논문 04-29-5C-17

커버로스 기반의 안전한 인증 및 허가 프로토콜에 관한 연구

정회원 김 은 환*, 김 명 희**, 전 문 석***

Study on a Secure Authentication and Authorization Protocol based on Kerberos

Eun-hwan Kim*, Myung-hee Kim**, Moon-seog Jun*** *Regular Members*

요 약

분산 네트워크 환경에서 커버로스는 시스템간의 신뢰를 바탕으로 대칭키를 사용하여 사용자를 인증한다. 그러나, 인증(authentication)과 함께 허가(authorization)는 보안의 필수적인 요소다. 본 논문에서는 기존의 커버로스에 프록시 권한 서버(proxy privilege server)를 두고 공개키/개인키를 적용하여 효율적이고 안전한 인증 및 허가 메커니즘을 설계하였다. 제안한 메커니즘에서는 사용자 인증을 위해 미리 정해진 long-term 키와 공개키를 통해 교환한 랜덤 수에 MAC 알고리즘을 적용하여 암호화에 사용하는 세션키 값을 매번 바꾸어주기 때문에 안전성을 높였다. 또한, 전체적인 인증 절차를 간소화하여 사용하는 키의 수를 줄였다. 프록시 권한 서버는 사용자의 권한 요구를 확인하고 권한 속성 인증서(privilege attribute certificate)를 발행한다. 권한 속성 인증서는 사용자의 권한 요구를 응용 서버에 전달하고 권한 위임에 사용된다. 제안한 메커니즘을 사용하여 기존의 커버로스에서 동작하는 효율적이고 안전한 인증 및 허가 알고리즘을 설계한다.

Key Words : Authentication; Authorization; Kerberos; Delegation; Proxy Privilege Server.

ABSTRACT

Kerberos authenticates clients using symmetric-key cryptography, and supposed to trust other systems of the realm in distributed network environment. But, authentication and authorization are essential elements for the security. In this paper, we design an efficient and secure authentication/authorization mechanism by introducing the public/private-key and installing the proxy privilege server to Kerberos. In the proposed mechanism, to make a system more secure, the value of the session key is changed everytime using MAC(message authentication code) algorithm with the long-term key for user-authentication and a random number exchanged through the public key. Also, we reduce the number of keys by simplifying authentication steps. Proxy privilege server certifies privilege request of client and issues a privilege attribute certificate. Application server executes privilege request of client which is included a privilege attribute certificate. Also, a privilege attribute certificate is used in delegation. We design an efficient and secure authentication/authorization algorithm with Kerberos.

Ⅰ. 서 론

분산 네트워크 환경과 인터넷으로 인한 다양한 응용 서비스와 이를 이용하려는 사용자들은 빠른

* 숭실대학교 전산원 인터넷 정보통신 학과 전임교수(ehkim@ssuci.ac.kr), ** 숭실대학교 컴퓨터학과 컴퓨터 통신 연구실
** 숭실대학교 정보과학대학 정교수
논문번호 : 030230-0602, 접수일자 : 2003년 6월 5일

속도로 증가하고 있다. 정보를 제공하는 응용 서버는 사용자와 사용하려는 시스템간의 신뢰를 위해서 인증을 필수 사항으로 취급하고 사용자들도 더욱 안전하게 통신 하기를 원한다. 그러나, 인증만으로 제공하는 서비스의 사용 여부를 결정 지을 수 없다. 즉, 인증(authentication)과 함께 허가(authorization)는 분산 네트워크 환경에서 필수적인 요소다. 인증은 사용자나 처리 과정의 주체를 증명하는 절차라고 한다면, 허가는 사용자나 처리 과정을 수행할 것인지를 판단하고 결정하는 절차다. 현재, 개방 네트워크 환경에서 가장 대표적인 인증 메커니즘으로 커버로스(Kerberos)[1,5,7,9]가 있다. 커버로스는 중앙 집중식 인증 서버를 사용하고 암호화 방식은 대칭키 암호화 방식을 사용하여 사용자를 인증한다. 허가 메커니즘[11,12]은 대부분 인증 메커니즘과 함께 사용한다. 분산 네트워크의 특성으로 인해 서비스의 분산이 이루어지고, 서비스의 분산은 권한 허가를 분산해서 사용하도록 한다. 즉, 권한 위임(delegation)[13,14]기능이다. 권한 위임은 개시자의 권한을 다른 중개자에게 위임하여 개시자의 편에서 행동하도록 하는 일련의 절차를 의미한다.

현재, 인증과 허가에 대한 연구는 계속 진행중이며 대표적인 예는 다음과 같다. 커버로스의 인증 메커니즘을 보완하기 위해 IETF의 CAT(common authentication technology) Working Group에서는 공개키와 인증서를 기반으로 PKINIT(Public Key Cryptography for Initial Authentication in Kerberos)[2]/ PKCROSS(Public Key Cryptography for Cross-Realm Authentication in Kerberos)[3]등의 초안을 발표했다. 또한, 인증과 허가 메커니즘을 적용하여 사용하고 있는 시스템으로 유럽 보안 시스템 연구 프로젝트로써 비밀키/공개키를 이용한 대표적인 분산 개방형 시스템인 SESAME(Secure European System for Application in a Multi-vendor Environment)[10,15]와 OSF의 DCE(distributed computing environment)[13]가 있다. 위에서 연구중이거나 사용중인 메커니즘은 모두 공개키/개인키를 사용하고[17,18], 공개키 보장을 위해 신뢰할 수 있는 인증 센터[4,6]를 가정하고 있다.

본 논문은 기존 커버로스를 동작 원리를 기반으로 공개키를 보장하기 위한 인증서와 인증 센터를 사용하지 않고, 단지 각 시스템에 공개키/개인키를 적용한 새로운 인증 메커니즘을 제안한다. 새로운 인증 메커니즘은 기존의 long-term키와 공개키를 통해 교환한 랜덤 수에 MAC(message authentication code) 알고리즘[8]을 적용하여 데이터 통신에 사용하는 세션키를 매번 바꾼다. 사용자 인증과 공개키에 대한 인증은 랜덤 수와 전자 서명 기반의 시도-응답(challenge-response)기법을 적용한 상호 인증(mutual authentication) 방법[8]을 사용한다. 또한, 외부 영역의 응용 서버에 접근하기 위한 동작 절차를 축소시켜 통신비용을 줄였다. 허가 메커니즘은 커버로스 서버에 프록시 권한 서버(PPS: proxy privilege server)를 두어 허가 기능을 제공한다. PPS는 자신이 속한 영역의 사용자, 서버 및 서비스에 대한 권한을 관리한다. PPS는 동작 알고리즘을 적용하여 권한 속성 인증서(PAC: privilege attribute certificate)를 발행하고 권한 위임 기능을 수행한다.

본 논문의 구성은 2장에서 기존 커버로스와 관련 연구에 대해서 소개한다. 3장은 제안하고 있는 메커니즘에 사용하는 알고리즘에 대해서 설명하고, 4장은 인증과 허가 메커니즘의 전송 프로토콜에 대해서 상세히 소개한다. 5장은 제안한 메커니즘에 대한 분석을 하고 6장에서 결론을 맺는다.

Ⅱ. 관련 연구

1. 커버로스

커버로스는 MIT에서 Athena 프로젝트[16]로써 인증 서비스를 제공한다. 커버로스가 안전하게 동작하기 위해서는 커버로스 서버(KDC: key distribution center), 사용자(C: client)와 응용 서버(S: application server)로 구성된다. KDC는 또다시 인증 서버(AS: authentication server)와 티켓 승인 서버(TGS: ticket granting server)로 구성된다. 사용자가 응용 서버에 접근하기 위해서는 AS에 티켓-승인 티켓을 신청하여 발급 받고, 다시 TGS에 티켓-승인 티켓을 사용하여 서비스-승인 티켓을 발급 받은 후에 응용 서버에 접근한다. 티켓을 신청하거나 서버에 접속을 할 때 인증자(authenticator)를 제시하여 사용자 인증을 한다. 커버로스의 자세한 동작은 [1][16]을 참고한다.

2. 티켓과 인증자의 구조

기존 커버로스는 허가 메커니즘을 위해서 티켓과 인증자의 구조에 선택적으로 허가를 사용하도록 제공하고 있지만 실제 구현되어 있지는 않다. 다음은 티켓과 인증자의 자료 구조를 나타낸다[1].

2.1 티켓

```
Ticket ::= [APPLICATION 1] SEQUENCE {
        tkt-vno[0]          INTEGER,
        realm[1]            Realm,
        sname[2]            PrincipalName,
        enc-part[3]         EncryptedData
}
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 authorization-data 부분이 옵션으로 지정되어 있다. 인증 메커니즘을 사용할 때는 이 부분을 사용하지 않는다. 제안하는 허가 메커니즘에서는 authorization-data 부분에 PAC의 이름과 무결성을 위한 해쉬값을 저장한다.

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 authorization-data 부분이 옵션으로 지정되어 있다. 인증 메커니즘을 사용할 때는 이 부분을 사용하지 않는다. 제안하는 허가 메커니즘은 authorization-data 부분에 PAC를 저장한다.

3. SESAME

ECMA(European Computer Manufacturer Association)는 OSI security 구조에 기초를 두고 응용 계층의 보안 프레임워크를 개발해 왔다. SESAME(Secure European System for Application in a Multi-vendor Environment)[10,15]는 EU(European Union)에서 추진해온 프로젝트의 하나로 커버로스 기반 분산환경에서의 공개키/개인키를 사용한 인증 시스템이다. SESAME의 전체적인 구조는 그림 1과 같다.

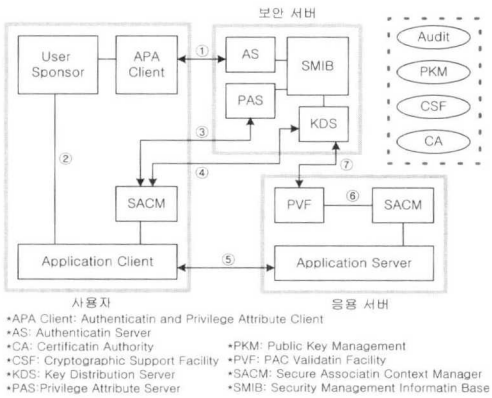


그림 1. SESAME 시스템의 구조
Fig. 1. Structure of SESAME system

사용자는 자신의 US와 접속하고 APA Client와 함께 보안 서버의 AS에 접근하여 티켓-승인 티켓(TGT)을 획득한다. 사용자가 TGT를 획득한 후에 SACM은 티켓을 저장하고 PAS로부터 PAC를 신청하여 얻는다. PAS는 자신의 개인키로 PAC의 유효기간과 권한 사항 등을 전자 서명한다. 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 감사 등은 SESAME를 기능적으로 지원한다. 그 외

에 자세한 동작은 [15]를 참고한다.

Ⅲ. 제안 알고리즘

1. 안전한 인증 프로토콜

기존 커버로스는 사전에 약속된 패스워드를 입력으로 key-derived function을 사용하여 long-term 키를 생성하여, 사용자와 지역 KDC간의 통신이나 티켓을 암호화하는 세션키나 비밀키로 사용한다. 그러나 long-term 키는 사전에 약속된 패스워드 기반에서 만들어진 동일한 값이기 때문에 노출되거나 사전 공격 등의 위험을 지닌다. 이런 취약점을 보완하기 위해서 기존 커버로스는 사전에 약속된 패스워드를 주기적으로 변경해야 하는 불편함이 있다.

본 논문에서 기존 커버로스의 메커니즘을 활용하기 때문에 인증서를 사용하지 않고 단순히 공개키만을 생성하고, 교환하여 기존 커버로스의 취약점을 보완한다.

제안하고 있는 안전한 인증 프로토콜은 그림 2와 같다.

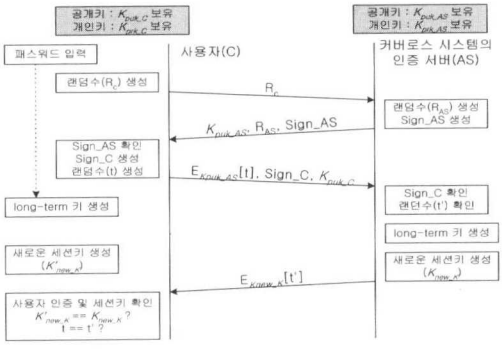


그림 6. 안전한 인증 프로토콜
Fig. 2. Secure-authentication protocol

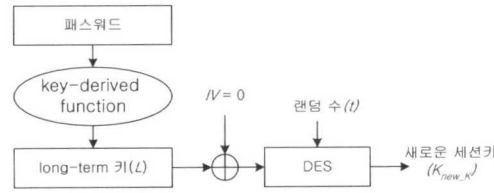
사용자는 패스워드 입력과 함께 보유하고 있는 공개키를 교환하고 각자의 전자서명을 통해서 상호 인증을 수행한다. 이때 사용하는 전자 서명은 다음과 같다.

$$Sign\_C = EKprk\_C[RC, RAS, IDAS]$$
$$Sign\_AS = EKprk\_AS[RAS, RC, IDC]$$

IDC와 IDAS는 사용자와 인증 서버의 이름을 의미한다.

상호 인증 과정중에서 랜덤 수(t)를 생성하고 공개키를 사용하여 사용자와 인증 서버간에 랜덤 수(t)를 교환하고, 랜덤 수와 long-term 키를 조합하여 통신에 사용하는 새로운 세션키(Knew_K)를 생성한다. 새로 생성된 세션키는 사용자가 패스워드를 입력하여 세션을 생성할때마다 랜덤수로 인해 통신에 사용되는 세션키 값은 바뀐다.

새로운 세션키 생성 과정은 다음 그림 3과 같다.



L = long-term 키 (64비트)
S = Et[L], 초기 벡터 IV = 0

그림 3. 새로운 세션키 생성 과정
Fig. 3. Process of new session-key generation

사용자로부터 패스워드를 입력받아 key-derived function을 통해 long-term 키를 생성한다. 이때 생성된 long-term 키와 랜덤 수(t)를 DES 알고리즘에 적용하여 새로운 세션키(Knew_K)를 생성한다.

제안하는 안전한 인증 알고리즘에서 사용자의 패스워드로부터 유도된 long-term 키를 사용하는 이유는 사용자를 인증하기 위함이다. 또한, 공개키와 개인키를 사용하여 랜덤 수와 전자 서명을 교환하여 공개키/개인키에 대한 인증과 사용자와 인증 서버에 대한 인증을 동시에 수행한다. 안전한 인증 알고리즘은 사용자와 단일 영역의 AS, 단일 영역 TGS와 다중 영역 TGS간에 각각 적용한다.

2. 안전한 허가 프로토콜

2.1 프록시 권한 서버

프록시 권한 서버(PPS: proxy privilege server)는 KDC안에 존재하고 TGS와만 통신한다. PPS는 사용자가 요청한 권한 요구에 대해서 허가 기능을 담당한다. 영역 내의 사용자, 응용 서버, 사용하려는 서비스에 대한 권한은 사전에 데이터 베이스에 등록되어 있어야 한다. PPS의 동작 과정은 그림 4와 같다.

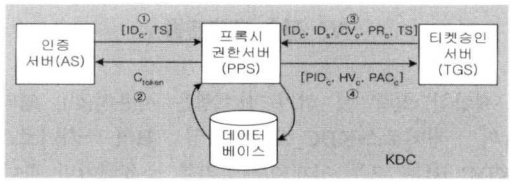


그림 4. PPS의 동작 과정
Fig. 4. Step of PPS processing

PPS의 동작 과정은 사용자가 티켓-승인 티켓(TGT: ticket-granting ticket) 발행을 요구할 때 사용자의 권한을 확인하고 토큰을 발행하는 절차(①, ②)와, 사용자가 응용 서버로의 접근을 위해서 서비스-승인 티켓(ST: service-granting ticket)을 요구할 때 권한 속성 인증서(PAC: privilege attribute certificate)를 발행하는 절차(③,④)로 구분한다.

2.1.1 토큰 발행

사용자가 TGT 발행을 요구할 때 PPS가 AS에게 발행하는 토큰의 내용은 다음과 같다.

$$Ctoken = [IDc \parallel GRPIDc \parallel TS]$$

토큰은 사용자의 이름(IDc), 사용자가 포함된 그룹의 이름(GRPIDc)과 토큰이 만들어진 시각을 표시한 타임 스템프(TS)를 포함한다. AS는 PPS에서 발행한 토큰을 TGT에 포함하여 사용자에게 전달하고, 사용자는 ST 발급을 위해 토큰이 포함된 TGT를 TGS에게 제공한다. 토큰은 TGT내에 포함되어 있으며 TGT는 AS내의 비밀키로 암호화되어 사용자가 토큰의 내용을 변조하거나 위조 할 수 없다.

2.1.2 권한 속성 인증서 발행

PPS는 PAC를 발급하기 위해서 요구자의 이름(IDc), 접근하고자 하는 서버나 서비스의 이름(IDs), 요구자들의 집합(CVc), 토큰의 내용이 포함된 권한 요구(PRc), 타임 스템프(TS)를 TGS로부터 입력받는다. PPS의 출력으로는 PAC의 이름(PIDc), 해쉬값(HVc), 권한 속성 인증서(PACc)를 TGS로 전송한다. PAC는 PAC의 이름(PIDc), 요구자들의 집합(CVc), 권한 요구(PRc)등을 포함한다. PIDc, HVc는 별도로 TGS로 전달되어 티켓에 포함하고, PACc는 커버로스 서버의 개인키Kprk_KDC로 암호화하여 사용자에게 전달한다.

$$PACc = EKprk\_KDC[PIDc \parallel CVc \parallel PRc]$$

PID는 PPS에서 랜덤 수를 생성하여 한 세션이 끝날 때까지 사용한다. CV(Control Value)는 요구자들의 집합으로 권한 위임(delegation)이 발생할 경우 각 요구자와 중개자들을 연결시키는 고리 역할을 한다.

$$CVc = [\{(IDc, IDs), ...\} \parallel \{PRc, ...\}]$$
$$PRc = [SVc \parallel GRPIDc]$$

PRc은 접근하려는 서버나 서비스에 대한 행동과 사용자의 소속 그룹 정보 등을 포함한다. PPS는 사용자가 요구하는 서비스(SVc)에 대해서 IDc나 GRPIDc를 자신이 관리하는 데이터 베이스에서 확인하고 권한 허가 여부를 판단한다.

HV(Hash Value)는 PAC를 해쉬 한 값이다. H는 해쉬 함수를 나타내고 HV는 ST에 포함되며 후에 PAC의 무결성을 확인하는데 사용한다.

$$HVc = H(PACc)$$

PPS의 PAC 발급 프로토콜에서 사용자가 접근하려는 응용 서버가 단일 영역에 존재하는지 다중 영역에 존재하는지를 사용자의 이름과 응용 서버의 이름을 통해서 확인한다. 응용 서버가 단일 영역에 존재하는 경우에는 사용자의 권한이 이용하려는 응용 서버의 서비스의 권한에 적합한지를 PPS가 관리하는 데이터 베이스에서 검색하여 유효성을 확인한다. 사용자의 권한이 유효하면 PAC를 작성하고 그렇지 않다면 권한 요구를 거부한다.

만일, 사용자가 접근하려는 응용 서버가 단일 영역 외에 존재하는 경우에는 ProxyPAC를 작성한다. ProxyPAC는 요구자의 권한 요구에 대해서 일단 허가하는 의미를 담고 있다. ProxyPAC는 다중 영역의 커버로스 시스템으로 전송되어 사용자가 접근하려는 외부의 영역에 있는 응용 서버와 서비스에 대해서 해당 영역의 PPS에게 한번 더 유효성을 확인한다. ProxyPAC를 사용하는 이유는 사용자가 속한 영역 외의 모든 영역에 있는 응용 서버나 서비스에 대한 권한을 PPS가 관리하고 저장할 수 없기 때문이다. 즉, PPS가 자신이 속한 영역에 대해서만 사용자, 응용 서버 및 서비스에 대한 권한을 관리하고 허가하기 위함이다.

2.2 권한 위임

권한 위임(delegation)은 PPS의 도움으로 수행하며 동작 과정은 그림 5와 같다.

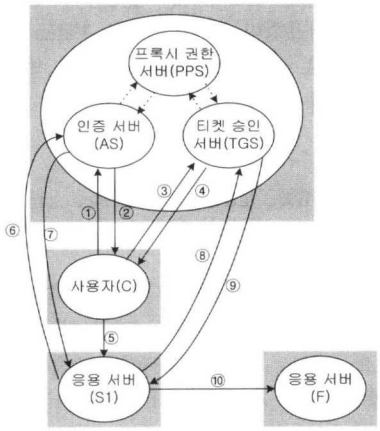


그림 5. 권한 위임 동작 과정
Fig. 5. Step of delegation processing

①~② : C에 대한 사용자 인증 단계로써 AS로부터 토큰이 포함된 TGT를 발급 받는다.

③~④ : 사용자는 S1에 접근하기 위해서 TGS에 TGT와 인증자를 제시하고 응용 서버 S1에 대한 ST를 발급 받는다. 이때, TGS의 개인키로 암호화된 사용자의 PAC를 데이터와 함께 발급 받는다. 사용자의 PAC는 ⑤와 같은 과정을 통해서 응용 서버 S1으로 전달된다. S1은 사용자가 원하는 작업을 진행한다.

⑥~⑦ : S1이 작업을 진행하는 중에 또 다른 응용 서버 F에 대한 접근을 해야 하는 경우가 발생하면, 응용 서버 S1은 커버로스 시스템에 인증 단계를 거쳐서 S1에 대한 토큰을 포함한 TGT를 발급 받는다.

⑧~⑨ : F에 대해서 사용자 C의 입장에서 F에 접근하기 위해서는 응용 서버 F에 대한 ST를 발급 받는다. 응용서버 S1이 응용 서버 F에 대해서 사용자 C의 입장에서 대신 작업을 수행하기 위해서는 ⑧에서 사용자 C로부터 받은 PACc와 서비스에 대한 행동 SV를 인증자에 포함해야 한다. TGS는 PPS에게 응용 서버 F로의 PAC를 발급 받은 후에 ST에 포함하여 ⑨와 같이 응용 서버 S1으로 전송한다. S1은 발급 받은 ST와 인증자를 생성하여 ⑩과 같이 권한 위임을 수행한다.

Ⅳ. 전송 프로토콜

제안한 인증 및 허가 시스템은 사용자(C), 영역 A의 커버로스(KDC_A), 영역 B의 커버로스(KDC_B)는 모두 공개키/개인키를 소유해야만 한다. 전체적인 동작 절차는 그림 6과 같다.

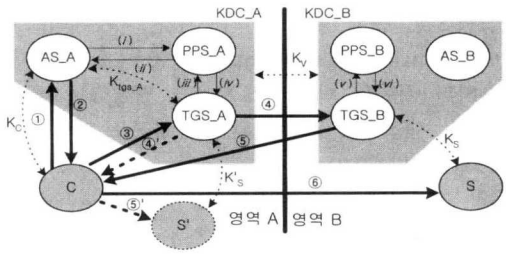


그림 6. 다중 영역에서 동작 프로토콜
Fig. 6. Protocol step in the multi-area

단일 영역에서의 동작은 ①-②-③'-④'-⑤'이며, 다중 영역에서의 동작은 ①-②-③-④-⑤-⑥이다. 제안한 안전한 인증 메커니즘의 동작 절차는 모두 3단계로 구성한다. 첫 번째 단계는 사용자 인증 단계(①~②)이고, 두 번째 단계는 서비스 승인 단계(단일영역:③~④', 다중영역:③~⑤)이며, 세 번째 단계는 응용 서버와의 데이터 전송 단계(단일영역:⑤', 다중영역:⑥)이다. 두 번째 단계를 모두 끝내면 사용자는 서비스-승인 티켓을 소유하고, 티켓 유효기간 내에 티켓을 사용하여 언제든지 응용 서버(단일영역:S', 다중영역:S)와 통신한다. 허가 메커니즘은 인증 메커니즘 수행 과정에 함께 포함된다.

- ADx : x의 IP 주소
- K_{puk_C}, K_{prk_C} : 사용자의 공개키/개인키
- CVx : PACx의 제어값
- HVx : CVx의 해쉬값
- ID_X : x의 이름
- $K_{puk\_KDC\_A}$, $K_{prk\_KDC\_A}$: 영역 A KDC의 공개키/개인키
- $K_{puk\_KDC\_B}$, $K_{prk\_KDC\_B}$: 영역 B KDC의 공개키/개인키
- $K_{a,b}$: a와 b간의 통신에 사용할 세션키
- K_a : a에서 사용할 비밀키
- Lx : PACx의 유효기간
- Lifetime : 유효 기간
- PRx : x의 권한 요구
- R_X : x에서 만든 랜덤 수
- TS : 타임 스템프

1. 표기법

전송 프로토콜에 사용할 표기법은 다음과 같다.

2. 프로토콜 단계

2.1 사용자 인증 단계

사용자 인증 단계는 사용자(C)와 영역 A의 KDC(AS_A)간의 인증 과정을 나타낸다.

① 사용자(C)와 지역 KDC(AS_A)간의 상호 인증 과정

- C → AS_A : IDc ‖ Rc
- AS_A → C : Kpuk_KDC_A ‖ Ras_A ‖ IDas_A ‖ Sig_asA
- C → AS_A : Kpuk_C ‖ IDc ‖ Sig_C ‖ IDtgs_A ‖ TS1 ‖ EKpuk_KDC_A[t1]
  - Sig_asA : EKprk_KDC_A[Rc ‖ Ras_A ‖ IDas_A]
  - Sig_C : EKprk_C[Ras_A ‖ Rc ‖ IDc]

Sig_asA는 영역 A에 속한 KDC의 전자 서명을 나타내고 Sig_C는 사용자의 전자 서명을 나타낸다. 제안하는 안전한 인증 프로토콜과 같이 공개키와 개인키를 사용하여 사용자 인증과 공개키에 대한 인증을 동시에 수행한다. 또한, AS_A에서 새로운 세션키를 생성하기 위한 랜덤 수(t1)는 AS_A의 공개키로 암호화하여 전송한다. Ras_A와 Rc는 사용자와 AS간의 상호 인증을 위해서 사용한 랜덤 수이다.

(i) AS_A → PPS_A : IDc ‖ TS1
(ii) PPS_A → AS_A : Ctoken
- Ctoken = IDc ‖ GRPIDc ‖ TS1

TGT 발행을 요청 받은 AS_A는 사용자 이름 IDc와 토큰 요구 시각을 나타내는 타임 스템프 TS1을 PPS_A에게 전달한다. PPS_A은 자신의 데이터 베이스에서 사용자에 대한 소속 그룹 정보 GRPIDc를 찾고 이를 포함한 토큰 Ctoken을 AS_A로 반환한다.

② AS_A → C : Tickettgs_A ‖ EK'c[Kc,tgs_A ‖ IDtgs_A ‖ TS2 ‖ Lifetime]
- Tickettgs_A = EKtgs_A[Kc,tgs_A ‖ IDc ‖ ADc ‖ IDtgs_A ‖ TS2 ‖ Lifetime ‖ Ctoken]

AS_A는 사용자가 KDC의 공개키로 암호화 한 랜덤 수(t1)를 개인키로 복호화하고 long-term 키와 조합하여 새로운 세션키(K'c)를 생성하여 사용자에게 전달할 데이터를 암호화한다. ③에서 사용할 세션키 Kc,tgs_A를 생성하여 사용자에게 전달할 데이터와 함께 티켓에 포함한다. 또한, AS_A는 PPS가 발행한 토큰 Ctoken을 티켓-승인 티켓(Tickettgs)에 포함하여 사용자에게 발급한다.

2.2 서비스 승인 단계

티켓-승인 티켓을 사용하여 서비스-승인 티켓을 발급 받는 과정이다. 단일 영역에서의 서비스 승인 단계는 ③-④'단계이고, 다중 영역에서의 서비스 승인 단계는 ③-④-⑤단계다.

③ C → TGS_A : IDs ‖ Tickettgs_A ‖ Authenticatorc_A
- Tickettgs_A = EKtgs_A[Kc,tgs_A ‖ IDc ‖ ADc ‖ IDtgs_A ‖ TS2 ‖ Lifetime ‖ Ctoken]
- Authenticatorc_A = EKc,tgs_A[IDc ‖ ADc ‖ TS3 ‖ SVc]

사용자는 서비스-승인 티켓을 신청한다. 응용 서버가 단일 영역에 있는 경우에는 IDs 대신에 IDs'로 바꾸어야 한다. 사용자는 새로운 세션키로 AS_A로부터 받은 데이터를 복호화하여 세션키 Kc,tgs_A를 얻은 후에 인증자를 생성하고 세션키 Kc,tgs_A로 암호화한다. AS_A에게 발급 받은 TGT와 생성한 인증자를 TGS_A에게 전송함으로써 ST를 신청한다. 인증자에는 사용자의 권한 요구 SVc 등을 포함한다.

(iii) TGS_A → PPS_A : IDc ‖ IDs ‖ CVc ‖ PRc ‖ TS3
(iv) PPS_A → TGS_A : PIDc ‖ HVc ‖ (PACc 또는 ProxyPACc)
- PRc = SVc ‖ GRPIDc
- PACc 또는 ProxyPACc = EKprk_KDC_A[PIDc ‖ CVc ‖ PRc]

(iii), (iv)는 PPS_A의 동작 절차이다. 단일 영역인 경우에는 PAC를 생성하기 위한 재료를 입력받고, PAC의 이름 PID, 권한 위임을 위한 제어값

CV 및 PR을 생성하여 PAC를 만든다. 다중 영역인 경우에는 IDs가 PPS_A 영역에 존재하지 않기 때문에 PPS_A는 ProxyPAC를 발급한다. ProxyPAC는 영역 B에 속한 PPS에서 다시 한번 권한 허가에 대한 유효성을 검사 받는다. PPS_B는 ProxyPAC의 이름(PID), 해쉬값(HV)과 제어값(CV)을 생성한다. PAC 또는 ProxyPAC는 KDC_A의 개인키 Kprk_KDC_A로 암호화한다.

④ TGS_A → C : IDc || Tickets' || EKc,tgs_A[Kc,s' || IDs' || TS4 || PACc]
- Tickets' = EK's[Kc,s' || IDc || IDs' || TS4 || Lifetime || Kpuk_KDC_A || PIDc || HVc]
- PACc = EKprk_KDC_A[PIDc || CVc || PRc]

④'은 단일 영역에서의 동작과정이다. PPS_A로부터 PIDc, HVc와 PACc를 받은 TGS_A는 KDC_A의 공개키 Kpuk_KDC_A, PIDc와 HVc를 티켓에 포함하여 long-term 키로 암호화하여 ST를 발급한다. PACc는 응용 서버와 통신할 때 사용할 세션키 Kc,s'과 함께 TGT로부터 얻은 비밀키 Kc,tgs_A로 암호화하여 사용자에게 전달한다. Tickets'는 서비스-승인 티켓이고 PACc는 권한 허가 인증서이다.

④ 다중 영역 영역에서 TGS_A와 TGS_B간의 상호 인증 과정
- TGS_A → TGS_B : IDtgs_B || Rtgs_A
- TGS_B → TGS_A : Kpuk_KDC_B || Rtgs_B || IDtgs_B || Sig_asB
- TGS_A → TGS_B : Kpuk_KDC_A || IDtgs_A ||Sig_asA ||EKpuk_KDC_B[t2]|| Tickettgs_A_B || Authenticatorc_A_B
  - Sig_asB : EKprk_KDC_B[Rtgs_A || Rtgs_B || IDtgs_B]
  - Sig_asA : EKprk_KDC_A[Rtgs_B || Rtgs_A || IDtgs_A]
  - Tickettgs_A_B = EK'v[Kc,tgs_A || IDc || ADc || IDtgs_A ||IDtgs_B || TS4 ||Lifetime || Kpuk_KDC_A || PIDc || HVc]
  - Authenticatorc_A_B = EKc,tgs_A[IDc || ADc ||IDtgs_A || TS4 || ProxyPACc]
  - ProxyPACc = EKprk_KDC_A[PIDc || CVc || PRc]

Sig_asB는 영역 B에 속한 TGS의 전자 서명이고, Sig_asA는 영역 A에 속한 TGS의 전자 서명이다. 그리고 Tickettgs_A_B은 TGS_B로부터 ST를 얻기 위한 TGT이다. 제안한 프로토콜과 같이 공개키와 개인키를 사용하여 사용자 인증과 공개키에 대한 상호 인증을 수행한다. 특히, KDC_B의 공개키로 암호화 한 랜덤 수(t2)를 전송하여 long-term 키와 조합하여 비밀키 K'v를 생성하여 티켓 Tickettgs_A_B을 암호화한다. PPS_A에서는 사용자가 접근하려는 응용 서버가 영역 B에 있음을 확인하고 ProxyPAC를 발행했다. 인증자에는 ProxyPAC를 포함하고, 티켓에는 ProxyPAC를 확인할 수 있는 공개키 TGS_A의 Kpuk_KDC_A와 무결성을 증명하는 정보인 PIDc, HVc가 포함된다.

티켓과 TGS_A의 인증자, 그리고 ProxyPAC를 받은 TGS_B는 자신의 PPS_B로부터 PR을 확인한다. PPS_B의 상세한 동작 프로토콜은 다음과 같다.

(v) TGS_B → PPS_B : IDc || IDs || CVc || PRc || TS5

(vi) PPS_B → TGS_B : PIDc || HVc,tgs_B || PACc,tgs_B
- PRc = SVc || GRPIDc
- PACc,tgs_B = EKprk_KDC_B[PIDc || CVc,tgs_B || PRc,tgs_B]

(v), (vi)는 PPS의 동작 절차이다. TGS_B로부터 PAC를 생성하기 위한 재료를 입력받아 PAC의 이름 PIDc, 해쉬값 HVc,tgs_B와 제어값 CVc,tgs_B를 생성하여 PACc,tgs_B를 만든다. PPS_B는 KDC_B의 개인키 Kprk_KDC_B로 PAC를 암호화하고 TGS_B로 전달한다.

⑤ TGS_B → C : IDc || Tickets || EKc,tgs_A[Kc,s || IDs || IDtgs_B || TS5 || Lifetime || PACc,tgs_B]
- Tickets = EKs[Kc,s || IDc || IDs || TS5 || Lifetime || Kpuk_KDC_B || PIDc,tgs_B || HVc,tgs_B]
- PACc,tgs_B = EKprk_KDC_B[PIDc,tgs_B ||CVc,tgs_B ||PRc,tgs_B]

영역 B에 속한 TGS는 사용자에게 직접 서비스-승인 티켓(Tickets)을 발급한다. PPS_B는 PPS_A로부터 받은 ProxyPACc에 대해서 유효성을 검사하고

PACc,tgs_B를 만들어 사용자에게 전송한다. 서비스-승인 티켓에는 PACc,tgs_B를 검사할 수 있는 TGS_B의 공개키 Kpuk_KDC_B와 무결성을 검사할 수 있는 정보인 PIDc,tgs_B, HVc,tgs_B를 포함한다. ⑥에서 사용할 세션키 Kc,s를 생성하여 사용자에게 전송할 데이터에 포함하고 세션키 Kc,tgs_A로 암호화한다.

2.3 데이터 전송 단계

응용 서버(단일영역: S', 다중영역: S)에게 서비스-승인 티켓과 인증자를 제출함으로써 별도의 인증 과정 없이 서버에 접근할 수 있다. 응용 서버는 PAC를 확인하여 사용자의 요구를 처리한다. 단일 영역에서의 데이터 전송 단계는 ⑤'단계이고, 다중 영역에서의 데이터 전송 단계는 ⑥단계이다.

⑤' C → S' : Tickets' || Authenticatorc2

- Tickets' = EKs'[Kc,s' || IDc || IDs' || TS4 || Lifetime || Kpuk_KDC_A || PIDc || HVc]
- Authenticatorc2 = EKc,s'[IDc || ADc || TS5 || PACc]
- PACc = EKprk_KDC_A[PIDc || CVc || PRc]

⑥ C → S : Tickets || Authenticatorc_s

- Tickets = EKs[Kc,s || IDc || IDs || TS5 || Lifetime || Kpuk_KDC_B || PIDc,tgs_B|| HVc,tgs_B]
- Authenticatorc_s = EKc,s[IDc || ADc || TS6 || PACc,tgs_B]
- PACc,tgs_B = EKprk_KDC_B[PIDc,tgs_B || CVc,tgs_B || PRc,tgs_B]

사용자는 세션키를 사용하여 TGS로부터 받은 데이터를 복호화하여 응용 서버와 사용할 세션키(단일영역:Kc,s', 다중영역:Kc,s)를 획득하고 인증자를 생성하여 Kc,s'또는 Kc,s로 암호화한다. 발급 받은 서비스-승인 티켓과 자신의 정보를 포함하는 인증자를 사용하여 응용 서버에 접근한다. 응용 서버는 long-term 키(단일영역:Ks', 다중영역:Ks)로 티켓을 복호화하여 사용자를 확인하고, PAC를 해석한 후 권한 요구를 처리한다.

V. 제안한 메커니즘의 성능 평가 및 비교 분석

1. 안전한 커버로스 시스템의 성능 평가

안전한 커버로스 시스템의 성능 평가를 위해서 기존 커버로스 시스템과 SESAME 시스템과의 티켓 발행 시간을 측정했다. 시간 측정은 3단계로 나누어 진행했다. 1단계는 티켓-승인 티켓을 발급 받는데 걸리는 시간을 측정했고, 2단계는 서비스-승인 티켓을 발급 받는데 걸리는 시간을 측정했다. 3단계는 응용 서버와의 데이터 교환시간을 측정했다.
다음 그림 7과 그림 8은 단일 영역과 다중 영역에서의 측정 결과이다.

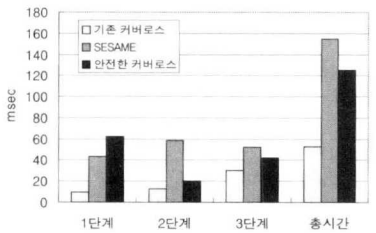


그림 7. 티켓 발행 시간 측정 그래프(단일 영역)
Fig. 7. Determination graph of ticket issuing time(single area)

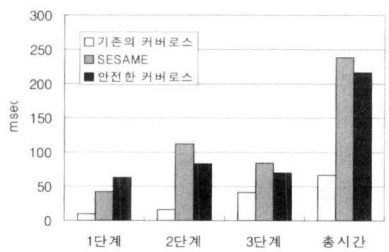


그림 8. 티켓 발행 시간 측정 그래프(다중 영역)
Fig. 8. Determination graph of ticket issuing time(multi-area)

단일 영역에서의 티켓 발행 시간 측정 결과에 의하면 1단계에서는 기존 커버로스와 많은 차이가 생긴 것을 알 수 있다. 그것은 공개키 암호화 알고리즘인 RSA가 대칭키 암호화 알고리즘인 DES에 비해서 암호화하고 복호화 하는데 상당히 많은 시간을 사용하기 때문이다. SESAME 시스템은 개선된 커버로스 시스템이 상호 인증 과정을 수행하기 때

문에 티켓-승인 티켓 발행시간에 있어서는 다소 빠르다.

2단계는 커버로스 시스템과 개선된 커버로스 시스템은 거의 같은 시간을 사용하고 있다. SESAME 시스템은 서비스-승인 티켓 발행에서 공개키를 사용하고 발행 절차도 더 복잡하기 때문에 개선된 커버로스 시스템보다 느린 것을 알 수 있다. 3단계는 각각의 시스템이 거의 같은 시간을 사용하는 것을 알 수 있다.

다중 영역에서의 티켓 발행 시간 측정 결과에 의하면 1단계는 단일 영역의 서비스 시간과 거의 비슷하다. 2단계는 개선된 커버로스 시스템이 영역간 TGS를 상호 인증 하기 위해서 공개키를 사용하므로써 서비스-승인 티켓을 발행하는 시간이 단일 영역에서보다 많이 소요되는 것을 알 수 있다. SESAME 시스템은 신뢰 센터로부터 공개키를 발급받고, 공개키를 사용하여 세션키를 교환하기 때문에 훨씬 더 많은 시간을 소요한다. 전체적으로 다중 영역에서의 티켓 발행 시간 측면에서 볼 때 SESAME 시스템보다 훨씬 좋은 성능을 나타낸다.

2. 안전한 커버로스 시스템과 비교 분석

제안하고 있는 안전한 인증 및 허가 시스템에 대한 연구는 기존의 커버로스 메커니즘을 최대한 활용할 수 있도록 개선하였다. 기존의 커버로스에 공개키/개인키 개념과 PPS를 두어 사용자 인증과 허가 기능을 추가했다. 기존 커버로스, SESAME와 제안한 메커니즘을 비교 분석한다.

기존 커버로스는 인증 메커니즘을 제공하고 허가 메커니즘에 대해서는 옵션사항으로 언급만을 하고 있다. SESAME와 제안하고 있는 메커니즘은 인증과 허가 메커니즘을 지원한다.

전체적인 전송 단계는 기존 커버로스인 경우 내부 영역인 경우는 5단계, 외부 영역인 경우는 7단계를 거쳐 수행한다. SESAME는 내부 영역인 경우 9단계, 외부 영역인 경우는 11단계이고, 제안하고 있는 메커니즘은 내부 영역은 커버로스와 같고 외부 영역인 경우 전송 단계를 6단계로 단순화하여 통신 비용을 줄였다.

공개키와 인증기관의 사용여부는 기존 커버로스인 경우 대칭키만을 사용했다. 그러므로 인증기관을 사용하지 않는다. SESAME는 공개키를 사용했으며 신뢰성 있는 인증기관을 가정하고 있다. 제안하고 있는 메커니즘은 공개키를 사용하고 자체 알고리즘을 사용하여 공개키와 사용자를 인증하므로 별도의 인증기관이 필요 없다.

서비스-승인 티켓을 얻기 위해서 기존 커버로스와 제안한 메커니즘은 4단계를 거친다. SESAME는 6단계를 거쳐 티켓을 얻는다.

인증과 허가를 위해서 사용하는 키의 개수에 있어서 기존 커버로스는 long-term 키(Kc, Kv, Ks, Ktgs) 4개, 세션키(Kc, Kc,tgs_A, Kc,tgs_B, Kc,s) 4개를 사용한다. SESAME는 long-term 키(LKxa, LKga, LKvg, LKvh, LKg, LKgh) 6개, 세션키(BKxg, BKxv, KIxyd, KCxyd) 4개, 공개키(KPUp, KPUg, KPUh) 3쌍과 인증기관의 공개키(KPUc) 1쌍을 가지고 있어야 한다. 제안한 메커니즘은 long-temr 키(Kc, Ks, Kv, Ktgs_A) 4개, 세션키(K'c, Kc,tgs_A, Kc,s) 3개, 각각의 시스템들은 공개키를 생성하여 소유하고 있어야 하기 때문에 3쌍(Kpuk_C, Kpuk_KDC_A, Kpuk_KDC_B)이 필요하다.

효율성 측면에서 볼 때 기존의 커버로스는 인증 기능만을 소개하고 있으며, 별도의 권한 서버 등이 없어 허가 기능은 없다. 또한, long-term 키의 계속적인 사용은 사전 공격을 당할 수 있으며, 만일 long-term 키가 노출되면 위장 공격 등 신뢰성에 치명적인 영향을 미친다. SESAME는 인증 및 허가 기능을 모두 구현하고 있다. 그러나 인증 및 허가 기능을 수행하기 위해서 추가되는 자료 구조가 많아져서 통신의 비용을 높인다. 또한, 공개키를 사용하고 공개키에 대한 인증을 위해서 신뢰할 수 있는 인증기관이 필요하다. 기존 커버로스와 같이 long-term 키를 계속 사용하고 있어 사전 공격과 위장 공격에 취약하다. 그리고 허가 메커니즘을 사용하기 위해 권한 서버를 두었으며 모든 영역의 권한에 대해서 권한 서버가 관리해야 하므로 권한 서버의 부담이 크다. 제안한 메커니즘은 기존 커버로스의 전송 단계를 줄이고 자료 구조를 그대로 사용하여 인증 및 허가 메커니즘을 설계했다. 공개키를 사용할 때 공개키를 인증 할 인증기관 없이 자체의 알고리즘에 의해서 사용자와 공개키를 인증하여 사용한다. 시스템간의 통신에 사용하는 세션키를 매번 바꾸어주기 때문에 사전공격과 위장 공격으로부터 안전하다. 허가 메커니즘을 위해서 권한 서버를 사용한다. 권한 서버는 단일 영역의 권한에 대해서만 관리하고, 다중 영역에 대해서는 ProxyPAC 기능을 두어 처리한다.

제안하고 있는 안전한 커버로스 시스템은 공개키를 생성하고 사용하기 때문에 기존의 커버로스 시

표 1. 메커니즘 비교 분석
Table 1. Comparative table of mechanism analysis

분석 / 구분	기존의 커버로스	SESAME	안전한 인증 메커니즘
인증/허가 과정 유무	▸ 있음/없음	▸ 있음/있음	▸ 있음/있음
전송 단계 (단일/다중영역)	▸ 5단계/7단계	▸ 9단계/11단계	▸ 5단계/6단계
공개키/인증기관	▸ 사용안함/사용안함	▸ 사용함/사용함	▸ 사용함/사용안함
티켓 발급 절차	▸ C ⇄ AS, C ⇄ TGS (4 단계)	▸ C ⇄ AS, C ⇄ PAS, C ⇄ KDS (6 단계)	▸ C ⇄ AS, C ⇄ TGS (4 단계)
키 관리	▸ long-term 키 : 4개 ▸ 세션키 : 4개	▸ long-term 키 : 6개 ▸ 세션키 : 4개 ▸ 공개키 : 3쌍 ▸ 인증기관 공개키 : 1쌍	▸ long-term 키 : 4개 ▸ 세션키 : 3개 ▸ 공개키 : 3쌍
효율성	▸ 인증기능만 구현, 허가 기능 없음 ▸ 사전공격과 위장공격에 취약함 ▸ 권한 서버 없음	▸ 인증/허가 기능 구현 위해 부가적인 전송 자료구조 사용 ▸ 공개키를 위한 인증기관을 전제로 비밀성, 기밀성, 신뢰성, 경로추적성, 부인봉쇄기능 수행 ▸ 사전공격/위장 공격에 취약 ▸ 권한 서버의 부담이 크다.	▸ 부가적인 자료구조 사용하지 않고 인증/허가기능 구현함 ▸ 별도의 인증기관 없이 공개키를 활용하여 비밀성, 기밀성, 신뢰성, 경로추적성, 부인봉쇄기능 수행 ▸ 사전공격/위장공격 강함 ▸ 권한 서버의 부담이 적다 (ProxyPAC기능 추가)

스템에 비해서 서버에 부담이 있고 처리 속도가 늦다. 그러나 인증 및 허가 프로토콜을 수행하는 다른 시스템에 비해서 우수한 성능을 나타내고 있다.
표 1은 기존 커버로스, SESAME와 제안한 메커니즘을 비교 분석한 것이다.

VI. 결 론

본 논문에서는 분산 환경에서의 대표적인 인증 시스템인 커버로스와 SESAME를 분석하고 기존의 커버로스에 공개키/개인키 개념과 프록시 권한 서버(PPS: proxy privilege server)를 두어 인증 및 허가 메커니즘을 수행하는 시스템을 제안했다. 기존의 커버로스와 SESAME는 사용자 인증을 위해서 대칭키를 사용하기 때문에 사전 공격등 보안에 취약한 부분이 발생하였다. SESAME는 허가 메커니즘을 수행하기 위해 공개키와 인증기관을 사용하였다. 제안한 인증 메커니즘은 공개키/개인키를 사용하여 제안한 알고리즘에 의해 키에 대한 인증과 사용자에 대한 인증을 시도하여 인증기관 없이 공개키를 사용했다. long-term 키와 공개키를 사용하여 전송한 랜덤 수를 사용하여 세션마다 새로운 세션키를 만들어 데이터를 암호화하기 때문에 사전 공격 등에 강하다. 사용자 인증을 위한 전체적인 전송 단계를 단순화하여 통신비용을 줄였다. 허가 메커니즘을 위해서 PPS를 두었으며 자신이 속한 영역내의 권한만을 관리하도록 했다. 외부 영역에 대해서는 ProxyPAC를 발행하여 PPS의 부담을 줄였다.

본 논문은 기존의 커버로스에 공개키/개인키를 적용하고 프록시 권한 서버를 두어 효율적이고 안전한 인증 및 허가 메커니즘을 설계하였으며 앞으로 실제 구현하여 소규모 네트워크 등에서 활용할 수 있을 것이다.

참 고 문 헌

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

[2] 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.

[3] 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.

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

[5] John T. Kohl, B. Clifford Neuman, Theodore Y. T'so, "The Evolution of the Kerberos Authentication System In Distributed Open Systems," IEEE Computer Society Press, pp. 78-94., 1994.

[6] Marvin A. Sirbu, John Chung-I Chuang, "Distributed Authentication in Kerberos Using Public Key Cryptography," Proc. 1997 Symposium on Network and Distributed System Security, 1997.

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

[8] Alfred J. Menezes, Paul C.van Oorschot, Scott A. Vanstone, Handbook of applied Cryptography, CRC Press, 1997.

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

[10] 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.

[11] 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.

[12] 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.

[13] 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.

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

[15] 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.

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

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

[18] 김철현, 정일용, "PKINIT 기반 새로운 커브로스 인증 메커니즘의 설계," 정보과학회 논문지, 제28권 제1호, Mar 2001.

김 은 환(Eun-hwan Kim) 정회원

1990년 2월 : 숭실대학교 전자계산학과 졸업
1997년 8월 : 숭실대학교 컴퓨터학과 석사
2003년 2월 : 숭실대학교 컴퓨터학과 박사
1990년 3월~1995년 8월 : 국방과학 연구소 연구원
1997년 9월~현재 : 숭실대학교 전산원 전임교수

<관심분야> 정보보호, 암호 알고리즘, 네트워크 및 인터넷 보안

김 명 희(Myung-hee Kim) 정회원

1995년 2월 : 경남대학교 전산통계학과 졸업
1998년 2월 : 창원대학교 전자계산학과 석사
1999년 9월~현재 : 숭실대학교 컴퓨터학과 박사과정
2001년 12월~현재 : 안철수 연구소 주임연구원

<관심분야> 암호 알고리즘, 정보이론, 인터넷 보안

전 문 석(Moon-seog Jun) 정회원

1980년 2월 : 숭실대학교 전자계산학과 졸업
1986년 2월 : University of Maryland 전산과 석사
1989년 2월 : University of Maryland 전산과 박사
1989년 3월 ~ 1991년 2월 : Morgan State University 부설 Physical Science Lab. 책임연구원
1991년 3월~현재 : 숭실대학교 정보과학대학 정교수

<관심분야> 네트워크 보안, 컴퓨터 알고리즘, 병렬처리, 암호학