분산 응용을 위한 보안 서비스 API 설계 모델

°김수광*, 강창구**, 최용락*
* 대전대학교 컴퓨터공학과, ** 한국전자통신연구원

A Design Model on Security Service API for Open Distributed Applications

°Su-Kwang Kim*, Chang-Goo Kang**, Yong-Rak Choi*
* Taejon Univ., ** ETRI

요 약

본 논문은 개방형 분산 시스템에서 여러 분산 응용 프로그래머들에게 보안 서비스 제공을 위한 설계모델에 대하여 분석한다. 범용 보안 서비스는 응용 프로그래머에게 공통적으로 사용할 수 있는 암호 알고리즘 및 인터페이스를 제공함으로써 프로그래머가 암호 알고리즘을 알지 못하여도 보안 서비스를 제공할 수 있다. 따라서, 본 논문에서는 다양한 응용 환경 또는 시스템 하부 구조에 독립적으로 사용될 수 있는 범용 암호 서비스 구조를 설계하기 위하여 PKI, GSS-API 및 GCS-API를 중심으로 각각 관련된 요구사항과 설계 모델을 분석 한다.

1. 서론

1992년에 미국의 엘 고어 부통령에 의하여 제시된 NII(National Information Infrastructure)을 시작으로 유럽의 Big Bang, 일본의 신사회 자본, 싱가포르의 TI2000등 초고속 정보 통신망 계획을 전세계적으로 추진하고 있다[1]. 특히, 인터넷을 기반으로 한 컴퓨터 통신망의 국제적 연동과 더불어 국제 사회의 제반 활동이 실시간으로 이루어지는 정보 사회의 개방성 및 편의성과 함께 이를 이용하는데 따른 정보 노출 등에 대한 역기능 또한 심각히 우려되고 있다.

개방형 분산 환경에서 정보에 대한 위협으로는 신분위장, 재전송, 도청, 부인, 불법적인 조작등 수많은 종류의 공격 형태가 있을 수 있다. 이에 대응한 서비스로서 실체인증, 부인봉쇄, 접근 제어 등의 기법이 연구 개발되고 있는데, 개방형 범용 정보 통신망을 이용하는 각종 응용에서는 국제적 표준안을 준수하면서, 시스템의 사용자 주체 및 객체 사이의 실체 인증과 기밀성, 무결성 등 다양한 형태의 보안 서비스가 지원되어야 한다[2,3].

그러나, 각 응용별 필요한 때마다 별도의 보안 구조를 설계하는 것은 대단히 비효율적이므로, 오랜 기간 동안 관련 공동체 기관들은 범용 보안 서비스의 필요성을 절감해 왔다. 1980년대에 Kerberos 및 DES(Data Encryption Standard)는 산업계에서 유효한 일종의 공통적 보안 서비스이었다. 그러나, Kerberos 보안 서비스에 대한 API는 Kerberos에 제한적이었고, 다른 보안 구조에서는 편리하게 사용될

수 없다[4].

범용 보안 API(Application Programming Interface)의 부족은 프로그래머들이 보안 서비스를 갖는 응용 분야 개발에 많은 어려움을 경험하게 하였다. 이러한 환경을 고려하여 국제적으로 OSI(Open System Interconnection), X/OPEN 및 IETF(Internet Engineering Task Force) 공동체들은 X.500, GCS(Generic Cryptographic Service), PKI(Public Key Infrastructure), GSS(Generic Security Service)등의 다양한 표준화 권고 사항들의 연구 결과를 발표하고 있다[5~8].

따라서, 개방형 초고속 정보 통신망의 활발한 응용 서비스를 위해서는 이러한 범용 보안 구조들의 심도 있는 조사분석과 함께 적용 모델을 정립하고 합리적인 정합 연구를 통하여 적기에 활용 기반을 마련할 필요가 있다. 본 논문에서는 분산 응용 프로그램의 개발에서 범용으로 사용할 수 있도록 PKI, GCS-API 및 GCS-API의 설계와 관련된 요구사항을 조사하고 설계 고려사항을 분석한다.

2. 공개키 기반 구조분석

디지털 서명 기술을 포함하는 공개키 암호방식은 기밀성, 무결성, 발신처 인증 등의 단대단 정보보호 서비스를 제공함으로써 이러한 공격들을 방어하기 위한 통합적 역할을 할 수 있다. 공개키 기반 구조(PKI : Public Key Infrastructure)는 공개키 암호 방식의 사용을 효과적으로 지원하기 위하여 NSA, IETF, CommerceNet등의 기관에서 폭 넓게 추진되고 있는 자동적인 공개키 관리 체제이다[9,10].

2.1. PKI 요구 사항 분석

공개키의 전달은 각 통신 실체들이 공개적으로 공표하는 방식, 공개적 사용 가능한 디렉토리 이용 방식, 공개키 관리 기관에 위탁하는 방식, 그리고 공개키 인증서를 이용하는 방식의 4 가지로 분류 할 수 있다. 현재, 시스템의 부담을 줄이고 융통성 있는 키 분배를 위하여 Kohnfelder에 의하여 처음 제안된 공개키 인증서를 이용한 방식이 널리 연구되고 있다[11]. 이 방식은 공개키 기관과는 직접적으로 접촉하지 않으면서 신뢰 된 공개키 기관으로부터 직접 공개키를 획득하는 것과 같은 효과가 있다.

각 인증서는 인증기관에 의하여 생성된 공개키와 기타 정보들이 포함되어 있으며, 해당 사용자에게 대응되는 개인키가 주어진다. 공개키 인증서를 사용하는 방식은 다음 요구 사항을 만족 시켜야 한다.

- 사용자는 인증서 소유자의 이름과 공개키를 결정할 수 있는 인증서를 읽을 수 있어야 한다.
- 사용자는 인증기관에 의하여 생성된 인증서를 검증할 수 있고 위조할 수 없어야 한다.
- 인증서는 합법적 인증기관에 의하여 생성 및 변경할 수 있어야 한다. 이러한 요구 조건들 외에 Denning은 다음의 요구 사항을 추가하였다[12].
- 사용자는 인증서의 유효 기간을 검증할 수 있어야 한다.

PKI는 X.509에서 정의된 공개키 인증서를 기반으로 각종 정보보호 서비스를 제공할 수 있는 종합적 메커니즘이다. 사용자는 다양한 정보보호 서비스 제공을 위하여 PKI 사용에 따르는 비용과 성능이 보장된 사용자 편의성 제공, 사용자의 각종 거래에 대한 법적 효력 유지, 그리고 사용자의 보안 정책 및 관리 체계의 반영이 가능한 구조를 요구할 것이다[15].

또한, 사용자는 PKI를 이용하여 인증서의 생성, 획득, 취소, 검증 등의 기능을 수행하고 PKI를 기반으로 하는 실체 인증, 메시지 무결성 인증 및 메시지 변조 검출, 기밀성을 위한 비밀키 교환 방식의 지원, 그리고 정보보호 서비스를 위한 기타 관련 정보의 제공을 위한 하부의 기반구조로 활용할 수 있도

록 설계되어야 한다.

PKI는 이상과 같은 사용자 요구 사항을 만족시키기 위하여 관리 및 운영적 측면에서 다음과 같은 요구 사항을 고려하여 개발되어야 한다.

가. 일반 요구 사항

(1) PKI를 기반으로 한 국내외의 안전한 거래를 지원 가능

(2) 법적 효력을 갖는 개방형 범용 정보보호 서비스 체제 구조 제공

(3) 특정 거래에 대한 기관 감사 및 합법적 통제력 확보

(4) 국가적 차원의 표준 및 보안 정책 반영 가능

나. 개발 구조 요구 사항

(1) 대형 분산 네트워크에서 일관된 정책하에 효율적 사용 지원을 위하여 융통성과 확장성을 갖는 CMA(Certification Management Authority) 구성정책

(2) 강한 안전성을 보장하는 키와 알고리즘, 신분 확인, 키 에스크로우 등의 기능과 PKI 이용 극대화를 위한 인증서 구조 선택

(3) 디렉토리 서버를 이용한 인증서의 분배 및 취소와 법적 효력을 갖는 기록 보존 정책 등의 인증서 이용을 위한 효율적 관리 체계 구성

(4) 특허 관련 유용한 서명 표준 및 관련 알고리즘을 수용하고, PKI와 관련된 인증기관, 디렉토리 서버, 사용자 및 각 영역간의 상호 연동성 보장

(5) PKI 실체들의 보존, 로깅, 감사 지침을 설정하고, 하위 기관들의 보안 요구 사항을 수립하며, 각 하부구조 기관들의 정기 감사 및 위반 사항을 탐지하기 위한 각종 상하위 정책설정 기능 포함

2.2. PKI 기능 분석

2.2.1 PKI 기능 및 특성

PKI 는 사용자 공개키 및 인증서 이용에 대한 키 생명 주기 관리, 분산된 네트워크에서의 인증서 관리, PKI 자체 보안 서비스, 타임 서비스, 그리고 국제적 표준과 관련된 상호 동작성 지원 기능 등으로 분류하여 해석할 수 있다[16,17].

기밀성 및 서명 등의 정보보호 서비스에 사용되는 키의 생명주기 관리를 위하여 키의 생성, 복구, 분배, 취소, 정지, 부인, 보존 기능을 지원해야 한다. 또한, PKI는 허가된 정책조건 및 특권에 따라서 법 집행을 위한 검색, 협동적 대행자 검색 및 개별적인 검색 등의 인가된 검색 시나리오를 지원할 수 있어야 한다.

PKI는 거래 또는 사업 영역의 요구 조건에 따라 분산된 인증서 관리 기능을 제공해야 한다. 즉, 키의 생성, 복구, 분배, 취소, 정지, 부인, 보존, 인가된 검색을 포함하는 모든 정책의 생성관리 및 공개키와 디렉토리 이름을 연결하는 인증서의 분산 관리기능을 제공하고, 인증기관 사이에 크로스 인증서를 설정하기 위하여 매핑하는 서비스 및 인증서 저장 기능과 인증 기관의 분산된 영역 지원기능을 보유해야 한다.

PKI 자신은 PKI 서비스의 기밀성, 무결성, 가용성을 보장하고 PKI 서비스 자신들 행위에 대하여 부인 봉쇄 서비스를 제공하며, 사용자 및 가입자들 자신의 행위에 대한 부인 봉쇄 서비스를 지원해야 한다.

그리고, 국제적으로 네트워크화 된 타임 서비스가 타임 스탬프 이용을 위하여 유용해야 한다.

다른 업체들에 의하여 제공되는 PKI 요소들이 상호 동작할 수 있도록 인증서와 관련된 데이터의 국제

적 표준을 지원하고, 또한 모든 인증서 서비스의 국제화를 지원해야 한다.

2.2.2 PKI 객체 및 구성 요소

PKI는 요구되는 보안 서비스를 지원하기 위하여 5 가지의 기본 객체를 관리해야 한다. 인증서는 서명을 지원하는 공개키 인증서, 기밀성 목적을 위하여 키 교환을 지원하는 키 관리 인증서, 사용자의 특권 및 직무를 명시하는 애트리뷰트 인증서 3가지의 기본적 종류로 구분할 수 있다.

PKI는 CA들 사이의 양방향 신뢰 체인을 구축하기 위하여 X.509 크로스-인증서-쌍 구조를 지원해야 한다. 크로스-인증서-쌍 구조는 전방(forward) 인증서와 후방(reverse) 인증서 2 가지 인증서를 포함 한다. 전방 인증서의 주체는 후방 인증서의 발행자이며, 거꾸로 후방 인증서의 주체는 전방 인증서의 발행자이다.

크로스-인증서-쌍은 구성 방식에 따라서 계층적 구조에서 상하 관계에 있는 계층적 크로스 인증서, 인증 경로 축소를 위한 전파 규칙이 적용된 일반적 크로스 인증서, 종단 인증 기관 사이의 직접 인증을 반영하는 특별 크로스 인증서가 있을 수 있다.

인증서 취소 리스트(Certificate Revocation List : CRL)는 X.509V.2 CRL에 기반하여 제공할 수 있다. CRL의 모듈과 표현은 중요한 성능 문제를 갖는다. 어떤 응용은 취소된 인증서에 대응하고, 시간 프레임과 취소된 이유에 의존하는 서명을 받아들일 수 있다.

손상된 키 리스트(Compromised Key List : CKL)는 하부구조를 통하여 손상된 키의 고속 배포를 지원할 수 있다. 이것은 모든 CA가 공통의 CKL형식을 지원하고, 하부구조 내에서 교환할 CKL을 생성하도록 요구한다. CKL의 생성 주기는 기술적 정책 문제이다.

또한, PKI는 CA에 의하여 발행되는 보안 정책을 지원하고, 정책 정보에 대한 표준 형식을 지원해야 한다.

한편, PKI를 구성하고 있는 기능적 주요 요소들은 크게 나누어 인증서 관리 기관, PKI 클라이언트, 그리고 디렉토리 서버 3 종류가 있다.

인증서 관리 기관은 융통성 및 확장성을 고려하여 다단계의 계층구조로 구성 할 수 있으며, 일반적으로 다음과 같은 기관들이 포함된다. PAA(Policy Approving Authority)는 PKI에 대한 종합적 지침을 생성하는 정책 인증 기관으로써, 국가적 인증서 관리 하부구조의 루트가 된다.

PCA(Policy Certification Authority)는 각 PCA는 그것의 영역 내에 있는 모든 인증 기관과 사용자들에 대한 정책을 설정 하는 두번째 요소로써 정책과 인증의 책임을 갖는다.

CA(Certification Authority)는 ORA와 사용자 조합을 갖는 PCA아래 요소로써 최소 단위의 정책 책임을 갖는 인증 기관이다.

ORA(Organizational Registration Authority)는 ORA는 CA와 사용자 사이의 중간 매체로 작용하는 엔티티로써 CA와 관련된 사용자의 가입과 등록을 관리한다.

PKI클라이언트는 X.509 디렉토리 서버로부터 인증서를 획득하고 인증서 상태의 결정, 정책 식별자 및 제한 필드들의 해석, 디지털 서명의 생성 및 검증 등의 기능을 갖는다.

PKI 디렉토리 서버는 발행된 모든 인증서들이 요청에 따라서 사용될 수 있도록 필요한 액세스 프로토콜 및 인증서 보관 기능을 지원 한다.

2.3. PKI 설계 모델

2.3.1 PKI 인증서 관리 구조

PKI에서 신뢰는 인증서의 체인으로 구성된 인증 경로에 따라 처리된다. 인증기관은 계층적이고 순서

적으로 인증서들을 발행하거나, 또는 보다 융통성 부여를 위하여 비계층적으로 발행할 수 있다. 인증 경로의 구성 방법에 따라 시스템의 오버헤드 및 편의성이 달라 질 수 있다.

계층 구조는 인증 기관들이 루트 CA 밑에 계층적으로 정렬되는 구조이다. CA들은 계층 구조에서 자신의 밑에 CA와 사용자에게 인증서를 발행한다. 루트 CA의 공개키는 모든 사용자에게 알려지고, 어떤 사용자의 인증서는 루트 CA로 연결되는 역방향 인증 경로를 따라 검증될 수 있다. 이 구조는 단순하고 인증경로 탐색이 용이하지만 루트 개인키의 손상 결과는 심각한 피해를 초래하는 단점이 있다.

네트워크 구조는 CA들 사이의 신뢰 관계를 반영하여 네트워크 형태로 독립된 CA들이 서로 인증 한다. 사용자는 자신에 근접한 CA의 공개키를 알고, 그것과 연결되는 신뢰 된 CA로 역방향 인증 경로를 따라 다른 쪽 사용자의 인증서를 검증할 수 있다. 상업적 상호 신뢰 관계를 반영하는 구조에 유리하며, 인증 경로 탐색이 복잡하고 다양한 인증경로에 따르는 문제점이 있다.

복합형 구조는 계층적 구조 및 네크워크 구조의 특성을 조합시킨 형태이다. 인증 경로를 효율적으로 처리하기 위한 계층적 크로스-인증서, 일반적 크로스-인증서, 특별한 크로스-인증서 3가지 형태의 크로스-인증서가 있다. 이 구조는 인증경로를 축소하고, 성능을 향상 시키기 위하여 종단 CA간에 특별한 직접 인증관계의 설정이 가능하다.

한편, 인증 경로 구조는 누가 보안 정책을 수립하고, 어떻게 그 정책에 사용자들을 조직하는가에 따라서 COI구조, 기관에 따른 구조, 보증단계에 따른 구조로 나누어 볼 수 있다.

COI(Communities Of Interest) 구조는 사용자들이 자주 거래하는 관심 주제에 따라서 그룹을 형성하는데 착안한 구조이다. 가장 자주 통신하는 사용자들은 그들이 물리적으로 멀리 떨어져 있어도 PKI에서는 가까이 인접한 관계일 것이다. 각 공동체는 사용자들이 수행하는 기능에 따라 그룹을 형성하고, 자신들의 보안 정책을 수립할 수 있다. 이것은 인증 경로가 거래 사용자간에 특별히 설정되어 저장할 인증서의 갯수를 최소한으로 줄일 수 있을 것이다.

기관에 따른 구조는 어떤 기관의 현재 계층적, 또는 부서별 조직 관계를 반영한 구조이다. 하나의 기관내에 사용자들이 그룹을 형성하고, 자신들의 보안 정책을 수립할 수 있다. 이것은 기관의 계층적 상하 관계가 분명한 조직에서 유리하며 구현이 용이하다.

보증 단계에 따른 구조는 적은 갯수의 보안 정책으로 만족시킬 수 있는 환경에서 보증하는 수준에 따라 구성된다. 어떤 등급의 보증 수준을 나타내는 대상자별로 그룹을 형성하며, 기본에서 부터 엄격한 보증 영역까지 3-4가지의 비교적 적은 수효의 보안 정책을 수립한다.

복합형 구조는 COI구조, 기관에 따른 구조, 그리고 보증 단계에 따른 구조 3 가지의 형태를 각각 하나의 세그먼트로써 허용하는 복합적인 구조이다. 한 국가의 종합적인 환경은 정부 기관, 일반 상거래 관계의 기업, 또는 특수한 분야의 서비스 조직 등 다양한 형태를 포함한다. 따라서, 이와 같이 다양한 속성의 사용자 및 기관을 수용하기 위해서는 융통성과 확장성이 있어야 할 것이다.

2.3.2 PKI 영역 구조

하나의 PKI 영역에 있는 사용자가 다른 영역에 있는 사용자와 상호 동작하는 경우에 관계를 단순히 표현하면 <그림1>과 같다.

① CMA는 인증의 계층적 구조 PAA, PCA, CA를 총괄하여 표현한 것으로써 이들 사이에는 어떤 신뢰의 체인으로 연결되어 있다.

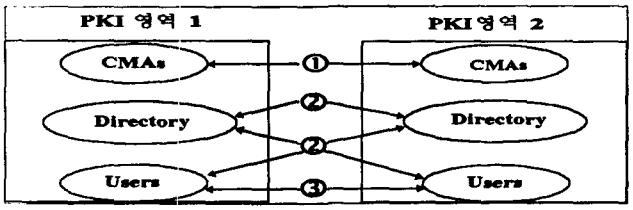
② 사용자는 다른 영역에 있는 디렉토리를 액세스하여 인증서 체인 및 CRL들을 얻을 수 있어야 한다.

③ 사용자는 상호 작용하여 각각의 응용 프로토콜을 수행할 수 있어야 한다.

PKI 영역 간에는 각각의 환경에 따라 다양한 측면에서 서로 다른 동작 체계를 갖고 있을 수 있으므로 각 영역 사이에는 다음과 같은 차이에서 오는 문제점을 해결해야 한다.

- 서명 알고리즘 : 공개키 암호방식의 알고리즘 차이

- 키 분배 알고리즘 : 영역간 키 분배 방식 차이
- 데이터 암호 방식 : 영역간 다른 암호방식 사용
- 데이터 형식 : 인증서/CRL형식, 표준, 선택사항 차이
- PKI 구조 : PKI 계층 구조의 차이
- 데이터 보급 : 데이터 보관/보급 방법의 차이
- 정책 : 상호 인증서 정책의 차이
- 이름 : 영역간 이름 사용 체계의 차이



<그림 1> PKI 영역 구조

2.3.3 PKI 시스템 구조

PKI 구조에서 포괄적인 기능적 범주는 <그림 2>와 같이 나타낼 수 있다. 시스템(System Security Enabling Services) 기능은 사용자 로그-온, 사용자 신임장 획득, 그리고 사용자의 프로세스 및 스레드에 관한 보안 상태 정보를 제공한다. 예를 들면, 일반 사용자가 스마트 카드 자신을 인증하여 신임을 얻으면, 그 사용자 프로세스는 스마트 카드에 저장된 개인키를 사용하여 데이터를 서명하기 위해서 스마트 카드 인터페이스를 사용할 수 있어야 한다.

암호 원시함수(Cryptographic Primitives) 요소는 키 생성, 데이터 버퍼에 대한 해쉬 함수 적용, 그리고 공개키 및 비밀키 알고리즘을 사용한 데이터 암복호화와 같은 하위 수준의 암호학적인 원시 함수에 대한 접근을 제공한다.

암호 서비스(Cryptographic Services)는 데이터 무결성, 기밀성, 디지털 서명등과 같은 암호학적 서비스에 대한 접근을 제공한다. 암호학적 연결 관리는 암호학적 서브시스템의 상태를 활성화 시키고, 사용 이후에는 암호학적 서브시스템의 상태를 해제하는 설비를 제공한다.

키 서비스(Long-term Key Service)는 인증서의 관리 기능을 포함하여 종합적인 키 생성 주기 관리 기능을 제공한다. 키의 생명 주기 관리는 키의 취소, 부인, 유효 기간 처리 등 다양한 관련 서비스를 제공한다.

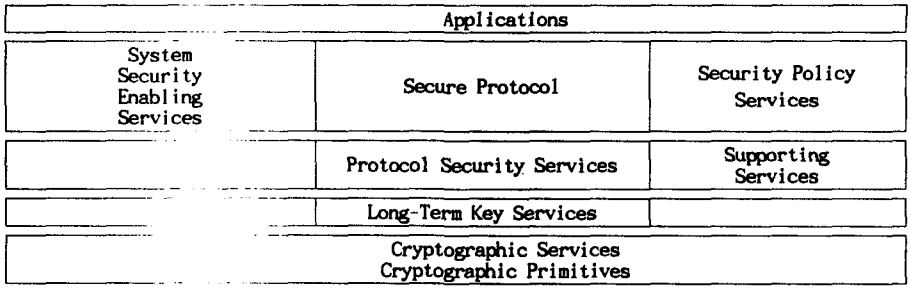
프로토콜 보안 서비스(Protocol Security Services)는 프로토콜의 측면에서 설계자가 이용하는데 적합한 보안 서비스를 제공한다. 세션 지향형에서 보안 서비스는 프로토콜 교환과 연관된 보안 상태 정보 유지를 요구하며, 축적 교환 방식에서는 보호된 메시지 토큰 내에 요구되는 보안 상태 정보를 포함한다.

안전한 프로토콜 서비스(Secure Protocol Service)는 보안 서비스에 대한 어떤 호출을 요구하지 않고 통신 파트너 사이에 보호된 데이터 전송을 제공한다. Secure Protocol을 이용하는 응용은 프로토콜 교환을 시작하기 전에 희망하는 보호의 수준을 명시하여야 한다.

보안정책 서비스(Security Policy Services)는 사용자 및 다른 통신 실체들의 특권과 자원 접근 제어 정책을 관리하고, 그 정보에 기반하여 접근제어 결정을 수행하며 사용자 정보관리 기능을 수행 한다.

지원 서비스(Supporting Services)는 보안 서비스에 의하여 요구되거나 네트워크 시스템의 안전한 오

퍼레이션을 위하여 요구되는 보안 감사, 타임 서비스 및 기타 디렉토리 지원기능들을 제공한다.



<그림 2> PKI 시스템 구조

3. GSS-API

GSS-API는 개방형 분산 네트워크 환경에서 응용프로그램의 소스레벨 이식성을 지원하기 위하여 정보보호 서비스에 대한 범용 인터페이스 규정을 위하여 제안된 표준이다. 1990년대 초반까지 괄목할 만한 정보보호 API가 없던 때에 1992년 Linn이 IETF에 범용 정보보호 API를 제안하였다[18].
IETF는 1993년에 그것을 인증하였고 RFC 1508 및 RFC 1509 로써 문서화 하였다[19].
인증이 된 이래 널리 수용이 되어 GSS-API는 인터넷 관련 기관, X/Open 및 ECMA 등에서 다양한 메커니즘을 기반으로 구현되고 있다. GSS-API는 통신 프로토콜의 독립성을 제공하고 정보보호 서비스가 어떻게 구현되는지 상세한 지식을 요구하지 않는 장점이 있다.

3.1. GSS-API 요구사항 분석

GSS-API는 분산 응용프로그램 환경의 넓은 영역에서 동작할 수 있도록 시도되었다. 그리고, 새로운 응용프로그램의 개발 또는 기존 시스템에서 추가적 개발을 위하여 사용될 수 있도록 범용 정보보호 서비스에 대한 간편하고 단순한 인터페이스 개발 요구를 만족시키는 것이 목적이다. 따라서, GSS-API는 다음과 같은 요구사항을 갖는다.

- 메커니즘 독립성 : 정보보호 서비스에 대한 인터페이스는 암호 서비스를 수행하는 상세한 구현과 독립적으로 제공된다. 특히, 공개키 및 비밀키 암호 시스템 사이에 구분이 없고, API 수준에서 어떠한 차이가 없이 수많은 다른 시스템들을 사용하여 구현될 수 있다.
- 프로토콜 독립성 : API는 하부 통신 프로토콜과 독립적이어야 한다. 즉, GSS-API는 TCP/IP 그리고 RPC와 같은 통신 설정을 위하여 사용된 특정한 응용프로그램 프로토콜 등과 독립적으로 동작한다.

메커니즘과 프로토콜 독립성은 GSS-API를 사용하는 응용프로그램의 소스레벨 이식성을 증가시키고 다양한 플랫폼에서 단일 구조의 구현을 허용한다. 이것은 또한 응용프로그램 프로그래머에게 다양한 정보보호 메커니즘의 상세한 내용을 알 필요가 없게 해준다. GSS-API의 전형적인 사용은 기존의 응용프로그램에 정보보호 기능을 수행시키는 것이다. 즉, 우선 메시지가 무결성 또는 기밀성 서비스를 위하여 GSS-API 인터페이스에 전달되고, 결과 토큰이 GSS-API에 의하여 생성된 다음 수신자 측으로 전달되는

것이다.

3.2. GSS-API 기능 분석

GSS-API는 분산 응용프로그램 환경에서 정보보호 서비스에 대한 범용 인터페이스를 제공한다. GSS-API는 정보보호 서비스의 집합에 대한 범용 인터페이스와 정보보호 서비스를 제공하는 메커니즘들의 2가지 논리적 부분으로 나눌 수 있다.

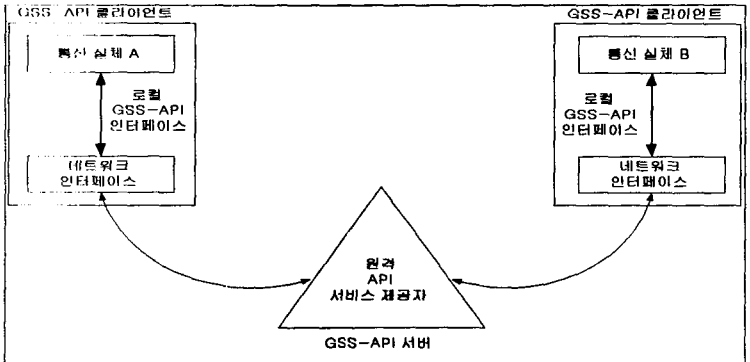
인터페이스는 각각의 알려진 메커니즘과 데이터 변환 및 호출 접속의 역할을 수행한다. 그러나, 메커니즘들은 일반적으로 신뢰된 제 3자 서버로써 전형적인 원격지의 안전한 호스트 형태를 갖는다.

정보보호 서비스를 제공하기 위하여 사용되는 메커니즘은 각 세션의 초기에 선택되고 세션동안 고정되어 있다. 메커니즘 선택은 GSS-API 구현에 사용 가능한 메커니즘에 영향을 받으며, 응용프로그램 선택에 달려 있다.

이러한 구조는 GSS-API가 클라이언트로써의 안전한 응용프로그램과 서버로써의 메커니즘으로 구성된 분산 응용프로그램 서비스임을 의미한다. 본 논문에서는 로컬 GSS-API 인터페이스 코드를 포함하고 있는 응용프로그램을 GSS-API 클라이언트로 부르기로 한다. 그리고 GSS-API와 원격지 부분을 형성하는 GSS-API 관련 메커니즘의 집합을 GSS-API 서버라고 한다.

GSS-API 서버는 메커니즘에서 사용되는 암호 기술에 따라서 약간의 다른 역할을 갖는다. 관용키 방식 메커니즘에서 서버는 키 관리자가 되는 반면에 공개키 메커니즘 방식에서는 인증서 관리자로 동작한다.

GSS-API 서버의 두 가지 경우 모두 제 3자는 양쪽의 통신 실체에 의하여 신뢰되고, 각 실체의 인증을 할 수 있는 위치에 있다. <그림 3>은 응용프로그램들과 GSS-API 서버 사이에 통신 패스와 GSS-API의 분산된 성질을 나타낸다.



<그림 3 > GSS-API의 분산 구조

3.3. GSS-API 설계 모델

RFC1508에서 제시하는 20개의 기본적 GSS-API 호출은 <표1>과 같이 요약할 수 있다. 신임장 관리 호출(Credential Management Calls)은 통신 실체에게 신임장의 획득 및 해제를 지원하고, 또한 여러 가지의 신임장 정보에 관한 조회를 제공한다. 정보보호 문맥의 연결은 연결의 초기화, 허가 및 삭제, 그리고 연결의 유효 시간과 연결 관련된 토큰들을 처리하는 문맥연결 관리 호출(Context Management Calls)

을 사용하여 관리한다.

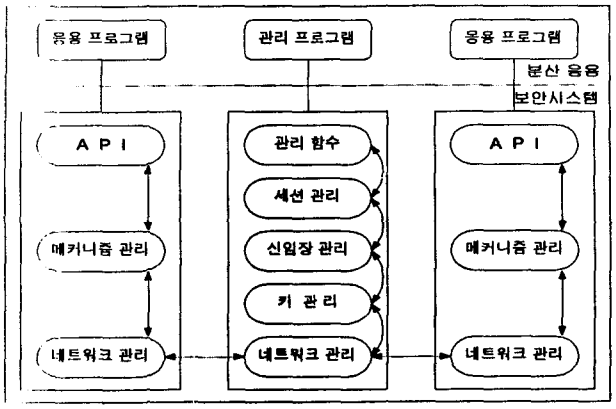
메시지 암호 호출(Per-Message Calls)이라고 알려진 암호학적 호출은 일단 안전한 세션이 연결되면 메시지 단위로 GSS-API 내에서 무결성 및 기밀성에 대한 접근을 제공한다. 마지막으로 지원 관련 호출(Support Calls)은 할당된 메모리를 해제하고 이름의 비교와 같은 일반적인 관리 및 지원 루틴들을 제공한다.

신임장 관리 호출	
gss_acquire_cred()	보안 연결을 시작하거나 얻기 위한 신임장 획득
gss_release_cred()	연결이 끝난 후에 신임장 영역을 해제
gss_inquire_cred()	신임장에 대한 정보를 조회
문맥 연결 관리 호출	
gss_init_sec_context()	나가는 암호 연결에 대한 초기화
gss_accept_sec_context()	들어오는 암호 연결 승인
gss_delete_sec_context()	더 이상 필요 없을 때 연결 종료
gss_process_context_token()	연결된 토큰의 처리 제어
gss_context_time()	연결에 남은 유효 시간 표기
메시지 암호 호출	
gss_sign()	메시지에 서명을 붙임
gss_verify()	서명된 메시지를 검증
gss_seal()	메시지를 서명, 암호화 및 캡슐화
gss_unseal()	메시지를 개봉, 필요시 복호화 및 서명 검증
지원 관련 함수	
gss_display_status()	상태 코드를 출력할 수 있는 형태로 변환
gss_compare_name()	2가지 이름이 같은가 비교
gss_indicate_mechs()	사용 가능한 메커니즘을 표시
gss_display_name()	이름을 출력 가능한 형태로 변환
gss_import_name()	사람이 읽을 수 있는 이름을 내부적 기호로 변환
gss_release_buffer()	할당된 버퍼 기억장소를 해제
gss_release_name()	이름 기억 장소를 해제
gss_release_oid_set()	OID 기억 장소를 해제

<표> GSS-API 호출

이상과 같은 API 호출을 구현하기 위해서는 <그림 4>와 같이 다중 레벨의 내부 구조를 갖도록 설계할 수 있다. API의 상위 레벨은 호출 응용프로그램에서 볼 수 있으며 지원 메커니즘을 포함하고 있는 하위 레벨과 내부적으로 인터페이스하고 있다. GSS-API는 메커니즘에서 제공될 수 있는 정보보호 서비스에 대한 일관성 있는 인터페이스를 제공하기 위하여 암호 메커니즘과 응용프로그램 사이에 존재하는 접착제 역할을 한다.

구현을 위하여 외형적인 API의 구현에 대한 고려뿐만 아니라 다수의 메커니즘들에 이러한 구조의 통합을 고려해야 한다. 구현상의 어려움은 메커니즘의 데이터 구조에 따라 관련 API 내부 데이터 구조를 변환 및 관리하는 구체적인 절차의 구현에 의존한다. GSS-API를 위한 공개키 기반 구조, 특권 허가 서비스 및 보안 정책 등이 향후 구현에 확장하여 고려되어야 할 것이다.



<그림 4 > API와 메커니즘 설계 모델

4. GCS-API

인터넷과 같은 네트워크 서비스의 증가는 분산 컴퓨터 시스템 하에서 보안의 필요성을 증가시키고 있다. 실체 인증, 데이터 출처 인증, 부인봉쇄, 데이터 분리, 기밀성과 무결성 보호를 제공하기 위한 보안서비스들은 암호화 서비스에 의존한다. 그러나, 다음과 같은 상황에 의해 응용프로그램에서 암호학의 공통적 사용은 많은 제한 점들이 있다.

- 합의 된 API의 부족
- 암호학적 기술의 지원과 사용, 수/출입에 관해 적용할만한 법적인 근거의 부족
- 암호 서비스 계층 부재
- 많은 컴퓨터에 각각의 암호 서비스를 제각기 구현하게 함
- 암호 서비스 구현에 있어서 불필요한 노력으로 인한 낭비
- 임의의 암호 메커니즘에 전형적 인터페이스 제공의 필요
- API의 어떠한 변경이 없이 모든 구조에 적합하도록 설계 되어져야 함
- 보안에 관한 전문가 혹은 비전문가에게도 쉽게 암호 서비스를 제공해야 함
- GCS-API메커니즘 구현에 유용해야 함
- 다양한 메커니즘의 접근/사용을 가능하게 해야 함

따라서, 암호화 서비스들을 위한 표준 응용 프로그래밍 인터페이스 명세가 필요하게 되었고, 범용 암호화 서비스의 필요성이 제기되게 되었다. 개방 그룹은 범용 암호화 서비스들을 필요로 하는 NIST(National institute of Standards and Technology), TIS(Trusted Information Systems), NSA(National Security Agency) 등을 중심으로 몇개의 주요 파트너들과 협력하고 있다. 이들 파트너들은 BULL, Hewlett Packard, IBM, ICL, Olibetti, Open Vision, 그리고 Siemems Nixdorf와 같은 회사들과 함께 작업하고 있다.

4.1. GCS-API 요구사항 분석

기본 GCS-API는 암호 알고리즘과 응용에 대한 독립성을 제공하고, 암호학적 서브 시스템과 독립되어 하드웨어와 소프트웨어 어느 쪽에든지 적절하게 수행되어야 한다. 또한, 암호 서비스를 위해 접근하는

운영체제 커널에게 어떠한 제한을 두게 해서는 안되며 확장성을 보장해야한다.

기본 GCS-API는 암호 함수들의 형식에 대한 간략한 소개와 GCS-API구조의 단순화된 모델을 제시하고 있다. 범용 암호화 기능의 최소 집합을 지원하며, 암호화 서비스를 사용하기 바라는 응용 프로그램의 요구를 지원할 수 있다. 일반적인 응용 개발자들이 필요로 하는 암호화 서비스의 대부분은 기본 GCS-API기능으로 수행 될 수 있다.

기본 GCS-API는 알고리즘을 아는/모르는 응용 모두에 암호 서비스들을 제공하고, 암호 서비스들과 키 관리 서비스들에 의존하는 응용 프로그램을 개발하는 프로그래머에 의해 사용될 인터페이스 명세를 제공하는 기능을 갖는다. 기본 GCS-API의 기능적 요구사항은 다음과 같다.

- 데이터 암호/복호화
- 무결성 체크 생성/점검
- 해쉬 생성
- 난수 생성
- 키 생성, 유지, 삭제
- 키의 출입

한편, 상급 GCS-API는 기본 GCS-API와 동일한 요구사항을 갖는다. 그러나, 상급 GCS-API는 GCS-API의 보다 세부적 개념들과 자료구조 및 추가 함수의 집합을 지원 한다.

현재 명세에서의 범위는 단지 암호학을 아는 호출자를 지원하는 서비스만을 고려한다. 인터페이스 명세는 암호학을 아는 사용자나 암호 서비스 및 키 관리 서비스에 의존하는 응용 개발자인 프로그래머에 의해 사용되어진다. 상급 GCS-API의 기능적 요구사항은 다음과 같다.

- 무결성과 점검값 생성
- 데이터 암/보호화
- 반복하지 않은 해쉬 데이터의 생성
- 난수의 생성
- 이용 가능한 키와 키 관련 데이터의 요구
- 키의 생성, 유출, 삭제
- 키의 수/출입
- 키와 그 키와 관련된 정보의 저장과 검색
- 키와 그 키와 관련된 데이터의 구성과 반환

4.2. GCS-API 기능 분석

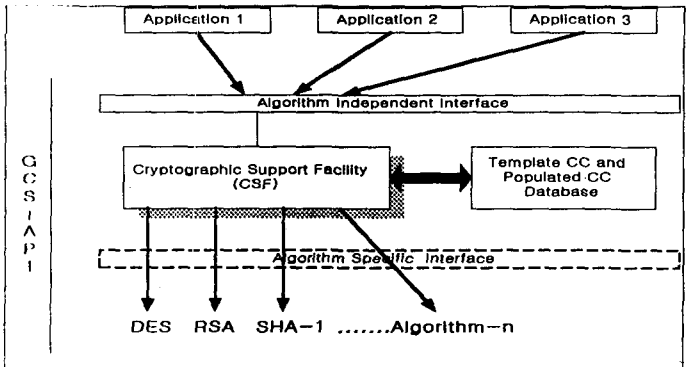
GCS-API는 네트워크 응용 프로그램 개발자들이 보안 메커니즘을 알거나, 알지 못하거나 상관없이 보안 서비스를 받을 수 있도록 지원해 주는 하부 구조이다. 그러나, GCS-API 자체가 어떠한 암호 알고리즘이나, 키 교환 알고리즘을 지원해 주는 것이 아니라, 응용 프로그램이 필요한 보안 서비스를 요청 했을 때, 필요한 암호 알고리즘을 호출 응용 프로그램에 서비스를 수행해 주는 역할을 한다. GCS-API의 개략적 구조는 아래 <그림 3>에서 보는 바와 같다.

응용에서 정보보호 서비스를 요청하면, Cryptographic Support Facility(CSF)는 Cryptographic Context(CC)를 참조하여 필요한 암호 서비스를 호출한다.

여기서 응용은 정보 보호를 요청하는 객체이다. GCS-API는 호환성 있는 응용 개발을 위하여 두 가지 특징을 제공한다. 첫째로 알고리즘 독립적이다. GCS-API는 호출자에게 알고리즘에 관한 세부 사항과 복잡한 부분을 숨긴다. 따라서, 호출자는 사용되는 암호 알고리즘이나 파라메터에 관하여 알지 못하여도

암호 서비스를 제공받을 수 있다. 두 번째로 수행에 독립적이다. 즉, GCS-API는 호출자에게 수행에 관한 세부 사항을 숨긴다. 호출자는 암호 알고리즘의 수행이 하드웨어 또는 소프트웨어 어디에서 수행되든 상관없이 암호 서비스를 받을 수 있다.

또한 GCS-API는 암호 서비스의 여러 정보를 위해 CC라는 데이터베이스를 관리 유지한다. 일반적으로 암호 서비스를 수행하는데 있어서 단순히 데이터와 키에 대한 정보 외에도 어느 알고리즘이 사용되고, 어떻게 사용되는가 등에 관련된 여러 파라메터를 필요로 한다. 따라서, CC는 암호 오퍼레이션의 모든 정보를 캡슐화하여 유지함으로써, GCS-API가 알고리즘 독립적으로 수행할 수 있도록 한다.



<그림 5> 기본 CSF 모델

4.3. GCS-API 설계 모델

GCS-API의 구조는 크게 3개의 영역으로 나눌 수 있다. 그 첫 번째로 응용과의 인터페이스를 유지하는 부분으로써의 CSF영역, 두 번째로 암호 서비스의 모든 정보를 관리하는 CC영역, 마지막 세 번째로 암호 알고리즘 영역으로 나눌 수 있다.

4.3.1 Cryptographic Support Facility(CSF)

가. 서비스 지원 함수

CSF는 인터페이스의 집합이다. 즉 응용에서 CSF에 보안 서비스를 요청하면 CSF는 적당한 암호 알고리즘을 이용하여 응용에 보안 서비스를 제공한다. CSF 서비스는 <표 2>와 같이 CSF 초기화를 위한 함수, 키 관리 함수, CC 관리 함수, 전송 데이터 보호를 위한 함수로 구성된다.

<표 2 : CSF의 함수 분류>

분 류	내 용
Session Management	응용프로그램과 CSF와의 연결을 설정/해제
gcs_initialize_session	세션 초기화에 사용
gcs_terminate_session	세션을 종결하고 보안 연결을 해제
Cryptographic Context Retrieval Function	CC에 대한 처리
gcs_delete_cc	CC에 대한 처리를 제거 / 해제
gcs_list_cc	CC의 검색을 제공한다.
gcs_retrieve_cc	CC를 사용하고 검색할 수 있게 한다.
Key Creation	Key 생성에 대한 처리
gcs_derive_key	입력 파라메터로부터 키를 얻어낸다.
gcs_generate_key	키 값을 생성한다.
Hash and Signature Function	무결성 및 디지털 서명을 제공
gcs_generate_checkvalue	디지털 서명을 생성한다.
gcs_verify_checkvalue	디지털 서명을 검증한다.
gcs_generate_hash	해쉬값을 생성한다.
gcs_generate_random_number	난수를 생성한다.
Data Encipherment Function	데이터의 암호화 즉 기밀성을 제공
gcs_encipher_data	데이터에 대한 암호화를 제공한다.
gcs_decipher_data	데이터에 대한 복호화를 제공한다.
gcs_protect_data	암호화, 디지털 서명, 인증을 제공한다.
gcs_decipher_verify	복호화 및 디지털 서명, 인증을 검증한다.
Cryptographic Context Storage Function	CC 저장 및 삭제를 제공
gcs_store_cc	CC의 저장과 정의된 이름을 할당한다.
gcs_remove_cc	CSF로부터 CC를 제거한다.

나. 함수 권한 정책

사용하고자 하는 특정 함수들은 합법적인 접근 권한을 갖고 있어야 한다. GCS-API에서의 권한 정책은 GCS-API 함수와 접근 권한의 시험으로 정의되고 특정키를 사용한다. GCS-API 호출자들은 호출자가 운용하는 주체에 의해 생성된 키에 접근하거나 생성하는 주체가 허가된 권한을 갖는 키로의 접근에 권한이 부여된다. 이 권한 정책이 집행되고 관리되어지는 메커니즘들은 구현에 의존적이다. 호출자와 CSF사이의 세션 초기화를 위한 지원이 이 명세서에 포함되어 있고 호출자의 신원에 의해서 인증 되고 적절한 접근 제어 정보가 설정된다.

호출자가 특정 키 상에서 수행하는 함수들은 GCS-API의 호출자에게 할당된 기능들의 집합에 기초하는 권한 정책에 의해서 결정된다. 이러한 기능들은 호출자의 활동을 대신하는 주체보다는 호출자 자신에 더욱 관련이 있다. 호출자는 추가적으로 권한이 부여된 함수가 특정 주체가 요구하는 서비스를 허가하는 제어 정책을 집행해야한다.

다. 보안정책

특별한 제어가 분산 시스템 보안에 있어서 중요한 직무에 기인하는 암호 소프트웨어의 사용에 적용되어야 한다. 암호화 소프트웨어의 수출은 많은 나라들에 의해 법적인 제한을 갖는다. 예를 들어, ITAR(USA Government International Traffic in Arms Regulations)은 암호 서비스를 포함하는 제품의 수출을 제한한다. 또 실제로 데이터 기밀성 서비스는 자국에서 공급되고 제어될 수 있게 한다.

CSF구현은 다음과 같은 엄격한 보안 요구를 고려해야 한다.

- CSF는 암호 서비스에 비권한을 갖는 접근을 막아야 한다.

- CSF는 개인 혹은 비밀키와 같은 데이터에 접근을 막아야 한다.
- CSF는 사용 전에 키와 관련된 제어 정보를 검증해야 한다.
- 집행되는 정책에 따라, CSF는 그들이 그 서비스에 접근하기 전에 그것의 호출자가 인증 되는 것을 요구해야 한다.
- GCS-API의 이점은 키는 결코 비인가 된 호출자에 의해 노출된 상태로 참조되지 않는다는 것이다. CSF인터페이스 상위에서 오퍼레이션 키들은 보호되어야 한다.

4.3.2 암호 시스템 구성

암호 서비스 호출자에 대한 서비스 지원 시스템 구성은 <그림 6>과 같이 나타낼 수 있다. 제일 상위 레벨에서 응용은 중간 및 하부구조 서비스들을 통한 데이터 보호 서비스를 요구한다. 이러한 응용은 일반적으로 암호학을 모른다. 그 보다 하부층들은 암호 보안 정책의 상세한 부분까지 책임을 담당하고 암호 구문을 설정한다. 이것은 특정 키 분배 프로토콜과 알고리즘들의 선택을 통하여 내려가면서 안전한 연결 생성의 부분으로써 메커니즘 독립적 키 분배 서비스로부터 진행한다.

이 명세서의 서비스는 CSF에서 구현된다. 다른 계층의 인터페이스에서 나타나는 경계들도 중요하다. CSF 인터페이스는 암호학적 키가 저장되지 않거나 권한이 없는(비암호학 시행) 호출자에 의하여 노출된 상태로 조작되는 것에 대하여 통과할 수 없는 경계를 나타낸다. CSF위에 키들은 암호학적으로 보호된 데이터로 불투명하게 처리되거나 참조된다.

CSF는 암호학의 일반적 집합과 여러 다른 알고리즘의 상위에 위치하는 키 관리 서비스 인터페이스를 제공하고 그 알고리즘들의 다양한 구현들을 제공한다. CSF서비스 인터페이스는 어느 특정 알고리즘을 숨길 수 있다.

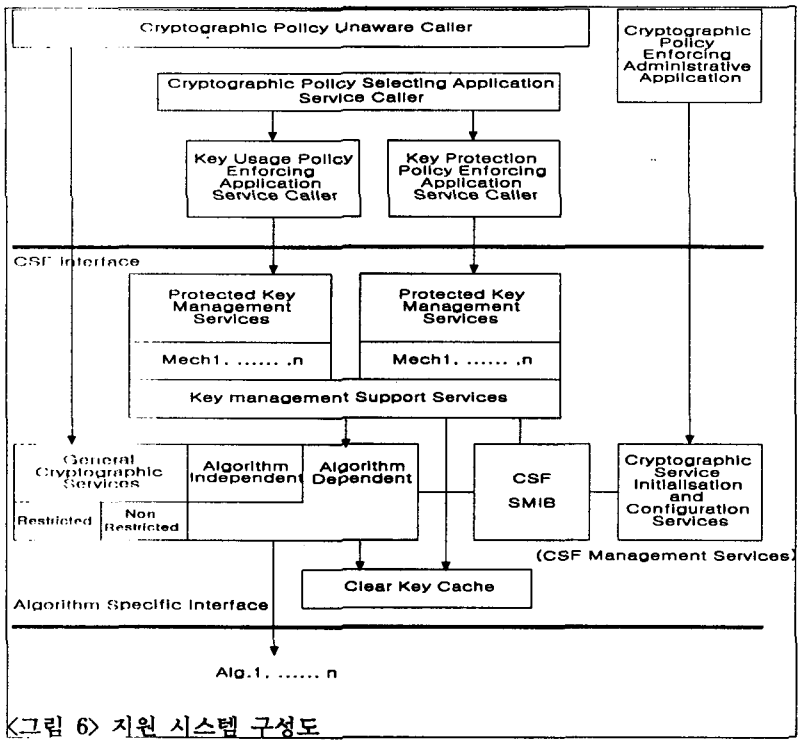
위의 그림에서 보이는 것처럼 CSF는 다양한 암호학을 아는 호출자와 그에 따르는 서비스의 형태를 사이에서 API와 SPI 두개의 프로그래밍 인터페이스를 제공한다.

API는 범용 암호화 서비스와 키 보호 관리 서비스를 위해 인터페이스들을 구성한다.

범용 암호 서비스(General Cryptographic Services)는 데이터 암호화, 복호화, 점검값의 생성, 점검값 검증을 제공하며 CSF의 호출자와 키 관리 지원을 위한 내부 CSF함수들에 의해 구동된다.

보호된 키관리 지원 서비스(Protected Key Management Support Services)는 암호 정책 선택 호출자와 키 사용 정책 집행 호출자에게 키 생성, 저장, 분배 서비스를 제공한다.

SPI는 노출된 키관리 서비스들을 구성하며, 노출된 키관리 지원 서비스(Clear Key Management Support Services)는 키 보호 정책 집행 호출자에게 노출된 키생성, 저장, 분배 서비스를 제공한다.



<그림 6> 지원 시스템 구성도

5. 결론

본 연구에서는 개방형 분산 환경에서 다양한 응용프로그램들에게 효율적인 보안 서비스들을 지원하기 위하여 범용 보안 서비스 구조와 관련된 PKI, GSS-API 및 GCS-API를 중심으로 각각의 요구사항과 기능을 분석하고 설계모델을 분석 하였다. 이러한 구조는 응용프로그래머에게 데이터의 암복호화, 무결성 점검, 해쉬값 생성, 난수생성 등의 보안 서비스 및 다양한 키 관리 서비스를 응용프로그램과 운영 체제의 제약없이 공통적으로 제공할 수 있는 기반 모델의 장점을 갖는다.

PKI는 원활한 활용을 위하여 인증서와 사용자 공개키 이용에 대한 키 생명주기관리, 분산된 인증서관리, PKI 서비스 자체보안, 국제적 타임서비스, 그리고 국제적 표준과 상호 동작성 지원기능등을 수행해야 한다. 또한, 시스템의 오버헤드와 사용자의 편의성을 고려하여 인증서 발행기관을 선정하고, 우리나라에 가장 적합한 인증경로를 반영한 인증서 관리구조를 구축해야 할 것이다.

GSS-API는 응용 프로그래머가 보안 메커니즘의 상세한 내용을 알 필요없이 보안 서비스을 제공받을 수 있게 해 주지만, 시스템 개발자는 범용 인터페이스와 특정 보안 메커니즘에 관한 상세한 지식을 갖고있어야 한다. 또한, 원활한 동작성 지원을 위하여 완전한 표준 API의 수용과 함께 메커니즘 관리, 세션 관리, 신임장 관리, 키 관리, 및 각 응용프로그램 사이의 네트워크 관리등의 기능을 종합적으로 지원할 수 있도록 설계되어야 할 것이다.

GCS-API는 다양한 계층의 사용자을 고려해야하고 관련된 키 관리 지원서비스, CSF 관리서비스 등 전체적인 서비스 체계가 복잡하므로 기본적 API 및 상급 API 서비스로 구분하여 단계적 구현전략이 필요하다. 또한, 함수사용에 대한 권한부여 정책과 암호 기술 사용과 관련된 법적인 제한 등 종합적인 보안정책이 함께 수립되어야 할 것이다.

앞으로 공개키 관리 하부구조 및 다른 보안 관련 메커니즘과의 연계성을 폭넓게 수용하여 보다 범용성과 이식성이 있는 이상적인 서비스 지원 체제가 계속 연구되어야 할 것이다.

참 고 문 헌

[1] 김문현, 이안형, 이정희, "초고속 정보화 추진을 위한 소프트웨어기술 방향과 전략", 정보과학회지, 제13권 제2호, 1995. PP.7-18.

[2] William Stalling, "Network and Internetwork Security", Prentice Hall, 1995.

[3] Sead Muftic, Morris Sloman, "Security Architecture for Distributed Systems", Computer Communications Volume 17 Number 7, July, 1994.

[4] Vijay Ahuja, "Networkt & Internet Security", Acdemic Press, 1996.

[5] Per Kaijser, Tom Parker, Denis Pinkas, "SESAME: The Solution to Security for Open Distributed Systems", Computer Communications Volume 17 Number 7, July, 1994.

[6] W. Caelli, I. Graham, L. O'Connor, "Cryptographic Application Programming Interfaces(APIs)", Computer & Security, 1993.

[7] D. P. Barton, L. J. O'Connor, "Implementing Generic Security Services in a Distributed Environment", May. 1995.

[8] X/Open Company Ltd., "Generic Cryptographic Service API(GCS-API) Base - Draft 8", X/Open Preliminary Specification, April. 1996.

[9] Noel Nazario Ed., "Federal Public Key Infrastructure(PKI) Technical Specification : Part B-Technical Security Policy", TWG96-001, 24 January 1996.

[10] N. A. Nazario, "Security Policies for the Federal Public Key Infrastructure", 19th NISSC, Baltimore, Oct. 22-25, 1996, pp.445-451.

[11] L. kohnfelder, "Towards a Practical Public-key Cryptosystem", Bachelor's Thesis, M.I.T., May, 1978.

[12] D. Denning, "Protection Publickeys and Signature Keys", Computer, Febrary. 1983.

[13] 최용락, 강창구, 김대호, "디렉토리 모델과 정보보호 서비스", 통신정보보호학회지, 제5권 제3호, 1995. pp.49-68

[14] CCITT Recommendation X.509, "The Directory : Authentication Framework", 1993.

[15] W.E Bur, "Federal PKI Concepts of Operations", Federal PKI Technical Working Group(DRAFT), Nov, 1995.

[16] Public Key Infrastructure Study, The MITRE Corporation for the National Institute of Standards and Technology, April 1994.

[17] PKI Working Group, Federal Public Key Infrastructure(PKI) Technical Specification : Pard D - Interoperability Profiles, CygnaCom Solution, Inc. Draft 27 Sept,, 1995.

[18] J. Linn, "GSS API", RFC1508, Sept. 1993.

[19] J. Wray, "GSS API : C-binding", RFC1509, Sept., 1993.