(2019春季　课程编号：011184)

信息安全导论

曾凡平
2019信安导论

第6章 主机系统安全技术

中国科学技术大学 曾凡平
billzeng@ustc.edu.cn

课程回顾：第5章 信息隐藏技术

5.1 信息隐藏的概念	
5.2 隐藏信息的基本方法	• 空域算法，变换域算法 • 压缩域算法，NEC算法，生理模型算法
5.3 数字水印	• 技术模型，分类与应用 • 空域水印，DCT水印
5.4 数字隐写	• 技术模型， • 典型数字图像隐写算法
5.5 数字指纹	• 基本概念和模型 • 数字指纹编码，数字指纹协议

第6章 主机系统安全技术

6.1 操作系统安全技术

- 6.1.1 基本概念
- 6.1.2 可信计算机评价标准(TCSEC)
- 6.1.3 操作系统安全的基本原理
- 6.1.4 操作系统安全机制
- 6.1.5 Linux的安全

6.2 数据库安全技术

- 6.2.1 传统数据库安全技术
- 6.2.2 外包数据库安全
- 6.2.3 云数据库/云存储安全

6.3 可信计算技术

- 6.3.1 概念和基本思想
- 6.3.2 TCG可信计算系统

第6章 操作系统安全技术

- 主机系统安全，即保证主机数据存储和处理的保密性、完整性、可用性，其核心内容包括安全应用交付系统、应用监管系统、操作系统安全增强系统和运维安全管控系统等。
- 主机系统安全包括硬件、固件、系统软件的自身安全，以及一系列附加的安全技术和安全管理措施，从而建立一个完整的主机安全保护环境。

- 本章主要介绍操作系统安全、数据库安全技术和可信计算技术。

6.1 操作系统安全技术

6.1.1 基本概念

- 计算机系统由硬件系统和软件系统组成，软件系统又可以分为系统软件和应用软件，其中，系统软件主要由操作系统、数据库系统等组成。因此，关于**计算机系统的安全可以划分为硬件安全、操作系统安全、数据库系统安全、应用软件安全以及互联网时代的网络系统安全。**
- 操作系统运行在硬件系统之上，为用户提供接口，用户只能通过操作系统提供的接口来操作硬件，因此要保证硬件的安全，就必须让操作系统提供安全的接口。数据库系统、应用软件以及网络软件都运行在操作系统之上，要保证的它们的安全性，除了依靠自身的安全性以外，关键在于其底层操作系统的安全性。
- **操作系统安全是主机系统安全的核心。**

安全操作系统

- AT&T 实 验 室 的 S.Bellovin 博 士 曾 经 对 美 国 CERT(Computer Emergency Response Team)提供的安全报告进行过分析，分析结果表明，大约50%的计算机网络安全问题是由软件工程中产生的安全缺陷引起的，其中，很多问题的根源都在操作系统的安全脆弱之中。因此，**操作系统安全是所有计算机系统安全的基石和关键**。要真正解决硬件系统、数据库系统、应用软件以及网络系统的安全问题，就必须解决操作系统的安全问题。
- 所谓**安全操作系统，是在传统操作系统的基础上实现了一定安全技术的操作系统**。它提供访问控制、最小特权管理和安全审计等机制，采用各种安全策略模型，在系统硬件和资源以及用户和应用程序之间进行符合预定义安全策略的调用，限制对系统资源的非法访问和阻止黑客对系统的入侵。

6.1.2 可信计算机评价标准(TCSEC)

- 如何评价一个操作系统是否是一个安全操作系统以及达到了哪种程度的安全级别，需要一个评价其安全性和安全级别的标准。因此，制定安全评价标准的目的在于对安全操作系统（也可以是其他类的安全产品）进行评价。
- 可信计算机系统评价标准(trusted computer system evaluation criteria，TCSEC)于1983年由美国国防部发布，是计算机系统安全评价的第一个正式标准。1985年美国国防部又发布了其修订版，从而成为公认的计算机系统安全级别的划分标准。
- TCSEC最初只是军用标准，后来延至民用领域。安全产品在美国必须按照TCSEC的标准进行严格地测试并颁发相应级别的证书，才能用于销售和使用。

TCSEC 6个规范性的安全要求

- (1)计算机系统必须实施一种定义清晰明确的安全策略。即给定主体和客体，系统必须有一套定义好的策略来判断该主体是否有权访问该客体。
- (2)客体必须与其访问标签相关联，以标明其安全级别。
- (3)主体在访问客体前必须通过严格的鉴别和认证。
- (4)审计信息必须单独保存，并由专门人员负责。在可信计算机系统中，与安全相关的事件被记录在审计日志中，审计的开启尽可能不影响系统的整体性能。另外，审计数据应该易于分析，并确保相应的数据不会被非法存取和篡改。
- (5)计算机系统必须能够独立评估用以实现上述(1)～(4)的软硬件机制本身的安全性。
- (6)实现安全需求的可信机制自身必须受到保护，以避免被篡改或削弱。

TCSEC的4个等级、7个级别

1) D类安全等级

- D类安全等级只包含一个级别—D级。D1级的安全等级最低，并且未通过评测，系统不可信任，对硬件来说也没有任何保护，操作系统很容易受到损害，不提供身份验证和访问控制等机制。
- 所有不满足任何较高安全可信性的系统都可以归入D1级，D1系统最普通的形式是本地操作系统，如MS-DOS等，D1系统只为文件和用户提供安全保护。
- 最初的Windows系统（95、98）就是D1级系统。

2)C类安全等级

- **该类为自主保护类**，能够为用户的行动和责任提供审计机制。
- C1（自主安全保护）：C1级的系统对硬件来说提供了某种保护。用户必须通过身份验证才能访问系统，并且通过用户的标识和鉴别来确定用户对客体的访问权限，但对系统管理员则没有这个限制，因而，很容易会由于系统管理员的误操作而导致诸多的安全问题。
- C2(受控存取保护)：C2级的系统具有C1级系统的所有安全特征，此外，C2级具有 DAC机制以及审计机制。基于许可权限和身份验证的DAC机制限制了主体执行某些命令或访问某些客体的能力。在这种安全级别下，系统采用审计机制跟踪记录所有与系统安全有关的事件。

3)B类安全等级

- **该类为强制保护类。**B类级别的系统建立了敏感度标签并维护其完整性，实施强制访问控制机制。B类安全等级可分为B1、B2和B3三个子级别。
- B1(标签安全保护，labelled security protection)：B1级要求具有C2级的所有安全特征，并具有强制访问控制(MAC)机制。
- B1级系统满足如下要求：
- (1)系统对每个客体或主体都进行敏感度标记。
- (2)系统使用敏感度标记作为所有强制访问控制的基础。
- (3)系统在把非标记的客体或主体对象导入系统前必须标记它们。

B1级系统的要求

- (4)敏感度标记必须准确地表示其所联系的客体或主体的安全级别。
- (5)当系统管理员创建系统或者增加新的客体（如通信通道或I/O设备）时，管理员必须指定该客体是单级还是多级，并且管理员只能手工改变指定。
- (6)单级设备并不保持传输信息的敏感度。
- (7)所有直接面向用户位置的输出（无论是虚拟的还是物理的）都必须产生敏感度标记来指示关于输出对象的级别。
- (8)系统必须使用口令来决定主体的安全访问级别。
- (9)系统必须通过审计来记录未授权访问的企图。
- **B1级是支持多级安全的第一个级别。在该级别下，不允许客体的拥有者改变其存取的许可权限。**

B2(结构化保护，structured protection)

- B2级要求具有B1级系统的所有安全特征。此外，它要求具有形式化的安全模型、可信通路机制、最小特权管理以及隐通道的分析和处理等安全特征。

- B2级要对系统中所有的对象（包括磁盘、磁带或终端）定义不同的安全标签，并相应地分配单个或多个安全级别。

B3(安全区域保护，security domain protection)

- B3级除了具有所有B2的安全特征外，还要求具有全面的访问控制机制、审计的实时报告机制以及严格的系统结构化设计等安全特征。

- 同时，B3级要求提供登录系统的可信通路，并通过硬件保护系统的安全区域。

- B3级系统具有很强的监视委托管理访问能力和抗干扰能力。B3级系统必须设有安全管理员。

B3级系统应满足的要求

- (1)除了控制对个别客体的访问外，B3级系统必须产生一个可读的安全列表。
- (2)每个被命名的客体提供对该客体没有访问权的主体列表说明。
- (3)B3级系统在进行任何操作前，要求主体进行身份验证。
- (4)B3级系统验证每个主体，同时还会发送一个取消访问的审计跟踪消息。
- (5)设计者必须正确区分可信通路和其他路径。
- (6)可信通路机制为每一个被命名的客体建立安全审计跟踪。
- (7)可信通路机制支持独立的安全管理。

4)A类安全等级

- A类安全等级(**验证保护，verified protection**)：A系统的安全级别最高。目前，A类安全等级只包含一个安全级别—A1(验证设计，verified design)，该级别包含了较低级别的所有安全特征。它的显著特征是，系统的设计者必须按照一个正式的设计规范来分析系统。对系统分析后，设计者必须运用核对技术来确保系统符合设计规范，它的设计必须是经过数学验证的，并且对隐通道也要进行形式化分析。A1级的系统必须满足下列要求：
- (1)系统管理员必须从开发者那里接收到一个安全策略的正式模型。
- (2)所有的安装操作都必须由系统管理员进行。
- (3)系统管理员进行的每一步安装操作都必须有正式文档。
- A1级系统是安全级别最高的系统，目前很少有系统能够达到A1级别。

6.1.3 操作系统安全的基本原理

- 操作系统的安全取决于安全功能在系统中实现的完整性、系统文档的清晰性、系统测试的完备性以及形式化验证程度。操作系统可以分成内核部分和应用部分，在操作系统内核中存在错误或设计缺陷的情况下，即使在应用部分采用必要的安全机制也不能保证系统具有满意的安全性。
- 由于这些错误或缺陷的存在，进程可以对某些客体进行未经授权的访问或避免受到安全机制的限制，并且，要验证整个操作系统的安全性是十分困难的。所以**应该使用操作系统中尽量小的部分来提供整个操作系统的安全性，这就提出了安全核的概念。**

安全核

- 安全核是安全操作系统设计的最关键问题。它必须能够保证系统中所有的访问控制要求都满足系统的安全策略。
- **基于安全核构建安全操作系统**具有两个方面的优势：一方面能够减轻应用系统的负担，避免出现安全隐患；另一方面，由于对系统的安全进行评估的内容集中在安全内核，它有利于评估的进行，使之可以进行严格的形式化验证。

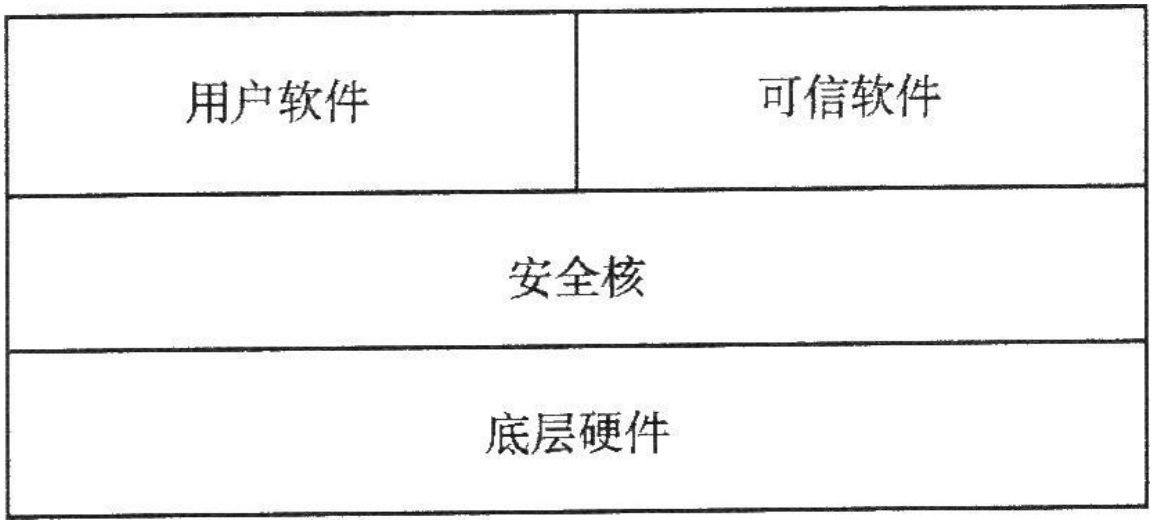


图6-1 安全操作系统的通用结构

1.引用监视器与安全核

- J.P.Anderson于1972年在一份研究报告中提出了引用监视器(reference monitor)、引用验证机制(reference validation mechanism)、安全核(security kernel)和安全建模(modeling)等重要思想。这些思想是在研究系统资源受控共享(controlled sharing)问题的背景下产生的。
- **引用监视器和安全核把授权机制与能够对程序的运行加以控制的系统环境结合在一起，可以对受控共享提供支持。**授权机制负责确定用户（程序）对系统资源（数据、程序、设备等）的引用许可权，程序运行控制负责把用户程序对资源的引用控制在授权的范围之内。

系统资源的受控共享

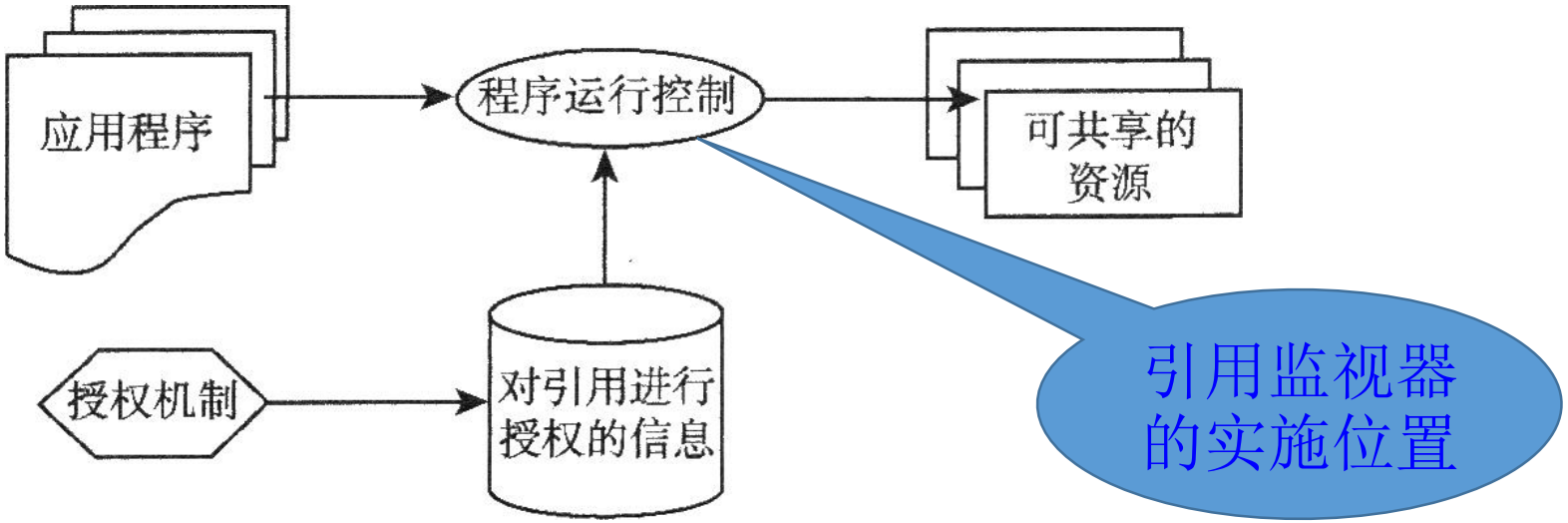


图6-2 系统资源的受控共享

- 引用监视器的思想是为了解决用户程序的运行控制问题而引入的，其目的是在用户（程序）与系统资源之间实施一种授权的访问关系。
- J.P. Anderson把引用监视器的职能定义为：以主体（用户等）所获得的引用权限为基准，验证运行中的程序（对程序、数据、设备等）的所有引用。

引用验证机制的3原则

- 引用监视器是一个抽象的概念，它表现的是一种思想。J.P.Anderson把引用监视器的具体实现称为**引用验证机制**，它是实现引用监视器思想的硬件和软件的组合。引用验证机制需要同时满足以下3个原则：

(1)必须具有自我保护能力。

- 保证引用验证机制即使受到攻击也能保持自身的完整性。

(2)必须总是处于活跃状态。

- 保证程序对资源的所有引用都得到引用验证机制的仲裁。

(3)必须设计得足够小，以便分析和测试。

- 保证引用验证机制的实现是正确的和符合要求的。

安全核

- **安全核是系统中与安全性的实现有关的部分，包括引用验证机制、访问控制机制、授权机制和授权的管理机制等**(J.P.Anderson的定义)。
- J.P.Anderson指出，要开发安全系统，首先必须建立系统的安全模型。**安全模型给出安全系统的形式化定义，正确地综合系统的各类因素**。这些因素包括：系统的使用方式、使用环境类型、授权的定义、共享的客体（系统资源）、共享的类型和受控共享思想等。这些因素应构成安全系统的形式化抽象描述，使得系统可以被证明是完整的、反映真实环境的、逻辑上能够实现程序的受控执行的。完成安全系统的建模之后，再进行安全内核的设计与实现。

2.安全核与TCB

- TCSEC标准是在基于安全核技术的安全操作系统研究的基础上制定出来的，标准中使用的**可信计算基(trusted computing base，TCB)**就是安全核研究结果的表现。
- **TCB在TCSEC中的定义**是：**一个计算机系统中的保护机制的全体**，它们共同负责实施一个安全策略，包括硬件、固件和软件；一个TCB由在一个产品或系统上共同实施一个统一的安全策略的一个或多个组件构成。
- 在操作系统中，实施安全策略的可信软件和硬件以及负责安全管理的人员共同组成了系统的可信计算基(TCB)，具体包括以下方面。

可信计算基(TCB)的构成

- (1)固件和硬件：包括CPU、内存、寄存器和I/O设备等，为了保证系统的安全性，这些部分必须能够可信地完成它们的设计任务。
- (2)与安全策略相关的文件：比如安全策略库、标识与鉴别的数据库等。
- (3)负责安全管理的人员：他们一般具有比较大的权限，所以很容易引起系统的安全问题。
- (4)安全核：它为整个操作系统提供安全机制，是判断一个操作系统是否安全的基础。
- (5)具有特权的进程或命令。

TCB的基本功能

- **(1)进程的活动**：在多任务系统中，多个进程并发执行，进程间会进行频繁地切换。进程间切换必然会导致内部寄存器、内存映像和进程的状态信息等的变化，而这些大部分都是安全的敏感信息。
- **(2)执行域交换**：在一个安全域中运行的进程往往会调用其他安全域的进程来获得必要的信息和服务，这些信息和服务大部分也都是敏感数据。
- **(3)I/O操作**：I/O操作涉及大量数据的流动，这些数据中必然包含敏感的信息。
- 安全核是TCB的一个子集。
- **安全核在TCSEC中的定义是：一个TCB中实现引用监视器思想的硬件、固件和软件成分**，它必须仲裁所有访问、必须保护自身免受修改、必须能被验证是正确的。

3.安全操作系统的设计方法

- 在一个原有的通用操作系统的基础上开发一套安全机制，提高其安全性，是人们目前常用的方法。在一个原有的非安全操作系统基础上，开发其安全性，一般有以下三种方法，如图6-3所示。

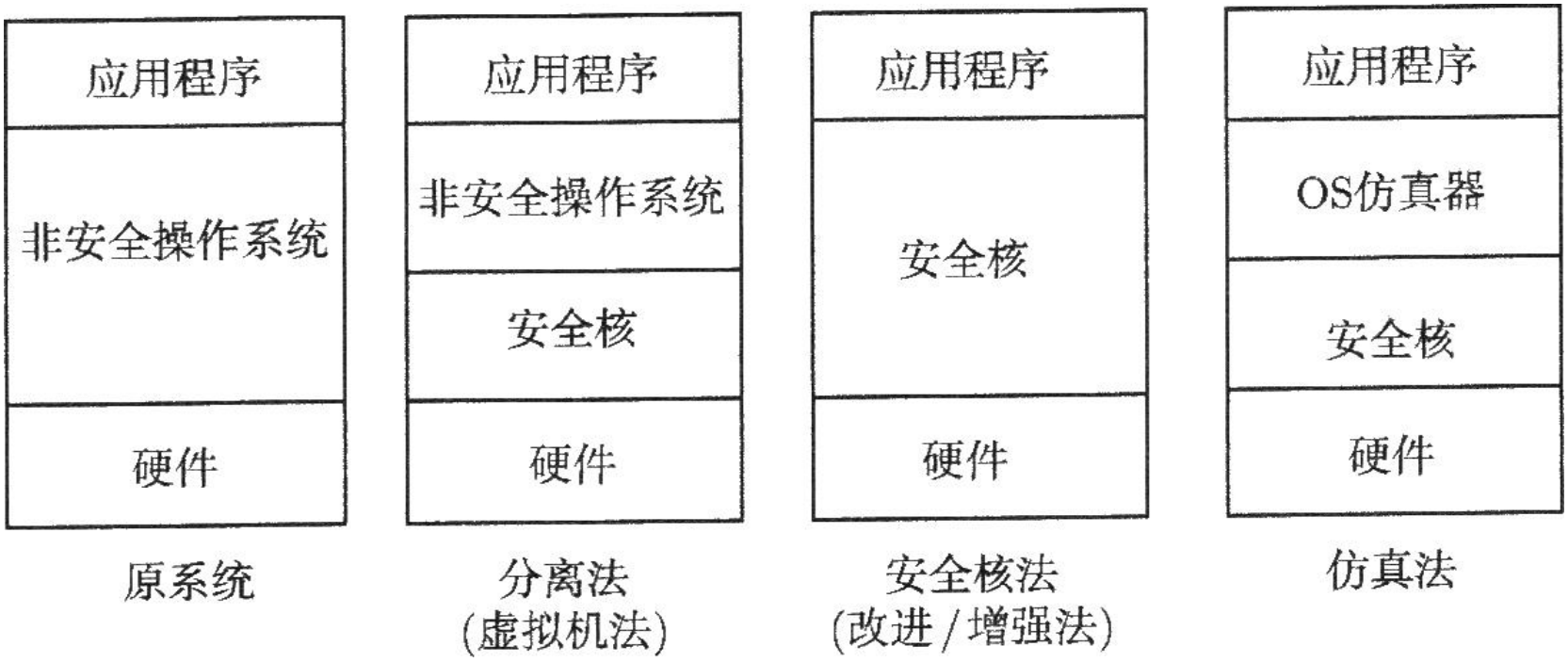
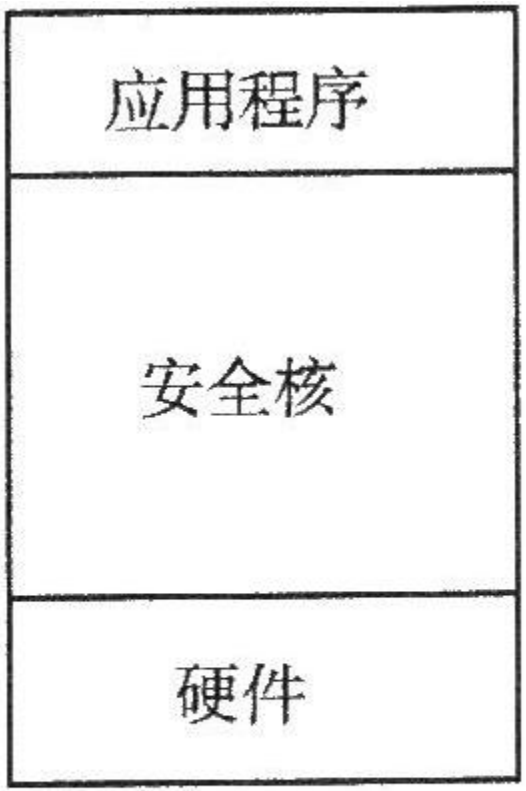


图6-3 安全操作系统的设计方法

1)分离法（虚拟机法）

- 分离法也称为虚拟机法。分离用户的办法有四种：物理分离、时间分离、密码分离及逻辑分离。物理分离指用户使用不同的硬件设施；时间分离指用户在不同时间使用机器；密码分离是指将数据加密；逻辑分离是通过引用监视器将每个用户和其他用户隔离开。
- 安全操作系统通过上述方法的综合运用分离用户，仅允许用户间进行严格控制下的交互作用，如IBM MVS使用多虚存空间将操作系统的拷贝放在每个用户的逻辑地址空间来为多用户提供单个工作环境，用户的程序看似运行在完全分离的机器上。而IBM VM操作系统则提供了更强一些的保护机制，它为每个用户提供整个虚拟机。

2)安全核法（改进 / 增强法）



安全核法
(改进 / 增强法)

- 通过重新生成操作系统以使与安全相关的软件构成操作系统的一个可信安全核，而操作系统的大部分不必承担安全任务，即一个大型操作系统软件的一小部分是用于安全目的。
- 安全核必须予以适当的保护，同时，绝不能绕过安全核的存取控制检查。此外，安全核必须尽可能小，以便进行正确性验证。

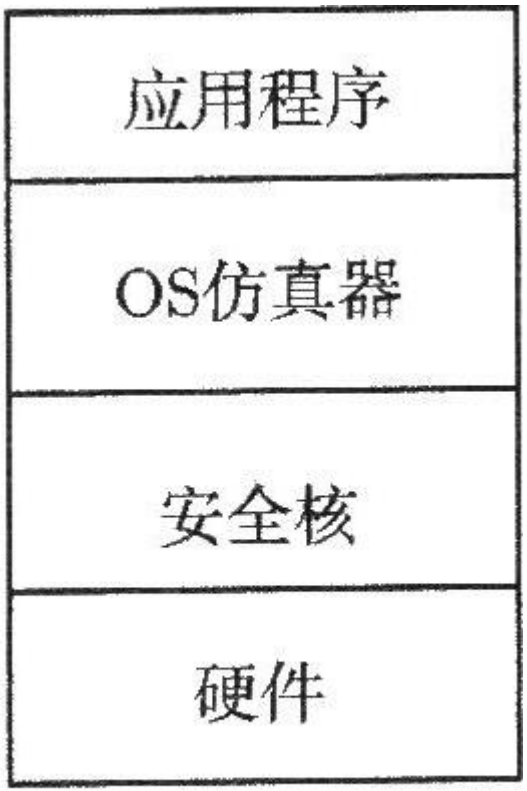
安全核的设计方法

- **(1)在操作系统内核中加入安全功能：**
- 这种情形是目前最受限制和最有实际意义的。在多数情况下，安全是对系统改进的唯一动力，需要尽可能多地应用已有操作系统的代码且必须保持与已有应用程序完全兼容。
- 在这种方法中，原有的操作系统几乎整个地被保留下来，从已实现的操作系统中分离出安全功能，建立安全核，作为已有操作系统的新层次。

- **(2)先设计安全核功能：**
- 先设计安全核，然后围绕它设计操作系统。
- 设计者可定义自己的操作系统接口，允许定义有关与安全核执行的安全策略兼容的操作系统接口。这个接口应与安全核提供的功能非常吻合，或为一对一的，或为操作系统功能到安全核功能的有关简单映射。由于能自由而灵活地选择接口，所以可设计出最小的安全核和良好的操作系统。

3)仿真法

- 仿真法是将非安全操作系统的内核进行修改，加入相应的安全机制和安全策略，使之成为安全核，然后再在安全核和原来的非安全操作系统的用户接口层中间编写一个仿真层。这种设计方法可以完全自由地定义安全核和仿真层的操作接口，不会受到现有应用程序的限制。
- 它的不足在于要同时设计和编写安全核和仿真程序，工作量比较大，而且在仿真的时候，有些功能的仿真有一定难度。



仿真法

6.1.4 操作系统安全机制

- 操作系统安全的主要目标有：标识用户身份及身份鉴别；按访问控制策略对系统用户的操作进行控制；防止用户和外来入侵者非法存取计算机资源；监督系统运行的安全性和保证系统自身的完整性等。要完成这些目标，需要建立相应的安全机制，包括硬件安全机制和软件安全机制。

I. 硬件的安全机制主要包括：内存管理、运行域保护和I/O管理。

II. 软件的安全机制主要包括：标识与鉴别机制、访问控制机制、最小特权管理机制、可信通路机制、隐通道的分析与处理以及安全审计机制等。

1．硬件系统安全机制

- 要保证操作系统的安全性，必然要保证硬件层操作的安全性。

1)内存保护

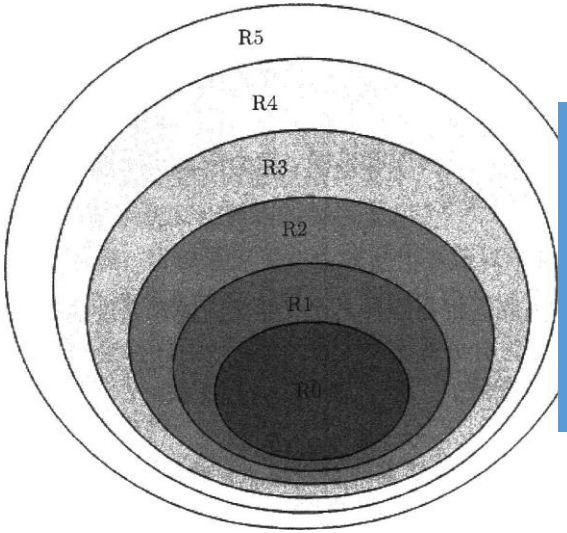
- 确保存储器中的数据能够被合法地访问。保护单元是存储器中最小的数据范围，可以分为块、段或页等，保护单元越小，则存储保护的精度越高。
- 在多任务的环境中，应该防止用户程序访问操作系统内核的存储区域以及进程间非法访问对方的存储区域。
- 内存保护与内存管理是紧密相关的，内存保护是为了保证系统各进程间互不干扰以及用户进程不非法访问系统空间，而内存管理则是为了更有效地利用系统的资源—内存空间。

- 系统会区分用户空间和系统空间，在用户模式下运行的非特权程序应该禁止访问系统空间，而在内核模式下则可以访问任何内存空间，包括用户空间。用户模式和内核模式的切换应该通过一条特权指令来完成，这种**访问控制一般可以由硬件来实现**。
- 如Intel的CPU可以在四个不同的等级下运行，其中0级为特权级，3级为用户级。在Linux的实现中，0级对应内核模式，而3级则对应用户模式，中间两级没有使用。
- 除了通过硬件的限制来实现内存保护，还可以通过软件实现对内存的保护，如基于描述符的地址解释机制，该机制可以解决段／页访问权限的标识问题。

- 在这种机制下，系统会给每一个进程分配一个私有的地址描述符，进程对系统中内存段 / 页的访问模式都在该描述中进行了说明，指出了该进程对内存段 / 页的访问模式。
- 比如可以有两种访问模式集，一类用于在用户模式下运行的进程，另一类用于在内核模式下运行的进程。访问模式包括读、写、执行，各占地址描述符的一位。
- 因为在地址解释的同时，系统调用会检查地址描述符，所以，这种机制在运行模式切换和进程切换的过程中只需要很少的额外开销，比较适合用于内存管理的访问控制。

2)运行域保护

- **运行域是进程运行的区域。**一般操作系统都会包含硬件层、内核层、应用层、用户层等层次，而每个层次又会包含子层，这种分层的设计方法是为了隔离运行域，达到运行域保护的目的。



R0-硬件；
R1-安全核；
R2-操作系统内核；
R3-操作系统；
R4-系统应用程序；
R5-用户应用程序

图6-4 环结构的运行域

- **运行域可以看成是一系列的同心圆，最内层的特权最高，最外层的特权最低。**一个进程的可信度和其访问权限可以通过它与中心的接近程度来衡量，特权等级越高则越接近中心。
- 这种达到运行域保护的系统是一种分级的环结构，以最底层硬件层为中心，最后到特权最低的用户层(图6-4)。

多环结构

- 多环结构最重要的安全概念是：等级域机制应该保护内层环不被其他外层环侵入，每一个进程都在特定的环层运行，特权越高的进程在环号越低的层上运行。
- 环是重叠的，即在i层环运行的进程必然拥有所有j(j>i）层环的特权。环号越低，特权越高，相对于该层的操作保护越少。
- 等级域机制和进程隔离机制是互不影响的，一个进程可以在任意时刻在任意环内运行，在运行时还可以在各环间转移。当进程在特定环运行时，进程隔离机制将避免该进程遭同环内其他进程的破坏，系统会隔离在同一环内同时运行的进程。

内存保护中的等级域机制

- 以在内存保护中介绍的地址解释机制为例，在等级域机制中，需要在每个段描述符中保存多个独立的读、写、执行位集，每个位集提供该进程对相应环的访问模式，这种做法比较占用空间，而且效率不高。
- 一般采用等级简化的管理机制，例如，环N对某段具有写操作的权限，那么所有0～N-1环对该段都具有写操作。
- 因此，只要在段描述符中存入具有相应访问模式的最大环号，这样只要三个区域分别用来保存具有**写、读、执行访问模式中的最大环号，这三个环号称为环界**。
- 比如在某个段描述符中，环界集为(1，2，5)，则表示：0～1环的进程可以对该段进行写操作；0～2环的进程可以对该段进行读操作；0～5环的进程可以对该段进行执行操作。

内存保护中的等级域机制

- 此外，如果某段对于具有较低特权的环是可写的，那么在具有较高特权的环中执行它将是危险的，因为较低特权的环可能对该段写入了对系统具有破坏作用的代码。
- 如果某段对于具有较高特权的环是可写的，那么在具有较低特权的环中读取该段将会导致敏感数据的泄露。
- 因此，从安全的角度，不应该允许较低特权的环中可写的段在较高特权的环中执行，也不允许在较高特权的环中可写的段在较低特权的环中可读。

3) I/O保护

- 安全的缺陷往往可以从操作系统的I/O部分找出来，因此，为保证安全性，I/O应该是只有操作系统才可以完成的特权操作。对于一般的I/O设备，操作系统都会提供该设备的系统调用。对于网络访问，一般也提供标准的调用接口，用户不需要操作I/O的细节。
- I/O设备最简单的访问控制方式是把一个I/O设备看成是一个客体，所有对I/O设备的操作，比如读设备、写设备等都必须经过相应的访问控制机制，操作系统内核通过比较安全策略数据库来决定相应主体对相应客体的访问权限。
- 在比较关键的安全系统中，除了采用CPU的隔离保护机制外，还需要专用的硬件如智能卡等加以进一步的保护。当然，如果要对系统提供足够的安全强度，必须将硬件和软件很好地结合起来，采用适当的安全机制才能更好地保护系统。

2．软件系统安全机制

1)标识与鉴别机制

- 标识与鉴别是操作系统中的重要技术。标识是用来标明用户的身份，确保用户在系统中的唯一性和可确认性。一般可以用名称和标识符(ID)来标明系统中的一个用户。名称和标识符可以都是公开的明码信息。鉴别是对用户身份的真实性进行识别，用于鉴别的信息一般是非公开的，并且难以伪造。标识与鉴别机制的安全性不但是操作系统本身安全性盼重要部分，也是操作系统实现其他安全功能的保证。不管操作系统实现了什么样的访问控制，都需要标识与鉴别机制的实现作为基础。
- 一般可以采用三种方法实现标识与鉴别机制：第一种是采用口令机制；第二种就是采用智能卡等用户本身携带的信息作为鉴别机制；第三种就是使用用户本身的信息，如签名、指纹、视网膜等

2)访问控制

- **最小特权(least privilege)指的是在完成某种操作时授予每个主体（用户或进程）必不可少的特权。**它的思想是，**系统只给用户执行任务所需的最少的特权，也就是用户所得到的特权仅能完成当前任务。**最小特权原则是系统安全中最基本的原则之一，它限定每个主体所必需的最小特权，确保可能的事故、错误、网络部件的篡改等原因造成的损失最小。
- 最小特权原则一方面给予主体“必不可少”的特权，这就保证了所有的主体都能在所赋予的特权之下完成所需要完成的任务或操作；另一方面，它只给予主体“必不可少”的特权，这就限制了每个主体所能进行的操作。

审计机制

- 一个安全操作系统的审计机制就是对系统中有关安全的活动进行记录、检查及审核。它的主要目的就是检测和阻止非法用户对计算机系统的入侵，并显示合法用户的误操作。
- 审计是一种被信任的机制，是TCB的一个部分。引用监视器也使用审计把它的活动记录下来。引用监视器记录的信息应包括主体和客体的标识、访问权限请求、日期和时间、参考请求结果(成功或失败）等。审计记录应以一种确保可信的方式存储，审计机制为记录下列事件负责，如开始一个程序、结束一个程序、系统重新启动、增加用户、改变用户口令、安装上新的磁盘驱动器等。
- 审计过程一般是一个独立的过程，它应与系统其他功能隔开，操作系统必须能够生成、维护及保护审计过程，防止其被非法修改、访问和毁坏，特别是要保护审计数据，严格限制未经授权的用户访问。

6.1.5 Linux的安全机制

1. 标识与鉴别机制

- 一般来说，Linux使用用户名和用户ID标识用户，使用口令来鉴别用户。系统为每个用户分配一个用户名和用户ID以及一个组名和组ID，当用户登录时，用户名被映射为整数，来标明自己的“UID”(用户ID)和作为其中成员的“GID”(组ID)。UID和GID是核心进行访问控制检查的主要依据。在Linux中，UID为“0”是传统上被称为root的特权用户，root可以不受安全性检查，负责管理系统。
- 尽管这是几乎所有类UNIX系统一直沿循采用的访问控制机制，但很显然，这样是违反“完全仲裁”和“最小特权”安全原则的。再加上 SUID和GUID机制的存在，进一步加重了它的危害性。

Linux的标识与鉴别机制(演示)

2．安全注意键

- 为了使用户确信自己的用户名和口令不被别人窃走，Linux提供了安全注意键。
- 安全注意键(secure attention key，SAK)是一个键或一组键(在Intel x86平台上，SAK是 ALT+SySRq+k)，按下它（们）后，保证用户看到真正的登录提示，而非登录模拟器，即保证是真正的登录程序（而非登录模拟器）读取用户的账号名和口令。这最早是在TCSEC中可信路径所要求的一项功能。
- SAK可以用下面命令来激活：
  - echo 1>/proc/sys/kernel/sysrq。
- 严格地说，Linux中的SAK并未构成一个可信路径。因为尽管它会杀死正在监听终端设备的登录模拟器，但它不能阻止登录模拟器在按下SAK后立即开始监听终端设备。

3．LKM机制

- Linux内核是一个宏内核结构，它包括两部分：内核空间与用户空间。系统中的绝大多数程序运行在用户空间，它们各自运行在核心分配给它们的地址空间中，并且不能直接访问内核空间，内核通过系统调用向应用程序提供服务。
- 从版本2.0开始，Linux开始支持LKM机制，也就是可加载内核模块，简单地说就是在内核里动态载入代码的能力。系统调用create_module、init_module、query_module以及 delete_module等分别用于创建、初始化、查寻和删除模块。LKM可以用来在运行时支持新的文件系统和设备驱动，而不用重启系统。
- LKM的设计是合理和先进的，并且它有利于Linux的并发开发模式。但是，由于加载以后的LKM能够不受控制地使用内核的所有功能和内存，所以很容易引起恶意程序作为模块加入内核空间后破坏系统。

4．能力机制

- Linux中的能力机制是通过是一组比特位来实现的，该机制将root拥有的特权分割成一组特权。Linux的能力机制沿用了POSIX.1e中的能力机制，从2.2版本开始就增加了对进程的POSIX能力的支持。
- 一个进程有三组位图，分别表示三组能力：有效的、可继承的和许可的能力。位图中的每一位表示一个能力，值为1时表示具有此能力。
- 如表6.1所示，Linux中实现了POSIX.1e中的9个能力，其他21个能力是Linux所特有的。

表6-1 Linux的能力表

0	CAP_CHOWN	15	CAP_LOCK
1	CAP_DAC_OVERRIDE	16	CAP_IPC_OWNER
2	CAP_DAC-READ_SEARCH	17	CAP_SYS_MODULE
3	CAP_FOWNER	18	CAP_SYS_RAWIO
4	CAP_FSETID	19	CAP_SYS_CHROOT
5	CAP_KILL	20	CAP_SYS_PTRACE
6	CAP_SETGID	21	CAP_SYS_PACCT
7	CAP_SETUID	22	CAP_SYS_ADMIN
8	CAP_FS_MASK	23	CAP_SYS_BOOT
9	CAP_SETPCAP	24	CAP_SYS_NICE
10	CAP_LINUX_IMMUTABLE	25	CAP_SYS_RESOURCE
11	CAP_NET_BIND_SERVICE	26	CAP_SYS_TIME
12	CAP_BROADCAST	27	CAP_SYS_TTY_CONFIG
13	CAP_NET_ADMIN	28	CAP_MKNOD
14	CAP_NET_RAR	29	CAP_LEASE

能力机制

- 其中，0~8这9个能力是从POSIX.1e中沿用的。
- 从版本2.2.11开始，Linux增加了“能力绑定设置”的特性，使能力的应用更直接有用。能力绑定设置是一组允许被系统中任意进程所拥有的能力（否则，只有init进程可以拥有这些能力）。如果某能力不在此绑定设置中，则无论任何进程都不可以使用它。例如此特性可用来禁止内核模块加载。从版本2.4开始，Linux通过修改prctl系统调用，允许一个实现能力的程序在由root用户转为非root用户时可以保留它现有的能力。很明显地，这对实现安全的后台服务程序非常有用。不足的是，Linux只实现了贴在进程上的能力，尚未实现贴在文件系统客体上的能力。

5. 日志系统

- Linux的日志信息的来源通常有内核、系统服务程序、用户程序以及网络。
- 系统日志守护进程Syslogd是各类日志信息的管理者，它作为启动文件的一部分在开机后自动执行。日志信息由系统统一存放在/var/log目录下，可以通过/etc/syslog.confd对Syslogd的行为进行配置。默认情况下，Linux缺省地将用户的登录信息记录到/var/log/wtmp（一个用户每次登录进入和退出事件的永久记录）中。
- 内核开辟了一个16kB的循环消息缓冲区，通过printk向其中写入不同级别的消息，共有8个级别（如紧急、错误、警告、调试等），级别附在消息首部。

Linux日志系统

- 系统中的另一个守护程序Klogd通过系统调用或虚拟文件系统(/proc)读取内核缓冲区，解释内核消息，并发送给 Syslogd。
- 应用程序(如login)通过库函数syslog直接向Syslogd发消息，可同时指定级别和消息类型。与内核消息不同的是，它的传递借助于UNIX域的套接字/dev/log，不涉及内核消息缓冲区。主机之间可以通过Syslogd传递日志信息，Syslogd接收这些消息并根据消息性质的不同进行分发，分别记入文件、显示在控制台、发送到指定的用户终端、送入命令管道或者发送给其他主机上的Syslogd。
- 缺省配置的日志文件是messages记录一般信息，secure记录与用户认证有关的信息。

Linux日志系统的基本结构

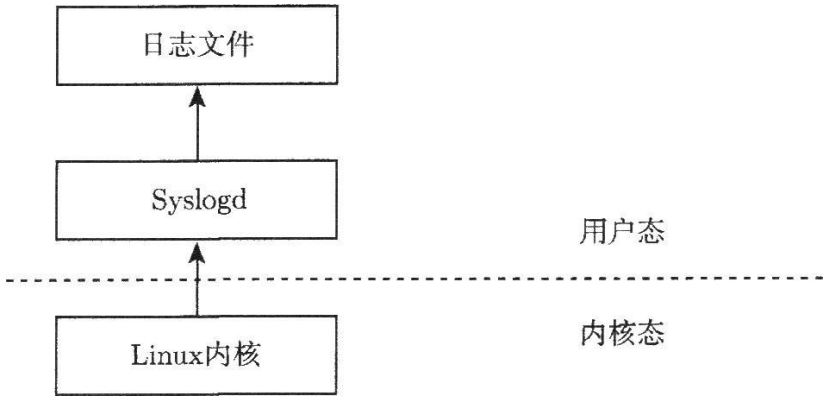


图6-5 Linux日志系统的基本结构

- 由图6-5可知，Linux的日志系统是独立于Linux内核的，它是在用户态运行的守护进程，不具备安全性。
- 因此，如果某入侵者取得了一定权限，他可以按照Syslogd的工作方式抹掉所有的日志信息和入侵记录。

- 总的来说，Linux已有的日志功能是不完善的。
- (1)它们只在各自关心的范围做记录，不能全面表达系统的活动。
- (2)日志记录的内容和格式都不同，因而不利于程序的自动分析。
- (3)文本格式的日志文件所含信息量少，存储效率低。
- (4)日志文件可靠性差，容易被伪造和篡改。

6．防火墙机制

- Linux2.0系列中的ipfwadm和2.2系列中的ipchains都在内核的网络层提供了防火墙功能。
- 从Linux2.4开始，又实现了一个称为netfilter的网络层数据包过滤框架，其不仅具备良好的结构、完整的防火墙功能，还实现了许多新功能，如完整的动态NAT(ipchains中实际是多对一的“地址伪装”)、基于MAC及用户的过滤、真正的基于状态的过滤（不再是简单地查看TCP的标志位等）、包速率限制等。
- Netfilter框架为每种网络协议（如IPv4、IPv6等）定义一套钩子函数，这些钩子函数在数据报流过协议栈的几个关键点被调用。在这几个点中，协议栈将把数据包及钩子函数标号作为参数调用netfilter框架。
- 例如，IPv4定义了5个钩子函数。

IPv4定义的5个钩子函数

(1)NF_IP_PRE_ROUTING。

(2)NF_IP_LOCAL_IN。

(3)NF_IP_FORWARD。

(4)NF_IP_LOCAL_OUT。

(5)NF_IP_POST_ROUTING。

- 数据报进入系统进行IP校验以后，经过函数NF_IP_PRE_ROUTING进行处理，然后就进入路由代码，路由代码判断该数据包是需要转发的还是发给本机的。
- 若该数据包是发给本机的，则该数据包经过函数NF_IP_LOCAL_IN处理以后再传递给上层协议；
- 若该数据包应该被转发，则它被函数NF_IP_FORWARD处理，经过转发的数据包经过函数NF_IP_POST_ROUTING处理以后，再传输到网络上。本地产生的数据经过函数NF_IP_LOC AL_OUT处理后，进行路由选择处理，然后经过NF_IP_POST_ROUTING处理后再发送到网络上。

- 内核的任何模块可以对每种网络协议的一个或多个钩子函数进行注册并实现连接，这样当某个数据包被传递给netfilter框架时，内核能够检测到是否有模块对该协议和钩子函数进行了注册。若注册了，则调用该模块注册时使用的回调函数，这样这些模块就能够检查（可能还会修改）该数据包、丢弃该数据包及指示netfilter将该数据包传入用户空间的队列。
- 排队的数据包可以被传递给用户空间异步地进行处理。一个用户进程能检查数据包，修改数据包，甚至可以重新将该数据包通过内核中的同一个钩子函数输入到内核中。所有的包过滤/NAT等都可以基于该框架实现，这样内核网络代码中不会到处都是混乱的修改数据包的代码。目前，netfilter框架在IPv4、IPv6及Decnet网络协议栈中已经实现。

实例：Ubuntu系统的防火墙

- sudo ufw status
- sudo ufw allow ssh
- sudo ufw deny ssh

6.2 数据库安全技术

6.2.1 传统数据库安全技术

- 对于数据库而言，安全性主要包括三个方面：机密性、完整性、可用性。确保机密性、完整性、可用性分别指的是防止、检测、阻隔信息的非法泄露、非法篡改以及系统拒绝信息的访问服务。
- 面对可能的安全威胁，数据库的安全保护要针对各种因素和保护需求来制定防范措施，建立安全模型，以保护其资源尤其是存储的数据免受无意或有意的非授权访问攻击。
- 保护需求主要为以下几类：
- **(1)防止不适当访问。**只对授权的合法用户授予访问权限。

- **(2)分级保护**。依据数据敏感级别设立多级进行保护。在包含敏感、非敏感混合数据的数据库中，就需要严格控制对敏感数据的访问请求，只能是经过授权的用户有权进行某些操作，并且不允许其权力传播或转让。
- **(3)防止推断性攻击**。防止从非保密信息中获得保密数据，尤其对于统计型数据库。
- **(4)数据库的完整性**。该需求涉及防止更改数据内容的非授权访问，以及病毒，蓄意破坏，或是系统级错误及故障等。这主要由DBMS通过系统控制以及备份、恢复机制执行并完成保护工作。
- **(5)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。
- **(6)数据的语义完整性**。确保对数据在允许的范围之内修改，以保持数据的一致完整性。
- **(7)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，根据结果进行分析与追踪。

保证数据库管理系统安全的基本方法

1)用户身份认证

- 用户身份认证是安全系统的第一道防线，目的是防止非法用户访问系统。除口令控制外，用户身份认证还可以采用比较复杂的计算过程和函数来完成。而智能卡技术、数字签名技术和生理特征（如指纹、体温、声纹、视网膜纹等）等认证技术的迅速发展也为具有更高安全要求的用户身份认证提供了实用可行的技术基础。

2)存取控制

- 数据库的存取访问控制机制是定义和控制用户对数据库数据的存取控制权限，以确保只授权给有资格的用户访问数据库的权限，并防止和杜绝对数据库中数据的非授权访问，通常在认证已成功的基础之上进行。第四章介绍了常见的访问控制技术，这些技术都可以应用于数据库的安全控制。

- “计算机信息系统安全保护等级划分准则”(DB17859-1999)的数据库系统安全策略是以数据库系统内核强制访问控制（以系统的主、客体敏感标识为基础）、自主访问控制、责任保证构成的多级安全控制策略。概括来说，就是以多级强制访问控制为核心的系统安全策略。

3)数据加密

- 数据加密是防止数据库中数据泄露的有效手段，与传统的通信或网络加密技术相比，由于数据保存的时间要长得多，对加密强度的要求更高。而且，由于数据库中数据是多用户共享，对加密和解密的时间要求也更高，要求不会明显降低系统性能。

4)审计追踪与攻击检测

- 审计功能在系统运行时，自动将对数据库的所有操作记录在审计日志中。攻击检测系统则是根据审计数据分析检测内部和外部攻击者的攻击企图，再现导致系统现状的事件，分析发现系统安全的弱点，追查相关责任者。

6.2.2 外包数据库安全

- 信息系统的外包服务，是指组织将其信息系统的开发、管理或维护的工作外包给专业性外部组织来完成的一种运行模式。
- 随着信息技术在组织中发挥的作用日益增加，信息系统规模越来越大，涉及的技术也越来越复杂，外包服务把组织自己不擅长的东西（非核心业务）交给专业的外部组织去做，将主要精力集中于核心业务，从而优化资源配置，降低软硬件投资成本，提高组织机能和效率，充分发挥自身核心竞争力并增强自身应变能力。
- 日益完善的计算机网络环境为这种服务提供了可能。

外包数据库运行模式带来的最大挑战就是安全问题

- 2002年，美国加州大学Irvine分校Hakan Hacigumus提出了作为服务的数据库的概念，即外包数据库(out sourced database)，并基于Internet构建了一个名为NetDB2的原型系统。
- 在外包数据库系统中，组织将自己的数据库业务外包给外部数据库服务器运行，外包服务提供者为数据拥有者及数据库用户提供远程的数据库创建、存储、更新与查询服务。**这种运行模式带来的最大挑战就是安全问题。**
- 由于数据存储在非完全可信的第三方服务器中，因此，外包数据库系统的安全机制不但要防止来自外部的恶意攻击，而且要充分考虑来自服务提供者本身的恶意操作。为了实现对外包数据库的安全控制，必须根据其结构特点，采取特定的安全机制。

外包数据库系统的特定安全机制

- 外包数据库系统的安全机制包含了数据库安全的一般机制，如身份鉴别与认证、多级安全访问控制、用户跟踪与审计、数据备份与恢复等。同时，根据外包数据库系统的结构特点，还需采用特定的安全机制。
- 目前，对外包数据库系统安全机制的研究已经取得了一些成果，主要包括：数据库加密、密文数据库查询、隐私保护、数据完整性验证、数据库版权保护等。

1. 数据库加密技术

- 数据库加密机制可以分为紧耦合加密与松耦合加密，分别对应于库内核加密与库外加密方式。外包数据库系统中，由于数据库服务器非完全可信，加密解密都应在客户端完成。因此，库外加密方式正受到越来越多的关注。数据库加密的粒度一般分为四种：表、属性、记录和记录属性值。

2. 密文数据查询策略

- 常用的密文数据库查询有两种策略：一种是不用解密而直接操作密文数据；另一种是分步查询。
- **直接操作密文数据**的应用场合包括数据库的秘密同态加密、数据库的序列加密等。采用序列密码算法把表数据与随机数发生器所产生的随机数进行位异或加密。查询时，把查询词与密文异或后与随机数比较，以确定文件中是否包含该词。
- **分步查询**是一种更具有实用价值的密文查询策略，也是目前研究的热点。这种方法一般需要进行查询分解，先对密文数据进行范围查询，缩小解密范围，快速解密后再执行精确查询。优化的查询计划还将一次查询任务分解为客户端和服务器的多次交互，以得到最优的查询树。

3．数据库隐私保护

- 隐私被认为是个体来控制自己的信息的权力。越来越多的因故意或疏忽造成的信息泄露的事件，使人们对数据库中的隐私保护问题日益重视。
- 外包数据库的隐私保护包括两个方面：一方面是对以明文存储的隐私数据内容的保护；另一方面是对用户查询行为及查询结果的保护，即保密信息检索(private information retrieval, PIR)。

1)基于推理控制的隐私内容保护

- 目前常用的推理控制方法可以分为4种：语义数据模型方法、形式化方法、多实例方法和查询限制方法。

2)保密信息检索

- 保密信息检索的需求源自外包数据库服务器的不可信。用户向服务器提交查询请求，同时不希望对服务器公开他获取的查询结果。主要有两类：多服务器方式和单服务器方式。

4. 数据完整性验证

- 外包数据库系统的数据完整性要求数据库内容及其在网络中的传输具有正确性、一致性与有效性，以确保接收到的数据库内容是真实有效的，并且在传输过程中没有被攻击者插入、篡改、伪造、重排等。在电子商务、电子政务中，数据完整性是最基本、最重要的安全要求。
- 外包数据库的数据完整性要求的特点是，既要考虑外部攻击，又要考虑服务器自身的安全性。
- 实现数据完整性的主要措施就是增加攻击者所不能控制的冗余信息。外包数据库中，根据系统结构模式不同，所采取的完整性验证机制也不相同。
- 对于统一客户端模式，采用加密 Hash函数，对数据库记录进行数字签名。面对于多查询用户模式及多主模式，有文献提出了一种基于公钥机制的压缩RSA(condensed RSA)算法的批量数字签名方案，适用于多查询用户模式的数据完整性控制，也还有其他解决方案，但并不实用。

5. 外包数据库版权保护

- 在外包数据库服务模式中，数据经过若干年的积累，往往蕴含有巨大的社会价值与经济价值，成为宝贵的数据资源。而由于数据库服务器由第三方提供，数据库的物理文件可以轻易地被拷贝而造成资源盗用，因此，数据拥有者对数据库实施版权保护的需求日益迫切。
- 利用数字水印实现对外包数据库的版权保护，具有较高的研究和应用价值。目前，数据库水印技术在水印宿主数据类型扩展、XML数据水印、数据库脆弱性水印、数据库数字指纹等研究方向已取得一定的进展。随着研究的深入，数据库水印将获得更好的透明性、鲁棒性与可用性，其在外包数据库安全领域的应用空间必将得到进一步拓展。

6.2.3 云数据库 / 云存储安全

- “云存储(cloud storage)是在云计算概念上延伸和发展出来的一个新的概念，是指通过集群应用、网格技术或分布式文件系统等功能，将网络中大量各种不同类型的存储设备通过应用软件集合起来协同工作，共同对外提供数据存储和业务访问功能的一个系统。”
- 还有一个定义，是全球网络存储工业协会(SNIA)给出的云存储的定义：“云存储是通过网络提供可配置的虚拟化的存储及相关数据的服务。云存储的内涵是存储虚拟化和存储自动化”。SNIA给出的定义更多的是站在使用和服务的角度来说的。
- 这两个定义都有几个共同的特征：首先它是基于网络的；其次它是可以配置、按需分配的；同时它是一种虚拟化的存储和数据管理。

云存储模式也有一定的安全问题

- **首先是身份认证和访问控制问题**，由于鉴别的措施太弱，导致数据或者存储的信息可能会被假冒和窃取。
- **第二个风险是数据存储和传输的保密性问题**。企业的经营数据和个人的数据，如果放到云端存储以后无法保证信息在存储和传输过程中的保密性，可能产生商业信息和隐私泄露的问题。
- **第三是数据隔离问题**。云计算最重要的一个核心就是云的虚拟化问题，对不同的云用户来说，云存储的系统是一个相同的物理系统，不再像传统网络一样有物理的隔离和防护边界，所以就存在虚拟系统被越界访问等无法保证信息隔离性的问题。
- **第四是应用安全问题**。对于运行在云存储平台之上的云应用，如果其本身未遵循安全规则或存在应用安全漏洞，就可能导致云存储数据被非法访问或破坏等问题。

云存储安全机制

- 云的安全问题从本质上来说更多的是信任问题，云安全的核心是密码技术和加固技术，通过采取大量的密码技术的加固技术来向用户提供可信任的安全云。云存储安全机制简单归纳为3个方面
- **(1)云存储平台安全机制**是保护整个云存储平台系统自身的安全，其中主要有两个技术，第一个是**密码技术**，保证所有的程序和应用系统的完整性，提供基于PKI的强身份鉴别以及存储节点的透明加密。另一个是**加固技术**，采用主动防御技术保障服务器、主机的安全性；采用操作系统内核加固实现对计算 / 存储节点、虚拟主机的保护，免遭病毒、木马攻击；实现主机虚拟化技术，实现对虚拟主机的保护，实现数据隔离。

- **(2)云存储管控安全机制**主要解决安全管理的问题，包括对云节点服务器密钥的统一管理、密钥生命周期的可控性、云数据接口／云客户端密钥的自主性等。从管理安全的角度来说，云存储的管理需要满足“相互约束、相互独立”的三员管理。
- **(3)云存储应用安全机制**主要从以下几方面来实现：存储加密（在访问云入口加密，保障传输的安全性，只有授权用户采用访问数据）、备份加密（在云数据中心采用专用算法，对文件和数据库数据加密、备份、保存）、交换加密（对交换数据采用数字信封方式加密实现）、身份认证与访问控制（接入前，基于PKI和RBAC机制）、接口安全（提供多种接口模式，采用密码技术实现安全）、手机安全（采用软件与SD卡方式保障手机模式的安全）以及云端数据库（对云端数据库采用加密存放）。

6.3 可信计算技术

6.3.1 概念和基本思想

- 人们已经认识到，相当多的安全隐患来自于计算机系统终端，因此必须提高主机的安全性，以从终端源头控制绝大多数不安全因素。对于计算机系统来说，只有从芯片、主板、操作系统开始，综合采取措施才能提高其安全性，正是这一技术思想推动了可信计算的产生和发展。
- 可信计算(trusted computing)是一种信息系统安全新技术。可信计算的思想源于人类社会，是把人类社会成功的管理经验用于计算机信息系统和网络空间，以确保计算机信息系统和网络空间的安全可信。

可信计算中的可信定义

- ISO/IEC将可信定义为：参与计算的组件、操作或过程在任意的条件下是可预测的，并能够抵御病毒和一定程度的物理干扰。IEEE给出的可信定义为：计算机系统所提供服务的可信赖性是可论证的。
- 可信计算组织(trusted computing group，TCG)的定义为：一个实体是可信的，如果它的行为总是以预期的方式，朝着预期的目标。
- TCG认为，可信计算的总体目标是提高计算机系统的安全性，现阶段的主要目标是确保系统数据的完整性、数据的安全存储和平台可信性的远程证明。
- **TCG的可信计算技术思路是通过在硬件平台上引入可信平台模块TPM(trusted platform module)来提高计算机系统的安全性，这种技术思路目前得到了产业界的普遍认同。**

可信计算的基本思想

- 可信计算的宗旨是以可信计算安全芯片为核心改进现有平台体系结构，增强通用计算平台和网络的可信性。其**基本思想**是：

A. 首先在计算机系统中建立一个信任根，信任根的可信性由物理安全、技术安全与管理安全共同确保；

B. 再建立一条信任链，从信任根开始到硬件平台，到操作系统，再到应用。一级测量认证一级，一级信任一级，把这种信任扩展到整个计算机系统，从而确保整个计算机系统的可信。

- 信任是一种二元关系，它可以是一对一、一对多（个体对群体）、多对一（群体对个体）或多对多（群体对群体）的。

信任和信任的获得

- 信任不一定具有对称性，即A信任B不一定就有B信任A；信任可度量，即信任的程度可划分等级；信任可传递，但不绝对，而且在传播过程中有损失；信任具有动态性，即信任与环境（上下文）和时间因素相关。
- 信任的获得方法主要有直接和间接两种方法。设A和B以前有过交往，则A对B的可信度可以通过考察B以往的表现来确定。称这种通过直接交往得到的信任值为直接信任值。设A和B以前没有任何交往，这种情况下，A可以去询问一个与B比较熟悉的实体C来获得B的信任值，并且要求实体C与B有过直接的交往经验，称之为间接信任值。

TCG的信任链模型

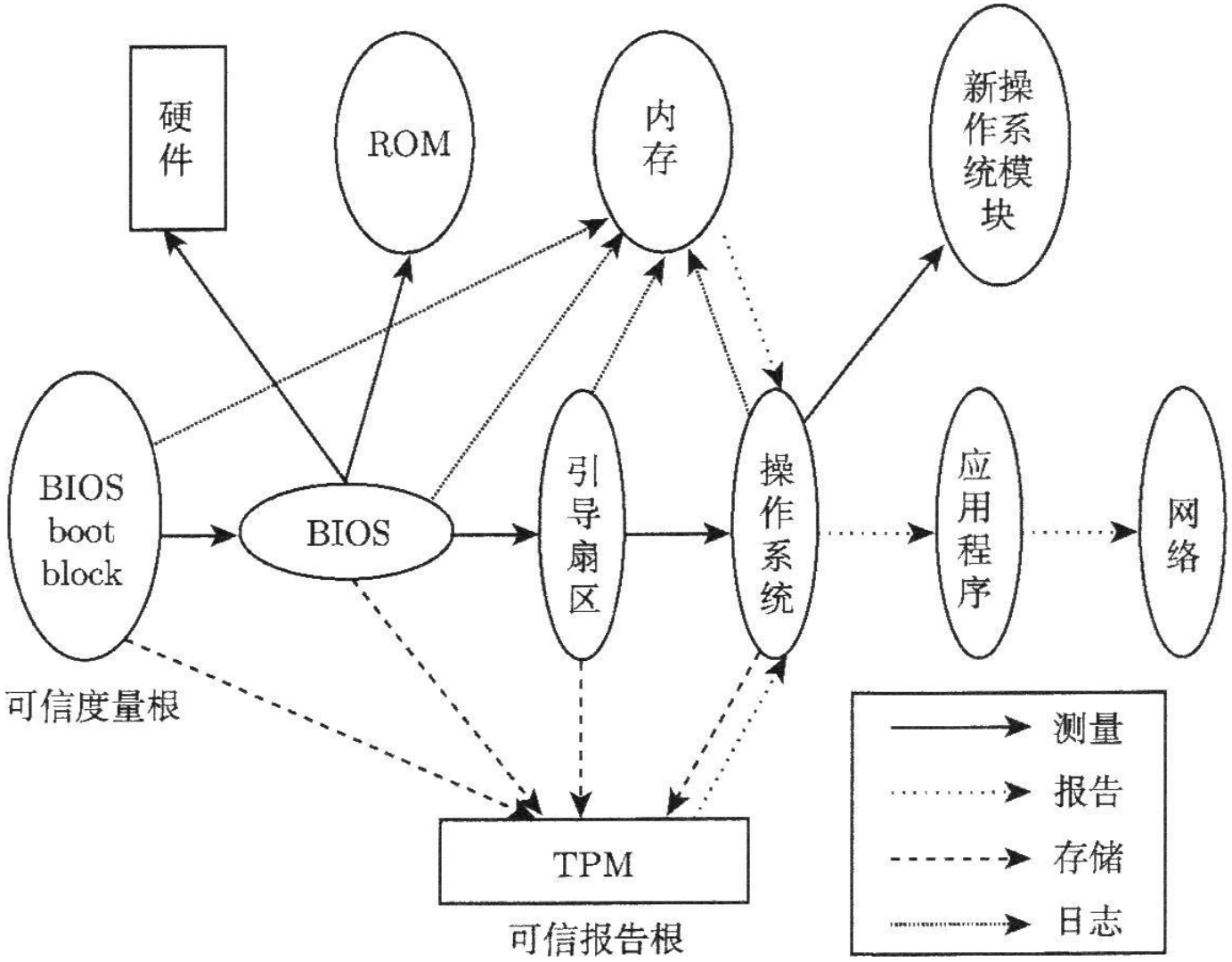


图6-6 TCG的信任链

6.3.2 TCG可信计算系统

- TCG在现有体系结构上引入硬件安全芯片TPM，利用TPM的安全特性来保证通用计算平台的可信。
- 可信平台模块TPM(trusted platform module，TPM)是一种SOC芯片，它是可信计算平台的信任根，其结构如图6-7所示。
- 它由CPU、存储器、I/O、密码协处理器、随机数产生器和嵌入式操作系统等部件组成，完成可信度量的存储、可信度量的报告、密钥产生、加密、签名、数据安全存储等功能。

TCG的可信计算系统结构

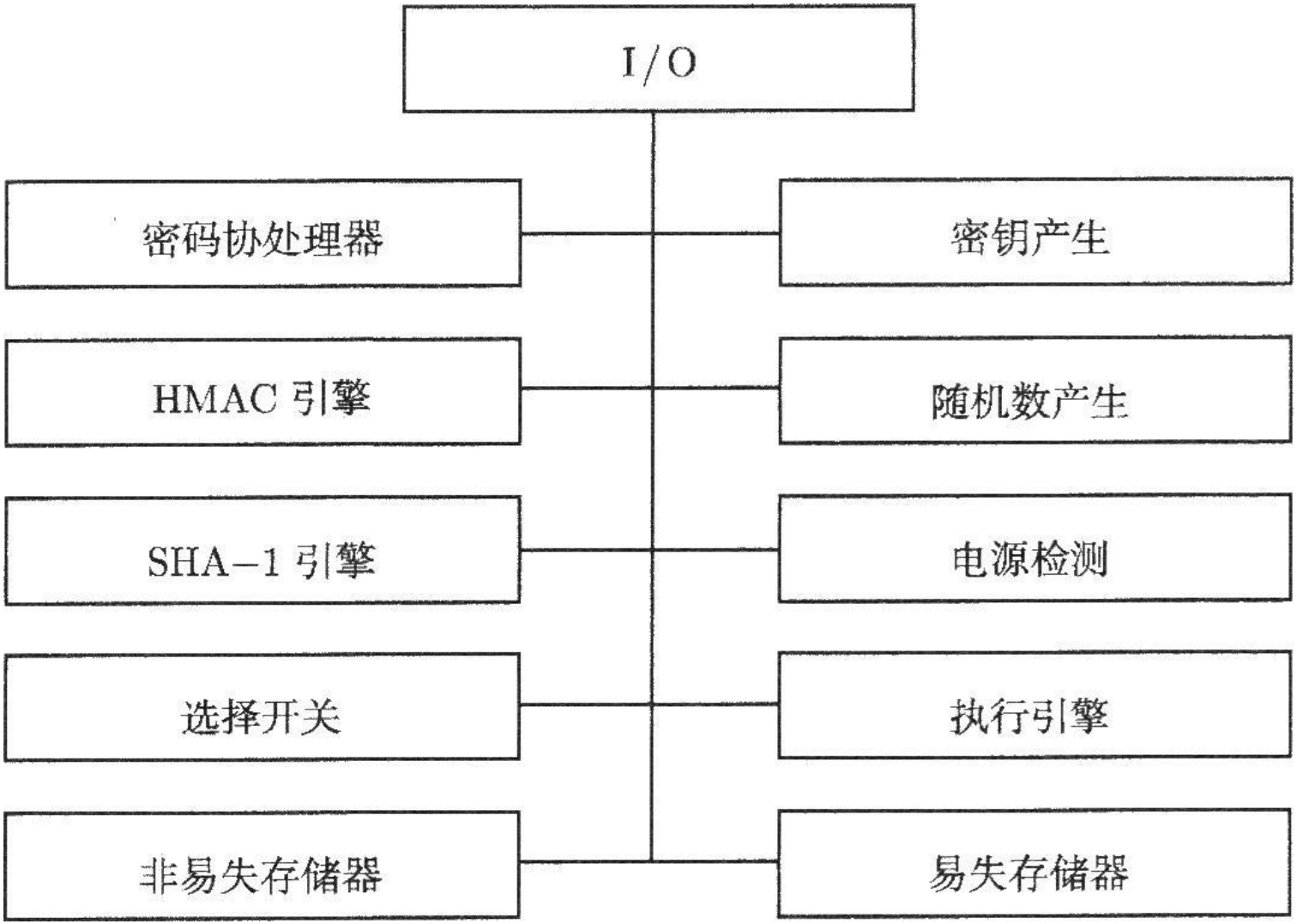


图6-7 TCG的可信计算系统结构

TCPA(Trusted Computing Platform Alliance)可信计算结构

- TCPA可信计算结构建立在常见的计算机系统结构之上，系统结构如图6-8所示。

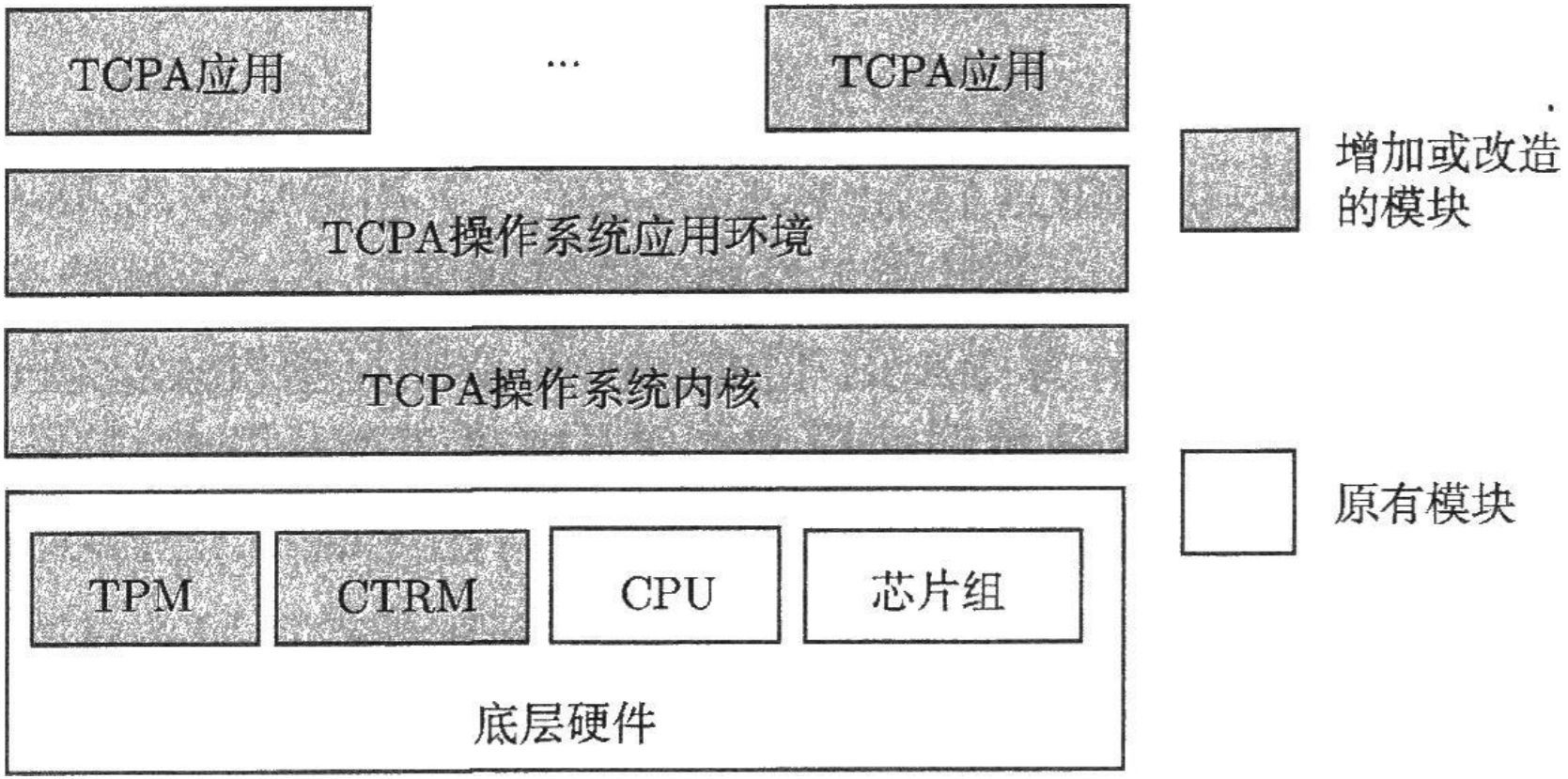


图6-8 TCPA系统结构

TCPA可信计算系统结构

- 它通过引入可信平台模块、核心根信任模块(core root trust module，CRTM)、TCPA操作系统及它们的连接、作用方式，支撑可信计算的应用。TPM是硬件安全结构的核心，具有生成加密密钥、高速数据加解密及保护BIOS和操作系统重要配置完整性的功能。CRTM负责初始化系统、认证BIOS、硬件配置等。TCPA操作系统能够利用可信计算硬件平台提供的功能进行系统中实体、资源的认证。
- 整个TCPA可信计算系统结构可以划分为三个层次：TPM、TPM软件栈(tpm software stack，TSS)和可信平台应用软件。TSS位于TPM与可信平台应用软件之间，它主要包括 TCPA操作系统部分(图6-8)，负责对可信平台应用软件提供可信计算支持，包括提供对 TPM的访问和操作、安全认证、密码操作调用和资源管理等功能。

第6章作业

- 作业
  - 3．什么是TCB?主要包括哪些成分？
  - 6．什么是最小特权原则？
  - 11．可信计算的基本思想是什么？

- 实践（自己研究，不考核）
  - 熟悉Windows2003的“本地安全策略”功能
  - 熟悉Ubuntu Linux系统的防火墙配置命令ufw