基于 CCS 的加密协议分析*

丁一强

(中国科学院软件研究所计算机科学开放研究实验室 北京 100080)

摘要 加密协议的分析需要形式化的方法和工具.该文定义了加密协议描述语言 PEP (principals+environment=protocol),并说明对于一类加密协议,其 PEP 描述可以转化为有穷的基本 CCS 进程,由此可以在基于 CCS 的 CWB(concurrency workbench)工具中分析加密协议的性质.此方法的优点在于隐式地刻画攻击者的行为,试图通过模型检查(model checking)发现协议潜在的安全漏洞,找到攻击协议的途径.

关 键 词 加密协议,协议分析,形式化方法,CCS,模型检查.

中图法分类号 TP309

1 加密协议及其形式化分析

网络加密协议的目的在于运用加密技术保证开放网络的安全性.在加密协议中,加密技术是非常重要的因素,但另一方面,如果协议逻辑设计不当,则无异于在坚固的堡垒中留了个后门,攻击者根本不用费事去解密就可以达到其目的.为了保证加密协议设计的正确性,避免发生潜在的错误,就需要形式化的工具来精确地描述协议的行为以及协议所要达到的目标,并帮助分析此协议能否达到其预定目标.事实已经证明,形式化方法是有效的.

本文定义加密协议描述语言 PEP(principals+environment=protocol),说明对于一类加密协议,其 PEP 描述可以转化为有穷的基本 CCS[1]进程,由此可以在基于 CCS 的 CWB(concurrency workbench)工具[2]中分析加密协议的性质.本文提出的方法的优点在于隐式地刻画攻击者的行为,可以通过模型检查(model checking)[3]来发现协议潜在的安全漏洞,从而找到攻击协议的途径.

本文第 2 节介绍加密协议的诸要素.第 3 节定义加密协议描述语言 PEP 的语法和语义.第 4 节举例分析 Needham-Schroeder 公开密钥协议.第 5 节是结论.

2 加密协议的模型

2.1 消 息

主体之间发送和接收的消息有如下几种:主体标识、正文、随机量(nonce)、密钥、加密消息和复合消息.

主体标识唯一标明了主体的身份,在开放网络环境中,主体标识一般是公开的.正文是一个字串.随机量是主体随机产生的,其他主体无法通过猜测得到它.在加密协议中,消息发送者在要传送的消息中加入随机量,以后,当他接收到包含此随机量的消息时,就可以断定此消息是新近生成的.因此,正确运用随机量可以防止重放(replay)攻击.

当前的密钥体制分为两种.一种是秘密密钥体制,也称对称密钥体制,其加密密钥与解密密钥是同一个密钥,除非密钥体制被攻破,用一个密钥加密的消息只有用此密钥才能解密.除了用于保证信息的保密性之外,

---

* 本文研究得到国家自然科学基金资助.作者丁一强,1970 年生,博士,主要研究领域为形式化方法,网络安全协议分析.

本文通讯联系人:丁一强,北京 100080,中国科学院软件研究所计算机科学开放研究实验室

本文 1998-04-30 收到原稿,1998-10-22 收到修改稿

主体通过辨认加密信息,还可以判断此信息的来源,因为有能力用此密钥加密信息的主体是确定的.另一种是公开密钥体制,其密钥成对存在,一个是公开密钥,另一个是私有密钥,消息发送者可以用接收者的公开密钥加密消息,只有拥有对应私有密钥的接收者才能解密此消息.

复合消息是消息的复合,主体很容易把复合消息分解为其组成部分.

2.2 主 体

加密协议的主体可以是人、机器或进程.主体行为包括发送和接收消息以及内部运算(包括加密、解密、逻辑判断等).主体之间是并发运行的.

我们假设主体有区分各类消息的能力,并且对不符合协议的消息拒绝接收.例如,按照协议,主体 A 应该从主体 B 接收密钥 k,那么此时主体 A 就拒绝接收除密钥以外的其他消息.另外,主体也能辨别同类消息的不同值.例如,主体可以辨别接收到的随机量是否是该主体刚才产生的.

2.3 环 境

在开放的网络环境中,潜在的攻击者可能控制整个环境,对网络上传送的消息侦听、篡改和重播.在本文中,我们可以把环境看作攻击者的知识库.一方面,随着协议的运行,知识库在增长;另一方面,协议主体的行为也由当前的环境所决定.

2.4 加密协议的性质

各种加密协议要达到的目的不同,要满足的性质也不同.本文讨论主体认证(authentication)协议,但本文提出的方法不仅仅局限于认证协议.

认证协议是加密协议中很重要的一类.MIT 的雅典娜计划提出的 Kerberos 协议就是使用共享密钥的认证协议.认证协议的目的是用来确认协议中主体的身份,防止攻击者假冒其他主体访问资源.

Woo 和 Lam 在文献[4]中提出了认证协议要满足的两个性质:对应性(correspondence)和保密性(secrecy).本文分析对应性.非形式地说,对应性是指认证协议中的主体的动作有着对应关系.例如,当协议应答方 B 结束协议时,对应的协议发起方 A 必须已开始进行此协议.我们用

$$T \text{ authenticate } R$$

表示动作 T 之前一定有动作 R,则

$$end\_response \text{ authenticate } begin\_init$$

就反映了上述对应关系.

3 加密协议描述语言 PEP

CCS 是一种用于描述和分析并发和通信系统的形式体系[1].并发和通信系统通常可描述为 CCS 进程,系统的性质则可用模态逻辑来刻画.检查一个 CCS 进程是否满足其模态性质称为模型检查(model checking).基本 CCS(basic CCS)有很成熟的理论结果以及相应的工具,如 Concurrency WorkBench[2].但用它来描述协议中常见的传值进程时,数据域的无穷会导致状态迁移图的无穷分叉,工具对于这种情况就无法处理了.

我们现在考察如何基于 CCS 来描述和分析加密协议.

根据上一节的协议模型,协议主体之间可以传递各种消息,因此协议是传值进程,其值域是整个消息空间,显然是无穷的.这给加密协议的分析带来了困难.另外,对于加密协议的环境也必须给出合适的刻画.我们希望隐式地刻画环境(攻击者)的行为,分析协议时,如果协议有安全漏洞,我们的方法应该能推导出攻击的路径.

考察加密协议中主体和环境的行为可以发现,协议主体对于由环境传递的由其他主体(可能是攻击者)发送的消息可看作是有选择地接受的.这样,对于许多加密协议来说(其中包括著名的 Needham-Schroeder 公开密钥协议),在任何状态,主体可做的动作是有限的,这一点对于用 CCS 来分析加密协议带来了方便.但另一方面,主体的行为也是受环境影响的.随着协议的运行,主体之间传送的消息越多,主体受攻击的可能性就越大.因此,同样的协议在第 2 遍运行时产生的状态分叉就比第 1 遍运行时要多.

协议模型的上述特点,标准 CCS 是无法直接刻画的.因此,本文基于 CCS 定义 PEP 语言,以达到描述和分析加密协议的目的.

3.1 消息空间

由上节所述,消息包括主体标识、正文、随机量、密钥、加密消息和复合消息.在本文中,我们用 *uval*(*a*)表示主体标识值,*uvar*(*x*)表示主体标识变量;用 *nval*(*n*)表示随机量值,*nvar*(*x*)表示随机量变量;用 *kpub*(*a*),*kpvt*(*a*),*kshd*(*a*)分别表示公开密钥、私有密钥和共享密钥;用 *text*(*t*)表示正文;用 *enc*(*k*,*m*)表示加密的消息;用(*m*1,*m*2)表示消息的复合.

为了避免状态迁移图的无穷分叉,我们限制消息空间中不允许出现消息变量,只允许出现主体标识变量和随机量变量.如何放宽这个限制,使得本文提出的方法应用更广泛是我们将来的努力方向.但即使有此限制,我们仍然可以用它来刻画一大类的加密协议.

3.2 消息推理引擎

环境(攻击者)的消息推理可由如下的二元关系 $\vdash$ 来刻画.

R1: 如果 $m\in S$,则 $S\vdash m$.

R2: 如果 $S\vdash m$,并且 $S\vdash k$,则 $S\vdash enc(k,m)$.

R3: 如果 $S\vdash enc(k,m)$,且 $S\vdash k^{-}$,则 $S\vdash m$.其中,$(kpub(a))^{-}=kpvt(a)$,$(kshd(a))^{-}=kshd(a)$.

R4: 如果 $S\vdash m1$,$S\vdash m2$,则 $S\vdash(m1,m2)$.

R5: 如果 $S\vdash(m1,m2)$,则 $S\vdash m1$,且 $S\vdash m2$.

3.3 加密协议描述语言PEP

3.3.1 动　作

协议主体能做的动作是 *send*(*m*)和 *recv*(*m*).在协议运行的任何状态,主体只能发送封闭的消息,即消息内部无变量出现.对于接收动作,它可接收的消息是一个消息模式,其中可以出现变量.只有满足此模式的消息主体才接收.

3.3.2 主体+环境=协议

PRINCIPALS ::= stop ; 终止
| *ACTION*. *PRINCIPALS* ; 前缀
| *PRINCIPALS* + *PRINCIPALS* ; 选择
| *PRINCIPALS* | *PRINCIPALS* ; 并发

ENVIRONMENT ::= {*MESSAGE*,...,*MESSAGE*} ; 消息集合

PROTOCOL ::= *PRINCIPALS* in *ENVIRONMENT*

在 PEP 中,主体进程的描述与 CCS 的进程非常类似,只是其行为与环境有着相互作用.我们定义协议的操作语义:

$$\text{SEND}\quad \frac{}{send(M).\ P \text{ in } S \xrightarrow{send(M)} P \text{ in } S\cup\{M\}}$$

$$\text{RECV}\quad \frac{}{recv(M(\vec{z})).\ P \text{ in } S \xrightarrow{recv(M(\vec{v}))} P[\vec{v}/\vec{z}] \text{ in } S}\quad \text{如果} S\vdash M(\vec{v})$$

$$\text{SUM1}\quad \frac{P \text{ in } S \xrightarrow{\alpha} P' \text{ in } S'}{P+Q \text{ in } S \xrightarrow{\alpha} P' \text{ in } S'}\qquad \text{SUM2}\quad \frac{Q \text{ in } S \xrightarrow{\beta} Q' \text{ in } S'}{P+Q \text{ in } S \xrightarrow{\beta} Q' \text{ in } S'}$$

$$\text{PAR1}\quad \frac{P \text{ in } S \xrightarrow{\alpha} P' \text{ in } S'}{P|Q \text{ in } S \xrightarrow{\alpha} P'|Q \text{ in } S'}\qquad \text{PAR2}\quad \frac{Q \text{ in } S \xrightarrow{\beta} Q' \text{ in } S'}{P|Q \text{ in } S \xrightarrow{\beta} P|Q' \text{ in } S'}$$

其中 $P[\vec{v}/\vec{z}]$ 是指用 $\vec{v}$ 替换 P 中的 $\vec{z}$ 得到的结果.

根据协议的操作语义,我们可以由协议的 PEP 描述得到其状态迁移图,然后再转化为基本 CCS 进程描述.并且,由于我们限制 *recv* 动作中的消息只含有主体标识变量和随机量变量,因此集合 $\{M(\vec{v})\mid S\vdash M(\vec{v})\}$ 是有穷的,对应的 CCS 转化是有穷分叉的.另外,我们也不允许主体进程的递归定义,这样,协议模型的 CCS 转化就是有穷的进程.

3.4 对应性

我们用模态逻辑公式[3]:

$$T \text{ authenticate } R = \nu X.([T]\text{false} \wedge [-R]X)$$

来刻画协议主体动作的对应性,意即进程开始不能做 T 动作,并且对于做任何除 R 以外的其他动作之后得到的进程,也不能做 T 动作,而且这以后对于做任何除 R 以外之后得到的进程,也不能做 T 动作,等等.

3.5 模型检查与协议分析

在上文中,我们定义了加密协议描述语言 PEP,并且说明 PEP 描述可以转化为基本 CCS 进程.另外,我们用模态逻辑中的 ν 公式刻画了主体动作的对应性.

在 CCS 中,一个进程模型是否满足模态公式,可以用模型检查算法来检验.我们因此就可以验证协议是否满足对应性了.目前,有许多工具都可以作此验证,并且当验证失败时,工具往往可以给出导致性质不成立的动作序列,对于我们分析的加密协议来说,此动作序列正是对应了攻击者进攻的路径.

4 Needham-Schroeder 公开密钥协议的分析

我们现在举例说明如何应用本文提出的方法来分析加密协议.我们的协议例子是简化的 Needham-Schroeder 公开密钥协议[5].

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

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

(3) $A \rightarrow B$: $A,B,\{N_b\}_{k_b}$

在此协议中,首先,协议发起方 A 选择一个随机量 N_a 与其标识一起用响应方 B 的公开密钥 k_b 加密后发送给 B;B 接收到消息并解密后,就得到了 N_a,随之也生成一个随机量 N_b,并与 N_a 一起用 A 的公开密钥 k_a 加密后发送给 A;A 从解密后得到的 N_a 试图判断与其通信的正是主体 B,同样地,A 把 N_b 加密后返回给 B;B 解密后试图判断与其通信的是主体 A.

这里,协议发起方 A 与协议响应方 B 可以与多个主体通信.为简单起见,我们考虑 A 与 B,C 通信的情况.

用 PEP 语言,此协议可描述如下:

```
A1=send(uval a,uval b,enc(kpub b,nval na1,uval a)).
   recv(uval a,uval b,enc(kpub a,nval na1,nvar x)).
   send(uval a,uval b,enc(kpub b,nvar x)).
   stop
A2=send(uval a,uval c,enc(kpub c,nval na2,uval a)).
   recv(uval a,uval c,enc(kpub a,nval na2,nvar x)).
   send(uval a,uval c,enc(kpub c,nvar x)).
   stop
B =recv(uval b,uval a,enc(kpub b,nvar y,uval a)).
   send(uval b,uval a,enc(kpub a,nvar y,nval nb)).
   recv(uval b,uval a,enc(kpub b,nval nb)).
   stop
```

这里,我们用 $A1$,$A2$ 分别表示 A 与 B,A 与 C 的局部通信协议.

初始环境为

$$S0=\{uval\ a,uval\ b,uval\ c,kpub\ a,kpub\ b,kpub\ c,kpvt\ c\}.$$

我们这里假设初始环境中包含主体 C 的私有密钥 $kpvt\ c$,这对应于主体 C 作为攻击者或者主体 C 的私有密钥泄露的情况.

协议的说明可写为

$$Spec=(A1+A2)|B \text{ in } S0.$$

主体 *B* 意想的通信对象是 *A*,因此当它做最后一个动作:

$$recv(uval\ b,uval\ a,enc(kpub\ b,nval\ nb)) \tag{1}$$

时,他必须确信此协议正是 *A* 发起的,即 *A* 做过

$$send(uval\ a,uval\ b,enc(kpub\ b,nval\ na1,uval\ a)) \tag{2}$$

动作.用对应性描述表示为

(1) authenticate (2).

我们用 ML 语言实现了 PEP 到 CWB 的 CCS 语言的转换程序.然后在 CWB 中,模型检查协议是否满足对应性,其结果是否定的,并且很容易就得到了攻击的途径:

send(uval a,uval c,enc(kpub c,nval na2,uval a))
recv(uval b,uval a,enc(kpub b,nval na2,uval a))
send(uval b,uval a,enc(kpub a,nval na2,nval nb))
recv(uval a,uval c,enc(kpub a,nval na2,nval nb))
send(uval a,uval c,enc(kpub c,nval nb))
recv(uval b,uval a,enc(kpub b,nval nb))

5 结　论

我们定义了加密协议描述语言 PEP 用于刻画一类不含消息变量的加密协议,避免了状态迁移图的无穷分叉.这样,协议的 PEP 描述就可以转化为有穷的基本 CCS 进程,并且可以在基于 CCS 的 CWB 工具中分析加密协议的性质.本文提出的方法与同类工作相比,其优点在于隐式地刻画攻击者的行为,可以通过模型检查发现协议潜在的安全漏洞,找到攻击协议的途径.

参考文献

1 Milner R. Communication and Concurrency. New York: Prentice Hall, Inc., 1989
2 Bruns G. Distributed Systems Analysis with CCS. London: Prentice Hall, Inc., 1997
3 Stirling C. Modal and temporal logics for processes. Technical Report, ECS-LFCS-92-221, Department of Computer Science, University of Edinburgh, 1992
4 Woo T Y C, Lam S S. A semantic model for authentication protocols. In: Proceedings of IEEE Symposium on Research in Security and Privacy. Oakland, CA: IEEE Computer Society Press, 1993
5 Lowe G. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In: Margaria T, Steffen B eds. Tools and Algorithms for the Construction and Analysis of Systems, vol 1055. Lecture Notes in Computer Science. Springer-Verlag, 1996. 147~166

CCS-based Cryptographic Protocol Analysis

DING Yi-qiang

(*Laboratory of Computer Science　Institute of Software　The Chinese Academy of Sciences　Beijing*　100080)

Abstract　Formal methods and tools are key aspects for the analysis of cryptographic protocols. In this paper, a formal language PEP (principals+environment=protocol) for the specification of cryptographic protocols is proposed. For some cryptographic protocols, their PEP specifications can be translated into finite basic CCS processes, so it is possible to analyze the security properties using CCS-based tools such as CWB (concurrency workbench). The advantage of the mothod proposed in this paper is that the actions of the attacker can be implicitly specified, and if the potential back door of the protocol analyzed exists, the attacking action trace can be explicitly found out by model checker.

Key words　Cryptographic protocols, protocol analysis, formal methods, CCS, model checking.