容忍非信任组件的可信终端模型研究

秦 晰[1],常朝稳[1],沈昌祥[2], 高 丽[1]
(1.信息工程大学电子技术学院,河南郑州 450004;2.北京工业大学,北京 100022)

摘 要: 可信计算规范要求可信计算平台上运行的所有组件均要保证可信,这一机制严重制约了可信计算平台的应用.本文提出一种容忍非信任组件的可信终端模型,与现有可信计算平台相比,该模型允许非信任组件的存在,但同时能保证安全结果可预测和可控性.模型分为可信域和容忍非信任组件的不可信域.基于信息流无干扰理论和域间无干扰思想,给出非信任组件容忍机制并推导出可信终端应满足的充分条件.在此基础上给出具体的物理模型设计,并证明该模型为可信终端模型.

关键词: 可信终端模型; 非信任组件; 无干扰; 嵌入式可信系统
中图分类号: TP309.2 **文献标识码:** A **文章编号:** 0372-2112 (2011) 04-0934-06

Research on Trusted Terminal Computer Model Tolerating Untrusted Components

QIN Xi[1], CHANG Chao-wen[1], SHEN Chang-xiang[2], GAO Li[1]
(1. *Institute of Electronic and Technology, Information Engineering University, Zhengzhou, Henan* 450004, *China*;
2. *Beijing University of Technology, Beijing* 100022, *China*)

Abstract: The Trust Computing Group Specifications specify all components running on the trusted computing platform should be trusted, which seriously restrained the applications of trusted computing platform. This paper proposes a trusted terminal computer model tolerating untrusted components. Comparing with the existing trusted platforms, the model allows untrusted components loading and assures the security results be expected and controlled. The model includes trusted domains and untrusted domains tolerating untrusted components. With the non-interference theory, an untrusted component-tolerating mechanism is designed and reasons out the sufficient conditions about the trusted domain can run trustfully. Based on the theory model, provides a detailed physical model and proves it be a trusted terminal computer model.

Key words: trusted terminal model; untrusted components; non-interference; embedded trusted system

1 引言

可信网络连接 TNC(Trusted Network Connect)[1]要求接入网络的终端提供其计算平台可信性证明[2],并根据安全策略对计算平台可信状态进行评估,确保网络连接的可信性.

在 TNC 架构中,要求访问请求者的计算平台是可信的.从现有可信计算平台解决方案来看,Intel 的 LaGrande[3]技术、微软的 NGSCB[4]等技术均基于硬件安全芯片,配合安全操作系统,为用户提供高安全强度的运行环境.国内的一些可信计算平台解决方案[5]也多采用可信安全芯片结合专用主板、并对 BIOS 和操作系统进行安全增强的方式来构造可信计算终端.因此,构造可信计算平台不仅需要可信的硬件安全芯片的支持,还需要 I/O 设备、操作系统和相关软件的共同配合.鉴于此,在传统计算平台上构造可信计算环境是困难的.

本文提出一种容忍非信任组件(tolerating untrusted components,TUC)的可信终端模型.TUC 模型在不改变现有终端硬件结构和上层操作系统前提下,保证非信任组件在终端上的存在不会造成严重的信息安全威胁,实现域间隔离和无干扰,保证安全结果的可预测和可控性.

2 可信计算与信息流无干扰理论

2.1 TCG 可信平台规范应用局限性

按照 TCG 可信计算的基本概念[6,7],一个组件是可信的当且仅当其通过了完整性评估.基于这一观点,如果某一组件的度量结果与参考值不一致,则组件被判为"不可信",该组件不能被启动和加载.但实际上,不可信的组件并不代表该组件就一定恶意破坏过,如版本升级、新的组件等,在这种情况下,被判定不可信的组件也许是可以信任的.如果平台中增加新硬件或升级重要系统部件,就必须对可信平台上的 TPM 模块中相应完整

收稿日期:2010-03-16;修回日期:2010-11-18
基金项目:国家 863 高技术研究发展计划(No.2007AA01Z479)

性度量参照值进行重写,这一过程复杂而漫长.另外,现有计算终端往往缺少 TPM 模块支持,无法直接基于 TPM 构建可信应用环境.正是 TCG 的这一处理机制严重制约了可信计算平台的应用并受到多方质疑[8].

2.2 信息流无干扰理论与可信模型

Goguen 和 Meseguer 等[9]提出的信息流无干扰思想认为:如果某一操作在安全域 u_1 中且该操作会改变其所在域的安全状态,但该操作不会对另一安全域 u_2 所观察到的系统状态产生影响,那么就可认为域 u_1 对于域 u_2 是无干扰的;1992 年,RUSHBY[10]又在此基础上,采用状态机的无干扰模型,给出系统关于传递和非传递无干扰策略是安全的定义,并给出了无干扰模型的三个性质:单步一致性、结果一致性和局部一致性.

在可信计算领域,信息流无干扰理论被许多研究者用来建立安全终端模型.赵佳等[11]针对可信链传递缺乏理论模型的问题,从动态的角度建立了基于无干扰理论的可信链模型,并进行了形式化描述和验证;张兴等[12]借鉴信息流的无干扰理论,利用进程代数和逻辑推理方法,给出了进程运行可信的条件和性质;赵勇[13]针对无干扰理论高度抽象且条件过于严格、难以实际应用的特点,提出了一个面向可信应用环境的隔离模型,模型将来自环境的有害干扰判断转化为度量干扰源实体任务的可信性,具有更好的实用性.但上述研究都要求组件可信或只和可信组件进行交互来保证系统的可信性,难以有效克服 TCG 可信平台的应用局限性.

3 容忍非信任组件的可信终端模型

容忍非信任组件的可信终端模型的核心设计思想是,终端分为两个域,即可信域和非信任域.可信域被要求是运行可信的,包含终端的信任根,负责终端和外部实体的信息封装和交换;非信任域允许非信任组件的加载,非信任域的行为对终端是有害的,不管非信任组件出于何种动机需要和外部实体交互信息,终端输入/输出信息一定是被可信运行的组件封装的;通过可信封装,可以隔离终端和其它非信任实体之间的信息有效交换;通过隔离交换,防止终端对接入的受保护网络进行攻击.含有非信任组件的终端在接入受保护网络时,既不破坏网络平台的安全性,也不损害网络的信息私密性和完整性,即非信任组件的存在对网络安全和系统安全没有影响,基于信息无干扰理论,就可以认为是一种可信的接入.

3.1 模型元素

本模型是在 RUSHBY[10]无干扰模型基础上建立的.

定义 1 设终端 $M=\{S,D,A,C,O,F,R\}$,且 $M\in TNC$.其中:

S 表示终端状态的集合,初始状态 $s_0\in S$.终端的状态可以用数量有限的客体对象 N(客体名称 n 的集合)及其取值 V(客体取值 v 的集合)表示,即 $S\times N\rightarrow V$.

D 表示终端包含域的集合,包括可信域 D_T 和非信任域 D_N 两个子集.D_T 所有组件都是静态可信的、完整的;D_N 域允许有非信任组件存在并加载.

A 表示终端的操作集合.$A=\{r,w\}$ 表示组件访问资源的模式.r 表示读操作,w 表示写操作.

令 Λ 表示空操作,$\circ$ 表示动作级联关系,s 表示状态,a 表示单个操作,α 表示操作序列,A^* 表示操作集合的闭包.

C 表示终端组件的集合,元素用 c 表示.组件 C 包括 C_T 和 C_N 两个子集,C_N 表示终端不可信域的组件集合;C_T 表示终端可信域包括的组件集合.

O 表示终端的客体资源.包括网络、数据、内存和物理设备等.

F 表示终端系统函数.$F=\{step,output,run,domain,component,seal,unseal\}$,$F$ 有 7 个函数组成,即:

(1)转移函数 $step:S\times A\rightarrow S$,$step(s,a)$表示操作 a 使终端从状态 s 转换成另一种状态.

(2)系统输出函数 $output:S\times A\rightarrow O$ 表示终端的输出只与状态和操作有关.

(3)状态多步转移函数 $run:S\times A^*\rightarrow S$ 表示操作序列 A^* 使终端状态发生的转换,满足 $run(s,\Lambda)=s$ 和 $run(s,a\circ\alpha)=run(step(s,a),\alpha)$.

(4)域隶属函数 $domain:C\rightarrow D$ 表示组件所在的域.

(5)操作与组件影射函数 $component:A\rightarrow C$,$component(a)=c$ 表示操作 a 由组件 c 实施的.

(6)封装函数 $seal:C\times V\rightarrow V$,$seal(c,v)$表示组件 c 对客体对象的取值 v 进行一次封装操作.

(7)解封函数 $unseal:C\times V\rightarrow V$,$unseal(c,v)$表示组件 c 对客体对象的取值 v 进行一次解封操作.

R 表示终端组件集合上的关系,包括等价关系和干扰关系两种.

(1)等价关系 $\overset{c}{=}$,表示状态集合上关于组件 c 的等价关系.

(2)干扰关系 ~>,组件集合 C 上的干扰关系,具有自反性.当组件 c 在执行操作时,从组件 d 的角度能够观察到的系统发生了变化,则称组件 c 对组件 d 有干扰,用 $c\rightsquigarrow d$ 表示;否则,则称组件 c 对组件 d 无干扰,用 $c\not\rightsquigarrow d$ 表示.

定义 2 过滤函数 $filtrate:A^*\times D\rightarrow V$,$filtrate(\alpha,c)$表示从动作序列(中过滤掉所有不干扰组件 c 的动作序列,其中:$\alpha\in A^*$,$c\in C$.

$$filtrate(\Lambda, c) = \Lambda$$

$$filtrate(a \circ \alpha, c) = \begin{cases} a \circ filtrate(\alpha, c), \text{if } component(a) :> c \\ filtrate(\alpha, c), \text{otherwise} \end{cases}$$

定义 3　可信组件(TC)是完整性没有得到破坏的组件. $TC \in C$,一个组件是可信的条件是:

iff Hash(c) = $c_0 | c \in C$(c_0 是组件 c 的完整性期望值)

定义 4　不可信组件(MC)是完整性被破坏的组件. $MC \in C$,一个组件是不可信的条件是:

iff Hash(c) $\neq c_0 | c \in C$(c_0 是组件 c 的完整性期望值)

定义 5　如果 $\exists\, \alpha$ 使得 $run(s_0, \alpha) = s, s \in S$,则称 s 为可达状态,可用二元组(c, α)表示(c 为终端处于状态 s 时执行的组件).

3.2　基于可信组件的无干扰可信域

根据前一部分的定义,在终端的可信域 D_T,所有的组件(用 tc 表示)都是静态可信的,即:$tc \in TC$, $domain(tc) = D_T, TC \in C$.

文献[6,7]研究认为,组件 $c = component(a)$运行可信的条件可表示为:

$$output(run(s_0, \alpha), a) = output(run(s_0, filtrate(\alpha, component(a))), a) \quad (1)$$

该条件可解释为:如果没有潜在的组件干扰该组件的运行,或者该组件只受那些可预知的组件影响,就说明该组件是运行可信的.本文认为,式(1)的表达是不充分的,如果可以预知该组件会受哪些组件影响,那该组件一定是确定的、没有被改变的,即该组件一定是静态可信的、完整的,即 $c \in TC$.基于上述考虑,组件运行可信的定义为:

定义 6　组件 $c = component(a)$运行可信的条件可表示为:

$$output(run(s_0, \alpha), a) = output(run(s_0, filtrate(\alpha, component(a))), a) \wedge component(a) \in TC$$

根据 3.1 模型元素的定义,因可信域组件是静态可信的,故在终端可信域,定义 6 和式(1)等价.

定义 7　观察等价性质,对于 $\forall\, tc \in C$,都有一个关于终端状态的观察等价关系,记为$\overset{tc}{=}$,即如果 $s \overset{tc}{=} t$,则从组件 tc 的角度观察,终端状态 s 和 t 相同.

可以证明,在具有观察等价性质的可信域中,如果一个具有输出隔离性质的组件 tc 满足:

$$\text{run}(s_0, \alpha) \overset{tc}{=} \text{run}(s_0, filtrate(\alpha, component(a)))$$

则该组件运行可信.

定义 8　可执行组件 tc 满足输出隔离性质的条件:

$$s \overset{tc}{=} t \Rightarrow output(s, a) = output(t, a), tc = component(a)$$

该性质说明,如果 $s \overset{tc}{=} t$,那么,执行操作 a 时无论终端处于状态 s 还是 t,执行同一操作 a 后系统输出相同.

定义 9　可执行组件 tc 满足无干扰隔离性质条件:

$$domain(a) \not\rightsquigarrow tc \Rightarrow s \overset{tc}{=} step(s, a), \quad tc = component(a)$$

该性质说明,如果一个操作 a 不对组件 tc 的运行产生干扰影响,那么对于组件 tc 而言,执行操作 a 前后组件 tc 观察到的系统状态是相同的.

定义 10　可执行组件 tc 满足单步隔离性质条件:

$$s \overset{tc}{=} t \Rightarrow step(s, a) \overset{tc}{=} step(t, a), tc = component(a)$$

该性质说明,如果 $s \overset{tc}{=} t$,那么,执行操作 a 时无论终端处于状态 s 还是 t,执行同一操作 a 后系统状态相同.

文献[6,7]研究表明,有如下系统运行可信判定定理:

定理 1　当满足如下 3 个条件时,系统 M 可信:

(1)M 从可信根开始运行;

(2)M 所有组件($\forall\, tc \in C$)均满足单步隔离性和输出隔离性;

(3)组件满足可信验证,即:if run($s0, \alpha$) $\overset{component(a)}{\Longrightarrow}$ run($s0$, $filtrate(\alpha, component(a))$) $\Rightarrow$ run($s0, \alpha$) $\overset{component(b)}{\Longrightarrow}$ run($s0$, $filtrate(\alpha, component(b))$).

基于定理 1,结合模型元素及其定义,可得:

定理 2　终端 M 可信域运行可信的充分条件是:

(1)该可信域从可信根开始运行;

(2)该域所有组件是是静态完整的,且该域所有组件是顺序加载的;

(3)该域所有组件加载时进行完整性度量.

证明:设可信域满足上述条件.

根据本定理的条件(2),该域所有组件是完整的、顺序加载的.也就是说,组件 tc 加载执行时系统状态是唯一的、确定的和可预知的;tc 加载执行时系统不存在第二种状态的可能,在这种条件下,如果有 $\forall\, tc(s \overset{tc}{=} t)$, $tc \in TC$,则一定有 $s = t$,进而有:

$$s \overset{tc}{=} t \Rightarrow output(s, a) = output(t, a), s \overset{tc}{=} t \Rightarrow step(s, a) \overset{tc}{=} step(t, a)$$

由定义 8 和 10, D_T 所有组件($\forall\, tc \in C$)均满足输出隔离性和单步隔离性.即满足定理 1 的条件(2).

根据本定理条件(3),该域所有组件基于完整性度量加载.假设 tc_i 是可信运行的组件,根据定义 6,有:

$$run(s_0, \alpha) \overset{component(a)}{\Longrightarrow} run(s_0, filtrate(\alpha, component(a)), \quad tc_i = componet(a) \quad (2)$$

假设 tc_j 是 tc_i 之后加载的组件($j = i + 1$), tc_i 度量

tc_j 的完整性,即判断如下条件是否成立:

$\mathrm{Hash}(tc_j) = tc_{j0}$($tc_{j0}$是组件 tc_j 的完整性期望值) (3)

如果式(3)成立,即 tc_j 通过完整性度量,则:站在 tc_i 的角度 tc_j 是确知的,进一步说,tc_j 加载时会受哪些组件影响是确知的.

又:tc_i 是可信运行的,没有潜在的组件干扰该组件 tc_i 的运行,而 tc_i 又确知 tc_j 是确定的并按确定的方式加载,没有潜在的组件干扰该组件 tc_j 的运行,tc_i 将可信性质传递给了 tc_j,所以,tc_j 也是可信运行的,即:

$$run(s_0,\alpha)\xrightarrow{component(b)} run(s_0,filtrate(\alpha,component(b))),\quad tc_j = componet(b) \tag{4}$$

由此可知,式(2)成立必然可推式(4)成立,即:

$$\text{if } run(s_0,\alpha)\xrightarrow{component(a)} run(s_0,filtrate(\alpha,component(a))) \Rightarrow run(s_0,\alpha)\xrightarrow{component(b)} run(s_0,filtrate(\alpha,component(b))) \tag{5}$$

式(5)满足定理 1 的条件(3),即组件满足可信验证.本定理条件 1)和定理 1 条件 1)相同.故:可信域运行可信,本定理得证.

3.3 基于域间无干扰的非信任组件容忍机制

本小节从域视图的角度,通过域间信息流无干扰理论来描述对非信任组件的容忍机制.

定义 11 不属于终端可信域 D_T 的所有组件组成终端的非信任域 D_N.即:

if $(domain(c)\in D)\wedge(domain(c)\notin D_T)$ then $domain(c)\in D_N$

定义 12 域 $u\in D$ 所能访问的客体对象集合记为 $N_u=\{n_1,n_2,\cdots,n_k\}$($k$ 为能访问的客体对象个数),N_u 称为域 u 可见的客体对象集合,N_u 的取值 $V_{Nu}=\{v_{n1},v_{n2},\cdots,v_{nk}\}$称为域 u 的视图 view.

定义 13 终端的视图可用 $S\times(N_T\cup N_N\cup N_{IO})\rightarrow(V_{NT}\cup V_{NN}\cup V_{NIO})$表示.$N_T$ 表示可信域可见的客体对象集合,N_N 表示非信任域可见的客体对象集合,N_{IO}表示终端和其他外部实体交换时使用的客体对象集合,V_{NT}、V_{NN}、V_{NIO}分别表示 N_T、N_N、N_{IO}客体对象的取值,分别称为终端的可信域视图、非信任域视图、输入输出视图.

定义 14 函数 $write: D\rightarrow P(N_u)$表示域 u 可以写的客体对象集合,$P(N)$表示 N 的子集.函数 $read: D\rightarrow P(N_u)$表示域 u 可以读的客体对象集合.

定义 15 w 表示非信任域组件 c 参与完成的一次向外部实体输出信息的任务,完成任务 w 终端执行了一个动作序列 α.设 w 任务执行前系统视图 $view1=\{V_{NT1}、V_{NN1}、V_{NIO1}\}$,而 w 任务完成后系统视图 $view2=\{V_{NT2}、V_{NN2}、V_{NIO2}\}$.如果满足以下条件,称组件 c 的输出信息流是安全可控的.

$$\text{if } V_{NIO1}\neq V_{NIO2}\Rightarrow(a_1\circ a_2\in\alpha)\wedge V_{NIO2}=seal(c_T,f(V_{NN2})),\quad component(a_1)=c_T\in D_T, component(a_2)=c\in D_N$$

(f 为业务规则函数,随 w 不同而不同) (6)

式(6)说明,如果任务 w 向终端外部实体输出信息时造成了输入输出视图 V_{NIO2}的改变,必然在本次任务动作序列 α 中包含一个属于可信域组件发出的动作 a_1,a_1 对任务的输出进行了封装.即:当组件 c 向外部实体发送信息时,信息必须经可信域组件 c_T 安全封装.

定义 16 r 表示非信任域组件 c 完成一次从外部实体读信息的任务,a_2 是组件 c 发出的一个读取实体对象集 V_{NN}的一个动作.设 r 任务执行前系统视图 $view1=\{V_{NT1}、V_{NN1}、V_{NIO1}\}$,而 r 任务完成后系统视图 $view2=\{V_{NT2}、V_{NN2}、V_{NIO2}\}$,完成任务 r 终端执行了一个动作序列 α.如果满足以下条件,组件 c 的输入信息流是安全可控的.

$$\exists a_1((a_2\circ a_1\in\alpha)\wedge(V_{NN2}=unseal(c_T,g(V_{NIO2})))),\quad component(a_1)=c_T\in D_T \tag{7}$$

式(7)中,g 为业务规则函数,随 r 不同而不同.式(7)说明,如果组件 c 输入信息流是安全可控的,任务 r 从外部实体读取信息时,必然在本次任务动作序列 α 中包含一 D_T 组件发出的动作 a_1,该操作对外部实体的输入进行了解封和成功验证.也就是说,当组件从外部实体接收信息时,信息必须经 D_T 组件解封并验证.

定理 3 一个终端 $M\in$ TNC 的组件可分为非空的可信域 D_T 和非信任域 D_T,当 M 满足以下条件时,称 M 是在 TNC 架构下容忍非信任组件的可信终端.

(1)D_T 是运行可信的,即:满足定理 2 要求的充分条件;

(2)$(N_{IO}\subset N_T)\wedge(N_{IO}\not\subset N_N)$;

(3)$N_T\cap write(D_N)=\varnothing$;

(4)终端对 N_{IO}的读写均是基于 TNC 协议规范安全封装的.

证明:由条件(3):

$N_T\cap write(D_N)=\varnothing$

由定义 13 和 14,$write(D_N)$表示非信任域 D_N 能写(修改)的客体对象集,N_T 表示可信域 D_T 能访问的客体对象集,二者交集为空,说明非信任域 D_N 的操作对可信域 D_T 的能访问的客体对象集合的取值不产生影响,根据 Goguen 等关于信息流无干扰的定义,显见:

$$D_N\nrightarrow D_T(D_N\text{ 对 }D_T\text{ 无干扰}) \tag{8}$$

由条件(2)的右半部分,$N_{IO}\not\subset N_N$,则:$N_N\cap N_{IO}=\varnothing$

由定义 13,N_N 是非信任信域可见的客体对象集合,N_{IO}是终端和其他外部实体用于交换的客体对象集

合,二者交集为空,说明非信任域 D_N 内的操作对该终端外的其它外部实体的状态无影响,基于 Goguen 等关于信息流无干扰的定义,显见:

$$D_N \not\rightsquigarrow D_{IO}(D_{IO}\text{表示终端外部输入输出域}) \quad (9)$$

由条件(2)的左半部分,$N_{IO} \subset N_T$,则:$N_T \cap N_{IO} \neq \varnothing$

即:可信域 D_T 和外部输入输出域能共同访问的客体对象集不为空,这个条件允许终端的可信域 D_T 和外部实体之间是交换信息.

结合式(8)和式(9),可知:非信任域组件只能通过 D_T 这个中间层和终端外部实体交换信息,且非信任域组件只能通过读方式和 D_T 域组件交换信息.

联合条件(2)和条件(3),有:

$$(N_T \cap write(D_N) = \varnothing) \wedge (N_{IO} \not\subset N_N)$$

说明 D_N 对 N_{IO} 的访问只能通过 D_T 进行,而通过条件(1)可知,D_T 是运行可信的;结合条件(4),终端对 N_{IO} 的读写均是基于 TNC 协议规范安全封装的,根据定义 15 和 16,可以认为非信任组件 c 的输入/输出信息流是安全可控的.

综上所述,不管非信任组件出于什么动机和终端外部实体交互信息,终端输入/输出信息一定是被可信运行的组件封装的,这种封装/解封机制可确保终端 M 上的非信任组件不能和 TNC 架构外的其它实体实现有效信息交互,进而保证受保护网络的信息安全.本定理得证.

4　容忍非信任组件的可信终端模型设计

4.1　模型组成

基于第 3 节提出的 TUC 形式化理论模型,给出该模型具体的物理设计与实现,如图 1.

(1)信任域　该域由硬件层嵌入式可信系统和可信虚拟机组成.

嵌入式可信系统作为终端的可信根.嵌入式系统中的可信密码服务模块为终端提供可信支撑,包括可信度量、可信存储和可信报告,在这一方面,它对应 TCG 可信平台的可信平台控制模块 TPM.嵌入式可信系统还具有可信引导加载、可信平台控制和安全策略管理的作用.嵌入式可信系统位于系统的硬件层.

虚拟机 VM(virtual machine)是支持多操作系统并行运行在单个物理服务器上的一种系统,能够提供更加有效的底层硬件使用[14,15].可信虚拟机通过虚拟化技术实现对硬件资源的控制.通过存储设备虚拟化,实现透明加密存储保护;通过 I/O 端口资源(USB、串口、并口)虚拟化,实现对 I/O 端口资源的安全封装;基于 VMM 监视控制器,实现安全机制和应用操作系统的隔离.

嵌入式可信系统作为可信度量根对虚拟机的完整性进行度量,保证虚拟机的可信性.

(2)非信任域　非信任域包括应用层操作系统和应用层组件两部分组成,应用层组件允许非信任组件存在.

应用操作系统层包括支持白名单机制的可信计算软件基(TCSB)、可信基础支撑软件系统服务(TSS)和可信基础支撑软件应用服务(TAS).TCSB 的安全机制包括应用层的进程控制、访问控制及其访问决策、VPN 连接、网络包过滤、安全代理和审计.TSS 运行在 TCSB 之上,基于嵌入式可信系统,向应用程序提供可信平台控制模块的证书,密钥,密码功能和完整性数据管理等接口.TAS 运行在 TSS 之上,向用户提供可信计算支撑软件的高层应用接口,包括完整性保护、可信认证和数据保护等三个部分.

应用层组件为可信组件或非信任组件.应用层组件在 TCSB 白名单机制的控制下加载运行.

4.2　模型分析与证明

该物理模型同样包括可信域 D_T 和非信任域 D_N 两部分.根据定理 3,有:

(1)对于可信域 D_T,嵌入式可信系统作为可信根是无条件可信的并被首先加载,虚拟机是在通过完整性度量后顺序加载的,根据定理 2,D_T 是可信的,满足定理 3 的条件(1).

(2)非信任域运行在虚拟机之上,非信任域只能访问虚拟化后的资源和设备,也就是说终端和外部实体交换时使用的资源和设备(客体对象集合)对非信任域来说是不可见的.根据定义 14,有:$(N_{IO} \subset N_T) \wedge (N_{IO} \not\subset N_N)$,即定理 3 的条件(2)成立.

(3)VMM(Virtual Machine Monitor),直接运行于硬件裸机上,处于特权级 ring 0,负责虚拟机调度以及共享资源控制访问等.特权域 0 的内核运行在 ring 1,是具有特殊地位的虚拟机,实现安全封装机制.基于 VMM 监视

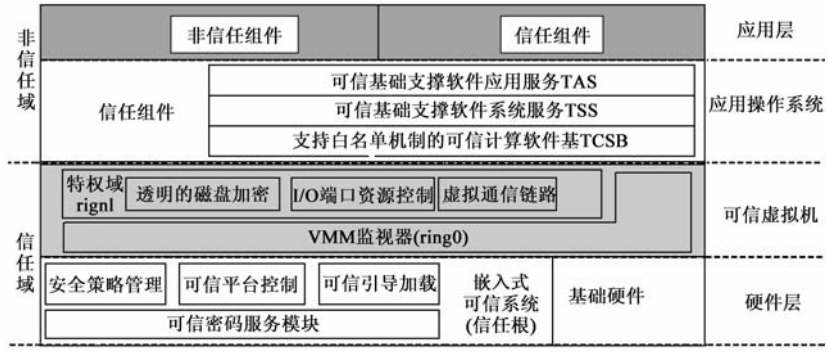

图1　可容忍非信任组件的可信终端

控制器,可实现特权域和应用操作系统的隔离,因此 D_N 对 D_T 无干扰,即定理 3 的条件(3)成立.

(4)在物理模型中,终端的输入/输出信息都是经过安全封装的,当然非信任域组件的输入/输出信息流是安全可控的,故定理 3 的条件(4)成立.

5 结论及下一步的工作

本文提出一种容忍非信任组件的可信终端模型 TUC.与现有可信计算平台不同,TUC 模型允许非信任组件加载,但能保证安全结果的可预测和可控性.基于信息流无干扰理论,对模型的可信域进行形式化分析;基于域间无干扰思想,给出了非信任组件容忍机制并推导出可信终端应满足的充分条件,进而得到了形式化的理论模型.在此基础上,设计了一个可实现的物理模型并证明其为可信终端.

下一步需要做的工作是,按照可信终端物理模型的层次结构,分别针对嵌入式可信系统、可信虚拟机、操作系统进程和应用组件给出不同的完整性度量方法,形成可信终端模型完整的综合度量体系.

参考文献

[1] Trusted Computing Group. TNC Architecture for Interoperability [EB/OL]. http://www.trustedcomputinggroup.org/resources/tnc_architecture_for_interoperability_version_13.

[2] 李晓勇,左晓栋,沈昌祥.基于系统行为的计算平台可信证明[J].电子学报,2007,35(7):1234-1239.
Li Xiaoyong, Zuo Xiaodong, Shen Changxiang. System behavior based trustworthiness attestation for computing platform[J]. Acta Electronica Sinica, 2007, 35(7): 1234-1239. (in Chinese)

[3] Intel. Intel Trusted Execution Technology[EB/OL]. http://www.intel.com/technology/security/downloads/TrustedExec_Overview.pdf, 2007.

[4] Microsoft Coporation.. Next-generation secure computing base [EBOL]. http://www.microsoft.com/res-ources/ngscb/default.mspx 2003.

[5] 刘毅,余发江.瑞达可信计算平台[J].信息网络安全,2006(11):23-25.
Liu Yi, Yu Fajiang. JetWay trusted computing platrom[J]. Netinfo Security, 2006(11): 23-25. (in Chinese)

[6] Trusted Computing Group. TCG specification architecture overview [EB/OL]. http://www.trustedcomputinggroup.org/resources/tcg_architecture_overview_version_14, 2010-03.

[7] Trusted Computing Group. Infrastructure work group integrity report schema specification[EB/OL]. http://www.trustedcomputinggroup.org/resources/infrastructure_work_group integrity_report_schema_specification_version_10, 2010-03.

[8] 周明辉,梅宏,可信计算研究的初步探疑[J].计算机科学,2004,31(7):5-8.

[9] Goguan J A. Meseguern J. Security policies and security model [A]. The 1982 IEEE Symposium on Security and Privacy[C]. Qakland, California, 1982. 11-20.

[10] RUSHBY J. Noninterference, Transitivity, and Channel-Control Security Policies[R]. CSL-92-02, Menlo Park: Stanford Research Institute, 1992.

[11] 赵佳,沈昌祥,刘吉强等.基于无干扰理论的可信链模型[J].计算机研究与发展,2008,45(6):974-980.
Zhao Jia, Shen Changxiang, Liu Jiqiang, et al. A noninterference-based trusted chain model[J]. Journal of Computer Research and Development. 2008, 45(6): 974-980. (in Chinese)

[12] 张兴,陈幼雷,沈昌祥.基于进程的无干扰可信模型[J].通信学报,2009,30(3):6-11.
Zhang Xing, Chen Youlei, Shen Changxiang. Non-interference trusted model based on processes[J]. Journal on Communications, 2009, 30(3): 6-11. (in Chinese)

[13] 赵勇.重要信息系统安全体系结构及实用模型研究[D].北京:北京交通大学,2008.

[14] Dunlap G W, King S T, Cinar S, et al. ReVirt: enabling intrusion analysis through virtual-machine logging and replay[J]. ACM SIGOPS Operating Systems Rev, 2002, 36(SI): 211-224.

[15] Garfinkel T, Pfaff C, Chow J. Terra: A virtual-machine-based platform for trusted computing[A]. Proc of the 19th ACM Symp on Operating Systems Principles[C]. Lake George, NY, United states: Association for Computing Machinery, 2003. 192-206.

作者简介

秦 晰 女,1978 年生于河南焦作,信息工程大学电子技术学院博士研究生.主要研究方向为可信计算和移动网络安全.
E-mail: qxcathy@126.com

常朝稳 男,1966 年生于河南滑县,信息工程大学电子技术学院教授,硕士导师.主要研究方向为信息安全与可信计算.