Computer Science and Application 计算机科学与应用, 2013, 3, 302-306
http://dx.doi.org/10.12677/csa.2013.37053 Published Online October 2013 (http://www.hanspub.org/journal/csa.html)

Group Key Management Based on B-Tree and LKH*

Di Liu[1], Xin Zhou[2], Hui Li[1], Chao Xu[1]

[1]School of Computer Science, Beijing University of Posts and Telecommunications, Beijing
[2]The First Research Institute of the Ministry of Public Security of PRC, Beijing
Email: ld_bupt@163.com, casiezhou@sonicom.com.cn, lihuill@bupt.edu.cn, jessia19891012@126.com

Received: Sep. 3rd, 2013; revised: Sep. 24th, 2013; accepted: Oct. 8th, 2013

Abstract: The existing secure group schemes had following shortages: 1) Individual has to maintain large amount of keys with increasing number of users; 2) There will be too many key renew process affecting the efficiency. This paper proposes an efficient group key management scheme based on B-Tree and LKH, which can add branches into the tree when new members join in order to impose restrictions on height of tree, and reduce the number of stored key. In this scheme, the group key will be renewed when the secure group members join or quit in order to provide a safe multicast module. This scheme overcomes the above defects and improves the performance under large group size, and could appropriately be applied on large and dynamic multicast groups.

Keywords: Multicast; Key Management; LKH Tree; One-Way Function Tree

基于 LKH 树和 B 树的组密钥管理方案*

刘　迪[1]，周　昕[2]，李　晖[1]，徐　超[1]

[1]北京邮电大学计算机学院，北京
[2]公安部第一研究所，北京
Email: ld_bupt@163.com, casiezhou@sonicom.com.cn, lihuill@bupt.edu.cn, jessia19891012@126.com

收稿日期：2013 年 9 月 3 日；修回日期：2013 年 9 月 24 日；录用日期：2013 年 10 月 8 日

摘　要：已经提出的更新组密钥方法普遍存在 1) 随着用户量的递增，用户需要维护的密钥数量将几何增长；2) 密钥更新过程中的加密次数多，使得更新效率低等问题。本文提出一种基于 B 树的 LKH 组密钥管理方法，使得密钥树节点可以横向递增，从而限制了树的高度，减少了需要存储的密钥数量。当加密通话组的成员关系发生变化时，本方法通过更新组密钥保证前向后向安全，从而提供一种安全的多播服务。此协议的分析结果证明，该方法可以有效的减少密钥的存储量和发送量，并提高组密钥更新效率，适用于动态大用户量加密通话组。

关键词：多播；密钥管理；逻辑密钥树(LKH)；单向函数(OFT)

1. 引言

作为一种高效的通信方式，多播实现一点对多点的通信。目前，多播技术被广泛应用在网络音频/视频广播、AOD/VOD、网络视频会议、多媒体远程教育等方面。这些应用均允许组成员自由的加入或退出，并要求只有组成员可以接收到组内的通信内容。如果不采取安全措施，则多播就会带来信息被窃听、篡改、重放等安全危险。为了保护信息的机密性和有效性，就需要引入保密措施，即在通话组内引入组建共享的密钥——组密钥。在使用对称密钥算法情况下，对称组密钥是所有组成员共同拥有的密钥，通信时，发送消息的组成员把通信内容使用组密钥加密，这时候只有组内成员才能对通信内容进行有效解密。

多播的安全要求大体分为前向安全和后向安全前向安全保证了当有组成员离开此通话组时，离开的

*基金项目：国家自然科学基金资助项目(61070207)。

成员不能再获取退出时间以后的有效通信内容；后向安全保证了通话组有新成员加入此通话组时，新成员不能获取加入时间以前的有效通信内容。为了满足这样的安全需求，组密钥就必须随着组成员的变更而进行更新，并下发给有效的组成员。

组密钥在更新的过程中必须保证只被组成员成功获取，所以需要引入组密钥管理协议。对于组成员变更频繁的通话组而言，更新过程开销巨大。开销主要包括密钥生成，密钥加密解密次数，密钥分发，密钥存储和通信量等。近几年提出的密钥管理方面的协议例如 DHSA 和 CRC 协议等[1-6]，主要是探讨降低消息大小，密钥加密次数。这些协议主要是针对于系统存在一个中央管理系统——密钥管理系统(KMS)而提出的，这里的 KMS 需要负责管理组成员，管理组密钥，为组成员下发组密钥等。本文提出的 B-Tree-LKH 协议吸取了 LKH、OFT 以及 CRC 协议的优点，并加入新的思路。B-Tree-LKH 协议的优势在于使得通话组可以同时存在大量组成员并且每个成员只需保存很少的密钥数量，KMS 也只需维护少量的密钥数量，并极大的降低了加入通话组的开销，同时维持组成员离开通话组的开销保持在 O(n)，并且大规模减少了 KMS 和用户间的通信量。

2. 相关研究介绍

2.1. 层次逻辑密钥树 (Logical Key Hierarchy, LKH)

1998 年，D.Wallne 等人提出了应用逻辑密钥树(LKH)的密钥管理方法[7]，后来其他人陆续提出针对 LKH 改进方法[8-11]，对有安全要求的通话组进行组密钥管理。逻辑密钥树(LKH)不是某种实体，逻辑密钥树是 KMS 上维护的一种逻辑构架，每个组成员需要维护用户密钥和整个分支路径上的密钥。这个方法中的密钥树通常使用平衡二叉树，对于平衡二叉树而言，组成员需要维护至少 $O(\log_2 n)$个密钥，其中 n 是用户数。

当一个新组成员加入一个通话组时，需要在密钥树的某个非完全分支上增加一个节点或者增加一层，空出一个分支，并且此时组密钥和与此新增节点相关联的路径上节点密钥都需要更新。KMS 为新用户下发密钥更新消息，包含使用用户密钥加密的新组密钥和分支密钥；KMS 为其他在此路径上的组成员下发使用组成员自身用户密钥加密的路径上的新密钥(不同组用户的路径不同，需要的新密钥也不同)；KMS 使用各个可以使用的可覆盖最大范围的节点密钥为所有用户下发新组密钥。当一个组成员需要离开通话组时，同上面的步骤基本相似，同样更新相关路径节点密钥和下发新组密钥。通过以上步骤，保证了通话组的前向和后向通信安全。

2.2. 单向方程树(One-Way Function Tree, OFT)

用单向方程树管理密钥的方法[12,13]应用了单向方程(One-way Function, OF)计算密钥的方法替代了仅由 KMS 下发密钥的模式，所以密钥之间是相互依赖的，这个模式极大的降低了密钥更新的开销。每个组成员可以通过一个相同的单向函数计算用户密钥而无需 KMS 下发，单向函数树是一个混合方程，如下：

$$k_i = f\left(h\left(k_{\text{left}(i)}\right), h\left(k_{\text{right}(i)}\right)\right)$$

其中 k_i 是指用户密钥，$h\left(k_{\text{left}(i)}\right)$ 是新加入用户节点左边的节点提供的密钥经过单向函数获得的值，$h\left(k_{\text{right}(i)}\right)$ 由右边节点提供。

用此方法组成员维护的密钥个数和 LKH 比是不变的，但是降低了下发密钥消息的长度，与此同时，用户可自行计算用户密钥，无需 KMS 把用户密钥加密后下发，从而减少了 KMS 加密的次数。

2.3. 引入密码的密钥计算 (Coding for Key Calculation, CKC)

该协议[1,3]在逻辑密钥树的基础上引入密码的密钥计算，结合了逻辑密钥树和单向方程的优势，降低了更新密钥的开销，另外在密钥计算中引入 Code——即一个由用户在树形结构中维护的随机数，在单向方程中加入新变量，在用户离开，通话组更新组密钥和结点密钥的情况下，有效降低了消息长度和密钥加密次数。由于 Code 是自上向下继承的，所以 KMS 中应维护叶子节点所持有的随机数，以便在密钥树向下增加用户时可以继续继承密钥，而且在目前的条件下，此协议中 Code 的长度小于 1024 时，在实际应用中是不安全的，Code 至少为 1024。随着用户的增加，密钥树的深度是对数增加的，用户需要保存的 Code 是自身长度乘以密钥树深度，对于组成员而言开销是巨

大的，并且对于也需要存储所有用户 Code 的 KMS 存储开销也是很大的。

3. 本文提出的基于B树的LKH组密钥管理方案(B-Tree-LKH)

本文提出一种基于 B-Tree 和逻辑密钥树的组密钥更新协议，吸取了以上协议的优点，并且采用时 B 树的结构。下面将具体阐述一下 B-Tree-LKH 协议。

本协议采用集中型结构并且基于逻辑密钥树，即存在 KMS 对密钥进行管理下发，并且采用了 B 树的结构。故此协议适用于用户数量很大的通话组，保证了大量用户的情况下，加密通话组能够正常构建，不论用户数量多少，组成员只需保存常数个密钥，分别为组密钥、节点密钥和用户密钥。

协议中的组成员的用户密钥存在于树中叶子结点上，组密钥存在于树中根节点上，密钥树上的非叶子节点上存放着结点密钥，并且存在一个 KMS 用来维护密钥树。在使用对称密钥情况下，用户密钥只有 KMS 和此用户两者共享，是 KMS 下发只有此用户能够有效解密的消息时候使用的。结点密钥为此结点下的用户共同拥有，是 KMS 下发只有此结点下用户能够有效解密的消息使用的。组密钥是组内所有用户共同拥有的，用户把发送的消息使用组密钥，保证此消息只能由本组成员有效解密。在新组成员加入该通话组时，KMS 为该新组成员下发由自身用户密钥加密的新组密钥和新的节点密钥。其中新组密钥是利用单向函数使用旧组密钥计算出来的，新节点密钥是使用旧节点密钥和新组密钥计算出来的，故其他的成员可以自行计算出新的组密钥和节点密钥。当旧成员离开的时候，需要更新组密钥，其他与离开旧成员在不同分支上的组成员，会收到由 KMS 下发的由所属节点密钥加密的新组密钥，与离开旧成员属于同一分支上的组成员，会收到由 KMS 下发的由各自用户密钥加密的新组密钥，并且组成员使用单向函数和新组密钥异或旧节点密钥的值计算出新的节点密钥。

3.1. 新成员加入通话组

当新成员加入通话组之前，默认其已经配置了用户密钥，即由 KMS 和新成员共同拥有的密钥。新成员给 KMS 发送消息，并取得同 KMS 的连接，双方经过认证，确定是合法成员和合法 KMS 以后，KMS 查询密钥树中，是否有分支可加入新节点，如果存在不完全分支则把新节点配置在此分支下，如果不存在不完全分支则把新节点配置在新分支下，下发新组密钥，(新组密钥)生成方式如下：

$$k'_G = f\left(k_G\right) \tag{1}$$

也需要下发节点密钥，计算方式如下：

$$k'_{\text{node}} = f\left(k'_G \oplus k_{\text{node}}\right) \tag{2}$$

其中 k_{node} 不存在时，即此分支为新分支时，KMS 直接发送一个新密钥即可。其他处于同一节点下的节点需要更新节点密钥，计算方式如方程(2)。所有属于本通话组的用户需要通过方程(1)获得新的组密钥。

接下来举例说明，图 1 表示存在一个通话组，现存用户数量为 8 个，分别是{u1,u2,u3,u4,u5,u6,u7,u8}。现在有一新用户 u9 需要加入此通话组。

那么此时，KMS 将给 u9 下发使用 k_9 加密新组密钥 k'_G 和新节点密钥 k'_{N3}，消息如下：

$$Msg' = f\left(\left(k'_G\right)_{k9} \middle| \left(k'_{N3}\right)_{K9}\right)$$

{u7,u8}通过方程(1)(2)计算新组密钥 k'_G 和新节点密钥 k'_{N3}，{u1,u2,u3,u4,u5,u6}通过方程(1)计算新组密钥 k'_G。本过程相当于只下发一条消息并且旧组成员避免了解密新密钥的开销。新成员 u9 加入通话组如图 2 所示。

此时，如果存在一个用户 u10 需要加入此通话组，同样需要 KMS 和新成员相互认证，当密钥管理系统查询密钥树的时候，并不存在不完全分支，这时需要把 u10 放在新分支下，KMS 下发新组密钥和分支密钥，消息如下：

$$Msg' = f\left(\left(k''_G\right)_{k10} \middle| \left(k_{N4}\right)_{K10}\right)$$

{u1,u2,u3,u4,u5,u6,u7,u8,u9}通过方程(1)计算新组密钥 k''_G，节点密钥无需更新。新成员 u10 加入通话组如图 3 所示。

3.2. 旧成员退出通话组

当已有成员退出加密通话组时，需要更新组密钥和节点密钥，首先与退出已有成员在同一节点下的组成员将会收到由 KMS 发送的新组密钥，此时因为离开的已有成员拥有旧组密钥和单向方程，因此，新组密钥不能简单由旧组密钥通过单向方程生成。已有成

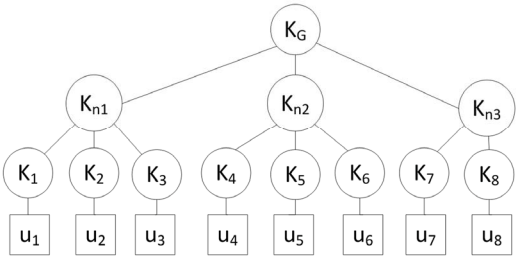

Figure 1. The group before new member u9 joins
图 1. 新成员 u9 未加入时的通话组

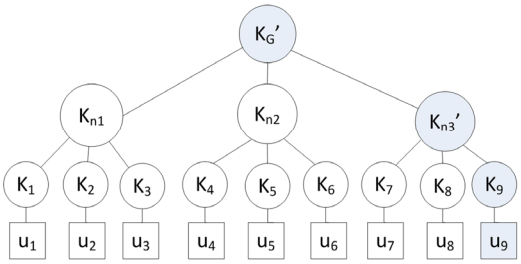

Figure 2. The group after new member u9 joins
图 2. 新成员 u9 加入通话组

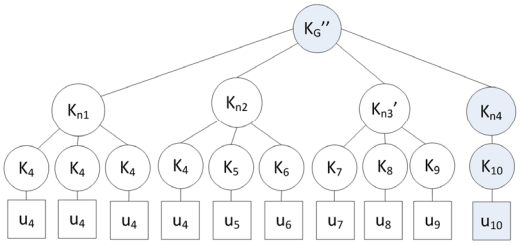

Figure 3. New member u10 joins the group
图 3. 新成员 u10 加入通话组

员所属的密钥树分支需要更新节点密钥，但此新节点密钥无需下发，可以由属于此节点的组成员利用新组密钥、旧节点密钥和单向方程自身生成。

$$k'_{\text{node}} = f\left(k'_G \oplus k_{\text{node}}\right) \tag{3}$$

举例说明，图 4 表示存在一个通话组，现存用户数量为 9 个，分别是{u1,u2,u3,u4,u5,u6, u7,u8,u9}，现在用户 u9 准备退出此通话组。

那么此时，k_G 和 k_{N3} 需要更新，如图 5 所示，其中{u1,u2,u3}将收到使用节点密钥加密的新组密钥，消息如下：

$$KMS \xrightarrow{\text{Multicast}} \begin{cases} u1\sim u3:\left\{\left(k'_G\right)_{Kn1}\right\} \\ u4\sim u6:\left\{\left(k'_G\right)_{Kn2}\right\} \\ u7:\left\{\left(k'_G\right)_{K7}\right\} \\ u8:\left\{\left(k'_G\right)_{K8}\right\} \end{cases}$$

其中{u7,u8}使用公式(3)计算新节点密钥 k'_{N3}。本过程

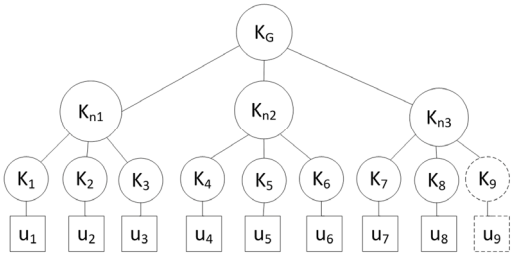

Figure 4. Old member u9 yet quit the group
图 4. 旧成员 u9 未退出时的通话组

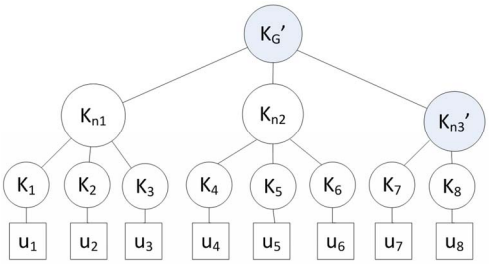

Figure 5. Existed member u9 quits the group
图 5. 旧成员 u9 已退出的通话组

中 KMS 只下发新的组密钥。

当退出的组成员分散退出，形成稀松密钥树的时候，重新构建此密钥树很容易。举例说明，当通话组形成稀松密钥树时，如图 6 所示。

此时 KMS 可以跟每个现存用户相互认证，然后可以发送重组消息。现存用户{u1,u4,u7}可以集中到一个节点分支中，这时用户收到 KMS 发送的重新构建的信息，用户通过公式(1)计算新组密钥，并且把旧组密钥当做新节点密钥，无需 KMS 发送新密钥就完成了密钥树重建。重构后的通话组密钥树如图 7 所示。

4. 安全分析和性能对比

本协议适用于一个动态加密通话组中需同时存在大量用户的情况。本协议可以满足加密通话组对于前向和后向安全的需求，在此基础上相对于其他几种集中式通话组管理方式性能有所提升。本协议的优势在于解除了随着用户量的增加，KMS 需要存储指数递增数量的密钥，KMS 只需要存储线性递增的密钥。表 1 中列出了几种协议的安全和性能的对比。

由此表中可看出，在保证安全性的基础上，B-Tree-LKH 协议中用户的存储开销为常数量级，相较于其他协议，极大地降低了同一通话组同时服务于大规模用户的开销。与此同时，KMS 的存储开销减少了很多，由指数递增变为线性递增。在新用户加入加

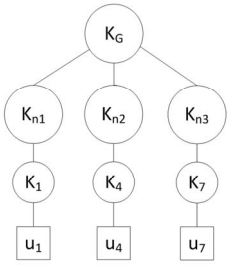

Figure 6. The group reform into sparse tree
图 6. 通话组形成稀松密钥树

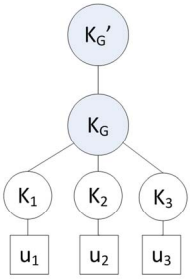

Figure 7. The group key tree (Refactored)
图 7. 已重构的通话组密钥树

Table 1. Security analysis and performance comparison
表 1. 安全分析和性能对比

模式	安全性		需要下发的密钥		存储开销	
	前向	后向	加入	离开	KMS	用户
LKH	Y	Y	O (Log_2n)	O (Log_2n)	O (2^n)	O (Log_2n)
OFT	Y	Y	O (Log_2n)	O (Log_2n)	O (2^n)	O (Log_2n)
CKC	Y	Y	O (1)	O (Log_2n)	O (2^n)	O ($nlog_2n$)
B-Tree-LKH	Y	Y	O (1)	O (n)	O (n)	O (1)

密通话组的操作中，维持了常数的开销量级。已存在用户离开的开销略高于其他模式，但是也控制在线性开销量级。另外，虽然 B-Tree-LKH 协议中比其它已有协议增加了使用单向方程重新计算密钥的开销，但是减少了对密钥的加密操作。由于加密解密操作的开销远大于使用单向函数计算密钥，所以避免加密解密操作将有效降低整个协议的开销。

5. 总结

本文提出了一个新型组密钥管理协议——基于 LKH 和 B-Tree 的组密钥管理方案。当一个新组成员加入该组的时候，KMS 只需向此新成员发送一个新组密钥和相应的节点密钥。接下来已经存在的组成员可以通过已有的组密钥和单向函数计算出新组密钥，并且生成新节点密钥。与此同理，当组成员离开的时候，KMS 需要向所有其余组成员发送一个新的组密钥，然后其余组成员可以通过新组密钥和旧节点密钥计算出新组密钥。通过上述方式，本协议保证了前向安全和后向安全，并且在同一通话组中含有大量用户的情况下，极大的降低了用户和 KMS 需要的存储空间。当用户加入时，更新加密通话组只需发送常数级的加密密钥消息，其他已存在成员可以自行更新。整体而言，通过单向方程避免了大量加密解密运算，并且减少了需要发送的消息量，节省了带宽。

参考文献 (References)

[1] S. Rafaeli, D. Hutchison. A survey of key management for secure group communication. ACM Computing Surveys, 2003, 35(3): 309-329.
[2] M. Hajyvahabzadeh, E. Eidkhani, S. A. Mortazavi and A. N. Pour. A new group key management protocol using code for key calculation: CKC. IEEE, 2010.
[3] S. Anahita Mortazavi, A. N. Pour and T. Kato. An efficient distributed group key management using hierarchical approach with diffie-Hellman and symmetric algorithm: DHSA. IEEE, 2011.
[4] H. R. Hassen, H. Bettahar, A. Bouadbdallah and Y. Challal. An efficient key management scheme for content access control for linear hierarchies. Computer Networks, 2012, 56: 2107-2118.
[5] R. Velumadhava Rao, K. Selvamani and R. Elakkiya. A secure key transfer protocol for group communication. Advanced Computing: An International Journal (ACIJ), 2012, 3(6): 83-90.
[6] Y.-R. Chen, W.-G. Tzeng. Efficient and provably-secure group key management scheme using key derivation. IEEE 11th International Conference on Trust, Security and Computing and Communications, 2012.
[7] D. Wallne, E. Harder and R. Agee. Key management for multicast: Issues and architectures. National Security Agency, RFC2627, 1999.
[8] D. Je, S. Seo, Y. Park and J. Lee. Computation and storage efficient key tree management protocol for secure multicast communications. Computer Communications, Elsevier, 2009.
[9] B. Jiang, X. Hu. A survey of group key management. International Conference on Computer Science and Software Engineering, 2008: 994-1002.
[10] Z. He and Y. Li. Dynamic key management in a user hierarchy. 2nd International Conference on Anti-Counterfeiting, Security and Identification, 2008: 298-300.
[11] Y. Piao, et al. Polynomial-based key management for secure intra-group and inter-group communication. Computers and Mathematics with Applications, 2012.
[12] D. Balenson, D. McGrew and A. Sherman. Key management for large dynamic groups: One-way functions trees and amortized initialization. IETF Internet Draft, 1999.
[13] D. A. McGrew, A. T. Sherman. Key establishment in large dynamic groups using one-way function trees. IEEE Transactions on Software Engineering, 2003, 29(5): 444-458.