

P2P를 이용한 패치 분배 시스템의 효율적인 패치 파일 분배 방안 연구

*이수영 **이인용 ***문종섭

정보경영공학전문대학원

*leesuyoung1@naver.com

An Efficient Patch File Distribution Method for PMS Using P2P

*Lee, Su-Young **Lee, In-yong ***Moon, Jong-Sub

Graduate School of Information Management & Security

요약

패치 분배 시스템은 중앙 집중 방식의 구조를 가지고 있다. 이런 중앙 집중 방식의 구조는 많은 사용자가 존재할 경우 영향력이 서버에 집중된다. 따라서 한 노드로 미치는 영향을 최소화하기 위해서 본 논문에서는 P2P 통신 구조의 유연함과 분산적인 성격을 이용한 패치 분배 시스템 구조를 제시한다. 또한 패치 파일이 정상적으로 안전하게 전송 하도록 P2P 프레임 워크를 제안 한다. 이 프레임 워크는 파일에 대한 내용변조에 민감하게 반응 하도록 설계 되었다. 그러므로 우리는 웹, 바이러스, 그리고 해킹에 입구가 되는 취약점을 없애기 위한 패치 파일을 빠른 속도로 네트워크의 구성원에 전파 할 수 있게 된다.

1. 서론

응용 프로그램을 개발할 경우 소프트웨어는 복잡도, 확장성 그리고 연결성의 증가로 결점을 가지게 된다. 개발 이후에 이런 결점을 보완하기 위해 시스템 개발자는 패치를 배포 한다. 하지만 이런 결점을 보완하는 패치는 역으로 의도적인 공격자에게 시스템의 취약점을 노출 한다. 그래서 네트워크 환경에서 패치를 분배 할 경우 고려해야 할 무결성, 기밀성 그리고 가용성이 존재 한다[1].

현재 패치는 소프트웨어의 복잡성에서 나오는 취약점으로 인하여 불가결하게 필요하다. 하지만 이를 효과적으로 분배하고 관리함에 있어서 적지 않은 노력이 필요하다. 그리고 패치의 효과가 발휘되기 위해서는 취약점이 발생 했을 경우, 웹이 전파 되는 속도보다 빠르게 패치 파일이 전파 될 수 있어야 한다. 특히 사람의 행동에 의해 전파되는 이메일 웹 같은 경우에 신속히 대응할 수 있어야 하며, Active Worm의 새로운 보고에서는 적어도 하루 안에 시스템의 모든 자원에 패치가 이뤄 질 수 있어야 한다.

현재는 서버-클라이언트(S/C) 구조에 새로운 트리를 추가하여 확장성을 추가하는 구조의 시스템이 패치 시스템의 주종을 이루었다[2]. S/C 구조의 시스템에서는 클라이언트의 수와 요청의 수가 증가 할수록 서버에서 처리해야 하는 작업은 지수 적으로 증가 한다. 이런 구조적인 부하 문제를 어떻게 효율적으로 분산 하는가가 중요한 문제 중 하나이다. 따라서 본 논문에서는 P2P를 이용해서 패치 분배 시스템의 부하를 효과적으로 처리하는 문제를 다루려 한다. 특히 P2P의 분산적인 협업 구조를 이용하면 패치 파일 분배의 가용성을 향상 시킬 수 있다. 그리고 패치 파일 분배의 기밀성 유지를 위해 암호화된 패치를 전송하여 기밀성을 유지 했다. 그리고 P2P패치 구조에서 중요한 보안적

문제점인 파일의 무결성 문제를 해결할 프레임워크를 제안하려 한다.

다음 2장에서는 패치 시스템과 P2P에 관한 관련 연구를 살펴보고 패치 시스템에 필요한 사항을 확인해 보며, 3장에서는 P2P를 이용한 패치 분배 시스템의 프레임 워크를 제안 하고, 마지막으로 4장에서 결론을 맺으려 한다.

2. 관련 연구

본 장에서는 패치 분배 시스템의 기존 연구를 살펴보고, P2P의 구조적인 부분과 패치 분배 시스템에서 고려해야 할 사항을 살펴보려 한다.

2.1. 패치 분배 시스템

확장성을 고려한 계층적 패치 분배 시스템 프레임워크 설계에서는 새로운 서브 트리의 추가 시에 DB에 저장되어 있는 정보를 어떻게 변환 할지를 고려하여 새로운 트리에 대한 새로운 통합함을 다루고 있으며, S/C 구조에서 서버가 받는 영향력에 보완은 미비 하다[2]. 일반화된 보안패치 분배 및 관리 시스템을 위한 프레임워크 설계에서는 Primary Server와 Secondary Server로 나누어서 보안패치 분배 관리 시스템을 관리 한다. Primary Server 에서는 주로 시스템의 관리와 사용자 인증 및 사용자 패치 정보에 대하여 다루지며 Primary Server 에 집중되는 파일 분배의 과부하를 줄이기 위해 Secondary Server를 이용해 트리를 형성하여 부하 분산을 활용했지만 이 또한 최하위 노드가 증가하면 증가할수록 패치 시스템의 운영 유지에 어려움을 갈질 수 있으며, 능동적인 웹(Active Worm)과 같은 빠른 전파의 웹에서 필요한 패치 분배의 가용성이 떨어지게 된다[3]. XML을 이용한 보안패치 중앙 관리 시스템 설계에서는 DB의 내용을 클라이언트에 XML로 저장 하므로 서버에 DB에 접속하는 부하를 줄였지만, 정작 서버에 많은 부하인 파일 IO를 일으키는 패치 파일 분배에서는 크게 기여 하지 못하

* 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음.

(IITA-2006-(C1090-0603-0025))

고 있다[4].

위에서 살펴본 대로 관련된 논문에서는 패치 시스템의 구조를 다루는데 있어 중앙 집중 방식을 이루고 있다. 일반적으로 관리 시스템에서는 매우 유용한 구조임에 틀림이 없다. 그렇지만 본 논문에서는 패치 파일을 일부 사용자만 독점해야할 정보가 아닌 누구나 쉽게 접근 하며, 응용프로그램의 취약점에 안전하고 신속히 적용 시킬 수 있는 정보의 하나로서 생각한다. 정보화 사회의 인프라 구조를 위협하는 위협들로부터 기업의 자산과 개인의 정보를 보호할 수 있도록 패치를 제공해야 한다고 생각 하며, 따라서 본 논문에서는 어떻게 하면 패치 파일을 안전하게 분배하고 신속하게 적용하는 방안에 대하여 고려했다.

2.2. P2P 관련 연구

서버의 사용자가 급증할수록 이에 대응하는 서버의 능력은 반감하기 마련이다. 이런 문제점을 해결하기 위해 피어-투-피어(P2P) 구조가 부각되기 시작했다.

이런 대표적인 예가 냅스터, 소리바다, 그누텔라 등이 있다[5]. 이런 P2P는 일반적인 S/C 구조 보다 자료의 탐색, 다운로드가 복잡하기 때문에 효과적인 방법이 필요하다. 현재 연구된 방안에는 냅스터처럼 자료의 정보만 서버가 가지고 파일 전송은 Peer들끼리 하는 기존의 S/C 구조와 Peer 의 혼합형 centralized system 이 있다. 이 모델은 기존의 S/C 모델에 비해 파일 전송과 같은 오버헤드가 큰 작업들이 서버를 거치지 않는 장점이 있다. 그누텔라(Gnutella)는 decentralized system 의 한 종류로서 KaZaA, eDonkey 등이 있으며 그누텔라에 평판(Reputation)을 첨가한 Xrep 방식의 P2P 구조가 있다[6].

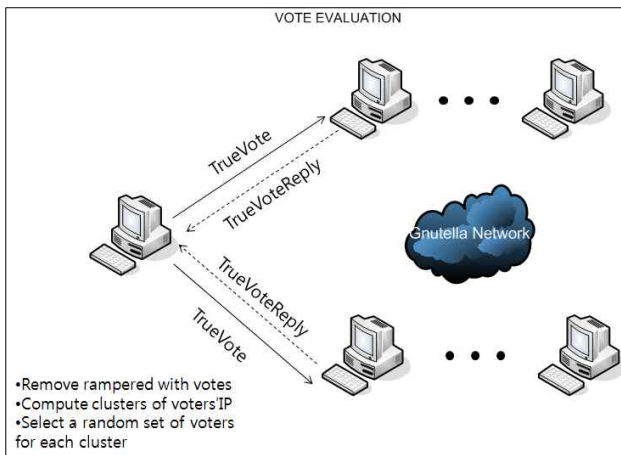


그림 1 평판 방식의 투표 과정 [6]

그리고 울트라피어(ultra-peer) 방식의 leaf-node들에서 네트워크 신뢰도와 여러 기능을 보아 leaf들 중에 울트라피어를 선출하여 leaf들 약 10~100대를 서브 트리로 구성하여 관리하는 모델 구조가 있다. 울트라피어는 브로드 캐스팅 되는 ping이나 Query문으로부터 서브 트리를 대표하여 대응 한다. 실제로 울트라피어는 QRP(query routing protocol) 테이블과 실제로 어떤 파일이 리프에 의해 공유 되는지에 대해서는 헤시로만 가지고 있다. 울트라피어가 대표하여 leaf들을 관리하기 때문에 효율적으로 파일을 검색할 수 있다. 그누텔라의 단점을 피하기 위해서 질의할 때는 브로드캐스팅을 하지만 응답할 때는 최적 경로의 라우팅으로 응답을 하므로 응답속도를 향상 시킬 수 있는 동적 라우팅 모델 구조가 있다. 그누텔라의 응답메시지 프로토콜인 Query Hit

와 Pong 메시지는 라우팅 한다. 질의가 브로드캐스팅 될 때, 인접한 요청자의 IP 주소와 함께 UUID(Universally Unique Identifier)는 지역적으로 저장된다. 이렇게 저장된 UUID는 해당하는 Query Hit는 인접한 요청자에게 라우팅 되며 최종적으로 요청자에게 라우팅 된다[7]. 분산 해시 테이블(DHT: Distributed Hash Table)은 구조적인 분산 P2P 알고리즘 이다[8]. 분산 해시 테이블의 대표적인 프로토콜로는 Chord가 있다[9]. P2P 시스템에 참여하는 모든 피어들 중 이웃하는 피어들에 대한 라우팅 정보만을 유지하여 간단한 검색 알고리즘에 의해 $O(\log N)$ 크기의 범위만 검색함으로써 모든 파일에 대한 빠른 검색이 가능하다. 그리고 정확한 일치 질의로 검색하는 것 보다 키워드 검색이 더 일반적이고 희귀한 파일에 대한 정확하고 신속한 검색이 가능하다. 하지만 분산 해시 테이블 구조를 유지하는 오버헤드가 크며 정확히 일치하는 파일에 검색만 가능하기 때문에 확장성이 떨어진다. 그 외에 P2P의 한 구조로서 비구조적인 GIA가 있다. 이는 그누텔라의 기초적인 문제점을 해결하기 위해 제안 되었다. 그 문제는 많은 질의가 있을 경우에는 노드에 미치는 영향력이 매우 크기 때문에 시스템이 멈출 수 있다는 단점이 있다. 따라서, GIA는 많은 질의를 응답하며, 시스템 크기가 증가함에 따라 그 기능을 유지할 수 있도록 확장성과 오버헤드를 감소시키는 데에 초점을 맞추어 모델링 되었다[10]. GIA는 파일을 검색할 때 랜덤 워크 방식을 변형하여 사용한다. 랜덤 워크는 질의의 충분한 Query Hit가 있을 때 까지 질의 메시지가 각 스텝(step)으로 임의로 선택된 이웃에게 전송하는 방식이다[11]. 이 방식은 초기 그누텔라가 가지는 희귀 파일에 대한 응답이 높아지는 것을 해결하지 못하고 희귀한 파일을 못 찾을 수 있는 문제점을 여전히 가지고 있다[12].

현재 위와 같은 많은 방법으로 P2P에 관련되어서 연구가 진행되어지고 있다. 본 논문은 이런 P2P 연구 중에서 오픈된 패치 분배 시스템에 적용하여 더욱 효과적으로 패치를 분배 하려 한다. 그래서 다음장에서 우리는 Xrep을 이용해서 평판 관리하는 P2P를 패치 분배 시스템에 맞게 적용하려 한다. 이런 P2P는 Peer들 끼리 평판을 관리하기 때문에 패치 파일에 정보뿐만 아니라 각 Peer들에 대한 위험 평판을 관리할 수 있는 장점을 가진다.

2.3. 패치 시스템에 필요한 고려 요소

본 장에서는 앞에 나온 P2P 관련 연구에서 선택한 Xrep에 필요한 고려 사항과 이를 적용 할 때 패치 분배 시스템에 고려사항 제시하고자 한다.

- 기밀성 : 이 요건은 단순히 보안패치 분배 및 관리 시스템만이 아니라 모든 것이 공개되어 있는 인터넷 환경에서 일정한 목적을 가지고서 중요한 일련의 작업들을 처리하는 과정에서는 가장 기본적으로 해결되어야 하는 문제이다. P2P로 파일을 검색하고 부하 없이 파일을 전송하지만 어느 누구나 내가 필요한 파일을 알 필요는 없을 것이다. 따라서 본 논문에서 제안하는 모든 패치 파일에 대한 기밀성은 유지 되어야 한다.
- 무결성 : P2P를 이용한 패치 분배 시스템에서는 파일의 변조에 대해 민감하게 반응해야 한다. 패치 파일에 대한 신뢰도는 완벽히 유지해야한다. 패치 파일이 변조 되었을 경우 악의적인 사용자가 의도하는 취약점을 모든 시스템 내부의 자원에 전파 시킬 수 있기 때문에 시스템이 워 전파 프로그램이 될 수 있기 때문이다. 그래서 패치 파일에 대한 인증을 제공할 수 있는 인증 체계가 필요하다. 따라서 이는 프레임 워크 제안에 가장 중요한 요소가 될 것이다.

- **가용성**: 가용성은 패치 파일을 얼마나 신속하게 패치 관리 시스템이 관리하는 네트워크 자원들에게 배포하는 지가 문제이다. 패치 관리 시스템에 새로운 패치가 존재 할 경우 이를 인증하고 빠르게 이웃 노드에 전파할 수 있어야 한다.
- **범용성**: 보안패치 분배 및 관리 시스템이 관리하는 네트워크 환경에는 일반적으로 하나 이상의 상이한 운영체제를 가진 시스템들로 이루어져 있다. 이러한 환경에서 단순히 하나의 운영체제에 대한 보안패치 서비스를 제공하는 것은 의미가 없으며, 필수적으로 멀티 플랫폼 환경을 지원할 수 있어야만 한다. 또한 네트워크 환경은 동적 IP, 사설 IP환경 등 매우 다양한 형태를 띠고 있기 때문에 보안패치 분배 및 관리 시스템이 범용성을 갖추기 위해서는 다양한 네트워크 환경을 지원할 필요가 있다.
- **편리성**: 보안패치 분배 및 관리 시스템을 통한 일련의 작업들을 분명히 매우 중요한 작업이다. 그러나 사용자의 특수한 사정으로 인하여 분배받은 즉시 보안패치를 설치할 수 없어서 일정한 시간 만큼의 설치 기한 연장을 요구할 수 있으며, 일부 사용자들은 분배 받은 보안패치를 설치할 때마다 승인을 받지 않고 백그라운드에서 자동으로 설치할 것을 요구할 수도 있다[3].

P2P를 패치에 적용할 때는 검색 자가 필요한 데이터를 어떤 Peer가 가지고 있는지 알 수 있는 방법을 제공해주어야 한다. 그리고 검색을 이용해서 찾은 패치 파일에 대한 무결성도 보장해 주어야 한다. 따라서 평판(Reputation)과 인증 서버를 이용한 하이브리드 방식으로 구성 하려 한다[13,14,15]. 패치 파일 무결성에 대한 검증은 권한 있는 서버에서 처리를 하거나 패치 파일에 대한 평판이 가장 높은 수준을 가지고 있을 때 추가적인 검증절차를 할 수 있도록 구성해야할 것이다. 서버에서 패치 파일에 대한 무결성을 검사할 때에는 패치 파일에 대한 직접적인 정보를 이용하는게 아니라 해시값을 이용해서 무결성을 검증한다. 이렇게 검증된 패치 파일에 대해서 가장 네트워크 환경 평판이 좋은 Peer로부터 다운(PUSH) 받아서 패치 분배의 가용성을 보장한다. 네트워크 평판이 좋지 못하더라도 네트워크 구조상 내부 네트워크에 패치 파일이 존재하면 이를 이용할 수 있도록 한다. 자원을 다운 받을 때에는 기밀성을 유지할 수 있도록 암호화된 파일을 전송하도록 한다. 새로운 패치 파일에 대한 전파 방식도 고려해야 보면, 관리자 또는 평판이 높은 Peer가 패치를 게시하고 이를 전파 할지, 아님 Peer에 존재하는 모든 패치 파일을 패치에 적용할지를 패치 관리 시스템을 구성하는 방식에 따라 확인해야할 것이다.

3. 패치 P2P 프레임워크

패치 시스템에 P2P구조를 적용하기 위해서 우리는 위에서 몇 가지 가정을 했다. 패치를 누구에게나 전파가 되어야 하고 패치의 무결성이 확인되어야 한다는 것이었다. 소리바다나 Napster, KaZaA 등의 P2P구조에서는 파일에 대한 어떠한 정보도 얻을 수 없다는 단점이 존재 한다. 이는 악성 사용자가 바이러스나 웜을 패키징하여 일반적인 파일에 삽입할 경우 일반적인 사용자는 이를 확인할 방법이 없다. 따라서 본 논문은 최근 P2P에서 해결방안으로 제시하는 평판 기반의 시스템을 차용 하려 한다. 그 중 대표적인 Xrep-시스템은 각 개인이 다른 사용자들에 대한 경험과 정보를 저장하고 이에 대하여 사용자의 요구가 있을 시 이를 공유하도록 구성 되어 있다. 이 때 각 사용자(servent)는 자신의 공개키를 hash한 값과 자료의 내용을 hash한 값을 두 가지를 이용한다. 그리고 각 사용자는 두 가지의 평판을 관리하는데 하나는 사

용자 평판과 자료에 대한 평판을 관리한다[14].

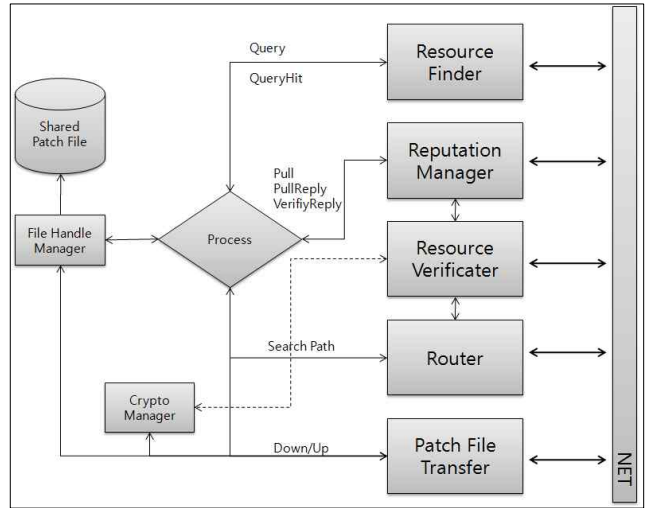


그림 2 P2P 패치 프레임 워크

P2P패치 분배 시스템에서는 다음과 같은 프로토콜을 이용해서 패치 파일에 대한 무결성을 지원하며, 만약 관리자를 두거나 울트라 피어가 인증하는 파일에 대하여 확인을 하고 싶을 경우도 고려하였다.

단계	내용
1	네트워크 대상 자원 찾기
2	자원 선택
3	자원 및 대상 평판 확인
4	자원 검증과 대상 확인
5	최적 제공자 확인 및 검증
6	패치 파일 받기

표 1 프로토콜

기준에 S/C 구조에 비교해서 평판 관리 시스템을 이용하는 P2P구조에서는 패치 파일에 대한 무결성 검증이 최우선적으로 해결 되어야 한다. 따라서 다음과 같이 평판 시스템에 검증 서버를 추가하는 하이브리드 방식의 P2P 구조를 제안 한다.

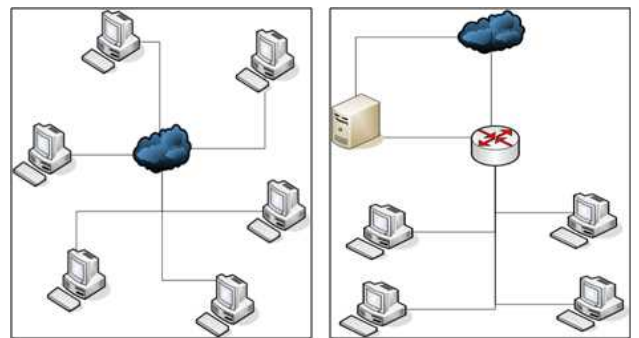


그림 3 구조도

P2P를 이용한 패치 시스템은 위와 같이 두 가지 가능한 구조를 고려했다. 왼쪽에서처럼 각 servent가 존재하여 완벽한 Peer-to-Peer로 구성되며 각각 패치를 게시하고 각 노드에 대한 평판을 수행 한다. 오른쪽 그림은 패치 파일에 대한 인증 서버를 두고 관리자가 내부 네트워크를 관리하고 패치 파일에 대한 무결성을 인증해 준다.

프로토콜 순서에서 첫 번째, P2P를 이용한 패치 시스템에서 원하는 대상 패치 파일 자원을 찾는 단계이다. 이는 모든 노드에 브로드 캐스팅 된다.

두 번째, 검색된 자원에 대한 선택 단계이다. server에 필요한 자원을 검색하여 필요한 패치 파일만 선택한다.

세 번째 과정에서는 선택된 패치 파일에 대한 평판에 대해 확인 과정을 거친다. 특히 여기서는 패치 파일에 대한 무결성과 각 server의 평판을 동시에 확인해서 일정 수준 이하의 패치 자원 및 지원자는 탈락시킨다.

네 번째, 만약 관리 및 울트라 server가 존재하면 패치 파일에 대한 검증과 평판을 함께 검증을 시도 한다.

다섯 번째, 이렇게 선출된 패치 파일과 제공 server에 대한 환경 평가를 실시하여 가장 최적으로 패치 파일을 제공할 수 있는 server를 검색한다. 만약 관리 및 울트라 server가 존재한다면 여기서 최적의 경로를 라우팅 해줄 것이다.

여섯 번째, 최종적으로 패치 파일을 다운로드하고 적용하는 단계이며 패치 파일에 대한 전송 단계에서 기밀성을 유지하기 위해 암호화를 수행한다.

4. 결론

현재 대부분 패치는 S/C 구조로 되어 있다. 하지만 이런 구조는 제로데이 공격에 능동적으로 대처하기 어렵다. 그래서 본 논문에서는 패치 파일에 대한 빠른 전파와 부하 분산을 고려해서 P2P시스템 평판 구조를 이용해서 Peer 끼리 패치 파일을 전파함에 패치 파일에 대한 무결성을 확인하는 과정을 추가 하였다. 따라서 신속히 모든 Peer에게 안정한 패치 파일을 전파할 수 있도록 구성 하였다. 앞으로는 패치 시스템에 적용하여 사용자 인증에 평판 시스템을 도입하여 각 노드에 대한 인증을 확인 하도록 구성해야 할 것이다.

5. Reference

[1] Greg Hoglund, Gray McGraw, "EXPLOITING SOFTWARE: How to break code", Addison Wesley Professional, pp. 44~53, 2004

[2] 김윤주(YunJu Kim), 이상원(SangWon Lee), 손태식(Tae-Shik Sohn), 문중섭(Jong-Sub Moon), 서정택(JungTaek Seo), 윤주범(JuBum Yun), 박응기(EungKi Park), "확장성을 고려한 계층적 패치 분배 시스템 프레임워크 설계", 한국정보과학회 2004년도 봄 학술발표논문집 제31권 제1호(A), pp. 199~201, 2004. 4

[3] 이상원(Sangwon Lee), 김윤주(Yun-Ju Kim), 손태식(Tae-Shik Sohn), 문중섭(Jong-Sub Moon), 서정택(Jung-Taek Seo), 이은영(Eun-Yong Lee), 이도훈(Do-Hoon Lee), "일반화된 보안패치 분배 및 관리 시스템을 위한 프레임워크 설계", 한국정보과학회 2004년도 가을 학술발표논문집 제31권 제2호(I), pp. 502~504, 2004. 10

[4] 김윤주(YunJu Kim), 이상원(SangWon Lee), 손태식(Tae-Shik Sohn), 문중섭(Jong-Sub Moon), 서정택(JungTaek Seo), 이은영(Eun-Yong Lee), 박응기(EungKi Park), "XML을 이용한 보안패치 중앙 관리 시스템 설계", 한국정보과학회 2004년도 가을 학술발표논문집 제31권 제2호(I), pp. 505~507, 2004. 10

[5] D.Milohicic, V.Kalogerki, R.Lukose, and K.Ngaraha, J.Pruyne, and B. Richard, "Peer-to-Peer computing", Technical Report HPL-2002-57, HP Laboratories, March 2002.

[6] E. Damiani, S. Vimercati, S.paraboschi, P. Samarati, F. Violante, "A reputation-based approach for choosing reliable resources in peer-to-peer networks," Conference on Computer and Communications Security archive Proceedings of the 9th ACM conference on Computer and communications security, pp. 207~216, 2002.

[7] http://www.dcs.ac.kr/~iraklis/fyp_report/fyp_report.html

[8] B. Karp, S. Rantnasamy, S.Rhea, and S. Shenker, "Spurring Adoption of DHTs with OpenHash, a Public DHT Service", In Proc. Of IPTPS, 2004 ...

[9] F.DABEK, E. BRUNSKILL, M. F. KAASHOEK, D. KARGER, R. MORRIS, I. STOICA, AND H. BALAKRISHNAN, "Building peer-to-peer systems with Chord, a distributed location service", In Proceedings of the 8th IEEE workshop on Hot Topics in Operating Systems (HosOs-VIII), May 2001.

[10] Q. LV, S. RATNASAMY, and S. SHENKER, "Can Heterogeneity Make Gnutella Scalable", In Proceedings of the 1st International Workshop on Peer-to-Peer Systems(IPTPS '02), Mar. 2002

[11] Q. LV, P. CAO, E. COHEN, K. LI, AND S. SHENKER, "Search and Replication in Unstructured Peer-to-Peer Network", In Proceedings of 16th ACM International Conference on Supercomputing(ICS'02), June 2002.

[12] 안진호(Jinho Ahn), "파일공유를 위한 P2P 네트워크에서의 검색매커니즘", 한국정보기술학회지 제4권 제1호, pp. 75~82 2006. 12

[13] 조남수(Nam Su Jho), 김우환(Woo Hwan Kim), 윤효진(Hyo Jin Yoon), 이인석(In-Sok Lee), 천정희(Jung Hee Cheon), 김태성(Tae-sung Kim), 진승현(Seung-hun Jin), 추정균(Kyung-kyun Choo), "P2P 환경의 자기 평판 관리 시스템", 정보보호학회논문지 Vol.14 No.2, pp. 35~47, 2004. 4

[14] E. Damiani, S. Vimercati, S.paraboschi, P. Samarati, F. Violante, "A Reputation-Based Approach for Choosing Reliable Resources in Peer-to-Peer Network," Conference on Computer and Communications Security archive Proceedings of the 9th ACM conference on Computer and communications security, pp. 207~216, 2002.

[15] B. Ooi, C. Liau, K. Tan, "Managing Trust in Peer-to-Peer Systems Using Reputation-Based Techniques", The 4th International Conference on Web Age Information Management (WAIM), August 2003. Keynote Paper.