对一类迭代混沌分组密码的分析与改进

刘加伶[1] 张 红[1] 王 勇[2]

(重庆工学院计算机科学与工程学院 重庆 400050)[1] (重庆邮电大学管理学院 重庆 400065)[2]

摘 要 用选择明文攻击的方式对一种基于迭代混沌映射的加密算法进行了分析,并提出了相应的改进算法。在改进算法中,子密钥序列以密文反馈和从混沌映射中抽取数据相结合的方式产生,使子密钥序列在保持良好的均匀分布和随机统计特性的同时,还与明文相关,有效地增强了算法的安全性。最后对设计加密算法中应注意的问题进行了分析和总结。

关键词 混沌,密码分析,分组加密,信息安全

Cryptanalysis and Improvement on a Block Cipher Based on Iterating a Chaotic Map

LIU Jia-ling[1] ZHANG Hong[1] WANG Yong[2]

(School of Computer Science and Engineering, Chongqing University of Science and Technology, Chongqing 400050, China)[1]

(School of Management, Chongqing University of Posts and Telecomunications, Chongqing 400065, China)[2]

Abstract A chosen plaintext attack is presented to a block encryption system based on iterating a chaotic map. Moreover, an improved cryptosystem is proposed. With the ciphertext feed back, the subkey sequences are generated from the chaotic maps. The subkey sequences have good uniform distribution property, random statistical property and have relation to the plaintext, which enhance the security of the cryptosystem. Issues which should be considered are analyzed and pointed out when designing a chaotic cryptosystem.

Keywords Chaos, Cryptanalysis, Block cipher, Information security

随着 Internet 技术的飞速发展,网络通信逐渐成为人们进行信息交流的重要手段,信息的安全与保密显得越来越重要。近几年来,研究人员提出了许多基于混沌的加密算法。现有的研究成果表明混沌和密码学之间有着密切的联系,比如传统的密码算法敏感性依赖于密钥,而混沌映射依赖于初始条件和映射中的参数;传统的加密算法通过加密轮次来达到扰乱和扩散,混沌映射则通过迭代,将初始域扩散到整个相空间。传统加密算法定义在有限集上,而混沌映射定义在实数域内。虽然混沌映射具有参数和初值敏感、伪随机等良好的密码学特性,但是在使用它进行密码算法设计时还必须与密码学相关知识接合,进行详细的安全考证,否则设计出的密码系统很可能会存在安全漏洞而被攻击者击破。

文献[1]等对 Baptista 的加密方案[2]进行了改进。改进方案通过迭代混沌映射,从混沌序列中抽取比特值,然后通过对明文的移位和掩码操作来进行加密。该算法很好地解决了密文分布不均和加密速度慢的问题,但是其中还是存在安全漏洞。下面我们将对此加密方案进行分析,并提出相应的改进方案和在设计加密算法时应该注意的一些问题。

1 对文献[1]中加密算法的介绍

选用 logistic 映射:$f(x)=\mu x(1-x)$作为迭代产生混沌序列的映射,为了消除瞬时值的影响,将 logistic 映射迭代 N_0 次。将明文 m 划分为若干个长度为 l 字节的子块(取 $l=8$)。将 8 字节的明文 $p_j\ p_{j+1},\cdots,\ p_{j+7}$ 合并在一起组成一个 64 比特的明文块 P_j。然后再将 logistic 映射迭代 70 次,从每次的状态值中抽取出其小数点后的第 3 个比特,将其组合在一起构成二进制序列 $A_j=B_i^1B_i^2\cdots B_i^{64}$ 和 $D_j=B_i^{65}B_i^{66}\cdots B_i^{70}$,其中 B_i^n 表示第 n 次迭代时状态值的小数点后的第 i 比特(此处取 $i=3$)。对明文块 P_j 进行循环左移 D_j 位的操作,得到新的明文块为 P'_j,再用 A_j 对移位后明文块 P'_j 进行异或操作,得密文块 C_j。如果所有明文块均已被加密,则结束。否则将 logistic 映射迭代 D_j 次,然后重新产生加密下一个明文块的 A_j 和 D_j。整个加密过程和循环移位过程如图 1 和图 2 所示。

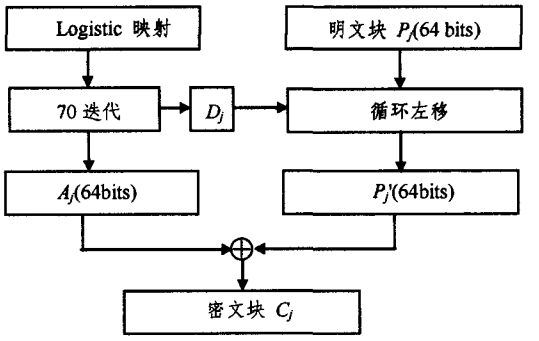

图 1 明文块加密方案

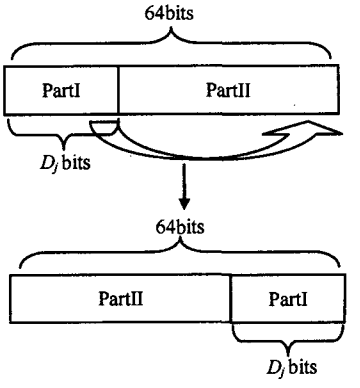

图 2 循环左移

解密方案与加密方案类似，对密文块解密时只需先进行异或，然后进行循环右移操作即可。

2 对加密方案的密码分析[3]

在上述算法中，对明文进行加密时主要是借助迭代 logistic 映射产生的两个二进制序列 A_j 和 D_j 来完成，因此可以将序列 A_j 和 D_j 看作是此密码系统的子密钥。只要具有了 A_j 和 D_j，就能进行加密和解密操作。仔细分析此密码系统，我们发现序列 A_j 和 D_j 仅由 logistic 映射完全确定，即只与混沌映射的初始值 x_0 和参数 μ(算法密钥)相关，而与明文、密文等其他因素无关。这意味着当算法密钥一定时，加密不同明文的 A_j 和 D_j 均是相同的，或者说当攻击者获取了加密任一明文的 A_j 和 D_j 后，就可以用此 A_j 和 D_j 直接攻击用相同密钥加密的密文。这正是其中存在的安全漏洞。

为了更形象地说明序列 A_j 和 D_j 完全依赖于密钥 x_0 和 μ，我们用上述算法加密两段完全不同的明文“abcdefghijklmnopqrstuvwxyz123456 …”和“0123456789ABCDEFGHIJKLMNOPQRSTUV…。密钥设置为 $\mu=3.9999995$，$x_0=0.1777$。对应以十六进制形式表示的 P_j，A_j，D_j 和 C_j 分别如表 1 和表 2 所示。从表 1 和表 2 中可以看到，虽然明文 P_j 不同，但是加密明文的 A_j，D_j 是相同的。

表 1 在 $\mu=3.9999995$，$x_0=0.1777$ 时，对明文“abcdefghijklmnopqrstuvwxyz123456…”的加密

j	P_j	D_j	A_j	C_j
1	61,62,63,64,65,66,67,68	1d	b2,93,08,6d,8d,93,50,cf	3e,3f,c4,80,81,bf,1c,a3
2	69,6a,6b,6c,6d,6e,6f,70	08	e1,ed,d3,bb,8b,27,be,b8	8b,86,bf,d6,e5,48,ce,d1
3	71,72,73,74,75,76,77,78	19	71,b9,86,3d,81,a4,d8,8f	99,53,6a,d3,71,46,3c,69
4	79,7a,31,32,33,34,35,36	20	9e,ad,e4,c9,7b,39,e7,e9	ad,99,d1,ff,02,43,d6,db
-	-	-	-	-

表 2 在 $\mu=3.9999995$，$x_0=0.1777$ 时，对明文“0123456789ABCDEFGHIJKLMNOPQRSTUV…”的加密

j	P_j	D_j	A_j	C_j
1	30,31,32,33,34,35,36,37	1d	b2,93,08,6d,8d,93,50,cf	d4,15,ae,ab,6b,95,76,89
2	38,39,41,42,43,44,45,46	08	e1,ed,d3,bb,8b,27,be,b8	d8,ac,91,f8,cf,62,f8,80
3	47,48,49,4a,4b,4c,4d,4e	19	71,b9,86,3d,81,a4,d8,8f	e5,2f,1e,a7,1d,2a,48,1d
4	4f,50,51,52,53,54,55,56	20	9e,ad,e4,c9,7b,39,e7,e9	cd,f9,b1,9f,34,69,b6,bb
-	-	-	-	-

针对上述分析，我们采用选择明文方式对密码系统进行攻击，获取加密过程中的子密钥序列 A_j 和 D_j。由加密算法知，A_j 和 D_j 等效于加密过程中密钥，因此获取完整的子密钥序列 A_j 和 D_j 等效于破解了整个密码系统。具体的攻击步骤如下，对应的流程图如图 3 所示。

步骤 1 选择一个全零的特殊的明文 Pz，其长度与所要攻击的密文长度相同。设 Pz_j 为 Pz 的第 j 个明文块(64 比特)，Cz_j 为 Pz_j 对应的密文块，则：

$$Cz_j=F(Pz_j,D_j)\oplus A_j \tag{1}$$

其中函数 $F(A,B)$ 表示将 A 循环左移 B 个比特。由于 Pz_j 为一个全零的明文块，故 $F(Pz_j,D_j)=Pz_j$，于是有

$$Cz_j=F(Pz_j,D_j)\oplus A_j=Pz_j\oplus A_j=A_j \tag{2}$$

即序列 Cz_1，Cz_2，… 与序列 A_1，A_2，… 相同。在 $\mu=3.9999995$，$x_0=0.1777$ 的情况下，加密 Pz_j 时的 A_j，D_j 和密文 Cz_j 如表 3 所示，表中的数值为十六进制形式。

步骤 2 选择另一个特殊的明文 Ps，其长度与所要攻击的密文长度相同。Ps 中每个块(64 比特)均为 $0\underbrace{11\cdots11}_{63}$，即其中仅有 1 个比特为 0，其余比特为 1。设 Ps_j 为 Ps 的第 j 个块，Cs_j 为对应的密文，则

$$Cs_j=F(Ps_j,D_j)\oplus A_j \tag{3}$$

由于在步骤 1 中已经获取了 A_j，因此有

$$Cs_j\oplus A_j=F(Ps_j,D_j)\oplus A_j\oplus A_j=F(Ps_j,D_j) \tag{4}$$

由于在 $F(Ps_j,D_j)$ 中只有 1 个比特的值为零，因此能很方便地找到 0 比特所在的位置。假设 0 比特所在的位置为 y_j，显然 y_j 的值等于 D_j。这样，我们就获得了加密算法中另一重要的序列 D_1，D_2，… 。在 $\mu=3.9999995$，$x_0=0.1777Pz_j$ 的情况下，加密 Ps_j 时的 $F(Ps_j,D_j)$，y_j 和对应的密文 Cz_j 如表 4 所示，表中的数值为十六进制形式。

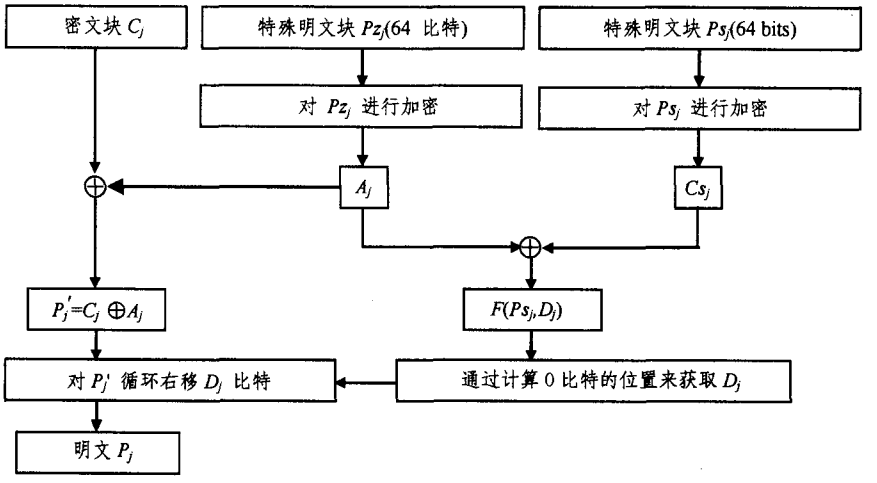

图 3 选择明文攻击的流程图

表 3　在 $\mu=3.9999995, x_0=0.1777$ 时，对明文“00000000…”的加密

j	Pz_j	D_j	A_j	C_j
1	00,00,00,00,00,00,00,00	1d	b2,93,08,6d,8d,93,50,cf	b2,93,08,6d,8d,93,50,cf
2	00,00,00,00,00,00,00,00	08	e1,ed,d3,bb,8b,27,be,b8	e1,ed,d3,bb,8b,27,be,b8
3	00,00,00,00,00,00,00,00	19	71,b9,86,3d,81,a4,d8,8f	71,b9,86,3d,81,a4,d8,8f
4	00,00,00,00,00,00,00,00	20	9e,ad,e4,c9,7b,39,e7,e9	9e,ad,e4,c9,7b,39,e7,e9
-	-	-	-	-

表 4　在 $\mu=3.9999995, x_0=0.1777$ 时，对明文“”7FFFFFFFFFFFFFFF7FFFFFFFFFFFFFFF…”(十六进制形式)的加密

j	Ps_j	D_j	$F(Ps_j, D_j)$	C_j
1	7f,ff,ff,ff,ff,ff,ff,ff	1d	ff,ff,ff,ff,ef,ff,ff,ff	4d,6c,f7,92,62,6c,af,30
2	7f,ff,ff,ff,ff,ff,ff,ff	08	ff,ff,ff,ff,ff,ff,ff,7f	1e,12,2c,44,74,d8,41,c7
3	7f,ff,ff,ff,ff,ff,ff,ff	19	ff,ff,ff,ff,fe,ff,ff,ff	8e,46,79,c2,7f,5b,27,70
4	7f,ff,ff,ff,ff,ff,ff,ff	20	ff,ff,ff,ff,7f,ff,ff,ff	61,52,1b,36,04,c6,18,16
-	-	-	-	-

3　对迭代混沌映射密码系统的改进

从我们掌握的资料来看，G. Alvarez 等人对 Baptista 算法及其改进方案的攻击[4-7]、Jun Wei 等对 Pareek 所提出的密码系统的攻击[8,9]，以及上述我们的攻击之所以能够成功，本质上是因为这些密码系统都存在类似的安全漏洞：通过混沌映射产生的子密钥只与混沌映射本身有关，而与其他因素(比如明文或密文)无关。

要改进此迭代混沌映射的密码系统，提高其安全性，必须弥补此安全漏洞。为此，可采用明文反馈或密文反馈方式对混沌映射的迭代产生影响，从而使子密钥序列 A_j 和 D_j 与明文或密文相关。此处，我们以密文反馈方式对算法的改进予说明。改进后的算法为：

产生第一个分组密文的方法与原方案相同，但是在产生密文 C_j 后，按照式(5)和(6)计算新的迭代次数 D^*，然后将迭代 Logistic 映射的次数由 D_j 更改为 $D*$。

$$f(C_j)=c_1+c_2+\cdots+c_8 \tag{5}$$

$$D^*=D+f(C_j)\bmod 64 \tag{6}$$

其中，C_j 为第 j 个密文块，$c_i(i=1, 2, \cdots, 8)$ 为 C_j 的第 i 个字节的值。显然，在改进的算法中，不同的明文会得到不同的密文，从而得到不同的 D^*，并最终产生不同的 A_j 和 D_j。即 A_j 和 D_j 不仅与混沌映射相关，而且与明文相关，从而有效地避免了原加密系统的安全漏洞。

为对改进后的算法进行测试，进行了如下检验：

(1) 取加密两段不同的明文，对应的 A_j，D_j 和 C_j 分布如表 5 和表 6 所示。

表 5　在 $\mu=3.9999995, x_0=0.1777$ 时，用改进算法对明文“abcdefghijklmnopqrstuvwxyz123456…”加密

j	P_j	D_j	A_j	C_j
1	61,62,63,64,65,66,67,68	1d	b2,93,08,6d,93,50,cf	76,55,c0,a7,41,5d,80,0d
2	69,6a,6b,6c,6e,6f,70	30	71,64,f7,d7,04,37,38,dc	18,0e,9c,bb,69,59,57,ac
3	71,72,73,74,75,76,77,78	0e	1d,8c,43,9e,ad,e4,c9,7b	0a,ab,74,d9,fa,83,be,fc
4	79,7a,31,32,33,34,35,36	34	fd,30,51,7a,63,6c,d9,e9	cc,02,62,4e,56,5a,a0,93
-	-	-	-	-

表 6　在 $\mu=3.9999995, x_0=0.1777$ 时，用改进算法对明文“0123456789ABCDEFGHIJKLMNOPWRSTUV…”加密

j	P_j	D_j	A_j	C_j
1	30,31,32,33,34,35,36,37	1d	b2,93,08,6d,8d,93,50,cf	d0,f7,6e,05,e7,ff,3e,af
2	38,39,41,42,43,44,45,46	0e	ba,77,71,64,f7,d7,04,37	39,e3,65,40,c3,93,50,54
3	47,48,49,4a,4b,4c,4d,4e	31	61,8f,60,69,36,23,d9,8e	f7,17,fa,f5,b8,b3,4b,1a
4	4f,50,51,52,53,54,55,56	1d	3f,4c,14,5e,98,db,36,7a	9f,ee,b0,f8,30,71,9a,e4
-	-	-	-	-

(2) 随机选择一个明文，将其第一个字节的值改变 1 个比特得到另一个新的明文，然后对这两个明文进行加密，比较其密文的差值，前 500 个字节之间的差值如图 4 所示。

从检验测试中可以看出，我们的改进是有效的。同时，由于式(5)和式(6)的计算量很小，改进后的算法仍然保持了原算法的高效性和密文分布的均匀性。

4　设计迭代混沌的密码方案时应该注意的问题

利用混沌技术设计加密系统时常采用如下的思想：首先选择一个适当的混沌映射，以设定的映射中的参数值和初始状态作为加密算法的主密钥。然后对映射进行迭代，产生子密钥序列。最后利用这些子密钥序列来加密明文。以这种思路设计密码系统时应注意如下的问题。

(1)混沌映射的选择是设计加密算法时的关键步骤之一，除了注重映射本身的特点之外，还应关注映射中参数的个数和参数的取值范围。通常应该选择参数个数多且取值范围广的映射，或者人为地扩展参数的个数和取值范围。因为这样能够有效增加主密钥空间，增强加密算法抗穷举破解的能力。

(2) 根据选定的混沌映射如何产生相应的子密钥序列，是在设计加密系统时应注意的另一个重要问题。因为子密钥序列直接作用于各明文分组，其自身的特性对密码系统的安全性有非常重要的影响。当前已有许多混沌序列的生成算法，虽然利用这些算法可获取一些具有均匀分布和随机统计特性的子密钥序列，但是这些子密钥序列常仅依靠于混沌映射中设定的参数值和初始状态(主密钥)。即主密钥一旦确定，相应的子密钥序列也确定下来，对不同的明文均采取相同的子密钥序列进行加密，这为攻击者进行选择明文攻击提供了一定的条件。为了保证加密系统的安全性，必须相应地增加变换的复杂性和变换的轮次，显然这不利于提高加密的速度，也未能充分发挥混沌加密的优势。为此，在设计子密钥序列的生成算法时，可将明文的信息引入其中，使子密钥序列不仅与主密钥相关，而且与明文相关。同时还得注意子密钥序

列对明文的依赖性不能过大，以免造成信息的泄漏。

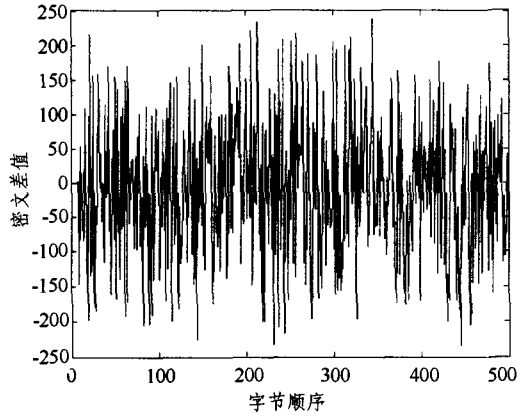

图 4　用改进后的算法对两个仅首字节相差 1 比特的明文进行加密后所得密文的差值

结束语　对一种基于迭代混沌映射的加密系统进行了详细分析，指出了其中存在的安全漏洞，并用选择明文的方式对其进行了攻击。然后，我们提出了相应的改进方法。最后对设计基于迭代混沌映射的分组加密算法时应当注意的问题进行了总结。

参 考 文 献

[1] Xiang Tao, Liao Xiaofeng, Guo Ping, et al. A novel block cryptosystem based on iterating a chaotic map. Physics Letters A, 2006, 349: 109-120

[2] Baptista M S. Cryptography with chaos. Physics Letters A, 1998, 240: 50-57

[3] Wang Yong, Liao Xiaofeng, Xiang Tao, et al. Cryptanalysis and improvement on a block cryptosystem based on iteration a chaotic map. Physics Letters A, 2007, 363: 277-281

[4] Alvarez G, Montoya F, Romera M, et al. Cryptanalysis of dynamic look-up table based chaotic cryptosystems. Physics Letters A, 2004, 326: 211-218

[5] Alvarez G, Montoya F, Romera M, et al. Keystream cryptanalysis of a chaotic cryptographic method. Computer Physics Communications, 2004, 156: 205-207

[6] Wong K-W. A fast chaotic cryptographic scheme with dynamic lookup table. Physics Letters A, 2002, 298: 238-242

[7] Wong K-W. A combined chaotic cryptographic and hashing scheme. Physics Letters A, 2003, 307: 292-298

[8] Wei Jun, Liao Xiaofeng, Wong K-W. Analysis and improvement for the performance of Baptista's cryptographic scheme. Physics Letters A, 2006, 354 (1): 101-109

[9] Pareek N K, Patidar V, Sud K K. Cryptography using multiple one-dimensional chaotic maps. Commun Nonlinear Science and Numerical Simulation, 2005, 10: 715-723

(上接第 140 页)

个数据标识的大小为 4 个字节。实验中数据类有 4 个数值属性，共有 5000 个数据。查询均为对数据类的 4 个属性的联合查询。实验测试了在各种大小的查询结果集下两种算法的通信开销(数据传输量)和查询时间(指发出查询请求到收到全部查询结果之间的时间长度)。

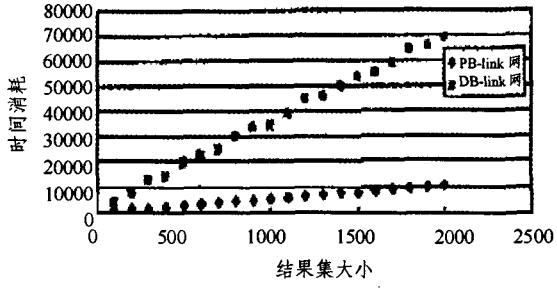

图 2　PB-link Tree 与 DB-link Tree 进行复杂查询时的查询时间

从图 1 和 2 可以算出 PB-link 树的通信开销和查询时间平均是 DB-link 树的 0.21 和 0.143。这说明 PB-link 树优化了 P2P 环境分布式结构化索引的效率，降低了查询代价，可见在广域网环境下 PB-link 的带宽开销和查询时间均远小于 DB-link Tree。

4　Lazy Update 与 Active Update 的性能对比

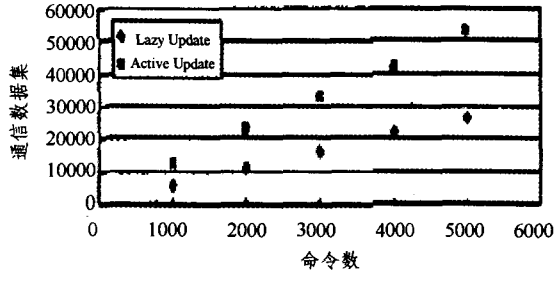

图 3　Lazy Update 和 Active Update 执行插入、删除命令时的平均查询时间对比

在 PB-link Tree 的节点分裂、合并策略上分别使用 Active Update 策略和 LazyUpdate 策略。实验首先进行一定次数的随机插入、删除操作，然后进行数据查询。同样测试了两种策略下的数据传输量和查询时间(见图 3，图 4)。

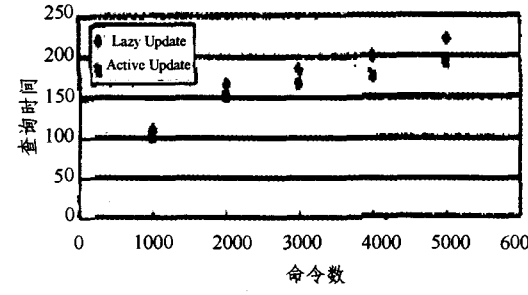

图 4　Lazy Update 和 Active Update 执行插入、删除命令时的数据传输量对比

从图 3 和图 4 中可以看到，Lazy Update 执行插入、删除命令时所需的通信开销远小于 Active Update，而二者执行一定的插入、删除命令后的查询时间相差很少。因此，Lazy Update 更加适合于 P2P 环境下的索引维护。

结束语　PB-link Tree 算法解决了已有算法在执行联合查询时带宽开销过大的问题。通过使用分布式 B+树实现数值属性区段查询、字符串属性的子串查询和多属性之的联合查询。通过哈希定位策略将 B+树叶节点分布到 P2P 系统中的多个节点上，保证一份数据的多属性索引项存储于同一个节点，避免联合查询中中间节点求交集时的大量数据传输，减少了查询开销，提高了查询效率。

参 考 文 献

[1] Rowstron A, Druschel P. Pastry: Scalable, distributed object location and routing for large -scale peer-to-peer systems. Lecture Notes in Computer Science, 2001, 2218: 329-350

[2] Shi S. Making Peer-to-Peer Keyword Searching Feasible Using Multi-levelartitioning // 3rd International Workshop onPeer-to-Peer Systems. February 2004: 400-408

[3] Zegura E W. How to modal an internetwork // Proc. of the INFOCOM'96. 1996: 594-602

[4] Dingledine R. The free haven project: Distributed anonymous storage service // Proc. of the Workshopon Design Issues in Anonymity and Unobservability. 2000(3): 67-95