개선된 Kerberos를 이용한 P2P 인증 시스템에 관한 연구

김종우*, 하태진**, 조경옥**, 한승조**
*조선대학교 전자공학과, **조선대학교 정보통신공학과

Study on The P2P Authentication System Using Advanced Kerberos

Jong-woo Kim*, Tae-jin Ha**, Kyoung-ok Cho**, Seung-jo Han**
*Department of Electronic Eng. Chosun Univ.
**Department of Information Communication Eng. Chosun Univ.
E-mail : mmm@7.co.kr

본 연구는 산업자원부의 지역혁신 인력양성사업의 연구결과로 수행되었음.

요 약

본 논문에서는 개방형 네트워크상의 P2P 시스템에서 강력한 상호 인증을 위한 알고리즘을 제안하였다. P2P 시스템의 특성에 맞게 인증을 위하여 서버의 부담을 최소화할 수 있는 알고리즘을 제안한다. 본 논문에서는 강력한 인증 알고리즘인 Kerberos를 P2P 시스템에 적합하도록 개선하였다. Kerberos의 티켓 승인 서버의 기능을 상대측의 Peer에 부여함으로써, 티켓 승인을 위한 트래픽 및 부하량을 감소시킬 수 있다. Peer에 티켓 승인 서버의 기능을 부여하기 위하여 티켓 승인을 위한 일회용 키값을 발행함으로써, 더욱 강력한 상호 인증과 서버의 부담을 최소화 할 수 있다. 또한 P2P 시스템의 특성에 적합하도록 시간위주의 인증 만료 시간보다. 사용 횟수와 인증 만료 시간 모두를 이용하여 인증 제한 값을 주었다.

키워드
Kerberos, 인증, P2P, 보안

Ⅰ. 서 론

현재까지의 인터넷은 Server와 Client 구조 위주의 컴퓨팅 기술의 발달이 주도되어 왔으나, 최근 Peer to Peer(P2P) 구조 위주의 컴퓨팅 기술이 연구되고 있다. Server와 Client 구조로 이루어진 컴퓨팅은 Server에 집중되어진 부하로 인해서 효율적인 Server 리소스를 사용하지 못했다. 그래서 서버 리소스의 부하를 감소시키고, 효율적인 서버 운용을 위해서 분산되어진 Peer에 직접 접근하여 원하는 리소스를 직접 얻는 P2P 시스템이 도입되었다.

P2P(Peer to Peer)란 인터넷에서 서버에 의존하지 않고 정보의 요구자와 제공자간의 컴퓨터를 직접 연결시켜 데이터를 공유할 수 있게 해주는 기술가 그 기술을 응용해서 제공되는 서비스를 말한다[1]. 인터넷에서 정보 검색을 위해서 검색엔진을 이용하여 찾아야하는 기존 방식과는 달리 인터넷에 연결되어 있는 모든 개인 컴퓨터로부터 직접 정보를 검색하고 제공받을 수 있는 기술이다.

P2P 서비스를 사용하기 위해 연결되는 가상의 공유 시스템은 개방 시스템이기 때문에 근본적으로 보안상의 문제점을 안고 있다[2]. 그래서 적절한 보안 수준을 유지하기가 쉽지 않다. 그러므로 부정 사용자의 접근, 보안과 개인정보 노출, 컴퓨터 바이러스 확산, 비 공유 영역에 대한 침해(일종의 해킹)와 같은 부당한 침해에 대한 문제점이 심각하게 대두될 수 있다. 이러한 환경에서 안전한 콘텐츠 공유를 위하여 강력한 상호 인증 기능은 필수적이라 할 수 있다.

Ⅱ. P2P에서의 보안과 Kerberos

2.1 P2P 시스템 환경에서 정보보호

기업이나 개인이 P2P를 이용하는 것은 불특정다수의 이용자에 대해 자사의 PC의 자원이나 서비스의 이용을 가능하도록 하기 위해서이다. 따라서 이것에 의해 새로운 취약성이 발생한다. P2P 네트워크의 사용은 악의적인 소프트웨어를 전달하는 능력뿐만 아니라 악의적인 소프트웨어에 의해 통신하는 프로토콜의 사용까지도 허가한다.

그런데, P2P 소프트웨어는 일반적으로 방화벽에 의해 방지되지 않는다. 왜냐하면 이것이 중앙 디렉토리 서비스 또는 다른 서밴트들과 송신연결을 하기 때문이다. 일단 송신 연결이 발생하면, 중앙 디렉토리 서비스 또는 서밴트는 클라이언트로 정보를 전달할 수 있다.

이 같은 방법은 P2P 네트워크를 불법적으로 사용하도록 수행되어진다. 예를 들면, 악의적인 위협은 P2P의 중앙 서버를 통해 등록될 수 있고 특정한 파일의 목록을 전달할 수 있다. 정보의 수집과 시스템의 제어는 방화벽을 우회하고 해커의 익명성을 보증하는 방법으로 수행된다.

2.1 Kerberos 인증방식

Kerberos는 MIT에서 Athena 프로젝트이 일환으로 개발된 인증 서비스이다[3][4]. Kerberos는 분산망에서의 인증 시스템으로서 프로세스인 클라이언트가 특정 사용자를 대신하여 검증자에게 사용자의 신분을 확인시켜 주기 위한 일련의 암호화된 메시지를 교환하는 과정이다. 현재 Kerberos는 일반적으로 두 가지 버전이 사용된다. 버전 4[MILL88, STEI88]가 가정 널리 사용된다. 버전 5[kohl94]는 버전 4의 보안 결함 몇 가지를 수정했고, 인터넷 표준의 초안으로 발표되었다(RFC1510).

Kerberos의 동작은 AS(Authentication Server)라는 인증 서버, TGS(Ticket Granting Server)라는 티켓 승인 서버, 그리고 클라이언트(C:Client)와 서버(S:Server)간의 티켓 발행 단계, 서비스 승인 단계를 각각 거쳐서 이루어지는 것이 기본적인 것으로, Kerberos V.4의 단계별 동작 설명은 다음과 같다.

(a) 인증 서비스 교환

- [단계1](메시지1)

$C \rightarrow AS : ID_C \| ID_{tgs} \| TS_1$

- [단계2](메시지2)

$AS \rightarrow C : E_{K_c}[K_{c,tgs} \| ID_{tgs} \| TS_2 \| lifetime_1 \| Ticket_{tgs}]$

$Ticket_{tgs} = E_{K_{tgs}}[K_{c,tgs} \| ID_c \| AD_c \| ID_{tgs} \| TS_2 \| lifetime_1]$

(b) 티켓-승인 서비스 교환

- [단계3](메시지3)

$C \rightarrow TGS : ID_s, Ticket_{tgs}, Authenticator_c$

$* Authenticator_c = E_{K_{c,tgs}}[ID_c \| AD_c \| TS_3]$

- [단계4](메시지4)

$TGS \rightarrow C : E_{K_{c,tgs}}[K_{c,s} \| ID_s \| TS_4 \| Ticket_s]$

$Ticket_s = E_{K_s}[K_{c,s} \| ID_c \| AD_c \| ID_s \| TS_4 \| lifetime_3]$

(c) 클라이언트/서버 인증 교환

- [단계5](메시지5)

$C \rightarrow S ; Ticket_s, Authenticator$

$* Authenticator = E_{K_{c,s}}[ID_c \| AD_c \| TS_5]$

- [단계6](메시지6)

$S \rightarrow C ; E_{K_{c,s}}[TS_5 + 1]$

III. P2P를 위한 개선된 Kerberos

P2P 시스템의 시장이 급속하게 성장함에 따라 보안의 위험성도 증가하고 있다. Peer간의 허가받지 않은 자의 접근 방지와 신뢰성 있는 상호 인증이 절실히 필요하다. 그러나 기존의 인증 방법을 P2P 시스템에 그대로 적용한다면, 클라이언트와 서버 중심의 인증 방식은 P2P 시스템에 잘 맞지 않을 뿐 아니라, 인증을 위한 서버의 부하량과 트래픽의 증가로 서버의 부하량과 트래픽 분산이라는 P2P 시스템의 개념에 맞지 않는다. 이에 본 논문에서는 인증시 서버의 부담을 최소화하고, 최대한 Peer를 활용할 수 있는 알고리즘을 제안한다.

본 논문에서는 P2P 시스템에서 강력한 인증을 위하여 Kerberos를 P2P 시스템에 적용할 수 있도록 개선하였다. 제안한 알고리즘은 기존의 P2P 시스템을 그대로 활용하여 Kerberos를 적용할 수 있는 알고리즘이다. 인증 서버(AS)의 기능은 P2P 시스템의 메인서버가 역할을 맡게 된다. 그리고 P2P 시스템에서 Peer는 서밴트(Servant=Server+Client)의 개념으로 Peer간에 서로 서버와 클라이언트의 기능을 모두 가지고 있다. 티켓 승인 서버(TGS:Ticket Granting Server)의 기능은 각각의 Peer가 하게 된다. 즉, 제안된 인증 알고리즘은 메인 서버가 인증 서버(AS)를, 파일을 제공하는 Peer가 티켓 승인 서버(TGS)를, 파일을 받고자 요청하는 Peer가 클라이언트의 기능을 한다. 제안된 알고리즘의 상호 인증 과정은 클라이언트 Peer와 인증 서버간의 티켓-승인 티켓 발행 단계, 서비스-승인 티켓 발행 단계, 서비스 승인 단계를 거쳐서 이루어진다.

3.1 제안된 알고리즘의 동작 과정

본 논문에서 제안하는 개선된 Kerberos 인증 알고리즘의 단계별 동작은 다음과 같다. P2P 시스템은 메인 서버와 Peer A, Peer B로 이루어져 있다. 메인 서버는 모든 Peer들의 ID와 Password를 관리하고 있고, 인증 서버(AS)의 역할을 한다. Peer A는 파일이나 정보를 필요로 하고, 요청하는 사용자이다. 그리고, 여기서 Peer B는 정보나 파일을 제공하는 Peer로써 티켓 승인 서버(TGS)와 서버의 역할을 같이하고 있다. 다시 말하면, Peer B와 TGS B는 같은 Peer B이다.

[단계1](메시지1)

$Peer\ A \rightarrow AS : ID_a \| ID_b \| TS_1$

*Peer A*가 *Peer B*에 접근하여 파일을 다운로드

하고자 할때는 ID_a, ID_b, TS_1를 AS에 보내어 서비스를 요청한다. 여기서, ID_a는 사용자의 신분을 알리기 위한 *Peer A*의 *ID*이고, ID_b는 엑세스를 원하는 *Peer B*(*TGS B*)의 *ID*이다. *Peer B*로부터 티켓을 구하기 때문에, 직접 ID_b를 사용하였다. TS_1는 *Timestamp* 정보로 현재 서비스를 요구하는 시각을 나타낸다. TS_1 정보는 replay 공격을 방지할 수 있다.

[단계2](메시지2)

$$AS \rightarrow Peer\ A : E_{K_a}\ [K_{a,tgsb} \| ID_b \| TS_2 \| lifetime_1 \| Ticket_{tgsb}]$$

$$AS \rightarrow Peer\ B(TGS\ B) : E_{K_b}\ [K_{tgsb} \| ID_a \| TS_2]$$

$$* \quad Ticket_{tgsb} = E_{K_{tgs}}\ [K_{a,tgsb} \| ID_a \| AD_a \| ID_b \| TS_2 \| lifetime_1]$$

AS는 클라이언트와 *Peer B*(*TGS B*)간의 세션키($K_{a,tgsb}$)와 $Ticket_{tgsb}$를 각각 생성하여 *Peer A*의 패스워드를 일방향 해쉬함수에 적용하여 구한 사용자 키 K_a로 암호화하여 *Peer A*에게 전달한다. $K_{a,tgsb}$는 *Peer A*와 *Peer B*(*TGS B*)간의 안전한 암호통신을 위한 일회용 세션키로 AS에서 생성한다. $K_{a,tgsb}$는 K_a로 암호화된 메시지 내부에 있기 때문에 *Peer A*만이 $K_{a,tgsb}$를 알 수 있다. K_a는 *Peer A*의 패스워드로부터 유도된 비밀암호키로서 패스워드를 일방향 해쉬 함수 MD5를 이용하여 구한다. TS_2는 $Ticket_{tgsb}$가 발행된 시간이며, *lifetime1*는 $Ticket_{tgsb}$의 유효시간을 나타낸다. 여기서, AD_a는 $Ticket_{tgsb}$를 사용할 *Peer B*의 IP 주소를 사용한다.

또한 동시에 AS는 *Peer B*(*TGS B*)만을 위해 생성한 일회용 키 K_{tgsb}와 ID_b, TS_2를 *Peer B*(*TGS B*)의 패스워드를 일방향 해쉬함수에 적용하여 구한 사용자 키 K_b로 암호화하여 *Peer A*에게 전달한다.

[단계3](메시지3)

$$Peer\ A \rightarrow Peer\ B(TGS\ B) : Ticket_{tgsb},\ Authenticator_a$$

$$* \ Authenticator_a = E_{K_{a,tgsb}}\ [ID_a \| AD_a \| TS_3]$$

*Peer A*는 AS로부터 수신하여 구한 $K_{a,tgsb}$를 이용하여 생성한 $Authenticator_a$와 $Ticket_{tgsb}$를 *Peer B*(*TGS B*)에게 전송한다. 여기서 $Authenticator_a$는 $Ticket_{tgsb}$의 유효성을 검증하기 위한 정보로 *Peer A*와 $Ticket_{tgsb}$간의 세션키 $K_{a,tgsb}$로 ID_a, AD_a, TS_3를 암호화한 암호문이다. TS_3는 인증자가 생성된 시간을 나타내는 정보로서, 재생 공격을 방지하기 위하여 매우 짧은 유효시간을 준다.

Peer B(*TGS B*)는 단계2에서 받은 일회용 비밀키인 K_{tgsb}를 이용하여 $Ticket_{tgsb}$를 복호화하여 $K_{a,tgsb}$, ID_a, AD_a, ID_{tgs}, TS_2, $lifetime_1$를 복구한다. 그 다음 $K_{a,tgsb}$를 사용하여 인증자를 복호화한 후, *Peer B*(*TGS B*)는 $Ticket_{tgs}$에서 복구된 ID_a, AD_a와 인증자로부터 복구된 ID_a, AD_a가 서로 일치 하는지 확인함으로써, *Peer A*의 정당성을 확인하게 된다.

[단계4](메시지4)

$$Peer\ B(TGS\ B) \rightarrow Peer\ A : E_{K_{a,tgs}}[K_{a,b} \| TS_4 \| Ticket_b]$$

$$* \quad Ticket_b = E_{K_b}\ [K_{a,b} \| ID_a \| AD_a \| ID_b \| TicketNum \| downloadcount]$$

Peer B(*TGS B*)는 $Ticket_{tgsb}$로부터 구한 $K_{a,tgsb}$를 이용하여, *Peer B*와 *Peer A*와의 세션키 $K_{a,b}$, 그리고 $Ticket_b$를 암호화하여 *Peer A*에게 보낸다. 여기서 ID_b는 *Peer B*의 *ID*를, TS_4는 $Ticket_b$가 발행된 시간을 나타낸다. $Ticket_b$는 $K_{a,b}$, ID_a, AD_a, ID_b, *TicketNum*, *downloadcount*를 *Peer B*와 AS가 공유하고 있는 비밀키인 K_b(*Peer B*의 패스워드를 일방향 해쉬함수에 적용하여 구한 사용자 키)로 암호화한 암호문이다. $Ticket_b$내의 *TicketNum*는 $Ticket_b$의 고유번호로써 *Peer B*의 $Ticket_b$의 유효기간까지 메모리에 저장된다. *downloadcount*는 파일을 다운받을 수 있는 개수를 의미한다. P2P 시스템에서는 파일이나 정보를 받기 위하여 많은 시간이 소요될 수도 있다. 이러한 P2P의 특성상 Ticket의 유효시간보다 다운로드 횟수를 제한하는 것이 바람직하다. downloadcount 값은 Peer가 임의로 설정할 수 있다(여기서는 Peer B).

[단계5](메시지5)

$$Peer\ A \rightarrow Peer\ B\ ;\ Ticket_b,\ Authenticator$$

$$* \ Authenticator = E_{K_{a,b}}\ [ID_a \| AD_a \| TS_5]$$

*Peer A*는 *Peer B*(*TGS B*)와의 세션키 $K_{a,tgsb}$를 이용하여 암호문으로부터 *Peer A*와 *Peer B*용의 세션키 $K_{a,b}$를 복구한 후, 이를 이용하여 구한 인증자 *Authenticator*를 $Ticket_b$와 함께 서버 S에 전송한다. *Authenticator*는 $K_{a,b}$로 ID_a, AD_a, TS_5를 암호화한 값이다. $Ticket_b$는 *Peer A*가 AS에 의해 인증 받았음을 확인하기 위한 정보로, 재사용이 가능하다. ID_a는 $Ticket_{tgsb}$의 정당한 소유자의 *ID*이고, AD_a는 *Peer A*의 IP주소로서 다른 주소를 갖는 Peer로부터의 $Ticket_{tgsb}$의 사용을 방지하기 한다. 인증자 *Authenticator*는 $Ticket_b$를 제시하고 있는 Peer A가 $Ticket_b$를 발행받은 정당한 사용자임을 입증하기 위한 정보이며, TS_5는 인증자가 생성된 시간이다.

[단계6](메시지6)

$$Peer\ B \rightarrow Peer\ A\ ;\ E_{K_{a,b}}[TS_5+1]$$

*Peer B*는 $Ticket_b$에서 복구된 ID_a, AD_a가 $K_{a,b}$를 이용하여 인증자로부터 복구된 ID_a, AD_a가 일치하는지 검사하여 *Peer A*의 정당성을 확인한다. 정당한 *Peer A*이면, *Peer B*는 자신이 정당한 *Peer B*임을 증명하기 위해 메시지 6을 *Peer A*에게 보내며, *Peer A*는 이를 복호화한 후, TS_5+1를 확인하여 *Peer B*의 정당성을 확인한다.

Ⅳ. 알고리즘의 효율성과 분석

본 논문에서 제안한 알고리즘은 P2P 시스템에 적합하게 개선된 Kerberos 알고리즘을 제안하고 있다. Kerberos 알고리즘은 서버와 클라이언트의 인증을 위하여 인증 서버(AS)와 티켓 승인 서버(TGS)가 필요하다. 서버의 부담을 분산시키기 위한 개념으로 발달된 P2P 시스템에서는 인증을 위하여 Kerberos를 그대로 적용시키는 것은, 티켓 발행과 인증으로 인한 서버의 부담을 가중시키는 결과를 초래한다. 또한 티켓 승인 서버(TGS)라는 또 하나의 서버가 필요로 하게 된다.

본 논문에서 제안한 알고리즘은 P2P의 특성상 서버와 클라이언트의 기능을 모두 가질 수 있는 Peer에게 티켓 승인 서버의 기능을 부여함으로써 이를 해결하였다. 하지만 또 하나의 인증을 위한 서버인 티켓 승인 서버를 사용하지 않음으로써 생길 수 있는 인증의 문제는 [단계2]에서 다음과 같은 메시지를 상대측(*Peer B*)으로 보냄으로써 해결할 수 있다.

$$AS \rightarrow Peer\ B(TGS\ B) : E_{K_b}[K_{tgsb} \| ID_b \| TS_2]$$

K_{tgsb}라는 서버와 상대측(*Peer B*)만이 알 수 있는 일회용 키를 사용함으로써 상대측은 티켓 승인 서버의 기능을 할 수 있다. 이러한 메시지의 추가로 인한 또다른 이점이 있다. 티켓 발행의 위하여 K_{tgsb}라는 일회용 키를 사용함으로써 재전송(replay) 공격에 강력하고 견고하다. 이러한 K_{tgsb}는 외부에 노출이 되더라도, 일회용 키로써 재사용이 불가하다.

표 1. 기존 Kerberos와 제안된 알고리즘과의 비교 분석

	Kerberos	제안된 알고리즘	비 고
총 시스템 개수	4 (AS, TGS, C, S)	3 (AS, Peer A, Peer B)	
서버의 총개수	2 (AS, TGS)	1 (AS)	서버의 개수 감소
티켓 발행	중앙처리	분산처리	서버 부하량과 트래픽 분산
티켓 발행시 서버 이용단계	4(AS→2, TGS→2)	2	서버 부하량과 트래픽 분산
Ticket 발행을 위한 키	고정 비밀키	일회성 비밀키	replay 공격에 강력하고, 일회성 비밀키는 노출이 되어도 재사용 불가
Ticket 발행	TGS	Peer	상대측의 Peer에서 Ticket을 발행함으로써 Ticket 발행 부담 분산
Ticket 인증 유효 기간	lifetime (시간)	downloadcount (다운 횟수)	P2P 환경에 맞도록 인증 유효 시간이 아닌 다운 횟수를 인증 유효 설정 값으로 사용

또한, [단계4]에서 티켓의 고유번호인 *TicketNum*와, 실행횟수 제한값인 *downloadcount*를 추가함으로써 P2P 시스템의 특성상 시간적 인증보다는 횟수의 인증과 시간적 인증을 겸하였다.

한번 인증을 받은 Peer는 재접속을 위하여 인증을 받을 필요가 없다. 티켓에 허용된 횟수만큼은 언제든지 [단계5]와 [단계6]의 반복만으로 상호 인증이 가능하다. 또한 제안된 알고리즘은 상호 인증을 위해 인증 서버(메인 서버)가 수행해야할 단계는 [단계1], [단계2]뿐이다. 그러므로 제안된 알고리즘은 P2P 시스템에서 인증을 위한 메인서버의 기능을 최소화할 수 있다.

Ⅴ. 결 론

본 논문에서는 상호 인증을 위한 알고리즘인 Kerberos를 개선하여 P2P 시스템에 맞는 알고리즘을 제안하였다. 제안된 알고리즘은 서버의 부담을 최소화하기 위하여 티켓 승인 서버의 기능을 상대측의 Peer에 부여하였다. 이러한 방법을 이용하여 티켓 승인 서버를 위한 서버의 개수 증가를 막고, 인증을 위한 서버의 기능을 최소화하여 서버의 부담을 최소화하였다. 제안된 알고리즘은 상호 인증을 위하여 서버는 최소한의 역활만하고, Peer간에 인증의 역활을 부여하면서도 강력한 상호 인증을 할 수 있다. 또한 P2P 시스템에 맞도록 시간위주의 인증 만료 시간보다, 횟수 위주의 인증 제한 값을 주었다.

또한 이러한 인증과정에서 생기는 키값을 이용한 암호화 통신은 기밀성을 보장한다. 향후 연구과제로는 P2P 시스템에서 부인 방지 기술 및 컨텐츠 보호 기술의 적용이 되어야 할 것이다.

참고문헌

[1] David Barkai, Peer-to-Peer Architecture Group, Microcomputer Research Lab and Intel Corporation, "An Introduction to Peer-to-Peer Computing", Intel®Devloper-UPDATE Magazine, pp.1-7, February 2000.

[2] I. Foster, C. Kesselman, G. Tsudik, S. Tuecke. "A Security Architecture for Computational Grids." Prpc. 5th ACM Comference on Computer and Communications Security Conference, pp. 83-92, 1998

[3] J. Kohl and C. Neuman, "The Kerberos Network Authentication Service(V5)," RFC 1510, September, 1993.

[4] John T. Kohl. B. Clifford Neuman, Theodore Y. T'so, "The Evolution of the Kerberos Authentication System, pages 78-94. IEEE Computer Society Press, 1994.