DRM 시스템을 위한 익명성을 갖는 효율적인 라이센스 다운로드 방식†

김소진*, 최재귀**, 김창수*, 박지환*
*부경대학교 전자계산학과
**부경대학교 정보보호학과

An Efficient License Download Method with Anonymity for DRM System

So-Jin Kim*, Jae-Gwi Choi**, Ji-Hwan Park*
*Dept of Computer Science, PuKyong University

요 약

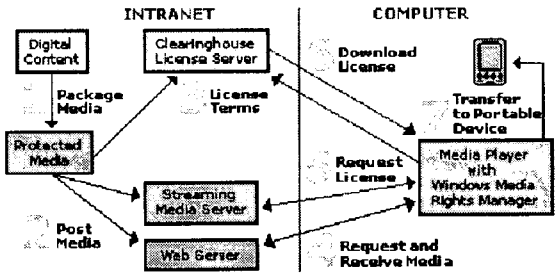
DRM(Digtal Right Management)은 컨텐츠의 불법 사용을 방지하여 저작권을 보호하고, 컨텐츠의 생성·유통·사용·관리 등에 필요한 모든 처리를 지원하는 종합 솔루션이다. DRM 체계에서 사용자(customer)는 컨텐츠를 정당하게 사용하기 위해서 클리어링하우스(clrearinghouse)로부터 라이센스를 발급받아야 한다. 일반적으로 라이센스(license)는 사용자의 공개키로 암호화된다. 그래서 사용자의 계산적 부담이 크다. 특히 무선 DRM 환경이라면, 사용자의 부담은 더 클 것이다. 따라서 본 논문은 익명성을 갖는 제안 1회용 대리서명 기법과 Y.Zheng의 Singcryption 기법을 적용하여 사용자의 계산량을 줄이고, 익명성을 갖는 효율적인 라이센스 다운로드 방식을 제안한다.

1. 서론

DRM(Digital Rights Management)은 다양한 채널을 통해 유통되는 전자서적, 음악파일, 영상정보, 게임, 소프트웨어, 이미지 등의 각종 디지털 컨텐츠를 불법 복제로부터 안전하게 보호하고, 이렇게 보호된 컨텐츠를 사용하게 함으로 컨텐츠 서비스의 유료화를 가능하게 하는 기술이다. 그러므로 사용자(customer)는 컨텐츠의 정당한 사용을 위해서 얼마의 돈을 지불하고, 클리어링하우스(clearinghouse)로부터 라이센스(license)를 발급받아야 한다. 일반적으로 라이센스는 사용자의 공개키로 암호화된다. 그래서 사용자의 계산적 부담이 크다. 특히 무선 DRM 환경인 경우, 사용자의 단말기는 유선 시스템에 비해 메모리 용량과 계산 능력이 떨어지므로 그 부담이 더 클 것이다. 따라서 본 논문은 사용자의 계산량을 줄이고, 익명성을 추가한 효율적인 라이센스 다운로드 방식을 제안한다.

2. WMRM의 라이센스 다운로드 방식[1,2]

Microsoft의 DRM인 WMRM(Windows Media Rights Manager)은 자사의 Windows OS에 포함하고 있는 윈도우미디어플레이어 버전6.4부터 이미 탑재되어 배포되었으므로 전 세계적으로 4억개 가까이 설치되어있다. 따라서 사용자가 별도의 클라이언트 모듈을 설치할 필요가 없기 때문에 널리 사용되고 있다.



<그림1> Windows Media Rights Manager Flow

WMRM에서는 컨텐츠와 라이센스가 분리되어 배포된다. 그리하여 전체적인 시스템의 관리가 용이하다.

컨텐츠는 미리 대칭키 암호로 암호화되어 있고, 쇼핑몰과 같은 공개장소에서 자유로이 다운로드 및 재분배가 가능하다. 라이센스는 컨텐츠의 처음 사용시, 혹은 사용자의 요청에 의해 클리어링하우스에서 발행된다. 클리어링하우스는 DRM의 핵심 부분으로 사용자 인증, 라이센스 발행 서비스 및 정산처리 등을 담당한다. 라이센스는 컨텐츠 사용의 시작일, 사용기간,

†본 연구는 ITRC 연구지원에 의해 수행되었음.

운용횟수, 운용방식 등에 관한 권한을 규정하고, 컨텐츠 ID, 컨텐츠 복호키, 라이센스 인증서 등을 포함한다.

▶ 라이센스 다운로드 방식

• 표기법

- x_u, y_u: 사용자의 비밀키, 공개키
  $x_u \in Z_q^*$, $y_u \equiv g^{x_u} \pmod{p}$
- x_c, y_c: 클리어링하우스의 비밀키, 공개키
  $x_c \in Z_q^*$, $y_c \equiv g^{x_c} \pmod{p}$
- req_u: 사용자가 작성한 컨텐츠의 사용조건
- S_u, ID_u: 사용자의 서명, ID
- m: 라이센스
- Key: 컨텐츠의 복호키
- $rights$: 권리명세언어로 생성된 컨텐츠의 사용조건
- S_c: 클리어링하우스의 서명
- E_{y_u}: 사용자의 공개키 (y_u)로 암호

가. 라이센스 요청

① 문서(inf_u, req_u)작성: 사용자는 개인정보(inf_u)와 컨텐츠에 대한 사용조건(req_u) 작성

② 사용자의 문서(inf_u, req_u, S_u, $certificate_{사용자}$)와 컨텐츠 헤더정보 전송: 사용자는 작성된 inf_u, req_u를 자신의 비밀키로 서명하고, 인증서, 컨텐츠 헤더정보($Key\ ID$ 등) 등과 함께 클리어링하우스로 전송하여 라이센스 요청

나. 인증 및 라이센스 발급

① 인증 및 서명 검증: 클리어링하우스는 전달받은 인증서와 서명값으로 정당한 사용자임을 확인하고, 정당한 문서임을 확인

② 라이센스 생성: 요청된 컨텐츠의 사용요구 조건에 맞게 다음과 같은 정보들을 포함한 라이센스(m)를 생성

m=($Key\ ID$, Key, $rights$, $certificate_{라이센스}$ 등)

③ 라이센스 서명 및 암호화: 클리어링하우스는 생성된 라이센스를 자신의 비밀키로 서명하고, 사용자의 공개키로 암호화($E_{y_u}(m)$, $S_c(m)$)

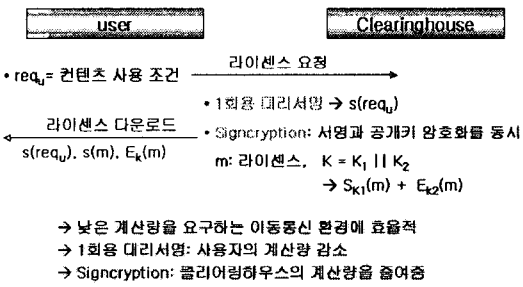
다. 라이센스 전송

클리어링하우스는 암호화된 라이센스를 서명값과 함께 안전한 채널을 통해 사용자에게 전송한다.

WMRM의 라이센스는 공개키 암호를 사용하기 때문에 라이센스에 많은 정보와 기능을 부가할수록 라이센스 검증 및 획득시 사용자의 계산적 부담이 커진다. 특히 무선 DRM의 경우, 사용자의 단말기가 제한적인 메모리 용량과 계산력으로 부담은 더 크게 될 것이다. 그러므로 익명성을 갖는 일회용 대리서명[3]과 Signcryption[4]을 이용한 효율적인 라이센스 다운로드 방식을 제안한다.

3. 효율적인 라이센스 다운로드 방식

제안 방식은 대리 서명과 Singcrypiton을 이용한 방식이다. 대리 서명을 통해서는 사용자의 서명을 클리어링하우스가 대신 수행하여 사용자의 계산량을 감소시키고, Singcryption을 통해서는 대리 서명에 따른 클리어링하우스의 추가된 계산량을 줄이고자 한다. Signcryption 기법은 메세지의 서명과 암호화를 동시에 해결하는 방식으로 클리어링하우스의 라이센스 서명과 암호화를 한꺼번에 해결하기 위해 사용한다.



<그림2> 제안 라이센스 다운로드 방식

그림2의 제안 방식에서는 라이센스 요청시 사용자의 개인 정보, 인증서 등에 대한 전송은 생략하고, 컨텐츠의 사용조건 req_u만을 전송함을 가정한다. 그리고 req_u는 시간정보를 포함한 값이므로 일정한 값으로 설정됨을 전제한다. 또한 공개키의 인증과 공개를 위한 등록센터를 가정하며, 사용자의 익명성을 보장하는 임시 비밀키/공개키 쌍은 이를 통해 생성하고, 등록한다. 여기서 일회용 대리서명은 KCP 방식[3]을 사용한다.

• 표기법

- x_{c_i}, y_{c_j}, y_{c_k}: 클리어링하우스의 비밀키, 공개키
  $x_{c_i} \in Z_q^*$ $i=\{1,2,3,4\}$
  $y_{c_j} \equiv g^{x_{c_j}}$ $j=\{1,2\}$, $y_{c_k} \equiv y_r^{x_{c_k}}$ $k=\{3,4\}$
- bid_c: 클리어링하우스의 컨텐츠 확인/판매 정보
- msg: h(임시ID_u, ID_c, req_u, bid_c), 대리서명메세지

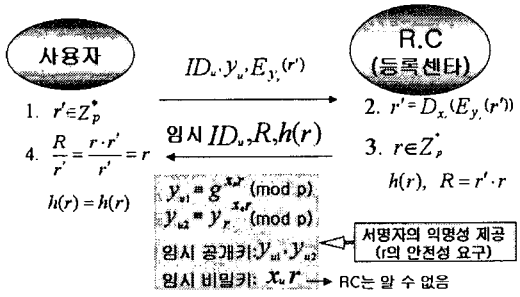
- m: 라이센스,
  $m=(Key\ ID,\ Key,\ rights,\ Certificate_{라이센스}$ 등)

• 등록센터(R.C)의 초기화

- $x_r,\ y_r$: 등록센터의 비밀키, 공개키
  $x_r \in Z_q^*,\ y_r \equiv g^{x_r} \pmod p$
- p : 512비트 이상의 큰 소수, 공개
- q : $q|p-1$인 큰 소수, 공개
- g : $\alpha \in Z_p^{order\ q}$, 공개

가. 등록

사용자는 임시 비밀키/공개키 쌍을 얻기 위해 그림 3과 같은 과정을 수행하여 임시ID_u를 얻고, 자신이 계산한 $h(r)$ 값과 전송받은 $h(r)$ 값이 같으면, y_{u_1}, y_{u_2}, $x_u r$를 생성한다. 이때 r의 유효 기간에 따라 등록 횟수는 달라진다. 만약 r의 유효 기간을 설정하지 않는다면, 등록은 한번만 하면 된다. 그러면 다음 등록 단계는 생략하고, 라이센스 요청 단계를 수행하면 된다. 등록센터는 자신의 DB에 ID_u, y_u, $E_{y_r}(r)$를 저장하고, 모든 임시 공개키는 공개한다.



<그림3> 임시 비밀키/공개키 생성 과정

나. 라이센스 요청

등록단계에서 생성한 임시ID_u, y_{u_1}, y_{u_2}, $x_u r$를 가지고 사용자는 아래와 같이 위임키를 생성하여 클리어링하우스에게 전송하고, 라이센스를 요청한다.

① $k \in Z_{p-1},\ K_1 \equiv g^k,\ K_2 \equiv y_r^{\ k} \pmod p$

② 임시$ID_u,\ req_u,\ K_1, K_2$로 다음을 계산
$e = h(\text{임시}ID_u, req_u, K_1, K_2)$

③ 자신의 임시 비밀키 $x_u r$로 위임키 생성
$s \equiv x_u r \cdot e + k \pmod q$

④ 클리어링하우스에게 임시ID_u, K_1, K_2, s, req_u, e 을 전송

다. 위임키 검증 및 대리서명

클리어링하우스는 컨텐츠 확인정보 bid_c를 이용하여 서명할 메시지 $msg = h(\text{임시}ID_u, ID_c, req_u, bid_c)$를 계산한 후, y_{u_1}, y_{u_2}, K_1, K_2, s, req_u를 이용하여 위임키를 검증하고, 대리서명을 수행한다.

① $g^s \equiv y_{u_1}^{h(\text{임시}ID_u,\ req_u,\ K_1,\ K_2)} \cdot K_1 \pmod p$
$y_r^{\ s} \equiv y_{u_2}^{h(\text{임시}ID_u,\ req_u,\ K_1,\ K_2)} \cdot K_2 \pmod p$

② 대리 서명키 s_1, s_2, s_3, s_4 생성
$s_1 \equiv s + x_{c_1},\ s_2 \equiv s + x_{c_2} \pmod q$
$s_3 \equiv s + x_{c_3},\ s_4 \equiv s + x_{c_4} \pmod q$

④ 대리 서명에 대한 공개키 β 계산
$\beta \equiv g^{(s_1+s_2)} \cdot y_r^{\ (s_3+s_4)} \pmod p$

⑤ 메시지 msg에 대한 서명 σ_1, σ_2 생성
$\sigma_1 \equiv (s_1 + s_2)/(msg + x_{c_1} - x_{c_2}) \pmod q$
$\sigma_2 \equiv (s_3 + s_4)/(msg + x_{c_3} - x_{c_4}) \pmod q$

라. 라이센스 생성

클리어링하우스는 컨텐츠의 $Key\ ID$, Key, $rights$ 등을 포함하는 라이센스 $m=(Key\ ID,\ Key,\ rights,\ Certificate_{라이센스}$ 등)를 생성한 후, Signcrytpion을 이용하여 다음과 같이 라이센스를 암호화하고 서명한다.

① 비밀키 K는 사용자의 공개키를 이용해서 생성하고, K_1, K_2로 분리
$k' \in Z_{p-1},\ K = (y_{u_1} \cdot y_{u_2})^{k'} \pmod p$
$K = K_1 \| K_2$

② K_1, K_2로 라이센스 m를 암호화 및 해쉬함수 적용
$E = E_{K_1}(m),\quad H = h_{K_2}(m)$

④ 메세지 H에 대한 서명 τ_1, τ_2 생성
$\tau_1 \equiv k'/(H + x_{c_1} + x_{c_2}) \pmod q$
$\tau_2 \equiv k'/(H + x_{c_3} + x_{c_4}) \pmod q$

마. 라이센스 다운로드

클리어링하우스는 자신의 ID_c, 대리서명에 대한 bid_c, msg, β, (σ_1, σ_2)과 라이센스 발급에 대한 E, H, (τ_1, τ_2)를 사용자에게 전송한다.

바. 대리서명 검증

사용자는 전달받은 ID_c, bid_c, msg, β, (σ_1, σ_2)값들을 이용하여 클리어링하우스가 생성한 대리서명에 대한 인증을 수행한다.

① $m=h$(임시ID_u, ID_c, req_u, bid_c)로 $m=msg$ 검사

② 서명 공개키를 검증하여 클리어링하우스가 서명키를 정당하게 생성했는지 검증

$$\beta \equiv y_{u_1}^{2e} \cdot K_1^2 \cdot y_{c_1} \cdot y_{c_2} \cdot y_{u_2}^{2e} \cdot K_2^2 \cdot y_{c_3} \cdot y_{c_4}$$

③ 서명 검증

$$\beta \equiv (y_{c_1} \cdot y_{c_2}^{-1} \cdot g^{msg})^{\sigma_1} \cdot (y_{c_3} \cdot y_{c_4}^{-1} \cdot y_r^{msg})^{\sigma_2}$$

사. 라이센스 획득

사용자는 전달받은 $E, H, (\tau_1, \tau_2)$ 값들을 이용하여 클리어링하우스가 발급한 라이센스를 획득한다.

① 자신의 비밀키 $x_u r$를 이용하여 다음과 같은 K 값을 계산

$$K \equiv (y_{c_1} \cdot y_{c_2} \cdot g^H)^{\tau_1 \cdot x_u r} \cdot (y_{c_3} \cdot y_{c_4} \cdot y_r^H)^{\tau_2 \cdot x_u r}$$

② K를 $K_1 \| K_2$로 나누고, 라이센스 m을 복호화

$$m = D_{K_1}(E)$$

③ 위의 식에서 얻은 m으로, $H' = h_{K_2}(m)$을 계산

④ $H' = H$인 경우에만, 정당한 라이센스를 획득

4. 제안 방식의 평가 및 고찰

① **안전성** - 전체적으로 이산대수 문제의 어려움에 기반하며, 일회용 대리서명과 Signcryption의 안전성에 기반한다.

- **대리서명**: 일회용 대리서명의 안전성에 기반한다.
- **라이센스 암호화 및 서명**: Signcryption의 안전성에 기반한다.

② **라이센스의 기밀성** - 이산대수 문제의 어려움에 기반하여 정당한 사용자만이 라이센스를 복호화하고, 서명을 확인할 수 있다.

$$K \equiv (y_{u_1} \cdot y_{u_2})^{k'}$$

$$\equiv (y_{c_1} \cdot y_{c_2} \cdot g^H)^{\tau_1 \cdot x_u r} \cdot (y_{c_3} \cdot y_{c_4} \cdot y_r^H)^{\tau_2 \cdot x_u r}$$

③ **클리어링하우스의 부정 방지** - 대리서명이 일회성을 보장함으로 부정을 방지할 수 있다[정리3].

④ **인증성** - 대리서명과 라이센스 발급시에 송/수신자를 지정하여 계산과정에 포함시켜 인증성을 제공하고, 전송도중 공격자로부터의 위조 및 변경에 대한 확인도 가능하다.

⑤ **효율성** - 라이센스 다운로드는 대리서명과 Signcryption으로 처리함으로 사용자측의 계산량이 낮아진다.

⑥ **부인봉쇄** - 대리서명시 사용자와 클리어링하우스의 비밀정보를 포함하고, Singcryption의 키 생성에도 사용자와 클리어링하우스의 비밀정보를 포함하여 서로에 대한 부정을 할 수 없다.

⑦ **익명성** - 이산대수 문제에 기반한 사용자의 익명성을 보장한다. 임시 공개키 $y_{u_1} \equiv g^{x_u r}$, $y_{u_2} \equiv y_r^{x_u r}$에서 $x_u r$은 오직 사용자만 알고, 원 공개키 y_u는 등록센터만 알기 때문이다.

▷ **일회용 대리서명의 안전성**

[정리1] 원 서명자만이 위임키를 생성할 수 있다.

$$s \equiv x_u r \cdot e + k$$

- 위임키에 대한 안전성은 이산대수 문제의 어려움에 기반하여 원 서명자의 비밀키를 모르면 생성할 수 없다.

[정리2] 원 대리서명자만이 대리서명을 할 수 있다.

$$s_1 \equiv s + x_{c_1}, \quad s_2 \equiv s + x_{c_2}, \quad s_3 \equiv s + x_{c_3}, \quad s_4 \equiv s + x_{c_4}$$

- 서명키의 안전성도 이산대수 문제의 어려움에 기반함으로 대리자의 비밀키를 모르면 키를 생성할 수 없으며, 서명도 불가능하다.

[정리3] 대리 서명자의 서명이 1회성임을 보장한다.

- 2번 이상 사용하면 아래와 같이 비밀키가 노출된다.

[표1] 일회용 대리서명임을 증명

(1) σ_1 증명

$$\sigma_1(msg + x_{c_1} - x_{c_2}) = s_1 + s_2 \quad ①$$

$$\sigma'_1(msg' + x_{c_1} - x_{c_2}) = s_1 + s_2 \quad ②$$

두 식 ①, ②는 s_1, s_2의 값으로 같다.

$$\sigma_1(msg + x_{c_1} - x_{c_2}) = \sigma'_1(msg' + x_{c_1} - x_{c_2}) \quad ③$$

∴ ③은 식 2개, 미지수 2개로 이것을 계산하면 대리자의 비밀키 x_{c_1}, x_{c_2}가 노출된다.

(2) σ_1 증명

$$\sigma_2(msg + x_{c_3} - x_{c_4}) = s_3 + s_4 \quad ④$$

$$\sigma'_2(msg' + x_{c_3} - x_{c_4}) = s_3 + s_4 \quad ⑤$$

두 식 ④, ⑤는 s_3, s_4의 값으로 같다.

$$\sigma_2(msg + x_{c_3} - x_{c_4}) = \sigma'_2(msg' + x_{c_3} - x_{c_4}) \quad ⑥$$

∴ ⑥은 식 2개, 미지수 2개로 이것을 계산하면 대리자의 비밀키 x_{c_3}, x_{c_4}가 노출된다.

대리 서명키를 2번 이상 사용한 경우: 대리 서명키 (s_1, s_2, s_3, s_4)와 클리어링하우스의 비밀키 노출!

[정리4] 서명단계는 실패-중단 서명기법의 안전성과 동일하다.

- 이산대수 문제의 어려움에 기반함으로 서명의 위조는 등록센터의 비밀키 x_r를 알아야 한다.

$$(g^{x_{c_1}} \cdot g^{-x_{c_2}} \cdot g^{msg})^{(\sigma_1-\tau_1)} = (y_r^{x_{c_3}} \cdot y_r^{-x_{c_4}} \cdot y_r^{msg})^{(\tau_2-\sigma_2)}$$
$$= (g^{x_{c_1}} \cdot g^{-x_{c_2}} \cdot g^{msg})^{x_r(\tau_2-\sigma_2)}$$

▶ 계산량 비교

대리서명은 원 서명자를 대신하여 대리자가 서명하는 것으로 전자상거래와 같은 전자시스템 수행시에 사용자의 계산량 부담을 줄여주는 장점을 가진다. 이것은 여러 논문 [5-7]에서 제시되었기에 계산량 비교는 생략한다. 따라서 라이센스의 서명/암호 알고리즘에 RSA서명/RSA암호, DSS서명/ElGamal암호, Schnorr서명/ElGamal암호의 사용을 가정하고, Y. Zheng의 Signcrytpion[4]의 Table2를 참고하여 표2와 같이 제안방식을 비교한다. 이때 지수연산(exp)에 비해 다른 연산(mul, add 등)들의 계산량은 상대적으로 낮기 때문에 지수연산만을 나타낸다.

[표2] 기존 서명과 암호 방식과의 계산량 비교

사용자의 계산량	컨텐츠 사용에 대한 서명			
	기존의 RSA 서명, DSS 서명, Schnorr 서명으로 직접서명을 수행하면, 사용자측의 시스템에 부담이 크다. 그러므로 제안 1회용 대리서명 방식으로 사용자측의 부담을 덜어준다.			
클리어링 하우스의 계산량	라이센스 서명 후 암호화 방식			제안 방식 (1회용 대리서명 + Signcryption)
	1회용 대리서명 + RSA 서명과 RSA 암호	1회용 대리서명 + DSS 서명과 ElGamal 암호	1회용 대리서명 + Schnorr 서명과 ElGamal 암호	
	Exp: 4번	Exp: 5번	Exp: 5번	Exp: 3번

제안 방식은 일회용 대리서명을 이용해서 사용자의 계산량을 줄이고, 상대적으로 증가한 클리어링하우스의 계산량은 Signcrytpion 기법을 적용하여 줄였다. 그리하여 전체적인 계산량이 표2에 제시한 서명과 암호를 사용하는 것에 비해 효율적임을 알 수 있다.

5. 결론

본 논문은 DRM의 라이센스 발급시 사용자의 계산량을 줄여 무선 DRM으로 확장 가능한 효율적인 라이센스 다운로드 방식을 제안하였다.

익명성을 갖는 일회용 대리서명[3]을 이용하여 사용자의 계산량을 감소시켰고 클리어링하우스가 악의를 가지고 위임키를 2번 이상 사용할 경우, 정리3에 의해 클리어링하우스의 비밀키가 노출되기 때문에 부정을 막을 수 있다. 또한 무선 환경은 유선 환경에 비해 도청자나 그 밖의 위조, 불법적 변경 등과 같은 위협들에 매우 취약하므로 사용자를 보호하는 익명성을 추가해 사용자가 안전하게 라이센스를 다운로드 받고 컨텐츠를 구입할 수 있다. 대리서명으로 증가한 클리어링하우스의 계산량은 Signcryption을 이용하여 감소시켰다.

[참고문헌]

[1] http://www.microsoft.com/windows/windowsmedia/drm.asp

[2] http://www.microsoft.com/windows/windowsmedia/wm7/drm/licensing.asp

[3] 김소진,최재귀,박지환 "익명성을 갖는 효율적인 1회용 대리서명", 한국정보처리학회, 추계학술발표논문집, 2002. 11.

[4] Y.Zheng, "Digital Signcryption or How to Achieve Cost(Signature & Encryption) << Cost(Signature) +Cost(Encryption)", CRYPTO '97, Springer-venlag, LNCS 1294, pp165-179, 1997

[5] S.J.Kim, S.J.Park and D.H.Won, "Nomiative Signatures", Procee dings of the 1995 Symposium on Cryptography and Information Security(SCIS'95), pp. B1.1.1~17, 4~27, Jan, 1995

[6] M.Mambo, K.Usuda and E.Okamoto, "Proxy Signatures for delegating signing operation", Proc. Third ACM Conference on Computer and Communications Security, pp. 48~57, 1996

[7] 박희운, 이임영, "이동통신에서 적용 가능한 수신자 지정 대리서명 방식", 한국정보보호학회 논문지 제10권 2호, pp.43-54, 2000.6