安全组播密钥批更新算法研究

屈 劲[1],葛建华[1],蒋 铭[2]
(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. 上海交通大学电子信息学院,上海 200030)

摘 要: 密钥更新效率是影响组播安全性能的重要因素,将密钥组织成密钥逻辑树,对密钥树进行周期性批更新是提高密钥分发效率、降低密钥管理系统和用户负担的有效途径. 本文提出一种密钥批更新算法,利用该算法可以进一步降低密钥更新对密钥管理系统和用户的负担;并利用该算法分析了密钥批更新的性能.

关键词: 组播; 密钥分发; 密钥树; 指数分布
中图分类号: TP393 **文献标识码**: A **文章编号**: 0372-2112 (2003) 07-1046-03

On Batch Rekey Algorithm for Secure Multicast

QU Jin[1], GE Jian-hua[1], JIANG Ming[2]
(1. *Ministry of Education Key Lab of Computer Network and Information Security, Xidian University, Xi'an, Shaanxi* 710071, *China*;
2. *School of Electronics & Information Technology, Shanghai Jiaotong University, Shanghai* 200030, *China*)

Abstract: The efficiency of re-keying is a main factor that influences the performance of secure multicast. Grouping keys into logical tree and periodic batch re-keying for users join/leave are effective ways to improve the efficiency of re-keying and reduce the burden that re-keying puts on key management as well as users. In this paper, we present a re-key algorithm which reduces the burden that re-keying put on key management system and users. We then investigate the performance of batch re-keying using our algorithm.

Key words: multicast; key distribution; key tree; exponential distribution

1 引言

随着互联网的迅速发展,出现了很多新兴的基于安全组播模型的业务,如收费多媒体分发、用户受限远程会议、收费多方游戏、多方计算、股票及彩票信息分发等等. 在这个模型中,组播密钥管理系统 KMS(Key Management System)为合法用户分发和更新密钥,所有组播数据流均由数据密钥 DK(Data Key)加密传输,DK 由会话密钥 SK(Session Key)加密分发,保证只有合法用户才能得到 DK 并解密组播数据流.

组播安全应满足前向安全和后向安全. 前向安全,即用户离开组播组后,该用户不能解密将来的组播数据流;后向安全必须保证新用户加入组播组后,该用户可以解密将来的组播数据流,但不能解密过去的组播数据.

由于组播环境下密钥管理问题复杂,迄今还没有一种很好的密钥管理方法. 如果分别为每位合法用户分发和更新 SK,n 为用户数,则一个用户离开组播组的密钥更新量为 $O(n)$. 文[1,2,3]采用密钥逻辑树将密钥更新代价降低到 $O(\log_d n)$,其中 d 为树的维数,当用户变更频繁时,该方法的密钥更新量仍然很大. 文[4,5]在前者基础上批更新密钥,该方法收集一段时间内的用户加入或离开组的请求,每隔一段时间集中更新密钥,这种方法可以进一步减小了密钥更新量,提高密钥更新可靠性. 但批更新不可避免的引入用户请求响应时延,即申请加入(离开)的用户在一定时间内不能(可以)接收组播数据流. 文献[5,7]提出了保证前后向安全的密钥批更新算法,该算法在用户离开和加入组播组时都需要更新密钥树,算法复杂,密钥更新量较大.

用户动态行为使系统具有一定后向安全性. 在组播组中,用户离开时为保证前向安全需要更新密钥树,此前的组播信息对新加入用户是安全的,而此后的组播信息不安全,因此如果新加入用户所在子组在该用户加入前存在用户离开,就可认为系统具有一定的后向安全性,只要这段时间足够短,就可以认为后向安全.

本文依据上述思想提出一种密钥批更新算法,该算法只在用户离开时更新密钥树,进一步降低了密钥更新量;并分析了密钥批更新性能;最后本文还分析了该算法的安全性,表明可以通过增加子组用户数提高后向安全性.

2 密钥树

文献[1,2,3]使用密钥逻辑树管理 DK,DK 由 SK 加密分发. 密钥树 $K(V, E)$ 是一个有向树,树中每一个节点代表一个密钥,根节点表示 SK,叶节点代表用户私钥 PK(Private Key),中间节点代表组密钥 GK(Group Key). V 为树的节点集合,E

收稿日期:2001-04-04;修回日期:2002-07-02
基金项目:国家教育部高校骨干教师资助项目;陕西省自然科学基金(No. 2002f08)

为树的边集合. 由于 PK和用户一一对应,本文不区分用户和 PK. 任意节点指向树根的方向称为该节点所在树枝(路径)方向,由于该节点指向树根的路径唯一,定义节点的路径 P_{PK_i}为 PK_i 所在路径的节点的集合,如图 1 中左图 PK_4 的路径为 $P_{PK_4}=\{PK_4, GK_{4-6}, SK_{1-9}\}$. 对任意节点而言,方向指向该节点的路径称为接入路径,背离该节点的路径称为接出路径. 用户私钥 PK_i 的路径长度为 h_i,树的高度(层数) $h=\max_i\{h_i\}$. 树的维数 d 为树节点的最大接入路径数. KMS 利用密钥树管理密钥,任意组用户必须存储该用户私钥所在路径的所有密钥才能同步更新 SK,解密 DK进而解密组播数据流.

图 1 中,用户 u_9 离开组播组时,KMS 为保证前向安全需更新离开用户所在路径密钥,发送下述信息更新密钥树:$(\{GK_{7-8}\}_{PK_7}, \{GK_{7-8}\}_{PK_8}, \{SK_{1-8}\}_{GK_{1-3}}, \{SK_{1-8}\}_{GK_{4-6}}, \{SK_{1-8}\}_{GK_{7-8}})$,其中$\{m\}_k$ 表示用密钥 k 加密信息 m.

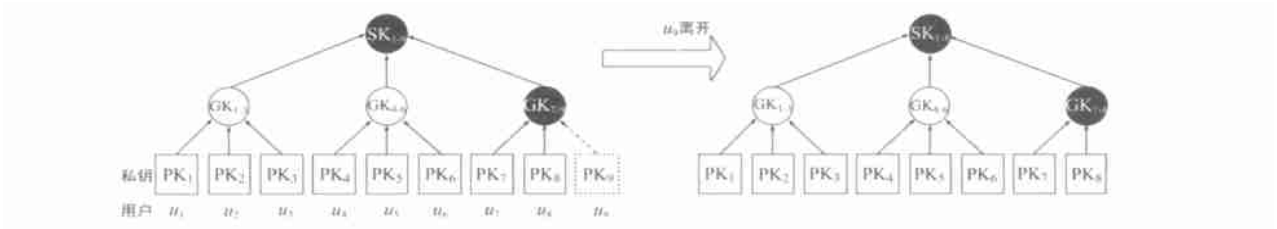

图 1　密钥树及用户离开时的密钥更新

3　密钥批更新算法

本节首先提出一种新的节点标记方法,然后在该方法基础上提出密钥批更新算法.

文[5]中的标记方法给密钥树中所有节点从上到下、从左到右统一安排序号,主要缺点是节点分裂加入新用户时原有的排序规律被打破,这给以后确定节点位置带来困难. 本文采用坐标的方式标记节点,保证新加入用户继续遵循原有坐标定义规则,而用户离开时很容易确定需要更新的组密钥位置.

节点坐标标记方法:

® 所有节点的输入路径从左到右分别标记为 1,2, …, d, 其中 d 为树的维数.

® 叶节点坐标由所在路径标记的逆向(由树根指向树叶)排列组成,中间节点的坐标由该节点指向根节点的路径标记逆向排列加上 0 组成,根节点的坐标由全 0 组成;

® 坐标的维数为树高 h.

如图 2 所示, PK_1 的坐标为[1,1], PK_4 的坐标为[2,1], GK_{1-3}的坐标为[1,0], SK的坐标为[0,0]等等.

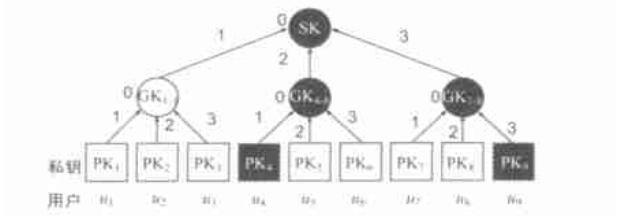

图 2　节点坐标及删除树

为了便于描述密钥更新算法,定义下述概念:

定义 1　如果存在密钥树 $K(\mathbf{V},\mathbf{E})$,离开用户集 $\mathbf{PK_L}=\{PK_{i+1}, \cdots, PK_{i+k}\}$,其中 k 为离开用户数,则更新节点集合 $\mathbf{V_d}=\{\bigcup_{j=1}^{k} P_{PK_{i+j}} \mid PK_{i+j}\in\mathbf{PK}\}$,其中 P_{PK_i}表示 PK_i 所在路径,删除子树 $K_d(\mathbf{V_d},\mathbf{E_d})$为 $K(\mathbf{V},\mathbf{E})$在 $\mathbf{V_d}$ 上的导出子树.

定义 2　如果存在密钥树 $K(\mathbf{V},\mathbf{E})$, $K_d(\mathbf{V_d},\mathbf{E_d})$为密钥树的删除子树, $K_d(\mathbf{V_d},\mathbf{E_d})$的相关节点集合 R_{K_d}定义为 $\mathbf{V_d}$ 的邻接节点的集合.

如图 2 所示,$\{u_4, u_9\}$离开组播组,则节点$\{$SK, GK_{4-6}, GK_{7-9}, PK_4, $PK_9\}$组成删除子树,$\{GK_{1-3}, PK_5, PK_6, PK_7, PK_8\}$为删除子树的相关节点.

3.1　多个用户加入组播组的密钥分配算法

在用户加入时并不更新密钥树,只是将用户加入密钥树的所在路径节点密钥分配给用户. 算法如下:

® 采用文献[5]的方法确定用户 u_i 加入位置,确定加入位置坐标$[a_1, a_2, \cdots, a_h]$,其中 $a_i\in\{1,2,\cdots,d\}$;

® 根据 u_i 的坐标$[a_1, a_2, \cdots, a_h]$确定用户所在路径各节点坐标,$[a_1, a_2, \cdots, a_{h-1}, 0]\cdots[a_1, 0\cdots0], [0, 0\cdots0]$;

® 将上述节点处密钥用 PK_i 加密传输给 u_i.

有关用户加入密钥树的位置确定方法请参考文献[5]. 根据上述用户加入的密钥分配算法很容易得到下面结论:

定理 1　如果批更新间隔的加入用户数为 n_J,加入用户位置高度$\{h_i \mid i=1, \cdots, n_J\}$,则密钥分发量 $W_J=\sum_{i=1}^{n_J} h_i$.

3.2　多个用户离开组播组的密钥树更新算法

本算法只在用户离开组播组时更新密钥树. 设离开用户集合 $\mathbf{PK_L}=\{PK_i, PK_{i+1}, \cdots, PK_{i+k}\}$,算法描述如下:

® 根据离开用户坐标确定离开用户路径$\{P_{PK_{i+j}} \mid j=1, \cdots, k\}$;

® 确定删除树节点集合 $\mathbf{V_d}=\{\bigcup_{j=1}^{k} P_{PK_{i+j}} \mid PK_{i+j}\in\mathbf{PK_L}\}$及其相关节点集合 R_{K_d}及相应节点坐标;

® 使用相关节点处密钥加密其各级上层密钥并广播分发.

根据上述多个用户离开的密钥更新算法很容易得到下面结论.

定理 2　如果删除树 $K_d(\mathbf{V_d},\mathbf{E_d})$的上层节点(除叶节点外)数为 n_d,该删除树的相关节点数 $n_R=|R_{K_d}|$,则总的密钥更新量 $W_L=n_d+n_R-1$.

4　密钥批分发性能分析

为便于分析,假设密钥树为平衡树,即 $N=d^h$,其中 h 是

树高,d 为树的维数.

在证明定理 3 前,先说明下述引理.

引理 1 平衡密钥树中任意用户离开组播组的密钥更新量 $W_1 = dh - 1$,只与该用户的高度和树的维数有关,而与该用户所处的子树位置无关.

当离开用户均匀分布在密钥树的叶节点时所需的密钥更新量最大,可以得到下述定理:

定理 3 当离开用户在密钥树的叶节点均匀分布时密钥更新量最大,如果 N 表示总用户数,n_L 表示批更新间隔的离开用户数,h 表示树高:当 $n_L > d^{h-1}$,最大密钥更新量 $W_{L\max} = d(d^h - 1)/(d - 1) - n_L$;当 $n_L < d^{h-1}$,表示 $n_L = d^k + r$,其中 $0 \leq k \leq h - 2, 0 < r < (d - 1)d^k$,则 $W_{L\max} = d(d^{k+1} - 1)/(d - 1) + n_L d(h - k - 1) - 2n_L$.

证明 当 $n_L > d^{h-1}$,由于假设密钥树为平衡树,且离开用户在密钥树中均匀分布,第 $h - 1$ 层以上所有密钥均需要更新,该结构的删除树上层节点(除叶节点外)数为:$n_d = d^{h-1} + d^{h-2} + \cdots + d + 1 = (d^h - 1)/(d - 1)$,相关节点数为:$n_R = N - L$,由定理 2 可知:$W = n_d + n_R - 1 = d(d^h - 1)/(d - 1) - n_L$.

当 $n_L < d^{h-1}$,表示 $n_L = d^k + r$,其中 $0 \leq k \leq h - 2, 0 < r < (d - 1)d^k$,即 $n_L > d^k$,整个问题的求解分为两部分:k 层以下密钥更新问题;第 k 层及以上密钥更新问题.

第 k 层以上的密钥更新等价于 $n_L > d^{h-1}$,因此,$W_{high} = d(d^{k+1} - 1)/(d - 1) - n_L$.

在 k 层以下的删除节点实际是一个个的孤立节点,所以离开用户和第 $k + 1$ 层的节点之间构成孤立树,树高 $h^* = h - k - 1$ 和其他的删除节点没有关系,K 层以下的密钥更新量是各删除节点单独密钥更新的数量和:$W_{\text{low}} = n_L W_1$,其中 W_1 表示平衡树中任意用户离开的密钥更新量,根据引理 1 得:$W_1 = dh^* - 1 = d(h - k - 1) - 1$. 则 $W_{\text{high}} = n_L W_1 = n_L d(h - k - 1) - n_L$.

综上所述,$W_{L\max} = W_{\text{low}} + W_{\text{high}} = d(d^{k+1} - 1)/(d - 1) + n_L d(h - k - 1) - 2n_L$.

综合上述定理 1、定理 3 的内容可以得到定理 4.

定理 4 对于用户数为 N 的密钥树 $K(V, E)$,如果批更新间隔加入用户数为 n_J,离开用户数为 n_d,则批更新间隔内的最大密钥分发量 $W_{\max} = W_{L\max} + \sum_{i=1}^{n_J} h_i$.

由于本算法将用户加入和离开分别考虑,而用户加入并不更新密钥树,因此密钥更新量小.

5 安全性分析

在上述密钥更新算法中,用户离开时密钥树更新密钥以保证前向安全,而当新用户加入组播组时只给新用户分发了保证用户正常接收未来组播数据流的密钥,而没有更新密钥树的原有密钥,因此系统对于用户的存储攻击存在一定安全隐患. 下面分析这个问题.

定理 5 假设用户在 t_0 时刻加入组播组,且任意用户在组播组中的持续时间满足负指数分布[6],$1/\lambda$ 为用户平均持续时间,如果该用户对 $(t_0 - t_s, t_0)$ 时间段内组播信息存储攻击成功的概率大于 P_0 时系统不安全,则不安全的时间 $t_s < -(1/\lambda d)\ln(P_0)$,其中 d 为子组用户数.

证明 新加入用户攻击 $(t_0 - t_s, t_0)$ 内组播信息成功的概率等于这段时间内没有用户离开的概率,这段时间里用户 u_i 不离开组的概率为:$P(T > t_0 \mid T > t_0 - t_s) = \exp(-\lambda t_s)$. 又因为子组中 d 个用户独立同分布,$(t_0 - t_s, t_0)$ 内没有用户离开的概率为:$P = (\exp(-\lambda t_s))^d = \exp(-\lambda d t_s)$. 如果这个概率大于 P_0 时不安全,则不安全的时间:

$$t_s < -(1/\lambda d)\ln(P_0)$$

上述结论表明只要 t_s 足够小就可保证本文算法具有一定后向安全性,并可通过增加 d 提高安全性.

6 总结

密钥批更新可以降低密钥分发量并提高系统可靠性. 本文提出一种密钥批更新算法,只在用户离开时更新密钥树,该算法可以降低密钥更新量,而且很容易实现;本文进一步讨论了采用该算法的密钥批更新性能;最后本文分析了该算法的后向安全性,表明可以通过增加子组用户数提高后向安全性.

参考文献:

[1] RFC2236. Internet Group Management Protocol, V. 2. [S].

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

[3] C K Wong, M G Gouda, S S Lam. Secure group communications using key graphs [A]. Proceedings of ACM SIGCOMM '98 [C]. New York: ACM Press, 1998. 68 - 79.

[4] S Setia, S Koussih, S Jajodia, E Harder. Kronos: a scalable group rekeying approach for secure multicast [A]. IEEE Symposium on Security and Privacy [C]. Oakland (USA) CA: IEEE Computer Society Press, 2000. 215 - 228.

[5] X S Li, Y R Yang, M G Gouda, S S Lam. Batch rekeying for secure group communications [A]. Proceedings of Tenth International World Wide Web Conference [C]. Hongkong: ACM, 2001.

[6] K Almeoth, M Ammar. Collection and modeling of the join/leave behavior of multicast group members in mbone [A]. Proceedings of High Performance Distributed Computing Focus Workshop (HPDC '96) [C]. New York: HPDC, 1996.

[7] Y R Yang, X S Li, X B Zhang, S S Lam. Reliable group rekeying: a performance analysis [A]. Proceedings of ACM SIGCOMM '01 [C]. San Diego, California: ACM, 2001.

作者简介:

屈 劲 男,1971 年生于陕西宁强县,1994 年毕业于重庆大学应用物理系,获学士学位,1999 年于西安电子科技大学获硕士学位,目前为西安电子科技大学在读博士生,主要兴趣方向是信息安全和数字电视.

葛建华 (见本期第 1091 页)