一种动态组播的密钥更新算法

高俊伟　杨宗凯
（华中科技大学电子与信息工程系　武汉 430074）

摘　要　为了保证组播通信的机密性，安全组播使用不为组外成员所知的密钥来加密数据，并随组成员关系的变化而动态更新。基于树型分层式密钥管理方式使用户变更时的密钥更新代价减小，但前提是密钥树必须保持平衡。本文提出了一种应用 m 序 B 树作为组密钥树的密钥管理方法，在组播组中加入一个新成员，本方案比传统方案减少了密钥更新开销量，提高密钥更新效率。

关键词　密钥管理，安全组播，密钥更新

A Rekeying Algorithms for Dynamics Multicast

GAO Jun-Wei　YANG Zong-Kai
(Dept. of Electronics and Information Engineering, Huazhong Univ. of Sci. & Tech., Wuhan 430074)

Abstract　To provide communication confidentiality in multicasting applications, traffic data in secure multicast is encrypted with a session key known only by certificated group members. Whenever there is a change in the group membership, the session key must be updated dynamically. The hierarchical key-tree approach can achieve logarithmic rekeying costs. However, the efficient of hierarchical key-tree depends on whether the key tree remains balanced. In this paper, we propose to use an order-m B-tree as multicasting key management trees. the rekeying cost of inserting a new member is reduced compared to using the traditional hierarchical key-tree.

Keywords　Key management, Secure multicast, Rekeying

1　引言

自从 20 世纪 80 年代提出组播通信以来，随着 Internet 技术的发展，组播通信研究已经取得了巨大的成果，出现了很多新兴的基于组播的业务，如视频点播、计算机协同工作、网络电视会议等。组播路由协议、组播管理协议及组播安全是组播研究的重点。用于管理组播的因特网组管理协议 IGMP[1]不提供成员接入控制，用户只要获知特定业务使用的组播地址就可向路由器发送 IGMP 成员报告，不经审核地加入群组并获得 UDP 数据的拷贝。因此，现有组播通信并不保障数据的安全。保护组播数据机密、建立安全通信系统是安全组播研究的主要目标[2]。

组播通信的安全基础是所有组员共享一个不为组外用户所知的会话加密密钥 SEK(Session Encryption Key)加密组播通信数据。由于组播通信的组员随时可能加入或离开通信组，所以密钥必须随时更新，以使新加入成员无法访问过去的通信数据(后向安全性)，以及离开的成员无法解读将来的通信数据(前向安全性)。相比单播的密钥管理，前向加密(Forward Confidentiality)、后向加密(Backward Confidentiality)和同谋破解(Collusion)是组播密钥管理特有的问题。密钥管理是安全组播研究的核心问题。

2　相关研究工作

根据组密钥的产生方式，组密钥管理可分为两大类：分散式密钥管理和集中式密钥管理。

2.1　分散式组密钥管理方案

分散式密钥管理是组播通信没有固定的组管理员，由所有组成员共同建立组密钥，也称为组密钥协商(Groupkey Agreement)。在这种方案中，所有组成员视为同等的，这种密钥管理方式适用于分散的、没有领导的组结构。Steiner 等[3]将 Diffie-Hellman 密钥交换算法推广到组通信密钥管理，可为频繁变化的动态群组提供会话密钥分配服务。然而 Diffie-Hellman 算法的速度类似 RSA 公钥算法的速度，使得这个方案不能很好应用于大数量的群组。在文献[4]中，Steiner 等进一步发展了基于 Diffie-Hellman 算法的组密钥协商，提出具有可证明安全性的密钥协商协议套：Cliques。A Teniese 等[5]讨论了组密钥协商应用程序接口(Application Programming Interface, API)的设计，而 API 是基于密钥协商协议套 CLIQUES 实现的。这些方案的缺点是计算量大，难以应用在大数量的群组中。

2.2　集中式密钥管理方案

集中式密钥管理(Centralized Key Management)，是组播中存在一个组控制器(Group Controller，简称 GC)，负责全组的密钥生成、分发和更新，系统开销正比于组规模 N。集中式密钥分配方法是一种被大量采用的方法，适合于有组领导者的组结构。集中式密钥管理又可进一步分为非树型密钥管理方案(平坦型密钥管理方案)和基于树型密钥管理方案(层次型密钥管理方案)。

2.2.1　非树型密钥管理方案

这种管理方法是由组管理员将组密钥直接分发给每个组成员。GKMP[6]是非树型密钥管理方案的代表，此方案中全体成员除共享 SEK 外，还使用一个加密 SEK 的“密钥加密密钥”(Key Encryption Key) KEK。当 SEK 过期时，GC 使用 KEK 加密 SEK，记作 KEK(SEK)。KEK(SEK)表示用 KEY

来加密 SEK，SEK 是待加密消息。每个成员 M_i 还各自与 GC 共享一个秘密的 KEK_i。新成员 M_j 加入时，GC 生成 SEK'和KEK'，用原有成员皆知的 KEK 加密它们，然后将 $KEK(SEK', KEK')$组播给所有 $M_i(i\neq j)$；GC 再把 KEK_j (SEK', KEK') 单播给 M_j。这样新成员加入的 rekey 操作只包括一次组播和一次单播。当规模为 N 的组中有成员 M_k 退出时，GC 依次对所有成员 $M_i(i\neq k)$单播 $KEK_i(SEK', KEK')$共是 $N-1$ 次单播。GC 的存储复杂度和通信复杂度都是 $O(n)$，故该方案只适合小规模组播。

2.2.2 基于树型密钥管理(Tree Based)

非树型密钥管理的共同特点是只适合于小型组播，随组规模 n 的增大其性能显著下降。Ballardie[7] 基于核心树的组播结构提出了可扩展组播密钥分配方案。在这个方案中，组密钥分配中心的功能是控制对群组的访问，其基本思想是：初级核心节点产生会话密钥，然后将其分发到下一级的核心节点，依次扩张树的结构，形成分配树。这个方案需要绝对信任每一个核心节点，并且不能更换组密钥，除非建立一个新的群组。文献[8,9]分别各自提出了基于二叉树结构的集中密钥管理方案。在文献[8]提出的管理方案中，所有密钥材料全部由组管理员集中管理，用二叉树结构存储密钥材料，树的叶节点表示组成员，根节点对应着组密钥，而其余各节点对应一个 KEK，每一个内节点的所有后代共享这个节点密钥 KEK，每个组成员各自持有从所在叶节点到根节点的路径上所有节点密钥。文献[10]通过引入辅助 KEK 构建密钥树，把 SEK 更新的通信开销降到 $O(\log N)$，称为逻辑密钥层次(Logical Key Hierarchy，LKH)方案。

基于分层的管理方法的优点在于可以很好适应频繁变化的群组，易于扩展，可应用于任意大小型号的组播群组。缺点是管理过于集中，对组管理者的安全性依赖过强，且加重组管理者的管理负担，同时密钥树必须保持平衡，加大了通信开销。

3 对逻辑密钥层次树的改进

3.1 LKH 树的密钥更新分析

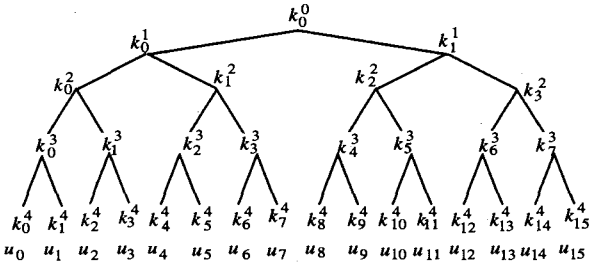

图 1 安全组播密钥管理树

图 1 所示的是一棵度为 2、叶节点个数为 16 的逻辑密钥树 LKH，对应于一个成员个数为 16 的群组。树根节点代表组控制器 GC(Group Controller)，拥有会话加密密钥 SEK；叶节点代表组成员，分别拥有 $k_0^4 \sim k_0^{15}$，为 U_i 与 GC 共享的 $KEK_i, i\in[0,15]$。虚拟的内部节点对应于中介 KEK，用于 SEK 更新时向组内特定范围的组成员传递新密钥(新密钥既包括 SEK，也包括那些需要更换的中介 KEK)。每个成员拥有从代表它的叶节点到根节点这条路径上的所有密钥，例如成员 U_0 拥有的密钥是$\{k_0^4, k_0^3, k_0^2, k_0^1\}$，其中 k_0^0 是为所有成员共有的 SEK，k_0^4 是仅为 GC 和 U_1 所知的 KEK_1，而 k_0^1，k_0^2 和 k_0^3 则从 GC 处获得，并用来限制密文消息，只组播到包含 U_0 在内的有限集合(例如 $U_8 \sim U_{15}$ 不能解读用 k_0^1 加密的消息，$U_4 \sim U_7$ 能解读用 k_0^1 加密的消息但不能解读用 k_0^2，k_0^3 加密的消息)。

假设密钥树为平衡树或接近平衡，取树为全满情形(即 N 为 d 的整次幂) 作分析，此时树高度为 $h=\log_d N$，成员 U_{15} 退出组播所要进行的密钥更新次数是[10]

① $GC\rightarrow u_{14}$ 单播 $k_{14}^4(k_7^3\rightarrow k_7'^3)$；

② $GC\rightarrow\{u_{14}\}$ 单播 $k_7'^3(k_3^2\rightarrow k_3'^2)$；

③$GC\rightarrow\{u_{12}, u_{13}\}$ 组播 $k_6^3(k_3^2\rightarrow k_3'^2)$；

④$GC\rightarrow\{u_{12}, u_{13}, u_{14}\}$ 组播 $k_3'^2(k_1^1\rightarrow k_1'^1)$；

⑤$GC\rightarrow\{u_8, u_9, u_{10}, u_{11}\}$ 组播 $k_2^2(k_1^1\rightarrow k_1'^1)$；

⑥$GC\rightarrow\{u_8\sim u_{14}\}$ 组播 $k_1'^1(k_0^0\rightarrow k_0'^0)$；

⑦$GC\rightarrow\{u_0\sim u_7\}$ 组播 $k_0^1(k_0^0\rightarrow k_0'^0)$；

成员 U_{15} 加入组播所要进行的密钥更新是[11]

① $GC\rightarrow u_{15}$ 单播 $k_{15}^4(k_7^3\rightarrow k_7'^3)$；

② $GC\rightarrow u_{14}$ 单播 $k_7^3(k_7^3\rightarrow k_7'^3)$；

③ $GC\rightarrow u_{15}$ 单播 $k_{15}^4(k_3^2\rightarrow k_3'^2)$；

④ $GC\rightarrow u_{15}$ 单播 $k_{15}^4(k_1^1\rightarrow k_1'^1)$；

⑤$GC\rightarrow\{u_{12}, u_{13}, u_{14}\}$ 组播 $k_3^2(k_3^2\rightarrow k_3'^2)$；

⑥$GC\rightarrow u_{15}$ 单播 $k_{15}^4(k_0^0\rightarrow k_0'^0)$；

⑦$GC\rightarrow\{u_8, u_9, u_{10}, u_{11}\}$ 组播 $k_1^1(k_1^1\rightarrow k_1'^1)$；

⑧$GC\rightarrow\{u_0\sim u_{14}\}$ 组播 $k_0^0(k_0^0\rightarrow k_0'^0)$；

LKH 将组成员关系变动时的 Rekey 次数降低到 $O(\log_d n)$。可是维持密钥树平衡需要加大通信开销，势必加重组控制器负担。本文提出了一种应用 m 序 B 树作为组密钥树，不论密钥树是否平衡，在组播组中插入一个新成员，本方案比传统方案减少了密钥更新开销量，提高密钥更新效率。

3.2 对 LKH 树的改进

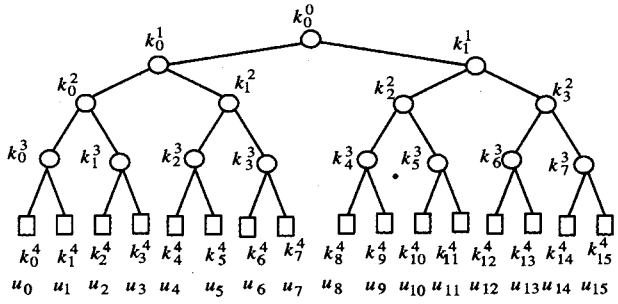

图 2 安全组播密钥管理树，(树高 $h=4$ 不包括外部节点，圆圈表示中间节点，矩形表示外部节点)

图 2 中小圆点代表中间节点，小矩形代表组播成员即元素。树的级(Level)：指定树根的级为 1，Root. Level=1。其孩子与父亲的关系为：x. Son. Level=x. Parent. Level+1。元素的度(Degree of an Element)是指其孩子的个数。叶节点的度为 0，树的度(Degree of a Tree)是其元素度的最大值。二叉树的高度(height)是指该二叉树的层数，图 2 中树的层是 5。

定义 m 序 B-树(B-Tree of Order m)是一棵 m 叉搜索树，m 叉搜索树可以为空树，如果 B-树非空，那么相应的扩充树满足下列特征：根节点孩子个数介于$(2, m)$；除了根节点以外，所有内部节点孩子个数介于(d, m)，$d=\lceil m/2\rceil$($\lceil m/2\rceil$指对 $m/2$ 取整)；所有外部节点位于同一层上。

(下转第 86 页)

为了更好地描述数据特征，我们进行了多组实验，其中部分实验结果见表1和表2。

表1 $w=5,\rho=0.95,\varepsilon=33$ 的 F_{rate}

SCCTS	PAA	RPAA
Normal	0.67	0.54
Cyclic	0.53	0.47
Increasing Trend	0.60	0.23
Decreasing Trend	0.59	0.26
Upward Shift	0.51	0.44
Downward Shift	0.47	0.43

表2 $w=5,\rho=0.99,\varepsilon=15$ 的 F_{rate}

EEG	PAA	RPAA
A_1_co2a0000364	0.33	0.27
C_1_co2c0000337	0.28	0.15
A_m_co2a0000364	0.37	0.20
C_n_co2c0000337	0.31	0.14

4.4 实验分析

从表1和表2的实验结果中可以看出，RPAA较之PAA的结果有较大进步，能够进一步减少错查率。对于EEG的表示则能够比较好地拟合。

但PAA和RPAA均对SCCTS存在较大的错查率，这对数据量巨大的情形(如以G、T计算)，则是一种挑战。分析其原因是下界定理虽然保证了不漏查，但是在索引空间中的点距仍然比较大，所以在索引空间中不能够较大数量地剔除 $D(T,C)\leqslant\varepsilon$ 的记录。

结论 本文将时间特性引入到时间序列表示和距离度量中，提出了RPAA的表示方法。在时间序列相似性查找时兼顾了时间序列的整体特征和随时间变化的特性，进一步缩小了索引空间中点对距离，在查询过程中能够有效地减少查错率。

但在实验中我们发现，中间结果中错查率还有进一步缩小的空间，初步分析是PAA和RPAA虽然满足了时间序列降维的下界定理，保证无漏查，但是在索引比较的过程中的距离计算仍有缩小的余地。下一步工作将进一步研究时间序列降维下界定理，争取给出更加严格的时间序列降维下界定理，以缩小索引空间与实际数据空间的差距。

参考文献

1 Agrawal R, Faloutsos C, Swami A. Efficient Similarity search in sequence databases. In: Lomet D, ed. Proceedings of the 4th Conference on Foundations of Data Organization and Algorithms. Chicago, Illinois: Springer Verlag, 1993. 69～84

2 Chan K, Fu W. Efficient Time Series Matching by Wavelets. In: Proceedings of the 15th IEEE International Conference on Data Engineering. Sydney, Australia, 1999. 126～133

3 Faloutsos C, Ranganathan M, Manolopoulos Y. Fast subsequence matching in time-series databases. In: Proceedings of ACM SIGMOD Conference. Minneapolis, 1994. 419～429

4 Keogh E, Kasetty S. On the need for time series data mining benchmarks: a survey and empirical demonstration. In: the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Edmonton, Canada, 2002. 102～111

5 Keogh E, Pazzani M. A simple dimensionality reduction technique for fast similarity search in large time series databases. In: 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining. Kyoto, 2000

6 UCI KDD Archive. http://kdd.ics.uci.edu/, 2006

7 Zhu Yunyue. High Performance Data Mining in Time Series: Techniques and Case Studies: [PHD. Thesis]. New York City: N. Y. University, 2004. 138～139

(上接第55页)

定理1 设B树的高为 h，n 是外部节点的个数，且 $d=\lceil m/2\rceil$($\lceil m/2\rceil$ 指对 $m/2$ 取整)，则有：

$d^{h-1}+1\leqslant n\leqslant m^h$ 且 $\log_m n\leqslant h\leqslant\log_d(n-1)+1$。

证明：在树的第一层有1个节点，第二层有至多 m^1 个节点，第三层有至多 m^2……第 x 层有至多 m^x 个节点，因此该树至多有 m^h 个外部节点。在树的第一层有1个节点，第二层有至少 $d^{2-1}+1$ 个节点，第三层有至少 $d^{3-1}+1$……第 $x-1$ 层有至少 $d^{x-2}+1$ 个节点，第 x 层有至少 $d^{x-1}+1$ 个节点。因此有 $d^{h-1}+1\leqslant n\leqslant m^h$ 成立，由 $d^{h-1}+1\leqslant n\leqslant m^h$ 直接得到 $\log_m n\leqslant h\leqslant\log_d(n-1)+1$。

定理2 设 T 是一棵高度为 h 的 m 序B树，$d=\lceil m/2\rceil$ 且 n 是 T 中的节点个数，则 $2d^{h-1}\leqslant n\leqslant m^h$ 且 $\log_m n\leqslant h\leqslant\log_d(n/2)+1$。

证明：n 的上限源于 T 是一棵 m 叉搜索树。对于下限，注意相应的扩充B-树的外部节点都在 $h+1$ 层，而1,2,3,4,…,$h+1$ 层的节点最小数目是 $1,2,2d,2d^2,\cdots,2d^{h-1}$，因此B-树中外部节点的最小数是 $2d^{h-1}$。由于外部节点的数量比元素的个数多1，因此 $2d^{h-1}\leqslant n$，由 $2d^{h-1}\leqslant n\leqslant m^h$ 可直接得到 $\log_m n\leqslant h\leqslant\log_d(n/2)+1$。

成员加入且密钥更新开销最大情况下，如果加入点在第 r 层，那么密钥更新量是组播 $r+1$ 次，单播 h 次，总数是 $c_{join}=r+1+h$。因 $r\leqslant h-1$，所以：$c_{join}\leqslant 2h$。令 c_{join_max1} 和 c_{join_max2} 是最差情况下，密钥管理树分别为 m 序B树和B-树时，成员加入组播的最大密钥更新开销。有：$c_{join\_max1}\leqslant 2h_1$ 和 $c_{join\_max2}\leqslant 2h_2$，其中 h_1 和 h_2 分别是最差情况下，密钥管理树分别为 m 序B树和B-树时密钥树高。比较其开销量得：$c_{join\_max2}>c_{join\_max1}$。可见成员加入且组播密钥更新开销最大(即最差)情况下，本方案具有更小的密钥开销。

结束语 本文提出了一种利用m序B树实现密钥管理的方法。该方法的显著优点是新成员加入时在最差状况下的密钥更新开销比传统方案更小，提高了组播密钥更新效率。

参考文献

1 Cain B, Deering S, Kouvelas I, et al. Internet group management protocol [S]. Version 3. RFC3376, 2002

2 Wong Chung Kei, Gouda M, Lam S S. Secure group communications using key graphs [J]. IEEE/ ACM Trans. on Networking, 2000, 8(1): 16～30

3 Steiner M, Tsudik G, Waidner. Diffie-Hellman key distribution extended to group communications. In: Proc. the 3rd ACM Conference on Computer and Communications Security, New Delhi, India, 1996. 31～37

4 SteinerM, Tsudik G, Waidner. CLIQUES: A new approach to group key agreement. In: Proc. 18th IEEE International Conference on Distributed Computing Systems, Amsterdam, Netherlands, 1998. 380～387

5 A teniese G, Chevassut D, Detal H. The design of a group key agreement A P I. In: Proc. DARPA Information Survivability Conference and Exposition, SC, USA, 2000. 115～126

6 Harney H, Muckenhirn C. Group key management protocol (GKMP) architecture [S]. RFC2094, 1997

7 Ballardie T. Scalable Multicast Key Distribution. RFC 1949, 1996

8 Caronni G, WaldvogelM, Sun Detal. Efficient security for large and dynamic groups. In: Proc. the 7th Work shop on Enabling Technologies, (WETICE'98), Stanford, California

9 Dinsmore P T, Balenson D M, Metal H. Policy based security management for large dynamic groups: An overview of the DCCM project. In: Proc. the DARPA Information Survivability Conference & Exposition, SC, USA, 2000. 64～73

10 Wallner D, Harder E, Agee R. Key management for multicast: issues and architectures [S]. RFC2627, 1999

11 Goshi J, Ladner R E. Algorithms for Dynamic Multicast Key Distribution Trees. In: Proc. ACM Symp. Principles of Distributed Computing (PODC 2003), 2003