认证协议两种形式化分析方法的比较*

卿斯汉+

(中国科学院 信息安全技术工程研究中心,北京 100080)
(中国科学院 软件研究所 信息安全国家重点实验室,北京 100080)

A Comparison Between Two Formal Analysis Methods on Authentication Protocols

QING Si-Han+

(Engineering Research Center for Information Security Technology, The Chinese Academy of Sciences, Beijing 100080, China)
(State Key Laboratory of Information Security, Institute of Software, The Chinese Academy of Sciences, Beijing 100080, China)
+ Corresponding author: Phn: 86-10-62635150, Fax: 86-10-62635150, E-mail: qsihan@yahoo.com
http://www.ercist.iscas.ac.cn
Received 2003-05-19; Accepted 2003-06-30

Qing SH. A comparison between two formal analysis methods on authentication protocols. *Journal of Software*, 2003,14(12):2028~2036.
http://www.jos.org.cn/1000-9825/14/2028.htm

Abstract: Strand space model and CSP method are two popular approaches to formal analysis for authentication protocols. In this paper, the different characteristics of the above approaches are outlined through a concrete authentication protocol.
Key words: authentication protocol; formal analysis; strand space model; CSP method

摘 要: 串空间模型和 CSP 方法是当前最著名的分析认证协议的形式化方法.通过一个具体的认证协议例子,比较两种方法的不同特点.
关键词: 认证协议;形式化分析;串空间模型;CSP 方法
中图法分类号: TP309 **文献标识码**: A

认证协议的设计是一个众所周知的难题,在用非形式化的方法进行分析时容易发生错误,这方面的例子是很多的[1,2].近 20 年来,认证协议的设计与形式化分析一直受到重视和广泛关注,涌现出了众多的研究方法.目前,基于串空间模型的方法[3~6]和基于顺序通信进程的方法——CSP[7,8]最为著名,它们都基于协议迹进行分析,严格地规范了攻击者的能力与运行环境,使我们对认证协议的设计和分析的认识更加深入.

本文通过一个具体的认证协议,说明应用串空间模型和 CSP 方法分析的不同特点.

* Supported by the National Natural Science Foundation of China under Grant No.60083007 (国家自然科学基金); the National Grand Fundamental Research 973 Program of China under Grant No.G1999035810 (国家重点基础研究发展规划(973))

第一作者简介: 卿斯汉(1939－),男,湖南隆回人,研究员,博士生导师,主要研究领域为信息系统安全理论和技术.

1 串空间模型

本节介绍串空间模型的基本符号、概念以及构造串空间的方法,特别阐述基于串空间理论的认证测试方法,它比单纯应用串空间模型更为简捷与直观.有关串空间理论的详细介绍,请见文献[3~6].

1.1 基本概念

令集合 A 中的元素为协议主体交换的消息,称 A 的元素为项.项中最重要的关系是子项关系,记为$\sqsubset$.$t_0\sqsubset t_1$ 表示 t_0 是 t_1 的子项.

子项关系$\sqsubset$递归定义如下:

(1) $a\sqsubset a$;

(2) $a\sqsubset\{g\}_K$,如果 $a\sqsubset g$;

(3) $a\sqsubset gh$,如果 $a\sqsubset g\vee a\sqsubset h$.

符号项是一个二元组$\langle\sigma,a\rangle$,其中 $a\in A$ 且$\sigma=\{+,-\}$,记符号项为$+t$ 或$-t$.$(\pm A)^*$是符号项的有限序列集合,记$(\pm A)^*$中的元素为$\langle\langle\sigma_1,a_1\rangle,\ldots,\langle\sigma_n,a_n\rangle\rangle$.

A 上的串空间是集合Σ以及迹映射 $tr:\Sigma\rightarrow(\pm A)^*$.

构造串空间的方法如下:

(1) 结点是二元组$\langle s,i\rangle$,其中 $s\in\Sigma$且 i 为满足 $1\leq i\leq length(tr(s))$的整数.结点集合记为 $\mathcal{N}$,称结点$\langle s,i\rangle$属于串 s.显然,每个结点属于惟一的串.

(2) 若 $n=\langle s,i\rangle\in\mathcal{N}$,则 $index(n)=i$,且 $strand(n)=s$.定义 $term(n)$ 为$(tr(s))_i$,即 s 的迹中的第 i 个符号项.定义 $uns\_term(n)=((tr(s))_i)_2$,即 s 的迹中的第 i 个符号项的无符号部分.

(3) 存在一个边 $n_1\rightarrow n_2$,当且仅当 $term(n_1)=+a$ 且 $term(n_2)=-a$,其中 $a\in A$.因此,这类边意味着结点 n_1 发送消息 a,结点 n_2 接收消息 a,记录了串间的一种因果连接.

(4) 若 $n_1=\langle s,i\rangle\in\mathcal{N},n_2=\langle s,i+1\rangle\in\mathcal{N}$,则存在边 $n_1\Rightarrow n_2$.这类边表示 n_1 是 n_2 在串 s 上的直接因果前驱.用 $n'\Rightarrow^+n$ 表示 n'是 n 在同一个串上的因果前驱(不一定是直接因果前驱).

(5) 一个无符号项 t 出现在 $n\in\mathcal{N}$,当且仅当 $t\sqsubset term(n)$.

(6) 令 I 为无符号项集合.结点 $n\in\mathcal{N}$ 是 I 的进入点,当且仅当 $term(n)=+t$,其中 $t\in I$,且对所有的 $n'\Rightarrow^+n$,$term(n')\notin I$.

(7) 无符号项 t 产生于 $n\in\mathcal{N}$,当且仅当 n 是集合 $I=\{t':t\sqsubset t'\}$的进入点.

(8) 无符号项 t 是惟一产生的,当且仅当 t 惟一产生于 $n\in\mathcal{N}$.

$\mathcal{N}$以及两类边 $n_1\rightarrow n_2$ 和 $n_1\Rightarrow n_2$ 的集合是一个有向图$\langle\mathcal{N},(\rightarrow\cup\Rightarrow)\rangle$.

丛是这个图的一个有限子图,它的边表示结点之间的因果依赖关系.

假设$\rightarrow_C\subset\rightarrow$;$\Rightarrow_C\subset\Rightarrow$.假设 $C=\langle\mathcal{N}_C,(\rightarrow_C\cup\Rightarrow_C)\rangle$是$\langle\mathcal{N},(\rightarrow\cup\Rightarrow)\rangle$的子图,$C$ 是丛当且仅当

(1) C 是有限的无环图;

(2) 若 $n_2\in\mathcal{N}_C$,且 $term(n_2)$为负,则存在惟一的 n_1,使得 $n_1\rightarrow_C n_2$;

(3) 若 $n_2\in\mathcal{N}_C$,且 $n_1\Rightarrow n_2$,则 $n_1\Rightarrow_C n_2$.

若 $n\in\mathcal{N}_C$,则称结点 n 在丛 $C=\langle\mathcal{N}_C,\rightarrow_C,\Rightarrow_C\rangle$中,记为 $n\in C$;若串 s 所有的结点均在 $\mathcal{N}_C$ 中,则称串 s 在丛 C 中.

若 C 为丛,则 s 串的 C 高度,记为 $C\text{-}hight(s)$,是满足$\langle s,i\rangle\in C$ 的最大的 i 值.$C\text{-}trace(s)=\langle(tr(s))_1,\ldots,(tr(s))_m\rangle$,其中 $m=C\text{-}hight(s)$.

丛 C 中结点间的关系形成偏序关系,用$\preccurlyeq_C$表示.丛 C 的任何非空结点子集,在偏序关系$\preccurlyeq_C$下都有最小元.

1.2 认证测试方法

称项 t_0 为项 t 的分量,若 $t_0\sqsubset t,t_0$ 不是级连项,且任何满足 $t_0\sqsubset t_1\sqsubset t$ 的 $t_1\neq t_0$ 是级连项.显然,分量或者是原子值或者是加密项.称项 t 在结点 $n=\langle s,i\rangle$是新项,如果 t 是项 $term(n)$的分量,但 t 不是结点$\langle s,j\rangle$的分量,其中 $j<i$.

在串空间Σ下,称正常串的某个部分为测试,它的存在将保证其他正常串在丛中的存在.称项 $t=\{h\}_k$ 为项 a

在结点 n 中的测试分量,如果

(1) $a \sqsubset t$ 且 t 是 n 的分量;

(2) t 不是任何正常结点 $n' \in \Sigma$的分量的真子项.

称边 $n_1 \Rightarrow^+ n_2$ 是对于 $a \in A$ 的被变换边(变换边),如果 n_1 为正且 n_2 为负(如果 n_1 为负且 n_2 为正),$a \sqsubset term(n_1)$,且存在一个 n_2 的新分量 t_2,使得 $a \sqsubset t_2$.

称边 $n_0 \Rightarrow^+ n_1$ 是对于 a 的测试,如果 a 惟一地产生在 n_0,且 $n_0 \Rightarrow^+ n_1$ 是对于 a 的被变换边.

共有 3 种重要的认证测试方法:出测试方法、入测试方法与测试方法.

称边 $n_0 \Rightarrow^+ n_1$ 是 a 在 $t=\{h\}_K$ 中的出测试,如果它是对于 a 的测试,且 $K^{-1} \notin \mathcal{K}$,这里 $\mathcal{K}$ 表示不安全密钥集合.除 t 外 a 不在 n_0 的任何分量中出现,且 t 是 a 在 n_0 中的测试分量.

称边 $n_0 \Rightarrow^+ n_1$ 是对于 a 在 $t_1=\{h\}_K$ 中的入测试,如果它是对于 a 的测试,且 $K \notin \mathcal{K}$,并且,t_1 是对于 a 在 n_1 中的测试分量.

称负结点 n 是对于 $t=\{h\}_K$ 的主动测试,如果 t 是对于 n 中任何 a 的测试分量,且 $K \notin \mathcal{K}$.

出测试原理:令 C 为丛,$n' \in C$,$n \Rightarrow^+ n'$ 是 a 在 t 中的出测试.于是,

(1) 存在正常结点 m,$m' \in C$,使得 t 是 m 的分量,且 $m \Rightarrow^+ m'$ 是对于 a 的变换边;

(2) 假设除此之外,a 只在 m' 的分量 $t_1 = \{h_1\}_{k_1}$ 中出现,且 t_1 不是任何正常分量的真子项,并且 $K_1^{-1} \notin \mathcal{K}$.于是,存在一个负正常结点 m'',其中 t_1 是 m'' 的分量.

入测试原理:令 C 为丛,$n' \in C$;令 $n \Rightarrow^+ n'$ 为对于 a 在 t' 中的入测试.于是,存在正常结点 m,$m' \in C$,使得 t' 是 m' 的分量,且 $m \Rightarrow^+ m'$ 是对于 a 的变换边.

主动测试原理:令 C 为丛,$n \in C$,n 是对于 $t=\{h\}_K$ 的主动测试.于是,存在一个正常结点 $m \in C$,使得 t 是 m 的分量.

2 CSP 方法简介

CSP 是 Hoare[9]提出来的一种用于描述和分析并行系统通信的进程代数,进程与事件是 CSP 中的重要概念.从通信的角度,进程通过它能执行的通信事件来定义.CSP 中定义了两种中止进程:Stop 与 Skip,其中 Stop 表示死锁,Skip 表示成功中止执行.例如,下述进程 P 先后执行事件 x 与 y 之后,处于死锁状态:

$$P_1 = x \rightarrow y \rightarrow Stop.$$

CSP 将事件名视为信道,可以规定它所传递的数据类型,例如,下述进程

$$P_2 = in?x:T \rightarrow out.x \rightarrow Stop$$

表示只要 x 为 T 类型数据,进程 P_2 就可以执行这个 $in.x$ 事件.上式中如果不指明数据类型,则表示只要 x 为信道 in 范围内定义的数据类型,进程 P_2 即可执行该 $in.x$ 事件.信道 out 也具有相应的数据类型 x,并且一旦执行了事件 $in.x$,并给变量 x 赋值 v,则它的后续事件 $out.v$ 就可以执行.在该进程中,out 信道的数值与 in 信道输入的数值相同,因此数据类型 T 必定是 x 的子集.下述进程

$$P_3 = in?x:T \rightarrow out.x \rightarrow P_3$$

是循环进程,由无限多个 in 和 out 事件组成.进程也可以通过参数方式表示,例如

$$P_4(X) = in?x:T \rightarrow out.x \rightarrow P_4(X \bigcup x).$$

以下是 CSP 中应用的几种重要的操作符.

□是外部选择操作符,例如:

$$P_5 = a \rightarrow P_5;\ P_6 = b \rightarrow P_6;\ P_7 = P_5 \,\square\, P_6.$$

P_7 为外部环境提供了两种选择,如果外部环境选择事件 a,则 $P_7=P_5$;如果外部环境选择事件 b,则 $P_7=P_6$.

∏是内部选择操作符,允许进程间进行非确定性的选择,亦即由进程内部进行选择.例如,

$$P_8=P_5 \prod P_6.$$

P_8 可以内部选择执行事件 a 或 b.如果外部环境希望执行事件 a,而 P_8 的内部选择是执行事件 b,则事件 a 将被阻塞.

$P \underset{X}{\|} Q$ 表示通常的并行操作,进程 P 和 Q 将在事件集合 X 的所有事件上同步,并且可以进一步表示为 $P[[X]]Q$.

$P_X\|_Y Q$ 表示基于字母表的同步,即进程 P 和 Q 在事件 $X \cap Y$ 上同步.

$P|||Q$ 表示穿插操作.进程 P 和 Q 并不在任何事件上同步,它们之间完全相互独立.

$P\backslash E$ 表示隐藏操作,其中 P 是进程,E 是在 P 的外部环境中被隐藏的事件集.例如,

$$P_9 = P_7 \backslash \{a\}.$$

如果 P_7 的可见事件为 $\{a,b\}$,则 P_9 的可见事件为 $\{b\}$.

$P\|R\|$ 表示 P 被重命名后的进程,除了根据重命名关系 R 被重新命名的事件以外,该进程与原进程相同.例如,

$$P_{10} = P_7 \left\| a \leftarrow c, b \leftarrow d \right\|.$$

P_{10} 除了将事件 a 和 b 分别替换为 c 和 d 之外,与 P_7 相同,从而与下述进程等价:

$$P_{11} = c \to P_{11} \,\square\, d \to P_{11}.$$

Lowe[10,11]用 CSP 方法和 FDR 模型检测工具发现了 Needham-Schroeder 公开密钥协议[12]的缺陷,提出了 Needham-Schroeder 协议的改进协议.同时,通过 CSP 方法和 FDR 证明了 Lowe 改进后的 Needham-Schroeder 协议是正确的.

3 Helsinki 协议

3.1 Helsinki协议简介

Helsinki 协议[13]是国际标准 ISO/IEC DIS 11770-3 中提出的安全协议草案.协议如下:

(1) $A \to B$: $\{AK_I N_a\}_{K_B}$;

(2) $B \to A$: $\{K_R N_a N_b\}_{K_A}$;

(3) $A \to B$: N_b.

这里,主体 A 是发起者,主体 B 是响应者. $\{h\}_{K_A}$ 表示应用主体 A 的公开密钥 K_A 加密数据 h;N_a 和 N_b 是临时值;K_I 和 K_R 分别是 A 和 B 产生的部分密钥.Helsinki 协议的目标是为主体 A 和 B 安全地分配一个共享会话密钥 K_{AB},它最终由 K_I,K_R 和单向函数 f 确定,即 $K_{AB}=f(K_I,K_R)$.

3.2 Horng-Hsu攻击

Horng 和 Hsu[14]指出,He1sinki 协议存在安全缺陷,不能达到它的安全目标.Horng-Hsu 攻击是一种内部攻击,即攻击者必须是合法的协议主体,而且其他合法主体希望与他通过协议分配共享通信密钥.Horng-Hsu 攻击的过程如下:

首先,攻击者 P 要求主体 A 发起一次 A 与 P 的协议运行:

$$A \to P: \{AK_I N_a\}_{K_P}.$$

攻击者 P 解密这条消息获得 N_a,冒充主体 A 发送下述伪造消息给主体 B:

$$P(A) \to B: \{AK_P N_a\}_{K_B}.$$

主体 B 误认为攻击者 P 是主体 A,根据协议发送下述响应消息给攻击者 P:

$$B \to P(A): \{K_R N_a N_b\}_{K_A}.$$

攻击者 P 截获这条消息,原封不动地将它转发给主体 A:

$$P \to A: \{K_R N_a N_b\}_{K_A}.$$

主体 A 根据协议发送 N_b 给攻击者 P:

$$A \to P: N_b.$$

最后,攻击者 P 冒充主体 A 转发 N_b 给主体 B:

$$P(A) \to B: N_b.$$

经过上述两回合协议运行之后,主体 A 相信,他成功地完成了一次与主体 P 的会话,并获得共享会话密钥

$K_{AP}=f(K_I,K_R)$,但是,主体 P 并不知道会话密钥 K_{AP} 的组成部分 K_R.同时,主体 B 相信,他与主体 A 成功地进行了一次会话,并获得共享会话密钥 $K_{PB}=f(K_P,K_R)$.但是,主体 A 并不知道会话密钥 K_{PB} 的组成部分 K_P.因此,攻击者 P 的攻击是有效的.

3.3 Mitchell-Yeun的改进协议

此后,Mitchell 和 Yeun[15]对 Helsinki 协议进行了改进.他们认为,Horng-Hsu 攻击成功的原因是,A 相信 B 发送的第 2 条消息 $\{K_RN_aN_b\}_{K_A}$ 来源于 P.这是因为消息 2 没有明确指出消息来源,从而造成了消息的重放.因此,攻击者 P 虽然并不知道消息 2 的真实内容,但 P 可以将消息 2 转发给 A,使 A 相信消息 2 来源于 P.

基于上述理由,Mitchell-Yeun 的改进协议如下:

(1) $A\rightarrow B$: $\{AK_IN_a\}_{K_B}$;

(2) $B\rightarrow A$: $\{BK_RN_aN_b\}_{K_A}$;

(3) $A\rightarrow B$:N_b.

亦即,在消息 2 中增加了主体 B 的标识符.

4 Helsinki 协议的形式化分析

Horng 和 Hsu 对 He1sinki 协议进行了成功的攻击,Mitchell 和 Yeun 提出了 Helsinki 协议的改进协议,但是,他们均未对 He1sinki 协议及其改进版本进行形式化分析和证明.本文通过串空间模型与 CSP 方法严格证明了 Helsinki 协议改进版本的正确性,并揭示原 Helsinki 协议的安全缺陷.通过两种流行的形式化分析方法的比较,说明应用串空间模型和 CSP 方法分析的不同特点.

4.1 应用串空间模型方法进行分析

我们首先证明改进后的 Helsinki 协议的正确性.Helsinki 改进协议的串空间构造如下:

发起者串 $s_i\in Init\ [A,B,N_a,N_b,K_I,K_R]$,其迹具有下述形状:

$$\langle +\{AK_IN_a\}_{K_B},-\{BK_RN_aN_b\}_{K_A},+N_b\rangle.$$

响应者串 $s_r\in Resp\ [A,B,N_a,N_b,K_I,K_R]$,其迹具有下述形状:

$$\langle -\{AK_IN_a\}_{K_B},+\{BK_RN_aN_b\}_{K_A},-N_b\rangle.$$

我们进行如下基本假设:(1) K_A^{-1}, $K_B^{-1}\notin\mathcal{K}$;(2) N_a 和 N_b 惟一产生;(3) $N_a\neq N_b$.

第 1 步证明响应者成功地认证发起者.

设 C 为丛, $s_r\in Resp\ [A,B,N_a,N_b,K_I,K_R]$,且 C-$hight(s_r)$=3.由于 $K_A^{-1}\notin\mathcal{K}$,$N_b$ 惟一产生在$\langle s_r,2\rangle$,边$\langle s_r,2\rangle\Rightarrow^+\langle s_r,3\rangle$是 N_b 在 $\{BK_RN_aN_b\}_{K_A}$ 中的出测试.根据出测试的原理(1),存在正常结点 $m,m'\in C$,使得 $\{BK_RN_aN_b\}_{K_A}$ 是 m 的分量,且 $m\Rightarrow^+m'$是 N_b 的变换边.这时,m 只可能为$\langle s_i,2\rangle$,其中 $s_i\in Init\ [A,B,N_a,N_b,*,K_R]$.于是,变换边 $m\Rightarrow^+m'$必为$\langle s_i,2\rangle\Rightarrow^+\langle s_i,3\rangle$,且 C-$hight(s_i)$=3.因此我们证明了,在 Helsinki 改进协议中响应者成功地认证了发起者.

第 2 步证明发起者成功地认证响应者.

设 C 为丛, $s_i\in Resp\ [A,B,N_a,N_b,K_I,K_R]$,且 C-$hight(s_i)$=3.由于 $K_B^{-1}\notin\mathcal{K}$, N_a 惟一产生在$\langle s_i,1\rangle$,边$\langle s_i,1\rangle\Rightarrow^+\langle s_i,2\rangle$是 N_a 在 $\{AK_IN_a\}_{K_B}$ 中的出测试.根据出测试的原理(1),存在正常结点 $m,m'\in C$,使得 $\{AK_IN_a\}_{K_B}$ 是 m 的分量,且 $m\Rightarrow^+m'$是 N_a的变换边.这时,m 只可能为$\langle s_r,1\rangle$,其中 $s_r\in Resp\ [A,B,N_a,N_b',K_I,*]$.

基于同样理由,边 $\langle s_r',2\rangle\Rightarrow^+\langle s_r',3\rangle$ 是 N_b' 在 $\{BK_RN_aN_b'\}_{K_A}$ 中的出测试.根据出测试的原理(2), $\{BK_RN_aN_b'\}_{K_A}$ 是某个负正常结点 m''的分量.但是,m''只能是 $\langle s_i',2\rangle$,其中 $s_i'\in Resp\ [A,B,N_a,N_b',*,K_R]$,因为只有发起者串的第 2 个结点才收到这种形式的分量.由发起者串的形状,N_a 产生于 $\langle s_i',1\rangle$.因为 N_a 是惟一产生的,故 $\langle s_i',1\rangle=\langle s_i,1\rangle$,因此 $s_i'=s_i$ 且 $N_b=N_b'$.于是,C 包含响应者串 $s_r\in Resp\ [A,B,N_a,N_b,K_I,*]$,且 C-$hight(s_r)$=2.因此我们证明了,在 Helsinki 改进协议中发起者成功地认证了响应者.

对于原 Helsinki 协议,由于第 2 条消息没有指出消息的来源,因此我们在证明响应者认证发起者时,只能得到如下结论:设 C 为丛, $s_r\in Resp\ [A,B,N_a,N_b,K_I,K_R]$,且 C-$hight(s_r)$=3.于是,存在变换边$\langle s_i,2\rangle\Rightarrow^+\langle s_i,3\rangle$,且

$C\text{-}hight(s_i)$=3,其中 $s_i \in Init\ [A,B',N_a,N_b,*,K_R]$.亦即,我们没有对期待的响应者 B 证明 $s_i \in Init[A,B,N_a,N_b,*,K_R]$,而是对某个其他的响应者 B'证明了这一点.Horng-Hsu 的攻击就是一个反例,其中 $B'=P\neq B$.因此,我们再次揭示了 Horng-Hsu 首先发现的原 Helsinki 协议的安全缺陷.

4.2 应用CSP方法进行分析

下面,我们用 CSP 方法分析 Helsinki 协议,并纠正文献[16]中的若干错误.第 1 步,我们需要构造协议的有限状态系统模型.参与协议的主体是发起者 A,响应者 B 和攻击者 P.有关集合包括:(1) 发起者集合 Initiator:$\{A,P\}$;(2) 响应者集合 Responder:$\{B,P\}$;(3) 临时密钥集合 Key1:$\{K_I,K_R,K\}$;(4) 公开密钥集合 Key2:$\{K_A,K_B,K_P\}$;(5) 临时值集合 Nonce:$\{N_A,N_B,N_P\}$.

其次,定义系统通信的信道:(1) 信道 comm:表示主体正常通信的信道;(2) 信道 fake:表示攻击者伪造消息的信道;(3) 信道 intercept:表示攻击者窃听消息的信道;(4) 信道 user:表示用户会话请求的外部接口信道;(5) 信道 session:表示用户会话的外部接口信道;(6) 主体状态信道,其中,I_running.a.b 表示发起者相信他正在进行与响应者的会话;R_running.a.b 表示响应者相信他正在进行与发起者的会话;I_commit.a.b 表示发起者相信他已经完成与响应者的会话,开始进行数据传输;R_commit.a.b 表示响应者相信他已经完成与发起者的会话,开始进行数据传输.

然后,通过 MSG1,MSG2 和 MSG3 表示 Helsinki 协议中的 3 条消息:

$$MSG1 \cong \{Msg1.A.B.Encrypt.K.A.k.N \mid A\in Initiator, B\in Responder, k\in Key1, K\in Key2, N\in Nonce\};$$

$$MSG2 \cong \{Msg1.B.A.Encrypt.K.k.N_a.N_b \mid A\in Initiator, B\in Responder, k\in Key1, K\in Key2, N_a\in Nonce, N_b\in Nonce\};$$

$$MSG3 \cong \{Msg1.A.B.N \mid A\in Initiator, B\in Responder, N\in Nonce\};$$

$$MSG \cong MSG1\cup MSG2\cup MSG3.$$

发起者的 CSP 进程 $INITIATOR(A,K_I,N_a)$ 进行如下规范:

$$INITIATOR(A,K_I,N_a) \cong user.A?B \rightarrow I\_running.A.B \rightarrow comm.Msg1.A.B.Encrypt.K_B.A.K_I.N_a) \rightarrow$$

$$\square_{\substack{K_R\in Key1\\ K_A\in Key2\\ N_a,N_b\in Nonce}} \left[\begin{array}{l} comm.Msg2.B.A.Encrypt.K_A.K_R.N'_a.N_b \rightarrow \\ \text{if}\ \ N'_a=N_a\ \ \text{then} \\ \quad comm.Msg3.A.B.N_b \rightarrow \\ \quad I\_commit.A.B \rightarrow session.A.B \rightarrow Skip \\ \text{else} \\ \quad \text{Stop} \end{array}\right].$$

响应者 B 的 CSP 进程 $RESPONDER(B,K_R,N_b)$ 进行如下规范:

$$RESPONDER(B,K_R,N_b) \cong R\_running.A.B \rightarrow$$

$$\square_{\substack{K_I,K_R\in Key1\\ K_A,K_B\in Key2\\ N_a,N_b\in Nonce}} \left[\begin{array}{l} comm.Msg1.A.B.Encrypt.K_B.A.K_I.N_a \rightarrow \\ comm.Msg2.B.A.Encrypt.K_A.K_R.N_a.N_b \rightarrow \\ comm.Msg3.A.B.EncryptK_B.N_a.N'_b \rightarrow \\ \text{if}\ \ N'_b=N_b\ \ \text{then} \\ \quad R\_commit.A.B. \rightarrow session.A.B \rightarrow Skip \\ \text{else} \\ \quad \text{Stop} \end{array}\right]$$

为了模拟攻击者的行为,需要通过 CSP 的重命名机制对上述进程重新定义.亦即,应当允许发起者 A 的消息 1 和消息 3 被窃听,消息 2 被伪造;允许响应者 B 的消息 1 和消息 3 被伪造,消息 2 被窃听.

假设攻击者 P 是协议的合法用户,我们可以通过他的知识和能力参数化 P 的状态.亦即,通过攻击者 P 不能解密的消息 1、消息 2、消息 3 的集合 Ml,M2,M3;P 知道的临时值的集合 ns 和临时密钥集合 ks 参数化 P 的状态.

$$P(M1,M2,M3,ns,ks) \cong$$

$$\square$$

$$A \in Initiator$$
$$B \in Responder$$
$$k,K_I,K_R \in Key1$$
$$N_a,N_b,N,N' \in Nonce$$
$$K \in Key2$$

$$\left[\begin{array}{l}
\square\ \text{int}ercept.Msg1.A.B.Encypt.K.A.K_I.N_a \rightarrow \\
\quad \text{if } K = K_P \text{ then } I(M1,M2,M3,ns \cup \{N_a\},ks \cup \{K_I\}) \\
\quad \text{else } P(M1 \cup \{Encypt.K.A.K_I.N_a\},M2,M3,ns,ks) \\
\square\ \text{int}ercept.Msg2.B.A.Encypt.K.K_R.N_a.N_b \rightarrow \\
\quad \text{if } K = K_P \text{ then } P(M1,M2,M3,ns \cup \{N_a,N_b\},ks \cup \{K_R\}) \\
\quad \text{else } P(M1,M2 \cup \{Encypt.K.K_R.N_a.N_b\},M3,ns,ks) \\
\square\ \text{int}ercept.Msg3?A.B.N_b \rightarrow P(M1,M2,M3,ns \cup \{N_b\},ks) \\
\square\ fake.Msg1.A.B.M : M1 \rightarrow P(M1,M2,M3,ns,ks) \\
\square\ fake.Msg2.A.B.M : M2 \rightarrow P(M1,M2,M3,ns,ks) \\
\square\ fake.Msg3.A.B.M : M3 \rightarrow P(M1,M2,M3,ns,ks) \\
\square\ fake.Msg1.A.B.Encrypt.K.A.k : ks.N : ns \rightarrow P(M1,M2,M3,ns,ks) \\
\square\ fake.Msg2.A.B.Encrypt.K.k : ks.N : ns.N' : ns \rightarrow P(M1,M2,M3,ns,ks) \\
\square\ fake.Msg3.A.B.N : ns \rightarrow P(M1,M2,M3,ns,ks)
\end{array}\right]$$

将上述协议规范和实现进程输入 FDR 后,得到的迹如下:

$\langle user.A.P, I\_running.A.P, Comm.Msg1.A.P.Encypt.K_P.A.K_I.N_a,$
$fake.Msg1.A.B.Encrypt.K_B.A.K_P.N_a, \text{int}ercept.Msg2.B.A.Encrypt.K_A.K_R.N_a.N_b,$
$comm.Msg2.P.A.Encrypt.K_A.K_R.N_a.N_b, Comm.Msg3.A.P.N_b,$
$fake.Msg3.A.B.N_b, R\_Commit.A.B\rangle$

由此可见,上述结果与 Horng-hsu 攻击产生的协议迹相同.事件 $R\_Commit.A.B$ 存在,但没有相应的 $I\_running.A.B$ 事件.

4.3 两种形式化分析方法的特点与异同

通过对一个具体例子的分析,使我们对两种形式化分析方法的理解更加深入.串空间模型是一种结合定理证明和协议迹的混合分析方法.串是参与协议的主体可以执行的事件序列.对于诚实的主体,该事件序列是根据协议定义,由发送事件和接收事件组合而成的.此外,该模型还定义了攻击者串,描述攻击者的行为.串空间是由协议参与者,包括诚实主体和攻击者的串组成的串集合.串集合之间可以穿插组合,使一个串的发送消息对应于另一个串的接收消息.串空间模型的核心是它的丛结构.丛是一个良构(well-defined)集合,表示一个完整的协议交换串空间的子集.丛可以表示为有限无环图,其中的边表示结点间的因果依赖关系.根据串空间的基本模型,容易对其进行不同方向的扩展、增强和优化.例如,将串空间模型推广到分析三方认证协议.应用串空间模型,开发认证协议的自动检验工具.检验工具的证明从一些初始状态开始,进行回退搜索.初始状态是满足某种安全属性的丛.认证测试方法,是根据串空间模型形式化分析认证协议的一种重要工具.它的基本原理是,在串空间Σ下定义正常串的某个部分为测试,它的存在将保证其他正常串在丛中的存在.这些串上的变换边对测试分量进行运算.认证测试方法的应用十分简捷与直观.通过有向图的辅助,使得分析过程与分析结果十分清晰.

CSP 方法也是一种基于协议迹的形式化分析方法.CSP 方法将协议主体说明为 CSP 中的进程,将消息描述为事件,将协议说明为通信顺序进程的集合.这些进程并行运行,并且与它们的环境交互作用.对认证协议进行分析,相当于根据协议说明构造一个系统模型,通常是建立一个有限状态迁移系统.在应用 CSP 方法进行分析时,首先必须将协议转变为 CSP 的规范;其次,必须根据 CSP 方法规范攻击者的行为;最后,需要通过自动检测工具,例如,FDR 进行校验.FDR 接受两个 CSP 进程作为输入,一个是协议规范,即形式化地表示迹之间的关系;另一个是协议实现,然后检查协议的每一个迹是否为协议规范的每一个迹.此外,由于 FDR 通过穷尽状态空间方式工

作,因此被验证系统的状态空间必须是有限的.在 CSP 的框架下,也容易扩展 CSP 的分析范围,使其不仅能够分析双方认证协议,还可以分析三方协议、分析非否认协议等.

总之,两种方法都是严格的基于协议迹的形式化分析方法,能够分析并找出认证协议中微妙的安全缺陷.在它们各自的结构下,都易于进行功能的扩展、增强与优化.但是,它们又各自具有不同的特点.首先,串空间模型更为简洁与直观,特别是通过有向图的辅助使得分析过程十分清晰;CSP 方法则比较抽象.其次,CSP 方法易于进行自动分析,避免过多的人工参与.但是,CSP 方法不易于对简单协议进行人工分析.因此,CSP+FDR 被认为是典型的作法,自动化分析程度高.相反地,针对串空间模型的自动分析工具不如 FDR 等模型检测工具成熟.但是从另一个角度来看,串空间模型适于对认证协议进行手工分析.再有,CSP 方法的另一个优点是,如果协议存在安全缺陷,在分析结果中会生成攻击者迹,明确指出了攻击者的攻击路径,便于对协议进行改进.在揭示攻击者迹方面,串空间模型不如 CSP 方法更为直接.

此外,两种方法还具有一些共同点.第 1,关注协议本身的结构,假设协议所采用的密码算法是“完善”的.亦即,除非获得正确的解密密钥,无法通过密文还原为明文.此外,还作了一些其他假设.例如,密文块不能被篡改,也不能用几个小的密文块组成一个新的大密文块;一条消息中的两个密文块被视为分两次分别到达;无加密项冲突,即,若有

$$\{m_1\}_{K_1}=\{m_2\}_{K_2},$$

则一定有 $m_1=m_2, K_1=K_2$;参与协议的主体包含诚实的合法用户和攻击者.合法用户将遵循协议的规定执行协议.攻击者也可以是系统的合法用户,但他不会完全遵守协议.第 2,服从 Dolev-Yao 模型[17].该模型认为,攻击者可以控制整个通信网络,并应当假定攻击者具有相应的知识与能力.例如,应当假定,攻击者不但可以窃听、阻止、截获所有经过网络的消息,而且还具备以下知识和能力:(1) 熟悉加解、解密、散列(hash)等密码运算,拥有自己的加密密钥和解密密钥;(2) 熟悉参与协议的主体标识符及其公钥;(3) 具有密码分析的知识和能力;(4) 具有进行各种攻击,例如重放攻击的知识和能力.第 3,基于协议迹进行形式化分析.关于安全协议及其分析方法更多的讨论,请参考文献[18~20].

5 结 论

串空间模型和 CSP 方法都是基于协议迹的形式化分析方法.它们严格地规范了合法主体的行为、攻击者的能力与运行环境,是近年来出现的分析认证协议的有效方法.本文通过对两种方法的分析和比较,使我们对两种方法的内在特点有了更为深刻的认识.我们可以在不同情形下择优选用不同的分析方法,也可以对两种方法的分析结果进行比较与验证,使两种方法互为补充.

References:

[1] Qing SH. Cryptography and Computer Network Security. Beijing: Qinghua University Press, 2001. 127~147 (in Chinese).

[2] Qing SH. Formal analysis of authentication protocols. Journal of Software, 1996,7:107~114 (in Chinese with English abstract).

[3] Thayer FJ, Herzog JC, Guttman JD. Strand spaces: Why is a security protocol correct? In: Proceedings of the 1998 IEEE Symposium on Security and Privacy. Los Alamitos: IEEE Computer Society Press, 1998. 160~171.

[4] Thayer FJ, Herzog JC, Guttman JD. Strand spaces: Proving security protocols correct. Journal of Computer Security, 1999, 7(2,3):191~230.

[5] Thayer FJ, Herzog JC, Guttman JD. Strand spaces: Honest ideals on strand spaces. In: Proceedings of the 1998 IEEE Computer Security Foundations Workshop. Los Alamitos: IEEE Computer Society Press, 1998. 66~77.

[6] Guttman JD, Thayer FJ. Authentication tests. In: Proceedings of the 2000 IEEE Symposium on Security and Privacy. Los Alamitos: IEEE Computer Society Press, 2000. 150~164.

[7] Schneider SA. Security properties and CSP. In: Proceedings of the 1996 IEEE Symposium on Security and Privacy. Los Alamitos: IEEE Computer Society Press, 1996. 174~187.

[8] Schneider S, Sidiropoulos A. CSP and anonymity. In: Proceedings of the Computer Security-ES-ORICS'96. LNCS 950, Berlin: Springer-Verlag, 1996. 198~218.

[9] Hoare CAR. Communicating Sequential Processes. Prentice Hall, 1985.

[10] Lowe G. An attack on the Needham-Schroeder public-key authentication protocol. Information Processing Letters, 1995,56: 131~133.

[11] Lowe G. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. Software-Concepts and Tools, 1996,17: 93~102.

[12] Needham R, Schroeder M. Using encryption for authentication in large networks of computers. Communications of the ACM, 1978, 21(12):993~999.

[13] ISO/IEC. 2nd DIS 11770-3, Information technology—Security techniques-key management; part 3; Mechanisms using asymmetric techniques. 1997.

[14] Horng G, Hsu CK. Weakness in the Helsinki protocol. Electronic Letters, 1998,34:354~355.

[15] Mitchell CJ, Yeun CY. Fixing a problem in the Helsinki protocol. Operating Systems Review, 1998,32:21~24.

[16] Qin C. Study on analysis of cryptographic protocols based on formal methods [Ph.D. Thesis]. Beijing: Peking University, 2002 (in Chinese with English abstract).

[17] Dolev D, Yao A. On the security of public key protocols. IEEE Transactions on Information Theory, 1983,29(2):198~208.

[18] Qing, SH. Design and logical analysis of security protocols. Journal of Software, 2003,14(7):1300~1309 (in Chinese with English abstract).

[19] Qing, SH. 20 Years Development of Security Protocols Research. Journal of Software, 2003,14(10):1740~1752 (in Chinese with English abstract).

[20] Qing, SH. The TTP Roles in Electronic Commerce Protocols. Journal of Software, 2003,14(11):1936~1943 (in Chinese with English abstract).

附中文参考文献:

[1] 卿斯汉.密码学与计算机网络安全.北京:清华大学出版社,2000.127~147.

[2] 卿斯汉.认证协议的形式化分析.软件学报,1996,7:107~114.

[16] 秦超.基于形式化方法的密码协议分析研究[博士学位论文].北京:北京大学,2002.

[18] 卿斯汉.安全协议的设计与逻辑分析.软件学报,2003,14(7):1300~1309.

[19] 卿斯汉.安全协议 20 年研究进展.软件学报,2003,14(10):1740~1752.

[20] 卿斯汉.电子商务协议中的可信第三方角色.软件学报,2003,14(11):1936~1943.