피어 투 피어 네트워크에서 스워밍 기법을 위한 보안 프로토콜

이관섭* · 이광식** · 이장호*** · 한승철**

A Security Protocol for Swarming Technique in Peer-to-Peer Networks

Kwan-seob Lee* · Kwan-sik Lee** · Jang Ho Lee*** · Seung-chul Han**

이 연구는 한국연구재단 중견연구자 핵심연구지원사업(2011-0015187),
일반연구자 지원사업(2011-0003930), 2010년도 명지대학교 교책중점연구소 지원으로 연구되었음

요 약

초고속 통신망의 일반화와 다양한 온라인 서비스의 출현으로 인터넷을 통한 대용량 콘텐츠 배포에 대한 수요가 증가하고 있다. 이에 따라 서버와 네트워크의 부담이 커지자 P2P 네트워크 기반으로의 전환 움직임이 활발해지고 있다. P2P 스워밍 기법은 서버의 역할을 최소화하고, 트래픽을 고르게 분산시켜 네트워크 과부하로 인한 각종 문제들을 해결할 수 있으며, 자원의 유지보수 비용 또한 절감시킬 수 있다. 하지만 P2P 스워밍 기법의 특성상 보안 서비스제공을 위해선 많은 메시지 교환이 필요하다. 본 논문에서는 P2P 스워밍기법에서 기밀성, 인증, 무결성, 접근 제어 등의 보안 서비스를 제공하는 효율적인 보안 프로토콜을 제안한다. 제안된 프로토콜은 안드로이드 스마트폰 플랫폼에서 구현하여 실험을 하였다. 제안된 프로토콜은 스워밍기법을 이용하는 상용시스템에 사용될 수 있을 것으로 기대된다.

ABSTRACT

With fast deployment of high-speed networks and various online services, the demand for massive content distribution is also growing fast. An approach that is increasingly visible in communication research community and in industry domain is peer-to-peer (P2P) networks. The P2P swarming technique enables a content distribution system to achieve higher throughput, avoid server or network overload, and be more resilient to failure and traffic fluctuation. Moreover, as a P2P-based architecture pushed the computing and bandwidth cost toward the network edge, it allows scalability to support a large number of subscribers on a global scale, while imposing little demand for equipment on the content providers. However, the P2P swarming burdens message exchange overheads on the system. In this paper, we propose a new protocol which provides confidentiality, authentication, integrity, and access control to P2P swarming. We implemented a prototype of our protocol on Android smart phone platform. We believe our approach can be straightforwardly adapted to existing commercial P2P content distribution systems with modest modifications to current implementations.

키워드

피어 투 피어 네트워크, 스워밍, 보안 프로토콜, 안드로이드, 인터넷

Key word

Peer-to-peer Network, Swarming, Security Protocol, Android, Internet

* 준회원 : 명지대학교 컴퓨터공학과
** 정회원 : 명지대학교 컴퓨터공학과
*** 정회원 : 홍익대학교 컴퓨터공학과(교신저자, janghol@cs.hongik.ac.kr)

접수일자 : 2011. 07. 28
심사완료일자 : 2011. 08. 09

I. 서 론

네트워크 기술의 발전으로 초고속 통신망이 일반화되고 다양한 온라인 서비스 및 비즈니스가 출현함에 따라 인터넷을 통한 대용량 콘텐츠 배포(content distribution)에 대한 수요가 폭발적으로 증가하고 있다[1]. 현재 인터넷을 이용한 배포방식은 크게 중앙집중(centralized) 방식과 분산시스템(distributed) 방식으로 분류할 수 있다.

중앙집중 방식은 한 곳에 집단으로 수용되어 동작되는 일련의 서버들(혹은 서버 클러스터)에 모든 콘텐츠들을 저장하고 중앙 서버에서 콘텐츠 전송, 관리, 사용자 접근 제어 등을 수행하는 방식이다. 이 방식은 서버, 사용자, 콘텐츠 등을 일관성 있게 관리하기가 용이하고, 시스템 업데이트나 유지보수가 간편하다는 장점이 있다. 하지만 중앙 집중 방식의 경우는 네트워크 병목현상(bottleneck), 단일 오류지점(single point of failure), 서버 부하의 급증 등의 단점 때문에 대규모 사용자가 동시에 접속하는 경우에 서버와 네트워크에 심각한 과부하 문제를 야기할 수 있다. 현재 Flash Media Server, Firefly Media Server, VideoLAN 등 웹 중심의 서비스에 주로 사용되고 있다.

분산시스템 방식은 서버와 동일한 내용을 가진 미러 서버(mirror server)들을 네트워크의 여러 곳에 전략적으로 분산 배치해서 네트워크의 병목현상을 줄이고 서버 부하를 분산시키는 방식이다. 현재 최초로 상용 서비스를 시작한 Akamai를 필두로 Amazon, EdgeCast, CDNetworks 등 많은 업체들이 서비스를 제공하고 있으며 대규모의 서비스에 적합하고 한 곳의 서버가 다운되더라도 피해를 최소화할 수 있다는 장점이 있어 중앙집중 방식에 비해 상대적으로 우위에 있는 방식으로 평가 받고 있다. 하지만, 여러 곳에 흩어져 있는 미러 서버들의 동기화와 인프라 유지관리에 많은 비용이 드는 단점으로 인해 사업성적인 측면에서는 한계를 보이고 있다.

이러한 기존 방식들의 문제점들 때문에 차세대 대용량 콘텐츠 배포 방식으로 피어-투-피어(Peer-to-Peer, 이하 'P2P'라 지칭함) 방식이 주목 받고 있다. 이 방식은 네트워크에 연결된 컴퓨터들이 자원(CPU, 메모리, 디스크, 네트워크 대역폭 등)을 공유해서 콘텐츠 배포 시스템을 구성하는 방식으로, 고정된 클라이언트와 서버 대신 동등한 계층의 피어 노드(peer node)들이 클라이언트와 서버 역할을 동시에 수행하여 데이터를 서로 주고받는다. P2P 네트워크는 병목현상, 서버 부하 등의 중앙집중 방식의 문제를 해결할 수 있고, CDN(Content Delivery Network)등과 달리 전용 인프라가 필요하지 않기 때문에 경제적 측면에서 충분히 매력적이다[2].

P2P 네트워크에서의 콘텐츠 배포기술 중 가장 주목받고 있는 것은 스워밍(Swarming) 기법이다(그림 1). 최초 콘텐츠 공급자(seeder)는 배포하고자 하는 콘텐츠를 작은 조각들로 쪼개어 네트워크상의 피어들에게 분산 배포하고, 피어들은 각 조각을 서로 다른 피어들로부터 전송 받아 원래의 파일로 복구한다. 따라서 참여하는 피어 수가 증가할수록 확장성(scalability)이 증가하며, 규모가 커지더라도 추가적인 서버 및 네트워크 인프라 비용이 거의 들지 않는 장점이 있다. 또한 빠른 성능(throughput), 오류 저항(fault tolerant), 부하 분산 (load balance) 등의 특성들로 인해 차세대 기술로 큰 관심을 받고 있다[3].

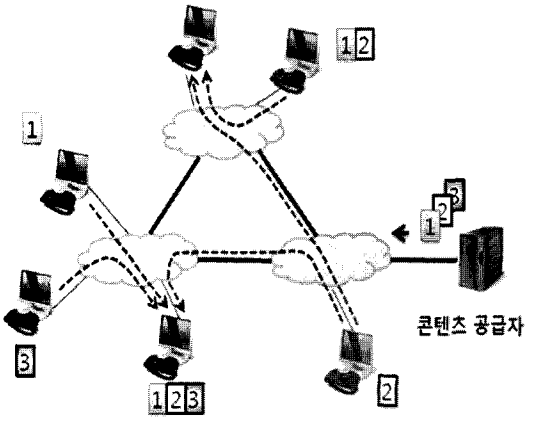

그림 1. P2P 스워밍
Fig. 1 P2P swarming

하지만, 콘텐츠를 작은 조각들로 나누어 분산시키고, 다수의 피어들로부터 전송 받는 스워밍 기법은 인프라 비용적 측면에서는 장점이지만, 중앙집중 방식과 분산 시스템 방식에 비해 콘텐츠 및 사용자 인증 및 관리가 어렵고, 보안서비스 제공을 위한 암/복호화, 메시지 오버헤드가 크다.

P2P 스워밍 기법에서 사용자가 하나의 파일 조각을 다운로드 받을 때마다 기밀성(confidentiality), 인증(authentication), 무결성 (integrity), 접근 제어(access control) 등의 보안 서비스를 제공하기 위해서는 사용자-서버, 사용자-피어, 피어-서버 간의 여러 단계의 메시지 교환을 거쳐야 한다. 더구나 P2P 네트워크와 같이 피어들이 수시로 참여/탈퇴하는 동적인 네트워크 환경(high-churn)에서는 메시지 교환 횟수가 크게 증가하게 된다. 따라서 P2P 네트워크의 성공을 위해선 스워밍 기법에서 사용될 수 있는 효율적인 보안 프로토콜이 반드시 요구된다.

본 논문에서는 P2P 네트워크에서 스워밍 기법을 사용할 때 효율적으로 보안 서비스를 제공하는 프로토콜을 제안한다. 제안하는 프로토콜은 커버로스(Kerberos)에서와 유사한 티켓(ticket)을 사용한다.

제안하는 프로토콜의 실용성을 보이기 위해 모바일 플랫폼으로 주목 받는 안드로이드(Android)에서 구현하고, 실제 스마트폰(삼성 갤럭시S)에 탑재하여 성능을 평가하였다.

이후 본 논문의 구성은 다음과 같다. 2장에서는 관련된 연구들을 소개하고, 3장에 본 연구자가 제안하는 P2P 스워밍 기법을 위한 보안 프로토콜을 기술하고, 4장에서는 제안된 프로토콜을 안드로이드 플랫폼을 기반으로 구현하고 실제 스마트폰 상에서 성능을 평가한 결과를 보인다. 마지막으로, 5장에서 결론을 도출한다.

Ⅱ. 관련 연구

2.1 P2P 콘텐츠 배포

P2P 네트워크는 저렴한 비용, 확장성과 빠른 전송 속도로 인해 전 세계적으로 인기를 얻고 있다. 이와 관련해 MINTS(Minnesota Internet Traffic Studies)[4]에서 조사한 자료에 따르면 일본에서는 전체 인터넷 트래픽의 약 62%, 유럽에서는 65.5%가 P2P 어플리케이션에 의한 것이라 파악되고 있다. CacheLogic[5]이 인터넷 백본과 ISP 데이터 스트림에 대해 수행한 패킷 모니터링의 결과에 의하면 P2P 어플리케이션에 의한 트래픽의 61.4%가 비디오에 의한 트래픽이며, 11.4%는 오디오, 27.2%는 기타 트래픽인 것으로 나타났다.

따라서 현재 대용량 콘텐츠 전송을 필요로 하는 멀티미디어는 주로 P2P 네트워킹과 밀접한 연관성이 있음을 유추할 수 있다.

현재 가장 대중화된 P2P 콘텐츠 배포 어플리케이션은 BitTorrent이다[6]. 사용자는 트래킹 서버(tracker)에 접속하여 파일과 피어들의 정보를 가져온 후, 파일의 각 부분을 여러 피어들로부터 전송 받는다. PPLive[7]는 상용화된 P2P 네트워크 기반의 멀티미디어 콘텐츠 배포시스템 가운데 최대 규모로, 스워밍기반의 비디오 전송과 가십 (gossip) 프로토콜을 이용한 사용자와 콘텐츠 관리를 수행한다. TVAnts[8]와 SopCast[9]는 메쉬(mesh) 기반의 구조를 가지고 있으며, BitTorrent와 유사하게 동작한다. 사용자는 슈퍼 노드에 접속하여 원하는 콘텐츠를 소유한 피어들을 검색한다. 검색된 피어들로부터 버퍼맵(buffer map)을 전송 받은 후, 이에 따라 콘텐츠를 여러 피어들로부터 다운로드 한다.

2.2 P2P 보안 프로토콜

BitTorrent는 배포하려는 파일을 작은 조각들로 쪼개고 각 조각들을 SHA-1 해쉬 함수로 처리함으로 데이터 무결성을 제공한다. Floodgate[10]는 BitTorrent에 기밀성과 인증의 보안서비스를 추가로 제공하는 프로토콜을 제안하였다. 하지만 다수의 피어들로부터 전송이 이루어지는 경우를 고려하지 않으므로 스워밍기법에서 사용하기에 부적합하다. Dandelion[11]은 BitTorrent에서의 인센티브 프로토콜을 제안하였다. 여기서는 각 피어에 접속할 때마다 기밀성과 상호인증을 위해서 트래킹 서버로부터 매번 세션키를 생성하고 인증과정을 거쳐야 하므로 비효율적이다. PPay[12]는 P2P 콘텐츠 배포를 위한 보안 프로토콜을 제안하였다. 그러나 짧은 시간 동안 접속해서 파일 조각들을 무료로 다운로드 받을 수 있는 프로토콜상의 치명적인 약점이 있어 상용시스템에서 사용하기에 적합하지 않다. Scrivener[13]는 P2P 네트워크 피어들간의 신용(credit)기반의 콘텐츠 전송시스템을 제안하였다. 하지만 이는 폐쇄적인 형태의 시스템으로 사용자들이 동적으로 활동하는 경우에는 비효율적이다.

따라서 다수의 동적인 피어들에 연결하여 동시에 다운로드 받는 스워밍 기법에서 효율적으로 동작하는 새로운 보안 프로토콜이 필요하다.

Ⅲ. 스워밍 기법을 위한 보안 프로토콜

3.1 컴포넌트

본 논문에서 제안하는 프로토콜은 트래킹 서버(tracking server), 콘텐츠 수신자 피어(이하 '수신자'로 지칭함), 콘텐츠 송신자 피어(이하 '송신자'로 지칭함), 티켓(ticket)의 네 가지 컴포넌트로 구성된다.

○ 수신자

수신자는 콘텐츠를 수신하기 원하는 피어다. 수신자는 트래킹 서버에 원하는 콘텐츠를 요청하고 콘텐츠 제공이 가능한 피어들의 목록과 티켓을 수신하고, 송신자에게 콘텐츠 전송을 요청하여 수신한다.

○ 송신자

송신자는 수신자가 요청한 콘텐츠를 전송하는 피어다. 송신자는 티켓을 검증 후, 콘텐츠를 수신자에게 전송한다.

○ 트래킹 서버

트래킹 서버는 피어 관리, 콘텐츠 검색, 티켓 생성, 공개키 관리 등의 역할을 수행한다. 공개키 관리를 위해 인증서버를 논리적으로 분리할 수 있으나, 여기서는 간편성을 위해 트래킹 서버로 통합하였다.

○ 티켓

티켓은 수신자, 송신자, 콘텐츠, 송신자공개키, 타임스탬프 정보가 트래킹 서버의 개인키로 암호화된 하나의 묶음으로, 수신자와 송신자 사이에서 콘텐츠의 요청과 전송 등을 위한 검증수단으로 사용된다.

3.2 프로토콜

본 보안 프로토콜의 콘텐츠 송수신은 일반적인 P2P 어플리케이션에서 발생하는 것과는 차이가 있다. 일반적인 P2P 네트워크에서 콘텐츠 송수신은 보안성을 고려하지 않으며 콘텐츠 전송만을 보장한다. 하지만 비즈니스에서 콘텐츠란 서비스 공급자의 자산으로 이윤을 발생시키기 위한 수단으로 이용되기 때문에 P2P 네트워크에 기반한 콘텐츠 배포 시 콘텐츠의 기밀성과 사용자의 인증은 매우 중요한 요소가 된다. 이를 위해 본 보안 프로토콜은 티켓과 공개키 기반 암호화 알고리즘을 이용하여, 콘텐츠의 기밀성을 보장함과 동시에 사용자에 대한 인증까지 효율적으로 제공하도록 설계되었다.

본 논문에서 제안하는 보안 프로토콜은 크게 두 과정으로 구분된다. 프로토콜 기술에 쓰이는 기호들은 표 1과 같다.

표 1. 기호 및 설명
Table 1. Notation

기호	설명
Kas	서버와 피어A의 대칭키
Ks-	서버의 개인키
Kb+	피어B의 공개키
Kab	피어A와 피어B의 세션키
FileName	콘텐츠 이름
A	수신자 피어A
B, C, D	송신자 피어 및 송신자 피어 후보
S	트래킹 서버
List	송신자 및 송신자 후보 목록
TS	타임 스탬프

3.2.1 송신자 선택 (과정 1)

다음의 그림 2와 같이 수신자가 트래킹 서버에 콘텐츠 전송을 요청하면, 콘텐츠 서버는 가능한 피어들의 목록을 수신자에게 전송한다. 수신자는 콘텐츠를 송신할 피어들을 선택하여 트래킹 서버에게 알려준다. 트래킹 서버는 콘텐츠 수신을 위한 티켓을 생성하여 수신자에게 전송한다. 그림2의 모든 과정은 수신자와 송신자가 공유하고 있는 대칭키로 암,복호화 된다.

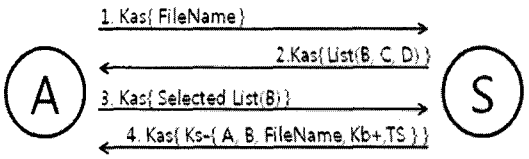

그림 2. 송신자 선택
Fig. 2 Sender selection

Step 1. 수신자는 원하는 콘텐츠의 정보를 자신과 트래킹 서버가 공유하고 있는 대칭키를 이용하여 암호화하여 트래킹 서버에 전송한다.

Step 2. 트래킹 서버는 해당 콘텐츠 제공이 가능한 피어들을 검색하여 피어 목록을 작성하고 대칭키로 암호화하여 수신자에게 전송한다. 콘텐츠 및 피어 검색은 분산 해시 테이블 (distributed hash tables), 가십 (gossip) 프로토콜 등을 이용할 수 있다.

Step 3. 수신자는 피어 목록에서 콘텐츠를 전송받을 송신자들을 선택하고 대칭키로 암호화하여 트래킹 서버에 전송한다. P2P 스워밍 피어 선택 알고리즘은 기존 연구에서 제안되었다 [2,3].

Step 4. 트래킹 서버는 수신자가 송신자에게 콘텐츠 전송 요청 시, 수신자의 인증과 콘텐츠의 암호화에 이용될 티켓(그림 3)을 생성하여 전송한다. 티켓은 수신자와 송신자의 정보, 콘텐츠 정보, 송신자의 공개키, 타임스탬프의 정보를 담고 있다. 티켓은 트래킹 서버의 개인키로 암호화 되어 있으므로 수신자는 서버의 공개키로 복호화하며 이 과정을 통해 트래킹 서버를 인증하게 된다.

그림 3. 티켓
Fig. 3 Ticket

3.2.2 콘텐츠 전송 (과정 2)

두 번째 과정은 그림 4와 같이 수신자가 송신자에게 콘텐츠의 전송을 요청하고 수신하는 과정이다.

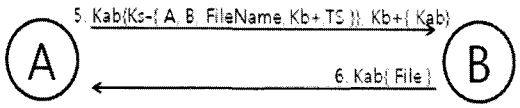

그림 4. 콘텐츠 전송
Fig. 4 Content transmission

Step 5. 수신자는 송신자와 콘텐츠 송수신에 쓰일 세션키를 생성하고, 트래킹 서버가 보내준 티켓을 암호화하여 송신자에게 전송한다. 또한, 생성된 세션키는 티켓 내의 송신자의 공개키로 암호화하여 송신자에게 전송한다.

Step 6. 송신자는 자신의 비밀키로 세션키를 구한 후, 티켓을 복호화한다. 티켓은 트래킹 서버의 공개키로만 복호화 되므로 유효한 티켓발행처의 유효성을 검증할 수 있다. 티켓 속에 있는 콘텐츠정보, 수신자정보, 타임스탬프가 정상적으로 기록되어있을 경우 송신자는 수신자를 정상적인 사용자로 판단하고, 세션키로 콘텐츠를 암호화하여 전송한다.

다음의 그림 5는 지금까지 설명한 Step 1부터 Step 6까지의 보안 프로토콜의 제어 흐름을 보여준다.

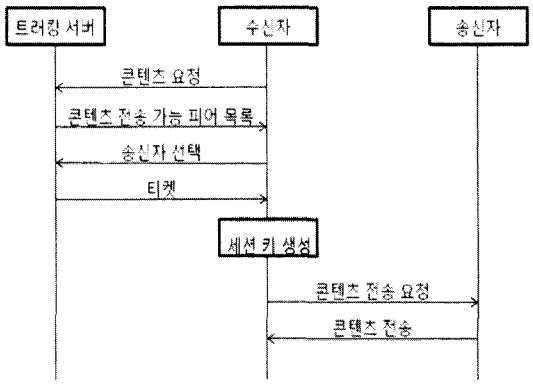

그림 5. 보안 프로토콜 제어 흐름
Fig. 5 Security protocol flow control

3.3 보안성 분석

본 논문의 보안 프로토콜에서 제공하는 보안적 요소는 크게 기밀성(confidentiality), 사용자 인증(user authentication), 접근 제어(access control)로 구분할 수 있다.

3.3.1 기밀성

기밀성은 정보를 인가된 대상에게만 공개하는 것으로 네트워크를 통해 주고 받는 데이터의 내용을 비인가자에게 공개하지 않는 것을 말한다.

P2P 네트워크에서 기밀성을 고려해야 할 주요 요소로는 사용자의 정보와 콘텐츠가 있다. 사용자 정보는 네트워크에 기반한 모든 서비스에서 최선으로 고려해야 할 사항으로, 만일 사용자 정보가 노출될 경우 2차, 3차 피해로 확산될 우려가 있다. 본 보안 프로토콜에서 피어와 트래킹 서버 사이에서 주고 받는 모든 메시지

는 대칭키로 암호화와 복호화 과정을 거치게 된다. 피어와 피어 사이에서 주고 받는 콘텐츠 또한 효율성과 기밀성을 보장하기 위한 수단으로 대칭키를 이용해야 하지만 피어와 피어는 일회성 연결을 생성하여 통신하므로 키의 분배에 어려움이 따른다. 이에 본 보안 프로토콜에서는 티켓과 공개키 기반 암호화 알고리즘을 이용하여 대칭키 분배의 어려움을 해결하였다.[3.2절 Step 4, 5]

결과적으로 본 보안 프로토콜에서 발생하는 모든 통신 메시지는 암호화되어 송수신되므로 사용자 정보와 콘텐츠의 기밀성이 보장된다.

3.3.2 사용자 인증

멤버십이 요구되거나 유상으로 제공되는 서비스의 경우 피어의 신분을 확인하기 위한 수단은 반드시 강구되어 비정상적인 서비스 이용을 단절시켜야 한다. 본 보안 프로토콜은 수신자가 송신자에게 콘텐츠의 전송을 요청하고, 콘텐츠를 수신하는 과정에서 트래킹 서버에서 수신자에게 발급한 티켓을 이용하여 사용자 인증을 하게 된다.

참여하는 피어들에 대한 인증은 크게 두 가지 측면에서 이루어진다. 콘텐츠의 전송을 요청한 수신자에 대한 인증과 콘텐츠를 전송하는 송신자에 대한 인증이다.

○ 수신자 인증

수신자 인증은 콘텐츠의 전송을 요청하는 수신자가 트래킹 서버로부터 인증 받은 정상적인 사용자인지 송신자가 판단하는 과정을 말한다. 송신자는 티켓을 복호화하여 티켓에 기록된 수신자, 송신자 정보와 콘텐츠의 정보, 그리고 송신자의 공개키와 타임스탬프를 확인하게 된다.

티켓은 트래킹 서버의 개인키로 암호화되어 모든 사용자가 트래킹 서버의 공개키를 이용하여 티켓의 내용을 확인하는 것은 가능하지만 티켓을 생성하는 것은 불가능하다는 특성을 이용해, 만일 티켓에 기록된 수신자, 송신자 등의 정보가 일치한다면 수신자를 정상적인 사용자로 판별할 수 있게 된다.

○ 송신자 인증

수신자는 자신에게 콘텐츠를 전송하는 송신자가 정상적인 사용자인지 판단하여 콘텐츠의 수신을 결정할 필요가 있다.

수신자는 송신자의 사용자 인증을 위해 티켓에 기록된 송신자의 공개키를 이용한다. 공개키로 암호화된 메시지는 개인키를 이용해야만 복호화가 가능하다는 공개키 기반 암호화 알고리즘의 특성상 수신자가 생성하여 송신자에게 전송하는 암호화된 대칭키는 오직 송신자만이 자신의 개인키를 이용해 복호화가 가능하다. 따라서 정상적인 송신자가 아닐 경우, 대칭키의 확인이 불가능하므로 수신자는 송신자의 정상 유무를 판단할 수 있다.

3.3.3 접근 제어

P2P 네트워크는 각 피어가 콘텐츠 배포에 참여하므로 확장성이 뛰어나다는 이점이 있지만, 사용자의 관리, 콘텐츠 제어 등에는 어려움이 존재한다. 기존방식의 콘텐츠 공급자는 사용자를 여러 등급으로 나누어 차등요금을 부과하고 요금에 따라 콘텐츠 및 품질을 차별적으로 제공할 수 있었지만, P2P 네트워크에서는 이러한 방식을 그대로 적용할 수가 없다.

이러한 문제를 해결하기 위한 수단으로 본 보안 프로토콜은 수신자가 트래킹 서버에 송신자의 목록을 요청하고, 티켓을 수신하는 과정에서 티켓의 수를 제한하는 방법을 이용해서 차별적인 서비스를 제공할 수 있다.

Ⅳ. 구현 및 실험

4.1 구현

본 장에서는 논문에서 제안하는 보안 프로토콜의 실용성검증을 위해 안드로이드 기반의 스마트폰 플랫폼에 모듈로 구현을 하고 실제 스마트폰(삼성 갤럭시S)에 탑재하여 성능을 평가한다.

4.1.1 트래킹 서버

트래킹 서버는 그림 6과 같은 구조로 Java와 Eclipse를 이용하여 Windows 환경에서 구현되었으며, 각각의 모듈은 다음과 같은 역할을 수행한다.

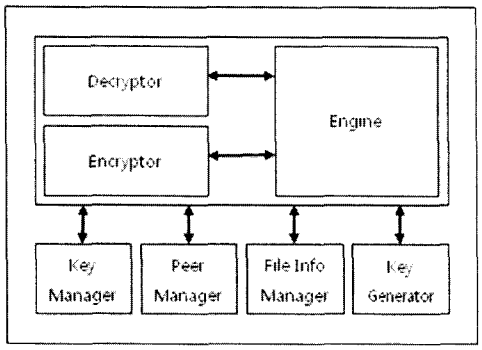

그림 6. 트래킹 서버 모듈구조
Fig. 6 Tracking server module architecture

Decryptor & Encryptor: 기밀성과 사용자 인증을 제공하기 위해 필요한 암호화, 복호화 과정을 담당한다. 암호화와 복호화를 위해 대칭키 기반 알고리즘으로는 DES, 공개키 기반 알고리즘으로는 RSA를 이용한다.

Key Manager: Decryptor와 Encryptor에서 사용되는 공개키, 대칭키를 관리한다.

Peer Manager: 등록되어있는 피어의 정보와 접속여부 등을 관리한다.

File Info Manager: 콘텐츠 검색과 해당 콘텐츠를 보유 중인 피어 정보를 관리한다.

Ticket Generator: 티켓 생성 및 관리를 담당한다.

4.1.2 피어

피어는 안드로이드 프레임워크 API(Level 8)[14]를 이용하여 안드로이드 2.2 기반으로 구현되었으며(그림 7), 각각의 모듈은 다음과 같은 역할을 수행한다.

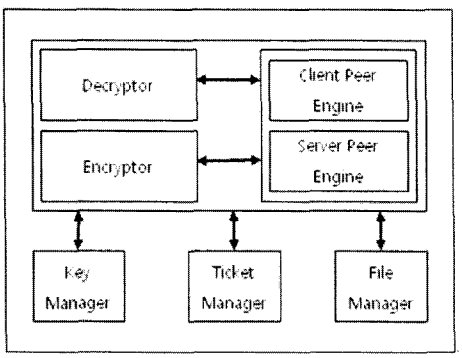

그림 7. 피어 모듈구조
Fig. 7 Peer module architecture

Decryptor & Encryptor: 암호화와 복호화를 위해 대칭키 기반 알고리즘으로는 DES, 공개키 기반 알고리즘으로는 RSA를 이용하였다.

Key Manager: Decryptor와 Encryptor에서 사용되는 공개키, 대칭키를 관리하며, 송신자와 통신하기 위한 세션키를 생성한다.

Ticket Manager: 서버 또는 수신자로부터 수신한 티켓을 관리한다.

File Manager: 수신하거나 송신하는 콘텐츠를 관리한다. 파일 조각(chunks)들을 통합하여 원래의 파일로 복구한다.

4.2 실험결과

실험은 WiFi(802.11g) 무선 환경에서 트래킹 서버 역할을 수행하는 데스크탑 PC와 피어 역할을 수행하는 스마트폰을 사용하였다. 각각의 제원은 표 2와 같다.

표 2. 실험 환경 제원
Table 2. Experiment environment

구분	제원
PC	Intel Core i5, 4GB RAM, 250GB HDD
스마트폰	삼성 갤럭시S, 안드로이드2.2, 1GHZ CPU, 512MB RAM, 802.11b/g/n

4.2.1 프로토콜 단계별 수행시간

보안 프로토콜의 각 단계의 수행시간을 비교하기 위해 그림 8과 같이 전체시간에서 각 단계가 차지하는 시간을 백분율로 나타내었다.

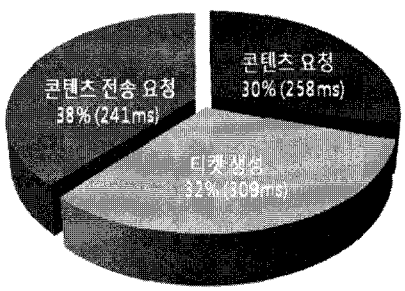

그림 8. 프로토콜 단계별 수행시간 비교
Fig. 8 Execution time of protocol steps

콘텐츠 요청: 수신자가 원하는 콘텐츠를 트래킹 서버에 요청하고, 콘텐츠의 송신이 가능한 송신자의 목록을 서버로부터 수신하는 과정으로, 보안 프로토콜의 Step 1, 2에 해당한다.

티켓 생성: 수신자는 콘텐츠의 전송을 요청할 피어들을 선택하여 트래킹 서버에 전송하고, 트래킹 서버는 티켓을 생성하여 수신자에게 전송한다. 프로토콜의 Step 3, 4에 해당한다.

콘텐츠 전송 요청: 수신자는 피어들에게 티켓을 전송함으로써 콘텐츠 전송을 요청하고, 각 피어는 티켓을 복호화하여 수신자에 대한 사용자 인증을 시행한다. 프로토콜의 Step 5부터 Step 6 이전까지에 해당한다.

그림 8에서 보듯이 각 단계에 드는 시간은 비슷함을 알 수 있다. 이는 본 프로토콜에서는 메시지 전송회수를 크게 줄임으로써 각 단계의 시간을 결정하는 주요 요인은 암호화와 복호화에 드는 시간이기 때문이다. 따라서 메시지 전송시간보다는 어떠한 알고리즘을 사용하는지가 성능에 중요한 영향을 준다. 또한 과정이 1초 이내에 수행이 되므로 스마트폰에서 실용성이 있다.

4.2.2 스워밍 피어 수에 따른 시간 측정

스워밍기법을 이용하여 여러 송신자들로부터 파일을 작은 조각들로 분할하여 병렬로 다운로드 받고 원래의 파일로 복원하는 경우, 스워밍에 참여하는 피어 수에 따른 처리시간을 측정하였다. 원본 파일의 크기는 8 MB이고, 이를 2 MB로 분할하고 송신자수가 1, 2, 4 일 때의 처리시간을 측정하였다.

그림 9에서 보듯이 스워밍에 참여하는 피어 수가 증가 할수록 처리시간이 줄어드는 것을 볼 수 있다. 단일 피어로부터 전송 받는 경우, 전체 파일을 받기 때문에 파일조각들을 통합하는데 걸리는 시간은 없으나 전송시간이 많이 걸린다. 이에 반하여 다수의 피어로부터 전송 받는 경우, 전송시간이 단축되나 각 피어들로 받은 조각들을 통합하여 원래의 파일로 복구하는 시간이 걸린 다. 4개의 송신자로부터 스워밍기법을 이용하면 단일 송신자 경우 보다 2배 이상의 성능향상을 보였다.

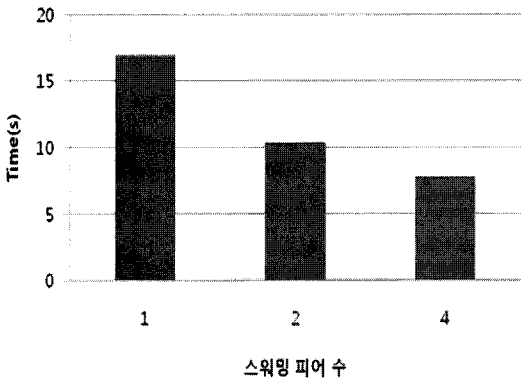

그림 9. 스워밍 피어 수에 따른 처리시간
Fig. 9 Processing time for different number of swarming peers

4.2.3 콘텐츠 크기에 따른 시간 측정

콘텐츠의 크기에 따른 시간 소모량을 측정하였다. 실험은 제안된 프로토콜의 암호화 및 복호화과정을 포함한 여섯 단계의 과정을 통해 피어가 보유중인 2-20MB 크기의 동영상 콘텐츠를 수신자에게 전송 완료하는데 걸리는 시간을 측정하였으며, 콘텐츠 검색과 피어 선택에 드는 시간은 제외하였다.

그림 10에서 보듯이 콘텐츠 크기에 따라 전송시간은 선형적인 증가를 보인다.

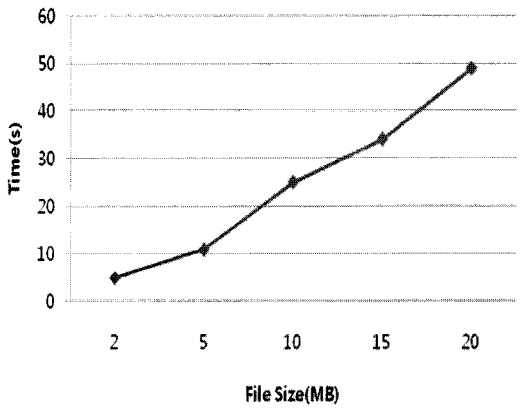

그림 10. 콘텐츠 크기에 따른 처리시간
Fig. 10 Processing time for different content size

V. 결 론

본 논문에서는 P2P 네트워크에서 스워밍 기법을 위한 보안 프로토콜을 제안하였다. 제안되는 프로토콜의 주요 기능은 콘텐츠의 기밀성, 사용자 인증 및 접근제어 제공이고, 티켓을 이용하여 사용자-서버-피어간의 메시지 오버헤드를 줄임으로써 스워밍 기법을 효율적으로 사용하는데 기여할 수 있다.

제안된 보안 프로토콜의 성능을 평가하기 위해서, 안드로이드 기반의 스마트폰 플랫폼 상에서 구현하고 실험을 수행하여 실용성을 검증하였다. 후속 연구과제로는 다중 소스로부터 콘텐츠를 병렬다운로드(Parallel Downloading) 하는 P2P IPTV, 클라우드 컴퓨팅, 분산 VoD 등에도 적용할 수 있을 것으로 보이며, 인센티브 등의 부가적 서비스를 추가할 수 있을 것이다.

참고문헌

[1] "Cisco Visual Networking Index: Forecast and Methodology, 2009-2014", Cisco Systems Inc., June 2010

[2] S. Han, Y. Xia, "Construction an optimal server set in structured P2P networks ", IEEE Journal on Selected Areas in Communications, 25(1), pp.170-178, 2007.

[3] S. Han, Y. Xia, "Optimal node selection algorithm for parallel download in overlay content distribution networks ", Computer Networks, 53(9), pp.1480-1496, 2009.

[4] http://www.dtc.umn.edu/mints

[5] http://cachelogic.net

[6] http://www.bittorrent.com

[7] http://www.pptv.com

[8] http://tvants.en.softonic.com

[9] http://www.sopcast.com

[10] S. Nair, E. Zentveld, A. Tanenbaum, "Floodgate: A micropayment incentivized P2P content delivery network", Proc. of International Conference on Computer Communications and Networks.2008.

[11] D. Sirivianos, J.H. Park, S. Jarecki, "Dandelion: Cooperative content distribution with robust incentives", Proc. of USENIX, pp.157-170, 2007.

[12] B. Yang, H.Garcia-Molina, "PPay: Micropayments for P2P systems", Proc. of ACM Conference on Computer and Communications Security, pp.300-310, 2003.

[13] P.Druschel, A.Nandi, T.Ngan, "Scrivener: Providing incentives in cooperative content distribution systems", Proc. of Middleware, pp.270-291, 2005.

[14] http://developer.android.com/sdk/android-2.2.html

저자소개

이관섭(Kwan-seob Lee)

2010년 명지대학교 컴퓨터공학과 학사
현재 명지대학교 컴퓨터공학과 석사과정

※관심분야 : 네트워크, 보안, 임베디드

이광식(Kwang-sik Lee)

1995년 한양대학교 경영학과 학사
2001년 삼성SDS
2007년 연세대학교 경영학과 석사
현재 신한금융지주회사IT팀부부장, 명지대학교컴퓨터공학과 박사과정

※관심분야 : IT전략, IT융합, 보안, 모바일컴퓨팅

이장호(Jang Ho Lee)

1990년 서울대학교 컴퓨터공학과 학사
1992년 서울대학교 컴퓨터공학과 석사
2000년 University of Michigan, Electrical Eng. and Computer Science 공학박사
2000년 IBM T.J.Watson Postdoctoral Researcher
2001년 (주)유비쿼스 수석연구원
2002년~현재 홍익대학교 컴퓨터공학과 부교수

※관심분야 : CSCW, 그리드 컴퓨팅, 모바일컴퓨팅

한승철(Seung-chul Han)

2003년 Purdue University
컴퓨터학과 석사
2007년 University of Florida
컴퓨터공학과 박사

현재 명지대학교 컴퓨터공학과 조교수
※관심분야 : P2P IPTV, VOD, Network Security, Android