안전한 EDIFACT 시스템 구현 소요 기법 연구

안금혁°*, 장청룡*

* 한국통신 연구개발원

A Study for Mechanisms in Secure EDIFACT System

Ahn Keum-Hyug, Jang Chung-Ryong

Korea Telecom Research Laboratories

1. 서론

컴퓨터 기술의 빠른 발전에 따라 사회의 각 분야에서는 업무에 필요한 정보를 컴퓨터시스템에 저장하고 이를 이용하게 되었다. 이런 컴퓨터의 기술 발전은 통신 기술의 발전과 맞물려 신속함과 정확성을 필요로 하는 곳에서 컴퓨터를 이용한 빠른 업무 처리를 요구하게 되었고, 특히 경제적인 이유로 기업체에서의 요구가 매우 크다. 일반 기업체나 관공서에서 많은 양의 서류를 작성하여 신속하고 정확하게 주고받는데 있어, 거래 문서나 거래 정보 등은 정형화된 자료이며 이를 규격화된 양식에 의해 전자식으로 상호 교환하는 EDI(electronic data interchange)가 대두되었다.

이에 따라 미국 및 유럽을 중심으로 문서 표준인 ANSI X12 및 GTDI(guidelines for trade data interchange)가 등장했다. 이를 통합한 것이 UN에서 1987년에 ISO와 유럽 경제 위원회 등의 공동 작업으로 EDIFACT(EDI for administration, commerce and transport)가 마련되어 ISO 9735로 표준화되었다[1]. 한편 ITU-T에서는 만들어진 EDI 문서를 이기종 시스템을 사용하는 상대방과 자유롭게 송수신할 수 있도록 통신 표준인 X.400 MHS에 기반을 둔 X.435를 권고하고 있다[2].

본 고에서는 중간 또는 작은 규모에서 적용 가능하며, 주로 PC를 사용하고 있는 환경에서 문서 표준인 EDIFACT 시스템에 관련된 사항만을 다루고 통신 표준인 X.400 시리즈에 관련된 사항은 처리 규모나 구현 복잡성 등으로 제외하였다. 더우기 EDIFACT 시스템에서의 통신은 널리 사용하고 있는 TCP/IP 등을 사용하여 메시지를 전달하는 것이 일반적이다.

2장에서는 EDI 시스템에서의 위협 및 요구사항과 대처 방안을 소개하고, 3장에서 전자서명 등 안전한 EDIFACT 구현에 필요한 소요 기법과 4장에서 이를 이용한 구현 모델을 제안하였다.

2. 위협 및 요구사항과 대처 방안

공중통신망 또는 EDI 전용망을 통하여 메시지를 교환할 때 다음과 같은 위협이 있을 수 있다.

- 메시지가 비인가된 3자에 의해 가로채어 수정될 수 있다.
- 메시지가 분실되거나 재사용될 수 있다.
- 메시지가 비인가된 제3자에 의해 읽혀질 수 있다.
- 비인가된 제3자가 송신자 또는 수신자로 위장할 수 있다.
- 송신자 또는 수신자가 특정 메시지에 대하여 송신 또는 수신 행위를 부인할 수 있다.

또한 기존의 종이 문서에 수기 서명을 하여 전달할 때 서명 위조나 문서의 내용 변조가 어려운 반면 통신망을 통한 EDI 문서 교환에는 서명자의 확인이 어렵고 문서가 복사되거나, 내용 위조가 상대적으로 쉬운 단점이 있다. 따라서 통신망을 통한 문서 교환도 기존의 종이 문서 처리

와 같은 효과를 얻을 수 있는 방법이 필요하다. 위와 같은 위협 및 요구사항을 만족하는 대처 방안은 다음과 같다.

- 전자서명 (digital signature) : 공개키 암호계 기법을 이용함으로써 메시지와 서명자만이 알고 있는 비밀키로 서명을 생성하여 전달하면 수신자는 송신자의 공개키로 서명을 검증하여 송신자의 확인과 메시지의 변경 유무를 확인할 수 있다. 또한 송신자만이 알고 있는 비밀키로 서명을 생성할 수 있으므로 송신 사실을 부인할 수 없다.
- 메시지 인증 코드 (MAC : message authentication code) : 메시지 인증 코드는 메시지로부터 직접 계산되며, 전자서명과는 달리 대칭키를 사용한다. 이 방법은 송신자의 확인 및 메시지 변경 유무를 확인할 수 있다. 그러나 송수신자간의 동일한 공유키를 사용하므로 부인 봉쇄 서비스는 불가능하다.
- 순차 번호 (sequence number) : 수신자는 어떤 메시지가 분실되거나 재사용 되었는지 확인하는데 사용된다. 이 순차 번호 자체는 위 두 가지 방법과 함께 사용함이 바람직하다.
- 암호화 (encryption) : 메시지에 대한 기밀성을 요구하는 응용에서는 메시지를 암호화하여 전달하면 수신자는 암호화된 메시지를 복호화하여 원래의 메시지를 얻는다.

<표 1> EDIFACT에서의 위협 및 안전성 대응 기법

위협 \ 대응기법	전자서명	MAC	순차 번호	암호화
메시지 가로채고 변경	o 또는 o			o
메시지 분실 또는 재사용	o 또는 o		+ o	
메시지 노출				o
제3자의 가장	o 또는 o			
송수신 행위 부인	o			

<표 1>에서와 같이 위협 요인에 대한 대처 방안의 일환으로 전자서명 또는 MAC을 사용할 수 있는데, 전자서명 기법은 메시지 변경 유무 확인과 송신자 확인 뿐만 아니라 송수신 행위의 부인을 방지하므로 본 고에서는 MAC보다 기능이 많은 전자서명을 선택하였다.

또한 EDIFACT 시스템 구현시 메시지 분실 또는 재사용 방지를 위하여 순차 번호를 도입하여야 하며, 메시지의 기밀성 유지를 위한 응용에서는 암호화 알고리즘을 추가 구현하여야 한다.

3. 안전한 EDIFACT 구현에 필요한 소요 기법

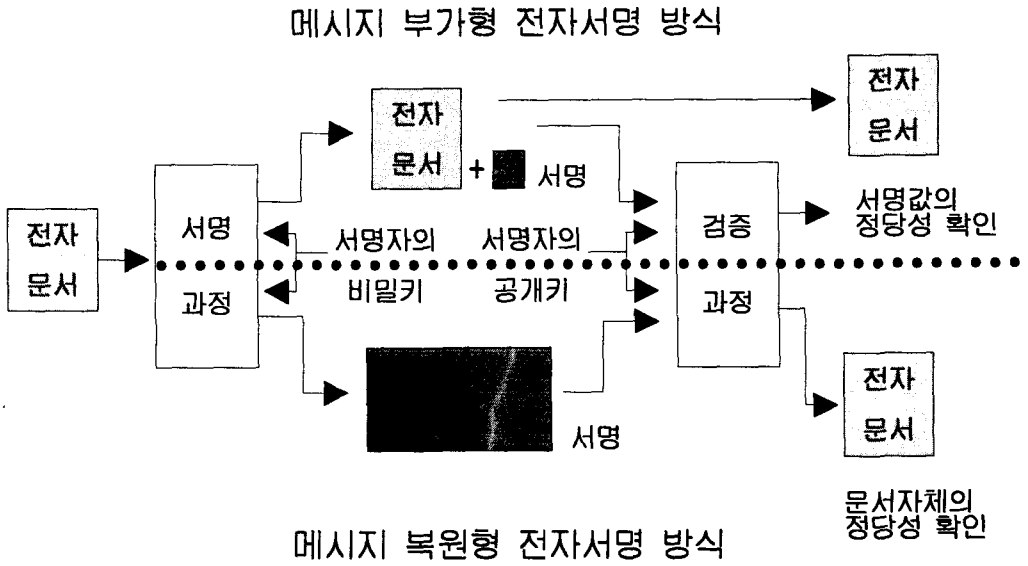
EDIFACT 문서 교환 서비스를 제공하기 위하여 문서에 대한 서명자의 서명 생성과 서명 검증으로 문서의 발신처 확인 및 문서의 변경 유무 확인 등에 전자서명 메카니즘이 필요하다. 전자서명 메카니즘은 공개키 암호시스템에 바탕을 두었다. 즉 서명자만이 알고 있는 비밀키로 서명을 생성하여 검증자에게 전달하면 서명자의 비밀키에 대응하는 서명자의 공개키를 획득할 수 있는 검증자는 수신한 서명을 검증함으로써 서명자의 확인 및 서명된 문서의 변경 유무를 확인할 수 있다. 서명자의 비밀키는 유일하며 서명자만이 갖고 있으므로 송신 사실을 부인할 수 없다. 이러한 환경에서 검증자는 서명자의 공개키를 획득할 수 있어야 하고 획득한 공개키가 진정한 서명자의 공개키 여부를 확인할 수 있어야 한다. 이에 따라 서명자 및 검증자가 신뢰할 수 있는 제3자 (TTP: trusted third party)는 각 사용자의 공개키를 자신의 비밀키로 서명한 사용자 확증서를 발급하여

이를 안전하게 관리하기 위한 확증서 디렉토리가 필요하다. 또한 서명자는 메시지의 서명을 생성하기 위하여 자신만이 간직하고 있는 비밀키를 사용한다. 그러나 비밀키는 암기하기 어려운 문자로 구성되어 있고 길이가 크기 때문에(140비트 이상 사용) 이를 저장하기 위한 사용자 토큰이 필요하다. 또한 서명자 및 검증자 간의 분쟁시 또는 안전한 시스템의 관리를 위하여 모든 처리 행위를 기록으로 남기는 감사 증적이 필요하며 이들을 다음절에서 각각 설명한다.

가. 전자서명

전자서명 기법은 데이터의 발신처를 확인하는 인증 서비스, 데이터의 변경 유무를 확인하는 무결성 서비스와 데이터의 송수신 행위의 부인을 방지하는 부인봉쇄 서비스를 제공한다. 전자서명 기법은 공개키 암호시스템에 기초로 한다. 따라서 서명자의 비밀키와 이에 대응하는 서명자의 공개키를 사용한다. 서명 생성 과정은 서명자의 비밀키를 사용하여 메시지에 대한 서명을 생성하고, 서명 검증 과정은 서명자의 공개키로 서명을 검증한다.

전자서명은 크게 2가지 형태로 분류할 수 있는데, 메시지 복원형은 메시지에 서명자의 비밀키를 사용하여 서명 메시지를 생성하며, 서명 검증은 서명자의 공개키를 사용하여 서명 메시지로부터 원래의 메시지로 복원한 뒤 메시지가 의미 있는 것인지 확인함으로써 검증하게 된다. 한편 메시지 부가형은 메시지에 서명자의 비밀키를 사용하여 서명을 생성하여 메시지에 부가하고 서명 검증시에는 부가된 서명으로부터 서명자의 공개키를 사용하여 서명값을 확인하여 메시지의 정당성을 검증한다. (그림 1)은 메시지 복원형과 인증자 부가형 전자서명을 도식화시킨 것이다.



(그림 1) 전자서명의 형태상 분류

메시지 복원형 전자서명 방식은 제한된 길이(512비트 이하)의 메시지에 대해서만 서명 가능하며 서명의 크기가 크게 증가하는 문제점이 있다. 이러한 문제점들을 해결하기 위하여 부가형 전자서명 기법이 개발되었으며, 이는 다시 TTP가 확증서 발급 및 분배에 따라 ID-기반 기법 및 확증서(certificate)-기반 기법으로 분리한다[3].

부가형 전자서명 방법은 소인수분해 문제, 이산대수 문제 및 냅색(knapsack) 문제 등의 계산적으로 해를 구하기 어려운 문제에 기초한다. 소인수분해 문제에 근거한 전자서명 기법은 대표적으로 RSA 방법이 있고[4], 이산대수 문제에 근거한 전자서명 방법은 미국의 DSS[5] 등이 있다. 부가형 전자서명은 서명하기 전에 메시지 축약 함수(hash function)로 작은 길이의 해쉬코드를 생성하는데 해쉬 함수도 블럭암호시스템을 이용한 해쉬함수[6] 및 RIPEMD, SHA 등 전용해쉬 함수[7] 등이된다.

다음 표는 대표적인 암호화 알고리즘인 DES, 해쉬 함수로 MD5와 RSA를 사용한 서명 및 검증 시간을 각 전산환경에서의 처리 시간을 정리한 것이다[11].

<표 2> 안전한 EDIFACT 소요 기법에 대한 성능 비교

Machine	Speed	DES Encryption (kilobits/sec.)	Signature Verification RSA 17-bit key(kilobits/sec.)	Digital Signature RSA 512-bit key(kilobits/sec.)	Hashing MD5 or ISO 10118-2 (kilobits/sec.)
μVAX2	0.9 MIPS	75	0.45	3.95	825
VAX	2.4 MIPS	n.a.	0.20	1.68	n.a.
SUN 3/160	16.6 MHz	n.a.	0.21	2.21	n.a.
PC386	25 MHz	177	0.06	0.39	2.084
PC486	33 MHz	442	0.028	0.18	5420

나. TTP(trusted third party)

송신자는 자신의 비밀키로 메시지에 서명을 생성하여 수신자에게 전달하면 수신자는 송신자의 공개키를 사용하여 서명을 검증한다. 그러나 검증에 사용할 공개키가 서명자의 진정한 공개키인지 확신이 필요하다. 그러므로 안전한 EDI 시스템을 구축하기 위하여 신뢰성 있는 제3자가 필요하며, 이는 EDI 시스템에 직접 또는 간접적으로 참여한다. 또한 TTP는 EDI 시스템 운용 환경에 따라 하나일 수도 있고 여럿일 수도 있다. TTP의 주요 임무는 사용자 등록, 확증서 발급 및 분배이며 어떤 환경에서는 키 생성도 포함한다. ITU-T에서는 분산 환경에서 확증서 생성, 분배 및 획득 방법에 대해 X.509 디렉토리 인증서비스 프레임워크로 표준화하였으며 사용자 등록, 확증서 발급 및 분배는 다음과 같다[10].

(1) 안전한 사용자 등록

EDI 시스템에서는 먼저 안전한 사용자 등록이 필요하며, 이것은 메시지 전달시 정확한 사용자 식별을 허용한다. 사용자 등록 내용은 구별 가능한 사용자의 이름과 주소 및 개인의 특성인 추가 정보로 구성된다.

(2) 확증서 발급

전자서명 관점에서 TTP의 가장 중요한 임무중 하나는 각 사용자의 공개키에 대한 확증서를 발급하는 것이다. 그러므로 TTP를 CA(certification authority)라고 하기도 한다. 사용자의 공개키는 서명을 검증하는데 필요하며, 사용자의 공개키가 진정한 것인지 확신하여야 한다. 따라서 확증서는 신뢰성 있는 사용자의 공개키, 확증서의 무결성 등이 보장되어야 하므로 TTP에 의해 서명을 생성하여 발급한다. 또한 TTP의 공개키는 모든 사용자가 분명하며 유일한 방법에 의

해 획득 가능하여야 한다. 확증서에는 사용자의 식별자와 공개키 외에 선택적으로 공개키의 유효기간, 사용된 서명 알고리즘 식별자 등 추가 정보가 포함될 수 있다.

(3) 확증서 분배

확증서는 EDI 시스템의 모든 사용자가 접근할 수 있는 시스템 내에 정당한 모든 사용자의 공개키를 포함하여야 한다. 또한 확증서의 디렉토리 관리도 보증하여야 한다. 주요 관리 기능은 유효기간, 블랙 리스트 관리 및 확증서의 갱신 등이다.

다. 사용자 토큰

사용자의 비밀키를 저장하기 위한 사용자 토큰에는 사용 환경의 안전도 요구에 따라 다음의 3가지가 사용될 수 있다.

(1) 스마트카드

사용자의 비밀키와 서명자의 확증서로부터 서명자의 공개키를 획득하기 위한 TTP의 공개키는 스마트카드의 안전한 영역에 저장되어 있어야 하며, 사용자의 접근을 통제하기 위하여 스마트카드에 부여된 PIN(personal identification number) 값을 사용하여 분실 및 도난에 대처할 수 있다.

(2) 개인 플로피 디스켓

사용자의 비밀키는 디스켓에 부여된 PIN과 유사한 형태의 패스워드(password)에 의해 암호화하여 저장되며, 스마트카드처럼 사용자의 접근을 통제하기 위하여 디스켓에 패스워드를 부여하여 사용하는데, 디스켓은 copy가 용이하기 때문에 물리적으로 안전한 곳에 디스켓을 보관하여야 하며 틀린 패스워드가 여러번 시도될 때 디스켓의 내용을 자동 파괴하는 등 부가적인 기능이 추가되어야 한다.

(3) 하드디스크

사용자의 비밀키는 호스트의 마스터키로 암호화하여 저장하여야 하며, 호스트 시스템의 접근 제어 또는 물리적으로 통제된 호스트 시스템 내에서 사용되어야 한다.

라. 감사 증적(audit trail)

시스템 사용에서 안전성에 관한 사건이 발생했을 때, 참조하기 위하여 시스템에서는 자동적으로 다음과 같은 정보를 기록으로 남겨야 한다.

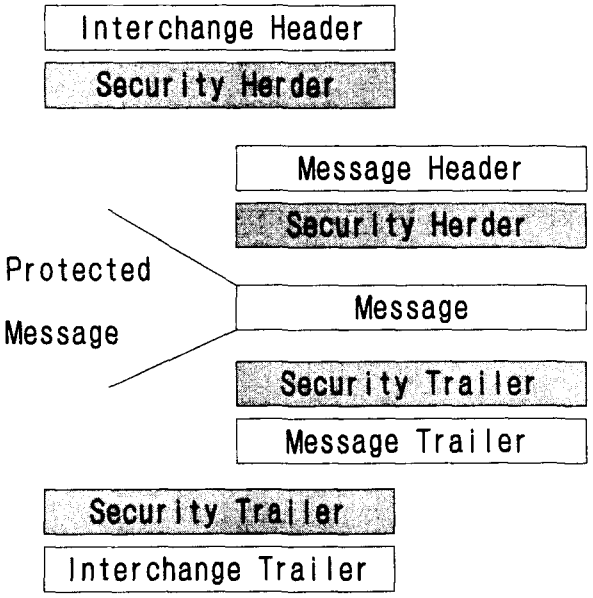
- 오류(error)가 발생했을 때 연산의 종류
- 오류 메시지
- 오류가 발생한 함수의 이름
- 오류가 발생한 날자 및 시간
- 서명 또는 검증자
- 오류 발생시 전달된 메시지
- 오류가 검출된 후에 메시지에 가해진 처리 내용 등

4. 안전한 EDIFACT 구현 모델

가. 안전한 EDIFACT 메시지 구조

EDIFACT 구문은 데이터의 교환을 구조화하기 위하여 사용되는 원소로 정의된다. 하나 또는 그 이상의 메시지 각각은 특정 transaction에 관련되며, 이는 하나의 interchange로 그룹화 된다. interchange는 EDIFACT 메시지의 가장자리에 해당되며, 헤더(header)에는 송수신자의 주소 등이

포함되며, 꼬리(trailer)에는 interchange 내에 있는 메시지의 합계 등이 포함된다. 메시지는 세그먼트로 구성되고 이는 데이터 원소로 구성된다. 데이터 원소는 데이터 원소의 식별자에 의해 가리키는 세그먼트내의 위치에 의해 식별되고 가변의 길이이다. 세그먼트는 자신의 세그먼트 식별자를 갖고 처음에 3 문자로 식별되며 반복 사용될 수 있다. (그림 2)는 안전한 EDIFACT 메시지의 형식이다.



(그림 2) 안전한 EDIFACT 메시지 형식

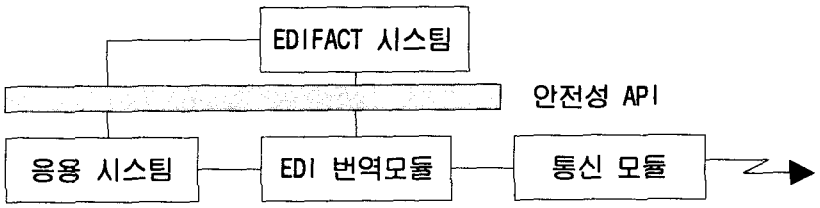
다음은 안전성 헤더 및 꼬리에 포함될 내용이다.

- 헤더에 포함될 내용
  . 인증, 무결성 및 부인봉쇄 등 제공될 안전성 기능의 유형
  . 수신자 부인봉쇄 방지를 위한 안전성 증명(acknowledgement)이 필요한지 유무
  . 송수신자의 식별 정보
  . 순차 번호 또는 timestamp
  . 사용된 해쉬 함수, 전자서명 및 암호화 알고리즘 식별자
  . 꼬리에 상응하는 헤더 수 등
- 꼬리에 포함될 정보
  . 전자서명의 서명 값 등 안전성 계산 결과
  . 헤더에 상응하는 꼬리 수 등

나. 시스템 구성 및 키 관리

안전한 EDIFACT 시스템은 기존의 EDIFACT 시스템에 세그먼트와 데이터 원소의 헤더 및 꼬리를 추가하여 안전성을 제공하는 것이다. 이를 위해 전자서명을 사용한 서명값 계산, 필요시 메시지 암호화, 사용자의 비밀키 및 공개키 생성 등을 통하여 안전한 시스템이 운용되어야 한다. 이러한 안전한 시스템은 보통 안전성 응용 프로그램 인터페이스(API)로 제공되며 기존 시스템의

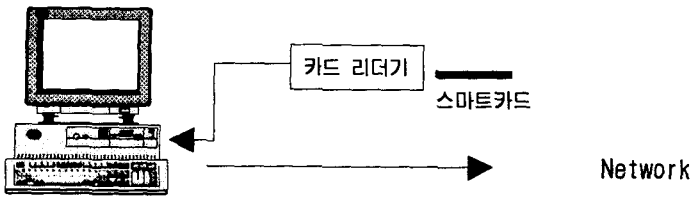
응용 프로그램에서 쉽게 적용된다. (그림 3)은 EDI시스템에 안전성 API를 통합한 것으로 영국 및 네덜란드 등에서 사용하고 있으며 이를 따른 API의 표준화 형태에 대한 지침이 마련되고 있다 [9].



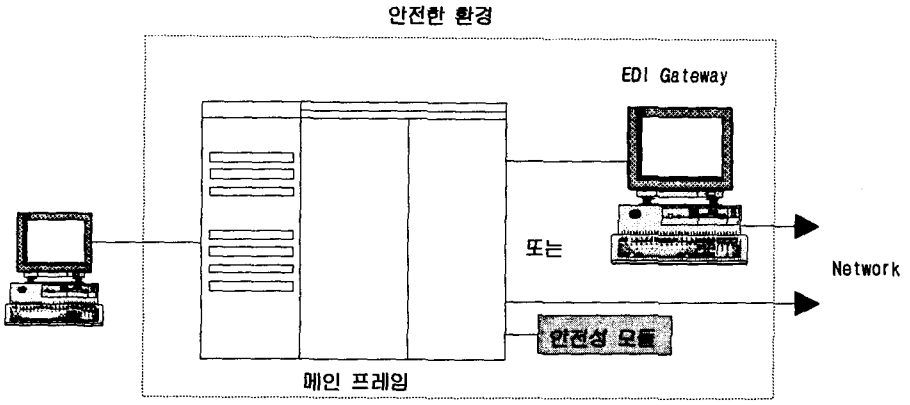
(그림 3) 안전한 EDIFACT 시스템의 구현 형태

또한 각 사용자의 키를 생성하고 분배하기 위하여 TTP가 필요하며, 시스템 규모나 응용에 따라 다양하게 구성될 수 있다. 중간 또는 작은 규모의 조직에서는 PC에서 데이터를 입력하고 조회할 것이다. 따라서 자신의 PC에서 스마트카드 등 사용자 토큰을 사용하여 transaction에 대한 서명을 생성하고 이를 EDI 통신망을 통하여 전달하면, 수신자측에서는 전달받은 서명값을 송신자의 공개키로 검증한 후 타당하면 접수한다.

한편 큰 규모의 조직에서는 메인프레임 컴퓨터를 사용하고 있으므로 시스템 자체의 안전성 환경을 구축하여 운용한다. 이러한 환경에서 EDI시스템에 안전성을 부가하기 위하여 작은 규모의 시스템에서처럼 PC를 통하여 메인프레임에 접속하여 데이터를 다운로드 받아 서명 생성 또는 검증하며 또다른 방법으로는 메인프레임에서 자동적으로 안전성 모듈에 의해 처리될 수 있다. (그림 4)는 시스템 규모에 따라 EDI 시스템의 구성을 도식화한 것이다.



(가) 작은 규모의 안전한 EDI 구성



(나) 큰 규모의 안전한 EDI 구성

(그림 4) 시스템 규모에 따른 안전한 EDI 시스템 구성

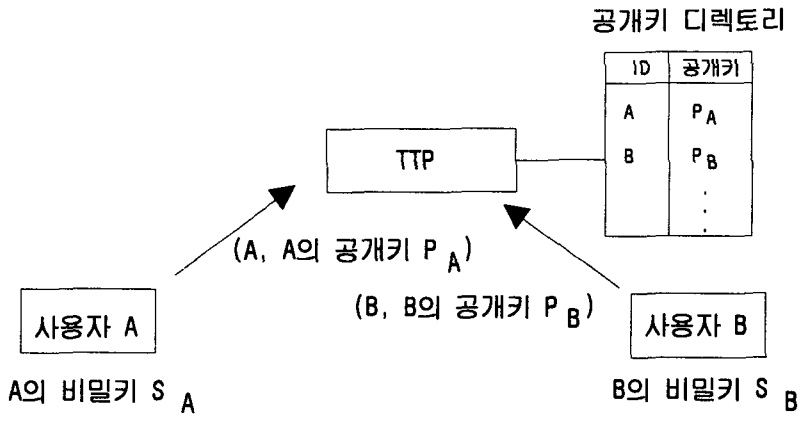
다. 안전한 EDIFACT 구현 모델

EDIFACT 시스템 구성은 시스템 규모, 망구조 및 응용 프로그램에 따라 여러 형태의 구성이 가능하며 TTP가 하나로 구성된 중앙 집중식과 여러개의 TTP로 구성된 분산 형태가 있을 수 있으며, TTP의 운용도 on-line과 off-line으로 구성할 수 있다.

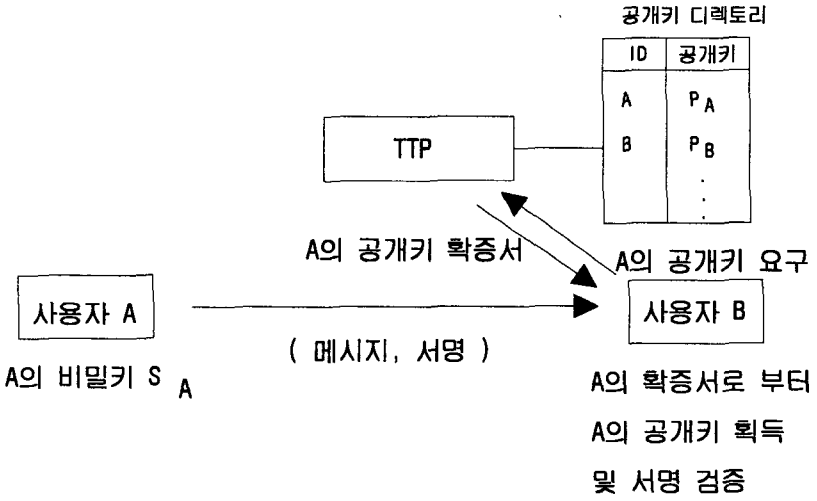
EDIFACT 시스템은 키 관리시스템, TTP와 메시지 서명 및 검증을 수행하는 모듈로 구성된다. 먼저 키 관리시스템에서는 각 사용자에게 비밀키를 사용자 토큰에 발급하여 주고, 공개키는 TTP의 확인서를 발급 받아 TTP에서 공개키 디렉토리를 관리하는 것으로 한다.

그런후 EDI 응용 프로그램에서 작성한 EDIFACT 형태의 전송 가능한 문서는 EDIFACT 시스템에 입력된다. 그러면 각 사용자에게 발급된 비밀키를 사용하여 서명을 생성한 후 서명과 함께 문서를 수신자에게 전달한다. 수신자는 전달받은 문서의 무결성, 송신자의 확인 등을 위하여 서명을 검증한다. 서명 검증은 TTP에게 확증서를 요구하여 전달받은 후 TTP의 공개키를 사용하여 확증서 내의 송신자 공개키를 검증하여 획득한 후 이를 이용하여 문서의 검증을 수행한다. 검증이 확인되면 수신 문서를 접수하고 그렇지 않으면 폐기한다.

기존의 X.509에서 권고한 on-line TTP의 확증서 발행에 의한 서명 처리 과정은 (그림 5)와 같다.



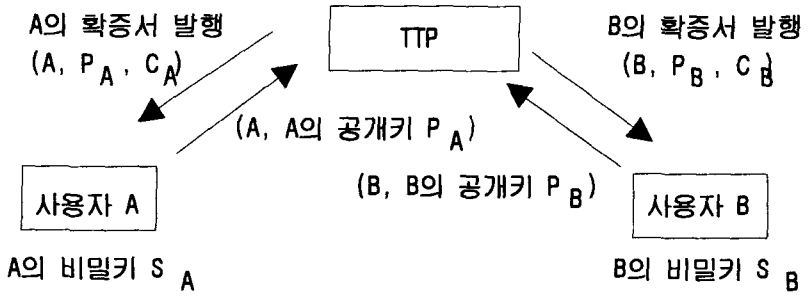
(가) TTP의 확증서 발행 과정



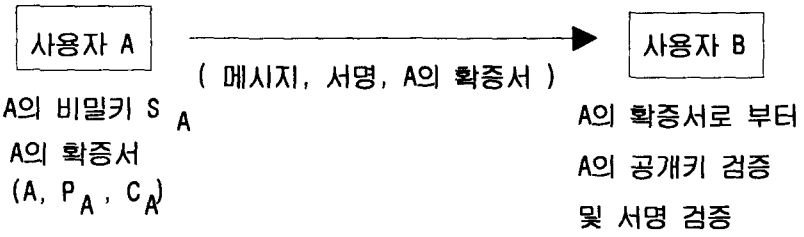
(나) 서명 생성 및 검증 과정

(그림 5) On-line TTP의 확증서 발행에 의한 서명 처리 과정

이와 같은 on-line TTP 구성 방식은 전달받은 메시지를 검증할 때마다 서명자의 공개키 확증서를 TTP에게 요구하여야 하므로 항상 접속 가능한 상태를 유지해야 하며 TTP와의 통신 시간이 추가로 소요된다. 중간 또는 작은 규모에서 주로 PC를 사용하는 환경에서의 EDI 시스템 구현을 위해 off-line TTP 구성 방식을 채택함으로 서명 검증시마다 확증서를 얻는데 요구하는 통신 시간을 절약함으로써 신속히 서명 검증을 처리할 수 있다. (그림 6)은 off-line TTP 구성의 확증서 발행에 의한 서명 처리 과정을 나타낸 것이다.



(가) TTP의 확증서 발행 과정



(나) 서명 생성 및 검증 과정

(그림 6) Off-line TTP 구성과 확증서 발행 및 서명, 검증 과정

따라서 적용 시스템의 특성과 시스템의 규모에 따라 TTP가 하나로 구성된 중앙 집중식과 여러개의 TTP로 구성된 분산 형태가 있을 수 있으며, TTP의 운용도 on-line과 off-line으로 구성할 수 있다. TTP의 on-line 및 off-line의 특성은 <표 3>과 같다.

<표 3> on-line 및 off-line TTP 구성의 특징

TTP 구성 / 특징	안전성 측면	통신량(빈도)	장 점
on-line TTP	안전함 (TTP는 각 사용자의 비밀키를 알지 못함)	- 서명 검증시 TTP에게 확증서 요구	- 공개키 디렉토리 변경시 용이 - 분산 디렉토리 관리가 가능
off-line TTP	안전함 (TTP는 각 사용자의 비밀키를 알지 못함)	- 서명 검증시 TTP에게 확증서 요구 안함	- 구현이 용이함 - 확증서 획득 시간이 필요 없으므로 서명 검증이 빠름

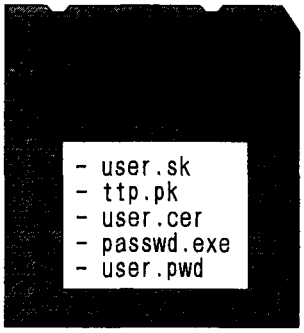
사용자 토큰으로는 스마트카드 또는 디스켓 등을 사용할 수 있다. 스마트카드 사용시 스마트카드 및 카드리더기가 필요하며, 스마트카드에 각 사용자의 비밀키를 발급하여야 한다. 따라서 추가 비용이 소요된다.

따라서 안전성에는 다소 떨어지지만 사용자 토큰으로 디스켓 사용을 제시한다. 사용자의 비밀

키는 디스켓의 패스워드를 사용하여 암호화한 후 저장하며, 서명시 서명자의 비밀키는 복호화하여 사용한다. 스마트카드의 분실 및 도난 등을 위하여 사용하는 PIN처럼 디스켓에서도 패스워드를 도입하였으며, 패스워드는 주기적으로 사용자가 변경 가능하다. 한편 패스워드 입력이 세번이상 틀리면 디스켓의 내용을 파기하는 등 추가 기법이 필요하다.

사용자의 공개키는 off-line TTP가 서명한 확증서를 발급하고, 서명된 문서의 검증시 TTP의 확증서 검증시 사용할 TTP의 공개키와 함께 사용자 디스켓에 저장한다. 또한 사용자의 암호화된 비밀키 저장 외에 확증서로 부터 공개키를 획득하기 위하여 TTP의 공개키 등이 저장된다.

(그림 7)은 사용자 인증 토큰으로 사용된 디스켓의 화일 내용을 나타낸 것이다.



- user.sk : 사용자의 패스워드에 의해 암호화된 사용자의 비밀키
- ttp.pk : TTP의 공개키
- user.cer : TTP의 비밀키에 의해 서명된 사용자의 공개키 확증서
- passwd.exe : 사용자 디스켓의 패스워드 변경 실행 화일
- user.pwd : 사용자 자신의 패스워드로 암호화된 패스워드 화일

(그림 7) 안전한 디스켓 구성도

그러면 사용자는 발급받은 디스켓을 사용하여 문서에 서명을 생성한 후 문서와 서명 및 TTP가 발급한 서명자의 공개키와 함께 수신자에게 전송한다. 수신자는 전달받은 확증서를 자신의 디스켓에 저장된 TTP의 공개키로 검증하여 서명자의 공개키를 획득한 후 이를 이용하여 문서의 정당성을 검증한다.

5. 결론

본 고에서는 안전한 EDIFACT 시스템 구성에 필요한 소요 기술을 중심으로 설명하였다. 모든 안전성 관련 시스템 개발시 항상 키 관리가 필요하며, 키 값의 저장 매체 결정도 중요한 이슈이다. 또한 적용 시스템의 특성과 시스템 규모 등을 고려하여 키 분배 프로토콜이 결정되는데, 확증서 기반의 키분배는 X.509에서 권고되었으며 이는 분산 환경에서도 적용 가능하다.

그러나 본 고에서는 중간 또는 작은 규모에서 적용 가능한 시스템을 구성하였고, PC를 주로 사용하고 있는 환경에서 EDIFACT 시스템 구현 모델과 사용자 토큰으로 디스켓 사용 방안을 제안하였다. 여기서 제안한 모델의 구현과 X.509에서 제안한 모델에 대한 성능 분석은 향후 수행할 예정이다.

[참고문헌]

1. ISO, "Electronic Data Exchange for Administration, Commerce and Transport(EDIFACT) - Application Level Syntax Rules", ISO 9735, 1988.
2. CCITT, "Message Handling Systems : EDI Messaging System", CCITT Rec. X.435, 1991
3. ISO/IEC, "Information Technology - Security Techniques - Digital Signature with Appendix; Part 1 : General", ISO/ IEC CD 14888-1, 1995. 9.
4. R. L. Rivest, A, Shamir and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", Comm. ACM, vol. 21, pp. 120-126, 1978
5. NIST, "Specifications for the Digital Signature Standard(DSS)", Federal Information Processing Standards Publication 182, 1994. 5.
6. ISO/IEC, "Information Technology - Security Techniques - Hash Functions - Part 2: Hashing Functions Employing an n-bit Block Cipher Algorithm", ISO/IEC CD 10118-2, 1995
7. ISO/IEC, "Information Technology - Security Techniques - Hash Functions - Part 3: Dedicated Hashing Functions", ISO/IEC CD 10118-3, 1995
8. Marijke De Soete, "The Key to Open EDI : Digital signature", EEMA Winter Conference '93, 1993. 1
9. Terry Dosdale, "Security in EDIFACT Systems", Computer Communications, vol. 17, No 7, pp. 532-537, 1994. 7
10. ITU-T, "The Directory - Authentication Framework", ITU-T Rec. X.509, 1993
11. Philips Communication Systems, "ETHOS : Electronic Trade Handling - Office Security, Product Range Description", 1992.