一种并行可复原可信启动过程的设计与实现*)

谭 良[1,2] 周明天[1]

(电子科技大学计算机科学与工程学院 成都 610054)[1] (四川师范大学电子工程学院 成都 610066)[2]

摘 要 操作系统可信性的建立是从整个计算机系统加电引导开始直至操作系统运行环境最终的创建,对任意一次可能降低操作系统可信性的执行代码操作都要进行一致性度量。本文基于可信计算联盟的规范,分析了基于TPM的可信引导过程,提出了一种新的可信引导过程:并行可复原可信引导过程—在主机CPU与TPM之间采用并行工作方式,并支持被验证组件代码的备份和恢复。然后利用通道技术设计和实现了这一引导过程。最后对该引导过程进行了安全和性能分析,分析表明该引导过程可以使计算机获得更高的安全保障,为进一步建立可信计算环境提供了基础。

关键词 可信计算,可信平台模块,完整性度量

Design and Implementation of a Parallel Recovery Trusted Startup Process Based on the Trusted Platform Module

TAN Liang[1,2] ZHOU Ming-Tian[1]

(School of Comp. Sci. & Engn., Univ. of Electronic Sci. & Tech. of China, Chengdu 610054)[1]

(Collese of Electronic Engineering, Sichuan Normal University, Chengdu 610066)[2]

Abstract For establishing the trustworthiness of the operating system, every program code of the operating system, which possibly reduces the trustworthiness, must be attest the integrality during the whole process form firstly switching on power to finally establishing the running environment in the computer system. According to the standards of the trusted computing group, based on the analysis to the trusted startup process of the common operator system, a trusted boot process, called a Parallel Recovery Trusted Startup Process (PRTSP), which takes a parallel working between CPU and TPM, and supports backup and recovery, has been put forward, and then designed and implemented by channel technology. Finally, we discuss the security and performance of the PRTSP. Based on the PRTSP, high assurance of system security has been gained, thus the basis for building the trusted computing environment has been provided.

Keywords Trusted computing, Trusted platform module (TPM), Attestation of integrity

1 引言

操作系统的可信不是凭空而来的。可信性的建立从整个计算机系统加电引导开始直至操作系统运行环境最终的创建,对任意一次可能降低系统可信性的执行代码操作都要进行一致性度量。分析一般的操作系统,如Linux、Windows的启动[1,2]过程可以看出,从操作系统的启动到用户通过应用进程对资源进行访问,整个过程呈现出阶段性特征。按照功能主要分为两个部分:1)操作系统的引导:从终端加电开始BIOS运行到操作系统运行前的整个过程,此阶段的特点是流程相对固定的,引导按照一定的次序进行;2)操作系统的运行:当内核开始运行后,进入多任务的调度,执行过程没有明显的先后顺序,但是内核会通过调度程序,在应用程序执行时将CPU控制权转交给它。本文专注于讨论操作系统的引导过程,只对引导过程中需要执行的程序进行完整性度量,因为它是操作系统动态执行环境可信的基础。

要保证操作系统的可信,就必须解决操作系统引导过程的可信;要解决引导过程可信,就必须解决初始信任的来源问题;并保证信任关系在这两个阶段中能够逐层传递。那么初始信任源究竟是软件还是硬件呢?单纯依靠软件来构筑系统的安全性是远远不够的,这一点早已被业界所认知,可信计算[3]的兴起正是以此为出发点,它是一种软硬件相结合的技术,通过在平台内部引入可信硬件设备作为可信根,为建立可信计算环境提供有效途径。本文基于可信计算联盟的规范,详细分析了基于TPM的可信引导过程,提出了一种新的可信启动过程:并行可复原可信启动过程,并利用通道技术设计和实现了这一过程。最后进行了安全和性能分析。该引导过程可以使计算机获得更高的安全保障,为进一步建立可信计算环境提供了基础。

2 相关技术和研究现状

AEGIS[4]是一个在FreeBSD系统上实现了从系统加电到应用程序层逐级安全验证的原型系统,它将系统启动的过程分为五个级别,BIOS的核心代码构成了第零级别,这部分代码被安全存放并被无条件信任。按照系统启动的流程,在将控制传递到下一个级别代码前,首先对代码进行完整性验证,验证通过才能将控制向下传递,依此类推,若某一个环节的验证失败,则强制通过预先的备份数据抓进行恢复。由于

*)基金项目:国家863宽带VPN项目863-104-03-01课题资助;2003年度四川省科技攻关项目03GG007-007支持。谭 良 博士,主要研究方向为信息安全、中间件;周明天 教授,博士生导师,主要研究方向为网络计算,信息安全,分布并行处理。

在启动中和启动后实施了严格的验证，系统可以在预先设定的状态中执行，有效杜绝了恶意程序对系统造成的破坏。该系统的实现需要添加扩展的ROM，并且验证失败时恢复策略带有强制性。作为一个原型系统，思想是值得借鉴的。文[5]已讨论了类似的问题。

1999年由Compap、HP、IBM、Intel和Microsoft牵头组成了TCPA联盟，专注于从计算平台体系结构上增强其安全性，2003年3月改组为TCG[3]，逐渐成为研究热点[6~18]。TCG规范提倡采用一个单独的安全模块作为信任根，这是一个软硬件相结合的子系统，该子系统被设计成能够度量、存储和报告系统可信赖属性的模块，是建立信任串的起始点。TCG规范旨在提供开放的、平台无关的标准从而被应用到不同的平台设备中，TCG中的信任模型是以TPM(Trusted Platform Module)硬件模块为基础的。TPM是一个可独立进行密钥生成、加解密的装置，内部拥有独立的处理器和存储单元，可存储密钥和敏感数据，为各种计算平台提供完整性度量、数据安全保护和身份认证服务。但TCG只为系统引导阶段建立初始信任过程中参数的度量和报告制定了标准，但对于如何实施未作探讨。

3 基于TPM的可信引导过程分析

3.1 可信引导的层次性

可信引导过程必须保证以下三点。首先，保证信任是逐层传递，当前一个可执行实体被度量并验证是可信并执行后，其转移控制权至下一个可执行实体之前，必须先对其进行度量，验证可信后方可转移控制权，信任从而传递至下一个执行实体。其次，所有在可信串的建立过程中涉及到的度量和验证方法不能采用软件实现，而是依赖于TPM的实现。所有的度量和验证调用将最终由TPM验证模块来完成。最后，可信串是建立在逐层的度量和验证基础上的，这种验证要求由TPM来保证重要秘密数据的完整性和保密性。在可信串传递过程中涉及的所有重要数据、需要预存的验证码都必须由TPM来保存，不能使用可移动的存储装置或PC机的系统内存。并且这些数据在使用过程中不能脱离TPM。TPM也不能提供访问这些数据的外部调用接口，以保证这些重要数据的秘密和可信。

可信引导过程如图1所示，信任串贯穿系统从加电启动到操作系统动态执行环境建立的全过程。在信任串传递过程中，最低层是0层，0层包含TPM，这一层的完整性和可信性被假设为总是受到保证的；1层则包含了BIOS引导代码；2层是19号中断程序；3层包含操作系统引导程序Boot；4层包括操作系统装载程序Loader，它们驻留在可引导的设备上，并负责装载操作系统内核；5层包含操作系统内核；6层包括操作系统的Iinit进程，再上层是可信应用程序。

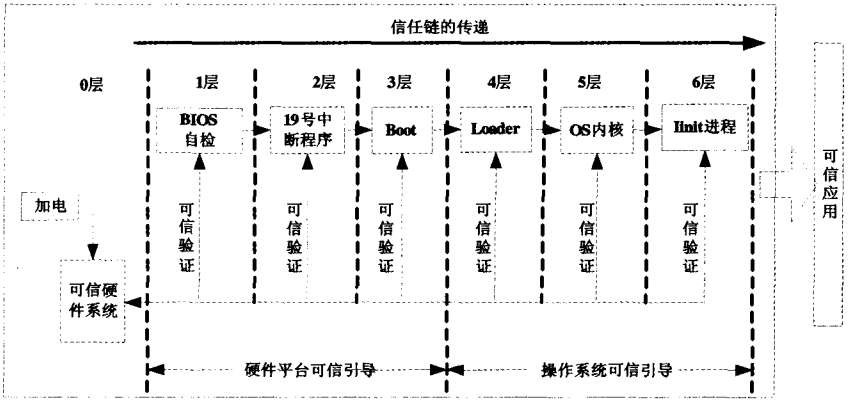

图1 基于TPM的可信引导过程

整个可信引导过程总体来说应该分为两个主要阶段：硬件平台的引导阶段和操作系统的启动阶段。硬件平台的引导包括从平台加电、BIOS运行、到BIOS将控制权交给Boot之前，这期间主要是保证硬件环境的可信。操作系统的启动阶段从主引导区调入操作系统装载程序一直到操作系统内核运行完毕，并运行Init进程之前。该阶段主要保证系统的启动过程和操作系统内核的可信。

3.1.1 硬件平台可信引导过程

硬件平台的引导定义为从系统加电开始到BIOS运行完毕并将控制权交给主引导区之前。该阶段应该保证整个硬件平台加电引导过程的可信。如图2，具体过程如下：

1)用户在可信硬件提供的USB接口中插入开机身份卡。

2)硬件平台加电，TPM首先复位并进行初始化。

3)TPM对用户的开机身份卡进行认证。如果通过开机身份认证，则进入可信引导过程(否则，停机或进入普通引导过程)。

4)TPM从硬件平台中取出BIOS自检程序，TPM可信测量程序对BIOS中的自检程序进行验证。如果摘要值匹配成功，则将控制权交给硬件平台的CPU。

5)硬件平台的CPU将CS段寄存器设置成FFFFH，清零所有其他寄存器，然后执行CS:IP(FFFF:0000地址也就是BIOS地址F000:FFF0)处指令进入硬件平台的自检。

6)硬件平台自检成功后，并不立刻执行int19引导系统，而是将控制权转移到TPM，由TPM负责对19号中断程序进行验证。如果通过验证，立刻执行该中断程序引导系统。根据CMOS中预设的顺序到软盘，硬盘，光盘等寻找引导扇区中的boot程序。

7)找到boot程序后，并不立刻将boot读到内存0000:7c00h处引导操作系统启动。而是由TPM可信测量程序对Boot进行验证。如果验证成功，将Boot读到内存0000:7c00h处，再将控制权转交给硬件平台的CPU运行Boot程序，进入操作系统引导阶段。

3.1.2 操作系统的可信启动过程

操作系统可信启动阶段被定义为从主引导区开始执行到Init进程开始执行之前。此过程包括主引导区的执行，操作系统装载程序的执行，安全操作系统内核的装载，内存模式设

置程序的执行四个步骤。该阶段应该保证操作系统本身启动加载过程的可信。

在操作系统加载和启动流程中，由于 Boot 已经被 TPM 认证过，因此主引导区在执行时已经是可信的，信任从它开始向后传递，图 3 显示了具体过程。

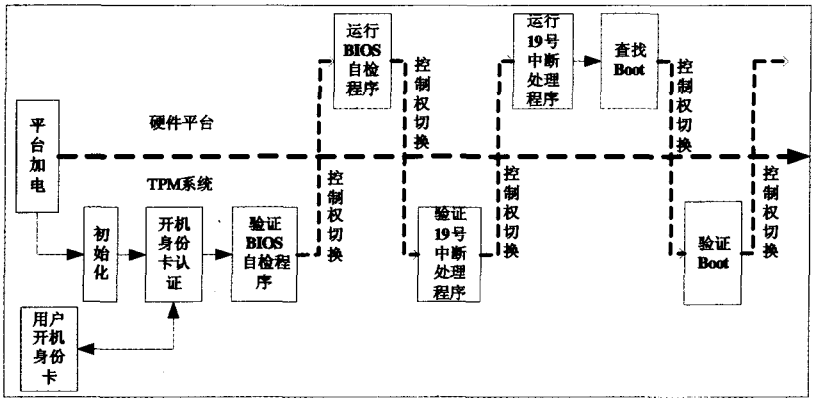

图 2　硬件平台的可信引导过程

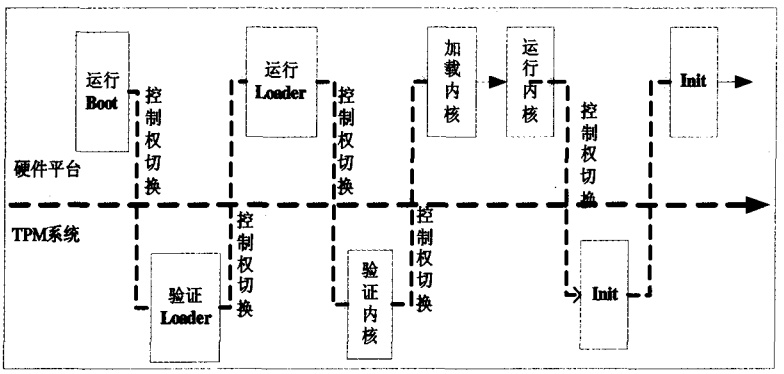

图 3　操作系统的可信引导过程

3.2　可信引导过程中需要解决的问题

从 3.1 节的分析可以看出，由于在计算机硬件层引入了可信硬件 TPM，与普通系统的引导流程相比，整个可信引导过程要复杂得多，如运行流程控制权需要在硬件平台和 TPM 之间切换、完整性验证等。因此，可信引导过程要得以实现，需要解决以下问题：

• 主机 CPU 与 TPM 的工作模式。可信引导过程不是一个简单的顺序过程，不能采取简单顺序串的方式度量，其中有很多是并行过程。通过分析发现，整个过程由 2 种基本的模式组合而成。(1)单组件过程 SCP(Single Component Procedure)：该过程最大的特点就是过程的行为仅与某一个组件(m)相关，例如 Boot、Loader 等。对于 SCP 过程，只需要度量该过程唯一组件的完整性就可以。(2) 多组件过程 MCP (Multiple Component Procedure)：该过程由一系列组件($\langle m_1, m_2, \cdots, m_n \rangle$) 组成，每个组件按预先的顺序获得系统的控制权，整个过程的完整性不仅与该过程的所有组件完整性相关，也与组件获得执行权的顺序相关。如：BIOS 自检程序、操作系统内核等。显然，为了加快启动过程，主机 CPU 与 TPM 之间的工作模式应该采用并行方式，信息的输入输出方式应该采用 I/O 通道方式。特别是多组件过程，主机 CPU 与 TPM 采用并行 I/O 通道方式可以提高验证和运行速度，并降低备份和恢复的粒度。

• 可信测量程序。启动过程中，TPM 需要完成对各个组件的完整性校验，在此过程中需要设计“可信测量程序”，因此，需要考虑可信测量程序的体系结构和功能模块。

• 备份和恢复。在进行完整性校验的过程中，如果发现被破坏的组件，则不能将运行控制权交给该模块。正确的方法是用先前备份的模块替换该模块，该模块才能获得控制权被执行。

4　一种并行可复原可信启动过程的设计与实现

并行可复原可信启动过程(PRTSP：Parallel Recovery Trusted Startup Process)是指主机 CPU 与 TPM 之间的工作方式采用并行方式、主机 CPU 与 TPM 之间的输入输出方式采用 I/O 通道方式，并支持备份和恢复的一种新的引导过程。为什么主机 CPU 与 TPM 之间的输入输出方式采用高速通道呢？一方面，通道方式可支持主机 CPU 与 TPM 之间并行工作方式，输入输出速度快；另外一方面，将 TPM 看成是外设具有较强的灵活性、可扩展性，方便升级和替换。因此，要实现 PRTSP，首先必须设计主机 CPU 新 I/O 指令，主机 CPU 需要新 I/O 指令将需要验证的组件快速传递给 TPM。其次，由于主机 CPU 与 TPM 之间的输入输出方式采用 I/O 通道方式，需要设计 TPM 的中断处理程序。最后，还要考虑可信测量程序、备份和恢复等。

4.1　主机 CPU 和 TPM 之间的通道方式

图 4 是主机 CPU、通道和 TPM 之间的连接示意图。通道的主要硬件包括寄存器部分和控制部分。寄存器部分有：数据缓冲存储器、主存地址计数器、传输字节数计算器、通道

命令字存储器、通道状态字存储器。控制部分有：分时控制、地址分配、数据传送、数据格式变换等逻辑。

CPU 新增的 I/O 指令如下：

SEND_CODE_TO_TPM[R/M], n (1)

RECEIVE_STATUS_FORM_TPM[R/M] (2)

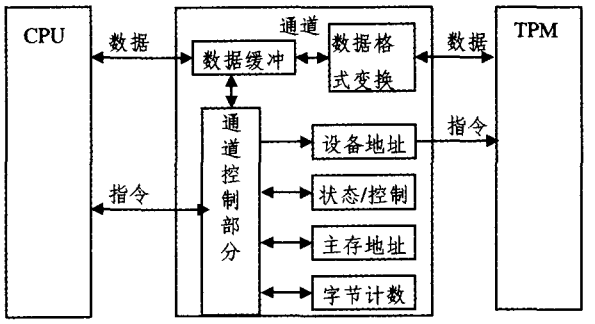

图 4 主机 CPU、通道和 TPM 之间的连接

对于指令(1)，SEND_TPM_CODE 为操作码，实现向 TPM 传递需要验证的代码。该指令需要两个操作数，第一个操作数表示需要验证代码在内存中的首地址，可以采用寄存器寻址和存储器寻址方式，可以使直接寻址，也可以是间接寻址。第二个操作数 n 指明代码的长度。对于指令(2)，RECEIVE_STATUS_FORM_TPM 为操作码，实现从相应的缓冲区中取回 TPM 的验证状态码。值得注意的是，这不是两条普通的 I/O 指令，是两条广义指令。在多用户或多任务系统中，该指令属于特权指令，一般用户程序不允许使用这些指令。如果在用户程序中需要与 TPM 进行输入输出操作，必须使用该指令进入操作系统，通过操作系统的管理程序来使用 TPM。下面阐述该指令的执行过程。

1)在处于就绪状态的进程中使用访管指令进入管理程序，由 CPU 通过管理程序组织一个通道程序，并启动通道。如图 5。值得注意的是，处于就绪状态的进程在获得 CPU 后还不能立刻运行，必须将该进程对应的程序代码经通道传给 TPM 验证，验证通过后方可被调度执行。将这种通过验证的进程所处的状态称为“就绪可执行”状态。

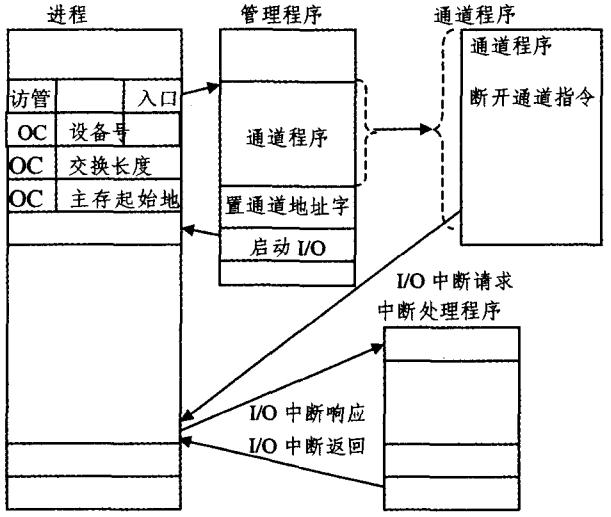

图 5 通道完成一次数据传输的主要过程

2)通道执行 CPU 为它组织的通道程序，完成指定的数据输入输出工作。从图 6 可以看出，通道执行通道程序与 CPU 执行引导程序是并行进行的。

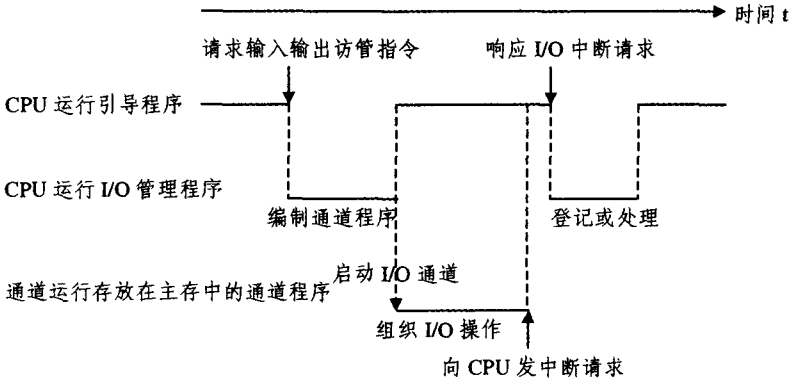

图 6 通道程序、管理程序和引导程序之间的执行时间关系

3)通道程序结束后向 CPU 发送中断请求。CPU 响应这个中断请求，调用管理程序对输入输出传输工作进行登记或异常处理。

4.2 TPM 中断程序的设计与实现

TPM 接受到通道传递过来的需要验证的数据，立即调用相关的可信测量程序模块对该代码进行验证。验证结束后，需要将验证的结果返回给主机 CPU。该过程描述如下：

- TPM 验证代码，获得验证结果；
- TPM 产生中断信号并将中断信号传递给通道；
- 通道将该中断信号转发给 CPU；
- CPU 响应该中断信号，执行 TPM 中断程序；
- 根据验证的结果，设置被验证的引导进程的状态。如果验证通过，则将该进程置为“就绪可执行”状态，并将该进程放入“就绪可执行”队列，等待调度。否则，停止引导过程。

在此过程中，CPU 需要执行 TPM 的中断程序。该中断程序如下：

算法：

1、关中断；
2、启动通道；
3、检查验证结果，如果验证通过，则转到 5；否则，执行下一步；
4、启动恢复模块，将存储在 TPM 中的备份组件替换被损坏的组件；
5 设置被验证的引导进程的状态标志，并根据状态标志将该引导进程放入不同的队列；
6、关闭通道；
7、开中断。

图 7 TPM 中断程序算法

为 TPM 中断程序选择一个保留的中断号，存入中断向量表。

4.3 可信测量程序的体系结构和功能模块

驻留在 TPM 中的可信测量程序由传输管理、验证模块、备份和恢复模块、密钥管理模块、文件管理模块等组成。图 8 为可信测量程序的体系结构。

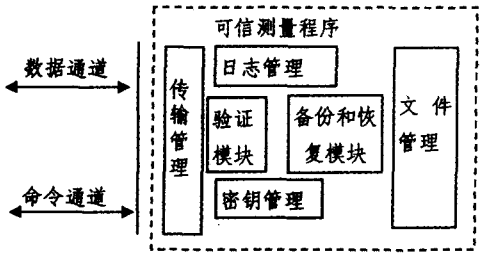

图 8 可信测量程序的体系结构

• 传输管理模块。注意 TPM 不仅需要与 CPU 通信，还需要外设通信，因此传输管理模块包含有与读卡器连接的串口、与主机 CPU 相连的总线接口和 USB 接口的驱动。可信测量程序通过这些驱动来进行数据传输和实现对主机的信息传递。

• 验证模块。该模块负责对主机 CPU 和外部设备传递过来的程序(信息)进行验证。验证算法包括：SHA. 1、MD5、RAS 等。

• 密钥管理模块。该模块负责对相关的验证密钥进行管理，确保这些密钥被安全存放。

• 文件管理模块。该将各种用户数据按照设计的文件结构格式存储在 Flash 中，并设置了一定的安全访问策略和机制，确保数据存储的安全和访问的快速便捷。

• 备份和恢复模块。在系统第一次引导过程中，TPM 通过运算得到各组件的验证码，并将这些组件和相关的验证码存入 TPM 中。之后的每一次启动，TPM 在验证过程忠，如果发现有破坏的组建，就将事先存在 TPM 中的组件替换被破坏了的组件。

5 安全性和性能分析

5.1 安全性分析

从整个过程中可以看出，只有通过破坏计算机系统的物理安全，即对 0 层中 TPM 进行修改，才能破坏整个过程。而这可通过增强物理安全和加大管理措施得以保障[19,20]。如果信任根的安全得到了保证，整个可信引导过程的安全性也就得到了保证。

信任根的安全体现在以下几方面。首先，为了证明 TPM 的身份是合法的，可以用公钥证书体系来增强整个验证过程的安全性。将 TPM 和由 CA 签署的身份证言证书绑定(也可以直接就是一个私钥，称为背书密钥 EK)，通过使用代表它身份的密钥签名数据来表明它自身是可信赖的。签名密钥只有 TPM 自己知道，并且是公钥对中的私钥。

其次，要保证引导组件及其验证码在可信测量模块中的存储是安全的。那么验证码可以不仅是一个哈希值，而是以证书 C 的形式存在，证书中包括一个唯一的部件标识符，一个过期日期和哈希值 H。可信测量模块用 TPM 的私钥对 C 进行签名存储。当且仅当以下条件成立时，TPM 才返回一个验证通过的状态标志：①证书 C 没有过期；②C 的签名是有效的；③存储在证书中哈希值 H 和引导组件的计算值相匹配。

当可信测量模块在报告验证码和度量值时，是用代表它身份的私钥对数据签名，引导组件获取数据后根据签名可以判断数据是否在传送过程中被篡改，同时还可以通过该签名来证明可信测量模块的身份。

5.2 性能分析

相对于普通的引导过程，可信引导过程需要启动通道将组件传递给 TPM，TPM 需要执行可信测量程序来验证该组件。由此带来了额外的时间开销。时间开销用 T 来表示，包括两个部分，一部分是通信开销，另一部分是计算(验证)开销。

首先来计算 CPU 与 TPM 之间的通信开销。通信开销用 T_C 表示，包括两个部分，一部分是通道传送时间，用 T_T 来表示；另一部分是数据传送时间，用 T_Z 表示。对于通道时间 T_T：

$$T_T = T_S + n \times T_D \tag{3}$$

其中 T_S 为设备选择时间。从通道响应设备发出数据传送请求开始，到通道实际为这台设备传送数据所需要的时间。T_D 为传送一个字节所需要的时间，实际上就是通道执行一条通道指令，即数据传送指令所用的时间。n 是传送的字节数。对于数据传送时间 T_Z：

$$T_Z = L/v \tag{4}$$

其中 v 是主机 CPU 与通道、通道与 TPM 之间总线的数据传输速率，L 表示主机 CPU 与 TPM 之间的距离。所以，通信开销：

$$T_c = T_T + T_Z = T_S + n \times T_D + L/v \tag{5}$$

其次来计算验证时间。验证时间用 T_v 来表示，验证时间：

$$T_v = \sum_{i=0}^{n} t_i \tag{6}$$

其中 t_i 表示验证第 i 个组件所化的时间。实际上，验证过程中需要运行散列函数 SHA. 1。根据文[2]的计算，在 1GHz Pentium 机器上，用 SHA-1 计算一个 1MB 文件的摘要值大约需要 13ms。

所以总的时间开销为：

$$T = T_c + T_v = T_S + n \times T_D + L/v + \sum_{i=0}^{n} t_i \tag{7}$$

对于高速总线，$T_Z \to 0$，所以(7)式可以变为：

$$T = {}_v = T_S + n \times T_D + \sum_{i=0}^{n} t_i \tag{8}$$

实验条件：一台可信 PC 机(P4，1. 8GHz)，平台为 Linux(内核版本为 2. 6. 9)，CPU 与 TPM 用速率为 1MB/s 的通道连接。用(8)式估算该 PC 机启动过程总的时间开销。Linux 启动过程中需要验证的代码包括 BIOS、19 中断程序、Boot、Loader、操作系统内核映像以及初始化进程，这些代码的总量不会超过 10M，按照 1MB 文件的摘要值大约需要 13ms 来计算，验证这些代码的时间大概为 0. 13s；用通道来传送这些数据需要 10s，所以增加的额外时间开销约为 10. 13s。Linux 的启动总时间平均约为 56s 左右，因此 PRTSP 给系统带来了约 16%额外开销。当然该开销可以通过提高通道速率和 SHA. 1 运算速率来降低该额外开销。

总结 可信计算是近年来信息安全界和系统结构领域关注的重点。本文提出的并行可复原可信引导过程具有重要的意义，主要表现在：

(1) 解决了“可信引导”的问题。为可信计算机环境提供了基础；

(2) 并行可复原可信引导过程可以确保操作系统启动过程的安全性和可信性；

(3) 与相关的文献比较起来，并行可复原可信引导过程在系统结构层次进行了阐述，并进行了安全性分析和性能分析。

进一步的工作包括：(1)将 PRTSP 过程在 Linux 平台上的实施。(2)需要进一步研究操作系统动态执行环境的可信性。

参 考 文 献

1 程耕国，刘先勇，鲍考明. Linux 内核启动过程分析[J]. 计算机工程与设计，2006，27(9)：1528～1529
2 方艳湘，黄涛. Linux 可信启动的设计与实现[J]. 计算机工程，2006，32(9)：51～53
3 Trusted Computing Group(EB/OL). http//www. trustedcomputinggroup. org，2001
4 Arbaugh W A，Farbert D J，Smith J M. A Secure and Reliable Bootstrap Architectu2[C]. In：Proceedings of the IEEE Symposium on Security and Privacy，1997
5 Cai Yi. Research on Secure Operating System for Supporting Trusted Operating Platform[D]. Naval University of Engineering，2005
6 沈昌祥. 构造积极御综合防护体系[J]. 信息安全与保密，2004(5)：17～18
7 周明辉，梅宏. 可信计算初探[J]. 计算机科学，2004，31(7)：5～8
8 林闯，彭雪梅. 可信网络研究[J]. 计算机学报，2005，28(25)：751～758
9 侯方勇，王志英. 可信计算研究[J]. 计算机应用研究，2004(12)：1～4
10 刘鹏，刘欣. 可信计算概论[J]. 信息安全与通信保密，2003(7)：17～19
11 谭良，周明天. CRL 分段-过量发布新模型[J]. 电子学报，2005，33(2)：227～230
12 谭良，周明天. CRL 增量-过量发布新模型[J]. 计算机科学，2005，32(4)：133～136
13 郑宇，何大可，何明星. 基于可信计算的移动终端用户认证方案. 计算机学报，29(8)：1255～1264
14 Oppliger R，Rytz R. Does trusted computing remedy computer security problems[J]. Security & Privacy Magazine(IEEE)，2005，3(2)：16～19
15 Reid J，Nieto J M G，Dawson E，et al. Privacy and trusted computing[J]. Database and Expert Systems Applications. In：14th International Workshop，Digital Object Identifier 10. 1109/DEXA. 2003. 1232052，2003. 383～388
16 Felten E W. Understanding trusted computing：will its benefits outweigh its drawbacks [J]. Security & Privacy Magazine (IEEE)，2003，1(3)：60～62
17 Iliev A，Smith S W. Protecting client privacy with trusted computing at the server[J]. Security & Privacy Magazine(IEEE)，2005，3(2)：20～28
18 Arbaugh B. Improving the TCPA specification[J]. Computer，2002，35(8)：77～79
19 Quisquater J J，Samyde D. Electro Magnetic Analysis(EMA)：Measures and Countermeasures for SmartCards[M]. LNCS2140，Springer-Ver2lag，2001
20 Boneh D，DeMillo R，Lipton R. On the Importance of Checking Crypgraphic Protocols for Faults[M]. LNCS1233，Springer-Verlag，1997

(上接第 283 页)

试成员发现一个隐错 (Bug)，然后把这个隐错报告给测试组其他成员，组织成员迅速地确认了此隐错并确定其所报告软件问题的严重性。在此过程中，最为核心的知识就是这个隐错，它是测试成员经验和创造性思想的结晶。事实上，这一过程也是测试知识在个人和组织间传递和转化的过程，本文提出的知识管理模型也从个人和组织两个方面提供支持。其中，对组织知识管理活动的支持体现在 K-Engineer 上。

· K-User：

知识的社会化(socialization)：用户通过与组织其他成员的交流、讨论获取新的测试知识，并根据获取的新知识产生创造性的新思想；

知识的外在化(Externalization)：用户的个人知识在图 4 的处理过程中变成本体索引的组织知识，以达到个人知识的外在化；

· K-Engineer：此模型对 K-Engineer 的支持除了包括 K-User 所有之外还有

知识的授权：对组织知识的使用进行授权；

知识的提炼：对进入系统的的个人知识进行提炼，确认成为组织知识的文档都具有一定的价值。

结束语 知识管理不仅仅是一个技术问题，而是一个复杂的"社会-技术"系统，IT 技术对其支持作用有限，却又是必不可少的因素。本文主要讨论了如何将一个管理理论模型"SECI 知识螺旋"与本体技术结合起来，以支持软件知识在软件测试组织和个人之间的转化活动，提出一个以软件测试领域本体为核心的知识管理模型。该模型具有较好的实用性，我们下一步的主要计划就是根据本文提出的基于本体的软件测试知识管理模型，开发出支持软件测试知识管理活动的软件平台。

致谢 在此，向对本文的工作给予支持和建议的北航软件所的所有同学和老师表示感谢！

参 考 文 献

1 Brent Gallupe R. Knowledge Management Systems：Surveying the Landscape. Queen's University at Kingston，Queen's Management Research Centre for Knowledge-Based Enterprises，October 2000
2 Gruber TR. A translation approach to portable ontology specification. Knowledge Acquisition，1993，5：199～220
3 Abran A，Moore J，Bourque P，Dupuis R L，Tripp L. Guide to the Software Engineering Body of Knowledge - SWEBOK，Trial Version 1. 0，IEEE-Computer Society Press，May 2003. URL：http：//www. swebok. org
4 Nonaka，Takeuchi. The knowledge creating company. Oxford University Press，Oxford，1995
5 Guarino N. Semantic Matching：Formal Ontological Distinctions for Information Organization，Extraction，and integration. In：Pazienza M T，ed. Information Extraction：A Multidisciplinary Approach to an Emerging Infromation Technology，Springer Verlag，1997. 139～170
6 Davenport T H，Prusak L. Working Knowledge：How organizations manage what they know，Harvard Business School Press，1997
7 Mizoguchi R，Ikeda M. Towards Ontology Engineering The Institute of Scientific and Industrial Research，Osaka University，1998
8 夏敬华，金昕. 知识管理. 北京：机械工业出版社，2003