计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2019.20180466
56(6):1275-1289,2019

基于事件逻辑的 WMN 客户端与 LTCA 认证协议安全性分析

肖美华[1] 李娅楠[1,2] 宋佳雯[1] 王西忠[1] 李 伟[1] 钟小妹[1]
[1](华东交通大学软件学院 南昌 330013)
[2](中国铁建重工集团有限公司 长沙 410100)
(xiaomh@ecjtu.edu.cn)

Security Analysis of Authentication Protocol of WMN Client and LTCA Based on Logic of Events

Xiao Meihua[1], Li Yanan[1,2], Song Jiawen[1], Wang Xizhong[1], Li Wei[1], and Zhong Xiaomei[1]
[1](*School of Software*, *East China Jiaotong University*, *Nanchang* 330013)
[2](*China Railway Construction Heavy Industry*, *Changsha* 410100)

Abstract Wireless mesh network is a new type of broadband wireless network structure, which combines the advantages of wireless local area network and ad-hoc network. The research on wireless mesh network is one of the emerging research focuses about wireless networks. Based on the logic of events, the substitution rule is proposed to ensure the equivalent conversion of user interaction information in the process of property substitution by combining event structures, event classes, axiom clusters and random number lemma. With the basic sequences of authentication protocol between client and LTCA constructed by logic of events, the protocol actions between client and LTCA are formally described, and strong authentication property of the protocol is proved. Under reasonable assumptions, the security property of the authentication protocol between WMN client and LTCA is verified, and the research shows that both the security attributes of wireless network protocols and the authentication property between different principals of cryptographic protocols can be proved by logic of events. By simplifying the formal proof steps with flow chart, the process of logic of events proving protocol's security property is described, similarly, by comparing and analyzing logic of events with other logical reasoning methods, the universal applicability of logic of events is shown.

Key words event classes; logic of events theory; substitution rule; strong authentication property; WMN client and LTCA authentication protocol; universal applicability

摘 要 无线 Mesh 网络是一种新型的宽带无线网络结构,融合无线局域网与点对点模式两者的优势,是无线网络研究的热点之一.基于事件逻辑理论,结合事件结构、事件类、公理簇以及随机数引理,提出置换规则保证用户交互信息在性质置换过程中的等价转换.通过事件逻辑构建客户端与 LTCA 认证协议的基本序列,对协议交互动作进行形式化描述并证明协议强认证性质.在合理假设下,无线 Mesh 网络

收稿日期:2018-06-26;**修回日期**:2018-12-20
基金项目:国家自然科学基金项目(61163005,61562026);江西省自然科学基金项目(20161BAB202063);江西省主要学科学术和技术带头人资助计划项目(20172BCB22015)
This work was supported by the National Natural Science Foundation of China (61163005, 61562026), the Natural Science Foundation of Jiangxi Province of China (20161BAB202063), and the Major Academic and Technical Leaders Foundation of Jiangxi Province (20172BCB22015).

客户端与 LTCA 间认证协议的安全性得证,研究表明事件逻辑理论不仅可以论证无线网络协议的安全属性,还能对安全协议不同身份主体间的认证性进行证明.通过流程图简化协议形式化证明步骤,阐述事件逻辑理论证明协议安全属性过程,比较分析事件逻辑理论与其他逻辑推理方法,表明事件逻辑理论具有通用性.

关键词 事件类;事件逻辑理论;置换规则;强认证性质;WMN 客户端与 LTCA 认证协议;通用性

中图法分类号 TP393.08; TN925.93

网络发展给世界带来巨大变化,人们的日常生活离不开信息科技.信息在安全信道传输会遇到各种攻击,如消息窃听、消息拦截、消息篡改[1]等.网络安全问题是全球性问题,2017 年 3 月,维基解密(WiKiLeaks)公布数千份资料,其中揭秘一些黑客入侵事件[2],不法组织不仅入侵 iPhone 手机、Android 手机、智能电视,还攻击操作系统获取机密文件,甚至控制智能汽车发起暗杀活动[3].同年 5 月,在全球范围内爆发"WannaCry"比特币勒索病毒,结合蠕虫方式[4]进行传播,利用 MS17-010 漏洞向用户的 445 端口发送数据包,远程执行代码后加密用户文件,并释放一个压缩包,压缩包中文件隐藏释放到本地目录,文件中毒后被加密锁死.对 150 多个国家和地区、10 多万组织和机构以及 30 多万网民产生恶劣影响,损失高达 500 多亿人民币.

无线 Mesh 网(wireless mesh network, WMN)[5-6]拥有固定且电源充足的主干路由器,设计时不需要考虑太多能耗和移动性问题,适合投放于覆盖大面积开放区域.认证技术[7]在有线网络中比较成熟,但是在无线网络中由于存储设备、带宽等限制因素使其发展相对困难,无线网络面临比常规有线网络更严重复杂的威胁,如何构建合理的 WMN 认证体系很是关键.认证是安全通信的基础,通过机密消息对移动节点身份进行认定,只有经过认证的节点才能访问网络资源[8].

形式化方法通过严格数学概念和逻辑方法对协议进行描述,语义清晰、无歧义,可以发现其他方法不易被发现的网络安全协议漏洞[9].定理证明描述并发与分布式系统,是一种协议和算法的逻辑,侧重协议正确性,难以自动化[10].定理证明将协议描述为公理系统,协议安全目标表示需要证明的定理,协议是否满足安全目标对应于公理系统中目标定理是否成立,可用于无限状态空间协议正确性证明[11].事件逻辑(logic of events theory, LoET)[12]是定理证明的一种逻辑方法,用来刻画加密协议在交互过程中的消息动作.肖美华等人[13-15]合作运用事件逻辑理论对安全协议进行形式化分析工作,克服协议组合逻辑 PCL 在协议分析过程中存在的不足.

1 WMN 客户端与 LTCA 认证协议

轻量级动态容侵证书授权中心(lite and tolerate certificate authority, LTCA)认证机制结合门限密码思想与轻量级公钥体制[8],并支持 WMN 基础设施,相较于 MANET 及 WSN 的节点资源丰富[16].

本文在研究通信双方交互过程中进行 5 点假设:

1) 随机数是新鲜的,移动用户不生成相同随机数;

2) 以多跳形式经过中间转发的用户节点忽略不计;

3) 不存在任何值得注意的网络延迟现象;

4) 无线网络环境包括非法 Mesh 客户端在内所有用户均可窃取传输信息;

5) 本文所涉及的用户特指本地用户.

WMN 客户端与 LTCA 间认证协议满足 5 点:

1) 用户或节点的私钥由自己生成,节点主要指负责认证的服务器[17].

2) 移动用户(mobile user, MU).合法用户生成主公钥和私钥,允许进入 WMN,在有线网络认证中心(certificate authority, CA)处注册,具有 WMN 颁发证书,初始化后获得密钥对$\{(PK_{MU}^{(Master)}, PK_{MU}^{(Slavery)}), SK_{MU}\}$,访问 WMN,得到相关数据,与其他合法用户或服务器进行通信.

3) LTCA.由 1 组 Mesh 网认证服务器组成(MASs 是 WMN 中负责认证的 Mesh 路由器),负责 WMN 中用户 MU 的初始化,将 MU 主公钥与身份 ID_{MU} 由 LTCA 私钥 SK_{LTCA} 绑定签名生成辅公钥,对 WMN 异地用户进行身份认证(此部分不做研究)[8].

4) LTCA 公/私钥被分配到 n 个服务器,各认证服务器获得 1 对子公/私钥$(Q_i, d_i)(i=1,2,\cdots,n)$,LTCA 用 n 个认证服务器子私钥对用户主公钥

即身份签名经重构后产生辅公钥[8].

5) 引入时间戳机制保证消息在 CA 加/解密过程的延迟是在协议允许范围内.

结合协议假设和满足条件,当本地用户 MU 与其他用户通信时,WMN 客户端与 LTCA 间双向认证协议详细过程如图 1 所示:

$$\begin{aligned}&\text{LTCA}\rightarrow\text{MU}:\{R_{\text{LTCA}}\}\\&\text{MU}\rightarrow\text{LTCA}:\{PK_{\text{MU}}^{(\text{Slavery})},Sign_{PK_{\text{LTCA}}}(R_{\text{MU}}),Sign_{SK_{\text{MU}}}(R_{\text{LTCA}}),\\&\qquad Sign_{PK_{\text{LTCA}}}(h(PK_{\text{MU}}^{(\text{Slavery})}\mid R_{\text{MU}}\mid Sign_{SK_{\text{MU}}}(R_{\text{LTCA}}))),\\&\qquad T_1\}\\&\text{LTCA}\rightarrow\text{MU}:\{Cert_{\text{LTCA}},Sign_{SK_{\text{LTCA}}}(Sign_{PK_{\text{MU}}^{(\text{Master})}}(R_{\text{MU}})),\\&\qquad Sign_{PK_{\text{MU}}^{(\text{Master})}}(h(Cert_{\text{LTCA}}\mid R_{\text{MU}})),T_2\}\end{aligned}$$

Fig. 1 Two-way authentication between user and LTCA

图 1 用户与 LTCA 双向认证

1) LTCA→MU.LTCA 在收到用户 MU 请求时,生成随机数 R_{LTCA},然后发送给用户 MU.

2) MU→LTCA.MU 收到 LTCA 回应后,用私钥 SK_{MU} 对 R_{LTCA} 进行加密生成 S_1,MU 生成随机数 R_{MU},MU 对辅公钥 $PK_{\text{MU}}^{(\text{Slavery})}$、随机数 R_{MU} 以及 S_1 进行散列加密生成 h_1,通过 LTCA 公钥 PK_{LTCA} 对 h_1 进行加密得到 S_2,同时通过 LTCA 公钥 PK_{LTCA} 对随机数 R_{MU} 进行加密得到 S_3,加入时间戳 T_1,将 MU 辅公钥 $PK_{\text{MU}}^{(\text{Slavery})}$、加密消息 S_1,S_2,S_3 以及时间戳 T_1 打包发送给 LTCA.

3) LTCA 收到用户 MU 发送消息后,在时间 T_1 内用私钥 SK_{LTCA} 对 S_2,S_3 进行解密,因为 $PK_{\text{MU}}^{(\text{Slavery})}=Sign_{SK_{\text{LTCA}}}(ID_{\text{MU}}\mid PK_{\text{MU}}^{(\text{Master})})$,LTCA 获得$(ID_{\text{MU}},PK_{\text{MU}}^{(\text{Master})})$,根据 $PK_{\text{MU}}^{(\text{Slavery})}$ 验证 MU 是否为已注册用户,如果 MU 为已注册用户,则通过 MU 公钥 $PK_{\text{MU}}^{(\text{Master})}$ 验证检验 S_1 是否为 LTCA 生成的随机数 R_{LTCA}.如果是对应的 R_{LTCA},则说明发送消息用户为 MU,此时 LTCA 完成了对用户 MU 的认证,生成证书 $Cert_{\text{LTCA}}$ 发送给用户 MU.

4) LTCA→MU.LTCA 通过散列加密对证书 $Cert_{\text{LTCA}}$ 以及随机数 R_{MU} 进行加密生成 h_2,通过 MU 公钥 $PK_{\text{MU}}^{(\text{Master})}$ 进行加密得到信息 S_4,将随机数 R_{MU} 用 MU 公钥 $PK_{\text{MU}}^{(\text{Master})}$ 进行加密得到信息 S_5,通过 LTCA 私钥 SK_{LTCA} 对 S_5 进行加密得到消息 S_6,加入时间戳 T_2,将用户证书 $Cert_{\text{LTCA}}$、加密消息 S_4,S_6 以及时间戳 T_2 打包发送给 MU.

5) 用户 MU 得到证书 $Cert_{\text{LTCA}}$ 后对 S_6 解密得到 S_5,然后用私钥 SK_{MU} 解密对比得到的 R_{MU} 是否与之前生成的随机数匹配;用私钥 SK_{MU} 解密 S_4 得到 h_2,对证书 $Cert_{\text{LTCA}}$ 以及随机数 R_{MU} 进行散列加密得到 h_2',比较 $h_2'=h_2$ 是否成立,验证 LTCA 对于 MU 的可信性.

WMN 客户端 MU 与 LTCA 之间确定双向认证可信性,具体交互过程如图 2 所示:

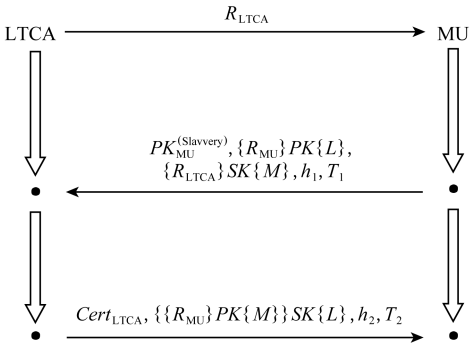

Fig. 2 Two-way authentication description

图 2 双向认证描述

2 事件逻辑理论

2.1 事件逻辑

事件逻辑对安全协议基本原语进行形式化规约,在协议研究中建立包含地址和事件的模型来证明协议安全属性,信息以多种形态存储在地址中,以消息形式在不同地址间进行传递.通过事件逻辑理论验证加密协议安全性,定义事件逻辑方法的布尔值(B)、标示符(Id)、原子($Atom$),其中原子表示随机数、签名、密文以及加密密钥等不可预测数据[9].定义事件、事件类以及事件结构构建认证理论,对协议强认证性质进行证明.

2.1.1 基本定义

1) $Atom$ 类型

$Atom$ 类型表示保密信息,其成员用 $atoms$ 表示,$atoms$ 是不可预测的.$Atom$ 类型成员是基本元素,没有结构且不能被生成.$Atom$ 类型用来评价规范形式 $tok(a)$,$tok(b)$,…,记为 $tokens$,其中 a,b,…为其中参数.

置换规则.事件逻辑中对任意判断 $J(a,b,\cdots)$,其中 a,b,…为有限集,则 $J(a',b',\cdots)$ 的真值过程 $a\mapsto a',b\mapsto b',\cdots$ 是一个置换.

2) 独立性

事件逻辑理论中定义(state after e)为事件 e 发生后主体出现的状态,事件 e 本质是一个空间或时间点.

$(x:T \parallel a)$表示 T 类型元素不包含原子 a，当 x 属于 T 类型时，$x \parallel a$ 表示 x 和 a 独立，即 T 类型的 x 独立于 a.

3) 事件结构

分布式计算的形式化模型可定义分布式系统的运算以及认证[9]，在运算中，e，$events$ 等消息是在迁移的，其中信息交互初始阶段为 $info(e)$.事件集是信息在一些存储位置(如主体在事件发生时进程、线程)中事件出现的空间/时间点.事件在单个位置时间点没有重叠，是整体序.无论如何迁移(消息传递、消息共享)，事件各主体间均会产生因果序.

计算系统语义通过事件结构语言描述，事件语言是任何语言的扩展.$\langle E, loc, <, info\rangle$即事件序(event-ordering)，其中 loc 是事件 E 上的函数，$<$ 是事件 E 的一个因果关系，是一个本地有限集.事件包含有限个前驱.$e \in E$，$loc(e)$是事件存储单元，$info(e)$表示事件发生时将消息交付给本地 $loc(e)$. $e_1 < e_2$ 表示事件 e_1 发生在事件 e_2 之前，事件结构存储单元表示主体、进程或者线程[10].

认证协议交互信息包括随机数、签名以及名字等元组信息，可以转化为

$$Data \equiv_{\text{def}} Tree(Id + Atom).$$

事件结构中事件语言建模时需满足：

① $\leqslant$表示局部有限偏序(每个事件 e 包含有限个前驱)[10]；

② $e_1 <_{\text{loc}} e_2 \equiv e_1 < e_2 \wedge loc(e_1) = loc(e_2)$表示局部序，事件集成员拥有相同存储单元的全序；

③ 认证理论中，$loc(e)$的位置是事件 e 发生的主体，对一些标示符 X 来说 $loc(e) = X$；

④ 协议安全性证明中 $e' < e \wedge loc(e') = loc(e)$ 可简化为 $e' < e$.

事件逻辑为消息自动机提供健全语义，消息可靠发送包含 named，fifo，link，每一条信息都有关联标签，是头信息.事件包含类型、价值，收到的信息在 link l 包含 tag tg，是类型为 $rcv(l, tg)$的事件 e.消息类型依靠(link，tag)来判断，例如 $sender(e) < e$ 表示发送消息事件.

事件 e 发送消息独立于原子 a，$sends(e) \parallel a$ 即 $\forall e' : E.(isrcv(e') \wedge sender(e') = e) \Rightarrow val(e') \parallel a$.

2.1.2 加密系统建模

构建加密协议层，定义随机数、加密消息，建立公/私钥对，以及执行复杂加密操作，例如消息共享等.

1) 随机数 n (nonce)

主体 i 通过虚服务 s_i 对主体生成随机数能力进行模型构建，该服务用来构建 s_i 到 i 的连接 l_i 以及 i 到 s_i 的连接 l^i，记录在无重复包含指针的列表 ats 内.主体 i 需要随机数时在连接 l^i 发送一个包含随机数的请求消息，通过 s_i 收到的请求消息在 $E(Nonce_i)$中生成事件 n 响应，即 $Nonce_i(n)$，其中主体 $atoms$ 在 ats 中是独立的.

域 $E(Nonce_i)$中，若随机数 $n \in E(Nonce_i)$和事件 e 满足$n \lessdot e \Rightarrow (val(e) \parallel Nonce_i(n))$，则从 events 到 $atoms$ 中偏序函数 $Nonce_i$ 属于随机数交互，即除非事件在随机数生成后与之产生因果，否则其值不包含该随机数.

2) 事件类(event class)

事件逻辑理论中，通过事件对协议进行分类描述，使用事件序语言、事件类对协议进行构建.T 类型事件类从事件到 T 类型值是简单的偏函数，如果 X 是一个事件类，则规定 $E(X)$是 X 范围内的事件集，$E(X)$中事件属于类 X.对于事件 $e \in E(X)$，$X(e)$是类 X 分配给 e 的 T 类型值.

一个事件类可以是任何不同事件类中的成员，如果 Y 是类型 T'的事件类，f 是 $T \to T'$的函数，那么 $X(v) \Rightarrow Y(f(v))$等价于 $\forall e. e \in X(v) \Rightarrow e \in Y(f(v))$.

若随机数 $n \in E(Nonce_i)$，事件 $e \in E(X)$没有关联，则$(loc(e) \neq s_i \Rightarrow X(e)) \parallel Nonce_i(n)$.

类中事件转换成原子 X 类中事件 e 信息包含原子 a，则：

$$X(e) \text{ has } a \equiv_{\text{def}} (e \in \exists(X) \wedge \neg(X(e) : T \parallel a)).$$

认证协议包含发送、接收、随机数、签名、认证、加密、解密 7 类事件[10]，事件类对应相关信息，信息类型取决于事件类，事件类中相关信息包含原子，基于 7 类事件类的形式化身份认证理论的类型列表具体定义如图 3 所示：

$$\begin{cases} \text{New}: EClass(Atom) \\ \text{Send}, \text{Rcv}: EClass(Data) \\ \text{Encrypt}, \text{Decrypt}: EClass(Data \times Key \times Atom) \\ \text{Sign}, \text{Verify}: EClass(Data \times Id \times Atom) \end{cases}$$

Fig. 3 Event classes of the authentication theory

图 3 认证理论的事件类

在图 3 中，类 Sign，Verify 中事件类型相同，均为三元组 $Data \times Id \times Atom$，其中类 Sign，Verify 为 $\langle signed(e), signer(e), signature(e)\rangle$.事件类型在

类 Encrypt,Decrypt 中为三元组 $Data \times Key \times Atom$,即$\langle encrypted(e), key(e), ciphertext(e)\rangle$.

对于 $e \in E(Sign)$,事件信息 $Sign(e)=\langle x, A, s\rangle$表示事件 e 是主体 A 对密文 x 进行签名生成 s.如果 A 是诚实主体,则 $loc(e)=A$,诚实主体不会释放私钥.对于事件 $e' \in E(Verify)$,该事件信息与签名信息相对应即 $Verify(e')=\langle x, A, s\rangle$表示事件 e' 是主体 $B=loc(e')$成功验证主体 A 签名生成的密文 x.

事件 $e \in E(Encrypt)$,$Encrypt(e)=\langle x, k, c\rangle$ 表示事件 e 是主体 A 通过密钥 k 对明文 x 加密生成密文 c,主体 A 拥有密钥 k 和消息 x.事件 $e' \in E(Decrypt)$中,$Decrypt(e')=\langle x, k', c\rangle$表示事件 e'是主体 B 通过密钥 k'解密密文 c 生成明文 x 的过程.密文 c 生成时,对应解密公理 $AxiomD$ 中存在一个匹配密钥.

2.2 事件逻辑公理、推论及性质

2.2.1 事件逻辑公理

对称密钥和私钥作为标示符是不可测的,两者并不相同,Key 类型为 $Key \equiv_{def} Id + Atom + Atom$.

事件逻辑中 $PrivKey$ 函数的主体包含原子,$MatchingKeys$ 函数构建密钥间的关系,如图 4 所示:

$$Honest: Id \rightarrow B$$
$$MatchingKeys: Key \rightarrow Key \rightarrow B$$
$$PrivKey: Id \rightarrow Atom$$

Fig. 4 Additional operators of the authentication theory

图 4 认证理论的附加操作

事件逻辑公理[9]包含密钥公理、诚实公理、因果公理、不相交公理、流关系,具体为

1) 密钥公理(key axiom,$AxiomK$)

密钥公理($AxiomK$)的匹配密钥间是对称的,密钥信息是主体特有,不同主体不会拥有相同私钥.主体 A 私钥仅能够与自身公钥匹配,具体表示为

$$\forall A, B: Id. \forall k, k': Key. \forall a: Atom$$
$$MatchingKeys(k; k') \Leftrightarrow MatchingKeys(k', k) \wedge$$
$$MatchingKeys(Symm(a); k) \Leftrightarrow k = Symm(a) \wedge$$
$$MatchingKeys(PrivKey(A); k) \Leftrightarrow k = A \wedge$$
$$MatchingKeys(A; k) \Leftrightarrow k = PrivKey(A) \wedge$$
$$PrivKey(A) = PrivKey(B) \Leftrightarrow A = B.$$

2) 诚实公理(honest axiom, $AxiomS$)

事件逻辑理论包含函数 $Honest: Id \rightarrow B$ 对诚实主体进行断言,诚实主体私钥不会释放,主体签名、加密以及解密事件均发生在诚实主体上,通过 $AxiomS$ 对诚实主体性质刻化为

$$\forall A: Id. \forall s: E(Sign). \forall e: E(Encrypt).$$
$$\forall d: E(Decrypt). Honest(A) \Rightarrow$$
$$\left\{\begin{array}{c} signer(s)=A \Rightarrow (loc(s)=A) \wedge \\ key(e)=PrivateKey(A) \Rightarrow (loc(e)=A) \wedge \\ key(d)=PrivateKey(A) \Rightarrow (loc(d)=A) \end{array}\right\}.$$

3) 因果公理(causal axioms)

因果公理是对事件类中接收事件 Rcv、验证事件 $Verify$、解密事件 $Decrypt$ 所对应事件公理(接收公理 $AxiomR$、验证公理 $AxiomV$、解密公理 $AxiomD$)关系的整合.$AxiomR$,$AxiomV$ 相似,任何接收或验证事件发生前存在一个与之相应且信息内容相同的发送或签名事件[10],具体为

$$AxiomR: \forall e: E(Rcv). \exists e': E(Send). (e' < e) \wedge Rcv(e) = Send(e').$$
$$AxiomV: \forall e: E(Verify). \exists e': E(Sign). (e' < e) \wedge Verify(e) = Sign(e').$$

$AxiomD$ 与 $AxiomR$ 和 $AxiomV$ 相似,解密事件主体在事件发生前收到一个与之密钥匹配、其他信息相同的加密事件,具体表示为

$$\forall e: E(Decrypt). \exists e': E(Encrypt).$$
$$e' < e \wedge DEMatch(e, e') DEMatch(e, e') \equiv_{def}$$
$$plaintext(e) = plaintext(e') \wedge$$
$$ciphertext(e) = ciphertext(e') \wedge$$
$$MatchingKeys(key(e); key(e')).$$

4) 不相交公理(disjointness axioms)

2 个原子间不相交假设主要考虑 2 方面.一方面任意这 7 个事件类不在其他类中,即:

$$ActionDisjoint: \exists f: E \rightarrow \mathbb{Z}. \forall e: E.$$
$$(e \in E(New) \Rightarrow f(e)=1) \wedge$$
$$(e \in E(Send) \Rightarrow f(e)=2) \wedge \cdots \wedge$$
$$(e \in E(Decrypt) \Rightarrow f(e)=7).$$

另一方面主体生成的随机数 n 与主体所持有的私钥、签名或密文不相同,且三者间不相交[18].签名可以是明文通过散列加密生成,而密文是明文加密生成,$Data$ 类型成员在散列加密后不等同于 $Data$ 类型成员,那么签名不等价于密文,具体表示为

$$\forall n: E(New). \forall s: E(Sign). \forall e: E(Encrypt). \forall A:$$
$$Id. New(n) \neq signature(e) \wedge New(n) \neq$$
$$ciphertext(e) \wedge New(n) \neq Private(A) \wedge$$
$$ciphertext(e) \neq Private(A) \wedge$$
$$signature(s) \neq Private(A) \wedge$$
$$signature(s) \neq ciphertext(e).$$

5) 流关系(flow relation)

流关系是随机数因果序事件间的关联. Act 类型

包含 7 个事件类,记作 $actions$.(e has a)为真当且仅当 $action$ 类型的 e 中包含 a,具体表示为

$$e \text{ has } a \equiv_{\text{def}} (e \in E(New) \wedge New(e) \text{ has } a) \vee (e \in E(Send) \wedge Send(e) \text{ has } a) \vee (e \in E(Rcv) \wedge Rcv(e) \text{ has } a) \vee \cdots$$

流关系 $e_1 \xrightarrow{a} e_2$ 表示 a 从动作 e_1 流向动作 e_2,包含 3 种情况.

1) a 从动作 e_1 到动作 e_2 发生在同一主体上;

2) 介于发送事件和接收事件间以明文发送 a;

3) a 在加密事件明文中,密文流向一个与之匹配的解密事件[18],具体流关系递归为

$$e_1 \xrightarrow{a} e_2 =_{\text{rec}} (e_1 \text{ has } a \wedge e_2 \text{ has } a \wedge e_1 \leqslant_{\text{loc}} e_2) \vee \left(\begin{array}{l} \exists s:E(Send).\exists r:E(Rcv).e_1 \leqslant s < r \leqslant e_2 \wedge \\ Send(s) = Rcv(r) \wedge e_1 \xrightarrow{a} s \wedge r \xrightarrow{a} e_2 \end{array} \right) \vee \left(\begin{array}{l} \exists e:E(Encrypt).\exists d:E(Decrypt).e_1 \leqslant e < \\ d \leqslant e_2 \wedge DEMatch(d,e) \wedge key(d) \neq \\ Symma(a) \wedge e_1 \xrightarrow{a} e \wedge e \xrightarrow{ciphertext} d \wedge d \xrightarrow{a} e_2 \end{array} \right).$$

引理 1. 如果 $e_1 \xrightarrow{a} e_2$,那么 $e_1 \leqslant e_2$ 且 e_2 has a.

用 $\rightsquigarrow$ 表示流关系动作,若第 1 个动作是加密,第 2 个动作包含前一个动作的密文,则:

$$e' \rightsquigarrow e \equiv_{\text{def}} e' \in Encrypt \wedge e \text{ has } ciphertext(e)'.$$

引理 2. 如果 $e_1 \xrightarrow{a} e_2$,那么 $release(a, e_1, e_2)$.

6) 随机数公理(nonce axiom, $AxiomF$)

随机数公理记为 $AxiomF$,包含 3 部分 $AxiomF_1$, $AxiomF_2$, $AxiomF_3$,其中 $AxiomF_1$ 关于流性质,应用在事件关联的随机数中,具体为

$$AxiomF_1: \forall e_1:E(New).$$

$$\forall e_2:E.e_2 \text{ has } New(e_1) \Rightarrow e_1 \xrightarrow{New(e_1)} e_2.$$

$AxiomF_2$, $AxiomF_3$ 介绍签名、密文以及包含 2 类事件的相关关系,没有规定签名或密文与特殊事件有关.如果一个动作包含签名或密文,则可以推断出具有相同信息的签名或加密动作[10],具体为

$$AxiomF_2: \forall e_1:E(Sign).\forall e_2:E.e_2 \text{ has } signature(e_1) \Rightarrow \exists e':E(Sign).$$

$$Sign(e') = Sign(e_1) \wedge e' \xrightarrow{signature(e_1)} e_2.$$

$$AxiomF_3: \forall e_1:E(Encrypt).\forall e_2:E.e_2 \text{ has } ciphertext(e_1) \Rightarrow \exists e':E(Encrypt).$$

$$Encrypt(e') = Encrypt(e_1) \wedge e' \xrightarrow{ciphertext(e_1)} e_2.$$

2.2.2 事件逻辑引理及性质

基本序列是协议动作的参数列表,参数是主体标识符,由 2 个及以上组成[9].遵守协议的主体参与到多个线程,线程是协议基本序列实例且遵守协议[10].

引理 3. 随机数引理.

协议 $Protocol(bss)$ 是合法的,主体 A 是诚实主体且遵守 Pr,线程 thr 是基本序列 bss 的一个实例,$n = thr[j]$, $n \in E(New)$, $e = thr[i]$ 和 $j < i$[10].如果 j 和 i 之间不存在 k, $E(Send)$ 中存在事件 $thr[k]$,则随机数 $New(n)$ 不会在事件 e 发生之前释放[10].

在协议强认证证明过程中,规定被证明协议合法,线程 thr_1 相邻事件 e_0, e_1 不存在发送动作(或任何的动作发生),由随机数引理可得,事件 e_1 发生前随机数不会被释放.

安全协议形式化分析过程中,诚实主体在执行相关动作时需要满足 5 个性质[19]:

性质 1. 多组合信息交互.

诚实主体 A 本身持有可信第三方(trusted third party, TTP)授权且遵守协议 $Protocol(bss)$,在认证过程中需要通过 TTP 授权进行验证,这属于诚实主体自身行为,即:

$$\forall A:Id.\forall e,e':E(e) \in TTP \wedge (e < e') \Rightarrow \forall e':A \mid\!= E[e].$$

在验证协议交互过程安全性问题中可以省略该证明,从而降低证明过程中的复杂度.

性质 2. 不叠加.

协议分析过程中,对于在匹配会话中已经验证过的动作,在新一轮的验证过程中可以直接引用验证结果来减少冗余.

$$\forall A:Id.\forall e_1,e_2:e_1 < e_2 \Rightarrow \forall e_2:A \mid\!= E[e_1].$$

性质 3. 事件匹配.

在 7 种事件类中,在遵守协议 $Protocol(bss)$ 的前提下发起者主体 A、响应者主体 B(事件响应者可以不是相同的主体)必须参与的事件类是双方的或者多方的,从而保障事件发生的有效性.

$$\forall A,B:(A \neq B).\forall e_1,e_2:((e_1 \in A, e_2 \in B) \wedge (e_1 < e_2)) \vee Send(e_1) = Rcv(e_2) \vee Sign(e_1) = Verify(e_2) \vee Decrypt(e_1) = Encrypt(e_2).$$

性质 4. 去重复性.

在验证事件匹配的过程中,如果出现多个事件需要同时进行验证,则依据从上到下的原则进行验证来减少验证过程中的重复操作.

性质 5. 去未来性.

在考虑匹配动作的过程中,以当前已发生事件

为基准,对于之后没有发生的动作不予考虑来减少验证过程中的不必要进程.

2.3 形式化方法描述协议

1) 线程

线程是动作在单个位置的有序列表,满足:

$$Thread \equiv_{\text{def}} \{thr : ActList \mid \forall i : thr[i] <_{\text{loc}} thr[i+1]\}.$$

$thr_1 \preccurlyeq thr_2$ 表示线程 thr_1 是线程 thr_2 前面发生的一个相邻线程,定义 $thr_1 \simeq thr_2$ 为

$$thr_1 \preccurlyeq thr_2 \vee thr_2 \preccurlyeq thr_1.$$

线程消息是线程中发送和接收动作的集合,即:

$$isMsg(e) \equiv_{\text{def}} e \in E(Send) \vee e \in E(Rcv),$$

$$messages(thr) \equiv_{\text{def}} filter(isMsg, thr).$$

对于消息 s 和 r,s 是发送消息,r 是接收消息,s 和 r 间传递消息相同,则 2 条信息间是一个弱匹配关系,即 $s \sim r$;如果 s 与 r 之间有直接因果关系,s 发生在 r 之前,则 s 和 r 之间是一个强匹配关系,即为 $s \mapsto r$,具体为

$$s \sim r \equiv_{\text{def}} s \in E(Send) \wedge r \in E(Rcv) \wedge Send(s) = Rcv(r),$$

$$s \mapsto r \equiv_{\text{def}} s \sim r \wedge s < r.$$

2) 基本序列(basic sequence)

基本序列是协议动作的基本参数列表,在协议主体 A,B 参数是标示符,主体 A 遵守协议,参与协议多个线程,线程是协议中的实例,与主体 B 在不同位置进行交互[19].事件逻辑中所研究的协议允许多个主体参与.

基本序列是 2 个位置与一个线程的联系,当线程是基本序列给定的位置参数,则这个关系为真,基本序列类型成员为

$$Basic \equiv_{\text{def}} Id \rightarrow Id \rightarrow Thread \rightarrow P.$$

注:其中 P 代表命题(proposition),在命题逻辑结构中跟布尔运算是不同的.

基本序列实例可以生成随机数、签名等,参数原子是在序列在关系中量化存在.线程 thr 是主体 A 中已知基本序列关系 bss,记作 $thr = oneof(bss, A)$.关系 $inoneof(e, thr, bss, A)$ 作为协议形式化定义,记作:$e \in thr \wedge thr = oneof(bss, A)$.

3) 匹配会话及协议动作

① 匹配会话

线程 thr_1 和 thr_2 构成长度为 n 的匹配会话,如果它们至少包含 n 个消息,当每个线程中前 n 个消息成对,每对 $\langle m_1, m_2 \rangle$ 满足 $m_1 \mapsto m_2 \vee m_2 \mapsto m_1$,则构成强匹配会话,即为 $thr_1 \overset{n}{\approx} thr_2$[10].如果每对 $\langle m_1, m_2 \rangle$ 只满足 $m_1 \sim m_2 \vee m_2 \sim m_1$,得到一个弱匹配会话,记作 $thr_1 \overset{n}{\sim} thr_2$.

② 协议动作

使用基本协议动作描述协议 $ProtocolAction$ 类来定义协议动作.所有成员类包含标签 tag 和值 $value$,$ProtocolAction$ 类成员包含 7 种动作,定义如下:

$$\{New(a) \mid a \in Atom\};$$

$$\{Send(x) \mid x \in Data\};$$

$$\{Rcv(x) \mid x \in Data\};$$

$$\{Sign(t) \mid t \in Data \times Id \times Atom\};$$

$$\{Verify(t) \mid t \in Data \times Id \times Atom\};$$

$$\{Encrypt(t) \mid t \in Data \times Key \times Atom\};$$

$$\{Decrypt(t) \mid t \in Data \times Key \times Atom\}.$$

协议动作 pa(protocol action)对应事件 e,则:

$$e \in E(New) \wedge pa = new(New(e)) \vee e \in E(Send) \wedge pa = send(Send(e)) \vee \cdots$$

4) 事件逻辑方法描述协议

① 定义协议

事件逻辑理论使用 $Basic$ 类型的基本序列关系表 bss 定义一个协议[10].协议是在存储位置的断言,类型为 $Id \rightarrow P$.

协议 $Protocol(bss)$ 定义为

$$\lambda A. \forall e : Act. loc(e) = A \Rightarrow (\exists thr. inOneof(e, thr, bss, A)) \wedge \forall thr_1, thr_2. (inOneof(e, thr_1, bss, A) \wedge inOneof(e, thr_1, bss, A)) \Rightarrow thr_1 \simeq thr_2.$$

主体 A 动作是基本序列的实例成员,如果动作是一个或多个实例成员,则实例是兼容的.

② 认证

主体 A,B 为诚实主体且遵守协议,主体 A 发起一个会话序列,主体 B 的一个应答序列与之构成合法匹配会话.类似地,如果主体 B 执行全应答序列的一个实例,那么有一个和主体 A 的匹配会话,即接收消息和相匹配的发送消息内容一致[10].消息双方完成身份认证保证不会有攻击者借助之前拦截消息进行伪装会话攻击,匹配会话过程满足强匹配,协议 Pr 中基本序列 bs 认证 n 个消息,认证过程满足:

$$pr \models auth(bs, n) \equiv_{\text{def}} \forall A, B. \forall thr_1. (Honest(A) \wedge Honest(B) \wedge Pr(A) \wedge Pr(B) \wedge A \neq B \wedge loc(thr_1) = A \wedge bs(A, B, thr_1)) \Rightarrow \exists thr_2. loc(thr_2) = B \wedge thr_1 \overset{n}{\approx} thr_2.$$

2.4 安全协议证明过程

事件逻辑理论描述协议强认证性质,具体如图 5 所示:

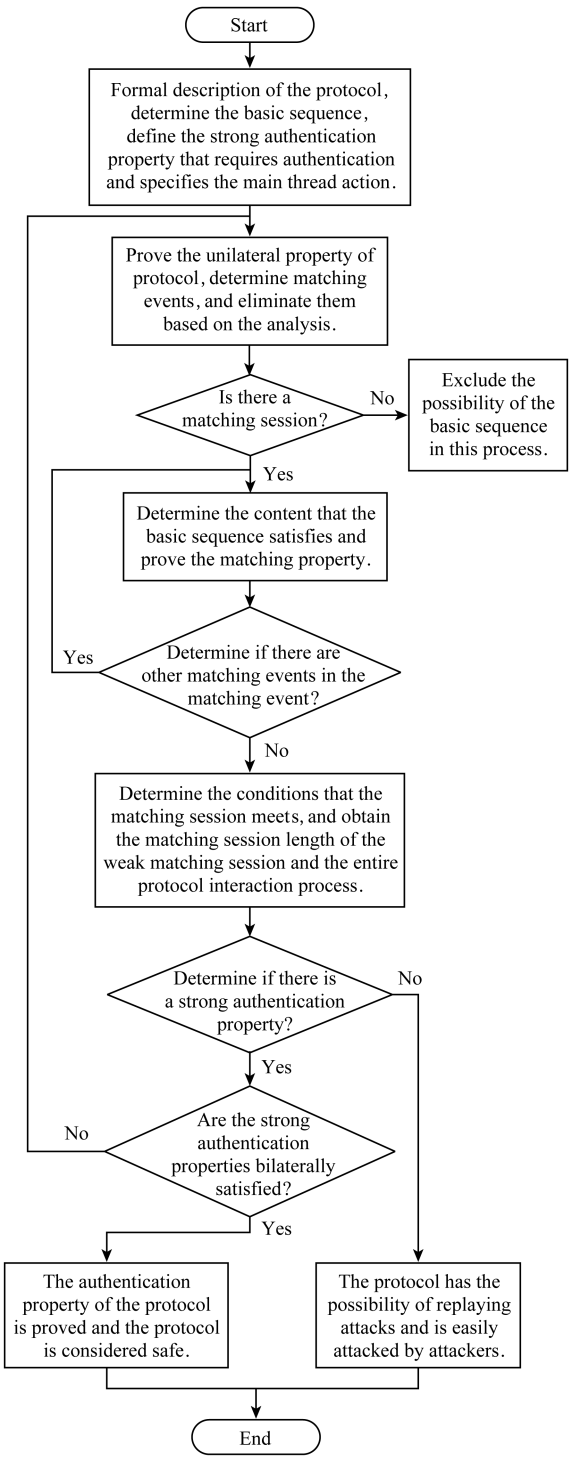

Fig. 5 The flowchart of event logic method which proves the protocol authentication

图 5 事件逻辑方法证明协议认证性流程图

1) 定义基本序列

利用事件逻辑理论方法对安全协议进行形式化描述,定义发起者、响应者序列的具体动作[9],规范协议基本序列,确认满足协议安全性需要证明的强认证性质.

2) 对强认证性质的单向进行证明

规定主体不同是诚实的且遵守协议,假设某一线程为协议基本序列的实例,定义线程上的动作,确定协议匹配事件.分析匹配事件,确认是否存在匹配会话以及匹配会话内部是否含有需要进一步证明的匹配事件.

3) 确定匹配事件,进行排除分析

查询相关匹配事件是否符合匹配会话,如果符合则进行由内而外对相关匹配会话的证明;如果不符合则进行下一轮匹配事件的筛选证明,确认整个匹配事件是否满足弱匹配.

4) 确认强认证性质

确认匹配会话属于弱匹配,分析协议交互过程的匹配会话长度,根据相关公理确认强匹配会话.

5) 单向证明成立后,进行双向强认证性质证明

如果证明成立,则说明协议是安全的;在整个证明过程中,如果一边的匹配事件不能满足弱匹配,说明协议同样不满足强认证性质,在认证阶段不能达成双向身份认证,协议易被攻击者伪装身份进行攻击,存在消息重放的可能,协议主体不安全[10].

以流程图的方法简化事件逻辑理论证明协议安全性.

3 事件逻辑理论证明 WMN 客户端与 LTCA 认证协议

通过事件逻辑理论对 WMN 客户端与 LTCA 双向认证协议进行基本序列排序,定义 I_1, I_2, I_3 为发起方在协议交互过程中产生的基本序,R_1, R_2, R_3 为协议响应者在协议交互过程中产生的基本序.LTCA 作为协议交互过程中动作发起方(initiator),用户 MU 作为协议交互过程中动作响应方(responder),协议具体描述如图 6 所示.

基于事件逻辑理论定义协议满足安全性需要满足的身份认证性质,验证 WMN 客户端与 LTCA 协议认证性[18],协议基本序列如图 7 所示.

根据协议中定义的基本序列,定义 WMN 客户端与 LTCA 间双向认证协议 $Protocol([I_1, I_2, I_3, R_1, R_2, R_3])$.

协议需要验证的强认证性质为

$$Nse \models auth(I_3, 2) \wedge Nse \models auth(R_3, 3).$$

1) 对 $Nse \models auth(I_3, 2)$进行证明

证明. 假设诚实主体 MU $\neq$ LTCA 且遵守 WMN 客户端与 LTCA 双向认证协议,线程 thr_1 是

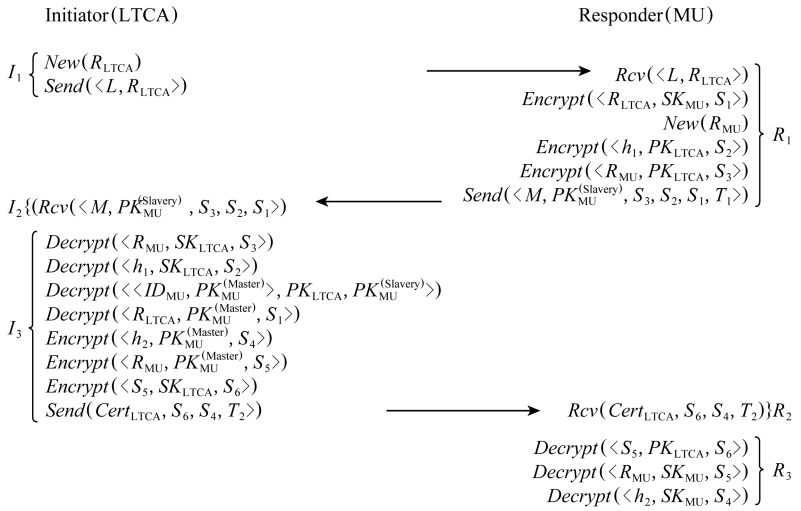

Fig. 6 Basic sequence description of two-way authentication

图 6 双向认证基本序列描述

$$\begin{cases} I_1 = New(R_{LTCA}), Send(\langle L, R_{LTCA}\rangle) \\ I_2 = I_1, Rcv(\langle M, PK_{MU}^{(Slavery)}, S_3, S_2, S_1\rangle) \\ I_3 = I_2, Decrypt(\langle R_{MU}, SK_{LTCA}, S_3\rangle), \\ \quad Decrypt(\langle h_1, SK_{LTCA}, S_2\rangle), \\ \quad Decrypt(\langle\langle ID_{MU}, PK_{MU}^{(Master)}\rangle, PK_{LTCA}, PK_{MU}^{(Slavery)}\rangle), \\ \quad Decrypt(\langle R_{LTCA}, PK_{MU}^{(Master)}, S_1\rangle), \\ \quad Encrypt(\langle h_2, PK_{MU}^{(Master)}, S_4\rangle), \\ \quad Encrypt(\langle R_{MU}, PK_{MU}^{(Master)}, S_5\rangle), Encrypt(\langle S_5, SK_{LTCA}, S_6\rangle) \\ \quad Send(Cert_{LTCA}, S_6, S_4, T_2) \\ R_1 = Rcv(\langle L, R_{LTCA}\rangle), Encrypt(\langle R_{LTCA}, SK_{MU}, S_1\rangle), \\ \quad New(R_{MU}), Encrypt(\langle h_1, PK_{LTCA}, S_2\rangle), \\ \quad Encrypt(\langle R_{MU}, PK_{LTCA}, S_3\rangle), \\ \quad Send(\langle M, PK_{MU}^{(Slavery)}, S_3, S_2, S_1, T_1\rangle) \\ R_2 = R_1, Rcv(Cert_{LTCA}, S_6, S_4, T_2) \\ R_3 = R_2, Decrypt(\langle S_5, PK_{LTCA}, S_6\rangle), \\ \quad Decrypt(\langle R_{MU}, SK_{MU}, S_5\rangle), \\ \quad Decrypt(\langle h_2, SK_{MU}, S_4\rangle) \end{cases}$$

Fig. 7 Basic sequence of two-way authentication protocol

图 7 双向认证协议的基本序列

基本序 I_3 的实例,$e_0 <_{loc} e_1 <_{loc} \cdots <_{loc} e_6$ 为线程 thr_1 上的动作,那么事件 $e_0, e_1, \cdots, e_6$ 发生的主体为 LTCA. 对原子 R_{LTCA}, SK_{LTCA}, $PK_{MU}^{(Slavery)}$, $PK_{MU}^{(Master)}$, ID_{MU}, R_{MU}, S_3, S_2, S_1, h_1 则有:

$$\begin{pmatrix} New(e_0) = \langle R_{LTCA}\rangle \wedge Send(e_1) = \langle L, R_L\rangle \wedge \\ Rcv(e_2) = \langle M, PK_{MU}^{(Slavery)}, S_3, S_2, S_1\rangle \wedge \\ Decrypt(e_3) = \langle R_{MU}, SK_{LTCA}, S_3\rangle \wedge \\ Decrypt(e_4) = \langle h_1, SK_{LTCA}, S_2\rangle \wedge Decrypt(e_5) = \\ \langle\langle ID_M, PK_{MU}^{(Master)}\rangle, PK_{LTCA}, PK_{MU}^{(Slavery)}\rangle \wedge \\ Decrypt(e_6) = \langle R_{LTCA}, PK_{MU}^{(Master)}, S_1\rangle \end{pmatrix} \tag{1}$$

由 $AxiomD$ 以及 $AxiomS$ 可知,存在事件 e', e'', e''', e'''' 使得主体 MU 同时满足 4 个条件:

① 事件 e' 中 $e' < e_3 \wedge DEMatch(e_3, e') \wedge loc(e') = M$ 成立;

② 事件 e'' 中 $e'' < e_4 \wedge DEMatch(e_4, e'') \wedge loc(e'') = M$ 成立;

③ 事件 e''' 中 $e''' < e_5 \wedge DEMatch(e_5, e''') \wedge loc(e''') = M$ 成立;

④ 事件 e'''' 中 $e'''' < e_6 \wedge DEMatch(e_6, e'''') \wedge loc(e'''') = M$ 成立.

对于事件发生的主体满足:

$$Encrypt(e') = \langle R_{MU}, PK_{LTCA}, S_3\rangle.$$

$$Encrypt(e'') = \langle h_1, PK_{LTCA}, S_2\rangle.$$

$$Encrypt(e''') = \langle\langle ID_{MU}, PK_{MU}^{(Master)}\rangle, PK_{LTCA}, PK_{MU}^{(Slavery)}\rangle.$$

$$Encrypt(e'''') = \langle R_{LTCA}, SK_{MU}, S_1\rangle.$$

根据性质 4,以事件 e' 为研究对象,因为主体 MU 遵守 WMN 客户端和 LTCA 双向认证协议,事件 e' 必然为 WMN 客户端与 LTCA 双向认证协议的基本序列之中的一个实例成员[10].在协议交互过程中包含加密 $Encrypt()$ 动作的是 I_3, R_1, R_2, R_3,根据性质 5 以及图 6 可知,MU 基本序 R_2, R_3 发生在基本序 I_3 后,故基本序 R_2, R_3 可以排除.

若 e' 是基本序 R_1 实例,根据置换规则定义 A 替换 LTCA,主体 A 满足 LTCA 的性质.主体 A 和原子 R_A, $PK_{A,1}$ $S_{1,1}$ $S_{2,1}$ $S_{3,1}$ $R_{MU,1}$ $h_{1,1}$ $PK_{MU}^{(Slavery)}$, M_1,在主体 MU 上存在事件 $e'_0, e'_1, e'_2, e'_3, e'_4, e'_5$,如此得到:

$$\begin{pmatrix} e_0'<_{\mathrm{loc}}e'<_{\mathrm{loc}}e_2'<_{\mathrm{loc}}e_3'<_{\mathrm{loc}}e_4'<_{\mathrm{loc}}e_5' \wedge \\ Rcv(e_0')=\langle A,R_A\rangle \wedge Encrypt(e_1')= \\ \langle R_A,SK_{\mathrm{MU},1}S_1\rangle \wedge New(e_2')={}_1R_{\mathrm{MU}} \wedge \\ Encrypt(e_3')=\langle {}_1h_1,PK_{A,1}S_2\rangle \wedge \\ Encrypt(e_4')=\langle {}_1R_{\mathrm{MU}},PK_{A,1}S_3\rangle \wedge \\ Send(e_5')=\langle M,{}_1PK_{\mathrm{MU}}^{(\mathrm{Slavery})},{}_1S_{1,1}S_{2,1}S_3,T_1\rangle \end{pmatrix} \tag{2}$$

在式(2)中,加密事件有 e_1',e_3',e_4',3 个加密事件分别进行匹配,可知 3 个加密事件 e_1',e_3',e_4' 满足:

$$Encrypt(e_4')=\langle {}_1R_{\mathrm{MU}},PK_{A,1}S_3\rangle=\langle R_{\mathrm{MU}},PK_{\mathrm{LTCA}},S_3\rangle=Encrypt(e');$$
$$Encrypt(e_3')=\langle {}_1h_1,PK_{A,1}S_2\rangle=\langle h_1,PK_{\mathrm{LTCA}},S_2\rangle=Encrypt(e'');$$
$$Encrypt(e_1')=\langle R_A,SK_{\mathrm{MU},1}S_1\rangle=\langle R_{\mathrm{LTCA}},SK_{\mathrm{MU}},S_1\rangle=Encrypt(e''').$$

在主体中,SK_{LTCA} 和 PK_{LTCA} 分别为 LTCA 的私钥和主公钥,符合 $AxiomD$ 定义构成匹配,则主体满足:

$${}_1R_{\mathrm{MU}}=R_{\mathrm{MU}},PK_A=PK_{\mathrm{LTCA},1}S_3=S_3,A=L,$$
$${}_1h_1=h_{1,1}S_2=S_{2,1}S_1=S_1,R_A=R_{\mathrm{LTCA}}.$$

可得到:

$$\begin{pmatrix} e_0'<_{\mathrm{loc}}e''''<_{\mathrm{loc}}e_2'<_{\mathrm{loc}}e''<_{\mathrm{loc}}e'<_{\mathrm{loc}}e_5' \wedge \\ Rcv(e_0')=\langle L,R_{\mathrm{LTCA}}\rangle \wedge Encrypt(e'''')= \\ \langle R_{\mathrm{LTCA}},SK_{\mathrm{MU}},S_1\rangle \wedge New(e_2')=R_{\mathrm{MU}} \wedge \\ Encrypt(e'')=\langle h_1,PK_{\mathrm{LTCA}},S_2\rangle \wedge \\ Encrypt(e')=\langle R_{\mathrm{MU}},PK_{\mathrm{LTCA}},S_3\rangle \wedge \\ Send(e_5')=\langle M_1,PK_{\mathrm{MU}}^{(\mathrm{Slavery})},S_1,S_2,S_3,T_1\rangle \end{pmatrix} \tag{3}$$

在式(3)中可知加密事件与基本序 R_1 中加密事件内容一致,且用户 $A=$LTCA.由 $AxiomD$ 以及 $AxiomS$ 可知,在基本序 I_3 中,事件 e_5 验证 $PK_{\mathrm{MU}}^{(\mathrm{Slavery})}$,其中加密解密中 SK_{LTCA} 与 PK_{LTCA} 相匹配.对于事件 e''' 在基本序 R_1 有对应加密事件,从而满足:

$${}_1PK_{\mathrm{MU}}^{(\mathrm{Slavery})}=PK_{\mathrm{MU}}^{(\mathrm{Slavery})}.$$

可得到:

$$\begin{pmatrix} e_0'<_{\mathrm{loc}}e''''<_{\mathrm{loc}}e_2'<_{\mathrm{loc}}e''<_{\mathrm{loc}}e'<_{\mathrm{loc}}e''' \wedge \\ Rcv(e_0')=\langle L,R_{\mathrm{LTCA}}\rangle \wedge Encrypt(e'''')= \\ \langle R_{\mathrm{LTCA}},SK_{\mathrm{MU}},S_1\rangle \wedge New(e_2')=R_{\mathrm{MU}} \wedge \\ Encrypt(e'')=\langle h_1,PK_{\mathrm{LTCA}},S_2\rangle \wedge \\ Encrypt(e')=\langle R_{\mathrm{MU}},PK_{\mathrm{LTCA}},S_3\rangle \wedge \\ Send(e''')=\langle M,PK_{\mathrm{MU}}^{(\mathrm{Slavery})},S_1,S_2,S_3,T_1\rangle \end{pmatrix} \tag{4}$$

由线程 thr_1 的初始式(1)可知,基本序 I_3 内可以构成完整($Send$,Rcv)的是事件 e_1 和 e_2.根据事件逻辑的认证规则,结合式(1)以及式(4)可得,$Send(e_1)=\langle L,R_{\mathrm{LTCA}}\rangle=Rcv(e_0')$ 和 $Rcv(e_2)=\langle M,PK_{\mathrm{MU}}^{(\mathrm{Slavery})},S_1,S_2,S_3,T_1\rangle=Send(e_5')$,得到一个长度为 2 的(弱)匹配.

协议满足强匹配会话则需证明 $e_1<e_0'$,$e_5'<e_2$,就事件 e_1,e_0' 进行举例说明.在初始阶段规定参与会话的双方用户 MU 和 LTCA 遵守 WMN 认证协议且 MU≠LTCA,根据 $AxiomF$、流关系,事件 e_1,e_0' 之间存在发送事件 s 释放散列数 h_1.如果 $e_1\leqslant s$,得到排序为 $e_1<e_0'$,再排除 $e_0<_{\mathrm{loc}}s<_{\mathrm{loc}}e_1$.如果 $e_0<_{\mathrm{loc}}s<_{\mathrm{loc}}e_1$,那么 s 为 A 的一些其他线程成员,由引理 1 可知线程 thr_1 中事件 e_0 和 e_1 之间不存在发送动作,则散列数 h_1 在 e_1 之前不会释放,是 $AxiomF$ 证明 $e_1<e_0'$ 所需要的部分.

综上所述,根据 $AxiomF$ 和引理 1 证明 $e_5'<e_2$.根据证明过程可知,主体 A 中的线程 thr_1 在协议存在 2 个完整交互的强匹配.

$$Nse\models auth(I_3,2).$$ 证毕.

2) 对 $Nse\models auth(R_3,3)$进行证明

证明.继承证明 1)对诚实主体的规定(MU≠LTCA),假设线程 thr_2 是基本序 R_3 实例,$e_0<_{\mathrm{loc}}e_1<_{\mathrm{loc}}\cdots<_{\mathrm{loc}}e_9$ 是线程 R_3 上的动作,在基本序 R_3 中,事件 $e_0,e_1,\cdots,e_9$ 发生主体为用户 MU.原子 S_1,S_2,S_3,S_4,S_5,S_6,R_{LTCA},R_{MU},PK_{LTCA},$PK_{\mathrm{MU}}^{(\mathrm{Slavery})}$,$SK_{\mathrm{MU}}$,$h_1$,$h_2$,$T_1$,$T_2$ 满足:

$$\begin{pmatrix} Rcv(e_0)=\langle L,R_{\mathrm{LTCA}}\rangle \wedge Encrypt(e_1)= \\ \langle R_{\mathrm{LTCA}},SK_{\mathrm{MU}},S_1\rangle \wedge New(e_2)=R_{\mathrm{MU}} \wedge \\ Encrypt(e_3)=\langle h_1,PK_{\mathrm{LTCA}},S_2\rangle \wedge \\ Encrypt(e_4)=\langle R_{\mathrm{MU}},PK_{\mathrm{LTCA}},S_3\rangle \wedge \\ Send(e_5)=\langle M,PK_{\mathrm{MU}}^{(\mathrm{Slavery})},S_1,S_2,S_3,T_1\rangle \wedge \\ Rcv(e_6)=\langle Cert_{\mathrm{LTCA}},S_6,S_4,T_2\rangle \wedge \\ Decrypt(e_7)=\langle S_5,PK_{\mathrm{LTCA}},S_6\rangle \wedge \\ Decrypt(e_8)=\langle R_{\mathrm{MU}},SK_{\mathrm{MU}},S_5\rangle \wedge \\ Decrypt(e_8)=\langle h_2,SK_{\mathrm{MU}},S_4\rangle \end{pmatrix} \tag{5}$$

由 $AxiomD$ 以及 $AxiomS$ 可知,主体 LTCA 满足存在事件 e'''',e'''''',e''''''' 使得加密解密事件匹配成立,即:

$$\begin{Bmatrix} e''''<e_7 \wedge DEMatch(e'''',e_7) \wedge loc(e'''')=L \\ e''''''\langle e_8 \wedge DEMatch(e'''''',e_8) \wedge loc(e'''''')=L \\ e'''''''\langle e_9 \wedge DEMatch(e''''''',e_9) \wedge loc(e''''''')=L \end{Bmatrix}.$$

则事件发生主体满足:

$$\begin{cases} Encrypt(e''''')=\langle S_5,SK_{\mathrm{LTCA}},S_6\rangle; \\ Encrypt(e''''')=\langle R_{\mathrm{MU}},PK_{\mathrm{MU}}^{(\mathrm{Master})},S_5\rangle; \\ Encrypt(e''''')=\langle h_2,PK_{\mathrm{MU}}^{(\mathrm{Master})},S_4\rangle. \end{cases}$$

根据性质 4,以事件e''''为研究对象,因为主体 LTCA 遵守 WMN 客户端和 LTCA 双向认证协议,事件e''''必然为认证协议的基本序列的实例成员.在协议交互过程中包含加密 $Encrypt(\,)$动作的是 I_3,R_1,R_2,R_3,根据性质 3 以及图 6 可知,对于基本序 R_1,R_2 可以进行排除.

如果事件e''''是基本序 I_3 的实例,根据置换规则定义 B 替换 MU,主体 B 满足 MU 的相关性质.主体 B 和原子${}_2R_{\mathrm{LTCA}}$,$PK_B^{(\mathrm{Slavery})}$,$PK_B^{(\mathrm{Master})}$,$ID_B$,$R_B$,${}_2S_6$,${}_2S_5$,${}_2S_4$,${}_2S_3$,${}_2S_2$,${}_2S_1$,${}_2h_1$,${}_2h_2$,在 LTCA 上存在事件 e''_0,e''_1,e''_2,e''_3,e''_4,e''_5,e''_6,e''_7,e''_8,e''_9,e''_{10},如此得:

$$\begin{pmatrix} e''_0<_{\mathrm{loc}}e''_1<_{\mathrm{loc}}e''_2<_{\mathrm{loc}}e''_3<_{\mathrm{loc}}e''_4<_{\mathrm{loc}}e''_5 \\ <_{\mathrm{loc}}e''_6<_{\mathrm{loc}}e''_7<_{\mathrm{loc}}e''_8<_{\mathrm{loc}}e''_9<_{\mathrm{loc}}e''_{10}\wedge \\ New(e''_0)=\langle{}_2R_{\mathrm{LTCA}}\rangle\wedge Send(e''_1)= \\ \langle L,{}_2R_{\mathrm{LTCA}}\rangle\wedge Rcv(e''_2)=\langle B, \\ PK_B^{(\mathrm{Slavery})},{}_2S_3,{}_2S_2,{}_2S_1\rangle\wedge \\ Decrypt(e''_3)=\langle R_B,SK_{\mathrm{LTCA}},{}_2S_3\rangle\wedge \\ Decrypt(e''_4)=\langle{}_2h_1,SK_{\mathrm{LTCA}},{}_2S_2\rangle\wedge \\ Decrypt(e''_5)=\langle\langle ID_B,PK_B^{(\mathrm{Master})}\rangle,PK_{\mathrm{LTCA}}, \\ PK_B^{(\mathrm{Slavery})}\rangle\wedge Decrypt(e''_6)= \\ \langle{}_2R_{\mathrm{LTCA}},PK_B^{(\mathrm{Master})},{}_2S_1\rangle\wedge Encrypt(e''_7)=\langle{}_2h_2, \\ PK_B^{(\mathrm{Master})},{}_2S_4\rangle\wedge Encrypt(e''_8)=\langle R_B,PK_B^{(\mathrm{Master})}, \\ {}_2S_5\rangle\wedge Encrypt(e''_9)=\langle{}_2S_5,SK_{\mathrm{LTCA}},{}_2S_6\rangle\wedge \\ Send(e''_{10})=\langle Cert_{\mathrm{LTCA}},{}_2S_6,{}_2S_4,T_2\rangle \end{pmatrix} \tag{6}$$

在式(6)中,加密事件有 e''_7,e''_8,e''_9,3 个加密事件分别进行匹配,可知 3 个加密事件 e''_7,e''_8,e''_9 满足:

$$\begin{gathered} Encrypt(e''_7)=\langle{}_2h_2,PK_B^{(\mathrm{Master})},{}_2S_4\rangle= \\ \langle h_2,PK_{\mathrm{MU}}^{(\mathrm{Master})},S_4\rangle=Encrypt(e''''), \\ Encrypt(e''_8)=\langle R_B,PK_B^{(\mathrm{Master})},{}_2S_5\rangle= \\ \langle R_{\mathrm{MU}},PK_{\mathrm{MU}}^{(\mathrm{Master})},S_5\rangle=Encrypt(e'''''), \\ Encrypt(e''_9)=\langle{}_2S_5,SK_{\mathrm{LTCA}},{}_2S_6\rangle= \\ \langle S_5,SK_{\mathrm{LTCA}},S_6\rangle=Encrypt(e''''). \end{gathered}$$

在主体中,SK_{MU}和 $PK_{\mathrm{MU}}^{(\mathrm{Master})}$分别为 MU 的私钥和主公钥,$SK_{\mathrm{LTCA}}$和 PK_{LTCA}分别为 LTCA 私钥和公钥,符合 $AxiomD$ 定义构成匹配,则主体满足:

$$\begin{gathered} {}_2h_2=h_2,PK_B^{(\mathrm{Master})}=PK_{\mathrm{MU}}^{(\mathrm{Master})},B=M, \\ {}_2S_4=S_{4,2}S_5=S_{5,2}S_6=S_6,R_B=R_{\mathrm{MU}}. \end{gathered}$$

可得到:

$$\begin{pmatrix} e''_0<_{\mathrm{loc}}e''_1<_{\mathrm{loc}}e''_2<_{\mathrm{loc}}e''_3<_{\mathrm{loc}}e''_4<_{\mathrm{loc}}e''_5 \\ <_{\mathrm{loc}}e''_6<_{\mathrm{loc}}e'''''<_{\mathrm{loc}}e'''''<_{\mathrm{loc}}e''''<_{\mathrm{loc}}e''_{10}\wedge \\ New(e''_0)=\langle{}_2R_{\mathrm{LTCA}}\rangle\wedge Send(e''_1)=\langle L,{}_2R_{\mathrm{LTCA}}\rangle\wedge \\ Rcv(e''_2)=\langle M,PK_B^{(\mathrm{Slavery})},{}_2S_3,{}_2S_2,{}_2S_1\rangle\wedge \\ Decrypt(e''_3)=\langle R_{\mathrm{MU}},SK_{\mathrm{LTCA}},{}_2S_3\rangle\wedge \\ Decrypt(e''_4)=\langle{}_2h_1,SK_{\mathrm{LTCA}},{}_2S_2\rangle\wedge \\ Decrypt(e''_5)=\langle\langle ID_{\mathrm{MU}},PK_{\mathrm{MU}}^{(\mathrm{Master})}\rangle, \\ PK_{\mathrm{LTCA}},PK_B^{(\mathrm{Slavery})}\rangle\wedge Decrypt(e''_6)= \\ \langle{}_2R_{\mathrm{LTCA}},PK_{\mathrm{MU}}^{(\mathrm{Master})},{}_2S_1\rangle\wedge Encrypt(e''''')= \\ \langle h_2,PK_{\mathrm{MU}}^{(\mathrm{Master})},S_4\rangle\wedge Encrypt(e'''''')= \\ \langle R_{\mathrm{MU}},PK_{\mathrm{MU}}^{(\mathrm{Master})},S_5\rangle\wedge Encrypt(e''''')= \\ \langle S_5,SK_{\mathrm{LTCA}},S_6\rangle\wedge Send(e''_{10})= \\ \langle Cert_{\mathrm{LTCA}},S_6,S_4,T_2\rangle \end{pmatrix} \tag{7}$$

在式(7)中解密事件 e''_3,e''_4,e''_6 所对应的加密事件已经在对 $Nse\models auth(I_3,2)$得证(即证明 1)中式(1)、式(2)以及式(4)),而加密事件 e''_5 所对应的解密事件同样在对 $Nse\mid\!=auth(I_3,2)$得证(即式(3)、式(4)),此时主体满足:

$$\begin{gathered} {}_2R_{\mathrm{LTCA}}=R_{\mathrm{LTCA}},PK_B^{(\mathrm{Slavery})}=PK_{\mathrm{MU}}^{(\mathrm{Slavery})}, \\ {}_2S_3=S_{3,2}S_2=S_{2,2}S_1=S_{1,2}h_1=h_1. \end{gathered}$$

可以得到:

$$\begin{pmatrix} e''_0<_{\mathrm{loc}}e''_1<_{\mathrm{loc}}e''_2<_{\mathrm{loc}}e''_3<_{\mathrm{loc}}e''_4<_{\mathrm{loc}}e''_5 \\ <_{\mathrm{loc}}e''_6<_{\mathrm{loc}}e'''''<_{\mathrm{loc}}e'''''<_{\mathrm{loc}}e''''<_{\mathrm{loc}}e''_{10}\wedge \\ New(e''_0)=\langle R_{\mathrm{LTCA}}\rangle\wedge Send(e''_1)=\langle L,R_{\mathrm{LTCA}}\rangle\wedge \\ Rcv(e''_2)=\langle M,PK_{\mathrm{MU}}^{(\mathrm{Slavery})},S_3,S_2,S_1\rangle\wedge \\ Decrypt(e''_3)=\langle R_{\mathrm{MU}},SK_{\mathrm{LTCA}},S_3\rangle\wedge \\ Decrypt(e''_4)=\langle h_1,SK_{\mathrm{LTCA}},S_2\rangle\wedge Decrypt(e''_5)= \\ \langle\langle ID_{\mathrm{MU}},PK_{\mathrm{MU}}^{(\mathrm{Master})}\rangle,PK_{\mathrm{LTCA}},PK_B^{(\mathrm{Slavery})}\rangle\wedge \\ Decrypt(e''_6)=\langle{}_2R_{\mathrm{LTCA}},PK_{\mathrm{MU}}^{(\mathrm{Master})},{}_2S_1\rangle\wedge \\ Encrypt(e''''')=\langle h_2,PK_{\mathrm{MU}}^{(\mathrm{Master})},S_4\rangle\wedge \\ Encrypt(e'''''')=\langle R_{\mathrm{MU}},PK_{\mathrm{MU}}^{(\mathrm{Master})},S_5\rangle\wedge \\ Encrypt(e''''')=\langle S_5,SK_{\mathrm{LTCA}},S_6\rangle\wedge \\ Send(e''_{10})=\langle Cert_{\mathrm{LTCA}},S_6,S_4,T_2\rangle \end{pmatrix} \tag{8}$$

线程 thr_2 所涉及式(5)可知,在基本序 R_3 中可以构成完整的$(Send,Rcv)$是事件 e_0,e_5,e_6.根据协议形式化认证规则,结合式(1)(4)(5)(8)可得,$Rcv(e_0)=\langle L,R_{\mathrm{LTCA}}\rangle=Rcv(e''_1)$,$Rcv(e_6)=\langle Cert_{\mathrm{LTCA}},S_6,S_4,T_2\rangle=Send(e''_{10})$,$Send(e_5)=\langle M,PK_{\mathrm{MU}}^{(\mathrm{Slavery})},S_1,S_2,S_3,T_1\rangle=Rcv(e''_2)$,此时就

可以得到一个长度为3的(弱)匹配.

协议满足强匹配会话,则需要证明 $e_1''<e_0, e_5<e_2'', e_{10}''<e_6$,就事件 e_5, e_2'' 进行举例说明.初始阶段规定参与会话双方用户 MU,LTCA 遵守 WMN 认证协议且 MU≠LTCA,根据 $AxiomF$、流关系,事件 e_5, e_2'' 间存在发送事件 s 释放随机数 R_B.如果 $e_5 \leqslant s$,得到排序为 $e_5<e_2''$,再排除 $e_2<_{\text{loc}} j<_{\text{loc}} e_5$.如果 $e_2<_{\text{loc}} j<_{\text{loc}} e_5$,那么 s 必须为 B 的线程成员,由引理1可知线程 thr_2 中事件 e_2 和 e_5 间不存在发送动作,意味着随机数 R_B 在 e_5 前不会释放,这是 $AxiomF$ 证明 $e_5<e_2''$ 所需要的部分[10].

综上所述,根据 $AxiomF$ 和引理1证明 $e_1''<e_0, e_{10}''<e_6$.根据证明过程可知,主体 B 中的线程 thr_2 在协议存在3个完整交互的强匹配.

$$Nse \models auth(R_3, 3). \qquad \text{证毕.}$$

综合证明1)和2)可知,强认证性质 $Nse \models auth(I_3, 2) \wedge Nse \models auth(R_3, 3)$ 得到完整证明,即 WMN 客户端和 LTCA 间双向认证协议同时满足2个强认证性质,此过程不存在消息重放的可能性,协议安全性得证,即 WMN 客户端和 LTCA 间双向认证协议是安全的且攻击者无法通过伪装合法用户进行重放攻击.此时客户端获得 LTCA 的认证证书,可以与其他的客户端用户进行有效通信.

4 形式化方法对比分析

形式化方法用数学模型描述推理基于计算机的系统概念或方法,即规范语言+形式推理.规范语言包括:抽象模型规范法、代数规范法、状态迁移规范法和公理规范法.形式化方法有形式化描述、形式化设计和形式化验证,几十年来国内外专家学者在这3方面研究工作做出巨大贡献.事件逻辑理论是定理证明方法的一种,定理证明优于其他形式化方法的地方在事件逻辑理论中均有体现,但事件逻辑理论相较于其他形式化方法也存在不足.本节综合其他形式化方法对事件逻辑理论进行概述.

4.1 事件逻辑与模态逻辑比较

模态逻辑是目前应用最广泛的形式化方法,包括 BAN 逻辑、类 BAN 逻辑,其中类 BAN 逻辑包含十多种逻辑方法.各类逻辑方法的语法定义各具特色,协议安全属性验证过程采用逻辑推理方式进行证明,这与基于事件逻辑理论的定理证明方法一致.

以 BAN 逻辑为例,相较事件逻辑理论方法,BAN 逻辑要进行大量理想化处理,这其中包含协议前提、协议本身、协议目标等,这些动作通过形式化描述实现.

1) BAN 逻辑形式化描述协议初始化假设、安全协议预期目标,这些步骤与事件逻辑理论方法类似,但 BAN 逻辑对所处执行环境、参与协议执行主体和协议使用密钥等作出的初始假设过分依赖.事件逻辑理论对协议初始化方面没有过分处理,但对协议客观环境进行假设要求,在协议交互过程进行合理抽象化处理.

2) BAN 逻辑对协议理想化处理过分依赖分析者直觉,理想化过程会产生问题,使得理想化后协议与原来协议有一定差距.例如忽略或增加协议前提或内容,对协议目标描述不够准确,易造成分析结果与原协议设计有一定出入[20].事件逻辑理论定义严谨的数学规则,规范一系列公理推论规则约束,从而保证强认证性质证明过程的严谨性.

3) BAN 逻辑利用规则进行按步推理,运用严谨推理逻辑证明协议.但是 BAN 逻辑语义刻画不够清晰,证明协议安全性并不能让人很好信服,同样 BAN 逻辑不能很好保证协议证明的实用性.事件逻辑理论定义的定理规则,结构清晰、无歧义,保证协议在证明过程中的真实可靠性,使得协议安全属性更具有可信性.

4) BAN 逻辑规则定义相较事件逻辑理论更加成熟,研究方面更广泛.在定义规则上,BAN 逻辑规则可读性更强,使用者不需要过高的数学基础以及逻辑推理能力.在使用 BAN 逻辑证明过程中,使用者可以通过调用定义规则对目标进行证明.

5) 模态逻辑与事件逻辑理论一样,需要大量的手工作业,在形式化自动化验证方面有待提高.

4.2 事件逻辑理论与 PCL 比较

协议组合逻辑(protocol composition logic, PCL)是一种证明加密协议安全系统的逻辑方法.PCL 与事件逻辑理论方法一样,协议形式化建模生成随机数、接收/发送消息、对消息执行加密操作以及签名验证.

1) 协议安全属性验证中,PCL 方法只能对部分协议性质进行刻画,对数据签名协议的认证性质不能进行刻画,而基于事件逻辑理论的定理证明方法可以对其他安全属性进行刻画认证.

2) PCL 方法在协议交互动作建模中不够严谨,对描述线程的前序动作序列机制缺乏定义.事件逻辑理论方法对协议形式化建模线程机制进行明确定义,通过原子独立性规范事件发生先后线程状态.

3) PCL 方法对协议信息数据类型缺乏必要约束、限制,事件逻辑理论定义特有 *Atom* 类型对没有结构且不能被生成的基本元素进行规范化定义,保证协议信息数据类型的规范充分.

4) PCL 方法描述 Diffie-Hellman 代数行为及散列函数的能力不足,以散列函数加密为例,事件逻辑理论在处理加密动作时选择对其进行刻画.以散列函数加密为例,在消息解密验证过程中,事件逻辑理论方法通过有效信息生成新的散列加密,然后与刻画的加密动作进行比较证明.

5) PCL 在应用中经历时间长久,特别是在协议动作分布自动化方面做了大量工作,此方面是事件逻辑理论方法亟待改进的.

4.3 事件逻辑通用性

事件逻辑理论对分布式系统协议和算法进行描述,通过捕捉分布式系统模型实现.在事件逻辑理论基本定义中,对原子类型、独立性、事件结构进行初步建模,保证加密系统建模过程中随机数、事件类建模的完备性.事件逻辑理论通用性具体表现在 3 个方面:

1) 复杂协议对象简化处理

协议对象的简化处理要求对协议抽象过程、协议通用属性进行标识,保证协议对象的平凡性;对特有属性进行特别标识,保证协议对象独特性.例如,在 WMN 客户端与 LTCA 交互认证协议中,客户端与 LTCA 具有共有属性公/私钥(PK,SK),但 2 个主体有各自独特性,客户端拥有特有辅公钥 $PK_{\mathrm{MU}}^{(\mathrm{Slavery})}$,LTCA 通过合法程序生成客户端用户辅公钥的权限,在认证过程中生成唯一的证书 $Cert_L$,需要在协议形式化描述中对这些性质进行特别刻画.

2) 协议交互环境合理化假设

有线网络与无线网络环境对于协议应用要求有很大不同,相较而言,无线网络环境下安全协议设计有挑战性.安全协议不仅要保证用户能够承受有线网络信息传递过程遇到的风险,还要面临无线网络环境独有的危险,对协议安全性认证也是挑战.

① 不同协议适用条件有所不同,协议形式化描述前要对协议执行条件进行了解,对协议交互条件进行合理化分析.

例如 WMN 客户端协议对比有线网络安全协议没有可靠的认证中心,对 LTCA/LCA 形式化描述要求具体化,轻量级 CA 与客户端认证中加入辅公钥以及证书机制,加入门限机制容侵的轻量级 CA 则在 LTCA 基础上加入时间戳 T 保证 LTCA 在加密解密信息迭代的时间有效性.

② 对不同协议提出合理化假设条件,在合理假设条件中要求研究者了解协议运行的基本环境和独特环境.在提出假设的过程中,假设条件不宜过多,反之体现协议运行条件的苛刻,对协议独特的运行环境进行合理化提取.

例如 WMN 客户端间认证协议要求移动主体 A 与移动主体 B 在有安全保障条件下进行通信,在 A,B 节点间进行双向认证并产生安全的会话密钥.安全保障不是说协议交互环境绝对安全,协议交互过程有受到攻击者攻击的可能,安全保障是指在协议交互过程中不会受到恶劣自然环境或大型设备故障等非人为蓄意情况的影响.客户端间交互时,双方都收到认证中心的认证,拥有认证中心颁发的证书.

3) 协议交互过程删繁就简

协议信息在无线网络通道传递的过程中包含各种基本信息,例如数据在物理层、网络层等的传输交互,这些信息在事件逻辑理论形式化描述过程中均不做具象处理,以信息 Msg 来定义.但是协议动作对消息处理要进行明确标注,保证事件逻辑理论方法在协议建模时信息传递过程的有效性.

例如 Needham-Schroeder 协议交互双方在进行信息传递过程中,主体 A 通过主体 B 的公钥 PK_B 对自己产生的随机数 Na 以及时间戳 T_1 进行加密,将消息传递给用户主体 B.在形式化描述过程中,对信息的各层封装交互并不做研究描述,直接总括这一过程,即 $A \rightarrow B: Sign_{PK_B}\{A, Na, T_1\}$.

5 结 论

5.1 总 结

信息安全涉及信息资源交互中的机密性、完整性以及可用性,安全协议形式化方法在长期积累中形成比较完善的理论体系以及模型,定理证明是形式化方法的一种,基于严格数学理论知识和逻辑推导,从而确定协议在合理假设条件下是否满足要验证的安全属性[21].本文基于事件逻辑理论,使用定理证明的方法对 WMN 客户端协议认证性进行分析,所取得的成果具体有 4 个方面:

1) 对事件逻辑理论进行扩展,提出置换规则保证协议交互用户在置换中性质的等价转换.将基于事件逻辑理论证明协议安全性的过程通过流程图表述,详细介绍事件逻辑理论证明过程.

2) 通过事件逻辑描述客户端与 LTCA 间交互

协议的基本序列,对协议交互动作形式化描述.证明协议强认证性质,得出 WMN 客户端与 LTCA 间认证协议在合理假设下是安全的.表明事件逻辑理论可以对安全协议不同身份主体间的认证性进行证明.

3) 结合文献[10,19]可知,事件逻辑不仅可以对有线网络安全协议的安全属性进行证明,对无线网络安全协议的安全属性也可以给予合理论证,进而保证安全协议在用户交互过程中的可靠性.

4) 分析事件逻辑理论方法与其他逻辑证明方法的优缺点,说明事件逻辑理论具有通用性.

5.2 展望

国内外专家学者对无线网络协议研究领域做了大量的研究,取得了丰硕成果.绝对安全协议是不存在的,不论多么完美的协议总会存在未被发现的漏洞,还需要进一步的研究证明甚至是改进论证.本文运用的事件逻辑理论方法也存在不足和需要进一步改善的方面,根据目前研究成果,今后进一步研究方向可以从 4 点进行考虑:

1) 事件逻辑理论证明方法更多依靠手工证明,此次对该方法各个步骤进行定义,下一步研究可以着手实现分布自动化.将分步实施的概念加入正在构建的安全协议验证系统中,从而实现网络安全协议认证性的自动化证明.

2) 在对协议安全属性进行分析研究时,目前更多的是通过事件逻辑理论方法对安全协议认证性证明,下一步可以考虑通过基于事件逻辑理论的定理证明方法对安全协议保密性、完整性等安全属性进行形式化证明.

3) 事件逻辑理论在安全协议证明过程中可能存在不足,可以对事件逻辑理论相关推论性质进行分析扩展及证明.

4) 单一的形式化方法证明存在缺陷,可以考虑在安全协议的研究证明中结合模型检测或模态逻辑等方法对协议安全属性进行分析,从而保障安全协议强认证性质刻画的完备性.

参 考 文 献

[1] Lu Laifeng. Study on theory and applications of security protocols formal analysis [D]. Xian: Xidian University, 2012 (in Chinese)
(鲁来凤. 安全协议形式化分析理论与应用研究[D]. 西安:西安电子科技大学, 2012)

[2] Xu Yiyun. The analysis and detection attack and tactics [J]. Network Security Technology & Application, 2011(2): 14-15 (in Chinese)
(许奕芸. 黑客入侵技术的分析和检测[J]. 网络安全技术与应用, 2011(2): 14-15)

[3] Liu Zhongqiang, Yu Chengli. Starting from Wannacry blackmail virus, this paper explores the security defense strategy of computer viruses in LAN [J]. Secret Science and Technology, 2017(6): 18-21 (in Chinese)
(刘中强, 于成丽. 从 Wannacry 勒索病毒着手探究局域网内计算机病毒的安全防御策略[J]. 保密科学技术, 2017(6): 18-21)

[4] Chen Xingyue. Cybersecurity capability construction: Coordination of consciousness, management and technology - thoughts triggered by the "eternal blue" event [J]. Journal of Information Security Reserach, 2017, 3(8): 765-768 (in Chinese)
(陈兴跃. 网络安全能力建设:意识、管理和技术的协同——"永恒之蓝"勒索蠕虫爆发事件引发的思考[J]. 信息安全研究, 2017, 3(8): 765-768)

[5] Zhang Weipeng. Research on attack techniques in wireless mesh network [D]. Xian: Xidian University, 2014 (in Chinese)
(张威鹏. 无线 Mesh 网络攻击技术研究[D].西安:西安电子科技大学, 2014)

[6] Sharma P K, Mahajan R, Surender. A security architecture for attacks detection and authentication in wireless mesh networks [J]. Cluster Computing, 2017, 20(3): 2323-2332

[7] Ji Qingguang, Feng Dengguo. Towards analyzing some kinds of critically formal models for network security protocols [J]. Chinese Journal of Computers, 2005, 28(7): 1071-1083 (in Chinese)
(季庆光, 冯登国. 对几类重要网络安全协议形式模型的分析[J]. 计算机学报, 2005, 28(7): 1071-1083)

[8] Guo Ping. Research on authentication architecture and related technologies of wireless networks [D]. Nanjing: Nanjing University of Science and Technology, 2012 (in Chinese)
(郭萍. 无线网络认证体系结构及相关技术研究[D].南京:南京理工大学, 2012)

[9] Liu Xinqian. Formal analysis of provable network security protocol based on logic of events [D]. Nanchang: East China Jiaotong University, 2016 (in Chinese)
(刘欣倩. 基于事件逻辑的可证明网络安全协议形式化分析[D].南昌:华东交通大学, 2016)

[10] Xiao Meihua, Liu Xinqian, Li Yanan, et al. Security certification of three-party network protocols based on strong authentication theory [J]. Journal of Frontiers of Computer Science and Technology, 2016, 10(12): 1701-1710 (in Chinese)
(肖美华, 刘欣倩, 李娅楠, 等. 基于强认证理论的三方网络协议安全性证明[J]. 计算机科学与探索, 2016, 10(12): 1701-1710)

[11] Bickford M. Unguessable Atoms: A logical foundation for security [C] //Proc of the 2nd Int Conf on Verified Software: Theories, Tools, Experiments. Berlin: Springer, 2008: 30-53

[12] Xiao Meihua. Proving authentication property of modified needham-schroeder protocol with logic of events [C] //Proc of Int Conf on Computer Information Systems and Industrial Applications. Paris: Atlantis, 2015: 379-384

[13] Xiao Meihua, Ma Chenglin, Deng Chunyan. A novel approach to automatic security protocol analysis based on authentication event logic [J]. Chinese Journal of Electronics, 2015, 24(1): 187-192

[14] Bickford M. Automated proof of authentication protocols in a logic of events [C] //Proc of Int Joint Conf on All Aspects of Automated Reasoning. Berlin: Springer, 2010: 13-30

[15] Constable R, Bickford M. Intuitionistic completeness of first-order logic [J]. Annals of Pure & Applied Logic, 2011, 165(1): 164-198

[16] Tan Yongzhou. Research on authentication protocol of wireless mesh networks based on lightweight CA [D]. Xiangtan, Hunan: Hunan University of Science and Technology, 2015 (in Chinese)
(谭永洲. 基于轻量级 CA 的无线 Mesh 网络认证研究[D]. 湖南湘潭: 湖南科技大学, 2015)

[17] Guo Ping, Fu Desheng, Zhu Jiezhong, et al. Scheme of lite and tolerant certification authority for wireless mesh network [J]. Computer Science, 2013, 40(12): 200-204 (in Chinese)
(郭萍, 傅德胜, 朱节中, 等. 无线 Mesh 网络轻量级容侵 CA 方案[J]. 计算机科学, 2013, 40(12): 200-204)

[18] Liu Xinqian, Xiao Meihua, Cheng Daolei, et al. Proving security properties of modified needham-schroeder protocol based on logic of event [J]. Computer Engineering and Science, 2015, 37(10): 1850-1855 (in Chinese)
(刘欣倩, 肖美华, 程道雷, 等. 基于事件逻辑理论的改进 Needham-Schroeder 协议安全性证明[J]. 计算机工程与科学, 2015, 37(10): 1850-1855)

[19] Li Yanan, Xiao Meihua. Li Wei, et al. Security certification of the authentication protocol of wireless mesh networks based on logic of events [J]. Computer Engineering and Science, 2017, 39(12): 2236-2244 (in Chinese)
(李娅楠, 肖美华, 李伟, 等. 基于事件逻辑的无线 Mesh 网络认证协议安全性证明[J]. 计算机工程与科学, 2017, 39(12): 2236-2244)

[20] Mei Chong. Research on security protocol analysis and checking based on Petri nets [D]. Guiyang: Guizhou University, 2008 (in Chinese)
(梅翀. 基于 Petri 网的安全协议分析与检测方法的研究[D]. 贵阳: 贵州大学, 2008)

[21] Zhang Xiaohong. The research of automatic validation algorithm for security protocols based on formal methods [D]. Changsha: Hunan University, 2010 (in Chinese)
(张孝红. 基于形式化方法的安全协议自动化验证算法的研究[D]. 长沙: 湖南大学, 2010)

Xiao Meihua, born in 1967. Professor and PhD supervisor. Senior member of CCF. His main research interests include information security and software formal method.

Li Yanan, born in 1992. Master. Her main research interests include information security and software formal method.

Song Jiawen, born in 1995. Master candidate. Her main research interests include information security and software formal method.

Wang Xizhong, born in 1992. Master candidate. His main research interests include information security and software formal method.

Li Wei, born in 1992. Master. His main research interests include information security and software formal method.

Zhong Xiaomei, born in 1980. PhD candidate. Her main research interests include information security and software formal method.