Needham-Schroeder 公钥协议的模型检测分析*

张玉清[1,2] 王磊[2] 肖国镇[2] 吴建平[1]

[1](清华大学信息网络工程研究中心 北京 100084)

[2](西安电子科技大学信息保密研究所 西安 710071)

E-mail: zhangyq@cernet.edu.cn

摘要 密码协议安全性的分析是当前网络安全研究领域的一个世界性难题.提出了运用模型检测工具SMV(symbolic model verifier)分析密码协议的方法,并对著名的Needham-Schroeder(NS)公钥协议进行了分析.分析结果表明,入侵者可以轻松地对NS公钥协议进行有效攻击,而这个攻击是BAN逻辑分析所没有发现过的.同时,给出了经SMV分析过的一个安全的NS公钥协议的改进版本.

关键词 模型检测,密码协议,形式方法.

中图法分类号 TP311

密码协议目前已广泛应用于计算机网络与分布式系统中,借助于密码算法来达到密钥分配、身份认证等目的.但密码协议安全性的论证仍是一个悬而未决的问题.90年代以来,密码协议的形式化分析成为国际上的研究热点.这种方法的出发点是希望将密码协议形式化,而后借助于人工推导,甚至计算机的辅助分析,来判别密码协议是否安全可靠.

早先的密码形式分析方法采用基于信仰的逻辑——BAN(Burrows,Abadi and Needham)逻辑[1]来说明和验证密码协议.随后,学者们经过研究发现,BAN逻辑在理想化步骤等方面存在着不可逾越的障碍[2,3].自1997年起,计算机科学家及密码学家开始陆续尝试用模型检测(model checking)这种新的形式方法来分析密码协议的安全性[4~7],我们正是基于这种背景开始了这方面的研究.本文将启用SMV(symbolic model verifier)[8]这种新的模型检测工具软件来分析密码协议,并以著名的Needham-Schroeder(简称NS)公钥协议[9]作为分析研究的对象.

1 Needham-Schroeder 公钥协议

Needham-Schroeder公钥协议按功能划分为两个部分:获取公开密钥和双方身份认证.这里我们研究其身份认证部分,协议为:

消息1. $A \rightarrow B: \{Na, A\}_{Kb}$.

消息2. $B \rightarrow A: \{Na, Nb\}_{Ka}$.

消息3. $A \rightarrow B: \{Nb\}_{Kb}$.

协议的参与者只有两个:主体A和主体B,其中A作为NS公钥协议的初始者,B为响应者.整个协议采用公开密钥系统,Ka,Kb分别是A和B的公开密钥,Na,Nb是A和B发布的具有新鲜性的随机数(也称临时值,nonce).协议的运行过程为:主体A向主体B发送包含Na和自己身份的消息1,并用B的公钥Kb加密消息1;

* 本文研究得到国家自然科学基金(No.69673025)资助.作者张玉清,1966年生,博士,讲师,主要研究领域为网络安全,电子商务,软件工程.王磊,1972年生,硕士,主要研究领域为网络安全,软件工程,密码算法.肖国镇,1934年生,教授,博士生导师,主要研究领域为网络安全,密码理论,编码理论.吴建平,1953年生,教授,博士生导师,主要研究领域为计算机网络体系结构,网络协议测试.

本文通讯联系人:张玉清,北京 100084,清华大学信息网络工程研究中心

本文1999-01-22收到原稿,1999-09-07收到修改稿

B 收到并解密消息 1 后按协议要求向 A 发送用 A 的公钥 Ka 加密的内含 Na 和 Nb 的消息 2;在协议最后,A 向 B 发送经 Kb 加密的 Nb. 经过这样一次协议的运行,主体 A 和 B 就建立了一个它们之间的共享秘密 Nb,这个共享秘密可为以后他们进行秘密通信确认双方身份时使用.

2 密码协议形式分析前提

(1) 完善保密(perfect encryption)前提,即协议采用的密码算法是完善保密的.

(2) 参与协议运行的主体除了诚实的合法用户外,还有不怀好意的入侵者.

(3) 入侵者的知识与能力

① 知道参与协议运行的各主体名及其公钥,并拥有自己的加密密钥和解密密钥;

② 可窃听或中途拦截系统中传送的任何消息,增加自己的知识或在系统中可插入新的消息,并可运用他知道的所有知识;

③ 即使不知道加密部分的内容,也可重放他所看到的任何消息(其中可改变明文部分).

3 NS 公钥协议的 SMV 分析

3.1 NS 公钥协议有限状态系统模型

众所周知,模型检测技术用于有限状态系统. 为此,我们为 NS 公钥协议构造了如下的有限状态系统模型:

参与协议运行的主体集合:{初始者 A,响应者 B,入侵者 I}.

其中 A 和 B 是诚实的主体,他们将严格按照初始者和响应者的身份参与协议运行,并只运行一次 NS 协议,而入侵者 I 则不受此限制.

有关集合为:

(1) 初始者集合:$\{A,I\}$;

(2) 响应者集合:$\{B,I\}$;

(3) 密钥集合:$\{Ka,Kb,Ki\}$;

(4) 临时值集合:$\{Na,Nb,Ni\}$.

根据初始者和响应者的组合,NS 公钥协议有以下两种运行情况:

(1) $A\longleftrightarrow B$	A 和 B 运行一次 NS 协议,其中 A 作为初始者,B 作为响应者;
$I\longleftrightarrow I$	I 和 I 自己运行一次 NS 协议,无实际意义.
(2) $A\longleftrightarrow I$	A 和 I 运行一次 NS 协议,其中 A 作为初始者,I 作为响应者;
$I\longleftrightarrow B$ 或 $I(A)\longleftrightarrow B$	I 或 I 冒充 A 和 B 运行一次 NS 协议,其中 I 或 $I(A)$ 作为初始者,B 作为响应者.

3.2 分析框架

根据入侵者的知识与能力,我们将入侵者 I 置于 A 和 B 之间,系统中传送的任何消息,I 都可以依据自己的需要拦截或转发,分析框架如图1所示.

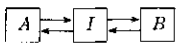


Fig. 1 Analysis frame of Needham-Schroeder public-key protocol

图1 Needham-Schroeder公钥协议分析框架

3.3 SMV 分析软件工作原理

SMV 分析软件以有限状态系统说明及其系统属性作为输入,若有限状态系统具有给定的属性,则输出真(true),否则输出假(false),并同时显示系统不满足属性的反例. SMV 工作原理如图2所示,其中系统说明部分用 SMV 分析软件规定的 SMV 语言编程,系统属性部分则用 CTL(computation tree logic)逻辑表达.

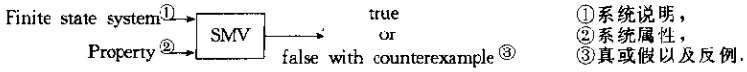


Fig. 2 Model checker SMV software
图2 SMV分析软件工作原理示意图

3.4 协议消息描述

为了描述NS协议中的消息,我们构造了记录类型message. Message各域为:

mtype:{none,msg1,msg2,msg3};消息的类型,msg1,msg2和msg3分别对应于NS协议中的消息1、2和3,none表示无消息.

source:{A,B,I};	消息的发出者
dest:{A,B,I};	消息的接受者
key:{Ka,Kb,Ki};	加密消息所用的密钥
data1:{Na,Nb,Ni,none};	消息的第1个数据取值集合,none表示无数据
data2:{Nb,Ni,none,A,I};	消息的第2个数据取值集合,none表示无数据对于消息1,data2可取值A或I;消息2,data2可取值Nb或Ni;消息3,data2取值none.

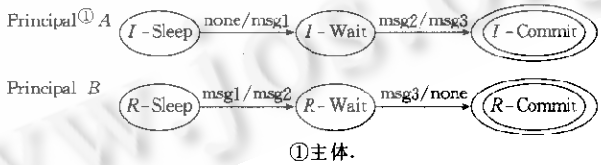
3.5 协议的系统说明及系统属性

(1) NS有限状态系统的SMV说明

在用SMV语言编程的NS协议有限状态系统说明中,我们用并发方式的MODULE来表示初始者A、响应者B和入侵者I. SMV程序中的有关说明为:

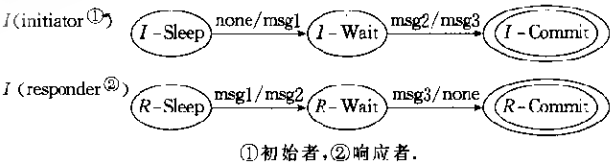
A:initiator($I.outMA$);	初始者,NS协议中的A,输入为$I.outMA$
B:responder($I.outMB$);	响应者,NS协议中的B,输入为$I.outMB$
I:intruder($A.outM$,$B.outM$);	入侵者I,输入为$A.outM$,$B.outM$

NS协议有限状态系统的状态转换图如图3和图4所示. 图3是主体A和B的状态转换图,A,B的状态集合分别为{I_Sleep,I_Wait,I_Commit}和{R_Sleep,R_Wait,R_Commit},状态集合中的3个状态对应于NS协议3个消息的传送. 对于主体A来讲,它从I_Sleep状态开始,首先选择协议运行的响应者,而后发送(输出)消息1(msg1)到达I_Wait状态,在此等待接受(输入)响应者的消息2,(msg2),一旦有符合协议要求的消息2,将自动发送消息3(msg3),并到达协议结束状态I_Commit. 主体B与A类似,值得说明的是:主体B在接受消息1时,将通过message记录的data2域识别协议的初始者,并记为$B.ini$. 这里,我们用message记录的mtype域值代表整个消息.



①主体.

Fig. 3 State transition graphs of principal A and B
图3 主体A和B的状态转换图



①初始者,②响应者.

Fig. 4 State transition graphs of intruder I as an initiator and a responder
图4 入侵者I作为初始者和响应者身份的状态转换图

入侵者I的状态转换最为复杂,图4说明了入侵者I作为初始者和响应者身份的状态转换,图中略去了入侵者I只是接受和转发消息的状态转换.

入侵者 I 的初始情况:

① 知识集合:$\{A,B,I,Ki,Ki^{-1},Ni\}$;

② 窃听的消息1、消息2和消息3的消息集合 $Set1=\{\}$,$Set2=\{\}$,$Set3=\{\}$;

③ 是否知道 Na,Nb 的布尔变量 $Know\_Na=0$,$Know\_Nb=0$;

入侵者 I 的知识获取工作过程:

① 将窃听的由 A,B 发出的且密钥为 Ki 的消息解密,若可获取 Na 或 Nb,则置 $Know\_Na=1$ 或 $Know\_Nb=1$,并将 Na 或 Nb 加入知识集合;

② 将窃听的由 A,B 发出的消息(密钥为 Ki 的除外)加入相应的 $Set1$,$Set2$,$Set3$ 集合.

入侵者 I 的冒充过程:

发送消息时进行冒充,I 作为初始者时,可冒充 A 的身份,以 $I(A)$ 参与协议运行;I 作为响应者时,可冒充 B 的身份,以 $I(B)$ 参与协议运行.I 在冒充时,根据知识集合可在发出的消息中运用 Na 或 Nb,并可重放 $Set1$,$Set2$,$Set3$ 中窃听的消息.一旦有冒充,置布尔变量 $IA=1$(即 $I.IA$,记录 I 是否冒充了 A)或布尔变量 $IB=1$(即 $I.IB$,记录 I 是否冒充了 B).

(2) NS 协议有限状态系统属性

NS 协议有限状态系统属性(即 NS 协议的安全性质)的 CTL 表示为

$$AG((A.state = I\_\text{ sleep}) \rightarrow AF((A.\text{state} = I\_\text{ commit})\&(B.state = R\_\text{ commit})\&(B.ini = A)\&(I.IA = 0)\&(I.IB = 0))).$$

其中 A,G,F,$\rightarrow$,$\&$ 都是 CTL 的符号,其含义分别为:AG 表示所有路径的所有状态;AF 表示所有路径的最后状态;$\rightarrow$ 是逻辑蕴含;$\&$ 是逻辑与.

整个 CTL 逻辑表达式的含义为:初始者由初始状态 $I_$sleep 开始运行 NS 公钥协议,最终到达状态:初始者 A 和响应者 B 按照各自的身份参与并运行完一个完整的 NS 公钥协议,到达各自的结束状态 $I_$commit 和 R commit,响应者 B 认为与他通信的初始者是 A,同时,在协议运行过程中,入侵者 I 既没有冒充初始者($I.IA=0$),也没有冒充响应者($I.IB=0$).

3.6 对 NS 公钥协议的攻击

将 SMV 编程的 NS 协议系统说明及协议系统属性作为 SMV 分析软件的输入并运行后发现,NS 公钥协议并不满足其安全性质.分析造成系统输出 false 的反例发现:入侵者 I 可以通过两次运行 NS 公钥协议来进行攻击,这个攻击打破了协议的安全性质,同时这个攻击是 BAN 逻辑分析所没有发现过的[1].攻击如下:

第1次 NS 公钥协议运行:

消息1.1. $A\rightarrow I:\{Na,A\}_{Ki}$.

此时,入侵者 I 开始第2次 NS 协议运行:

消息2.1. $I(A)\rightarrow B:\{Na,A\}_{Kb}$.

消息2.2. $B\rightarrow I(A):\{Na,Nb\}_{Ka}$.

消息1.2. $I\rightarrow A:\{Na,Nb\}_{Ka}$.

消息1.3. $A\rightarrow I:\{Nb\}_{Ki}$.

消息2.3. $I(A)\rightarrow B:\{Nb\}_{Kb}$.

入侵者 I 通过解密消息1.1和消息1.3获取发送消息2.1和消息2.3所需的 Na 和 Nb,消息1.2则是消息2.2的重放.上述协议运行完,主体 B 认为他与 A 共享秘密 Nb,实际上他与 I 共享 Nb,I 假冒 A 成功,攻击有效.对于网络系统中任何一个合法用户,只要接收到发给自己的 NS 公钥协议消息1,就可以发起上述攻击,以欺骗另外一个用户,故 NS 公钥协议是不安全的.

3.7 改进的 NS 公钥协议

对于 NS 公钥协议的缺陷,我们进行了改进,在消息2中增加了发送者的身份,改进的 NS 公钥协议重新进行 SMV 分析,没有发现新的攻击.NS 公钥协议的改进版本为:

消息1. $A\rightarrow B:\{Na,A\}_{Kb}$.

消息2. $B \rightarrow A: \{Na, Nb, B\}_{Ka}$.

消息3. $A \rightarrow B: \{Nb\}_{Kb}$.

4 结 论

本文运用模型检测工具SMV对NS公钥协议进行了成功的分析,并找到了BAN逻辑分析所没有发现过的攻击.我们下一步的研究方向是运用SMV对更多的密码协议进行分析,以求在广泛分析的基础上,建立密码协议设计的新准则,促进密码协议设计的发展.

致谢 英国Leicester大学的Gavin Lowe博士和牛津大学的A. W. Roscoe教授及时地为我们提供了最新的论文,美国Carnegie Mellon大学计算机学院和K. L. McMillan博士为我们提供了SMV软件1998年的最新版本,中国科学院软件研究所信息安全技术工程研究中心卿斯汉研究员为我们提供了最早的BAN逻辑文章,在此表示由衷的感谢.同时,也感谢上海贝尔实验室田建波博士后和西安电子科技大学的郑东博士所给予的帮助及有益的讨论.

参考文献

1 Burrows M, Abadi M, Needham R. A logic of authentication. ACM Transactions on Computer Systems, 1990,8(1):18~36

2 Boyd C, Mao W. On a limitation of BAN logic. In: Helleseth T ed. Advances in Cryptology——EUROCRYPT'93. Berlin: Springer-Verlag, 1993. 240~247

3 Zhang Yu-qing, Li Ji-hong, Xiao Guo-zhen. BAN logic for cryptographic protocols analysis and its limitations. Journal of Xidian University, 1999,26(3):376~378
(张玉清,李继红,肖国镇.密码协议分析工具——BAN逻辑及其缺陷.西安电子科技大学学报,1999,26(3):376~378)

4 Dang Z, Kemmerer R. Using the ASTRAL model checker for cryptographic protocol analysis. In: DIMACS Workshop on Design and Formal Verification of Security Protocols. 1997. http://dimacs.rutgers.edu/Workshops/Security/program2/program.html

5 Lowe G, Roscoe A. Using CSP to detect errors in the TMN protocol. IEEE Transactions on Software Engineering, 1997, 23(10):659~669

6 Marrero W, Clarke E, Jha S. A model checker for authentication protocols. In: DIMACS Workshop on Design and Formal Verification of Security Protocols. 1997. http://dimacs.rutgers.edu/Workshops/Security/program2/program.html

7 Mitchell J, Mitchell M, Stern U. Automated analysis of cryptographic protocols using Murφ. In: Storms P ed. Proceedings of the IEEE Symposium on Security and Privacy. Los Alamitos, CA: IEEE Computer Society Press, 1997. 141~151

8 SMV. http://www.cs.cmu.edu/~modelcheck/

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

Model Checking Analysis of Needham-Schroeder Public-Key Protocol

ZHANG Yu-qing[1,2] WANG Lei[2] XIAO Guo-zhen[2] WU Jian-ping[1]

[1](*Network Research Center Tsinghua University Beijing* 100084)

[2](*Information Security Institute Xidian University Xi'an* 710071)

Abstract It is an important and hard problem in the area of computer network security to analyze cryptographic protocols. A methodology is presented using a model checker of formal methods, SMV (symbolic model verifier), to analyze the well known Needham-Schroeder Public-Key Protocol. The SMV is used to discover an attack upon the protocol, which has never been discovered by BAN logic. Finally, the protocol is adapted, and then the SMV is used to show that the new protocol is secure.

Key words Model checking, cryptographic protocol, formal method.