通信情報保護學會誌
第4卷 第2號, 1994. 6

EDI 표준과 관련된 EDI 보안 서비스에 관한 고찰

A Study of EDI Security Services related to EDI Standards

전 윤 호*, 이 필 중**

요 약

EDI(Electronic Data Interchange)는 기업간에 서로 약속된 format에 의해서 전자적으로 문서를 주고 받을 수 있도록하는 규칙으로 그 성질상 보안서비스의 구현은 필수적이다. 본 고에서는 EDI 보안 서비스를 국제 표준과 관련하여 고찰하였다. 먼저 국제 표준을 통신표준과 문서표준으로 나누어 설명한 다음, 각 표준에서 제시하는 EDI 보안 서비스의 개념을 고찰하였다.

1. 서 론

컴퓨터 기술의 급속한 발달에 힘입어서 기업체를 포함하는 대부분의 조직체에서는 업무에 필요한 정보를 컴퓨터에 기반을 둔 시스템을 써서 처리하게 되었고, 이러한 경향은 점차로 가속화되고 있는 추세이다. 또한 컴퓨터를 이용하는 통신망의 확장과 통신기기의 보급으로 대량의 정보를 원거리에서도 신속하고 정확하게 원하는 장소에까지 전달할 수 있게 되자 기존의 보급된 컴퓨터 시스템을 이용하여 거래하고자 하는 상대방 기업체와 중요한 거래정보를 주고 받는 시스템의 구축이 가능하게 되었고 이는 EDI라는 전자 문서 교환 시스템으로 나타나게 되었다.

그런데 초기 단계의 EDI에서는 전자문서의 양식이 체계적으로 조직된 것이 아니고 교환에 있어서도 주로 거래 상대방과의 약속된 형식이나 몇몇 동종 업계의 기업들을 포함하는 소규모의 서비스 업자들에 의해서 정해진 방식으로 이루어졌다.

그러나 점차 교역의 범위가 확대되고 보다 많은 거래 상대방과 전자문서를 교환할 필요가 커지게 되고 이와 함께 서로 다른 양식의 전자문서를 교환하는 것이 어렵게 됨에 따라서 EDI 문서 및 전송 방식에 있어서 표준의 필요성이 대두되게 되었다. 이에 미국 및 유럽을 중심으로 하여 문서표준으로서 각각 ANSI X12(American National Standards Institute)[1], GTDI(Guidelines for Trade Data Interchange)가 제정되었고, 이를 통합할 필요성에 따라 UN(United Nation)에서 1987년 JEDI, ISO(International Standardization Organization), 유럽경제위원회 ECE의 공동작업에 의해서 EDI FACT(EDI For Administration, Commerce and Transport)가 마련되고, ISO에 의해서 ISO 9735[2]로 채택, 권고되기에 이르렀다.

* 종신회원, 포항공과대학 전자전기공학과
** 종신회원, 포항공과대학 전자전기공학과

한편 CCITT(The International Telegraph and Telephone Consultative Committee)에서는 생성된 EDI 문서를 이기종 컴퓨터 시스템을 사용하는 상대방에게도 자유롭게 보낼 수 있도록 하는 통신 표준으로 X.400(1988)MHS(Message Handling System)[3]에 기반을 둔 X.435를 발표하였다[4].

여기서 문서 표준은 EDI 거래에서 교환되는 자료의 형식과 내용에 관한 표준이며, 통신 표준은 EDI 문서를 통신망을 통하여 거래 상대방에게 보내는 것에 관한 표준을 의미한다.

위의 EDI 발달과정을 고려해 볼 때 현재의 EDI는 다양한 네트워크 환경에서 교환되는 기업간의 정보를 포함하는 문서들을 업체간 또는 공공기관간의 표준화된 format과 코드체계를 이용하여 인간의 중재없이 상호합의된 형태의 전자 거래 문서로 만들어서 독립된 컴퓨터 응용 프로그램간에 관리, 교환하는 전자식 시스템이라고 할 수 있다.

이와 같이 EDI에서 다루는 대상은 기업의 자산이나 신용에 관련된 중요한 거래 문서이므로 EDI 시스템 자체와 거래문서들을 안전하게 관리하는 것은 매우 중요한 문제라 할 수 있다. 국내에서도 EDI에서의 정보보호 프로토콜 설계와 EDI 보안 시스템 등에 관해서 연구가 있었다[5, 6, 7, 8].

본 고에서는 EDI 보안시스템을 구현하는데 있어서 필요한 문서 표준으로서 UN/EDIFACT와 통신 표준으로 X.435를 설명하고, 각 표준에서 제시하고 있는 보안 서비스에 관하여 설명한 후에, 각 표준을 보안 서비스의 구현이라는 관점에서 검토하기로 한다.

2. EDI 문서 표준 - UN/EDIFACT

EDIFACT(Electronic Data Interchange For Administration, Commerce and Transport) 는 1985년에 EDI의 보편화 추세와 함께 공통적인 세계 EDI 표준을 제정하자는 의견이 제기되고 여기서 비롯되어 UN 산하의 유럽경제위원회(UN/ECE)가 중심으로 제정하고 1987년 9월에 국제표준(ISO 9735)으로 승인된 국제적인 EDI 표준이다. 이 표준은 북미 표준인 ANSI X12 표준과 유럽 표준인 GTDI를 기반으로 개발되었으며, 여러 산업 및 은행, 정부기관 등의 국제적인 상호 데이타 교환을 용이하게 하는 국제 표준 규격이다. EDIFACT는 여러 표준을 근간으로 하고 있으므로 전체적인 기능을 가지고 있으면서도 유연성과 효율성이 높은 특징을 가지고 있다. 보다 명확하게 설명하면, EDIFACT는 제조업자, 수출자, 분배업자, 운송업자, 은행, 그리고 정부기관 사이의 거래 데이타를 국제적인 범위에서 전자적으로 용이하게 교환할 수 있도록 하는 원칙들의 집합으로, 물품 또는 용역과 관련된 데이타의 전자교환에 관한 다음과 같은 표준, 지침서 및 디렉토리의 집합을 총칭하는 말이다.

- 유엔 거래 데이타 항목집(UNTDED) (ISO 7392)
- 유엔 거래 데이타 교환집(UNTDID)
  - 메세지 구문 규칙(syntax rule) (ISO 9735)
  - UN/EDIFACT 표준 전송 항목집 : EDSD
  - UN/EDIFACT 표준 메세지집 : EDMD
  - 메세지 설계 지침서 및 구문실행 지침서
  - UNCID

또 EDIFACT에서는 EDI에 관련된 권고안으로 다음과 같은 사항을 정했다.

- 기존의 서류를 전자 file로 대체시킨다.
- 국제표준에 따라 작성된 메세지를 통일적으로 제공한다.
- 개방형 통신(open communication)을 통하여 EDI 응용 환경을 구현한다.
- 네트워크 및 서비스를 최대한 이용할 수 있게 한다

위의 규칙에 의해서 EDIFACT 메세지 작성 방법은 컴퓨터, 시스템, 응용 프로그램, 통신방법 등과는 상호 독립적으로 처리될 수 있고 이러한 EDIFACT는 OSI 7 계층에서 응용 계층(Application layer)의 통신 프로토콜 위에 탑재되어서 third party의 서비스를 거쳐서 전송되는 것으로 표현할 수 있다.

거래를 개시하고자 하는 사람이 상대방과 호(call)를 설정하고 나서 호가 끝날 때까지 원하는 거래를 수행하는 동안을 connection이라 할 때 하나의 connection은 여러 개의 interchange로 구성될 수 있는데 하나의 interchange는 여러 개의 functional group이나 message들로 구성되고 하나의 functional group 안에 여러 message가 있을 수 있다[2]. 각 서비스 segment를 포함하는 interchange의 구조적인 계통은 그림 1에 나타나 있다.

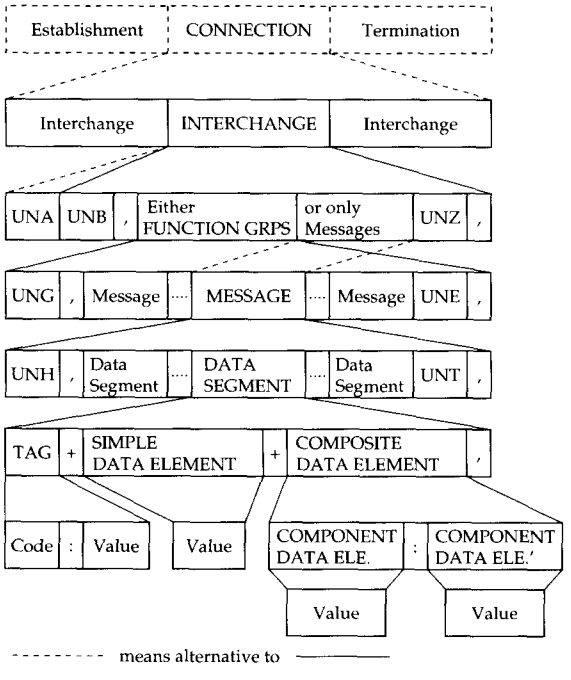


그림 1 : EDI interchange의 구조적인 계통도

EDIFACT 메세지는 기본적으로 다음의 5개 기본 요소로 구성된다.

- Interchange
- Functional group
- Message
- Segment
- Simple/composite data element

여기서 EDIFACT 메세지의 기본단위는 data element인데 data element들은 정해진 구문(Syntax)에 따라서 다양하게 서로 결합하여 message를 형성한다.

그 외에 표준에서 제시하고 있는 자세한 구문 규칙과 EDIFACT 메세지의 기본 구성 요소들을 기술하는 방법 등은 본 고의 범위를 벗어나므로 생략하기로 한다.

그런데 위와 같이 EDIFACT를 이용할 때 거래선이 한 곳일 경우에는 양 당사자만을 연결하는 독자적인 EDI 시스템을 구축하면 되지만 실제로는 거래 상대방이 여러 군데의 거래선을 가지고 있는 경우가 보통이므로 기업이 독자적인 시스템을 구현하기가 어려워진다. 왜냐하면 거래 상대방의 통신 프로토콜이 각각 다르고 통신하고자 하는 시간의 차이도 있을 수 있으며 상호 호환성이 없는 하드웨어나 응용 프로그램을 사용하고 있을 수도 있기 때문이다.

이와 같은 문제점을 해결하기 위해서는 국제적으로 표준화된 통신 프로토콜의 사용이 필요한데 여기에 적합한 것으로 CCITT/ISO에서 1990년에 X.400 MHS 프로토콜을 EDI 에 맞게 수정한 X.435 프로토콜을 들 수 있다.

3. EDI 통신표준(X.435)

기본적으로 EDI 메세지의 교환은 비동기 형태를 띄고 이루어지므로 축적 및 전송 방식에 기반한 X.400 MHS를 EDI에 적합한 통신 프로토콜로 생각하게 되었다[9].

X.435는 1990년에 EDI를 위해서 X.400을 수정하여 사용하던 기존의 방식인 북미의 P0 접근방식, 유럽의 P2 접근방식의 차이점을 없애고 EDI의 사용을 증진시키기 위하여 CCITT에 의해서 제정된 EDI를 위한 통신 프로토콜 권고안이다.

X.435에서는 1984년, 1988년 X.400과는 달리 EDI 데이타 교환을 위해서 최적화된 새로운 메세지 형식을 나타내는 Pedi라는 프로토콜과 EDI 응용 시스템을 MHS 환경안에 포함시키는 구조적인 모델(architectural model)도 정의하고 있다[4].

3.1 EDI 모델

X.435에서는 EDI의 기본적인 모델로 사용자, EDIMG-User(EDI Messaging System User)와 시스템, EDIMS(EDI Messaging System)를 포함하는 환경, EDIME(EDI Messaging Environment)를 정의하고 있는데 이는 그림 2와 같다.

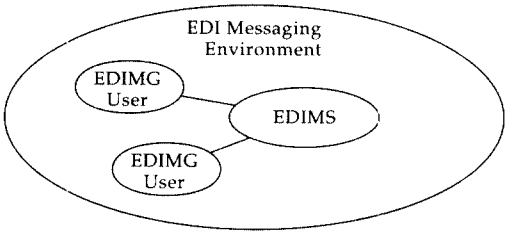


그림 2 : EDI Messaging Environment

EDIMS는 다시 EDI-UA(EDI-User Agent), EDI-MS(EDI-Message Store), EDI-AU(EDI-Access Unit), MTS(Message Transfer System)로 구성되며 그림 3은 EDIMS의 구성도를 나타낸다[10].

MTS는 EDIMS 내에서 실제로 메세지를 전송하는 부분으로서 축적 및 전송 방식의 통신 시스템이며 EDIMS의 근간을 이룬다.

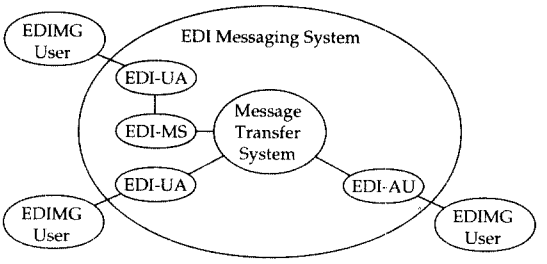


그림 3 : EDI Messaging System

EDI-AU는 일종의 게이트웨이(gateway)로서 텔렉스 네트워크(telex network)이나 우편 시스템(postal system)과 같은 다른 통신 시스템과 EDIMS를 연결하는 기능을 한다.

1990년 MHS에서는 1988년 MHS에서 정의된 UA[11]와 비슷한 개념인 EDI-UA를 정의하고 있다. EDI-UA는 EDI에 적합한 content type을 가지는 EDIM(EDI Message)를 생성하는데, EDIM의 heading에는 EDIFACT interchange header segment에 들어 있는 information fields들을 포함하게 된다. 또한 EDI-UA는 메세지 송신자가 원하는 여러가지 부가적인 서비스를 요청받아서 수행한다. EDI가 포함된 MHS 환경에서 Pedi 프로토콜과 다른 MHS 프로토콜과의 관계가 그림 4에 나타나 있다.

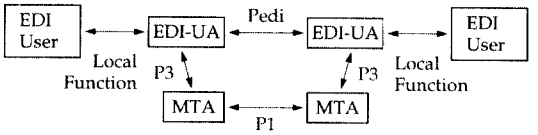


그림 4 : MHS에서의 Pedi 프로토콜의 위치

또한 EDI-UA 사용자들의 필요를 만족하는 EDI-MS도 정의하고 있는데 이것은 EDI-UA가 PC 등에서 수행되는 프로세스라고 할 때 항상 동작할 수는 없으므로 MTA에서 수신되는 EDI message들을 일단 맡아서 저장했다가 EDI-UA가 원할 때 받을 수 있도록 하는, 시스템의 선택적인 구성요소이다.

3.2 EDI 메세징(EDI Messaging)

EDI 메세징(Messaging)은 EDIM(EDI Message)와 EDIN(EDI Notification)라는 2가지의 정보객체(information object)의 교환으로 이루어진다[41].

- EDIM : 비즈니스 거래를 포함하는 EDI message
- EDIN : EDIMS에서 EDI interchange가 성공적으로 전달되었는가에 대한 정보를 포함하는 EDI 통지서.

EDIM은 사용자들 사이에서 전달되는 정보들의 집합으로 ASN.1 의 표기상 다음과 같다.

```
EDIM :: SEQUENCE OF {
                heading     Heading,
                body        Body }
```

Heading field가 EDI message의 특성을 결정하는 요소들로 구성되는데 반해서 Body field는 하나 또는 그 이상의 body part로 구성된다. Heading field에는 EDIM의 확인자, 제출자 및 수신자의 이름, body part 형태, 관련 메세지, EDI 응용 보안 요소 등이 포함된다. 반면에 body field에는 EDI의 정보 객체를 포함하는 하나의 "Primary body part"를 가지는데 이 body part는 EDI interchange 자체이거나 회송되는(forwarded) EDIM이다. EDI interchange는 EDIFACT, ANSI X12 등에서 정의된 정보객체를 포함한다. 다른 body part들은 primary part와 관련된 또 다른 형태의 메세지이며 상대적인 body part의 예로는 교환되는 메세지와 연관되어 사용되는 텍스트, 음성, 그래픽 정보들이 있다. 그림 5는 EDI 메세지의 구조를 나타낸 것이며, 그림 6은 EDIM Body part의 구조를 나타내고 있다.

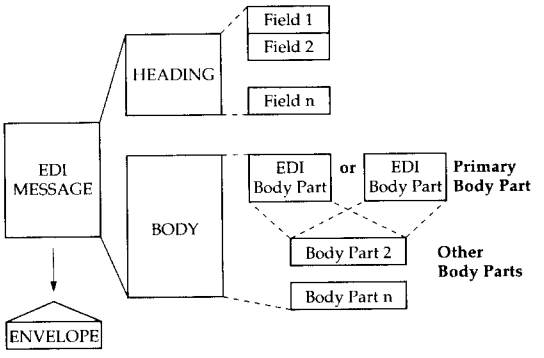


그림 5 : EDI message의 구조

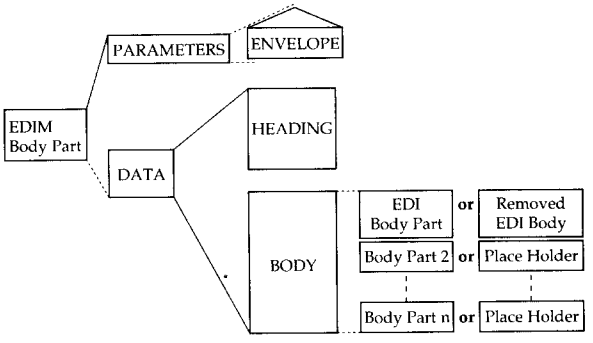


그림 6 : EDIM Body part의 구조

EDI는 결제의 수단으로 쓰일 수도 있으므로 수신자가 EDI message를 받는 것을 거부할 수 있는 기능이 포함되어야 하는데 이 경우에 필요한 것이 EDIN으로 이런 경우의 EDIN은 중간의 MS나 MTA는 관여하지 않는 end-to-end에 기초를 두고 동작해야 한다. 그런데 EDI 메시지를 수신하는데 관여하는 수신측 EDI-UA, EDI-MS, EDI-AU 등이 통신 상의 문제 등으로 인하여 EDI message 수령을 거부할 수도 있는데, 이 경우에는 수신측 EDI-UA 등도 EDIN 을 생성할 수가 있다.

사용자가 EDIM Responsibility를 받아들이는가의 여부에 따라서, EDIN은 다음과 같이 나뉠 수 있다.

- PN(Positive Notification) : EDIM Responsibility를 사용자가 받아들인다는 통지. 사용자는 EDIM을 임의로 처리(processing) 할 수 있다.
- NN(Negative Notification) : 사용자가 EDIM Responsibility를 받아들이기를 거절한다는 통지.
- FN(Forwarded Notification) : EDIM과 EDIM Responsibility가 하나 또는 그 이상의 EDI-UA에게 전송되었다는 표시.

여기서 EDIM Responsibility는 1990년 MHS에서 도입된 개념으로, EDI 환경하에서 EDIM의 "수용"을 확인하는 수단이다[12]. EDIN의 구조는 그림 7에 나타나 있다.

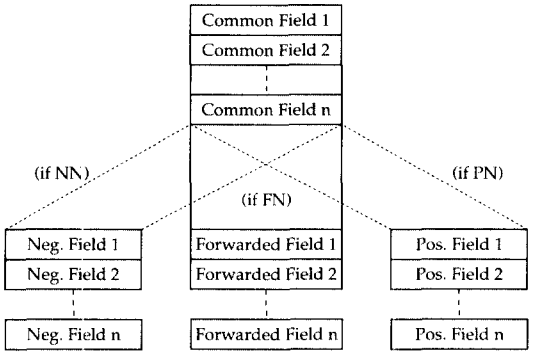


그림 7 : EDI Notification의 구조

3.3 X.435와 OSI 7 계층

X.435는 통신 프로토콜로서 X.400 MHS를 발전시킨 형태이기 때문에 OSI 7 계층과의 대응관계도 X.400 MHS와 동일하다고 할 수 있다. MHS는 각 구성요소의 기능적 측면에서 OSI 7 계층에서 응용계층과 표현계층을 포함하며, 실제로 시스템을 구현할 경우에는 응용 계층의 ACSE (Association Control Service Element)와 ROSE(Remote Operations Service Element) 부분과 연동하여 동작한다 [그림8]. 그림 8에서 X.435는 응용 계층과 표현 계층 모두를 포함함을 알 수 있다.

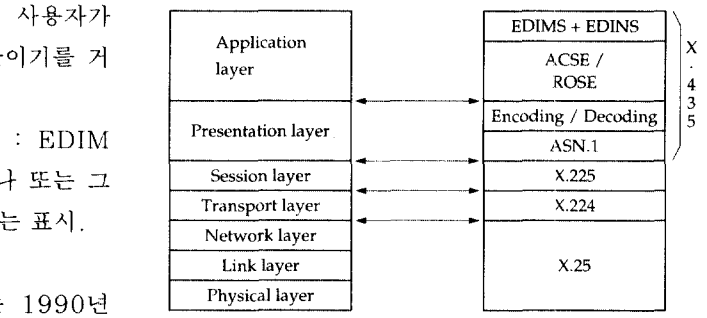


그림 8 : X.435와 OSI 7 계층과의 관계

3.4 EDI 서비스 (F.435)

F.435는 1990년 CCITT에서 X.435와 함께 발표한 EDI 서비스에 관한 권고안으로 EDI Messaging의 전체적인 시스템과 서비스(F.435에 설명된 서비스 중에서 EDI에 추가된 보안 서비스에 대해서는 6.2절에서 설명하기로 한다.)를 정의하고 서비스 요소 분류 및 기능을 설명하고 있다[10].

그 주요 내용으로는 EDI 메세지 처리 서비스에서의 시스템 동작 및 각종 외부 매체와의 연동 서비스 구성, EDI의 보안 서비스, 디렉토리(Directory), MS 등의 개념이 있다.

EDI의 기술적인 측면이 소개된 것이 X.435라면 F.435는 각종 서비스에 대한 개념을 소개한 것이라 볼 수 있고, 서비스 요소 개념들의 많은 부분이 F.400 series에서 연유하고 있다.

4. 공개 디렉토리 표준(X.500)과 디렉토리 인증 골격(X.509)

4.1 공개 디렉토리 표준, X.500

X.500 series는 복잡한 통신 시스템에서 통신

에 관계되는 실체(entity)들을 고유하게 지정하면서 사용자가 이용하기에 편리한 명칭을 실체와 연결시키는 서비스를 표준화한 것이다[13, 14].

디렉토리 서비스는 사람들이 이용하기 쉬운 이름을 통신에 관련된 실체들에 할당하고 그 외에 실체와 관련된 정보를 저장하고 관리하는 것을 말한다. 여기서 디렉토리는 디렉토리 서비스를 제공하는 개방형 시스템들의 집합이고 디렉토리가 관리하는 정보들을 DIB(Directory Information Base)라고 한다[16].

디렉토리 사용자는 디렉토리에 액세스함으로써 디렉토리 서비스를 받을 수 있다. DUA(Directory User Agent)는 디렉토리 사용자를 대표하는 응용 프로세스로 실제로 디렉토리에 액세스하고 사용자가 원하는 서비스를 받기 위하여 디렉토리와 interact하는 부분이다. 보통 디렉토리는 여러 조직에서 가지고 있는 많은 양의 정보로 구성되어 있으므로 디렉토리 시스템은 분산구조를 가지게 된다. 이러한 분산구조를 가지는 디렉토리 시스템은 그림 9와 같이 나타낼 수 있다[14].

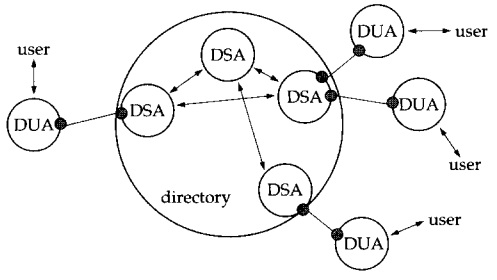


그림 9 : 디렉토리 구성요소 및 사용자와의 관계

그림 9에서 DSA(Directory System Agent)는 디렉토리를 구성하는 응용 프로세스들을 말하는데 하나의 디렉토리 안에 여러 개의 DSA가 존재함으로 해서 다수의 DUA가 각기 다른 장소에서 독립적으로 디렉토리를 액세스하는 것이 가능해진다. 각 DSA는 디렉토리의 일부분을 저장하고 관리하기 때문에 다른 DSA의 정보를 사용자에게 서비스해 주기 위해서 서로 연결되어 있어야 한다. 또한 DSA는 DUA 뿐만 아니라 다른 DSA에게도 서비스를 제공할 수 있다[16].

4.2 X.509 디렉토리 인증 골격

X.509 디렉토리 인증 골격은 디렉토리 내에서 인증 메카니즘과 인증서(certificate), 상대방의 공개키를 안전하게 얻는 방법, 키 생성/인증서관리 방법 등을 설명하고 있다. 본 절에서는 공개키 획득 메카니즘에 필요한 인증서에 관한 내용과 키 생성 방법을 살펴 보기로 한다.

4.2.1 Certificate

X.500 series 권고안의 보안 메카니즘은 주로 공개키 암호 시스템에 근거를 두고 있는데 공개키 암호 시스템을 쓸 때, 메세지 기밀성 서비스(message confidentiality service)를 메세지에 적용하여 송신하거나, 상대방의 디지탈 서명을 확인하기 위해서는 상대방 사용자의 공개키가 정말로 그 사람의 공개키가 맞는지 반드시 확신할 수 있어야 한다. 그러므로 통신을 시작하려고 하는 사용자가 상대방 사용자의 이름과 그 사람의 공개키 사이의 관계를 확인할 수 있는 수단이 필요한데, 이 수단을 제공하는 것이 X.509에서 정의된 인증서이다[17].

사용자의 공개키에 대한 인증서는 인증기관 CA (Certification Authority)가 발행하는데 CA는 사용자가 신뢰할 수 있는 실체이어야 한다. 인증서는 위조할 수 없도록 하기 위하여 그것을 발행한 CA의 비밀키로 암호화된 사용자의 정보와 공개키를 의미하며 그 정확한 형식은 식 1과 같다[15].

$$CA\langle\langle A\rangle\rangle = CA\{SN,\ AI,\ CA,\ ID_A,\ A_P,\ T^A\} \quad (1)$$

$CA\langle\langle A\rangle\rangle$는 사용자 A에 대해서 CA가 생성한

인증서라는 의미이고, $CA\{I\}$는 데이타 블럭 I와 CA의 비밀키에 의해서 암호화된 digest를 가리키는데, 이는 곧 CA에 의해서 디지탈 서명된 데이타 블럭 I를 나타낸다.

여기서 SN은 인증서의 일련번호이고, AI는 CA가 인증서에 대한 서명을 생성하는데 사용한 알고리듬으로 공개키 암호 시스템의 경우에는 asymmetric cipher와 해쉬함수[18]를 가리킨다. A_P는 사용자 A의 공개키이고 T^A는 인증서의 유효한 개시일과 마지막 날로 구성된 유효기간이다.

인증서가 유효한 것인지 확인하기 위해서는 식 2, 3와 같이 인증서의 내용을 해쉬한 결과와, CA의 공개키로 서명을 복호화한 것을 비교하는 과정이 필요하다.

$$CA\langle\langle A\rangle\rangle = \{AI, CA, A, A_P, T^A, CA_S\{h(AI, CA, A, A_P, T^A)\}\} \quad (2)$$

$$CA_P[CA_S\{h(AI, CA, A, A_P, T^A)\}] = h(AI, CA, A, A_P, T^A) \quad (3)$$

여기서 $CA_S\{I\}$, $CA_P\{I\}$는 각각 CA의 비밀키 및 공개키로 데이타 블럭 I를 암호화한 것을 나타내고 $h(I)$는 I를 일방성 해쉬함수 h에 대입하여 얻은 결과를 의미한다. 만약 비교한 두 가지 결과가 같다면 사용자는 지정된 CA가 인증서를 작성했음을 확신할 수 있다. 그 다음 과정으로 T^A를 검사하여 인증서의 유효기간을 확인한 후, 인증서에 포함된 공개키를 믿고 사용하게 된다.

4.2.2 공개키 / 비밀키 생성

사용자의 공개키 / 비밀키 쌍은 다음 3 가지 방법에 의해서 만들 수 있다.

1. 사용자가 자신의 키 쌍을 생성할 수 있는데 이 때 사용자는 암호학적으로 안전한 키 쌍을 만들 수 있는 능력이 있어야 한다.
2. 제 3 자가 키 쌍을 만들어서 비밀키를 안전한 방법으로 사용자에 전달하는 방법이 있을 수 있는데, 이 때 만들어진 키 자체와 키를 만드는데 관련된 정보들을 즉시 없애는 것이 필요하다. 제 3 자 자신과 키를 생성하는 과정이 tamper free하도록 하는 물리적인 보안 수단이 강구되어야 한다.
3. 2의 특별한 경우로서 CA가 키 쌍을 만드는 것도 가능하다.

1, 2, 3의 경우 모두 채택하는 암호 시스템이 키 생성에 부과하는 조건들을 만족해야 한다.

5. 메세지 수준의 보안 서비스

최근 EDI 국제 표준 메세지인 UN/EDIFACT에서는 사용자들의 급증하는 보안 요구를 수용하고 사용자 간에 보다 신뢰성있는 문서교환 시스템 구축을 위해 메세지 수준(message level)에서의 보안구조에 대한 권고안을 제시하였다[19].

이 권고안에서 제시한 방법은 EDI 서비스 수행에 있어서 보안 상의 제반 문제를 사용자들이 사용하고 있는 컴퓨터 시스템이나 통신망의 보안기술과는 독립적으로 최종 사용자 시스템간(End-to-End)에서 해결할 수 있도록 하는 것이다. EDIFACT의 메세지 구조는 사용자간에 교환하는 문서에 대한 전송정보를 담고 있는 interchange, functional group 등으로 구조화되어 있다. 그리고 사용자들이 중간에 어떠한 전송 경로나 시스템을 사용하든지 하부의 통신 프로토콜과 관계된 보안기술과는 독립적으로 EDI 사용자 시스템 자체의 완결적 구조를 가진 보안 시스템 구성을 위해 메세지 차원의 보안 구조를 제시하고 있다. 또한 권고안은 사용가능한 모든 보안 메카니즘들을 써서 지정된 보안 서비스를 구현할 수 있도록 하고 있으며, 보안 서비스를 적용할 수 있도록 개별적인 메세지 자체를 바꾸는 방식이 아니라 오히려 실제 응용에 무관하게 어떤 실제 메세지도 변경하지 않고 적용될 수 있도록 하는 총체적 접근방식

(global approach)을 채택하고 있다.

다음 절에서 이 권고안에서 상정하고 있는 보안의 위협요소와 그에 대한 해결방안들과 이러한 보안 서비스를 구현하기 위해서 UN/EDIFACT에서 채택하고 있는 EDIFACT 메세지 수준 보안 구조를 설명하기로 한다.

5.1 보안의 위협요소 및 해결방안

EDIFACT Message는 전자 매체(electronic media)를 통해서 전송되고 보관되는 과정에서 다음과 같은 위협상황에 노출될 수 있다.

- 고의적으로 또는 허가받지 않고 메세지 내용을 노출하는 것.
- 불법 메세지를 고의적으로 첨가하는 것.
- 메세지의 복제, 유실, 재전송.
- 메세지 내용의 수정.
- 메세지의 삭제.
- 송신자나 수신자가 메세지 책임(message responsibility)을 부인하는 것.

위에서 언급한 위협상황을 해결하기 위해서는 메세지의 안전한 전송에 관련된 실체(entity)들 - 메세지를 전송하기 전에 메세지를 안전하게 처리하는 송신자와 수신된 메세지의 안전성을 확인하는 수신자 - 을 식별하는(identify) 것이 필요하다. 이러한 실체들은 security segment에 표시될 수 있는데 공개키 암호 알고리듬과 비밀키 암호화 알고리듬의 두 가지 방법에 의해서 구현될 수 있다. 공개키 암호 알고리듬을 사용하게 되면 CA(Certification Authority)가 존재하여 모든 사용자들을 등록하고 사용자들의 공개키를 식별하는 제 3자의 역할을 하는 것이 필요하다. 사용자들의 공개키에 대한 정보는 인증서에 의해서 얻는 것이 가능하며, 인증서는 사용자의 공개키와 개인 정보가 포함된 메세지를 CA의 비밀키로 서명한 디지탈 서명이다. 사용자는 다른 사용자의 공개키를 얻고 싶을 때는 CA의 공개키로 원하는 사용자에 해당하는 인증서를 복호화하여 안전하게 공개키를 얻을 수 있다. 그런데 이러한 경우에 CA의 공개키에 대한 신뢰성이 문제될 수 있는데 CA의 공개키는 사용자들의 등록시에 각 사용자에게 스마트 카드나 TRM 등을 통해서 안전하게 미리 나누어 주는 방식을 쓴다면 이러한 문제는 해결된다.

그리고 대칭키 암호 알고리듬을 쓰게 되면 관련된 실체들의 identity에 대한 정보는 security sender / recipient field에 표시된다

또한 메세지 자체가 여러 실체들을 포함할 경우 (*e.g.*, 디지탈 다중 서명에서 서명에 참여하는 사람들), 여러 실체들이 관련된 서명이나 인증 실체들을 식별하기 위하여 관련 보안정보들을 반복 전송할 수 있다.

EDIFACT 메세지를 안전하게 처리하기 위한 필요조건과 기술적 사항들을 포함하는 서비스들은 아래와 같다.

- 메세지 순서 무결성(Message sequence integrity) : 메세지의 중복, 첨가, 삭제, 유실, 재전송 등을 막기 위한 것으로 유실된 메세지를 발견하기 위해서는 송신자가 메세지 순서 번호(Message sequence number)를 첨가하고, 수신자가 이를 확인하거나 송신자가 acknowledgement를 요구하고 이를 확인한다. 첨가된 메세지나 되풀이된 메세지를 찾기 위해서 송신자가 위에서 언급한 메세지 일련번호나 time stamp를 덧붙이고 수신자가 이를 확인하는 것도 가능하다. 완전한 보호를 위하여 time stamp나 일련번호 자체의 무결성을 송신자의 디지탈 서명 등을 써서 보장할 수 있어야 한다.
- 메세지 내용 무결성(Message content integrity) : 전송 도중의 메세지 변조(modification)를 막기 위한 것으로, 송신자가 공개키 또는 관용 암호 알고리듬을 사용하여 무결성 증명 값을 메세지에 덧붙여

서 송신하고 수신자는 메세지와 알고리듬, 매개변수 등을 이용하여 메세지 증명 값을 계산한 후에 이를 수신된 값과 비교하여 메세지의 변조 여부를 확인할 수 있다. 메세지 내용 증명은 메세지 발신처 인증(message origin authentication)이나 발신처 부인 봉쇄(non-repudiation of origin)의 한 부분으로 구현될 수도 있다.

- 메세지 발신처 인증(Message origin authentication) : 메세지를 실제로 송신한 사람이 나중에 메세지 발신처가 아니라고 주장하지 못하도록 하기 위한 것으로 메세지 발신자의 비밀키와 메세지 내용에 의존하는 메세지 인증 코드(MAC, Message Authentication Code)를 메세지와 함께 보냄으로써 가능해진다. 이것은 메세지 내용 무결성을 포함할 수 있으며, 발신처 부인 봉쇄(non-repudiation of origin)의 한 부분으로도 구현가능하다.
- 발신 부인 봉쇄(Non-repudiation of origin) : 송신자가 메세지를 보냈다는 사실을 부인하더라도 수신자는 이를 제 3 자에게 증명할 수 있도록 하는 것으로, 디지탈 서명을 메세지에 첨가함으로써 구현할 수 있다. 공개키 암호 알고리듬으로 디지탈 서명을 구현할 때, 서명의 확인은 서명 생성시에 사용된 비밀키에 대응되는 공개키로 할 수 있는데, 공개키는 미리 상대방과 공유하거나 CA의 인증서의 한 부분으로 포함시켜서 전체 송신 메세지의 일부분으로 보낼 수 있다. 디지탈 서명을 쓰면 메세지 내용 무결성과 발신처 인증을 동시에 구현할 수 있다.
- 수신 부인 봉쇄(Non-repudiation of receipt) : 수신자가 메세지를 받았다는 사실을 부인하지 못하도록 하는 것으로 메세지를 근거로 한 디지탈 서명을 포함한 acknowledgement를 수신자가 송신자에게 보냄으로써 구현할 수 있다. Acknowledgement는 수신자가 송신자에게 보내는 서비스 메세지의 형태를 가지게 된다.
- 메세지 기밀성(Confidentiality of content) : 메세지 내용을 허가받지 않고 보거나, 복사하거나, 노출하는 것을 방지하는 것으로 데이타를 암호화함으로써 구현할 수 있다. 그런데 실제로 빠른 속도의 암호화를 위하여 비밀키 암호화 알고리듬을 사용하는데 이 때는 비밀키를 안전하게 공유하는 방법이 문제가 된다. 그래서 공개키 암호 알고리듬을 써서 비밀키를 안전하게 공유하는 방안이 제시되기도 하였다[20].

위에서 언급한 것처럼 몇 가지 서비스는 그 특성상 다른 서비스를 포함하게 되는데 이에 대해서는 표 1에 나타나 있다[19].

implies: use of :	Content Integrity	Message Origin Authentication	Non-repudiation of Origin
Content Integrity	Y		
Message Origin Authentication	Y	Y	
Non-repudiation of Origin	Y	Y	Y

표 1 : EDIFACT Security Services interrelations

5.2 메세지 수준 Security

EDIFACT 메세지 수준의 Security 서비스는 메세지 자체에 포함될 수도 있고, 하나의 독립된 메세지로 만들어져서 메세지와 함께 전송될 수 있다. 기밀성(Confidentiality)을 제외한 모든 서비스는 message header(UNH)와 message trailer(UNT) 사이에서 security header와 trailer segment group을 포함시킴으로써 어떤 메세지에도 적용할 수 있는 형태로 구현될 수 있다. 여기서 message security header, trailer 쌍은 각 보안 서비스마다 필요하게 된다.

메세지의 전송에 관여하는 사람들은 적용된 보안 서비스와 사용된 메카니즘, 메카니즘의 응용범위 등을 알아야 하는데 이러한 정보들은 특별한 security segment의 형태로 전송된다.

메세지 보안 헤더(Message security header)의 목적은 메세지에 적용된 보안 방법을 명시하고 보안 서비스를 확인하기 위해 필요한 관련 데이타들을 포함하는 것이다. 상세한 보안 알고리듬이 포함될 수도 있고, 관련된 공개키 인증서가 들어 있을 수도 있다.

메세지 보안 트레일러(Message security trailer)는 관련된 메세지 보안 헤더에 명시된 security 기능들에 해당하는 security 결과값들을 포함하는데 사용된다.

Message security header와 message security trailer는 향후의 확장성을 위하여, 적용되는 보안 서비스마다 한 쌍씩 존재하도록 되어 있다. 그림 10에 EDIFACT 메세지에 Message security header와 message security trailer가 어떻게 위치하는가가 나타나 있다.

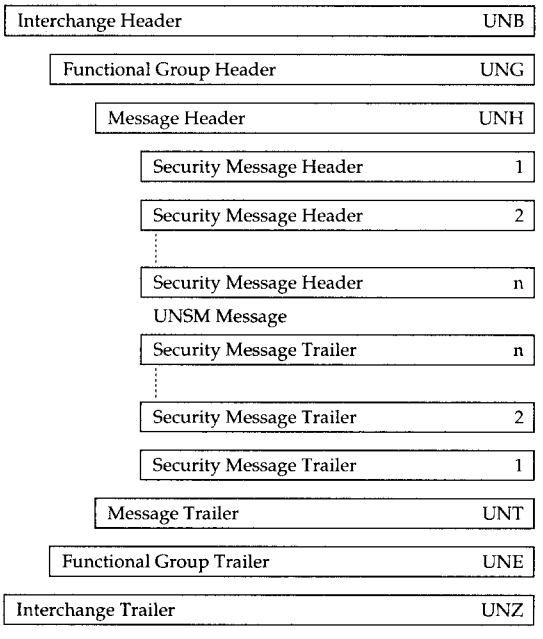


그림 10 : EDIFACT 메세지에서 message security header와 message security trailer의 위치

메세지 보안을 위한 또 한 가지 형태로 하나의 독립적인 형태의 메세지를 따로 만들어서 전송하는 방식이 있다. 이는 송신자 입장에서 볼 때 하나의 보안 메세지를 써서 하나 이상의 메세지에 대한 보안 서비스를 수행할 수 있게 하고, 원래 메세지를 돌려 주지 않고서 송신자에게 메세지를 제대로 받았다는 안전한 acknowledgement를 줄 수 있도록 하기 위해서이다. 실제적으로 FUNACK (functional acknowledgement message)를 써서 이러한 필요를 만족시킬 수 있다.

송신자의 입장에서 볼 때 FUNACK를 분리된 메세지로 보내면 기밀성 서비스를 제외하고 다른 모든 보안 서비스를 제공할 수 있다. 그러므로 보안 서비스는 나중에 또는 적당한 단계에 교환될 수 있다. 하나의 FUNACK가 부가적으로 몇 개의 원천 메세지를 포함할 수 있는데 이것은 하나의 보안 메세지를 써서 한 개의 메세지에 대한 보안 서비스를 수행하는 직접 결합 방식과는 다른 것이다. 단, 여러 메세지에 대해서 하나의 보안 메세지가 보안 서비스를 규정할 경우, 보안 메세지에는 보안 서비스가 원래 어느 메세지에 적용되는가에 대한 고유한 참조(unique reference)가 있어야 한다.

한편 수신자는 FUNACK를 써서 송신자가 요구하는 수신 부인 봉쇄에 대한 응답을 할 수 있다. 또한 FUNACK는 수신자가 송신자에게 보내는 안전성이 보장된 수령의 통지로 사용될 수 있는데, FUNACK가 만들어지는 방식과 기준에 의해서 송신자는 원하는 상대방이 메세지를 수신했다는 보장을 얻을 수 있다. 수신 부인 봉쇄를 제공하는 서비스 메세지는 여러 개의 디지탈 서명을 포함할 수 있는데 이것은 FUNACK가 하나 또는 그 이상의 메세지에 적용될 수 있기 때문이다. 그리고 FUNACK는 오직 메세지 수준에서만 믿을 수 있는 수신 통지를 제공한다.

6. X.435의 보안 서비스

본 절에서는 1990년에 EDI를 위해서 X.400

을 확장하여 CCITT에서 제정된 X.435에서 규정된 보안 서비스들을 EDI와 MHS에 공통인 것과 EDI에 추가된 것들로 나누어서 고찰하고 규정된 보안 서비스를 구현하기 위하여 어떠한 보안 메카니즘을 쓸 것인가에 대해서 설명하고자 한다.

6.1 EDI와 MHS에 공통인 보안 서비스

이 절에서는 X.435에서 규정하고 있는 보안서비스들 중에서 MHS 보안서비스와 공통인 서비스들, 즉 X.402와 X.411에서 정의하고 있는 보안 서비스들을 어떻게 구현할 것인가를 설명하고 구현한 보안 서비스들을 EDI 환경에 적합한 형태로 바꿔서 보내는 것을 설명하고자 한다[21, 22].

6.1.1 메세지 비밀 보장

이 서비스는 메세지 전체를 암호화하여 불법 침입자가 원래 메세지 내용을 알아 낼 수 없도록 하는 서비스로 X.411의 token 등을 이용하여 구현할 수 있다.

6.1.2 메세지 내용 무결성 검사

무결성 검사 서비스는 메세지 수신자가 메세지 내용이 전송 도중에 바뀌거나 삭제 또는 삽입되는 것을 확인할 수 있도록 하는 보안 서비스이다. 수신자의 입장에서 볼 때는 무결성 검사 서비스는 다음에 언급할 메세지 발신처 인증 서비스와 함께 제공되어야 한다[17].

메세지 무결성 검사 서비스는 message submission argument field에 있는 CIC(Content Integrity Check) field를 이용하거나, message token의 signed Data field 안에 있는 Content Integrity Check field를 이용하여 구현할 수 있다[17].

```
Content Integrity Check :: =
        SIGNATURE SEQUENCE {
        algorithm-identifier
        ContentIntegrityAlgorithmIdentifier,
        content Content }
```

6.1.3 메세지 발신처 인증

이 서비스는 메세지의 수신자를 비롯하여 메세지가 전달되는 MTA에게 메세지의 발신처를 인증하는 수단을 제공하는 서비스로 전체적으로 디지탈 서명 서비스(6.1.6 참조)로 구현할 수 있다.

메세지 발신처 인증 서비스는 앞에서 설명한 메세지 내용 무결성 검사 서비스와 하나가 다른 하나를 구현하는데 쓰일 수 있다는 점에 있어서는 비슷하지만, 메세지가 비밀보장 서비스를 위해서 암호화될 경우에 상당히 달라진다고 볼 수 있다. 전자는 암호화된 메세지 내용에 대해서 작용하므로 메세지 전송에 관련된 모든 component들(MTA 포함)이 메세지의 발신처를 확인할 수 있는 반면에 후자는 암호화되지 않은 평문(plain-text)에 대해서만 작용하므로 메세지가 평문의 형태로 나타나는 부분에서만 즉, end-to-end 목적으로만 사용될 수 있다고 할 수 있다. 또한 전자는 문제가 되는 송신자가 암호화된 메세지를 보냈다는 것만을 증명할 수 있지만 반드시 그 내용을 송신자가 알고 있다는 것을 증명하지는 못한다. 그러나, 후자는 송신자가 메세지 내용을 알고 있다는 점에서 차이점이 있다고 할 수 있다[14].

이 서비스는 Message submission argument의 MOAC(Message-Origin-Authentication-Check)를 이용하여 구현할 수 있는데, MOAC는 다음과 같은 요소들로 구성된다.

- message-origin-authentication-algorithm-identifier
- asymmetrically encrypted and hashed (message-origin-authentication-algorithm-identifier, message-content, content-identifier, message-security-label)

만약 비밀 보장 서비스가 함께 사용되면, 정해진 수신자 이외의 객체도 발신처 인증을 할 수 있도록 MOAC는 암호화된 메세지 내용에 의해서 계산된다. MOAC는 메세지 발신자가 자신의 secret-asymmetric-encryption-key를 써서 생성하고, 메세지의 수신자나 전송되는 MTA는 originator-certificate으로부터 발신자의 public-asymmetric-encryption-key(subject-public-key)를 추출하여 MOAC를 확인할 수 있다.

6.1.4 발신처 부인 봉쇄

이 서비스는 수신자가 자기 자신 뿐만 아니라, 제 3 자에게도 메세지의 발신처를 증명할 수 있는 수단을 제공한다는 점에서 메세지 발신처 인증보다 더 강력한 서비스라고 할 수 있다.

6.1.5 배달 증명

이 서비스는 수신자가 메세지를 읽을 수 있는 형태로 받았다는 사실을 증명하는 것으로 수신자의 UA가 복호화된 메세지에 서명하고 그 서명을 발신인에게 보냄으로써 구현할 수 있다. Delivery report의 EXTENSION field에 있는 proof of delivery field를 이용해서 이 서비스를 구현할 수 있는데 배달증명의 요청은 message의 proof-of-delivery request EXTENSION field를 써서 할 수 있다[22].

6.1.6 디지탈 서명 서비스

이 서비스는 메세지에 송신자의 비밀키로 메세지 전체 또는 메세지의 일부를 암호화한 값을 덧붙여서 수신자에게 보냄으로써 수신자는 메세지 자체의 무결성과 송신자 신분확인을 동시에 할 수 있도록 하는 서비스이다. 메세지에 대한 디지탈 서명을 이용하면 송신 부인봉쇄(non-repudiation of origin와 메세지 발신자 확인(message origin authentication)을 함께 구현할 수 있다.

6.2 EDI에 추가된 보안 서비스

F.435에서는 MHS 보안서비스에 추가하여 EDI에 필요한 보안 서비스들을 정의하고 있는데 추가된 항목을 정리하면 표 2와 같다[10].

EDIM responsibility authentication	Non-repudiation of EDIM responsibility service
• Proof of EDI notification	• Non-repudiation of EDI notification
• Proof of retrieval	• Non-repudiation of retrieval
• Proof of transfer	• Non-repudiation of transfer
	• Non-repudiation of content

표 2 : EDI 환경에 추가되는 보안 서비스

- EDIM Responsibility authentication service : EDIM 책임의 전송 과정을 확인함으로써 EDIM이 원하는 목적지로 제대로 전달되었는가에 대한 확인을 제공하는 서비스들이다.
  - — Proof of EDI Notification : 메세지의 송신자에게 메세지가 전달되었음을 확인하는 증명을 제공하는 서비스이다. 이 서비스는 EDI Notification(EDIN) 영역에 설정된 message submission에 대한 Content Integrity Check를 이용함으로써 구현 가능하다.
  - — Proof of retrieval : 특별한 메세지가 EDI-UA에 의해서 EDI-MS로부터 retrieval 되었음을 확인할 수 있는 증명을 MS(Message Store) 관리자에게 제공하는 서비스이다.
  - — Proof of transfer : 이 서비스는 MTA나 MD 사이에서 메세지가 전달되었음을 확인하는 증명을 제공하는 서비스이다.

- Non-repudiation of EDIM Responsibility services : 이것은 proof 서비스보다 더 강력한 형태로 서비스를 제공받는 측은 자기 자신이 서비스의 제공 자체를 확인할 수 있을 뿐만 아니라 제 3 자에게 서비스를 제공받은 사실 또한 증명할 수 있다[14].
  - -- Non-repudiation of EDI notification : 특별한 메세지가 EDI-UA에 의해서 EDIM Responsibility의 수용(accept), 회송(forward), 거부(refuse) 등에 대한 부인할 수 없는 증거를 제공하는 서비스이다.
  - — Non-repudiation of retrieval : 특별한 메세지가 EDI-UA에 의해서 EDI-MS로 복귀(retrieval)되었음을 부인할 수 없는 증거를 MS 관리자나 EDI-UA에게 제공하는 서비스이다.
  - — Non-repudiation of transfer : 한쪽 MTA나 MD에서 다른 MTA나 MD로 메세지가 전달되었음을 부인할 수 없는 증거를 제공하는 서비스이다.
  - — Non-repudiation of content : 이 서비스는 메세지 내용의 인증과 무결성을 보장하는 부인할 수 없는 증거를 EDIMG (EDI Messaging) 사용자에게 제공한다.

위의 보안 서비스들이 실현되는 과정을 나타내면 그림 11과 같다.

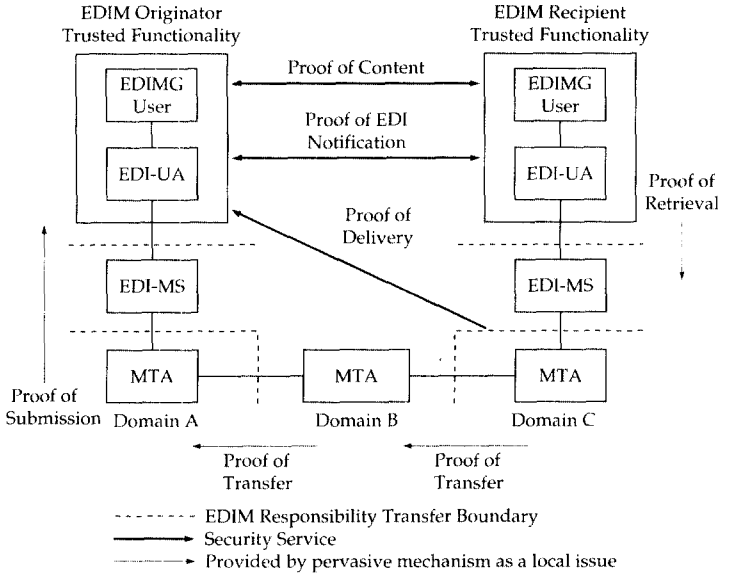


그림 11 : EDIM Responsibility Transfer

7. 결 론

현대의 발달하는 컴퓨터 기술과 이에 걸맞는 통신기술의 급속한 성장으로 인하여 기업간의 자료교환은 자료의 작성, 변환, 전송, 수신 등을 이전과 같은 수동식 작성, 전달이 아닌 통신 수단과 결합한 컴퓨터에 의해서 자동적으로 처리하는 EDI 시스템의 구축이라는 방향으로 진행되고 있다. 한편 거래의 범위가 점차로 넓어짐에 따라서 EDI 시스템을 구축하고 운영하는데 있어서 이기종 또는 서로 다른 컴퓨터 네트워크 간에도 거래 정보

를 안전하게 교환할 수 있도록 하는 것이 필수적인 조건이 되었고 이것은 네트워크상의 정보 보안의 취약성으로 해서 법적인 증명으로서 쓰이는 EDI 문서의 안전성을 저해하는 요인이 되어 왔다. 본 고에서는 국제적으로 인정되는 표준에 근거한 EDI 보안 시스템 모델을 제안하기 위하여 우선 안전한 EDI 시스템을 구성하기 위하여 고려해야 할 국제 표준들을 살펴보고, EDI에 고유한 서비스가 규정된 F.435와 EDI 암호 시스템으로 공개키 암호 시스템을 사용할 때 공개키 획득에 필요한 X.500을 설명하였다. 또한 EDI 보안 서비스 시스템이 기본적으로 제공해야 할 보안 서비스들을 규정하기 위하여 EDIFACT에서 권고하는 메세지 수준의 보안 서비스와 X.435에서 정한 보안 서비스들을 고찰하였다.

본 고에서는 지금까지 부분적으로 제시되었던 EDI 보안 서비스를 EDI 문서 작성 단계에서 필요한 문서표준과, 작성된 EDI 문서를 통신망을 통하여 거래 상대방에게 보내기 위해서 필요한 통신표준의 두 가지로 나누어 고찰하고 제시하였다.

참 고 문 헌

[1] American National Standards Institute ASC X12,"Electronic data interchange, Standards X12.1 to X12.22,"1986.

[2] ISO IS 9735,"Electronic data interchange for administration, commerce and transport(EDIFACT) - Application level syntax rules," 1988.

[3] CCITT Recommendation X.400, "Message Handling Systems : System Model - Service Elements," 1988.

[4] CCITT Recommendation X.435, "Message Handling Systems : EDI Messaging System," 1990.

[5] 정 진욱, "컴퓨터 네트워크에서의 정보보호 프로토콜에 관한 연구 - EDI에서의 정보보호 프로토콜 설계 및 구현을 중심으로," 한국통신정보보호학회 논문지, 제 3 권, 제 1 호, pp. 58-68, 1993.

[6] 조 광문, 김 태윤, "EDI 보안 시스템과 디지탈 서명," 한국통신정보보호학회 학술지, 제 3권, 제 1 호, pp. 14-25, 1993.

[7] 김정희, 김태윤, "전자식 문서 교환 (EDI)의 보안과 통제관리," 통신정보보호학회지 제 1 권, 제 3 호, pp.99-108, 1991.

[8] 임용진, 이강무, 고홍기, 나종근, 김동규, "EDI 시스템의 안전성 서비스 구현 방안에 관한 연구," 통신정보보호학회 학술발표논문집, pp.153-164, 1992.

[9] Genilloud G., "X.400 MHS: First Steps Towards an EDI Communication Standard," Computer Communication Review, vol.20, pp. 72-86, April, 1990.

[10] CCITT Recommendation F.435, "Message Handling: EDI Messaging Service,"1990.

[11] Jim Craigie, "ISO 10021 - X.400 (88) : A Tutorial for Those Familiar with X.400 (84), "Computer Networks and ISDN Systems vol.16, pp. 153-160, 1988/89.

[12] Sara Radicati,Electronic Mail: An Introduction to X.400 Message Handling Stadards, McGraw-Hill, 1992.

[13] CCITT Recommendation X.500, "The Directory - Overview of Concepts, Models and Services,"1988.

[14] B. Plattner et al., X400 message handling : standards, interworking, applications,Addison-Wesley, 1991.

[15] CCITT Recommendation X.509, "The Directory - Authentication Framework,"1988.

[16] CCITT Recommendation X.501, "The Directory - Models," 1988.

[17] C. Mitchell, M. Walker and D. Rush, "CCITT/ISO Standards for Secure Message Handling," IEEE J. of Sel. Areas in Communication, vol.7, no.4, pp. 517-524, May, 1989.

[18] I. Damgard, "Collision free hash functions and public key signature schemes," Proceedings of Euro Crypto'87, 1987, pp.203-216.

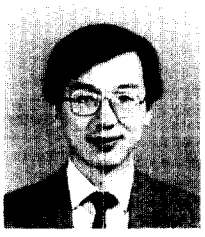
[19] UN/ECE, "Recommendations for the UN/EDIFACT Message Level Security," 1993.

[20] W. Diffie and M. E. Hellman, "New Directions in cryptography," IEEE Tran. on Inform. Theory, IT-22, 6, pp. 644-654, 1976.

[21] CCITT Recommendation X.402, "Message Handling Systems : Overall Architecture,"1988.

[22] CCITT Recommendation X.411, "Message Handling Systems : Message Transfer System : Abstract Syntax Definition," 1988.

□ 著者紹介



이 필 중(李 弼 中) 종신회원, 국제이사

1951년 12월 30일생
1974년 2월 서울대학교 전자공학과 학사
1977년 2월 서울대학교 전자공학과 석사
1982년 6월 U.C.L.A. System Science, Engineer
1985년 6월 U.C.L.A. Electrical Engineering, Ph.D.
1980년 6월 ~ 1985년 8월 Jet Propulsion Laboratory, Senior Engineer
1985년 8월 ~ 1990년 2월 Bell Communications Research, M.T.S.
1990년 2월 ~ 현재 포항공과대학 전자전기공학과, 부교수

전 윤 호

약력 미 접수