SAL 框架下的安全协议分析技术研究*)

李 超[1] 董荣胜[1] 郭云川[1,2]

(桂林电子科技大学计算机系 桂林 541004)[1](中国科学院计算技术研究所 北京 100080)[2]

摘 要 SAL是一种新的结合模型检验和定理证明技术的分析框架。它使用一种过渡性的中间语言将不同验证工具统一在其框架之中,以便保证安全协议的验证不会顾此失彼。本文对基于SAL框架下的安全协议分析技术进行了研究,介绍了SAL中间语言描述安全协议的方法,包括消息项中协议主体、入侵者、现时值、公私密钥、对称密钥、加密、解密、DH问题等的描述,协议规则与性质的描述,给出了在SAL下刻画入侵者模型和建立通信模型的一般方法,并以经典的NSPK协议为例,找到了一个已知的认证性攻击。

关键词 SAL(Symbolic Analysis Laboratory),模型检验,定理证明,安全协议,NSPK协议

1 引言

在安全协议的形式化研究工作中,模型检验技术和定理证明技术是重要的验证手段,它们有各自的优势与不足。模型检验工具高度的自动化与反例生成能力为协议分析提供了便利的途径,然而模型检验中存在状态空间爆炸的问题。定理证明技术虽然能很好的克服状态空间爆炸的问题,但目前还存在自动化程度较低,专业技能要求较高的不足。因此将模型检验技术与定理证明技术相结合,使其优势互补已成为一个趋势[1]。

在模型检验和定理证明相结合的研究工作中,SRI计算机科学实验室提出了一种将定理证明和模型检验等多种符号分析技术相整合的框架SAL(Symbolic Analysis Laboratory)。SAL主要用于分析并发问题,它通过一种过渡性中间语言,将不同的验证工具统一起来[2]。

安全协议是并发问题中的一类特定问题和目前的研究热点之一,为了保证协议的验证不会顾此失彼,就必然希望能在一个统一的框架下完成安全协议各种性质的验证,显然SAL是一个较为理想的框架。为此,本文对SAL框架下的安全协议分析技术进行了研究,介绍了SAL中间语言描述安全协议的方法,包括消息项中协议主体、入侵者、现时值、公私密钥、对称密钥、加密、解密、DH问题等的描述,协议规则与性质的描述,给出了在SAL下刻画入侵者模型和建立通信模型的方法,并以经典的NSPK协议为例,找到了一个已知的认证性攻击。

2 安全协议描述方法

2.1 消息项描述

SAL中间语言的消息类型继承了PVS的消息类型[3,4],因此它拥有强大的描述能力。用下面的方法来描述消息项。

(1)协议中参与者的描述

协议的参与者由合法主体与入侵者构成。将协议的参与者定义为枚举类型,a、b表示协议的合法主体,e表示攻击者,X表示加入的额外的错误元素。

```
ids: TYPE = {a, b, e, X};
participants: TYPE = {x: ids|x/=X};
intruder(x: participants): BOOLEAN =x=e;
intruders: TYPE = {x: participants|intruder(x)};
principals: TYPE = {x: participants | NOT intruder(x)};
```

intruder(x)为一个谓词,用于判断x是否为一个入侵者,并据此定义协议的入侵者intruders与合法主体principals。

(2)现时值nonce的描述

```
nonces: TYPE = ids;
nonce(a: participants): nonces = a;
```

协议中,nonce值主要用于保证消息的新鲜性,防止消息重放。

(3)明文的描述

协议中的明文数据用N元组表示,N的值视协议中的总的消息项个数而定。元组中的项对应协议中的消息项。如协议中某条消息由主体名,现时值两个消息项组成,那么对应的二元组为:

```
dmsg: TYPE =[ids, nonces];
```

值得指出的是,对于发送的每一条明文,如果某

*)广西自然科学基金项目(编号:0542052)资助。

项无数据，则用 X 表示。如[a，N_a，X]表示，协议中总共有三个消息项，而该条消息仅由一个主体名和一个现时值两个消息项组成。

(4)密钥与密文的描述

对于对称密钥体系，发送方将接收方的 id 作为其通信的对称密钥；对于非对称密钥体系，发送方将自己的 id 作为接收方的公开密钥，接收方将自己的 id 作为私有密钥。

将密文数据 emsg 定义为抽象数据类型：

emsg：TYPE = DATATYPE

enc(key：ids，payload：dmsg)

它包含一个加密动作 enc，其作用是将明文消息 payload 用以 id 为 key 的密钥进行加密。

(5) 解密的描述

dec(k：participants，m：emsg)：dmsg=

IF key(m)=k THEN payload(m) ELSE arb ENDIF；

定义一个 dec 函数，其作用是当密文 m 的 key 与主体自身的 id 相同时，解密该密文。根据(4)中对密钥的约定，不难发现它对对称密钥体系和非对称密钥体系都适用。

(6)DH 问题的描述

针对 DH 问题，进一步将密钥抽象为一个二元组(k_get，k_send)，并将其中的 k_send 与相应主体的 DH 私有数绑定。本文用记录类型来描述 DH 问题：

K_DH：TYPE =

[#k_get：ids，k_send：ids #]

k_get 指明了接收方用来计算各个密钥的 DH 公共数，k_send 指明了该发送方发送的 DH 公共数。相应地，当某个主体收到用如上定义的密钥加密的消息时，将检测该密钥的 k_get 值是否等于自己生成并发送出的 DH 公共数，以及在自己所收到过的 DH 公共数中是否有 k_send。如果这两方面都满足，则表明该主体拥有相应的解密密钥，从而可以读取相应的消息内容。

(7)消息的描述

在协议的传输过程中，通常需要记录消息的发送方，接收方，以及协议的各种数据。

本文用记录类型来描述协议中消息：

msg：TYPE =[# src：participants，dest：participants，em：emsg #]；

其中 src 表示消息的发送方，dest 表示消息的接收方，em 表示消息中通讯的密文。

2.2 协议规则描述

协议是一系列规则的序列。在 SAL 中，将协议运行中这些序列的状态用有限状态机进行刻画，从而对协议规则进行描述。

对于协议的参与者(包括合法主体和入侵者)，其既可能是协议的初始者，也可能是协议的响应者。在用 SAL 中间语言描述时，不需要针对主体为初始者和响应者的情况分别建立两个模块。由于在迁移中，仅仅需要考虑的是主体的状态，即各状态间之间的迁移。所以只需将主体的迁移放入到一个模块中。在这个模块中，协议合法主体与协议的参与者之间(除去自身，不与自己交互)进行交互。通过主体的状态和缓冲区的值来判断其在某次运行中是作为初始者还是响应者，从而执行不同的动作，迁移到不同的状态。

例如“pc = sleeping AND nstate = net! empty→pc'= waiting”，通过协议状态 sleeping 和缓冲区状态 empty，判断“→”之后为第一条消息中初始者的动作。“→”表示蕴含，pc'表示 pc 的下一状态。

2.3 入侵者模型

根据 Dolev-Yao 模型，假设入侵者对网络拥有完全的控制能力。在协议的运行中，入侵者将不依照协议的要求，而是根据自己拥有知识的情况，进行窃听、存储、删除、生成、转发和重放消息，在 SAL 中，给出图 1 以刻画该入侵者模型，a 和 b 为合法主体，e 为入侵者。

为了记录入侵者拥有和存储的知识，必须引入新的变量存储截获的数据。如引入 nmem 和 mmem 分别存储截获的 nonce 值和密文。

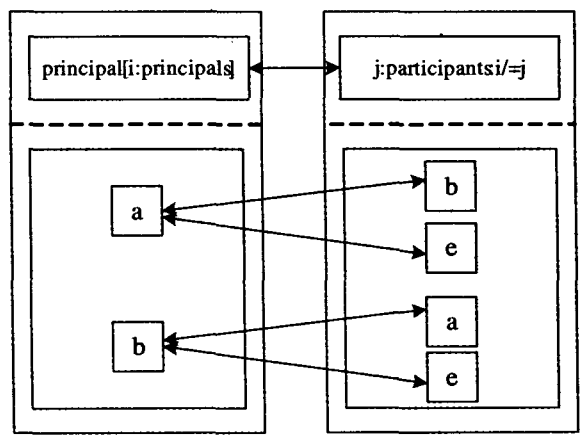

图 1 SAL 中入侵者模型

入侵者 e 的初始知识：

nmem = nonce(e)；

mmem = enc(X，(X,X,X))；

入侵者截获知识的过程：

① 解密发往自己的消息——当 e 判断接收的消息是以自己为目的地时，用私钥或对称密钥解密消息，捕获密文中的 nonce 值，存于 nmem(e 接收到消息后也可以不作任何响应，保持 nmem 不变)；

nmem′ IN {dec(x,imsg. em). 2，nmem}；

② 窃听经过自己的消息——当 e 判断接收的

消息不是以自己为目的地时，将这条密文捕获，暂存于 mmem 中(e 接收到消息后也可以不作任何响应，保持 mmem 不变)。

mmem′ IN {imsg. em, mmem}

入侵者冒充的过程：

① 冒充合法主体 i 与合法主体 j 进行通信，将暂存的消息发给 j；

omsg′ = (# src := i, dest := j, em := mmem #)

② 冒充合法主体 i 与合法主体 j 进行通信，利用自己的知识构造新的消息，发送给 j。

omsg′ = (# src := i, dest := j, em := enc (j, (i, n1′, n2′)) #)

2.4 通讯模型

本文给出 SAL 下进行安全协议分析建模的通信模型如图 2 所示。网络环境缓冲区与协议规则分别建立两个上下文(CONTEXT)：network 与 protocol。

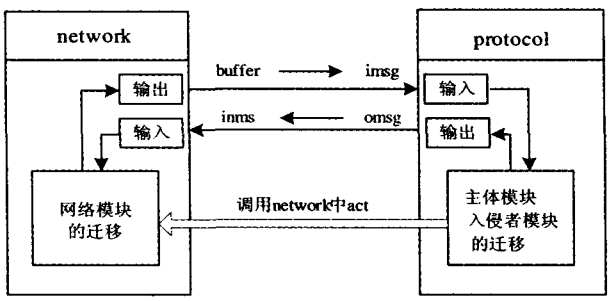

图 2 通讯模型

network 与 protocal 拥有各自的输入输出接口。对于 network，它接收来自 protocol 输出的消息 omsg，将其重命名为 inms 作为自己的输入，之后 network 判断当前的缓冲区状态和被 protocol 调用的动作，完成网络模块中读写的状态迁移。对于 protocol，它将 network 缓冲区中 buffer 值取出，重命名为 imsg 作为输入，之后 protocol 将 imsg 作为协议通信中的消息完成各模块的迁移。在这个过程中，主体模块和入侵者模块会对消息进行某些操作，这时只需对 network 中定义的操作进行调用，调用这些操作后，缓冲区的状态及缓冲区的值会发生改变，而对 protocol 而言，这些改变是透明的，它们在 network 中定义的自动机中自动完成。

3 NSPK 协议的验证

下面以经典的 NSPK 协议为例，介绍在 SAL 框架下对协议进行验证的一般方法。

3.1 NSPK 协议建模

首先给出了 NSPK 协议的描述[5]。

(1)A→B：$\{N_a, A\}_{Kb}$

(2)B→A：$\{N_a, N_b\}_{Ka}$

(3)A→B：$\{N_b\}_{Kb}$

将协议建模为有限状态自动机

(1)主体 A 为初始者的情况

A 可能的状态有{sleeping, waiting, engaged}。初始时，A 在 sleeping 状态，待发送完 msg1 后，迁移至 waiting 状态；等待 msg2，验证 msg2 合法后，发送 msg3 并迁移至 engaged 状态。至此，A 作为初始者的一次运行结束。其有限状态机如图 3。

图 3 A 为初始者的状态图

(2)主体 A 为响应者的情况

A 可能的状态有{sleeping, tentative , responding}。初始时，A 在 sleeping 状态等待 msg1，当接收到 msg1 并验证其合法后，发送 msg2，并迁移到 tentative 状态等待验证 msg3；当 msg3 到来，验证其合法后，迁移至 responding 状态，完成认证。至此 A 作为响应者的一次运行结束。由于篇幅关系，略去其状态图。

由于在迁移中仅仅需要考虑的是 A 的状态，即{sleeping, waiting, engaged, tentative, responding}之间的迁移，所以只需将主体 A 的迁移放入到一个模块中。通过主体的状态和缓冲区的值来判断其在某次运行中是作为初始者还是响应者，从而执行不同的动作，迁移到不同的状态。

3.2 认证性描述

认证有两方面，一种是响应者对于初始者的认证，另一种是初始者对于响应者的认证。由于篇幅原因，本文仅描述第一种情况。响应者对于初始者的认证要求描述：响应 A 承诺一个与 B 的会话，只有当 B 的确参与了会话。因此有

prop: THEOREM system|-G((FORALL (x, y: principals):

(pc[x]=responding AND responder[x]=y) =>

((pc[y] = waiting OR pc[y] = engaged) AND responder[y]=x)));

prop 表示对于主体 x 和 y，x 是响应者，并且此时它完成了作为响应者的动作，处于 responding 状态。如果它所要认证的主体是 y，则此时 y 要么处于 waiting 状态要么处于 engaged 状态，换句话说，协议的此次运行必然由 y 发起。

3.3 认证性验证

在 SAL 下拥有强大的工具集，SAL 内核可以调用不同的验证工具对规格进行验证。本文通过调

用SAL下的sal-smc工具对NSPK协议的认证性进行了分析，找到了一个已知的认证性攻击。通过生成的反例发现存在如下的攻击序列：

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

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

(3)$B \rightarrow I(A)$: $\{N_a, N_b\}_{Ka}$

(4)$I \rightarrow A$: $\{N_a, N_b\}_{Ka}$

(5)$A \rightarrow I$: $\{N_b\}_{Ki}$

(6)$I(A) \rightarrow B$: $\{N_b\}K_b$

入侵者在第(1)步解密消息，获得Na，并在第(2)步通过获得的Na冒充A与B进行通信，第(3)步B与I(B认为是A)进行通信，第(4)步I将B发给它的消息重放给A，第(5)步I解密A发送回来的消息得到Nb，第(6)步I冒充A向B发送得到的N_b。I假冒A成功。这样，B认为是与A建立了认证关系，而实际上一直与B进行通信的是I。

结束语 本文对SAL框架下的安全协议分析技术进行了研究，介绍了SAL中间语言对安全协议的描述方法，给出了在SAL框架下对入侵者模型和通信模型的刻画方法，并以经典的NSPK协议为例，找到了一个已知的认证性攻击。下一步的研究方向是针对开端问题等目前安全协议形式化分析领域所面临的公开问题[6]，尝试在SAL下对这类问题的验证方法进行研究，以求SAL在安全协议领域应用能不断拓展。

参考文献

1 Rajan S, Shankar N, Srivas M K. An integration of model checking with automated proofchecking. In: Proceedings of the 7th International Conference On Computer Aided Verification, volume 939 of LNCS, Liege, Belgium, Springer Verlag, 1995. 84～97

2 Bensalem S, Ganesh V, Lakhnech Y, et al. An overview of SAL. In: Holloway C M ed. Fifth NASA Langley Formal Methods Workshop (LFM 2000), 2000. 187～196

3 Owre S, Rushby J, Shankar N. PVS: A prototype verification system. In: Proc. 11th International Conference on Automated Deduction (CADE), volume 607 of Lecture Notes in Artificial Intelligence, Saratoga, NY, 1992. 748～752

4 de Moura L. SAL: Tutorial. Available at: http:// sal. csl. sri. com/doc/salenv_tutorial. pdf

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

6 Meadows C. Formal methods for cryptographic protocol analysis: Emerging issues and trends. IEEE Journal on Selected Areas in Communication, January 2003, 21(1): 44～54

(上接第36页)

全控制权限的敏感区域和文件的认证级别。

安全审计是安全管理中访问控制过程的一部分。审计系统可以追踪到特殊登录用户以及其访问资源。

代理服务器和防火墙是安全管理中的两个特定访问控制系统，主要用于防止公共网络如因特网成员访问内部网络资源。代理服务器允许内部用户作为用户代理访问因特网。因此内部IP地址可以隐藏，从而降低外部攻击的可能性。另外代理服务器也允许网络管理端管理内部用户的因特网访问权限。

在一定的规范和原则下，防火墙允许外部用户访问内部网络资源。

3.5 网络的计费管理

收集、缓冲、付款传送和计费信息。校园网络要做到可运营、可管理和用网公平性，网络使用身份验证、授权和计费是必不可少的，而且要实现灵活计费的策略。

另外，日志管理也是网络管理的重要组成部分之一。当网络发生问题时，如何第一时间找到问题的根源？如何满足公安部追查用户上网行为的要求？应用资源众多，如何掌握用户的上网习惯等。采用强大多样的日志保存方式，有效记录这些信息、保存日期较长的日志，管理员坐在办公室就可以轻松的掌握办公区、学生区、家属区等各层用户的上网行为；当出现问题时，也可以迅速找到问题的根源。

最后，对整个校园网中的硬件及软件都要分门别类地进行建档，有完整地使用日志、维护日志等。参数的调整、配置的改动都要有详细的记录，不会因为人员的调动或不到场网络管理就停顿下来，管理日志也为以后的网络管理积累经验和技术文档。建立健全各种规章制度，如网络使用规范、上网规范、操作管理制度、维修保养制度等。

总结 总之，校园网络管理实际上是一件既技术性很强又琐碎的事情，在旁人看来，即使没有人去管它，建好网也正常运转着，如果平时没有对网络进行诊断和管理，一旦出了故障，能否及时修复就成问题了。作为网络管理人员如果不能紧跟技术发展潮流，随时学习新的网络知识最后很可能被淘汰。所以在平时的工作中不断地学习新的知识，多多实践。

网络管理本身还有许多空间和领域值得我们去学习和创新，希望本文能够使大家对网络管理和维护的理解有所帮助。

参考文献

1 Tanenbaum A S, The Netherlands. 计算机网路(第四版). 潘爱民译. 清华大学出版社，2004

2 http://www. zdnet. com. cn/

3 http://tech. ccidnet. com/