보안 서비스 강화를 위한 다중구조 CAPI 모듈 설계

조상규*, 김광종, 이연식
군산대학교 컴퓨터정보과학과
E-mail: sangkyu@cs.kunsan.ac.kr

Design of Multi-Structured CAPI (Cryptographic Application Programming Interface) Module for Strengthening of Security Service

Sangkyu Joe*, Kwangjong Kim, Yonsik Lee
Dept. of Computer Information Science, Kunsan National University

요 약

최근 활발하게 이루어지고 있는 정보 보안에 관한 연구 및 개발 중 보안 API 는 보안 서비스를 제공하는 인터페이스 규격으로서의 중요성이 증대되고 있다. 그러나 대표적인 기존 보안 구조들인 CryptoKi, CryptoAPI, CSSM API, GSS-API, 및 GCS-API 등의 보안 API 는 응용개발자와 보안 장비 개발자의 편리성 및 독립성 보장 측면에서 다양한 문제점들을 가지고 있는 실정이다. 따라서 본 논문에서는 인터넷 응용환경에서의 신분위장, 통신내용의 도청 및 변조, 의도적인 업무 방해 등 수 많은 위협 요소들로부터의 정보 보호를 위한 사용자 인증, 데이터 기밀성 및 무결성 서비스를 제공하는 다중구조의 CAPI(Cryptographic Application Programming Interface) 보안 서비스 모듈을 설계한다. 설계된 다중구조 CAPI 는 사용자 인증, 접근통제 등 상위 어플리케이션 계층에 보안 시스템 서비스 체계를 적용하여 운용 시스템 환경에 따라서 다양하게 개발 및 적용될 수 있다.

1. 서 론

보안 API 는 보안 서비스를 제공하는 인터페이스 규격으로서 특히, 개방형 분산시스템 운용환경의 진입과 인터넷 이용이 확산됨에 따라 새로운 차원의 보안정책 수립과 관련된 대응 기술을 적기에 확보하여 정보화 사회의 역기능을 방지하기 위한 범용 정보보호 기술의 필요성이 증대되고 있다.

개방형 분산 환경에서 정보에 대한 위협으로는 신분위장, 재전송, 도청, 부인, 불법적인 조작 등 수많은 종류의 공격 형태가 있을 수 있다. 이에 대응한 서비스로서 실체 인증, 부인봉쇄, 접근통제, 및 기밀성, 무결성 등의 다양한 형태의 보안서비스가 지원되어야 한다[1,2].

그러나, 각 네트웍 서비스 어플리케이션별로 필요할 때마다 별도의 보안 구조를 설계하는 것은 많은 중복된 노력을 필요로 하며, 상호 호환성을 저해하는 요인이 되고, 또한 어플리케이션 개발자가 보안에 관한 지식을 갖고 있어야 한다는 부담이 있다. 따라서, 보안 메커니즘에 독립적이며, 여러 환경에서 공통적으로 사용할 수 있는 CAPI (Cryptographic Application Programming Interface)가 필요하게 되었다[3,4].

따라서, 본 논문에서는 국제적으로 널리 연구 검토되고 있는 정보보호 서비스를 위한 CAPI 들의 적용기술 및 특성들을 분석하고, 인터넷응용서비스의 다양한 위협 요소들을 분석하여, 보안 서비스 강화를 위한 사용자 인증, 데이터 기밀성 및 무결성 서비스 등을 제공하는 다중구조의 CAPI 모듈을 설계한다.

본 논문의 구성은 2 장에서는 보안 서비스를 위한 기존 CAPI 기술들을 분석하여 문제점들을 도출하고, 3 장에서는 분석 되어진 CAPI 들을 통합 보완 할 수 있는 보안성 강화 CAPI 모듈을 설계한다. 그리고 4 장에서 결론 및 향후 연구 방향을 제시한다.

2. 보안 서비스를 위한 기존 CAPI 기술 분석

CAPI 란 응용 프로그래머의 보안에 관한 지식에 상관없이 각종 어플리케이션에 공통적인 보안 서비스를 제공하는 API 를 의미한다. 응용 프로그래머는 이러한 CAPI 를 사용함으로써 각각의 보안모듈 개발 부담으로부터 해방될 수 있으며, 네트웍 응용 프로그램 소스 레벨의 호환성을 유지하며 보안 기능을 추가할 수 있다. 그러나 이러한 암호화를 어플리케이션 상에서 서비스할 수 있도록 하는 다양한 CAPI 들이 연구 개발되었지만 아직 해결되지않은 여러 가지 문제점들을 가지고 있다[4,5,6,7,8].

본 논문에서는 기존 CAPI 들을 다음과 같은 기준들을 기준으로 분석한다.

- 모듈화 설계 및 보조 서비스
- 암호학적 지식의
- 어플리케이션
- 알고리즘의 독립성
- 암호 모듈 독립성

RSA 사의 공개키 암호 표준인 CryptoKi 는 개발자가 단일 인터페이스를 통해서 슬롯에 있는 어떠한 장치라도 동일한 코딩 방식을 통해 응용 서비스를 개발할 수 있도록, 최상위에 여러 개의 어플리케이션이 존재하며 단말에는 암호학적 장치들이 제공되는 형태를 갖추고 있다. 그러나, 이 방식은 휴대성, 확장성, 일반성, 자원 공유 및 알고리즘 독립성 등의 장점은 있지만 키 관리와 암호 모듈증명과 같은 모듈 설계 및 보조 서비스는 CryptoAPI 모듈과 비교 했을 때 효율적이지 못한 단점을 가지고 있다.

Intel 사에서 만든 CDSA (Common Data Security Architecture)의 핵심 부분인 CSSM-API(Common Security Service Management-API)는 암호, 인증서 관리, 신뢰정책, 보안관리 데이터 저장 및 키 복구 기능과 같은 보안 서비스를 제공하지만[5,6,9] 이 방식 또한 키 관리와 암호 모듈 증명과 같은 서비스를 제공하지 못한다.

1993 년 IETF(Internet Engineering Task Force)에 의해 개발된 GSS-API 는 Security Context 라는 유용한 개념을 도입하여 신용장(Credential)을 사용하며 4 개의 그룹으로 분류하여 다른 보안 서비스에 우선하여 인증 서비스 방법을 다루고 있으나 [5], 이 방식은 다른 어플리케이션 서비스와 비교 해 볼 때 키 관리나 사용자 인증은 제공되지만 암호모듈 증명, 보증서 관리 및 질의 능력 등이 미흡하다.

X/OPEN 컨소시엄의 SWG(Security Working Group)이 보안 서비스 제공 목적으로 개발된 GCS-API 는 다른 보안 서비스 CAPI 에 비해 상대적으로 많은 암호학적 지식을 요구하고 특정 알고리즘 사용에 의존하지 않는 특성을 가지고 있으며 다양한 비밀키와 공개키 기반의 암호 시스템을 포함하고 있다. 또한 키 교환 알고리즘을 적용하여 키 관리를 지원한다.

마이크로소프트사가 개발한 CryptoAPI[10]는 표준화된 프로그래밍 인터페이스에 대한 암호 서비스 제공자(CSP)에게 통용되는 암호 모듈을 추상화 하는 하위 수준의 인터페이스로서, GDI(Graphic Device Interface) 와 비슷한 구조로 구성 되어 있으며 이러한 구조 덕분에 다양한 알고리즘을 골라 쓸 수 있다. 또한 이 CryptoAPI 는 CSP (Crypto-graphic Service Provider)라는 확장 모듈 형식을 제공한다.이 형식에 맞춰 제공되는 암호화 알고리즘은 그대로 CryptoAPI 에서 쓸 수 있다.

이와 같은 기존의 CAPI 들에 대하여 앞서 제시한 분석 기준을 기반으로 특성과 설계원리를 분석 비교한 결과는 <표 1>과 같이 요약할 수 있다.

<표 1> 범용 API 서비스 비교 분석

비 교 기 준		CryptoKi	CSSM	GSS-API	GCS-API	CryptoAPI
모듈설계 및 보조 서비스	키 관리	X	X	O	O	X
	암호모듈 증명	X	X	X	O	O
	사용자 인증	O	O	O	O	O
	보증서 관리	X	O	X	X	X
	질의 능력	O	X	X	O	O
	설치/해제 능력	O	O	O	O	O
암호학적 지식 정도		O	O	O	O	O
어플리케이션 독립성		O	O	O	O	O
암호 모듈 독립성		O	O	O	O	O
알고리즘 독립성		O	O	O	O	O

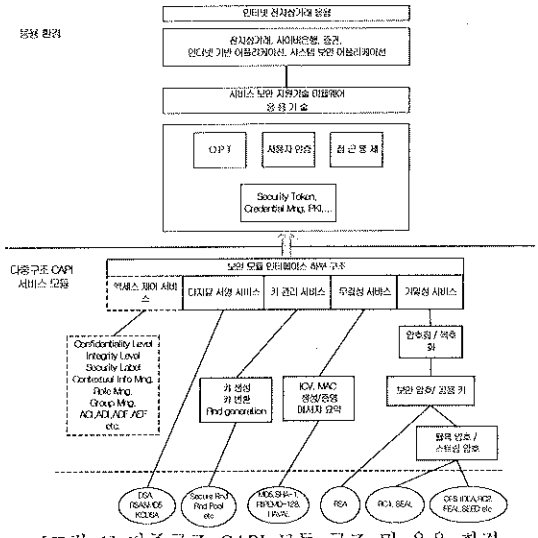
분석 결과 기존의 CAPI 들의 경우 암호학적 지식 정도, 어플리케이션 독립성, 암호 모듈 독립성 및 알고리즘 독립성면에서는 모두 만족할 만한 결과를 보이지만, 모듈설계 및 보조 서비스면에서 살펴보면 사용자 인증과 설치/해제 능력을 제외한 나머지 기준들에는 아직 미흡한 실정임을 알 수 있다. GSS-API 와 GCS-API 를 제외한 모든 CAPI 에서는 키 관리가 이루지지 않고 있으며, 암호 모듈 증명은 GCS-API 와 CryptoAPI 만이 서비스가 이루어 지고 있고, 보증서 관리 측면에서는 CSSM 만이 보증서 관리가 이루지고 있음을 알 수 있다. 또한 질의 능력면에서는 CryptoKi, GCS-API 및 CryptoAPI 만이 질의를 통한 서비스를 제공하고 있다. CAPI 를 이용하여 어플리케이션을 구현할 경우 모듈들을 특정 목적의 단위별로 분리하여 설계하기 위해서는 암호모듈 증명, 사용자 인증, 보증서 관리, 질의 능력 및 세션의 설치 해제 능력을 필요로 한다. 하지만 분석된 결과를 살펴보면 기존의 CAPI 를 이용하여 어플리케이션 보안 모듈을 구현한 경우 현재의 키 관리와 암호모듈 및 보증서 관리와 같은 서비스가 특정 CAPI 만이 제공하고 있음을 알 수 있다. 따라서 보안 어플리케이션 구현할 때 컴포넌트 구조의 모듈 단위 설계를 지원하지 않음으로 메커니즘에 독립적인 CAPI 를 구현하기에는 어려움이 있다. 즉, 이는 플랫폼이 바뀔 때마다 다시 설계 해야 하는 문제점이 있음을 나타낸다. 이러한 분석결과를 토대로 본 논문에서는 기존 CAPI 들이 가지고 있는 다양한 문제점을 보완 및 해결할 수 있는 다중구조 CAPI 모듈을 설계를 한다.

3. 다중구조 CAPI 모듈 설계

3.1 다중구조 CAPI 모듈 구조

기존의 CAPI 를 통합 보완한 다중구조 CAPI 설계 모듈의 핵심적 구조는 [그림 1]과 같다. 다중구조 CAPI 는 환경에서의 적용을 중심으로 구성 되어진 응용 환경과 데이터의 신제적 처리 모듈을 중심으로 구성 되어진 다중구조 CAPI 시비스 모듈로 나눌 수 있다. 응용 환경은 하위의 데이터 시비스 모듈들로부터 전달되어진 시큐리티 토큰을 사용자 인증 및 접근 통제를 이용하여 서비스 보안 지원 미들 웨어를 통해 인터넷 기반의 다양한 보안 어플리케이션 구현을 지원한다. 또한 하위의 다중구조 CAPI 서비스 모듈 구조는 데이터의 보안성 강화를 위한 액세스 제어, 디지털 서명, 키 관리, 무결성 증명 및 기밀성 서비스를 제공하고 있으며 각각의 서비스는 하위 모듈 토대로 구성되어 있다. 다중구조 CAPI 는 인터넷과 인트라넷 어플리케이션 서비스 공간에서 통신과 데이터 보안 문제들을 다루는 계층화된 보안 서비스들의 집합으로서, 가장 낮은 계층에서 암호학적 알고리즘, 난수, 유일한 식별 정보와 같은 기본 요소를 가지고 있으며, 통신 네트워크를 통해서 교환되는 디지털화 된 메시지들에 인증 서비스를 제공하기 위한 디지털 사명 시비스와 메시지 전달시의 보안성을 제공하기 위한 키 관리 서비스 및 전송된 데이터의 무결성을 보장하기 위한 무결성 시비스를 구조로 갖는다.

그리고 다중구조 CAPI 는 정보보호 시비스에 대한 인터페이스를 암호 서비스 수행 부분과 독립적으로 수행토록 하여 공개키 및 비밀키 암호 시스템 사이의 구분을 없애고, 시스템과 독립적으로 어플리케이션을 구현 할 수 있도록 한다. 또한 CAPI 를 하부 통신 프로토콜과 독립적으로 수행토록 하여 다중구조 CAPI 를 이용하여 어플리케이션을 구현할 때 소스레벨의 이식성을 증가 시키고 다양한 플랫폼에서 단일 구조의 구현을 허용하게 한다.



[그림 1] 다중구조 CAPI 모듈 구조 및 응용 환경

CAPI 의 동작 형태는 메시지가 무결성 또는 기밀성 서비스를 위하여 CAPI 에 전달되면 그 결과로 토큰이 CAPI 에서 생성되어 수신자 측으로 전달 된다. 이 때 메시지는 키 관리 서비스에 의한 키 생성과 변환 과정을 거치며 이러한 메시지 전달 동작은 CAPI 가 어플리케이션에 의해 사용되는 일반 프로토콜 인터페이스를 사용하지 않고 분리된 인터페이스를 이용하여 각각의 프로세스로 구분되어진 서비스 단위의 모듈별로 수행 된다. 각 모듈별 구조의 내용은 다음과 같다.

- **액세스 제어 서비스** : 비밀성 레벨과 무결성 레벨 그리고 보안 레벨을 중심으로 문맥 정보관리와 익할 관리 및 그룹을 관리하며 ACI, ADI, ADF, AEF 모듈 등을 이용한다.
- **디지털 서명 서비스** : 데이터의 무결성을 보장하기 위한 모듈로서 DSA, RSA&MD5, KCDSA 와 같은 하부 모듈을 이용하여 인증을 통한 무결성 보장 서비스를 수행한다.
- **키 관리 서비스** : 키의 생성과 키의 변환 과정과 같은 키 관리 어플리케이션 서비스를 제공 한다.
- **무결성 서비스** : ICV, MAC 생성/증명,메시지 요약을 통한 데이터 전송 시 악의적인 변형과 같은 피해를 줄이기 위한 모듈로서 MD5, SHA-1, RIPEMD-128, HAVAL 와 같은 하부 보안 모듈들을 이용한다.
- **기밀성 서비스** : 다중구조 CAPI 설계 구성 중 가장 중요한 암호화에 관한 부분을 담당하는 요소로서, 데이터의 암호화와 복호화를 통하여 데이터의 기밀성을 보장하기위한 모듈이다. 이 서비스 모듈은 보안 암호를 이용한 공용키 기반의 RSA 를 사용하는 하위 보안 모듈 형태와 블록 암호와 스트림 암호 형태로 나누어진다. 스트림 암호 방식으로는 RC4 와 SEAL 이 하부 모듈로 구성되며 블록 암호 방식으로는 DES, IDEA, RC2, FEAL, SEED 등 하부 모듈로 구성 되어 서비스 된다.

3.2 설계된 다중구조 CAPI 모듈의 특성

일반적인 CAPI 요구사항으로는 알고리즘 독립성, 어플리케이션 독립성, 암호 모듈의 독립성, 암호학적 지식의 정도, 모듈화 설계, 부가 서비스, 안전한 프로그래밍 및 보안 정책 등이 있지만 현재의 범용 CAPI 에서는 불필요한 기능의 구동, 오버헤드, 메모리 공간의 낭비등과 같은 문제점을 가지고 있기 때문에 이를 최소화 하는 것이 필요하다. 이에 본 논문에서 설계 제안된 다중구조 CAPI 는 내부 기능의 모듈별 설계 및 종속성 제거를 통하여 독립적인 모듈의 구현과 동작을 가능하게 하여 성능을 개선하며, OTP, 사용자 인증, 접근통제 등 상위 계층의 다단계 시스템 서비스 체계를 적용하여 설계 함으로써 운용 시스템 환경에 따라서 다양하게 적용될 수 있는 특징을 가진다. 또한 기존의 CAPI 구조는 각 네트웍 서비스 어플리케이션별로 필요할 때마다 별도의 보안 구조를 설계하여야만 하는 많은 중복된 노력이 요구되어 상호 호환성을 저해하는 요인이 되고, 어플리케이션 개발자가 보안에 관한 지식을 갖고 있어야 한다는 부담이 있지만 제안한 보안 CAPI 모듈은 보안 메커니즘에 독립적이며, 여러 환경에서 공통적으로 사용할 수 있는 CAPI 형태의 특성을 가지고 있다.

4. 결론

본 논문에서는 인터넷 이용에 따른 다양한 위협요소에 대처할 수 있는 범용적인 CAPI 를 조사하여 인터넷 응용서비스의 다양한 위협요소를 분석, 기존의 CAPI 의 문제점을 도출하고, 보다 안전한 보안성 확보를 위한 사용자 인증 및 데이터 기밀성과 무결성 서비스 제공을 위한 다중구조 CAPI 를 갖는 보안 모듈을 설계하였다. 이는 인터넷과 같은 네트워크 환경과 시스템 자체에 대한 보안기능을 제공하는 어플리케이션 개발에 사용될 수 있으며, 또한 네트워크를 통한 사용자 인증이나, 시스템 내의 자원에 대한 기밀성, 무결성, 접근통제 서비스, 보안 프로토콜 개발과 같은 보안 제품 개발에 필요한 필수적인 기능을 하부 메커니즘으로 제공함으로써 각종 전자상거래 응용 및 인터넷 기반 보안 서비스 등을 제공 할 수 있으며 새로운 어플리케이션 개발에 사용 될 수 있다.

그러나 설계된 CAPI 가 메커니즘에 독립적이고 키 관리와 암호 모듈 증명을 지원하며 상호 호환성면에서 다른 기존의 CAPI 보다 좋은 이점을 가지고 있지만 이러한 풍부 구조적 특성 때문에 발생하는 실행 시간 오버헤드 및 부수적 불일치성에 대한 문제점들을 해결하기 위한 연구가 향후 지속되어야 한다.

참고문헌

[1] X/Open Company Ltd., "Generic Cryptographic Service API (GCS-API) Base-Draft 8," X/Open Preliminary Specification, April, 1996
[2] DASS Distributed Authentication Security Service, http://www.ietf.org/rfc/rfc1507.txt
[3] Generic Security Service Application Program Interface, http://www.ietf.org/rfc/rfc1508.txt
[4] Generic Security Service API: C-bindings, http://www.ietf.org/rfc/rfc1509.txt
[5] J. Linn, "Generic Security Service Application Program Interface, Version 2," RFC2078, Jan., 1997
[6] http://spam.xopen.org/pubs/catalog/p442.htm
[7] William Stalling, "Network and Internet-network Security," Prentice Hall, 1995
[8] Sead Muftic, Morris Solman, "Security architecture for distributed systems," Computer Communication, Vol. 17, No.3, April 7, 1996
[9] Common Data Security Architecture Spec., Release 1.2, Intel, Feb., 1998
[10] Microsoft, "CryptoAPI2.0," http://preminum.microsoft.com/msdn/library