基于洗牌策略的 Sybil 攻击防御

聂晓文[1], 卢显良[1], 唐 晖[2], 赵志军[2], 李玉军[1]
(1. 电子科技大学计算机学院, 四川成都 610054; 2 中国科学院声学研究所, 北京 100080)

摘 要: 洗牌策略从理论上解决了分布式哈希表(DHT)的 Sybil 攻击问题. 为克服敌手作弊, 引入受信节点构成分布式认证系统, 由受信节点对新加入节点进行认证, 保证节点签名和 ID 不能伪造; 同时引入记录洗牌加入过程的票据来判定节点合法性, 杜绝了敌手积累过期 ID. 由于保存票据的数量决定了论文提出算法的应用效果, 通过理论分析和仿真实验证实设计的算法需要保存的票据数量不大, 保证了算法的可行性.

关键词: Sybil 攻击; 洗牌策略; 对等网; 分布式哈希表 (DHT)
中图分类号: TP393.08 文献标识码: A 文章编号: 0372-2112 (2008) 11-2144-06

Resisting Sybil Attacks with Cards-Shuffling Scheme

NIE Xiao-wen[1], LU Xiar-liang[1], TANG Hui[2], ZHAO Zhi-jun[2], LI Yu-jun[1]
(1. *School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan* 610054, *China*;
2. *Institute of Acoustics, the Chinese Academy of Sciences, Beijing* 100080, *China*)

Abstract: The Cards-Shuffling scheme solves the Sybil attack in distributed hash table (DHT) theoretically. To overcome cheats of the enemy, a distributed authentication system which constructed by trusted nodes is proposed. The joining nodes are authenticated by the trusted nodes to ensure that the sign and identifiers of the nodes can not be fabricated. Tickets recording the joining process of Cards-Shuffling scheme are also introduced to verify the nodes so that the accumulation of expired identifiers is impossible. The performance of the algorithm proposed by the paper is determined by the number of tickets to be stored. Both the analysis and simulation confirm that there won't be a great number of tickets to be stored, which guarantees the feasibility of the algorithm.

Key words: Sybil attack; cards-shuffling scheme; peer-to-peer network; distributed hash table (DHT)

1 引言

分布式哈希表(DHT)算法由于高度可扩展性而赢得了人们的广泛关注[1,2], 但是对等网的开放性和自组织特性却给 DHT 的安全性带来了挑战.

在 DHT 的安全性中, Sybil[3] 攻击是一个被着重关注的课题. DHT 的路由表往往设计成具有特定的结构, 这使它更容易遭到攻击. Douceur[3] 认为: 如果敌手能够获得足够数量的 ID, 她就能部分或者全部地控制整个 DHT 网络. Sybil 攻击的表现形式有多种: 如自适应加入攻击[3]、Eclipse 攻击[4] 等. 举例来说, 如果某个机要节点上存放着称为"Top Secrets"的文件, 敌手试图阻止这个文件在 DHT 网络内扩散. 首先她申请足够多 ID 号, 然后根据路由表的构造方式选择特定 ID, 把攻击节点安插在机要节点的路由表内. 当机要节点的路由表项都被攻击节点占据, 它就只能通过攻击节点来访问网络. 如何抵御这种 Sybil 攻击, 是 DHT 算法在开放的应用环境中不可回避的问题.

为应对 Sybil 攻击, 人们提出了各种防御方案. 在这些方案中, 洗牌策略[5] 最具应用前景. 但是洗牌策略并没有考虑敌手作弊的情况, 敌手可能伪造节点 ID, 也可能积累历史 ID; 通过这些手段, 她总能够获取足够多的 ID. 如何限制敌手伪造 ID, 如何防止敌手累积 ID, 以及如何判定一个 ID 是否合法, 是应用洗牌策略需要解决的问题. 本文的目标是: 克服敌手作弊, 高效应用洗牌策略.

2 相关工作

在 Pastry[2] 设计之初, Castro[6] 就曾经考虑过 DHT 网络的安全性. Castro 的方案试图通过提高用户的进入

收稿日期: 2007-05-24; 修回日期: 2008-05-28
基金项目: 国家发展改革委员会专项基金; 中国下一代互联网示范工程(CNGI) 子课题"基于 IPV6 的 P2P 弹性重叠网络智能节点的研制" (No. CNGI-04-12-1D)

门槛来限制敌手获得 ID, 具体做法是付费注册或者采用 CA 服务器认证——但这损害了 P2P 的开放性. Dinger[7]提出一种节点自注册的方案, 该方案对使用一个 IPv4 或一段 IPv6 地址进入网络的节点数量进行限制. 这种方案限制了 DHT 的应用范围.

在已有的应对方案中, 有两种比较有特点, 而又富于创意:

其一, Condie[8] 等采用随机数服务器方案, 随机服务器每隔一段时间生成一个随机数, 节点的 ID 是加入时刻随机数的函数. 所有节点都有一个生存期, 每隔一段时间必须重新加入网络. 由于节点 ID 不能预测但可被验证, 敌手无法伪造和累积 ID. 但该方案的问题是: 网络动态性比较高, 不太适用于写硬盘之类的大文件应用.

其二, Scheideler[5]提出了一种洗牌策略. 新加入节点并不是简单的插入到网络中的某个位置上, 而是经过几轮节点替换后, 被替换节点插入到一个新的位置. 只要攻击节点数目不太大, 就能够以高概率保证在长度为 $O(\log n)$ 的节点序列中诚实节点占多数. 这种方案从理论上解决了 Sybil 攻击问题; 但是在算法应用中, 如何维护这种替换规则、防止敌方作弊却成为新的问题.

Fiat[9]在洗牌策略的基础上引入 Byzantine 协议来处理敌手作弊问题. 应该说 Fiat 的解决方案是完整的, 他既考虑了节点加入攻击问题, 又考虑了对节点放置的对象的攻击问题. 但是, 这个方案的代价是高昂的, 加入消息复杂度为 $O(\log^3 n)$, 而对象查询复杂度为 $O(\log^2 n)$. 在网络规模比较大时, 这些复杂度使算法难以部署.

Fireflies[10] 在 Chord 单环的基础上, 引入多维环, 每个节点同时属于 $2t+1$ 个环; 节点的 $2t+1$ 维环前驱构成了该节点的仲裁集; 同时引入指控/辩护机制, 对节点行为指控进行仲裁.

3　系统模型

这一节先介绍洗牌策略, 然后讨论防止敌手作弊的基本思路, 并定义一些术语.

3.1　洗牌策略

洗牌策略把节点区分为诚实节点于攻击节点, 所有节点排列在一个圆环上; 同时假设所有攻击节点由一个敌手控制. 敌手可以选择加入一个攻击节点, 或者退出一个在线节点.

洗牌策略的核心是 k 轮旋转加入规则: 新节点随机选择圆环上一个位置, 它替换掉该位置上的原有节点, 这称为第 1 轮替换; 被替换节点重新选择一个位置, 再替换该位置上的节点, 这是第 2 轮替换; 第 k 轮, 被替换节点直接插入到一个随机位置上. Scheideler 证明: 当 $k=3$ 时, 能够以高概率保证圆环上任意一个长为 $O(\log n)$ 的节点序列中诚实节点占多数.

取 $k=3$ 的洗牌策略有一个前提条件: 敌手能够控制的节点数不能超过总数的 1/4. 本文认为, 当网络规模比较大时, 该条件容易满足.

3.2　基本思路

洗牌策略通过 k 轮替换规则把敌我双方节点充分混合, 避免了敌手在局部获得优势. 但是敌手完全可以不按照洗牌策略行事, 她可以通过伪造节点 ID, 或者积累历史 ID 来获得足够多的 ID.

如何避免敌手伪造 ID 是我们需要考虑的第一个问题. 其次, 敌手可能使用历史 ID, 通过拓扑维护算法加入网络, 如何判定一个 ID 已经过期是需要考虑的第二个问题. 这两个问题是相关的, 首先必须防止敌手生成任意多的 ID, 否则将无法判定节点 ID 是否合法.

为了解决第一个问题, 防止敌手伪造 ID, 本文引入一种特殊的节点——受信节点, 它们组成一个分布式认证系统. 新节点加入需通过受信节点认证, 这就防止了敌手伪造 ID. 敌手虽然不能伪造 ID, 但是敌手仍然可以累积历史 ID, 然后通过类似于 Chord 网络中的 Stabilize[11] 网络拓扑维护协议加入网络. 因此, 需要某种方法来判定一个 ID 是否过期.

注意到在 k 轮旋转加入过程中, 替换节点与被替换节点之间不可能存在在线节点, 称二者之间的间距为替换区间. 如果一个历史 ID 落在了一个替换区间上, 说明该 ID 在这个 k 轮旋转加入发生时并不在线, 由此可以判定该 ID 已经过期. 当网络中的节点记录的替换区间足以覆盖整个网络地址空间时, 就能够判定任意一个 ID 是否过期.

本文以 Chord 网络作为模型. 限于篇幅, Chord 网络的基本算法不再介绍.

3.3　术语定义

假设每个节点都有公钥(K^+)与私钥(K^-), 节点间使用某种密钥交换协议(IKE)交换公钥. 定义加密操作 $E(x,k)$, 表示明文 x 由键 k 加密; 定义 $\|$ 表示字符串连接操作.

新节点 a 需通过一个受信节点 u 认证才能加入网络, 它从 u 节点获取签名和票据. 定义签名为 $sig(a)=E(K^+(a)\|T, K^-(u))$, 表示 a 的签名由 a 的公钥与时戳 T 经过 u 的私钥加密而成. 节点 ID 定义为 $id(a)=hash(sig(a))$, 其中 hash 为某种哈希运算. 票据记录了一次洗牌策略的节点加入过程, 定义为: $ticket=E((K^+(a), sig(b), sig(c))\|(sig(a), sig(b'), sig(c')), K^-(u))$ 其中 a 是新加入节点, b 和 c 是第 1 轮和第 2 轮被替换节点, b' 和 c' 是这两个被替换节点的新位置.

4　算法描述

这一节给出节点加入算法、ID 合法性判定算法.

4.1 节点加入算法

算法 1 实现了洗牌策略, 本文规定被替换节点是替换节点的直接后继节点.

算法 1 节点加入算法

```
// a is a new joining node, u is a trusted node.
a to u: Join                                                    (1)
u: sig(a) = E(K+(a) || T, K-(u)),
    A = hash(sig(a)) and b = suc(A);                            (2)
u to b: Signature;                                              (3+)
b to u: (sig(b), v);                                            (4)
u to v: Substitute, sig(b);                                     (5)
v to u: sig(b') = E(K+(b) || T, K-(v)),
    B = hash(sig(b')) and c = suc(B);                           (6)
u to c: Signature;                                              (7)
c to u: (sig(c), w);                                            (8)
u to w: Substitute sig(c);                                      (9)
w to u: sig(c') = E(K+(c) || T, K-(w)),
    C = hash(sig(c'))                                           (10)
u to a, b, c: Substitute, ticket                                (11)
```

节点加入过程如图 1. 第(1)步, a 节点向 u 发送加入请求; 第(2)步, u 计算 a 的签名和 ID(位置为 A), 然后查找 A 的后继节点 b; 第(3)步, u 查询 b 的签名; 第(4)步, b 向 u 发送签名及其认证节点 v; 第(5)步, u 向 v 发送替换 b 的请求; 第(6)步, v 重新计算 b 的签名和 ID(位置为 B), 并找到 B 的后继节点 c; 第(7)步, u 查询 c 的签名; 第(8)步, c 向 u 应答它的签名和认证节点; 第(9)步, u 向 c 的认证节点 w 发送替换请求; 第(10)步, w 计算 c 节点的新签名和 ID 位置; 第(11)步, u 向 a、b 和 c 节点发送替换命令和替换票据.

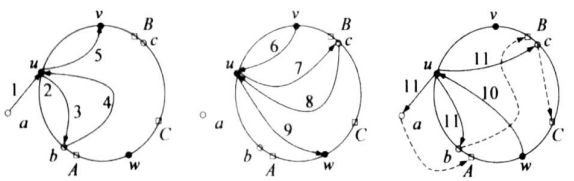

图1 节点加入过程

算法2 ID 合法性判定函数

```
idIsLegal(sig(a))
    // ticket1 是票夹中最新的票据;
    if ticket 1. time < sig(a). time                            (1)
        return true;                                            (2)
    // ticket2 是票夹中最老的票据;
    if ticket 2. time > sig(a). time                            (3)
        return false;                                           (4)
    if ∃ ticket ∈ wallet(u):
      ticket. time > sig(a). time ∧ (id(a) ∈ ticket)            (5)
        return false;                                           (6)
    else                                                        (7)
        return true;                                            (8)
```

算法 1 尽量避免了受信节点与普通节点的交互. 如果一个普通节点拒绝提供它的签名, 则受信节点按照该节点失效处理, 以下一个后继节点作为被替换节点. 根据后文的算法 2, 由于新加入节点的票据中的替换区间覆盖了沉默节点, 沉默节点将被认为是一个非法节点.

4.2 ID 合法性判定算法

节点 ID 由签名计算而来, 因此 ID 可以验证但无法伪造, 但是敌手仍然可以通过积累而获得大量 ID. 由于在一次洗牌加入中, 替换区间上不可能有节点存在, 所以可以借助替换区间来判定一个 ID 是否过期. 由于票据完整记录了一次洗牌过程, 所以在线节点只需保存其邻居范围内的时戳最新票据, 就能够对一个邻近 ID 做出判断.

称节点上保存票据的数据结构为票夹(wallet). 票夹的管理有两个要求: 票夹必须保存最新票据, 同时保存的票价能够覆盖节点的邻居范围. 票夹的大小在后文讨论.

算法 2 是 ID 合法性判定函数. 如果一个 ID 的签名比票夹中的所有票据都新, 则接受它为合法; 如果该票据不是最新的, 则检查票夹中是否有更新的票据覆盖了该节点, 如果节点被覆盖, 则说明该节点是非法的, 否则接受为合法节点.

有了算法 2, 就可以在网络拓扑维护协议中判定一个节点是否过期, 从而杜绝了敌手通过累积 *ID* 的企图.

5 理论分析

这一节首先分析算法 1 的消息复杂度, 然后讨论票夹的大小问题.

定理 **1** 算法 1 的消息复杂度为 $O(4\log n)$.

证明: Chord 网络中查找一个节点的消息复杂度为 $O(\log n)$, 而算法 1 需要访问 4 个节点: b、c、v 和 w, 所以算法 1 的消息复杂度为 $O(4\log n)$. □

对比 Fiat[9] 的方案, 本文提出算法在消息复杂度上的优势是明显的.

下面讨论票夹容量的问题. 算法 2 能够正确判定一个邻近节点是否合法的前提是: 票夹中的票据能够完整覆盖邻居范围. 如果票夹中需要保存的票据过多, 则本文提出的方案将难以应用. 因此票夹容量问题关系到本文提出算法的可行性.

设地址空间容量为 K, 在线节点数为 n, 新节点以速率 λ 加入网络, 在线节点以速率 μ 退出网络. 由于采用洗牌加入策略, 节点可划分为三类: 替换节点、插入节点和退出节点. 插入节点为最后一轮($k=3$ 轮)被插入到网络中的节点.

本文采用 Markov 过程对问题建模. 设网络在初始

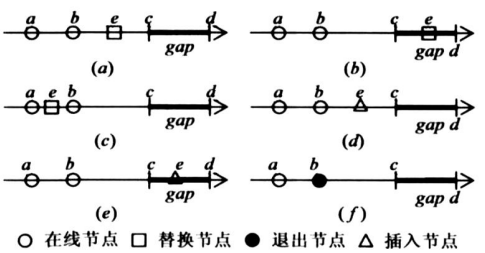

图2　节点替换、插入与退出对间隙的影响

状态时, 有 n 个节点, 地址空间的其余部分未被替换区间覆盖. 随着节点的不断加入与退出, 圆环上未被覆盖的区域被划分为许多小区间, 称为间隙(gap). 本文选择间隙作为Markov 状态, 节点替换、插入与退出事件都可能会对间隙产生影响, 如图 2 所示.

图 2 中, (c, d) 表示一个间隙, a、b 是间隙的两个前驱节点, c 是间隙的左端点, d 是间隙的右端点. 图 2 (a) 当 e 为一个替换节点并且 $e\in(b, c)$ 时, 间隙将被覆盖; 图2(b) 当 e 为一个替换节点并且 $e\in(c, d)$ 时, 部分 $[e, d)$ 将被覆盖, 间隙将减小为 (c, e); 图 2(c) 当 e 为一个替换节点并且 $e\in(a, b)$, b 被替换, 间隙维持不变. 图 2(d) 当 e 为一个插入节点并且 $e\in(b, c)$ 时, 间隙大小不变; 图 2(e) 当 e 为一个插入节点并且 $e\in(c, d)$ 时, 间隙将被分裂为两个: (c, e) 和 (e, d). 图 2(f) 当 b 为退出节点, a 就成为间隙的直接前驱.

用二元组(l, m) 来表征一个间隙: l 是间隙左端点到直接前驱的距离, m 是间隙的长度. 设 $G(l, m)$ 为系统中(l, m) 间隙的个数, 其中 $0\leqslant l\leqslant K-n-1\wedge 2\leqslant m\leqslant K-n$. 并且定义一个特殊的状态$(*, 0)$, 表示间隙被覆盖.

计算从 t_0 时刻到 t 的(l, m) 间隙的似然个数:

$$\begin{aligned}G(l, m, t)=\ & G(l, m, t_0)\\ & -\sum_{(l', m')\neq(l, m)} G(l, m, t_0)\,Q(l', m', t\mid l, m, t_0)\\ & +\sum_{(l', m')\neq(l, m)} G(l', m', t_0)\,Q(l, m, t_0\mid l', m', t_0)\end{aligned}\tag{1}$$

式(1)中, t 时刻(l, m) 间隙个数等于 t_0 时刻间隙 (l, m) 个数减去从(l, m) 状态转移到其它状态的个数, 加上从其它状态转移到(l, m) 状态的个数. $Q(l', m', t\mid l, m, t_0)$ 和 $Q(l, m, t_0\mid l', m', t_0)$ 表示状态转移概率.

根据图 2, 计算式(1)中地转移概率, 带入得到式(2)和式(3). 限于篇幅, 转移概率的计算这里省略. 在条件 $l\leqslant l\leqslant K-n-1\wedge 2\leqslant m\leqslant K-n$ 下, 间隙满足动态方程(2); 在条件 $l=0\wedge 2\leqslant m\leqslant K-n$ 下, 间隙满足动态方程(3). 令 $\Delta t=t-t_0$.

$$\begin{aligned}G(l, m, t)=\ & G(l, m, t_0)-G(l, m, t_0)\\ & \cdot[\lambda\Delta t(\frac{2l}{K-n}+\frac{2(m-1)}{K-n}+\frac{1}{n})+\frac{\mu\Delta t}{n}]\\ & +\frac{\lambda\Delta t}{K-n}(\sum_{x=m+1}^{K-n} G(l, x, t_0)\\ & +\sum_{x=1}^{l} G(l-x, m, t_0)\sum_{y=x+2}^{K-n} Q(y))\\ & +\frac{\lambda\Delta t}{K-n}(\sum_{x=l+1}^{K-n-1} G(x, m, t_0)+\sum_{x=m+1}^{K-n} G(l, x, t_0))\\ & +\frac{\mu\Delta t}{n}\sum_{x=1}^{l} G(l-x, m, t_0)\,Q(x)\end{aligned}$$

其中 $1\leqslant l\leqslant K-n-1\wedge 2\leqslant m\leqslant K-n$ (2)

$$\begin{aligned}G(0, m, t)=\ & G(0, m, t_0)\\ & -G(0, m, t_0)[\lambda\Delta t(\frac{2(m-1)}{K-n}+\frac{1}{n})+\frac{\mu\Delta t}{n}]\\ & +\frac{\lambda\Delta t}{K-n}(\sum_{x=m+1}^{K-n} G(0, x, t_0)\\ & +\frac{\lambda\Delta t}{K-n}(\sum_{x=l+1}^{K-n-1} G(x, m, t_0)+\sum_{x=m+1}^{K-n} G(0, x, t_0))\\ & +\sum_{y=0}^{K-n-1}\sum_{x=m+1}^{K-n} G(y, x, t_0))\end{aligned}$$

其中$(l=0\wedge 2\leqslant m\leqslant K-n)$ (3)

将方程(2)和(3)转化为偏微分方程:

$$\begin{aligned}\frac{\partial G(l, m, t)}{\partial t}=\ & -G(l, m, t)[\lambda(\frac{2l}{K-n}+\frac{2(m-1)}{K-n}+\frac{1}{n})+\frac{\mu}{n}]\\ & +\frac{\lambda}{K-n}(\sum_{x=m+1}^{K-n} G(l, x, t)\\ & +\sum_{x=1}^{l} G(l-x, m, t)\sum_{y=x+2}^{K-n} Q(y))\\ & +\frac{\lambda}{K-n}(\sum_{x=l+1}^{K-n-1} G(x, m, t)+\sum_{x=m+1}^{K-n} G(l, x, t))\\ & +\frac{\mu}{n}\sum_{x=1}^{l} G(l-x, m, t)\,Q(x)\end{aligned}$$

其中$(1\leqslant l\leqslant K-n-1\wedge 2\leqslant m\leqslant K-n)$ (4)

$$\begin{aligned}\frac{\partial G(l, m, t)}{\partial t}=\ & -G(0, m, t)[\lambda(\frac{2(m-1)}{K-n}+\frac{1}{n})+\frac{\mu}{n}]\\ & +\frac{\lambda}{K-n}\sum_{x=m+1}^{K-n} G(0, x, t)\\ & +\frac{\lambda}{K-n}+\sum_{x=l+1}^{K-n-1} G(x, m, t)+\sum_{x=m+1}^{K-n} G(0, x, t))\\ & +\sum_{y=0}^{K-n-1}\sum_{x=m+1}^{K-n} G(y, x, t))\end{aligned}$$

其中$(l=0\wedge 2\leqslant m\leqslant K-n)$ (5)

方程(4)和(5)是一组关于 l 和 m 的偏微分方程组, 求解整个方程组即可得到(l, m) 状态的似然个数 $G(l, m, t)$. 但是直接求解这个偏微分方程组比较困难, 下面对这两个方程进行稳态和动态分析.

定理 2　不包括特殊状态$(*, 0)$, 方程(5)和(6)在稳态时, (l, m) 间隙个数为零.

证明: 方程(4)和(5)中, 右边各项为负表示一个间隙从(l, m) 状态迁出, 记为 $(l, m)\rightarrow(*, m^-)$; 右边为正的项表示从某个间隙从其它状态迁移到$(l, m)$, 记为

$(*, m^+) \rightarrow (l, m)$.

考察$(l, m) \rightarrow (*, m^-)$, 必然有 $m \geqslant m^-$; 而对于$(*, m^+) \rightarrow (l, m)$, 则必然有 $m^+ \geqslant m$. 所以对于任意时刻 t, 网络中的间隙必然会随着节点加入和退出事件的不断发生, 而转化为较小的间隙. 而当一个间隙转化为$(*, 0)$状态后, 就不会从该状态脱离, 所以$(*, 0)$是一个吸收态, 最终网络中的间隙全部消失. 因而(l, m)的稳态间隙个数为零. □

定理 2 说明了由于存在吸收状态$(*, 0)$, 网络中的间隙最终会被完全覆盖.

定理 **3** 网络中最长间隙的数量近似以负指数形式衰减.

证明: 设在 t 时刻网络中最长间隙是m, 设此时网络中最长间隙集合为$(*, m)$, 对于某个 $l \geqslant 0$, 此时由于不存在比 m 更长的间隙, 根据式(4)得到:

$$\begin{aligned}\frac{\partial G(l, m, t)}{\partial t} = & -G(l, m, t)\left[\lambda\left(\frac{2l}{K-n} + \frac{2(m-1)}{K-n} + \frac{1}{n}\right) + \frac{\mu}{n}\right] \\ & + \frac{\lambda}{K-n}\sum_{x=1}^{l} G(l-x, m, t)\sum_{y=x+2}^{K-n} Q(y) \\ & + \frac{\lambda}{K-n}\sum_{x=l+1}^{K-n-1} G(x, m, t) \\ & + \frac{\mu}{n}\sum_{x=1}^{l} G(l-x, m, t)\, Q(x) \\ & \text{其中}(0 \leqslant l \leqslant K-n-1 \wedge 2 \leqslant m \leqslant K-n) \end{aligned} \tag{6}$$

由于式(6)右端正项是$(*, m)$状态之间的转换, 定义 $G^*(m) = \sum_l G(l, m)$, $G^*(m)$ 是网络中长度为 m 的间隙个数之和; 因此式(6)的右端正项不会对 $G^*(m)$ 产生影响, 由此得到:

$$\begin{aligned}\frac{\partial G^*(m, t)}{\partial t} = & -\sum_{l=0}^{K-n-1} G(l, m, t)\left[\lambda\left(\frac{2l}{K-n} + \frac{2(m-1)}{K-n} + \frac{1}{n}\right) + \frac{\mu}{n}\right] \end{aligned} \tag{7}$$

对式(7)计算近似解:

$$\begin{aligned}\frac{\partial G^*(m, t)}{\partial t} = & -G^*(m, t)\left[\lambda\left(\frac{2\bar{l}}{K-n} + \frac{2(m-1)}{K-n} + \frac{1}{n}\right) + \frac{\mu}{n}\right] \end{aligned} \tag{8}$$

$\bar{l}$ 表示$(*, m)$中 l 的均值, 由于 m 已确定, 式(8)是一个常微分方程, 求解得到:

$$G^*(m, t) \approx G^*(m, t_0)\exp\left(-2\lambda\Delta t\,\frac{\bar{l}+m-1}{K-n} - \frac{\lambda+\mu}{n}\Delta t\right) \tag{9}$$

式(9)中 $G^*(m, t_0)$ 是初始值, 这是一个负指数函数, 所以命题得证. □

根据定理 2 和定理 3, 网络中最长间隙以指数形式快速衰减到$(*, 0)$, 这说明了采用票据来判定 ID 合法性的可行性. 下面对每个节点需要保存的票据数做一个估计.

令 $\lambda = \mu$, 即节点加入速率与退出速率相等, 网络处于平衡状态. $\lambda\Delta t = p$, p 表示加入节点或退出节点个数. p/n 表示平均每个节点需要保存的票据数. 带入式(9), 得到:

$$G^*(m, t) \approx G^*(m, t_0)\exp\left(-2p\,\frac{\bar{l}+m-1}{K-n}\right)\cdot\exp\left(-\frac{2p}{n}\right) \tag{10}$$

假设系统初始时 n 个在线节点之间的距离相等, 都是 K/n; 此时所有间隙 $l = 0$; 并且间隙个数为 n, 即 $G^*(m, t_0) = n$; 同时假设最长间隙为 $m = K/n$. 由于 $K \gg n$, 所以可用得到近似值 $p(m-1)/(K-n) \approx (p/n)\dfrac{m}{K/n-1} \approx p/n$, 带入式(10), 得到系统初始时的最长间隙数的近似值:

$$G^*(m, t) \approx ne^{-4\frac{p}{n}} \tag{11}$$

当间隙数衰减到 1, $p = \dfrac{1}{4}n\ln(n)$. 式(10)由于做了过多假设, 并不能用来估算网络中间隙被完全覆盖需要的票据数. 但是通过式(11)可以推测: 间隙(l, m)的个数会近似地以负指数形式衰减; 并且式(11)暗示着当网络规模增大时, 每个节点需要保存的票据数与网络规模的对数成正比. 需要说明的是, 由于无法完整求解式(4)和(5), 所以本文只能利用仿真实验来进行检验.

6 仿真实验

由于难以求得式(4)和(5)的解析解, 所以上一节对这两个方程进行了定性分析. 定性分析指出间隙能够快速被票据覆盖, 同时对每个节点保存票据数(p/n)的规律进行了推测. 这一节通过实验来检验这些分析与推测.

测试方法 选择 32 比特作为 ID 空间; 以 C 语言的随机函数生成种子, 经过 SHA1 运算后生成摘要, 然后以摘要作为 ID 生成器. 每次测试开始之前, 先从 ID 生成器中生成 n 个节点, 加入到网络中. 实验开始后, 不停生成新节点加入网络, 同时选择在线节点退出网络, 直到网络中的间隙被彻底覆盖.

实验 检查平均每个节点需要保存的票据数(p/n). p 是 ID 空间被票据覆盖时地加入节点个数(同时也是退出节点个数); n 是网络规模, 即网络中的在线节点数. 网络规模从 100 增加到 60000 个节点, 每个网络规模测试 100 次.

实验结果如图 3 所示. (1)所有实验数据得到的 p/n 均小于 35, 说明了间隙能够快速被覆盖, 每个节点上保存的票据数不多, 算法可行. (2)每个网络规模下的股价图不对称, p/n 的重心(均值)偏向低位. (3)从总体趋势上来看, 随着网络规模的增加, p/n 在缓慢的增加, 网

络规模为 100 时, p/n 均值为 6 左右; $n=60000$ 时, p/n 均值大约为 24. (4) 可以观察到 p/n 的均值基本是线性增长; 使用线性拟合, 得到:

$$p/n = 2.7927\lg(n) - 7.3494 \quad (12)$$

在算法的实际部署中, 可以采用式(12)来预测票夹大小.

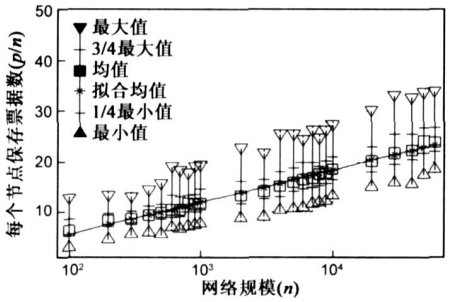

图3　每个节点需要保存的票据数仿真数据

7　结论

如何在开放的网络环境中, 高效地抵御 Sybil 攻击, 是一个困难的问题. 利用洗牌策略作为节点加入算法, 使敌我双方节点充分混合, 就能够在动态的网络中以高概率抵御 Sybil 攻击. 本文的主要目的是高效的应用洗牌策略来抵御 Sybil 攻击.

在洗牌策略的应用中, 关键问题是如何防止敌手作弊. 本文采用两个策略来杜绝敌手作弊: (1) 使用受信节点组成分布式认证系统, 对节点发布签名, 节点 id 从签名计算而来. 节点 ID 可以验算, 但无法伪造; (2) 利用票据记录洗牌加入过程, 节点通过连续累计的历史票据来验证一个邻近的 id 是否是过期的. 理论分析表明需要保存的历史票据不会太多, 同时仿真实验数据也表明平均每个节点需要保存的历史票据与网络规模的对数成正比. 这些说明了算法应用的可行性.

参考文献:

[1] Stoica I, et al. Chord: a scalable peer-to-peer lookup protocol for internet applications[J]. Networking, IEEE/ACM Transactions, 2003, 11(1): 17-32.

[2] Rowstron, A I T, P Druschel. Pastry: scalable, decentralized object location, and routing for large-scale peer-to-peer systems [A]. In Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg[C]. Springer Verlag, 2001. 329-350.

[3] Douceur, J R. The Sybil attack[A]. In Revised Papers from the First International Workshop on Peer-to-Peer Systems [C]. Springer Verlag, 2002. 2429: 251-260.

[4] Singh, A, et al. Defending against eclipse attacks on overlay networks[A]. In Proceedings of the 11th Workshop on ACM SIGOPS European Workshop: Beyond the PC[C]. Leuven, Belgium: ACM Press, 2004.

[5] Scheideler, C. How to spread adversarial nodes? rotate! [A]. In Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing [C]. Baltimore, MD, USA: ACM Press, 2005. 704-713.

[6] Castro M., et al. Secure routing for structured peer-to-peer overlay networks[A]. In Proceedings of the 5th Symposium on Operating Systems Design and Implementation[C]. Boston, Massachusetts: ACM Press, 2002. 299-314.

[7] Dinger J, H Hartenstein. Defending the Sybil attack in P2P networks: taxonomy, challenges, and a proposal for self-registration [A]. In Proceedings of the First International Conference on Availability, Reliability and Security (ARES'06)[C]. IEEE Computer Society, 2006. 756-763.

[8] Condie T, et al. Induced churn as shelter from routing table poisoning[A]. In Proceedings of the 13th Annual Network and Distributed System Security Symposium (NDSS)[C]. San Diego, California, USA: The Internet Society, 2006.

[9] Fiat A, J Saia, M Young. Making chord robust to byzantine attacks[A]. Lecture Notes in Computer Science (LNCS 3669)[C]. Springer Berlin / Heidelberg, 2005. 803-814.

[10] Johansen H, A Allavena, R v Renesse. Fireflies: scalable support for intrusion-tolerant networks overlay[A]. In Proceedings of the 2006 EuroSys conference[C]. ACM European Chapter, 2006. 3-13.

作者简介:

聂晓文　男, 1972 年生于四川绵阳, 电子科技大学博士生, 研究领域为 P2P 计算、分布式计算. E-mail: niexiaowen@uestc.edu.cn

卢显良　男, 1943 年生于河南洛阳, 电子科技大学教授、博导. 主要从事高级操作系统、计算机网络及分布式系统等研究.
E-mail: xlu@uestc.edu.cn