P2P 네트워크 구조에 기반한 보안 모델 구현

김경석[0] 박진영 이구연
강원대학교 정보통신공학과
{kyung016, atomos}@cnclab.kangwon.ac.kr, leegyeon@cc.kangwon.ac.kr

Implementation of Security Model Base on Peer to Peer Network

Kyung-suck Kim[0] Jin-young Park Goo-yeon Lee
Dept. of Computer and Information and Telecommunication Engineering, Kangwon National University

요 약

본 논문은 P2P 네트워크 구조에 기반한 보안 모델 구현에 관한 것을 논하고 있다. 현재 많이 사용되고 있는 P2P 프로토콜 중에는 중계서버가 있어 각각의 peer가 데이터를 공유하고, 검색할 수 있게 하는 방법과 중계서버가 존재하지 않고 peer간의 통신만으로 구성되어지는 방식이 있다. 본 논문에서는 두 가지 방식의 장점을 이용해서 서버를 통하여 다른 peer의 정보를 얻어 올 수도 있으나 그렇지 못한 경우에도 캐쉬된 기존 정보에서의 자체 검색과 다른 peer와의 통신을 통해서 P2P 동작이 가능하도록 네트워크을 구성하였다.

이러한 P2P 네트워크 구조에서 취약할 수 있는 보안환경을 SSL(Secure Socket Layer)을 이용하여 웹(web)을 통해 서버와 통신하는 단계와 peer 간에 정보 교환을 위한 통신에 적용하였으며, 그룹키(group key)를 이용하여 보안 멀티캐스트 환경을 구현하였다.

1. 서 론

최근 주목받고 있는 네트워크 서비스로 P2P(Peer to Peer)가 있다. P2P는 중간의 서버를 거치지 않고 정보를 얻고자하는 peer와 정보를 가지고 있는 peer간에 직접 통신을 하는 네트워트 구조를 말한다.

클라이언트-서버 모델은 데이터 관리적인 면에서는 매우 효과적이지만 서버에 데이터가 집중되어 시스템 성능 저하가 발생할 수 있고 서버 장애 발생 시 전체 시스템이 다운되는 결과를 초래하지만 P2P환경은 서버에 의존적이지 않아서 일부 peer가 멈추어도 전체 시스템에 영향을 최소화 하는 장점을 가지고 있다.

또한 P2P는 클라이언트-서버 모델에 반하여 각각의 peer간 상호작용에 중점을 두어서 개발된 시스템으로 정보를 공유하는데 좀더 유연하게 설계되어 자유로운 네트워크를 구성할 수 있고 중앙 서버의 기능을 없애거나 약화시켜 사용자가 동등하게 참여하는 분산된 네트워크 서비스를 제공할 수 있다.

P2P는 위와 같은 클라이언트-서버 모델의 문제점을 해결하기 위해 제시된 미들웨어를 기반으로 한 분산형 클라이언트-서버 모델과는 차별화 되는 모델로 향후 인터넷 패턴을 바꿀 주요 기술로 평가되고 있지만 걸림돌이 작용하는 것이 정보보호에 관련된 문제이다. 이제는 단순한 음악파일 전송에서 벗어나 각종 중요한 디지털 정보 전송의 요구가 늘어남에 따라 적절한 보안 상태를 유지하지 못한다면 사취(interruption)뿐만 아니라 조작(manipulation)에 따른 심각한 피해 발생이 우려되므로 더욱 더 강화된 보안 기술을 필요로 하게 된다.

이러한 네트워크 환경에서 문제점으로 대두될 수 있는 무결성 확보를 위해서 SSL(Secure Socket Layer)과 그룹키(group key)를 이용하여 보안환경을 구현하였다.

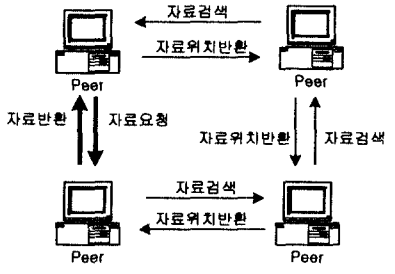
서론에 이어 2장에서는 P2P 네트워크 구조에 대해 설명하고 3장에서는 네트워크 모델을 설명하고 4장에서는 기본 구성 및 동작에 대해 설명하고 결론을 맺는다.

2. P2P 네트워크 구조

P2P 기술은 최초 컴퓨터 통신의 모델이지만 많은 변화를 하며 다양하게 발전하였다. 데이터 교환이 peer 들간에 직접적으로 이루어진다는 공통점이 있지만, 자료의 검색과 관리를 함에 있어서 서로가 차별점을 가지고 있어서 크게 순수 P2P형과 혼합 P2P형을 나눌 수 있다.

2.1 순수 P2P형

순수 P2P형은 중앙에 서버가 개입하는 부분 없이 노드들의 연결로만 구성된 네트워크 형태이다. 그림[1]은 순수 P2P의 모델을 보여주고 있다.



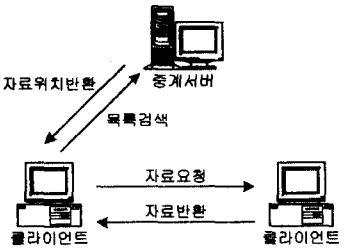
[그림 1] 순수 peer-to-peer형

이 방식은 중앙에서 매개하는 서버 없이 네트워크에 연결된 모든 노드들이 서로 통신하며 자료의 위치를 알아내고 자료를 복사해 오는 형태이기 때문에 하나의 노드에 문제가 생기더라도 다른 노드들에 영향을 주지 않

으므로 신뢰도가 높으며 대규모 시스템에서 병목으로 작용하는 중앙 서버가 없다는 점에서 높은 확장성을 갖는다는 장점이 있지만 이 방식에서는 자료 검색시에 순차적인 탐색 과정이 필요하므로 시간 지연이 커지는 문제가 있다. 대표적인 예로는 Gnutella와 Freenet이 있다.

2.2 혼합 P2P형

혼합 P2P는 클라이언트-서버 모델의 자취를 여전히 어느 정도 가지고 있으며 그림[2]는 혼합 P2P의 모델을 보여주고 있다.



[그림 2] 혼합 peer-to-peer형

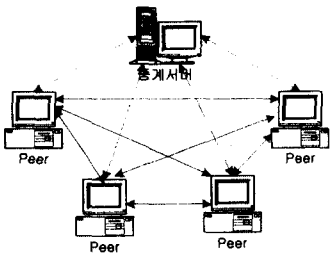
대규모 네트워크 구축에 사용된 클라이언트-서버 모델이라는 것은 계층구조이다. 하지만 대다수 소규모 사용자들은 계층구조의 하부에 위치하게 되므로, 이는 곧 인터넷 서비스 공급자 자신의 한계 또는 그들이 규정한 여러 가지 제약에 갇히게 되어 버리는 문제점을 해결하였다. 하지만 여러 사용자들이 갖고 있는 자료의 목록과 그 자료를 갖고 있는 노드의 위치 정보를 중앙 서버에 목록으로 갖고 있기 때문에 만약 서버에 문제가 생겼을 때에는 사용자들에서 서비스를 제공할 수 없게 된다. 대표적인 예로 냅스터가 있다.

3. 네트워 모델

서버가 없는 Gnutella 방식은 다른 peer와 데이터의 소재를 찾는데 어려움이 있고, 서버가 있는 Napster 방식은 중계 서버가 문제가 생길 경우 동작이 불가능해 진다는 단점을 가지고 있다.

여기서는 네트워크 구조를 두 가지 방식이 혼재 되어 있는 형태로 구성하였다. 서버를 통하여 다른 peer의 정보를 얻어 올 수 도 있으나 그렇지 못 할 경우에도 캐쉬된 기존 정보에서의 자체 검색과 다른 peer와의 통신을 통해서 P2P 동작은 여전히 가능하다. 다른 peer와 그들의 자료목록은 서버에서 관리되어 진다. 이 서버는 웹서버 형태로 존재하며 peer들간의 정보교환을 도와주는 것을 주목적으로 하고 있다. 정보교환에는 자료의 출처표시, 자료 검색의 편리성, 상대 peer의 신원확인과 동시에 익명성 보장 등이 반드시 필요할 것이며, 이러한 점들을 위해 보안환경과 자료검색 모듈, 멀티미디어 처리도구등이 추가적으로 필요하다.

새로운 peer가 참여하기 위해서는 서버에서의 인증을 통해서 자신을 등록해야 한다. 등록 후에는 서버에서의 자료검색이 가능해 지며, 검색한 자료를 이용해서 다른 peer에게 자료를 요청할 수 있다. 이때 서버나 다른 peer와의 통신은 SSL을 통한 보안 환경에서 이루어진다.



[그림 3] 구현 모델

서버에서 검색한 결과들은 캐쉬되어 보관되고, 서버로의 접속을 원하지 않는 경우나 접속이 불가능 할 경우 자체 검색에 사용된다. 인증 과정이나 자료 검색 등이 혼합 P2P형과 유사하지만 최초의 등록과정에서만 강요될 뿐이고, 이후에는 순수 P2P형으로의 통신도 가능한 네트워크 구조이다.

4. 기본 구성 및 동작

4.1 SSL

SSL은 메시지를 암호화하는 하나의 방법으로 TCP/IP 연결을 암호화하여 공개된 인터넷 환경에서 상호 신분확인 및 통신에 대한 보안을 제공한다. SSL은 암호화를 통한 데이터의 보안과 통신 주체간의 상호확인이란 문제를 공개키 암호학의 이론적 바탕 위에서 만들어 졌으며 TCP/IP계층과 어플리케이션 계층(HTTP, TELNET, FTP 등)사이에 소켓 계층으로 위치하여 보안 서비를 제공한다.

인증서의 내용을 명세 하는데는 여가가지 표준이 있으며 SSL은 ITU(International Telecommunication Union)에서 발표한 X.509를 지원하고 있다. 웹에서 SSL을 사용하는 경우에 브라우저는 handshake 과정에서 서버의 인증서를 클라이언트에게 보여주게 되며, 사용자가 서버의 인증서를 받아 들이는 경우에만 나머지 handshake 과정을 완료하고 메시지를 전송하게 된다.

기본적인 동작은 서버는 웹서버에서 동작하는 것으로 SSL의 설치만으로 https 프로토콜을 이용하여 소형에이전트들과 SSL통신을 가능하게 한다. 본 시스템에는 소형에이전트들의 서버프로그램과 클라이언트 프로그램에 SSL을 적용하여 동작하도록 하고 있다.

Java의 패키지인 JSSE(Java Secure Socket Extension)를 사용하여 서버 및 클라이언트 소켓 프로그램을 작성할 경우 인증서를 보관하는 파일(keystroe)이 필요하다. Java에서 제공되는 keytool은 소형에이젼트의 인증서를 import 할 때 사용된다. 암호화 알고리즘과 관련하여 JCE(Java Cryptpgraphy Extension), SSL과 관련하여 JSSE를 이용하였다.

4.2 보안 멀티캐스트

멀티캐스트는 송신자의 메시지가 여러 수신자에게 한번에 전송되도록 하여, 데이터의 중복 전송으로 인한 네트워크 자원의 낭비를 최소화할 수 있게 하며, 보안 멀티캐스트 환경은 하나의 그룹 키를 이용하여 특정 멀티캐스트 그룹 회원들에게 보내어지는 메시지를 암호화함으로써 멀티캐스트 통신을 보호하는 것이다. 이에 대한

연구로 Iolus, GKMP(Group Key Management Protocol)[1][2]등이 있으며 현재도 계속 개발되어 지고 있다.

GKMP는 멀티캐스트 그룹 회원들간 암호화 통신을 위해 대칭 키를 생성하여 사용한다. 각각의 멀티캐스트 그룹은 그룹 컨트롤러(Group Controller, GC)가 관리하게 된다. 최초의 그룹 생성은 생성하고자 하는 회원이 GC에게 그룹생성을 요청하게 되면 키 생성 과정을 통하여 그룹을 생성하게 된다. 이후, 그룹 회원 가입 요청을 받아 그룹 회원을 구성하게 된다.

GKMP는 계층적 구조를 가지는 다른 보안 멀티캐스트 기법들에 비하여 그 확장성에는 약점을 가지고 있지만, 프로토콜상의 안전성이나 그 구현방법에 있어서는 강점을 가지고 있는 프로토콜이며 그룹 회원 가입, 탈퇴 및 키 생성, 키 분배, rekey와 같은 프로토콜의 전반적인 관리는 GC가 모두 담당한다. 이 점이 최상위의 그룹 관리자와 중간 계층의 관리자를 두어 계층적 구조로 구성하는 다른 보안 멀티캐스트 환경 프로토콜과 구분되는 특징이다.

GKMP에서 그룹 키의 생성은 GC와 그룹 회원간에 키 교환 프로토콜을 이용하여 생성되고, 키의 분배는 GC에 의해 key packet의 형태로 각각의 회원들에게 분배되어진다. 여기서 key packet은 그룹 통신을 위한 그룹 키와 그 키를 회원들에게 전달하기 위해 암호화하는 key encryption key가 합쳐진 세션 키(session key)의 형태를 말한다.

보안멀티캐스트 시스템은 크게 GC의 기능을 담당하는 부분과 멤버의 기능을 담당하는 부분으로 나눌 수 있다. GC의 기능을 담당하는 부분은 다시 멀티캐스트 그룹의 모든 일을 컨트롤하는 CryptMulticastServer와 그 내부에서 그룹의 생성 및 삭제 등의 기능을 담당하는 GroupManager, 그룹키의 생성 및 분배를 담당하는 CryptMulticastClient 부분으로 나누어 변수들을 설정하여 기능을 수행할 수 있도록 하였다. 이러한 모든 기능은 각각의 모든 소형에이전트에서 수행 가능하며, 또한 중계 서버가 관장 할 수도 있다.

4.3 인증서버

인증서버란, 인터넷을 통한 데이터 전송에서 나에게 보낸 데이터의 송신자가 믿을 만한 사람인가를 확인할 수 있도록 인증해 주는 역할을 하는 서버를 말한다.

서버에서 하는 일은 크게 두 가지로 나눌 수 있으며, 그 첫 번째는 웹을 통해 서버에 접속한 사용자들에게서 회원 등록신청을 받고, 소형 에이전트로서 동작할 수 있는 응용프로그램을 다운로드 시켜주게 되며, 회원들의 인증서를 발급하는 일을 하게 된다. 서버는 웹을 통해 https 웹 프로토콜로 동작하게 된다. https는 http가 80번 포트를 사용것과는 달리 443포트를 사용한다.

인증서버를 구성하기 위해서 다양한 플랫폼에서 수행이 가능 하지만 여기서는 Solaris7를 이용하여 SSL과 TLS(Transport Layer Security)프로토콜을 만족하는 공개 소프트웨어인 OpenSSL과 웹서버 구성을 위해 Apache를 설치 사용하였다.

CA(Certificate Authority)를 운용하는 곳은 트리 구조로 되어있으며 최상부의 CA를 rootCA라고 하고 자기 자신이 생성한 인증 요구서를 자기 자신이 서명하여 개인이나, 기관에게 배포한다. rootCA로부터 인증을 받은 중계 인증기관은 인증 요구서를 rootCA로부터 인증받아 다른 사람이나 기관의 인증 요구서를 인증해 주는 역할을 한다. rootCA 생성과정을 살펴보면 openssl 명령을 사용하여 CA의 비밀키를 생성한 후 CA의 비밀키로 1024 bits의 보안강도를 가지는 RSA 키를 생성하도록 설정하고 비밀키 생성이 끝나면 생성한 키를 이용하여 selfsign된 인증 파일이 생성된다. 이렇게 만들어진 비밀키와 인증 파일은 알맞은 위치로 복사하고, 백업(backup)한다. Apache Server의 키 생성 및 인증과정 후에 웹서버를 가동하게 된다.

5. 결 론

P2P 구조에 기반의 시스템은 그 동안 주류를 이루었던 서버-클라이언트 구조에 기반의 시스템 기술로 대처하기 곤란했거나 비효율적이었던 응용 분야에 적용될 경우 큰 효과를 기대할 수 있다. 특히 개인용 컴퓨터의 보급이 증가하고 그 컴퓨터들의 능력이 향상될수록 이런 P2P 구조의 자료 공유 체계가 효용이 증가할 것이다.

예를 들면 방대한 자료을 서로 공유해야 할 필요가 있는 그룹에서 부서별로 세부 정보를 교환함에 있어 실무자간에 신속한 정보 교환은 상당히 중요하고 이 과정에서 신뢰성과 안정성은 무엇 보다도 우선이 되어야 한다.

하지만, 일괄적인 암호화 전송은 실제적으로 느끼기 못할 수도 있지만 암호화에 따른 추가적인 데이터 전송으로 인한 전송속도가 늦어질 수 있다. 따라서 암호화가 필요없는 정보나 멀티미디어 자료와 같은 대용량의 자료들은 암호화가 아닌 압축과 헤더와 같이 필요한 부분만 암호화하는 방법등이 추가적으로 요구된다. 그러기 위해서는 먼저 각각의 peer에 저장되어 있는 자료들의 분류가 먼저 선행되어진 후에 사용자의 선택에 의해서 암호화 통신이 이루어 질 수 있도록 보완한다면 더욱더 효과적인 정보의 공유가 이루어질 것이다.

이와 같이 컴퓨팅 환경과 응용 분야에 각별한 효용을 가질 수 있는 이 기술을 이용함으로써 정보 공유 기술과 문화 확산 및 효과적인 통신 서비스를 위한 핵심 기술 습득 등과 같은 분야에서 파급 효과를 기대할 수 있을 것이다.

참 고 문 헌

[1] H. Harney, and C. Muckenhirn "Group Key Management Protocol(GKMP) Specification. RFC 2093, July, 1997.

[2] H. Harney, and C. Muchenhirn "Group Key Management Protocol(GKMP) Archtecture", RFC 2094, July, 1997.

[3] "The Secure Sockets Layer Protocol-Enabling Secure Web Transactions" iVEA Technologies-SSl Rev: 1.2 2001.

[4] William Stalling, "Cryptography & Network Security: Principles & Practice 2nd edtion"에 따라 , Prentice Hall, 1998.