基于树语言逼近的安全协议形式化分析

刘 楠 朱文也 祝跃飞 陈 晨
(信息工程大学信息工程学院 郑州 450002)

摘 要 利用形式化方法或工具自动化分析实用安全协议十分必要,定理证明技术因其可解决无限状态系统的验证备受关注,但扩展其验证规模和自动化实现时仍然存在一些局限性。以定理证明和重写逼近理论为基础,以项重写形式化定义协议模型,以树自动机模拟协议攻击者知识集,给出攻击者知识集可达项逼近求解的算法,并根据上述模型讨论秘密性和认证性的验证方法,最后以 Needham-Schroeder 公钥认证协议为例验证模型的有效性,并指出下一步研究方向。

关键词 安全协议,项重写,树自动机,树语言,逼近,秘密性,认证性
中图法分类号 TP309 **文献标识码** A

Formally Analyzing Security Protocol Using Approximation of Tree Language

LIU Nan ZHU Wen-ye ZHU Yue-fei CHEN Chen
(Institute of Information Engineering, Information Engineering University, Zhengzhou 450002, China)

Abstract Formal method and automated tools are both necessary and efficient for analyzing practical security protocols. Theorem proving theory has been studied intensively because it can verify infinite-state system. But there are still limitations when verifying large scale system automatically. A formal model for security protocols was proposed based on theorem proving and rewriting approximation theory. It utilizes term rewriting system and tree automata to model protocol and the knowledge of intruder. An automatic generated approximation algorithm was designed to calculate the fix-point tree automata. In this model, secrecy and authentication were discussed. By a case of study, NSPK was analyzed effectively.

Keywords Security protocol, TRS, Tree automata, Tree language, Approximation, Secrecy, Authentication

1 引言

安全协议是解决网络安全最直接、最有效的手段之一,是构建安全信息系统的一个基本要素。实践表明,复杂的网络环境为安全协议的设计带来巨大的挑战,安全协议的形式化分析已经成为信息安全领域一个极为重要的研究方向。

安全协议的形式化分析研究始于上世纪80年代初,应用需求的不断扩展,促使各种形式化分析方法、理论和应用技术不断创新。目前安全协议的形式化分析主要包括基于知识、信念的模态逻辑推理,基于状态搜索的检测和定理证明等方法和技术。各个领域都已经取得了大量的、很有价值的成果,理论研究正趋于标准化,安全协议的发展呈现出这样的趋势:设计的实用化、分析的理论化、检测和评估的自动化[1]。

与基于状态搜索的模型检测技术相比,定理证明技术由于可以解决无限状态系统的验证问题而受到学者的广泛关注。本文以定理证明和重写逼近理论为基础,基于项重写TRS(Terms Rewriting System)[2]定义协议的形式化模型,用树自动机模拟协议攻击者知识集,提出了攻击者知识集可达项逼近求解的算法,给出了对安全协议的秘密性和认证性的验证方法,并通过相应的实例,验证了模型的有效性。

本文第2节简单介绍了相关的研究现状;第3节给出了协议的形式化模型和基于树自动机的攻击者模型;第4节提出了可自动化的逼近求解算法;第5节提出了秘密性和认证性的分析和验证方法;最后以 Needham-Schroeder 公钥认证协议为例进行分析,给出了该领域下一步研究的问题。

2 相关研究

定理证明技术使用类似于结构化的推导过程来证明具有无限状态的系统。树自动机[3]是定理证明领域的一种有效的模型,能够处理无限的对象,其主要不足在于普遍的机械化证明过程难以完全自动化推演和计算,在扩展系统验证规模时存在一定的局限性。而逼近[4]技术是抽象解释理论中的一种有效的方法,能够为计算机科学中很多不可判定问题或复杂问题的逼近求解提供系统性的构造方法和有效算法,简化证明过程。

Thomas Genet 和 Francis Klay[5]于2000年提出基于重

到稿日期:2009-02-20 返修日期:2009-04-28 本文受国家高技术研究发展计划(863)(2007AA01Z471)资助。
刘 楠(1981—),女,博士研究生,主要研究领域为信息安全、形式化分析,E-mail:liunanwb@163.com;朱文也(1984—),女,硕士研究生,主要研究领域为安全协议形式化分析;祝跃飞(1962—),男,博士,教授,主要研究领域为信息安全;陈 晨(1984—),男,硕士研究生,主要研究领域为安全协议形式化分析。

写和树自动机技术来证明安全协议，并指出该方法结合了定理证明和逼近的优点。以上述理论为基础开发了工具 Timbuk，用于在给定初始树自动机和 TRS 下，计算树自动机的逼近集，但此方法存在一定的局限性，工具的某些操作需要人工辅助，且要求 TRS 是左线性的，而实际上大部分安全协议形式化时不符合左线性条件。

H. Ohsaki 和 T. Takai 于 2004[6]年扩展了 Thomas Genet 和 Francis Klay 的树自动机理论研究，提出了 AC(Associative and Commutative)树自动机，主要是为了解决计算过程中特殊操作符的 AC 代数属性问题，提出 ACTAS 自动机。

Y. Boichut 等人于 2005[7,8]年提出类左线性条件，降低了对 TRS 的限制，开发了面对非专家用户的工具 TA4SP。TA4SP 是 AVISPA[8]工具中以定理证明为理论基础的证明工具，其优点在于能证明无限主体和无限会话下的协议安全性。TA4SP 目前只针对小规模协议进行验证，且尚未实现对多层次认证性、公平性等其他安全属性的证明，处理了个别协议特殊操作符的代数性质。其后续工作着重于探讨求解不动点自动机的存在性问题，成果尚不成熟[14]。

国内对安全协议形式化研究亦非常重视。中科院软件研究所信息安全国家重点实验室召开多次安全协议研讨会，讨论安全协议形式化分析中的问题和发展趋势[1]。中科院冯登国等教授在安全协议形式化分析理论方面做出巨大贡献。北京航空航天大学的怀进鹏教授等引入新的密码协议代数 CPA，设计并研发 SPA[9]；中山大学的苏开乐[11]教授等根据模型检测认知逻辑和时态认知逻辑开发安全协议分析器 SPV；李梦君等学者基于类 π 演算设计研发 SPVT 工具[10]。这一系列工作都大大推进了该领域的发展。目前基于定理证明技术的相应研究成果尚不成熟，有必要促进其相关理论和应用技术的发展。本文在定理证明和重写逼近理论的研究基础上，结合项重写和树自动机理论模拟协议和攻击者，研究其自动化分析方法和验证技术。

3 安全协议形式化建模

形式化模型是安全协议形式化分析的基础和依据。本文以项重写系统 TRS 模拟协议的运行，其语义模型是将主体(包括攻击者)之间通过消息交互触发协议步骤执行的这种动作，看作是状态转移系统中依据转换规则进行状态的变迁。

3.1 消息

消息是协议运行中的基本元素，构成了 TRS 中的项。

定义 1 A 为有限非空原子符号集，X 表示和 A 不相交的可数变量集合。F 是函数符号集合，对于每一个 F 中的符号 f 关联一个自然数 n，这个自然数称作 f 的阶，f 就称为 n 阶函数符号。由阶为 n 的函数符号构成的 F 的子集记作 F_n，则 F 按"阶"可划分为 n 个子集，$F=\bigcup_{i=1}^{n}F_i$。

消息就是由原子集合 A、变量集合 X 及其上的运算 F 构成的项重写系统 R 的基础，记作 $A_F(X)$。$A_F(X)$ 中元素由如下方式生成：

- 若 $x\in V$，则 $x\in A_F(X)$；
- 若 $t\in A$，则 $t\in A_F(X)$；
- 如果 $t_1,\cdots,t_n\in A_F(X)$，f 为 n 阶函数符号 $f\in F_n$，则 $f(t_1,\cdots,t_n)\in A_F(X)$。

除上述方式，$A_F(X)$ 中没有其他元素。$A_F(X)$ 形式化模拟了消息的组成，分为原子消息和复合消息。A 的语义为原子消息集合，是不可再分的最小消息单元，包括主体、密钥、随机数和原始数据(即不含其它函数的需要传输的数据串)，如主体名字($a|b|c|\cdots$)、密钥($pubkey|prvkey|shrkey|\cdots$)、随机数($Nonce|\cdots$)等；$F$ 的语义为密码运算函数符号集，复合消息是由原子消息或复合消息通过密码运算复合而成。将所有密码运算都以函数符号表示，如发送消息操作符 $msg(x,y,m)$、加密操作符 $enc(key,msg)$、散列函数 $hash(msg)$、主体标识符 $agt(\cdot)$等。$A_F(X)$的所有元素构成了安全协议的消息空间，简称消息项。

对任意的项 $s\in A_F(X)$，记 $Pos(s)$为项 s 的位置集合，记 $s|_p$或 $s(p)$为项 s 在位置 p 的子项，$s[t]_p$ 表示用 t 替换 s 在位置 p 的子项后所得的项。项 s 中变量的集合记为 $Var(s)$。

若 $A_F(X)$上的函数 $\sigma:V\to A_F(X)$，其定义域 $dom(\sigma)$为变量集合 V 或 V 的子集，其值域 $ran(\sigma)$为消息集合或其子集，满足 $\sigma(x)\neq x$，则称 σ 为替换。对给定的一组有限数量的消息对$\{(s_1,t_1),\cdots,(s_n,t_n)\}$，替换 σ 称作一个合一当且仅当所有$i\in\{1,\cdots,n\}$都满足 $s_i\sigma=t_i\sigma$。

3.2 转换规则

协议的运行由消息的交互推动，转换规则用来描述协议各方如何交换消息，是依据协议描述形式化协议参与角色在协议运行中的行为规范，其语义是标识协议角色在什么状态下可以接收或者发送消息以及消息应该具有的格式。形式化定义如下。

定义 2 一般地，项重写系统是重写规则 $l\to r$ 的有限集，其中 $l,r\in A_F(X)$，$l\notin X$，且 $Val(l)\supseteq Val(r)$。规则的左边 lhs 和右边 rhs 都是有限的参数化消息项的集合。当前提条件"lhs"和当前状态的某部分匹配时，该部分就能被相应的后续结果"rhs"替换(经过一些处理之后)。对应于协议步骤，左边表示一个当前状态的前提，即初始通信请求或接收到的消息，右边表示如果前提被满足后应该发送的消息。

以 NSPK 协议为例，该协议的初始规则定义如下：

$goal(agt(x),agt(y))\to msg(agt(x),agt(y),enc(pubkey(agt(y)),(N(agt(x)),agt(x))))$

左边是主体 $agt(x)$对主体 $agt(y)$发起通信请求，右侧的 $msg(x,y,c)$表示 x 给 y 发送消息 c，$enc(k,d)$表示用密钥 k 对 d 加密的结果。规则需要考虑无限主体和无限会话，因此在重写工作中，主体被抽象为带变量的项 $agt(x)$和 $agt(y)$。所有的规则构成规则集 R。

定义 3 由有限规则集 R 诱导出的重写关系 $\to_R$ 是指对任意项 $s,t\in A_F(X)$，若存在 R 中一条规则 $l\to r$、位置 $p\in Pos(s)$、替换 σ，使得 $s|_p=l\sigma$，$t=s[r\sigma]|_p$，则 $s\to_R t$。$\to_R$ 的反身传递闭包记为$\to_R^*$。

在规则集 R 的作用下，基础项集 E 的 R-后继为 $R^*(E)=\{t\in T(F)\mid\exists s\in E,s\to_R^* t\}$，称作 E 的可达项集合。

3.3 攻击者模型

攻击者能力定义基于 Dolev-Yao 模型。攻击者完全控制通信媒介，在无穷参与主体和无穷并行会话的前提下，攻击者的知识集经过协议模型 TRS 重写得到的可达项集合显然是无限的。

在传统 DY 模型中，攻击者的知识推演由操作 $DY(.)$记录，它是一个无限的消息集合。$t\in DY(M)$只能说明主体能

够从它的知识 M 中推导出 t。显然这是一种枚举的记录方式。

相对于这种集合和普通的字自动机(word automata)，树自动机描述能力更强，可以刻画这种无限的、复杂的树状消息项结构，其叶子节点都是原子消息项，中间节点是操作符。本文采用自底向上的不确定有限树自动机所接受的语言，即树语言来模拟攻击者的初始知识集。

定义 4 自底向上的不确定有限树自动机 A 是一个四元组 (Q,Q_f,Δ,F)。这里 $Q_f\subseteq Q$ 是终结状态集，Q 是有限状态集，Q 中的元素是特殊的常量符号；F 是有限符号集且 $F\cap Q=\emptyset$，Δ 是形如 $f(p_1,\cdots,p_n)\to q_1$ 的转换集，其中 $f\in F^{(n)}$，$p_1,\cdots,p_n,q_1,\cdots,q_n\in Q$。

由 Δ 诱导出的转移关系记为 $\to_A$，表示：

$t\to_A t'\Leftrightarrow\exists\, l\to r\in\Delta, p\in Pos(t), t|_p=l, t'|_p=r$

若项 $t\to_\Delta^* q$ 且 $q\in Q$，则 t 能够被树自动机 A 所接受。能够被树自动机 A 所识别的树语言即 $L(A)=\{t\in T(F)\mid\exists q\in Q_f, t\to_A^* q\}$。根据树自动机的性质，若存在一个自底向上的树自动机 A 满足 $L(A)=E$，则树语言 E(项集)是正规的。初始树自动机 A_0 接受的语言 $L(A_0)$ 是攻击者的初始知识集，即实例化后的协议初始状态，包括通信请求、初始攻击者知识、攻击者拆分消息能力 3 个部分，都是基础项，即不包含变量的消息项。

4 不动点树自动机 A_k 的求解

根据第 3 节，以重写中的项定义消息，以重写规则描述协议动作，构建了一个基于重写的安全协议模型。诚实主体和攻击者参与协议运行会话实例时，只要满足规则，就可以按照协议描述进行消息交互。交互过程中，攻击者知识集不断扩张，而对任何一个项集 E 在项重写系统 TRS 作用下其可达项集 $R^*(E)$ 一般都是不正规的[12]。而树自动机所能接受的语言集都是正规的，即无法用一个恰当的树自动机使得它接受的语言 $L(A)$ 为攻击者知识可达项集合 $R^*(E)$，因此需要通过重写来逼近攻击者可达项知识集的精确集。

重写逼近主要思想是定义逼近操作，通过生成一系列的树自动机，最终得到不动点树自动机，使得它接受的语言逼近于攻击者知识的精确集。根据逼近操作 γ 的不同，可得到过逼近集(大于精确集)或欠逼近集(小于精确集)[7]。

对应于第 3 节模型，即对协议模型中的每条 TRS 规则和当前初始树语言 $L(A_0)$ 合一求解，根据解得的替换来检测规则左边能否被 A_0 所接受。若满足，检查右边的项替换后能否被当前树自动机所接受，若可以说明自动机不变；若不能接受，就需要定义逼近操作添加转移关系，将右侧项添加进来，得到新自动机 A_1。重复以上步骤，直到协议所有重写规则都被应用，扩张生成一系列树自动机 $A_1,A_2,\cdots$，直至到达一个不动点树自动机 A_k，即攻击者知识扩张集的逼近集。其计算过程形式化定义如下。

输入：树自动机 $A_0=(F,Q,Q_f,\Delta_0)$，项重写系统 R

输出：树自动机 $A_k=(F,Q,Q_f,\Delta_k)$，使得 $L(A_K)\supseteq R^*(L(A_0))$

步骤 1 $k=0, A_i=A_0$；

步骤 2 取项重写系统 R 中的一条规则 $l\to r$，假设 l 中有 n 个变量 $x_1,\cdots,x_n$，其中 x_i 出现 t_i 次 $(0\leqslant i\leqslant n)$，分别对应 l 中的位置 p_{ij} $(0\leqslant j\leqslant t_i)$。定义替换 $\sigma: p_{ij}\mapsto q_{ij}$ (对 A_i 中的项和规则左侧 l 进行合一)使得

$$l\sigma\to_{A_k}^* q \tag{1}$$

$$r\sigma\not\to_{A_i}^* q \tag{2}$$

Case 1：若满足式(1)、式(2)，则执行步骤 3；

Case 2：若只满足式(1)，不满足式(2)，则执行步骤 4；

Case 3：若都不满足，则执行步骤 2。

步骤 3 由于 $r\sigma\not\to_{A_i}^* q$，将 $r\sigma$ 标准化，在当前树自动机 A_i 中增加相应转移使得 $r\sigma\to_{A_{i+1}}^* q_f$，即可得到新自动机 A_{i+1}。对 r 的非变量位置定义逼近操作 γ 完成 $Pos_F(r)\to Q$ 的操作。

Case 1：过逼近操作 γ_1

当 $r(p)$ 为常量时，若 $\min\{q\mid r(p)\to q\in\Delta\}\neq\emptyset$，则 $\gamma_1(p)$ 取原状态值；否则 $\gamma_1(p)=q_{new}$；

当 $r(p)$ 为操作符时，$\gamma_1(p)=\min\{q\mid r(p)(\beta_1,\cdots,\beta_n)\to q\in\Delta$，若 $r(p.i)\in X, \beta_i=\sigma(p.i)$，否则 $\beta_i=\gamma_1(p.i)\}$；

不为空则 $\gamma_1(p)$ 取原状态值；否则 $\gamma_1(p)=q_{new}$。

Case 2：欠逼近操作 γ_2

当 $r(p)$ 为常量时，$\gamma_2(p)=q_{r(p)}$；

当 $r(p)$ 为操作符时，则 $\gamma_2(p)=q_{new}$。

对上述位置所增加的新状态，增加相应的转移集，生成新转移集 Δ_{i+1}，定义如下：

$\{f(q_1,\cdots,q_n)\to q'\mid p\in Pos_F(r), t(p)=f, q'=q$，若 $p=\varepsilon$，否则 $q'=\gamma(l\to r,\sigma,p)(p)$

$q_i=\gamma(l\to r,\sigma,q)(p.i)$ 若 $p.i\notin Pos_\chi(r)$

否则 $q_i=\sigma(\min\{p'\in Pos_\chi(l)\mid l(p')=r(p.i)\})\}$

步骤 4 检查 Δ_{i+1}，若 $\Delta_{i+1}=\Delta_i$，循环停止，输出 $A_k=A_i$；否则 $k=k+1$，执行步骤 2。

由于逼近过程中标记各级子项的状态不确定，最后得到的攻击者知识并不是精确的集合，输出的不动点树自动机 A_k 所接受的语言是精确攻击者知识可达项集合的一个逼近集合，根据逼近操作的不同，可以得到过逼近集和欠逼近集。上述的逼近算法在完成 $r\sigma$ 标准化时，采用了 TA4SP 的思想，对非左线性的 TRS 系统是有效的。即用状态替换项中变量的位置，而不是替换变量本身，很大程度降低了左线性的限制。TRS 的左线性拓展研究不是本文的主要工作，相关研究进展可以参考文献[7]。

5 安全属性的分析和验证

5.1 秘密性

秘密性通常指参与安全协议的通信主体间不能公开的秘密信息的安全性，一般都与具体消息相关。如在密钥分发协议中，会话密钥对攻击者或其他无关主体是不能公开的。对于一个给定的安全协议，其秘密项通常是可确定的。

定义 5 秘密项自动机 A_{sec} 用来表示协议的所有秘密项集合，其接受的语言 $L(A_{sec})=\{c\mid c$ 是秘密项$\}$。

根据攻击者知识集不动点自动机 A_k，秘密性的检测即与 A_{sec} 求交，验证结论与逼近策略的选取有关。若选择过逼近操作，且 $L(A_{sec})\cap L(A_k)=\phi$，则秘密项都不被树自动机 A_k 所接受，协议满足秘密性，否则不能判别；若选择欠逼近操作，且 $L(A_{sec})\cap L(A_k)\neq\phi$，则说明存在秘密项被树自动机 A_k 所接受，协议违背秘密性的目标，否则不能判别。可以看出，这两种情况对属性的验证都是半可判定的。

5.2 认证性

认证性是协议安全属性中的一个重要属性。学术界对认证性的分类很多，如身份认证和数据源认证。这种分类是以认证的目的为出发点，易于理解，但不易于形式化。本文对认

证性的定义根据 Lowe 提出的层次化定义[13]，将认证划分 4 个层次：活跃性、弱一致性、非单射一致性和单射一致性。这种分层的好处在于结构清晰、易于形式化和自动化验证。

这 4 个层次认证强度逐渐加强，活跃性最弱，单射一致性性质最强。本文为了验证认证性，在协议的重写规则中，引入一对认证谓词标识，其形式化定义如下。

定义 6 $initiate(A,B,ds,sid)$表示主体 A 与主体 B 开始一次会话运行，置于主体 A 的最后一条规则前，sid 是会话号，ds 表示认证数据载荷。

定义 7 $finish(B,A,ds,sid)$表示 B 与 A 完成一次会话运行，置于 B 的最后一条规则后，ds 是数据载荷，sid 是会话号，ds 表示认证数据载荷。

添加标识的位置根据主体规则来确定。将 *initiate* 标识置于主体的最后一条规则前，是为了确保数据载荷 ds 已经发出。认证数据载荷的选取对认证性的验证至关重要，不适当的数据载荷对认证性验证没有意义。

由于非单射一致性保证了活跃性和弱一致性，本文只讨论非单射一致性和单射一致性的检测。

- 非单射一致性：若 $initiate(B,A,ds,sid)$存在，对每一个 $initiate(B,A,ds,sid)$，在此之后必存在一个 $finish(A,B,ds,sid)$与之是一致的，除 sid 外。
- 单射一致性：若 $initiate(B,A,ds,sid)$存在，对每个 $initiate(B,A,ds,sid)$，在此之后必存在唯一 $finish(A,B,ds,sid)$，与之是一致的。

当协议运行完毕只存在 *initiate* 标识，而没有 *finish* 标识时，不能判断是否破坏认证性；若只存在 *initiate* 标识，而没有 *initiate* 标识，则也无法保证任何安全目标。在对规则添加认证谓词后，规则被触发的时候认证谓词项也被加入到攻击者树自动机中。

根据第 4 节得到的攻击者知识集不动点自动机 A_k 检测认证性。在欠逼近策略下，若 $L(A_k)$中的认证性标识符满足上述两种定义，则被检测协议满足非单射一致性目标和单射一致性的目标；在过逼近策略下，若 $L(A_k)$中的认证性标识符不满足上述两种定义，即 *initiate* 和 *finish* 在认证载荷上存在不一致，或认证双方在协议运行上不唯一对应，则违背了单射一致性和非单射一致性的目标。除这两种情况以外，其他情况下对协议的认证目标都是不可判定的。

6 实例研究

本节以典型协议 NSPK 为例说明基于上述模型的安全协议秘密性和认证性分析。NSPK 协议最早由 Needham 和 Schroeder 于 1978 年提出，其 A&B 自然语言描述如下：

(1) $A\rightarrow B:\{NA,A\}_{K_B}$

(2) $B\rightarrow A:\{NA,NB\}_{K_A}$

(3) $A\rightarrow B:\{NB\}_{K_B}$

其中，A,B 是协议的参与双方 Alice 和 Bob 的简称。NA 和 NB 分别是由 A 和 B 生成的随机数，K_A 和 K_B 则分别是 A,B 公钥。

6.1 转换规则

以带有认证标识的 TRS 重写规则表示协议通信步骤，在每一条规则后面加入本条规则的认证标识符。

Rule1：

$goal(agt(x),agt(y),ses)\rightarrow mesg(agt(x),agt(y),enc(pubkey(agt(y)),[\![cons(N(agt(x),agt(x)))]\!])\cup initiate(agt(x),agt(y),ds(agt(x),agt(y)),ses)$

Rule2：

$msg(agt(x),agt(y),enc(pubkey(agt(y)),[\![cons(N(agt(x)),agt(x))]\!]))\rightarrow msg(agt(y),agt(x),enc(pubkey(agt(x)),[\![cons(N(agt(x)),N(agt(y)))]\!]))\cup initiate(agt(y),agt(x),N(agt(y)))$

Rule3：

$msg(agt(y),agt(x),enc(pubkey(agt(x)),[\![cons(N(agt(x)),N(agt(y))]\!]))\rightarrow msg(agt(x),agt(y),enc(pubkey(agt(y)),[\![N(agt(y))]\!]))\cup finish(agt(x),agt(y),N(agt(y)))$

Rule4：

$msg(agt(x),agt(y),enc(pubkey(agt(y)),[\![N(agt(y))]\!]))\rightarrow end\cup finish(agt(y),agt(x),N(agt(x)))$

最后一条规则中引入 *end* 标识符，表示该主体在当前会话中进入结束状态。

6.2 攻击者初始自动机和秘密项自动机

攻击者初始树自动机 $A_0(Q_0,Q_f,\Delta_0,F)$的转移关系定义如下，其中 q_{net} 是终结状态。

$0\rightarrow q_{int}$	$s(q_{int})\rightarrow q_{int}$	
$1\rightarrow q_s$	$s(q_s)\rightarrow q_s$	
$agt(q_{int})\rightarrow q_{net}$	$agt(q_s)\rightarrow q_{net}$	
$A\rightarrow q_A$	$agt(q_A)\rightarrow q_{agtA}$	
$B\rightarrow q_B$	$agt(q_B)\rightarrow q_{agtB}$	
$agt(q_A)\rightarrow q_{net}$	$agt(q_B)\rightarrow q_{net}$	$agt(q_int)\rightarrow q_{agtI}$
$goal()q_{agtA},q_{agtI},q_s)\rightarrow q_{net}$		$goal()q_{agtI},q_{agtA},q_s)\rightarrow q_{net}$
$goal()q_{agtA},q_{agtB},q_s)\rightarrow q_{net}$		$goal()q_{agtB},q_{agtO},q_s)\rightarrow q_{net}$
$goal()q_{agtB},q_{agtA},q_s)\rightarrow q_{net}$		$goal()q_{agtI},q_{agtB},q_s)\rightarrow q_{net}$
$goal()q_{agtA},q_{agtA},q_s)\rightarrow q_{net}$		$goal()q_{agtI},q_{agtI},q_s)\rightarrow q_{net}$
$goal()q_{agtB},q_{agtB},q_s)\rightarrow q_{net}$		
$cns(q_{net},q_{net})\rightarrow q_{net}$		
$null\rightarrow q_{net}$		
$enc(q_{net},q_{net})\rightarrow q_{net}$		
$mesg(q_{net},q_{net},q_{net})\rightarrow q_{net}$		

0,1 是常量，用于标识会话号，操作符 $s()$记录自然数即会话号的增加，A,B 表示协议参与角色，除 A,B 外的所有主体，$agt(i)i\in R$ 都可能是攻击者，$goal(x,y,ses)$表示 x 对 y 发起会话号为 ses 的会话，代表协议的启动请求。此外，攻击者根据其可操作的函数符号具有组合生成消息的能力，该能力以重写规则形式表示如下：

$cons(x,y)\cup z\rightarrow LHS\cup x$

$cons(x,y)\cup z\rightarrow LHS\cup y$

$msg(x,y,z)\cup u\rightarrow LHS\cup z$

$enc(pubkey(agt(0)),z)\cup u\rightarrow LHS\cup z$

$enc(pubkey(agt(s(x))),z)\cup u\rightarrow LHS\cup z$

$enc(shrkey(agt(0)),z)\cup u\rightarrow LHS\cup z$

$enc(shrkey(agt(s(x))),z)\cup u\rightarrow LHS\cup z$

根据该协议的描述和定义，NA 和 NB 是需要保密的秘密项，则定义秘密项树自动机 $A_{sec}(Q,Q_f,\Delta,F)$，其中 $F=\{agt(),N()\}$，$Q=\{q_1,q_2,q_3,q_4,q_5,q_6,q_7\}$，$Q_f=\{q_7\}$，转移集 Δ 为

$F=\{agt(),N(),A,B\}$

$Q=\{q_1,q_2,q_3,q_4,q_5,q_6,q_7\},Q_f=\{q_7\}$

$\Delta=\{A\rightarrow q_1,B\rightarrow q_2,agt(q_1)\rightarrow q_3,agt(q_2)\rightarrow q_4,N(q_3)\rightarrow$

$q_7, N(q_4) \rightarrow q_7\}$

根据定义，秘密项树自动机 A_{sec} 接受的语言集 $L(A_{sec}) = \{N(agt(A)), N(agt(B))\}$。

6.3 逼近求解和验证

对 NSPK 的每一条规则，执行第 4 节逼近算法，完成一系列的重写和知识扩张。例如首先对规则 1 和初始实例化树自动机 A_0 合一，得到一个从 $Pos_X(l)$ 到 Q 的替换 σ，使得 $\sigma(1.1) = q_1, \sigma(2.1) = q_2, \sigma(3) = q_s$，则根据 A_0 中的转移关系 $agt(q_A) \rightarrow q_{agtA}$ 和 $agt(q_B) \rightarrow q_{agtB}$，可知 $goal(q_{agtA}, q_{agtB}, q_s) \rightarrow q_{net}$，即满足可知 $l\sigma \rightarrow_{A_0}^{*} q_{net}$，但 $r\sigma \not\rightarrow_{A_0}^{*} q_{net}$。

对规则右边的项 $r\sigma$ 进行标准化，需要加入一个正规的转移集，对 msg 操作符的非变量替换的位置，定义过逼近函数 γ_1：

$\gamma_1(1) = q_4, \gamma_1(2) = q_5, \gamma_1(3) = q_9, \gamma_1(3.1) = q_8,$

$\gamma_1(3.2) = q_7, \gamma_1(3.1.1) = q_5, \gamma_1(3.2.1) = q_6, \cdots$

加入转移 Δ 如下：

$N(q_4) \rightarrow q_6, cons(q_6, q_4) \rightarrow q_7, pubkey(q_5) \rightarrow q_8,$

$enc(q_7, q_8) \rightarrow q_9, msg(q_4, q_5, q_9) \rightarrow q_{13}$

加入的标识 $initiate$ 加入正规的转移：$ds(q_{agtA}, q_{agtB}) \rightarrow q_{net}$

由于 $\Delta_{i+1} \neq \Delta_i$，树自动机 A_0 扩张为 A_1 且 $r\sigma \rightarrow_{A_1}^{*} q_{net}$。

根据该协议已知的攻击路径，依次应用其所有规则，经过手动推导，该条路径最后得到的树自动机包含如下标识：

$initiate(a, i, Na, s1), initiate(b, a, Nb, s2),$

$finish(a, b, Nb, s1), finish(b, a, Na, s2)$。

根据认证性的形式化描述可知，a 对 b 的存活性无法保证，b 对 a 也只能保证存活性。协议违背了认证性的安全目标，同时违背了秘密性的安全目标。

结束语 定理证明是一种“证明”的形式化分析方法，可以处理无限状态系统的验证问题，是分析安全协议的有效手段，但由于其推理过程复杂，不易于实现自动化推演。本文的主要贡献在于给出语义清晰的协议形式化模型和可自动化的不动点树自动机求解算法，提出秘密性和认证性的形式化描述和验证方法，最后以 NSPK 协议为例进行了验证，以便进一步开展自动化研究工作。在下一步研究中，将以此模型为基础着重研究其自动化分析过程中尚未处理的问题，如如何统一处理特殊运算符的代数属性，如何与模型检测方法结合进一步增强模型对大型实用安全协议（如 IPsec 等）的适用性，以及如何兼容其他类型安全属性等，都有待于进一步深入研究。

参 考 文 献

[1] 冯登国. 第三届安全协议研讨会[C]//北京：信息安全国家重点实验室，2007

[2] Baader F, Nipkow T. Term Rewriting and All that[M]. Cambridge: Cambridge University Press, 1999

[3] Comon H, Dauchet M, Gilleron R, et al. Tree automata and techniques and applications[EB/OL]. http://13ux02.univ-lille3.fr/tata/, 2002

[4] Cousot P, Cousot R. Abstract Interpretation and Application to Logic Programs[J]. The Journal of Logic Programming, 1992, 13: 103-179

[5] Gent T, Klay F. Rewriting for Cryptographic Protocol Verification[J]. Lecture Notes in Computer Science, 2000(1831/2000): 271-290

[6] Ohsaki H, Takai T. Actas: a system design for associative and commutative tree automata theory[A]//Proceedings of the 5th International Workspace on Rule-based Programming: RULE'2004[C]. Aachen: Elsevier, 2005: 97-111

[7] Boichut Y, Heam P C, Kouchnarenko L. Automatic Verification of Security Protocols Using Approximation[R]. INRIA CASSIS Project, RR-5727. 2005

[8] Automated Validation of Internet Security Protocols and Applications (AVISPA)[J/OL]. http://www.avispa-project.org/

[9] 李建新，李先贤，卓怀亮，等. SPA：新的高效安全协议分析系统[J]. 计算机学报，2005，28(3)：309-318

[10] 李梦君，李舟军，陈火旺. SPVT：一个有效的安全协议验证工具[J]. 软件学报，2006，17(4)：898-906

[11] 苏开乐，岳伟亚，陈清亮，等. 实例化空间：一种新的安全协议验证逻辑语义模型[J]. 计算机学报，2006，29(9)：1657-1665

[12] Gilleron R, Tison S. Regular tree languages and rewrite systems [J]. Fundamenta Informaticae, 1995, 24: 157-175

[13] Lowe G. A Hierarchy of Authentication Specifications[A]// Proceedings of the 10th IEEE Workshop on Computer Security Foundations[C]. Washington: IEEE Computer Society Press, 1997: 31-43

[14] Boichut Y. A theoretical limit for safety verification techniques with regular fix-point computations[J]. Information Processing Letters, 2008, 108(1): 1-2

(上接第 121 页)

[2] Guttman J D, Thayer F J. Authentication tests[C]//Proceedings of IEEE Symposium on Security and Privacy. 2000: 96-109

[3] Harn L, Xu Y. Design of generalised ElGamal type digital signature schemes based on discrete logarithm[J]. Electronics Letters, 1994, 30(24): 20-

[4] Thayer F F J, Herzog J C, Guttman J D. Mixed strand spaces [C]//Proceedings of the 12th IEEE Computer Security Foundations Workshop. 1999: 72-82

[5] Guttman J D. Security protocol design via authentication tests [C]//Proceedings of 15th IEEE Computer Security Foundations Workshop. 2002: 92-103

[6] Vadhan S P. An unconditional study of computational zero knowledge[C]//Proceedings of 45th Annual IEEE Symposium on Foundations of Computer Science. 2004: 176-185