基于量化权限的门限访问控制方案*

雷 浩[1,2+], 冯登国[1], 周永彬[1], 黄 建[3]

[1](中国科学院 软件研究所 信息安全国家重点实验室,北京 100080)

[2](中国科学院 研究生院,北京 100039)

[3](牛津大学 计算实验室,伦敦 OX1 3QD,英国)

Threshold Access Control Scheme Based on Quantifying Permission

LEI Hao[1,2+], FENG Deng-Guo[1], ZHOU Yong-Bin[1], HUANG Jian[3]

[1](State Key Laboratory of Information Security, Institute of Software, The Chinese Academy of Sciences, Beijing 100080, China)

[2](Graduate School, The Chinese Academy of Sciences, Beijing 100039, China)

[3](Computing Laboratory, Oxford University, London OX1 3QD, England)

+ Corresponding author: Phn: +86-10-62528254 ext 803, E-mail: leihao00@iscas.cn, http:// www.is.iscas.ac.cn

Received 2003-06-09; Accepted 2004-07-06

Lei H, Feng DG, Zhou YB, Huang J. Threshold access control scheme based on quantifying permission. *Journal of Software*, 2004,15(11):1680~1688.

http://www.jos.org.cn/1000-9825/15/1680.htm

Abstract: Secret protection is studied after introducing the quantifying permission idea in the view of access control. Meta-Permission is derived from the view of 'quality' and 'quantity' in philosophy. Compared with traditional cognition and permission used in access control, it comprehensively and deeply reflects the essence of permission. Combining with the threshold idea and RBAC, a threshold access control scheme based on quantifying permission is proposed. This scheme is applicable as widely as secret-sharing scheme, and shares the same function in protecting secret. Furthermore, it has some distinct advantages such as no relation in knowledge between the secret pieces hold by participants and the protective secret target, the ability for expressing the difference among participants' trustworthiness, and the low complexity in computing.

Key words: quantifying permission; threshold scheme; meta-permission; role

摘 要: 研究了在引入量化权限观点后从访问控制角度实现秘密保护的问题.元权限是从哲学上"质"和"量"的角度认识传统意义上的权限所探究出的新概念,较以往访问控制中认识和使用权限而言,它全面而深入地反映了权限这一概念的本质.进一步结合门限思想和基于角色的访问控制机制所提出的基于量化权限的门限访问控制方案,从

* Supported by the National Natural Science Foundation of China under Grant No.60273027 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2002AA141080 (国家高技术研究发展计划(863)); the National Outstanding Young Scientists Foundation of China Under Grant No.60025205 (国家杰出青年科学基金)

作者简介: 雷浩(1975－),男,陕西合阳人,博士生,主要研究领域为系统安全,信息安全理论与技术;冯登国(1965－),男,研究员,博士生导师,主要研究领域为密码学,信息安全;周永彬(1973－),男,博士,主要研究领域为应用密码学,网络与信息安全理论与技术;黄建(1976－),男,博士生,主要研究领域为信息安全技术.

访问控制的角度研究了秘密保护问题.在秘密保护方面,基于量化权限的门限访问控制方案具有一些独特的优点,比如分发给参与者的秘密分片和要保护的秘密无知识上的联系、可以反映出参与者信任度的差异以及运算量低.

关键词: 量化权限;门限方案;元权限;角色

中图法分类号: TP309 **文献标识码**: A

1 问题的提出

秘密保护是信息安全领域中的一个重要问题.Shamir[1]于 1979 年提出用门限思想来保护秘密.其基本观点是,将秘密信息 S 本身分割成 n 个秘密分片(shadows): $s_1,s_2,...,s_n$,然后分配给 n 个参与者.需要时可以利用任意 t 个秘密分片恢复出原秘密 S,当利用任意少于 t 个秘密分片时则由于信息短缺而不能计算出秘密 S.现在,已经运用门限思想利用诸如代数方法[1]、组合方法[2,3]和几何方法构造出了大量的适用于各种不同环境的秘密共享方案,诸如有欺骗者参加的秘密共享、可验证的秘密共享以及带预防性质的秘密共享方案.代数方法构造的典型方案有 Shamir[1]于 1979 年基于拉格朗日内插多项式以及 Asmuth 等人在 1980 年基于中国剩余定理提出的秘密共享方案;关于组合方法,Blundo[2]在 1993 年、HM Sun[3]于 1997 年,分别从图论的角度构造出了秘密共享方案;关于几何方法,Blakley 在 1979 年基于射影几何的方法构造出了一个秘密共享方案.另外,关于有欺骗者参加的秘密共享方案的最新进展,Chang[4]在 2000 年提出了如何防止秘密分发者欺诈,Fei[5]在 2003 年提出了如何识别非法参与者参与恢复秘密;关于可验证的秘密共享,最新的成果是 Cramer[6]在 2000 年基于计算复杂性所提出的秘密共享方案;关于带预防性质的秘密共享方案是 Stinson[7]在 1999 年提出的.

上述已有方案存在着以下 3 点不足:

(1) 分发给参与者的秘密分片与要保护的秘密存在着知识上的联系.比如,Shamir 从基于拉格朗日内插多项式(代数方法)构造的秘密分享方案(其他方案类似): $F(x)=a_{m-1}x^{m-1}+...+a_1x+a_0\in Z_p[x]$.其中, $a_0=S$ 是要保护的秘密.分发给用户的秘密分片是该多项式上的 n 个不同点: $y_i=F(x_i)$.在基于这个方法构造的 (t,n) 秘密分享方案中,通过任意 t 个秘密分片都可以建立 t 个方程,从而决定了该方程的 t 个系数,也就决定了秘密 S.也就是说,分发给用户的秘密分片 (x_i,y_i) 与要保护的秘密 $a_0=S$ 之间存在着知识上的联系.

(2) 秘密分片对于保护秘密的作用相同.文献[8]指出,在用户之间普遍存在着信任度和可靠性的差异.仍然以上述方案为例,假定用户 A 是将军,用户 B 是上校,要在他们之间分发关于导弹发射的口令.显然,将军的信任度要比上校的信任度高许多.但分发给将军的多项式上的点 (x_a,y_a) 与分发给上校的多项式上的点 (x_b,y_b) 对于恢复秘密(发射口令)而言,所起的作用相同,因而没有反映出参与者的信任度这一差别.

(3) 基于代数方法、组合方法和几何方法普遍存在着分发/恢复秘密的时候计算复杂度偏高的问题.

要做到分发给用户的秘密分片与要保护的秘密之间不存在任何知识上的联系,以及秘密分片反映参与者的信任度,从访问控制的角度来看,由秘密保护问题萌发出的一种新的思路是:将秘密看作客体(比如文件)所包含的内容,客体则视为包含秘密信息的容器,根据每个参与者的信任度和可靠性划分出权限片段(权限片段也就是“元权限”,其值的大小反映了用户之间的这种差异),并对量化分割后的权限片段应用基于门限思想的访问控制,从而达到保护秘密的目的.本文研究了一种保护秘密的新技术,该技术基于量化权限思想和角色访问控制(role based access control,简称 RBAC),并结合门限思想来实现秘密保护.

2 权限的量化属性与 RBAC 管理权限的优点

2.1 从质和量的角度认识权限

任何事物都是“质(quality)”和“量(quantity)”的统一体,权限也是如此.本质上,它不但具有“质”的属性,同时还有“量”的属性.举例来说:

例 1:在商业领域中常常有这样的规定:有关公司的重大决策(比如重大的商业投资),必须经过包括董事长在内的半数以上的董事会成员一致通过,方才有效.

例 2:在政治生活领域中,关于总统选举,宪法明确规定:年满 18 周岁的公民均有选举权,但必须至少有一半以上的公民参与投票,选举方才有效,

例 3:在军事领域中有类似的更为严格的规定:

1) 1 名将军和两名上校可以共同发射一枚导弹;

2) 4 名上校一起可以发射一枚导弹.

为什么需要多个相关人员才可以共同完成一项任务呢?究其原因,是现实生活中的各个领域内存在一些特别重要的问题,以至于级别再高的个人也无法满足该问题对权限的“量”的要求.也就是说,权限不但有“质”的属性,同时还有“量”的属性.然而,迄今为止,人们只看到了权限所拥有的“质”的属性,而忽视了其本身还具有的“量”的属性.

2.2 RBAC管理权限的优点

RBAC 中的基本元素包括权限、角色和用户.其基本思想是,把权限分配给适当的角色,然后用户通过角色来获得所需的权限.一般而言,一个角色所拥有的权限是和某一个确定的任务紧密相关的.这样,可以根据任务需要创建适当数量的角色,以之为单位在用户之间实现权限的分配和共享.换而言之,RBAC 通过对角色的配置和再配置,为管理大量客体的访问权限提供了一种灵活的、动态的方法,因而近年来得到了极大的发展.文献[9]中规范了 NIST 标准 RBAC 模型,统一了人们对 RBAC 的认识;文献[10,11]分别扩展了 RBAC 模型的描述能力和适用范围.

RBAC 中允许角色之间存在继承关系.当从一个角色(在继承关系下,称为“低级角色”)派生出另一个角色(在继承关系下称为“高级角色”)时,高级角色就拥有了低级角色的全部权限.这样,当出现对权限要求较高的任务时,可以从低级角色中派生出高级角色,并给高级角色增加新的权限片段,然后把高级角色授予高级别的用户.以发射导弹为例,“上校”是低级角色,它只有发射导弹的一个权限片段;“将军”是从“上校”派生出来的高级角色,它首先继承了“上校”发射导弹的一个权限片段,然后系统管理者再给它赋予两个发射导弹权限片段,也就是说,“将军”这个角色拥有发射导弹的 3 个权限片段,他的元权限为(“发射导弹”,3);如果发射导弹的门限值为 4 个权限片段,人数需求为 3,那么 4 个上校可以共同发射导弹,一个将军和两个上校也可以共同完成发射导弹任务.通过这个例子可以看出,利用角色的继承关系,可以方便地实现在有高级用户在场的情况下,减少对人数的需求;而规定人数的需求,则降低了共谋得逞的可能性.

3 基于量化权限的门限访问控制体系结构

基于量化权限和门限思想的秘密保护方案是从访问控制的角度来研究保护秘密问题的.与已有的从代数方法、组合方法和几何方法构造的秘密共享机制不同,分发给用户的控制分片与要保护的秘密之间没有知识上的联系.它首先根据每个参与者的信任度和可靠性,从第 2.1 节量化权限的思想出发,将存有秘密的某一具体客体的访问权限用三元组(客体,权限质,权限量)来刻画,并将此三元组称之为“元权限”;然后把关于包含秘密的客体元权限授予多个角色,角色数目的多少取决于参与者之间的信任度差异层次数,而且角色可以通过继承的方式来产生较大的元权限,并把每个角色授予合适的既定用户;用户的元权限值是通过其所拥有的角色动态计算的,系统只维护“元权限-角色列表(MR)”以及“角色-用户授权列表(RU)”;当用户要求通过行使权限访问秘密的时候,必须满足事先规定的人数以及权限“量”的需求.如此,可以避免秘密本身的分割与分发.而且,当剔除有恶意用户的时候,只要收回他所拥有的访问该客体的角色,就可以方便地剔除有恶意的用户;再者,不需要重构秘密,就可以方便地实现保护秘密的目的.图 1 反映了这一思想.

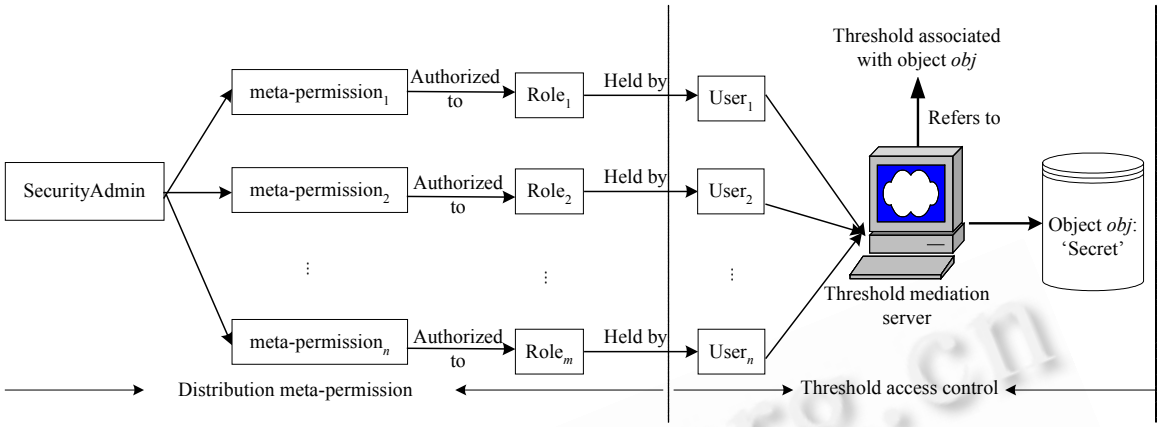

Fig.1 Secret protecting achitecture based on quantifying permission and RBAC

图1 基于量化权限和RBAC的保护秘密系统结构

3.1.1 门限裁决服务器(threshold mediation server)

所有要保护的秘密都以文件的形式保存在门限裁决服务器上,每个秘密对应一个文件.由于拥有元权限的用户是分布的,因而基于量化权限的门限访问控制方案重点要解决的是用户的身份认证(identity authentication)、通信过程中信息的不可篡改以及防止消息的重放攻击问题.

随着 PKI 技术的日益成熟和普及,基于证书的认证体制[12,13]在解决用户分布而管理集中的访问控制获得普遍认可.在此假定每个用户(包括下面的3个管理员)都向认证机构(certificate authority,简称CA)申请了证明自己身份的证书(certificate identity),从而每个用户都有可以进行信息交互的公、私钥对 (K_x, K_x^{-1});系统分别设置系统管理员(SysAdmin)、安全管理员(SecurityAdmin)和审计管理员(AuditAdmin),三者均为可信的.其中,安全管理员负责用户和角色的创建与删除,安全管理员负责给角色授予/收回访问客体的元权限以及将角色授予/收回秘密分享者(用户),并判定参与者是否达到门限的访问要求;审计管理员则负责系统中与安全有关的事件的记录.对三者的职责划分符合"最小特权原则",并在三者之间相互监督和制约.

3.1.2 元权限与同质元权限

若 OBJ 表示服务器中所有客体(文件)的集合,OP 表示可以施加在客体上的访问操作集合,则元权限集为 $MetaPerms = \{(obj, op, i) \mid obj \in OBJ, op \in OP, i \in N\}$.元权限 (obj, op, i) 是量化权限下关于存有秘密客体 obj 的某一访问方式 op 授权时的基本单位.其中,obj 和是元权限的质,并称 obj 和 op 都相同的元权限为"同质元权限";i 是元权限的量,同质元权限的量彼此不同,其值大小反映了不同用户的信任度和可靠性差异;只有同质元权限才可以在角色间继承、相加.函数 1 给出了创建元权限的过程.

函数 1. $CreateMetaPerm: OBJ \times OP \times N$ 创建关于存有秘密的客体 obj 的访问方式 op 的一个元权限.

$CreateMetaPerm(obj: OBJ, op: OP, i: N) \triangleleft$

$meta - perm = (obj, op, i);$

if $(meta - perm \notin MetaPerms)$

$MetaPerms' = MetaPerms \bigcup \{meta - perm\} \triangleright$

删除一个元权限的函数在"角色拥有的权限集合"后给出.

3.1.3 角色的元权限计算

系统需要维护元权限-角色列表 MR.若系统中所有角色的集合记为 $Roles$,则 $MR \subseteq MetaPerms \times Roles$ 是从元权限集合到角色集合的多对多映射,表示角色被赋予的元权限,$MR(\rho(Roles))$ 则表示角色集合 $\rho(Roles)$ 中每个角色所拥有的元权限列表集合.授予角色元权限的过程如函数 2 所示.

函数 2. $AuthorizeMetaPermtoRole: Roles \times MetaPerms$ 给角色 r 赋予元权限 $meta\text{-}perm$.

$$AuthorizeMetaPermtoRole(r:Roles, meta-perm:MetaPerms)\triangleleft$$
$$\text{if } (meta-perm\in MetaPerms\wedge meta-perm\notin MR(\{r\}))$$
$$MR(\{r\})'=MR(\{r\})\bigcup\{meta-perm\}\triangleright$$

考虑到角色间有继承关系,因而将角色的元权限数量定义为继承自低级角色的元权限数量与安全管理员授予给该角色的元权限数量之和;在角色间存在多继承的情况下,角色所继承的元权限数量是该角色继承的所有低级角色的元权限的最大值.这样做可以充分利用角色的配置和再配置特点,从而方便授权管理.函数 3 和函数 4 分别计算角色关于客体 *obj* 上的访问方式 *op* 继承来的元权限以及安全管理员所授予的元权限;函数 5 计算该角色关于此的总的元权限数量;函数 6 是安全管理员从角色收回关于客体 *obj* 的访问方式 *op* 的元权限.

函数 3. $GetRoleInheritMetaPerm:Roles\times OBJ\times OP$ 计算指定角色 r 从其祖先角色集合 $Ancestor(r:Roles)$ 中继承过来的、关于客体 *obj* 的某一具体访问方式 *op* 的最大的元权限:

$$GetRoleInheritMetaPerm(r:Roles,obj:OBJ,op:OP)\triangleleft$$
$$AncestorMetaPermSet=\varnothing$$
$$AncestorSet=Ancestor(r)$$
$$AncestorMetaPermSet=\{(obj,op,i)\mid(obj,op,i)\in MR(AncestorSet)\}$$
$$\text{if } (AncestorMetaPermSet==\varnothing), output=(obj,op,0)$$
$$\text{else return } output=(obj,op,i_{\max})\triangleright$$

其中,$(obj,op,i_{\max})$ 是 $AncestorMetaPermSet$ 中满足如下性质的元权限:

$$(obj,op,i_{\max})\in AncestorMetaPermSet\wedge(\forall(obj,op,i')\in AncestorMetaPermSet\Rightarrow i'\le i_{\max}).$$

函数 4. $GetRoleAuthorizedMetaPerm:Roles\times OBJ\times OP$ 计算角色 r 由安全管理员所授予的、关于客体 *obj* 的某一具体访问方式 *op* 的元权限:

$$GetRoleAuthorizedMetaPerm(r:Roles,obj:OBJ,op:OP)\triangleleft$$
$$\text{if } (\exists(obj,op,i)\in MR(\{r\}))$$
$$\text{return } (obj,op,i)$$
$$\text{else return } (obj,op,0)\triangleright$$

函数 5. $GetRoleMetaPerm:Roles\times OBJ\times OP$ 返回角色关于客体 *obj* 访问方式 *op* 的元权限,该元权限的数量包括继承来的以及安全管理员授予的:

$$GetRoleMetaPerm(role:Roles,obj:OBJ,op:OP)\triangleleft$$
$$(obj,op,sum)=GetRoleInheritMetaPerm(role,obj,op)+GetRoleAuthorizedMetaPerm(role,obj,op)$$
$$\text{return } output=(obj,op,sum)\triangleright$$

其中,元权限的相加必须是同质的,必须遵循以下规则:

$$SumMetaPerm((obj_1,op_1,i_1):MetaPerms,(obj_2,op_2,i_2):MetaPerms)\triangleleft$$
$$\text{if } ((obj_1==obj_2)\wedge(op_1==op_2))$$
$$\text{return } output=((obj,op,i_1+i_2)=(obj_1,op_1,i_1)+(obj_2,op_2,i_2))\triangleright$$

函数 6. $RevokeMetaPermFromRole:MetaPerms\times Roles$ 从角色 r 所拥有的元权限集合中收回关于客体 *obj* 的访问方式 *op* 的元权限 *meta-perm*.

$$RevokeMetaPermFromRole(r:Roles,meta-perm:MetaPerms)\triangleleft$$
$$\text{if } (meta-perm\in MR(\{r\}))$$
$$MR(\{r\})'=MR(\{r\})\setminus\{meta-perm\}\triangleright$$

在此给出删除关于客体 *obj* 的访问方式 *op* 的元权限 *meta-perm* 的函数 7.为了保持系统的一致性,对于拥有该元权限的角色,也从该角色拥有的元权限集合中删去.

函数 7. $DeleteMetaPerm(meta-perm:MetaPerms)$.

$$
\begin{aligned}
&DeleteMetaPerm(meta-perm:MetaPerms)\triangleleft\\
&\quad \text{if } (meta-perm\in MetaPerms)\\
&\qquad \forall r\in Roles\\
&\qquad \text{if } (meta-perm\in MR(\{r\}))\\
&\qquad MR(\{r\})'=MR(\{r\})\setminus\{meta-perm\}\\
&\qquad MetaPerms'=MetaPerms\setminus\{meta-perm\}\triangleright
\end{aligned}
$$

3.1.4 用户的元权限计算

系统中用户的集合记为 *Users* .拥有关于客体 *obj* 某一访问方式 *op* 的元权限可以是多个角色,为了防止出现元权限量过大的用户,规定每个用户只能被授予其中的一个角色,并定义用户的元权限为该角色元权限数量的大小;如果用户所拥有的角色集合中没有角色有此秘密权限,则返回 $(obj,op,0)$.用户的元权限量是在判定访问是否合法的时候,通过用户所拥有的角色动态计算的,系统并不维护元权限-用户列表.这样做的好处在于,安全管理员只需配置合适类型的角色,通过给用户授予/收回角色就可以实现权限的分配与回收,从而方便门限方案的更改.

函数 8. $GetUserMetaPerm:Users\times OBJ\times OP$ 返回用户关于客体 *obj* 某一访问方式 *op* 的元权限.

$$
\begin{aligned}
&GetUserMetaPerm(user:Users,obj:OBJ,op:OP)\triangleleft\\
&\quad role=RU(user,obj,op)\\
&\quad \text{if } (role=\varnothing)\ \text{return } (obj,op,0)\\
&\quad \text{else return } GetRoleMetaPerm(role,obj,op)\triangleright
\end{aligned}
$$

其中, $RU\subseteq Roles\times Users$ 是从角色集合到用户集合的多对多映射,表示用户被赋予的角色, $RU(user,obj,op)$ 则表示用户 *user* 所拥有的角色集合中,关于拥有客体 *obj* 某一访问方式 *op* 的角色.

3.1.5 客体访问时的门限判定

对客体访问所设置的权限门限,是结合门限思想,对保存秘密的客体所设置的权限保护关卡.不同的客体,其门限值是不同的.将访问客体的权限门限定义为 $T=\{obj\times op\times m\times d\mid obj\in OBJ,op\in OP,m\in N,d\in N\}$,门限四元组 (obj,op,m,d) 表示一组用户对客体 *obj* 行使访问方式 *op* 时必须达到权限量 *m* 以及人数 *d* 的要求,并且访问裁决是否满足门限值由门限裁决服务器来完成.

具体的判断函数如下:

函数 9. *bool IsSatisfied* : $Sessions\times OBJ\times OP\times N\times N$ 判断参与访问的一组用户能否对客体 *obj* 行使访问方式 *op* .把该组用户对客体的访问过程(包括发起访问请求、裁决是否达到门限值以及执行访问)看成是“会话(sessions)”.

$$
\begin{aligned}
&bool\quad IsSatisfied(s:Sessions,obj:OBJ,op:OP,m:N,d:N)\triangleleft\\
&\quad ParticipantsSet=session-users(s)\\
&\quad \forall u\in ParticipantsSet\\
&\quad \text{if } (GetUserMetaPerm(u,obj,op)==(u,obj,0))\\
&\quad /*\text{访问请求者集合中现没有相应元权限的欺骗者}*/\\
&\qquad \text{return false}\\
&\quad \text{if } (TotalMetaPerm(s,obj,op)<m)\vee(\#ParticipantsSet<d)\\
&\quad /*\text{不符合门限访问要求}*/\\
&\qquad \text{return false}\\
&\quad \text{else}\\
&\qquad \text{return true}\triangleright
\end{aligned}
$$

其中:

(1) 子函数 *session-users*($s:Sessions$) 返回参与会话 *s* 的所有用户, $\#ParticipantsSet$ 表示参与者的人数.

(2) 子函数 $TotalMetaPerm(s:Sessions,obj:OBJ,op:OP)=\sum_{u\in session-users(s)}GetUserPerm(u,obj,op)$ 计算参与会话 s 的所有用户关于客体 obj 的访问方式 op 的元权限值之和.

下面以 U_1 "读"客体 O 为例,给出这一过程,其中, K_x^{-1} 表示用户 X 的私钥, IC_x 表示 X 的身份证书.具体访问客体的交互过程如下:

1. 用户 U_1 向门限裁决服务器提交请求访问客体的消息.该消息包括目标客体及其访问方式、提出访问请求的原因,还有 U_1 产生的一个随机数(nonce) N_{U_1} ,并用 U_1 自己的私钥加密后传送,并附上身份证书:

$$U_1 \rightarrow SysAdmin \quad request:[\text{read object } obj,\ \text{`request reason'},N_{U_1}]K_{U_1}^{-1}+IC_{U_1}.$$

2. 系统管理员通过身份证书认证其身份,安全管理员通过系统维护的"角色-用户列表(RU)"以及"元权限-角色列表(MR)"核实其权限,审计管理员则记录本次事件;若身份或者权限检查不通过,则拒绝.

3. 以上都无误后,安全管理员首先产生一个随机数 N_{sec} ,然后通过检查系统维护的"角色-用户列表"以及"元权限-角色列表(MR)",向所有拥有同质元权限的其他 $n-1$ 个用户转发 U_1 的访问请求消息,并用自己的私钥加密后传送:

$$SecAdmin \rightarrow U_i(2\le i\le n)\quad request:[U_1\ \text{request to read object } obj,\ \text{`request reason'},N_{\text{sec}}]K_{\text{sec}}^{-1}.$$

4. 每个用户收到安全管理员签发的 U_1 的访问请求消息以后,返回响应消息.该消息包括:同意与否的态度以及步骤 3 中安全管理员发送给他们的随机数 N_{sec} ,然后用自己的私钥加密后传送,并附上每个用户自己的身份证书:

$$U_i(2\le i\le n)\rightarrow SecAdmin \quad response:[\text{`yes'}/\text{`no'},N_{\text{sec}}]K_{U_i}^{-1}+IC_{U_i}.$$

5. 安全管理员汇总所有 $n-1$ 个意见,判断同意的用户元权限之和是否达到门限值以及用户的人数是否满足规定的人数;然后将判定的结果以及 U_1 原先请求消息中的随机数 N_{U_1} 用自己的私钥加密后发送给 U_1 , U_1 遵从此判定结果:

$$SecAdmin \rightarrow U_1 \quad result:[\text{`yes'}/\text{`no'},N_{U_1}]K_{\text{sec}}^{-1}.$$

4 对比与讨论

4.1 表达功能与保护特点的比较

4.1.1 元权限与普通权限的对比

元权限是从哲学上"质"和"量"的角度认识传统意义上的权限,由此探究出的概念.较之以前访问控制中的认识和使用权限,它全面而深入地反映了权限这一概念的本质.下面以"读"为例,比较元权限与普通权限之间在表达功能方面的差别.

- 表达形式.元权限的表达形式是(客体,读,权值);普通权限的表达形式是(客体,读).
- 表达功能.元权限表达功能的优势在于:① 可以通过权值反映出用户信任度的差异;② 可以结合门限思想,从门限的角度提供对访问客体更严格的保护.而普通权限仅仅是元权限的一种特殊形式,刻画能力相对较弱.

4.1.2 已有的秘密保护方案与基于量化权限的门限控制方案的对比

已有的秘密保护方案(以下称前者)分发给用户的秘密分片与要保护的秘密之间有知识上的联系,而基于量化权限的门限控制方案(以下称后者)中的元权限与要保护的秘密之间没有任何知识上的联系.下面总结两者之间的差异:

- 分发给用户的分片.前者分发的是秘密的分片,后者分发的是元权限.
- 分发的分片与要保护的秘密有无知识上的联系.显然,前者有知识上的联系,而后者则没有.
- 每个分片在获取秘密时所起的作用.前者是相同的;后者根据元权限值大小的不同,所起的作用也不同.
- 是否需要恢复秘密.由于前者分割了秘密,故需要恢复秘密;后者本身并没有分割秘密,分割的是访问秘密的权限,因而不需要恢复秘密.

• 不能获取的原因与获取秘密的方法.前者是由于缺少恢复秘密所需要的知识,恢复秘密的方法是关于代数、组合以及几何方法的计算;后者则是由于缺少访问保存客体的访问权限或者参与人数,其恢复的方法是元权限的简单相加.

4.2 安全功能讨论

1. 基于量化权限的门限访问控制方案不存在“可信分发问题”与“可信验证问题”.

在通常的秘密保护方案中,由于分发给用户的秘密分片与要保护的秘密存在知识上的联系,因此在秘密分割的过程必定需要可信第三方来分发秘密,这样会把秘密暴露给分发者(dealer);而在秘密重构的过程中拥有秘密片段的所有者必须向验证者(combiner)出示其所拥有的共享片段,以便重构秘密,由于验证者要重构秘密,也必然会把秘密暴露给验证者.这在文献[14]中称为“可信分发问题”和“可信验证问题”.虽然文献[15]提出了不需要可信第三方帮助的方法,但在实际应用当中过于复杂.

在基于量化权限的门限访问控制方案中,将秘密以文件的形式保存在门限裁决服务器上,由于分发给参与者的是与要保护的秘密没有知识联系的元权限,而不是以往与要保护的秘密有联系的秘密分片,门限控制的载体是元权限而不是秘密分片.也就是说,门限思想是依托于元权限,而不是秘密分片得到实现的.因此,分割权限以及验证一组参与者的元权限之和是否满足要求的安全管理员并不会接触到秘密本身,故不会暴露秘密给分发者和验证者.

2. 基于量化权限的门限访问控制方案无须重构秘密分片便可以验证分片的有效性.

在通常的秘密保护方案中,由于分发给用户的秘密分片与要保护的秘密存在知识上的联系,因此每个参与者验证自己所持有的秘密分片是否有效的办法是重构整个秘密,而在基于量化权限的门限访问控制方案中无须重构秘密便可以实现验证.这是因为,在基于量化权限的门限访问控制方案中,分发给参与者的是与要保护的秘密没有知识联系的元权限,而不是以往与要保护的秘密有联系的秘密分片,门限控制的载体是元权限而不是秘密分片.因此,每次获取秘密时不需要重构秘密,只需验证一组用户的元权限值之和是否达到权限所规定的门限值以及参与访问的人数是否达到 d 个人.这与从数学方法上计算多项式系数来恢复秘密相比,无疑效率可以提高很多.

3. 基于量化权限的门限访问控制方案容易实现拥有访问秘密权限用户的增加与剔除.

在通常的秘密保护方案中,由于每个用户持有的秘密分片与要保护的秘密存在知识上的联系,因而将持有者除名非常困难,它涉及到更换多项式方程组等复杂的数学过程.但在基于量化权限的门限访问控制方案中,可以轻易地实现这一点.在基于量化权限的门限访问控制方案中,解除一个用户的访问秘密权限,只需从该用户收回拥有关于客体 obj 的访问方式 op 的元权限角色.进一步地,如果将一个 (k,n) 门限方案扩展成一个 (k_1,n_1) 门限方案,只需将原有的 (k,n) 门限方案删除,并从原先拥有元权限的 n 个用户收回关于客体 obj 的访问方式 op 的角色,将这些角色重新授予 n_1 个用户即可.

4. 任何 (k,n) 秘密分享方案可以在基于量化权限的门限访问控制方案中得到解决.

先将秘密以文件的形式保存在门限裁决服务器上,由安全管理员(或者秘密原始属主)确定出门限 (obj,op,m,d),其中 obj 是保存秘密的客体,op 是关于该客体的访问方式,比如“读”,m 是访问必须达到的权限门限值,d 是每次访问时要求的参与人数;然后创建合适大小的元权限集合(所谓合适大小,是对应于用户的信任度和可靠性),将此集合中的元权限分别授予不同角色,再把这些不同的角色授予 n 个不同的用户,要求行使该权限访问秘密的时候至少要有这 n 个用户中的 d 个人,而且这些用户的权限数量之和必须大于门限值 m .

5. 对于保护秘密,基于量化权限的门限访问控制方案是完善的.

所谓完善,就是指少于门限所要求的参与者,无法恢复出关于秘密的半点信息.由判断函数 9 可知,如果一组用户的元权限值之和没有达到权限所规定的门限值或者参与访问的人数小于 d 个人,则无法访问到秘密,因此得不到有关秘密的任何信息.所以,该方案是完善的.

在通常的秘密共享方案中,可能存在以下的骗子行为:通过故意出示错误的秘密分片,造成恢复过程中知识上的错误,从而阻止恢复秘密的不可行.

6. 基于量化权限的门限访问控制方案可以防止上述骗子行为.

在基于量化权限的门限访问控制方案中,元权限由安全管理员产生,经由角色授予参与者.一旦用户通过角色获得相应的元权限,该用户就无法修改系统所维护的"角色-用户列表(RU)"或伪装自己没有该权限,因而可以阻止这种骗子行为.

5 总 结

元权限是从哲学上"质"和"量"的高度认识传统意义上的权限,由此探究出的概念.较之以前访问控制中认识和使用权限,它全面而深入地反映了权限的本质.进一步结合门限思想和基于角色的访问控制机制所提出的门限访问控制方案,从访问控制的角度研究了秘密保护问题.与已有的秘密分享方案相比,它具有分发给参与者的秘密分片和要保护的秘密无知识上的联系,可以反映出参与者信任度的差异以及运算量低、使用灵活等优点.

多个参与者之间的串通与共谋一直是门限方案中的难点,基于量化权限的门限访问控制方案也不例外.问题的难点在于,当一组参与者提出获取秘密请求时,无法判断究竟是客观需要,还是他们之间串通好来获取秘密的.

References:

[1] Shamir A. How to share a secret. Communications of the ACM, 1979,22(11):612~613.

[2] Blundo C, de Santis A, Stinson DR, Vaccaro U. Graph decompositions and secret sharing schemes. In: Advances in Cryptology-Eurocrypt'92. LNCS 685, Springer-Verlag, 1993. 1~24.

[3] Sun HM, Shieh SP. Secret sharing in graph-based prohibited structures. In: Proc. of IEEE INFOCOM'97. IEEE, 1997. 718~724.

[4] Chang CC, Chan CW. Detecting dealer cheating in secret sharing systems. In: Proc. of the 24th Annual Int'l Computer Software and Applications Conf. (COMPSAC 2000). Taipei, 2000. 449~453.

[5] Fei RC, Wang LN. Cheat-Proof secret share schemes based on rsa and one-way function. Journal of Software, 2003,14(1):146~150 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/14/146.htm

[6] Cramer R, Ivan Damgård, Dziembowski S. On the complexity of verifiable secret sharing and multiparty computation. In: Proc. of the 32th Annual ACM Symp. on the Theory of Computing. 2000. ACM Press, 2000. 325~334.

[7] Stinson DR, Wei R. Unconditionally secure proactive secret sharing scheme with combinatorial structures. In: Proc. of the 6th Annual Workshop on Selected Areas in Cryptography (SAC'99). LNCS 1758, Springer Verlag, 1999. 200~214.

[8] Chopra K, Wallace WA. Trust in electronic environments. In: Proc. of the 36th Hawaii Int'l Conf. on System Sciences (HICSS 2003). IEEE, 2003. 331~340.

[9] Ferraiolo DF, Sandhu R, Serban G, Kuhn RD, Chandramouli R. Proposed NIST standard for role-based access control. ACM Trans. on Information and System Security, 2001,4(3):224~274.

[10] Osborn S, Sandhu R, Munawer Q. Configuring role-based access control to enforce mandatory and discretionary access control policies. ACM Trans. on Information and System Security, 2000,3(2):85~106.

[11] Ahn G, Sandhu R. Role-Based authorization constraints specification. ACM Trans. on Information and System Security, 2000,3(4): 207~226.

[12] Thompson M, Essiari A, Mudumbai S. Certificate-Based authorization policy in a PKI environment. ACM Trans. on Information and System Security, 2003,6(4):566~588.

[13] Thompson M, Johnston W, Mudumbai S, Hoo G, Jackson K, Essiari A. Certificate-Based access control for widely distributed resources. In: Proc. of the 8th USENIX Security Symp. Washington, 1999. 215~228.

[14] Levi A, Çağlayan UM. The problem of trusted third party in authentication and digital signature protocols. In: Proc. of the 12th Int'l Symp. on Computer and Information Sciences. 1997. http://people.sabanciuniv.edu/levi/

[15] Ingemarsson I, Simmons GJ. A protocol to set up shared secret schemes without the assistance of a mutually trusted party. In: Advances in Cryptology—EUROCRYPT'90. LNCS 473, 1991. 266~282.

附中文参考文献:

[5] 费如纯,王丽娜.基于 RSA 和单向函数防欺诈的秘密共享体制.软件学报,2003,14(1):146~150. http://www.jos.org.cn/1000-9825/14/146.htm