# 基于消息唯一起源的动态逻辑方法

谢鸿波[1,2], 吴远成[1], 周明天[1]
(1. 电子科技大学计算机科学与工程学院, 四川成都 610054; 2. 重庆通信学院三系数据链教研室, 重庆 400035)

摘 要: 本文提出了一种新的逻辑方法分析安全协议的安全性. 该方法给出了一种安全协议的动态分析模型, 从而克服了类 BAN 逻辑“理想化协议”步骤的缺陷, 提出了消息唯一起源的概念和判定规则, 严格区分“可靠信任”和“不可靠信任”, 解决了“相信事情的发生”和“相信事情的真实性”两种不同信任的区别, 并在此基础上建立了动态逻辑方法. 通过实例分析, 该方法可以发现类 BAN 逻辑不能发现的协议漏洞, 从而证明了方法的有效性.

关键词: 协议分析; 动态逻辑; 类 BAN 逻辑
中图分类号: TP393 文献标识码: A 文章编号: 0372-2112 (2007) 08-1516-05

# Dynamic Logic Method Based on Message Unique Origin

XIE Hong-bo[1,2], WU Yuan-cheng[1], ZHOU Ming-tian[1]
(1. *School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan 610054, China;*
2. *Data Like Staff Room of the 3rd Department, Chongqing Communication College, Chongqing 400035, China*)

**Abstract:** A new logic method for analyzing security protocols was presented in this paper. A dynamic model was presented, which overcame the flaw of the BAN-like logic in its protocol idealization step. The basic concept of Message Unique Origin (MUO) and its determinant rules ware presented, which could be used to distinguish between "sound trust" and "unsound trust". The difference between "believe the occurrence of the event" and "believe the truth of the event" was resolved. Based on the concept of MUO, a new dynamic logic is build up, whose validity is proved by an example protocol which is soundness in the BAN-like logic but is found to have some flaws by this dynamic logic.

**Key words:** protocol analysis; dynamic logic; BAN-like logic

## 1 引言

类 BAN 逻辑在安全协议的形式化分析方法中占有重要的地位. 他们在协议分析中通过“知识”和“信任”推理发现协议缺陷或漏洞. 然而, 大部分类 BAN 逻辑的推理规则都是单调的, 即: 一旦某些事物被“信任”或“知道”, 那么在整个协议分析过程中他们都被“信任”或“知道”. 这会导致协议实体不能根据收到的消息和协议规范来动态地修改对某些事物的“信任”或“知道”, 从而在协议运行结束时产生错误的信任.

单调推理规则是不能解决实体的动态信任问题的. 目前在解决推理规则的非单调性问题上有三个尝试. AT[1]提出了一个负结构“¬”, 以及相关的信任规则: $\neg P|\equiv\phi \Rightarrow P|\equiv(\neg P|\equiv\phi)$, 即: 如果 $P$ 不相信 $\phi$, 那么 $P$ 相信 $P$ 不相信 $\phi$. 由于没有给出“负”信任的算法描述, 使得该信任规则在推理过程中并没有起到推导实体动态信任的作用. Moser[2]提出一个信任的非单调逻辑, 通过一个“*unless*”操作符推导实体的动态信任, 即: 如果没有别的信任反驳, 那么实体对于 $\phi$ 的信任就为真. 该逻辑只是“信任”的非单调性推理, 没有考虑到实体因为对消息及其子消息的拥有而产生的“知识”的非单调性, 更没有涉及到两者之间的关系. Rubin[3]提出了一个方法来推导“知识”的非单调性. 由于该方法只是推导“知识”的非单调性, 同样没考虑到“信任”和“知识”两者的非单调性的关系.

本文针对推导实体的动态信任存在的问题, 提出了基于消息唯一起源的动态逻辑分析方法. 通过消息唯一起源的概念和判定规则, 本文将“信任”和“知识”两者的非单调性关联在一起, 进而给出了安全协议的动态分析模型.

本文的安排如下: 第二节描述本文方法所用到的符号定义, 给出消息唯一起源的概念和算法; 第三节给出逻辑推理规则和安全协议的动态分析模型; 第四节通过对 Otway-Rees 协议的分析证明了该方法的有效性; 第五节总结本文的工作.

收稿日期: 2005-05-24; 修回日期: 2007-05-28
基金项目: 电子科学基金(No. 5145001010DZ02); 国家 973 重点基础研究发展规划(No. G1999035803)

## 2 消息唯一起源的概念和判定规则

对于本文用到的符号, 我们有以下定义.

新鲜性: $\#(M)$, 在本轮协议运行前从没有发送过的数据.

说过: $P|\sim X$, 实体 $P$ 在某个时候说过消息 $X$.

好秘密: $Good(S)$, 除了秘密的产生者和预期接收者, 没有第三者知道该秘密. 如果 $S\in M$, 那么消息 $M$ 包含好秘密 $S$, 用 $Good(S)\in M$ 表示.

发送消息: $Send(P, M)$, 向实体 $P$ 发送了一个消息 $M$.

接收消息: $Receive(M)$, 实体 $P$ 接收到一个消息 $M$.

消息属性: $\langle Sender, Receiver, Run\_num, Msg\_num, Msg\_Format, Msg\_Content\rangle$ 六元组: 其中 $Sender$ 表示发送者, $Receiver$ 表示接收者, $Run\_num$ 表示协议运行轮次, $Msg\_num$ 表示本轮协议中的第几个消息, $Msg\_Format$ 表示该消息格式, $Msg\_Content$ 表示该消息内容. 消息属性中的 $Run\_num$ 和 $Msg\_Format$ 可以用来判定一个消息是否属于一个协议.

消息: $M(i)$, 在协议运行中第 $i$ 个消息, 在本文中它也表示子消息集合.

子消息: $m$, 构成消息 $M(i)$ 的独立数据单元, 如 Yahalom 协议中第 2 个消息 $\{B, N_b, \{A, N_a\}K_{bs}\}$, 其中 $B$, $N_b$, $\{A, N_a\}K_{bs}$ 都是子消息.

消息相似: $\approx$, 两个加密消息 $M(j)$ 和 $M(i)$ 由某些元素(其中部分元素是相同的, 包括位置和数值) 经过同样的密码学计算得到, 那么 $M(j)$ 和 $M(i)$ 相似, 用符号"$\approx$"表示; 反之用符号"$!\approx$"表示. 例如: 消息 $\{A, K\}K_{ab}$ 和消息 $\{A, N\}K_{ab}$ 是相似消息, 因为如果 $N$ 和 $K$ 具有相同长度, 那么这两个消息就很难分辨.

**定义 1** 消息唯一起源: 消息 $M(i)$ 是唯一起源的, 当且仅当 $M(i)$ 的消息属性能够被明确地判定属于本轮协议运行. 用符号 $Unique(M)$ 表示.

通过对实际协议的研究, 我们发现用定义 2.1 来判定协议消息很难操作, 因为很多实际的协议消息并不具备完整的消息属性. 因此, 本文给出了一个相对简单的判定规则.

当消息 $M(i)$ 同时满足以下的判定规则时, 我们说 $M(i)$ 是唯一起源的

(1) 消息格式满足协议规范第 $i$ 个协议消息的描述

(2) 消息包含有接收者可识别的临时值, 该临时值表明协议运行实例, 并且该临时值由接收者可识别的好密钥保护.

(3) 构成消息 $M(i)$ 的密文子消息的重新组合不能构成另一个合法的消息. 即: 一个协议运行中的消息实例, 不能通过其子消息的重新组合就变成协议运行的另一个实例中的有效消息.

(4) $\forall m\in M(i)$, if $m$ 是密文, then $m\ !\approx m(j)$ or $m\notin M(j), j<i$.

**定义 2** 消息属性的真实性: 消息 $m$ 的消息属性是真实的, 当且仅当 $m\in Unique(M)$, 即: $m$ 满足消息唯一起源.

**定义 3** 可靠信任: 实体对消息属性有可靠信任, 当且仅当该实体可以确定消息属性的真实性. 用符号 $P|\equiv M$, 表示实体 $P$ 可靠信任消息 $M$. 反之, 称为不可靠信任, 用符号 $P|\not\equiv M$ 表示, 即: 实体 $P$ 信任消息 $M$ 的发生, 但不能确定其真实性.

## 3 动态逻辑方法

### 3.1 集合定义与实体动作

"知识"和"信任"是有区别的. "知识"表明实体"知道"某件事, 而"信任"表示实体"相信"事件的真实性.

协议的实体集合: 参与协议运行的所有实体的集合 $\{P1, P2, P3, \ldots, Pn\}$. 符号 $E(P)$ 表示协议 $P$ 的实体集合.

实体 $Pi$ 的"知识"集合: 实体 $Pi$ 拥有的协议消息. $Poss(P)$ 表示实体 $P$ 的"知识"集合.

实体 $Pi$ 的"信任"集合: 实体 $Pi$ 所持有的对协议消息的信任, 分为可靠信任和不可靠信任两类. $Bef(P)$ 表示实体 $P$ 的"信任"集合.

消息 $m$ 的"观察者"集合: 如果消息是明文, 那么这个集合就是所有能够监听到网络通信的实体; 如果该消息是密文, 那么这个集合就是可以监听到这个消息并能解密该消息的所有实体, 以及产生该消息的所有实体. 符号 $Observer(m)$ 表示消息 $m$ 的"观察者"集合.

本文对消息操作和实体动作的定义来自 Rubin[3] 的方法. 能产生信任变化的动作有四个: "产生新鲜值", "产生秘密"、"忘记消息"和"忘记秘密". 其中, 后两个动作是将与被忘记的内容相关的信任从信任集合中去掉, 而前两个动作让实体产生新的信任, 这些新的信任将影响以后的逻辑推理, 所以需要重新定义这些动作的信任更新规则.

$Generate\_secret(s)|_{M(i)}$: 实体 $P$ 在收到消息 $M(i)$ 后, 根据协议规范产生秘密 $s$. 实体 $P$ 在完成这个动作之后, 将对秘密 $s$ 的信任更新到自己的信任集合中. 根据消息 $M(i)$ 的两种不同条件, 将产生不同的信任.

(1) 消息 $M(i)$ 符合协议规范

(2) 消息 $M(i)$ 具有唯一起源性, 即: $Unique(M)$.

$Bef(P)=Bef(P)\cup\{P|\equiv\#(s)\}$, 当且仅当消息 $M(i)$ 满足条件 1; $Bef(P)=Bef(P)\cup\{P|\equiv Good(s), P|\equiv\#(s)\}$, 当且仅当消息 $M(i)$ 同时满足条件 1 和 2.

$Generate\_nonce(n)|M(i)\ or\ NULL$: 实体 $P$ 在收到消息 $M(i)$ 后, 根据协议规范产生临时值 $n$; 或者实体 $P$

主动产生临时值 $n$. 其信任更新为: $Bef(P) = Bef(P) \cup \{P| \neq LINK(n)^{[3]}, P| \equiv \#(s)\}$.

**3.2** 推理模型及规则

动态逻辑方法的推理模型如图 1 所示. 其中“接收消息动作”是指协议实体在接收到协议消息后对该消息的处理动作; “产生消息动作”是指协议实体根据收到的消息按照协议规范的要求产生消息的动作. 信任更新过程是通过前面描述的“动作信任更新”和后面将描述的推理规则更新来实现.

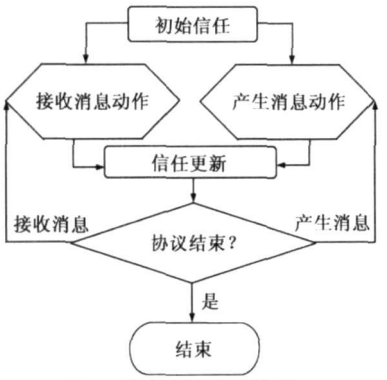

图 1 动态逻辑推理模型

这个推理模型有一个基本假设, 即: 协议实体的私钥和与可信第三方共享的秘密(或密钥)没有泄漏, 也就是说攻击者无法通过破解密钥来解密加密消息得到消息明文.

临时值验证规则:

$$\frac{(P| \equiv \#(M)) \in Bef(P), (P| \equiv (M \in Poss(P))) \in Bef(P)}{Bef(P) = Bef(P) \cup \{P| \equiv Q| \equiv \# M\}};$$

当实体 $P$ 相信 $M$ 的新鲜性, 并相信 $Q$ 都拥有 $M$, 那么实体 $P$ 相信 $Q$ 相信 $M$ 的新鲜性.

说过规则:

$$\frac{(P| \equiv Good(k)) \in Bef(P), \{P, Q\} \in Observer(k), \{Good(s) \subset M\}_k \in Poss(P)}{Bef(P) = Bef(P) \cup \{P| \equiv Q| \sim Good(s)\}};$$

实体 $P$ 相信密钥 $k$ 是它和实体 $Q$ 共享的好秘密, 而它又拥有用 $k$ 加密的含有好秘密 $s$ 含义的加密消息, 那么实体 $P$ 相信实体 $Q$ 说过秘密 $s$ 是个好秘密.

消息拥有规则:

$$\frac{(P| \equiv Good(k)) \in Bef(P), \{P, Q\} \in Observer(k), \{M\}_k \in Poss(P)}{Bef(P) = Bef(P) \cup \{P| \equiv (M \subset Poss(Q))\}};$$

实体 $P$ 收到一个用密钥 $K$ 加密的消息 $M$, 而该密钥只有 $P$ 和 $Q$ 知道, 那么 $P$ 相信实体 $Q$ 拥有 $M$.

好秘密规则:

$$\frac{(P| \equiv Q| \sim Good(s)) \in Bef(P), s \in Unique(M), M \in Poss(P)}{Bef(P) = Bef(P) \cup \{P| \equiv Q| \equiv Good(s)\}};$$

当实体 $P$ 相信实体 $Q$ 说过 $s$ 是好秘密, 并且 $s$ 包含在实体 $P$ 可以确定其唯一起源性的消息 $M$ 中, 那么实体 $P$ 相信 $Q$ 相信 $s$ 是好秘密.

信任传递规则:

$$\frac{\{P| \equiv Q| \equiv Good(x), P| \equiv Q| \equiv \#(x)\} \in Bef(P)}{Bef(P) = Bef(P) \cup \{P| \equiv Good(s)\}};$$

当 $P$ 相信 $Q$ 相信 $x$ 是好秘密, 并且 $P$ 相信 $Q$ 相信 $x$ 是新鲜的, 那么 $P$ 相信 $x$ 是好秘密.

消息含义规则:

$$\frac{\{x\}_k \in Poss(P), \{P, Q\} \in Observer(k), (P| \equiv Good(k)) \in Bef(P)}{Bef(P) = Bef(P) \cup \{x \in Poss(Q)\}};$$

$P$ 相信 $k$ 是 $P$ 和 $Q$ 的好密钥, 且 $P$ 收到了被 $k$ 加密的消息 $x$, 那么 $P$ 相信 $Q$ 拥有 $x$.

临时值子消息规则:

$$\frac{\#(x1) \in Bef(P), \{X \ contain \ x1, X \ contain \ x2\} \in Bef(P)}{Bef(P) = Bef(P) \cup \{\#(x2)\}};$$

这个规则直接引用类 *BAN* 逻辑的相似规则.

临时值信任判定规则 **1**

$$\frac{\{P| \equiv Good(k), P| \neq Link(N_a)\} \in Bef(P), f(N_a) \in Unique(\{M\}_k), \{M\}_k \in Poss(P)}{Bef(P) = Bef(P) \cup \{P| \equiv Link(N_a)\}};$$

临时值 $N_a$ 被实体 $P$ 用来标识本轮协议运行, 并且实体 $P$ 收到了包含 $N_a$ 的合法消息 $\{M\}_k$, 那么实体 $P$ 对临时值 $N_a$ 的信任由不可靠信任变成可靠信任.

临时值信任判定规则 **2**

$$\frac{\{P| \equiv Good(k), P| \equiv Link(N_a)\} \in Bef(P), X \ \ contain \ \ (f(N_a), x), \{X\}_k \in Poss(P)}{Bef(P) = (Bef(P) - P| \equiv Link(N_a)) \cup \{P| \equiv \#(x)\}\}};$$

消息 $X$ 被一个 $P$ 的好密钥加密, 并且 $X$ 中包含了 $P$ 的临时值 $N_a$, 那么 $P$ 将对 $N_a$ 的信任从他的信任集合中减去, 并将 $x$ 的新鲜性加入到 $P$ 的信任集合中.

将“消息唯一起源”概念引入逻辑推理公式中, 可以明确参与推理的消息来源, 从而得到一个可靠的信任结果, 它防止了并发协议运行时, 消息被攻击者滥用来进行攻击. 如“好秘密规则”中, 如果没有 $s \in Unique(M)$, 那么无法判定 $s$ 是否来自同一个协议的同一次运行, 因而, 也就得不到对 $Good(s)$ 的可靠信任.

上面定义的规则都是关于可靠信任的推导的. 我们用下面四个原则来表示可靠信任与不可靠信任之间仍的依赖关系:

原则 **1** 可靠信任+ 消息唯一起源性 ⇒ 可靠信任. 如果某消息具有消息唯一起源性, 那么将推导规则作用于它得到的结果也是可靠信任.

原则 **2** 可靠信任+ 不满足消息唯一起源性 ⇒ 不可靠信任. 如果某消息不满足消息唯一起源性, 那么推导规则作用于它得到的结果将是不可靠信任.

原则 **3** 不可靠信任+ 消息唯一起源性(该消息包含不可靠信任) ⇒ 可靠信任. 如果某消息具有消息唯一起源性, 那么该消息包含的不可靠信任通过推理规则可以得到可靠信任.

原则 **4** 不可靠信任+ 不满足消息唯一起源性 ⇒ 不可靠信任. 对于本身就是不可靠信任, 而接收到的消息又不具有唯一起源性, 那么应用推理规则的结果仍然是不可靠信任.

通过动作信任更新、推理模型及规则和依赖关系四原则, 我们描述了基于消息唯一起源性的动态逻辑方法.

## 4 Otway-Rees 协议分析

检验协议分析方法的最好办法就是通过实例来证明方法的有效性. 在本节中, 我们将用本文提出的动态逻辑方法对 Otway-Rees 协议进行分析. Otway-Rees 协议是一个共享密钥认证协议. 其协议描述如下:

***Message*** 1　$A \rightarrow B: M, A, B, \{N_a, M, A, B\}K_{as}$

***Message*** 2　$B \rightarrow S: M, A, B, \{N_a, M, A, B\}K_{as}, \{N_b, M, A, B\}K_{bs}$

***Message*** 3　$S \rightarrow B: M, \{N_a, K_{ab}\}K_{as}, \{N_b, K_{ab}\}K_{bs}$

***Message*** 4　$B \rightarrow A: M, \{N_a, K_{ab}\}K_{as}$

用类 BAN 逻辑对 Otway-Rees 协议进行分析, 可以得出该协议基本可靠的结论[4]. 我们用本文提出的动态逻辑方法重新分析了 Otway-Rees 协议.

实体 **A**　根据图 1 所示的动态逻辑推理模型, 对于实体 $A$ 来说有两个动作循环和相应的信任更新, 即: 产生消息动作$\rightarrow$接收消息动作.

初始信任: $Poss(A) = \Phi$, $Bef(A) = \{P \mid \equiv Good(K_{as})\}$, $Observer(K_{as}) = \{A, S\}$

**(1)** 产生消息动作

I 动作序列

(1) $Generate\text{-}secret(M) \mid NULL$

(2) $Generate\text{-}Nonce(N_a) \mid NULL$

(3) $Encrypt((N_a, M, A, B), K_{as})$

(4) $Concat((M, A, B), (\{N_a, M, A, B\}K_{as}))$

II 信任更新

动作(1)之后, $Poss(A) = Poss(A) \cup \{M\}$; $Bef(A) = Bef(A) \cup \{A \mid \equiv \#(M)\}$; 动作(2)之后, $Poss(A) = Poss(A) \cup \{N_a\}$; 根据"动作信任更新规则"信任更新为 $Bef(A) = Bef(A) \cup \{A \mid \neq LINK(N_a), P \mid \equiv \#(N_a)\}$; 动作(3)之后, $Poss(A) = Poss(A) \cup \{(\{N_a, M, A, B\}K_{as})\}$; 动作(4)之后, $Poss(A) = Poss(A) \cup \{N_c, (\{N_a, N_c\}K_{as})\}$.

**(2)** 接收消息动作

I 动作序列

(1) $Split(M_4)$, 即: 协议消息 4 分裂

(2) $Decrypt(\{N_a, K_{ab}\}K_{as}, K_{as})$

(3) $Split((N_a, K_{ab}))$

II 信任更新

动作(1)之后, $Poss(A) = Poss(A) \cup \{M, \{N_a, K_{ab}\}K_{as}\}$; 动作(2)之后, $Poss(A) = Poss(A) \cup \{(N_a, K_{ab})\}$; 动作(3)之后, $Poss(A) = Poss(A) \cup \{N_a, K_{ab}\}$, 根据临时值子消息规则、消息拥有规则、临时值验证规则、说过规则之后, 可以得到 $Bef(A) = \{A \mid \equiv \#(K_{ab}), A \mid \equiv (K_{ab} \in Poss(S)), A \mid \equiv S \mid \equiv \#(Kab), A \mid \equiv S \mid \sim Good(K_{ab})\}$, 由于 $\{N_a, K_{ab}\}K_{as} \approx \{N_a, M, A, B\}K_{as}$, 所以该消息不满足唯一起源性, 不能再更新于实体 $A$ 的信任集合.

协议完成后实体 $A$ 的信任集合为 $\{A \mid \equiv \#(M), A \mid \neq LINK(N_a), A \mid \equiv \#(N_a), A \mid \equiv \#(K_{ab}), A \mid \equiv (K_{ab} \in Poss(S)), A \mid \equiv S \mid \equiv \#(K_{ab}), A \mid \equiv S \mid \sim Good(K_{ab})\}$. 可以看出实体 $A$ 对与共享密钥 $K_{ab}$ 没有好秘密信任. 之所以没有好秘密的信任是因为消息 4 中, $\{N_a, K_{ab}\}K_{as} \approx \{N_a, M, A, B\}K_{as}$, 攻击者可以重放 $\{N_a, M, A, B\}K_{as}$ 消息来欺骗实体 $A$. 这样我们发现了 Otway-Rees 协议的一个漏洞.

实体 **B**　分析过程同 $A$ 类似, 可以发现存在类似 $A$ 的漏洞.

实体 **S**　实体 $S$ 有两个动作循环和相应的信任更新, 即: 接收消息动作$\rightarrow$产生消息动作.

初始信任: $Poss(S) = \Phi$, $Bef(S) = \{S \mid \equiv Good(K_{as}), S \mid \equiv Good(K_{bs})\}$, $Observer(K_{as}) = \{A, S\}$, $Observer(K_{bs}) = \{B, S\}$

**(1)** 产生消息动作

I 动作序列

(1) $Split(M, A, B, \{N_a, M, A, B\}K_{as}, \{N_b, M, A, B\}K_{bs})$

(2) $Decrypt(\{N_a, M, A, B\}K_{as}, K_{as})$

(3) $Decrypt(\{N_b, M, A, B\}K_{bs}, K_{bs})$

II 信任更新

动作(1)之后, $Poss(S) = Poss(S) \cup \{(M, A, B), (\{N_a, M, A, B\}K_{as}), (\{N_b, M, A, B\}K_{bs})\}$; 动作(2, 3)之后, $Poss(A) = Poss(A) \cup \{(N_a, M, A, B), (N_b, M, A, B)\}$.

**(2)** 接收消息动作

I 动作序列

(1) $Generate\text{-}secret(K_{ab}) \mid_{M2}$

(2) $Encrypt((N_a, K_{ab}), K_{as})$

(3) $Encrypt((N_b, K_{ab}), K_{bs})$

(4) $Concat(M, \{N_a, K_{ab}\}K_{as}, \{N_b, K_{ab}\}K_{bs})$

II 信任更新

动作(1)之后, $Bef(S) = Bef(S) \cup \{S \mid \equiv \#(K_{ab})\}$, 由于消息 2 不具有唯一起源性, 所以实体 $S$ 不能再得到进一步的信任; 动作(2, 3)之后, $Poss(S) = Poss(S) \cup \{\{N_a, K_{ab}\}K_{as}, \{N_b, K_{ab}\}K_{bs}\}$; 动作(4)之后, $Poss(S) = Poss(S) \cup \{M\}$.

由于消息 2 不具有唯一起源性, 因此, 实体 $S$ 可以被攻击者通过消息 2 欺骗. 过程为:

***Message*** 1　$C(A) \rightarrow B: M, A, B, \{N_c, M, C, B\}K_{cs}$

***Message*** 1'　$C \rightarrow B: M, C, B, \{N_c, M, C, B\}K_{cs}$

***Message*** 2' $B \rightarrow C(S): M, C, B, \{N_c, M, C, B\}K_{cs}, \{N_b, M, C, B\}K_{bs}$

***Message*** 2 $C(B) \rightarrow S: M, C, B, \{N_c, M, C, B\}K_{cs}, \{N_b, M, C, B\}K_{bs}$

***Message*** 3 $S \rightarrow B: M, \{N_c, K_{cb}\}K_{cs}, \{N_b, K_{cb}\}K_{bs}$

***Message*** 4 $B \rightarrow C(A): M, \{N_c, K_{cb}\}K_{cs}$

协议运行结果 $S$ 被入侵者 $C$ 欺骗产生$K_{cb}$, 并将结果传递给 $B$, 使得 $B$ 认为它在用$K_{cb}$与$A$ 通信.

Otway-Rees 协议被类 BAN 逻辑证明基本可靠, 但是, 通过本文的方法发现了两个有实际意义的漏洞.

## 5 结论

Mitchell 和 Datta 提出的 PCL 逻辑[5] 采用 cord 算子和迹来描述协议运行, 通过逻辑公理进行推导, 这种组合的分析方法与类 BAN 逻辑单纯的模态逻辑方法是不同的. PCL 逻辑明确提出了认证属性是协议动作之间的时间匹配关系, 并且通过模块化推理方法来分析协议组合的安全性, 这是类 BAN 逻辑不能实现的. 本文做的主要是工作是研究"消息唯一起源"在协议分析中的应用, 采用类 BAN 逻辑作为具体的实现方法, 用< Sender, Receiver, Run_ num, Msg_ num, Msg_ Format, Msg_ Content> 六元组表示消息的属性, PCL 逻辑也强调"消息唯一起源"的重要性, 定义了临时值的唯一性公理逻辑, 用 (source, destination, protocol-identifier, content) 四元组表示消息. 因此, 本文的工作与 PCL 逻辑从不同的方面研究和探讨了"消息唯一起源"在协议分析中的应用.

本文在消息唯一起源概念的基础上明确区分了可靠信任与不可靠信任, 进而提出了一种新的动态逻辑推理方法. 通过完善消息唯一起源判定规则, 增加新的推理规则, 可以对更多类型的协议进行安全性分析. 本文给出的推理模型也克服了其他逻辑方法中协议理想化步骤带来的分析缺陷. 最后通过实例的分析, 我们发现了 Otway-Rees 协议中类 BAN 逻辑不能发现的漏洞或缺陷, 从而证明了本文提出的动态逻辑方法是有效的.

参考文献:

[1] Abadi M, Tuttle M. A semantics for a logic of authentication [A]. Proceedings of the Tenth ACM Symposium on Principles of Distributed Computing[C]. New York: ACM Press, 1991. 201- 216.

[2] Louise E Moser. A logic of knowledge and belief for reasoning about computer security[A]. Proceedings of the Computer security Foundation Workshop II[C]. Los Alamitos: IEEE Comrputer Society Press, 1989. 57- 63.

[3] A D Rubin, P Honeyman. Nonmonotonic cryptographic protocols[A]. Proceedings of the 7th IEEE Computer Security Fourrdations Workshop VII[C]. Franconia, New Hampshire, USA: IEEE Computer Society Press, 1994. 100- 116.

[4] Burrows, M , Abadi, M, Needham, R. A logic of authentication [J]. ACM Trans Computer Systems, 1990, 8(1): 18- 36.

[5] Anupam Datta, Ante Derek, John C. Mitchell, Arnab Roy. Protocol composition logic( PCL) [J]. Electronic Notes in Theoretical Computer Science (ENTCS), 2007, 172: 311- 358.

作者简介:

谢鸿波 男, 1973 年 10 月生于四川仁寿, 1998 年获得解放军通信工程学院(现解放军理工大学) 硕士学位, 现为成都电子科技大学计算机应用专业博士研究生. 主要研究领域为网络与信息系统安全、分布对象技术.
E-mail: china_ xie2002@ 163. com

吴远成 男, 1973 年 1 月生于四川成都, 成都电子科技大学计算机应用专业博士研究生. 主要研究领域为网络与信息系统安全、分布对象技术.

周明天 男, 1939 年生于广西容县, 成都电子科技大学计算机学院教授, 博士生导师. 主要研究领域为计算机网络、分布对象技术、并行分布处理和网络与信息系统安全.