# 协议认证性安全属性测试方法

何云华[1,2],杨 超[1],张俊伟[1],马建峰[1]
(1. 西安电子科技大学计算机网络与信息安全教育部重点实验室,陕西西安 710071;
2. 北方工业大学计算机学院信息安全系,北京 100029)

**摘 要:** 认证性建立通信双方的信任关系,是安全通信的重要保障. 传统的协议测试方法只关注协议功能的正确性,无法满足认证性等安全属性测试的要求. 因此,提出了一种针对协议认证性的安全属性测试方法,利用带目标集合的有限状态机模型 SPG-EFSM 来扩展描述协议安全属性,并在攻击场景分类的基础上设计了认证攻击算法. 通过攻击算法找到了 Woo-lam 协议和 μTESLA 协议的认证性漏洞,该方法具有可行性、覆盖率高等特点.

**关键词:** 协议测试; 安全属性; 认证性测试; 形式化模型; 攻击分类
**中图分类号:** TP309 **文献标识码:** A **文章编号:** 0372-2112 (2016)11-2788-08
**电子学报 URL:** http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2016.11.031

# Authentication Testing of Security Protocols—A Method for Testing Protocol Security Properties

HE Yun-hua[1,2], YANG Chao[1], ZHANG Jun-wei[1], MA Jian-feng[1]
(1. *Key Laboratory of Computer Network and Information Security of Ministry of Education, Xidian University, Xi'an, Shaanxi* 710071, *China*;
2. *Department of Information Security, College of Computer Science, North China University of Technology, Beijing* 100029, *China*)

**Abstract:** Authentication builds the trust relationship between communication parties, which is a magnitude guarantee for secure communications. However, existing protocol testing techniques focus on validating the protocol specification. Those techniques can not satisfy the requirements of testing protocol authentication as their lack of the method for describing security properties. Therefore, a protocol security property testing method is proposed for testing protocol authentication. This testing method uses a new formal model-Symbolic Parameterized Goal Extended Finite State Machine (SPG-EFSM) for describing protocols and their security properties. Then, a protocol attack algorithm is designed for testing protocol authentication based on different attack scenarios. Through test experiments on the well-known protocol Woo-lam and μTESLA, it is found that the SPG-EFSM based attack algorithm can find several protocol security flaws and has better feasibility and high coverage.

**Key words:** protocol testing; security properties; authentication testing; formal model; attack classification

## 1 引言

通信协议作为网络和分布式应用的基础[1],其固有的复杂性和潜在的敌对环境,使得协议的安全性面临巨大的挑战. 大量基于逻辑推理的方法被用来分析、验证通信协议的安全性[2,3],如认证性、保密性、不可否认性和完整性等安全属性. 但是,这些方法大多集中在协议规范分析与验证方面,忽略了另一个不可信、不安全因素——协议实现引入的漏洞[4].

最常见的协议实现安全漏洞测试方法是随机测试[5](Random testing),通过注入大量随机、异常或错误的输入到待测协议中来观察系统是否会出现异常,但其测试数据的生成缺乏准确、高效的指导,测试数据的分布不受控制[6]. 针对该问题,描述协议数据流的测试方法逐步受到了关注,使用上下文无关文法(BNF)[7]或结构化的 Frame 语法[8]描述消息字段之间的相关性. 但是,此类方法不支持有状态转移的协议控制流的建模. 数据流与控制流相结合的测试方法成为了进一步的研

收稿日期:2015-10-27;修回日期:2016-05-19;责任编辑:李勇锋
基金项目:国家自然科学基金青年基金(No. 61303219);陕西省自然科学基础研究计划(No. 2014JQ8295);中央高校基本科研业务费(No. JB140303);国家自然科学基金面上基金(No. 61672415)

究热点. Jing C 等人[9]对 TTCN-3 控制流描述模型进行了语法和语义扩展,但所描述的语法变异类型有限. Yamaguchi 等人[10]将抽象语法树(AST)和控制流图(CFG)相结合,在实际测试中也获得了较好的效果. 但为了满足特定的安全需求,例如认证性或机密性,通信协议往往对协议消息本身进行了加密等处理,存在大量不可知参数,这使得上述测试方法,很难识别消息的各个字段,构造出合适的测试例. 因此,设计专用的协议安全属性测试方法是迫切需求的.

但是,此类研究才刚刚起步. Mashtizadeh 等人[11]提出密码-控制流模型,该模型假定攻击者能产生消息认证码用于控制流测试. 该方法侧重于控制协议状态的跳转和返回,未考虑协议消息构造方法. Shu G 等人[12]提出了以协议消息作为其输入/输出参数的协议描述模型,该模型利用密钥 $K$、随机数 $N$ 等参数组成的元组来表示消息,并建立了用于表示攻击者行为及协议安全属性的攻击者模型. 该方法对加密消息具有一定处理能力,但其攻击者模型不能准确的表示协议某些特定的安全属性. 例如协议认证性,必须要结合参与者初始认证目标与协议执行完后状态才能准确的描述. 另外,该方法没有考虑某些特殊情形,包括一次会话有多个参与者、一个参与者参与多个会话、拥有合法身份的攻击者参与会话等情形.

针对以上问题,本文提出了协议认证性安全属性测试方法. 首先提出一种描述协议安全属性的有限状态机扩展模型 SPG-EFSM(Symbolic Parameterized Goal-Extended Finite State Machine),扩展定义参与方目标集,用条件判断函数来对比目标集合与协议执行后的结果,以实现安全属性的验证;在此基础上,对协议攻击场景进行了分类,该分类综合考虑了拥有合法身份的攻击者参与,存在认证中心的多方参与,参与者参与多个会话等情况;然后,结合了 Dolev-Yao 攻击模型[13],设计了协议认证性攻击算法,该算法包涵了上述所有的分类;最后,通过对广泛应用的 Woo-lam 协议[14]和 μTESLA 协议[15]进行了测试,发现存在于 Woo-lam 协议及其各种更新版本中的已知和未知安全漏洞,以及 μTESLA 协议的认证漂移问题.

## 2　SPG-EFSM 模型

SPG-EFSM 是一种包含安全属性定义和验证的协议描述模型,是 Shu G 提出的 EFSM 扩展模型[12]的改进. SPG-EFSM 扩展描述了协议目标集合,基于目标集合定义安全属性,通过对比目标集合与协议执行后的结果来指示协议的安全属性是否存在漏洞,该验证规则由条件判断函数来实现. 认证性是协议安全属性的重要部分,本文着重对认证性安全属性测试.

定义 1、定义 2 给出了认证目标集、单向认证和双向认证的概念;SPG-EFSM 模型的描述由定义 3 给出.

**定义 1**　认证集一表示为 $C_1=\{\rightarrow m \mid m \in P\}$,$\rightarrow m$ 代表期望认证参与者 $m$ 身份标识,$P$ 为参与者集合;认证集二表示为 $C_2=\{m\rightarrow n \mid m \in P \cap n \in P\}$,$m\rightarrow n$ 代表参与者 $n$ 期望向参与者 $m$ 认证自己身份标识;参与者 $n$ 目标集合表示为:

$$g_n=\{c \mid ((\exists c=\rightarrow m)\in C_1 \cap m\neq n)\cup ((\exists c=m\rightarrow n')\in C_2 \cap n=n');m,n,n'\in P\} \quad (1)$$

认证结果集定义为 $R=\{Succeed,Failed\}$.

**定义 2**　已知 $m,n,m',n'\in P,\rightarrow n'\in g_m,m'\rightarrow n\in g_n$.

(1)如果有$(m=m')\cap(n=n')$,并且协议能正确执行完成,则称协议达到了 $m$ 认证了 $n$,记 $m\Rightarrow n$.

(2)如果有$(m\Rightarrow n)\cap(n\Rightarrow m)$,则称 $m$ 与 $n$ 达到了双向认证,记 $m\Leftrightarrow n$.

**定义 3**　SPG-EFSM 由七元组组成$\langle S,A,G,I,O,X,T\rangle$.

其中 $S$——状态集合;

$A$——原子集合$\{Key,Nonce,Int\}$和相应派生规则$\{E(),H(),MAC(),\cdot\}$,用于描述消息,如 $E(k_T,k_{ab}\cdot H(n_a))$;

$G$——目标集合 $G=\{g_n \mid n\in P\}$;

$I$——输入集合 $I=I'\cup G$,其中 $I'$是原有状态机输入集合;

$O$——输出集合 $O=O'\cup R$,其中 $O'$是原有状态机输出集合;

$X$——$L(A)$参数构成的有限集合,具有默认初值,其中 $L(A)$代表由 $A$ 构成的消息集合;

$T$——转移过程集合,$t=\langle s,s',i,o,p(x,\pi(i)),a(x,\pi(i),\pi(o))\rangle\in T$,其中 $s,s'$分别代表初态、终态,$\pi(i),\pi(o)$分别代表输入、输出的参数,$p(x,\pi(i))$是增加了定义 2 给出认证关系的条件判断函数,$a(x,\pi(i),\pi(o))$是关于变量、输入参数和输出参数的处理过程.

其中,协议的目标集合用于协议的安全属性需求,这里重点考虑认证性. 协议认证是通过一个转移过程 $t=\langle s,s',i,o,p(x,\pi(i)),a(x,\pi(i),\pi(o))\rangle$来验证的,$s,s'$分别是未认证态、认证态,输入 $i$ 为 $G$,输出 $o$ 为 $R$,条件判断函数 $p(x,\pi(i))$是定义 2 给出了的认证关系. 当 $p(x,\pi(i))=0$ 时,未通过认证,$o=Failed$. $a(x,\pi(i),\pi(o))$表示对消息的处理过程,如对加密消息进行解密、取消息中的某个元素.

为了确保测试有效地实施,协议测试要求状态机模型是确定的可达图[12]. 定理 1 证明了 SPG-EFSM 模型是一个确定的可达图.

**定理 1** SPG-EFSM 是确定性的 FSM,并且是一个可达图.

**证明** SPG-EFSM 可表示为 FSM($S^G, I^G, O^R, f_{\text{next}}, f_{\text{output}}$)的形式.

其中状态集合:

$$S^G = \{ < s, V, g_m > | (s \in S) \cap (V \subseteq X) \cap (g_m \in G) \} \tag{2}$$

输入/输出集合:

$$I^G = G \cup L(A), O^R = R \cup L(A) \tag{3}$$

状态转移函数:

$$\begin{aligned} f_{\text{next}}: & S^G \times I^G \to S^G, f_{\text{next}}( < s, V, g_m > , i) = \\ & \{ < s', V', g'_m > | \exists t \in T, \\ & t \ takes \ machine \ from \ < s, V, g_m > \\ & to \ < s', V', g'_m > \ upon \ i \} \end{aligned} \tag{4}$$

输出函数:

$$\begin{aligned} f_{\text{output}}: & S^G \times I^G \to O^R, f_{\text{output}}( < s, V, g_m > , i) \\ & = \{ o \in O^R | \exists t \in T, \\ & t \ outputs \ O^R \ from < s, V, g_m > upon \ i \} \end{aligned} \tag{5}$$

当 $< s, V, g_m > \in S^G$ 时,如果 $i \in I^G$,条件判断函数 $predict(x, \pi(i)) = 1$ 时,那么 $\exists t \in T$ 使得状态机 $M_i$ 从 $< s, V, g_m >$ 转移到 $< s', V', g'_m >$,并且输出 $o \in O^R$. 所以 SPG-EFSM 用于描述安全协议是一个确定性的有限状态机 FSM.

SPG-EFSM 的可达性问题可以转化为 FSM 的可达性问题. 对于安全协议而言,其状态 $S$ 是有限的,存储的变量 $X$ 也是有限的,可以构成具有有限结点和边的有向图. 假设协议的初始状态 $< s_0, V_0, g_{m0} >$,对于 $\forall < s, V, g_m > \in S^G$, $\exists seq_1 = \{ i_1 i_2 \cdots i_n \}$ 使得 $< s_0, V_0, g_{m0} >$ 达到 $< s, V, g_m >$,并且 $\exists seq_2 = \{ i'_1 i'_2 \cdots i'_n \}$ 使得 $< s, V, g_m >$ 达到 $< s_0, V_0, g_{m0} >$,其中 $seq_1, seq_2$ 是由一系列的输入 $i$ 组成的串,那么对 $\forall < s_j, V_j, g_{mj} >, < s_k, V_k, g_{mk} >$, $\exists seq_3 = \{ i'_{j+1}, \cdots, i'_n, i_1, \cdots, i_k \}$ 使得 $< s_j, V_j, g_{mj} >$ 达到 $< s_k, V_k, g_{mk} >$. 所以,协议中任意两个状态是可达的,即 SPG-EFSM 是一个可达图.

## 3 协议认证攻击算法

### 3.1 攻击场景分类

为了有效地实施攻击,本节对攻击场景进行了分类. 一方面,攻击场景分类可以快速的建立攻击策略,另一方面,攻击分类建立在 Dolev-Yao 强攻击模型[13]基础上,能够最大化攻击者的能力. 本节从是否存在多个参与者、是否利用接收者预言机服务[16]、是否拥有合法身份的攻击者等三个方面对攻击场景进行分类,具体定义如下:

本文用($L\_ID, Depend, T$)表示攻击情形,其中 $L\_ID$ 为攻击者的合法身份,$L\_ID = 1$($L\_ID = 0$)为拥有(无)合法身份;$Depend$ 为接收者的预言机服务,$Depend = 1$($Depend = 0$)为利用(不利用)接收者预言机服务;$T$ 为认证中心,$T = 1$($T = 0$)为存在(不存在)认证中心.

**定义 4** 设 $A, B$ 为参与者,$U$ 为用户集,$LU$ 为合法用户集,$M$ 为攻击者,$Attack$ 为 $M$ 进行的一次攻击,则 $Attack$ 可以分为以下八类:

①当($L\_CD, Depend, T$) = (0,0,0)时,存在 $Attack1$,即:

若 $M \in U - LU, A, B \in LU, (B \to A \in g_A) \cap (\to A \notin g_B) \cap (\to M("A") \in g_M)$,不存在 $T$,协议执行完毕,使得 $A \Rightarrow M("B")$ 或 $A \Rightarrow B$.

②当($L\_CD, Depend, T$) = (0,0,1)时,存在 $Attack2$,即:

若 $M \in U - LU, A, B \in LU, (B \to A \in g_A) \cap (\to A \notin g_B) \cap (\to M("B") \in g_M)$,存在 $T$,如果协议执行完毕,使得 $A \Rightarrow M("B")$ 或 $A \Rightarrow B$.

③当($L\_CD, Depend, T$) = (1,0,0)时,存在 $Attack3$,即:

若 $M \in U, A, B \in LU, (B \to A \in g_A) \cap (\to A \notin g_B) \cap (\to M("B") \in g_M)$,不存在 $T$,如果协议执行完毕,使得 $A \Rightarrow M("B")$ 或 $A \Rightarrow B$.

④当($L\_CD, Depend, T$) = (1,0,1)时,存在 $Attack4$,即:

若 $M \in U, A, B \in LU, (B \to A \in g_A) \cap (\to A \notin g_B) \cap (\to M("B") \in g_M)$,存在 $T$,如果协议执行完毕,使得 $A \Rightarrow M("B")$ 或 $A \Rightarrow B$.

⑤当($L\_CD, Depend, T$) = (0,1,0)时,存在 $Attack5$,即:

若 $M \in U - LU, A, B \in LU, (B \to A \in g_A) \cap (\to A \in g_B) \cap (\to M("B") \in g_M)$,不存在 $T$,如果协议执行完毕,使得 $A \Rightarrow M("B")$ 或无法完成 $A \Rightarrow B$.

⑥当($L\_CD, Depend, T$) = (0,1,1)时,存在 $Attack6$,即:

若 $M \in U - LU, A, B \in LU, (B \to A \in g_A) \cap (\to A \in g_B) \cap (\to M("B") \in g_M)$,存在 $T$,如果协议执行完毕,使得 $A \Rightarrow M("B")$ 或无法完成 $A \Rightarrow B$.

⑦当($L\_CD, Depend, T$) = (1,1,0)时,存在 $Attack7$,即:

若 $M \in LU, A, B \in LU, (B \to A \in g_A) \cap (\to A \in g_B) \cap (\to M("B") \in g_M)$,不存在 $T$,如果协议执行完毕,使得 $A \Rightarrow M("B")$ 或无法完成 $A \Rightarrow B$.

⑧当($L\_CD, Depend, T$) = (1,1,1)时,存在 $Attack8$,即:

若 $M \in LU, A, B \in LU, (B \to A \in g_A) \cap (\to A \in g_B) \cap (\to M("B") \in g_M)$,存在 $T$,如果协议执行完毕,使得 $A$

$\Rightarrow M("B")$或无法完成$A \Rightarrow B$.

### 3.2 攻击算法

根据定义4,提出了一种通用的攻击算法. 该算法是基于主动测试的思想,攻击者 Malice 可以任意截获和注入消息,攻击者知识集合 $\Omega$ 随攻击过程的进行而增大. 针对某一个具体的协议而言,并不是所有 *Attack* 都可选择,要根据协议规范$\{M_1, M_2, \cdots, M_C\}$来进行选择适合 *Attack*. 根据 *Attack* 来选取参与者的状态机 $M_i$. 例如,双方认证协议规范为$\{M_1, M_2\}$,$M_1$,$M_2$ 分别代表发起者、响应者的状态机,当实施 *Attack*1 时,参与者 Alice 选择 $M_1$,Malice 选择 $M_2$. 当涉及到多个 *session*,参与者 Alice 可能选择多个状态机,用 $M_{Ai}$来表示参与者 Alice 选择第 $i$ 个状态机. 算法通过状态机的推断,转移过程、消息的选择,剔除了不合理的测试分支,提高了算法的效率.

该攻击算法分四个部分:

①选择状态机. 先由协议规范$\{M_1, \cdots, M_C\}$确定适当的 *Attack*,再根据 *Attack* 来确定用户 $U$ 的 $M_U$、攻击者的 $M_M$.

②更新目标集合. 目标集合 $G$ 的更新包括用户 $g_U$ 的更新和攻击者 $g_M$ 的更新,由用户 $U$ 和 *Malice* 选取的状态机在协议中的角色来确定.

③具体攻击. 根据当前各状态机的状态进行有针对性的 *Intercept* 或 *Inject*. 先根据当前各状态机来推断截获或注入的状态机集合 $M_D$,再根据 $M_D$ 选择进行 *Intercept* 或 *Inject*. 如果选择 *Intercept*,就从 $M_D$ 中选取可以进行截取的 $M_k$,然后进行 *Intercept*,并更新 $M_k$ 和 *Malice* 的状态机 $M_D.S$,$M_M.S$. 如果选择 *Inject*,就从 $M_D$ 中选取可以进行注入的 $M_k$,求得 $M_k$ 中注入的转移过程 $T_{\text{ture}}$,并用 $lookahead(\Omega, M_k.S, X, t)$选择可以能产生有效输出的消息,然后更新状态 $M_k.S$,$M_M.S$.

④攻击判断. 如果找到认证性的漏洞,算法结束;如果没有找到漏洞,再重复步骤 1 到步骤 4,当尝试超过最大值 *Max*,算法结束. *Max* 是按需设定的尝试次数,用来避免无限测试.

该算法涵盖定义 4 给出的八种攻击类型. 如果该算法可以处理攻击者以合法或非法身份参与,是否利用接收者预言机服务,是否存在认证中心等情况,那么该算法涵盖这八种攻击类型. 该算法通过目标集合来表示攻击拥有合法或非法身份的情况. 例如,Malice 以合法身份与 Alice 进行通信可表示为:$g_M = \{B \rightarrow M\}$,$g_B = \{\rightarrow M\}$;Malice 以非法身份伪装 Alice 与 Bob 进行通信可表示为:$g_M = \{B \rightarrow M("A")\}$,$g_B = \{\rightarrow A\}$. 该算法通过参与者选取多个状态机来表示参与者在不同 *session* 中的角色. 例如,当 Malice 截获到 Alice 发给 Bob 的消息 *msg* 时,发现自己不能阅读,便伪装 Alice 发 *msg* 给 Bob 以获得预言机服务,这可表示为:$g_M = \{A \rightarrow M("B"), B \rightarrow M("A")\}$,$g_A = \{\rightarrow B\}$,$g_B = \{\rightarrow A\}$;当 Malice 没有利用 Bob 预言机服务可表示为:$g_M = \{A \rightarrow M("B")\}$,$g_A = \{\rightarrow B\}$. 该算法通过选取$\{M_1, \cdots, M_C\}$中 $C$ 的取值来表示是否存在认证中心的情况,即多方参与的情况. 因此,该算法涵盖了这八种攻击类型.

## 4 协议的测试与结果分析

### 4.1 Woo-lam 协议描述与测试

Woo-lam 协议是经典的认证协议. 在此协议中假定 Alice 和 Trent 共享对称密钥 $K_{AT}$,Bob 和 Trent 共享对称密钥 $K_{BT}$,协议的最终目标是 Alice 向 Bob 证实自己的身份. Woo-lam 协议的主要交互过程请参见文献[14].

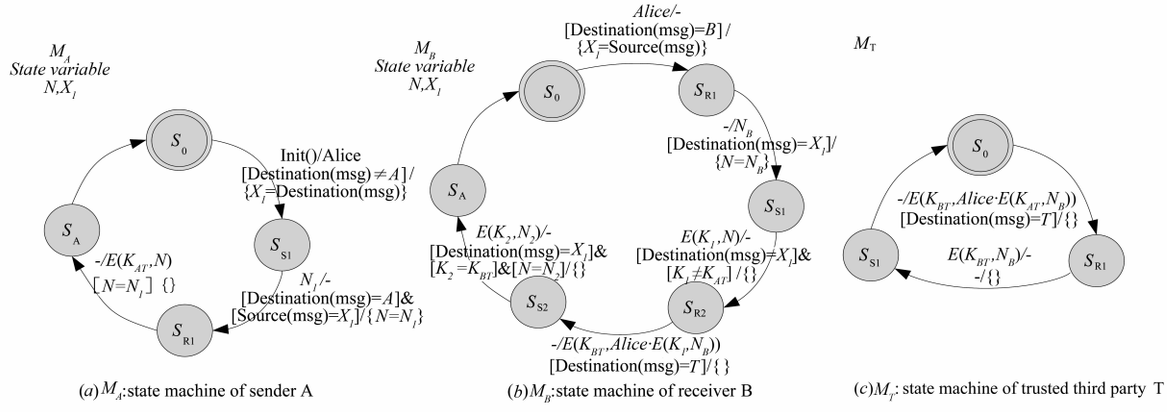


图1　利用SPG-EFSM模型描述的Woo-lam协议规范

#### 4.1.1 Woo-lam 协议规范描述

SPG-EFSM 模型描述 Woo-lam 协议的协议规范如图 1 所示. 假定攻击者 Malice 可以截获每一条消息,知道 Alice、Bob、Trent 从协议开始到结束的每一个状态 $S$,其中 $S_0$ 表示初始状态,$S_{Ri}$表示接受第 $i$ 条消息,$S_{Si}$表示发送第 $i$ 条消息,$S_A$ 表示认证状态.

#### 4.1.2 Woo-lam 协议测试

Woo-lam 协议测试结果通过表 1 给出. 攻击者 Malice 在协议中伪装了 Alice 与 Bob 进行会话,Malice 以合法身份与 Bob 进行会话,Malice 维护伪装 Alice 的状态机和合法身份会话的状态机. 由于存在两次会话,所以 Malice、Bob、Trent 分别都存在着两个状态机. 从表 1 可以看出,攻击算法发现了 Woo-lam 协议存在 *Attack*4,是典型攻击[16]中的平行会话攻击.

**表 1 检测到 Woo-lam 协议的认证错误**

| States $\langle M("A"),B_1,T_1,M,B_2,T_2\rangle$ | Action | Note |
|---|---|---|
| $\langle S_o,S_o,S_o,S_o,S_o,S_o\rangle$ | $M("A")\xrightarrow{\text{Alice}}B$ | $g_B=\{\rightarrow A,A\rightarrow B\}$ |
| $\langle S_{S1},S_{R1},S_o,S_o,S_o,S_o\rangle$ | $M\xrightarrow{\text{Malice}}B$ | $g_B\cup\{\rightarrow M,M\rightarrow B\}$ |
| $\langle S_{S1},S_{R1},S_o,S_{S1},S_{R1},S_o\rangle$ | $B\xrightarrow{N_B}M("A")$ | |
| $\langle S_{R1},S_{S1},S_o,S_{S1},S_{R1},S_o\rangle$ | $B\xrightarrow{N_{B'}}M$ | |
| $\langle S_{R1},S_{S1},S_o,S_{R1},S_{S1},S_o\rangle$ | $M("A")\xrightarrow{E(K_{MT},N_B)}B$ | |
| $\langle S_A,S_{R2},S_o,S_{R1},S_{S1},S_o\rangle$ | $M\xrightarrow{E(K_{MT},N_B)}B$ | |
| $\langle S_A,S_{R2},S_o,S_A,S_{R2},S_o\rangle$ | $B\xrightarrow{E(K_{BT},Alice\cdot E(K_{MT},N_B))}T$ | |
| $\langle S_A,S_{S2},S_{R1},S_A,S_{R2},S_o\rangle$ | $B\xrightarrow{E(K_{BT},Malice\cdot E(K_{MT},N_B))}T$ | |
| $\langle S_A,S_{S2},S_{R1},S_A,S_{S2},S_{R1}\rangle$ | $T\xrightarrow{E(K_{BT},垃圾)}B$ | |
| $\langle S_A,S_{S2},S_{S1},S_A,S_{S2},S_{R1}\rangle$ | $T\xrightarrow{E(K_{BT},N_B)}B$ | $B\Rightarrow A$ |
| $\langle S_A,S_A,S_{S1},S_A,S_{S2},S_{S1}\rangle$ | $predict(G,X)==Succeed$ | *Attack*4 |

Woo-lam 协议存在 *Attack*4 的一个修改方案是在 3. Alice→Bob:$E(K_{AT},N_B)$加入 Alice 的身份信息,但这种方案也存在攻击,由表 2 给出. 攻击者 Malice 在协议中伪装了 Alice、Trent 与 Bob 进行会话,Malice 维护伪装 Alice、Trent 的状态机,来指导攻击进行. 从表 2 可以看出,攻击算法发现了 Woo-lam 协议存在 *Attack*2,是典型攻击[16]中的反射攻击.

**表 2 检测到修改后的 Woo-lam 协议的认证错误**

| States $\langle M("A"),B,M("T")\rangle$ | Action | Note |
|---|---|---|
| $\langle S_o,S_o,S_o\rangle$ | $M("A")\xrightarrow{\text{Alice}}B$ | $g_B=\{\rightarrow A,A\rightarrow B\}$ |
| $\langle S_{S1},S_{R1},S_o\rangle$ | $B\xrightarrow{N_B}M("A")$ | |
| $\langle S_{R1},S_{S1},S_o\rangle$ | $M("A")\xrightarrow{N_B}B$ | |
| $\langle S_A,S_{R2},S_o\rangle$ | $B\xrightarrow{E(K_{BT},Alice\cdot N_B)}M("T")$ | |
| $\langle S_A,S_{S2},S_{R1}\rangle$ | $M("T")\xrightarrow{E(K_{BT},Alice\cdot N_B)}B$ | $B\Rightarrow M("A")$ |
| $\langle S_A,S_A,S_{S1}\rangle$ | $predict(G,X)==Succeed$ | *Attack*2 |

Woo-lam 协议的另一个更新方案是把第 4 个交互流程改为 4. Bob→Trent:$E(K_{BT},Alice\cdot N_B\cdot E(K_{AT},N_B))$,但这种方案也存在攻击,由表 3 给出. Bob 与拥有合法身份的 Malice 进行会话时,Malice 在协议中伪装

了 Alice、Trent 与 Bob 进行会话,Malice 维护伪装 Alice、Trent 的状态机以及合法身份会话的状态机. 从表 3 可以看出,攻击算法发现了 Woo-lam 协议存在 *Attack*4,是典型攻击[16]中的交错攻击.

**表 3　检测到 Woo-lam 协议另一更新方案的认证错误**

| States<br>$\langle M("A"),B_1,M("T"),B_2,M,T\rangle$ | Action | Note |
|---|---|---|
| $\langle S_o,S_o,S_o,S_o,S_o,S_o\rangle$ | $B\xrightarrow{\text{Bob}}M$ | $g_B=\{\rightarrow M,M\rightarrow B\}$ |
| $\langle S_o,S_o,S_o,S_{S1},S_{R1},S_o\rangle$ | $M("A")\xrightarrow{\text{Alice}}B$ | $g_B\cup\{\rightarrow A,A\rightarrow B\}$ |
| $\langle S_{S1},S_{R1},S_o,S_{S1},S_{R1},S_o\rangle$ | $B\xrightarrow{N_B}M("A")$ | |
| $\langle S_{R1},S_{S1},S_o,S_{S1},S_{R1},S_o\rangle$ | $M\xrightarrow{N_B}B$ | |
| $\langle S_{R1},S_{S1},S_o,S_{R1},S_{S1},S_o\rangle$ | $B\xrightarrow{E(K_{BT},N_B)}M$ | |
| $\langle S_{R1},S_{S1},S_o,S_A,S_{R2},S_o\rangle$ | $M("A")\xrightarrow{E(K_{MT},N_B)}B$ | |
| $\langle S_A,S_{R2},S_o,S_A,S_{R2},S_o\rangle$ | $B\xrightarrow{E(K_{BT},Alice\cdot N_B\cdot E(K_{MT},N_B))}M("T")$ | |
| $\langle S_A,S_{S2},S_{R1},S_A,S_{R2},S_o\rangle$ | $M("T")\xrightarrow{E(K_{BT},N_B)}B$ | $B\Rightarrow M("A")$ |
| $\langle S_A,S_A,S_{S1},S_A,S_{S2},S_{S1}\rangle$ | $predict(G,X)==Succeed$ | *Attack*4 |

### 4.2　μTESLA 协议描述与测试

μTESLA 协议是无线传感器网络中经典的广播认证协议[15],其运行过程包括安全初始化、节点加入、数据包认证等阶段,详细请参见文献[15].

#### 4.2.1　μTESLA 协议规范描述

SPG-EFSM 模型描述 μTESLA 协议规范如图 2 所示. 基站($B$)的状态机 $M_B$ 位于图 2($a$)中,先后经历了初始化、接收加入节点请求、给加入节点分发相关参数、数据包发送等过程. 节点($N$)的状态机 $M_N$ 位于 2($b$)中,它有五种状态:初始状态、请求加入、接收系统参数、接收基站数据包、认证基站数据包. 其中 $S_0$、$S_{Ri}$、$S_{Si}$的定义与 4.1.1 节相同,*pid* 和 *sid* 分别代表了节点和基站的身份标识,*Pj* 表示基站发送的第 $j$ 个数据包.

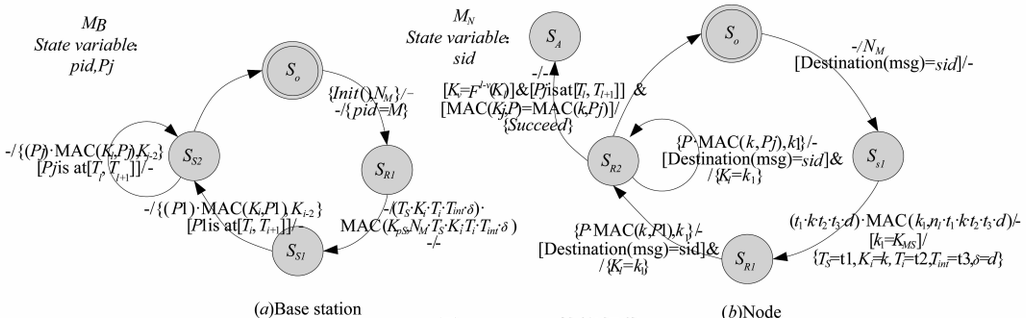


图2　μTESLA协议规范

#### 4.2.2　μTESLA 协议测试

μTESLA 协议测试结果通过表 4 给出. 攻击者 Malice 在协议中干扰基站(Base Station)发给节点(Node)的消息,促使发给节点的消息延时一个时间间隔,也称为“认证漂移”. 这个过程可等价于传统有线网络的拦截和转发过程,因此可将此过程表述为:Malice 伪装 Node 接收 Base Station 的消息,Malice 伪装 Base Station 发送消息给 Node. 从表 4 可以看出,攻击算法发现了 μTESLA 协议存在 *Attack*5.

### 4.3　结果分析与评估

(1)SPG-EFSM 的描述能力分析.

从图 1、图 2 中可以看出,SPG-EFSM 模型可以清晰直观地描述 Woo-lam 协议参与者状态机($M_A$、$M_B$、$M_T$)和 μTESLA 协议参与者状态机($M_B$、$M_N$). 协议运行时各参与者之间的认证关系也能被 SPG-EFSM 模型清晰、准确表现出来. 例如在表 1 中,第二行的 $g_B$ 说明了 Bob

在本次测试过程中将与 Alice 建立认证关系,从第三行的 $g_B$ 可以看出 Bob 又试图与 Malice 建立认证关系,倒数第二行中给出的 $A \Rightarrow B$ 表明了协议执行完后,达到了 Bob 认证 Alice 状态.

**表 4** μTESLA 协议的认证错误

| States $\langle B, M("B"), N, M("N")\rangle$ | Action | Notes |
|---|---|---|
| $\langle S_o, S_o, S_o, S_o\rangle$ | $\text{Node} \xrightarrow{N_M} \text{Base}$ | $g_B = \{N \rightarrow B\}$ |
| $\langle S_{R1}, S_o, S_{S1}, S_o\rangle$ | $\text{Base} \xrightarrow{(T_S \cdot K_i \cdot T_i \cdot T_{int} \cdot \delta) \cdot MAC(K_{BN}, N_M \cdot T_S \cdot K_i \cdot T_i \cdot T_{int} \cdot \delta)} \text{Node}$ | |
| $\langle S_{S1}, S_o, S_{R1}, S_o\rangle$ | $\text{Base} \xrightarrow{P1 \cdot MAC(K_i, P1)} M("N")$ | $P1$ is at $[T_i, T_{i+1}]$ |
| $\langle S_{S2}, S_o, S_{R1}, S_{S2}\rangle$ | $M("B") \xrightarrow{P1 \cdot MAC(K_i, P1)} \text{Node}$ | $P1$ is at $[T_{i+1}, T_{i+2}]$ |
| $\langle S_{S2}, S_{S2}, S_{R1}, S_{S2}\rangle$ | | $MAC(K_{i+1}, P) \neq MAC(K_i, P)$ *Attack*5 |

**表 5** 本文攻击算法与其他方法性能比较

| 测试类型 | 时间复杂度 | 安全属性描述能力 | 覆盖率 |
|---|---|---|---|
| Fuzz testing | $O(n^m)$ | Low | Low |
| Shu G's approach | $O(nm)$ | Moderate | Moderate |
| Our approach | $O(nm)$ | High | High |

(2)攻击算法的性能分析.

令 $Max = n$,协议轮数为 $m$,则该算法的时间复杂度为 $O(nm)$,等同于 Guoqiang Shu 提出算法的时间复杂度. 然而,该攻击算法能够表示拥有合法身份的攻击者参与、多个会话和多方参与等特殊情况. 与其他方法相比,攻击算法在覆盖率方面也有较大的优势,能够以较少的时间复杂度达到较高覆盖率,如表 5 所示.

(3)测试方案的新型漏洞探测能力分析.

该测试方案发现了 Woo-lam 协议及其更新协议当中的漏洞,包括平行会话攻击,反射攻击,交错攻击等典型攻击. 测试还发现了 μTESLA 协议的典型漏洞——认证漂移问题和 DoS 攻击. 从 *Attack*2、*Attack*4 中发现 Woo-lam 协议的安全性依赖于参与者的合法身份标识,也即认证中心与参与者之间的对称密钥,但这并不可靠. 例如,在 *Attack*4 中 Malice 利用 Trent 的预言机服务,获得了对 Alice 的合法身份标识,成功欺骗 Bob 认证了 Alice,而 Alice 认为他与 Bob 达到了认证关系.

## 5 总结

针对协议安全性测试缺乏对安全属性描述及其相关测试方法的问题,本文提出了一种用于检测协议认证安全属性的测试方法. 首先建立 SPG-EFSM 模型,扩展描述目标集合,通过结合参与者目标与协议执行后状态进一步描述认证安全属性. 然后,基于 SPG-EFSM 模型将协议攻击场景分类,以攻击者拥有合法身份、多方参与、多个会话等特殊情况;在此基础上,结合了 Dolev-Yao 攻击模型,设计了一种包含上述分类的协议认证性攻击算法. 通过对 Woo-lam 协议和 μTESLA 协议认证性的测试发现,本方法具有可行性、覆盖率高等特点.

## 参考文献


[1] 周彦伟,杨波,张文政. 异构无线网络可控匿名漫游认证协议[J]. 电子学报,2016,44(5):1117 – 1123.
ZHOU Yan-wei, YANG Bo, ZHANG Wen-zheng. Controllable and anonymous roaming protocol for heterogeneous wireless network [J]. Acta Electronica Sinica, 2016, 44(5):1117 – 1123. (in Chinese)

[2] Dalal Alrajeh, Jeff Kramer, Alessandra Russo, et al. Elaborating requirements using model checking and inductive learning [J]. IEEE Transactions on Software Engineering, 2013, 39(3):361 – 383.

[3] 李顺东,杨坤伟,巩林明,等. 标准模型下可公开验证的匿名 IBE 方案[J]. 电子学报,2016,44(3):673 – 678.
LI Shun-dong, YANG Kun-wei, GONG Lin-ming, et al. A publicly verifiable anonymous IBE scheme in the standard model[J]. Acta Electronica Sinica, 2016, 44(3):673 – 678. (in Chinese)

[4] Wen Tang, Sui Ai-Fen, Wolfgang Schmid. A model guided security vulnerability discovery approach for network protocol implementation[A]. Proceedings of the 13th International Conference on Communication Technology[C]. Beijing, China: IEEE, 2011. 675 – 680.

[5] Andrea Arcuri, Muhammad Zohaib Iqbal, Lionel Briand. Random testing: theoretical results and practical implications[J]. IEEE Transactions on Software Engineering, 2012, 38(2):258 – 277.

[6] G Fraser, A Arcuri. Whole test suite generation[J]. IEEE Transactions on Software Engineering, 2013, 39(2):276 – 291.

[7] Oulu University Secure Programming Group. PROTOS:Security Testing of Protocol Implementation [OL]. http://www.ee.oulu.fi/research/ouspg/protos/index.html, 2012-06-12.

[8] Tal O, Knight S, Dean T. Syntax-based vulnerability testing of frame-based network protocols[A]. Proceedings of the 21nd Conference on Privacy, Security and Trust[C]. Fredericton:IEEE, 2004. 13 - 15.

[9] Jing C, Wang Z, Yin X, et al. Mutation testing of protocol messages based on extended TTCN-3[A]. Proceedings of the 22nd International Conference on Advanced Information Networking and Applications[C]. Okinawa:IEEE, 2008. 667 - 674.

[10] Fabian Yamaguchi, Nico Golde, Daniel Arp, et al. Modeling and discovering vulnerabilities with code property graphs[A]. Proceedings of IEEE Symposium on Security and Privacy[C]. San Jose:IEEE, 2014. 590 - 604.

[11] Gali Mashtizadeh, Andrea Bittau, Dan Boneh, et al. CCFI: cryptographically enforced control flow integrity [A]. Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security[C]. New York: ACM, 2015. 941 - 951.

[12] Shu G, Lee G. Formal methods and tools for testing communication protocol system security[D]. Ohio, USA:Ohio State University, 2008.

[13] Dolev D, Yao A. On the security of public-key protocols [J]. IEEE Transaction on Information Theory, 1983, 29(2):198 - 208.

[14] Woo T, Lam S. Authentication for distributed systems [J]. ACM Transactions on Computer Systems, 1992, 25(1):35 - 39.

[15] Perrig A, Szewczyk R, et al. SPINS:Security protocols for sensor networks [J]. Wireless Networks, 2002, 8(5):521 - 534.

[16] Wenbo Mao. Modern Cryptography:Theory and Practice [M]. New Jersey, USA:Prentice Hall, 2004.

## 作者简介

**何云华**　男,1987 年出生,湖北荆门人,北方工业大学讲师,西安电子科技大学博士,主要研究方向为协议测试、漏洞挖掘.
E-mail:heyunhua610@163.com

**杨　超**　男,1979 年出生,陕西西安人,西安电子科技大学副教授,主要研究方向为密码学与网络安全,云计算及移动智能计算安全.
E-mail:chaoyang@mail.xidian.edu.cn