第 29 卷 第 9 期
2006 年 9 月
计 算 机 学 报
CHINESE JOURNAL OF COMPUTERS
Vol. 29 No. 9
Sept. 2006

实例化空间：一种新的安全协议验证逻辑的语义模型

苏开乐[1)] 岳伟亚[1)] 陈清亮[1),2)] ZHENG Xi-Zhong[2),3)]

[1)](中山大学计算机科学系 广州 510275)

[2)](勃兰登堡州理工大学计算机科学系 科特布斯 03046 德国)

[3)](江苏大学计算机系 镇江 212013)

摘 要 给出了一个称为"实例化空间(instantiation space)"的安全协议验证逻辑的语义模型. 该语义模型是建立在一种自然的加密信息交换(cryptographical message exchange)模型上的. 在此语义模型基础上,文章提出了一系列与安全属性相关的验证公理,由此可以证明它们在此语义模型下的正确性. 更重要的是,在此语义下的公理集在算法上是完全可以实现的,其对应的工具 SPV(Security Protocol Verifier)已经开发成功,并且可以验证复杂的协议. 在这套安全协议验证模型理论下,可以很方便地处理包括公钥、私钥、共享密钥和 Hash 函数组成的复杂信息格式. 而且,在此语义基础上的公理集是纯命题逻辑的,因此所需要的验证目标可以很方便地转化成可满足性问题(SAT),从而可以利用工业上快速高效的 SAT 求解器实现.

关键词 实例化空间; 加密信息交换模型;可满足性问题

中图法分类号 TP309

Instantiation Space: A New Model for Security Protocol Logic

SU Kai-Le[1)] YUE Wei-Ya[1)] CHEN Qing-Liang[1),2)] ZHENG Xi-Zhong[2),3)]

[1)](*Department of Computer Science, Sun Yat-Sen University, Guangzhou* 510275)

[2)](*Department of Computer Science, Brandenburg University of Technology*, 03046 *Cottbus, Germany*)

[3)](*Department of Computer Science, Jiangsu University, Zhenjiang* 212013)

Abstract The authors present a new model of security protocol logics, which is called Instantiation Space. This model is formally well grounded, based on the so-called Cryptographical Message Exchange (CME) model. A set of interesting axioms (security properties) can be shown to be sound within the Instantiation Space model. It is worth noting that, this set of axioms is algorithmic manageable, and hence a new Security Protocol Verifier (SPV) can be designed and implemented, particularly focused on automatic verification of complex protocols. The presented approach to security protocol verification is flexible enough to deal with complex message formats with arbitrarily nested encryptions by public, private, shared and hash keys as well as freshly generated keys. Moreover, the set of axioms for a protocol the authors generated based on Instantiation Space model is in propositional logic; for a lot of concerned security properties the verification problems can thus be formulated as satisfiability ones, which can be solved by efficient modern SAT solvers.

Keywords instantiation space; cryptographical message exchange (CME) model; satisfiability (SAT)

收稿日期:2006-04-01;**修改稿收到日期**:2006-05-30. **本课题得到国家"九七三"重点基础研究发展规划项目基金**(2005CB321902)、**国家自然科学基金**(60496327,10410638,60473004)、**广东省自然科学基金团队项目**(04205407)、**教育部留学回国人员科研启动基金**、**教育部"新世纪优秀人才"支持计划基金**、**德国研究基金**(DFG) 446 CHV113/240/0-1、**中山大学"凯思"基金以及上海市智能信息处理重点实验室(筹)开放课题资助**. **苏开乐**,男,1964 年生,教授,博士生导师,研究兴趣包括模型检测、知识推理、非单调推理、多智能体系统、模态逻辑、时态逻辑、概率推理、安全协议验证和逻辑程序设计等. E-mail: isskls@mail.sysu.edu.cn. **岳伟亚**,男,1985 年生,硕士研究生,研究兴趣包括模型检测、时态逻辑、模态逻辑、知识推理、多智能体系统、安全协议验证和逻辑程序设计等. **陈清亮**,男,1980 年生,博士研究生,研究兴趣包括模型检测、知识推理、多智能体系统、模态逻辑、安全协议验证和符号化算法等. **ZHENG Xi-Zhong**,男,1959 年生,研究兴趣为包括数理逻辑和计算复杂性等.

1 引 言

安全协议是一种在不安全的公共网络上为完成某种不可泄漏的信息交换而建立的通信协议. 所谓网络的不安全性,是指黑客的存在,他可以监听、阻止和伪造正在网上传输的信息,他们的标准模型称为 Dolev-Yao 模型[1]. 虽然安全协议使用密码系统来确保其安全性,但是,其中的漏洞很多时候是由于其设计中可能的逻辑错误造成的,而不是由于其加密算法不可靠(如文献[2]就是一个典型的例子). 并且安全协议验证问题是个公认的难题. 因此,如何开发出相应的协议验证工具,使得协议的潜在攻击能自动地被发现,或者某些安全的属性可以被证明在无穷会话下都成立,是个研究的热点.

安全协议的自动验证问题,有两种方法,一种是基于证伪的,另一种是证明的. 证伪的方法先假定一个协议的有穷会话,然后在此会话中搜索潜在的攻击漏洞,这种方法无法证明某些安全性质在协议的无穷会话中成立;证明的方法则弥补了这个缺陷,它着眼于证明某些安全属性在协议的任意无穷会话中都成立. 不过,基于证明的方法,存在以下两个问题:

不完备性. 现存的公理系统不够完备,使得某些属性因为这种不完备性而无法证明.

非自动化和不可扩展性. 在 Dolev-Yao 模型下证明协议的正确性的理论有很多,如 Durgin 的协议组合逻辑(protocol compositional logic)[3]、Paulson 的归纳法[4]、Blanchet 的 ProVerif[5] 等等. 但是,他们要么不能完全自动化实现,要么无法有效扩展和处理大规模的工业级的复杂协议.

本文所提出的方法是基于安全协议的证明的,并且着眼于上述的两个问题的解决. 对于验证公理系统的不完备性,由于对于一般的安全协议,理论上是不存在完备的验证公理系统的,因而协议的验证是个不可判的问题[6]. 因此,我们研究的重心是提供一套公理模式,它足以验证工业级的复杂协议的相关安全属性.

在文献[7～9]中,我们提出了一种新的验证公理模式,并在此公理系统中把单层和多层的认知规范转化成 SAT 问题处理,但他们都没有讨论公理的语义问题,也没有给出详细的实验结果. 本文中我们将给出一个称为"实例化空间(instantiation space)"的安全协议验证逻辑的新的语义模型,该语义模型是建立在一种自然的加密信息交换(cryptographical message exchange)模型上的. 在此语义模型基础上,我们提出了一系列与安全属性相关的验证公理,并可以证明他们在此语义模型下的正确性. 这些公理比现有的一些验证逻辑,如文献[3]中的谓词 Source 以及文献[10]中的 challenge-response axiom schema 和 send-receive axiom schema,能描述更一般的情形(详见备注 3 和 4).

由于本文的逻辑系统在算法上的可行性,对应的工具 SPV(Security Protocol Verifier)已经开发成功,并且可以验证复杂的协议. 由于在此语义基础上的公理集是纯命题逻辑的,因此所需要的验证目标可以很方便地转化成可满足性问题(SAT),从而可以利用工业上快速高效的 SAT 求解器实现. 使用 SPV,我们可以高效验证复杂的包括工业级的安全协议性质,验证的规范包括在 CAPSL(Common Authentication Protocol Specification Language (CAPSL),http://www.csl.sri.com/users/millen/capsl/)中的如"PRECEDES P: $Q|N,M$"的规范以及各种单层和多层的认知规范.

在第 2 节中我们首先给出消息代数的扼要介绍;第 3 节给出具体的加密信息交换(cryptographical message exchange)模型;在第 4 节中给出我们的语义模型"实例化空间(instantiation space)";第 5 节给出此模型上的公理系统;第 6 节给出我们工具的实验结果;第 7,8 节为相关工作的比较和总结.

2 消息代数(Message Algebra)

在协议中,存在三种消息类型:明文消息、组合消息和加密消息. 密钥属于明文消息. 一个消息代数是一个二元组$(\mathcal{M},\mathcal{K})$,其中 $\mathcal{K}\subseteq\mathcal{M}$, 并且满足下列条件:

逆钥:若 $k\in\mathcal{K}$, 则 $k^{-1}\in\mathcal{K}$.

组合:若 $m_1,m_2,\cdots,m_n\in\mathcal{M}$,则$[m_1,m_2,\cdots,m_n]\in\mathcal{M}$.

分解:若$[m_1,m_2,\cdots,m_n]\in\mathcal{M}$,则 $m_1,m_2,\cdots,m_n\in\mathcal{M}$.

加密:若 $k\in\mathcal{K}$ 和 $m\in\mathcal{M}$ 且 m 不具有$\{m'\}_{k^{-1}}$这种形式, 则$\{m\}_k\in\mathcal{K}$.

解密:若 $k\in\mathcal{K}$ 且$\{m\}_k\in\mathcal{K}$,则 $m\in\mathcal{M}$.

我们假定

(1) 若$\{m_1\}_{k_1}=\{m_2\}_{k_2}$ 则 $m_1=m_2$ 且 $k_1=k_2$;

(2) 若$[m_1,m_2,\cdots,m_n]=[m'_1,m'_2,\cdots,m'_{n'}]$则 $n=n'$ 且对于 $1\leq i\leq n$, $m_i=m'_i$.

给定一个消息集合 $\mathcal{M}'$，一个密钥集 $\mathcal{K}'$ 和一个消息 m_0. 我们给出下列符号的定义. 由于篇幅有限，我们略去那些需要在消息结构上进行归纳定义的符号.

(1) $cl(\mathcal{M}')$ 是一个由消息集 $\mathcal{M}'$ 通过加密、解密、分解和组合形成的消息集.

(2) $cl^-(\mathcal{M}')$ 是一个由消息集 $\mathcal{M}'$ 通过解密和分解形成的消息集.

(3) $Sub(m_0,\mathcal{K})$ 是一个由消息 m_0 通过用 $\mathcal{K}'$ 解密和分解形成的子消息集. 进而，我们定义 $Sub(\mathcal{M}',\mathcal{K}')=\bigcup_{m\in\mathcal{M}'}Sub(m,\mathcal{K}')$.

(4) $Fresh(m_0)$ 是一个含有消息 m_0 的消息集. 若 m_0 是一个新鲜的 nonce（第一次使用的），那么 $Fresh(m_0)$ 中的消息都是新鲜的.

3　加密信息交换模型

在本节中，我们定义一个自然的加密信息交换模型（Cryptographical Message Exchange model，CME）为下面一个 11 元组

$$\Sigma=(\mathcal{M},\mathcal{K},e,Ag,Pk,Sk,Stamp,Nonce,Init,Recv,Sent).$$

其中，

$\mathcal{M}$ 是一个消息集；

$\mathcal{K}$ 是密钥集，是 $\mathcal{M}$ 的子集；

e 是一个由黑客控制的通信环境；

Ag 是一组通信的主体. 为方便起见，我们假定 $Ag\subseteq\mathcal{M}$；

Pk 是一些三元组 $\langle a,k,k^{-1}\rangle$，它表示 k 是 a 的公钥而 k^{-1} 是对应的私钥；

Sk 是一些三元组 $\langle a_1,a_2,k\rangle$，它表示 k 是 a_1 和 a_2 的共享密钥. $Stamp$ 是一些标识时间的时戳集；

$Nonce$ 是一个把主体 $a\in Ag$ 映射到 a 产生的 nonces 或 challenge numbers 集合的函数；

$Init$ 是一个把主体 $a\in Ag\cup\{e\}$ 映射到 a 最初产生的明文集的函数 .

$Recv$ 是一个函数，它把主体 $a\in Ag\cup\{e\}$ 和每个自然数（表示时间点）t 映射到一个 a 到时间 t 为止收到的所有消息集.

$Sent$ 是一个函数，它把主体 $a\in Ag\cup\{e\}$ 和每个自然数（表示时间点）t 映射到一个 a 到时间 t 为止发出的所有消息集.

对于每个 $a\in Ag$，定义 $\mathcal{K}_a=Init(a)\cap\mathcal{K}$. 很明显，$\mathcal{K}_a$ 是 a 最初拥有的密钥集. 为了表示 e 最初拥有的密钥集，我们定义

$$\mathcal{K}_e=\mathcal{K}-(PrK\cup ShK).$$

其中，

$$PrK=\{k^{-1}\mid\langle a,k,k^{-1}\rangle\in Pk,\text{其中 } a\in Ag\};$$

$$ShK=\{k\mid\langle a_1,a_2,k\rangle\in Sk,\text{其中 } a_1,a_2\in Ag\}.$$

这里，我们假定环境 e 可以拥有除了 Ag 中的主体的私钥和共享密钥外的所有密钥.

我们再定义 $Poss$ 是一个函数，对于每个 $a\in Ag\cup\{e\}$ 和每个自然数（时间点）t，$Poss(a,t)=cl(Init(a)\cup Recv(a,t))$. 下面我们给出关于 CME 模型的相关假设：

(1) 对于每个 $a\in Ag$ 和 $m\in M$，我们有

(a) 若 $m\in Sent(a,t)$，则 $m\in Recv(e,t+1)$；

(b) 若 $m\in Recv(a,t)$，则 $m\in Sent(e,t-1)$.

此假设表明环境可以收到每个由主体发出的消息，而主体收到的每个消息都由环境发出 .

(2) 若 $m\in Recv(e,t)$，则对于某个 $a\in Ag$，$m\in Sent(a,t-1)$.

此假设表明环境可以包括除了 Ag 外的任意主体.

(3) 对于每个 $a\in Ag\cup\{e\}$，$Sent(a,t)\subseteq Poss(a,t-1)$.

(4) 对于每个 $a\in Ag\cup\{e\}$ 和 $t'<t$，$Sent(a,t')\subseteq Sent(a,t)$ 且 $Recv(a,t')\subseteq Recv(a,t)$.

(5) (a) 设 $a\in Ag$，$k\in\mathcal{K}$ 和 $m\in\mathcal{M}$，若 $\langle a,k,k^{-1}\rangle\in Pk$，则对于除了 a 外的所有 $a'\in Ag\cup\{e\}$，$k^{-1}\notin Init(a')$.

(b) 设 $a_1,a_2\in Ag$，$k\in\mathcal{K}$ 和 $m\in\mathcal{M}$，若 $\langle a_1,a_2,k\rangle\in Sk$，那么对于除了 a_1 和 a_2 外的所有 $a'\in Ag\cup\{e\}$，$k\notin Init(a')$.

(c) 设 $a\in Ag\cup\{e\}$ 和 $m\in Nonce(a)$，那么对于除了 a 外的所有 $a'\in Ag\cup\{e\}$ 我们可以得到 $m\in Init(a)$ 但 $m\notin Init(a')$.

上述假设说明环境或主体最初不能拥有他不该拥有的密钥，或者别人产生的 challenge number.

4　协议的实例化空间

4.1　协议和局部协议

为了描述一个协议，我们首先定义 5 个符号集：TA，TK，TN，TS 和 TG. 他们分别为主体符号集、密钥符号集、nonce 符号集、时戳符号集和一般明文符号集. 这些符号集也可称为原子消息符号集或明文符号集. 一个消息符号是指一个原子消息符

号；或者形如$[M_1,M_2,\cdots,M_n]$的表达式，其中每个$M_i(0<i\leq n)$是一个消息符号；或者形如$\{M'\}_K$的表达式，其中M'是消息符号而K是密钥符号.

我们现在定义一个安全协议$\mathcal{P}$为一个六元组$(AG,PK,SK,NONCE,INIT,BODY)$，其中，

AG是一个主体符号集，每个主体符号表示他们在协议中扮演的角色，如发起者(Initiator)、响应者(Responder)或认证服务器(Server)；

PK是由三元组$\langle P,K,K^{-1}\rangle$组成的集合，该三元组表示$K$是角色$P$的公钥和$K^{-1}$是对应的私钥；

SK是由三元组$\langle P,Q,K\rangle$组成的集合，该三元组表示K是P和Q的共享密钥；

$NONCE$是一个把每个角色$P\in AG$映射到P产生的nonce符号集的函数；

$INIT$是一个把每个角色$P\in AG$映射到P最初产生的原子消息符号集的函数；

$BODY$是协议的实体，由形如$\langle P,Q,M\rangle$的有限序列组成，其中P和Q是AG中的角色，M是由$\bigcup_{P\in AG}INIT(P)$中的原子命题消息以及和PK,SK合成的消息.

在下文中，P,Q,R代表主体符号，K代表密钥符号，N代表nonce符号，M代表消息符号. 注意P不是一个主体，而是TA中的一个符号. 作为对比，我们用ρ,τ表示具体的主体.

我们使用$MSG_{\mathcal{P}}$表示协议$\mathcal{P}$中的所有消息符号，定义为满足下列要求的最小集合：(1)$\bigcup_{P\in AG}INIT(R)\subseteq MSG_{\mathcal{P}}$；(2)若对于某个$P,R\in AG$，$\langle P,R,M\rangle$在$BODY$中，那么$M\in MSG_{\mathcal{P}}$；(3)若一个加密消息$\{M\}_K\in MSG_{\mathcal{P}}$，那么$M$和$K$都在$MSG_{\mathcal{P}}$中；(4)若一个组合消息$[M_1,M_2,\cdots,M_n]$在$MSG_{\mathcal{P}}$中，那么对于所有$i=1,2,\cdots,n,M_i\in MSG_{\mathcal{P}}$.

我们定义角色Q的局部协议$\mathcal{P}(Q)$为它在协议$\mathcal{P}$的$BODY$中所定义的收发消息的局部历史. 也就是一系列事件$\{E_j\}_{j<l_Q}$，其中l_Q是Q的局部历史的长度，每个E_j要么是$send(M)$或者是$recv(M)$，而且$M\in MSG_{\mathcal{P}}$是在$BODY$中出现的$\langle Q,P,M\rangle$或$\langle R,Q,M\rangle$.

4.2 相关符号

我们首先引入一些相关的辅助符号. 给定一个协议$\mathcal{P}$和一个角色Q，对每个$l\leq l_Q$，我们考虑$\{E_j\}_{j<l_Q}$中的初始部分$\{E_j\}_{j<l}$. 我们把根据协议$\mathcal{P}$，角色Q在l步前收到的所有消息记为$Send(Q,l)$. 同样的方法，我们引入$Recv(Q,l)$，$Poss(Q,l)$，$Said(Q,l)$和$Sees(Q,l)$的定义，详细的定义如下：

$Send(Q,l)$为所有满足$send(M)=t_j$且$j<l$的M的集合. 很明显，若$l=0$，则$Send(Q,l)$是个空集.

$Recv(Q,l)$为所有满足$recv(M)=t_j$且$j<l$的M的集合. 很明显$Recv(Q,0)$是个空集.

$Poss(Q,l)$为满足下列条件的最小集合$\Delta\subseteq MSG_{\mathcal{P}}$：(1)$INIT(Q)\cup Recv(Q,l)\subseteq\Delta$；(2)若$[M_1,M_2,\cdots,M_n]\in\Delta$，则对于$i=1,2,\cdots,n,M_i\in\Delta$；(3)若$\{M\}_K\in\Delta$且$K^{-1}\in\Delta$，则$M\in\Delta$.

$Said(Q,l)$为满足下列条件的最小集合Δ：(1)$Send(Q,l)\subseteq\Delta$；(2)对于$i=1,2,\cdots,n$，若$[M_1,M_2,\cdots,M_n]\in\Delta$，则$M_i\in\Delta$；(3)若$\{M\}_K\in\Delta,K\in Poss(Q,l)$且$M\in Poss(Q,l)$，那么$M\in\Delta$.

$Sees(Q,l)$为满足下列条件的最小集合Δ：(1)$Recv(Q,l)\subseteq\Delta$；(2)对于$i=1,2,\cdots,n$，若$[M_1,M_2,\cdots,M_n]\in\Delta$，那么$M_i\in\Delta$；(3)若$\{M\}_K\in\Delta$且$K^{-1}\in Poss(Q,l)$，则$M\in\Delta$.

$Recg(Q,l)$为使得满足$\{M,K\}\subseteq Poss(Q,l)$或$\{\{M\}_K,K^{-1}\}\subseteq Poss(Q,l)$的加密消息$\{M\}_K\in MSG_{\mathcal{P}}$的集合. 显然，若$\{M\}_K\in Recg(Q,l)$，那么角色$Q$可以识别出这是一个用$K$加密的消息$\{M\}_K$.

最后，对于$M\in MSG_{\mathcal{P}}$，我们用$Fresh(M)$表示含有M的新鲜消息组成的集合.

4.3 实例化函数和空间

4.3.1 实例化函数

对于一个协议$\mathcal{P}$和一个CME模型

$$\Sigma=(\mathcal{M},\mathcal{K},e,Ag,Pk,Sk,Stamp,Nonce,Init,Recv,Sent).$$

给定一个主体$\rho\in Ag$，设c是一个nonce或者是一个时戳，我们假定c最多在一个$\mathcal{P}$的会话中使用. 对于诚实的主体来说，这个假设是合理的. 基于此，我们把一个nonce或者一个时戳称为标识消息. 我们用一个二元组(ρ,c)来表示主体用ρ标识消息c的一个局部运行会话. 为了方便起见，我们用变量μ和υ来表示主体和标识消息组成的二元组，c来表示标识消息.

对于每个局部运行$\mu=(\rho,c)$，我们可以定义一个消息赋值函数$f:MSG_{\mathcal{P}}\rightarrow\mathcal{M}$，使得对每个$M\in MSG_{\mathcal{P}}$，$f(M)$是一个在$\mathcal{P}$中的局部运行$\mu$中扮演协议所定义的$M$的具体的消息. 下面的定义给出了这个函数的一些具体的性质.

定义1. 设$\rho\in Ag$，c是个标识消息，P是个角色且$l\leq l_P$. 我们称一个消息赋值函数$f:MSG_{\mathcal{P}}\rightarrow\mathcal{M}$

是一个(ρ,c,P,l)-函数如果它满足

(1) $f(M)$的消息类型由 M 决定. 也就是说，若 A 是个主体名，那么 $f(M)\in Ag$；对于每个时戳符号 $T\in MSG_{\mathcal{P}}$，$f(T)\in Stamp$.

(2) (a) $f(P)=\rho$.

(b) 对于每个 $N\in NONCE(P)$，$f(N)\in Nonce(\rho)$.

(c) 对于每个 $M\in INIT(P)$，$f(M)\in Init(\rho)$.

(d) 对于每个 $l'\leq l$ 和 $M\in Poss(P,l')-Recg(P,l')$，存在一个 t 使得 $f(M)\in Poss(\rho,t)-Recg(\rho,t)$.

(3) 对于 nonce 和时戳符号 $M\in Poss(P,l)$，$f(M)=c$.

(4) (a) $f([M_1,M_2,\cdots,M_n])=[f(M_1),f(M_2),\cdots,f(M_n)]$.

(b) $f(K^{-1})$是 $f(K)$的逆钥.

(c) $f(\{M\}_K)=f(M)_{f(K)}$，若$\{M\}_K\notin Poss(P,l)$或$\{M\}_K\in Recg(P,l)$.

(5) (a) 存在一时间序列点 $t_0<\cdots<t_{l-1}$ 使得对于每个 $j<l$，若对某个 $M\in MSG_{\mathcal{P}}$，$E_j=send(M)$，则 $f(M)\in Sent(\rho,t_j)$，但 $f(M)\notin Sent(\rho,t_j-1)$；而且若对某个 $M\in MSG_{\mathcal{P}}$，$E_j=recv(M)$，则 $f(M)\in Recv(\rho,t_j)$，但 $f(M)\notin Recv(\rho,t_j-1)$.

(b) 假设 $m\in Fresh(f(N))$，其中 N 是 nonce 或时戳. 如果 $m\in Sent(\rho,t)$，那么存在 $E_j=send(M)$ 使得 $m=f(M)$；更进一步，若 $m\in Recv(\rho,t)$，那么在局部协议 $\mathcal{P}(P)$的初始部分$\{E_j\}_{j<l}$存在 $E_j=recv(M)$ 使得 $m=f(M)$.

对于某个角色 P 且 $l\leq l_P$，一个实例化函数就是一个(ρ,c,P,l)-函数.

对于每个局部运行 $\mu=(\rho,c)$，如果 ρ 扮演 P 的角色且已正好完成局部动作 $\mathcal{P}(P)$中的 l 步，那么 μ 中的消息赋值函数或者就是 μ 本身，可以定义为一个(ρ,c,P,l)-函数.

注意若 f 是(ρ,c,P,l)-函数，且对于协议中的某个 nonce 或时戳符号 N，$f(N)=c'$，那么 f 也是一个(ρ,c',P,l)-函数.

4.3.2　实例化空间

定义 2.　一个实例化空间是一个三元组$(\Sigma,\mathcal{F},\mathcal{S})$，其中

(1) Σ 是一个 CME 模型.

(2) $\mathcal{F}$ 是一个实例化函数集合且对于一个给定的(ρ,c)，最多存在一个(ρ,c,P,l)-函数.

(3) $\mathcal{S}$ 是一个关于主体 Ag 行为的假设集；他们是下列形式之一：$Supp_1^{\rho,c}$，$Supp_2^{\rho,P}$ 和 $Supp_3^{\rho,P}$.

(a) 若 $Supp_1^{\rho,c}\in\mathcal{S}$，其中 c 是一个标识消息，那么对于某个 t，如果 $c\in Said(\rho,t)$或者 $c\in Sees(\rho,t)$，就意味着对于某个角色 P 和数字 $l\leq l_P$，存在一个 $\mathcal{F}$ 中的(ρ,c,P,l)-函数.

(b) 若 $Supp_2^{\rho,P}\in\mathcal{S}$，那么对于任意标识消息 c 和时间点 t，如果 $c\in Said(\rho,t)$或者 $c\in Sees(\rho,t)$，就意味着对于某个数字 $l\leq l_P$，存在一个 $\mathcal{F}$ 中的(ρ,c,P,l)-函数.

(c) 若 $Supp_3^{\rho,P}\in\mathcal{S}$，如果在协议 $\mathcal{P}$ 中，存在一个 ρ 的作为角色 P 的未完成的局部运行，那么主体 ρ 会通知别的所有通信主体. 也就是说，在协议 $\mathcal{P}$ 中所有的通信主体都会知道，是否有某个作为角色 P 主体 ρ 有一个未完成的局部运行.

根据定义 2 中的(2)，我们可以使用 $f^{\rho,c}$ 来表示(ρ,c,P,l)-函数(如果存在的话)，而且对于所有 $M\in MSG_{\mathcal{P}}$，我们用 $\overline{M}^{\rho,c}$ 表示 $f^{\rho,c}(M)$.

从直觉上说，$Supp_1^{\rho,c}$ 表示如果主体 ρ 看到或者说了标识消息 c，那么他就用标识消息 c 来运行协议 $\mathcal{P}$ 的会话. $Supp_2^{\rho,Q}$ 表示主体 ρ 只运行协议 $\mathcal{P}$ 并且只扮演 Q 的角色. 比如说，在某些电子商务协议中，我们可以假定一个银行的固有的通信链路只扮演银行的角色.

注 1.　我们在定义 1 和 2 中对实例化函数和空间的假设是很合理的，并且比现有的安全协议分析方法的假设更一般化，如文献[3]中的诚实规则(honesty rule)、文献[11]中的诚信假设(faithfulness assumption)以及文献[12]中的异常暂停假设(fail-stop). 诚实规则(honesty rule)只是假设每个主体都正确按照协议定义的执行步执行. 诚信假设(faithfulness assumption)只是假定主体只有在确认收到的消息是与协议中定义的相符并且是新鲜(fresh)的情况下，才继续执行协议. 异常暂停假设(fail-stop)是假设主体在执行协议的过程中若发现任何异常行为或攻击，他会停止执行协议.

注 2.　在我们所描述的协议情景中，协议 $\mathcal{P}$ 的具体会话是可以并发并且是数目不限的，我们还允许主体执行其他协议；但是，根据定义 1，一个主体在自己的某个运行中所用的新鲜的消息(the fresh messages)不能在别的协议的运行中再重复使用.

5 协议的验证逻辑

本节中我们将在实例化空间的模型上，建立起我们的逻辑理论 $\Theta_{\mathcal{P}}$. 我们首先给出原子表达式集 $V_{\mathcal{P}}$ 的定义.

5.1 原子表达式

5.1.1 非正式的描述

设 X 和 Y 是形如 $\overline{M}^{\rho,c}$ 的变量名或消息名的集合. 逻辑理论 $\Theta_{\mathcal{P}}$ 中的原子表达式包括：

$X=Y$：X 等于 Y. 特别需要说明的是，$\overline{P}^{\upsilon}=\rho$ 表示 ρ 在 $\mathcal{P}$ 的局部运行 υ 中扮演 P 的角色. 我们用 $role(\rho,c,P)$ 来表示 $\overline{P}^{\rho,c}=\rho$，意思是 ρ 在 $\mathcal{P}$ 的局部运行 υ 中扮演 P 的角色并且是使用标识消息 c.

$reach(\rho,c,P,l)$：ρ 在 $\mathcal{P}$ 的局部运行 (ρ,c) 中扮演 P 的角色并且到达了 $(l-1)$ 步. 特别需要说明的是，$reach(\rho,c,P,0)$ 就等价于 $role(\rho,c,P)$. 而且若 l_P 是角色 P 在协议 $\mathcal{P}$ 中的总的运行步，则 $reach(\rho,c,P,l_P)$ 表示 ρ 在局部运行 (ρ,c) 完成了角色 P 的角色. 为了方便起见，我们使用 $rch(\rho,c,P,l)$ 表示 $reach(\rho,c,P,l)\wedge\neg reach(\rho,c,P,l+1)$.

$norm(\rho,c,P)$：若 ρ 在协议 $\mathcal{P}$ 的局部运行 (ρ,c) 中扮演 P 的角色，那么他会一直执行完毕. 明显的，$norm(\rho,c,P)$ 逻辑等价于 $role(\rho,c,P)\Rightarrow reach(\rho,c,P,l_P)$；而且在 $Supp_3^{\rho,P}\in\mathcal{S}$ 的情况下，对所有的主体都是可观察的.

$poss(\rho,X)$：ρ 拥有消息 X.

$said(\rho,X)$：ρ 通过发送含有 X 的消息发送了 X.

$sees(\rho,X)$：ρ 能从收到的消息中得到 X.

$fresh(\mu,X)$：X 在当前协议会话之前，没有被发送过. 对于 $\mu=(\rho,c)$，我们也使用 $fresh(\rho,c,X)$ 表示 $fresh(\mu,X)$. 注意 $fresh(\tau,c,X)$ 表示 τ 通过检测 nonce 的值可以确定消息 X 是否属于局部会话 μ.

5.1.2 表达式的形式化语义

现在我们定义表达式集 $V_{\mathcal{P}}$ 在实例化空间语义模型 $\Omega=(\Sigma,\mathcal{F},\mathcal{S})$ 下的形式语义. 我们使用 $\Omega\models\psi$ 表示 ψ 在模型 Ω 中成立.

$\Omega\models X=Y$ 当且仅当 X 和 Y 在 $\mathcal{M}$ 中是相等的消息.

$\Omega\models reach(\rho,c,P,l)$ 当且仅当在 $\mathcal{F}$ 中存在一个 (ρ,c,P,l)-函数.

$\Omega\models norm(\rho,c,P)$ 当且仅当在 $\mathcal{F}$ 中不存在 $l<l_P$ 的 (ρ,c,P,l)-函数.

$\Omega\models poss(\rho,X)$ 当且仅当存在某个时间点 t，$X\in Poss(\rho,t)$.

$\Omega\models sees(\rho,X)$ 当且仅当存在某个时间点 t，$X\in Sees(\rho,t)$.

$\Omega\models said(\rho,X)$ 当且仅当存在某个时间点 t，$X\in Said(\rho,t)$.

$\Omega\models fresh(\rho,c,X)$ 当且仅当 $X\in Fresh(f(N))$，其中 f 是一个 $\mathcal{F}$ 中对某个角色 P 且在 $l<l_P$ 时的 (ρ,c,P,l)-函数，N 是协议定义的一个 nonce 名或时戳名.

5.2 核心定理

本节中我们只解释说明最核心的验证定理，其他定理请见文献[7].

我们用 $\models_{\mathcal{P}}\psi$ 表示 $(\Sigma,\mathcal{F},\mathcal{S})\models\psi$，意思是对所有的 CME 模型 Σ 和实例化空间 $\mathcal{F}$，ψ 都成立.

定理 1（角色假设）. 若 $Supp_1^{\tau,c}\in\mathcal{S}$，那么 $\models_{\mathcal{P}}(said(\tau,c)\vee sees(\tau,c))\Rightarrow\bigvee_{Q\in AG}role(\tau,c,Q)$；若 $Supp_2^{\tau,Q}\in\mathcal{S}$，那么 $\models_{\mathcal{P}}(said(\tau,c)\vee sees(\tau,c))\Rightarrow role(\tau,c,Q)$.

注意若 $Supp_1^{\tau,c}$ 和 $Supp_2^{\tau,Q}$ 在实例化空间中都不成立，我们就需要加一个形如 $RA_{\mathcal{P}}^{\tau,c,Q}\Rightarrow role(\tau,c,Q)$ 的角色假设，其中 $RA_{\mathcal{P}}^{\tau,c,Q}$ 表示 τ 在协议 $\mathcal{P}$ 中的局部运行 (τ,c) 里想要扮演 Q 的角色. 但是 $RA_{\mathcal{P}}^{\tau,c,Q}$ 往往不能通过协议自动得到；我们需要通过分析协议中该主体的局部历史（local history），像 strand spaces[13] 那样得到.

定理 2（$(n,1)$-Secrecy 定理）. 给定 P_i,ρ_i,c_i 和 $l_i(i=0,\cdots,n)$. 设 $M\in Recv(P_0,l_{P_0})$ 且 l_0 是最小的 l 使得 $M\in Recv(P_0,l)$. 我们有下列定理：

$$\models_{\mathcal{P}}reach(\rho_0,c_0,P_0,l_0)\wedge\bigwedge_{0<i\le n}rch(\rho_i,c_i,P_i,l_i)\Rightarrow GS(P_0,\rho_0,c_0,P_1,\rho_1,c_1,l_1\cdots,P_n,\rho_n,c_n,l_n,M).$$

其中 $GS(P_0,\rho_0,c_0,P_1,\rho_1,c_1,l_1\cdots,P_n,\rho_n,c_n,l_n,M)$（简写成 $GS(M)$）是一个公式，其直觉意义是某个不在 $\rho_i(i=1,2,\cdots,n)$ 中的主体拥有消息 $\overline{M}^{\rho_0,c_0}$.

$GS(M)$ 的精确定义和该定理的证明，可以通过对消息 M 实施结构归纳法得到，详细的内容限于本文篇幅略去. 这条定理是文献[7]中“知识获得约束（Knowledge Gain Constraints）”的进一步推广. 当 $\rho_0=\rho_1$，$c_0=c_1$ 和 $P_0=P_1$ 时，该定理称为 $(n,0)$-Secrecy 定理.

上述定理在验证中是非常重要的，因为它精确地刻画了当一个主体在收到和看到一个来源产生于 n 个主体中的群体的消息时，他可以推出一些什么信息.

注 3. 上述定理的一个特殊的情形是(1,0)-Secrecy 定理，相当于我们前任工作[7]的 Knowledge Gain Constraints. 也可以对应于文献[10]中的 challenge-response axiom schema 和文献[3]中的谓词 Source. 但是，我们的定理是完全根据消息的结构给出的，可以很直接地转化成算法实现，计算当一个主体收到一个消息时，他可以推出什么结论. 相比之下，文献[10]中的 challenge-response axiom schema 没有给出具体的公理的计算算法. 文献[3]中的谓词 Source 则要求 challenge number 在主体收到之前，没有被发送过两次. (1,0)-Secrecy 定理则没有这种要求. 假设一个情景，ρ 产生了一个 nonce c，并分别用 τ_1 和 τ_2 的公钥 k_1 和 k_2 进行加密，把产生的消息 $\{c\}_{k_1}$ 和 $\{c\}_{k_2}$ 分别发送出去. 利用我们的定理，当 ρ 收到 nonce c 时，他可以推出 $said(\tau_1,c)\vee said(\tau_2,c)$.

注 4. (1,1)-Secrecy 定理还优于文献[10]中的challenge-response axiom schema，因为 challenge-response axiom schema 的条件是一个主体说过一个 challenge number 并且他还要收到这个 challenge number，而(1,1)-Secrecy 定理只需要某个主体说过一个 challenge number 而别的主体(不一定原主体)收到这个 challenge number. $(n,1)$-Secrecy 定理甚至比文献[10]中的 send-receive axiom schema 还要更一般化因为后者只考虑两个主体的情况.

6 实验结果

根据上述的理论，我们开发了相应的协议验证工具 SPV. 表 1、表 2 给出了我们的一些实验结果. 实验是在一台 PC(AMD Opteron 242，2GB DDR memory，Redhat Linux 7.0 with gcc 3.2.2)进行的.

表 1

协议	产生变量数	产生定理数	验证时间(s)
Needham-Schroeder	3400	6190	56.407
Rev Needham-Schroeder	3400	6220	55.530
Den-Sacco shared key	1971	2851	27.061
Woo and Lam Pi	1612	2867	17.736
Kao Chow Authen v1	7734	14197	436.493
Kerberos V5	8369	14845	398.932
The SET purchase	59749	83747	28594.5

6.1 工具的效率

在验证过程中，对一个协议，产生变量和定理需要大部分时间，验证规范只需要很小一部分时间，通常只有几秒.

6.2 验证的规范

SPV 能验证许多规范. 如 CAPSL 中的“PRECEDES P: $Q|N,M$”，其中 N 是 nonce 名. 该规范表示在 Q 的最后状态，若有一个对应于 Q 的“Q”-主体拥有 P，N 和 M，那么也有一个对应于 P 的“P”-主体拥有同样值的 Q，N 和 M. 在我们的逻辑中可表示为

$$(reach(\tau,c_2,Q,l_Q)\wedge(\overline{P}^{\tau,c_2}=\rho)\wedge(\overline{N}^{\tau,c_2}=c_1))\Rightarrow(role(\rho,c_1,P)\wedge(\overline{Q}^{\rho,c_1}=\tau)\wedge(\overline{N}^{\rho,c_1}=c_1)\wedge(\overline{M}^{\rho,c_1}=\overline{M}^{\tau,c_2})).$$

更重要的是，SPV 可以验证单层和多层的认知规范，如单层规范“Bob 知道 Alice 说过某些消息”和多层规范“Alice 知道 Bob 知道 Alice 说过某些信息”. 具体这些认知规范如何转化成 SAT 问题，限于篇幅不作详细介绍，详见文献[7].

设 $S1_a$ 代表形如 $O_a\Rightarrow K_a(said(b,X))$ 的单层认知规范；$S2_a$ 代表形如 $O_a\Rightarrow K_aK_b(said(a,X))$ 的双层规范，其中 a 和 b 表示协议定义的 *initiator* 和 *responder*；O_a 表示 a 的可观察变量[7]；X 表示 a 和 b 共享的秘密如 nonce. 同样的 $S1_b$ 和 $S2_b$ 定义与此类似.

表 2

协议	$S1_a$	$S1_b$	$S2_a$	$S2_b$
Needham-Schroeder	Yes	Yes	Yes	No
Rev Needham-Schroeder	Yes	Yes	Yes	Yes
Den-Sacco shared key	No	No	No	No
Woo and Lam Pi	No	Yes	No	No
Kao Chow Authen v1	Yes	No	No	No
Kerberos V5	Yes	Yes	Yes	Yes
The SET purchase	Yes	Yes	Yes	Yes

7 相关工作

Cervesato，Meadows 和 Pavlovic[10]曾给出一个认证协议逻辑，以秘密性和诚实规则(honesty rules)为前提，对协议动作的偏序(partial orders of protocol actions)进行推导. 他们的理论和我们的理论都在主体的观察(*observation*)上进行推理. 但是，我们的工作是直接在协议的 Clark-Jacob 定义上进行操作的，并且实例化函数很好地表征了一个协议的语义. 我们的语义模型还克服了 Clark-Jacob 定义的模糊性，因为给定一个 CME 模型可以有不同的实例化空间(instantiation spaces). 而且，他们的逻辑没有给出针对任意协议如何生成对应所有验证

公理的具体算法，而我们给出了相应的算法并开发了相应的工具(详见备注 4).

在文献[3]中，Durgin，Mitchell 和 Pavlovic 提出了一种组合验证逻辑(compositional logic)，用来分析推导一族协议. 他们的理论和我们的理论都是在不考虑黑客的行为的前提下，对协议的所有可能的运行进行推理，从而依此去证明协议在无穷会话中的正确性. 但是，他们的理论与本文的理论有如下区别：

我们的$(n,1)$-Secrecy 定理比他们的相应的公理要更一般化(详见备注 3).

我们的验证定理是用命题逻辑表示的，它们的验证可以很直接转化成 SAT 问题，而且相应的工具已开发成功，但是他们的公理用模态动作逻辑，目前还没有见到相应的验证工具和结果.

组合验证逻辑(compositional logic)目前只能处理公钥加密的情形，而我们的逻辑则可以灵活地处理公钥密码、私钥密码、共享密码、动态生成的密码和 Hash 函数等加密系统.

最后组合验证逻辑禁止会话重叠(“crosstalk”)，而我们只是禁止那些含有只能一次性使用的新鲜消息的会话重叠(详见备注 2).

另外还有给安全协议赋予代数语义模型的工作，如文献[14,15]. 代数语义可以在一定程度上弥补逻辑语义在表达能力上的不足. 但代数语义没有逻辑语义那么直接，并且在算法上逻辑语义更易实现.

8 结 论

本文给出了协议验证逻辑的一个很自然的语义模型：实例化空间(instantiation space). 该逻辑能灵活地处理复杂的任意格式的各种加密的消息类型. 而且基于该理论的验证工具 SPV 已经开发成功. 使用 SPV，我们可以高效验证复杂的包括工业级的安全协议性质，验证的规范包括在 CAPSL 中的如“PRECEDES $P:Q|N,M$”的规范以及各种单层和多层的认知规范.

本文的未来工作，着眼于处理更多的密码原语，如 Diffie-Hellman 函数以及各种新型的安全协议，如电子投票协议和匿名通信协议等等.

参 考 文 献

1 Dolev D., Yao A. C.. On the security of public-key protocols. IEEE Transactions on Information Theory, 1983, 29(8): 198～208

2 Lowe G.. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In: Proceedings of the Tools and Algorithms for the Construction and Analysis of Systems, TACAS'96, Passau, Germany, 1996, Lecture Notes in Computer Science: 1055, 1996, Springer Verlag, 147～166

3 Durgin N. A., Mitchell J. C., Pavlovic D.. A compositional logic for proving security properties of protocols. Journal of Computer Security, 2003, 11(4): 677～722

4 Paulson L. C.. The inductive approach to verifying cryptographic protocols. Journal of Computer Security, 1998, 6(1/2): 85～128

5 Blanchet B.. An efficient cryptographic protocol verifier based on prolog rules. In: Proceedings of the 14th IEEE Computer Security Foundations Workshop CSFW-14 2001, Cape Breton, Nova Scotia, Canada, IEEE Computer Society, 2001, 82～96

6 Cervesato I., Durgin N. A., Lincoln P., Mitchell J. C., Scedrov A.. A meta-notation for protocol analysis. In: Proceedings of the 12th IEEE Computer Security Foundations Workshop(CSFW 1999), Mordano, Italy, 1999, 55～69

7 Su Kaile, Lv Guanfeng, Chen Qingliang. Knowledge structure approach to verification of authentication protocols. Science in China(Series F: Information Science), 2005, 48(4): 513～532

8 Su Kaile, Lv Guanfeng, Zhang Yan. Reasoning about knowledge by variable forgetting. In: Proceedings of the 9th International Conference on Principles of Knowledge Representation and Reasoning KR-2004, Whistler, Canada, 2004, 576～586

9 Ron van der Meyden, Su Kaile. Symbolic model checking the knowledge of the dining cryptographers. In: Proceedings of the 17th IEEE Computer Security Foundations Workshop(CSFW-17), Pacific Grove, CA, USA, IEEE Computer Society, 2004, 280～292

10 Cervesato I., Meadows C., Pavlovic D.. An encapsulated authentication logic for reasoning about key distribution protocol. In: Proceedings of the IEEE Eighteenth Computer Security Foundations Workshop—CSFW-18, Aix-en-Provence, France, IEEE Computer Society Press, 2005, 48～61

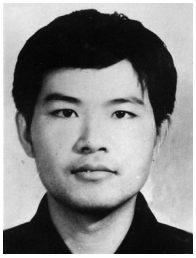
11 Syverson P. F.. Adding time to a logic of authentication. In: Proceedings of the 1st ACM Conference on Computer and Communications Security, Fairfax, Virginia, USA, 1993, 97～101

12 Li Gong, Paul Syverson. Fail-stop protocols: An approach to designing secure protocols. In: Proceedings of the 5th International Working Conference on Dependable Computing for Critical Applications DCCA-5, Urbana-Champaign, IL, 1995, 44～55

13 Ji Qing-Guang, Qing Si-Han, Zhou Yong-Bin, Feng Deng-Guo. Study on strand space model theory. Journal of Computer Science and Technology, 2003, 18(5): 553～570

14 Huai Jin-Peng, Li Xian-Xian. Algebra model and security analysis for cryptographic protocols. Science in China(Series F: Information Sciences), 2004, 47(2): 199～220

15 Xue Rui, Feng Deng-Guo. New semantic model for authentication protocols in ASMs. Journal of Computer Science and Technology, 2004, 19(4): 555～563



SU Kai-Le, born in 1964, professor, Ph. D. supervisor. His research interests include model checking, reasoning about knowledge, non-monotonic reasoning, multi-agent system, modal logic, temporal logic, probability reasoning, verification of security protocol, logic programming.

YUE Wei-Ya, born in 1985, master candidate. His research interests include model checking, temporal logic, modal logic, reasoning about knowledge, multi-agent system, verification of security protocol, logic programming.

CHEN Qing-Liang, born in 1980, Ph. D. candidate. His research interests include model checking, reasoning about knowledge, multi-agent system, modal logic, verification of security protocol and symbolic algorithms.

ZHENG Xi-Zhong, born in 1959, Privatdocent. His research interests include mathematical logic and its application to theoretical computer science and computational complexity.

Background

Security protocols are communication protocols which are intended to perform some information exchanges in the presence of malicious intruders over a public network, who can listen to the messages being sent and also block or forge messages as the Dolev-Yao model defines. It is widely acknowledged that logical flaws in security protocols are hard to analyze, and it is desirable to develop automatic tools by which attacks on protocols can be discovered or the concerned security properties can be proved.

There are two main approaches to analyzing security protocols: The falsification-oriented ones and the justification-oriented ones. However, in the justification-oriented approach there remain the following two problems.

Incompleteness: The existing axiomatic systems for general security protocols are far from being complete. Due to the incompleteness, an axiomatic system may fail to prove a valid security property for the protocol.

Automation and scaling: There are a number of tools and approaches that are focused on producing proofs of correctness in the Dolev-Yao model. However, they are either not fully automatic or not efficiently executable and flexible at least for large-scale and complex protocols.

This work is along the justification-oriented ones and aims to deal with the two problems above. As for the incompleteness problem, there theoretically exists no complete axiomatic system for general security protocols because the protocol analysis problem is known to be undecidable. Thus, the authors' aim is not to provide a complete axiomatic system; instead, is to looking for a set of evidently valid axiom schemas that is complete enough to enable us to verify the concerned security properties for various protocols including complex industrial ones.

In this paper, the authors present a new model of security protocol logics, which is called Instantiation Space. This model is formally well grounded, based the so-called Cryptographical Message Exchange (CME) model. A set of interesting axioms (security properties) can be proved to be sound with respect to this model. These axioms characterize a lot of interesting methods for reasoning about security protocols, which is beyond the predicate Source and the challenge-response axiom schema even the send-receive axiom schema.

Based on this logic, a new Security Protocol Verifier (SPV) is designed and implemented, particularly focused on automatic verification of complex protocols. The axiom schemas presented are in propositional logic, so that the proposed approach can be automated using SAT solvers. With SPV, some interesting complex properties can be also verified. The verified specifications include goals in CAPSL like "PRECEDES $P:Q|N,M$", as well as multi-level epistemic ones.