Journal of The Korea Institute of Information Security & Cryptology(JKIISC) ISSN 1598-3986(Print)
VOL.23, NO.4, August 2013 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2013.23.4.695

비 자율적 노드 위치 결정을 통한 DHT 네트워크 ID 매핑 공격 방지

이 철 호,[1†‡] 최 경 희,[1] 정 기 현,[1] 김 종 명,[2] 윤 영 태[2]
[1]아주대학교, [2]한국전자통신연구원 부설연구소

Preventing ID Mapping Attacks on DHT Networks through Non-Voluntary Node Locating

Cheolho Lee,[1†‡] Kyunghee Choi,[1] Kihyun Chung,[1] Jongmyung Kim,[2] Youngtae Yun[2]
[1]Ajou University, [2]The Attached Institute of ETRI

요 약

Kademlia와 같은 DHT(Distributed Hash Table) 네트워크는 참여노드들이 네트워크 토폴로지 상에서 자신의 위치를 자율적으로 결정하는 구조를 가지고 있으며 이를 악용한 ID 매핑 공격에 매우 취약하다. 그 결과 DHT 네트워크에서는 eclipse, DRDoS, 봇넷 은닉통신 등의 보안 문제가 지속적으로 발생되고 있다. 본 논문에서는 ID 매핑 공격을 방지하기 위한 비 자율적 노드 위치 결정 방법을 제안하고 NAT 호환성, 공격 저항성, 네트워크 다양성의 세 가지 측면의 분석을 통해 타 방어기법들과 비교를 수행하였다. 분석결과, 제안된 방법은 타 방어기법과 동등한 수준의 공격 저항성을 나타냄과 동시에 타 방어기법이 갖는 단점인 NAT 호환성과 네트워크 다양성 문제를 극복할 수 있음을 확인하였다.

ABSTRACT

DHT(Distributed Hash Table) networks such as Kademlia are vulnerable to the ID mapping attack caused by the voluntary DHT mapping structure where the location of a node is solely determined by itself on the network topology. This causes security problems such as eclipse, DRDoS and botnet C&C on DHT networks. To prevent ID mapping attacks, we propose a non-voluntary DHT mapping scheme and perform analysis on NAT compatibility, attack resistance, and network dynamicity. Analysis results show that our approach may have an equivalent level of attack resistance comparing with other defense mechanisms and overcome their limitations including NAT compatibility and network dynamicity.

Keywords: DHT, Kademlia, overlay network, P2P, ID mapping

I. 서 론

BitTorrent와 Kad로 잘 알려진 Kademlia와 같은 DHT(Distributed Hash Table: 분산 해쉬 테이블) 네트워크는 노드의 ID를 노드 스스로 결정하고 노드의 ID는 다시 DHT 네트워크 토폴로지 상에서 노드 자신의 위치로 인식된다[6][17][28]. 또한, DHT 네트워크에서 특정 리소스(예: 파일, 키워드 등)에 대한 검색은 해당 리소스의 고유 해쉬값과 가장 인접한 곳에 위치한 노드(즉, 가장 인접한 ID를 가진 노드)를 통해서 이뤄지게 된다[17].

이와 같은 DHT의 자율적 위치결정 구조로 인해서 공격자는 임의의 위치에 자신의 공격 노드를 배치하고 악의적인 행위를 수행할 수 있는데 이것을 ID 매핑

접수일(2013년 5월 3일), 게재확정일(2013년 6월 10일)
† 주저자, cheolholee@ajou.ac.kr
‡ 교신저자, cheolholee@ajou.ac.kr(Corresponding author)

공격이라 한다[7]. 예를 들어서, ID 매핑 공격을 통해 특정 위치에서 다른 노드로부터 수신한 메시지에 대해 악의적으로 조작된 응답을 보내서 특정 리소스의 유통을 차단하는 eclipse 공격이나, 조작된 응답에 따라 정상적인 노드들이 특정 IP 주소로 동시 다발적으로 접속하도록 유도하는 DRDoS(Distributed Reflected Denial of Service) 공격을 수행할 수 있다[2][3][4][18][19][20][21].

본 논문에서는 DHT 네트워크에서 발생되는 ID 매핑 공격의 원인과 그 파급효과에 대해서 알아보고 기존에 제안된 타 방어기법을 살펴본다[5][7][12]. 기존 방어기법에 대한 분석을 토대로 한계점을 도출하고 한계점을 극복할 수 있는 새로운 방어기법을 제안한다. 또한, 제안된 기법의 우수성을 증명하기 위해 NAT 호환성, 공격 저항성, 네트워크 다양성에 대한 분석을 수행한다.

본 논문의 2장에서는 DHT 네트워크의 동작과 ID 매핑 공격의 원리를 DHT 네트워크에 하나인 Kademlia의 예를 통해서 살펴본다. 3장에서는 ID 매핑 공격에 대한 타 방어기법을 소개한다. 그리고, 4장에서는 본 논문에서 제안하는 비 자율적 주소 결정 방법을 설명한다. 5장에서는 제안된 방법을 NAT 호환성, 공격 저항성, 네트워크 다양성 측면에서 분석한 결과를 제시한다. 마지막으로, 6장에서는 본 논문을 요약하고 향후 연구 과제를 제시한다.

II. DHT 네트워크에서의 ID 매핑 공격

본 장에서는 DHT 네트워크의 구조와 동작을 DHT 네트워크 중에 하나인 Kademlia의 예를 통해서 살펴보고, ID 매핑 공격 방법과 그 원인을 알아본다. 원활한 설명을 위해서 [표 1]과 같은 기호를 사용하고자 한다.

2.1 DHT 네트워크

구조적(structured) 오버레이 네트워크는 일정한 형태의 네트워크 토폴로지를 갖기 때문에 효율적인 검색이 가능하며 주로 DHT(Distributed Hash Table)의 개념이 사용된다. DHT 네트워크에서의 리소스와 참여노드는 DHT 네트워크 토폴로지 상에서 각각 위치가 결정되며, 토폴로지와 거리계산 방식에 따라 Kademlia, Chord, CAN, Pastry 등 다양한 DHT 네트워크 구조가 제안되어 활용되고 있다. 본 장에서는 DHT 네트워크 중에 하나인 Kademlia의 동작 메커니즘에 대해서 살펴보고자 한다.

Kademlia를 바탕으로 구현된 대표적인 오버레이 네트워크는 BitTorrent와 Kad가 있으며[6][28], 기본적으로 160 비트(BitTorrent는 160비트, Kad는 128비트)의 주소공간을 갖고, 리소스와 노드 또는 두 노드간 거리계산에 XOR 연산이 사용되며 라우팅은 prefix matching 방식을 따른다[17].

먼저, 오버레이 네트워크의 근간이 되는 라우팅 테이블의 구성을 알아보자. 특정 노드 N_A가 노드 N_B로부터 메시지를 수신한 경우 또는 N_B가 N_A로부터 메시지를 수신한 경우 N_A와 N_B는 각각 자신의 라우팅 테이블을 갱신하게 된다. 각각의 노드는 자신이 열어둔 UDP 포트를 통해서 노드 자신과 메시지를 주고받은 타 노드와의 거리를 기준으로 최대 160개의 버켓(bucket)으로 나누어 자신만의 라우팅 테이블을 구성하게 되는데, 각 버켓에는 최대 k개(BitTorrent의 경우 $k=8$[8][10][11])의 타 노드에 대한 정보

[표 1] 기호 정의

기호	정의
IP_x	노드 x의 공인 IP 주소
UDP_x	노드 x의 서비스 (외부) UDP 포트번호 (라우팅 및 검색용)
TCP_x	노드 x의 서비스 (외부) TCP 포트번호 (리소스 송신용)
$x \oplus y$	문자열 x와 y의 합 (concatenation)
$MSB_i(x)$	문자열 x의 상위 i 비트
$LSB_i(x)$	문자열 x의 하위 i 비트
$H(x)$	문자열(또는, 리소스) x의 해쉬값 (160 비트 크기)
N	네트워크를 구성하는 전체 노드의 개수 (약 2^{20}개[26][27])
R	네트워크에 등록된 리소스의 개수 (약 $2^{22} \sim 2^{23}$개[28])
NID_x	노드 x의 ID
RID_x	리소스 x의 ID ($\approx H(x)$)
$NADDR_x$	노드 x의 DHT 네트워크상의 주소
$RADDR_x$	리소스 x의 DHT 네트워크상의 주소
RC_x	노드 x의 DHT 네트워크상의 리소스 검색 담당영역
$d(x,y)$	DHT 네트워크의 노드(또는, 리소스) x와 노드(또는, 리소스) y의 거리

$\langle NID, IP, UDP \rangle$를 저장하게 되고, 타 노드로부터 특정 주소에 대한 라우팅 요청을 받게되면 해당 주소가 속한 버켓 및 인접 버켓에서 노드 리스트(즉, $\langle NID, IP, UDP \rangle$ 집합)를 구성해서 응답하게 된다. 이러한 질의 및 응답 과정을 반복하게 되면 결국 대상 주소에 가장 가까운 노드까지 찾아갈 수 있게 되는 것이다. 다시 말해, Kademlia는 IP 네트워크에서 사용되는 재귀(recursive) 라우팅의 개념이 아닌 반복(iterative) 라우팅의 개념을 가지고 있다[17].

다음으로, 리소스가 공표(publish)되고 검색(lookup)되며 공유(share)되는 과정을 알아보자. Kademlia는 DHT 네트워크 전체를 하나의 해쉬 테이블로 인식하고 특정한 $<key, value>$ 쌍의 정보를 key 값과 인접한 노드에 저장하고 다시 그 노드로부터 그 정보를 추출할 수 있다. 특정 리소스 r의 전부 또는 일부를 소유한 노드 N_{seeder}는 자신이 r을 소유하고 있으며 공유할 준비가 됐다는 사실을 타 노드들에게 알리기 위해서 리소스 r의 고유한 값인 RID_r 와 가장 가까운 곳에 위치한 노드 N_{adj}를 찾아내고(앞서 언급한 반복 라우팅 과정을 통해서 수행됨) N_{adj}에 $\langle RID_r, IP_{N_{seeder}}, TCP_{N_{seeder}} \rangle$를 저장하는 과정을 주기적으로 반복 수행하게 되고(반복 수행하는 이유는 고정된 서버가 없는 DHT 네트워크의 특성상 일정시각 T 의 N_{adj}와 t 만큼 시간이 경과한 $T+t$ 시각에서의 N_{adj}가 서로 다를 수 있기 때문임), 리소스 r을 얻고자 하는 노드 $N_{leecher}$는 마찬가지로 RID_r에 가장 인접한 노드 N_{adj}를 찾아내고 N_{adj}로부터 리소스 r의 위치정보 $\langle RID_r, IP_{N_{seeder}}, TCP_{N_{seeder}} \rangle$를 얻어서 N_{seeder}에 접속함으로써 리소스 r을 얻게 된다.

Kademlia 네트워크는 노드와 리소스를 위한 각각 동일한 크기(160비트)의 개념적인 선형 주소공간을 가지며 모두 DHT 네트워크 주소공간(이하 DHT Key Space)으로 투영(mapping, 이하 매핑)되는데, 노드와 리소스 각각의 주소공간 상에서의 위치는 매핑된 이후의 DHT Key Space에서의 위치와 동일하다. 다시 말해, 임의의 노드 또는 리소스 x에 대해서 주소 매핑함수는 $f(x)=x$로 나타낼 수 있고 $NID_x = NADDR_x$이고 $RID_x = RADDR_x$ 임을 의미한다[3][10][11][17].

[그림 1]의 (가)는 3개의 리소스 (R_1, R_2, R_3)와 4개의 노드(N_1, N_2, N_3, N_4)로 구성된 Kademlia 네트워크를 기하학적으로 나타낸 것이다. 앞서 살펴본 바와 같이, 특정 리소스의 위치를 중심으로 인접 노드 중에서 가장 가까운 노드가 해당 리소스에 대한 검색(lookup) 서비스를 담당하므로 점선 사각형으로 표시된 것과 같이 좌우측 인접 노드간 거리를 양분하여 리소스 검색 담당영역이 결정된다. 즉, 하나의 노드 x는 인접 노드들 간의 거리에 따라 리소스 검색 담당영역(즉, RC_x)이 결정되고 해당 영역 내에 위치한 리소스에 대한 검색 서비스를 제공할 의무를 갖게 된다. 예를 들어서, [그림1]의 (가)에서 리소스 R_2는 노드 N_2의 담당영역인 RC_{N_2}에 포함되므로 N_2가 리소스 R_2에 대한 검색 서비스를 다른 노드들에게 제공해야 한다. 이러한 과정을 통해서, 다수의 리소스를 노드들이 배분하여 분산된 검색 서비스를 제공하는 구조를 가질 수 있게 된다.

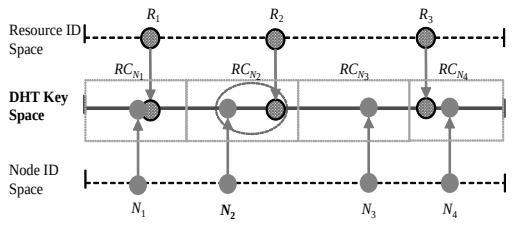
물론, 고정된 서버 없이 노드들이 예고 없이 수시로 네트워크에 진입하고 탈출하는 상황에서는 각 리소스 담당영역 RC가 수시로 변하게 되므로 특정 리소스에 대한 검색 서비스 제공 주체도 함께 수시로 변하게 된다.

이러한 특성을 네트워크 다양성(dynamicity)이라 하며 DHT 네트워크는 기본적으로 네트워크 다양성을 만족하면서도 안정적인 동작을 보장할 수 있어야 한다[5][25]. 만약, 어떤 리소스의 검색 서비스 권한을 점유한 노드가 성능이 저하되거나 올바르게 동작되지 않게 될 때에는 다른 정상적인 노드가 해당 리소스의 검색 서비스 권한을 가져갈 때까지 해당 리소스에 대한 검색 성능이 저하되거나 불가능 해질 수 있다. 따라서, 네트워크 다양성은 DHT 네트워크의 필수 요구조건이다. 본 논문에서는 이러한 네트워크 다양성의 관점에서 방어기법들을 평가한다.

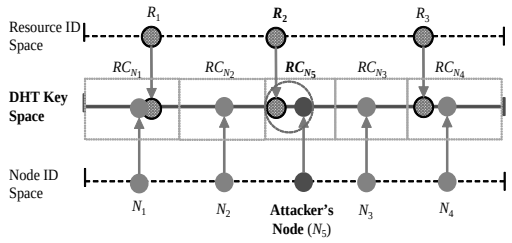
2.2 ID 매핑 공격

[그림 1]은 리소스 R_2를 기준으로 정상적인 상황과 ID 매핑 공격이 발생한 상황을 각각 (가)와 (나)로 나누어 도식화한 것이다. 앞서 설명한 것과 같이, 정상적인 상황에서는 $R_2 \in RC_{N_2}$이므로 리소스 R_2에 대한 검색(lookup) 서비스를 N_2가 제공하게 된다.

그러나 악의적인 공격자 노드 N_5가 리소스 R_2에 좌우로 인접한 노드 N_2와 노드 N_3보다 더 R_2에 가까운 위치를 점유한다면(즉, $d(R_2, N_5) < d(R_2, N_2)$를 만족하면), [그림 1]의 (나)와 같이 $R_2 \in RC_{N_5}$이 되므로, 이후 리소스 R_2에 대한 검색 서비스 권한을 공격자 노드 N_5가 가질 수 있게 된다. 이와 같이 공격자가 자신이



(가) 정상적인 DHT 네트워크



(나) ID 매핑 공격

〔그림 1〕 Kademlia DHT 네트워크에서 발생되는 ID 매핑 공격의 예

원하는 주소에 자신의 노드를 위치시켜서 특정한 리소스에 대한 검색 서비스 권한을 탈취하는 것을 ID 매핑 공격이라 한다〔5〕.

ID 매핑 공격은 공격 노드의 네트워크 토폴로지 상의 위치를 공격자가 임의대로 선택할 수 있다는 점에서 한명의 공격자가 두 개 이상의 노드를 네트워크에 참여시키는 Sybil 공격과는 차별되는 공격 방법이다〔1〕〔3〕〔5〕. 좀 더 자세히 설명하면, 기본적으로 Kademlia는 $NID_x = NADDR_x$이 성립하므로 한명의 공격자가 서로 다른 ID를 갖는 두개 이상의 노드를 공격자가 의도한 위치에 진입시킬 수 있으므로 Sybil 공격과 ID 매핑 공격이 엄격히 구분되지 않는다. 그러나 노드의 DHT Key Space 상에서의 주소를 노드 스스로 결정할 수 없는 상황이라면 아무리 많은 수의 노드(즉, Sybil)를 네트워크에 참여시킨다 하더라도 160 비트 주소공간에서 공격자가 의도한 주소에 정확히 노드를 위치시키는 것은 매우 어려울 것이다. 따라서 본 논문에서는 ID 매핑 공격을 DHT Key Space 상에서 의도적으로 특정한 주소에 노드를 위치시키는 것으로 본다는 점에서 Sybil 공격과는 다른 개념으로 이해되어야 한다.

ID 매핑 공격은 단순히 원하는 주소에 의도적으로 공격 노드를 위치시켜 특정한 자원의 검색 서비스 권한을 빼앗는데서 그치지 않고 2차적인 공격을 가능하게 한다. 예를 들어서, 특정 리소스에 대한 유통을 차단할 목적으로 검색 서비스 요청에 대해서 응답하지 않거나 잘못된 응답을 전송하는 eclipse 공격이 가능하다〔3〕〔4〕. 또한, 매우 인기 높은 리소스의 서비스 권한을 획득해서 다른 참여노드의 서비스 요청에 대한 응답으로 임의의 IP 주소와 TCP 포트번호를 전송하게 된다면 해당 서비스를 요청했던 노드들은 응답 메시지에 포함된 IP 주소를 대상으로 TCP 연결을 시도하게 되므로 DRDoS 공격을 유발시킬 수 있다〔2〕〔3〕〔21〕.

뿐만 아니라, 공격자와 악성 봇(bot)간의 은닉 통신을 위해서 미리 상호 약속된 DHT Key Space 상의 주소를 기준으로 인접한 다른 정상 노드에 공표(publish)하는 방법으로 공격자의 명령 데이터를 삽입한 후 다시 악성 봇이 해당 위치에서 명령 데이터를 가져가는 방식의 은닉통신이 가능하다〔18〕〔19〕〔20〕. 이러한 통신방법은 약속된 기준위치를 수시로 변경할 수 있다는 점과 정상적인 다른 노드의 개입으로 인해서 공격자의 IP 주소를 밝혀내기 매우 어렵다.

III. 관련연구

본 장에서는 ID 매핑 공격을 방지하거나 ID 매핑 공격 방지에 효과가 있는 것으로 알려진 선행 연구들을 살펴본다. 관련된 선행 연구들은 크게 세 가지로 분류될 수 있다. 첫째, 중앙 서버에 의해서 신뢰된 노드 ID를 발급하는 방법이 있다. 둘째, ID 결정에 대한 약속을 규정하고 그에 따라 스스로 노드 자신의 ID를 결정하고 해당 기준에 맞게 ID를 결정했는지 타 노드들이 검증하는 방법이 있다. 셋째, ID 매핑 공격이 성공하기 위한 전제조건들을 까다롭게 구성함으로써 공격자의 공격비용을 높여 공격을 방지하는 방법이 있다.

본 논문에서 제안하고자 하는 비 자율적 노드 위치 결정은 방어기법의 분류에 있어서 두 번째와 세 번째 방법에 연관되어 있으므로 본 장에서는 두 번째와 세 번째 방법을 위주로 살펴보기로 한다. 〔표 2〕는 두 번째와 세 번째 방법에 해당하는 기법들을 비교 분석한 것으로, 관련된 연구들에서 제시된 기법 여섯 가지를 순수 Kademlia와 비교하여 RW_1 부터 RW_6 까지 나타냈으며, 이후 해당 기법들을 지칭하는 이름으로 사용하였다. 참고로 Kademlia는 임의의 노드 또는 리소스 x에 대해서 $NID_x = NADDR_x$ 및 $RID_x = RADDR_x$이 성립하지만 몇 가지 선행 연구들

〔표 2〕 ID 매핑 공격 방어기법 요약

기법명 (가칭)	ID 결정	DHT 네트워크 주소 결정	문제점
순수 Kademlia	NID_x 제약 없음	$NADDR_x = NID_x$	ID 매핑공격 발생
RW_1〔12〕	$NID_x = H(IP_x)$	$NADDR_x = NID_x$	동일 NAT에 속한 노드들의 ID 중복 (네트워크 불안정)
RW_2〔12〕	$NID_x = MSB_t(H(IP_x)) \oplus LSB_{160-t}(H(UDP_x))$ (단, $1 \le t < 160$)		Symmetric NAT 노드의 ID가 접촉노드별로 상이함 (네트워크 불안정)
RW_3〔7〕	$NID_x = H(IP_x \oplus LSB_p(UDP_x))$ (단, $1 \le p < 16$)		
RW_4〔5〕	NID_x 제약 없음	① 공표(publish) 및 검색(lookup) $NADDR_x = H(\langle NID_x, IP_x, UDP_x \rangle)$ ② 라우팅 테이블 $NADDR_x = NID_x$	
RW_5〔7〕	RID_x 제약 없음	$RADDR_{(x,T)} = H(\langle RID_x, T \rangle)$ (단, T 는 일정 주기마다 변경)	시간동기화 및 T 전환시 검색 중복
RW_6〔7〕	NID_x 제약 없음	$NADDR_{(x,T)} = H(\langle NID_x, T \rangle)$ (단, T 는 일정 주기마다 변경)	시간동기화 및 T 전환시 부트스트랩 (네트워크 불안정)
본 논문	NID_x 제약 없음	(수식 1) 참조	-

(RW_4, RW_5, RW_6)과 본 논문의 제안기법은 $NID_x \neq NADDR_x$ 또는 $RID_x \neq RADDR_x$ 이 되므로 〔표 2〕에서는 ID 결정과 DHT 네트워크 주소 결정을 구분하였다.

3.1 신뢰된 노드 ID 발급

휴대폰 단문메시지(SMS)의 발송과 응답을 통해서 참여 노드의 ID를 발급하는 방법이 제안되었다〔3〕. 이 방법은 사전에 중앙 서버에 사용자 등록절차를 거쳐야 하며 이때 휴대폰 번호를 함께 기재하게 된다. 이후 네트워크에 참여할 때 발급되는 ID는 해당 사용자의 휴대폰 번호와 서로 연계되는 값을 가지도록 해서 하나의 휴대폰 번호에 대해서 한 개의 ID만 발급하고 해당 ID는 휴대폰 번호 값과의 연산을 통해서 중앙 서버에서 결정한 후 발급하므로 ID 매핑 공격을 방지할 수 있다. 반면에, 중앙 서버로부터 노드에게 발급하는 공개키(public key)의 해쉬값을 노드의 ID로 사용하고 다른 노드들에 의해서 올바른 ID인지 검증하는 방법도 있다〔9〕.

이러한 방법들은 매우 강력한 공격 방지효과를 가질 수 있지만, 중앙 서버에 의존하므로 중앙 서버가 동작하지 않거나 접속되지 않을 경우 DHT 네트워크 전체가 동작하지 않는 단점도 함께 가지고 있다.

3.2 자기 제한적(self-constrained) ID 결정 및 타 노드에 의한 검증

DHT 네트워크에 진입하고자 하는 노드 N_A가 자신의 ID를 결정할 때 타 노드들이 N_A에 대해서 네트워크 수준에서 즉시 확인할 수 있는 정보(예: IP 주소, UDP 포트번호 등)를 이용한 연산을 통해서 노드의 ID 값을 결정하도록 하고, 노드 N_A와 접촉하는 (즉, 메시지를 서로 송신하거나 수신하는) 타 노드 N_B는 N_A의 ID가 올바르게 결정되었는지 확인할 수 있다. N_B가 N_A의 ID를 올바른 것으로 판단하면 N_B의 라우팅 테이블에 N_A에 관한 정보를 추가하거나 갱신하게 된다. 하지만, 올바르지 않은 ID로 판단한다면 N_B는 N_A를 정상적인 노드로 인정하지 않게 되고 N_B의 라우팅 테이블에 N_A를 추가하지 않으며 N_A에게 어떠한 메시지도 전송하지 않고 N_A로부터 어떠한 메시지도 수신하지 않는다. 결과적으로 약속된 규칙에 따라 자신의 ID를 결정한 노드들만이 정상적으로 DHT 네트워크에 진입할 수 있게 되는 것이다. 물론, ID 검증에 소요되는 오버헤드가 있지만 무시할만한 수준이다. 이러한 ID 결정 방법을 도입하면 네트워크 정보(예: IP주소, UDP 포트번호 등)에 따라 노드 ID가 결정되기 때문에 ID 매핑 공격을 방지하는데 효과가 있다.

Wang 등은 노드의 공인 IP 주소를 해쉬 연산을 통해서 노드 ID로 사용하는 방법을 제안했다[12]. 이 방법은 [표 2]에서 RW_1로 나타냈다. 또한, 노드의 공인 IP 주소의 해쉬값에서 MSB_t와 노드의 UDP 포트번호의 해쉬값에서 LSB_{160-t}를 합쳐서 노드의 ID로 사용하는 방법이 제안되었고[12], 본 논문에서는 RW_2로 나타냈다. Cerri 등은 노드의 공인 IP 주소값에서 UDP 포트번호의 LSB_p(단, $p<16$) 값을 합친 문자열의 해쉬결과를 노드 ID로 사용할 것을 제안했으며[7], 이 방법은 RW_3로 언급하기로 한다. 여기에서 UDP 포트번호가 16 비트 크기를 갖지만 16 보다 작은 값을 취한 것은 Sybil 공격을 완화시키기 위한 것으로 볼 수 있다.

현재, 전 세계적인 IP 주소 자원의 부족과 보안 강화로 인해 NAT 기반으로 네트워크를 구축하는 것이 일반화된 상황이며, 지난 2009년에 발표된 P2P 클라이언트에 관한 통계 연구에 따르면 실제로 네트워크에 참여하는 동시접속 노드 수 $N(\approx 2^{20})$개 중에서 약 73%($\approx 2^{20}\times 0.73 \approx 765{,}000$ 노드)가 NAT 환경에서 동작하고, 전체의 약 16%($\approx 2^{20}\times 0.16 \approx 167{,}000$ 노드)는 symmetric NAT 환경에서 동작하고 있다[13]. 이러한 NAT 환경 의존성은 갈수록 심화될 것으로 예상된다. 따라서, ID 매핑 공격에 대한 방어기법도 NAT 환경에 대한 충분한 고려가 선행되어야 한다. 다시 말해, NAT traversal 문제는 논외로 하더라도 NAT 게이트웨이 내부에 위치한 노드가 스스로 자신의 외부 IP 주소와 UDP 포트번호를 획득하는 방법을 고려해야 한다는 것이다.

하지만, RW_1, RW_2, RW_3는 IP 주소를 노드 ID의 결정에 사용하므로 NAT 환경에 놓인 노드가 스스로 자신의 공인 IP 주소를 알아내는 데는 어려움이 따른다. 물론, STUN(Session Traversal Utilities for NAT)[30]이나 NAT-PMP[15], UPnP[16]와 같은 표준화된 공인 IP 주소 획득 서비스를 이용할 수는 있지만, STUN은 symmetric NAT 환경을 지원하지 않는다[13][14][23]. 뿐만 아니라, NAT-PMP나 UPnP는 비교적 최신의 기술인 관계로 최신 NAT 장비의 경우만 지원이 가능하다. 결과적으로, NAT 게이트웨이 안쪽에 놓인 노드 스스로 자신의 공인 IP 주소를 성공적으로 획득할 가능성 측면에서 볼 때 RW_1, RW_2, RW_3는 16~73% 가량의 노드를 지원하지 못하므로 올바른 대안으로 볼 수 없다.

더욱이, RW_1는 n개의 노드가 하나의 NAT 게이트웨이 내부에 위치하는 경우 n개의 노드에 대한 ID가 모두 동일하게 되므로 DHT 네트워크의 안정성을 저해할 수 있다. 반면, RW_2, RW_3는 노드 ID 결정시에 UDP 포트번호도 함께 사용하므로 노드 ID 중복 문제는 거의 발생하지 않을 것으로 보인다. 그러나, UDP 포트번호의 사용으로 인해 노드들이 다른 노드의 ID를 검증하는데 있어서 큰 문제점을 야기한다. 왜냐하면, ID 결정에 UDP 포트번호가 사용된다면 NAT 환경에서는 NAT 게이트웨이에서 할당하는 외부(external) UDP 포트번호의 사용을 의미하는데, symmetric NAT의 경우 그 특성상 내부 노드 N_{in}과 외부 노드 N_A의 연결에서 NAT 게이트웨이가 할당하는 외부 UDP 포트번호 $UDP(N_{in},N_A)$는 N_{in}과 다른 외부 노드 N_B의 연결에서 NAT 게이트웨이가 할당하는 외부 UDP 포트번호 $UDP(N_{in},N_B)$와 서로 같지 않으므로, 노드 ID 검증이 실패하게 된다. 따라서, UDP 포트번호를 사용하는 RW_2, RW_3는 symmetric NAT 환경을 고려할 때, 전체노드의 약 16% 가량을 정상적으로 지원하지 못한다. 결론적으로, RW_1, RW_2, RW_3는 NAT 환경을 고려할 때 ID 매핑 공격의 방지기법으로 사용할 수 없다.

3.3 공격비용 증대

만약, ID 매핑 공격을 수행하는데 많은 시간과 노력이 소요된다면 공격자는 ID 매핑 공격을 수행할 수 없게 될 것이다. Cerri 등은 시간의 경과에 따라서 주기적으로 *RADDR* 및 *NADDR*를 변경할 것을 제안했으며[7], 본 논문에서는 각각 RW_5과 RW_6로 언급하기로 한다. RW_5과 RW_6이 도입된다면 공격자는 1건의 ID 매핑 공격을 지속하기 위해서 *RADDR*와 *NADDR*의 변경주기마다 공격을 새롭게 수행해야만 한다. 즉, 변경주기가 짧을수록 공격억제 효과는 높아지지만 빈번한 주소 갱신에 따르는 오버헤드가 발생되는 단점을 가지고 있다. 또한, 모든 노드들이 동일한 시간 값 T를 가지도록 해야 하므로 시간 동기화가 필요하며, 각각 *RADDR*와 *NADDR*가 변경되는 시점에는 네트워크 전체에서 갱신이 완료되기까지 일정시간 동안 리소스 중복검색이 필요하거나 DHT 네트워크의 안정성이 크게 떨어지게 된다. 자세히 설명하면, RW_5의 경우 리소스 r에 대해서 모든 리소스가 인식하는 $RADDR_r$이 갱신되는 동안 리소스 r에 대한 검색을

$T-1$ 시간구간의 리소스 r의 주소 $RADDR_{(r,T-1)}$과 T 시간구간의 리소스 r의 주소 $RADDR_{(r,T)}$에 대해서 중복 수행해야만 올바른 검색이 이뤄진다[7]. 또한, RW_6의 경우는 $NADDR$가 변경되는 시점에 일제히 모든 노드가 부트스트랩 과정을 새롭게 수행해야 하므로 네트워크가 완전히 새롭게 초기화된다. 결과적으로 RW_5는 T 전환 시점에는 중복검색으로 인한 약간의 오버헤드가 따르지만 감내할 만한 수준으로 볼 수 있다. 반면, RW_6는 공격억제 효과에 비해 네트워크 초기화 등의 단점이 더 크다고 볼 수 있으므로 공격 방지기법으로 적합하지 않다.

Yu 등은 노드 x의 주소 $NADDR_x$를 라우팅 테이블 상에서 인식할 때에는 기본적인 NID_x로 사용하고, 리소스에 대한 공표(publish) 및 검색(lookup)시에는 $H(\langle NID_x, IP_x, UDP_x \rangle)$을 사용하는 이원화된 주소체계를 제안했으며[5], 이 방법은 본 논문에서 RW_4로 표기하기로 한다. 하지만, 이 방법은 라우팅 테이블에서 인식하는 주소와 공표 및 검색 시 인식하는 주소의 차이로 인해서 오히려 검색 성능을 저하시키는 악영향을 초래하는 것으로 분석되었다[5]. 뿐만 아니라, RW_4는 앞서 살펴본 RW_2, RW_3과 마찬가지로 노드의 주소 결정에 UDP 포트번호를 사용하므로 symmetric NAT 환경에 놓인 노드의 경우(전체 노드의 약 16%), NAT 게이트웨이가 할당하는 외부 UDP 포트번호가 서로 다른 값으로 할당되어 타 노드들이 해당 노드를 인식하는 주소가 서로 다르게 인식되고, 결과적으로 DHT 네트워크의 일체성(consistency)이 깨지는 문제점을 발생시키므로, NAT 환경을 고려할 때 부적합하다.

IV. 제안

본 장에서는 앞서 3장에서 살펴본 ID 매핑 공격 방어기법의 문제점들을 극복한 비 자율적 노드 위치 결정 기법을 제안한다.

4.1 가정

본 논문에서 제안하는 방법을 효과적으로 설명하기 위해서 다음 세 가지 사항을 가정하였다.

첫째, 해쉬 함수의 결과 문자열 전체 및 일부는 균등분포(uniform distribution)를 갖는다. 많은 연구에서 네트워크 규모를 평가하거나 공격 저항성을 측정하는데 이와 같은 가정을 사용한다[7][26][27]. 실제로, 한 연구결과에 따르면 MD4 해쉬 함수의 결과 문자열을 노드의 ID로 사용하는 Kad 네트워크에서 노드들이 DHT Key Space 상에서 균등하게 분포됨을 밝혔다[6].

둘째, 공격자가 발신지(source) IP 주소를 위조(spoof)해서 특정 IP 주소를 가장할 수 없다. 실제로, Kad 및 BitTorrent에서는 공격자 노드 $N_{malicious}$가 발신지 IP 주소를 위조해서 정상 노드 $N_{innocent}$에게 UDP 메시지를 전송한다 하더라도 $N_{innocent}$는 다시 $N_{malicious}$에게 메시지를 전송하고 그 결과를 받은 후에 비로소 자신의 라우팅 테이블이 $N_{malicious}$의 정보를 추가 또는 갱신하도록 함으로써 IP 주소 위조를 방지하고 있다[8][29].

셋째, 공격자가 짧은 시간 내에 특정 공인 IP 주소를 획득하거나 해당 IP 주소를 사용하는 호스트의 제어권을 획득하는 것은 불가능하다. 특정 IP 주소를 사용하는 호스트의 제어권을 획득하는 것은 해킹을 통해서만 가능한데 본 논문에서는 이러한 가능성이 없는 것으로 가정하였다. 뿐만 아니라, 공격자가 암시장(black market)을 통해 백만 대 가량의 좀비 PC를 획득한다고 하더라도 전체 IP 주소에서 예약된 주소를 제외한 약 2^{31}개 중에서 약 2^{20}개에 해당되며, 획득한 좀비PC 중에서 공격자가 의도한 IP 주소를 가지고 있을 확률은 약 $2^{-11} = 0.00048$에 지나지 않는다. 참고로, 이후 IP 주소를 지칭할 때에는 특별한 언급이 없는 한 공인 IP 주소를 의미한다.

4.2 비 자율적 노드 위치 결정(이하 *NVNL*)

앞서 2장에서 살펴본 바와 같이, 노드의 주소를 노드 자신이 전적으로 결정하는 자율적 위치결정이 바로 ID 매핑공격이 발생되는 구조적 원인이다. 따라서, ID 매핑공격을 방지하기 위해서는 노드 스스로 자신의 주소를 결정할 수 없도록 해야 한다.

본 논문에서는 (수식 1)과 같이 노드 x의 주소 결정시 타 노드에 의해서 식별가능한 공인 IP 주소(즉, IP_x)를 사용하고 해쉬를 통해서 노드 주소가 균등하게 분포되도록 할 것을 제안한다. 또한, 동일한 NAT 게이트웨이에 속한 서로 다른 노드들이 서로 다른 주소에 위치할 수 있도록 각 노드가 임의의(random) 값을 취한 노드 ID(즉, NID_x)도 사용하게 되는데, 이것 역시 NID_x가 주소 결정에 직접적인 영향을 미치지

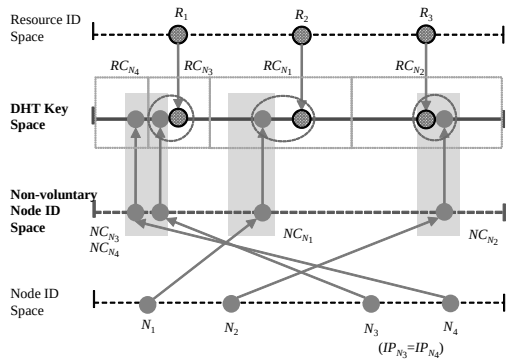
못하도록 해쉬값을 취한다. 최종적으로, (수식 1)과 같이 $H(IP_x)$의 앞부분 α 비트와 $H(NID_x)$의 뒷부분 $160-\alpha$ 비트를 합해서 노드 x의 주소로 사용한다. 다시 말해, 노드 x에 대해서 공인 IP 주소를 통해서 노드가 위치할 수 있는 주소구간(이하 NC_x)이 강제적으로 한정되고, 해당 주소구간 내에서 노드의 정확한 주소는 노드 스스로 제시하는 임의의 값(즉, $H(NID_x)$)에 의해서 결정되는 개념이다. 주소결정에 UDP 포트 번호를 사용하지 않은 이유는, 앞서 3장에서 RW_2, RW_3, RW_4에 대한 설명에서 언급한 바와 같이, UDP 포트를 주소결정에 사용할 경우 symmetric NAT 환경에서 동작하는 노드(전체 노드 중 16%에 해당)를 지원할 수 없기 때문이다.

$$NADDR_x = MSB_\alpha(H(IP_x)) \oplus LSB_{160-\alpha}(H(NID_x)) \quad (1)$$

노드 x가 노드 y와 접촉하면(즉, 메시지를 주거나 받으면) 노드 y는 (수식 1)에 따라 IP_x와 NID_x 그리고 네트워크 전체적으로 약속된 주소결정 변수 α를 사용해서 노드 x의 주소 $NADDR_x$를 계산하고 자신의 라우팅 테이블에 노드 x의 정보 $\langle NADDR_x, NID_x, IP_x, UDP_x \rangle$을 추가(또는, 갱신)한다.

2장에서 살펴본 Kademlia의 기본적인 라우팅 테이블 구성에 따르면 $\langle NID_x, IP_x, UDP_x \rangle$를 사용하도록 되어 있지만, 본 논문에서 제안하는 기법을 지원하기 위해서는 $\langle NADDR_x, NID_x, IP_x, UDP_x \rangle$를 사용하도록 라우팅 테이블 구성에 변경이 따라야 한다. 또한, 비 자율적 노드 위치 결정으로 인해서 더 이상 $NID_x = NADDR_x$의 관계가 성립되지 않으므로 거리 계산 시에는 NID_x가 아닌 $NADDR_x$을 사용해야 한다. 반면, 타 노드가 자신의 주소를 계산할 수 있도록 하기 위해서 노드 간 주고받는 RPC(remote procedure call) 메시지에서는 NID_x를 사용해야 한다.

〔그림 2〕는 *NVNL*의 개념을 기하학적으로 나타낸 것이다. 노드의 ID를 나타내는 공간(즉, Node ID Space)과 네트워크 주소를 나타내는 공간(즉, DHT Key Space) 사이에 (수식 1)과 같이 노드의 IP 주소와 결부시켜 노드 주소를 결정하는 공간(즉, Non-voluntary Node ID Space)을 추가하는 것이다. 이렇게 하면, 모든 노드들은 자신의 IP 주소에 따라서 Non-voluntary Node ID Space에 재배치되고 최종적으로 DHT Key Space에 까지 매핑 된



〔그림 2〕 비 자율적 노드 위치 결정 개념도

다. N_3와 N_4가 동일한 NAT 게이트웨이에 속해 있다고 하더라도 IP 주소는 서로 동일하지만(즉, $NC_{N_3} = NC_{N_4}$), NID_{N_3}와 NID_{N_4}가 서로 다를 것이므로 $NADDR_{N_3}$과 $NADDR_{N_4}$는 서로 다르게 된다. 만약, 공격자가 리소스 R_2를 대상으로 ID 매핑 공격을 하고자 한다면 현재 $R_2 \in RC_{N_1}$ 이므로, 공격자는 $d(R_2, N_{attacker's\ node}) < d(R_2, N_1)$이 성립하는 IP 주소를 획득하거나 최악의 경우 $MSB_\alpha(H(IP_{N_1})) = MSB_\alpha(H(IP_{attacker's\ node}))$가 성립하는 IP 주소를 확보해야만 한다.

즉, 공격자로서는 가능한 많은 수의 IP 주소를 확보해야 하며 최악의 경우 해킹 등의 방법을 통해서 공격자 자신이 특정한 IP 주소를 사용하는 호스트를 점거해야만 ID 매핑 공격이 가능해 지는 것이다. 물론, 주소 결정 변수 α에 따라서 ID 매핑 공격을 성공하기 위한 기대 IP 주소의 개수가 달라질 수 있으므로, 5장에서 α에 따른 공격 저항성, 네트워크 다양성 등을 분석한 결과를 살펴보기로 한다.

V. 평 가

Cerri 등의 연구에 따르면, *NID*의 중복이 없고 *NID*가 랜덤(random)한 값이라면 N 개의 노드가 참여하는 Kademlia DHT 네트워크는 전체 주소공간이 $N+1$ 개의 구간으로 균등분할된 것으로 볼 수 있으므로, 매회 임의로 주어지는 *NID* 값을 가지고 ID 매핑공격을 할 경우 확률적으로 $N+1$회 공격을 시도하면 원하는 구간으로 진입하는 것을 기대할 수 있다〔7〕. 앞서 2장에서 언급한 것과 같이 순수 Kademlia에서는 $NADDR = NID$의 관계가 성립하므

로, ID 매핑공격을 위한 NID 개수의 기대 값 $\|NID\|$은 (수식 2)와 같이 정의된다[7].

$$\|NID\| = N+1 \quad (2)$$

그러나, RW_1, RW_2, RW_3, RW_4와 본 논문의 $NVNL$에서는 주소 결정에서 IP 주소가 핵심적인 요소로 사용되므로, 해쉬 충돌 확률 등의 기타 요소를 고려하지 않는다면 기본적인 공격 저항성을 (수식 3)과 같이 표현할 수 있다. 즉, ID 매핑공격을 성공하기 위해서는 $N+1$개의 IP 주소가 필요함을 의미한다. (수식 2)와 비교할 때, 공격자가 얼마든지 바꿀 수 있는 서로 다른 NID 값을 $N+1$ 개 갖는 것과 서로 다른 공인 IP 주소를 $N+1$ 개 갖는 것은 공격자 입장에서 공격 성공의 조건이 매우 크게 까다로워지는 것을 나타낸다.

$$\|IP\| = N+1 \quad (3)$$

이제 본 논문에서 제안하는 비 자율적 노드 위치 결정의 공격 저항성을 살펴보자. 해쉬함수 SHA-1을 기준으로 $MSB_\alpha(H(IP))$의 충돌확률은 $2^{-\alpha/2}$ 로 알려져 있고[24], 유효한 공인 IP 주소의 개수를 약 2^{31}개로 보면[22], $62 \le \alpha < 160$ 인 경우 확률적으로 $MSB_\alpha(H(IP))$의 값에 충돌이 발생하지 않을 것이다. 즉, $62 \le \alpha < 160$ 이면 전체 주소공간은 최대 $2^{31}+1$ 개의 구간으로 분할될 수 있고 네트워크에 참여하는 노드가 N개라면 주소공간은 $N+1$ 개의 구간으로 분할된다.

하지만, $1 \le \alpha < 62$의 경우는 $MSB_\alpha(H(IP))$ 값에 충돌이 발생될 수 있다. 즉, 서로 다른 IP 주소를 가진 노드 x, y 에 대해서 $IP_x \neq IP_y$ 이고 $MSB_\alpha(H(IP_x)) = MSB_\alpha(H(IP_y))$인 상황이 발생할 수 있는 것이다. 확률 상 $MSB_\alpha(H(IP))$의 값이 동일하게 계산될 수 있는 공인 IP 주소 개수의 기대 값을 $I(\alpha)$로 정의할 때, $I(\alpha)$는 (수식 4)와 같이 계산할 수 있다. 이 경우 해쉬 충돌 확률을 고려하면 N개의 노드로 구성된 네트워크의 주소공간은 $N \times I(\alpha)^{-1}+1$개의 구간으로 분할된다.

$$I(\alpha) = \begin{cases} 2^{31} \times 2^{-\alpha/2} = 2^{31-\alpha/2} & 1 \le \alpha < 62 \\ 1 & 62 \le \alpha < 160 \end{cases} \quad (4)$$

따라서, 본 논문에서 제안한 $NVNL$은 (수식 5)와 같은 공격 저항성을 갖게 된다. 즉, 공격 저항성의 최대값은 $N+1$이며 $1 \le \alpha < 62$ 일 때 α값이 2의 지수 승으로 반영되므로, α값의 감소에 따라 공격 저항성은 기하급수적으로 감소함을 알 수 있다. 따라서, $62 \le \alpha < 160$ 으로 설정하고 최대의 공격 저항성을 갖도록 하는 것이 좋겠다. 하지만, 이 경우 $I(\alpha)=1$ 이 되므로 네트워크 다양성 문제가 발생될 수 있다.

(수식 5)에서 보는바와 같이 $NVNL$에서 $62 \le \alpha < 160$일 때 생기는 네트워크 다양성 문제를 극복하기 위해서 RW_5을 병행할 수도 있는데(즉, $NVNL+RW_5$), 이 경우 일정시간 주기마다 변경되는 T값으로 인해서 리소스의 주소(즉, $RADDR$)가 변경되므로 공격에 필요한 공인 IP 주소도 달라져야 하므로 $NVNL$만 사용할 때에 비해서 더 높은 공격 저항성을 갖게 될 뿐만 아니라 네트워크 다양성 문제도 시간요소인 T의 개입으로 인해서 해결될 수 있다. 즉, 네트워크 다양성 문제를 야기하는 상황이 발생한다 하더라도 일정시간이 경과하면 자연스럽게 해소될 수 있는 것이다. (수식 6)에서 보는바와 같이 일정 주기마다 변경되는 T로 인해서 공격 저항성은 $NVNL$만 단독으로 사용하는 것보다 더 우수하다.

RW_1의 공격 저항성은 (수식 3)과 동일하게 $\|IP\| = N+1$이지만 동일한 NAT 게이트웨이에 속한 노드들의 ID가 모두 동일하므로 네트워크 불안정을 초래하게 되고 네트워크 다양성 문제 또한 발생된다. RW_2는 UDP를 노드 임의로 설정할 수 있고 NID 결정 방식이 본 논문의 $NADDR$ 결정 방식과 유사하므로 (수식 4)와 (수식 5)에서 변수 α를 t로 바꾼 것과 동일하게 공격 저항성을 계산할 수 있겠다. 하지만, 네트워크 다양성 문제에 대한 분석과 그 해결방안이 제시되지 않았고 앞서 설명한 것과 같이 symmetric NAT와 문제가 발생된다. RW_3은 IP 주소와 UDP

$$\|IP\| = \begin{cases} N \times I(\alpha)^{-1}+1 = N \times 2^{(\alpha/2-31)}+1 & 1 \le \alpha < 62 \\ N+1 & 62 \le \alpha < 160 \end{cases} \quad (5)$$

$$\|<IP, T>\| = \begin{cases} N \times I(\alpha)^{-1}+1 = N \times 2^{(\alpha/2-31)}+1 & 1 \le \alpha < 62 \\ N+1 & 62 \le \alpha < 160 \end{cases} \quad (6)$$

〔표 3〕 공격 저항성 및 네트워크 다양성 문제 해결방안 비교

기법명 (가칭)	공격 저항성	네트워크 다양성		기타 문제점
		문제발생 여부	해결방안	
순수 Kademlia	$\|NID\| = N+1$	-	-	ID 매핑 공격 발생
RW_1〔12〕	$\|IP\| = N+1$	발생	제시 안 됨	동일 NAT에 속한 노드들의 ID 중복 (네트워크 불안정)
RW_2〔12〕	$\|IP\| = N \times 2^{(t/2-31)}+1$ (단, $1 \le t < 62$)	발생안함	-	Symmetric NAT 노드의 ID가 접촉노드별로 상이함 (네트워크 불안정)
	$\|IP\| = N+1$ (단, $62 \le t < 160$)	발생	제시 안 됨	
RW_3〔7〕	$\|IP\| = (N+1) \times 2^{-p}$ (단, $1 \le p < 16$)	발생안함	-	
RW_4〔5〕	$\|<NID, IP, UDP>\| = N+1$	알 수 없음	-	
RW_5〔7〕	$\|<NID, T>\| = 1$ (단, T는 일정 주기마다 변경)	발생안함	-	시간 동기화 및 T 전환시 검색 중복
RW_6〔7〕				시간 동기화 및 T 전환시 부트스트랩 (네트워크 불안정)
NVNL	$\|IP\| = N \times 2^{(\alpha/2-31)}+1$ (단, $1 \le \alpha < 62$)	발생안함 ($I(\alpha) > 1$)	-	-
	$\|IP\| = N+1$ (단, $62 \le \alpha < 160$)	발생 ($I(\alpha) = 1$)	RW_5	
NVNL+RW_5	$\|<IP, T>\| = N \times 2^{(\alpha/2-31)}+1$ (단, $1 \le \alpha < 62$) (단, T는 일정 주기마다 변경)	발생안함	-	시간 동기화 및 T 전환시 검색 중복
	$\|<IP, T>\| = N+1$ (단, $62 \le \alpha < 160$) (단, T는 일정 주기마다 변경)			

포트번호의 p비트를 합한 문자열의 해쉬값을 *NID*로 사용하므로, UDP 포트번호를 2^p개 까지 노드가 임의로 할당할 수 있으므로 공격에 필요한 IP 주소의 개수를 2^p로 나눈 것을 공격 저항성으로 볼 수 있다. 이에 반해, RW_4는 *NID*, *IP*, *UDP*가 어떻게 서로 연관되는지에 관해서 언급되지 않았으므로 각 값의 집합인 $\langle NID, IP, UDP \rangle$을 기준으로 공격 저항성은 $\|<NID, IP, UDP>\| = N+1$ 으로 계산될 수 있다. 마지막으로, RW_5와 RW_6은 일정 주기마다 변경되는 T 값과 연동하여 *RADDR* 또는 *NADDR*이 변경되므로 $\|<NID, T>\| = 1$ 의 공격저항성을 갖는다.

관련된 선행연구들에서 제안된 6가지 기법과 본 논문에서 제안하는 기법의 공격 저항성과 도입 시 문제점을 분석한 결과를 정리하면 〔표 3〕과 같다. 전체적으로 볼 때, 본 논문에서 제시하는 *NVNL*은 기존의 방법과 같거나 유사한 수준의 공격 저항성을 가지면서 기존 방법이 가지는 네트워크 불안정 문제를 해소할 수 있다. 특히, 기존 방법 중에서 RW_5는 약간의 오버헤드를 갖지만 *NVNL*과 결합되어 사용될 경우 기존 연구 대비 훨씬 높은 공격 저항성과 네트워크 다양성을 동시에 얻을 수 있을 것으로 판단된다.

VI. 결 론

DHT 네트워크에서 ID 매핑 공격을 방어하기 위해서 본 논문에서 제안하는 *NVNL*(비 자율적 노드 위치 결정 방법)은 기존 방어기법 대비 동등한 수준의 공격 저항성을 갖으면서 기존 방어기법들이 갖는 NAT 호환성 및 네트워크 다양성 문제를 효과적으로 극복할 수 있다. 뿐만 아니라, 기존 방어기법 중 리소스 ID 전환(rotation) 기법〔7〕(본 논문에서는 RW_5로 표기)과 *NVNL* 기법을 함께 사용하는 경우 공격 저항성을 획기적으로 높일 수 있을 것으로 분석되었다. 향후 연구로는 *NVNL* 기법의 네트워크 다양성을 향상시킬 수 있는 방법을 추가적으로 연구하고자 한다.

참고문헌

〔1〕 John R. Douceur, "The Sybil Attack," in Pro. 1st Int'l Workshop on Peer-to-Peer Systems (IPTPS), Mar. 2002.

〔2〕 Zhoujun Li and Xiaoming Chen, "Misusing Kademlia Protocol to Perform

DDoS Attacks," in Pro. IEEE International Symposium on Parallel and Distributed Processing with Applications, pp. 80-86, Dec. 2008.
〔3〕 Moritz Steiner, Taoufik En-najjary, and Ernst W. Biersack, "Exploiting KAD: possible uses and misuses," in Proc. ACM SIGCOMM Computer Communication Review, vol. 37, pp.65-70, Oct. 2007.
〔4〕 Thibault Cholez, Isabelle Chrisment, and Olivier Festor, "Monitoring and Controlling Content Access in KAD," in Proc. IEEE ICC, pp. 23-27, May 2010.
〔5〕 Jie Yu, Zhoujun Li, Peng Xiao, Chengfang Fang, Jia Xu and Ee-Chien Chang, "ID Repetition in Structured P2P Networks," The Computer Journal, vol. 54, no. 6, pp. 962-975, Mar. 2011.
〔6〕 Moritz Steiner, Taoufik En-Najjary, and Ernst W. Biersack, "Long Term Study of Peer Behavior in the KAD DHT," IEEE/ACM Trans. Netw., vol. 17, pp. 1371-1384, Oct. 2009.
〔7〕 D. Cerri, A. Ghioni, S. Paraboschi, and S. Tiraboschi, "ID mapping attacks in p2p networks," in Proc. IEEE Global Communications Conference, pp. 1785 - 1790, 2005.
〔8〕 Andrew Loewenstern, "DHT Protocol," 2008. 〔Online〕. Available: http://www.bittorrent.org/beps/bep 0005.html
〔9〕 Ingmar Baumgart and Sebastian Mies, "S/Kademlia: A Practicable Approach Towards Secure Key-Based Routing," in Proc. P2P-NVE 2007 in conjunction with ICPADS 2007, Hsinchu, Taiwan, vol. 2, pp. 1-8, Dec. 2007.
〔10〕 Arvid Norberg, "libtorrent-rasterbar-0.15.6," 2011. 〔Online〕. Available: http://www.rasterbar.com/products/libtorrent/
〔11〕 Jari Sundell, "libTorrent-rakShasa-0.13.0," 2011. 〔Online〕. Available: http://libtorrent.rakshasa.no/
〔12〕 PengWang, James Tyra, Eric Chan-Tin, Tyson Malchow, Denis Foo Kue, Nicholas Hopper, and Yongdae Kim, "Attacking the Kad Network," in Proc. SecureComm, no. 23, 2008.
〔13〕 L. DAcunto, J. A. Pouwelse, H. J. Sips, "A Measurement of NAT and Firewall Characteristics in Peer-to-peer Systems," in Proc. ASCI Conference, pp. 1-5, 2009.
〔14〕 B. Ford, P. Srisuresh, D. Kegel, "Peer-to-peer Communication Across Network Address Translators," in Proc. USENIX Annual Technical Conference, Apr. 2005.
〔15〕 Stuart Cheshire, Marc Krochmal, and Kiren Sekar, "NAT Port Mapping Protocol (NAT-PMP)," IETF Internet Draft, Jun. 2005.
〔16〕 ISO/IEC 29341, "UPnP Device Architecture," Dec. 2008.
〔17〕 P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the XOR metric." in Proc. Peer-to-Peer Systems, First International Workshop, IPTPS 2002, LNCS, vol. 2429. Springer, pp.53-65, Mar. 2002.
〔18〕 Ping Wang, Lei Wu, Baber Aslam, and Cliff C. Zou, "A Systematic Study on Peer-to-Peer Botnets," in Proc. ICCCN, pp. 1-8, Aug. 2009.
〔19〕 G. Starnberger, C. Kruegel, and E. Kirda, "Overbot - a botnet protocol based on kademlia," in Proc. Security and Privacy in Communication Networks (SecureComm '08), no. 13, Sep. 2008.
〔20〕 Ping Wang, Baber Aslam, and Cliff C. Zou, "Peer-to-Peer Botnets: The Next Generation of Botnet Attacks," in Stavroulakis, Peter; Stamp, Mark (Eds): Handbook of Information and Communication Security, Chapter 18, Springer Press, pp. 335 - 350, 2010.
〔21〕 X. Sun, R. Torres, and S. G. Rao, "On the

feasibility of exploiting P2P systems to launch DDoS attacks," in Proc. Peer-to-Peer Networking and Applications, pp. 36 - 51, 2010.
[22] RFC 5735, "Special Use IPv4 Address," IETF Network Working Group, Jan. 2010.
[23] Yangyang Liu and Jianping Pan, "The Impact of NAT on BitTorrent-like P2P Systems," in Proc. Peer-to-Peer Computing, pp. 241 - 251, 2009.
[24] Quynh Dang, "Recommendation for Applications using Approved Hash Algorithms," NIST Special Publication 800-107 rev. 1, pp. 8 - 11, Aug. 2012.
[25] Daniel Stutzbach and Reza Rejaie, "Understanding churn in peer-to-peer networks," in Proc. ACM SIGCOMM, 2006.
[26] Scott A. Crosby and Dan S. Wallach, "An Analysis of BitTorrent's Two Kademlia-Based DHTs," Department of Computer Science, Rice University, Houston, TX, USA, Tech. Rep., TR-07-04, pp. 20 - 22, 2007.
[27] M. Steiner and E. W. Biersack, "Crawling Azureus," Eurecom, Tech. Rep., RR-08-223, pp. 11 - 12, Jun. 2008.
[28] The Pirate Bay, Jan. 2013. [Online]. Available: http://thepiratebay.org
[29] Thibault Cholez, Isabelle Chrisment, and Olivier Festor, "Evaluation of Sybil Attacks Protection Schemes in KAD," In Proc. AIMS 2009, LNCS 5637, pp. 70-82, 2009.
[30] RFC 5389, "Session Traversal Utilities for NAT (STUN)," IETF Network Working Group, Oct. 2008.

〈저자소개〉

사 진

이 철 호 (Cheolho Lee) 정회원
2002년 2월: 아주대학교 정보및컴퓨터공학부 학사
2004년 2월: 아주대학교 정보통신공학과 석사
2010년 3월~현재: 아주대학교 NCW학과 박사과정
2004년 2월~현재: 한국전자통신연구원 부설연구소 선임연구원
〈관심분야〉 분산 시스템, 정보유출 방지, 악성코드 탐지

최 경 희 (Kyunghee Choi) 정회원
1976년 2월: 서울대학교 학사
1979년 6월: ENSEEIHT DE UNIVERSITY 석사
1982년 3월: Univ Toulouse 3 박사
1982년 3월~현재: 아주대학교 교수
〈관심분야〉 실시간 시스템, 정보보호, 소프트웨어 테스트

정 기 현 (Kihyun Chung) 정회원
1984년 2월: 서강대학교 학사
1988년 5월: University of Illinois at Chicago 석사
1990년 12월: Purdue Univ-West Lafayette 박사
1992년 3월~현재: 아주대학교 교수
〈관심분야〉 임베디드 시스템, 정보보호, 소프트웨어 테스트

사 진

김 종 명 (Jongmyung Kim) 정회원
2007년 2월: 성균관대학교 컴퓨터과학과 학사
2009년 2월: 성균관대학교 전자전기컴퓨터공학과 석사
2009년 1월~2011년 6월: 한국인터넷진흥원 연구원
2011년 7월~현재: 한국전자통신연구원 부설연구소 연구원
〈관심분야〉 악성코드 분석, 소프트웨어 취약점

사 진

윤 영 태 (Youngtae Yun) 정회원
1995년 2월: 충남대학교 컴퓨터과학과 학사
1995년~1997년: 현대전자 정보시스템사업본부
1999년 2월: 충남대학교 컴퓨터과학과 석사
2006년 8월: 충남대학교 컴퓨터과학과 박사
1999년~현재: 한국전자통신연구원 부설연구소 책임연구원
〈관심분야〉 소프트웨어 취약점, 네트워크 보안