ISSN1000-9825 Journal of Software 软件学报 2000,11(2):213~221

安全协议的验证逻辑*

白硕 隋立颖 陈庆锋 付岩 庄超

(国家智能计算机研究开发中心 北京 100080)

E-mail: bai@ncic.ac.cn

摘要 该文提出一种论证安全协议之安全性质的非单调动态逻辑.针对信息安全的特定需要,给出了一组与加密、解密、签名、认证和密钥分配等密码学操作有关的公理和推理规则,举例说明了这一逻辑框架在验证安全协议方面的应用,并讨论了需要进一步解决的问题.

关键词 信息安全,协议验证,动态逻辑,非单调逻辑.

中图法分类号 TP309

当前,信息化是一个世界范围的大趋势.以 Internet 为代表的国际联网的热潮正在向社会的每一个角落渗透.在这一大趋势中,信息安全问题成为学术界和工业界共同关注的关键问题.一般来说,信息安全包括访问控制、通信安全和信息监察.在访问控制和通信安全方面,以防火墙为主要防范手段、以局域网为保护对象的 Intranet和以加密通信为主要防范手段、以广域网上虚拟私用网为保护对象的 Extranet 已经走向实用.在网络体系的各个层次上,各种为保障加密通信而制定的安全协议正在许多对信息安全敏感的领域发挥着重要的作用.

一个安全协议在出台以后,需要从多方面对其安全性质进行验证.验证的手段有多种,比如,直观的分析、实际攻击手段的测试,还有形式化的逻辑推理.其中,后者以其手段的严密和断言的普遍适用见长,不仅在学术界独树一帜,也对工业界产生了一定的影响.

我们知道,安全协议是由参与通信的各方按确定的步骤做出一定的通信动作完成的.这些通信动作实现了通信本身,而动作的内容则隐含了一些密码学变换算法的实施.这些密码学变换算法,从数学上提供了达到一定程度的通信安全的基本机制.抛开算法本身的数学内容不谈,而对达到通信安全的机制在逻辑抽象的层次上进行讨论,就产生了安全协议验证的形式化方法.形式化方法在信任算法本身的安全强度的前提下,可以忽略算法的数学细节来讨论协议的安全性质,通过形式化的逻辑推理来寻找安全协议的漏洞.因此,它是信息安全领域中一个受到重视的研究方向.

为了对安全协议进行形式化的逻辑推理,首先必须对安全协议中所涉及的各种对象、状态和关系,如实体、消息、动作、知识状态和信任关系等进行逻辑抽象.这一步是十分关键的.不同的形式化框架之间的差别也正是在于逻辑抽象的不同.逻辑抽象的作用在于使关于安全性质的讨论能够在与算法的数学细节无关的层次上进行,以便直接理解安全问题的核心,展现问题的逻辑本质.

前人在安全协议的验证逻辑方面做了许多重要的工作,其中由 Burrows,Abadi 和 Needham 提出的 BAN 逻辑[1]是最经典的用于分析安全协议的逻辑.BAN 逻辑具有概念清晰、简单、易理解、易应用的特点,应用它可以发现安全协议中某些不易察觉的漏洞.但 BAN 的逻辑处理能力有限(即对于某些协议,BAN 逻辑的分析结果是错误的),且没有考虑窃听者的存在.并且由于 BAN 逻辑假设环境不受丢失消息或宿主机崩溃的影响,而在分布

* 本文研究得到国家 863 高科技项目基金(No.863-306-ZD-10-02)资助.作者**白硕**,1956 年生,博士,研究员,博士生导师,主要研究领域为人工智能,计算语言学,Internet/Intranet应用软件.**隋立颖**,女,1973 年生,硕士,主要研究领域为 Internet/Intranet应用软件,计算机理论.**陈庆锋**,1971 年生,助理工程师,主要研究领域为信息安全,电子商务.**付岩**,1974 年生,博士,主要研究领域为面向对象系统,Internet 应用.**庄超**,1968 年生,博士,主要研究领域为 Internet 内容版权保护,安全协议.

本文通讯联系人:白硕,北京 100080,国家智能计算机研究开发中心

本文 1998-04-03 收到原稿,1999-03-22 收到修改稿

式环境下,这种情况是很容易发生的,所以 BAN 逻辑的不足之处在于,逻辑无法确保保密的东西最终还是保密的.随后提出的逻辑,如 GNY[2],UEPS[3]等试图抓住加密领域的每个细节,扩充了分析能力,但这些逻辑本身太复杂而且不易应用.作为一种折衷,AUTLOG[4]逻辑应运而生.它在 BAN 逻辑的基础之上作了一些简单的扩充,如加入了"recognize"谓词,扩大了"see"操作符的涵义,模拟了一个监听者等.这些扩充并未使逻辑本身复杂化,同时又能分析一些 BAN 逻辑所不能处理的协议.但 AUTLOG 并未从根本上解决问题,因此我们仍然只有两种选择:或者用 GNY 等复杂逻辑来分析所有的协议,或者为每一协议设计与其相应的简单逻辑来进行分析.

而 Kailar 在"Accountability in Electronic Commerce Protocols"[5]中则直接针对电子商务提出了一种用于分析要求"责任性"的安全协议的新框架.它克服了以往逻辑只局限于分析密钥管理协议或其他一些认证协议的弱点,涉及进行电子商务的各个步骤,着重强调了进行电子商务的双方对对方的责任性,即对自己所发出的消息的不可否认性.Kailar 所提出的方法比 BAN 逻辑更加自然.

我们认为,从逻辑上看,通信安全问题具有动态性和非单调性.所谓动态性是指,安全协议本身是一个动作序列,因此,应该引入动态逻辑的机制,通过对动作的前提条件和后果进行表示和推理来整体把握安全协议作为动作序列的语义.所谓非单调性是指,有些安全性质是因为在仅有的前提和目前完成的动作情况下证明不了它不成立而默认它成立的,一旦前提扩大或者动作继续,这些性质的成立与否是会随着条件的变化而发生变化的,因此,应该引入非单调逻辑的机制,通过"失败即否定"等原则作为演绎推理的补充,来验证安全协议的非单调性质.总体来看,我们需要一种把普通的一阶逻辑和动态逻辑、非单调逻辑综合起来的逻辑框架.

本文的第 1 节介绍我们提出的安全问题的逻辑抽象.第 2 节介绍我们提出的公理、推理规则和便捷的标准推理格式.第 3 节给出 3 个安全协议的验证示例.第 4 节对本文内容作出总结并提出有待进一步研究的问题.

1 安全问题的逻辑抽象

在从逻辑角度讨论安全问题的时候,首要的任务是对所涉及到的领域进行逻辑抽象,即抽掉问题与算法的与数学细节有关的方面,保留反映安全问题的逻辑实质的方面.比如,当事各方的知识状态和信任关系随安全协议的执行步骤展开而发生的各种变化等等.在对安全问题的逻辑抽象中,我们会涉及到如下一些概念.

实体 实体是参加通信的主体,是收发消息并进行加密、解密、签名、认证和分配密钥的主体,也是对消息的知晓和对其他实体的信任的主体.其中,一些实体在密钥分配的过程中占有重要的地位,这就是所谓证书授权当局(certificate authorities,简称 CA).证书授权当局通过数字签名来向其管辖的通信范围确认参加通信的实体的公钥与其身份相符,从而确认其公钥的合法性.当然,在一个完备的公钥基础结构中,CA 的合法性还需要更高层的授权当局的确认.对这样的结构,我们将另文讨论,本文只限于讨论有唯一的 CA 的情况.

消息 消息是通信活动的对象,也是密码学操作的对象.安全通信的全部任务就是使某些消息"让该知道的人知道,不该知道的人不知道".密钥是密码学操作的核心,从逻辑抽象的观点看,密钥也是消息,也可以被另外的密钥加密并付诸传输.与特定实体相联系的密钥(如"A 的签名公钥","B 的数据交换私钥"等),逻辑上可以看成相关的密钥函数符号作用于该实体的表达式.从语义上说,这要求在协议执行周期内,与实体相关联的密钥不发生变化.在证书中,实体的身份也以消息的形态出现.密码学操作的结果(密文)也表现为消息.由于实体和作为消息的实体的身份以同样的形式表示并不会产生混淆,故无必要采用多种类逻辑对二者进行区分.

动作 动作是通信活动的构成单位.我们关心的基本动作只有两个:生成(generate)和发送(send).基本动作可以串行组合成动作序列.安全协议就是一些特定的动作序列(单钥加密、公钥加密、数字签名等密码学操作是把一些消息映射为另外一些消息的密码学变换,在逻辑上用一些特殊的函数符号来表示,不属于"动作"的范畴).动作在推理中的作用是改变当事的实体的知识状态以及实体之间的信任关系.

知识状态 知识状态是实体与消息之间的一种关系,即"知道"关系.我们在本文中处理的是一种积累性的"知道"关系.从语义上说,这可以理解为在协议执行周期内,一个实体"知道"的消息不会因任何动作的发生而遗忘或废除,变为"不知道".

信任关系 本文所说的信任关系特指因数字签名和认证而产生的信任关系,即一个带有数字签名的消息经认证后可以被认为:(1) 确系签名者所发;(2) 收到的消息确如发出时一样,没有经过任何篡改.

上述概念,经形式化表示,就构成了我们的逻辑框架NDL(non-monotonic dynamic logic)的语法规则.

NDL的语言构成包括个体常元、函数词、谓词、动作、断言和缩略表示6种成分.

个体常元 用大写字母(或加下标)表示.CA是一个特殊的个体常元,表示证书授权当局.

函数词 NDL的函数符号包括下面这些表示加密、签名等密码学操作和关联密钥的映射关系:

$E(m,k)$	以 k 为单钥对 m 加密的结果;
$S(m,k)$	以 k 为互逆的公钥密钥之一对 m 加密的结果;
$H(m)$	m 的消息文摘;
$Kpb(x)$	x 的数据交换公钥;
$Kpv(x)$	x 的数据交换私钥;
$Spb(x)$	x 的签名公钥;
$Spv(x)$	x 的签名私钥;
$\langle m_1,\ldots,m_n\rangle$	把消息 $m_1,\ldots,m_n$ 捆绑而成的、作为一个消息的整体.

谓词 它有两个,分别表示相关的知识状态和信任关系:

$Know(x,m)$	x 知道 m;
$Auth(x,y,m)$	x 确认 m 为 y 所发且发后未经修改.

动作 基本动作有如下两个:

$Generate(x,m)$	x 生成消息 m;
$Send(x,y,m)$	x 向 y 发送 m.

动作序列可递归地定义如下:(1) 基本动作是动作序列;(2) 如果 α,β 是动作序列,则 $\alpha\circ\beta$ 也是动作序列.协议就是一种动作序列.

断言 断言的一般形式为

$$P\vdash_{\alpha}Q,$$

其中 P,Q 为公式集合,α 为动作序列.这个断言表示,如果 P 成立,则可以执行动作序列 α,且动作序列执行完毕后,Q 成立.

缩略表示 一些重要的固定函数复合在安全协议中多次重复出现,且有明确的概念.对这些固定的函数复合,我们用一些缩略表示进行重写定义.在本文的逻辑系统里,我们定义了以下4个缩略表示:

$Sign(x,m)=\langle m,S(H(m),Spv(x))\rangle$,$x$ 对消息 m 的带正文数字签名;

$So(x,m)=S(H(m),Spv(x))$,x 对消息 m 的脱正文数字签名;

$CertK(x)=Sign(CA,\langle x,Kpb(x)\rangle)$,$x$ 的数据交换公钥证书;

$CertS(x)=Sign(CA,\langle x,Spb(x)\rangle)$,$x$ 的签名公钥证书.

在下文中,可以看到它们的具体应用.

2 推 理

本文提出的推理框架由公理、推理规则和推理格式构成.下面分小节逐一介绍.

2.1 公 理

(1) 加密公理

1-1 $Know(x,m)\wedge Know(x,k)\rightarrow Know(x,E(m,k))$

1-2 $Know(x,m)\wedge Know(x,Kpb(y))\rightarrow Know(x,S(m,Kpb(y)))$

(2) 密钥分配公理

2-1 $Know(x,Kpb(CA))$

2-2 $Know(x,Spb(CA))$

2-3 $Know(x,Kpv(x))$

2-4　$Know(x,Spv(x))$

(3) 解密公理

3-1　$Know(x,k)\wedge Know(x,E(m,k))\rightarrow Know(x,m)$

3-2　$Know(x,Kpv(y))\wedge Know(x,S(m,Kpb(y)))\rightarrow Know(x,m)$

(4) 签名公理

4-1　$Know(x,m)\rightarrow Know(x,H(m))$

4-2　$Know(x,m)\wedge Know(x,Spv(y))\rightarrow Know(x,S(H(m),Spv(y)))$

(5) 认证公理

5-1　$Know(x,m)\wedge Know(x,S(H(m),Spv(y)))\wedge Know(x,Spb(y))\rightarrow Auth(x,y,m)$

5-2　$Know(x,m)\wedge Auth(x,y,H(m))\rightarrow Auth(x,y,m)$

(6) 分合公理

6-1　$Know(x,m_1,\ldots,m_n))\leftrightarrow Know(x,m_2)\wedge\ldots\wedge Know(x,m_n)$

6-2　$Auth(x,y,\langle m_1,\ldots,m_n\rangle)\rightarrow Auth(x,y,m_1)\wedge\ldots\wedge Auth(x,y,m_n)$

从以上公理并结合缩略表示的函数 $So()$和 $Sign()$的定义,可以证明两个稍后会用到的比较实用的定理.

定理 1. $Know(x,m)\wedge Know(x,So(y,m))\wedge Know(x,Spb(y))\rightarrow Auth(x,y,m)$.

证明:

(1)	$Know(x,m)$	[前提]
(2)	$Know(x,So(y,m))$	[前提]
(3)	$Know(x,Spb(y))$	[前提]
(4)	$Know(x,S(H(m),Spv(y)))$	(2)[定义]
(5)	$Auth(x,y,m)$	(1)(3)(4)[5-1]
(6)	$Know(x,m)\wedge Know(x,So(y,m))\wedge Know(x,Sph(y))\rightarrow Auth(x,y,m)$	(1)(2)(3)(5)[$\rightarrow_+$]

□

定理 1 实际上是公理 5-1 的另一个等价的写法,其成立是十分显然的.

定理 2. $Know(x,Sign(y,m))\wedge Know(x,Spb(y))\rightarrow Auth(x,y,m)$.

证明:

(1)	$Know(x,Sign(y,m))$	[前提]
(2)	$Know(x,Spb(y))$	[前提]
(3)	$Know(x,\langle m,S(H(m),Spv(y))\rangle)$	[定义]
(4)	$Know(x,m)$	(3)[6-1]
(5)	$Know(x,S(H(m),Spv(y)))$	(3)[6-1]
(6)	$Know(x,So(y,m))$	(5)[定义]
(7)	$Auth(x,y,m)$	(2)(4)(6)[定理 1]
(8)	$Know(x,Sign(y,m))\wedge Know(x,Kpb(y))\rightarrow Auth(x,y,m)$	(1)(2)(7)[$\rightarrow_+$]

□

定理 2 反映的是认证的更典型的形式.

2.2 规　则

(R-1) 信息泄露规则

$$Know(x,m)\vdash_{Send(x,y,m)}Know(z,m).$$

这条规则的意思是,明文的消息无密可保.实际上,我们利用这个泄露规则,为模拟任何监听者提供了方便的逻辑表示.

(R-2) 自产自知规则

$$\vdash_{Generate(x,m)}Know(x,m).$$

这条规则的意思是,自己生成的消息自己一定知道.

(R-3) 积累规则

$$\frac{P\vdash Q}{P\vdash_{\alpha} Q}.$$

这条规则的意思是,已经证明成立的结论在做过任何动作后继续保持成立.

必须指出,积累规则对于某些安全协议是不适用的.问题出在两个地方:(1) 在协议执行过程的中途更换密钥的情况下,实体对原密钥的知识已经"作废",因此不满足"积累"的特性;(2) 在对某些消息没有"记忆"功能的情况下,原来对这些消息的知识可能被"遗忘",也不满足"积累"的特性.因此,本文提出的逻辑系统,只适用于这样的安全协议:(1) 协议执行中途不可更换密钥;(2) 实体具有贯穿协议执行过程始终的记忆功能.对于不符合这两个条件的协议,必须用另外的规则代替积累规则,对此我们将另文阐述.

(R-4) 合成规则

$$\frac{P\vdash_{\alpha} Q, Q\vdash_{\beta} R}{P\vdash_{\alpha\cdot\beta} R}.$$

这条规则的意思是,两个动作序列的合成效果相当于把前一个动作序列的后果作为后一个动作序列的前提、把后一个动作序列的结论当作合成动作序列的结论来使用.

由于合成规则的成立,可以把动作序列的合成看成一个具有结合性的代数运算.它还具有单位元,即"空序列".

(R-5) 非单调规则

$$\frac{\text{推不出 } P\vdash_{\alpha} Know(x,m)}{P\vdash\sim_{\alpha}\neg(Know(x,m)}.$$

这条规则的意思是,如果不能推出在前提 P 下做动作 α 后 x 知道 m,那么就可以非单调地假定 x 不知道 m.这里"非单调"的意思可以这样理解:如果"仅知道"P 和 α,那"推不出 x 知道 m 就假定 x 不知道 m"是合理的;但如果随着 P 的扩大或者 α 的加长,也许就可以推出 x 知道 m 了,那时原来的非单调假设就要按照新的情况进行修正.这正好反映了信息安全领域中的一个典型现象:我们很难肯定地、无条件地断定谁不知道什么,但可以默认如果"知道"的途径仅限于某几种的话,有些人是应该不知道某些消息的.

2.3 推理格式

由于上述的推理对连续发生的动作具有积累性,则可以有下列推理格式:

ε(代表空动作序列)

p_{01}
p_{02}
⋮
p_{0m_0}

α_1

p_{11}
p_{12}
⋮
p_{1m_1}

α_2

p_{21}
p_{22}
⋮
p_{2m_2}
⋮

α_n

p_{n1}
p_{n2}
⋮
p_{nm_n}

上述推理格式表示了以下的推理关系：

$$\{p_{01},\ldots,p_{0m_0}\}\vdash_{a_1}\{p_{11},\ldots,p_{1m_1}\}$$

$$\{p_{01},\ldots,p_{0m_0}\}\vdash_{a_1\cdot a_2}\{p_{21},\ldots,p_{2m_2}\}$$

$$\vdots$$

$$\{p_{01},\ldots,p_{0m_0}\}\vdash_{a_1\cdot a_2\cdot\ldots\cdot a_n}\{p_{n1},\ldots,p_{nm_n}\}$$

关于非单调结论的推导，由于众所周知的一阶逻辑下可证关系的不可判定性，我们甚至不可能把相对于“$\vdash$”推不出什么这种断言的正面根据在一阶逻辑范围内形式化. 在基于纯一阶逻辑的非单调逻辑里，反面的做法是构造一个赋值使前提和非单调结论共同为真. 如果这样的赋值能够构造成功，非单调结论即获得认可；但如果构造过程不终止则无结论. 而在基于动态逻辑的非单调逻辑里，要对付相对于“$\vdash_a$”推不出什么这种断言，连这一点也做不到，只能采取“失败即否定”这种直观的办法. 当然，本文提出的是一种具有积累规则的非单调动态逻辑，因此可以通过把动作转写成谓词来构造一个“虚拟的”基于纯一阶逻辑的非单调逻辑. 这个虚拟的逻辑系统可以借助 Prolog 语言本身的带有“失败即否定”语义的推理机制来实现其并不复杂的非单调性.

3 安全协议推导和验证示例

下面我们针对 3 个实例来说明如何在 NDL 中推导安全协议的性质.

例 1：已知：

$P=\{Know(Alice,Kpb(Bob))\}$,

$a=Generate(Alice,m)\circ Generate(Alice,k)\circ Send(Alice,Bob,\langle E(m,k),S(k,Kpb(Bob))\rangle)$,

$Q=\{Know(Bob,m)\}$.

求证：$P\vdash_a Q$

证明：

(1)	$Know(Alice,Kpb(Bob))$	[前提]
(2)	$Generate(Alice,m)$	[动作]
(3)	$Know(Alice,m)$	(2)[R-2]
(4)	$Generate(Alice,k)$	[动作]
(5)	$Know(Alice,k)$	(2)[R-2]
(6)	$Know(Alice,E(m,k))$	(3)(4)[1-1]
(7)	$Know(Alice,S(k,Kpb(Bob)))$	(5)(1)[1-2]
(8)	$Know(Alice,\langle E(m,k),S(k,Kpb(Bob))\rangle)$	(6)(7)[6-1]
(9)	$Send(Alice,Bob,\langle E(m,k),S(k,Kpb(Bob))\rangle)$	[动作]
(10)	$Know(Bob,\langle E(m,k),S(k,Kpb(Bob))\rangle)$	(9)[R-1]
(11)	$Know(Bob,E(m,k))$	(10)[6-1]
(12)	$Know(Bob,S(k,Kpb(Bob)))$	(10)[6-1]
(13)	$Know(Bob,Kpv(Bob))$	[2-3]
(14)	$Know(Bob,k)$	(12)(13)(14)[3-2]
(15)	$Know(Bob,m)$	(11)(15)[3-1]

式(15)即所求证的结论. □

例 1 的通信和密码学变换机制可以用图 1 来表示.

例 2：已知：

$P=\{Know(A,CertS(A)),Know(A,Kpb(B))\}$,

$a=Generate(A,m)\circ Generate(A,k)\circ Send(A,B,\langle E(m,k),S(\langle k,So(A,m),CertS(A)\rangle,Kpb(B))\rangle)$,

$Q=\{Know(B,m),Auth(B,A,m),Auth(B,CA,\langle A,Spb(A)\rangle)\}$.

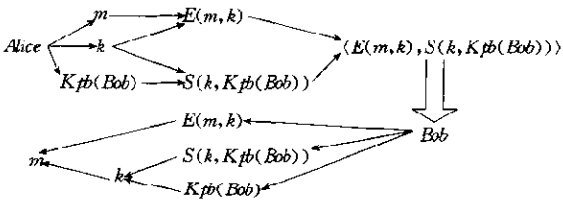

Fig. 1
图1

求证:$P \vdash_o Q$

证明:

(1) $Know(A,CertS(A))$ [前提]

(2) $Know(A,Kpb(B))$ [前提]

(3) $Generate(A,m)$ [动作]

(4) $Know(A,m)$ (3)[R-2]

(5) $Generate(A,k)$ [动作]

(6) $Know(A,k)$ (5)[R-2]

(7) $Know(A,E(m,k))$ (4)(6)[1-1]

(8) $Know(A,Spv(A))$ [2-3]

(9) $Know(A,So(A,m))$ (4)(8)[4-2]

(10) $Know(A,\langle k,So(A,m),CertS(A)\rangle)$ (6)(9)(1)[6-1]

(11) $Know(A,S(\langle k,So(A,m),CertS(A)\rangle,Kpb(B)))$ (10)(2)[1-2]

(12) $Know(A,\langle E(m,k),S(\langle k,So(A,m),CertS(A)\rangle,Kpb(B))\rangle)$ (7)(11)[6-1]

(13) $Send(A,B,\langle E(m,k),S(\langle k,So(A,m),CertS(A)\rangle,Kpb(B))\rangle)$ [动作]

(14) $Know(B,\langle E(m,k),S(\langle k,So(A,m),CertS(A)\rangle,Kpb(B))\rangle)$ (13)[R-1]

(15) $Know(B,E(m,k))$ (14)[6-1]

(16) $Know(B,S(\langle k,So(A,m),CertS(A)\rangle,Kpb(B))$ (14)[6-1]

(17) $Know(B,Kpv(B))$ [2-3]

(18) $Know(B,\langle k,So(A,m),CertS(A)\rangle)$ (15)(16)(17)[3-2]

(19) $Know(B,k)$ (18)[6-1]

(20) $Know(B,So(A,m))$ (18)[6-1]

(21) $Know(B,CertS(A))$ (18)[6-1]

(22) $Know(B,m)$ (15)(19)[3-1]

(23) $Know(B,\langle A,Spb(A)\rangle)$ (21)[定义,6-1]

(24) $Know(B,Spb(A))$ (23)[6-1]

(25) $Auth(B,A,m)$ (20)(22)(24)[定理1]

(26) $Know(B,Sign(CA,\langle A,Spb(A)\rangle))$ (21)[定义]

(27) $Know(B,Spb(CA))$ [2-2]

(28) $Auth(B,CA,\langle A,Spb(A)\rangle)$ [定理2]

式(22)、(25)、(28)即所求证之结论. □

本例是综合应用单钥加密解密、公钥加密解密、数字签名与认证、公钥证书等手段的一个安全协议的验证过程.其中,(25)和(28)两式合起来,具有很明显的认证效果:m 是由 A 用自己的签名公钥签过名的,而 A 的签名公钥连同 A 的身份说明又是用 CA 的签名公钥签过名的.这样,只要 B 信任 CA,B 就有理由相信 $Spb(A)$ 确系 A 的签名公钥,从而确信 A 发过 m.

例 3:双重签名.

这是一个涉及到 3 方的安全协议,是从安全电子交易 SET(secure electronic transactions)协议中截取的片段并作了一定的简化. 已知:

$P=\{Know(B,Spb(A)),Know(C,Spb(A))\}$,

$\alpha=Generate(A,m_1)\circ Generate(A,m_2)\circ$

$Send(A,B,\langle m_1,H(m_2),So(A,\langle H(m_1),H(m_2)\rangle)\rangle)\circ$

$Send(A,C,\langle m_2,H(m_1),So(A,\langle H(m_1),H(m_2)\rangle)\rangle)$,

$Q=\{Auth(B,A,m_1),Auth(C,A,m_2),Auth(B,A,H(m_2)),Auth(C,A,H(m_1))\}$.

求证:$P\vdash_\alpha Q$

证明:

(1)	$Know(B,Spb(A))$	[前提]
(2)	$Know(C,Spb(A))$	[前提]
(3)	$Generate(A,m_1)$	[动作]
(4)	$Know(A,m_1)$	(3)[R-2]
(5)	$Generate(A,m_2)$	[动作]
(6)	$Know(A,m_2)$	(5)[R-2]
(7)	$Know(A,H(m_1))$	(4)[4-1]
(8)	$know(A,H(m_2))$	(6)[4-1]
(9)	$Know(A,\langle H(m_1),H(m_2)\rangle)$	(7)(8)[4-1]
(10)	$Know(A,So(A,\langle H(m_1),H(m_2)\rangle))$	(9)[4-2]
(11)	$Know(A,\langle m_1,H(m_2),So(A,\langle H(m_1),H(m_2)\rangle)\rangle)$	(4)(8)(10)[4-1]
(12)	$Send(A,B,\langle m_1,H(m_2),So(A,\langle H(m_1),H(m_2)\rangle)\rangle)$	(11)[动作]
(13)	$Know(A,\langle m_2,H(m_1),So(A,\langle H(m_1),H(m_2)\rangle)\rangle)$	(6)(7)(10)[4-1]
(14)	$Send(A,C,\langle m_2,H(m_2),So(A,\langle H(m_1),H(m_2)\rangle)\rangle)$	(13)[动作]
(15)	$Know(B,\langle m_1,H(m_2),So(A,\langle H(m_1),H(m_2)\rangle)\rangle)$	(12)[R-1]
(16)	$Know(C,\langle m_2,H(m_1),So(A,\langle H(m_1),H(m_2)\rangle)\rangle)$	(14)[R-1]
(17)	$Know(B,m_1)$	(15)[6-1]
(18)	$Know(B,H(m_1))$	(17)[4-1]
(19)	$Know(B,H(m_2))$	(15)[6-1]
(20)	$Know(B,\langle H(m_1),H(m_2)\rangle)$	(18)(19)[6-1]
(21)	$Know(B,So(A,\langle H(m_1),H(m_2)\rangle))$	(15)[6-1]
(22)	$Auth(B,A,\langle H(m_1),H(m_2)\rangle)$	(20)(21)[定理 1]
(23)	$Auth(B,A,H(m_1))$	(22)[6-2]
(24)	$Auth(B,A,H(m_2))$	(22)[6-2]
(25)	$Auth(B,A,m_1)$	(17)(23)[5-2]
(26)	$Know(C,m_2)$	(16)[6-1]
(27)	$Know(C,H(m_2))$	(26)[4-1]
(28)	$Know(C,H(m_1))$	(16)[6-1]
(29)	$Know(C,\langle H(m_1),H(m_2)\rangle)$	(28)(27)[6-1]
(30)	$Know(C,So(A,\langle H(m_1),H(m_2)\rangle))$	(16)[6-1]
(31)	$Auth(C,A,\langle H(m_1),H(m_2)\rangle)$	(29)(30)[定理 1]
(32)	$Auth(C,A,H(m_1))$	(31)[6-2]
(33)	$Auth(C,A,H(m_2))$	(31)[6-2]

(34) $Auth(C,A,m_2)$ (27)(33)[5-2]

式(24)、(25)、(33)、(34)即所求证之结论. □

从双重签名中我们可以看到,我们的两个认证公理的不同作用.公理 5-1 及其派生出来的定理 1 和定理 2 的作用是从签名直接进行认证.公理 5-2 的作用则是通过对文摘的认证和原文来形成对原文的间接认证.后一种间接认证的方式是完全合理的.通俗地说,就是:如果你不能抵赖你在某原文的文摘上进行过数字签名,而我又获得了该原文,那我就有理由认为你不能抵赖你发过了该原文.如果从我获得的该原文做出的文摘和已经确认无改动的文摘相符,那我就有理由认为该原文也没有改动过.可以看出,没有类似于公理 5-2 这样的公理,要想论证双重签名所承诺的安全性质是有很大困难的.人的直观分析在这种细微之处很不容易把握,而有了明确的、合理的公理,验证起来就方便多了.

4 结 论

以上介绍了我们提出的非单调动态逻辑系统 NDL.这个系统通过一套基于"一阶逻辑+动态逻辑+非单调逻辑"的公理和推理规则,界定了推导安全协议的性质的一种新的形式化方法.通过协议验证的实例,我们看到,这样的方法确实能够对很大一类安全协议所承诺的安全性质进行验证.特别是在积累规则适用的情况下,整个验证过程可以用 Prolog 程序实现出来,包括其中的非单调部分.

当然,积累规则并不是对所有安全协议都适用.确实有一些涉及到协议执行的中途发生密钥更改和遗忘的情况需要更复杂的逻辑机制来处理.对此我们需要进行本文范围以外的专门研究.

本文工作的重点是逻辑方法的应用.关于这个逻辑系统自身的形式化性质(比如语义等),也需要进行本文范围以外的专门研究.

按照本文的方法,结合电子交易本身的一些领域知识,可以对一些规模比较大的电子交易协议,比如说 SET[6],进行全面的、严格的验证.我们已经对 SET 的一部分业务流程成功地进行了这样的验证,推广到其他部分的工作正在进行之中.

参考文献

1 Michael B, Martin Abadi, Roger Needham. A logic of authentication. ACM Transactions on Computer System, 1990, 8 (1):18~36

2 Abadi M, Tuttle M. A semantics for a logic of authentication. In: Proceedings of the 10th Annual ACM Symposium on Principles of Distributed Computing. Montreal: ACM Press, 1991. 201~216

3 Kailar R, Gligor V. On belief evolution in authentication protocols. In: Catherine Harris, Madallum A C eds. Proceedings of 4th IEEE Computer Security Foundations Workshop. Los Alamitors, CA: IEEE Computer Society Press, 1991. 103~116

4 Volker Kessler, Cabriele Wedel. AUTLOG——an advanced logic of authentication. In: Bob Wernered ed. Proceedings of the 7th IEEE Computer Security Foundations Workshop. Los Alamitors, CA: IEEE Computer Society Press, 1994. 90~99

5 Kailar R. Accountability in electronic commerce protocols. Proceedings of IEEE Transactions on Software Engineering, 1996, 22(5):313~328

6 SET Secure Electronic Transaction Specification. Book 1: Business Description, Version 1.0. 1997

The Verification Logic for Secure Protocols

BAI Shuo SUI Li-ying CHEN Qing-feng FU Yan ZHUANG Chao

(National Research Center for Intelligent Computing Systems Beijing 100080)

Abstract In this paper, a non-monotonic dynamic logic that verifies properties of security protocols is introduced. In accordance with the specific requirement of information security, it provides axioms and inference rules about various cryptographic operations such as encryption, decryption, signature, authentication and key assignment. Several instances are given to illustrate its applications in security protocol verification. Open problems for further study are also discussed.

Key words Information security, protocol verification, dynamic logic, non-monotonic logic.