通信情報保護學會誌
第7卷 第2號, 1997. 6

개방형 분산 시스템 환경의 인증 메커니즘 분석

Analysis on Authentication Mechanism in Open Distributed Systems

강 창 구*, 최 용 락**

요 약

분산 시스템 환경은 다수의 컴퓨터가 네트워크에 연결되어 있고, 다양한 사용자가 공통적으로 접근할 수 있기 때문에 심각한 보안 문제를 안고 있다. 본 논문은 개방형 분산 시스템 환경에서 각종 위협에 대응할 수 있는 인증 메커니즘에 관하여 다룬다. 기본적인 위협 요소, 인증 요구조건 및 서비스를 검토하고 관련된 메커니즘들을 조사 분석함으로써 분산 시스템의 응용에서 암호학적 인증 메커니즘의 설계 기준을 마련할 수 있을 것이다. 본 논문에서는 Kerberos, X.509, SPX 및 SESAME 인증 메커니즘을 실체 인증의 개념적 관점에서 분석하였다.

1. 서론

분산 시스템은 다수의 호스트가 네트워크에 상호 연결되어 있고, 다양한 사용자가 공통적으로 이용하기 때문에 심각한 보안 문제를 갖고 있다. 분산 시스템에서 호스트들 사이에 분산된 여러가지 자원들은 서버에 의하여 제공되는 네트워크 서비스 형태로 공유된다. 어떤 자원을 액세스하고 싶은 개별적 프로세스는 원하는 서버에게 서비스를 요청할 수 있으며, 이때 상호간에 서로의 신분을 확인할 수 있는 인증 절차가 필요하다[1],[2].

분산 시스템에서 정보보호 위협은 시스템을 구성하고 있는 개별적인 호스트들의 손상과 메시지들을 통신하는데 관련된 손상으로 나눌 수 있다. 호스트를 위태롭게 하는 요소들은 프로세스 보호 모드와 같은 하드웨어 기법과 참조 모니터와 같은 소프트웨어 기법의 통합적인 방법을 이용하여 보호된다. 이것은 컴퓨터 시스템 보안의 주제로서 여기서는 다루지 않는다.

메시지 통신에 관련된 위협은 통신망을 통하여 전송되는 정보의 도청 및 메시지의 불법적인 수정, 삽입, 삭제와 과거 메시지의 재전송 공격등이 있을 수 있다. 도청은 소극적 공격 형태로써 시스템 동작에 영향을 주지않지만 시스템에 의하여 발견되지 않기 때문에 예방 수단을 이용하여야 한다. 불법적인 수정과 재전송은 적극적인 공격 형태로서 시스템 운

* 한국전자통신 책임연구원
** 대전대학교 컴퓨터공학과 교수

용 자체에 심각한 영향을 줄 수 있으며, 예방, 발견 및 회복 수단을 공통적으로 이용할 수 있다.

안전한 통신을 위하여 요구되는 기본적인 정보보호 서비스는 비밀유지와 무결성이다. 비밀유지는 메시지를 정당한 수신자만 읽을 수 있도록하고, 무결성은 모든 메시지가 그것이 전송되었던 것과 동일한 형태로 수신되도록 보장하는 서비스이다. 한편, 인증은 신분위장 및 재전송 위협으로부터 보호할 수 있으며, 다른 정보보호 서비스들과 다양한 형태로 연합하여 이용될 수 있으므로 분산 시스템 응용에서 중요한 서비스이다[3].

분산 시스템에서 통신 대상이 되는 사람, 장비의 일부, 컴퓨터 시스템에서 실행되고 있는 프로세스들을 통신 주체(principal)이라고 하며, 이들은 필요에 따라 인증되어야 하는 하나의 실체이다. 인증과정에서 본인의 신분을 나타내고, 본인임을 주장하는 실체를 신청자(claimant)라고 하며, 그 주장이 합법적인지 확인하려는 다른쪽 실체를 검증자(verifier)라고 한다[4].

분산 시스템에서 인증은 신청자와 검증자 사이에 교환되는 정보에 의하여 수행될 수 있으며, 인증 내용에 따라서 다음과 같은 세가지 형태로 구분할 수 있다.

(A1) 메시지 내용 인증 : 수신된 메시지가 그것이 전송되었을 때의 내용과 동일함을 증명.

(A2) 메시지 출처 인증 : 데이터 항목에 따라 표현된 신분의 확인으로서 수신된 메시지의 송신자가 메시지의 송신자 필드에 기록된 것과 동일함을 증명.

(A3) 실체 인증 : 통신 연결 또는 세션에 개입된 통신주체에 의하여 표현된 신분의 확인으로서 그 통신주체의 실제 신분이 주장하는 것과 동일함을 증명.

일반적으로 (A1)은 메시지를 전송하기 전에 메시지 인증 코드(MAC)를 부가하여 수행한다. 수신자가 MAC를 재계산하여 수신된 MAC와 비교함으로써 메시지 무결성이 확인될 수 있다. (A2)는 데이터가 변조될 수 없다는 적절한 보장을 할 필요가 있으며, 데이터가 변경될 수 없는 환경을 사용하거나, 수신한 데이터가 전송된 데이터의 디지털 서명과 일치함을 검사함으로써 확인될 수 있다. (A3)은 신청자와 검증자 사이의 통신 관계를 배경으로 통신 주체의 신분확인을 제공할 수 있으며, 이러한 확인은 일방향 인증 및 양방향의 상호 인증으로 수행될 수 있다[1],[3]. 여기서는 분산 시스템 환경의 프로세스, 사용자 및 개방 시스템등 개별 식별이 가능한 실체들 사이의 실체 인증에 관해서만 논한다.

본 연구에서는 분산 시스템 환경에서 통신에 개입된 주체들 사이의 신분 위장 및 재전송 공격의 위협 요소와 인증 서비스 방법들을 원리적 측면에서 조사 한다. 그리고, 실체 인증 메커니즘의 사례로서 Kerberos, X.509 인증 구조, SPX 및 SESAME를 실체 인증의 관점에서 비교 분석하고자 한다.

2. 인증 서비스의 요구조건

2.1 인증 위협 요소

인증의 목적은 통신 주체에 대한 신분 보장을 제공하는 것이며, 메커니즘은 이러한 인증 서비스를 위하여 신분위장 및 재전송 공격들의 위협으로 부터 보호해야 한다[1],[5].

신분위장이란 어떤 엔티티가 다른 엔티티인 것 처럼 가장하는 것이다. 이 위협요소는 합법적인 실체로부터 전달되는 자료에 불법적인 실체가 위장하여 개입하여 할 수 있는 모든 행위를 포함할 수 있다. 즉, 불법적인 실체는 전송 메시지에 합법적인 실체가 하는 것처럼

메시지의 생성, 삽입, 삭제 및 변조 등을 수행할 수 있다. 또한, 전송 되는 메시지에 대하여 불법적인 실체가 합법적인 실체로 위장하여 수신 또는 비수신의 응답을 할 수 있음을 포함한다. 따라서, 인증은 정보 보호 서비스의 가장 기본적이면서 폭 넓은 위협 요소에 대한 대응 수단이다.

통신 주체 A와 B가 각각 인증을 수행하는데 이용될 어떤 데이터베이스(DB)를 갖고 있고, A가 B에게 통신을 시작 하려고 한다고 가정하자. 이때, 공격자 C가 할 수 있는 신분 위장의 위협 형태는 다음과 같은 것들이 있을 수 있다[1],[6].

1) 도청 : 네트워크를 통하여 전달되는 A와 B사이의 메시지를 도청 할 수 있고, 계속되는 교환에서 A또는 B로 위장 개입할 수 있다.
2) A로 위장 : C가 A라고 주장하면서 통신을 개시하여 B에게 메시지를 보내고, B를 속일 수 있다.
3) B로 위장 : C가 B의 네트워크 주소에서 기다리다가 A로부터 연결이 되면 B라고 위장하여 A를 속일 수 있다.
4) A의 DB를 접근 : C가 A의 DB에 접근할 수 있다면, 자신이 A라고 주장하면서 B를 속일 수 있다.
5) B의 DB를 접근 : C가 B의 DB에 접근할 수 있다면 자신이 B라고 주장하면서 A를 속일 수 있다.
6) A와 B사이에 쌍방 위장 : A와 B사이에서 C가 A에게는 B로 위장하고, B에게는 A로 위장하여 메시지를 전달 및 관찰, 메시지를 수정 또는 메시지의 방향을 재정열/재전송/역방향 전송 할 수 있다.

재전송 공격은 전달되는 메시지를 불법적인 실체가 나중에 다시 재연하는 것으로서, 일반적으로 불법적인 수정등을 수행하여 재전송 될 수 있다. 이것은 분산된 통신 네트워크의 시간 동기화 어려움을 이용하여 다른 위협 요소들과 연합되어 수행 될 수 있다. 재전송 공격의 예는 다음과 같이 다양하게 나타날 수 있다[1].

- 단순 재전송 : 공격자가 메시지를 단순히 복사하고, 그것을 후에 재전송 하는것.
- 시간 범위 내에서 재전송 : 공격자는 유효한 시간 윈도우내에서 타임 스탬프가 있는 메시지를 재전송 할 수 있다.
- 발견되기 어려운 재전송 : 이런 상황은 원본 메시지의 사용을 정지시켜서 그것의 목적지에 도착하지 않고, 오직 재전송 메시지만 도착할 경우에 일어날 수 있다.
- 수정없이 역방향 재전송 : 이것은 메시지 송신자에 대한 재전송 반환이다. 이 공격은 관용 암호방식이 사용되고 송신자가 보낸 메시지들과 받은 메시지들 사이의 내용적 차이를 쉽게 알 수 없을 때 가능하다.

2.2 인증 서비스 방법

어떤 통신 주체가 자신이라고 주장하는 것을 증명하기 위해서 이용할 수 있는 원리는 "what you know"(proof by knowledge), "what you have"(proof by possession), "what you are"(proof by property)의 3가지 범주로 대분류 할 수 있다. 예를 들면, 실체 인증 서비스를 제공할 수 있는 방법으로서 다음과 같은 수단이 있을 수 있다[3],[6].

1) 패스워드 처럼 무엇인가 알고 있음을 증명.

2) 물리적인 키 또는 카드처럼 무엇인가 소유하고 있음을 증명.
3) 지문 처럼 어떤 변하지 않는 고유의 특성을 증명.
4) 이미 확인된 인증을 갖고 있는 신뢰된 제3자로 부터 확인.
5) 어떤 특정의 시간 또는 장소에 있음을 증명.

무엇인가 알고 있는 수단을 이용한 인증방법은 가장 일반적으로 사용되는 것이 패스워드이다. 시스템을 액세스할 때 사용자가 패스워드를 제시하면 인증 시스템이 보유하고 있는 패스워드 목록에 대응하는 값을 비교하여 사용자의 신분을 확인하는 기법이다. 그러나, 패스워드 목록이 노출될 위험이 있고, 추측이 가능하기 때문에 패스워드의 선택, 메모리 저장방식 및 관리등이 매우 신중하게 수행 되어야한다.

무엇인가 소유하고 있는 수단을 이용한 인증 방법은 자기 테이프를 붙인 카드 또는 IC 카드와 같은 물리적 토큰을 이용하는 것이다. 카드를 갖는 시스템을 액세스하면 인증 시스템이 그 카드의 인증 정보를 읽고, 자신이 보유하고 있는 정보와 비교하여 사용자 신분을 확인하는 기법이다. 이런 카드의 가장 큰 취약점은 쉽게 복제될 수 있는 것이며, 또한 무자격자가 소유하게 될 때 본래의 인증 목적을 달성하기 어려운 점이 있다. IC 카드는 자기 테이프 카드보다 쉽게 복제할 수 없는 장점이 있다.

어떤 변하지 않는 개인 고유의 특성을 이용한 인증 방법은 서명, 지문 및 목소리 등을 이용하는 것이다. 사용자가 서명을 하거나 특별한 단어를 발음하면 인증 시스템이 기록된 사용자의 프로필과 특성을 비교하여 신분을 확인하는 기법이다. 이것은 패스워드가 노출의 위험이 있고, 물리적 토큰 또한 도난의 위험이 있는것 보다 장점을 갖고 있지만, 반드시 실제의 사용자가 인증 시스템 위치에 있어야 하는 단점이 있다.

이와 같이 대부분의 방법들은 한가지로서 이용될 때 고유의 취약점을 갖고 있다. 예를 들면, 무엇인가 소유하고 있다는 것을 증명한다는 것은 그것의 소유자가 합법적인 실체인지 인증하는 것이 아니라, 소유된 객체 자신만을 인증하는 것이다. 이러한 약점은 몇 가지 다른 방법들을 연합하여 사용함으로써 극복될 수 있다. 예를 들면, 스마트 카드 소유에 의한 인증의 약점은 카드의 실제 사용자임을 증명하기 위하여 PIN(Personnel Identification Number)을 추가 사용함으로써 극복할 수 있다. 특히, 5번항은 취약하여 보통 신뢰될 수 있는 제3, 제4의 실체들과 연대 보증을 설정하여 이용될 수 있다.

한편, 인증 서비스는 다른 여러가지 정보 보호 서비스들과 상호 작용하여 제공될 수 있다[3].

(1) 액세스 제어

사용자는 액세스 제어 정책에 따라 대상 자원에 접근을 허용하게 될 액세스 제어 정보의 획득이 허용되기 전에 먼저 인증이 될 필요가 있을 것이다. 따라서, 인증 서비스는 액세스 제어 서비스에 의하여 사용 될 수 있도록 인증의 결과를 전달 할 수도 있다. 인증 정보의 취소는 기존 액세스의 취소를 의미 할 수도 있다.

(2) 데이터 무결성

인증의 연속성을 보장하고, 데이터 출처의 확증을 설정하기 위하여 데이터 무결성과 연관하여 인증이 사용될 수 있다. 또한, 어떤 인증 메커니즘은 무결성 서비스를 위하여 사용될 수 있는 키를 분배하는데 이용될 수 있다. 이러한 키가 함축된 데이터 안에 정의 될때,

전송되는 메시지로 부터 그 키를 유도해내기 위한 방법이 인증 교환 중에 명시되어야 한다. 또한, 이러한 키가 명확한 방법으로 정의 된다면, 추가적인 데이터가 인증 교환 중에 어느 방향에서든 전달 되어야 할 것이다.

(3) 데이터 기밀성 및 부인 봉쇄

어떤 인증 메커니즘은 기밀성 및 부인 봉쇄 서비스를 위하여 사용 될 수 있는 키를 함축적으로 또는 명시적으로 분배하기 위하여 이용될 수 있다. 무결성 서비스를 위한 키의 전달과 마찬가지로 인증 교환 중에 키를 유도해낼 수 있는 방법이 함께 정의 되어야 한다.

(4) 감사

감사를 위하여 사용될 수 있는 인증 관련 정보는 다음과 같은 것들이 있다.

- 확인된 신분과 같은 인증의 결과.
- 인증 정보의 취소에 관련된 정보.
- 인증의 연속성 보장에 관한 정보.
- 인증의 과정에 관련되어 있는 기타 정보.

3. 인증 메커니즘의 사례분석

3.1 KERBEROS

가. 개요

Kerberos는 MIT에서 Athena 프로젝트의 일환으로 개발된 인증 서비스이다. Kerberos가 해결하고자 했던 문제는 다음과 같다 : 워크스테이션에 있는 어떤 사용자가 네트워크를 통해 연결된 어떤 서버에 접속하려고 하는 개방된 분산 환경이 있다고 하자. 우리는 서버가 허가받은 사용자에게만 접속을 제한 할 수 있기를, 그리고 서비스에 대한 요구를 인증해 줄 수 있기를 바란다. 이러한 환경에서 워크스테이션은 네트워크 서비스에 대한 자신의 사용자를 올바르게 구별할 수 있다고 믿을 수 없다. 따라서, 서버에게 사용자를, 그리고 사용자에게 서버를 인증해 주는 서비스가 필요하다.

만약 여러 사용자에게 네트워크 연결이 되어 있지 않는 전용 개인 컴퓨터를 제공한다면, 각 사용자의 자원과 파일은 물리적으로 각 개인용 컴퓨터를 안전하게 함으로서 보호 될 수 있다. 그러나, 사용자들이 중앙 집중형 시분할 시스템을 제공받는다면, 시분할 운영체제는 반드시 보안을 제공해야 한다. 운영체제는 사용자의 확인에 기반을 둔 액세스 제어 정책을 강화할 수 있으며 사용자를 확인하기 위해 로그온 절차를 사용할 수 있다. 오늘날, 이 두 시나리오는 대표적인 것은 아니다. 좀더 일반적인 것은 사용자 전용 워크스테이션(클라이언트)과 분산 또는 중앙 집중형 서버로 구성된 분산 구조이다. 이 환경에서, 보안에 대한 세가지 접근법을 생각할 수 있다.

1) 사용자의 신원을 확인하기 위해 개별 클라이언트 워크스테이션에 의존한다. 그리고, 사용자 ID에 기반을 둔 보안 정책을 강화하는 각 서버에 의존한다.
2) 클라이언트 시스템은 서버에게 자신들을 인증하도록 요구한다. 그러나 클라이언트 사용자 확인에 대해서는 클라이언트 시스템을 신뢰한다.
3) 사용되는 각 서비스의 신원 증명을 사용자에게 요구하다. 또한, 서버가 이들의 신원을 클라이언트에게 증명하도록 요구한다.

소규모이면서 제한된 환경에서 즉, 모든 시스템이 단일 기관에 소유되어 운용되는 곳에서는 첫번째 또는 아마도 두번째 전략이면 만족 할 수도 있다. 그러나, 보다 개방된 환경에서 즉, 다른 기계에 대한 네트워크 접속이 지

원되는 곳에서는 사용자 정보와 서버에 저장된 자원을 보호하기 위해 세 번째 접근법이 필요하다. 이 세번째 접근법이 Kerberos에 의해 지원된다.

Kerberos는 분산된 클라이언트/서버 구조를 가정하고 인증 서비스를 제공하기 위해 하나 또는 그 이상의 Kerberos 서버를 사용한다. Kerberos는 공개키 암호방식을 전혀 사용하지 않고 관용 암호방식에 의존한다. 초기에 발표된 보고서에서는 다음과 같은 요구 사항을 기술하고 있다.

- 안전성(secure) : 네트워크 침입자는 정당한 사용자인 것처럼 가장하여도 필요한 정보를 결코 얻을 수 없어야 한다. 가능하다면 Kerberos는 잠재적인 침입자가 약한 고리를 발견 할 수 없을 만큼 강력해야 한다.
- 신뢰성(reliable) : 액세스 제어에 관하여 Kerberos에 의존하고 있는 모든 서비스에 대해 Kerberos 서비스 능력의 결함은 지원되는 서비스 능력의 결함을 의미한다. 그러므로 Kerberos는 신뢰성이 높아야 하고 한 시스템이 다른 시스템을 백업할 수 있는 분산된 서버 구조를 채택해야 한다.
- 투명성(transparent) : 이상적으로, 사용자는 접속하는 동안에 암호를 입력하는 것 이상으로 복잡한 인증 과정이 일어나고 있다는 것을 깨닫지 못해야 한다.
- 규모(scalable) : 시스템은 반드시 대규모의 클라이언트와 서버를 지원해야 한다. 다시 말하면, 이것은 모듈로 구성된 분산 구조이다.

나. 인증 절차 분석

보호장치가 없는 네트워크 환경에서는 모든 클라이언트가 어떤 서버에서든지 서비스를 받기 위해 접속 할 수 있다. 이런 경우 명백한 보안상의 위험은 정당한 사용자로 위장하는 것이다. 침입자가 다른 클라이언트인것 처럼 해서 서버에 대한 허가받지 않은 권한을 가질 수 있다. 이런 위험을 막기 위해서 서버는 반드시 서비스를 요청하는 클라이언트의 신원을 확인할 수 있어야 한다. 각 서버가 각각의 클라이언트/서버의 대화에서 이러한 임무를 수행하기 위한 요구를 받을 수 있으나 실제로 이것은 각 서버에게 과다한 부하를 갖게 한다.

Kerberos는 모든 사용자의 패스워드(암호)를 알고 이것을 중앙 집중식 데이터베이스에 저장하는 인증 서버(AS: Authentication Server)를 사용하여 각 서버의 과다한 부하를 해결하고 있다.

즉, 사용자는 워크스테이션에 로그인하고 서버에 접속하기 위한 요청을 한다고 가정할 때 Kerberos 버전 4의 인증 절차에서 사용된 각 메시지 교환 내용은 〈 표 1 〉과 같다.

AD_C는 전송되는 메시지를 가로채고 ID_C 이름을 도용하여 다른 워크스테이션에서 이용하는 위장을 방어한다. AD_C는 처음 티켓을 요구했던 당시의 사용자 주소이므로 동일한 워크스테이션에서만 가능하다.

패스워드를 이용한 인증 절차는 사용자 세션별로 단지 1번만 필요하며, (1), (2) 과정을 통하여 패스워드의 평문 전송 없이 AS로 부터 인증을 받을 수 있다. 따라서, 패스워드 전송에 대한 도청 및 가로채기 위협을 방어한다.

$Ticket_{tgs}$는 티켓-승인 티켓으로서 새로운 서비스를 액세스 할 때마다 C가 TGS에게 제공하여 AS와의 새로운 인증 절차 없이 이 티켓으로 자신을 인증 할 수 있다. 따라서, 각 서비스별로 패스워드를 이용한 인증 과정이 필요 없고 티켓을 재사용하여 새로운 서비스에 대한 요청이 가능하다. 이 티켓이 적에 의하여 획득되면 사용자가 로그 오프할 때까지 기다린 다음 TGS를 속이기 위하여 이 티켓을 재

〈 표 1 〉 Kerberos 버전 4의 인증절차

(a) 인증 서비스 교환 : 티켓-승인 티켓을 얻기 위해
(1) C → AS : $ID_C \| ID_{tgs} \| TS_1$
(2) AS → C : $E_{KC}[K_{C,tgs} \| ID_{tgs} \| TS_2 \| Lifetime_2 \| Ticket_{tgs}]$
$Ticket_{tgs} = E_{Ktgs}[K_{C,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifetime_2]$
(b) 티켓-승인 서비스교환 : 서비스-승인 티켓을 얻기 위해
(3) C → TGS : $ID_V \| Ticket_{tgs} \| Authenticatorc$
(4) TGS → C : $E_{KC,tgs}[K_{C,V} \| ID_V \| TS_4 \| Ticket_V]$
$Ticket_{tgs} = E_{Ktgs}[K_{C,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifetime_2]$
$Ticket_V = E_{KV}[K_{C,V} \| ID_C \| AD_C \| ID_V \| TS_4 \| Lifetime_4]$
$Authenticatorc = E_{KC,tgs}[ID_C \| AD_C \| TS_3]$
(c) 클라이언트/서버 인증 교환 : 서비스를 얻기 위해
(5) C → V : $ticket_V \| Authenticatorc$
(6) V → C : $E_{KC,V}[TS_5+1]$ (상호인증을 위하여)
$Ticket_V = E_{KV}[K_{C,V} \| ID_C \| AD_C \| ID_V \| TS_4 \| Lifetime_4]$
$Authenticatorc = E_{KC,V}[ID_C \| AD_C \| TS_5]$

- C 클라이언트
- AS 인증서버
- V 서버
- ID_C C에 있는 사용자 식별자
- ID_V 서버 V의 식별자
- AD_C C의 네트워크 주소
- K_V AS와 V에 의해 공유된 비밀 암호화 키
- TGS Ticket-Granting Server
- $Ticket_{tgs}$ Ticket-Granting Ticket
- K_c 사용자 패스워드(사용자의 클라이언트만 읽을 수 있는)
- $K_{C,tgs}$ C와 TGS의 세션키
- ID_{tgs} C와 TGS 서비스 사용을 요청하는 TGS 식별자
- E_{Kv} TGS와 서버만 알고 있는 비밀키로 암호화한다.
- K_{tgs} AS와 TGS에 공유된 비밀키
- $K_{C,V}$ C와 V 에 공유된 비밀키(TGS 발행)

사용 할 수 있기 때문에 타임 스탬프와 유효 기간을 티켓에 포함 시킨다. 티켓은 $Ticket_{tgs}$로 암호화 되어 있으므로 변조가 불가능하며, K_C로 재 암호화하여 C에게 전달하므로 인증 절차를 밟지 않은 제 3자가 티켓을 이용할 수 없다.

(2)의 메시지 내에 $K_{C,tgs}$를 포함시키고, 또한 티켓 내에도 포함시킴으로써 C 와 TGS에게 안전한 세션키를 전달 할 수 있다. 이 키를 이용하여 TGS가 인증자를 복호하고, 티켓의 소유자를 인증하는 것이 가능하다. TGS는 C로부터 수신된 티켓을 K_{tgs}로 복호하고 인증자는 $K_{C,tgs}$로 복호하여 $ID_C AD_C$ 를 비교한 다음 또한 들어오는 메시지의 네트워크 주소와 비교함으로써 티켓 소유자가 정당한 사용자임을 증명한다.

$Ticket_V$는 재사용 가능한 서비스-승인 티켓으로서 서비스가 필요할 때마다 C가 V에게 제공하며 TGS와의 새로운 인증 절차 필요 없이 이 티켓으로 자신을 인증 할 수 있다. C는 이 티켓을 V에게 제시하여 서버로부터 궁극적인 서비스를 제공 받을 수 있다.

(4)의 메시지 내에 $K_{C,V}$를 포함시키고, 또한 티켓 내에도 $K_{C,V}$를 포함시킴으로써 C와 V에게 안전한 세션 키를 전달 할 수 있다. 이 키를 이용하여 V가 인증자를 복호하고 C로부터 수신된 티켓을 K_V로 복호하여 티켓 소유자가 정당한 사용자임을 증명한다.

상호 인증이 요구된다면 V가 인증자로부터 추출된 타임스탬프+1을 반환하여 응답한다. 이 응답 메시지는 C가 $K_{C,V}$로 복호하여 1 증가된 타임스탬프를 확인함으로써 V가 정당한 서버이고 지나간 응답을 이용한 재전송 공격이 아님을 확인할 수 있다.

이제 C와 V가 비밀 키를 공유하고 있기 때문에 둘 사이의 향후 메시지 교환에 이 키를 이용할 수 있으며, 필요하다면 새로운 랜덤 세션 키 교환이 가능하다.

Kerberos 프로토콜에는 여러 가지 선택 사항들이 많이 있다. 예를 들면, 분리된 기관의 제어하에 있는 독립된 인증 서버들을 채택한 다중 영역(realms)의 사용을 포함하고 있다. 그러나, 여기서는 인증과 관련된 기본적인 원리만을 조사하였다.

Kerberos의 주요 장점은 비교적 비싸지 않은 기술을 사용하여 훌륭한 수준의 방어를 제공하는 것이다. 그러나, 다른 인증 구조와 비교하여 다음과 같은 단점들을 갖고 있다.

- 고도의 사용 가능성을 유지해야하며, 물리적으로 안전한 신뢰된 온-라인 서버를 필요로 한다.
- 재전송 공격 발견을 위하여 타임 스탬프에 의존적인데, 이것은 동기화 되고 안전한 클럭이 요구됨을 의미한다.
- 인증 과정에서 사용된 키가 손상을 입는다면 인증을 위하여 그 키를 사용했던 어떤 세션에서 이동되는 보호된 모든 데이터들을 위태롭게 할 것이다.

3.2 X.509

가. 개요

OSI디렉토리 표준이 1988년에 처음 발행되었는데, 그 중 X.509는 인증 구조를 다루고 있다. 디렉토리는 실제로 서버나 사용자에 관한 데이터베이스를 관리하는 분산된 서버의 집합으로써 공개키 인증서 형식의 저장소 역할을 할 수도 있다. 각 인증서에는 공개키의 사용자가 포함되어 있고, 인증기관의 개인키로 서명되어 있다.

X.509는 X.500디렉토리 서비스가 광범위하게 사용될 전망이기 때문에 중요한 표준으로 평가되고 있다. 또한, X.509에 정의되어 있는 인증 구조와 프로토콜은 PEM(Privacy

Enhanced Mail)과 같은 다른 상황에서도 이용될 수 있다. X.509는 공개키 암호화 기법과 디지털 서명에 근거를 두고 있다. 이 표준에서 특정한 알고리즘은 묘사하지는 않지만 RSA를 권고하고 있다.

X.509구조의 핵심은 각 사용자와 관계된 공개키 인증서이다. 이 사용자 인증서는 어떤 신뢰할 만한 인증 기관(CA: Certification Authority)에 의해 발행되는 것으로 가정되고 CA나 사용자에 의해 디렉토리에 위치하게 된다. 디렉토리 서버 그 자체는 공개키의 제작이나 인증 기능에 대한 책임이 없다. 이것은 단순히 사용자가 인증서를 얻을 수 있는 손쉬운 접속 장소를 제공할 뿐이다.

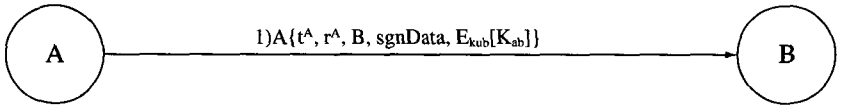
표준은 인증서를 정의하기 위해 다음의 표기법을 사용한다.

$$CA\langle\langle A\rangle\rangle = CA\ \{\ V, SN, AI, CA, T, A, A_p\ \}$$

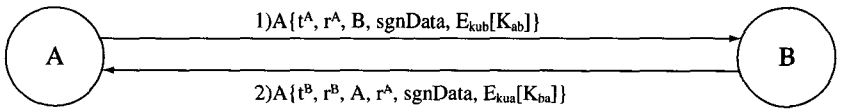
여기서, $CA\langle\langle A\rangle\rangle$ 는 인증기관 CA에 의해 발행된 사용자 A의 인증서이며, CA{I}는 CA에 의한 I의 서명으로써 이것은 암호화된 해쉬 코드가 추가된 I로 구성되어 있다.

CA는 자신의 개인키로 인증서에 서명을 한다. 대응하는 공캐키가 사용자에게 알려져 있으면 그 사용자는 CA에 의해 서명된 그 인증서가 유효한지 확인할 수 있다. 그리고, 사용자 B가 사용자 A의 인증서를 갖고 있으면, A의 공개키를 갖고 메시지를 암호화하여 기밀성을 제공하거나, 또는 A 자신의 비밀키로 암호화하여 인증 서비스를 제공할 수 있다.

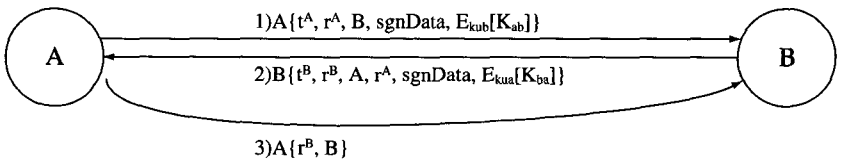
나. 인증 절차 분석



(1) 일방향 인증



(2) 양방향 인증



(3) 세방향 인증

〈 그림 1 〉 X.509 인증 절차

비밀키의 소유를 증명함으로서 신분을 확인할 수 있는 다양한 인증 절차들이 가능하다. X.509에서는 〈그림 1〉과 같이 여러 범주의 응용에서 유용하게 사용될 수 있는 3가지의 특정 인증 절차를 제시하고 있다. 사용된 심볼들은 다음과 같다.

- r^A : A에서 다시 반복되지 않는 임의의 수
- r^B : B 에서 다시 반복되지 않는 임의의 수
- t^A : 사용자 A에서 생성된 토큰 시작과 만기 날짜
- t^B : 사용자 B에서 생성된 토큰 시작과 만기 날짜
- sgnData : 데이터 출처인증을 위한 디지탈 서명
- K_{ab} : A와 B 사이의 비밀키
- K_{ba} : B와 A 사이의 비밀키
- KUb : B의 공개키
- KUa : A의 공개키
- E_{KUb} [I] : KUb을 사용하여 I를 암호화
- E_{KUa} [I] : KUa을 사용하여 I를 암호화
- A{I} : I에 대한 사용자 A의 서명
- B{I} : I에 대한 사용자 B의 서명

(1) 일방향 인증

(1) A : r^A 생성

(2) A B : A{ t^A, r^A, B, sgnData, $E_{KUb}[K_{ab}]$ }

(3) B : (a) A_P를 구하고 A의 인증서 유효기간 확인
(b) 서명을 확인하고, 서명된 정보의 무결성 검사
(c) B 자신이 정당한 수신자인지 검사
(d) t^A가 현재 시간인지 검사
(e) r^A가 재전송 되지 않았는지 검사

일방향 인증은 발신측 개인키로 서명된 인증 토큰을 추가하여 송신 함으로써 성취되며 다음과 같은 서비스를 제공한다.

- 실제로 메시지가 A에 의해서 생성되었다는 A의 신분 확인
- 메시지가 B측으로 보내질 의도라는 것
- 메시지의 무결성 및 다수로 송신되지 않았다는 고유성

이 경우는 단지 발신처의 신분만 확인될뿐이며 응답측은 아니다. 포함되는 메시지는 타임 스탬프 t^A, 고유번호 r^A, B의 신분 및 A의 공개키로 서명된 것을 포함한다. t^A는 토큰의 생성 및 만료 날짜로 구성되며 이것은 메시지의 지연 배달을 방지한다. 고유번호 r^A는 재전송 공격을 방지하기 위해 사용되며 이 번호는 메시지가 만료되기까지의 범위내에서 유일해야 한다. 따라서, B는 기간이 만료될때까지 그 번호를 저장하고 동일한 번호를 갖는 새로운 메시지는 거부할 수 있다. 순수한 의미의 인증은 A가 B로 보내는 신뢰성 확인을 위하여 단순히 사용되는 것이지만 메시지에는 전송될 정보들을 포함할 수도 있다. 이러한 정보 sgnData가 그것의 인증성과 무결성을 보장하면서 서명의 범주내에 포함된다. 이 메시지는 또한 B의 공개키로 암호화된 세션키를 B로 전송하기 위해서 사용될 수도 있다.

(2) 양방향 인증

(1), (2), (3)은 일방향 인증과 동일

(4) B : r^B를 생성

(5) BA : B{t^B, r^B, A, r^A, sgnData, $E_{KUa}[K_{ba}]$ }

(6) A : (a) 서명을 확인하고, 서명된 정보의 무결성 검사
(b) A가 정당한 수신자인지 검사
(c) t^A가 현재 시간인지 검사
(d) r^A가 재전송 되지 않았는지 검사

일방향 인증에서 제공되는 3가지 요소외에 다음의 사항들이 추가된다.

- 응답으로 생성된 메시지가 실제로 B에 의하여 생성되었다는 B에 대한 신분 확인
- 메시지가 A측으로 보내질 의도라는 것
- 응답의 무결성 및 고유성

양방향 인증은 통신상의 양측이 서로 상대방의 신분 확인을 허용한다. 응답 메시지는 응답을 확인하기 위하여 A로 부터의 고유번호를 포함하며, 또한 B에 의하여 생성된 타임 스탬프와 고유 번호, 그리고 서명된 추가정보 및 A의 공개키로 암호화된 세션키를 포함할 수 있다.

(3) 세방향 인증

(1), (2), (3), (4), (5), (6)은 타임 스탬프가 검사될 필요가 없다는 것을 제외하고 양방향 인증과 동일

(7) A : 수신한 r^A가 전송한 r^A와 동일한지 검사

(8) AB : $A\{r^B, B\}$

(9) B : (a) 서명을 확인하고, 서명된 정보의 무결성 검사
(b) 수신한 r^B가 전송한 r^B와 동일한지 검사

세방향 인증에서는 양방향 인증과 동일한 성질을 가지고 있지만 고유번호 r^B의 서명된 복제를 갖고 있는 A로부터 B로의 마지막 메시지가 포함된다. 이러한 설계의 목적은 타임스탬프가 검사될 필요가 없다는 것이다. 즉, 각 고유번호가 양측에 반향되기 때문에 재전송 공격을 검출하기 위하여 반향되는 고유번호를 검사할 수 있다.

3가지 종류의 각 절차는 인증정보의 교환되는 수효가 다른데, 결과적으로 사용자간의 다른 인증 유형을 제공한다. Kerberos와 비교하여 X.509는 인증서가 인증기관의 서명을 갖고 있기 때문에 물리적으로 안전한 온라인 서버를 요구하지 않는다는 장점이 있다. 공개키 인증서는 분산 디렉토리 서비스를 이용하여 오프라인으로 분배될 수 있다. 양방향 X.509와 Kerberos가 타임 스탬프에 의존적인 제한점을 세방향 X.509 인증 교환에서는 극복하고 있다. 그러나, 공개키 암호 방식을 이용한 대칭-키 분배 방식 때문에 X.509교환은 인증에 사용된 키가 손상을 입을때 그 인증키가 사용 되었던 어느 세션에서 전달되는 보호된 모든 데이터를 위태롭게 한다는 문제를 갖고 있다.

3.3 SPX

가. 개요

SPX는 분산 시스템 보안 구조(DSSA : Distributed System Security Architecture)의 일부분으로써 DEC(Digital Equipment Corporation)에서 인증과 기타 다른 보안 설비들을 포함하여 개발되었다[11],[15]. 인증은 이름과 통신 주체의 장기적인 개인키를 포함하고 있는 신임장, 공개키에 통신 주체를 연결하는 인증서, 그리고 인증 토큰들을 포함한다. 공개키 인증서는 오프-라인 인증 기관에 의하여 발행되고, 인증서 분배 기관(CDC : Certificate Distribution Centers)에 특정 관리자에 의하여 배치된다. 장래의 버전에서 CDC의 기능은 naming 서비스에 의하여 대체될 수 있다. 신임 초기화는 통신 주체의 패스워드를 포함한다. CDC는 인증 중에 온-라인이다. 단지 장기적인 비밀키의 암호화와 패스워드의 일방향 해쉬가 통신 주체에 공개될 뿐이다. SPX는 관용 및 공개키 암호화와 디지털 서명을 사용한다. 현재는 CDC 데이터 베이스로부터 인증서의 삭제에 의해서만 취소가 가능하다. SPX는 또한 인증서의 위임 기능을 제공한다.

나. 인증 절차 분석

〈그림 2〉는 SPX의 인증 절차를 요약하여 나타내고 있다.

〈그림 2〉 SPX 인증 절차

(1) $A \rightarrow CDC : ID_B$

(2) $CDC \rightarrow A : \boldsymbol{E}KR_{CA,A}[ID_B \| KU_B \| L1]$

(3) $A \rightarrow B : ID_A \| \boldsymbol{E}KR_A[KU_{DA} \| L2] \| \boldsymbol{E}KU_B[K_{A,B}] \| KR_{DA}[\boldsymbol{E}KU_B[K_{A,B}]] \| T \| \boldsymbol{E}K_{A,B}[T \| AD_A]$

(4) $B \rightarrow CDC : ID_A$

(5) $CDC \rightarrow B : \boldsymbol{E}KR_{CA,B}[ID_A \| KU_A \| L3]$

(6) $B \rightarrow A : T \| \boldsymbol{E}K_{A,B}[T \| AD_B]$

ID_A, ID_B	: 사용자 A, B의 식별 스트링
KU_{DA}	: 사용자 A의 위임 공개키
KR_{DA}	: 사용자 A의 위임 개인키
KU_A, KR_A	: 사용자 A의 공개키 및 개인키
$K_{A,B}$	: A와 B사이에 공유된 세션키
L1, L2, L3	: 토큰의 수명
$KU_{CA,A}$, $KR_{CA,A}$	: A에 의해 신뢰되는 CA의 공개키, 개인키
$EKR_A[M]$	: A의 개인키로 M을 서명
$EK_{A,B}[M]$	: 키 $K_{A,B}$로 M을 암호화
T	: 타임 스탬프
AD_A, AD_B	: A, B의 워크스테이션 주소

프로토콜의 첫번째 메시지에서 신청자 A는 검증자 B의 공개키를 CDC에게 요청한다. (2)에서 CDC는 A가 신뢰하는 인증 기관의 공개키를 사용하여 A 자신이 검증할 수 있는 검증자의 공개키에 관련된 인증서로써 응답한다. 이제, (3)에서 신청자는 위임 및 세션키를 위한 공개키 짝을 생성한다. 신청자는 수신자에게 위임 공개키를 그것의 유효시간과 함께 자신의 비밀키로 서명하고 서명된 복사물에 검증자의 공개키 하에서 암호화된 세션키(그것의 비밀 위임키 사용)와 인증자를 보낸다. 인증자는 세션키하에서 암호화된 타임 스탬프 및 기타 정보를 포함한다. (4)에서 검증자는 CDC에게 (5)에서 제공되는 신청자의 공개키 인증서를 요청한다. 이 인증서를 갖고 B는 신청자의 신분을 검증한다. B는 세션키 $K_{A,B}$를 KR_B를 사용해서 복호한 다음, 인증자를 복호하기 위하여 사용한다. 다음으로, 그것의 로컬 클럭을 갖고 시간을 검사하고, KU_A로 KU_{DA}를 검색하며, 암호화된 세션키의 서명을 검사한다.

상호 인증을 위하여 검증자 B는 식별 타임 스탬프를 갖고 (3)에서 사용된 형태의 인증자와 함께 (6)에서 응답한다. A는 인증자를 복호하기 위하여 세션키를 사용하고 자신의 복사물에 대하여 T를 검증한다. 만일 결과가 일치한다면 B는 A에 의하여 인증된 것이다. 이 인증 프로토콜에서 적용된 각 단계별 특징을 요약하면 다음과 같다.

- 통신 주체는 개인키의 지식을 증명함으로써 인증하며,상호 인증을 지원한다.
- 공유된 세션키가 설정된다. 이 세션키들은 시작하는 통신 주체에 의하여 생성된다.
- SPX는 인증중에 디지털 서명과 암호를 위하여 공개키 암호를 사용한다.
- SPX는 동기화를 위해서 타임 스탬프를 사용하므로 동기화와 보호된 클럭이 요구된다.
- 인증기관 인증서를 정확히 발행하도록 신뢰되어야 한다.
- CDC는 두 통신주체 사이의 인증중에 온-라인이어야 한다.
- 권한의 취소는 CDC에서 무효 인증서의 목록을 유지하거나, CDC로 부터 인증서를 삭제함으로써 지원된다.
- 인증기관의 축차적 구조가 구현될 수 있다.

3.4 SESAME

가. 개요

ECMA(European Computer Manufacture Association)는 정보 및 통신 시스템 분야의 표준화 작업을 위하여 1961년에 창설 되었다. SESAME(a Secure European System for Applications in a Multivendor Environment)는 CEC(Commission of the European Communities)의 지원에 의하여 ECMA에서 시작된 연구개발 프로젝트이다[16],[17].

SESAME는 본질적으로 2단계 프로젝트로써 구성되어 있다. 첫 단계에서는 보안 서비스의 구조적 아이디어와 원리들이 타당성이 있고 실제적인지 보이기 위한 시범적 구현 개발 프로젝트였다. 이것은 1991년에 끝났고, 결과 프로토타입 구현이 일반적으로 SESAME V1 으로써 언급된다.

두번째 단계에서는 상용 보안 제품의 구성에서 사용할 수 있도록 보안요소들을 개발하는 것이었다. 이 단계는 SESAME V2로 불리는 중간버전의 개발로써 SESAME 사용자 그룹 및 다른 CEC 지원 프로젝트의 멤버들에게 제한된 베타 테스팅을 위하여 1994년 7월에 발표되었다.

이러한 두 단계의 끝에 SESAME V3로 구체화 되었으며, CEC 지원 다른 프로젝트와 마찬가지로 1994년에 면밀한 검사 및 감사를 받았고 1995년 인준을 받았다. ECMA 보안 구조의 검증을 위하여 SESAME V1은 프로필 ECMA 인증 서비스를 포함해야 한다. 그러나, SESAME 2단계에서는 기존에 설치된 기반으로부터 쉬운 이전을 위하여 Kerberos V5의 사용 결정이 있었다. 따라서, SESAME V2는 권한의 안전한 전송과 공개키 암호의 사용에 의한 영역간 키 분배를 위하여 Kerberos V5에서 제공되는 기능들을 효과적으로 이용 하고 확장 보완 하였다.

SESAME 소스 코드는 다음 기관들 협력으로 개발 되었다.

- Bull SA
- International Computers Ltd(ICL)
- Siemens Nixdorf Informationssysteme(SNI) AG
- Software and System Engineering(SSE) Ltd.

SESAME 문서는 협력 개발자 Bull, ICL, SNI 및 SSE의 판권에 따라 보호된다. 그러나, 넓은 관심을 갖을 것으로 생각되는 선택적 부

분은 공개 된다. 특히, SESAME의 기능 명세와 관련된 문서는 SESAME 프로젝트 관리 위원회에 의해 분류 되고, 필요에 따라서 또는 관련된 표준화 활동을 위해 발표 된다.

SESAME 프로젝트는 ISO/IEC 보안 구조에 소개된 용어를 사용하였다. 특히, 시스템에 등록되고 인증할 수 있는 시스템 엔티티 또는 사용자를 언급하기 위하여 통신 실체(principal)를 사용한다.

- 접근을 요청하는 능동적 역할의 통신 실체를 개시자(initiator)라고 부른다.
- 접근되는 수동적 역할의 통신 실체를 타겟(target)이라고 한다.

서비스는 수많은 별도의 서버들로써 구현될 수 있는 기능적 결합체이다. 클라이언트/서버 모델에서 개시자로써 동작하는 클라이언트 응용 요소는 타겟으로써 동작하는 서버 응용 요소와 통신한다. SESAME는 인증, 접근제어, 데이터 기밀성 및 데이터 무결성 서비스를 제공한다.

나. 인증 절차 분석

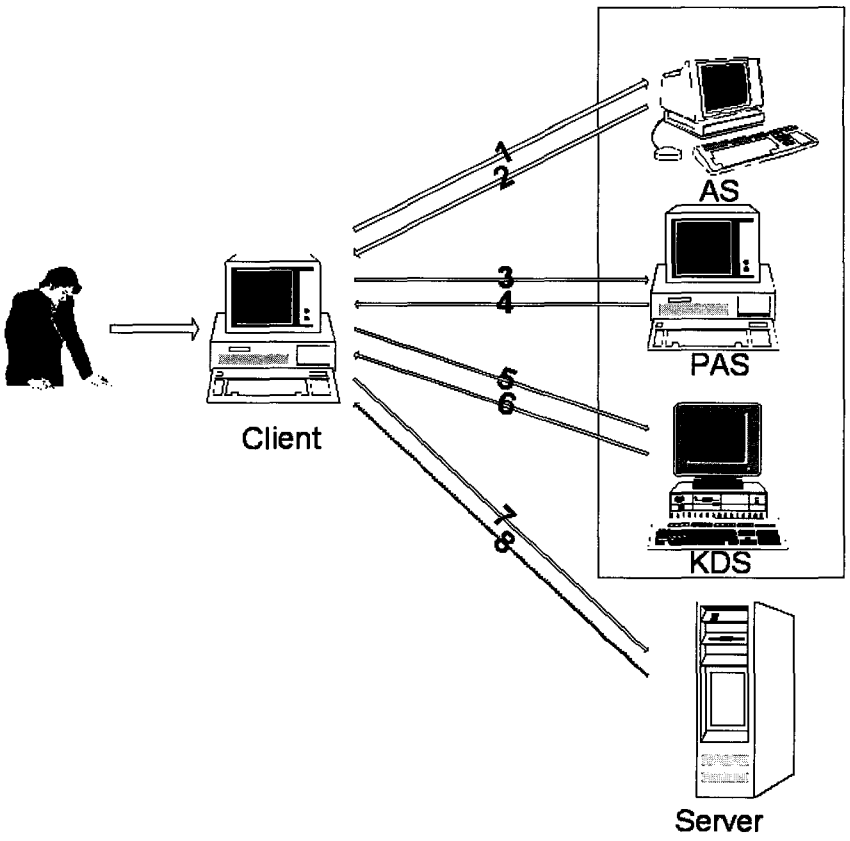
기본적으로 Kerberos 모델에서 클라이언트는 AS로 부터 TGT를 요청하고, TGS로 부터 티켓 요청을 위하여 이 TGT를 사용할 수 있다. SESAME 모델에서 동일한 방법이 권한 부여와 접근제어를 위하여 사용 되었다. 만일 클라이언트가 서비스 사용을 원한다면 AS에 의해서 인증 되어야 하고, PAS(Privilege Attribute Server)에 의하여 권한 속성이 인증 되어야 한다. SESAME는 권한 속성의 인증된 집합을 언급하기 위하여 PAC(Privilege Attribute Certificate)를 사용한다.

PAC는 사용자의 특권과 대응하는 제어 정보들로 구성된다. 사용자의 특권은 사용자의 신분, 직무, 조직의 그룹, 보안 등급과 같은 데이터이며, 반면에 제어 정보는 PAC가 언제, 어디에 사용 될 수 있으며, 권한 위임이 될 수 있는지 여부를 나타낸다. PAC는 개념상으로 ISO/IEC 에서 명시된 ACC(Access Control Certificate)와 유사하다. SESAME 모델에서 PAS는 인증의 표현을 위하여 PAC를 생성하는데, 이것은 대응하는 PAS의 개인키로 디지털 서명 되어 있다.

단순화된 SESAME 모델은 〈그림 3〉과 같다. 클라이언트와 보안 및 응용 서버들 사이에 교환되는 메시지는 그림에 나타난 번호와 같다. 이 번호들은 〈표 2〉에 요약된 SESAME 프로토콜의 각 단계와 대응된다.

〈표 2〉 SESAME 프로토콜의 메시지

단 계	메 시 지
(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



〈그림 3〉 단순화된 SESAME 모델 개관

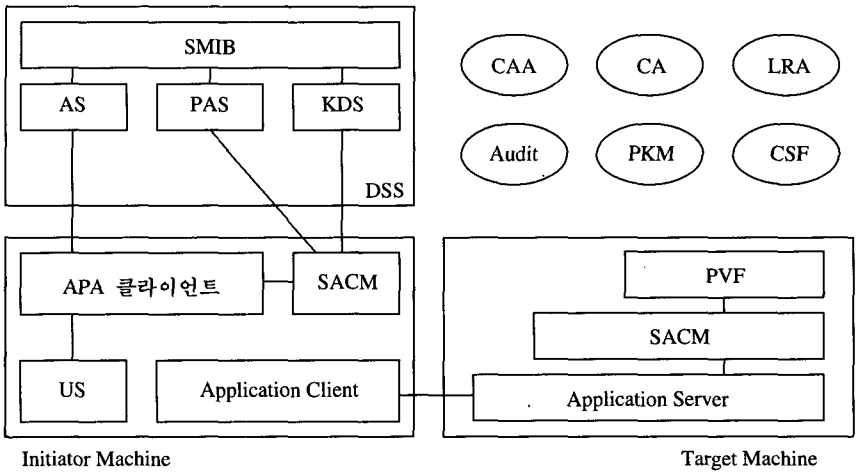
SESAME V3의 구조는 실제로 〈그림 3〉에서 보는 것 보다 훨씬 복잡하다. 네트워크의 응용 관점에서 보면 SESAME V3의 구조는 〈그림 4〉와 같다.

왼쪽의 개시자 기계에서 실행하는 응용 서버가 오른쪽 타겟 기계에서 실행하는 응용 서버에 인증을 원한다고 가정하자. 개시자와 타겟 기계는 동일한 SESAME 보안 영역, 또는 다른 영역에 등록 될 수 있다. SESAME 모델에 개입된 다양한 구조적 요소들 중 개시자의 측면에 설치되어야 하는 요소들은 다음과 같다.

- US는 시스템에 사용자 인터페이스를 제공한다. SESAME V3에서는 단지 최소한의 명령어가 제공된다.
- APA 클라이언트는 초기 인증과 디폴트 PAC 획득에 대한 상세한 서술로부터 숨기기 위해 US에 의하여 사용되는 구조적 요소이다.
- 일반적으로, SACM은 하나의 특별한 응용에 대한 클라이언트 및 서버 사이의 보안 연결을 설정하고 유지하는 책임이 있다.

또한, SACM 요소는 타겟측에 설치되어야 하며, 이 요소는 타겟 SACM으로써 언급된다. 타겟 SACM은 PVF에 의하여 지원된다. PAF는 PAC의 유효성을 검사하고, 개시자와 안전한 대화를 위해 필요한 대화 키를 반환 한다. 개시자

또는 타겟쪽에 설치된 이러한 요소들 외에 수 많은 다른 장소들에서 요구되는 3가지 범용 요소들이 있다.



APA : Authentication and Privilege Attribute
SACM : Secure Association Context Manager
PVF : PAC Validation Facility
US : User Sponser

〈그림 4〉 네트워크 구조

- PKM(Public Key Management)은 공개키 암호 시스템의 키 생성과 취소와 같은 사용을 위해서 명령어 관리적 툴의 집합과 마찬가지로 여러가지 인증 및 키 관리 기능들에 대한 접근을 제공하는 연결 라이브러리를 구성 한다.
- CSF(Cryptographic Support Facility)는 SESAME 요소 또는 SESAME 지원 응용에서 사용되는 암호학적 알고리즘을 구현한다. SESAME V3에서 현재 구현되고 사용된 알고리즘들은 DES-CBC, RSA, MD5 및 DES-MD5이다. 그러나, CSF는 지역의 법률 또는 미래의 보다 나은 알고리즘 출현에 따라서 특정 알고리즘이 대체되고 키 크기가 조정될 수 있도록 설계 되었다.
- Audit는 시스템에서 발생하는 보안사건 관련 조치로써 나중의 분석을 위한 기록에 가치가 있다. SESAME V3에서 감사 설비는 자신의 사용자 식별하에서 실행하고 대응 파일에 감사 정보를 저장하는 감사 데몬에 기록용 파이프를 통한 감사 사건들을 기록한다. 따라서 감사 파일은 응용 프로세스에 의한 수정으로 부터 보호된다.

한편, SESAME는 기본 키(basic key)와 대화키(dialogue key) 2-단계의 키 구조를 사용한다. 기본키는 개시자 SACM과 대응된 PAC및 키 설정 정보를 보호하기 위한 타겟 SACM의 PVF 사이에 사용 된다. 대화키는 일방향 해쉬 함수를 사용한 기본 키로부터 유도된다. 대화키는

보안 연결에서 교환되고 있는 데이터를 보호하는 것이다.

사용자가 클라이언트 워크스테이션에 로그인 할때, US에게 이름, 패스워드 및 요청 직무를 제공해야 한다. US는 이 정보를 APA 클라이언트에게 전송하고 SESAME 프로토콜 단계(1)에서 APA 클라이언트는 TGT 요청을 위해 KRB-AS-REQ 메시지를 AS에게 보낸다. KRB-AS-REQ 메시지의 형식은 본질적으로 Kerberos 프로토콜과 동일하다. AS는 PAS 티켓과 대응 기본키를 생성하고, 단계 (2)에서 APA 클라이언트에 KRB_AS_REP 메시지를 반환 한다.

Kerberos AS는 PAS 티켓과 대응 기본키의 획득 결과를 교환 하고, 개시자 쪽의 SACM에게 맡긴다. 지금부터 클라이언트와 DSS 사이의 모든 상호 작용은 개시자 SACM에 의하여 수행된다.

단계 (3)에서 개시자 SACM은 KRB-PAS-REQ 메시지를 PAS에게 보낸다. 메시지는 PAS 티켓과 요구되는 사용자 직무를 포함한다. 요구된 사용자 직무를 고려하여 PAS는 PAC을 생성하고, 자신의 개인키로 디지털 서명하며, 부가적으로 KDS 티켓을 생성한다.

단계 (4)에서 PAS는 KRB_PAS_REP 메시지를 개시자 SACM에게 반환하는데, 이 메시지는 PAC과 KDS 티켓을 포함하고 있다. 또한, PAC을 위임 할 수 있다면 메시지는 필요한 제어값(CV)들을 포함할 수 있다. KRB_PAS_REP 메시지는 개시자 SACM이 PAS와 공유하고 있는 기본키로 암호화 된다.

개시자 SACM은 지금까지 받은 PAS, KDS 티켓, CV 및 기본키 모든 정보를 저장한다. 이제 사용자 프로그램은 대응하는 API 호출을 통하여 PAC에 있는 속성들을 접근할 수 있고, 사용자는 현재 작업중인 특권이 무엇인지 알 수 있다. 이러한 초기화 단계들이 성공적으로 수행되면 개시자 쪽의 응용 클라이언트는 대응하는 GSS-API 호출을 통하여 SACM 요소에 요청할 수 있다. 만일 어떤 이유 때문에 SACM에 의하여 저장된 PAC이 유용하지 않다면, SACM은 PAS로 부터 새로운 PAC를 요청 한다.

클라이언트가 타겟쪽에 있는 특정 응용 서버를 사용하고 싶다면 단계 (5)에서 KRB_TGS_REQ 메시지를 KDS에게 보냄으로써 대응하는 티켓을 개시자 SACM에 요청한다. KDS가 대응하는 타겟 SACM의 PVF와 비밀키를 공유한다면 KDS는 단계 (6)에서 개시자 SACM에 KRB_TGS_REP 메시지를 반환한다. 메시지는 요청된 타겟 응용 서버를 위한 서비스 티켓 뿐만 아니라 마찬가지로 대응 키 매체를 포함한다.

단계 (7)에서 개시자 SACM은 서비스 티켓, 기밀성과 무결성 키 seed를 포함하고 있는 대화 키 패키지 및 PAC을 포함하는 SES_INIT_CTXT 메시지를 생성 한다. 응용 프로토콜이 개시자 SACM으로 부터 타겟 SACM에 연결 설정 토큰을 나를 때, 검증을 위하여 나중에 PVF로 그것을 전송 한다. PVF는 기본 키를 추출하기 위해 키 적용 정보를 처리하고, 기밀성과 무결성 두개의 대화키를 생성하기 위해서 그 대화 키 패키지를 사용한다. 만일 티켓이 유효하고 상호 인증이 요구된다면, 타겟 SACM은 단계 (8)에서 개시자 SACM에게 SES_INIT_CTXT_COMPLETE 메시지를 반환 한다.

이제 개시자와 타겟 SACM 사이에 보안 연결이 성립 되었고 어떤 응용이 데이터 전송을 위하여 이 보안 연결을 사용할 수 있다. 응용이 완료된 후에 연결 종료가 요청 되면 SES_CTXT_ABORT 메시지가 종료 요청을 위해 전송 된다.

SESAME V3는 실체 인증은 물론이고 데이터 기밀성, 무결성 서비스, 권한 부여 및 접근 제어 서비스를 제공 한다. 결과적으로, SESAME는 Kerberos와 다른 선택 대안을 제공한다. SESAME의 다른 장점은 시스템이 독점

적인 개발이 아니라는 사실이다. 대신에, ECMA의 표준 작업에 근간을 두고 어떤 특정 시스템 플랫폼 또는 통신 프로토콜에 묶여 있지 않다. 시작 초기에는 SESAME 인증 및 키 분배 시스템이 다수의 판매업체 환경을 위하여 설계 되었는데 이러한 설계 목적은 시스템이 대형의 협력적 조직들에서 통합이 이루어질 때 효용성이 있을것이다.

4. 결 론

본 연구에서는 분산 시스템 환경에서 통신에 개입된 주체들 사이의 실체 인증 기법에 관하여 위협요소, 인증 서비스 요구조건 및 인증 방법을 조사하고, 인증 메커니즘 사례를 실체 인증의 관점에서 비교 분석하였다.

분산 시스템에 개입된 주체들 사이의 가장 심각한 위협요소는 불법적인 변조, 도청, 신분 위장 및 재전송 등이 있다. 실체 인증 기법은 이러한 위협들에 대하여 기밀성 및 무결성 서비스와 함께 신분 확인 서비스를 제공함으로써 안전한 메시지 교환을 보장할 수 있다.

인증 서비스는 원리의 측면에서 "proof by knowledge"(P1), "proof by possession"(P1), "proof by property"(P3)의 3가지 범주로 제공될 수 있다. 대부분의 인증 프로토콜은 (P1)과 (P2)의 기능을 이용하고 있으며, 인증 서비스는 액세스 제어 수행 이전의 실체 인증, 데이터 인증의 연속성 보장에 관한 무결성, 그리고 기밀성 및 부인봉쇄등을 위한 키 교환의 측면에서 복합적으로 연합되어 제공될 수 있다.

Kerberos는 클라이언트와 서버의 환경에서 인증서버 및 티켓-승인 서버를 두어 다단계 인증 서비스를 제공하는 메커니즘을 제공하고 있다. 그러나, Kerberos는 전적으로 관용 암호화 방식에 의존하고 있으며, 항상 사용 가능한 온라인 서버를 요구하고 재전송 공격 방어를 위하여 타임스탬프에 의존적이므로 동기화 되고 안전한 클럭이 요구된다.

X.509 는 일반적인 정보통신 네트워크 환경의 폭 넓은 디렉토리 서비스를 정의하고 있는 X.500 시리즈 표준중에서 인증 구조를 정의하고 있는 부분이다. X.509 인증 구조의 핵심은 각 사용자와 관련된 공개키 인증서 이다. 이 인증서를 이용하여 일방향, 양방향, 세방향의 선택적인 인증 서비스를 제공할 수 있다. Kerberos와 비교하여 X.509는 물리적으로 안전한 온라인 서버를 요구하지 않는 장점이 있다. 또한, 세방향 인증에서는 Kerberos가 타임 스탬프에 의존적인 문제를 극복할 수 있다.

SPX는 주로 X.509, GSS-API 및 RFC 1507 표준과 연관이 있으며, 인증은 관용 및 공개키 암호방식을 이용하여 수행된다. 그러나, 이러한 표준과 관련된 다양한 응용성에도 불구하고 시스템이 DEC의 연구 실험실 이외에는 널리 사용된적이 없으며, 버전 2.4이후에 더 이상 개발되지 않고 있다.

SESAME는 개방형 분산 시스템에서 공개키 암호 기법에 기반한 보다 정교한 인증, 접근제어, 기밀성 및 무결성 서비스를 제공한다. SESAME는 ECMA, GSS-API 및 X.509 표준을 따르고 있으며 인가를 통하여 공개 사용이 가능하나 상업적 버전의 시스템이 공개적 영역에서 사용되는 버전보다 좋고 전문적 지원을 받을 수 있음은 물론이다.

결과적으로, 키의 운용 측면에서 Kerberos는 전적으로 관용암호 방식을 이용하는데 비하여 X.509는 특별한 방식을 지정하지 않지만 RSA 공개키 암호방식을 권고하고 있다. SPX 와 SESAME는 관용 암호 방식과 공개키 암호 방식을 복합적으로 사용하고 있으며, 두 가지 메커니즘 모두 X.509 권고 사항에 밀접하게 연관된 인증 구조를 사용하지만 SESAME는 주로 ECMA 표준에 영향을 받는다. 그러나, 이러한 메커니즘들은 각각 다른 단체에 의하여 별도의 시스템 환경에서 자신들 고유의 설계

목적을 갖고 개발되었고, 시스템의 규모 및 키의 운용 체계가 다르므로 각 기관 자신의 응용 환경에 가장 적합한 보안정책을 수립하고, 기존의 컴퓨터 운용체계와 연관하여 고유의 인증 메커니즘을 꾸준히 연구해야 할 것이다.

참 고 문 헌

[1] William Stalling, "Network and Internetwork Security", Prentic Hall, pp.315-340, 1995.

[2] Thomas Y.C. Woo and Simon S. Lam "Authentication for Distributed Systems", Computer, Jan. 1992.

[3] ISO/IEC DIS 10181-2, "Information technology- Open Systems Interconnection - Security Frameworks for Open Systems: Authentication Framework", 1993.

[4] Warwich Ford, "Computer Communication Security"(Principles, Standard Protocols and Techniques), Prentice Hall, pp.377-399, 1994.

[5] CCITT Recommandation X.509 : The Directory-Authentication Framework, 1993.

[6] Kaufman, R. perlman and M. Speciner, "Network Security", Prentice Hall, 1995.

[7] Bellovin, S., and Merritt M., "Limitations of the Kerberos Authentication System", Computer Communications Review, Oct. 1990.

[8] I' Anson, C., and Mitchell, C. "Security Defects in CCITT Rec. X.509 - The Directory Authentication Framework", Computer Communications Review, April, 1990.

[9] ISO/IEC 9798-1 : Information technology -Security technique - Entity Authentication Mechanisms - Part 1 : General Model, Sept. 1991.

[10] M. Burrows, M. Abadi and R. Needham, "A logic of authentication", ACM Trans. on Computer Systems 8(1), pp.18-36, 1990.

[11] D. Gollmann, T. Beth and F. Mamm, "Authentication Services in Distributed Systems", Computers & Security, 12, pp. 753-764, 1993.

[12] CCITT Recommendation X.500 : The Directory-Overview of Concepts, Models and Services, 1993.

[13] P. J. Bumbulis, D. D. Cowan, C. M. Durance, and T.M. Stepien, "An Introduction to the OSI Directory Services", Computer Networks and ISDN Systems, 26, pp.239-249, 1993.

[14] S. Mendes, and C. Huitema, "A new Approach to the X.509 Framework : Allowing a Global Authentication Infrastructure Without a Global Trust Model", Symp. on Network and Distributed System Security, pp.172-189, 1995.

[15] J. Tardo and K. Alagappan, "SPX: Global Authentication Using Public Key Certificates", in Proceedings of the IEEE Symposium on Security and Privacy, IEEE Computer Society Press, Los Alamitos, CA, 1991, pp.232 -244.

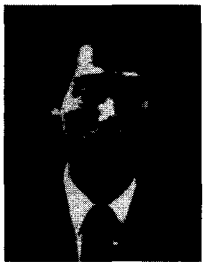
[16] Tom Parker, "A Secure European System for Applications in a Multi-vender Environment", in Proceedings of the 14th National Computer Security Conference, 1991.

[17] Tom Parker and Denis Pinkas, "SESAME V4-OVERVIEW", Dec. 1995.

□ 著者紹介

강 창 구

1975년 한국항공대학 항공전자공학과 졸업(공학사)
1986년 충남대학교 대학원 전자공학과(공학석사)
1993년 충남대학교 대학원 전자공학과(공학박사)
1979년 ~ 1982년 한국공군 기술장교
1987년 ~ 현재 한국전자통신연구원 부호5실장 책임연구원



최 용 락

1976년 중앙대학교 전자계산학과
1982년 중앙대학교 전자계산학과 석사
1989년 중앙대학교 전자계산학과 박사
1982년 ~ 1986년 한국전자통신연구원 선임연구원
1986년 ~ 현재 대전대학교 컴퓨터공학과 교수
1997년 한국통신정보보호학회 충청지부 지부장

※ 관심분야 : 운영체제, 분산처리체제, 컴퓨터통신보안