第 33 卷 第 6 期 计 算 机 工 程 2007 年 3 月
Vol.33 No.6 Computer Engineering March 2007
·安全技术· 文章编号：1000—3428(2007)06—0126—03 文献标识码：A 中图分类号：TP393.09

Otway-Rees 协议并行攻击的 SG 逻辑分析

王小锐[1]，陈连俊[2]，季庆光[3]，曹正君[2]
（1. 解放军信息工程大学电子技术学院，郑州 450004；2. 总参 51 所，北京 100072；
3. 中国科学院软件研究所信息安全国家重点实验室，北京 100080）

摘 要：网络信息安全很大程度上取决于密码协议的安全，重放攻击和并行攻击是对密码协议的常见攻击，能够分析并行攻击的形式化分析方法并不多见。该文介绍了一种分析密码协议并行攻击和重放攻击的逻辑方法——SG 逻辑，应用它对改进版的 Otway-Rees 协议进行了分析，找出了 BAN 类逻辑所不能分析出来的缺陷，针对该缺陷给出了协议的进一步改进，并推证了改进后的协议对 SG 逻辑的分析是安全的。

关键词：SG 逻辑；并行攻击；Otway-Rees 协议；安全性分析

SG Logic Analysis of Otway-Rees Protocol Interleaving Attack

WANG Xiaorui[1], CHEN Lianjun[2], JI Qingguang[3], CAO Zhengjun[2]
（1. Institute of Electronic Technology, PLA Information Engineering University, Zhengzhou 450004; 2. The 51th Institute, General Staff Head Quarters, Beijng 100072; 3. State Key Laboratory of Information Security, Institute of Software, Chinese Academy of Sciences, Beijing 100080）

【Abstract】 The security of network relies to a great extend on the security of cryptographic protocol, while interleaving attack and replaying attack are very common in everyday time, and there are little methods for analyzing this inaccuracies. This paper introduces a method for dealing with this problem, which is called SG logic. It shows the syntax and the semantic of SG logic in a great detail, finds the fault of Otway-Rees protocol, and gives a method to improve it. At last, it proves that the improved protocol is secure under the given attack.

【Key words】 SG logic; Interleaving attack; Otway-Rees protocol; Security analysis

近年来，随着网络的普及和通信环境的日渐复杂，人们对信息安全的要求也越来越高。密码协议作为信息安全的重要基础，其安全性也越来越受到了重视。密码协议分析方法也是层出不穷，BAN逻辑[1]的出现标志着密码协议的分析技术进入了形式化分析阶段，形式化方法由于其精炼性、简洁性和无二义性，已经成了密码分析的主流方法。继BAN逻辑后，又相继出现了很多新的逻辑，如AT逻辑、SOV逻辑、GNY逻辑、MB逻辑等，它们在不同方面对BAN逻辑进行了改进，统称为BAN类逻辑。BAN类逻辑容易使用，表达力丰富，分析出了很多协议的缺陷和冗余。但是，BAN类逻辑也有很多不足之处[2]，其中一个就是它不能很好地分析重放攻击和并行攻击。为了弥补BAN类逻辑的这一不足，1997 年Gurges在文献[3]中提出了SG逻辑。

本文将介绍 SG 逻辑的语法语义及其对并行攻击的分析，重点对文献[2]中给出的改进版 Otway-Rees 协议进行分析，指出改进版的协议仍然存在容易受到并行攻击的缺陷，并给出了进一步的改进，最后推证了改进后的协议在 SG 逻辑分析下是安全的。

1 SG 逻辑

1.1 SG 逻辑的符号规定

P：参与方和参与方身份的集合；

$[P,\dots,Q]$：表示 P 的子集（用方括号是为了和协议参与方发送的密文相区别）；

$\{k;m\}$表示用密钥 k 对消息 m 加密或签名；

k^{-1}：表示解密密钥；

k_{AB}：A 和 B 的共享密钥；

(r,s)：协议第 r 次运行中的第 S 步；

$T(P,(r,s))$：表示(r, s)中接收到的所有的时间变量参数集合，比如 nonce、时戳；

$R(P,(r,s))$：表示在(r, s)中所有可被 P 识别的消息集合；

$G(P,(r,s))$：表示(r, s)以前所有的可能由 P 产生的消息；

$G(p,(r,s))$：表示(r, s)以前所有的可能由任何协议参与方产生的消息。

1.2 逻辑的语义

SG 逻辑的函数和谓词的语义规定如下：

$\varepsilon: k \to IP(P)$：函数 ε 把 k 映射到所有拥有密钥或者是在协议运行结束后将拥有密钥的参与方的集合，用它可以认证由 k^{-1}生成的密文（例如：用它验证签名）。

$\delta: k \to IP(P)$:函数 δ 把 k 影射到拥有密钥或者是在协议运行结束后将拥有密钥的参与方的集合，可以用它产生签名和认证码。

$in(P,V)$:表示 P 是 $\varepsilon(k)$ 或者 $\delta(k)$ 中的元素，P 可以根据自己的目的使用密钥 k。

$number(v,w)$: v, w ∈ IP(p)，并且 V、W 有相同的势。

$leq(x,y)$：x、y 都是协议的步骤，并且 x=y 或者步骤 y 在步骤 x 的后面。

$enc(k)$：k 是一个用于加密、数字签名或消息认证的密钥。

作者简介：王小锐(1979－)，男，硕士生，主研方向：密码协议与信息安全；陈连俊，博士、研究员；季庆光，博士、高工；曹正君，硕士、研究实习员

收稿日期：2006-03-30 **E-mail：**major791109@163.com

$sig(k)$：k 是一个密钥用来生成数字签名和消息认证码，解密时用 k^{-1}。

$sees(P,m,r,s)$：P 在(r, s)中接收到消息 m。

$has(p,m)$：在某一个时刻，P 接收到了 m 并且拥有了它。

$said(p,k,m,r,s)$:直到协议 r 的第 s 步为止，P 发送了{k；m}，但是没有接收它。

$not\_said(p,k,m,r,s)$:直到(r, s)为止，P 没有产生和{k；m}一样的消息 m′。

$recog(P,m,r,s)$：P 在(r, s)中可以识别 m。

$in\_time(p,m,r,s)$:P 在(r, s)接收到消息 m，并且 m 拥有一些永久的属性。

$\det erm(p,m)$:消息 m 由 P 决定（适用于服务器发放会话密钥时的情况）。

says(P, k, m):P 在一个可以接受的时间间隔内发送经过 k 处理过的 m。

$control(P,\varphi)$:P 对声明 φ 有控制权，φ 可以被信任。

$believes(p,\varphi)$：P 相信声明 φ。

1.3 推理规则

该推理规则主要应用于认证协议。因为篇幅关系，这里仅仅给出本文所用到的规则。

(1)PR 规则：（PR） $\dfrac{sees(p,\{k;m\},r,s)\wedge in(p,\delta(k^{-1})}{sees(p,m,r,s)}$

(2)MM1 规则：根据接收到的密文可以推导出消息的产生者

(MM1)

$$\frac{sees(p,\{k^{-1};m_1,\cdots,m_n\},r,s)\wedge has(p,(m_1,\cdots m_n)\wedge(recog(p,m_1,r,s)\vee\cdots\vee recog(p,m_n,r,s))\wedge in(p,\varepsilon(k))\wedge believes(p,\delta(k^{-1})=[p_1\cdots,\ p_n])}{believes(p,said(p_1,k^{-1},(m_1,\cdots,m_n),r,s)\vee\cdots\vee said(p_n,k^{-1},(m_1,\ldots,m_n),r,s)}$$

如果分析一个非对称密码协议，规则（MM1）足够得出结论说明私钥的拥有者产生了数字签名。然而，当采用一个对称密码协议时，规则（MM1）仅仅能推导出 $(p,\delta(k)=[P,Q])$，即 P 或者 Q 产生了密文。这种情况下，需要另外一种推理规则来证明密文不是 P 产生的。

（MM2） $\dfrac{not\_said(P,K,m,r,s)}{believes(P,\neg(said(P,k,m,r,s)))}$

(3)NV 规则（随机数验证规则）：如果 P 相信消息 m 是由 Q 产生的并且具体永久的属性，则它可以相信该消息产生于一个可以接受的时间间隔内：

(NV)

$$\frac{believes(P,said(Q,K,(m_1,...,m_n),r,s))\wedge(in\_time(P,m_1,r,s)\vee\ldots\vee in\_time(P,m_n,r,s)\vee in\_time(P,K,r,s)}{believes(P,says(Q,K,(m_1,\ldots,m_n)))}$$

(4)判定规则（J1）：如果 P 相信 Q（密钥服务器）关于声明 φ 是可以信任的，并且 P 相信该声明产生于一个可以接受的时间间隔，则 P 可以相信该声明 φ：

（J1）

$$\frac{believes(P,says(Q,K,\varphi))\wedge believes(P,controls(Q,\varphi))}{believes(P,\varphi)}$$

(5)下面的规则特别的指当协议参与方拥有一个特定的消息或者一个特定的密钥，当 P 用该会话密钥证明它的拥有时后者是必要的：

(H_2) $\dfrac{believes(p,says(Q,k,m))}{believes(p,has(Q,m))}$ (H_4) $\dfrac{sees(p,m,r,s)}{has(p,m)}$

(6)密钥规则（K1、K2）：协议参与方可以用任何它拥有的密钥：

(K_1) $\dfrac{has(p,k)\wedge enc(k)}{in(P,\varepsilon(K))}$

(K_2) $\dfrac{has(P,k)\wedge sig(K)}{in(P,\delta(K))\wedge believes(P,in(P,\delta(K)))}$

(K_3)

$$\frac{(believes(P,says(Q,k',(R,K)))\vee believes(P,says(Q,k',(K,R))))\wedge sig(k^{-1})}{believes(P,says(Q,k,in(R,\delta(k^{-1}))))}$$

(K_4)

$$\frac{believes(P,in(P_1,\delta(K))\wedge\ldots\wedge in(P_n,\delta(K)))\wedge number([P_1,....,p_n],\delta(K))}{believes(P,\delta(K)=[p_1,....,p_n])}$$

(7)合成和分解规则

(C) $\dfrac{has(P,m_1)\wedge has(P,m_2)}{has(P,(m_1,m_2))}$

（D1） $\dfrac{sees(P,(m_1,m_2),r,s)}{sees(P,m_1,r,s)\wedge sees(P,m_2,r,s)}$

(D2) $\dfrac{has(P,(m_1,m_2))}{has(P,m_1)\wedge has(P,m_2)}$

(D3) $\dfrac{believes(P,said(Q,k,(m_1,m_2),r,s)}{believes(P,said(Q,k,m_1,r,s)\wedge believes(P,said(Q,k,m_2,r,s))}$

（D4）

$$\frac{believes\ (P,says\ (Q,k,(m_1,m_2)))}{believes\ (P,says\ (Q,k,m_1))\wedge believes\ (P,says(Q,k,m_2))}$$

(S)

$$\frac{believes(P,\varphi_1\vee\ldots\vee\varphi_n)\wedge believes(P,\neg\varphi_1\wedge\ldots\wedge\neg\varphi_{i-1}\wedge\neg\varphi_{i+1}\wedge\ldots\wedge\neg\varphi_n}{believes(P,\varphi_i)}$$

2 改进版Otway-Rees协议的SG逻辑分析

文献[3]中给出的 Otway-Rees 协议改进如下：

A→B：M，A，B，{K_{AS}；M，N_A，A，B}

B→S：M，A，B，{K_{AS}；M，N_A，A，B}，N_B，{K_{BS}；M，A，B}

S→B：M，{K_{AS}；A，N_A，K_{AB}}，{K_{BS}；B，N_B，K_{AB}}

B→A：M，{K_{AS}；A，N_A，K_{AB}}

改进版的Otway-Rees协议用BAN逻辑方法和strand space方法分析都是安全的，但是经过手工推导证明[4]，它仍然存在缺陷，容易受到下面的攻击：

1′ $P_{(A)}\rightarrow B:M,A,B,\{k_{PS};M,N_P,P,B\}$

2′ $B\rightarrow P_{(s)}:M,A,B,\{k_{PS};M,N_P,P,B\}\ ,N_B,\{k_{BS};M,A,B\}$

2″ $P_{(B)}\rightarrow S:M,P,B,\{k_{PS};M,N_P,P,B\},N_B,\{k_{BS}\ ;M,P,B\}$

3′ $S\rightarrow B:M,\{k_{PS}\ ;P,N_P,k_{PB}\},\{k_{BS}\ ;B,N_B,K_{PB}\}$

4′ $B\rightarrow P_{(A)}:M,\{k_{Ps}\ ;P,N_P,k_{PB}\}$

用 SG 逻辑分析协议第 r 次运行时 B 的安全性，来证明这种并行攻击的存在性。首先确定 $T(B,(r,s))$、$R(B,(r,s))$ 和 $G(B,(r,s))$ 以及初始假设。由于 B 收到了消息 3，因此有

Sees（B，（M，{ k_{AS}；A，N_A，k_{AB}}，{k_{BS}；B，N_B，k_{AB}}），r，3) (1)

由于上面协议基于对称密码体制，因此有

$k_{BS}=k_{BS}^{-1}$ (2)

$k_{AB}=k_{AB}^{-1}$ (3)

sig(k_{AB}) (4)

number([P，Q]，$\delta(k_{AB})=[A,B]$) (5)

对称认证密钥K_{BS}是B和S共享的，所以有

in (B, $\delta(k_{BS}^{-1})$) (6)

in (B, $\varepsilon(k_{BS}^{-1})$ (7)

believes($B,\delta(k_{SB}^{-1})$) (8)

由于B在$(r,2)$中发送随机数N_B并在$(r,3)$中接收它，它是B判定消息新鲜度的一个参数，因此有

in _ time(B,N_B, r, 4) (9)

B 在$(r,3)$中接收到了消息 3，由于$\neg in(B,\delta(k_{AS}))$，并且 B

不能识别 k_{AB}，因此

$recog(B,(B,N_B),r,3)$ (10)

在做好所有的预备假设后，开始实际的分析。协议的目标之一是协议运行结束后，A 与 B 共享 k_{AB}，因此必须证明 $believes\ (B,\delta(k_{AB})=[A,B])$ 成立。

(1) $\xrightarrow{D1}$ $sees(B,\{k_{BS};B,N_B,k_{AB}\},r,3)$ (11)

(11)(6) $\xrightarrow{PR}$ $sees(B,(B,N_B,k_{AB}),r,3)$ (12)

(12) $\xrightarrow{D1}$ $sees(B,k_{AB},r,3)$ (13)

(13) $\xrightarrow{H4}$ $has(B,k_{AB})$ (14)

(14) (4) $\xrightarrow{k2}$ $believes(B,in(B,\delta(k_{AB})))$ (15)

现在为止，已经证明了 $believes\ (B,in(B,\delta(k_{AB})))$ 成立，只需再证明 $believes(B,in(A,\delta(k_{AB})))$ 成立即可。由⑨⑩可知，在 $(r,3)$ 中B没有获得关于对方身份的任何信息，仅通过验证 N_B 来确定消息的合法性，因而不能确定对方的真实身份。考虑协议并行运行的情况，假设在 r 之前B和P有一次由P发起的通信 $r_0\ (r_0<r)$，并且 r_0 中所用到的随机数 M_0 与 r 中用到的 M 完全相同（即 $M_0=M$，参见前面）。在 r_0 中，S通过发送（$M,\{k_{PS};p,N_P,k_{PB}\},\{k_{BS};B,N_B,k_{PB}\}$）来分发P和B的共享密钥 k_{PB}，由于（$M,\{k_{PS};P,N_P,k_{PB}\},\{k_{BS};B,N_B,k_{PB}\}$）$\in R(B,(r_0,3)\subset R(B,(r,3)$，对于B来说，（$M,\{k_{PS};p,N_P,k_{PB}\},\{k_{BS};B,N_B,k_{PB}\}$）在 $(r,3)$ 中是合法的。事实上，任意的消息（$M,\{k_{*S};*,N_*,k_{*B}\},\{k_{BS};B,N_B,k_{*B}\}$）（*代表任意的通信方身份名）对B 来说都是合法的。于是B把 k_{PB} 当成了它与A的共享密钥， P成功的欺骗了B.具体逻辑推理如下：

Sees（$p,(\{k_{BS};M,P,B\}),r_0,2)$ (16)

(16) $\xrightarrow{H4}$ $has(p,(\{k_{BS};M,P,B\}))$ (17)

由于知识具有继承性，在 r 中(17)式仍然成立，在(r，2)后，P 发送消息(M,P,B,$\{k_{PS};M,N_P,P,B\}$, N_B,$\{k_{BS};M,P,B\}$）给S，有

$sees$ (S,($M,P,B,\{k_{PS};M,N_P,P,B\},N_B,\{k_{BS};M,P,B\}$), r, 3) (18)

再由初始条件：

$in(S,\delta(k_{BS}^{-1}))$ (19)

$in(S,\delta(k_{ps}^{-1}))$ (20)

$believes(s,\delta(k_{BS}^{-1})=[B,s])$ (21)

$believes(s,\delta(k_{ps}^{-1})=[P,S])$ (22)

$recog(s,(P,B),r,4)$ (23)

继续推导有

((18) (19) (20) (21) (22) (23)) $\xrightarrow{MM1}$ $believes(s,said(p,\{k_{PS};M,N_P,p,B\}\wedge said(B,(\{k_{BS};M,P,B\})$ (24)

(19)(20)(24) $\xrightarrow{D3}$ $believes(s,said(p,k_{ps},(P,B))\wedge said(B,k_{BS},(P,B)))$ (25)

由(23) (25)及密钥服务器 S 的授权性可知，S 认为在 r 中它负责给 P 和 B 分发密钥，所以有

believes($s,\delta(k_{PB})=[P,B]$, r, 4) (26)

再由 $believes(B,controls(S,in(P,\delta(k))))$ 可知，B 只能在式(26)中推导 r 中 S 发放密钥的可信性，但 B 并不能在式(26)中得出 $in\ (A,\delta\ (k_{AB}))$，从而在理论上证明了协议缺陷的存在性。

3 对 Otway-Rees 协议的进一步改进

针对上面的缺陷，给出协议的下列改进：

(1)A→B：M，A，B，{K_{AS}；M, N_A，A，B}

(2)B→S：M，A，B，{K_{AS}；M，N_A，A，B}，N_B，{K_{BS}；M，A，B}

(3)S→B：M，{K_{AS}；B，N_A，K_{AB}}，{K_{BS}；A，N_B，K_{AB}}

(4)B→A：M，{K_{BS}；B，N_A，K_{AB}}

对于改进后的协议，显然有 recog($B,(A,N_B),r,3$)成立，由于 B 看到了 A 的身份标识，因此(r_0,3)中 S 发送给 B 的消息在(r,3)中不再是合法的消息，P 不能再伪装成 A 了。继续前面的分析：

$believes(B,says(s,(M,\{k_{AS};B,N_A,k_{AB}\},\{K_{BS};A,N_B,k_{AB}\})),r,3)$ (27)

(27) $\xrightarrow{D1}$ $believes(B,says(s,(\{k_{BS};A,N_B,k_{AB}\})),r,3)$ (28)

(28) $\xrightarrow{D4}$ $believes(B,says(s,(\{K_{BS};A,k_{AB}\}),r,3)$ (29)

(29) (4) $\xrightarrow{k3}$ $believes(B,says(s,K_{BS},in(A,\delta(k_{AB}))))$ (30)

由于 B 对密钥服务器 S 总是相信的，因此有

$believes(B,control(S,in(A,\delta(k_{AB}))))$ (31)

(30) (31) $\xrightarrow{J1}$ $believes(B,in(A,\delta(k_{AB})))$ (32)

(15) (5) (32) $\xrightarrow{k4}$ believes($B,\delta(k_{AB})=[A,B]$) (33)

已经证得了改进后的协议在该分析方法下的完备性；由此可知，改进后的协议可以抵抗文献[4]中提出的并行攻击。

4 结束语

本文给出了 SG 逻辑的详细语法和语义。运用 SG 逻辑，我们找出了 Otway-Rees 协议改进版的缺陷，并给出了进一步的改进，指出了我们给出的改进可以抵抗并行攻击。

参考文献

1 Burrows M, Abadi M. A Logic of Authentication[J]. ACM Transactions on Computer System, 1990, 8(1): 1-50.

2 GÜrgens S. A Formal Analytics Technique for Authentication Protocol[C]//Proceedings of Pragocryt. 1996: 159-176.

3 Boyd C, Mao W. On a Limitation of Ban Logic[C]//Ted H. Advances in Cryptology, EUROCRYPT`93. Berlin: Springer-verlag, 1993: 240-247.

4 李　军, 何大可. 密码协议的健全性分析[J]. 通信技术, 2002, (2).

(上接第 125 页)

4 Microsoft Corporation. Microsoft Windows 2000 Driver Development Kit, Volume 1: Design Guide[M]. Microsoft Press, 2001-08: 708-720.

5 Microsoft Corporation. Microsoft Windows 2000 Driver Development Kit, Volume 2: Reference[M]. Microsoft Press, 2001-08: 234-240.

6 郭益昆. VC++.NET 开发驱动程序详解[M]. 北京: 北京希望电子出版社, 2002-04: 2-40.

7 Walnum C. Windows 2000 Programming Secrets[M]. IDG Books Worldwide Press, 2002-01: 203-204.

8 Rajagopal R, Monica S. Windows 2000 程序设计[M]. 天宏工作室, 译. 北京: 清华大学出版社, 2002-06: 201-202.

9 Malik D S. C++ Programming Program Design Including Data Structures[M]. 北京: 电子工业出版社, 2003-06.

10 汪晓平, 钟　军. Visual C++ 网络通信协议分析与应用实现[M]. 北京: 人民邮电出版社, 2003-02: 12-18.