情報保護學會論文誌
第15券 第3號, 2005. 6

Kerberos 인증메커니즘에 관한 연구

김 철 현,[1†] 이 연 식[2‡]
[1]홍성기능대학, [2]군산대학교

A study on Kerberos Authentication mechanism

Cheol-hyun Kim,[1†] Yon-Sik Lee[2‡]
[1]Hongseong Ploytechnic College, [2]Kunsan National University

요 약

본 논문에서는 원거리에서 티켓승인 티켓 교환시 인증절차의 간소화를 가지는 Kerberos시스템 방법을 제안한다. 이를 위해 PKINT에 기반에 영역간의 공개키 획득 과정은 X.509를 사용한다. 상호 신뢰를 강화하기 위해서 전후방 인증서 체인으로 연결하고 신뢰성을 향상시키기 위해서 검증 서버를 적용하여 해당 경로를 검증 하도록 했으며, 티켓 교환단계는 티켓 저장소 및 인증서 검증 모듈, 클라이언트와 서비스 영역, 디렉토리 시스템을 상호 연결을 위한 GSS-API구조를 접목하여 티켓의 이동과정 및 진행사항을 수시로 갱신하여 티켓 저장소에 저장하도록 하였다. 티켓 저장소를 활용하여 키 교환방식을 이용하여 복구가 가능하고 안전한 서비스를 지원하는 인증 모델을 제안하였다. 제안된 Kerberos 시스템을 사용함으로써 원거리 티켓교환 및 검증 과정을 보다 더 안전성이 제공 되며, Kerbreos, 클라이언트, 티켓의 전송 상태를 확인 할 수 있도록 하였다.

ABSTRACT

In this paper, proposes Kerberos certification mechanism that improve certification service of PKINIT base that announce in IETF CAT Working Group. Also proposed Authentication Mechanism for reusability of Ticket that after Ticket's Lifetime is ended, message exchange that Local Client receives Remote Server's service. Since any suggestion to regional services are not described in Kerberos, authentication between regions can be performed via PKINIT(Public Key Cryptography for Initial Authentication) presented by IETF(Internet Engineering Task Force) CAT working group. The new protocol is better than the authentication mechanism proposed by IETF CAT Working group in terms of communication complexity and mechanism according to simplified Ticket issue processing.

Keywords : *Kerberos, PKINIT*

Ⅰ. 서 론

분산 네트워크 환경에서 자원보호는 사용자와 서버 간의 신원증명과 안전한 비밀키 교환을 필요로 한다. 신원증명과 비밀키 교환의 요구를 만족시키기 위하여 인증, 무결성, 데이터 보안기능이 필요하다. 이러한 환경에서 대표적인 인증 메커니즘으로 Kerberos와 Yaksha 인증방식이 있다. 본 논문에서는 네트워크 상에서 여러 문제점들을 해결할 수 있는 방안들 중 IETF의 Working Group에서 활발하게 연구 중인 Kerberos 인증에 관해 중점적으로 연구하였다. Kerberos는 통신망 인증시스템의 개념과 모델로써 중앙 집중식 인증 서버를 제공하는 관용암호방식으로 개

접수일 : 2005년 2월 15일 ; 채택일 : 2005년 5월 17일
† 주저자 : tiger7604@kopo.or.kr
‡ 교신저자 : yslee@kunsan.ac.kr

발되었다[1-4]. 관용암호 인증방식은 네트워크 환경에서 키 관리의 문제로 광범위한 영역의 지원에 제한적이므로 IETF CAT Working Group에서는 영역과 영역사이, 인증기관과 지역을 공개키 암호방식을 사용하여 상호 서비스하는 메커니즘으로 PKINIT(Public Key Cryptography for Initial Authentication) 와 다중 Kerberos 개념의 영역간 인증구조인 PKCROSS (Public Key Cryptography for Cross-Realm Authentication)를 연구하고 있다[5-7]. PKINIT는 Kerberos를 공개키 환경과 연계하기 위한 메커니즘으로 각 개체에 대해 공개키 인증서를 사용하여 AS(Authentication Server)에 인증한 티켓을 받을 수 있도록 한다. 또한 PKCROSS는 원격 티켓을 획득하기 위해 상호영역 인증에 대한 구조를 정의하고 있다. 분산 환경의 인증 메커니즘은 1988년 Kerberos 인증기법이 발표된 이래 1995년에 Kerberos 메커니즘을 그대로 수용하면서 RSA 암호시스템을 도입한 Yaksha와 X.509 공개키 인증기법을 도입한 SESAME이 발표되었다. 이 두 프로토콜 모두 Kerberos 프로토콜의 단점을 보완하였으나, Yaksha는 영역간 인증에 대해 공개키 관리 및 인증기관을 필요로 하는 제약을 가지고 있으며 SESAME은 인증, 접근제어, 데이터 무결성, 기밀성, 부인봉쇄를 지원하는 등 안전성에서 뛰어나지만 다자를 고려한 복잡성을 가지고 있다. 현재 Kerberos를 기반으로 하는 정보보호 응용프로그램은 많은 연구와 함께 제품들이 발표되었다. 네트워크에서 사용되는 여러 응용 프로그램인 rlogin, e-mail의 pop 서버, telnet, NFS 데몬, 데이터베이스 응용, X-Window 응용 등에서 Kerberos 인증기법을 사용한다. Kerberos(KDC: Key Distribution center) 시스템의 제약점으로는 패스워드 기반의 비밀키 운용으로 사전공격(Dictionary Attack)에 대한 안전성에 문제가 있으며 디지털서명이 지원되지 않기 때문에 각 개체들은 AS을 전적으로 신뢰해야 한다는 가정을 두고 있다[8,9]. AS는 도메인에 속해 있는 각 개체의 비밀키를 데이터베이스에 비밀리에 보관하고 있어야 하는 부담이 있으며 도메인이 다른 타 영역으로의 인증서비스를 지원해야 할 때는 키 관리의 문제로 많은 제약을 받는다. 또한 사용자와 응용서버 간에 암호화된 데이터 전송이 이루어진 이후에 데이터를 암호화한 세션 비밀키를 유실했을 때 키 복구를 할 수 있는 대책이 없는 실정이다. 본 논문의 구성은 2장에서 Kerberos 인증메커니즘을 설명하였고 3장에서는 인증서비스 재요청에 따른 알고리즘을 제시하고 마지막에서는 이러한 메커니즘에 대한 분석 및 효과를 살펴보고 마지막 장에서 결론을 맺는다.

II. Kerberos 인증과 디렉토리 시스템

2.1 Kerberos 인증 메커니즘

Kerberos 인증 메커니즘은 여러 가지 요소로 구성된 복합시스템으로 Kerberos서버와 TGS(Ticket Granting Server), 티켓(Ticket), 인증자로 구성되어 있다. Kerberos 서버와 TGS(Ticket Granting Server)가 티켓을 생성하여 TGS와 서비스 서버와의 통신에 사용되며 티켓의 구성정보는 서버와 클라이언트 이름, 타임스탬프(TimeStamp), 유효시간, 세션키를 포함한다. 인증자는 클라이언트에 의해 생성되고 생성된 인증자는 사용을 1회로 제한하고 있으며 인증정보는 클라이언트의 사용자 이름과 네트워크 주소 , 현재의 시간을 포함하고 있다. Kerberos의 Local 영역에서의 인증메커니즘 그림 1 은 다음과 같다.

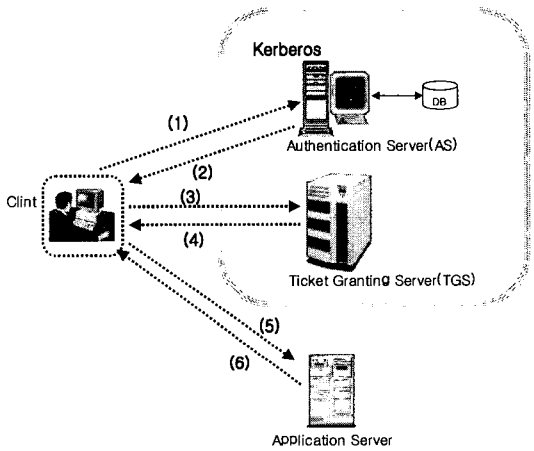

그림 1. Local Realm Kerberos 인증 메커니즘

(1) 클라이언트의 ID와 TGS 상요에 대한 요구를 의미하는 TGS ID를 AS에 보내 TGT (Ticket Granting Ticket)을 요청한다.
(2) AS는 클라이언트의 패스워드로부터 알아낸 키를 가지고 암호화된 티켓으로 응답을 한다.
(3) TGT를 TGS에 전송하여 SGT(Server Granting Ticket)티켓을 요청한다.

(4) TGS는 들어온 티켓을 복호화하고 유효시간을 체크한 후, 클라이언트 정보와 데이터베이스의 정보를 비교 한 후 요청한 서비스를 승인하는 SGT를 발행한다.
(5) TGS로부터 받은 서버용 티켓(Server Granting Ticket)을 전송, 서버는 해당 자료를 비교한 후 권한을 부여 한다.
(6) 서버를 사용할 수 있는 권한을 승인 받는다.

IETF의 Working Group에서 Kerberos인증 메커니즘에서는 티켓을 발급 받기 위해 Remote Kerberos(Remote KDC)가 Local Client를 확인하는 과정을 갖는다. 클라이언트가 서비스 서버에 접속하게 하고, 그 서비스 서버가 클라이언트 입장에서 다른 서비스 서버에 접속할 수 있게 해 준다. Local Kerberos(Local KDC)를 통하여 TGS (Ticket Granting Server)를 접근할 수 있는 티켓과 Remote TGS가 서버에 접근할 수 있는 티켓인 SGT(Server Granting Ticket)을 발급하는 과정 그림 2 의 메커니즘으로 구성되어 있다[4,5].

(1) 인증서비스
클라이언트는 자신의 ID와 원하는 서비스 영역을 Local KDC에 전송하여 서비스(①)를 요청하고, 서비스 영역이 다를 경우 디렉토리 시스템을 통하여 해당영역에 관하여 상호인증을 한다. Local KDC는 클라이언트와 자신의 정보를 전송(②)하여 신원확인 및 티켓 승인을 요청한다. Remote KDC는 정당한 사용자라고 인증한 결과를 Local KDC에게 전송(③)한다.

(2) TGT(Ticket Granting Ticket) 서비스
Local KDC는 Remote KDC로부터 받은 정보를 복호화 한 후 자신의 데이터베이스 및 TGS에 해당 정보를 저장 한다. 그리고 클라이언트에게 전송할 티켓(TGT), 세션키와 티켓, 무결성을 보장하는 정보 등을 클라이언트에게 전송(④)한다.

(3) SGT(Server Granting Ticket) 서비스
클라이언트는 Local KDC로부터 받은 정보를 복호화 한 후 티켓과 세션키, 자신의 정보 등을 세션키로 암호화 하여 Remote TGS에게 티켓과 인증자 서비스를 요청한 서버의 ID를 포함한 메시지(⑤)를 전송한다. 메시지(⑥)는 서버를 사용할 수 있는 티켓과 세션키를 생성하고 원격 서버의 정보를 암호화 한 후 전송 한다. 클라이언트는 서버의 비밀키로 된 내용을 확인 할 수는 없다.

(4) 서비스 요청
메시지(⑦)에서 클라이언트는 서버를 사용하기 위한 요청으로 서버용 티켓과 인증자, Remote TGS로부터 받은 정보를 서버로 전송한다. 해당 서버는 클라이언트로부터 받은 정보를 복호화 하여 정당한 사용자 이면 서비스를 받을 수 있는 티켓을 전송한다.

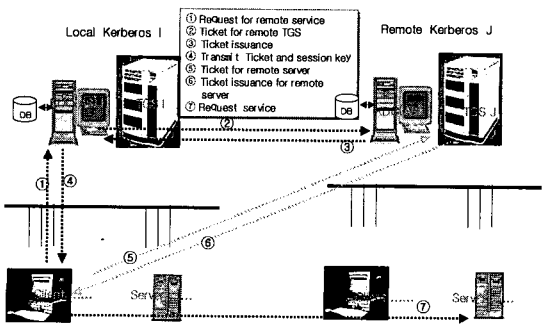

그림 2. 영역간 Kerberos(KDC) 인증절차

2.2 디렉토리 시스템(Directory System)

Kerberos의 공개키는 디렉토리 시스템에 의해서 얻는다. 사용자의 인증서를 보관하고 수신자가 송신자의 신원확인을 하기 위하여 송신자의 공개키 또는 인증서를 확인 후 가져갈 수 있게 보관하는 시스템이다. 저장되는 KDC의 공개키는 디렉토리 시스템에 의해 데이터 무결성과 데이터의 인증을 보장받는다. 이 공개키 인증서는 PKCROSS/PKINIT[5-6]에 의한 초기 인증을 목적으로 Remote KDC의 공개키를 획득하기 위해 디렉토리 시스템을 이용한다. 디렉토리 서비스는 데이터베이스, 파일, 호스트 연결,

표 1. 파라미터

$EK_{C,TGS}$: Client와 TGS사이의 세션키
EK_{TGS} : AS와 TGS의 세션키
EK_{C} : Client와 AS가 공유하는 비밀키
$EK_{TGS,S}$: TGS와 Server가 공유하는 비밀키
$EK_{C,S}$: Client와 Server가 공유하는 비밀키
$EKDC_{\_PK}$: Remote KDC의 공개키
$RealmTGS_{REM}$: 원하는 서비스 영역
ID_{C} : Client 자신의 ID
AD_{C} : Client의 네트워크 주소
SignedAuthPack : 지역영역의 신원인증에 필요한 정보
TrustedCertifiers : Local KDC의 인증서

표2. 디렉토리 시스템의 형식

Domain : host.subdomain.domain X500 : C=US/O=OSF Other : NameType Reserved : reserved DS : directory server (디렉토리 서버) VS : validation server (검증 서버) - 인증서를 수집하고 구축, 검증, 저장 OSCP : online certificate status protocols SCVP : simple certificate validation server

사용자 서비스 등 모든 자원에 대한 관리를 허용하고 위치 서비스로써 인터넷 DNS[7,11]을 사용하여 여러 도메인을 양방향트리구조로 연결시킨다. Local KDC 지역 클라이언트가 요청한 영역이 동일영역이 아닐 경우에는 DNS를 사용하여 외부 영역의 경로를 찾는다. 디렉토리 서버는 클라이언트들에게 인증서를 획득하는데 쉽게 접근할 수 있는 경로만을 제공하며, 인증과 키 교환을 위한 디렉토리 시스템의 구조는 그림 3와 같다. 디렉토리 서비스는 TCP/IP 네트워크 주소변환 기능인 DNS와 핵심 프로토콜로 X.500, LDAP을 사용하여 서로 다른 사이에서 작업할 수 있도록 지원한다. 여기에서 X.500의 디렉토리 시스템의 형식 표 2은 Domain, X500, Other 그리고 Reserved로 구성된다[12,13].

① 클라이언트가 요청한 영역이 동일 영역이 아닐 경우 Local KDC는 DNS 서버를 사용하여 외부 영역을 찾는다.

② DNS는 Remote Realm에 관한 정보를 찾는다.

③ DNS로부터 받은 Remote Realm에 관한 정보를 전·후방 인증 체인을 생성하여 상호인증(Cross Certification) 하기 위한 세션을 연결한다.

④ 디렉토리 시스템에 의해 생성된 인증경로를 검증하고 자신의 서버에 저장한다.

```
AHTENA.MIT.EDU = {
 database-name = /usr/local/var/krb5kdc/principal
 admin_keytab = /usr/local/var/krb5kdc/kadm5.keytab
 |
 key_stash_file= /usr/local/var/krb5kdc/.k5.ahtena.mit.edu
 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 4. Kerberos의 도메인영역

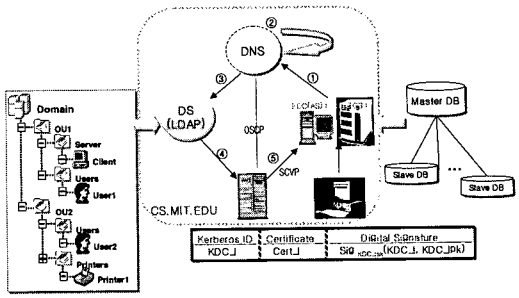

그림 3. 디렉토리 시스템 구조

⑤ KDC는 검증서버(VS: Validation server)로 받은 인증서를 검증하고 전·후방 인증서를 통하여 공개키를 획득한다.

문제는 침해자가 서비스를 요청한 클라이언트처럼 가장하여 서비스를 가로채거나 변경시킬 수 있기 때문에 상호영역간에 있어서 클라이언트를 인증하는 절차를 필요로 하게 된다. KDC는 하나의 Master와 여러 개의 Slave로 구성되며 각각 Kerberos 데이터베이스를 보유한다. 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 KDC의 역할 그림 5 이다. 클라이언트가 요청한 서비스가 동일한 영역 내에 있는 서비스이면 KDC의 데이터베이스에서 클라이언트의 정보로 인증을 하게 되고 요청한 서비스가 동일 영역 내에 존재하지 않으면 KDC는 클라이언트가 요청한 영역이 어디에 존재하는지 디렉토리 시스템을 통하여 DNS에게 검색을 의뢰한다.

DNS 서버는 정방향 조회 그림 6 영역, 캐쉬 루트서버를 이용하여 리졸빙 후 캐쉬영역에 저장 한 후 KDC로부터 의뢰를 받은 영역을 검색한 후에 이

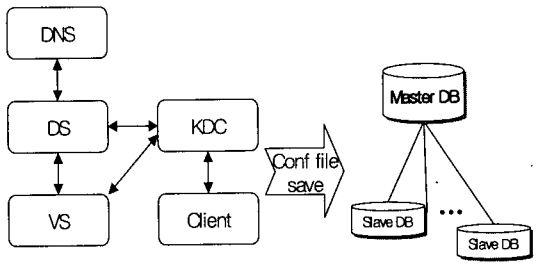

그림 5. Master-Slave KDC 구조

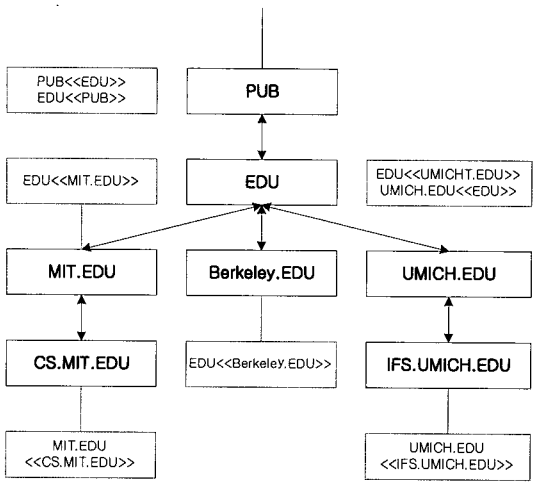

그림 6. 공개키 인증서 체인

웃(Pre-authentication)하는 영역을 디렉토리 시스템에게 전송한다. 클라이언트는 원격 Kerberos에게 X.509[14]를 이용하여 획득한 원격 영역의 공개키로 정보를 암호화하여 전송함으로써 클라이언트와 원격 영역간의 통신을 방해하는 침입자로부터 보호할 수 있게 한다[16,17].

KDC는 구성(Configuration)파일을 작성하며 루트 도메인 EDU과 중간 도메인 MIT, 하위 도메인 CS를 갖는 도메인 이름구조를 CS.MIT.EDU라는 도메인 영역을 생성하고 Ticket 저장소에 저장 그림 7 한다. KDC는 티켓을 요구한 개체에 대하여 인증과정을 수행한 후 티켓 발행이 이루어진다. 통신하고자 하는 영역과 세션이 설정되면 KDC 이름저장소의 영역을 저장한다.

Issuer Name	User's ID	Key	realm	max_life
KDC_r	IDc	PEpk_kdc_r(kc,tgs_r)	cs.mit.edu	10:00:00

max_renewable	expiration	
	from	till
7 days	00:00:00:00	00:00:00:00

그림 7. Kerberos Ticket 저장소

Ⅲ. 효율적인 Kerberos 인증 메커니즘 설계

3.1 인증경로 검증

검증서버는 클라이언트의 인증경로 검증을 대행해

프로토콜	인 증 방 법
OCSP	사용자가 서버에게 Online으로 인증서 상태검증을 요구하는 것으로 서버는 인증서 상태의 증명을 되돌린다.
DPV	OCSP를 기본으로 하고 그 타입을 달리 해 기본타입의 OCSP와 구분할 수 있다. DPVType이라는 새로운 타입을 정의함으로써 OCSP 서버에서 체인생성, 기본검증, 경로검증을 수행하게 한다.
DPD	인증서검증 절차에서 사용자가 체인생성을 서버에게 위임하는 프로토콜로 OCSP를 기본으로 하고 타입을 달리 해 기본타입의 OCSP 또는 DPV와 구분할 수 있다.
SCVP	사용자가 인증서 검증과정의 일부나 전부를 서버에게 위임할 수 있다. 서버는 인증서에 대한 여러 가지 검증결과 뿐 아니라 사용자 자신이 인증서를 검증할 때 필요한 정보들을 제공할 수 있다.또한 체인형성검증, 경로검증, 상태검증을 수행하게 한다.
DVCS	인증서 검증절차, 경로검증, 서명문서의 유효성 확인, 데이터 타임스탬프 서비스, 데이터의 소유 증명서비스를 서버가 제공한다.

그림 8. 인증서 검증 방법

주는 서버로 모든 클라이언트는 검증서버를 신뢰함을 가정한다. 검증서버는 인증경로를 생성하고 인증서 상태를 실시간으로 검증하며 다른 영역의 검증서버와 KDC들과의 상호연동을 통하여 인증경로 전체를 검증하는 서비스를 제공한다. 검증서버에서 사용되는 프로토콜은 클라이언트의 요청을 위한 OCSP, SCVP, LDAP, VADC가 있다. OCSP는 클라이언트로부터 인증서 상태정보 요청 또는 검증서버가 타 도메인의 인증서 상태정보를 얻기 위해서 해당영역의 검증서버에게 요청할 때 사용된다. SCVP는 클라이언트가 인증서 검증, 인증경로 생성, 인증경로 검증 등을 요청할 때 사용된다. LDAP는 검증서버가 인증서 상태정보를 얻기 위해서 사용되고 VADC는 실시간의 인증서 상태정보를 취득하기 위하여 KDC의 DB 정보를 얻기 위한 프로토콜이다. 사용자의 부담을 줄이기 위해 검증과정의 일부나 전부를 인증서 검증서버에 위임하는 서버기반의 여러 가지 방법들인 OCSP(Online Certificate Status Protocols), DPV (Delegated Path Validation), DPD(Delegated Path Discovery), SCVP(Simple Certificate Validation Protocol), DVCS(Data Validation Certification Server Protocols)을 제안하고 있으며 인증방법은 그림 8과 같다

그림 9는 검증서버와 관련하여 사용되는 프로토

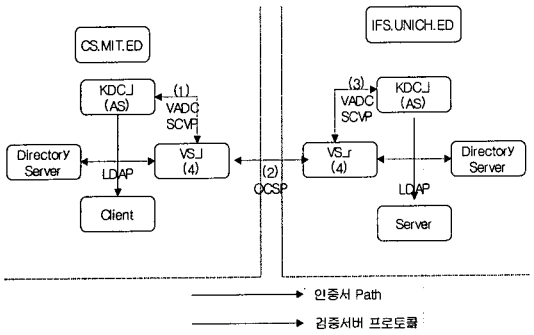

그림 9. Kerberos 도메인 검증

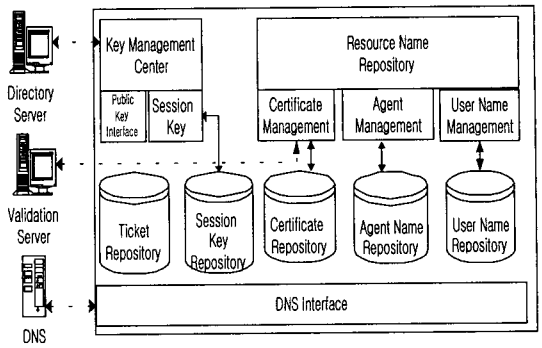

그림 10. Kerberos AS 구조

콜로써 검증서버 상호간 인증서 상태정보(취소여부)를 획득하기 위해 사용되는 OCSP, 서버가 인증서의 검증과 인증경로 생성, 인증경로 검증을 요청할 때 사용되는 SCVP, 검증서버가 인증서의 상태정보를 얻기 위해 사용하는 LDAP가 있다. VADC(VS and AS Data Connection)는 KDC의 인증서 상태정보를 획득하기 위한 프로토콜로 구성된다. VS_r은 IFS.UNICH.ED 도메인에 속해 있는 클라이언트의 인증서 상태정보를 얻기 위하여 VS_l에게 OCSP로 상태검증을 요청한다. VS_l은 VADC로 클라이언트의 실시간 인증서 상태정보를 취득하거나 디렉토리 서버로부터 CRL(인증서 취소목록)을 취득하여 클라이언트 인증서의 상태정보를 얻어서 VS_r에게 응답을 보낸다.

CD.MIT.ED 와 IFS.UMICH.ED 두 도메인 간 인증서 경로생성 과정으로 다음과 같다.

① CD.MIT.ED 도메인의 KDC_l이 KDC_r의 인증서를 검증하기 위해 VS_l에게 요청한다.

② VS_l은 KDC_r의 인증서 상태정보를 얻기 위해 VS_r에게 OCSP로 상태검증을 요청한다.

③ VS_r은 VADC로 KDC_r의 실시간 인증서 상태정보를 획득하거나 디렉토리 서버로부터 CRL을 획득하여 KDC_r의 인증서 상태정보를 얻고 VS_l에게 응답을 보낸다.

④ VS_l은 인증서의 상태정보가 유효한 경우 인증경로를 생성하며 인증경로는 KDC_l(Local KDC)로부터 KDC_r(Remote KDC)까지 (KDC_l)->(KDC_r)을 생성하고 인증경로를 검증한다. 인증서 경로구축과 검증을 위해 검증서버를 사용하며 디렉토리 서버와 DNS는 기존방식을 이용한다. 키 관리센터는 공개키 등록과 인증서 발행, 일정기간 비밀통신을 위한 세션 키 생성과 분배를 담당하며 Ticket 저장소 그림 7, 그림 10과 같이 세션키 저장소를 운용한다[13,14,16].

KDC의 Ticket 저장소는 End Entity들의 Ticket이 생성되고 소멸될 때마다 수시로 갱신된다. 네트워크를 이용한 분산응용 서비스가 증대되는 만큼 보안위협 요소도 증가되어 암호 API가 필요하다. 정보의 일원화된 관리와 동적으로 구성 파일을 변경하기 쉽게 하기 위하여 통제가 가능하도록 그림 11과 같이 Kerberos는 GSS(Generic Security

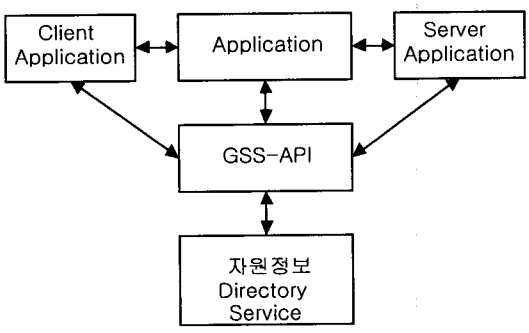

그림 11. Kerberos와 GSS-API 구조

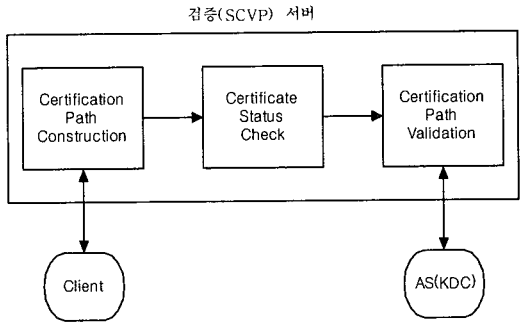

그림 12. 검증서비스 모듈

Service)-API구조를 갖는다.

디렉토리 데이터베이스를 담고 있는 도메인 컨트롤러는 Main server로써 각 도메인에 하나 이상 존재하며 모든 보안관련 사용자 및 도메인 상호작용을 처리하고 관리를 집중시킨다. 클라이언트의 인증서 검증을 실행시켜 주는 검증서비스 모듈(VSM : Validation Service Module)은 그림 12과 같이 인증경로 생성 모듈, 인증서 상태 검증 모듈, 인증경로 검증 모듈로 구성한다. Certificate Status Check는 인증서의 상태를 검증하는 모듈로써 검증서버가 KDC의 DB정보를 얻기 위하여 검증서버는 VADC 프로토콜을 사용한다. VADC는 KDC의 DB가 변경되면 검증 서버측의 DB에게 해당 데이터를 전송한다. 전송되는 메시지의 무결성을 제공하기 위하여 KDC와 검증서버는 인증서를 이용하여 세션키를 교환하고 데이터의 전송시 MAC(Message Authentication Code)을 함께 전송한다.

CRL을 취득하는 방법은 검증서버가 관리하는 디렉토리 시스템의 CRL 주기에 맞추어 CRL을 획득하여 검증서버에 저장해 두는 방법과 인증서 검증이 요구될 때 CRL을 획득하는 방법이 있다. 인증서 검증을 위해서는 가장 최근의 정보를 이용하고 인증서 상태 정보의 취득방법을 클라이언트에게 제공하기 위하여 OCSP, SCVP의 확장필드를 사용한다. Certification Path Construction은 인증경로를 생성하는 모듈로 KDC의 인증서로 이루어진 인증기관 인증경로(CertPath)와 요청에 의한 응답으로 생성된 경로를 사용하여 인증경로를 생성한다. 검증서버는 클라이언트의 요청에 대한 신속한 응답을 지원하고 효율을 높이기 위해 인증기관 인증서로 이루어진 인증경로를 미리 생성하여 저장한다. 검증서버가 지원하는 도메인이 확장됨에 따라 도메인에 속하는 모든 인증기관의 인증서를 수집하여 지원하는 도메인 내에서 생성될 수 있는 모든 인증경로를 생성하고 저장한다. 인증경로를 생성한 후 인증경로 검증작업을 수행함으로써 검증대상 인증서와 인증기관 인증경로를 결합하고 인증경로 검증을 수행할 때 신속하게 처리할 수 있다. 또한 클라이언트 요청으로 인증서 검증에 사용된 인증경로는 저장하여 다른 인증서 검증요청에 재사용할 수 있도록 구성되어 있어 인증경로 생성시간에 소요되는 시간을 효율적으로 줄일 수 있다. Certification Path Validation은 인증경로를 검증하는 모듈로 인증경로가 주어지면 인증서 정책정보, 인증서정책 사상정보, 인증서 상태 체크 모듈을 이용하여 검증한다. 인증경로 검증은 인증 경로구축 모듈에서 생성된 인증경로를 이용하여 검증하게 되며 인증서 상태 체크모듈을 이용하여 인증경로 상의 각 인증서 상태 정보를 획득하여 인증경로 검증을 수행한다[10,15,20].

3.2 티켓의 재획득을 위한 인증 교환 메커니즘

하나의 KDC는 여러 개의 인증서버와 TGS를 둘 수 있다. 각각을 영역으로 정의하고 서로 다른 영역 사이의 인증을 영역간 인증이라 한다. 한 영역에 있는 사용자는 다른 영역에 있는 서버를 접근 하려고 한다면, 어떤 서버는 다른 영역으로부터 온 사용자에게 인증만 된다면 서비스를 제공해야 한다. 따라서 클라이언트가 Remote 서버와 서비스를 종료 후 다시 동일 서비스를 받고자 한다면 먼저 Local 영역에서 인증을 받은 후 Remote 영역에서 접속하여 서버를 재사용할 수 있다. KDC(Key Distribution center)에서 각각 클라이언트에게 요구되는 서비스를 이용할 수 있는 Ticket을 발행해서 각 서비스에 대해 인증한다. Local 데이터베이스 정보 중에서 minimun_lifetime, renewable_lifetime, empty address, proxiable 등이 유효한 경우에는 새로운 Ticket의 maximun_ticket_lifetime을 지연시키고 새로운 Nonce, TimeStamp등을 발행함으로써 통신의 복잡도를 감소시킬 수 있다.

이 Ticket은 Lifetime이 있어 그 유효한 시간 안에서 사용이 가능하며 임의로 폐기하고 다시 얻을 수 있다. 또한 KDC는 모든 인증 단계를 암호화하여 PKI와 같은 방법으로 DES암호화 방법을 접목하여 인증 메커니즘을 구현하고 있다. 따라서

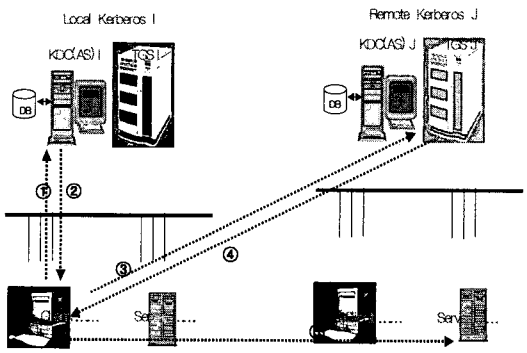

그림 13. 티켓의 재사용 메커니즘

Single sign on으로써 KDC인증 서버로부터 한번 인증을 거치면 다른 시스템을 사용하기 위한 인증 단계가 필요 없다. 즉, 한번 부여 받은 티켓을 통해 forwarding 명령으로 티켓을 사용 가능하다. 사용자는 획득한 티켓을 Lifetime이 끝나기 전에 임의로 티켓을 폐기 할 수 있다. 클라이언트가 Remote Server에 대한 서비스를 받기 위한 메시지 교환 내용은 다음과 같다.

표 3. 파라미터

KDC_l : Local KDC, KDC_r : Remote KDC
TGT : Ticket Granting Ticket
TGS : Ticket Granting Server
SGT : Server Granting Ticket
ID_S : Remote Server
ESKc : Client 개인키로 암호화
EKDC_r : Remote KDC 개인키로 암호화
$Ticket_{TGSREM}$: 서버에 접근권한을 가진 티켓
TGS_{REM} : Remote 영역의 TGS
SGT_{REM} : Remote 영역의 SGT
$EK_{C,TGSREM}$: 클라이언트와 Remote TGS와의 세션키
$EK_{C,SGTREM}$: 클라이언트와 Remote SGT의 세션키
Remote Realms : 원거리 외부영역 -삭제요망
Flag : 티켓의 옵션 상태
KeyIDlist : KDC간 공유하는 함수
SignedAuthPack : 지역영역의 신원인증에 필요한 정보
TrustedCertifiers : Local KDC의 인증서
CertPath : Local KDC와 Remote KDC 간의 인증서체인 경로
CheckSum Type : KeyIDlist의 옵션

(1) 인증서비스

① Client -> KDC_l : (ID_C, ID_S , ID_{TGS})
클라이언트는 자신의 정보를 KDC_l(Local KDC) 로 전송한 후 인증서비스 재요청을 한다.

② KDC_l -> Client : ($Ticket_{TGSREM}$, Nonce, TS, (KeyIDlist))ESK_C
KDC_l은 클라이언트의 Ticket의 Flag확인하여 현재 사용 중인지 여부를 체크, 사용가능한 티켓의 정보(발생시간, 생명주기, 유효시간)등의 유무를 체크한다. 해당 영역의 데이터베이스에서 클라이언트가 요청한 영역정보를 체크 한 후전후방 생명주기(max_renewable_life), 전후방 생명시간(max_life_time)이 유효하면 지연시키고 KDC간 공유하는 함수로 해쉬한 결과(KeyIDlist)와 티켓, 기타 정보(Nonce, TimeStamp)등을 클라이언트의 비밀키로 전공한다. 이때 전·후방 체인경로를 인증티켓에 첨부하고 사용자의 데이터는 암호화하여 첨부한다.

(2) SGT 서비스

③ Client-> TGS_{REM} :(ID_S, $Authenticator_C$, $Ticket_{TGSREM}$, (KeyIDList), ($Ticket_{TGSREM}$, TS, checksum, Nonce, ID_S)E_{KDC_r})$EK_{C,TGSREM}$

④ SGT_{REM} -> Client :(($K_{C,SGTREM}$, $Ticket_{SGTREM}$), TS, Nonce, ID_S , ($K_{C,SGTREM}$, ID_C, AD_C, ID_S, TS, Nonce) EK_{SGTREM})$EK_{C,TGSREM}$
$Authenticator_C$ = (ID_C, AD_C, TS, Nonce) $EK_{C,SGTSREM}$
$Ticket_{SGTREM}$ = EK_{SGTREM}{flags, $K_{C,SGTREM}$, ID_C, AD_C, TS, Nonce}

클라이언트는 티켓($Ticket_{TGSREM}$)과 {KeyIDList}를 가지고 TGS_{REM}에 접근을 요청(③)한다. 클라이언트는 TGS_{REM}에게 티켓과 인증자, 서비스를 요청할 서버의 ID를 포함한 메시지를 보낸다. 부수적으로 클라이언트는 인증자를 전송하는데 여기에는 클라이언트의 ID와 주소, 타임스탬프, 임의수(Nonce)수가 포함되어 있다. TGS_{REM}은 KDC_r의 공유키와 세션키, 자신의 비밀키를 가지고 티켓을 복호화 한다. 이 티켓은 클라이언트에게 세션키($EK_{C,TGSREM}$)로 제공되고 클라이언트 자신만 사용할 수 있다.

Remote TGS_{REM}은 클라이언트로부터 전송된 {KeyIDlist}은 KDC_r에 의해 클라이언트를 보증하고 인증자와 티켓의 정보를 비교하여 무결성 하면 티켓의 소유자라고 인증 할 수 있다. 그리고 경로과정(CertPath)을 통해 신원확인 Local KDC와 클라이언트의 정보(SignedAuthPack, TrustedCertifies)을 Local TGS와 Remote TGS간 전송이 이루워 진다. 메시지(④)는 서버를 사용할 수 있는 새로운 티켓($EK_{C,SGTREM}$,)과 세션키($EK_{C,TGSREM}$)를 생성하고 Remote TGS_{REM}은 서버의 비밀키(EK_{SGTREM})로 티켓을 암호화하여 클라이언트에게 전송한다.

(3) 서비스 요청

⑤ Client-> ID_S : ($Ticket_{SGTREM}$, $Authenticator_C$, ($K_{C,SGTREM}$, ID_C, AD_C, ID_S, Nonce)) $EK_{C,SGTREM}$
$Authenticator_C$ = (ID_C, AD_C, TS, Nonce) $EK_{C,SGTSREM}$
$Ticket_{SGTREM}$ = EK_{SGTREM}{flags, $K_{C,SGTREM}$,

메커니즘	분 석			효 과
	IETF CAT Working Group	신뢰센터	제안 메커니즘	
Local KDC ↔ Remote KDC	Local KDC<-> Remote KDC PKCORSS/ PKINIT를 사용하여 공개키 획득	Local KDC <-> 신뢰센터 후 LocalKDC<-> Remote KDC 후 Remote KDC <-> 신뢰센터확인	Local KDC<-> Remote KDCPKCROSS/PKINIT와 디렉토리 시스템 연계 티켓 저장소	-디렉토리 시스템과 X.509 서명한 공개키 등록과 분배로 안전성 보장
신원확인 정보	ClientPublicValue X509인증서	ClientPublicValue X509인증서	ClientPublicValue X509인증서 SignedAuthPAck,TrustedCertifiers, CertPath	알고리즘식별자, 파라미터, 공개키 정보를 X.509로 효율적인 관리 티켓 저장소에 Flag를 기술하여 티켓을 상태를 파악
안전성	상호인증	신뢰센터 도입	상호인증	디렉토리 시스템과 인증서 사용 인증 검증 서버 도입
신뢰성	상호영역간 Kerberos 신뢰	신뢰센터에서 인증서 및 난수 값 발생	상호영역간 Kerberos 신뢰 인증서 체인 및 검증 서버 티켓 저장소및 세션키 저장소 운용	전후방 체인 사용 티켓저장소를 사용하여 티켓의 상태를 저장
키 분배	키 센터	신뢰센터	키 센터	동일
키관리	KDC에 의해 공개키 비밀키 관리 사용자 마다 세션키 생성 및 보관	신뢰센터 공개키 보관 KDC 비밀키 보유 신뢰센터 공유키 발생	KDC에 의해 공개키 비밀키 관리	KDC 티켓 정장소에 수시 갱신
전송단계 및 재획득 단계	-7단계 -7단계	-12 단계	- 7단계 - 5단계 - OSCP,DPV,SCVP,DVCS	영역간 인증을 사용하여 상호신뢰를 강화 검증기법을 기술하여 상호 신뢰를 강화
티켓 발급	- 1회 사용		- 티켓의 유효시간 활용	티켓의 재획득 과정을 간소화

그림 14. 메커니즘의 비교분석

ID_C, AD_C, TS, Nonce]

클라이언트는 서버를 사용하기 위한 요청으로 서버용 티켓($Ticket_{SGTREM}$)과 인증자, Remote TGS_{REM}로부터 전송된 내용을 보냄으로써 서버로 하여금 인증자와 TGS_{REM}로부터 온 내용을 비교하여 인증하고 세션키($EK_{C,SGTREM}$)로 송수신할 수 있다.

Ⅳ. 메커니즘 분석 및 효과

본 논문에서 제시된 알고리즘은 Kerberos을 기반으로 IETF Working Group에서 사용하고 있는 PKCROSS/PKINIT 메커니즘이며, Kerberos와 X.509에서 보장해 주는 안전성과 DS/DNS에 의한 경로에 대하여 인증서 체인으로 보관하기 때문에 Remote Kerberos에서 클라이언트로 직접 전송할 수 있다. 원거리 통신에서의 보안성을 보장하기 위해서 인증정보를 전달할 때 Kerbeos의 비밀키와 PKCROSS/PKINIT를 이용한 공개키를 사용하였고 상호인증을 위해 디렉토리 시스템과 X.509를 이용 하였다. 제안된 Kerberos 인증 알고리즘 모델링은 3단계 인증 프로토콜로 나누어진다. 그 첫 번째 단계는 (1)~(4)단계로 두 영역간의 연결과 Remote KDC로부터 Ticket(TGT)을 승낙하도록 인정하는 과정을 표현하고, 두 번째 단계는 (5)~(6) Remote TGS로 Ticket를 승낙한 서비스 과정과 클라이언트와 서버사이의 인증을 표현한다. 마지막으로 클라이언트와 서버간의 과정을 표현 했다. 티켓 내에 클라이언트와 TGS_{REM} 클라이언트와 서버사이의 세션키($EK_{C,TGSREM}$, $EK_{C,SGTREM}$) 를 포함시킴으로써 티켓 소유자가 정당한 사용자임을 증명하고 이외에도 클라이언트는 KDC_r에 TGT를 획득하기 위한 별도의 요청을 필요로 하지 않는다. 서버용 티켓은 TGS의 키로 암호화되어 있으므로 변조가 불가능할 뿐만 아니라 클라이언트의 공개키

로 재 암호화하므로 제 3자가 티켓을 이용할 수 없다. 제한된 메커니즘에 대한 비교 분석은 다음과 같다.

V. 결 론

Kerberos는 인증 메커니즘으로 정보보호 기반 기술의 중요한 요소 중의 일부분으로서 동일영역에서 사용자 인증과 키 분배를 위해 사용되는 상호인증 알고리즘이다. 많은 개체들이 안전한 서비스를 지원하기 위해서는 효율적인 상호인증 과 키관리의 측면이 지원 되어야 한다. 분산 네트워크 환경에서 통신하고자 하는 다수의 워크스테이션들과 응용 서버의 인증을 위해서 Kerberos는 X.509 공개키 기반구조를 갖는 PKINIT/PKCROSS(을)를 통해 공개키와 비밀키를 제공하여 안전한 서비스를 지원한다. 즉, 인증기관이 해당 사용자를 확인 및 서명, 공포하여 각 개체들에게 신뢰성과 안전성을 보장 한다. 신뢰센터를 각 영역마다 한 개 이상을 두어서 Kerberos의 키분배 및 생성을 담당 하도록 하였으며, 영역간의 상호 인증을 신뢰센터가 맞도록 되어 있습니다. 신뢰센터 메커니즘은 영역간의 키분배 및 생성에 대한 모든 작업을 신뢰센터에 의해서 이루어지므로 한 영역에 여러 개의 신뢰센터를 두어야 하며 따라서 기존 단계 비하여 더 많은 데이터 전송 단계를 가져오게 된다. 또한 Kerberos간의 상호 신뢰를 기본으로 하는 IETF 그룹과 상이 않을뿐더러 더 많은 정보를 보관하기 위한 작업과 키를 암복화하는 과정이 복잡해진다. Kerberos는 상호신뢰를 기본으로 하고 있기 때문에 신뢰센터 역학을 같이 담당하고 있으며, Kerberos또는 KDC라 하며 AS와 KDC 부분으로 나뉘며 각각 인증부분과 키분배 센터 부분으로 분할된다. 본 논문에서는 IETF 그룹의 기반으로 공개키 암호에 의한 상호인증을 통해 안전하게 키(티켓)를 분배 할 수 있도록 하였다. 상호 신뢰를 강화하기 위해서 전후방 인증서 체인으로 연결하고 신뢰성을 향상시키기 위해서 검증 서버를 적용하여 해당 경로를 검증 하도록 했으며, 티켓 교환단계는 티켓 저장소 및 인증서 검증 모듈, 클라이언트와 서비스 영역, 디렉토리 시스템을 상호 연결을 위한 GSS-API구조를 접목하여 티켓의 이동 과정 및 진행사항을 수시로 갱신하여 티켓 저장소에 저장하도록 하였다. 티켓 저장소를 활용하여 키 교환방식을 이용하여 복구가 가능하고 안전한 서비스를 지원하는 인증 모델을 제안 하였다. 다음 여섯 가지로 요약하면 첫째, 공개키 기반의 X.509, 디렉토리 시스템(DNS포함)을 적용하여 영역간의 인증과 서비스를 제공 하였으며 둘째, 경로를 경로검증 서버(VS)와 디렉토리 시스템을 적용하여 영역간 체인을 통하여 다른 영역의 개체를 인증 하도록 하였으며 셋째, 정보의 관리와 동적으로 구성 파일을 변경하기 쉽게 하기 위하여 통제가 가능하도록 GSS (Generic Security Service)-API 의해 키를 사용자 및 도메인 상호작용을 처리하고 관리를 집중하며, 검증서버로 확인 자료를 티켓 저장소에 실시간으로 변경 하도록 설계하였다. 넷째, Remote TGS 요청에 지역 Kerberos로 요청하도록 하여 Remote Kerberos를 경유하지 않고 티켓을 전송함으로써 통신상의 절차를 간소화를 가지는 Kerberos 인증 메커니즘을 설계 하였다. 다섯째, 티켓의 도용을 대비한 해쉬함수, TimeStamp와 Lifetime 적용하고 변조방지를 위한 KDC와 TGS간의 공유 비밀키로 암호화하며 클라이언트의 식별을 위한 인증자 (Authenticator)를 사용하였다. 여섯째, 재전송 공격을 대비한 임의의 수(Flag, Nonce, Time-Stamp)를 사용하여 현 상태를 티켓 저장소에 보관할 수 있도록 하였다. 신뢰센터를 각 영역마다 한 개 이상을 두어서 Kerberos의 키분배 및 생성을 담당하도록 하였으며, 영역간의 상호 인증을 신뢰센터가 맞도록 되어 있습니다. Kerberos는 상호신뢰를 기본으로 하고 있기 때문에 신뢰센터 역학을 같이 담당하고 있으며, AS(인증부분)와 KDC(키분배센터)을 통틀어 Kerberos또는 KDC라 부르기도 합니다. 신뢰센터 메커니즘은 영역간의 모든 작업을 신뢰센터에 의해서 이루어지므로 한 영역에 여러 개의 신뢰센터를 두어야 하며 따라서 기존 단계 비하여 더 많은 데이터 전송 단계를 가져오게 된다. 그러나 본 논문에서는 IETF 그룹의 기반으로 디렉토리 시스템을 사용하여 X.509를 이용하여 공개키를 획득, 상호 신뢰를 강화하기 위해서 전후방 인증서 체인으로 연결하고 신뢰성을 향상시키기 위해서 검증 서버를 적용하여 해당 경로를 검증 하도록 했으며, 티켓 교환단계는 티켓 저장소 및 인증서 검증 모듈, 클라이언트와 서비스 영역, 디렉토리 시스템을 상호 연결을 위한 GSS-API구조를 접목하여 티켓의 이동 과정 및 진행사항을 수시로 갱신하여 티켓 저장소에 저장하도록 하였다. 클라이언트와 서비스 영역간의 동일 영역의 재접속 과정을 제안하고 있다.

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] J. G. Steiner, B. C. Neuman, and J. I. Schiller, " Kerberos: An Authentication Service for Open Network System," pp. 191-202 in Usenix Conference Proceedings, Dallas, texas (Feb, 1988).

[3] 김철현, "공개키 기반구조하에서 Kerberos 인증 메커니즘의 설계", 조선대학교 석사학위 논문,1999 .

[4] 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-15.txt.

[5] 김철현,정일용,"X.509와DNS이용한 분산인증 알고리즘의 설계",한국정보처리학회추계학술발표논문집,pp.1169-1172, 2000.

[6] 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-08.txt.

[7] K. Hornstein, J.Altman,"Distributing Kerberos KDC and Realm Information with DNS".draft-ietf-krb-wg-krb-dns-locate-02.txt.

[8] J. Kohl, C. Neuman, "The Kerberos Network Authentication Service (V5)", draft-ietf-cat-kerberos-revisions-10.txt.

[9] M. Hur, J. Salowey, " Kerberos Cipher Suites in Transport Layer Security (TLS)", draft-ietf-tls-kerb-01.txt.

[10] A. Medvinsky, M. Hur, S. Medvinsky, C. Neuman. "Public Key Utilizing Tickets for Application Servers (PKTAPP)".

[11] K. Hornstein, J.Altman,"Distributing Kerberos KDC and Realm InformZation with DNS".draft-ietf-krb-wg-krb-dns-locate-02.txt.

[12] A. Gulbrandsen, P. Vixie, " A DNS RR for specifying the location of services (DNS SRV)", RFC2052, October 1996.

[13] P. Mockapetris, "DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION",RFC1035, November 1987.

[14] IETF Draft, "Internet X.509 Public Key Infrastructure Certificate and CRL profile," 1998.

[15] K.Raeburn,"Encryption and Checksum Specifications for Kerberos 5",draft-ietf-krb-wg-crypto-00.txt.

[16] 김철현, 신광철, 김창원,"X.509 인터넷 공개키기반구조에서 Kerberod인증에 관한 연구", 한국컴퓨터산업교육학회, pp.641-652, 2002.

[17] 김철현, 신광철, 정진욱, "The Design of an Optimun Kerberos Mechanism Modeling with Resuability of Ticket", ICIS 2002, pp.231-237, 2002.

[18] 김철현, 정일용, "An Efficient Kerberos Authentication Mechanism Associated With X.509 and DNS", IEICE 2002, pp.1384-1389, 2002.

[19] 김철현, 이여진, 정일용, "The design of an Efficient Kerberos Authentication Mechanism Associated With Directory Systems", GCC2003, pp.721-728, 2003.

[20] 신광철, "공개키 기반구조의 Kerberos의 관한연구", 성균관대 박사 논문, 2003.[20] 신광철, "공개키 기반구조의 Kerberos의 관한연구", 성균관대 박사 논문, 2003.

〈著者紹介〉

김 철 현 (Cheol-hyun Kim)
1996년 2월: 광주대학교 전자계산학과 졸업
2000년 2월: 조선대학교 교육대학원 전자계산교육 석사
2001년 8월~현재: 홍성기능대학 컴퓨터정보과 조교수
2004년 3월~현재: 군산대학교 컴퓨터과학과 박사과정
〈관심분야〉 정보보호, 네트워크

이 연 식 (Yon-sik Lee)
1982년 2월: 전남대학교 전자계산학과 졸업
1984년 2월: 전남대학교 전자계산학과 석사
1994년 2월: 전북대학교 전산응용공학 박사
1986년~현재: 군산대학교 컴퓨터과학과 교수
〈관심분야〉 언어번역 시스템, 능동 및 객체지향 시스템, 에이전트 시스템 응용