이학박사 학위논문

# 새로운 공개키 프레임워크와 PAN을 위한 응용


2007년 8월

부 경 대 학 교 대 학 원

전 자 계 산 학 과

장 화 식

이학박사 학위논문

# 새로운 공개키 프레임워크와 PAN을 위한 응용

지도교수 이 경 현

이 논문을 이학박사 학위논문으로 제출함

2007년 8월

부 경 대 학 교 대 학 원

전 자 계 산 학 과

장 화 식

# 장화식의 이학박사 학위논문을 인준함

2007년 8월

주 심 공학박사 김 창 수 ㊞

위 원 공학박사 이 훈 재 ㊞

위 원 이학박사 신 상 욱 ㊞

위 원 이학박사 신 원 ㊞

위 원 이학박사 이 경 현 ㊞

# < 차 례 >

## < 표 차 례 >

## < 그림차례 >

# A New Public Key Framework and its Applications to PAN

Jang, Hwa Sik

Department of Computer Science, Graduate School,
PuKyong National University

## Abstract

Without doubt, the promise of Public Key Infrastructure (PKI) technology has attracted a significant amount of attention to support secure and authenticated services in the heterogeneous networks. The IETF PKIX Working Group is developing the Internet standards to support an X.509-based PKI, which provides a framework on services related to issuing public key certificates and distributing revocation information. The lack of a mechanism that provides efficient and timely distribution of certification revocation information is a main issue for implementing more efficient PKI environment. The existing certificate revocation schemes place a considerable processing, communication, and storage overheads on certificate authority(CA) as well as the relying parties.

In this thesis, the main target of our research is to provide a new public key framework which reduces the overheads of computation for digital signature generation/verification and communication for verifying the validity of X.509 certificate. Specially, we focus on developing new public key frameworks in Internet and Personal Area Network (PAN) environment.

For Internet environment, we review J. Zhou et al's public key framework from the view point of the actual deployment, and propose a new public key framework by changing security parameters into more suitable ones to enhance the actuality and reduce the overheads of computation and

communication. Moreover, we analyze the security of our new public key framework from the vulnerability window point of view.

A PAN is the interconnection of fixed, portable, or moving components within a range of an individual operating space, typically within a range of 10 meters. In PAN the communication between components should be secure and authenticated since private information and personal data will be transmitted over radio links. Secure and authenticated communication can be achieved by means of proper security protocols and appropriate security associations among PAN components. For the sake of supporting key management in a PAN, a personal CA in the personal PKI concept is responsible for generating public key certificates for all mobile devices within the PAN. The personal CA is used by an ordinary user at home or small office deployment distinguished from large scale or global CA functions. Although the personal PKI concept seems to be properly applied to PAN environment, the adaptation of PKI concept to PAN is not suitable due to the limited resource of the mobile devices.

For PAN environment, we propose a new public key framework that reduces computational overheads for generating and verifying signatures on mobile devices. Especially, we focus on eliminating the traditional public key operations on mobile devices by means of one-time signature scheme, and differentiating it from previously proposed server-assisted signatures relied on assistances of a signature server. As a result, the proposed protocol gets rid of inherent drawbacks of server-assisted signatures such as problematic disputes, and high computational and storage requirements on the server side. Moreover, our framework provides simplified procedure for certificate status management based on hash chain to alleviate communication and computational costs for checking certificate status information.

# 1. 서론

네트워크 접속 기술의 발전과 확산으로 다양한 형태의 통신을 이용한 전자적 서비스, 거래 및 결재를 수행하는 이용자의 수가 급속도로 증가하고 있으며, 각종 전자적 서비스들의 다양성과 활용 범위가 넓어지고 있다. 네트워크 접속을 통한 전자적 서비스를 안전하게 수행하기 위하여, 공개키 기반구조(Public Key Infrastructure, PKI)는 전자서명을 통한 인증, 부인 방지 서비스의 제공과 함께 대칭키와 비대칭키의 결합을 통한 기밀성 보장 및 키 관리 서비스의 사용을 위한 기반구조로서 현재 가장 널리 사용되고 있으며, 향후에 새로이 개발되는 네트워크 접속 기술의 보안 기반구조 설계를 위한 기준을 제시하고 있다.

공개키 기반구조에서 통신하는 각 개체(장치 또는 사용자)들은 인증서를 통하여 상대방의 공개키를 신뢰하는 방법으로 획득함으로써, 상호간의 신분확인을 위한 암호학적 절차를 수행 가능하도록 한다. X.509 인증서는 IETF PKIX 워킹그룹에 의하여 재정된 인증서 표준 형식으로, 공개키와 그 공개키 소유자의 신원정보를 암호학적 수단을 통하여 결합한 것이며 인증기관(Certificate Authority, CA)이라 불리는 신뢰된 제3자에 의하여 발행된다[14]. 인증기관에 의해 발급된 인증서는 유효기간이 종료하는 시점(즉, 만기일) 이전에 인증서 소유자의 의도 또는 악의적인 공격으로 인하여 취소될 수 있으므로, 유효기간이 종료되기 이전에 취소되어진 인증서에 대한 정보를 적시적으로 분배하기 위한 인증서 상태 검증 기술은 PKI에서 인증서의 오용을 방지하기 위한 매우 중요한 연구 과제라 할 수 있다.

현재 연구되어진 Certificate Revocation List(CRL), Online Certificate Status Protocol(OCSP), Delta CRL, Indirect CRL, Certificate Revocation Tree(CRT) 및 Certificate Revocation System(CRS) 등과 같은 다양한 형태의 인증서 상태 검증 기술들은 사용자뿐만 아니라 인증기관에게 높은 통신 및 계산 비용을 요구하기 때문에, 가까운 미래에 개발될 다양한 형태의 네트워크 접속 기술에서 좀 더 안전한 전자적 거래를 수행하기 위하여 효율적이고 적시적인 인증서 상태 검증 기술의 개발은 반드시 선행되어져야 한다[1][4][19][20].

Zhou는 인증서의 유효 기간을 짧은 기간들로 나누고, 인증서가 인증서 소유자의 통제아래 각 기간의 끝 지점에서 취소될 수 있는 새로운 공개키 프레임워크[31]를 제안하였으나, 인증기관이 인증서의 취소를 직접적으로 제어할 수 없기 때문에 악의적인 사용자는 인증서가 취소되었을지라도 다른 사용자와 인증을 위한 절차를 성공적으로 수행할 수 있다는 취약점을 가지고 있다. 비록 Zhou는 보안 서버라는 새로운 신뢰기관을 통한 대안을 제시하였지만, 보안 서버는 위의 취약점을 해결하기 위한 불필요한 신뢰 기관일 뿐이며, 인증서의 유효 기간을 짧은 기간들로 나누는 것은 인터넷 환경에서 타임 동기화 문제를 발생시킬 수 있는 단점을 가지고 있다. 따라서, 본 논문에서는 Zhou의 공개키 프레임워크내의 보안 파라메타들을 재조명한 후, 실제 구현 환경에 적합하도록 개선하고, Zhou의 공개키 프레임워크에서 불필요한 신뢰기관을 제거함으로써 인터넷 환경에 적용 가능한 새로운 형태의 공개키 프레임워크를 제안하고자 한다.

최근 PDA 및 랩탑과 같은 이동 디바이스들의 기능과 성능이 나날이 발전하고 다양한 형태의 네트워킹 능력을 소유한 이동 디바이스들이 널리 사용되어지고 있어, 이에 따라 다양한 네트워크 접속 기술들이 새로이 개

발될 것이다. 따라서, 가까운 미래의 모바일 통신에서는 네트워크 접속에 사용되는 디바이스들이 물리적으로 사용자에 근접한 범위 내에서 위치하여 Personal Area Network(PAN)을 구성 가능할 것이며, 이러한 PAN은 개인의 작업 공간인 10 미터 이내에 고정, 휴대 또는 이동 디바이스들로 구성될 것이다. PAN은 개인이 구성하는 네트워크로써 PAN내의 통신은 기밀정보와 개인 데이터가 주를 이루기 때문에, 신뢰성 있고 인증된 통신 서비스가 필수적이며, 이와 같은 보안 서비스를 위한 기반구조로써 Personal PKI(Personal Public Key Infrastructure)가 제안되었다[18]. 그러나, PAN을 구성하는 대다수의 모바일 디바이스들은 일반적으로 제한된 컴퓨팅 능력을 가지고 있으므로, PKI에서 사용되는 전자서명과 인증서 상태 검증 기술과 같이 높은 통신 및 계산 비용을 요구하는 기법들은 적합하지 못하다. 따라서, 본 논문에서는 PAN을 구성하는 디바이스들을 위한 효율적인 전자서명 및 인증서 상태 검증 기술을 지원하기 위한 새로운 공개키 프레임워크를 제안하고자 한다. 제안된 PAN을 위한 새로운 공개키 프레임워크는 일회용 전자서명 기법(One-Time Signature)[25]을 통하여 이동 장치들이 전통적인 공개키 연산을 수행할 필요가 없으며, 서명 서버의 도움에 의존하는 서버 지원된 전자서명(Server-Assisted Signature)[7]에서 야기되는 분쟁 및 서버에서 요구되는 높은 계산 비용과 많은 저장 공간의 요구들이 불필요하다. 더욱이, PAN을 위한 새로운 공개키 프레임워크는 인증서 상태 검증을 위하여 해쉬체인을 적용함으로써, 디바이스의 인증서 상태 검증을 위한 통신 및 계산 비용을 경감하는 단순화된 절차를 제공한다.

공개키 기반구조의 인증서는 인증 및 신뢰성 있는 통신과 같은 보안 서비스를 제공하기 위하여 다양한 형태의 네트워크 접속환경에서 적용되고

있으며, 기 제안된 인증서 상태 검증 기술은 보안 서비스를 수행할 시점에서 인증서의 취소 유무를 검증하기 위해서 높은 통신 및 계산 비용을 요구하고 있다. 본 논문에서는 기존의 인터넷 환경뿐만 아니라 가까운 미래에 실현될 PAN과 같은 새로운 네트워크 접속 환경에서의 보안 서비스를 위한 새로운 공개키 프레임워크를 제안함으로써 요구되는 다양한 형태의 보안 서비스를 효율적으로 제공하기 위한 기틀을 제공하며, 제안된 새로운 인증서 상태 검증 기술을 통하여 효율적으로 인증 가능한 네트워크 접속 방식을 제공하는데 그 목적이 있다.

2장에서는 관련 연구로 인터넷 환경에서의 공개키 기반구조, 인증서의 형식, 그리고 인증서 상태 검증 기술에 대해 소개하며, 3장에서는 PAN 및 Personal PKI의 개요와 구성 요소 및 보안 기술들을 소개한다. 4장에서는 인터넷 환경을 위한 새로운 공개키 프레임워크를 제안하고, 성능 및 보안성을 평가한다. 5장에서는 PAN을 위한 새로운 공개키 프레임워크를 제안하고, 성능 및 보안성을 평가한다. 최종적으로, 6장에서 결론을 맺는다.

# 2. 공개키 기반구조 및 인증서 상태 검증 기술

인터넷의 발달로 인하여 전자상거래가 활발하게 사용되는 실생활 환경에서 더욱 신뢰할 수 있는 네트워크 환경을 제공하기 위해 공개키 기반구조의 사용이 널리 확산되고 있다. 특히 거래 당사자 간의 신뢰가 더욱 요구되는 인터넷 상의 전자상거래와 금융서비스, 증권거래 등에서는 공개키 기반구조가 필수요소로 인식하고 있다. 이러한 공개키 기반구조의 기술은 ITU-T의 X.509 인증서 표준을 근간으로 하며 IETF에서는 이를 바탕으로 인터넷에 적합한 인증서 표준을 제정하고 있다[14]. 공개키 기반구조에서 상대방의 공개키를 사용하기 위해 인증서를 사용하고자 하는 사용자는 인증서를 사용하기 전에 반드시 인증서의 유효성을 검증하는 인증서 검증 과정을 수행해야 한다[22][27]. 인증서의 검증 과정이란 인증서와 인증 경로상의 인증서들이 사용하고자 하는 시점에서 그 효력이 취소되었거나 정지되었는지를 검증하는 것을 말하며, 이러한 인증서 검증 과정에서 사용하고자 하는 인증서의 취소 상태를 판단하는 것은 매우 중요하다[14].

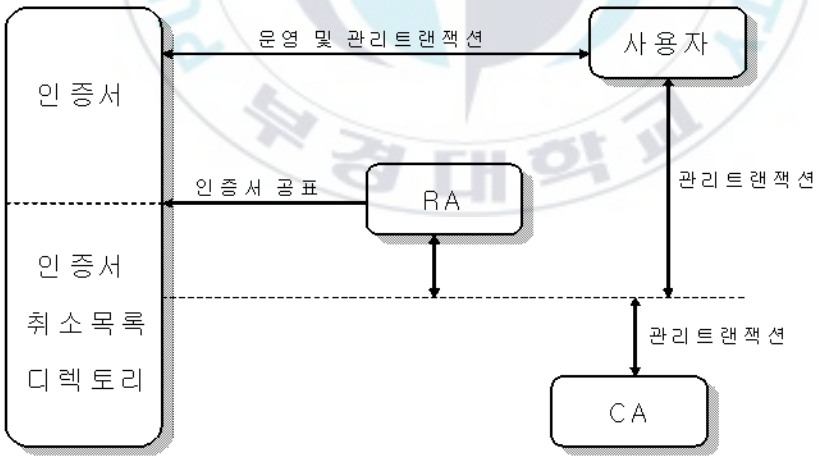
인증서 상태를 검증하기 위한 방법으로는 현재 가장 널리 사용되고 있는 인증서 취소 목록(Certificate Revocation List, CRL) 방식이 있다. 사용자가 인증서 취소를 많이 하였을 경우, CA에게 사용자는 인증서 취소를 요청하고, 이를 접수한 CA는 인증서 취소 목록에 이 정보를 추가하는데, 이로 인하여 인증서 취소 목록이 커져서 다루기 힘들어질 경우, 부분적인 인증서 취소 목록을 나타내기 위한 방법이 요구된다. CRL 분배점(CRL distribution point)과 Delta CRL은 각각 이러한 문제점을 해결하기

위한 방안으로 제시되었다. 인증서 취소 목록이 클 경우 이 목록을 여러 개로 나누어 전송 받을 수 있다면, 인증서 사용자는 제한된 시스템 자원 내에서 효율적으로 인증서를 검증할 수 있다. 이때 인증서 취소 목록을 여러 개의 부분으로 분리하여 사용하는 방법이 CRL 분배점이다. 또한 특정 시점의 기본 CRL을 마지막으로 발급한 이후에 새로 변경되거나 추가된 목록만을 사용자에게 전달하는 방법이 Delta CRL이다. CA는 모든 내용을 포함한 기본 CRL 생성과 동시에 Delta CRL을 생성한다[6][15][32].

CRL 분배점과 Delta CRL 방법에도 불구하고 CRL을 이용한 방법은 실시간으로 인증서의 상태 검증을 제공하지 못한다는 한계에 부딪친다. 이 문제점을 보완하기 위해서 인증서 검증만을 수행하는 독립적인 서버에 관한 연구가 진행되었다. 온라인 상에서 실시간으로 인증서의 취소 여부를 확인해 주는 Online Certificate Status Protocol(OCSP)이 제안되었다. 또한 취소 여부만이 아닌 전체 인증서 검증을 대행해 주는 Simple Certificate Validation Protocol(SCVP)이 제안되었다. 공개키 인증서의 검증과 데이터의 소유 증명 등을 서비스하여 부인방지 서비스를 제공하는 Data Validation and Certification Server(DVCS)가 제안되고 있다. 다음은 공개키 기반구조와 인증서 상태 검증을 위한 메커니즘들에 대하여 알아본다[1][4][19][20].

## 2.1 공개키 기반구조

오늘날 네트워크의 발전으로 실생활에서 인터넷을 사용하여 모든 정보를 교환하고 있다. 인터넷을 통하여 정보 교환을 할 때 자칫 중요한 정보를 노출시키거나 정보의 위조 및 변조 등과 같은 위험성을 내포하고 있다. 따라서 위와 같이 중요 정보에 대한 정보보호의 필요성이 강조되기 시작하였다. 인터넷에서 정보보호를 위해서는 모든 정보 교환에 대하여 정보의 노출을 방지하기 위한 기밀성(Confidentiality), 정보의 위조 및 변조 여부를 판단하는 무결성(Integrity), 정보의 송신자와 수신자 사이에 송수신 사실을 부인하지 못하도록 하는 부인방지(Non-Repudiation), 전송된 정보의 송신자와 수신자를 확실하게 증명해 주는 인증(Authentication) 등의 기능들이 제공되어야 한다. 공개키 기반구조는 공개키의 인증 문제를 해결하여 이와 같은 기능을 제공하는 정보보호 기반구조이다[27][35]. <그림 1>은 공개키 기반구조의 구성요소를 보여주고 있다.




<그림 1> PKI 구성요소

### 2.1.1 인증기관(Certificate Authority, CA)

최종개체 즉 사용자에게 공개키 인증서를 발급하며 다른 인증기관에 대한 상호인증서를 발급하는 신뢰된 개체이다. CA의 개인키를 사용하여 전자서명을 생성하며 이 서명을 사용자의 인증서에 첨부하여 사용자 인증서를 신뢰할 수 있도록 보장을 해준다. 또한 CA의 공개키를 사용하여 사용자 인증서를 확인하도록 한다. CA는 인증서뿐만 아니라 CRL을 생성하며 그 인증서 취소 목록을 저장소(Repository)에 전달하여 여러 실체들이 인증서 취소 상태를 검증하도록 CRL을 공포한다.

### 2.1.2 등록기관(Registration Authority, RA)

CA를 대신하여 사용자의 신분을 확인하거나 개인사용자 확인, 토큰 분배, 취소 보고, 이름 할당, 키 생성, 키 쌍 등록 등의 기능을 수행하는 인증기관과 인증서 주체가 될 실체 사이의 중간 매개체 역할을 수행하는 실체이다. 등록기관은 사용자의 신분을 확인하는 것을 인증기관으로부터 위임받아 수행한다.

### 2.1.3 디렉토리(Directory)

인증서와 사용자 관련 정보, 상호 인증서 쌍 및 인증서 취소 목록 등을 저장하는 곳으로 사용자가 정보를 검색하고자 할 때 이 장소에 접속하여 원하는 정보를 얻도록 하는 장소이다. 디렉토리를 관리하는 서버(인증기

관)는 DAP(Directory Access Protocol)나 LDAP(Lightweight DAP)를 이용하여 X.500 디렉토리 서비스를 제공한다. 인증서와 상호 인증서 쌍은 유효기간이 경과된 후에도 서명 검증을 위해 일정기간동안 디렉토리에 저장된다.

### 2.1.4 인증서 취소 목록(Certificate Revocation List, CRL)

공개키 기반구조에서 사용자 혹은 인증기관의 인증서가 만료되지 않은 시점에서 주체의 소속 변경 혹은 개인키의 노출 등의 이유로 무효화된 인증서에 대한 목록을 알리기 위하여 이들 정보를 인증기관의 개인키로 서명한 데이터 구조이다.

### 2.1.5 사용자(User)

인증서를 사용하는 실체로서 인증서를 원할 때 CA나 RA에게 인증서 발급 요청을 한다. 인증서를 발급받은 사용자는 이 인증서를 이용하여 사용자가 원하는 서비스를 제공받는다. 사용자는 기본적으로 다음의 기능을 수행할 수 있어야 한다.

- 자신의 비밀키/공개키 쌍을 생성할 수 있다.
- 공개키 인증서를 요청하고 획득할 수 있다.
- 전자서명을 생성 및 검증할 수 있다.
- 특정 사용자에 대한 인증서를 획득하고 그 상태를 결정할 수 있다.

- 인증 경로를 해석할 수 있다.
- 디렉토리를 이용하여 자신의 인증서를 다른 사용자에게 제공할 수 있다.
- 인증서 취소목록을 해석할 수 있다.
- 비밀키가 분실 또는 손상되거나 자신의 정보가 변했을 때(예: 조직의 탈퇴) 인증서 취소를 요청할 수 있다.

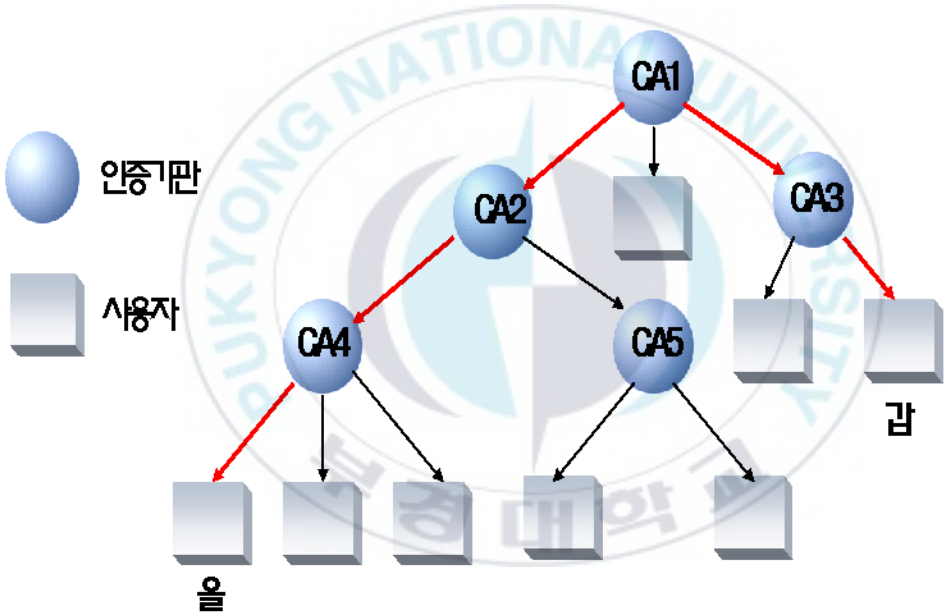
## 2.2 공개키 기반구조 구성 형태

공개키 기반구조 내에서 인증서에 대한 신뢰는 인증경로를 통해 전달된다. 전자서명을 검증할 경우, 전자서명의 검증자는 자신이 신뢰하는 인증기관의 공개키만을 알고 있으므로 그 인증기관의 공개키를 이용하여 인증경로를 검증함으로써 서명자의 공개키를 획득한다. 이렇게 획득한 공개키는 무결성이 보장된다. 검증자는 무결성이 보장된 공개키를 이용하여 서명을 검증할 수 있는 것이다. 이러한 신뢰가 인증 경로를 통해 어떻게 전달되는지에 따라 공개키 기반구조는 다음 두 가지 형태로 구성될 수 있다.

### 2.2.1 계층적 구성

인증기관들이 하위 인증기관에게 인증서를 발행하는 최상위 인증기관(PAA)아래에 계층적으로 배열되어 있는 구성으로 인증기관들은 자신의 아래 인증기관들에게 인증서를 발행한다. 계층적으로 구성된 공개키 기반구조에서 최상위 인증기관의 공개키는 모든 사람에게 알려져 있어 최상위

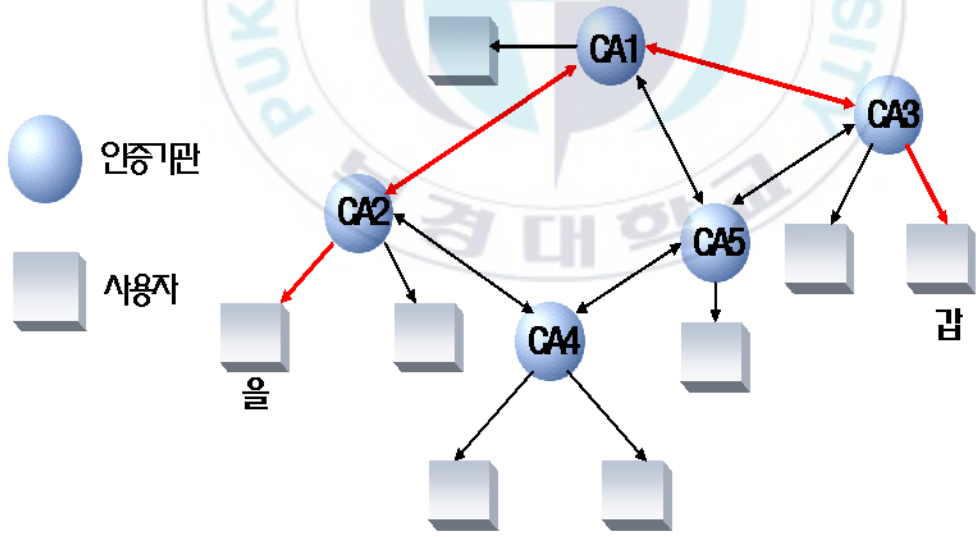
인증기관의 인증서로부터 사용자 인증서를 발행한 인증기관까지의 인증 경로를 검증함으로써 사용자의 인증서를 검증한다. PKI의 계층적 구성은 〈그림 2〉와 같다. 〈그림 2〉에서 갑이 을의 전자서명을 검증하기 위해 을의 공개키를 필요로 하며, 이를 획득하기 위해 을은 갑에게 서명문과 함께 인증기관 CA1에서 CA4까지의 인증 경로를 전송한다. 갑은 CA1과 CA3를 신뢰하고 을은 CA1과 CA4를 신뢰하므로 갑은 을이 자신과 같은 도메인에 있음을 확인한 후 자신이 알고 있는 CA1의 공개키를 이용해 CA1에서 CA4까지의 인증경로를 검증하여 을의 공개키를 획득한 후 서명문을 검증한다.




〈그림 2〉 계층적 구성

### 2.2.2 네트워크 구성

인증기관이 각각의 도메인을 형성하여 독립적으로 존재하는 구성으로 인증기관들이 서로를 상호 인증하여 서로에게 인증서를 발행한다. 네트워크로 구성된 PKI의 사용자는 자신의 인증서를 발행한(즉, 자신이 신뢰하는) 인증기관의 공개키만를 알고 있다. PKI의 네트워크 구성은 다음 〈그림 3〉와 같다. 〈그림 3〉에서 갑은 CA3를 신뢰하고 을은 CA2만을 신뢰하고 있다. 갑이 을의 서명문을 검증하고자 할 때, 을에서 갑으로의 인증 경로는 여러 개가 존재하므로 이중에서 가장 짧은 인증 경로를 찾는 탐색 과정이 필요하다. 가장 짧은 인증 경로는 CA3→CA1→CA2→을이다. 갑은 이 인증 경로를 이용해 을의 공개키를 획득하고 그를 이용해 전자서명을 검증한다. 네트워크로 구성되었을 경우에는 인증 경로가 여러 개 존재할 수 있으므로 이중 짧은 경로를 찾는 것이 중요 관건이다.




〈그림 3〉 네트워크 구성

### 2.2.3 계층적 구성 vs. 네트워크 구성

실제 PKI 구축시에는 기본적으로는 계층적 구조를 구성하면서 효율성과 다른 PKI와의 통신을 위해 한 도메인 내 또는 다른 도메인 내의 인증기관들 사이에 네트워크 구조를 허락한다[36][39]. 〈표 1〉은 PKI의 계층적 구성과 네트워크 구성에 대한 장점과 단점을 보여주고 있다.

〈표 1〉 계층적 구성과 네트워크 구성의 장•단점 비교

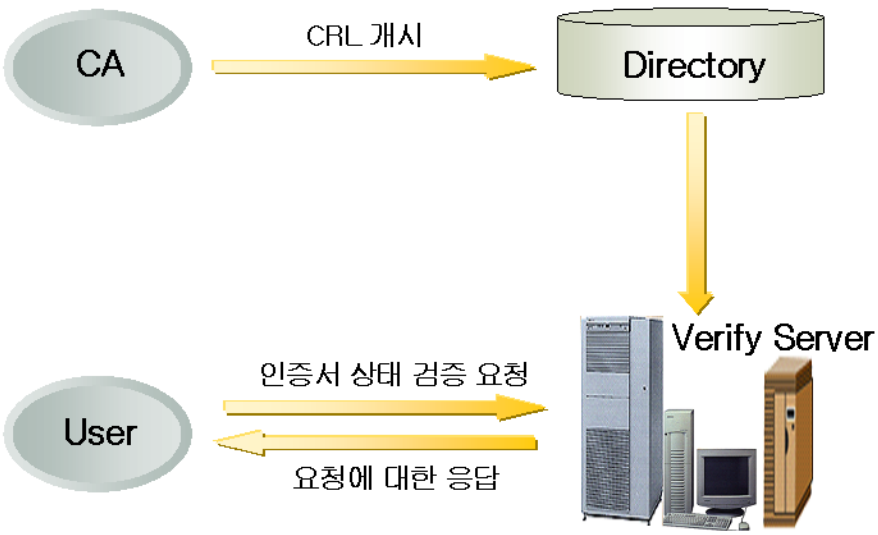
| PKI 구성 | 〈표 1〉 계층적 구성과 네트워크 구성의 장·단점 비교<br>장 점 | 단 점 |
|---|---|---|
| 계층적 구성 | • 많은 조직의 관리 구조가 계층적이므로 자연스럽게 부합<br>• 계층적 디렉토리 이름과 잘 부합<br>• 인증 경로 탐색 전략이 간단<br>• 인증 경로 검증이 용이 | • 각 국가별 PKI를 모두 통합하고자 하는 경우 하나의 루트 CA구축이 문제<br>• 상업적인 분야에서는 계층적일 필요가 없음<br>• 루트 키 노출 시 PKI내의 모든 사용자에게 새로운 공개키의 안전한 분배가 필요 |
| 네트워크 구성 | • 유연성을 가지며 사업 기관의 상호 신뢰 관계를 잘 반영<br>• 조직적으로는 멀리 떨어져 있지만 CA들이 서로 직접적으로 상호 인증될 수 있음<br>• CA의 비밀키가 손상되어 복구할 경우, CA는 새로운 공개키를 자신의 사용자들에게만 분배 | • 인증 경로 탐색 전략이 계층적 구성에 비해 훨씬 복잡<br>• 사용자는 PKI의 다른 사용자에게 단일 인증 경로를 제공할 수 없음 |

## 2.3 인증서 상태 검증 기술

### 2.3.1 Certificate Revocation List

Certificate Revocation List(CRL)을 이용하는 방식은 현재 가장 널리 사용되고 있는 인증서 상태 검증 방법이다. 이 방식은 사용자가 인증기관에게 인증서를 취소해 줄 것을 요청할 경우 인증기관이 인증서 취소 목록을 생성하여 배포함으로써 다른 사용자의 인증서 사용을 중지시키는 방식으로, 취소된 인증서의 일련번호와 사유를 포함하여 전자서명 한 후 디렉토리와 같은 공개된 장소에 게시하고 클라이언트는 이를 다운받아 필요로 하는 인증서를 검색하여 인증서의 상태 정보를 획득하는 방법이다. 이 방식은 인증서의 취소 상태를 파악할 수 있는 가장 일반적인 방법으로 인증서의 취소 사유가 발생하였을 경우 인증서 취소 목록에 새롭게 추가하여 주기적으로 발행하게 된다[14][35].

그러나 주기적으로 발행되는 인증서를 다운받기 때문에 인증서 상태 정보를 실시간으로 반영할 수 없으며, 인증서 처리 속도 측면에서는 인증서를 다운 받아서 상태 검증을 수행하기 때문에 다른 방식에 비해 월등히 빠르나 취소된 인증서를 다운 받는 과정에서 네트워크 과부하가 많이 발생하는 단점이 있다[29]. CRL 기반의 인증서 상태 검증 시스템의 수행과정은 다음 <그림 4>와 같다.




〈그림 4〉 CRL 기반의 인증서 상태 검증 시스템의 수행 과정

### 2.3.2 CRL Distribution Point

성능이 제한적인 환경에서 CRL을 이용하여 특정 인증서의 유효성을 검증하고자 하는 경우 CRL의 크기가 커지면 처리작업이 어려워진다. 이와 같은 경우, 크기가 큰 CRL를 여러 개로 나누어 전송받을 수 있다면, 인증서 사용자는 제한된 시스템 자원 내에서 효율적으로 인증서를 검증할 수 있다. 이때 CRL을 여러 개의 부분으로 분리하는데 사용하는 방법이 CRL 분배점(CRL Distribution Point)이다[6][15].

CRL 분배점은 CA가 CRL을 분할하고, 각기 다른 분배점을 이용하여 각 부분들을 발행한다. CRL분배는 영역이나 취소 이유 등에 의해 나뉠 수 있다. 만약 한 회사에 대해 CRL을 발행할 경우 CA가 각 부서별로 CRL을 나누어 발행한다면 사용자는 전체 CRL을 검사하지 않고 해당 부서의 CRL을 검사하면 된다. 인증서의 취소 이유로 나눈다면 이름의 변화에 의

해 취소된 인증서들의 CRL과 안정성 문제로 취소된 CRL을 나누어 발행할 수 있다. CRL 분배점은 인증서의 확장영역에 표기된다.

### 2.3.3 Delta CRL

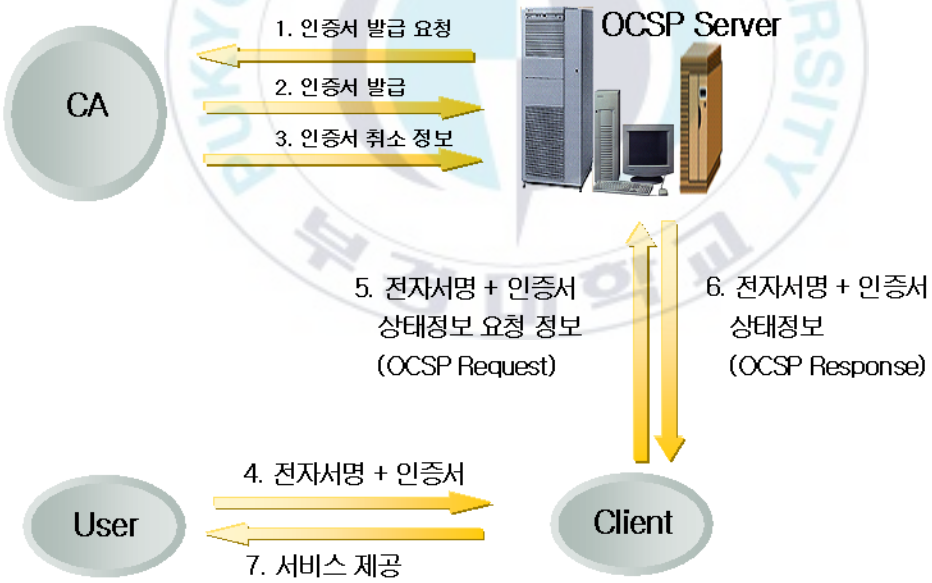
Delta CRL 방식은 CRL 방식의 통신 부담과 주기적으로 발행되는 문제를 개선하기 위해 제안된 방식이다. 이 방식은 전체 CRL을 대상으로 하지 않으며 최근 CRL이 발행된 시점에서 새로운 CRL이 발행된 시점까지의 변화된 목록만을 대상으로 관리하는 방식이다. CRL보다 빈번하게 발행되지만 크기가 작기 때문에 통신 부담을 줄이고 적절한 시기에 인증서의 상태를 제공할 수 있는 장점이 있다. 하지만 Delta CRL과 함께 전체 CRL도 발행해야 하는 문제점을 가지고 있다. 일반적으로, 트랜잭션 서버와 같이 많은 양의 메시지를 처리하는 고성능 서버에서는 CRL과 관련된 검증 작업을 백그라운드 작업으로 처리한다[32].

### 2.3.4 Online Certificate Status Protocol

Online Certificate Status Protocol(OCSP)은 CRL기반의 인증서 검증 방식의 문제점인 인증서에 대한 실시간 상태 검증을 해결하기 위해 제안된 인증서 상태 검증 방식으로 1999년 6월 IETF RFC 260 문서에 의해 공포되었다[19]. OCSP 기반의 인증서 검증 방식은 OCSP 클라이언트가 CRL을 요청하지 않고 인증서의 현재 상태를 검증하기 때문에 실시간으로 인증서에 대한 상태 검증을 할 수 있다는 장점이 있다. 반면 실

시간으로 인증서에 대한 유효성 검사를 수행해야 하기 때문에 많은 통신량으로 인한 네트워크 과부하 문제를 발생시킨다는 단점과 네트워크 상태에 따라 인증서 유효성 검사의 수행시간이 달라진다는 단점이 있다.

OCSP 인증서 상태 검증 방식은 사용자가 CA로부터 인증서를 발급받은 후, 정해진 형식으로 OCSP 클라이언트에게 전자서명을 하여 서비스를 요청하면, OCSP 클라이언트는 정해진 형식으로 OCSP 서버에게 인증서 상태를 요청하고, OCSP 서버는 요청받은 인증서에 대한 상태 정보를 검색하여 전자서명을 수행한 후 수행결과에 대한 응답을 OCSP 클라이언트로 넘겨줌으로써 실시간으로 인증서에 대한 유효성 검사를 수행하는 방식이다[19]. OCSP 기반의 인증서 검증 방식 수행과정은 〈그림 5〉와 같다.




〈그림 5〉 OCSP 기반의 인증서 검증 방식 수행 과정

### 2.3.5 Simple Certificate Validation Protocol

CRL을 사용하지 않고 인증서의 상태 정보를 제공하는 OCSP가 제안되었으나, 인증서 검증을 위한 다양한 정보가 요구됨으로써 인증서 상태 정보 외에 인증 경로에 관한 검증 정보들을 제공하는 Simple Certificate Validation Protocol(SCVP)가 제안되었다. SCVP의 목적은 클라이언트의 인증서 유효성 검증 관련 기능의 부담을 서버에게 위임함으로써 구현을 단순하게 만드는 것으로, 이를 통해 PKI 구현이 용이하게 되며 정책의 관리를 집중화 할 수 있는 장점을 가지게 된다. SCVP는 IETF에서 1999년 처음 초안이 제시되었으며, 대리인증 경로검증 서비스(Delegated Path Validation, DPV)와 대리인증 경로발견 서비스(Delegated Path Discovery, DPD)를 제공하는 프로토콜로 선정되었다[4][37].

SCVP는 SCVP 서버와 SCVP 클라이언트 간에 수행되는 프로토콜로 클라이언트와 서버간의 요청(Request)과 응답(Response) 메시지를 정의하고 있다. 클라이언트는 서버를 통해 온라인으로 특정 인증서의 유효성과 효력 정지 및 폐지 상태를 확인할 수 있으며 또한 인증서 유효성 검사 경로 등 다양한 정보를 이용할 수 있다. 이러한 서비스는 인증서 검증 과정에 대한 클라이언트의 부담을 덜어 줄 수 있다.

### 2.3.6 Data Validation and Certification Server

Data Validation and Certification Server(DVCS)는 신뢰할 수 있는 부인방지(Non-repudiation) 서비스를 구축하는데 필요한 하나의 구성요소로

사용될 수 있는 제 3의 신뢰기관(Trusted Third Party, TTP)을 제공할 수 있는 프로토콜로서 2001년 2월 IETF RFC 3029로 등록되었다[1][37].

DVCS의 역할은 서명된 문서 또는 공개키 인증서의 유효성과 데이터의 소유 또는 존재를 증명하는 것으로 DVCS로 제공될 수 있는 서비스로 데이터 소유 인증(Certification of Possession of Data), 전자서명된 문서의 정당성 검증(Validation of Digitally Signed Documents), 공개키 인증서의 유효성 검증(Validation of Public Key Certificates)이 있다.

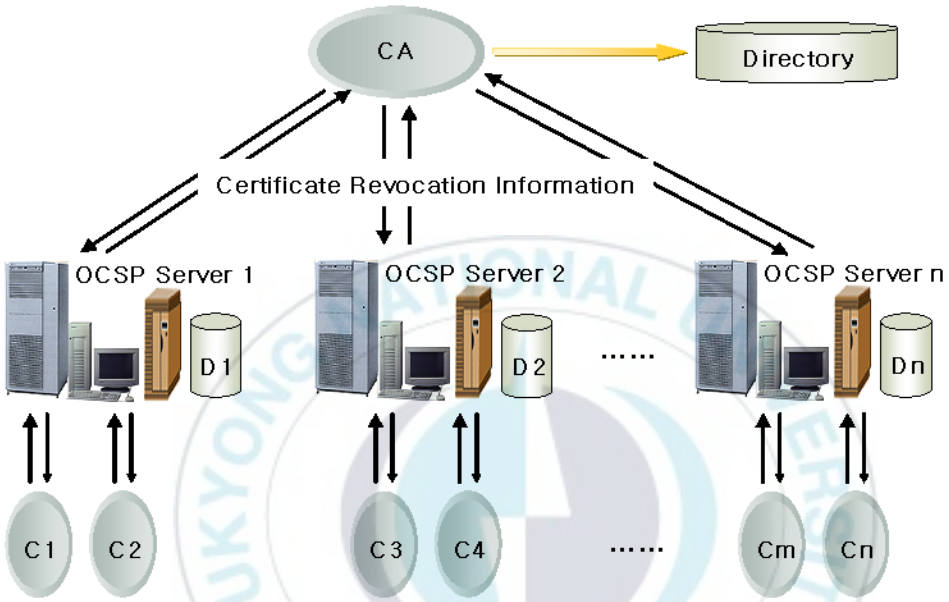
### 2.3.7 Distributed OCSP

모든 클라이언트들이 하나의 OCSP 서버 및 한곳에 서비스를 요청한다면 OCSP 서버는 부담을 가지게 된다. 게다가 최근에는 전자상거래를 이용하는 사용자가 급격히 증가하고 있는 상태에서 OCSP에게 서비스를 요청하는 일이 많이 발생한다. 그러므로 하나의 OCSP 서버 대신 여러 개의 OCSP 서버를 두는 방식이 제시되었다. 이러한 OCSP 서버 모델을 분산 OCSP(Distributed OCSP) 서버라고 한다[33][34].

분산 OCSP 서버는 CA에서 관리를 하게 되고, CA는 사용자가 인증서 취소를 요청하면 즉시 그 정보를 각 OCSP 서버에게 전달한다. 즉 여러 개의 OCSP 서버가 각 지점에 분산되어 있을지라도 분산되어 있는 OCSP 서버에 인증서 취소 정보를 보내고 각 OCSP는 동일한 인증서 취소 정보를 가지고 있고 CA와 같은 데이터를 보관하고 관리한다. 〈그림 6〉은 CA가 전체 인증서 취소 정보를 각 OCSP 서버에게 전달하고 있으며 동시에 주기적으로 CRL을 발생하여 디렉토리에게 전달하고 있다. 각 OCSP 서버

는 같은 데이터베이스를 관리하고 있다.

집중 OCSP 서버와 비교할 때 분산 OCSP 서버의 장점은 클라이언트의 요청을 분산시킴으로써 하나의 OCSP 서버에 모이는 부하를 줄일 수 있다.




〈그림 6〉 분산 OCSP 구조

## 2.4 요 약

본 장에서는 인터넷 환경에서의 인증 및 신뢰성 있는 통신과 같은 보안 서비스를 제공하기 위한 공개키 기반구조(Public Key Infrastructure, PKI)를 구성하는 인증기관, 등록기관, 디렉토리, 사용자 등과 같은 구성

요소와 네트워크 구성 방법에 대하여 소개하였고, 보안 서비스를 수행할 시점에서의 인증서의 취소 유무를 검증하기 위한 인증서 상태 검증 기술의 특징을 살펴보았다.

최근 이동 디바이스들의 기능과 휴대성이 나날이 발전하고, 다양한 형태의 네트워킹 능력을 소유한 이동 디바이스들로 구성된 새로운 형태의 네트워킹 환경이 고안되고 있다. 따라서, 새로운 형태의 네트워킹 환경에서의 보안 서비스를 제공하기 위한 보안 기반구조에 대한 연구의 필요성이 대두되고 있으며, 인터넷 환경의 PKI를 새로운 형태의 네트워킹 환경에 적용하기 위한 연구가 진행되고 있다.

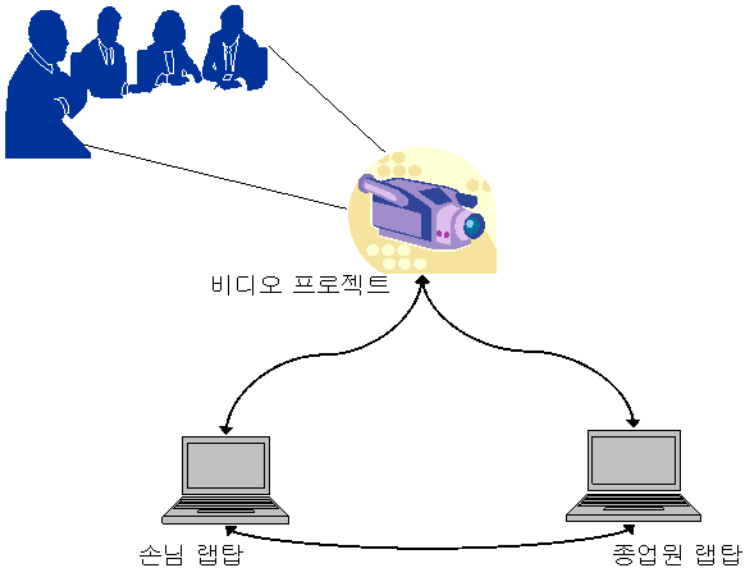
다음 장에서는 네트워크 접속에 사용되는 디바이스들이 물리적으로 사용자에 근접한 범위 내에 위치하는 Personal Area Network(PAN)에서 보안 서비스를 제공하기 위한 Personal PKI (Personal Public Key Infrastructure)의 개념 및 구성 기술에 대하여 살펴본다.

# 3. PAN 및 Personal PKI

## 3.1 PAN의 구조

Personal Area Network(PAN)은 어떤 개인의 근접지역에 존재하는 제한된 수의 장치들 간의 상호연결 편이성을 제공하기 위한 개인 영역의 네트워크로써, 데이터 전송속도보다는 이기종간 네트워크 연결 및 상호 운용성, 개인 데이터 교환의 편이성 등에 더 중점을 두고 있으며, 네트워크 인프라의 척추에 해당하는 백본망에서부터 가입자망까지 확장된 네트워크를 최후의 수십 미터 단위의 정보 실핏줄에 해당하는 개인 영역까지 확장시킬 수 있는 기술이다. LAN이나 홈 네트워크 등과 비교한다면, PAN은 10m 이내의 개인영역이라는 네트워크 범위와 이동환경에 더 중점을 둔 것이라고 볼 수 있다.

국내 인터넷 사용자의 폭발적인 증가와 인터넷 기술의 진보로 인해, 정보망 발전에 있어 세계적인 선두에 있는 미국보다 훨씬 높은 증가율을 보이고 있으며, 이러한 추세는 이제 홈 네트워크와 PAN 영역으로 확장되고 있다. PAN은 휴대폰과 PDA, PC 등의 데이터를 동기화하고 공유하며, 기존의 블루투스 및 HomeRF 등의 서로 다른 표준간의 호환성을 개선하고자 하는 목적에서 시작되었다. 휴대성을 중시하는 PAN의 특성상 물리계층은 유선보다 무선 방식이 선호된다. PAN에서의 〈그림 7〉과 같은 미팅 시나리오를 고려해보자.




〈그림 7〉 미팅 시나리오

두 명(종업원, 손님)이 비디오 프로젝터가 설치된 방에서 미팅을 한다. 두 명은 근거리 무선 인터페이스를 가지는 랩탑을 소유하고, 각각은 자신의 발표 자료를 랩탑에 저장하고 있으며, 상대를 위해서 발표를 한다. 손님은 발표가 끝나면 종업원에게 발표 자료를 전달하기를 원한다. 이 시나리오에는 비디오 프로젝터, 손님 랩탑, 종업원 랩탑과 같이 세 가지의 컴포넌트들이 존재하며, 비디오 프로젝터의 경우는 종업원 랩탑이나 손님 랩탑과 연결될 수 있으며, 비디오 프로젝트와 종업원 랩탑의 경우는 둘 다 동일한 조직 내에 존재하기 때문에, 종업원 랩탑은 손님 랩탑보다도 더욱 비디오 프로젝트를 신뢰할 것이라고 가정할 수 있다. 하지만, 발표 자료는 소중한 정보이기 때문에, 종업원과 손님은 보안 프로토콜에 의하여 근거리 무선 통신이 도청으로부터 보호되기를 원할 것이다. 보안 프로토콜을 설정하기 위하여 종업원 랩탑과 비디오 프로젝터 사이의 보안 연관성을 설정

하기 위하여 사용자의 개입(Interaction)이 거의 불필요하나, 손님 랩탑과 비디오 프로젝터(또는 종업원 랩탑)사이의 보안 연관성을 설정하기 위해서는 사용자의 개입이 요구된다.

PAN내에서의 통신은 기밀정보와 개인 데이터가 주를 이루므로 신뢰성 있고 인증된 통신 서비스가 필수적이며, 이러한 보안 서비스를 위한 기반 구조로써 Personal PKI가 제안되었다[12].

## 3.2 Personal PKI

### 3.2.1 Personal PKI의 개념

PKI에서의 중앙 집중적인 CA의 경우, 통신 개체들이 사용하는 모든 인증서들을 CA가 발행해야 하며, 모든 개체들은 신뢰되는 CA의 공개키를 공유해야만 한다. 또한, 아주 많은 수의 사용자들을 처리할 수 있는 잘 제어되고 안전한 인증 처리를 유지관리하기 위한 비용 소모가 급증한다. 따라서, 사용자의 로컬 환경(즉, PAN)을 관리하고자 하는 사용자는 PAN을 위한 CA를 사용함으로써, PAN을 위한 인증 인프라를 구축 가능하며, 사용자는 PAN을 위한 CA 기능을 자신의 로컬 환경 밖의 중앙 집중적인 CA에게 위임하는 것을 원하지 않을 것이다.

Personal PKI는 PAN내의 컴포넌트들 중 하나가 Personal CA로 동작하며, 그 컴포넌트는 모든 다른 컴포넌트들을 위하여 인증서를 발행하고, 발행된 인증서의 상태를 관리한다. 일반적으로 Personal CA는 디스플레이와 키패드를 소유하고, 생산자에 의해서 미리 구성된 키 쌍(Key Pair)을 소유

하거나, 키 쌍을 안전하게 생성할 수 있어야 한다. 따라서, PAN 내의 모든 컴포넌트들은 Personal CA의 공개키를 공유하며, 인증서 내의 공개키들은 세션키 교환과 PAN 내의 다른 컴포넌트 인증을 위해서 사용된다.

### 3.2.2 Personal CA의 동작

〔CA 초기화〕

Personal CA를 위한 키 쌍을 생성하는 것으로 스마트카드나 휴대용 결함 저항 장치(Portable Tamper-resistant Device)들을 사용한다.

〔장치 초기화〕

이동 장치는 다음의 단계를 수행한다.

(1 단계) 이동 장치는 필요한 키 쌍을 생성하고, Personal CA가 어떤 장치인가에 대한 정보를 받거나, PAN을 통해서 Personal CA로 동작하는 장치를 발견한다.

(2 단계) Personal CA의 공개키가 이동 장치에게 전달된다. 이때, 이동 장치는 Personal CA의 공개키에 대한 무결성과 출처 검증을 수행한다.

(3 단계) 이동 장치는 자신의 공개키와 인증서에 포함될 정보를 Personal CA에게 전달한다. 이때, Personal CA는 이동 장치의 공개키에 대한 무결성과 출처 검증을 수행한다.

(4 단계) Personal CA는 그 이동 장치를 위한 인증서를 생성하고, 이동 장치에게 전달한다.

(5 단계) 이동 장치는 수신한 인증서를 Personal CA의 공개키로 검증한다.

[소유의 증명]

Personal CA가 인증서를 발행하기 이전에 인증서 요청자가 대응되는 개인키를 알고 있는 가를 확인해야 한다. 따라서, 이동 장치는 위의 (3 단계)에서 개인키를 위한 소유의 증명(Proof of Possession, PoP)를 제공해야만 한다. 소유의 증명은 인터넷 환경을 위한 PKI에서도 사용되고 있는 것으로 이동 장치의 공개키와 식별자 등과 같은 정보에 대한 자기 서명된 인증서(Self-signed Certificate)가 소유의 증명으로 사용될 수 있다. 소유의 증명은 '공격자가 다른 사람의 공개키에 대한 인증서를 요청하고, 동시에 다른 사람의 신원을 스푸핑'하거나, '공격자가 다른 공격자의 신원에 대한 스푸핑없이 인증서를 요청하고, 그 외의 신원을 사용'하는 소스 대체 공격(Source Substitution Attack)을 방지할 수 있다. 소유의 증명은 키가 생성되는 장소와 시간을 참작하는 것이 중요하며, 아래의 세 가지 경우가 고려할 수 있다.

- 사용자의 장치에서 키 쌍을 생성 : 인터넷 환경의 PKI와 동일한 시나리오
- 키 쌍을 생산자가 고객에게 전달되기 이전에 생성 : 소유의 증명의 필요성은 사용자가 공개키/개인키를 읽을 수 있는가에 따라 결정된다. 즉, 사용자나 공격자가 아무런 제약 없이 개인키를 읽는 것이 불가능하면, 인가될 키는 안전한 방법으로 전송되어 요청하는 장치에게 설정된 것으로 판단될 수 있으므로 소유의 증명이 불필요하다.

■ 인증서가 요청되면, 키 쌍을 인가된 제3자가 생성 : 소유의 증명은 불필요하지만, 개인키 전송을 위한 인증된 기밀 채널이 지원되어야만 한다.

## 3.3 인증서 관리

일단 이동 장치가 Personal CA에게 등록(Imprinting)되고 공개키 인증서를 발급받으면, 키 쌍들과 인증서들에 대한 지속적인 관리가 필요하다. PAN에서 인증서 관리를 위하여 아래의 세 가지 이슈가 해결되어야 한다.

■ 인증서 및 키 쌍 갱신 : 인증서의 만기가 지난 후에, 동일한 키 쌍 또는 새로운 키 쌍을 위한 새로운 인증서를 발급 가능해야 한다.
■ 인증서 상태 검증 : 이동 장치의 개인키가 침해되었거나 분실되었을 때, PAN 내의 모든 장치들은 그 이동 장치의 인증서가 취소되었다는 것을 알 수 있어야 한다. 또한, Personal CA가 침해되었을 때에도, Personal CA의 개인키는 취소되어야 한다. 따라서, '어떠한 키들이 취소되는가?'에 대한 정보는 즉시적이고 효율적으로 분배되어야 한다.
■ 신뢰 관리 : 이동 장치와 Personal CA사이에 Personal CA의 키 갱신 및 Personal CA 장치의 분실로 인한 Personal CA의 대체와 같은 상호 관련성은 관리되어야 한다.

### 3.3.1 인증서 및 키 쌍 갱신

이동 장치가 현재 소유하고 있는 공개키에 대한 새로운 인증서를 얻기 위하여 아래의 두 가지 경우를 고려할 수 있다.

- Personal CA가 주축 : 이동 장치들은 비교적 소수이기 때문에, Personal CA는 자신이 생성했던 인증서들을 위한 모든 공개키들을 안전하게 보유 가능하므로, Personal CA는 주기적으로 만기된 인증서들이 있는지를 확인 가능하다. 만약, 새로운 인증서가 필요하면, Personal CA는 사용자에게 '현재 사용하는 키 쌍을 위한 인증서를 재생(Renewal)할 것인가?'를 문의한다. 만약 그 사용자가 동의하면, 새로운 인증서가 무선 인터페이스를 통해서 생성 및 전달된다.
- 사용자가 주축 : 모든 공개키들을 Personal CA에 저장하는 것이 불가능한 경우가 발생하면, 새로운 인증서를 필요로 하는 이동 장치는 만기된 인증서를 Personal CA에게 전달한다. 그러면, 사용자가 인증서의 재생(Renew)에 동의 유무를 확인하기 위하여, Personal CA는 인증서에 관련 정보(공개키 및 장치 신원 등)를 사용자에게 전달한다. 만약, 그 사용자가 동의하면, 새로운 인증서가 생성 및 전달된다.

새로운 공개키가 이동 장치에 사용되기 위한 재생 처리는 복잡해진다. 즉, 예전의 키 쌍이 Personal CA와 이동 장치간의 안전한 정보 교환을 위해서 사용 가능할 경우, 예전의 키 쌍이 충분히 안전한데 새로운 키 쌍이 왜 필요한지에 대한 문제점이 발생한다. 따라서, 다수의 이동 장치들의 경우는 동일한 키 쌍을 막연히 재생 없이 사용할 것이다. 반면, 새로운 키

쌍이 반드시 필요한 경우 또는 예전의 키 쌍이 Personal CA와 이동 장치간의 안전한 정보 교환을 위하여 사용 불가능할 경우, 새로운 등록(Imprinting)이 반드시 필요하지만, 이러한 경우는 빈번하지 않을 것이다.

### 3.3.2 인증서 상태 검증

인증서 상태 정보를 유포하기 위하여 온라인 상태 유포(Online Status Dissemination) 및 Ad hoc 상태 유포(Ad hoc Status Dissemination)와 같은 두 가지 형태를 고려할 수 있다.

■ 온라인 상태 유포 : Personal CA가 모든 이동 장치에게 영구적이거나 빈번한 간격으로 온라인일 경우에 고려될 수 있다.

- 영구적으로 온라인일 경우 : OCSP를 사용한다.
- 빈번한 간격으로 온라인일 경우 : Personal CA가 CRL을 일정한 기간마다 생성하여, 모든 이동 장치들에게 자동적으로 분배한다. Personal CA와 이동 장치들 모두는 항상 온라인이 아니기 때문에, 이러한 기법은 Personal CA가 충분히 자주 온라인이 될 수 있는 환경에 적합하다.

■ Ad hoc 상태 유포 : Personal CA가 간헐적 또는 드물게 온라인일 경우에 고려될 수 있다. 즉, Personal CA와 이동 장치들 모두가 항상 온라인이 아니므로, 직접적인 CRL 분배는 적합하지 않기 때문에,

새로운 CRL 분배 방법이 고려되어야 한다. 따라서, Personal CA는 규칙적인 간격으로 CRL들을 생성하고(여기서, Personal CA는 가장 최근의 CRL을 분배할 만큼은 자주 온라인 되어야 함), 적어도 한 개의 이동 장치에게 전달하면, 연속되는 CRL의 분배는 이동 장치들 간의 Ad hoc 형태로 이루어진다. 하지만, 최초에 Personal CA로부터 CRL을 수신한 장치가 악의적이면, Ad hoc 형태로 CRL이 전달되지 못하는 경우가 발생하기 때문에, 최초에 Personal CA는 단일 홉 사이에 있는 모든 장치에게 CRL을 배포한다.

### 3.3.3 신뢰 관리

Personal CA의 공개키/개인키를 갱신하고자 할 경우, 예전의 공개키가 취소되지 않았으면, '예전의 개인키로 서명된 새로운 Personal CA 공개키를 위한 인증서를 분배'하거나, '모든 이동 장치들과 새로운 등록 절차를 수행'한다. 하지만, Personal CA의 개인키가 침해되거나 분실된 경우는 모든 이동 장치들에게 즉시적으로 이 사실을 알려야 하며, 이와 같은 경우 장치들 사이의 안전한 통신은 불가능하게 된다. 따라서, 신뢰 관리를 위하여 다음의 두 가지 접근법이 존재한다.

- 다수의 Personal CA들 : 모든 장치들은 다수의 Personal CA들의 공개키를 소유하고, 그 공개키들을 위한 다수의 인증서들을 소유한다. 만약 2개 이상의 Personal CA들이 이동 장치가 등록할 시간에 가용하면, 동시적으로 등록과 인증서 생성을 수행한다. 단일 Personal CA의 공개키가 취소되면, 이동 장치들은 나머지 Personal CA들로부터 취소된 Personal CA에 대한 취소 정보를 수신한다.

■ 대체 Personal CA : Personal CA의 개인키가 침해되거나 분실된 경우, 대체 Personal CA를 설정하고, 가능한 빨리 모든 장치들을 재등록시킨다. 이와 같은 경우 이동 장치들은 예전의 Personal CA의 취소와 새로운 Personal CA로의 등록을 동시적으로 수행한다.

## 3.4 수동 인증

초기화를 위한 중요한 단계는 장치들의 초기 인증을 수행하는 것이며, 장치에게 자신의 신원을 PAN 내의 다른 이동 장치들에게 증명할 때 필요한 정보들을 제공하는 것이다. PAN 내의 이동 장치들이 초기 인증을 위하여 사용하는 통신 링크는 케이블, USB, 바코드 리더 또는 스마트카드 리더 등과 같은 보조적인 안전한 채널을 사용되며, 사람의 개입을 필요로 한다. 이러한 초기 인증 단계를 수동 인증이라 하며 PAN 내의 이동 장치들의 설비에 따라서 다음과 같은 세 가지 형태의 수동 인증 프로토콜(Manual Authentication Protocol)이 존재한다. 여기서, 단일 사용자가 두 개의 장치들을 가지고 있는 것을 가정하나, 두 장치에 대한 각각의 사용자들이 존재할 경우는 그 사용자들은 상대방과 안전한 채널을 통해서 통신해야만 한다[18].

■ MANA Ⅰ : 첫 번째 장치는 알파벳과 숫자 심볼로 구성된 짧은 문자열에 적합한 출력 인터페이스와 단순한 입력 인터페이스를 소유하며, 두 번째 장치는 알파벳과 숫자 심볼로 구성된 짧은 문자열에 적합한 입력 인터페이스와 단순한 출력 인터페이스를 소유한다.

■ MANA II : 두 장치들은 알파벳과 숫자 심볼로 구성된 짧은 문자열을 출력할 수 있는 출력 인터페이스를 소유하며, 사용자가 확인 메시지(확인 메시지)를 입력할 수 있는 단순한 입력 인터페이스를 소유한다.

■ MANA III : 두 장치들이 알파벳과 숫자 심볼로 구성된 짧은 문자열에 적합한 입력 인터페이스와 단순한 출력 인터페이스를 소유한다.

### 3.4.1 MANA I

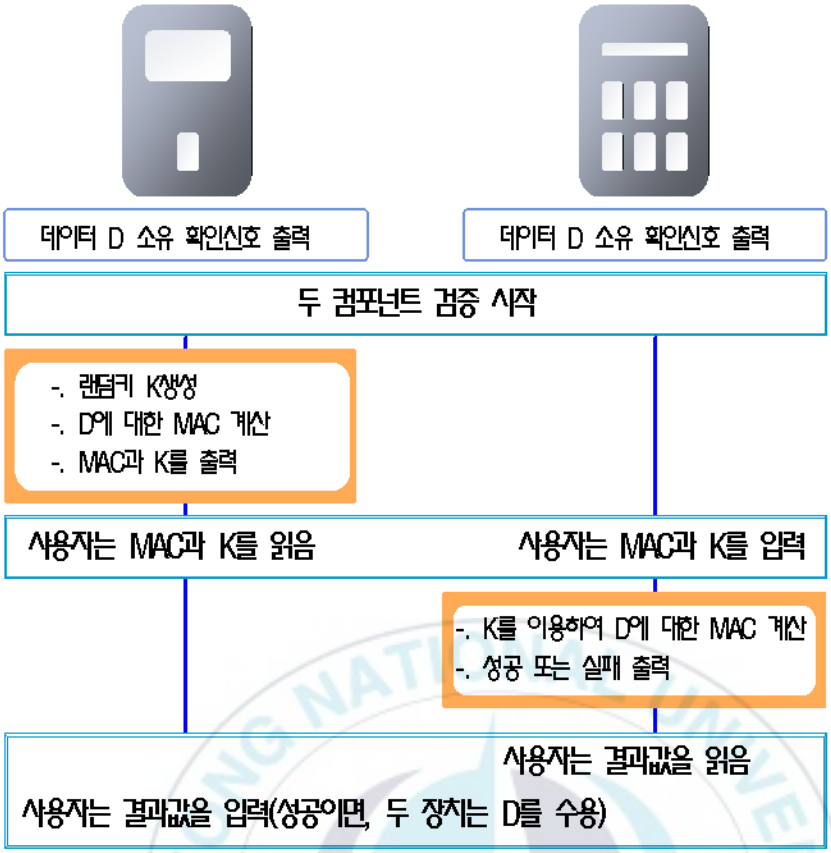
〈그림 8〉은 MANA I에 대한 동작 절차를 보여주고 있으며, 상세한 내용은 다음과 같다.

(1 단계) 두 장치들은 데이터 $D$를 수신했으며, 검증을 수행할 준비가 되었다는 것을 알리는 신호를 출력한다.

(2 단계) 첫 번째 장치는 랜덤키 $K$를 생성하고, 데이터 $D$에 대한 MAC을 계산하고, 첫 번째 장치의 출력 인터페이스를 통하여 MAC과 $K$를 출력한다. 그리고, 사용자는 출력된 MAC과 $K$를 읽는다.

(3 단계) 사용자는 첫 번째 장치의 출력물을 두 번째 장치의 입력 인터페이스를 사용하여 입력하고, 두 번째 장치는 자신이 소유한 데이터 $D$에 대한 MAC을 입력된 $K$를 사용하여 계산한다. 두 MAC이 일치하면 두 번째 장치는 성공 신호를 사용자에게 출력하나, 두 MAC이 일치하지 않으면 실패 신호를 출력한다.

(4 단계) 사용자는 첫 번째 장치에게 결과를 입력한다. 만약 성공일 경우, 두 장치들은 데이터 $D$를 수용하게 된다.




〈그림 8〉 MANA Ⅰ의 동작 절차

### 3.4.2 MANA Ⅱ

〈그림 9〉는 MANA Ⅱ에 대한 동작 절차를 보여주고 있으며, 상세한 내용은 다음과 같다.

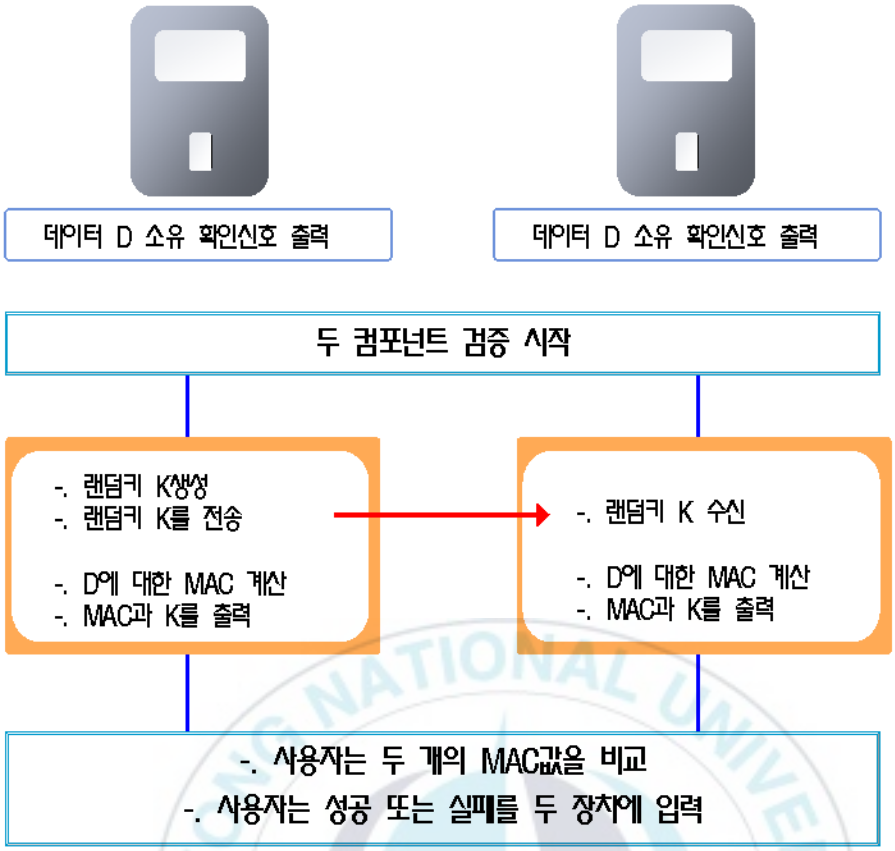
(1 단계) 두 장치들은 데이터 $D$를 수신했으며, 검증을 수행할 준비가 되었다는 것을 알리는 신호를 출력한다. 두 장치들로부터 위와 같은 신호를 수신한 사용자는 두 장치 중 하나에게 어떤 신호를

입력한다. 여기서, 사용자가 신호를 입력한 장치를 첫 번째 장치라고 한다.

(2 단계) 첫 번째 장치는 랜덤키 $K$를 생성하고, 데이터 $D$에 대한 MAC을 계산한다. 그리고, 첫 번째 장치는 MAC을 출력하고, 키 $K$를 안전하지 않은 무선 링크를 통해서 전송한다.

(3 단계) 두 번째 장치는 $K$를 사용하여, 자신이 저장하고 있는 $D$에 대한 MAC을 계산하고, $K$와 계산된 MAC값을 전송한다.

(4 단계) 사용자는 두 장치들의 $K$와 MAC값을 비교한다. 두 값이 일치하면, 사용자는 두 장치에 채택을 나타내는 신호를 입력한다. 그러면, 두 장치는 데이터 $D$를 수용한다. 만약 MAC값이 일치하지 않으면 $D$는 수용되지 않는다.




〈그림 9〉 MANA Ⅱ의 동작 절차

### 3.4.3 MANA III

〈그림 10〉은 MANA III에 대한 동작 절차를 보여주고 있으며, 상세한 내용은 다음과 같다.

(1 단계) 두 장치들은 데이터 $D$를 수신했으며, 검증을 수행할 준비가 되었다는 것을 알리는 신호를 출력한다. 두 장치로부터 신호를 수신한 사용자는 짧은 랜덤 키 $K$를 생성한다. 사용자는 $K$를 두

장치에 입력하고, 두 장치 중 하나에게 어떤 신호를 입력한다. 여기서, 사용자가 신호를 입력한 장치를 첫 번째 장치라고 한다.

(2 단계) 첫 번째 장치는 랜덤한 수 $R_1$을 생성하고, $K$를 사용하여 데이터 $D$와 $R_1$에 대한 $MAC_1$을 계산하고, 안전하지 못한 무선 링크를 통하여 두 번째 장치로 전송한다.

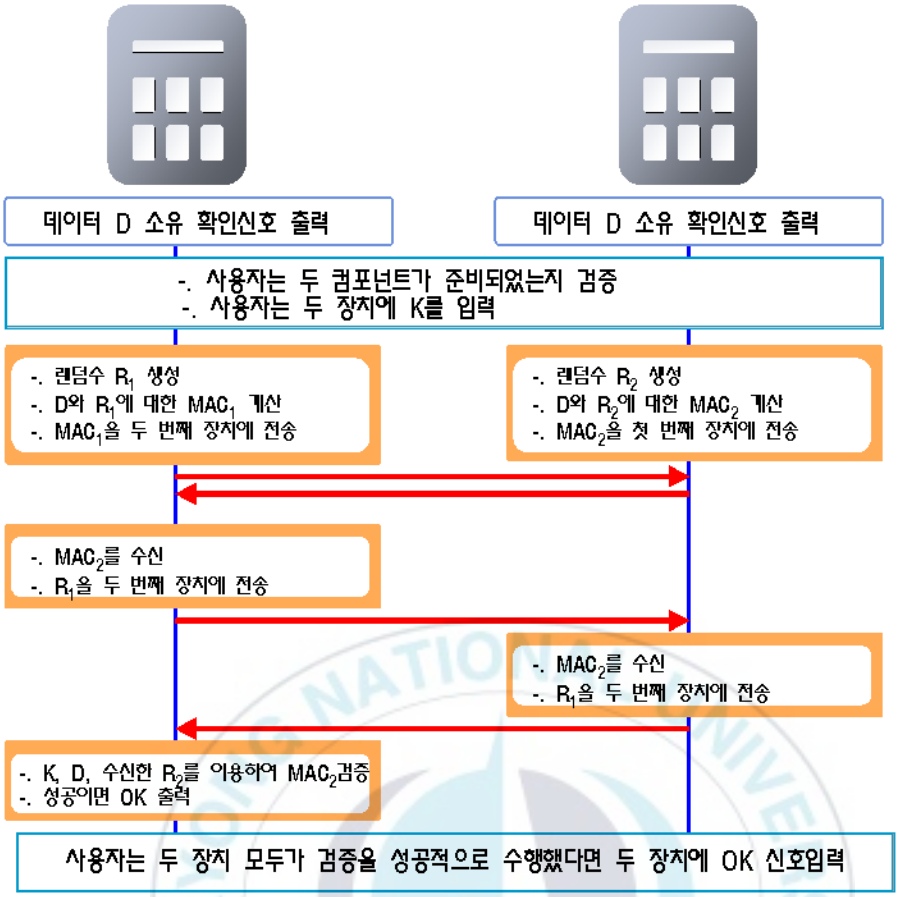
(3 단계) 두 번째 장치는 랜덤한 수 $R_2$를 생성하고, $K$를 사용하여 자신이 저장하고 있는 $D$와 $R_2$에 대한 $MAC_2$을 계산하고, 안전하지 못한 무선 링크를 통하여 첫 번째 장치로 전송한다.

(4 단계) 첫 번째 장치는 자신의 랜덤한 수 $R_1$을 두 번째 장치로 전송한다.

(5 단계) 두 번째 장치는 저장된 $K$, $D$, 수신한 $R_1$을 사용하여, 수신한 $MAC_1$을 검증한다. 만약 검증이 성공적이면, 두 번째 장치는 채택을 나타내는 신호를 출력하고, 자신의 $R_2$를 첫 번째 장치로 전송한다.

(6 단계) 첫 번째 장치는 저장된 $K$, $D$, 수신한 $R_2$을 사용하여, 수신한 $MAC_2$을 검증한다. 만약 검증이 성공적이면, 첫 번째 장치는 채택을 나타내는 신호를 출력한다.

(7 단계) 사용자는 두 장치 모두가 검증을 성공적으로 수행한 것을 확인하고, 두 장치로 수용을 위한 신호를 입력한다.




〈그림 10〉 MANA Ⅲ의 동작 절차

## 3.5 요 약

본 장에서는 개인의 근접지역에 존재하는 제한된 수의 장치들로 구성된 PAN과 PAN에서의 보안 서비스를 제공하기 위한 Personal PKI에 대한 개념을 소개하였고, Personal PKI에서의 인증기관인 Personal CA에 대한 요구사항과 인증서 발행, 검증 및 상태 검증 기술을 구현하기 위하여 인터넷 환경의 PKI를 적용하기 위한 방법론에 대하여 언급하였다.

인터넷 환경과 PAN에서 PKI를 사용하는데 있어서, 인증서 상태 검증을 위한 통신 및 계산비용을 감소시키는 것은 전체 보안 서비스를 위한 비용을 감소시키는 것에 직접적인 영향을 미치고 있어, 효율적이고 적시적인 인증서 상태 검증 기술의 개발은 많은 연구자들에게 주요한 연구 이슈가 되고 있다. 따라서, 다음 장에서는 Zhou가 인터넷 환경에서 인증서 상태 검증을 효율적이고 적시적으로 수행하기 위하여 제안한 공개키 프레임워크를 소개하고, Zhou의 공개키 프레임워크 내의 보안 파라메타들을 재조명하여, 실제 구현 환경에 적합하도록 개선할 것이며, 이는 인터넷 환경뿐만 아니라 PAN과 같은 새로운 형태의 네트워크 환경에서의 보안 서비스를 제공하기 위한 기반구조에서의 성능 향상을 위해 적용 가능할 것이다.

# 4. 새로운 공개키 프레임워크

오늘날 공개키 기반구조(Public Key Infrastructure, PKI)는 전자서명을 통한 인증, 부인방지 서비스의 제공과 함께 대칭키와 비대칭키의 결합을 통한 기밀성 보장 및 키 관리 서비스의 사용을 위한 기반구조로써 널리 이용되어지고 있다[2]. 공개키 기반구조에서 인증서는 공개키와 소유자의 신원정보를 결합한 것으로 인증기관(Certificate Authority, CA)이라 불리는 신뢰된 제3자에 의하여 발급된다. X.509 인증서는 IETF PKIX 워킹 그룹에 의하여 제정된 인증서 표준 형식으로써 X.509 인증서에 기반한 PKI는 인터넷 표준으로 자리 잡고 있다. 한편 CA에 의해 발급된 사용자 인증서는 만기일 이전에 취소되어질 수 있을 것이며, 이러한 사용자 인증서에 대한 취소 정보의 효율적이고 적시적인 분배는 PKI에서 매우 중요한 부분이다. 그러나, 현재 존재하는 인증서 기법들은 PKI 사용자뿐만 아니라 인증기관에게 상당한 작업처리와 저장 공간의 오버헤드를 요구한다. 따라서 인증서 취소 정보를 적시적으로 분배할 수 있는 메커니즘의 개발은 더욱 효율적인 PKI 환경을 구현하기 위하여 해결되어져야 하는 문제로 거론되어지고 있다.

## 4.1 Zhou의 공개키 프레임워크 취약성 분석

Zhou는 인증서 취소에 대한 효율적인 방안을 제시하고자 두 가지 모델을 제안하였다. 기본 프레임워크(Basic Framework)라는 해쉬체인에 기반한 새로운 인증서 상태 검증 기법을 제안하였으며, 기본 프레임워크의

문제점을 해결하기 위하여 관리자 제어된 인증서(Manager Controlled Certificate)를 제안하였다 [31].

### 4.1.1 Zhou의 기본 프레임워크

본 절에서는 Zhou가 제안한 기본 프레임워크(J. Zhou et al.'s Basic Framework, ZBF)에 대한 소개를 한다. ZBF에서의 공개키 인증서의 취소 유무에 대한 상태 검증은 오직 인증서 소유자에 의해서만 제어된다.

[용어 정의]

- $U$ : 사용자
- $V$ : 검증자
- $r$ : 사용자 $U$가 정의한 패스워드
- $SK_U$ : 사용자 $U$의 비밀키
- $PK_U$ : 사용자 $U$의 공개키
- $T$ : 인증서의 유효기간
- $D$ : 인증서 유효기간의 시작시점
- $L$ : 인증서 유효성 갱신기간
- $j = T/L$ : 정수값
- $SIGN_A(M)$ : 메시지 $M$에 대한 통신개체 $A$의 전자서명
- $CERT_U$ : 사용자 $U$의 인증서

사용자 $U$의 공개키 인증서는 아래와 같은 절차로 생성된다.

〔인증서 생성 과정〕

(1 단계) $U$는 자신을 위한 키 쌍을 생성한다 : $SK_U$, $PK_U$

(2 단계) 인증서 유효성 갱신 지점들은 아래와 같이 정의한다.

$$D_1 = D + L, D_2 = D + 2*L, \cdots, D_j = D + j*L$$

(단, $j = T/L$ : 정수값으로 가정)

(3 단계) $U$는 일방향 해쉬체인을 생성한다. 이때, $r$은 오직 $U$만이 알고 있어야한다.

$$H^0(r) = r, \quad H^i(r) = H(H^{i-1}(r)), \quad i=1,2,\cdots,j$$

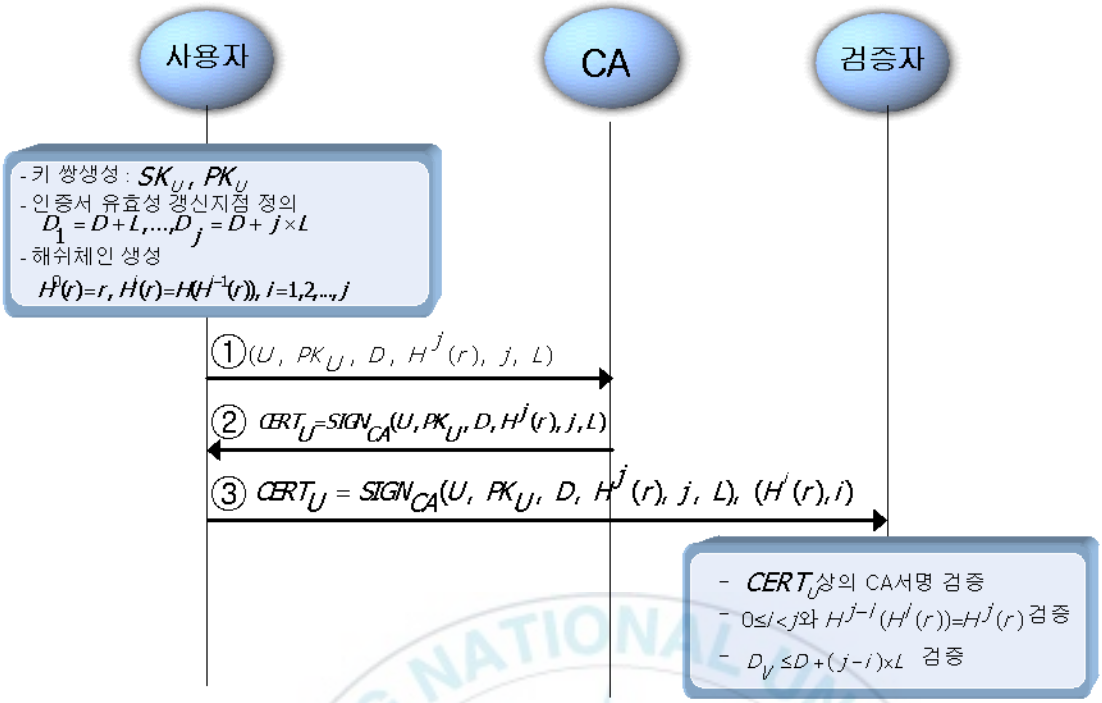
(4 단계) $U$는 $(PK_U, D, H^j(r), j, L)$을 CA에게 전송한다.

(5 단계) CA는 $U$의 요청을 인증한 후 아래와 같이 $U$의 인증서를 생성하여, $U$에게 전달한다.

$$CERT_U = SIGN_{CA}(U, PK_U, D, H^j(r), j, L)$$

ZBF에서는 기존의 공개키 인증서와 비교할 때, $CERT_U$는 $(H^j(r), j, L)$와 같은 추가 파라메타를 포함한다. 추가된 파라메타는 ZBF에서 $CERT_U$의 유효성을 제어하는데 사용되어 진다.

$CERT_U$의 다음 유효성 갱신 지점이 $D_e$라고 가정하면, $U$는 $SK_U$를 사용하여 전자서명문을 생성할 시 $i = j-(D_e-D)/L$인 $(H^i(r), i)$를 전자서명문에 첨부시킨다. $U$의 전자서명 검증을 위해 검증자 $V$는 먼저 $CERT_U$의 취소 유무를 확인해야 한다. $CERT_U$를 검증하는 지점이 $D_V$라고 가정하면, $V$는 아래의 검증과정을 수행하여 $CERT_U$의 상태를 확인한다.




〈그림 11〉 ZBF에서 인증서 생성 및 검증 과정

〔인증서 검증과정〕

(1 단계) $V$는 $CERT_U$ 상의 CA의 서명을 검증한다. 만약 올바르다면, $V$는 $U$의 공개키가 $PK_U$이고, 인증서 유효기간의 시작지점이 $D$, 인증서 유효 기간이 $T = j * L$, 인증서 유효성 갱신 기간이 $L$, 해쉬체인 상의 마지막 해쉬값이 $H^j(r)$임을 확인한다.

(2 단계) $V$는 $0 \le i < j$ 와 $H^{j-i}(H^i(r)) = H^j(r)$을 검증한다. 만약 올바르다면, $V$는 $H^i(r)$은 $H^j(r)$로 끝나는 일방향 해쉬체인 상의 올바른 해쉬값임을 믿게 된다.

(3 단계) $V$는 $D_V \le D+(j-i)*L$을 검증한다. 만약 올바르다면, $V$는 $CERT_U$가 현재 유효하고 $D_e = D+(j-i)*L$까지 유효할 것이라고 결론짓는다.

〈그림 11〉은 ZBF에서 인증서를 생성하고 검증하기 위한 절차를 보여주고 있다. 위의 방법에서, $U$는 전자서명문 생성 시에 관련된 $H^i(r)$을 배포하는 것에 의해 $CERT_U$의 유효성을 제어할 수 있다. 또한, $V$는 CA로부터 취소 정보를 획득하는 절차 없이 $CERT_U$의 취소 유무 상태를 확인할 수 있으므로, CA는 인증서의 유효성을 검증하는 절차로부터 제외된다. 또한, 인증서 소유자 $U$는 $CERT_U$의 만기일을 제어하기 위하여 해쉬체인의 루트 값인 $r$값을 유일하게 알고 있어야 하므로, $U$의 장치는 $r$과 $SK_U$를 안전하게 보관해야만 한다.

### 4.1.2 Zhou의 관리자 제어된 인증서

ZBF에서는 인증서 소유자 $U$가 인증서 유효 기간 $T$까지 $CERT_U$의 유효성을 제어한다. 이는 비밀키 훼손 등으로 인하여 발생하는 인증서 소유자에 의한 인증서 취소의 필요성에 관해서만 고려하고 있다. 하지만, 인증서는 회사 퇴직이나 직책의 변동 등 여러 가지 다른 이유로 인하여 인증서 소유를 반대하는 관리자에 의하여 취소될 필요가 있다. Zhou는 위와 같은 상황을 해결하기 위한 방안으로 사용자들을 대신하여 해쉬체인의 루트를 생성하는 보안서버(Security Server, SS)를 사용하는 ZMCC을 제안하였다.

ZMCC(J. Zhou et al.'s Manager Controlled Certificate)에서는 $CERT_U$의 유효성 갱신 지점이 다가오면, SS는 사용자 $U$에게 현재 요구되는 해쉬값을 분배한다. $U$는 해쉬체인을 계산하는 것에 의해 전송받은 해쉬값의 유효성을 쉽게 검증할 수 있다. 만약, SS가 어떠한 이유로 인하여 $U$의 인증서를 취소하기를 원한다면 $U$의 해쉬값의 배포를 중지함으로써 $CERT_U$를 취소시킨다. 따라서, $CERT_U$는 다음 유효성 갱신 지점에서 취소될 것이다.

### 4.1.3 Zhou의 공개키 프레임워크에 대한 실용적인 사용 방안

공격자에 의해서 사용자 $U$의 $SK_U$와 $r$이 노출되거나 사용자가 악의적일 경우, ZBF의 취약성 윈도우(Window of Vulnerability)는 인증서의 예정된 만기일까지 지속되는 심각한 문제점을 가지고 있다. 여기에서 취약성 윈도우(Window of Vulnerability)는 검증자가 상대방의 인증서에 대한 취소 사실을 알지 못하고 지속적으로 유효하다고 판단하는 최대 시간을 의미한다[21]. 따라서, Zhou는 SS라는 신뢰기관에게 인증서 취소에 대한 처리를 위임하는 ZMCC를 제안하였다. 하지만, SS는 단지 앞서 언급된 취약점을 극복하기 위해 추가된 부가적 신뢰기관일 뿐이며, 실질적 시스템 구현 시에 SS를 안전하게 유지하기 위한 추가적인 비용이 요구될 것이다. 또한, SS는 소규모 조직 내의 인증서 상태 검증을 위해서 적용되므로, 인터넷과 같은 글로벌 환경에서 SS를 사용하는 것은 바람직하지 못하다. 본 절에서는 Zhou의 공개키 프레임워크를 실질적 구현을 위한 고려사항을 소개한다[3].

〔고려사항 1〕 구현 환경에서 $L$ 파라메타의 적용

ZBF에서 인증서의 취소 유무 상태를 확인하기 위해서는 $D$, $L$ 그리고 $T$와 같은 다수의 시간 파라메타들이 존재한다. 특히 $L$ 파라메타가 아주 짧게 선택되어지면 사용자와 검증자가 소유한 각각의 로컬 시스템내의 시스템 클럭의 차이로 인하여 서명검증이 실패할 확률이 매우 높아진다. 따라서, 시간 동기화 문제를 제거하기 위하여 서버 지원된 서명(Server-Supported Signatures)〔5〕이나 S/Key 시스템〔13〕과 같이 해쉬체인의 반복수 감소는 시간 파라메타에 독립적이어야 한다. 즉, 사용자가 유효한 해쉬값을 생성할 필요가 있을 때마다 해쉬체인의 반복횟수가 줄어드는 방식을 사용해야 한다.

〔고려사항 2〕 추가적인 보안 파라메타 $r$의 관리

ZBF에서 사용자 $U$는 $SK_U$와 $r$ 모두 안전하게 보관해야 하는 반면, ZMCC에서는 사용자는 $SK_U$만을 안전하게 보관하며, SS는 조직 내의 멤버들에 대한 다수의 $r$들을 안전하게 보관해야 하는 부담을 가진다. 실질적 시스템에서 $U$는 $SK_U$를 보호하기 위하여 사용자-정의 패스워드를 사용한다. ZBF에서 $U$는 패스워드로써 $r$을 기억해야 하거나, 패스워드로 $r$ 값을 보호해야 한다. 따라서, $U$는 $SK_U$의 보호를 위한 패스워드와 $r$의 보호를 위한 패스워드 모두를 기억해야 한다. $SK_U$의 보호를 위한 패스워드는 $SK_U$와 직접적인 관련이 없기 때문에, $SK_U$의 보호와 $r$의 보호를 위하여 동일한 패스워드로 사용하는 것은 두 개의 패스워드를 사용하는 것보다 더욱더 사용자 친화적인 방법이 될 수 있다.

[고려사항 3] 보안서버(SS)의 제거

ZMCC에서 SS는 조직 내의 사용자들을 대신하여 $r$값을 소유함으로써, 사용자들에 대한 인증서 취소를 수행한다. SS는 글로벌 영역을 위한 PKI에서 정의되지 않는 추가적인 신뢰기관이며, 특수한 소규모 조직을 위해서만 안전하게 인증서 취소 업무를 수행할 수 있다. 특히, SS를 통한 ZMCC는 ZBF에서의 취약성 윈도우의 지속성에 대한 문제를 해결하기 위한 방안으로 간주될 수 있다. 따라서, ZMCC에서 SS를 제거함과 동시에 SS의 역할을 대신할 새로운 해결책을 제시하는 것이 바람직하다. 본 논문에서는 SS를 제거하기 위하여 사용자가 제한된 시간 구간(Time Period)에서 자신의 인증서를 제어할 수 있는 해결책을 제안한다.

Definition. 1 제어 윈도우(Control Window)란 검증자가 오직 인증서 송신자의 해쉬체인 검증을 통해서만 송신자의 인증서 취소 유무 상태를 신뢰할 수 있도록 하는 허용된 시간 구간을 의미한다.

송신자의 인증서를 수신한 검증자는 CA에게 인증서 취소 정보를 질의한다. 만약 그 인증서가 취소되지 않았으면, 검증자는 현재의 로컬 시간을 '유효성 시작 지점'으로 설정하고 유효성 시작 지점에서 제어 윈도우만큼 증가시킨 시간을 '유효성 종료 지점'으로 설정한다. 그리고, 검증자는 종료 지점까지 송신자의 인증서를 캐쉬한다. 따라서, 검증자는 '유효성 종료 지점'까지 송신자의 인증서 상태를 송신자가 전송한 해쉬값의 검증을 통해서 판단한다. 즉, 인증서 송신자는 제어 윈도우 동안은 단지 해쉬값을 계산하는 것으로 자신의 인증서 상태를 제어할 수 있음을 의미하며, 검증자는

제어 윈도우 동안 송신자의 인증서 상태에 대한 제어능력(해쉬값)을 신뢰함으로써 CA로부터 인증서 취소 정보를 더 이상 질의할 필요가 없다. 또한, 제어 윈도우를 사용함으로써 Zhou가 제안한 공개키 프레임워크에서 SS를 제거할 수 있다.

## 4.2 새로운 공개키 프레임워크 제안

앞 절에서는 Zhou의 공개키 프레임워크를 실질적 구현에 적합하게 하기 위하여 3가지 사항을 고려하였다. 본 절에서는 소개되었던 고려사항을 개선한 "인증서 생성" 및 "인증서 검증"을 위한 새로운 공개키 프레임워크를 제안한다.

〔인증서 생성과정〕

(1 단계) 사용자 $U$는 자신을 위한 키 쌍을 생성한다 : $SK_U$와 $PK_U$

(2 단계) $U$는 사용자 정의 패스워드 $r$을 생성한다. $r$은 $SK_U$를 암호화와 일방향 해쉬체인을 생성하기 위해서 사용된다.

$$H^0(r)=r,\quad H^i(r)=H(H^{i-1}(r)),\quad i=1,2,\cdots,j.$$

(3 단계) $U$는 $(PK_U, H^j(r), j)$을 CA에게 전송한다.

(4 단계) CA는 $U$의 요청을 인증한 후 아래와 같이 $U$의 인증서를 생성하여, $U$에게 전달한다.

$$CERT_U = SIGN_{CA}(U, PK_U, H^j(r), j, CW)$$

여기서, $CW$는 CA의 보안 정책에 의해 선택된 제어 윈도우(Control Window)를 나타낸다.

〔인증서 검증과정〕

(1 단계) 사용자 $U$는 검증자 $V$에게 $CERT_U$를 전송한다.

(2 단계) $V$는 $CERT_U = SIGN_{CA}(U, PK_U, H^j(r), j, CW)$ 내의 CA의 전자서명을 검증한다. 만약 검증이 성공하면, $V$는 CA에게 취소 정보(예, CRL)를 질의하여 $CERT_U$의 취소 유무를 판단한다. 만약 $CERT_U$가 취소되지 않았다면, $V$는 $U$의 공개키가 $PK_U$이며, 일방향 해쉬체인의 마지막 해쉬값이 $H^j(r)$임을 알게 된다. 그리고,

- $V$는 현재 자신의 로컬 시간을 $CERT_U$에 대한 "유효성 시작 지점"으로 설정하고, 시작 지점에서 $CERT_U$내의 $CW$만큼 증가시킨 시간을 "유효성 종료 지점"으로 설정한다.
- $V$는 "유효성 종료 지점"까지 $CERT_U$를 캐쉬한다.

(3 단계) $U$가 $V$에게 전자서명문을 전송할 때, $U$는 패스워드 $r$을 입력하여 $SK_U$를 복호화하고 현재 반복수에 해당하는 해쉬값을 계산한다. 실질적으로 $U$의 장치는 해쉬체인상의 다음 반복수를 저장하고 있어야 한다. $U$는 복호된 $SK_U$로서 계산된 전자서명문과 현재의 해쉬값 $(H^i(r), i)$를 $V$에게 전송한다.

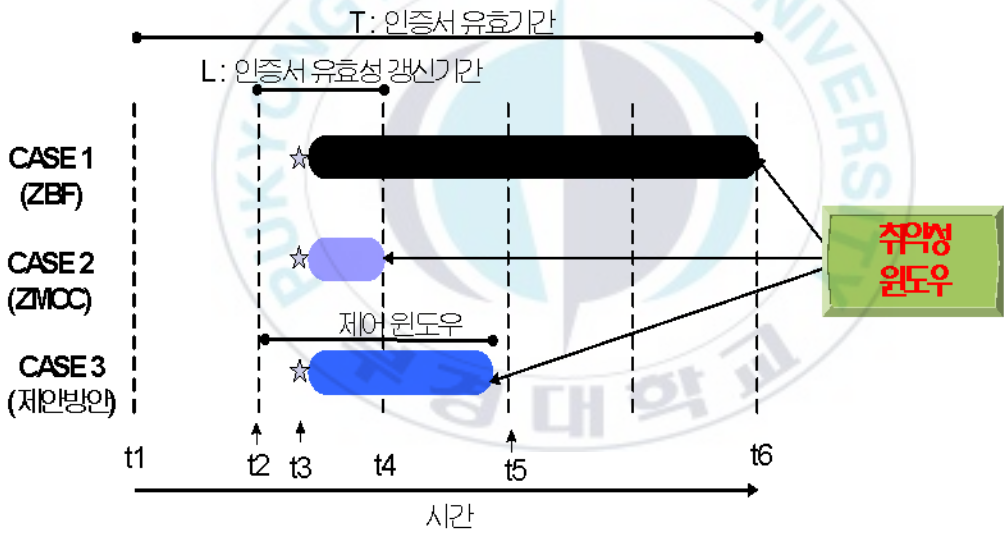
(4 단계) $V$는 $U$로부터 전자서명문을 받은 시간이 "유효성 종료 지점"을 지났는지를 확인한다. 만약 시간이 지나지 않았다면, $V$는 $H^{j-i}(H^i(r)) = H^j(r)$을 만족하는지 검사한다. 만족할 경우 $V$는 CA에게 인증서 취소 유무 정보에 대한 질의를 수행하지 않고, $CERT_U$가 현재 유효한 것으로 판단한다. 따라서, $V$는 $PK_U$로서 수신한 $U$의 전자서명문을 검증한다.

## 4.3 보안성 및 성능분석

### 4.3.1. 취약성 윈도우

본 절에서는 아래와 같이 3가지 유형에서의 취약성 윈도우(Window of Vulnerability)를 분석한다.

- CASE 1 : ZBF에서의 악의적 사용자
- CASE 2 : ZMCC에서의 정직한 사용자 또는 악의적 사용자
- CASE 3 : 제안된 공개키 프레임워크에서의 악의적 사용자




〈그림 12〉 취약성 윈도우(Windows of Vulnerabilities)

〈그림 12〉에서 사용자 $U$의 인증서 유효 기간( $T$)은 t1에서 t6까지라고

가정한다. $U$는 $CERT_U$를 t2시점에서 검증자 $V$에게 전송하며, $CERT_U$가 t3시점에서 취소된다고 가정한다. $V$는 수신한 $CERT_U$의 취소 유무 상태에 대한 검증 및 캐쉬를 다음의 3가지 유형에서 사용되는 각각의 기법에 따라서 수행할 것이다.

〔CASE 1〕 ZBF에서의 악의적 사용자

$V$는 정직한 사용자 $U$로부터 수신한 해쉬값을 검증하며, $t2 \leq t4$인가를 확인한다. 여기서, $V$는 t4를 $D+(j-i)*L$를 통해서 계산 가능하다. 만약 모든 검사가 성공적이면, $V$는 t4까지 $CERT_U$를 캐쉬한다. 즉, $CERT_U$가 t4까지 취소되지 않을 것으로 간주한다. 따라서, $V$는 t4까지는 $U$가 전송한 해쉬값을 통해서 $CERT_U$의 취소 유무에 대한 상태 검증을 수행한다. 여기서, 악의적인 사용자를 고려해 보자. 비록 $CERT_U$는 t3에서 취소되었지만 악의적 사용자 $U$는 언제나 자신 혼자서 다음 유효성 갱신 기간을 위한 해쉬값을 생성 가능하며, $V$는 단지 $U$로부터 수신한 해쉬값에 의존하여 $CERT_U$의 상태 검증을 수행한다. 결국, 취약성 윈도우는 t3에서 t6까지 된다.

〔CASE 2〕 ZMCC에서의 정직한 사용자 또는 악의적 사용자

$V$는 CASE 1과 동일한 방식으로 $CERT_U$의 상태와 시간을 검증한다. 만약 $CERT_U$가 올바르다면, $V$는 $CERT_U$를 캐쉬하고, t4까지 취소되지 않을 것이라고 간주한다. 비록 $U$가 악의적 사용자일지라도 단독으로 다

음 인증서 유효성 갱신 기간을 위한 해쉬값을 생성할 수 없다. 즉, SS가 취소된 $CERT_U$를 위한 해쉬값을 $U$에게 분배하지 않기 때문에, 취약성 윈도우는 t3에서 t4까지 된다.

〔CASE 3〕 제안된 공개키 프레임워크에서의 악의적 사용자

$V$는 4.2절에서 제안된 〔인증서 검증과정〕의 (1 단계)와 (2 단계)에 의해서, t2를 '유효성 시작 지점'으로 t5를 '유효성 종료 지점'으로 설정하고 $CERT_U$를 t5까지 캐쉬한다. 따라서, $V$는 t5까지 $CERT_U$의 취소 유무 상태를 악의적 사용자 $U$가 전송한 해쉬값의 검증 결과에 의존한다. 따라서, 취약성 윈도우는 t3에서 t5까지 된다.

〈그림 12〉와 같이, CASE 3의 취약성 윈도우 보다 CASE 2의 취약성 윈도우가 작음을 알 수 있다. 하지만, CASE 2는 SS라는 부가적인 신뢰 개체를 의존함으로써 얻어진 결과이며, SS는 인터넷과 같은 글로벌 환경에서 적용 불가능하기 때문에 본 논문에서 제안되는 공개키 프레임워크가 인터넷 환경에서 좀 더 현실적인 대안이 될 것으로 판단된다. 또한, CASE 3에서의 취약성 윈도우의 크기는 사용자의 인증서 내에 정의된 제어 윈도우의 크기에 전적으로 의존될 수 있다. 그러므로, 제어 윈도우의 크기는 충분히 주의 깊게 이뤄져야 할 것이다.

### 4.3.2. 통신비용

본 절에서는 CRL, ZBF, ZMCC 그리고 제안된 공개키 프레임워크에서의 일간의 통신비용(Daily Communication Cost)을 분석한다. CA는 발행된 인증서들에 대한 취소 정보를 분배하기 위해서 기본적으로 CRL을 사용하는 것으로 가정한다. 또한, 아래의 파라메타들을 사용한다.

- $n$ : 발행된 인증서들의 전체 추정 수 ( $n = 300,000$ )
- $p$ : 만기일 이전에 취소될 인증서들의 추정 비율 ( $p = 0.1$ )
- $q$ : 하루에 발생되는 인증서 상태 검증 질의 횟수
- $t$ : CRL이 하루에 갱신되는 횟수. $t = 2$이면, 12 시간마다 주기적인 갱신이 발생
- $L$ : ZBF 또는 ZMCC에서의 인증서 유효성 갱신 기간. $L = 1$이면, 하루를 의미
- $C$ : 제안된 공개키 프레임워크에서의 제어 윈도우의 크기. $C = 2$이면, 제어 윈도우의 크기는 2일
- $l_{sn}$ : 인증서의 시리얼 번호(Serial Number)를 위해 요구되는 비트 수 ( $l_{sn} = 20$ )
- $l_{sig}$ : 전자서명문의 비트 수 ( $l_{sig} = 1024$ )
- $l_{hash}$ : 일방향 해쉬함수를 통해 계산된 해쉬값의 비트 수 ( $l_{hash} = 160$ )

$n$, $p$, $q$, $l_{sn}$을 위한 값들은 [24]와 동일한 값들을 사용했으며, $l_{sig}$와 $l_{hash}$는 [26]과 동일하다. 한편, 본 논문에서는 $t$, $L$, $C$를 새로이 정의하

였다. CRL, ZBF, ZMCC 그리고 제안된 공개키 프레임워크에서 하루에 각각 소요되는 평균 통신비용은 다음과 같다.

〔CRL의 일간 통신비용〕

모든 인증서 상태 검증 질의에 대해서, CA는 CRL 전체를 응답으로 전송해야 한다.

$$C_{CRL} = t \cdot q \cdot (p \cdot n \cdot l_{sn} + l_{sig})$$

〔ZBF의 일간 통신비용〕

사용자들은 $L$ 마다 $q$개의 해쉬값들을 검증을 위해 전송한다.

$$C_{ZBF} = \frac{q \cdot l_{hash}}{L}$$

〔ZMCC의 일간 통신비용〕

SS들은 주기적으로 CA로부터 CRL을 다운로드해야만 하며, 사용자들에게 $q$개의 해쉬값들을 전송해 주어야 한다. 그리고, 사용자들 또한 수신받은 $q$개의 해쉬값들을 검증하기 위해 전송한다.

$$t \cdot q \cdot (p \cdot n \cdot l_{sn} + l_{sig}) + 2 \cdot \frac{q \cdot l_{hash}}{L}$$

$$= C_{CRL} + 2 \cdot C_{ZBF}$$

따라서 ZMCC의 일간 통신비용은 CRL의 통신비용에 ZBF의 두 배만큼 통신비용이 요구된다.

〔제안된 공개키 프레임워크의 일간 통신비용〕

검증자들은 인증서를 최초 검증하기 위해서 "유효성 시작 지점"에서 CA에게 CRL을 질의한다. 그리고, 사용자들은 $q$개의 해쉬값들을 검증을 위해서 전송한다.

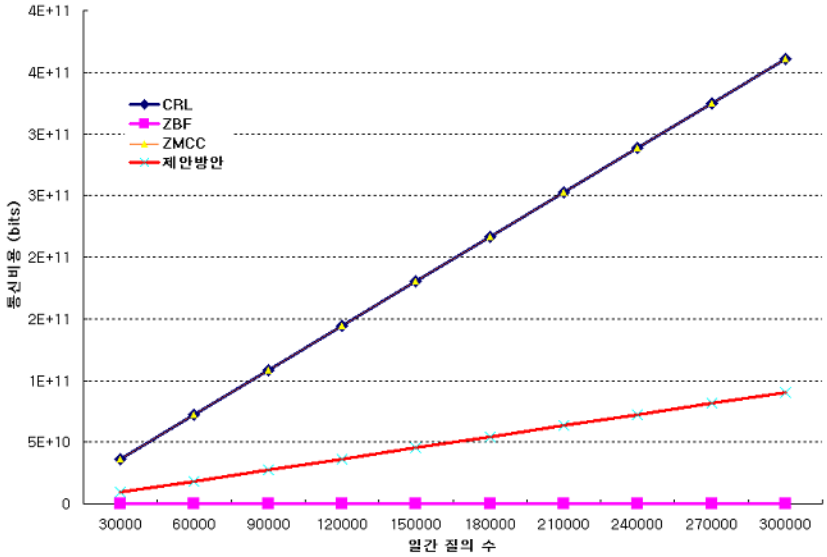
$$\frac{q \cdot p \cdot n \cdot l_{sn}}{C} + \frac{q \cdot l_{sig}}{C} + q \cdot l_{hash}$$

$$= \frac{C_{CRL}}{tC} + L \cdot C_{ZBF}$$

따라서 제안된 공개키 프레임워크의 일간 통신비용은 CRL이 하루에 갱신되는 횟수와 제어 윈도우의 크기에 반비례한다.

〈표 2〉 새로운 공개키 프레임워크에 관한
일간 통신비용($q = 300{,}000$)

| 인증서 검증 기술 | 통신비용(비트) |
|---|---|
| CRL | $3.606 \times 10^{11}$ |
| ZBF | $4.800 \times 10^{7}$ |
| ZMCC | $3.607 \times 10^{11}$ |
| 제안 방안 | $9.020 \times 10^{10}$ |




〈그림 13〉 질의 수 변화에 따른 새로운 공개키 프레임워크에 관한 일간 통신비용

〈표 2〉는 인증서 상태 검증 기법 CRL, ZBF, ZMCC 및 제안 방안에 대하여 일간 통신비용을 비교하였으며, 〈그림 13〉은 소개된 4가지 방안들에서 통신비용을 계산하기 위하여 공통적으로 사용되는 '인증서 상태 검증 질의 횟수'의 변화에 따른 전체 통신비용의 증가량을 보여주고 있다. Zhou의 주장과는 달리 [31]의 ZMCC는 CRL과 거의 유사하게 아주 높은 통신비용을 요구하며, ZBF의 경우 가장 효율적인 통신비용을 나타내고 있다. 하지만, 앞 절에서 소개된 바와 같이, ZBF의 취약성 윈도우에 대한 취약점으로 인하여, 실질적인 시스템으로의 적용은 불가능하기 때문에 효율적인 통신비용의 결과는 의미가 없다 하겠다. 반면, 제안된 공개키 프레임워크는 CRL과 ZMCC에 비하여 적은 통신비용을 요구하는 것을 알 수 있다.

## 4.4 요 약

본 장에서는 Zhou가 제안한 공개키 프레임워크의 개념과 적용된 보안 파라메타를 고찰하고, 실질적 구현에 보다 적합한 새로운 공개키 프레임워크를 제안하였다. 제안된 새로운 공개키 프레임워크는 불필요한 신뢰기관을 제거함으로써 인터넷 환경에 실질적으로 적용 가능하며, 제어 윈도우를 적용함으로써 PKI의 기존의 인증서 상태 검증 기술에 비하여 낮은 통신 및 계산 비용을 요구하며, 취약성 윈도우 관점에서의 높은 보안성을 제공하였다.

본 장에서 제안된 인터넷 환경을 위한 새로운 공개키 프레임워크는 높은 성능과 보안성에 대한 요구를 만족시키기 위한 것으로, 이는 PAN과 같은 새로운 형태의 네트워크 환경을 위하여 적용 가능할 것이다. 따라서, 다음 장에서는 PAN 환경에서 효율적이고 적시적인 보안 서비스를 제공하기 위한 새로운 인증 프로토콜을 소개하고자 한다.

# 5. PAN 환경에서의 효율적인 공개키 프레임워크

차세대 이동 통신은 네트워크에 접속하는 형태와 그 네트워크에 접속하기 위하여 사용되는 터미널에 따라서 현재의 통신과 많은 차이점을 보일 것이다. 특히, 분산 다기능 이동 터미널(Distributed Multi-function Mobile Terminals)들은 여러 개의 상이한 컴포넌트들로 이루어져 있으며, 각 컴포넌트들은 물리적으로 가까운 거리 내의 다른 컴포넌트 또는 사용자들과 로컬 무선 통신을 통하여 연결된다. Personal Area Network(PAN)은 개인의 작업 공간(전형적으로 10 미터 이내) 내의 고정, 휴대 또는 이동 컴포넌트들이 상호 연결된 네트워크이다. PAN에서는 기밀 정보나 개인 데이터가 라디오 링크(Radio Link)를 통해서 전송되기 때문에, 각 컴포넌트는 안전하고 인증된 채널을 통하여 통신해야 하며, 안전하고 인증된 통신은 PAN 컴포넌트들 사이의 적절한 보안 프로토콜과 보안 연계(Security Association)를 통하여 이루어질 수 있다.

Gehrmann은 PAN에서 키 관리를 지원하기 위하여 PAN내의 모든 컴포넌트들에게 공개키 인증서를 발급하는 Personal CA를 소개하였다[12]. Personal CA는 집이나 작은 사무실에서 일반 사용자에 의해서 관리된다. 초기화 단계는 [28]의 임프린팅(Imprinting)을 확장한 것으로, 모든 이동 장치들이 공개키 인증서를 가지도록 부트스트랩(Bootstrap)하며, 전통적인 공개키 전자서명 기법에 의하여 PAN은 정해진 동작들을 수행하게 된다.

Personal PKI 개념을 PAN 환경에 적용시키는 데는, 다음과 같은 두 가지 문제점이 존재한다.

(1) 전통적인 공개키 전자서명 기법들은 자원이 제한된 이동 장치들에게 매우 높은 작업부담을 주게 된다.

(2) 인증서 상태 정보를 관리하는데 기존의 인증서 상태 검증 기법들이 그대로 적용되고 있어, 이를 최적화하기 위한 기법이 고안되지 않았다.

따라서, 위의 두 문제를 해결하는 효율적인 인증 프로토콜 및 인증서 상태 검증 기술을 설계하는 것이 PAN 환경에서의 중요한 연구 이슈라 할 수 있다. 본 장에서는 이동 장치에서 전자서명의 생성과 검증을 위해 계산 부담을 줄일 수 있는 새로운 인증 프로토콜을 제안한다. 새로운 인증 프로토콜은 일회용 전자서명 기법(One-Time Signature)을 사용함으로써 이동 장치들이 전통적인 공개키 연산을 수행할 필요가 없으며, 또한 서명 서버(Signature Server)의 도움에 의존하는 서버 지원된 전자서명(Server- Assisted Signature)과도 차별화된다. 따라서, 제안 프로토콜은 서버 지원된 전자서명 기법에서 야기되는 분쟁 및 서버에서 요구되는 높은 계산 부담과 많은 저장 공간의 요구들이 불필요하다. 더욱이, 제안 프로토콜은 인증서 상태 정보를 검사하기 위하여 해쉬체인을 적용함으로써, 인증서 상태 검증을 위한 통신 및 계산 비용을 경감하는 단순화된 절차를 제공한다[38].

## 5.1 암호 프리미티브

본 절에서는 PAN환경에서 새로운 공개키 프레임워크 제안을 위한 기본적인 암호 프리미티브를 간략히 기술한다.

### 5.1.1 일회용 전자서명(One Time Signature, OTS)

OTS는 한 개의 메시지를 전자서명하기 위한 기법으로, 전통적인 공개키 전자서명 기법과 같이 트랩도어 함수에 기반하는 것이 아니라, 일방향 해쉬 함수에 기반 함으로써 전자서명 생성 및 검증이 매우 효율적이다. 하지만, 일회용 전자서명 기법은 다음의 두 가지 이유로 인하여, 실용적이지 못한 것으로 간주되어져 왔다.

(1) 전통적인 공개키 전자서명 기법에 비하여, 전자서명문의 길이가 상대적으로 길다.
(2) 일회용이라는 것은 매번 키 생성이 새로이 이루어져야 하는 것을 의미하여, 공개키는 인증된 방법으로 분배되어야 함을 의미한다.

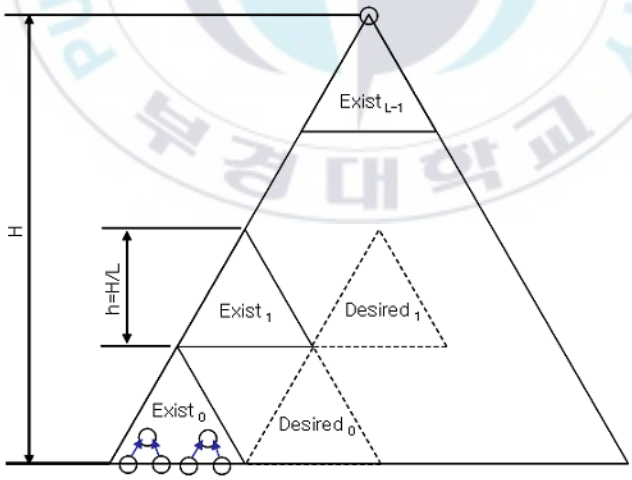
따라서 빠르고 효율적인 일방향 함수의 유용성으로 인한 이득은 명백히 손실된다. OTS의 길이를 줄이기 위하여, 메시지 다이제스트는 160 비트 출력을 가지는 SHA-1 함수를 사용하여 계산할 수 있으며, 메시지 다이제스트를 서명하기 위하여 168비트의 비밀값(Secrets)이 필요하다[23]. 최근 SHA-1의 암호학적 안전성에 대한 문제가 제기됨에 따라 보다 긴 길이의

안전한 해쉬함수를 대신하여 사용할 수 있으며, 그에 따라 보다 긴 길이의 비밀값이 필요로 할 것이다.

### 5.1.2 프렉탈 머클 트리 이동

Merkle은 많은 수의 일회용 전자서명을 인증하기 위하여 해쉬 함수를 사용하는 인증 경로(Authentication Path)라는 개념을 소개하였다[23]. 인증 경로란 주어진 잎(Leaf)과 근(Root) 사이의 경로에 있는 모든 노드들의 형제노드들 값들로써, 기본적으로 일회용 전자서명을 검증하기 위하여 서명자는 각각의 서명과 함께 관련된 인증경로를 검증자에게 제공한다.

또한, Jakobsson은 잎들(Leaves)이 차례로 사용되어질 때, 각 잎(Leaf)의 유효성 검증을 위한 인증 경로로 사용되어지는 머클 해쉬 트리(Merkle Hash Tree)의 순차 이동(Sequential Travel)을 제공하는 프렉탈 머클 트리(Fractal Merkle Tree)를 제안하였다[17].




〈그림 14〉 프렉탈 머클 트리의 구조

〈그림 14〉는 프렉탈 머클 트리의 전반적인 구조를 나타내고 있다. 높이 $H$를 가지는 해쉬 트리 $T$는 각각 높이 $h$를 가지는 $L$ 레벨들로 나누어지며, 해쉬 트리의 잎들은 왼쪽으로부터 오른쪽으로 $\{0,1,....,2^H-1\}$과 같이 색인된다. 노드 $n$의 높이는 최대의 서브트리의 높이로써 정의되며, 서브트리의 루트의 높이가 $ih$ (여기서, $i \in \{1,2,...,L\}$)일 때 $h$-서브트리는 '레벨 $i$'에 있다고 정의한다. 각각의 $i$에 대하여, 레벨 $i$상에 $2^{H-ih}$개의 $h$-서브트리가 존재하고, 만약 모든 $i < L$에 대하여 $Tree_i$의 루트가 $Tree_{i+1}$의 잎이 될 때, 일련의 연속적인 $h$-서브트리들 $\{\mathrm{Tree}_i\}_{i=1}^{L}$들은 스택화된다.

프렉탈 머클 트리의 목적은 최소한의 저장공간과 계산량으로 현재의 잎(전자서명)의 인증경로를 제공하는 것으로 서명자는 전체의 해쉬 트리를 저장하는 것 대신에 단지 다음과 같은 Exist 서브트리들과 Desired 서브트리들의 집합들만 저장하면 된다.

- Exist 서브트리란 $\{\mathrm{Exist}_i\}_{i=0}^{L-1}$로 색인되는 일련의 연속적으로 스택화된 $h$-서브트리들로써, 현재 서명값에 대한 인증경로를 포함한다.
- Desired 서브트리란 $\{\mathrm{Desired}_i\}_{i=0}^{L-2}$로 색인되는 일련의 연속적으로 스택화된 $h$-서브트리들로써, 각 Desired 서브트리는 같은 레벨의 Exist 서브트리에 대한 이웃 트리이다.

현재 사용중인 Exist 서브트리가 더 이상 다음 인증경로를 포함하고 있지 않으면, 현재 Exist 서브트리는 자신의 이웃인 Desired 서브트리로 대체된다. 프렉탈 머클 트리의 키 생성 단계에서, 서명자는 모든 Exist 서브트리와 Desired 서브트리를 초기화하고 전체 해쉬 트리의 루트값을 계산한다. 이후, 서명자는 Exist 서브트리들을 제외한 모든 계산된 값들을 폐

기하나, 계산된 해쉬 트리의 루트값은 검증자에 의해 인증된 방식으로 처리되어질 수 있다. 프렉탈 머클 트리에서 주목할 점은 모든 해쉬 트리의 노드들이 키 생성 단계에서 계산되어지는 것이다.

결론적으로, 프렉탈 머클 트리에서 요구되는 전체 저장 공간은 $1.5\log^2 N/\log\log N$ 해쉬 값들로 제한되며, 최악의 경우의 계산상 비용은 각 출력마다 $2\log N/\log\log N$ 해쉬 연산이 요구된다. 최근, Naor은 Merkle의 일회용 전자서명과 Jakobsson의 알고리즘을 결합한 기법이 적은 전자서명 길이와 저장 공간으로 효율적인 전자서명을 생성할 수 있음을 실험을 통하여 보였다[25].

## 5.2 제안 시스템 모델

### 5.2.1 설계 원칙

본 절에서는 PAN 환경에서 이동 장치들 사이에 효율적인 인증과 인증서 상태 검증을 제공하기 위하여 제안 방안의 설계 원칙을 제시한다. 제안 방안의 설계를 위한 주요 원칙은 아래와 같다.

- 이동 장치에서 공개키 연산들의 제거 : 전통적인 공개키 전자서명 기법들은 전자서명 생성 및 검증을 위하여 계산적으로 복잡한 연산들을 요구하기 때문에, PAN의 자원 제한적인 이동 장치들(8 비트나 16 비트의 매우 저속의 CPU 속도에서 동작하는 마이크로컨트롤러들을 소유하는 장치들)을 위해서는 적합하지 않다. 그러므로, 어떠한 공개키 연산도 수행할 필요가 없는 인증 프로토콜의 설계는 PAN 환경에서 매우 중요한 과제이다.

■ 서명 서버의 도움 불필요 : 공개키 전자서명문 생성과 같이 계산량이 많은 연산을 수행하지 않기 위하여, 서명 서버(Signature Server)를 이용하여 연산을 수행하는 암호학적 프로토콜들이 제안되었다[7][10]. 하지만, 이와 같은 프로토콜들은 분쟁 해결을 위하여 서버 또는 서버와 이동 장치들에게 높은 저장 공간을 필요로 한다. 더욱이, 검증자들이 공개키 암호 연산을 해야 하기 때문에, 모든 전자서명이 서명 서버로부터 수행됨으로써 발생되는 라운드 트립으로 인한 지연(Delay)이 불가피하다. 따라서, 서명 서버의 도움이 불필요한 인증 프로토콜을 설계하는 것이 바람직하다.

■ 인증서 상태 유효성 검사를 위한 적은 계산 및 통신비용 : OCSP[20]와 같은 온라인 인증서 상태 검증 메커니즘은 CRL[14]과 비교했을 때, 자원의 소모가 적어서 이동 장치에게 적절하지만, Personal CA는 높은 통신비용 및 많은 수의 전자서명문 생성으로 인하여 높은 계산 비용을 부담해야만 한다. 따라서, Personal CA의 작업 부담을 완화시키기 위하여, Personal CA의 전자서명문을 생성하는 횟수와 전체 통신 횟수를 줄일 필요가 있다.

### 5.2.2 구 조

본 절에서 제안되는 모델의 구조를 좀 더 명확히 정의하기 위하여, 다음을 가정한다.

〔가정〕

1) PAN은 무선 인터페이스를 통하여 서로 통신이 가능한 휴대 가능한 컴포넌트들로 구성된다.
2) PAN이 구성될 시점에, PAN의 루틴 연산들을 안전하게 만들기 위해서 필요한 모든 보안 연계(Security Association)가 설정된다. 즉, PAN내의 모든 이동 장치들은 초기 단계 동안 보안 정보(Security Quantities)와 인증서들을 가지고 부트스트랩된다.

본 절에서의 PAN은 아래와 같이 Personal CA와 이동 장치들로 구성된다.

- Personal CA : Personal CA는 PAN에서 유일하게 신뢰되는 제3자이며, 명령어를 입력하기 위한 간단한 입력 장치와 디스플레이 장치를 소유하고 있다. 또한, Personal CA는 다른 모든 PAN 컴포넌트들에게 인증서들과 인증서 상태 정보를 제공하기 위하여, 영구적으로 온라인 상태로 이용가능하다.
- 이동 장치 : PAN내에 존재하는 컴포넌트들로 네트워크킹이 가능하며, 비교적 낮은 컴퓨팅 능력을 소유하고 있다.

본 절에서 제안되는 모델에서, 모든 이동 장치들은 초기화 단계 동안에 자신의 공개키/개인키 쌍과 대응되는 인증서를 소유한 상태로 부트스트랩된다. 모든 이동 장치들이 보안 정보들을 가지고 초기 등록(Imprint)된 후, 이동 장치들은 메시지를 서명하고 검증하기 위하여 본 절에서 제안되는 인증 프로토콜을 수행한다.

### 5.2.3 용 어

본 절에서 제안되는 프로토콜의 기술을 위하여, 아래의 용어들을 사용한다.

〔용어〕

- $PCA, M$ : Personal CA 및 이동 장치의 신원정보
- $h()$ : 암호학적 일방향 해쉬 함수
- $SK_X$ : 이동 장치 $X$가 랜덤하게 선택한 비밀키
- $sk_X^{i,j}$ : 이동 장치 $X$의 OTS의 비밀값(Secrets)으로 $h(SK_X|i|j)$이다. 여기서, $i$는 전자서명문 번호, $j$는 비밀값의 색인 그리고 $\|$ 는 메시지의 연접이다.
- $pk_X^{i,j}$ : 각 $sk_X^{i,j}$를 위한 커미트먼트(Commitment)로서 $h(sk_X^{i,j})$이다.
- $PLC_X^i$ : $i$ 번째 공개 잎 커미트먼트(Public Leaf Commitment)로서, 단일 OTS의 모든 커미트먼트들의 해쉬값으로 $h(pk_X^{i,1}\| \cdots \|pk_X^{i,t})$이다.
- $PK_X$ : 이동 장치 $X$의 공개키로서, 프렉탈 머클 해쉬 트리의 트리 루트이다.
- $AuthPath_X^i$ : 이동 장치 $X$의 $i$ 번째 공개 잎 커미트먼트의 인증 경로
- $VK_X^{n-i}$ : 이동 장치 $X$의 $i$ 번째 검증 키(Validation Key)로서, $h()$의 범위 내에서 랜덤하게 선택된 보안 정보 $VK_X$를 기반하여, 이동 장치 $X$는 해쉬체인 $VK_X^0, VK_X^1, \cdots, VK_X^n$을 계산한다.

  여기서, $VK_X^0 = VK_X$, $VK_X^i = h_X^i(VK_X) = h_X(VK_X^{i-1})$이다. $VK_X^n$를 $X$의 루트 검증 키(Root Validation Key)라고 하고, $VK_X^{n-i}$를 $X$의 현재 검증 키(Current Validation Key)라고 한다.

- $Sig_X^i$ : 이동 장치 $X$의 $i$ 번째 OTS
- $Cert_X$ : 이동 장치 $X$의 인증서

## 5.3 시스템 동작

### 5.3.1 초기화 프로토콜

본 논문의 초기화 프로토콜(Initialization Protocol)은 [12]에서 소개된 수동 인증 프로토콜(Manual Authentication Protocol)을 변경하였다. 제안되는 초기화 프로토콜의 구체적인 단계는 아래와 같다.

(1 단계) Personal CA는 자신의 식별자 $PCA$와 공개키 $PK_{PCA}$를 이동 장치에게 전송한다.

(2 단계) 이동 장치는 랜덤하게 두 개의 보안 정보인 $SK_M$과 $VK_M$을 생성하고, 다음을 수행한다.

1) 전자서명문들의 전체 수 $n$ 만큼의 일회용 비밀값/커미트먼트 쌍들 및 대응되는 공개 잎 커미트먼트들을 생성한다(PAN 환경을 고려하면, 전자서명문의 전체 수가 $2^{16}$보다 작을 것으로 가정한다).

2) 잎으로 공개 잎 커미트먼트 값들, $PLC_M^i$을 가지는 높이가 $\log n$인 프렉탈 머클 해쉬 트리를 초기화하고, 공개키 $PK_M$을 계산한다. 여기서 $i = 1, \cdots, n$.

3) 루트 검증키로서 $VK_M^n = h^n(VK_M)$을 계산한다.

4) 전자서명 번호 $i$를 0으로 설정한다.

그리고, 이동 장치는 $M, PK_M, n, VK_M^n$을 Personal CA에게 전송한다.

(3 단계) Personal CA와 이동 장치는 다음의 수동 인증절차를 수행한다.

1) Personal CA는 랜덤 키 $k$를 생성하고, $k$를 키로 하여 $PCA, PK_{PCA}, M, PK_M, n, VK_M^n$의 함수로써 MAC을 계산한다. 그리고, MAC과 키 $k$가 Personal CA에 의해 디스플레이된다.

2) 사용자는 MAC 값과 $k$를 이동 장치에 입력한다. 이동 장치는 입력된 $k$를 MAC 값을 재계산하기 위해서 사용한다. 만약 두 값이 동일하면, 이동 장치는 사용자에게 성공 신호를 보낼 것이다. 두 값이 일치하지 않으면, 실패 신호를 보낼 것이다.

(4 단계) 만약 이동 장치가 성공 신호를 보내면, 사용자는 Personal CA가 인증서를 생성하도록 명령한다. Personal CA는 인증서를 생성하기 위하여, 시스템 보안 정책에 따라서 제어 윈도우 $CW$를 설정하고, 이동 장치를 위한 OTS로 전자서명된 인증서 및 그 인증서의 인증 경로 $AuthPath_{PCA}^i$를 발행한다.

$Cert_M = \{Ser\#, M, PK_M, n, VK_M^n, CW, Sig_{PCA}^i\}$ 여기서, $Ser\#$은 시리얼 넘버이다.

(5 단계) 이동 장치는 발행된 인증서의 올바름을 검증하기 위하여 다음을 검사한다.

1) $PK_{PCA}$와 $AuthPath_{PCA}^i$를 사용하여, 인증서를 위한 Personal CA의 OTS를 검증한다.

2) 인증서 내의 데이터 필드들이 유효한지 아닌지를 검사한다. 만약 모든 검사들이 성공적이면, 프로토콜이 완료된다.

### 5.3.2 인증 및 인증서 상태관리 프로토콜

본 절에서는, 서명 서버의 도움 없이 빠르게 전자서명문을 생성 및 검증 가능한 효율적인 인증 프로토콜을 제안하며, 제어 윈도우 메커니즘을 이용한 단순화된 인증서 상태 검증 기법을 제안한다. 제안 방안의 상세한 내용은 아래와 같다.

〔전자서명문 생성〕

어떤 메시지 $m$을 전자서명 하고자 하는 이동 장치 $M_s$는 다음을 수행한다.

(1 단계) Merkle의 OTS 기법[23]을 다음과 같이 수행한다.

(1) 전자서명 번호 $i$를 증가시킨다.

(2) $m$을 위한 메시지 다이제스트 $md = h(m)$를 계산하고, $md$내의 '0' 비트들의 수를 $C$로 설정하고, $msg = md \| C$로 설정한다.

(3) $\{sk_{M_s}^{i,j}\}_{j=1}^{t}$와 대응되는 $\{pk_{M_s}^{i,j}\}_{j=1}^{t}$를 생성한다. 여기서 $t = |msg|$이다.

(4) $Sig_{M_s}^{i} = \{sk_{M_s}^{i,j} \forall j \in \{j | msg_j = 1\}, pk_{M_s}^{i,j} \forall j \in \{j | msg_j = 0\}\}$를 계산한다.

(2 단계) $AuthPath_{M_s}^{i}$를 계산하고, 프렉탈 머클 트리 알고리즘[17]을 이용하여, 인증 경로를 갱신한다.

(3 단계) 현재 검증 키 $VK_{M_s}^{n-i}$를 계산한다.

그리고, 이동 장치는 $m$, $Sig_{M_s}^{i}$, $AuthPath_{M_s}^{i}$, 전자서명 카운터 $i$ 및 현재 검증 키 $VK_{M_s}^{n-i}$를 대상 이동 장치 $M_v$에게 전송한다.

〔전자서명문 검증〕

이동 장치 $M_v$는 이동 장치 $M_s$의 상태를 검사하기 위하여, 다음을 수행한다.

(1 단계) $Cert_{M_s}$를 얻고, $Cert_{M_s}$의 상태가 유효한지 아닌지를 질의한다.

(2 단계) $Cert_{M_s}$가 유효한 경우, 인증서 내의 루트 검증키를 바탕으로 현재 검증키를 검사한다.

$$h^{i}(VK_{M_s}^{n-i})? = VK_{M_s}^{n}$$

(3 단계) 만약 모든 검사가 성공적이면, 이동 장치 $M_v$는 $Cert_{M_s}$를 캐쉬하고, 현재 로컬 시간을 신뢰 시간의 시작점으로 설정한 후, $Cert_{M_s}$내의 제어 윈도우를 바탕으로 신뢰 시간의 종료점을 설정한다.

수신된 전자서명문을 검증하기 위하여, 이동 장치 $M_v$는 다음을 수행한다.

(1 단계) 메시지 다이제스트 $md' = h(m)$를 계산하고, $C'$를 $md'$에서의 0 비트 수로 설정한 뒤, $msg' = md' \| C'$로 설정한다.

(2 단계) $Sig_{M_s}' = Sig_{M_s}$로 설정한다. 여기서, $t = |msg'|$일 때, $Sig_{M_s}' = \{sig_j'\}_{j=1}^{t}$이다. 그리고, $\forall j \in \{j | msg_j' = 1\}$에 대하여 $sig_j' \leftarrow h(sig_j')$를 갱신하고, $PLC_{M_s}^{i} = \{sig_1' \| \cdots \| sig_t'\}$를 계산한다.

(3 단계) $AuthPath_{M_s}^{i}$를 가지고 $PLC_{M_s}^{i}$를 반복적으로 해쉬하고, $Cert_{M_s}$내의 $PK_{M_s}$과 그 결과를 비교한다.

기존에 제안되었던 서버 지원된 전자서명[7][10]과 비교하여, 제안 프로토콜은 공개키 연산을 수행하지 않아 자원 제한적인 이동 장치들의 계산 비용을 줄이며, 어떠한 서명 서버도 필요로 하지 않는다. 또한, 검증자는 신뢰 시점의 종단점까지 서명자의 인증서를 해쉬체인에 기반하여 신뢰함으로써, 검증자는 서명자의 인증서 상태 정보를 Personal CA에게 질의할 필요가 없다. 따라서, 제안 프로토콜은 OCSP[19]나 CRL[14] 등과 같은 인증서 상태를 검증하기 위한 기술보다 보다 낮은 통신 및 계산 비용을 요구하게 된다.

## 5.4 보안성 및 성능 평가

### 5.4.1 보안성 평가

안전한 연산을 제공하기 위하여, 제안 프로토콜에서 사용되는 OTS와 제어 윈도우 메커니즘의 보안성이 증명되어야 한다. 비밀값 생성 및 Merkle의 OTS내의 해쉬 연산을 위하여 사용되는 일방향 해쉬 함수 $h()$가 충돌 회피성을 가진다는 것은 어떤 메시지 $m' \neq m$를 위한 전자서명문

을 위조 불가능하다는 것을 의미한다. 또한, 제어 윈도우 메커니즘에서 이동 장치의 $i$ 번째 OTS에 대응되는 현재 검증키를 위조하기 위하여, 공격자가 이동 장치의 인증서 내에 있는 루트 검증 키 $VK^n$의 $(n-i)$ 번째 $h()$의 역원(Inverse)을 계산하는 것은 불가능하다.

### 5.4.2 성능 평가

〈표 3〉은 서버 지원된 전자서명 [7]과 제안 프로토콜을 계산 및 저장 공간 요구사항의 관점에서 비교한 결과이다. 〈표 3〉에 사용되는 용어는 아래와 같다.

[용어]

- $H$ : 해쉬 연산
- $S$ : 전통적인 전자서명 생성
- $V$ : 전통적인 전자서명 검증
- $p$ : OTS 검증을 위한 해쉬 연산의 수
- $m$ : OTS내의 공개 커미트먼트의 수
- $K$ : 보안 정보의 크기
- $C$ : 전자서명 카운터의 크기
- $A$ : 인증 경로를 위한 해쉬 트리의 크기
- $T_c$ : 인증 경로의 계산
- $T_v$ : 인증 경로의 검증

〈표 3〉 계산 및 저장 공간 요구사항의 비교

| 복잡도 \ 방안 | | 서버 지원된 전자서명 [7] | 제안 방안 |
|---|---|---|---|
| 전자서명 계산 | 서명자 | $(m+1)H$ | $(m+1)H+1T_c$ |
| | 서버 | $(p+2)H+1S$ | - |
| | 검증자 | $1H+1V$ | $(p+1)H+1T_v$ |
| 저장공간 | 서명자 | $mH$ | $2K+1C+1A$ |
| | 서버 | $(m+p+1)H$ | - |
| | 검증자 | $1Cert$ | $1Cert$ |

제안 프로토콜에서 서명자의 계산 비용은 서버의 높은 부담 없이 서버 지원된 전자서명 기법과 유사하다. 더욱이, 제안 프로토콜에서의 전자서명 검증은 전통적인 전자서명 검증을 수행할 필요가 없으므로, 서버 지원된 전자서명 기법보다 효율적이다. 서버 지원된 전자서명에서 분쟁을 해결하기 위하여, 모든 전자서명문을 저장해야 하는 문제점을 가지고 있었으나, 제안 방안은 서명 서버를 제거함으로써 서버에서의 높은 저장 공간을 필요하지 않다. 한편, 서명자에서의 저장 공간 요구사항을 고려하면, 제안 프로토콜은 대략 1.9 KB만을 필요로 한다(두 개의 20 바이트 보안 정보, 1920 바이트의 해쉬 트리 및 4 바이트 전자서명 카운터). 반면에, 서버 지원된 전자서명 기법은 3.3 KB를 필요로 한다(168×20 바이트 = 대략 3.3 KB).

제어 윈도우 메커니즘의 효율성을 고려하면, 검증자가 제어 윈도우 기간 동안 Personal CA에게 인증서 상태 정보를 질의하지 않기 때문에, 제안 프로토콜이 전자서명문 생성과 Personal CA와의 통신 패스의 수를 명백

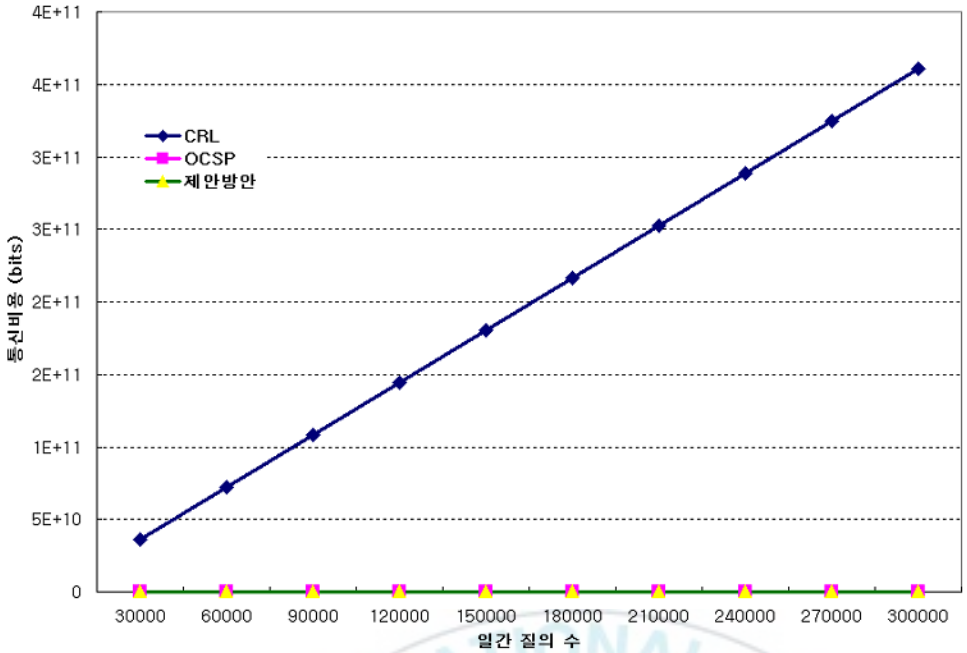
히 감소시키는 것을 알 수 있다. 통신비용에 대한 구체적이고 일반적인 측도를 위하여, 4.3.2절에서 소개된 파라메타들을 사용하여 OCSP 및 CRL과의 통신비용을 비교한다.

〈표 4〉 PAN에서 제안된 공개키 프레임워크에 관한 일간 통신비용 ($q = 300{,}000$)

| 인증서 검증 기술 | 통신비용 (비트) |
|---|---|
| CRLs | $3.606 \times 10^{11}$ |
| OCSP | $3.132 \times 10^{8}$ |
| 제안 방안 | $2.046 \times 10^{8}$ |

〈표 4〉는 추정되는 일간 통신비용을 CRL, OCSP와 제안 방안의 세 가지 인증서 상태 검증 기법에 관하여 아래의 측정 기준으로 비교하였으며, 〈그림 15〉는 일간 질의 수 변화에 따른 PAN환경에서의 제안된 공개키 프레임워크에 관한 일간 통신비용의 변화를 보여주고 있다.

- CRL의 일간 통신비용 : $t \cdot q \cdot (p \cdot n \cdot l_{sn} + l_{sig})$
- OCSP의 일간 통신비용 : $q \cdot l_{sn} + q \cdot l_{sig}$
- 제안 방안의 일간 통신비용 : $\frac{q \cdot l_{sn} + q \cdot l_{sig}}{C} + q \cdot l_{hash}$




〈그림 15〉 질의 수 변화에 따른 PAN에서 제안된 공개키 프레임워크에 관한 일간 통신비용

만약, OCSP를 대신하여 제안되는 제어 윈도우 메커니즘을 사용하면, 인증서 상태 검증을 위한 통신비용은 대략 65%를 줄일 수 있게 된다.

## 5.5 요 약

본 장에서는 PAN 내의 이동 장치에서 전자서명의 생성 및 검증을 요구되는 비용을 줄이기 위한 새로운 인증 프로토콜을 제안하였다. 특히, 일회용 전자서명 기법을 도입함으로써 이동 장치들이 전통적인 공개키 연산을 수행할 필요가 없으며, 서명 서버의 도움에 의존하는 서버 지원된 전자서명에서의 서버의 전자서명 연산을 위한 부담 또한 제거하였다. 더욱이, 제

안 프로토콜은 인증을 수행하기 위하여 요구되는 인증서 상태 정보를 검사하기 위하여 해쉬체인과 4장에서 제안된 새로운 공개키 프레임워크의 제어 윈도우를 적용함으로써, 인증서 상태 검증을 위한 통신 및 계산 비용을 경감한 단순화된 절차를 제공하였다. 따라서, 본 장에서 제안된 새로운 인증 프로토콜은 Personal PKI의 기반구조 하에서 요구되는 다양한 형태의 보안 서비스를 효율적으로 제공하기 위한 기틀을 제공할 수 있을 것으로 판단된다.

# 6. 결론

공개키 기반구조는 인증 및 신뢰성 있는 통신과 같은 보안 서비스를 제공하기 위하여 다양한 형태의 네트워크 접속환경에서 적용되고 있으며, 기제안된 인증서 상태 검증 기술은 보안 서비스를 수행할 시점에서 인증서 취소 유무를 검증하기 위하여 높은 통신 및 계산 비용을 요구하고 있다.

본 논문에서는 인터넷 환경에서 인증서 상태 검증 시 통신 및 계산 비용을 줄이기 위한 새로운 공개키 프레임워크를 제안하였다. 또한, 이기종 네트워크 장치들이 근접한 지역에서 컴퓨팅 환경을 구성하는 PAN에서의 보안 서비스를 제공하기 위하여, 제안된 새로운 공개키 프레임워크를 적용하기 위한 연구를 수행하였다.

Zhou의 공개키 프레임워크에서 인증서 상태 검증 기술을 고찰하고, Zhou의 공개키 프레임워크 내의 보안 파라메타들을 재조명하고, 실제 구현 환경에 적합하도록 개선하였다. 또한, Zhou의 공개키 프레임워크에서 불필요한 신뢰기관을 제거함으로써 인터넷 환경에 적용 가능한 새로운 공개키 프레임워크를 제안하였다. 제안된 공개키 프레임워크는 PKI의 기존 인증서 상태 검증 기술에 비하여 통신 및 계산 비용을 줄였으며, 합리적인 취약성 윈도우를 제공하였다.

또한, PAN환경 내의 장치들이 전자서명의 생성과 검증을 위해 계산 비용을 줄이기 위한 새로운 공개키 프레임워크와 인증 프로토콜을 제안하였다. PAN의 장치들은 서명 서버의 도움 없이 일회용 전자서명(One-Time Signature)을 통하여 전통적인 공개키 연산을 수행할 필요가 없다. 더욱이, 제안된 새로운 공개키 프레임워크는 인증서 상태를 검사하기

위하여 해쉬체인을 적용함으로써, 인증서 상태 검증을 위한 통신 및 계산 비용을 경감하는 단순화된 절차를 제공하였다.

본 논문의 연구는 기존의 인터넷 환경뿐만 아니라 가까운 미래에 실현될 PAN과 같은 새로운 네트워크 접속 환경에서의 보안 서비스를 위한 새로운 공개키 프레임워크를 제안함으로써 요구되는 다양한 형태의 보안 서비스를 효율적으로 제공하기 위한 기틀을 제공하며, 제안된 새로운 공개키 프레임워크를 통하여 효율적으로 인증 가능한 네트워크 접속 방식을 제공할 수 있을 것으로 기대된다.

# 참고문헌


[1] C. Adams, P. Sylvester, M. Zolotarev and R. Zuccherato, "Internet X.509 Public Key Infrastructure Data Validation and Certification Server Protocols", IETF RFC 3029, February, 2001.

[2] C. Adams and S. Lloyd, "Understanding public-key infrastructure: concepts, standard, and deployment considerations," Indianapolis: Macmillan Technical Publishing, 1999.

[3] J. Yang, C. Sur, H. Jang, Kyung-Hyune Rhee, "Practical Modification of an Efficient Public-key Framework", 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service, IEEE, 2004.

[4] M. Ambarish, H. Russ and F. Trevor, "Simple Certificate Validation Protocol(SCVP)", IETF draft-ietf-pkix-scvp-07.txt, February, 2002.

[5] N. Asokan, G.Tsudik and M.Waidner, "Server-Supported Signatures", European Symposium on Research in Computer Security, pp.131-143, 1996.

[6] N. Andrew, D. William, J. Celia and B. Derek, PKI : Implementing and Managing E-Security, McGraw-Hill.

[7] K. Bicakci and N. Baykal, "Server assisted signature revisited", Topics in Cryptology - CT-RSA 2003, pp.143-156 March 2003.

[8] Peter Buhler, Thomas Eirich, Michael Stenier and Michael Waidner, "Secure Password-Based Cipher Suite For TLS", In Symposium on Network and Distributed Systems Security (NDSS '00), pp.129-142, 2000.

[9] Steven M. Bellovin, "Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks", Proceedings of the IEEE Symposium on research in Security and Privacy, Oakland, pp.72-84, 1992.

[10] X. Ding, D. Mazzocchi and G. Tsudik, "Experimenting with Server-Aided Signatures", 2002 Network and Distributed Systems Security Symposium (NDSS'02), February 2002.

[11] Y. Elley, A. Anderson, S. Hanna, S. Mullan, R. Perman and S. Proctor, "Building Certification Paths : Forward vs. Reverse", Network and Distributed System Security Symposium Conference Proceedings, 2001.

[12] C. Gehrmann, K. Nyberg and C. Mitchell, "The personal CA - PKI for a Personal Area Network", Proceedings - IST Mobile & Wireless Communications Summit 2002, June 2002.

[13] N. Haller, "The S/Key One-time Password System", Proceeding of ISOC Symposium on Network and Distributed System Security, pp.151-157, 1994.

[14] R. Housley, W. Ford, W. Polk and D. Solo, "Internet X.509 public key infrastructure certificate and CRL profile", RFC 2459, January 1999.

[15] R. Hously and T. Polk, Planning for PKI, John Wily &Sons, 2001.

[16] S. Halevi and H. Krawczyk, "Public-Key Cryptography And Password Protocols", In 5th ACM Conference on Computer and Communication Security, San Francisco, California, pp.122-131, 1998.

[17] M. Jakobsson, F. Leighton, S. Micali and M. Szydlo, "Fractal Merkel tree representation and traversal", Topics in Cryptology - CT-RSA 2003, pp.314-326, 2003.

[18] C. J. Mitchell, Security for Mobility, IEEPress, 2004.

[19] M. Myers, R. Ankney, A. Malpani, S. Galperin and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol OCSP", IETF RFC 2560, June, 1999.

[20] M. Myers, R. Ankney, A. Malpani, S. Galperin and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol OCSP", IETF draft-ietf-pkix-rfc2560bis-01.txt, February, 2002.

[21] P. McDaniel and S. Jamin, "Windowed certificate revocation", Proceedings of IEEE INFOCOM'2000, Tel-Aviv, Israel, pp.1406-1414, 2000.

[22] Richard A. Mollion, An Introduction to Cryptography, Second Edition, Chapman&Hall/CRC Press, 2006

[23] R. C. Merkle, "A digital signatures based on a conventional encryption function", Advances in Cryptology - CRYPTO'87, pp.369-378, 1987

[24] S. Micali, "Efficient Certificate revocation", Technical Memo MIT/LCS/TM-542b, 1996.

[25] D. Naor, A. Shenhav and A. Wool, "One-Time Signature Revisited: Have They Become Practical?", Cryptology ePrint Archive, Report 2005/442, 2005.

[26] M. Naor and K. Nissim, "Certificate revocation and certificate update", Proceedings 7th USENIX Security Symposium, San Antonio, Texas, pp.217-228, 1998.

[27] V. Prasad, S. Potakamuri, M. Ahern, M. Lerner, L. Balabine and P. Dutta "Scalable Policy Driven and General Purpose Public Key Infrastructure(PKI)", IEEE 2002.

[28] F. Stajano and R. Anderson, "The resurrecting duckling: security issues for ad-hoc wireless networks", Proceeding of the 7th International Workshop on Security Protocols, LNCS 1796, 1999.

[29] C. Yae Liau, S. Bressan and T. Kian-Lee "Efficient certificate Revocation : A P2P Approach", ASIAN 2002 Workshop on southeast Asian Computing Research, 2002.

[30] J. Yang, W. Shin and K. Rhee, "An end-to-end authentication protocol in Wireless Application Protocol", ACISP 2001, LNCS 2119, pp.247-259, 2001.

[31] J. Zhou, F. Bao and R. Deng, "An Efficient Public-Key Framework," 5th International Conference on Information and Communications Security, LNCS 2836, pp.88-99, 2003.

[32] 곽진, 이승우, 조석향, 홍순좌, 원동호, "시간 정보를 이용한 인증서 상태 검증 정보 제공에 관한 연구", 한국정보처리학회 춘계학술발표 논문집, 제9권 1호, pp.833-836, 2002.

[33] 고훈, 김대원, 장의진, 신용태, "보안성을 고려한 분산된 OCSP 서버 구축 제안", 한국정보과학회 추계학술발표논문집, 제30권 2호, p.793, 2003.

[34] 고훈, 장의진, 신용태, "분산 OCSP 서버로의 안전한 정보 전달 설계", 한국정보과학회 추계학술발표논문집, 제30권 2호, p.640, 2003.

[35] 김현철, 이광형, 백주호, 오해석, "축약 서명 기반의 효율적인 인증서 상태 검증 시스템에 관한 연구", 한국정보과학회 추계학술논문집, 제30권 2호, pp.829-831, 2003.

[36] 노종혁, 김태성, 원형석, 진승헌, "인증서 검증 서버의 인증 경로 생성", 한국정보과학회 추계학술논문집, 제29권 2호, pp.604-606, 2002.

〔37〕 이승우, 곽진, 조석향, 주미리, 원동호, "실시간 인증서 검증 시스템 모델에 관한 연구", 한국정보처리학회 춘계학술발표논문집, 제9권 1호, pp.833-836, 2002.

〔38〕 장화식, 이경현, "PAN에서 인증 및 인증서 상태 관리를 위한 효율적인 프로토콜", 한국멀티미디어학회, 제10권 3호, pp.373-380, 2007.

〔39〕 최연희, 박미옥, 추연수, 전문석, "비계층적 PKI에서의 인증 경로 처리 기법에 대한 새로운 방안", 한국정보과학회 추계학술논문집, 제30권 2호, pp.601-603, 2003.