論文
3-5-13

X.509 인터넷 공개키 기반구조에서 Kerberos 인증에 관한 연구 (A study on an Efficient Kerberos Authentication based on X.509)

김 철 현* 신 광 철** 김 창 원***
(Cheol-Hyun Kim) (Kwang-Cheul Shin) (Chang-Won Kim)

요 약

본 논문에서는 IETF CAT Working Group에서 발표한 PKINIT기반의 인증서비스를 향상시킨 Kerberos 인증 메커니즘을 제안한다. PKINIT기반의 X.509, DS/DNS를 적용하여 영역간의 서비스를 제공하는 인증과 키 교환 방식으로 DNS를 통해 외부영역의 위치를 탐색하고 X.509 디렉토리 인증 시스템을 적용, 영역간 인증은 DNS 서버로부터 공개키를 획득하여 다른 영역을 인증하도록 하였다. 영역간 인증과 키 교환은 Kerberos의 관용키 암호방식을 사용하고 세션 연결은 X.509 공개키 방식에 기반을 두고 있다. 효율적인 TGT(티켓승인 티켓) 교환과 티켓의 재사용으로 통신상의 Overload를 감소시키는 효과와 인증절차의 간소화를 가지는 Kerberos시스템을 설계하였다.

ABSTRACT

In this paper, proposes Kerberos certification mechanism that improve certification service of PKINIT base that announce in IETF CAT Working Group. Did to certificate other realm because search position of outside realm through DNS and apply X.509 directory certification system, acquire public key from DNS server by chain (CertPath) between realms by certification and Key exchange way that provide service between realms applying X.509, DS/DNS of PKINIT base. In order to provide regional services, Certification and key exchange between realms use Kerberos symmetric method and Session connection used Directory service to connection X.509 is designed using an asymmetric method. By efficient TGT (Ticket Granting Ticket) exchange and reusability of ticket, A Design of Kerberos system that have effect and simplification of certification formality that reduce overload on communication.

1. 서론

분산 환경의 자원보호는 사용자와 서버간의 신원 증명과 안전한 비밀키 교환을 필요로 하다. 신원증명과 비밀키 교환이라는 필요성을 만족시키기 위하여 인증, 무결성, 데이터 보안기능이 필요하다. 이러한 환경에서 대표적인 인증 메커니즘으로 Kerberos[1]와 Yaksha 인증방식[2]이 있으며 여러 응

* 정회원 : 홍성기능대학 전자계산기과 전임강사
** 정회원 : 벽성대학 소프트웨어개발전공 조교수
*** 정회원 : 동강대학 컴퓨터정보과 교수

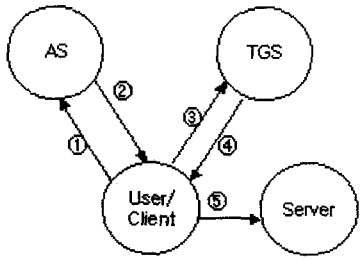
논문접수 : 2002. 4. 18
심사완료 : 2002. 5. 10

용시스템에 호환성을 갖도록 구성된 정보보호 하부 구조로써 Kerberos 메커니즘을 확장한 SESAME[3]이 있다. 본 논문에서는 네트웍 상에서 여러 문제점들을 해결할 수 있는 방안들 중 Kerberos 인증에 관해 중점적으로 연구하였다. Kerberos는 통신망 인증 시스템의 개념과 모델로 중앙 집중식 인증서버를 제공하는 관용암호방식으로 개발되었다[4]. 네트웍 환경에서의 지역을 극복하기 위해 IETF(Internet Engineering Task Force) CAT에서 두 영역과 영역 사이, 인증기관과 지역을 공개키로 상호 서비스해 주는 메커니즘으로 PKINIT(Public Key Cryptography for Initial Authentication)/PKCROSS(Public Key Cryptography for Cross-Realm Authentication)를 사용하고 있다[5]. PKINIT는 DES뿐 아니라 RSA 등 공개키 암호화와 X.509 인증서 기반구조의 Key 관리를 포함하며 Cross-Realm에 대한 인증으로 DNS 사용을 언급하고 있다[6]. 본 논문의 제안은 X.509와 PKCORSS/PKINIT에 기반을 둔 효율적인 Kerberos 인증서비스 교환과 유효시간 만료 후 티켓의 재사용을 위한 메커니즘을 설계하였다. 본 논문의 구성으로 2장에서 Kerberos 인증과 X.509 프로토콜을 설명하였고 3, 4장에서 효율적인 키 교환과 인증, 티켓의 재사용을 위한 메커니즘 설계와 분석을 통하여 제안하고 마지막 장에서 결론을 맺는다.

2. Kerberos 인증과 X.509 프로토콜

2.1 Kerberos 인증

Kerberos 인증 메커니즘[그림 1]은 여러 가지 요소로 구성된 복합시스템으로 Kerberos서버와 TGS, 티켓(Ticket), 인증자로 구성되어 있다. Kerberos 서버와 TGS가 티켓을 생성하여 TGS와 서비스 서버와의 통신에 사용되며 티켓의 구성정보는 서버와 클라이언트 이름, TimeStamp, 유효시간, 세션키를 포함한다. 인증자는 클라이언트에 의해 생성되고 생성된 인증자는 사용을 1회로 제한하고 있으며 인증정보는 클라이언트의 이름과 워스테이션의 IP 주소, 현재의 시간을 포함하고 있다[7][8][9][10].



① Request for TGS Ticket
② Ticket for TGS
③ Request for Server Ticket
④ Ticket for Server
⑤ Request for service

[그림 1] 커버러스 인증 메커니즘

[Fig. 1] Kerberos Authentication Mechanism

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

② AS는 클라이언트의 패스워드로부터 알아낸 키를 가지고 암호화된 티켓으로 응답한다.

③ 서비스-승인 티켓을 클라이언트의 편에서 요구한다. 클라이언트의 ID, 요구하는 서비스의 ID 그리고 티켓-승인 티켓을 포함하고 있는 메시지를 TGS로 전송한다.

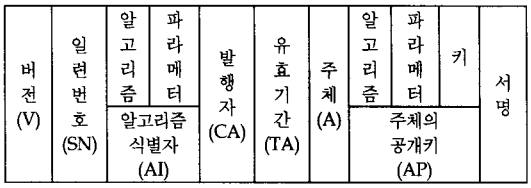
④ TGS는 들어온 티켓을 복호화하고 유효시간을 점검한다. 클라이언트의 ID와 네트워크 주소를 클라이언트의 확인을 위해 들어온 정보와 비교한 후 요구한 서비스에 접속을 승인하는 티켓을 발행한다.

⑤ 서버에 클라이언트의 ID, 서비스-승인 티켓이 포함된 메시지가 전송되면 서버는 메시지의 내용을 이용하여 승인한다.

2.2 X.509 프로토콜

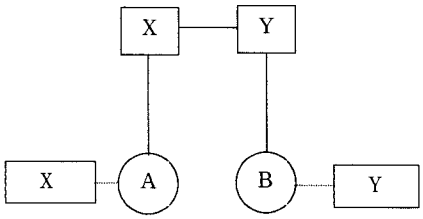
분산 디렉토리 서비스를 구축하기 위한 X.509는 디렉토리에 접근하는 사용자에게 인증 서비스에 대한 규정의 골격을 제공한다. 디렉토리는 공개키의 인증서를 포함할 수 있으며 각 인증서는 공개키와 사용자의 ID를 포함하여 CA(certificate Authority)의 비밀키로 서명하여 생성하고 각 사용자는 CA의 공

개키로 상대방 사용자의 공개키에 대한 확신을 보장받는다. 인증서의 형식은 CA≪A≫=CA{V, SN, AI, CA, A, AP, T}와 같으며 [그림 2]의 표기법을 가진다[11].



[그림 2] X.509 인증서
[Fig. 2] X.509 Certificate

Client가 인증서를 획득하는 과정으로 다음 [그림 3]과 같이 클라이언트 A, B가 있고 인증기관 X, Y가 있다고 가정한다. X.509에서 공개키를 획득하기 위한 인증서의 연결(chain)은 표기로 인증기관 (CA)인 X에 의해 발행된 클라이언트 A의 인증서(X≪A≫)이며 B의 공개키를 획득하기 위하여 인증서의 체인 X≪Y≫Y≪B≫를 사용하고 동일한 방법으로 B는 역방향 체인 Y≪X≫X≪A≫를 이용하여 A의 공개키를 획득한다.



[그림 3] X.509 계층구조
[Fig. 3] X.509 Structure

디렉토리 서버는 DNS와 X.500을 기반으로 도메인을 갖는 한 단위조직의 검색엔진이면서 통합저장소로 영역 내의 객체에 대한 속성을 가지며 Kerberos 내의 위치정보를 DNS, SRV(Service Resource Records), RR[RFC2052]를 사용하여 저장한다[12]. 디렉토리서버(DS)의 구조는 모든 Object와 Attribute를 생성할 수 있는 기반정보를 저장할 수 있는 Schema(스키마 파티션)와 디렉토리내의 Domain구조와 구성정보를 저장하는 Configuration(구성 파티션), Domain 자체의 개체정보가 저장되는 Domain 파티션으로 구성하며 DNS는 호스트명과 IP 어드레스를 서로 매핑시키고 E-mail의 라우팅 정보를 제공하기 위하여 TCP/IP 어플리케이션에 의해 사용되는 기반 서비스이자 프로토콜로 호스트 이름에 대한 분산된 데이터베이스라고 할 수 있다. 즉 계층적 이름을 인터넷의 주소로 또는 그 반대로 변환하는 것을 의미한다. 본 논문에서 디렉토리 인증 프로토콜인 X.509를 이용하며 연결, 외부 영역에 있는 서비스를 얻는다.

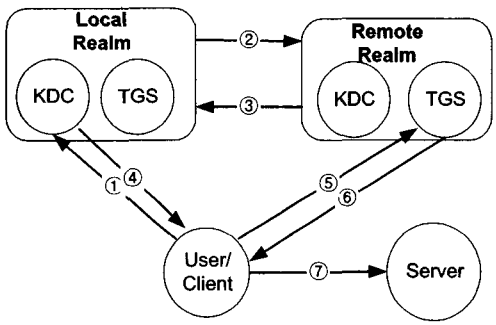
3. 효율적인 Kerberos 인증 메커니즘 설계

3.1 개요

본 논문에서 제시한 인증 메커니즘은 영역과 영역간 서비스를 위한 인증으로 기본 환경은 KDC와 티켓을 발행하는 TGS, 영역 내 객체의 위치를 제공하고 DNS를 사용하기 위한 디렉토리시스템, 비밀키의 생성 및 분배를 위한 키 관리센터, End User의 ID와 패스워드를 저장한 중앙DB, 자원서비스를 위한 서버, 서버를 사용할 Client가 하나의 도메인으로 구성되어 있다. 인증서는 인증기관이 전자서명을 통하여 전자서명 공개키와 이를 소유하는 자연인 또는 법인과의 귀속관계를 확인, 증명하는 전자적 정보로써 본 논문에서는 Kerberos의 KDC가 인증기관과 키분배 센터의 역할을 담당하며 상호 신뢰성을 확인하는데 사용한다. 각 영역의 Client와 자원들은 KDC에게 ID와 패스워드를 Install 때 등록하여 AS의 데이터베이스에 저장되며 이는 동일영역에서의 인증을 위해 사용하고 영역간에는 KDC가 Client를 인증하고 X.509 인증서(KDC≪C≫)를 발행하여 Client를 보증해 준다. 다수의 웍스테이션들이 서비스를 받기 위해서는 TGS로부터 받은 티켓을 사용하여 서버에게 인증 받는다. 이 티켓은 그 사용시간이 서비스의 종류나 클라이언트의 권한에 따라 제한되어 있으며 티켓을 사용하는 클라이언트의 신분을 증명해 주기 위해 여러 가지 인증정보를 포함하고 있다.

3.2 IETF의 Kerberos 인증 메커니즘

IETF의 Kerberos인증 메커니즘에서는 티켓을 발급 받기 위해 원격 Kerberos가 지역 클라이언트를 확인하는 과정을 갖는다. 지역 Kerberos를 통하여 TGS(Ticket Granting Server)를 접근할 수 있는 티켓과 원격 TGS가 서버에 접근할 수 있는 티켓인 SGT(Server Granting Ticket)을 발급하는 과정의 메커니즘으로 구성되어 있다.

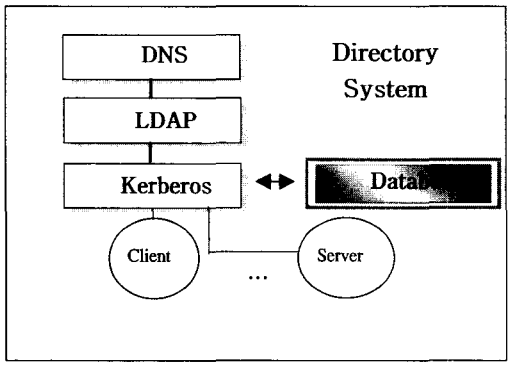


[그림 4] IETF의 인증 메커니즘
[Fig. 4] Authentication Mechanism of IETF

[그림 4]는 IETF의 영역간 Kerberos 인증절차로써 상호간 인증서와 암호알고리즘 등 인증서비스를 교환하고 원격 TGS 접근을 위한 티켓과 세션키를 교환하는 TGT 서비스 교환(①-④), 서버 접근용 티켓과 세션키를 교환하는 TGS 교환(⑤-⑥), 세션키에 의한 서비스 요청과정(⑦)으로 원격 KDC가 지역 KDC의 정보를 인증한 후 티켓을 발급하고 있다.

3.3 디렉토리 시스템(Directory System)

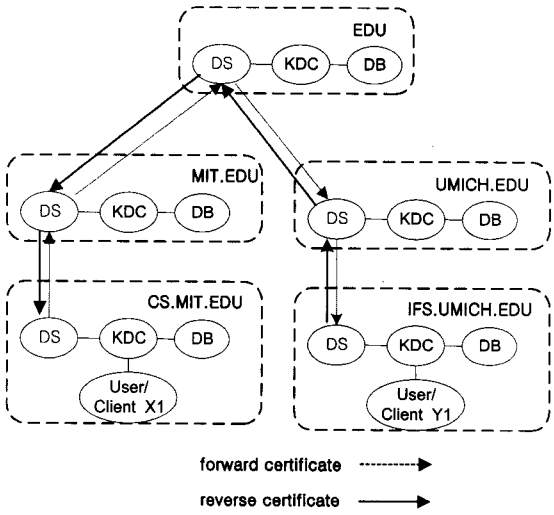
모든 Kerberos의 공개키는 디렉토리시스템에서 획득하게 된다. 저장되는 KDC의 공개키는 디렉토리시스템에 의해 데이터 무결성과 데이터의 인증을 보장받는다. 이 공개키 인증서는 PKCROSS/PKINIT에 의한 초기 인증을 목적으로 원격 KDC의 공개키를 획득하기 위해 디렉토리시스템을 이용한다. 디렉토리 서비스는 데이터베이스, 파일, 호스트 연결, 사용자 서비스 등 모든 자원에 대한 관리를 허용하고 위치 서비스로써 인터넷 DNS을 사용하여 여러 도메인을 트리구조로 연결시킨다. 지역 Kerberos는 지역 클라이언트가 요청한 영역이 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 5]와 같다. 구성은 DNS와 디렉토리 서비스에 접근하기 위한 인터넷 표준(RFC1777) 프로토콜인 LDAP, 인증과 키 교환을 실현 할 Kerberos, Database로 되어 있다.



[그림 5] 디렉토리 서비스의 구조
[Fig. 5] Structure of Directory Service

[그림 6]은 디렉토리를 인증하기 위해 X.500의 디렉토리 시스템을 이용하여 외부영역에 있는 목적지까지 경로를 연결하는 세션과정을 도식한 것이다. 여기에서 X.500의 디렉토리 시스템의 형식은 Domain, X500, Other 그리고 Reserved로 구성된다.

```
Domain : host.subdomain.domain
X500   : C=US/O=OSF
Other  : NameType
Reserved : reserved
```



[그림 6] 디렉토리 서버인증

[Fig. 6] Directory Server Authentication

CS.MIT.EDU영역에 있는 클라이언트가 IFS.UMICH.EDU영역에 있는 서비스를 사용하기 위한 내용으로 MIT.EDU영역의 클라이언트는 선인증으로 EDU영역과 연결을 한 후 다시 UMICH.EDU영역과 연결을 하게 된다. UMICH.EDU영역은 서브영역인 IFS.UMICH.EDU영역과 연결을 하게 된다. MIT.EDU와 EDU연결, EDU영역과 UMICH.EDU영역 연결, 그리고 UMICH.EDU영역과 IFS.UMICH.EDU영역 연결한다. MIT.EDU영역의 클라이언트는 IFS.UMICH.EDU영역에 있는 서비스를 사용하기 위한 전방 인증서(Forward Certificate : CS.MIT.EDU《MIT.EDU》MIT.EDU《EDU》EDU≪UMICH.EDU≫UMICH.EDU≪IFS.UMICH.EDU≫)와 후방 인증서(Reverse Certificate : IF.UMICH.EDU《UMICH.EDU》UMICH.EDU≪EDU≫EDU≪MIT.EDU≫MIT.EDU《CS.MIT.EDU》CS.MIT.EDU《X1》)가 체결된다. 클라이언트가 있는 영역인 CS.MIT.EDU영역과 IFS.UMICH.EDU간 연결이 직접적으로 이루어졌다. 문제는 침해자가 서비스를 요청한 클라이언트처럼 가장하여 서비스를 가로채거나 변경시킬 수 있기 때문에 상호영역간에 있어서 클라이언트를 인증하는 절차를 필요로 하게 된다. 클라이언트는 원격 Kerberos에게 X.509를 이용하여 얻은 원격 영역의 공개키로 정보를 암호화하여 전송함으로써 클라이언트와 원격 영역간의 통신을 방해하는 침입자로부터 보호할 수 있게 한다. 디렉토리 시스템이 전·후방 인증서를 통하여 체인으로 연결되면 KDC는 Kerberos Ticket을 발행한다. KDC는 하나의 Master와 여러 개의 Slave로 구성되며 각각Kerberos 데이터베이스를 보유한다. Slave KDC는 데이터베이스의 복사본들을 유지하며 데이터베이스의 추가나 변경 삭제 등은 Master KDC에서만 가능하다.

```
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 7] Kerberos의 도메인영역

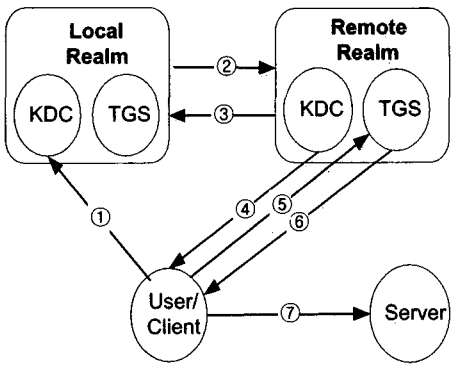
[Fig. 7] Kerberos's Domain Realm

즉 Salve KDC는 Ticket만을 발급해 주는 역할만 하고 경로를 설정하는 것은 Master KDC의 역할이다. [그림 7]은 루트 도메인 EDU와 최상위 도메인 MIT, 그리고 서브 도메인 AHTENA를 갖는 도메인 이름구조를 AHTENA.MIT.EDU라는 [Realms]에 생성하고 데이터베이스에 저장한다.

3.4 승인 티켓을 위한 인증 서비스 교환 메커니즘

Client가 요청한 서비스가 동일한 영역 내에 있는 서비스이면 KDC의 데이터베이스에서 Client의 정보로 인증을 하게 되고 요청한 서비스가 동일 영역 내에 존재하지 않으면 KDC는 Client가 요청한 영역이 어디에 존재하는지 디렉토리시스템을 통하여 DNS에게 검색을 의뢰한다. DNS 서버는 정방향 조회영역, 캐쉬 루트서버를 이용하여 리졸빙 후 캐쉬영역에 저장 한 후 KDC로부터 의뢰를 받은 영역을 검색한

후에 이웃(Pre-authentication)하는 영역을 디렉토리시스템에게 전송한다. 클라이언트는 원격 Kerberos에게 X.509를 이용하여 획득한 원격 영역의 공개키로 정보를 암호화하여 전송함으로써 클라이언트와 원격 영역간의 통신을 방해하는 제3자로부터 보호할 수 있게 한다.



[그림 8] 승인 티켓을 위한 메커니즘
[Fig. 8] Mechanism for Granting Ticket

[그림 8]은 서로 다른 영역의 Local Realm(MIT.EDU)과 Remote Realm(IFS.UMICH.EDU)의 환경으로 KDC는 TGS가 사용할 티켓(TicketTGSREM) 만을 발급하는 역할을 하며 티켓에는 발급자, 세션키, 발급대상의 ID와 주소, 발행시간, Reply 방지용 값을 포함한다. TGS는 SGT인 TicketSGTREM를 발급하는 서비스를 담당한다. Kerberos는 인증과 KDC의 역할을 하며 Local Client가 Remote Server의 서비스를 받기 위한 메시지 교환 내용은 다음과 같다.

(1) 인증 서비스

① ID_C, $Realm_S$
② $E_{KDC\_r_{PK}}$[SignedAuthPack, TrustedCertifiers, CertPath]
③ $E_{KDC\_l_{PK}}$[KDC_r 《KDC_l》 ,KDC_r 《C》]
Client는 자신의 ID와 서비스를 원하는 영역을 자신의 영역에 있는 Local KDC에게 전송(①)하여 서비스를 요청하고 서비스 영역이 다를 경우 KDC_{LOC}에 의해 Directory System의 DNS를 통하여 검색을 의뢰하게 된다. 해당영역의 Directory System은 DNS로부터 받은 서비스 영역에 관하여 상호인증을 하기 위한 전·후방 인증 체인을 생성하여 Remote 영역의 공개키를 획득(PKINIT)하게 된다. Local KDC는 Client와 자신의 정보(SignedAuthPack, TrustedCertifiers) 그리고 CertPath를 전송(②)하여 신원확인 및 티켓승인 티켓을 요청하며 이 전송 정보에는 KDC_l과 Client에 대한 시간, 암호알고리즘, 유형, 인증서, 자신의 위치를 확인시키기 위하여 URL 값을 갖는 CertPath를 포함하고 있다. KDC_r은 정당한 사용자라고 인증한 결과를 KDC_l에게 전송(③)하여 인증 받았음을 확인한다.

○ Notation

- ID_C : Client의 식별자(C의 ID)로 KDC_l에 알림
- $Realm_S$: 서버 S의 영역에 접근요구
- KDC_r_{PK} : Remote KDC의 공개키로 PKINIT로 획득
- SignedAuthPack : 지역영역의 신원인증에 필요한 정보로 PkAuthenticator, ClientPublicValue의 값
  - PkAuthenticator : 지역영역의 KDC(AS)의 정보로 cusec, ctime, nonce, PaChecksum으로 구성
    - . cusec : Client의 인증자를 발행한 시간
    - . ctime : KDC_l의 인증자를 발행한 시간
    - . Nonce : Reply가 아니라는 정당한 데이터의 무결성 보장
    - . PaChecksum : ASN.1에서 정의한 암호화 알고리즘의 종류로 cksumtype과 checksum을 포함
      - cksumtype : 암호 알고리즘 유형을 선택하는 정수 값
      - checksum : 암호 알고리즘으로 crc32, rsa-md4, rsa-md4des, rsa-md5, rsa-md5des,des-mac
  - ClientPublicValue : Client의 공개정보로 알고리즘과 인증서 소유자의 공개키 값
- TrustedCertifiers : KDC_l의 인증서로 principalName, caName, issuerAndSerial, UserCert 중 선택
  - principalName : KerberosName
  - caName : X.500, X.509 검증을 거친 Name
  - issuerAndSerial : Client와 KDCs가 신뢰할 수 있는 CA 번호
  - UserCert : Client의 RSA암호화 증명서
- CertPath : KDC_l 과 KDC_r과의 인증서 체인
- KDC_l_{PK} : Local KDC의 공개키
- KDC_r 《KDC_l》 : 원격 KDC가 지역 KDC에 발행하는 인증서
- KDC_r 《C》 : 원격 KDC가 지역 Client에 발행하는 인증서

(2) TGT 서비스

④ E_{PKC}[TicketTGSREM, KC,TGSREM, TimeStamp, Nonce, RealmTGSREM, E_{KDC_rSK}[TicketTGSREM, TimeStamp, PaChecksum, Nonce, RealmTGSREM]]
TicketTGSREM = $E_{KTGSREM}$[flags, KC,TGSREM, IDC, ADC, TimeStamp, Nonce]

KDC_r은 KDC_l로 Client를 인증하고 KDC_r 영역의 TGS_{REM}에게 Client의 인증을 확인시키기 위해 공유키(E_{KDC_rSK})로 티켓과, 티켓발행시간, 암호알고리즘, 임의의 수, TGS_{REM}의 영역을 암호화하고 세션키($K_{C,TGSREM}$)와 티켓($Ticket_{TGSREM}$)을 KDC_l로부터 추출한 Client의 공개키로 전송(④)한다. 이때 KDC_r은 무결설을 보장하는(Nonce)와 $Ticket_{TGSREM}$을 자신 영역의 TGSREM에게 전송함으로써 Client로부터 전송될 내용과 비교하도록 하여 정당한 사용자라는 것을 확신시킨다.

○ Notation

- P_{KC} : Client C의 공개키
- $Ticket_{TGSREM}$: $E_{KTGSREM}$[flags, $K_{C,TGSREM}$, ID_C, AD_C, TimeStamp, Nonce]로 원격 TGS 사용권한을 가진 티켓, KDC_r에 의해 인증
- $E_{KTGSREM}$: 원격 TGS의 비밀키, KDC_l과 TGS만이 키로 암호화
- flags : 티켓의 옵션상태
- $K_{C,TGSREM}$: Client와 원격 TGS와의 세션 키
- AD_C : Client의 Address로 초기에 티켓을 요구, Client외의 사용을 예방
- TimeStamp : 티켓이 생성된 시간
- Nonce : Replay 방지를 위한 임의의 수
- $Realm_{TGSREM}$: 원격 TGS의 영역
- KDC_rsk : 원격 KDC와 TGS의 공유키
- PaChecksum : 암호알고리즘 유형

(3) SGT 서비스

⑤ $E_{KC,TGSREM}$[ID_S, A_C, $Ticket_{TGSREM}$, E_{KDC_rSK}[$Ticket_{TGSREM}$, TimeStamp, PaChecksum, Nonce, $Realm_{TGSREM}$]]

⑥ $E_{KC,TGSREM}$[$K_{C,SGTREM}$, $Ticket_{SGTREM}$, TimeStamp, Nonce, $Realm_{SGTREM}$, ID_S, $E_{KSGTREM}$[$K_{C,SGTREM}$, ID_C, AD_C, ID_S, TimeStamp, nonce]]

A_C = $E_{KC,TGSREM}$[ID_C, AD_C, $Realm_{TGSREM}$, TimeStamp, Nonce]
$Ticket_{SGTREM}$ = $E_{KSGTREM}$[flags, $K_{C,SGTREM}$, $Realm_{SGTREM}$, ID_C, AD_C, TimeStamp, Nonce]

Client는 이제 티켓($Ticket_{TGSREM}$)과 세션키($K_{C,TGSREM}$)를 가지면 TGS_{REM}에 접근할 준비가 된다. 메시지(⑤)에서 Client는 TGS_{REM}에게 티켓과 인증자, 서비스를 요청할 서버의 ID를 포함한 메시지를 보낸다. 부가적으로 Client는 인증자를 보내는데 여기에는 ID와 Client의 주소, Timestamp, 임의의 수가 포함되어 있다.

재사용할 수 있는 티켓과는 다르게 인증자는 한 번만 사용되기 때문에 짧은 유효시간을 갖는다. TGS_{REM}은 KDC_r의 공유키와 세션키, 자신의 비밀키를 가지고 티켓을 복호할 수 있다.

이 티켓은 Client에게 세션키($K_{C,TGSREM}$)가 제공되었음을 가리키기 때문에 $K_{C,TGSREM}$을 사용하는 사람은 Client 뿐이다. 원격 TGS_{REM}은 Client로부터 전송된 인증자와 티켓의 정보와 비교하여 일치하면 티켓을 보낸 사람은 실제 티켓의 소유자라고 인증할 수 있다.

메시지(⑥)는 서버를 사용할 수 있는 티켓($Ticket_{SGTREM}$)과 세션키($K_{C,SGTREM}$)를 생성하고 원격 TGS_{REM}은 서버의 비밀키(K_{SGTREM})로 티켓과, 유효시간, 임의의 수, 영역을 암호화하고 Client와 TGS_{REM} 간의 세션키로 암호화하여 Client에게 전송한다. Client는 서버의 비밀키(K_{SGTREM})로 된 내용을 확인할 수 없다.

○ Notation

- $K_{C,TGSREM}$: Client와 원격 TGS_{REM}와의 세션키로 변조방지의 비밀키
- ID_S : 서버 S의 식별자로 Client가 TGS_{REM}에 대한 Access 요구
- A_C : Client의 인증자로 $Ticket_{TGSREM}$과 비교, 인증
- $Ticket_{TGSREM}$: 원격 TGS 사용권한을 가진 티켓
- $Realm_{TGSREM}$: 원격 TGS의 영역
- $K_{C,SGTREM}$: Client와 원격 서버 S간의 비밀키
- $Ticket_{SGTREM}$: 서버 S에 접근권한을 가지는 티켓
- $Realm_{SGTREM}$: 서버 S의 영역
- K_{SGTREM} : 서버S의 비밀키로 변조 방지를 위해 TGS_{REM}과 서버만이 알고 있는 비밀키
- AD_C : Client의 Address로 티켓의 내용과 동일해야 한다
- $Ticket_{SGTREM} = E_{KSGTREM}[flags, K_{C,SGTREM}, Realm_{SGTREM}, ID_C, AD_C, TimeStamp, Nonce]$

(4) 서비스 요청

⑦ $EK_{C,SGTREM}[Ticket_{SGTREM}, A_C, EK_{SGTREM[KC,SGTREM}, ID_C, AD_C, ID_S, Nonce]]$

$A_C = EK_{C,TGSREM}[ID_C, AD_C, Realm_{TGSREM}, TimeStamp, Nonce]$

$Ticket_{SGTREM} = EK_{SGTREM}[flags, K_C, SGTREM, Realm_{SGTREM}, ID_C, AD_C, TimeStamp, Nonce]$

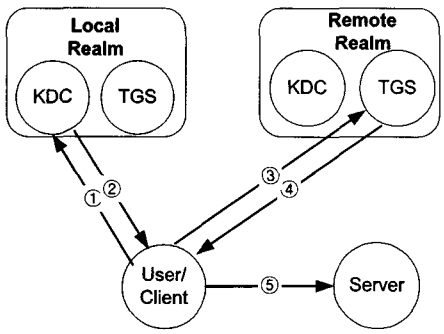
메시지(⑦)에서 Client는 서버를 사용하기 위한 요청으로 서버용 티켓($Ticket_{SGTREM}$)과 인증자, 원격 TGS로부터 전송된 내용을 보냄으로써 서버로 하여금 인증자와 TGS로부터 온 내용을 비교하여 인증하고 세션키($K_{C,SGTREM}$)로 송수신할 수 있다.

○ Notation

- $K_{C,SGTREM}$: Client와 원격 서버 S간의 세션키로
- $Ticket_{SGTREM}$: 서버 S에 접근권한을 가진 티켓
- A_C : Client의 인증정보가 수록되어 $Ticket_{SGTREM}$과 비교하여 인증
- K_{SGTREM} : 원격 서버 S의 비밀키로 TGS_{REM}이 전송한 것을 확인
- Nonce : 메시지에 대한 무결성 확인용
- $A_C = EK_{C,TGS}[ID_C, AD_C, Realm_{TGS}, TimeStamp, Nonce]$

3.5 티켓의 재사용을 위한 인증 교환 메커니즘

Client가 Remote 서버와 서비스를 종료한 후 다시 동일 서버의 서비스를 받고자 한다면 [그림 8]과 같은 서비스를 단계를 다시 갖는다. Local 데이터베이스 정보 중에서 maximum_ticket_lifetime이 경과했지만 minimum_lifetime 및 renewable_lifetime, empty addresses, proxiable 등이 유효한 경우에는 새로운 티켓의 maximum_ticket_lifetime만을 지연시키면 통신의 복잡도를 감소시킬 수 있다.



[그림 9] 티켓의 재사용 메커니즘

[Fig. 9] Reusability Mechanism of Ticket

티켓의 Lifetime이 종료된 후 Local Client가 이전 영역에 대한 동일한 유형의 서비스를 받기 위한 메시지 교환은 다음과 같다.

(1) 인증 서비스

① ID_C, $Realm_S$

② $ESK_C[Ticket_{TGSREM}, Nonce, TS, \{KeyIDList\}]$

KeyIDList	: cksumtype	(crc-32, rsa-md4, rsa-md4des, rsa-md5, rsa-m5des, des-mac, rsa-md4-des-k, desmac-k)
	checksum	{F(PKC)}SKC
		F( ) Hash Function

KDC_{LOC}는 데이터베이스의 영역정보를 체크한 후 max_renewable_life의 범위에서 max_life_time을 지연시키고 KDC간 공유하는 함수로 해쉬한 결과(KeyIDList)와 티켓, 임의의 수, TS를 Client의 비밀키로 전송(②)한다.

(2) SGT 서비스

③ $EK_{C,TGSREM}[ID_S, A_C, Ticket_{TGSREM}, \{KeyIDList\}$ $E_{KDC\_RSK}[Ticket_{TGSREM}, TimeStamp, PaChecksum, Nonce, Realm_{TGSREM}]]$

④ $EK_{C,TGSREM}[KC,SGTREM, TicketSGTREM1, TimeStamp, Nonce, Realm_{SGTREM}, ID_S, EK_{SGTREM}[K_{C,SGTREM}, ID_C, AD_C, ID_S, TimeStamp, nonce]]$

$A_C = EK_{C,TGSREM}[ID_C, AD_C, Realm_{TGSREM}, TimeStamp, Nonce]$

$Ticket_{SGTREM}1 = E_{KSGTREM}[flags, K_{C,SGTREM}, Realm_{SGTREM}, ID_C, AD_C, TimeStamp, Nonce]$

Client는 티켓($Ticket_{TGSREM}$)과 {KeyIDList}를 가지고 TGS_{REM}에 접근을 요청(③)한다. Client는 TGS_{REM}에게 티켓과 인증자, 서비스를 요청할 서버의 ID를 포함한 메시지를 보낸다. 부가적으로 Client는 인증자를 보내는데 여기에는 ID와 Client의 주소, Timestamp, 임의의 수가 포함되어 있다. TGS_{REM}은 KDC_r의 공유키와 세션키, 자신의 비밀키를 가지고 티켓을 복호할 수 있다.

이 티켓은 Client에게 세션키($K_{C,TGSREM}$)가 제공되었음을 가리키기 때문에 $K_{C,TGSREM}$을 사용하는 사람은 Client 뿐이다. 원격 TGS_{REM}은 Client로부터 전송된 (KeyIDList)은 KDC_{REM}에 의해 Client를 보증하고 인증자와 티켓의 정보를 비교하여 일치하면 티켓을 보낸 사람은 실제 티켓의 소유자라고 인증할 수 있다. 메시지(④)는 서버를 사용할 수 있는 새로운 티켓($Ticket_{SGTREM}1$)과 세션키($K_{C,SGTREM}$)를 생성하고 원격 TGS_{REM}은 서버의 비밀키(K_{SGTREM})로 티켓과, 유효시간, 임의의 수, 영역을 암호화하고 Client와 TGS_{REM}간의 세션키로 암호화하여 Client에게 전송한다. Client는 서버의 비밀키(K_{SGTREM})로 된 내용을 확인할 수 없다.

(3) 서비스 요청

⑤ $EK_{C,SGTREM}[Ticket_{SGTREM}1, A_C, EK_{SGTREM}[K_{C,SGTREM}, ID_C, AD_C, ID_S, Nonce]]$

$A_C = EK_{C,TGSREM}[ID_C, AD_C, Realm_{TGSREM}, TimeStamp, Nonce]$

$Ticket_{SGTREM}1 = E_{KSGTREM}[flags, K_{C,SGTREM}, Realm_{SGTREM}, ID_C, AD_C, TimeStamp, Nonce]$

Client는 서버를 사용하기 위한 요청으로 서버용 티켓($Ticket_{SGTREM}$)과 인증자, 원격 TGS로부터 전송된 내용을 전송(⑤)함으로써 서버에게 인증자와 TGS로부터 온 내용을 비교하여 인증하고 세션키($K_{C,SGTREM}$)로 송수신할 수 있다.

4. 메커니즘 분석 및 효과

IETF CAT Working Group에서 사용하고 있는 Kerberos 메커니즘은 PKINIT의 기반의 PKIX로 공개키와 공통키를 사용하여 인증정보에 대한 무결성을 보장하고 있으나 도메인간 연결정보에 대해서는 DNS방법을 언급만 하고 있다.

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 체인(CertPath : Domain Value)으로 보관하기 때문에 원격 Kerberos에서 Client로 TGT를 직접 전송할 수 있다[그림 10].

즉 Client는 KDC_r에 TGT를 획득하기 위한 별도의 요청을 필요로 하지 않는다. 서버용 티켓은 TGS의 키로 암호화(K_{TGSREM})되어 있으므로 변조가 불가능할 뿐만 아니라 Client의 공개키로 재 암호화하므로 제 3자가 티켓을 이용할 수 없다.

메커니즘	분 석		효 과
	IETF CAT Working Group	제안 메커니즘	
KDC_l ↔ KDC_r	- PKCORSS/ PKINIT를 사용하여 공개키 획득	- PKCROSS/PKINIT와 디렉토리시스템 연계	디렉토리시스템과 X.509 서명한 공개키 등록과 분배로 안전성 보장
신원확인	- ClientPublicValue - X509인증서	- ClientPublicValue - X509인증서 SignedAuthPack, TrustedCertifiers, CertPath	알고리즘식별자, 파라미터, 공개키 정보를 X.509로 효율적인 관리
전송단계	- 7단계	- 7단계 - 5단계	- 영역간 인증을 사용하여 상호신뢰를 강화
티켓발급	- KDC_l→ KDC_r→ KDC_l→ Client→TGSREM,	- KDC_r → Client	절차의 간소화

[그림 10] 메커니즘 분석

[Fig. 10] Mechanism Analysis

티켓 내에도 Client와 TGS_{REM}, Client와 서버사이의 세션키($K_{C,TGSREM}$, $K_{C,SGTREM}$)를 포함시킴으로써 티켓 소유자가 정당한 사용자임을 증명한다. 또한 동일 영역에 서비스를 요청해도 유효시간이 소멸되면 처음과정의 절차를 수행하지만 Ticket Lifetime 및 renewable lifetime 이 유효하면 TGS_{REM}에게 서비스를 직접 요청할 수 있도록 하였다.

본 논문에서 제시된 알고리즘은 원거리 통신에서의 보안성을 보장하기 위해서 인증정보를 전달할 때 Kerberos의 비밀키와 PKCROSS/PKINIT를 이용한 공개키를 사용하였고 상호인증을 위해 X.509와 디렉토리시스템을 이용한 체인방식으로 원거리 통신을 보다 더 안전성이 보장되는 Kerberos시스템을 설계하였다. 뿐만 아니라 Client의 유효시간이 종료된 경우에도 처음 과정을 반복을 배제함으로써 티켓 요청 단계를 간소화하였다.

5. 결 론

정보보호 기반기술의 중요요소인 인증 메커니즘으로 관용 암호방식을 사용하는 Kerberos는 동일영역에서 최적의 상호인증 알고리즘이다. 분산 네트워환경에서 통신하고자 하는 다수의 워스테이션들과 응용서버의 인증을 위해서 Kerberos는 X.509 공개키 기반구조를 갖는 PKINIT를 통해 공개키와 비밀키를 제공하여 안전한 서비스를 지원한다. 본 논문에서는 인증 메커니즘인 Kerberos와 초기 인증과정에서 공개키 암호 사용에 대한 정의를 기술한 PKINIT/PKCROSS와 PKIX의 인증시스템인 X.509를 고찰하였다. PKINIT기반의 X.509, 디렉토리시스템/DNS를 적용하여 영역간의 인증과 서비스를 제공하는 인증과 키 교환방식과 티켓의 유효시간 만료 후 Remote TGT에 재접속을 위해 Remote KDC와 공유하는 해쉬함수를 적용하여 티켓을 재사용하는 메커니즘을 제안하였다. 경로(CertPath)는 DNS를 통해 외부영역의 위치를 탐색하여 데이터베이스에 저장하고 공개키 획득은 X.509 디렉토리 인증 시스템인 디렉토리시스템을 적용하였으며 영역간 체인을 통하여 다른 영역을 인증하도록 하였다. Local Kerberos를 경유하지 않고 Client로 직접 티켓을 전송함으로써 통신상의 Overload를 감소시키는 효과와 인증절차의 간소화를 가지는 Kerberos 시스템을 설계하였다.

※ 참고문헌

[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] 최용락, 소우영, 이재광, 이임영 "통신망 정보보호", 그린출판사, pp.343-393, 2001.

[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] http://www.ietf.org/internet draft-ietf-dnsop/keyhand-00.txt, IETF, 1999.

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

[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-08.txt

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

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

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

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

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

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

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

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

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

김 철 현

2000년 조선대학교 교육대학원 전자계산교육(교육학석사 : 컴퓨터전공)
1996년 광주대학교 공과대학(공학사)
2001년 현재 홍성기능대학 전자계산기과 전임강사
관심분야 : 네트워크, 컴퓨터 보안

신 광 철

1995년~1999년 성균관 대학원 정보공학과 수료
1989~1990 국방대학원 전자계산과졸업
1991년~1995년 전쟁연습 프로그램관 및 전산실장 (육군대학)
1985년~1988년 제도분석 및 프로그램관(중앙전산소)
1996년~현재 벽성대학 소프트웨어개발전공 조교수
관심분야 : 정보보호기술, 객체지향 분석/설계, 전자상거래 응용, Visual Programming,

김 창 원

2002년 조선대학교 대학원 전산통계학과(이학박사 : 컴퓨터전공)
1985년 조선대학교 대학원 산업공학과(공학석사 : 컴퓨터전공)
1981년 조선대학교 공과대학(공학사)
1987년~ 현재 동강대학 컴퓨터정보과 교수
관심분야 : 인공지능, 영상처리, 신경망, 컴퓨터 보안