2000년 한국정보처리학회 추계 학술논문발표집 제7권 제2호

X.509와 DNS이용한 분산 인증 알고리즘의 설계

김철현*, 정일용*
*{kch7604,iyc}@mina.chosun.ac.kr, 조선대학교 전자계산학과

The Design of Distributed Authentication Algorithm Employing X.509 and DNS

CheolHyun Kim* and IlYong Chung*
*Dept of Computer Engineering, College Engineering, Chosun University #375 Seosuk-dong, Dong-gu, Kwangju, 501-759, Korea

요 약

본 논문에서는 X.509와 DNS를 연관하여 Kerberos를 기반으로 분산 인증 알고리즘을 제안한다. Kerberos에서는 영역간의 서비스에 대하여 언급을 하지 않았기 때문에 영역간 인증은 X.509와 Domain Name System(DNS)를 사용하여 얻을 수 있는 체인에 의해서 수행하는 PKINIT를 통하여 이루어진다. 두 개의 프로토콜은 상이한 키관리 방식을 갖고 있는데 Kerberos는 공통키에 기반을 두고 있는 반면에 X.509는 공개키 방식에 기반을 두고 있으므로 이들을 상호 연동시키기 위해 연결 세션은 Directory Service(DS)를 이용하였고, 실제적인 인증을 위해서는 Kerberos를 적용하였다. 새로운 알고리즘은 통신복잡도의 관점에서 고찰하면 IETF CAT 그룹에서 제안한 알고리즘을 개선하였다.

1. 서론

컴퓨터와 정보통신의 발전에 따라서 다양한 응용서비스가 창출하고 있으며 신뢰성있고 안전하게 서비스들을 제공하기 위해서 해결되어야 할 중요한 문제는 정보보호이다. 정보통신서비스 사용을 위협하는 대표적인 요소는 합법적인 사용자로 가장, 비인가 자원에 대한 접속시도, 서비스 제공의 부인, 자료 수정 등이며 이를 해결하는 정보보안 메카니즘은 다양한 형태로 구현될 수 있다. 본 논문에서는 네트워크 상에서 여러 가지 문제점들을 대처하는 방안으로 인증에 대해 중점적으로 다루고자 한다. 네트워크 상에서 사용권한이 없는 사용자가 실제 사용자인 것처럼 위장을 하여 서버에 접속, 위조한 메시지의 재전송 등을 통한 서비스를 요청할 경우 인증을 통하여 적법한 사용자만이 서비스를 사용할 수 있으며 또한 사용자가 사용하려고 하는 영역, 서버 확인과 같은 상호확인 과정을 통하여 소중한 정보가 제3의 사용자에게 유출되지 않도록 방지할 수 있도록 한다.

분산 환경에서 대표적인 인증 메커니즘으로 개발된 Kerberos[1]와 SESAME[2]가 있다. Kerberos는 MIT의 Athena 계획의 일환으로 개발된 인증 서비스이며 안전한 서비스를 통하여 사용자들을 인증 할 수 있도록 한 인증 메커니즘이며 키 분배 센터(KDC: Key Distribution Center)[3,4]의 개념을 적용하고 있다. IETF CAT Working group에서 KDC 기반 인증 메커니즘을 설계하고 있는데 영역과 영역사이, 인증기관과 지역사이에 서비스를 수행하기 위하여 PKINIT(Public Key Cryptography for Initial Authentication)를 이용하고 있다. PKINIT[5]는 Kerberos를 기반으로 두 영역과 영역사이를 공개키로 상호 서비스해주는 메커니즘[6,9]으로 현재의 IETF CAT Working group에서 사용하고 있는 메커니즘을 살펴보면 Kerberos를 기반으로 하여 공통키를 사용하고 있으며 PKINIT를 기반으로 공개키를 사용으로 하고 있다. X.509[8]는 디렉토리 서비스를 정의하는 X.500[7,10] 서비스 권고안의 일부분으로 자신의 사용자에게 X.500의 디렉토리에 의한 인증의 준비에 대해 골격을 정의하고 있으며 공개키 암호화 기법의 사용과 디지털 서명에 근거를 두고 있다.

2. Kerberos 인증 절차

kerberos은 다양한 요소로 구성된 복잡한 시스템이며 중요한 요소로는 kerberos 서버, 티켓승인서버

(TGS), 티켓, 인증자 등이 있으며, 티켓은 kerberos서버와 티켓승인 서버가 생성하여 티켓승인 서버와 서비스 서버와의 통신에 이용되며, 티켓의 구성정보는 서버의 이름, 클라이언트의 이름, 클라이언트의 인터넷 주소, 타임스탬프, 유효시간과 세션키를 포함한다[1,4]. (그림 1)는 Kerberos 인증 프로토콜에 관한 개괄적인 그림을 표현하고 있다. Kerberos 프로토콜의 특징을 살펴보면 다음과 같다.

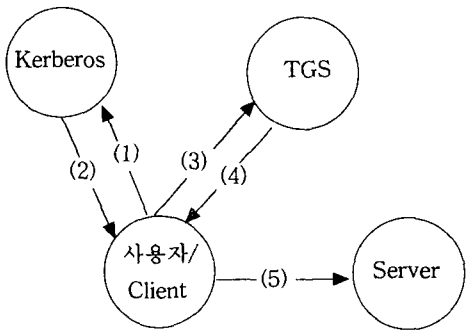

그림 1. Kerberos 인증 절차

① 클라이언트는 사용자의 ID와 TGS 사용에 대한 요구를 의미하는 TGS ID를 함께 AS에 보내는 것으로 사용자의 편에서 티켓-승인 티켓을 요구한다.

② AS는 사용자의 패스워드로부터 알아낸 키를 가지고 암호화된 티켓으로 응답한다. 이 응답이 클라이언트에 도착했을 때, 클라이언트는 사용자에게 패스워드를 입력하라고 요구하고, 키를 생성하고, 들어온 메시지를 복호화한다. 정확한 패스워드가 입력되었으면 티켓은 성공적으로 만들어진다.

③ 클라이언트는 서비스-승인 티켓을 사용자의 편에서 요구한다. 이런 목적으로 클라이언트는 사용자의 ID, 요구하는 서비스의 ID 그리고 티켓-승인 티켓을 포함하고 있는 메시지를 TGS로 전송한다.

④ TGS는 들어온 티켓을 복호화하고 그 ID가 존재하는가에 의해 복호화의 성공 여부를 결정한다. 유효시간이 넘지 않았는지 점검한다. 그런 다음 사용자의 ID와 네트워크 주소를 사용자의 확인을 위해 들어온 정보와 비교한다. 끝으로, 요구한 서비스에 접속을 승인하는 티켓을 발행한다.

⑤ 클라이언트는 사용자의 편에서 서비스에 접속을 요구한다. 이 목적으로 위하여 클라이언트는 서버에게 사용자의 ID 그리고 서비스-승인 티켓이 포함된 메시지를 보낸다. 서버는 메시지의 내용을 이용하여 승인한다.

3. DNS(Domain Name System)

DNS서버는 (그림 2)과 같이 연산 인터페이스와 관리 인터페이스의 두 인터페이스를 가진다. 연산 인터페이스를 사용하여 DNS 클라이언트는 DNS질의를 보내고 DNS 응답을 한다[6]. 관리 인터페이스는 CA가 인증서와 CRL을 등록하기 위해 사용된다. 사용자나 관리자가 DNS서버에 요청을 보내고자 할 때는 적절한 매개 변수를 사용한다[11,12].

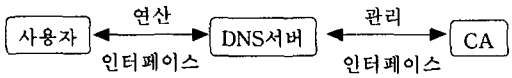

그림 2 . DNS 서버의 인터페이스

먼저 X.509 인증서를 DNS서버에서 가져오는 과정을 설명한다. 두 사용자 A와 B가 U,V,X의 존에 등록되어 있고 사용자 C는 Z,Y,X의 존에 등록되어 있다고 가정하자. 만약 사용자 A가 B의 인증서를 원한다면 질의는 DNS서버 U에 전달되고 U는 CA_U의 개인키로 서명된 B의 인증서를 받을 것이다. 사용자 A는 CA_U로부터 B의 공개키를 받아 인증서를 확인할 수 있다. 만약 사용자 A가 다른 존에 있는 C의 인증서를 요청하면 이 요청은 반복적인 방법[11,12]이나 순환적인 방법으로 처리된다. 그러나 사용자 A는 CA_Z의 공개키를 가지고 있지 않으므로 C의 공개키 인증서를 확인할 수 없다. 따라서 사용자 A는 C의 인증서를 확인할 수 있기 위해서는 Directory Service(DS)를 이용하여 전방 인증서와 후방인증서를 연결하는 인증서 경로가 필요하게 된다. 전방인증서와 후방인증서는 다음과 같이 표현할수 있다

- forward certificate :

A≪CA_U≫CA_U≪CA_V≫CA_V≪CA_X≫CA_X≪CA_Y≫CA_Y≪CA_Z≫CA_Z≪C≫

- reverse certificate :

C≪CA_Z≫CA_Z≪CA_Y≫CA_Y≪CA_X≫CA_X≪CA_V≫CA_V≪CA_U≫CA_U≪A≫

4. 분산 인증 알고리즘의 설계

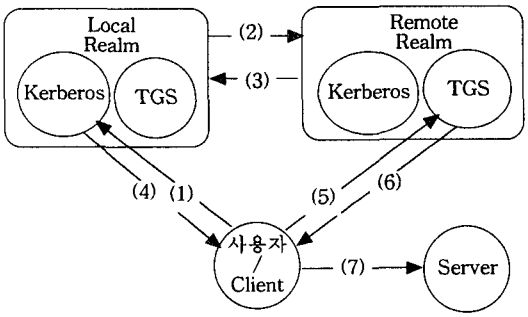

그림 3. 사용자와 Remote 서버간의 절차

4.1 Notation

PK_C : C의 공개키
K_C : C의 개인키
CS : Certificate Serial Number
A_p : 인증서 주체의 공개키(알고리즘, 파라메터, 키)
A_I : 알고리즘 식별자(알고리즘, 파라메터)
ID_C : C에 있는 사용자의 식별자(C의 ID)
AD_C : C에 있는 Address값(C의 IP)
KDC<<C>> : Kerberos로 발행된 C의 인증서

4.2 KerbInit_Authentication Algorithm

① 사용자는 Remote Realm의 서비스를 요청한다.

C-> Kerberos$_{LOC}$: ID_C, R

② Local 는 자신의 정보 (Auth-Pack, SigAuth-Pack), Certificate의 형식(User-type), 사용자의 인증서, CS(CertificateSerialNumber)를 전방체인으로 획득한 Remote Kerberos의 공개키(PK_{KDC}^{REM})으로 암호화하여 전송한다. 사용자가 요청한 서비스가 동일한 영역 내에있는 서비스라면 영역간 인증이던가 외부 영역에 있는 서버들의 공개키를 얻는 과정은 필요가 없게 된다. 만약 동일 영역 내에 존재하지 않으면 Kerberos는 Client가 요청한 영역이 어디에 존재하는지 DNS에게 검색한 후에 이웃(Preauthentication)하는 영역을 전·후방 인증 체인을 생성하여 Remote Realm의 공개키를 획득한다.

$Kerberos_{LOC}$ -> $Kerberos_{REM}$:

{Auth-Pack, {SigAuth-Pack}PK_C,
User-type, KDC_{LOC}<<C>>,CS}PK_{KDC}^{REM}

SigAuth-Pack : {A_I, PK_C}
Auth-Pack : {Kerberos, Realm, cusec, ctime, nonce, A_I, A_p}
User-type : X.509V3 (DER encoding) PGP (PGP specification) PKIX (PKCS #6)

③ Remote Kerberos는 Remote TGS와 Client사이에 사용할 정보를 Local Kerberos의 공개키로 암호화(PK_{KDC}^{LOC})하여 전송한다.

KDC_{REM}-> KDC_{LOC} :

{V, KDC-cert, nonce, kdcPublicValue,
{$K_{C,TGS}$, TGS, TS, nonce}PK_C,
{$K_{C,TGS}$, ID_C, AD_C, TGS, TS, nonce}K_{TGS},
User-Type}PK_{KDC}^{LOC}

kdcPublicValue : {A_i, A_p}
User-type : X.509V3 (DER encoding) PGP (PGP specification) PKIX (PKCS #6)
KDC-cert : Issuer (인증서를 발행하고 서명한 CA) SerialNumber (인증서 일련번호)

④ Local Keberos는 정보를 Client의 개인키로 암호화하여 전송한다. 이 정보는 Local Client와 Remote TGS의 세션키($K_{C,TGS}$)및 Remote TGS Ticket을 포함하고 있다.

KDC_{Loc} -> C :

{Auth-Pack, V, A_i, User-type, TGS, TS, nonce, $K_{C,TGS}$}K_C,
{$K_{C,TGS}$, ID_C, AD_C, TGS, TS, nonce}K_{TGS}

⑤ Client자신이 사용할 서버와 TGS용 티켓, 자신의 인증서를 세션키($K_{C,TGS}$)로 암호화 한 후 서버용 티켓을 요청한다.

C -> TGS_{REM} :

S, {$K_{C,TGS}$, ID_C, AD_C, TGS, TS, nonce}K_{TGS},
{ID_C, AD_C, TS }$K_{C,TGS}$

⑥ Remote TGS는 Client와 서버가 사용할 세션키($K_{C,S}$)등을 포함한 정보를 Client와 TGS용 세션키($K_{C,TGS}$)로 암호화하고, Client와 서버사이에 사용할

서버용 티켓을 서버의 개인키(K_S)로 암호화하여 Client에게 전송한다.

TGS_{REM} -> C : {$K_{C,S}$, S, TS, nonce}$K_{C,TGS}$,
{Kc,s, ID_C, AD_C, TS, S, nonce}K_S

⑦ Client는 서버용 티켓 그리고 자신의 인증서를 서버와 사용될 세션키($K_{C,S}$)로 암호화하여 서버에게 서비스를 요청한다.

C -> S : {Kc,s, ID_C, AD_C, TS, S, nonce}K_S
{ID_C, AD_C, TS}$K_{C,S}$

5. 분산 인증 알고리즘의 비교분석

현재의 IETF CAT Working group에서 PKINIT기반의 공개키와 공통키를 모두 사용하여 Kerberos인증 시스템을 설계하고 있다. 사용자가 Remote Realm 서비스를 받기 위해서는 먼저 자신의 영역에서 인증을 받은 후 원하는 서비스가 외부영역에 있는 경우 Local Realm에 위치한 Kerberos와 Remote Realm 에 있는 Kerberos사이에 PKINIT를 이용하여 정보를 송·수신한다. Local Realm에 위치한 Kerberos에게 전송된 정보 속에는 Local Realm에 존재하는 Client를 인증할 수 있는 난수값이 포함되어 있다. Local Realm에 존재하는 Client는 이 난수 값을 보관해야 하며, 이 키로 Remote Realm에 위치한 Kerberos에게 서비스를 요청할 수 있다. 본 논문에서는 Kerberos을 기반으로 IETF CAT Working group에서 사용하고 있는 PKINIT에 포함된 X.509 프로토콜을 적용하였으며, X.509에 포함된 디렉토리 인증 시스템인 DS를 적용하여 영역간에 연결된 체인을 이용하여 다른 영역을 인증할 수 있는 공개키를 얻어서 서비스를 제공할 수 있도록 한다.

6. 결론

분산환경에서 대표적인 인증 메커니즘인 Kerberos와 PKINIT에 포함된 디렉토리 인증 시스템인 X.509와 DNS 고찰하였다. 본 논문에서는 Kerberos를 기반으로 하여 IETF CAT Working group에서 사용하고 있는 PKINIT에 포함된 X.509와 DNS 적용하여 분산환경에서 서비스를 제공하는 인증 방식을 제안하였다.

Kerberos는 동일 영역에서는 다양한 정보보안 서비스를 제공하지만 외부 영역에서의 서비스에 대한 제안이 없다. 이를 보완하기 위해서 외부영역의 위치탐색을 위한 DNS를 적용하였고, PKINIT에 포함된 X.509를 적용하여 영역간에 연결된 체인을 이용하여 다른 영역을 인증할 수 있는 공개키를 얻어서 서비스를 제공받도록 한다.

두 개의 프로토콜은 상이한 키 관리 방식을 가지고 있는데 Kerberos는 공통키에 기반을 두고 있으며, PKINIT에 포함된 X.509는 공개키 방식에 기반을 두고 있으므로 상호 접목시키기 위해 연결 세션은 X.509 디렉토리 인증 시스템인 DS를 적용하는 공개키 획득할 수 있게 하였다. 이때 Clint는 난수값을 보관하지 않아도 되며 Remote Realm에 위치한 Kerberos와의 재확인 절차과정을 배제하였으며 Local Client와 Remote TGS 세션은 이전 세션 방식을 사용하여 공개키 방식과 비밀키 방식이 상호 충돌하는 문제점이 없도록 설계하여 IETF Working Group 에서 제시한 방법보다도 통신 복잡도를 감소하였다.

참고문헌

[1]Jennifer G. Steiner, Clifford Neuman and Jeffrey I. Schiller, "Kerberos: An Authentication Service for Open Network Systems", In Proc. of the Winter 1988 Usenix Conference. Feb. 1988.

[2]http://www.cosic.esat.kuleuven.ac.be/sesame

[3]http://cd.donga.ac.kr/~shwan/doc/data/kerberos/kerberos.html

[4]http://netsec.ajou.ac.kr/~gonswing/main/research/kerberos.html

[5]http://www.ietf.org/internet-draft-ietf-cat-kerberos-pk-init-09.txt

[6]심희원, 김진성, 심영철, 임찬순, 변옥환, "확장된 DNS보안 메커니즘의 설계 및 구현", 한국정보처리학회지, 제6권, 제1호, pp.134-147, 1999.

[7]http://www.opendiroectory.com/whitepapers/x500tut.html

[8]http://www.opendiroectory.com/whitepapers/x509tut.html

[9]Warwick Ford: Computer Communications Security, New Jersey, Prentice-Hall, 1994

[10]http://sorec.chungnam.ac.kr/~CALS/directory/dirstd.htm

[11] 모영범, 송주석, "반복 인증을 고려한 인증 프로토콜 제안 및 분석", 통신정보보호학회논문지, 제 5권 제 2호, pp.45-60, June 1995.

[12]http://www.ietf.org/internerdraft-ietf-dnsop/keyhand-00.txt