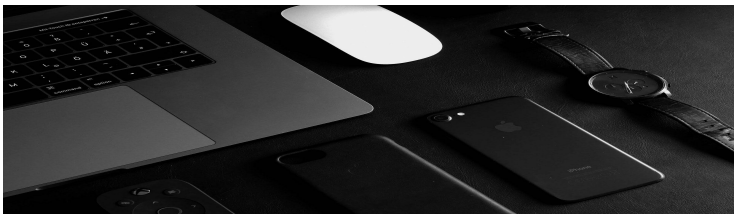
Please transfer one
million dollars to my
Swiss Bank account six
two two ...

密文

AFLLSKSOSELAWAIA
TOOSSCTCLNMOMANT
ESIL YNTWRNNTSOWD
FAEDOBNO ...



基本加密方法 – 置换法



置换法是指按照一定的规则，
用一个字符去置换（替代）另
一个字符以形成密文

这种密码很容易被破译

例如，将26个英文字母通过密钥
QWERTYUIOPASDFGHJKLZX
CVBNM 映像到另外26个特定字
母中，利用置换法和密钥，可将
attack加密而使其变为QZZQEA，

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Q	W	E	R	T	Y	U	I	O	P	A	S	D	F	G	H	J	K	L	Z	X	C	V	B	N	M



对称加密算法的加密算法和解密算法之间，存在着一定的相依关系

- 加密算法和解密算法往往使用相同的密钥；
- 或者知道了加密密钥 K_e 后，就很容易推导出解密密钥 K_d
- 最具代表性的算法：数据加密标准DES、DEP



非对称加密算法，也称公开密钥算法

- 加密密钥 K_e 和解密密钥 K_d 不同，而且难以由 K_e 推导出 K_d
- 可将其中的一个密钥公开而成为公开密钥

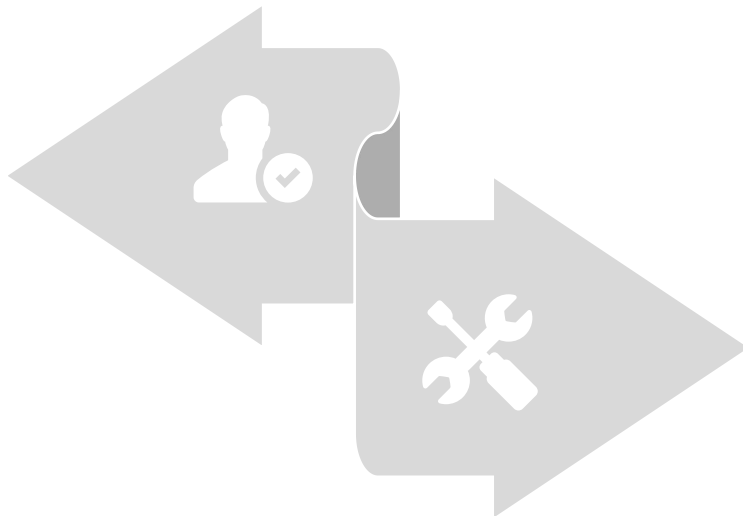


两者**区别**：

- 非对称加密算法要比对称加密算法处理速度慢，但密钥管理简单
- 当前通常同时应用这两种加密算法
 - ❑ 利用公开密钥技术传递对称密码，而利用对称密钥技术来对实际传输的数据进行加密和解密

数字签名

在利用计算机网络传送报文时，可将公开密钥法用于电子（数字）签名，以代替传统的手写签名



需要满足3个条件：

- 接收者能够核实发送者对报文的签名
- 发送者事后不能抵赖其对报文的签名
- 接收者无法伪造对报文的签名



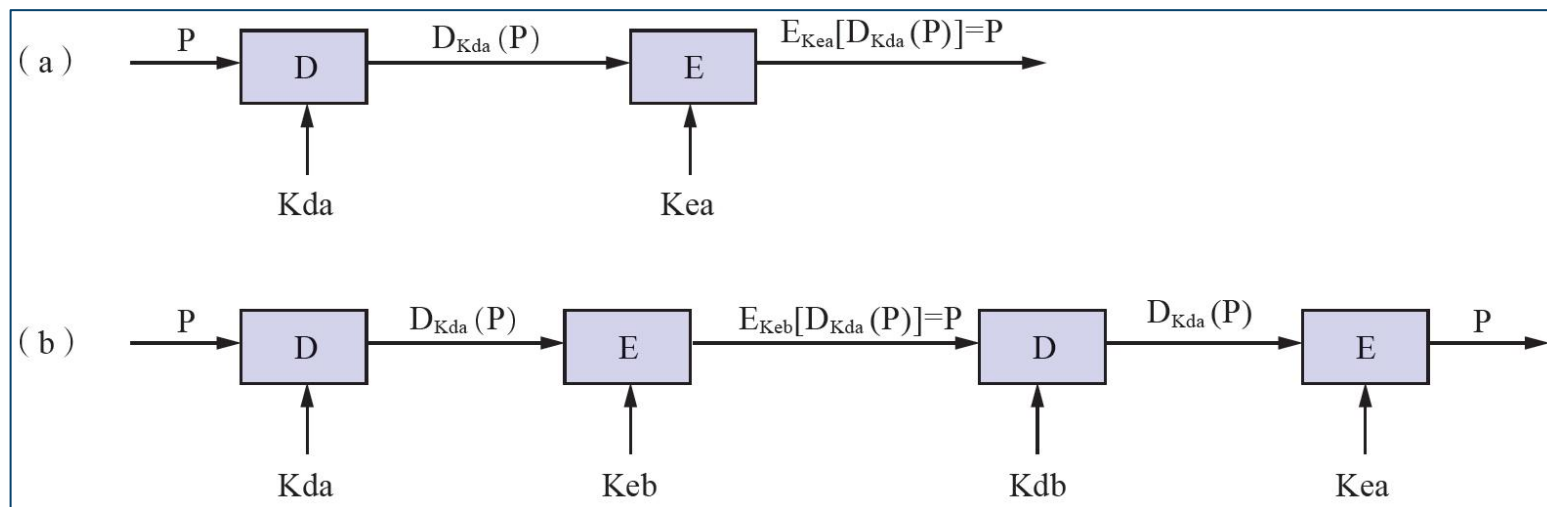
简单数字签名(a)

- 发送者A可使用私用密钥 K_{da} 对明文 P 进行加密，形成 $D_{K_{da}}(P)$ ，然后将其传送给接收者B。
- 接收者B可利用A的公开密钥 K_{ea} 对 $D_{K_{da}}(P)$ 进行解密，得到 $E_{K_{ea}}[D_{K_{da}}(P)] = P$



保密数字签名(b)

- 为了实现在发送者A和接收者B之间的保密数字签名，要求A和B都具有密钥，再加密和解密





由大家都信得过的认证机构CA为公开密钥发放一份公开密钥证明书，该公开密钥证明书又称为**数字证明书**，用于证明通信请求者的身份。

国际电信联盟（ITU）制定的X.509标准中，规定了数字证明书的内容

- 用户名称、发证机构名称、公开密钥、公开密钥的有效日期、数字证明书的编号以及发证者的签名。



内容导航:



12.1 安全环境



12.2 数据加密技术



12.3 用户验证



12.4 来自系统内部的攻击



12.5 来自系统外部的攻击



12.6 可信系统

第12章 保护和安全



验证又称为识别或认证



用户验证：当用户要登录一台多用户计算机时，OS将对该用户进行验证，目的在于确定被验证对象（包括人和事）是否真实

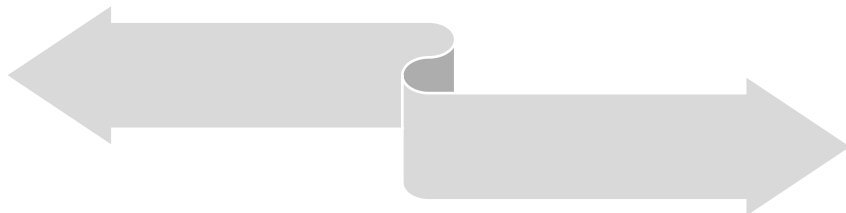


通常将验证技术的应用作为保障网络安全的第一道防线



目前验证主要依据3方面的信息来确定被验证对象的身份

- ① 所知，基于用户所知道的信息，如系统的登录名、口令等
- ② 所有基于用户所具有的东西，如身份证、信用卡等。
- ③ 用户特征，基于用户所具有的特征，特别是生理特征，如指纹、声纹、DNA等



口令

- 登录程序通过输入的用户名和口令，验证用户的合法性
- 口令由字母、数字和特殊符号混合组成，它可由系统自动生成，也可由用户自己选定
 - 系统所产生的口令往往不便于记忆
 - 用户自己规定的口令通常容易记忆，但很容易被攻击者猜中

提高口令安全性的方法：

- 口令应适当长
- 口令应包含多种字符
- 口令机制中应具有自动断开连接功能
- 系统不应将口令回送
- 系统应记录和报告用户登录情况

■ 一次性口令：口令被使用一次后就换另一个口令

- 用户给系统提供一张口令表，其中记录有其使用的口令序列
- 系统为该表设置一指针，用于指示下次用户登录时所应使用的口令

■ 口令文件：保存合法用户口令和用户特权的文件

- 保证口令文件安全性最有效的方法是利用加密技术
- 应该妥善保管好已加密的口令文件

■ 挑战-响应验证：

- 由用户自己选择一个算法，并将该算法告知服务器。
- 每当用户登录时，服务器就给用户发来一个随机数，用户收到随机数后，按所选算法对其进行运算，并将结果作为口令
- 服务器再将所收到的口令与自己（利用 算法）计算的结果进行比较，若两者相同，则允许用户登录，否则拒绝用户登录



基于磁卡的验证技术

- 将磁卡上的数据读入计算机，验证该用户是否合法
- 可增设了口令机制，即磁卡验证时，要求用户输入口令

基于IC卡的验证技术

- IC卡类型：存储器卡、微处理器卡、密码卡
- 可采用不同的验证机制，如挑战-验证机制

- 利用人体具有的、不可模仿的、难以伪造的特定生物标志来进行验证，因此具有很高的可靠性
- 常用于身份识别的生理标志



被选用的生理标志应具有3个条件

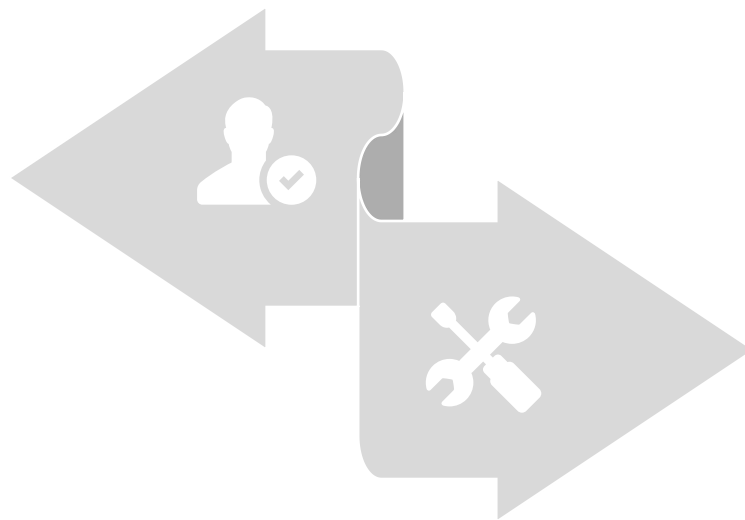
- 足够的可变性
- 应保持稳定，不会经常发生变化
- 不易被伪装



● 指纹 ● 眼纹 ● 声音 ● 人脸

对生物识别系统的要求：

- 性能需求：应具有具有很强的抗欺骗和防伪造能力。
- 易于被用户接受
- 成本合理



生物识别系统的组成：

- 生物特征采集器
- 注册部分
- 识别部分



指纹识别系统逐步小型化，该技术已进入广泛应用阶段。



指纹传感器：实现指纹图像采集的硬件。







- 是指纹识别系统的重要组成部分；
 - 指纹图像采集质量的好坏，直接影响所形成的指纹图像的质量；
 - 光学式和压感式指纹传感器应用较广。
-



指纹识别系统，包括了嵌入式指纹识别系统。



内容导航:

-  12.1 安全环境
-  12.2 数据加密技术
-  12.3 用户验证
-  **12.4 来自系统内部的攻击**
-  12.5 来自系统外部的攻击
-  12.6 可信系统

第12章 保护和安全



系统攻击

- **内部攻击**：攻击来自系统的内部
- **外部攻击**：攻击来自系统的外部



内部攻击

- 以合法用户身份直接进行攻击
 - ❑ 攻击者获得或假冒合法用户身份，再利用合法用户所拥有的权限来读取、修改、删除系统中的文件，或对系统中的其他资源进行破坏
- 通过代理功能进行间接攻击
 - ❑ 攻击者将一个代理程序置入被攻击系统的一个应用程序中，当应用程序执行并调用到代理程序时，它就会执行攻击者预先设计的破坏任务





恶意软件：攻击者专门编制的一种程序，用来对系统进行破坏。

■ 独立运行类

- 这一类恶意软件（如蠕虫、僵尸等）可以通过OS调度执行

■ 寄生类

- 这一类恶意软件本身不能独立运行，经常需要寄生在某个应用程序中
- 如逻辑炸弹、特洛伊木马、病毒等

恶意软件是一种极具破坏性的软件，但它不能进行自我复制，也不会感染其他程序。



逻辑炸弹实例

- 程序员为了应对自己可能被突然解雇，OS中预先被放入破坏程序，只要程序员每天输入口令，该程序就不会发作。如果突然被解雇，在第二天（或第二周）由于OS得不到口令，逻辑炸弹就会被引爆——执行一段带有破坏性的程序

逻辑炸弹“爆炸”的条件

- 时间引发
- 事件引发
- 计数器引发



陷阱门概念

- 陷阱门是一段代码，同时也是进入一个程序的隐蔽入口点。有此陷阱门，程序员可以不经过程序的安全检查，即可对程序进行访问
- 即程序员通过陷阱门可跳过正常的验证过程



陷阱门实例：

- 使用登录名为“**zzzzz**”时，无论用什么口令，都能成功登录上机

```
while(TRUE)
{
    printf( " login: " );
    get_string (name);
    disable_echoing ();
    printf ( " password: " );
    get_string (password);
    enable_echoing ();
    v=check_validity (name, password);
    if (v) break ;
}
execute_shell (name);
```

```
while(TRUE)
{
    printf( " login: " );
    get_string (name);
    disable_echoing ();
    printf ( " password: " );
    get_string (password);
    enable_echoing ();
    v=check_validity (name, password);
    if (v||strcmp (name, " zzzzz " )=0) break ;
}
execute_shell (name);
```



特洛伊木马的基本概念

- 特洛伊木马是一种恶意软件，它是一个嵌入到有用程序中的、隐蔽的、危害安全的程序。当该程序执行时会引发隐蔽代码执行，产生难以预期的后果



特洛伊木马的实例

- 编写特洛伊木马程序的人，将其隐藏在一个新游戏程序中，并将该游戏程序送给某计算机系统的系统操作员
- 操作员在玩新游戏程序时，前台确实是在玩游戏，但隐藏在后台运行的特洛伊木马程序，却将系统中的口令文件复制到了该黑客的文件中
- 由于操作员在打游戏时系统是在高特权模式下运行的，此时，隐藏在游戏程序中的特洛伊木马就继承了高特权，因此能够访问口令文件

以UNIX系统为例来说明登录欺骗



攻击者编写欺骗登录程序，该程序同样会在屏幕上显示“Login:”，用于欺骗其他用户进行登录。



当有一用户输入登录名后，欺骗登录程序会接着要求它输入口令。当该用户将口令输入完毕后，欺骗登录程序就会把刚输入的登录名和口令写入一份事先准备好的文件中，并发出信号以请求结束shell程序，于是欺骗登录程序退出登录，同时也去触发真正的登录程序，并在屏幕上再次显示出“Login:”



用户以为是自己输入发生了错误，系统要求重新输入



C语言编译器**存在着某些漏洞**，如它对数组不进行边界检查。

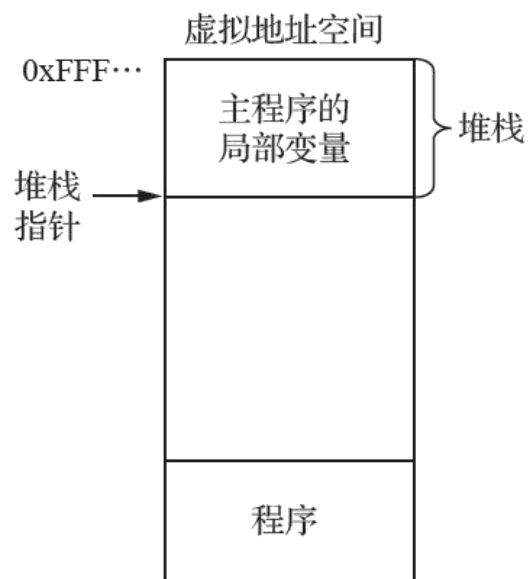


下面的代码是不合法的，数组范围是1024，其所包含的数字却有12000个，而且在编译时未对此事进行检查。

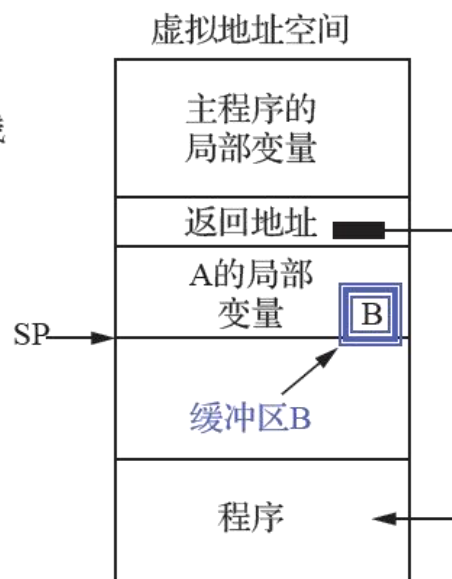


攻击者可能会利用
此漏洞进行攻击。

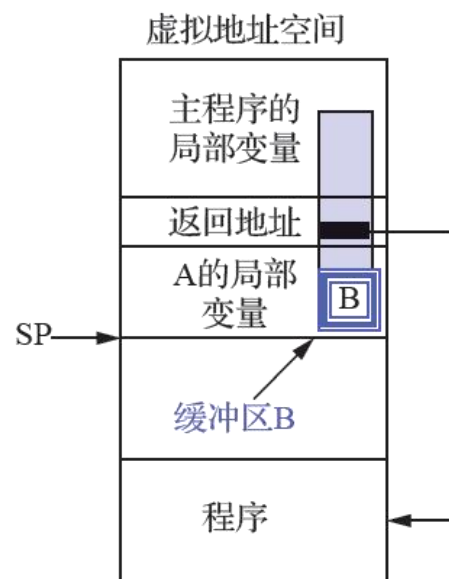
```
int i;  
char C[1024];  
i=12000;  
C[i]=0;
```



(a) 最初的情况









(b) 调用A后的情况



(c) 缓冲区溢出后的情况



内容导航:

-  12.1 安全环境
-  12.2 数据加密技术
-  12.3 用户验证
-  12.4 来自系统内部的攻击
-  **12.5 来自系统外部的攻击**
-  12.6 可信系统

第12章 保护和安全



病毒

- 计算机中的病毒是一段程序，它能把自己附加在其他程序之中，并不断地自我复制，然后去感染其他程序，进而借助被感染的程序和系统传播出去。



蠕虫

- 蠕虫与病毒相似，也能进行自我复制，并可传染给其他程序，给系统带来有害影响，它们都属于恶意软件
 - 与病毒的区别
 - 蠕虫本身是一个完整的程序，能作为一个独立的进程运行，因而它不需要寄生在其他程序上
 - 蠕虫的传播性没有病毒的强
 - 蠕虫由两部分组成，即引导程序和蠕虫本身



什么是移动代码？

- 如果一个程序在运行时，能在不同机器之间来回迁移，那么该程序就被称为移动代码

移动代码能否安全运行？

- 移动代码将占用进程的内存空间，并作为合法用户的一部分运行，同时拥有用户的访问权限，因此不能保证系统安全，需要防范移动代码



防范移动代码的方法——沙盒法

- 基本思想是采用隔离方法，具体做法是把虚拟地址空间分为若干个大小相同的区域，每个区域称为一个沙盒
- 将不可信程序放入一个沙盒中运行，如果发现沙盒中的程序要访问沙盒外的数据，或者有跳转到沙盒外某个地址去运行的任何企图，则系统将停止该程序的运行



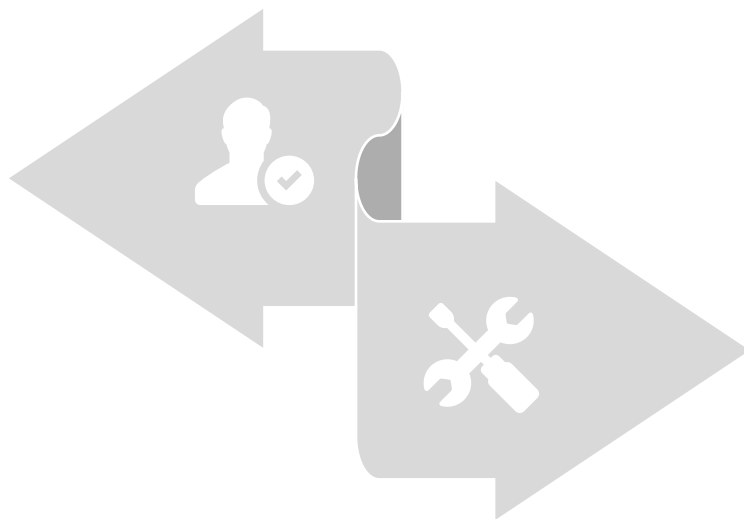
防范移动代码的方法——解释法

- 解释法的基本思想是对移动代码的运行采取解释执行方式
- 若移动代码是可信的（来自本地硬盘），则按正常情况处理；否则（如来自互联网），将其放入沙盒中以限制其运行



计算机病毒的特征

- (1) 寄生性
- (2) 传染性
- (3) 隐蔽性
- (4) 破坏性



计算机病毒的类型

- 文件性病毒
- 内存驻留病毒
- 引导扇区病毒
- 宏病毒
- 电子邮件病毒

□ 伪装

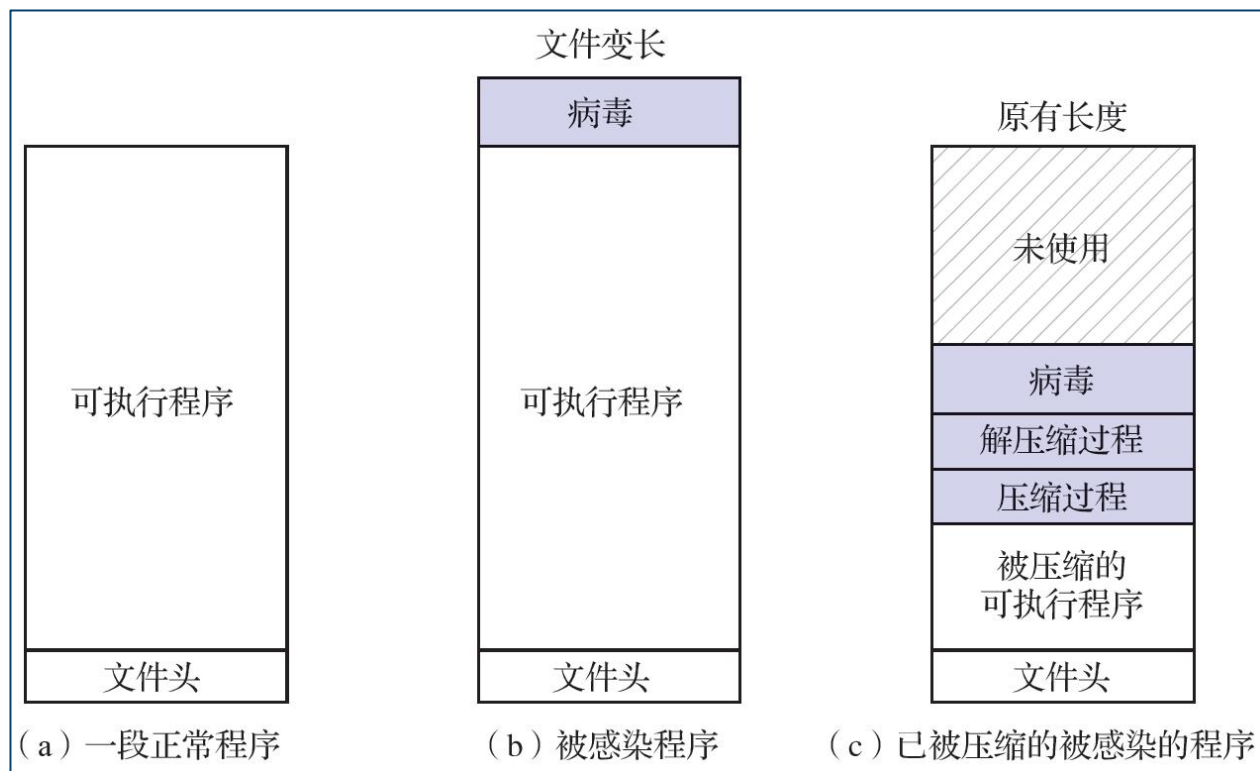
- ① 通过压缩法伪装
- ② 通过修改日期和时间伪装

□ 隐藏

- ① 隐藏于目录和注册表空间
- ② 隐藏于程序的页内零头里
- ③ 更改用于磁盘分配的数据结构
- ④ 更改坏扇区列表

□ 多形态

- ① 插入多余的指令
- ② 对病毒程序进行加密



病毒的预防

- ① 对重要的软件和数据，应当定期将它们备份到外部存储介质上
- ② 使用具有高安全性的OS
- ③ 使用正版软件
- ④ 购买性能优良的反病毒软件
- ⑤ 对于来历不明的电子邮件，不要将其轻易打开
- ⑥ 要定期检查硬盘及U盘，用反病毒软件来清除其中的病毒



基于病毒数据库的检测方法：

- 建立病毒数据库；
- 扫描硬盘上的可执行文件。









完整性检测方法：

- 完整性检测程序在检测病毒之前，首先会扫描硬盘，检查其中是否有病毒，当确信硬盘“干净”时，才正式工作；
- 首先计算每个文件的检查和，然后计算目录中所有相关文件的检查和，将所有检查和写入一个检查和文件中；
- 在检测病毒时，完整性检测程序将重新计算所有文件的检查和，并将它们分别与原文件的检查和进行比较，若不匹配，则表明该文件已感染上病毒。



内容导航:

-  12.1 安全环境
-  12.2 数据加密技术
-  12.3 用户验证
-  12.4 来自系统内部的攻击
-  12.5 来自系统外部的攻击
-  **12.6 可信系统**

第12章 保护和安全

- 要建立一个可信系统，应在OS核心中构建一个安全模型，模型要非常简单以确保自身的安全性
- 安全策略：
 - 根据系统对安全的需求所定义的一组规则以及相应的描述
- 安全模型：用于精确描述系统的安全需求和策略
 - 安全模型首先应当是精确的、同时它也应当是简单和容易理解的，而且不涉及安全功能的具体实现细节
 - 比较实用的模型：访问矩阵模型、信息流控制模型
- 访问矩阵模型，也称保护矩阵
 - 系统中的每个主体（用户）都拥有矩阵中的一行
 - 矩阵中的交叉项表示某主体对某客体的存取权限集
 - 每个客体都拥有矩阵中的一列。客体可以是程序、文件或设备
 - 访问矩阵模型决定在任何域中的进程可以执行的操作



信息流控制模型是对访问矩阵模型的补充，用于监管信息在系统中流通的有效路径，控制信息流从一个实体沿着安全路径流向另一实体。



Bell-La Padula模型：最广泛使用的信息流控制模型

- 信息分为4等：无密级、秘密级、机密级和绝密级
- 只要系统严格执行两项规定，就能确保信息的安全流动
 - 不能上读：在密级 k 层中运行的进程，只能读相同或更低密级层中的对象
 - 不能下写：在密级 k 层中运行的进程，只能写相同或更高密级层中的对象



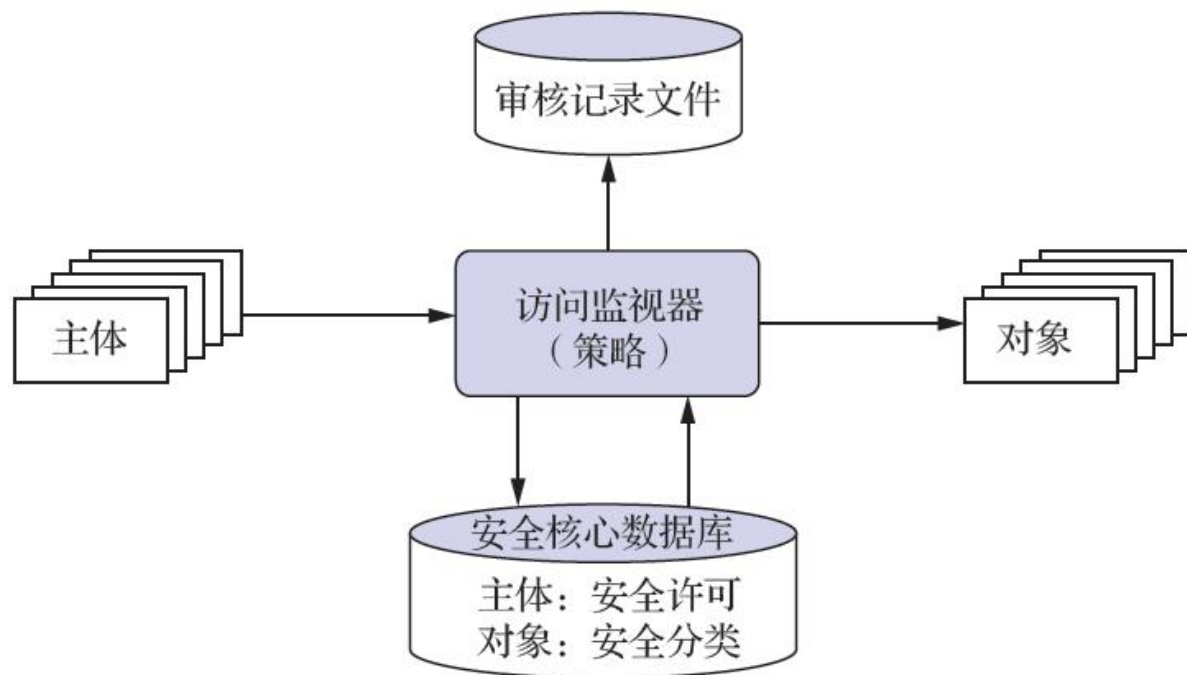


可信系统的核心是可信计算基TCB，它包含了用于检查所有与安全问题有关的访问监视器和存放着许多与安全有关的信息的安全核心数据库等。



TCB的功能：

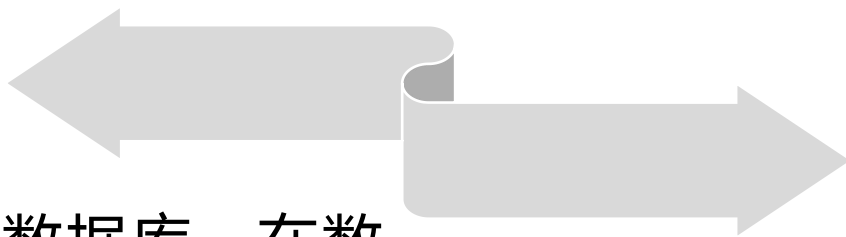
- 典型的TCB在硬件方面与一般计算机系统相似，只是少了一些不影响安全性的I/O设备；
- TCB中应配置OS最核心的功能，如进程创建、进程切换、内存映射、部分文件管理以及设备管理等功能；
- **访问监视器**：它要求所有与安全有关的问题都必须集中在一处进行检查。





在TCB中配置安全核心数据库，在数据库内存放了许多与安全有关的信息

- ① 访问控制模型，用于实现对用户访问文件的控制，其中列出了每个主体的访问权限和每个对象的保护属性；
- ② 信息流控制模型，用于控制信息流从一个实体沿着安全路经流向另一个实体



访问监视器

➤ 它基于主体和被访问对象的安全参数来控制主体对该对象的访问，进而实现有效的安全接入控制

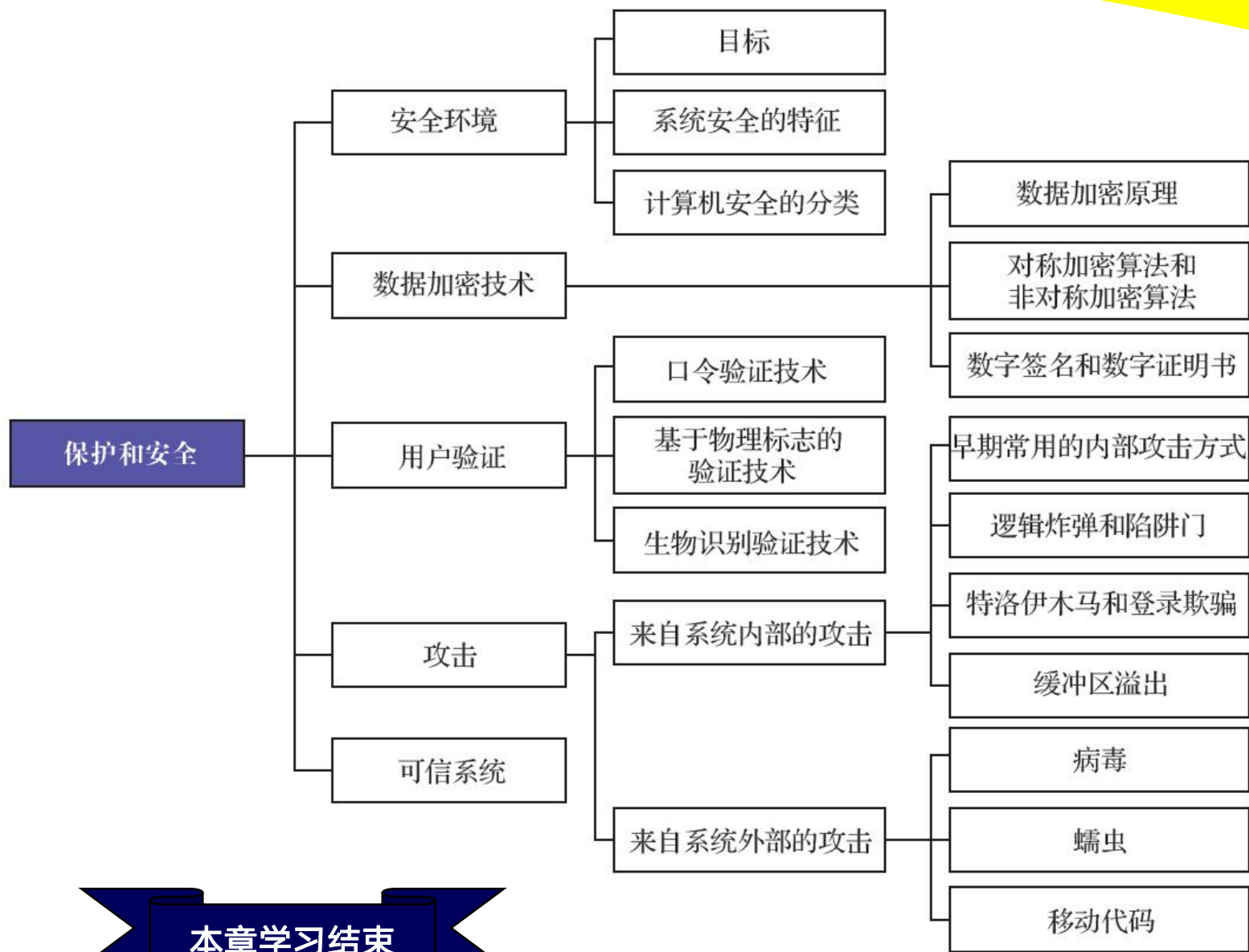
- ① 完全仲裁
- ② 隔离 ③ 可证实性





学而时习之（第12章总结）

第1章	操作系统引论
第2章	进程的描述与控制
第3章	处理机调度与死锁
第4章	进程同步
第5章	存储器管理
第6章	虚拟存储器
第7章	输入/输出系统
第8章	文件管理
第9章	磁盘存储器管理
第10章	多处理机操作系统
第11章	虚拟化和云计算
第12章	保护和安全的



本章学习结束



简答题

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27					

标黄色为本次作业



简答题

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27					

标黄色为本次作业

互联网行业的“苦行僧”，由始至终从未言弃

早在20世纪70年代，国内许多科研院所就已参与到了UNIX自主操作系统的研发中。伴随着Linux系统的诞生，其凭借开源特征得以迅速取代UNIX成为国产操作系统开发的主流。

20世纪90年代，曾是中国Linux公社（Linuxfans）的发起人和重要参与者之一的黄建忠，做了一个彻底改变他人生轨迹的选择——投身北京中科红旗软件技术有限公司（简称中科红旗）。他带领团队以Linux系统为基础二次开发操作系统，经过多年的努力，国产操作系统无论是布局还是操作方式，都同Windows XP相差无几，并在易用性等方面基本具备了Windows XP替代能力。然而，当黄建忠真正走进市场才发现，原本以为是“蓝海”的操作系统领域，其实是“死海”！



互联网行业的“苦行僧”，由始至终从未言弃

2009年，中国电子信息产业的国家队——中国电子科技集团，整合集团优势资源，投资设立普华公司，黄建忠又一次义无反顾地投身其中，成为普华技术部研发总监，中科红旗的创业团队也先后投奔过来。普华肩负提升国家基础软件产业核心竞争力的重要使命，2014年初正式进军国产操作系统领域。

雄厚的资金支持加上富有经验的团队，使得当年9月普华操作系统3.0版本便正式发布。但是走进市场才发现，绝大多数人依旧只认Windows！

直到“棱镜门”事件爆发与WannaCry（音译“想哭”）病毒肆虐全球，网络安全才逐步上升到国家战略层面，具备网络安全优势的国产操作系统逐渐深入人心，得到了业内外的普遍认同，并逐步进入国家政府部门以及金融、能源等经济社会运行的神经中枢。



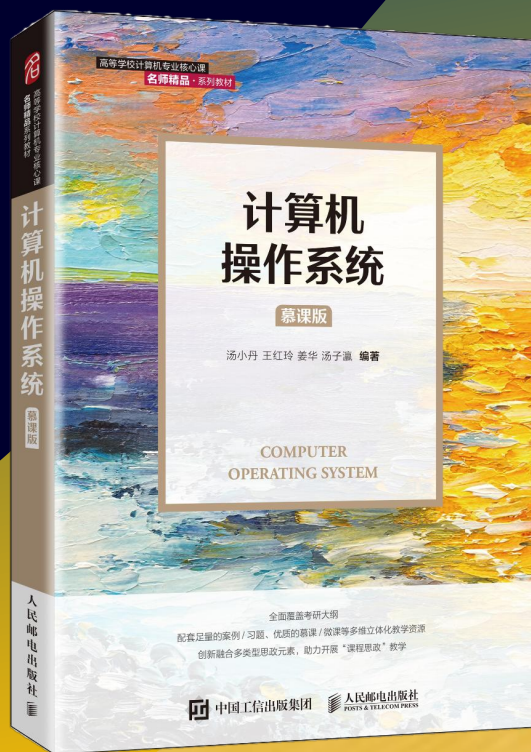
互联网行业的“苦行僧”，由始至终从未言弃

诚然，操作系统的自主可控是网络强国的关键基石。

最后，特向所有为国产操作系统研发工作做出贡献、付出心血的各方人士，
致敬！



经典教材《计算机操作系统》**最新版**



学习进步！

作者：汤小丹、王红玲、姜华、汤子瀛