ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.18, No.7, July 2007, pp.1746−1755
DOI: 10.1360/jos181746

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一种改进的密码协议形式化模型*

张 畅+, 王亚弟, 韩继红, 郭渊博

(解放军信息工程大学 电子技术学院,河南 郑州 450004)

An Improved Formal Model of Cryptographic Protocol

ZHANG Chang+, WANG Ya-Di, HAN Ji-Hong, GUO Yuan-Bo

(Institute of Electronic Technology, PLA Information Engineering University, Zhengzhou 450004, China)

+ Corresponding author: Phn: +86-013526816225, E-mail: zcmail123@yahoo.com.cn

Zhang C, Wang YD, Han JH, Guo YB. An improved formal model of cryptographic protocol. *Journal of Software*, 2007,18(7):1746−1755. http://www.jos.org.cn/1000-9825/18/1746.htm

Abstract: MSR (multiset rewriting) model is a technique of protocol modeling based on multiset rewriting. According to the results of the current study, this model is not perfect. Since the intruder model of MSR model is not suitable for the verification of the protocol, it was improved. Using this model, the secrecy and authentication of the cryptographic protocols are described. The practice proves the improvement of the former model.
Key words: cryptographic protocol; secrecy; authentication; formal verification

摘 要: 多重集重写MSR(multiset rewriting)模型是一种基于多重集重写的协议形式化建模方法.从目前的研究成果来看,该模型并不完善.针对其攻击者模型验证协议存在的不足,对MSR模型进行了改进,并给出了基于MSR模型的秘密性和认证性描述.实践表明,对模型的改进进一步完善了原模型.
关键词: 密码协议;秘密性;认证性;形式化验证
中图法分类号: TP309 **文献标识码**: A

对密码协议进行形式化分析时,首先面临的问题是:如何形式化描述协议,即如何建立协议的形式模型.目前已经提出来的比较著名的协议模型有Dolev-Yao模型、Millen-Paulson模型、CSP(communicating sequential processes)通信顺序进程模型、Strand Space模型等[1].为了建立一种更有效的密码协议形式化分析模型,Cervesato等人提出了一种基于线性逻辑的多重集重写形式系统(multiset rewriting system)[2].在此工作的基础上,进一步发展形成了MSR(multiset rewriting)模型[3],它是Dolev-Yao模型在多重集重写系统中的解释.与以往的协议模型相比,该模型可以简洁、精确地描述协议和攻击者,但该模型在密码协议安全特性验证方面有待进一步的完善.本文对该模型进行了改进,使其可以更有效地验证密码协议.

1 MSR 模型

在 MSR 模型中,协议是一个靠交换消息来通信的角色集合,每个角色有一个所有者(一般实体、服务器、攻

* Supported by the National Natural Science Foundation of China under Grant Nos.60503012, 90104005, 90204012 (国家自然科学基金)

Received 2005-12-20; Accepted 2006-04-27

击者),在接收到期望的消息后,实体作出执行规则的响应.每个角色是参数化的多重集重写规则集,规则模拟消息的接收和传输.传输中的消息、角色的状态、角色已知的信息和其他数据构成了协议执行的状态(设置的主体),规则完成这些状态(设置的主体)之间的转换.模型图示如图 1 所示.

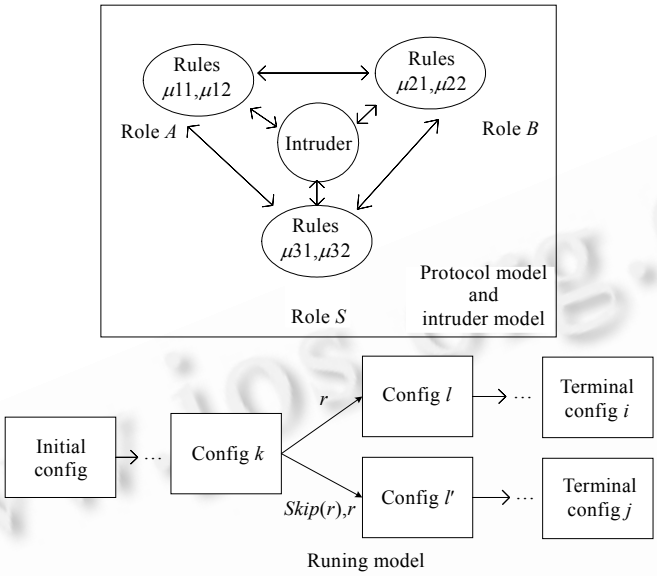

Fig.1 MSR model

图 1 MSR 模型图示

1.1 基本概念

消息

MSR模型中,数据的基本成分是消息,原子消息的语法为$a::=A|k|n|m$.其中:A表示实体;k表示密钥;n表示随机数;m表示未加工数据.消息的语法为$t::=a|x|t_1t_2|\{t\}_k|\{\{t\}\}_k$,其中:$a$表示原子消息;$x$表示变量;$t_1t_2$表示消息串联;$\{t\}_k$表示对称密钥加密;$\{\{t\}\}_k$表示非对称密钥加密.在某些情况下,要用到既可以是变量也可以是原子消息常量的对象.这种项为基本项,用字母e表示.

类型

类型是一种对消息(项)进行分类的句法结构,类型给每个项一个意义.如一个对象的类型是密钥,那么,该对象就不能作为随机数使用.类型的语法为$\tau::=principal|nonce|shKAB|pubKA|privKk|atm|msg$,其中:*principal* 表示实体类型;*nonce* 表示随机数类型;*shKAB* 表示 A 和 B 共享密钥类型;*pubKA* 表示实体 A 的公钥类型;*privKk* 表示公钥 k 的逆的类型;*atm* 表示原子消息类型;*msg* 表示消息类型.

状态

状态是设置的主体,是 MSR 分析协议的场景假设的基础.为了形式化描述状态,先要定义消息元组和消息谓词.消息元组的语法为$\vec{t}::=\bullet|t,\vec{t}$,其中:•表示空元组;$t,\vec{t}$ 表示元组扩展.消息元组的类型是参数化的类型有序序列:$\vec{\tau}::=\bullet|\tau^{(x)}\times\vec{\tau}$,当变量 x 在 $\vec{\tau}$ 中不出现时,可省略标志(x).

消息谓词是状态的基本成分,是以 0 项和多项作为自变量的原子一阶逻辑公式.该谓词有 3 种:

1) 网络谓词 $N(_)$表示在分布式网络中的内容:对每一个在传输中的消息 t,状态包含一个形式为 $N(t)$的成分;
2) 角色状态谓词$L(_,\ldots,_)$记录角色在执行过程中所知的某些参数,它的第 1 个参数的值是其所属角色所有者的实体名,其中:L 是一个变量;λ_l 是其实例化的形式,其下标 l 是唯一标识;

3) 记忆谓词$M_A(_,\ldots,_)$表示实体A用其私有的记忆谓词存储数据,这里,A是一个变量.

状态 S 是一个有限的基础状态谓词集,其语法结构为 $S ::= \bullet \mid S, N(t) \mid S, 1_l(\vec{t}) \mid S, M_A(\vec{t})$.

规则

规则的主要形式为 $lhs \rightarrow rhs$.协议执行可表示成设置之间的转换,规则是设置转换的依据.简单地说,当规则的先行 lhs 与当前状态 S 的部分相匹配时,即当 lhs 是 S 的子集时,S 的下一个状态可为$(S \backslash lhs) \cup rhs$,即在多重集合 S 中用规则的后继 rhs 替换掉 lhs 成分.其语法为 $r ::= lhs \rightarrow rhs \mid \forall x{:}\tau.r$.规则中的所有量词∀的作用范围在角色范围内,即规则中的变量在角色中是封闭的.

规则的左半部分叫做规则的先行,是一个参数化的消息谓词序列: $lhs ::= \bullet \mid lhs, N(t) \mid lhs, L(\vec{e}) \mid lhs, M_A(\vec{t})$.与状态不同的是,规则中角色状态是参数化的,而状态中角色状态谓词必须是实例化的.

规则的右半部分叫做规则的后继,由消息谓词序列和冠以有限新鲜数变量声明的消息谓词序列组成.存在量词∃描述新鲜数变量声明:$rhs ::= lhs \mid \exists x{:}\tau.rhs$.存在量词∃的作用范围在整个协议理论内,新鲜数变量与该规则所属角色的角色状态谓词中的变量一致.

角色和协议理论

角色是一个包括所有者的规则集,首先给出规则集的语法: $\rho ::= \bullet \mid \exists L : \vec{\tau}.\rho \mid r, \rho$.其中, $\exists L : \vec{\tau}.\rho$ 表示角色状态谓词声明.角色联系着角色所有者A和一个规则集ρ.MSR模型中,协议是由不同的角色组成的,协议理论P的语法为$P ::= \bullet \mid P, \rho^{\forall A} \mid P, \rho^{\mathrm{A}}$.其中:$P, \rho^{\forall A}$表示一般角色扩展;$P, \rho^{\mathrm{A}}$表示锚定角色扩展.

动态角色

在协议执行过程中,角色可以任意实例化,可以任意停止.为了记录这些在协议执行中的角色的特征,这些特征包括该角色完成了哪些操作、其所有者是谁等.MSR模型用动态角色集来记录这些角色特征.动态角色集的语法为$R ::= \bullet \mid R, \rho^{\mathrm{A}}$.这里,$\rho^{\mathrm{A}}$不一定是锚定角色,动态角色$\rho^{\mathrm{A}}$表示角色实体参数$A$已经实例化了,A是该角色的所有者.

1.2 执行模型

MSR 模型用从设置 C 转移到另一个设置 C'的动作来模拟协议的运行.设置 C 包括状态 S、动态角色集合 R 和笔记Σ,形式为 $[S]^R_\Sigma$ .R 记录了可用来继续执行的角色、角色可以停下来的点以及角色的实例化情况.设置中不包含自变量.判断 $P > C \rightarrow C'$ 表示一步(应用一个规则)将设置 C 转换成设置 C'.

执行协议首先要激活协议理论中的角色,将它们添加到动态角色集中.下面的两个推理规则详细说明怎样扩展当前角色集合 R.

$$\frac{}{(P, \rho^{\mathrm{A}}) > [S]^R_\Sigma \rightarrow [S]^{R, \rho^{\mathrm{A}}}_\Sigma}\, ex\_arole \quad , \quad \frac{\Sigma \mid -\mathrm{A} : principal}{(P, \rho^{\forall A}) > [S]^R_\Sigma \rightarrow [S]^{R, ([\mathrm{A}/A]\rho)^{\mathrm{A}}}_\Sigma}\, ex\_grole \ .$$

规则 ex_arole 表示锚定角色被简单地复制到动态角色集,因为其语法满足动态角色的要求.规则 ex_grole 表示给一般的角色分配实体名 A 使该角色具有实际意义.

一旦角色被激活,那么,其状态谓词参数必须在其规则执行前实例化.推理规则 ex_rsp 中,λ_l 是一个在当前设置中没有出现的新符号(它不能在Σ中).推理规则 ex_all 实例化规则中所有的变量.

$$\frac{}{P > [S]^{R, (\exists L:\vec{\tau}.\rho)^{\mathrm{A}}}_\Sigma \rightarrow [S]^{R, ([\lambda_l / L]\rho)^{\mathrm{A}}}_{(\Sigma, \lambda_l : \vec{\tau})}}\, ex\_rsp \quad , \quad \frac{\Sigma \perp t : \tau}{P > [S]^{R, ((\forall x:\tau.r), \rho)^{\mathrm{A}}}_\Sigma \rightarrow [S]^{R, (([t/x]r), \rho)^{\mathrm{A}}}_\Sigma}\, ex\_all \ .$$

下一步考虑应用动态角色集R中的完全实例化的规则($lhs \rightarrow rhs$)改变协议的设置.其中,先行lhs必须是基本的,并且符合类型检测[2].该规则匹配当前状态中和lhs一样的成分,并用后继的子状态lhs'对其进行替换.

$$\frac{(rhs)_\Sigma >> (lhs')_{\Sigma'}}{P > [S, lhs]^{R, ((lhs \rightarrow rhs), \rho)^{\mathrm{A}}}_\Sigma \rightarrow [S, lhs']^{R, (\rho)^{\mathrm{A}}}_{\Sigma'}}\, ex\_core \ .$$

子状态lhs'是将规则的后继rhs实例化后得出的,这由实例判断$(rhs)_\Sigma >> (lhs')_\Sigma$来描述.

$$\frac{}{(lhs)_{\Sigma} >> (lhs)_{\Sigma}} ex\_seq \ , \frac{([\partial / x]rhs)_{(\Sigma,\partial:\tau)} >> (lhs)_{\Sigma'}}{(\exists x:\tau.rhs)_{\Sigma} >> (lhs)_{\Sigma'}} ex\_nnc \ .$$

协议执行过程中可以出现许多分支,这可以模拟协议在执行过程中有多个规则可以执行的情况.比如,在遇到某个可执行的规则时产生分支:应用该规则得到下一个状态和跳过该规则应用其他的规则.跳过某个规则的推理规则如下:

$$\frac{}{P > [S]_{\Sigma}^{R_i(r,\rho)^{\mathrm{A}}} \to [S]_{\Sigma}^{R_i(\rho)^{\mathrm{A}}}} ex\_skp \ , \frac{}{P > [S]_{\Sigma}^{R_i(\cdot)^{\mathrm{A}}} \to [S]_{\Sigma}^{R}} ex\_dot \ .$$

推理规则 ex_dot 表示抛弃空的动态角色.

判断 $P > C \to^{*} C'$ 表示从设置 C 到 C'的多步转换.该判断由下面的推理规则完成:

$$\frac{}{P > C \to^{*} C} ex\_it0 \ , \frac{P > C \to C' \quad P > C' \to^{*} C''}{P > C \to^{*} C''} ex\_itn \ .$$

1.3 攻击者模型

MSR模型是Dolev-Yao模型的实例.MSR模型将Dolev-Yao攻击者描述成多个单规则角色,这些规则构成标准的Dolev-Yao攻击者理论P_{DY}.由于受篇幅的限制,这里不再详细介绍该模型.

2 MSR 模型的改进与安全特性的描述

2.1 MSR模型的改进

我们采用 CSP 模型的验证思想,即对协议的形式化验证可转化成对运行轨迹的分析.MSR 模型将协议的执行过程描述成从一个设置转换到另一个设置的过程,所以,验证密码协议就是分析由这些设置形成的踪迹是否违反协议的安全需求.

定义 1. 协议踪迹 $C_0 \xrightarrow{\alpha_1,\varsigma_1} C_1 \xrightarrow{\alpha_2,\varsigma_2} \ldots \xrightarrow{\alpha_n,\varsigma_n} C_{n+1}$ 是一个设置序列,C_i和C_{i+1}是连续的设置,即在设置C_i和C_{i+1}之间不应用推理规则 $P > C_i \to^{*} C_{i+1}$.其中:α表示角色的实例化情况;ς表示应用规则的情况.

如果协议执行到某设置时无法继续运行了,即对于当前设置,所有的规则(包括攻击者规则)都无法满足推理规则 ex_core,则该踪迹终止.

MSR 模型对 Dolev-Yao 攻击者模型的描述是很全面的,攻击者可以任意分解合成消息.然而,这种“任意”操作增加了推理的难度,因为“任意”操作包含了许多无用的操作:分解消息,然后又将其合成,形成的无限循环.这在协议验证过程中表现为踪迹无法终止,而且会出现许多对协议分析无用的设置,给分析带来了很大的工作量.我们在不削弱攻击者能力的前提下对攻击者模型做了一定的改进,以去掉这种“任意性”,使得协议验证更有效.

我们将攻击者操作分成两个阶段:消息分解和消息合成.攻击者截获消息后,尽可能地将消息分解,然后将其记忆谓词中的消息合成新消息.我们用 3 个记忆谓词 $D\_:msg$,$A\_:atm$,$C\_:msg$ 取代攻击者模型中的记忆谓词 $M\_:msg$.谓词 $D_$记忆分解和待分解的消息;谓词 $A_$记忆分解过程中得到的原子消息;谓词 $C_$记忆合成和待合成的消息.

攻击者截获某消息后,将其添加到消息谓词 $D_$中开始分解消息.当消息被分解成最小的单位后,就可以转移到谓词 $C_$中.在转移过程中,需要一些过渡规则来限制转移的消息是最小单位.为了使分解合成操作不构成循环,我们首先应用所有可应用的分解规则,然后应用合成规则合成消息.其中,只有合成规则中谓词 $C_$中的消息才可以发送到网络上.

我们将原来的攻击者规则分成 4 种:分解规则、合成规则、过渡规则、获得基本信息的规则:

(1) 分解规则

INT':$(\forall t{:}msg.N(t) \to D(t))^{\mathrm{I}}$

DCM':$(\forall t_1,t_2{:}msg.D(t_1t_2) \to D(t_1)D(t_2))^{\mathrm{I}}$

$$SDC'\colon \begin{pmatrix} \forall A,B:principal. \\ \forall k:shKAB. \quad D(\{t\}_k)\,\mathrm{A}(k)\rightarrow D(t) \\ \forall t:msg. \end{pmatrix}^{\mathrm{I}}$$

$$PDC'\colon \begin{pmatrix} \forall A:principal. \\ \forall k:pubKA. \quad D(\{\{t\}\}_k)\,\mathrm{A}(k')\rightarrow D(t) \\ \forall t:msg. \\ \forall k':privKk. \end{pmatrix}^{\mathrm{I}}$$

(2) 合成规则

TRN':$(\forall t{:}msg.C(t)\rightarrow N(t))^{\mathrm{I}}$

CMP_1' :$(\forall t_1,t_2{:}msg.C(t_1)C(t_2)\rightarrow C(t_1t_2))^{\mathrm{I}}$

CMP_2' :$(\forall t_1,t_2{:}msg.C(t_1)C(t_2)\rightarrow C(t_2t_1))^{\mathrm{I}}$

$$SEC'\colon \begin{pmatrix} \forall A,B:principal. \\ \forall k:shKAB. \quad C(t)\,\mathrm{A}(k)\rightarrow C(\{t\}_k) \\ \forall t:msg. \end{pmatrix}^{\mathrm{I}}$$

$$PEC'\colon \begin{pmatrix} \forall A:principal. \\ \forall k:pubKA. \quad C(t)\mathrm{A}(t)\rightarrow C(\{\{t\}\}_k) \\ \forall t:msg. \end{pmatrix}^{\mathrm{I}}$$

(3) 过渡规则

ATM_1: $(\forall a{:}atm.D(a)\rightarrow A(a))^{\mathrm{I}}$

ATM_2: $(\forall a{:}atm.\ A(a)\rightarrow C(a)A(a))^{\mathrm{I}}$

ATM_3: $(\forall t{:}temp.\ D(t)\rightarrow C(t)D(t))^{\mathrm{I}}$

消息在分解过程中得到原子消息,原子消息不能继续分解,规则ATM_1表示将原子消息添加到谓词$A_$中,ATM_2表示原子消息可以用来合成消息.并不是所有的消息都可以分解成原子消息,因为加密消息在不知道解密密钥的情况下不能被分解成原子消息.对于攻击者来说,这些没有解密的消息也可以用来合成新的消息.MSR模型中加密消息的类型为msg,规则ATM_1,ATM_2不能将非原子消息类型的消息从记忆谓词$D_$中转移到记忆谓词$C_$中,所以,我们在原来的模型中添加类型$temp$,并用规则ATM_3实现加密消息从记忆谓词$D_$转移到记忆谓词$C_$中.

(4) 攻击者获得基本信息的规则

IPR':$(\forall A{:}princ.\ \bullet\rightarrow A(A))^{\mathrm{I}}$

$$IS1'\colon \begin{pmatrix} \forall A:princ. \\ \forall k:shKIA. \quad \bullet\rightarrow \mathrm{A}(k) \end{pmatrix}^{\mathrm{I}}$$

$$IS2'\colon \begin{pmatrix} \forall A:princ. \\ \forall k:shKAI. \quad \bullet\rightarrow \mathrm{A}(k) \end{pmatrix}^{\mathrm{I}}$$

$$IPB'\colon \begin{pmatrix} \forall A:princ. \\ \forall k:pubKA. \quad \bullet\rightarrow \mathrm{A}(k) \end{pmatrix}^{\mathrm{I}}$$

$$IPV'\colon \begin{pmatrix} \forall k:pubKI. \\ \forall k':privKk. \quad \bullet\rightarrow \mathrm{A}(k') \end{pmatrix}^{\mathrm{I}}$$

GNC':$(\bullet\rightarrow\exists n{:}nonce.A(n))^{\mathrm{I}}$

GMS':$(\bullet\rightarrow\exists m{:}msg.A(m))^{\mathrm{I}}$

下面我们讨论对攻击者模型的改进是否合理.首先我们来看改进后攻击者模型的攻击能力与原攻击者模型相比有没有削弱.前面我们提到用3个记忆谓词:$C\_$,$A\_$,$D_$置换原攻击者模型中的记忆谓词$M_$,这将攻击者的操作分成两个阶段:消息分解和消息合成.原攻击者模型对消息的操作不分这两个阶段,也就是说在原攻击者模

型中,这两个操作是不分先后的.假设原攻击者模型可以将一个消息t转换成另一个消息t',在这一过程中使用了规则$r_1,r_2,\ldots,r_n$,这些规则可以在我们改进的模型中找到对应的规则,只不过在改进的攻击者模型中,分解规则和合成规则之间需要用规则ATM_1,ATM_2,ATM_3衔接.MSR模型中,攻击者的攻击能力表现在对消息的任意处理上.因为原攻击者模型可以生成的消息也可以用改进的攻击者模型生成,所以,我们对攻击者模型的改进没有削弱攻击者能力.

其次,我们来看改进的攻击者模型与原攻击者模型相比有什么优点.改进的攻击者模型把攻击者的操作分成分解和合成两个动作,对于一个消息,先对其进行彻底的分解,将消息分解成原子消息或者加密消息,再用这些原子消息或加密消息通过合成规则合成其他消息.而原模型则不是这样的,攻击者不必完全分解消息就可以对消息进行合成,随后可能又将其分解,这产生了无限循环.

2.2 安全特性

MSR 模型的执行过程是一个状态推理过程,用该模型分析密码协议就是推导出协议运行的所有可达状态.如果在这些状态中没有对协议的攻击,则协议满足安全特性,所以,该模型适合从反面来描述密码协议安全特性,即定义协议攻击对应的状态,这些状态是不希望出现的.我们给出对秘密性的攻击和对认证性的攻击对应的状态或者状态序列的描述.

2.2.1 秘密性

在与协议参与者所有可能的交互过程中,如果攻击者不能得到协议的秘密信息,那么协议满足秘密性.一般来说,密码协议分析通过寻找攻击者得到秘密信息的踪迹来验证秘密性:如果在所有协议执行踪迹中存在一条踪迹包含攻击者得到的秘密信息,则协议不满足秘密性,该踪迹对应一条对协议的攻击.

考虑到MSR模型用设置描述协议执行特点,我们用投影算法$Prj_a(_)$来描述设置C中消息的特征.该算法的定义如下:

定义 2. 投影算法$Prj_a(_)$满足$Prj_a(S)=\{t|a(t)\in S\}$,$Prj_a(C)=Prj_a(S)$,其中:a是谓词名;t是项;S是状态;C是设置.

定义 3(秘密性). C是协议的设置,M是基本消息.C不保证M的秘密,当且仅当$\exists C'$使得$C\rightarrow^* C'$,且$M\in Prj_D(C')$.

因为MSR设置的推导是一个穷尽的过程,文献[4]证明了Dolev-Yao模型在密码算法不可破的前提下是最强的攻击者.所以,在密码算法安全的前提下,协议模型和攻击者模型交互执行,得到的所有设置描述了协议在攻击者环境下运行的所有状态.如果协议的秘密信息M泄露给攻击者,根据我们的Dolev-Yao攻击者模型,则必定存在设置C满足$M\in Prj_D(C)$.反过来说,"$\exists C'$使得$C\rightarrow^* C'$,且$M\in Prj_D(C')$"描述了在协议的某个设置中,攻击者分解记忆谓词含有秘密信息,即攻击者得到了协议的秘密信息.

2.2.2 认证性

Gavin Lowe 在文献[5]中给出了新鲜性、弱一致性、非单射一致性和单射一致性的定义.我们采用作者的思想,并根据 MSR 模型的特点,转换成 MSR 模型的认证性描述.

我们在角色中添加 *begin* 和 *end* 事件.当协议发起者发起一个会话时,发起者角色触发 *begin*(A,B)事件,*begin*(A,B)添加到当前状态中.协议响应者完成该次会话时,响应者角色触发 *end*(B,A)事件,*end*(B,A)添加到当前状态中.这两个事件中的实体参数 A,B 和事件所在角色中的角色状态谓词中的参数 A 和 B 相同,在参数实例化时要保持一致.正常协议运行中,*begin* 事件和 *end* 事件是一一对应的.接下来,我们把 Lowe 的定义描述成基于 MSR 模型状态的描述.

定义 4(弱一致性). C 是协议的初始设置,S 是设置的状态,Σ表示笔记,T 是协议的一条"完整"踪迹,A 是协议的发起方,B是协议的响应方.T违反了弱一致性,当且仅当 $\exists[S]_\Sigma^R,[S]_\Sigma^R\in T$,有$end(B,A)>^S begin(A,B)$."完整"踪迹的意思是该踪迹是协议的一次运行,这里说的"协议的一次运行"就是达成协议认证目的的运行.$end(B,A)>^S begin(A,B)$表示S中的$end(B,A)$事件的个数大于$begin(A,B)$事件的个数.

在协议正常运行过程中,协议发起者发起一个会话,然后由期望的协议响应者完成此次会话."$\exists[S]_\Sigma^R,[S]_\Sigma^R\in T$,有$end(B,A)>^S begin(A,B)$"表明在一次协议运行结束时,至少有一个$end(B,A)$事件没有与其对应

的*begin*(*A*,*B*)事件,即当协议的发起者*A*完成协议的一次运行时,协议响应方*B*之前没有与*A*运行过一次该协议,这违反了Lowe的弱一致性定义.反过来说,*T*违反弱一致性,则协议发起方*A*完成协议的一次运行之前*A*没有与响应方*B*发生通信,协议运行完毕后的状态*S*中必定包含*end*(*B*,*A*)事件;否则,协议不会完成,并且没有与*end*(*B*,*A*)事件相对应的*begin*(*A*,*B*)事件,所以,在踪迹*T*中有$end(B,A)>^{S}begin(A,B)$.

定义 5(非单射一致性). 设 *ds* 是在协议运行过程中的一组自由变量,*T* 是协议的一条"完整"踪迹,如果踪迹 *T* 满足弱一致性,且通信双方对 *ds* 中自变量有一致的看法,则踪迹 *T* 满足非单射一致性.

非单射一致性在弱一致性的基础上增加了双方对其会话对象的肯定条件,且要求发起者和响应者对用于交换的数据项的值保持一致.因此,非单射一致性相对于弱一致性对认证性的定义更进一步.对于 MSR 模型,*ds* 是通信双方两个角色包含的自变量,"通信双方对 *ds* 中自变量有一致的看法"表现在通信双方接收到的消息中各个参数值和角色中参数值是一致的.

定义 6(单射一致性). *T*是协议的一条"完整"踪迹,*S*是踪迹*T*的最后状态,*ds*是在协议运行过程中的一组自由变量,若$end(B,A)=^{S}begin(A,B)$,且通信双方对相应的*ds*中变量的数据值保持一致,则*T*满足单射一致性.其中,$end(B,A)=^{S}begin(A,B)$表示*S*中的*end*(*B*,*A*)事件对应于*begin*(*A*,*B*)事件.

Lowe的单射一致性是在非单射一致性的基础上对发起者和响应者之间通信做了进一步的约束,即"协议发起者*A*的每次协议的运行对应于协议响应者*B*的唯一一次运行",该约束可用$end(B,A)=^{S}begin(A,B)$来描述.

3 协议验证实例

我们用改进的 MSR 模型对简单 NS 公钥认证协议(Needham-Schroeder public-key protocol)进行分析.

3.1 协议描述

简单 NS 公钥认证协议非形式描述如下:

(1) $A \to B:\{\{n_A A\}\}_{k_B}$

(2) $B \to A:\{\{n_A,n_B\}\}_{k_A}$

(3) $A \to B:\{\{n_B\}\}_{k_B}$

协议的 MSR 模型描述如下:

$$\left(\begin{array}{l}\exists L:\ principal^{(A)}\times principal^{(B)}\times pubKB^{(k_B)}\times nonce.\\ \forall B: pricipal.\\ \forall k_B: pubKB. \quad START-A\xrightarrow{r11}\exists n_A: nonce.N(\{\{n_A A\}\}_{k_B})L(A,B,k_B,n_A)\\ \qquad\qquad\qquad\qquad\qquad\qquad begin(A,B)\\ \forall k_A: pubKA.\\ \forall k'_A: privKk_A. \quad N(\{\{n_A n_B\}\}_{k_A})L(A,B,k_B,n_A)\xrightarrow{r12}N(\{\{n_B\}\}_{k_B})\\ \forall n_A,n_B: nonce.\end{array}\right)^{\forall A}$$

$$\left(\begin{array}{l}\exists L: principal^{(B)}\times principal^{(B)}\times pubKA^{(k_A)}\times nonce\\ \forall k_B: pubKB.\\ \forall k'_B: privKk_B.\\ \forall A: principal \quad START-B.N(\{\{n_A A\}\}_{k_B})\xrightarrow{r21}\exists n_B: nonce.N(\{\{n_A n_B\}\}_{k_A})\\ \qquad\qquad\qquad\qquad\qquad\qquad L(B,A,k_A,n_B)\\ \forall n_A: nonce.\\ \forall k_A: pubKA.\\ \forall n_B: nonce. \quad N(\{\{n_B\}\}_{k_B})L(B,A,k_A,n_B)\xrightarrow{r22}end(A,B)\end{array}\right)^{\forall B}$$

3.2 协议验证过程

令第 1 个角色为ρ_1,第 2 个角色为ρ_2,初始设置为$[START-\mathrm{A},START-\mathrm{B}]_{\Sigma}^{\rho^{\mathrm{I}}}$.推理过程如下:

$$[START-\mathrm{A},START-\mathrm{B}]^{\rho^{\mathrm{I}}} \xrightarrow[\text{角色激活}]{\rho_1^A\rho_2^B} [START-\mathrm{A},START-\mathrm{B}]^{\rho^{\mathrm{I}},\rho_1^{\mathrm{A}},\rho_2^{\mathrm{B}}} \quad ①$$

推理进行到①处,下一步实例化角色的角色状态谓词,角色ρ_1有 3 种情况,它们是:

$$[\mathrm{l}_{\mathrm{A}}(\mathrm{A},\mathrm{A},k_{\mathrm{A}},n_{\mathrm{A}})/L]_{\rho_1},[\mathrm{l}_{\mathrm{A}}(\mathrm{A},\mathrm{B},k_{\mathrm{B}},n_{\mathrm{A}})/L]_{\rho_1},[\mathrm{l}_{\mathrm{A}}(\mathrm{A},\mathrm{I},k_{\mathrm{I}},n_{\mathrm{A}})/L]_{\rho_1}.$$

角色ρ_2也有 3 种情况,它们是:

$$[\mathrm{l}_{\mathrm{B}}(\mathrm{B},\mathrm{A},k_{\mathrm{A}},n_{\mathrm{B}})/L]_{\rho_2},[\mathrm{l}_{\mathrm{B}}(\mathrm{B},\mathrm{B},k_{\mathrm{B}},n_{\mathrm{B}})/L]_{\rho_2},[\mathrm{l}_{\mathrm{B}}(\mathrm{B},\mathrm{I},k_{\mathrm{I}},n_{\mathrm{B}})/L]_{\rho_2}.$$

那么实例化角色ρ_1和ρ_2的所有情况有 9 种:($[\mathrm{l}_{\mathrm{A}}(\mathrm{A},\mathrm{A},k_{\mathrm{A}},n_{\mathrm{A}})/L]_{\rho_1}$,$[\mathrm{l}_{\mathrm{B}}(\mathrm{B},\mathrm{A},k_{\mathrm{A}},n_{\mathrm{B}})/L]_{\rho_2}$),($[\mathrm{l}_{\mathrm{A}}(\mathrm{A},\mathrm{A},k_{\mathrm{A}},n_{\mathrm{A}})/L]_{\rho_1}$,$[\mathrm{l}_{\mathrm{B}}(\mathrm{B},\mathrm{B},k_{\mathrm{B}},n_{\mathrm{B}})/L]_{\rho_2}$),($[\mathrm{l}_{\mathrm{A}}(\mathrm{A},\mathrm{A},k_{\mathrm{A}},n_{\mathrm{A}})/L]_{\rho_1}$,$[\mathrm{l}_{\mathrm{B}}(\mathrm{B},\mathrm{I},k_{\mathrm{I}},n_{\mathrm{B}})/L]_{\rho_2}$),($[\mathrm{l}_{\mathrm{A}}(\mathrm{A},\mathrm{B},k_{\mathrm{B}},n_{\mathrm{A}})/L]_{\rho_1}$ $[\lambda_{\mathrm{B}}(\mathrm{B},\mathrm{A},k_{\mathrm{A}},n_{\mathrm{B}})/L]_{\rho_2}$),($[\mathrm{l}_{\mathrm{A}}(\mathrm{A},\mathrm{B},k_{\mathrm{B}},n_{\mathrm{A}})/L]_{\rho_1}$,$[\mathrm{l}_{\mathrm{B}}(\mathrm{B},\mathrm{B},k_{\mathrm{B}},n_{\mathrm{B}})/L]_{\rho_2}$),($[\mathrm{l}_{\mathrm{A}}(\mathrm{A},\mathrm{B},k_{\mathrm{B}},n_{\mathrm{A}})/L]_{\rho_1}$,$[\mathrm{l}_{\mathrm{B}}(\mathrm{B},\mathrm{I},k_{\mathrm{I}},n_{\mathrm{B}})/L]_{\rho_2}$),($[\mathrm{l}_{\mathrm{A}}(\mathrm{A},\mathrm{I},k_{\mathrm{I}},n_{\mathrm{A}})/L]_{\rho_1}$,$[\mathrm{l}_{\mathrm{B}}(\mathrm{B},\mathrm{A},k_{\mathrm{A}},n_{\mathrm{B}})/L]_{\rho_2}$),($[\mathrm{l}_{\mathrm{A}}(\mathrm{A},\mathrm{I},k_{\mathrm{I}},n_{\mathrm{A}})/L]_{\rho_1}$,$[\mathrm{l}_{\mathrm{B}}(\mathrm{B},\mathrm{B},k_{\mathrm{B}},n_{\mathrm{B}})/L]_{\rho_2}$),($[\mathrm{l}_{\mathrm{A}}(\mathrm{A},\mathrm{I},k_{\mathrm{I}},n_{\mathrm{A}})/L]_{\rho_1}$, $[\mathrm{l}_{\mathrm{B}}(\mathrm{B},\mathrm{I},k_{\mathrm{I}},n_{\mathrm{B}})/L]_{\rho_2}$).但$[\mathrm{l}_{A}(\mathrm{A},\mathrm{A},k_{\mathrm{A}},n_{\mathrm{A}})/L]_{\rho_1}$和$[\mathrm{l}_{\mathrm{B}}(\mathrm{B},\mathrm{B},k_{\mathrm{I}},n_{\mathrm{B}})/L]_{\rho_2}$不符合实际操作,即自己和自己进行通信是不可能的.排除包含$[\mathrm{l}_{\mathrm{A}}(\mathrm{A},\mathrm{A},k_{\mathrm{A}},n_{\mathrm{A}})/L]_{\rho_1}$和$[\mathrm{l}_{\mathrm{B}}(\mathrm{B},\mathrm{B},k_{\mathrm{I}},n_{\mathrm{B}})/L]_{\rho_2}$的组合,9 种组合还剩 4 种.下面接着①处进行推理.

① $\xrightarrow[\text{角色状态谓词实例化}]{[\mathrm{l}_{\mathrm{A}}(\mathrm{A},\mathrm{B},k_{\mathrm{B}},n_{\mathrm{A}})/L]\rho_1,[\mathrm{l}_{\mathrm{B}}(\mathrm{B},\mathrm{A},k_{\mathrm{A}},n_{\mathrm{B}})/L]\rho_2}$ 我们省略了这条分支的推理,从这里推导下去可以得到一条正确的协议执行踪迹.

$$① \xrightarrow[\text{角色状态谓词实例化}]{[\mathrm{l}_{\mathrm{A}}(\mathrm{A},\mathrm{B},k_{\mathrm{A}},n_{\mathrm{A}})/L]\rho_1,[\mathrm{l}_{\mathrm{B}}(\mathrm{B},\mathrm{I},k_{\mathrm{I}},n_{\mathrm{B}})/L]\rho_2} \quad ②$$

② $\xrightarrow[\text{应用规则}r11]{r11}$ $[START-\mathrm{B},N(\{\{n_{\mathrm{A}}\mathrm{A}\}\}_{k_{\mathrm{B}}}),L(\mathrm{A},\mathrm{B},k_{\mathrm{B}},n_{\mathrm{A}}),begin(\mathrm{A},\mathrm{B})]^{\rho^{\mathrm{I}},\rho_1^{\mathrm{A}}/r11,\rho_2^{\mathrm{B}}}$

$\xrightarrow[\text{攻击者操作}]{r_I}$

$[START-\mathrm{B},N(\{\{n_{\mathrm{A}}\mathrm{A}\}\}_{k_{\mathrm{B}}}),L(\mathrm{A},\mathrm{B},k_{\mathrm{B}},n_{\mathrm{A}}),begin(\mathrm{A},\mathrm{B}),N(\{\{n_{\mathrm{I}}\mathrm{I}\}\}_{k_{\mathrm{B}}})]^{\rho^{\mathrm{I}},\rho_1^{\mathrm{A}}/r11,\rho_2^{\mathrm{B}}}$

$\xrightarrow[\text{应用规则}r21]{r21}$

$[N(\{\{n_{\mathrm{A}}\mathrm{A}\}\}_{k_{\mathrm{B}}}),L(\mathrm{A},\mathrm{B},k_{\mathrm{B}},n_{\mathrm{A}}),begin(\mathrm{A},\mathrm{B}),N(\{\{n_{\mathrm{I}}n_{\mathrm{B}}\}\}_{k_{\mathrm{I}}})]^{\rho^{\mathrm{I}},\rho_1^{\mathrm{A}}/r11,\rho_2^{\mathrm{B}}/r21}$

该踪迹推理到此无法再进行下去,即没有规则可供应用.

② $\xrightarrow[\text{跳过规则}r11,\text{攻击者操作}]{skip(r11),r_I}$

$[START-\mathrm{A},START-\mathrm{B},N(\{\{n_{\mathrm{I}}\mathrm{I}\}\}_{k_{\mathrm{B}}})]^{\rho^{\mathrm{I}},\rho_1^{\mathrm{A}},\rho_2^{\mathrm{B}}}$

$\xrightarrow[\text{应用规则}r21]{r21}$

$[START-\mathrm{A},N(\{\{n_{\mathrm{I}}n_{\mathrm{B}}\}\}_{k_{\mathrm{I}}})]^{\rho^{\mathrm{I}},\rho_1^{\mathrm{A}},\rho_2^{\mathrm{B}}}$ 该踪迹表示攻击者 I 和 B 进行通信,这和 A 与 B 进行通信没有什么区别,所以没有必要再对其进行推理.

① $\xrightarrow[\text{角色状态谓词实例化}]{[\mathrm{l}_{\mathrm{A}}(\mathrm{A},\mathrm{I},k_{\mathrm{A}},n_{\mathrm{A}})/L]\rho_1,[\mathrm{l}_{\mathrm{B}}(\mathrm{B},\mathrm{A},k_{\mathrm{I}},n_{\mathrm{B}})/L]\rho_2}$

$\xrightarrow[\text{应用规则}r11]{r11}$

$$[START-\mathrm{B},N(\{\{n_{\mathrm{A}}\mathrm{A}\}\}_{k_{\mathrm{I}}}),L(\mathrm{A},\mathrm{I},k_{\mathrm{I}},n_{\mathrm{A}}),begin(\mathrm{A},\mathrm{I})]^{\rho^{\mathrm{I}},\rho_1^{\mathrm{A}}/r11,\rho_2^{\mathrm{B}}} \quad ③$$

③ $\xrightarrow[\text{攻击者操作}]{r_{\mathrm{I}}}$

$[START-\mathrm{B},L(\mathrm{A},\mathrm{I},k_{\mathrm{I}},n_{\mathrm{A}}),begin(\mathrm{A},\mathrm{I}),N(\{\{n_{\mathrm{A}}\mathrm{A}\}\}_{k_{\mathrm{B}}})]^{\rho^{\mathrm{I}},\rho_1^{\mathrm{A}}/r11,\rho_2^{\mathrm{B}}}$

$\xrightarrow[\text{应用规则}r21]{r21}$

$[L(\mathrm{A},\mathrm{I},k_{\mathrm{I}},n_{\mathrm{A}}),begin(\mathrm{A},\mathrm{I}),N(\{\{n_{\mathrm{A}}n_{\mathrm{B}}\}\}_{k_{\mathrm{A}}}),L'(\mathrm{B},\mathrm{A},k_{\mathrm{A}},n_{\mathrm{B}})]^{\rho^{\mathrm{I}},\rho_1^{\mathrm{A}}/r11,\rho_2^{\mathrm{B}}/r21}$

$\xrightarrow[\text{攻击者操作}]{r_1}$

$[L(\mathrm{A},\mathrm{I},k_1,n_\mathrm{A}),begin(\mathrm{A},\mathrm{I}),N(\{\{n_\mathrm{A}n_\mathrm{B}\}\}_{k_\mathrm{A}}),L'(\mathrm{B},\mathrm{A},k_\mathrm{A},n_\mathrm{B})]^{\rho^\mathrm{I},\rho_1^\mathrm{A}/r11,\rho_2^\mathrm{B}/r21}$

$\xrightarrow[\text{应用规则}r12]{r12}$

$[begin(\mathrm{A},\mathrm{I}),L'(\mathrm{B},\mathrm{A},k_\mathrm{A},n_\mathrm{B}),N(\{\{n_\mathrm{B}\}\}_{k_\mathrm{B}})]^{\rho^\mathrm{I},\rho_1^\mathrm{A}/r11/r12,\rho_2^\mathrm{B}/r21}$

$\xrightarrow[\text{攻击者操作}]{r_1}$

$[begin(\mathrm{A},\mathrm{I}),L'(\mathrm{B},\mathrm{A},k_\mathrm{A},n_\mathrm{B}),N(\{\{n_\mathrm{B}\}\}_{k_\mathrm{B}})]^{\rho^\mathrm{I},\rho_1^\mathrm{A}/r11/r12,\rho_2^\mathrm{B}/r21}$

$\xrightarrow[\text{应用规则}r22]{r22}$

$[begin(\mathrm{A},\mathrm{I}),end(\mathrm{A},\mathrm{B})]^{\rho^\mathrm{I},\rho_1^\mathrm{A}/r11/r12,\rho_2^\mathrm{B}/r21/r22}$

根据定义 4,这条踪迹违反了弱一致性.

③ $\xrightarrow[\text{攻击者操作}]{r_1}$

$[START-\mathrm{B},N(\{\{n_\mathrm{A}\mathrm{A}\}\}_{k_\mathrm{I}}),L(\mathrm{A},\mathrm{I},k_1,n_\mathrm{A}),begin(\mathrm{A},\mathrm{I}),N(\{\{n_\mathrm{I}\mathrm{A}\}\}_{k_\mathrm{B}})]^{\rho^\mathrm{I},\rho_1^\mathrm{A}/r11,\rho_2^\mathrm{B}}$

$\xrightarrow[\text{应用规则}r21]{r21}$

$[N(\{\{n_\mathrm{A}\mathrm{A}\}\}_{k_\mathrm{I}}),L(\mathrm{A},\mathrm{I},k_\mathrm{I},n_\mathrm{A}),begin(\mathrm{A},\mathrm{I}),N(\{\{n_\mathrm{I}n_\mathrm{B}\}\}_{k_\mathrm{A}})]^{\rho^\mathrm{I},\rho_1^\mathrm{A}/r11,\rho_2^\mathrm{B}/r21}$

该踪迹推理到此无法再进行下去,即没有规则可供应用.

① $\xrightarrow[\text{角色状态谓词实例化}]{[1_\mathrm{A}(\mathrm{A},\mathrm{I},k_\mathrm{A},n_\mathrm{A})/L]\rho_1,[1_\mathrm{B}(\mathrm{B},\mathrm{I},k_\mathrm{I},n_\mathrm{B})/L]\rho_2}$ 该实例化表示 A 和攻击者 I 通信,B 和攻击者 I 通信.攻击者没有必要破坏自己的通信,所以,这种情况没有必要分析.

验证结果表明该协议不满足弱一致性,Lowe 针对该漏洞对协议进行了修改.

4 结束语

MSR 模型是一种精确的协议描述框架,一种密码协议建模方法.文献[3,6,7]用该模型对几种协议进行了建模.但是,该模型自己的形式系统并不完善,所以在验证协议时,需要结合其他形式系统.针对这个问题,我们完善了该模型,并使其更适合于验证密码协议.另外,该模型的攻击者模型在协议验证时存在着不确定性,我们对攻击者模型进行了改进,使其可以更有效地验证密码协议.我们在以后的工作中将进一步完善 MSR 模型,使其不仅能分析认证协议,还能分析更复杂的协议,如电子商务协议等.

致谢 本文在撰写过程中得到了北京航空航天大学李先贤教授和中国科学院软件研究所季庆光博士的细心指导,在此,向他们表示最诚挚的谢意.

References:

[1] Ji QG, Feng DG. Towards analyzing some kinds of critically formal models for network security protocols. Chinese Journal of Computers, 2005,28(7):1071−1083 (in Chinese with English abstract).

[2] Cervesato I, Durgin NA, Lincoln PD, Mitchell JC, Scedrov A. A meta-notation for protocol analysis. In: Syverson P, ed. Proc. of the 12th IEEE Computer Security Foundations Workshop—CSFW'99. Mordano: IEEE Computer Society Press, 1999. 55−69.

[3] Cervesato I. A specification language for crypto-protocols based on multiset rewriting, dependent types and subsorting. In Delzanno G, Etalle S, Gabbrielli M, eds. Proc. of the Workshop on Specification, Analysis and Validation for Emerging Technologies—SAVE'01. Paphos, 2001. 1−22.

[4] Cervesato I. Data access specification and the most powerful symbolic attacker in MSR. In: Software Security, Theories and Systems. LNCS 2609, Springer-Verlag, 2003. 384−416. http://citeseer.ist.psu.edu/cervesato02data.html

[5] Lowe G. A hierarchy of authentication specifications. In: Proc. of the 10th IEEE Computer Security Foundations Workshop. Los Alamitos: IEEE Computer Society Press, 1997. 31−43.

[6] Cervesato I. Typed MSR: Syntax and examples. In: Proc. of the 1st Int'l Workshop on Mathematical Methods, Models and Architectures for Computer Network Security—MMM 2001. St. Petersburg: Springer-Verlag, 2001. 159−176. http://citeseer.ist.psu.edu/cervesato01typed.html

[7] Butler F, Cervesato I. Jaggard A, Scedrov A. A formal analysis of some properties of Kerberos 5 using MSR. Technical Report, MS-CIS-04-04, Philadelphia: University of Pennsylvania, 2004. http://theory. stanford.edu/~iliano/by Year.html

附中文参考文献:

[1] 季庆光,冯登国.对几类重要网络安全协议形式模型的分析.计算机学报,2005,28(7):1071−1083.

张畅(1981－),男,湖北当阳人,助教,主要研究领域为协议分析.

王亚弟(1953－),男,教授,博士生导师,主要研究领域为信息安全,计算机应用.

韩继红(1966－),女,博士,副教授,主要研究领域为信息安全,计算机应用.

郭渊博(1975－),男,博士,副教授,CCF 会员,主要研究领域为秘密共享,容忍入侵,密码协议形式化设计.