文章编号:1001 - 9081(2005)07 - 1548 - 03

密码协议的 Promela 语言建模及分析

龙士工,王巧丽,李 祥
(贵州大学 计算机软件与理论研究所,贵州 贵阳 550025)
(ie. sglong@ guz. edu. cn)

摘 要:给出了利用 SPIN 模型检测分析密码协议的一般方法。作为一个实例,对 Needham-Schroeder 公钥密码协议用 Promela 语言建模,并利用 SPIN 进行了分析验证,发现了其安全漏洞。该方法很容易推广到有多个主体参与的密码协议的分析。

关键词:密码协议; 模型检测;SPIN;Promela
中图分类号: TP309.2 **文献标识码**:A

Promela modeling and analysis for security protocol

LONG Shi-gong, WANG Qiao-li, LI Xiang
(*Institute of Computer Software and Theory, Guizhou University, Guiyang Guizhou* 550025, *China*)

Abstract: The normal model checking technology to analyse security protocol was introduce. As an example, a model for Needham-Schroeder Public-Key Protocol was constructed by using Promela language. SPIN was used to check and discover an attack upon the protocol. The method is easy to extend to check the security protocol which involves several agents.

Key words: security protocol; model checking; SPIN; Promela

0 引言

Promela(Protocol/Meta Language)语言用来对有限状态系统进行建模。它类似于 C 程序语言,允许动态创建所有的进程,并可在进程之间通过消息通道进行同步(使用会面点(rendezvous port)和异步(使用缓冲)进行通信。该语言表达力强,具有直观的 Kripke 结构语义。SPIN(Simple Promela Interpreter)是由贝尔实验室开发的以 Promela 为输入语言的模型检测工具[1],适合于并行系统,尤其是协议一致性的辅助分析检验。

SPIN 可以用在三个基础模型中:

1)作为一个模拟器,允许快速对原型进行随意的或引导性或交互性仿真。

2)可以作为一个详尽的分析器,严格地证明用户提出的正确性要求是否满足(使用偏序简约进行最优化检索)。

3)大型系统近似性证明,用 SPIN 可以对大型的协议系统所覆盖的最大限度的状态空间进行有效的正确性分析。

基本的 SPIN 模型检测的结构如图 1 所示。

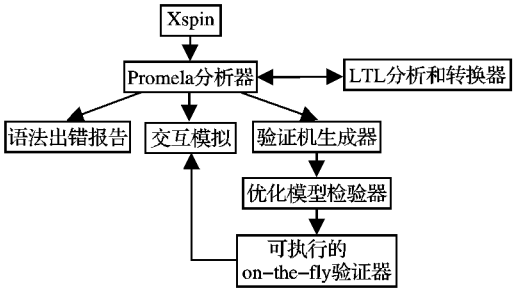

图 1 SPIN 模拟和检验的基本过程

在 SPIN 模型中,用声明的时序公式来刻画系统的行为,如果不满足,运行后将给出反例。LTL 公式还可以表示系统的安全性和活性[2]。基于时序逻辑的模型检测的思想,最先由 Clarke 和 Emerson 等人提出,Vardi 和 Wolper 等人用自动机理论扩展了这项工作。用 Promela 语言描述的并发系统由一个或多个用户定义的进程构成,这些进程定义了系统的动作行为。SPIN 将每个进程转化为一个自动机,这些自动机的交叉积则反映出并发系统的全局动作行为。

一般说来,模型的状态数与并发系统的大小成指数关系,因此随着所要检测的系统的规模增大,所要搜索的状态空间呈指数增大,算法验证所需的时间、空间复杂度将超过实际所能承受的程度。如何有效缓解"状态爆炸",SPIN 主要采用了偏序归约、On-the-fly 等方法。

本文针对密码协议中需要研究解决的问题,给出了 SPIN 模型检测的一般规则和方法。我们用 Needham-Schroeder 公钥密码协议作为实例,利用 SPIN 模型检测方法成功地找出了一条攻击的轨迹。

1 密码协议的 SPIN 检测模型

1.1 密码协议的分析前提

1)完善保密前提,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一般包括四个方面:首先,知道参与协议运行的各主体名及公钥,并拥有自己的加密密钥和解密密钥;其次,可窃听或中途拦截系统中传送的任何消息,并增加自己的知识;第三,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);最后,在系统中可插入新的消息,并可运用他知道的所有知识。

根据入侵者的知识与能力,我们在分析密码协议时将入侵者 I 置于参与协议运行的 N 个主体间(假定密码协议是一

收稿日期:2004 - 12 - 15;**修订日期**:2005 - 04 - 12 **基金项目**:贵州省自然科学基金资助项目(20043029)

作者简介:龙士工(1967 -),男,湖南石门人,讲师,博士研究生,主要研究方向:计算机网络、信息安全; 王巧丽(1979 -),女,河南人,硕士研究生,主要研究方向:模型检验与协议分析; 李祥(1942 -),男,贵州人,教授,主要研究方向:计算机网络、信息安全.

Journal of 计算机应用 Computer Applications

个有 N 方主体参加运行的协议),系统中传送的任何消息,I 可以依据自己的需要拦截或转发。

1.2　协议的基本动作

主体可以执行的动作可被分成内部动作和通信动作。内部动作包括:加密、解密、等待接收、准备发送、确认数据包等;通信动作包括发送和接收。有 4 种关键的动作:BegInit, EndInt, BegRespond 和 EndRespond,其语义如下:

BegInit(j),表示向 j 发起会话请求;

EndInit(j),表示完成与 j 的会话请求;

BegRespond (j) 表示开始响应 j 的会话请求;

EndRespond (j),表示完成对 j 的会话响应。

1.3　协议检测的属性要求

主要考虑协议的时序性质,它被 Woo 和 Lam 称为协调性,通常检查一个密码协议是否受到攻击,应该考虑如下几条性质:

性质 1　如果主体 j 决定只与 k 通信,则总是只与 k 通信,直到完成协议为止。

性质 2　如果随机数临时值 NoncA 在主体 A 与 B 的会话中使用过,则在下一次与 k 的会话中不允许重复使用。

性质 3　通信的发起者和通信的响应者之间能够相互认证对方的身份。如果 B 要认证 A 的身份,那么在任何情况下 A 作为发起者开始与 B 会话的次数 A_Session 不小于 B 作为响应者与 A 完成会话的次数 B_Count_auth,用 LTL 描述为:G(A_session ≥ B_count_auth),如果存在 A_session < B_count_auth,则说明存在入侵者冒充 A 向 B 进行认证。同样,我们可以用 G(B_session ≥ A_count_auth) 来描述初始者 A 对响应者 B 的身份认证。

1.4　模型的 SPIN 实现

一个模型由进程、消息通道和变量组成,相当于一个有限转换系统。如图 2 所示。

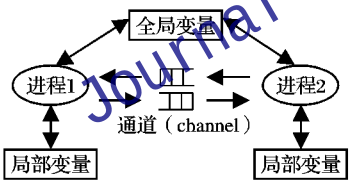

图 2　模型的 SPIN 实现

其大体结构为:类型说明、通道说明、变量说明、进程说明、初始进程。进程是全局对象,在一个进程内,消息通道和变量既可以是全局的也可以是局部的变量。进程定义了模型的行为,而通道和变量则定义了进程运行的环境。模型包括发起者进程、响应者进程和入侵者进程,发起者进程和响应者进程严格按照协议描述的规范传递消息。入侵者进程或者执行合法的动作如发起会话、发送或接收协议所允许的消息和转发消息,或者通过组合临时值(nonces)、密钥和用户 id 产生恶意的消息。入侵者可以将其窃取到的信息存储在它的局部变量中。模型由若干模块的交叉复合构成,协议的一次运行包括协议主体和入侵者动作的交叉序列。

SPIN 采用深度优先搜索遍历状态空间。对于每个状态,考虑其所有可能的转移,但仅执行那些目标状态没被访问过的转移。SPIN 跟踪系统执行过程中每一个访问过的状态,当遇到一个已经访问的状态,SPIN 返回到上一个决策点尝试其他的执行路径。按照这种方式,SPIN 能够遍历所有可能到达的状态,并且每个状态只考虑一次。

2　密码协议的分析实例

2.1　Needham-Schroeder 公钥协议描述

Needham-Schroeder 目的是建立初始者 A 和响应者 B 之间的相互鉴别,协议使用公钥体制,每一个主体如 A 拥有一个公钥,用 K_A 表示,另一个主体也可以从公钥服务器上得到该公钥。K_A^{-1} 是其私钥。$\{M\}_K$ 表示信息 M 用密钥 K 加密。协议中的任何主体都可以用 A 的公钥加密产生 $\{M\}_{K_A}$,但只有 A 才能解开它。协议中还使用临时值 nonces 表示传递信息的新鲜性,N_A 和 N_B 分别表示 A 和 B 的临时值。一个简化的协议表示如下:

消息 1　$A \rightarrow B$: $\{A, N_A\}_{K_B}$

消息 2　$B \rightarrow A$: $\{N_A, N_B\}_{K_A}$

消息 3　$A \rightarrow B$: $\{N_B\}_{K_B}$

这里 A 是初始者,负责与 B 建立联系,A 选择一个临时值 N_A,连同自己的标识 A 用 B 的公钥加密(消息 1)发送给 B。当 B 收到这个消息后,解开这个消息获得 N_A,连同自己的临时值 N_B 用 A 的公钥加密发送给 A(消息 2)。当 A 收到这个消息时,他似乎可以确信他在和 B 通信,因为只有 B 才能解开消息 1 获得 N_A。接下来,用 B 的公钥将 N_B 加密并传给 B(消息 3)。B 似乎可以确信他在和 A 通信,因为只有 A 才能解开消息 2 获得 N_B。而实际上,在该协议发布 17 年后,发现了存在严重缺陷。

2.2　协议的消息描述

为了描述密码协议中的消息,我们构造枚举类型,其中包含一些符号常量,mtype 的各域为:

```
mtype = { ok, err, msg1, msg2, msg3, keyA, keyB,
          keyI, agentA, agentB, agentI, nonceA, nonceB, nonceI };
```

msg1,msg2,msg3 分别表示协议中的消息 1,2,3;keyA, keyB, keyI 分别表示通信主体 A、B 和入侵者 I 的公钥;agentA, agentB, agentI 分别表示协议的通信主体;nonceA, nonceB, nonceI 分别表示 A、B 和入侵者 I 的临时值。

定义消息记录类型:

```
typedef Crypt { mtype key, data1, data2 };
```

协议运行过程中涉及 3 个消息,为了简便起见,采用定长的数据结构,key 表示加密钥,data1, data2 表示加密项。

接下来定义消息通道(chan)来对进程之间的消息传递进行建模。通道是按先进先出的顺序来传递消息。定义通道和定义基本的数据类型一样。为了减少模型的大小,我们定义了一个会面点,根据 Promela 的约定,通过这样的会面点的消息相互是同步的,也就是说向通道送数据和从通道接收数据必须是同时发生。

```
chan network = [0] of { mtype, mtype, Crypt };
```

通道的第一个数据类型 mtype 为消息号;第二个数据类型 mtype 为响应者;Crypt 为传递的消息内容。

2.3　协议的 Promela 建模

1) 主体 A

主体 A 作为协议发起的初始者,它的第一个通信动作是发送信息 1,通信对象可以是 B 也可以是 I,这是两个并发的通信动作,不确定地选择一个通信对象执行。用序列符号"::"表示执行的不确定性。当他接到消息 2 后,A 要检验消息 2 是否以他的公钥加密,并且检查消息中是否含有他发出的临时值 nonceA。否则的话,进程将处于阻塞状态。用 Promela 语句表示为(data.key == keyA) && (data.info1 == nonceA);如果消息 2 中是他期望的值,就把消息 3 发给他的

《计算机应用》Journal of Computer Applications

通信对象,并认为通信成功。

2)主体 B

主体 B 的通信过程和 A 类似。其通信动作含有 2 次接收和 1 次发送。

3)相比之下,入侵者 I 不是固定地按照协议的步骤运行,目的是让 SPIN 发现协议中可能存在的攻击。我们描述入侵者 I 的动作是非常不确定的,让 SPIN 选择它们执行。比如 I 存在发送和接收 2 个不确定的动作,如果选择接收动作,则接下来是拦截或接收 1 条信息,并将拦截的信息存放在结构变量 intercepted 中;如果信息是以 NonceI 加密,则 I 可以解开并可以获得 noncA 或 nonceB,定义 2 个布尔变量 knows_nonceA 和 knows_nonceB 表示是否知道这些临时值。如果选择发送动作,则入侵者 I 又有 2 个可选择的动作:重放一个拦截的数据包或者从已知的信息中构造一个新的数据包发出去。I 作为冒充者,可冒充 A 的身份,以 I(A)参与协议运行。当然,入侵者 I 发出去的一些包可以被合法主体 A 和 B 检测不合法,因而我们的模型中会存在一些死锁,但这并不妨碍对协议的分析。

2.4 协议的系统属性

协议的目的是在秘密状态下确保协议主体的相互鉴别,换言之,如果 A 和 B 成功地运行了一次协议,那么 A 相信他的通信对象是 B 当且仅当 B 相信他的通信对象是 A。而且如果 A 成功地与 B 完成了一次协议的运行,则入侵者 I 不可能知道 A 的临时值;同样,I 也不可能知道 B 的临时值。用 LTL 公式表示如下:

G(statusA = ok ∧ statusB = ok ⇒ (partnerA = agentB ⇔ partnerB = agentA))

G(statusA = ok ∧ partnerA = agentB ⇒¬ knows_nonceA)

G(statusB = ok ∧ partnerB = agentA ⇒ ¬ knows_nonceB)

2.5 对 Needham-Schroeder 公钥协议的攻击

通过运行用 Promela 语言编写的 Needham-Schroeder 公钥协议的 SPIN 模型,发现它并不满足其安全性质,存在违背安全性质的反例,该反例正好是对 Needham-Schroeder 公钥协议的攻击[3,4],攻击如下:

(1) $A \rightarrow I$: $\{A, N_A\}_{K_I}$

(1′) $I(A) \rightarrow B$: $\{N_A\}_{K_B}$

(2′) $B \rightarrow I(A)$: $\{N_A, N_B\}_{K_A}$

(2) $I \rightarrow A$: $\{N_A, N_B\}_{K_A}$

(3) $A \rightarrow I$: $\{N_B\}_{K_I}$

(3′) $I(A) \rightarrow B$: $\{N_B\}_{K_B}$

上述协议运行完,A完全相信与I完成了一次协议的运行;而 I 则成功地冒充 A 与 B 通信,B 被欺骗。

利用 SPIN/Promela 模型模拟 Needhan-Schroeder 公钥协议,运行协议模型后的计算机输出结果如图 3 所示。

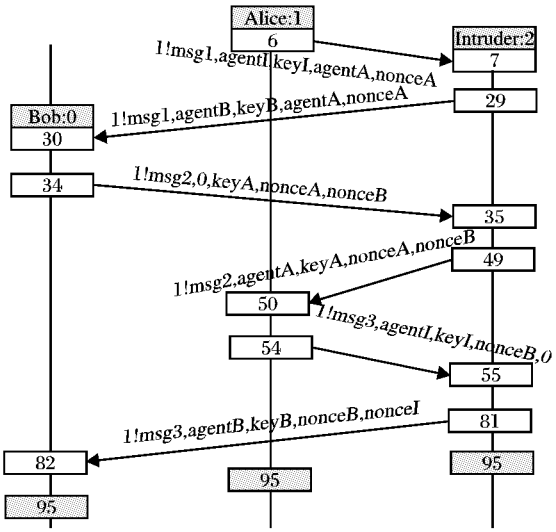

图 3 Needham-Schroeder 公钥协议受攻击的轨迹

3 结语

本文给出了利用 SPIN 模型检测分析密码协议的一般方法。作为一个实例,对 Needham-Schroeder 公钥协议用 Promela 语言建模,并利用 SPIN 进行了分析验证,发现了其安全漏洞。该方法很容易推广到有多个主体参与的密码协议的分析,对于 SPIN 模型,在限制状态空间的情况下,每增加一个主体,只需要增加一个在有限状态系统下运行的并发进程即可。

参考文献:

[1] HOLZMANN GJ. Design and Validation of Computer Protocols[M]. Englewood Cliffs, New JerSey: Prentice-Hall, 1991.

[2] PNUELI A. The Temporal Logic of Programs[A]. Proceedings of 18th IEEE Symposium on Foundations of Computer Science[C], 1977. 46 – 57.

[3] LOWE G. An attack on the Needham-Schroeder public-key authentication protocol[J]. Information Processing Letters, 1995, 56: 131 – 133, 1995.

[4] LOWE G. Breaking and fixing the Needham-Schroeder public key protocol using FDR[A]. Tools and Algorithms for the Construction and Analysis of Systems (TACAS'96), Lecture Notes in Computer Science 1055[C]. Springer-Verlag, 1996. 147 – 166.

[5] MERZ S. Model Checking: A Tutorial Overview[EB/OL]. http://spinroot.com/spin/Doc/course/mc-tutorial.pdf, 2003 – 10.

(上接第 1547 页)

[8] DENG BH, GONG L, LAZAR AA, *et al*. Practical Protocols for Certified Electronic Mail[J]. Journal of network and systems management, 1996, 4(3): 279 – 297.

[9] PUIGSERVER MM, GOMILA JLF, ROTGER LH. Certified Electronic Mail Protocol Resistant to a Minority of Malicious Third Parties[A]. Proceedings of IEEE Infocom 2000[C], 2000. 1401 – 1405.

[10] GOMILA LF, PAYERAS-CAPELLA M, ROTGER LH. An Efficient Protocol Certified Electronic Mail[A]. ISW'2000, LNCS1975[C]. Springer-verlag, 2000. 237 – 248.

[11] NITA-ROTARU C. TURMS: A Non-invasive Certified Email System[EB/OL]. http://citeseer.ist.psu.edu/39111.html, 2005 – 01

[12] ATENIESE G, de MEDEIROS B, GOODRICH MT. TRICERT: A Distributed Certified E-Mail Scheme[A]. ISOC2001 Network and Distributed Systems Security Symposium[C], 2001.

[13] STADLER M. Publicly Verifiable Secret Sharing[A]. In Eurocrypt'96, LNCS1070[C]. Springer-verlag, 1996. 190 – 199.

[14] BAO F, DENG BH, MAO WB. Efficient and Practical Fair Exchange Protocols with Off-line TTP[A]. IEEE symposium on Security and Privacy[C], 1998. 77 – 85.

《计算机应用》Journal of Computer Applications