情報保護學會論文誌
第13券 第1號, 2003. 2

PMI 인증서 검증 위임 및 검증 프로토콜

이 승 훈*, 송 주 석**

Delegated Attribute Certificate Validation And Protocol

Seung-Hoon Lee*, Joo-Seok Song**

요 약

공개키 인증서와 마찬가지로 속성 인증서 역시 신뢰되기 위해서는 인증서에 대한 유효성 여부를 검증 받아야 한다. 속성 인증서의 유효성 검증은 속성 인증서 체인과 그 체인에 존재하는 각 개체들의 공개키 인증서 체인에 대해 유효성 여부를 검사함으로써 이루어진다. 이렇듯 속성 인증서의 유효성 확인은 공개키 인증서보다 더 복잡하고 많은 처리 과정을 거치게 된다. 이러한 검증 과정은 서비스제공 서버에게는 많은 부담이 될 수 있다. 본 논문에서는 속성 인증서 검증을 효과적으로 대신 할 수 있도록 전문화된 검증 서버의 사용을 제안하고 검증 서버와 클라이언트간에 사용될 검증 프로토콜을 정의한다.

ABSTRACT

PMI(Privilege Management Infrastructure) certificates as well as Public-Key certificates must be validated before being used. Validation for a PMI certificate requires PMI certificate path validation, and PKC(Public-Key Certificate) path validations for each entity in the PMI certificate path. This validation work is quite complex and burdened to PMI certificate verifiers. Therefore, this paper suggests a delegated PMI certificate validation that uses specialized validation server, and defines a validation protocol which is used between validation server and client.

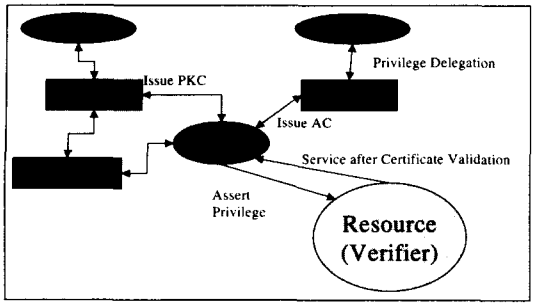
Keyword : *PMI, 속성 인증서, 인증서 검증, 검증 위임, AC, DPV, DPD*

Ⅰ. 서 론

21세기는 지식 기반 정보화 사회라고 불린다. 다양하고 무한한 정보의 저장소이며 소통 수단인 인터넷은 이미 물리적인 기간망인 도로와 철도의 역할을 조금씩 대신하고 있다. 사람들이 직접 만나서 처리해야 할 일들이 온라인 상에서도 가능해지고 있기 때문이다. 이런 온라인 상의 일 처리는 물리적 접촉이 없이도 서로 상대방의 신원을 확인할 수 있어야만 가능하다. 이것을 가능하게 해주는 기반 기술이 PKI (Public-Key Infrastructure) 이다. PKI에서는 사용자의 신원 정보에 대해 신뢰할 수 있는 기관이 서명하여 신원을 보증하게 된다[1,2].

이와 더불어 최근에는 신원뿐만 아니라 사용자의 속성 또는 권한까지도 온라인 상에서 신뢰할 수 있게 하는 기반 기술인 PMI에 대한 연구가 활발히 이루어지고 있다. PMI(Privilege Management Infrastructure), 즉 권한 관리 기반 구조는 사용자에 대한 속성 및 권한 등의 정보에 대해 신뢰할 수 있는 속성 인증기관이 서명함으로서 신원과 속성의 관계를 보증하는 정보 보호 기반 구조이다. [그림 1]과 같이 PMI는 PKI와 연동되어 사용되며 PKI를 통해 신원이 인증된 사

* LG전자/정보통신(zzboy@lge.com)
** 연세대학교 컴퓨터과학과 정보통신 연구실(jssong@emerald.yonsei.ac.kr)



SOA : Source of Authority, AA: Attribute Authority
CA : Certificate Authority, RA: Registration Authority

〔그림 1〕 PKI와 PMI

용자에 대한 속성 정보를 제공하는 기능을 하게된다[1,3].

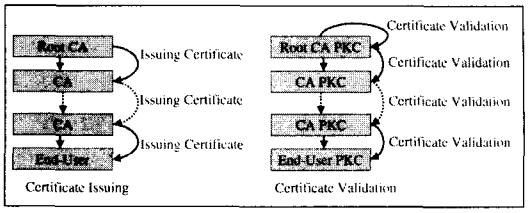
PMI가 널리 사용되면 지금 보다 훨씬 많은 일들이 온라인 상에서 처리되어 편리함과 높은 효율성을 제공하게 될 것이다. 그러나 PMI가 실생활에 널리 사용되기 위해서는 효율성이나 안정성 등에서 많은 개선이 필요하며, 현재 연구가 활발히 진행되고 있는 검증 과정의 효율성 개선 노력은 그 대표적인 예라고 할 수 있다.

공개키 또는 속성 인증서에 담겨진 정보를 신뢰할 수 있기 위해서는 사용하기에 앞서 인증서에 대한 유효성 검증을 필요로 하게된다. 유효성 검증이란 사용하려는 인증서가 올바른 인증서이며 인증서의 내용이 신뢰할 수 있는지를 판단하는 작업을 말한다. 본 논문에서는 속성 인증서에 대한 유효성 검증을 검증 담당 서버에게 위임하는 구조를 도입하여 일반 자원 서버의 검증에 대한 부담을 줄이는 방안을 제시한다. 또한 위임된 검증서버와 검증 클라이언트(자원 서버) 사이의 프로토콜을 정의한다.

II. 인증서 검증 알고리즘

2.1 공개키 인증서에 대한 유효성 검증

공개키 인증서에 대한 유효성 검증은 IETF RFC 3280의 검증 절차를 따른다. [그림 2]는 공개키 인증서에 대한 유효성 검증 과정을 보여주고 있다. 공개키 인증서의 유효성 검증에는 최하위 사용자에서 신뢰된 인증기관까지의 일차원 검증 체인을 필요로 하게된다. 이러한 체인 검증을 위해서 검증 알고리즘에서는 체인 구성에 필요한 인증서들을 획득할 수 있어야 하며, 각 인증서들의 폐지 상태 정보를 확인할 수 있어야 한다. 경로를 구성하는 모든 인증서가 폐지되지 않았고, 유효 기간을 벗어나지 않았으며, 상



CA : Certificate Authority, PKC: Public-Key Certificate

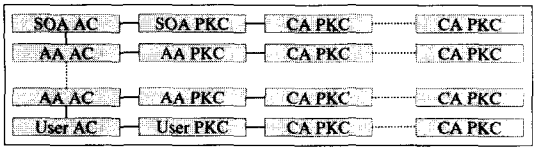
〔그림 2〕 공개키 인증서 발급 및 검증

위 인증기관에 의한 인증서 서명 값이 모두 올바른 값을 가지고 있다면 사용자 인증서는 유효하다고 할 수 있다[1,2].

2.2 속성 인증서 유효성 검증

속성 인증서에 대한 유효성 검증은 ITU-T X.509[1] 및 IETF RFC 3281[3]의 속성 인증서 경로 처리 절차를 따른다. X.509 PMI Framework에서는 일반적인 환경에서의 속성 인증서 유효성 검증을 고려하고 있는 반면 RFC 3281에서는 권한 위임을 사용하지 않는 단순한 경우만을 고려한 검증 절차를 권고하고 있다. 두 표준의 검증 절차를 종합한 검증 과정은 다음과 같다[1,3].

(1) AC(Attribute Certificate) 소유자가 신원 인증을 위해 PKC (Public Key Certificate)를 사용할 경우, AC 검증자는 해당 사용자의 PKC를 찾을 수 있어야 하고, 그 PKC는 RFC 3280의 검증 절차에 따라 검증되어야 한다.
(2) AC 서명이 발행기관의 서명키에 의해 정확한 값으로 서명되어 있어야 하며, 발행자의 PKC 인증 경로가 검증되어야 한다.
(3) 발행자의 AC를 통해 권한 할당 또는 위임이 정당하게 이루어졌는지 검사한다.
(4) AC가 사용되는 시간이 유효 시간 안에 있어야 한다.(일부 예외상황 제외)
(5) AC에 대한 폐지 상태 정보를 이용할 경우 AC는 폐지되어 있지 않아야 한다.
(6) AC가 지원되지 않는 critical 확장을 가지고 있으면 AC는 거절되어야 한다.
(7) AC 발행자가 직접적으로 신뢰되는 기관이 나올 때까지 발행자의 AC에 대해서도 (2)~(6)의 과정을 반복한다.
(8) 더 이상의 상위 발행 기관을 찾을 수 없고 현재 최 상위 인증기관이 직접 신뢰되지 못할 경우 하위 인증서들은 유효하지 않다.



〔그림 3〕 속성 인증서 검증 사각형

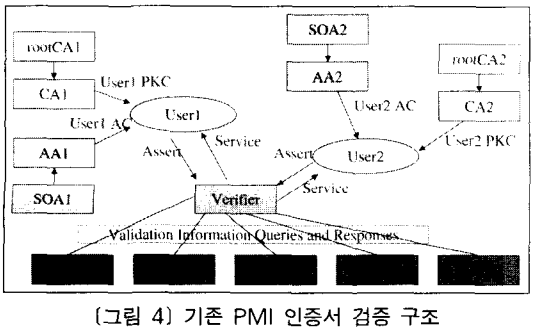
[그림 3]은 속성 인증서 검증 사각형(AC validation square)을 나타내고 있다. 그림에서 알 수 있듯이 PMI는 PKI와 함께 사용됨으로 인해서 검증경로가 2차원으로 구성되어진다. SOA(Source Of Authority)는 PMI에서 직접적으로 신뢰되는 최상위 인증기관으로 PKI의 root CA와 같다. 실제 환경에서는 SOA와 사용자 사이에 하나 이상의 AA(Attribute Authority)가 존재할 수 있으며, 따라서 사용자의 속성 인증서의 유효성 검증을 위해서는 12개 이상의(공개키/속성) 인증서가 필요하게 된다.

Ⅲ. 속성 인증서 검증 서버

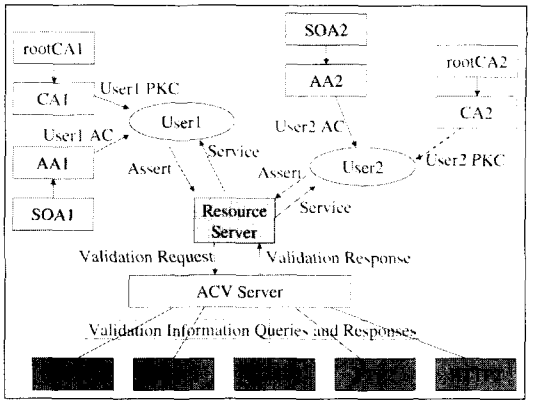
3.1 속성 인증서 검증 서버

이미 살펴보았듯이 속성 인증서의 유효성 검증은 복잡하고 많은 작업을 요구한다. 저장소에서 사용자의 속성 인증서 및 공개키 인증서를 찾아 와야 하며 가져온 인증서들을 사용하여 인증 경로를 구성할 수 있어야 한다. 또한 인증 경로상의 각 인증서들에 대해서 인증서 폐지 정보를 확인 해야한다. [그림 4]가 보여주고 있듯이 유효성을 확인하기 위한 이런 정보들은 LDAP, HTTP, FTP, OCSP 등의 다양한 프로토콜을 통해 제공되므로 서비스 제공자는 이런 모든 프로토콜을 사용할 수 있어야 한다.

이렇듯 자원을 가진 서버가 직접 속성 인증서의 유효성을 검증하는 것은 많은 오버헤드를 발생시키게 되며 전화기나 PDA등 제한된 연산 능력과 메모



〔그림 4〕 기존 PMI 인증서 검증 구조



〔그림 5〕 제안된 속성 인증서 검증 환경

리를 가진 환경에서는 사용이 역부족일 수도 있다. 이런 이유들로 인해서 자원 서버들의 인증서 경로 검증에 대한 오버헤드를 줄이고 PKI 및 PMI 전체 시스템에서 중복된 정보 전달을 줄이기 위한 방법이 요구되고 있다. PKI영역에서는 PKC 유효성 검증에 대한 부담을 줄여주기 위한 방법으로 CRL(인증서 폐지 목록) 대신 DeltaCRL 이나 OCSP 서비스를 이용하는 방법을 비롯해 최근에는 DPV(Delegated Path Validation) 및 DPD(Delegated Path Discovery)의 도입 등과 같은 연구 및 표준화 작업이 진행되고 있다. 그러나 이러한 노력은 이미 널리 보급된 PKI 환경에 중점을 두고 있으며 PMI 환경에 대한 고려는 부족한 현실이다. 본 논문에서 PKI에서의 검증 위임 및 검증 경로 구성 위임 등의 개념을 PMI로 확대 적용한 속성 인증서 검증 위임의 사용을 제안한다.

[그림 5]는 PKI및 PMI와 함께 ACV(Attribute Certificate Validation) 서버가 사용되는 환경을 나타내고 있다. 자원 서버는 사용자가 요청한 서비스의 제공만 고려할 뿐 사용자 속성 인증서의 유효성에 대해서는 ACV 서버에 검증을 일임한다. ACV 서버를 사용함으로써 자원 서버는 경로 검증과 같은 오버헤드를 줄일 수 있으며 실시간으로 인증서 유효성 여부를 확인할 수 있게된다. 또한 필요한 인증서 등을 얻기 위해 기존에는 저장소에 여러 번의 요청 및 수신을 필요로 했지만, ACV 서버가 존재할 경우 단 한번의 요청만으로 인증서 등의 필요한 정보를 획득할 수 있게된다.

3.2 속성 인증서 검증 위임 요구사항

속성 인증서 검증 위임에서는 속성 인증서뿐만 아니라 공개키 인증서도 처리할 수 있어야 한다. 따라서

PKI에서의 DPV, DPD 요구 사항과 함께 속성 인증서 처리와 관련된 요구사항을 모두 만족하여야 한다.

검증 정책은 인증서 유효성 검증에 적용될 규칙들의 집합을 말한다. 신뢰되는 CA 또는 AA 등의 인증기관들을 명시하거나 인증 경로 생성 및 경로 위임 처리에 대한 규칙, 인증서의 폐지 상태 정보 처리에 대한 규칙 등을 제공한다. 검증 서버는 검증 정책에 따라 하나 또는 그 이상의 속성 인증서 및 공개키 인증서에 대해 유효성 검사를 수행할 수 있어야한다.

특정 환경에서는 현재가 아닌 과거 혹은 미래의 시간에 대해 속성 인증서의 유효성을 검증할 필요가 있으며 이런 경우 검증 클라이언트는 현재 시간이 아닌 시간에 대해서도 인증서의 유효성 여부 판단을 요청할 수 있다. 검증 서버는 이러한 요청을 처리할 수 있도록 요청된 검증시간의 인증서 폐지 정보 등을 보존 또는 획득할 수 있어야 한다.

검증 서버는 요청된 인증서에 대한 폐지 상태를 확인 할 수 있도록 OCSP 응답, CRL, 그리고 Delta CRL 등의 다양한 폐지 정보를 조합하거나 또 다른 검증 서버의 검증 응답을 이용할 수 있어야 한다. 요청된 시간의 인증서에 대한 폐지 상태 정보를 얻을 수 없을 경우에는 검증 결과는 유효하지 않음을 명시해야 한다.

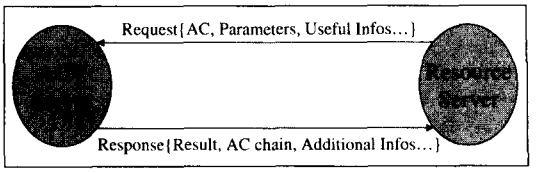
검증 클라이언트는 속성 인증서 또는 공개키 인증서를 요청 메시지에 직접 포함하거나 인증서에 대한 참조를 통해 검증을 요청할 수 있다. 참조에 의해 검증이 요청된 경우 검증 서버는 참조된 속성 인증서 및 공개키 인증서를 가져올 수 있어야 한다.

검증 서버가 사용되는 환경에서는 검증 클라이언트는 인증서의 유효성 여부를 전적으로 검증 서버의 응답에 의존하게 된다. 이런 환경에서 Replay 공격이나 서비스 거부(DoS또는 DDoS)공격 등을 받게 된다면 그 신뢰성이 크게 손상될 것이다. 따라서 검증서버는 다양한 보안 공격에 대응할 수 있도록 구현되어야 한다.

Ⅳ. 속성 인증서 검증 프로토콜

4.1 속성 인증서 검증 프로토콜(ACVP) 개요

속성 인증서 검증 프로토콜은 [그림 6]에서 보여주고 있듯이 속성 인증서 검증 서버와 검증 클라이언트인 자원 서버간의 주고받는 메시지에 대한 약속이다. 검증 프로토콜은 검증 요청, 검증 응답 메시지에 대한 포맷을 정의하여야 하며 검증 프로토콜로서 요구되는 조건들을 충족시킬 수 있어야 한다.



[그림 6] 검증 서버와 자원 서버의 검증 요청 및 응답

4.2 속성 인증서 검증 프로토콜 요구 사항

검증 요청 메시지의 가장 중요한 정보는 검증 대상이 되는 속성 인증서 또는 공개키 인증서이다. 따라서 검증 요청 메시지에서는 검증 대상 인증서를 직접 포함하거나 또는 명확한 참조를 제공할 수 있도록 지원해야 한다.

검증 클라이언트는 검증 과정에 도움이 될 수 있는 인증서나 폐지 정보 등을 검증 서버에게 제공할 수 있어야한다. 그러기 위해서 검증 요청 메시지는 요청 문법에 이러한 정보들을 담을 수 있도록 정의되어야 한다.

검증 프로토콜은 보안 공격에 대처할 수 있도록 필요한 메커니즘을 지원해야한다. 즉, 요청 메시지는 고유한 아이디를 가짐으로서 재 사용되는 것을 막아야 하며, 허용된 검증 클라이언트만이 검증 서비스를 요청할 수 있도록 인증 기능을 제공해야 한다.

검증 클라이언트의 검증 요청은 요청 사실 자체만으로도 중요한 정보의 노출을 야기할 수 있다. 이런 경우 검증 요청 및 검증 응답 메시지는 비밀성을 요구할 수 있으며 검증 프로토콜에서 이런 요구를 수용할 수 있도록 메시지 암호화 메커니즘을 제공해야 한다.

검증 서버 및 클라이언트는 검증 프로토콜 메시지를 신뢰하기 위해서 응답 메시지가 검증 서버 혹은 클라이언트가 작성한 원래의 것인지 확인이 필요할 수 있다. 따라서 검증 프로토콜에서는 메시지 무결성을 제공할 수 있도록 해쉬 및 서명 기능을 지원해야 한다.

검증 클라이언트가 검증 서버를 전적으로 신뢰하지 못하는 환경에서도 사용될 수 있도록 응답 메시지는 전체 속성 인증서 경로 및 각 속성 인증서 소유자의 공개키 인증서 경로를 제공할 수 있어야 한다.

4.3 검증 요청 메시지

검증 프로토콜은 IETF PKIX의 다른 프로토콜들과의 호환을 위해 ASN.1 문법으로 정의하였다. 검증요청은 요청 메시지인 RequestData 그리고 요청자의 속성 인증서, 메시지에 대한 암호화 또는 서명 정보들로 이루어져 있다. 검증 요청자는 필요에 따라 요청 메시지에 대한 비밀성 및 인증 기능을 제공할 수 있다.

```
ACVRequest ::= SEQUENCE {
    requestData         RequestData,
    requesterAC         [0] CertChoice OPTIONAL,
    encrypted           [1] EncryptionInfo OPTIONAL,
    signed              [2] SignInfo OPTIONAL }
RequestData ::= CHOICE {
    rawReq              [1] RequestInfo,
    encryptedReq        [2] OCTET STRING }
CertChoice ::= CHOICE {
    pkc         [1] Certificate,
    ac          [2] AttributeCertificate,
    pkcRef      [3] CertReference,
    acRef       [4] CertReference }
CertReference ::= SEQUENCE {
    certIssuer          GeneralNames,
    serial              INTEGER }
```

RequestData는 RequestInfo의 비밀성 제공 여부에 따라 rwqReq 혹은 encryptedReq 둘 중의 하나를 선택해서 사용하게 된다.

```
EncryptInfo ::= SEQUENCE {
    usedCert            CertChoice,
    keyEncryption       AlgorithmIdentifier,
    encrytAlgor         AlgorithmIdentifier,
    encryptKey          OCTET STRING    }
SignInfo ::= SEQUENCE {
    singerPKC           CertChoice,
    signAlgor           AlgorithmIdentifier,
    signValue           BIT STRING      }
```

EncryptInfo는 요청 메시지의 암호화에 사용된 알고리즘과 암호화 키에 대한 정보를 포함한다. keyEncryption은 암호화에 사용된 세션 키를 상대방의 공개키로 암호화할 때 사용한 알고리즘을 명시한다. encryptKey는 수신자의 공개키에 의해 세션 키(암호화 키)가 암호화된 값이다.

SignInfo는 요청 메시지에 대한 서명에 사용된 공개키 인증서, 해쉬 및 서명 알고리즘, 그리고 서명 값을 전달한다.

RequestInfo는 검증에 사용될 정책, 검증 응답으로 받고자 하는 정보, 그리고 개별적인 인증서들에 들에 대한 요청 정보들로 이루어져 있다. 검증 정책은 정책 식별자, 신뢰하는 AA, 신뢰하는 CA, AC 경로 길이 제한, 그리고 PKC 경로 길이 제한 등의 정보로 이루어져 있다. ResponseType은 검증 응답 메시지의 암호화 및 서명을 요청하기 위한 정보이다. 암호화 및 서명을 요구하지 않는 경우 responseType 필드는 생략되어야 한다.

```
RequestInfo ::= SEQUENCE {
    valPolicy           ValidationPolicy,
    responseType        [0] ResponseType OPTIONAL,
    valRequests         SEQUENCE OF ValRequest }
ValidatoinPolicy ::= SEQUENCE {
    policyID            OBJECT IDENTIFIER,
    trustedAA           CertChoice, -- SOA
    trustedCA           CertChoice, -- rootCA
    acPathLen           INTEGER, -- AC path length
    pkcPathLen          INTERGER -- PKC path length }
ResponseType ::= ENUMERATED {
    encrypt     (0), -- Encrypt Response Data
    sign        (1), -- Sign Response Data
    both        (2)  -- Sign and Encrypt Response }
ValRequest ::= SEQUENCE {
    reqCert             CertChoice,
    validationTime      GeneralizedTime,
    reqID               INTEGER,
    returnInfo          ReturnInfo,
    usefulCerts         [0] Certificates OPTIONAL,
    revokedCerts        [1] Certificates OPTIONAL }
Certificates ::= SEQUENCE OF CertChoice
ReturnInfo ::= ENUMERATED {
    resultOnly  (0), -- Validation Rusult Only
    pkc         (1), -- requested PKC
    ac          (2), -- validated AC
    pkcPath     (3), -- PKC path
    acPath(4), -- AC path
    ACSquare    (5)  -- Full AC validation Square }
```

ValRequest는 검증 요청의 기본 단위로서 검증할 인증서와 필요한 정보들로 구성되어 있다. 검증 대상 인증서는 속성 인증서 또는 공개키 인증서가 될 수 있으며 인증서를 직접 전달하거나 참조에 의해 전달하게 된다. 검증 시간은 인증서 검증에 적용될 시간을 나타낸다. reqID는 replay 공격을 막고 각각의 검증 요청을 고유하게 식별하기 위해 사용된다. returnInfo는 검증 결과로서 요청자가 얻고자 하는 정보를 나타내며, 검증 대상 인증서, PKC 또는 AC 경로 등을 요청할 수 있다. returnInfo의 값이 ACSquare인 경우에는 응답 메시지는 속성 인증서 경로와 각 개체들

의 공개키 인증서 경로를 모두 포함해야 하며 이 정보는 [그림 3]과 같은 속성 인증서 검증 사각형(AC Validation Square)을 형성할 수 있어야 한다. usefulCerts와 revokedCerts는 검증 과정에서 사용될 수 있는 인증서나 폐지 정보 등 유용한 정보를 전달하기 위해 사용된다.

4.4 검증 응답 메시지

검증 응답 메시지는 responseData와 검증 서버의 속성 인증서, 그리고 암호화 및 서명과 관련된 정보 등을 담고 있다.

```
ACVResponse ::= SEQUENCE {
   responseData      ResponseData,
   responderAC       [0] CertChoice OPTIONAL,
   encrypted         [1] EncryptionInfo OPTIONAL,
   signed            [2] SignInfo OPTIONAL }
ResponseData ::= CHOICE {
   rawResponse       [1] ResponseInfo,
   encryptedRes      [2] OCTET STRING }
```

ResponseData는 ResponseInfo가 암호화되었는지 여부에 따라 rawReasponse 와 encryptedRes 중의 하나의 값을 사용하여야 한다. ReasponseInfo는 응답 메시지의 핵심 내용을 담고 있으며 암호화 및 서명의 대상이 된다.

```
ResponseInfo ::= SEQUENCE {
   valPolicy         ValidationPolicy,
   requestStatus     RequestStatus,
   valReponses       SEQUENCE OF ValResponse }
```

ResponseInfo는 검증에 사용한 검증정책, 검증 요청 메시지의 처리결과, 그리고 개별적인 검증 요청에 대한 응답들로 구성되어 있다. 요청 메시지 처리 결과 값은 succeed이면 모든 요청 메시지가 정상적으로 처리되었음을 나타낸다. partialError는 일부 요청 메시지가 처리되지 않았음을 나타내며, notSupportedValPolicy는 요청자가 지정한 검증 정책이 지원되지 않는 정책임을 명시한다. notSupportedAlgor는 요청자가 사용한 암호 알고리즘 또는 서명 알고리즘이 지원되지 않는 경우를 나타낸다. requesterNotAllowed는 검증 요청자가 요청 권한을 가지고 있지 않는 경우를 나타낸다. 마지막으로 serverBusy는 검증 서버가 일시적인 과부하로 인해서 요청 메시지를 처리지 못한 경우를 나타낸다.

```
RequestStatus ::= ENUMERATED {
   succeed                (0), -- All processed
   partialError           (1), -- Some Errors
   notSupportedValPolicy  (2), -- Bad Policy
   notSupportedAlgo       (3), -- Bad Algorithm
   requestorNotAllowed    (4), -- Bad Requester
   serverBusy             (5), -- ACV Server busy }
```

ValResponse는 개별적인 검증 요청에 대한 결과를 제공하는데 사용된다. 검증을 요청한 인증서, 요청 식별자, 요청메시지에 대한 해쉬 값, 검증 결과, 그리고 요청자가 요구한 선택적인 검증 정보들을 제공한다. 검증 요청에 대한 해쉬 값은 검증 서버가 검증 요청을 정상적으로 처리했는지 여부를 확인하기 위해서 사용된다. 인증서에 대한 검증 결과는 유효, 유효하지 않음, 결정할 수 없음의 세 가지 값 중의 하나를 가진다.

```
ValResponse ::= SEQUENCE {
   valCert           CertChoice,
   reqID             INTEGER,
   reqHash           RequestHash,
   valResult         ValidationResult,
   requestedInfo     [0] RequestedInfo OPTIONAL }
RequestHash ::= SEQUENCE {
   hashAlgo          AlgorithmIdentifier,
   hashValue         OCTET STRING }
ValidationResult ::= ENUMERATED {
   valid     (0), -- Queried Cert is Valid
   invalid   (1), -- Queried Cert is not Valid
   unknown   (2)  -- Cannot Decide }
RequestedInfo ::= SEQUENCE OF CertAndCRI
CertAndCRI ::= SEQUENCE {
   cert      CertChoice,
   cri       CertificateRevocationInfo }
```

검증 클라이언트가 요청한 검증 정보들은 인증서와 그 인증서에 대한 폐지 정보를 기본 단위로 하는 CertAndCRI를 통해 전달된다. 요청자가 하나 이상의 인증서를 요구했을 경우에는 필요한 개수 만큼의 CertAndCRI를 포함하여야 한다. CertAndCRI의 cert필드는 인증서를 참조대신 직접 포함하고 있어야 한다. cri는 인증서의 폐지 정보를 제공해줄 수 있도록 CRL, deltaCRL, 그리고 OCSPResponse 등의 정보를 담을 수 있다.

```
CertificateRevocationInfo ::= CHOICE {
   crl               [1] CRL,
   deltaCRL          [2] dCRL,
   ocspResponse      [3] OCSPResponse }
```

V. 효율성 평가

5.1 검증 클라이언트

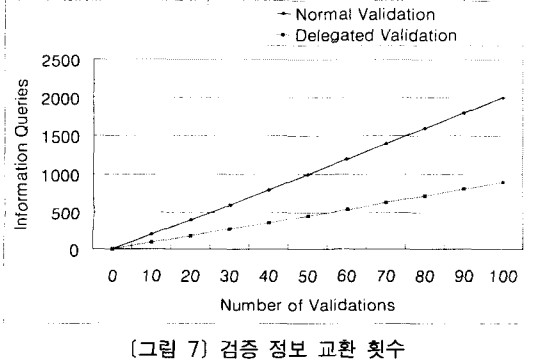
검증 위임을 사용하는 환경에서 검증 클라이언트는 사용자 인증서를 검증하기 위해 단 한번의 요청만을 필요로 한다. 검증 위임을 사용하지 않는 환경에서 필요한 정보 요청의 횟수는 인증서의 개수와 인증서 폐지 정보 개수의 합으로 나타낼 수 있다. [표 1]은 인증서의 경로 길이에 따른 필요한 정보 개수이다.

〔표 1〕 인증 경로 길이에 따른 필요한 정보 수

	PKC path = 2	3	4
AC path = 2	9	13	17
AC path = 3	14	20	26
AC path = 4	19	27	35
필요한 인증서 개수 = ACpath × (AC+PKCpath) 폐지정보 개수 = SOA와 RootCA를 제외한 인증서 수 (ACpath: AC 경로 길이, PKCpath: PKC 경로 길이)			

5.2 PKI 및 PMI 전체 시스템

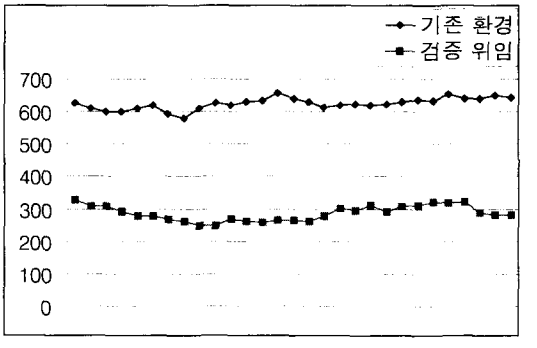
속성 인증서 및 공개키 인증서의 검증 경로는 최소 길이가 2이지만 일반적인 환경에서는 길이가 3이상이다. [그림 7]에서는 인증 경로가 3인 경우를 기준으로 전체 시스템의 정보 교환 횟수를 비교하였다. 일반 환경에서는 매 검증 시마다 인증서 검증 경로를 구성해야 하고 각 인증서의 폐지 정보를 확인해야 한다. 그러나 검증 위임이 사용되는 환경에서는 검증에 사용된 속성 인증서, 공개키 인증서, CRL, 그리고 delta CRL 등을 내부에 저장해 둠으로써 외부 저장소와의 중복되는 정보 교환을 줄일 수 있다. 따라서 ACV 서버는 OCSP등의 저장이 불가능한 인증서 폐지 정보만을 추가로 가져오면 되므로 전체 시스템의 정보 교환 횟수는 현저히 줄어들 수 있다.



〔그림 7〕 검증 정보 교환 횟수

5.3 PMI 사용자

검증 위임을 사용하게 되면 속성 인증서의 유효성을 결정하는데 걸리는 시간도 크게 줄어들 수 있다. 이러한 시간 단축은 PMI 사용자의 서비스 대기 시간에 직접적인 영향을 미치게된다. [그림 8]은 속성 인증서와 공개키 인증서의 검증 경로 길이가 3인 경우, 검증 위임 사용 여부에 따른 시뮬레이션 결과이다. 네트워크 상황에 따라서 약간의 차이가 있기는 하지만 검증 위임을 사용했을 때, 서비스를 받기까지의 대기 시간이 현저히 줄어드는 것을 볼 수 있다.



〔그림 8〕 검증 위임 사용과 서비스 대기 시간(단위:ms)

5.4 검증 프로토콜

공개키 인증서에 대한 검증 프로토콜은 현재 IETF Internet-Draft인 SCVP와 CVP 두 가지가 있다. [표 2]는 이들 프로토콜의 특징과 제안된 ACVP(ACV Protocol)에 대한 비교이다.

〔표 2〕 검증 프로토콜 비교

Protocol	PKC 검증	AC 검증	인증	비밀성	Transport
SCVP	○	×	○	○	S/MIME
CVP	○	×	○	×	Any
ACVP	○	○	○	○	Any

VI. 결 론

속성 인증서 검증 서버의 사용은 복잡하고 많은 처리 과정을 요구하는 인증서 유효성 확인 작업을 간소화 시켜줄 수 있다. 일반 사용자에게 서비스를

제공하는 자원 서버는 자신의 서비스에만 집중할 수 있으며 힘든 검증 작업은 전문화된 검증 서버에 일임함으로서 검증 클라이언트의 효율성을 높일 수 있다. PKI 및 PMI 시스템에서는 검증 위임을 사용함으로서 전체적인 정보 교환의 횟수가 현저히 줄어들게 되며, 따라서 시스템 전체의 성능 향상을 가져올 수 있다. 이런 효율성 증가는 최종적으로 사용자에게 전달되어 사용자의 서비스 대기 시간을 크게 줄여 줄 수 있다.

속성 인증서 검증 위임을 사용하기 위해 정의한 검증 프로토콜은 필요에 따라 비밀성 및 인증 서비스를 제공하며 전송 프로토콜에 종속되지 않도록 설계되어 어떤 환경에서도 사용될 수 있다. 또한 속성 인증서 및 공개키 인증서 모두를 지원함으로써 통합된 인증서 검증 기능을 제공할 수 있다.

이제 한창 관심이 커져가고 있는 PMI가 더욱더 빠르고 폭넓게 보급되기 위해서는 PMI 전체의 효율성을 좌우하는 인증서 유효성 검증의 부담을 줄이는 것이 필수 불가결하다.

본 논문에서 제시한 검증 위임과 검증 위임을 지원하기 위해 정의한 검증 프로토콜은 PMI의 효율을 개선할 수 있는 하나의 대안이 될 수 있을 것으로 기대한다.

참 고 문 헌

[1] ITU-T Recommendation X.509, *"Public-Key And Attribute Certificate Frameworks"*, ISO/IEC 9594-8, May 2001.

[2] R. Housley, W. Polk, W. Ford, D. Solo, *"Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List Profile"*, IETF PKIX RFC 3280, April 2002.

[3] S. Farrell, R. Housley, *"An Internet Attribute Certificate Profile for Authorization"*, IETF PKIX RFC 3281, April 2002.

[4] Denis Pinkas, Russ Housley, *"Delegated Path Validation and Delegated Path Discovery Protocol Requirements"*, IETF PKIX Internet-Draft, May 2002.

[5] Denis Pinkas, *"Certificate Validation Protocol"*, IETF PKIX Internet-Draft, June 2002.

[6] A. Malpani, R. Housley, T. Freeman, *"Simple Certificate Validation Protocol"*, IETF PKIX Internet-Draft, June 2002.

[7] M. Myers, X. Liu, J. Schaad, J. Weinstein, *"Certificate Management Messages over CMS"*, IETF PKIX RFC 2797, April 2002.

[8] P. Yee, *"Attribute Certificate Management Messages over CMS"*, IETF PKIX Internet-Draft, March 2002.

[9] D. W. Chadwick, S. Legg, *"Internet X.509 Public Key Infrastructure LDAP Schema and Syntaxes for PKIs"*, IETF PKIX Internet-Draft, 26 June 2002.

[10] D. W. Chadwick, S. Legg, *"Internet X.509 Public Key Infrastructure LDAP Schema and Syntaxes for PMIs"*, IETF PKIX Internet-Draft, June 2002.

[11] A. Arsenault, S. Turner, *"Internet X.509 Public Key Infrastructure: Roadmap"*, IETF PKIX Internet-Draft, July 2002.

[12] M. Myers, R. Ankney, A. Malpani, S. Galperin, C. Adams, *"X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP"*, IETF PKIX Internet-Draft, Feb 2002.

[13] C. Adams, S. Farrell, *"Internet X.509 Public Key Infrastructure Certificate Management Protocols"*, IETF PKIX Internet-Draft, December 2001.

[14] D. W. Chadwick, *"Internet X.509 Public Key Infrastructure Operational Protocols - LDAPv3"*, IETF PKIX Internet-Draft, January 2002.

[15] S. Farrell, *"TLS extensions for Attribute Certificate based authorization"*, IETF PKIX Internet-Draft, August 1998.

[16] Peter Gutmann, *"Internet X.509 Public Key Infrastructure Operational Protocols: Certificate Store Access via HTTP"*, IETF PKIX Internet-Draft, July 2002.

[17] C. Francis, D. Pinkas, *"Attribute Certificate Policies Extension"*, IETF PKIX Internet-Draft, May 2002.

[18] D. W. Chadwick, *"An X.509 Role-based Privilege Management Infrastructure"*, Business Briefing: Global Infosecurity, 2002.

[19] 강명희, *"PMI: Privilege Management Infrastructure 개요"*, 퓨처시스템 Technical Report: FS-TR02-01, June 2002.

[20] R. Housley, *"Cryptographic Message Syntax(CMS)"*, IETF PKIX RFC 3369, August 2002.

〈著者紹介〉

이 승 훈 (Seung-Hoon Lee) 정회원
2001년 2월 : 연세대학교 컴퓨터과학과 졸업
2003년 2월 : 연세대학교 컴퓨터산업시스템공학과 석사
2003년 1월~현재 : LG전자 연구원
<관심분야> 정보보호, 무선 이동 통신

송 주 석 (Joo-Seok Song)
1976년 2월 : 서울대학교 전기공학 졸업
1979년 2월 : 한국과학원 전기·전자 석사
1979년 3월~1982년 2월 한국전자통신연구소 연구원
1988년 2월 : Univ. of California at Berkeley 전산과학 박사
1989년 3월~현재 : 연세대학교 교수
<관심분야> 정보통신, 정보보호