효율적인 위임서명을 위한 검증자 중심의 위임관리 프로토콜

정회원 박 세 준*, 이 용 준*, 오 동 열*, 오 해 석**

Proxy Management Protocol for based on Verifier for Efficient Proxy Signature

Se-Joon Park*, Yong-Joon Lee*, Dong-Yeol Oh*, Hae-Seok Oh** ***Regular Members***

요 약

실생활에서 권한의 위임을 통한 위임서명 기법들이 최근 많이 연구되고 있다. 그러나 현재 사용되고 있는 위임서명 기법은 많은 보안상의 문제점을 가지고 있다. 가장 큰 문제점은 인증서와 개인키의 오남용을 막기가 힘들다는 것이다. 본 논문에서는 기존의 원서명자와 위임서명자와의 관점을 원서명자와 검증자와의 관점으로 전환하여 구조적으로 보안이 강화된 위임관리 프로토콜을 제안한다. 제안하는 기법은 기존의 위임서명 기법보다 강력하며 현실적인 PKI 기반 구조를 준용하므로 다양한 응용 환경에 적용할 수 있다.

Key Words : proxy signature, proxy management protocol, verifier

ABSTRACT

Proxy signature schemes based on delegation of warrant are frequently studied in these days. Proxy signature schemes that used in these days have some problems about the security. Especially, it is difficult to prevent misuse of certification and private key.

In this thesis, we propose the more stronger security structure by turning the point from original signer with proxy signer to original signer with verifier, and the proposed protocol is more stronger than previous scheme and can be applied to various PKI based application.

Ⅰ. 서 론

최근 유무선 인터넷의 발달에 따라 전자서명의 중요성은 점점 커지고 있다. 전자서명은 서명 권한이 있는 서명자에 의해서 이루어지며 제 3자에 의해 서명에 대한 유용성을 검증할 수 있어야 한다. 이러한 전자서명에는 여러 가지 다양한 기법들이 존재한다. 특히 서명의 권한을 가진 사람이 부득이한 사정으로 서명할 수 없는 상황에서는 서명자가 지정한 위임서명자를 통해 서명을 할 수 있다. 이러한 위임서명 기법은 권한의 위임을 위해 인증서와 개인키를 위임서명자에게 전송하는 방법을 일반적으로 사용하고 있다[3, 5]. 하지만 인증서가 가지고 있는 모든 권한을 위임하는 것은 인증서 및 개인키의 오남용이 가능하며 위임서명 후 위임서명자의 부인방지를 막을 수가 없고 위임을 받은 직원이 제 3자에게 원서명자의 동의 없이 인증서와 개인키를 유출시킴으로서 위임서명 능력을 가지게 할 수 있으

* 숭실대학교 컴퓨터학과 멀티미디어 연구실 (atprince@empal.com, (bigman2u@korea.com, javarian99@empal.com)
** 경원대학교 소프트웨어대학 (oh@kyungwon.ac.kr)
논문번호 : KICS2005-05-199, 접수일자 : 2005년 5월 19일

며 개인키 자체의 노출이 늘어남에 따라 안전성에 심각한 문제가 나타나는 등의 보안상으로 많은 문제점을 가지고 있다[2][7].

기존의 위임서명 기법에서는 원서명자, 위임서명자, 검증자의 관계로 구성되어 있으며 제안하는 기법도 세 부분의 관계를 기초로 하고 있다. 기존의 위임서명 기법에서는 원서명자와 위임서명자의 관계를 핵심으로 하여 원서명자가 위임서명자에게 개인키를 생성해 주거나 혹은 위임내용에 함께 전송하여 처리하지만 제안하는 위임관리 프로토콜은 기존의 원서명자와 위임서명자의 관계가 아니라 원서명자와 검증자의 관계로 관점을 전환하였다. 이러한 관점의 전환을 통하여 보안이 상대적으로 취약한 검증자에게 최우선으로 위임을 등록하기 때문에 효율적인 보안이 설정될 수 있다.

본 논문에서는 PKI 기반의 구성요소를 준용하며 원서명자와 위임서명자가 기존의 인증서를 발급받은 환경에서 원서명자가 위임서명자에 대하여 검증자를 통하여 위임정보를 등록하고 해제하며 실시간으로 권한을 변경할 수 있는 위임관리 프로토콜을 제안한다.

Ⅱ. 위임서명의 보안 요구사항

위임서명을 하기 위해서는 다음과 같은 8가지의 보안 요구사항들을 만족해야 한다.

2.1 검증성

검증자는 위임서명으로부터 원서명자의 서명 권한 위임에 대한 동의를 확인할 수 있어야 하며 선택적으로 위임서명자의 신원을 확인할 수 있어야 한다[4].

2.2 위조 불가능성

원서명자에 의해 지정된 위임서명자만이 유효한 서명을 생성할 수 있어야 하며 원서명자나 제 3자는 위임서명자를 가장하여 유효한 서명을 생성할 수 없어야 한다[6].

2.3 신원 확인성

누구나 위임서명으로부터 위임서명자의 신원을 확인할 수 있어야 한다.

2.4 부인 불가능성

위임서명자는 유효한 위임서명의 생성 후 서명한 사실에 대한 부인을 할 수 없어야 한다[1][9].

2.5 오용 방지

원서명자에 의해서 발급되어진 위임장은 원서명자가 지정한 범위 내에서 사용되어야 한다. 위임서명자는 원서명자로부터 위임받은 권한범위 이외에는 위임서명키를 사용할 수 없어야 한다[8][12].

2.6 권한의 제약

명백한 권한의 제약을 통하여 위임된 권한의 범위 안에서만 위임서명키를 사용할 수 있어야 한다[11][14].

2.7 양도 불가

위임서명자가 생성한 위임서명키는 제 3자에게 양도할 수 없어야 한다[10][13].

2.8 적합성 확인

검증자는 위임서명자의 위임서명 권한에 표시된 제약사항에 대한 적합성을 체크해야 한다[7].

Ⅲ. 구조적 보안강화를 통한 검증자 중심의 위임관리 프로토콜

3.1 위임관리 프로토콜의 구조

그림 1은 위임관리 프로토콜을 이용한 위임서명 기법의 구성요소와 시나리오를 나타내었다.

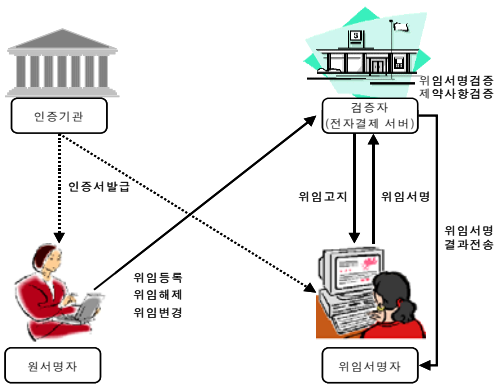

그림 1. 위임관리 프로토콜의 구조 및 구성요소

본 논문에서 제안하는 위임서명을 위한 보안 관리 프로토콜을 기술하기 위한 용어는 다음과 같다.

- *Park* : 원서명자
- *Lee* : 위임서명자
- *HTS* : 검증자
- S_{Park} : *Park*의 개인키

- M : 원문
- $_sS_{Park}(M)$: 원문 M을 $Park$의 개인키로 수행한 전자서명
- f : 서명의 목적을 명시하는 플래그
- P : 위임 내용
- L : 제약 사항
- R : 전송 결과
- T_{start} : 위임시작시간
- T_{end} : 위임종료시간
- $T_{registration}$: 위임등록시간
- T_{notice} : 전송고지시간
- $T_{confirm}$: 전송확인시간
- T_{sign} : 위임서명 생성시간
- $T_{revocation}$: 해제요청시간
- $reason$: 위임등록/해제 거부사유
- $change$: 변경내용

3.2 위임등록 프로토콜

그림 2는 위임등록 절차의 시나리오를 나타낸 것이다.

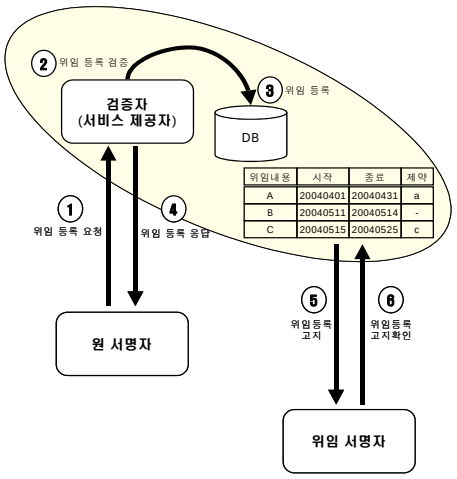

그림 2. 위임등록 시나리오

위임등록 프로토콜은 4가지의 토큰으로 구성되며 PRQ, PRR, PRN, PRC로 정의한다.

- $PRQ = {}_sS_{Park}(f_{PRQ}, HTS, Lee, P, L, T_{start}, T_{end})$
- $PRR = {}_sS_{HTS}(f_{PRR}, Park, R, T_{registration}, [f_{PRQ}, HTS, Lee, P, L, T_{start}, T_{end}, PRQ])$
- $PRN = {}_sS_{HTS}(f_{PRN}, Lee, T_{notice}, [f_{PRQ}, HTS, Lee, P, L, T_{start}, T_{end}, PRQ])$
- $PRC = {}_sS_{Lee}(f_{PRC}, HTS, R, reason, T_{confirm})$

1) $Park \rightarrow HTS : f_{PRQ}, HTS, Lee, P, L, T_{start}, T_{end}, PRQ$

Park은 HTS에게 Lee에 대하여 위임등록을 요청한다. Park은 Lee의 DN과 $Serial$ 번호를 포함하는 위임내용, 제약사항, 위임시작시간, 위임종료시간을 자신의 개인키로 서명하여 HTS에게 전송한다.

2) $HTS \rightarrow Park : f_{PRR}, Park, R, T_{registration}, [f_{PRQ}, HTS, Lee, P, L, T_{start}, T_{end}, PRQ], PRR$

HTS는 Park의 위임등록 요청에 대한 전자서명 및 위임등록 내용에 대한 유효성을 검증한다. Park의 위임등록이 유효하면 HTS의 데이터베이스에 상세하게 세분화된 위임권한을 설정한 후 위임등록에 대한 결과를 자신의 개인키로 서명하여 Park에게 응답한다. PRQ는 추후 위임등록 사항에 대한 부인방지를 위하여 추가하였다.

3) $HTS \rightarrow Lee : f_{PRN}, Lee, T_{notice}, [f_{PRQ}, HTS, Lee, P, L, T_{start}, T_{end}, PRQ], PRN$

HTS는 Park의 위임등록이 유효한 경우 Park의 위임내용에 대한 내용을 자신의 개인키로 서명하여 Lee에게 고지한다.

4) $Lee \rightarrow HTS : f_{PRC}, HTS, R, reason, T_{confirm}, PRC$

Lee는 HTS로부터 고지된 Park의 위임내용에 대해서 확인하고 위임에 대한 동의여부와 함께 인지하였다는 사실을 자신의 개인키로 서명하여 HTS에게 응답한다.

3.3 위임서명 프로토콜

그림 3은 위임서명 절차의 시나리오를 나타낸 것이다. 위임서명자는 원서명자로 위임받은 권한을 인

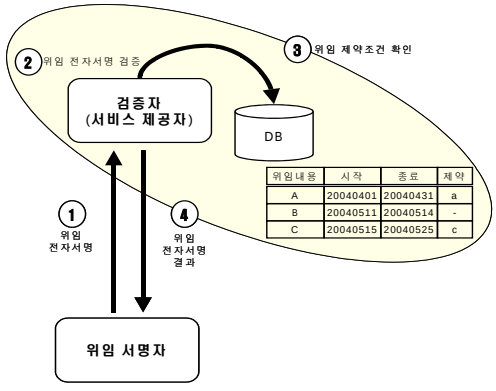

그림 3. 위임서명 시나리오

지한 후 원서명자를 대신하여 전자서명을 수행한다.

위임서명 프로토콜은 2가지의 토큰으로 구성되며 *PSS*, *PSR*로 정의한다.

- $PSS = {}_{s}S_{Lee}(f_{PSS}, HTS, M, T_{sign})$
- $PSR = {}_{s}S_{HTS}(f_{PSR}, Lee, R)$

1) $Lee \rightarrow HTS : f_{PSS}, HTS, M, T_{sign}, PSS$

Lee는 Park을 대신하여 *M*에 대하여 전자서명을 수행한다. Lee는 기존의 발급받은 공인된 인증서와 개인키를 사용하여 위임서명을 생성한다.

2) $HTS \rightarrow Lee : f_{PSR}, Lee, R, PSR$

HTS는 Lee의 서명에 대한 유효성을 검증하고 서명이 유효하면 HTS의 데이터베이스에 설정된 위임권한을 확인한다. HTS는 Lee의 서명이 위임등록 시 설정되었던 위임내용 및 제약사항과 비교하여 등록된 위임권한과 비교하여 검증한 후 위임서명시간이 위임시작시간과 위임종료시간의 범위내에 있는지를 검증한다. HTS는 Lee에게 위임서명에 대한 결과를 전송하며 이 결과로서 Lee는 위임서명의 유효여부를 확인할 수 있다.

3.4 위임해제 프로토콜

그림 4는 위임해제 절차의 시나리오를 나타낸 것이다. 원서명자는 위임기간 내에도 보안상 문제가 발생하거나 기타의 사유로 인해 실시간으로 해제할 수 있다.

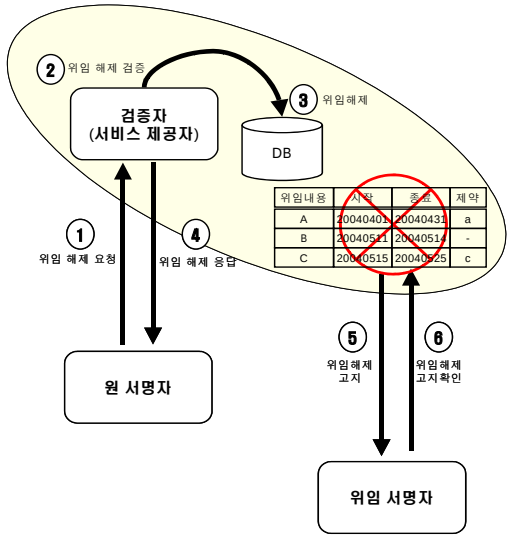

그림 4. 위임해제 시나리오

위임해제 프로토콜은 4가지의 토큰으로 구성되며 *RRQ*, *RRR*, *RRN*, *RRC*로 정의한다.

- $RRQ = {}_{s}S_{Park}(f_{RRQ}, HTS, reason, T_{revocation}, [f_{PRQ}, HTS, Lee, P, L, T_{start}, T_{end}, PRQ])$
- $RRR = {}_{s}S_{HTS}(f_{RRR}, Park, R, [f_{RRQ}, HTS, reason, T_{revocation}, [f_{PRQ}, HTS, Lee, P, L, T_{start}, T_{end}, PRQ], RRQ])$
- $RRN = {}_{s}S_{HTS}(f_{RRN}, Lee, T_{notice}, [f_{RRQ}, HTS, reason, T_{revocation}, [f_{PRQ}, HTS, Lee, P, L, T_{start}, T_{end}, PRQ], RRQ])$
- $RRC = {}_{s}S_{Lee}(f_{RRC}, HTS, R, T_{confirm})$

1) $Park \rightarrow HTS : f_{RRQ}, HTS, reason, T_{revocation}, [f_{PRQ}, HTS, Lee, P, L, T_{start}, T_{end}, PRQ]), RRQ$

Park은 필요한 시기에 Lee에 대한 위임등록을 해제할 수 있다. Park은 위임해제 내용을 자신의 개인키로 서명하여 HTS에게 전송한다.

2) $HTS \rightarrow Park : f_{RRR}, Park, R, [f_{RRQ}, HTS, reason, T_{revocation}, [f_{PRQ}, HTS, Lee, P, L, T_{start}, T_{end}, PRQ], RRQ], RRR$

HTS는 Park의 서명과 위임해제에 대한 내용을 검증하여 위임해제 요청을 반영하고 Lee는 Park이 설정한 권한 내에서의 위임권한을 실시간으로 상실한다. HTS는 Park의 위임해제 요청에 대해 응답하게 되고 위임해제 응답을 확보한 Park은 HTS에게 위임해제에 대한 부인방지 기능을 가진다.

3) $HTS \rightarrow Lee : f_{RRN}, Lee, T_{notice}, [f_{RRQ}, HTS, reason, T_{revocation}, [f_{PRQ}, HTS, Lee, P, L, T_{start}, T_{end}, PRQ], RRQ], RRN$

위임해제가 정상적으로 이루어졌을 경우 HTS는 Park의 위임해제에 대하여 Lee에게 고지하며 Lee는 자신의 위임권한이 상실되었음을 인지하게 된다.

4) $Lee \rightarrow HTS : f_{RRC}, HTS, R, T_{confirm}, RRC$

Lee는 위임해제 고지에 대하여 HTS에게 응답하게 되고 Lee가 전송한 위임해제 응답을 통해서 HTS는 위임해제 프로토콜이 종료되었음을 확인한다.

3.5 위임변경 프로토콜

그림 5는 위임변경 절차의 시나리오를 나타낸 것이다.

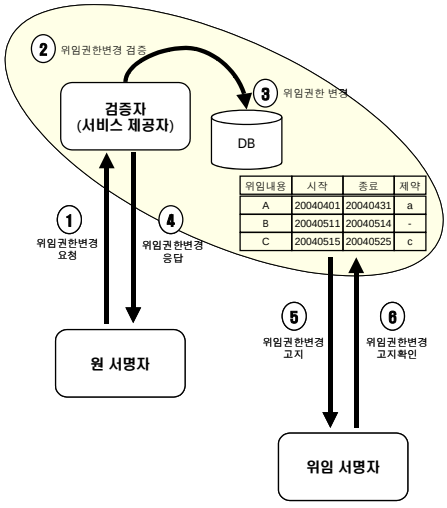

그림 5. 위임변경 시나리오

위임변경 프로토콜은 4가지의 토큰으로 구성되며 *RCQ, RCR, RCN, RCC*로 정의한다.

- $RCQ = {}_sS_{Park}(f_{RCQ}, HTS, change)$
- $RCR = {}_sS_{HTS}(f_{RCR}, Park, R, T_{change}, [f_{RCQ}, HTS, change, RCQ])$
- $RCN = {}_sS_{HTS}(f_{RCN}, Lee, T_{notice}, [f_{RCQ}, HTS, change, RCQ])$
- $RCC = {}_sS_{Lee}(f_{RCC}, HTS, R, T_{confirm})$

1) *Park* → *HTS* : f_{RCQ}, *HTS, change, RCQ*

Park은 필요한 시점에 위임에 대한 사항을 재등록 하지 않고 Lee에 대한 위임권한을 변경할 수 있다. Park은 위임변경 내용을 자신의 개인키로 서명하여 HTS에게 전송한다.

2) *HTS* → *Park* : f_{RCR}, *Park, R,* T_{change}, $[f_{RCQ}$, *HTS, change, RCQ],* RCR

HTS는 Park의 서명을 검증하고 변경할 위임내용에 대한 유효성을 검증하여 위임변경에 대한 요청을 반영하게 되고 Lee의 위임에 관련된 사항이 실시간으로 변경된다. HTS는 Park의 위임변경 요청에 대해 응답하게 되며 위임변경 응답을 확보한 Park은 HTS에게 위임변경에 대해 부인방지가 가능하다.

3) *HTS* → *Lee* : f_{RCN}, *Lee,* T_{notice}, $[f_{RCQ}$, *HTS, change, RCQ], RCN*

위임변경이 정상적으로 이루어졌을 경우 HTS는 Park의 위임변경에 대하여 Lee에게 고지하며 이를 통해서 Lee는 자신의 위임권한이 변경되었음을 확인하게 된다.

4) *Lee* → *HTS* : f_{RCC}, *HTS, R,* $T_{confirm}$, *RCC*

Lee는 위임변경 고지에 대하여 HTS에게 응답하게 되며 Lee가 전송한 위임변경 응답을 통해 HTS는 위임변경 프로토콜이 정상적으로 종료되었음을 확인하게 된다.

3.6 위임서명을 위한 보안 요구사항

1) 검증성

- $PRQ = {}_sS_{Park}(f_{PRQ}, HTS, Lee, P, L, T_{start}, T_{end})$
- $PSS = {}_sS_{Lee}(f_{PSS}, HTS, M, T_{sign})$

원서명자는 위임등록에 대한 사항을 검증자에게 최우선적으로 등록하게 되며 *PRQ* 토큰에는 *Lee*에 대하여 *P, L*의 위임내용과 제약사항으로서 위임한다는 내용이 포함되어 있다. 그러므로 검증자는 원서명자의 위임등록에 대한 정보를 보유하게 되며 위임등록이 정상적으로 이루어졌을 경우 위임서명자가 *PSS* 토큰을 이용하여 위임서명을 생성할 때 검증자는 위임서명에 대한 원서명자의 동의를 확인할 수 있다.

2) 위조불가능성

- $PRQ = {}_sS_{Park}(f_{PRQ}, HTS, Lee, P, L, T_{start}, T_{end})$
- $PSS = {}_sS_{Lee}(f_{PSS}, HTS, M, T_{sign})$

원서명자는 위임등록시 *PRQ* 토큰에 위임서명자의 *DN*을 명시하여 지정된 위임서명자만이 유효한 서명을 할 수 있다. 또한 위임서명자의 인증서에는 *DN*과 함께 *Serial* 번호가 포함되어 있으므로 동명이인과 같은 사용자의 충돌을 방지할 수 있다.

3) 신원확인성

- $PRQ = {}_sS_{Park}(f_{PRQ}, HTS, Lee, P, L, T_{start}, T_{end})$
- $PSS = {}_sS_{Lee}(f_{PSS}, HTS, M, T_{sign})$
- $P = DN|Serial|L|T$, $H = h(P)$, $S \equiv H^{dA} \bmod n_A$

검증자에게 M, S, $Cert_C$ *값을 전송*

$H' = h(M)$, $H \equiv S^{eC} \bmod n_c$,

$H = H'$ *검증*

원서명자는 *PRQ* 토큰에서 위임서명자의 *DN*과 *Serial* 번호를 지정하였고 위임서명자는 *PSS* 토큰을

이용하여 위임서명을 생성할 때 기존에 발급받은 공인된 인증서를 사용하였으며 검증자는 위임서명자의 인증서와 서명을 검증하는 절차를 수행하므로 위임서명으로부터 위임서명자의 신원을 확인할 수 있다. 이 절차는 기존의 인증서를 사용하기 때문에 일반적인 전자서명에 대한 신원확인 절차와 동일하다.

4) 부인 불가능성

- $PRQ = {}_sS_{Park}(f_{PRQ}, HTS, Lee, P, L, T_{start}, T_{end})$
- $PRR = {}_sS_{HTS}(f_{PRR}, Park, R, T_{registration}, [f_{PRQ}, HTS, Lee, P, L, T_{start}, T_{end}, PRQ])$
- $PRN = {}_sS_{HTS}(f_{PRN}, Lee, T_{notice}, [f_{PRQ}, HTS, Lee, P, L, T_{start}, T_{end}, PRQ])$
- $PSS = {}_sS_{Lee}(f_{PSS}, HTS, M, T_{sign})$

위임등록시 원서명자는 *PRQ* 토큰을 이용하여 검증자에게 위임내용과 제약사항을 상세히 명시하고 *PRR* 토큰과 *PRN* 토큰은 *PRQ* 토큰의 데이터와 전자서명값을 포함하고 있으므로 원서명자 및 검증자 관점에서 원서명자의 위임등록, 해제, 변경에 대해서 부인할 수 없으며 위임서명자는 자신의 인증서와 개인키로 서명을 생성하고 위임서명 시간이 기록되므로 위임서명자는 서명 생성 후 부인할 수 없다.

5) 오용 방지

- $PRQ = {}_sS_{Park}(f_{PRQ}, HTS, Lee, P, L, T_{start}, T_{end})$
- $PSS = {}_sS_{Lee}(f_{PSS}, HTS, M, T_{sign})$
- $PSR = {}_sS_{HTS}(f_{PSR}, Lee, R)$

위임등록시 원서명자는 *PRQ* 토큰을 이용하여 검증자에게 위임내용과 제약사항을 상세히 명시하며 위임서명자가 *PSS* 토큰을 이용하여 위임서명을 생성할 때 검증자는 위임서명에 대한 유효성 검증와 함께 위임에 대한 제약사항을 검증하는 절차를 수행하므로 위임된 권한 밖의 내용에 대해서는 *PSR* 토큰의 결과값 *R*에 위임서명에 대한 *false*값이 포함되어 전송된다.

6) 권한의 제약

- $PRQ = {}_sS_{Park}(f_{PRQ}, HTS, Lee, P, L, T_{start}, T_{end})$
- $L \in DB\_Info_{original\_signer}$

원서명자는 *PRQ* 토큰을 이용하여 위임내용과 제약사항을 상세히 설정할 수 있으며 제약사항은 필요한 서비스의 종류에 따라서 확장할 수 있다. 검증자는 *PRQ* 토큰을 전송받게 되면 원서명자가 위임한 제약사항의 유효성에 대한 검증절차를 수행하므로 원서명자는 정당한 위임 제약사항만을 등록할 수 있다.

7) 양도 불가

- $PSS = {}_sS_{Lee}(f_{PSS}, HTS, M, T_{sign})$

제안하는 구조에서는 기존의 위임키가 위임서명자의 개인키로 대체되었으므로 위임서명자에게는 양도가 가능한 위임키가 존재하지 않는다. 또한 위임등록시 원서명자가 위임서명자의 *DN*과 *Serial*을 지정하였고 위임서명자의 개인키는 자신만이 알고 있으므로 양도가 불가하다.

8) 적합성 확인

- $PRQ = {}_sS_{Park}(f_{PRQ}, HTS, Lee, P, L, T_{start}, T_{end})$
- $PSS = {}_sS_{Lee}(f_{PSS}, HTS, M, T_{sign})$
- $PSS \in L$

검증자는 위임서명자가 *PSS* 토큰을 이용하여 위임서명을 생성하였을 때 위임서명이 *PRQ* 토큰의 위임내용 *P*, 제약사항 *L*의 범위내에 있는지 적합성을 확인하는 절차를 수행하며 T_{sign}이 T_{start}와 T_{end}의 범위내에 존재해야만 유효한 위임서명이 가능하다.

9) 실시간 처리

- $RRQ = {}_sS_{Park}(f_{RRQ}, HTS, reason, T_{revocation}, [f_{PRQ}, HTS, Lee, P, L, T_{start}, T_{end}, PRQ])$
- $RRR = {}_sS_{HTS}(f_{RRR}, Park, R, [f_{RRQ}, HTS, reason, T_{revocation}, [f_{PRQ}, HTS, Lee, P, L, T_{start}, T_{end}, PRQ], RRQ])$
- $RCQ = {}_sS_{Park}(f_{RCQ}, HTS, change)$
- $RCR = {}_sS_{HTS}(f_{RCR}, Park, R, T_{change}, [f_{RCQ}, HTS, change, RCQ])$

RRQ 토큰을 이용한 위임해제는 검증자가 원서명자의 인증서와 서명을 검증하여 유효하다면 해당되는 위임등록에 관한 사항을 데이터베이스에서 바로 삭제하고 *RRR* 토큰을 이용하여 결과값을 전송하는 절차를 수행하므로 실시간적이고 능동적인 위임해제가 가능하다. *RCQ* 토큰을 이용한 위임변경은 검증자가 원서명자의 인증서와 서명 및 위임변경에 관한 제약사항을 검증하여 유효하면 해당되는 기존의 위임등록 사항을 변경하고 *RCR* 토큰을 이용하여

결과값을 전송하는 절차를 수행하므로 위임변경을 위한 위임재등록 과정이 필요하지 않으며 실시간적인 위임변경이 가능하다.

10) 위임키쌍 필요여부

- $PRQ = {}_{s}S_{Park}(f_{PRQ}, HTS, Lee, P, L, T_{start}, T_{end})$
- $PRR = {}_{s}S_{HTS}(f_{PRR}, Park, R, T_{registration}, [f_{PRQ}, HTS, Lee, P, L, T_{start}, T_{end}, PRQ])$
- $PSS = {}_{s}S_{Lee}(f_{PSS}, HTS, M, T_{sign})$
- $PSR = {}_{s}S_{HTS}(f_{PSR}, Lee, R)$

원서명자는 *PRQ* 토큰을 이용하여 위임등록을 할 때 자신의 개인키로 서명하여 검증자에게 전송하며 검증자는 유효성 검증절차를 수행하고 *PRR* 토큰을 이용하여 결과를 전송한다. 위임등록 절차가 정상적으로 수행된 후 위임서명자가 *PSS* 토큰을 이용하여 위임서명을 생성할 때에도 마찬가지로 원문 *M*에 대해서 자신의 개인키로 서명하여 검증자에게 전송하고 검증자는 위임서명자의 서명과 서명에 대한 제약사항을 검증한 후 *PSR* 토큰을 이용하여 결과를 전송한다. 즉, 위임서명자는 전자서명을 생성할 경우와 동일하게 위임서명을 생성할 경우에도 기존의 발급된 인증서와 자신의 개인키를 사용하므로 위임을 위한 위임키쌍을 따로 생성할 필요가 없다.

11) 보안채널 필요여부

- $PRQ = {}_{s}S_{Park}(f_{PRQ}, HTS, Lee, P, L, T_{start}, T_{end})$
- $PRR = {}_{s}S_{HTS}(f_{PRR}, Park, R, T_{registration}, [f_{PRQ}, HTS, Lee, P, L, T_{start}, T_{end}, PRQ])$
- $PSS = {}_{s}S_{Lee}(f_{PSS}, HTS, M, T_{sign})$
- $PSR = {}_{s}S_{HTS}(f_{PSR}, Lee, R)$

제안하는 구조에서는 *PRQ*, *PRR* 토큰을 이용하여 원서명자가 검증자에게 직접 위임등록 절차를 수행하므로 기존의 원서명자가 위임서명자에게 전송했었던 위임장을 생성할 필요가 없고 위임서명자가 *PSS*, *PSR* 토큰을 이용하여 검증자와 직접적으로 위임서명 절차를 수행하는 과정에서 위임키쌍의 생성이 필요하지 않으므로 위임장과 위임키쌍을 비밀리에 전송할 보안채널이 필요하지 않다.

Ⅳ. 성능 평가

그림 6에서는 기존 위임서명 기법들의 구조상에서 위임서명이 이루어지는 절차를 나타내었다. 기존

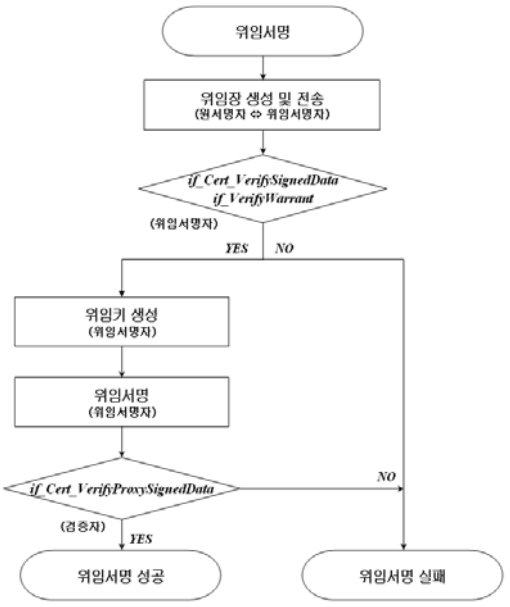

그림 6. 기존 기법들의 위임서명을 위한 절차

위임서명 기법들의 구조에서는 원서명자와 위임서명자의 관점에서 위임서명을 위한 절차들이 수행된다. 원서명자는 위임서명자에게 위임에 관한 정보를 포함하는 위임장을 생성하여 자신의 개인키로 서명하고 이를 위임서명자에게 전송한다. 위임서명자는 원서명자의 서명과 위임장에 대한 유효성을 검증하여 검증결과가 유효하면 위임장의 정보를 이용해서 위임서명키를 생성하고 생성된 위임서명키를 이용하여 위임서명을 생성하게 된다. 원서명자와 위임서명자의 서명 혹은 위임장에 대한 유효성이 검증되지 않는다면 위임서명 절차는 종료된다.

기존의 기법에서는 위임서명자에 의해서 생성된 위임서명만이 검증자에 의해서 유효성을 검증받게 된다. 즉, 실질적으로 검증을 처리하는 검증자는 위임서명 자체에 대한 검증절차 부분만을 수행하게 되므로 상대적으로 보안이 취약하다. 이러한 문제를 해결하기 위하여 제안하는 구조에서는 원서명자와 위임서명자와의 관점이 아닌 원서명자와 검증자와의 관점에서 위임서명을 위한 절차들이 수행될 수 있도록 구성하였다. 기존의 기법에서 상대적으로 보안이 취약했던 검증자에게 원서명자가 최우선적으로 위임등록을 요청하고 검증자는 원서명자의 서명과 위임내용 및 제약사항에 관한 유효성을 검증하는 절차를 수행함으로서 기존의 위임서명 기법의 구조에서 상대적으로 취약했던 검증자의 보안성을 강화시켰다.

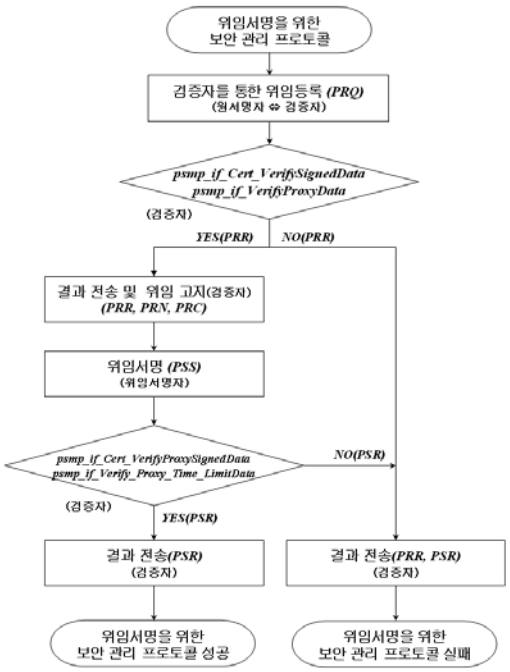

그림 7. 제안하는 기법의 위임서명을 위한 절차

그림 7에서는 제안하는 기법의 구조상에서 위임서명 절차를 나타내었다. 원서명자는 검증자와 최우선적으로 위임에 관한 사항을 등록하고 원서명자의 서명 및 위임등록 사항이 모두 검증자를 통해서 이루어진다. 검증자가 위임서명자에게 위임등록에 대한 사항을 고지하고 응답을 받음으로서 원서명자와 위임서명자의 위임등록과 위임확인에 대한 사항을 모두 보유하게 된다. 위임서명자에 의해서 위임서명이 생성 되었을 경우 검증자는 보유된 정보를 기반으로 하여 등록된 위임권한에 벗어나지 않는 범위내에서 유효한 위임서명이 이루어지도록 검증절차를 수행한다. 그러므로 제안하는 구조에서는 실질적으로 검증을 처리하는 검증자에게 모든 정보가 보유되어 있고 검증자가 직접 검증절차를 수행하므로 기존의 위임서명 기법보다 구조적인 차원에서 보안이 강화되었다.

V. 결 론

위임서명은 사용자가 자신의 서명 권한을 위임할 필요가 있을 경우 유용하게 사용될 수 있는 기술이다. 그러나 인터넷과 같은 분산 환경에서 원서명자나 위임서명자를 신뢰하기는 매우 어려운 문제이기 때문에 안전한 위임서명 기법에 대한 연구는 매우 중요하다. 본 논문에서 제안하는 위임관리 프로토콜은 기존의 원서명자와 위임서명자와의 관계를 원서명자와 검증자와의 관계로 관점을 전환함으로서 실질적으로 검증을 처리하는 검증자의 보안을 강화시켰다. 그리고 원서명자가 검증자와의 위임등록 절차를 수행하므로 기존 위임서명 기법에서 원서명자가 생성하였던 위임장이 필요하지 않으며 위임서명자가 기존의 발급받은 공인된 인증서와 개인키를 사용하기 때문에 위임서명을 위한 위임키쌍도 생성할 필요가 없다. 이러한 이유로 위임서명을 위한 전체적인 수행속도도 향상되었으며 위임장 혹은 위임키쌍을 비밀리에 전송할 보안채널의 필요성도 사라지게 된다. 또한 원서명자의 필요에 의해서 위임서명자에 대한 위임등록사항을 해제할 수 있으며 위임등록사항을 해제하지 않은 상황에서 등록된 위임권한의 변경 또한 가능하다. 위임해제와 위임변경 절차는 실시간으로 이루어지기 때문에 원서명자가 원하는 시점에 위임해제 요청과 위임변경 요청을 하여 절차가 정상적으로 수행되었을 경우 그 이후의 위임서명 생성시에 바로 적용이 가능하다. 이러한 관점에서 볼 때 기존의 실시간으로 이루어지지 않은 경우에 비하여 보안이 강력하다.

본 논문에서 제안된 위임관리 프로토콜은 PKI 기반 구조를 준용하고 실제 환경의 보안 요구사항을 분석하여 이루어지며 정의된 요구 사항에 맞는 구조를 제공할 수 있다. 그러므로 전자 상거래에서의 위임서명을 보다 안전하게 제공할 수 있고 전자결제 및 증권거래 시스템과 같은 응용 환경에 적용될 수 있다.

참 고 문 헌

[1] D.Chaum and H.Van Antwerpen, "Undeniable signatures," *Proc. Cryptology-CRYPTO'89 Proceedings, Springer-Verlag,* pp.458-464, 1990.

[2] B.Lee, H.Kim and K.Kim, "Self-certificate: PKI using self-certified key," *Proc. of Conference on Information Security and Cryptology,* pp.1-9, 2000.

[3] B.Lee, H.Kim and Y.Chang, "Efficient proxy-protected proxy signature scheme based on discrete logarithm," *Proceedings of 10th Conference on Information Security, Hualien, Taiwan, ROC,* pp.4-7, 2000.

[4] L.Yi, G.Bai and G.Xiao, "Proxy multi-signature scheme: A new type of proxy sig-

nature scheme," *Electronics Letters, Vol.36, No.6,* pp.527-528, 2000.

[5] M.Abe and T.Okamoto, "Provably secure partially blind signatures," *In Advances in Cryptology Crypto'2000,* pp.271-299, 2000.

[6] M.Hwang, I.Lin and E.J.Lu, "A secure non-repudiable threshold proxy signature scheme with known signers," *International Journal of Informatica,* vol.11, no.2, pp.1-8, 2000.

[7] S.M.Yen, C.P.Hung and Y.Y.Lee, "Remarks on some proxy signature schemes," *Proceedings of the 2000 ICS: Workshop on Cryptology and Information Security,* pp.54-59, 2000.

[8] B.Lee, H.Kim and K.Kim, "Strong proxy signature and its applications," *Proc. of SCIS 2001, International Conference on Information Security*, pp.603-608, June. 2001.

[9] D.R.Stinson and R.Strobl, "A provably secure distributed Schnorr signatures and *a(t,n)* threshold scheme for implicit certificates," *Sixth Australasian Conference on Information Security and Privacy LNCS 2119, Springer-Verlag,* pp.417-434, 2001.

[10] G.Itkis and L.Reyzin, "Forward-secure signatures with optimal signing and verifying," *Crypto'01,* pp.68-72, 2001.

[11] S.J.Hwang and C.C.Chen, "A new proxy multi signature scheme," *International workshop on cryptology and network security, Tamkang University Taipei, Taiwan,* pp.26-28, 2001.

[12] K.Shum and K.Wei, "A strong proxy signatures scheme with proxy signer privacy protection," *Proceeding of Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises,* pp.55-56, 2002.

[13] J.Cha and J.Cheon, "An identity-based signature from gap diffie-hellman groups," *Springer-Verlag, Advances in Cryptology, Proceedings of PKC '03, LNCS 2567,* pp. 18-30, 2003.

[14] J.Hwang, D.Lee and J.Lim, "Digital signature scheme with restriction on signing capability," *Proc. of ACISP 2003, LNCS vol.2727,* pp.324-335, 2003.

박 세 준 (Se-Joon Park) 정회원

1996년 2월 숭실대학교 수학과 학사
1998년 2월 숭실대학교 컴퓨터학과 석사
2004년 8월 숭실대학교 컴퓨터학과 박사
<관심분야> 멀티미디어, 암호학, 유무선 PKI

이 용 준 (Yong-Joon Lee) 정회원

1999년 2월 강남대학교 전사계산학과 학사
2001년 2월 숭실대학교 컴퓨터학과 석사
2005년 2월 숭실대학교 컴퓨터학과 박사
<관심분야> 정보보호, 암호학, 유무선 PKI

오 동 열 (Dong-Yeol Oh) 정회원

1999년 경희대학교 전자계산학과 졸업
2002년 숭실대학교 컴퓨터학과 석사
2004년 숭실대학교 컴퓨터학과 박사 수료
<관심분야> 유비쿼터스 컴퓨팅, P2P, 멀티미디어

오 해 석 (Hae-Seok Oh) 정회원

1975년 서울대학교 응용수학과 학사
1981년 서울대학교 계산통계학과 석사
2004년 서울대학교 계산통계학과 박사
1982년~2003년 숭실대학교 정보과학대학 교수
1976년~1982년 태평양화학(주), (주)삼호 전산실
1990년~1991년 일본 동경대학교 객원교수
1997년~1999년 숭실대학교 부총장
2000년~2001년 스탠포드대학교 객원교수
2003년~현재 경원대학교 소프트웨어대학 교수
<관심분야> 멀티미디어, 데이터베이스, 영상처리, 정보보호, 멀티미디어 암호, 암호학, 유무선 PKI