**文章编号**:1000-582X(2012)02-071-07

# 有效的多协议攻击自动化检测系统

杨元原[1],马文平[1],刘维博[1],张笑笑[2]
(1.西安电子科技大学 教育部计算机网络与信息安全重点实验室,陕西 西安 710071;
2.公安部第三研究所,上海 200031)

**摘 要**:针对当前计算机网络中多个安全协议并行运行时可能出现的多协议攻击问题,提出了一个多协议攻击自动化检测系统(ADMA)。该系统由协议搜索子系统和攻击确认子系统两部分组成,其中协议搜索子系统根据多协议攻击中目标协议与辅助协议加密消息类型一致性条件,自动化搜索可能对目标协议构成威胁的候选辅助协议。攻击确认子系统通过改进的 SAT 模型检测方法,自动化确认目标协议与候选辅助协议是否存在多协议攻击。试验结果表明,ADMA 系统能够实现多协议攻击自动化检测,并且检测中发现了新的多协议攻击。

**关键词**:全协议; 形式化分析; 模型检测;重写规则; 多协议攻击
**中图分类号**:TP 393.08 **文献标志码**:A

# An effective automatic detection system for multi-protocol attack

*YANG Yuan-yuan*[1], *MA Wen-ping*[1], *LIU Wei-bo*[1], *ZHANG Xiao-xiao*[2]
(1. Key Laboratory of Computer Netwrok and Information Security, Ministry of Education Xidian University, Xi'an 710071, Shaanxi, P. R. China; 2. The Third Research Institute of Ministry of Public Security, Shanghai 200031, P. R. China)

**Abstract**: Since there exists multi-protocol attack when several security protocols are co-executed in a computer network, an automatic detection system for multi-protocol attack (ADMA) is proposed. The system is composed of two parts named protocol search subsystem and attack verification subsystem. According to the consistency condition of the type of encrypted messages between the target protocol and the secondary protocol, the protocol search subsystem can automatically search for the candidate secondary protocols, which may be used to attack the target protocol. By improving the SAT-based model checking, attack verification subsystem can automatically verify whether multi-protocol attack exists between the target protocol and the candidate secondary protocols or not. The experiment results show that ADMA system can implement automatic detection for multi-protocol attack, and some new multi-protocol attacks are found in the detection.

**Abstract**: security protocols; formal analysis; model checking; rewrite rules; multi-protocol attack

随着计算机网络的迅猛发展,其对安全协议的需求也日益增加。针对不同目标,不同环境的安全协议不断涌现,甚至对于同一个安全目标,也有众多的安全协议可供选择。因此,在当前的网络中,同时

**收稿日期**:2010-03-09
**基金项目**:国家 863 计划资助项目(2007AA01Z472);国家自然科学基金资助项目(60773002);高等学校创新引智计划资助项目(B08038);高等学校博士学科点专项科研基金(20100203110003)
**作者简介**:杨元原(联系人),男,西安电子科技大学博士,主要从事安全协议形式化分析方向研究,(Tel) 029-88206390;(Email) yangyuan2_2001@yahoo.cn。

运行着大量不同的安全协议。然而,当前的形式化分析工具[1-7]及相关研究[8-11]往往只能分析协议在独立运行时的安全性,不能分析其在多协议环境下的安全性,因此,开发一个多协议攻击检测工具日趋紧迫。

文献[12-13]指出了多协议攻击检测在未来安全协议分析中的必要性和重要性。文献[14]实现了任意2个协议之间多协议攻击的自动化确认,但其并不能在任意的多协议的环境中查找多协议攻击。文献[15]对文献[14]进行了补充,发现了新的多协议攻击。文献[16]提出了一个可以降低多协议攻击检测复杂性的方法。然而,所有的这些文献都不能自动化查找多协议攻击,即不能确认某个协议在任意多协议环境下的安全性。

提出了一个多协议攻击自动化检测系统ADMA(automatic detection system for multi-protocol attack)。该系统可以在任意的多协议环境中,自动化检测某个协议是否存在多协议攻击。ADMA系统由协议搜索子系统和攻击确认子系统两部分组成。其中协议搜索子系统利用协议搜索算法,自动化查找可能被攻击者利用,并发起多协议攻击的协议,搜索到的协议称为候选辅助协议。攻击确认子系统利用扩展的SAT模型检测工具,对在协议搜索子系统中找到的目标协议和候选辅助协议进行多协议攻击检测,确认是他们是否存在多协议攻击。

# 1 候选辅助协议

## 1.1 多协议攻击

当前安全协议形式化分析往往只关注协议在独立运行时的安全性,忽略了其在多协议环境中的安全。文献[14]首先分析了多协议攻击的可行性,并给出了多协议攻击的定义,其定义如下

**定义** 1:多协议攻击是一个二元组$(ps,c)$,$ps$表示一组没有攻击的安全协议集合,$c$是安全目标。如果存在多协议攻击,当且仅当下面条件成立:

$(|ps|>1)\wedge(\text{attack}(ps,c))\wedge(\forall ps'\subset ps:\neg\text{attack}(ps',c))$其中$|ps|>1$表示协议组中必须包含多个协议,$\text{attack}(ps,c)$表示$ps$中存在对安全目标$c$的攻击,$\forall ps'\subset ps:\neg\text{attack}(ps',c)$表示$ps$中所有协议都是攻击的组成部分。

$ps$中所有的协议都是安全的,但是他们的组合却是不安全的。在多协议攻击中,用来测试安全性的协议(即被攻击的协议)称为目标协议,$c$是目标协议的不安全状态;目标协议之外的协议称为辅助协议,辅助协议被攻击者利用,生成一些攻击者不能生成的消息,实现攻击目标协议的目的。

## 1.2 候选辅助协议

要在多协议环境中确定目标协议是否存在多协议攻击,首先要确定哪些协议可能会被攻击者利用,并对目标协议发起攻击。ADMA系统根据多协议攻击的必要性条件,提出了候选辅助协议的概念,候选辅助协议包含了多协议环境中所有可能被攻击者利用,对目标协议发起攻击的协议。

在多协议环境中,攻击者能否利用其他协议获得目标协议发起者(或响应者)期待的加密消息是攻击成功的关键,因为根据D-Y模型的完美密码假设,攻击者不能随意生成加密消息,他向目标协议的发起者(或响应者)发送的加密消息只能由其他协议生成。对于一个普通协议,虽然不能直接判断出其是否会对目标协议构成多协议攻击,但是可以根据其加密消息的类型来判断多协议攻击的可能性。

在安全协议中,每条消息都有固定的类型,消息的类型可分为原子类型和合成类型,其中原子类型包括临时值类型(nonce,$N$),对称密钥类型(symmetric key,$K$),公钥类型(public key,$PK$),私钥类型(private key,$SK$)和主体类型(agent,$Ag$)等等,合成类型包括连接类型和加密类型。下面定义的类型函数$T(m)$将根据协议的消息返回其对应的类型。

**定义** 2:类型函数$T(m)$定义如下

i)如果$m$是原子类型消息,则$T(m)=t$,$t$表示$m$的类型,$t\in\{N,K,PK,SK,Ag\}$;

ii)如果$m$是连接类型的消息,即$m=<m_1,\cdots,m_n>$,则$T(m)=[T(m_1),\cdots,T(m_n)]$;

iii)如果$m$是加密类型的消息,即$m=\{m'\}_k$,则$T(m)=[T(m'),T(k)]$。

为了标识协议的加密消息及其类型,ADMA系统定义了加密消息集合和加密消息类型集合的概念。

**定义** 3:协议$P$中所有加密消息构成的集合称为$P$的加密消息集合,记为$S^P=\{l_1,\cdots,l_n\}$,其中$l_i=\{m_i\}_{ki}$,$(1\leqslant i\leqslant n)$,$T(S^P)=\{T(l_1),\cdots,T(l_n)\}$称为协议$P$的加密消息类型集合;协议$P$中某个主体$X(X\in\{I,R\}$,$I$表示发起者,$R$表示响应者)期待的加密消息构成的集合称为$X$的加密消息集合,记为$S_X^P$,$S_X^P\subseteq S^P$,$T(S_X^P)$称为$X$的加密消息类型集合。

如果一个目标协议$P$存在多协议攻击,以其发起者$I$被攻击为例,那么攻击者必然利用辅助协议

$Pa$ 生成了发起者 $I$ 期待的所有的加密消息 $S_I^P$,即 $S_I^P \subseteq S^{Pa}$,而这些加密消息的类型必然是一致的,即目标协议发起者 $I$ 的加密消息类型集合 $T(S_I^P)$ 必然是辅助协议加密消息类型集合 $T(S^{Pa})$ 的子集,也就是说 $T(S_I^P) \subseteq T(S^{Pa})$,这是多协议攻击的必要性条件。根据这个条件,ADMA 定义了候选辅助协议的概念。

定义 4:令目标协议 $P$ 的发起者和响应者期待的加密消息集合分别为 $S_I^P$ 和 $S_R^P$,协议 $Pa$ 的加密消息集合为 $S^{Pa}$,则 $Pa$ 是 $P$ 的候选辅助协议当且仅当 $(T(S_I^P) \subseteq T(S^{Pa})) \vee (T(S_R^P) \subseteq T(S^{Pa}))$。

ADMA 利用多协议攻击的必要性条件来查找候选辅助协议。在多协议环境中,如果目标协议存在候选辅助协议,那么目标协议和候选辅助协议之间可能存在多协议攻击,如果目标协议不存在候选辅助协议,那么目标协议肯定不存在多协议攻击。

下面以 MAP1 协议和 EVE1 协议为例来说明定义 2、3 和 4 中的概念,MAP1 协议由以下 3 条消息构成

Mess1　$A \rightarrow B: N_A$;

Mess2　$B \rightarrow A: N_B, \{B, A, N_A, N_B\}_{Kab}$;

Mess3　$A \rightarrow B: \{A, N_B\}_{Kab}$,

其中:$A$,$B$ 表示发起者和响应者;$N_A$,$N_B$ 分别是 $A$ 和 $B$ 生成的临时值;Kab 为 $A$ 和 $B$ 之间的对称密钥。EVE1 协议与 MAP1 协议相似,只有第二条消息与 MAP1 协议不同,EVE1 协议如下

Mess1　$A \rightarrow B: N_A$;

Mess2　$B \rightarrow A: N_B, \{A, B, N_A, N_B\}_{Kab}$;

Mess3　$A \rightarrow B: \{A, N_B\}_{Kab}$。

令 MAP1 协议为目标协议,EVE1 协议为普通协议。根据 MAP1 协议,其发起者 A 期待的消息为 Mess2,因此 $S_A^{MAP1} = \{\{B, A, N_A, N_B\}_{Kab}\}$,$T(S_A^{MAP1}) = \{[Ag, Ag, N, N, K]\}$。其响应者 $B$ 期待的消息为 Mess1 和 Mess3,但由于只有 Mess3 中包含加密信息,因此 $S_B^{MAP1} = \{\{A, N_B\}_{Kab}\}$,$T(S_B^{MAP1}) = \{[Ag, N, K]\}$。而 $EVE1$ 协议的 $S^{EVE1} = \{\{A, B, N_A, N_B\}_{Kab}, \{A, N_B\}_{Kab}\}$,其类型集合 $T(S^{EVE1}) = \{[Ag, Ag, N, N, K], [Ag, N, K]\}$。由于 $T(S_A^{MAP1}) \subseteq T(S^{EVE1})$ 与 $T(S_B^{MAP1}) \subseteq T(S^{EVE1})$ 都成立,因此 EVE1 协议是 MAP1 协议的候选辅助协议,而且 MAP1 协议的发起者 $A$ 和响应者 $B$ 都可能被攻击。

## 2　协议搜索子系统

要确定目标协议是否存在多协议攻击,首先要查找目标协议的候选辅助协议。ADMA 系统采用协议搜索子系统来查找目标协议的候选辅助协议。该子系统首先构建了一个协议库来模拟多协议环境,该库从 SPORE[17] 中选取了 16 个协议,并加入了 MAP1 协议,EVE1 协议,ISO/IEC 9798-2 和 ISO/IEC 9798-3 协议,库中所有协议在独立环境中都是安全的。构造的协议库如图 1 所示,其中 1-20 是基于对称密钥的协议,21-24 是基于公钥的协议。

协议 1. MAP1　　协议 2.　EVE1

协议 3. BAN modified Andrew secure RPC

协议 4. Lowe modified BAN concrete Andrew secure RPC

协议 5. ISO/IEC 9798-2 one-pass unilateral authentication

协议 6. ISO/IEC 9798-2 two pass mutual authentication

协议 7.　Lowe modified Denning-Sacco shared key

协议 8.　Kao Chow authentication v.2

协议 9. Kao chow authentication v.3

协议 10. lowe modified KSL

协议 11. Amened Needham Schroeder symmetric key

协议 12. Lowe modified Wide Mouth Frog

协议 13. Woo and Lam Pi　　协议 14. Woo and Lam Pi 1.

协议 15. Woo and Lam Pi 2.　　协议 16. Woo and Lam Pi 3

协议 17. Woo and Lam Pif　　协议 18. Yahalom

协议 19. Lowes modified version of Yahalom

协议 20. Paulson's strengthened version of Yahalom

协议 21. ISO/IEC 9798-3 one-pass unilateral authentication

协议 22 ISO/IEC 9798-3 two-pass unilateral authentication

协议 23. ISO/IEC 9798-3 two-pass mutual authentication

协议 24. ISO/IEC 9798-3 three-pass mutual authentication

**图 1　协议库**

根据图 1 中的协议,协议搜索子系统首先构造所有协议的加密消息类型集合 $PTS$,$PTS = \{T(S^i) | 1 \leqslant i \leqslant 24\}$,$T(S^i)$ 表示协议 $i$ 的加密消息类型集合。

该子系统利用协议搜索算法实现候选辅助协议的自动化搜索,协议搜索算法的搜索过程如下:假设以协议库中的第 $m$ 个协议为目标协议,该算法首先确定目标协议的发起者和响应者的加密消息类型集合 $T(S_I^m)$ 和 $T(S_R^m)$,当以发起者 $I$ 为攻击目标进行候选辅助协议搜索时,设 $T(S_I^m) = \{T(l_i) | l_i \in S_I^m, 1 \leqslant i \leqslant |S_I^m|\}$,其中 $l_i$ 为 $S_I^m$ 中的第 $i$ 个加密消息,$|S_I^m|$ 表示集合元素个数,如果存在协议 $k$ 使 $(T(S_I^m) \subseteq T$

$(S^k))\wedge(k\neq m)$成立，那么协议库中的第 $k$ 个协议就是该目标协议的候选辅助协议。

协议搜索算法的伪码如下

Input:$(m,T(S_I^m),\mathrm{PTS}/\{T(S^m)\})$,Output:$Cp$

$k=1$； $n=|S_I^m|$； $Cp=\emptyset$；

while($k\leqslant 23$)

{ $i=1$； $Bv=$True；

while($i\leqslant n$)

{ if($T(l_i)\in T(S^k)$) then $i=i+1$；

else Bv=False； break；

}

if($Bv=$True) then $Cp=Cp\cup\{k\}$；

else $Cp=Cp$；

$k=k+1$；

}

在算法的输入$(m,T(S_I^m),\mathrm{PTS}/\{T(S^m)\})$中，$m$ 表示目标协议的编号，$\mathrm{PTS}/\{T(S^m)\}$表示协议库中去掉目标协议 $m$ 后得到的加密消息类型集合。$i$ 用来判断 $T(S_I^m)$中每个 $T(l_i)$是否都在 $T(S^k)$中，如果都在其中，则 $Bv$ 返回 *true*，协议 $k$ 就是协议 $m$ 的候选辅助协议，否则，$Bv$ 返回 false，算法搜索第 $k+1$ 个协议。输出 $Cp$ 是一整数集合，表示搜索到的候选辅助协议的编号。

协议搜索子系统利用 SML 语言实现了协议搜索算法，并从协议库中选取了 8 个协议进行了测试。在测试中，对称密钥类型 $K$ 又被进一步分为 $K1$，$K2$，$K3$ 三种类型，其中 $K1$ 表示普通主体之间的密钥类型(协议 1-4)，$K2$ 表示主体与服务器之间的密钥类型，$K3$ 表示协议中新生成的密钥类型。测试的结果如表 1 所示。

**表 1 候选辅助协议列表**

| 目标协议 | 候选辅助协议—发起者 | 候选辅助协议—响应者 |
|---|---|---|
| 协议 1 | 协议 2 | 协议 2 |
| 协议 3 | $\emptyset$ | $\emptyset$ |
| 协议 6 | 协议 5 | 协议 5 |
| 协议 8 | 协议 9 | $\emptyset$ |
| 协议 11 | $\emptyset$ | $\emptyset$ |
| 协议 18 | 协议 19 | $\emptyset$ |
| 协议 23 | 协议 21 | 协议 21 |
| 协议 24 | 协议 22 | 协议 22 |

在表 1 中，目标协议表示被测试的协议，候选辅助协议－发起者(响应者)表示以目标协议发起者(响应者)的加密消息类型集合为目标搜索得到的候选辅助协议。从表 1 可以看出，协议 3 和 11 不存在候选辅助协议，因此他们不存在多协议攻击。协议 1、6、23 和 24 的发起者和响应者都存在候选辅助协议；协议 8 和 18 的发起者存在候选辅助协议。

## 3 攻击确认子系统

对于表 1 中存在候选辅助协议的目标协议，ADMA 系统利用其攻击确认子系统进行多协议攻击自动化确认，自动化确认基于 SAT 模型检测[4]思想。SAT 模型检测是一种高效的安全协议形式化分析方法，但其原始的模型也不能检测多协议攻击。为此，攻击确认子系统对其进行了多协议扩展，在诚实主体重写规则和攻击者重写规则中增加了协议编号 $Pn$(protocol number)，并使攻击者能够识别多个协议，而且能与不同的协议进行交互，由此实现了多协议攻击自动化检测。

**定义** 5：协议编号 $Pn$ 是重写规则中用来区分不同协议的正整数，$Pn\in\{1,2\}$。当 $Pn=1$ 时，表示该协议为目标协议；当 $Pn=2$ 时，表示该协议为候选辅助协议。

在 SAT 模型检测中，协议行为和攻击者行为分别被转化为诚实主体重写规则和攻击者重写规则，随后协议的初始状态和不安全状态也被确定。系统从协议的初始状态出发，根据重写规则，不断更新系统状态，并且利用 SAT 分析器检测是否存在从初始状态到目标状态的路径，如果 SAT 分析器返回真，则协议存在攻击，攻击路径被返回；否则在检测的范围内，不存在攻击。下面以 MAP1 和 EVE1 协议为例来说明攻击确认子系统如何实现多协议攻击自动化确认。

### 3.1 MAP1 协议的重写规则

在攻击确认子系统中，MAP1 协议被转化为带有协议编号 $Pn$ 的重写规则(1)-(4)。

$$\begin{aligned}&\mathrm{state}(0,A,A,[A,B,Kab],Se,Pn)\\&\xrightarrow{\mathrm{step}_0(Pn,A,B,Kab,Se)}\exists N_A:\\&\mathrm{state}(2,B,A,[A,B,N_A,Kab],Se,Pn)\cdot\\&\mathrm{msg}(1,A,B,\mathrm{Mess1},Pn)\cdot\\&\mathrm{witness}(A,B,na0,N_A,Pn),\qquad(1)\\&\mathrm{state}(1,A,B,[A,B,Kab],Se,Pn)\cdot\\&\mathrm{msg}(1,A,B,\mathrm{Mess1},Pn)\end{aligned}$$

$$\xrightarrow{\text{step}_1(Pn,A,B,N_A,Kab,Se)}$$

$$\exists N_B:\text{msg}(2,B,A,\text{Mess2},Pn)\cdot \text{state}(3,A,B,[A,B,N_A,N_B,Kab],Se,Pn)\cdot \text{witness}(B,A,nb0,N_B,Pn), \quad (2)$$

$$\text{state}(2,B,A,[A,B,N_A,Kab],Se,Pn)\cdot \text{msg}(2,B,A,\text{Mess2},Pn)$$

$$\xrightarrow{step_2(Pn,A,B,N_A,N_B,Kab,Se)}$$

$$\text{msg}(3,A,B,\text{Mess3},Pn)\cdot \text{state}(4,B,A,[A,B,N_A,N_B,Kab],Se,Pn)\cdot \text{request}(A,B,nb0,N_B,Pn), \quad (3)$$

$$\text{state}(3,A,B,[A,B,N_A,N_B,Kab],Se,Pn)\cdot \text{msg}(3,A,B,\text{Mess3},Pn)$$

$$\xrightarrow{\text{step}_3(Pn,A,B,N_A,N_B,Kab,Se)}$$

$$\text{request}(B,A,na0,N_A,Pn)\cdot \text{state}(5,A,B,[A,B,N_A,N_B,Kab],Se,Pn), \quad (4)$$

其中:state($J,A_1,A_2,[T_1,T_2,\cdots,T_k],Se,Pn$)表示主体 $A_2$ 的状态,$[T_1,T_2,\cdots,T_k]$是主体 $A_2$ 已知的消息集合;$Se$ 表示协议会话号;$Pn$ 是表示该协议的编号;msg($J,A_1,A_2,M,Pn$)表示在协议 $Pn$ 中,主体 $A_1$ 在第 $J$ 步向主体 $A_2$ 发送了消息 $M$;witness($A_1,A_2,na0,N_A,Pn$)表示在协议 $Pn$ 中,$A_1$ 希望与 $A_2$ 用临时值 $N_A$ 作为认证的身份标识,$na0$ 用来标识 $N_A$ 的身份;request($A_2,A_1,na0,N_A,Pn$)表示在协议 $Pn$ 中,$A_2$ 同意与 $A_1$ 用临时值 $N_A$ 作为认证的身份标识;step$_i$(·)($0\leqslant i\leqslant 3$)是重写规则的行为标签,每个行为标签都唯一标识了一个重写规则。

攻击确认子系统同时也将 EVE1 协议转化为重写规则,由于 EVE1 协议与 MAP1 协议只有第二条消息不同,,因此其重写规则与 MAP1 协议类似,这里不再赘述。

### 3.2　攻击者重写规则

在攻击确认子系统中,攻击者重写规则由以下 6 个规则表示。该子系统的攻击者可以截获任意协议的数据,并利用这些数据与其他协议进行交互,以获得有用信息,达到攻击目标协议的目的。攻击者重写规则如下

$$\text{msg}(J,A,B,M,Pn)\xrightarrow{\text{divert}(Pn,A,B,J,M)} ik(M), \quad (5)$$

$$ik(M)\cdot ik(K)\xrightarrow{\text{encrypt}(K,M)} ik(M)\cdot ik(K)\cdot ik(\{M\}_K), \quad (6)$$

$$ik(\{M\}_K)\cdot ik(K^{-1})\xrightarrow{\text{decrypt}(K,M)} ik(M), \quad (7)$$

$$ik(M_1)\cdot ik(M_2)\xrightarrow{\text{pairing}(M_1,M_2)} ik(<M_1,M_2>), \quad (8)$$

$$ik(<M_1,M_2>)\xrightarrow{\text{decompose}(M_1,M_2)} ik(M_1)\cdot ik(M_2), \quad (9)$$

$$ik(M)\cdot ik(A)\cdot ik(B)\xrightarrow{\text{fake}(Pn,A,B,M,J)} ik(M)\cdot ik(A)\cdot ik(B)\cdot \text{msg}(J,A,B,M,Pn)。 \quad (10)$$

$ik$(·)表示攻击者获得的知识。规则(5)表示攻击者截获 Pn 协议中 $A$ 发送给 $B$ 的消息 $M$,规则(6)、(7)、(8)和(9)分别表示攻击者对消息进行加密,解密,组合,分解等操作,规则(10)表示攻击者在协议 Pn 的第 $J$ 步假冒主体 $A$ 向 $B$ 发送消息 $M$。

在重写规则(1)-(10)中,所有的符号如 $A,B$,Kab,$N_A$ 等都由大写字母表示,用来表示变量,在进行多协议攻击检测时,这些变量会由代表具体参数的小写字母替换。

### 3.3　多协议攻击确认

在攻击确认子系统中,MAP1 协议的 $Pn$ 值被设为 1,表示目标协议,EVE1 协议的 $Pn$ 值被设为 2,表示候选辅助协议。

多协议攻击检测的初始状态为:state(0,$a,a,[a,b,$kab$],1,1)\cdot$ state(0,$b,b,[a,b,$kab$],1,2)\cdot$ state(1,$a,b,[a,b,$kab$],1,1)\cdot$ state(1,$b,a,[a,b,$kab$],1,2)\cdot ik(a)\cdot ik(b)$在初始状态中,state(·)和 $ik$(·)中的参数都由小写字母 $a,b$,kab 等表示,他们代表具体执行协议的主体及参数,这些参数将会替换重写规则中相应的变量以执行重写规则。

当目标协议 MAP1 的发起者 $A$ 被设为攻击目标时,攻击确认子系统的目标状态(即不安全状态)为

(request($A,B,nb0,N_B,1$)$\wedge\neg$witness($B,A,nb0,N_B,1$))它表示协议 1 中主体 $A$ 接受了 $B$ 的请求(request($A,B,nb0,N_B,1$))但主体 $B$ 并未发送过这样的请求($\neg$witness($B,A,nb0,N_B,1$))。

在检测过程中,攻击确认子系统从多协议攻击检测的协议初始状态出发,根据 MAP1 协议,EVE1 协议和攻击者的重写规则,寻找下一步可执行的行为和可到达的新的状态,然后通过图形编码,将每步可执行的重写规则都转化为合取范式,并将其与协议的初始状态和目标状态结合,构造 SAT 的命题公式。SAT 命题公式的表达式为

$〔\prod〕_k = I(f^0) \wedge \bigwedge_{i=0}^{k-1} T_i(f^i, a^i, f^{i+1}) \wedge B(f^k)$，其中：$k$ 表示执行的步骤；$I(f^0)$表示初始状态；其对应多协议攻击检测的初始状态，$T_i(\cdot)$表示执行第 $i$ 步后的状态变化；$f^i$ 表示第 $i$ 步的状态；$f$ 由 state(·)，msg(·)，witness(·)，request(·)，$ik$(·)等组成，$a^i$ 表示第 $i$ 步行为的集合，$a$ 包括诚实主体的行为 $\text{step}_i(\cdot)$和攻击者的行为 divert(·)，encrypt(·)等等。$f^{i+1}$是第 $i+1$ 步的状态集合。$B(f^k)$表示检测的目标。

在$〔\Pi〕_k$中，每个元素都被看作变量，$i$ 每增加一步，系统都要检测是否存在一组变量赋值，使$〔\Pi〕_k$为真。判断$〔\Pi〕_k$是否为真的问题称为 SAT 问题。解决 SAT 问题的算法称为 SAT 分析器。如果 SAT 分析器返回结果为真，那么目标可达，该协议存在攻击，攻击路径被返回；如果为假，那么说明在检测的 $k$ 步内，协议不存在攻击。

ADMA 系统利用 SML 语言实现了多协议攻击确认子系统。在该子系统中，采用图形编码实现重写规则到 SAT 命题公式$〔\Pi〕_k$的转化，采用 GRASP[18]算法实现 SAT 分析器。按照以上的设定的数据，该子系统检测到 MAP1 和 EVE1 协议之间存在多协议攻击，其检测结果如下，攻击步骤：$k=7$，变量个数：185，范式个数：213，求解时间：23.485 s。其中，攻击步骤指检测到攻击时$〔\Pi〕_k$执行的步骤，变量个数和范式个数指检测到攻击时$〔\Pi〕_k$中变量个数和合取范式个数，求解时间指检测到攻击时所用的时间。检测到的攻击路径如下

1) $\text{step}_0(1, a, b, kab, 1)$；

2) $\text{divert}(1, a, b, 1, na)$；

3) $\text{fake}(2, b, a, na, 1)$；

4) $\text{step}_1(2, a, b, na, kab, 1)$；

5) $\text{divert}(2, a, b, 2, <na', \{b, a, na, na'\}_{kab}>)$；

6) $\text{fake}(1, b, a, <na', \{b, a, na, na'\}_{kab}>, 1)$；

7) $\text{step}_2(1, a, b, na, na', kab, 1)$。

在这个攻击中，步骤 1)表示主体 $a$ 在 MAP1 协议中向主体 $b$ 发送了临时值 $na$，通过步骤 2)-3)，攻击者截获了 $na$，并在 EVE1 中冒充主体 $b$ 向 $a$ 发送了 $na$，步骤 4)表示主体 $a$ 在协议 EVE1 中对 $na$ 进行了响应，向 $b$ 发送了消息$<na', \{b, a, na, na'\}_{kab}>$，5)-6)表示攻击者截获了这条消息，并在 MAP1 协议中冒充 $b$ 向 $a$ 发送了$<na', \{b, a, na, na'\}_{kab}>$，在步骤(7)中，$a$ 接受了该消息，并认为他与主体 $b$ 完成了 MAP1 协议，但实际上 $b$ 并未参与任何协议，攻击者利用 EVE1 协议产生了 MAP1 协议中 $a$ 期待的消息，并与 $a$ 完成了 MAP1 协议。

攻击确认子系统对表 2 中的协议进行了检测，检测的结果如表 2 所示。

**表 2 多协议攻击检测结果**

| 协议 | | 是否存在多协议攻击 | |
|---|---|---|---|
| 目标协议 | 候选辅助协议 | 发起者 | 响应者 |
| 协议 1 | 协议 2 | 是 | 是 |
| 协议 6* | 协议 5 | 否 | 是 |
| 协议 8 | 协议 9 | 是 | / |
| 协议 18 | 协议 19 | 是 | / |
| 协议 23* | 协议 21 | 否 | 是 |
| 协议 24* | 协议 22 | 是 | 否 |

从表 2 可以看出，协议 1 的发起者和响应者都会被协议 2 攻击。协议 8 和 18 的发起者会被其候选辅助协议攻击，而他们的响应者不存在候选辅助协议，所以不存在多协议攻击。协议 6 和 23 的响应者会被其候选辅助协议攻击，发起者不会被其候选辅助协议攻击。协议 24 的发起者会被其候选辅助协议攻击，而发起者不会受到多协议攻击。带 * 号的协议 6，23 和 24 表示新发现的多协议攻击。

## 4 结 论

提出了多协议攻击自动化检测系统 ADMA。该系统由协议搜索子系统和攻击确认子系统 2 部分组成。协议搜索子系统负责自动化查找所有可能被攻击者利用，并对目标协议发起攻击的候选辅助协议，攻击确认子系统负责对目标协议和候选辅助协议之间是否存在多协议攻击进行自动化检测。未来的研究是对 ADMA 系统进一步优化，提高检测效率，并检测更多的多协议攻击。

**参考文献：**

[1] BASIN D, MODERSHEIM S, VIGANO L. OFMC: a symbolic model checker for security protocols [J]. International Journal of Information Security, 2005, 4 (3): 181-208.

[2] TURUANI M. The CL-atse protocol analyser[C]// Proceedings of the 17th International Conference on Term Rewriting and Applications, August 12-14, 2006. WA, USA: Springer Berlin / Heidelberg, 2006:

277-286.

[3] BOICHUT Y, HEAM P C, KOUCHNARENKO O. Tree automata for detecting attacks on protocols with algebraic cryptographic primitives[J]. Electronic Notes in Theoretical Computer Science, 2009, 239: 57-72.

[4] ARMANDO A, COMPAGNA L. SAT-based model-checking for security protocols analysis [J]. International Journal of Information Security, 2008, 7 (1): 3-32.

[5] BLANCHET B. Automatic verification of correspondences for security protocols[J]. Journal of Computer Security, 2009, 17(4):363-434.

[6] CREMERS C. The scyther tool: verification, falsification and analysis of security protocols[C] // Proceedings of the 20th International Conference on Computer Aided Verification, July 7-14, 2008. Princeton, USA: Springer Berlin / Heidelberg, 2008: 414-418.

[7] 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.
LI MENG-JUN, LI ZHOU-JUN, CHEN HUO-WANG. SPVT: an efficient verification tool for security protocol[J]. Journal of Software, 2006, 17 (4): 898-906.

[8] 杨明,罗军舟. 基于认证测试的安全协议分析[J]. 软件学报, 2006, 17 (01): 148-156.
YANG MING, LUO JUN-ZHOU. Analysis of security protocols based on authentication test[J]. Journal of Software, 2006, 17 (01): 148-156.

[9] 卓继亮, 李先贤, 李建欣, 等 . 安全协议的攻击分类及其安全性评估[J]. 计算机研究与发展, 2005, 42 (7): 1100-1107.
ZHUO JI-LIANG, LI XIAN-XIAN, LI JIAN-XIN, et al. A new taxonomy of attacks on security protocols and their security evaluation[J]. Journal of Computer Research and Development, 2005, 42 (7): 1100-1107.

[10] 周永彬, 张振峰, 冯登国. 一种认证密钥协商协议的安全分析及改进[J]. 软件学报, 2006, 17 (04): 868-875.
ZHOU YONG-BIN, ZHANG ZHEN-FENG, FENG DENG-GUO. Analysis and improvement of a security-provable mutually authenticated key agreement protocol [J]. Journal of Software, 2006, 17 (04): 868-875.

[11] BHARGAVAN K, FOURNET C, GORDON A D, et al. Verified interoperable implementations of security protocols [J]. ACM Transactions on Programming Languages and Systems, 2008, 31(1): 1-61.

[12] GIAMPAOLO B. What is correctness of security protocols[J]. Journal of Universal Computer Science, 2008, 14(12): 2083-2107.

[13] KHOURY P E, HACID M S, SINHA S K, et al. A study on recent trends on integration of security mechanisms[J]. Studies in Computational Intelligence, 2009, 223: 203-224.

[14] CREMERS C. Feasibility of multi-protocol attacks[C] // Proceedings of the first International Conference on Availability, Reliability and Security, April 20-22, 2006. Vienna, Astria: IEEE Computer Society, 2006: 287-294.

[15] MATHURIA A, SINGH A R, SHARAVAN P V, et al. Some new multi-protocol attacks[C] // Proceedings of the 15th International Conference on Advanced Computing and Communications, Dec. 18-21, 2007. Guwahati, India: IEEE Computer Society, 2007: 465-471.

[16] ANDOVA S, CREMERS C, GJOSTEEN K, et al. A framework for compositional verification of security protocols[J]. Information and Computation, 2008, 206 (2-4): 425-459.

[17] Security protocols open repository [EB/OL]. (2010-06-05)[2010-10-01]. http: // www. lsv. ens-cachan. fr/Software/spore/table. html.

[18] MARQUES-SILVA J P, SAKALLAH K A. GRASP-a search algorithm for propositional satisfiability [J]. IEEE Transactions on Computers, 1999, 48 (5): 506-521.

(编辑　侯　湘)