강한 Diffie-Hellman가정을 이용한 추이 서명 스킴

박태준, 홍도원

*한국전자통신연구원, 정보보호연구단

A Transitive Signature Scheme based on Strong Difffie-Hellman Assumption

Tae-Jun Park, Dowon Hong

*Information Security Research Division, ETRI

요 약

추이 서명 스킴은 edge에 대한 서명을 연결하여 경로에 대한 서명을 만드는 서명 스킴이다. 본 논문은 강한 Diffie-Hellman 가정을 이용하여 랜덤 오라클을 가정하지 않은 추이서명 스킴을 제안한다

I. 서론

전자서명프로토콜은 1976년 Diffie와 Hellman [1]에 의해 제시된 이후 많은 발전을 해왔다. 최근, Rivest[2](2000년)에 의해 추이서명에 대한 개념이 생겨났고 그 이후 Micali와 Rivest[3] (2002년)에 의해 최초의 추이서명 스킴이 개발 되었다. 같은 해 Bellare와 Neven[4](2002)은 RSA 가정을 이용한 추이서명 스킴이 개발 되었다. 이 추이서명 스킴들은 방향성이 없는 추이서명이며, 방향성이 있는 추이서명에 대한 연구는 2003년 Hohenberger[5]에 의해 이루어 졌다

Hohenberger는 방향성이 있는 추이서명 스킴을 만드는 데는 실패했지만 스킴이 존재하기 위한 필요충분조건에 대한 연구를 하는 도중 Pseudo-freeness라는 개념을 발견하여 이후 많은 연구가 진행 중이다.

전자서명의 효율성에서 서명의 길이는 매우 중요한 요소이다. 서명의 길이를 줄이기 위해 여러 가지 암호학적 가정을 이용하는 데 2001년 Boneh, Lynn, Shacham[6]은 Computational Diffie-Hellman가정을 이용하여 DSA 서명보다 절반 정도 짧으면서 같은 정도의 안전성을 지니는 전자서명 스킴을 제안했다. 이 스킴은 랜덤 오라클(random oracle)모델에서의 능동 선택 평문 공격에 대한 존재 위조 불가(existential unforgeable under the adaptive chosen message attack)에 따른 안전성 증명을 갖고 있다. 2004년에 Boneh와 Boyen[7]은 강한 Diffie-Hellman 가정을 이용하여 짧은 서명을 개발하였고 이 서명 스킴은 랜덤 오라클 모델을 사용하지 않으면서 능동 선택 평문 공격에 대한 존재 위조 불가에 따른 안전성 증명을 갖고 있다.

Zhu, Bao, Deng[8]은 무선 네트워크에서 개개의 모바일 디바이스를 네트워크 그래프의 노드로 간주하여 추이서명 스킴을 적용, 트러스트를 계산하는 방법을 제시하였다.

본 논문에서는 Boneh와 Boyen이 제안한 강한 Diffie-Hellman 가정을 이용한 짧은 서명 스킴을 이용하여 능동 선택 평문 공격에 대한 존재 위조 불가에 따른 안전성 증명을 갖는 추이서명 스킴을 제시한다.

II. 추이서명의 정확성과 안전성

추이서명이란, 비밀키 tsk와 공개키 tpk를 갖고 있는 서명자가 노드 쌍 $\{i,j\}$에 대한 서명을 언제든지 만들어 내어 그래프에 edge $\{i,j\}$를 추가할 수 있는 서명이며, 덧붙여 연결 성질(composability property)을 만족해야한다. 여기서 연결 성질이란, edge $\{i,j\}$에 대한 서명과 edge

$\{j,k\}$에 대한 서명이 주어졌을 때, 공개키를 가진 사람이 edge $\{i,k\}$에 대한 서명을 만들어 낼 수 있는 것을 말한다.

추이서명의 안전성이란, 연결성질에 의해 만들어지는 새로운 edge에 대한 서명이외에 edge에 대한 위조가 불가능한 것이며, 다시 말해 노드 i와 j가 서명자에 의해 서명이 된 경로(path)를 통하여 이어진 경우를 제외하고, 비밀키 없이 edge $\{i,j\}$에 대한 유효한 서명을 만들지 못함을 뜻한다.

그래프 $G=(V,E)$는 node 집합 V와 edge 집합 $E\subseteq V\times V$로 구성되며, node i에서 node j로 가는 edge는 순서쌍 (i,j)로 표기하는 데 본 논문에서는 방향이 없는 그래프만 다루므로 $\{i,j\}$로 표기한다.

그래프 $G=(V,E)$에 대해, 모든 노드 $i,j\in V$가 경로 $\{i,j\}\in\tilde{E}$ 에 의해 연결되어 있을 때, 그래프 $\overline{G}=(V,\overline{E})$를 $G=(V,E)$의 추이 closure라고 부른다. 또, $\{i,j\},\{j,k\}\in E^*$인 임의의 노드 $i,j,k\in V^*$에 대해 $\{i,k\}\in E^*$일 때 $G^*=(V^*,E^*)$를 추이적으로 닫혀 있다고 한다.

추이서명스킴 TS=(**TKG,TSign,TVf,Comp**)는 다음과 같이 4개의 polynomial-time 알고리즘으로 구성되어 있다.

• 랜덤화된 키 생성 알고리즘 **TKG** : **TKG**는 1^k ($k\in\mathbb{N}$는 security parameter)를 input으로 받아서 (tpk,tsk)을 돌려준다. 여기서 tpk는 공개키이고, tsk는 이에 대응하는 비밀키다.

• 서명 알고리즘 **TSign** : **TSign**은 비밀키 tsk와 노드 $i,j\in\mathbb{N}$을 받아 edge $\{i,j\}$에 대한 서명을 돌려준다.

• 결정화된 확인 알고리즘 **TVf** : **TVf**은 공개키 tpk와 노드 $i,j\in\mathbb{N}$ 그리고 서명 σ을 받아 0 또는 1을 돌려준다.

• 결정화된 연결 알고리즘 **Comp** : **Comp**은 공개키 tpk와 노드 $i,j,k\in\mathbb{N}$ 그리고 서명 σ_1,σ_2을 받아 새로운 서명 σ 또는 실패를 의미하는 $\perp$을 돌려준다.

추이서명에 있어서 정확성이란, 연결 알고리즘에 의해 만들어진 edge의 서명과 서명자가 만들어낸 서명이 일치해야 한다는 것을 의미한다. 추이서명 스킴이 stateful일 경우 정확성에 대한 정의가 불명확하다. 왜냐하면 연결 알고리즘에 의해 만들어진 서명과 서명자에 의한 서명이 다른 시간대에 다른 state에서 만들어 질 수 있기 때문이다. 따라서 정확성에 대한 정의는 서명 알고리즘의 statefulness를 고려하여 재정립되어야 한다.

Bellare와 Neven[4]은 이를 위하여 **TSign**과 **Comp** 알고리즘을 오라클로 하는 그림 1의 실험을 수행하는 알고리즘 A를 생각하였다.

$(tpk,tsk)\xleftarrow{\$}$ **TKG**(1^k)

$S\leftarrow 0$; ***Legit***$\leftarrow$**true** ; ***NotOK***$\leftarrow$**false**

Run **A** with its oracles until it halts, replying to its oracle queries as follows;

If **A** makes **TSign** query i,j then

  If $i=j$ then ***Legit*** $\leftarrow$ **false**

  Else

    Let σ be the output of the **TSign** oracle and let $S\leftarrow S\cup\{(\{i,j\},\sigma)\}$

    If **TVf**$(tpk,i,j,\sigma)=0$

      then ***NotOK***$\leftarrow$**true**

If **A** makes **Comp** query i,j,k,σ_1,σ_2 then

  If $[(\{i,j\},\sigma_1)\notin S$ or $(\{j,k\},\sigma_2)\notin S$ or i,j,k are not all distinct]

    then ***Legit***$\leftarrow$**false**

  Else

    Let σ be the output of the **Comp** oracle and let $S\leftarrow S\cup\{(\{i,k\},\sigma)\}$

    Let $\tau\leftarrow$**TSign**(tsk,i,k)

    If $[(\sigma\neq\tau)$ or **TVf**$(tpk,i,k,\sigma)=0]$

      then ***NotOk***$\leftarrow$**true**

When A halts, output $(\textit{Legit}\wedge\textit{NotOK})$ and halt

그림 1

$(\textit{Legit}\wedge\textit{NotOK})$=**true**이려면 ***Legit***와 ***NotOK***가 동시에 **true**이어야 하는 데 이런 경우는 **A**가 합법적인 쿼리를 하면서, 정확성을 위배해야 하는 상황을 의미한다. 따라서 정확성의 정의는 다음과 같이 내려진다.

정의 1] 어떠한 알고리즘 **A**와 모든 $k\in\mathbb{N}$에 대해, 그림 1의 experiment의 결과가 **true**일 확률이 0일 때, 추이 서명 스킴 TS를 **정확**하다고 한다.

추이서명에서의 안전성은 그래프$G=(V,E)$의 추이 closure에 있지 않은 edge에 대한 유효한 서명을 만들어 내지 못하는 것을 말한다.

정형화된 안전성은 다음과 같다(MR[]). 추이서명 스킴 TS=(**TKG,TSign,TVf,Comp**)과 알고리즘 F(tu-cma 공격자) 그리고 시큐러티 파라미터 $k \in \mathbb{N}$에 대해, F가 공격에 성공하면 1을 리턴하는 실험 $\mathrm{Exp}_{TS,F}^{tu-cma}$을 생각한다. 이 실험에서 우선 **TKG**를 이용하여 (tpk, tsk)을 얻어낸다. tpk와 $\mathrm{TSign}(tsk, \cdot, \cdot)$에 대한 오라클접근을 통해 공격자 F를 가동시킨다. F가 오라클 쿼리 i,j를 한 edge $\{i,j\}$를 모두 모은 집합을 E라고 하고 이와 연관된 노드의 집합을 V라고 한다. 그래프 $G=(V,E)$의 추이 closure안에 없는 edge $\{i,j\}$의 tpk에 대해 유효한 서명 σ'을 위조해 낼 수 있을 때 F가 이겼다고 말한다. $\mathrm{Exp}_{TS,F}^{tu-cma}$은 F가 이기면 1을 리턴하고 아니면 0을 리턴한다. F의 어드밴티지는 다음과 같다. (단, $k \in \mathbb{N}$)

$$ADV_{TS,F}^{tu-cma}(k) = \Pr[Exp_{TS,F}^{tu-cma}(k) = 1]$$

$ADV_{TS,F}^{tu-cma}(\cdot)$이 가동시간이 시큐러티 파라미터 k에 대한 다항식인 모든 공격자 F에 대해 무시할 수 있을 정도로 작을 때, TS가 능동 선택평문 공격에 대한 존재 위조 불가(existential unforgeable under the adaptive chosen message attack)하다고 말한다.

III. 강한 DH가정을 이용한 추이서명 스킴

강한 Diffie-Hellman가정은 2004년 Boneh와 Boyen[7]이 제안했다. 랜덤 오라클을 가정하지 않고 프로토콜의 증명가능 안전성을 증명할 수 있기 때문에 중요하며, 유사한 가정으로 RSA에서는 강한 RSA 가정이 있다.

강한 Diffie-Hellman 가정은 다음과 같다. G_1, G_2을 원소의 개수가 소수 p개인 순환군 이라 하고 $G_1, = G_2$일수도 있다. g_1은 G_1의 생성자이고 g_2은 G_2의 생성자이며, $g_1 = \psi(g_2)$의 관계를 만족한다. 이 때 q-SDH 문제는 다음과 같다. 주어진 $q+2$ tuple $(g_1, g_2, g_2^x, g_2^{(x^2)}, ..., g_2^{(x^q)})$에 대해, $c \in \mathbb{Z}_p^*$인 순서쌍$(c, g_1^{1/(x+c)})$을 찾아내는 문제이다. 랜덤 선택한 g_2와 $x \in \mathbb{Z}_p^*$ 그리고 알고리즘 A에 의한 랜덤 비트에 대해

$$\Pr[A(g_1, g_2, g_2^x, g_2^{(x^2)}, ..., g_2^{(x^q)}) = (c, g_1^{\frac{1}{x+c}})] \geq \epsilon$$

일 때, 알고리즘 A는 (G_1, G_2)에서 q-SDH를 풀기 위한 ϵ만큼의 어드밴티지를 갖는다고 말한다.

정의] (G_1, G_2)에서 q-SDH를 풀기 위한 ϵ만큼의 어드밴티지를 갖는 t 시간 알고리즘이 존재하지 않을 때, (q,t,ϵ)-SDH 가정이 (G_1, G_2)에서 성립한다.

강한 DH 가정을 이용하여 아래와 같은 추이서명 스킴을 구성할 수 있다.

- weakly secure 추이서명 스킴

• 키 생성 알고리즘 **TKG**(1^k) :

1) **SKG**(1^k)을 가동시켜 SDS를 위한 키 쌍 (spk, ssk)를 만들어냄
2) K_{cg}를 통해 (G_1, G_2, g_2, p)을 만들어 낸 후 $g_1 \leftarrow \psi(g_2)$을 계산함. $x \xleftarrow{R} \mathbb{Z}_p^*$을 선택한 후 $v \leftarrow g_2^x$을 계산한다.
3) 공 개 키 tpk는 $(\langle G_1\rangle, \langle G_2\rangle, g_1, g_2, v, spk)$이고 비밀키 tsk는 (x, ssk)이다.

• 추이 서명 알고리즘 **TSign** :

state (V, l, L, Σ)을 유지한다. 여기서 $V \subseteq \mathbb{N}$은 쿼리된 모든 노드의 집합이고, l, L은 V에서 G_1으로의 함수이며, $\Sigma : V \rightarrow \{0,1\}^*$는 노드 i에 대해 $i \| L(i)$을 ssk을 이용하여 서명하는 함수이다.

1) If $i > j$ then swap(i,j)
2) If $i \notin V$ then
   $V \leftarrow V \cup \{i\}$
   $L(i) \xleftarrow{R} \mathbb{Z}_p : l(i) \leftarrow g_1^{\frac{1}{x+L(i)}}$
   $\Sigma(i) \leftarrow SSign(ssk, i \| L(i))$
3) If $j \notin V$ then
   $V \leftarrow V \cup \{j\}$
   $L(j) \xleftarrow{R} \mathbb{Z}_p : l(j) \leftarrow g_1^{\frac{1}{x+L(j)}}$
   $\Sigma(j) \leftarrow SSign(ssk, j \| L(j))$
4) $\delta \leftarrow l(i) l(j)^{-1}$
5) 노드에 대한 증명서로서
   $C_i \leftarrow (i, L(i), \Sigma(i))$;
   $C_j \leftarrow (j, L(j), \Sigma(j))$
   을 발급한다.
6) $\sigma = (C_i, C_j, \delta)$을 edge $\{i,j\}$에 대한 서명으로 리턴한다.

• 추이서명에 대한 확인 알고리즘 **TVf** :
input은 tpk이며,
1) if $i > j$ then swap(i,j)
2) if $SVf(spk, i\|L_i, \Sigma_i) = 0$ or
$SVf(spk, j\|L_j, \Sigma_j) = 0$
then return 0
3) if $e(\delta, g_2^{L_i \cdot L_j} \cdot X^{L_i+L_j} \cdot u)^{\frac{1}{-L_i+L_j}} = e(g_1, g_2)$, then return 1 else return 0

• 연결 알고리즘 **Comp** :
input은 $tpk, i, j, k, \sigma_1, \sigma_2$이며,
1) if $i > k$ then swap(i,k); swap(σ_1, σ_2)
2) parse σ_1 as (C_1, C_2, δ_1); parse σ_2 as (C_3, C_4, δ_2)
3) if $i > j$, then swap(C_1, C_2); $\delta_1 \leftarrow \delta_1^{-1}$
4) if $j > k$, then swap(C_3, C_4); $\delta_2 \leftarrow \delta_2^{-1}$
5) $\delta \leftarrow \delta_1 \cdot \delta_2$
Return (C_1, C_4, δ)

2. Strongly Secure 추이서명 스킴

• **TKG**(1^k) :
1) **SKG**(1^k)을 가동시켜 SDS를 위한 키 쌍 (spk, ssk)를 만들어냄
2) K_{cg}를 통해 (G_1, G_2, g_2, p)을 만들어 낸 후 $g_1 \leftarrow \psi(g_2)$을 계산함. 랜덤하게 $x, y \xleftarrow{R} \mathbb{Z}_p^*$을 선택한 후 $u \leftarrow g_2^x \in \mathbb{G}_2$ 와 $v \leftarrow g_2^y \in \mathbb{G}_2$ 을 계산.
3) 공개키 tpk는 $(\langle G_1\rangle, \langle G_2\rangle, g_1, g_2, u, v, spk)$이고 비밀키 tsk는 (x, y, ssk)이다.

• **TSign** : state (V, l, L, Σ)을 유지한다. 여기서 $V \subseteq \mathbb{N}$은 쿼리된 모든 노드의 집합이고, l, L은 V에서 G_1으로의 함수이며, $\Sigma : V \rightarrow \{0,1\}^*$는 노드 i에 대해 $i\|L(i)\|g_1^{r_i}$을 ssk을 이용하여 서명하는 함수이다.
1) If $i > j$ then swap(i,j)
2) If $i \notin V$ then
$V \leftarrow V \cup \{i\}$
$L(i), r(i) \xleftarrow{R} \mathbb{Z}_p$:
$l(i) \leftarrow g_1^{\frac{1}{x+L(i)+r(i)y}}$
$\Sigma(i) \leftarrow SSign(ssk, i\|L(i))$
3) If $j \notin V$ then
$V \leftarrow V \cup \{j\}$
$L(j), r(j) \xleftarrow{R} \mathbb{Z}_p$:
$l(i) \leftarrow g_1^{\frac{1}{x+L(j)+r(j)y}}$
$\Sigma(j) \leftarrow SSign(ssk, j\|L(j))$
4) $\delta \leftarrow l(i)l(j)^{-1}$
5) 노드에 대한 증명서로서
$C_i \leftarrow (i, L(i), \Sigma(i))$;
$C_j \leftarrow (j, L(j), \Sigma(j))$
을 발급한다.
6) $\sigma = (C_i, C_j, \delta)$을 edge $\{i,j\}$에 대한 서명으로 리턴한다.

• **TVf** : input은 tpk이며,
1) if $i > j$ then swap(i,j)
2) if $SVf(spk, i\|L_i\|L_i', \Sigma_i) = 0$ or
$SVf(spk, j\|L_j\|L_j', \Sigma_j) = 0$
then return 0
3) if

$$\left(\frac{e(\delta, u \cdot X^{L_i+L_j} \cdot v^{r_i+r_j} \cdot Y^{L_ir_j+L_jr_i} \cdot w^{r_ir_j} g_2^{L_iL_j})}{e(Y^{-r_i+r_j}, g_2)}\right)^{\frac{1}{-L_i+L_j}} = e(g_1, g_2)$$

then return 1, else return 0

• **Comp** : input은 $tpk, i, j, k, \sigma_1, \sigma_2$이며,
1) if $i > k$ then swap(i,k); swap(σ_1, σ_2)
2) parse σ_1 as (C_1, C_2, δ_1); parse σ_2 as (C_3, C_4, δ_2)
3) if $i > j$, then swap(C_1, C_2); $\delta_1 \leftarrow \delta_1^{-1}$
4) if $j > k$, then swap(C_3, C_4); $\delta_2 \leftarrow \delta_2^{-1}$
5) $\delta \leftarrow \delta_1 \cdot \delta_2$
Return (C_1, C_4, δ)

IV. 안전성 및 결론

본 논문은 Boneh와 Boyen[7]의 강한 DH 가정을 이용한 추이서명 스킴을 제시하였다. 안전성 분석은 강한 DH가정만으로 증명이 불가능하며 일반화된 강한 DH가정이 필요하다. 이에 대한 증

명은 다음 논문에 제시할 것이다. 이 추이서명 스킴은 랜덤 오라클을 가정하지 않고 안전성 증명이 가능하며 능동 선택 평문 공격에 대한 존재 위조 불가에 따르는 안전성을 갖고 있다.

추이서명의 실용적 응용으로 분산 컴퓨팅 환경을 생각할 수 있다. 원하는 서버에 접근하는 경로(path)를 검색한 후에 경로를 구성하는 edge들에 대한 서명을 결합하여 전체 경로에 대한 서명을 구성할 수가 있고 이를 통해 원하는 서버에 접근할 수 있도록 할 수 있다.

[참고문헌]

[1] W. Diffie and M.E. Hellman, "New directions in cryptography", IEEE Transactions on Information Theory 22, pp. 644-654, 1976

[2] R. Rivest, "Two new signature schemes", Slides from talk given at Cambridge University, 2000

[3] S. Micali, R. Rivest, "Transitive signature schemes", CT-RSA 2002, LNCS 2271, pp. 236-243, 2002

[4] M. Bellare, G. Neven, "Transitive signature based on factoring and RSA", ASIACRYPT 2002, LNCS 2501, pp. 297-414, 2002

[5] S. Hohenberger, "The cryptographic impact of groups with infeasible inversion", S.M. thesis, MIT, 2003

[6] D. Boneh, B. Lynn, H. Shacham, "Short signatures from the Weil pairing", ASIACRYPT 2001, LNCS 2248, pp. 514-532, 2001

[7] D. Boneh, X. Boyen, "Short signatures without random oracles", EUROCRYPT 2004, LNCS 3027, pp. 56-73, 2004

[8] H. Zhu, F. Bao, R.H. Deng, "Computing of turst in wireless networks", Vehicular Technology Conference VTC 2004-Fal, 2004 IEEE 60th Volume 4, pp.26-29, 2004