

引入加入树的贡献型组密钥更新方案

曹震寰¹, 李黎², 顾小卓³, 车彦刚¹

(1. 甘肃省信息中心, 兰州 730030; 2. 甘肃省财政厅, 兰州 730000;

3. 兰州城市学院 信息工程学院, 兰州 730060)

(matergg@126.com)

摘要:在大规模动态群组中,一个高效的能实时更新的组密钥管理算法是提供组通信内容保护的前提。提出了一种基于加入树的分布式组播密钥管理方案 JDH。首先给出了一个包含主树和加入树的新型树形结构。其次,为了降低成员加入时的时间复杂度,给出了一种新的加入算法。最后,根据最优化方法选取了最优的加入树的大小。理论分析和仿真表明,JDH 将成员加入时密钥更新的时间复杂度降低为 $O(1)$ 。

关键词:组安全通信;贡献型组密钥管理;逻辑密钥树

中图分类号: TP393 **文献标志码:** A

Join-tree-based contributory group key management scheme for key update

CAO Zhen-huan¹, LI Li², GU Xiao-zhuo³, CHE Yan-gang¹

(1. Gansu Information Center, Lanzhou Gansu 730030, China;

2. Finance Department of Gansu Province, Lanzhou Gansu 730000, China;

3. School of Information Engineering, Lanzhou City University, Lanzhou Gansu 730060, China)

Abstract: To provide content protection in large groups with highly dynamic memberships, a secure group key management efficient in key establishment and update is the foundation. In this paper, a join-tree-based contributory group key management (JDH) was presented to achieve better time efficiency. First, a new key tree topology comprised of main tree and join tree was put forward. Then, a new join algorithm in the join tree was proposed to reduce the time complexity. Last, optimal capacity of the join tree was selected through optimization method. The theoretical analysis and simulations show that the asymptotic average join time is complexity reduced to $O(1)$.

Key words: secure group communication; contributory group key management; logical key hierarchy

0 引言

由于安全组通信中数据的私密性和完整性通常和组密钥紧密地联系在一起,建立和管理组密钥就成为组安全机制中最基础的内容。同时由于组播组中成员是动态变化的,需要实时更新组密钥来防止新加入的成员获得过去的通信内容,已经离开的成员获得以后的通信内容。这就需要组成员共同协商一个密钥建立和更新的密钥管理算法。

组密钥管理争论的焦点是由谁来产生、如何产生以及什么时候产生组密钥。什么时候产生组密钥相对来说比较简单,因为当组成员关系发生变化时,组密钥必须更新。由谁来产生和如何产生相对比较棘手,因为这两个因素决定了实际的组密钥管理算法。根据这两个因素,文献[1-2]将组密钥管理分为三种类型:集中式、分布式和贡献式。集中式组密钥管理方案^[3-6]采用单个实体作为密钥服务器来产生和安全地分发组密钥。分布式组密钥管理方案^[7]更适合于对等群组通信,它动态选择组成员作为密钥分发服务器,但是同样需要密钥服务器与每个现有的成员维持长期的双方安全通道来分发组密钥。研究发现与贡献型组密钥管理方案依靠模指数运算不同,许多集中式和分布式的密钥管理方案依靠对称加密来分发组密钥。因此,集中式和分布式的组密钥管理方案不能提供完美向前保密(Perfect Forward Secrecy, PFS)。但是,它们扩展到大型群组时比贡献型的方案开销小。

贡献型组密钥管理算法的核心是每个成员都对密钥的管理和产生贡献自己的份额,减轻了在集中式和分布式组密钥管理算法中的单点失效问题和信任问题。它使用一个包含了所有成员贡献的函数来生成组密钥,可以提供密钥的独立性和完美向前保密。但是,与集中式和分布式密钥算法相比,贡献型密钥管理需要相对较重的模指数运算和额外的组成员之间的通信开销。

许多早期的贡献型组密钥管理方案^[8-11]在研究组密钥建立的效率时致力于减小密钥协商时的轮数或者消息的大小。Ingemarsson 等人^[8]提出了基于环状拓扑结构的会话密钥分发系统(Conference Key Distribution System, CKDS)。Burmester 等人^[9]提出了一个以高的通信量为代价,但是计算高效的密钥管理方案 BD。Steiner 等人在文献[10]中将两方的 Diffie-Hellman (DH)^[12]协议扩展为多方 DH 协议,并以此为基础在文献[11]中提出了 Cliques。在 Cliques 中,每一个成员都接收到前一个成员传递的密钥,并把自己的贡献计算到新的密钥中,然后传递给下一个成员。Cliques 在成员离开和组分裂时是高效的,但是环中的最后一个成员的工作量很大。Kim 等人^[2]提出了基于树形的贡献型密钥管理算法并命名为 TGDH。TGDH 按照逻辑密钥树组织密钥,每一个成员仅需要知道它所在路径上的密钥。这就表明当密钥更新时,工作量被分配给了每个成员。Mao 等人^[13]提出了 JET,通过采用加入树和离开树将成员加入和离开时间开销的近似上界

收稿日期:2010-07-06;修回日期:2010-08-31。

作者简介:曹震寰(1976-),男,甘肃庄浪人,工程师,主要研究方向:网络安全;李黎(1977-),女,甘肃兰州人,主要研究方向:信息系统;顾小卓(1978-),女,甘肃白银人,副教授,博士,主要研究方向:网络安全;车彦刚(1979-),男,陕西渭南人,主要研究方向:网络安全。

从 $O(\log n)$ 降为 $O(\log(\log n))$ 。JET 使用多个加入事件或离开事件的平均时间开销来衡量密钥更新的复杂度。此外, JET 中离开树的引入也是基于一种前提, 即成员在组中的停留时间已知。

对于组成员具有不同计算能力和代价约束的情况, Trappe 等人认为仅仅减少总的带宽或者轮数的数量是不够的。他们提出了基于 Huffman 编码的会话密钥树来优化综合了成员不同耗费和资源约束的耗费函数, 称之为 Huffman 方案^[14]。Zheng 等人^[17]提出了一种新的组密钥协商方案 CCEGK, 这种方案总是把新成员插入到密钥树的根部, 因此在成员加入时密钥更新的开销很少。这种算法带来的结果是一棵不平衡的密钥树, 虽然在达到一定条件后, 密钥树会进行调整, 但是在调整之前, 密钥树的高度将远远超出平衡密钥树, 因此会增加成员离开时的时间开销, 并且平衡密钥树的工作量也很大。

本文借鉴 JET 方案中的加入树, 将密钥树组织为主树和加入树的树型拓扑结构。加入树作为一个缓存来接纳新加入的成员, 等加入树达到一定容量或者有成员离开时, 将加入树中的成员批量移动到主树中。与 JET 不同, 本文在加入树中采用了新的加入算法, 将新成员加入到加入树的根部, 通过这种加入算法降低了成员加入时的时间复杂度, 并把成员加入时间开销的近似渐进上界降低为 $O(1)$ 。当成员离开时, 移动加入树中的成员到主树, 降低了平均开销。本文分析的重点是密钥更新的效率, 假定加入的成员都是经过认证的, 也存在可靠的广播信道。

1 JDK

1.1 树拓扑结构

JDH 的密钥树结构如图 1 所示。用 N_M 和 N_J 分别表示主树和加入树中成员的个数, C_J 表示加入树的容量。

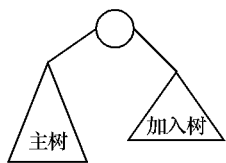


图1 JDH 的密钥树拓扑结构

1.2 加入算法描述

新成员的加入过程分为以下几个步骤。

1) 新成员向组播组发送一条包含它自身盲钥的组播消息。

2) 组中成员根据组播树确定新成员的插入位置和辅助者, 有三种情况:

a) 当加入树为空并且不需要被激活时, 新成员被插入到主树中。插入点的位置与 TGDH^[2] 中相似, 是主树中深度最浅的最右边节点; 辅助者是插入点所在子树中最右边的叶子节点。

b) 当加入树为空且需要被激活时, 将新加入成员加入到主树的根部, 并使新成员成为加入树的根节点, 辅助者是插入点所在子树中最右边的叶子节点。

c) 当加入树不为空时, 将新成员插入到加入树的根部, 辅助者是插入点所在子树中最右边的叶子节点。

成员连续加入加入树的情况如图 2 所示。

3) 辅助者更新自己的私钥, 计算它所在组播树路径上的所有私钥和盲钥, 并将新的盲钥向全组广播;

4) 其他成员在接到辅助者的消息后, 计算新的组密钥;

5) 如果加入树中的成员数量超过了加入树的容量, 则将加入树中的成员批量移动到主树中。加入树中成员的批量移动过程如图 3 所示, 分为以下几个步骤:

a) 在主树中发现 N_J 个最浅的叶子节点;

b) 将加入树中 N_J 个成员同时移动到主树中, 并对需要更新的密钥作标记;

c) 自下而上的更新所有打上标记的密钥;

d) 重新计算加入树的容量, 每个成员独立更新自己的密钥树。

采用上述批量移动方法是为了尽量维持主树的平衡, 减少主树的高度, 从而减少成员离开时的开销。

在移动过程中, 组通信没有中断。在新的组密钥产生以前, 可以用旧的组密钥进行通信。

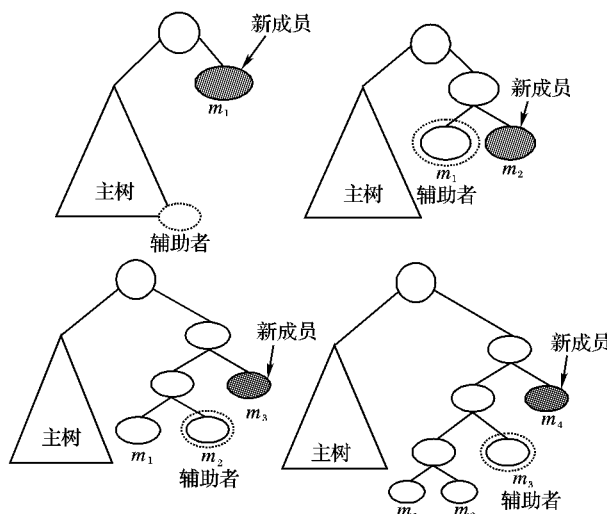


图2 成员连续加入加入树的变化情况

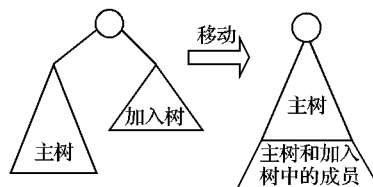


图3 加入树成员批量移动过程

1.3 离开算法描述

离开事件有自愿和被动的两种类型。如果成员在离开时发送离开请求, 那么这个成员更像是自愿离开的; 否则成员可能是被迫离开的。除了在第一情形下多了一条离开请求消息以外, 这两种离开操作是相似的。

JET^[13]在基于成员在组中的停留时间可知的情形下, 引入了离开树, 将要离开的成员移动到离开树中, 通过这种方式减少了开销。本文假设成员的离开信息不可获得, 没有采用离开树。当成员离开时, 将加入树中的成员移动到主树中, 通过这种方式分担了成员离开时的开销。

文献 [13] 表明, 成员在组中的停留时间通常较长, 因此成员更可能从主树中离开。当成员从主树离开时, JDH 将加入树中的成员移动到主树。最坏的情形是, 加入树为空, 在这种情况下, 离开开销与 TGDH^[2] 的开销相似, 为 $3h - 3$, h 为主树的高度。当成员在组中停留很短的时间, 该成员可能在还没有移动到主树中时就已经从加入树离开。离开开销依赖于成员离开的位置。最坏的情况是加入树已满, 但还没有进行批量移动, 这时离开所需的计算开销为 $3h' - 3$, h' 为加入树

的高度。在成员离开时,会将加入树中的成员批量移动到主树中,因此移动的成员分担了成员离开时的开销。

组密钥更新的步骤如下。

a) 每个成员单独更新自己的密钥树,删除离开成员的节点,由离开成员的兄弟节点代替离开成员的父节点。标记需要更新的密钥。

b) 当加入树不为空时,将加入树中的成员批量移动到主树中。标记需要更新的密钥。

c) 自下而上的更新所有标记的密钥。

d) 计算新的加入树的容量,更新密钥树。

1.4 性能分析

1) 指标说明。

文献[2,14]使用文献[15]中由 Alves-foss 等人提出的连续指数运算来衡量密钥更新时的计算开销,本文也使用该指标来衡量计算开销。与指数运算的开销相比,消息在高速局域网中的传递时延可以忽略不计,但当以较高的通信开销来换取较低的计算开销时,这个开销也是值得重视的。当然,在高时延的网络或者广域网中就显得更为重要。但在本文中,假定成员处于局域网中,忽略了通信时延,重点考虑组密钥生成过程中的计算开销。

DH 算法通常要求计算出一个模指数运算后,才能继续进行下一个运算。连续指数运算指一系列串行进行的模指数运算。在密钥更新时,根据文献[2],生成新的组密钥时所需的连续指数运算个数最多为 $3h-3$ (其中 $2h-2$ 是辅助者计算个数, $h-1$ 是其他成员计算个数), h 是密钥树的高度。

2) 加入树容量。

成员加入组播组的开销包括两部分,可表示为:

$$T_{\text{join}} = T_{\text{join}_s} + T_{\text{join}_b} \quad (1)$$

其中: T_{join_s} 表示成员加入加入树的开销, T_{join_b} 表示成员批量移动到主树中的平均开销。

在批量移动中,所需的连续指数运算次数最多为 $3h-3$, 其中 h 是成员移动后主树的高度。主树的高度不会超过 $\lceil \lg(N_M + N_J) \rceil + 1$ 。由于加入树的成员数目要远小于主树中的成员数目,主树高度的上限为 $\lceil \lg N_M \rceil + 2$ 。

使用连续指数运算表示的加入开销可表示为:

$$T_{\text{join}} = 6 + \frac{3\lceil \lg N_M \rceil + 6}{x} \quad (2)$$

其中 6 表示成员加入到加入树中的连续指数运算次数, x 表示加入树中成员的个数, $(3\lceil \lg N_M \rceil + 6)/x$ 表示批量移动时加入树中每个成员的平均开销。从式(2)可以看出,当 $x \rightarrow \infty$, $T_{\text{join}} \rightarrow 6$ 。也就是说,加入树中成员的个数越多,成员加入组播组所需的计算开销越小。但在现实中,当加入树增大时,成员从加入树离开的概率也随之增大。从图2可以看出,加入树中成员的数目与加入树的高度是相同的,即 $h' = C_J$, 因此,当成员从加入树离开时,一个大的加入树会带来较多的计算开销。因此,加入树的高度选择基于这样一个考虑,即成员从加入树离开的开销不应超过成员从主树离开的开销。因此,加入树的高度不应超过主树的高度。得到:

$$C_J = h' \leq \lceil \lg N_M \rceil + 1 \quad (3)$$

考虑到加入树中成员越多,加入开销越小,选择加入树容量为:

$$C_J = \lceil \lg N_M \rceil + 1 \quad (4)$$

由于主树的成员个数在两次批量移动之间是不变的,将式(4)代入式(2)可得平均加入时间的上限为:

$$T_{\text{join}} = 6 + \frac{3\lceil \lg N_M \rceil + 6}{\lceil \lg N_M \rceil + 1} \leq 10.5; N_M > 1, N_M \in \mathbf{Z}^+ \quad (5)$$

式(5)表明,一个成员为了加入安全组播组所需连续指数运算次数的上限为 10。成员加入时的密钥更新开销的渐近上界为 $O(1)$ 。

3) 加入树激活。

本文认为,当采用加入树后所需的加入开销小于不采用加入树的加入开销时,加入树应该被激活。采用加入树后,每个成员的最大平均加入时间可以表示为:

$$T_{\text{join}} = 6 + \frac{3\lceil \lg N_M \rceil + 6}{C_J} \quad (6)$$

如果没有采用加入树,则最大平均加入时间为 $3h-3$, 其中 $h = \lceil \lg N_M \rceil + 1$ 。当式(7)成立时,采用加入树可以减小加入开销:

$$6 + \frac{3\lceil \lg N_M \rceil + 6}{C_J} < 3\lceil \lg N_M \rceil + 3 \quad (7)$$

或者

$$\lceil \lg N_M \rceil > (C_J + 2)/(C_J - 1) \quad (8)$$

当 $C_J = \lceil \lg N_M \rceil + 1$ 时,不难证明式(8)对于所有的 $N_M > 4$ 都成立。因此,当组中的成员大于 4 时,就应该采用加入树。

2 理论分析及仿真验证

2.1 理论分析

本文对 TGDH、JET 和 JDH 的通信开销和计算开销进行了分析。进行比较的指标有简单轮数^[16]的个数、单播消息的个数、多播消息的个数、连续指数运算次数的个数和近似的计算开销的上界。表1对这些指标在最坏情况下的状况进行了总结。

表1 通信开销和计算开销比较

协议		通信开销			连续指数运算次数	计算开销上界
		简单轮数	单播	多播		
TGDH	加入	h	0	3	$3h-3$	$O(\lg n)$
	离开	h	0	1	$3h-3$	$O(\lg n)$
JET	加入	h'	0	2	$3h'-3$	$O(\lg(\lg n))$
	离开	h''	0	1	$3h''-3$	$O(\lg(\lg n))$
JDH	加入	3	0	2	6	$O(1)$
	离开	h	0	1	$3h-3$	$O(\lg n)$

注: h 是主树的高度, h' 是加入树的高度, h'' 是离开树的高度。

成员加入 从表1可以看出, TGDH 所需的计算开销最大, 为 $3h-3$, h 是密钥树的高度; JET 次之, 为 $3h'-3$, h' 是加入树的高度; JDH 最少, 只需要 6 次。它们所需计算开销的理论上限分别为: $O(\lg n)$, $O(\lg(\lg n))$ 和 $O(1)$ 。对于 JET 和 JDH, 只计算了成员加入加入树的开销, 而没有包含加入树中的成员移动时的开销。三个方案所需的消息数目基本相似。

成员离开 TGDH 与 JDH 所需的计算开销相似, 为 $3h-3$, 近似渐近上界为 $O(\lg n)$ 。在获得成员在组中的停留信息时, JET 要优于 TGDH 和 JDH 方案, 它的计算开销与离开树的高度有关, 为 $3h''-3$, h'' 为离开树的高度, 近似渐近上界为 $O(\lg(\lg n))$ 。

2.2 仿真

本文仅对成员加入时所需的计算开销进行了分析。在最坏情况下, JDH 所需的计算开销与 TGDH 相似。

2.2.1 成员加入时的平均计算开销

图 4 比较了 TGDH、JET 和 JDH 在成员陆续加入时所需的平均连续指数运算次数。从图中可以看出,当组规模小于 8 时,JET 和 TGDH 获得同样的性能;但是随着组的继续增大,JET 的性能要优于 TGDH。而对于 JDH,当组规模小于 4 时,它与 TGDH 和 JET 获得同样的性能,当组规模大于 4 时,由于加入树的激活,它的性能开始优于 TGDH 和 JET。另一个值得注意的现象是,JDH 的性能并不随着组规模的增大而下降。这说明 JDH 所需计算开销的近似渐进上界为 $O(1)$ 。

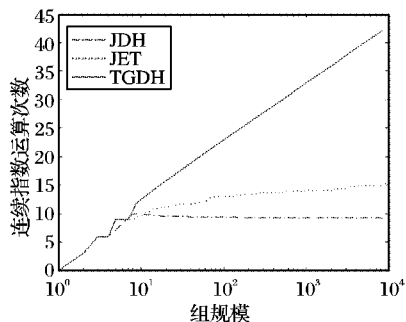


图 4 成员连续加入时所需计算开销比较

2.2.2 实验结果

假定安全组初始有 100 个成员,新成员陆续加入安全组。与文献[17]中的方法相似,将成员分为三类:计算能力强的成员,计算能力一般的成员和计算能力弱的成员。从文献[18]中可知,一次指数运算所需的平均时间为 2 ms,因此假设成员的计算时间来自于三个分布。对于每 10 个成员,2 个是计算能力强的成员,计算一次指数运算所需的时间服从均匀分布 $[0.1, 1]$; 5 个成员是计算能力中等的成员,计算一次指数运算所需的时间服从均匀分布 $[2.1, 3]$; 3 个是计算能力弱的成员,服从均匀分布 $[5.1, 6]$ 。图 5 比较了成员陆续加入安全组时所需密钥更新时间(没有考虑消息的传递时间),密钥更新时间是对 100 次仿真结果的平均。

对于 TGDH 方案,组密钥的更新时间与组中的成员数有关。因为在 TGDH 方案中,新成员总是被加入深度最浅的最右边节点。从图 5 可以看出,当组规模是 129 时,TGDH 所需的密钥更新时间最少。图 6 表示了 TGDH 在组规模为 129 时的密钥树结构,当新成员加入时,辅助者为 M_{129} ,仅需更新 $\langle 0,0 \rangle$ 和 $\langle 1,1 \rangle$ 节点的密钥,因此所需的密钥更新时间最少。之后随着组成员数的增加所需的加入时间也逐渐增多。

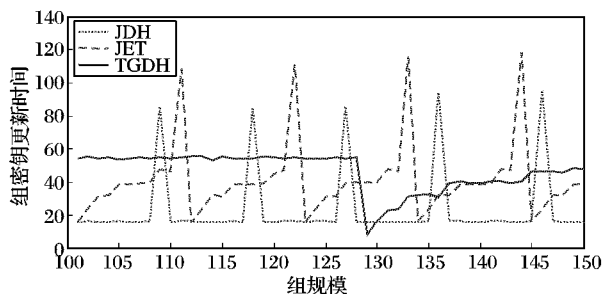


图 5 三种方案在成员陆续加入安全组时所需密钥更新时间比较

对比 JET 和 JDH 方案,可以看出 JDH 优于 JET。因为它们都使用加入树作为缓冲区,因此所需的加入时间要低于 TGDH 方案,但是当加入树中的成员批量移动到主树时,所需的加入时间要大于 TGDH。这个加入时间是成员移动时所需的总时间,而不是平均时间。与期望的相同,JDH 的加入时间是基本不变的,而 JET 是随着加入树规模的增加缓慢增大的。

由于 JET 的加入树容量要大于 JDH,JET 批量移动的时刻要晚于 JDH,批量移动时所需的时间也要大于 JDH。

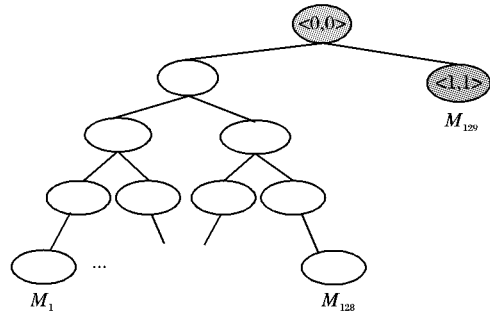


图 6 TGDH 在组规模为 129 时的密钥树结构

3 结语

借鉴 JET 方案中的加入树,本文提出了一种新的组密钥管理方案 JDH,加入树作为成员加入时的一个缓存区位于密钥树的根部。通过在加入树中采用新的加入算法,将新成员加入到加入树根部,成员加入时的密钥更新开销被降低为 $O(1)$ 。与现有的组播密钥管理算法的比较和仿真实验表明,JDH 在成员加入时是高效的。当成员离开时,同时移动加入树中的成员到主树,降低了平均离开开销。在最坏情况下,离开开销为 $O(\log n)$ 。

参考文献:

- [1] AMIR Y, KIM Y, NITA-ROTARU C, *et al.* On the performance of group key agreement protocols[J]. *ACM Transactions on Information System Security*, 2004, 7(3): 437-488.
- [2] KIM Y, PERRIG A, TSUDIK G. Tree-based group key agreement[J]. *ACM Transactions on Information System Security*, 2004, 7(1): 60-96.
- [3] WONG C K, GOUDA M, LAM S S. Secure group communications using key graphs[J]. *IEEE/ACM Transactions on Networking*, 2000, 8(1): 16-30.
- [4] RFC2627. Key management for multicast: Issues and architectures[S], 1999.
- [5] CANETTI R, GARAY J, ITKIS G, *et al.* Multicast security: A taxonomy and some efficient constructions[C]// 18th Annual Joint Conference of the IEEE Computer and Communications Societies. New York: IEEE, 1999: 708-716.
- [6] BANERJEE S, BHATTACHARJEE B. Scalable secure group communication over IP multicast[J]. *IEEE Journal on Selected Areas in Communications*, 2002, 20(8): 1511-1527.
- [7] BIRMAN R K, DOLEV D. Optimized group rekey for group communication systems[R]. Jerusalem: Hebrew University, 1999.
- [8] INGEMARSSON I, TANG D T, WONG C K. A conference key distribution system[J]. *IEEE Transactions on Information Theory*, 1982, 28(5): 714-720.
- [9] BURMESTER M, DESMEDT Y. A secure and efficient conference key distribution system[C]// Proceedings of EUROCRYPT. Berlin: Springer, 1994: 275-286.
- [10] STEINER M, TSUDIK G, WAIDNER M. Diffie-Hellman key distribution extended to group communication[C]// Proceedings of 3rd ACM Conference on Computer and Communications Security. New York: ACM, 1996: 31-37.
- [11] STEINER M, TSUDIK G, WAIDNER M. CLIQUES: A new approach to group key agreement[C]// IEEE International Conference on Distributed Computing Systems. New York: IEEE, 1998: 380-387.

$\alpha = 0.7, \beta = 0.2, \gamma = 0.1$, 交互次数 500, 节点总数 1000, 节点提供服务次数 200, 下载要求的信任值为 0.6, 反馈计算的变量值为 0.1。

在仿真中没有考虑系统后续更新信任值的操作, 只是针对本文解决的问题进行实验模拟, 并没有真正建立一个完善的 P2P 声誉系统。在实验中忽略的因素有: 未能考虑信任值的传输, 没有考虑节点的上传和下载因素, 所有实验节点没有中途退出的可能性, 一直模拟到实验结束。

1) 声誉系统中提供服务均衡性仿真实验。

为了验证本文提出的 P2P 声誉系统解决 P2P 网络中资源访问“热点”问题的效果, 本文在 JXTA 平台下使用 Java 语言编写了模拟程序, 在实验模拟中, 随机生成 1000 个节点, 其中规定节点序号是 200, 400, 600, 800 的节点可以提供网络中其他节点的请求下载服务其信任值根据本文提出的式(3)计算得出的关系为 200 号节点的信任值 > 400 号节点的信任值 > 600 号节点的信任值 > 800 号节点的信任值, 其余节点均没有相应的请求服务。本次实验一共进行了 500 次, 每次实验时规定请求服务的节点向网络中的其他节点请求 600 次服务, 并记录提供服务的节点提供的次数。最终, 对每个节点提供服务的次数取平均值。

绘制的图形如图 4 所示。图 4 中 A 代表增强型声誉系统, B 代表本文提出的 P2P 声誉系统。由实验可知, 增强型声誉系统在选择节点提供服务时总是选择信任值最高的节点, 因此很容易造成该节点拥塞, 而本文提出的 P2P 声誉系统会优先考虑信任值最高的节点提供服务, 但当服务次数到规定值 200 次时, 就选择其他节点提供服务, 不会造成资源访问“热点”问题。

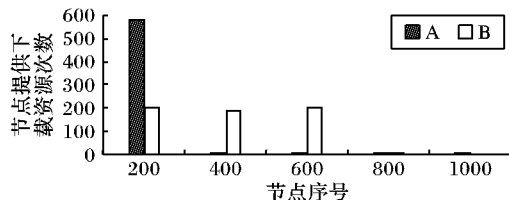


图4 声誉系统中提供服务均衡性仿真实验

2) 混合系统和增强型声誉系统交互成功率仿真。

在仿真过程中的恶意节点包括提供虚假的资源下载服务和带有病毒的节点。在实验中交互了 500 次, 对每次实验结果取平均值, 仿真图形如图 5 所示。

仿真刚开始时, 由于网络中恶意节点比较少, 其需要自动信任协商中访问控制策略的节点不多, 加入自动信任协商使协商成功率受到影响。由于执行自动信任协商比较费时, 影响了节点间交互的效率。但是, 随着恶意节点数的增多, 采用

访问控制策略相应增多的情况下, 增强型声誉系统中节点间交互成功率明显下降, 而本文提出的混合系统节点间交互成功率比增强型声誉系统有较大的提高, 这充分体现出本文提出的混合系统的价值。

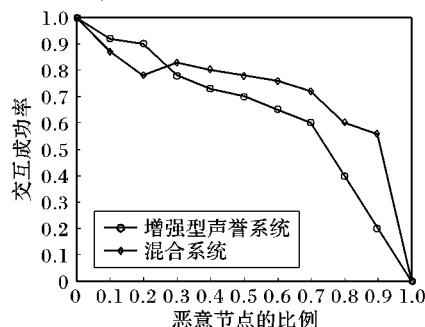


图5 交互成功率仿真实验

4 结语

本文在增强型声誉系统的基础上进行改进, 提出一种改进的 P2P 声誉系统, 该系统解决了 P2P 网络中节点提供服务的“热点”问题, 并在该系统基础上, 引入了自动信任协商机制, 进一步解决了 P2P 网络中的敏感信息泄露问题, 随着恶意节点的数量增多, 混合系统的优势得以充分体现。

参考文献:

- [1] 冯真, 张红旗, 刘育楠. 自动信任协商在 P2P 系统中的应用[J]. 计算机工程, 2007, 33(6): 132-133, 136.
- [2] EBay [EB/OL]. [2010-04-13]. <http://www.ebay.com>.
- [3] 冯真. P2P 环境下文件共享的声誉系统研究[D]. 郑州: 信息工程大学, 2006.
- [4] 廖振松, 金海, 李赤松, 等. 自动信任协商及其发展趋势[J]. 软件学报, 2006, 17(9): 1933-1948.
- [5] HOLT J E, BRANDSHAW R W, SEAMONS K E, et al. Hidden credentials[C]// WPES'03. New York: ACM, 2003 [2010-03-22]. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.2.5797&rep=rep1&type=pdf>.
- [6] 李建欣, 怀进鹏, 李先贤. 自动信任协商研究[J]. 软件学报, 2006, 17(1): 124-133.
- [7] LI JIANGTAO, LI NINGHUI. OACerts: Oblivious attribute certificates[J]. IEEE Transactions on Dependable and Secure Computing, 2006, 3(4): 340-352.
- [8] LI JIANGTAO, LI NINGHUI, WINSBOROUGH W H. Automated trust negotiation using cryptographic credentials[C]// ACM Conference on Computer and Communications Security. New York: ACM, 2005: 100-108.
- [9] 冯真, 张红旗, 刘育楠. 自动信任协商在 P2P 系统中的应用[J]. 计算机工程, 2007, 33(6): 132-133, 136.
- [10] EBay [EB/OL]. [2010-04-13]. <http://www.ebay.com>.
- [11] 冯真. P2P 环境下文件共享的声誉系统研究[D]. 郑州: 信息工程大学, 2006.
- [12] 廖振松, 金海, 李赤松, 等. 自动信任协商及其发展趋势[J]. 软件学报, 2006, 17(9): 1933-1948.
- [13] HOLT J E, BRANDSHAW R W, SEAMONS K E, et al. Hidden credentials[C]// WPES'03. New York: ACM, 2003 [2010-03-22]. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.2.5797&rep=rep1&type=pdf>.
- [14] 李建欣, 怀进鹏, 李先贤. 自动信任协商研究[J]. 软件学报, 2006, 17(1): 124-133.
- [15] LI JIANGTAO, LI NINGHUI. OACerts: Oblivious attribute certificates[J]. IEEE Transactions on Dependable and Secure Computing, 2006, 3(4): 340-352.
- [16] LI JIANGTAO, LI NINGHUI, WINSBOROUGH W H. Automated trust negotiation using cryptographic credentials[C]// ACM Conference on Computer and Communications Security. New York: ACM, 2005: 100-108.
- [17] DIFFIE W, HELLMAN M. New directions in cryptography[J]. IEEE Transactions on Information Theory, 1976, 22(6): 644-654.
- [18] MAO YINIAN, SUN YAN, WU MIN, et al. JET: Dynamic join-exit-tree amortization and scheduling for contributory key management[J]. IEEE/ACM Transactions on Networking, 2006, 14(5): 1128-1140.
- [19] ZHENG SHANYU, MANZ D, ALVES-FOSS J. A communication-computation efficient group key algorithm for large and dynamic groups[J]. Computer Networks, 2007, 51(1): 69-93.
- [20] ALVES-FOSS J. An efficient secure authenticated group key exchange algorithm for large and dynamic groups[EB/OL]. [2010-02-10]. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.74.1326>.
- [21] BECKER K, WILLE U. Communication complexity of group key distribution [C]// Proceedings of 5th ACM Conference on Computer and Communications Security. New York: ACM, 1998: 1-6.
- [22] TRAPPE W, WANG Y, LIU K J R. Resource-aware conference key establishment for heterogeneous networks[J]. IEEE/ACM Transactions on Networking, 2005, 13(2): 134-146.
- [23] AGARWAL D A, CHEVASSUTY O, THOMPSON M R, et al. An integrated solution for secure group communication in wide-area networks[C]// Proceedings of the 6th IEEE Symposium on Computers and Communications. New York: IEEE, 2001: 22.

(上接第 146 页)