安全组播的 Huffman 层次密钥管理*

屈 劲[1+], 葛建华[1], 蒋 铭[2]

[1](西安电子科技大学 国家 ISN 重点实验室,陕西 西安 710071)

[2](上海交通大学 电子信息与电气工程学院,上海 200030)

Key Management for Secure Multicast Using Huffman Hierarchy

QU Jin[1+], GE Jian-Hua[1], JIANG Ming[2]

[1](National Key Laboratory of ISN, Xidian University, Xi'an 710071, China)

[2](School of Electronics & Information and Electric Power Engineering, Shanghai Jiaotong University, Shanghai 200030, China)

+ Corresponding author: Phn: 86-29-8202516, Fax 86-29-8202515, E-mail: mikequ@263.net

http://www.xidian.edu.cn

Received 2001-12-17; Accepted 2002-04-10

Qu J, Ge JH, Jiang M. Key management for secure multicast using Huffman hierarchy. *Journal of Software*, 2003,14(1):151~156.

Abstract: Key management system is an important part of secure multicast, while the number of keys held by each user and the cost of re-keying are crucial factors which closely related to the performance of key management system. The key management system with less number of keys held by each user and less cost of re-keying is efficient. In this paper, the key management problem is investigated based on the user probability model by using the source coding theory. And it is proved that in the key management system using Huffman hierarchy, the average cost of re-keying and the number of keys held by each user are minimal. Meanwhile, the lower bounds of the average cost of re-keying and the number of keys held by each user in theory are provided.

Key words: multicast; key management; Huffman tree

摘 要: 密钥管理系统是安全组播的重要组成部分,而用户密钥存储量和密钥更新代价又是衡量密钥管理系统性能的关键因素.一个高性能的密钥管理系统应具有较小的用户密钥存储量和密钥更新代价.利用信源编码理论深入研究了基于用户概率模型的密钥管理问题,证明了采用密钥 Huffman 层次结构的密钥管理系统的平均密钥更新代价和用户密钥存储量最小,同时还推导得出了密钥层次结构的理论平均密钥更新代价和用户密钥存储量的下限.

关键词: 组播;密钥管理;霍夫曼树

中图法分类号: TP309 **文献标识码**: A

随着网络的迅速发展,出现了许多基于组播的新业务,如视频广播、网络会议、新闻及股票信息分发等.它

* Supported by the Assisting Project of Ministry of Education of China for Backbone Teachers of University and College under Grant No.2992 (国家教育部高等院校骨干教师资助计划)

第一作者简介: 屈劲(1971－),男,陕西宁强人,博士生,主要研究领域为信息安全及条件接收.

们的共同特点是将信息由一点传输到多点.由于组播采用点到多点的通信方式,它对安全性的要求与单播不同.

组播安全应该满足:(1) 静态安全,即所有组用户都可以接收到组播信息,非组成员不能接收组播信息;(2) 动态安全,当用户变更时,新加入的用户可以接收组播信息,而被撤销的原组用户不能接收组播信息.因此,安全组播需要维护一个组播组,任何需要加入或离开组的用户都必须提出申请并得到许可才可加入或离开组播组.

组播的安全性可以通过密码机制来实现:组播组设置一个密钥服务器,密钥服务器和组用户共享会话密钥,加密密钥由会话密钥加密,组播信息由加密密钥加密,任何组用户都可以利用共享的会话密钥得到加密密钥,并解密组播信息,任何非组用户由于不知道会话密钥的内容,无法获得加密密钥,因此不能解密组播信息;当用户变更时,密钥服务器更新会话密钥,为新用户分发新会话密钥,使被撤销用户的旧会话密钥失效.由此可知密钥管理是安全组播的重要环节.

组播环境下密钥管理问题复杂,迄今为止还没有一个很好的密钥管理方法.例如,一个用户数为 n 的组播组,如果分别为每位组用户分发会话密钥,则组用户需要存储用户私钥和会话密钥,密钥存储量为 2;而当一个用户离开组时,密钥服务器需要为每位用户更新会话密钥,密钥更新代价为 $O(n)$,密钥更新代价会随组用户数线性变化.因此,降低密钥更新代价是安全组播密钥管理的一个关键问题.解决这个问题主要有两种途径:文献[1~3]采用逻辑密钥层次结构减轻了用户变更时的密钥更新代价,该方法以增加用户密钥存储量为代价将密钥更新代价降低到 $O(\log_d n)$,其中 d 为层次结构的维数,但当用户变更频繁时更新代价仍很大;文献[4,5]在前者基础上加入周期性更新会话密钥的思想,使密钥更新代价和用户变更频率无关,进一步降低了密钥更新代价,由于放宽了安全性条件,用户加入组或离开组时存在时延.文献[6]利用信源编码思想考察了采用密钥层次结构的用户密钥存储量的下限问题,但没有考虑密钥更新问题.

由于密钥更新对密钥管理系统性能的影响更为重要,本文着重考虑密钥更新问题.本文在密钥层次结构基础上将信源编码非等长码的 Huffman 编码方法引入密钥层次结构的划分,根据用户离开组的概率特性构造密钥 Huffman 树层次结构,证明该结构具有最低的平均用户密钥更新代价和密钥存储量,并推导得出了层次结构的平均密钥更新量和存储量下限.

1 密钥 Huffman 层次结构及密钥更新策略

1.1 密钥树

文献[1~3]采用逻辑密钥层次结构解决密钥管理问题,密钥由密钥服务器产生,并被组织成逻辑层次结构,如图 1 中的左图所示.这种层次结构类似于方向树,因此又被称为密钥树.密钥树中的每一节点代表一个密钥,根节点表示会话密钥 SK(session key),叶节点代表与用户一一对应的用户私钥 PK(private key),中间节点代表组密钥 GK(group key).为了便于描述密钥树结构,本文作如下定义:定义树的方向为:树叶到树根的方向;对任意节点而言,连接该节点指向根节点的路径称为接出路径,连接叶节点指向该节点的路径称为接入路径;任意用户 u_i 的私钥 PK_i 到树根的路径长度称为 PK_i 的高 h_i;树的高度(层数) $h=\max_i\{h_i\}$;树的维数 d 定义为树中节点的最大接入路径数.由上述结构可知,任意组用户需要存储该用户所在路径所有节点对应的密钥.

为了保证组播安全,任意用户加入或离开组时必须更新密钥树,使新用户可以解密组播信息,离开用户不能解密组播信息.如图 1 所示,当用户 u_9 离开组时,密钥服务器发送下述信息更新密钥树:($\{GK_{7\text{-}8}\}_{PK_7}$,$\{GK_{7\text{-}8}\}_{PK_8}$,$\{SK_{1\text{-}8}\}_{GK_{1\text{-}3}}$,$\{SK_{1\text{-}8}\}_{GK_{4\text{-}6}}$,$\{SK_{1\text{-}8}\}_{GK_{7\text{-}8}}$),其中 $\{m\}_k$ 表示用密钥 k 加密信息 m.

1.2 密钥概率分层结构

密钥概率分层结构的主要依据是用户在密钥更新周期里,以一定概率离开组播组,文献[6]中也使用了相同的概念.

设组用户数为 n,用户集 $U=\{1,2,3,...,n\}$ 表示组播用户全体,密钥更新周期内仅有用户 u_i 离开组的概率为 p_i,对于 $1\le i\le n$ 满足:$\sum_{i=1}^{i=n}p_i=1$,用户平均密钥存储量的理论下限为用户离开组的事件熵:

$$H=-\sum p_i \log_d p_i \ . \tag{1}$$

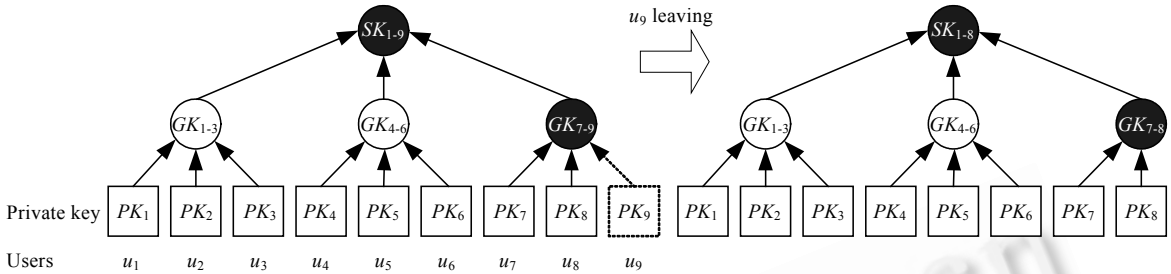

Fig.1 Key tree and keys updating for user leaving

图 1 密钥树及用户离开时的密钥更新

在密钥树结构中,用户 u_i 的密钥存储量为 h_i+1,而 h_i 又等价于信源编码中非等长码的码长,因此,用户的最小平均密钥存储量等价于非等长码的最佳平均码长问题.利用信源编码的非等长编码理论[7]可以证明,当用户量有限时,密钥 Huffman 层次结构必定具有最小的平均用户密钥存储量.本文还将证明其具有最小的平均密钥更新量.

1.3 密钥Huffman层次结构

本文利用 Huffman 编码方式构造密钥 Huffman 层次结构.在构造 d 维 Huffman 满树时,密钥树从根开始分裂可形成 d 个叶节点,叶节点每次再分裂可增加 d–1 个叶节点,因此如果分裂 s 次,Huffman 满树的总叶片数 m 应满足下式:

$$m=(s-1)(d-1)+d \ . \tag{2}$$

构造 Huffman 树,给定 h 或 d,首先计算总叶片数 m,验证用户数 n 是否满足式(2),若满足,则 m=n;若不满足,则($\lceil x \rceil$表示大于或等于 x 的最小整数):

$$m=\left\lceil \frac{n-d}{d-1} \right\rceil (d-1)+d \ . \tag{3}$$

为了保证满树需增加 m–n 个概率为 0 的空用户位置,可以证明 $m-n<d-1$.

根据用户离开概率对用户进行分组.先将 m 个用户按离开概率大小排队,概率大的排在上面,概率小的排在下面;将离开概率最小的 d 个用户分成一组形成一个组节点,并将这 d 个用户的离开概率相加作为该组节点的概率,再将该组节点和其他用户重新排队,再取概率最小的 d 个节点或用户组成一个组节点,如此下去直到树根构成一个Huffman满树,树根的概率为1,最后给节点安排密钥,这样就构成了密钥 Huffman 层次结构.为了保证满树,加入了若干离开概率为 0 的空用户位置,这些位置虽分配密钥,但实际是冗余用户,这样分配密钥仍具有最小平均用户密钥量.

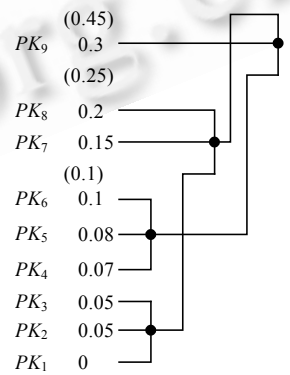

Fig.2 The construction of Huffman tree

图 2 Huffman 树的构造

下面举例说明,假设存在一个 8 个用户的组播组,每位用户的离开概率分别为{0.05,0.05,0.07,0.08,0.1,0.15,0.2,0.3},设计 d=3 的密钥 Huffman 层次结构:根据式(3),m=9,需要增加一个概率为 0 的空用户位置;概率最小的 3 个用户合成一个组节点,而后重新排序再取概率最小的 3 个节点或用户组成一个组节点,以此类推构成如图 2 所示的 Huffman 树,然后安排密钥构成如图 3 所示的分层结构.

1.4 密钥更新策略

当有用户加入或离开组时,为保证组播安全必须更新密钥.下面分别讨论用户加入方法和密钥更新策略.

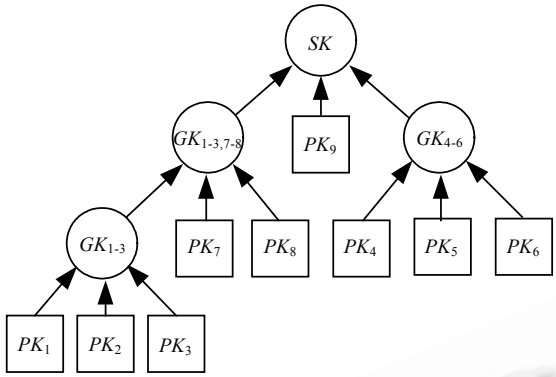

Fig.3 Grouping keys with Huffman hierarchy
图 3 组织密钥为 Huffman 层次

1.4.1 用户加入方法

用户变动存在两种情况:加入用户数 N_J 少于或等于删除用户数 N_L 及空节点数 N_n(包括构造满树时加入的空用户位置和上个密钥更新周期用户离开后没有填充的位置)之和;加入用户数 N_J 大于删除用户数 N_L 及空节点数 N_n 之和.下面分别讨论上述情况的用户加入方法.

$N_J \le N_L + N_n$,根据用户的离开概率,将用户分别安排到概率接近的空节点位置和用户离开后留下的删除位置.

$N_J > N_L + N_n$,根据加入用户的离开概率选择加入点,先将用户加入到相应组中空节点和删除节点,再将未加入用户根据其离开概率尽量安排在新用户加入节点,对该节点进行分裂加入用户,如图 4 所示.

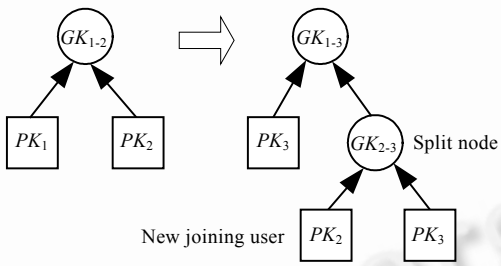

Fig.4 Example of node splitting for $N_J > N_L + N_n$
图 4 节点分裂的例子($N_J > N_L + N_n$)

1.4.2 密钥更新策略

根据用户加入方法,加入用户被安排在 4 种可能的位置,不同位置具有不同的密钥更新策略.(1) 空节点,只需给新加入的用户分配该节点路径上的所有组密钥,组密钥无须更新,密钥分发量小.(2) 删除节点,需要更新所有该节点路径的组密钥,并为新加入的用户分配新密钥,密钥更新量大,主要由删除用户引起的密钥更新量决定.(3) 分裂节点位于空节点,密钥更新量小,与空节点情况相似.(4) 分裂节点位于删除节点,密钥更新情况与删除节点情况类似,密钥更新量主要由删除用户引起的密钥更新量决定.

2 密钥 Huffman 层次结构的密钥更新性能分析

在保证组播安全性的前提下,衡量密钥管理系统性能的主要指标是密钥更新量和用户存储密钥量.由第 1.4.2 节可知,密钥更新量主要取决于删除用户引起的密钥更新量,而被删除用户可能同组,也可能不同组,同组时密钥更新量小些,不同组时更新量大些,但都取决于一个用户离开时的用户密钥更新量,所以本文主要分析用户密钥存储量和一个用户离开时的密钥更新量.

2.1 非概率密钥平衡树分层结构

在文献[3]的平衡树模型中,任意用户 u_i 的密钥存储量为

$$s_i = h + 1 = \lceil \log_d n \rceil + 1 . \tag{4}$$

任意用户 u_i 离开时的密钥更新量为

$$c_i = dh - 1 = d\lceil \log_d n \rceil - 1 . \tag{5}$$

对于概率分层结构而言,非概率密钥分层结构实际上就是概率模型的等概情况,由信息论[7]可知,等概时熵最大,即密钥存储量和更新量最大.

2.2 密钥Huffman层次结构

在 Huffman 满树时删除用户 u_i 的密钥更新量和该用户的密钥量与该用户 PK_i 在密钥树中的高度 h_i 有关,c_{h_i} 表示删除高度 h_i 的用户 u_i 时的密钥更新量:

$$c_{h_i} = h_i d - 1 . \tag{6}$$

u_i 密钥存储量为

$$s_{h_i} = h_i + 1 . \tag{7}$$

一个用户离开时的平均密钥更新量为

$$\bar{c} = \sum_{i=1}^{n} p_i c_{h_i} = \sum_{i=1}^{n} p_i (h_i d - 1) = d\sum_{i=1}^{n} p_i h_i - 1 , \tag{8}$$

平均的用户密钥存储量为

$$\bar{s} = \sum_{i=1}^{n} p_i s_{h_i} = \sum_{i=1}^{n} p_i h_i + 1 . \tag{9}$$

由第 1.2 节的 Huffman 树构造规则可知,h_i 等效于非等长码的码长,因此,根据非等长码编码理论[7],密钥 Huffman 层次结构具有最小的平均密钥存储量,根据式(8)可知,它还具有最小平均密钥更新量.同时,根据 Shanno 不等长码编码理论 $\sum_{i=1}^{n} p_i h_i \geq -\sum_{i=1}^{n} p_i \log_d p_i$,采用文献[7]中的方法可以证明,当用户数 $n \to \infty$ 时,上式等号成立,说明用户平均密钥更新量和用户平均密钥存储量分别存在理论下限:

$$\bar{c} \geq -d\sum_{i=1}^{n} p_i \log_d p_i - 1 , \tag{10}$$

$$\bar{s} \geq -\sum_{i=1}^{n} p_i \log_d p_i + 1 . \tag{11}$$

当用户数 $n \to \infty$ 时,上式等号成立.由于密钥非概率层次结构等价于等概情况,且此时熵最大,因此,上述两个下限就是密钥层次结构平均密钥更新代价和用户密钥存储量下限.这两个下限可以作为衡量密钥管理系统性能的重要参数.

2.3 密钥更新策略的动态性能分析

由上述分析可知,采用密钥 Huffman 层次管理的密钥管理系统可以得到最优的密钥存储量和密钥更新量.然而,对于存在用户频繁变更的动态组播系统,维护密钥 Huffman 树是困难的.如果每次用户变更后都需要构造新的 Huffman 树,以保证最优的密钥存储量,所有用户都需要更新密钥,密钥更新量太大,这显然不现实.同时,对于网络环境的密钥分发系统而言,密钥更新量对网络负荷影响很大.因此,本文采用第 1.4 节的密钥更新策略,新加入用户的 PK_i 被安排在 $h_i \approx -\log_d p_i$ 的位置,并尽量减少原有用户密钥更新量.这样的密钥更新量近似最优,但新树不满足 Huffman 层次规律,平均密钥存储量并非最优.

3 结束语

密钥管理系统是安全组播的重要组成部分.本文根据用户离开组的概率模型,详细讨论了衡量安全组播密钥管理系统性能的两个主要指标:平均密钥更新代价和平均用户密钥存储量,证明了基于用户离开概率模型的

密钥 Huffman 层次结构在用户数有限时具有最小的密钥更新代价和平均用户密钥存储量,还推导出了密钥层次结构的密钥更新量和平均用户存储量的理论下限,这两个下限可以作为衡量密钥管理系统性能的重要参数.

References:

[1] Fenner W. Internet group management protocol, Version 2. IETF RFC 2236, 1997.

[2] Wallner D, Harder E, Agee R. Key management for multicast: issues and architectures. IETF Internet Draft, 1998.

[3] Wong CK, Gouda MG, Lam SS. Secure group communications using key graphs. In: Neufeld G, ed. Proceedings of the ACM SIGCOMM'98. New York: ACM Press, 1998. 68~79.

[4] Setia S, Koussih S, Jajodia S, Harder E. Kronos: a scalable group re-keying approach for secure multicast. In: Reiter M, Needham R, eds. Proceedings of the IEEE Symposium on Security and Privacy. Oakland, CA: IEEE Computer Society Press, 2000. 215~228.

[5] Li XS, Yang YR, Gouda MG, Lam SS. Batch rekeying for secure group communications. In: Proceedings of the 10th International World Wide Web Conference (WWW10). 2001. http://www10.org/cdrom/papers/frame.html.

[6] Poovendran R. Key management for secure multicast communications [Ph.D. Thesis]. College Park: University of Maryland, 1999.

[7] Zhou JP, Ding XM. Source coding theory. Beijing: People's Post and Telecommunications Publishing House, 1996 (in Chinese).

附中文参考文献:

[7] 周炯磐,丁小明.信源编码理论.北京:人民邮电出版社,1996.

敬 告 作 者

《软件学报》创刊以来,蒙国内外学术界厚爱,收到许多高质量的稿件,其中不少在发表后读者反映良好,认为本刊保持了较高的学术水平.但也有一些稿件因不符合本刊的要求而未能通过审稿.为了帮助广大作者尽快地把他们的优秀研究成果发表在我刊上,特此列举一些审稿过程中经常遇到的问题,请作者投稿时尽量予以避免,以利大作的发表.

1. 读书偶有所得,即匆忙成文,未曾注意该领域或该研究课题国内外近年来的发展情况,不引用和不比较最近文献中的同类结果,有的甚至完全不列参考文献.

2. 做了一个软件系统,详尽描述该系统的各个方面,如像工作报告,但采用的基本上是成熟技术,未与国内外同类系统比较,没有指出该系统在技术上哪几点比别人先进,为什么先进.一般来说,技术上没有创新的软件系统是没有发表价值的.

3. 提出一个新的算法,认为该算法优越,但既未从数学上证明比现有的其他算法好(例如降低复杂性),也没有用实验数据来进行对比,难以令人信服.

4. 提出一个大型软件系统的总体设想,但很粗糙,而且还没有(哪怕是部分的)实现,很难证明该设想是现实的、可行的、先进的.

5. 介绍一个现有的软件开发方法,或一个现有软件产品的结构(非作者本人开发,往往是引进的,或公司产品),甚至某一软件的使用方法.本刊不登载高级科普文章,不支持在论文中引进广告色彩.

6. 提出对软件开发或软件产业的某种观点,泛泛而论,技术含量少.本刊目前暂不开办软件论坛,只发表学术文章,但也欢迎材料丰富,反映现代软件理论或技术发展,并含有作者精辟见解的某一领域的综述文章.

7. 介绍作者做的把软件技术应用于某个领域的工作,但其中软件技术含量太少,甚至微不足道,大部分内容是其他专业领域的技术细节,这类文章宜改投其他专业刊物.

8. 其主要内容已经在其他正式学术刊物上或在正式出版物中发表过的文章,一稿多投的文章,经退稿后未作本质修改换名重投的文章.

本刊热情欢迎国内外科技界对《软件学报》踊跃投稿.为了和大家一起办好本刊,特提出以上各点敬告作者.并且欢迎广大作者和读者对本刊的各个方面,尤其是对论文的质量多多提出批评建议.