一种基于系统动作的非确定不干扰模型

司天歌, 谭智勇, 刘 铎, 戴一奇
(清华大学计算机科学与技术系, 北京 100084)

摘 要: 本文提出一种基于系统动作的非确定不干扰模型, 把不干扰关系拓展到系统动作之间, 并表明信息流的产生同时依赖于发起者和观察者的动作, 可通过允许发起者动作而阻止观察者动作的方法避免信息流动. 最后设计了一个多级安全系统, 并为排除隐蔽信道提供了一种新的方法.

关键词: 计算机网络安全; 不干扰模型; 访问控制
中图分类号: TP393.08 文献标识码: A 文章编号: 0372-2112 (2008) 11-2205-05

A Noninterference Model Based on Actions for Nondeterministic Systems

SI Tian-ge, TAN Zhi-yong, LIU Duo, DAI Yi-qi
(*Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China*)

Abstract: A noninterference model based on actions for nondeterministic systems was developed to enforce information confidentiality. With redefined noninterference relationship on system actions, information flows depend on actions of both initiators and observers, and can be stopped by allowing actions of initiators and denying the following ones of observers. To show usability of new noninterference relationship and model, an example multilevel security system was designed and a new method was provided to excluding covert channels.

Key words: computer network security; noninterference model; access control

1 引言

多级安全模型是在安全系统中应用最广泛的一类安全策略. 从信息流的角度而言, 多级安全策略可定义为防止数据从高密级主体流向低密级主体. 而不干扰模型将信息流理解为: 如果安全域 v 观察不到安全域 u 的任意行为产生的结果, 则没有信息从 u 流向 v, 即 u 不干扰 v[1~3]. 显然, 不干扰模型对安全策略的描述更加接近信息流的实质, 因此可从不干扰模型入手证明多级安全模型的安全性. 目前, 不干扰模型主要用于基于语言的形式化验证[4].

现有的不干扰模型通常要求不干扰关系与状态无关. 但这种要求过于严格, 而且不符合实际系统的运行情况. 例如, 设 u 发起动作 a 产生的输出可被 v 通过发起动作 b 看到, 但在系统的运行中 a 或者 b 若未被执行, 那么在事实上 u 并未干扰 v. 因此, 文献[5]把不干扰关系扩展为系统动作对安全域的关系, 但是文献[5]对不干扰关系的扩展只针对发起方, 未针对受动方.

另一方面, 目前的安全模型都是通过引用监控机检查主体在当前状态下所执行的动作是否满足安全策略以决定该动作是否执行. 但从信息流角度而言, 系统拒绝执行的动作不一定会导致泄密, 例如, 在支持返回客体"不存在"错误的系统中, 由于低密级主体访问不存在的客体时得到了"不存在"的返回值, 则高密级主体可通过有规律地创建和删除某客体向低密级主体隐蔽地泄露信息[6]. 为避免这种存储隐蔽信道, 现有不干扰模型只能阻止高密级主体创建和删除客体的操作, 使得高密级主体的正常行为被拒绝执行, 降低了系统的可用性.

针对以上两个问题, 本文首先提出一个基于系统动作的非确定不干扰模型, 该模型把不干扰概念拓展到系统动作之间, 以支持比较复杂的多级安全策略的描述. 随后, 以一个多级安全模型为例, 利用提出的不干扰模型对该安全模型进行了证明. 通过限制后续动作的方法, 该示例模型支持系统返回"不存在"的错误, 允许高密级主体创建和删除客体, 但不会向低密级客体泄露信息, 从而阻止了这类存储性隐蔽信道.

2 一个基于系统动作的非确定不干扰系统

不干扰模型是一个基于状态机的信息流模型, 最早在文献[1]中提出, 而后经过不断改进, 本文所使用的符

收稿日期: 2007-08-13; 修回日期: 2008-04-28
基金项目: 国家自然科学基金(No. 60673065)

号主要来自文献[2]. 非确定系统被定义为执行同一组动作序列可以产生不同的输出. 本文所定义的非确定系统与文献[5]类似: 系统对可执行动作的选择是不确定的, 但每个动作的执行结果是确定的.

定义 **1** 系统 M 包括以下集合与函数:

(1) 系统状态集合 S, 初始状态 $s_0 \in S$;

(2) 安全域集合 D, 表示系统中的主体;

(3) 系统动作集合 A;

(4) 系统输出集合 O;

(5) 状态转换函数 step: $S \times A \mid\!\rightarrow S$;

(6) A^* 表示由集合 A 中元素组成的系统动作序列, 运行函数 run: $S \times A^* \mid\!\rightarrow S$, $\mathrm{run}(s, a \circ \alpha) = \mathrm{run}(\mathrm{step}(s, a), \alpha)$, $\mathrm{run}(s, \Lambda) = s$, Λ 表示空动作串;

(7) dom: $A \mid\!\rightarrow D$, $\mathrm{dom}(a)$ 表示动作 a 的发起者;

(8) output: $S \times A \mid\!\rightarrow O$, $\mathrm{output}(s, a)$ 表示状态 s 下动作 a 产生的输出;

(9) enabled: $S \times A \mid\!\rightarrow \{\mathrm{true}, \mathrm{false}\}$, $\mathrm{enabled}(s, a)$ 表示在状态 s 下可发起动作 a;

(10) 定义 $S \times A$ 上的二元关系 $\rightarrow$, $(s, a) \rightarrow (t, b)$ 表示在状态 s 时发起的动作 a 所产生的输出在状态 t 时被动作 b 的发起者观察到;

(11) 函数 purge: $S \times A^* \times A \mid\!\rightarrow A^*$ 表示可执行动作序列的提取操作. 对 $a, b \in A$, $\alpha \in A^*$, $s \in S$, purge 函数可递归的定义为

$$\mathrm{purge}(s, \Lambda, b) = \Lambda$$

$$\mathrm{purge}(s, a \circ \alpha, b) = \begin{cases} a \circ \mathrm{purge}(\mathrm{step}(s, a), \alpha, b), \\ \qquad \text{if } (\mathrm{run}(s, \alpha), a) \rightarrow (\mathrm{run}(s, a \circ \alpha), b) \\ \mathrm{purge}(s, \alpha, b), \quad \text{if } (\mathrm{run}(s, \alpha), a) \not\rightarrow (\mathrm{run}(s, a \circ \alpha), b) \end{cases}$$

对系统的一个可执行动作序列执行提取操作, 即删掉在当前状态下不影响 b 观察结果的动作, 这表明安全域 u 对安全域 v 的干扰关系在事实上是否成立与两个因素有关: u 是否执行了可干扰 v 的动作和 v 是否执行了可观察到 u 之前产生的输出结果的动作.

定义 **2** 对系统 M 和干扰关系 $\rightarrow$, 如果 $\forall \alpha \in A^*$, $a \in A$, $\mathrm{enabled}(\mathrm{run}(s_0, \alpha), a)$, 满足

$$\mathrm{output}(\mathrm{run}(s_0, \alpha), a) = \mathrm{output}(\mathrm{run}(s_0, \mathrm{purge}(s_0, \alpha, a)), a) \tag{1}$$

则 M 关于 $\rightarrow$ 是安全的.

定义 2 指出, 当系统 M 执行了动作序列 α 到达状态 s 后, 安全域 v 发起动作 a 对系统进行观察以确定序列 α 与序列 $\mathrm{purge}(s, \alpha, a)$ 是否相同, 如果不相同, 则说明在序列 α 中必然含有 $\mathrm{purge}(s, \alpha, a)$ 中不存在的动作 b, 并且 b 在某状态 t 下影响了动作 a 的观察结果, 即 $(t, b) \rightarrow (s, a)$, 但这与 purge 函数的定义矛盾, 即系统不安全. 所以, 若要系统 M 安全, 必须有式(1)成立.

为便于系统验证, 下面给出定理 1, 用于证明只涉及单步状态的系统安全展开条件.

定理 **1** 对 $\forall s, t \in S$, $\forall a, b \in A$, 若都存在 S 上的等价关系 $\overset{a}{\sim}$, 且系统 M 在运行中满足

$$(1)\ s \overset{a}{\sim} t \Rightarrow \mathrm{output}(s, a) = \mathrm{output}(t, a) \tag{2}$$

$$(2)\ s \overset{b}{\sim} t \wedge \mathrm{enabled}(s, a) \wedge \mathrm{enabled}(t, a) \Rightarrow \mathrm{step}(s, a) \overset{a}{\sim} \mathrm{step}(t, a) \tag{3}$$

$$(3)\ (s, a) \not\rightarrow (\mathrm{step}(s, a), b) \Rightarrow s \overset{b}{\sim} \mathrm{step}(s, a) \tag{4}$$

则 M 关于 $\rightarrow$ 是安全的.

在定理 1 中, $s \overset{a}{\sim} t$ 表示状态 s 和状态 t 在安全域 $\mathrm{dom}(a)$ 发起动作 a 进行观察后是等价的; 条件(1)表示在关于 a 等价的状态下发起该动作所得到的输出是相同的; 条件(2)表示在等价状态下执行相同动作后得到的状态仍然等价; 条件(3)表示如果在状态 s 下执行动作 a 不干扰 t 状态发起的动作 b, 则状态 s 与执行动作 a 得到的状态对 b 动作的发起者来说是等价的.

证明 首先证明

$$s \overset{b}{\sim} t \Rightarrow \mathrm{run}(s, \alpha) \overset{b}{\sim} \mathrm{run}(t, \mathrm{purget}(t, \alpha, b)) \tag{5}$$

对式(5)中的 α 的长度使用数学归纳法. $\alpha = \Lambda$ 时, 式(5)显然成立, 并假设式(5)在 α 成立. 考虑 $a \circ \alpha$, 根据定义有,

$$\mathrm{run}(s, a \circ \alpha) = \mathrm{run}(\mathrm{step}(s, a), \alpha) \tag{6}$$

根据归纳假设, 对任意不大于当前 α 长度的动作序列式(5)、(6)都成立, 所以必有

$$\mathrm{enabled}(s, a) \wedge \mathrm{enabled}(t, b) \tag{7}$$

对 $\mathrm{run}(t, \mathrm{purge}(t, a \circ \alpha, b))$, 有两种情况:

(1) 若 $(\mathrm{run}(t, \alpha), a) \not\rightarrow (\mathrm{run}(t, a \circ \alpha), b)$

$$\mathrm{run}(t, \mathrm{purge}(t, a \circ \alpha, b)) = \mathrm{run}(t, \mathrm{purge}(t, \alpha, b)) \tag{8}$$

根据式(4)有 $\mathrm{run}(t, \alpha) \overset{b}{\sim} \mathrm{run}(t, a \circ \alpha)$, 由归纳假设 α 的长度可知 $t \overset{b}{\sim} \mathrm{step}(t, a)$, 再根据式(3)、(5)左侧得到 $\mathrm{step}(s, a) \overset{b}{\sim} \mathrm{step}(t, a)$, 考虑到 $\overset{b}{\sim}$ 是等价关系, 所以得到 $\mathrm{step}(s, a) \overset{b}{\sim} t$, 由归纳假设有

$$\mathrm{run}(\mathrm{step}(s, a), \alpha) \overset{b}{\sim} \mathrm{run}(t, \mathrm{purge}(t, \alpha, b)) \tag{9}$$

再由式(6)、(8)得到

$$\mathrm{run}(s, a \circ \alpha) \overset{b}{\sim} \mathrm{run}(t, \mathrm{purge}(t, a \circ \alpha, b)) \tag{10}$$

(2) 若 $(\mathrm{run}(t, \alpha), a) \not\rightarrow (\mathrm{run}(t, a \circ \alpha), b)$

$$\begin{aligned} &\mathrm{run}(t, \mathrm{purge}(t, a \circ \alpha, b)) \\ &= \mathrm{run}(t, a \circ \mathrm{purge}(\mathrm{step}(t, a), \alpha, b)) \\ &= \mathrm{run}(\mathrm{step}(t, a), \mathrm{purge}(\mathrm{step}(t, a), \alpha, b)) \end{aligned} \tag{11}$$

根据式(3)、(7)和式(5)左侧得到 $\mathrm{step}(s, a) \overset{b}{\sim} \mathrm{step}(t, a)$, 根据归纳假设有

$$\mathrm{run}(\mathrm{step}(s,a),\alpha)\overset{b}{\sim}\mathrm{run}(\mathrm{step}(t,a),\mathrm{purge}(\mathrm{step}(t,a),\alpha,b)) \tag{12}$$

再由式(6)、(11)得到

$$\mathrm{run}(s,a^{\circ}\alpha)\overset{b}{\sim}\mathrm{run}(t,\mathrm{purge}(t,a^{\circ}\alpha,b)) \tag{13}$$

综合式(10)、(13), 可知式(5)在 $a^{\circ}\alpha$ 时成立. 再令 $s=t=s_0$, 并考虑到 $s_0\overset{b}{\sim}s_0$ 恒成立, 则有

$$\mathrm{run}(s_0,\alpha)\overset{b}{\sim}\mathrm{run}(s_0,\mathrm{purge}(s_0,\alpha,b)) \tag{14}$$

由式(2)、(14)得到式(1), 即定理 1 成立.

3　一个多级安全模型及其安全性证明

为说明改进后的不干扰模型在描述和证明多级安全策略方面的优势, 本节设计了一个多级安全模型, 并以此为例用改进后的不干扰模型进行证明. 该模型具有 BLP 模型的基本特点[7], 可支持主体对客体的读取、修改、创建和删除操作, 尽管该模型规定低密级主体读取不存在客体时返回客体不存在的错误, 并允许主体对高密级客体的创建和删除操作, 但仍能保证数据不被泄露.

定义 3　令 N 表示客体名字集合, V 表示客体内容集合, contents: $S\times N\to V$ 表示客体内容, observe: $A\to P(N)$ 表示动作可读取的客体集合, alter: $A\to P(N)$ 表示动作可修改的客体集合, exist: $S\times N\to\{\mathrm{true},\mathrm{false}\}$ 表示客体是否存在, 则系统的等价关系定义为

$$s\overset{a}{\sim}t\Leftrightarrow\left(\begin{array}{l}\forall n\in\mathrm{observe}(a):(\neg\mathrm{exist}(s,n)\wedge\neg\mathrm{exist}(t,n))\vee\\(\mathrm{exist}(s,n)\wedge\mathrm{exist}(t,n)\wedge\mathrm{contents}(s,n)=\mathrm{contents}(t,n))\end{array}\right)$$

系统的引用监控机假设定义为

$$(1)\ s\overset{a}{\sim}t\Rightarrow\mathrm{output}(s,a)=\mathrm{output}(t,a) \tag{15}$$

(2)

$$\left(\begin{array}{l}s\overset{a}{\sim}t\wedge\left(\begin{array}{l}\mathrm{contents}(\mathrm{step}(s,a),n)\neq\mathrm{contents}(s,n)\vee\\\mathrm{contents}(\mathrm{step}(t,a),n)\neq\mathrm{contents}(t,n)\end{array}\right)\wedge\\\left(\begin{array}{l}\mathrm{exist}(s,n)\wedge\mathrm{exist}(\mathrm{step}(s,a),n)\wedge\\\mathrm{exist}(t,n)\wedge\mathrm{exist}(\mathrm{step}(t,a),n)\end{array}\right)\end{array}\right)$$

$$\mathrm{contents}(\mathrm{step}(s,a),n)=\mathrm{contents}(\mathrm{step}(t,a),n) \tag{16}$$

$$(3)\ s\overset{b}{\sim}t\Rightarrow\left(\begin{array}{l}\forall n\in\mathrm{observe}(b):(\mathrm{exist}(\mathrm{step}(s,a),n)\\\wedge\mathrm{exist}(\mathrm{step}(t,a),n))\\\vee(\neg\mathrm{exist}(\mathrm{step}(s,a),n)\wedge\neg\mathrm{exist}(\mathrm{step}(t,a),n))\end{array}\right) \tag{17}$$

$$(4)\left(\begin{array}{l}s\overset{a}{\sim}t\wedge\neg\mathrm{exist}(s,n)\wedge\neg\mathrm{exist}(t,n)\wedge\\(\mathrm{exist}(\mathrm{step}(s,a),n)\vee\mathrm{exist}(\mathrm{step}(t,a),n))\end{array}\right)\Rightarrow$$

$$\mathrm{contents}(\mathrm{step}(t,a),n)=\mathrm{contents}(\mathrm{step}(s,a),n) \tag{18}$$

$$(5)\left(\begin{array}{l}\mathrm{enabled}(s,a)\wedge\\\mathrm{contents}(\mathrm{step}(s,a),n)\neq\mathrm{contents}(s,n)\end{array}\right)\Rightarrow n\in\mathrm{alter}(a) \tag{19}$$

与文献[5]类似, 引用监控机假设表明系统 M 中所有主体对客体的操作都无法绕开引用监控机. 其中, 条件(1)表示动作输出只与动作可观察到的对象有关, 条件(2)表示动作对客体值的改变只依赖于客体的原值和该动作观察的结果, 条件(3)表示动作对客体存在关系的改变只依赖于动作本身, 条件(4)表示新创建的客体取值只与动作观察到的结果有关, 条件(5)表示动作只能改变可修改对象的取值.

定义 4　令 L 表示密级集合, $f: N\cup D\to L$ 表示主客体的密级, $x\leftarrow y$ 表示把 y 赋给 x, 系统 M 满足引用监控机假设, 并对 $s,t\in S$, $a,b\in A$, $\mathrm{enabled}(s,a)\wedge\mathrm{enabled}(t,b)$, 满足

$$(1)(s,a)\to(t,b)\Leftrightarrow f(\mathrm{dom}(a))\le f(\mathrm{dom}(b)) \tag{20}$$

$$(2)\ n\in\mathrm{observe}(a)\Rightarrow f(n)\le f(\mathrm{dom}(a)) \tag{21}$$

$$(3)\ n\in\mathrm{alter}(a)\Rightarrow\mathrm{exist}(s,n)\wedge f(\mathrm{dom}(a))\le f(n) \tag{22}$$

$$(4)\ \mathrm{exist}(s,n)\wedge\neg\mathrm{exist}(\mathrm{step}(s,a),n)\Rightarrow f(n)\leftarrow\max\{f(n),f(\mathrm{dom}(a))\} \tag{23}$$

$$(5)\ \neg\mathrm{exist}(s,n)\wedge\mathrm{exist}(\mathrm{step}(s,a),n)\Rightarrow f(n)\leftarrow\max\{f(n),f(\mathrm{dom}(a))\} \tag{24}$$

为简化处理, 系统规定创建和删除客体的操作只影响读取操作的观察结果, 并且只有当发起的读取操作确实观察到之前创建或删除操作的结果时, 才禁止该读取操作进行. 为此, 在删除或创建某客体时, 令该客体的密级不低于主体的密级; 对读取不存在客体 n 的操作 a, 系统在 $f(n)\le f(\mathrm{dom}(a))$ 时返回不存在, 否则系统假设 n 存在并按照 $f(\mathrm{dom}(a))<f(n)$ 返回. 定义 4 给出了符合要求的操作规范, 其中, 式(20)给出了系统要求的干扰策略定义; 式(21)、式(22)给出了系统观察、修改动作的限制, 式(23)、(24)分别给出了删除和创建客体时的操作. 需要说明的是, 对于 $(s,a)\to(t,b)$, 如果在状态 s 和 t 之间存在其它动作删除或创建了 b 所观察的客体, 则该客体的密级与动作 a 无关, 并且动作 a 对 b 的干扰关系也不再成立, 因此本文不考虑这种情况.

在对定义 4 给出的系统进行安全性证明之前, 首先证明引理 1.

引理 1　若系统 M 符合引用监控机假设, 并满足

$$(1)(s,a)\to(t,b)\Rightarrow\mathrm{observe}(a)\subseteq\mathrm{observe}(b) \tag{25}$$

$$(2)\left(\begin{array}{l}\mathrm{enabled}(s,a)\wedge n\in\mathrm{alter}(a)\wedge\\\mathrm{enabled}(t,b)\wedge n\in\mathrm{observe}(b)\end{array}\right)\Rightarrow(s,a)\to(t,b) \tag{26}$$

$$(3)\left(\begin{array}{l}n\in\mathrm{observe}(b)\wedge\\\left(\begin{array}{l}(\neg\mathrm{exist}(s,n)\wedge\mathrm{exist}(\mathrm{step}(s,a),n)\wedge\mathrm{exist}(t,n))\vee\\(\mathrm{exist}(s,n)\wedge\neg\mathrm{exist}(\mathrm{step}(s,a),n)\wedge\neg\mathrm{exist}(t,n))\end{array}\right)\end{array}\right)$$

$$\Rightarrow(s,a)\to(t,b) \tag{27}$$

则 M 关于 $\overrightarrow{}$ 是安全的.

证明 只需证明引理 1 的条件符合定理 1 的式(2)~(4)即可. 其中, 式(2)可由式(15)直接得到, 下面证明式(3)和(4).

(Ⅰ)式(3)证明过程: 根据定义, 式(3)可重写为

$$s \overset{a}{\sim} t \wedge \mathrm{enabled}(s,a) \wedge \mathrm{enabled}(t,a) \Rightarrow \forall n \in \mathrm{observe}(b): (\neg \mathrm{exist}(\mathrm{step}(s,a),n) \wedge \neg \mathrm{exist}(\mathrm{step}(t,a),n)) \vee \begin{pmatrix} \mathrm{exist}(\mathrm{step}(s,a),n) \wedge \mathrm{exist}(\mathrm{step}(t,a),n) \wedge \\ \mathrm{contents}(\mathrm{step}(s,a),n) = \mathrm{contents}(\mathrm{step}(t,a),n) \end{pmatrix} \tag{28}$$

对 $\forall n \in \mathrm{observe}(b)$, 根据定义有

$$\begin{pmatrix} (\neg \mathrm{exist}(s,n) \wedge \neg \mathrm{exist}(t,n)) \vee \\ (\mathrm{exist}(s,n) \wedge \mathrm{exist}(t,n) \wedge \mathrm{contents}(s,n) = \mathrm{contents}(t,n)) \end{pmatrix}$$

根据式(17)有

$$\begin{pmatrix} (\mathrm{exist}(\mathrm{step}(s,a),n) \wedge \mathrm{exist}(\mathrm{step}(t,a),n)) \vee \\ (\neg \mathrm{exist}(\mathrm{step}(s,a),n) \wedge \neg \mathrm{exist}(\mathrm{step}(t,a),n)) \end{pmatrix}$$

分情况讨论:

(1)若 $\neg \mathrm{exist}(\mathrm{step}(s,a),n) \wedge \neg \mathrm{exist}(\mathrm{step}(t,a),n)$, 则式(28)右侧成立.

(2)若 $\mathrm{exist}(\mathrm{step}(s,a),n) \wedge \mathrm{exist}(\mathrm{step}(t,a),n) \wedge \neg \mathrm{exist}(s,n) \wedge \neg \mathrm{exist}(t,n)$, 根据式(18)有 $\mathrm{contents}(\mathrm{step}(s,a),n) = \mathrm{contents}(\mathrm{step}(t,a),n)$, 即式(28)右侧成立.

(3)若 $\mathrm{exist}(\mathrm{step}(s,a),n) \wedge \mathrm{exist}(\mathrm{step}(t,a),n) \wedge \mathrm{exist}(s,n) \wedge \mathrm{exist}(t,n)$

(a)若 $\mathrm{contents}(\mathrm{step}(s,a),n) \neq \mathrm{contents}(s,n)$, 则根据式(19)有 $n \in \mathrm{alter}(a)$, 再根据式(26)有 $(s,a) \rightarrow (t,b)$, 所以由式(25)得到

$$\mathrm{observe}(a) \subseteq \mathrm{observe}(b) \tag{29}$$

根据 $\overset{a}{\sim}$ 定义可知 $s \overset{a}{\sim} t$, 再根据式(17)、(29)和式(28)左侧, 得到式(28)右侧.

(b)若 $\mathrm{contents}(\mathrm{step}(t,a),n) \neq \mathrm{contents}(t,n)$, 同(a)证明过程, 可得式(28)右侧.

(c)若 $\begin{pmatrix} \mathrm{contents}(\mathrm{step}(t,a),n) = \mathrm{contents}(s,n) \wedge \\ \mathrm{contents}(\mathrm{step}(t,a),n) = \mathrm{contents}(t,n) \end{pmatrix}$, 则根据等价关系 $s \overset{a}{\sim} t$ 的定义可知 $\mathrm{contents}(\mathrm{step}(s,a),n) = \mathrm{contents}(\mathrm{step}(t,a),n)$, 即得到式(28)右侧.

综合以上, 可证明式(28), 即式(3)成立.

(Ⅱ)式(4)证明过程: 采用反证法证明式(4), 即只需证明

$$\neg \left(s \overset{b}{\sim} \mathrm{tep}(s,a) \right) \Rightarrow (s,a) \rightarrow (\mathrm{step}(s,a),b) \tag{30}$$

式(30)左侧可重写为

$$\exists n \in \mathrm{observe}(b): (\mathrm{exist}(s,n) \wedge \neg \mathrm{exist}(\mathrm{step}(s,a),n)) \vee (\neg \mathrm{exist}(s,n) \wedge \mathrm{exist}(\mathrm{step}(s,a),n)) \vee (\neg \mathrm{exist}(s,n) \wedge \mathrm{exist}(\mathrm{step}(s,a),n) \wedge \mathrm{contents}(\mathrm{step}(s,a),n) \neq \mathrm{contents}(s,n)) \tag{31}$$

分情况讨论:

(1)若 $\mathrm{exist}(s,n) \wedge \neg \mathrm{exist}(\mathrm{step}(s,a),n)$, 根据式(27)有 $(s,a) \rightarrow (\mathrm{step}(s,a),b)$.

(2)若 $\neg \mathrm{exist}(s,n) \wedge \mathrm{exist}(\mathrm{step}(s,a),n)$, 根据式(27)有 $(s,a) \rightarrow (\mathrm{step}(s,a),b)$.

(3)若 $\mathrm{exist}(s,n) \wedge \mathrm{exist}(\mathrm{step}(s,a),n) \wedge \mathrm{contents}(\mathrm{step}(s,a),n) \neq \mathrm{contents}(s,n)$, 根据式(19)有 $n \in \mathrm{alter}(a)$, 再由式(26)得到 $(s,a) \rightarrow (\mathrm{step}(s,a),b)$.

综合以上可证明式(30), 即式(4)成立, 进而引理 1 得证.

定理 **2** 符合定义 4 的系统 M 关于 $\rightarrow$ 是安全的.

证明: 只需证明引理 1 中的式(25)~(27).

式(25)证明过程: 式(25)可重写为

$$(s,a) \rightarrow (t,b) \wedge n \in \mathrm{observe}(a) \Rightarrow n \in \mathrm{observe}(b)$$

根据 $\rightarrow$ 定义和式(21)有 $f(n) \leqslant f(\mathrm{dom}(a)) \leqslant f(\mathrm{dom}(b))$, 所以 $n \in \mathrm{observe}(b)$, 即式(25)成立.

式(26)证明过程: 根据式(21)、(22)和(23)左侧, 得到

$$\mathrm{exist}(t,n) \wedge f(\mathrm{dom}(a)) \leqslant f(n) \leqslant f(\mathrm{dom}(b))$$

再由 $\rightarrow$ 定义有 $(s,a) \rightarrow (t,b)$, 即式(26)成立.

式(27)证明过程: 式(27)左侧可重写为

$$\begin{pmatrix} (\neg \mathrm{exist}(s,n) \wedge \mathrm{exist}(\mathrm{step}(s,a),n) \\ \wedge \mathrm{exist}(t,n) \wedge f(n) \leqslant f(\mathrm{dom}(b))) \vee \\ (\mathrm{exist}(s,n) \wedge \neg \mathrm{exist}(\mathrm{step}(s,a),n) \\ \wedge \neg \mathrm{exist}(t,n) \wedge f(n) \leqslant f(\mathrm{dom}(b))) \end{pmatrix}$$

若 $\mathrm{exist}(t,n)$, 则根据式(24)有 $f(\mathrm{dom}(a)) \leqslant f(n) \leqslant f(\mathrm{dom}(b))$, 即得到式(27)右侧, 否则 $\neg \mathrm{exist}(t,n)$, 根据式(23)有 $f(\mathrm{dom}(a)) \leqslant f(n) \leqslant f(\mathrm{dom}(b))$, 也得到式(27)右侧. 所以式(27)成立.

综上可知, 定理 2 得证.

在现有的不干扰模型中, 高密级主体创建或删除客体的动作会干扰低密级主体, 因此不能授权它们执行. 但通常情况下, 这些动作中非法行为的比率远远低于合法动作, 所以拒绝它们执行会大大降低模型的可用性. 定理 2 表明, 即使这些动作被授权执行, 干扰关系仍未成立, 只有当后续的低密级主体真正发起了访问到这些客体的动作时, 干扰关系才真正成立, 因此, 可在这个时刻拒绝该动作, 以避免产生非法信息流.

4 结论及进一步工作

本文提出了一种基于系统动作的非确定不干扰模型, 通过把安全域间的不干扰关系拓展到系统动作之间, 可描述更加复杂和细致的信息流规则, 也更为接近系统的实际运行情况.

下一步我们将利用该模型对其它访问控制模型进

行分析. 同时, 考虑到实际系统的用户行为具有相互依赖关系, 所以需要进一步研究模型中系统动作的依赖关系与时序关系.

参考文献:

[1] Goguen J, Meseguer J. Security policies and security models [A]. Proceedings of the 1982 IEEE Symposium on Research in Security and Privacy[C]. Los Alamitos: IEEE Computer Society Press, 1982. 11- 20.

[2] Rushby J. Noninterference, Transitivity, and Channel-Control Security Policies[R]. Menlo Park: Stanford Research Institute, 1992.

[3] Heiko M. Unwinding possibilistic security properties[A]. Proceedings of the 6th European Symposium on Research in Computer Security[C]. Toulouse: Springer-Verlag, 2000. 238- 254.

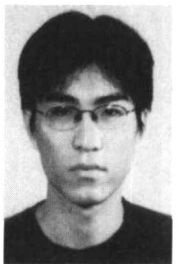
[4] Sabelfeld A, Myers A C. Language-based information-flow security[J]. IEEE Journal on Selected Areas in Communications, 2003, 21(1): 1- 15.

[5] 谢钧, 黄皓. 一个非确定系统的不干扰模型[J]. 软件学报, 2006, 17(7): 1601- 1608.
Xie Jun, Huang Hao. A noninterference model for nondeterministic systems[J]. Journal of Software, 2006, 17(7): 1601- 1608. (in Chinese)

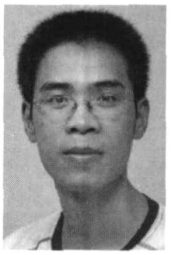
[6] 卿斯汉. 高安全等级安全操作系统的隐蔽通道分析[J]. 软件学报, 2004, 15(12): 1837- 1849.
Qing Si-han. Covert channel analysis in secure operating systems with high security levels[J]. Journal of Software, 2004, 15(12): 1837- 1849. (in Chinese)

[7] Bell D E, LaPadula L J. Secure Computer Ssytem: Unified Exposition and MULTICS Interpretation[R]. Bedford, MA: The MITRE Corporation, 1976.

作者简介:



司天歌　男, 1977 年生于辽宁, 清华大学计算机科学与技术系博士生, 主要研究方向为网络信息安全.
E-mail: sitg@theory.cs.thinghua.edu.cn



谭智勇　男, 1979 年生于四川, 清华大学计算机科学与技术系博士生, 主要研究方向为多级安全模型和网络信息安全.

刘　铎　男, 1978 年生于北京, 清华大学计算机科学与技术系博士后, 博士, 主要研究方向为公钥密码学、组合算法的设计与分析.

戴一奇　男, 1946 年生于浙江, 清华大学计算机科学与技术系教授, 博士生导师, 主要研究方向为网络信息安全.