电子商务协议形式化分析的一种新方法

郭云川　古天龙　董荣胜　蔡国永
(桂林电子工业学院计算机系　桂林541004)

摘　要　在电子商务领域,其协议的安全性和原子性是两个重要的问题,有必要对不断发展的协议进行分析和检验。文[3]提出了逻辑分析和进程演算相结合的技术,但这种技术在分析电子商务协议的安全性和原子性时存在一些局限性,因而本文提出了一种新的分析和检验原子性的方法。

关键词　电子商务协议,安全性,原子性,形式化方法

A New Approach for the Formal Analysis of Electronic Commerce Protocols

GUO Yun-Chuan　GU Tian-Long　DONG Rong-Sheng　CAI Guo-Yong
(School of Computer Science, Guilin University of Electronic Technology, Guilin 541004)

Abstract　In e-commerce field, security and atomicity of e-commerce protocols are two important issues. It is a necessary step to analyze and verify them in developing protocol. The technique of logic analysis combined with process calculi is proposed in [3]. However, the technique has some limitations in analyzing security and atomicity of e-commerce. In the paper, a novel technique for analyzing and verifying atomicity is presented.

Keywords　E-commerce protocol, Security, Atomicity, Formal methods

1　引言

现在电子商务得到了迅猛发展,安全性和原子性是电子商务协议需要满足的两个非常重要的性质。但电子商务协议的设计是非常复杂且容易出错,一个貌似良好的协议可能存在重大的安全漏洞。因此,在协议设计后期采取一定的方法对协议进行分析检验是非常必要的。目前分析电子商务协议的方法主要有两种:攻击检验方法和形式化分析方法[1],其中形式化分析方法在学术界得到了广泛的应用,目前正逐渐应用到工程界。

为了用形式化方法对电子商务协议进行分析,首先要对协议及其必须满足的性质进行形式化描述,然后用某种方法(比如,逻辑分析、模型检测等)对协议进行分析。逻辑方法,比如BAN逻辑[2],可以对交易主体的消息和信仰建模,但它不能对交易主体的行为建模。进程演算[4]可以对交易主体的行为建模,但是其建模能力以及逻辑推理能力都比较弱。文[3]中将逻辑分析和进程演算有机地结合了起来。但该方法还不能对协议的认证性进行分析,不能对协议的原子性[5]进行分析。

本文讨论了文[3]的局陷,分析了原子性的特点,提出了一种分析电子商务协议的新方法,这种方法可以对协议的原子性进行形式化描述,进而分析协议是否符合原子性要求,从而克服了文[3]不能分析原子性的局限。限于篇幅,我们不再详细介绍文[3]中的方法,同时,为了方便阅读,本文的记法与文[3]相同。

2　文[3]工作的局限

文[3]提出了7条规则,其中最重要的规则有3条:Knows规则,Extract规则和Construct规则。使用这7条规则的时候,我们只能分析协议的保密性,而不能分析协议的原子性和认证性,其分析能力相当有限。下面以IBS协议[7]的提供服务阶段为例来说明。IBS协议是由卡内基-梅隆大学开发的电子商务协议,其提供服务阶段过程如图1所示。

(1)$E\rightarrow S:\{\{\Pr ice\}_{K_S^{-1}},\Pr ice\}_{K_E^{-1}}$;

(2)$S\rightarrow Invoice:\{\{\Pr ice\}_{K_S^{-1}},\Pr ice\}_{K_E^{-1}}$;

(3)$S\rightarrow E:\{Service\}_{K_S^{-1}}$

(4)$E\rightarrow S:\{ServiceAcknowledge\}_{K_E^{-1}}$

(5)$S\rightarrow Invoice:\{\{ServiceAcknowledge\}_{K_E^{-1}}\}_{K_S^{-1}}$

图1　IBS协议提供服务阶段

首先用户E(以后用E表示用户)向服务提供方S(以后用S表示服务提供方)发送一个服务请求,S把这条消息复制到发票上,并发送一条签名的服务消息给E,E收到服务后,发送一个签名的服务认可消息给S,S把它复制到发票上。E收到$\{Service\}_{K_S^{-1}}$,说明E就收到了服务。S收到$\{ServiceAcknowledge\}_{K_E^{-1}}$,说明$S$就能够收到付费。

按照文[3]对协议的分析过程,首先建立交易各方的初始模型如图2;然后列举需要判定的公式:S^1 *Knows Price*,$Invoice^2$ *knows Price*,E^3 *Knows Service*,S^4 *Knows ServiceAcknowledge*,$Invoice^5$ *knows ServiceAcknowledg*,除此之外还需要判定E^3 *knows Service SignedBy* E,E^4 *knows ServiceAcknowledge SignedBy* S等是否成立。其中S',E',$Invoice'$分别表示交易方S,E,*Invoice*的第i步演化结果,如图3。S^1 *Knows Price*表示S在运行完成第i步之后,应该拥有*price*这条信息,其它公式与此类似,不再说明。在下面的描述中,只描述了

郭云川　助教,硕士,研究方向:信息安全,形式化技术;**古天龙**　教授,博士生导师,主要研究方向:形式化技术,符号模型检验,协议工程等;**董荣胜**　副教授,主要研究方向:形式化技术,计算机科学与技术方法论等;**蔡国永**　副教授,主要研究方向:实时系统,面向对象软件工程,协议验证技术等。

E 和 S 的当前行为而忽略了的其它行为，其中 $lstv_E$、$lstv_S$ 分别表示协议运行前 E、S 拥有的信息。

$E \equiv E[K_E^{-1}, K_E, K_S, S, E, Invoice]([\{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}} \{Service?\}_{K_S^{-1}} -> \{ServiceAcknowlege\}_{K_S^{-1}} \hat{}].[])$

$S \equiv S[K_S^{-1}, K_E, K_S, S, E, Invoice([\{\{Price?\}_{K_S^{-1}}, Price?\}_{K_E^{-1}} -> \{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}} \hat{} \{Service\}_{K_S^{-1}} \hat{} \{ServiceAcknowlege?\}_{K_E^{-1}} -> \{\{ServiceAcknowlege\}K_E^{-1}\}_{K_S^{-1}} \hat{}].[])$

$Inoice \equiv Invoice[K_E, K_S, S, E]([\{\{Price?\}_{K_S^{-1}}, Price?\}_{K_E^{-1}} -> \{\{ServiceAchnowlege\}_{K_E^{-1}}\}_{K_S^{-1}} ->].[])$

图2 协议模型

对于 S^1 *Knows Price*，分析如下：

(1)

$$\frac{E[lstv_E]([\{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}} \hat{} \cdots].[]) \xrightarrow{\{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}}} E[lstv_E \cup \{\{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}}\}]([\cdots].[])}{\cdots \quad \cdots\cdots 对\ E\ 使用\ out\ 规则}$$

即，$E^0 \xrightarrow{\overline{m}} E^1$.

(2) $\{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}} \sim \{\{Price?\}_{K_S^{-1}}, Price?\}_{K_E^{-1}}$ ……由 pattern-message 定义

(3)：

$$\frac{\{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}} \sim \{\{Price?\}_{K_S^{-1}}, Price?\}_{K_E^{-1}} \quad S[lstv_S]([\{\{Price?\}_{K_S^{-1}}, Price?\}_{K_E^{-1}} -> \cdots].[]) \xrightarrow{\{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}}} S[lstv_S \cup \{\{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}}\}]([\cdots].[])\{Price/Price?\}}{\cdots\cdots 由\ in\ 规则}$$

即：$S^0 \xrightarrow{m} S^1$

(4) $E^0 \& S^0 \Rightarrow E^1 \& S^1$ ……Comm 规则

(5)

$$\frac{S[lstv_S \cup \{\{Price\}_{K_S^{-1}}, Price\}_{K_S^{-1}}]([\cdots].[])\ knows\ \{\{Price\}_{K_S^{-1}}, Price\}_{K_S^{-1}},\ S[lstv_S \cup \{\{Price\}_{K_S^{-1}}, Price\}_{K_S^{-1}}]([\cdots].[])\ knows\ K_E, K_E K_{S_1} K_{K_S^{-1}}}{S[lstv_S \cup \{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}}]([\cdots].[]) \equiv S[lstv_S \cup \{\{Price\}_{K_S^{-1}}, Price\}_{K_S^{-1}} \cup Price]([\cdots].[])}$$

……由 Extract 规则

(6) $S[lstv_S \cup \{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}} \cup Price]([\cdots].[])\ knows\ Price$

……由 Knows 规则

即 S^1 *Knows Price* 成立。同样可以判定 S^1 *Knows Price*，$Invoice^2$ *knows Price*，E^3 *Knows Service*，S^4 *Knows ServiceAcknowledge*，$Invoice^5$ *knows ServiceAcknowledg* 成立。然而文[3]的方法不能对 E^3 *knows Service SignedBy E*，E^4 *knows ServiceAcknowledge SignedBy S* 作出判定，这是因为这种方法没有对加密解密签名过程所蕴涵的信息进行进一步分析。另外，IBS 协议并不满足电子商务协议原子性要求：按协议的设计，E 收到服务后在第5步提供给 S 一个签名的确认消息。但如果 E 不诚实，不响应 S 所提供的服务，那么 S 得不到服务应答消息。如果协议至此中止，那么 S 就无法提供它已向 E 提供服务的证据，因此 S 就不能获得付费，这样 IBS 协议不满足电子商务协议商品原子性（商品原子性的具体含义在后文叙述）的要求。文[3]中的方法并不能分析出 IBS 协议的这种缺陷，因为这种方法没有考虑通信故障和交易各方不诚实的情况，不能针对这种情况进行相应的分析。从上面的分析，可以看出文[3]的方法不能分析协议的原子性和认证性。下面我们在文[3]的基础上引入一种新方法使其能够分析电子商务协议的原子性。

$E^1 \equiv E[K_E^{-1}, K_E, K_S, S, Invoice, \{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}}]([\{Service?\}_{K_S^{-1}} -> \{ServiceAcknowlege\}_{K_S^{-1}} \hat{}].[])$

$S^1 \equiv S[K_S^{-1}, K_E, K_S, E, Invoive, \{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}}]([\{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}} \hat{} \{Service\}_{K_S^{-1}} \hat{} \{ServiceAcknowlege?\}_{K_E^{-1}} -> \{\{ServiceAcknowlege\}_{K_E^{-1}}\}_{K_S^{-1}} \hat{}].[])$

$Inoice^1 \equiv Invoice[K_E, K_S, S, E]([\{\{Price?\}_{K_S^{-1}}, Price?\}_{K_E^{-1}} -> \{\{ServiceAcknowlege\}_{K_E^{-1}}\}_{K_S^{-1}} ->].[])$

$E^2 \equiv E[K_E^{-1}, K_E, K_S, S, Invoice, \{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}}]([\{Service?\}_{K_S^{-1}} -> \{ServiceAcknowlege\}_{K_S^{-1}} \hat{}].[])$

$S^2 \equiv S[K_S^{-1}, K_E, K_S, E, Invoice, \{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}}, \{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}}]([\{Service\}_{K_S^{-1}} \hat{} \{ServiceAcknowlege?\}_{K_E^{-1}} -> \{\{ServiceAcknowlge-g\}_{K_E^{-1}}\}_{K_S^{-1}} \hat{}].[])$

$Inoice^2 \equiv Invoice[K_E, K_S, S, E, \{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}}]([\{\{ServiceAcknowlege\}_{K_E^{-1}}\}_{K_S^{-1}} ->].[])$

$E^3 \equiv E[K_E^{-1}, K_E, K_S, S, invoice, \{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}}, \{Service\}_{K_S^{-1}}]([\{ServiceAcknowlege\}_{K_S^{-1}} \hat{}].[])$

$S^3 \equiv S[K_S^{-1}, K_E, K_S, E, invoice, \{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}}, \{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}}, \{Service\}_{K_S^{-1}}]([\{ServiceAcknowlege?\}_{K_E^{-1}} -> \{\{ServiceAcknowlege\}_{K_E^{-1}}\}_{K_S^{-1}} \hat{}].[])$

$Inoice^3 \equiv Invoice[K_E, K_S, S, E, \{\{Price\}_{K_S^{-1}}, Price\}_{K_i^{-1}}]([\{\{ServiceAcknowlege\}_{K_E^{-1}}\}_{K_S^{-1}} ->].[])$

$E^4 \equiv E[K_E^{-1}, K_E, K_S, S, Invoice, \{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}}, \{Service\}_{K_S^{-1}}, \{ServiceAcknowlege\}_{K_S^{-1}}]([].[])$

$S^4 \equiv S[K_S^{-1}, K_E, K_S, E, Invoice, \{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}}, \{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}}, \{Service\}_{K_S^{-1}}, \{ServiceAcknowlege\}_{K_S^{-1}}]([\{\{Service\ Acknowlege\}_{K_E^{-1}}\}_{K_S^{-1}} \hat{}].[])$

$Inoice^4 \equiv Invoice[K_E, K_S, S, E, \{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}}]([\{\{ServiceAcknowlege\}_{K_E^{-1}}\}_{K_S^{-1}} ->].[])$

$E^5 \equiv E[K_S^{-1}, K_E, K_S, S, Invoice, \{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}}, \{Service\}_{K_S^{-1}}, \{ServiceAcknowlege\}_{K_S^{-1}}]([].[])$

$S^5 \equiv S[K_S^{-1}, K_E, K_S, E, Invoice, \{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}}, \{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}}, \{Service\}_{K_S^{-1}}, \{ServiceAcknowlege\}_{K_S^{-1}}, \{\{ServiceAcknowlege\}_{K_E^{-1}}\}_{K_S^{-1}}]([].[].)$

$Inoice^5 \equiv Invoice[K_E, K_S, S, E, \{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}}, \{[\{\{ServiceAcknowlege\}_{K_E^{-1}}\}K_S^{-1}]([].[])$

图3 产易各方的演化

3 电子商务协议形式化分析的新方法

事务的原子性是数据库最根本的概念之一，T. D. Tygar 在电子商务中引入了原子性的概念，以规范电子商务中的资金流、信息流和物流。T. D. Tygar 将原子性分为三级[5]：钱原子性，商品原子性和确认发送原子性。为了分析协议的原子性，首先对协议语句按照可否中断进行如下分类，其中 P，Q

表示任意不可信任主体。

(1)i_P:P->Q:M 表示在第 i_P 步,主体 P 向主体 Q 发送消息 M。此语句是可中断的,因为可能存在通信故障或 P 欺诈。

(2)i_P:P<-->TTP:M 表示主体 P 在第 i_P 步,通过一次或多次 ftp 操作从可信任第三方 TTP(Truested Third Party)中取得消息。此语句是不可中断的,因为在通信信道不可靠的情况下,P 可以多次向 TTP 进行 ftp 操作从而获得 M[7]。

定义 函数 $f(i_P,P,Q)$表示在协议语句中断的情况下,主体 P 运行完成第 i_P 步之后,主体 Q 至多能够运行完成的步数。

例1 在 IBS 协议的提供服务阶段,当 $i_E=3$的时候,$f(i_E,E,S)$的值为3。因为当客户 E 执行完成协议第3步(即 $i_E=3$)的时候,如果通信中断或者 E 不诚实,那么商家 S 不能接收到消息4,即在这种情况 S 至多能够执行完成第3步。

函数 $f(i_P,P,Q)$的计算方法如下:

如果协议语句的格式为"i_P:P->Q:M",则朝后(即协议语句编号小于 i_P 的方向)查找格式为"j:X->P:M"的协议语句,其中 X 为不同于 P 的其他任何主体。如果找到一条这样的语句,则 $f(i_P,P,Q)$取 j;如果找到多条这样的语句,那么 $f(i_P,P,Q)$取最大的 j;否则 $f(i_P,P,Q)$取0。

如果协议语句的格式为"i_P:Q->P:M",则 $f(i_P,P,Q)$的值取 i_P。

如果在第 i_P 步,协议语句的发送接受双方不同时为 P 和 Q,那么就不必计算 $f(i_P,P,Q)$

我们将上述函数定义扩展到不可中断的情况,那么如果协议语句格式为"i_P:<-->TTP:M",则 $f(i_P,TTP,P)$的值取 i_P。

例2 计算 IBS 协议提供服务阶段的 $f(i_E,E,S)$和 $f(i_S,S,E)$。

表1 例2的计算结果

i_E	$f(i_E,E,S)$	i_S	$f(i_S,S,E)$
1	0	1	1
3	3	3	1
4	3	4	4

为了分析协议是否满足原子性,需要判断在协议执行的每一步,在通信故障或交易方欺诈的情况下,交易各方是否拥有应该拥有的信息。我们用 RP(Receiving Payment)表示商家 M 收到货款,用 SP(Sending Payment)表示客户 C 支付了货款,用 RG(Receiving Good)表示客户接收到了货物。下面对电子商务协议的原子性进行形式化描述。

(1)钱原子性。商家收到钱当且仅当客户支付了钱。即要求:

R1 C' knows SP->$M^{f(i_C,C,M)}$ knows RP

R2 M' knows RP->C' knows SP

R1意味着如果客户 C 在第 i 步支付了货款,那么商家 M 运行完成第 $f(i_C,C,M)$步就必须收到货款。因为在协议语句中断的情况下,客户 C 运行完成第 i 步,商家最多运行完成 $f(i_C,C,M)$步,如果商家 M 在运行完成 $f(i_C,C,M)$步之后不能收到货款,商家 M 就可能收不到货款。这样协议不满足R1,即此协议不具有钱原子性。R2表示商家 M 在第 i 步收到了货款,那么客户 C 在运行完成第 i 步之前必须支付了货款。->表示逻辑蕴含。R3~R6的含义与此类似,以后不再说明。

(2)商品原子性。商品原子性要求商家收到货款当且仅当客户收到商品。

对于商家收到货款之后才发送货物的情况,商品原子性要求协议必须满足 R3,R4:

R3: M' knows RP -> $C^{f(i_M,M,C)}$ knows RG

R4: C' knows RG -> M' knows RP

对于客户要求收到货物之后才支付货款的情况,商品原子性必须满足公式 R5,R6:

R5: C' knows RG -> $M^{f(i_C,C,M)}$ knows RP

R6: M' knows RP -> C' knows RG

(3)确认发送原子性,要求对客户购买和商家销售的商品的内容和品质进行确认。这涉及到安全性中的非否认性,我们就不对其分析。

显然,如果对于资金只在银行内部服务器上划拨的协议,比如 Netbill 协议[5],那么它必定满足 R1和 R2,即满足钱原子性,因此对于这种协议只需分析其商品原子性。基于上述分析,此方法对电子商务协议的原子性分析包括如下步骤:

(1) 建立系统的初始模型;

(2) 列出 RP 和 RG 或者 RG,SP 和 RP;

(3) 根据不同的支付方式以及发送货物与支付货款的不同顺序,选择相应的逻辑公式,计算相应的函数 f,并列出协议所必需满足的逻辑公式;

(4)用文[3]中的 Knows,Extract,Construct 等规则判定第3步列出的公式是否全部成立,如果全部成立,则此协议满足协议原子性要求,否则不满足。

4 应用举例

我们仍然以 IBS 协议的提供服务阶段为例来说明。

首先描述协议模型,如图2;然后列举交易方应该拥有的信息:客户 E 的 RG 为 service,提供服务方 S 的 RP 为 ServiceAcknowledge;计算 $f(i_E,E,S)$如表1。IBS 协议采用转帐的方式,所以它满足钱原子性。此协议属于先提供服务,再付费的情况,因此其商品原子性须满足如下规则:

对于所有 i ($0<i<=4$)必须有

(1) E' knows service -> $S^{f(i_E,E,S)}$ knows ServiceAcknowledge

(2)S' knows ServiceAcknowledge ->E' knows service

即

(A1) E^1 knows service -> S^0 knows ServiceAcknowledge;

(A2) S^1 knows ServiceAcknowledge→E^1 knows service;

(A3) E^3 knows service -> S^3 knows ServiceAcknowledge

(A4) S^3 knows ServiceAcknowledge-> E^3 knows service;

(A5) E^4 knows service -> S^3 knows ServiceAcknowledge;

(A6) S^4 knows ServiceAcknowledge->E^4 knows service

如果上述公式任何一个公式不成立,那么此协议将不满足电子商务协议的商品原子性。为了判定公式(A1)是否成立,我们首先检查公式 E^1 knows service。因为 $E^1\equiv E[lstv_E\cup\{\{Price\}_{K_S^{-1}}\}_{K_S^{-1}}]([\cdots],[])$,显然 Service $\notin lstv_{E1}$,根据

(下转第112页)

路径一致的空间关系进行编码。

表2 区域/区域关系的拓扑解释

区域/区域关系	拓扑约束
$Dis_{a/a}(A,B)$	$CA\cap CB=\varnothing, IA\cap IB=\varnothing$
$Tan_{a/a}(A,B)$	$IA\cap IB=\varnothing, CA\cap CB\neq\varnothing$
$Ove_{a/a}(A,B)$	$IA\cap IB\neq\varnothing, CA\cap CB\neq\varnothing, IA\neq IB$
$Equ_{a/a}(A,B)$	$CA=CB, IA=IB$
$T\text{-}Con_{a/a}(A,B)$	$IB\subset IA, CA\cap CB\neq\varnothing$
$T\text{-}Ins_{a/a}(A,B)$	$IA\subset IB, CA\cap CB\neq\varnothing$
$S\text{-}Con_{a/a}(A,B)$	$IB\subset IA, CA\cap CB=\varnothing$
$S\text{-}Ins_{a/a}(A,B)$	$IA\subset IB, CA\cap CB=\varnothing$

4 具体域可满足性检验算法

可是，一般来说路径一致性并非充分的一致性标准。因此，一个补充步骤是要求保证全局的一致性。根据 Nebel 和 Renz 的讨论[8,9]，最坏的情况的复杂性取决于在约束网络中实际遇到的关系(基本关系的析取)，在这种情况下，为了实现全局一致性，指数算法是必需的。可是，Nebel 和 Renz 证明了：对于确定的 EDP 子集来说，全局一致性等效于路径一致性。因此，一个用于具体域可满足性检验的算法可以分为如下三个主要步骤：

(1)"非"谓词通过相应的基本谓词束语的析取来置换。然后通过单个谓词或谓词的析取组成每个合取式。此外，对于每个合取式来说，可以添加一个新的表示逆关系的合取式。

(2) 位置谓词引进了具体空间对象。借助于计算几何的标准算法，具体空间对象之间的基本拓扑关系可以被简化为两个具体空间对象交集的检验，可以把具体空间对象视为变量。

(3) 第三步是检验关系的一致性。必须计算全局一致解(路径一致性可以被简化为修剪步骤)。如果没有找到全局一致解，返回"不可满足"，否则返回到"可满足的"。

可见，具体领域 DP 是容许的。

结束语 在本文中，我们定义了适用于由点、线、和区域共同组成的具体域 DP，给出了具体域可满足性检验算法，为提出了适用于空间推理的 DLSR(DP)描述逻辑奠定基础。在接下来的研究中，我们将提出描述逻辑 DLSR(DP)，讨论其语法、语义、约束和算法。

参 考 文 献

1 Baader F, Hanschke P. A scheme for integrating concrete domain into concept languages. In: Twelfth Intl. Joint Conf. on Artificial Intelligence, Aug. 1991. 452～457

2 Lutz C. Reasoning with concrete domain. In: Proc. of the Sixteenth Intl. Joint Conf. on Artificial Intelligence IJCAI-99, Stockholm, Sweden, 1999

3 Lutz C. Reasoning with Complexity of terminological reasoning revisited. In: Proc. of the 6th Intl. Joint Conf. on Logic for Programming and Automated Reasoning (LPA'99), number 1705 in Lecture Notes in Artificial Intelligence, Springer-Verlag, Step. 1999. 181～200

4 Horrocks I, Sattler U, Tessaris S, Tobies S. Query Containment using a DLR ABox: [LTCS-Report 00-15]. LuFG Theoretical Computer Science, RWTH Aachen, Germany

5 Faltings B. Qualitative spatial reasoning using algebraic topology. In: A U Frank and W Kuhn, eds. Spatial Information Theory: a theoretical basis for GIS, volume 988 of Lecture Notes in Computer Science, Berlin, Springer-Verlag, 1995. 17～30

6 Randell D A, et al. A spatial logic based on regions and connection. In: B. Nebel, W. Swartout, C. Rich, eds. Principles of Knowledge Representation and Reasoning of the 3rd Intl. Conf. - Cambridge MA, Morgan Kaufmann, 1992. 165～176

7 Baader F, Hanschke P. A scheme for integrating concrete domains into concept languages. In: Twelfth Intl. Joint Conf. on Artificial Intelligence, Aug. 1991. 425～457

8 Nebel B. Computational properties of qualitative spatial reasoning: First results. In: I. Wachsmuth, C. -R. Rollinger, W. Brauer, eds. Proc. KI-95: Advances in Artificial Intelligence, 19th Annual German Conf. on Artificial Intelligence, Volume 981 of LNAI, Springer-Verlag, Sep. 1995. 233～244

9 Renz J, Nebel B. On the complexity of qualitative spatial reasoning: A maximal tractable fragment of the region connection calculus. In: Proc. of 15th Intl. Joint Conf. on Artificial Intelligence (IJCAI97), Aug. 1997. 522～527

(上接第88页)

Knows 规则，需要判定能否找到一个等价的主体 $E^{1'}$ 使得 $E^{1'}$ knows service。对 E^1 使用 Extract 规则，有：

$E[lstv_E \cup \{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}}]([\cdots].[]) \equiv$

$E[lstv_E \cup \{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}} \cup Price]([\cdots].[]) \equiv E^{1'}$

因此 $Service \in lstv_{E^{1'}}$，除此之外还可以用 Construct 规则对 E^1 进行构造，但构造中不可能生成 service，因此 E^1 knows service 为 false，因此(A1)成立，同样，我们可以判定(A2)成立。现在判定(A3)是否成立。

$E^3 \equiv E[lstv_E \cup \{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}} \cup \{Service\}_{K_S^{-1}}]([\cdots].[])$利用 Extract 规则，有

$E[lstv_E \cup \{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}} \cup \{Service\}_{K_S^{-1}}])([\cdots].[]) \equiv$

$E[lstv_E \cup \{\{Price\}_{K_S^{-1}}, Price\}_{K_E^{-1}} \cup \{Service\}_{K_S^{-1}} \cup Service]([\cdots].[])$

再使用 Knows 规则，可知 E^3 knows service 为 True。检查公式 S^3 knows ServiceAcknowledge，

$S3 \equiv S[lstv_S \cup \{\{price\}_{K_S^{-1}}, price\}_{K_E^{-1}}, \{Service\}_{K_S^{-1}}]([\cdots].[])$明显 S^3 knows ServiceAcknowledge 为 false。因此公式(A3)不成立。因此 IBS 协议不满足商品原子性。

结束语 本文讨论了文[3]所提出的分析电子商务协议的方法，指出了文[3]所存在的一些局限，并在分析了电子商务原子性的特点的基础上，提出了一种新的方法使其能够对电子商务协议的原子性进行分析。然而这种方法还不分析协议的认证性，如何将进程演算和逻辑分析近一步结合，以分析协议认证性等安全性质，这仍需要进一步研究。

参 考 文 献

1 Gritzalis S, Spinellis D. Georgiadis P. Security Protocols over Open Networks and Distributed Systems: Formal Method for Their Analysis, Design and Verification. Computer Communications, 1999, 22(8): 695～707

2 Burrows M, Abadi M, Needham R. A Logic of Authentication. ACM Transactions on Computer Systems, 1990, 8(1): 18～36

3 Papa M, Bremer O, Hale J, Shenoi S. Formal Analysis of E-Commerce Protocols. In: Proc. of the 5th Intl. Symposium on Autonomous Decentralized Systems, Vol. E84-D, 2001. 19～28

4 Abadi M, Gordon. Reasoning about Cryptographic Protocols in the Spi Calculus. CONCUR'97: Concurrency Theory, 1997, 1243: 59～73

5 Tygar J D. Atomicity in Electronic Commerce. In: Proc. of the 15th Annual ACM Symposium on Principles of Distributed Computing. 1996. 8～26

6 Zhou Jian-ying, Gollman D. A Fair Non-repudiation Protocol. In: Proc. of the 1996 IEEE Symposium on Security and Privacy. 1996. 55～61

7 O'Toole K R. The Internet Billing Server Transaction Protocol Alternatives. INITR-1, Carnegie Mellon University, Information Networking Institute. 1994, 4