EPCglobal Network 기반 안전한 메시지 전송 기법의 설계 및 구현

김정재*, 이종희*, 한정훈*, 안재명*, 이종태**
*(주)리테일테크, **동국대학교 산업시스템 공학과
e-mail:argniss@retailtech.co.kr

Design and Implementation of The Method for Secure Message Transfer base on EPCglobal Network

Jung-Jae Kim*, Jong-Hee Lee*, Jung-Hoon Han*,
Jae-Myung Ahn*,Jong-Tae Rhee**
*Retailtech Co., LTD.
**Dept of Industrial and Systems Engineering,
Dongguk University

요 약

본 논문에서는 EPCglobal Network 기반에서 시스템간의 메시지를 안전하게 암호화하여 전송하고 복호화 하는 방법을 제안한다. 제안하는 시스템은 AES 대칭키 암호화 방법을 통해 메시지를 암호화 하고, PKI의 X.509 암호화 기법인 주체자의 개인키를 통해 전자서명을 하여 위변조를 판별하며, AES에 사용된 키는 대상자의 인증서에 포함되어 있는 공개키를 통해 전송해 해준다. 특히 여러 사용자들에게 동시에 편리성과 안전성을 보장할 수 있게끔 하기 위하여 인증서 발급 및 메시지 전송은 웹서비스 기반의 SOAP 메시지를 사용하여 처리한다.

1. 서론

EPCglobal 아키텍처 프레임워크는 Electronic Product Code(EPC)를 사용하여 공급/유통망 강화라는 공통 목표를 위해 서비스되는 모든 것, EPCglobal과 위임기관이 운용하는 코어 서비스(EPCglobal Core Services)와 데이터 인터페이스, 소프트웨어, 하드웨어의 관련 표준(EPCglobal Standard)의 종합으로, EPCglobal 가입자가 EPCglobal과 EPCglobal 아키텍처 프레임워크 요소를 사용하는 개별 가입자와 상호작용하는 시너지 효과를 비공식으로 "EPCglobal Network"라고도 한다. 최근 이와 같은 EPCglobal Network의 중요성과 RFID(Radio Frequency IDentification) 기술 및 응용에 대해 활발한 연구가 이루어지고 있으며, 유통 및 물류 분야뿐 아니라 각 산업 분야에서 다양한 응용 시스템들이 제안되고 있다[3]. 본 논문에서는 EPCglobal Network를 구성하는 웹서비스에 대해 메시지를 안전하게 전달하고, 보내온 메시지가 유효한 메시지인제 검증하는 방법, 안전하게 전달하기 위한 암호화 방법, 시스템간의 인증기법을 제안한다. 본 논문의 구성은 다음과 같다. 2장은 관련연구로서 EPCglobal Network에 대한 전반적인 개념 및 암호화 방법을 소개하고, 3장에서는 제안하는 시스템 암호화 방법, 4장에서는 구현 방법, 5장에서는 결론 및 향후 연구 방향을 제시한다.

※ 본 논문은 산업자원부의 2006년도 성장동력기술개발사업의 지원으로 이루어졌음.

2. 관련연구

2.1 EPCglobal Network

EPCglobal Network는 EPC 코드와 RFID 기술을 근간으로 비즈니스(거래) 파트너 간에 정보교환을 위해 교환할 데이터 구조 및 의미, 물리적 객체(상

품, 박스, 팔레트 등)를 교환하기 위해서 객체에 EPC(Electronic Product Code)를 할당하는 방법에 대한 표준을 제공한다. 기업들은 EPCglobal Network 상에서 발생하는 데이터를 각 기업과 기관의 방화벽 안에서 개별적으로 관리하고, 이 데이터는 ONS(Object Naming Service)와 Discovery Service를 통해 공유하는 방식으로 운영된다[2]. [그림 1]은 EPCglobal Network에서 발생된 데이터의 흐름과 적용되는 표준을 명시하고 있는 개념도이며 개념도상의 각 구성에 대한 설명은 다음과 같다.

- 각각의 기업은 프록시 안에서 독립적인으로 구성
- 상품에 부착되는 EPC는 Tag Data Standard에 의해 정의
- EPC와 리더기는 Gen2 Air Interface Protocol을 통해 통신
- 미들웨어는 ALE를 통해 Filtering과 Collection으로 이루어짐
- EPCIS간의 통신은 인증을 통해 이루어짐
- Event Registry에 이벤트 정보를 등록하고 ONS를 포함하는 Discovery Service를 통해 각각의 EPCIS로의 검색이 이루어짐

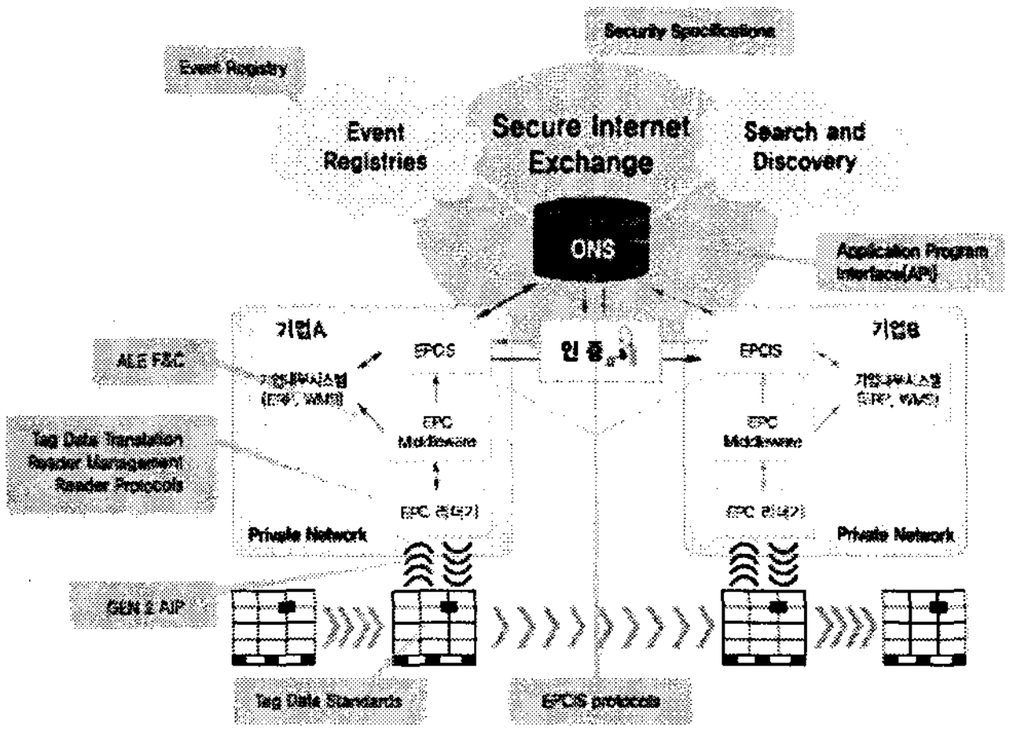

그림 1. EPCglobal Network 흐름 개념도

2.2 AES 암호화 방법

AES 암호 알고리즘은 대칭키 블록 암호 알고리즘으로 암호화/복호화의 기본 단위인 블록의 길이를 128 비트로 하며 키 길이를 128, 196, 256 비트 중에서 선택할 수 있는 알고리즘으로 키 길이에 의해 라운드가 결정이 된다[5]. AES 암호 알고리즘의 암호화 연산은 SubByte 연산, ShiftRow 연산, MixColumn 연산, AddRoundKey 연산 등으로 구성된다. [그림 2]는 AES 암호화 알고리즘의 흐름도이다.

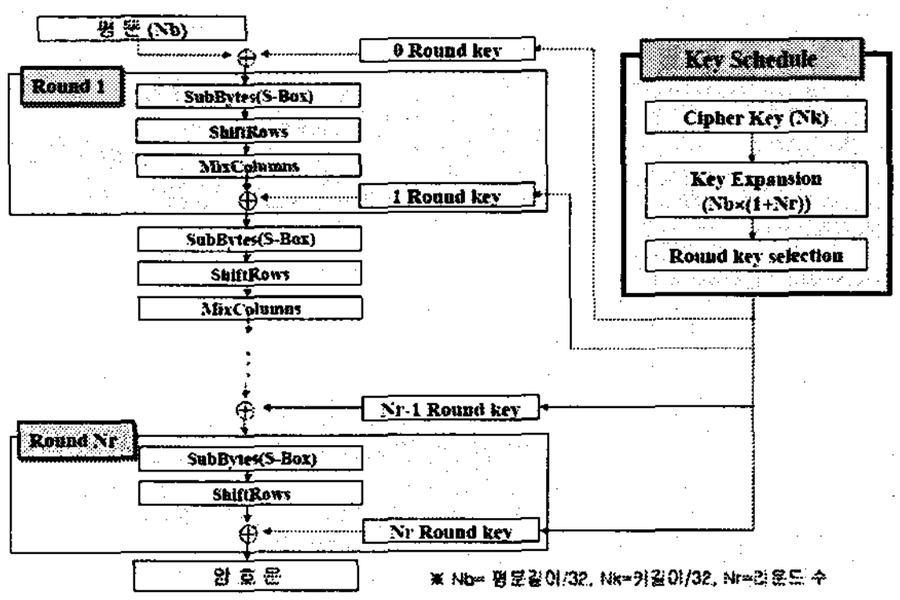

그림 2. AES 알고리즘 흐름도

2.3 PKI 암호화 시스템

PKI는 공개키 암호기술이 안전하게 적용될 수 있는 기반구조로써 공개키와 그 소유자를 연결해 주는 전자 증명서, 키와 인증서를 안전하게 관리해주는 서비스, 그리고 인증서의 유효성 여부를 확인할 수 있는 구조라고 정의한다[1, 4].

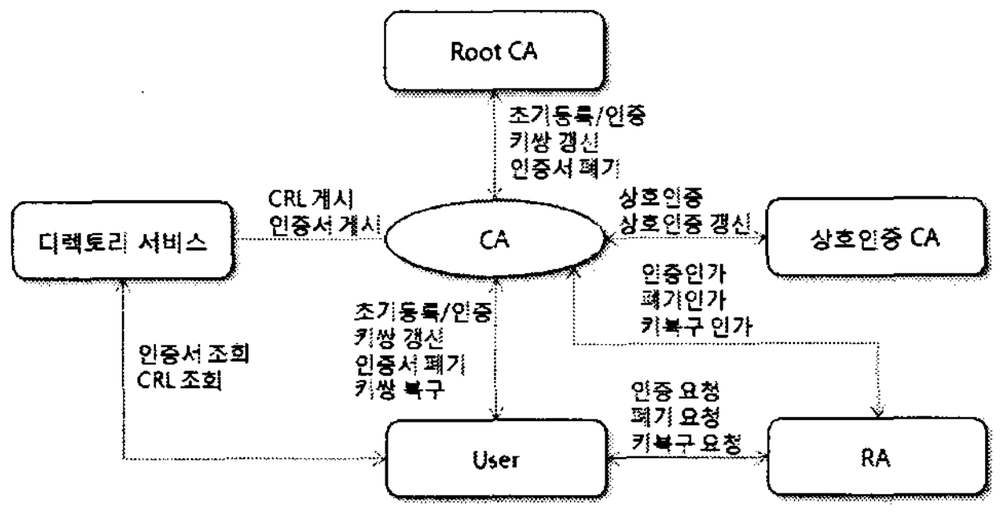

그림 3. PKI 구성요소

3. 제안하는 시스템 암호화 방법

3.1 메시지 암호화 과정

EPCglobal Network에서 각각의 기업간의 전송된 데이터를 통해 해당 데이터베이스에 Select, Insert, Update, Delete 간의 작업이 일어나게 되며, 만약 허위정보가 들어왔을 때에는 더 이상 데이터를 신용할 수 없는 데이터이기 때문에 시스템간의 인증과정과 중간의 공격자에 의해 전송한 메시지의 무결성을 검증해야 한다. 본 논문에서 제안하는 암호화 메시지를 통해 시스템간의 인증 및 무결성, 데이터 암호를 모두 할수 있는 방법을 제시하며, 다음 [그림 3]과 같다.

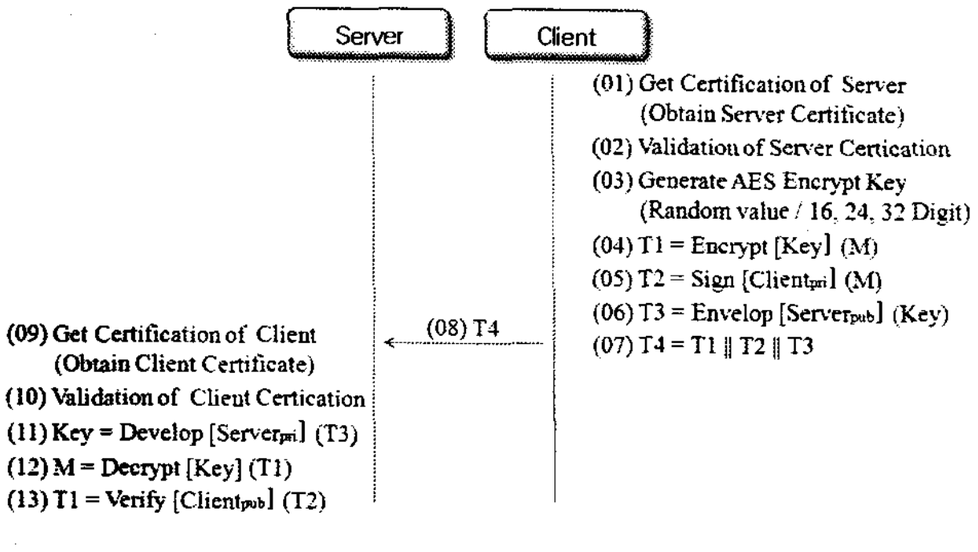

그림 4. 암호화 메시지 전송 방법

Client에서 Server에게 메시지를 전송하기 위해 13번의 단계를 거치며 각 단계별 프로세스는 다음과 같다.

① Server의 인증서를 SOAP 메시지로 획득
② Server의 인증서 상태 검증 (CA에 요청)
③ AES 암호화 Key를 난수값으로 생성 (T1)
④ 전송할 메시지를 AES 암호화 키로 암호화 (T2)
⑤ 메시지를 Client의 개인키로 전자서명 (T3)
⑥ 단계 ①에서 획득한 Server의 인증서 안에 포함된 공개키를 이용하여 데이터 암호화
⑦ 단계 ④⑤⑥의 각각의 값들을 연접
⑧ 단계 ⑦의 값을 SOAP 메시지로 Server로 전송
⑨ Client의 인증서를 SOAP 메시지로 획득
⑩ Client의 인증서 상태 검증 (CA에게 요청)
⑪ Server의 개인키를 통해 T3의 메시지 복호화 (AES 암호화 키 획득)
⑫ AES 암호화 알고리즘을 가지고 메시지 T1 값을 단계 ⑪에서 획득한 암호화 키를 사용하여 복호화 (원문 메시지 획득)
⑬ 단계 ⑨에서 획득한 Client의 공개키를 통해 메시지의 위변조가 없는지 전자 서명값 검사

4. 제안 시스템 구현

본 논문에서는 닷넷 기반의 CA, 클라이언트 및 서버를 구축하고 해당 작업의 요청을 위해 Web Service 방법으로 데이터를 전송 및 전달받게 된다. 구현에 사용된 프로그램은 Microsoft Visual Studio 2005 C#, Microsoft Web Service Enhancements 3.0을 사용하였다.

4.1 CA의 인증서 생성 및 CA Web Service

다음 그림은 EPCCA에서 개인키와 공개키, 그리고 자신의 키로 인증서를 생성하는 인터페이스이다.

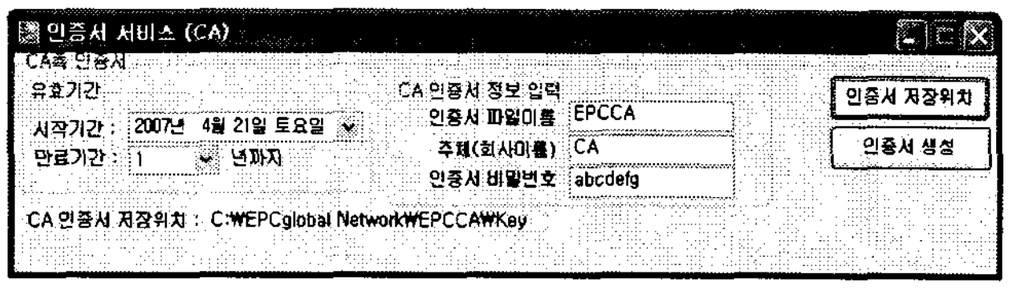

그림 5. CA 인터페이스

인증서를 생성하는 코드는 다음과 같다.

```
// CA의 개인키와 공개키 생성
nRet = Rsa.MakeKeys(CAPublicKeyFileName, CAPrivateKe
yFileName, 1024, Rsa.PublicExponent.Ex p_EQ_65537, 1024,
CACertPassword, Rsa.PbeOpti ons.Default, true);

//CA 인증서 생성
nRet = X509.MakeCertSelf(CACertFileName, CAPrivateKey
FileName, 1, CAExpireYear, "C=KR;O=EPCGlobal Network;
OU=RetailTech;CN=" + CAName + "", "", X509.KeyUsage
Options.DigitalSignature | X509.KeyUsageOptions.KeyCertSi
gn, CACertPassword, X509.Options.FormatPem);
```

다음 코드는 다른 원격지의 컴퓨터에서 공개키를 보내주면 인증서로 생성하여 전송시켜 주는 EPCCA의 Web Service 코드이다.

```
// CA의 개인키와 공개키 생성
nRet = Rsa.MakeKeys(CAPublicKeyFileName, CAPrivateKe
yFileName, 1024, Rsa.PublicExponent.Ex p_EQ_65537, 1024,
CACertPassword, Rsa.PbeOpti ons.Default, true);

//CA 인증서 생성
nRet = X509.MakeCertSelf(CACertFileName, CAPrivateKey
FileName, 1, CAExpireYear, "C=KR;O=EPCGlobal Network;
OU=RetailTech;CN=" + CAName + "", "", X509.KeyUsage
Options.DigitalSignature | X509.KeyUsageOptions.KeyCertSi
gn, CACertPassword, X509.Options.FormatPem);
```

4.2 Client의 인터페이스 및 Web Service

다음 그림은 Client에서 개인키와 공개키를 생성하고 공개키를 인증서서버(EPCCA)의 웹서비스에 요청하게되면 EPCCA에서는 EPCCA 인증서와 개인키를 가지고 전자서명한 후, Client 인증서를 생성하여 다시 Client로 전송해주게 된다.

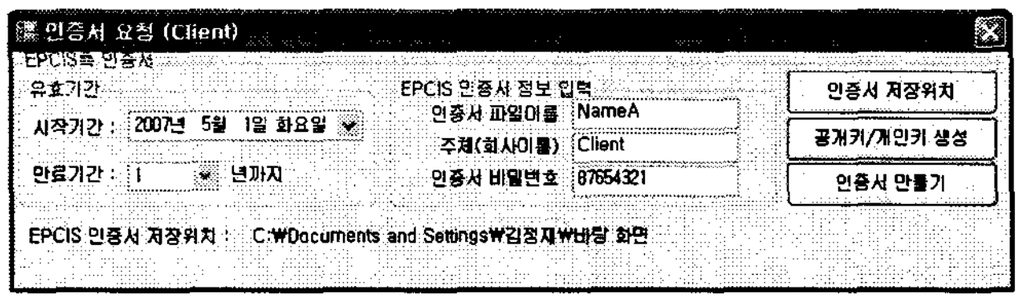

그림 6. Client의 인증서 요청 인터페이스

이때 CA에서 발급된 인증서를 클라이언트에게 전송할 때는 MTOM(Message Transmission Optimizati on Mechanism) 메시지로 전송하게 된다. 다음 그림은 EPCCA에서 Client로 전송하는 SOAP 메시지의 일부분이다.

```
- <soap:Body>
  - <CertMakeFromPubKeyResponse xmlns="http://tempuri.org/">
    - <getFileResponse
        xmlns="http://locahost/WebService/EPCCA">
        <fileName>Client.Cer</fileName>

          <fileData>MIICEjCCAXsCAWYwDQYJKoZIhvcNAQEEBQ
      </getFileResponse>
    </CertMakeFromPubKeyResponse>
  </soap:Body>
```

그림 7. 인증서 요청 SOAP 메시지

다음 그림은 Client 메시지를 Server에게 전송하고 그 결과값을 다시 Client가 다시 복호화하는 인터페이스이다.

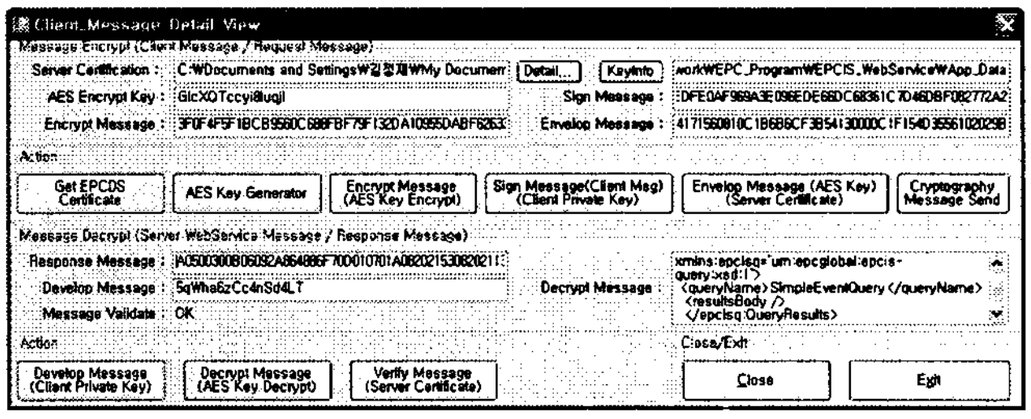

그림 8. Client 암호화 인터페이스

4.3 Server Web Service

다음 그림은 Client에서 Server로부터 메시지를 전송해 주는 SOAP 메시지이다.

<AES_Encrypt_Msg>의 엘리먼트는 본 논문의 3.2 Message 암호화 전송과정에서의 T1에 해당하는 메시지이며, 원문 메시지가 AES 암호화 알고리즘으로 암호화 된 메시지이다. <Sign_Msg> 엘리먼트는 T2에 해당하는 메시지이며, 원문 메시지를 해쉬한 값으로 원문메시지를 복호화 하였을 때, 위변조가 없는지 확인하기 위한 값이다. <Envelop_Key> 엘리먼트는 T3에 해당하는 메시지로, Server의 공개키를 가지고 암호화 시킨 값으로 구성되어 있다.

```
- <soap:Body>
  - <EncryptMsgResponse xmlns="http://tempuri.org/">
      <SystemName>Client</SystemName>

        <AES_Encrypt_Msg>C4228209C2DD16E397A98C78A10560

        <Sign_Msg>3082034D06092A864886F70D010702A082033

        <Envelop_Key>424B18D9B1444E224C2B710CC69112FE5C
    </EncryptMsgResponse>
  </soap:Body>
```

그림 9. Server가 전송받은 SOAP 메시지

Server에서 Client로 SOAP 메시지를 전송하기 위해서 T1, T2, T3 메시지를 복호화 하여 처리한 후, 다시 새로운 T1, T2, T3 메시지를 생성하여 Client로 보내주게 된다.

```
- <soap:Body>
  - <EncryptMsgResponseResponse
      xmlns="http://tempuri.org/">
    - <getEncryptResponse
        xmlns="http://localhost/WebService/Server">

          <AES_Encrypt_Data>V70J9LyXc39zgqjf8ZNNA4Y2pbv5G9gIGw

          <EPCDS_Sign_Data>MIIDWgYJKoZIhvcNAQcCoIIDSzCCA0cCAQ

          <Envelop_Data>Cay3IQrAGVghDgf1i8szQSqvfp3copAMePjr0+u
      </getEncryptResponse>
    </EncryptMsgResponseResponse>
  </soap:Body>
```

그림 10. Client로 전송할 SOAP 메시지

5. 결론

본 논문에서는 EPCglobal Network상에서 각 시스템간의 데이터 통신시에 안전하게 메시지를 보낼 수 있도록 할 수 있는 암호화 시스템을 제안 및 구현하였다. 이는 EPCglobal Network에서 Spec으로 나오는 X.509의 기본 원칙을 구대로 지켰을 뿐 아니라 메시지 복호화시 대량의 데이터를 속도 향상을 위해 대칭키를 사용하여 암호화 하였으며, 여기에 사용된 키만을 개인키로 암호화 시켰으며, 데이터의 무결성을 위해 메시지를 전자서명을 통해 수정하는 것을 방지하였다. 또한 시스템간의 인증을 하기 위해 인증서를 통해 서로 인증할 수 있도록 하였다.

참고문헌

[1] 김정재외 4명, "서명자의 신원정보 해쉬값을 이용한 실시간 인증서 상태 검증 메커니즘의 설계," 한국정보처리학회지 논문지C. Vol.13-C No. 02, 147~154pp., 2006. 04

[2] EPCglabal network기반의 RFID 기술 및 활용, 글로벌

[3] EPCglobal, http://www.epcglobalinc.org, 2006.

[4] John Linn, "Trust Models and Management in Public Key Infrastructures," Technical Notes and Reports of RSA Laboratories, November 2000.

[5] http://www.rsasecurity.com