韓國 컴퓨터情報學會 論文誌
第11卷 第3號, 2006. 7.

2006-11-3-1-7

인증서기반의 Multi_Kerberos 인증시스템에 관한 연구

신 광 철*, 조 성 제*

A Study on Multi_Kerberos Authentication Mechanism based on Certificate

Shin Kwang Cheul*, Cho Sung Je*

요 약

본 논문에서는 IETF CAT Working Group에서 발표한 PKINIT기반의 인증서비스를 향상시킨 Multi_Kerberos 인증 메커니즘을 제안한다. PKINIT기반의 X.509, DS/DNS를 적용하여 영역간의 서비스를 제공하는 인증과 키 교환방식으로 DNS를 통해 외부영역의 위치를 탐색하고 X.509 디렉터리 인증 시스템을 적용, 영역간 체인(CertPath)으로 DNS 서버로부터 공개키를 획득하여 다른 영역을 인증하도록 하였다. 검증서버를 활용하여 인증서 경로생성 및 구축, 세션키의 복구, 인증서 기반의 키 관리를 포함한 상호영역(cross realm)에 대한 효율적인 인증서비스를 지원하는 메커니즘을 제안하였다. 이에 통신상의 절차를 감소시키는 효과와 인증절차의 간소화를 가지는 Multi_Kerberos 시스템을 설계하였다.

Abstract

In this paper, proposes Multi_Kerberos certification mechanism that improve certification service of based on PKINIT that made public in IETF CAT Working Group. This paper proposed to a certificate other realm because search position of outside realm through DNS and apply X.509 directory certification system, to get 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 based on PKINIT, in order to provide regional services. This paper proposed mechanism that support efficient certification service about cross realm including key management, the path generation and construction of Certificate using Validation Server, and recovery of Session Key. A Design of Multi_Kerberos system that have effects simplify of certification formality that reduce procedures on communication.

▸ Keyword : 케르베르스(Kerberos), 인증(Authentication), 인증서(X.509), 검증서버(Validation Server), 초기인증(PKINIT), 키 복구(Key Recovery)

• 제1저자 : 신광철, 교신저자 : 조성제
• 접수일 : 2006.05.10, 심사일 : 2006.05.13, 심사완료일 : 2006.07.15
* 성결대학교 e-비즈니스 IT학부 교수

Ⅰ. 서 론

분산 환경에서 자원보호를 위해서는 사용자와 서버 간 신원증명과 안전한 비밀 키 교환이 요구된다. 이를 만족시키기 위하여 인증, 무결성, 데이터 기밀성이 필요하다. 이러한 환경에서 대표적인 인증 메커니즘으로 Kerberos와 Yaksha 인증방식이 있으며 Kerberos 메커니즘을 응용하여 다른 시스템과 호환성을 갖도록 확장시킨 SESAME이 있다[1][2][3]. Kerberos는 통신망 인증시스템의 모델로써 중앙 집중식 인증서버(Authentication Server)를 제공하는 관용암호방식으로 개발되었다[1]. 관용암호방식은 키분배의 곤란으로 분산네트워크 환경에 매우 제약적이다. 이러한 문제를 해결하기 위해서 IETF working group에서는 두 영역과 영역사이, 인증기관과 지역을 공개키 암호방식을 사용하여 상호 서비스하는 메커니즘으로 PKINIT(Public Key Cryptography for Initial Authentication)를 제안하고 있다[4]. 영역 간 인증구조인 PKCROSS (Public Key Cryptography for Cross-Realm Authentication)는 원격 인증티켓을 획득하기 위해 상호영역 인증에 대한 구조를 정의하고 있으나 구체적인 표준안이 제시되지 않고 있다[5]. Kerberos 시스템은 패스워드 기반의 비밀키 운용으로 디지털 서명이 지원되지 않기 때문에 각 개체들은 KAS(Kerberos Authentication Server, 약칭 AS)을 전적으로 신뢰해야 한다는 가정을 두고 있다[6]. 따라서 본 논문의 제안은 네트워크 상에서 신뢰성과 키 분배, 디지털 서명 등의 문제점들을 해결할 수 있도록 인증서기반의 Multi_Kerberos 인증에 관해 중점적으로 연구하였다. 본 논문은 멤버의 초기인증과 세션키 분배[7], Kerberos 인증경로 구축, 영역 간 키 교환 및 키 복구 메커니즘으로 구성한다. 인증서 발행단계는 KAS로부터 X.509 인증서가 발급된 것으로 가정하고 본문에서는 생략한다.

Ⅱ. Kerberos 인증시스템

2.1 Multi_Kerberi 인증 프로토콜

Kerberos는 여러 가지 요소로 구성된 복합시스템으로 Kerberos 서버와 티켓승인서버(TGS : Ticket Granting Server), 티켓(Ticket), 인증자(Authenticator)로 구성되어 있으며 Kerberos 서버와 TGS가 티켓을 생성하여 TGS와 서비스 서버와의 통신에 사용된다. 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.

버전(Ver)4에서 KAS는 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(IDC, PC, IDV)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(kc)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 Client와 TGS간, Client와 서버 V간에 세션키(kc,tgs ,kc,v)를 제공하여 신원을 확인하고 유효시간(Lifetime)을 인증자(Authenticatorc)에게 짧게 하여 가로채기의 위험을 방지하고 있다. Ver 5에서 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 영역 간에 상호동작을 가능하도록 확장성을 더욱 용이하게 개선되었다.

- Kerberos V.5의 시스템 계수

#：개체
C：Client
V：Application Server
ID#：#개체의 식별자

#_rem : Remote의 개체
AI : 알고리즘 식별자
A# : #개체의 인증자
AD# : #개체의 주소
ID# : #의 식별자
E : Encryption
k# : #개체 비밀키
k#,# : 해당 #개체와 #개체간의 세션 키
Times : 티켓의 시작, 종료, 유효시간
TS : Time Stamp

표 1. 다중 Kerberi 인증
Table 1. Multi Kerberi Authentication

(1) ID_c, ID_{tgs}, TS_1
(2) $E_{Kc}[K_{c,tgs}, ID_{tgs}, TS_2, LifeTime_2, \mathbf{Ticket_{tgs}}]$
(3) ID_{tgsrem}, $Ticket_{tgs}$, $Authenticator_c$
(4) $E_{Kc,tgs}[K_{c,tgsrem}, ID_{tgsrem}, TS_4, Ticket_{tgsrem}]$
$\mathbf{Ticket_{tgsrem}}=E_{Ktgsrem}$ [flag, $K_{c,tgsrem}$, $Realm_c$, ID_c, Times1, AD_c]
(5) ID_{vrem}, $Ticket_{tgsrem}$, $Authenticator_c$
(6) $E_{Kc,tgsrem}[K_{c,vrem}, ID_{vrem}, TS_6, \mathbf{Ticket_{vrem}}]$
(7) $Ticket_{vrem}$, , $Authenticator_c$
$\mathbf{Ticket_{vrem}}=E_{Kvrem}$[flag, $K_{c,vrem}$, $Realm_c$, ID_c, Times1, AD_c]

<표1>과 <그림 1>에서 메시지(1)과 (2)는 local TGS에 접근하기 위한 티켓을 요구하고 응답한다. 메시지(3)과 (4)에서 Remote TGS용 티켓을 획득하기 위한 요구 및 응답이다. TGS는 원격 TGS의 식별자인 ID_{tgsrem}와 원격 TGS와 지역 client간의 세션키($k_{c,tgsrem}$)를 생성하여 분배한다. 상호인증이 필요한 영역들은 사전에 AS, TGS, Application Server와의 공유키를 교환하여 비밀리에 보관하고 있어야 하며 인증에 참여 영역이 많아질 경우 키 교환과 관리의 문제가 발생한다.

<표1>과 <그림 1>의 Multi_Kerberi 인증프로토콜을 요약하면 다음과 같다.

(1) Local TGT(Ticket Granting Ticket) Ticket요청
(2) 인증 및 Local TGT Ticket발급
(3) Remote TGT 요청
(4) 인증정보 확인 및 Remote TGT 발급
(5) Remote SGT(Server Granting Ticket) 티켓 요청
(6) Remote SGT 티켓 발급
(7) 서비스 요구

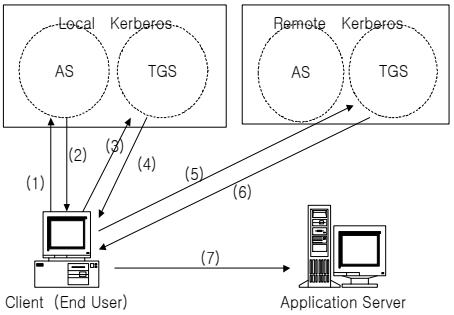

그림 1. 다중 Kerberi 메커니즘
Fig. 1 Multi Kerberi Mechanism

단일 인증 프로토콜로써 최적의 메커니즘이라고 하는 Kerberos는 분산 환경에서 효율적으로 사용하기 위해서는 공개키 도입이 필수적이라 하겠다[12].

PKINIT는 공개키 암호로 기밀 유지를 제삼자에게 전적으로 의지해야 할 필요를 없애기 위해 생긴 것이다. Kerberos 체제는 관리 영역 내에서만 사용된다면 문제가 없는 것으로 판단된다. 하지만 상이한 영역들 간에 사용되기 위해서라면 더 강력한 기능과 특성을 갖춘 공개키 체제가 Kerberos보다 더 적당할 것이다. 이를 위하여 Kerberos 체제와 공개키 암호 체제간의 접목을 위한 연구가 있어 왔다[4][8][9][10]. 대칭키를 사용하는데 있어서 가장 큰 문제점은 통신의 주체 간에 하나의 키를 교환하여 보유하고 있어야만 통신이 가능하다는데 있다. 또한 사용자의 서비스요구 시마다 TGS에서 발급하는 Ticket을 사용하는 것으로 최초 서비스에 접근하는 메시지만 패스워드를 해시(hash)함수를 이용해서 산출된 값을 키로써 사용하고 있다.

Ticket은 도용을 대비하여 timestamp와 lifetime 적용하고 변조방지를 위한 KAS와 TGS간의 공유키로 암호화하며 Client의 식별을 위한 인증자(Authenticatorc)를 사용한다. TGT를 가로채기 당한 경우와 소유자와의 일치성 결정을 위해 암호 키와 세션키를 사용하고 있으며 재전송 공격을 대비한 임의의 수 nonce를 적용하고 있다. Kerberos는 서버의 신뢰를 가정한 영역 간 인증 메커니즘을 제공하나 다음과 같은 문제점을 갖는다.

첫째, Ticket 발행에 의해 사용자인증은 실현하나 디지털서명기능은 제공하지 못하고 있다. 둘째, Kerberos 서버의 보안문제로는 다른 모든 Client와 응용서버, 응용서버와 응용서버 간 완전한 신뢰상태를 위하여 상호 완전한 신뢰를 가정하고 있다. 셋째, 패스워드에서 유도된 long term key

를 Client와 Kerberos가 비밀키로 공유하기 때문에 사전공격에 취약하다. 넷째, IETF CAT에서 PKINIT에 의한 공개키 사용의 대한 언급만 했을 뿐 구체적인 메커니즘이 제시되지 않고 있다. 다섯째, 각 유형별 서비스마다 각기 다른 세션 키를 사용하게 되는데 세션 키들의 분실에 대비한 키 복구지원기능이 없다.

2.2 IETF Kerberos 인증 프로토콜

[그림 2]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 원격 KDC가 지역 KDC의 정보를 인증한 후 티켓을 발급하고 있다.

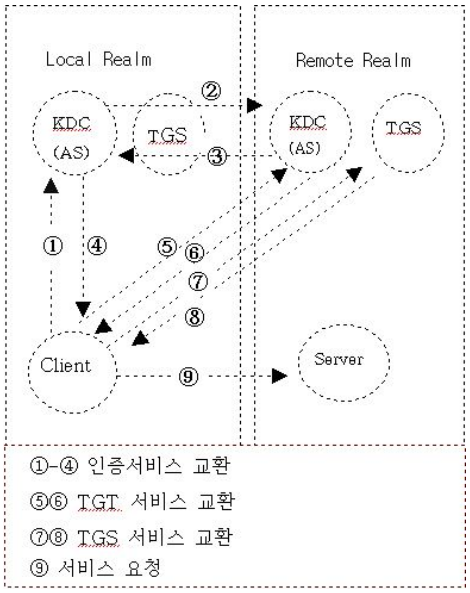

그림 2. IETF의 인증 메커니즘
Fig. 2 Authentication mechanism of IETF

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(Krand)에 의한 공통키의 생성과 암호화 과정을 수행하지 않는다. KDC_r은 KDC_l에 인증결과(KDC_Cert)만을 되돌려주고 Client에게로 직접 티켓을 할당하는 메커니즘이다.

III. Multi_Kerberos 인증시스템 설계

3.1 초기인증과 세션키 분배

3.1.1 시스템 구성요소

본 논문에서의 Kerberos 인증시스템 구조는 도메인 skku.ac.kr(local Kerberos)의 client와 mnd.go.kr(remote Kerberos)의 Application Server간 인증으로 해당 KAS들과 상호인증 및 키 분배 프로토콜을 통하여 안전한 통신채널을 형성한다. 인증시스템 구성으로 인증서를 발급하고 이를 활용하여 문서가 인터넷상에서 전자적으로 처리되기 위해서는 Client의 신청내용을 등록하고 인증서를 발급, 관리하며 필요시 인증서의 진위 여부를 확인할 수 있게 하는 인증용 시스템이 갖추어져야 한다.

Kerberos 인증모델의 구성은 인증기관의 역할을 수행하는 KAS와 티켓발급 서버인 TGS, 자원들을 쉽게 위치시키고 관리하기 위해 X.509 인증서 정보를 저장하며 X.500과 LDAP(Lightweight Directory Access Protocol)를 지원하는 디렉터리 서버, 인증서 경로생성 및 검증을 위한 VS (Validation Server), 개체 자원정보를 저장하는 Repository, User/Client, Application Server로 구성된다.

3.1.2 멤버 초기인증과 키 교환 알고리즘

・Multi_Kerberos의 시스템 계수
#_l(r) : Local(Remote)의 개체
AI : 알고리즘 식별자
S : Application Server
Authpack : 재전송 방지 값을 포함한 KAS 및 entity의 공개정보
ctime : INTEGER(for replay prevention as in RFC1510)
cusec : Kerberos time(for replay prevention as in RFC1510)
Dk#,# : # 개체간의 D-H 키 교환에 의해 생성된 공유키
nonce : INTEGER(binds response to the request)
PE[] : 공개키 암호알고리즘(RSA)으로 생성된 암호문
pk_# : # 해당개체의 공개키
SE[] : 대칭키 암호알고리즘(DES)으로 생성된 암호문
Sig[] : 공개키 암호알고리즘(RSA)으로 생성된 서명문
sk_# : # 해당개체의 개인키
xi_# : # 개체의 비밀 값

yi_# : # 개체의 공개 값

본 절은 도메인 멤버에 대한 공개키 인증은 개인키 소유에 대한 증명과 세션 키 분배를 목적으로 한다. X.509 인증서로 운용되고 있는 모든 암호시스템에 적용시킬 수 있다. 공개 값을 이용하여 사용자 개체와 인증서버(KAS)간의 인증과 교환정보를 기초로 한 인증서버는 Diffie-Hellman(이후 D-H로 표기) 키 교환 방식을 이용하여 세션 키를 생성 및 분배한다. 공개키 초기인증 요청에서 영역에 등록하는 모든 개체가 생성하여 전송한 공개 값(Subject Public Value)을 KAS의 세션 키 저장소에 보관한다. 응용서버와 개체 간에 암호화된 데이터의 세션 키가 유실되더라도 인증서버인 KAS에 의해서 만이 키를 복구할 수 있는 인증시스템이다.

사용자의 인증정보를 검증하기 위해 개인키의 생성여부, 공개키 인증자 정보, 자신이 발급한 인증서 일련번호를 검증서버를 통해 확인한다. 검증내용은 CA(Certification Authority)의 인증서 서명여부, SigAuth-pack 검증실패, 인증서 유효기간, 인증서의 취소여부, 인증서의 entity이름, entity의 서명검증, KAS와 entity의 time 들을 말한다. entity의 subject Public Value로 D-H 세션 키(Dkc,KAS)와 세션 키(kc,KAS)를 산출하고 KAS의 세션 키 저장소에 사용횟수, 유효기간, 재사용 방지를 위한 생성시간을 보관한다.

제안 메커니즘의 Multi_Kerberos(KAS, TGS)는 암호키 관리, 공개키 디렉토리 관리, 공개키 인증, 세션 키의 생성 및 분배 등의 역할을 담당한다. 인증서 경로구축과 검증을 위해 검증서버를 사용하며 디렉토리 서버와 DNS는 기존방식을 이용한다. 세션 키 저장소는 각 개체와 KAS간에 공개키 암호에 의한 멤버의 초기인증에서 개체로부터 수집되는 D-H 값(비밀 값), 분배된 세션 키를 KAS의 공개키로 암호화하여 저장하며 개체에 의한 키 복구 요청 시 D-H 값을 활용한다. 이 세션 키는 만료시간이 경과되면 새로운 세션 키를 KAS로부터 발급 받기 때문에 주기적으로 갱신이 이루어진다.

표 2. 세션 키 저장소
Table 2. Session Key Repository

User ID		· ID_C
Session Key		· $PE_{pk\_kas}[K_{c,kas}]$ · $PE_{pk\_kas}[DK_{c,kas}]$
D-H Value	비밀 값	· $PE_{pk\_kas}[xi\_c]$
	공개 값	· yi_c
	base	· g
	prime	· p
Max_life		· 00:00:00:00
Expiration		· 00:00:00:00
trans_num		· 5
mod_date		· 00:00:00:00
Mod Name		· .
KAS Time		· 00:00:00:00

모든 멤버들과 공개키 암호에 의한 초기인증이 완료되면 KAS는 세션키 저장소에 생성한 D-H 공유키와 D-H Value 등을 인증결과로 저장<표 2>하며 저장된 D-H value는 사용자가 응용서버와 통신할 때 세션 키를 생성하는 인자이며 KAS가 사용자의 키 분실 시 복구해 주는 역할을 한다.

3.2 영역 간 인증 및 키 교환 프로토콜

<그림 3>은 서로 다른 영역을 갖는 지역영역 KAS_l (skku.ac.kr)과 원격영역인 KAS_r (mnd.go.kr)의 환경으로 KAS는 인증 프로시딩과 TGS가 사용할 티켓을 발급하는 역할을 한다. 티켓의 구성요소는 발급자, 세션키, 발급대상의 식별자와 주소, 발행시간, 재 전송 방지용 값(timestamp, nonce)을 포함한다. 또한 TGS는 Tickettgs_r을 소유한 개체에 대하여 Ticket SGT(Tickets_r)를 발급하는 서비스를 담당한다.

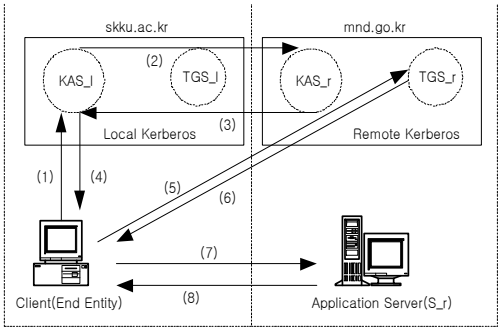

그림 3. 영역 간 Kerberos 인증 및 키 교환
Fig. 3 Kerberos Authentication and Key Distribution Structure in the inter-realms

Kerberos는 인증과 동시에 키 분배, 저장소 및 키 관리를 수행하며 지역 client가 원격 server의 서비스를 받기 위한 인증과 메시지 교환을 수행한다. <그림 3>의 제안 메커니즘에 대한 단계별 처리 절차는 다음과 같다.

(1) KRB_KAS_REQ (End Entity → KAS_l)

원격 서버인 KAS_r에 접속을 위해서는 인증이 필요하다. 인증과 원격 TGT 요청을 위해 Client와 local KAS간의 세션키로 암호화한 메시지를 전송한다.

[pvno, msg-type, padata, req-body]

padata ::= SEk_{c,kas_l}[patimestamp, pausec]

req-body ::= [ID_c, cname, $realm_{s\_r}$, sname]

KAS_l의 KRB_KAS_REQ 검증

KAS_l은 client로부터 KRB_KAS_REQ 메시지를 수신하여 저장소로부터 옵션들을 검색, 유효성과 적법성을 확인한다. Directory Server(DS)는 각 영역을 상호인증하기 위한 전·후방 인증 체인을 생성하여 원격영역의 공개키를 획득한다. Kerberos VS는 클라이언트의 요청에 대한 신속한 응답을 지원하고 효율을 높이기 위해 인증기관 인증서로 이루어진 인증경로를 미리 생성하여 저장한다. 인증경로를 생성한 후 인증경로 검증작업을 수행하여 유효하지 않은 인증경로는 제거한다. 이는 클라이언트의 요청으로 인증경로를 생성과 인증경로 검증시보다 신속하게 처리할 수 있게 한다. 도메인 <그림 4>에서 검증서버는 Client로부터 수신한 검증대상 인증서(Queried Cert)를 수신하게 된다[step 1]. 인증서버는 이미 저장된 인증경로 테이블(KASCertpath table)에서 생성될 수 있는 인증경로를 수집한다[step 2]. 인증경로가 수집되면 인증서 검증을 수행하며 이미 저장된 인증경로에 요청에 적합한 인증경로가 없을 경우 검증서버에 대하여 인증경로 생성을 요청한다[step 3]. 인증경로 생성 요청을 받은 검증서버는 자신의 이미 저장된 인증기관 인증경로 테이블(VSCertPath Table)에서 해당 검증대상 인증서에 합당한 인증경로를 선택하고 검증대상 인증서와 인증경로를 연결하여 요구되는 인증경로를 생성한다[step 4]. 인증경로 생성이 성공하는 경우 다음 요청을 위하여 생성된 인증경로를 인증경로 테이블에 저장한다. 이미 저장된 인증경로가 없으면 인증경로를 생성[step 5]하고 인증경로 생성을 실패하는 경우 검증서버는 실패처리를 수행하고 그 결과에 대한 로그 및 응답을 생성한다[step 6]. 인증경로 생성이 성공하면 KAS의 인증경로 테이블에 저장[step 7]한다. 사용자의 인증서 획득은 인증서의 신뢰체인을 사용하며 전방인증서는 skku.ac.kr《ac.kr》ac.kr《kr》kr≪go.kr≫go.kr≪mnd.go.kr≫이고 후방인증서는 mnd.go.kr《go.kr》go.kr≪kr≫kr≪ac.kr≫ac.kr《skku.ac.kr》이 된다.

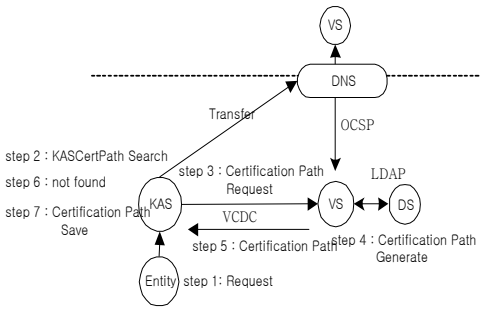

그림 4. 검증서버 모델
Fig 4. Kerberos Validation Model

(2) KRB_RMT_KAS_REQ (KAS_l → KAS_r)

KAS_l은 메시지 (2)에서 client와 자신의 정보를 통합한 Auth-Pack과 Certificate 유형인 User-Type, 사용자의 인증서 KAS_l≪C≫를 디렉토리 체인에 의해 획득한 KAS_r의 공개키(pk_kas_r)로 암호화하여 전송함으로써 신원확인 및 원격 TGS 접근을 위한 티켓승인 티켓을 요청한다.

ID_c, $realm_{s\_r}$, enc-part, User-Type, KAS_l≪C≫, CertPath, clientPublicValue

enc-part ::= $PE_{pk\_kas\_r}$[Sig_{sk_c}[ID_c, A_I, $pk_{\_c}$], Auth-Pack, keyIDlist]

Auth-Pack ::= [ID_{kas_l}, Realm, cusec, ctime, nonce, A_I, A_P]

KeyIDList ::= cksumtype(crc-32, rsa-md4, rsa-md4des, rsa-md5, rsa-sha1des, des-mac, rsa-md4-des-k)

(3) KRB_RMT_KAS_REP (KAS_r → KAS_l)

KAS_r은 정당한 사용자라고 인증한 결과를 KRB_RMT_KAS_REP로 KAS_l에게 전송(3)하여 인증이 되었음을 확인시킨다. KRB_RMT_KAS_REQ의 응답으로 KAS_r은 메시지(3)에서 enc-part, KAS_l《C》와 keyIDList 인자로 client와 KAS_l을 인증하게 되고 KAS_l에 대한 추가적인 인증은 Auth-Pack 인증요소를 KAS_l의 공개키로 복호화하

여 이루어진다. KAS_r 영역의 TGS_r에게 client의 인증을 확인시키기 위해 Ticket은 TGS_r과의 공유키(k_{tgs_r})가 사용되며 세션 키와 client의 정보, 티켓발행시간이 포함된다. Client와 TGS_r간에 사용되는 세션키(k_{c,tgs_r})와 TGS_r에 접근할 수 있는 티켓($Ticket_{tgs\_r}$)을 KAS_l《C》로부터 추출한 client의 공개키로 암호화하여 전송한다. 이때 KAS_r은 임의의 수(nonce)와 $Ticket_{tgs_r}$을 자신 영역의 TGS_r에게 전송함으로써 client로부터 전송될 내용과 비교하도록 하여 허가된 사용자라는 것을 확신시킨다.

ID_{kas_r}, enc-part, nonce, User-Type, $Ticket_{tgs\_r}$, subjectPublicValue]

enc-part ::=[$PE_{pk\_kas\_l}$][PE_{pk_c}[k_{c,tgs_r}, TS_1, Nonce], ID_{tgs_r}]

subjectPublicValue = yi_s_r = $g^{xi\_s\_r}$ mod p

$Ticket_{tgs\_r}$ ::= SEk_{tgs_r}[k_{c,tgs_r}, IDc, ADc, TS_1, $realm_{tgs\_r}$]

(4) KRB_KAS_REP (KAS_l → End Entity)

이 형식은 client가 원격TGT 요청에 대한 KAS_l의 응답으로 KAS_r로부터 전송된 메시지 (3)을 KRB_AS_REP 형식에 맞추어 KAS_l과 client가 공유하는 세션키로 전송한다. 이 세션키는 KAS_l이 인증과 세션 키를 발급한 주 서버임을 입증하는 키가 된다. Client는 자신이 전송한 Ticket의 유효시간과의 일치여부, 임의 값(nonce), 세션키를 통하여 안전한 수신을 확신한다.

$Ticket_{tgs\_r}$, enc-part, Auth-Pack, User-type, DHValue[A, B]]

enc-part::= [SEk_{c,kas_l}][PE_{pk_c}[k_{c,tgs_r}, TS_1, Nonce, ID_{tgs_r}]]]

A = $(g^{xi\_c})^{xi\_kas-1}$ mod p, B = $(g^{xi\_s\_r})^{xi\_kas-1}$ mod p

(5) KRB_RMT_TGS_REQ (End Entity → TGS_r)

Client는 이제 티켓($Ticket_{tgs\_r}$)과 세션키(kc,tgs_r)를 보유하게 됨으로써 TGS_r에 접근할 준비가 되었다. 메시지 (5)에서 client는 TGS_r에게 티켓과 인증자, 서비스를 요청할 서버의 식별자를 포함한 메시지를 보낸 A_c는 식별자와 client의 주소, timestamp, 임의의 수가 포함되어 있다. 재사용할 수 있는 티켓과는 다르게 인증자는 한번만 사용되기 때문에 짧은 유효시간을 갖는다. 생성되는 인자는 client가 원격 server에 접근할 것임을 알리는 ID_{s_r}인자와 원격 TGS가 티켓과 정보를 비교하여 client를 인증하도록 인증자 A_c를 생성한다. $Ticket_{tgs_r}$을 소유한 사용자는 KDC_r에 의해 인증되었음을 TGS_r에게 보장해 준다.

ID_{s_r}, A_c, $Ticket_{tgs\_r}$

A_c ::= SEk_{c,tgs_r}[IDc, ADc, TS_2, nonce]

(6) KRB_RMT_TGS_REP (TGS_r → End Entity)

TGS_r은 메시지 (6)에서 원격서버를 사용할 수 있는 티켓($Ticket_{s\_r}$)과 세션 키(k_{c,s_r})를 생성하고 서버의 비밀키(k_{s_r})로 티켓을 생성한다. TS_3은 티켓의 생성시간으로 client에서 원격서버로 전송되어야 하는 값이며 nonce는 현재 세션에서 재전송이 아니라는 것을 확인하기 위해 포함한다.

SEk_{c,tgs_r}는 KDC_r에서 생성하여 비밀리에 client와 TGS_r에 전송된 세션키로 TGS_r이 생성함을 알리고 변조를 방지한다. TGS_r은 Ticket 소유자인 client와 원격서버가 사용할 세션키(k_{c,s_r}) 생성한다. $Ticket_{s-r}$을 소유한 client는 동일한 서버에 대해 서비스를 실행할 때 TGS_r로부터 새로운 티켓을 요구하지 않고 재사용할 수 있다.

SEk_{c,tgs_r}[k_{c,s_r}, TS_3, $nonce_1$, $Realm_{s\_r}$], $Ticket_{s-r}$

$Ticket_{s\_r}$ ::= SEk_{s_r}[flag, k_{c,s_r}, IDc, ADc, ID_{s_r}, TS_3, $nonce_1$]

(7) KRB_RMT_AP_REQ (End Entity → S_r)

메시지 (7)에서 client는 서버 접근권한을 확인시키고 서비스를 위한 요청으로 서버용 티켓($Ticket_{s\_r}$)과 생성한 인증자, 원격 TGS로부터 전송된 계수들을 포함하여 전송한다. 원격서버는 인증자(A_c)와 TGS_r로부터 전송된 $Ticket_{s_r}$값(IDc, ADc, $Realm_{tgs\_r}$, TS_2, Nonce)을 비교하여 client를 인증하고 TGS_r이 생성하여 비밀리에 분배한 세션 키(k_{c,s_r})로 송수신할 수 있다.

$Ticket_{s\_r}$, Ac, DHValue[A, ASValue]]

$Ticket_{s\_r}$::=SEk_{s_r}[flag, k_{c,s_r}, IDc, ADc, ID_{s_r}, TS_3, $nonce_1$, $realm_{s\_r}$]

A_c ::= SEk_{c,s_r}[$nonce_1$, ADc, TS_4, IDc]

A = $(g^{xi\_c})^{xi\_kas\_l}$ mod p

ASValue ::= [prime(p), base(g)]

(8) KRB_RMT_AP_REP (s_r → End Entity)

원격서버는 client의 인증자로부터 추출한 타임스탬프 TS_3에 1을 추가하여 TS_4을 만들고 인증자의 nonce값과 임의 난수 r에 의해 산출된 E값을 추가하여 전송하게 된다.

enc-part ::= $SEk_{c,s\_r}[TS_4, nonce, E]$

$E = g^r \bmod p$ /* r random number ($r \in_R Z$)

Ⅳ. Multi_Kerberos 메커니즘 분석

4.1 세션키 복구 신뢰성

사용자는 데이터를 암호화한 키가 분실되었을 때 복구필드와 대응되는 인증정보와 함께 KAS에 전송한다. entity의 키 복구는 KAS의 비밀키가 있어야 해결할 수 있다. 키를 복구해야 하는 상황이 발생할 경우 entity의 요청에 의해 다음과 같이 수행된다.

- entity는 server로부터 전송받은 g^{xi_s} 값에 자신의 비밀키를 지수승하여 KAS로 전송한다.
- KAS는 전송된 값 C에 자신의 비밀키로 지수승하여 entity로 전송한다.
- entity는 D에 server로부터 전송된 g^r을 곱하여 $k_{c,s}$를 생성한다.

$$\cdot k_{c,s} = (((g^{xi\_s\_r})^{xi\_c} \bmod p)^{xi\_kas\_} \bmod p$$
$$= g^{xi\_s\_r * xi\_kas\_ * xi\_c + r} \bmod p$$

Diffie-Hellman 프로토콜은 이산대수가 자체보안을 위해 사용된다. $Z(p)^*$의 생성기 g는 x로부터 $y=g^x \bmod p$를 계산하기는 쉽지만 소수 p가 매우 큰 값일 경우에는 y로부터 x의 연산이 어렵다. 먼저 client와 KAS간에 D-H 알고리즘에서 사용되는 큰 소수인 p와 p보다 작은 정수 $Z(p)^*$의 생성기 $g(1 < g < p)$를 지정한다. 양 개체는 공개정보를 비보호채널을 통해 교환하고 세션키 $Dk_{c,kas_}$을 계산한다. 제3자가 안전하지 않은 채널을 이용하여 Client와 KAS간에 송, 수신한 데이터, yi_c와 yi_kas_]을 가로챌 수 있지만 xi_c와 xi_kas_]이 알려지지 않는 한 안전하다. 즉, g^{xi_c}의 값과 $g^{xi_kas_}$의 값을 이용하여 $g^{xi_c + xi_kas_}$을 알아낼 수 없기 때문에 보안성이 유지된다.

각 사용자의 비밀정보인 xi_#를 $Z(p)^*$ 상의 생성기인 g에 xi_kas_]을 멱승하여 생성함으로 비밀정보를 알기 위해서는 g^r와 $g^{xi_s_r}$을 찾아야 하기 어렵기 때문에 각 사용자의 비밀정보를 찾아내는 것은 RSA 암호시스템과 동일한 수준의 보안성을 유지할 수 있다. 기존의 암호시스템에서 개인키와 공개키는 주기적 또는 수시로 변경되어야 하나 본 시스템에서는 무작위로 생성한 난수에 의해 전송되는 메시지마다 변경된다. 이러한 난수를 제3자가 계산하려면 GF(p)상의 이산대수를 계산하여야 하기 때문에 전송정보에서 난수를 찾아내기란 계산적으로 거의 불가능하다.

4.2 Multi_Kerberos의 효율성

새로운 멤버가 등록할 때 가입의 용이성과 키 관리의 문제를 IETF의 다중 Kerberi와 제안 메커니즘을 비교하여 살펴본다. <그림 1>의 Multi Kerberi의 제한점은 2.1절 마지막 문단에서 다루었다. <표 3>에서는 <그림 1>의 다중 Kerberi 메커니즘 대한 문제가 되는 제한된 인증과정을 요약하였다.

표 3. 다중 Kerberi 메커니즘의 문제점
Table 3. Limitation of Multi Kerberi Mechanism

단계	문제 점
(1)	· AS가 client를 인증할 수 있는 인증자 결여
(2)	· client에 대한 사전인증 없이 Ticket이 발행 · 패스워드기반의 비밀키를 사용한 사전공격
(3)	· 패스워드 노출 시 제3자가 client로 위장용이 · client의 Address를 활용하여 인증자인 Ac (Ekc,tgs[IDc, ADc, TS5]) 생성한다.
(4)	· TGS_i은 TGS_r의 비밀키를 사전 분배관리 · 도메인 n과 상호 키 교환을 위해서는 안전한 채널을 통해 n(n-1)/2개의 비밀키가 교환되어야 한다. · 다른 도메인이 확장될 때 키 관리의 문제로 응용이 어렵다
(5)	· Tickettgsrem : Ektgsrem[kc,tgsrem, IDc, ADc, IDtgsrem, TS, LifeTime] · AC : Ekc,tgsrem[IDc, ADc, TS5] · TGS_r은 인증자 Ac의 암호화 키 kc,tgsrem으로만 client를 인증한다.
(6)	· 없음
(7)	· Ticketvrem과 Ac를 제3자는 획득하여 Ticket의 유효시간 범위(비록 짧지만)에서 재 사용할 경우 위험요소를 방어할 수 없다.

티켓을 생성하기 위해서는 Kerberos는 동일 도메인에 있는 응용서버의 비밀키를 불가피하게 보유하여야 하며 end entity들에 비해서 응용서버의 수가 제한적이기 때문에 사전에 보호채널을 통한 비밀키의 공유가 가능하다. 그러나 다중영역의 다른 도메인에 있는 Kerberos 서버와 비밀키를 공유하는 것은 매우 제한적일 수밖에 없으며 특히 전자상거래나 전자문서 교환 등에 응용하기 위해서는 공개키에 의한 암호화와 디지털 서명은 반드시 필요하다. <표 4>는 <표 3>

의 다중 Kerberi 메커니즘의 문제점을 보완하여 인증서 기반의 Kerberos 인증 메커니즘에 대한 단계별 분석결과이고 <표 5>는 <그림 2>의 IETF에서 제안한 영역 간 인증메커니즘을 개선하여 분석하였다.

표 4. 제안 메커니즘 평가
Table 4. Evaluation of Proposed Mechanism

단계	안전성	신뢰성	효율성
(1)	· 1회용세션키사용 · 사전 공격방어	· 사전인증절차 수행	· 동일
(2)	· 공개키 사용	· 인증서 활용 · 디지털 서명 · etype	· TGS_I 미경유 · 원격영역직접 접근
(3)	· 공개키 사용 · 재전송 공격방어	· 동일	· KAS_r에서 TGT직접발행
(4)	· 세션키 사용 · 공개키 사용	· 세션키 공유 확인	· 동일
(5)	· 동일	· 동일	· 동일
(6)	· 동일	· 동일	· 동일
(7)	· 동일	· D-H Value 활용	· 세션키의 선택적 사용
(8)	· 동일	· 동일	· 키 복구 가능

<표 4>의 단계(1)에서 단계(8)까지의 제안 메커니즘에 대한 평가는 3.1.2절의 멤버의 초기인증과 세션키 교환을 설계함으로써 Kerberos 시스템의 안전성, 신뢰성, 효율성 측면에서 다중서비스를 도출하였다.

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 디렉터리 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 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 티켓승인티켓(TGT)을 직접 전송할 수 있는 메커니즘이다. 즉 <그림 2>에서 Client는 KDC_r에 TGT를 획득하기 위한 별도의 요청신호를 필요로 하지 않는다. 또한 KDC_r에서 Client의 상호인증을 위해 생성한 임시난수 키(Krand)값을 배제한 인증서(KDC_I 《C》)를 사용함으로써 Client와 원격 TGS에서 암.복호화하는 과정이 생략되었으며 이로 인해 이중 암호화와 통신절차가 간소화되었다[표 5]. 서버용 티켓은 TGS의 키로 암호화되어 있으므로 변조가 불가능할 뿐만 아니라 Client의 공개키로 재 암호화하므로 제 3자가 티켓을 이용할 수 없다. 티켓 내에도 Client와 TGS_r, Client와 서버사이의 세션키를 포함시킴으로써 티켓 소유자가 정당한 사용자임을 증명한다.

표 5. IETF 메커니즘 분석
Table 5. Analysis of IETF Mechanism

메커니즘	분 석	
	IETF CAT Working Group	제안 메커니즘
안전성	▪ PKINIT의 공개키 획득으로 Dictionary 공격에 취약	▪ CA가 서명한 공개키 등록과 분배, D-H공유키 사용으로 안전성 보장
Client 정보	▪ sk_c[SigAuth-Pack] ▪ SigAuth-Pack: AI, pk_c	▪ clientPublicValue, KAS_I 《C》 ▪ 인증서, 알고리즘 식별자, 파라미터, 공개키 정보관리
Client 확인과정	▪ pk_c와 키 값(K_{rand})를 전송하여 확인	▪ CertPath, KAS_I 《C》 ▪ 경로값을 보유하는 인증서 체인으로 키 생성과 이중암호 배제
전송단계	▪ 8단계	▪ 8단계, 인증서기반의 상호인증으로 통신절차 간소화
티켓 발급절차	▪ KDC_r → KDC_I → Client	▪ TGS_r → Client ▪ 절차의 간소화
주요 인증인자	▪ User-Type, KDC_{LOC} 《C》, K_{rand}	▪ TrustedCertifiers, KAS_I 《C》 ▪ 인증인자의 단순화

VI. 결론 및 향후 과제

정보보호 기반기술의 중요한 요소 중 하나가 관용 암호방식을 사용하는 Kerberos 인증 메커니즘으로 현재 동일 도메인에서 사용자 인증과 키 분배를 위해 사용되는 메커니즘들 중 수행능력과 효율성 측면에서 가장 우수한 메커니즘이다. 관용암호방식인 Kerberos를 영역간 인증을 수행하기 위해서 인증서기반의 환경으로 전환하였다. 따라서 본 논문에서 제안한 주안점은 AS와 DS, VS를 연계한 X.509 인증서와 공개키를 획득하고 공개키 암호에 의한 상호인증을 통해 안전하게 키를 분배하는 것이다. KAS에 의해 키 복구가 가능하고 안전한 서비스를 지원하는 Kerberos 인증과 영역 간 효율적인 Multi_Kerberos 인증모델을 다음과 같이 제시하였다.

첫째, 공개키 기반의 X.509를 DS를 적용하여 영역간의 인증과 서비스를 제공하는 메커니즘으로 경로(CertPath)를 VS와 DNS를 통해 외부영역의 위치를 탐색하여 저장소에

저장하고 공개키 획득은 영역 간 체인을 통하여 다른 도메인의 개체를 인증하도록 하였으며 둘째, 원격 TGT요청을 지역 Kerberos로 요청하도록 하여 원격영역의 Kerberos를 경유하지 않고 원격 TGS로 직접 인증인자와 티켓을 전송함으로써 통신상의 overload를 감소시키는 효과와 인증절차의 간소화를 가지는 Kerberos 인증 메커니즘을 설계하였다. 셋째, 초기 인증에서 Diffie-Hellman의 키 교환으로 KAS의 저장소에 보관중인 공개키 정보에 의해 키 복구가 가능한 메커니즘을 제안하였으며 넷째, Kerberos 시스템의 약점인 사전공격의 취약점과 서버와 Client간의 비밀 키 보관, 디지털 서명의 제한을 해소하였고 공개키와 관용암호 인증방식의 장점을 절충하여 공개키 암호에 의한 인증과 세션 키에 의한 암호통신을 함으로써 기존의 제한적인 문제점을 충족시키고 있다.

향후 연구과제로서 인증관리 인터페이스에 대한 연구가 필요하다. Kerberos는 강력한 인증시스템이면서 보안적으로 많은 정보를 제공해 준다. 모든 서버와 client의 접속을 log와 config 파일로 유지하며 세션이 설정되면 데이터베이스에 저장된다. 다수 사용자들의 티켓을 발급하고 관리하는 일은 인증서버에 있어서 많은 부담이 된다. 각 개체들을 인증하고 인증결과와 티켓을 생성, 갱신하는 등의 인증정보를 제어할 수 있는 메커니즘이 필요하다.

참고문헌

[1] C. Neuman, T. Yu, S. Hartman, K. Raeburn. "The Kerberos Network Authentication Service (V5). 2005

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

[3] T. Parker, "A Secure European System for application in a Multi-vendor Environment," Proc. of the 14th National Computer Security Conf., 1991

[4] RFC 1510, Public Key Cryptography for Initial Authentication in Kerberos, draft-ietf-cat-kerberos-pk-init-34.txt, IETF, 2006

[5] 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

[6] A. Harbitter and D. MeKASce, "Performance of Public Key Enabled Kerberos Authentication in Large Networks." Proc. 2001. IEEE Symposium on Security and Privacy, Oakland, CA, May 13-16, 2001

[7] 신광철, 조성제, "초기인증에서 키 분배 및 복구를 지원하는 공개키 암호 인증 시스템에 관한 연구", 한국인터넷정보학회 논문지, 제 7권 3호, June. 2006, p71-79

[8] Carlisil Adams, P.Sylverster, M.Zolotarev and R.Zuccherato, "Internet X.509 Public Key Infrastructure Data Validation and Certification Server Protocol", RFC3029, 2001. 2

[9] 신광철, 정일용, 정진욱, "PKINIT기반의 Kerberos 인증과 키 교환에 관한 연구", 한국정보처리학회 논문지, 제 9-C권 3호, June. 2002. p313-322

[10] 신광철, 정진욱 "네트웍 환경에서 안전한 Kerberos 인증 메커니즘에 관한 연구", 한국정보보호학회 논문지, 제 12권 2호. April. 2002. p123-133

[11] 노종혁, 진승헌, 이균하, "인증서 검증서버의 인증서 검증방법", 정보처리학술대회, 2002

[12] M.Thomas, et al, Kerberized Internet Negotiation of Keys(KINK), draft-ietf-kink -kink-o1.txt, IETF, July 2001

저 자 소 개

신 광 철

2003년 8월 : 성균관대학교 정보공학과 공학박사

2004년 ~ 현재 : 성결대학교교수

관심분야 : 전자지불시스템, 전자상거래보안기술, 라우터보안, RFID 보안응용

조 성 제

1997년 2월 : 홍익대학교 전자계산학과 공학박사

2005년 ~ 현재 : 성결대학교교수

관심분야 : 모바일 컴퓨팅, 실시간 처리, 주기억 데이터베이스, 전자상거래