基于结构化操作语义的安全协议分析框架研究 *)

高三海　董荣胜　钱俊彦

(桂林电子科技大学计算机系　桂林 541004)

摘　要　操作语义模型是一种用来分析安全协议的新模型，它以操作语义学为基础，结合了多种协议分析模型的优点，能直接分析多个协议的组合问题。本文在对安全协议操作语义模型进行研究的基础上，构建了一个基于结构化操作语义的安全协议分析框架，给出了该框架中的协议规格，协议运行，威胁模型和安全性质等形式化定义。最后，以经典的 Needham Schroeder Lowe 协议为例，用该分析框架分析了其机密性和认证性。

关键词　安全协议，形式化方法，分析框架，操作语义

A New Framework Based on Structural Operational Semantics for Analyzing Security Protocol

GAO San-Hai　DONG Rong-Sheng　QIAN Jun-Yan

(Department of Computer Science, Guilin University of Electronic Technology, Guilin 541004)

Abstract　Operational semantics of security protocols is a new model for analyzing security protocols, which is based on operational semantics and combines many merits of the current models such as strand spaces model. Further Characteristics of the model is a straightforward handing of parallel of multiple protocols. In this paper, we first study the operational semantics of security protocols, then, based on which a new framework for security protocol is constructed. In this new framework, protocol specification, protocol run, threat model and security requirements are formally defined. Finally, we validate our framework by analyzing the Needham- Schroeder-Lowe protocol.

Keywords　Security protocol, Formal method, Framework, Operational semantics

1　引言

近年来，安全协议的形式化方法已有统一的趋势[1]。1997 年，Fabrega 等人建立了串空间模型[2]，该模型将定理证明和协议迹结合起来对安全协议进行分析；2003 年，Cas Cremers 等人提出了操作语义模型[3,4]，该模型将以往多种模型和技术的优点融为一体，并以语义学为基础，将安全协议的性质以角色事件的形式统一定义在协议的角色规格中，把协议看成一系列角色的集合，通过在一个协议中加入多个协议的角色来分析协议的组合问题[5]。

在对安全协议的操作语义模型进行研究的基础上，本文构建了一个基于结构化操作语义的安全协议分析框架，给出了协议规格，协议运行，入侵者威胁模型和安全性质的形式化定义。

2　基于操作语义模型的协议分析框架

基于结构化操作语义的分析框架包含 4 个部分，如图 1 所示。

协议规格
协议运行
威胁模型
安全性质

图 1　安全协议分析框架

2.1　协议规格

协议规格是一个静态的概念，由协议的参与者的角色规格构成，描述了各角色的初始知识和行为，包含角色项，角色事件的关系等内容。

2.1.1　角色项

定义 1　基本消息项包括变量、常量、角色和函数，其中：(1)变量集由 Var 表示，$V,W,X,V,Z\in Var$；(2)常量集由 $Const$ 表示，$ni,nr,seccionkey\in Const$；(3)角色集由 $Role$ 表示，$i,r,s\in Role$；(4)函数集由 $Func$ 表示，$sk,pk,k,hash\in Func$。

定义 2　角色项由基本消息项和组合项和加密项构成，即，

$$Role\ Term::=Var\mid Role\mid Const\mid Func(Role\ Term^{*})\mid (Role\ Term,Rle\ Tmer)\mid\{\mid RoleTerm\mid\}_{RoleTerm}$$

定义 3　角色事件是由发送事件、读事件以及声明事件等构成的序列，即，

$$RoleEvent=\{send_l(R,R',t),read_l(R',R,t),claim_l(R,c,rt)\mid l\in Lable,R,R'\in Role,rt\in RoleTerm,c\in Claim\}$$

其中：(1)$Lable$ 表示标记的集合，$Claim$ 表示所有安全性质的集合；(2)$send_l(R,R',t)$是发送事件，表示角色 R 将消息 t 发送给角色 R'；(3)$read_l$ 是(R',R,t)读事件，表示角色 R'接受来自角色 R 的事件 rt；(4)$claim_l(R,c,rt)$是声明事件，表示角色 R 的角色项 rt 满足安全性质 c。

定义 4　每个角色事件有唯一的角色生成，即，

$$role:RoleEvent\rightarrow Role$$

*) 本文得到国家自然科学基金项目(60663005)和广西自然科学基金项目(0542052)的资助。**高三海**　硕士研究生，主要研究方向：形式化技术，协议工程；**董荣胜**　教授，主要研究领域：形式化技术，协议工程；**钱俊彦**　副教授，主要研究领域：程序验证，形式化技术。

定义 5 子项操作"⊑"用于表示一个项的子项，即，

$rt \sqsubseteq rt$

$rt1 \sqsubseteq (rt1, rt2), rt2 \sqsubseteq (rt1, rt2)$

$rt1 \sqsubseteq \{|rt1|\}_{rt2}, rt2 \sqsubseteq \{|rt1|\}_{rt2}$

定义 6 子项中不含变量的角色项构成的集合称作角色知识，即，

$RoleKnow = p(\{rt \in RoleTerm \mid \forall rt' : rt' \sqsubseteq rt \Rightarrow rt' \notin Var\})$

定义 7 角色规格包含角色的初始知识和一系列角色事件构成的序列，即，

$RoleSpec = RoleKnow \times RoleEvent^*$

定义 8(协议规格) 协议的规格将角色映射成角色规格，即

$ProtSpec = Role \rightarrow RoleSpec$

在协议规格 P 中，用 $MP^P(R)$ 表示角色 R 初始知识。若 P 是确定，则 $MR^P(R)$ 可简写成 MR。

2.1.2 角色事件的关系

定义 9(角色事件的偏序关系 $\prec_R$) 若 $\varepsilon 1, \varepsilon 2 \in RoleEvent$ 且 $role(\varepsilon 1) role(\varepsilon 2) = R$，则

$\varepsilon 1 \prec_R \varepsilon 2 \vee \varepsilon 1 = \varepsilon 2 \vee \varepsilon 1 \succ_R \varepsilon 2$

定义 10(角色事件的通讯关系 $\twoheadrightarrow$) 若事件 $\varepsilon 1, \varepsilon 2$ 是两个通讯的事件，且 $m1, m2 \in Role \times Role \times RoleTerm$，$l \in Label$，则

$\varepsilon 1 \twoheadrightarrow \varepsilon 2 \Leftrightarrow \exists l, m1, m2 : \varepsilon 1 = send_l(m1) \wedge \varepsilon 2 = read_l(m2)$

2.1.3 静态的规则

协议规格由协议的参与者的角色规格构成，而合法的角色规格则需要一些静态的规则来保证。

定义 11(知识推理操作 $\vdash$) 若 M 表示一个角色的知识，$rt, rt1, rt2, k \in RoleTerm$，则

$rt \in M \Rightarrow M \vdash rt$

$M \vdash rt_1 \wedge M \vdash rt_2 \Rightarrow M \vdash (rt_1, rt_2)$

$M \vdash rt \wedge M \vdash k \Rightarrow M \vdash \{|rt|\}_k$

$M \vdash (rt1, rt2) \Rightarrow M \vdash rt1 \wedge M \vdash rt2$

$M \vdash \{|rt|\}_k \wedge M \vdash k^{-1} \Rightarrow M \vdash rt$

定义 12(可读谓词 RD) 若 M 表示角色的知识且 $rt \in RoleTerm$，则谓词 RD 定义如下：

$RD(M, rt) =$

$$\begin{cases} True & \text{if } rt \in Var \\ RD(M \cup \{rt2\}, rt1) \wedge RD(M \cup \{rt1\}, rt2) & \text{if } rt \equiv (rt1, rt2) \\ (M \vdash \{|rt1|\}_{rt2}) \vee (M \vdash rt2^{-1} \wedge RD(M, rt1)) & \text{if } rt \equiv \{|rt1|\}_{rt2} \\ M \vdash rt & \text{otherwise} \end{cases}$$

谓词 RD 用于判断角色项 rt 能否当作一个具有知识 M 的主体的读事件的消息模式。

定义 13(良构谓词 WF) 若 M 表示角色知识，S 和 S' 表示角色事件序列且 $R \in RoleTerm$，则谓词定义为 $WF : Role \times RoleSpec$ 定义为

$WF(R, (M, S)) =$

$$\begin{cases} True & \text{if } s \equiv \varepsilon \\ M \vdash (R', R) \wedge RD(M(rt) \wedge WF(R, (M \cup \{rt\}, s')) & \text{if } s \equiv rad_l(R', R, rt) \cdot s' \\ M \vdash (R, R', rt) \wedge WF(R, (M, s')) & \text{if } s \equiv rad_l(R', R, rt) \cdot s' \\ M \vdash R \wedge WF(R, (M, s')) & \text{if } s \equiv rad_l(R', R, rt) \cdot s' \\ False & \text{otherwise} \end{cases}$$

谓词 WF 用于判断协议规格中的角色的定义是否满足规范化要求。

2.2 协议运行

协议运行是一个动态的概念，它是协议规格的实例，包含运行项，运行事件和网络迁移规则等内容。

2.2.1 运行项

定义 14 运行项由常量、协议参与主体、函数项、组合项、加密项以及入侵者构造的项构成，即，

$RunTerm ::= Const \# RID \mid Agent \mid Func(RunTerm^*) \mid (RunTerm, RunTerm) \mid \{|RunTerm|\}_{RunTerm} \mid InturderConst$

其中，RID 表示运行项标记的集合，$Agent$ 表示执行协议的主体的集合，包括信任的主体集合 $Agent_T$ 和非信任主体集合 $Agent_u$。

定义 15(实例化过程 $Inst$) 角色项转化成运行项的过程称为实例化，即

$Inst = RID \times (Rold \rightarrow Agent) \times (Var \rightarrow RunTerm)$

定义 16(角色项的实例) 若 $inst = (\theta, \rho, \sigma) \in Inst$，$f \in Func, r1, rt_1, \cdots, rt_n \in RoleTerm$，且 $roles(rt) \subseteq dom(\rho)$，$vars(rt) \subseteq dom(\sigma)$ 则角色实例函数 $\langle inst \rangle : RoleTurm \rightarrow RunTerm$，定义为：

$\langle inst \rangle (rt) =$

$$\begin{cases} \rho(R) & \text{if } rt \equiv R \in Rold \\ f(\langle inst \rangle (rt1), \cdots, \langle inst \rangle (rt1)) & \text{if } rt \equiv f(rt1, \cdots, rtn) \\ c \# \theta & \text{if } rt \equiv \mid \in Const \\ \sigma(v) & \text{if } rt \equiv v \in Var \\ (\langle inst \rangle (rt1), \langle inst \rangle (rt2)) & \text{if } rt \equiv (rt1, rt2) \\ \{|\langle inst \rangle (rt1)|\}_{\langle inst \rangle (rt2)} & \text{if } rt \equiv \{|rt1|\}_{rt2} \end{cases}$$

其中，函数 $roles : RoleTerm \rightarrow \mathcal{P}(Role)$ 表示角色项中的角色，函数 $vas : RoleTerm \rightarrow \mathcal{P}(var)$ 表示角色项中的变量。

定义 17(良类型替换 $Welltyped$) 若在一个实例中，所有变量都实例化为运行项，则该替换为良类型，即，

$Welltyped(\sigma) = \forall v \in dom(v) : \sigma(v) \in type(v)$

其中，函数 $type : Var \rightarrow P(RunTerm)$，表示一个变量可实例化为运行项的集合。

定义 18(运行) 运行是角色事件序列的实例，即

$Run = Inst \times RoleEvent^*$

定义 19(运行事件) 运行事件是一个二元序偶 $(inst, \varepsilon)$，其中 $inst \in Inst, \varepsilon \in RoleEvent$，运行事件的集合记为 $RunEvent$。

2.2.2 网络迁移系统

协议运行在开发的网络环境中，包含多个运行，运行间的通讯是异步的，并采用输出缓存和输入缓存进行路由通讯。每个缓存包含多个消息项，即 $Buffer = \mathcal{M}(MSG)$，而消息则由发送者，接受者和运行项构成。此外，完整的网络模型还包括 2.3 节中定义的入侵者模型。

定义 20(网络状态) 网络状态由入侵者知识 $\mathcal{P}(RunTerm)$、输出缓存、输入缓存和等待执行的运行 $\mathcal{P}(Run)$ 构成，即，

$State = \mathcal{P}(RunTerm) \times Buffer \times Buffer \times \mathcal{P}(Run)$

其中 $Buffer = \mathcal{M}(MSG)$ 且 $MSG = Agent \times Agent \times RunTerm$，若缓存中的消息与读事件中所规格的消息模式匹配，则该消息能被协议参与主体接受。

定义 21(Math) 若 $inst, inst' \in inst$ 且 $pt, m \in RoleTerm$，则谓词 $Match$ 定义如下：

$Match(inst, pt, m, inst') \Leftrightarrow$

$$inst=(\theta,\rho,\sigma)\wedge inst'=(\theta,\rho,\sigma')$$
$$\wedge\sigma\subseteq\sigma'\wedge dom(\sigma')=dom(\sigma)\cup vars(pt)$$
$$\wedge Welltyped(\sigma')\wedge(rid,\rho\sigma')(pt)=m$$

定义 22(*runsof*) 函数 $funsof:Protocol\to\mathcal{P}(Run)$表示协议 P 创建的运行的集合，即

$runsof(p)=\{((\theta,\rho,\phi),P(R))\mid\theta\in RID\wedge\rho\in roles(P(R))\times Agent\wedge dom(p)\}$

定义 23(*runIDs*) *runIDs* 表示在运行集合 F 中，处于活跃状态的元素的集合，即

$runIDs(F)=\{((\theta,\rho,\sigma),ev)\in F\}$

定义 24（替换） 若 $F\in\mathcal{P}(Run)$，则 $F[x'/x]$表示用 x' 替换 F 中 x。

定义 25（初始网络状态） 若用 S_0 表示初始网络状态，则

$s_0=\langle M_0,\phi,\phi,\phi\rangle$

其中，M_0 表示入侵者的知识。

若 $P\in ProSpec$，则系统基本的起源规则可定义如下：

(1)*Create* 规则。一个新的运行仅能在它的运行标记没有被使用时才可以被构造；

$$\frac{run=((\theta,\rho,s),s)\in runsof(P),\theta\notin runIDs(F)}{\langle m,BS,BR,F\rangle\xrightarrow{create(run)}\langle M,BS,BR,F\cup\{run\}\rangle}$$

(2)*send* 规则。若一个运行执行一个发送事件，则该发送的消息会被添加到输出缓存中，该运行继续执行下一个事件；

$$\frac{run=(inst,send_l(m)\cdot s)\in F}{\langle m,BS,BR,F\rangle\xrightarrow{(inst,send_l(m))}\langle M,BS\cup\{\langle inst\rangle(m)\},BR,F[(inst,s)/run]\rangle}$$

(3)*Read* 规则。该规则决定了输入缓存中的事件何时被执行，它要求在读事件里所有规格的实例模式必须同输入缓存中的消息模式相匹配。当读事件被执行时，输入缓存中的这个消息将被删除，运行将继续执行下一个事件；

$$\frac{run=(inst,read_l(pt)\cdot s)\in F,m\in BR,Match(inst,pt,m,inst')}{\langle M,BS,BR,F\rangle\xrightarrow{(inst',read_l(pt))}\langle M,BS,BR\backslash\{m\},F[(inst',s)/run]\rangle}$$

(4)*Claim* 规则。表示一个使能的声明事件总能被执行；

$$\frac{run=run=(inst,claim_l(R,c,t)\cdot s)\in F}{\langle m,BS,BR,F\rangle\xrightarrow{(inst,claim_l(R,c,t))}\langle M,BS,BR,F[(inst,s)/run]\rangle}$$

(5)*transmit* 规则。表示在没有入侵者情况下，消息从输出缓存到输入缓存的迁移；

$$\frac{m\in BS}{\langle m,BS,BR,F\rangle\xrightarrow{transmit(m)}\langle M,BS\backslash\{m\},BR\cup\{m\},BR\cup\{m\},F\rangle}$$

定义 26（迁移标记） 每个迁移都对应迁移标记集合中的一个元素标记，该集合定义为，

$Transitionlabel::=RunEvent\mid create(Run)\mid Networkrulename$

其中，*Create*(*run*)表示由 *Create* 规则创建运行，而 *Networkrulename* 是系统的一个参数表示，表示网络入侵规则名。

一个协议状态的迁移系统是从网络初始状态 s_0 出发，有限次运用 *Transitionlabel* 中的规则得到的。若用迹(*trace*)来表示协议状态的迁移序列，则所有状态的迁移序列表示为迹的集合(Tr)，即 $Tr=\{a\in Transitional^*\mid s_0\xrightarrow{a}\}$，其中 s_0 表示协议的初始状态，a_i 表示迹 a 的第 i 行为，$|a|$表示序列 a 的长度并用 $s_0\xrightarrow{a}$表示 $\exists_s s\xrightarrow{a}s'$。

2.3 威胁模型

本文将威胁模型分成两种类型，一种是假设网络可部分或全部被入侵者控制的网络威胁模型，另一种是假设入侵者可通过攻击协议的合法参与主体来完成对协议攻击的主体威胁模型。

2.3.1 网络威胁模型

在安全协议的验证领域，Dolev-Yao 入侵者模型是一个通用的一般模型，该模型假设入侵者可以完全控制整个网络，消息可被入侵者截获、修改和伪造。然而，在实际情况中，入侵者可能仅部分地控制整个网络，例如，在无线通讯网络模型中，入侵仅能窃听网络中的消息。因此，本文认为入侵者可部分或全部被入侵者控制并将入侵者模型定义成一系列入侵规则的集合，每个规则定义了入侵者某种能力，其定义如下：

(1) *take* 规则。该规则表示具备偏转(*deflection*)能力的入侵者能够从输出缓存中删除任何一个消息；

$$\frac{m\in BS}{\langle M,BS,BR,F\rangle\xrightarrow{take(m)}\langle M\cup\{M\},BS\backslash\{M\},BR\langle F\rangle}$$

(2) *fake* 规则。该规则表示能从入侵者知识里推理出来消息都能插入到输入缓存中；

$$\frac{M\vdash m}{\langle M,BS,BR,F\rangle\xrightarrow{fake(m)}\langle M,BS,BR\cup\{m\},F\rangle}$$

(3)*eavesdrop* 规则。该规则表示若入侵者具有窃听的能力，则它可以截获在传播过程中的消息；

$$\frac{M\in BS}{\langle M,BS,BR,F\rangle\xrightarrow{eavesdrop(m)}\langle M\cup\{m\},BS\backslash\{m\},BR\cup\{m\},F\rangle}$$

(4) *jam* 规则。该规则表示入侵者可读取被偏转过(*deflected*)的消息，并可将其加入到自己的知识集中；

$$\frac{M\in BS}{\langle M,BS,BR,F\rangle\xrightarrow{jam(m)}\langle M,BS\backslash\{m\},BR,F\rangle}$$

2.3.2 主体威胁模型

入侵者可以通过攻击多个协议主体来完成对协议的攻击。当入侵者成功地攻击了一个协议参与者时，它就可以获取这个主体的所有的角色的知识，并且可以扮演该主体参与协议的运行。

定义 27（非信任主体） 非信任的主体是遭到入侵者成功攻击的主体，通常用 E 来表示且 $E\in Agent_U$。

定义 28（入侵者初始知识） 在协议 P 中，入侵者初始知识由所有非信任主体的知识构成，即，

$$M_0=InstruderConst\cup\bigcup_{\substack{R\in Role\\ \rho\in Role\to Agent\\ \rho(R)\in Agent_U}}\{\langle\_,\rho,\_\rangle(rt)\mid rt\in MR(R)\wedge rt'\hat{o}\ rt:rt'\notin Const\}$$

其中，*InstruderConst* 表示入侵创建的临时值的集合。

定义 29（执行运行事件后的入侵者知识） 若 t 是一个迹，i 是迹的索引号且 $0<i\leqslant|t|$，则 M_i 表示入侵者在执行运行事件 $t_0,t_1,\cdots,t_{i-1}$ 后的知识。

定理 1（执行运行事件后的最大的入侵者知识） 若 t 是一个迹，i 是迹的索引号且 $0<i\leqslant|t|$，则

$$M_i\subseteq\{\langle inst\rangle(m)t_j=(inst,send_l(R1,R2,m))\wedge j\wedge i\wedge j\in N\wedge inst\in Inst\wedge l\in label\wedge R1,R2\in Role\wedge m\in RoleTerm\}$$

证明：根据入侵者规则，只有发送缓存的元素能够改变入侵者的知识集 M，易知，定理 1 得证。

2.4 安全性质

安全协议的性质在本质上也是协议规格的一部分，本文将协议的安全性质以声明事件的形式统一定义在协议规格中。限于篇幅，本文仅给出协议的机密性和同步性(本文采用

文[5]的分类方法，将同步性列为一种强认证性）的形式化定义。

定义 30（机密性） 在协议 p 中，若 $t\in RleTerm$，$claim(R,secrte,t)$ 是机密性声明事件且(1)式对于任何迹 $a\in Tr(p)$ 和 $i\in N$ 都成立，则称协议 p 满足机密性。

$$a_i=((\theta,\rho,\sigma),claim(R,secrte,t))\wedge rng(\rho)\subseteq Agent_T\Rightarrow \forall_{i\leqslant j\leqslant|a|}(\theta,\rho,\sigma)(t)\notin M_j^p \quad (1)$$

认证性是要在协议主体间确定身份，本文首先形式化地定义了两个辅助谓词 *1L-SYNCH* 和 *ML-SYNCH* 的，并在此基础上给出了同步性的形式化定义，限于篇幅，其它形式的认证性的形式化定义可以参考文[5,6]。

定义 31（*1L-SYNCH*） 若 $a\in Tr(p)$，$k\in N$，$l\in Lable$ 和运行的号 θ_1，θ_2，则谓词 *1L-SYNCH* 定义如下：

$$\begin{aligned}&1L\text{-}SYNCH(a,k,l,rid1,rid2)\\&\Leftrightarrow\exists_{i,j\in N,inst1,inst2\in Inst,m1,m2\in MSG\,i<j<k}\\&\quad a_i=(inst_1,send_l(m_1))\wedge runidof(inst_1)=\theta_1\wedge\\&\quad a_j=(inst_2,send_l(m_2))\wedge runidof(inst_2)=\theta_2\wedge\\&\quad inst_1(m_1)=inst_2(m_2)\end{aligned}$$

其中，$runidof$：$Inst\rightarrow Runid$ 表示一个实例的运行号，即 $runidof(\theta,\rho,\sigma)=\theta$。

定义 32（*ML-SYNCH*） 若 $a\in Tr(p)$，$k\in N$，$l\in Label$，则谓词 *ML-SYNCH* 定义如下：

$$\begin{aligned}&ML\text{-}SYNCH(a,k,L,cast)\\&\Leftrightarrow\forall_{l\in L}1L\text{-}SYNCH(a,k,l,cast(sendrole(l)),cast(readrole(l)))\end{aligned}$$

其中，用 $Sendrole(l)$ 来表示出现在事件 $send_l$ 中的角色，用 $readrole(l)$ 表示事件 $read_l$ 中的角色；$Cast$：$Role\rightarrow Runid$，用来表示实例为运行后的运行标记，即 $cast(Role)=\theta$，其中 θ 表示运行标记号。

定义 33（同步性） 在协议 p 中，若 $nisynch\in claim$ 且任何一个迹 $a\in Tr(P)$ 都满足(2)式，则称协议 p 满足同步性（$NI-SYNCH$）。

$$\begin{aligned}&a_i=(\theta,\rho,\sigma,claim_l(r,nisynch))\wedge rng(\rho)\subseteq Agent_T\Rightarrow\\&\exists_{castRale\rightarrow Runial}(cast(r)=\theta\wedge ML\text{-}SYNCH(a,prec(p,l),cast))\end{aligned}\quad (2)$$

其中，$prec(p,cl)=\{l|read_l(\_,\_,\_)\prec claim_d(\_,\_)\}$

3 实例

本节以 *Needham-Schroeder-Lowe*（*NSL*）协议为例，用所构建的协议分析框架，证明 *NSL* 协议满足机密性和认证性。

在证明前，先给出 4 个引理，即，

引理 1 若在协议 p 中，$inst\in Inst$，t 是一个基本项且 t 不是任何已发送消息项的子项，$\langle inst\rangle(t)$ 也不是任何入侵者初始知识的子项，则入侵者不能获取 $\langle inst\rangle(t)$，即

$$\forall_{t'\in msgs(p)}t\subsetneq t'\wedge\forall_{m;M_0\vdash m}\langle inst\rangle(t)\,mp\,\forall_{a\in Trce(p),0\leqslant j\leqslant|a|}M_j^a-\langle inst\rangle(t)$$

其中 $msgs(p)$ 表示协议中发送的角色消息的集合。根据 2.3 节中的网络状态的迁移规则，易证该引理是正确的。

引理 2 若在协议 p 中，$e\,e'\in RoleEvent$，$e'\prec_R e$，$a\in Tr(p)$，$(\theta,\rho,\sigma)\in Inst$，$(0\leqslant i<|a|)$ 且 $a_i=((\theta,\rho,\sigma),e)$，则存在 j，$(0\leqslant j<i)$ 和 $\sigma'\subseteq\sigma$，使得 $a_j=((\theta,\rho,\sigma'),e')$。

根据 2.3 节中的网络状态的迁移规则和谓词 *Match* 的定义，易证该引理是正确的。

引理 3 若在协议 p 中，$a\in Tr(p)$，$(0\leqslant i<|a|)$ 且有 $a_i=(inst,read_l(x,y\{|m|\}_k))$，$M_0\nvdash\langle inst\rangle(\{|m_0|\}_k)M_i^a\nvdash\langle inst\rangle(m)$，则存在 $j<i$ 使 $a_j=(inst',send_l(x',y',m'))$ 且 $(inst)(\{|m|\}_k)\sqsubseteq(inst')(m')$。

该引理基于以下事实，即，若入侵者不能通过加密消息项 m 构造 $\{m\}_k$，则 $\{m\}_k$ 是某个已经发送的消息项的子项，证明略。

引理 4 若在协议 p 中，(θ,ρ,σ)，$(\theta',\rho',\sigma')\in Inst$ $n\in Const$ 且 $(\theta,\rho,\sigma)(n)=(\theta',\rho',\sigma')(n)$，则 $\theta=\theta'$。

该引理基于以下事实，即，若一个常量（如临时值和会话密钥）的两个实例是相等的，那么这两个实例是在同一次运行中被构造出来的，证明略。

根据 2.1 节中的协议规格定义可知 *NSL* 协议的规格如下：

$nsl(i)=(\{i,r,ni,pk(r),pk(i),sk(i)\})$，$nsl(r)=(\{i,r,ni,pk(r),pk(i),sk(i)\})$，

$send_1(i,r,\{\|i,ni\|\}_{pk(r)})$.	$read_1(i,r,\{\|i,W\|\}_{pk(r)})$.
$radl_2(r,i,\{\|ni,V,r\|\}_{pk(i)})$.	$sead_2(ri,\{\|W,nr,r\|\}_{pk(r)})$.
$send_3(i,r,\{\|V\|\}_{PK(r)})$.	$send_4(i,r,\{\|nr\}_{pk(r)})$.
$Claim_4(i,secrent,ni)$.	$Claim_7(i,secrent,nr)$.
$Claim_5(i,secrent,V)$.	$Claim_8(i,secrent,W)$.
$Claim_6(i,nisynch)$.	$Claim_9(i,nisynch)$.

根据 2.3 节中的定义 28，*NSL* 协议的入侵者的初始知识定义如下：

$$M_0=InstruderConst\cup\bigcup_{a\in Agent}\{a,pk(a)\}\cup\bigcup_{a\in Agent_u}\{sk(e)\}$$

根据引理 1，在 *NSL* 协议中任何消息都不含加密密钥，因此，入侵者不可能获知 *NSL* 协议中任何信任主体的密钥。换言之，入侵者从一个加密的消息中获取了基本消息项，则该加密消息一定由非信任的主体加密得到的。限于篇幅，本文仅构造机密性声明事件 $claim_7$ 和同步性（一种强认证性）声明事件 $claim_9$ 的证明，其它的声明事件的证明类似。

(1)机密声明事件 $claim_7$ 的证明

反证法，假设入侵不能获取临时值 nr，机密性声明事件 $claim_7$ 不成立。

首先，设 a 是 *NSL* 协议的一个迹，$r7\in N$ 且 $a_{r7}=((\theta_{r7},\rho_{r7},\sigma_{r7}),claim_7(r,secret,nr))$，$rng(\rho_{r7})\subseteq Agnet_T$。根据假设，若机密性声明事件 $claim_7$ 不成立，则入侵者能够获取临时值 nr，则存在 $k\in N$，使得 $(\theta_{r7},\rho_{r7},\sigma_{r7})(nr)\in M_{k+1}$ 且 $(\theta_{r7},\rho_{r7},\sigma_{r7})(nr)\notin M_k$。根据 2.3 节中的网络状态的迁移规则可知，*send* 规则可以增加入侵者知识。因此，存在 $p<k$ 且 $p\in N$，使得 $a_p=((\theta',\rho',\sigma'),send_1(m))$，$(\theta_{r7},\rho_{r7},\sigma_{r7})(nr)\hat{o}(\theta',\rho',\sigma')(m)$。

在 *NSL* 协议中有三种可能的发送事件，分别用 $l=1,2,3$ 表示，即

当 $l=1$ 时，$a_p=((\theta',\rho',\sigma'),sneld_1(i,r,\{|i,ni|\}_{pk(r)}))$，因为 i 和 ni 都是不同于 nr 的常量，所以入侵者不能通过 $(\theta',\rho',\sigma')(i,r,\{|i,ni|\}_{pk(r)})$ 获取 $(rid_{r7},\theta_{r7},\sigma_{r7})(nr)$，即入侵者不能通过这种方式获取临时值 nr。

当 $l=2$ 时，有 $a_p=((\theta',\rho',\sigma'),send_2(ri,k\{|W,nr,r|\}_{pk(r)}))$，因为 $\rho'(i)$ 是非信任主体，所以入侵者能获取 nr，即 $(\theta_{r7},\rho_{r7},\sigma_{r7})(nr)=(\theta',\rho',\sigma')(W)$ 或 $(\theta_{r7},\rho_{r7},\sigma_{r7})(nr)=(\theta',\rho',\sigma')(nr)$，下面分别讨论这两个等式：

①若 $(\theta_{r7},\rho_{r7},\sigma_{r7})(nr)=(\theta',\rho',\sigma')(W)$ 则 $\langle\theta',\rho',\sigma'\rangle(W)\notin M_p$，根据引理 2 和 3，存在 $i1\in N$，使得 $a_{i1}=((\theta_{i1},\rho_{i1},\sigma_{i1}),sendl_1(i,r\{|i,ni|\}_{pk(i)}))$，进而 $(\theta_{i1},\rho_{i1},\sigma_{i1})(ni)=(\theta',\rho',\sigma')(W)=(\theta_{r7},\rho_{r7},\sigma_{r7})(nr)$。但是，因为 ni 和 nr 是不同的常量，所以这种情况是不可能的。

②若 $(\theta_{r7},\rho_{r7},\sigma_{r7})(nr)=(\theta',\rho',\sigma')(nr)$，则根据引理 4，可知 $\theta_{r7}=\theta'$，因为运行标记是唯一的，则 $\rho_{r7}=\rho'$，从而 $\rho_{r7}(i)$

（下转第 176 页）

2 D'haeseleer P, Liang S, Somogyi R. Genetic network inference: from co-expression clustering to reverse engineering. Bioinformatics, 2000, 16(8):707～726
3 Sharan R, Elkon R, Shamir R. Clustering Analysis and its Application to Gene Expression Data. 2001
4 Han J, Kamber M. Data Mining: Concepts and Techniques. In: The Morgan Kaufmann Series in Data Management Systems, Jim Gray, Series Editor Morgan Kaufmann Publishers, ISBN 1-55860-489-8, August 2000
5 Kohonen T. Self-Organization and Associative Memory. Spring-Verlag, Berlin, 1984
6 Beyer K, Goldstein J, Ramakrishnan R, et al. When is the nearest neighbor meaningful? Lecture Notes in Computer Science, 1999, 1540:217～235
7 Hastie T, Tibshirani R, Eisen M B, et al. Gene shaving' as a method for identifying distinct sets of genes with similar expression patterns. Genome Biology, 2000, 2(1)
8 Efron B, Tibshirani R, Goss V. et al. Microarrays and Their Use in a Comparative Experiment: [Tech. report]. Stanford University, 2000
9 Tang Chun, Zhang Aidong. An Iterative Strategy for Pattern Discovery in Multi-dimensional Data Sets. In: 11 th International Conference on Information and Knowledge Management (CIKM 2002). McLean, VA, November 2002
10 Tang Chun, Zhang Li, Zhang Aidong, Ramanathan M. Interrelated Two-way Clustering: An Unsupervised Approach for Gene Expression Data Analysis. In: Proc. of 2nd IEEE International Symposium on Bioinformatics and Bioengineering. Bethesda, MD. November 2001
11 Jiang Daxin, Tang Chun, Zhang Aidong. Clustering Analysis for Gene Expression Data: A Survey. IEEE Transactions on Knowledge and Data Engineering (TKDE)

(上接第 104 页)

$=\rho'(i)$，这与 $\rho_{r7}(i)$ 是一个信任的主体的假设相矛盾。

当 $l=3$ 时，$a_p=((\theta',\rho',\sigma'),sned_3(i,r,\{|V|\}_{pk(r)}))$，为了从 $(\theta',\rho',\sigma')(i,r,\{|V|\}_{pk(r)})$ 获取 $(\theta_{r7},\rho_{r7},\sigma_{r7})(nr)$，必有 $(rid',\rho',\sigma')(V)=((rid_{r7},\rho_{r7},\sigma_{r7}))(nr)$ 且 $\rho'(r)$ 是一个非信任的主体。根据引理 2，可知存在 $i2$ 使得 $a_{i2}=((\theta',\rho',\sigma'),read_2(r,i,\{|ni,V,r|\}_{pk(i)}))$，又因 $(\theta',\rho',\sigma')(V)\notin M_p$，则据引理 3，存在 $r2$，使得 $a_{r2}=((\theta_{r2},\rho_{r2},\sigma_{r2}),send_2(r,i,\{|W,nr,r|\}_{pk(i)})$，从而 $\rho'=\rho_{r2}(r)$，且 $(\theta_{r2},\rho_{r2},\sigma_{r2})(nr)=(rid',\rho',\sigma')(V)=(rid_{r7},\rho_{r7},\sigma_{r7})(nr)$，根据引理 4，可知 $\theta_{r2}=\theta_{r7},\rho_{r2}=\rho_{r7}$，即 $\rho'(r)=\rho_{r2}(r)=\rho_{r7}(r)$，但因为 $\rho'(r)$ 是一个非信任的主体，$\rho_{r7}(r)$ 而是一个信任的主体，这就得出了矛盾。

上面的证明可以看出，入侵不能获取临时值 nr，故假设不成立，机密声明事件 $claim_7$ 得证。

(2)同步性(一种强认证性)声明事件 $claim_9$ 的证明

证明：设 $a\in Tr(nsl),r9\in N$，且 $(\theta_r,\rho_r,\sigma_{r9})\in Inst,rng(\rho r)\subseteq Agent_T,a_{r9}=((rid_r,\rho_r),claim_9(r,nisynch))$。

因为 $prec(nsl,9)=\{1,2,3\}$，所以为了证明声明事件 $claim_9$ 正确性，必须找一个执行初始角色的运行，该运行中的事件与标记 1,2,3 事件同步。根据引理 2，可知存在 $(0\leqslant r1<r2<r3<r9)$ 和 $\sigma_{r1}\subseteq\sigma_{r2}\subseteq\sigma_{r3}\subseteq\sigma_{r9}$ 使得

$$a_{r1}=((\theta_r,\rho_r,\sigma_{r1}),read_1(i,r,\{|i,W|\}_{pk(r)}))$$
$$a_{r2}=((\theta_r,\rho_r,\sigma_{r2}),sead_2(i,r,\{|W,nr,r|\}_{pk(i)}))$$
$$a_{r3}=((\theta_r,\rho_r,\sigma_{r3}),read_1(i,r,\{|nr|\}_{pk(r)}))$$

从 $claim_7$ 证明过程中可知 nr 满足机密性，因此根据引理 3，存在 i_3 和 $(\theta_i,\rho_i,\sigma_{i3})$，对于 $i_3<r_3$ 和 $a_{i3}=((\theta_i,\rho_i,\sigma_{i3}),sned_3(i,r,\{|V|\})_{pk(r)})\wedge(\theta_r,\rho_r,\sigma_{r3})(nr)=(\theta_i,\rho_i,\sigma_{i3})(V)$，根据引理 2，可知存在 $i1<i2<i3$，使得

$$a_{i1}=((\theta_i,\rho_i,\sigma_{i1}),read_1(i,r,\{|i,ni|\}_{pk(r)}))$$
$$a_{i2}=((\theta_i,\rho_i,\sigma_{i2}),read_2(i,r,\{|ni,V,r|\}_{pk(i)}))$$
$$a_{i3}=((\theta_i,\rho_i,\sigma_{i3}),read_1(i,r,\{|V|\}_{pk(r)}))$$

现在仅需证明运行 θ_i 是运行 θ_r 满足同步性，为此，创建 $r_1<i_2,i_1<r_1$ 并且使它们对应的发送事件和读事件是相互匹配的。

首先，对于 a_{i2}，因为 $(\theta_r,\rho_r,\sigma_{r3})(nr)$ 是机密的，$(\theta_i,\rho_i,\sigma_{i2})(V)$ 也是机密，所以根据引理 3，存在 $r2'<i2$，使得 $a_{r2'}=((\theta_r,\rho_r,\sigma_{r2}))$，$send_2(r,i,\{|W,nr,r|\}_{pk(i)})$ 从而 $(\theta_i,\rho_i,\sigma_{i2})(\{|ni,V,r|\}_{pk(i)})=(\theta_{r'},\rho_{r'},\sigma_{r2'})(\{|W,nr,r|\}_{pk(i)})$，进而可知 $(\theta_r,\rho_r,\sigma_{r3})(nr)=(\theta_i,\rho_i,\sigma_{i3})(V)=(\theta_{r'},\rho_{r'},\sigma_{r2'})(nr)=(\theta_i,\rho_i,\sigma_{i3})(V)=(\theta_{r'},\rho_{r'},\sigma_{r2'})(nr)$，所以，根据引理 4，可知 $\theta_r=\theta_{r'}$，而且 $r2=r2'$，即证明了事件 $ai2$ 和 $ar2$ 是同步的。

其次，对于 $ar1$，因为 $(\theta_r,\rho_r,\sigma_{r1})(W)$ 是秘密的，所以根据引理 3，存在 $i1'<r1$，使得 $a_{i1'}=((\theta_{i'},\rho_{i'},\sigma_{i1'}))$，$send_1(i,r,\{|i,ni|\}_{pk(r)})$ 和 $(\theta_r,\rho_r,\sigma_{r1}))(\{|i,W|\}_{pk(r)})=(\theta_{i'},\rho_{i'},\sigma_{i1'})(\{|i,ni|\}_{pk(r)})$，则两个对应的 a_{i2} 和 a_{r2} 满足 $(\theta_i,\rho_i,\sigma_{i2})(ni)=(\theta_r,\rho_r,\sigma_{r2})(W)=(\theta_{i'},\rho_{i'},\sigma_{i1'})(ni)$。从而根据引理 4 可知 θ_i 和 $\theta_{i'}$ 是相等的，即证明了事件 a_{r1} 和 a_{i1} 是同步的，因此，同步性声明事件 $claim_9$ 得到证明。

结束语 本文在对操作语义模型进行研究的基础上，构建了基于结构化操作语义的安全协议分析框架，该框架不仅可以分析具体的安全协议，更重要的是，它为安全协议的形式化分析提供了一个统一的语义模型。在后继的工作中我们将用此框架进一步规范更多的安全性质并研究基于此框架的自动推理技术。

参 考 文 献

1 Meadows C. Formal methods for cryptographic protocol analysis: Emerging issues and trends. IEEE Journal on Selected Areas in Communications, 2003, 21(1):44～54
2 Thayer F J, Herzog J C, Guttman J D. Strand spaces: Why is a security protocol correct? In: Proc. of the 1998 IEEE Symposium on Security and Privacy Los Alamitos: IEEE Computer Society Press, 1998. 160～171
3 Cremers C J F, Mauw S. Operational Semantics of security protocols. Scenarios: Models, Transformations and Tools. In: International Workshop Models, Dagstuhl Castle, Germany: Springer, 2005. 66～89
4 Cremers C J F, Mauw S, Vink E P. Defining Authentication in A Trace Model. In: Proc. of the First International Workshop on Formal Aspects in Security and Trust, Pisa, Italy, 2003. 131～145
5 Cremers C J F. Compositionality of Security Protocols: A Research Agenda. Vodca 2004 ENTCS, 2006, 142(3):99～110
6 Lowe G. A Hierarchy of Authentication Specifications. In: Proceedings of the 10th IEEE Computer Security Foundations Workshop, IEEE Computer Society Press, 1997. 31～43