P2P 응용에서의 보안 기능 설계

[1]박진원[O] [1]이준형 [1]김명균
[1]울산대학교 컴퓨터.정보통신공학과
{[1]zwsonic@cnlab.ulsan.ac.kr[O], [1]comfuny@cnlab.ulsan.ac.kr, [1]mkkim@mail.ulsan.ac.kr}

Design of security in functionality P2P applications

[1]Zin-Won Park[O] [1]Joon-Hyung Lee [1]Myung-Kyun Kim
[1]Dept. of Computer Engineering and Information Technology, University of Ulsan

요 약

P2P 어플리케이션은 일대일 통신 혹은 그룹간의 공동 작업에 많이 사용된다. 이러한 어플리케이션은 메시지 전송 및 파일, 데이터의 공유기능을 가지고 있어 신뢰할 수 있는 상대방과의 통신을 전제로 하고 있다. 하지만 사용자의 의도에 따라 혹은 의도하지 않는 경우로 인하여 상대방과 그룹 전체에 악영향을 미칠 위험성도 가지고 있다. 본 논문에서는 이러한 일대일 혹은 그룹간의 안전하고 신뢰성 있는 통신을 위해 보안성을 지닌 P2P 어플리케이션을 설계해 보았다.

1. 서 론

오늘날 대다수의 인터넷 사용자는 메신저를 비롯한 많은 P2P 어플리케이션을 사용하고 있다. P2P 어플리케이션은 일대일 통신을 비롯하여 일대다 통신 기능을 가지고 있어 메시지 및 파일 전송 기능뿐만 아니라 데스크탑 공유기능까지 가지고 있다. 특정 단체들이 사용하는 그룹웨어 또한 이러한 P2P 통신을 하기도 한다. 이렇듯 인터넷 통신은 서비스를 원하는 사용자가 서비스 제공자의 서버에 접속하는 클라이언트-서버 개념을 넘어 클라이언트-클라이언트, 서버-서버의 개념으로 확산되어 가고 있다.

기업이나 혹은 단체에서는 그들만의 목적에 맞는 P2P 어플리케이션을 개발하여 사용자에게 그것을 사용하도록 하고 있다. 하지만 대다수의 어플리케이션들이 성능이나 이용의 편이성을 위해 보안을 고려하지 않은 채 개발되어 많은 문제점을 유발하였다. 한 예로 음악 파일 공유 프로그램으로 유명한 소리바다의 경우 상대방의 로컬 하드디스크의 내용이 노출되는 경우도 있었으며 악성 코드가 포함된 파일이 유포된 경우도 있었다. 또한 MSN 메신저의 경우 메시지가 텍스트 형태 그대로 전송되기 때문에 메시지 전달 과정에 노출되어 있으며 사용자가 조심하기만을 요구하고 있기도 하다.

따라서 P2P 어플리케이션 운용시 발생할 수 있는 위험성을 연구하고 개발 시점에서 이에 대한 방안을 고려한다면 보다 안전하고 신뢰성 있는 통신을 할 수 있다. 본 논문에서는 보안적 측면을 고려하여 P2P 어플리케이션을 설계해 보았다.

2. 기존 P2P 어플리케이션의 보안 문제점

현재 사용되고 있는 다수의 P2P 어플리케이션들은 보안을 고려하지 않고 제작되었기 때문에 많은 문제점을 가지고 있다.

암호화되지 않는 메시지들은 공개된 인터넷을 통과하므로 악의의 공격자에 의한 스니핑(sniffing) 공격에 의해 메시지와 전송되는 파일이 도청될 수 있다. 또한 통신하는 상대방이나 통신을 중계하는 중계 시스템에 대한 신뢰할만한 인증과정을 거치지 않을 경우 악의로 자신을 속이고 있는 공격자를 신뢰할 수 있는 사용자로 오인하고 통신을 하게 되는 수도 있다[2,3].

파일·공유 어플리케이션의 경우 어플리케이션의 버그나 디자인 오류 혹은 악성 코드의 유입으로 인하여 불특정 다수에게 자신의 로컬 저장장치의 데이터 목록이 노출될 수도 있으며 이것이 악의의 상대방에게 노출될 경우 심각한 손실을 가져올 수 있게 된다. 요즈음 대부분의 어플리케이션에서 제공하는 자동 업데이트 기능은 업데이트 서버와 신뢰할 수 있는 인증 절차를 거치지 않았을 경우 악의의 실행코드를 다운로드 할 수 있어 바이러스나 웜 같은 더 큰 확산을 일으킬 수 있을 뿐만 아니라 공격자에게 사용자의 시스템을 노출시키는 결과를 초래할 수 있게 된다[1,2,4].

기존의 많은 P2P 어플리케이션들은 이러한 보안 문제를 해결하지 못하고 있기 때문에 중요한 데이터의 전달을 원하는 사용자들로부터 외면 받고 있다.

3. P2P 어플리케이션 보안 요구사항

안전한 P2P 통신을 위해서는 인증과 데이터 암호화, 접근제어(Access Control) 기능을 갖추어야한다[1]. 인증은 사용자가 통신을 하는 상대방이 현재 인지하고 있는 상대방임을 확신할 수 있음을 말한다. 인증을 하지 않을 경우 A라는 사용자는 B라는 사용자와 통신을 하는 것으로 알고 있지만 실제로는 C라는 악의의 사용자가 자신을 B로 가장하여 A와 통신을 할 수 있다. 혹은 A와 B의 통신 중간에 C가 개입하여 A에서 B로 전송하는 데이터가 C를 거쳐 그대로 전달되거나 혹은 변조되어 전달될 수 있고 B에서 A로의 전달 역시 마찬가지다.

데이터 암호화는 인터넷 구간을 거쳐가는 데이터가 공격자에 의해 도청되지 않게 한다. 암호화되지 않는 데이터는 인터넷에 공개된 많은 스니핑 도구들에 의해 쉽게 도청되어 신용카드 번호, 은행 계좌의 비밀 번호, 기밀문서 같은 기밀 데이터들이 외부로 유출되어 커다란 피해를 입을 수 있다.

접근제어 기능은 자신과 통신을 하고자 하는 상대방의 접근의 허용 여부를 결정하는 것이다. 만약 파일을 공유하는 어플리케이션의 경우 접근제어 기능을 가지지 않는 어플리케이션은 공유를 원치 않은 사용자도 접근이 가능하게 된다. 이것을 꺼려하여 공유를 하지 않으면 공유하고 싶은 다른 사용자에게도 파일을 공유할 수 없게 된다. 이러한 공유데이터의 관리 문제나

통신 요청을 차단하고자 하는 요구를 해결하기 위한 접근제어 기능이 필요하다.

또한 피해를 입게 되더라도 그 피해를 최소화할 수 있어야하며 차후 피해를 예방하고 이후의 대처를 위한 로그 기록 역시 중요한 문제이다.

4. P2P 어플리케이션 설계

2, 3절에서 언급한 P2P 어플리케이션의 문제점은 어플리케이션 설계 및 구현과정에서 그대로 반영되어야 하며 개발자가 미처 예측하지 못한 공격 형태에 대해서도 심각한 피해로의 확산을 예방 할 수 있어야 한다[5]. 본 절에서는 이러한 관점을 바탕으로 여러 가지 보안기능을 갖춘 모듈을 이용하여 P2P 어플리케이션을 설계하였다.

4.1 전체 시스템 구성

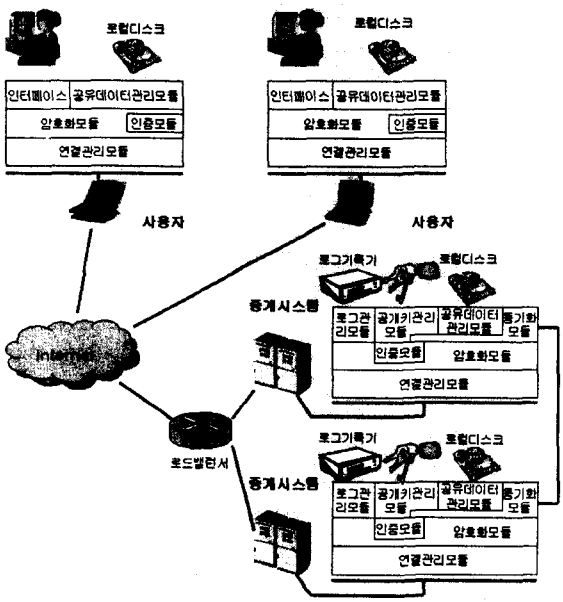

그림 1 전체 시스템 구성도

그림 1은 P2P 어플리케이션이 사용하는 전체 시스템 구성을 나타낸다. 모든 사용자는 P2P 어플리케이션 실행 후 로그인과 동시에 중계 시스템으로 접속을 하게 된다. 로그인은 사용자와 중계 시스템 사이의 공개키 알고리즘을 이용하여 세션키를 주고받는 것으로 이루어진다. 사용자는 중계 시스템의 공개키가 포함된 X.509 형식의 인증서를 보유하고 있으며 중계 시스템은 모든 사용자의 공개키를 보유하고 있다. 세션키는 DH(Diffie-Hellman)키 분배 알고리즘과 DSA(Digital Signature Algorithm) 알고리즘을 사용하여 생성한다. 그림 2는 세션키 분배 과정을 보여주고 있다.

4.2 사용자 측 연결관리모듈

연결관리모듈은 중계 시스템과의 연결 혹은 다른 사용자와의 연결을 관리하는 역할을 한다. 연결은 로그인 이후에는 중계 시스템과 지속적으로 유지되는데 그 이유는 세션키가 자주 생성되는 것을 피하고 NAT (Network Address Translators) 환경에 있는 사용자가 연결 요청을 수신할 수 있게 하기 위해서

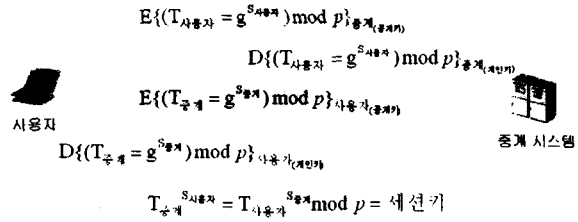

그림 2 세션키 분배 과정

이다. 또한 통신하는 상대방에게 자신의 IP 주소가 노출되지 않는 방법이기도 하다. 만약 상대방 사용자 혹은 사용자 그룹을 신뢰할 수 있다면 연결은 중계 시스템을 통하지 않고 직접 연결을 할 수도 있다. 이것은 요청 수락시 사용자간의 협의에 의해 결정된다.

4.3 암호화 모듈

암호화 모듈은 Openssh2 라이브러리를 이용해 암호화를 수행한다. Openssh 라이브러리는 최초 연결 설정시 공개키 알고리즘을 이용해 세션키를 생성하고 이 세션키를 이용해 데이터 암호화를 수행한다. Openssh2가 지원하는 공개키 알고리즘은 ssh-dss, ssh-rsa, x509v3-sign-rsa, x509v3-sign-dss, spki-sign-rsa, spki-sign-dss, pgp-sign-rsa, pgp-sign-dss 가 있다[6]. 암호화 알고리즘은 양측이 모두 보유한 알고리즘 중 하나가 자동으로 선택되어 진다. 데이터를 암호화 하는데 사용되는 암호 알고리즘은 3des-cbc, blowfish-cbc, twofish-cbc, twofish192-cbc, twofish128-cbc, aes256-cbc, aes192-cbc, aes128-cbc 이 있다[6]. 세션키는 유효시간을 가지고 있기 때문에 유효시간이 지난 후에는 새로운 세션키를 생성하든지 시간내에 통신을 마치게 하여 키 보호를 더욱 강하게 하였다.

4.4 인증모듈

인증은 사용자와 중계시스템 사이에 인증서를 이용해 이루어진다. 사용자는 중계 시스템의 인증서만을 보유하고 있음으로 해서 공개키 관리를 용이하게 하였다. 다만 중계 시스템은 모든 사용자의 인증서를 보유하고 있어야만 한다. 사용자측 인증모듈은 사용자의 개인키와 중계 시스템의 공개키를 이용하여 상호 인증하는 역할을 수행한다.

4.5 사용자 측 공유데이터 관리모듈

사용자의 데이터를 보호하고 관리하는 역할을 한다. 본 논문의 P2P 어플리케이션은 기존의 공유 디렉토리를 지정하는 형식이 아니라 프로세스의 최상위 디렉토리를 공유하고자 하는 데이터를 저장하는 디렉토리로 제한함으로 해서 혹시 있을지 모르는 오버플로우 공격으로부터 피해를 최소화할 수 있다. 또한 상대방이 데이터를 요청할 경우 사용자의 허가를 얻어 데이터로의 접근을 허용한다.

4.6 인터페이스

어플리케이션 인터페이스는 사용자가 전송하는 메시지를 암호화 모듈로 전달하고 연결 요청이나 공유데이터로의 접근 요청을 사용자에게 알려주는 역할을 한다. 악의의 데이터가 유입되어 시스템에 피해를 주는 것을 막기 위하여 인터페이스는 다른 사용자 혹은 중계 시스템으로부터 다운로드 한 데이터를 실행하지 못하도록 보호하는 역할을 한다. 바이너리 데이터의 경우 메모리의 데이터 공간에 저장하여 악의의 코드로부터 실행되지 않도록 하며 디스크에 저장하거나 화면에 표시할 수만 있다.

또한 악의의 스크립트로부터 시스템을 보호하기 위해 스크립트 실행을 하지 않는다. 대량의 데이터 유입으로 인해 메모리가 고갈되는 것을 막기 위해 유효 데이터 크기를 주기적으로 체크하는 역할을 한다.

4.7 로그관리모듈

중계 시스템은 사용자가 로그인하여 세션키를 취득하는 과정, 두 사용자의 직접연결 혹은 간접 연결을 설정하는 과정, 공유데이터의 중계 시스템으로 업로드, 사용자가 로그아웃하는 과정 등 시스템 전체에서 일어나는 모든 과정을 로그관리모듈을 통해 기록한다. 로그는 쓰기와 읽기는 가능하지만 삭제나 수정은 불가능한 CD-ROM이나 DVD에 실시간으로 기록함으로써 관리자도 이를 삭제하거나 수정할 수 없다. 따라서 이 기록은 차후 법적인 증거자료로의 효력도 지니고 있을 수 있다.

4.8 중계 시스템 측 연결관리모듈

중계 시스템의 연결관리 모듈은 연결을 유지하고 있는 모든 사용자들과 통신을 한다. 사용자 A와 사용자 B가 중계 시스템을 통해 연결되어 있을 경우 사용자 A가 사용자 B로 전송하는 데이터는 사용자 A와 중계 시스템 사이의 세션키로 암호화되어 있으므로 암호화 모듈에서 이를 복호화하고 다시 사용자 B와의 세션키로 암호화한 다음 연결관리모듈을 통해 사용자 B에게 전송한다. 그룹 통신의 경우 중계 시스템은 한 사용자로부터 수신한 데이터를 그룹 내의 다른 사용자들에게 모두 전송해 준다.

4.9 중계 시스템 측 공유데이터 관리모듈

중계 시스템은 공동작업시 사용자가 부재중일 때에도 다른 사용자에게 부재중인 사용자의 데이터를 전달받을 수 있게 하기 위한 공유데이터를 저장하고 있다. 이것은 사용자가 직접 업로드 해야하며 업로드 할 때 해당 데이터에 접근할 수 있는 사용자들을 지정할 수 있다.

4.10 공개키 관리모듈

사용자들의 공개키를 관리한다. 사용자들의 공개키는 유효시간을 가지기 때문에 주기적으로 키의 유효시간을 체크하여 만료가 된 키의 경우 사용자에게 미리 갱신할 수 있도록 알려주는 역할을 한다. 로그인을 원하는 사용자가 인증 과정을 거칠 때 인증 모듈에게 해당 사용자의 공개키를 전달하는 역할도 한다.

4.11 동기화 모듈

중계 시스템은 수많은 사용자들과 연결을 유지하고 암호화와 복호화 연산을 많이 하기 때문에 복수의 호스트를 이용하여 구현하여야 한다. 동기화 모듈은 서로 다른 중계 시스템에 접속되어 있는 사용자들간의 연결 요청이나 데이터를 전달해 주는 역할을 한다. 중계 시스템이 요청 받은 공유데이터가 로컬 디스크에 없을 경우 해당 데이터를 가진 중계 시스템으로부터 데이터를 읽어와 전달해 준다. 또한 사용자들의 공개키를 모든 중계 시스템이 동기화 될 수 있도록 해 준다.

4.12 로드밸런서

많은 수의 사용자가 중계 시스템을 이용할 경우 사용자들을 중계 시스템의 부하에 따라 분산시켜주는 역할을 한다. 한번 정해진 사용자-중계 시스템 사이의 연결은 로그아웃할 때까지 유지하는 역할도 수행한다.

5. 결론 및 기대효과

본 논문에서는 기존의 보안을 고려하지 않은 P2P 어플리케이션 문제점에 관해 알아보았다. P2P 어플리케이션은 그 특성상 메시지와 데이터 이동의 보안이 매우 중요한 이슈이다. P2P 어플리케이션에서는 통신을 하는 상대방에 대한 인증 절차를 통하여 상대방임을 확신할 수 있어야 한다. 이것은 X.509 인증서를 사용하는 공개키 암호 알고리즘을 이용함으로써 해결하였다. 또한 주고받는 메시지와 데이터는 공개키 암호 알고리즘보다 속도가 빠른 비밀키 암호 알고리즘을 이용하여 암호화한 후 전송해야 도청으로부터 안전할 수 있다. 암호화에 사용되는 비밀키는 특정 세션에 한정된 세션키를 사용함으로 해서 보안성을 높일 수 있다. IP주소의 보호와 키 관리의 문제점, NAT환경의 사용자를 위해 통신은 중계 시스템을 통해 하는 것이 보다 더 효율적이다. 또한 사용자의 데이터를 검증하는 과정 대신에 이에 대한 맹목적인 신뢰를 하지 않음으로 해서 개인 사용자 및 전체 사용자들로의 피해 확산을 예방할 수 있게 하였다. 중계 시스템은 사용자들의 세션키 생성 기록 및 사용현황을 실시간으로 기록하여 차후 발생할 수 있는 후속조치를 위해 수정이 불가능한 로그 기록을 하여 신뢰도를 높였다. 그리고 공유되는 데이터를 보호하기 위해 프로세스의 권한을 제한하는 방법을 사용하였다.

P2P 어플리케이션은 활용도가 높으나 신뢰성이 떨어지는 관계로 중요한 데이터 전달에는 사용되지 못하고 있는 것이 사실이었다. 하지만 신뢰할만한 암호 알고리즘을 이용한 사용자 인증 및 데이터 암호화와 발생할 수 있는 피해를 최소화할 대책을 마련해 둠으로 해서 어플리케이션의 사용자는 좀 더 안심하고 P2P 어플리케이션을 활용할 수 있을 것이다. 이러한 효과로 원격의 공동작업이 활발해질 수 있으므로 많은 비용과 노력을 절감할 수 있을 것이다.

6. 참고문헌

[1] David Barkai, *"Peart-to-Pear Computing Technologies for Sharing and Collaborating on the Net"*, Intel Press, 2002

[2] Wooyoung Kim; Graupner, S.; Sahai, A., *"A secure platform for peer-to-peer computing in the Internet"*, System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on , 7-10 Jan. 2002 Pages:3948 - 3957

[3] Xiaolin Pang; Catania, B.; Kian-Lee Tan, *"Securing your data in agent-based P2P systems"*, Database Systems for Advanced Applications, 2003. (DASFAA 2003). Proceedings. Eighth International Conference on , 26-28 March 2003, Pages:55 - 62

[4] Welch, V.; Siebenlist, F.; Foster, I.; Bresnahan, J.; Czajkowski, K.; Gawor, J.; Kesselman, C.; Meder, S., *"Security for Grid services"*, High Performance Distributed Computing, 2003. Proceedings. 12th IEEE International Symposium on , 22-24 June 2003, Pages:48 - 57

[5] Surridge, M.; Upstill, C., *"Grid security: lessons for peer-to-peer systems"*, Peer-to-Peer Computing, 2003. (P2P 2003). Proceedings. Third International Conference on , 1-3 Sept 2003, Pages:2 - 6

[6] T. Ylonen, T. Kivinen, *"SSH Transport Layer Protocol"*, SSH Communications Security Corp, Internet-Draft, Sep. 18, 2002