Improvement of One Anonymous Identity-Based Encryption

Zhengjun Cao [1], Lihua Liu [2,*]

Abstract

In 2009, Seo et al. proposed an anonymous hierarchical identity-based encryption (IBE). The ciphertext consists of (C_1, C_2, C_3, C_4), where C_1 is the blinded message, C_4 is the blinded identity, both C_2 and C_3 are used as decrypting helpers. To prove its security, the authors defined five games and introduced a *strong simulator* who is able to select different Setups for those games. In this paper, we optimize the IBE scheme by removing one decrypting helper and the strong simulator. We show its security under the ℓ-computational Diffie-Hellman assumption with a normal simulator who only requires a unique Setup.

Keywords: Anonymous identity-based encryption; bilinear groups of composite order; doubly randomized key; strong simulator.

1 Introduction

The concept of identity-based encryption (IBE) was introduced by Shamir in 1984 [12]. In the scenario, one can encrypt messages using a ure's identity information. Of course, some system public parameters should be involved. In 2002, Horwitz and Lynn [10] defined the notion of hierarchical ID-based encryption (HIBE), which can handle IDs hierarchically. In 2005, Abdalla et al. [1] introduced the concepts of anonymous IBE and anonymous HIBE. But they did not give a concrete construction of anonymous HIBE. An anonymous IBE requires that the ciphertext does not leak any information about the receiver's identity. In 2006, Gentry [9] proposed a concrete construction of anonymous IBE in the standard model. Boyen and Waters [5] provided a concrete construction of anonymous HIBE. In 2009, Seo et al. [14] proposed an anonymous HIBE that has constant size ciphertexts, i.e., the size of the ciphertext does not depend on the depth of the hierarchy. The SOKS-IBE scheme [14] is based on bilinear groups of composite order, which was introduced by Boneh, Goh, and Nissim [4]. The SOKS-IBE is inspired by BBG-HIBE [2]. The BBG-HIBE provides constant size ciphertexts but does not satisfy the requirement of anonymity.

[01] Department of Mathematics, Shanghai University, Shanghai, China.

[2] Department of Mathematics, Shanghai Maritime University, China. liulh@shmtu.edu.cn

In the SKOS-IBE scheme, the ciphertext consists of (C_1, C_2, C_3, C_4), where C_1 is the blinded message, C_4 is the blinded identity, both C_2 and C_3 are used as decrypting helpers. But the two helpers are *generated and used in parallel*. To reduce its cost, it is better to remove one helper. We also observe that the ciphertext is *repeatedly randomized*. Concretely, in the ciphertext $(ME^s, G^sZ_1, F^sZ_2, (V\prod_{i=1}^k H_i^{I_i})^sZ_3)$, the session key s is used for randomizing the message M and the ID as ME^s and $(V\prod_{i=1}^k H_i^{I_i})^s$, respectively. The other session keys Z_1, Z_2, Z_3 are used for randomizing $G^s, F^s, (V\prod_{i=1}^k H_i^{I_i})^s$, respectively. That means C_2, C_3, C_4 are repeatedly randomized. Apparently, it will incur more computational cost.

To prove the security of SKOS-IBE, the authors defined five games: $\mathrm{CT}_1 = (C_1, C_2, C_3, C_4)$, $\mathrm{CT}_2 = (C_1 \cdot R_p, C_2, C_3, C_4)$, $\mathrm{CT}_3 = (C_1 \cdot R = R_1, C_2, C_3, C_4)$, $\mathrm{CT}_4 = (R_1, R_2, C_3, C_4)$, $\mathrm{CT}_5 = (R_1, R_2, R_3, R_4)$, where R_p is a randomly chosen element from $\mathbb{G}_{T,p}$; R, R_1 are uniformly distributed in $\mathbb{G}_T$; and R_2, R_3, R_4 are uniformly distributed in $\mathbb{G}$ ($\mathbb{G}_{T,p}, \mathbb{G}_T, \mathbb{G}$ are different bilinear groups). To deal with different games, it has to introduce a *strong simulator* who is able to select different Setups for those games.

Our contribution. In this paper, we improve the SKOS-IBE scheme by removing one decrypting helper and the strong simulator. We show its security under the ℓ-computational Diffie-Hellman assumption with a normal simulator who only requires a unique Setup. The analysis skills developed in the paper, we believe, are helpful to optimize other cryptographic protocols.

2 Preliminary

Bilinear groups of composite order [4]. Let $\mathcal{G}$ be a group generation algorithm that takes security parameter 1^λ as input and outputs tuple $(p, q, \mathbb{G}, \mathbb{G}_T, e)$ where p and q are distinct primes, $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of order $n = p\,q$, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a non-degenerate bilinear map; i.e., e satisfies the following properties:

(1) bilinear: for $\forall\, g_1, h_1 \in \mathbb{G}$ and $\forall\, a, b \in \mathbb{Z}$, $e(g_1^a, h_1^b) = e(g_1, h_1)^{a\,b}$;

(2) non-degenerate: for generator g_1 of $\mathbb{G}$, $e(g_1, g_1)$ generates $\mathbb{G}_T$.

Let $\mathbb{G}_p$ and $\mathbb{G}_q$ denote the subgroups of $\mathbb{G}$ of order p and q, respectively. Then $\mathbb{G} = \mathbb{G}_p \times \mathbb{G}_q$. If g_1 is a generator of $\mathbb{G}$, then g_1^q and g_1^p are generators of $\mathbb{G}_p$ and $\mathbb{G}_q$, respectively. Let g_p and g_q denote generators of $\mathbb{G}_p$ and $\mathbb{G}_q$, respectively. Notice that $e(h_p, h_q) = 1$ for all random elements $h_p \in \mathbb{G}_p$ and $h_q \in \mathbb{G}_q$ because $e(h_p, h_q) = e(g_p^a, g_q^b)$ for some integers a, b, and $e(g_p^a, g_q^b) = e(g_1^{q\,a}, g_1^{p\,b}) = e(g_1, g_1)^{p\,q\,a\,b} = 1$ for some generator g_1 in $\mathbb{G}$.

ℓ-computational Diffie-Hellman assumption. Given a cyclic group $\mathbb{G}$ of prime order p, a random generator g and $(g^a, g^{a^2}, \cdots, g^{a^\ell})$ for some random $a \in \mathbb{Z}_p^*$, it is computationally intractable

to compute $g^{a^{\ell+1}}$.

Security definitions of anonymous HIBE. We refer to [1, 3] for the formal security definitions of anonymous HIBE, and refer to [6, 7] for a weaker notion of security that the adversary commits ahead of time to the public parameters that it will attack.

3 Analysis of the SKOS-IBE scheme

3.1 Review

Setup: Given a security parameter λ and the maximum hierarchy depth L, the algorithm generates $(p, q, \mathbb{G}, \mathbb{G}_T, e)$. Pick random elements

$$g, f, v, h_1, \cdots, h_L, w \in \mathbb{G}_p, \qquad R_g, R_f, R_v, R_1, \cdots, R_L \in \mathbb{G}_q.$$

and compute $G = gR_g$, $F = fR_f$, $V = vR_v$, $H_1 = h_1R_1$, $\cdots$, $H_L = h_LR_L$, $E = e(g, w)$. Publish the description of a group $\mathbb{G}$ and public system parameters as $[g_q, G, F, V, H_1, \cdots, H_L, E]$. The master secret key is set as $[p, q, g, f, v, h_1, \cdots, h_L, w]$. The group description contains n but not p, q.

KeyGenerate: Given ID=$[I_1, I_2, \cdots, I_k] \in (\mathbb{Z}_n)^k$, pick random $r_1, r_2, s_1, s_2, t_1, t_2 \in \mathbb{Z}_n$ such that $s_1t_2 - s_2t_1 \neq 0 \bmod p$ and $\neq 0 \bmod q$. Output

$$\mathrm{Pvk}_{\mathrm{d}}^{\mathrm{ID}} = [w(v\prod_{i=1}^{k} h_i^{I_i})^{r_1} f^{r_2}, g^{r_1}, g^{r_2}, h_{k+1}^{r_1}, \cdots, h_L^{r_1}].$$

$$\mathrm{Pvk}_{\mathrm{r}}^{\mathrm{ID}} = [[(v\prod_{i=1}^{k} h_i^{I_i})^{s_1} f^{s_2}, g^{s_1}, g^{s_2}, h_{k+1}^{s_1}, \cdots, h_L^{s_1}], [(v\prod_{i=1}^{k} h_i^{I_i})^{t_1} f^{t_2}, g^{t_1}, g^{t_2}, h_{k+1}^{t_1}, \cdots, h_L^{t_1}]].$$

where $\mathrm{Pvk}_{\mathrm{d}}^{\mathrm{ID}}$ is used for decryption and delegation, and $\mathrm{Pvk}_{\mathrm{r}}^{\mathrm{ID}}$ is used for re-randomization.

Derive: Given a private key for the parent,

$$\mathrm{Pvk}^{\mathrm{ID}_{|k-1}} = [\mathrm{Pvk}_d^{\mathrm{ID}_{|k-1}}, \mathrm{Pvk}_r^{\mathrm{ID}_{|k-1}}]$$

$$= [[a_0, a_1, a_2, b_k, \cdots, b_L], [[\alpha_0, \alpha_1, \alpha_2, \beta_k, \cdots, \beta_L], [\alpha_0', \alpha_1', \alpha_2', \beta_k', \cdots, \beta_L']]],$$

pick random $\gamma_1, \gamma_2, \gamma_3, \delta_1, \delta_2, \delta_3 \in \mathbb{Z}_n$ such that $g_p^{\gamma_2\delta_3 - \gamma_3\delta_2} \neq 1$ and $g_q^{\gamma_2\delta_3 - \gamma_3\delta_2} \neq 1$. Output

$$\mathrm{Pvk}_d^{\mathrm{ID}_{|k}} = [\zeta_0\theta_0^{\gamma_1}\theta_0'^{\delta_1}, \zeta_1\theta_1^{\gamma_1}\theta_1'^{\delta_1}, \zeta_2\theta_2^{\gamma_1}\theta_2'^{\delta_1}, \eta_{k+1}\phi_{k+1}^{\gamma_1}\phi_{k+1}'^{\delta_1}, \cdots, \eta_L\phi_L^{\gamma_1}\phi_L'^{\delta_1}]$$

$$\begin{aligned}\mathrm{Pvk}_r^{\mathrm{ID}_{|k}} \;=\; & [[\theta_0^{\gamma_2}\theta_0'^{\delta_2}, \theta_1^{\gamma_2}\theta_1'^{\delta_2}, \theta_2^{\gamma_2}\theta_2'^{\delta_2}, \phi_{k+1}^{\gamma_2}\phi_{k+1}'^{\delta_2}, \cdots, \phi_L^{\gamma_2}\phi_L'^{\delta_2}],\\ & [\theta_0^{\gamma_3}\theta_0'^{\delta_3}, \theta_1^{\gamma_3}\theta_1'^{\delta_3}, \theta_2^{\gamma_3}\theta_2'^{\delta_3}, \phi_{k+1}^{\gamma_3}\phi_{k+1}'^{\delta_3}, \cdots, \phi_L^{\gamma_3}\phi_L'^{\delta_3}]]\end{aligned}$$

where

$$[\zeta_0, \zeta_1, \zeta_2, \eta_{k+1}, \cdots, \eta_L] = [a_0 \cdot b_k^{I_k}, a_1, a_2, b_{k+1}, \cdots, b_L]$$

$$[\theta_0, \theta_1, \theta_2, \phi_{k+1}, \cdots, \phi_L] = [\alpha_0 \cdot \beta_k^{I_k}, \alpha_1, \alpha_2, \beta_{k+1}, \cdots, \beta_L]$$

$$[\theta'_0, \theta'_1, \theta'_2, \phi'_{k+1}, \cdots, \phi'_L] = [\alpha'_0 \cdot \beta_k'^{I_k}, \alpha'_1, \alpha'_2, \beta'_{k+1}, \cdots, \beta'_L]$$

Encrypt: To encrypt message $M \in \mathbb{G}_T$ for a given identity $ID = [I_1, \cdots, I_k] \in (\mathbb{Z}_n)^k$, pick a random $s \in \mathbb{Z}_n$ and random $Z_1, Z_2, Z_3 \in \mathbb{G}_q$. Output the ciphertext

$$(ME^s, G^s Z_1, F^s Z_2, (V \prod_{i=1}^{k} H_i^{I_i})^s Z_3).$$

Decrypt: To decrypt ciphertext (C_1, C_2, C_3, C_4) with respect to $ID = [I_1, \cdots, I_k]$, using the first three elements of subkey $\mathrm{Pvk}_d^{\mathrm{ID}} = [a_0, a_1, a_2, b_{k+1}, \cdots, b_L]$, compute

$$M = C_1 \cdot e(a_1, C_4) \cdot e(a_2, C_3)/e(a_0, C_2)$$

3.2 Analysis

On the doubly randomized key. The ciphertext consists of (C_1, C_2, C_3, C_4), where C_1 is the blinded message, C_4 is the blinded identity, both C_2 and C_3 are decrypting helpers. The reason to set two decrypting helpers is that the authors adopt the doubly randomized key, i.e.,

$$\mathrm{Pvk}_{\mathrm{d}}^{\mathrm{ID}} = [w(v \prod_{i=1}^{k} h_i^{I_i})^{r_1} f^{r_2}, g^{r_1}, g^{r_2}, h_{k+1}^{r_1}, \cdots, h_L^{r_1}].$$

Notice that $a_1 = g^{r_1}$ and $a_2 = g^{r_2}$ are used for decryption in parallel. But we know the setting is unnecessary because it incurs more computational cost. Based on this observation, we can set the decrypting key as

$$\mathrm{Pvk}_{\mathrm{d}}^{\mathrm{ID}} = [w(v \prod_{i=1}^{k} h_i^{I_i})^{r_1}, g^{r_1}, h_{k+1}^{r_1}, \cdots, h_L^{r_1}],$$

and the re-randomizing key as

$$\mathrm{Pvk}_{\mathrm{r}}^{\mathrm{ID}} = [[(v \prod_{i=1}^{k} h_i^{I_i})^{s_1}, g^{s_1}, h_{k+1}^{s_1}, \cdots, h_L^{s_1}], [(v \prod_{i=1}^{k} h_i^{I_i})^{t_1}, g^{t_1}, h_{k+1}^{t_1}, \cdots, h_L^{t_1}]].$$

Correspondingly, the system parameters can be optimized as

$$[G, V, H_1, \cdots, H_L, E], \qquad [p, q, g, v, h_1, \cdots, h_L, w]$$

for the public system parameters and the master secret key, respectively.

Taking into account that the *fixed argument* for bilinear map using the Miller algorithm [11] is more efficient than that for unfixed argument, we can further optimize the SKOS-IBE scheme by setting that $w = v$. We will show the change does not endanger its security.

On repeatedly randomizing the ciphertext. To encrypt a message $M \in \mathbb{G}_T$ for a given identity $ID = [I_1, \cdots, I_k] \in (\mathbb{Z}_n)^k$, it randomly picks $s \in \mathbb{Z}_n$, $Z_1, Z_2, Z_3 \in \mathbb{G}_q$, and computes the ciphertext $(ME^s, G^s Z_1, F^s Z_2, (V \prod_{i=1}^k H_i^{I_i})^s Z_3)$. We here stress that it is unnecessary to repeatedly randomizing $G^s, F^s, (V \prod_{i=1}^k H_i^{I_i})^s$ with Z_1, Z_2, Z_3, respectively. The structure of $(V \prod_{i=1}^k H_i^{I_i})^s$ suffices to blind the identity $[I_1, \cdots, I_k]$ because one can not recover the secret exponent s, which is usually called *session key*. Therefore, it is better to remove those redundant blinders Z_1, Z_2, Z_3.

On the strong simulator. To prove its security, the authors defined five games and introduced a *strong simulator* who is able to select different Setups for those games. See Lemma 1, Lemma 3, and Lemma 4 in the Section 3.2 [14] for details. We will show the security of the improvement under the ℓ-computational Diffie-Hellman assumption with a normal simulator who only requires a unique Setup.

4 Improvement of SKOS-IBE

4.1 Construction

Setup: Given a security parameter λ and the maximum hierarchy depth L, the algorithm generates $(p, q, \mathbb{G}, \mathbb{G}_T, e)$. Pick random elements

$$g, v, h_1, \cdots, h_L \in \mathbb{G}_p, \qquad R_g, R_v, R_1, \cdots, R_L \in \mathbb{G}_q.$$

and compute $G = gR_g$, $V = vR_v$, $H_1 = h_1 R_1, \cdots, H_L = h_L R_L$, $E = e(g, v)$. Publish the description of a group $\mathbb{G}$ and public system parameters as $[G, V, H_1, \cdots, H_L, E]$. The master secret key is set as $[p, q, g, v, h_1, \cdots, h_L]$.

KeyGenerate: Given ID=$[I_1, I_2, \cdots, I_k] \in (\mathbb{Z}_n)^k$, pick random $r_1, s_1, t_1 \in \mathbb{Z}_n$, output

$$\mathrm{Pvk}_{\mathrm{d}}^{\mathrm{ID}} = [v(v \prod_{i=1}^{k} h_i^{I_i})^{r_1}, g^{r_1}, h_{k+1}^{r_1}, \cdots, h_L^{r_1}].$$

$$\mathrm{Pvk}_{\mathrm{r}}^{\mathrm{ID}} = [[(v \prod_{i=1}^{k} h_i^{I_i})^{s_1}, g^{s_1}, h_{k+1}^{s_1}, \cdots, h_L^{s_1}], [(v \prod_{i=1}^{k} h_i^{I_i})^{t_1}, g^{t_1}, h_{k+1}^{t_1}, \cdots, h_L^{t_1}]].$$

Derive: Given a private key for the parent,

$$\mathrm{Pvk}^{\mathrm{ID}_{|k-1}} = [\mathrm{Pvk}_d^{\mathrm{ID}_{|k-1}}, \mathrm{Pvk}_r^{\mathrm{ID}_{|k-1}}]$$

$$= [[a_0, a_1, b_k, \cdots, b_L], [[\alpha_0, \alpha_1, \beta_k, \cdots, \beta_L], [\alpha'_0, \alpha'_1, \beta'_k, \cdots, \beta'_L]]],$$

pick random $\gamma_1, \gamma_2, \gamma_3, \delta_1, \delta_2, \delta_3 \in \mathbb{Z}_n$ such that $g_p^{\gamma_2\delta_3 - \gamma_3\delta_2} \neq 1$ and $g_q^{\gamma_2\delta_3 - \gamma_3\delta_2} \neq 1$. Output

$$\mathrm{Pvk}_d^{\mathrm{ID}_{|k}} = [\zeta_0 \theta_0^{\gamma_1} \theta_0'^{\delta_1}, \zeta_1 \theta_1^{\gamma_1} \theta_1'^{\delta_1}, \eta_{k+1} \phi_{k+1}^{\gamma_1} \phi_{k+1}'^{\delta_1}, \cdots, \eta_L \phi_L^{\gamma_1} \phi_L'^{\delta_1}]$$

$$\begin{aligned}\mathrm{Pvk}_r^{\mathrm{ID}_{|k}} &= [[\theta_0^{\gamma_2}\theta_0'^{\delta_2}, \theta_1^{\gamma_2}\theta_1'^{\delta_2}, \phi_{k+1}^{\gamma_2}\phi_{k+1}'^{\delta_2}, \cdots, \phi_L^{\gamma_2}\phi_L'^{\delta_2}],\\ &\quad [\theta_0^{\gamma_3}\theta_0'^{\delta_3}, \theta_1^{\gamma_3}\theta_1'^{\delta_3}, \phi_{k+1}^{\gamma_3}\phi_{k+1}'^{\delta_3}, \cdots, \phi_L^{\gamma_3}\phi_L'^{\delta_3}]]\end{aligned}$$

where

$$[\zeta_0, \zeta_1, \eta_{k+1}, \cdots, \eta_L] = [a_0 \cdot b_k^{I_k}, a_1, b_{k+1}, \cdots, b_L]$$
$$[\theta_0, \theta_1, \phi_{k+1}, \cdots, \phi_L] = [\alpha_0 \cdot \beta_k^{I_k}, \alpha_1, \beta_{k+1}, \cdots, \beta_L]$$
$$[\theta_0', \theta_1', \phi_{k+1}', \cdots, \phi_L'] = [\alpha_0' \cdot \beta_k'^{I_k}, \alpha_1', \beta_{k+1}', \cdots, \beta_L']$$

Encrypt: To encrypt message $M \in \mathbb{G}_T$ for a given identity $ID = [I_1, \cdots, I_k] \in (\mathbb{Z}_n)^k$, pick a random $s \in \mathbb{Z}_n$ and output the ciphertext

$$(ME^s, G^s, (V\prod_{i=1}^{k} H_i^{I_i})^s).$$

Decrypt: To decrypt ciphertext (C_1, C_2, C_3) with respect to $ID = [I_1, \cdots, I_k]$, using the first two elements of subkey $\mathrm{Pvk}_d^{\mathrm{ID}} = [a_0, a_1, b_{k+1}, \cdots, b_L]$, compute

$$M = C_1 \cdot e(a_1, C_3)/e(a_0, C_2)$$

Correctness.

$$\begin{aligned}C_1 \cdot e(a_1, C_3)/e(a_0, C_2) &= ME^s \cdot e(a_1, (V\prod_{i=1}^{k} H_i^{I_i})^s)/e(a_0, G^s)\\ &= Me(g,v)^s \cdot \frac{e\left(g^{r_1}, (vR_v\prod_{i=1}^{k}(h_iR_i)^{I_i})^s\right)}{e\left(v(v\prod_{i=1}^{k} h_i^{I_i})^{r_1}, G^s\right)}\\ &= Me(g,v)^s \cdot \frac{e\left(g^{r_1}, (v\prod_{i=1}^{k} h_i^{I_i})^s\right)}{e\left(v(v\prod_{i=1}^{k} h_i^{I_i})^{r_1}, g^s\right)}\\ &= Me(g,v)^s/e(v, g^s) = M\end{aligned}$$

Notice that we here have to use the property that $e(h_p, h_q) = 1$ for all $h_p \in \mathbb{G}_p$ and $h_q \in \mathbb{G}_q$.

Table 1: SKOS-IBE and the improvement

	SKOS-IBE	The improvement
Setup	PK: $g_q, G, F, V, H_1, \cdots, H_L, E$ SK: $p, q, g, f, v, h_1, \cdots, h_L, w$	PK: $G, V, H_1, \cdots, H_L, E$ SK: $p, q, g, v, h_1, \cdots, h_L$
KeyGenerate	Pick $r_1, r_2 \in \mathbb{Z}_n$, compute $\mathrm{Pvk}_{\mathrm{d}}^{\mathrm{ID}}$ as $a = (w(v\prod_{i=1}^{k} h_i^{I_i})^{r_1} f^{r_2}, g^{r_1}, g^{r_2}, h_{k+1}^{r_1}, \cdots, h_L^{r_1})$	Pick $r_1 \in \mathbb{Z}_n$, compute $\mathrm{Pvk}_{\mathrm{d}}^{\mathrm{ID}}$ as $a = (v(v\prod_{i=1}^{k} h_i^{I_i})^{r_1}, g^{r_1}, h_{k+1}^{r_1}, \cdots, h_L^{r_1})$
Encrypt	Pick $s \in \mathbb{Z}_n$, $Z_1, Z_2, Z_3 \in \mathbb{G}_q$, compute $C = (ME^s, G^sZ_1, F^sZ_2, (V\prod_{i=1}^{k} H_i^{I_i})^sZ_3)$	Pick $s \in \mathbb{Z}_n$, compute $C = (ME^s, G^s, (V\prod_{i=1}^{k} H_i^{I_i})^s)$
Decrypt	$M = C_1 \cdot e(a_1, C_4) \cdot e(a_2, C_3)/e(a_0, C_2)$	$M = C_1 \cdot e(a_1, C_3)/e(a_0, C_2)$

4.2 Security proof

Theorem 1. *If the Setup and KeyGenerate algorithms satisfy the* $(t, \epsilon)-$ ℓ*-computational Diffie-Hellman assumption, then there is no adversary with running time* t *that succeeds to decrypt a ciphertext with advantage* ϵ*.*

Proof. We assume there exists adversary $\mathcal{A}$ that succeeds to decrypt a ciphertext with advantage ϵ. We show that there is a simulator $\mathcal{B}$ using $\mathcal{A}$ to solve the ℓ-computational Diffie-Hellman problem with advantage ϵ. The adversary $\mathcal{A}$ and simulator $\mathcal{B}$ run the following game.

Initialization. $\mathcal{A}$ chooses identity $ID = [I_1, I_2, \cdots, I_m]$, and sets $I_{m+1} = \cdots = I_L = 0$. Then $\mathcal{A}$ picks a random $a \in \mathbb{Z}_n$ and sets $A_i = g_p^{a^i}$ for $1 \leq i \leq L$. $\mathcal{A}$ sends ID and A_i $(1 \leq i \leq L)$ to the simulator $\mathcal{B}$, and keeps the secret a.

Setup. $\mathcal{B}$ picks random integers and random elements

$$y, x_1, \cdots, x_L \in \mathbb{Z}_n, \qquad R_g, R_v, R_{h,1}, \cdots, R_{h,l} \in \mathbb{G}_q.$$

Notice that a random element of $\mathbb{G}_p(\mathbb{G}_q)$ can be chosen by raising g_p (g_q, respectively) to random exponents from $\mathbb{Z}_n$. $\mathcal{B}$ computes $v = g_p^y \prod_{i=1}^{L} (A_{L-i+1})^{I_i}$ and sets

$$G = g_p R_g,\ V = (g_p^y \prod_{i=1}^{L} (A_{L-i+1})^{I_i}) R_v,\ E = e(A_1, v),\ H_i = g_p^{x_i} / A_{L-i+1} R_{h,i}, \text{for} \, 1 \leq i \leq L.$$

Then $\mathcal{B}$ sends $(v, h_1, \cdots, h_L)$ to $\mathcal{A}$, where $h_i = g_p^{x_i} / A_{L-i+1}$ for $1 \leq i \leq L$. $\mathcal{B}$ finally publishes these parameters $(G, V, E, H_1, \cdots, H_L)$.

Query. For $ID^* = [I_1^*, I_2^*, \cdots, I_u^*]$, where $u \leq L$ is distinct from ID and all its prefixes, $\mathcal{B}$ chooses random integers $r_1 \in \mathbb{Z}_n$ and sends (r_1, ID^*) to $\mathcal{A}$.

Response. Let k be the smallest integer such that $I_k \neq I_k^*$. $\mathcal{A}$ sets $\hat{r}_1 = r_1 + a^k / (I_k^* - I_k)$ and picks random $s_1, t_1 \in \mathbb{Z}_n$. Then $\mathcal{A}$ computes

$$\mathrm{Pvk}_{\mathrm{d}}^{\mathrm{ID}} = [v(v \prod_{i=1}^{k} h_i^{I_i^*})^{\hat{r}_1}, g^{\hat{r}_1}, h_{k+1}^{\hat{r}_1}, \cdots, h_L^{\hat{r}_1}],$$

$$\mathrm{Pvk}_{\mathrm{r}}^{\mathrm{ID}} = [[(v \prod_{i=1}^{k} h_i^{I_i^*})^{s_1}, g^{s_1}, h_{k+1}^{s_1}, \cdots, h_L^{s_1}], [(v \prod_{i=1}^{k} h_i^{I_i^*})^{t_1}, g^{t_1}, h_{k+1}^{t_1}, \cdots, h_L^{t_1}]],$$

and sends $\mathrm{Pvk}^{\mathrm{ID}}$ to $\mathcal{B}$.

Output. Denote the first component of $\mathrm{Pvk}_{\mathrm{d}}^{\mathrm{ID}}$ by

$$\tau = v(v \prod_{i=1}^{k} h_i^{I_i^*})^{\hat{r}_1},$$

then we have

$$\begin{aligned}
\tau/v &= (v\prod_{i=1}^{k} h_i^{I_i^*})^{\hat{r}_1} = (v\prod_{i=1}^{k} h_i^{I_i^*})^{r_1} \cdot (v\prod_{i=1}^{k} h_i^{I_i^*})^{a^k/(I_k^*-I_k)} \\
&= (v\prod_{i=1}^{k} h_i^{I_i^*})^{r_1} \cdot \left(g_p^y \prod_{i=1}^{L} (A_{L-i+1})^{I_i} \prod_{i=1}^{k} (g_p^{x_i}/A_{L-i+1})^{I_i^*} \right)^{a^k/(I_k^*-I_k)} \\
&= (v\prod_{i=1}^{k} h_i^{I_i^*})^{r_1} \cdot \left(g_p^y A_{L-k+1}^{I_k-I_k^*} \prod_{i=k+1}^{L} (A_{L-i+1})^{I_i} \prod_{i=1}^{k} g_p^{x_i I_i^*} \right)^{a^k/(I_k^*-I_k)} \\
&= (v\prod_{i=1}^{k} h_i^{I_i^*})^{r_1} \cdot \left(A_k^y A_{L+1}^{I_k-I_k^*} \prod_{i=k+1}^{L} (A_{L+k-i+1})^{I_i} \prod_{i=1}^{k} A_k^{x_i I_i^*} \right)^{1/(I_k^*-I_k)} \\
&= (v\prod_{i=1}^{k} h_i^{I_i^*})^{r_1} \cdot A_{L+1}^{-1} \left(A_k^y \prod_{i=k+1}^{L} (A_{L+k-i+1})^{I_i} \prod_{i=1}^{k} A_k^{x_i I_i^*} \right)^{1/(I_k^*-I_k)}
\end{aligned}$$

Hence,

$$g^{a^{L+1}} = A_{L+1} = \frac{v}{\tau} \cdot (v\prod_{i=1}^{k} h_i^{I_i^*})^{r_1} \cdot \left(A_k^y \prod_{i=k+1}^{L} (A_{L+k-i+1})^{I_i} \prod_{i=1}^{k} A_k^{x_i I_i^*} \right)^{1/(I_k^*-I_k)}.$$

Since $\mathcal{B}$ knows $k, L, \tau, v, y, r_1, x_i, A_i, h_i$, for all $1 \le i \le L$, he can compute the right side. Thus, $\mathcal{B}$ can obtain $g^{a^{L+1}}$. That is, $\mathcal{B}$ can solve the ℓ-computational Diffie-Hellman problem. (We refer to the following Table 2 for the security proof simulation) □

5 Conclusion

In this paper, we improve the SKOS-IBE scheme and prove its security under ℓ-computational Diffie-Hellman assumption. The original scheme adopts the paradigm of doubly randomized key which incurs more computational cost. Although the paradigm is rarely used in those practical protocols, such as, RSA, DSA and Schnorr cryptosystem [13], the setting is more apt for constructing a subliminal channel [8] because the redundant keys can be privately shared by the users who want to communicate over the channel. It seems possible to further improve the SKOS-IBE scheme by blinding the message and the identity *simultaneously* with a single blinder. But it seems difficult to prove its security under the general ℓ-computational Diffie-Hellman assumption with a normal simulator.

Table 2: Simulation for our construction

$\mathcal{A}$		$\mathcal{B}$
Pick $ID = [I_1, I_2, \cdots, I_m]$ and, set $I_{m+1} = \cdots = I_L = 0$. Pick $a \in \mathbb{Z}_n$ and set $A_i = g_p^{a^i}$, $1 \le i \le L$.		
	$\xrightarrow{ID, A_i, i=1,\cdots,L}$	
		Pick $y, x_1, \cdots, x_L \in \mathbb{Z}_n$, $R_g, R_v, R_{h,1}, \cdots, R_{h,l} \in \mathbb{G}_q$. Compute $v = g_p^y \prod_{i=1}^{L} (A_{L-i+1})^{I_i}$ and set $G = g_p R_g, E = e(A_1, v)$, $V = (g_p^y \prod_{i=1}^{L} (A_{L-i+1})^{I_i}) R_v$, $H_i = g_p^{x_i} / A_{L-i+1} R_{h,i}, 1 \le i \le L$. Set $h_i = g_p^{x_i} / A_{L-i+1}$, $1 \le i \le L$. Publish $(G, V, E, H_1, \cdots, H_L)$.
	$\xleftarrow{(v, h_1, \cdots, h_L)}$	
		For $ID^* = [I_1^*, I_2^*, \cdots, I_u^*]$, $u \le L$, pick $r_1 \leftarrow \mathbb{Z}_n$.
	$\xleftarrow{(r_1, ID^*)}$	
Set k be the smallest integer such that $I_k \ne I_k^*$. Set $\hat{r}_1 = r_1 + a^k/(I_k^* - I_k)$ and pick $s_1, t_1 \in \mathbb{Z}_n$. Compute $\mathrm{Pvk}_{\mathrm{d}}^{\mathrm{ID}} = [v(v \prod_{i=1}^{k} h_i^{I_i^*})^{\hat{r}_1}, g^{\hat{r}_1}, h_{k+1}^{\hat{r}_1}, \cdots, h_L^{\hat{r}_1}]$, $\mathrm{Pvk}_{\mathrm{r}}^{\mathrm{ID}} = [[(v \prod_{i=1}^{k} h_i^{I_i^*})^{s_1}, g^{s_1}, h_{k+1}^{s_1}, \cdots, h_L^{s_1}], [(v \prod_{i=1}^{k} h_i^{I_i^*})^{t_1}, g^{t_1}, h_{k+1}^{t_1}, \cdots, h_L^{t_1}]]$		
	$\xrightarrow{\mathrm{Pvk}^{\mathrm{ID}}}$	
		Output $A_{L+1} = \frac{v}{\tau} \cdot (v \prod_{i=1}^{k} h_i^{I_i^*})^{r_1} \cdot \left(A_k^y \prod_{i=k+1}^{L} (A_{L+k-i+1})^{I_i} \prod_{i=1}^{k} A_k^{x_i I_i^*} \right)^{\frac{1}{(I_k^* - I_k)}}$

References

[1] Abdalla, M., et al.: Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 205-222. Springer, Heidelberg (2005)

[2] Boneh, D., Boyen, X., Goh, E.-J.: Hierarchical identity based encryption with constant size ciphertexts. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440-456. Springer, Heidelberg (2005)

[3] Boneh, D., Franklin, M.: Identity-Based Encryption from the Weil Pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213-229. Springer, Heidelberg (2001)

[4] Boneh, D., Goh, E., Nissim, K.: Evaluating 2-dnf formulas on ciphertexts. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 325-341. Springer, Heidelberg (2005)

[5] Boyen, X., Waters, B.: Anonymous hierarchical identity-based encryption (without random oracles). In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 290-307. Springer, Heidelberg (2006)

[6] Canetti, R., Halevi, S., Katz, J.: A forward-secure public-key encryption scheme. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 255-271. Springer, Heidelberg (2003)

[7] Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207-222. Springer, Heidelberg (2004)

[8] Desmedt Y.: Subliminal-free authentication and signature (Extended Abstract). In: Günther C.(ed.) EUROCRYPT 1988. LNCS, vol. 330, pp. 23-33. Springer, Heidelberg (1988)

[9] Gentry, C.: Practical identity-based encryption without random oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445-464. Springer, Heidelberg (2006)

[10] Horwitz, J., Lynn, B.: Towards hierarchical identity-based encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 466-481. Springer, Heidelberg (2002)

[11] Miller V.: The Weil Pairing, and Its Efficient Calculation. J. Cryptology 17(4), pp. 235-261. Springer, Heidelberg (2004)

[12] Shamir, A.: Identity-Based Cryptosystems and Signature Schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47-53. Springer, Heidelberg (1985)

[13] Schnorr C.: Efficient signature generation for smart cards. In: Brassard G. (ed.) CRYPTO'1989. LNCS, vol. 435, pp. 239-252, Springer Heidelberg (1989)

[14] Seo J., Kobayashi T., Ohkubo M., Suzuki K.: Anonymous Hierarchical Identity-Based Encryption with Constant Size Ciphertexts. In: Jarecki S., Tsudik G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 215-234, Springer, Heidelberg (2009)