基于 Diffie-Hellman 算法的分层密钥分配方案

阎军智[1],李凤华[2,1],马建峰[1]
(1.西安电子科技大学计算机网络与信息安全教育部重点实验室,陕西西安 710071;
2.北京电子科技学院研究生处,北京 100070)

摘 要: 在基于内容的访问控制系统中,主体对客体只有允许访问和拒绝访问两种权限,且主体之间和客体之间都存在一种偏序关系,传统的访问控制策略需要对主客体单独进行管理,效率较低.本文利用其中的偏序关系设计一种分层密钥分配方案,使分配的密钥既能实现保密通信又能达到实施访问控制的目的,提高系统效率.该方案利用客体之间的偏序关系使所有客体形成一个有向无环图,以多方 Diffie-Hellman 算法为基础为图中每个节点分配密钥,使得每个节点都可以通过自己的密钥计算出其子节点的密钥,每个节点的密钥用于加密对应于该节点的资源,从而通过对密钥的分配实现对访问权限的管理.该方案分为系统建立、密钥更新、节点加入和节点删除等部分,其安全性基于 DDH 假设,支持成员以及分层拓扑结构的动态变化,可用于解决基于内容的分层访问控制问题.

关键词: 分层密钥分配; DH 算法; 分层访问控制; 群组密钥管理
中图分类号: TP39 **文献标识码:** A **文章编号:** 0372-2112 (2011) 01-0119-05

A Hierarchical Key Assignment Scheme Based on Diffie-Hellman Algorithm

YAN Jun-zhi[1], LI Feng-hua[2,1], MA Jian-feng[1]
(1. *Key Laboratory of Computer Networks and Information Security* (*Ministry of Education*), *Xidian University*, *Xi'an*, *Shaanxi* 710071, *China*;
2. *Graduate School*, *Beijing Electronic Science and Technology Institute*, *Beijing* 100070, *China*)

Abstract: In content-based access control systems, the subject is only allowed or denied to access the object. There are partial orders between different subjects and objects. The traditional access control policy manages these subjects and objects independently, and does not consider the partial orders which may improve the efficiency. By considering the partial orders, a hierarchical key assignment scheme is proposed in this paper, so as to make the assignment of keys to achieve secure communication and access control, improving the efficiency. The objects can be formed into a Directed Acyclic Graph (DAG) using the partial orders between these objects. Then, assign each vertex in the DAG an encryption key based on Diffie-Hellman algorithm, while each vertex may derive the encryption keys of its child vertices by the encryption key of itself. These assigned keys are used to encrypt the resources of the vertices. Thus, the access control of the resources can be achieved by the assignment of the encryption keys. The proposed scheme consists of the phases of system initialization and key updating, and supports user dynamics and topology changes. The security is based on DDH assumptions. It can be used for content-based hierarchical access control.

Key words: hierarchical key assignment; DH algorithm; hierarchical access control; group key management

1 引言

在数字广播、收费电视等众多基于内容的访问控制系统中,主体对客体只有允许访问和拒绝访问两种权限,并且主体之间和客体之间都存在一种偏序关系.若采用传统的访问控制策略,则需要对这些主客体单独进行管理,效率较低.以表 1 所示的收费电子信息广播系统为例,广播新闻分为体育新闻、财经新闻、股票信息以及天气预报,系统中的用户分为高级用户、体育用户、财经用户和基本用户,其中高级用户能够访问所有的广播信息,体育用户能够访问体育新闻和天气预报,财经用户能够访问财经新闻、股票信息和天气预报,基本用户则仅能够访问天气预报.这些信息需要在公开的信道进行传输,为了保证广播数据的安全,可以使用不同的密钥分别加密这些数据流,并将密钥分发给相应的用户,从而利用密码技术实现基于内容的访问控制.

分层密钥分配策略是解决这种问题的有效方法之一,其核心思想是为资源和用户分配访问权限,也称安

收稿日期:2010-01-07;修回日期:2010-01-27
基金项目:国家 863 高技术研究发展计划(No.2007AA01Z429, No.2007AA01Z472, No.2007AA01Z482, No.2009AA012420);国家自然科学基金(No.60633020, No.6087204);教育部重点项目(No.209156);北京市自然科学基金(No.4102056);北京电子科技学院信息安全重点实验室基金(No.YZDJ0807).

全等级(Security Class)[1],将具有相同安全等级的资源和用户分别归为同一资源类和用户组,然后为每个安全等级分配加密密钥,该密钥用于加密对应安全等级的资源,同时将这些密钥分配给相应安全等级用户组中的用户,因此只有拥有加密密钥的用户才能访问相应的资源.于是,对用户访问权限的管理就转化为对密钥的管理.这种利用密码技术实现访问控制的方法也称分层访问控制技术[2].

表 1 某电子信息广播系统中的访问权限[3]

	体育新闻	财经新闻	股票信息	天气预报
高级用户	√	√	√	√
体育用户	√			√
财经用户		√	√	√
基本用户				√

自从 Akl 和 Taylor 等人首次提出分层访问控制的有效解决方案以来[1],众多学者对分层访问控制的密钥分配展开了大量的研究.文献[4]利用中国剩余定理提出了一种应用于分布式环境的多级安全密钥管理方案,但该方案和文献[1]所述方案的扩展性较差.文献[5]针对实际应用需求提出两种访问模型,并利用这两种模型分别在信息系统和群组通信领域对现有的分层密钥分配策略进行了描述.Crampton 等人在文献[2]中提出了 5 种分层密钥分配算法模型,并改进了文献[1]中的方法,但改进后的方法依然缺乏扩展性.文献[6]研究了分层访问控制中用户、资源的分层结构,但是没有考虑在不同分层结构中的密钥分配问题.文献[7]对这些分层访问控制的密钥层次结构进行了研究,将分层访问控制中的密钥结构归纳为三类,并研究了这些结构之间的相互关系.文献[8]利用文献[1]中的算法提出了一种适用于密钥撤销的无状态分级群组密钥管理方案,利用本文提出的算法能够对该方案进行优化.

本文利用资源之间的偏序关系使所有资源形成一个有向无环图,基于多方 Diffie-Hellman 算法为图中节点实施密钥分配,使得图中每个节点都可以通过自己的密钥计算出其子节点的密钥.该方案分为系统建立、密钥更新、节点加入和节点删除等部分,其安全性基于 DDH 假设,支持成员以及分层拓扑结构的动态变化,与文献[1]中的方案相比具有良好的扩展性.

2 基本概念

称集合 X 为偏序集,如果该集合上存在一个二元关系"$\leqslant$",并且该二元关系具有自反性、反对称性和传递性.如果 $y<x$,并且不存在 $z\in X$ 使得 $y<z<x$ 成立,那么记 $y \lessdot x$,通常称二元关系"$\lessdot$"为覆盖关系,且称 y 为 x 的子节点,x 为 y 的父节点.

令 $x\in X$,定义 $\downarrow x=\{y\in X: y\leqslant x\}$,$\uparrow x=\{y\in X: y\geqslant x\}$.假设 Y 是集合 X 的非空子集,定义

$$\downarrow Y=\bigcup_{y\in Y}\downarrow y,\quad \uparrow Y=\bigcup_{y\in Y}\uparrow y.$$

偏序集中的元素按照偏序关系形成一个有向无环图(Directed Acyclic Graph,简记为 DAG).有关以上术语的详细描述参见文献[9]和文献[10].

Diffie-Hellman 算法(也称为指数密钥交换)是一种基本的密钥交换技术,两方的 Diffie-Hellman 密钥协商首次发表在 Diffie 和 Hellman 的论文[11]中.在两方 Diffie-Hellman 密钥协商的基础上,众多学者相继提出了基于多方的 Diffie-Hellman 群组密钥协商算法[12].设 p 是一个大素数,α 是 Z_p^* 的生成元,群组中共有 n 个成员,每个成员拥有的秘密信息(指数)分别为 $s_0, s_1, \cdots, s_{n-1}$,那么群组密钥 K 可通过以下方式计算,

$$K=\alpha^{s_0 s_1 \cdots s_{n-1}} \bmod p \tag{1}$$

$$K=\alpha^{s_0 s_1+s_1 s_2+\cdots+s_{n-1}s_0} \bmod p \tag{2}$$

$$K=\alpha^{s_{n-1}\alpha^{s_{n-2}\alpha^{\cdots s_2\alpha^{s_1 s_0}}}} \bmod p \tag{3}$$

其中等式(1)所述算法由 Ingemarsson 等人于 1982 年提出[13],是最早将两方 Diffie-Hellman 密钥协商扩展到群组密钥协商的方法.等式(2)所述算法也称为 BD 协议,由 Burmester 和 Desmedt 提出[14],该方案仅需两轮通信,且成员之间无需同步.等式(3)所述算法由 Steer 等人在文献[15]中首次提出.本文将以等式(1)和(3)所描述的算法为基础,设计一种分层密钥分配方案.

3 基于 Diffie-Hellman 算法的分层密钥分配方案

分别将能够访问资源 a 和 b 的用户集合记为 A 和 B,若 $A\subseteq B$,则 $a\geqslant b$.从而可以为系统中的所有资源建立分层结构,形成一个有向无环图.文献[7]详细论述了利用访问控制矩阵为用户和资源建立分层结构的方法,本文对此不再赘述,下文着重讨论针对 DAG 中节点的密钥分配方法.

记偏序集 X 中元素 x 的父节点集合为 $Upper(x)$,即 $Upper(x)=\{y\in X: x \lessdot y\}$.

与传统的 Diffie-Hellman 密钥协商方案类似,本节提出的分层密钥分配方案需要为每个节点分配一个加密密钥、一个公开的生成元和一组公开数值.每个节点的加密密钥可以根据其父节点的加密密钥和该节点的生成元通过等式(1)计算得出.

3.1 系统建立

分别记 DAG 中的节点为 $v_1, v_2, \cdots, v_N$,节点 v_i 对应的加密密钥和生成元分别为 K_i 和 g_i,记每个节点对应的一组公开数值为 pub_i.具体建立过程如下:

Step 1:选择并公布一个合适的素数 p;

Step 2:为每一个节点 v_i 分配 Z_p^* 上的一个公开的

生成元 $g_i(2\leqslant g_i\leqslant p-2)$;

Step 3:计算节点的加密密钥.为每个根节点随机选取一个加密密钥,对于其他节点 v_i,如果 v_i 有 m 个父节点,分别为 $v_{i_1},v_{i_2},\cdots,v_{i_m}$,即 $v_{i_j}\in Upper(v_i),1\leqslant j\leqslant m$,那么 K_i 可以由其父节点的加密密钥和 v_i 的生成元计算得出,$K_i=g_i^{K_{i_1}\cdots K_{i_m}} \bmod p$;

Step 4:计算节点的公开数值.如果节点 v_i 有多个父节点,那么需要为 v_i 定义一组公开数值,记这组公开数值为 pub_i,那么

$pub_i=\{(g_i)^{t_{i_l}} \bmod p;t_{i_l}=\dfrac{\prod_{j=1}^{m}K_{i_j}}{K_{i_l}}$,其中 K_{ij} 为分配给 v_i 父节点的密钥,$l\in\{1,\cdots,m\}\}$.

一般的,如果节点有 $m(m\geqslant 2)$个父节点,那么需要为其定义 m 个公开数值.

以图 1 描述的 DAG 为例,首先需要为图中每个节点分配一个生成元 g_i,由于节点 v_1 为根节点,所以需要为其随机选取一个加密密钥 K_1,分配给其他节点 v_2,v_3,v_4,v_5,v_6 的加密密钥分别计算如下:

$K_2=g_2^{K_1} \bmod p$,$K_3=g_3^{K_1} \bmod p$,

$K_4=g_4^{K_2} \bmod p$,$K_5=g_5^{K_2K_3} \bmod p$,

$K_6=g_6^{K_3} \bmod p$.

由于节点 v_5 有两个父节点,所以 v_5 对应的公开数值为　　$pub_5=\{g_5^{K_2} \bmod p,g_5^{K_3} \bmod p\}$.

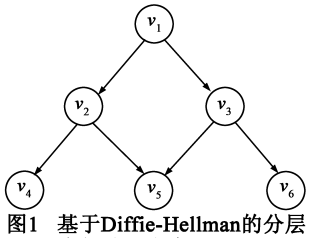

图1　基于Diffie-Hellman的分层密钥分配方案

利用自身的加密密钥和相应的公开数值,每个节点都能够计算出其子节点的加密密钥.以节点 v_2 为例,利用其自身的加密密钥 K_2 以及节点 v_4 的生成元 g_4,可以计算出节点 v_4 的加密密钥 $K_4=g_4^{K_2} \bmod p$;类似的,利用自身的加密密钥 K_2、节点 v_5 的生成元 g_5 和公开数值 $g_5^{K_3}$,可以计算出节点 v_5 的加密密钥 $K_5=(g_5^{K_3})^{K_2} \bmod p$.

3.2　密钥更新

群组中成员动态变化时,需要更新相应的密钥以确保群组通信中的前向安全性和后向安全性.如果某节点离开群组,那么该节点拥有的加密密钥都需要进行更新;如果某节点加入群组,那么该节点所能够访问的资源对应的加密密钥都需要进行更新.本方案中,对应于节点 v_i 的成员可以计算出集合 $\downarrow v_i$ 中所有节点的加密密钥,因此一旦对应于该节点的成员退出群组,或者有新成员加入该节点所对应的成员组,分配给集合 $\downarrow v_i$ 中所有节点的加密密钥都需要更新.

节点 v_i 的加密密钥的更新过程如下:

Step 1:为节点 v_i 选取一个新的生成元 g_i';

Step 2:利用 g_i' 重新计算节点 v_i 的加密密钥,以及集合 $\downarrow v_i$ 中所有节点的密钥;

Step 3:重新计算集合 $\downarrow v_i$ 中节点的公开数值,方法与系统建立时的 Step4 相同.

此时,利用 K_i 能够计算出的加密密钥都得以更新.对于 DAG 中大于 v_i 的节点,即集合 $\uparrow v_i$ 中的节点,可以利用节点 v_i 新的生成元计算出新加密密钥,进而也能计算出其所能访问节点的新加密密钥.

以图 1 中节点 v_2 为例,如果该节点有成员加入或者离开,那么集合 $\downarrow v_2$ 中所有节点的加密密钥都需要更新.首先为节点 v_2 选取一个新的生成元 g_2';然后计算 $\downarrow v_2$ 中的节点,即 v_2、v_4 和 v_5 的加密密钥,

$K_2'=g_2'^{K_1} \bmod p$,$K_4'=g_4^{K_2'} \bmod p$,$K_5'=g_5^{K_2'K_3} \bmod p$.

除此之外,节点 v_5 还需要更新相应的公开数值,

$$pub_5=\{g_5^{K_2'} \bmod p,g_5^{K_3} \bmod p\}.$$

集合 $\uparrow v_2$ 中的节点 v_1 能够利用 v_2 的新生成元计算出新加密密钥 $K_2'=g_2'^{K_1} \bmod p$,并且能够计算出集合 $\downarrow v_2$ 中所有节点新的加密密钥,即节点 v_4 和 v_5 的新加密密钥 K_4' 和 K_5'.

3.3　节点加入

以上讨论了网络中成员动态变化,即成员加入或离开相应群组时,相应节点加密密钥的更新过程.事实上,除此之外,网络拓扑结构也可能发生变化,因此设计的方案需要能够支持 DAG 中节点的变化,以下将分别讨论当 DAG 中节点加入和删除时的密钥更新过程.

记新加入的节点为 v_i,根据该节点在 DAG 中有没有父节点,密钥分配和更新过程分为以下两种情况:

(1)若 v_i 无父节点

Step 1:为 v_i 选取一个生成元 g_i 和一个加密密钥 K_i;

Step 2:重新计算集合 $\downarrow v_i$ 中所有节点的加密密钥,每个节点都需要根据父节点更新之后的加密密钥计算自己的加密密钥.如果 $v_j\prec v_i$,那么节点 v_j 的新加密密钥为 $K'_j=g_i^{K_{j1}\cdots K_{jm}K_i}=K_j^{K_i} \bmod p$,其中 $K_{j1},\cdots,K_{jm}$ 分别为节点 v_i 加入之前 v_j 的父节点 $v_{j1},\cdots,v_{jm}$ 对应的加密密钥.

Step 3:　重新计算集合 $\downarrow v_i$ 中节点的公开数值,方法与系统建立时的 Step 4 相同.

以图 2(a)中节点 v_7 为例,若该节点加入 DAG,首先为 v_7 选取一个生成元 g_7 和加密密钥 K_7;然后更新集合 $\downarrow v_7$ 中其他节点 v_3、v_5 和 v_6 的加密密钥,

$K'_3=g_3^{K_1K_7} \bmod p$,$K'_5=g_5^{K_2K_3} \bmod p$,$K'_6=g_6^{K'_3} \bmod p$.

最后更新 $\downarrow v_7$ 中节点的公开数值,由于 $\downarrow v_7$ 中节点 v_3 和 v_5 有两个父节点,故

$$pub_3 = \{g_3^{K_1} \bmod p, g_3^{K_7} \bmod p\},$$
$$pub_5 = \{g_5^{K_2} \bmod p, g_5^{K'_3} \bmod p\},$$

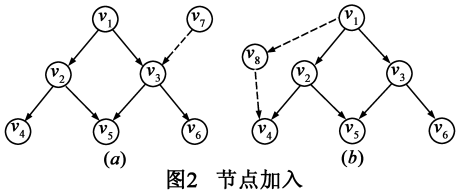

图2 节点加入

(2)若 v_i 有父节点

Step 1: 为 v_i 选取一个生成元 g_i;

Step 2: 计算节点 v_i 的加密密钥,重新计算集合 $\downarrow v_i$ 中其他节点的加密密钥,每个节点都需要根据父节点更新之后的加密密钥计算自己的加密密钥.

Step 3: 重新计算集合 $\downarrow v_i$ 中节点的公开数值,方法与系统建立时的 Step 4 相同.

以图 2(*b*)中节点 v_8 为例,该节 v_8 点加入 DAG 时,集合 $\downarrow v_8$ 中的节点 v_8 和 v_4 需要更新自己的加密密钥和相应的公开数值.首先为 v_8 选取一个生成元 g_8;然后计算节点 v_8 的加密密钥,$K_8 = g_8^{K_1} \bmod p$;更新节点 v_4 的加密密钥,$K'_4 = g_4^{K_8 K_2} \bmod p$;最后更新 $\downarrow v_8$ 中节点的公开数值,由于节点 v_4 有两个父节点,故 $pub_4 = \{g_4^{K_8} \bmod p, g_4^{K_2} \bmod p\}$.

3.4 节点删除

设删除的节点为 v_i,则需要根据 v_i 在 DAG 中是否为根节点进行讨论.

(1)若 v_i 为根节点

设 v_j 为 v_i 的子节点,删除 v_i 以后,需要删除从 v_i 到 v_j 的路径.之后,如果 v_j 为根节点,那么 v_j 的加密密钥无需更新;否则,需要重新计算 v_j 的加密密钥,以及集合 $\downarrow v_j$ 中节点的加密密钥和公开数值.

以图 3(*a*)为例,节点 v_1 有两个子节点,v_2 和 v_3,删除 v_1 后,需要删除 v_1 到其子节点的路径.在新的 DAG 中,v_2 成为根节点,故 v_2 的加密密钥无需更新;节点 v_3 不是根节点,所以需要重新计算 v_3 的加密密钥,以及集合 $\downarrow v_3$ 中其他节点的加密密钥和公开数值:

$$K'_3 = g_3^{K_7} \bmod p, K'_5 = g_5^{K_2 K'_3} \bmod p, K'_6 = g_6^{K_3} p,$$
$$pub_5 = \{g_5^{k_2} \bmod p, g_5^{K'_3} \bmod p\}$$

(2)若 v_i 不是根节点

如果 v_i 不是根节点,在删除 v_i 以及所有与 v_i 相连的路径后,需要添加从 v_i 每个父节点到 v_i 所有子节点的路径.如果 v_i 没有子节点,那么删除该节点后无需更新密钥;否则,删除该节点后需要更新集合 $\downarrow v_i$ 中节点的加密密钥和公开数值.

以图 3(*b*)为例,节点 v_2 有两个子节点,v_4 和 v_5,删除节点 v_2 以后,需要删除从其父节点 v_1 到 v_2 的路径,并且删除从 v_2 到其子节点 v_4 和 v_5 的路径,同时添加从 v_2 的父节点到 v_2 的子节点的路径,即,从节点 v_1 到节点 v_4 和 v_5 的路径.因此,需要更新集合 $\downarrow v_2$ 中除 v_2 以外其他节点的加密密钥和公开数值:

$$K'_4 = g_4^{K_1} \bmod p, K'_5 = g_5^{K_1 K_3} \bmod p,$$
$$pub_5 = \{g_5^{K_1} \bmod p, g_5^{K_3} \bmod p\}$$

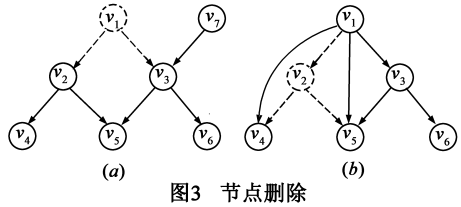

图3 节点删除

4 性能分析

4.1 安全性分析

在本文提出的方案中,系统首先需要选择并公布一个合适的素数 p,同时为每个节点 v_i 选取一个公开的生成元,从而每个节点的加密密钥都是通过 Diffie-Hellman 算法计算得出.若 v_i 在 DAG 中有多个父节点,节点 v_i 的加密密钥是利用其父节点的加密密钥,通过等式(1)所述算法计算得出,其每个父节点则利用自己的加密密钥和 v_i 的公开数值,通过等式(1)所述算法计算 K_i.Steiner 等人讨论了该算法的安全性,该算法的安全性基于 DDH 假设,即,如果两方 Diffie-Hellman 算法中密钥与随机数不可区分,那么多方 Diffie-Hellman 算法中的密钥也与随机数不可区分[12].如果 $v_i \geqslant v_j$,且存在 v_k 使得 $v_i > v_k > v_j$,那么利用节点 v_i 的加密密钥 K_i 计算节点 v_j 加密密钥 K_j 的算法即等式(3)所描述的算法.该算法的安全性同样基于 DDH 假设[16].

4.2 计算开销

在实际应用中,分层密钥分配方案在不同密钥结构中的性能存在较大差异.文献[7]讨论了分层密钥分配算法在不同密钥结构中的性能,有关在具体密钥结构中的性能可以参考该文献中的结论.本节将针对该方案自身的计算开销进行分析.

设 DAG 中节点总数为 N,A 为 DAG 的子集.定义集合 $PUB_A = \bigcup_{v_i \in A} pub_i$.在系统建立阶段,管理节点需要计算每个节点的加密密钥 K_i 以及公开数值 pub_i,由于加密密钥和公开数值的计算过程都是模指数运算,且根节点的加密密钥由管理节点随机选取,因此至多需要 $N + |PUB|$ 次模指数运算.

如果有成员加入或者离开对应于节点 v_i 的群组,那么对应于集合 $\downarrow v_i$ 中节点的加密密钥和公开数值都

需要进行更新.其中需要更新的加密密钥个数为 $|\downarrow v_i|$,需要更新的公开数值的上界为 $|PUB_{\downarrow v_i}|$,即,至多需要 $|\downarrow v_i|+|PUB_{\downarrow v_i}|$ 次模指数运算.

当节点 v_i 加入 DAG 以后,如果 v_i 没有父节点,那么需要为其随机选取一个加密密钥,并更新集合 $\downarrow v_i$ 中除 v_i 以外节点的加密密钥和公开数值;如果 v_i 有父节点,那么需要更新集合 $\downarrow v_i$ 中所有节点的加密密钥和公开数值.如果在 DAG 中删除节点 v_i,那么需要重新计算集合 $\downarrow v_i$ 中所有节点的加密密钥和公开数值.因此,当 DAG 拓扑结构变化时,需要更新的加密密钥个数为 $|\downarrow v_i|$,需要更新的公开数值的上界为 $|PUB_{\downarrow v_i}|$,即,至多需要 $|\downarrow v_i|+|PUB_{\downarrow v_i}|$ 次模指数运算.

文献[1]提出的算法在系统建立阶段需要 N 次模指数运算,优于本文提出的算法,但用户进行密钥计算时的计算开销与本文方案相同,均为一次模指数运算.文献[3]中的方案利用中国剩余定理,用户计算密钥时需要进行一次模乘运算,优于本文提出的方案,但该方案不适用于多个父节点并存的结构.除此之外,上述两种算法均不具有扩展性,如需添加或者删除节点,所有节点的密钥都需重新分配.

5 结束语

利用 Diffie-Hellman 算法设计了一种分层密钥分配方案.该方案分为系统建立、成员加入和退出、节点加入和删除三个部分,能够支持成员以及分层拓扑结构的动态变化,具有实现简单、易于扩展等特点,可用于解决基于内容的资源的权限分配问题.

参考文献:

[1] S G Akl, P D Taylor. Cryptographic solution to a problem of access control in a hierarchy[J]. ACM Transaction on Computer System, 1983, 1(3): 239 – 248.

[2] J Crampton, et al. On key assignment for hierarchical access control[A]. In Proceedings of the 19th IEEE workshop on Computer Security Foundations (CSFW'06)[C]. Washington, DC, USA: IEEE Computer Society, 2006. 98 – 111.

[3] Q Zhang, Y Wang. A centralized key management scheme for hierarchical access control[A]. In Proceedings of IEEE Global Telecommunications Conference (GLOBECOM'04) Vol. 4 [C]. Dallas TX, 2004. 2067 – 2071.

[4] 卿斯汉,蒙杨,刘克龙.分布式应用中的多级安全密钥管理[J].电子学报,2001,29(2):269 – 271.
Qing Si-han, Meng Yang, Liu Ke-long. Key management for multilevel security in distributed applications[J]. Acta Electronica Sincia, 2001, 29(2): 269 – 271(in Chinese).

[5] H R Hassen, A Bouabdallah, H Bettahar, et al. Key management for content access control in a hierarchy[J]. Computer Networks, 2007, 51(11): 3197 – 3219.

[6] J C Birget, X Zou, G Noubir. Hierarchy-based access control in distributed environments[A]. In IEEE International Conference on Communications (ICC '01) Vol. 1[C]. 2001. 229 – 233.

[7] J Z Yan, J F Ma, H Y Liu. Key Hierarchies for Hierarchical Access Control in Secure Group Communications[J]. Computer Networks, 2009, 53(3): 353 – 364.

[8] F H Li, J Z Yan, J F Ma, et al. Leveled Group Key Management with Efficient Revocations for Wireless Sensor Networks [J]. Chinese Journal of Electronics, 2009, 18(3): 494 – 499.

[9] B Davey, H Priestley. Introduction to Lattices and Order[M]. Cambridge University Press, 1990.

[10] F Harary. Graph Theory[M]. Reading, MA: Addison-Wesley, 1994.

[11] W Diffie. New directions in cryptography[J]. IEEE Transactions on Information Theory, 1976, 22(6): 644 – 654.

[12] M Steiner, G Tsudik, M Waidner. Diffie-hellman key distribution extended to group communication[A]. Proceedings of the 3rd ACM Conference on Computer and Communications Security[C]. New York, NY, USA: ACM, 1996. 31 – 37.

[13] I Ingemarsson, D Tang, C Wong. A conference key distribution system[J]. IEEE Transactions on Information Theory, 1982, 28(5): 714 – 720.

[14] M Burmester, Y Desmedt. A secure and efficient conference key distribution system[A]. In Advances in Cryptology-EUROCRYPT'94[C], LNCS 950, 1995. 275 – 286.

[15] D G Steer, L Strawczynski, W Diffie, et al. A secure audio teleconference system[A]. In Proceedings of the 8th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO'88)[C]. London, UK, 1990. 520 – 528.

[16] Y Kim, A Perrig, G Tsudik. Tree-based group key agreement [J]. ACM Transactions on Information and System Security, 2004, 7(1): 60 – 96.

作者简介:

阎军智　男,1981 年 1 月出生于河南省郑州市,西安电子科技大学博士生,主要研究方向为密码学与网络安全.
E-mail: jzyan@mail.xidian.edu.cn

李凤华　男,1966 年 3 月出生于湖北省浠水县,北京电子科技学院研究生处处长、博士、教授、博士生导师,主要研究方向为网络安全与可信计算.
E-mail: lfh@besti.edu.cn

马建峰　男,1963 年 10 月出生于陕西省西安市,西安电子科技大学计算机学院院长、博士、教授、博士生导师,主要研究方向为密码学与网络安全.
E-mail: jfma@mail.xidian.edu.cn