通信情報保護學會誌
第 7 卷 第 4 號, 1997. 12

분산 환경에서의 정보보호 서버 : SESAME

A Security Server in a Distributed Environment : SESAME

이 정 현, 윤 이 중, 이 대 기, 김 대 호*

요 약

본 논문에서는 분산 컴퓨팅환경에서 자원을 보호하기 위한 기술적인 해결책으로서 정보보호 시스템 하부구조인 유럽의 SESAME 시스템을 분석하였다. 이 시스템에서는 Kerberos, GSS-API, 대칭키 및 비대칭키 인증 메카니즘 등의 정보보호 기반기술을 활용하여 인증 및 키 분배, 접근제어 등의 정보보호 서비스를 특정 응용 시스템 또는 특정 프로토콜에 의존하지 않고 어떠한 시스템 환경에서도 서비스 제공이 가능한 시스템 구조를 제시하였다. 향후 정보보호 응용시스템의 설계시 이러한 SESAME와 유사한 시스템 설계 방식을 취한 시스템을 많이 볼 수 있게 될 것으로 전망된다.

1. 서 론

SESAME(Secure European System for Application in a Multivendor Environment)는 CEC(Commission of the European Communities)에서 ECMA(European Computer Manufacturer Association) 초창기 작업에 의해 시작된 연구개발 프로젝트이다. 이 프로젝트의 주 목적은 개방형 분산 시스템에서 자원을 보호하기 위한 기술적인 해결책으로서 대칭키(symmetric key)와 비대칭키(asymmetric key) 암호 기법을 모두 사용하여 단일 사인-온(single sign-on)과 접근 제어(access control) 기술을 개발하는데 있었다. 즉, 시스템 개발자들을 위해 정보보호 하부구조(security infrastructure)로서 일종의 빌딩 블럭으로 제공하고자 하였다. 이렇게 함으로써 서로 다른 업체들이 상용 제품을 만들고자 할 때, 핵심적인 정보보호 기능들을 별도의 변경없이 그대로 사용할 수 있도록 제공하자는 것이다.

이러한 하부구조로서의 정보보호 기술을 제공하기 위한 SESAME의 주된 특징은 다음과 같다. 첫째로는 분산 환경에서 어떤 시스템과도 연동될 수 있도록 하기 위하여 현재 광범위하게 사용되고 있는 GSS-API(Generic Security Service - Application Programming Interface)[3][4]를 사용한 인터페이스 방식을 채택하였다. 둘째, 동일 도메인내에서의 세션키(session key) 전달을 위해 Kerberos와 같은 패스워드 인증기법을 도입하였고, 세째, 서로 다른 도메인간에 키 분배을 위해 ITU-T X.509

* 한국전자통신연구원

의 공개키 인증기법[6]을 적용하였다. 네째, PAC(Privilege Attribute Certificate)라는 개인의 신원정보들의 집합인 특성 어트리뷰트(Privilege Attribute)들에 대한 보증서를 이용하여 분산 환경에서의 접근제어 서비스를 제공한다. 마지막으로, 가능한한 Kerberos, DES, RSA, MD5등과 같은 기존의 메카니즘들을 부분 수정 혹은 그대로 채택하였고, 또한 언제든지 시스템 개발그룹에서 이러한 메카니즘들을 자신의 독자적인 메카니즘으로 교체가능토록 설계하였다.

2. 배 경

1961년에 유럽에서는 정보 및 통신 시스템의 표준화를 위하여 ECMA를 설립하였다. OSI 정보보호 아키텍쳐(security architecture)[5]와 ISO/IEC JTC1에서 개발한 정보보호 프레임워크(security framework)에 기반하여 ECMA는 1988년에 ECMA TR/46이라는 기술보고서[1]를 발간하였고, 최근에는 분산 정보보호 서비스를 위한 기능성(functionality)과 프로토콜들을 정의하는 작업들을 완료하였다. 그 결과 ECMA-219 표준[2]이 1994년 12월에 ECMA 총회에서 인준받았고 공식적으로는 1995년 1월에 배포되었다.

SESAME는 RACE(Research and Development in Advanced Communications for Europe) 프로젝트의 일부(RACE R2051)로서 CEC로 부터 일부 지원받고 Bull SA, ICL PLC, SNI AG, SSE Ltd에서 공동 개발한 프로젝트이다.

이 프로젝트는 원래 2단계로 출발하였다. 첫번째 단계에서는 앞서 언급한 ECMA에서 개발 아키텍쳐 개념(architectural idea)과 원칙들이 실현가능한 지를 보여주기 위한 데모 시스템을 구현하고자 하였다. 이 작업은 1991년에 완료되었고, 이 프로토타입(prototype) 구현이 버전 1 시스템이 되었다. 두번째 단계에서는 상업용 정보보호 제품 개발에 사용할 수 있는 정보보호 컴포넌트들을 개발하는데 역점을 두었다. 이 과정에서의 중간 단계의 버젼들이 버젼 2(1994.7.) 버젼 3(1995.6.)으로 불려지게 되었고, 현재 버젼 4(1995.12.)가 나와 있다.

버젼 1시스템에서는 ECMA 정보보호 프레임워크를 준수하기 위하여 ECMA 인증 서비스의 구현에 촛점을 두었지만, 두번째 단계부터는 좀더 간편한 이식성(portability)을 제공하기 위하여 이미 개발되어 있는 Kerberos V5 인증 프로토콜을 채택하였고, 또한 서로 다른 도메인 간에 키 전달을 위해 X.509 공개키 인증기법을 도입하였다.

1992년에는 OSF(Open Software Foundation)의 DCE(Disttibuted Computing Environment) 환경에서 SESAME 기술이 사용될 수 있도록 상당한 노력을 하기도 하였었다. 그래서 그해 12월에 RFC를 OSF에 제출하였으나 DCE 버젼 1.1[7]에는 포함되지 않았다. 그후로 SESAME는 독자적으로 계속 개발되었지만, 이 시스템은 근본적으로 DCE 환경에서 적합하도록 개발되었고, 독립적으로 뿐만아니라 DCE와 통합될 수 있도록 설계되어 있다.

현재 SESAME V4에서 지원하는 플랫폼(platform)들을 [표 1]에 요약하였다. 표에서 알수 있듯이 UNIX 환경에서는 클라이언트와 서버 소프트웨어가 모두 구동가능하지만, MS-DOS/Windows하에 동작하는 PC에서는 클라이언트만 사용가능하고, 또한, CA(Certification Authority) 시스템의 경우에는 PC에서만 동작한다.

SESAME 기술은 전세계적으로 사용할수 있도록 개발되었기 때문에 소스코드[9]가 세계 각국에 수출될 수 있는 형태로 제공된다. 제각기 다른 각국의 수출입법과 이에 준하는 규제을 준수하기 위하여 SESAME는 모든 암호 알고리즘과 해쉬 함수들을 교체할 수 있도록 설계

[표 1] SESAME V4 지원 플랫폼

플랫폼	Client	Server	CA
Bull DPX 20 (AIX 3.2)	X	X	
SNI MX300I (SINIX(SVR4))	X	X	
ICL DRS6000 (SVR4)	X	X	
IBM RS6000 (AIX 3.2)	X	X	
Sun OS	X	X	
PC (MS-DOS/Windows)	X		X

하였다.

SESAME관련 문서들[9]은 개발자 당사자인 Bull, ICL, SNI, SSE의 공동 저작권으로 보호되어 있다. 그러나, 많은 사람들이 관심을 가지는 대부분의 문서들은 공개되어 있다. 특히, SESAME의 기능 명세에 관한 문서들은 SESAME 프로젝트 관리 위원회에서 비밀 문서로 취급하지 않았기 때문에 언제든지 사용가능하며, 이는 관련 표준화 활동에 기여하기 위한 것 같다.

3. 시스템 모델

이미 언급했듯이 SESAME의 주 목적은 개방 분산 환경에서 단일 사인-온과 접근제어를 지원하는 기술을 개발하자는 것이다. 단일 사인-온과 접근제어 기술 모두 인증 메카니즘이 필요하고, 또한 역으로 이 인증 메카니즘은 데이타 기밀성(data confidentiality)과 데이타 무결성(data integrity) 서비스를 제공하는데 필요한 키를 분배해 줄 수 있어야 한다. 결론적으로 SESAME가 제공하는 서비스에는 인증서비스, 접근제어 서비스, 데이타 기밀성 서비스, 데이타 무결성 서비스가 있다. 이러한 서비스들은 확장 Kerberos V5 인증(extended Kerberos V5 authenticatoin) 메카니즘과 ECMA형 특성 어트리뷰트 (ECMA-style privilege attribute) 메카니즘을 내장 혹은 조합함으로써 제공가능하다.

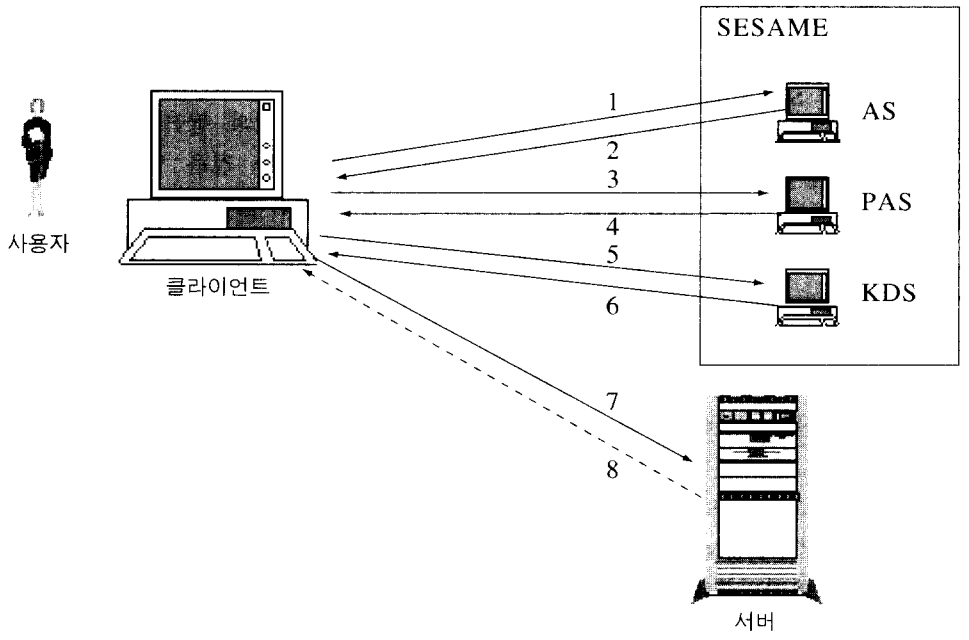
확장 Kerberos V5 인증 메카니즘은 패스워드 방식에 기반한 Kerberos V5의 인증 메카니즘 뿐만아니라 공개키 암호기법에 기반한 인증 메카니즘을 함께 지원한다. ECMA형 특성 어트리뷰트 메카니즘은 역할기반 접근제어(role-based access control) 정책을 구현하는데 사용된다. 역할기반 접근제어 정책을 고려한 기본 생각에는 일반적으로 조직에서 사용자들은 직책 또는 직종에 따라 시스템 접근 권한(access right)이 다르다는 사실에 기반하였다. 이러한 접근권한을 특성 어트리뷰트 또는 특성이라고 부르고, 각 사용자들에게 적용될 특정 역할 및 제어와 관련한 특성 어트리뷰트의 집합을 결정하는 것은 일반적으로 시스템 관리자의 몫이다. 게다가 시스템 관리자는 또한 어떤 사용자가 어떤 역할을 수행할 수 있는지를 알아야만 한다. 사용자는 자신에게 관련 특성들이 주어지는 자신의 역할을 초기 세션(session) 설정시 기술하여 주면, 이후에는 자동적으로 접근가능하게 된다.

기본적인 Kerberos 모델에서, 클라이언트는 AS(Authentication Server)로 부터 TGT (Ticket Granting Ticket)를 요구하고, 이 TGT를 TGS(Ticket Granting Server)서버로 부터 서비스 티켓(Service Ticket)을 요구하기 위하

여 사용한다. SESAME 모델에서도 인증 및 접근제어를 위해 유사한 접근방법이 사용된다. 클라이언트의 서비스 요구가 발생하면 AS로부터 인증을 받아야만 하는 것 뿐만아니라 PAS(Privilege Attribute Server)서버에 의해 보증된 자신의 특성 어트리뷰트를 포함하고 있는 PAC(Privilege Attribute Certificate)를 가지고 있어야만 한다. PAC는 사용자의 특성 정보와 제어 정보들로 이루어져 있다. 사용자 특성 정보는 사용자의 신원(identity), 역할, 조직 그룹, 보안 등급 등과 같은 데이타인 반면, 제어 정보는 PAC가 언제 어디서 사용될 것인지 및 대상 시스템이 대표성을 가지는지 아닌지를 나타내는 데이타이다. PAC는 개념적으로 X.509에 기술되어 있는 보증서에 접근제어용 데이타 필트를 추가한 것이다. SESAME 모델에서 PAS는 인증 증거로서 PAC를 생성하고 이 PAC는 해당 PAS의 개인키(private key)로 서명된다. 한편, SESAME에서는 X.509 형태를 취한 두개의 보증서를 사용하는데, 하나는 도메인간의 키분배를 위한 디렉토리 보증서(Directory Certificate)이고 또다른 하나가 PAC이다.

SESAME의 개념 모델을 나타낸 것이 [그림 1]이다. 그림에서 알수 있듯이 SESAME 모델은 Kerberos 모델과 아주 흡사하다. 단순한 차이점은 PAS가 하나 더 추가되었다는 것과 Kerberos에서의 TGS가 SESAME에서 KDS (Key Distribution Server)로 이름이 바뀌었다는 것이다.

간단하게 설명하면, 우선 클라이언트는 확장 Kerberos V5 AS로부터 PAS 티켓을 요구한다. 클라이언트가 PAS 티켓을 얻었다면, 클라이언트는 이 티켓을 PAS로 부터 PAC 및 KDS 티켓을 요구하기 위해 사용한다. 그다음 클라이언트는 서버의 응용프로그램용 서비스 티켓과 티켓 발급에 필요한 키관련 정보(Keying information)를 얻기 위하여 PAS로 부터 받은 PAC와 KDS 티켓을 사용한다. 마지막으로 클라이언트는 서버에게 인증을 요구하게 되고, 만약 상호 인증(mutual authentication)이 요구되었을 경우에는 서버도 클라이언트에게 인증을 요구하게 된다.



[그림 1] SESAME의 개념 모델

클라이언트와 정보보호 서버 및 응용프로그램 서버사이에 메세지 교환을 위한 프로토콜 단계들은 [표 2]에 정리되어 있다.

[표 2] SESAME 프로토콜 단계

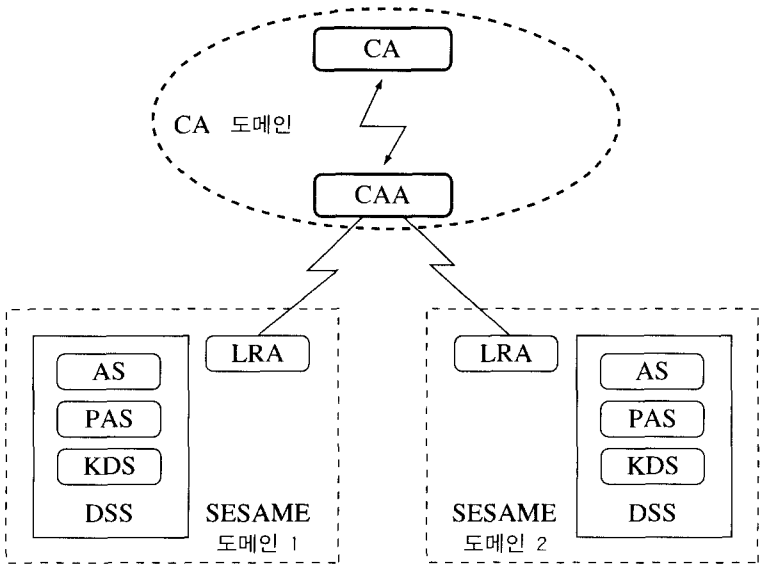
단계	메세지
1	KRB_AS_REQ
2	KRB_AS_REP
3	KRB_PAS_REQ
4	KRB_PAS_REP
5	KRB_TGS_REQ
6	KRB_TGS_REP
7	SES_INIT_CTXT
8	SES_INIT_CTXT_COMPLETE

7단계 또는 8단계 후에 클라이언트와 응용프로그램 서버사이에 보안 문맥(security context)이 설정된다. 이 이후부터 클라이언트와 응용프로그램 서버는 정보보호 기술이 적용된 SES_DATA 메세지를 송수신할 수 있게 된다.

실제 SESAME 구조는 [그림 1]의 개념 모델보다 더 복잡하다. SESAME 도메인에서 AS, PAS, KDS는 하나의 특정 호스트에서 동작하는 전형적인 프로세스들이고 이 호스트를 DSS (Domain Security Server) 서버라고 부른다. 또한 DSS에는 모든 정보보호 관련 데이타를 저장하고 있는 SMIB (Security Management Information Base)와, 사용자가 CA로 부터 통신상대자의 디렉토리 보증서의 송수신을 담당하는 LRA(Local Registration Authority) 서버를 가지고 있어야 한다.

현재 CA는 자기 자신의 도메인내에서만 동작하고, 이 도메인은 다른 도메인과의 연결이 제공되지 않는다. 또한 CA는 오프라인(off-line)으로 존재하며 네트워크으로부터 직접 액세스할 수 없다. 그 대신에 CA 도메인내에 있는 오프라인 CAA(Certificate Authority Agent) 서버를 통해 온라인(on-line) LRA와 오프라인 CA 사이의 연결이 가능하도록 되어 있다. CA의 관점에서 보면 CAA 서버는 CA에게 데이타를 송수신하는 기능을 제공하고, LRA의 관점에서 보면 CAA 서버는 CA에게 공개키 보증서 생성 요구들을 임시 저장 관리하는 스풀러(spooler)의 기능과 CA에 의해 보증된 키쌍을 클라이언트에게 제공하기 위한 에이전트(agent) 기능을 제공한다. 특히 CA와 CAA 서버는 데이타를 교환하기 위하여 임의의 화일 전송 메카니즘을 사용할 수 있는데, 화일 전송은 CAA 서버와 CA 시스템 모두가 사용가능한 플로피디스크(floppy disk)같은 매체를 사용하여 제공한다.

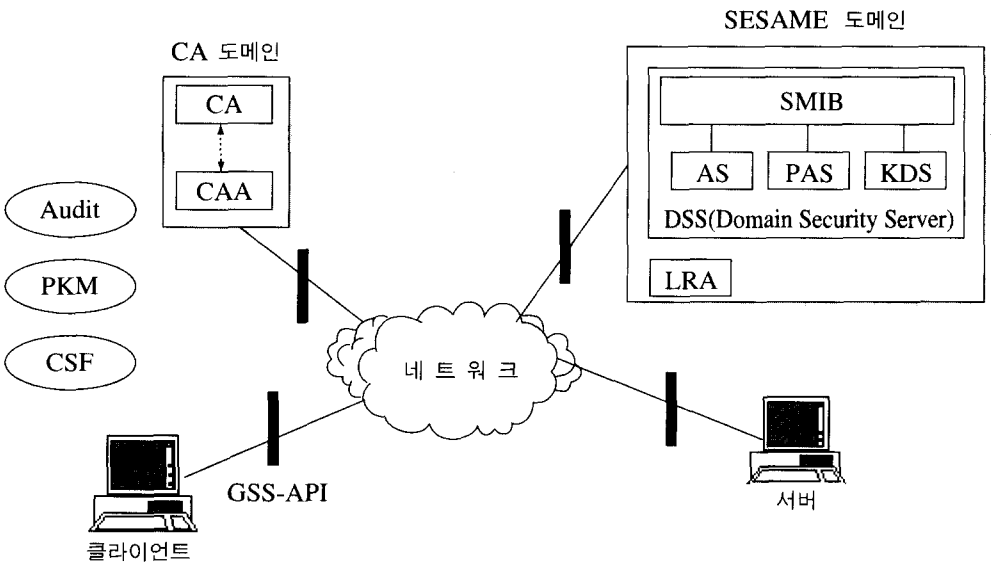
[그림 2]는 서로 다른 정보보호 서버와 정보보호 도메인과의 관계를 나타낸 것이다. 그림에서 위의 타원 점선은 CA 도메인을 표현한 것이고 아래의 두개의 사각형 점선은 두개의 SESAME 정보보호 도메인을 나타낸다. CA 도메인은 오프라인 CA, 온라인 CAA 서버와 그들사이에는 오프라인으로 테이타를 교환한다. SESAME 도메인은 DSS와 LRA 서버로 이루어져 있고, DSS는 실제로 AS, PAS, KDS로



[그림 2] 정보보호 서버와 정보보호 도메인

이루어져 있으며, LRA와 CA 도메인내의 CAA와의 통신은 온라인으로 제공된다.

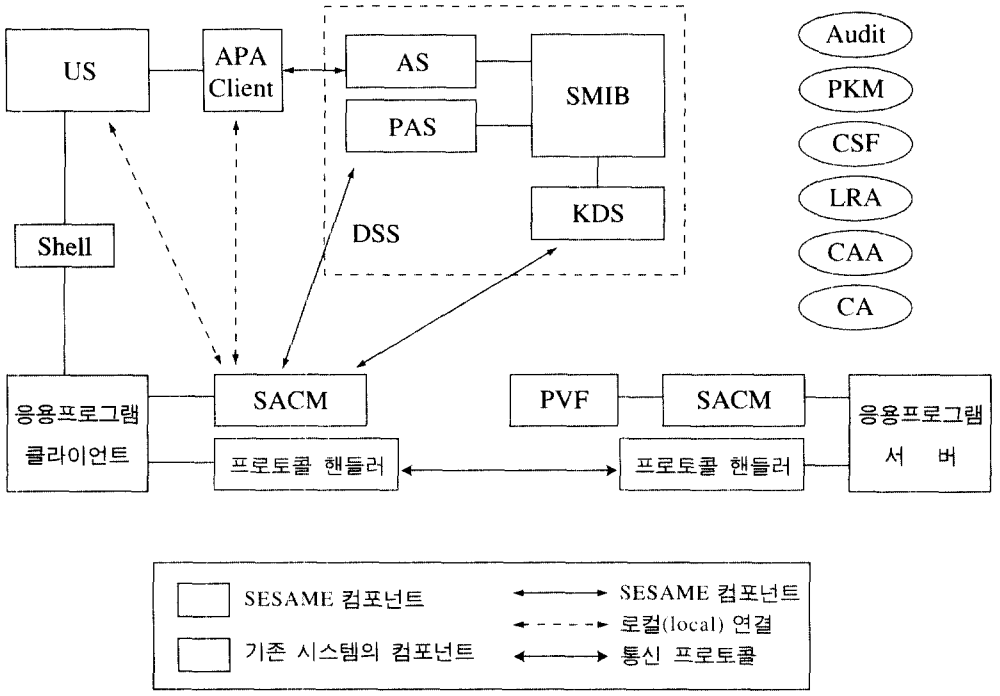
네트워크 응용프로그램의 관점으로 살펴본 시스템 구성이 [그림 3]이다. 정보보호 서비스를 제공하기 위하여 클라이언트와 서버를 제외한 나머지 컴포넌트들이 SESAME 시스템의 구성요소들이다. 크게는 CA 도메인내에 존재하는 CA와 CAA 서버와, SESAME 도메인내의 DSS 서버, LRA 서버 및 별도의 감사추적을 담당하는 Audit 모듈, 공개키의 관리를 담당하는 PKM 모듈, 암호알고리즘의 제공을 담당하는 CSF 모듈로 이루어져 있다. 또한 각 시스템간에 상호 인터페이스를 위해 GSS-API를 사용한다.



[그림 3] SESAME 시스템의 구성

4. 시스템 구성요소

클라이언트/서버 시스템상에서의 SESAME 구성요소를 나타낸 것이 [그림 4]이다. 먼저 왼쪽에 있는 응용프로그램 클라이언트가 오른쪽에 있는 응용프로그램 서버와 인증한다고 가정하자. 클라이언트 및 서버 시스템 모두 (반드시 그렇지는 않지만) 동일한 도메인내에 등록되어 있다고 가정한다.



[그림 4] SESAME 시스템 구조

SESAME 모델은 여러 컴포넌트들로 이루어져 있는데, 우선 클라이언트에 설치되어야 하는 컴포넌트들은 다음과 같다.

US(User Sponsor)는 시스템에 대한 사용자 인터페이스를 제공한다. SESAME V4에서는 단순한 명령형 라인(command line) 방식의 US가 제공된다. 이것은 사용자에게 로그인(seslogin), 특성 및 제어 어트리뷰트 변경(chattr), 로그 아웃(seslogout) 기능을 제공한다. 또한, 이 US는 개발업체가 만든 특정 인터페이스로 교체할 수 있도록 되어 있다.

APA(Application and Privilege Attribute) 클라이언트는 US가 초기 인증 및 디폴트(default) PAC 획득에 관한 세부사항들을 알 필요없이 DSS와 연결이 편리하도록 US에게 인터페이스 역할을 하는 컴포넌트이다.

SACM(Security Association Context Manager)는 특정 응용프로그램의 클라이언트/서버간에 보안 문맥을 설정, 관리하는 기능을 담당한다. 클라이언트 SACM은 클라이언트 시스템의 SACM 컴포넌트를 의미한다.

SACM 컴포넌트는 서버 시스템에도 설치되어 있어야만 하고, 이를 서버 SACM이라 부른다. 서버 SACM은 PVF(PAC Validation Facility)로 부터 지원을 받는다. 좀더 구체적으로 설명하면, PVF는 PAC의 유효성을 검사

하고 발신자와 안전한 통신을 하는데 필요한 대화키(dialogue key)들을 결과값으로 돌려준다. 이를 위해서 PVF는 PAS의 공개키를 알아야 하고 또한 DSS와의 비밀키(secret key)를 사전에 공유하고 있어야 한다.

클라이언트 또는 서버 시스템에 설치된 이러한 컴포넌트외에 별도로 필요한 3가지 일반적인 컴포넌트들이 필요한데 다음과 같다.

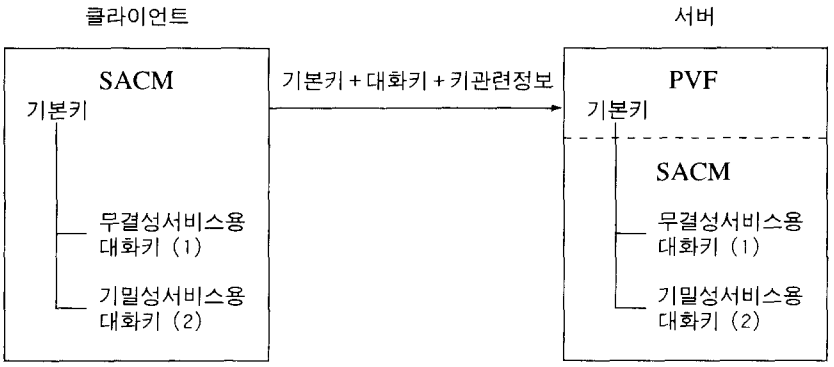
PKM(Public Key Management)은 예를들면, 공개키 암호시스템의 키 생성 및 취소시에 사용될 수 있는 관리 도구(tool)들 뿐만 아니라 다양한 보증서 및 키관리 함수들에 대한 액세스를 제공하기 위한 라이브러리들로 이루어져 있다.

CSF(Cryptographic Support Facility)는 여러 컴포넌트 또는 응용프로그램들에 의해 사용되는 암호알고리즘들을 제공해 주는 컴포넌트이다. 현재 SESAME V4에서 구현되어 사용하고 있는 알고리즘들은 DES-CBC, RSA, MD5, DES-MD5이다. 그러나 CSF는 향후 더욱 성능이 개선된 알고리즘들이 나타날 경우 언제든지 쉽게 교체할 수 있고 각국의 법령에 따라 키 크기 등도 언제든지 조정가능하도록 설계되어 있다. 또한 수출통제 등의 문제로 인하여 SESAME V4의 공개 버젼은 단순한 XOR 함수만 사용하여 모든 알고리즘의 기능을 처리하였으며, 상업용 버젼에만 DES와 별도의 비밀키 암호 시스템을 사용하였다.

감사사건(Audit Event)들은 시스템에서 발생하는 정보보호 관련 행동들이기때문에 사후의 분석을 위해 잘 보관해 둘 필요가 있다. 감사추적 모듈은 사용자 자신의 신원확인하에서 동작하는 데몬(daemon) 프로세스에 쓰기 전용 파이프(write-only pipe)를 사용하여 감사 사건들을 로깅(logging)한후, 해당 화일에 감사 정보를 저장한다. 이렇게하여 감사 화일은 응용 프로세스에 의한 변조로부터 보호될 수 있다. SESAME에서 감사 행위가 발생하는 컴포넌트에는 KDS, PVF, 발신자 및 수신자 SACM, APA클라이언트가 있다. 그러나, 이 감사 정보들의 검색 및 분석 기능은 현재 제공하고 못하고 있다.

5. 기본키와 대화키

SESAME는 기본 키(basic key)와 대화키(dialogue key)로 이루어진 2단계의 키 구조를 사용한다. [그림 5]는 기본키와 대화키의 관계를 나타낸 것이다.



(1) 데이터 무결성 서비스를 제공하기 위해 기본키로부터 파생된다.
(2) 데이터 기밀성 서비스를 제공하기 위해 기본키로부터 파생된다.

[그림 5] 기본키와 대화키의 관계

- 기본 키는 PAC와 키 생성 정보를 보호하기 위하여 클라이언트 SACM과 서버 시스템의 PVF사이에 설정되어 사용되는 세션 키이고, 이 기본키와 발신자의 이름, 타임스템프(timestamp) 등의 키관련 정보를 통칭하여 기본키 패키지(basic key package)라 부른다.
- 대화 키는 단방향 해쉬함수를 통하여 기본키로 부터 파생된 세션키로서 이는 보안문맥상에서 서로 교환되는 데이타를 보호하는 데 그 목적이 있으며 데이타 기밀성 서비스용과 데이타 무결성 서비스용의 2개의 별도의 키로 이루어져 있다. 즉, 이 키들이 실제 서버스에 사용될 키가 되는 것이다. 또한, 이 대화키와 발신자의 이름, 타임스템프 등의 키관련 정보를 통칭하여 대화키 패키지(dialogue key package)라 부른다.

6. GSS-API

GSS-API는 네트워크 응용프로그래머들에게 정보보호 기술, 특히 인증 및 키 분배 프로토콜에 관한 기술을 구현함에 있어서 공통적으로 사용가능한 함수들을 정의하여, 정보보호 서비스를 제공시 이 인터페이스 함수만을 호출함으로써 개발자들에게 편리성을 제공하자는 데 가장 큰 목적을 두고, IETF(Internet Engineering Task Force)의 CAT(Common Authentication Technology) 활동그룹에서 1993년 9월에 RFC 1508로 제안한 인터페이스 방식이다.

SESAME는 GSS-API를 가능한한 그대로 준수하였으며, 부분적으로 일부 확장하였다. [표 3]은 GSS-API의 기본 호출함수(basic calls)의 구현상태를 정리한 것이고, [표 4]는 GSS-API 확장 호출함수(extension calls)의 구현상태를 정리한 것이다.

[표 3] GSS-API 기본 호출함수와 구현상태

GSS-API 호출함수	구현상태
GSS_Acqire_cred()	완전지원
GSS_Release_cred()	완전지원
GSS_Inquire_cred()	부분지원
GSS_Init_sec_context()	부분지원
GSS_Accept_sec_context()	완전지원
GSS_Delete_sec_context()	완전지원
GSS_Process_context_token()	완전지원
GSS_Context_time()	완전지원
GSS_Display_status()	완전지원
GSS_Indicate_mechs()	완전지원
GSS_Compare_name()	완전지원
GSS_Display_name()	완전지원
GSS_Import_name()	완전지원
GSS_Release_name()	완전지원
GSS_Release_buffer()	완전지원
GSS_Release_oid_set()	완전지원

[표 4] GSS-API 확장 호출함수와 구현상태

GSS-API 호출함수	구현상태
GSS_Get_mic()	부분지원
GSS_Verify_mic()	완전지원
GSS_Wrap()	부분지원
GSS_Unwrap()	완전지원
GSS_Modify_cred()	완전지원
GSS_Get_attributes()	완전지원
GSS_Compound_cred()	비기능적지원
GSS_Get_delegate_creds()	비기능적지원
GSS_Set_default_cred()	완전지원

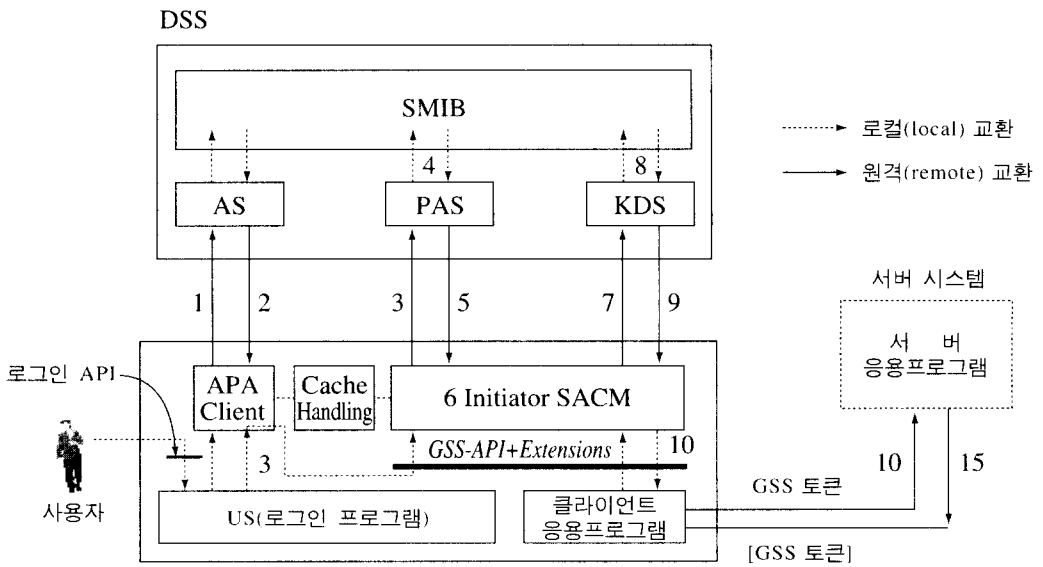
[표 4]의 GSS-API 확장 호출함수에서 위에서부터 4개의 호출함수는 응용프로그램에서 클라이언트/서버간의 통신구간을 보호하기 위한 것이고, 그 나머지는 정보보호 어트리뷰트와 대표 확장을 다루기 위한 것이다.

7. 인증 프로토콜

이절에서는 전형적인 사용자 세션에서 SESAME 시스템내의 컴포넌트들이 어떻게 상호작용하는지를 설명한다.

7.1 클라이언트의 인증절차

클라이언트 시스템의 프로토콜 교환에 내장된 SESAME 컴포넌트들을 나타낸 것이 [그림 6]이다.



[그림 6] 클라이언트 시스템과 SESAME 컴포넌트

클라이언트 시스템과 서버시스템간의 보안 문맥을 설정하기 위한 클라이언트 시스템에서의 내부적인 키 인증절차를 간략하게 설명하면 다음과 같다.

- 1단계
  US가 사용자의 신분을 확인하기 위하여 로그인-API를 통하여 APA클라이언트를 호출하여 AS에게 사용자이름과 인증관련 정보를 전달한다. 이때 사용자 패스워드와 특성 어트리뷰트는 캐쉬에 보관해 둔다.
- 2단계
  AS는 수신한 사용자 정보들을 통해 SMIB로부터 관련 정보들을 대조 확인후 PAS 티켓과 AS 인증정보들을 APA 클라이언트에게 다시 전달한다. 이때 APA 클라이언트에서는 수신한 PAS 티켓을 캐쉬에 보관해둔다.
- 3단계
  US가 사용자별로 정해진 직책(role name)과 사용자 특성 어트리뷰트들을 APA클라이언트를 통해서 SACM에게 전달하고, SACM은 GSS-API를 호출하여 사용자명, 특성 어트리뷰트, PAS 티켓을 PAS에게 전달한다.
- 4단계
  PAS는 수신한 PAS 티켓을 검증한 후 관련 어트리뷰트들을 SMIB로 부터 검색하여 가져온다.
- 5단계
  PAS가 PAC와 KDS 티켓을 발행하여 SACM에게 전달한다.
- 6 단계
  SACM은 수신한 PAC, KDS 티켓, 제어정보를 캐쉬에 저장한다.
- 7 단계
  SACM은 KDS에게 KDS 티켓과 응용 프로그램명을 전달한다.
- 8 단계
  KDS는 수신한 KDS 티켓을 검증한 후 SMIB로 부터 서버 시스템의 응용 프로그램의 PVF 비밀키를 가져온다.
- 9 단계
  KDS가 SACM에게 PVF 비밀키를 포함하고 있는 서비스 티켓을 송신한다.
- 10 단계
  캐쉬에 저장되어 있는 PAC, 서비스 티켓, 제어정보, 대화키쌍, 서버 시스템 주소등으로 구성된 GSS 토큰을 생성하여 서버 시스템에게 송신한다.

(11-14단계는 서버 시스템에서의 처리과정이다.)

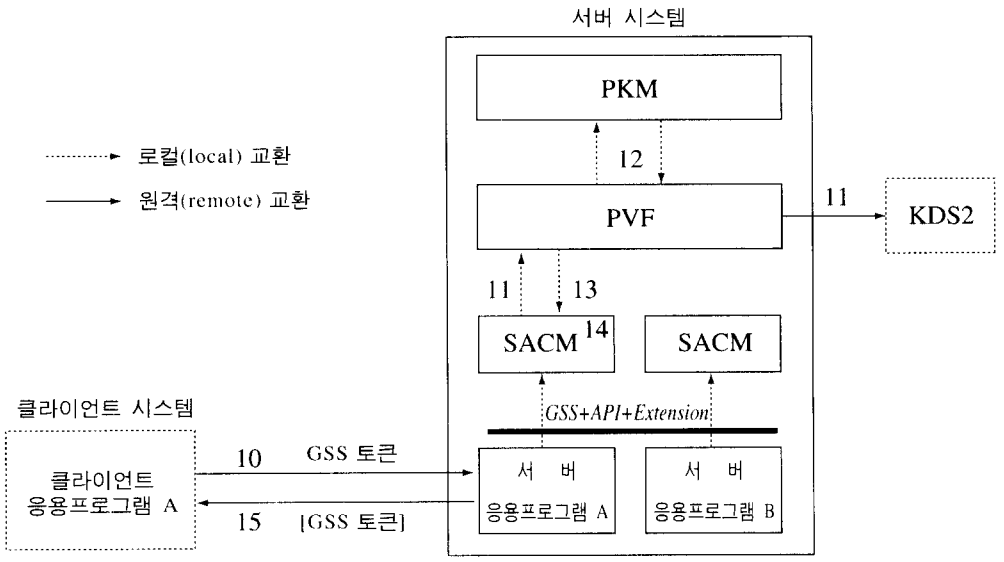
- 15단계
  선택적으로 상호 인증 메카니즘이 요구되었을 경우 서버 시스템에서 클라인언트 시스템에게 상호 인증하기 위한 GSS 토큰을 보내오는데, 이 토큰을 수신하였다면 이를 검증한다.

7.2 서버의 인증절차

서버 시스템에 포함되는 SESAME 컴포넌트들을 나타낸 것이 [그림 7]이다.

클라이언트 시스템과 서버시스템간의 보안 문맥을 설정하기 위한 서버 시스템에서의 내부적인 인증절차를 간략하게 설명하면 다음과 같다.

- 10 단계
  클라이언트 시스템으로부터 GSS 토큰을 수신한다.
- 11 단계
  수신한 GSS 토큰을 검증한 후, SACM을 통하여 PVF에게 PAC와 서비스 티켓을 전달한다.



[그림 7] 서버 시스템과 SESAME 컴포넌트

- 12 단계
  PVF는 PKM으로 부터 PAC를 서명한 PAS의 공개키와 PAS 보증서를 전달 받는다.
- 13단계
  PVF는 PKM으로부터 전달받은 PAS의 공개키를 가지고 PAC를 검증하고 그 결과를 SACM에게 전달한다.
- 14단계
  PVF로 부터 전달받은 PAC와 제어정보를 캐쉬에 저장한다.
- 15단계
  선택적으로 상호 인증 메카니즘이 요구되었을 경우 서버 시스템에서 클라이언트 시스템에게 상호 인증하기 위한 GSS 토큰을 생성하여 송신한다.

8. 키분배

이 절에서는 클라이언트 시스템과 서버 시스템의 PVF사이에 공유하는 기본키를 생성 관리하는 프로토콜과 클라이언트/서버 응용프로그램사이에 공유하는 대화키를 생성하는 방법들을 기술한다. 이 키분배 프로토콜에는 KDS를 사용하는 프로토콜과 KDS를 사용하지 않고 비대칭키 암호기법만을 이용한 프로토콜이 있다.

이 절에서 사용하는 프로토콜의 표기법은 다음과 같다.

Admin	관리 정보. (이것은 토큰마다 다를 수 있다.)
BKip	클라이언트 SACM과 서버 PVF간의 기본키
BKP	기본키 패키지
DESK	랜덤하게 발생되는 대칭키
DKCit	클라이언트 SACM과 서버 응용프로그램 사이의 기밀성 서비스용 대화키
DKIit	클라이언트 SACM과 서버 응용프로그램 사이의 무결성 서비스용 대화키
DKP	대화키 패키지

IS	클라이언트 SACM
IKPR	클라이언트의 개인키(private key)
KDS1	클라이언트의 KDS
KDS2	서버의 KDS
Kik	IS와 자신의 KDS사이의 기본키
KnPR	KDSn의 개인키
KnPU	KDSn의 공개키
Kpk	동일 도메인내의 PVF와 KDS간에 공유하고 있는 비밀키
PKPR	PVF의 개인키
PVF	PVF(PAC Validation Facility)
TA	서버 응용프로그램 이름
(Ticket)K	서비스 티켓
(DATA)K	K 키로 암호화된 데이타
[DATA]K	K 키로 해쉬된 데이타
[DATA]KnPR	KnPR 키로 서명된 데이타

8.1 KDS를 사용하는 프로토콜

이 프로토콜에서는 항상 다음과 같은 키들을 KDS가 사전에 저장 관리하고 있어야 한다.

- 서버의 PVF와 자신의 KDS사이에 공유하고 있는 대칭키 : Kpk
- 도메인간의 키 관리시에 사용될 KDS 자신의 개인키 : KnPR

여기서는 클라이언트 시스템의 KDS1로부터 IS로의 서비스 티켓 전달 단계부터 언급하기로 한다.

1. KDS1에서 IS에게 전달되는 다음의 메세지를 전달한다.

   [(BKip)Kik+BKP+Admin]Kik

   도메인내(Intra-domain)의 경우와 도메인간(Inter-domain)의 경우 모두 전달되는 메세지 구조는 항상 위와 같이 동일하지만, 한가지 다른점은 각각의 경우에 따라 BKP의 내용이 달라진다는 것이다. 각각의 경우의 BKP 구성은 다음과 같다.

   - 도메인내의 경우 : (Ticket)Kpk
   - 도메인간의 경우 : (Ticket)DESK+[(DESK)K2PU+Admin]K1PR

   여기서, Ticket에는 IS 식별자, BKip, TA가 포함되어 있다.

2. IS는 GSS 토큰을 생성한다. GSS 토큰의 구조는 다음과 같다.

   [PAC+(CV)Bkip+BKP+DKP+Admin]BKip

   여기서, DKP에는 DKIit, DKCit와 키 발급 관련 정보를 포함하고 있다.

3. 이제, IS는 TA의 PVF에게 다음의 메세지를 송신한다.

   BKP+GSS 토큰+Admin

4. TA의 PVF에서는 클라이언트와 서버간의 기본키를 추출하기 위해서는 GSS토큰내의 Ticket을 복호하여야 하는데, 같은 도메인내에서는 수신 서버의 PVF가 이미 알고 있는 비밀키(BKip)로 암호화하여 IS가 보내왔기 때문에 PVF에서 복호하여 BKP내의 BKip를 추출하는 것은 간단하다. 그래서, 여기서는 서로 다른 도메인간에 BKip를 공유하기 위한 절차만 설명하면 다음과 같다.

   4.1 PVF가 수신한 기본키를 직접 복호할 수 없으므로, 자신의 도메인내에 있는 KDS2와 공유하고 있는 비밀키로 다음

의 같이 수신한 BKP와 Admin을 암호화하여 KDS2에게 송신한다.

[BKP+Admin]Kpk

4.2 KDS2는 수신한 메세지를 복호하고, 그 다음 KDS1의 개인키인 K1PR로 서명된 BKP를 복호하면 (DESK)K2PU+Admin를 구할 수 있고, 그다음 DESK을 복호하여 IS가 생성한 Ticket를 구할 수 있게 된다. 그런후, 이 Ticket을 다시 PVF와 KDS2가 공유하고 있는 비밀키 Kpk로 암호화하여 PVF에게 전달하면, 이제 자기 자신의 도메인에서 생성한 Ticket처럼 사용할 수 있게 되고, 이를 통해 토큰내에 있는 IS가 생성한 기본키 BKip를 공유하게 되는 것이다.

8.2 KDS를 사용하는 않는 프로토콜

SESAME는 공개키 암호기법만을 사용하여 기본키를 설정하는 방법을 제공한다. 이러한 프로토콜에서는 더이상 KDS를 필요로 하지 않는다.

클라이언트 시스템이 서버 시스템과 기본키를 설정하고자 하는 시점부터 프로토콜 교환을 설명하면 다음과 같다.

1. 먼저, 클라이언트는 서버시스템이 개인키를 가지고 있는지를 확인하는데, 이를 위해서는 클라이언트의 SACM이 자신의 로컬에 있는 SMIB를 통해서 서버 응용프로그램의 디렉토리 보증서를 검색한다.
2. 클라이언트 SACM은 서버 PVF의 공개키를 가지고 기본키를 암호화한후, 이 기본키를 포함하는 토큰을 구성하고, 이 토큰을 다시 자신의 개인키로 서명하여 이를 자신의 디렉토리 보증서와 함께 서버 시스템에게 송신한다.
3. 토큰을 수신한 서버 시스템의 PVF는 토큰과 함께 수신한 클라이언트의 디렉토리 보증서내에 존재하는 공개키를 사용하여 이 토큰이 변경, 혹은 변조되지 않았는지를 검증한다.
4. 토큰의 검증을 통하여 기본키를 성공적으로 추출하였다면, 이를 통하여 대화키를 생성한다. 또한 상호 인증이 요구되었을 경우, 서버 SACM은 클라이언트 시스템에게 토큰을 생성하여 보낸다.

9. 결 론

지금까지 본 논문을 통해서 SESAME 프로젝트에서 Kerberos 인증 메카니즘을 어떻게 확장하였고, 키 분배를 위하여 공개키 암호기법을 어떻게 사용하였는지와 공개키 암호기법의 사용이 어떻게 시스템의 확장성(scalability)과 관리성(manageability) 측면에서 장점들을 제공하는지를 살펴보았다. 결과적으로 분산환경에서의 인증 및 키분배 시스템인 SESAME는 인증 서비스, 데이타 기밀성 서비스, 데이타 무결성 서비스 뿐만 아니라 접근제어 서비스도 제공한다.

SESAME의 또다른 장점은 시스템이 독점적으로 개발되지 않았다는 점이다. 대신에 ECMA의 표준 작업에 그 뿌리를 두고 있고, 어떤 특정 시스템 플랫폼이나 통신 프로토콜 장비에 의존하지 않았다. 개발 초기부터 여러 개발업체들의 다양한 환경에도 적합하도록 설계되었고 이러한 설계 방향들은 대규모 조직 혹은 여러 조직들로 이루어진 그룹에서 시스템 통합시에 더욱 그 효과가 나타날 것은 자명한 일이다. 앞으로, 정보보호 응용시스템의

설계시 반드시 SESAME 기술은 아니더라도, 특히 분산망 정보보호 기술로는 이러한 시스템 설계 방식을 취한 정보보호 시스템을 많이 볼 수 있을 것으로 전망된다.

참 고 문 헌

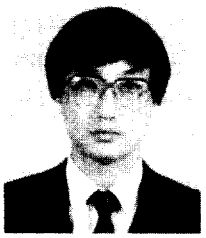
1. ECMA TR/46 " Security in Open Systems - A Security Framework", 1988.7.
2. Standard ECMA-219 "Authentication and Privilege Attribute Security Application with Related Key Distribution Functions" Parts1, 2, 3, 1994. 12.
3. "Generic Security Service API (GSS-API) Base" , X/OPEN Preliminary Specification, Document Number P308.
4. "Generic Security Service API (GSS-API) Security Attiribute and Delegation Extensions" , X/OPEN Snapshot, Document Number P307.
5. ISO/IEC DIS 10181(Parts 1 to 8), "Information Technology - Security Frameworks in Open Systems."
6. ISO/IEC 9594-8(X.509), "Information Processing Systems - Open Systems Interconnection - The Directory - Part 8: Authentication Framework."
7. S.B.Fairthorne, "Security Extension for DCE 1.1," OSF DCE SIG Request for Comments: 19.0, 1992. 12.
8. Rolf Oppliger, "Authentication Systems for Secure Networks," Artech House, 1996.
9. http://www.esat.kuleuven.ac.be/cosic/sesame.html

□ 著者紹介

이 정 현

1993년 숭실대학교 전자계산학과(공학사)
1995년 숭실대학교 대학원 전자계산학과(공학석사)
1995년 ~ 현재 한국전자통신연구원

※ 주관심분야 : 컴퓨터/네트워크 보안



윤 이 중

1988년 인하대학교 전산학과(공학사)
1990년 인하대학교 대학원 전산학과(공학석사)
1990년 ~ 현재 한국전자통신연구원

※ 주관심분야 : 컴퓨터/네트워크 보안, DBMS

이 대 기

1966년 한양대학교 전자공학과(공학사)
1987년 한양대학교 산업대학원 전자공학과(공학석사)
1980년 ~ 현재 한국전자통신연구원 책임기술원

김 대 호

1977년 한양대학교 전자공학과(공학사)
1984년 한양대학교 산업대학원 전자공학과(공학석사)
1993년 Visiting Scholar(University of Maryland at College Park Dept. of Computer Science)
1977년 ~ 현재 한국전자통신연구원 책임연구원

※ 주관심분야 : 전송분야, 통신 및 컴퓨터 보안