情報保護學會論文誌
第17卷 第6號, 2007. 12

랜덤 워크 기반의 P2P 익명 프로토콜*

조 준 하† , 이 현 숙, 박 현 아, 이 동 훈
고려대학교 정보경영공학전문대학원

Peer to Peer Anonymous Protocol Based Random Walk

Jun-Ha Cho† , Hyun-Sook Rhee, Hyun-A Park, Dong-Hoon Lee
Korea University Graduate School of Information Management and Security

요 약

P2P 시스템을 이용하여 파일을 검색하면 상대방의 프로그램에 설정되어 있는 공유폴더의 파일을 검색해서 결과를 보내준다. 이러한 과정에서 보내주는 정보 중에는 경로명 및 파일 정보가 포함되게 되고, 어떤 검색자가 어떠한 정보를 검색했는지 모두 드러날 수 있는 문제점이 발생한다. 이것을 해결하기 위한 방법으로 P2P 익명 파일의 송·수신에 관한 연구가 현재 활발하게 이루어지고 있으나 지금까지의 연구에는 아직 몇 가지 한계점이 있다. 따라서 우리는 그러한 문제점을 분석하고 이를 극복하기 위해 비집중화(Decentralized)되고 비구조화된(Unstructured) P2P 시스템에서 랜덤 워크(Random Walk)를 기초로 하여 파일 요청자가 다른 peer들의 시스템 접속 상태를 모르고도 동적인 Onion 라우팅(Dynamic Onion Routing) 을 가능하게 하는 프로토콜과 멀티캐스트 기법을 이용하여 계산 효율성을 향상시킨 스킴 2가지를 제안한다.

ABSTRACT

The P2P file sharing system sends the results to users by searching the files in the shared folders. In the process of it, the problem is that the transferred information includes the pathname and file information and it can be revealed who searches which files. In related to this problem, anonymous file sharing P2P protocol has been an active research area where a number of works have been produced. However, the previous studies still have a few of weakness. Therefore, We propose two anonymous P2P file sharing protocols based on the decentralized and unstructured Random Walk. The first scheme uses the dynamic onion routing where the requester can receive the wanted file without knowing other peers' IDs. The second scheme uses the IP multicast method which lowers the computational overhead. Both of them are more suited for the dynamic P2P system

Keywords : *anonymity, peer to peer, random walk, dynamic, multicast*

Ⅰ. 서 론

NGN2005(2005년 9월) 워크숍에서 인터넷에서 가장 많은 트래픽을 2003년을 기점으로 P2P 서비스가차지하고 있다는 통계가 발표되었고, 같은 해 Network World지에서도 인터넷 트래픽의 60~89% 정도의 트래픽을 P2P 서비스가 차지하고 있다고 게재된 바 있다[17]. 이처럼 P2P 컴퓨팅 시대로 대표될 수 있는 제 3세대 인터넷 시대가 우리가 생활하고 있는 일상인 것이다.

접수일: 2007년 8월 10일; 채택일: 2007년 10월 18일
* 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT 연구센터 지원사업의 연구결과로 수행되었음 (IITA-2007-(C1090-0701-0025))
† 주저자, wearegoodman@hanmail.net

P2P(peer-to-peer)는 컴퓨터 사이의 직접적 교환을 통한 컴퓨터 리소스를 공유하는 시스템을 말한다. 각 컴퓨터가 동등한 능력을 가지고 있어, 어떤 컴퓨터에서라도 통신 세션을 시작할 수 있는 통신 모델의 특성을 갖기 때문에 동등 계층 통신이라고도 부른다. 그 의미는 네트워크에 연결되어 있는 모든 컴퓨터들이 서로 대등한 동료의 입장에서 데이터나 주변장치 등을 공유할 수 있는 특성을 나타낸다.

이런 P2P 시스템을 이용하여 파일을 검색하면 상대방의 프로그램에 설정되어 있는 공유폴더의 파일을 검색해서 결과를 보내준다. 여기서 보내주는 정보에는 일부 키워드에 경로명 및 파일 정보를 포함하게 되고, 어떤 검색자가 어떠한 정보를 검색했는지 모두 드러날 수 있는 문제점이 발생한다. 이러한 문제점을 해결하기 위한 방법으로 P2P 익명 파일의 송·수신에 관한 연구가 현재 활발하게 이루어지고 있다.[1],[2],[3],[4],[5],[6],[7],[9],[10]

그러나 지금까지의 연구는 각 peer들의 시스템 접속 상태가 서버를 통해 다른 사용자들에게 노출될 수 있는 잠재적인 프라이버시 위협과 계산 비효율성 및 키 전송 시 익명성 노출 등 여러 가지 해결되지 못한 문제를 가지고 있다. 뿐만 아니라 P2P 시스템은 서로 다른 peer들이 끊임없이 확장되고, 이탈과 참여의 변화가 동적이라는 것이 큰 특성중의 하나이다. 즉, 다양한 접속 수단과 다양한 접속 공간 등의 변화에 대응하는 것이 P2P의 근본적인 목적이자 요구조건이다.

따라서 본고에서는 P2P 서비스에서 사용자들의 익명성을 보호하면서 다이나믹한 환경에 적합한 시스템 구현을 위해 다른 peer들의 시스템 접속 상태를 모르고도 동적인 Onion 라우팅(Dynamic Onion Routing)을 가능하게 하는 프로토콜과 멀티캐스트 기법을 이용하여 계산 효율성을 향상시킨 스킴 2가지를 제안한다.

관련 연구. 지금까지 개발된 P2P 파일 공유는 크게 Unstructured P2P 시스템과 Structured P2P 시스템 2가지로 나누어진다. Unstructured P2P는 네트워크에서 peer와 원하는 파일 간에 어떠한 연관성 설정이 없는 구조이고, Structured P2P는 분산 인덱싱을 제공하는 분산 해쉬 테이블(DHT)로 파일과 peer 정보들을 공통의 단일 주소 공간으로 매핑하여 파일을 검색하는 시스템이다[18]. Unstructured P2P 시스템은 예를 들어 Freenet[11], Gnutella[12] 등이 있고 Structured P2P 시스템은 Chord[8], Tapestry[13], CAN[14] 등이 있다. 이 중 Unstructured P2P 시스템이 현재 보다 많은 사용자에 의해 사용되어지고 있는 구조이다.

P2P를 이용한 파일공유에 있어서 사용자의 익명성 문제는 중요한 요소로 부각되고 있고, 최근 Unstructured P2P 시스템 상황하에서 사용자의 익명성을 제공하는 파일 공유에 대한 연구로는 J. Han et al[1]과 Xiao et al[2]의 시스템을 들 수 있다.

[2]는 파일 요청자가 쿼리를 전송하기 전에 Onion을 통해 파일을 되돌려 받기 위해 reply block을 형성한다. 쿼리를 수신하는 각각의 peer는 확률 pv로 쿼리 에이전트 peer로 활동할지를 결정한다. 만일 어떤 peer가 쿼리 에이전트로 활동하는 것을 결정한다면, 그 peer는 이러한 쿼리를 P2P 시스템 상에서 플러딩(flooding)한다. 쿼리에 해당되는 파일 소유자를 찾지 못하면 임의의 peer에서 브로드캐스트를 하여 파일 소유자를 검색한다. 쿼리에 대한 파일이 있는 파일 응답자는 쿼리 에이전트 peer에게 파일을 익명으로 전송하기 위해 또 다른 Onion Path를 형성한다. Onion Path를 통해 파일에 관련된 내용을 수신한 쿼리 에이전트 peer는 reply block을 통해 파일 요청자에게 파일에 관련된 내용을 전송한다. [1]은 키 경로와 암호문 경로를 달리한 랜덤 워크(Random Walk)를 이용하여 voluntary middle peer를 결정한다. 이때 Onion을 통해 원하는 파일을 수신받기 위해 키 경로상으로 전송하는 메시지에 Return Path가 포함되어 있다. voluntary middle peer는 원하는 파일을 검색하기 위해 쿼리를 플러딩(flooding)을 하고, 파일 응답자는 Return Path를 통해 파일 요청자에게 파일에 관련된 내용을 전송한다. 지금까지 우리는 본 논문에서 다루고자하는 Unstructured P2P 시스템에 관한 연구를 살펴보았다. 하지만 이 분야의 연구에는 앞에서도 언급했듯이 아직 해결하지 못한 몇 가지 한계점이 있다. 다음에서 위의 논문 [1],[2]를 자세히 분석해 보기로 한다.

1. 환경적인 측면

① peering node list를 제공하는 server의 존재는 사용자의 프라이버시를 보장할 수 없다. [1], [2]가 reply block 혹은 Onion 구성을 가능하게 한 요인 중의 하나는 I(파일 요청자)와 R(파일 응답자)이 시스템에 접속해 있는 모든 peer들을 peering node list를 제공하는 server를 통해서 알 수 있고, 그에 따라 PKI 상황 하에 peer들의

공개키를 이용하여 Return Path를 구성하기 때문이다. 각각의 peer가 시스템에 접속한 peer의 모든 정보를 알 수 있는 환경은 사용자 프라이버시의 잠재적인 위협이 된다.

② 파일의 반환 경로가 고정되어 있어 이탈과 참여의 변화가 역동적인 P2P 환경의 기본적인 요구 조건을 만족시키지 못함으로써 문제가 발생한다. 즉, reply block 혹은 Onion이 파일 탐색전에 미리 구성되어 있어 Onion의 구성이 다이나믹하지 못하다는 것이다. 만일 파일 탐색 중에 이전에 형성된 reply block 혹은 Onion상의 하나의 peer라도 시스템에서 탈퇴를 한다면 R이 전송한 파일을 I가 수신하지 못할 것이다. 그리고 R과 RP상의 첫 번째 peer의 거리가 멀다면 R이 RP상의 첫 번째 peer를 찾아서 I에게 메시지가 수신될 수 있도록 하는데 있어서 비효율적인 면이 관측된다.

2. 보안적인 측면

① voluntary middle peer A이후 flooding 검색하는 peer들은 PKI 상황하에서 K_{I+}(I의 공개키)를 포함하는 메시지를 수신하고 또한 전송하게 된다. 그렇다면 flooding 검색 경로상의 peer들은 I가 파일을 찾고 있다는 것을 알게 된다. 이것은 결국 파일 요청자의 익명성을 보장하지 못하게 됨을 의미한다.

기여도. 이 논문에서 우리는 비구조화된 그리고 비집중화된 랜덤 워크를 기초로 하는 P2P 시스템에서, PKI구조를 가정하지 않는다. 기존의 논문에서는 파일 요청자가 RP를 구성하기 위해서 RP구성에 포함될 peer들의 공개키를 알 수 있어야 했다. 하지만 우리의 논문에서는 동적으로 Onion을 이용한 RP를 구성하면서 파일요청자가 다른 peer들의 공개키로 RP를 구성하는 것이 아닌 각각의 peer들이 자신들이 선정한 공개키로 RP를 구성하게 된다. 따라서 기존 논문과는 다르게 PKI 구조를 가정하지 않는 장점을 갖는다. 즉, 우리의 논문에서 사용되는 공개키는 공개키의 주인만 알면 되는 암호화 키인 것이다.

1.1. 다이나믹한 RP 설정

① 첫 번째 스킴인 Dynamic Onion Pure P2P Protocol에서 파일 요청자가 키 경로와 암호문 경로를 달리한 랜덤 워크를 이용하여 voluntary middle peer를 선정 시, 암호문 경로 상에서 파일 요청자를 시작으로 voluntary middle peer까지의 경로를 Onion으로 구성함으로써 dynamic하게 RP(Return Path)를 설정하여 RP가 사전에 peering node list를 제공하는 server를 통해 미리 정해져 있을 때보다 파일을 요청하는 사용자가 원하는 파일을 수신할 수 있는 확률을 증가시켰다. 또한 파일 응답자의 파일 전송 시 voluntary middle peer가 RP상의 첫 번째 peer를 검색하는 시간도 없기 때문에 이에 따른 파일 전송의 효율성이 향상되었다.

② Peer들의 시스템 접속 상태 및 그들의 ID 리스트가 없이도 반환 경로를 설정할 수 있게 함으로써 사용자들의 프라이버시를 보장할 수 있었다.

1.2. IP 멀티캐스트 기법

Onion 라우팅 기술의 사용은 peer들의 계산량을 증가시킨다. 이에 대한 해결방안으로 두 번째 스킴인 IP Multicast Pure P2P Protocol에서는 IP 멀티캐스트 기법을 사용한다. IP 멀티캐스트 기법에 대한 사항은 3장에서 다루기로 한다. 파일 요청자는 멀티캐스트를 할 IP를 임의로 생성하여 랜덤워크 메시지를 전송하고 다음의 peer들은 voluntary middle peer까지 IP 멀티캐스트 그룹에 참가할지를 결정하게 된다. 파일 검색단계 후에 파일 응답자로부터 파일에 관계된 내용을 수신한 voluntary middle peer가 IP 멀티캐스트 방법으로 파일에 관계된 내용을 전송하므로 효율적으로 파일을 검색・수신 할 수 있게 되는 것이다. 이 방법은 파일 요청자, 응답자, 파일 요청자와 응답자 상호간의 익명성을 유지하면서 peer들 각각이 Onion Routing 할 때의 공개키 계산이 요구되지 않는다. 뿐만 아니라 파일 요청자가 포함된 다수의 멀티캐스트 그룹에게 전송하므로 P2P 시스템의 역동적인 환경에 적합한 방법이다.

1.3. 전송간 익명성 향상

① [1]과 [2]에서는 파일 요청자가 생성한 공개키가 voluntary middle peer A 이후에 그대로 노출

되어 파일요청자의 익명성을 보장하지 못하였으나, 제안 프로토콜은 PKI상의 파일 요청자의 공개키가 아닌 파일 송·수신 목적의 공개키를 해쉬 함수를 통해 설정함으로써 파일 요청자의 익명성을 보장할 수 있었다.

② 브로드캐스트(broadcast)된 파일에 대해 파일 응답자가 파일을 전송할 때 파일응답자의 익명성보호를 위해 IP를 난수 혹은 암호화 처리하여 파일에 해당하는 내용을 전송하도록 하였다.

Ⅱ. 구성

이 장에서는 본 논문의 전개를 위한 기본적인 사항과 보안 요구 사항인 익명성을 정의한다. 3장에서는 우리의 논문에서 이용하는 그누텔라, 랜덤워크와 IP 멀티캐스트의 개념에 대해 설명하고, 4장과 5장에서는 우리가 제안하는 프로토콜인 Dynamic Onion Pure P2P Protocol과 IP Multicast Pure P2P Protocol을 설명한다. 6장에서는 제안 프로토콜들을 분석하고 7장에서 결론을 맺는다.

2.1. 기호 표시

1. F: 요청 파일
2. f: query 파일
3. I: 파일 요청자
4. R: query 응답자
5. SN: sequence number
6. $\{\ \}_K$: K를 이용한 공개키 암호화
7. E_K, D_K: K를 이용한 대칭키 암·복호화
8. H: 해쉬 함수
9. K_{2+}: peer P_2의 공개키
10. K_{2-}: peer P_2의 개인키

2.2. 익명성(Anonymity) 정의

정의 1.

$$d_{x,e}(A) = \sum_{y \in S \neq x} \Pr_e(y) = 1 - \Pr_e(x) \quad (1)$$

여기서 $\Pr_e(x)$는 entity x가 entity e에 대해서 파일 요청자일 확률이고, x가 집합 S(프로토콜 A에서 익명성이 유지되어 있는 entity들의 집합)의 멤버라고 하면, $\Pr_e(x)$는 $\sum_{y \in S} \Pr_e(x)$에 해당된다. $d_{x,e}(A)$는 어떤 entity x가 entity e에 대해 어떤 특정 프로토콜 A에 의해 익명화된 정도를 나타낸다고 하면, entity x의 익명화 정도는 위와 같은 식으로 나타낼 수 있다[6].

정의 2.

$$d(A) = \min\{(d_{x,e}(A))\},\ \forall e \in E,\ \forall x \in S \quad (2)$$

$d(A)$는 서로 공모하는 entity들에게 있어서 어떤 특정 프로토콜 A에서의 익명성 정도를 나타내는 것으로 위와 같이 정의할 수 있다. 여기서 E는 네트워크에 있는 모든 entity들의 집합이다[6].

[15]의 정의와 [16]의 논문을 참조하여 [6]은 P2P 파일 공유 시스템에서의 익명성의 정도를 다음과 같이 확장하였다.

정의 3.

1. Provably exposed: 공격자는 x가 파일 요청자 혹은 응답자라는 사실을 입증할 수 있다. 즉, $d_{x,e}(A)=0$.

2. Exposed: x가 파일 요청자 혹은 응답자가 아닐 수도 있다. $0 < d_{x,e}(A) < 0.5$

3. Probable Innocence: x가 확신할 수는 없지만 파일 요청자 혹은 응답자가 아닌 것 같고, 다른 entity들 보다는 파일 요청자 혹은 응답자일 확률이 낮다. $\forall y \neq x \in S$에 대해, x는 다음을 만족한다.

$$0.5 < d_{x,e}(A) < d_{y,e}(A),\ d_{x,e}(A) < 1 - \frac{1}{|S|} \quad (3)$$

4. Beyond Suspicion: x는 어떠한 다른 entity들보다 파일 요청자 혹은 응답자일 확률이 낮다.

$$|S| > 1, 1/|S| \leq d_{x,e}(A), d_{y,e}(A) \leq d_{x,e}(A),\ \text{여기서}\ y \neq x \in S \quad (4)$$

5. Absolute Privacy: 공격자는 통신로상의 어떠한 존재도 알 수 없다. $|S| = \infty$라고 하면, 우리는 $d_{x,e}(A)=1$ 이라고 정의한다.

Ⅲ. Building blocks

3.1. 그누텔라

그누텔라는 분산 검색 환경에서 사용되어지는 peer to peer, decentralized 모델이다. 모든 peer는 클라이언

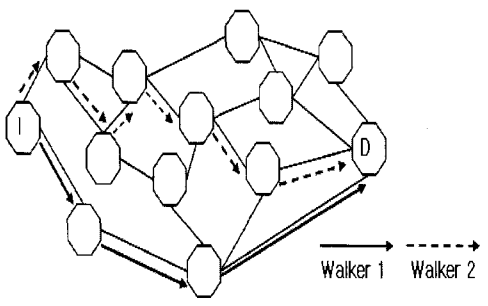
트와 서버 두 가지 역할을 모두 할 수 있다. 그누텔라에서 각각의 peer는 오버레이 네트워크에 가입하게 된다. 오버레이 네트워크는 단지 자신의 neighbor peer에 대해서만 알 고 있다. 그누텔라에서 사용자들이 원하는 객체를 찾고자 하면 질의 플러딩(query flooding)을 하게 되는데 이 때 질의 플러딩의 범위에 제한 설정을 함으로써 네트워크의 오버헤드를 막는다. 객체를 가진 사용자가 파일 전송 시에는 그누텔라 네트워크를 통해 파일을 전송하는 것이 아닌, out-of-network를 통해 직접적으로 파일 요청자에게 파일을 전송하게 된다[20].

3.2. 랜덤워크

시스템을 확장이 쉽게 되도록 하고 또한 파일 요청자의 검색동안 지연을 최대한 줄이기 위한 한 방법으로 개발된 기술이다. 가장 간단한 랜덤워크는 다음의 규칙을 따른다[19].

- 시작점이 있다.
- 경로상에 있는 어느 하나의 점에서 다음 점까지 계속적으로 이어지게 된다.
- 경로상에 있는 어느 하나의 점에서 다음 점은 랜덤하게 선택되어지고 방향은 알 수 없다.

예를 들어 다른 두 방향으로 walker를 보냈을 때 어느 일정한 정도의 홉에서 2개의 walker가 우연히 일치했을 때의 상황이 발생할 수 있게 된다. 우리는 이러한 랜덤 워크가 일치했을 때의 경우를 이용한다. 랜덤 워크의 초기 버전은 [그림 1]과 같이 단지 하나의 walker들을 이용하는 것이다. 플러딩(flooding) 검색과 비교하였을 때, 하나의 walker를 이용한 검색 방법이 효율성 측면에서 우월한 위치에 있다[1].



[그림 1] 기본 랜덤 워크

3.3. IP 멀티캐스트

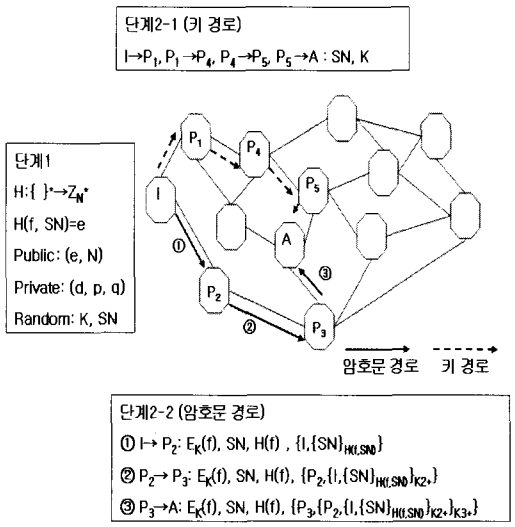
하나 이상의 송신자에서 그룹의 수신자에게 패킷을 전송하는 것이 요구되어 질 때 IP 멀티캐스트를 사용한다. 그렇다면 송신자는 여러 수신자들의 모든 IP 주소를 가지고 있어야 하는가에 대한 문제가 생기는데, 이러한 이유 때문에 멀티캐스트에서는 간접 주소를 사용한다. 즉, 송신자가 동일한 간접 IP 주소를 가진 수신자 그룹에게 메시지를 전송하는 방법이 IP 멀티캐스트이다. 간접 주소는 인터넷 그룹 수신자를 나타내기 위한 단일 인식자 D 클래스를 멀티캐스트 주소로 사용한다. D 클래스와 관련된 수신자 그룹을 멀티캐스트 그룹이라고 하며, 각 호스트는 멀티캐스트 그룹과는 완전히 독립적인 단일한 IP 유니캐스트 주소를 가지고 있다. 멀티캐스트 그룹에 참가하고 탈퇴하는 것은 인터넷 그룹 관리 프로토콜(IGMP)를 이용한다[20].

Ⅳ. Dynamic Onion Pure P2P Protocol

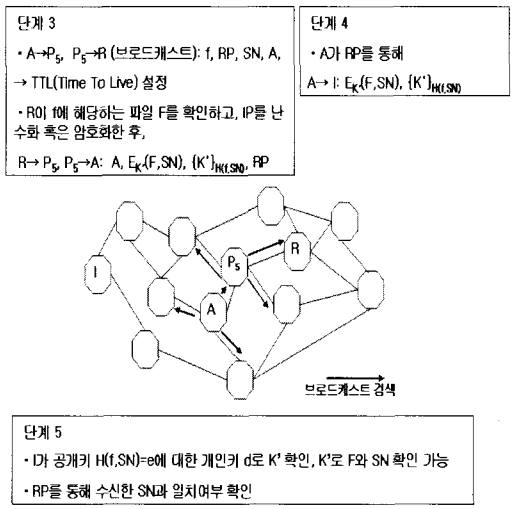
우선 우리의 프로토콜은 현재 P2P 시스템에 접속해 있는 사용자를 알 수 있는 peering node list를 제공하는 server가 존재하지 않는다고 설정한다. 그리고 파일 요청자에 의한 파일 검색부터 파일 확인 과정은 총 5단계로 나누어진다.

단계 1: 환경 및 키 설정

키 생성단계에서 파일 요청자 I는 랜덤한 sequence number SN과 검색을 원하는 쿼리 정보 f 그리고, 공개된 해쉬 함수 $H:\{\}^{*}\rightarrow Z_N^{*}$를 이용해서, $H(f,SN)=e$를 생성한다. 이때의 공개키 값은 파일 송·수신 목적으로 생성한 것이기 때문에 다른 사용자들이 본다고 하여도 그 사람의 ID는 확인할 수 없다. 공개된 값은 $(e,N=p\cdot q)$ 이고 비밀값은 (d,p,q)가 된다. 파일 요청자 I는 비밀키 d를 사용하여 파일 응답자로부터 전송받은 암호화된 파일로부터 원하는 정보를 얻을 수 있다. 이때 사용되어지는 암호 알고리즘은 128bit - AES와 같은 대칭키 암호 알고리즘을 사용한다.



[그림 2] Dynamic Onion Pure P2P Protocol의 익명 쿼리



[그림 3] Dynamic Onion Pure P2P Protocol의 파일검색 및 전송

단계 2: 검색 단계

파일 요청자 I는 익명성을 제공받으면서 원하는 정보를 얻기 위해서 voluntary middle peer A를 설정하여 대신 정보를 검색하도록 하게 되는데, 이 voluntary middle peer의 결정과정은 다음과 같다. 파일 요청자 I는 다음의 두 개의 메시지 패킷을 생성하여 임의로 선택한 서로 다른 2개의 peer들에게 시간차를 두어 각각 전송한다.

① 키경로: $\{SN, K\}$

② 암호문경로와 Onion경로 :

$$\{E_K(f),\ SN,\ H(f)\},\ \{I,\{SN\}_{H(f,SN)}\} \quad (5)$$

이 때 Onion 경로는 자신이 원하는 정보를 획득하기 위해서 요구되는 return path RP를 동적으로 (dynamic) 설정한다. 이러한 두 개의 메시지 패킷들은 여러 peer를 통과하면서 다음과 같이 변형된다.

① 키경로: $\{SN, K\}$

② 암호문 경로와 RP :

$$\{E_K(f),\ SN, H(f)\},\ \left\{P_3,\left\{P_2,\left\{I,\{SN\}_{H(f,SN)}\right\}_{K_{2+}}\right\}_{K_{3+}}\right\} \quad (6)$$

이 두 개의 메시지 패킷 $\{SN,K\}$과 $\{E_K(f),\ SN,\ H(f)\}$, $\left\{P_3,\left\{P_2,\left\{I,\{SN\}_{H(f,SN)}\right\}_{K_{2+}}\right\}_{K_{3+}}\right\}$ 을 함께 전송 받는 네트워크상의 peer는 다음 검증 과정을 수행하여 조건이 만족되면 voluntary middle peer가 된다.

① 키경로와 암호문 경로에서 받은 SN의 일치여부를 확인한다.

② 암호문경로에서 수신한 $E_K(f)$를 키 경로에서 수신한 K로 복호화한 후, f에 대해 해쉬 함수를 적용하여 peer가 계산한 $H(f)$값과 암호문 경로에서 수신한 $H(f)$값이 동일한지 확인한다.

단계 3: 파일 검색

[그림 2]에서와 같이 voluntary middle peer A는 F를 소유한 R을 효율적으로 찾기 위해 $\{f, RP, SN, A\}$를 TTL(Time To Live)만큼 브로드캐스트 한다. $\{f, RP, SN, A\}$를 수신한 R은 query 파일 f에 해당하는 파일 F가 있는 것을 확인하고, 랜덤한 symmetric key K'값을 선정한다. R은 메시지를 받았던 peer P_5에게 (7)과 같은 메시지를 전송하고 P_5는 A에게 같은 메시지를 전송한다. 만약 P_5 이후에 다른 peer가 있는 상황일지라도 그 peer는 A의 ID를 확인할 수 있기 때문에 결국에 A에게 동일한 메시지를 전송한다. 이때,

R은 패킷 전송간 자신의 IP주소가 나타나지 않도록 처리(임의의 난수를 추가 혹은 암호화)하여 파일 응답자의 익명성을 제공한다.

$$R \rightarrow P_5,\ P_5 \rightarrow A\ :\ A, E_{K'}(F,SN), \{K'\}_{H(f,SN)},\ \left\{P_3, \left\{P_2, \left\{I, \{SN\}_{H(f,SN)}\right\}_{K_{2+}}\right\}_{K_{3+}}\right\} \quad (7)$$

단계 4: 파일 F 전송

A는 $A, E_{K'}(F,SN), \{K'\}_{H(f,SN)}, \left\{P_3, \left\{P_2, \left\{I, \{SN\}_{H(f,SN)}\right\}_{K_{2+}}\right\}_{K_{3+}}\right\}$ 메시지를 수신한 후에, I로부터 A까지 설정된 RP $\left\{P_3, \left\{P_2, \left\{I, \{SN\}_{H(f,SN)}\right\}_{K_{2+}}\right\}_{K_{3+}}\right\}$를 통해 I에게 $E_{K'}(F,SN), \{K'\}_{H(f,SN)}$를 전송한다. RP상에서 P_3는 A에게서 $\left\{P_3, \left\{P_2, \left\{I, \{SN\}_{H(f,SN)}\right\}_{K_{2+}}\right\}_{K_{3+}}\right\}$ 메시지를 수신한 후 자신의 ID를 확인하고 개인키로 Onion을 벗겨낸 후, $\left\{P_2, \left\{I, \{SN\}_{H(f,SN)}\right\}_{K_{2+}}\right\}$를 획득한다. 같은 방법으로 P_3는 P_2의 ID를 확인하고, P_2에게 $\left\{P_2, \left\{I, \{SN\}_{H(f,SN)}\right\}_{K_{2+}}\right\}$ 메시지를 전송하면 P_2는 개인키로 Onion을 벗겨내고 $\left\{I, \{SN\}_{H(f,SN)}\right\}$ 메시지를 확인하고 I에게 $\left\{I, \{SN\}_{H(f,SN)}\right\}$를 전송하게 된다.

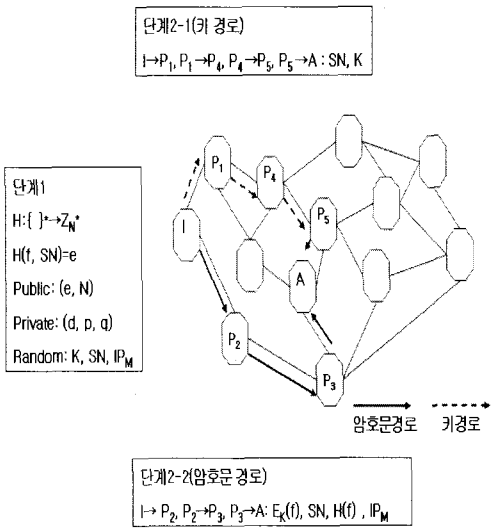
단계 5: I의 파일 획득

I는 공개키 $H(f,SN)=e$에 대한 개인키 d로부터 K'를 알 수 있고, K'를 통해 원하는 파일 F와 SN을 확인할 수 있다. 또한 RP를 통해 수신한 SN과 $E_{K'}(F,SN)$을 통해 확인한 SN이 동일할 때 파일 F의 무결성을 확신한다.

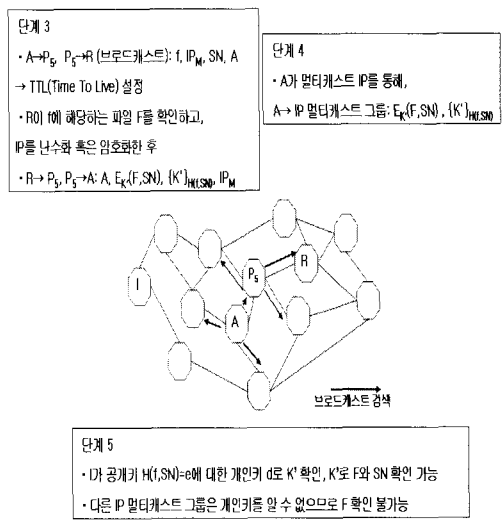
V. IP Multicast Pure P2P Protocol.

IP Multicast Pure P2P Protocol은 Onion 라우팅 기법을 사용하지 않고 IP 멀티캐스트 기법을 사용하며 총 5단계로 나누어진다.

단계 1: 환경 및 키 설정



[그림 4] IP Multicast Pure P2P Protocol의 익명 쿼리



[그림 5] IP Multicast Pure P2P Protocol의 파일검색 및 전송

환경과 키 설정에서 Dynamic Onion Pure P2P Protocol과 다른 점은 랜덤한 멀티캐스트 IP를 추가로 생성한 것이다.

단계 2: 검색 단계

검증과정을 거쳐 voluntary middle peer를 결정하는 것은 Dynamic Onion Pure P2P Protocol 과 동일하다.

하지만 암호문 경로상에서 peer들이 Onion을 형성하는 것이 아니라 [6]에서와 유사하게 walker가 오버레이 될 때 각각의 peer들이 IP 멀티캐스트 그룹에 참가할지를 결정한다.

단계 3: 파일 검색

Dynamic Onion Pure P2P Protocol과 동일하다.

단계 4: 파일 F 전송

[그림 5]와 같이 R은 패킷에서 자신의 IP 주소가 나타나지 않도록 처리(임의의 난수를 추가 혹은 암호화)하여 R을 익명화 시킬 수 있도록 한 후 A를 향해 $A, E_{K'}(F,SN), \{K'\}_{H(f,SN)}, IP_M$ 메시지를 전송한다. A는 IP 멀티캐스트 방법으로 멀티캐스트 그룹에게 $E_{K'}(F,SN), \{K'\}_{H(f,SN)}$ 메시지를 전송한다.

단계 5: I의 파일 획득

I가 파일 F를 얻는 방법은 Dynamic Onion Pure P2P Protocol과 동일하다. 차이점은 I를 포함하여 다른 IP 멀티캐스트 그룹 peer들 역시 A가 전송한 메시지를 수신하지만 다른 peer들은 I의 개인키를 알 수 없으므로 파일 F의 내용을 알 수 없다.

Ⅵ. 논 의

6.1. 익명성 분석

이 장에서는 제안 프로토콜이 P2P 네트워크에서 다양한 참여자들로부터 어떻게 어느 정도의 익명성을 보장할 수 있는지를 분석한다. 익명성 정도는 공격자가 파일 요청자나 응답자의 신원(identity)을 확인할 수 없는 확률로 결정되어 진다. 현재 P2P 시스템에 접속한 전체 peer의 수가 n이며, a는 시스템에 있는 공격자들의 수를 나타낸다. 익명성 정도는 다음의 5종류의 참여자 - 파일 요청자(또는 응답자), 중간 peer, voluntary middle peer, local 도청자, 공모하는 peer들로 나누어서 분석한다. 이 분석은 [5]와 [6]에서의 분석방법에 기반하여 우리의 시스템에 맞게 확장한 것이다.

1. 파일 요청자 (또는 응답자): 세션 네트워크에 있는 모든 peer는 파일 요청자 혹은 응답자로서 활동할 같은 확률을 가지고 있다. 그러므로 파일 요청자와 응답자는 정확히 $1/(n-1)$의 같은 확률로 서로의 신원을 추측할 수 있다. 즉, 파일 요청자가 응답자의 신원을 확인할 수 있는 확률(또는 파일 응답자가 요청자의 신원을 확인할 수 있는 확률)은, $\Pr_I(R) = \frac{1}{n-1}$(혹은 $\Pr_R(I) = \frac{1}{n-1}$)로 나타낼 수 있다.

2. 중간 peer: 우리 시스템에서 각 peer들은 바로 앞의 peer와 다음의 peer 이외의 정보는 알 수 없으며, 바로 앞의 peer가 파일 요청자인지 확신할 수도 없다. 이것은 voluntary middle peer A까지는 랜덤워크 방법으로 메시지가 전송되고, A 이후로는 브로드캐스트에 의해 메시지가 전송되는 상황에서 오버레이 네트워크의 특징에 해당하는 것이다. 따라서 어떤 peer가 파일 요청자 혹은 응답자인지 중간 peer가 랜덤하게 추측하는 확률은 $1/(n-1)$이다. 그러나 중간 peer들의 수가 k개라면 파일 요청자나 응답자의 신원을 정확히 추측하는 확률은 $1/(n-k)$로 변화된다. 중간 peer들을 m이라 한다면, $\Pr_m(I) = \frac{1}{n-k}$

(또는 $\Pr_m(R) = \frac{1}{n-k}$)로 나타낼 수 있다.

3. voluntary middle peer A : A가 파일 요청자나 응답자의 신원을 확인할 수 있는 확률은 $\Pr_A(I) = \frac{1}{n-k}$(혹은 $\Pr_A(R) = \frac{1}{n-k'}$)이다. k는 파일 요청자부터 A까지의 peer들의 수를 말하고 k'는 파일 응답자로부터 A까지의 peer들의 수를 말한다. A 역시 중간 peer들과 마찬가지로 전 · 후의 peer들외의 정보는 알 수 없다. A는 파일 요청을 확인하면, 브로드캐스트를 통해 요청 파일 F를 얻은 후 RP나 IP 멀티캐스트의 방법으로 요청자에게 파일을 전송하기 때문이다.

4. local 도청자: 도청자는 peer와 peer 사이에서 통신되는 모든 메시지를 모니터할 수 있는 공격자이다. 하지만 우리는 파일 요청자의 공개키를 식별할 수 없도록 하였기 때문에 local 도청자는 peer들 사이에서 교환되는 메시지의 내용을 확인할 수는 있지만 파일

요청자에 대한 어떠한 정보도 알 수 없다. 그러므로 local 도청자를 le라 한다면, le가 정확히 파일 요청자나 응답자를 추측할 확률은 $\Pr_{le}(l)=\frac{1}{n}$이다.

5. 공모하는 peer들: 공모하는 peer들에 대한 요청자나 응답자의 익명성은 요청 경로와 반환 경로의 다른 설정으로 인해 세부적인 과정에서 다소 차이가 있으므로 이 두 가지 경우를 나누어 생각하기로 한다.

정리1. 제안 프로토콜은 공모하는 공격자들에 대하여 Beyond Suspicion 정도의 파일 요청자의 익명성을 보장한다.

우리의 시스템은 요청자가 파일을 요청하면, 랜덤워크 방법을 사용하여 두 가지 경로로 나누어 라우팅되어 가다가 해당 조건을 만족하는 voluntary middle peer A를 설정하게 된다. 이것은 파일 요청자(혹은 응답자)의 익명성을 보장하기 위한 방법으로, 이 A는 요청자(혹은 응답자)를 대신하여 역할을 수행하게 된다. 따라서 공모하는 peer들에게 파일 요청자의 신원이 드러날 위험은 voluntary middle peer A를 설정하기 앞 단계까지에 국한된다. A의 설정 이후부터는 A가 요청자를 대신하여 역할을 수행하기 때문에 요청자의 신원이 드러날 경우는 고려하지 않아도 무방하다. 따라서 공모하는 peer들이 파일 요청자를 추측할 수 있는 확률은 voluntary middle peer A에게까지 파일이 도달하는 두 가지 경로로 나누어 생각할 수 있는데, 이 과정은 랜덤워크 방법을 사용하여 진행하는 [1]의 논문과 상황이 일치한다. [1]에서 공모하는 공격자들이 정확히 파일 요청자를 추측할 확률은 $2(a+1)/n$이다. 따라서 우리의 스킴이 [1]과 같은 익명성을 가지는 것은 명백하며, 이 확률 $2(a+1)/n$은 우리가 정의한 Beyond Suspicion에 해당된다. 세부 증명 과정은 [1]과 거의 유사하므로 이 논문에서는 생략한다.

정리2. 제안 프로토콜은 공모하는 공격자들에게 Beyond Suspicion 정도의 파일 응답자의 익명성을 보장한다.

$z\in Z$를 파일 F 전송 간에 있는 공모하는 공격자들이라고 가정한다. 응답자의 익명성 역시 voluntary middle peer A가 응답자의 역할을 대신해주기 때문에 파일 F를 전달하는 데 있어서 응답자로부터 A까지 도달하는 경로 까지만 고려하고 A부터 파일 요청자까지의 경로(RP와 IP 멀티캐스트)는 고려하지 않아도 된다. 이 경우, 파일 요청자의 익명성처럼 두 가지 경로로 나누어 랜덤워크 방법으로 진행하지 않고 파일 응답자가 voluntary middle peer A에게 일반적인 라우팅 방법으로 파일 F를 전송하며 이 때 송신자(즉, 응답자)의 주소지는 암호화 또는 다른 블라인딩 기법을 이용하여 처리하는 점이 요청자의 익명성과는 다른 점이다. 자세한 과정은 다음과 같다.

$H_k(k\geq 1)$는 경로상에 있는 첫 번째 공격자가 k번째에 위치하고 있는 확률이다. 여기서 파일 응답자는 0번째에 위치한다. $\Pr(R)$을 응답자가 공격자(공모하는 peer)의 바로 전의 peer일 확률로 정의한다. $H_{m+}=H_m\vee H_{m+1}\vee H_{m+2}\vee\ldots$이라 하면 $H_1=R$이다. 여기서 우리의 목적은 공모하는 공격자들이 파일 응답자의 신원을 추측할 확률 $\Pr[R|H_{1+}]$을 계산하는 것이다.

$$\Pr[H_i]=\left(\frac{n-a}{n}\right)^{i-1}\frac{a}{n} \tag{8}$$

$$\Pr[H_{2+}]=\frac{a}{n}\sum_{k=1}^{\infty}\left(\frac{n-a}{n}\right)^k=\frac{a}{n}\left(\frac{\frac{n-a}{n}}{1-\frac{n-a}{n}}\right)=\frac{n-a}{n} \tag{9}$$

$$\Pr[H_{1+}]=\frac{a}{n}\sum_{k=0}^{\infty}\left(\frac{n-a}{n}\right)^k=\frac{a}{n}\left(\frac{1}{1-\frac{n-a}{n}}\right)=1 \tag{10}$$

우리는 위의 식으로부터

$$\Pr[H_1]=\frac{a}{n},\ \Pr[R|H_1]=1,\ \Pr[R|H_{2+}]=\frac{1}{n-a}$$

을 구할 수 있으며, 다음을 얻어낼 수 있다.

$$\Pr[R]=\Pr[H_1]\cdot\Pr[R|H_1]+\Pr[H_{2+}]\cdot\Pr[R|H_{2+}]$$

$$=\frac{a+1}{n} \tag{11}$$

$$\Pr[R|H_{1+}]=\frac{\Pr[R\wedge H_{1+}]}{\Pr[H_{1+}]}\leq\frac{\Pr[R]}{\Pr[H_{1+}]}=\frac{a+1}{n} \tag{12}$$

그리고 (10)의 식은 2장에서의 정의에 의해 $\sum_{z\in Z}\Pr_z(x)$와 같은 의미이다. 즉,

$$\Pr[R|H_{1+}]=\Pr[R|H_1\vee H_2\vee H_3\vee\ldots]=\sum_{z\in Z}\Pr_z(x)$$

이므로, 다음과 같은 결과를 얻을 수 있다.

$d(A) = \sum_{y \in S \neq x} \Pr_e(y) = 1 - \sum_{z \in Z} \Pr_z(x) = 1 - \frac{a+1}{n}$

$= \frac{n-a-1}{n}$. 이것은 정의 3에 의해 Beyond Suspicion에 해당한다.

$\{|S| > 1, 1/|S| \leq d_{x,e}(A), d_{y,e}(A) \leq d_{x,e}(A),$ 여기서 $y \neq x \in S\}$

결과적으로 우리가 제안한 Dynamic Onion Pure P2P Protocol과 IP Multicast Pure P2P Protocol에서 파일 요청자나 응답자의 익명성 정도는 n의 수가 증가할수록 커진다고 말할 수 있다.

6.2. 효율성 분석

Dynamic Onion Pure P2P Protocol은 [1], [2]에서 peering node list를 알 수 있는 서버를 통해 peer들의 신원이 노출되는 문제를 해결하기 위해 우리는 다이나믹한 RP를 구성하였으며, 그 결과로 peer들에게 공개키 계산을 추가적으로 요구하게 된다. 이것은 trade-off의 문제로, 개개인의 잠재적인 프라이버시를 문제를 해결하고, [1], [2]에서와 같이 RP가 미리 정해져 있을 때보다는 파일 요청자가 원하는 파일을 수신할 수 있는 확률을 높이고, RP상의 첫 번째 peer가 파일 응답자와 멀리 떨어져 있을 때의 비효율성을 해결한 반면 voluntary middle peer A가 정해이기 이전에 peer들의 공개키 계산이 요구되는 약점을 가진다.

IP Multicast Pure P2P Protocol에서는 peer들이 Dynamic Onion Pure P2P Protocol에서와 같이 Onion을 구성해 가는 것이 아닌, IP 멀티캐스트 그룹을 이용한 파일 전송이므로 peer들의 공개키 계산이 요구되지 않아서 계산적인 면에서 상당히 효율적이다. 그러나 이 또한 trade-off의 문제로 IP 멀티캐스트 그룹에 속한 peer들은 자신이 요구하지 않는 파일에 대해 가용성 여부를 검증하는 절차를 수행하게 된다.

한편 우리가 제안한 두 개의 프로토콜들은 공통적으로 voluntary middle peer가 브로드캐스트(TTL설정)를 한다. 이러한 방법은 [1], [2]에서의 임의의 peer에서 실행하는 브로드캐스트 검색 및 플러딩 검색을 할 때 보다 원하는 파일을 검색할 수 있는 확률을 증가시킨다.

[표 1]은 앞의 내용을 기초로 관련 논문들의 프로토콜과 우리의 프로토콜을 비교·분석한 것이다.

[1],[2]와 같이 파일 요청자와 파일 응답자의 익명성은 동일하게 보호되지만, 우리가 제안한 프로토콜들은 peering node list를 제공하는 서버가 없는 상태에서 파일 응답자가 다이나믹한 RP를 통해서 그리고 멀티캐스트 IP를 통해서 파일 요청자에게 파일에 관련된 내용을 전송함으로써 시스템에 가입한 모든 peer들의

[표 1] 기존 논문 프로토콜과 우리의 프로토콜 비교 · 분석

	Xiao's 프로토콜[2]	J. Han's 프로토콜[1]	Dynamic Onion Pure P2P Protocol	IP Multicast Pure P2P Protocol
PKI 적용	○	○	×	×
P2P의 다이나믹한 시스템의 적합성	×	×	○ (Dynamic RP)	○ (RP대신 Multi 이용)
peering node list를 제공하는 server의 존재 유무	○	○	×	×
I의 익명성	○	○	○	○
R의 익명성	○	○	○	○
I와 R의 상호 익명성	○	○	○	○
전송간 익명성 향상	×	×	파일 송·수신을 위한 공개키 설정	파일 송·수신을 위한 공개키 설정
검색 신속성 및 정확성을 위한 메소드	Agent가 플러딩 검색 후, 파일을 찾지 못하면 적당한 하나의 peer에서 브로드캐스트	VMP에서 플러딩	VMP에서 브로드캐스트	VMP에서 브로드캐스트

* Dynamic RP : 다이나믹한 Return Path 구성
* Multi : Multicast Mothed
* Agent : 쿼리 에이전트 peer
* I : 파일 요청자
* R : 파일응답자
* VMP : voluntary middle peer

신원이 노출될 수 있는 프라이버시 위협으로부터 보호하였다. 또한 전송간의 안전성을 고려하여 키를 설정하였으며, TTL을 설정한 브로드캐스트 검색을 통하여 원하는 파일을 검색할 확률을 높였다.

Ⅶ. 결 론

지금까지 우리는 다이나믹한 P2P 파일 공유 시스템에 적합하면서 익명성을 제공하는 두 가지 스킴을 제안하고 그것의 안전성과 효율성을 분석하였다. 이 스킴들은 기존 논문들이 필요로 했던 peering node list를 제공하는 서버를 사용하지 않고 사용자들의 접속 상태를 모르고도 파일 요청자의 익명 파일 검색·수신이 가능하게 함으로써 사용자들의 프라이버시 보호를 한 층 더 강화시켰다. 서론의 보도 자료를 보아도 알 수 있듯이 정보화 사회에서 P2P의 사용은 날로 증가할 것이며 따라서 그 편이성의 이면에 잠재해 있는 사용자들의 프라이버시 문제 역시 간과할 수 없는 것이 사실이다. 이러한 맥락에서 익명성을 제공하는 P2P 시스템에 대한 연구는 앞으로도 계속 지속되어져야 할 것이며, 뿐만 아니라 이런 익명성을 악용하여 발생할 수 있는 문제점(예; 허가받지 않은 파일의 불법 소유)들 또한 방어할 수 있는 스킴의 개발 역시 빼 놓을 수 없는 과제이다.

참고문헌

[1] Jingsong Han, Yunhao Liu, "A Random Walk Based Anonymous Peer-to-Peer Protocol Design", *ICCNMC 2005*, LNCS 3619, pp. 143-152, 2005.

[2] L. Xiao, Z. Xu, and X. Zhang. "Low-Cost and Reliable Mutual Anonymity Protocols in Peer-to Peer Networks", *IEEE Transactions on Parallel and Distributed Systems*, 2003.

[3] Jingsong Han, Yunhao Liu, "A Mutual Anonymous Peer-to-peer Protocol Design", *IPDPS'05*, 2005.

[4] Byungryong Kim, "Client and Server Anonymity Preserving in P2P Networks", *ASWC 2006*, LNCS 4185, pp. 689-695, 2006.

[5] Chin-Chen Chang, Chin-Yang Lin, "Simple efficient mutual anonymity protocols for peer-to-peer network based on primitive roots", *Elsevier Journal of Network and Computer Applications 30*, pp. 662-676, 2007.

[6] Baoliu YE, Minyi, "A Multicast Based Anonymous Information Sharing Protocol for Peer-to-Peer Systems", *IEICE*, 2006.

[7] Li Xiao, Yunhao Liu, "Mutual anonymous overlay multicast", *Elsevier J. Parallel Distrib.Comput. 66*, pp. 1205-1216, 2006.

[8] Ion Stoica, Robert Morris, "Chord: A Scalable Peer-to-peer Lookup Service for internet Applications", *ACM*, SIGCOMM'01, 2001.

[9] Brian Neil Levine, Clay Shields, "Hordes: a multicast based protocol for anonymity", *IOS*, Journal of Computer Security 10, pp. 213-240, 2002.

[10] V. Scarlata, B. Levine, and C. Shields, "Responder Anonymity and anonymous peer-to-peer File Sharing", in Proc. of *IEEE International Conference on Network Protocols(ICNP)*, Riverside, CA, 2001.

[11] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, "Freenet: A distributed anonymous information storage and retrieval system", In *Workshop on Design Issues in Anonymity and Unobservability*, pp. 46-66, 2000., http://citeseer.nj.nec.com/clarkeoofreenet.html.

[12] The Gnutella Protocol Specification v0.41 Document Revision 1.2., http://rfc-gnutella.sourceforge.net/developer/stable/index.html/

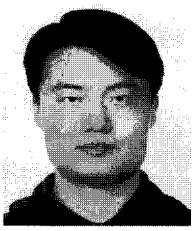
[13] Ben Y.Zhao, Ling Huang, Jeremy Stribling, Sean C.Rhea, Anthony D.Joseph, and John Kubiatowicz, "Tapestry: A Resilient Global-scale Overlay for Service Deployment", *IEEE Journal on Selected Areas in Communications*, 2004.

[14] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, Scott Schenker, "A scalable content-addressable network", *Proceedings of the 2001 conference on Applications*, technologies, architectures, and protocols for computer communications table of contents.

[15] C. Shields and B.N. Levine, "A protocol for anonymous communication over the Internet," *Proc. 7th ACM Conference on Computer and Communication Security*, ACM CCS 2000, Nov. 2000.

[16] M.K. Reiter and A.D. Rubin, "Crowds: Anonymity for Web transactions," *ACM Trans. Information and System Security*, vol.1, no.1, pp. 66-92, Nov. 1988.
[17] 권혁찬, 문용혁, "P2P 표준화 및 기술 동향", *ETRI*, 전자통신동향분석, 제 22권, 제 1호, pp. 11-23, 2월, 2007.
[18] 김병오, 김일우, "분산 해시 테이블 기반 P2P 기술 동향", *ETRI*, 전자통신동향분석, 제 21권, 제 6호, pp. 179-189, 12월, 2006.
[19] http://en.wikipedia.org/wiki/Random_walk
[20] 강현국, 신용태, "컴퓨터 네트워킹", *PEARSON Addison Wesley Korea*, 제 3판, pp. 394-403, 11월, 2005.

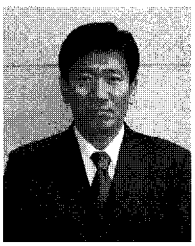
<著者紹介>



조 준 하 (Jun-ha Cho) 정회원
2000년 3월 : 육군사관학교 무기공학과 졸업
2006년 3월~현재 : 고려대학교 정보경영공학전문대학원 정보경영학과 석사과정
<관심분야> 암호프로토콜, 암호이론, PET기술, 익명성 연구

이 현 숙 (Hyun-Sook Rhee) 정회원
1998년 2월 : 단국대학교 수학과 졸업
2000년 2월 : 단국대학교 수학과 석사 졸업
2001년 3월~현재 : 고려대학교 정보보호대학원 박사수료
<관심분야> 암호프로토콜, 암호이론, 익명성연구, PET기술

박 현 아(Hyun A Park) 정회원
2003년 2월 : 고려대학교 수학과 졸업
2005년 2월 : 고려대학교 정보보호대학원 석사 졸업
2005년 3월~현재 : 고려대학교 정보보호대학원 박사수료
<관심분야> 암호프로토콜, 익명성 연구, DB 프라이버시



이 동 훈(Dong Hoon Lee) 정회원
1983년 8월 : 고려대학교 경제학사
1987년 12월 : Oklahoma University 전산학 석사
1992년 5월 : Oklahoma University 전산학 박사
1993년 3월 ~ 1997년 2월 : 고려대학교 전산학과 조교수
1997년 3월 ~ 2001년 2월 : 고려대학교 전산학과 부교수
2001년 2월~현재 : 고려대학교 정보경영공학전문대학원 교수
<관심분야> 암호프로토콜, 암호이론, USN 이론, 키 교환, 익명성 연구, PET 기술.