커버로스 기반의 효율적인 허가 메커니즘 설계

김 은 환[†]·전 문 석[††]

요 약

분산 네트워크 환경의 보안에 있어서 인증(authentication)과 함께 허가(authorization)는 필수적인 보안 기능이다. 허가는 사용자나 과정이 특별한 운영을 수행할 것인지를 판단하고 결정하는 절차다. 본 논문에서는 인증 메커니즘은 기존 커버로스를 사용하고 효율적인 시스템을 만들기 위해 허가 메커니즘을 설계하여 첨가한다. 제안한 허가 메커니즘은 커버로스 서버 안에 프록시 권한 서버를 운영한다. 프록시 권한 서버는 제안한 알고리즘을 사용하여 자신이 속한 영역의 사용자, 서버 및 서비스의 권한을 관리하고 허가한다. 또한, 프록시 권한 서버가 발행하는 권한 속성 인증서는 권한 위임에 사용된다. 기존의 커버로스의 인증 메커니즘에 제안하고 있는 효율적인 허가 메커니즘을 추가함으로써 보다 안전한 커버로스를 설계했다.

Design of a effective Authorization Mechanism based on Kerberos

Eun-Hwan Kim[†]·Moon-Seog Jun[††]

ABSTRACT

Authentication and authorization are essential functions for the security of distributed network environment. Authorization is determining and to decide whether a user or process is permitted to perform a particular operation. In this paper, we design an authorization mechanism to make a system more effective with Kerberos for authentication mechanism. In the authorization mechanism, Kerberos server operates proxy privilege server. Proxy privilege server manages and permits right of users, servers and services with using proposed algorithm. Also, privilege attribute certificate issued by proxy privilege server is used in delegation. We designed secure kerberos with proposed functions for effective authorization at the same time authentication of Kerberos mechanism.

키워드 : 허가(Authorization), 인증(Authentication), 커버로스(Kerberos), 권한 위임(Delegation), 프록시 권한 서버(Proxy Privilege Server)

1. 서 론

분산 네트워크를 통해서 안전한 정보 서비스를 제공하기 위해서 사용자와 사용하려는 시스템간의 인증은 필수적인 사항이다. 그러나, 인증 자체만으로 응용 서버의 서비스 사용 여부를 결정 지을 수는 없다. 그러므로, 인증(authentication)과 함께 허가(authorization)는 분산 시스템 환경의 보안에 필수적인 항목이다. 인증이 사용자나 과정의 주체를 증명하는 절차라고 한다면, 허가는 사용자나 과정이 특별한 운영을 수행할 것인지를 판단하고 결정하는 절차다. 즉, 허가는 사용자, 프로그램, 프로세스에게 허가한 권한을 의미한다. 예를 들어 임의의 사용자가 시스템 내의 임의의 파일에 접근할 수 있도록 부여된 권한 등을 의미한다. 또한, 분산 네트워크의 특성으로 인해 서비스의 분산이 이루어진다. 서비스의 분산은 권한 허가를 연결해서 사용하도록 하고 있다. 즉, 권한 위임(delegation)[9, 14] 기능이다. 권한 위임은 개시자(initiator)가 다른 중개자(delegator)에게 권한을 위임하여 개시자의 편에서 행동하도록 하는 일련의 절차를 의미한다.

분산 네트워크에서 가장 대표적인 인증 메커니즘으로 커버로스[2, 3, 10-12]가 있다. 커버로스는 중앙 집중식 인증 서버를 사용하고, 암호화 방식은 대칭키 암호화 방식을 사용하여 인증을 수행한다. 현재, 커버로스 기반의 권한 허가 메커니즘에 대한 연구[6, 7, 13, 14]는 진행중이며 일부는 사용하고 있다. 허가 메커니즘을 사용하는 시스템에는 유럽 보안 시스템 연구 프로젝트로서 공개키/개인키 개념을 도입하고 PAS(privilege attribute service)를 두어 PAC(privilege attribute certificate)를 발행하여 허가 메커니즘을 구성한 SESAME(Secure European System for Application in a Multi-vendor Environment)[4, 5] 프로젝트가 있다. 또한, OSF의 DCE(distributed computing environment)[8]도 PAC 개념을 사용하여 허가 메커니즘과 권한 위임 개념을 구현하였다.

본 논문은 기존의 커버로스 인증 메커니즘에 프록시 권한 서버(PPS : proxy privilege server)를 두어 권한 허가 메커니

† 정 회 원 : 숭실대학교 전자계산원 교수
†† 종신회원 : 숭실대학교 컴퓨터학부 교수
논문접수 : 2002년 8월 5일, 심사완료 : 2003년 3월 17일

즘 기능을 제공한다. PPS는 자신이 속한 영역의 사용자, 서버 및 서비스에 대한 권한을 관리하고 권한 허가를 해준다. PPS 동작 알고리즘은 권한 요구등의 내용을 포함한 PAC를 발행하여 권한 위임 기능을 수행한다. 또한, 제안한 허가 메커니즘은 기존의 커버로스의 인증 절차를 최대한 유지하면서 별도의 자료구조나 인증 단계를 추가하지 않은 권한 허가 메커니즘이다.

본 논문의 구성은 다음과 같다. 2장은 기존의 커버로스와 커버로스에서 사용하는 티켓과 인증자의 구조를 소개한다. 3장은 제안한 허가 메커니즘에 대한 알고리즘과 이를 적용한 분야별 프로토콜을 상세하게 설명한다. 4장은 제안한 메커니즘에 대해 비교 분석을 하고, 5장에서 결론을 맺는다.

2. 관련 연구

2.1 커버로스

커버로스는 MIT에서 Athena 프로젝트[10]로서 인증 서비스를 제공한다. 커버로스는 중앙 집중식 인증 서버를 사용하고, 암호화 방식은 대칭키 암호화 방식을 사용하여 인증을 수행한다. 커버로스가 안전하게 동작하기 위해서는 커버로스 서버, 사용자 그리고 응용 서버로 구성된다. 사용자가 응용 서버에 접근하기 위해서는 커버로스 서버에 티켓-승인 티켓을 신청하여 발급 받고, 다시 티켓-승인 티켓을 사용하여 서비스-승인 티켓을 발급 받은 후에 응용 서버에 접근한다. 티켓을 신청하거나 서버에 접속 할 때 인증자(authenticator)를 제시하여 사용자 인증을 한다. 커버로스의 자세한 동작은 [1, 10]을 참고한다.

2.2 티켓과 인증자의 구조

커버로스는 인증을 위한 메커니즘을 제공한다. 그러나 분산 환경에서는 인증(authentication)과 함께 허가(authorization)가 반드시 공존해야 한다. 커버로스는 허가를 위해서 티켓과 인증자의 구조에 선택적으로 허가를 사용하도록 자료구조를 제공하고 있지만 구현되어 있지 않다. 다음은 티켓과 인증자의 구조를 나타낸다[1].

2.2.1 티켓

```
EncTicketPart ::=       [APPLICATION 3] SEQUENCE {
        flags [0]               TicketFlags,
        key [1]                 EncryptionKey,
        crealm [2]              Realm,
        cname [3]               PrincipalName,
        transited [4]           TransitedEncoding,
        authtime [5]            KerberosTime,
        starttime [6]           KerberosTime OPTIONAL,
        endtime [7]             KerberosTime,
        renew-till [8]          KerberosTime OPTIONAL,
        caddr [9]               HostAddresses OPTIONAL,
        authorization-data [10] AuthorizationData OPTIONAL
}
```

티켓의 마지막에 authorization-data 부분이 옵션으로 지정되어 있다. 인증 메커니즘을 사용할 때는 이 부분을 사용하지 않는다. 제안하는 허가 메커니즘에서는 authorization-data 부분에 개시자나 중개자의 이름과 무결성을 위한 해쉬값을 저장한다.

2.2.2 인증자

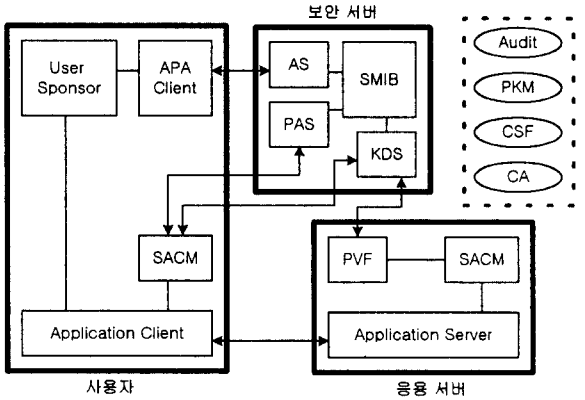
```
Authenticator ::=       [APPLICATION 2] SEQUENCE {
        authenticator-vno [0]   INTEGER,
        crealm [1]              Realm,
        cname [2]               PrincipalName,
        cksum [3]               Checksum OPTIONAL,
        cusec [4]               INTEGER,
        ctime [5]               KerberosTime,
        subkey [6]              EncryptionKey OPTIONAL,
        seq-number [7]          INTEGER OPTIONAL,
        authorization-data [8]  AuthorizationData OPTIONAL
}
```

인증자의 마지막 부분에 subkey와 authorization-data 부분이 옵션으로 지정되어 있다. 인증 메커니즘을 사용할 때는 이 부분을 사용하지 않는다. 제안하는 허가 메커니즘에서는 subkey를 이용하여 권한 속성 인증서(PAC)를 암호화하고, authorization-data 부분에 PAC를 저장한다.

2.3 SESAME

SESAME(Secure European System for Application in a Multi-vendor Environment)[4, 5]은 EU(European Union)에서 추진해온 프로젝트의 하나로, 커버로스 기반 분산환경에서의 공개키/개인키를 사용한 인증 시스템이다.

SESAME의 전체적인 구조는 (그림 1)과 같다.



(그림 1) SESAME의 구조

사용자는 자신의 US와 접속하고 APA Client와 함께 보안 서버의 AS에 접근하여 티켓-승인 티켓(TGT)을 획득한다. 사용자가 TGT를 획득한 후에 SACM은 티켓을 저장하

고 PAS로부터 PAC(privilege attribute certificate)를 얻는다. PAC는 인증서의 유효기간과 권한 사항 등에 대해서 PAS가 자신의 개인키로 전자 서명한 것이다. 사용자는 TGT와 PAC를 사용하여 KDS로부터 서비스-승인 티켓을 얻는다. 서비스-승인 티켓을 서버측 SACM에게 전송한다. 서버측의 PVF는 PAC와 서비스-승인 티켓을 검증한다. 검증을 확인하면 권한에 따른 데이터 전송을 시작한다. CA, CSF, PKM과 시스템 감사등은 SESAME을 기능적으로 지원한다.

3. 제안하는 메커니즘

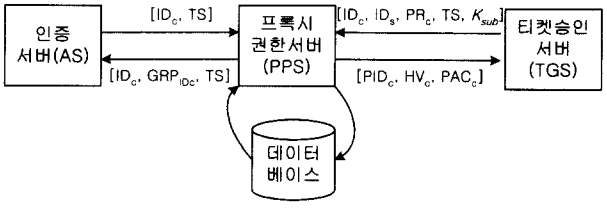
제안하는 메커니즘의 구성은 사용자, 커버로스 서버 및 응용 서버로 이루어지고, 커버로스 서버는 인증 서버, 티켓 발행 서버, 프록시 권한 서버로 구성된다. 인증 서버, 티켓 발행 서버는 기존 커버로스의 역할을 수행한다. 프록시 권한 서버는 사용자가 요청한 요구 권한에 대해서 허가해 주는 역할을 담당한다.

3.1 프록시 권한 서버와 권한 속성 인증서

프록시 권한 서버(PPS : proxy privilege server)는 커버로스 서버 내에 위치하고 인증 서버(AS : authentication server)와 티켓 승인 서버(TGS : ticket granting server)와만 직접 통신이 가능하다. PPS는 영역 내의 사용자, 응용 서버, 사용하려는 서비스에 대한 권한이 사전에 PPS에 등록되어 있어야 한다. 처음 사용자가 AS에 티켓-승인 티켓을 신청할 때, AS는 PPS에 사용자에 대한 권한 정보를 얻어 토큰(token)형태로 티켓-승인 티켓에 포함하여 사용자에게 발급한다. PPS가 발급한 토큰에는 사용자 이름(ID_c), 사용자가 포함된 소속 그룹 이름(GRP_{IDc})과 토큰이 생성된 타임 스템프로 구성된다.

$$C_{token} = [ID_c, GRP_{IDc}, TS]$$

사용자는 티켓-승인 티켓을 이용해서 접근하고자 하는 서버의 서비스에 대한 권한 요구를 TGS에게 요청한다. TGS는 토큰과 그 외 정보를 PPS에게 전달한다. PPS는 사용자의 요구 권한에 대해서 권한 여부를 확인하고 올바른 권한 사용자인 경우에 권한 속성 인증서(PAC : privilege attribute certificate)를 발급한다. (그림 2)는 PPS가 PAC를 발급하는 과정이다.



(그림 2) PPS의 동작 과정

PAC를 발급하기 위해서 요구자의 이름(ID_c), 접근하고자 하는 서버나 서비스의 이름(ID_s), 권한 요구(PR_c), 타임 스템프(TS), 서브키(K_{sub})등을 입력받는다. PPS의 출력으로 PAC의 내용은 PAC의 이름(PID_c), 요구자들의 집합(CV_c), 권한 요구(PR_c)이며 사용자로부터 전해진 서브키를 사용하여 암호화한다. 사용자로부터 전달된 서브키는 사용자가 세션 동안만 사용하도록 임의로 만든 키다. PID_c, HV_c는 TGS로 전달되어 티켓에 포함된다.

$$PAC_c = E_{Ksub}[PID_c \parallel CV_c \parallel PR_c]$$

PID는 PAC의 이름으로 PPS에서 랜덤 수를 만들어 한 세션이 끝날 때까지 사용한다. PID는 티켓에 포함되어 후에 PAC 확인을 위해서 사용한다. CV(control value)는 요구자들의 집합으로 권한 위임(delegation)이 발생할 경우 각 요구자와 중개자들을 연결시켜 놓는다.

$$CV_c = [\{(ID_c, ID_s)\} \parallel \{PR_c\}]$$

PR은 접근하려는 서버나 서비스에 대한 행동(SV_c)과 요구자의 소속 그룹 정보(GRP_{IDc}) 등을 포함한다.

$$PR_c = [SV_c, GRP_{IDc}]$$

HV(hash value)는 PAC의 해쉬값으로 다음과 같이 구한다. H는 해쉬 함수를 나타낸다. HV는 티켓에 포함되며 후에 PAC의 무결성 확인에 사용된다.

$$HV_c = \boldsymbol{H}(PID_c \parallel CV_c \parallel PR_c)$$

PPS의 동작 알고리즘은 다음 (그림 3)과 같다.

```
(a) PPS로 입력
   IDc : 요구자(혹은 중개자)      IDs : 목적지(혹은 중개자)
   PRc : 권한 요구                TS : 타임 스템프

(b) PPS의 동작
   IF IDs in inter-realm THEN
     IF SVc in database THEN
       IF IDc in database or GRPIDc in database THEN
         Generate_CV ;
         make HV
         Generate PACc ;
       ELSE
         Deny PRc
       END IF
     END IF
   ELSE
     Generate ProxyPACc
   END IF

(c) PPS의 출력
   PACc 또는 ProxyPACc : PIDc, CVc, PRc, Lc, HVc
```

(그림 3) PPS의 동작 알고리즘

PPS의 동작 알고리즘은 접근하려는 목적지 응용 서버가

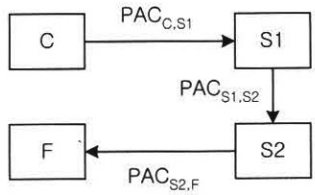
내부 영역에 존재하는지 외부 영역에 존재하는지를 확인한다. 목적지 응용 서버가 내부 영역에 존재하는 경우에는 요구자의 서비스와 이름 및 그룹 정보를 PPS가 관리하는 데이터 베이스에서 검색하여 권한 요구가 유효한지를 확인한다. 유효하면 PAC를 작성한다. 그렇지 않다면 권한 요구를 거부한다. 만일 외부 영역에 목적지가 존재하는 경우에는 ProxyPAC를 작성한다. ProxyPAC는 요구자의 권한 요구에 대해서 일단 허가하는 의미를 담고 있다. ProxyPAC는 외부 영역의 커버로스로 전송되어 외부 영역의 PPS에서 한번 더 유효성을 확인한다. ProxyPAC를 사용하는 이유는 모든 영역의 응용 서버나 서비스에 대한 권한을 PPS가 관리하고 저장할 수 없기 때문이다. 즉, 자신이 속한 영역에 대해서만 응용 서버나 서비스에 대한 권한을 관리하고 허가하기 위함이다.

3.2 권한 위임

권한 위임(delegation)은 분산 환경에서 개시자(initiator)가 다른 중개자(delegate)에게 개시자의 편에서 행동하도록 권한을 부여하는 일련의 과정이다. 권한 위임을 허용하는 경우에는 중개자를 통해서 원하는 작업을 대신 할 수 있기 때문에 무결성, 비밀성, 신뢰성, 경로 추적성 등의 조건을 만족해야 한다.

비밀성과 신뢰성은 기존의 커버로스 인증을 바탕으로 이루어진다. PPS 자체가 커버로스 서버 내부에 존재하고, 내부의 AS 및 TGS와만 직접 통신하기 때문에 PAC 자체의 내용 변경이나 경로 변경이 발생하지 않는다. 또한, 다른 시스템과 통신할 때는 세션키를 사용하여 내용을 암호화하기 때문에 안전하다. 즉, 커버로스의 인증을 통해서 신뢰성을 보장하고 세션키를 통해서 암호화하여 전송하기 때문에 비밀성을 지킨다.

무결성은 PAC의 해쉬값(HV)과 PID값을 저장한 티켓과 세션키로 암호화한 PAC를 목적지로 전송한다. 목적지에서 티켓과 PAC를 복호화하고 PID와 해쉬값을 비교하여 무결성을 확인한다. 티켓과 PAC는 각각 비밀키와 세션키로 암호화하기 때문에 안전하다.



(그림 4) 권한 위임의 예

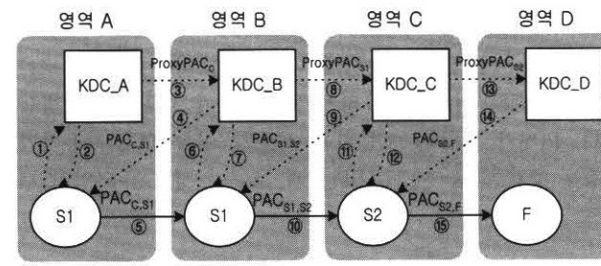
경로 추적성과 권한 확인을 위해서 제어값(CV : control value)과 권한 요구(PR)를 적용한다. CV는 권한 위임이 발생하는 경우 개시자, 중개자, 목적지의 이름을 연결 고리식으로 저장하는 객체이다. PR은 개시자의 권한 요구와 속성을 관리하는 객체이다. (그림 4)는 단일 영역 내에서 모두 커버로스 인증이 이루어진다는 가정 하에서 CV와 PR의 사용 예를 든 것이다.

개시자(C)는 $PAC_{c,s1}$를 생성하여 중개자(S1)에 전달한다. 중개자(S1)는 또 다른 중개자(S2)에게 $PAC_{s1,s2}$를 생성하여 전달한다. S2는 최종 목적지(F)로 $PAC_{s3,F}$를 생성하여 전달한다. 각각의 PAC에 저장된 CV값과 PR값의 내용은 다음과 같다.

$PAC_{c,s1}$: CV_C = {(ID_c, ID_{s1})}, PR_C = {(SV_c, GRP_{IDc})}
$PAC_{s1,s2}$: CV_{s1} = {(ID_c, ID_{s1}), (ID_{s1}, ID_{s2})},
PR_{S1} = {(SV_c, GRP_{IDc}), (SV_{s1}, GRP_{IDs1})}
$PAC_{s2,F}$: CV_{s2} = {(ID_c, ID_{s1}), (ID_{s1}, ID_{s2}), (ID_{s2}, ID_{s3})},
PR_{S2} = {(SV_c, GRP_{IDc}), (SV_{s1}, GRP_{IDs1}), (SV_{s2}, GRP_{IDF})}

CV는 권한 위임의 경로를 벡터 형태로 저장하고 있어 경로를 추적할 수 있다. PR은 개시자인 경우 접근하고자 하는 서버나 서비스에 대한 행동과 그룹 정보를 포함한다. 중개자인 경우에는 개시자의 PR과 다음 중개자에 대한 자신의 PR을 포함한다.

(그림 5)는 다중 영역에서의 권한 위임과 절차(①~⑮)를 나타낸다.



(그림 5) 다중 영역의 권한 위임

영역간에 세션키(K_{sub})로 암호화한 ProxyPAC를 발행하고, 상대 영역에서는 ProxyPAC를 통해서 요구자의 서비스와 권한 요구에 대해서 유효성을 확인한다. ProxyPAC의 구성은 다음과 같다.

$$ProxyPAC_c = E_{Ksub}[PID_c \parallel CV_c \parallel PR_c]$$

각각의 PAC에 저장된 CV값과 PR값의 내용은 단일 영역에서의 값과 같다.

3.3 전송 프로토콜

기존 커버로스를 기준으로 권한 허가 메커니즘을 적용한 전송 프로토콜이다. 전체적인 동작 단계는 기존 커버로스를 그대로 적용했으며, 권한 허가 메커니즘을 위한 자료 구조는 티켓과 인증자의 옵션 부분을 활용했다.

3.3.1 표기

전송 프로토콜에 사용할 표기법은 다음과 같다.

ADx : x의 IP 주소

ID_X : x의 이름

E_{Ka} : 키 Ka로 암호화

HV : PAC의 해쉬값

$K_{a,b}$: a와 b간의 통신에 사용할 세션키

K_a : a가 사용할 비밀키

Lx : x의 PAC 유효 기간

Lifetime : 유효 기간

PIDx : x에 대한 PAC의 이름

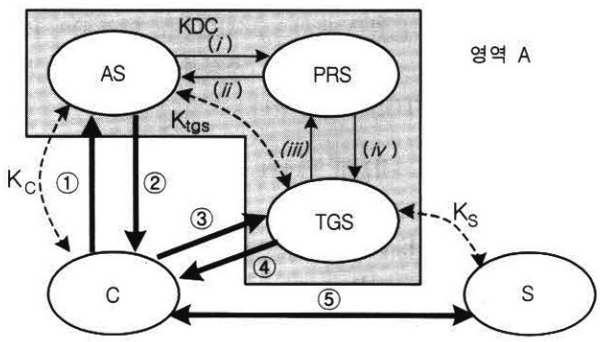
SVx : x의 서비스에 대한 행동

K_{sub} : PAC 암호화 키

TS : 타임 스템프

3.3.2 단일 영역의 동작 절차

제안하고 있는 허가 메커니즘을 위해서는 기존의 커버로스에 프록시 권한 서버(PPS : proxy privilege server)를 KDC 내부에 두어 동작하게 하였다. PPS가 KDC내부에서 동작하기 때문에 AS, TGS 및 PPS간의 데이터 전달은 특별한 암호화나 전자서명 없이 데이터를 교환한다. 단일 영역의 허가 메커니즘의 동작 절차는 (그림 6)와 같다. 굵은 실선은 시스템간에 데이터 전달을 나타내고, 가는 실선은 KDC 내부적인 데이터 전달을 의미한다. 점선은 커버로스에서 사용하는 long-term 키를 의미한다.



(그림 6) 단일 영역의 동작 절차

① C → AS : ID_c || ID_{tgs} || TS1

사용자는 자신의 이름와 TGS 사용을 요구하는 TGS의 이름을 인증 서버(AS)에 전송함으로써 티켓-승인 티켓을 요청한다.

(*i*) AS → PPS : ID_c || TS1

(*ii*) PPS → AS : C_{token}

- C_{token} = [ID_c || GRP_{IDc} || TS1]

티켓-승인 티켓을 요청받은 AS는 사용자에 대한 권한 허가 데이터를 PPS에게 요청한다. PPS는 자신의 데이터 베이스에서 사용자에 대한 소속 그룹 정보(GRP_{IDc})를 찾고 이를 포함한 토큰을 AS로 반환한다.

② AS → C : ID_c || $Ticket_{tgs}$ || E_{Kc} [$K_{c,tgs}$ || ID_{tgs} || TS2 || Lifetime]

- $Ticket_{tgs}$ = E_{Ktgs} [$K_{c,tgs}$ || ID_c || AD_c || ID_{tgs} || TS2 || Lifetime || C_{token}] : 티켓-승인 티켓

AS는 사전에 약속된 long-term 키(K_c)를 사용하여 TGS와 통신할 때 사용할 세션키($K_{c,tgs}$)등을 암호화하여 사용자로 전송한다. 또한, 티켓-승인 티켓($Ticket_{tgs}$)을 발급한다. 티켓에는 선택사항으로 사용할 수 있는 authorization-data 필드에 생성한 토큰을 첨가한다.

③ C → TGS : ID_s || $Ticket_{tgs}$ || $Authenticator_c$

- $Ticket_{tgs}$ = E_{Ktgs} [$K_{c,tgs}$ || ID_c || AD_c || ID_{tgs} || TS2 || Lifetime || C_{token}] : 티켓-승인 티켓
- $Authenticator_c$ = $E_{Kc,tgs}$ [ID_c || AD_c || TS3 || K_{sub} || Autho-data] : 인증자
- Autho-data = SV_c

사용자는 티켓-승인 티켓($Ticket_{tgs}$)과 인증자($Authenticator_c$)를 TGS에게 제공함으로써 서비스-승인 티켓을 요청한다. 이때, 인증자는 접속하려는 대상의 권한 사용 여부를 알아보기 위해 KDC에 있는 PPS에게 전송할 내용(Autho-data)과 PAC를 암호화할 서브키(K_{sub})를 포함한다. Autho-data에는 사용자가 사용하려는 서비스에 대한 행동(SV_c)를 포함한다.

(*iii*) TGS → PPS : ID_c || ID_s || PR_c || TS3 || K_{sub}

- PR_c = [SV_c || GRP_{IDc}]

TGS는 PPS에게 권한을 요청한 사용자의 이름과 접속할 서버의 이름, 권한 요구, 타임 스템프, 서브키등의 데이터를 전송함으로써 권한 확인을 요구한다.

(*iv*) PPS → TGS : PID_c || HV_c || PAC_c

- PAC_c = E_{Ksub} [PID_c || CV_c || PR_c]

PPS는 3.1절의 내용과 같이 서버에 대한 권한을 확인하고 PAC(privilege attribute certificate)를 생성하여 서브키로 암호화하여 TGS로 전송한다.

④ TGS → C : ID_c || $Ticket_s$ || $E_{Kc,tgs}$ [$K_{c,s}$ || ID_s || TS4 || PAC_c]

- $Ticket_s$ = E_{Ks} [$K_{c,s}$ || ID_c || ID_s || TS4 || Lifetime || PID_c || HV_c] : 서비스-승인 티켓
- PAC_c = E_{Ksub} [PID_c || CV_c || PR_c]

PPS로부터 PAC_c를 받은 TGS는 서비스-승인 티켓을 발급할 때 PID_c와 HV_c를 티켓에 포함하여 발급한다. TGS는

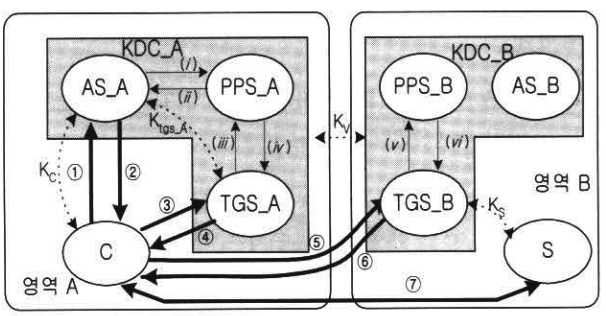
티켓을 비밀키로 암호화하고, PAC_c는 서브키로 암호화하여 사용자에게 전송한다.

⑤ C → S : $Ticket_s$ || $Authenticator_c$
- $Ticket_s = E_{Ks}[K_{c,s} \| ID_c \| ID_s \| TS4 \| Lifetime \| PID_c \| HV_c]$: 서비스-승인 티켓
- $Authenticator_c = E_{Kc,s}[ID_c \| AD_c \| TS5 \| K_{sub} \| PAC_c]$: 인증자
- $PAC_c = E_{Ksub}[PID_c \| CV_c \| PR_c]$

사용자는 서비스-승인 티켓과 인증자를 응용 서버에 제공함으로써 별도의 승인 과정 없이 응용 서버에 접속할 수 있다. 응용 서버는 티켓과 인증자를 사용하여 사용자를 인증하고 PAC_c를 확인한다. PAC에 대해서 해쉬한 결과가 티켓에 포함된 HV값과 같으면 KDC가 올바르게 권한 부여를 수행한 것을 확인할 수 있다.

3.3.3 다중 영역의 동작 절차

응용 서버는 제공하는 서비스에 대해서 자신이 속한 영역의 PPS에게 사용자나 서비스에 대한 권한을 등록한다. 그러므로 접근하려는 서버나 서비스가 다른 영역에 있는 경우는 다중 영역의 TGS에게 직접 사용자의 권한에 대한 요청을 한다. 다중 영역으로의 권한 요구에 대해서 PAC 대신에 ProxyPAC로 응답한다. 다중 영역으로의 허가 메커니즘에 대한 동작은 (그림 7)과 같다.



(그림 7) 다중 영역의 동작 절차

(그림 7)에서 굵은 실선은 커버로스의 인증 단계이고, 가는 실선은 허가 메커니즘을 위한 단계이다. 점선은 사전에 약속된 long-term 키를 나타낸다. ①~③과 (i)~(iv)의 동작은 3.1절의 내용과 3.3.2절에서 제안한 단일 영역에서의 동작 절차와 같다. 이후로는 알고리즘이 다른 부분만을 소개한다.

④ TGS_A → C : ID_c || $Ticket_{tgs\_B}$ || $E_{Kc,tgs}[K_{c,tgs\_B} \| ID_s \| ID_{tgs\_B} \| TS4 \| ProxyPAC_c]$
- $Ticket_{tgs\_B} = E_{Kv}[K_{c,tgs} \| ID_c \| ID_s \| TS4 \| Lifetime \| PID_c \| HV_c]$
- $ProxyPAC_c = E_{Ksub}[PID_c \| CV_c \| PR_c]$

영역 A의 PPS로부터 $ProxyPAC_c$를 받은 TGS는 원격 티켓-승인 티켓을 발급할 때 PID_c와 HV_c를 티켓에 포함하여 발급한다. 티켓은 비밀키로 암호화하고 $ProxyPAC_c$는 서브키를 이용하여 암호화하여 사용자에게 전송한다.

⑤ C → TGS_B : $Ticket_{tgs\_B}$ || $Authenticator_c$
- $Ticket_{tgs\_B} = E_{Kv}[K_{c,tgs\_B} \| ID_c \| ID_s \| TS4 \| Lifetime \| PID_c \| HV_c]$
- $Authenticator_c = E_{Kc,tgs\_B}[ID_c \| AD_c \| TS5 \| K_{sub} \| ProxyPAC_c]$: 인증자
- $ProxyPAC_c = E_{Ksub}[PID_c \| CV_c \| PR_c]$

사용자는 원격 티켓-승인 티켓과 인증자를 영역 B의 TGS에 제공하고, 영역 B의 PPS에게 영역 B의 응용 서버의 사용 권한과 원격 서비스-승인 티켓을 요청한다.

(v) TGS_B → PPS_B : $ID_c \| ID_s \| PR_c \| TS5 \| K_{sub}$

영역 B의 TGS는 PPS에게 권한을 요청한 사용자의 이름과 접속할 서버의 이름, 권한 요구, 타임 스템프등의 데이터를 전송함으로써 사용자의 영역 B에 속한 응용 서버에 대한 권한 확인을 요구한다.

(vi) PPS_B → TGS_B : $PID_{c,tgs\_B} \| HV_{c,tgs\_B} \| PAC_{c,tgs\_B}$
- $PAC_{c,tgs\_B} = E_{Ksub}[PID_{c,tgs\_B} \| CV_{c,tgs\_B} \| PR_{c,tgs\_B}]$

영역 B의 PPS는 사용자의 영역 B에 속한 응용 서버에 대한 권한을 3.1절과 같은 방법으로 확인하고 영역 B의 PAC (privilege attribute certificate)를 생성하고 서브키로 암호화하여 영역 B의 TGS로 전송한다.

⑥ TGS_B → C : $ID_c \| Ticket_s \| E_{Kc,tgs}[K_{c,s} \| ID_s \| TS4 \| PAC_{c,tgs\_B}]$
- $Ticket_s = E_{Ks}[K_{c,s} \| ID_c \| ID_s \| TS5 \| Lifetime \| PID_{c,tgs\_B} \| HV_{c,tgs\_B}]$
- $PAC_{c,tgs\_B} = E_{Ksub}[PID_{c,tgs\_B} \| CV_{c,tgs\_B} \| PR_{c,tgs\_B}]$

영역 B의 PPS로부터 PAC_{c,tgs_B}를 받은 영역 B의 TGS는 서비스-승인 티켓을 발급할 때 PID_{c,tgs_B}와 HV_{c,tgs_B}를 티켓에 포함하여 발급한다. 영역 B의 TGS는 티켓은 비밀키로 암호화하고 PAC_{c,tgs_B}는 서브키를 이용하여 암호화하고 사용자에게 전송한다.

⑦ C → S : $Ticket_s \| Authenticator_c$
- $Ticket_s = E_{Ks}[K_{c,s} \| ID_c \| ID_s \| TS5 \| Lifetime \| PID_{c,tgs\_B} \| HV_{c,tgs\_B}]$
- $Authenticator_c = E_{Kc,s}[ID_c \| AD_c \| TS3 \| PAC_{c,tgs\_B}]$: 인증자
- $PAC_{c,tgs\_B} = E_{Ksub}[PID_{c,tgs\_B} \| CV_{c,tgs\_B} \| PR_{c,tgs\_B}]$

사용자는 서비스-승인 티켓과 인증자를 응용 서버에 제공함으로써 별도의 승인 과정 없이 원격지 응용 서버에 접속할 수 있다. 원격지 응용 서버는 티켓과 인증자를 이용하여 사용자를 인증하고 티켓에 포함된 PID_{c,tgs_B}, HV_{c,tgs_B}와 PAC_{c,tgs_B}에 포함된 PID_{c,tgs_B}, HV_{c,tgs_B}를 확인하고 PR_{c,tgs_B}을 수행한다.

4. 제안한 메커니즘의 분석

제안한 허가 메커니즘은 기존 커버로스의 인증 메커니즘을 최대한 활용 할 수 있도록 개선했다. 즉, 기존의 커버로스 인증 메커니즘에 허가 메커니즘을 접목했다. 기존 커버로스 서버에 프록시 권한 서버를 두어 권한 허가 메커니즘을 설계했다. 그리고 제안하고 있는 메커니즘과 기존의 SESAME 시스템을 비교 분석한다.

제안한 메커니즘은 전송 단계의 추가 없이 기존의 커버로스의 동작 절차를 그대로 적용했다. 단일 영역의 동작 절차는 5단계, 다중 영역인 경우는 7단계를 수행한다. 그리고 기존의 티켓과 인증자의 선택사항 부분을 사용했기 때문에 별도의 데이터 구조를 추가하지 않았다. 권한 허가 인증서에 대한 부인 봉쇄 기능은 서브키를 사용하여 구현했다. SESAME는 허가 메커니즘을 위해서 PAS를 두고, PAS와 별도의 데이터 구조를 운영한다. 전송단계는 단일 영역인 경우에는 모두 9단계, 다중 영역인 경우는 11단계를 수행한다. 또한, 공개키를 사용하여 권한 허가 인증서에 대한 부인 봉쇄 기능을 수행했으며, 공개키 사용으로 인한 신뢰센터를 따로 운영해야 하는 부담이 생긴다.

권한 서버의 사용 용도에 있어서 제안한 메커니즘은 권한 서버가 자신이 속한 영역내의 응용 서버에 대한 서비스에 대한 권한만을 저장하고 처리하기 때문에 권한 서버의 부담을 줄였다. 다중 영역에 대한 권한 허가는 ProxyPAC를 발행하여 해당 영역의 권한 서버에서 처리하도록 했다. SESAME는 모든 응용 서버에 대한 서비스들의 권한 내용을 해당하는 모든 권한 서버가 저장하고 있어야 한다. 그러므로 권한 서버의 부담이 크다.

인증과 허가를 위해서 사용하는 키의 개수에 있어서 제안한 메커니즘은 long-term 키 (K_c, K_v, K_s, K_{tgs}) 4개, 세션키 ($K_{c,tgs}, K_{c,tgs\_B}, K_c, s$) 3개와 서브키 ($K_{sub}$)를 사용한다. SESAME는 long-term 키 ($LK_{xa}, LK_{ga}, LK_{vg}, LK_{vh}, LK_g, LK_{gh}$) 6개, 세션키 ($BK_{xg}, BK_{xv}, KI_{xyd}, KC_{xyd}$) 4개, 공개키 ($KPU_p$, KPU_g, KPU_h) 3쌍과 신뢰 센터의 공개키 (KPU_c) 1쌍을 가지고 있어야 한다[5].

다음 <표 1>은 제안한 메커니즘과 SESAME를 비교 분석했다.

5. 결 론

본 논문에서는 기존 커버로스의 인증 메커니즘과 티켓, 인증자의 구조를 분석했다. 기존 커버로스에는 인증 메커니즘만을 사용하고 있으며 허가 메커니즘에 대해서는 언급하고 있지 않다. 그러므로 기존 커버로스의 개념에 제안한 허가 메커니즘을 추가하여 인증과 더불어 효율적으로 커버로스를 사용할 수 있도록 설계했다. 기존 커버로스에 프록시 권한 서버(PPS)를 두어 영역내의 권한을 관리하도록 했다. PPS는 제안하고 있는 알고리즘을 이용하여 권한 허가 개념을 처리하고 권한 속성 인증서(PAC)를 생성하여 제공한다. PAC를 통해 권한 위임(delegation)을 처리한다. 다중 영역에서의 권한 위임 기능은 ProxyPAC를 발행하여 원격 PPS로부터 PAC를 발행하도록 하여 PPS의 부담을 줄였다. PAC를 사용하여 개시자의 권한을 위임하여 중개자로 하여금 개시자의 권한을 갖고 처리하도록 했다. 또한, 제안한

〈표 1〉 프로토콜 비교 분석

	제안한 메커니즘	SESAME
효 율 성	• 전송 단계는 기존 커버로스와 같다. • 기존 데이터 필드의 옵션 부분을 사용한다. (티켓, 인증자 부분의 옵션 사용) • 전체적인 동작 과정을 단순화 했다. • 별도의 키를 사용하지 않고 비밀성, 기밀성, 신뢰성, 경로 추적성을 보장한다. • 서브키를 사용하여 부인 봉쇄기능 수행	• 전송 단계가 복잡하다. • 별도로 부가된 데이터 필드가 많다. • 비밀성, 기밀성, 신뢰성, 경로 추적성을 위해 별도의 시스템과 키를 사용했다. • 공개키를 사용하여 부인 봉쇄 기능 수행
권한 서버의 사용 용도	• 영역내의 응용 서버나 서비스에 대한 권한만을 저장 (권한 서버의 부담이 적다) • ProxyPAC 기능 추가	• 모든 응용서버와 서비스에 대한 권한을 저장 (권한 서버의 부담이 크다)
티켓 발급 절차	• C ⇄ AS, C ⇄ TGS (4 단계)	• C ⇄ AS, C ⇄ PAS, C ⇄ KDS (6 단계)
사용하는 키 개수 (단일 영역인 경우)	long-term 키 : 4개 세션키 : 3개 서브키 : 1개	long-term 키 : 6개 세션키 : 4개 공개키 : 4쌍
공개키 사용(신뢰센터)	• 공개키 사용 안함(신뢰센터 사용 안함)	• 공개키를 사용함(신뢰센터 필요)

메커니즘은 기존 커버로스 동작 단계에는 영향을 주지 않도록 했다. 그러므로 추가되는 키의 개수나 동작 단계는 그대로 유지하면서 권한 허가 기능을 추가하여 성능을 향상시켰다.

본 논문은 기존의 커버로스에 권한 허가 메커니즘을 추가하여 인증 메커니즘과 함께 권한 허가 메커니즘을 사용한 효율적인 커버로스를 설계하였으며 향후, 많은곳에서 인증과 허가 시스템으로 사용될 것으로 기대한다.

참 고 문 헌

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

[2] J. Steiner, C. Neuman, J. Schiller, "Kerberos : An Authentication Service for Open Network System," Proc. of the Winter 1988 Usenix Conference, Feb., 1988.

[3] W. Stallings, "Network Security Essentials applications and standard," prentice hall, 2000.

[4] T. T. Parker, "A Secure European System for Applications in a Multi-vendor Environment(The SESAME Project)," Proceedings of the 14th American National Security Conference, 1991.

[5] P. V. McMahon, "SESAME V2 Public Key and Authorization Extensions to Kerberos," In Proceedings of the 1995 Symposium on Network and Distributed System Security, pp.114-131, Feb., 1995.

[6] B. Clifford Neuman, "Proxy-Based Authorization and Accounting for Distributed system," In Proceedings of the 13th International Conference on Distributed Computing systems, pp.283-291, 1993.

[7] Jonathan T. Trostle, B. clifford Neuman, "A Flexible Distributed Authorization Protocol," Internet Society 1996 Symposium on Network and Distributed System Security, pp.43-52, May, 1996.

[8] Marlena E. Erdos and Joseph N. Pato, "Extending the OSF DCE Authorization System to Support Practical Delegation," In Proceedings of the PSRG Workshop on Network and Distributed System Security, pp.93-100, Feb., 1993.

[9] M. Gasser and E. McDermott, "An Architecture for Practical Delegation in a Distributed System," IEEE Symposium on Security and Privacy, pp.20-30, 1999.

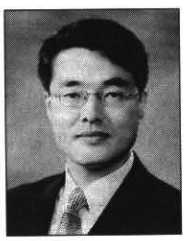
[10] http://web.mit.edu/kerberos/www/.

[11] 김은환, 전문석, "공개키를 이용한 커버로스 기반의 강력한 인증 메커니즘 설계", 정보보호학회논문지, 제12권 제4호, pp.67-76, August, 2002.

[12] 신광철, 정진욱, "네트웍 환경에서 안전한 Kerberos 인증 메커니즘에 관한 연구", 정보보호학회논문지, 제12권 제2호, April, 2002.

[13] 김철현, 정일용, "PKINIT 기반 새로운 커브로스 인증 메커니즘의 설계", 정보과학회논문지, 제28권 제1호, Mar, 2001.

[14] 유정각, 이건희, 이상하, 김동규, "PKI 기반에서 X.509 인증서를 사용한 권한 위임", 정보보호학회 종합 학술발표회논문집, 제11권 제1호, Nov., 2001.



김 은 환

e-mail : ehkim@soongsil.or.kr

1990년 숭실대학교 전자계산학과(공학사)

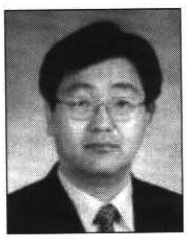
1997년 숭실대학교 대학원 컴퓨터학과 (공학석사)

2003년 숭실대학교 대학원 컴퓨터학과 (공학박사)

1990년~1995년 국방과학연구소 연구원

1997년~현재 숭실대학교 전자계산원 교수

관심분야 : 정보보호, 인증 시스템, 네트워크 및 인터넷 보안,



전 문 석

e-mail : mjun@computing.ssu.ac.kr

1981년 숭실대학교 전자계산학과(공학사)

1986년 University of Marylnad Computer Science(공학석사)

1988년 University of Maryland Computer Science(공학박사)

1989년 Morgan State Univ. 부설 Physical Science Lab. 책임연구원

1991년~현재 숭실대학교 컴퓨터학부 부교수

관심분야 : 침입차단시스템, 인증 시스템, 인터넷 보안, 전자상거래 보안, 병렬처리시스템