协议入侵者攻击能力的统一建模

张 靖

(攀枝花学院网络中心 四川 攀枝花 617000)

【摘要】基于协议预言机模型是一个开放的模型，能为准确地描述入侵者的攻击能力提供统一的框架。该文提出了一种协议预言机模型，使用协议预言机来形式化地描述入侵者对协议的攻击能力，通过协议生成树，该模型可以跟踪入侵者攻击协议的轨迹。

关 键 词 形式化分析; 入侵者攻击能力; 协议预言机; 协议生成树
中图分类号 TP309 文献标识码 A

Unified Modeling of the Intruder's Attack Ability Based on the Protocols

ZHANG Jing

(Campus Network Center, Panzhihu University Panzhihua Sichuan 617000)

Abstract Protocols oracle model is an open model, which provides a unified framework to describe the intruder's attack ability precisely. Therefore, This article presents a protocols oracle model which makes a formalized description of the intruder's attack ability on the protocols. This model can also follow the tracks of the intruder's attack on the protocols through the protocols spanning tree.

Key words formal analysis; intruder's attack ability; protocols oracle; protocols spanning tree

自20世纪80年代末BAN逻辑[1]被用来分析密码协议的安全性后，很多方法被设计来进行密码协议的形式化分析[2]，且都基于一个假设，即密码协议中的密码算法是没有漏洞的，或者说密码算法的漏洞由算法分析。而协议漏洞分析主要针对协议消息交互产生的漏洞，即著名的Dolev-Yao安全模型[3]。在密码协议的形式化分析中，对入侵者攻击协议消息交互的能力的描述是影响协议分析方法有效性的一个重要因素。因此，在Dolev-Yao安全模型中，协议入侵者被赋予了以下的能力[3]：(1) 它可以得到任何网络上传输的消息；(2) 它是网络的合法用户，可以与任何其他用户发起会话；(3) 它有机会成为某个用户的消息接收者。几乎所有的协议分析方法都沿用了Dolev-Yao的入侵者模型。不同的分析方法根据Dolev-Yao的入侵者模型为自己的方法定义了入侵者攻击能力，最典型的是串空间模型。文献[4]在串空间分析方法中，根据Dolev-Yao的入侵者模型定义了8个入侵者串，通过分析协议串是否包含入侵者串，判断协议是否满足安全属性。文献[4]根据Dolev-Yao的入侵者模型定义了8个入侵者可能的操作，并证明具有这8个操作的入侵者是攻击能力最强的入侵者。

Dolev-Yao的入侵者模型没有统一的形式化描述，造成各个分析方法对协议入侵者的攻击能力不能完整地进行形式化的建模。如文献[5]通过增加B和Bf两个入侵者串，对串空间方法中的8个入侵者串进行补充，从而扩大了串空间方法的适用范围。此外，入侵者可以利用协议自身作为入侵工具，但在Dolev-Yao的入侵者模型中不会被明确地体现出来。

因此，本文提出了协议预言机模型，它是一个开放的模型，可将入侵者的密码攻击能力和利用协议自身作为攻击工具的攻击能力统一在一个开放的框架中。

1 协议消息计算模型

协议的运行是通过协议消息的交换来完成的，典型的协议消息是由子消息通过组合和密码函数的计算得到。本文采用文献[6]提出的原子消息的定义，并在此基础上构造了协议消息计算模型。

定义 1 原子消息集合M_0由四种原子消息组成：(1) 协议实体标识集合P_{ID}，集合中的每个元素

收稿日期：2006 –10 –09
作者简介：张 靖(1972 –), 男, 硕士, 副教授，主要从事软件工程和计算机网络方面的研究.

表示一个协议实体；(2) 密钥集合K，集合包括各实体的公钥、私钥、会话密钥、共享密钥等；(3) 新鲜值集合N，协议实体在协议运行期间产生的临时值。(4) 数据消息D，实体之间通过协议传送的随机数据或信息，即$M_0=P_{\mathrm{ID}}\cup K\cup N\cup D$。

定义 2 消息的三种操作为：(1) 消息联合操作符η(或·)，两个消息通过联合操作符得到一个新的消息；(2) 逆密钥操作符∂，密钥集合中的密钥通过密钥操作符得到另一个相对应的密钥；(3) 密码函数操作符δ，消息通过密码函数计算，可以得到一个新的消息。

对于三种操作符中的密码函数操作符$\delta(m,k)$，通过密钥k对消息m作密码计算，用符号E表示通过密码函数计算得到的消息集合。在文献[4]和文献[7]提出的“ 自由假设 ”下，有$P_{\mathrm{ID}}\cap K\cap N=\phi$，如果$\delta(m_0,k_0)=\delta(m_1,k_1)\in E$，那么$m_0=m_1$，$k_0=k_1$，如果$m_0m_1=m_0'm_1'$，那么$m_0=m_0'$，且$m_1=m_1'$成立。

定义 3 消息m是简单消息，当且仅当$m\in(M_0\cap E)$，用符号s表示。反之，则称消息m为复合消息。

定义 4 在原子消息和消息的3种操作的基础上，协议π所能产生的所有消息的集合M可由以下的方法得到：(1) 如果$m\in M_0$，那么$m\in M$；(2) 如果$m_0\in M$，且$m_1\in M$，那么$\eta(m_0,m_1)\in M$(或用$m_0\cdot m_1$表示)；(3) 如果$m\in M$，且$k\in K$，那么$\delta(m,k)\in M$。

定理 1 任意消息都可以表示成简单消息的联合，即$\forall m\in M$，$\exists m=s_0\cdot s_1\cdot\cdots\cdot s_n$，其中$s_i$是简单消息，$0\le i\le n$。

定理 2 任意消息存在一个唯一的简单消息联合表达式，即$\forall m\in M$，如果$m=s_0\cdot s_1\cdot\cdots\cdot s_M$，$m=s_0's_1'\cdot\cdots\cdot s_N'$，那么$M=N$，且$\forall j$，$s_j'=s_j$，即所有对应的简单消息相等。

定义 5 因为每个协议消息都有一个唯一确定的简单消息联合表达式，按如下方式定义协议消息矩阵$M_\Pi=[M_1,M_2,\cdots,M_n]'$，其中$M_i=[s_{i_1}\cdot s_{i_2}\cdot\cdots\cdot s_{in_i}]$，$s_{ij}$表示是消息$M_i$的第$j$个简单消息，而$n_i$是消息$M_i$的简单消息的个数。

协议消息的交互规则是由密码协议规定的。可用图1的协议运行模型表示协议消息和规则之间的关系。图中的“ 步骤 ”i都是由某个协议实体执行的，其输入是消息m_{i-1}，输出是消息m_i。协议实体在执行协议步骤时会执行消息分解操作和消息合成操作两类操作。其中的分解操作将消息分解成协议规则中该步骤可以直接操作的子消息；而合成操作则是根据协议规则中对该步骤的规定产生输出消息。两类操作通过四个规则体现。

图1 协议运行模型

1) 组合规则为：$\frac{m_1m_2}{m_1\cdot m_2}$，该规则对应消息联合操作符$\eta$，用符号$r_c$表示，特殊情况为$\frac{m}{m}$。2) 分解规则分别为：(1) $\frac{m_1\cdot m_2}{m_1}$，取联合消息中左边的子消息，用符号r_{dl}表示；(2) $\frac{m_1\cdot m_2}{m_2}$，取联合消息中右边的子消息，符号r_{dr}表示。3) 密码函数规则为：$\frac{m\quad[k]}{\delta(m,k)}$，该规则对应密码函数操作符$\delta$，符号$r_\delta$表示。4) 解密码函数规则为：$\frac{\delta(m,k)\quad\partial(k)}{m}$，该规则对应密码函数的逆函数，符号$r_\partial$表示。

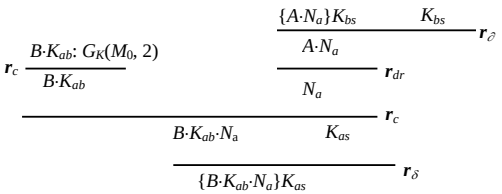

图2 消息$\{B, K_{ab}, N_a\}K_{as}$的生成树

除以上四个规则外，“ 步骤 ”i的执行者P还可能执行的一个规则(或者说动作)是“ 产生新原子消息规则 ”，用$G_m(M_0,i)$表示，即P在步骤i产生一个消息m，且$m\in M_0$，同时用$G(M_0,i)$表示P在步骤i产生的所有新原子消息。在执行该步骤i时，实体P所拥有的消息集合为$M_P(i)=M_P(i-1)\cup M_{i-1}\cup G(M_0,i)$，$M_{i-1}$为步骤$i$的输入消息集合。利用上述规则和消息集合$M_P$，$P$可以生成输出消息$m_i$。由于$m_i=s_0\cdot s_1\cdot\cdots\cdot s_n$，因此，其中每个简单消息$s_i$都有一个由消息集合$M_p$得到的生成树$T_i(P)$，即$(M_T)T_i(P)=s_i$，表示在协议步骤$i$由实体$P$通过生成树$T_i$产生$S_i$，其中$M_T$是由生成树的“ 叶子 ”构成的消息，且$M_T\subset M_p$。对于消息自身到自身的情况，本文称为零阶生成树，用符号$(s_i)T_0=s^i$表示。图2表示简化版Yahalom协议[1]第二步的简单消息$\{B,K_{ab},N_a\}K_as$的生成树(其执行者是密钥服务器S)，输入消息为$\{B,N_b,\{A,N_a\}K_{bs}\}$，输出消息为$\{N_b,\{B,K_{ab},N_a\}K_{as},\{A,K_{ab},N_b\}K_{bs}\}$)，该生成树的“ 叶子 ”为：

$$M_T=\{B,K_{ab},\{A,N_a\}K_{bs},K_{bs},K_{as}\}\subset M_p=\{A,B,N_b,\{A,N_a\}K_{bs},K_{bs},K_{as},K_{ab}\}$$

2　协议预言机模型

为了攻击密码协议，在保持“自由假设”[4,8]的前提下，最强的协议入侵者需能监听/截获所有网络消息和知道协议运行规范。而在形式上，入侵者对协议的攻击有两个途径：(1) 用自己拥有的消息和密钥构造假消息、解析秘密消息；(2) 通过参与协议运行，让协议参与者产生自己想要的消息。因此，入侵者利用协议规范作为入侵工具来攻击协议。为形式化地描述入侵者的这种攻击能力，本文提出了协议预言机模型，即协议规范是入侵者的预言机，根据入侵者提供的消息产生入侵者希望得到的数据或者拒绝入侵者的请求。

入侵者对于协议的攻击方法有4种类型：(1) 利用已知消息M_I和密钥K_I构造符合协议规范的假消息，M_I和K_I分别为入侵者拥有的消息和密钥；(2) 利用协议步骤i中的消息m，在协议步骤j中重放，$j \neq i$；(3) 利用协议的多个运行构造符合协议规范的假消息；(4) 利用协议的多个运行产生入侵者符合协议规范且可被入侵者解密的消息，该消息在协议的正常运行中是秘密消息。可见，攻击手段核心是构造符合协议规范的假消息，而攻击工具就是协议规范，用如图3所示协议预言机模型来描述。其中M_I和K_I分别为入侵者拥有的消息和密钥，它作为协议预言机的输入，通过协议规范和预言规则产生符合协议规范的消息Ms。

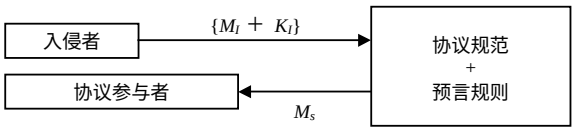

图3　预言机模型

在引入“预言规则”之前，须定义两个基本概念，即不可识别消息和相似消息。

定义 6　消息m为实体P的不可识别消息，当且仅当$m \notin M(P)$。其中$M(P)$表示实体P在消息集合$M_0(P)$的基础上，通过定义4得到的消息集合，且$M_0(P)$表示实体P拥有的密钥集合、实体P产生的临时值和数据集合、以及协议实体集合，即$M_0(P)$是实体P可识别的原子消息。本文用Unknown(P)表示实体P所不识别的消息的集合。如果消息m不是由实体P可识别的原子消息构成的消息，那么对于实体P来说该消息就是不可识别的消息。

定义 7　当满足以下五个条件之一时，简单消息s与简单消息s'相似，用Like(s,s')表示。(1) $s \in P_{ID}$，且$s' \in P_{ID}$；(2) $s \in K$，且$s' \in K$；(3) $s \in N$，$s' \in N$；(4) $s \in D$，$s' \in D$；(5) $s=\delta(m_1,k) \in E$，且$s'=\delta(m'_1 m'_1,k) \in E$，且Like($m_1, m'_1$)。

对于消息$m(m=s_0 \cdot s_1 \cdot \cdots \cdot s_M)$和$m'(m'=s'_0 \cdot s'_1 \cdot \cdots \cdot s'_N)$，当其简单消息的个数相等，即$M=N$，并且相对应的简单消息相似，即$\forall i$，Like($s_i$, s'_1)，那么两个消息相似，即Like(m, m')。可见，定义7是个递归定义，它是一个强的相似性定义，它在相似性规则中不包含协议实体的知识，因此，其相似性是对协议而言的。结合定义6，还可以得到一个与协议实体相关的相似性定义。

定义 8　当同时满足以两个条件时，消息$m(m=s_0 \cdot s_1 \cdot \cdots \cdot s_M)$和$m'(m'=s'_0 \cdot s'_1 \cdot \cdots \cdot s'_N)$对于实体$P$是相似的，用Like$_P$($m$,$m'$)表示。(1) 如果对于消息$m$和$m'$，有$\exists i$，使得Like($s_i$,$s'_i$)，那么$m$可表示为$m=m_0 \cdot s_i \cdot m_1$；$m'$可表示为$m'=m'_0 \cdot s'_i \cdot m'_1$。(2) $\{m_0 \cdot m_1 \cdot m'_0 \cdot m'_1\} \subset$ Unknown(P)，且$m_\pi=m_{\pi_0} \cdot s_i \cdot m_{\pi_1}$（其中$\{m_{\pi_0}, m_{\pi_1}\} \subset$ Unknown(P)）是实体P在协议规范中的一个可以处理的协议消息。

定义 9　如果消息m和m'满足定义8，且其中的两个简单消息s_i和s'_i相等，即$s_i=s'_i$，那么消息m和m'对于实体P是相等的，用Equal$_P$(m,m')表示。

定义 10　如果$m=\delta(m',k)$，且m'在整个协议运行中都是以密文方式出现在协议消息中的，那么说m'是秘密消息，即$m' \notin M_\Pi$。

定理 3　如果m'是秘密消息，且$m=\delta(m',k)$，则m的生成树里必然包含形如$m''=\delta(m',k')$的简单消息，即m的生成树为$(M_T)T_i$，则$\exists m'' \in M_T$使得$m''=\delta(m', k')$。

入侵者利用协议规范作为工具，让协议产生入侵者需要的消息，并以此欺骗协议执行者或者得到协议执行的某些秘密。协议规范作为入侵者对于消息的“预言者”，它根据入侵者提供的消息数据“预言”可能的新消息。在图3中，协议规范的作用体现在预言规则中，通过预言规则为入侵者产生可能的新消息。

定义 11　若消息m中的简单消息s_i的生成树为$T_i(P)$，其“叶子”为M_T，如果$M_T \subset M_I$，那么该消息s_i的生成树对于入侵者而言是一棵预言树，用符号$(M_T)T_\pi(s_i,i,P)$表示，即在步骤i由P产生的简单消息s_i有一棵入侵者的预言树。

预言树的存在说明一个简单消息可以由入侵者构造而不仅仅是协议运行的某个参与者。而下面的预言规则则说明了入侵者如何利用预言树来构造协议运行消息的。其中，预言规则来自上面提到的四种攻击类型。

1) 消息相等规则为：如果m_j是实体P在协议步

骤j的输入消息，m_i是协议规范的步骤i的消息，且$i\neq j$，如果$\mathrm{Equal}_P(m_j,m_i)$，那么$m_i$可以对实体$P$在协议步骤$j$的时候重放。实际上这个时候存在一个零阶预言树$T\pi_0(s_i,j,P)$。

2) 消息伪造规则为：$m=s_0\cdot s_1\cdot\ \cdots\ \cdot s_M$；$M=\{s_0\cdot s_1\cdot\ \cdots\ \cdot s_M\}$；$M_\Delta=M-(\mathrm{M}\cap M_I)$，$M_\Delta$表示消息集合$M$中，入侵者不拥有的消息集合。$M_\Sigma=M_\Delta-(M_\Delta\cap \mathrm{Unknown}(P))$，$M_\Sigma$表示在入侵者不拥有的消息集合中，可以被实体$P$识别的消息集合。有四种情况：(1) $M_\Delta=\phi$，$M_\Sigma=\phi$。表明m可以由入侵者的消息集合M_I组合得到，而不是由协议运行的合法参与者产生。(2) $M_\Delta=\phi$，$M_\Sigma\neq\phi$。其实不存在，因为由M_Σ的表达式可知，如果M_Δ为空，则M_Σ必为空。(3) $M_\Delta\neq\phi$，$M_\Sigma=\phi$。入侵者不拥有的消息对于实体P而言都是不可识别的，那么任意的数据都是可行的，则与情况(1)一样，m可以由入侵者的消息集合M_I与任意数据组合得到，而不是由协议运行的合法参与者产生。(4) $M_\Delta\neq\phi$，$M_\Sigma\neq\phi$。对于M_Σ中的每个元素s_i，有两种情况：(1) 如果$\exists s_k\in M_\Pi$，且$\mathrm{Equal}_P(s_i,s_k)$，若$s_k$有预言树存在，那么$m$可以由入侵者的消息集合$M_I$与任意数据组合得到；(2) 如果$\exists s_k\in M_\Pi$，且$\mathrm{Like}P(s_i,s_k)$，$s_k$的生成树为$(M_k)T$，如果$M_k\subset M_I$，那么可能存在一个预言树使得$(M'_K)T_\pi(s_i,i,P)$，其中$M_K'\subset M_\mathrm{I}$。其中后者是最复杂的情况，并且与具体的协议有关。

3) 消息解密规则为：m'是秘密消息，且$m=\delta(m',k)$，$k\notin K_I$，$m\in M_\Pi$，m的生成树为$(M_T)T_i(P)$，根据定理3，$\exists(m''=\delta(m',k'),k)\in M_T$。为了构造$m_I=\delta(m',k_I)$，其中$k_I\in K_I$（即消息$m'$成为入侵者可以得到的秘密），假设$M_T\subset M_I$，则存在四种情况：(1) 使用$m$的生成树$(M_T)T$，$m$为叶子元素之一；(2) 使用$m$的生成树$(M_T)_T$，$m''$为叶子元素之一；(3) 使用$m$的相似消息$m_1$的生成树$(M_{T_1})T_1$，$m$为叶子元素之一；(4) 使用$m_1$的生成树$(M_{T_1})T_1$，$m''$为叶子元素之一。如果$M_\Delta\neq\phi$，则通过规则2)的消息计算之后的消息仍然可以分为这四种情况。故根据假设$M_T\subset M_I$来设置四种情况是具有一般性的。情况(1)中，由于$(m''=\delta(m',k'),k)\in M_T$，如果替换其中的$m''$和$k$分别为$m$和$k_I$，且把与$k$有关的消息都替换成与$k_I$有关的消息，如果仍然存在该生成树$T$，那么入侵者可以得到$m_I=\delta(m',k_I)$。情况(2)同情况(1)，只是将$k$以及与$k$有关的消息分别替换成$k_I$ 和与k_I 有关的消息。情况(3)与情况(4)其实是一样的，因为对于m的“相似”消息来说，m与m''没有区别。需要说明的是，情况(3)的“相似”与定义7和定义8的“Like”略有不同，它没有定义的那么严格，只要两个消息的加密处理函数一样就可以了，即如果$m=\delta_1(m_1,k_1)$，$m'=\delta_2(m_2,k_2)$，如果$\delta_1=\delta_2$，那么$m\approx m'$。从这个角度来说情况(3)和(4)的处理与情况(1)一样。

以上可见，预言树对于预言规则是非常重要的，还可以通过预言树找到入侵者攻击协议的轨迹。例如，对于简单消息s_i，其生成树为$(M_T)T_i(P)$，若$M_T\subset M_I$，则T_i就是一棵预言树，如果存在$s_k\in M_T$，且$s\notin M_I$，则用与分析s_i的方法分析s_k，一直分析下去直到M_T的每一个不属于M_I的简单消息都有一棵预言树，那么s_i就是入侵者可以构造的，且其中多有的预言树构成了入侵者构造s_i的轨迹。

3 结 论

本文在协议消息计算模型的基础上，用协议预言机模型来形式化地描述入侵者利用协议规范作为工具来攻击协议运行的能力。通过实例分析及使用情况，可以准确地描述入侵者是如何利用协议规范来攻击协议运行的，提高协议安全性的形式化分析方法是有效性的重要方法。该模型是一个开放的模型，可以通过添加新的预言规则来形式化地描述新的入侵者攻击能力。但在预言规则里，如何形式化地描述相似消息到相等消息的转换，如何更准确地描述生成树表达式等都是有待解决的问题，同时预言机与协议安全性之间的关系仍需要继续研究。

参 考 文 献

[1] BURROWS M, ABADI M, NEEDHAM R. A logic of authentication[J]. In Proceedings of the Royal Society of London, 1989，426: 233-271.
[2] MEADOWS C. Formal methods for cryptographic protocol analysis: emerging issues and trends[J]. IEEE Journal on Selected Areas in Communication, 2003, 21(1): 44-54.
[3] DOLEV D, YAO A. On the security of public-key protocols[J]. IEEE Transactions on Information Theory, 1983, 29(2): 198-208.
[4] JAVIER F, ABREGA T, JONATHAN C. Strand spaces: Proving security protocols correct[J]. Journal of Computer Security, 1999, 7(2-3): 191-230.
[5] GLARKE E M, JHA S, MARRERO W. Verifying security protocols with brutus[J]. ACM Transactions on Software Engineering and Methodology, 2000, 9(4): 443-487.
[6] MANEKI A P. Honest functions and their application to the analysis of cryptographic protocol[C]//. In: Proc of the 1999 IEEE Computer Security Foundations Workshop. IEEE Computer Society. Washington, DC, USA[s.n.]: 1999.
[7] 李梦君. 安全协议形式化验证技术的研究与实现[EB/OL]. http://www.cnki.cn, 2007-02-03.
[8] 王焕宝. 安全协议分析的形式化理论与方法[EB/OL]. http://www.cnki.cn, 2006-09-10.

编 辑 熊思亮