计算机安全中的无干扰模型^{*}

周 伟 尹 青 郭金庚

(信息工程大学网络工程实验室 郑州450002)

摘 要 与基于访问控制的形式安全模型相比,基于信息流的安全模型对于定义什么是安全来说更为本质。自提出信息流的无干扰概念以来,信息流模型成为安全研究的中心之一,并提出了多种无干扰信息流模型。本文基于进程代数框架研究这些模型,并给出了一些新的结果和证明。

关键词 安全模型,信息流,无干扰,进程代数

Non-Interference Models in Computer Security

ZHOU Wei YIN Qing GUO Jin-Geng

(Network Engineering Laboratory, Information Engineering University, Zhengzhou 450002)

Abstract Compared with security models based on access control, security models based on information flow theory are more fundamental to capture the essentials of what is confidentiality. Since the Goguen-Meseguer definition of the concept of non-interference, variations of non-interference models have been presented. Adapted to process algebra framework, this paper gets into details of these models, some new proofs and new results are given.

Keywords Security model, Information flow, Non-interference, Process algebra

1 引言

自 Goguen 和 Meseguer 发现信息流的无干扰模型^[6]以来,多种关于无干扰信息流的安全模型相继发表。除 Goguen 和 Meseguer 的基本无干扰模型之外,这些安全模型还包括广义无干扰模型^[11]、不可演绎模型^[23,24,7]、解析模型^[10,8]、不可推断模型^[14,11]、隔离模型^[11]等。无干扰模型的早期研究主要是在非经典状态机模型与迹理论框架内进行的,在这方面较为完整的理论描述可见文^[25]。

1987年,Simon N. Foley^[4]使用进程代数 CSP^[16]研究信息流理论。1990年 Ryan 使用 CSP 对无干扰模型进行了重新表述^[18],此后,进程代数被作为研究信息流安全性质的合适工具之一^[19],Allen^[1]、Foster^[5]、Ryan^[16-20]、Roscoe^[15,17]、Schneider^[21]等人在 CSP 框架下对信息流安全性质进行研究,基于进程代数 CCS^[12,13]的简单扩展语言 SPA, Focardi 等人研究进程的可复合安全性质^[2,3]。除 CSP 和 CCS 之外, π 演算^[13]等其它进程代数也都被作为分析和证明安全性质的有效工具。

本文的目标是在进程代数框架内讨论安全信息流的迹语义模型,并按照下面的框架来组织。首先在第2节简要地介绍安全进程代数 SPA。第3和第4节为进一步的讨论作一点概念上的准备。从第5节到第10节,在统一的进程代数框架内讨论自 Goguen 和 Meseguer 以来所发展的各种无干扰模型及其性质。

2 安全进程代数

本文使用的安全进程代数语言称为 SPA,是 CCS 的变体。其中一个主要扩展是将有名动作集(即标记集) L 划分为

两个不相交的子集, Act_H 和 Act_L , 且 $\overline{Act_H} = Act_H, \overline{Act_L} = Act_L$ 。这样划分的目的是为了描述多级安全系统中不同安全级之间的信息流关系, Act_H 和 Act_L 分别对应于高安全级的进程动作和低安全级的进程动作。为了能够更方便地刻画信息流性质,本文在 SPA 中显式地引入了一些派生算子,包括同步并行算子、交替并行算子、隐藏算子。

2.1 SPA 的基本语法

除括号外,SPA 的基本语法符号包括:

(1) 进程动作集 Act , 它由三部分组成,无穷名集 A , A 的伴名集 $\bar{A}$, 一个称为哑名的元素 τ 。在其中,令 $L = A \cup \bar{A}$ 称为标记集,并记 $Act = L \cup \{\tau\}$ 。更进一步,SPA 将标记集划分到两个安全级:高安全级动作 Act_H 和低安全级动作 $Act_L, \overline{Act_H} = Act_H, \overline{Act_L} = Act_L$ 。这样, $Act = Act_H \cup Act_L \cup \{\tau\}$ 。

(2) 进程变量集 X , 并以 X, Y, Z 等表示进程变量,进程常量集 κ , 并以 A, B 等表示进程常量。

(3) 算子符号 $\cdot, +, |, ||, \backslash, \mu$, 分别表示前缀算子、选择(和)算子、并行算子、(并行)交替算子、限制算子和不动点算子。派生算子符号 $/, \parallel, \triangleright$ 分别表示隐藏、同步、顺序算子。

定义2.1 进程项的集合 ϵ 是如下语法构造的集合:

$$E \triangleq 0 \mid X \mid A \mid \alpha. E \mid \alpha. E_1 + \beta. E_2 \mid E_1 \mid E_2 \mid E_1 \mid \mid E_2 \mid E \setminus N \mid \mu X. E[X]$$

定义2.2 简单进程项是如下定义的不含递归的进程项:

$$E \triangleq 0 \mid \alpha. E_1 + \beta. E_2 \mid E_1 \mid E_2$$

在进程项 $\mu X. E[X]$ 中 X 是约束变量,不受约束的变量就是自由变量。进程是不含自由进程变量的进程项。我们用 P 记所有进程的集合。

2.2 SPA 的操作语义

SPA 及派生算子的操作语义规则见图1。

^{*} 基金项目:国家863计划信息安全技术主题“网络安全防御技术”(编号:863-104-06-1)。周 伟 博士研究生,研究方向:软件理论与技术,网络安全。尹 青 博士研究生,研究方向:软件理论与技术,网络安全。郭金庚 教授,博士生导师,研究方向:软件理论与技术。

前缀 $\frac{-}{\alpha.E \xrightarrow{\alpha} E}$ 和 $\frac{E \xrightarrow{\alpha} E'}{E+F \xrightarrow{\alpha} E'+F}$ $\frac{F \xrightarrow{\alpha} F'}{E+F \xrightarrow{\alpha} E+F'}$ 限制 $\frac{E \xrightarrow{\alpha} E'}{E \setminus L \xrightarrow{\alpha} E' \setminus L} (\alpha \notin L)$
并行 $\frac{E \xrightarrow{\alpha} E'}{E \mid F \xrightarrow{\alpha} E' \mid F}$ $\frac{F \xrightarrow{\alpha} F'}{E \mid F \xrightarrow{\alpha} E \mid F'}$ $\frac{E \xrightarrow{\alpha} E', F \xrightarrow{\alpha} F'}{E \mid F \xrightarrow{\tau} E' \mid F'}$
同步 $\frac{E \xrightarrow{i} E', F \xrightarrow{i} F'}{E \parallel F \xrightarrow{\tau} E' \parallel F'}$ $\frac{E \xrightarrow{\alpha} E'}{E \parallel F \xrightarrow{\alpha} E' \parallel F} (\alpha \notin L(F))$ $\frac{F \xrightarrow{\alpha} F'}{E \parallel F \xrightarrow{\alpha} E \parallel F'} (\alpha \notin L(E))$
交替 $\frac{E \xrightarrow{\alpha} E'}{E \text{ III } F \xrightarrow{\alpha} E' \text{ III } F}$ $\frac{F \xrightarrow{\alpha} F'}{E \text{ III } F \xrightarrow{\alpha} E \text{ III } F'}$ 顺序 $\frac{E \xrightarrow{\alpha} E'}{E_{\checkmark} F \xrightarrow{\alpha} E'_{\checkmark} F} (\alpha \neq \checkmark)$ $\frac{E \xrightarrow{\checkmark} E'}{E_{\checkmark} F \xrightarrow{\tau} E'_{\checkmark} F}$
递归 $\frac{E[\mu X.E[X]/X] \xrightarrow{\alpha} E'}{\mu X.E[X] \xrightarrow{\alpha} E'}$ 隐藏 $\frac{E \xrightarrow{\alpha} E'}{E/L \xrightarrow{\alpha} E'/L} (\alpha \notin L)$ $\frac{E \xrightarrow{\alpha} E'}{E/L \xrightarrow{\tau} E'/L} (\alpha \in L)$

图1 SPA及派生算子的语义规则

2.3 SPA的迹语义

引用Milner的定义^[16],设 $P \in \mathcal{P}$ 是一个进程, P 的迹定义为 $T(P) = \{\hat{\gamma} \in L^* : P \xrightarrow{\hat{\gamma}}\}$. 设 $P, Q \in \mathcal{P}$ 是两个进程, 若 $T(P) \subseteq T(Q)$, 则记 $P \subseteq_{\tau} Q$. $\xrightarrow{\tau}$ 表示转移闭包, 表达式 $P \xrightarrow{\tau} P'$ 表示 $P \xrightarrow{\tau} \cdot \xrightarrow{\tau} \cdot \dots \xrightarrow{\tau} \cdot \xrightarrow{\tau} P'$, 而表达式 $P \xrightarrow{\gamma} P'$ 表示存在一个 $\gamma, P \xrightarrow{\tau} P'$. 有时还用 $P \xrightarrow{\gamma}$ 表示存在 P' 使得 $P \xrightarrow{\tau} P'$, 并用 $P \xrightarrow{\gamma}$ 表示存在 P' 使得 $P \xrightarrow{\tau} P'$.

定义2.3 设 $P, Q \in \mathcal{P}$ 是两个进程, 称 P 和 Q 是迹等价的, 记为 $P \approx_{\tau} Q$, 当且仅当 $P \subseteq_{\tau} Q$ 且 $Q \subseteq_{\tau} P$, 即 $T(P) = T(Q)$.

引理2.1 若设 $P_1 \approx_{\tau} Q_1, P_2 \approx_{\tau} Q_2$, 则 (1) $P_1 + P_2 \approx_{\tau} Q_1 + Q_2$; (2) $P_1 \mid P_2 \approx_{\tau} Q_1 \mid Q_2, P_1 \text{ III } P_2 \approx_{\tau} Q_1 \text{ III } Q_2$; (3) $P_1 \setminus L \approx_{\tau} Q_1 \setminus L, P_1 / L \approx_{\tau} Q_1 / L$.

引理2.2 (1) $\alpha.(P+Q) \approx_{\tau} \alpha.P + \alpha.Q$; (2) $(P+Q) \mid R \approx_{\tau} P \mid R + Q \mid R, (P+Q) \text{ III } R \approx_{\tau} P \text{ III } R + Q \text{ III } R$; (3) $(P+Q) \setminus L \approx_{\tau} P \setminus L + Q \setminus L, (P+Q) / L \approx_{\tau} P / L + Q / L$; (4) $P \approx_{\tau} T.P$.

关于迹语义可参见文[16], 那里还给出了如何计算CSP进程的迹.

2.4 测试语义

测试的意思是使被测试进程与一个测试进程同步并行执行, 并通过观察复合进程的行为来推论被测试进程的行为. 测试有一些可区分的表示成功的动作 $\omega, \omega \in L$. 设 T 是测试进程, $L(T) = L \cup \{\omega\}$, 如果当被测试进程与 T 同步执行时, 并行复合进程能够达到某个状态并执行动作 ω , 则该进程通过 may 测试.

定义2.4 设 P 是一个进程, T 是一个测试进程, $L(T) \subseteq L \cup \{\omega\}$, $P \text{ may } T$ 当且仅当 $(P \mid T) \setminus L \xrightarrow{\omega}$.

若 $\tau = \langle a_1 \dots a_n \rangle$, 令 $\bar{\tau} = \langle \bar{a}_1 \dots \bar{a}_n \rangle$, 因此对于 $T \subseteq L^*$ 可定义 $\bar{T} = \{\bar{\tau} : \tau \in T\}$. $P \text{ may } T$ 意味着存在进程 $(P \mid T) \setminus L$ 的一个有限计算 $S_0 \xrightarrow{\tau} \dots \xrightarrow{\tau} S_n \xrightarrow{\omega}$. 这就是说存在 $\tau \in T(P)$, 使得 $\bar{\tau} \wedge \langle \omega \rangle \in \bar{T}(T)$.

定义2.5 进程 P 和 Q 满足 $P \sqsubseteq_{\text{may}} Q$, 当且仅当 $P \text{ may } T \Rightarrow Q \text{ may } T$ (在此及以后, 本文使用 $\Rightarrow$ 代替常用的逻辑符号 $\Rightarrow$, 因为后者已被用于表示 LTS 转移). 而且称进程 P 和 Q 是 may 测试等价的, 记为 $P \approx_{\text{may}} Q$, 当且仅当 $P \sqsubseteq_{\text{may}} Q$ 且 $Q \sqsubseteq_{\text{may}} P$.

3 输入/输出定义

基于状态机理论的模型都自然地地区分输入与输出语义.

为了能够模拟状态机模型中的输入输出性质, 可以考虑对 SPA 的动作集做出进一步的划分. 因此定义如下的动作集:

(1) $Act_I \subseteq Act$, 称为进程的输入动作集, $Act_O \subseteq Act$, 称为进程的输出动作集. 其中输入动作集和输出动作集满足 $Act_I \cap Act_O = \emptyset, Act_I \cup Act_O = Act - \{\tau\}$.

(2) $Act_{HI} = Act_H \cap Act_I$, 称为进程的高(安全)级输入动作集, $Act_{HO} = Act_H \cap Act_O$, 称为进程的高(安全)级输出动作集, $Act_{LI} = Act_L \cap Act_I$, 称为进程的低(安全)级输入动作集, $Act_{LO} = Act_L \cap Act_O$, 称为进程的低(安全)级输出动作集.

4 无干扰性质的一般概念

所谓信息流就是发生在信源端的变化能够在信宿端引起变化. 所以, 讨论进程的无干扰性质时, 实质是在说进程的某个动作子集上的行为与另一个动作子集上行为具有相关性. 将这一事实一般化, 可以将进程 P 的动作子集之间的无干扰性质定义成 $\mathbb{P}(Act) \times \mathbb{P}(Act)$ 上的谓词 Φ :

$$\Phi: \mathbb{P}(Act) \times \mathbb{P}(Act) \rightarrow \{\text{true}, \text{false}\}$$

对任意两个动作子集 N 和 M , 若 $\Phi(N, M) = \text{true}$, 则称动作子集 N 对动作子集 M 是 Φ -无干扰的. 此时动作子集 M 称为 Φ 的一个观察动作集(窗口), 动作子集 N 称为被保护的行动集合. 观察到的结果是进程在动作集 M 上行为, 称为(观察)视图. 当 $N = Act_H$ 时, Φ 称为全保护性质.

称进程 P 在动作子集 $H \subseteq Act_H$ 上的行为对 P 在 $L \subseteq Act_L$ 上的行为是无干扰的, 若 $\Phi(H, L) = \text{true}$. 记为 $P \models \Phi$. 无干扰性质实际上是说, 根据观察到的结果不能推测进程在动作子集 N 上的行为.

5 基本无干扰模型

Goguen 和 Meseguer 最初给出的无干扰模型是面向确定性系统的. 使用 SPA 语言, 即是面向确定进程的. 所谓一个进程 P 是确定的, 如果进程 P 不能既接受同时又拒绝某个动作 a . 即对任意 $a \in L$ 和 $\tau \in T(P)$, 若存在 Q, R 和 γ 使得 $\tau = \hat{\gamma}$ 且 $P \xrightarrow{\tau} Q \xrightarrow{a} R$, 那么就不能找到 Q' 和 γ' , 使得 $\tau = \hat{\gamma}'$ 且 $P \xrightarrow{\tau} Q'$, 然而却没有从 Q' 发出的转移动作 a . 否则称进程 P 是非确定性的.

5.1 非确定无干扰模型

确定性模型不仅在理论上存在局限, 在实践上也限制了其应用范围. 因此, 随后的研究大都考虑非确定模型. 下面引用的定义5.1是由Focardi^[2,3]针对非确定系统给出的. 根据无干扰性质的Goguen-Meseguer定义, 系统的低级输出在有高

级输入和没有高级输入时是一样的。其进程代数表达式即是(1)式。

定义5.1 设 P 是一个进程,称进程 P 是无干扰安全的,即 P 是非确定无干扰的,记为 $P \models NNI$,当且仅当它满足下面的等式:

$$(P \setminus Act_H) / Act_H \approx_{\tau} P / Act_H \quad (1)$$

定义5.1所定义的无干扰性质保护的子集是高级输入。其基本思想是,高级用户的安全活动是通过进程的高级输入来影响进程的,如果高级的输出动作与高级的输入动作相关,那么保护高级输入动作集,也就保护了相应高级输出动作集。如果高级的输出动作与高级的输入动作无关,那么即使低安全级的用户能够从低级观察视图获得高级输出动作的信息,也还是不能获得任何有关高级用户行为的信息。例如在下面的进程:

$$P \triangleq hi. ho_1. l_1. 0 + hi. ho_2. l_2. 0 + ho_1. l_1. 0 + l_2. 0$$

(其中 $h_i \in Act_H, ho_1, ho_2 \in Act_{HO}, l_1, l_2 \in Act_L, P \models NNI$)中,高级输入动作与高级输出动作无关。虽然高级输出动作与低安全级动作之间存在明显相关,低安全级用户无法推测高级输入动作是否发生。

如上所述,仅保护输入动作集源于这样一种设想,即认为高级输入动作不受高级输出动作影响。这种设想并不总是成立,因为在很多系统中,高级用户的活动是根据高级输出反馈而设计下一步的活动,在这样的系统中,进程的高级输入动作部分地或全部由进程高级输出序列决定,因此,若低安全级用户获得关于高级输出动作的信息,也就获得了有关高级用户行为的信息。如果仅考虑保护高级输入动作集,则进程:

$$P \triangleq l. h'. l'. h. 0 + l. 0, l \in Act_L, l' \in Act_{LO}, h \in Act_H, h' \in Act_{HO}$$

是 NNI 安全的。不妨假设进程 P 是一个含有敏感信息的缓冲区。 P 接受低安全级用户的读请求,向高安全级用户发出反馈使高级用户执行高级输入动作 h 。 P 在向高安全级用户发出反馈的同时,通过 l' 向低级用户报告高级用户的下一步行为。在这种解释之下进程 P 是不安全的,虽然它满足 NNI 性质。不安全的原因是没有保护对高级输入动作有影响的高级输出动作集。因此,在某些应用中,保护高级输出动作集是必须的。当需要保护高级输出动作集时,(1)式就应该写成如下形式:

$$P \setminus Act_H \approx_{\tau} P / Act_H \quad (2)$$

实际(2)式的直接写法应是 $(P \setminus Act_H) / Act_H \approx_{\tau} P / Act_H$ 。但容易证明 $(P \setminus Act_H) / Act_H \approx_{\tau} P \setminus Act_H$ 。在 Focardi 等人的 NDC 模型^[2,3]中,(2)式被称为强无干扰性质。

5.2 广义无干扰模型

在非确定系统中定义无干扰性质的另一种方式是 McCullough 的广义无干扰性质^[9,10]。使用迹模型的语言,McCullough 的定义可翻译成如下直观表示:

$$\begin{aligned} \forall \tau \in T(P) \wedge P \Rightarrow Q &\Rightarrow \forall h \in Act_H, \\ \exists Q' \cdot P &\xrightarrow{r} Q' \wedge Q / Act_{HO} \approx_{\tau} Q' / Act_{HO} \end{aligned} \quad (3)$$

它是由 Roscoe 在 CSP 中给出的^[15]。使用 SPA 语言可以更进一步给出简洁的代数的表示形式。

定义5.2 (广义无干扰, GNI) 设 P 是一个进程, P 满足广义无干扰性质即 $P \models GNI$,当且仅当:

$$(P / Act_H) ||| RUN_{HI} \approx_{\tau} P / Act_{HO} \quad (4)$$

$$RUN_{HI} = \sum_{a \in Act_{HI}} a. RUN_{HI}$$

命题5.1 式(3)与(4)等价。

证明: 在迹语义模型中,(4)式蕴含(3),即对任意 $\tau \in T$

$(P), h \in Act_{HI}$,若令 $P \xrightarrow{r} Q, \delta = \tau \upharpoonright (L - Act_{HO})$,我们证明(4)式蕴含(3)式。

设 $Q \xrightarrow{r'} R$,则 $P \xrightarrow{r'} R$ 。定义 $\eta = \sigma \upharpoonright (L - Act_{HO})$,则 $\delta \wedge \eta \in T(P / Act_{HO})$,并且 $P / Act_{HO} \xrightarrow{\delta \wedge \eta}$ 。根据(4)式,存在 U 使得 $(P / Act_H) ||| RUN_{HI} \xrightarrow{\delta \wedge \eta} (U / Act_H) ||| RUN_{HI}$ 。由算子 $|||$ 的定义,也一定有 $(P / Act_H) ||| RUN_{HI} \xrightarrow{\delta \wedge \eta} (U / Act_H) ||| RUN_{HI}$ 。再次根据(4)式, $P / Act_{HO} \xrightarrow{\delta \wedge \eta} U$ 。这也就是说,存在 $\sigma', \eta = \sigma' \upharpoonright (L - Act_{HO})$,使得 $P \xrightarrow{r'} Q' \xrightarrow{\sigma'} R'$ 。更进一步,即 $Q / Act_{HO} \sqsubseteq_{\tau} Q' / Act_{HO}$ 。

与上同样可以证明 $Q' / Act_{HO} \sqsubseteq_{\tau} Q / Act_{HO}$ 。

至此,我们证明了(4)式蕴含(3)式。现在证明(3)式蕴含(4)式。因为在(4)式中 $P / Act_{HO} \sqsubseteq_{\tau} (P / Act_H) ||| RUN_{HI}$ 是明显的,所有只需证明

$$(P / Act_H) ||| RUN_{HI} \sqsubseteq_{\tau} P / Act_{HO} \quad (4a)$$

设 $\tau \in T((P / Act_H) ||| RUN_{HI})$,我们使用关于长度的归纳法证明 $\tau \in T(P / Act_{HO})$ 。

$\tau = \langle \rangle$ 是显然的。这是长度为0的情况。

设对于长度小于 n 的情况,(4a)成立。当迹长度为 n 的情况,我们两种情形,(1) $\tau = \delta \wedge \langle l \rangle (l \in Act_L)$,和(2) $\tau = \delta \wedge \langle h \rangle (h \in Act_H)$,其中 δ 的长度为 $n-1$ 。根据归纳假设, $\delta \in T(P / Act_{HO})$ 。因此存在 $\eta \in T(P)$,使得 $\delta = \eta \upharpoonright (L - Act_{HO})$ 。根据(3)式, $\eta \wedge \langle h \rangle \in T(P)$,因此 $\tau = \delta \wedge \langle h \rangle \in T(P / Act_{HO})$,即(2)成立。对于情形(1),因为动作转移 $l \in Act_L$ 只能由 P 产生,因此 $\tau \in T(P / Act_{HO})$ 。□

与迹模型公式相比,代数式往往更简洁明快,如代数式(4)。但有时为相应的迹公式寻找代数公式却比较难。

6 不可推断模型

不可推断性质(noninference)是由 O'Helloran^[14]提出的,它的基本思想是分离低级安全动作和高级安全动作。非形式地说,不可推断性要求进程的迹在删除所有高级动作之后,仍是该进程的一个迹。也就是说, $T(P \upharpoonright Act_L) \subseteq T(P)$ 。

定义6.1 (不可推断性, $N-INF$) 进程是 P 不可推断安全的,即 $P \models N-INF$,当且仅当:

$$P \approx_{\tau} P / Act_H + P \quad (5)$$

上一节曾说明,安全性质仅保护高级输入动作子集,对于某些系统是不够的。但如果安全性质要求保护全部高级动作子集,可能又太强了。McLean 指出^[11],不可推断性不仅保护进程的高级输入,而且保护进程的高级输出,对于那些仅有高级输出,而不需要高级输入的进程来说,这一要求过强,可以放松到只保护高级输入就行了。McLean 称这种仅保护高级输入的不可推断性质为广义的不可推断性质($GN-INF$)。广义的不可推断性是一种弱不可推断性,在迹模型中它表示为:

$$\begin{aligned} \forall \tau \in T(P), \exists \delta \in T(P) \cdot \tau \upharpoonright Act_L \\ = \delta \upharpoonright Act_L \wedge \delta \upharpoonright Act_{HI} = \langle \rangle \end{aligned} \quad (6)$$

即低安全级动作子集上的观察视图与高安全级输入子集不存在显式的依赖关系。因为(6)式并不要求 $\delta \in T(P / Act_{HI})$ 也还是 P 的迹,因此它比(5)式以及下面(7)式都要弱。

命题6.1 若进程 $P \models GNI$,则 P 满足(6)式。

证明: 显然若设

$$\tau \in T(P / Act_H) \subseteq T(P / Act_H) ||| RUN_{Act_{HI}}$$

则存在 $\delta \in T(P)$,以及 $\tau' = \delta \upharpoonright (L - Act_{HO}) \in T(P / Act_{HO})$,使得 $\tau' = \tau$ 。由于 τ 中没有高级动作,这意味着 δ 中也没有高级输入动作,也即 $\delta \upharpoonright Act_{HI} = \langle \rangle$ 。注意到 τ 是某个 $\sigma \in T(P)$ 在

低安全级动作集上的投影。因此(4)式蕴含(6)式。 $\square$

定义6.2 (输入不可推断性, IN-INF) 进程 P 是广义不可推断安全的, 即 $P \models \text{IN-INF}$, 当且仅当:

$$P \approx_{\tau} P / \text{Act}_{HI} + P \quad (7)$$

命题6.2 若进程 $P \models \text{GNI} \Rightarrow P \models \text{IN} \Rightarrow P \models \text{GN-INF}$.

7 不可演绎模型

7.1 不可演绎模型

基本的不可演绎模型源自 Sutherland 提出的信息流安全理论^[23]。根据不可演绎模型, 所谓不存在信息流, 意味着从进程行为的低级观察中, 不能演绎地推导出关于进程高级行为的任何性质。更准确地说, 进程 P 具有不可演绎性质, 如果进程行为的任何低级观察 τ 与任何对进程输入的高级观察 τ' 之间不存在矛盾。即 $P \models \text{ND}$ 当且仅当

$$\forall \tau_{low} \in T(P \upharpoonright \text{Act}_L) \cdot \{\tau_h = \tau \upharpoonright \text{Act}_{HI} : \tau \in T(P) \wedge \tau \upharpoonright \text{Act}_L = \tau_{low}\} = T(P \upharpoonright \text{Act}_{HI}) \quad (8)$$

而且可以证明:

命题7.1 $P \models \text{ND}$ 当且仅当, 对于任意进程 $\Pi_1, \Pi_2 \in P_{HI} = \{P \in P : L(P) \subseteq \text{Act}_{HI}\}$, 和任意测试 T :

$$(\Pi_1 \parallel_{\text{Act}_{HI}} P) / \text{Act}_{HO} \text{ may } T \Leftrightarrow (\Pi_2 \parallel_{\text{Act}_{HI}} P) / \text{Act}_{HO} \text{ may } T \quad (9)$$

可把式(9)写成等价的代数表示式, 即下面的(10)式。因此, 我们可以如下地定义不可演绎性质。

定义7.1 $P \models \text{ND}$ 当且仅当, 对于任意进程 $\Pi_1, \Pi_2 \in P_{HI} = \{P \in P : L(P) \subseteq \text{Act}_{HI}\}$

$$\Pi_1 \parallel_{\text{Act}_{HI}} P / \text{Act}_{HO} \approx_{\text{may}} \Pi_2 \parallel_{\text{Act}_{HI}} P / \text{Act}_{HO} \quad (10)$$

不可演绎性有一些不期望的性质。Wittbold 曾给出了一个双密钥加密系统^[24], 它是满足 ND 性质, 但高级用户可以根据高级输出影响后续的高级输出, 并向低安全级用户传递明文。为此, Wittbold 还引入了带策略的不可演绎模型^[24]。关于它的讨论放在第8节。

7.2 带输出保护的不可演绎模型

在 Guttman 之前, 信息流安全模型主要是保护高安全级输入。Guttman 则强调说明, 在某些场合, 保护高安全级输出也是必要的^[7]。至少有两方面的理由需要保护高级输出动作集。一个理由是, 在某些系统中, 高级输出动作依赖低安全级动作, 两者可合作构成隐蔽通道, 如 Wittbold 的双密钥加密系统^[24]。另一个理由是, 当两个系统复合成一个大系统时, 需要连接两个子系统的输入与输出, 前一个子系统可能通过控制高级输出来间接地控制系统的高级输入。考虑下面的进程 WriteUp , 它根据低级输入修改高安全级文件, CodeM 根据写入内容选择密钥对高安全级文件进行加密。

$$\text{WriteUp} \triangleq \text{lread} \cdot \overline{\text{hwirt}} \cdot \text{WriteUp}, \text{CodeM} \triangleq \text{hwrite} \cdot \text{key} \cdot \text{crypt} \cdot \text{CodeM}$$

$\text{lread} \in \text{Act}_L, \text{hwrite}, \text{key}, \text{crypt} \in \text{Act}_{HI}, \overline{\text{hwirt}} \in \text{Act}_{HO}$ 。显然 $\text{WriteUp} \models \text{ND}, \text{CodeM} \models \text{ND}$ 。现在若令 $\text{Trojan} \triangleq (\text{WriteUp} \parallel \text{CodeM}) \setminus \{\text{hwrite}, \overline{\text{hwirt}}\}$, 则 $\text{Trojan} \not\models \text{ND}$ 。

上例的不安全性源于高级输入依赖于低级输入, 因此 Guttman 建议在不可演绎模型中同时保护高安全输出, 消除高级输出对低级输入的依赖关系。使用迹语言表述, Guttman 的系统 P 满足如下性质(保护输出的不可演绎性质 NDO):

$$\forall \tau, \tau' \in T(P) \cdot \exists \delta \in T(P) \cdot \tau \upharpoonright \text{Act}_L = \delta \upharpoonright \text{Act}_L \wedge \tau' \upharpoonright (L - \text{Act}_L) = \delta \upharpoonright (L - \text{Act}_L)$$

而且(8)式一样可证明:

命题7.2 $P \models \text{NDO}$, 当且仅当 P 在相同的低安全级输入上, 任何高级行为, 包括高级输入与高级输出, 都是兼容的,

即对于任意进程 $\Pi_1, \Pi_2 \in P_{HI} = \{P \in P : L(P) \subseteq \text{Act}_{HI}\}$ 和任意测试 T :

$$(P \parallel_{\text{Act}_{HI}} \Pi_1) / \text{Act}_{LO} \text{ may } T \Leftrightarrow (P \parallel_{\text{Act}_{HI}} \Pi_2) / \text{Act}_{LO} \text{ may } T \quad (11)$$

定义7.2 $P \models \text{NDO}$ 当且仅当对于任意进程 $\Pi_1, \Pi_2 \in P_{HI} = \{P \in P : L(P) \subseteq \text{Act}_{HI}\}$

$$(P \parallel_{\text{Act}_{HI}} \Pi_1) / \text{Act}_{LO} \approx_{\text{may}} (P \parallel_{\text{Act}_{HI}} \Pi_2) / \text{Act}_{LO}$$

8 可复合的不可演绎模型

8.1 可复合性质的定义

以上讨论的安全性质都是不可复合的。所谓一种安全性质 Φ 是可复合的, 是指当两个或多个满足性质 Φ 的进程, 通过同步算子构成的复合进程 P 时, 这个复合进程 P 仍满足 Φ 。复合安全性质是一个非常重要的概念, 它是构成复合系统安全理论的基础。可复合安全性是 McCullough 在分析复合系统安全性时引入的一个概念^[9,10]。在 SPA 中, 我们使用更直观的定义(见下)。

定义8.1 称安全性质 Φ 为可复合的, 如果 $P, Q \models \Phi \Rightarrow P \parallel Q \models \Phi$ 。

Foster 构造了一个 ND 不可复合的例子^[6], McCullough 则证明, NNI 和 GNI 都是不可复合的^[10]。不过我们可以证明下面的命题。

命题8.1 若 $P, Q \models \text{NNI}$, 则: (1) $P + Q \models \text{NNI}$; (2) $P \parallel Q \models \text{NNI}$ 。

证明: (1) 由定理2.5直接可证。(2) 的证明需要用到下面的引理8.1。由引理8.1及 NNI 的定义:

$$(P \parallel Q) / \text{Act}_H \approx_{\tau} P / \text{Act}_H \parallel Q / \text{Act}_H \approx_{\tau} (P \setminus \text{Act}_{HI}) / \text{Act}_H \parallel (Q \setminus \text{Act}_{HI}) / \text{Act}_H$$

再根据引理8.1可得证:

$$(P \setminus \text{Act}_{HI}) / \text{Act}_{HI} \parallel (Q \setminus \text{Act}_{HI}) / \text{Act}_{HI} \approx_{\tau} (P \setminus \text{Act}_{HI} \parallel Q \setminus \text{Act}_{HI}) / \text{Act}_{HI}$$

但

$$(P \setminus \text{Act}_{HI} \parallel Q \setminus \text{Act}_{HI}) / \text{Act}_H \sqsubseteq_{\tau} ((P \parallel Q) \setminus \text{Act}_{HI}) / \text{Act}_H$$

所以

$$(P \parallel Q) / \text{Act}_H \sqsubseteq_{\tau} ((P \parallel Q) \setminus \text{Act}_{HI}) / \text{Act}_H$$

$(P \parallel Q) / \text{Act}_H \sqsupseteq_{\tau} ((P \parallel Q) \setminus \text{Act}_{HI}) / \text{Act}_H$ 是显然的。所以 $(P \parallel Q) / \text{Act}_H \approx_{\tau} ((P \parallel Q) \setminus \text{Act}_{HI}) / \text{Act}_H$, 即 $P \parallel Q \models \text{NNI}$ 。 $\square$

引理8.1 (Focardi-Gorrieri^[2]) $(P / \text{Act}_H) \parallel (Q / \text{Act}_H) \approx_{\tau} (P \parallel Q) / \text{Act}_H$

安全性质的可复合性很重要。一方面分解与复合是构造复杂系统的自然方法; 另一方面, 复杂的计算机系统是开放的, 系统的运行时间配置可以随时间变化。因此, 复合安全性质对于定义基于构件的安全系统是非常必要的。

8.2 可复合的不可演绎模型

Wittbold 引入带策略的不可演绎性^[24]是为了避免不可演绎性的一些不期望的性质, 即满足不可演绎性定义的系统可能存在隐蔽通道。带策略的不可演绎性质(NDS)是一类相当强的安全性质, 它是可复合的。 NDS 的迹模型可由下式定义。

$$\forall \tau_{low} \in T(P \upharpoonright \text{Act}_L) \cdot \{\tau_h = \tau \upharpoonright \text{Act}_H : \tau \in T(P) \wedge \tau \upharpoonright \text{Act}_L = \tau_{low}\} = T(P \upharpoonright \text{Act}_H) \quad (12)$$

这个定义保护所有的进程高级动作, 而不是像(8)式那样仅仅保护高级输入。这意味着, 从低级观察中得不到任何高级活动的信息。对于(12)式, 我们同样也有命题7.1那样的命题。

命题8.2 $P \models \text{NDS}$ 当且仅当, 对于任意进程 $\Pi_1, \Pi_2 \in$

$P_H = \{P \in P : L(P) \subseteq Act_H\}$, 和任意测试 T :

$$\Pi_1 \parallel_{Act_H} P \text{ may } T \Leftrightarrow \Pi_2 \parallel_{Act_H} P \text{ may } T.$$

定义 8.2 (带策略的不可演绎性质, NDS) $P \models NDS$ 当且仅当, 对于任意进程 $\Pi_1, \Pi_2 \in P_H = \{P \in P : L(P) \subseteq Act_H\}$

$$P \parallel_{Act_H} \Pi_1 \approx_{may} P \parallel_{Act_H} \Pi_2 \quad (14)$$

定理 8.1 $P \models NDS$, 当且仅当

$$\forall \Pi \in P_H = \{E \in P : L(E) \subseteq Act_H\}, P / Act_H \approx_{\tau} P \parallel_{Act_H} \Pi \quad (15)$$

证明: (14) $\Rightarrow$ (15) 只要取 $U_1 \triangleq RUN_H \equiv \sum_{h \in Act_H} h$. RUN_H 即可。而 (15) $\Rightarrow$ (14) 是显然的。□

从定理 8.1 的证明过程可以看出, 当仅取 $\Pi \triangleq 0$ 时, 定理 8.6 仍然成立。实际上, 我们还可以得到更强的结果。将它表述成定理的形式, 就是定理 8.2。

定理 8.2 $P \models NDS$, 当且仅当

$$P / Act_H \approx_{\tau} P \setminus Act_H \quad (16)$$

证明: 如上所述, 令 $\Pi \triangleq 0$ 可证 (15) $\Rightarrow$ (16)。需证 (16) $\Rightarrow$ (15)。

因为 $\forall \Pi \in P_H = \{E \in P : L(E) \subseteq Act_H\}, (P | \Pi) \setminus Act_H \sqsubseteq_{\tau} (P | RUN_H) \setminus Act_H$ 。考虑证明 $(P | RUN_H) \setminus Act_H \sqsubseteq_{\tau} (P | \Pi) \setminus Act_H$ 。根据 (16) 式, $(P | RUN_H) \setminus Act_H \approx_{\tau} P / Act_H \approx_{\tau} P \setminus Act_H$, 但又由于 $P \setminus Act_H \sqsubseteq_{\tau} (P | \Pi) \setminus Act_H$ 是显然的, 所以 $(P | RUN_H) \setminus Act_H \approx_{\tau} (P | \Pi) \setminus Act_H$ 。因此, 对 $\forall \Pi \in P_H = \{E \in P : L(E) \subseteq Act_H\}, (P | \Pi) \setminus Act_H \approx_{\tau} (P | RUN_H) \setminus Act_H$ 。这很自然地推出 (14) 式。□

Schneider 将 (14) 式定义为测试无干扰性质^[21]。Focardi 等人将 (15) 式称为可复合的不可演绎性质^[2], 简称为 NDC。因为下面的命题 8.3, (16) 式被称为强无干扰性质^[2]。NDS 对于 SPA 选择算子、限制算子以及并行复合算子是保持的。

命题 8.3 (Focardi-Gorrieri^[2]) 若 $P \models NDS$, 则 $P \models NNI$, 并且 $NDS \neq NNI$ 。

证明: $P \models NDS$, $P / Act_H \approx_{\tau} P \setminus Act_H$, 但 $P \setminus Act_H \sqsubseteq_{\tau} (P \setminus Act_H)$, 并且进一步由于 $(P \setminus Act_H) / Act_H \approx_{\tau} (P \setminus Act_H)$, 因此可证得 $P \setminus Act_H \sqsubseteq_{\tau} (P \setminus Act_H) / Act_H$, 也即 $P / Act_H \sqsubseteq_{\tau} (P \setminus Act_H) / Act_H$ 。由于反包含关系是显然的, 因此 (1) 式成立。

以上已证明 NDS 与 NNI 之间的包含关系。因为 $P \triangleq \bar{h}.l.o$ 是 NNI 但非 NDS 的, 所以相等关系不成立, 即 $NDS \neq NNI$ 。□

定理 8.3 若进程 $P, Q \models NDS$, 则 $P+Q, P \setminus L$ 和 $P|Q$ 也是 NDS 的。

证明: 仅证 $P|Q \in NDS$ 。另两种情形可由引理 2.1 和 2.2 直接推证。

对于 $P|Q$ 的情形, 首先我们有 $(P|Q) \setminus Act_H \sqsubseteq_{\tau} (P|Q) / Act_H$ 。由 (16) 式和引理 8.1, $(P|Q) / Act_H \approx_{\tau} (P / Act_H) | (Q / Act_H) \approx_{\tau} (P \setminus Act_H) | (Q \setminus Act_H)$ 。

但 $P \setminus Act_H | Q \setminus Act_H \sqsubseteq_{\tau} (P|Q) \setminus Act_H$, 所以 $(P|Q) / Act_H \sqsubseteq_{\tau} (P|Q) \setminus Act_H$ 。□

9 解析安全模型

限制 (restrictiveness) 安全性质或耦合 (hook-up) 安全性质^[9,10], 是 McCullough 定义的可复合安全性质, 它也是第一个可复合安全性质。此后不久, Johnson 和 Thayer 引入了前向可修正 (forward correctability) 安全性质^[8], 它是一种比限制安全性质弱的可复合性质。与第一节所讨论的性质相比, 前向可修正安全性质和限制安全性质是迹集中的一种复杂解析性质, 即本文所称的 (分段) 解析性质, 它首先将迹分解成片

段, 然后在分片上分析迹的构造性质。

9.1 有限终止进程和迹分解

在研究分段解析性质之前, 还需要做一点概念上的准备: 首先定义什么是有限终止进程, 然后根据 μ -演算^[22] 的一般结论给出一个引理, 这个引理将进程的每个迹都看成是进程分段执行的结果。

定义 9.1 P 称为有限终止进程, 如果它所有的迹都是有限的。进程 P 称是 n -有限终止的, 如果它所有的迹长度都不大于 n 。

根据 μ -演算的一般结论, 对任意 $\tau \in L^*$ 和模态公式 $\langle t \rangle$ true, 都一定存在 $P, P \models \langle t \rangle$ true, 也即 $P \Rightarrow$ 。根据这样事实: 对任意 $\tau \in L^*$ 都存在 $s \in L^*$ 和 $t \in L^*, \tau = s \hat{\ } t$, 可以给出引理 9.2。

引理 9.1 在迹模型中, 对任意 $\tau \in L^*$, 都存进程 $P \triangleq S \triangleright F$, 使得 $\tau \in T(P)$ 。

9.2 可修正性质

基于迹模型的安全性质, 往往都可以归为可修正性质。设 P 是一个进程, 所谓进程 P 的一个干扰, 是指使用某个进程 $\Pi \in P_h = \{P \in P : L(P) \subseteq Act_H\}$ 与 P 复合所得到的进程 $Q \triangleq (P / Act_H) || \Pi$ 。同样, 我们称后者的任意一个迹 $\tau_Q \in T(Q)$ 都是前者某个迹 $\tau_P \in T(P)$ 的一个扰动。称 τ_P 的扰动 τ_Q 是可修正的, 如果存在 $\tau'_P \in T(P)$, 它与 τ_Q 仅在高级输出事件上不同, 即 $\tau'_P / Act_H = \tau_Q / Act_H$ 。这时也称 τ'_P 是 τ_Q 的一个修正。一种安全性质称为可修正安全性质, 如果满足该性质的进程的每个迹扰动都存在一个修正。可修正性质与 GNI 很相似。实际上, 以上讨论的安全性质如 NNI、GNI、GN-INF, 甚至 ND 等, 都是可修正性质。

9.3 前向可修正性质

前向可修正性质是由 Johnson 和 Thayer 在状态机模型中定义的迹分段解析性质^[8]。设 $\tau = \alpha \hat{\ } \langle h \rangle \hat{\ } \delta \hat{\ } \beta$ 是一个迹, 其中 h 是一个高安全级动作 ($h \in Act_H$), $\delta \in Act_L$ 是一个长度不大于 n ($n \geq 0$) 的低安全级输入动作串, $\beta \in Act^*$ 不包含高级输入动作 ($\beta \uparrow Act_H = \langle \rangle$), 在这样的迹 τ 中, 称动作 h 是段 β 在 τ 中的 n -前邻。

若对于进程 P 的每个迹 $\tau \in T(P)$, $\tau = \alpha \hat{\ } \delta \hat{\ } \beta$ 是 τ 的一个分段表示, $\delta \in Act_L, \beta \uparrow Act_H = \langle \rangle$ 。又设 τ' 是通过在 β 的 n -紧前邻位置删除或增加一个高级输入动作而得到 τ 的扰动, $\tau' = \alpha \hat{\ } \langle h \rangle \hat{\ } \delta \hat{\ } \beta'$, 则我们总可以找到 τ' 的一个修正迹 $\tau'' \in T(P)$, $\tau'' = \alpha \hat{\ } \langle h \rangle \hat{\ } \delta \hat{\ } \beta'$, $\beta' \uparrow Act_H = \langle \rangle$ 并且 $\beta \uparrow Act_L = \beta' \uparrow Act_L$, 则称进程 P 是 n -前向可修正安全的。显然前向可修正安全性是迹可修正性质。

作为迹分段解析性质, 前向可修正安全性质也可以这样表述: 若 τ 是形如 $S \triangleright \Omega \triangleright F$ 的进程的一次执行, 则 τ' 是形如 $S \triangleright \Pi \triangleright \Omega \triangleright E$ 的进程的一次执行。所以, n -前向可修正性质可以翻译成进程的如下的性质。

定义 9.2 (n -前向可修正性质, n -FC) 进程 P 是 n -前向可修正安全的, 当且仅当对任意测试进程 T 和 T' :

$$T \equiv S \triangleright \Omega \triangleright F \quad (17)$$

$$T' \equiv S \triangleright \Pi \triangleright \Omega \triangleright (F || RUN_{H0}) \quad (18)$$

进程 P 满足:

$$P \text{ may } T \Leftrightarrow P \text{ may } T' \quad (19)$$

其中 $L(F) \subseteq (Act - Act_H) \cup \{\omega\}$, Ω 是 n -有限终止的简单进程, $L(W) \subseteq Act_L \cup \{\omega\}$, Π 是 1-有限终止进程, $L(\Pi) \subseteq Act_H \cup \{\omega\}$, $\omega \in L$ 。

直观上, 这个定义与 Johnson 和 Thayer 的是等价的。式 (19) 蕴含 Johnson 和 Thayer 的定义是显然的。同样, 满足

Johnson 和 Thayer 的进程,都能通过式(19)的测试。

定义9.3 (ω -前向可修正性质, ω -FC) 若对于任意 $n \geq 0$, 进程 P 都是 n -前向可修正安全的, 则称 P 是 ω -前向可修正安全的。

引理9.1 若 $P \setminus L \text{ may } T$, 当且仅当 $P \text{ may } T \setminus \bar{L}$ 。

证明: $P \setminus L \text{ may } T$, 即 $(P \setminus L | T) \setminus \bar{L} \xrightarrow{\omega}$, 所以存在 $\alpha, L(\alpha) \cap L = \emptyset, (P \setminus L) \xrightarrow{\alpha}, T \xrightarrow{\omega}$, 且 $L(\bar{\alpha}) \cap \bar{L} = \emptyset$ 。故 $P \setminus L \text{ may } T \setminus \bar{L}$, 由 $P \setminus L \sqsubseteq_{\tau} P$, 推出 $P \text{ may } T \setminus \bar{L}, L(\alpha) = \{\alpha: \exists s, t \in Act^* \cdot s \hat{\ } \langle \alpha \rangle \wedge t = \alpha\}$ 。

反之亦成立。 $\square$

引理9.2 $T \triangleq S \triangleright F$, 则 $T \setminus L \approx_{\tau} S \setminus L \triangleright F \setminus L$ 。其中 $L \subseteq L(T)$ 。

证明: 根据 $S \triangleright F$ 的定义, 在 S 执行终止之前, F 不能执行。设 $\sigma \in T(T \setminus L), \sigma \upharpoonright L = \langle \rangle$, 且 $\sigma \in T(T)$ 。若 $\sigma \in T(S)$, 则显然 $\sigma \in T(S \setminus L) \subseteq T(S \setminus L \triangleright F \setminus L)$ 。若 $\sigma = \tau \hat{\ } \delta, \tau \in T(S), \delta \in T(F)$, 则 $\tau \in T(S \setminus L), \delta \in T(F \setminus L)$, 因此也有 $\sigma \in T(S \setminus L) \subseteq T(S \setminus L \triangleright F \setminus L)$ 。

反之, 若设 $\sigma \in T(S \setminus L \triangleright F \setminus L)$, 则必存在 $\sigma = \tau \hat{\ } \delta$ 使得 $\tau \in T(S \setminus L) \subseteq T(S), \delta \in T(F \setminus L) \subseteq T(F)$ (δ 可能为 $\langle \rangle$), 因此 $\sigma \in T(S \triangleright F)$ 。 $\square$

定理9.1 若进程 P, Q 是 n -前向可修正安全的, 则: (1) $P+Q$ 也是 n -前向可修正安全的; (2) $P \setminus L$ 也是 n -前向可修正安全的; (3) $P|Q$ 也是 n -前向可修正安全的。

证明: 已知对于任意可表示为式(17)和(18)的 T, T', P, Q 满足式(19)。

(1) 根据引理2.1和2.2, 易证 $F+Q$ 满足式(19)。

(2) 若 $P \setminus L \text{ may } T$, 则引理9.1, $P \text{ may } T \setminus \bar{L}$, 再由引理9.2, $P \text{ may } T' \setminus \bar{L}$, 继续使用引理9.1, $P \setminus L \text{ may } T'$ 。反之, 若 $P \setminus L \text{ may } T'$, 同样易证 $P \setminus L \text{ may } T$ 。

(3) $P|Q \text{ may } T$, 即 $(P|Q) \xrightarrow{\omega}, T \xrightarrow{\omega}$, 分三种情况

情况1: $P \xrightarrow{\alpha}, T \xrightarrow{\omega}$, 即 $P \text{ may } T$, 则 $P \text{ may } T'$, 因此, $P|Q \text{ may } T'$ 。

情况2: $Q \xrightarrow{\alpha}, T \xrightarrow{\omega}$, 即 $Q \text{ may } T$, 同情况1, $P|Q \text{ may } T'$ 。

情况3: 此时若 $\bar{\alpha} \in T(S)$, 则显然 $(P|Q) \text{ may } T'$ 。因此不妨设 $\bar{\alpha} \in T(S \triangleright \Omega) - T(S)$ (对于 $\bar{\alpha} \in T(T) - T(S \triangleright \Omega)$ 的情形, 仿此可证)。此时有 $P \xrightarrow{\alpha}, Q \xrightarrow{\alpha}, \alpha_1 \in \delta | \gamma$ (并行迹计算可参考文[20]), $\alpha_1 \in \sigma | \tau, \alpha = \alpha_1 \hat{\ } \alpha_2, \alpha_1 \in T(S), \alpha_2 \in T(\Omega)$ 。于是存在测试 S_1 和 $S_2, P \text{ may } S_1$ 或 $Q \text{ may } S_2$, 由定义, 对任意 Π_1 或 $\Pi_2, P \text{ may } S_1 \triangleright \Pi_1$ 或 $Q \text{ may } S_2 \triangleright \Pi_2$, 也即对任意 $\pi_1, \pi_2 \in Act_{HI}$ 以及 π_1 和 π_2 的某个 $\pi \in Act_{HI}$, 由 $P \xrightarrow{\alpha_1} \text{或} Q \xrightarrow{\alpha_2} \text{得} P|Q \xrightarrow{\alpha_1 \hat{\ } \pi \hat{\ } \alpha_2}$ 。因此, 显然 $(P|Q) \text{ may } T'$ 。

若 $(P|Q) \text{ may } T'$, 同样也有三种情况, 情况1和2简单如上。在第三种情况的证明中, 与上稍有不同的是, 在 $P \xrightarrow{\alpha}, Q \xrightarrow{\alpha}$, 必然存在 $\pi \in Act_{HI}$, 要么 $\delta = \delta_0 \hat{\ } \pi$, 要么 $\gamma = \gamma_0 \hat{\ } \pi$, 以及 $\bar{\alpha} = \bar{\alpha}_0 \hat{\ } \bar{\pi}, \alpha_0 \in \delta | \gamma$, 并且必然有 $P \xrightarrow{\alpha_0}, Q \xrightarrow{\alpha_0}$, 以及 $S \xrightarrow{\alpha_0}$ 。因此, 显然有 $P|Q \text{ may } T$ 。这证明了结论(3)。 $\square$

9.4 限制安全性质

限制安全性质是最早提出的可复合安全性质, 它也是迹上的分段解析性质和可修正性质。在这里, 我们仍仿照前向可修正安全性质, 定义限制安全性质如下:

定义9.4 (限制安全性质) 进程 P 满足限制安全性质, 即 $P \models RES$, 当且仅当对任意测试进程 T 和任意测试进程 T' :

$$T \equiv S \triangleright M \triangleright F \quad (20)$$

$$T' \equiv S \triangleright ((M/Act_{HI}) || \Pi) \triangleright (F || RUN_{HO}) \quad (21)$$

其中 $L(M) \subseteq Act_L \cup \{\omega\}, L(F) \subseteq (Act - Act_{HI}) \cup \{\omega\}, \Pi \in P_H, = \{P \in P: L(P) \subseteq Act_{HI}\}$, 且满足:

$$P \text{ may } T \Leftrightarrow P \text{ may } T' \quad (22)$$

比较定义9.4与9.2, 可以看出, 限制安全性质是比前向可修正安全更强的性质。因此易证下面的定理。

定理9.2 若进程 P 是限制安全的, 则 P 是 ω -前向可修正安全的, 即 $P \models RES \Rightarrow P \models \omega\text{-FC}$ 。

定理9.3 若进程 P, Q 是限制安全的, 则: (1) $P+Q$ 也是限制安全的; (2) $P \setminus L$ 也是限制安全的; (3) $P|Q$ 也限制安全的。

证明: 证明过程与定理9.1相同。 $\square$

10 隔离安全模型

隔离安全模型是 McLean^[11] 提出的是一种理想的安全模型, 隔离安全是非常强的信息流性质, 但它的定义形式却非常简单:

定义10.1 进程 P 是隔离安全的, 或称 P 满足理想安全性质, $P \models ISP$, 如果存在进程 $\Pi \in P_H = \{P \in P: L(P) \subseteq Act_H\}$ 和进程 $\Omega \in P_L = \{P \in P: L(\Omega) \subseteq Act_L\}$, 使得 $P \approx_{\tau} \Pi || \Omega$ 。

根据引理2.1和2.2, 很容易证明下面的定理。

定理10.1 若设 P, Q 是隔离安全的, 则 $P+Q, P \setminus L (L \subseteq L), P|Q$ 都是隔离安全。

可以证明, 与以上各节所定义的安全性质相比, 隔离安全性质是最强的信息流安全性质。Zakinthinos 在状态机语境中给出了安全之间的偏序关系^[25]。

结束语 安全模型可以自然地分为访问控制模型和信息流模型^[26]。访问控制模型, 典型地如 BLP 模型^[26], 是行为规则模型, 一直试图定义怎样做才是安全的。信息流模型, 如本文讨论的无干扰模型, 则试图定义安全是什么, 即什么样的信息流是安全的。因此, Ryan 等人认为, 无干扰模型对于定义什么是安全来说更为本质^[20]。本文在统一的进程代数框架内讨论了自 Goguen 和 Meseguer 以来所提出的各类无干扰模型, 并给出了一些新的结论及证明。

参考文献

- Allen P G. A Comparison of non-Interference and non-Deducibility using CSP. In: Proc. of the Fourth IEEE Computer Security Foundations Workshop, Franconia, New Hampshire, June 1991. 43~54
- Focardi R, Gorrieri R. A Classification of Security Properties for process algebras. Journal of Computer Security, 1994/1995, 3(1): 5~33
- Focardi R, Gorrieri R. Classification of security properties (Part I: Information Flow). In: Foundations of Security Analysis and Design - Tutorial Lectures (R. Focardi and R. Gorrieri Eds), LNCS, Vol. 2171, Springer-Verlag, 2001. 331~396
- Foley S N. A universal theory of information flow. In: Proc. of the IEEE Symposium on Research in Security and Privacy. IEEE Press, 1987. 116~121
- Forster R. Non-interference Properties for Nondeterministic Processes. [Ph. D Thesis]. Trinity College, University of Oxford, 1999
- Goguen J, Meseguer J. Inference control and unwinding. In: Proc. of the IEEE Symposium on Research in Security and Privacy, 1984. 75~86
- Guttman J D, Nadal M E. What needs securing? In: Proc. of the Computer Security Foundations Workshop, IEEE Computer Society, June 1988. 34~57
- Johnson D, Thayer F. Security and the composition of machines. In: Proc. of the Computer Security Foundations Workshop, IEEE Press, 1988. 14~23
- McCullough D. Specifications for multi-level security and a hook-

- up property. In: Proc. of the IEEE Symposium on Research in Security and Privacy, 1987
- 10 McCullough D. Noninterference and the composition of security properties. In: Proc. of the IEEE Symposium on Research in Security and Privacy, 1988
 - 11 McLean J. Security models and information flow. In: Proc. of 1990 IEEE Symposium on Research in Security and Privacy, IEEE Press, 1990. 177~186
 - 12 Milner R. Communication and Concurrency. Prentice-Hall, 1989
 - 13 Milner R. Communicating and Mobile Systems: the Pi-Calculus. Cambridge University Press, 1999
 - 14 O'Halloran C. A calculus of information flow. In: Proc. of First European Symposium on Research in Computer Security (SORICS 1990), 1990. 147~159
 - 15 Roscoe A W. CSP and determinism in security modeling. In: Proc. of the 1995 IEEE Symposium on Security and Privacy, IEEE Computer Society, 1995. 114~127
 - 16 Roscoe A W. The Theory and Practice of Concurrency. Prentice-Hall, 1997
 - 17 Roscoe A W, Woodcock J C P, Wulf L. Non-interference through Determinism. In: Proc. of European Symposium on Research in Computer Security 1994 (ESORICS'94), LNCS, Vol. 875, Springer-Verlag 1994. 33~53
 - 18 Ryan P Y A. A CSP formulation of non-interference and unwind-

- ing. Presented at CSFW 1990 and published in Cipher, Winter 1990/1991. 19~30
- 19 Ryan P Y A. Mathematical models of computer security. In: Foundations of Security Analysis and Design - Tutorial Lectures (R. Focardi and R. Gorrieri Eds), LNCS, Vol. 2171, Springer-Verlag, 2001. 1~62
 - 20 Ryan P Y A, Schneider S A. Process algebra and non-interference. Journal of Computer Security, 2001, 9(1,2): 75~103
 - 21 Schneider S A. May Testing, Non-interference and Compositionality: [Technical Report CSD-TR-00-02 2001]. Royal Holloway, University of London
 - 22 Stirling C. Modal and temporal logics for processes. LNCS, Vol. 1043, Springer-Verlag, 1996. 149~237
 - 23 Sutherland D. A model of information. In: Proc. of the ninth National Computer Security Conf. 1986. 175~183
 - 24 Wittbold J T, Johnson D M. Information flow in nondeterministic systems. In: Proc. of the 1990 IEEE Symposium on Research on Security and Privacy, 1990. 144~161
 - 25 Zakinthinos A, Lee S. A general theory of security properties. In: Proc. of the 1997 IEEE Symposium on Research in Security and Privacy, IEEE Computer Society Press, 1997. 94~102
 - 26 周伟,尹青,王清贤. 计算机安全中的经典模型. 计算机科学, 2004, 31(3)

(上接第136页)

图3. a 中 FSRM 曲线同时也存在数个突起(如19~37区间段),对应图3. b 中,我们可以知道这些正是系统对文件写操作比较频繁的时段。从图3. c 中可以看到 FSRM 自动减少了副本数量,避免大量的文件同步更新操作占用网络资源,且图3. b 中的系统写操作高峰期和图3c 中副本数量低谷吻合得比较好,证明了 FSRM 正确地反映了系统对文件读写模式的变化。仔细观察,在图3. a 中 FSRM 的每个突起的开始处,都有一个尖峰出现(如时间点19处),这是由于系统写操作的突然增加,FSRM 还来不及反应所造成的。但是我们也看到,FSRM 及时地调整了副本数量,使得网络中的操作数迅速减少,从而形成了这样的一个尖峰。这恰恰表明了 FSRM 对系统行为反映的灵敏度。

观察图3. a 中 non-FSRM 的操作数曲线,同样可以发现存在数个突起,但是这些突起没有迅速下降到一个合理的水平,而是保持较高的高度。对应 c 中我们可以知道这是由于 non-FSRM 的副本数量没有减少造成的。分析其原因,主要是因为各个副本的周期长度都相同,基本上都同时到期。于是各副本都发现系统写操作的增加,几乎同时提出删除自身的请求。由于同时提出请求的副本过多,反而不能收集到足够的同意应答,导致删除自身副本的操作失败,副本数维持在一个较高的水平。而反观 FSRM,由于各副本的周期各不相同,减少了同时提出删除请求的可能,因此使得副本数量变化更加灵敏。

再来看副本数量的平均值:FSRM 在实验时段内平均保有4.26个副本,而 non-FSRM 有6.22个,明显高于 FSRM。特别地,FSRM 的平均副本数4.26大于静态管理策略的3个,网络操作数反而远小于静态,也表明了 FSRM 能十分有效地提高系统性能,减少网络流量。

通过记录节点5上文件 *f* 副本的扫描周期,可以知道,在实验的时段内,节点5扫描文件 *f* 的本地副本共57次,最长周期达到了2233秒,最短周期为20秒,平均周期长度为176秒。而在 non-FSRM 中,节点5共扫描了500次,远远高于 FSRM 的次数。这说明了 FSRM 能以更小的系统代价获得更好的系统性能。

若将 non-FSRM 的扫描周期定为40秒,可以预测,由于扫描周期的放大,non-FSRM 更加不能迅速地反映系统操作

行为的变化,性能将比20秒时更低。表2是经过与上节相同长度的时间段试验之后的结果,从表中我们可以看出 FSRM 保持了良好的性能,而 non-FSRM 性能下降,几乎与静态管理策略一样了。

表2

	FSRM	Non-FSRM	Static
操作总数	29061	41687	41273

结论 本文在分析现有的分布式文件系统动态副本管理机制的基础上,提出了一种新颖的基于访问频率的分布式文件系统自适应动态副本管理机制——FSRM。在 FSRM 中,节点周期性地扫描本地副本的读写情况,根据系统对文件读写模式的变化,自主决定本地副本的增删或迁移,以调整文件副本数量以及副本存放位置,提供更高的系统性能,减少网络流量。FSRM 的副本扫描周期长度不固定,能根据本地副本被访问的频率自动增减,以更少的系统开销提供灵敏的系统适应性,并可以对系统中各个副本实现不同精度的差异化管理,合理分布系统荷载。

通过实验,我们证明了 FSRM 能在不增加系统负载的条件下,更加灵活高效地实现对文件副本的分布式动态管理,提高系统性能,减少网络开销,是一种有效的动态副本管理机制。

参考文献

- 1 Dowdy D, Foster D. Comparative models of The File Assignment Problem. Computing Surveys, 1982, 14(2)
- 2 Bartal Y, Fiat A, Rabani Y. Competitive algorithms for distributed data management. In: Proc. 24th ACM Symp. on Theory of Computing, 1992. 39~50
- 3 Cabri G, Corradi A, Zambonelli F. Experience of Adaptive Replication in Distributed File Systems. In: Proc. of EUROMICRO-22, 1996. 459~466
- 4 Acharya S, Zdonik S B. An Efficient Scheme for Dynamic Data Replication. Brown University CS-93-43, 1993
- 5 Wolfson O, Jajodia S, Huang Y. An adaptive data replication algorithm. ACM Transactions on Database Systems, 1997, 22: 255~314
- 6 Ranganathan K, Iamnitchi A, Foster I. Improving Data Availability through Dynamic Model-Driven Replication in Large Peer-to-Peer Communities. In: Proc. of the Workshop on Global and Peer-to-Peer Computing on Large Scale Distributed Systems, Berlin, May 2002