

文章编号:1001-9081(2007)09-2189-05

基于一阶逻辑的非否认协议模型

范钰丹, 韩继红, 王亚弟, 赵宇, 朱玉娜
(信息工程大学 电子技术学院, 郑州 450004)
(vanedd1982@yahoo.com.cn)

摘要:为了将密码协议的非否认性和公平性统一在一个框架之下更好地进行分析,提出了一套适用于分析非否认性和公平性的一阶逻辑语法和语义。在此基础上建立了一个用于分析非否认性和公平性的一阶逻辑模型,并以 Fair ZG 非否认协议为例进行了分析,发现了该协议的一个已知攻击,证明了模型的有效性和正确性。

关键词:非否认性;公平性;一阶逻辑;形式化分析

中图分类号:TP309 **文献标志码:**A

Non-repudiation protocols model based on first-order logic

FAN Yu-dan, HAN Ji-hong, WANG Ya-di, ZHAO Yu, ZHU Yu-na
(Institute of Electronic Technology, the PLA Information Engineering University, Zhengzhou Henan 450004, China)

Abstract: In order to analyze the non-repudiation and fairness properties under one frame, a set of first-order logic syntax and semantics for non-repudiation protocols was proposed, and a model was built. The Zhou-Gollmann fair non-repudiation protocol was analyzed with the model and a known attack to the protocol was found. The results show that our model is valid and correct.

Key words: non-repudiation; fairness; first-order logic; formal analysis

0 引言

多年来,密码协议的形式化研究主要集中在认证协议和密钥建立协议上,其研究方法不能直接用于分析非否认协议。非否认协议与认证协议有着根本的不同。认证协议的主要问题是解决攻击者的出现,而非否认协议主要考虑的是参与实体间可能存在的欺骗行为。这就决定了非否认协议的建模不同于认证协议。

近年来,出现了一些分析非否认协议的例子,但基本上都是采用已有的密码协议形式化分析方法进行的^[6~16]。文献[16]用基于 Prolog 规则的协议验证器对一个认证邮件协议进行了分析,这是唯一一个用一阶逻辑分析非否认协议的例子,但它是在现有的验证器上进行的,不是专门为分析非否认性而设计的,而且还需要一定的手工辅助。目前为止,还没有发现专门使用一阶逻辑对非否认协议进行建模和分析的公开例子。本文针对非否认协议的特点,提出了非否认协议的一阶逻辑模型,专门用来分析非否认性和公平性,并以文献[1]提出的一个 Fair ZG 非否认协议为例进行了分析,发现了该协议的一个已知攻击,证明了模型的有效性和正确性。

1 ZG 非否认协议

由于经典的带可信第三方的公平非否认协议(简称 ZG 协议)^[1]简单、高效,得到了广泛的关注和讨论。对非否认性的研究大部分都是以该协议为例的。ZG 协议的具体描述如下:

1) $A \rightarrow B: f_{EOO}, B, L, C, EOO$

2) $B \rightarrow A: f_{EOR}, A, L, EOR$
3) $A \rightarrow TTP: f_{SUB}, B, L, K, sub_K$
4) $A \leftrightarrow TTP: f_{CON}, A, C, L, K, con_K$
5) $B \leftrightarrow TTP: f_{CON}, A, C, L, K, con_K$

其中:

$sS_A(X)$: 主体 A 用私钥对 X 的数字签名;
 $C = \{M\}_K$: 用 K 对 M 加密所得的密文;
K: 用来对 M 加密的对称密钥,由 A 给出;
L: 连接一次协议运行所有消息的唯一标识符;
 $EOO = sS_A(f_{EOO}, B, L, C)$: 消息 M 的非否认发起证据;
 $EOR = sS_B(f_{EOR}, A, L, C)$: 消息 M 的非否认接收证据;
 $sub_K = sS_A(f_{SUB}, B, L, K)$: 密钥 K 的提交证据;
 $con_K = sS_{TTP}(f_{CON}, A, B, L, K)$: TTP 发出的密钥 K 的确认证据;
 $f_{EOO} f_{EOR} f_{SUB} f_{CON}$: 各个证据的标记;
 \leftrightarrow : ftp 操作。

2 一阶逻辑方法

Bruno 提出了一套分析密码协议秘密性和认证性的一阶逻辑语法和语义^[5],我们在其基础上,针对非否认协议的特点,定义了一套自己的语法和语义。其中,语法结构主要由项、事实和规则组成(如表 1 所示)。

表中:

变量可以描述任意项。

名用来描述需要区分的原子值,如密钥和新鲜数等,通过名中参数的不同来区分不同的项。

收稿日期:2007-03-06;修回日期:2007-06-08。

作者简介:范钰丹(1982-),女,河南南阳人,硕士研究生,主要研究方向:信息安全、密码协议自动化验证; 韩继红(1966-),女,山西定襄人,副教授,博士,主要研究方向:计算机网络安全、信息系统安全; 王亚弟(1953-),男,甘肃兰州人,教授,博士生导师,博士,主要研究方向:计算机网络安全、信息系统安全; 赵宇(1983-),男,山东成武人,硕士研究生,主要研究方向:信息安全、密码协议自动化验证; 朱玉娜(1985-),女,山东菏泽人,硕士研究生,主要研究方向:信息安全、密码协议自动化验证。

函数用来描述协议中出现的密码学原语,函数作用在项上,产生另一个新的项。

事实 $agent(m, p(X, flag))$ 表示参与主体发送或接收消息 m 。为了区分协议主体是诚实的还是恶意的,我们定义 $agent$ 谓词的第二个参数 $p(X, flag)$ 用来表示该消息的拥有者,它包含两个项: X 标识消息的发送者或接收者, $flag$ 标记该主体的性质($true$ 表示诚实主体, $false$ 表示恶意主体)。 $agent$ 谓词的引入将消息和消息来源绑定在一起,并可以区分是诚实主体还是恶意主体的行为,解决了非否认性的“责任”问题。

$mess(c, m)$ 表示消息 m 可以出现在信道 c 上。

$askJSays(p, \rho)$ 表示某事件执行结束的同时,触发请求 J 的仲裁,其中 p 为与该事件相关的参数, ρ 为变量和名到项的映射; $answerJSays(p, s)$ 表示 J 已经对某事件进行了仲裁, p 表示与该事件相关的参数, s 表示该事件执行的会话的标记符。

$match(a, b)$ 谓词,用来解决消息的检验问题。在主体收到了某条消息之后,或者仲裁对提供来的证据进行检验时,都要对消息进行检验,如果检验成功,才继续执行协议或得出相应结论。因此,引入了 $match(a, b)$ 谓词来解决这一问题。它表示当 $a = b$ 时,规则的前提为真,则结论为真;否则,前提为假,则结论为假。其中, a, b 均为项。

规则都是 $F_1 \wedge \dots \wedge F_n \rightarrow F$ 形式的 Horn 子句, F_1, \dots, F_n 为事实,当其都为真时, F 为真。

表1 非否认协议的一阶逻辑语法结构

符号	含义
$M, N ::=$	项
x, y, z, i, j	变量
$a[M_1, \dots, M_n], i_0, j_0$	名
$f(M_1, \dots, M_n)$	函数应用
$F ::=$	事实
$agent(m, p(X, flag))$	$agent$ 谓词
$mess(c, m)$	信道上的消息
$askJSays(p, \rho)$	$askJSays$ 事件
$answerJSays(p, s)$	$answerJSays$ 事件
$match(a, b)$	$match$ 谓词
$R ::=$	规则
$F_1 \wedge \dots \wedge F_n \rightarrow F$	逻辑规则

通过逻辑规则之间的合一化消解实现密码协议安全特性的验证。

3 非否认协议的建模方法

由于非否认协议受到的主要威胁来自于互不信任的通信双方,因此,在我们的一阶逻辑非否认协议模型中,没有外来的攻击者,模型中的要素包括机密可靠的通信信道,协议主体,仲裁和安全性目标。假设一次协议会话中最多只能有一个主体是不诚实的,协议主体不进行合谋欺骗,也不进行不利于自己的欺骗。

模型分以下三种情况考虑:

- 1) A, B 均为诚实实体;
- 2) A 是恶意实体, B 是诚实实体;
- 3) B 是恶意实体, A 是诚实实体。

3.1 非否认协议的目标

3.1.1 非否认性

非否认性,是指协议结束后,参与双方能够提供有效的证据用以证明主体声称的某一行为的出现或不出现是一个不可

反驳的事实,即有的文献中所说的“证据的有效性”^[6,10]。它分为两个方面:

发方非否认性,需要有发方非否认证据(EOO),用于证明发送方确实发送过某个消息。

收方非否认性,需要有收方非否认证据(EOR),用于证明接收方确实接收了某个消息。

如果协议运行结束后,某个主体否认其协议行为并引发纠纷,协议参与双方可以向仲裁提供证据,仲裁通过对证据的检验来判定主体的否认行为是否成立。于是,我们这样处理非否认性问题:

对于发方非否认性,在发方 A 完成了协议之后,触发一个 $askJSaysAsend$ 事件,表明他已经完成了该次协议会话。一旦 A 否认发送过消息, B 就将证据 EOO 提交给仲裁,如果仲裁检验成功,就触发 $answerJSaysAsend$ 事件。如果在 $askJSaysAsend$ 事件发生的情况下, $answerJSaysAsend$ 事件也发生了,则称协议满足发方非否认性;如果 $answerJSaysAsend$ 事件没有发生,即仲裁检验没有通过,仲裁认为 A 没有发送该消息,于是 A 否认成功,则称协议不满足发方非否认性。

收方非否认性与发方非否认性类似。

在我们的模型中为每一次协议会话都加入一个仲裁过程,假设每运行完一次协议会话,参与双方都将证据提交给仲裁,仲裁进行判断。那么当无穷多个协议会话并行运行时,就会出现无穷多次仲裁判断。为了将仲裁的某次判断与某次会话的某个主体的行为对应起来,就需要采用 Woo 和 Lam 的对应性断言来解决非否认性问题,即当某个主体完成了一次协议会话时,仲裁就认为该主体确实参与了这次会话运行。也就是说,如果能够证明“在 $askJSays$ 事件发生的前提下,对应的 $answerJSays$ 事件也发生了”,则称协议满足相应的非否认性。

令 $B_b = \{askJSays(p_1, \rho_1), \dots, askJSays(p_n, \rho_n)\}, askJSays(p_i, \rho_i)$ 是以 $askJSays$ 作为谓词符号的原子公式,于是非否认性的一阶逻辑定义为:

定义1 非否认性。 B_0 是基于一阶逻辑的非否认协议模型中的逻辑规则构成的集合,设 P_0 为待验证协议,对于任意的 $askJSays(p, \rho) \in B_b$,都有 $answerJSays(p, s)$ 可以从 $B_0 \cup B_b$ 中导出且 $getsession(\rho) = s$ 成立,则称 $P_0 sat NRO$ (或 $P_0 sat NRR$),即协议 P_0 满足发方(或收方)非否认性。

3.1.2 公平性

公平性必须保证在至少一方诚实的前提下,协议执行到任何一个状态中止,双方都不占优势,即要么双方都得到期望的非否认证据,要么都得不到。

在这个定义中,“优势”和“任何一个状态”是最难形式化的,我们采用一种新方法来解决公平性问题。

首先,从非否认协议的特点出发。非否认协议的设计思想是将发起方要发送的消息 m 分成若干子消息 m_1, \dots, m_n 分别发送,每发送一条子消息 m_i ,都将该子消息的发方非否认证据 EOO_i 一同发给对方,同时在收到对方发来的收方非否认证据 EOR_i 之后,才给对方继续发送其他的子消息,只有拥有了所有的子消息,才能获得消息 m 。因此,对方为了获得消息 m 和所有证据,就不会随意中止协议,而发方为了得到所有的接收证据,也不会轻易中止协议。但是,如果协议设计中存在漏洞,就会使某个恶意主体通过重放、伪造信息等手段提前获得了所有的证据,从而单方中止协议运行,导致没有完全获得证据的一方处于劣势,这就是协议的不公平现象。

针对这个特点,可以根据发送每条子消息的时机,将协议分为若干个子阶段,将每个子阶段都视为一个完整的子协议

运行,通过验证协议在运行到某个状态 k 之前的所有子阶段的公平性,来验证协议在此状态中止时的公平性。当状态 k 之前的所有子阶段的公平性都满足时,协议才满足公平性;若有一个子阶段的公平性不满足,则协议不满足公平性。

于是公平性的一阶逻辑定义为:

定义2 公平性。令协议要发送的消息 m 由若干子消息 m_1, \dots, m_n 组成,假设根据发送每条子消息 m_i 的时机将协议分成 n 个子阶段。假设协议运行到某个状态 k 结束,则协议的公平性定义为:

$$\text{Fairness} = (NRO_1 \odot NRR_1) \wedge (NRO_2 \odot NRR_2) \wedge \dots \wedge (NRO_k \odot NRR_k)$$

其中: NRO_i 是第 i 个子阶段的发方非否认性; NRR_i 是第 i 个子阶段的收方非否认性 ($0 \leq i \leq k \leq n$)。

3.2 对信道的建模

文献[14]将协议中不诚实主体的恶意行为分担了一部分给信道,如:将主体不发消息或滥发消息的恶意行为归结为交易方之间的不可靠信道。而在我们的模型中,为了充分考虑协议主体之间的恶意行为,将这部分责任还给恶意主体,假设所有信道都是可靠的,即在协议运行中不会由于信道的原因而丢失消息。

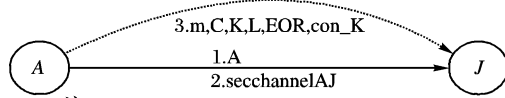
根据非否认协议的特点,我们将信道分为两种:通用信道和专用信道。

通用信道:协议中的合法主体都可以使用的一种信道。

专用信道:只供通信双方使用的专有信道,其他任何实体都不能在上面进行监听、截获、收发消息等,一般在通信方与仲裁之间使用专用信道。

对于通用信道,只需要按正常的信道建模,用自由名(如 $c[]$)来表示。如主体在通用信道 c 上发送消息 m ,则表示为 $\text{mess}(c[], m)$ 。由于通用信道 c 可以被任何合法主体使用,故此时的 $\text{mess}(c[], m)$ 就可以直接用 $\text{agent}(m, p(X, \text{flag}))$ 来表示。

对于专用信道,由于消息的收发双方是固定的,并非所有的主体都可以使用,因此我们采用一种特殊的方法进行建模,步骤如图1所示(以ZG协议中A和仲裁J之间的信道为例)。



注: \longrightarrow 一个非对称的通用信道ChannelToJ

$\cdots \longrightarrow$ 专用信道secchannelAJ

图1 ZG协议中的专用信道模型

1) 当A在接收到消息 M_1, \dots, M_n 之后,如果要和仲裁J通信,则首先在信道 ChannelToJ 上向仲裁J发送自己的身份标识 $\text{host}(sk_A[])$ 。其中 ChannelToJ 是一个非对称信道,任何合法主体在此信道上都可以写,但只有仲裁J可以读,故这个信道实际上提供了对仲裁J的认证。

$$\text{agent}(M_1, p(A, \text{true})) \& \dots \& \text{agent}(M_n, p(A, \text{true})) \rightarrow \text{mess}(\text{ChannelToJ}[], A) \quad \text{R1}$$

2) A创建一个和仲裁之间的专用信道 secchannelAJ ,并在非对称信道 ChannelToJ 上将该专用信道名发送给仲裁J,告诉仲裁他们将会在这个专用信道上通信。

$$\text{agent}(M_1, p(A, \text{true})) \& \dots \& \text{agent}(M_n, p(A, \text{true})) \rightarrow \text{mess}(\text{ChannelToJ}[], \text{secchannelAJ}[M_1, \dots, M_n]) \quad \text{R2}$$

3) 在专用信道上定义从A到J的读信道($J, \text{secchannelAJ}$)和从J到A的写信道($A, \text{secchannelAJ}$),在读、写信道上开始A与J之间的消息传递(在ZG协议中由于仲裁只

需读取证据,所以不需要定义写信道)。A在读信道上向仲裁发送证据($m[], C, k[x], L[], \text{EOR}, \text{con}_K$)。

$$\text{agent}(M_1, p(A, \text{true})) \& \dots \& \text{agent}(M_n, p(A, \text{true})) \rightarrow \text{mess}(J, \text{secchannelAJ}[M_1, \dots, M_n], (m[], C, k[x], L[], \text{EOR}, \text{con}_K)) \quad \text{R3}$$

至此,建立了A与J之间的专用信道,其他任何主体都不能监听该信道,也不能截取上面的消息,或者在上面收发消息等。这就保证了只要J在该信道上收到了某条消息,那么此条消息一定是来自主体A的。

3.3 对协议主体的建模

非否认协议中除TTP外所有的实体都有可能是不诚实的。因此,在对非否认协议的实体A和B进行建模时,需要分两种情况:诚实主体和恶意主体。

3.3.1 诚实主体

诚实主体严格按照协议规定的步骤执行。我们为协议中的每个主体分别建模,根据协议的描述为每个主体对应一组形式如下的逻辑规则:

$$\text{agent}(M_1, p(X, \text{true})) \wedge \dots \wedge \text{agent}(M_n, p(X, \text{true})) \rightarrow \text{agent}(M, p(X, \text{true})) \quad (1)$$

或

$$\text{askJSays}(M_1, M_1') \wedge \dots \wedge \text{agent}(M_1, p(X, \text{true})) \wedge \dots \wedge \text{match}(M, M') \rightarrow \text{answerJSays}(M, M') \quad (2)$$

规则(1)表示诚实主体X必须在先接收到消息 M_1, \dots, M_n 的前提下才能发送消息M。规则(2)是在解决非否认性问题时用到的规则,表示当前件中的条件都为真时,才触发 $\text{answerJSays}(M, M')$ 事件。

3.3.2 恶意主体

在非否认协议中,除TTP外,所有的主体都有可能是不诚实的。在协议运行过程中,恶意主体扮演着双重角色,即诚实角色和恶意角色:

1) 诚实角色是指该主体有可能按照协议规定步骤执行的行为,对于这部分行为的描述,将按照前面介绍的诚实主体的建模方法进行。

2) 恶意主体最主要的是他所扮演的恶意角色,即不按照协议要求执行协议,而是在执行协议的同时,根据自己所拥有的初始知识和在执行协议过程中获得的知识不停地计算,实施恶意行为。由于这部分行为是不确定的,因此,采用对恶意主体的初始知识和计算能力建模的方式进行。

恶意主体的初始知识包括:(1)协议的所有参与主体的名称、信道名以及它们的公开密钥;(2)恶意主体自己的私钥以及与其他主体的共享密钥;(3)恶意主体自己产生的随机数、会话标识等。可表示成如下形式:

$$\forall a \in S, \text{agent}(a[], p(X, \text{false})) \quad \text{Init}$$

其中,S表示恶意主体拥有的所有名的集合。

在协议运行中,恶意主体可以按照要求执行协议,但更重要的是不按照协议要求,而是根据自己拥有的情况不断地计算,期望占据某种优势。通过总结恶意主体的各种攻击行为,将恶意主体的能力描述为规则 $\text{Rn}', \text{Rh}, \dots, \text{Rd}$ 。

1) 组合消息的能力:

$$(1) \text{产生新鲜数的能力:} \quad \text{agent}(b, p(X, \text{false})) \quad \text{Rn}'$$

(2) 进行哈希操作的能力:

$$\text{agent}(x, p(X, \text{false})) \rightarrow \text{agent}(H(x), p(X, \text{false})) \quad \text{Rh}$$

(3) 加密消息的能力:

$$\text{agent}(x_1, p(X, \text{false})) \wedge \text{agent}(x_2, p(X, \text{false})) \rightarrow$$

$agent(\{x_1\}_{x_2}, p(X, false))$
 2) 分解消息的能力:
 $agent(\{[m]\}_{M^+}, p(X, false)) \wedge$
 $agent(M^-, p(X, false)) \rightarrow agent(m, p(X, false))$ Rad
 $agent(\{[m]\}_{M^-}, p(X, false)) \wedge$
 $agent(M^+, p(X, false)) \rightarrow agent(m, p(X, false))$ Rcs
 $agent(\{m\}_k, p(X, false)) \wedge agent(k, p(X, false)) \rightarrow$
 $agent(m, p(X, false))$ Rsd
 3) 对信道的操作能力:
 监听能力:
 $mess(x, y) \wedge agent(x, p(X, false)) \rightarrow$
 $agent(y, p(X, false))$ RI
 发送能力:
 $agent(x, p(X, false)) \wedge agent(y, p(X, false)) \rightarrow$
 $mess(x, y)$ Rs
 4) 随意中止协议的能力:
 $agent(x, p(X, false)) \rightarrow$ Rd

3.4 对仲裁的建模

仲裁是为解决非否认性问题专门加入的一个诚实主体,在我们的模型,它是由一个区别于 TTP 的独立实体担任。通常仲裁这个角色不参与协议的运行。对仲裁的建模步骤如下:

- 1) 建立一个和参与方之间的专用信道,在专用信道上接收消息;
- 2) 对接收到的消息进行验证;
- 3) 如果验证成功,则触发 $answerJSays$ 事件。

我们为每一次协议会话都加入一个仲裁过程。如果协议运行结束后出现纠纷,就可以通过提出否认的主体的 $askJSays$ 事件和仲裁的 $answerJSays$ 事件的对应关系判断协议是否满足非否认性。

以 ZG 协议为例,协议运行完成后,可能产生的一种争端是:A 发送了 m 给 B ,但 B 否认接收过 m 。针对这种情况,仲裁 J 的具体描述规则如下:

$mess(ChannelToJ[], x) \wedge$
 $mess(ChannelToJ[], secchannelAJ) \wedge$
 $mess((hostJ[], secchannelAJ), (m, C, K, L, EOO, con_K)) \wedge$
 $macth((f_{CON}, A, B, L, K), checksign(con_K, pk(skTTP))) \wedge$
 $macth((f_{EOR}, A, L, C), checksign(EOO, pk(skB))) \wedge$
 $macth(m, sdecrypt(C, K))$
 $\rightarrow answerJSaysBreceived((host(sk_B[]), m), sid21)$

表示仲裁 J 在建立了与 A 之间的专用信道 $secchannelAJ$,且在 $inchannelAJ = (hostJ, secchannelAJ)$ 上收到了 A 传来的消息 (m, C, K, L, EOR, con_K) 后,进行检验:

- 1) 检验 con_K 是 TTP 对 f_{CON}, A, B, L, K 的签名;
- 2) 检验 EOR 是 B 对 f_{EOR}, A, L, C 的签名;
- 3) 检验 $C = \{m\}_K$ 。

如果以上三次检验都正确,仲裁就相信 B 接收了消息 m ,触发 $answerJSays$ 事件,其中 $(host(sk_B[]), m)$ 是仲裁判断的结果主体 B 和消息 m , $sid21$ 是会话标识符。

4 ZG 协议的一阶逻辑模型及分析

本节将根据上节提出的建模方法对 ZG 协议进行建模和分析。

4.1 ZG 协议建模

在 ZG 协议中,我们主要考虑两种安全特性:非否认性和

公平性。

发方非否认性描述为:

$askJSaysAsent(A, m) \rightarrow answerJSaysAsent(A, m)$ G1

收方非否认性描述为:

$askJSaysBreceived(B, m) \rightarrow answerJSaysBreceived(B, m)$ G2

公平性描述为:

$Fairness = Fairness_1 \wedge Fairness_2 = (NRO_1 \odot NRR_1) \wedge (NRO_2 \odot NRR_2)$ G3

其中:

NRO_1 为:

$askJSaysAsent(A, C) \rightarrow answerJSaysAsent(A, C)$

NRR_1 为:

$askJSaysBreceived(B, C) \rightarrow answerJSaysBreceived(B, C)$

NRO_2 为:

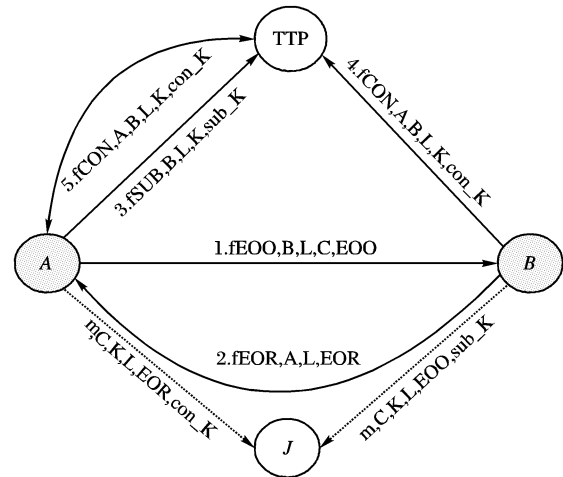
$askJSaysAsent(A, K) \rightarrow answerJSaysAsent(A, K)$

NRR_2 为:

$askJSaysBreceived(B, K) \rightarrow answerJSaysBreceived(B, K)$

在 ZG 协议中,假设 A 、 B 、 TTP 之间的信道为通用信道, A 、 B 与 J 之间的信道为专用信道。

ZG 协议的整体模型如图 2 所示。



注: \longrightarrow 通用信道
 \dashrightarrow 专用信道

A 、 B 有可能为恶意主体, TTP 和 J 永远诚实可信。

图2 ZG协议的整体模型

分三种情况分析:

1) 当 A 、 B 均为诚实主体时,严格按照协议规定的步骤执行。只需根据上节提出的建模方法分别对 A 、 B 、 TTP 和仲裁 J 进行描述即可。

2) 当 A 为恶意主体时,他一方面执行协议,另一方面实行恶意的行为,试图在协议中占据某种优势,他的这种恶意行为是不确定的。我们可以在正常描述 A 、 B 的诚实行为的同时,增加对 A 的恶意能力及初始知识的描述。

3) B 为恶意主体时与第 2 种情况类似。

4.2 ZG 协议模型的分析

将上述模型中得到的描述诚实主体和仲裁的规则(情况 2 和 3 中还包括描述恶意主体初始知识和计算能力的规则)合称为初始规则集合 B_0 ,连同待验证的安全目标一起输入验证器中进行验证。通过规则间的合一化消解等运算来验证某个安全性目标能否从规则基础中推导出来,从而实现对密码协议的非否认性、公平性验证。结合不同的主体模型,对 G1、G2、G3 分别进行了验证,结果发现:

1) 当 A 、 B 均为诚实主体时, $G1$ 、 $G2$ 、 $G3$ 均满足, 即 ZG 协议满足非否认性和公平性。

2) 当 A 为恶意主体时, $G1$ 、 $G2$ 满足, 说明协议满足非否认性。而 $G3$ 不满足, 说明协议不满足公平性。通过查看两个子阶段的公平性结果发现: $Fairness_1$ 满足, 即 A 、 B 分别拥有了 EOR 和 EOO ; $Fairness_2$ 不满足, 由于 A 不可能做出对自己不利的事情, 因此 A 得到了 con_K , 而 B 没有得到, A 在协议中占了优势。故, 该协议对 B 不公平。

3) 当 B 为恶意主体时, ZG 协议的非否认性和公平性均满足。

这个分析结果与文献[15]得到的结果一致, 表明了我们提出的模型的有效性和正确性。

5 结语

本文针对非否认协议提出了一个专门用于分析非否认性和公平性的一阶逻辑模型:

1) 引入了谓词 $agent(m, p(X, flag))$, 将消息和消息来源绑定在一起, 为解决非否认性和公平性问题奠定了基础。引入了 $match(a, b)$ 谓词描述对消息的检验和仲裁过程, 使协议刻画更加精确。

2) 在参与方和仲裁的描述中添加了对应性事件 $askJSays$ 和 $answerJSays$ 解决非否认性问题, 更接近于非否认协议实际。文献[13]将非否认性看作了认证问题来解决, 通过参与方之间的相互认证来判断非否认性是否满足, 这种方法不太精确, 有时会漏掉一些攻击。

3) 对公平性的定义充分考虑了协议运行中的各个阶段并形式化了“优势”的概念, 相比较于那些只考虑协议完成之后的状态更加精确, 更接近公平性的本质。

目前还没有专门针对非否认协议的一阶逻辑模型。通过具体的协议验证实例可以看出, 本文提出的模型是有效的; 同时, 通过比较还可以看出该模型具有针对性强、更加精确、更加接近于非否认协议实际的优点。下一步的工作将对本文提出的方法进行完善和扩展, 使其能够在安全特性不满足时构造出攻击路径, 并可以对带时间戳的非否认协议进行分析。

参考文献:

- [1] ZHOU J Y, GOLLMANN D. A fair non-repudiation protocol [C]// Proceedings of the 1996 IEEE Symposium on Security and Privacy. [S. l.]: IEEE Press, 1996: 55 – 61.
- [2] ASOKAN N. Fairness in Electronic Commerce [D]. Waterloo: University of Waterloo, 1998.
- [3] ASOKAN N, SHOUP V, WAIDNER M. Asynchronous protocols for optimistic fair exchange [C]// Proceedings of IEEE Symposium on Research in Security and Privacy. [S. l.]: IEEE Press, 1998: 86 – 99.
- [4] GARAY J A, JAKOBSSON M, MACKENZIE P. Abuse-free optimistic contract signing [C]// Proceedings of Advances in Cryptology (Crypto '99). Berlin: Springer-Verlag, 1999: 449 – 466.
- [5] BLANCHET B. An efficient cryptographic protocol verifier based on Prolog rules [C]// 14th IEEE Computer Security Foundations Workshop. [S. l.]: IEEE Press, 2001: 82 – 96.
- [6] SCHNEIDER S. Formal analysis of a non-repudiation protocol [C]// Proceedings of the 11th IEEE Computer Security Foundations Workshop. [S. l.]: IEEE Press, 1998: 54 – 65.
- [7] KREMER S, RASKIN J F. A game-based verification of non-repudiation and fair exchange protocols [C]// LARSEN KG, NIELSEN M, ed. 12th International Conference on Concurrency Theory, CONCUR. LNCS 2154. Berlin: Springer-Verlag, 2001: 551 – 565.
- [8] 蓝荣胜, 陈大伟, 郭云川, 等. 公平非否认协议的有限状态分析 [J]. 计算机科学, 2005, 32(8): 83 – 86.
- [9] ZHOU J Y, GOLLMANN D. Towards verification of non-repudiation protocols [C]// Proceedings of the 1998 International Refinement Workshop and Formal Methods Pacific. Berlin: Springer-Verlag, 1998: 370 – 380.
- [10] 范红, 冯登国. 一个非否认协议 ZG 的形式化分析 [J]. 电子学报, 2005, 33(1): 171 – 173.
- [11] 黎波涛, 罗军舟. Zhou-Gollmann 不可否认协议的一种新的改进 [J]. 计算机学报, 2005, 28(1): 35 – 45.
- [12] BELLA G, PAULSON L. Mechanical proofs about an on-repudiation protocol [C]// Proceedings of 14th International Conference on Theorem Proving in Higher Order Logic. Berlin: Springer-Verlag, 2001: 91 – 104.
- [13] Judson Santiago. Study for Automatically Analysing Non-repudiation [EB/OL]. [2006 – 10 – 10]. <http://www.avispa-project.org/papers/SantiagoV-CRISIS05.pdf>.
- [14] 卿斯汉, 李改成. 公平交换协议的一个形式化模型 [J]. 中国科学, 2005, 35(2): 161 – 172.
- [15] GURGINS S, RUDOLPH C. Security analysis of (un-) fair non-repudiation protocols [C]// Formal Aspects of Security (FASec'02). Berlin: Springer-Verlag, 2002: 97 – 114.
- [16] ABADI M, BLANCHET B. Computer-assisted verification of a protocol for certified email [C]// The 10th Int'1 Symposium (SAS'03), LNCS 2694. Berlin: Springer-Verlag, 2003.
- [17] 范钰丹, 韩继红, 王亚弟, 等. 非否认协议形式化分析技术 [J]. 计算机应用, 2006, 26(11): 2610 – 2614.

(上接第 2186 页)

参考文献:

- [1] 柳岸, 龙雅琴, 古乐野. 基于包过滤技术的网络安全的研究 [J]. 计算机应用, 2006, 26(9): 2160 – 2161.
- [2] 刘更楼, 丁常福, 姜建国. 基于状态检测的防火墙系统研究 [J]. 航空计算技术, 2004, 34(1): 122 – 125.
- [3] 王栋. 防火墙深度包检测技术研究 [M]. 西安: 西安电子科技大学, 2005.
- [4] DENNING D. An intrusion detection model [J]. IEEE Transactions on Software Engineering, 1987, SE-13(2): 222 – 223.
- [5] COMMENTZ-WALTER B. A string matching algorithm fast on the average [C]// Proceedings of the 6th International Colloquium on Automata, Language and Programming. LNCS 71. Berlin: Springer-Verlag, 1979: 118 – 132.
- [6] WU S, MANBER U. A fast algorithm for multi-pattern searching, TR-94-17 [R]. Arizona: University of Arizona, 1994.
- [7] TUCK N, SHERWOOD T, CALDER B, et al. Deterministic memory-efficient string matching algorithms for intrusion detection [C]// Proceedings of IEEE Infocom. [S. l.]: IEEE Press, 2004: 333 – 340.
- [8] THOMPSON K. Programming techniques: regular expression search algorithm [J]. Communications of the ACM, 1968, 11(6): 419 – 422.
- [9] 卢开澄. 计算机算法导引 [M]. 北京: 清华大学出版社, 1996.
- [10] LEVANDOSKI J, SOMMER E, STRAIT M. Application layer packet classifier for Linux [EB/OL]. [2006 – 12 – 10]. <http://l7-filter.sourceforge.net/>.
- [11] PAXSON V. Flex: A Fast Scanner Generator: version 2.5 [EB/OL]. [1998 – 12 – 10]. http://www.gnu.org/software/flex/manual/html_mono/flex.html.