Removing Key Escrow from the LW-HIBE Scheme

Peixin Chen[1(✉)], Xiaofeng Wang[1], Baokang Zhao[1], Jinshu Su[1,2], and Ilsun You[3]

[1] College of Computer, National University of Defense Technology, 410073 Changsha, China
{chenpeixin,xf_wang,bkzhao,sjs}@nudt.edu.cn
[2] National Key Laboratory for Parallel and Distributed Processing, National University of Defense Technology, 410073 Changsha, China
[3] School of Information Science, Korean Bible University, Nowon District, Korea
isyou@bible.ac.kr

Abstract. Hierarchical Identity-Based Encryption (HIBE) provides an efficient solution to the security problems existed in cloud storage. However, the key escrow problem, which is an inherent problem in HIBE, primarily hinders the widespread adoption of the cryptographic scheme in practice. To address the key escrow problem, this paper introduces a provably-secure escrow-free model, which employs multiple Key Privacy Authorities (KPAs) to restrict the power of Public Key Generators (PKGs) in HIBE scheme. We instantiate the model into an escrow-free HIBE scheme that is referred to as the EF-LW-HIBE scheme, based on the HIBE scheme introduced by Lewko and Waters. Utilizing the Dual System Encryption methodology, we prove that our EF-LW-HIBE scheme is IND-ID-CCA secure.

Keywords: Hierarchical identity-based encryption · Key escrow · IND-ID-CCA Security · Dual system encryption · Cloud storage

1 Introduction

Cloud storage is becoming increasingly popular with the rapidly network technology development. It does provide convenient and offer more flexibility to people that one can access the data on cloud storage system via the Internet instead of carrying around a physical storage. However, cloud storage has the potential for security and compliance concerns. Many works on securing the cloud storage have been presented [9,11,12,17,19]

Identity-Based Encryption (IBE) can be easily apply to the cloud storage. IBE is a public key encryption scheme which allows a sender to encrypt message for a receiver using the receiver's identity, such as IP address or email address, as the public key [16]. Boneh and Franklin first formulate the concept of IBE and propose a full functional scheme (BF-IBE) based on bilinear maps between

G. Wang et al. (Eds.): ICA3PP 2015 Workshops, LNCS 9532, pp. 593–605, 2015.
DOI: 10.1007/978-3-319-27161-3_54

groups [4]. The IBE scheme uses a trusted authority called Private Key Generator (PKG) to generate private key for users. In order to reduce the workload of the PKG, Gentry and Silverberg present the first construction of Hierarchical IBE (HIBE) with a root PKG and several domain PKGs in different levels [10]. To improve the efficient and security, a number HIBE schemes [1–3,7,15,18] have been presented.

Since the user private keys are generated by the PKGs, the construction of HIBE will inevitably lead to the key escrow problem. That is, the PKG knows all the private keys of its descendant and thus can unscrupulously decrypt the message intended for the users and maliciously make users' private keys public. Many prior works have been proposed to solve the problem [4,6,8,13,14]. One intuitional approach is to use distributed PKGs to reduce the power of single PKG. In the first IBE scheme, Bonech et al. apply the threshold method to suggest an (n, t) distributed PKG mechanism [4]. They distribute the master key into n parts that each PKG owns only one portion, and any more than $t+1$ PKGs can jointly compute a private key. However, they do not provide a formal security model and a proof. Kate and Goldberg present an efficient distributed PKGs model and construct the schemes for three well-known IBE schemes: BF-IBE, SK-IBE and BB_1-IBE [13]. However, the model cannot apply to the HIBE scheme and the presented schemes can only achieve security in the random oracle model. Other than the multiple PKGs mechanisms, Lee et al. present a key issuing model which introduced Key Privacy Authorities (KPAs) to protect the user private key privacy so that the PKG cannot obtain the complete information of the keys [14]. The idea of the multiple-KPAs model is inspired by the real word scenario such as elections, in which there is a single election administrator organizing the election procedures and multiple observers dispatched by major political parties to the voting office to prevent any illegal activity. Key escrow problem can be effectively reduced based on the assumption that at least one of the KPAs is honest. However, the key privacy service in their model needs to be sequential processed. Therefore, the multiple-KPAs model would introduce too high overhead to the basic HIBE scheme to make it practical. Moreover, the security of this key issuing mechanism has not been proved by formal approach. Because of the possibility of theoretically insecurity, this mechanism cannot be applied in practice. Cao et al. also use KPAs to achieve an escrow-free HIBE scheme (SA-HIBE) [6]. SA-HIBE avoids the inefficient sequential procedure in [14] and allows users to interact with KPAs synchronously. However, the scheme efficiency is still low because of the ciphertext and private keys, as well as the encryption and decryption time in SA-HIBE grow linearly in the depth of the hierarchy. And the scheme security is also proved in the random oracle model.

In this paper, we propose an efficient and provably-secure EF-LW-HIBE scheme, which is an escrow-free HIBE scheme based on the LW-HIBE. The EF-LW-HIBE scheme makes use of multiple KPAs to restrict the power of PKGs in HIBE so that the PKGs cannot obtain the full information of the private keys. On account of the synchronous key securing procedure with KPAs, our

escrow-free model introduces acceptable cost to LW-HIBE. With the help of Dual System Encryption, we prove the full security of EF-LW-HIBE without random oracle model.

2 Preliminaries

2.1 Composite Order Bilinear Groups

Composite order bilinear groups were first introduced by Boneh et al. [5]. Let p_1, p_2, p_3 be distinct primes, and set $N = p_1p_2p_3$. For two multiplicative cyclic groups G and G_T of order N, we say a map $e : G \times G \rightarrow G_T$ is a bilinear map if it meets the following properties:

1. Bilinear: $\forall g, h \in G, a, b \in \mathbb{Z}_N, e(g^a, h^b) = e(g, h)^{ab}$
2. Non-degenerate: $\exists g \in G$, s.t. $e(g, g) \neq 1$
3. Computable: $\forall g, h \in G$, there is an efficient algorithm to compute $e(g, h)$.

Group G is referred to as a composite order bilinear group. Let G_{p_1}, G_{p_2}, and G_{p_3} denote the subgroups of order p_1, p_2 and p_3 in G respectively. Lewko and Waters have illuminated that, when $h_i \in G_i$, and $h_j \in G_j$ for $i \neq j$, $e(h_i, h_j)$ is an identity element in G_T [15]. Such property is referred to as *orthogonality property* of $G_{p_1}, G_{p_2}, G_{p_3}$.

2.2 Complexity Assumptions

We prove the security of EF-LW-HIBE scheme based on the same complexity assumptions as the LW-HIBE scheme [15] does. Let $G_{p_ip_j}$ denote subgroup of order p_ip_j in G, the assumptions are defined as follows.

Assumption 1. Let g, T_2 be distinct random elements of G_{p_1}, X_3 be a random element of G_{p_3} and T_1 be a random element of $G_{p_1p_2}$. Randomly picking $T \in \{T_1, T_2\}$, we assume that given g, X_3, there is no probabilistic polynomial time (PPT) algorithm $\mathcal{A}$ can determine $T \in G_{p_1p_2}$ or $T \in G_{p_1}$ with negligible advantage.

Assumption 2. Let g, X_1 be distinct random elements of G_{p_1}, X_2, Y_2 be distinct random elements of G_{p_2}, X_3, Y_3 be distinct random elements of G_{p_3}, T_1 be a random element of G, and T_2 be a random element of $G_{p_1p_3}$. Randomly picking $T \in \{T_1, T_2\}$, we assume that given g, X_1X_2, X_3, Y_2Y_3, there is no PPT algorithm $\mathcal{A}$ can determine $T \in G$ or $T \in G_{p_1p_3}$ with negligible advantage.

Assumption 3. Randomly pick $\alpha, s \in \mathbb{Z}_N$. Let g be a random element of G_{p_1}, X_2, Y_2, Z_2 be distinct random elements of G_{p_2}, X_3 be a random element of G_{p_3}. Set $T_1 = e(g, g)^{\alpha s}$ and let T_2 be a random element of G_T. Randomly picking $T \in \{T_1, T_2\}$, we assume that given $g, g^{\alpha}X_2, X_3, g^sY_2, Z_2$, there is no PPT algorithm $\mathcal{A}$ can determine $T = e(g, g)^{\alpha s}$ or T is a random element of G_T with negligible advantage.

2.3 Dual System Encryption

Dual System Encryption is a scheme that is used for proving security of encryption schemes [18]. For proving, two additional structures called semi-functional key and semi-functional ciphertext are used. A semi-functional key is an efficient mathematical transformation of a normal key, and so as a semi-functional ciphertext. Suppose CT_{normal} is a ciphertext with regard to normal key K_{normal}. Let K_{semi} be a semi-functional key w.r.t. K_{normal}, and let CT_{semi} be a semi-functional ciphertext w.r.t. CT_{normal}. The abilities of decryption between the key-ciphertext pairs are listed as in Table 1.

Table 1. Decryption ability between different types of keys and ciphertexts. $\surd$ means that a key K_X with type X is able to decrypt a ciphertext CT_Y with type Y, $\times$ means that a key K_X with type X fail to decrypt a ciphertext CT_Y with type Y.

key / ciphertext	K_{normal}	K_{semi}
CT_{normal}	$\surd$	$\surd$
CT_{semi}	$\surd$	$\times$

In the formal security proof, an attack game with an attacker and a challenger is used for an encryption scheme. The encryption scheme is regarded as secure if the attacker cannot win the game with a non-negligible advantage. Using the Dual System Encryption, a sequence of games are needed. Among the games, the first game is a real game and the others are modified games with the semi-functional keys and semi-functional ciphertexts. To prove that an attacker cannot break the game, the challenger provides the last bogus game which is proved unbreakable to the attacker and proves that the attacker cannot distinguish one game from the others. We will introduce the details of games and designing of semi-functional keys and semi-functional ciphertexts in Sect. 4.2.

3 Overview of Escrow-Free HIBE

In this section, we firstly introduce the intuition of our solution to the key escrow problem of HIBE. Then, we briefly describe the components of our scheme. Finally, we present the full security definition by illuminating the IND-ID-CCA game for our escrow-free approach.

3.1 Intuition of Escrow-Free HIBE

The essence of key escrow problem is that the Private Key Generator (PKG) exclusive owns the scheme master key. In order to restrict the power of PKG, we divide the master key into a PKG master key and a set of secret keys.

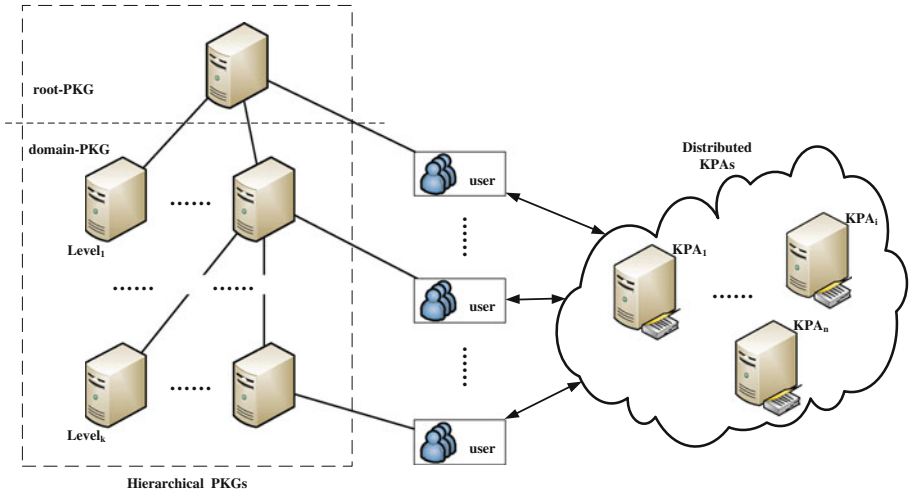

Fig. 1. The infrastructure of our escrow-free HIBE scheme.

We introduce multiple Key Privacy Authorities (KPAs) to keep the partial secret keys. A private key is jointly computed by the PKG and all the KPAs. Based on the assumption that at least one of the KPA is honest, we can keep the privacy of the private key. The infrastructure is showed as in Fig. 1. In our escrow-free HIBE scheme, each private key is generated by a domain PKG and the multiple KPAs. In order to reduce the authentication overhead caused by the KPAs, PKG can generate and assign the user a signature with regard to the its identity. The KPAs verify the signature so as to verify the user's identity.

3.2 Definitions

Our escrow-free hierarchical identity-based encryption scheme consists of four algorithms: Setup, KeyGen, Encrypt and Decrypt.

Setup. The setup algorithm comprises the PKG and KPAs setup stages.

- The PKG takes a security parameter as input and outputs the public parameters $Param_{PKG}$ and a PKG master key MK.
- KPA$_i$ then inputs $Param_{PKG}$ and outputs KPA parameter $Param_{KPA_i}$ as well as a secret key SK_i.

KeyGen. The key generation algorithm takes the PKG master key, multiple secret keys as well as an identity ID $= (ID_1, \ldots, ID_n)$ as input and output the user private key. It also consists of two stages:

- KeyIssue. With the identity, PKG launches the key issuing stage to generate a raw private key, and assigns it to the user.
- KeySec. After the KeyIssue stage, user synchronously asks for key securing from the KPAs and finally get the decrypt key DK.

Encrypt. The encryption algorithm takes the public parameters $Param_{PKG}$, KPA parameters $Param_{KPA_i}(i = 1, \ldots n)$, a message M, and an identity as input and outputs a ciphertext CT.

Decrypt. The decryption algorithm takes the public parameters $Param_{PKG}$, KPA parameters $Param_{KPA_i}(i = 1, \ldots n)$, a ciphertext CT, and a decrypt key DK as input and outputs the message M.

3.3 Security Model

The full security model (IND-ID-CCA) for HIBE schemes is firstly suggested in [3]. We modify the model to present an IND-ID-CCA security for our escrow-free HIBE scheme, which is defined via the following game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$.

Setup. $\mathcal{C}$ runs the PKG and KPA setup algorithms, and gives $\mathcal{A}$ the resulting scheme parameters $Param_{PKG}$ and $Param_{KPA_i}(i = 1, \ldots n)$, keeping the PKG master key MK and KPA secret keys $SK_i(i = 1, \ldots n)$ to itself.

Phase 1. $\mathcal{A}$ issues private key queries and decryption queries.
For a private key query (ID), $\mathcal{C}$ runs the KeyGen algorithm and gives $\mathcal{A}$ the raw private key as well as the KPA key securing factors.

For a decryption query (ID, CT), $\mathcal{C}$ runs the KeyGen algorithm to generate the private key of ID and decrypts the ciphertext CT utilizing the private key.

Challenge. $\mathcal{A}$ gives $\mathcal{C}$ two messages M_0 and M_1, and a challenge identity $\mathrm{ID} = (ID_1, \ldots, ID_n)$. The challenge identity must satisfy the property that no query identity in *Phase 1* is a prefix of it. And the challenge message must not be one of the messages of decryption queries. $\mathcal{C}$ randomly selects $\beta \in \{0, 1\}$ and encrypts M_β with the identity. It sends the ciphertext to $\mathcal{A}$.

Phase 2. This is the same as Phase 1 except that $\mathcal{A}$ cannot query the private key of the challenge identity and private keys of its ancestors.

Guess. $\mathcal{A}$ output a guess β' for β. The advantage of $\mathcal{A}$ is defined to be $Pr[\beta' = \beta] - \frac{1}{2}$.

Definition 1. *We say that the escrow-free hierarchical identity based encryption is secure if no polynomial time adversaries can achieve a non-negligible advantage in the security game.*

4 EF-LW-HIBE Scheme

We build our escrow-free HIBE scheme based on the Lewko-Waters HIBE. Similar to the LW-HIBE, our construction uses composite order groups of order $N = p_1p_2p_3$ and identities in $\mathbb{Z}_N$. Based on the knowledge of Dual System Encryption, we prove that the EF-LW-HIBE scheme is IND-ID-CCA secure.

4.1 Construction

The EF-LW-HIBE consists of the following four algorithms:

Setup. The setup algorithm comprises the PKG setup and KPA setup stages.

- PKG Setup: The PKG chooses a bilinear group G of order $N = p_1p_2p_3$. Let l denote the maximum depth of the HIBE, PKG then randomly chooses $g, h, u_1, \ldots, u_l \in G_{p_1}, X_3 \in G_{p_3}$, and $\alpha_0 \in \mathbb{Z}_N$. PKG publishes the public parameters $Param_{PKG} = \{N, g, h, u_1, \ldots, u_l, X_3, e(g,g)^{\alpha_0}\}$, and keep α_0 as the PKG master key.
- KPA$_i$ Setup: Each key privacy authority randomly chooses $\alpha_i \in \mathbb{Z}_N$. It takes the $Param_{PKG}$ and α_i as input and computes $e(g,g)^{\alpha_i}$. KPA$_i$ publishes parameter $Param_{KPA_i} = \{e(g,g)^{\alpha_i}\}$, and keeps α_i as KPA secret key.

KeyGen($d_{\mathrm{ID}_{|j-1}}$, ID). The key generation algorithm comprises the key issuing stage by PKG and the key securing stage by KPAs.

- KeyIssue: To generate a private key $\nabla d_{\mathrm{ID}} = (K_1, \nabla K_2, E_{j+1}, \ldots, E_l)$ for identify ID=$(ID_1, \ldots, ID_j)$ $(j \leqslant l)$, the key generation algorithm of PKG picks a random $r \in \mathbb{Z}_N$, random elements $R_3, R'_3, R_{j+1}, \ldots, R_l$ of G_{p3}, and outputs: $K_1 = g^r R_3$, $\nabla K_2 = g^{\alpha_0}\left(u_1^{ID_1} \cdots u_j^{ID_j} h\right)^r R'_3$, $E_{j+1} = u_{j+1}^r R_{j+1}$, $\ldots$, $E_l = u_l^r R_l$.

  We refer to the private key generated by KeyIssue algorithm as a raw private key. Actually, the raw private key for ID can be generated by just given a raw private key for $\mathrm{ID}_{|j-1} = (ID_1, \ldots, ID_{j-1})$ as required. Let $(K'_1, \nabla K'_2, E'_j, \ldots, E'_l)$ be the raw private key for $\mathrm{ID}_{|j-1}$. To generate the private key for ID, the algorithm picks $r' \in \mathbb{Z}_N$ and $\tilde{R}_3, \tilde{R}'_3, \tilde{R}_{j+1}, \ldots, \tilde{R}_l \in G_{p_3}$ randomly. The raw private key of ID can be computed as:

  $$K_1 = K'_1 g^{r'} \tilde{R}_3 = g^{r+r'} R_3 \tilde{R}_3,\ \nabla K_2 = g^{\alpha_0}\left(u_1^{ID_1} \cdots u_j^{ID_j} h\right)^{r+r'} R'_3 R_j^{ID_j} \tilde{R}'_3$$
  $$E_{j+1} = E'_{j+1} u_{j+1}^{r'} \tilde{R}_{j+1} = u_{j+1}^{r+r'} R_{j+1} \tilde{R}_{j+1},\ \ldots,\ E_l = E'_l u_l^{r'} \tilde{R}_l = u_l^{r+r'} R_l \tilde{R}_l.$$

  This raw private key is a properly raw private key for ID=$(ID_1, \ldots, ID_j)$.
- KeySec: With the raw private key, user asks for key securing by the KPAs. Each KPA$_i$ randomly chooses $\hat{R}_i \in G_{p_3}$ and compute the securing factor $g^{\alpha_i}\hat{R}_i$. User retrieves the securing factors from all the KPAs and compute a complete private key $d_{\mathrm{ID}} = \{K_1, K_2, E_{j+1}, \ldots, E_l\}$, where

  $$K_2 = \nabla K_2 g^{\alpha_1}\hat{R}_1 \cdots g^{\alpha_n}\hat{R}_n = g^{\sum_{i=0}^n \alpha_i}\left(u_1^{ID_1} \cdots u_j^{ID_j} h\right)^r R'_3 \textstyle\prod_1^n \hat{R}_i$$

  Note that, key securing of each KPA can be synchronously implemented.

Encrypt($M, (ID_1, \ldots, ID_j)$). Given the message M and an identity, a user encrypt the message with PKG and KPA parameters. It chooses $s \in \mathbb{Z}_N$ randomly and generates the ciphertext $CT = \{C_0, C_1, C_2\}$. There are
$C_0 = Me(g,g)^{\sum_{i=0}^n \alpha_i s}$, $C_1 = \left(u_1^{ID_1} \cdots u_j^{ID_j} h\right)^s$, $C_2 = g^s$.

Decrypt(d_{ID}, CT)**.** The decryption algorithm can decrypt the message by computing the blinding factor:

$$\frac{e(K_2, C_2)}{e(K_1, C_1)} = \frac{e(g,g)^{\sum_{i=0}^{n} \alpha_i s} e(u_1^{ID_1} \cdots u_j^{ID_j} h, g)^{rs}}{e(g, u_1^{ID_1} \cdots u_j^{ID_j} h)^{rs}} = e(g,g)^{\sum_{i=0}^{n} \alpha_i s}.$$

Note that, some bilinear computation results in identities of G_T due to the orthogonality property of $G_{p_1}, G_{p_2}, G_{p_3}$.

4.2 Security Proofs

We prove full security of EF-LW-HIBE utilizing the Dual System Encryption. As described above, the proof utilizes the semi-functional key and semi-functional ciphertext, and relies on a sequence of security games. We design the semi-functional key and ciphertext as follows.

- **Semi-functional Ciphertext.** Let (C_0', C_1', C_2') denote the normal ciphertext generated by the encryption algorithm. Set $C_0 = C_0'$, $C_1 = C_1' g_2^{x z_c}$ and $C_2 = C_2' g_2^{x}$, where g_2 is a generator of the subgroup G_{p2}, and $x, z_c \in \mathbb{Z}_N$ are chosen in random. (C_0, C_1, C_2) is referred to as a semi-functional ciphertext.
- **Semi-functional Key.** Let $(K_1', K_2', E_{j+1}', \ldots, E_l')$ denote the normal key generated by the key generation algorithm. Set $K_1 = K_1' g_2^{\gamma}$, $K_2 = K_2' g_2^{\gamma z_k}$, $E_{j+1} = E_{j+1}' g_2^{\gamma z_{j+1}}$, …, $E_l = E_l' g_2^{\gamma z_l}$ where g_2 is a generator of the subgroup G_{p2}, and $\gamma, z_k, z_{j+1}, \ldots, z_l \in \mathbb{Z}_N$ are chosen in random. (K_1, K_2) is referred to as a semi-functional key.

To prove the security of our HIBE scheme, we introduce a series of indistinguishable games, illustrated as in Fig. 2. The left side of Fig. 2 are the lemmas that ensure the indistinguishability of each two attack games. We describe and prove the four lemmas as follows:

Lemma 1. *Suppose there exists an algorithm $\mathcal{A}$ that can distinguish $Game_{Real}$ and $Game_{Restricted}$ with advantage ε. Then we can build an algorithm with advantage $\geqslant \frac{\varepsilon}{2}$ in breaking either Assumption 1 or Assumption 2.*

Proof. Since we define the $Game_{Restricted}$ as [15] does, we can prove that the games $Game_{Real}$ and $Game_{Restricted}$ are indistinguishable following the same procedures. Details can be found in the proof of *Lemma* 5 in [15].

Lemma 2. *Suppose there exists an algorithm $\mathcal{A}$ that can distinguish $Game_0$ and $Game_{Restricted}$ with advantage ε. Then we can build an algorithm with advantage ε in breaking Assumption 1.*

Proof. $\mathcal{B}$ is given g, X_3, T. To simulate $\text{Game}_{Restricted}$ or Game_0 with $\mathcal{A}$, $\mathcal{B}$ first chooses random exponents $\alpha_0, a_1, \ldots, a_l, b \in \mathbb{Z}_N$ and sets $g = g, u_i = g^{a_i}$ for i from 1 to l and $h = g^b$. PKG parameters $\{N, g, h, u_1, \ldots, u_l, X_3, e(g,g)^{\alpha_0}\}$ and KPA parameters $\{e(g,g)^{\alpha_i}\}$ $(i = 1 \ldots n)$ are send to $\mathcal{A}$.

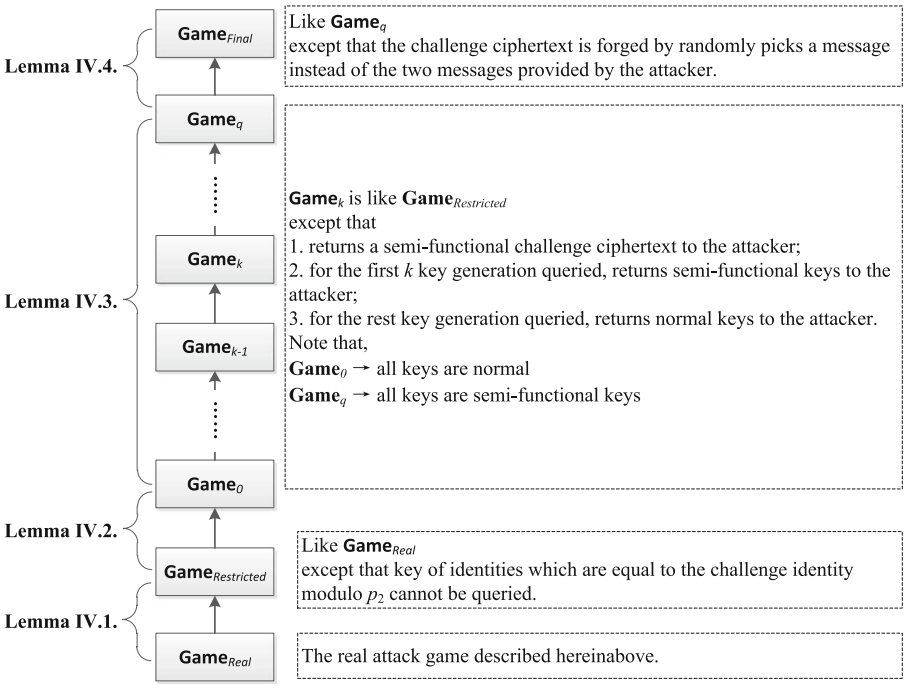

Fig. 2. A series of indistingusihable games. The indistinguishability between two games is ensured by a corresponding lemma. q denotes the maximum number of key queries the attacker makes, and p_2 is one of the prime factor of the composite order.

When $\mathcal{A}$ queries a key for identity $(ID_1, \ldots, ID_j)$, $\mathcal{B}$ chooses random exponents $r, t, w, v_{j_1}, \ldots, v_l \in \mathbb{Z}_N$ and returns key: $K_1 = g^r X_3^t$,
$K_2 = g^{\sum_{i=0}^{n} \alpha_i} (u_1^{ID_1} \cdots u_j^{ID_j} h)^r X_3^w$, $E_{j+1} = u_{j+1}^r X_3^{v_{j+1}}, \ldots, E_l = u_l^r X_3^{v_l}$.

In challenge phase, $\mathcal{A}$ sends $\mathcal{B}$ two messages M_0, M_1 and a challenge identity $(ID_1^*, \ldots, ID_j^*)$. $\mathcal{B}$ randomly chooses $\beta \in \{0, 1\}$, and forms ciphertext:
$C_0 = M_\beta e(T, g)^{\sum_{i=0}^{n} \alpha_i}$, $C_1 = T^{a_1 ID_1^* + \cdots + a_j ID_j^* + b}$, $C_2 = T$.
If $T \in G_{p_1 p_2}$, then this is a semi-functional ciphertext with $z_c = a_1 ID_1^* + \cdots + a_j ID_j^* + b$. If $T \in G_{p_1}$, this is a normal ciphertext. As supposed, $\mathcal{A}$ is able to distinguish the semi-functional and normal ciphertext. Therefore, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish T. That is, it can break Assumption 1.

Lemma 3. *Suppose there exists an algorithm $\mathcal{A}$ that can distinguish $Game_{k-1}$ and $Game_k$ with advantage ε. Then we can build an algorithm with advantage ε in breaking Assumption 2.*

Proof. $\mathcal{B}$ is given $g, X_1 X_2, X_3, Y_2 Y_3, T$. To simulate Game_{k-1} or Game_k with $\mathcal{A}$, $\mathcal{B}$ first chooses random exponents $a_1, \ldots, a_l, b \in \mathbb{Z}_N$. and sets the public parameters of PKG as $g = g$, $u_1 = g^{a_1}$, …, $u_l = g^{a_l}$, $h = g^b$, $e(g, g)^{\alpha_0}$, public parameters of KPA_i as $e(g, g)^{\alpha_i}$. Public parameters are sent to $\mathcal{A}$.

When $\mathcal{A}$ requests the i^{th} key for identity $(ID_1, \ldots, ID_j)$.

- If $i < k$, $\mathcal{B}$ generates a semi-functional key. It chooses random exponents $r, z, t, z_{j+1}, \ldots, z_l \in \mathbb{Z}_N$ and sets:
  $K_1 = g^r(Y_2Y_3)^t$, $K_2 = g^{\sum_{i=0}^{n}\alpha_i}(u_1^{ID_1}\cdots u_j^{ID_j}h)^r(Y_2Y_3)^z$,
  $E_{j+1} = u_{j+1}^r(Y_2Y_3)^{z_{j+1}}, \ldots, E_l = u_l^r(Y_2Y_3)^{z_l}$.
  Note that this is a properly distributed semi-functional key with $g_2^\gamma = Y_2^t$.
- If $i = k$, $\mathcal{B}$ lets $z_k = a_1ID_1^* + \cdots + a_jID_j^* + b$, chooses random exponents $w_k, w_{j+1}, \ldots, w_l \in \mathbb{Z}_N$, and sets:
  $K_1 = T, K_2 = g^{\sum_{i=0}^{n}\alpha_i}T^{z_k}X_3^{w_k}$, $E_{j+1} = T^{a_{j+1}}X_3^{w_{j+1}}, \ldots, E_l = T^{a_l}X_3^{w_l}$.
  If $T \in G_{p_1p_3}$, this is a normal key with g^r equal to the G_{p_1} part of T.
  If $T \in G$, this is a semi-functional key.
- If $i > k$, $\mathcal{B}$ generates normal keys by calling the usual key generation algorithm.

In challenge phase, $\mathcal{A}$ sends $\mathcal{B}$ two messages M_0, M_1 and a challenge identity $(ID_1^*, \ldots, ID_j^*)$. $\mathcal{B}$ randomly chooses $\beta \in \{0, 1\}$, and forms ciphertext:
$C_0 = M_\beta e(X_1X_2, g)^\alpha$, $C_1 = (X_1X_2)^{a_1ID_1^*+\cdots+a_jID_j^*+b}$, $C_2 = X_1X_2$.

We notice that this sets $g^s = X_1$ and $z_c = a_1ID_1^* + \cdots + a_jID_j^* + b$. Since the k^{th} key is not a prefix of the challenge key modulo p_2, z_k and z_c will seem randomly distributed to $\mathcal{A}$. Though it is hidden from $\mathcal{A}$, this relationship between z_c and z_k is crucial: if $\mathcal{B}$ attempts to test itself whether key k is semi-functional by creating a semi-functional ciphertext for this identity and trying to decrypt, then decryption will work whether key k is semi-functional or not, because $z_c = z_k$. In other words, the simulator can only create a nominally semi-functional key k. If $T \in G_{p_1p_3}$, then $\mathcal{B}$ has properly simulated Game_{k-1}. If $T \in G$, then $\mathcal{B}$ has properly simulated Game_k. As supposed, $\mathcal{A}$ is able to distinguish Game_{k-1} and Game_k. Therefore, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish between these possibilities for T. That is, it can break Assumption 2.

Lemma 4. *Suppose there exists an algorithm $\mathcal{A}$ that can distinguish $Game_q$ and $Game_{Final}$ with advantage ε. Then we can build an algorithm with advantage ε in breaking Assumption 3.*

Proof. $\mathcal{B}$ is given $g, g^\alpha X_2, X_3, g^sY_2, Z_2, T$. To simulate Game_q or Game_{Final} with $\mathcal{A}$, $\mathcal{B}$ first chooses random exponents $a_1, \ldots, a_l, b \in \mathbb{Z}_N$ and sets the public parameters of PKG as $g = g$, $u_1 = g^{a_1}$, …, $u_l = g^{a_l}$, $h = g^b$, $e(g, g)^{\alpha_0} = e(g^{\alpha_0}X_2, g)$. It also randomly choose $\alpha_1, \ldots, \alpha_n \in \mathbb{Z}_N$ and sets public parameters of KPA_i as $e(g, g)^{\alpha_i} = e(g^{\alpha_i}X_2, g)$. Public parameters are sent to $\mathcal{A}$.

When $\mathcal{A}$ requests key for identity $(ID_1, \ldots, ID_j)$, $\mathcal{B}$ chooses random exponents $c, r, t, w, z, z_{j+1}, \ldots, z_l, w_{j+1}, \ldots, w_l \in \mathbb{Z}_N$ and returns a semi-functional key: $K_1 = g^rZ_2^zX_3^t$, $K_2 = g^{\alpha_0}X_2Z_2^c(u_1^{ID_1}\cdots u_j^{ID_j}h)^rX_3^w$,
$E_{j+1} = u_{j+1}^rZ_2^{z_{j+1}}X_3^{w_{j+1}}, \ldots, E_l = u_l^rZ_2^{z_l}X_3^{w_l}$.

Note that $\mathcal{B}$ returns a raw private key to $\mathcal{A}$, which is also a properly distributed semi-functional key. And $\mathcal{A}$ cannot distinguish it from the complete key. In challenge phase, $\mathcal{A}$ sends $\mathcal{B}$ two messages M_0, M_1 and a challenge identity $(ID_1^*, \ldots, ID_j^*)$. $\mathcal{B}$ chooses $\beta \in \{0, 1\}$ randomly, and forms ciphertext:

$C_0 = M_\beta T$, $C_1 = (g^s Y_2)^{a_1 ID_1^* + \cdots + a_j ID_j^* + b}$, $C_2 = g^s Y_2$.

This sets $z_c = a_1 ID_1^* + \cdots + a_j ID_j^* + b$. We notice that the value of z_c only matters modulo p_2, whereas $u_1 = g^{a_1}, \ldots, u_l = g^{a_l}$, and $h = g^b$ are elements of G_{p_1}. So when $a_1, \ldots, a_l$ and b are chosen randomly modulo N, there is no correlation between the values of $a_1, \ldots, a_l, b$ modulo p_1 and the value $z_c = a_1 ID_1^* + \cdots + a_j ID_j^* + b$ modulo p_2.

If $T = e(g, g)^{\alpha s}$, then this is a properly distributed semi-functional ciphertext with message M_β. If T is a random element of G_T, then this is a semi-functional ciphertext with a random message. As supposed, $\mathcal{A}$ is able to distinguish these two kinds of ciphertext. Therefore, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish between these possibilities for T. That is, it can break Assumption 3.

Theorem 1. *If Assumptions 1, 2, and 3 hold, then our EF-LW-HIBE scheme is secure.*

Proof. If Assumptions 1, 2, and 3 hold, $Game_{Real}$ is indistinguishable from $Game_{Final}$ according to the *Lemma* 1 to 4. $Game_{Final}$ information-theoretically hiding the value of β is the de facto game provided to the attacker. Therefore, the attacker can attain no advantage in breaking the EF-LW-HIBE scheme.

5 Conclusion

In this work, we introduced a provably-secure solution to the key escrow problem of PKGs in HIBE scheme. The main idea of the solution is to restrict the power of PKGs by employing multiple Key Privacy Authorities (KPAs) which partitions the main secret for generating private keys and each KPA obtains a portion of the secret. According to the idea, we presented the escrow-free HIBE scheme based on the LW-HIBE. We proved the full security of EF-LW-HIBE scheme, utilizing the methodology of Dual System Encryption. Although the PKG-KPAs model was instantiated into a specific scheme in this work, it can also be applied to other HIBE schemes.

Acknowledgment. This research is supported in part by the project of the National High Technology Research and Development Program of China(863 Program) No. 2011AA01A103; the program of Changjiang Scholars and Innovative Research Team in University (No. IRT1012); Science and Technology Innovative Research Team in Higher Educational Institutions of Hunan Province (network technology); and Hunan Province Natural Science Foundation of China (11JJ7003).

References

1. Blazy, O., Kiltz, E., Pan, J.: (Hierarchical) identity-based encryption from affine message authentication. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 408–425. Springer, Heidelberg (2014)

2. Boneh, D., Boyen, X.: Efficient selective-ID secure identity-based encryption without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
3. Boneh, D., Boyen, X., Goh, E.-J.: Hierarchical identity based encryption with constant size ciphertext. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (2005)
4. 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)
5. Boneh, D., Goh, E.-J., Nissim, K.: Evaluating 2-DNF formulas on ciphertexts. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 325–341. Springer, Heidelberg (2005)
6. Cao, D., Wang, X.F., Wang, F., Hu, Q.L., Su, J.S.: Sa-ibe: a secure and accountable identity-based encryption scheme. Dianzi Yu Xinxi Xuebao (J. Electron. Inf. Technol.) **33**(12), 2922–2928 (2011)
7. Chen, J., Wee, H.: Fully, (almost) tightly secure IBE and dual system groups. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 435–460. Springer, Heidelberg (2013)
8. Chow, S.S.M.: Removing escrow from identity-based encryption. In: Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 256–276. Springer, Heidelberg (2009)
9. Fu, S., Wang, D., Xu, M., Ren, J.: Cryptanalysis of remote data integrity checking protocol proposed by L. Chen for cloud storage. IEICE Trans. **97**–**A**(1), 418–420 (2014). http://search.ieice.org/bin/summary.php?id=e97-a_1_418
10. Gentry, C., Silverberg, A.: Hierarchical ID-based cryptography. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 548–566. Springer, Heidelberg (2002)
11. Huang, K., Xian, M., Fu, S., Liu, J.: Securing the cloud storage audit service: defending against frame and collude attacks of third party auditor. IET Commun. **8**(12), 2106–2113 (2014). http://dx.doi.org/10.1049/iet-com.2013.0898
12. Kamara, S., Lauter, K.: Cryptographic cloud storage. In: Sion, R., Curtmola, R., Dietrich, S., Kiayias, A., Miret, J.M., Sako, K., Sebé, F. (eds.) RLCPS, WECSR, and WLC 2010. LNCS, vol. 6054, pp. 136–149. Springer, Heidelberg (2010)
13. Kate, A., Goldberg, I.: Distributed private-key generators for identity-based cryptography. In: Garay, J.A., De Prisco, R. (eds.) SCN 2010. LNCS, vol. 6280, pp. 436–453. Springer, Heidelberg (2010)
14. Lee, B., Boyd, C., Dawson, E., Kim, K., Yang, J., Yoo, S.: Secure key issuing in id-based cryptography. In: Proceedings of the Second Workshop on Australasian Information Security, Data Mining and Web Intelligence, and Software Internationalisation, vol. 32, pp. 69–74. Australian Computer Society, Inc. (2004)
15. Lewko, A., Waters, B.: New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 455–479. Springer, Heidelberg (2010)
16. 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)
17. Wang, C., Chow, S.S., Wang, Q., Ren, K., Lou, W.: Privacy-preserving public auditing for secure cloud storage. IEEE Trans. Comput. **62**(2), 362–375 (2013)

18. Waters, B.: Dual system encryption: realizing fully secure IBE and HIBE under simple assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (2009)
19. Zeng, W., Zhao, Y., Ou, K., Song, W.: Research on cloud storage architecture and key technologies. In: Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, pp. 1044–1048. ACM (2009)