# 보다 효율적인 Hierarchical ID-Based Cryptosystem

김태구, 염대현, 이필중

포항공과대학교, 전자전기공학과

{tgkim, dhyum, pjl}@postech.ac.kr

# More Efficient Hierarchical ID-Based Cryptosystem

Tae Gu Kim, Dae Hyun Yum, Pil Joong Lee

Department of Electronic and Electrical Engineering, POSTECH

## 요 약

C. Gentry와 A. Silverberg의 Hierarchical ID-Based Cryptography[5]에서는 Hierarchy의 깊이에 따라 암호문 길이가 선형적으로 증가한다. 본 논문에서는 Hierarchical ID-based Signature(HIDS)의 서명을 활용해 암호문의 길이를 일정하게 만드는 방법을 제안한다.

## *I.* Introduction

ID-Based cryptosystem은 1984년 A. Shamir에 의해 제안되었다[1]. 송신자는 수신자의 ID로부터 암호키를 생성한 후 평문을 암호화한다. 수신자는 받은 암호문을 PKG (Private Key Generator)에서 받은 복호키를 사용해서 복호화한다. 이는 기존의 PKI (Public Key Infrastructure)가 가지는 인증서를 사용하지 않고 사용자들이 이미 가지고 있는 ID (e-mail주소 등)를 사용한다는 장점을 가지고 있다. Shamir는 논문에서 signature scheme은 제시했으나 encryption scheme은 구체적으로 제시하지 않았다. 그 후 D. Boneh가 2001년에 Weil pairing을 사용하는 ID-based encryption[2]과 서명의 길이가 짧은 signature scheme[3]을 제안하였다. 최근에는 2단계의 Hierarchical ID-based encryption[4]이 발표되었고 이를 임의의 n단계까지 확장한 Hierarchical ID-based cryptography[5]가 제안되었다. 그러나 [5]의 방법은 Hierarchy의 깊이에 따라 암호문과 서명의 길이가 선형적으로 길어진다는 단점이 있다. 본 논문에서는 이러한 단점을 서명을 통해 해결하는 보다 효율적인 방법을 제안한다.

## *II.* HIDC (Hierarchical ID-based Cryptography)

### *1.* 정 의

1) Admissible pairing

다음과 같은 특성을 가지는 $G_1 \times G_1$에서 $G_2$로의 함수 $e$을 admissible pairing이라고 한다.

□ Bilinear: $\hat{e}(aQ, bR) = \hat{e}(Q, R)^{ab}$, $Q, R \in G_1$, $a, b \in Z$

□ Non-degenerate: $e$는 $G_1 \times G_1$의 모든 원소를 $G_2$의 항등원으로 보내지는 않는다.

□ Computable: 임의의 $Q, R \in G_1$에 대한 $\hat{e}(Q, R)$값을 계산할 수 있는 효율적인 algorithm이 존재한다.

2) Bilinear Diffie-Hellman (BDH) Assumption

$G_1$과 $G_2$를 큰 소수 위수 $q$를 가지는 가환군이라고 하자. $G_1$의 원소 $P$를 임의로 선택하고 $Z_q$의 원소 $a$, $b$, $c$를 임의로 선택해서 $aP$, $bP$, $cP$가 주어졌을 때, $\hat{e}(P, P)^{abc}$를 계산하는 것이 어렵다.

### *2.* HIDE (Hierarchical ID-based Encryption) Scheme

1) BasicHIDE

가. Root Setup

㉠. BDH parameter generator $lg$는 security

parameter K를 입력으로 받아 $G_1$, $G_2$, $e$를 생성.

㉡. 임의의 생성원 $P_0$를 $G_1$에서 선택.

㉢. 임의의 $s_0 \square Z_q$를 선택 후 $Q_0 = s_0 P_0$를 계산.

㉣. 다음과 같은 해쉬 함수$H_1$, $H_2$를 선택.

$$H_1 : \frac{1}{2}0,1\frac{3}{4}^* \rightarrow G_1, \quad H_2 : G_2 \rightarrow \frac{1}{2}0,1\frac{3}{4}^n$$

나. Lower-level Setup

$Level_i$를 $i$ 단계에 있는 entity들의 집합이라고 할 때($Level_0 = \frac{1}{2}\ RootPKG\ \frac{3}{4}$), entity $E_t \square\ Level_t$는 $s_t$를 $Z_q$에서 임의로 선택해서 그것을 안전하게 보관한다.

다. Extraction

$E_t$를 ID-tuple $(ID_1, ..., ID_t)$를 가지는 $Level_t$의 entity라고 하자. 이때 $(ID_1, ..., ID_i)$ $(1 \le i < t)$는 $E_t$의 조상들의 ID-tuple이다. 그러면 $E_t$의 부모는 다음과 같은 작업을 수행한다.

㉠. $P_t = H_1(ID_1, ..., ID_t)$을 계산.

㉡. $E_t$의 secret point $S_t$를 다음과 같이 계산.

$$S_t = S_{t-1} + s_{t-1} P_t \quad (S_0\text{는 } G_1\text{의 항등원})$$

㉢. $S_t$와 $Q_i = s_i P_0$ $(1 \le i < t)$들을 $E_t$에게 전달.

라. Encryption

$E_t$에게 평문 $M$을 암호화해서 보내기 위해서는 다음과 같은 작업을 수행한다.

㉠. $P_i = H_1(ID_1, ..., ID_i)$ $(1 \le i \le t)$들을 계산.

㉡. 임의의 $r \square Z_q$을 선택.

㉢. 다음과 같은 암호문 전송.

$$C = [rP_0, rP_2, ..., rP_t, M \square H_2(g^r)] \qquad (1)$$

where $g = \hat{e}(Q_0, P_1)$

마. Decryption

받은 암호문을 $C = [U_0, U_2, ..., U_t, V]$라고 하면, $E_t$는 다음과 같이 복호화한다.

$$V \square H_2\left(\frac{\hat{e}(U_0, S_t)}{\prod_{i=2}^{t} \hat{e}(Q_{i-1}, U_i)}\right) = M$$

## 3. HIDS (Hierarchical ID-based Signature) Scheme

가. Root Setup

㉠. $Ig$는 K를 입력으로 받아 $G_1$, $G_2$, $e$를 생성.

㉡. 임의의 생성원 $P_0$를 $G_1$에서 선택.

㉢. 임의의 $s_0 \square Z_q$를 선택 후 $Q_0 = s_0 P_0$를 계산.

㉣. 다음과 같은 해쉬 함수$H_1$, $H_3$를 선택.

$$H_1 : \frac{1}{2}0,1\frac{3}{4}^* \rightarrow G_1, \quad H_3 : \frac{1}{2}0,1\frac{3}{4}^* \rightarrow G_1$$

나. Lower-level Setup: BasicHIDE와 동일

다. Extraction: BasicHIDE와 동일

라. Signing

㉠. $P_M = H_3(ID_1, ..., ID_t, M)$을 계산.

㉡. $Sig = S_t + s_t P_M$을 계산.

㉢. $Sig$와 $Q_i = s_i P_0$ $(1 \le i \le t)$들을 전송.

마. Verification

받은 서명 $[Sig, Q_1, ..., Q_t]$에 대해 다음과정을 수행해서 검증한다.

$$\hat{e}(P_0, Sig) = \hat{e}(Q_0, P_1)\hat{e}(Q_t, P_M) \prod_{i=2}^{t} \hat{e}(Q_{i-1}, P_i)$$

## 4. Shortening the Ciphertext and Signatures

1) Authenticated Lower-level Root PKGs

Entity $E_y$가 CSU라는 대학에 있는 사람에게 자주 메일을 보낸다고 가정하면, CSU를 authenticated lower-level root PKG로 여기고 이 CSU의 서명을 사용해서 암호문의 길이를 줄이는 방법이다. 이를 위해 root PKG는 추가적으로 임의의 평문 $M^*$를 parameter로 가진다. $S_t$를 secret point로 가지는 CSU는 authenticated root

PKG의 set up 과정으로 $M*$에 서명한다.

다음과 같은 ID-tuple $(ID_1, ..., ID_v, ..., ID_v)$을 가지는 CSU의 $E_z$라는 entity에게 $E_y$는 CSU의 서명 $[Sig(=S_t + s_t P_{M*}), Q_1, ..., Q_t]$에서 $Sig$와 $Q_t$를 사용해서 다음과 같은 암호화 과정을 수행한다.

가. Encryption

㉠. $P_i = H_1(ID_1, ..., ID_i)$ $(t+1 \le i \le z)$을 계산.

㉡. 임의의 $r \square Z_q$을 선택.

㉢. 다음과 같은 암호문 전송.

$$C = [rP_0, rP_{t+1}, ..., rP_z, M \square H_2(g_t^r)]$$

$$\text{where } g_t = \frac{\hat{e}(P_0, Sig)}{\hat{e}(Q_t, P_{M0})} = \hat{e}(P_0, S_t)$$

나. Decryption

받은 암호문을 $C = [U_0, U_{t+1}, ..., U_z, V]$라고 하면, $E_z$는 다음과 같이 복호화한다.

$$V \square H_2\left(\frac{\hat{e}(U_0, S_z)}{\prod_{i=t+1}^{z} \hat{e}(Q_{i-1}, U_i)}\right) = M$$

## *III.* Our result

II장에서 보였던 것과 같이 HIDE의 가장 큰 문제점은 암호문의 길이가 식(1)처럼 Hierarchy의 깊이에 따라 선형적으로 늘어난다는 것이다. 이의 개선을 위해 II.4와 같은 방법을 사용하는데, 여기서는 받은 서명의 $Q_t$만 사용했다. 그러나 나머지 유효한 $Q_i$들을 사용하면 $Level_t$위의 단계에 있는 entity들에게 암호문을 일정한 길이로 보낼 수 있다.

송신자 Alice가 다음과 같은 ID-tuple $(ID_1, ..., ID_t)$을 가지는 entity $E_t$의 서명 $[Sig, Q_1, ..., Q_t]$을 얻었다면, HIDS의 검증과정을 통해 $Q_1, ..., Q_t$값이 유효하다는 것을 알 수 있다. Alice는 $Level_t$보다 위의 $Level_v (v < t)$에 있고 $E_t$의 $ID_1, ..., ID_v$와 같은 ID-tuple $(ID_1, ..., ID_v)$을 가지는 entity $E_v$와 다음과 같은 과정을 거쳐 암호화된 통신을 할 수 있다.

가. Encryption

㉠. 임의의 $r \square Z_q$을 선택.

㉡. 다음과 같은 암호문 전송.

$$C = [rP_0, M \square H_2(g_v^r)]$$

$$\text{where } g_v = \hat{e}(Q_0, P_1)\hat{e}(Q_1, P_2) \square \hat{e}(Q_{v-1}, P_v)$$

나. Decryption

받은 암호문을 $C = [U, V]$라고 하면, $E_v$는 다음과 같이 복호화한다.

$$V \square H_2(\hat{e}(U, S_v)) = M$$

여기서 $g_v^r$은 다음과 같이 계산할 수 있다.

$$\begin{aligned} g_v^r &= (\hat{e}(Q_0, P_1)\hat{e}(Q_1, P_2) \square \hat{e}(Q_{v-1}, P_v))^r \\ &= \hat{e}(Q_0, rP_1)\hat{e}(Q_1, rP_2) \square \hat{e}(Q_{v-1}, rP_v) \\ &= \hat{e}(s_0 P_0, rP_1)\hat{e}(s_1 P_0, rP_2) \square \hat{e}(s_{v-1} P_0, rP_v) \\ &= \hat{e}(rP_0, s_0 P_1)\hat{e}(rP_0, s_1 P_2) \square \hat{e}(rP_0, s_{v-1} P_v) \\ &= \hat{e}(rP_0, s_0 P_1 + s_1 P_2 + \square + s_{v-1} P_v) \\ &= \hat{e}(rP_0, S_v) \end{aligned}$$

그러므로 secret point $S_v$를 가진 $E_v$는 $rP_0$만 알면 Alice가 계산한 $g_v^r$을 얻을 수 있게 된다. $S_v$나 $r$을 모르는 경우 $g_v^r$을 구하는 것은 BDH문제로 볼 수 있고 이는 어려운 문제로 알려져 있다.

이렇게 하면, 암호문이 Hierarchy의 깊이에 따라 선형적으로 늘어나지 않는다. 다만 $E_t$의 서명을 얻어야 하는데, 이는 HIDC의 선택사항으로 임의의 $M*$에 대한 서명을 요구함으로써 해결할 수 있다. 더 하위단계에 있는 entity의 서명을 얻을수록 위와 같은 암호문을 보낼 수 있는 경우는 많아진다. 이와 같은 경우는 실제적으로 쓰일 때 자주 발생하기 때문에 본 논문에서 제안한 방법은 실제적으로 유용하게 사용될 수 있다.

또한 암호문을 보내고자 하는 entity의 모든 $Q_i$들은 모르지만 알고 있는 $Q_i$들을 사용해서 보내는 암호문의 길이를 줄일 수 있다.

$Level_w$에 있으면서 $E_t$와 같은 $ID_1, ..., ID_v$를 가지고 다음과 같은 ID-tuple $(ID_1, ..., ID_v, ..., ID_w)$을 가지는 entity $E_w$가 있다고 하자. 유효한 $Q_i$ $(1 \le i \le v, v < t)$들을 이미 가지고 있으므로 다음과 같은 과정을 거쳐 암호화된 통신을 할 수

있다.

가. Encryption

㉠. 임의의 $r \square Z_q$을 선택.

㉡. 다음과 같은 암호문 전송.

$$C = [\, rP_0, rP_{v+2}, \ldots, rP_w, M \square H_2(g_w^r)\,]$$

where

$$g_w = \hat{e}(Q_0, P_1)\hat{e}(Q_1, P_2) \square \hat{e}(Q_v, P_{v+1})$$

나. Decryption

받은 암호문을 $C = [\, U_0, U_{v+2}, \ldots, U_w, V\,]$라고 하면, $E_w$는 다음과 같이 복호화한다.

$$V \square H_2\left(\frac{\hat{e}(U_0, S_w)}{\prod_{i=v+2}^{w} \hat{e}(Q_{i-1}, U_i)}\right) = M$$

여기서 $g_w^r$은 다음과 같이 계산할 수 있다.

$$g_w^r = (\hat{e}(Q_0, P_1)\hat{e}(Q_1, P_2) \square \hat{e}(Q_v, P_{v+1}))^r$$

$$= \hat{e}(s_0 P_0, rP_1)\hat{e}(s_1 P_0, rP_2) \square \hat{e}(s_v P_0, rP_{v+1})$$

$$= \frac{\hat{e}(s_0 P_0, rP_1) \square \hat{e}(s_{w-1} P_0, rP_w)}{\hat{e}(s_{v+1} P_0, rP_{v+2}) \square \hat{e}(s_{w-1} P_0, rP_w)}$$

$$= \frac{\hat{e}(rP_0, S_w)}{\prod_{i=v+2}^{w} \hat{e}(Q_{i-1}, U_i)}$$

## *IV*. Conclusion

HIDE의 가장 큰 문제는 암호문의 길이가 Hierarchy의 깊이에 따라서 선형적으로 길어진다는 것이었다. 이 문제를 개선하기 위해 Authenticated lower-level root PKG와 같은 방법을 사용했는데 여기서는 서명에서 $Q_i$만 사용하여 그 밑 *Level*의 암호문의 길이를 줄였다. 본 논문에서는 서명의 나머지 유효한 $Q_i$값들을 사용하여 그 위 *Level*의 암호문을 일정하게 하는 방법과 그 위 *Level*이 아니더라도 같은 $Q_i$값들을 활용하여 암호문의 길이를 줄이는 방법을 제안하였다.

## 참고문헌

[1] A. Shamir, "Identity-based cryptosystems and signature schemes," Advances in Cryptology - Crypto '84, Lecture Notes in Computer Science 196, Springer, pp. 47-53, 1984.

[2] D. Boneh and M. Franklin, "Identity based encryption from the Weil pairing," Advances in Cryptology - Crypto 2001, Lecture Notes in Computer Science 2139, Springer, pp. 213-229, 2001.

[3] D. Boneh, B. Lynn and H. Shacham, "Short signatures from the Weil pairing," Advances in Cryptology - Asiacrypt 2001, Lecture Notes in Computer Science 2248, Springer, pp. 514-532, 2001.

[4] J. Horwitz and B. Lynn, "Toward hierarchical Identity-based encryption," Advances in Cryptology - Eurocrypt 2002, Lecture Notes in Computer Science 2332, Springer, pp. 466-481, 2002.

[5] C. Gentry and A. Silverberg, "Hierarchical ID-based cryptography," to appear in Advances in Cryptology - Asiacrypt 2002, Lecture Notes in Computer Science, Springer, 2002.