身份认证协议的模型检测分析

徐蔚文 陆鑫达
(上海交通大学计算机科学与工程系 上海 200030)

摘 要 提出一个直观、易用的模型来模拟和验证身份认证协议,并给出基于 Spin(模型检测工具)的实现,它不仅可以模拟多对参与者同时进行会话,而且还有效缩减了状态空间,从而避免了以前文献中提到的状态爆炸现象.同时该文用 Needham-Schroeder 公钥协议和 TMN 协议来说明如何应用该模型.

关键词 身份认证协议;模型检测;Spin
中图法分类号: TP309

Model Checking of Authentication Protocols

XU Wei-Wen LU Xin-Da
(*Department of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai* 200030)

Abstract The increasing popularity of distributed systems and the emergence of new technologies, such as electronic commerce, demand new security solutions. The corresponding cornerstone of security is often authentication, therefore easy to use methods and tools for modeling and verification of authentication are needed. This paper develops a way of verifying authentication protocols using model checking. Model checking has been proven to be a very useful technique for verifying hardware designs. By modeling circuits as finite-state machines, and exploring all possible execution traces, model checking can find a number of errors in real world designs. Like hardware designs, authentication protocols are very subtle, and can also have bugs which are difficult to find. Specially, this paper presents a simpler model for modeling and verifing authentication protocols, which not only adapts to the situation with multi-pairs of participants but also efficiently reduces the sizes of the state space and avoids states explosion problem mentioned in previous literatures. Also this paper implements the model using Model Checker Spin. Needham-Schroeder Public Key protocol and TMN protocol examples are illustrated to show how this framework is applied.

Keywords authentication; model checking; Spin

1 引 言

近年来,对身份认证协议的形式化模拟和验证受到越来越来多的关注.因为身份认证协议一般很短,而且并不十分复杂,可以用一些非形式化的证明方法来分析或验证它的正确性.但如果这些协议并发执行,分析它们的行为就非常困难.因此许多学者都致力于研究有效分析这些协议的方法,目前关于模拟和验证身份认证协议的方法大致有三种:数学

收稿日期:2002-03-07;修改稿收到日期:2002-09-24.本课题得到国家自然科学基金(60173103)资助.**徐蔚文**,女,1975 年生,博士研究生,主要研究协议的形式化验证方法.E-mail: xu-ww@cs.sjtu.edu.cn.**陆鑫达**,男,1938 年生,教授,博士生导师,主要研究方向为高性能计算、网络计算.

证明、逻辑演绎和模型检测.

Bellare, Rogaway, Shoup 和 Rubin 给出了一个关于安全协议正确性的严格的数学证明[1]. 他们利用伪随机函数的性质和数学证明方法证明:当入侵者试图从协议会话(session)所产生的密钥中截获一个密钥时,不具有统计优势.

关于逻辑演绎方面,一个成功推理安全协议的最早尝试是 BAN 逻辑,它是由 Burrows, Abadi 和 Needham[3] 三人提出的,因而称为 BAN 逻辑. BAN 逻辑是一种基于信任的逻辑,它可以表达和演绎安全协议所涉及的属性,也能表达入侵者的行为规则. 尽管应用这个框架自动证明协议的正确性不很直观,但 BAN 逻辑仍以其简单和高度抽象等优点受广大研究者的欢迎.

第三种技术是模型检测(model checking). 它的基本思想是将协议表示成一个状态集合和一个状态转移集合,而且入侵者的各种行为、往来于协议参与者之间的消息以及那些所有参与者都知道的信息也都被考虑进这个状态集合和状态转移集合. 遍历整个状态空间,检查是否能够到达某个感兴趣的状态,或者到达可以引发某个感兴趣的行为的状态. 已经有很多种模型检测工具被用来模拟和验证安全协议. Meadows 利用 Dolev-Yao 的推广模型[4] 开发了一个基于 PROLOG 的 model-checking 工具[5]. 在他们的系统中,用户可以利用一组描述入侵者如何产生知识的规则来模拟协议. Woo 和 Lam 提出了一个更直观的身份认证协议的模型[2]. 该模型类似于顺序编程,独立模拟每个协议参与者. Lowe 利用 CSP 的 model-checking 工具 FDR 分析了 Needham-Schroeder Public-Key 身份认证协议[6],成功地发现了以前未公布的错误. 但不幸的是,安全协议的 CSP 模型非常不直观,而且它是以参与者的时限(nonce,参与者发布的具有时效性的随机数)为参数的,也就是说,这个模型只能模拟协议的一次运行. 若要证明整个协议的正确性,还需要做进一步证明. 此外,Zhe Dang 等利用 ASTRAL Model Checking 工具分析了加密协议[7]. 张玉清等利用 SMV 分析了只有一对主体参加的 Needham-Schroeder 公钥协议[10].

关于对 Spin 的应用,Audun Jeasang[8] 提出了一些关于如何用 Spin/Promela 模拟和验证安全协议的思想,并给出了关于实现技术的简短描述,同时也提出了一些困难,如系统描述、模型实现以及状态爆炸等.

本文给出一个简单的模型,它不仅可以模拟协议的多次运行,而且还可以模拟多对主体同时进行身份认证. 给出了该模型的 Spin 实现,从而使身份认证协议的验证变得简单易用.

2 身份认证协议模型

身份认证协议通常运行在连接协议参与者(也称主体)的网络上,并且参与者可以在这个网络(通信媒体)上进行异步通信. 通信媒体本质上是被动的,可以被怀有恶意的主体利用、截获或干涉网络通信的内容. 网络包含一个主体(principal)集合,其中一部分主体是可靠的,它们总是按照协议的规则进行动作,相应地,另外一些主体是不可靠的,它们可以进行破坏性操作. 因为不可靠主体能破坏通信媒体,所以可以将它与通信媒体一起模拟成一个进程,称为入侵者. 入侵者能够存储、解密传输在媒体中的消息,它也能加密数据产生新的消息,并将其发送出去以误导可靠主体. 同一主体在不同的协议会话中起不同的作用,例如发起一个协议会话的主体称为发起者,而被发起者所联系的主体称为响应者.

容易看出,协议可以被转化为一个命令(如 SEND, RECEIVE, NEWNONCE)序列[9]. 实事上,这种转化可从协议运行中产生的消息序列自动地对应过来. 一旦生成了这个动作序列,它们的交叉复合便构成了整个协议的模型.

2.1 主体模型

主体可用一个二元组$\langle S,p\rangle$来表示,其中,S 是局部信息,包括会话的局部状态、一个密钥集和一个时限(nonce)集;p 是一个在一组规则约束下的动作序列,通过这组规则可以确定在给定的通话中,究竟哪个用户充当发起者,哪个用户充当响应者.

2.2 协议模型

身份认证协议模型实质上是一组主体动作的异步复合. 所以全局状态可被模拟成一组主体动作的异步复合和一组计数器,这组记数器以一对相互会话主体的 id 为下标. 因此身份认证模型可以表示成一个三元组$\langle \prod, C_i(j,k), C_r(j,k)\rangle$表示,其中,$\prod$ 是一组可靠主体与一个不可靠主体动作的异步复合,它产生一个异步交叉语意,但一对相互进行通信的进程仍是同步的;$C_i(j,k)$为主体 $id\times$主体 $id\to N$ 时,主体 j 向主体 k 发起会话的次数与主体 k 完成响应主体 j 的次数之差;$C_r(j,k)$为主体 $id\times$主体 $id\to N$ 时,主体 j 开始响应主体 k 的次数与主体 k

完成发起主体 j 的次数之差.

2.3 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. 内部动作可以异步执行，任何主体都允许执行内部动作，并当有多个主体同时参与协议时，用交叉复合来模拟所有可能行为. 通信动作包括发送和接收，它仅当一对进程同时进行状态转移时出现.

有 4 种关键的动作：*BegInit*, *EndInit*, *BegRespond* 和 *EndRespond*, 其语义如下：

BegInit(k)表示向 k 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.

这些动作蕴含在每个主体的具体动作中，例如 Needham Schroeder 协议中主体 A 通过向 B 发送消息 $\{N_a, A\}_{K_b}$，首先发起会话请求，这个发送消息的动作就是 *BegInit*(B)动作. 类似的，主体 B 收到这个消息时，首先要对其解密，这就是 *BegRespond*(A)动作. 当 A 收到由 B 发来的响应信息 $\{N_a, N_b\}_{K_a}$ 后，便完成了发起动作(即 *EndInit*(B))；当 B 收到由 A 发来的确认信息后，便完成了响应动作(*EndRespond*(A)).

2.4 模型的动作语意

用以上 4 个动作触发全局变量的更新就可以实时监视以下内容：当用户 j 完成了与 k 的会话(执行 *EndInit*(k))时，用户 k 是否已经参与到此次通话(执行 *BegRespond*(j)). 具体如下：

$$\text{主体 } j \xrightarrow{BegInit(k)} C_i(j,k) \leftarrow C_i(j,k)+1;$$
$$\text{主体 } k \xrightarrow{EndRespond(j)} C_i(j,k) \leftarrow C_i(j,k)-1;$$
$$\text{主体 } j \xrightarrow{BegRespond(k)} C_r(j,k) \leftarrow C_r(j,k)+1;$$
$$\text{主体 } k \xrightarrow{EndInit(j)} C_r(j,k) \leftarrow C_r(j,k)-1.$$

第一行表示，当主体 j 向 k 发出动作 *BegInit* 时，全局变量 $C_i(j,k)$ 加 1；第二行表示，当主体 k 完成响应 j 的动作时，全局变量 $C_i(j,k)$ 减 1；第三行表示，当主体 j 开始响应 k 时，全局变量 $C_r(j,k)$ 加 1；第四行表示，当主体 k 向 j 发起通话的动作结束时，全局变量 $C_r(j,k)$ 减 1.

3 身份验证协议的属性要求

本文主要考虑身份认证协议的时序性质，它被 Woo 和 Lam 称为协调性(correspondence)[2]. 具体地，检查是否满足“如果主体 j 确信它已经完成与主体 k 的通话，则主体 k 必须已经开始与主体 j 的通话”.

注意到，全局变量 $C_i(j,k)$ 和 $C_r(j,k)$ 可以描述这种协调性. 如果 $C_i(j,k)$ 取得负值，就意味着 k 已经完成响应 j，但 j 却根本没有参加此次通话. 类似的，如果 $C_r(j,k)$ 取得负值就说明 k 已经完成发起与 j 通话的动作，但 j 却没加入本次协议. 因此，协议的属性要求在任何时候，对所有 j,k 都有以下性质.

性质 1.　$C_i(j,k) \geq 0$ and $C_r(j,k) \geq 0$.

若在搜索过程中，发现全局变量 $C_i(j,k)$ 或 $C_r(j,k)$ 出现负值，则说明协议受到攻击，记录出现负值的路径就可以发现协议是如何受到攻击的.

另外，还必须保证对任意主体 k 开始响应主体 j 的次数 $C_r(k,j)$ 不得小于主体 j 发起与主体 k 会话的次数 $C_i(j,k)$，即保证在任何时候都有以下性质.

性质 2.　$C_i(j,k) \geq C_r(k,j)$.

若发现 $C_i(j,k) \geq C_r(k,j)$，则说明，攻击者可能冒充合法主体发起会话.

4 模型的 Spin 实现

Spin 是一个模拟有限状态并发系统的模型检测工具，它将并发系统分解为若干模块，然后通过将它们进行交叉复合来构造整个模型. 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过的状态，Spin 返回到上一个决策点(一个输入或异步事件)尝试其它的执行路径. 如果 Spin 从某个特定点出发已经访问过了所有的执行路径，它就返回到上一个决策点，直到遇到一个没被访问过的分支. 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.

对于身份认证协议，Spin 将其分解为若干模块，然后通过将它们进行交叉复合来构造整个身份认证协议的模型. 协议的一次运行包括参与者和入侵者动作的交叉序列.

从以上论述可知，一条轨迹是一个全局状态和我们感兴趣的动作的交叉序列. 可见，每一个主体仅有有限多个后继. 而且，尽管入侵者可以产生无限多个消息，但只允许它发出有限多个消息，因为每一个 SEND 动作必需与一个 RECEIVE 动作匹配. 因为只有有限多个后继，才只需要考虑有限多个运行，因此我们可以通过执行深度优先搜索生成所有可能的

轨迹,并检查是否有一个可达状态违反身份认证协议的属性要求.

4.1 身份认证协议的 Spin/Promela 模型

注意到身份认证协议的攻击大部分来源于网络的被动性,而 Spin/Promela 是一种基于通道的描述语言.因此可以用一个进程 intruder 来模拟网络的各种行为.其他所有参与协议的主体之间的通信都要通过网络(intruder).为了减少状态空间,本文仅用三个进程来抽象协议,发起者(initiator)进程、响应者(responder)进程和入侵者(intruder)进程.发起者进程模拟发起者发起协议的一次会话,响应者进程模拟响应者对此次会话的响应.当然发起者进程可以模拟多个发起者,每个发起者被标以一个唯一的标识(id),响应者进程也可以模拟多个响应者.而入侵者的破坏行为可以用 intruder 进程来模拟,如图 1.

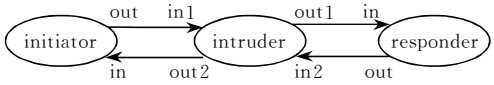

图 1 两方的身份认证协议模型

发起者进程与响应者进程通过入侵者进程按照具体协议所描述的规范传递消息.入侵者进程充当网络和入侵者,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.

为了提高效率,我们不显式地加密每条信息,而是采用一个限制.只有那些被自己的公钥加密的消息,才能被入侵者识别,否则它只能转发、存储或删除.

当主体 j 想与 k 发起一次协议通话时,发起者进程代表 j 发起一次协议通话,产生临时值并存在发起者进程的局部变量中,然后将加密的信息由通道 out 发往 k.

当响应者进程从它的 in 通道收到一个发起信息时,它可以从中获得响应者的 id(不妨设为 k).之后它代表响应者 k 响应此次通话,将生成的临时值存储在响应者进程的局部变量中,并将加密后的响应信息由它的 out 通道发出.

此外,还有两组全局变量 $C_i[j][k]$ 和 $C_r[j][k]$,每当主体 j 向 k 发起协议通话时,$C_i[j][k]$就加 1.当 k 完成响应 j 时,$C_i[j][k]$就减 1.每当主体 j 开始响应 k,$C_r[j][k]$就加 1,当主体 k 完成发起通话的动作后,$C_r[j][k]$就减 1.认证协议的属性要求对任意 j,k,满足 $C_i[j][k]\geq 0$ 且 $C_r[j][k]\geq 0$.

当认证协议涉及三方(发起者、响应者和第三方服务者)时,采用图 2 所示的模型.而协议的属性要求仍然是,对任意 j,k,满足 $C_i[j][k]\geq 0$ 且 $C_r[j][k]\geq 0$.

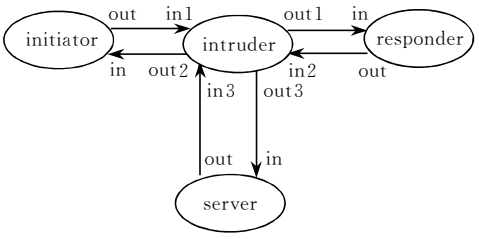

图 2 有第三方参与的身份认证协议的 Spin 模型

4.2 状态爆炸的克服

在 Audun Josang 的 Spin 模型中[8],他通过定义正常的协议会话和被控制(或被破坏)的协议会话来描述协议模型.另外为模拟入侵者,引入两个选择机制:一个是选择用来操纵协议通话的消息,另一个是选择由恶意实体发出消息或消息内容.他还估计了按这种方法模拟 X.509 协议所产生的状态空间为 10^{19},导致状态爆炸.

与之相比,本文所提出的模型及其 Spin 实现,不仅可以模拟协议的多次运行,而且还可以模拟多对主体参加的协议.并采用以下优化策略避免状态爆炸:

(1)由于在 Spin 的实现中,用一个进程模拟多个主体的发起或响应动作,只需为每个主体分配一些存储空间记录它的 id 和必要的局部数据,比为每个主体都分配一个进程节省存储空间.

(2)模型仅利用一个进程(intruder 进程)就模拟了网络的各种行为,包括入侵者的破坏动作,减少了进程和通道的数目.

这里我们考虑的入侵者的破坏动作包括:

(1)双方之间传递的消息没被网络(入侵者)转发;

(2)每个主体包括入侵者都有一个局部变量称为 Msg,并可以被能解密它的其它主体观察到.入侵者有能力聚集偷听来的消息,在可能的时候将其解密并与以前截获的消息组合重发或者构造一个恶意的消息欺骗诚实的主体.

从 Needham-Schroeder 公钥协议(第 5 节)的验证结果(表 1)中可以看到,在 Spin 默认的状态向量限制(1024bytes)下,最多可以验证 17 对主体同时参加身份认证,仅消耗内存 4.18Mbytes.

4.3 模型的简明之处

与以往的工作相比，本文所提的模型具有以下优点：

(1) 直观. 模型的构造来源于对协议参与者的动作序列的观察，因此非常直观、易于理解.

(2) 易于实现. 在我们的模型中引入了一组全局记数器，全局变量易于操作，根据不同主体的不同动作，更新相应记数器的值.

(3) Spin/Promela 语言类似于编程语言，易学、易读：

①Promela 是基于通道的，可模拟主体间的通信；

②Promela 能充分描述主体的行为；

③Promela 可处理全局变量；

④Promela 的 assert 声明适合描述协议的属性要求.

5 实　例

5.1 Needham-Schroeder 公钥协议

Needham-Schroeder 公钥协议通过交换发起者与响应者的 nonces 实现互斥的身份认证. 若假设每个参与者都知道公钥，则协议可被简化为 3 个步骤：

(1) $A \rightarrow B$：$\{N_a, A\}_{K_b}$.

(2) $B \rightarrow A$：$\{N_a, N_b\}_{K_a}$.

(3) $A \rightarrow B$：$\{N_b\}_{K_b}$.

为了发起协议会话，发起者 A 产生一个 nonce N_a，与他自己的 id 一同被响应者 B 的公钥 K_b 加密，而 B 通过解密得知 N_a 后，产生一个 nonce N_b，并将 N_b 和 N_a 用 A 的公钥 K_a 加密. 当 A 收到第二个消息后，A 通过比较两个消息中的 N_a，便知道 B 是否得到了发起消息. 随后 A 向 B 发最后一条消息，当 B 收到后，B 便认证了 A 的身份.

利用本文所给出的 Spin/Promela 模型模拟 Needham-Schroeder 公钥协议，在有 2 对主体参加协议的情况下，在 177 depth 时，违反了断言 $C_i(j,k) \geq 0$，其轨迹如图 3 所示，从而发现了一个著名的攻击：

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

(2) $I(A) \rightarrow B$：$\{N_a, A\}_{K_b}$.

(1) $B \rightarrow I(A)$：$\{N_a, N_b\}_{K_a}$.

(2) $I \rightarrow A$：$\{N_a, N_b\}_{K_a}$.

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

(4) $I(A) \rightarrow I$：$\{N_b\}_{K_b}$.

Low 给出一个修改方案，在第 2 条消息中加入响应者的标识，如下：

(1) $A \rightarrow B$：$\{N_a, A\}_{K_b}$.

(2) $B \rightarrow A$：$\{N_a, N_b, B\}_{K_a}$.

(3) $A \rightarrow B$：$\{N_b\}_{K_b}$.

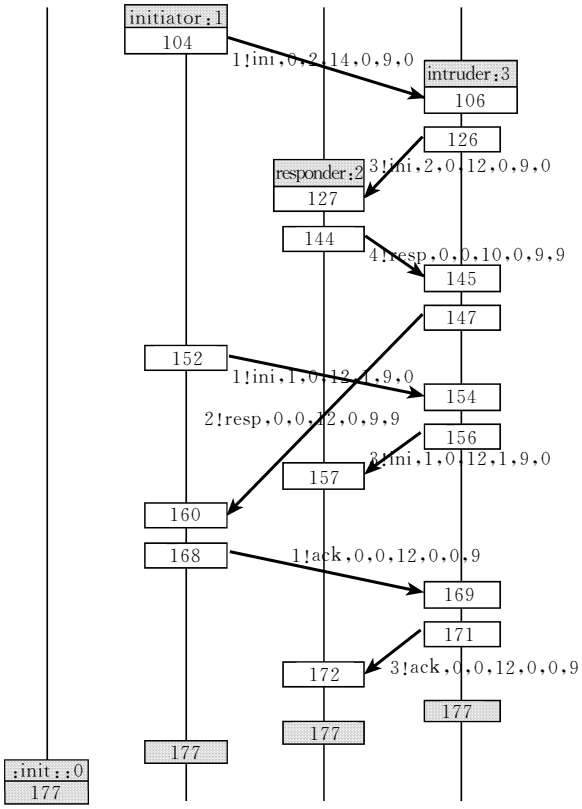

图 3　Needham Schroeder 公钥协议受到攻击的轨迹

本文所给出的模型是以参与协议的主体数目为参数的，当参与协议的主体数目不同时，模型所产生的状态向量、发现错误的深度、内存使用情况等参数均不同，具体对应情况见表 1.

表 1　主体数目与模型输出参数的对应关系

主体数目	反应数目	状态向量(bytes)	错误深度	内存(Mbytes)	存储状态
1	1	164	110	2.950	3144
2	2	180	176	2.542	178
4	4	236	300	2.542	302
8	8	396	692	2.644	694
12	12	620	1276	3.054	1278
15	15	836	1840	3.668	1842
17	17	996	2276	4.181	2278
18	18	1076			

由于 Spin 限制状态向量最大为 1024bytes，因此该模型最多可以模拟 17 对主体同时进行身份认证. 若需验证更多主体参与协议的情况，只需修改宏变量 VECTORSZ 的值(其缺省值为 1024).

5.2 TMN 协议

TMN(Telecommunications Management Net-

work)协议涉及 3 个参与者:发起者,响应者和第三方服务者,它使用两种加密方式:

标准加密. 对于给定的原文 m,发起者和响应者都能利用加密函数 E 产生密文 $E(m)$,但只有第三方服务者,才知道如何解密.

维纳加密. 维纳加密将一对密钥进行位异或,即 $V(k_1,V(k_1,k_2))=k_2$,也就是说,一个实体如果知道密钥 k_1,则它可以通过解密 $V(k_1,k_2)$ 获得 k_2. 这里假设,密钥包含足够的冗余,以确保正确解密.

TMN 协议建立会话密钥,需要交换 4 个消息:

(1) $A\rightarrow S:A.S.B.E(k_a)$.

(2) $S\rightarrow B:S.B.A$.

(3) $B\rightarrow S:B.S.A.E(k_b)$.

(4) $S\rightarrow A:S.A.B.V(k_a,k_b)$.

当发起者 A 想要与响应者 B 建立会话,它选择密钥 k_a 进行加密,并发给第三方服务者(消息 1). 服务者向 B 发一个消息,告诉它 A 想发起一次会话(消息 2). B 确认此次会话,选择密钥 k_b 加密并发给服务者(消息 3). 服务者将这两个密钥进行维纳加密,并发返回给 A(消息 4). 当 A 收到这个维纳密文时,它可以用 k_a 解密,从而获得 k_b.

利用本文所给的 Spin/Promela 模型模拟并验证 TMN协议,在 111 步违反断言 $C_i(j,k)\geq 0$ 或 $C_r(j,k)\geq 0$,其轨迹如图 4,从而发现如下攻击:

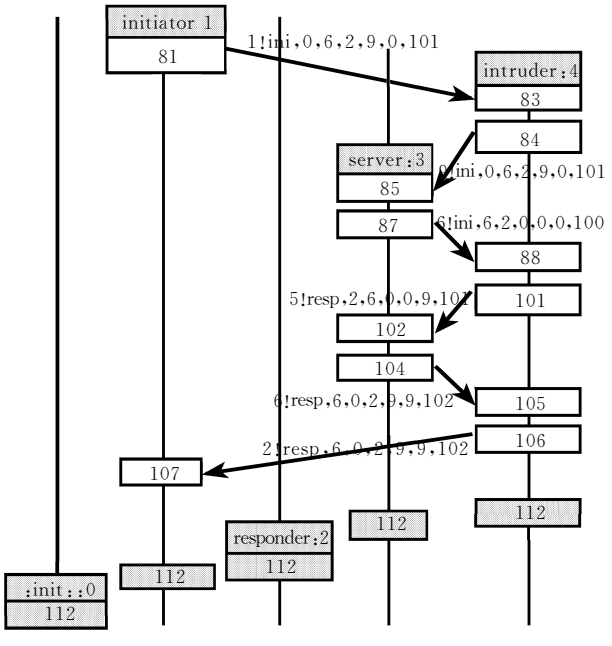

图 4 TMN 协议受到攻击的轨迹

(1) $A\rightarrow S:A.S.B.E(k_a)$.

(2) $S\rightarrow C_B:S.B.A$.

(3) $C_B\rightarrow S:B.S.A.E(k_c)$.

(4) $S\rightarrow A:S.A.B.V(k_a,k_c)$.

如果增加一个断言 $C_i[j][k]\geq C_r[j][k]$,在 91 步发现另外一个攻击,其轨迹如图 5 所示.

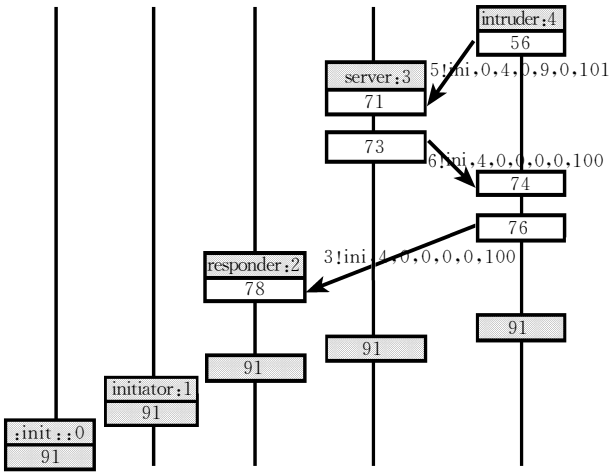

图 5 TMN 协议受到攻击的轨迹

(1) $C_A\rightarrow S:A.S.B.E(k_c)$.

(2) $S\rightarrow B:S.B.A$.

(3) $B\rightarrow S:B.S.A.E(k_b)$.

(4) $S\rightarrow C_A:S.A.B.V(k_c,k_b)$.

与 Needham Schroeder 协议类似,对于 TMN 协议,该模型同样可以验证多对主体同时进行身份认证,可以获得类似的结果.

6 结 论

本文给出了一个简单、直观的模拟和验证身份认证协议的模型及其 Spin 实现,并给出两个实例(Needham Schroeder 公钥协议和 TMN 协议). 该模型直观、简单,而且是以参与协议的主体的数目为参数的,因此可以模拟多对主体同时进行身份认证. 另外,该模型用一个进程模拟多个主体,从而有效地节省了状态空间.

致谢 中国科学院软件研究所计算科学重点实验室的张健、张文辉研究员和中国科学院软件研究所信息安全国家重点实验室的季庆光、王贵林博士阅读过本文,提出许多宝贵意见,在此表示感谢.

参 考 文 献

1 Bellare M, Rogaway P. Provably secure session key distribu-

tion-the three party case. In: Proceedings of the 27th Annual ACM Symposium on Theory of Computing, Las Vegas, 1995. 57～66

2 Woo T Y C, Lam S S. A semantic model for authentication protocols. In: Proceedings of the IEEE Symposium on Research in Security and Privacy, Oakland, 1993. 178～195

3 Michael Burrows, Martin Abadi, Roger Needham. A logic of authentication. In: Proceedings of the ACM Symposium on Operating Systems Principles, Arizona, USA, 1989. 1～13

4 Dolev D, Yao Y. On the security of public key protocols. IEEE Transactions on Information Theory, 1989, 29(2): 198～208

5 Meadows C. Applying formal methods to analysis of a key management protocol. Journal of Computer Security, 1992, 1(1): 5～36

6 Lowe G. Breaking and fixing the needham-schroeder public-key protocol using FDR. In: Tools and Algorithms for the Construction and Analysis of Systems. Passau, Germany: Spring-Verlag, LNCS 1055, 1996. 147～166

7 Zhe Dang, Richard A Kemmerer. Using the ASTRAL model checker for cryptographic protocol analysis. In: Proceedings of the DIMACS Workshop on Design and Formal Verication of Security Protocols, New Jorsey, 1997

8 Audun, Josang. Security protocol verification using spin. In: Proceedings of the 1th Workshop on Automata Theoretic Verification with the SPIN Model Checker——SPIN95, Montreal, Quebee, 1995

9 Will Marrero, Edmund Clarke, Somesh Jha. Model checking for security protocols. Carnegie Mellon University: Technical Report CMU-SCS-97-139, 1997

10 Zhang Yu-Qing, Wang Lei, Xiao Guo-Zhen, Wu Jiang-Ping. Model checking analysis of needham-schroeder public-key protocal. Journal of Software, 2000, 11(10): 1348～1352(in Chinese)
(张玉清，王　磊，肖国镇，吴建平. Needham Schroeder 公钥的协议模型检测分析. 软件学报, 2000, 11(10): 1348～1352)

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

XU Wei-Wen, female, born in 1975, Ph. D. candidate. Her research interest is in automatic verification of protocols.

LU Xin-Da, male, born in 1938, professor, Ph. D. supervisor. His research interests focus on network computing.