SPKI2.0安全性分析及改进

洪 帆 朱 贤 王绍斌

(华中科技大学计算机科学与技术学院 武汉430074)

摘 要 本文介绍了SPKI这种广泛使用的信任管理系统。从自主授权、传递授权和名字证书等方面,对其安全性进行了分析,指出将名字理解为角色的情况下,名字证书机制存在一定的安全问题。提出了对名字证书增加深度控制和将名字绑定改为不可传递两种改进方案。

关键词 SPKI,信任管理,名字证书,授权,角色

Security Analysis and Improvement of SPKI

Hong Fan Zhu Xian Wang Shaobin

(College of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074)

Abstract In this paper, SPKI is introduced, which is used in widespread area. The security properties are analyzed from the issues of discretionary authorization, transitive authorization and name certificate. The security flaws of name certificate are pointed out if name is treated as role. Two improving schemas, including adding depth control in name certificate and modifying the name binding to non-transitive, are proposed.

Keywords SPKI, Trust management system, Name certificate, Authorization, Role,

1 SPKI2.0介绍

SPKI2.0包括SPKI(simple public key infrastructure)和SDSI(simple distributed security infrastructure),是一种用于分布式访问控制的信任管理系统的标准[1]。目前,SPKI证书在许多范围内获得了支持,如SSL和IPSec等。SPKI和PKIX(IETF的标准,包含x.509PKI和PMI)是两种不同的方案,虽然SPKI不像PKIX,得到了较多的工业界的认同,但SPKI在分布式访问控制方面却有较多的优点[2]。

1.1 证书类型

SPKI中包括两种证书,授权证书和名字证书,使用S-EXPRESSION的形式表达。

授权证书表达权限的委托,是公钥或名字等与权限的绑定,如下面的证书(为说明方便,公钥仅用符号表达,如Kalice。并略去一些属性):

(cert (issuer Kalice)(subject (name "poker buddies"))(tag (play super-poker)))

Alice将权限"play super-poker"委托给名字(name Kalice "poker buddies")。

名字证书表达名字之间的映射关系,是名字与名字的绑定。如下面的证书:

(cert (issuer(name Kalice "poker buddies"))(subject(name Sam)))

Kalice将名字(name Kalice "poker buddies")授给名字(name Kalice Sam)。将上面两个证书结合起来,Sam拥有权限"play super-poker"。

授权证书和名字证书可分别抽象为:

5元组(Issuer, Subject, Delegation, Authorization, Validity dates)和

4元组(Issuer, Name, Subject, Validity dates),

其中的issuer为公钥,subject可以是公钥、名字或其它(很灵活)。

SPKI授权证书使用委托机制进行权限的传递,名字证书表达了名字之间的关系。多个证书可形成证书链。

1.2 复合名字

SPKI整合了SDSI的技术,其中最突出的是局部名字的概念,其对立于x.509全局名字。局部的名字最重要的一个优点是不易导致错误。局部名字的名字空间是定义此名字的公钥,与名字一起构成全资格SDSI名字。如(name Ka Alice),即为公钥Ka中定义的名字Alice。名字可链接成复合名字,如Alice定义了名字(name Kalice Bob),则可使用复合名字(name Ka Alice Bob)。在某此情况下,名字可理解为一个实体中定义的角色。

名字长度指公钥后名字的数量,用符号"||"表

洪 帆 教授,博导,研究方向为安全模型、密码学和网络安全。朱 贤 博士生,研究方向为安全模型。王绍斌 博士生,研究方向为网络安全和安全电子商务。

示。例如|Ka|=0,|Ka Bob|=1。

SPKI 证书中,授权证书的 issuer 长度必为0,而名字证书的长度必为1。对于 subject,则没有限制。

通过复合名字可大大提高了授权的灵活性。

1.3 证书归约

证书归约实际上是通过已有的证书得出一些结论的方法。

例如纯授权证书的归约:

有两个5元组:〈I1,S1,D1,A1,V1〉+〈I2,S2,D2,A2,V2〉

产生:〈I1,S2,D2,AIntersect(A1,A2),VIntersect(V1,V2)〉

当:AIntersect()和 VIntersect()都成功,S1 = I2

且 D1 = TRUE。

其中:Aintersect()和 VIntersect()分别是求权限和时间的交集的函数。

对于名字证书与授权证书一起的归约,通过1.1节的例子,已经说明。

1.4 证书查找算法

在不考虑元组中的 Delegation、Authorization 和 Validity dates 域的情况下,并假设证书序列的最前面为授权证书,可将上面的例子表达为(issuer,subject)形式,例如:

(Kalice,Kalice "poker buddies"),

(Kalice "poker buddies",Kalice Sam)。

注意,这里的 issuer 与1.1节的元组定义中的不同。对于5元组,此处相同。但对于4元组,相当于其中的(Issuer Name)。

有时也将(Kalice "poker buddies")表述为 Kalice.poker buddies,这时证书可描述为(Kalice.-poker buffies,Kalice.Sam)。

Elien 提出的 SPKI/SDSI 中的算法是一种集中式的算法[3]。它假定证书已经分发到了用户本地的机器上。用户为了访问某项服务,需要在本地证书集中找到一个证书链,证明其访问权限。算法使用归约方法,如有两个证书:〈K1 Student,K1 Jean Classmate〉,〈K1 Jean,K2〉,经过归约可得到推论:〈K1 Student,K2 classmate〉。

一系列证书进行归约时,每一步产生的 subject 称为工作 subject。算法本质上是工作 subject 根据证书进行前缀改写的过程。

证书也是一种推论,推论和推论归约也可产生推论。证书也被称为规则。如果允许推论任意进行归约,有时会产生无穷大的推论集合,造成算法的不可终止。因此,定义了有限闭包的概念:

规则集 R 的有限闭包(Γ)定义为:R 与使用下面方法构造的推论的并集:

给定两个推论{I1→S1,I2→S2}∈Γ,如果 I2是 S1的前缀,按归约方法进行归约,产生推论 I3→S3(I3=I1,S3=(S1,使用 S2替换其中的前缀 I2))∈Γ,当|S3|≤|S1|。

此定义实际上要求每次归约产生的 subject 长度为非递增的。

上面实际上集中于名字证书的归约算法,但将其简单进行扩充,即可用于 SPKI 所有证书类型的归约。

算法实际上是先基于本地证书集,产生有限闭包。然后,在其中查找是否有所需要的推论,即某用户是否有某访问权限,从而得到所需的证书链。

1.5 SPKI 的优点

·非常适合于分布式访问控制。支持系统不认知的主体访问系统;任何拥有对象的实体均可发布属性证书;授权数据分布,同时可保证其一致性;安全的委托链,可伸缩性强;有较完善的证书分布和查找算法。

·不需要 PKI 和 PMI 同时存在,就可表达授权,即只需要一个基础设施(infrastructure)。

·名字可理解为角色,非常适合基于角色的访问控制。

·局部名字不易导致错误。

2 安全性分析

分布式访问控制往往都采用委托的技术。Sand-Hu 给出的委托的定义是,系统中的一些活动实体将其拥有的一些权限授予其它活动实体,使后者代表前者执行一些功能[4]。

2.1 自主访问控制的问题

SPKI 可使任何拥有对象的实体发布属性证书。在分布式环境下,有多个安全域(以下简称为域)存在,每个域中资源的控制者是域。实际上可理解为每个域可自主地将其控制的资源上的访问模式授给其它用户。如果将分布式环境类比于主机环境,每个域类比于资源的拥有者,则可将 SPKI 的安全策略类比于自主访问控制策略。而自主访问控制策略有下面这些固有的缺陷[5,6]:

a. 自主性本身可能带来的一些安全问题。由于资源的拥有者(主体)可自主地将自己的资源(客体)上的访问模式授给其它用户(主体),因此,可能将一些资源授给了一些不需要这些资源的用户。

b. 用户容易无意或有意地使信息泄密。例如:用户可能将一些机密的文件的读权限无意中授给了其它用户,而后者不应该有读机密文件的权力。

c. 容易受到特洛伊木马攻击。

因为具有类比性,所以需要分析这些缺陷在 SPKI 中是否也会存在。

在分布式环境中，如何理解资源和资源的拥有者？

从个人计算环境来说，个人开发或发布的任何服务，或一个 CORBA 分布式对象，都可认为是一个资源。个人提供的所有资源此时构成一个域。资源的拥有者即为这个开发者或发布者。SPKI 可满足这一类需求。在 CORBA 或其它分布式对象技术中，对对象实施访问控制，可指定对象的所有者(owner)属性，将其绑定为一个公钥。那么，这个公钥对应的实体，是此资源的拥有者。也可理解为，此对象的访问控制验证器(verifier)信任这个实体对其访问，这种信任是本地可信的数据，在 SPKI 中称为访问控制列表(ACL)。在这种情况下，owner 将权限授给其信任的用户，不存在安全问题。

从企业计算环境来说，企业提供的所有服务都是资源，也构成一个域。资源的拥有者是企业[7]，这里可认为拥有者是域。由于域由其安全管理员来管理，因此，可认为安全管理员，代表域执行拥有者的职能。安全管理员将对象的访问权限授给用户，代表企业的需求或策略。从这点来看，具有一定的安全性。

根据上面的分析，a 问题不存在。

对于 b 和 c 问题，如果不使用多级安全技术，不可能解决这种问题。在分布式环境下，除非域成员都采用一致或可相互映射的安全级，否则不可能实现这种控制。而这一点通常是不可实现的，除非局限在某一些专用的分布式系统间，例如军方的分布式系统。

在企业计算环境下，对于 b 和 c 问题，由于授权者是系统安全员(对比于自主访问控制中的客体的 owner)，因此，其安全性可得到一定程度的保障。

本文后面讨论的均是企业计算环境。在这种环境下应用 SPKI，均应将域的安全员设为授权源(SOA)。

2.2 传递授权的问题

SPKI 支持传递授权的机制。任何用户通过证书，获得了授权。如果证书中允许其继续将权限委托出去，该用户就可将相应的权限委托给任何用户。问题在于：

a. 对于一个客体，即使知道有哪些主体可以控制其访问模式，但没有任何主体会负责该客体的安全[5]。

b. 传递不易控制。x 信任 y，但不能保证 y 不会滥用权限，将权限授给非法用户(对于 x 来说)。

对于上面的 a 问题，只要是可传递性的授权机制，且是复制型的(即授权者将权限委托出去后，授权者仍然拥有权限)，则必然存在这一问题。但也可理解为由多个主体同时负责该客体的安全。

b 问题涉及到信任的传递问题。对于信任本身，显然是不可传递的。如 x 信任 y，y 信任 z，但并不能推出 x 信任 z。可是将信任约束在一些特殊的概念内，例如委托，则可以传递。对于 b 问题，只要同时对委托加以了深度限制，则在一定程度上解决了这种问题。实际上委托的深度也表达了 issuer 对 subject 的信任程度[8]。如果 x 不信任 y 信任的用户，则 x 发布给 y 的证书中，将 delegation 域设为 false。这时，y 不能将 x 委托给他的权限委托给其它人。如果将 delegation 域设为 true，则表示 x 信任 y 信任的用户，当 y 将权限委托给 z 时，x 没有理由不信任 z。

2.3 名字证书的问题

名字证书是 SPKI 中最重要的一项技术，SPKI 提供的一些重要的特性均来自于此。在 SPKI 文档中，说明授权证书表达权限的传递。而名字证书不表达权限的传递，仅表示名字的绑定。

SPKI 作为一种分布式安全基础设施，可以支持很多类型的应用。如果将名字理解为角色，则可将 SPKI 用于基于角色的分布式访问控制。

在基于角色的策略中，权限实际上包含两种类型，即管理权和使用权。管理权是对相应角色分配用户的权限，而使用权是相应角色所拥有的权限。根据下面的分析，名字证书这时表达了权限的传递。

1)名字证书(A. R1，B. R2)实际上表达了下面的权限传递：

A 将 A. R1 角色的管理权委托给 B，并成为 B. R2的子角色(R1的使用权 R2都有，分配了 B. R2 角色的用户都拥有 A. R1角色的使用权)。但 B 不能将 A. R1的管理权直接委托出去，只能将 B. R2的管理权委托出去，此时 A. R1的管理权间接地被委托出去了。B 也不能直接使用对 A. R1的管理权，即 B 不能直接为 A. R1角色分配用户，只能为 B. R2角色分配用户，此时间接地为 A. R1角色分配了用户。

例如：当存在(A. R1，B. R2)、(B. R2，C)和(B. R2，D. R3)时，C 获得了 A. R1的使用权。D 获得了 B. R2的管理权，同时间接获得了 A. R1的管理权。

2)在权限传递方面，同授权证书的区别在于：

. 管理权和使用权的获得不是一致的。对于(A. R1，B. R2)，B 仅获得了 A. R1的管理权。但 B 可将 A. R1 角色的使用权，间接通过另一个证书(B. R2，C)授给 C。而授权证书这两者是一致的，即接受权限的实体，同时获得管理权和使用权。

. B 在委托 A. R1的管理权和对 A. R1分配用户时，不能直接使用 A. R1，只能通过 B. R2操作。

3)在 SPKI 之后出现了一些系统，如 RT0、dRBAC 等，并不使用授权证书，这些系统主要使用名字证书的思想，并将名字视为角色[9,10]。在这些系统

中，显然是用名字证书来表达权限的传递。

上面的分析中用到的例子，仅是SPKI中名字证书的一部分类型，而SPKI支持非常灵活的名字证书方案。但仅就上面的分析，已可说明名字证书表达了权限的传递。

由于SPKI中对名字的传递不进行深度控制，因此将名字理解为角色时，对其表达的权限传递未进行深度控制。当名字在一个域内传递时，即此时，subject中的公钥和issuer相同，此时，虽然不作深度限制，但其传递是安全的，相当于RBAC角色层次的实现。但当在域间传递时，即subject中的公钥和issuer不同时，如果不对其传递进行深度控制，则存在一定的安全问题。

3 安全性改进

经过上面的分析，对于名字证书，需要进行一定的改进，控制传递的深度。

最直接的改进方案是对名字证书增加深度控制，即在名字证书上添加一个Delegation域，其值可取true/false/-。"-"表示与其无关，当subject中的公钥和issuer相同，此域不起作用。但当不同时，则受此域控制。

例如一系列名字证书(A.R1,B.R2,false)、(B.R2,B.R3,-)、(B.R3,C.R4,true)、(C.R4,D.R5,flase)，可归约得到下面推论：

(A.R1,B.R3,false)、(B.R2,C.R4,true)、(B.R2,D.R5,false)、(B.R3,D.R5,false)。

虽然第一个证书设为不可委托，但第二个证书是同一个域内名字的绑定，因此可不受限制，仍可得到推论(A.R1,B.R3,false)。

归约不能得到(A.R1,C.R4,-)，因为推论(A.R1,B.R3,false)阻止了A.R1通过B.R3的传递。

但这种办法存在一个缺点。由于名字包含了本域内的权限，也包含其它域通过证书传递给它的权限，名字证书的签发者往往难以决定Delegation域的取值。例如上面例子中，C对C.R4定义名字证书时，难以决定Delegation的取值。因为C.R4包含了C域内的权限，也包括B.R3的权限，C可能想将C.R4本域内的权限委托出去，而同时不想将B.R3的权限委托出去。通过这种方法，无法实现这种控制。造成这种因难的本质原因是名字可能包含了多个域的权限，而这些权限是作为一个整体通过名字证书委托出去的。

另一种改进行方案是将域间的名字绑定定义为不可传递的，即当subject中的公钥和issuer不同时，不可传递。另外，为了保证名字证书的可伸缩性，增加可传递的名字委托机制，即将名字作为授权证书中的Authorization来处理，通过名字委托得到权限的实体，可在这个名字上定义名字证书。

例如一系列名字证书(A.R1,B.R2)、(B.R2,B.R3)、(B.R3,C.R4)、(C.R4,D.R5)可归约得到推论(A.R1,B.R3)、(B.R2,C.R4)。

而不能得到：(A.R1,C.R4)、(B.R2,D.R5)、(B.R3,D.R5)。

如果存在名字委托证书(用5元组表示)(B,E,true,B.R3,V1)，则E可发布名字证书(用4元组表示)(E,B.R3,D.R5,V2)。另外，E也可以自主发布托证书(E,F,true,B.R3,V3)。这实际上是第三方委托机制。

这种方案实际上使用名字委托机制，仅传递角色的管理权。而某实体只要通过名字委托得到了授权，就可以对别的域中的名字定义名字证书。另外，由于多域间名字证书的不可传递性，使得名字证书仅传递名字在其本域内的权限，就可以做到这一点。这样解决了第一种方案中本质性的问题。这种方案对SPKI作了较大的修改。

总结 SPKI发布之后，很多新的系统采取了其中的思想，如RT0、dRBAC等。这此系统主要采用SDSI的思想，因此同样有名字证书的问题。本文首先对SPKI相关技术进行了介绍，然后从自主授权、传递授权和名字证书等方面，对SPKI的安全性进行了分析，指出将名字理解为角色的情况下，名字证书机制存在一定的安全问题。提出了对名字证书增加深度控制，和将名字绑定改为不可传递两种改进方案。

参考文献

1 Ellison C, Frantz B, Lampson B, Rivest R, Thomas B, Ylonen T. SPKI Certificate Theory. RFC 2693. 1999

2 Nykā nen T. Attribute Certificates in X.509. http://www.hut.fi/~ tpnykane/netsec/complete/toni_ac_complete.pdf

3 Elien JE. Certificate Discovery Using SPKI SDSI 2.0 Certificates. Master thesis at M.I.T ECSE. 1998

4 Barka E, Sandhu R. A Role-Based Delegation Model and Some Extensions. In: Proc. of 23rd National Information Systems Security Conf. (NISSC 2000). 2000

5 陈爱民，于康友，管海明. 计算机的安全与保密. 电子工业出版社，1992. 164～166

6 Clark D D, Wilson D R. A comparison of commercial and military computer security policies. In: Proc. of 1987 IEEE Symposium on Security and Privacy. 1987. 184～194

7 Sandhu RS, Conyne EJ, Lfeinstein H, Youman CE. Role based access control models. IEEE Computer, 1996, 29(2): 38～47

8 Ninghui L. Delegation Logic_A Logic-based Approach to Distributed Authorization. ACM Transactions on Information and System Security, 2003, 6(1): 128～171

9 Ninghui L, Winsborough WH, Mitchell JC. Distributed Credential Chain Discovery in Trust Management (Extended Abstract). In: Proc. 8th ACM Computer and Communication Security (CCS01). 2001. 156～165

10 Freudenthal E, Pesin T, Port L, Keenan E, Karamcheti V. dRBAC: Distributed Role-based Access Control for Dynamic Coalition Environments: [Technical Report TR2001-819]. New York University. 2001