COPINO 전자문서를 위한 보안 메커니즘의 설계 및 구현

오 천 보, 이 인 실, 이 경 현

부경대학교 전자계산학과

A Design and Implementation of Security Mechanism for COPINO EDI

Cheon-Bo Oh, In-Sil Lee, Kyung-Hyune Rhee

Dept. of Computer science

Pukyong Nat'l University

요 약

본 논문에서는 평문 COPINO 전자문서의 기밀성 제공을 위해 기존의 UN/EDIFACT 보안 기법을 수정하여 재설계한 후 이를 구현하였다. 또한 전자서명의 효율성 증대를 위해 사용되는 해쉬함수를 기존 UN/EDIFACT에서 제공하는 MD5대신 안전성이 보다 증대된 SMDx를 사용하여 구현하였다. 사용된 보안 메커니즘들은 수행속도측면에서 기존 방식과 거의 차이가 없으므로 안전성 측면에서 효과적임을 컴퓨터 시뮬레이션을 통해 확인하였다.

1. 서론

고도 정보화사회의 도래에 따른 컴퓨터와 통신 기술의 급속한 발전으로 기업체를 포함한 대부분의 조직체에서는 업무에 필요한 대량의 정보를 신속하고 정확하게 원하는 장소에까지 전달할 수 있게 되자 기존의 보급된 컴퓨터 시스템을 이용하여 거래하고자 하는 상대방 기업체와 중요한 거래정보를 주고받는 시스템의 구축이 가능하게 되었고 이는 EDI(Electronic Data Interchange)라는 전자문서 교환의 급속한 확대로 나타나게 되었다. EDI를 사용함으로써 정보처리에 있어서 많은 효과를 가져올 수 있는 반면에 정보의 불법적인 접근, 도청, 수정, 삭제, 재전송, 삽입, 순서변경, 메시지 송수신자의 송수신 사실 부인 및 기타 위협요소가 상존하고 있다.

따라서 본 논문에서는 컨테이너의 인도나 인출에 관련된 COPINO(Container Pre-Notification Notice)전자문서를 UN/EDIFACT(EDI for Administration Commerce and Transport)에서 제시한 메시지 보안 기법[1]을 기준으로 메시지 보안을 위한 전송항목을 재 설계하고 전자서명을 효율적으로 사용하는 방법인 해쉬 기법중 현재 일반적으로 사용하는 MD5와 부경대학교 네트워크 및 정보보호 연구실에서 새로이 개발된 SMDx(Strengthened Message Digest series)[2]를 COPINO 문서에 적용한 결과를 서로 비교 분석하였다. 또한 UN/EDIFACT에서는 제안되지 않은 UNSM(UN Standard Message)의 메시지 자체를 암호화하기 위해 데이터 기밀성에 대한 별도

의 항목을 추가하여 COPINO EDI 보안 메커니즘을 설계 및 구현하였다.

2. EDI 개요 및 UN/EDIFACT

2.1 EDI의 정의 및 특징

EDI란 기업내부나 기업간에 컴퓨터 응용프로그램을 이용해서 DATA의 재입력 과정 없이 즉시 업무에 활용할 수 있도록 전달하는 전자적 방법[3]으로 메시지를 표준화하여 컴퓨터간의 DATA를 교환하는 것이다. EDI를 요약해서 정의하면 "다른 조직에 전달할 메시지를 일정한 표준화된 규약에 의해 컴퓨터간의 통신 교환을 하는 것"이라고 할 수 있다. EDI에 의한 효과는 정보 교환 시간과 처리 시간이 단축되고 데이터 입력시에 발생하는 에러를 감소시킬 수 있으며 주문 확인의 신속화, 물류 비용 감소, 사무 인력 및 비용의 감소와 우송 및 문서 작성에 소요되는 비용, 시간의 감소 등이 있다. 오늘날과 같이 급속하게 정보가 변화하고 많은 데이터량이 교환되는 시점에서 EDI의 적용은 비용과 시간을 절감하는 효과가 있다.

2.2 EDIFACT의 정의[4]

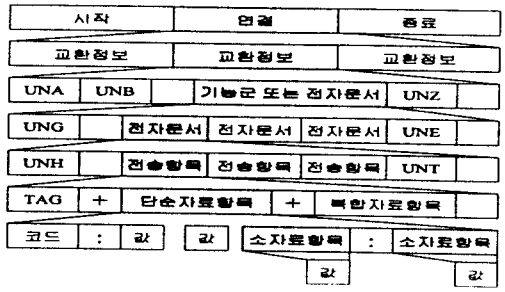
EDIFACT는 행정이나 상업 및 운송 분야의 관계자간의 전자문서 교환에 있어서의 사용자 데이터 및 관련 보조 자료의 구조화에 관한 어플리케이션 계층의 구문규칙을 정한 규격이다. EDIFACT에서는 EDI에 관련된 권고안으로 다음과 같은 사항을 규정하였다.

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

EDIFACT메시지는 다음과 같은 5개 기본 요소로 구성된다.

-. 교환정보(Interchange)
-. 기능군(Functional group)
-. 전자문서(Message)
-. 자료전송항목(Segment)
-. 단순/복합자료항목(data element)

<그림1>은 교환정보의 계층적 구조를 보여주며 여기서 EDIFACT메시지의 기본단위는 자료항목(DATA Element)이다.



<그림 1> 교환정보 계층적 구조

2.3 문서표준

EDI는 표준화된 문서를 전달하기 때문에 UN에서는 1989년 9월 국제표준 ISO 9735 EDIFACT을 국제적인 EDI표준으로 승인하였다. EDI로 교환되는 데이터의 내용과 구성은 UNTDID(UN Trade Data Interchange Directory, UN무역데이터교환디렉토리지침서)로 제정되어 있으며 문서표준인 UN/EDIFACT은 다음과 같은 사항들로 구성된다.

(1) 구문규칙(EDIFACT Syntax Rules : ISO 9735)
데이터 요소 및 서비스 세그먼트를 전자문서로 포맷팅하는데 필요한 표준 구문규칙이 정의되어 있다.

(2) 데이터 요소 목록
행정, 상업, 운송분야에 사용되는 데이터의 속성 및 정의가 규정되어 있으며 경우에 따라서는 데이터 항목별 코드와 한정어 목록도 규정되어 있다.

(3) 표준 데이터 세그먼트 목록
표준문서 목록에 사용되는 표준 데이터 서비스 세그먼트가 모두 기술되어 있다.

(4) 표준문서 목록(EDIFACT Message Directory)
UN에서 승인한 표준문서가 수록되어 있으며 목록의 최신화를 위한 관리절차도 포함되어 있다.

(5) 표준문서 설계지침서
(EDIFACT Message Design Guidelines)

3. EDI보안 시스템

3.1 EDI보안의 필요성[5]

EDI메시지는 두 가지 요인에 의해 안전에 위협을 받는데 첫 번째는 기술적인 면에 있어서 기계의 오동작일 경우이며, 두 번째는 인간의 실수나 악의로 유발되는 경우로 이러한 위협을 해결하기 위하여 EDI메시지를 주고 받는 송,수신자외에는 EDI문서를 알아 볼 수 없도록 암호화 기법을 적용하거나 메시지의 진위를 인증하는 알고리즘으로 해결할 수 있

다.

3.2 보안의 위협요소 및 해결방안

송,수신자 외에 제3자가 EDI메시지를 변경시키지 못하도록 암호 알고리즘을 적용하는데 암호 알고리즘으로는 공개키 암호 알고리즘과 비밀키 암호알고리즘의 두 가지 방법이 있으며 EDIFACT메시지를 안전하게 처리하기 위한 필요조건과 기술적 사항들을 포함하는 서비스들은 다음과 같다.

(1)메시지 순서 무결성(Message sequence integrity)
메시지의 중복, 첨가, 삭제, 유실, 재전송 등을 막기 위한 것으로 디지털 서명 등을 써서 안전성을 보장할 수 있다.

(2)메시지 내용 무결성(Message content integrity)
전송 도중의 메시지 변조를 막기 위한 것으로, 발신처 인증과 발신처 부인봉쇄의 한 부분으로 구현할 수 있다.

(3)메시지발신처인증(Message origin authentication)
실제로 메시지를 송신한 사람이 나중에 메시지 송신을 부인하지 못하도록 메시지 내용 무결성도 포함하면서 발신처 부인 봉쇄의 일부분으로 구현할 수 있다.

(4)발신,수신 부인 봉쇄(Non-repudiation of origin)
송신자나 수신자가 메시지를 보내거나 받은 사실을 부인하는 경우에 디지털 서명 기법을 적용하여 부인 봉쇄를 할 수 있다.

(5)메시지 기밀성(Confidentiality of content)
메시지의 송,수신 도중에 제3자가 메시지를 가로채어 메시지 내용을 변경할 수 없도록 메시지를 암호화 알고리즘을 적용함으로써 메시지 기밀성을 보장할 수 있다.

4. 암호화 시스템

4.1 암호화 기법(Cryptography)[6]

암호화되지 않은 상태의 평문(Plaintext)을 암호문(Ciphertext)으로 만드는 것을 암호화(Encryption)과정이라 하고 반대로 암호문을 평문으로 변환하는 것을 복호화(Decryption)과정이라 한다.

송신한 데이터가 제3자에 의해 조작되지 않도록 암호화 기법을 적용하는데 코드가 해독되지 않도록 하는 두 가지 최선의 방법은 복잡한 키를 이용하는 것과 복잡한 알고리즘을 이용하는 것으로 대표적인 알고리즘으로는 대칭키 또는 비밀키 암호화 시스템의 DES(Data Encryption Standard)[7]와 공개키 암호 알고리즘인 RSA가 있다. DES는 컴퓨터 데이터를 보호하기 위한 수학적인 알고리즘으로 이 알고리즘은 BCD(Binary Code Decimal)데이터를 사용하여 64비트의 정보를 암호화하기 위하여 64비트 키를 사용한다. 알고리즘은 공개된 것이므로 결국 키를 안전하게 보호하는 것이 중요하다. DES는 64비트 키 중에서 56비트를 사용하고 나머지 8비트는 패리티 검사용으로 사용한다. RSA는 1977년 MIT공대의 R.L.Rivest, A.Sharmir, L.Adleman이 소인수 분해의 복잡함을 이용하여 개발된 알고리즘으로 암호화 키와 복호화 키가 서로 다르며 큰 합성수 n(n=pq, p,q는 소수)의 인수분해(Factorization) 문제의 어려움에 안전성을 둔다.

4.2 해쉬 함수

해쉬함수는 디지털 서명, 메시지 인증, 키 유도와 같은 분야에서 많이 사용하고 있으며 임의 길이의 비트스트링을 입력으로 받아 고정된 짧은 길이(128, 160 비트)의 비트스트링을 출력하는 함수이다.

4.2.1 MD5(Message Digest 5)

1990년 Rivest가 제안한 MD4의 설계원리에 기반을 두고 MD4의 취약점을 개선시켜 개발한 것으로 MD5에서 사용되는 4개의 비선형 함수는

$F(X, Y, Z) = (X \vee Y) \vee ((\sim X) \wedge Z)$

$G(X, Y, Z) = (X \wedge Z) \vee (Y \wedge (\sim Z))$

$H(X, Y, Z) = X \oplus Y \oplus Z$

$((X, Y, Z) = Y \oplus (X \vee (\sim Z))$

와 같고 각 라운드마다 사용하는 부울함수는 다음과 같다.

$FF(a, b, c, d, M_{j,s,t_i}): a = b + ((a + F(b, c, d) + M_j + t_i) \ll S)$

$GG(a, b, c, d, M_{j,s,t_i}): a = b + ((a + G(b, c, d) + M_j + t_i) \ll S)$

$HH(a, b, c, d, M_{j,s,t_i}): a = b + ((a + H(b, c, d) + M_j + t_i) \ll S)$

$II(a, b, c, d, M_{j,s,t_i}): a = b + ((a + I(b, c, d) + M_j + t_i) \ll S)$

MD5는 512bit의 메시지 블록단위로 처리하여 128bit의 해쉬값을 출력하고 각 라운드는 16단계로 이루어진다.

4.2.2 제안된 새로운 해쉬 기법 SMDx[2]

본 연구실에서 개발한 해쉬알고리즘은 기존 MD(Message Digest)계열을 근간으로 이를 안전성 관점에서 강화시킨 알고리즘으로 SMDx (Strengthened version of MD-series)라 부른다. SMDx의 개략적인 구성은 다음과 같다.

-. 임의 길이 메시지를 512bit 단위로 처리하며 160bit를 출력한다.

-. 입력 데이터에 의존하는 순환이동이 기존의 MD5와 비교되는 특징이다.

-. 16개 입력 메시지는 8개 메시지 변수를 추가로 생성한다.

추가된 8개변수의 생성 함수와 부울함수는

$X_{16+i}=(X_{0+i}\oplus X_{2+i}\oplus X_{7+i}\oplus X_{12+i})^{\lll 1}$

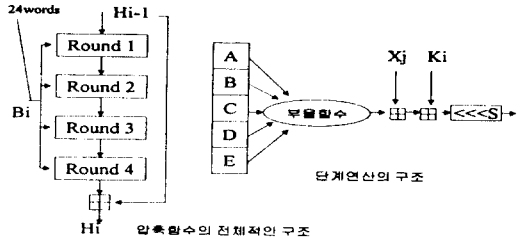
$f_0(x_1,x_2,x_3,x_4,x_5)=x_1x_2\oplus x_3x_4\oplus x_2x_3x_4\oplus x_5$

$f_1(x_1,x_2,x_3,x_4,x_5)=x_2x_3\oplus x_4x_5\oplus x_1$

$f_2(x_1,x_2,x_3,x_4,x_5)=x_1x_3\oplus x_2x_5\oplus x_3x_5\oplus x_4$

와 같고 SMDx의 구조도는 <그림 2>와 같다.

B는 메시지를 나타내고 H는 초기값, '$\lll S$'는 S만큼 좌측순환 쉬프트를 의미한다.

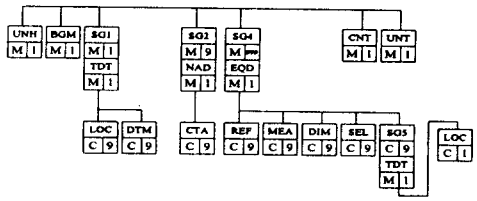


<그림 2> 제안된 해쉬 구조도

5. COPINO 전자문서의 서비스별 보안 메커니즘의 구현

COPINO 전자문서란 COPINO(반입반출계) 선사가 터미널 내에 있는 컨테이너의 인도나 인출을 사전에 통지하여 컨테이너를 탑재한 차량이 게이트를 신속하게 출입하기 위해 사용하는 전자문서이다.

<그림 3>은 COPINO 전자문서 구조도이다.



<그림 3> COPINO 전자문서 구조도

5.1 UN/EDIFACT 보안 메시지

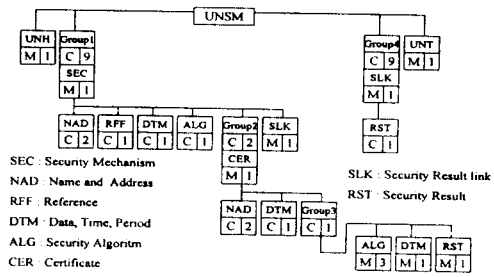
UN/EDIFACT의 보안 필드는 <그림 4>와 같으며 항목을 간단히 설명하며

-. 그룹 2,3은 공개키 암호 알고리즘이 사용될 때 사용하는 항목이다.

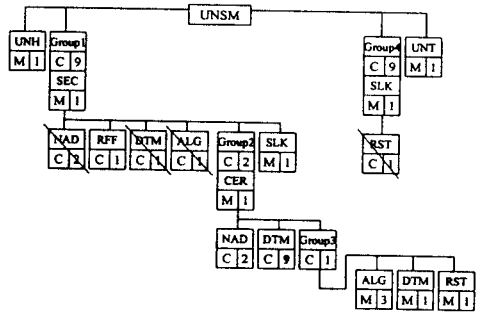
-. 그룹 1의 ALG는 관용암호 알고리즘이나 해쉬 알고리즘을 쓰는 경우에만 관련된다.

-. 공개키 알고리즘을 사용할 때는 그룹 2에 적어도 하나의 CER세그먼트는 존재해야 하며 CER내의 2개의 NAD는 CA와 Certificate Owner(발신자, 수신자)를 나타낸다. 3개의 ALG세그먼트는 공개키 암호 알고리즘을 사용할 경우에 해당되며 사용자의 암호와 알고리즘, CA가 인증서(Certificate)를 생성하는데 이용하는 암호화 알고리즘과 해쉬 알고리즘을 각각 가리킨다.

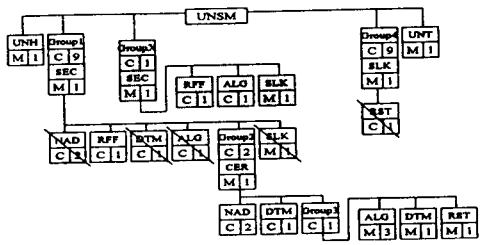
<그림 5>와 <그림 6>은 COPINO 전자문서를 UN/EDIFACT 보안지침서를 참고로 구성한 구조도들로서 <그림 6>에서는 기존 UN/EDIFACT에서 제시된 보안 지침서에서는 제공하지 않는 메시지 보안과 관련된 필드들을 추가로 정의하였다. 또한 <그림 5>와 <그림 6>에서 사용되지 않는 필드는 사선으로 표시하였다.



<그림 4>UN/EDIFACT 보안 메시지 구조도



<그림 5> 전자서명 구조도



<그림 6> 전자서명 + 대칭키 기밀성 구조

<표 1>은 실제 공개키 전자서명을 나타내는 무결성 보안 메시지의 항목을 나타내고 <표 2>는 <그림 6>의 추가된 부분(GroupX)에 해당하는 UNSM 부분의 기밀성을 나타내고 있다.

5.2 보안메커니즘 구현

본 논문에서 제시한 보안 메커니즘의 구현환경은 다음과 같다.

-. Hardware : Micro sparc II, 32Mbyte
-. O/S : Solaris 2.5
-. Compiler : GCC
-. RSA 전자서명의 메시지는 226byte의 크기를 사용 (n : 84bit, 소수 p : 40 bit, 소수 q : 44bit)
-. 공개키, 비밀키 : 각각 84bit

<표 1> 무결성 보안 메시지

UNH+326+COPINO:1:921:KN	UNH : 전자문서머리
	326 : 참조번호
	COPINO : 전자문서명
SEC+NRO+NAK+HEX+AS8	SEC : SECURITY MECHANISM
	NRO : 발신인 부인 방지를 서명
	NAK : MESSAGE 수신인 확인
	HEX : 암호알고리즘 키를 위해 사용
	AS8 : ASCII 8bit
RFF+SSN:001	Message 순서
CER+0000001++HEX+AS8+931+PK1++ DAT:2B:COM:3A:SEG:27:REL:3F	CER : CERTIFICATE
	0000001 : 인증기관의 인증서 ID
	HEX : FILETER CODE
	AS8 : CODE PAGE
	931 : VERSION
	931 : VERSION
	PKI : 인증발급자 KEY NAME
	DAT-3F : 서명을 위한 분리자
NAD+OW+++KOREA	NAD : NAME ADDRESS
	OW : 인증자 보유자
	KOREA : 인증자 보유자 이름
NAD+AX+++PKUN	AX : 인증서 작성자
	PKUN : 인증서 작성자 이름
DTM+273:9805021310:717	DTM : DATE/TIME/PERIOD
	273:9805021320:717 인증유효기간
ALG+OSG:20+EXP:001001:MOD: CA056F9C89:MLN:0512	OSG : 발신지
	20 : RSA
	EXP : 지수
	001001 : 공개키
	MOD : 모듈
	CA056F9C89 : 공개키 모듈값
	MLN : 모듈길이
ALG:IHA:52	IHA : 발급인
ALG+ISG:20+EXP:00010001:MOD: FC5959E40:MLN:512	ISG : 비대칭알고리즘사용신호
	20 : RSA
	EXP : 지수
	001001 : 공개키
	MOD : 모듈
	FC5959E40 : 공개키 모듈값
	MLN : 모듈길이
DTM+CGT:9805021310:202	인증서 확인기관에서 인증일시
RST+C80209FC7B	RST : RESULT
	C80209FC7B : 서명값
SLK+S+1	LINK 부분 START
UNSM	원문부분
SLK+T+1	LINK 부분 TAIL
UNT+25+326	전자문서꼬리

<그림 8>은 <그림 7>의 EDI문서화일(edi.dat)을 MD5와 SMDx기법으로 각각 해쉬한 결과를 나타내고 있다.

<표 2> 기밀성 보안 메시지

SEC+ZZZ	SEC : 암호 메카니즘
	ZZZ : UNSM부분 기밀성
RFF+SSN:002	RFF : REFERENCE
	SSN :002 2번째순번
ALG+ZZZ:21+DES:KEY-VALUE	ZZZ:21 기밀성 사용알고리즘
	DES:KEY-VALUE 초기값

```
BGM+655::KE:COPINO+9802101156+9+NA'
TDT+1+KY187692+31'
LOC+60+KRPUS+:::PUSAN'
DTM+137:980210:101'
EQD+CN+TRLU4011206+4200++6+5'
REF+ACE:2'
MEA+WT++KGM:06700'
SEL+1842253+CA'
TDT+20+24+++ZCS:172:20+++:::ZIT0₁'
LOC+12+USNYC'
CNT+16:1:CH'
```

<그림 7> EDI 문서화일(edi.dat 화일)일부

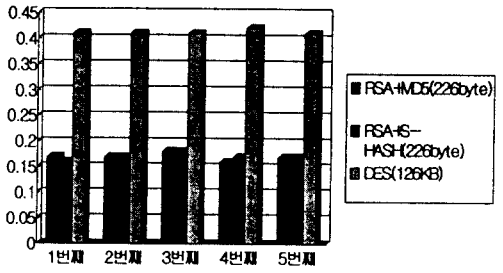
```
[cipher:/user1/lis/rsa/rsa]# s-sh edi.dat

messagefile (binary): edi.dat
hashcode: 709cd9ed475ed7b52caf6ab536de08809428cab6
[cipher:/user1/lis/rsa/rsa]# mdx edi.dat
1d9b9962a9735d5716c95c1fe86b7b91
[cipher:/user1/lis/rsa/rsa]#
```

<그림 8> 해쉬값 생성화면

<그림 9>의 DES암호화(UNSM 자체의 암호화)는 속도비교에서 기존의 입력 데이터로는 수행 시간을

검사하지 못할 정도로 짧은 시간이 걸리므로 126Kbyte의 문서를 암호화하였다.

시뮬레이션을 수행한 결과 값은 <그림 9>와 같이 수행 속도를 5회 측정한 결과 MD5나 제안된 SMDx 기법이 수행 속도면에서는 차이가 없음을 알 수 있었다.



<그림 9> 시뮬레이션 결과 도표

6. 결 론

현재 상용화 되고 있는 전자문서는 평문으로 자료가 전송되고 있어 제3자의 불법적인 접근이 항시 가능하므로 자료의 불법적인 접근, 도청, 수정, 삭제, 재전송, 삽입, 순서변경, 메시지 송수신 사실의 부인 등의 위험이 있다. 본 논문은 UN/EDIFACT에서 제시한 보안 표준안을 기초로 평문 COPINO 전자문서의 전송항목을 보안기법을 추가하여 재구성하였다. 전자서명 기법으로 인증(데이터 무결성) 문제를 해결하였으며 이 전자서명을 효율적으로 하기 위해 해쉬 알고리즘을 사용하는데 기존의 MD5와 제안된 SMDx 기법을 각각 적용하여 비교 분석하였다. 더 나아가 UNSM부분 자체를 DES 암호화 알고리즘을 사용하여 보다 더 안전한 메커니즘을 설계 및 구현하였다. 시뮬레이션 결과, MD5나 SMDx 기법 모두 수행 속도면에서는 차이가 없고 안전성 측면에서는 SMDx가 유리하므로 기존 MD계열의 해쉬함수를 대신하는 방안으로 고려할 수 있다. 또한 기존 메커니즘에 기밀성 부분을 추가하였을 경우(데이타 크기는 126KByte)에 수행 속도면에서 기밀성을 제공하지 않았을 때와 차이가 거의 없으므로 안전한 EDI 자료 전송을 위해 제안된 보안 메커니즘을 적용하는 것이 효율적이라 사료된다.

참고문헌

[1] 임승택, EDI통신과 보안, 대청출판사, 1997.

[2] Sang Uk Shin, Kyung Hyune Rhee, Dae Hyun Ryu, Sang Jin Lee, "A new hash function based on MDx-family and its application to MAC",1998, International Workshop on Practice and Theory in public Key Cryptography.

[3] 해운항만청, "EDIFACT 921 DIRECTORY", 1994.

[4] 도필락, 이혜주, 박지환, "컨테이너EDI데이터를 위한 압축 알고리즘의 설계 및 구현", 대한전자공학회, 한국통신학회, 부산경남지부 1997년도 추계 합동학술 발표회 논문집, pp107-112, 1997.12.

[5] Paul Christmas, "EDI implementation and Security", Elsevier Advanced Technology, 1994.

[6] Man Yong Rhee, Cryptography and Secure Communication, McGrowHill. 1997.

[7] E.Biham, A.Shamir, "Differential cryptanalysis of DES-like cryptosystems", Advances in Cryptology- Crypto'90. Lecture Notes in Computer Science, vol.537, Springer-Verlag, pp2-21, 1991