情 報 保 護 學 會 誌
第 11 券 第 3 號, 2001. 6

인증서 관리 프로토콜(CMP)의 최근 동향에 관한 고찰

임 양 규*, 편 석 진*, 장 우 진**, 원 동 호*

요 약

최근 공개키 암호 시스템의 사용이 급증하면서 공개키의 무결성과 신뢰성 문제를 해결하고자 등장한 인증서기반의 공개키 기반 구조(PKI)와 관련된 다양한 응용 프로토콜에 관한 연구가 활발히 진행되고 있다. 본 고에서는 PKI 응용프로토콜 중에서 PKI 개체들간의 인증서관련 메시지 교환 표준인 인증서 관리 프로토콜(Certificate Management Protocols)을 분석하였다. CMP에 관련된 문서는 IETF에서 표준화한 RFC2510과 2001년 2월에 제안된 드래프트가 있는데, 본고에서는 RFC2510을 중점적으로 분석한 후, 드래프트에서 새로 제안된 부분을 추가하여 두 문서를 비교 분석하였다.

Ⅰ. 서 론

최근 통신기술의 발달로 이메일의 전송이나 전자상거래와 같은 서비스를 인터넷을 통해 이용하고 있다. 그러나, 이들 서비스 이용시 전송 데이터에 암호 기술을 적용하지 않고 평문 그대로 전송하게 되면, 개인 정보의 노출 등의 위험이 따르므로 안전한 암호기술을 적용해야 한다. 이 문제를 해결하기 위해 암호 기술 중 공개키 암호 시스템[1]을 사용하는데, 이는 다시 공개키의 신뢰성 및 무결성에 관한 문제를 내포하고 있다. 이에 대한 해결책으로 제시된 것이 신뢰기관인 인증기관이 서명한 인증서를 사용하는 인증서 기반의 공개키 기반 구조(Public Key Infrastructure:PKI)이다. PKI에 관한 응용 프로토콜은 다양한 워킹그룹에서 연구가 활발히 진행중이며 IETF(Internet Engineering Task Force)나 ISO등에서 표준으로 제정하고 있다.

본 고에서는 PKI의 응용 프로토콜 중에서 PKI 개체인 인증기관(Certificate Authority:CA), 등록기관(Registration Authority:RA), 최종 개체(End Entity:EE)간의 인증서관련 온라인 메시지 전송 표준으로 1999년 3월에 IETF에서 표준화한 RFC 2510[2]을 중점적으로 분석하였으며, 2001년 2월에 제안된 드래프트[3]는 새로 제안된 부분에 대해서 분석한 후 두 문서를 비교 분석하였다.

본 고의 구성은 다음과 같다. II장에서는 PKI 관리 요구사항 및 관리를 함에 있어서의 가정 및 제약사항에 대해서 알아보았으며, III장에서는 실제 전송시 사용되는 데이터의 구조에 대해서 조사하였다. IV장에서는 PKI관리 기능이 필수적으로 가져야 할 기능에 대해서 분석하였으며, V장에서 결론을 맺는다.

II. 요구사항 및 제약 사항

인증서 관리 프로토콜을 설계하기 위해 프로토콜 설계시 요구되는 키 업데이트 및 개인 키(Private key) 소유의 증명 등 다양한 요구사항에 관하여 알아본 후, 요구사항을 만족시키는 PKI 구성요소간의 메시지 교환 동작을 상위 레벨로 분류하였으며, 분류 기준을 토대로 실제 프로토콜에 맞게 각 개체의 기능을 가정 및 제약하였다.

1. PKI관리 요구사항

프로토콜 설계시 다양한 요구사항을 반영해야 하는데, 그 요구사항은 다음과 같다.

① PKI 관리는 ISO9594-8[4] 표준과 그와 관련된 표준을 따라야 한다. 즉, 인증서는 X.509

* 성균관대학교 전지 전자 및 컴퓨터공학과 정보통신보호 연구실(yklim@dosan.skku.ac.kr)
** (주) 싸이버텍 홀딩스

의 형식을 준수해야 한다.

② PKI 관리는 PKI 관리와 관련되는 여타의 표준들을 준수해야 한다.

③ 키 업데이트 시, 다른 용도의 키에 영향을 주지 않고 해당하는 키만 업데이트 가능해야 한다. 이 요구사항은 다음과 같은 경우에 적용된다. 사용자가 서명 키, 암호 키 및 키 동의 키 쌍을 소유하고 있을 경우, 서명 키의 손상으로 인해서 서명 키 만을 업데이트 하고자 할 경우, 다른 용도로 사용하는 암호 키 및 키 동의 키는 변경하지 않고, 손상된 서명용 키만 업데이트 하고자 할 경우에 사용된다.

④ PKI 관리 프로토콜은 인증기관간의 규정문제를 고려해야 한다.

⑤ PKI 관리 프로토콜은 RSA, DSA, MD5 및 SHA-1과 같은 다양한 산업 표준 암호 알고리즘의 사용이 가능해야 한다.

⑥ PKI 관리 프로토콜은 PKI 구성요소의 키 생성을 방해해서는 안된다. 즉, 최종개체, 등록기관 및 인증기관 어느 개체나 키 쌍을 생성할 수 있다.

⑦ PKI 관리 프로토콜은 인증서의 공표(publication)를 지원해야 한다.

⑧ PKI 관리 프로토콜은 인증서 폐지목록(Certificate Revocation List:CRL[5])의 생성을 지원해야 한다.

⑨ PKI 관리 프로토콜은 mail, http, ftp등의 전송 기술을 이용할 수 있어야 한다.

⑩ 인증서 생성의 최종 권한은 인증기관에게 있으며, 최상위 인증기관만이 인증서 필드의 값을 운영 정책에 따라서 변경/추가/삭제/확장 필드 변경 등을 할 수 있다.

⑪ 인증기관의 키 업데이트를 지원해야 한다.

⑫ 등록기관의 기능을 구현 환경에 따라서 인증기관이 대체할 수 있다.

⑬ 최종 개체는 인증기관 혹은 등록기관에게 개체 인증서 요청시 사용된 공개키와 그에 대응하는 개인키의 소유를 증명할 수 있어야 한다.

위의 요구사항을 만족시키는 PKI 구성 요소간의 메시지 교환 동작을 상위 레벨로 분류해 보면 표 1과 같다.

2. 가정과 제약사항

PKI 구성 요소간의 상호 작용을 정의하기 위해서는 구성요소들의 기능과 동작에 대한 가정과 제약이 필요하게 되는데, RFC2510에서는 최종 개체의 초기화, 최종 개체의 초기등록 및 인증, 개인 키 소유의 증명 및 최상위 인증기관의 키 업데이트에 대해서 가정과 제약사항을 두고 있다.

2.1 최종 개체의 초기화

최종 개체가 PKI 관리개체들과의 상호작용을 위해서 PKI가 제공하는 기능에 관한 정보를 요청할 수 있으며, 이는 연관된 최상위 인증기관의 공개키를 안전하게 획득하려는 과정이다.

〔표 1〕 PKI 구성요소의 동작 상위 레벨

동작	분 류	설 명
CA 설립	·	새로운 CA설립시 초기 CRL의 생성과 인증기관 공개키의 export 등의 여러 가지 단계가 필요
EE 초기화	·	Root CA의 공개키를 import하고 PKI 관리 개체가 지원하는 선택사항들에 관한 정보를 요청하는 작업
인증	초기등록/인증	EE가 처음으로 CA나 RA에게 알리는 과정
	키 쌍 업데이트	EE의 주기적인 키 쌍 갱신
	인증서 업데이트	인증서의 사용기간이 만료되면 새로 발급
	CA 키 쌍 업데이트	EE와 다른 메커니즘을 사용한 CA 키 쌍 갱신
	상호 인증 요청	한 CA의 다른 CA에 대한 상호인증서의 발행 요구
인증서/CRL 공표	인증서 공표	인증서 생성후의 공표
	CRL 공표	CRL 생성후의 공표
복구	키쌍 복구	사용자의 비밀키 분실시 복구
취소	취소 요청	인가된 사람이 인증서 취소를 요구하는 비정상적인 상황을 알림
PSE	·	Personal Security Environment의 동작

2.2 초기등록 및 인증

최종 개체의 초기 등록 및 인증을 하는 스킴은 인증기관의 정책에 따라서 다양한 스킴이 존재하지만,

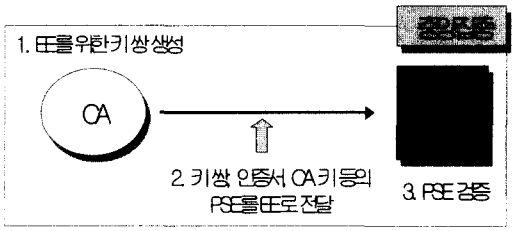
문서에서는 최종 개체가 PKI와 사전접촉이 없었다는 가정하에 초기 등록 및 인증 스킴을 분류한다.

분류에 사용된 기준은 초기 등록을 요청하는 구성요소, 최종 개체의 메시지 출처 인증, 키 생성 장소 및 인증의 성공을 확인하는 메시지 여부 등에 따라서 중앙 집중 스킴과 기본 인증 스킴으로 분류된다. 두 스킴의 차이점은 표 2와 같다.

〔표 2〕 중앙 집중 스킴과 기본 인증 스킴의 비교

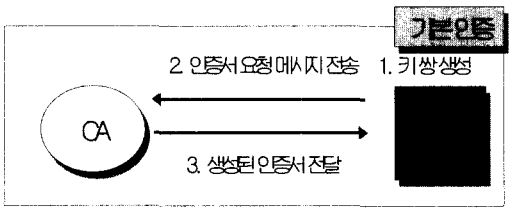
	중앙 집중 스킴	기본 인증 스킴
초기화 발생 장소	CA	EE
메시지 인증여부	필요 없음	필요
키 생성 장소	CA	EE
확인 메시지	필요 없음	필요

중앙 집중 스킴의 경우 초기 등록 및 인증의 과정은 다음과 같다. 인증기관이 최종 개체를 위한 키 쌍을 생성한 후에 키 쌍, 인증서 및 인증기관의 공개키 등의 PSE(Personal Security Environment)를 최종 개체에게 전달하면, 이를 받은 최종 개체는 PSE를 검증해서 초기화 과정을 마친다. 그림 1은 중앙 집중 스킴을 나타낸 것이다.



〔그림 1〕 중앙 집중 스킴

기본 인증 스킴의 경우 초기 등록 및 인증 과정은 중앙 집중 스킴의 과정과는 다르다. 최종 개체가 키 쌍을 생성한 후, 인증서 요청 메시지를 인증기관에게 전송하면 이를 받은 인증기관은 요청 메시지를 검증하고 처리한 후 응답 메시지를 생성하여 최종 개체에게 전송한다. 최종 개체는 응답 메시지를 처리한 후 확인 메시지를 인증기관에게 보내주는데 확인 메시지 검증이 실패하면 인증기관은 인증서를 폐지하여야 한다.



〔그림 2〕 기본 인증 스킴

그림 2는 기본 인증 스킴을 나타낸 것이며, RFC2510에서는 기본 인증 스킴의 사용을 권장한다.

2.3 개인키 소유의 증명

최종 개체와 관련된 키 쌍의 유효성을 검사하기 위해서 Proof of Possession of Private Key (POP)를 하게되는데, 여러 가지 비표준 프로토콜이 쓰이고 있기 때문에 인증기관이나 등록기관은 강제적으로 POP를 제공해야 한다. POP는 키의 용도에 따라서 서명용 키에 대한 POP, 암호화 키에 대한 POP 및 키 동의 키에 대한 POP의 세 가지 형태가 있다.

· 서명용 키

서명용 키에 대한 POP를 위해서 최종 개체는 POP를 요청한 최종 개체에게 특정한 값에 서명을 해서 전송함으로써 서명용 키에 대한 POP를 할 수 있다.

· 암호용 키

암호용 키에 대한 POP를 하기 위한 방법으로는 크게 두 가지 방법이 있다. 한 가지 방법은 개인 키를 인증기관이나 등록기관에게 전송해 주는 방법이 있을 수 있고, 다른 방법은, 특정한 값을 복호화 해서 전송함으로써 POP를 할 수 있다. 개인 키를 인증기관이나 등록기관에게 전송해 주는 방법은 위험요소가 많기 때문에 실제적으로는 잘 사용하지 않고, 특정한 값을 복호화 하는 POP방법의 사용을 권장한다. 특정한 값을 복호화 해서 POP를 하는 방법은 다시 직접적인 방법과 간접적인 방법으로 분류된다. 직접적인 방법은 인증기관이나 등록기관이 최종 개체에게 최종 개체의 공개키로 랜덤 수를 암

호화해서 전송하면, 이를 받은 최종 개체는 복호화한 값을 다시 요청한 인증기관이나 등록기관의 공개키로 암호화해서 전송하는 방법이고, 간접적인 방법은 인증서 발행시에 최종 개체의 공개키로 인증서를 암호화한 후 전송함으로써 최종 개체의 인증서 사용여부로 POP를 검증하는 방법이다. RFC2510에서는 특정한 값을 복호화해서 POP를 하는 방법 중에서 후자의 방법을 권장하는데, 그 이유는 간접적인 방법을 사용하면 부가적인 작업 및 메시지가 필요 없기 때문이다.

· 키 동의 키

키 동의 키의 POP를 위해서 최종 개체와 인증기관이나 등록기관 두 개체 사이의 공유키를 설정한 후, 랜덤 수 전송 등의 방법을 통해서 올바른 공유키가 설정되었는지를 확인해 봄으로써 가능하다.

2.4 최상위 인증기관의 키 업데이트

최상위 인증기관의 키 업데이트의 기본 절차는 업데이트 이전의 비밀키를 사용해서 새로 업데이트한 인증서에 서명한 후, 새로 업데이트한 키를 사용해서 업데이트 이전의 인증서에 서명하는 것이다. 그러므로, 인증서의 caCertificate 필드에 OldWIthOld, OldWithNew, NewWithOld, NewWithNew의 속성 값을 생성해야 한다. 그러나, 데이터 구조는 X.509 v1과의 상호호환을 위해서, X.509 v3 구조를 사용하지 않는다.

2.4.1 인증기관 Operator 작업

최상위 인증기관의 키 업데이트를 위한 인증기관 Operator는 다음과 같은 작업을 수행해야 한다.

① 새로운 키 쌍 생성
② 새로운 개인 키로 서명된 기존의 공개키 인증서 생성(OldWithNew)
③ 기존의 개인 키로 서명된 새로운 공개키 인증서 생성(NewWithOld)
④ 새로운 개인 키로 서명된 새로운 공개키 인증서 생성(NewWithNew)
⑤ 디렉토리에 새로운 인증서 공표
⑥ 물리적인 방법으로 새로운 공개키를 획득할 수 있도록 export

위의 작업을 마무리하게 되면 더 이상 기존의 개인 키는 필요 없게되지만, 기존에 보관된 데이터에 대한 검사를 위해서 보관할 수도 있다.

OldWithNew의 사용시간은 Old 키의 생성 시간과 만료 시간을 표시하고, NewWithOld의 사용시간은 새로운 키 쌍 생성시간과 모든 최종 개체가 안전하게 새로운 인증기관 공개키를 획득한 시간을 표시하며, NewWithNew에는 새로운 키 쌍 생성 시간과 다음 번 키 업데이트 시간을 표시해야 한다.

2.4.2 인증서 검증

인증기관의 키가 업데이트되면, 최종 개체의 측면에서는 표 3과 같은 검증자가 서명자의 공개키를 담고 있는 인증서를 검증하기 위한 다양한 상황이 발생할 수 있는데, 각각의 상황을 저장소 접근 여부, 직접 검증 여부 등에 따라서 세부적으로 알아보도록 하겠다.

〔표 4〕 인증서 검증과정에서 발생할 수 있는 경우의 수

		저장소가 New/Old 공개키 저장		저장소가 Old 공개키만 저장	
		PSE		PSE	
		New 공개키	Old 공개키	New 공개키	Old 공개키
서명자의 인증서	New 공개키	***Case 1***	Case 3	***Case 5***	Case 7
	Old 공개키	Case 2	***Case 4***	Case 6	***Case 8***
결론	· Case 1, 4, 5, 8 : 저장소에 접근하지 않고 직접 검증 가능 · Case 2, 3 : 저장소에 접근해서 검증 가능 · Case 6, 7 : 검증 실패				

① Case 1

저장소가 New 공개키와 Old 공개키 모두 저장하고 있고, 최종 개체의 PSE는 New 공개키만 저장하고 있으며 서명자의 인증서가 인증기관의 New 키로 서명된 경우이다. 이 경우에 검증자는 저장소에 접근하지 않고 직접 검증이 가능하다.

② Case 2

저장소가 New 공개키와 Old 공개키 모두 저장하고 있으며, 최종 개체의 PSE는 New 공개키만 저장하고 있고, 서명자의 인증서가 인증기관의 Old 키로 서명된 경우이다. 이 경우에 검증자는 저장소에 접근해서 caCertificate 속성에 있는 OldWithNew 속성 값을 획득한 후 검증 가능하다.

③ Case 3

저장소가 New 공개키와 Old 공개키 모두 저장하고 있으며, 최종 개체의 PSE는 Old 공개키만 저장하고 있고, 서명자의 인증서를 인증기관의 New 키로 서명했을 경우이다. 이 경우에 검증자는 저장소에 접근해서 caCertificate 속성에 있는 NewWithOld 속성 값을 획득한 후 검증가능 하다.

④ Case 4

저장소가 New 공개키와 Old 공개키 모두 저장하고 있으며, 최종 개체의 PSE는 Old 공개키만 저장하고 있고, 서명자의 인증서가 인증기관의 Old키로 서명된 경우이다. 이 경우에 검증자는 저장소에 접근하지 않고 직접 검증이 가능하다.

⑤ Case 5

저장소가 Old 공개키만 저장하고 있으며, 최종 개체의 PSE는 New 공개키만 저장하고 있고, 서명자의 인증서가 인증기관의 New키로 서명된 경우이다. 이 경우에 검증자는 저장소에 접근하지 않고 직접 검증이 가능하다.

⑥ Case 6

인증기관이 저장소를 업데이트 하지 않고, 검증자에게 New 공개키를 저장하고 있는 PSE를 발행했을 경우이다. 이 경우에 검증자는 인증기관의 old 공개키를 신뢰할 수 있는 방법으로 획득할 수 없으므로 검증에 실패한다.

⑦ Case 7

인증기관이 저장소를 업데이트 하지 않고, New 키로 서명된 인증서를 발행했을 경우이다. 이 경우에 검증자가 인증기관의 새로운 공개키를 획득할 신뢰할 수 있는 방법이 없으므로 검증에 실패한다.

⑧ Case 8

인증기관이 저장소를 업데이트 하지 않고, PSE에 Old 공개키가 있고, 서명자의 인증서가 인증기관의 Old 키로 서명된 경우이다. 이 경우에 검증자는 저장소에 접근하지 않고 직접적으로 검증이 가능하다.

위에서 살펴본 8가지의 경우를 정리하면, Case 1, Case 4, Case 5, Case 8의 경우는 저장소에 접근하지 않고, 검증자가 직접적으로 검증이 가능하며, Case 2, Case 3의 경우에는 검증자가 저장소에 접근해서 인증서를 획득한 후 검증이 가능한 반면에, Case 6, 7번의 경우는 인증기관의 키 업데이트 지연으로 인해 검증자의 신뢰할 수 있는 키 획득 방법이 없으므로 검증에 실패한다.

III. 데이터 구조

1. 전체적인 구조

PKI 구성요소들간의 메시지 교환을 위한 PKI 메시지의 데이터 구조는 그림 3과 같이 구성되어 있으며, header, body, protection 등의 필드로 구성되어 있다.

PKIHeader는 PKI 메시지에 사용되는 공통적인 정보를 나타내며, PKIBody는 CMP에서 제공하는 인증서와 관련된 기능을 의미하는 메시지 정보로 구성되어 있다. PKIProtection은 PKI메시지를 보호하는 비트로 구성되어 있으며, extraCerts는 수신자가 이용 가능한 인증서를 나타낸다.

본 고에서는 CMP 데이터 구조의 핵심인 PKIHeader와 PKIBody의 ASN.1[6]에 대해서만 알아보고자 한다.

```
PKIMessage ::= SEQUENCE {
    headerPKIHeader,
    bodyPKIBody,
    protection [0]   PKIProtection OPTIONAL,
    extraCerts [1]   SEQUENCE SIZE (1..MAX) OF
                     Certificate OPTIONAL
}
```

[그림 3] PKI 메시지의 전체 구조

2. PKI 메시지 Header

PKIHeader는 그림 4와 같은 구조로 되어있으며, 각 필드에 대한 설명은 표 4와 같다. pvno 필드가 취할 수 있는 값의 범위가 RFC2510의 경우는 1로 고정되어 있었으나, 2001년 2월 드래프트에서는 CMP1999(1), CMP2000(2)의 값을 갖는다.

```
PKIHeader ::= SEQUENCE {
    pvno        INTEGER     { ietf-version2 (1) },
    sender      GeneralName,
    recipient   GeneralName,
    messageTime [0] GeneralizedTime OPTIONAL,
    protectionAlg [1] AlgorithmIdentifier OPTIONAL,
    senderKID   [2] KeyIdentifier  OPTIONAL,
    recipKID    [3] KeyIdentifier  OPTIONAL,
    transactionID [4] OCTET STRING OPTIONAL,
    senderNonce [5] OCTET STRING OPTIONAL,
    recipNonce  [6] OCTET STRING OPTIONAL,
    freeText    [7] PKIFreeText  OPTIONAL,
    generalInfo [8] SEQUENCE SIZE (1..MAX) OF
                    InfoTypeAndValue OPTIONAL
```

[그림 4] PKIHeader의 구조

〔표 4〕 PKIHeader 필드의 의미

필드 명	설 명
pvno	버전을 표시 RFC2510은 1의 값을 가지며, 2001년 2월 드래프트는 2의 값을 가진다.
sender	송신자의 이름
recipient	수신자의 이름
messageTime	메시지 생성 시간
protectionAlg	protection bit를 계산하는데 사용된 알고리즘
senderKID	송신자가 메시지 보호에 사용한 키의 ID
recipKID	수신자가 메시지 보호에 사용한 키의 ID
transactionID	트랜잭션 식별자
senderNonce	replay attack을 방지하기 위해서 송신자가 선택한 랜덤 수
recipNonce	replay attack을 방지하기 위해서 송신자가 선택한 랜덤 수
freeText	수신자가 읽을 수 있는 텍스트
generalINfo	수신자의 기기가 처리할 부가적인 데이터

```
PKIBody ::= CHOICE {
    ir          [0]  CertReqMessages,
    ip          [1]  CertRepMessage,
    cr          [2]  CertReqMessages,
    cp          [3]  CertRepMessage,
    p10cr       [4]  CertificationRequest,
    popdecc     [5]  POPODecKeyChallContent,
    popdecr     [6]  POPODecKeyRespContent,
    kur         [7]  CertReqMessages,
    kup         [8]  CertRepMessage,
    krr         [9]  CertReqMessages,
    krp         [10] KeyRecRepContent,
    rr          [11] RevReqContent,
    rp          [12] RevRepContent,
    ccr         [13] CertReqMessages,
    ccp         [14] CertRepMessage,
    ckuann      [15] CAKeyUpdAnnContent,
    cann        [16] CertAnnContent,
    rann        [17] RevAnnContent,
    crlann      [18] CRLAnnContent,
    conf        [19] PKIConfirmContent,
    nested      [20] NestedMessageContent,
    genm        [21] GenMsgContent,
    genp        [22] GenRepContent,
    error       [23] ErrorMsgContent,
    certConf    [24] CertConfirmContent
}
```

〔그림 5〕 PKI메시지 Body의 구성

3. PKI 메시지 Body

PKI메시지 전체 구조 중에서 PKI 메시지 Body는 메시지의 요청이나 응답 형태에 따라 여러 가지 필드 중에서 다양하게 선택적으로 사용할 수 있도록 그림 5와 같이 구성되어있으며, 필드 중에서 밑줄로 표시된 마지막 부분인 certConf[24] 부분이 2001년 2월 드래프트에서 추가된 부분이다. certConf 필드는 최종 개체가 인증기관이나 등록기관에게 인증서를 수락하거나 거부하는 확인 메시지를 전송하는데 사용된다. PKIBody의 사용 예는 다음과 같다. 초기화 요청의 경우에 PKIBody에 ir 필드를 사용하고, 초기화 요청에 대한 응답으로는 ip필드를 사용하게 된다. 각각의 필드가 나타내는 의미는 표 5와 같다.

〔표 5〕 PKI Body 필드의 의미

필드 명	설명
ir[0]	초기화 요청
ip[1]	초기화 요청에 대한 응답
cr[2]	인증 요청
cp[3]	인증 요청에 대한 응답
p10cr[4]	PKCS#10 형식의 인증서 요청
popdecc[5]	POP 요청
popdecr[6]	POP 요청에 대한 응답
kur[7]	키 업데이트 요청
kup[8]	키 업데이트 요청에 대한 응답
krr[9]	키 복구 요청
krp[10]	키 복구 요청에 대한 응답
rr[11]	인증서 폐지 요청
rp[12]	인증서 폐지 요청에 대한 응답
ccr[13]	상호 인증 요청
ccp[14]	상호 인증 요청에 대한 응답
ckuann[15]	CA 키 업데이트의 발표
cann[16]	CA의 새로운 인증서 발행 발표
rann[17]	인증서 폐지 발표
crlann[18]	CA가 새로운 CRL을 발행
conf[19]	프로토콜 교환에 사용
nested[20]	메시지를 forward할 때 사용
genm[21]	새로운 CMP요청에 사용
genp[22]	genm에 대한 응답
error[23]	에러 메시지
certConf[24]	인증서의 수락/거부

IV. 필수기능

지금까지 분석한 내용을 토대로 RFC2510과 2001년 2월 드래프트를 비교 분석해서 추가된 부분을 정리하면 표 6과 같다. 새로 추가된 부분은 데이터 구조 중에서 PKIHeader 필드에 버전을 표시하는 pvno 필드 값의 변경이다. 또한, PKIBody 필드에 인증서 수락 여부를 나타내는 필드가 추가된 것이다.

〔표 6〕 드래프트에서 추가된 사항

위치	필드	비고
PKIHeader	pvno	· RFC2510 : 1 · Draft : 2
PKIBody	certConf	새로 추가

2장에서 설명된 기능을 포함한 PKI관리 프로토콜이 필요로 하는 필수 기능은 최상의 인증기관의 초기화 및 키 업데이트, 하위 인증기관의 초기화, 인증서 폐기 목록 생성, PKI 정보요청, 상호 인증, 최종 개체 초기화, 새로운 인증서 요청 및 최종 개체의 키 업데이트의 9가지이다.

1. 최상위 인증기관 초기화

최상위 인증기관을 새로 설립하면, 자기 자신이 서명한 인증서(self-certificate)를 생성해야 하며, 물리적인 수단으로 인증서를 획득하지 못하는 최종 개체가 인증서를 검증할 수 있도록 최상위 인증기관의 공개키에 대한 핑거프린트를 생성해야 한다.

2. 최상위 인증기관 키 업데이트

OldWithOld 인증서나 NewWithNew 인증서를 소유한 최종 개체를 위해서 최상위 인증기관의 키가 업데이트되면, NewWithOld 인증서나 OldWithNew 인증서를 생성해서 인증서 검증을 할 수 있게 해야 한다.

3. 하위 인증기관 초기화

PKI 관리 관점에서 보면, 하위 인증기관의 초기화는 최종개체의 초기화와 같은 과정이나, 하위 인증기관의 초기화시 초기 인증서 폐지 목록을 생성한다는 것이 차이점이다.

4. 인증서 폐지목록(CRL) 생성

인증서를 발행하는 새로 설립된 인증기관은 인증서를 발행하기 전에, CRLEntryList 값이 비어있는 인증서 폐지 목록을 만들어야 한다.

5. PKI 정보 요청

PKI 개체(인증기관, 등록기관, 최종 개체)는 인증기관의 현재 상태에 관한 정보를 인증기관에게 요청해서 획득할 수 있다. PKI 정보에 관한 요청을 받은 인증기관은 PKI개체가 요청한 정보에 관한 모든 정보를 제공해 주어야 하는데, 제공해 주지 못하는 정보에 대해서는 에러 메시지를 전송해 주어야 한다.

6. 상호 인증

CMP는 상호 인증서 생성을 위한 요청, 응답 및 확인 메시지를 제공해 주는데, 상호인증을 통해서 서로 같은 도메인에 있거나 혹은 다른 도메인에 있는 사용자의 인증서에 대한 신뢰를 할 수 있으며, 상호인증은 인증기관 사이에서 수행한다.

7. 최종 개체 초기화

최종개체도 인증기관과 마찬가지로 초기화 과정을 수행하는데, 최상위 인증기관의 공개키에 관한 정보 획득 등의 PKI 정보획득이나 최상위 인증기관의 공개키를 획득한 후 핑거프린트를 물리적인 수단으로 검증하는 두 가지 정도의 단계를 거쳐야한다.

8. 인증서 요청

초기화를 마친 최종 개체는 certification request (cr)를 통해서 개인 정보 업데이트 등의 이유로 인증서를 다시 요청할 수 있는데, 인증기관은 새로운 인증서를 CertRep 메시지를 통해서 발행해 준다.

9. 최종 개체 키 업데이트

최종 개체의 키는 키의 사용기간 만료 등의 이유

로 키를 업데이트 해야 하는데, 그 과정은 다음과 같다. 최종 개체가 key update request(kur) 메시지를 통해서 인증기관에게 키 업데이트를 요청하게 되면, 키 업데이트 요청 메시지를 전송 받은 인증기관은 key update response(kup) 메시지를 사용해서 새로 생성한 인증서를 최종 개체에게 전송해 준다.

V. 결 론

다양한 PKI 응용프로토콜 중에서 인증서 관리 프로토콜은 최상위 인증기관의 초기화 및 키 업데이트, 하위 인증기관의 초기화, 인증서 폐기 목록 생성, PKI 정보 요청, 상호인증, 새로운 인증서 요청 및 최종 개체의 키 업데이트 등의 필수 기능을 제공하도록 구현해야 한다. 본 고에서는 인증서 관리 프로토콜 표준인 IETF의 RFC2510 분석과 2001년 2월에 제안된 드래프트를 비교 분석하였다.

국내 공인 인증기관과 사용자간의 인증서 관리를 위한 인증서 관리 프로토콜도 RFC2510이나 2001년 드래프트를 토대로 구현하면 국제적인 호환성을 유지할 수 있을 것이다.

참 고 문 헌

[1] Diffie M. E. Hellman, "New Directions in Cryptography," IEEE Transactions on Information Theory, Vol.IT-22, 644-654, 1976.

[2] C. Adams, S. Farrell, Internet X.509 Public Key Infrastructure Certificate Management Protocols, IETF standard, RFC2510, Mar., 1999.

[3] C. Adams, S. Farrell, Internet X.509 Public Key Infrastructure Certificate Management Protocols, IETF draft, Feb., 2001.

[4] ISO/IEC 9495-8, Information Technology - Open Systems Interconnection - The directory: Public-key and attribute certificate frameworks, Mar.,2000.

[5] R. Housley, W. Ford, W. Polk, D. Solo, Internet X.509 Public Key Infrastructure Certificate and CRL profile, IETF standard, RFC2459, Jan., 1999.

[6] ISO/IEC 88240-1, Information Technology - Abstract Syntax Notation One (ASN.1): Specification of Basic Notation, Dec., 1997.

〈著者紹介〉

임 양 규 (Yang-Kyu Lim)
1999년 2월 : 국민대학교 정보관리학과(학사)
2000년 3월 ~ 현재 : 성균관대학교 전기전자 및 컴퓨터공학과 석사과정
관심분야 : PKI, 전자상거래 보안, 암호 프로토콜

편 석 진 (Suk-Jin Pyun)
1999년 2월 : 성균관대학교 수학과 졸업(학사)
2001년 2월 : 성균관대학교 수학과 석사
2001년 3월 ~ 현재 : 성균관대학교 전기 전자 및 컴퓨터 공학과 박사과정
관심분야 : 대수학, 정수론, 타원곡선

장 우 진 (Woo-Jin Jang)
1997년 2월 : 아주대학교 수학과 졸업(학사)
1997년 1월 ~ 현재 : 싸이버텍 홀딩스 근무
관심분야 : PKI, 암호이론, 전자지불

원 동 호 (Dong-Ho Won)
성균관대학교 전자공학과 (학사, 석사, 박사)
1978년~1980년 : 한국전자통신연구소 전임 연구원
1985년~1986년 : 일본 동경공대 객원연구원
1992년~1994년 : 성균관대학교 전산소장
1995년~1997년 : 성균관대학교 교학처장
1996년~1998년 : 국가정보화 추진위원회 자문위원
1998년~1999년 : 성균관대학교 정보통신기술연구소 소장
1990년~1999년 : 한국정보보호학회 이사
1999년~2001년 : 성균관대학교 전기전자 및 컴퓨터 공학부장
1999년~2001년 : 성균관대학교 정보통신대학원 원장
1982년~ 현재 : 성균관대학교 전기전자 및 컴퓨터 공학부 교수
1999년~현재 : 한국정보보호학회 수석 부회장
2000년~현재 : 정통부 지정 정보보호인증기술연구센터 센터장
관심분야 : 암호이론, 전자 상거래