# EDI 시스템의 안전성 서비스 구현 방안에 관한 연구

*
임용진, 이강무, 고홍기, 나종근, 김동규

아주 대학교 전자계산학과

# A Study on the Implementation Method for EDI Security Services

*
Yongjin Lim, Kangmoo Lee, Hongki Ko, Jongkeun Na, Dongkyoo Kim

Depart of Computer Science , Ajou University

· 요 약

EDI 체계가 정착되기 위해서는 EDI 메세지가 법적 구속력이 있는 상용 전자문서이므로 전자문서 전달에 관련된 사람이나 장비가 기존의 상거래에서 유지되는 안전성보다 더욱 철저하게 전자문서를 관리하여야 하며,안전성 문제의 해결책이 없이는 EDI화의 진척은 한계에 부딪치게 된다. 본 논문에서는 EDI 통신에 관련된 분야 중에서 메세징 시스템 분야의 안전성을 위해 X.435에서 정의하고 있는 안전성 서비스를 구현 하기 위한 방안에 대해서 연구하였다.

## 1. 서 론

EDI(Electronic Data Interchange)란 다양한 Network 환경에서 교환되는 생산, 주문 결재, 판매등 일련의 문서정보를 업체간 또는 공공기관간의 표준화된 Format과 Code 체계를 이용하여, 인간의 중재없이 상호 합의된 형태의 거래문서를 독립된 컴퓨터 응용 프로그램끼리 관리 및 교환하는 시스템이다. 그러나 이러한 EDI 체계가 정착되기 위해서는 EDI 메세지가 법적 구속력이 있는 상용 전자문서이므로 전자문서 전달에 관련된 사람이나 장비가

기존의 상거래에서 유지되는 안전성보다 더욱 철저하게 전자문서를 관리하여야 하며, 안전성 문제의 해결책이 없이는 EDI화의 진척은 한계에 부딪치게 된다.

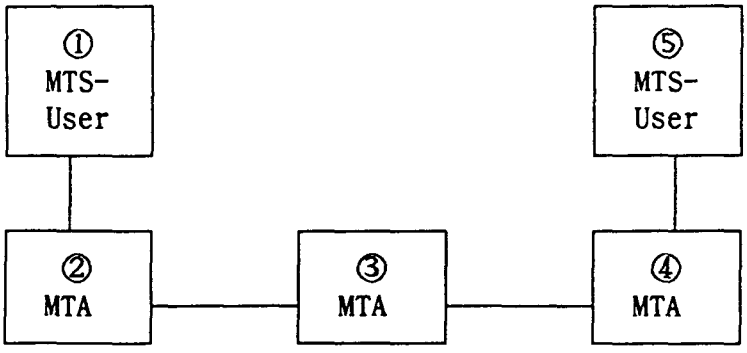
안전한 EDI 메세징을 위해서는 발신처 신분확인(Origin Authentication), 비밀성(Content Confidentiality), 무결성(Content Integrity), 부인봉쇄(Non-Repudiation)서비스등 여러가지 안전성 서비스들이 요구된다[6].

본 논문에서는 X.435에서 정의하고 있는 안전성 서비스를 구현 하기 위한 방안을 모색하기 위해 X.402 MHS에서 제공되어져야 되는 여러가지 안전성 서비스들과 메카니즘, 그리고 Pedi 프로토콜을 위해 추가적으로 제공되어져야 하는 안전성 서비스들과 관련 메카니즘을 어떻게 제공할 수 있는지에 대해서 연구하였다.

## 2. EDI 안전성 서비스의 구현 방안

### 2.1 신분 확인 서비스

안전한 통신 환경을 유지하기 위해서는 불법적인 사용자의 접근을 막는 일이 가장 우선적으로 이루어져야 한다. 이를 위해 가장 먼저 해야할 일은 통신 상대자가 적법한 상대자인지를 알아야 하며, 상대방에게도 내가 적법한 통신 당사자라는 것을 인식시켜 주어야 한다. 이러한 안전한 채널의 확립을 위해 상대방의 신분을 확인하고 자신의 신분을 증명하는 것을 신분확인(Authentication)이라 한다[8]. EDIMS(EDI Messaging System)와 같은 Store-and-forward 통신환경에서는 agent와 agent 사이의 신분확인을 위해 대등실체 신분확인이 필요하고, 양단간 EDI-UA 사이에서 발신처의 신분확인을 위해 메세지 발신처 신분확인이 필요하다.



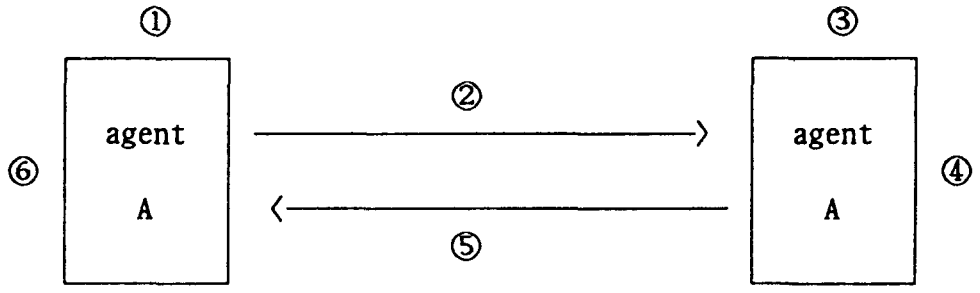

〈그림 1〉 MTS의 기능적 모델

### 2.1.1 대등실체 신분확인 서비스

EDIMS의 대등실체 신분확인 서비스는 각각의 agent 사이 즉, 〈그림 1〉에서 1-2, 2-3, 3-4, 4-5 사이에서 각각 제공되어 져야 한다.

본 논문에서는 대등실체 신분확인 서비스를 Simple 신분확인과 Strong 신분확인의 두가지 방안으로 모색해 보았으며, Simple 신분확인의 경우 간단하게 Password를 교환함으로써 이루어 질 수 있다. 또한 Strong 신분확인의 경우 암호화된 토큰을 교환하게 됨으로써 달성될 수 있는데 EDIMS의 경우 MTS_Bind, MS_Bind 그리고 MTA_Bind 시에 이러한 신분확인을 위한 정보를 교환하게 된다. 이때 사용되는 아규먼트로는 InitiatorCredentials과 bind 오퍼레이션에 대한 결과로써 ResponderCredentials가 사용된다[5].

InitiatorCredentials과 ResponderCredentials를 이용한 상호신분확인(Mutual Authentication)서비스는 Certification Path와 상대방의 공개키로 암호화된 신분확인을 위한 토큰을 보내고 상대방으로 부터 자신의 공개키로 암호화된 토큰을 받는 방법으로 이루어 질 수 있다. 자세한 작동과정을 나타내면 〈그림 2〉와 같다[7].




〈그림 2〉 상호신분확인(Mutual Authentication)의 동작 과정

▷ $P_X[msg]$ : X의 공개키로 msg를 암호화
▷ $S_X\{msg\}$ : X의 비밀키로 msg를 서명(signing)
▷ X->Y : Certification Path
▷ $r_X$ : X의 random number ▷ $t_X$ : X의 timestamp

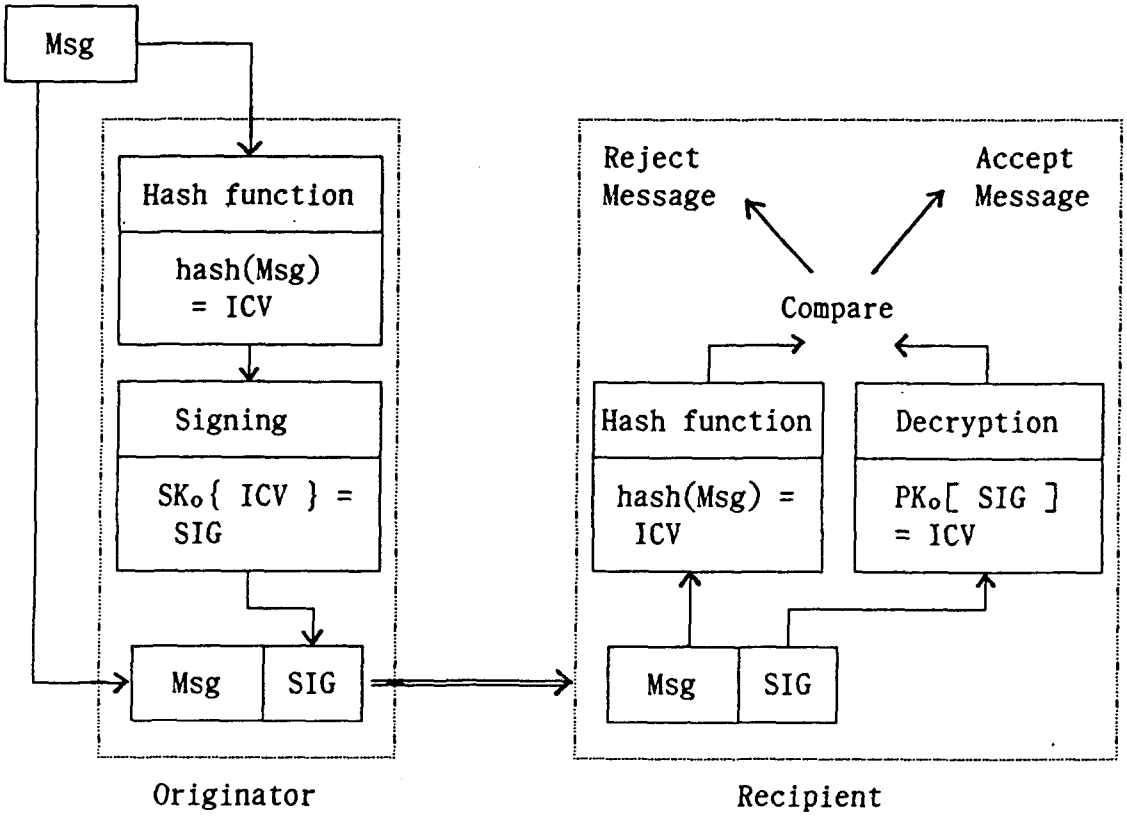
Step 1 : $r_A$를 생성
Step 2 : 다음과 같은 메세지를 B에 전송 : B->A, $P_B[\ S_A\{t_A, r_A, B\}\ ]$
Step 3 : - Certification Path B->A로 부터 $P_A$를 구한다.
- 서명된 메세지를 확인하고 timestamp가 'current'인가 확인.
Step 4 : $r_B$를 생성

Step 5 : 다음과 같은 메세지를 A에 전송 : $P_A[\ S_B\{t_B, r_B, A, r_A\}\ ]$
Step 6 : 서명된 메세지를 확인하고 timestamp가 'current'인가 확인.

### 2.2.2 메세지 발신처 신분확인 서비스

EDIMS와 같은 One-way Communication은 송신자와 수신자 간에 상호 연락 없이, 송신자측에서 일방적으로 전문을 전송하게 되는 형태이다. 따라서 〈그림 1〉에서 볼때 1-5 사이에서 즉, 양단간의 MTS-User인 Originator와 Recipient사이에 메세지 발신처 신분확인 서비스가 제공되어져야 한다.

발신처 신분확인 서비스는 수신된 메세지가 메세지의 발신처(Originator)파라미터에 있는 ORName의 사용자로 부터 발신되어진 것인지를 증명하는 서비스로 메세지에 대한 서명(Signature)과 같은 개념이다. EDIMS에서 이 서비스는 MHS의 P1 엔빌로프(envelope)의 발신처 신분확인(Original




ICV = Integrity Check Value
Msg = EDIM의 Body Part + 기타 아규먼트
SKo{ ICV } = MAC(Message Authentication Code)
= ICV를 발신처의 비밀키로 서명

〈그림 3〉 메세지 발신처 신분확인 절차

Authentication)에 의해 달성될 수 있으며 공개키 암호화 알고리즘을 사용할 경우 〈그림 3〉과 같이 나타낼 수 있다.

이러한 과정에서 서명(signature:SIG)은 P1 엔빌로프를 구성하는 Message_origin_authentication_check 아규먼트를 사용하여 전송할 수 있으며 이를 이용해서 메세지 발신처 신분확인(Proof of the origin of the message, Message origin authentication)서비스를 제공할 수 있다[5].

## 2.2 메세지 무결성 서비스

메세지 무결성 서비스는 발신처에서 제출(submit)된 메세지가 수신처에 수신되기 전에 어떠한 방법으로도 변경되지 않았음을 증명하는 서비스로서 크게 내용(content)무결성과 순서(sequence)무결성 서비스로 나누어 생각할 수 있다[8].

내용무결성은 발신처에서 전송되는 정보로 부터 메세지에 대한 특성값을 계산하여 전송되는 정보에 추가시키는 것으로서 수신처에서는 수신된 정보로 부터 계산된 특성값과 전송된 특성값을 비교함에 의해 메세지의 수정, 삽입, 제거 혹은 재전송 공격을 검출 할 수 있다.

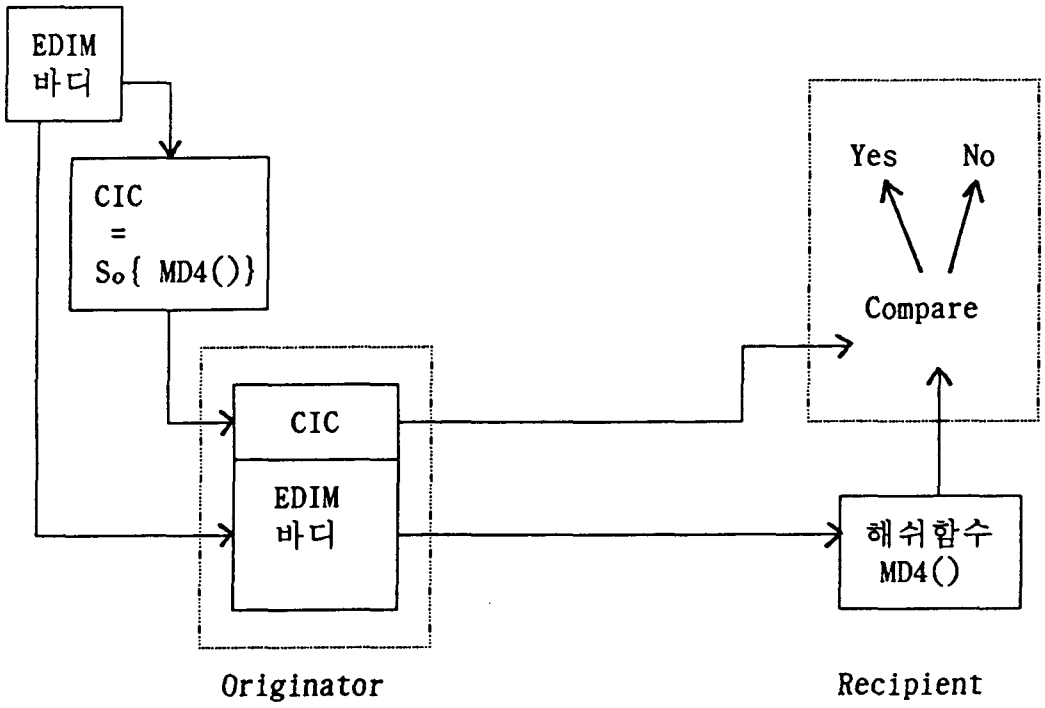
EDIMS의 경우 메세지 제출(submission)시에 ContentIntegrityCheck 아규먼트를 이용하여 EDIM(EDI Message)의 바디부분의 내용무결성을 〈그림 4〉와 같은 방법으로 구현 할 수 있다.

우선 발신처에서는 본 논문에서 시험적으로 구현 사용한 해쉬함수 MD4의 입력으로 content와 기타 아규먼트를 이용하여 내용무결성을 위한 코드를 생성하고 이를 자신의 비밀키로 암호화하여 ContentIntegrityCheck 아규먼트를 만들고 EDIM의 바디부분에 덧붙여 함께 전송한다. 이를 수신한 수신처에서는 수신된 EDIM과 나머지 아규먼트를 사용하여 해쉉을 하여 구한값과 수신된 ContentIntegrityCheck 아규먼트의 값을 복호화한 값과 비교한다.

또한, EDIMS의 경우 MessageToken을 이용하여 ContentConfidentiality-AlgorithmIdentifier, MessageSecurityLabel, ProofOfDeliveryRequest등의 아규먼트 무결성을 제공할 수 있다. Message Token은 메세지 제출(Submission), 메세지 전송(Transfer), 메세지 배달(Delivery)서비스에 각각 존재하는 아규먼트이다[5].

그리고, 메세지의 순서전도나 중복을 방지하기 위한 메세지 순서 무결성 서비스는 각각의 메세지에 개별적으로 일련의 번호를 부여함으로서 전체 메세지 내에서 각 메세지의 위치를 나타내는 방법으로 구현 될 수 있는데 EDIMS의 경우 P1 엔빌로프를 구성하는 아규먼트 중에서 MessageToken 아규먼트의 message-sequence-number 필드를 이용하여 각 메세지의 해당 순서번

호를 전송할 수 있으며 이것으로 메세지의 순서 무결성을 제공할 수 있다.




〈그림 4〉 메세지 무결성 서비스의 절차

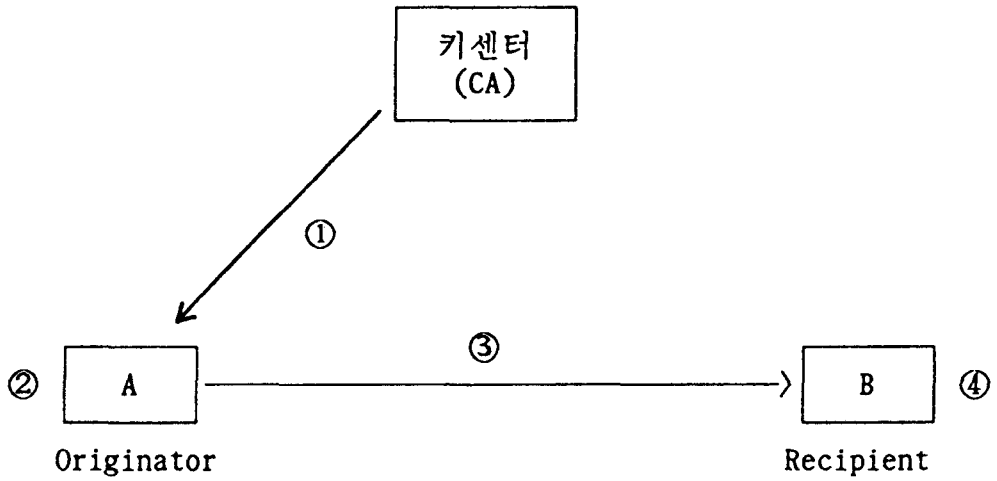
## 2.3 메세지 비밀성 서비스

메세지 비밀성 서비스는 통신되는 메세지가 불법적으로 그 내용이 노출되는것을 방지하는 서비스로서 네트워크상에서 비밀성을 유지하기 위하여 일반적으로 가장 많이 사용되는 것이 암호화 방법이다. 암호화 기술은 네트워크 통신에 대한 안전성을 보장할 뿐만 아니라 정보에 대한 확인(authentication)을 제공한다[8].

자세한 과정을 보면 〈그림 5〉와 같다. 먼저 사용자 A는 키센터에게 통신하고자 하는 상대방 B의 확인표(Certificate)를 요청하고 분배받는다[3]. 사용자 A에서는 키센터의 비밀키로 서명되어 있는 확인표를 가지고 미리 분배 받은 키센터의 공개키를 이용하여 인증된 B의 공개키 $P_B$를 얻게되고 이를 이용하여 A에 의해서 생성된 대화키 $WK_{AB}$를 암호화 한다. 또한 A에서는 생성한 $WK_{AB}$를 이용하여 EDIM의 바디 부분을 암호화 한다. 다음으로 A는 신분확인을 위한 정보(AuthInfo)와 $WK_{AB}$를 content-confidentiality-key 필드를 이용하여 전송한다. 또한, P1 엔빌로프를 구성하는 Originator-Certificate 파라미터를 이용하여 발신처 A의 확인표를 전송한다. 만약 A와 B가 같은 CA(Certification Authority) 내에 있지 않을 경우에는 A는 B에게

역방향의 확인표 경로(Reverse Certification Path)인 B->A를 전송하게 되고 이를 이용하여 수신처 B에서는 인증된 A의 공개키 $P_A$를 구하게 되고, AuthInfo를 이용하여 <그림 3>에서와 같은 방법으로 신분확인 과정을 수행하게 된다[8]. 신분확인이 이루어진 후에 B는 자신의 비밀키 $S_B$를 가지고 $P_B\{WK_{AB}\}$를 복호화 하여 $WK_{AB}$를 얻고 이를 이용하여 본래의 EDIM 바디 부분을 얻게된다. 이러한 절차를 통해 EDIMS에서는 양단간 즉,<그림 1>에서 1-5 사이의 메세지 비밀성 서비스를 구현할 수 있다.

또한, 메세지의 흐름을 관찰하고자 하는 위협요소에 대한 메세지의 흐름 비밀성(flow confidentiality)서비스는 간단하게 교통의패딩(traffic padding)과 같은 Double Enveloping 기법을 사용하여 메세지 헤드에 있는 어드레스 정보들을 숨기는 방식으로 구현될 수 있다[8].




step 1 : KC를 통해 사용자 B의 인증표를 분배 받는다.
step 2 : 인증표로 부터 인증된 상대방의 공개키 $P_B$를 구한다.
step 3 : $P_B[WK_{AB}]$, $WK_{AB}$[ EDIM_Body ], AuthInfo, B->A
step 4 : - Certification Path B->A로 부터 $P_A$를 구하고 AuthInfo를 이용하여 신분확인 과정을 수행한다.
- $S_B$를 이용하야 $WK_{AB}$를 구하고 EDIM_Body를 구한다.

<그림 5> 메세지 비밀성 서비스의 수행 절차

## 2.4 증명 및 부인봉쇄 서비스

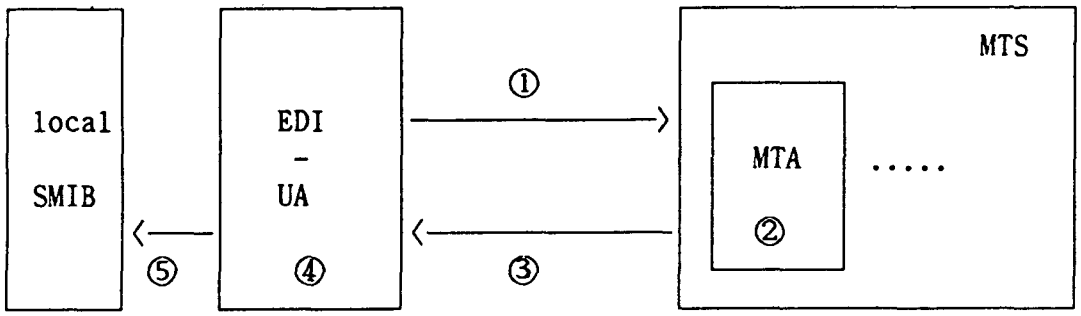
어떤 전문을 컴퓨터 통신망을 통하여 전송하였을 때 수신자가 고의적으로 전문 내용을 자신에게 이롭도록 고치거나, 비양심적인 송신자가 전문을

보내지 않았다고 부인하거나, 혹은 전문 내용을 수신자가 위조해서 자신이 보낸 내용과 틀리다고 주장한다면 EDI 시스템과 같이 법적 구속력이 있는 문서를 다루는 경우 결국 송신자와 수신자 사이에 심각한 분쟁이 발생하게 될 것이다. 이와 같은 분쟁을 해결하기 위해 전문의 전송에 대한 확인과 이미 발생한 통신사실을 부인할 수 없도록 하는 것을 증명 및 부인봉쇄(Proof / Non-repudiation) 서비스라 한다[8].

### 2.4.1 제출(Submission) 증명 / 부인봉쇄 서비스

이 서비스는 〈그림 1〉에서 1-2 사이에 제공 되어져야 할 서비스로서 제출 증명 서비스는 메세지의 발신자가 본래 의도한 수신처(들)에게로 배달하고자 하는 메세지를 MTS가 수신했다는 확인응답을 얻고자 하는 서비스이며, 제출 부인봉쇄 서비스는 메세지가 MTS에 제출 되어 졌다는 사실을 부인할 수 없도록 하는 서비스이다[4,6].

이들 서비스는 메세지 제출 아규먼트(P3 Envelope Argument)인 proof-of-submission-request와 proof-of-submission을 이용해서 이루어 질 수 있다. 자세한 과정은 〈그림 6〉과 같다.




〈그림 6〉 제출 증명 및 부인봉쇄 서비스의 수행 절차

step 1 : P3 엔빌로프의 ProofOfSubmissionRequest를 true로 세팅함으로서 제출 증명 서비스를 요구하고 또한 메세지 제출 Id와 제출 시간도 세팅하여 제출한다.
ProofOfSubmissionRequest의 default 값은 false이다.

step 2 : MTS에서 메세지 제출 요청을 받은 MTA는 자신의 비밀키를 이용하여 ProofOfSubmission을 만든다.
ProofOfSubmission = $S_{MTA}$(arguments) $S_{MTA}$ : MTA의 비밀키

step 3 : 만들어진 ProofOfSubmission을 제출에 대한 결과의 일부분인

proof-of-submission 아규먼트를 이용하여 메세지를 제출한 EDI-UA에게 돌려 보낸다.

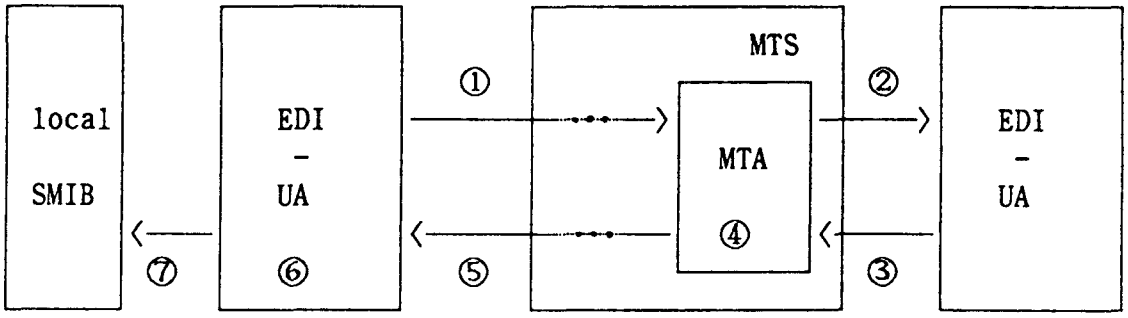
step 4 : 메세지의 발신처인 EDI-UA에서는 ProofOfSubmission을 수신함으로서 자신이 제출했던 메세지가 MTS에 안전하게 제출되었다는 사실을 확인하게 된다.

step 5 : EDI-UA에서는 수신한 ProofOfSubmission을 후에 발생될 수 있는 분쟁에 대비하여 자신의 로칼 SMIB에 저장함으로서 log file을 유지하게 된다.

### 2.4.2 배달(Delivery) 증명 / 부인봉쇄 서비스

이 서비스는 〈그림 1〉에서 1-4 사이에 제공 되어져야 할 서비스로서 배달 증명 서비스는 MTS에 의해서 발신처에서 의도한 수신처(들)에게 배달 했다는 확인응답을 얻고자 하는 서비스이며, 배달 부인봉쇄 서비스는 이를 이용하여 메세지 헤더에 명시된 수신처(들)에게 배달을 했다는 사실을 MTS가 부인할 수 없도록 하는 서비스이다[4,6].

이들 서비스는 메세지 제출 아규먼트 중에 하나로 P1 엔빌로프를 구성하는 것중에 하나인 proof-of-delivery-request와 메세지 제출의 결과에 해당하는 레포트 아규먼트 중에 하나인 proof- of-delivery를 이용하여 이루어질 수 있다. 자세한 과정은 〈그림 7〉과 같다.




〈그림 7〉 배달(Delivery) 증명 및 부인봉쇄 서비스의 수행 절차

step 1 : 메세지 제출 아규먼트인 ProofOfDeliveryrequest를 true로 세팅함으로서 배달 증명 서비스를 요구한다.
ProofOfDeliveryRequest의 default 값은 false이다.

step 2 : 메세지를 수신한 최종 MTA는 메세지 헤더에 명시된 수신처(들)에게 메세지를 배달한다.

step 3 : 메세지를 수신한 수신처(들)에서는 proof-of-delivery-request가 true일 경우 ProofOfDelivery를 생성하여 메세지를 배달한 MTA에게 전송한다.

ProofOfDelivery = $S_r${arguments}  $S_r$ : 수신처의 비밀키

step 4 : 배달을 끝낸 MTA에서는 메세지의 제출에 대한 결과로서 step 3에서 수신한 ProofOfDelivery와 또다른 아규먼트들을 생성하여 레포트(report)를 만든다.

step 5 : 만들어진 레포트를 메세지 제출시에 경유했던 경로를 통해 메세지의 발신처로 전송한다.

step 6 : 레포트 배달 서비스에 의해 수신한 레포트의 아규먼트중에서 ProofOfDelivery를 통해 이전에 제출한 메세지가 자신이 의도한 수신처(들)에게 안전하게 배달되었다는 확인을 얻게된다.

step 7 : 레포트를 수신한 EDI-UA는 ProofOfDelivery를 후에 발생될 수 있는 분쟁에 대비하여 자신의 로칼 SMIB에 저장함으로서 log file을 유지한다.

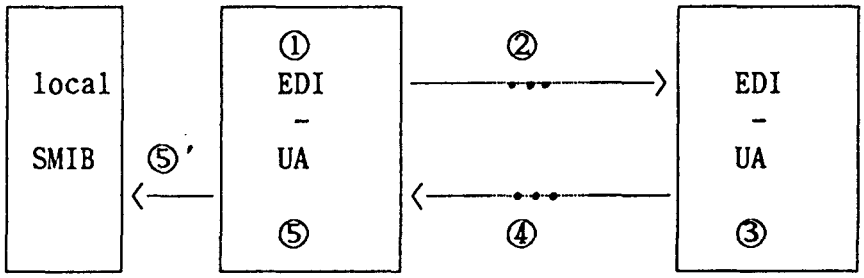
### 2.4.3 EDI 통지(Notification) 증명 / 부인봉쇄 서비스

EDIMS에서 EDI-UA 간에 통지 증명 서비스는 발신처에서 제출된 메세지가 의도한 수신처에 배달되었는지를 확인하고 또한, EDIM에 대한 책임이 수용(accepted)되었는지, 회송(forwarded)되었는지 또는 거부(refused)되었는지를 확인하기 위한 것으로 수신한 메세지에 대해 서명을 생성하여 발신처로 되돌려 보내는 방식으로, EDI 통지(Notification)를 사용하여 이루어 질 수 있다[4,6].

EDI 통지 증명 서비스를 요구하는 방법은 EDIM 헤딩내의 관련 필드를 세팅함으로써 이루어 질 수 있다. 먼저 EDIM의 발신처에서는 Notification Requests 필드에 PN(Positive Notification), NN(Negative Notification) 또는 FN(Forwarded Notification)을 요구하고 Notification Security 필드를 "proof"로 세팅한다. 이를 수신한 EDIM의 수신처이자 EDIN(EDI Notification)의 발신처에서는 수신된 메세지에 대한 서명을 생성하여 EDIN을 만들고 이것을 다시 EDIM 발신처에 전송하게 되며 이를 수신한 EDIM 발신처에서는 EDIN 발신처의 신분을 확인하고 EDIM이 안전하게 배달되었는지를 확인할 수 있다. EDI 통지 증명 서비스의 자세한 수행 과정은 〈그림 8〉과 같다. 다음으로 EDI 통지 부인봉쇄 서비스의 경우 우선 Notification Security와 Reception Security 필드를 "non-repudiation"으로 세팅함으

로써 이 서비스를 요구하게 된다.

그 이외의 과정은 증명(Proof) 서비스와 같은 과정으로 이루어 지며 step 3에서 수신된 EDIM 헤딩에서 얻어진 CIC를 그대로 복사하여 사용하지 않고 수신처 자신의 비밀키로 서명한 새로운 CIC를 만들어 EDIN의 헤딩 필드중에 하나인 Original_Content_integrity_check 필드를 이용하여 EDIM의 발신처로 전송하게 되고, step 5에서는 후에 발생될 수 있는 분쟁에 대비하여 step 5'와 같이 수신된 EDIN으로 부터 EDIM 수신을 부인할 수 없는 EDIN 발신처에 대한 확인으로 Original_Content_integrity_check 필드의 값을 얻어 자신의 디렉토리에 log 화일을 유지한다.




step 1 : EDIM 헤딩내의 NotificationRequest 필드를 이용하여 PN, NN 또는 FN을 요구하고 NotificationSecurity 필드와 ReceptionSecurity 필드를 "proof"로 세팅하여 서비스를 요청한다.

step 2 : 발신처에서는 메세지를 해쉬함수에 적용하고 그 결과를 자신의 비밀키를 이용하여 서명을 생성하고 메세지 제출(Submission)아규먼트 중에서 CIC(Content Integrity Check) 필드를 이용하여 본래의 메세지와 함께 전송한다. ContentIntegrityCheck = $S_o\{ hash(msg) \}$

step 3 : 수신처에서는 수신한 EDIM으로부터 얻은 CIC를 EDIN의 공통 필드중에서 Notification_security_elements의 서브필드인 Original_Content_integrity_check 필드에 복사한다. 만약 발신처에서 해쉬함수와 암호화 메카니즘이 제공되지 않을 경우에는 EDIN의 발신처에서는 Original_Content_integrity_check 필드를 사용하지 않고 Original_content 필드를 이용하여 수신한 EDIM의 모두를 복사하여 전송한다.

step 4 : EDIM의 수신처는 만들어진 EDIN을 EDIM의 발신처로 전송한다.

step 5 : EDIM의 발신처에서는 step 1에서 전송되어진 CIC와 수신한 EDIN으로 부터 얻은 CIC를 비교하여 EDIN 발신처의 신분을 확인하고 자신이 전송했던 EDIM이 안전하게 전해진것에 대한 확인을 얻게된다.

〈그림 8〉 EDI 통지 증명 서비스의 수행 절차

## 3. 결 론

안전한 EDI 메세징을 위해서는 발신처 신분확인(Message Origin Authentication),비밀성(Content Confidentiality), 무결성(Content Integrity), 부인봉쇄(Non-Repudiation)서비스등 여러가지 안전성 서비스들이 요구된다.

본 논문에서는 CCITT X.435에서 정의하고 있는 안전성 서비스에 따라 EDIMS에서 각각의 agent 사이에서의 대등실체 신분확인과 양단간 EDI-UA 사이에서의 메세지 발신처 신분확인 서비스의 구현 방안을 설계하였고, 메세지 무결성 서비스와 비밀성 서비스 그리고 X.402에 정의된 증명 및 부인봉쇄 서비스와 X.435에 EDIMS를 위해 추가적으로 정의되어 있는 증명 및 부인봉쇄 서비스의 구현 방안을 설계하였다. 이러한 서비스의 구현 방안 설계에 있어 키관리 방식은 Internet 표준에서 권고하고 있는 확인표 방식에 의한 키분배 메카니즘을 이용하였다. 또한, 암호화 알고리즘으로 RSA와 FEAL 그리고 해쉬함수로 MD4를 구현하여 Sun 4/75M과 Sun 3/140을 Ethernet으로 연결한 환경에서 Socket을 이용하여 각각의 서비스 수행 절차를 시험적으로 구현하여 보았다. 앞으로 안전성과 관련된 모든 메카니즘 및 알고리즘을 구현하고 실제 EDIMS에 Integration시키기 위한 연구가 진행 될 것이다.

## 참고 문헌


[1] Richard Hill, "EDI and X.400 using $P_{edi}$:The Guide for Implementors and Users", Technology Appraisals, 1990.

[2] R.L. Rivest, "The MD4 Message Digest Algorithm", CRYPTO'90.

[3] S. Kent & J.Linn,"Privacy Enhancement for Internet Electronic Mail : Part Ⅱ -- Certificate-Based Key Management", RFC-1114, Aug 1989.

[4] CCITT Recommendation F.435, "Message Handling Service : Operations and Definition of Service", 1991.

[5] CCITT Recommendation X.400 Series (X.400-X.420), 1988.

[6] CCITT Recommendation X.435, " Message Handling systems : EDI Messaging System", version 6.0, Nov 1990.

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

[8] 김동규, 임용진 외,"OSI 통신망 구조에서의 네트워크 안전체제 연구", 과학기술처 최종 보고서 (3차 년도), 아주대, 1991.6.