doi:10.3772/j.issn.1002-0470.2016.12.002

DHT 系统的安全性优化方法研究①

史建焘② 夏清泉 张兆心

(哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001)

摘 要 对分布式哈希表(DHT)系统的安全脆弱性问题进行了研究,提出了多种安全性优化策略,并给出了一个原型系统。进行了真实网络实验,实验数据表明,现有 DHT 网络易受索引毒害和路由污染攻击,产生的错误查询结果甚至会引发更大规模的网络安全事件。通过改进一个个 DHT 系统的节点 ID 生成机制、路由表更新机制和搜索路径选择机制,从系统运行的各个阶段提升其安全性,抵御攻击者共谋。基于上述方法设计的原型系统在保证平均查询跳数增加不到 1 跳的情况下,在共谋攻击节点占比 60% 的网络中,将系统查询成功率保持在 65% 以上,其方法适用于各种分布式哈希表结构,具有重要的实际应用前景。

关键词 对等网络, 分布式哈希表(DHT), 安全优化, 路由污染, 索引毒害

0 引 言

分布式哈希表(distributed Hash table,DHT)是传统 P2P 领域中的重要技术,DHT 使得 P2P 结构不再依赖于传统的中心组件,成为了真正意义上的完全分布式网络结构。此外,DHT 技术也能够解决泛洪等无结构化分布式路由方法盲目搜索的缺点,通过结构化的方式较准确地定位资源。因此,作为分布式计算研究领域重要的技术手段,DHT 技术在云计算[1],电子商务[2],命名数据网络[3]等新兴研究领域中也具有重要的应用价值,在新网络体系结构下也有着良好的发展前景,可以与其他现有技术相互融合。但是,由于 DHT 在设计上缺乏节点身份验证机制,这样的安全性缺陷也限制了其发展。因此,提高 DHT 技术安全性的研究十分重要,尤其是在节点路由过程中的安全性,不仅限于当前的应用环境和应用模式下,而在网络的大背景下,也有着重大研究意义。本文以 BT 的 Mainline DHT 为主要研究对象,通过实际系统证明 DHT 网络易受攻击,针对主要攻击手段,提出了多种安全性改进机制,并给出了仿真实验结果。

1 相关工作

在 DHT 网络中,资源和节点都被分配了长度相同的 ID 值,通过特定计算法则(如异或运算),来计算不同 ID 值之间的距离。在资源发布和检索过程中,通过递归查找网络中距离目标 ID 值最近的节点或资源,提供网络服务。常见的攻击方式都是根据 DHT 网络这种基于距离的查找方式,通过伪造节点和污染路由表进行攻击,使查找结果落入到攻击者制造的陷阱中,造成查找失败。文献[4]最先系统全面地分析了 DHT 网络的安全脆弱性。文献[5]则将这些攻击方式归纳为女巫攻击、日蚀攻击以及路由和存储攻击等。针对这些攻击方式,目前的研究多是通过提高路由表中冗余节点的方式,降低恶意节点在路由表中的比重[6,7]。文献[8]将这些方

① 国家自然科学基金(61402137)资助项目。

② 男,1980 年生,博士,讲师;研究方向:网络安全,云安全,P2P 系统安全,联系人,E-mail: shijiantao@hit.edu.cn

(收稿日期:2016-09-08)

法分类为冗余消息方法、异常检测方法和社会网络方法。文献[9-11]为典型的冗余消息方法,主要实现方式包括增加路由消息数量、构造多个不同的路由路径等,其目的都是为了增加正确节点收到请求消息的概率。文献[12-14]通过旁路监听的方式发现消息路由转发过程中存在的异常,从而识别恶意节点,并在出现异常的路由位置重发和转发消息,保证查询正常完成。文献[15-17]通过在 DHT 网络中引入社交网络的方式,将良性节点和攻击节点割裂,提高恶意攻击的代价。

本文提出的几种 DHT 安全性优化策略,分别从节点 ID 生成过程、路由表构造过程和搜索路径选择过程出发,从不同角度增加攻击者的攻击代价,降低恶意节点信誉值,提高良性节点间的协作,从而提高 DHT 网络的安全性。对于节点 ID 生成机制,文献[13]提出了一种安全性解决方案,但是需要额外引入中心验证组件,改变了 DHT 系统的网络结构,破坏了完全分布式的设计理念。而文献[16]给出的方法,由于验证过程复杂,难以在实际系统中应用。针对提供节点信誉的方法,文献[19]提出了 FFP 系统,但是由于无法避免攻击节点通过协作方式进行虚假交易,也难以引入到实际系统中。本文借鉴了上述研究工作部分思路的同时,给出了更易在实际 DHT 系统中实现的安全性提升策略。

2 DHT 脆弱性分析

2.1 攻击验证系统设计

为分析 DHT 网络的安全脆弱性,本文设计了一种能够应用于 BT 的 Mainlie DHT 中的攻击验证系统。图 1 是系统整体结构,通过节点爬虫爬取 DHT 网络中的节点信息和路由表信息,通过索引毒害攻击和路由污染攻击破坏 DHT 网络的路由过程和搜索过程。攻击时需要通过数据库存储获取的节点信息和路由信息,并通过中心调度模块管理虚假节点和攻击节点的行为。

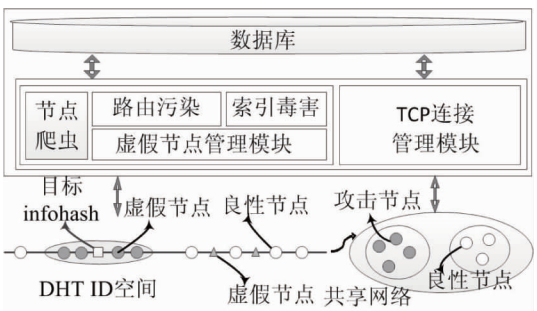

图 1 攻击验证系统结构

2.2 索引毒害

DHT 网络中每个资源的索引信息都由 ID 空间上距离其最近的一组节点负责,本文称之为索引节点集(IndexPeerSet),所有指向索引节点集的节点则称之为关键节点集(CriticalPeerSet),索引毒害攻击的核心思想就是在索引节点集和关键节点集中插入大量伪造的虚假节点。算法 1 描述了索引毒害攻击的过程,首先通过节点爬虫获取网络中大量的节点,并以这些节点为初始节点,通过发送节点查询报文的方式,不断发现索引节点和关键节点,并在这些节点上发布虚假节点链接。

算法 1: IndexPoisonAttack(Target)

输入: Target – 要攻击的目标文件的 Infohash

1. *get RandomPeerSet from* DHT *crawler*; //通过 DHT 爬虫获取随机节点集合
2. *CriticalPeerSet*←Φ; //初始化关键节点结合
3. *IndexPeerSet*←Φ; //初始化索引节点集合
4. *Target*←*target infohash*;
5. **for** $p \in$ RandomPeerSet **do**
6. *send Get _peers*(*Target*) *Message to p*; //随机节点集合向 DHT 网络发布节点请求消息
7. **endfor**;

```
8.  while (规定时间间隔收到返回消息)
9.    q←source peer of the message;
10.   type←responded type tag; //根据返回消息的类型分别处理
11.  if type = NODES then // 消息中返回的是 8 个 DHT 节点
12.  iNode[ ]←received DHT nodes;
13.  for i←1 to 8 do
14.  send Get _peers(Target) Message to iNode[i]; //继续向 8 个新节点发送节点请求
15.  iNode[i]. previous←q;
16.  else //Type = INDEX
17.  send Find _ node(target) message to q;//节点 q 为内容索引
18.  if q. previous ∉ IndexPeerSet then
19.  CriticalPeerSet. insert(q. previous); //将节点 q 的前序节点加入关键节点集合
20.  IndexPeerSet. insert(q); //将节点 q 加入索引节点集合
21.  endif;
22.  endif;
23.  endwhile;
24.  for p ∈ IndexPeerSet∪CriticalPeerSet do
25.    send Annouce _ peer(Sybil IDs) to p;  //迭代完成后对关键节点和索引节点进行污染
26.  endfor;
27.  return;
```

2.3 路由污染

路由污染的核心是找到一个合理的 ID 区间范围,污染这个范围内正常节点的路由表,使这些节点的路由表中包含恶意虚假节点。通过多次实验对比,本研究选择 ID 值与目标 Infohash 具有 20 至 50 个公共子前缀的节点,通过主动和被动方式污染其路由表。主动污染过程通过爬虫爬行网络中距离目标哈希一定范围内的节点,通过与这些节点通信以图污染其路由表。被动污染通过响应外来的查询消息来获得更好的污染效果,其中对 Ping,Find _ node 和 annouce _ peer 消息处理和真实客户端类似,只有对 get _ peers 消息的处理比较复杂,过程如图 2 所示。如果查询请求落入到本研究的虚假协作节点集中,则返回 8 个距离目标 Hash 值更近的 ID。

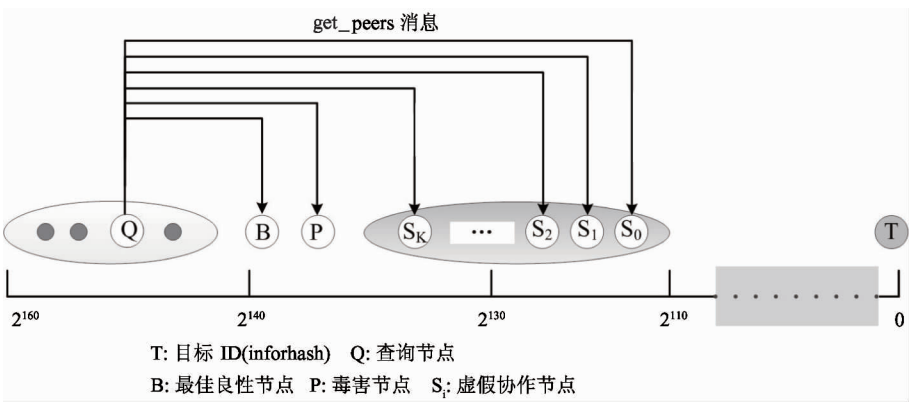

图 2 get _ peers 消息路由污染过程

2.4 实验结果

为不破坏 DHT 网络,本文将虚假节点设计为只具有路由功能的轻客户端,进行了多组对比实验。为了验证对搜索过程的影响,本研究每隔一小时在 DHT 网络中发送 100 次针对目标 Hash 值的 get _ peers 请求,图 3 显示了返回的结果中虚假索引所占的比例,最后可以稳定在 75% 以上,表明搜索过程已经被攻击系统所控制。

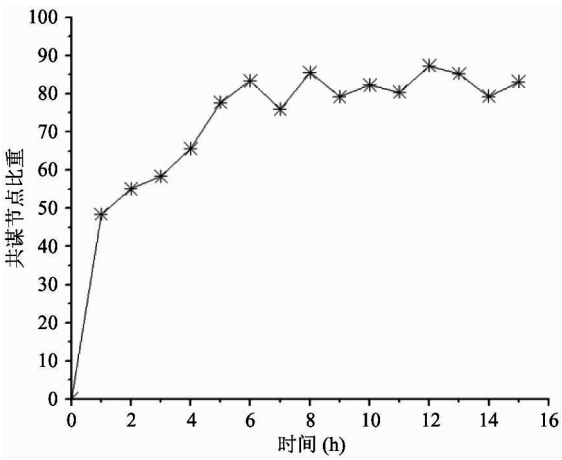

图 3 搜索结果中虚假节点的比例

最后,为了更直观地观察网络中节点路由表被污染的情况,选取了 DHT 网络中两段 ID 区间内的节点,通过爬虫爬取其路由表,记录路由表中虚假节点的所占比例。图 4 显示,经过一段时间的路由污染,两个区间的路由表污染程度可分别达到 50% 和 70%,这样的污染率完全可以控制 DHT 网络的路由过程。

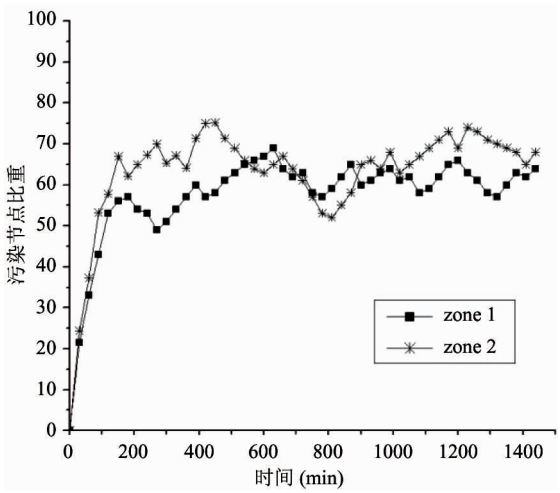

图 4 路由表污染情况

3 安全性优化方法

DHT 网络的核心设计思想包括定向搜索策略、趋向化设计和基于 K 桶的路由表,然而这些经典设计同时也是造成 DHT 网络安全性缺陷的主要原因。本节介绍的 DHT 安全性优化方法将分别围绕节点 ID 的生成,路由表更新算法和搜索路径生成方法进行改造。

3.1 节点 ID 生成机制

现在大多数标准的 DHT 协议,都采用在节点初始化时随机生成节点 ID 的方式,没有更为严格的 ID 生成规则和合规性检查方法。这成为了攻击者所利用的主要漏洞,本节所设计的节点 ID 生成策略如图 5 所示。

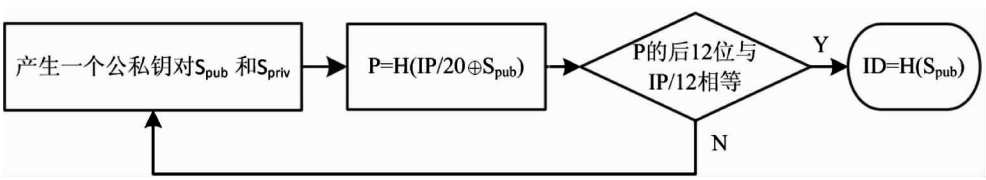

图 5 节点 ID 生成策略

由客户端通过系统调用产生一组公私钥,产生 ID 时将公钥与节点网络地址的前 20 位进行异或并求其 Hash 值,重复此过程直到产生的结果后 12 位与节点 IP 地址的后 12 位相等,则将此结果 P 作为节点的 ID。通过实验,产生一个符合要求的 ID 值大概需要几分钟的时间,即需要大概 2^{12} 次计算。攻击者要产生一个符合攻击要求的节点(与目标 Hash 具有 20 到 50 个公共子前缀),则需要 2^{12+20} 到 2^{12+50} 次计算,计算代价造成难以产生足够数量的虚假 ID。

3.2 路由表更新机制

目前 DHT 系统的路由表更新算法是:对于最深层 K 桶满时根据对应层数的二进制值分裂成 2 个 K 桶,对于已经分裂过的 K 桶,有选择地替换已有路由表中的节点,一般是用新节点替换老节点。这就造成攻击者可以通过不断与被攻击节点通信污染其路由表。本小节设计的路由表更新算法如算法 2 所示。

算法 2: UpdateRoutingTables (r)
输入: r:待插入的路由节点

1. 找到 r 中负责目标 id 的 K 桶 B;
2. L_B 是 B 在路由表中的层数;
3. s 表示节点本身;
4. **if** B is not full **then**
5. *insert*(B, r); //桶未满,将 r 插入到 K 桶 B。
6. *Calculate* D_k of B;
7. **If** $D_k(2^{n-L-\beta}$ **then**
8. *remove*(B,r); //插入 r 后, D_k 不满足公式(2),将 r 从路由表删除。
9. **endif**;
10. **else**
11. **if** $s \in$ B **then**
12. split(B);//最深层 K 桶分裂成两个桶,新的桶 B 仍为 r 要插入的桶。
13. **endif**;
14. Calculate D_k of B; //计算原始 K 桶中的 D_k
15. **for** $n \notin$ B **do**
16. Replace (n,r);
17. Calculate $D_{k'}$ of B;
18. **if** $D_{k'} < D_k$ **then**
19. Replace (r,n); //尽量保持 K 桶的 D_k 不增加
20. **endif**;
21. **endfor**;
22. **endif**;
23. **return**;

为了防止同一个 K 桶中被攻击者插入多个距离目标 ID 更近的节点,本文以 K 桶中最近的两个节点间的距离为判断依据,决定新节点是否可以插入到路由表中。

由于 BT 网络中节点总数相对稳定,因此节点路由表的分裂次数和 K 桶的层数都是相对稳定的,当节点在网络中停留一定时间后其 K 桶一般不再分裂,最深层一般是和节点自身距离最近的节点。设 K 桶容量为 $K = 2^{\sigma}$,n 为节点 ID 的长度,$N = 2^r$ 表示网络中节点的总数,当 K 桶稳定时,最深层桶内两个节点的距离应在区间 $[2^{\sigma+n-r-1}, 2^{\sigma+n-r}]$。由于每个桶内节点最小公共子前缀为其层数,如果 K 桶最大层数为 L, 则有 $2^{\sigma+n-r-1} < 2^{n-L}$, 即 $L < \tau + 1 - \sigma$。

本文定义第 i 层 K 桶中最近两个节点的距离为 D_k, 则其满足

$$D_k = \min\{ID_i \oplus ID_{i+1}\},\ i \in [1, 2^{\delta} - 1],\ ID_i < ID_{i+1} \tag{1}$$

插入到第 L_R 层 K 桶中的节点需要满足下式才会被插入:

$$D_k > 2^{n-L_R-\beta} \tag{2}$$

β 为可调整参数。

3.3 搜索路径选择机制

改进的路由表更新算法,避免了攻击者在一个节点的 K 桶中插入多个虚假攻击节点,为了进一步防止虚假节点作为最近节点被选为搜索过程中的下一跳节点,本文定义了节点信誉值 HP, 如下式所示:

$$HP(N_i) = \lambda \cdot \frac{\sum_{j=m-d}^{m} P_j(N_i)}{d} + (1 - \lambda) \frac{\sum_{j=0}^{m-d-1} P_j(N_i)}{d} \tag{3}$$

其中 $P_j(N_i)$ 表示当前节点与节点 N_i 第 j 次通信的可信度,通过参数 $\lambda \in [0,1]$ 来增加最近 d 次通信的可信度所占的权重。

为了防止攻击者构造多个更靠近目标 Hash 值的虚假节点,还给出了责任区间的概念,扩大了负责目标 Hash 值的节点集范围,责任区间内的节点 ID 满足公式

$$ID_r \oplus h < 2^{\omega} \tag{4}$$

其中 ω 为调节责任区间大小的参数, h 为目标 Hash 值:

查询发起节点选择路由表中距离目标 Hash 最接近的 α 个 K 桶中 HP 最高的节点作为候选节点。查询路径中的其他节点,判断自己是否是查询责任区间中的节点,如果是则同样选择 K 个 HP 值最高节点返回,并结束查询。HP 评分过程根据查询是否成功来评判,成功则路径上所有节点评分为 1,失败则所有节点评分为 0。

4 实验与性能评价

DHT 网络的核心是分布式的资源发布和搜索功能,因此本节以查询成功率来评价不同攻击场景下优化后协议的安全性。查询成功率(lookup success ratio, LSR)的定义如下:

$$LSR = \frac{Lookup_{\text{sucessful}}}{Lookup_{\text{total}}} \tag{5}$$

其中, $Lookup_{\text{total}}$ 表示每组实验的总查询数, $Lookup_{\text{sucessful}}$ 表示每组实验中查询成功的次数。实验通过 Oversim 平台在仿真环境下进行,共模拟了 5000 个节点,攻击节点采用的攻击方式为索引毒害和路由污染。

图 6 给出了网络中的虚假攻击节点数占网络总结点数 20%,责任区间内节点数为 2,协议分别采用 2 路和 3 路并行查询时的查询成功率。图 7 给出了同样攻击场景下,责任区间节点数为 16 时的查询成功率。可以得出如下结论:(1)优化后的 DHT 协议查询成功率都要高于原始协议。(2)采用2路并行查询要优于 3 路并行查询。(3)责任区间节点数为 16 时更能抵御攻击。这是因为随着查询次数的增加,恶意攻击节点获得的评价会不断降低,良性节点间协作的路由表健壮性不断提高。责任区间节点数越高,查询过程所需的跳数越少,查询有更大的概率收敛于良性节点。

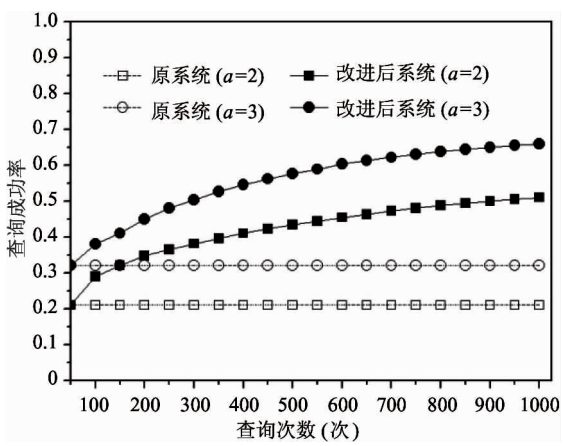

图 6 查询成功率(责任区间 =2,攻击节点占比 20%)

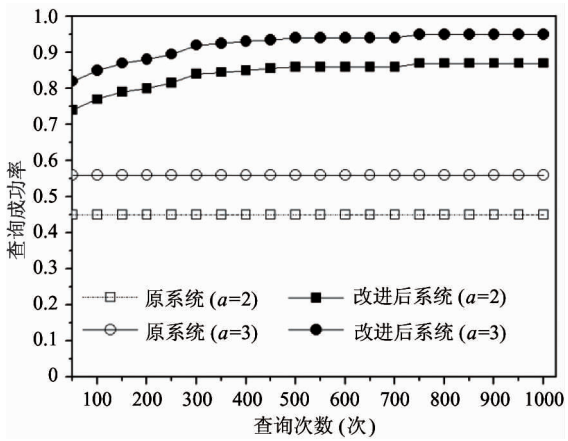

图 7 查询成功率(责任区间 =16,攻击节点占比 20%)

图 8 给出了网络中的虚假攻击节点数占网络节点总数的 30%,责任区间内节点数为 2 时,K 桶中虚假攻击节点占比。可以得到如下结论:(1)高层 K 桶中的虚假节点更多。(2)随着查询次数的增加,K 桶中虚假节点占比不断减少。

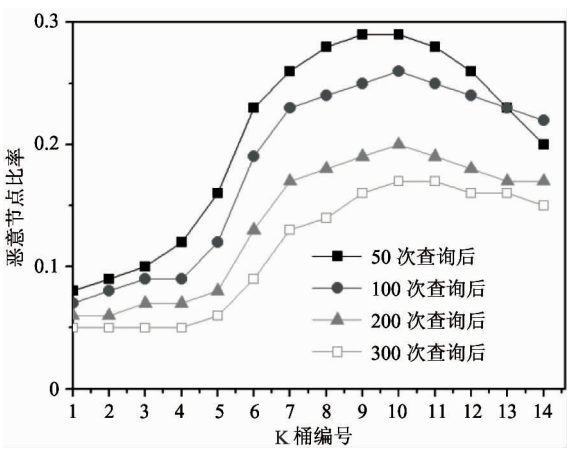

图 8 K 桶中恶意节点所占比例随查询次数的变化

从前面的实验可以看出,优化后的 DHT 协议安全性更高。为了验证增强安全性的同时,系统仍然具有较高的 DHT 查询效率,本研究在无攻击的环境下统计了改进前后 DHT 网络中成功查询的平均跳

数。如图 9 所示,责任区间内节点数分别为 2 和 16,分别采用 1 到 3 路并行查询策略。可以得到如下结论:(1)责任区间内节点数越多,平均查询跳数越小,查询效率越高。(2)改进后的协议,平均查询跳数与查询并行度无关。这是由于改进后算法的查询路径相互独立。(3)改进后的协议查询效率略低于原始协议,但对于用户体验来说影响不大。

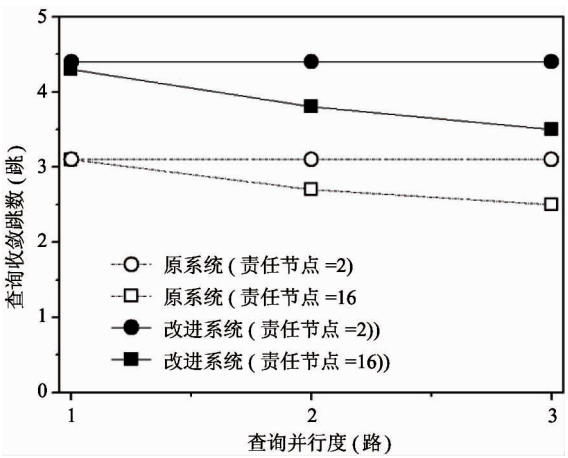

图 9　不同条件下 DHT 查询收敛的平均跳数

图 10 ~ 图 12 给出了网络中攻击节点所占比例变化时,不同参数下改进前后 DHT 系统中的查询成功率。可以得出如下结论:(1)攻击节点比率越高,查询成功率越低。(2)随着网络中总查询次数的增加,改进后协议的查询成功率明显增加,即使攻击节点在网络中占比超过 60% 时,查询成功率也能达到 65% 以上。

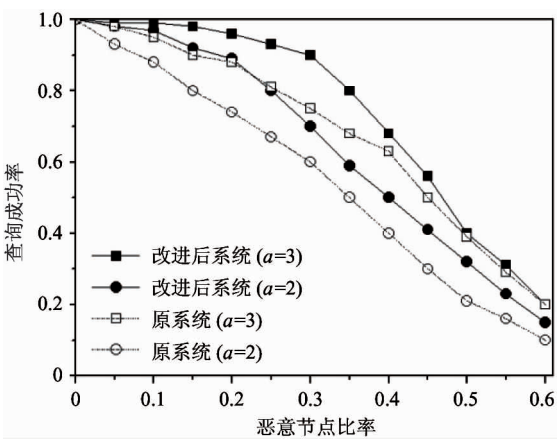

图 10　不同攻击节点比率下查询成功率(100 次查询后)

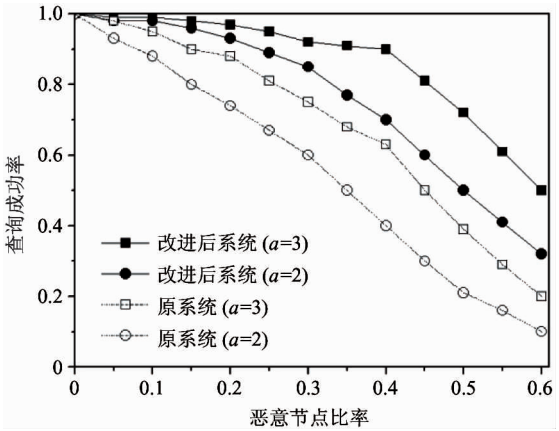

图 11　不同攻击节点比率下查询成功率(500 次查询后)

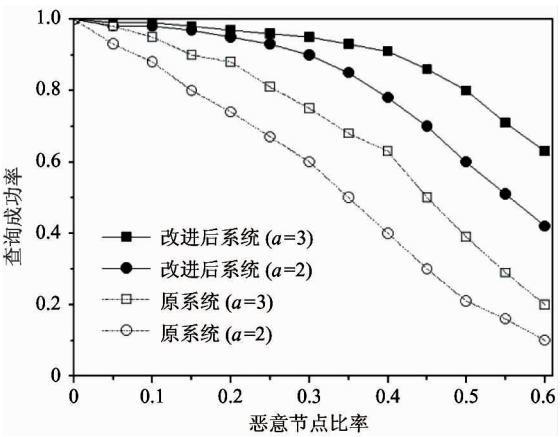

图 12　不同攻击节点比率下查询成功率(1000 次查询后)

当然,仿真实验中的攻击场景更为极端,实际环境中攻击节点很难达到这么高的占比,同时由于采用了基于公私钥机制的 ID 生成算法,攻击者更难构造足够多满足攻击条件的节点,可以说优化后的 DHT 系统具有很好的安全性。

5　结　论

DHT 网络在 P2P 文件共享系统以及新一代网络体系结构研究中都有重要的应用价值。DHT 系统的主要功能是资源的发布和检索,其基本操作和节点的路由表紧密相关,但传统的 DHT 协议缺少安全性方面的设计。攻击者可以通过大规模的构造虚假共谋节点进行索引毒害和路由污染攻击,破坏路由查询结果的准确性,甚至可以造成更严重的安全威胁。本文以 BT 的 Mainline DHT 为例开展研究,

相同的分析方法和改进策略同样适用于其他的 DHT 实例。提出的节点 ID 生成机制、路由表节点更新机制以及搜索路径选择机制,能够在不降低 DHT 网络核心服务效率的同时,提高 DHT 系统的安全性,有效抵御共谋攻击和路由攻击,具有重要的实际应用前景。

参考文献

[1] Chaabouni R, Garcia-Lopez P, Sanchez-Artigas M,et al. Boosting content delivery with BitTorrent in online cloud storage services. In: Proceedings of the 13th IEEE International Conference on Peer-to-Peer Computing, Trento, Italy, 2013

[2] Musau F, Guojun W, Shui Y, et al. Securing recommendations in grouped P2P e-commerce trust model. *IEEE Transactions on Network and Service Management*, 2012, 9(4): 407-420

[3] Dannewitz C, Ambrosio M, Karl, Vercellone V. Hierarchical DHT-based name resolution for information-centric networks. *Computer Communications*, 2013, 36(7):736-749

[4] Chaabouni R, Garcia-Lopez P, Sanchez-Artigas M, et al. Boosting content delivery with BitTorrent in online cloud storage services. In: Proceedings of the 13th IEEE International Conference on Peer-to-Peer Computing, P2P'13, Trento, Italy, 2013 .1-2

[5] Chan C, Chan S. Distributed Hash tables: Design and Applications. In: Handbook of Peer-to-Peer Networking. Springer Science, 2010. 257-280

[6] Urdaneta G, Pierre G, Steen V M. A survey of DHT security techniques. *ACM Computing Surveys*, 2011, 43 (43): 1-8

[7] Neil B, Shields L C, Margolin N B. A survey of solutions to the sybil attack. Amherst: University of Massachusetts Amherst, 2006. 1-12

[8] Singh A, Castro M, Druschel P, et al. Defending against eclipse attacks on overlay networks. In: Proceedings of the 11th Workshop on ACM SIGOPS European Workshop, 2004. 1-21

[9] Villanueva R, Villamil M, Arnedo R. Secure routing strategies in DHT-based systems. In: Proceedings of the 3rd International Conference on Data Management in Grid and Peer-to-Peer Systems, Munich, German, 2010. 62-74

[10] Villanueva R, Villamil M. Secure routing DHT: A protocol for reliable routing in P2P DHT-based systems. In: Proceedings of the 7th International Conference on Internet and Web Applications and Services (ICIW 2012), Stuttgart, Germany, 2012. 260-267

[11] Sanchez-Artigas M, Garcia-Lopez P, Gomez A. A novel methodology for constructing secure multipath overlay. *IEEE Internet Computing*, 2005, 9(6): 50-57

[12] Sanchez-Artigas M, Garcia-Lopez P, Gomez A. Bypass: providing secure DHT routing through bypassing malicious peers. In: Proceedings of Symposium on Computers and Communications, Tokyo, Japan, 2008. 934-941

[13] Srivatsa M, Liu L. Vulnerabilities and security threats in structured overlay networks: A quantitative analysis. In: Proceedings of the 20th Annual Computer Security Applications Conference, Tucson, USA, 2004. 252-261

[14] Wang P, Osipkov L, Hopper N, et al. Myrmic: secure and robust DHT routing: [Technical Report]. University of Minnesota-Twin Cities, 2006. 1-12

[15] Roh B, Kwon O, Hong S, et al. The exclusion of malicious routing peers in structured P2P systems. In: Proceedings of the 5th International Workshop on Agents and Peer-to-Peer Computing, Hakodate, Japan, 2006. 43-50

[16] Marti S, Ganesan P, Garcia-Molina H. DHT routing using social links. In: Proceedings of the 3rd International Workshop on Peer-to-Peer Systems (IPTPS'04), La Jolla, USA, 2004. 100-111

[17] Sanchez-Artigas M, Garcia-Lopez P. On routing in distributed hash tables: is reputation a shelter from malicious behavior and churn? In: Proceedings of the 9th International Conference on Peer-to-Peer Computing, Linkoping, Sweden, 2009. 31-40

[18] Cerri D, Ghioni A, Paraboschi S, et al. ID mapping attacks in p2p networks. In: Proceedings of the 2005 IEEE Global Telecommunications Conference (GLOBECOM 2005), St. Louis, USA, 2005. 3-6

Study on the security optimization of DHT systems

Shi Jiantao, Xia Qingquan, Zhang Zhaoxin

(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001)

Abstract

The security vulnerability of distributed Hash table (DHT) systems was studied, a variety of security optimization strategies were proposed, and a prototyhpe system was designed. Real world network experiments were performed, and the results show that existing DHT networks are vulnerable to index poisoning and routing pollution attacks, so the wrong query results caused by this will even lead to a larger network security event. By improving the node ID generation mechanism, the routing table update mechanism and the search path selection mechanism of a DHT system, the study improved the security of the DHT system from all working stages to resist attackers' collusion attack. The desinged prototype system based on these methods can remain the query success rate of more than 65% in the attacking seniro with 60% of collusion attack nodes. The only cost is increasing the average querying hop of less than 1. Thus, the method is applicable to a variety of distributed Hash table structures and has important practical prospects.

Key words: peer-to-peer network, distributed Hash table (DHT), security optimization, routing pollution, index poisoning