可信计算平台信任链安全性分析

徐明迪 张焕国 赵 恒 李峻林 严 飞

(武汉数字工程研究所 武汉 430074)

(武汉大学计算机学院 武汉 430072)

(空天信息安全与可信计算教育部重点实验室 武汉 430072)

摘 要 可信计算规范是指导可信计算产品研制的依据,可信计算规范本身的安全性需要得到验证.信任链是可信计算平台中保障系统安全可靠的主要技术手段,它是可信计算平台整个系统安全的中心问题.针对可信计算平台信任链规范的信息流安全问题,文中通过安全进程代数对信任链系统接口进行形式化建模,用可复合的不可演绎模型刻画信任链实体间的交互关系,把规范定义的信任链行为特性抽象为多级安全输入输出集,在讨论高级和低级输入输出依赖关系的基础之上,对信任链复合系统进行信息流分析,并给出结论和证明.

关键词 信任链;安全进程代数;可复合的不可演绎模型

中图法分类号 TP309 **DOI 号**: 10.3724/SP.J.1016.2010.01165

Security Analysis on Trust Chain of Trusted Computing Platform

XU Ming-Di ZHANG Huan-Guo ZHAO Heng LI Jun-Lin YAN Fei

(*Wuhan Digital and Engineering Institute*, *Wuhan* 430074)

(*School of Computer*, *Wuhan University*, *Wuhan* 430072)

(*Key Laboratory of Aerospace Information Security and Trust Computing*, *Ministry of Education*, *Wuhan* 430072)

Abstract The specifications of trusted computing are guidance for products. But securities of specifications themselves need to be verified. The chain of trust is the key technical method to assure system security and is the focus of security in trusted computing platform. Aiming at information flow issue on specification of chain of trust, this paper uses secure process algebra to model trust chain, and describes mutual relationships between entities by non deducibility on composition, and abstracts the behavior and characters of specification of trust chain to multi-level secure inputs and outputs. After discussing associated relations of I/O of high level and low level, we analyzed the system of trust chain. Finally we put forward some conclusions and sounds prove.

Keywords chain of trust; secure process algebra; non deducibility on composition

1 引 言

目前,可信计算的相关研究已成为当前国内外信息安全方面的研究热点和趋势之一.一方面,可信计算在计算机系统上有着具体的实现和应用,如Vista 操作系统的 BitLocker 通过信任链机制保护加密分区和卷主密钥,为了实现对虚拟机的动态度量,AMD 和 Intel 生产的 CPU 增加了支持动态可信度量根的指令.另一方面,可信计算产品的安全测评也是业界所关注的问题,可信计算组织 TCG (Trusted Computing Group)针对可信平台模块

收稿日期:2009-10-08;最终修改稿收到日期:2010-04-23.本课题得到国家"八六三"高技术研究发展计划项目基金(2007AA01Z411)、国家自然科学基金(60673071)资助.**徐明迪**,男,1980 年生,博士,工程师,主要研究方向为可信计算与系统安全.E-mail: mingdixu@163.com.**张焕国**,男,1945 年生,教授,博士生导师,主要研究领域为信息安全与可信计算.**赵 恒**,女,1966 年生,博士,高级工程师,主要研究方向为软件工程技术与应用.**李峻林**,男,1966 年生,研究员,主要研究领域为软件工程技术与应用.**严 飞**,男,1980 年生,讲师,主要研究方向为可信计算.

TPM(Trusted Platform Module)的设计安全,给出了相应的保护轮廓,并通过了国际通用标准认证 CC(Common Criteria). Atmel 公司的 TPM 产品 AT97SC3201 通过了认证实验室 CygnaCom 的 EAL3 认证. Infineon 公司已经开始对生产的 TPM 进行最严格的硬件安全评估流程审核,计划要达到 EAL4 硬件安全水平. 从其保护轮廓的文档描述来看,其主要讨论的是 TBB 与平台的唯一关联性问题,对于其它安全性目标并未涉及[1].

可信计算产品的设计依据源于规范说明,因此可信计算产品的安全程度和规范说明有着直接联系. 一般来说,规范通常是采用自然语言或者非形式化语言的一种描述,很难直接从规范中发现其潜在的安全缺陷或漏洞. 目前,TCG 尚未对其规范和产品的安全性进行全面的分析和验证. 从公开的文献来看,TCG 技术规范中仅对直接匿名认证 DAA (Direct Anonymous Attestation)等协议进行了较为严格的安全性分析.

信任链的交互模型建立过程是将信任链由可信计算规范进行粗粒度抽取的过程. 在这方面,国内外已经有了少量工作,Abadi 和 Wobber 在文献[2]中使用授权逻辑对下一代安全计算基 NGSCB 的基本框架和 API 函数进行了形式化描述,Chen 等人用谓词逻辑分析了可信启动过程存在的信任链传递所带来的信任损失问题,并提出一种集中式度量的改进启动模型[3],但该文所提出的模型无法应用于信任链的测试研究工作. Gürgens 等人使用模型检查器对 TPM 的若干协议进行了分析,发现了授权协议和远程证明协议中存在的问题[4]. Millen 等人开发了一个符号模型检查器来描述可信启动过程中 TPM 内部组件、BIOS 和平台其它组件之间的信任关系[5],但该文主要围绕 PCR 的度量问题对信任启动过程进行建模,并未讨论信任链的具体交互问题. Lin 使用 Otter 和 Alloy 分析 TPM 中 API 调用序列的安全性[6]. 文献[7-8]通过安全进程代数 SPA (Security Process Algebra)和时序逻辑对信任链中的静态度量根 SRTM(Static Root of Trusted Measurement)和动态度量根 DRTM(Dynamic Root of Trusted Measurement)进行了形式化建模和证明,但是该文对于 SRTM 的一些前提假设,如平台配置寄存器 PCR(Platform Configuration Register)的写操作问题、内存写保护问题,在现有系统中过于理想.

同时,信任链由系统中的多个组件构成,它建立于原有计算机硬件系统基础之上,但不是以整体形式存在于系统中,而是各组件与原有系统存在时间、空间上的交错状态[9],这种空间上的离散型,时间上的并行性使得信任链的建模变得困难. 另一方面,信任链和系统之间通过相互作用形成一个复杂的系统整体结构,TPM、RTM 和 System 都有各自的输入、输出和安全策略,从信息流的角度来看,复合系统中不应出现直接或间接的信息泄露.

安全性质的可复合性很重要,一方面分解与结合是构造复杂系统的有效方法;另一方面,复杂的计算机系统是开放的,系统配置可以随时间变化,因此,复合安全性质对于定义安全系统是非常必要的[10].

本文从可复合的不可演绎模型出发,对可信计算信任链规范进行形式化建模,描述信任链传递过程中的实体交互关系,通过 SPA 描述信任链系统的信息流性质,刻画信任链实体动作,抽象出高低安全级进程的输入和输出,最后用 CoPS 工具对安全属性进行验证,找出规范中存在的安全缺陷.

2 研究背景

2.1 基于语言的安全模型

现有的基于语言的安全模型大致上都是扩展无干扰安全的思想,在不同层次上刻划不同安全等级的访问主体之间的无干扰关系[11],如 Focardi 等人扩展了 Milner 的 CCS 建立了安全进程代数[12],将系统中的名赋予“高”、“低”两种安全级别,对无干扰模型进行重新定义及分类,分析不同安全属性的强度. Ryan 和 Schneider 使用 CSP 得出多个无干扰安全属性的定义,并提出一个建立统一无干扰定义属性的构想. Volpano 等人将系统的可靠性视为某种无干扰属性,通过证明标准程序语义建立系统的可靠性,使得所有类型定义良好的程序都具有无干扰安全属性[13].

2.2 安全进程代数

2.2.1 SPA 的基本语法

安全性质是信息流性质[10]. 直接或间接的信息泄露都被看作系统中的信息流动,都可以使用信息流分析的技术找到系统中潜在的各种安全问题,对于信息流安全而言,安全属性刻画了一个安全的多级系统应该满足的属性,信息流分析的技术源于基本无干扰(NI)模型[14],后来又相继出现了不可推断(N-INF)模型[15]、输入不可演绎(NDI)模型[16-17]. 但这些安全属性都是不可复合的. 后来又有学者提出

了策略不可演绎(NDS)模型和可复合的不可演绎(NDC)模型[12].

进程代数作为描述互作用系统的一般框架，适合于研究安全信息流性质. 在进程代数框架内研究信息流性质，不仅可以对其进行概括和比较，而且还可以利用进程代数中的某些结果更抽象更深刻地理解安全性及其结合性质[10]. 本文使用的安全进程代数语言 SPA，是 CCS 的简单扩展. SPA 的基本语法符号包括：

(1) 进程动作集 Act. 它由 3 部分组成：无穷名集 I，I 的伴名集 O，一个称为哑名的元素 τ. 其中，$I=\{a,b,\cdots\}$ 表示输入动作集；$O=\{\bar{a},\bar{b},\cdots\}$ 表示输出动作集；$\mathcal{L}=I\cup O$ 表示可见动作集合，τ 表示不可见动作. 更进一步，为了描述多级安全系统中不同安全级之间的信息流关系，SPA 将可见动作集 $\mathcal{L}$ 划分为两个安全级：高安全级动作 Act_H 和低安全级动作 Act_L，并且 $Act_H=Act_{HI}\cup Act_{HO}$，$Act_L=Act_{LI}\cup Act_{LO}$，$Act_H\cup Act_L=\mathcal{L}$，$Act_H\cap Act_L=\varnothing$，$Act=Act_H\cup Act_L\cup\tau$，通常用 μ 表示进程动作集.

(2) 进程变量集 χ，并以 X,Y,Z 等表示进程变量；进程常量集 $\mathcal{K}$，并以 A,B 等表示进程变量.

(3) 算子符号 $\cdot$，$+$，$|$，$\|$，$\backslash$，$/$，分别表示前缀算子、选择算子、并行算子、复合算子、限制算子和隐藏算子.

定义 1.　无穷名集 I 与伴名集 O 之间的二元关系 $\bar{\cdot}:\mathcal{L}\to\mathcal{L}$ 为 $\forall a\in I\Rightarrow\bar{a}\in O$，$\forall\bar{a}\in O\Rightarrow\bar{\bar{a}}=a\in I$.

根据定义 1，对于集合 L，$L\subseteq\mathcal{L}$ 的伴名集 $\bar{L}$，有 $\bar{L}=\{\bar{\mu}:\mu\in L\}$，为了表示方便，以后我们用 f 表示二元关系 $\bar{\cdot}$，对于 $\forall\mu\in Act$ 有 $f(\bar{\mu})=\overline{f(\mu)}$，如果 $\mu=\tau$，那么 $f(\tau)=\tau$.

进程项集合 $\mathcal{E}$ 是如下语法构造的集合：$E::=0|\mu.E|E+E|E|E|E\backslash L|E\backslash_I L|E/L|E[f]|\chi$，其中 $L\subseteq\mathcal{L}$. 符号 0 用于表示不做任何动作的空进程；$\mu.E$ 表示做完动作 μ 后的系统行为 E；E_1+E_2 表示选择 E_1 或者 E_2；$E_1|E_2$ 表示系统 E_1 和 E_2 之间进行交织并发，并且对互补的输入/输出动作进行同步，将其作为一个内部动作 τ；$E\backslash L$ 执行属于 E 但不属于 $L\cup\bar{L}$ 集合的动作；$E\backslash_I L$ 执行属于 E 但不属于 $L\cap I$ 集合的动作；E/L 将 E 中所有属于 L 的动作转换为内部动作 τ；如果 E 可以执行动作 μ，那么 $E[f]$ 执行 $f(\mu)$；常量 χ 的定义为 $\chi\overset{\text{def}}{=}E$，也就是 χ 具有 E 的所有功能[12].

SPA 除了对 CCS 中的动作集 L 进行高低安全等级划分以外，还引入了两种算子：限制算子和隐藏算子. 限制算子 $E\backslash L$ 执行属于集合 $\{E-L\cup\bar{L}\}$ 的动作，带输入限制的算子 $E\backslash_I L$ 执行属于集合 $\{E-L\cap I\}$ 的动作；隐藏算子 E/L 把在集合 $\{E\cup L\}$ 中的动作转换为不可见动作 τ.

用 $\mathcal{L}(E)$ 表示进程 E 所有的动作执行迹，那么高安全级进程和低安全级进程分别被定义为 $\mathcal{E}_H\overset{\text{def}}{=}\{E\in\mathcal{E}|\mathcal{L}(E)\subseteq Act_H\cup\{\tau\}\}$ 和 $\mathcal{E}_L\overset{\text{def}}{=}\{E\in\mathcal{E}|\mathcal{L}(E)\subseteq Act_L\cup\{\tau\}\}$.

2.2.2　SPA 的操作语义

SPA 的操作语义模型是标记变迁系统模型 $(\mathcal{E},Act,\to)$，其中二元关系 $\to\subseteq\mathcal{E}\times Act\times\mathcal{E}$ 的结构化操作语义[18]，如图 1 所示.

前缀	$\dfrac{-}{\mu.E\xrightarrow{\mu}E}$		
选择	$\dfrac{E_1\xrightarrow{\mu}E_1'}{E_1+E_2\xrightarrow{\mu}E_1'}$	$\dfrac{E_2\xrightarrow{\mu}E_2'}{E_1+E_2\xrightarrow{\mu}E_2'}$	
并行	$\dfrac{E_1\xrightarrow{\mu}E_1'}{E_1\|E_2\xrightarrow{\mu}E_1'\|E_2}$	$\dfrac{E_2\xrightarrow{\mu}E_2'}{E_1\|E_2\xrightarrow{\mu}E_1\|E_2'}$	$\dfrac{E_1\xrightarrow{\alpha}E_1',E_2\xrightarrow{\bar{\alpha}}E_2'}{E_1\|E_2\xrightarrow{\tau}E_1'\|E_2'}$
限制	$\dfrac{E\xrightarrow{\mu}E'}{E\backslash L\xrightarrow{\mu}E'\backslash L},\ \mu\notin L\cup\bar{L}$		
输入限制	$\dfrac{E\xrightarrow{\mu}E'}{E\backslash_I L\xrightarrow{\mu}E'\backslash_I L},\ \mu\notin L\cap I$		
隐藏	$\dfrac{E\xrightarrow{\mu}E'}{E/L\xrightarrow{\mu}E'/L},\ \mu\notin L$	$\dfrac{E\xrightarrow{\mu}E'}{E/L\xrightarrow{\tau}E'/L},\ \mu\in L$	
转换	$\dfrac{E\xrightarrow{\mu}E'}{E[f]\xrightarrow{f(\mu)}E'[f]}$		
常量	$\dfrac{E\xrightarrow{\mu}E'}{A\xrightarrow{\mu}E'},\ A\overset{\text{def}}{=}E$		

图 1　SPA 的结构化操作语义

除了图 1 中对常用操作算子的描述以外，文献[12]还提出了使用并行算子和限制算子来实现复合算子的功能. 复合算子要求两个系统 E 和 F 通过并行方式构成一个新的复合系统，并同步 E,F 之间的互补动作.

定义 2. 系统 E 和 F 的复合系统为

$$E \parallel F \stackrel{\text{def}}{=} (E|F)\backslash(\mathcal{L}(E)\cap\mathcal{L}(F)).$$

2.2.3 SPA 的迹语义和互模拟语义

迹是 LTS 理论中的重要概念，它是系统中可观察动作所组成的一个有限序列.

定义 3. $\forall E\in\mathcal{E}, E$ 的迹 $T(E)=\{\mu\in\mathcal{L}^* \mid E\stackrel{\mu}{\Rightarrow}E'\}$. 其中 $\mu=\alpha_1\cdots\alpha_n\in\mathcal{L}^*$. $E\stackrel{\mu}{\Rightarrow}E'$ 当且仅当 $\exists E_1,E_2,\cdots,E_n\in\mathcal{E}\Rightarrow E\stackrel{\alpha_1}{\Rightarrow}E_1\stackrel{\alpha_2}{\Rightarrow}\cdots\stackrel{\alpha_n}{\Rightarrow}E_n$，其中 $E_n=E'$. $E\stackrel{\alpha}{\Rightarrow}E'$ 表示 $E(\stackrel{\tau}{\rightarrow})^* E_1\stackrel{\alpha}{\rightarrow}E_2(\stackrel{\tau}{\rightarrow})^* E'$. $(\stackrel{\tau}{\rightarrow})^*$ 表示内部动作序列.

当 $E\stackrel{\mu}{\Rightarrow}E'$ 成立，我们认为系统从状态 E 到达状态 E'，从观察的角度来看，系统从状态 E 到状态 E' 所执行的动作轨迹是 $\alpha_1,\alpha_2,\cdots,\alpha_n$.

定义 4. $\forall E,F\in\mathcal{E}, E$ 和 F 迹等价当且仅当 $T(E)=T(F)$，记为 $E\approx_T F$.

为了能比较迁移系统的中间状态，文献[19]提出了互模拟等价的概念. 互模拟刻画了两个系统之间的单步模拟思想，也就是说，当进程 E 执行一个动作到达 E' 后，进程 F 也必须能够通过执行同样的动作到达 F'，进而模拟 E 所完成的单步过程，进程 E' 和 F' 继续重复进行接下来的单步过程.

定义 5. 二元关系 $R\subseteq\mathcal{E}\times\mathcal{E}$ 是弱互模拟，如果 $(E,F)\in R, \forall\mu\in Act\Rightarrow$

(1) 无论何时 $E\xrightarrow{\mu}E'$，都存在 $F'\in\mathcal{E}$ 使得 $F\stackrel{\hat{\mu}}{\Rightarrow}F'$ 和 $(E',F')\in R$ 成立.

(2) 无论何时 $F\xrightarrow{\mu}F'$，都存在 $E'\in\mathcal{E}$ 使得 $E\stackrel{\hat{\mu}}{\Rightarrow}E'$ 和 $(E',F')\in R$ 成立.

两个进程项 $E,F\in\mathcal{E}$ 的观察等价记为 $E\approx_B F$，如果存在着一个包含 (E,F) 的弱互模拟二元关系 R.

3 信任链规范安全性分析

3.1 可复合的不可演绎安全性质

复合安全性质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[20]. 可复合性是安全性质的重要特性，所谓一种安全性质 Φ 是可复合的，是指两个或多个满足性质 Φ 的进程，通过复合算子构成的复合进程 P 时，这个复合进程 P 仍满足 Φ. 在对不可演绎模型进行形式化描述之前，我们先给出相关的符号说明：

T：系统的执行序列.

V：系统执行序列 t，$t\in T$ 后的值.

$f: T\rightarrow V$：从执行序列 T 到值 V 的信息函数，表示系统当前视图，或者是一组值或变量.

Image(f)：所有执行序列的视图集合，即 $\forall t\in T, f(t)\in\text{Image}(f)$.

定义 6. 令 $\gamma,\gamma'\in(\mathcal{L})^*$ 是两个迹，γ' 是 γ 的一个子序列(用 $\gamma'\prec\gamma$ 表示)当且仅当 γ' 和 γ 满足关系：$\gamma=\alpha_1\cdots\alpha_n$，$\gamma'=\alpha_{k_{\gamma',\gamma}(1)}\cdots\alpha_{k_{\gamma',\gamma}(m)}$，其中 $m\leq n$，$k_{\gamma',\gamma}:[1,m]\rightarrow[1,n]$ 是一个单调递增函数，$\alpha_i\in\gamma$ 表示 α_i 出现在序列 γ 中.

定义 7. 视图函数 $low, highinput, lowinput, input$ 分别被定义为

(1) $low: \mathcal{L}^*\rightarrow Act_L^*$.

那么有 $\gamma'=low(\gamma)$，$\gamma'\prec\gamma$，如果 $\forall\alpha_i\in\gamma, \alpha_i\in Act_L\Rightarrow\exists j: i=k_{\gamma',\gamma}(j)$.

(2) $highinput: \mathcal{L}^*\rightarrow(Act_H\cap I)^*$.

那么有 $\gamma'=highinput(\gamma)$，$\gamma'\prec\gamma$，如果 $\forall\alpha_i\in\gamma$，$\alpha_i\in Act_H\cap I\Rightarrow\exists j: i=k_{\gamma',\gamma}(j)$.

(3) $lowinput: \mathcal{L}^*\rightarrow(Act_L\cap I)^*$.

那么有 $\gamma'=lowinput(\gamma)$，$\gamma'\prec\gamma$，如果 $\forall\alpha_i\in\gamma$，$\alpha_i\in Act_L\cap I\Rightarrow\exists j: i=k_{\gamma',\gamma}(j)$.

(4) $input: \mathcal{L}^*\rightarrow I^*$.

那么有 $\gamma'=input(\gamma)$，$\gamma'\prec\gamma$，如果 $\forall\alpha_i\in\gamma, \alpha_i\in I\Rightarrow\exists j: i=k_{\gamma',\gamma}(j)$.

low 得到动作序列中的所有低安全级动作，$highinput$ 得到动作序列中的高安全级输入动作，$lowinput$ 得到动作序列中的低安全级输入动作，$input$ 得到动作序列中的所有输入动作.

定义 8. 令 f_1 和 f_2 是两个信息函数且 $\exists w\in\text{Image}(f_2)$. 对于视图 w 而言，信息从 f_1 流向 f_2 当且仅当 $\exists v\in\text{Image}(f_1)$，$\forall t\in T$，$f_1(t)=v\Rightarrow f_2(t)\neq w$.

定义 8 假定了 f_1 和 f_2 分别对应着高安全级进程和低安全级进程，如果存在从 f_1 和 f_2 的信息流，那么对于系统中的任意迹 t 来说，即使曾有 $w\in\text{Image}(f_2)$，但由于低安全级进程 f_2 的当前视图已经被高安全级进程 f_1 所影响，因此导致了 $f_2(t)\neq w$. 定义 8 的另外一层含义是：如果信息从 f_1 流向了

f_2，说明了 f_1的输入影响了 f_2的输出，那么就认为 f_2能够演绎出 f_1的输入.

那么在什么情况下 f_2不能演绎出 f_1的输入呢？我们给出如下定义.

定义 9.　给定 T，f_1和 f_2，信息没有从 f_1流向 f_2当且仅当联合函数(f_1，f_2)与视图积 Image(f_1)×Image(f_2)之间满足映成关系(onto function).

这里的映成关系指(f_1，f_2)与 Image(f_1)×Image(f_2)满足一一映射.换句话说，如果对于每一个 $\gamma\in Act_L^*$，都存在着迹 $t\in T(E)$使得 $\gamma=low(t)$，并且对于每一个 $\gamma'\in(Act_H\cap I)^*$，也都存在着迹 $t'\in T(E)$使得 $\gamma'=highinput(t')$，那么一定存在着一条迹 t''，其低安全级视图是 γ，高安全级视图是 γ'.

输入不可演绎模型源自 Sutherland 提出的信息流安全理论[16].根据输入不可演绎模型，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，或者说进程 E 具有不可演绎性质，如果进程的任何低级可观察行为 low 都不能推导出高级输入 $highinput$ 的任何信息.

前面我们对输入不可演绎模型进行了定义，下面我们用 SPA 给出可复合的不可演绎性质的定义.

定义 10.　安全性质 Φ 是可复合的，如果 P，$Q\models\Phi\Rightarrow P\parallel Q\models\Phi$.

定义 11.　令 $H\subseteq Act_H$，$E\in\mathcal{E}$，那么 E 具有可复合的不可演绎性质当且仅当

$\forall\Pi\in\mathcal{E}_H\Rightarrow lowviews(E)=lowviews((E|\Pi)\backslash H)$，这里的 $lowviews(\ )$ 函数是 $low(\ )$ 函数的幂集：$lowviews(E)\overset{\text{def}}{=}\{\gamma\in Act_L^*\mid\exists\gamma'\in T(E)\}$.

3.2　信任链规范说明

我们在文献[9]中提出了可信计算 PC 规范说明模型，将可信计算平台抽象为 TPM、RTM 和 System 3 个实体间的交互模型，该模型描述了信任链建立过程中的实体交互关系.

从安全进程代数的角度考虑，为了能够分析信任链建立过程的信息流安全，我们需要进一步对实体的交互关系进行细化，定义出实体间的输入和输出，从安全的角度对这些输入输出进行等级划分，验证信任链系统所符合的安全属性.

可信计算 PC 规范给出了静态可信度量根(SRTM)的实现流程和方法，图 2 刻画了 SRTM 运行期间 3 个进程间的输入和输出.其中 System 包括了 OptionROMs、IPL 和 GRUB. System 通过 RTM 所提供的 BIOS 服务接口对 TPM 进行访问请求；RTM 包含了 3 种访问 TPM 的方式：应用层驱动、TPM-MA 驱动和 TPM-MP 驱动，RTM 会自动把来自 System 的应用层驱动请求转换为 TPM-MP 驱动方式发送给 TPM，TPM-MA 驱动通过 BBB(BIOS Boot Block)通道与 TPM 进行通信，TPM-MP 驱动通过 SMM(System Management Mode)通道与 TPM 进行通信.图 2 给出了这些输入输出接口的函数，可以看出，TPM-MA 驱动不对外提供任何服务，仅完成 CRTM 阶段的完整性度量，而 TPM-MP 驱动则更多地处理来自内部或外部的访问请求.下面我们用 SPA 语法对这 3 个进程实体做进一步说明.

首先，为了精确刻画文献[9]中对 TPM、RTM 和 System 3 个实体间的交互关系，我们需要细化信任链规范说明中 3 个实体交互时的输入输出.

定义 12.　令 SRTM 表示信任链规范说明，则 SRTM$\triangleq$(System|RTM|TPM)$\backslash Sy$，其中限制集合 Sy 用于实体间的动作同步.

SRTM 主要包含了迭代($extend$)PCR、度量($read$)内存代码、建立(log)度量日志和执行($call$)被度量代码等操作，这里，我们把 $read$、$extend$、$call$ 和 log 动作分别定义为读(r)、写(w)、执行(e)和追加(a)操作，并将图 2 中的接口函数进行归纳，那么可以得到如下描述：

RTM$\triangleq$CRTM|POSTBIOS

CRTM$\triangleq r_POSTBIOS.\overline{ma_HashAllExtendTPM}.\overline{e_POSTBIOS}.$CRTM+
$a_SYSTEM_ACPI.\overline{smiUpdateLogEvent}.$CRTM+$a_RTM_ACPI.\overline{smiUpdateLogEvent}.$CRTM

POSTBIOS$\triangleq e_POSTBIOS.r_OptionRoms.\overline{mp_TPMTransmit}.\overline{a_RTM_ACPI}.\overline{e_OptionRoms}.$
$\overline{r\_IPL}.\overline{mp\_TPMTransmit}.\overline{a\_RTM\_ACPI}.\overline{e\_IPL}.\underline{0}$

System$\triangleq e_OptionRoms.e_IPL.\overline{r_GRUB}.\overline{tcg_PassThroughToTPM}.\overline{a_System_ACPI}.\overline{e_GRUB}.\underline{0}$

TPM$\triangleq tcg_PassThroughToTPM.w_PCR.$TPM+$ma_HashAllExtendTPM.w_PCR.$TPM+
$mp_TPMTransmit.w_PCR.$TPM

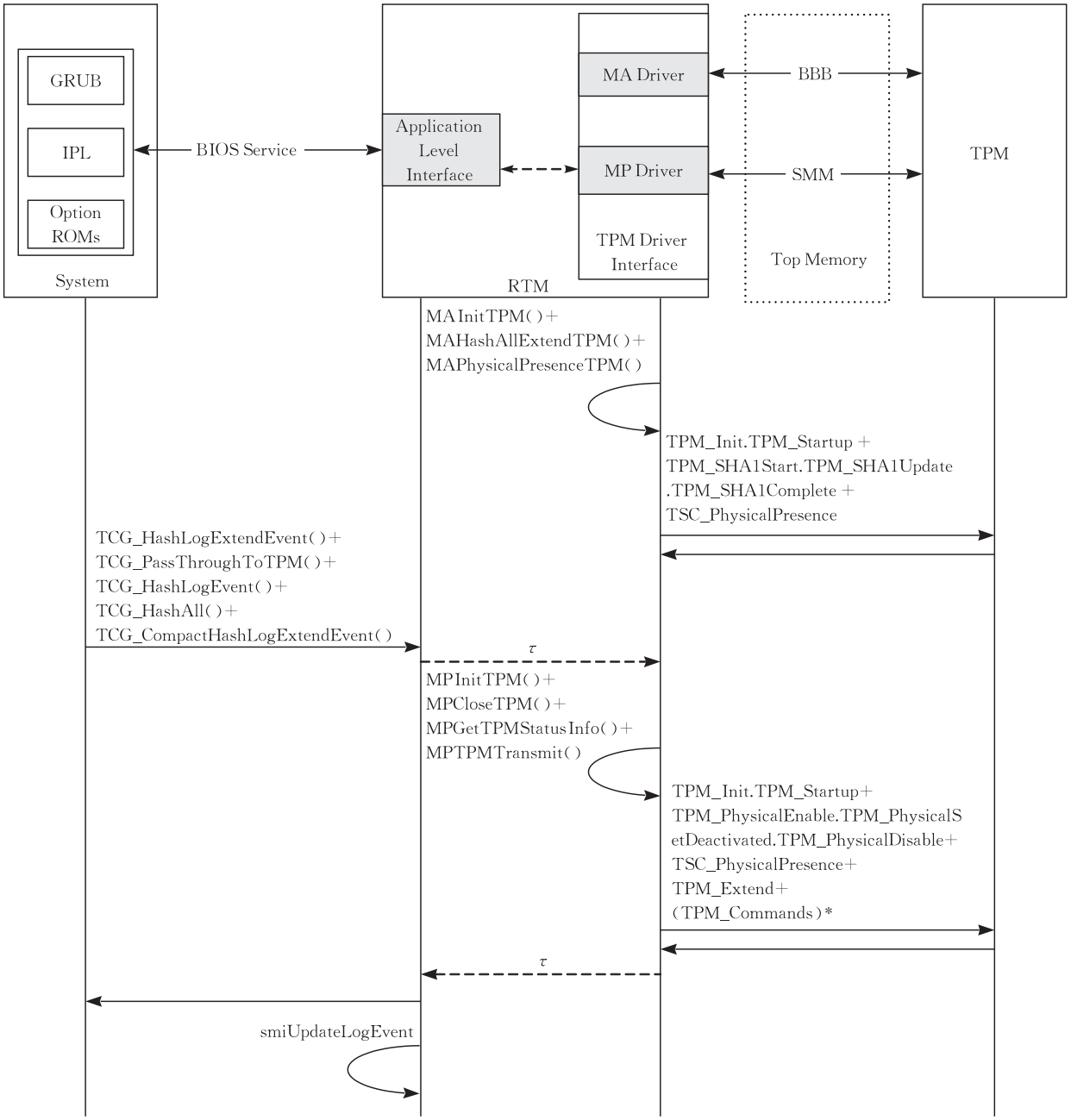

图 2 可信计算平台规范说明输入输出接口

System 中的动作 $\overline{a\_System\_ACPI}$ 和 $\overline{tcg\_PassThroughToTPM}$ 分别表示创建度量日志请求和应用层访问 TPM 请求；RTM 中的动作 $\overline{ma\_HashAllExtendTPM}$ 和 $\overline{mp\_TPMTransmit}$ 表示 BIOS 层访问 TPM 请求；对于前面 System 和 RTM 的输出动作，TPM 对应的有 3 个同步动作：$ma\_HashAllExtendTPM.w\_PCR.TPM$、$mp\_TPMTransmit.w\_PCR.TPM$ 和 $tcg\_PassThroughToTPM.w\_PCR.TPM$. 其余同步动作用于保证代码和数据的完整性，防范 TOCTTOU 攻击[21].

3.3 安全性分析

我们在文献[9]中刻画出了信任链交互模型，其中 RTM 与 TPM 组成 TBB，RTM 与 TPM 之间通过位于高端内存的两个驱动程序进行通信，其动作对于 System 来说是不可见的. 我们再对 RTM 做进一步细化，RTM 被划分为 CRTM 和 POSTBIOS，CRTM 与 POSTBIOS 在 RTM 内部进行复合，其内部动作对于 TPM 和 System 来说也是不可见的. 并且我们认为 RTM 和 TPM 是绝对可信的，所有的动作都属于高安全级动作. 另一方面，RTM 与 TPM 组成的 TBB 需要再次与 System 进行复合，由于

System 中的所有组件的安全级别都要比 TBB 低，因此 System 中的所有动作都属于低安全级动作，这里我们考虑的问题是：System 能够观察出 TBB 中的高安全级动作或者输出策略吗？在分析该问题之前，我们先给出相关的符号定义.

in_i^E：系统 E 的外部输入序列；

out_j^E：系统 E 的外部输出序列.

c_m：系统 E 的指令输入序列；

$\bar{c}_m$：系统 E 的指令输出序列.

d_n：系统 E 的数据输入序列；

$\bar{d}_n$：系统 E 的数据输出序列.

s_y：系统 E 的同步输入序列；

$\bar{s}_y$：系统 E 的同步输出序列.

r_k：系统 E 的结果输入序列；

$\bar{r}_k$：系统 E 的结果输出序列.

这里的外部输入和外部输出序列是指系统通过吸收外部激励动作或事件而产生的外部输出动作或事件. 对于信任链复合模型而言，System 以触发相关动作与 TBB 进行通信而完成某种功能，如 IPL 通过 RTM 来访问 TPM，进而完成对 GRUB 的完整性度量和完整性存储. 因此 IPL 的输入事件对于 RTM 来说就是一种外部激励输入，经过内部对该输入动作的处理，RTM 完成与 TPM 之间的内部同步并将结果作为外部输出返还给 IPL.

下面我们对 CRTM 与 TPM 的复合、POSTBIOS 与 TPM 的复合、System 与 TBB 的复合分别进行讨论.

3.3.1　CRTM 与 TPM 的复合

在信任链建立初期，CRTM 需要通过 TPM-MA 驱动对 POSTBIOS 进行完整性度量和完整性存储，为了方便表示，这里我们用 in_0^{CRTM}，$\overline{out}_0^{\mathrm{CRTM}}$ 分别表示完整性度量输入和输出，in_1^{CRTM}，$\overline{out}_1^{\mathrm{CRTM}}$ 分别表示完整性存储输入和输出，它们都是 CRTM 对外所呈现的视图；c_0，$\bar{c}_0$ 表示 CRTM 发给 TPM 的指令序列输入和输出，d_0，$\bar{d}_0$ 表示 CRTM 发给 TPM 的数据序列；s_y，$\bar{s}_y$ 表示 CRTM 与 TPM 之间的同步信号，对于实际的信任链系统而言同步信号通常是硬件总线；符号 r_0，$\bar{r}_0$，r_1，$\bar{r}_1$ 对应着 TPM 所完成的完整性度量结果和完整性存储结果.

我们在前面已经说明了 CRTM 和 TPM 中的所有动作都属于高安全级动作，因此当 CRTM 与 TPM 进行复合之后，需要同步的高安全级动作统统都转换为内部动作 τ，该动作和高安全级动作对于低安全级视图是不可见的. CRTM 与 TPM 的复合模型如图 3 所示.

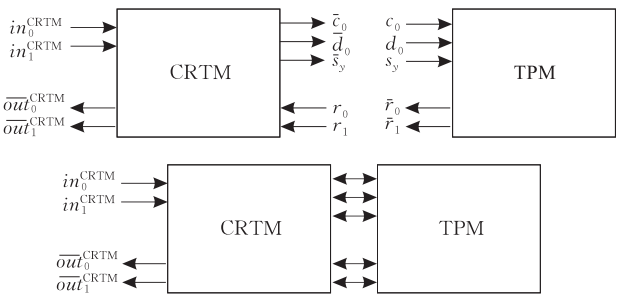

图 3　CRTM 与 TPM 的复合模型

用 SPA 进行描述的 CRTM 与 TPM 复合系统的 SPA 表示如下所示：

$$\mathrm{CRTM\_TPM} \overset{\mathrm{def}}{=} (\mathrm{CRTM} \parallel \mathrm{TPM}) \backslash Sy;$$

$$\mathrm{CRTM} \overset{\mathrm{def}}{=} in_0^{\mathrm{CRTM}}.\bar{s}_y.\bar{c}_0.\underline{0} + in_1^{\mathrm{CRTM}}.\bar{s}_y.\bar{d}_0.\underline{0} + r_0.out_0^{\mathrm{CRTM}}.\underline{0} + r_1.out_1^{\mathrm{CRTM}}.\underline{0};$$

$$\mathrm{TPM} \overset{\mathrm{def}}{=} s_y.c_0.r_0.\mathrm{TPM} + s_y.d_0.r_0.\mathrm{TPM} + s_y.c_0.r_1.\mathrm{TPM} + s_y.d_0.r_1.\mathrm{TPM};$$

$$Sy = \{s_y, \bar{s}_y, c_0, \bar{c}_0, d_0, \bar{d}_0, r_0, \bar{r}_0, r_1, \bar{r}_1\};$$

$$Act_H = \{s_y, \bar{s}_y, c_0, \bar{c}_0, d_0, \bar{d}_0, r_0, \bar{r}_0, r_1, \bar{r}_1, in_0^{\mathrm{CRTM}}, out_0^{\mathrm{CRTM}}, in_1^{\mathrm{CRTM}}, out_1^{\mathrm{CRTM}}\}.$$

经过复合算子操作后的系统如图 4 所示，为了图形显示方便，我们只选取了并发系统中的一条分支，可以看出，该分支中含有 3 个内部动作 τ 和两个高安全级动作 in_0^{CRTM}，out_0^{CRTM}.

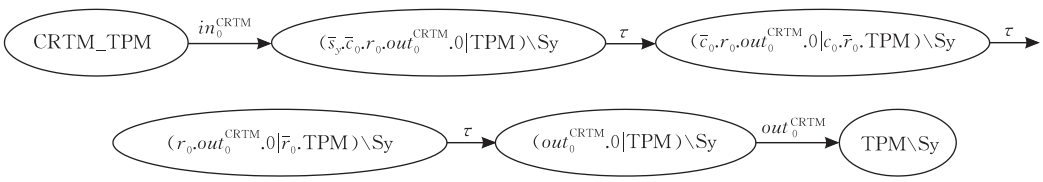

图 4　CRTM 与 TPM 的 LTS 系统

3.3.2　POSTBIOS 与 TPM 的复合

当 CRTM 对 POSTBIOS 度量完毕后，POSTBIOS 需要对 System 的第一个启动组件进行完整性度量和完整性存储. 这里的符号描述和 SPA 描述与上一小节基本相同，不再赘述.

3.3.3　System 与 TBB 的复合

从上面两小节可以知道，CRTM、POSTBIOS 和 TPM 所组成的 TBB 中所有的输入输出动

作都是高安全级动作.那么对于动作序列 $in_0^{\mathrm{CRTM}}.\tau.\tau.\tau.\overline{out}_0^{\mathrm{CRTM}}$ 和 $in_0^{\mathrm{POSTBIOS}}.\tau.\tau.\tau.\overline{out}_0^{\mathrm{POSTBIOS}}$ 来说,由于 in_0^{CRTM} 和 $\overline{out}_0^{\mathrm{CRTM}}$ 是高安全级动作,且没有低安全级输入/输出对高安全级动作的依赖,因此复合系统 CRTM_TPM、POST_TPM 都满足 NDI 和 NDS 安全属性.

那么满足 NDI 和 NDS 安全性质的 TBB 与 System 复合之后是否仍然满足 NDI 和 NDS 安全属性呢?我们先给出 System 与 TBB 的复合模型,如图 5 所示.该模型和上面两个复合模型的不同之处在于,由于 System 中的所有动作都是低安全级动作,因此,当 System 与 TBB 进行复合的时候,完成完整性度量和完整性存储任务的指令输入输出序列、数据输入输出序列和结果输入输出序列,不能简单地将它们视为同步操作而化简为内部动作,因此这里需要对这些序列进行安全视图上的区分.

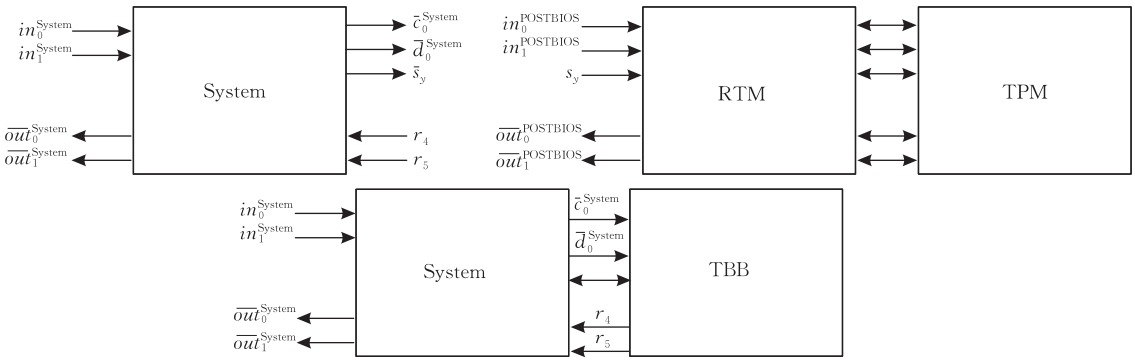

图 5 System 与 TBB 的复合

这里我们将同步动作集 Sy 仅包含信号同步动作 $\{s_y,\bar{s}_y\}$,系统 System 与 TBB 进行交互的动作集包括 $\{\bar{c}_0^{\mathrm{System}},\bar{d}_0^{\mathrm{System}},r_4,r_5\}$,分别表示指令序列、数据序列、完整性度量结果序列和完整性存储结果序列.TBB 的输入输出集分别为 $\{in_0^{\mathrm{POSTBIOS}},in_1^{\mathrm{POSTBIOS}},\overline{out}_0^{\mathrm{POSTBIOS}},\overline{out}_1^{\mathrm{POSTBIOS}}\}$,复合系统的 SPA 描述如下所示.

$$\mathrm{System\_TBB}\overset{\mathrm{def}}{=}(\mathrm{System}\parallel\mathrm{TBB})\backslash Sy;$$

$$\begin{aligned}\mathrm{System}\overset{\mathrm{def}}{=}&in_0^{\mathrm{System}}.\bar{s}_y.\bar{c}_0^{\mathrm{System}}.r_4.out_0^{\mathrm{System}}.\mathrm{System}+\\&in_0^{\mathrm{System}}.\bar{s}_y.\bar{d}_0^{\mathrm{System}}.r_4.out_0^{\mathrm{System}}.\mathrm{System}+\\&in_1^{\mathrm{System}}.\bar{s}_y.\bar{c}_0^{\mathrm{System}}.r_5.out_0^{\mathrm{System}}.\mathrm{System}+\\&in_1^{\mathrm{System}}.\bar{s}_y.\bar{d}_0^{\mathrm{System}}.r_5.out_0^{\mathrm{System}}.\mathrm{System};\end{aligned}$$

$$\begin{aligned}\mathrm{TBB}\overset{\mathrm{def}}{=}&s_y.in_0^{\mathrm{POSTBIOS}}.\overline{out}_0^{\mathrm{POSTBIOS}}.\mathrm{TBB}+\\&s_y.in_0^{\mathrm{POSTBIOS}}.\overline{out}_0^{\mathrm{POSTBIOS}}.\mathrm{TBB}+\\&s_y.in_1^{\mathrm{POSTBIOS}}.\overline{out}_1^{\mathrm{POSTBIOS}}.\mathrm{TBB}+\\&s_y.in_1^{\mathrm{POSTBIOS}}.\overline{out}_1^{\mathrm{POSTBIOS}}.\mathrm{TBB};\end{aligned}$$

$$Sy=\{s_y,\bar{s}_y\};$$

$$Act_H=\{s_y,in_0^{\mathrm{POSTBIOS}},\overline{out}_0^{\mathrm{POSTBIOS}},in_1^{\mathrm{POSTBIOS}},\overline{out}_1^{\mathrm{POSTBIOS}}\}.$$

若考虑所有的动作集合那么最终的 LTS 系统将异常庞大,根据上述描述,我们只考虑动作集合 $\{in_0^{\mathrm{System}},\overline{out}_0^{\mathrm{System}},s_y,\bar{s}_y,\bar{c}_0^{\mathrm{System}},r_4\}$.为了显示方便,我们对变迁系统的片段进行分析,如图 6 所示.可以看出,当系统 System_TBB 处于状态 $\overline{out}_0^{\mathrm{System}}$.System $\parallel$ $\overline{out}_0^{\mathrm{POSTBIOS}}$.TBB$\backslash S_y$ 时,可以选择动作 $\overline{out}_0^{\mathrm{POSTBIOS}}$ 或者 $\overline{out}_0^{\mathrm{System}}$,由于动作 $\overline{out}_0^{\mathrm{System}}$ 是一个低安全级输出,到达状态 $\overline{out}_0^{\mathrm{System}}$.System $\parallel$ $\overline{out}_0^{\mathrm{POSTBIOS}}$.TBB$\backslash S_y$ 是通过动作 r_4 或者 in_0^{POSTBIOS},而动作 in_0^{POSTBIOS} 是一个高安全级输入,也就是低安全级输出依赖高安全级输入,因此复合系统 System_TBB 不再符合 NDI 安全属性,对于实际的信任链系统而言,我们将在 3.5 节详细说明 System 中的哪些动作依赖于 TBB 中的动作.

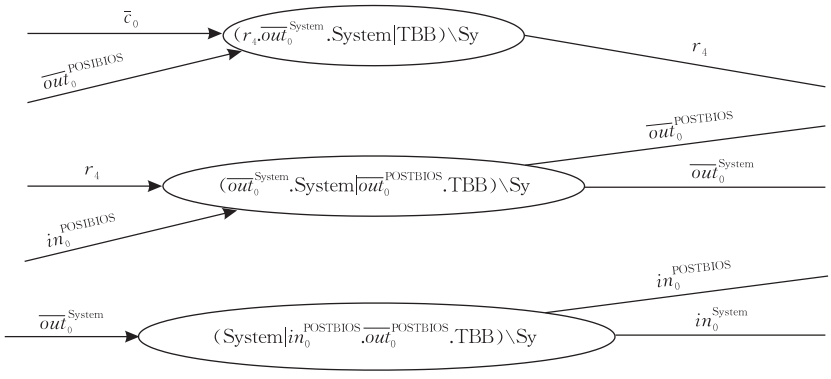

图 6 System 与 TBB 的部分 LTS 系统

3.4　进一步的分析

通过 3.3 节对信任链复合模型的分析得知，System 与 TBB 的复合系统 System_TBB 既然不满足 NDI 安全属性，同样也就不满足 NDS 属性. 那么究竟 System_TBB 要满足什么样的条件才能达到这一要求呢？定义 11 对进程 E 的可复合不可演绎性质定义采用的是 NDC 模型，并采用视图对其进行定义. 接下来我们将使用迹语义和互模拟语义对 NDC 进行了定义，通过语义表示上的差别对 NDC 进行安全属性上的提升. 我们先给出迹语义下可复合的不可演绎 NDC 安全属性的定义.

定义 13.　$E\in\mathcal{E}$，E 满足 NDC 性质，如果 $\forall\Pi\in\mathcal{E}_H$，$E/Act_H\approx_T((E|\Pi)\backslash H)/Act_H$.

Focardi 在文献[12]中认为 NDC 安全属性和 NDS 安全属性是等价的，并用互模拟语义对 NDC 安全属性进行了定义. 因为对于存在死锁的系统而言，基于迹语义的安全属性不能检测死锁的发生，譬如高安全级进程拒绝进行同步而导致的死锁，那么低安全级进程可以通过有效的尝试去探测高安全级进程的行为，即输入和输出. 因此，我们需要借助 CCS 中的弱互模拟语义[19]解决该问题，基于互模拟语义的 NDC 安全属性表示如下.

定义 14.　$E\in\mathcal{E}$，E 满足 BNDC 性质，如果 $\forall\Pi\in\mathcal{E}_H$，$E\in\text{BNDC}\Leftrightarrow E/Act_H\approx_B(E|\Pi)\backslash Act_H$.

接下来，我们将使用定义 14 所定义的 BNDC 性质刻画信任链中 TPM、RTM 和 System 之间的关系，并论证在何种情况下这三者组成的复合系统才能满足 BNDC 安全属性，即复合系统中不存在非法信息流.

命题 1.　令 $\text{TBB}\triangleq(\text{RTM}|\text{TPM})\backslash Sy$，其中 $Sy\subseteq(f(Act_H^{\text{RTM}})\cap Act_H^{\text{TPM}})\cup(f(Act_H^{\text{TPM}})\cap Act_H^{\text{RTM}})$，那么 $\text{TBB}\in\text{BNDC}$.

证明.　根据 TBB 的定义可知，RTM 和 TPM 中的所有动作都是高安全级动作，又根据定义 1，有 $f(Act_H^{\text{RTM}})=Act_H^{\text{RTM}}$，$f(Act_H^{\text{TPM}})=Act_H^{\text{TPM}}$，因此 $Sy\subseteq(f(Act_H^{\text{RTM}})\cap Act_H^{\text{TPM}})\cup(f(Act_H^{\text{TPM}})\cap Act_H^{\text{RTM}})$ 可以化简为 $Sy\subseteq Act_H^{\text{TPM}}\cap Act_H^{\text{RTM}}$，那么 $(\text{RTM}|\text{TPM})\backslash Sy$ 把 RTM 和 TPM 的所有接口动作都转化为内部动作 τ，令余下的所有高安全级动作集合为 $Act_H^{\text{RTM\_TPM}}$.

令 $E=\text{TBB}$，显然 $\forall E'$，E''，$\mu\in\mathcal{L}^*$，$h\in Act_H^{\text{RTM\_TPM}}$，$E\xrightarrow{\mu}E'$，$E'\xrightarrow{h}E''$，其中 μ 属于集合 $Act_H^{\text{RTM\_TPM}}$，那么 $E'\backslash Act_H^{\text{RTM\_TPM}}$ 和 $E''\backslash Act_H^{\text{RTM\_TPM}}$ 将消去动作集合 $Act_H^{\text{RTM\_TPM}}$，根据定义 6 可知，$E'\backslash Act_H\approx_B E''\backslash Act_H$ 成立.

综上所述，命题成立.　　证毕.

命题 2.　令 $\text{SRTM}\triangleq(\text{TPM}|\text{RTM}|\text{System})\backslash Sy$，其中 $Sy\subseteq(f(Act_H^{\text{RTM\_TPM}})\cap Act_H^{\text{System}})\cup(f(Act_H^{\text{System}})\cap Act_H^{\text{RTM\_TPM}})$，那么 $\text{SRTM}\in\text{BNDC}$.

证明.　设 $\text{SRTM}\xrightarrow{\mu}\text{SRTM}'$，$\text{SRTM}'\xrightarrow{h}\text{SRTM}''$，那么有 $\mu\subseteq Act_L^{\text{RTM,TPM}}\cup Act_L^{\text{System}}$，$h\subseteq Act_H^{\text{RTM,TPM}}\cup Act_H^{\text{System}}$.

又 $Act_H=Act_H^{\text{RTM,TPM}}\cup Act_H^{\text{System}}$，显然 $Sy\subseteq Act_H$，因此有 $\text{SRTM}\backslash Sy=(\text{SRTM}\backslash Sy)\backslash Act_H$，只需证明 $\text{SRTM}'\backslash Sy\approx_B\text{SRTM}''\backslash Sy$.

若 $h\subseteq Sy$，则 $\text{SRTM}'=\text{SRTM}''$，命题显然成立.

若 $h\not\subseteq Sy$，根据定义 3 有 $T(\text{SRTM}'')=T(\text{SRTM}')$，根据定义 5，显然有 $\text{SRTM}'\approx_B\text{SRTM}''$，而 $(\text{SRTM}'\backslash Sy)\subseteq\text{SRTM}'$，$(\text{SRTM}''\backslash Sy)\subseteq\text{SRTM}''$，因此 $\text{SRTM}'\backslash Sy\approx_B\text{SRTM}''\backslash Sy$.

综上所述，命题成立.　　证毕.

命题 1 和命题 2 说明了如果复合系统的 Act_H 集合中的元素对偶和 Sy 集合的所有元素存在着双射关系，也就是一一映射，那么 E 一定满足 BNDC 安全性质，这意味着从低级观察中得不到任何高级活动的信息，这与 Sutherland 提出的映成函数的观点是等价的.

3.5　实例分析

下面我们以信任链接口安全模型为依据，以 TCG 的可信 PC 信任链规范为对象，刻画信任链实体动作，抽象出高低安全级进程的输入和输出，用 CoPS 工具对安全属性进行验证，找出规范中存在的安全缺陷.

3.5.1　接口安全等级划分

在信任链运行期间，RTM 和 TPM 通过接口进行了复合，一方面，我们需要发现这种复合是否能够保护高级输入输出，另一方面，RTM 和 TPM 所组成的 TBB 子系统与 System 又进行了复合，我们需要进一步发现该复合能否满足不可演绎安全性. 为了能验证这些安全属性，我们给出了限制集合 Sy 和高安全级别动作集合 $acth$，如下所示.

$Sy=\{ma\_HashAllExtendTPM,\ mp\_TPMTransmit,\ a\_RTM\_ACPI,\ e\_POSTBIOS,\ r\_OptionRoms,\ e\_OptionRoms,\ e\_IPL\}$,

$acth=\{ma\_HashAllExtendTPM,\overline{ma\_HashAllExtendTPM},\ mp\_TPMTransmit,\overline{mp\_TPMTransmit},\ W\_PCR,\ tcg\_PassThroughToTPM,\overline{smiUpdateLogEvent},\ a\_System\_ACPI,\ r\_POSTBIOS,\ r\_OptionRoms,\ a\_RTM\_ACPI,\overline{a\_RTM\_ACPI},\ e\_POSTBIOS,$

$\overline{e\_POSTBIOS}$, $e\_OptionRoms$, $\overline{e\_OptionRoms}$, r_IPL, $\overline{e\_IPL}$}.

Sy 集合用于同步 RTM 与 TPM 之间的输入输出,可以看出集合

$drv=\{ma\_HashAllExtendTPM$, $\overline{ma\_HashAllExtendTPM}$, $mp\_TPMTransmit$, $\overline{mp\_TPMTransmit}\}$

也出现在 $acth$ 中,这是因为 drv 中的动作是运行在 SMM 模式下,此时 CPU 是运行在最高特权级上,因此将集合 drv 中的所有元素都视为高安全级动作,$tcg\_PassThroughToTPM$ 动作是 TPM 响应应用层请求的输入动作,该动作和 w_PCR 都应视为高安全级动作,a_RTM_ACPI 动作用于创建度量日志,其后续动作是 $\overline{smiUpdateLogEvent}$,这两个动作的运行环境也同样是在 SMM 模式下. 除此以外,高安全级别动作集合 $acth$ 中还包括了 $r\_POSTBIOS$, $e\_POSTBIOS$, $r\_OptionRoms$, $\overline{e\_OptionRoms}$, r_IPL, $\overline{e\_IPL}$. 附录表 1 对每个动作的含义给出了描述.

3.5.2 接口安全测试

为了验证可信计算PC规范说明的SRTM是否满足 BNDC,我们使用 CoPS 对其进行验证,CoPS 是 Pivato 等人开发的用于自动化验证多级安全属性的工具,用于验证系统是否满足 P_BNDC、PP_BNDC 或 SBNDC 等安全性质[22].

将以上用 SPA 描述的 System、RTM 和 TPM 的接口函数转化为 CoPS 后,得到的验证结果如图 7 所示. 在验证过程中我们发现,根据上述划分的 SRTM 并不满足 BNDC 安全性质,这是因为在 System 中存在着低安全级动作对高安全级动作的依赖,因此低安全级的 System 就可以通过隐式的方法获得 TBB 中的高级输入或者输出;我们可以通过消除这些依赖使得修改后的信任链规范说明满足 BNDC 安全属性,例如,我们将高安全级输出动作 w_PCR 作为同步动作,那么也就意味着该动作对于 TPM 之外的实体而言是不可见的,因此就消除了低安全级动作与 w_PCR 之间的关联关系,这样的系统将满足定义 14 中所描述的 BNDC 性质,如图 8 所示,同时也证实了命题 2 的正确性. 表 1 给出了 SRTM 中所存在的低安全级动作对高安全级动作的依赖.

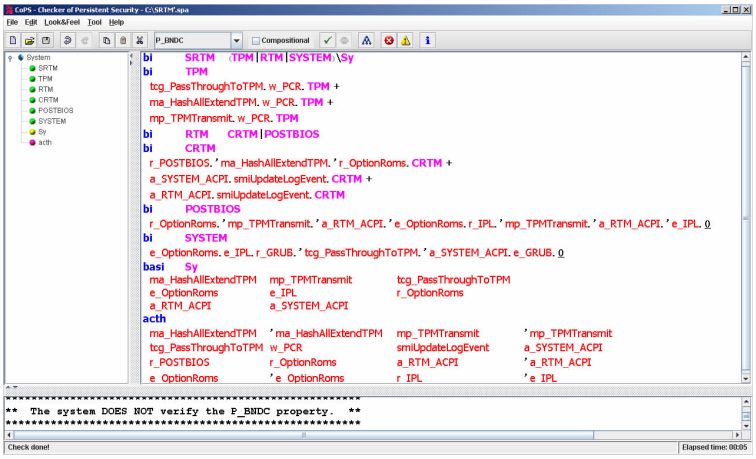

图 7 CoPS 验证修改前的信任链规范说明

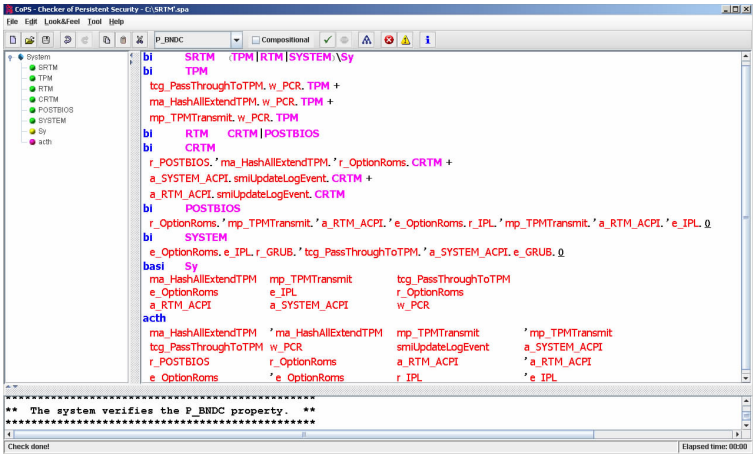

图 8 CoPS 验证修改后的信任链规范说明

命题 1 和命题 2 所提出的信任链复合系统是理想情况下的安全系统，实际情况下的规范说明和系统实现很难全部满足这些安全属性，譬如，低级输出 $tcg\_PassThroughToTPM$ 影响着高级输出 w_PCR，这说明低级进程可以间接地对 PCR 进行任意迭代；低级输出 $'a\_System\_ACPI$ 影响着高级输出 $smiUpdateLogEvent$，这表明低级进程能通过高级进程对度量日志进行追加操作，而这些进程动作都是远程证明协议的一部分，可见它们将直接影响其安全性，关于这一点我们将在以后进行讨论.

4　结　论

信任链是由不同安全属性的子系统组成的复合系统，复合后的高级子系统安全性质应该保持. 其中的每一个子系统都满足一定的安全性质，这些子系统组合而成的系统是否仍然满足给定的安全性质？另一方面，在信任链建立阶段，各个子系统之间的访问是否存在违反规范约束的行为或操作，进而破坏信任传递，甚至获取机密信息？

针对上述问题，本文将信任链抽象为多个实体的进程交互模型，通过接口细化的方式对进程间的高低安全级动作进行描述，并以信任链规范为实例，通过抽取规范中的高低等级输入输出，找出它们之间的关联规则，把对信任链安全属性的验证转换为对表示进程的代数项的语法检查上，我们认为高级进程的动作元素对偶与限制集合满足双射关系是信任链满足复合安全性质的充分条件.

结合实际规范说明，我们用 CoPS 对提出的安全模型进行了安全验证，发现其中潜在的安全缺陷及其可能造成的影响.

参　考　文　献

[1] Zhang Huan-Guo, Yan Fei, Fu Jian-Ming et al. Research on theory and key technology of trusted computing platform security testing and evaluation. Science in China Series F: Information Sciences, 2010, 53(3): 405-433
(张焕国，严飞，傅建明等. 可信计算平台测评理论及其关键技术研究. 中国科学 F 辑：信息科学，2010，40(2)：167-188)

[2] Abadi M, Wobber T. A logical account of NGSCB//Proceedings of the Formal Techniques for Networked and Distributed Systems, FORTE 2004. Madrid, Spain. LNCS 3235. 2004: 1-12

[3] Chen S, Wen Y, Zhao H. Formal analysis of secure bootstrap in trusted computing//Proceedings of the 4th International Conference on Autonomic and Trusted Computing. Hong Kong, China. LNCS 4610. Springer, 2007: 352-360

[4] Gürgens S, Rudolph C, Scheuermann D et al. Security evaluation of scenarios based on the TCG's TPM specification//Biskup Joachim, Lopez Javier eds. Proceedings of the 12th European Symposium on Research in Computer Security (ESORICS). Dresder, Germany. LNCS 4734. Springer, 2007: 438-453

[5] Millen J, Guttman J, Ramsdell J et al. Analysis of a measured launch. The MITRE Corporation, Bedford, MA: Technical Reports 07-0843, 2007

[6] Lin A H. Automated analysis of security apis [M. S. dissertation]. Massachusetts Institute of Technology, 2005

[7] Deepak G, Jason F, Dilsun K et al. Towards a theory of secure systems. Cylab, Carnegie Mellon University, Pattsburgh, PA: Technical Reports CMU-CyLab-08-003, 2008

[8] Datta A, Franklin J, Garg D et al. A logic of secure systems and its application to trusted computing. Cylab, Carnegie Mellon University, Pattsburgh, PA: Technical Reports CMU-CyLab-09-001, 2009

[9] Xu Ming-Di, Zhang Huan-Guo, Yan Fei. Testing on trust chain of trusted computing platform based on labeled transition system. Chinese Journal of Computers, 2009, 32(4): 635-645(in Chinese)
(徐明迪，张焕国，严飞. 基于标记变迁系统的可信计算平台信任链测试. 计算机学报，2009，32(4)：635-645)

[10] Zhou Wei, Yin Qing, Wang Qing-Xian. Abstract security properties in process algebra. Journal of Computer Research and Development, 2005, 42(12): 2100-2105(in Chinese)
(周伟，尹青，王清贤. 进程代数上的抽象安全性质. 计算机研究与发展，2005，42(12)：2100-2105)

[11] Wang Li-Bin, Chen Ke-Fei. Language-based security model. China Information Security, 2005, 7: 214-218(in Chinese)
(王立斌，陈克非. 基于程序设计语言的安全模型. 信息安全与通信保密，2005，7：214-218)

[12] Focardi R, Gorrieri R. Classification of security properties (Part I: Information Flow)//Proceedings of the Foundations of Security Analysis and Design-Tutorial Lectures. Bertinoro, Italy. LNCS 2171. Springer, 2001: 331-396

[13] Smith G, Volpano D. Secure information flow in a multithreaded imperative language//Proceedings of the 25th ACM Symposium on Principles of Programming Languages (POPL 98). San Diego, CA, 1998, 1: 355-364

[14] Goguen J A, Meseguer J. Security policies and security models//Proceedings of the IEEE Symposium on Security and Privacy. Oakland, CA, 1982, 12: 11-20

[15] O'Halloran C. A calculus of information flow//Proceedings of the 1st European Symposium on Research in Computer Security. Toulouse, France, 1990: 147-159

[16] Sutherland D. A model of information//Proceedings of the 9th National Computer Security Conference. Gaithersburg, MD, 1986: 175-183

[17] Guttman J D, Nadal M E. What needs securing?//Proceedings of the Computer Security Foundations Workshop. Franconia, New Hampshire, USA, IEEE Computer Society, 1988: 34-57

[18] Keller R M. Formal verification of parallel programs. Communications of the ACM, 1976, 19(7): 371-384

[19] Milner R. Communication and Concurrency. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989

[20] McCullough D. Specifications for multi-level security and a hook-up property//Proceedings of the IEEE Symposium on Research in Security and Privacy. Los Alamitos, CA, IEEE Computer Society Press, 1987: 161-166

[21] Shi E, Perrig A, Doorn L V. BIND: A fine-grained attestation service for secure distributed systems//Proceedings of the IEEE Symposium on Security and Privacy. Oakland, California, 2005: 154-168

[22] Piazza C, Pivato E, Rossi S. CoPS-checker of persistent security//Proceedings of the Tools and Algorithms for the Construction and Analysis of Systems. Barcelona, Spain. LNCS 2988. Springer, 2004: 144-152

附表 1 信任链动作安全等级划分

进程	动作	安全级别	输入/输出	同步	描述
TPM	*tcg_PassThroughToTPM*	高	输入	是	BIOS 中断服务例程提供的 TPM 访问接口
TPM	*ma_HashAllExtendTPM*	高	输入	是	BIOS Boot Block 阶段的 TPM 访问接口
TPM	*mp_TPMTransmit*	高	输入	是	POST BIOS 阶段的 TPM 访问接口
RTM	*r_POSTBIOS*	高	输入	否	读取 POST BIOS 代码
RTM	$\overline{ma\_HashAllExtendTPM}$	高	输出	是	CRTM 访问 TPM 请求
RTM	$\overline{e\_POSTBIOS}$	高	输出	是	加载 POST BIOS 代码至内存
RTM	*a_System_ACPI*	高	输入	是	BIOS 外部的度量日志创建请求
RTM	*a_RTM_ACPI*	高	输入	是	BIOS 内部的度量日志创建请求
RTM	$\overline{smiUpdateLogEvent}$	高	输出	否	追加度量日志
RTM	*e_POSTBIOS*	高	输入	是	执行 POST BIOS 代码
RTM	*r_OptionRoms*	高	输入	否	读取 OptionRoms 信息
RTM	$\overline{mp\_TPMTransmit}$	高	输出	是	POST BIOS 访问 TPM 请求
RTM	$\overline{a\_RTM\_ACPI}$	高	输出	是	POST BIOS 追加度量日志请求
RTM	$\overline{e\_OptionRoms}$	高	输出	是	加载 OptionRoms 代码
RTM	$\overline{r\_IPL}$	高	输出	是	读取 IPL 代码
RTM	$\overline{e\_IPL}$	高	输出	是	加载 IPL 代码至内存
System	*e_OptionRoms*	低	输入	是	执行 OptionRoms 代码
System	*e_IPL*	低	输入	是	执行 IPL 代码
System	$\overline{r\_GRUB}$	低	输出	否	读取 GRUB 代码
System	$\overline{tcg\_PassThroughToTPM}$	低	输出	是	System 访问 TPM 请求
System	$\overline{a\_System\_ACPI}$	低	输出	是	System 追加度量日志请求
System	$\overline{e\_GRUB}$	低	输出	否	执行 GRUB 代码

XU Ming-Di, born in 1980, Ph. D., engineer. His research interests include trusted computing and system security.

ZHANG Huan-Guo, born in 1945, professor, Ph. D. supervisor. His research interests include information security and trusted computing.

ZHAO Heng, born in 1966, Ph. D., senior engineering. Her research interests focus on software engineeirng.

LI Jun-Lin, born in 1966, researcher. His research interests focus on software engineering.

YAN Fei, born in 1980, lecturer. His research interests focus on trusted computing.

Background

This research work is based on author's research direction which aimed at testing and evaluation of Trusted Computing Platform.

At present, the Trusted Computing has become a major development trend in the field of computer security. China's Trusted Computing technology and industry is in a phase of vigorous development, with the Trusted Computing product maturity, the Trusted Computing Platform applications will become more and more widely. Trusted Computing Platform can be used to greatly enhance the security of information systems, but must be under the Trusted Computing Platform evaluation, otherwise, neither the quality of trusted computing products, nor the security of information systems can be guaranteed.

This paper is supported by the National High Technology Research and Department Program (863 Program) of China (2007AA01Z411), National Natural Science Foundation of China (60673071) and Open Fund of the Key Laboratory of Aerospace Information Security and Trust Computing, Ministry of Education.