情報保護學會論文誌
第12券 第2號, 2002. 4

네트웍 환경에서 안전한 Kerberos 인증 메커니즘에 관한 연구

신 광 철*, 정 진 욱**

A Study on Secure Kerberos Authentication using Trusted Authority in Network Structure

Kwang-Cheul Shin*, Jin-Wook Chung**

요 약

네트웍 환경에서 Kerberos 인증 메커니즘은 Server가 사용자를 인증하기 위해 다른 영역에 있는 Kerberos Server를 신뢰할 것을 요구하고 있다. 또한 상호운용 영역에 있는 Kerberos Server는 다른 영역의 Server와 비밀키를 반드시 공유하여야 한다. 이러한 두 가지의 문제점을 해결하기 위하여 본 논문에서는 IETF CAT의 공개키 기반구조를 갖는 PKINIT/PKCROSS 알고리즘을 적용하고 디렉토리 시스템(Directory System)을 사용하여 영역과 영역사이의 서비스가 신뢰센터를 통하여 각 Kerberos를 증명 할 수 있도록 하는 신뢰센터 기반의 안전한 보안 메커니즘을 제안하였다. 또한 서로 다른 영역의 Kerberos Server는 각 Server의 비밀키와 공유키를 미리 알고 있어야 하지만 신뢰기관을 통하여 각 영역의 공개키 값 및 공유 키 값을 획득하도록 하여 응용 Server가 KDC(Key Distribution Center)에 키를 등록해야 하는 과정을 배제하였다.

ABSTRACT

In Network Environment, Kerberos certification mechanism to require Kerberos server in other area unconditionally belief. Also, Kerberos server in cooperation area must be share server of other area and secret key. To solve these two problems, this paper proposed safe security mechanism of doing to apply IETF CAT's PKINIT/PKCROSS algorithm with Public Key Infrastructure and use Directory System and service between realms do trust and prove each Kerberos trust center base. Also, Although Kerberos server of each area must be foreknowing each server's secret key and public key, Obtain through Trust center and acquire each area's public key and common symmetric key, Application server excluded process that must register key in Key Distribution Center.

keyword : *Kerberos, PKINIT/PKCROSS, Directory System, X.509*

Ⅰ. 서 론

분산환경에서 네트웍을 통한 안전한 정보서비스를 제공하기 위해 해결되어야 하는 중요한 과제는 상호 인증을 통한 정보보호이다. 특히 불법 사용자가 합법적인 사용자로 가장하여 비인가 자원에 대한 접속 시도, 서비스 제공의 부인, 자료수정 등이 정보서비스 사용을 위협하는 대표적인 요소이다. 이러한 개

* 벽성대학 컴퓨터계열 소프트웨어개발전공(kcshin@mail.byuksung.ac.kr)
** 성균관대학교 전기전자 및 컴퓨터공학부(jwchung@songgang.skku.ac.kr)

방된 환경에서 대표적인 인증 메커니즘으로 Kerberos와 Yaksha 인증방식이 있으며 여러 응용시스템에 호환성을 갖도록 구성된 정보보호 하부구조로써 Kerberos 메커니즘을 확장한 SESAME이 있다. 본 논문에서는 네트웍 상에서 여러 문제점들을 대처할 수 있는 방안 중 Kerberos 인증에 관해 중점적으로 다루었다. Kerberos 메커니즘은 Server에 접근하는 사용자들에게 Server 자신을 인증해 주는 기능을 갖도록 중앙 집중식 인증서버를 제공하는 관용암호방식으로 개발되었다[1]. 이 메커니즘은 동일영역에서 Client와 Server간 인증 알고리즘으로써 최적의 메커니즘을 갖는다. 물론 네트웍 환경에서의 인증을 위해서는 Kerberos Server, 다수의 Client와 Application Server로 구성된 안전한 서비스의 Kerberos 환경이 구성되어야 하며 다음과 같은 조건을 필요로 한다[2]. 첫째, Kerberos Server는 반드시 ID(UID)와 모든 사용자의 해쉬된 패스워드를 데이터베이스에 가지고 있어야 한다. 둘째, Keberos Server는 반드시 각 Server와 비밀키를 공유하여야 한다. 모든 Server는 Kerberos Server에 등록해야 한다. 셋째, 각 상호 운영영역에 있는 Kerberos Server는 비밀키를 다른 영역에 있는 Server와 공유한다. 두 Kerberos Server는 서로 등록되어야 한다. 넷째, 한 영역에 있는 Kerberos Server가 사용자를 인증해 주기 위해 다른 영역에 있는 Kerberos Server를 신뢰할 것을 요구한다. 두 번째 영역에 있는 Server는 반드시 첫 번째 영역에 있는 Server를 신뢰해야만 한다. 이러한 메커니즘은 분산 네트웍 환경에서 다수의 영역(Realms)이 존재할 경우 각 Server에 대한 비밀키의 교환과 저장의 부담(n(n-1)/2)이 가중되어 바람직하지 않다. 이로 인하여 IETF CAT Working Group에서는 네트웍환경에서 공개키 관리와 인증서 기반구조를 갖는 PKINIT(Public Key Cryptography for Initial Authentication) 기반의 새로운 인증서비스를 발표하였다[3]. PKINIT는 DSA키들과 Diffie-Hellman키의 조합을 이용하여 공개키와 비밀키의 암호화를 이용한 초기 인증기준을 제공함으로써 안전한 서비스를 지원한다. PKINIT 메커니즘에 의한 공개키 사용으로 위 ①, ②, ③번째까지의 과제는 해결되었다. ④번째의 신뢰문제를 해결하기 위해서는 제3자(신뢰센터)에 의해 인증된 공개키를 필요로 하나 PKINIT는 신뢰를 전제로 영역간 인증을 제시하고 있다. 본 논문에서는 신뢰센터(TC :Trusted Center)를 두어 Kerberos를 이용한 영역간 안전한 인증과 공유키 분배를 할 수 있는 메커니즘을 제안하였다. Kerberos는 키 분배센터(KDC : Key Distribute Center)와 인증서버(AS : Authentication Server)의 역할을 수행하며 티켓승인서버(TGS : Ticket Granting Server)를 포함하고 있다[4].

II. 인증방식

정보시스템을 통해서 교환되는 정보를 보장하기 위하여 상대확인, 불법수정 여부, 의도된 상대방에게 전달되었는가 하는 확인의 기능이 필요한데 이를 인증(Authentication)이라 하며 공개키 암호방식에서 공개키의 무결성을 보증(Assurance)하는 의미로도 인증(Certification)이라 한다. Authentication은 메시지의 생성, 전송, 수신, 저장 등 일련의 과정에 관련된 객체(Objects)들이 진정한 사용자라는 것을 증명할 수 있도록 하는 사용자 인증과 전송되는 메시지의 내용에 대한 무결성을 보장하는 메시지인증으로 구분된다. 이와 같은 인증방식으로 비밀키 암호에 근거한 안전성이 높은 Kerberos 인증과 Kerberos 방식의 문제점을 해결하기 위해 공개키 암호방식을 시도한 Yaksha 인증방식[5], 그리고 개방형 분산시스템에서 자원보호 기반구조를 목적으로 비밀키와 공개키를 모두 사용하면서 Kerberos 방식를 수용한 유럽방식인 SESAME[6]이 있다.

2.1 Kerberos

Kerberos는 복합시스템으로 Kerberos Server와 티켓승인서버(TGS), 티켓(Ticket), 인증자(Authenticator)로 구성되어 있으며 Kerberos Server와 TGS가 Ticket을 생성하여 TGS와 서비스 Server와의 통신에 사용되며 Ticket의 구성정보는 Server와 Client 이름, 타임스탬프(TimeStamp), 유효시간(Lifetime), 세션키(Session Key)를 포함한다. 인증자는 Client에 의해 생성되고 생성된 인증자는 한번만 사용할 수 있으며 인증정보는 Client의 이름과 웍스테이션의 IP 주소, 현재의 시간을 포함하고 있다.

Client가 Server접근 Ticket을 요청하기 위해 로그인하고 Server에 접속하기 위한 요구정보를 전송하여 AS에 의해 인증이 되면 Ticket를 생성하여 Client로 보냄으로써 Server로부터 Client가 허가

를 받았다는 사실을 확인시킨다. 사용자는 Ticket을 보관하여 서비스에 접속할 때마다 이 Ticket을 이용하여 TGS에게 접속한다. AS는 자신의 DB에 저장되어 있는 Client의 패스워드로 Client의 암호키(K_C)를 생성하여 Ticket을 발급한다. 패스워드의 입력 시기는 Ticket이 도착한 후에 자신의 패스워드를 입력하여 키를 생성하고 Ticket을 복호화한다. 또한 Ticket의 가로채기를 봉쇄하기 위해 Ticket이 발행된 시간과 유효시간을 포함하고 있다. AS가 Client와 TGS간, Client와 Server간에 세션키($K_{C,TGS}$, $K_{C,V}$)를 제공하여 신원을 확인시켜 주고 유효시간을 짧게 Client의 인증자($Authenticator_c$)에게 두어 가로채기의 위험을 방지하고 있다.

2.2 Yaksha

Yaksha는 인증기관(CA : Certification Authority)과 Yaksha 인증서버, Client, TGS 서버, 자원서버들로 구성되어 있으며 Kerberos와 유사한 구조를 가진다. 암호키는 비밀키와 공개키를 사용하기 때문에 디지털 서명과 키의 공유, 암호화 통신을 제공한다. Kerberos 기반 시스템으로 공개키 암호를 사용함으로써 다음과 같은 Kerberos의 약점을 보완하는 프로토콜이다.

① Authentication Server의 Client에 대한 비밀키 보관으로 노출은 불리한 결과를 초래한다.
② Kerberos는 패스워드에 의한 dictionary 공격을 당하기 쉽다.
③ Kerberos는 Non-Repudiation Services를 제공하지 않는다.

이와 같은 약점을 없애기 위해 User와 Yaksha Server(AS in Kerberos)간에 RSA 비밀키 d를 사용자의 패스워드인 d_u와 Yaksha Server의 패스워드인 d_s로 나누어 공유한다. 이러한 방법으로 인해 Yaksha System은 Key escrow scheme으로 사용될 수 있으며 공개키 암호를 뒷받침하기 위해 각각의 message는 확장이 가능하다. Yaksha는 그러한 message를 사용하는 Kerberos와 같은 단계를 사용하게 된다. 공개키 암호의 사용으로 Yaksha system이 Kerberos의 약점을 보완하였지만 접근제어(access control)와 위임(delegation)은 아직 고려되지 않은 단계이다. 특히 Yaksha는 영역간 보호(multi-domain security)를 제공하지 못하고 있으며, 단일영역(single domain)에 한정되어 있는 프로토콜이다.

2.3 SESAME

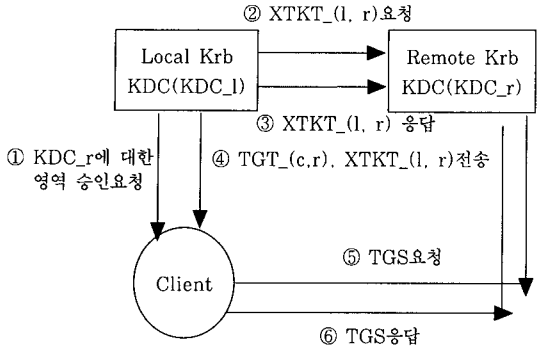
SESAME(Secure European System for Application in a Multi-vendor Environment)은 CEC(Commission of the European Communities)에서 시작된 Multi-vendor 환경에서 Application들에 대한 유럽 보안시스템 연구 프로젝트로써 주목적은 분산 개방형 시스템에서 전사적인 자원보호를 위해 비밀키와 공개키 암호를 모두 사용하여 개발되었다. 주요역할은 네트웍 환경에서 single sign-on과 개인 신원정보인 PAC(Privilege Attribute Certificate)를 사용하여 분산 접근제어를 지원하고 있다. 또한 서로 다른 보호정책으로 다중 도메인을 지원하고 공개키 암호의 사용으로 대규모 네트웍 환경에서 동작되도록 설계되었으며 다른 시스템과 연동을 고려하여 GSS-API(Generic Security Service Application Programming Interface)을 폭넓게 수용하고 있다.

Ⅲ. PKINIT/PKCROSS 프로토콜

RFC2026의 Internet Draft는 공개키 암호를 이용하여 상호영역간 인증을 할 수 있는 Kerberos 프로토콜을 정의하고 있다[7].

Kerberos는 TLS(Transport Layer Security)와 함께 대칭, 비대칭 암호를 모두 사용하며 초기인증교환과정에서 공개키 암호의 사용을 정의하는 PKINIT와 응용 서비스가 공개키 암호를 이용하여 인증한 후 Kerberos Ticket을 어떻게 발급할 것인지를 PKTAPP(Public Key Utilizing Tickets for Application Servers)에서 기술하고 있다. 또한 상호영역에 대한 키(Key)들을 유지하는데 관리적 부담을 간소화하기 위해 PKI[X.509](Public Key Infrastructure)를 이용하도록 기술한 PKCROSS(Public Key Cryptography for Cross-Realm Authentication)정의 명세서가 있다[8~10].

Kerberos에서 Local Client가 Remote Server와 상호영역 인증을 위하여 공개키 암호를 사용하여 Ticket을 요청하고 티켓승인티켓(TGT : Ticket Granting Ticket)을 발급받는 과정은 [그림 1]과 같다.



〔그림 1〕 PKCROSS/PKINIT 메커니즘

① 영역 서비스를 위한 승인요청 : Client는 Remote Realm(KDC_r)에 대한 Ticket을 KDC_l(Local KDC)에 요청한다.

② 영역 사용을 위한 티켓-승인 티켓 요청(PA-PK-AS-REQ) : Local Realm의 KDC_l은 적법한 PKCROSS Ticket인 (XTKT_l, r)을 요구하기 위하여 KDC_r(Remote KDC)에 PKINIT를 요구한다.
즉, KDC_l은 Cross-Realm의 인증을 위해 유효한 PKCROSS Ticket을 가지고 있는지를 캐쉬(Cache)에서 XTKT_(l, r)를 확인하여 합법적인 XTKT_(l ,r)를 가지고 있지 않을 경우에는 PKINIT를 이용하여 Cross-Realm Key를 설치하고 XTKT_(l, r)를 KDC_r에 요청한다. 여기에서 XTKT_(l, r)는 Remote KDC에서 Local KDC로 발행하는 Ticket이다.

③ Remote TGS용 Ticket 승인(PA-PK-AS-REP) : KDC_r은 Ticket에 Ticket Extension(Kerberos 명세서 정보들)을 기록하여 PKINIT/PKCROSS로 응답한다. Ticket Extension은 Client에 대하여 KDC_l에 의해서 발생된 Cross-Realm Ticket들의 Lifetime과 같은 정책들을 포함하고 있다. KDC_l은 Client에 신원증명서 정보를 반영해야 하며, 이 Ticket은 KDC_r이 KDC_l을 신뢰하도록 입증하기 위하여 XTKT_(l, r)을 호출한다.

④ 승인서 발급(PA-PK-AS-SIGN) : KDC_r으로부터 PKINIT/PKCROSS에 의해 수취된 자료를 KDC_l은 Client에 전송한다. KDC_l은 Client에 Ticket(TGT_(c, r) : Local KDC가 Client에 발급하는 티켓승인티켓)을 보낸다. 이 Ticket은 Ticket(XTKT_(l, r)을 Ticket Extension 기록란에 포함되며, XTKT_(l, r)은 Cross Realm Key를 포함하고 있다.

⑤ Ticket 승인서버 요청(PA-PK-KEY-REQ) : Client는 KDC_r에게 Remote Sever를 접근하기 위한 티켓승인서버(TGS : Ticket Granting Server)를 요청한다.

⑥ 티켓승인티켓 발급(PA-PK-KEY-REP) : KDC_r은 해독을 위해 Ticket Extension으로부터 XTKT_(l, r)을 추출하여 비교한 후 서버승인티켓(SGT : Server Granting Ticket)으로 응답한다. 서버승인티켓은 Client가 일정한 유효시간 동안 TGS로부터 새로운 Ticket을 요구하지 않고 재 사용할 수 있다.

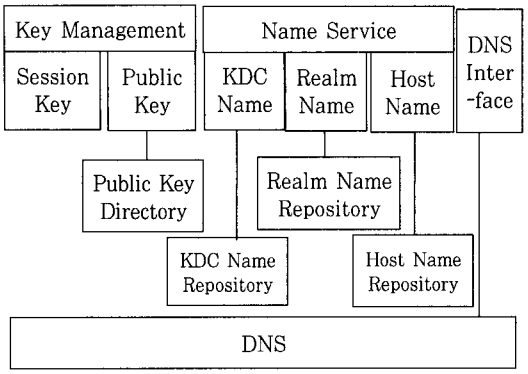
Ⅳ. 신뢰센터를 이용한 Kerberos 인증 프로토콜 설계

본 논문에서는 Kerberos Server가 다른 영역에 있는 Kerberos Server를 신뢰하여야 하는 가정을 해결하기 위하여 신뢰센터(Trusted Center)를 두어 그 역할을 담당하도록 하며 안전한 절차를 수행하도록 하는 새로운 Kerberos 인증 메커니즘을 설계한다. 본 논문의 보안 메커니즘에 사용되는 신뢰센터는 각 국가의 최상위 단일 인증기관으로 상호인증과 인증관리 체계를 운영하며 인증서를 발행하고 인증에 대한 심사 및 평가를 관장한다. 이 외에 신뢰센터의 역할은 다음과 같다.

① 신뢰센터는 외부로부터의 공격에 안전하다.
② 신뢰센터는 KDC 대한 인증 및 안전한 통신에 사용할 공유키를 생성 분배한다.
③ 불법적인 행위의 KDC에게 법적인 책임을 부가한다.

4.1 신뢰 센터의 구조와 서비스

제안되는 메커니즘의 신뢰센터구조는 키 관리 응용(Key Management Application)과 이름 서비스 응용(Naming Service Application)으로 구분되며 공개키 등록과 인증서 발행, 일정기간 비밀통신을 위한 세션 키 생성과 분배를 키 관리 응용이 담당한다. 이름 서비스 응용에서는 자신의 서명정보와 이동경로를 테이블로 보유하는 KDC 이름저장소와 각 KDC의 영역을 IP주소로 유지하는 Realm 저장소,



〔그림 2〕 신뢰센터의 구조

자원 이름저장소를 운영한다. PKINIT/PKCROSS 알고리즘을 적용하며 영역간 체인을 위해 기존의 DNS/DS(Domain Name System/Directory Server)를 사용한다. 이 신뢰 센터의 구조는 〔그림 2〕와 같다.

4.1.1 공개키 등록과 인증서 발행

공개키 암호시스템에서 사용자의 공개키는 공개된 디렉토리에 위치하여 누구나 접근하여 사용할 수 있어야 한다. 그러나 누구나 공개 디렉토리에 접근하여 공개키를 사용할 수 있는 특성으로 공개키가 쉽게 위, 변조되는 문제가 발생한다. 따라서 공개키의 무결성을 보장하기 위해 공개키와 공개키 소유자를 증명하기 위한 인증서가 필요하다. 신뢰센터는 각 영역의 Kerberos의 공개키(KDC_lpk, KDC_rpk)와 공개키 소유자(KDC_l, KDC_r)를 명시한 인증서(Cert)를 공개키 디렉토리에 공고〔그림 3〕하며 이 인증서는 다른 사용자들에 의해 변조가 불가능하도록 신뢰센터의 개인키(TSK)로 전자 서명(Sig_{TSK})되어 저장되고 이를 확인하기 위해 신뢰센터가 발행하는 고유번호(issuerAnd -Serial)를 각 영역 Kerberos의 공개키로 전송하며 키 관리 응용과 이름 서비스 응용에 등록한다. Kerberos가 Remote Kerberos의 서비스를 이용할 때 신뢰센터로부터 KDC에 발행하는 X.509인증서(TC≪KDC≫)가 필요하다. 신뢰센터는 인증서를 Kerberos의 확인용으로 사용한다.

Kerberos ID	인증서	서명 정보
KDC_l	$Cert_l$	Sig_{TSK}(KDC_l, KDC_lpk)
KDC_r	$Cert_r$	Sig_{TSK}(KDC_r, KDC_rpk)
:	:	:

〔그림 3〕 공개키 디렉토리

4.1.2 Name Service

이름 서비스(Name Service)는 각 Kerberos의 KDC의 이름과 영역정보, Host Name을 관리하며 KDC 이름 저장소는 〔그림 4〕와 같은 각 KDC의 식별자 및 이동 영역정보와 자신의 서명정보를 갖는 이동 경로 테이블을 갖는다. 이름 저장소의 구조를 살펴보면 다음과 같다.

① Kerberos ID 필드 : 생성될 때 신뢰센터로부터 부여되는 KDC의 식별자이다.
② Moving Realm : 이동하는 KDC의 영역 도메인 값을 갖는다.
③ 서명정보 : 신뢰센터로부터 인증을 위한 KDC의 비밀키(KDC_lsk, KDC_rsk)로 서명된 Kerberos의 신원정보로 Kerberos Name, X.509, Client 및 KDC가 신뢰할 수 있는 신뢰센터의 유일한 번호, 신뢰센터가 발행한 KDC의 인증서를 갖는다.
④ Flag(이동 중 플래그)

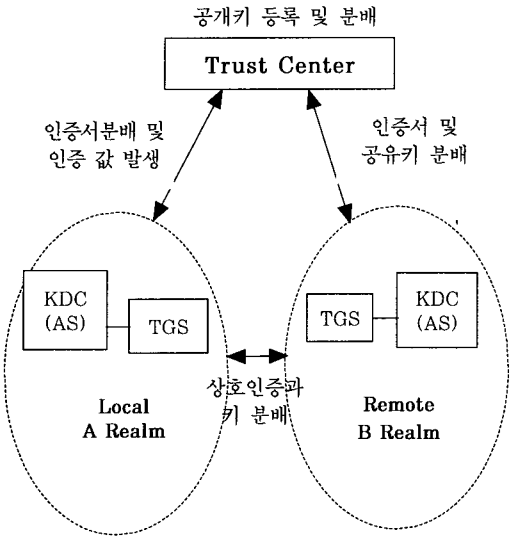
Kerberos ID	Moving Realm	서명 정보	Flag
KDC_l	KDC_r	$SigKDC_{lsk}$ (TrustedCertifiers)	1/0
KDC_r	KDC_l	$SigKDC_{rsk}$ (TrustedCertifiers)	1/0
:	:	:	:

〔그림 4〕 KDC Name 저장소 구조

이동 중일 때 Moving Realm과 서명정보의 필드를 갱신하게 되는 데 이때 플래그를 '1'로 하여 이동 중임을 체크하고 이동이 완료되면 플래그를 '0'으로 한다. KDC Name 테이블에 Moving Realm과 서명 정보 필드가 모두 채워진 경우는 실행이 완료된 상태를 의미한다. 그렇지 않은 경우는 전 KDC가 전송을 요청한 경우로 이동 중임을 의미한다. 이때 이름 저장소의 이동 중 플래그 필드는 '1'로 세팅된다. 이때 만약 KDC가 반복 전송을 시도하거나 재전송 공격을 시도하는 경우에 이를 검출할 수 있게 된다. 자료를 전송 받은 KDC가 서명 정보를 신뢰 센터로 보내 올 때 정상적으로 전송되었음을 확인하고 플래그 필드를 '0'으로 세팅한다.

4.2 전송 프로토콜

신뢰 센터의 구조를 기반으로 한 전송 프로토콜의



〔그림 5〕 전송 프로토콜 동작

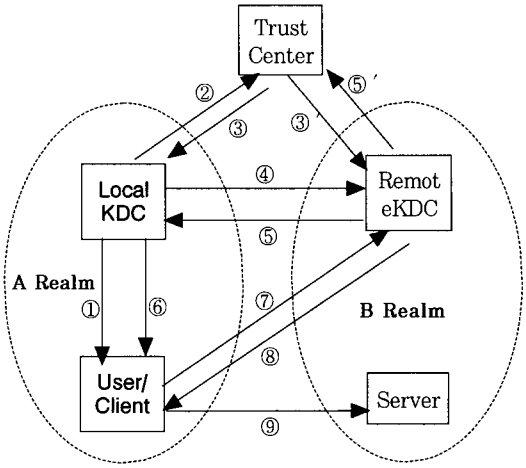
동작은 〔그림 5〕와 같다. 최초 신뢰센터는 각 Realm의 공개키를 등록하고 자신의 비밀키로 서명한 인증서를 공개한다. Remote Realm을 사용하려는 Local KDC가 신뢰 센터에 전송요청을 하면 신뢰센터는 수신 호스트의 서명된 공개키를 분배한다. 또한 수신 호스트에 인증 값으로 임의의 난수(K_{Rand})와 공유키를 분배하여 송신 호스트와의 인증과정을 거쳐 티켓승인티켓과 서버승인티켓을 발급하여 서비스를 하는 세션(Session)을 설정하도록 한다.

위의 동작을 바탕으로 한 전송 프로토콜의 세부명세는 〔그림 6〕과 같으며 전송 요청에서 수신 확인까지의 세부단계는 9단계로 구성된다. 프로토콜에 대한 설명은 전송 요청, 호스트 상호인증 및 키 분배, 전송, 수신 확인의 단계로 기술한다.

〔그림 6〕에서 각 Realm의 Kerberos를 세부적으로 도시하면 〔그림 7〕과 같이 디렉토리 시스템(Directory System)과 연관되어 있다. 모든 Kerberos의 공개키는 디렉토리 시스템에서 획득하여 무결성과 데이터의 인증을 보장받는다.

이 공개키 인증서는 PKINIT에 의한 초기 인증을 목적으로 Remote KDC의 공개키를 획득하기 위해 디렉토리 시스템을 이용한다.

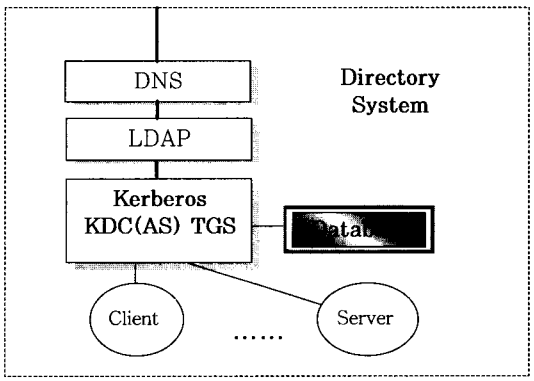
디렉토리 서비스는 데이터베이스, 파일, 호스트 연결, 사용자 서비스 등 모든 자원에 대한 관리를 허용하고 위치 서비스로써 인터넷 DNS을 사용하여 여러 도메인을 트리구조로 연결시킨다. Local Kerberos는 Local Client가 요청한 영역이 동일영역이 아닐



〔그림 6〕 영역간 Kerberos 인증 프로토콜

[Kerberos 인증 프로토콜]

① Remote TGT Ticket요청
② Local KDC 인증
③③′ 난수 및 공유키 분배
④ 인증확인 및 TGT 요청
⑤ Remote TGT 발급
⑤′ 경로테이블 수정정보
⑥ 인증정보확인 및 Remote TGT 전송
⑦ Remote SGT 티켓 요청
⑧ Remote SGT 티켓 발급
⑨ 서비스 요구



〔그림 7〕 디렉토리 시스템의 구조

경우에는 DNS를 사용하여 외부 영역의 경로를 찾는다. 디렉토리 Server는 Client들에게 인증서를 획득하는데 쉽게 접근할 수 있는 경로만을 제공한다. 구성은 도메인 이름과 위치 서비스, 확장성, 표준을 제공하며 관리구조가 용이한 DNS와 디렉토리 서비

```
songgang.skku.ac.kr = {
                    database-name = /usr/local/var/krb5kdc/principal
                    admin_keytab = /usr/local/var/krb5kdc/kadm5.keytab
                                 |
                    key_stash_file= /usr/local/var/krb5kdc/.k5.songgang.skku.ac.kr
                    kadmin_port = 749
                    max_life = 10h 0m 0s
                    max_renewable_life = 7d 0h 0m 0s
                    master_key_type = des-cbc-crc
                    supported_enctypes = des-cbc-crc:normal
                   }
```

〔그림 8〕 Domain 영역구조

스에 접근하기 위한 인터넷 표준(RFC1777) 프로토콜인 LDAP, 인증과 키 교환을 실현할 Kerberos, 객체(Object)들의 식별자와 키, Ticket의 유효시간, 버전, 도메인 정보 〔그림 8〕등을 속성으로 보유하는 Kerberos Database로 되어있다.

Database는 하나의 Master와 여러 개의 Slave로 구성되며 Slave KDC는 데이터베이스의 복사본들을 유지하며 데이터베이스의 추가나 변경 삭제 등은 Master KDC에서만 가능하다. 즉 Salve KDC는 Ticket만을 발급해 주는 역할만 하고 경로를 설정하는 것은 Master KDC가 역할을 담당한다.

〔그림 8〕은 songgang.skku.ac.kr의 도메인으로 Database 이름, 상위 도메인에 대한 목록, 체인(Chain) 경로, 포트번호, 파일의 open 시간, Chain의 연결시간, 마스터 키, 암호방식등의 정보를 보유하며 Ticket 사용시간 내에 재 접속시 저장된 경로로 직접 접근하여 서비스를 받는다.

4.2.1 인증 서비스 교환

① ID_C, $Realms$

- ID_C : Client에 있는 사용자의 식별자(C의 ID)로 자신의 신원을 KDC에 통보
- $Realms$: Client는 Server S의 영역에 대한 Access를 요구

② E_{TPK}〔KDC_l, KDC_r, E_{KDC_lsk}〔TrustedCertifiers〕〕

- TPK : 신뢰센터의 공개키
- KDC_l : Local Kerberos KDC의 신원을 신뢰센터에 통보
- KDC_r : 서비스 받을 Remote Kerberos KDC의 이름을 통보
- KDC_lsk : Local KDC의 비밀키
- TrustedCertifiers : (principalName, caName, issuerAndSerial, $Cert_A$)의 Kerberos 정보
  - principalName = Kerberos Name
  - caName = X.500, X.509 Name
  - issuerAndSerial = Client 및 KDCs가 신뢰할 수 있는 신뢰센터의 유일한 번호
  - Cert = TC가 발행한 KDC의 인증서

③ E_{KDC_lpk}〔K_{Rand}, TC≪KDC_l≫〕, E_{KDC_rpk}〔K_{Rand}, TC≪KDC_l≫〕

- KDC_lpk : Local KDC의 공개키
- K_{Rand} : 인증용으로 생성하는 임의 난수 값으로 신뢰센터에서 발행된 것임을 확인
- TC≪KDC_l≫ : 신뢰센터가 Local KDC에 발행하는 X.509인증서
- KDC_rpk : Remote KDC의 공개키

③′ 〔E_{TSK}〔E_{KDC_rpk}〔K_{Rand}, $_{KDC\_l,r}$, KDC_l, TC≪KDC_l≫〕〕

- TSK : 신뢰센터의 비밀키로 서명
- KDC_l, r : Local KDC와 Remote KDC간 공유키

Client는 자신의 ID와 서비스를 원하는 영역(Realms)을 자신의 영역에 있는 Local KDC(①)에 보낸다. Local KDC는 서비스를 요청한 Client가 정당한 사용자인지를 Database에서 검색하여 유효한 사용자인지 적법성을 검사한다. 요청한 영역이 외부 영역일 경우 Local KDC(②)는 신뢰센터로부터의 인증 확인을 위해 Kerberos 자신의 정보 Trusted-Certifiers를 비밀키로 암호화하고 다시 자신과 Remote KDC의 영역을 신뢰센터의 공개키로 암호화하여 전송함으로써 Kerberos 정보가 Local KDC

로부터 전송되었음을 확인시킨다. 신뢰센터는 Remote KDC_r에서 신뢰센터가 인증하였다는 사실을 확인시키기 위한 난수값으로 K_{Rand}와 신뢰센터만이 생성할 수 있는 Local KDC와 Remote KDC간의 공유키(KDC_l, r), 인증서 TC≪KDC_l≫를 신뢰센터의 비밀키로 서명하여 Local KDC(③)와 동시에 Remote KDC(③')에 전송함으로써 이 값들은 KDC_l로부터 전송되는 정보(④)와 비교하여 신뢰센터가 인증하였다는 사실을 증명하는 값으로 사용된다.

4.2.2 티켓승인 서비스

④ E_{KDC_rpk}[AuthPack, KdcCert, K_{Rand}, TC≪KDC_l≫, E_{KDC_rpk}[K_{Rand}, TC≪KDC_l≫]]

- AuthPack : Local KDC의 영역정보로 {KDC_l, Realm, cusec, ctime, Nonce} 값을 갖는다.
  - KDC_l = Local Kerberos KDC의 신원
  - Realm = Local kerberos 영역
  - cusec = Client의 Timestamp(for replay prevention as in RFC1510)
  - ctime = Kerberos Time(for replay prevention as in RFC1510)
  - Nonce = 데이터의 무결성을 위한 임의의 수
- KdcCert : Client가 이전에 가지고 있는 특별한 KDC의 증명서로 IssuerSerialNumber (인증서를 발행하고 서명한 CA 일련번호)

⑤ E_{KDC_lpk}[KDC_l, r, $E_{KDC\_l,r}$[$Ticket_{TGS}$, TS, Nonce, $Realm_{TGSREM}$], $E_{KTGSREM}$[$Ticket_{TGS}$, TS, Nonce, $Realm_{TGSREM}$]]

- KDC_lpk : Local KDC의 공개키
- KDC_l, r : Local KDC와 Remote KDC의 공유키
- $Ticket_{TGS}$: $E_{KTGSREM}$[flags, $K_{DC\_l,r}$, ID_C, AD_C, TS, Nonce]
  - KTGSREM = Remote TGS의 비밀키
  - flags = 데이터의 이동경로
  - ID_C = Cilent의 ID
  - AD_C = Cilent의 Address
  - TS = TimeStamp, Ticket 발행시간
  - Nonce = 데이터의 무결성을 위한 임의의 수
- $Realm_{TGSREM}$: Remote TGS의 영역

⑤' E_{TPK}[E_{KDC_rsk}[flags, TrustedCertifiers]]

- KDC_rsk : Remote KDC의 비밀키

⑥ E_{KC}[$Ticket_{TGS}$, KDC_l,r, TS, Nonce, $Realm_{TGSREM}$], $E_{KTGSREMr}$[$Ticket_{TGS}$, TS, Nonce, $Realm_{TGSREM}$]]

- K_C : Client의 비밀키
- TS : Timestamp
- KTGSREM : Remote TGS의 비밀키로 티켓의 변조를 막기 위해 Remote KDC와 TGS만이 알고 있는 키

Local KDC(④)는 Remote KDC에게 신뢰센터로부터 인증되었음을 확인시키고 티켓승인티켓인 TGT(Ticket Granting Ticket)를 요청하기 위해 Kerberos 영역정보(Auth-Pack)와 증명서번호, KDC_l의 확인용으로 K_{Rand}를 Remote KDC의 공개키(KDC_r_{pk})로 암호화하여 전송한다. Remote KDC의 공개키는 PKINIT로 획득한 신뢰센터에 의해 서명된 값이다. 이들 정보는 신뢰센터로부터 인증을 받았다는 신뢰성(TC≪KDC_l≫) 확인용과 TGT를 획득하기 위한 정보들이다. Remote KDC는 KDC_l로부터 수신한 정보와 신뢰센터로부터 수신한 [E_{KDC_rpk} [K_{Rand}, $K_{DC\_l,r}$, TC≪KDC_l≫]를 비교하여 Local KDC를 인증하고 $Ticket_{TGS}$과 Session Key($K_{C,TGTREM}$)를 생성하여 공유키로 암호화하여 전송(⑤)함으로써 Remote KDC는 신뢰센터의 인증결과를 확인시킨다. 동시에 Local KDC로부터 수신된 정보가 정확하다는 의미를 자신의 Kerberos 정보와 이동경로를 전송(⑤')하여 신뢰센터의 이동경로 테이블을 수정시킨다. Local KDC(⑥)는 Client의 패스워드에서 추출한 Client의 개인키(K_C)로 수신정보를 암호화하여 전송함으로써 오로지 사용자인 Client만이 읽을 수 있다.

4.2.3 서버승인 서비스

⑦ $E_{KDC\_l,r}$[ID_S, A_C, $Ticket_{TGS}$, $E_{KTGSREM}$[$Ticket_{TGS}$, TS, Nonce, $Realm_{TGSREM}$]]

- ID_S : Remote Server의 ID
- A_C : Client의 Authenticator(인증자)로 $E_{KC,TGSREM}$ [ID_C, AD_C, $Reaml_{TGSREM}$, TS, Nonce]

⑧ $E_{KDC\_l,r}$[$K_{C,S}$, $Ticket_{SGTREM}$, TS, Nonce, $Realm_{SREM}$, ID_S, E_{KS}[$K_{C,S}$, ID_C, AD_C, TS, ID_S, nonce]]

- $Ticket_{SGTREM}$ = E_{KS}[flags, $K_{C,S}$, $Realm_{SREM}$, ID_C, AD_C, TS, Nonce]

- $K_{C,S}$: Client와 Remote Server 만이 공유하는 세션 키
- $Ticket_{SGTREM}$: Remote Server를 액세스할 수 있는 티켓
- $Realm_{SREM}$: Remote 영역의 서버 S
- KS : Server S의 비밀키
- ID_S : Remote Server의 식별자(ID)
- AD_C : Client에 있는 Address값(C의 IP)

⑨ $E_{KC,S}[Ticket_{SGTREM}, A_C, E_{KS}[K_{C,S}, ID_C, AD_C, TS, ID_S, Nonce]]$

메시지 ⑦에서 Client는 서버승인티켓(SGT)을 획득하기 위해 Remote KDC로부터 유효시간내에 재사용이 가능하도록 허가받은 $Ticket_{TGS}$와 함께 자신의 인증자(A_C)를 포함하여 Remote TGS와 자신만이 알 수 있는 세션키($K_{C,TGS}$)로 암호화하여 전송한다. Remote TGS는 KDC_r에서 전송된 정보와 Client와의 정보를 확인한 후 Server와의 세션키($K_{C,S}$)와 서버승인 티켓($Ticket_{SGTREM}$)을 전송(⑧)한다. Client(⑨)는 서버승인티켓과 자신의 인증자, Remote TGS로부터 받은 정보를 전송함으로써 TGS_{REM}으로부터 인증되었고 Server에 접근할 수 있다는 확인을 하게 된다.

4.3 제안된 메커니즘의 분석 및 효과

Kerberos 프로토콜은 Local 네트웍에서 Client와 Server, KDC(AS, TGS)간에 인증받은 주체만이 사용할 수 있도록 설계된 대칭키(Symmetric Key) 기반의 최적 알고리즘이다. 그러나 각 Kerberos 간 비밀키 관리와 교환의 문제를 해결하기 위해 공개키 암호를 사용하여 상호 인증을 할 수 있도록 RFC2026에서 프로토콜로 정의한 것이 PKINIT 메커니즘이다. 이로써 공개키 기반의 Yaksha 프로토콜과 같이 인증(Authentication)과 무결성(Identity), 데이터보안(Privacy)을 제공하고 있다. 제안된 메커니즘은 통신 수행의 각 절차에서 공개키를 기반으로 암호화되고 있고, 전송 절차에서는 신뢰센터로부터 부여받은 송수신자간의 공통키 기반의 공유키를 사용하여 암호화됨으로써 안전한 통신이 형성된다. 따라서 본 논문에서는 Kerberos의 인증문제 중 한 영역에 있는 Kerberos서버가 사용자를 인증해 주기 위해 다른 영역에 있는 Kerberos Server를 신뢰할 것을 요구하며 두 번째 영역에 있는 Server는 첫 번째 영역에 있는 Server를 신뢰해야만 하는 가정을 해결하는 것에 중점을 두고 최상위 인증기관인 신뢰센터를 두어 상호간 신뢰가 이루어지도록 하였다. 신뢰센터는 각 KDC의 공개키를 등록하고 상호 인증을 위한 확인 값 및 공유키를 생성하여 분배하며 X.509인증서를 발행함으로써 KDC간의 인증과 암호프로토콜의 상호운용이 자연스러울 뿐 아니라 Kerberos내의 KDC(AS)는 다른영역의 Kerberos에 대한 인증과 자신의 영역에 대한 객체들을 인증한다. Client와 Remote Kerberos의 TGS, Client와 Remote영역의 Server간에 공통키 기반의 세션키를 사용함으로써 공개키와 혼용으로 네트워크 서비스에 대한 자신의 사용자들을 올바르게 구별할 수 있다. 본 논문에서 신뢰센터를 운용함으로써 기존의 Kerberos 인증에 비해 KDC의 안전한 공개키 사용(신뢰센터 서명키)과 이름 저장소의 경로테이블 유지로 재 전송 공격을 방지할 수 있으며 Kerberos 간에 별도의 비밀인수를 생성하여 확인하는 절차가 불필요하게 되었다. 신뢰센터의 공유키 생성은 KDC_r로부터 Client와 TGS_{REM} 간에 사용할 세션키의 생성을 대신하여 서비스 유형에 따른 키의 생성 과정이 줄게 되었다.

제안된 프로토콜은 대규모 영역에서 PKINIT에 의한 최초 인증 후 KDC를 경유하여 Ticket이 발급되면 Ticket의 유효시간 범위에서는 Kerberos의 데이터베이스에 경로를 유지함으로써 네트웍 구조와 계층적 구조에 국한되지 않고 직접 경로를 통해 해당 Server로부터 서비스를 받을 수 있다.

본 메커니즘[그림 9]는 신뢰센터의 운용으로 취약하였던 Dictionary 공격이 보완되었으며 키 관리의 이원화로 보다 안전하고 신뢰성을 보장할 수 있다 .

V. 결 론

분산환경에서 대표적인 인증 메커니즘인 Kerberos는 PKCROSS/PKINIT기반으로 영역과 영역간의 서비스를 제공한다. 최초 Kerberos는 공통키를 기반으로 다중영역에 대한 인증을 제공하였으나 분산환경에서 비밀키 관리의 제한으로 영역과 영역사이를 공개키로 상호 서비스해 주는 메커니즘을 사용하고 있다. KDC(AS)에 의해 발행되는 X.509는 인증의 준비에 대해 골격을 정의하고 있으며 공개키

구분	분 석		
	Kerberos	Yaksha	제안 메커니즘(TC 운용)
안전성	■ 다중영역(Multi-Realms)상호인증지원을 위해 서명되지 않은 공개키를 PKINIT로 획득, 사용함으로써 Dictionary 공격에 취약	■ 단일영역(Single-Realm)에서의 공개키 방식으로 Kerberos의 단점을 보완하여 Dictionary 공격에 강함	■ 신뢰센터(TC)에 의해 서명된 공개키를 PKINIT/PKCROSS 메커니즘으로 획득 및 티켓을 발급하고 티켓의 유효시간동안 이동경로 테이블 유지로 Replay공격 방어
효율성	■ 세션을 설정할 때마다 DNS를 검색하여 Chain으로 연결 ■ 서비스 세션마다 KDC_l → KDC_r→KDC_l→Client→ TGS_{REM}의 티켓발급과정을 반복	■ 일시적인 공개키와 비밀키의 사용으로 서명을 이용한 티켓발급과정이 YAS→Client→TGS→Client→Server로 간단	■ Chain 형성 후 Kerberos Database의 경로 값 보유로 디렉토리 시스템(DS)을 통한 직접접근 ■ 유효시간 범위 내에서 티켓발행 절차의 간소화(Client→TGS_{REM})
신뢰성	■ 상호영역간 Kerberos 신뢰를 가정함으로써 신뢰성 결여	■ 다중영역(Multi-Realm)에 대한 CA간 지원메커니즘이 없슴	■ 신뢰센터 발행의 인증서 및 난수 값, 공유키에 의해 상호영역간 인증 을 확인함으로써 신뢰성 보장
키관리	■ KDC(AS)에서 공개키와 비밀키 동시관리 ■ 사용자 세션마다 Remote KDC의 세션키($K_{C,TGS}$)생성 및 보관 ■ 서비스 유형마다 Remote TGS의 세션키($K_{C,S}$)생성 및 보관	■ CA와 YAS에서 공개키와 비밀키를 별도 관리 ■ 키 공유와 키 위탁방식 제공	■ 신뢰센터와 KDC(AS)에서 공개키 및 비밀키 별도 관리 ■ 신뢰센터의 공유키(KDC_l,r)생성으로 Client와 TGS_{REM}간 키생성 불필요 ■ 서비스 유형마다 Remote TGS의 세션키($K_{C,S}$) 생성

[그림 9] 메커니즘 분석

암호화 기법의 사용과 디지털 서명에 근거를 두고 있다.

본 논문에서는 영역간에 Kerberos Server가 사용자를 인증하기 위해 다른 영역에 있는 Kerberos Server를 무조건 신뢰할 것을 요구하지 않고 신뢰센터를 통하여 영역과 영역사이의 서비스를 제공 할 수 있는 메커니즘을 제안하였다. 자신의 영역에 존재하지 않는 Server의 서비스를 요청할 때는 X.509를 적용하고 디렉토리 시스템(DS)을 연계하여 영역간에 체인을 형성하고 경로(Path) 값은 Kerberos Database에 저장되어 인증경로 탐색이 용이하다. 영역과 영역사이의 서비스를 위해서 반드시 신뢰기관을 통하도록 하여 각 Kerberos를 증명할 수 있게 하였다. 또한 각 영역의 Kerberos Server는 각 Server의 비밀키와 공유키를 미리 보유하여야 하나 신뢰센터를 통하여 각 영역의 공개키 값 및 공유키 값을 획득하도록 하여 Authentication Server가 키를 등록해야 하는 과정을 배제하였다. 또한 영역과 영역 사이의 서비스를 요청할 때 티켓의 유효시간을 고려한 플래그 상태를 확인함으로써 반복 전송 공격을 탐지 할 수 있으며 Client가 서비스 유형마다 서버승인티켓(SGT)을 직접 Remote TGS로부터 발급됨으로써 절차가 간소화되었다. 본 논문에서는 서명된 공개키와 공유키를 사용하고 영역과 영역간의 서비스에 발생하는 상호 신뢰에 대한 문제점을 해결하고자 신뢰센터 기반의 안전한 보안 메커니즘을 제안하였다.

참 고 문 헌

[1] B.C. Neuman, Theodore Ts'o. Kerberos, "An Authentication Service for Computer Networks", IEEE Communications, 32(9): 33-38. September. 1994.

[2] 최용락, 소우영, 이재광, 이임영 "통신망 정보보호", 도서출판 그린, p. 357, 1997.

[3] B. Tung, C. Neuman, M. Hur, A. Medvinsky, S. Medvinsky, J. Wray, J. Trostle, "Public Key Cryptography for Initial Authentication in Kerberos", draft-ietf-cat-kerberos-pk-init-09.txt

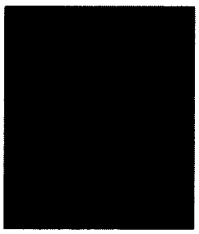
[4] J. Kohl and C. Neuman, "The Kerberos Network Authentication Service (V5)", RFC 1510, September 1993.

[5] R. ganesan, "Yaksha : Augmenting Kerberos with Public Key Cryptography", Proc. of ISOC Symposium on Network and Distributed System Security, pp. 132~143, 1995

[6] Http://www.esat.kuleuven.ac.be/cosic/sesame.html
[7] B. Tung, B.C. Neuman, M. Hur, A. Medvinsky, S. Medvinsky, "Public Key Cryptography for Cross-Realm Authentication in Kerberos", draft-ietf-cat-kerberos-pk-cross-07.txt.
[8] K. Hornstein, J.Altman, "Distributing Kerberos KDC and Realm Information with DNS", draft-ietf-krb-wg-krb-dns-locate-02.txt.
[9] B. Tung, C. Neuman, M. Hur, A. Medvinsky, S. Medvinsky, J. Wray, J. Trostle, "Public Key Cryptography for Initial Authentication in Kerberos", draft-ietf-cat-kerberos-pk-init-13.txt
[10] A. Medvinsky, M. Hur, S. Medvinsky, C. Neuman. "Public Key Utilizing Tickets for Application Servers(PKTAPP)", draft-ietf-cat-kerberos-pk-tapp-04.txt.
[11] ITU-T(formerly CCITT) Information technology - Open Systems Interconnection - The Directory: Authentication Framework Recommendation X.509 ISO/IEC 9594-8.
[12] John T. Kohl and B. clifford Neuman, "The Kerberos Network Authentication Service", Version 5 Revision 5, project Athena, Massachusetts Institute of technology(April 1992).

〈著者紹介〉

신 광 철 (Kwang-cheul Shin) 정회원
1991년~1995년 전쟁연습 프로그램관 및 전산실장(육군대학)
1995년~1999년 성균관 대학원 정보공학과 수료
1996년~현재 벽성대학 소프트웨어개발전공 조교수
〈관심분야〉 정보보호기술, 객체지향 분석/설계, 전자상거래응용, Visual Programming



정 진 욱 (Jin-wook Chung) 종신회원
1973~1985 한국과학기술연구소 실장
1991년 서울대학교 대학원 계산통계학과(이학 박사)
1998년 3월~1999년 2월 성균관대학교 정보통신대학원장
현재 성균관대학교 전기전자 및 컴퓨터공학부 교수
〈관심분야〉 컴퓨터 네트워크, 네트워크 관리, 네트워크 보안