基于无干扰理论的信任链传递模型

陈 亮[1] 曾荣仁[1] 李 峰[1,2] 杨伟铭[1]
(中国人民解放军后勤科学研究所 北京 100071)[1] (北京航空航天大学 北京 100191)[2]

摘 要 针对现有的信任链传递模型可用性不强、缺乏将信任链扩展到网络环境的缺点,提出了一种新的基于无干扰理论的信任链传递模型。该模型将系统抽象为进程、动作和执行,从可信根出发,通过度量程序及其动态库完整性来保证进程静态可信;分析交互进程之间的关系,利用无干扰理论判定其合法性;通过对接入终端的可信度量,将信任链扩展到整个网络系统。最后给出了相应的形式化定义及安全性证明。

关键词 无干扰理论,进程可信,系统运行可信,安全接入可信
中图法分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.10.026

Trust Chain Transfer Model Based on Non-interference Theory

CHEN Liang[1] ZENG Rong-ren[1] LI Feng[1,2] YANG Wei-ming[1]
(Logistics Science Institute of PLA, Beijing 100071, China)[1] (Beihang University, Beijing 100191, China)[2]

Abstract As the existing trust chain transfer model is lack of availability and has the disadvantage of extending the credibility of the trust chain to the network system, a new model of trust chain transfer based on non-interference theory was proposed. The model abstracts the system to processes, actions and implementation. The model measures the integrity of static process and dynamic library to ensure the static process credible, uses non-interference theory analysis of the relationship between the interactive processes to determine its legitimacy, and extends the chain of trust to the whole network system through measuring the credibility of the access terminal. Finally, the corresponding formal definition and security proof were given.

Keywords Non-interference theory, Process credible, System operation credible, Secure access credible

1 引言

在信息安全实践的过程中,人们逐渐认识到大多数的安全隐患来自微机终端,其原因是当初设计 PC 时缺乏对其安全性的考虑,导致资源可任意被利用,尤其是执行代码可被修改、恶意程序可被植入等[1]。为了解决上述问题,TCG(Trusted Computing Group)提出了可信计算的思想来构建可信平台,并通过研究可信网络连接 TNC(Trusted Network Connection)将可信扩展到整个网络。

TCG 提出的可信思想可以总结为:首先构建一个信任根,再建立一条信任链,从信任根开始到硬件平台、操作系统、应用,一级验证一级,一级信任一级,最终将信任过程扩展到整个网络系统中。因此,信任链的建立是构建可信平台的关键。TCG 提出的信任链采用装载前度量的方式,从可信根开始依次对各模块进行完整性度量,因此也将其称为静态的信任链。由于系统平台上应用的多样性及无序性,静态信任链的构建并不适用于操作系统到应用程序及终端到网络之间的可信传递。

基于信息流的无干扰模型从动作和运行结果的角度建立系统安全策略模型,从而确保计算机系统的安全性和完整性。文献[2,3]在 Rushby 的无干扰理论[4]的基础上将系统安全域集实体化为进程集,给出了进程运行的可信条件,推导出系统运行可信定理,保证了终端的安全。但是其模型中的可信传递函数 check()和 clear()有待进一步证明,并且缺乏对进程的动态保护。文献[5]提出了基于非传递的无干扰理念的二元多级安全模型,在 Rushby 无干扰理论的基础上重新定义了清除函数,将传递的无干扰理论过渡到非传递的无干扰理论,并依据 BLP 和 Biba 模型保护了信息的机密性和完整性,然而同样存在文献[2]中的问题。文献[6]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合法,试图在不安全的操作系统中建立安全的应用支撑。文献[7]扩展了非传递无干扰理论,并试图通过重新定义静态干扰和动态干扰,使其支持进程自身代码的修改,但是其静态干扰和动态干扰的定义过于抽象,难以和实际的终端系统相对应,因此并不能在实际上完成其所描述的支持自身代码修改的功能。文献[8]提出了一种容忍非信任组件的可信终端模型,该模型利用可信组件对非信任组件的输出进行封装,保证了非信任组件在终端上的存在不会造成严重的安全威胁,实现了域间隔离和无干扰,保证了结果的可预测性和可控性。

此外,上述模型都只是从终端系统的角度构建安全模型,

到稿日期:2015-09-25 返修日期:2016-01-11

陈 亮(1991-),男,硕士,助理工程师,主要研究方向为网络与信息安全,E-mail:yixiu199151@sina.com;曾荣仁(1973-),男,硕士,高级工程师,主要研究方向为软件工程、信息安全;李 峰(1982-),男,博士,工程师,主要研究方向为计算语言学、数据挖掘;杨伟铭(1982-),男,硕士,工程师,主要研究方向为软件工程、大数据处理。

并未给出信任链从终端到网络上的扩展。本文在分析上述研究的基础上，将系统抽象为进程、动作和状态输出，提出了一个新的基于无干扰理论的信任链传递模型。通过对程序及动态链接库完整性的保护，确保进程静态可信；借鉴无干扰理论，分析进程运行过程的无干扰特性，保证进程动态可信；最后通过对接入终端的可信度量，将信任链传递到网络环境中，从而达到整个网络环境的可信，并基于有限状态机和无干扰理论对信任链在传递过程中的行为进行了形式化描述和证明。

2 信息流的无干扰理论

Geguen 和 Meseguer[9] 于 1985 年最早提出了信息流的无干扰思想。1992 年，Rushby[4] 提出了采用状态机分析的方法，并定义了系统关于传递和非传递无干扰策略是安全的，具体描述如下所示。

系统 $A=(S,A,O,D)$，其中 S 表示系统状态集，A 表示动作集，O 表示输出集，D 表示安全域集，定义在这些集合上的 4 个函数为：单步状态转换函数 $step:S\times A\rightarrow S$；系统运行函数 $run:S\times A^{*}\rightarrow S$；输出函数 $output:S\times A\rightarrow O$；主域函数 $dom:A\rightarrow D$ 表示系统每个动作所属的域。

在传递无干扰理论模型中，$\sim$表示安全域间的干扰关系。对于 $\alpha\in A,v\in D$，函数 $purge(\alpha,v)$表示从动作序列 α 中删除所有从不干扰域 v 发出动作后的动作序列，则有：

$$purge(\Lambda,v)=\Lambda$$

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

其中，Λ 表示空的动作序列，$dom(a)=v$。

系统 M 对策略$\sim$的安全条件为：

$$output(run(s_0,\alpha),a)=output(run(s_0,purge(\alpha,dom(a))),a)$$

3 基于无干扰理论的信任链传递模型

3.1 基本符号定义

定义 1 基于无干扰理论的终端系统表示为 $M=(S,P,L,A,O,F_M,U,R_M)$，其中：

S：系统状态集合，包含一个初始状态 $s_0\in S$，S_T 表示可信状态集合，且有 $s_0\in S_T$，$S_T\in S$，元素用 $s_1,s_2,\cdots$表示。

P：系统进程集合，用 $p_1,q,p_2,\cdots$表示。

L：表示系统加载的动态库集合，$\{p_i\}$代表执行进程 p_i 时系统所加载的集合 L 中动态库的元素。

A：表示系统动作集合，用 $a_1,a_2,\cdots$代表动作集合中的元素，用 $\alpha,\beta,\cdots$代表动作序列。

O：表示进程运行输出结果集合，系统由进程构成，因此系统的输出结果实际上是由进程输出的结果构成。

U：表示用户集合，即终端设备的使用者，元素用 $u_1,u_2,\cdots$表示，其中 U_t 表示可信用户集合，U_u 表示不可信用户集合。

F_M：函数集

1）单步执行函数 $step:S\times A\rightarrow S$，表示当完成一个系统动作之后，系统由当前状态转化为下一个状态。

2）动作序列执行函数 $run:S\times A^{*}\rightarrow S$，表示系统在运行一系列序列之后的状态变迁，是单步动作的叠加。

根据单步执行函数和动作序列执行函数，则有

$$run(s,\Lambda)=s$$

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

其中，Λ 表示一个空的动作序列，$\circ$表示动作间的连接操作。

3）结果输出函数 $output:S\times A\rightarrow O$，表示系统在当前状态下执行发出的动作后产生的结果。

4）动作与进程映射函数 $proc:A\rightarrow P$，其中 $proc(a)=p$，$a\in A,p\in P$表示动作 a 由进程 p 发出。

5）完整性判定函数 $int:P\times D^{*}\rightarrow N$，表示判定一个进程及其加载的动态库的完整性。其中 P 表示系统进程集合，D^{*} 表示进程所依赖的动态链接库集合，N 表示判定结果。

6）可信判定函数 $trust:P\rightarrow N$，表示判定一个进程的动态可信性。其中 P 表示系统进程集合，N 表示判定结果。

R_M：二元关系集合

1）状态集合上关于进程 p 的等价关系“$\overset{p}{=}$”；

2）进程集合 P 上的关系“$\sim>$”。当 $p\sim>q$，表示进程 p 的执行会对应用程序 q 的执行产生影响，即进程 p 对进程 q 有干扰。当 $p\not\sim>q$，表示进程 p 的执行对进程 q 的执行不产生影响，即进程 p 对进程 q 无干扰。

进程之间的干扰关系表明了其在运行过程中的信息流动关系，本模型的建立不是为了阻止进程之间的信息流动，而是使其符合系统规定的安全策略。即模型关注的是非预期干扰，防止恶意的、潜在的干扰行为。

定义 2 基于无干扰理论的安全域状态表示为 $V(Net,T,B,F_V)$，其中：

Net：安全域集合，即终端系统所属的安全域，元素用 $net_1,net_2,\cdots$表示。

T：节点集合，即系统所包含的终端节点，元素用 $t_1,t_2,\cdots$表示。

B：节点接入属性，$B=\{c,d\}$，其中 c 为接入操作，d 为断开连接操作。

F_V：函数集

1）接入请求函数，对于 $x\in A$，$rq(t,net,x)$表示对安全域 net 的 x 请求操作。

2）系统接入状态转换函数 $R\times M\rightarrow D\times M$，其中 R 为请求操作集，D 为判定函数，输出结果为 yes 或 no，yes 表示允许执行请求的动作，no 表示不允许执行请求的动作。

定义 3 对于 $\forall p\in P$ 和一个动作序列 $\alpha\in A^{*}$，清除函数 $purge(\alpha,p):A^{*}\times P\rightarrow A^{*}$ 归纳定义为：

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

$$purge(a\circ\alpha,p)=\begin{cases}a\circ purge(\alpha,p), & \text{if } proc(a)\sim>p\\ purge(\alpha,p), & \text{otherwise}\end{cases}$$

函数的目的是保留发生干扰关系的动作，将没有干扰关系的动作清除掉，从而简化动作序列。

3.2 信任链传递形式化描述及证明

进程主要是用于表示应用程序在内存环境中的执行情况，是系统资源分配的基本单位。进程可信包括进程静态可信和动态执行可信。进程静态可信是通过程序及其所依赖的系统动态链接库的完整性来描述的，动态执行可信则是通过进程执行动作前后的状态变化来描述的。

3.2.1 进程静态可信

木马或病毒一般是通过感染可执行文件或替换系统的动

态链接库来达到传播和破坏的目的。因此可以通过度量静态程序及系统动态链接库的完整性来保证进程静态可信。

定义 4 当满足如下条件时，进程 p 是静态可信的。

$\mathrm{int}(p,\{p\})=1$

定理 1 进程 p 可执行，当仅且当其是静态可信的。

证明：(反证法)设某一个被允许执行的进程 p 不是静态可信的，即 $\mathrm{int}(p,\{p\})=0$，则该应用不能被执行，这与假设矛盾。

3.2.2 进程动态可信

进程静态可信只是保证了进程在执行初态可信，但在执行过程中会与其它进程交互，若交互的进程为不可信进程，则其可信状态就可能发生变化。因此，为了保证进程执行过程中的可信，提出进程动态可信。

定义 5 进程 p 为动态可信的，当满足

$$output(run(s_0,\alpha),a)=output(run(s_0,purge(\alpha,proc(a))),a)$$

称该条件为进程 p 的动态可信条件，该定义说明了可信与结果预期性的符合性。

当系统从初始状态 s_0 开始运行，执行动作序列 $\alpha\in A^*$，达到状态 $run(s_0,\alpha)$。此时，进程 p 发出动作 a，从进程 p 的角度观察系统在执行该动作后的结果。若执行动作 a 后，从进程 p 的角度观察能够区分出动作序列 α 和 $purge(\alpha,proc(a))$，则说明存在潜在的进程 $q(q\not\sim proc(a))$干扰了进程 p $(p=proc(a))$，即进程 p 的运行过程不可信。

定义 6 若 $\forall p\in P$，都有一个关于系统状态的观察等价关系 $s\overset{p}{=}t$，则称系统满足观察等价性质。即从进程 p 的角度观察，系统状态 s 和 t 相同。

定义 7 当进程满足下列条件时，称其满足结果隔离性质：

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

其中，$p=proc(a)$。结果隔离性表明，若存在两个状态对于进程 p 是观察等价的，那么进程在这两个状态下执行相同的单个动作将产生相同的输出结果。

定理 2 若进程 p 在具有观察等价性质的系统中具有结果隔离性，并且满足以下条件，则进程运行可信。

$$run(s_0,\alpha)\overset{p}{=}run(s_0,purge(\alpha,p)) \tag{1}$$

证明：设 $p=proc(a)$，代入式(1)中得

$$run(s_0,\alpha)\overset{proc(a)}{=}run(s_0,purge(\alpha,proc(a)))$$

进程 p 满足结果隔离性，则有

$$output(run(s_0,\alpha),a)=output(run(s_0,purge(\alpha,proc(a))),a)$$

从而进程 p 动态可信。

定义 8 称一个进程满足单步隔离性质，当 $s\overset{p}{=}t\Rightarrow step(s,a)\overset{p}{=}step(t,a)$，其中，$q=proc(a)$。

单步隔离性表明，当两个状态对进程 p 是观察等价的，则在此两个状态下，执行相同的单个动作系统的状态不会发生改变。

定义 9 称一个进程满足局部无干扰隔离性质，当 $proc(a)\not\sim p\Rightarrow s\overset{p}{=}step(s,a)$

局部无干扰隔离性表明，若发出动作 a 进程不干扰进程 p，则从进程 p 的角度来看，系统状态和执行动作 a 之前的状态是观察等价的。

定理 3(进程动态可信定理) 若进程满足结果隔离性、单步隔离性和局部无干扰隔离性，则该进程是动态可信的。即若

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

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

$$proc(a)\not\sim p\Rightarrow s\overset{p}{=}step(s,a)$$

由定理 2 可知，即证下式成立。

$$s\overset{p}{=}t\Rightarrow run(s,\alpha)\overset{p}{=}run(t,purge(\alpha,p)) \tag{2}$$

证明：对动作序列 α 的长度做归纳。

当 $\alpha=\Lambda$ 时，式(2)成立。

假设 α 的长度为 n 时，式(2)成立，考虑当长度为 $n+1$ 时，记 $\alpha'=a\circ\alpha$，则式(2)左边有

$$run(s,a\circ\alpha)=run(step(s,a),\alpha) \tag{3}$$

对于 $run(t,purge(a\circ\alpha,p))$，下面分两种情况进行讨论：

1) $proc(a)\sim\!\!>p$

由 $purge$ 函数的定义得

$$run(t,purge(a\circ\alpha,p))=run(t,a\circ purge(\alpha,p))$$

由局部隔离性可得

$$run(t,purge(a\circ\alpha,p))=run(step(t,a),clear(\alpha,p)) \tag{4}$$

此外，由于进程满足单步隔离性，则有

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

根据归纳假设 α 的长度为 n 时，式(2)成立，有

$$run(step(s,a),\alpha)\overset{p}{=}run(step(t,a),purge(\alpha,p)) \tag{5}$$

根据式(3)－式(5)可得

$$run(s,a\circ\alpha)\overset{p}{=}run(t,purge(a\circ\alpha),p)$$

2) $proc(a)\not\sim p$

由 $purge$ 函数定义得

$$run(t,purge(a\circ\alpha,p))=run(t,purge(\alpha,p)) \tag{6}$$

由进程满足无干扰隔离性可得

$$s\overset{p}{=}step(s,a)$$

根据等价关系的性质及 $s\overset{p}{=}t$，有

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

根据归纳假设 α 的长度为 n 时，式(2)成立，有

$$run(step(s,a),\alpha)\overset{p}{=}run(t,purge(\alpha,p)) \tag{7}$$

由式(3)、式(6)、式(7)得

$$run(s,a\circ\alpha)\overset{p}{=}run(t,purge(a\circ\alpha))$$

综上可知，当动作序列长度为 $n+1$ 时，式(2)成立，所以式(2)对任意长度的动作序列都成立。

令 $s=t=s_0$，由式(2)可得

$$run(s_0,\alpha)\overset{p}{=}run(s_0,purge(\alpha,p))$$

由于进程 p 满足结果隔离性质，根据定理 2 可得进程 p 是动态可信的，证毕。

3.2.3 系统运行可信

系统的运行过程可以看作是一系列状态的转移，这些状态的变化是在执行动作的基础上构成的，因此，在任意时刻动作、发出动作的进程和一个系统状态是相关联的。

定义 10（可信状态定义） 1）$\varphi \in S$ 为可信状态（其中 φ 表示空状态）；2）$s=run(\varphi,\alpha)\in S$ 为可信状态，当且仅当动作序列 α 均为合法、可预测的动作，即发出此动作序列的进程均可信。

定理 4（系统运行可信定理） 对于系统 M，当满足如下 4 个条件时：

（1）M 从可信根 s_0 开始运行；

（2）系统中所有处于非运行状态的进程满足 $int(p,\{p\})=1$，即进程静态可信；

（3）M 中的 $\forall p\in P$ 均满足单步隔离性和输出隔离性；

（4）进程满足可信验证，$trust(p)=Trust$。

终端系统 M 是可信系统。

证明：由条件（1）可知，系统从可信状态 s_0 开始运行，满足定义 10。假设 $s=(p,\alpha)$，对动作序列长度 α 作归纳法。当长度为 1 时，系统状态为 s_0，由条件（1），s_0 为可信状态。假设当长度为 n 时，s 为可信状态，则当前执行进程 p 满足

$$run(s_0,\alpha)\overset{p}{=}run(s_0,purge(\alpha,p))$$

然后，系统执行动作 a，并进入下一个状态 $s'=(p',\alpha\circ a)$，即要证明

$$run(s_0,\alpha\circ a)\overset{p'}{=}run(s_0,purge(\alpha\circ a,p'))$$

由上式左边得

$$run(s_0,\alpha\circ a)=step(run(s_0,\alpha),a)$$

由右边可得

$$run(s_0,purge(\alpha\circ a,p'))=run(s_0,clear(\alpha,p')\circ a)=step(run(s_0,purge(\alpha,p')),a)$$

即根据单步隔离性质可知，需证明

$$step(run(s_0,\alpha),a)\overset{p}{=}step(run(s_0,purge(\alpha,p')),a)$$

即 $run(s_0,\alpha)\overset{p}{=}run(s_0,purge(\alpha,p'))$。

下面对动作 a 与进程 p 的关系分情况讨论：

（1）$p=proc(a)$。此时，$p'=p$，由归纳假设得证。

（2）$p\neq proc(a)$。即动作 a 由非 p 进程执行，根据条件（4）系统中的任意进程满足可信验证 $trust(p)=Trust$，则根据验证函数定义及归纳假设条件有 $run(s_0,\alpha)\overset{p'}{=}run(s_0,purge(\alpha,p'))$。

因此，综上可得，假设成立，证毕。

3.2.4 安全接入可信

若不可信终端接入到安全域中，容易导致恶意代码流入，从而导致终端系统不再安全。因此，必须对接入安全域中的节点进行可信度量。

定义 11 系统在安全状态下有请求操作 $rq(t_j,net_i,c)$，则对请求操作的处理如下：

（1）$D(rq,s)=no$，若 $\exists p\in P$，有 $int(p,\{p\})=0$；

（2）$D(rq,s)=yes$，并构造 $T^*=T\cup t_j$，若对于 $\forall p\in P$，都有 $\mathrm{int}(p,\{p\})=1$。

由完整性判定函数 $int:P\times D^*\rightarrow N$ 检查终端进程及加载的动态链接库的完整性，确保接入安全域的终端未被恶意篡改。

定义 12 系统在安全状态下有请求操作 $rq(t_j,net_i,c)$，用户为 $u\in U$，则对请求操作的处理如下：

（1）$D(rq,s)=no$，若 $u\in U_u$；

（2）$D(rq,s)=yes$，并构造 $T^*=T\cup t_j$，若 $u\in U_t$。

即只有可信用户才能够接入到安全域中。

定理 5（安全接入可信定理） 系统状态为可信状态，当网络接入终端满足如下条件时：1）接入终端完整性度量通过；2）用户可信性认证通过，则允许终端接入安全域。

证明：即证明当终端接入动作完成之后，仍然保持了系统状态的安全性。

下面分两种情况讨论：

（1）若 $D(rq,s)=no$，系统状态没有发生改变，因此保持了安全状态。

（2）若 $D(rq,s)=yes$，系统状态 $R\times V\rightarrow D\times V^*$，其中 $V^*-V=T^*-T=(t_j)$。

即完整性度量和用户可信性认证不会破坏系统的安全状态，得证。

4 相关工作比较

本文从进程静态完整性度量、进程动态无干扰及系统安全接入 3 个方面出发，提出了一个基于无干扰理论的信任链传递模型。与已有文献中的模型[2,3,5,8,13]相比，本文的信任链传递模型具有如下优点：

（1）模型从广义的进程定义出发，将系统状态的变化归结为进程动作的执行过程，能够具体地与终端系统相对应，相对于系统状态，进程状态更容易理解，且更加实体化。

（2）模型从可信根出发，保证系统初始运行状态可信。从进程静态完整性度量，保证了进程执行之前的完整性。借鉴无干扰理论，研究了进程执行过程中的结果隔离性质、单步隔离性质和无干扰隔离性质，保证了进程之间的干扰符合相应的安全规则。最后将终端系统的可信扩展到网络中，通过对接入终端的可信验证，保证了整个安全域的可信。

结束语 本文提出了一个新的基于无干扰理论的信任链传递模型。系统从可信根出发，通过对构成系统运行的动态库完整性度量，保证了进程静态可信；借鉴无干扰理论，研究进程执行过程可信，并构建系统可信模型，最后通过对接入终端的可信度量，将信任链从终端传递到网络环境中，保证了整个网络环境的可信。

参 考 文 献

[1] Shen Chang-xiang, Zhang Huang-guo, Feng Deng-guo, et al. The summary of information security[J]. Science in China Press, 2007, 37(2): 129-150(in Chinese)
沈昌祥，张焕国，冯登国，等. 信息安全综述[J]. 中国科学 E 辑：信息科学，2007，37(2)：129-150

[2] Zhang Xing, Chen You-lei, Shen Chang-xiang. A non-interference Trusted model based on process. Journal on Communications, 2009, 30(3): 6-11(in Chinese)
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)：6-11

[3] Zhao jia, Shen Chang-xiang, Liu Ji-qiang, et al. A noninterference-based trusted chain model [J]. Journal of Computer Research and Development, 2008, 45(6): 974-980(in Chinese)
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

（下转第 181 页）

除了 WT-Logic 以外，所有语言均不具有表达谓词权重、可选谓词、多方投票、执行顺序等概念。WT-Logic 也具有推理的确定性，因而在表达工作流授权、多方授权等信息安全领域，WT-Logic 确实具有独特的优势。

结束语　本文针对 Datalog 语言缺乏表达事务和多方决策模型的能力，对 Datalog 进行了扩展，提出了带权重谓词的可选事务逻辑 WT-Logic，并对 WT-Logic 的语法、语义进行了说明，解释了其评价方法。最后，对 WT-Logic 语言在工作流授权和多方投票机制中的应用进行了描述和举例，说明了 WT-Logic 的表达力和可应用性。

下一步，将对 WT-Logic 在社交网络多方授权模型中的应用进行具体研究。

参 考 文 献

[1] Vera Z M, Julia S, Serge A, et al. Collaborative Access Control in WebdamLog[C]// Proceeding of the ACM Sigmod Conference on Data Management, 2015. Melbourne, Australia, 2015: 197-211

[2] Liu X, Alechina N, Logan B. Expressing User Access Authorization Exceptions in Conventional Role-Based Access Control[C]// International Conference on Information Security Practice and Experience. 2013: 233-247

[3] Galland A. Distributed data management with access control: social Networks and Data of the Web[J]. Nature Communications, 2011, 5: 4864

[4] Zhong Yong, Qin Xiao-lin, Liu Feng-yu. Obligation Authorization Model and Its Implementation Frame-work for DRM[J]. Journal of Software, 2010, 21(8): 2059-2069(in Chinese)
钟勇，秦小麟，刘凤玉. 一种面向 DRM 的责任授权模型及其实施框架. 软件学报，2010，21(8)：2059-2069

[5] Zhong Yong, Zhang Hong, Liu Feng-yu, et al. A Digital Rights Management Mechanism and Implementation Based on Logic Framework[J]. Journal of Computer Research and Development, 2010, 47(2): 223-230(in Chinese)
钟勇，张宏，刘凤玉，等. 一种基于逻辑框架的数字版权管理机制和实现[J]. 计算机研究与发展，2010，47(2)：223-230

[6] Hu H, Ahn G J, Jorgensen J. Multiparty Access Control for Online Social Networks: Model and Mechanisms[J]. IEEE Transactions on Knowledge and Data Engineering, 2013, 25(7): 1614-1627

[7] Amrutha P, Sathiyaraj R. Privacy Management of Multi User Environment in Online Social Networks (OSNs)[J]. GJCST-E: Network, Web & Security, 2013, 13(10)

[8] Basseda R, Kifer M. Planning with Regression Analysis in Transaction Logic. Web Reasoning and Rule Systems[M]. Springer International Publishing, 2015: 45-60

[9] Basseda R, Kifer M. State Space Planning Using Transaction Logic[M]. Practical Aspects of Declarative Languages, Springer International Publishing, 2015: 17-33

[10] Montesi D, Bertino E, Martelli M. Transactions and updates in deductive databases[J]. IEEE Trans. Knowl. Data Eng., 1997, 9(5): 784-797

[11] Bertino E, Catania B, Gori R. Active-U-Datalog: integrating active rules in a logical update language[C]// Proc. of International Seminar on Logic Databases and the Meaning of Change, Schloss Dagstuhl. Germany, 1998: 107-133

[12] Tom J A, Bas K, Frank N, et al. Datalog Queries Distributing over Components[C]// Proc. of 18th International Conference on Database Theory (ICDT 2015). Dagstuhl, Germany, 2015: 308-323

[13] Radhakrishnan D. Dynamics of Belief: Horn Knowledge Base and Database Updates[C]// Proc. of 38th Annual German Conference on AI(KI 2015), LNCS 9324. Dresden, Germany, 2015: 341-348

(上接第 144 页)

[4] Rushby J. Noninterference, transitivity, and channel-control security policies[M]. SRI International, Computer Science Laboratory, 1992

[5] Liu Wei-peng, Zhang Xing. Research of duality and multi-level security model based on intransitive noninterference theory[J]. Journal on Communications, 2009, 30(2): 52-58(in Chinese)
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

[6] Chen Ju, Tan Liang. Trusted Terminal Model Based on Process Protection[J]. Computer Science, 2011, 38(4): 115-117(in Chinese)
陈菊，谭良. 一个基于进程保护的可信终端模型[J]. 计算机科学，2011，38(4)：115-117

[7] Xu Fu. Intransitive Noninterference Trusted Model Supporting Process Codes Modification[J]. Computer Engineering, 2013, 39(11): 150-153, 168(in Chinese)
徐甫. 支持进程代码修改的非传递无干扰可信模型[J]. 计算机工程，2013，39(11)：150-153，168

[8] Qin Xi, Chang Chao-wen, Shen Chang-xiang, et al. Research on Trusted Terminal Computer Model Tolerating Untrusted Componenets[J]. Chinese Journal of Electronics, 2011, 39(4): 934-939(in Chinese)
秦晰，常朝稳，沈昌祥，等. 容忍非信任组件的可信终端模型研究[J]. 电子学报，2011，39(4)：934-939

[9] Goguen J A, Meseguer J. EQLOG: Equality, types, and generic Modules for logic programming[M]. Functional and Logic Programming. DeGroot D, Lindstrom G. eds., Springer-Verlag, 1986

[10] 梁元. 基于云计算环境下的可信平台设计[D]. 成都：电子科技大学，2013

[11] Zhu C, Dai X. Model of trust management based on finite state machine[C]// 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec). IEEE, 2012: 161-164

[12] Si Li-min, Cai Mian, Chen Yin-jing, et al. Research of a Trust Chain Transfer Model[J]. Computer Science, 2011, 38(9): 79-81(in Chinese)
司丽敏，蔡勉，陈银镜，等. 一种信任链传递模型研究[J]. 计算机科学，2011，38(9)：79-81

[13] Zhang Xing, Huang Qiang, Shen Chang-xiang. A Formal Method Based on Noninterference for Analyzing Trust Chain of Trusted Computing Platform[J]. Chinese Journal of Computers, 2010, 33(1): 74-81(in Chinese)
张兴，黄强，沈昌祥. 一种基于无干扰模型的信任链传递分析方法[J]. 计算机学报，2010，33(1)：74-81