EDI 정보보호 시스템 설계 및 구현

윤이중, 이정현, 이대기, 김대호
한국전자통신연구원

A Design and Implementation of EDI Security System

Jeong Hyun Yi, E Joong Yoon, Dai Ki Lee, Dae Ho Kim
Electronics and Telecommunications Research Institute

요 약

본 논문에서는 전자 문서 교환 시스템인 EDI에서의 정보보호 서비스를 제공하는 EDI 정보보호 시스템의 개발 기술에 대하여 언급한다. 이 시스템은 주로 ITU-T X.400 MHS 통신 프로토콜에 기반을 둔 X.435 EDI 정보보호 서비스를 제공함에 그 목적을 두고 있다. 본 논문에서는 정보보호 서비스 처리시 발생하는 중복되는 작업을 최소화하고 사용 알고리즘을 언제든지 교체 또는 변경할 수 있는 시스템 구조와 기존 EDI 시스템의 최소변경만으로 정보보호 시스템 구축이 가능토록 하는 이식성을 극대화하는 정보보호 시스템 모델과 관리 방안을 제시하였다.

1. 서론

정보 사회의 도래와 함께 기업체 혹은 국가 기관에서는 전자문서를 서로 교환할 수 있는 정보통신 시스템을 구축하게 되었고, 이는 EDI(Electronic Data Interchage)라는 전자문서 교환 시스템의 등장을 이루게 되었다. EDI 시스템에서 주로 다루어지는 문서는 주문서, 계약서, 협정서 등 계약 당사자간에 이해관계가 있는 중요한 정보 뿐만 아니라 기업체간의 신용 및 거래에 관

본 논문은 한국전기통신공사의 출연금에 의하여 수행한 연구결과입니다.

련된 내용이기 때문에 주고받는 메세지 내용을 안전하게 관리하는 것은 매우 중요한 문제라 할 수 있다.

EDI 문서는 주로 상업적인 거래문서이므로 보안상 생긴 문제는 많은 분쟁의 소지를 갖게 된다. EDI 시스템에서의 위협요소로서는 불법적인 액세스를 위한 실체 위장, 메세지의 지연 혹은 재전송을 위한 메세지 순번 변조, 전달정보를 위조하는 정보변조, 메세지 전송, 제출, 배달에 대한 부인, 메세지 전송감시에 의한 정보 노출 및 기타 정보보호 레이블 관련 위협 요소가 있을 수 있다.[1][7][8]

EDI로 업무를 처리하기 위해서는 이러한 위협들로 부터 사전에 예방할 수 있는 보안 대책과 사후에 분쟁 발생시 부인하지 못하도록 하는 보안 관리 대책 또한 필요하다.

EDI 표준에는 크게 문서 표준[10]과 통신표준[12][13][14][15][16]으로 구분할 수 있는데, 본 논문에서 기술하는 EDI 정보보호 시스템이란 ITU-T에서 표준으로 제정한 X.400 MHS(Message Handling System) 통신 프로토콜에 기반을 둔 X.435 EDI 시스템에서의 각종 위협들을 방지하는 시스템으로 정의한다. EDI 정보보호 시스템의 구성은 요구된 서비스의 처리를 위한 처리 순서의 결정, 요구된 서비스의 기능별 처리함수의 결정, 서비스 처리를 위한 SES(Secure EDI Subsystem) 시스템과, 정보보호 서비스 처리에 요구되는 공개키(public key) 및 개인키(private key)의 생성, 분배 및 관리, 보증서(certificate)의 발급, 등록, 조회 등을 담당하는 KMS(Key Management Subsystem) 시스템, 향후 발생될 수 있는 분쟁에 대한 증빙자료를 제시할 수 있는 보안 감사 추적 시스템인 SAS(Security Audit Subsystem) 시스템으로 이루어져 있다.

2. EDI 정보보호 서비스와 메카니즘

2.1 정보보호 서비스

가. 발신처 인증 서비스(Origin Authentication)

발신처 인증 서비스는 수신자에게 메세지가 정당한 송신자로부터 왔음을 인증해 주는 서비스로서, 이는 디지탈 서명 메카니즘으로 구현가능하다. 이 서비스의 종류에는 인증 대상이 무엇이냐에 따라 크게 3가지로 구분된다. 송수신 당사자간의 메세지의 정당성 여부를 확인해주는 메세지 발신처 인증(message origin authentication), 메세지가 잘 전달될수 있는지의 여부를 확인하기 위한 프로브(probe)가 정당한 송신자로부터 왔는지를 확인하는 프로브 발신처 인증(probe origin authentication), 수신자가 송신자에게 메세지 수신여부에 대한 보고서를 송신자에게 송신하게 되는데, 이때 이 보고서의 발신처가 정당한지를 인증해주는 보고서 인증서비스(report origin authentication)가 있다. 이외에도, 송신자가 전송구간상의 MTA들에게 제출한 메세지가 잘 수신되었음을 송신측 MTA들이 송신자에게 증명해 주는 제출 증명 서비스(proof of submission)와 메세지 송신자에게 메세지가 변조없이 정당한 수신자에게 배달 되었음을 증명해 주는 배달 증명 서비스(proof of delivery)가 있다.[1][13]

나. 보안 액세스 관리 서비스(Secure Access Management)

이 서비스에는 상대 실체 인증 서비스(peer entity authentication)와 보안 문맥 서비스(security context)가 있다. 상대 실체 인증 서비스는 기능별 컴포넌트들(즉, UA, MS, MTA) 간에 상대방 컴포넌트가 자신과 통신 가능한 컴포넌트인지를 서비스 제공 전단계의 바인딩 시간(binding time)에 사전 확인하여 주는 서비스이다. 보안 문맥 서비스는 기능별 컴포넌트 상호간에 주고받는 메세지에 허용되는 보안 레이블들을 설정하도록 해주는 서비스로서 보안 정책(액세스 제어 등)을 지원해 준다.[1][13]

다. 데이타 기밀성 서비스(Data Confidentiality)

기밀성 서비스란 송수신자간의 메세지의 내용을 암호화하여 기밀성을 보호해 주는 서비스로서 통신로 상의 연결 정보을 보호해 주는 연결 기밀성 서비스(connetction confidentiality), 송수신자 사이의 메세지 내용의 기밀성을 보호해 주는 내용 기밀성 서비스(content confidentiality), 메세지 흐름 관찰에 의한 정보 누출 방지를 위한 메세지 흐름 기밀성 서비스(message flow confidentiality)가 있다.[1][13]

라. 데이타 무결성 서비스(Data Integrity)

무결성 서비스는 수신한 데이타가 전송도중에 아무런 변조가 없었다는 것을 인증해 주는 서비스이다. 이에는 통신 컴포넌트들간의 전송메세지에 대한 데이타 무결성을 보호해 주는 연결 무결성 서비스(connection integrity), 송수신자간의 전송 메세지 내용에 대한 무결성을 유지해 주는 내용 무결성 서비스(content integrity), 송신자와 수신자 사이에서 메세지 순서의 재구성, 재전송, 삭제 등을 방지하기 위해 메세지 순번에 대한 무결성을 제공해 주는 메세지 순번 무결성 서비스(message sequence integrity)가 있다.[1][13]

마. 부인봉쇄 서비스(Non-repudiation)

이 서비스는 메세지의 송신자 혹은 수신자가 메세지의 송수신 사실을 부인하는 행위를 방지해 주는 서비스이다. 메세지의 송신자가 송신한 사실에 대해 부인하지 못하도록 하는 송신 부인봉쇄 서비스(non-repudiation of origin), 송신 MTA가 송신 UA로 부터 메세지를 제출받은 사실에 대해 부인하지 못하도록 하는 제출 부인봉쇄 서비스(non-repudiation of submission), 수신 MTA가 수신 UA 혹은 MS에게 메세지를 배달한 사실에 대해 부인하지 못하도록 하는 배달 부인봉쇄 서비스(non-repudiation of delivery)가 있다.[1][13]

바. 메세지 보안 레이블 서비스(Message Security Labelling)

이 서비스는 메세지에 보안 레이블을 첨부하여 컴포넌트들간 뿐만아니라 각 사용자들에게 보안 등급을 설정하여 액세스제어가 가능하도록 해주는 서비스이다.[1][13]

사. 보안 관리 서비스(Security Management)

UA가 자신의 보안레이블 등을 포함하는 증명서(credential)를 MTA에게 등록하는 등록 서비스(register), 등록된 증명서를 변경하는 증명서 변경 서비스(change credential), MS가 존재하는

시스템일 경우 등록과 증명서 변경 서비스를 별도로 처리해 주는 MS-등록 서비스(MS-register)가 이 서비스에 해당된다.[1][13]

아. EDIM 책임 인증 서비스(EDIM Responsibility Authentication)

수신자가 EDI 메세지를 수신한 후에는 이 메세지에 대한 모든 처리 권한을 갖게 되는데, 이를 EDIM 책임(responsibility)이라 부른다. 이에는 메세지 송신자에게 EDI 메세지가 수신되었고, EDI 메세지가 수락/ 거부/ 회송되었음을 알리는 EDI 통지(notification)의 발신처를 증명해 주는 EDI 통지 증명 서비스(proof of EDI notification), 메세지가 EDI-UA 에 의해 EDI-MS 로 부터 검색되었음을 MS 운영자에게 증명해 주는 검색 증명 서비스(proof of retrieval), 메세지가 다른 영역내의 MTA 에게로 전달되었음을 MTA에게 증명해 주는 전달 증명 서비스(proof of transfer), EDI 메세지 수신자에게 수신한 메세지 내용은 송신자가 보낸 메세지 내용과 같음을 증명해 주는 내용 증명 서비스(proof of content)가 있다.[1][15][16]

자. EDIM 책임 부인봉쇄 (Non-repudiation of EDIM Responsibility)

이 서비스에는 메세지 송신자에게 EDI 메세지가 수신되었고 EDI 메세지가 수락/ 거부/ 회송되었음을 알리는 EDI 통지의 발신처를 메세지 수신자가 부인할 수 없도록 하는 EDI 통지 부인봉쇄 서비스(non-repudiation of EDI notification), MS 운영자에게 특정 메세지가 EDI-UA 에 의해 EDI-MS 로 부터 검색되었음을 EDI-UA가 부인할 수 없도록 하는 검색 부인봉쇄 서비스(non-repudiation of retrieval), 메세지가 다른 영역내의 MTA에게로 전달되었음을 MTA가 부인할 수 없도록 하는 전달 부인봉쇄 서비스(non-repudiation of transfer), EDI 메세지 수신자에게 수신한 메세지 내용은 송신자가 보낸 메세지 내용과 같음을 송신자가 부인할 수 없도록 하는 내용 부인봉쇄 서비스(non-repudiation of content)가 있다.[1][15][16]

2.2 EDI 정보보호 메카니즘

네트워크를 통한 다양한 공격들로부터 메세지를 보호하기 위한 정보보호 서비스들의 제공은 다양한 정보보호 메카니즘에 그 기반을 두고 있다. EDI 시스템에서는 해쉬함수, 디지탈 서명, 메세지 인증, 대칭키 암호 기법을 사용하여 정보보호 서비스를 제공한다.

가. 해쉬함수

해쉬함수는 메세지의 압축된 새로운 표현인 메세지 다이제스트(message digest)를 생성하는 것으로 일방향성과 충돌 회피성을 만족하는 임의의 크기의 입력에 대한 일정한 출력을 생성하는 함수로 정의된다. 여기서 해쉬함수의 출력 비트 스트링을 해쉬 코드(hash code)라고 말한다. 해쉬함수는 주로 디지탈 서명 메카니즘에서 수행 시간을 단축시키기 위해 사용될 뿐만 아니라 신분확인 메카니즘, 무결성 메카니즘, 컴퓨터 바이러스 예방 등의 다양한 용도로 사용될 수 있다.[6][7]

나. 디지탈 서명

디지탈 서명은 공개키 암호가 제공할 수 있는 하나의 특징으로, 수신자가 받는 메세지의 변조나 위조를 방지하며, 메세지의 송신자가 추후 부인할 수 없도록 하는 것으로 메세지 인증과 사용자 인증을 동시에 수행한다. 비밀키를 이용한 메세지 인증에서는 송신자와 수신자 사이의 분쟁 발생시 문제 해결이 곤란하지만, 디지탈 서명에서는 제3자의 중재를 통하여 분쟁을 해결할 수가 있어 EDI 와 같은 전자 상거래에 널리 활용될 수 있다.[6][11][28][29]

공개키 암호를 이용한 디지탈 서명 방식을 간략히 소개하면 다음과 같다.

먼저, 서명자는 자신의 개인키 Kpri와 메세지 m을 이용하여 서명문 S=Kpri(m)을 발생하여 메세지 m과 함께 수신자에게 보낸다. 수신자는 수신한 서명문 S를 서명자의 공개키 Kpub를 이용하여 복호화한 결과와 수신한 메세지를 비교하면 된다. 어느 누구도 서명자의 개인키는 알지 못하므로 m에 대한 서명문을 만들 수 없게 되며, 서명자가 추후 부인할 수 없는 사유가 되기도 한다.

다. 메세지 인증

메세지 인증이란 메세지가 송신 또는 전달 도중에 어떠한 변경이나 위조없이 수신자에게 전달되었다는 것을 확인하는 절차를 말하는데, EDI 정보보호 시스템에서는 해쉬코드를 이용하여 확인하는 해쉬코드 조회법[7]을 도입하였다. 해쉬코드 조회법은 패리티 검사 부호와 원리적으로 유사하지만 패리티 비트에 해당하는 인증자를 해쉬함수를 이용하여 발생하는 점이 다르다. 먼저, 송신자가 메세지 m에 일방향 해쉬함수 h를 이용하여 해쉬코드 h(m)을 발생하여 메세지 m과 함께 수신자에게 보낸다. 수신자는 수신된 메세지 m과 해쉬함수 h를 이용하여 새로운 해쉬코드 h(m)을 만들어 송신자가 생성하여 보내온 해쉬코드와 비교하여 본다. 만일 일치하면 송신자의 메세지는 변형없이 정확한 것임을 인증받게 된다.

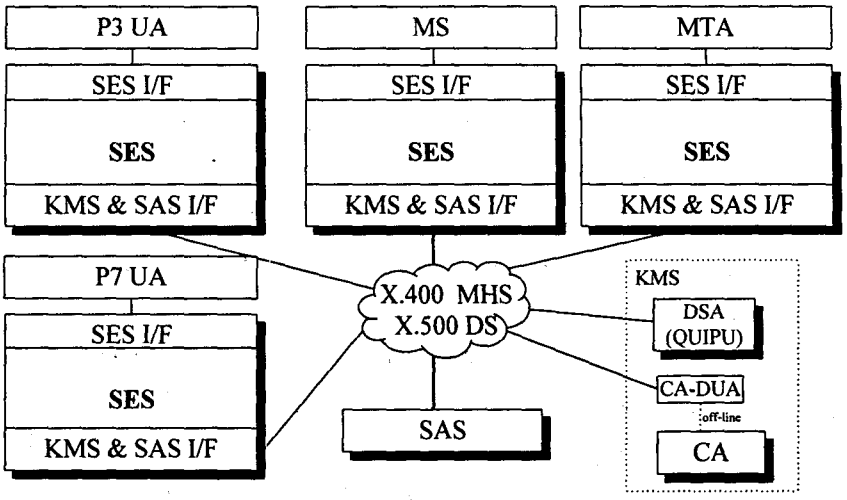
라. 대칭키 암호 시스템

대칭키 암호 시스템의 암호화 과정[8][28][29]은 알고리즘과 키로 구성되는데, 키는 알고리즘을 제어하는 평문과 무관하게 독립된 값이며, 알고리즘은 당시 사용된 특정 키에 따라 상이한 결과를 생성해 내게 된다. 키를 바꾸게 되면 그에 따라 알고리즘의 결과도 변하게 된다. 일단 생성된 암호문은 전송되고, 수신된 암호문은 복호알고리즘과 암호문 생성시에 사용되었던 키와 동일한 키를 사용하여 평문으로 재변환된다. 이때, 암복호하는데 사용한 동일한 키를 대칭키(symmetric key) 혹은 비밀키(secret key) 라 부른다.

3. EDI 정보보호 시스템의 구조

3.1 시스템 구성

EDI 정보보호 시스템의 구성은 [그림 1]과 같이 SES, KMS, SAS의 세개의 서브시스템으로 이루어져 있다.



[그림 1] EDI 정보보호 시스템의 구성

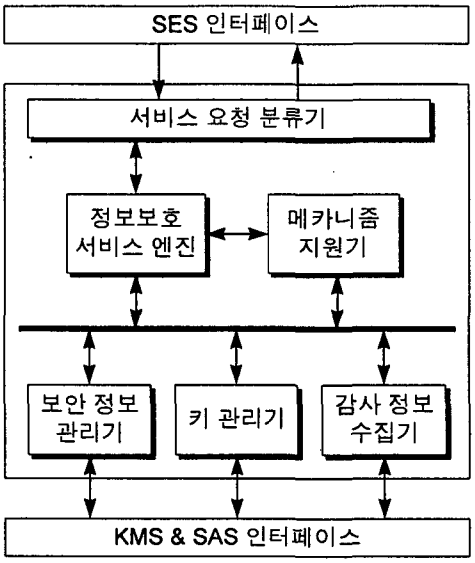
SES는 EDI 시스템의 각 컴포넌트에서 제공해야 하는 정보보호 서비스를 처리하는 핵심 기능을 담당한다. SAS는 EDI 시스템에서 발생하는 정보보호 관련 행위의 기록, 관리, 출력을 담당하고, KMS는 SES에서 발생하는 키관련 정보의 생성, 분배, 관리를 총괄한다.[2][3][4][5]

SES는 EDI 시스템의 컴포넌트인 UA, MS, MTA에 각각 위치하도록 했다. SES의 기본적인 구성 및 구조는 UA, MS, MTA에서 모두 동일하지만, EDI 시스템 각각의 컴포넌트가 수행해야 하는 정보보호 서비스의 종류 및 기능이 다르기 때문에 UA, MS, MTA 별로 그 내용이 다르게 설계 구현되었다.

3.2 SES 시스템

가. SES 의 구조 및 기능

SES는 UA, MS, MTA들이 제공해야 하는 정보보호 서비스를 처리해 주는 주기능을 담당한다. 이를 위해서 SES는 [그림 2]와 같이, EDI 시스템 각 컴포넌트들의 서비스 요청 및 결과의 통보를 위한 SES 인터페이스, 인테페이스를 통해 요구된 정보보호 서비스의 처리를 조정하는 서비스 요청 분류기, 여러 가지의 정보보호 서비스에서 공통으로 사용하는 메카니즘들을 통합한 메카니즘 지원기, 실질적인 정보보호 서비스 처리를 담당하는 정보보호 서비스 엔진, 정보보호 서비스의 처리시 필요한 정보의 관리 및 저장을 위한 보안 정보 관리기, 키의 관리 및 검색을 위한 키 관리기, 감사추적 정보의 처리를 위한 감사 정보 수집기로 구성된다.



[그림 2] SES 구조

1) SES 인터페이스

SES 인터페이스는 EDI 시스템의 각 컴포넌트와 SES와의 통신을 위한 시스템 호출 형태의 함수로 제공한다. 기본 골격은 메세지 송수신에 관련된 통신 포트별로 하나의 함수를 제공하고, 이 함수의 입력은 기본적으로 각 포트에서 처리하는 PDU(Protocol Data Unit) 단위로 정의한다. 이와 같은 인터페이스 방식의 장점은 입출력이 간단하다는 것과, SES와 EDI 각 컴포넌트 사이의 독립성을 유지함으로써 SES 또는 EDI 시스템 어느 한쪽의 변경이 서로에게 미칠수 있는 영향을 최소화할 수 있다는 것이다.

2) 서비스 요청 분류기

서비스 요청 분류기는 첫째, 요청된 서비스들간의 관계를 조사하여 서비스 처리 순서의 결정 또는 서비스들간의 충돌성 등을 조사 결정하여 정보보호 서비스 엔진으로 서비스를 요청하거나, 서비스 처리에 문제가 있다고 판단된 경우는 SES 인터페이스를 통해서 서비스의 요청을 거절한다. 둘째, 요청된 서비스의 종류에 따라서 서비스를 처리하기 위한 초기화 및 선행 작업을 수행한다.

3) 정보보호 서비스 엔진

정보보호 서비스 엔진은 SES의 핵심 부분이라고 할 수 있다. 이 부분은 어떤 컴퍼넌트와 결합하느냐에 따라서 달라질 수 있다. 즉 이 부분은 각 컴포넌트들에 관련된 서비스들을 처리하는 부분이다. 이의 구조에 대해서는 다음 절에서 구체적으로 살펴보기로 한다.

4) 메카니즘 지원기

메카니즘 지원기는 정보보호 서비스를 제공하는데 사용되는 함수들 중에서 공통적으로 사용하는 메카니즘들을 모아놓은 공통 사용 함수 처리부이다. 이렇게 함으로써 해쉬함수, 서명 알고리즘, 암호 알고리즘 등의 메카니즘들을 언제든지 변경, 교체할 수 있는 장점이 있다.

5) 보안 정보 관리기

보안 정보 관리기는 정보보호 서비스의 처리과정에서 처리 결과 또는 서비스 제공을 위한 부가 기능을 위해 필요하다. 부가 기능에는 자신과 연결되는 컴포넌트들의 보안등급 정보, 보안 문맥 등을 등록, 관리하는 기능 등이 포함된다.

6) 키 관리기

키 관리기의 기능은 크게 두가지로 나누어진다. 첫째, 정보보호 서비스를 제공하는데 필요한 송수신자의 개인키 또는 공개키를 제공하고 관리하는 기능을 담당한다. 여기에서 특히 공개키의 획득은 DSA(Directory System Agent)를 이용해야 하는데 이를 위한 인터페이스를 제공한다. 둘째, EDI 정보보호 시스템에서는 인증에 사용되는 공개키 및 개인키의 생성을 각 UA에서 하고, 여기서 생성한 공개키를 CA(Certification Authority)에게 보내서 인증하고, DSA에 등록하게 하는 키분배의 자동화 개념을 도입해야 하는데, 이를 위해 UA와 CA사이의 공개키 전송을 위한 인터페이스를 제공한다.

7) 감사정보 수집기

감사정보 수집기는 SAS로 전송할 감사 데이타의 수집, 전송 기능을 담당한다.

나. 정보보호 서비스 엔진의 구조

SES에서는 앞서 기술한 디지탈 서명 메카니즘, 대칭키 암호 알고리즘 등을 기반으로 하여 기밀성 서비스, 무결성 서비스, 신분확인 서비스 등 다양한 EDI 정보보호 서비스를 제공하게 된다. 이들 다양한 서비스들을 송수신자의 관점에서 살펴보면, 우선 송신측에서 수신자에게 메세지 변경 유무를 확인할 수 있도록 해주는 증거를 생성하는 과정과, 수신측에서의 송신자로부터 받은 증거를 검증하는 과정으로 나눌 수 있다.

1) 증거 생성과정

송신측에서 증거를 생성하는 경우는 디지탈 서명 메카니즘을 통해서 인증 서비스를 제공하고자 하는 경우인데, 대부분의 서비스가 이에 해당한다. 디지탈 서명 메카니즘을 적용하기 위해서는 메세지의 크기를 줄이기 위한 해쉬함수와 공개키 알고리즘이 필요하며, 만약 기밀성 서비스가 요구되었을 경우에는 메세지 내용 자체를 암호화하기 위한 대칭키 알고리즘이 부가적으로 필요하다. 송신측에서 서명 메카니즘을 통한 증거 생성 과정을 기밀성 서비스가 요구되지 않았을 경우를 먼저 설명하면 다음과 같고, [그림 3(a)]는 이 과정을 나타낸 것이다.

제 1단계 : 서명값 생성

- 송신하고자하는 메세지를 해쉬함수를 이용하여 해쉬코드를 생성해낸다.
- 해쉬코드를 송신자의 개인키를 가지고 서명하여 메세지에 대한 서명값을 구한다.

제2단계 : 메세지토큰 생성

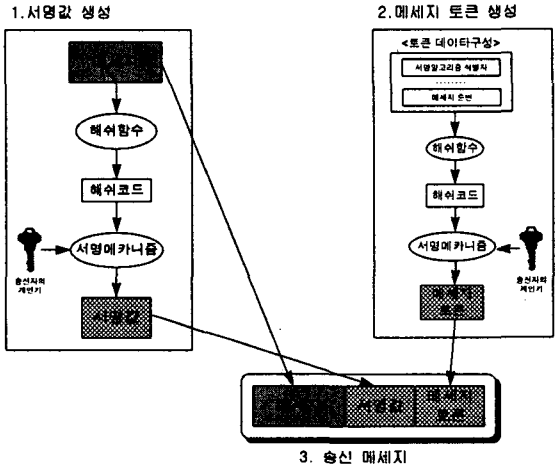
- 서명 알고리즘 식별자, 메세지 순번 등으로 이루어진 메세지 토큰 내용에 포함될 데이타를 구성한다.

● 해쉬함수를 사용하여 메세지 토큰 데이타에 대한 해쉬코드를 구한다.

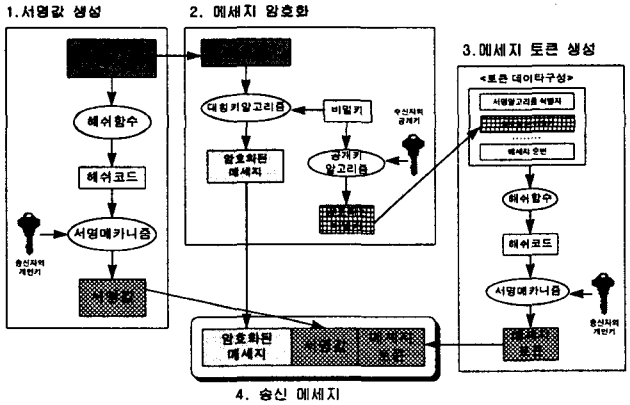
● 해쉬코드를 송신자의 개인키로 서명하여 메세지 토큰을 생성해낸다.

제3단계 : 송신 메세지 구성

● 수신자에게 메세지, 서명값, 메세지 토큰으로 이루어진 PDU를 송신한다.



(a) 기밀성 서비스가 요구되지 않았을 경우



(b) 기밀성 서비스가 요구되었을 경우

[그림 3] 디지탈 서명을 위한 증거 생성 과정

메세지 기밀성 서비스가 요구되었을 경우에는 메세지를 암호화하는 과정이 추가되고, 메세지를 암호화하는데 사용된 비밀키를 수신자에게 안전하게 전달하기 위하여 이를 메세지 토큰에 저장하여 전달하고, 기밀성 서비스가 요구되지 않았을 경우와는 달리 송신 PDU의 구성이 원래의 메세지가 아닌 암호화된 메세지를 수신자에게 송신하게 된다.이 과정을 설명한 것이 [그림 3(b)]이다.

제 1단계 : 서명값 생성

- 송신하고자하는 메세지를 해쉬함수를 이용하여 해쉬코드를 생성해낸다.
- 해쉬코드를 송신자의 개인키를 가지고 서명하여 메세지에 대한 서명값을 구한다.

제 2단계 : 메세지 암호화

- 랜덤하게 생성된 비밀키와 대칭키 알고리즘을 이용하여 메세지를 암호화한다.
- 비밀키를 수신자에게 안전하게 전달하기 위하여 수신자의공개키로 비밀키를 암호화한다.

제3단계 : 메세지토큰 생성

- 서명 알고리즘 식별자, 암호화된 비밀키, 메세지 순번 등으로 메세지 토큰의 필드들을 구성한다.
- 해쉬함수를 사용하여 메세지 토큰 데이타에 대한 해쉬코드를 구한다.
- 해쉬코드를 송신자의 개인키로 서명하여 메세지 토큰을 생성해낸다.

제4단계 : 송신 메세지 구성

- 수신자에게 암호화된 메세지, 서명값, 메세지 토큰으로 이루어진 PDU를 송신한다.

2) 증거 검증 과정

수신측에서 송신자로부터 메세지를 수신하였을 경우에 이 메세지에 대한 변경, 변조, 조작등의 유무에 대한 판단 근거로서 송신자가 보내온 증거 즉, 서명값, 메세지 토큰 등을 확인하는 절차가 필요하다. 이 검증 절차에 사용되는 메카니즘으로서는 서명값 확인을 위한 디지탈 서명 메카니즘, 메세지 인증 확인 및 메세지 토큰내에 암호화된 비밀키를 복호하기 위한 공개키 알고리즘, 메세지 인증을 위한 해쉬코드 생성용 해쉬함수가 필요하며, 수신된 메세지가 암호화되어 있을 경우 이를 복호하기 위한 대칭키 암호 알고리즘이 사용된다. 이와같이 수신측에서 증거를 검증하는 과정을 기밀성 서비스가 요구된 경우와 요구되지 않았을 경우를 각각 설명하면 다음과 같고, 이를 나타낸 것이 [그림 4]이다.

제1단계 : 메세지 수신

- 송신측으로 부터 메세지, 서명값, 메세지 토큰으로 이루어진 PDU를 수신한다.

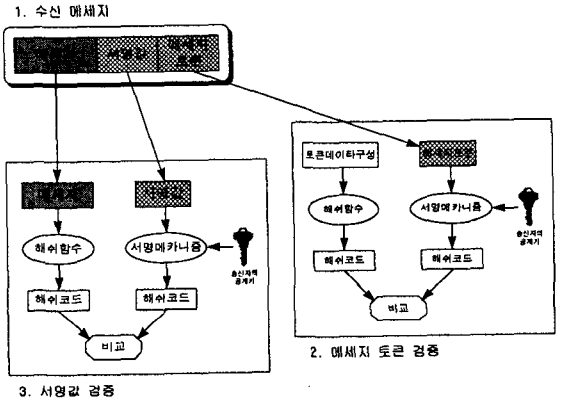
제2단계 : 메세지 토큰 검증

- 수신한 메세지 토큰을 송신자의 공개키를 가지고 서명메카니즘을 통하여 해쉬코드를 생성해낸다.
- 메세지 토큰 생성에 사용된 데이타들로 토큰 데이타를 구성한후, 이를 해쉬함수를 통하여 해쉬코드를 구한다.
- 서명 메카니즘을 통해 구한 해쉬코드와 해쉬함수를 통해 새로이 구한 해쉬코드 값을 비교한다. 여기서, 서로 값이 동일하면 PDU에 아무런 이상이 없는 것으로 판단하여 다음 단계인 서명값 확인 절차로 넘어가며, 만약 서로 값이 다를 경우에는 메세지 혹은 서명값등에 이상이

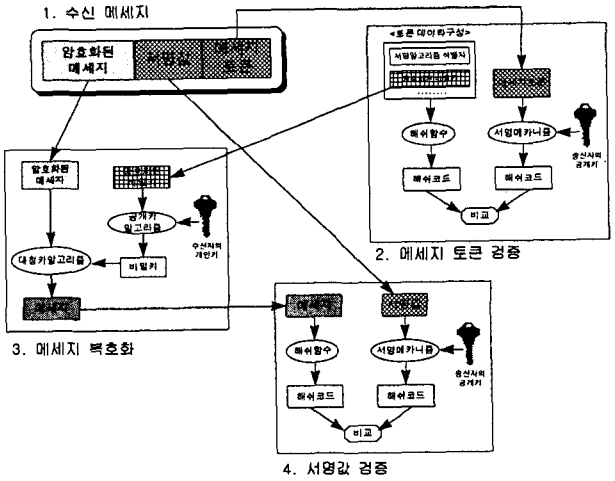
발생한 것이므로 에러 처리 과정으로 제어를 넘긴다.

제3단계 : 서명값 검증

- 수신한 서명값을 송신자의 공개키를 가지고 서명메카니즘을 통하여 해쉬코드를 생성해낸다.
- 수신한 메세지를 해쉬함수를 이용하여 해쉬코드를 구한다.
- 서명 메카니즘을 통해 구한 해쉬코드와 해쉬함수를 통해 새로이 구한 해쉬코드 값을 비교한다. 여기서, 서로 값이 동일하면 PDU에 아무런 이상이 없는 것이고, 만약 서로 값이 다를 경우에는 메세지에 이상이 발생한 것이므로 에러 처리 과정으로 제어를 넘긴다.



(a) 기밀성 서비스가 요구되지 않았을 경우



(b) 기밀성 서비스가 요구되었을 경우

[그림 4] 디지탈 서명값 검증 과정

수신된 메세지가 암호화된 메세지일 경우, 즉 기밀성 서비스가 요구되었을 경우에는 서명값 검증을 하기 전에 메세지 자체를 복호화 하는 과정이 부가적으로 필요하게 된다. 이의 처리과정은 다음과 같다.

제1단계 : 메세지 수신

- 송신측으로 부터 암호화된 메세지, 서명값, 메세지 토큰으로 이루어진 PDU를 수신한다.

제2단계 : 메세지 토큰 검증

- 수신한 메세지 토큰을 송신자의 공개키를 가지고 서명메카니즘을 통하여 해쉬코드를 생성해낸다.
- 메세지 토큰 생성에 사용된 데이타들로 토큰 데이타를 구성한후, 이를 해쉬함수를 통하여 해쉬코드를 구한다.
- 서명 메카니즘을 통해 구한 해쉬코드와 해쉬함수를 통해 새로이 구한 해쉬코드 값을 비교한다. 여기서, 서로 값이 동일하면 PDU에 아무런 이상이 없다는 것이 확인된 것이므로 다음 단계인 서명값 확인 절차로 넘어가게 된다. 또한 서명값을 확인하려면 서명의 대상이 본래의 메세지를 대상으로 하기 때문에, 이에 앞서 암호화된 메세지를 복호화 하는 과정이 필요하게 된다. 만약 해쉬코드 비교값이 서로 다를 경우에는 메세지 혹은 서명값등에 이상이 발생한 것이므로 에러 처리 과정으로 제어를 넘긴다.

제3단계 : 메세지 복호화

- 메세지 토큰 검증후, 이 토큰으로부터 암호화된 비밀키를 가져온다.
- 수신자의 개인키를 가지고 송신자로부터 전달받은 암호화된 비밀키를 복호한다.
- 복호된 비밀키를 가지고 암호화된 메세지를 대칭키 알고리즘을 통하여 복호하여 원래의 메세지를 구해낸다.

제4단계 : 서명값 검증

- 수신한 서명값을 송신자의 공개키를 가지고 서명메카니즘을 통하여 해쉬코드를 생성해낸다.
- 복호된 메세지를 해쉬함수를 이용하여 해쉬코드를 구한다.
- 서명 메카니즘을 통해 구한 해쉬코드와 해쉬함수를 통해 새로이 구한 해쉬코드 값을 비교한다. 여기서, 서로 값이 동일하면 PDU에 아무런 이상이 없는 것이고, 만약 서로 값이 다를 경우에는 메세지에 이상이 발생한 것이므로 에러 처리 과정으로 제어를 넘긴다.

3.3 KMS 시스템

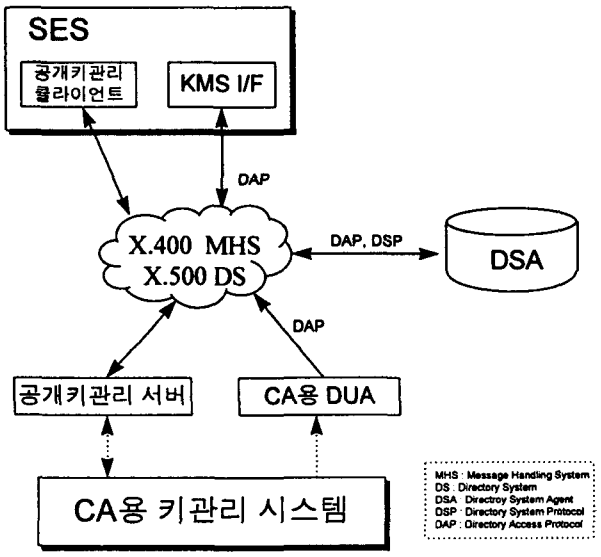
X.500 디렉토리 시스템[17][18][19][20][21]은 X.400 MHS 사용자의 다양한 정보를 관리하기 위하여 사용될 수 있다. X.400 MHS는 길고 복잡한 O/R(Originator/Recipient) 주소 대신 사용하기 편리한 디렉토리 주소로 각 X.400 MHS 사용자를 식별할 수 있으며, 디렉토리로 부터 O/R 주소, 배포목록, EDI 관련 정보를 조회하여 사용할 수 있다. 디렉토리는 또한, X.400 MHS의 정보보호 서비스를 구현하기 위해 인증에 관련된 정보를 관리하는 저장소로 사용될 수 있다.

[그림 5]는 EDI 정보보호 서비스용 키 관리 시스템 구조를 제시하고 있다. 제시된 키 관리 시스템은 그림과 같이 네가지 모듈로 구성된다. 첫째, 사용자들의 공개키에 대한 보증서를 발급, 관리하는 CA용 키관리 시스템, 둘째, CA와 DSA와의 통신을 담당하는 CA용 DUA(Directory User Agent), 셋째, SES가 필요로하는 키관련 정보를 제공해주는 KMS 인터페이스, 넷째, 사용자의 개인키 및 공개키 쌍을 생성하고 생성된 공캐키를 CA에게 전송하는 공개키 관리 클라이언트 및 서버로 구성된다.

강한 인증에 사용되는 공개키는 보증서 형태[18]로 디렉토리에 저장된다. 사용자는 공개키를 생성하여 CA에게 전달하고 CA는 수신된 공개키에 대한 보증서를 만들어 디렉토리에 저장한다. 디렉토리에 저장된 보증서는 분산된 디렉토리 시스템간에 자유롭게 통신 될 수 있고 디렉토리의 사용자들에 의해 통신 상대자를 인증하는데 사용될 수 있다.

사용자들이 상호 인증하는 것을 보다 효율적으로 하기 위하여 1개 이상의 CA가 존재하는 경우 CA들이 계층적으로 구성된다고 가정한다.

CA용 키 관리 시스템은 물리적으로 안전하기 위해 오프라인으로 존재한다. 사용자의 공개키는 공개키 관리 서버에 의해 수신되어 디스켓을 통하여 CA용 키 관리 시스템으로 옮겨진다.



[그림 5] KMS 시스템의 구조

가. 공개키 관리 클라이언트/서버

개인키와 공개키쌍은 각 사용자에 의해 생성된다. 공개키 관리 클라이언트는 사용자의 개인키 및 공개키를 생성한 후 공개키 관리 서버에 접속하여 사용자의 공개키 및 사용자 정보를 CA에게 전송하고 CA의 공개키를 수신한다.

나. CA용 키 관리 시스템

CA용 키관리 시스템은 공개키 관리 서버에 의해 수신된 사용자의 공개키에 대하여 보증서를 생성하는 키 생성 모듈, 발급된 보증서 또는 유효기간이 만료되거나 신뢰성에 문제가 생긴 보증서를 관리하는 키 관리 모듈, 보증서, 상호 보증서쌍, 취소 목록을 분배하는 키 분배 모듈, 모든 키의 생성, 관리, 분배 사항을 기록하여 문제 발생시 사용하기 위한 감사 추적 모듈, 사용자 인터페이스 등으로 구성된다.

다. CA용 DUA

사용자의 공개키에 대한 보증서는 오프라인 CA에 의해 생성된다. CA용 키 관리 시스템은 오프라인으로 존재하여야 하기 때문에 보증서, 상호 보증서쌍, 취소목록을 디렉토리에 저장하기 위해 통신망에 연결된 CA용 DUA를 이용한다. 보증서, 상호 보증서쌍, 취소목록을 디스켓을 통하여 CA용 DUA가 있는 컴퓨터로 옮긴 후, CA용 DUA를 디렉토리에 접속하여 키 관련 정보를 디렉토리에 저장한다.

라. KMS 인터페이스

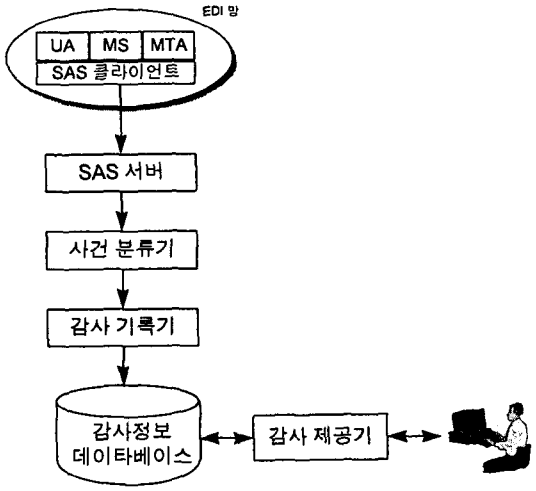
UA, MS, MTA에 정보보호 서비스를 구현하기 위해서는 키 관련 정보를 디렉토리에서 조회하여 사용하여야 한다. KMS 인터페이스는 정보보호 서비스를 쉽게 구현할 수 있도록 하기 위하여 통신 상대자의 신뢰할 수 있는 공개키를 얻는 복잡한 문제들을 내부적으로 처리하고 그것을 사용하기 위한 인터페이스를 제공한다. 이 인터페이스는 키 관련 정보를 얻기 위해 독립된 프로세서인 DUA의 도움을 받아 키 관련 정보를 얻는다. [표 4]는 KMS 시스템에서의 키 관리 인터페이스를 정리한 것이다.

[표 1] KMS 인터페이스

함수명	기 능
InitKeyman();	DUA를 이용하여 키 관리 인터페이스를 초기화한다
GetPrivateKey();	사용자 자신의 개인키를 제공한다
GetPublicKey();	보증서로부터 사용자의 공개키를 찾아 낸다
GetPublicKeyDN();	디렉토리명(DN)으로 표시되는 사용자의 신뢰할 수 있는 공개키를 찾아 낸다
GetCertificates();	사용자의 보증서들을 제공한다

3.4. SAS 시스템

EDI에서 보안 감사 추적을 위해 처리되어야 할 기능에는 사건 구별 기능, 감사 기록 기능, 감사 제공 기능, 감사 분석 기능, 감사 저장 기능 등이 있는데[9][25], 각각의 기능을 담당하는 모듈들이 모여서 전체 EDI 보안감사 시스템인 SAS를 구성하게 된다. SAS 시스템은 UA, MS, MTA에 감사처리를 위한 정보를 요청하면 각 UA, MS, MTA는 요청한 정보를 EDI 망을 통해 SAS에게 전달한다.

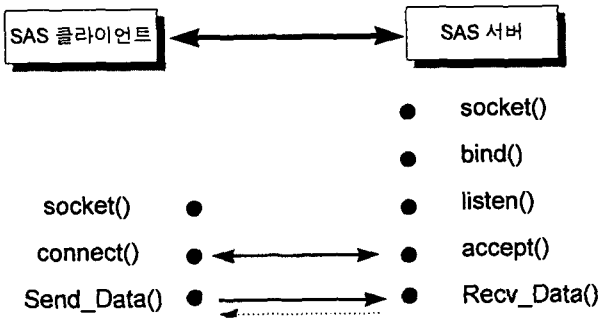


[그림 6] SAS 시스템의 구조

SAS의 구조는 [그림 6]과 같다. 첫째, SES로 부터 SAS로 감사 데이타를 전송하는 SAS 클라이언트/서버, 둘째, SES에서 전송된 데이타를 수신하고 해석하는 사건 분류기, 셋째, 분석된 데이타를 데이타베이스화하는 감사 기록기, 넷째, 저장된 데이타를 사용자의 요구에 따라 출력하는 감사 제공기로 구성된다.

가. SAS 클라이언트/서버

SAS 클라언트/서버의 주된 기능은 각 클라이언트에서 수집한 감사 데이타를 서버로 전송하는 것을 목적으로 한다. 이때 전송된 감사 데이타는 사건 분류기의 입력으로 사용된다. SAS 클라이언트/서버 간의 통신은 소켓(socket)을 이용한다.



[그림 7] SAS 클라이언트/서버 간의 통신

[그림 7]은 SAS 클라이언트/서버간의 메세지 통신시 호출되는 함수들을 나타낸 것이다. 먼저 서버에서는 socket() 시스템 호출을 수행하고, bind() 시스템 호출로 이름없는 소켓에 이름을 부여한다. Listen() 시스템 호출후, accept() 시스템 호출을 수행함으로써 클라이언트로부터 실제적

인 연결을 기다리는 상태가 된다. 반면, 클라이언트는 서버와의 연결을 설정하기 위해 socket() 시스템 호출후, 소켓 지정번호를 connect() 시스템 호출에 사용한다.

나. 사건 분류기

사건 분류기는 발생되는 여러 메시지들 중 EDI 보안 감사 추적에 필요한 메시지만을 선별하여 레코드 구성을 하는 모듈이며, 메시지 종류에 따라 적절한 감사기록 함수를 호출하도록 구성되어있다. 사건 분류기의 입력 정보는 EDI 시스템에서 통신을 위해 정의된 각 포트별로 분류되고, 다시 각 포트에서 처리하는 메세지의 유형별로 분류되어 처리된다. [그림 8]은 사건 분류기 함수의 내부 모듈을 나타낸 것이다.

```
EventDiscriminator( );
      {
         ID = GetMessageID(msg);
         switch(ID) {
            case Submission :
               call SubmissionGenerateRecord( );
               call SubmissionAuditWriter( );
               break;
            case Delivery :
               call DeliveryGenerateRecord( );
               call DeliveryAuditWriter( );
               break;
            case Report :
                     :
            }
         }
```

[그림 8] 사건 분류기 내부 모듈

다. 감사 기록기

감사 기록기는 사건 분류기에 의해 생성된 레코드를 입력으로 받아 이를 DBMS(DataBase Management System)를 이용하여 데이타베이스를 생성할 수 있는 질의를 만들어 데이타를 저장하는 기능을 담당한다.

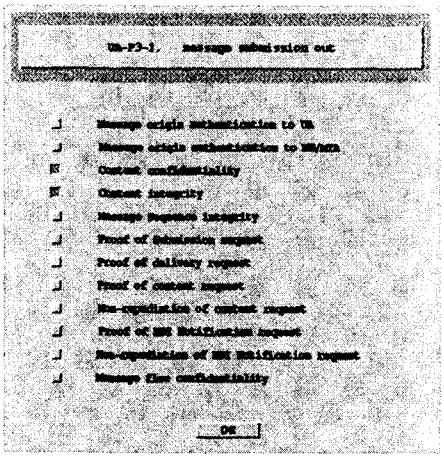
라. 감사 제공기

감사 제공기는 감사자로부터 감사자료 제공을 요청받으면 감사 정보 데이타베이스에 저장된 자료를 감사정보 관리 DBMS 질의어를 이용하여 감사자료를 제공받아 감사자에게 감사추적 레코드를 제공하는 역할을 수행한다. 감사 제공기에 의해 제공되는 감사 서비스는 부인봉쇄 서비스, 증명 및 검증 서비스, 사용자 등록 서비스, 검색 및 삭제 증명 서비스가 있다.

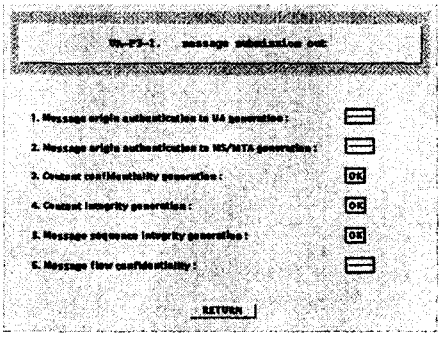
4. EDI 정보보호 시스템의 구현

이 장에서는 시스템 구현을 위한 요구사항 분석단계의 결과와 이 결과에 의한 설계에 따라 구현한 EDI 정보보호 시스템에 대하여 개략적으로 기술한다.

먼저 개발환경, 운용 및 유지보수 환경들을 살펴보면, SES, KMS, SAS 시스템 모두 Solaris 2.4 운영체제를 탑재한 Sun Sparc 20 워크스테이션상에서 개발되었다. 사용자 인터페이스는 X-Window/Motif를 기반으로 구현하였고, KMS에서의 디렉토리는 ISODE(International Standard Organization Development Environment) 8.0[22][23]에서 제공하는 QUIPU[24]를 사용하였고, SAS 시스템에서 DBMS는 UniSQL을 이용하였다.



[그림 9] 제출포트 서비스 요구 입력화면



[그림 10] 증거 생성 결과화면

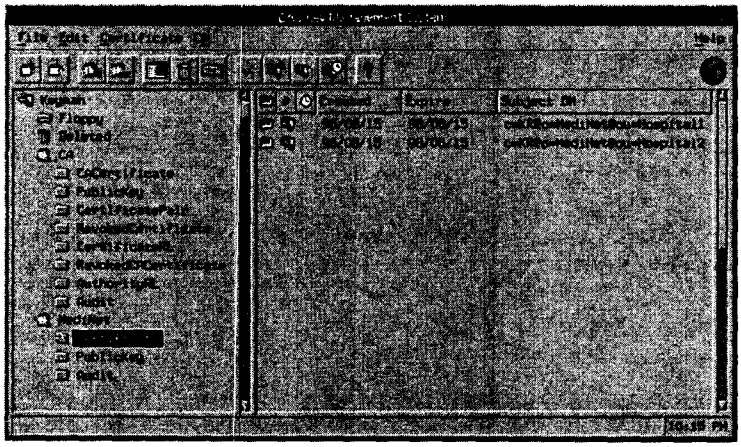
```
[CICHASH] ===> f0864fb5b8b8d8ed064ed6fd7967f927
[CI -> CIGen()]
Signature {
  algorithm {
    algorithm 2.5.1.8.1.1,
    parameters '10000000'B
  },
  signature
'19493d43d408c959ddfcf80e9bf8efba5901e9f626fbeac9caf4b94168c998d11d9e834c32002f8c441b45449
5db323386e3bad151136b571c81bf3c4f4f45800de02893920d93b9c2035da07ffcfb5f9a264fe57459fc0f37a
9924426d534e626119840a62fab941dba4dc2dea75c4ddb422d94e804fcfe9fc3cf26ea03f543'H
}
[SIG_TOKEN]
signature
'790d7873ad1501551b661f29eb4480e62634e60ee58b0b7efe7f745a2d0b326b724e87b8eb1c676272514bb
47fac05bdca30c672227f1c0af6351a511a5db7db84a1e7b66689566c848ee7dcd3b9f7b73e779d9cf98e08c
eddbf03aa9f523ecf516e305e84d23a8040f3934926698448086034faf58b8e6967eeca64b0b8d5c0'H
}
```

[그림 11] 증거 생성 결과값의 예

SES 시스템은 메뉴방식으로 구성되어 있다. 주 화면에 있는 주 메뉴바에은 EDI 시스템을 구성하는 UA, MS, MTA들이 각 프로토콜 별로 UA-P3, UA-P7, MS-P3, MS-P7, MTA-P3, MTA-P1으로 구분되어 있다. 주 메뉴 바에 있는 각 프로토콜 항목에 EDI 정보보호 서비스를 위한 부 메뉴들이 있는데, 모든 SES 서비스들은 이들 부메뉴를 선택함으로써 수행된다. 각 서비스들은 그 기능별로 제출 포트, 배달 포트, 관리 포트, 전달 포트, 검색 포트 등으로 구분되어 있다.

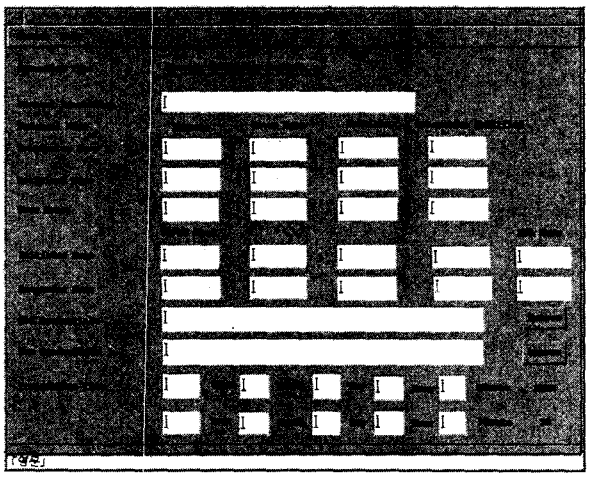
[그림 9]는 제출포트에서의 정보보호 서비스 요구 화면을 나타낸 것이고, [그림 10]은 이의 결과 화면을 나타낸 것이다. [그림 11]는 [그림 10]을 통해 내용 무결성 서비스용 증거 생성 결과값의 샘플 데이타를 나타낸 것이다.

KMS 시스템에서 KMS 인터페이스는 XDS(X.509 Directory System) API를 이용하여 구현하였으며, CA용 키 관리 시스템 및 CA용 DUA는 X-Window/Motif 환경하에서 구현하였다. 그리고, 키 관리 서버 및 클라이언트가 소켓 인터페이스를 이용하여 구현하였다. 다음 [그림 12]는 CA용 키관리 시스템의 주 화면이다.



[그림 12] CA 용 키관리 시스템의 주 화면

[그림 13]은 감사제공기의 감사 자료 검색 서비스 화면이다.



[그림 13] 감사 제공기의 검색 데이타 입력 화면

SAS 시스템은 감사자로부터 감사자료 제공을 요청받으면 감사 정보 데이타베이스에 저장된 자료를 UniSQL 질의문을 이용하여 감사자료를 제공받아 감사자에게 감사추적 레코드를 제공한다. 감사 제공기의 주 메뉴에서 서비스를 받기를 원하는 메뉴를 선택하면 각 주 메뉴에 대한 팝업 메뉴로 실제 서비스 받고자 하는 메뉴를 선택하다. 지원되는 서비스는 부인봉쇄, 증명과 검증, 보안 관리, MS 서비스 등이 있다.

5. 결론

EDI 정보보호 시스템은 크게 SES, KMS, SAS 시스템으로 이루어져 있다. SES 시스템은 PDU 단위의 인터페이스 방식을 취함으로써 X.435 표준에 부합한 기개발된 모든 EDI 시스템에 적용 가능하도록 설계되었고, 공통 사용 메카니즘들을 하나의 모듈로 모듈화하여 EDI 시스템과는 독립적으로 필요에 따라 사용 알고리즘을 변경, 교체 가능토록 하였다.

KMS 시스템은 공개키 암호 시스템을 사용하는데 있어 필수적인 키관리를 위해 X.509에서 정의하고 있는 인증구조에 따라 설계되었으며, 이 시스템은 OSI 뿐만아니라 TCP/IP를 근간으로하는 모든 네트워크에 구축될 수 있도록 설계되어 있어 X.400 MHS 뿐만 아니라 다른 OSI 응용 프로그램, 더 나아가 TCP/IP를 이용한 응용 프로그램에서도 사용될 수 있다. 뿐만 아니라, 최근 전자상거래, 웹 등 상거래용 응용프로그램들에 있어서 공개키를 이용한 정보보호 서비스가 필요적인 요소로 등장하면서 공개키 기반구조(Public Key Infrastructure:PKI)[26][27]에 대한 관심이 고조되고 있는데, PKI 구축에 핵심 요소인 CA시스템의 개발은 CA에 필요한 종합적인 기술을 확보함에 따라 향후 PKI 구축에 많은 기반기술을 제공할 수 있을 것이다.

SAS 시스템은 EDI 정보보호 서비스 처리시 향후에 논란의 여지가 발생할 수 있는 정보를 획득, 저장하여 분쟁 발생시 신뢰할 수 있는 제3자에게 제출할 증거를 저장, 관리할 수 있도록 하였다. 특히, 보안 감사 추적에 적용되는 사건들은 시간 종속적을 발생하는 사건들이므로 발생한 시점이 매우 중요한 역할을 한다. 이 시스템은 시간지원 개념을 적용하여 구축한 DBMS이므로 기존의 로그를 이용하여 처리하는 방식보다 관리나 이용 측면에서 많은 장점들을 가지고 있다.

본 논문에서 기술한 SES 설계 기술, KMS 설계 기술, SAS 설계 기술, 정보보호 메카니즘 활용 기술 등은 정보보호에 대한 주요 응용기술 능력 배양 뿐만 아니라 정보보호 시스템 개발의 기반을 구축하게 될 것이다.

본 연구 결과를 토대로 전자문서 거래에 있어서 정보 전달, 보관, 처리하는데 안전성을 제공하게 되어 기술적, 경제적, 사회적 측면에서 많은 진보를 가져다 줄 뿐만 아니라 통신 사업자의 사업성 측면에서도 신규 서비스 도출에 따른 수입원의 창출 등 많은 이점이 있을 것으로 사료된다.

참고문헌

[1] 강창구, "EDI 정보보호 서비스 분석", 제2차 안전한 EDI관련기술 심포지움, pp.3-18, 1996.3.

[2] 이정현, 윤이중, 김대호, 이대기, "X.435 EDI 정보보호 서비스 데이타 구조 분석", 한국통신정보보호학회지 제5권 제3호, pp.69-85, 1995.9.

[3] 윤이중, "안전한 EDI 시스템의 구조 설계", 제2차 안전한 EDI 관련기술 심포지움, pp. 31-43, 1996.3.

[4] 윤이중, "EDI용 안전성 서버 구현", 제3차 안전한 EDI 관련기술 심포지움, pp. 5-12, 1996.8.

[5] Chang Goo Kang, E Joong Yoon, Dae Ho Kim, Dai Ki Lee, "A Design of Secure EDI Systems," 8th Annual Canadian Computer Security Symposium, pp.397-416, 1996.

[6] B, Schneier, "Applied Cryptography(2/e)", John Wiley & Sons, pp1-758, 1996.

[7] Warwick Ford, "Computer Communications Security", PTR Prentice Hall, 1994.

[8] Paul Christmas, "EDI Implementation and Security", Elsevier Advanced Technology, 1994.

[9] Albert J. Marcella, Sally Chan, "EDI Security, Control, and Audit", Artech House, 1993.

[10] UN/ECE Recommendations for the UN/EDIFACT Message Level Security, 1993.

[11] ISO 7498-2, Network Security Architecture, 1989.

[12] ITU-T Recommendation X.400, Message Handling: System and Service Overview, 1988.

[13] ITU-T Recommendation X.402, Message Handling Systems: Overall Architecture, 1988.

[14] ITU-T Recommendation X.411, Message Handling Systems: Message Transfer System: Abstract Service Definition and Procedures, 1988.

[15] ITU-T Recommendation X.413, Message Handling Systems: Message Store: Abstract Service Definition, 1988.

[16] ITU-T Recommendation X.435, Message Handling Systems: EDI Messaging Systems, 1991.

[17] ITU-T Recommendation X.500, The Directory: Overview of Concepts, Models, and Services, 1988.

[18] ITU-T Recommendation X.509, The Directory: Authentication Framework, 1988.

[19] ITU-T Recommendation X.519, The Directory: Protocol Specifications, 1988.

[20] ITU-T Recommendation X.520, The Directory: Selected Attribute Types, 1988.

[21] ITU-T Recommendation X.521, The Directory: Selected Object Classes, 1988.

[22] The ISO Development Environment: User's Manual, Volume 1: Applications Services, 1991.

[23] The ISO Development Environment: User's Manual, Volume 3: Applications Cookbook, 1991.

[24] The ISO Development Environment: User's Manual, Volume 5: QUIPU, 1991.

[25] ACM SIGSAC, Security Audit & Control Review, vol. 13, no. 3, July 1995.

[26] IETF X.509 PKI Working Group, Internet Public Key Infrastructure Part I: X.509 Certificate and CRL Profile, R. Housley, W. Ford and D. Solo.

[27] IETF X.509 PKI Working Group, Internet Public Key Infrastructure Part III: Certificate Management Protocols, S. Farrell, C. Adams and W. Ford.

[28] ISO/IEC DIS 11770-1, key management, part 1: key management framework.

[29] ISO/IEC DIS 11770-1, key management, part 2: mechanisms using symmetric techniques.