

Advanced Node Insertion Attack with Availability Falsification in Kademlia-based P2P Networks

Yeonju Lee^a, Hyelim Koo^a, Seungoh Choi^a, Byeong-hee Roh^a, and Cheolho Lee^b

^aDepartment of Computer Engineering, Graduate School, Ajou University
San 5 Wonchon-Dong Youngtong-Gu, Suwon, 443-749, Korea
{yjlee11, milkystar, uwer1, bhroh}@ajou.ac.kr

^bElectronics and Telecommunications Research Institute
chlee@ensec.re.kr

Abstract— Demand on the exchange of very large sized multimedia data over the Internet is getting higher. P2P overlay networks have been used for the purpose of file sharing to meet the users' demands on their contents exchanges. However, P2P networks have security problems since unspecified users can join the network and publish files. In this paper, we propose a new advanced node insertion attack in Kademlia-based P2P (Kad) networks. To maximize the impact by node insertion attacks, the proposed method falsifies availability values. We implemented the attacks by using an open source program called eMule, and tested the impact of the proposed attack in our own constructed real Kad network test bed.

Keywords— Kademlia, Kad Network, DHT, peer-to-peer (P2P), Node Insertion Attack, Availability Falsification, eMule

1. INTRODUCTION

Nowadays, peer-to-peer (P2P) network has become a popular way for distributed users to share and retrieve information or files. The Kad network is a distributed hash table (DHT) network based on Kademlia[1] for P2P services. The Kad network does not require any official or common servers.

The Kad network improves capacity and reliability of P2P system by reducing the bottleneck usually appeared in file sharing applications. In addition, it is independent of the network size, e.g. number of nodes in the network. It solves the scalability problem when node joins to network newly in early P2P system. However, despite these advantages, it has severe security problems since it allows file sharing among unspecified nodes.

A node insertion attack is one of the possible attacks in Kad networks. In a node insertion attack, an attacker seeks to attract search request for a file, then it answers with bogus information to nodes that search for the file[2]. For a node

insertion attack, the attacker offers polluted distance information, which makes the probability for the searching node to access the attack node preferably. However, the attack by using the faked distance has a weakness. That is, the faked distance gives a chance for the information polluted by the attacker to be appeared as one of the lists displayed on the user's screen. However, it does not guarantee for the user to select the list provided by the attacker.

In this paper, we propose an advanced node insertion attack with availability falsification in the Kad network. The availability is defined as the number of nodes that contain the requested file. The larger the availability is, the higher the probability that user selects the information is. We implemented the proposed attack as well as the ordinary node insertion attack using an open eMule source[3] on our own constructed real Kad network test bed.

The remainder of this paper is organized as follows. In Section 2, we describe about the concept and operation of Kad network and node insertion attacks. Section 3 describes the proposed advanced node insertion attack with availability falsification. In Section 4, some experimental results on the proposed method are given. Finally, the paper concludes in Section 5.

2. BACKGROUND

2.1. KAD Network Overview

The Kad network is a P2P network based on the Kademlia P2P DHT (Distributed Hash Table) routing protocol, which are implemented by several P2P services such as Overnet, eMule, aMule and so on. Each peer in a Kad network has an identifier called KadID of 128 bits. The KadID is randomly generated by using a cryptographic hash function when a client joins to the Kad network at first. The KadID is not changed on its subsequent joins, but it is deleted when the user deletes the application or its preference file.

DHT provides an efficient information lookup service by mapping of data to location for P2P applications. DHT-

This research was partially supported by Basic Science Research Program through the NRF of Korea funded by the MEST (2011-0025544).

enabled systems provide three interfaces such as *put()*, *get()*, and *lookup()*. The interface *put()* is used for publishing a file to the network while *get()* offers a service for users to retrieve contents. Users can find a location where the contents are by using *lookup()*.

The procedure for a peer (or node) to join and share files in a Kad network is briefly given as following: A peer (or node) registers itself to join the Kad network. To get the location of the network, the node can use any one of the following three options: <IP address, port>, Node.dat, and well-known client. After it knows the network location, it can connect to the network. When the node searches a file after joining the network, it sends a request to the closest three nodes in its routing table by using a 128 bits binary keyword. Once the request is received at an intermediate node which knows the node (content-node) that has the content in a tolerance zone, it responds with the information on the content-node. Using the information, the node can request to the content-node to get the file.

The Kad network has potential weakness by several attacks such as Sybil[6] and Eclipse[7] attacks from the following two reasons:

First, nodes in the network accept all incoming messages without any verification mechanism. There is no restriction on the KadID or the IP address of the sender. With a good knowledge on peers' KadIDs and the routing table construction mechanisms, attackers can take controls on legitimate peers. For example, an attacker sends <Hello Request> message to the network, then receivers update their routing tables without any verification. As a result, routing table is polluted, and hence the network can be controlled by the attacker.

Second, nodes don't verify IP addresses. In the Sybil attack [4], an attacker needs many IDs for the attack. Then, the attacker generates numerous KadIDs for an IP address. It means that the attacker can make zombie nodes and controls the network. Kad mechanism does not have any ID management policy. As a result, nodes in Kad network add the attacker as their neighbors without any doubt, and it causes a Sybil attack.

2.2. Node Insertion Attack

Routing in the Kad network is performed based on the distance between two nodes, which is the bitwise XOR of the two identifiers. Let $\underline{a}$ and $\underline{b}$ be the identifiers of the two nodes, then the distance between the two nodes is given by *distance* ($\underline{a}, \underline{b}$) = $\underline{a} \oplus \underline{b}$.

When a user searches for content, it sends the search request message with 128-bits KeywordID, which is the representation of the content, to nodes in its routing table. Then, Kad network provides the identifiers of the nodes that contain the content or know information on the other nodes that have the content in the tolerance zone. The user calculates the distances to the nodes by bitwise XOR operations of the KeywordID and the nodes' identifiers. Fig. 1 shows the file retrieve scenario in a normal situation. With the nodes'

identifiers, the user calculates distances to the nodes. As a result, the user selects the node B with the smallest distance of 18 in Fig.1, and retrieves the file from the node B.

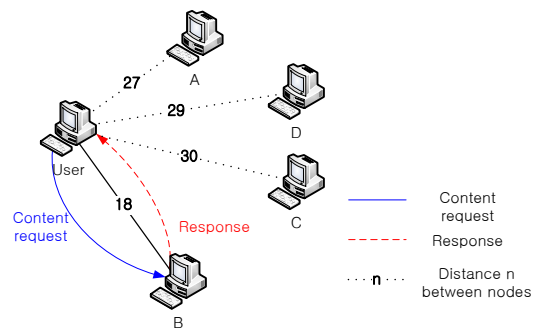


Fig 1. A normal scenario for content retrieval

A node insertion attack can make a normal user impossible to download a file which the user wants. An attacker calculates 128-bit KeywordID for a file that are very popular and published by many different peers before it joins the Kad network. Then, the attacker makes its fake KadID manually such that its KadID matches the hash value of the targeted KeywordID, which makes the distance between the user and the attacker to be shortest. When the user searches the content, the Kad network provides the attacker as the node with the shortest distance as shown in Fig. 2. As a result, the user may request the file to the attacker preferentially. Then, the attacker will receive most of the corresponding search requests from the user, and it gives polluted information on the location of the content.

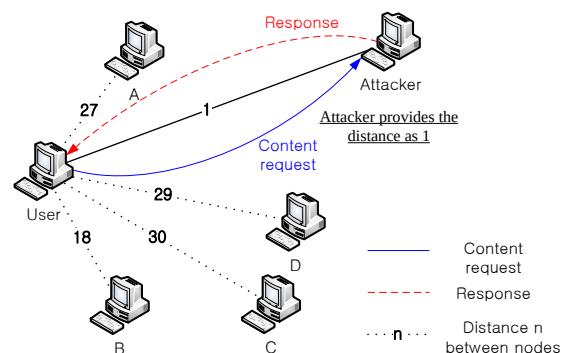


Fig. 2. A scenario for a node insertion attack

3. ADVANCED NODE INSERTION ATTACK

3.1. Weakness of Existing Node Insertion Attack

The node insertion attack illustrated in the previous Section makes the probability for the user to access the wrong peer(s) polluted by the attacker too much high. However, the ordinary node insertion attack by using the faked distance has the following weakness.

Since the user may access not only to the attacker but also to other normal peers according to its selection preference, the user can get contents from normal peers. By the reason, the success rate of existing node insertion attack is too low when there is only one attacker in the Kad network. In [2], the authors showed that it needs more than three attackers in the network to make the success rate of the attack to be larger than 95%. Even though the success rate of the attack is too high, the probability that the user selects one of the faked results may be too low. That is, as a result of the attack, the most responses that the user gets contain the information faked by attackers, but the faked information may not be listed top order of list displayed on the user's screen since the responses returned by a few normal peers may have more preferable information. With the reason, the actual probability that the user selects the faked information may be too low.

3.2. Proposed Method

Here, we explain our proposed advanced node insertion attack with faked availability information. With the proposed method, the attack can be done at a very higher probability that users can access to the attacker than that for an ordinary node insertion attack.

The availability is defined as the number of nodes that contain the requested file. The larger the availability is, the higher the probability that a user selects the information is.

A user sends a KADEMLIA_SEARCH_REQ message to search for a content. If there exists at least one node that has the content or knows the other nodes that have the content, the user receives a KADEMLIA_SEARCH_RES message with METATAGS, which includes the availability information in its SOURCES field as shown in Table 1[5].

Table 1. List of the tags for the metadata

Name	Type	Description
File ID	128-bit	A Hash ID of the file
Client ID	128-bit	Hash ID of the publishing client
FILENAME	Text	The whole name of the file
FILESIZE	Number	The byte size of the file
...		
MEDIA_BITRATE	Number	Bitrate of a media file
MEDIA_CODEC	Number	Codec to read the media file
SOURCES	number	Indicates the availability

In Fig.3, a pseudo code to implement the proposed advanced node insertion attack is given. Before joining the Kad network, the attacker should create its KadID to be matched to the target KeywordID for an ordinary node insertion attack as its preprocess. In addition, the target value for the availability should be found by searching the Kad network with the given KeywordID. Then, a target value larger enough than the provided availability values from the network is chosen and stored for the attack. In the open eMule source [3], the availability value is written to METATAGS of KADEMLIA_SEARCH_RES message at the

CKnownFile::UpdatePartsInfo() module. For the proposed attack, the availability is falsified by the target value found at the pre-process as shown in Fig. 3.

```

Preprocess:
    Given a KeywordID,
    Create KadID;
    Find the target availability;

Execution Process:
    CKnownFile::UpdatePartsInfo(){
        Availability = Get_Availability( );
        Set_Availability(Availability);
    }

```

Fig. 3. Pseudo code to implement the proposed attack

4. EXPERIMENTAL RESULTS

4.1. Experiment Setup

We modified the open eMule client source code of version 0.50 to implement the proposed attack on PCs running Windows XP. Then, we constructed a real Kad network test bed as shown in Fig. 4.

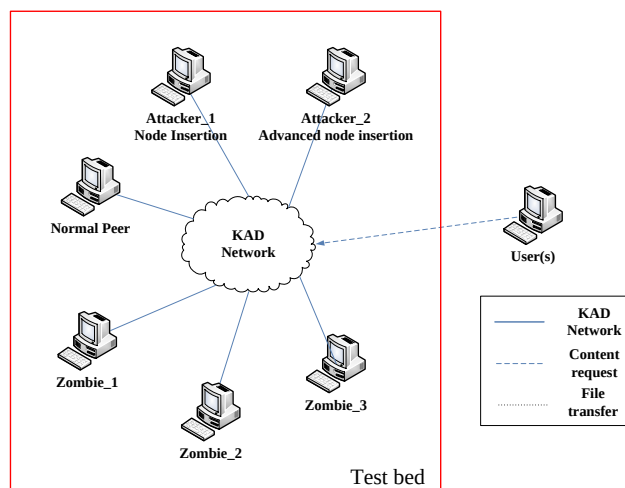


Fig. 4. Test bed for the experiments

The test bed consists of one normal peer, two attackers, and three zombies. The roles of them are as following:

- Normal peer is a peer that has the full original (not corrupted) file to be requested
- Attacker_1 has the file faked from the original and the function of existing node insertion attack.
- Attacker_2 has the file faked from the original and the function of the proposed advanced node insertion attack.
- Each zombie has a fake KadID close to KeywordID for the requested file, and returns the fake response to user's KeywordID search request. In the response, malicious information that the information on the requested file is

stored in both attacker_1 and attacker_2 with corresponding availabilities. It is noted that attacker_2 alters its availability and makes zombies to maintain it.

Any user in the Kad network can access to the test bed through the Internet since the test bed is connected to the Internet. When a user searches the target file with the KeywordID, the zombies inform the user of the nodes' IDs that the file is stored, which are attacker_1 and attacker_2 in our test bed. Normal peer and other peers in the Kad network may provide their response to the user, but since the distances for them are larger than those from zombies, the response may be ignored with very high probability. It is noted that the responses provided by zombies include the availability information on attacker_1 and attacker_2. So, the user may select the attacker_2 with very highest ratio. Then, the attacker_1 is selected with the next highest probability since it has lower availability.

4.2. Results

Fig. 5 shows the search results when the proposed attack is applied to. As shown in Fig. 5, since most P2P applications order the list according to the availability field first, the fake information provided by the attacker is appeared at the top of the list. In general, the top list has the highest probability to be selected as in most of search tools, e.g. google, yahoo and so on.

파일명	크기	가용 소스	완전체 소스	유형	아티스트
2011 Billboard Hot 100 Songs (Top Hits).zip	9.12 MB	739		에 압축	
2011 Billboard Hot 100 Songs (Top Hits).zip	6.82 MB	739		에 압축	
Billboard 2011 Year-End Hot 100 Songs george.o...	6.71 MB	735		에 압축	
2011 Billboard Hot 100 Songs (Top Hits).exe	9.44 MB	733		? 프로그램	
2011 Billboard Hot 100 Songs (Top Hits).rar	3.45 MB	730		에 압축	
Billboard 2011 Year-End Hot 100 Songs george.o...	4.50 MB	728		에 압축	
VA - Billboard Hot 100 Songs [Top Hits] (2011) - S...	6.54 MB	726		에 압축	
2011 Billboard Hot 100 Songs (Top Hits).exe	3.78 MB	725		에 프로그램	
2011 Billboard Hot 100 Songs (Top Hits).exe	4.70 MB	724		에 프로그램	
2011 Billboard Hot 100 Songs (Top Hits).exe	7.66 MB	723		에 프로그램	
Billboard 2011 Year-End Hot 100 Songs george.o...	3.52 MB	723		에 프로그램	
VA - Billboard Hot 100 Songs [Top Hits] (2011) - S...	6.53 MB	723		에 프로그램	
2011 Billboard Hot 100 Songs (Top Hits).zip	3.89 MB	721		에 압축	
2011 Billboard Hot 100 Songs (Top Hits).exe	4.26 MB	721		에 프로그램	

Fig. 5. Example of a search result by the proposed attack

To show the effectiveness of the proposed advanced node insertion attack, we run the test bed for 24 hours, and measured the number of ‘content requests’, which is the request for the retrieval of the file to normal peer, attacker_1, and attacker_2, respectively. The measured counts are shown in Fig. 6. As we can see from Fig. 6, the ratios of the counts for normal peer, attacker_1, and attacker_2 to the total counts are 5.42%, 19.38%, and 75.19%, respectively. That is, the probability that the information provided by the proposed attack is selected is 3.8 times higher than that for the existing node insertion attack.

5. CONCLUSIONS

The node insertion attack, one of the possible attacks in the Kad network, can make normal users very hard to download a file which the users want. In addition, the attack can corrupt

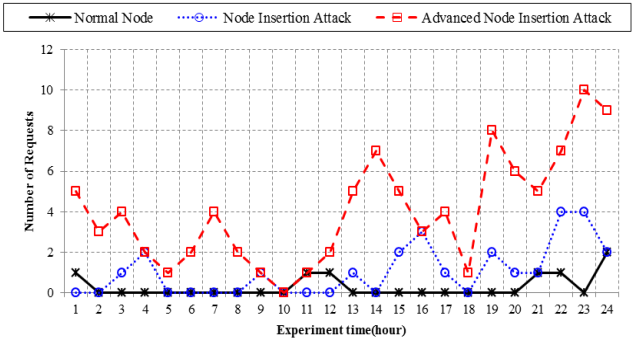


Fig. 6. Number of content requests

the Kad network operation by polluting peers' routing tables. However, the probability that users select the information provided by attackers in existing node insertion attacks may be too low. In this paper, we proposed an advanced node insertion attack with faked availability, which makes the probability that the users select the malicious information provided by attackers to be very high. We showed the effectiveness of the proposed method on the test bed that the real Kad network works by using the open eMule source.

The degree of the use of P2P applications to share files is getting increase exponentially. With the increasing trend of P2P services, it is expected that the threats on P2P network also increase. To secure the P2P networks, we have to find the possible attacks as many as possible and develop the countermeasure schemes on those attacks. The attack proposed in this paper may be one of possible attacks to threaten the secure use of P2P networks. For the further work, we will study on the counteract scheme on the attack.

REFERENCES

- [1] P. Maymounkov and D. Mazieres, "Kademlia: A Peer-to-Peer Information System Based on the XOR Metric," IEEE IPTPS 2002, March 2002
- [2] T. Locher, D. Mysicka, S. Schmid, and R. Wattenhofer. "Poisoning the kad network," ICDCN '2010, LNCS Vol. 5935, pp.195-206, January 2010
- [3] eDonkey network. <http://www.edonkey2000.com>
- [4] T. Cholez, I. Chrisment, and O. Festor. "Evaluation of Sybil attack protection schemes in KAD," AIMS'2009, pp. 70–82, 2009
- [5] R. Brunner and E. Biersack., "A Performance Evaluation of the Kad Protocol," Technical report, Eurecom, 2006
- [6] O. Jetter, J. Dinger, and H. Hartenstein, "Quantitative Analysis of the Sybil Attack and Effective Sybil Resistance in Peer-to-Peer Systems," IEEE ICC 2010, May 2010
- [7] A. Singh, T.-W Ngan, P. Druschel, D. S. Wallach, "Eclipse Attacks on Overlay Networks: Threats and Defenses," IEEE INFOCOM'2006. April 2006 .