第36卷 第21期 计 算 机 工 程 2010年11月
Vol.36 No.21 Computer Engineering November 2010
·安全技术· 文章编号：1000—3428(2010)21—0149—03 文献标识码：A 中图分类号：TP309

基于秘密共享的组播密钥更新算法

赵龙泉，苏锦海
(解放军信息工程大学电子技术学院，郑州 450004)

摘 要：提出一种基于秘密共享的组播密钥更新算法。采用二叉逻辑密钥树结构，根据组成员状态变化，利用秘密共享的思想构造广播消息，使组成员可以逐步计算组密钥，而非组成员不能计算组密钥，从而实现组密钥更新。分析表明，与采用逻辑密钥树的算法相比，该算法能降低密钥更新时的通信量和计算量，适用于大型的动态群组通信。

关键词：组密钥管理；秘密共享；密钥树；密钥更新

Group Re-keying Algorithm Based on Secret Sharing

ZHAO Long-quan, SU Jin-hai
(Institute of Electronic Technology, PLA Information Engineering University, Zhengzhou 450004, China)

【Abstract】 The ideas of group re-keying algorithms based on secret sharing using the LKH tree are proposed in this paper. The algorithms construct a message using secret sharing with the dynamic change of group members. The group members can reconstruct the new group key. It is proved that the new algorithms have obvious superiority than that of the previously proposed algorithms on communication and computation, and are suitable for large dynamic group.

【Key words】 group key management; secret sharing; key tree; re-keying

1 概述

组播是一种面向组接收者的高效通信方式，有效地节约了带宽，降低了服务器的负担，可广泛地应用于多媒体远程教育、分布式系统、网络视频会议等。安全组播的关键问题是如何实现有效的组密钥管理。在组播通信中，组成员关系是动态变化的，在成员加入/离开时要及时更新组密钥。尤其是在一个大型动态多播组中，组成员变动频繁时，如何解决密钥更新，是组密钥管理的核心问题。

文献[1-2]分别提出了采用逻辑密钥层次(LKH, Logical Key Hierarchy)的组密钥管理机制，通过增加辅助节点密钥，有效地降低组成员状态变化时密钥更新的通信次数，减少了通信量。但是增加了密钥服务器和组成员的密钥存储量。文献[3]提出的单向函数树(One-Way Function Tree, OFT)算法是LKH算法的一种改进，通过设置单向函数，减少密钥更新时的消息长度，将通信代价降低为LKH算法的一半。文献[4]中提出了一种基于多项式展开的PE-LKH方案，由于不使用加解密算法，降低了密钥更新时的计算量。

文献[5]提出有恢复能力的组密钥分发方案，文献[6]提出一种基于秘密共享的、具有无条件安全的多轮撤消算法，使用门限秘密共享的技术，降低了密钥更新的通信量。但是这2个方案是平级结构，扩展性受到一定的限制。文献[7]提出基于门限秘密共享的动态组播密钥协商方案，采用两级分层分组结构。

本文在LKH的基础上结合秘密共享的思想，提出一个基于秘密共享的组密钥更新算法。该算法基于二叉逻辑密钥树结构，利用秘密共享的思想，当组成员状态变化时，通过组控制器广播的公开消息，使组成员可以逐步计算出组密钥，而非组成员不能计算组密钥，从而实现组密钥更新。

2 方案描述

本文提出的组密钥更新算法基于二叉逻辑密钥树结构，使用秘密共享理论和单向函数的方法优化密钥更新的通信量和计算量。与以往算法比较，本算法具有更好的性能。

2.1 符号定义

符号定义如下：

p：为一个大素数；

Zp：表示模 p 的剩余类群，所有运算都 Zp 中；

r：表示密钥更新次数，

$SK(r)$：表示 r 时的组会话密钥，

$K_i(r)$：表示 r 时的为节点 i 的密钥，

$B:(a_h,b_h)$：表示组控制器广播的 h 层的信息，

i：表示节点位置的 Huffman 编码，

$h()$：是一个密码学意义上的单向函数，令 $y_i(r)=h(k_i,\mu_r)$；

$f()$：表示一个从 Huffman 编码域到 Zp 的一一映射，且 $f(x)_{x=0,\cdots,0}\neq 0$；

μ_r：表示 r 次会话时的新鲜因子。

2.2 二叉逻辑密钥树的初始化

组控制器根据潜在的成员数 N 构建一棵二叉平衡树，密钥树的高度为 $h=\lceil 1\mathrm{b}N\rceil$。树根为组管理者，树叶为组成员，树的中间节点为逻辑节点。组控制器利用 Huffman 编码方法对二叉逻辑密钥树的节点编码，作为节点的位置信息。每个成员则拥有从所在的叶节点到根的路径上的所有密钥。

在初始化时，即 $r=0$，组控制器 GC 负责初始化密钥树，

作者简介：赵龙泉(1982－)，男，硕士研究生，主研方向：密钥管理；苏锦海，教授、博士

收稿日期：2010-04-10 **E-mail：**zhaolongquan1982@sina.com

随机的选择更新因子 μ_0。然后，利用单向函数和秘密共享理论从下往上配置密钥。假设有一节点为 i，2 个儿子节点为 $2i, 2i+1$，组控制器根据 2 个儿子节点的密钥和位置信息，利用 Shamir 门限方案构造 1 阶多项式，将 0 代入，计算节点 i 的密钥：$K_i(0) = y_{2i}(0) - f(2i)\dfrac{(y_{2i+1}(0) - y_{2i}(0))}{f(2i+1) - f(2i)}$。依次类推，可以计算出初始化时的组会话密钥。

以 3 层的平衡二叉树为例说明逻辑密钥树初始化，如图 1 所示。组成员 u_{010} 拥有的密钥为 $\{K_{010}, K_{01}, K_0, SK\}$，其中，$SK$ 为组会话通信密钥，K_{010} 为组成员的用户私钥，K_0, K_{01} 为辅助节点密钥。以节点密钥 K_{01}, K_{010}, K_{011} 为例，父子节点密钥的关系为 $K_{01}(0) = y_{010}(0) - f(010)\dfrac{(y_{011}(0) - y_{010}(0))}{f(011) - f(010)}$。从初始化过程可以看出，每个用户的存储量为 $h = \lceil \mathrm{lb}N \rceil$，组控制器的存储量为 $2N-1$，与基本的密钥树结构相同。

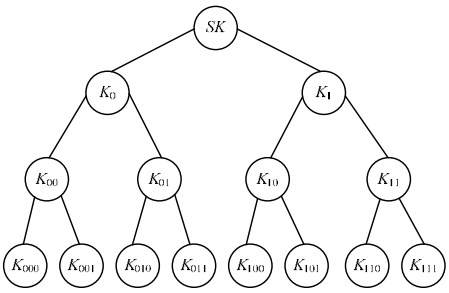

图 1　二叉平衡逻辑密钥树

2.3　密钥更新

为保证组密钥的安全性，在用户加入和退出时，必须更新组密钥。针对大型动态用户组，组密钥更新是关键问题。在密钥更新时，从下而上采用广播的方式，更新发生变化的节点密钥。下面以图 1 为例，说明密钥更新的过程。

2.3.1　用户加入

当有新成员申请加入组播组时，为了确保新成员不能访问以前的通信数据，必须对会话密钥进行更新，更新过程描述如下：

(1)新成员申请加入组播组。

假设节点 011 是 1 个空节点。用户 u_i 申请加入组播组，组控制器首先验证成员的身份，确认后将该成员插入成员节点 011，并产生密钥 K_{011} 作为该成员的私有密钥，秘密分发给该成员。为了实现向前访问控制，组控制器需要产生新组会话密钥 $SK(r+1)$，则需要更新的节点密钥为 $SK(r), K_0(r), K_{01}(r), K_{011}(r)$。

(2)组控制器建立广播信息。

1)组控制器根据两点 $(f(010), y_{010}(r+1))$，$(f(011), y_{011}(r+1))$，利用插值定理构造直线：

$$F_3(x,y): y - y_{010}(r+1) = \frac{y_{011}(r+1) - y_{010}(r+1)}{f(011) - f(010)}(x - f(010))$$

2)组控制器将 $x=0$ 代入直线方程 $F_3(x,y)$，计算父亲节点的密钥 $K_{01}(r+1) = F_3(0)$；

3)组控制器随机的选择一点 $(a_3, b_3) \in F_3(x,y)$ 且不等于以上 3 点，作为广播消息 B_3；

同理，确定 $SK(r+1), K_0(r+1)$，选择广播消息 $B_1: (a_2, b_2)$，$B_2: (a_1, b_1)$。

(3)组控制器广播消息，消息包括 $B = \{B_1, B_2, B_3, \mu_{r+1}\}$。

(4)组成员计算更新后的密钥。

当组成员收到消息后，根据所掌握的密钥计算会话密钥。成员 u_{010}, u_{011} 利用私钥和 B_3 重构 $F_3(x,y)$，将 $x=0$ 代入获得 $K_{01}(r+1)$，然后再依次计算 $K_0(r+1), SK(r+1)$；成员 u_{000}, u_{001} 利用 $K_{00}(r+1)$ 依次计算 $K_0(r+1), SK(r+1)$；成员 $u_{00}, u_{101}, u_{110}, u_{111}$ 利用 $K_1(r+1)$ 和 B_1 直接计算 $SK(r+1)$。

通过以上过程分析可知，在有单个成员加入时，组控制器只需要广播一条公开消息，组成员就可以计算更新后会话密钥，节省了密钥更新时的通信量。同时，不同位置的用户的计算量是不同，在密钥更新过程中，不需要使用加密算法，减少了成员的计算。

2.3.2　用户退出

当有成员离开组播组时，为了确保离开成员不能访问未来的通信数据，必须对其所拥有的密钥进行更新，更新过程描述如下：

(1)组成员申请退出组播组。

假设成员 u_{011} 退出组播组，则需要更新的节点密钥为 $SK(r), K_0(r), K_{01}(r), K_{011}(r)$。

(2)组控制器建立广播信息。

1)组控制器随机的选择广播消息 $B_3: (a_3, b_3)$，然后根据两点 $(a_3, b_3), (f(010), y_{010})$，利用插值定理构造一条直线：

$$F_3(x,y): y - y_{010}(r+1) = \frac{b_3 - y_{010}(r+1)}{a_3 - f(010)}(x - f(010))$$

2)组控制器将 $x=0$ 代入 $F_3(x,y)$，计算节点密钥 $K_{01}(r+1) = F_3(0)$；

3)组控制器根据两点 $(f(00), y_{00}(r+1)), (f(01), y_{01}(r+1))$ 利用插值定理构造直线：

$$F_2(x,y): y - y_{00}(r+1) = \frac{y_{01}(r+1) - y_{00}(r+1)}{f(01) - f(00)}(x - f(00))$$

4)组控制器将 $x=0$ 代入，确定 $K_0(r+1) = F_2(0)$，确定广播消息 $B_2: (a_2, b_2) \in F_2(x,y)$，且不等于以上 3 点；

同理，组控制器确定会话密钥 $SK(r+1)$ 和广播消息 $B_1: (a_1, b_1)$。

(3)组控制器广播消息 $B = \{B_1, B_2, B_3, \mu_{r+1}\}$。

(4)组成员计算更新后的密钥。

当成员收到消息后，根据所掌握的辅助密钥计算会话密钥。成员 u_{010} 利用私钥和 B_3 重构 $F_3(x,y)$，将 $x=0$ 代入获得 $K_{01}(r+1)$，然后再依次计算 $K_0(r+1), SK(r+1)$；成员 u_{000}, u_{001} 利用 $K_{00}(r+1)$ 依次计算 $K_0(r+1), SK(r+1)$；成员 $u_{00}, u_{101}, u_{110}, u_{111}$ 利用 $K_1(r+1)$ 和 B_1 直接计算 $SK(r+1)$。

3　方案分析

3.1　安全性分析

从组密钥管理的安全性角度来看，本文算法满足组密钥管理要求：

(1)组会话密钥安全：

1)在用户变化时，组控制器随机的选取更新因子 μ_r，通过单向函数和更新因子 μ_r 保证组密钥和辅助节点密钥的新鲜性和随机性。

2)在密钥更新时，组控制器利用秘密共享理论，构造一阶多项式，然后广播公开信息，合法组成员可以根据广播信

息和自身掌握的节点密钥，利用插值定理重构一阶多项式，计算会话密钥。

3)由于节点和兄弟节点可以根据广播信息重构相同的多项式，而节点的位置信息是公开的，为了防止节点密钥被兄弟节点获得，设置单向函数 $h()$。即使兄弟节点获得 $y_i(r)$，也不能获得节点的密钥。

(2)前后向安全：在组成员发生变化时，组控制器随机的选取更新因子，更新会话密钥。组控制器根据组成员的节点密钥，利用 Lagrange 插值定理逐层构造一阶多项式 $F(x,y)$，随机在 $F(x,y)$ 上选取广播信息 B，只有组成员可以根据 B 利用插值定理，计算当次的会话密钥，保证了会话密钥的前后向安全。

(3)抗合谋攻击：对于退出组播组的一组成员，他们的节点密钥信息不属于 $F(x,y)$，不能通过广播信息 B 重构 $F(x,y)$，其合谋也无法计算出组会话秘密。

3.2 代价分析

在本文算法中，当成员发生变化时，组控制器需要更新会话密钥和相应的节点密钥，广播信息 $B=\{B_1,B_2,\cdots,B_j\}_{j=[1,2,\cdots,l\,bn]}$ 的大小只与密钥树的高度和变化成员数量有关，通信量比基本的 LKH 方案小。

在本文算法中，密钥更新时，用户利用广播信息和私钥通过 Lagrange 插值公式计算 $SK(r+1)$，不使用加密算法，降低了用户的计算量。组控制器需要根据需要更新的节点密钥构造一阶多项式，确定广播信息，计算量比文献[5-6]中的方案计算量小。

在本文算法中，组控制器和用户密钥存储量与基本 LKH 方案相同。

4 结束语

安全组播正在成为一个活跃的研究领域。解决大型动态组的组密钥更新问题是组播密钥管理的核心问题，本文结合 LKH 和秘密共享方案的优点，提出一个应用密钥树结构的基于秘密共享的组密钥更新算法，降低了用户的计算量，利用广播进行密钥更新，降低了通信量，适合有大量组成员且变化频繁的实时动态组播应用。

参考文献

[1] Wanner D, Harder E, Agee R. Key Management for Multicast: Issues and Architectures[S]. RFC 2627, 1999.

[2] Wong K, Gouda M, Lam S. Secure Group Communications Using Key Graphs[J]. IEEE/ACM Trans. on Networking, 2000, 8(1): 16-30.

[3] Sherman A, McGrew D. Key Establishment in Large Dynamic Groups Using One-way Function Trees[J]. IEEE Trans. on Software Engineering, 2003, 29(5): 444-458.

[4] 朱文涛, 熊继平, 李津生, 等. 安全组播中密钥分配问题的研究[J]. 软件学报, 2003, 14(12): 2052-2059.

[5] Staddon J, Miner S, Franklin M, et al. Self-healing Key Distribution with Revocation[C]//Proc. of IEEE Symposium on Security and Privacy. Berkeley, California, USA: IEEE Press, 2002: 224-240.

[6] Yang Ming. An Unconditionally Secure Multi-round Revocation Scheme Using Secret Sharing[C]//Proc. of CCN'02. Marina Del Rey, California, USA: [s. n], 2005: 31-37.

[7] 陈礼青, 张福泰. 基于秘密共享的动态安全组播密钥协商[J]. 计算机工程, 2008, 34(12): 147-149.

编辑 金胡考

~~~~~~

（上接第 148 页）

4.4 效率分析

(1)将需求归类，针对不同的需求进行相应的计算和存储分布，提高了文件存储和传输的效率。例如：类型 3 不需经过身份识别，直接定位获得文件或通过查询重组文件，不但保证了及时的可用性，同时也提高存储网络的效率。

(2)区分大小文件提高了文件存储效率，小文件不需要进行分片计算，不但可以减少计算量，也可节省带宽和节点的存储资源。

(3)采用纠删码来进行分片编码，在提高了安全性的同时，也提高了存储和发布的效率。系统中 RS 码可以采用系统码，以降低计算代价，使存储和重组文件效率更高[3]。

(4)对等式存储固有的特点使它本身就有很高的效率，并可以节省用户的存储资源。节点与节点之间可以直接通信，使其传输速度和效率都相对传统模式的存储网络有很大提高。

5 结束语

本文利用纠删码与对等网络的思想根据不同用户需求构建了一种新的可生存的开放式对等存储网络，该网络能够有效提高用户私密文件的机密性与安全性，并能够保证文件的可用性。提高了文件在网络中的抗灾性与传输效率，使网络服务、共享资源更便捷，适合于现实生活中大部分的网络用户需求，并可应用于现在大量的对等存储网络。下一步研究可根据系统设计分模块进行实验仿真，以改进完善设计，细化最优算法。

参考文献

[1] 田 敬, 代亚非. P2P 持久存储研究[J]. 软件学报, 2007, 18(6): 1379-1399.

[2] 王文奎, 吴国新. 一种对等式存储系统的设计与实现[J]. 计算机技术与发展, 2008, 18(4): 237-239.

[3] 田 敬, 代亚非. 对等存储系统中的数据可用性与安全性研究[D]. 北京: 北京大学, 2007.

[4] Tian Jing, Yang Zhi, Dai Yafei. SEC: A Practical Secure Erasure Coding Scheme for Peer-to-Peer Storage System[C]//Proceedings of the 14th Symposium on Storage System and Technology. Wuhan, China: [s. n.], 2006.

[5] Mirsky L. An Introduction to Linear Algebra[M]. [S. l.]: Dover Publications, 1990.

[6] 姜大光, 奚加鹏. 分布式存储系统(OceanStore)的复制策略[J]. 计算机工程与科学, 2008, 30(8): 144-146, 149.

编辑 金胡考
~~~~~~