一种信任链传递模型研究

司丽敏 蔡 勉 陈银镜 郭 颖
(北京工业大学计算机学院 北京 100124)

摘 要 通过度量应用程序及其动态库的完整性来保障应用的静态可信,并且分析有交互的应用之间的关系,建立了一种信任链传递模型来保障应用运行过程中的动态可信,从而构建可信的应用环境。并基于无干扰理论,将系统抽象为应用程序、动作和状态输出,形式化地定义了应用运行可信,给出应用运行可信的条件和性质,推出应用运行可信定理,并证明了基于该模型的应用环境可信判定定理。

关键词 动态库,信任链,无干扰,应用环境
中图法分类号 TP316 **文献标识码** A

Research of a Trust Chain Transfer Model

SI Li-min CAI Mian CHEN Yin-jing GUO Ying
(Dept. of Computer, Beijing University of Technology, Beijing 100124, China)

Abstract Based on measurement applications and its dynamic library to protect the integrity of the application of static credible, and analyzing the relation between interactive applications, this article established chain of trust of the transfer model, to protect applications running in the process of dynamic credible, to build reliable application environment. Based on the intransitive noninterference model, this article abstracted the system as applications, actions, states and outputs, and formally defined to run trusted applications. Application trusted theorem was verified formally. Furthermore, by associating application with system state, the definition and the theorem of application environment trusted were proposed.

Keywords Dynamic library, Chain of trust, Noninterference, Application environment

1 引言

造成计算机安全问题的根本原因在于现有 PC 本身的不安全性。当初设计 PC 时就没有考虑其安全性,缺乏很好的硬件防御措施,使得现有的安全方案很多都是纯软件的,缺乏硬件上的支持,从而十分脆弱。为了解决这个问题,TCG (Trusted Computing Group)提出了用可信计算的思想构建可信平台的方法。

TCG 提出的"可信计算"从行为的角度给出实体可信的定义,强调行为的可预测性和可控性,认为"当一个实体始终沿着预期的方式(操作或行为)达到既定目标,则它就是可信的"[1]。

文献[2]将可信计算的思想总结为:首先构建一个信任根,再建立一条信任链(Chain of Trust),从信任根开始到硬件平台、到操作系统、再到应用,一级认证一级,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。因此,信任链的建立是构建可信计算平台的关键。

TCG 提出的信任链,其构建方案采用装载前度量。文献[3]给出了信任链传递和控制权转移的过程:从可信度量根核心 CRTM(Core of Root Trusted Measurement)开始,依次对各模块进行完整性度量,先对 BIOS(Basic Input/Output System)进行杂凑运算,如果与参考值匹配,则度量通过,将控制权转移给 BIOS,信任链也向前扩展了一步。再度量操作系统加载代码,这样逐步建立信任链。所以信任链就是从底部 CRTM 开始到顶部用户应用程序的链。

在信任链传递过程中,操作系统的启动阶段是一个顺序固定的单一链式过程,通常把 TCG 构建的信任链称为静态的,静态信任链传递已经得到了广泛的研究。由于终端平台上的应用具有多样性和无序性等特点,应用的可信性会发生变化,系统引导的静态信任链构建过程不适用于操作系统到应用之间的可信传递。因此,要研究操作系统到应用、应用到应用间的信任链传递,即如何构建可信的应用环境。

基于信息流的无干扰理论模型从动作和运行结果的角度建立系统安全策略模型。文献[4]研究了基本的无干扰理论模型,给出无干扰模型的 3 个性质,即单步一致性、结果一致性和局部一致性。这一完全模型对可信计算的研究具有借鉴意义。目前基于无干扰理论的可信模型研究还处于尝试阶段。文献[5]初步研究了非传递无干扰理论在可信计算平台中的应用,提出了隔离内核设计思想。

到稿日期:2010-10-18 返修日期:2010-12-29 本文受国家重点基础研究发展计划 973(2007CB311100)资助。
司丽敏(1983—),女,硕士生,主要研究方向为信息安全、可信计算;蔡 勉(1960—),女,博士,教授,主要研究方向为信息安全、可信计算;陈银镜(1985—),女,硕士生,主要研究方向为信息安全、可信计算;郭 颖(1985—),女,硕士生,主要研究方向为信息安全、可信计算。

本文在静态可信概念的基础上，提出了一种动态化的信任链传递模型，分析了交互应用之间相互影响的关系。并基于无干扰理论对信任链在动态传递过程中的性质和行为，进行形式化描述和证明。

2 可信运行环境信任链传递模型构建

2.1 应用环境

假设一个信息系统 $M1$，其中运行着 $p_1, p_2, \cdots, p_n$，n 个应用程序，通过虚拟机把 n 个应用程序放在不同的虚拟机中执行，即隔离在不同的域中，以减少或者屏蔽不同应用之间、没有依赖关系应用之间的非预期干扰，每个域都有不同的可信度。并且应用都具有初始完整性，每个应用可能与其它的应用有交互，而且是运行在单平台上的。

2.2 可信运行环境信任链传递模型建立

TCG 提出的信任链构建方案通常称为静态的。在静态的基础上，本文提出一种动态的构建方案，即构建一种动态的信任链传递模型。

模型包括两个部分：应用的静态可信和应用的动态可信。静态可信可以保障应用的初态可信；动态可信可以保障应用运行过程中的可信。

静态可信是先对所有待运行应用程序的完整性进行检查，只有完整性校验值一致的应用程序才允许运行；其次对应用程序加载的动态库进行检查，判断其是否是系统允许加载的动态库，并对其完整性进行检查。最后符合条件的才允许加载，否则不允许加载。

动态可信是对系统允许运行的应用程序进行控制，对于有交互行为的应用，要对交互的应用进行控制，保障其行为可信。

根据以上的想法，提出一种信任链传递模型，如图 1 所示。图中虚线表示每个应用可能与其它的应用有交互，①对应用程序的完整性进行检查；②对动态库的完整性进行检查；③对应用程序加载的动态库进行检查并对其完整性进行检查。它们属于静态可信，所有待运行的应用程序都包含这 3 步。④⑤对有交互行为的应用进行控制，属于动态可信，只有有交互行为的应用之间才包含这两步。

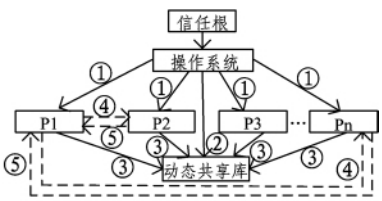

图 1 可信运行环境信任链传递模型

3 形式化描述与证明

3.1 信息流的无干扰理论和模型

信息流的无干扰思想最早在 1982 年由 Goguen 和 Meseguer[6] 提出，此后出现了大量基于无干扰思想的信息流模型[7,8]。1992 年，Rushby[4] 提出了采用状态机的无干扰模型，并给出系统关于传递和非传递无干扰策略是安全的定义，以下沿用文献[4]中的部分术语对该模型进行简要描述。

系统 M 中包括系统状态集 S、动作集 A、输出集 O 和安全域集 D 4 个集合，以及这些集合上定义的 4 个函数：

单步状态转换函数 $\mathrm{step}: S\times A\to S$；

系统运行函数 $\mathrm{run}: S\times A^*\to S$；

输出函数 $\mathrm{output}: S\times A\to O$；

主域函数 $\mathrm{dom}: A\to D$ 表示系统每个动作所属的域。

在传递无干扰模型中，定义～为安全域 D 上的关系，表示安全域间的干扰关系。例如 $u, v\in D$，则 $u\sim v$ 可以简单地理解为信息可以从域 u 流向域 v。为了定义系统 M 在传递的无干扰模型中的安全条件，还要定义一个辅助函数 $\mathrm{purge}: A^*\times D\to A^*$。对于 $\alpha\in A, v\in D$，$\mathrm{purge}(\alpha, v)$ 表示从动作序列 α 中删除所有从不干扰域 v 的域所发出的动作后的动作序列，表示如下：

$$\mathrm{purge}(\Lambda, v)=\Lambda$$

$$\mathrm{purge}(a\circ\alpha, v)=\begin{cases} a\circ\mathrm{purge}(\alpha, v), & \text{if } \mathrm{dom}(a)\sim v \\ \mathrm{purge}(\alpha, v), & \text{otherwise}\end{cases}$$

式中，$\mathrm{dom}(a)=v$，Λ 表示空的动作序列即无动作执行。

系统 M 对策略～的安全条件是：

$$\mathrm{out\ put}(\mathrm{run}(s_0, \alpha), a)=\mathrm{out\ put}(\mathrm{run}(s_0, \mathrm{purge}(\alpha, \mathrm{dom}(a))), a)$$

3.2 基本符号定义

本文的研究着眼于系统的行为、动作、状态和观察结果，在证明过程中借鉴了文献[9,10]无干扰模型及其证明方法和文献[11,12]的证明方法，并借用了其中的符号定义。下面给出基本的符号定义。

定义 1 一个系统 $M1=(S, P, D, A, O, N, Q, F, R)$ 包含如下元素。

S：系统状态集合，包含一个初始状态 $s_0\in S$。S_T 为可信状态集合，且 $s_0\in S_T\circ S_F=S-S_T$ 为不可信状态集合。元素用小写字母 $s_0, s_1, \cdots$ 表示；

P：系统可运行的应用程序集合，用元素 $p_1, q, p_2, \cdots$ 表示；

D：系统可加载的动态库集合，$\{p_1\}\{p_2\}$ 表示可执行代码 p_1, p_2 加载的动态库集合；

A：系统动作集合，用 $a_1, a_2, \cdots$ 表示动作集中的元素；用 $\alpha, \beta, \cdots$ 表示动作序列；

O：应用程序运行输出结果集合；

N：静态可信状态，包括 $\{1, 0\}$；

Q：可信状态，包括 $\{\mathrm{True}, \mathrm{False}\}$；

F：函数集。

①$\mathrm{step}: S\times A(P)\to S$，系统状态经过执行单个动作（或运行某一应用程序）后到达下一个状态。

②$\mathrm{go}: S\times A^*(P^*)\to S$，系统在运行一个动作序列（或多个应用程序）后的状态变迁。状态变迁函数具有下述性质：

$$\mathrm{go}(s, \Lambda)=s,$$

$$\mathrm{go}(s, a\circ\alpha)=\mathrm{go}(\mathrm{step}(s, a), \alpha),$$

$$\mathrm{go}(s, \alpha\circ a)=\mathrm{step}(\mathrm{go}(s, \alpha), a),$$

$$\mathrm{go}(s, (p\circ q\circ\cdots))=\mathrm{go}(\mathrm{step}(s, p), (q\circ\cdots)),$$

$$\mathrm{go}(s, (p\circ q\circ\cdots))=\mathrm{step}(\mathrm{go}(s, (q\circ\cdots)), p)。$$

其中，Λ 表示一个空的动作序列（或没有一个新应用程序运行），$\circ$ 表示动作间（或应用程序运行）的连接操作。

③$\mathrm{output}: S\times A\to O$，表示系统执行动作后产生的结果。

④$\mathrm{proc}: A\to P$，表示一个动作所属的应用程序，即动作的发出者。

⑤pro：$A \to P$，表示一个动作使用的另一个应用，即与 $proc(a)$ 交互的应用。

⑥int：$P \times D^* \to N$，完整性状态判定函数，即判定一个应用程序及其加载的动态库的完整性。

⑦ trust：$P \to Q$，可信判定函数，即判定一个应用程序的可信性。

R：二元关系集合

①状态集合上关于应用程序 q 的等价关系“$\overset{q}{=}$”。

②应用程序集合 P 上的关系“$\sim>$”。当 $p \sim> q$，表示应用程序 p 的执行会对应用程序 q 的执行产生影响。当 $p \not\sim> q$ 时，表示应用程序 p 的执行对应用程序 q 的执行毫不影响。

定义 2 对于 $\forall q \in P$ 和一个动作序列 $\alpha \in A^*$，函数 $delete(\alpha, p)$：$A^* \times P \to A^*$ 归纳定义为

$$\text{delete}(\Lambda, q) = \Lambda,$$

$$\text{delete}(a \circ \alpha, q) = \begin{cases} a \circ \text{delete}(\alpha, q), & \text{if pro}(a) \sim> q \\ \text{delete}(\alpha, q), & \text{otherwise} \end{cases}$$

式中，$q = \text{proc}(a)$。该函数的目的是将没有干扰关系的动作忽略掉，保留那些发生干扰关系的动作，以便简化动作序列。

上述 step 函数、go 函数、delete 函数分别和文献[4]中的 step 函数、run 函数、purge 函数相对应，代表的意义是相同的，不同的是本文是从应用程序的角度来描述而文献[4]是从域的角度描述。

3.3 应用程序可信的形式化描述

应用可信包括静态可信和动态可信。静态可信是从应用程序及动态库的完整性来描述，动态可信是从应用执行动作前后的状态变化来描述。

系统的运行过程可以看成是一系列状态的转移，而系统状态的转移又是建立在执行动作的基础上[9]。因此，如果当前的系统状态是可信的，就表明此时的应用环境是可信的。一个状态到另一个状态是由一个或多个动作构成的，并且产生一系列运行结果。一个动作也称为一个单步，因此证明单步可信和结果可信从而证明状态可信，表明应用的动态可信。

3.3.1 应用的静态可信

入侵型恶意代码的感染或传播途径一般是通过感染可执行文件或替换系统的动态库。因此，可以通过度量应用程序及动态库的完整性，保障应用的初态可信，通过这种方法实现应用的静态可信。并且操作系统以前都是静态可信的。

定义 3 某一应用程序 q 是静态可信的，当其满足如下条件：

$$\text{int}(q, \{q\}) = 1$$

定理 1 某一应用程序 q 可执行，当且仅当其是静态可信的。

证明：反证法。

设某一个允许的应用程序不是静态可信，即 $\text{int}(q, \{q\}) = 0$，则该应用不能被执行，与题设矛盾。

在传递模型中，$p_1, p_2, \cdots, p_n$ 满足下列条件：$\text{int}(p_1, \{p_1\}) = 1, \text{int}(p_2, \{p_2\}) = 1, \cdots, \text{int}(p_n, \{p_n\}) = 1$。

因此，它们是静态可信的。

根据定理 1 可知，应用程序 $p_1, p_2, \cdots, p_n$ 是静态可信的，它们可执行。

3.3.2 应用的动态可信

静态可信只是保证了应用的初态可信，对应用进行隔离，保证了应用自身运行的可信。但应用程序在运行过程中会与其它应用交互信息，如果与不可信的应用交互，在交互过程中可信状态可能发生改变。因此，为了保证应用在运行过程中的可信，提出动态可信。要证明应用是动态可信，只需证明与该应用交互的应用的所有行为都不影响该应用所观察到的输出。

在 $M1$ 中，假设 a 为 p_1 与 p_2 交互的动作，并且 $\text{proc}(a) = p_1, \text{pro}(a) = p_2, p_2 \sim> p_1$。

利用无干扰模型的安全条件[4]，定义应用运行可信。

定义 4 应用程序 q 称为运行可信，当满足

$$\text{output}(\text{go}(s_0, \alpha), a) = \text{output}(\text{go}(s_0, \text{delete}(\alpha, \text{proc}(a))), a)$$

式中，$q = \text{proc}(a)$，称该条件为应用程序 q 的运行可信条件，该定义说明可信与结果预期性的符合性。

在无干扰理论模型中，提出了单步一致性和结果一致性[4]。单步一致性能保证系统的状态不变，结果一致能保证相同的输出结果。利用这两个性质，定义单步隔离性和结果隔离性。

定义 5 称一个应用程序满足单步隔离性质，当 $s \overset{q}{=} t \Rightarrow \text{step}(s, a) \overset{q}{=} \text{step}(t, a)$，其中，$q = \text{proc}(a)$。

单步隔离性说明，当 2 个状态对某个应用 q 是观察等价的话，则在这样 2 个状态下，执行相同的单个动作系统的状态不会改变。

定义 6 当一个应用程序满足下列条件时，称该应用满足结果隔离性质，其中，$q = \text{proc}(a)$。

$$s \overset{q}{=} t \Rightarrow \text{output}(s, a) = \text{output}(t, a)$$

结果隔离性说明，当 2 个状态对某个应用 q 是观察等价的话，则在这样 2 个状态下，执行相同的动作将产生相同的输出结果。

定理 2 在具有观察等价性质的系统中，如果一个应用 q 具有结果隔离性，并且满足下列条件，则该应用运行可信。

$$\text{go}(s_0, \alpha) \overset{q}{=} \text{go}(s_0, \text{delete}(\alpha, q)) \tag{1}$$

证明：设 $q = \text{proc}(a)$，把它代入式(1)，得

$$\text{go}(s_0, \alpha) \overset{\text{proc}(a)}{=} \text{go}(s_0, \text{delete}(\alpha, proc(a)))$$

应用 q 满足结果隔离性，则有

$$\text{output}(\text{go}(s_0, \alpha), a) = \text{output}(\text{go}(s_0, \text{delete}(\alpha, \text{proc}(a))), a)$$

所以应用 q 运行可信。

在模型中，假设所有待运行的应用都隔离在不同的域中。因此应用 p_1 满足单步隔离性和结果隔离性，即

$$s \overset{p_1}{=} t \Rightarrow \text{step}(s, a) \overset{p_1}{=} \text{step}(s, a)$$

$$s \overset{p_1}{=} t \Rightarrow \text{output}(s, a) = \text{output}(s, a)$$

在系统运行时，p_1 发出了一个动作序列 α。要证明 p_1 运行可信，由定理 2 可知，即证下式成立

$$s \overset{p_1}{=} t \Rightarrow \text{go}(s, \alpha) \overset{p_1}{=} \text{go}(t, \text{delete}(\alpha, p_1)) \tag{2}$$

对 α 的长度做归纳。

当 $\alpha = \Lambda$ 时，式(2)成立。

假设 α 的长度为 n 时式(2)成立，考虑当长度为 $n+1$ 时的情况，记 $\alpha' = a \circ \alpha$，则式(2)的左边有

（下转第 107 页）

一步研究的重点，期望能够找到更好的一般性证明方法。

参 考 文 献

[1] Dolev D, Yao A. On the security of public key protocols [J]. IEEE Trans on Information Theory, 1983, 29(2): 198-208

[2] 卿斯汉. 安全协议[M]. 北京：清华大学出版社，2005

[3] 薛雨杨，周颢，赵保华. 无线局域网 802.1X 协议安全性分析与检测[J]. 西安交通大学学报，2009，43(10)：52-55

[4] 皮建勇，杨雷，刘心松，等. 一种新的安全协议及其串空间模型分析[J]. 计算机科学，2010，37(1)：118-121

[5] 张玉清，王春琳，冯登国. 运行模式法分析 ISO/IEC 密钥建立协议[J]. 通信学报，2005，26(2)：15-18

(上接第 81 页)

$$go(s, a\circ\alpha) = go(step(s,a), \alpha) \quad (3)$$

因为 $p_2 \sim\!> p_1$，由 delete 函数的定义得

右边

$$go(t, delete(a\circ\alpha, p_1)) = go(t, a\circ delete(\alpha, p_1)) = go(step(t,a), delete(\alpha, p_1)) \quad (4)$$

p_1 满足单步隔离性，因此有

$$s \overset{p_1}{=} t \Rightarrow step(s,a) \overset{p_1}{=} step(s,a)$$

根据归纳假设

$go(s,\alpha) \overset{p_1}{=} go(t, delete(\alpha, p_1))$，所以

$$go(step(s,a),\alpha) \overset{p_1}{=} go(step(t,a), delete(\alpha, p_1)) \quad (5)$$

由式(3)－式(5)可得

$$go(s, a\circ\alpha) \overset{p_1}{=} go(t, delete(a\circ\alpha, p_1))$$

所以，当序列长度为 $n+1$ 时式(2)成立。

于是式(2)对任意长度的动作序列都成立。

令 $s=t=s_0$，由式(2)可得

$$go(s_0,\alpha) \overset{p_1}{=} go(s_0, delete(\alpha, p_1))$$

由于应用程序 p_1 满足结果隔离性质，根据定理 2，可得应用程序 p_1 是运行可信的。

同理可证应用程序 $p_2, p_3, \cdots, p_n$ 是运行可信的。

应用可信的概念包括：应用的静态可信和动态可信。因此应用程序 $p_1, p_2, \cdots, p_n$ 是可信的，即 $trust(p_1)=True$，$trust(p_2)=True, \cdots, trust(p_n)=True$。

3.4 应用环境的可信

在系统 $M1$ 中有 n 个运行的应用程序，任意时刻，可以有一个或多个应用同时运行。由此，可以认为如果系统中的所有应用都可信，就表明系统 $M1$ 是可信的。

定义 7 若一个应用程序 q 是可信的，则系统 $M1$ 的状态 $s\in S_T$ 执行 q 后的状态也是可信的，即

$$step(s,q) = \begin{cases} s_1 s_1 \in S_T, & \text{if } trust(q)=True \\ False, & else \end{cases}$$

定理 3 若系统 $M1$ 中执行的所有应用程序都可信，则系统 $M1$ 是可信的。

证明：在系统 $M1$ 下，因为 $p_1, p_2, \cdots, p_n \in P$，并且 $trust(p_1)=True, trust(p_2)=True, \cdots, trust(p_n)=True$，则

$$\begin{aligned} go(s_0, (p_1\circ p_2\circ\cdots)) &= go(step(s_0,p_1), (p_2\circ\cdots)) \\ &= go(s_{p_1}, (p_2\circ\cdots)) \\ &= go(step(s_{p_1}, p_2), \cdots) \\ &= go(s_{p_1 p_2}, \cdots) \\ &\cdots \\ &= s_n \end{aligned}$$

所以系统 $M1$ 是可信的。

推论 1 信任链传递模型能够保证应用的静态可信和动态可信，保证信任链在应用环境中传递，从而保证了整个应用环境的可信。

证明：由定理 1、定理 2 以及定理 3 可知，该信任链传递模型可以保证任何一新运行应用的静态可信和动态可信，从而使应用环境中的信任链传递至该应用，保证了整个应用环境的可信。

结束语 本文提出了一个满足通用的生产信息系统的动态化的信任链传递模型，并基于无干扰理论将系统抽象为应用程序、动作和状态输出，利用应用间的干扰性对提出的信任链传递模型进行了形式化描述和证明。

参 考 文 献

[1] Trusted Computing Group. TCG Specification Architecture Overview, Version1. 2 [OL]. http://www.trusted_computinggroup.org, 2003

[2] 沈昌祥，张焕国，冯登国，等. 信息安全综述[J]. 中国科学 E 辑：信息科学，2007，37(2)：129-150

[3] Trusted Computing Group. TCG Specification Architecture Overview[EB/OL]. https://www.trustedcomputinggroup.org/groups/TCG_1_2_Architecture_Overview.pdf, 2007-8-8

[4] Rushb J. Noninterference, transitivity, and Channel-Control Security policies [R]. CSL-92-02. Menlo Park: Stanford Research Institute, 1992

[5] 黄强. 基于可信计算的终端安全体系结构研究[D]. 武汉：海军工程大学，2007

[6] Goguen J A, Meseguer J. Security policies and security models [C]// Proc of the 1982 IEEE Symp on Security and Privacy. Los Alamitos: IEEE Computer Society Press, 1982: 11-20

[7] Mclena J. Security model s and information flow[C]// Proc of the 1990 IEEE Symp on Research in Security and Privacy. Los Alamitos: IEEE Computer Society Press, 1990: 177-186

[8] Phalloran C O. A calculus of information flow[C]// Proc of 1st European Symp on Research in Computer Security. Berlin: Springer-Verlag. 1990: 147-159

[9] 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)：7-11

[10] 赵佳. 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

[11] 王飞，刘威鹏，沈昌祥. 应用可信传递模型研究[J]. 计算机工程与应用，2007，43(29)：1-3

[12] 谭良，徐志伟. 基于可信计算平台的信任链传递研究进展[J]. 计算机科学，2008，35(10)：15-18