DOI :10. 3969/J. ISSN. 1672-6693. 2009. 04. 016

基于 UPPAAL 的 NS 密码协议模型检测分析*

李忠慧[1,2], 张广泉[1]
(1. 重庆师范大学 数学与计算机科学学院, 重庆 400047 ;2. 盐城师范学院 信息科学与技术学院, 江苏 盐城 224002)

摘要 :形式化分析方法是目前密码协议分析的主流方法。然而,典型的密码协议形式化验证方法没有考虑时间因素,这个选择使得分析简单化。本文提出了运用基于时间自动机的模型检测工具 UPPAAL 分析密码协议的方法,并对著名的 Needham-Schroeder 协议(简称 NS 协议)的简单版本进行了分析。在对 NS 协议进行建模时,考虑消息实际传输需花费时间,引入消息的时间信息,从而构建 NS 协议的时间自动机模型。该方法利用 UPPAAL 的检查引擎所用的先进技术使其克服了一般时间自动机应用存在的状态空间爆炸问题。实验结果 UPPAAL 给出了 NS 协议认证失败的一种可能之一。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。从而证明 UPPAAL 工具可以成功检测出 NS 协议的缺陷。

关键词 :UPPAAL ;密码协议 ;模型检测
中图分类号 :TP311 文献标识码 :A 文章编号 :1672-6693(2009)04-0078-04

随着 Internet 的飞速发展,计算机网络已经渗透到人们生活的各个领域,网络中数据传输和交易的安全都需要密码协议的保障和支持。网络系统的并发性、异步性和多样性,使得人们不可能用直觉方法设计出高质量的协议,协议的安全性难以保证[1]。密码协议形式化分析的出发点是将协议形式化,然后借助人工推导或计算机辅助分析来判定密码协议是否可靠。密码协议形式化分析由协议系统的数学或者逻辑模型、系统规约以及一个有效的用来判定该系统是否满足其需求的证明程序组成。从使用的方法思想上可以将安全协议分析方法分为 3 类 :逻辑推理、模型检测和定理证明[2]。本文将利用 UPPAAL 这种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)公钥协议作为分析研究的对象。

1 UPPAAL 简介

自动验证工具 UPPAAL 主要采用一组带有整型变量的时间自动机对实时系统的行为进行模拟、对它的性质进行验证。UPPAAL 采用的模型验证机制可以避免状态空间爆炸问题,它已经被广泛应用于算法分析和协议验证方面。

UPPAAL 的用户界面包括 3 个主要部分 :一个系统编辑器、一个模拟器和一个验证器。系统编辑器用于创建和编辑要分析的系统。模拟器是一个确认工具,它用于检查所建系统模型可能的执行是否有错,以此在验证前发现一些错误。验证器通过快速搜索系统的状态空间来检查时钟约束和反应限制性质,它还为系统要求的规范和文件提供了一个需求规范编辑器。

UPPAAL 中使用的模型是时间自动机,并对时间自动机进行了一些扩展。用户可以声明一般值变量、全局时钟和用于同步的管道(channel)。UPPAAL 中存在两种形式的约束(g) :时钟约束(gc)和数据约束(gd)。

$$g ::= gc \mid gd \mid g, g$$
$$gc ::= x < n \mid x < y + n$$
$$gd ::= Expr \ \text{op} \ Expr$$
$$< \in \{<, <=, =, >, >=\}$$
$$\text{op} \in \{<, <=, =, >, >=, !=\}$$
$$Expr ::= i \mid i[Expr] \mid -Expr \mid Expr + Expr \mid Expr - Expr \mid Expr * Expr \mid Expr / Expr \mid (gd\ ?\ Expr : Expr)$$

此外,UPPAAL 中除了常规的控制位置外还有起始位置、紧迫位置和约束位置。通道主要用于保

* 收稿日期 :2009-04-01
资助项目 :重庆市自然科学基金项目(No. 2006BB2259)
作者简介 :李忠慧,女,硕士研究生,研究方向为形式化方法 ;通讯作者 :张广泉,E-mail :zgqxyz@ sina. com

证两个或多个进程模板的同步通信和相互操作,这可以通过在不同进程模板的位置转移上标注互补的同步动作得以实现。除了常规的通道外,通道还可以被声明为紧急通道和广播通道。

UPPAAL 使用的规范验证语言形如:$Prop ::= A[]p \mid E<>p \mid E[]p \mid A<>p \mid p \rightarrow q$。其中,$E<>p$ 表示 Possible,$E<>p$ 为真,当且仅当在时间自动机中存在一个序列 $s_0 \rightarrow s_1 \rightarrow \cdots \rightarrow s_n$,使得 s_0 是开始状态,s_n 是 p。$A[]p$ 表示 Invariantly,等价于 not $E<>$not p。$E[]p$ 表示 Potentially always,在时间自动机中,$E[]p$ 为真,当且仅当存在一个序列 $s_0 \rightarrow s_1 \rightarrow \cdots \rightarrow s_i \rightarrow \cdots$,使得 p 在所有状态 s_i 中得以满足,并且这个序列是无穷的或者在状态(l_n, v_n)终止。$A<>p$ 表示 Eventually,等价于 not E [] not p。$p \rightarrow q$ 表示 Lead to,等价于 $A[](p$ imply $A<>q)$。

2 NS 协议的 UPPAAL 分析

2.1 NS 公钥协议

Needham-Schroeder 公钥协议[3-5] 按功能划分为两个部分 获取公开密钥和双方身份认证。这里研究其身份认证部分,协议为:

消息1 $A \rightarrow B: \{N_a, A\} K_b$

消息2 $B \rightarrow A: \{N_a, N_b\} K_a$

消息3 $A \rightarrow B: \{N_b\} K_b$

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,K_a,K_b 分别是 A 和 B 的公开密钥。N_a,N_b 是 A 和 B 发布的具有新鲜性的随机数(也称临时值)。协议的运行过程为:主体 A 向主体 B 发送包含 N_a 和自己身份的消息1,并用 B 的公钥 K_b 加密消息1;B 收到并解密消息1后按协议要求向 A 发送用 A 的公钥 K_a 加密的内含 N_a 和 N_b 的消息2 在协议最后,A 向 B 发送经 K_b 加密的 N_b。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 N_b,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。

2.2 NS 协议的形式化建模

在不影响对协议关键性质检测的前提下,模型中对公钥协议做如下假设。假设网络系统中只有 A、B 和 I 三者,其中 A 是初始者,B 是响应者,I 是入侵者。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,而入侵者不受此限制。

NS 协议有限状态系统的状态转换图如图1所示。图1中 a、b、c 分别是主体 A、B 和入侵者 I 的状态转换图,A 和 B 的状态集合分别为 {I_create, I_send, I_receive, I_commit} 和 {R_create, R_send, R_receive, R_commit}。对于主体 A 来讲,它从 I_create 状态开始,生成消息,而后发送消息1到达接收状态,在此等待接收响应者的消息2,一旦有符合协议要求的消息2,将自动发送消息3,并达到结束状态。

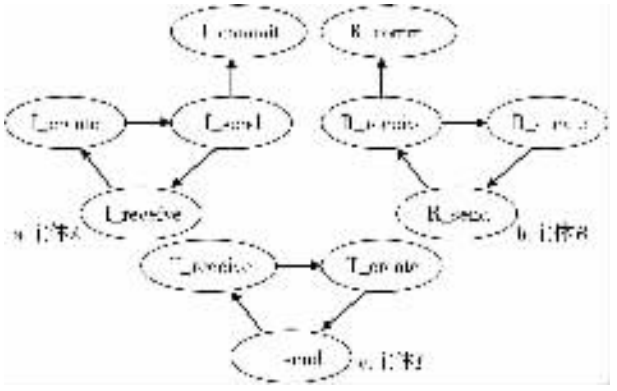

图1 NS 协议有限状态系统的状态转换图

用 UPPAAL 中的自动机[6] 网络表示协议模型系统,用自动机表示进程主体,用同步模型表示模型中的收发消息动作。为了描述 NS 协议中的消息,笔者构造了结构类型 network。network 各域为 {msg, recpt, key, data1, data2}。msg 为消息的类型,取值为消息1、2 和 3。recpt 为消息的接收者,data1 和 data2 分别存放两个临时值。定义布尔变量 IA 和 IB。当 I 冒充 A 给 B 发送消息时 IA = true,当 I 冒充 B 给 A 发送消息时 IB = true。该 NS 协议有限状态系统在 UPPAAL 中的具体描述如图2、图3、图4 所示。

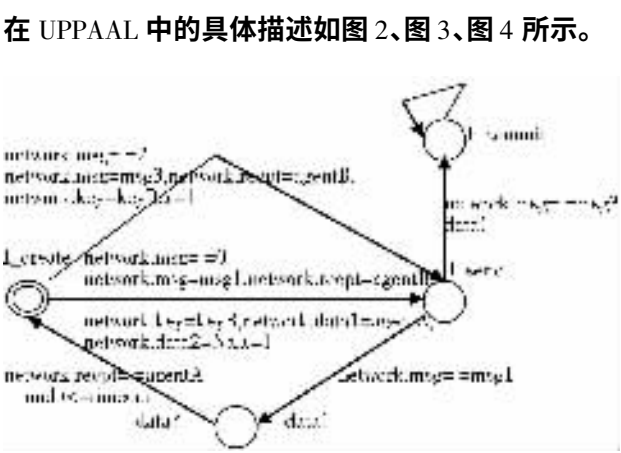

图2 协议发起者的状态转换图

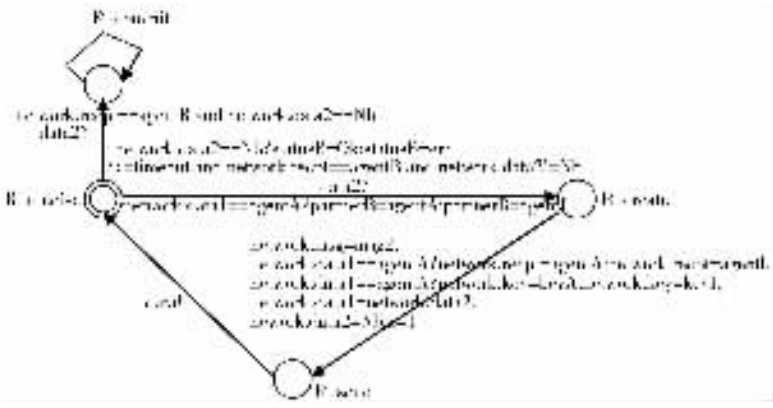

图 3　协议响应者的状态转换图

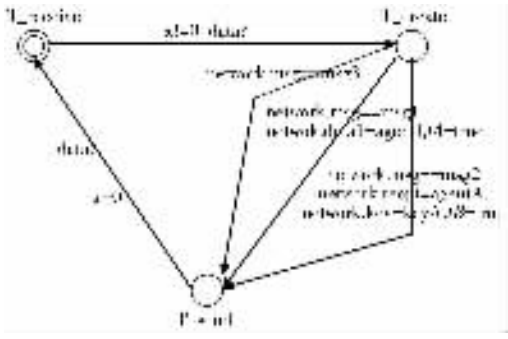

图 4　网络中入侵者的状态转换图

3 协议检测结果分析

初始者 *A* 由初始状态 I_create 开始运行 NS 公钥协议,最终达到状态:初始者 *A* 和响应者 *B* 按照各自的身份参与并行运行完成一个完整的 NS 公钥协议,达到各自的结束状态 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 *A*,也没有冒充响应者。所以,NS 协议有限状态系统属性的 CTL 表示为

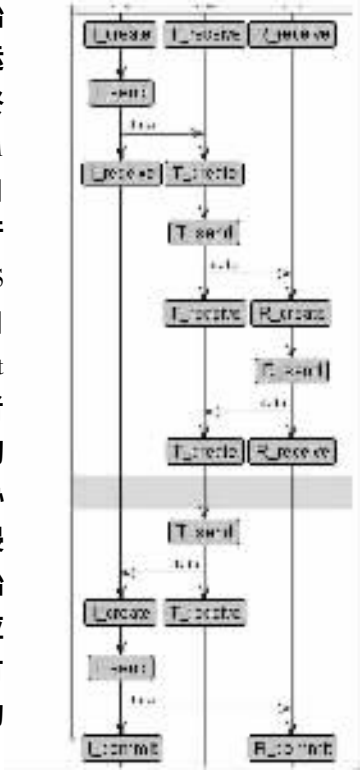

图 5　系统的消息序列图

A < > (P0. I_commit and P1. R_commit and (partnerB = = agentA) and (*IA* = = false) and (*IB* = = false))。

利用 UPPAAL 工具中模拟器可以显示自动验证器产生的执行过程,图 5 就是系统运行的一个序列图。自动验证器检查所建模[7]的系统应满足属性的 CTL 公式,并给出结论。实验中对上述 CTL 公式进行了检查,得出 Property is not satisfied,表明 NS 协议存在漏洞。

4 结语

本文在对 NS 协议分析中,运用了模型检测[8-9]工具 UPPAAL,清楚地指出了 NS 协议中存在漏洞。在对协议进行形式化分析时,利用模型检测工具可以生成系统的模拟运行路径。

参考文献:

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

[2] 王汝传. 密码协议形式化分析方法研究[J]. 信息网络安全,2005,55(7):24-26.

[3] 张玉清,王磊,肖国镇,等. Needham-Schroeder 公钥协议的模型检测分析[J]. 软件学报,2000,11(10):1348-1352.

[4] 陈道喜,张广泉,陈冬火. NSPK 协议的 Spin 模型检测[J]. 微电子学与计算机,2008,25(10):58-60,64.

[5] 龙士工,王巧丽,李祥. 密码协议的 Promela 语言建模及分析[J]. 计算机应用,2005,25(7):1548-1550.

[6] 古天龙. 软件开发的形式化方法[M]. 北京:高等教育出版社,2005.

[7] 张广泉,杨敬中. 面向方面的软件体系结构建模研究[J]. 重庆师范大学学报(自然科学版),2008,25(1):1-6.

[8] 林惠民,张文辉. 模型检测:理论、方法与应用[J]. 电子学报,2002,30(12):1907-1912.

[9] 刘锋,李军舟,李梦君,等. 基于 SMV 的安全协议模型检验[J]. 计算机工程与科学,2004,26(2):28-31,62.

Model Checking of the NS Cryptographic Protocol and Analysis Based on the UPPAAL

LI Zhong-hui[1,2], *ZHANG Guang-quan*[1]

(1. College of Mathematics & Computer Science, Chongqing Normal University, Chongqing 400047;
2. College of Information Science and Technology, Yancheng Teachers University, Yancheng Jiangsu 224002, China)

Abstract: Formal analysis methods are currently the mainstream method of cryptographic protocol analysis. Typically, the methods for formal verification of security protocols do not take time into account, and this choice also simplifies the analysis. A methodology is presented here by using a model checker of formal methods based on timed automaton, UPPAAL, to analyze a simplified version of the well known Needham-Schroeder Public-Key Protocol (NS for short). Since the actual sending of the message takes time, timeliness of the message is introduced when modeling the NS protocol. Thus timed automaton for the NS protocol is obtained. Because the check engine of the UPPAAL adopts advanced technology, this methodology can void the state space explosion problem arisen in the general timed automaton application. One of the possible forms of NS protocol authentication failure is presented in UPPAAL tool. This experimental result indicates an intruder's attack upon the NS public-key protocol. Therefore the UPPAAL could find the flaw of the NS protocol.

Key words: UPPAAL; cryptographic protocol; model checking

(责任编辑 游中胜)