低维护开销的小世界 P2P 网络*)

王向辉 张国印 张 闯
(哈尔滨工程大学计算机科学与技术学院 哈尔滨 150001)

摘 要 为降低结构化 P2P 网络的维护开销,提高路由和查询的效率,提出了具有低维护开销的小世界 P2P 网络(LMCS),并描述了网络的创建和维护方法。小世界特征使结构化 P2P 网络具有较高的路由和查询效率,同时利用成簇机制和扩展 COU 策略,有效地降低网络的维护开销。通过模拟仿真,LMCS 呈现明显的小世界网络特征。与 Chord 相比,LMCS 具有更低的维护开销和更高的查询效率。

关键词 小世界,维护开销,P2P,分布式哈希表

Low Maintenance Cost Small-world P2P Networks

WANG Xiang-hui ZHANG Guo-yin ZHANG Chuang
(College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China)

Abstract In order to decrease the maintenance cost of structure P2P network, and increase the efficiency of the route and quire, we proposed a low maintenance cost small-world P2P networks (LMCS), described the methods of network of the creating and maintenance. Small-world characteristic makes the better efficiency of the route and quire, and the clustering mechanism and extension of the COU strategy were used to decrease the maintenance cost. By Simulation, LMCS obviously presents the small-world characteristic. In contrast to Chord, LMCS has the lower maintenance cost and the higher query efficiency.

Keywords Small-world, Maintenance cost, P2P, DHT

1 引言

分布式哈希表(DHT)是构造结构化 P2P 网络的主要方法,一般将节点和对象(资源)映射到唯一的键值空间中,因此节点和对象的查找就成为了路由问题。同时,每个节点维持一定数量的其它节点的信息,用以将消息路由加速到目标节点。目前主流的结构化 P2P 网络,路由和消息查询一般需要 O(log$N$)步,N 为网络中的节点数量。

为进一步提高结构化 P2P 网络路由和查询效率,研究人员利用小世界原理改进 P2P 网络。小世界理论源于社会科学领域,目前已在物理学、计算机科学和数学等领域展开深入研究[2]。研究人员发现在社交网络、生物网络、通讯网络和 P2P 网络[3]中广泛地存在小世界现象,而且利用小世界原理构造 P2P 网络,能够进一步提高网络的路由和查询效率[4]。

尽管 DHT 的路由和查询效率较高,但现存的 P2P 系统多是采用非结构化体系设计的 P2P 网络,如 Napster,eMule 和 KaZaA,其主要原因是 DHT 对网络的节点波动非常敏感[1],导致网络维护开销巨大。因为 DHT 的高效路由要依靠于每个节点维持正确、一致的路由表,保持 P2P 网络的结构特性,但 DHT 网络一般都没有高效的网络维护方法。例如 Chord 为了维护路由表的正确性,每个节点在一个周期内需要发送 O(log^2N)个消息。网络的波动越大,刷新路由表的周期越短,网络的维护开销越大。

本文利用小世界原理,基于 DHT 构造了低维护开销的结构化 P2P 网络(LMCS)。LMCS 利用成簇机制和扩展 COU 策略降低网络的维护开销,利用簇分裂和簇合并方式实现节点的负载均衡,利用小世界特性使网络有较高的路由和查询效率。

2 相关工作

Migram[5]首先发现了小世界现象,描述了社会学领域的"六度分隔"理论,即"小世界"理论,该理论认为平均通过 6 个熟人就可以把世界上任意的两个人联系起来。Kleinberg[6]提出了用于分析小世界特性的理论框架;文献[3]提出了具有小世界特征的非结构化 P2P 网络;文献[4]在非结构化 P2P 网络上构建了具有小世界特征的结构化网络。

在降低结构化 P2P 网络维护开销上,文献[7]为 Pastry 提出自调整策略,节点根据网络中的节点总数量和节点失败率,动态调节、刷新路由表入口地址的时间间隔,从而保证路由消息的可靠性和低网络维护开销。文献[8]提出 COU 策略,利用查询消息的反馈信息更新路由表的入口地址。文献[9]在 Chord 上实现了 COU 策略,并利用虚拟无限空间缓存其它节点地址,以提高查询效率。

3 小世界理论

具有小世界特征的网络可以认为是一个连通图,图中的任意两点都可以通过 6 个边连接在一起,即任意两点间的平均最短距离应接近 6。

小世界网络的两个重要特性是特征路径长度和聚合系数。为了说明这两个参数,用 $g=(V,E)$ 表示小世界网络的

*)黑龙江省自然科学基金(F2004-06)资助。王向辉 博士研究生,讲师,研究方向为 P2P 网络;张国印 博士,教授,博士生导师,研究方向为网络信息安全和嵌入式系统;张 闯 硕士研究生,研究方向为 P2P 网络和分布式计算。

连通图，g 中有 N 个节点，即 $|g|=N$；$D(i,j)$ 表示节点 i 和结点 j 之间的最短路径长度，其中 $i,j\in V$。

定义 1(特征路径长度 $L(g)$) g 中所有节点对的最短路径长度的平均值，定义为网络的特征路径长度：

$$L(g)=\frac{1}{\binom{N}{2}}\sum_{i,j\in V}D(i,j) \tag{1}$$

定义 2(节点聚合系数 C_v) 任意节点 $v,v\in V$，v 的出度为 k_v，相邻节点集合为 $\Gamma_v=\{u:D(u,v)=1\}$，用 v 相邻节点之间实际存在的边数除以最多可能存在的边数，就得节点 v 的聚合系数：

$$C_v=|E(\Gamma_v)|/\binom{K_v}{2} \tag{2}$$

定义 3(聚合系数 $C(g)$) 所有节点的节点聚合系数的均值，定义为网络的聚合系数：

$$C(g)=\frac{1}{N}\sum_{v\in V}C_v \tag{3}$$

规则网络的特征路径长度和聚合系数都较大，而随机网络这两个值都较小。小世界网络的特征路径长度较小，但聚合系数却比较大。小世界特征保证在 P2P 网络中，消息经过较少的节点就可以路由到目标节点。

4 LMCS 网络

4.1 网络构成

LMCS 网络使用相容哈希函数 h(例如 MD5,SHA-1)，将节点映射到唯一的键值空间中，并可以根据节点的键值确定节点在逻辑空间中的位置。同样，节点可以使用相同的相容哈希函数 h 将对象(或资源)加入到 LMCS 网络中，每个对象 o 有唯一的一个键值 $h(o)$。用 $N(o)$ 表示网络中所有键值小于或等于 $h(o)$ 的节点集合，在此集合中具有最大键值的节点保存并维护这个对象 o。

LMCS 网络使用环形逻辑空间表示小世界网络，并将逻辑空间划分为多个簇，每簇有一个簇头节点，其它节点为簇内节点。簇头节点负责簇间的消息路由和维护簇内节点，簇内节点负责簇内消息路由。节点 i 是簇头节点，$C(i)$ 表示以节点 i 为簇头的簇的节点集合，$|C(i)|$ 表示簇内节点数量，节点 j 是簇 $C(i)$ 内的节点，则 $j\in C(i)$。同一个簇内簇头节点是簇内节点中键值最小的节点，因此 $j>i$。例如图 1 中簇 $C(0)$ 包含 4 个簇内节点{7,15,25,29}，簇头节点 0 是簇内键值最小的节点。

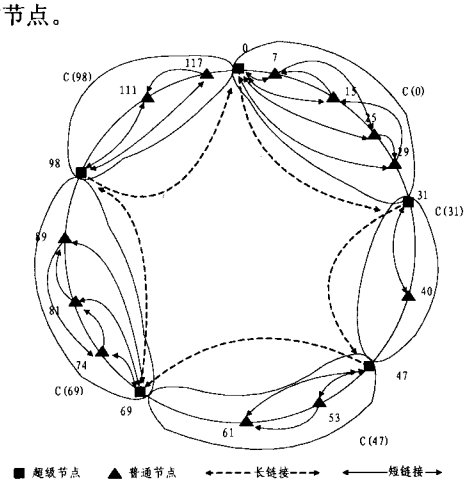

图 1 LMCS 网络

连接不同簇头之间的链接为长链接，长链接使消息可以在不同的簇之间传递；连接簇内节点之间的链接为短链接，短链接使消息可在簇内传递。每个簇头节点维持 k 个长链接，簇内节点维持 m 个短链接，同时每个簇内节点和本簇的簇头节点之间一定存在一个短链接。如图 1 中，长链接数量 $k=2$，短链接数量 $m=2$。

4.2 消息路由

定义 4(簇域 $R(i)$) 是 LMCS 网络上的一段逻辑空间，表示簇 $C(i)$ 所覆盖的逻辑空间范围。簇域 $R(i)$ 的起点是簇 $C(i)$ 的簇头节点 i，簇域 $R(i)$ 的终点是 $C(i)$ 在环形空间顺时针方向的下一个簇头节点 j，因此 $R(i)=[i,j)$。如果 $h(o)\in R(i)$，则 $h(o)$ 一定由簇 $C(i)$ 的簇内节点进行保存和维护。例如图 1 中，簇 $C(0)$ 的簇域为[0,31)，簇 $C(31)$ 的簇域为[31,47)。

消息路由可分为两种模式，簇内路由和簇间路由，节点根据簇域选择消息路由模式。节点 i 向节点 j 发送消息，$i\in C(x)$，如果 $j\in R(x)$，则进行簇内路由；如果 $j\notin R(x)$，则进行簇间路由。

簇内路由是消息的发送节点和目标节点在同一个簇内的消息路由模式，节点根据“同向相近”原则转发消息，因此消息在节点间是双向、多跳传递的。同时，如果转发节点无法找到“相近”的目标节点，则将消息发送给簇头节点。例如图 1 节点 29 向节点 7 发送消息，消息的转发路径为〈29,15,7〉；而节点 25 向节点 7 发送消息，消息的转发路径为〈25,0,7〉。簇内路由尽量避免利用簇头节点转发消息，有利于减轻簇头节点的负载压力。

簇间路由是消息的发送节点和目标节点在不同簇的消息路由模式。消息首先发送给所在簇的簇头节点，然后由簇头节点在簇间进行转发，到达目标簇的簇头后，再转发给簇内的目标节点。消息在簇间转发时，由起始簇头节点随机选择路由方向(顺时针或逆时针)，消息在簇间按照此方向传递，并且不允许中途转发节点改变此消息的路由方向。例如节点 15 向节点 61 发送消息，起始簇头节点 0 选择顺时针方向作为簇间消息转发方向，则转发路径为〈15,0,31,47,61〉；相反，起始簇头节点 0 选择逆时针方向作为簇间消息转发方向，则转发路径为〈15,0,98,69,47,61〉。消息在簇间随机双向传递，有利于降低 P2P 网络的维护开销。

4.3 节点加入、退出

节点加入时，节点 i 随机选择一个簇头节点 j 作为引导节点，节点 i 向节点 j 发送请求加入 P2P 网络的消息，节点 j 根据 4.2 节的消息路由机制，将此消息转发到所属的簇域 $R(k)$ 中簇头节点 k，$i\in R(k)$，节点 k 向节点 i 发送应答消息，完成节点 i 加入簇 $C(k)$ 的过程。

节点退出时，节点 $i,i\in C(k)$，向簇头节点 k 和存在短链接的簇内节点发送退出消息，完成节点 i 退出簇 $C(k)$ 的过程。

4.4 簇分裂与合并

LMCS 网络最初只有一个簇，所有新节点都加入到这个起始簇内。当该簇的簇内成员到达一定数量时，该簇进行分裂；随着节点退出，部分簇的簇内节点过少，此时对这些簇进行簇合并。簇分裂的目的是避免瓶颈节点的出现，实现负载均衡；簇合并的目的是较少簇的数目，提高路由效率。

为了实现簇分裂和簇合并，首先说明两个系统参数。

- 簇分裂值 Gmax：簇内节点数量的最大值。节点数量大于此值时，进行簇分裂。
- 簇合并值 Gmin：簇内节点数量的最小值。节点数量小于此值时，进行簇合并。

簇间消息需要通过簇头节点进行路由转发，簇内节点数量过多，会导致簇头节点资源耗尽，产生性能瓶颈。因此，当簇内节点数量$|C(i)|>Gmax$时，簇$C(i)$进行簇分裂。

簇$C(i)$的分裂过程分为两个步骤：第一步要选择稳定、强大的节点成为新的簇头节点j。节点的能力可以根据节点的带宽、存储容量和CPU速度等条件进行衡量；节点的稳定性可以根据节点的连续在线时间、平均下载时间或可用比例等进行衡量。为了简单明了，本文采用连续在线时间衡量节点的稳定性。同时还要考虑到节点在簇内逻辑空间的位置，在簇域中心位置的节点优先选择。第二步将新簇头节点j和部分簇内节点$x|x\in C(i),x>j$都分裂出去，分裂出去的节点形成新簇$C(j)$，过程如图2所示。

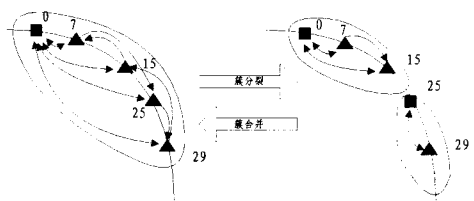

图2　簇分裂、合并过程

P2P网络的演化过程中，部分簇$C(i)$会因为簇内节点的退出而使簇内节点数量$|C(i)|<Gmin$，导致簇合并。簇合并是簇分裂的逆过程。部分簇合并后可能再次导致新簇的分裂，为了避免过程反复，Gmax应远大于2Gmin。

4.5　长链接选择算法

为了使LMCS网络呈现小世界特性，即较小的特征路径长度和较大的聚合系数，每个簇头节点需要选择k个长链接，并且k个长链接分布在逻辑空间的k个分隔区域中。

簇头节点的分隔区域构造方法如下：节点n有k个长链接，键值空间为$[0,2^q-1]$，节点n以$(n+2^{q-1}) \bmod 2^q$为切分点；将整个逻辑空间分隔为A和B两个部分，其中切分点顺时针方向为A区域，依次为$A_1,A_2,A_3,A_4\cdots\cdots,A_r=[(n-2^{q-r}) \bmod 2^q,(n-2^{q-r-1}) \bmod 2^q),1\leqslant r\leqslant\lceil k\rceil/2$；切分点逆时针方向为$B$区域，依次为$B_1,B_2,B_3,B_4\cdots\cdots,B_r=((n+2^{q-r-1}) \bmod 2^q,(n+2^{q-r}) \bmod 2^q],1\leqslant r\leqslant\lceil k\rceil/2$。例如图3是节点0当$q=7$和$k=8$的分隔区域，切分点为64，产生了$A_1,A_2,A_3,A_4$和$B_1,B_2,B_3,B_4$的8个分隔区域。

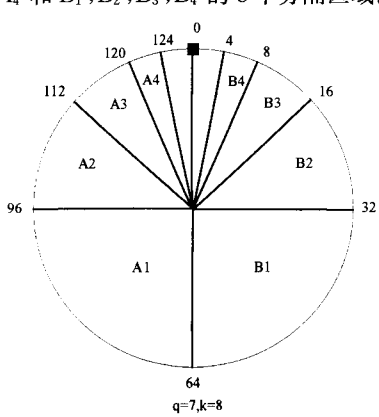

图3　节点0的分隔区域

长链接选择算法的核心思想：(1)每个分隔区域内仅能包含键值在此分隔区域内的一个长链接；(2)当一个分隔区域内没有长链接时，任何一个属于该分隔区域的长链接都可以加入此分隔区域，并设置一个固定的生存周期T，过期的长链接从分隔区域内删除；(3)当一个分隔区域内存在长链接时，新的长链接属于该分隔区域，并且更靠近切分点，则新的长链接可以更换掉原有长链接，并重新设置生存周期T；(4)如果在生存周期内接收到分隔区域原有长链接的消息，则重新设置生存周期T。当任意节点n接收到来自n'的消息时，长链接选择算法的伪代码描述如下：

```
n. Refresh(n′):
(1) if (n′>(n+2^(m-1)) mod 2^m)
(2)   for i=1 to |k/2|
(3)     if (n′>(n-2^(m-i)) mod 2^m)
(4)       if (A_i. Link=null||A_i. Link>n′)
(5)         A_i. Link=n′; A_i. Link. TTL=T
(6)       else if (A_i. Link==n′)
(7)         A_i. Link. TTL=T
(8)else
(9)   for i=1 to |k/2|
(10)    if (n′>(n+2^(m-i-1)) mod 2^m)
(11)      if (B_i. Link=null||B_i. Link<n′)
(12)        B_i. Link==n′;B_i. Link. TTL=T
(13)      else if (B_i. Link==n′)
(14)        B_i. Link. TTL=T
```

4.6　路由表维护

Alima[8]等人提出的“Correction-On-Use”(COU)策略，其核心思想是通过对象加入和搜索的反馈信息，更新路由表中过期的入口地址信息，但对象的加入和搜索速率必须大于节点的加入、离开和失效速率。

COU策略只能通过反馈信息更新路由表，而不能通过正在路由的消息更新路由表，主要是因为其消息是单向转发的。消息向一个方向转发，消息中仅能保存转发反方向的节点信息，而路由表入口节点一般在转发方向上，因此节点无法利用正在路由消息更新自身的路由表，而仅能通过反馈消息收集转发方向上的节点信息来更新路由表。

本文提出的扩展COU策略，不仅可以利用反馈消息更新路由表，而且可以利用正在路由的消息更新路由表。当消息经过每个簇头节点时，簇头节点将自身信息(如IP地址和节点编号)加入到消息中。因为消息在簇间是双向传递的，而且簇头节点的长链接路由表是基于自身在环形空间的位置对称分布的，所以节点路由表的所有入口地址均有机会得到刷新。

为了维持LMCS网络的结构，每个簇头节点需要以主动“心跳”的方式维持簇内节点，以扩展COU策略维护长链路由表。因此每个周期的网络维护开销为$2m(N-h)+h\log^2(h)$，其中h是簇头节点的数量。簇内维护开销小于等于$2m(N-h)$，簇间维护开销为$h\log^2(h)$。扩展COU策略可以利用路由消息和反馈信息维护长链接路由表，因此簇间实际维护开销小于$h\log^2(h)$。

5　仿真与性能评价

本文使用C#语言开发了P2P网络模拟器，实现网络层和覆盖层的模拟仿真。在仿真实验中，为了提高模拟效率，实验忽略网络层模拟，仅在覆盖层实现Chord和LMCS仿真。在实验中，网络由N个节点构成，随机产生N个对象加入的网络中，每个节点随机选择网络中存在的对象查询40次，依此计算消息查询的平均跳数，系统参数Gmax=50，Gmin=10。

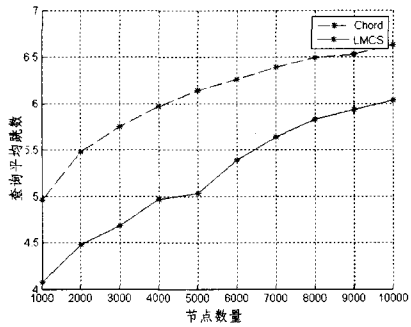

图 4　节点的查询平均跳数

为了公平比较 LMCS 与 Chord 算法的查询效率，将 Chord 的节点路由表大小设置为 24，LMCS 的 $k=24$ 和 $m=3$。从图 4 可以看出，节点数量 N 从 1000 增长到 10000，LMCS 与 Chord 有相似的变化趋势，但 LMCS 的查询消息平均跳数明显小于 Chord。

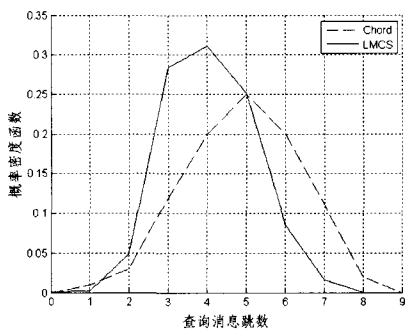

图 5　$N=1000$ 时的查询消息跳数的概率密度函数

图 5 显示的是当 $N=1000$ 时 LMCS 与 Chord 查询消息跳数的概率密度函数。LMCS 中所有查询消息都在 8 跳内到达，而且近 90%的消息在 3 至 5 跳内到达。因此 LMCS 的跳数分布比 Chord 更加集中在平均跳数附近。

表 1　不同 k 的查询平均跳数

N	LMCS				
	k=8	k=12	k=16	k=20	k=24
1000	4.0915	4.0804	4.0805	4.0805	4.0805
2000	4.6499	4.4846	4.4847	4.4847	4.4847
3000	5.0269	4.6786	4.6786	4.6786	4.6786
4000	5.5644	4.9694	4.9671	4.9671	4.9671
5000	5.7034	5.2328	5.0328	5.0328	5.0328

LMCS 中不同的长链接数量 k 对算法的查询效率的影响如表 1 所示，其中节点数量 N 从 1000 增长到 5000，长链接数量 k 从 8 增大到 24。结果说明，适当增大 k 能够提高查询效率，减少查询消息的平均跳数；同时较小的 $k(k\in[8,12])$ 就能够产成较小的查询消息的平均跳数。因此在较大的网络中，较小的长链接数量便能产生较好的查询效率。

表 2　不同 m 的聚类系数

N	Chord	LMCS			
		m=1	m=2	m=3	m=4
1000	0.2884	0.9715	0.6911	0.5712	0.5260
2000	0.2615	0.9700	0.6861	0.5760	0.5334
3000	0.2537	0.9727	0.6863	0.5699	0.5176
4000	0.2464	0.9718	0.6879	0.5705	0.5186
5000	0.2405	0.9715	0.6858	0.5723	0.5211

为了比较 LMCS 和 Chord 的聚类系数，Chord 的路由表大小等于 $O(\log N)$，LMCS 的 k 同样等于 $O(\log N)$，根据公式(3)计算聚类系数。如表 2 所示，LMCS 比 Chord 具有更大的聚类系数；同时聚类系数的变化趋势只与短链接数量 m 有关，不随节点数量 N 的变化而变化。

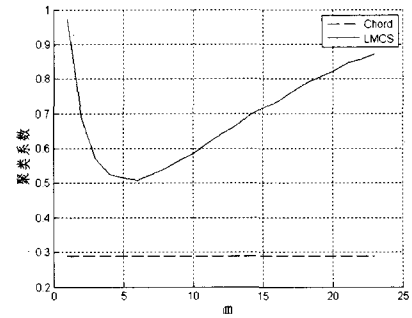

图 6　聚类系数曲线

图 6 显示的是 $N=1000$ 时，m 从 1 增加到 23 时 LMCS 和 Chord 的聚类系数变化情况，聚类系数在 $m=6$ 时达到最小值。当 $m<6$ 时，因为簇内节点和簇头节点之间的必然链接对聚类系数影响较大，因此聚类系数随 m 的增大而减少；当 $m>6$ 时，簇内节点的短链接数量增加，节点之间的重合程度也随之增加，而且簇内节点和簇头节点之间的短链接对聚类系数的影响也变得较小，因此聚类系数随 m 的增大而增大。

表 3　不同 m 的维护开销

N	Chord	LMCS			
		m=1	m=2	m=3	m=4
1000	28761	1873	3870	5830	9715
2000	73629	3890	7769	11671	15524
3000	115482	5854	11672	17516	23350
4000	157646	7798	15666	23337	31108
5000	189023	9729	19449	29161	38882

Chord 的路由表大小等于 $O(\log N)$，LMCS 的 k 同样等于 $O(\log N)$，根据表 3 所示，LMCS 的维护开销小于 Chord 一个数量级；同时，LMCS 的维护开销随 m 的增大而增加。因此当选择 $m=2$ 时，即可保证网络具有较大的聚类系数，又可以减少网络的维护开销。

结束语　本文提出了基于 DHT 的低维护开销结构化 P2P 网络，网络具有明显的小世界特征、较低的平均路径长度和较高的聚类系数，同时利用成簇机制、扩展 COU 策略和随机双向簇间路由策略，极大地降低网络维护开销。

参 考 文 献

[1] Rhea S, Geels D, Roscoe T, et al. Handling churn in a DHT[A]//Proceedings of the 2004 USENIX Annual Technical Conference(USENIX'04)[C]. Proceedings of Boston, USA, 2004: 127-140

[2] Kleinberg J. The small-world phenomenon: an algorithmic perspective[R]. Technical Report. Cornell Computer Science 99-1776, 2000

[3] Zhang H, Goel A, Govindan R. Using the small-world model to improve freenet performance[A]//Proc. IEEE Infocom[C]. 2002

[4] Merugu S S S, Zegura E. Adding structure to unstructured peer-to-peer networks: the use of small-world graphs[J]. Parallel Distributed Comput, 2005, 65(2): 142-153

(下转第 115 页)

$(mn-\sum_{i=1}^{m}t_i+m)H(K)$，则称该方案为最优的门限多重秘密共享方案。

4 一种最优的门限多重秘密共享方案

在这部分，我们提出了一种最优的门限多重秘密共享方案。该方案基于 Shamir 秘密共享，包含两个阶段：共享分发阶段和秘密重构阶段。前者由秘密持有者执行，后者由参与者实施。具体描述如下：

共享分发阶段：①随机选取共享 $S_j\in Z_p^*$，$j=1,\cdots,n$（p 是一个大素数），并把 S_j 秘密发送给参与者 j；②独立随机生成秘密 $K_1,K_2,\cdots,K_m\in Z_p^*$，对于每个秘密 K_i，由 $n+1$ 个点 $(0,K_i),(1,S_1),(2,S_2),\cdots,(n,S_n)$ 在域 Z_p 上构造 n 次多项式 $f_i(x)$，计算并公开 $f_i(n+1),f_i(n+2),\cdots,f_i(n-t_i+1)$ for $i=1,\cdots,m$。

秘密重构阶段：任意 t_i 个共享 $S_{j_1},S_{j_2},\cdots,S_{j_{t_i}}$，加上公开信息 $f_i(n+1),f_i(n+2),\cdots,f_i(n-t_i+1)$，得到 $n+1$ 个点 $(j_1,S_{j_1}),\cdots,(j_{t_i},S_{j_{t_i}}),(n+1,f_i(n+1)),\cdots,(n-t_i+1,f_i(n-t_i+1))$，从而可以构造一个 n 次多项式 $h(x)$。若用 $(X_i,Y_i)(i=1,2,\cdots,n+1)$ 依次表示这 $n+1$ 个点，则构造的多项式：

$$h(x)=\sum_{i=1}^{n+1}Y_i\prod_{j=1,j\neq i}^{n+1}\frac{x-X_j}{X_i-X_j}\bmod p$$

计算得到秘密 $K_i=h(0)$。

显然，这是一种多阶段使用的多重秘密共享方案，一次共享过程，可以分享多个秘密，每一阶段根据各自独立的公开信息可以重构一个秘密。以上方案基于 Shamir 秘密共享，因而是一种完备的门限方案；又因为 $|S_j|=|K|$，$j=1,\cdots,n$，所以该方案是理想的；其中公开的信息量为 $(mn-\sum_{i=1}^{m}t_i+m)|K|$，因而该方案是一种最优的门限多重秘密共享方案。

结束语 借鉴完全动态的门限秘密共享方案思想，重新定义了门限多重秘密共享方案，该方案是一种多阶段使用的多重秘密共享方案，一次共享过程可以分享多个秘密，每个秘密对应一个独立的公开信息，每一个重构阶段授权子集中的所有参与者合作，根据相应的公开信息可以恢复一个秘密；接着分析了共享和公开信息的下界，为寻找其它高效的门限多重秘密共享方案提供了理论基础；最后提出了一种最优的门限多重秘密共享方案，该方案基于 Shamir 秘密共享，是一种理想、完备的门限方案，能多阶段地共享多个秘密。其中共享与单个秘密同样大小，公开信息达到最优下界 $(mn-\sum_{i=1}^{m}t_i+m)H(K)$。因而非常有效，有着很好的应用前景。

参 考 文 献

[1] Shamir A. How to share a secret. Communications of the ACM 22,1979:612-613

[2] Blakley G. Safeguarding cryptographic keys // Proc. of the 1979 AFIPS National Computer Conference. AFIPS Press,1979,48:313-317

[3] Ito M,Saito A,Nishizcki T. Secret sharing scheme realizing general access structure// Proc. of IEEE Global Telecommunication Conf. Globecom 87. 1987:99-102

[4] Benaloh J C, Leichter J. Generalized secret sharing and monotone functions // Advances in Cryptology-CRYPTO'88, LNCS 403. 1990:27-35

[5] He J, Dawson E. Multistage secret sharing based on one-way function. Electronics Letters,1994,30 (19):1591-1592

[6] Harn L. Comment: Multistage secret sharing based on one-way function. Electronics Letters,1995,31 (4):262

[7] Harn L. Efficient sharing (broadcasting) of multiple secrets // IEEE Proceedings- Computers and Digital Techniques. 1995,142 (3):237-240

[8] He J,Dawson E. Multisecret-sharing scheme based on one-way function. Electronics Letters,1995,31 (2):93-95

[9] Karnin E D, Greene J W, Hellman M E. On secret sharing system. IEEE Trans. ,1983,IT-29 (1):35-41

[10] Jackson W A,Martin K M,O'keefe C M. Multisecret Threshold Schemes// Advances in Cryptology-CRYPTO'93, LNCS 773. 1994:126-135

[11] Blundo A,Santis A D,Crescenzo G D,et al. Multi-Secret Sharing Schemes // Advances in Cryptology-CRYPTO'94, LNCS 839. 1994:150-163

[12] Karnin E, Greene J, Hellman M. On secret sharing systems. IEEE Transactions on Information Theory,1983,IT-29(1):35-41

[13] Blundo C,Cresti A,Santis A D,et al. Fully dynamic secret sharing schemes // Advances in Cryptology, CRYPTO'93, LNCS 773. 1994:110-125

[14] Harn L, Hwang T, Laih C, et al. Dynamic threshold scheme based on the definition of cross-product in a n-dimensional linear space// Advances in Cryptology, Eurocrypt'89. 1990:286-298

[15] Sun H U, Shieh S P. On dynamic threshold schemes. Information Processing Letters,1994,52(4):201-206

[16] Li Ming-yan, Poovendran R. Disenrollment with Perfect Forward Secrecy in Threshold Schemes. IEEE Transactions on Information Theory,2006,52(4):1676-1682

(上接第 48 页)

[5] Milgram S. The small world problem [J]. Psychol Today, 1967, 2:60-67

[6] Kleinberg J. Navigation in a small-world[J]. Nature,2000: 406

[7] Mahajan R, Castro M, Rowstron A. Controlling the cost of reliability in peer-to-peer overlays[A] // Proceedings of IPTPS 2003[C]. vol. 2735 of Lecture Notes in Computer Science. Springer-Verlag, 2003:21-32

[8] Alima L, El-Ansary S, Brand P, et al. DKS(N,k,f): a family of low communication, scalable and fault-tolerant infrastructures for P2P applications[A]// Proceedings of 3rd IEEE/ACM Int'l Symposium on Cluster Computing and the Grid[C]. 2003: 344-350

[9] Leong B, Liskov B, Demaine E. EpiChord: parallelizing the Chord lookup algorithm with reactive routing state management [A] // Proceedings of the 12th IEEE ICON 2004[C]. Singapore, 2004:270-276

[10] Saroiu S, Gummadi P K, Gribble S D. A measurement study of peer-to-peer file sharing systems[A] // Proceedings of the 2002 Multimedia Computing and Networking (MMCN 2002)[C]. The International Society of Optical Engineering, 2002

[11] 杨峰，李凤霞，余宏亮，等. 一种基于分布式哈希表的混合对等发现算法[J]. 软件学报，2007，18(3)：714-721

[12] 李伟荣，吴国新，李建飞. Small-World 在对等网络中的应用研究[J]. 计算机工程与应用，2006，6：158-161