一个基于进程保护的可信终端模型

陈 菊 谭 良

(四川师范大学计算机科学学院 四川省可视化计算与虚拟现实重点实验室 成都 610101)

摘 要 针对计算机终端是网络系统中安全风险的根源问题之一,提出了一种新的基于进程保护的可信终端模型。该模型通过进程静态、动态保护和进程间无干扰来判定系统的可信性。进程静态保护的主要功能是确保进程代码和辅助文件的完整性,进程动态保护的主要功能是防止进程运行的相关数据被篡改,进程间无干扰的功能是基于无干扰理论判断进程交互的合法性。理论分析结果表明,该模型的可信性与基于可信根的无干扰可信模型等价。但该模型不仅有效克服了基于可信根的无干扰可信模型中的可信传递函数 check()的不合理性,而且将系统的状态、动作具体化,使得该模型更直观、具体,更容易理解,与实际的终端系统更相符。

关键词 无干扰理论,进程保护,可信终端,可信根

中图法分类号 TP309 **文献标识码** A

Trusted Terminal Model Based on Process Protection

CHEN Ju TAN Liang

(Key Lab of Visualization in Scientific Computing and Virtual Reality of Sichuan, College of Computer, Sichuan Normal University, Chengdu 610101, China)

Abstract Aimed at the problem that the computer terminal is the source of the safe risk in the network system, this paper proposed a new reliable terminal model based on non-interference theory. It asserts the system's trust by the protection of static and dynamic process and non-inference among processes. The main function of static process protection is to protect the integrity of the process's code and the auxiliary file. The main function of dynamic process protection is to prevent the related data from being tampered. The function of non-inference among process is to judge the legitimacy of process alternation. The theoretical security analysis shows that the credibility of this model equals the non-interference model based on the trusted root. However, this model not only overcomes the trusted transfer function's irrationality of the non-interference trusted model based on trusted root, but also externalizes system static and action. Then it is more intuitive, concrete, easier to understand and in line with the actual terminal system.

Keywords Non-interference theory, Process protection, Trusted terminal, Trusted root

1 引言

随着科学技术的发展,互联网已深深渗入到人们的生活之中。互联网的信息系统由服务器、网络、终端组成。从已有的研究和技术来看,人们更加重视服务器和网络的保护,忽视了对终端的保护,这样极不合理。首先,终端往往是创建和存放重要数据的源头;其次,绝大多数攻击事件都是从终端发起的。因此有必要大力加强终端可信的研究,将互联网的安全从根源抓起。

目前已有终端可信模型这方面的研究成果,如基于概率统计的信任模型[1,2]、基于模糊数学的信任模型[3]、基于主观逻辑的模型[4]、基于证据理论的信任模型[5]等,其中无干扰模型是取得成果最多的一种。无干扰模型也称为"完美模型",能够确保计算机系统的安全性和完整性。现有的终端无干扰可信模型多以 Rushby 的无干扰理论[6]为基础而扩展。如基于进程的无干扰可信模型[7]、基于无干扰理论的可信链模型[8],这两个模型在 Rushby 的理论上将其定义的系统安全域集实体化为进程集,整个系统就抽象为进程、动作、状态和输出,并且给出了进程运行的可信条件形式化的定义和描述,利用进程运行可信,推导出系统运行可信定理,保证了终端的安全。但它们缺少对进程的动态保护,定义的系统动作和输出过于抽象,很难和实际的终端系统相对应,且模型中的可信传递函数 check()、clear()有待进一步证明;基于非传递的无干扰理论的二元多级安全模型研究[9],所利用的非传递无干扰理论是在 Rushby 的基础上修改了清除函数,从而将 Rushby 传递的无干扰理论过渡到非传递性的无干扰理论。并且该模型分别依据 BLP 和 Biba 模型的思想保护信息的机密性和完整性,对模型进行了严格的形式化描述,证明了其安全性,但是它同样存在文献[7]的问题;文献[10]是对基于无干扰原理的终端安全模型的研究,该模型以终端的访问行为为

到稿日期:2010-05-17 返修日期:2010-09-27 本文受国家自然科学基金面上项目(60970113),四川省科技厅项目(2008JY0105-2),四川省教育厅项目(07ZA091)资助。

陈 菊(1987—),女,硕士,主要研究方向为网络安全;谭 良(1972—),男,博士,教授,主要研究方向为信息安全、网络计算。

基本元素，详细讨论了访问行为安全应满足的条件，指出安全策略与隔离性是保障终端安全的根本，并在此基础上扩展到整个终端安全。然而，该终端安全模型解决的是终端安全，而不是可信，仍然沿用传统的访问控制思想，应用无干扰理论仅仅是解决进程的隔离性。

因此本文提出了一个新的基于进程保护的可信终端模型，模型对进程进行了静态保护和动态保护，进程之间存在的相互干扰利用无干扰理论证明其之间的干扰策略是安全的。

2 一个新的基于进程保护的可信终端模型

2.1 基于进程保护的可信终端模型

进程是在麻省理工学院的 MULTLCS 系统 IBM 公司的 CTSS/360 系统中引入的，主要是用于表示应用程序在内存环境中的执行情况，它是系统资源分配的基本单位。在本文中，我们把进程看作是运行的程序，那么计算机从开机启动到操作系统加载成功这个过程中所运行的程序，也可以广义地认为是进程。因此，整个系统也就可以看成是一系列的进程在运行。下面给出可信系统的形式化定义。

定义 1 一个系统 $M=\{P,A,S,O,R_{i\in[1,2]},F_{j\in[1,7]}\}$，其中 P 为系统进程集合，分为静态进程和动态进程。所谓静态进程，即在系统 t 时刻磁盘存储介质上未运行的程序，表示为：

$$P_t^s=\sum_{i=1}^{n}p_i^s$$

式中，n 是正整数，p_i^s 表示第 i 个静态进程(程序)。所谓动态进程，即在系统 t 时刻已启动的进程集合，表示为：

$$P_t^d=\sum_{i=1}^{m}p_j^d$$

式中，m 是正整数，p_j^d 表示第 j 个动态进程。由此可得在系统 t 时刻的进程总数：

$$P=P_t^s+P_t^d$$

A 为系统的动作集，表示为各进程动作之和。

$$A=\sum_{i=1}^{k}a_{p_j^d}$$

式中，k 为正整数，$a_{p_j^d}$ 表示为进程 p_j^d 的动作集，$a_{p_j^d}^t$ 表示进程 p_j^d 在 t 时刻的动作。从微观来看，系统在 t 时刻的动作即为某进程在该时刻的动作 $a_{p_j^d}^t$。S 为系统在 t 时刻的状态集合，表示为各个进程的状态之和：

$$S=\sum_{i=1}^{L}s_{p_j^d}^t$$

式中，L 为正整数，$s_{p_j^d}^t$ 表示第 j 个动态进程 p_j^d 在 t 时刻的状态。O 为系统的输出结果集合，表示为所有进程的输出结果之和：

$$O=\sum_{i=1}^{X}o_{p_j^d}$$

式中，X 为正整数，$o_{p_j^d}$ 表示为进程 p_j^d 的输出结果集，$o_{p_j^d}^t$ 表示进程 p_j^d 在 t 时刻的输出结果。从微观来看，系统在 t 时刻的输出结果集即为某进程在该时刻的输出结果 $o_{p_j^d}^t$。

系统 M 含有两种关系：

R_1：$\approx$为进程集合 $P\times P$ 上的关系，当 $p\approx q$ 时表示进程 p 的执行会对进程 q 的执行产生影响。当！$p\approx q$ 时表示进程 p 的执行对进程 q 的执行毫无影响。

R_2：$\overset{p}{=}$为进程集合 $S\times S$ 上的一个关于系统状态的观察等价关系，$s\overset{p}{=}t$ 表示从进程 p 的角度观察，系统状态 s 和 t 是相等的。

系统 M 含有 6 个函数：

F_1：函数 $step$　$S\times A\to S$ 表示系统运行一个动作后的状态变迁。

F_2：函数 run　$S\times A^*\to S$ 表示系统在运行一个动作序列后的状态变迁。因此可以得到下面的公式：

$run(s,\Lambda)=s$

$run(s,a\circ\alpha)=run(step(s,a),\alpha)$

$run(s,a\circ\alpha)=step(run(s,\alpha),a)$

式中，Λ 表示一个空的动作序列，$\circ$表示动作间的连接操作。

F_3：函数 pro　$A\to P$ 表示一个动作所属的进程，即该动作的发出者。

F_4：函数 $output$　$S\times A\to O$，如 $s\in S,a\in A$，则 $output(s,a)$表示执行完动作 a 后系统的输出结果。

F_5：函数 sh　$P\to O$ 表示对进程 P 启动前的整个源程序进行摘要后的输出结果。

F_6：函数 dh　$P\to O$ 表示对进程 P 的运行代码页进行摘要后的输出结果。

F_7：函数 del　$A^*\times P\to A^*$，对于 $\forall p\in P$ 和一个动作序列 $\alpha\in A^*$，$del(\alpha,p)$表示删除动作序列 α 上所有由不干扰进程 P 的进程所发出的动作后剩余的序列。并且有：

$del(\Lambda,p)=\Lambda$

$$del(\alpha\circ a,p)=\begin{cases}del(\alpha,p)\circ a, & \text{if } pro(a)\approx p\\ del(\alpha,p), & \text{oherwise}\end{cases}$$

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。

定义 2 设 $p_i^s\in P$，当 p_i^s 在启动之前满足下列条件时，称为系统对 p_i^s 进行了静态保护。

$sh(p_i^s)=h$

h 表示进程 p_i^s 启动前整个源程序摘要的期望值。

定义 3 设 $p_j^d\in P$，当 p_j^d 在运行时满足下列条件时，称为系统对 p_j^d 进行了动态保护。

$dh(p_j^d)=m$

m 表示进程 p_j^d 运行时代码页摘要的期望值。

定义 4 当满足如下条件时，称进程 $pro(a)$与其他进程无干扰。

$output(run(s_0,\alpha),a)=output(run(s_0,del(\alpha,pro(a))),a)$

式中，s_0 是初始状态，$a\in A,\alpha\in A^*$。

定义 5 可信状态的定义

(1)$\phi\in S$为可信状态(ϕ为空状态)；

(2)$s=run(\phi,\alpha)\in S$为可信状态，当且仅当 α 均为合法、可预测动作，即这个动作序列的动作发出者(进程)均可信。

可以认为系统在刚刚加电的瞬间是空状态 ϕ，初始化后的状态 s_0 是唯一确定的。如果 s_0 满足可信状态的定义，即初始化进程可信，那么就可以称这一特殊的可信状态 s_0 为该系统的可信根。值得注意的是，通过定义 2 可以保证 s_0 是可信的。下面的描述均从可信根 s_0 出发。

定理 1 可信终端系统，当满足下面 4 个条件时：

(1)s_0 是可信根；

(2)系统中所有处于非运行状态的进程满足 $sh(p_i^s)=h$，即对进程静态保护；

(3)系统中处于运行状态的进程满足 $dh(p_j^d)=m$，即对

进程动态保护；

(4) 系统中当前运行进程与其他进程无干扰；

则该终端系统是可信系统。

证明：假如 t_0 时刻系统的状态为 s_0，s_0 为可信初始状态。在 $t_0+\Delta t$ 时刻，系统启动了 n 个进程($p_1^d, p_2^d, \cdots, p_n^d$)，此时系统的状态为 $S_{t_0+\Delta t}=\sum_{i=1}^{n} s_{p_j^d}^{t_0+\Delta t}$

由条件(2)得

$sh(p_i^s)=h$

由条件(3)得

$dh(p_j^d)=m$

则各进程($p_1^d, p_2^d, \cdots, p_n^d$)按照合法、可预测的方式运行。

由条件(4)得 $output(run(s_0,\alpha),\alpha)=output(run(s_0,del(\alpha,pro(\alpha))),\alpha)$，则各进程($p_1^d, p_2^d, \cdots, p_n^d$)之间无恶意干扰。由定义 5 得 $t_0+\Delta t$ 时刻系统的状态 $S_{t_0+\Delta t}$ 是可信状态。

将可信状态 $S_{t_0+\Delta t}$ 看作是下一可信根，则依此类推，同理可证明 $t_0+\Delta t+\Delta t$ 时刻的系统状态仍为可信状态。

综上所述，该终端系统任意时刻的状态均为可信状态，因此该终端系统是可信终端系统。

下面详细介绍进程的静态保护、动态保护。

2.1.1 *进程静态保护 p_i^s*

本文采用文件系统完整性保护思想来保护系统中没有运行的进程。通常病毒和木马会修改静态进程的执行文件或所关联的动态链接库，以便被修改的程序执行时病毒和木马获得系统的控制权。因此保证进程静态存储的完整性是必要的，保证进程的静态完整性即需要保证进程磁盘存储介质上的执行文件和运行时辅助文件的完整性。下面根据文献[11]对进程静态保护进行概括。

我们用 $p_1^s, p_2^s, \cdots, p_i^s$ 来表示静态进程，并用集合($f_{i1}, f_{i2}, \cdots, f_{in}$)表示与 p_i^s 相关的文件，其中 i 表示静态进程号，n 表示文件号。多个进程可能共享一个动态链接库，因此 $f_{i_k j_k}$ 和 $f_{i_l j_l}$ 可能是同一个文件。

由此我们定义文件集合(应包括执行文件和辅助文件) $F_{ij}=\bigcup_{k=1}^{i-1}(\bigcup_{l=1}^{n} f_{kl})\cup\bigcup_{l=1}^{j-1} f_{il}$。当检查静态进程 p_i^s 的第 j 个文件 f_{ij} 时，集合 F_{ij} 包含静态进程 p_1^s 到 p_{i-1}^s 的所有相关文件，以及进程 p_i^s 已经检查了的文件 f_{i1} 到 f_{ij-1}。

定义辅助函数 $Q(f_{ij})$：

$$Q(f_{ij})=\begin{cases}1, & f_{ij}\text{是}p_{ij}\text{的相关文件，且 } f_{ij}\notin F_{ij}\\ 0, & \text{否则}\end{cases}$$

因此，可以知道 m 个静态进程要保护的文件数为 $\sum_{i=1}^{m}\sum_{j=1}^{n} Q(f_{ij})$，并且每个文件都要用 HASH 算法计算文件的摘要值 $H(f_{ij})$并保存起来。

定义 6 静态进程 p_i^s 的完整性度量值为 $I(p_i^s)$，可由下式获得：

$$I(p_i^s)=H(H(f_{i1})|H(f_{i2})|\cdots|H(f_{im}))$$

首先将静态进程所有的相关文件分别用 HASH 算法摘要一次，(符号"|"为字节串的连接操作)，然后将所有摘要值连接起来再次摘要，得到进程的完整性度量值。

在完成了静态进程 p_i^s 的文件 f_{ij} 的完整性度量值 $H(f_{ij})$ 的计算后，将其保存到检测模块管理的磁盘文件中，每一个被保护的静态进程的完整性度量值均被放入 USBKey 中。USBKey 可以被用户随身携带，保护的进程在启动之前都需要按照上述过程进行检测，并和存放在 USBKey 当中的完整性度量值进行比对。这一过程称为进程的完整性认证。通过认证进程的完整性，用户能够确认自己启动的静态进程是否安全。

初始进程运行后得到的状态 s_0 利用静态保护就可以保证可信，因为初始进程运行时只有它一个进程在运行，不存在进程之间的干扰，且电脑每次启动只运行一次。该进程运行前对其完整性度量值进行对比，通过则证明它在非运行状态时没有被篡改，当它运行完一次后又被静态保护起来。

2.1.2 *进程动态保护 P_j^d*

进程静态完整性保护能够确保进程处在非活跃状态时的进程代码和辅助文件的数据的完整性，但它不能确保进程加载到内存后不被篡改。保证进程在活跃状态下的完整性，需要采用进程动态完整性保护技术。下面根据文献[11]对进程动态保护进行概括。

本文讨论的进程内存保护采用监控内存引用的办法：活跃在系统中的进程 P_j^d 的私有空间可看作由一系列(设最多 n 个)内存页面组成：$G_1, G_2, \cdots, G_n$。每一个页面中的数据都对应一个完整性度量值 $h_1, h_2, \cdots, h_n$。如果进程全部加载到内存中，则进程在内存中的完整性度量值 $M(P_j^d)$为：

$$M(P_j^d)=H(h_1|h_2|\cdots|h_n)$$

系统实现一个内核态模块监控操作系统中进程的创建、调度和内存的引用过程如下：

1)当进程被创建和调度时，操作系统将该进程的执行代码从磁盘调入内存后，监控模块即对这些进程占用的内存页进行登记，对每一个页面进行完整性度量并记录度量结果，即是上述的 $h_1, h_2, \cdots, h_n$。

2)系统运行过程中，一旦监控模块发现内存引用的区域是进程敏感区域，首先判断该次引用是否为修改：

①如果是修改，则继续判断引用主体是否为包含该内存区域的进程。如果是，说明是进程代码对自身的修改，则在完成进程修改页面后重新计算该页面的完整性度量值并保存下来；否则，拒绝对该页面执行修改。

②如果不是修改而是读取：页面引用前验证其完整性。验证通过，则允许进程继续执行；否则，通知用户当前进程被篡改并结束进程的执行。

进程动态完整性保护模型通过进程体在内存中的"写保护"阻断了不同进程间的随意篡改，杜绝了木马和病毒在内存中对进程的直接破坏和附着。同时该模型通过进程的"读校验"机制确保了进程自身的完整性，消除了黑客利用系统漏洞在进程中插入代码的威胁。由于进程结束时对应的内存区域将被操作系统释放，进程下一次启动时将重新从磁盘读取执行体并被操作系统重新分配新的内存区域，因此进程动态保护不需要对进程的内存区域进行备份。

3 安全性分析和相关工作的比较

从系统微观角度看，任意时刻，系统中只可能有一个进程在运行，只要保证系统正在运行的进程可信，就可以保证系统的可信。怎样保证正在运行的进程是可信的？本文主要是从 3 个方面来保证进程可信。首先进程处于非运行状态的时候对其静态保护，防止恶意代码的修改；其次是进程运行时对其

(下转第 150 页)

[10] Ye M, Li C F, Chen G H, et al. EECS: an energy efficient clustering scheme in wireless sensor networks[C]// Proceedings of the IEEE International Performance Computing and Communications Conference(IPCCC). 2005:535-540

[11] 刘明,曹建农,陈贵海,等. EADEEG:能量感知的无线传感器网络数据收集协议[J]. 软件学报,2007,18(5):1092-1109

[12] Soro S, Heinzelman W B. Prolonging the lifetime of wireless sensor networks via unequal clustering[C]// Proceedings of the 19th IEEE international on Parallel and Distributed Processing Symposium. San Francisco: IEEE Computer Society Press, 2005:236-240

[13] 李成法,陈贵海,叶懋,等. 一种基于非均匀分簇的无线传感器网络路由协议[J]. 计算机学报,2007,30(1):27-36

[14] Bhardwaj M, Garnett T, Chandrakasan A. Upper bounds on the lifetime of sensor networks[C]// Proceedings of the IEEE International Conference on Communications(ICC 2001). 2001:785-790

[15] 陈少华. 无线传感器网络的数据存储与查询技术[J]. 重庆工学院学报:自然科学版:2009,23(1):93-97

(上接第 117 页)

动态保护,最后保证运行的进程之间不存在恶意干扰。

1)进程的静态保护:系统将进程所有相关文件 HASH 后生成的摘要值再次 HASH 放入 USBkey 中,待下次用户使用电脑时将 USBkey 插入。系统重复上述过程,计算出 HASH 值与 USBkey 中的 HASH 值加以比较。如果相等,证明在系统没有使用的过程中进程没有遭到恶意修改,则可以让其运行。

2)进程的动态保护:进程在运行时主要是保护进程在内存中的安全,通过对比进程在内存当中所有页面的摘要值来防止其他进程修改该进程。进程动态保护和静态保护的作用是将进程自身保护起来。

3)进程间的无干扰策略:在终端系统对进程进行了静态保护和动态保护后还不能确保终端系统的可信,首先静态保护和动态保护针对的是已经存在于系统当中的进程。如果现有一新的应用进程需要创建并运行,就不能正确地判断这个进程是合法还是非法。其次进程可以看作是由执行代码、内部数据和外部 IO 组成的,外部 IO 可以是用户输入、网络传送和文件操作,通过进程静态保护和动态保护可以保证进程自身的完整性。但如果有恶意进程修改了其他进程的外部 IO 的数据,就无法保证该进程的可信。再者动态保护的任务主要是保护进程自身,阻止其他进程对自己修改。如果多个进程通过协作共同完成一个任务,需要共享内存里某段数据,假如一个进程已被病毒感染成为危险进程,它对共享数据进行了改动,那么其他共享该数据的进程可能被破坏。

本文利用进程间的无干扰策略确保进程之间的干扰均为合法来解决以上问题。进程间的无干扰策略如定义 4 所述,如果满足定义 4,则可以说进程 $pro(a)$ 与已启动的进程无恶意干扰。与此同时,还可以判断一个新创建的进程是否合法。通过定义 4 可以看出,如果新创建的进程跟其他进程没有恶意干扰,就可以认为该进程是运行可信的,允许其运行。

与文献[7,8]相比,本文的可信模型具有如下优点:

①它的系统状态 S 由进程状态组成,文献[7,8]所述的系统状态 S 没有给出明确定义,很抽象,难以和真实终端系统的实现过程相对应,相对于系统状态,进程状态更加实体化。

②它的动作集合 A 是指进程的动作,同样文献[7,8]中的动作集合指的是系统的动作集合。系统动作也是一个很模糊的概念,相比之下进程的动作更易理解。如在一个正在运行当中的 word 中插入一张图片,插入图片就是 word 进程的动作。

③它的可信是让终端系统所能达到的系统状态 S 都是可信状态。由于系统状态是由进程状态组成的,保证系统状态可信主要是通过保证进程的可信来实现。系统状态的变化过程为 $S_0 \rightarrow S_1 \rightarrow \cdots \rightarrow S_t$,从可信根 S_0 开始,如果每个运行进程可信,那么由这些进程的状态所组成的系统状态也均为可信,这样就形成了一个可信链。这和文献[7,8]的可信链是相同的,所达到的效果也是等价的,不同的是可信链的实现方式不同。

本文可信链是通过对进程静态保护、动态保护和进程间无干扰来保证进程可信,从而实现系统状态可信。文献[7,8]使用可信传递函数和无干扰理论来传递进程的可信,从而保证系统状态的可信,但它们所使用的可信传递函数(check()、clear()函数)还有待进一步证明。

结束语 本文提出了一个新的基于无干扰理论的可信终端模型,它从进程数据和代码完整性检测出发,利用无干扰理论保证进程之间的操作都是合法的。该系统可以在不安全的操作系统中建立安全的应用支撑,排除病毒和木马对关键应用程序的破坏。相对文献[7-9]的无干扰模型,本文的可信模型更容易与现实终端系统相对应,实现起来更容易,可用性更强。终端保证可信后,可以进一步扩展到网络可信和服务器可信,最终结合起来共同构成整个网络系统可信。

参 考 文 献

[1] Patel J, Teacyw T, Jennings N R, et al. A probabilistic trust model for handing inaccurate reputation sources[A]// Third International Conference, trust management[C]. Paris, France, 2005: 193-209

[2] Beth T, Borcherding M, KLEIN B. Valuatilon of thust in open-network [A]// Proceedings of the European Symposium on Research in Security(ESORICS)[C]. Brighton: Springer-Verlag, 1994:3-18

[3] 唐文,陈钟. 基于模糊集合理论的主管信任管理模型研究[J]. 软件学报,2003,14(8):1401- 1408

[4] Audu J. An algebra for assessing trust in certi- fication cha- ins [EB/OL]. http://sky. fit. qut. edu. au- /~josang/papers /jos-1999-NDSS. Pdf, 1999

[5] 袁禄来,曾国荪,王伟. 基于 DemPster-shafer 证据理论的信任评估模型[J]. 武汉大学学报:理学版,2006,52(5):627-63

[6] Rushby J. Noninterferenee, transitivity, and channel-contr- ol security Policies[R]. CSL-92-02, Menlo Park: Stanford Research Institute, 1992

[7] 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):3-11

[8] 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

[9] 刘威鹏,张兴. 基于非传递无干扰理论的二元多级安全模型研究[J]. 通信学报,2009,30(2):52-58

[10] 王飞,刘毅,李勇. 基于无干扰原理的终端安全模型研究[J]. 武汉大学学报:信息科学版,2008,33(10):1092-1094

[11] 任江春. 系统可信赖安全增强关键技术的研究与实现[D]. 长沙:国防科学技术大学,2006

[12] 周伟,尹青,郭金庚. 计算机安全中的无干扰模型[J]. 计算机科学,2005,32(2):159-165