PKINIT기반의 Kerberos 인증과 키 교환에 관한 연구

신 광 철[†] · 정 일 용[††] · 정 진 욱[†††]

요 약

본 논문에서는 IETF CAT Working Group에서 발표한 PKINIT 기반의 인증서비스를 향상시킨 Kerberos 인증 메커니즘을 제안한다. PKINIT기반의 X.509, DS/DNS를 적용하여 영역간의 서비스를 제공하는 인증과 키 교환방식으로 DNS를 통해 외부영역의 위치를 탐색하고 X.509 디렉토리 인증 시스템을 적용, 영역간 체인(CertPath)으로 DNS 서버로부터 공개키를 획득하여 다른 영역을 인증하도록 하였다. 영역간 인증과 키 교환은 Kerberos의 관용키 암호방식을 사용하고 세션 연결은 공개키 방식에 기반을 둔 X.509를 연동시키기 위하여 디렉토리서비스를 이용하였다. 이로써 Client를 확인하기 위한 임의 난수 키(K_{rand}) 생성과 이로 인한 이중 암호화 과정을 배제하였으며 통신상의 Overload를 감소시키는 효과와 인증절차의 간소화를 가지는 Kerberos 시스템을 설계하였다.

A study on Kerberos Authentication and Key Exchange based on PKINIT

Kwang-Cheul Shin[†] · Il-Yong Chung[††] · Jin-Wook Chung[†††]

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. Excluded random number (K_{rand}) generation and duplex encryption progress to confirm Client. A Design of Kerberos system that have effect and simplification of certification formality that reduce Overload on communication.

키워드 : Kerberos, PKINIT/PKCROSS, X.509, DNS

1. 서 론

분산 개방형 환경에서 네트워크 접속이 필요한 곳에서는 사용자 정보와 서버에 저장된 자원을 보호하기 위하여 사용자와 서버간의 신원증명과 안전한 비밀키 교환을 필요로 하다. 이와 같이 신원증명과 비밀키 교환이라는 필요성을 만족시키기 위하여 인증(Authentication), 무결성(Identify), 데이터 보안(Privacy) 기능을 제공하면서 키 분배 센터(KDC : Key Distribution Center)의 개념을 갖는 메커니즘이 Kerberos이다[1]. 최초 Kerberos는 서버에 접근하는 사용자와 사용자들에게 서버 자신을 인증해 주는 기능을 갖도록 중앙 집중식 인증서버를 제공하는 관용암호방식으로 개발되었다[2]. 이 메커니즘은 상호영역 인증 서비스를 지원하기 위해서는 두 영역에 공통으로 다음 두 가지의 조건이 요구된다. 첫째, 각 영역에 있는 Kerberos 서버는 비밀키를 다른 영역에 있는 서버와 서로 등록되어 공유해야 한다. 둘째, 상호간 Kerberos 서버는 신뢰를 가정하고 더욱이 두 번째 영역에 있는 서버는 반드시 첫 번째 영역에 있는 서버를 신뢰해야 한다[3]. 이러한 제약을 극복하기 위해 IETF (Internet Engineering Task Force) CAT에서 두 영역과 영역사이, 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[4]. PKINIT는 DES뿐 아니라 RSA 등 공개키 암호화와 인증서 기반구조의 Key 관리를 포함하며 Cross-Realm에 대한 인증으로 DNS (Domain Name System) 사용을 언급하고 있다[5]. IETF의

† 정 회 원 : 벽성대학 컴퓨터계열 소프트웨어 개발 조교수
†† 종신회원 : 조선대학교 컴퓨터공학부 교수
††† 종신회원 : 성균관대학교 전기전자 및 컴퓨터공학부 교수
논문접수 : 2002년 1월 21일, 심사완료 : 2002년 3월 20일

PKINIT를 이용한 Kerberos 인증 메커니즘에서는 티켓을 발급 받기 위해 임의의 난수(K_{rand}) 키로 원격 Kerberos가 지역 클라이언트를 재확인하는 과정을 통하여 티켓발급서버(Ticket Granting Server : 이하 TGS)를 접근할 수 있는 티켓과 원격 TGS가 서버에 접근할 수 있는 티켓인 서버승인 티켓(Server Granting Ticket : 이하 SGT)을 발급하는 과정의 메커니즘(9 단계)으로 구성되어 있다[6]. 본 논문의 제안은 X.509와 PKINIT에 기반을 둔 향상된 Kerberos 인증과 키 교환 메커니즘으로 영역간 검색서비스를 DNS를 통하여 수행하고 X.509를 통하여 디렉토리 인증을 정의하며 PKINIT알고리즘을 적용하여 Kerberos간 전·후방 선인증(Pre-Authentication)으로 연결하도록 설계함으로써 새로운 메커니즘(6 단계)으로 개선하였다.

2. Kerberos와 X.509 인증 프로토콜

본 절에서는 중앙집중식 관용암호시스템인 Kerberos 이론과 분산 네트워 환경에서 공개키 암호 사용을 정의한 PKCROSS/PKINIT, 그리고 PKI(Public Key Infrastructure) 관리를 위해 IETF PKIX분과의 드래프트에 의해 정의되고 있는 X.509 인증 프로토콜에 대해 고찰한다.

2.1 Kerberos 인증

Kerberos는 여러가지 요소로 구성된 복합시스템으로 Kerberos 서버와 티켓승인서버(TGS), 티켓(Ticket), 인증자(Authenticator)로 구성되어 있으며 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 한번만 사용할 수 있으며 인증정보는 클라이언트의 이름과 웍스테이션의 IP 주소, 현재의 시간을 포함하고 있다[7].

버전(Ver)4에서 KDC는 모든 사용자의 ID 및 패스워드를 보유하는 DB와 인증서버(AS : Authentication Server)를 사용하며 각 응용서버와 고유의 비밀키를 물리적으로 안전하게 분배하여 공유하도록 설계되었다. Client가 서버접근 티켓을 요청하기 위해 로그인하고 서버 V에 접속하기 위한 요구정보(ID_C, P_C, ID_V)을 전송하여 AS에 의해 인증이 되면 Ticket를 생성하여 Client에게 보냄으로써 서버에게 Client가 허가를 받았다는 사실을 확인시킨다. 문제는 패스워드가 평문으로 전송되며 서버에 접근이 필요할 때마다 티켓을 발급받기 위해 패스워드를 입력해야 하는데 이러한 문제를 해결하기 위하여 AS와 함께 TGS를 사용한다. 사용자는 티켓을 보관하여 서비스에 접속할 때마다 이 티켓을 이용하여 TGS에게 접속한다. AS는 자신의 DB에 저장되어 있는 Client의 패스워드로 암호키(K_C)를 생성하여 티켓을 발급한다. 패스워드의 입력 시기는 Ticket이 도착한 후에 자신의 패스워드를 입력하여 키를 생성하고 티켓을 복호화한다. 또한 티켓의 가로채기를 봉쇄하기 위해 티켓이 발행된 시간과 유효시간을 포함하고 있다. 문제는 티켓-승인 티켓과 관련된 유효시간으로 네트웍 서비스(TGS 또는 응용서버)는 티켓을 사용하고 있는 사람이 티켓이 발행된 사람과 같다는 것을 증명할 수 있어야 한다. AS가 C와 TGS간, C와 서버 V간에 세션키($K_{C,TGS}$, $K_{C,V}$)를 제공하여 신원을 확인시켜 주고 유효시간(Lifetime)을 짧게 인증자(Authenticator$_C$)에게 두어 가로채기의 위험을 방지하고 있다.

Ver 5(그림 1)에서 Ver 4와 다른 점은 realm(영역), Options(플래그 값), Times(티켓에 시간설정), Nonce(Replay를 방지)가 추가되었고 Subkey와 Seq#는 실제 통신상에서 별도의 세션키로 사용하고 메시지에 순서를 부여하여 재전송이 아님을 보증하도록 한다. 또한 Ticket에 대한 이중암호화가 제거되었고 임의 암호형식을 사용할 수 있으며 여러 영역간에 상호동작을 가능하도록 확장성을 더욱 용이하게 개선되었다[8].

2.2 PKCROSS/PKINIT 메커니즘

RFC2026의 Internet Draft는 공개키 암호를 이용하여 상호 영역간 인증할 수 있는 PKCROSS/PKINIT Kerberos 프로

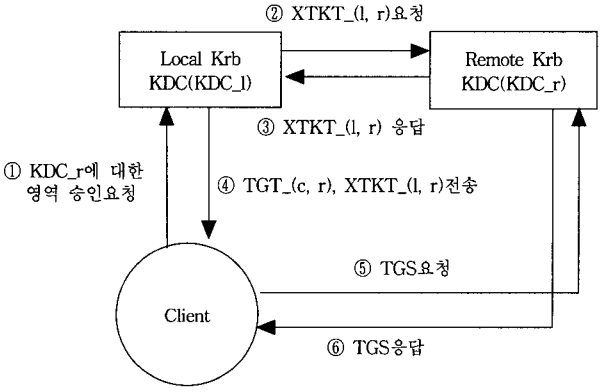
Version	Client	Kerberos		Server	표 기
		AS	TGS		
V5	① ● --------> ● Options, Realm$_C$, ID_C, ID_{TGS}, Times, Nonce1, AD_C ② ● <-------- ● Realm$_C$, ID_C, **Ticket$_{TGS}$**, E_{Kc}[$K_{C,TGS}$, nonce1, Flag, Times1, Realm$_{TGS}$, ID_{TGS}, AD_C] ③ ● ----------> ● options, ID_V, Times2, nonce2, Ticket$_{TGS}$, Authenticator$_C$ ④ ● <---------- ● Realm$_C$, ID_C, Ticket$_V$, $E_{KC,TGS}$[$K_{C,V}$, nonce2, Flag, Times2, Realm$_V$, ID_V, AD_C] ⑤ ● --- options, Ticket$_V$, Authenticator$_C$ ------> ● ⑥ ● <-------- $E_{KC,V}$[TS2, Subkey, Seq#] ------ ●				**Ticket$_{TGS}$** = E_{KTGS}[Flag, $K_{C,TGS}$, Realm$_C$, ID_C, Times1, AD_C] Ticket$_V$ = E_{KV}[Flag, $K_{C,V}$, Realm$_C$, ID_C, Times2, AD_C] Authenticator$_C$ = $E_{KC,V}$[ID_C, Realm$_C$, TS2, Subkey, seq#]

(그림 1) 커버러스 V5

토콜을 정의하고 있다[9].

Kerberos는 TLS(Transport Layer Security)와 함께 대칭, 비대칭 암호를 모두 사용하며 초기인증 교환과정에서 공개키 암호의 사용을 정의하는 PKINIT와 응용 서비스가 공개키 암호를 이용하여 인증 한 후 Kerberos 티켓을 어떻게 발급할 것인지를 PKTAPP (Public Key Utilizing Tickets for Application Servers)에서 기술하고 있다[10]. 또한 상호영역에 대한 키(Key)들을 유지하는데 관리적 부담을 간소화하기 위해 PKI[X.509]를 이용하도록 기술한 PKCROSS 정의 명세서가 있다.

Kerberos에서 Local Client가 Remote Server와 상호영역 인증을 위하여 공개키 암호를 사용하여 티켓을 요청하고 티켓승인 티켓(TGT : Ticket Granting Ticket)을 발급 받는 과정인 PKCROSS(연결)/PKINIT(인증)메커니즘은 (그림 2)와 같다.



(그림 2) PKCROSS/PKINIT 프로토콜

① 영역 서비스를 위한 승인요청 : Client는 Remote Realm에 대한 티켓을 KDC_l(Local KDC)에 요청한다.

② 영역 사용을 위한 티켓-승인 티켓 요청(PA-PK-S-EQ) : KDC_l은 적법한 PKCROSS 티켓인(XTKT_l, r) 요구하기 위하여 KDC_r(Remote KDC)에 PKINIT를 요구한다.

즉, KDC_l은 Cross-Realm의 인증을 위해 유효한 PKCROSS 티켓을 보유하는지를 캐쉬(Cache)를 통해 XTKT_(l, r)를 확인한다. 만약 합법적인 XTKT_(l, r)를 가지고 있지 않을 경우에는 PKINIT를 이용하여 Cross-Realm Key를 설치하고 XTKT_(l, r)를 KDC_r에게 요청한다. 여기에서 XTKT_(l, r)는 원격 KDC가 Local KDC에 발행하는 Ticket이다.

③ 원격 TGS용 Ticket 승인(PA-PK-AS-REP) : KDC_r은 티켓에 TicketExtension(Kerberos명세서 정보들)을 기록하여 PKINIT/PKCORSS로 응답한다. TicketExtension은 Client에 대하여 KDC_l에 의해서 발생된 Cross-Realm 티켓들의 Lifetime과 같은 정책들을 포함하고 있다. KDC_l은 Client에 신원증명서 정보를 반영해야 하며, 이 티켓은 KDC_r이 KDC_l을 신뢰하도록 입증하기 위하여 XTKT_(l, r)을 호출한다.

④ 승인서 발급(PA-PK-AS-SIGN) : KDC_r으로부터 PKINIT/PKCROSS에의해 수취된 자료를 KDC_l은 Client에게 전송한다. KDC_l은 Client에 티켓(TGT_(c, r) : Local KDC가 Client에 발급하는 티켓승인티켓)을 전송한다. 이 티켓은 티켓 XTKT_(l, r)을 TicketExtension 기록란에 포함되며, XTKT_(l, r)은 Cross Realm Key를 포함하고 있다.

또한 TGT_(c, r) 티켓은 XTKT_(l, r)에 서명된 키를 가지고 암호화되어 있다. KDC_l은 KDC_r을 위해 Hostname, IP주소를 표시한 다른 TicketExtension을 포함할 수도 있다.

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

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

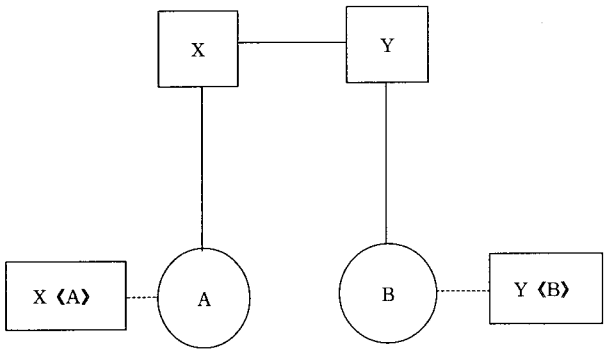
2.3 X.509 디렉토리 인증 프로토콜

X.509는 디렉토리 서비스를 구성하는 X.500계열의 일부로 사용자들에 대한 정보를 보유하는 일종의 서버인 X.500 디렉토리를 통해서 제공되는 인증서비스의 구성을 명시하고 있으며 PKI(IETF에서의 PKIX)구조를 제공한다. X.500 디렉토리는 DIT(Directory Information Tree)로 구성되며 각 엔트리는 유일한 이름(DN : Distinguished Name)을 가진다. Root를 정점으로 각 국가를 나타내는 RDN(: Relative Distinguished Name)이 할당되며 각 국가는 하부 엔트리로 조직의 유일한 RDN을 부여한다. X.509 인증서는 인증서 취소목록(CRL : Certificate Revocation List)과 인증서

(그림 3) X.509 인증서

를 정의하는 다양한 환경에 맞는 조건과 서명 알고리즘의 선택이 가능하도록 확장영역을 가진다. 본 논문에서 디렉토리 인증 프로토콜인 X.509를 이용하며 연결, 외부 영역에 있는 서비스를 얻는다. 디렉토리 서버는 클라이언트들에게 인증서를 획득하는데 쉽게 접근할 수 있는 경로만을 제공한다. X.509 인증서의 구성은 (그림 3)과 같다[11].

Client가 인증서를 획득하는 과정으로 다음 (그림 4)와 같이 클라이언트 A, B가 있고 인증기관 X, Y가 있다고 가정한다. X.509에서 공개키를 획득하기 위한 인증서의 연결(chain)은 표기로 인증기관 CA(Certification Authority)인 X에 의해 발행된 클라이언트 A의 인증서(X ≪A≫)이며 X ≪A≫ = X{V, SN, AI, CA, TA, A, AP}로 정의된다. A가 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 4) X.509 계층구조

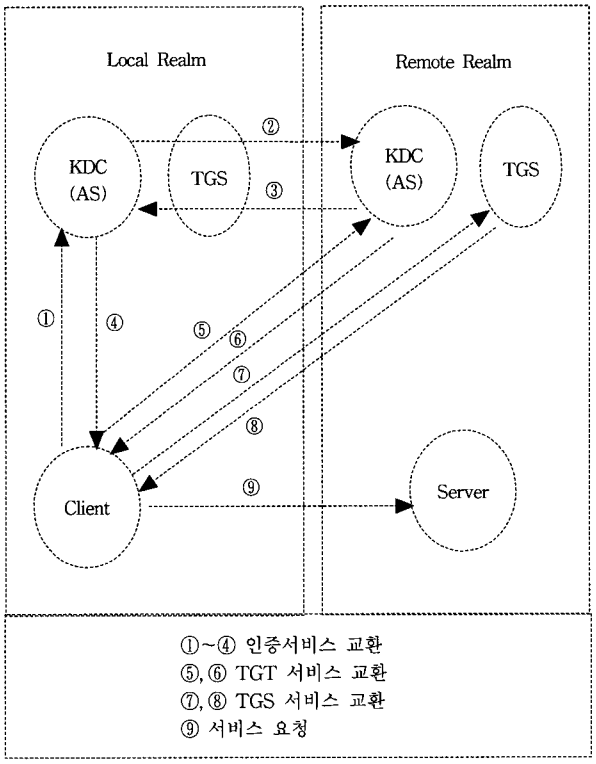
이러한 체인은 N개 요소로 구성된 X≪Y≫Y≪Z≫…X_N ≪B≫체인을 형성할 수 있다. 이 경우에, (Xi, Xi + 1)연결에 있는 CA의 각 쌍은 서로가 인증서를 가지고 있어야 한다. X.509는 진행과정이 직선적으로 이루어지도록 CA를 중심으로 계층적으로 정렬하도록 제시하고 있다. 인증기관 X에 대한 디렉토리 엔트리는 2가지 타입의 인증서를 포함하고 있는데 다른 CA에 의하여 생성된 클라이언트 인증서는 CA 이외의 누구도 검출 및 수정할 수 없다는 특성을 가지고 있다. 그리고 인증서는 위조할 수 없기 때문에 인증서를 보호하기 위한 다른 조치 없이 디렉토리에 저장할 수 있다. 디렉토리 서버는 DNS와 X.500을 기반으로 Domain을 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), 그리고 디렉토리내의 Domain 구조와 구성정보를 저장하는 Configuration(구성 파티션), Domain 자체의 개체정보가 저장되는 Domain partition으로 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. 새로운 Kerberos 인증 및 키교환 메커니즘

3.1 개 요

본 논문에서 제시한 인증 메커니즘은 영역과 영역간 서비스를 위한 인증으로 기본 환경은 KDC(AS)와 티켓을 발행하는 TGS, 영역 내 객체의 위치를 제공하고 DNS를 사용하기 위한 디렉토리 서버(DS), 비밀키의 생성 및 분배를 위한 키 관리센터(KMC), End User의 ID와 패스워드를 저장한 중앙DB, 자원서비스를 위한 서버, 서버를 사용할 Client가 하나의 도메인(Domain)으로 구성되어 있다. 인증서는 인증기관(CA)이 전자서명을 통하여 전자서명 공개키와 이를 소유하는 자연인 또는 법인과의 귀속관계를 확인, 증명하는 전자적 정보로써 본 논문에서는 Kerberos의 KDC(AS)가 인증기관과 키분배 센터의 역할을 담당하며 상호 신뢰성을 확인하는데 사용한다. 각 영역의 Client와 자원(Server)들은 KDC(AS)에게 ID와 패스워드를 Install때 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 티켓발급 서버(TGS)로부터 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. Kerberos의 보안정도는 티켓발급서버가 얼마나 확실하게 보호되는지에 달려 있으며 Kerberos를 사용하는 서버와 클라이언트들은 티켓발급서버를 신뢰함을 전제로 한다. 본 논문에서는 IETF Draft에서 제시한 PKINIT 기반의 Kerberos 인증 프로토콜에 인증정보(SignedAuthPack, TrustedCertifiers)를 추가하여 영역간 새로운 Kerberos 인증 프로토콜을 제시한다.

(그림 5)는 IETF의 영역간 Kerberos 인증절차로써 상호간 인증서와 암호알고리즘 등 인증서비스를 교환(①~④)하고 원격 TGS 접근을 위한 티켓과 세션키를 교환하는 TGT 서비스 교환(⑤~⑥), 서버 접근용 티켓과 세션키를 교환하는 TGS 교환(⑦~⑧), 세션키에 의한 서비스 요청과정(⑨)으로



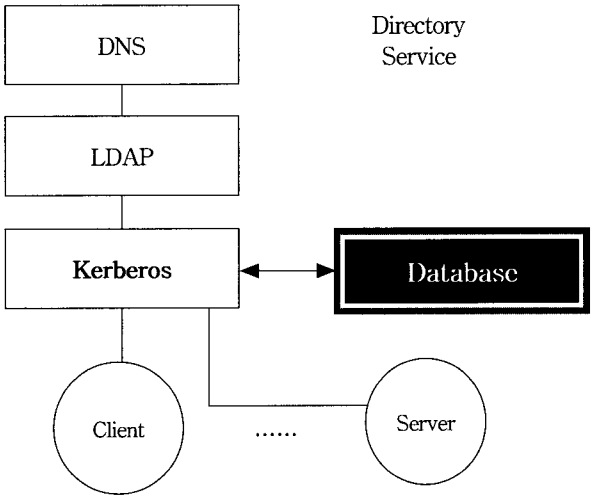
(그림 5) IETF의 인증 메커니즘

로 원격 KDC가 지역 KDC의 정보를 인증한 후 티켓을 발급하고 있다. 제안되는 수정된 Kerberos 인증과 키 교환 메커니즘은 원격 Kerberos(KDC_r)가 인증과 동시에 티켓을 Client에게 발급함으로써 메시지 교환의 단축으로 처리모듈과 통신부담이 줄어들게 되었으며 임시적인 난수 키 값(K_{rand})에 의한 공통키의 생성과 암호화 과정을 수행하지 않는다. KDC_r은 KDC_l에 인증결과(KDC_Cert)만을 되돌려주고 Client에게로 직접 티켓을 할당하는 메커니즘이다.

3.2 원격영역 Kerberos 인증 프로토콜

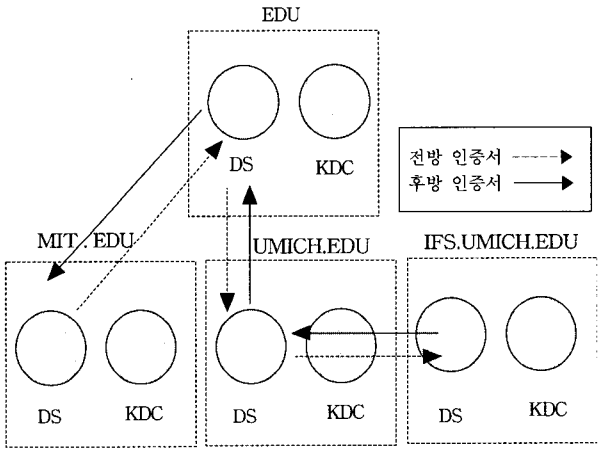
3.2.1 DNS와 디렉토리 서비스

모든 Kerberos의 공개키는 DNS 서버로부터 획득하게 된다. 저장되는 KDC의 공개키는 DNS 서버에 의해 전자서명되어 무결성과 데이터의 인증을 보장받는다. 이 공개키 인증서는 PKINIT에 의한 초기 인증을 목적으로 원격 KDC의 공개키를 획득하기 위해 디렉토리서버(서비스)를 이용한다. 디렉토리 서비스는 데이터베이스, 파일, 호스트 연결, 사용자 서비스 등 모든 자원에 대한 관리를 허용하고 위치 서비스로써 인터넷 DNS를 사용하여 여러 Domain을 트리(Tree)구조로 연결시킨다. 지역 Kerberos는 지역 클라이언트가 요청한 영역이 동일영역이 아닐 경우에는 DNS를 사용하여 외부 영역의 경로를 찾는다. 본 논문에서 인증과 키 교환을 위한 디렉토리 서비스의 구조는 (그림 6)과 같다. 구성은 Domain 이름과 위치 서비스, 확장성, 표준을 제공하며 관리구조가 용이한 DNS와 디렉토리 서비스에 접근하기 위한 인터넷 표준(RFC1777) 프로토콜인 LDAP, 인증과 키 교환을 실현 할 Kerberos, 객체(Object)들의 식별자와 키, 티켓의 유효시간, 버전 등을 속성으로 보유하는 Kerberos Database로 되어있다.



(그림 6) 디렉토리 서비스의 구조

(그림 7)은 디렉토리를 인증하기 위해 디렉토리 서버(Dictory Server)를 이용하여 외부영역에 있는 목적지까지 경로를 연결하는 세션과정을 도식한 것이다. 여기에서 디렉토리 서버가 세션연결을 위한 체인(Chain)을 설정하고 Kerberos (KDC)는 인증과 키 교환 절차를 실행한다.



(그림 7) 디렉토리 서버인증

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.

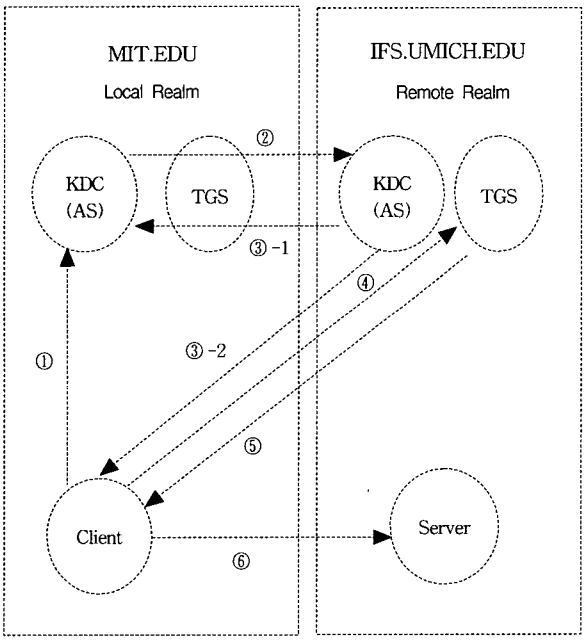
MIT.EDU와 EDU연결, EDU영역과 UMICH.EDU영역 연결, 그리고 UMICH.EDU영역과 IFS.UMICH.EDU영역 연결한다. MIT.EDU영역의 클라이언트는 IFS.UMICH.EDU영역에 있는 서비스를 사용하기 위한 전방 인증서(Forward Certificate)와 후방 인증서(Reverse Certificate)는 다음과 같다.

전방 인증서 : EDU≪UMICH.EDU≫UMICH.EDU
≪IFS.UMICH.EDU≫)
후방 인증서 : UMICH.EDU≪EDU≫EDU≪MIT.EDU≫)

클라이언트가 있는 영역인 MIT.EDU영역과 IFS.UMICH.EDU간 연결이 직접적으로 이루어졌으므로 상호영역간에 있어서 침해자가 서비스를 요청한 클라이언트처럼 가장하여 서비스를 가로채거나 변경시킬 수 있는 요소를 배제하기 위하여 클라이언트를 인증하는 절차를 필요로 하게 된다. 클라이언트는 원격 Kerberos에게 X.509를 이용하여 획득한 원격 영역의 공개키로 정보를 암호화하여 전송함으로써 클라이언트와 원격 영역간의 통신을 방해하는 침입자로부터 보호할 수 있게 한다.

3.2.2 알고리즘

Client가 요청한 서비스가 동일한 영역 내에 있는 서비스이면 KDC(AS)의 데이터베이스에서 Client의 정보로 인증을 하게 되고 요청한 서비스가 동일 영역 내에 존재하지 않으면 KDC(AS)는 Client가 요청한 영역을 디렉토리 서버를 통하여 DNS에게 검색을 의뢰한다. DNS 서버는 정방향 조회와 캐쉬 루트서버를 이용하여 리졸빙 하고 캐쉬영역에 저장되며 KDC(AS)로부터 의뢰를 받은 영역을 검색한 후



(그림 8) 수정된 Kerberos 메커니즘

에 이웃(Preauthentication)하는 영역을 디렉토리 서버에 전송한다. 클라이언트는 원격 Kerberos에 X.509를 이용하여 획득한 원격 영역의 공개키로 정보를 암호화하여 전송함으로써 클라이언트와 원격 영역간의 통신을 방해하는 제3자로부터 보호할 수 있게 한다.

X.509 프로토콜을 사용하여 전·후방 체인으로 원격영역의 공개키를 획득함으로써 비밀키를 교환하는 번거로움을 배제한 PKINIT기반의 Kerberos 서버 상호지원 메커니즘이다.

(그림 8)은 서로 다른 영역의 Local Realm(MIT.EDU)과 Remote Realm(IFS.UMICH.EDU)의 환경으로 KDC는 TGS가 사용할 티켓($Ticket_{TGSREM}$)만을 발급하는 역할을 하며 티켓에는 발급자, 세션키, 발급대상의 ID와 주소, 발행시간, Reply 방지용 값을 포함한다. TGS는 서버승인 티켓인 $Ticket_{SGTREM}$를 발급하는 서비스를 담당한다. Local Client가 Remote Server의 서비스를 받기 위한 메시지 교환 내용은 다음과 같다.

3.2.2.1 인증 서비스

① ID_C, $Realm_S$

② E_{KDC_rPK}[SignedAuthPack, TrustedCertifiers, KDC_l ≪C≫, CertPath]

③-1 E_{KDC_lPK}[KDC_Cert]

메시지 ①에서 Client는 자신의 ID와 서비스를 원하는 영역을 자신의 영역에 있는 Local KDC에게 보내 서비스를 요청한 Client가 정당한 사용자인지를 Database에서 검색하여 유효성과 적법성을 검토하도록 한다. 서비스 영역이 다를 경우 DNS를 통하여 검색을 의뢰하고 해당영역의 디렉토리 서버는 DNS로부터 받은 서비스 영역(Remote 영역)에 관하여 상호인증을 하기위한 전·후방 인증 체인을 생성하여 Remote 영역의 공개키를 획득(PKINIT)한다. 이 공개키로 메시지 ②에서 Local KDC는 Client와 자신의 정보(SignedAuthPack, TrustedCertifiers), 그리고 KDC_l≪C≫, CertPath를 전송하여 신원확인 및 티켓승인 티켓을 요청한다. 이 메시지에는 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_Cert로 KDC_l에게 전송(③-1)하여 인증 받았음을 확인한다.

● **주요 요소에 대한 이론적 해석**

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 값을 가진다.

PkAuthenticator : 지역영역의 KDC(AS)의 정보로 cusec, ctime, nonce, PaChecksum으로 구성된다.

---

cusec : Client의 인증자를 발행한 시간을 알린다.
ctime : KDC_1의 인증자를 발행한 시간을 알린다.
Nonce : Reply가 아니라는 정당한 데이터의 무결성을 보장한다.
PaChecksum : ASN.1에서 정의한 암호화 알고리즘의 종류로 cksumtype과 checksum을 포함한다.

---

cksumtype : 암호 알고리즘 유형을 선택하는 정수 값
checksum : 암호 알고리즘으로 crc32, rsa-md4, rsa-md4des, rsa-md5, rsa-md5des,des-mac이다.

ClientPublicValue : Client의 공개정보로 SigAuth-Pack과 User-Type를 갖는다.
SigAuth-Pack : 암호Algorithm과 Parameter 값을 갖는다.
User-Type : 인증서 형식으로 X.509v3, PGP, PKIX를 갖는다.

TrustedCertifiers : KDC_1의 인증서로 principalName, caName, issuerAndSerial, UserCert 값을 갖는다.
principalName : KerberosName
caName : X.500, X.509 검증을 거친 Name
issuerAndSerial : Client 및 KDCs가 신뢰할 수 있는 CA의 번호
UserCert : Client의 RSA암호화 증명서
KDC_1≪C≫ : KDC_1이 Client에게 발행하는 X.509인증서
CertPath : DNS를 통하여 세션이 설정된 주소를 갖는 인증서 체인 값이다.
$KDC_{\_PK}$: Local KDC의 공개키
KDC_Cert : KDC_1을 KDC_r이 정당한 사용자로 인증한 부분

3.2.2.2 TGT 서비스

③-2 E_{PKC}[$Ticket_{TGSREM}$, $K_{C,TGSREM}$, TimeStamp, Nonce, $Realm_{TGSREM}$, E_{KDC_rSK}[$Ticket_{TGSREM}$, TimeStamp, PaChecksum, Nonce, $Realm_{TGSREM}$]]

$Ticket_{TGSREM}$ = $E_{KTGSREM}$[flags, $K_{C,TGSREM}$, ID_C, AD_C, TimeStamp, Nonce]

메시지 (③-2)에서 KDC_r은 KDC_1≪C≫로 Client를 인증하고 KDC_r 영역의 TGS_{REM}에게 Client의 인증을 확인시키기 위해 공유키(E_{SK})로 티켓과, 티켓발행시간, 암호알고리즘, 임의의 수, TGS_{REM}의 영역을 암호화하고 세션키($K_{C,TGSREM}$)와 티켓($Ticket_{TGSREM}$)을 KDC_1≪C≫로부터 추출한 Client의 공개키로 전송한다. 이때 KDC_r은 임의의 수(Nonce)와 $Ticket_{TGSREM}$을 자신 영역의 TGS_{REM}에게 전송함으로써 Client로부터 전송된 내용과 비교하도록 하여 허가된 자라는 것을 확신시킨다.

- **주요 요소에 대한 이론적 해석**

PK_C : Client C의 공개키
TGS_{rem} : 원격 TGS 영역

$Ticket_{TGSREM}$ = $E_{KTGSREM}$[flags, $K_{C,TGSREM}$, ID_C, AD_C, TimeStamp, Nonce]

$Ticket_{TGSREM}$: 원격 TGS 사용권한을 가진 티켓으로 KDC_r에 의해 인증되었음을 보장
K_{TGSREM} : 원격 TGS의 비밀키로 KDC_1과 TGS만이 아는 키로 암호화
flags : 티켓의 옵션상태
$K_{C,TGSREM}$: Client와 원격 TGS와의 세션 키
AD_C : Client의 Address로 초기에 티켓을 요구한 Client외의 사용을 예방한다.
TimeStamp : 티켓이 생성된 시간을 알린다.
Nonce : Replay 방지를 위한 임의의 수
$Realm_{TGSREM}$: 원격 TGS의 영역
K_{DC_rSK} : 원격 KDC와 TGS의 공유키(E_{SK})
PaChecksum : 암호알고리즘 유형

3.2.2.3 SGT 서비스

④ $E_{KC,TGSREM}$[ID_S, A_C, $Ticket_{TGSREM}$, E_{KDC_rSK}[$Ticket_{TGSREM}$, TimeStamp, PaChecksum, Nonce, $Realm_{TGSREM}$]]

⑤ $E_{KC,TGSREM}$[$K_{C,SGTREM}$, $Ticket_{SGTREM}$, TimeStamp, Nonce, $Realm_{SGTREM}$, ID_S, EK_{SGTREM}[$K_{C,SGTREM}$, ID_C, AD_C, ID_S, 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 보유하게 됨으로써 TGS_{REM}에 접근할 준비가 된다. 메시지 ④에서 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. 재 사용할 수 있는 티켓과는 다르게 인증자는 한번만 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 뿐이라는 것을 알 수 있다. 원격 TGS_{REM}은 Client로부터 전송된 인증자와 티켓의 정보와 비교하여 일치하면 티켓을 보낸 사람은 실제 티켓의 소유자라고 인증할 수 있다. 메시지 ⑤는 서버를 사용할 수 있는 티켓($Ticket_{SGTREM}$)과 세션키($K_{C,SGTREM}$)를 생성하며 원격 TGS_{REM}은 서버의 비밀키(K_{SGTREM})로 티켓과, 유효시간, 임의의 수, 영역을 암호화하고 Client와 TGS_{REM}간의 세션키로 다시 암호화하여 Client에게 전송한다. Client는 서버의 비밀키(K_{SGTREM})로 된 내용을 확인할 수 없다.

● 주요 요소에 대한 이론적 해석

$K_{C,TGSREM}$: Client와 원격 TGS_{REM}와의 세션키로 변조의 방지를 위한 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 티켓으로 KDC_r에 의해 인증되었음을 보장
K_{DC_rSK} : 원격 KDC와 TGS_{REM}의 공유키로KDC_r로부터 전송되었음을 TGS_{REM}이 확인
Nonce : Replay 방지용 값
$Realm_{TGSREM}$: 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} : 서버 S의 비밀키로 변조 방지를 위해 TGS_{REM}과 서버만이 알고 있는 비밀키
AD_C : Client의 Address로 티켓의 내용과 동일해야 한다
$Ticket_{SGTREM} = E_{KSGTREM}[flags, K_{C,SGTREM}, Realm_{SGTREM}, ID_C, AD_C, TimeStamp, Nonce]$

3.2.2.4 서비스 요청

⑥ $E_{KC,SGTREM}[Ticket_{SGTREM}, A_C, E_{KSGTREM}[K_{C,SGTREM}, ID_C, AD_C, ID_S, Nonce]]$

$A_C = E_{KC,TGSREM}[ID_C, AD_C, Realm_{TGSREM}, TimeStamp, Nonce]$

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

Client는 원격서버를 사용할 수 있는 권한을 가진다. 메시지 ⑥에서 Client는 서버 접근권한을 확인시키고 서비스를 위한 요청으로 서버용 티켓($Ticket_{SGTREM}$)과 인증자, 원격 TGS로부터 전송된 내용을 보냄으로써 서버로 하여금 인증자와 TGS로부터 전송된 값(ID_C, AD_C, $Realm_{TGSREM}$, Time Stamp, Nonce)을 비교하여 인증하고 TGS_{REM}이 생성하여 비밀리에 분배한 세션키($K_{C,SGTREM}$)로 송수신할 수 있다.

● 주요 요소에 대한 이론적 해석

$K_{C,SGTREM}$: Client와 원격 서버 S간의 세션키로 이 메시지가 Client로부터 온 것임을 서버에게 보증
$Ticket_{SGTREM}$: 서버 S에 접근권한을 가진 티켓으로 TGS_{REM}으로부터 인증을 보장
A_C : Client가 생성한 인증자로 인증정보가 수록되어 $Ticket_{SGTREM}$과 비교하여 인증
K_{SGTREM} : 원격 서버 S의 비밀키로 TGS_{REM}이 전송한 것을 확인
Nonce : 메시지에 대한 무결성 확인용
$A_C = E_{KC,TGS}[ID_C, AD_C, Realm_{TGS}, TimeStamp, Nonce]$

4. 메커니즘 분석 및 효과

IETF의 Working Group에서 사용하고 있는 RFC1510의 Kerberos 메커니즘은 PKINIT의 기반의 PKIX(Public Key Infrastructure)로 공개키와 공통키를 사용하여 인증정보에 대한 무결성을 보장하고 있으나 Domain간 연결정보에 대해서는 디렉토리 서버와 DNS에 대한 언급만 했을 뿐 구체적인 사용방법에 대해서는 기술되지 않고 있다. 본 논문에서 제시된 알고리즘은 IETF Working Group에서 사용하고 있는 PKINIT/PKCROSS 메커니즘을 기반으로 하였으며 Kerberos(KDC) 비밀키에 의한 인증과 X.509에서 보장해주는 안전성, 그리고 DS/DNS에 의한 세션경로를 보관하여 사용할 수 있는 인증서 체인(CertPath : Domain Value)에 의한 메커니즘 구성으로 원격 Kerberos에서 Client로 티켓 승인티켓(TGT)을 직접 전송할 수 있는 메커니즘이다. 즉 Client는 KDC_r에 TGT를 획득하기 위한 별도의 요청신호를 필요로 하지 않는다. 또한 KDC_r에서 Client의 상호인증을 위해 생성한 임시난수 키(K_{rand})값을 배제한 인증서(KDC_l≪C≫)를 사용함으로써 Client와 원격 TGS에서 암.복호화하는 과정이 생략되었으며 이로 인해 이중 암호화와 통신부담이 감소되었다<표 1>. 서버용 티켓은 TGS의 키로 암호화(K_{TGSREM})되어 있으므로 변조가 불가능할 뿐만 아니라 Client의 공개키로 재 암호화하므로 제 3자가 티켓을 이

〈표 1〉 메커니즘 분석

메커니즘	분 석		효 과
	IETF CAT Working Group	제안 메커니즘	
안 전 성	• PKINIT를 사용한 공개키 획득으로 Dictionary 공격에 취약	• PKINIT와 연계된 Directory Server의 DNS사용(서명 키)	• DNS서버가 서명한 공개키 등록 과 분배로 안전성 보장
Client 정보	• S_{KC}[SigAuth-Pack] • SigAuth-Pack : AI, P_{KC}	• ClientPublicValue, KDC_l 《C》	• 알고리즘식별자, 파라미터, 공개키 정보를 X.509에 의한 효율적 인 관리
Client 확인과정	• P_{KC}와 키 값(K_{rand})를 전송하여 확인	• CertPath, KDC_l 《C》	• 경로값을 보유하는 인증서 체인으로 키 생성과 이중암호 배제
전송단계	• 9단계	• 6단계	• Kerberos의 관용키와 공개키의 사용으로 동일한 인증효과, 암호 화 과정과 통신부담 감소
티켓 발급절차	• KDC_r → KDC_l → Client	• KDC_r → Client	• 절차의 간소화
주요 인증인자	• KDC_{LOC} 《C》, K_{rand}	• TrustedCertifiers, KDC_l 《C》	• 인증인자의 단순화

용할 수 없다. 티켓 내에도 Client와 TGS_{REM}, Client와 서버 사이의 세션키($K_{C,TGSREM}$, $K_{C,SGTREM}$)를 포함시킴으로써 티켓 소유자가 정당한 사용자임을 증명한다.

본 논문에서 제시된 알고리즘은 원거리 통신에서의 보안성을 보장하기 위해서 인증정보를 전달할 때 Kerberos의 비밀키와 PKINIT를 이용한 공개키를 사용하였고 상호인증을 위해 X.509와 DNS를 이용한 체인(Directory System)방식으로 Domain간(Kerberos to Kerberos) 원거리 통신을 보다 더 안전성이 보장되는 Kerberos 시스템을 설계하였을 뿐만 아니라 Client가 Remote TGS에 서버용 티켓을 요청할 때 Remote KDC(AS)의 재확인 과정을 생략함으로 인한 임의의 공통키 생성을 제거하였고 이로 인한 통신 복잡도가 감소되었다.

5. 결 론

인터넷의 전자상거래, 응용서비스 등의 사용자 증가와 더불어 보안의 중요성이 확산됨에 따라 상호인증 기술이 정보보호 기반기술의 중요요소로 대두되고 있다. 인증 메커니즘으로 관용 암호방식을 사용하는 Kerberos는 동일영역에서의 상호인증 알고리즘으로 손색이 없는 최적의 메커니즘을 갖는다. 분산 네트웍환경에서 통신하고자 하는 다수의 웍스테이션들과 응용서버의 인증을 위해서 Kerberos는 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, DS/DNS를 적용하여 영역간의 인증과 서비스를 제공하는 인증과 키 교환방식을 제안하였다. DNS를 통해 외부영역의 위치를 탐색하여 경로(CertPath)를 저장하고 X.509 디렉토리 인증 시스템인 디렉토리 서버를 적용, 영역간 체인을 통하여 공개키를 획득하여 다른 영역을 인증하도록 하였다. 이로써 Client를 확인하기 위한 임의 난수 키(K_{rand}) 생성과 이로 인한 이중 암호화 과정을 배제하였으며 Local Kerberos를 경유하지 않고 Client로 직접 티켓을 전송함으로써 통신상의 Overload를 감소시키는 효과와 인증 절차의 간소화를 가지는 Kerberos 시스템을 설계하였다. 향후 연구해야할 과제로 Naming Service기능을 담당하는 DNS서버에 대한 보안 메커니즘의 연구와 인증된 공개키를 분배, 요청 및 응답에 대한 메시지의 무결성을 제공하는 기능이 필요하다.

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] 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-14.txt.

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

[6] RFC 1510, Public Key Cryptography for Initial Authentication in Kerberos, draft-ietf-cat-kerberos-pk-init-09.txt, IETF, 1999.

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

[8] 김해영, "일회성 티켓 사용자에게 동기화 된 시계를 요구하지 않는 Kerberos", 한국과학기술원 석사학위논문, 1998.

[9] 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.

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

[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] http : //www.kr.freebsd.org/doc/PoweredByDNS/resolving.html.

신 광 철

e-mail : kcshin@mail.byuksung.ac.kr

1985년~1988년 제도분석 및 프로그램관 (중앙전산소)

1991년~1995년 전쟁연습 프로그램관 및 전산실장(육군대학)

1995년~1999년 성균관 대학원 정보공학과 수료

1989년~1990년 국방대학원 전자계산과(과학석사) 졸업

1996년~현재 벽성대학 소프트웨어개발전공 조교수

관심분야 : 정보보호기술, 객체지향 분석/설계, 전자상거래응용, Visual Programming,

정 일 용

e-mail : iyc@mina.chosun.ac.kr

1987년~1991년 City University of New York in U.S.A 전산학박사

1979년~1983년 한양대학교 공과대학 공학사

1999년~2000년 정보전산원장

1997년~1999년 정보과학대학 학장보

1991년~1994년 한국전자통신연구소 선임연구원

현재 조선대학교 전자정보공과대학 컴퓨터공학부 교수

관심분야 : 네트워크 보안, 전자상거래, 분산시스템 관리, 코딩 이론, 병렬 알고리즘

정 진 욱

e-mail : jwchung@songgang.skku.ac.kr

1973년~1985년 한국과학기술연구소 실장

1991년 서울대학교 대학원 계산통계학과 (이학 박사)

1998년~1999년 성균관대학교 정보통신대학원장

현재 성균관대학교 전기전자 및 컴퓨터공학부 교수

관심분야 : 컴퓨터 네트워크, 네트워크 관리, 네트워크 보안