Identity-Based Hierarchical Strongly Key-Insulated Encryption and Its Application

Yumiko Hanaoka[*], Goichiro Hanaoka[†], Junji Shikata[‡] and Hideki Imai[§]

May 31, 2005

Abstract

In this paper, we discuss non-interactive updating of decryption keys in identity-based encryption (IBE). IBE is a public key cryptosystem where a public key is an arbitrary string. In practice, key revocation is a necessary and inevitable process and IBE is no exception when it comes to having to manage revocation of decryption keys without losing its merits in efficiency. Our main contribution of this paper is to propose novel constructions of IBE where the decryption key can be renewed without having to make changes to its public key, i.e. user's identity. We achieve this by tactfully extending the hierarchical IBE (HIBE). Regarding security, we address semantic security against adaptive chosen cipher-text attack for a very strong attack environment that models all possible types of key exposures in the random oracle model. Straightforward extension of the HIBE, however, does not achieve our goal and such scheme is completely insecure under our attack model. In addition to this, we show method of constructing (partially collusion resistant) HIBE from arbitrary IBE in the random oracle model. By combining these results, we can construct an IBE with non-interactive key update from only an arbitrary IBE.

1 Introduction

Background. As to our best of knowledge, current public key infrastructures involve complex construction of *certification authorities* (CA), consequently requiring expensive communication and computation costs for certificate verification. In 1984, Shamir introduced an innovative concept called, *identity-based encryption* (IBE) [27](later actualized by [7]), where any public key is determined as an arbitrary string, e.g. user's name, e-mail address, etc. and simplifies certificate management in public key infrastructures.

In this paper, we address non-interactive updating of user's decryption key in IBE. Revocation and renewal of decryption keys is a necessary process carried out in practical situation, and designing of IBE that can renew and update decryption keys without any loss in its merits in efficiency has considerable implications in the practical crypto-infrastructure. In conventional public key schemes, certification revocation list (CRL) is utilized to minimize the damage caused by key compromization. Users can become aware of other users' revoked keys by referring to the CRLs, and abort it, if necessary. In IBE, however, some caution must be taken and straightforward implementation of CRL is not the sensible solution, as it means terminating the link between the public key and the user's identity, and loosing one of principal advantages of IBE. One of suitable application of IBE is of a mobile phone scenario, in which case, phone number represents the user identity. It will be both simple and convenient for the mobile

[*] NTT DoCoMo, Inc. yamamotoyumi@nttdocomo.co.jp

[†] National Institute of Advanced Industrial Science and Technology. hanaoka-goichiro@aist.go.jp

[‡] Graduate School of Environment and Information Sciences, Yokohama National University.

[§] Institute of Industrial Science, the University of Tokyo.

phone users to be able to communicate and identify each other by their phone numbers only. Yet, the problem in IBE system arises at time when the user needs to renew his key, and it is necessary for him to be able to renew the key without making changes to his phone number. Solving such critical problem of IBE in the use in practical settings is our main concern and is the core subject of our discussion.

Our Results. Our main contribution of this paper is to study and propose an efficient method to renew and revoke decryption keys in IBE system. We begin our discussion by looking into the difficulty of solving this issue when considering the conventional model of IBE as the starting point. We propose a new IBE model and show a generalized method that allows efficient renewing of the decryption keys. Based on our proposed model, we construct an IBE that can update the decryption keys non-interactively, that is, *allow user to renew and update his key without help from the central authority, and most importantly, without having to change his identity.* In our scheme, similarly to [14], we assume a *private device* (PD) which is isolated from the main hardware where the actual decryption takes place. PD is not connected to the network except for time the decryption key needs to be updated and the *helper key* stored in the PD is used to update the key at each fixed time period. All secret operations are done by the user alone. Our proposed scheme can be regarded as the first construction of an identity-based version of *strongly secure* key insulated encryption [14]. Here, we mean "strongly" by a system guaranteeing its security even when the PD is physically compromised. Our scheme is different from [14] in a way that the PD is divided into multiple levels forming a hierarchical structure and its security is improved.

In brief, our proposed scheme can be said as an extension of hierarchical identity-based encryption schemes (HIBE) [26, 25]. Straightforward extension of HIBE, however, will be completely vulnerable for our attack model. In this paper, we propose two different secure constructions of IBE that can update the decryption key non-interactively. The first construction is a generic construction built from HIBE where an arbitrary (chosen plaintext secure) HIBE is used to construct a chosen ciphertext secure IBE with non-interactive key update. Also, the underlying assumption of such scheme can be flexibly selected depending on the requirement of the system. As a by-product, this method can be further applied to generically construct a (standard) strongly secure key-insulated encryption from arbitrary (H)IBE and standard public key encryption allowing unlimited number of key updates. The second construction is a specific construction of IBE with non-interactive key update. Specific construction is more efficient than the generic construction. In addition to these constructions, we also show a technique to construct a (partially collusion resistant) HIBE from an arbitrary IBE. This result can further be applied to the above generic construction to construct an IBE with non-interactive key update from only an arbitrary IBE. Note that we mean "partial collusion resistant" in a sense that we argue based on the security definition in [26] and not of [25]. Security of our schemes is proved in the random oracle model.

Applications: Mobile Phone Scenario. Now let's consider the suitability of assuming a private device in the mobile phone scenario (see also **Background.**). At first glance, it seems like a hassle to having to bring the PD whenever you need to update your decryption key, although, it is not as you might think so. As a mobile phone user, it is your routine job to re-charge your battery every now and then. Now, assume a PD-BC (i.e. a private device that can function also as a battery charger). PD-BC can provide a convenient mean to update the decryption key whenever you have to re-charge the battery (which you have to do it anyways). Our IBE system can also guarantee the security even if the PD-BC is compromised. Mobile phone is just one of many attractive applications of IBE and any mobile device users (e.g. laptop PC users) who are in high risks of losing their decryption keys can benefit from this system. Our IBE with non-interactive key update can further improve its security by stratifying the PD. To simplify things, we assume the PD to be hierarchically structured into two levels: let the lower level PD be the battery charger (therefore, is the PD-BC) which stores the helper key used to update the decryption key as frequent as every day, and let the higher level PD be the one that updates the helper key (of the lower level PD) every once in a while (maybe every 2-3 months). Therefore, lower level PD must kept in places more handy (i.e. at home, work place) while the higher level PD (used less frequently) to be kept somewhere not as convenient but physically safer (i.e. safe). Our IBE system can guarantee the security even when any of the level PD is compromised even of the higher one.

Related Works. The problem of revocability of private keys in identity-based schemes was initially discussed by Shinozaki, Itoh, Fujioka and Tsujii [28]. It, however, required prior communication for revocation and therefore, did not show advantage over conventional public key schemes in terms of cost efficiency, and also required prior interaction between the user and the certificate authority. Furthermore, their scheme was specific to Fiat-Shamir identification scheme [20, 21] and could not generally be applied to identity-based schemes. Recently, Baek and Zheng [2] showed an application of the threshold decryption method to IBE. This, does decrease the possibility of getting the key to be exposed, however, does not deal with the case after the key exposure has actually occurred. In [17], Dodis and Yung proposed an interesting idea that refreshes the private keys in HIBE. Their scheme provides a solution to the problem of *gradual* key exposure in which the private key is assumed to slowly compromise over time. Boneh and Franklin in their paper ([7], Section 1.1.1) showed the first generalized method for key revocation in IBE schemes. In their scheme, a privileged Private Key Generator (PKG) generates each user's decryption key and its corresponding public key. Public key is set to be the concatenation of user identity and fixed length of time the key is available, e.g. "`recipient@xxx.xxx || 2004.01.01-2004.12.31`". In such a setting, the public key, despite of whether it is revoked or not, is renewed regularly by the PKG, and also, the renewal interval must be set short (e.g. per day) to alleviate the damage caused by key exposure. Therefore, having to set the interval short and require frequent contact with the PKG implies increase in the total communication and computation cost, consequently, loosing one of primary advantages of IBE (i.e. low costs in communication and computation). Further, it needs to work out a way to establish a secure channel between the PKG and the user. For instance, it needs to compensate for additional transmission for key issuing and also has to deal with complicated transactions if the secret information used to setup the secure channel is ever exposed. Moreover, forward security must be considered. It is, hence, not desirable to have to require frequent communication via secure channel with the PKG in IBE as it implicates loss of primary advantages of IBE.

While, on the other hand, as a solution to key exposure and revocation problem in conventional public key systems, Dodis, Katz, Xu and Yung [14] proposed a scheme called *key-insulated encryption.* As said earlier, this scheme also assumes a PD in which it stores the *helper key.* The helper key assists the user to renew his decryption key by generating secret necessary to update the key. Here, the public key is fixed. In [15, 16], Dodis, Franklin, Katz, Miyaji and Yung further improved [14] with additional property as forward security. Notice that being able to renew the decryption key without having to make any changes to the corresponding public key as in the key-insulated encryption scheme, is the very technique, also desired in IBE. Possible harmonization of the advantages of the two schemes, of constructing an identity-based version of a (strongly secure) key-insulated encryption scheme, has never been carried out before. There has also never been a construction built of a hierarchical version of key-insulated encryption where the PD is organized in a hierarchical tree structure. Besides the related works shown so far, there are other interesting researches done on the topic of key exposure and revocation as well, for example, [24, 1], but are both looked from the conventional PKI perspective.

We mentioned earlier that our IBE with non-interactive key update is constructed by extending the HIBE [26, 25]. HIBE is a powerful cryptographic tool and also forms the basis of various cryptographic techniques, e.g. [11]. However, the only methods known to construct HIBE [26, 25, 11, 4, 6] are ones that require specific assumptions in elliptic curve cryptography, e.g. the bilinear Diffie-Hellman (BDH) problem [7, 8] as the underlying assumption and therefore lacks flexibility in selecting the underlying assumption. (While for IBE, besides BDH, there could also be a construction based on quadratic residuosity problem [10].) There is an open problem for a generic construction of HIBE based on arbitrary

IBE and is one of important research topics in this area.

2 Model and Definitions

Overview of the Model. Before we start discussing the details of the actual construction of our IBE scheme, recall earlier how we said it was impossible to construct an IBE that allows an essential property as key revocation if based on the model of conventional IBE. To be more specific, it is impossible, based on the conventional IBE model, for the user to *immediately* revoke and renew his decryption key *only* at time he needs to renew the decryption key without loosing the advantage of IBE in terms of communication cost, as in the conventional IBE, public parameter distributed at system set up phase and user's identity are the only parameters used to encrypt message. Consequently, a new model for IBE is required to add a key renewal property.

As already mentioned, [7], showed the first generalized method for key revocation based on the conventional IBE model. Their scheme, however, required to establish a secure channel between the user and the PKG which also needed to be available at all times. Moreover, the burden on the PKG was heavy and required the PKG to periodically renew the users' decryption keys at fixed and also frequent time intervals. Their model is simple and generally has no problem in using, and may even be practical for some applications. However, there are other cases where their assumption is neither preferred nor available.

In our proposing model of IBE that allows renewal and updating of decryption keys without losing the advantage of IBE in terms of communication cost (i.e., non-interactive), we introduce a *private device* (PD) which stores the helper key used to renew the key. In our model, decryption keys are renewed at regular time intervals without having to require interactions with other entities. We further improved the security by hierarchically constructing the PD, letting the helper key of each level to be renewed using the helper key of one level higher (See **Applications: Mobile Phone Scenario** in Sec. 1.). In fact, our model can be regarded as both hierarchical and identity-based extension of key-insulated encryption [14]. Similarly to [14], we address *random-access key updating*, namely, it allows one-step renewal of current decryption key to any of the decryption keys of any time period (even the past keys). Random-access key updating lets any ciphertext of any time period to be decrypted at any time.

Model. We assume the user's PD to be structured hierarchically into ℓ-levels, and for $i = 1, \cdots, \ell$, *i-th level helper key* is stored in the i-th level. If considering the mobile phone scenario, mobile phone, or the 0-level PD, is where the actual decryption takes place and the decryption key is stored. The data generated using the *i-th level helper key* is used to renew the $(i-1)$-th level helper key for $i = 2, \cdots, \ell$. And, helper key in the battery charger (PD-BC), or the 1st-level helper key, is used to renew the decryption key. For simplicity, we let $\ell = 2$, where the 1st- and 2nd-level PDs correspond to the battery charger and device that updates the PD-BC helper key, respectively. (Note that our scheme can easily be generalized for arbitrary $\ell \geq 1$.)

Now, let $T_0(\cdot)$ and $T_1(\cdot)$ be functions which map *time* to corresponding time periods for decryption key and 1st-level helper key, respectively. For example, we have $T_0(\texttt{2005/Aug./26th/17:00}) = \texttt{2005/Aug./26th}$ and $T_1(\texttt{2005/Aug./26th/17:00}) = \texttt{2005/Jul.-Sep.}$ assuming the decryption key to be updated every day and the 1st-level helper key every 2-3 months. In addition, let $T_2(\cdot)$ be function such that for all $\texttt{time}$, $T_2(\texttt{time}) = 0$. For our model, at time $\texttt{time}$, user updates his decryption key if 1st-level helper key is valid for time period $T_1(\texttt{time})$ and the 1st-level helper key can be updated at any time. Def. 1 formally addresses this, and Fig. 1 illustrates the key-updating mechanism.

Definition 1 (IKE) A 2-level *identity-based key-insulated encryption scheme (IKE)* IKE consists of 8 algorithms: $\mathsf{IKE} = (\mathsf{PGen}_{\mathsf{IKE}}, \mathsf{Gen}_{\mathsf{IKE}}, \Delta\text{-}\mathsf{Gen}^i_{\mathsf{IKE}}, \mathsf{Upd}^i_{\mathsf{IKE}}\ (i = 1, 2), \mathsf{Enc}_{\mathsf{IKE}}, \mathsf{Dec}_{\mathsf{IKE}})$ and each of them are described as follows.

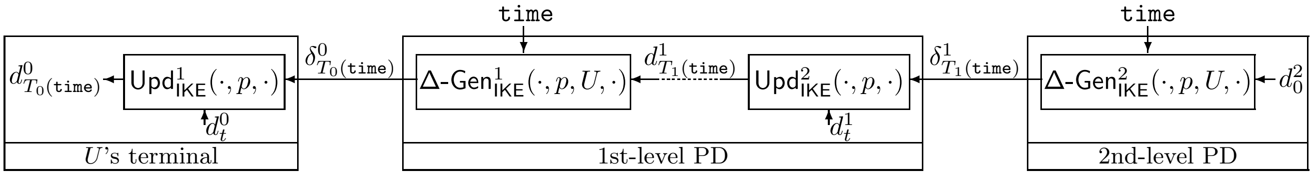
time
time
$d^0_{T_0(\texttt{time})}$
$\mathsf{Upd}^1_{\mathsf{IKE}}(\cdot,p,\cdot)$
d^0_t
$\delta^0_{T_0(\texttt{time})}$
$\Delta\text{-}\mathsf{Gen}^1_{\mathsf{IKE}}(\cdot,p,U,\cdot)$
$d^1_{T_1(\texttt{time})}$
$\mathsf{Upd}^2_{\mathsf{IKE}}(\cdot,p,\cdot)$
d^1_t
$\delta^1_{T_1(\texttt{time})}$
$\Delta\text{-}\mathsf{Gen}^2_{\mathsf{IKE}}(\cdot,p,U,\cdot)$
d^2_0
U's terminal
1st-level PD
2nd-level PD

Figure 1: Key-Updating in IKE.

$\mathsf{PGen}_{\mathsf{IKE}}$. The *public-parameter generation algorithm* $\mathsf{PGen}_{\mathsf{IKE}}(1^k)$, where k is the security parameter, outputs master key s and public parameter p. Note that $\mathsf{PGen}_{\mathsf{IKE}}$ and $\mathsf{Gen}_{\mathsf{IKE}}$ are used by the PKG only.
$\mathsf{Gen}_{\mathsf{IKE}}$. The *user-secret generation algorithm* $\mathsf{Gen}_{\mathsf{IKE}}$ takes s, p and user's identity U as inputs, and outputs U's initial private keys (d^0_0, d^1_0, d^2_0), where d^0_0 is the U's initial decryption key. d^i_0 $(i = 1, 2)$ are each stored in U's i-th level PD as initial i-th helper key.
$\Delta\text{-}\mathsf{Gen}^i_{\mathsf{IKE}}$. A helper key stored in the 1st- (resp. 2nd-) level PD and $\Delta\text{-}\mathsf{Gen}^1_{\mathsf{IKE}}$ (resp. $\Delta\text{-}\mathsf{Gen}^2_{\mathsf{IKE}}$) are used to generate the data required to renew the decryption key (resp. the 1st-level helper key). More specifically, for $i = 1, 2$, the *key-update information generation algorithm* $\Delta\text{-}\mathsf{Gen}^i_{\mathsf{IKE}}$ takes d^i_t, p, U and $\texttt{time}$ as inputs, and outputs key-update information $\delta^{i-1}_{T_{i-1}(\texttt{time})}$ only if $t = T_i(\texttt{time})$.
$\mathsf{Upd}^i_{\mathsf{IKE}}$. U's decryption key (resp. U's 1st-level helper key), key-update information $\delta^0_{T_0(\texttt{time})}$ (resp. $\delta^1_{T_1(\texttt{time})}$) and $\mathsf{Upd}^1_{\mathsf{IKE}}$ (resp. $\mathsf{Upd}^2_{\mathsf{IKE}}$) are used to generate U's decryption key (resp. U's 1st-level helper key) for $\texttt{time}$. More specifically, for $i = 1, 2$, the *key-update information generation algorithm* $\mathsf{Upd}^i_{\mathsf{IKE}}$ takes d^{i-1}_t, p and $\delta^{i-1}_{T_{i-1}(\texttt{time})}$ as inputs for any t, and outputs a new key $d^{i-1}_{T_{i-1}(\texttt{time})}$ for time period $T_{i-1}(\texttt{time})$.
$\mathsf{Enc}_{\mathsf{IKE}}$. The *encryption algorithm* $\mathsf{Enc}_{\mathsf{IKE}}$ inputs m, U, p and $\texttt{time}$ where m is a plaintext, U is the user identity and $\texttt{time}$ indicates the time at which m is encrypted, and outputs ciphertext $\langle c, \texttt{time}\rangle$.
$\mathsf{Dec}_{\mathsf{IKE}}$. The *decryption algorithm* $\mathsf{Dec}_{\mathsf{IKE}}$ inputs $\langle c, \texttt{time}\rangle$, d^0_t and p, and outputs m or $\perp$ where $\perp$ indicates failure. $\mathsf{Dec}_{\mathsf{IKE}}$ correctly recovers the plaintext only if $t = T_0(\texttt{time})$.

Security Definition. Security of IKE is based on the assumption that adversary does not (illegally) obtain all of the victim user's keys all at once. It is important to note that these keys of the different levels of the hierarchy are managed in different manners (and most likely at different places) and tells us how unlikely for such an even to occur, leaving the fact that each level PD is disconnected from the network most of the time. We also remind that it gets much harder for an adversary to obtain the keys as it goes higher in the hierarchy (as higher level PDs are connected to the network less frequently and also managed in places physically safer, i.e. less handy).

We now know that it is indeed a rare case for the above assumption to collapse, we still need to consider for compromisation of keys of each level. We discuss this in the following. There, we consider an attack model based on the standard IND-ID-CCA setting of [7, 8] but also considering the case when the adversary is allowed access to any of victim user's keys and the helper keys, but excluding the combinations of keys that can trivially determine the target key (from the definition of IKE).

Next, we start giving examples of key exposures in our attack model.

Examples of Key Exposures. We consider a 2-level IKE in which the decryption key is renewed every day, 1st-level helper key renewed every three months and the 2nd-level helper key never updated. Then, any ciphertext for 2005/Dec./31st should not be decrypted by dishonest means even for the following cases:

1. Exposure of the victim's 1st-level helper key for 2005/Jan.-Mar., $\cdots$, 2005/Jul.-Sep. and decryption key for 2005/Jan./1st, $\cdots$, 2005/Dec./30th

2. Exposure of the victim's 2nd-level helper key and decryption key for 2005/Jan./1st, $\cdots$, 2005/Dec./30th
3. Exposure of the victim's 2nd-level helper key and 1st-level helper key for 2005/Jan.-Mar., $\cdots$, 2005/Oct.-Dec.

It should be noticed that in case of the exposure of the victim's 1st-level helper key for 2005/Oct.-Dec. and decryption key for 2005/Dec./30th, the decryption key for 2005/Dec./31st is easily computable from the definition of IKE. These types of key exposures are out of our scope.

Next, we formally address our security definition. In our attack model, adversary is allowed access to the following four types of oracles: first, is a *key generation oracle* $\mathsf{KG}(\cdot, s, p)$, which on input U, returns U's initial decryption keys (d_0^0, d_0^1, d_0^2). The second, is a *left-or-right encryption oracle* $\mathsf{LR}(\cdot, \cdot, \cdot, \cdot, p, b)$ [3], which for given U, time and equal length messages m_0, m_1, returns *challenge ciphertext* $c := \mathsf{Enc}_{\mathsf{IKE}}(m_b, U, p, \mathsf{time})$ where $b \in_R \{0,1\}$. This models encryption requests of an adversary for a victim's identity and message pairs of his choice. The third is a *decryption oracle* $\mathsf{D}(\cdot, \cdot, s, p)$ which on input U and $\langle c, \mathtt{time} \rangle$, returns decryption result of c with the corresponding decryption key d_t^0 where $t = T_0(\mathtt{time})$. This models the chosen ciphertext attack. With these three oracles, KG, LR and D, the standard IND-ID-CCA setting can be modeled. In addition to the above, we introduce a *key issue oracle* $\mathsf{KI}(\cdot, \cdot, \cdot, s, p)$ which on input i, U and $\mathtt{time}$, returns d_t^i where $t = T_i(\mathtt{time})$. This models partial exposure of honest user's keys including the ones of the victim's. The adversary may query the four oracles adaptively, in any order he wants, subject to the restriction that he makes only one query to LR. Let U^* be the user's identifier of this query, and let $\langle c^*, \mathsf{time}^* \rangle$ denote the challenge ciphertext returned by LR in response to this query. Also, the adversary is not allowed to ask KG and KI for queries which can trivially determine U^*'s decryption key for time^* from the definition of IKE. The adversary succeeds the attack by guessing the value b, and the scheme is considered to be secure if any probabilistic polynomial time adversary has success probability negligibly close to 1/2.

Definition 2 (KE-CCA security) Let IKE be a 2-level identity-based key-insulated encryption scheme. Define adversary A's succeeding probability as:

$$\mathsf{Succ}_{A,\mathsf{IKE}} := \Pr[(s,p) \leftarrow \mathsf{PGen}_{\mathsf{IKE}}(1^k); b \in_R \{0,1\}; b' \leftarrow A^{\mathsf{KG}(\cdot,s,p),\mathsf{LR}(\cdot,\cdot,\cdot,\cdot,p,b),\mathsf{D}(\cdot,\cdot,s,p),\mathsf{KI}(\cdot,\cdot,\cdot,s,p)} : b' = b],$$

where U^* is never asked to $\mathsf{KG}(\cdot, s, p)$ and A is not allowed to query $\mathsf{D}(U^*, \langle c^*, \mathsf{time} \rangle, s, p)$ if $T_0(\mathtt{time}) = T_0(\mathtt{time}^*)$.

A can ask KI about any keys of any users if there exists a "special level" $j \in \{0,1,2\}$ such that

- $\mathsf{KI}(j, U^*, \mathtt{time}, s, p)$ is never asked for any $\mathtt{time}$, and
- $\mathsf{KI}(i, U^*, \mathtt{time}, s, p)$ is never asked for any $(i, \mathtt{time})$ such that $i < j$ and $T_i(\mathtt{time}) = T_i(\mathtt{time}^*)$.

Then, IKE is *KE-CCA secure* (KE-CCA stands for *key exposure & chosen ciphertext attack*) if, for any probabilistic polynomial time adversary A, $|\mathsf{Succs}_{A,\mathsf{IKE}} - 1/2|$ is negligible. (Note that a "special level" is a level in which the PD of U^* is not compromised. Also, recall 0-level PD is a user's terminal, i.e. the mobile phone.)

Exposure of Key-Updating Information. If we look closer into the security of IKE, we can realize that exposure of key-update information should also be considered in addition to the above discussion. Although, we can also see that it is obvious that if $\delta^i_{T_i(\mathtt{time})}$ can be computed from $d^i_{T_i(\mathtt{time})}$ and d_t^i for any $\mathtt{time}$ and t, then, exposure of key-update information can be simulated by using KI. Hence, the security definition so far discussed will be sufficient alone, even against exposure of the key-update information, if we assume that this property holds. As a matter of fact, all of our constructions satisfy this property.

3 Straightforward IKE from HIBE is Insecure

Although HIBE and IKE are alike in some sense, it is not as simple as bringing HIBE as building blocks to construct KE-CCA secure IKE. We further give discussion on this, but first, we clarify the relation between HIBE and IKE.

Brief Review of HIBE. HIBE distributes the workload of the PKG in IBE by organizing the PKGs in a hierarchical tree structure. Next we give definition of HIBE and its security. This definition runs parallel with [25] which is the hierarchical extension of Boneh and Franklin's [7, 8]. Note that 1-level HIBE refers to a standard IBE.

In HIBE, a user has a position in the hierarchy which is defined as a tuple of identities: $(D^{t-1}.D^{t-2}.\cdots.D^0)$ where t denotes depth of the hierarchy. The user's ancestors in the hierarchy tree are the root-PKG and users/sub-PKGs whose identities are $\{(D^{t-1}.D^{t-2}.\cdots.D^i : 0 \leq i \leq t-1)\}$.

Definition 3 (HIBE) A *t-level hierarchical identity-based encryption (HIBE)* HIBE consists of $3+t$ algorithms: $\mathsf{HIBE} = (\mathsf{PGen}_{\mathsf{HIBE}}, \mathsf{Gen}^i_{\mathsf{HIBE}}\ (1 \leq i \leq t), \mathsf{Enc}_{\mathsf{HIBE}}, \mathsf{Dec}_{\mathsf{HIBE}})$ and are defined as follows:
$\underline{\mathsf{PGen}_{\mathsf{HIBE}}.}$ The *public-parameter generation algorithm* $\mathsf{PGen}_{\mathsf{HIBE}}(1^k)$ where k is the security parameter, outputs root-master key s and public parameter p. $\mathsf{PGen}_{\mathsf{HIBE}}$ is used only by the root-PKG.
$\underline{\mathsf{Gen}^i_{\mathsf{HIBE}}.}$ The *user-secret generation algorithm* $\mathsf{Gen}^t_{\mathsf{HIBE}}$ inputs D^{t-1}, s and p, and outputs D^{t-1}'s key $s_{D^{t-1}}$. Similarly, for $2 \leq i \leq t$, $\mathsf{Gen}^{t-i+1}_{\mathsf{HIBE}}$ takes $D^{t-1}.D^{t-2}.\cdots.D^{t-i}$, $s_{D^{t-1}.D^{t-2}.\cdots.D^{t-i+1}}$ and p as inputs, and outputs $D^{t-1}.D^{t-2}.\cdots.D^{t-i}$'s key $s_{D^{t-1}.D^{t-2}.\cdots.D^{t-i}}$. Here, for $1 \leq i \leq t-1$, $s_{D^{t-1}.D^{t-2}.\cdots.D^{t-i}}$ is the sub-master key which enables $D^{t-1}.D^{t-2}.\cdots.D^{t-i}$ to generate his descendant's keys, and $s_{D^{t-1}.D^{t-2}.\cdots.D^0}$ is the decryption key of $D^{t-1}.D^{t-2}.\cdots.D^0$.
$\underline{\mathsf{Enc}_{\mathsf{HIBE}}.}$ The *encryption algorithm* $\mathsf{Enc}_{\mathsf{HIBE}}$ takes m, $D^{t-1}.D^{t-2}.\cdots.D^0$ and p as inputs where m is a plaintext and $D^{t-1}.D^{t-2}.\cdots.D^0$ is the receiver's identity, and outputs a ciphertext c.
$\underline{\mathsf{Dec}_{\mathsf{HIBE}}.}$ The *decryption algorithm* $\mathsf{Dec}_{\mathsf{HIBE}}$ takes c, $s_{D^{t-1}.D^{t-2}.\cdots.D^0}$ and p as inputs, and outputs m or $\perp$ which means failure. $\mathsf{Dec}_{\mathsf{HIBE}}$ recovers the plaintext only if c has been encrypted correctly by using $D^{t-1}.D^{t-2}.\cdots.D^0$ as the encryption key.

Security of HIBE is defined as follows. An adversary adaptively selects a target user's identity and equal length messages m_0, m_1 and submits to *left-or-right encryption* oracle LR which returns ciphertext of m_b such that $b \in_R \{0,1\}$ for target user. The adversary may also have access to *decryption oracle* D which gives decryption results of any ciphertext except for the challenge ciphertext itself from LR. And last is *key generation oracle* KG which exposes any entity's key except for the target's and its ancestors'. HIBE is *secure* if an adversary can correctly determine b with probability at most $1/2 + neg$ where neg is negligible. Especially, HIBE is called $\mathsf{IND\text{-}HID\text{-}CCA}$ (resp. $\mathsf{IND\text{-}HID\text{-}CPA}$) if unlimited access to D and KG (resp. only KG) is allowed for the adversary [25]. Also, HIBE is called IND-$w\mathsf{HID}$-CCA (resp. IND-$w\mathsf{HID}$-CPA) if unlimited access (resp. no access) to D is allowed, while the number of queries to KG is bounded as follows [26]. For at least one of the levels in the hierarchy, unlimited access is allowed, but for the rest of the levels, the number of queries may not exceed a threshold value w such that $w = O(\mathsf{poly}(k))$. See Appendix A for more details.

An Insecure IKE from HIBE. Here, based on a 3-level HIBE, we consider the following (insecure) 2-level IKE: In the initial phase, PKG generates $(s,p) := \mathsf{PGen}_{\mathsf{HIBE}}(1^k)$, and the user U's helper keys and decryption key at $\mathtt{time}$ are set as $d^2_0 := \mathsf{Gen}^3_{\mathsf{HIBE}}(U,s,p)$ and $d^i_{T_i(\mathtt{time})} := \mathsf{Gen}^{i+1}_{\mathsf{HIBE}}(T_i(\mathtt{time}), d^{i+1}_{T_{i+1}(\mathtt{time})}, p)$ for $i = 1, 0$. When encrypting a message m for U at time, a ciphertext c is generated as follows: $c = \mathsf{Enc}_{\mathsf{HIBE}}(m, U.T_1(\mathtt{time}).T_0(\mathtt{time}), p)$. Such a method of the renewal of decryption keys in IBE from HIBE is described in [26] as well.

Above, IKE constructed straightforwardly from HIBE, is insecure (i.e. not KE-CCA secure), however, at first glance, for some, it seems secure. For instance, this construction does not provide security for 2.

```
$\mathsf{PGen}_{\mathsf{IKE}}(1^k)$:
    $(s_h, p_h) \leftarrow \mathsf{PGen}_{\mathsf{HIBE}_h}(1^k)$, $1 \le h \le 3$
    choose $H_h : \{0,1\}^{2n+3k_1} \rightarrow \mathcal{COIN}$, $1 \le h \le 3$
return $s := (s_1, s_2, s_3)$
    $p := (p_1, p_2, p_3, H_1, H_2, H_3)$
```

```
$\mathsf{Gen}_{\mathsf{IKE}}(s, p, U)$:
    parse $s = (s_1, s_2, s_3)$
    parse $p = (p_1, p_2, p_3, H_1, H_2, H_3)$
    $s_{h,U} \leftarrow \mathsf{Gen}^h_{\mathsf{HIBE}_h}(U, s_h, p_h)$, $1 \le h \le 3$
    $d^0_0 := (s_{1,U}, \cdot, \cdot)$, $d^1_0 := (s_{2,U}, \cdot)$, $d^2_0 := s_{3,U}$
return $(d^0_0, d^1_0, d^2_0)$
```

```
$\Delta\text{-}\mathsf{Gen}^1_{\mathsf{IKE}}(d^1_t, p, U, \mathtt{time})$:
    parse $d^1_t = (\sigma_2, \sigma_3)$
    $\sigma'_h \leftarrow \mathsf{Gen}^1_{\mathsf{HIBE}_h}(T_0(\mathtt{time}), \sigma_h, p_h)$, $h = 2, 3$
return $\delta^0_{T_0(\mathtt{time})} := (\sigma'_2, \sigma'_3)$
```

```
$\Delta\text{-}\mathsf{Gen}^2_{\mathsf{IKE}}(d^2_0, p, U, \mathtt{time})$:
    parse $d^2_0 = \sigma_3 (= s_{3,U})$
    $\sigma'_3 \leftarrow \mathsf{Gen}^2_{\mathsf{HIBE}_3}(T_1(\mathtt{time}), \sigma_3, p_3)$
return $\delta^1_{T_1(\mathtt{time})} := \sigma'_3$
```

```
$\mathsf{Upd}^1_{\mathsf{IKE}}(d^0_t, p, \delta^0_{T_0(\mathtt{time})})$:
    parse $d^0_t = (\sigma_1, \sigma_2, \sigma_3)$ and $\delta^0_{T_0(\mathtt{time})} = (\sigma'_2, \sigma'_3)$
return $d^0_{T_0(\mathtt{time})} := (\sigma_1, \sigma'_2, \sigma'_3)$
```

```
$\mathsf{Upd}^2_{\mathsf{IKE}}(d^1_t, p, \delta^1_{T_1(\mathtt{time})})$:
    parse $d^1_t = (\sigma_2, \sigma_3)$ and $\delta^1_{T_1(\mathtt{time})} = \sigma'_3$
return $d^1_{T_1(\mathtt{time})} := (\sigma_2, \sigma'_3)$
```

```
$\mathsf{Enc}_{\mathsf{IKE}}(m, U, p, \mathtt{time})$:
    $\overline{m}_1, \overline{m}_2 \leftarrow \{0,1\}^n$, $\overline{m}_3 := m \oplus \overline{m}_1 \oplus \overline{m}_2$
    $r_1, r_2, r_3 \leftarrow \{0,1\}^{k_1}$
    $R_h := H_h(m, \overline{m}_h, r_1, r_2, r_3)$, $1 \le h \le 3$
    $U_1 := U$, $U_2 := U.T_0(\mathtt{time})$,
    $U_3 := U.T_1(\mathtt{time}).T_0(\mathtt{time})$
    $c_h := \mathsf{Enc}_{\mathsf{HIBE}_h}(\overline{m}_h || r_h, U_h, p_h; R_h)$, $1 \le h \le 3$
return $\langle c, \mathtt{time} \rangle := \langle (c_1, c_2, c_3), \mathtt{time} \rangle$
```

```
$\mathsf{Dec}_{\mathsf{IKE}}(\langle c', \mathtt{time} \rangle, d^0_t, p)$:
    output $\perp$ and halt if $t \ne T_0(\mathtt{time})$
    parse $c' = (c'_1, c'_2, c'_3)$ and $d^0_t = (\sigma_1, \sigma_2, \sigma_3)$
    $(\overline{m}'_h || r'_h) \leftarrow \mathsf{Dec}_{\mathsf{HIBE}_h}(c'_h, \sigma_h, p_h)$, $1 \le h \le 3$
    $m' := \oplus_{1 \le h \le 3} \overline{m}'_h$
    validity check by re-encryption
    output $\perp$ and halt if invalid
return $m'$
```

Figure 2: Generic Construction of KE-CCA Secure IKE from IND-HID-CPA HIBE.

and 3. of the Examples of Key Exposures. in the previous section. Namely, if the 1st-level PD (or the PD-BC) is stolen at `2005/Oct./1st/0:00`, then confidentiality of the ciphertexts generated during period `2005/Oct.-Dec.` will all be lost. Moreover, exposure of one of 2nd-level helper key can alone compromise the security for any time period. Therefore, it is not KE-CCA secure.

4 Generic Construction

Basic Idea. As shown in the previous section, constructing IKE straightforwardly from HIBE was vulnerable, and loss of user's PD alone even implied loss of security of the entire system. We next show a generic construction of IKE from three distinct HIBEs. Here's the general idea: each HIBE plays a part to mutually secure the different types of key exposures, consequently, protecting the system totally, guaranteeing security even if compromization of PD occurs. We extend a technique called *multiple encryption* proposed in [30] to construct secure IKE from HIBE that achieves KE-CCA security. It is important to note that the original [30] scheme is applied only to standard public key encryption, so, straightforward adoption of this scheme, again, cannot realize construction of secure IKE.

Construction. Fig. 2 shows the generic construction of KE-CCA secure IKE from any HIBE where each of HIBEs with only a *chosen plaintext security*, i.e. IND-HID-CPA (See Appendix A). Here, we give supplementary explanation of the Fig. 2 and give discussion on our generic construction in more details.

Let $\mathsf{HIBE}_h = (\mathsf{PGen}_{\mathsf{HIBE}_h}, \mathsf{Gen}^i_{\mathsf{HIBE}_h} (1 \le i \le h), \mathsf{Enc}_{\mathsf{HIBE}_h}, \mathsf{Dec}_{\mathsf{HIBE}_h})$ be h-level HIBE for $1 \le h \le 3$ and construct a 2-level IKE $\mathsf{IKE} = (\mathsf{PGen}_{\mathsf{IKE}}, \mathsf{Gen}_{\mathsf{IKE}}, \Delta\text{-}\mathsf{Gen}^i_{\mathsf{IKE}}, \mathsf{Upd}^i_{\mathsf{IKE}} (i = 1, 2), \mathsf{Enc}_{\mathsf{IKE}}, \mathsf{Dec}_{\mathsf{IKE}})$ as follows.

$\mathsf{PGen}_{\mathsf{IKE}}$ sets up the master keys and public parameters of HIBE_h as well as cryptographic hash functions H_h for $1 \le h \le 3$ where n denotes the size of a message of IKE. $\mathcal{COIN}$ is the internal coin-flipping space of $\mathsf{Enc}_{\mathsf{HIBE}_h}$, assuming that $n + k_1$ is the size of a message in HIBE_h.[1] The security analysis will view H_h as random oracles. $\mathsf{Gen}_{\mathsf{IKE}}$ generates U's secrets of HIBE_h for $1 \le h \le 3$ as U's initial key for IKE. $\Delta\text{-}\mathsf{Gen}^1_{\mathsf{IKE}}$ generates decryption keys of HIBE_2 and HIBE_3 for identities $U.T_0(\mathtt{time})$ and $U.T_1(\mathtt{time}).T_0(\mathtt{time})$, respectively, as the "differential" of the U's previous key and of the next renewed key at $\mathtt{time}$. Then, $\mathsf{Upd}^1_{\mathsf{IKE}}$ generates U's decryption key of IKE for $\mathtt{time}$ by combining the differential with the U's previous key. Similarly, $\Delta\text{-}\mathsf{Gen}^2_{\mathsf{IKE}}$ generates a sub-master key of HIBE_3 for $U.T_1(\mathtt{time})$, and $\mathsf{Upd}^2_{\mathsf{IKE}}$ generates U's 1st-level helper key of IKE for $\mathtt{time}$ by combining U's previous key and $\Delta\text{-}\mathsf{Gen}^2_{\mathsf{IKE}}$'s output. $\mathsf{Enc}_{\mathsf{IKE}}$ *securely* integrates three encryption algorithms of h-level HIBE for $1 \le h \le 3$. Plaintext m is divided into three shares $\overline{m}_1, \overline{m}_2, \overline{m}_3$, and each $\overline{m}_h$ $(1 \le h \le 3)$ is encrypted by h-level HIBE HIBE_h for identity U_h where $U_1 := U$, $U_2 := U.T_0(\mathtt{time})$ and $U_3 := U.T_1(\mathtt{time}).T_0(\mathtt{time})$. Here, the technique in [30] is applied (but not straightforwardly, as mentioned earlier) to securely integrates the three composites HIBEs. $\mathsf{Dec}_{\mathsf{IKE}}$ recovers each three shares and composes them into a plaintext. It also checks the validity of the ciphertext by re-encryption. Namely, $R'_h := H_h(m', \overline{m}'_h, r'_1, r'_2, r'_3)$ and $\nu_h \leftarrow \mathsf{Enc}_{\mathsf{HIBE}_h}(\overline{m}'_h || r'_h, U_h, p_h; R'_h)$ are computed for $1 \le h \le 3$. Unless $\nu_h = c'_h$ for all h, output $\perp$, otherwise output m'.

The above scheme can easily be generalized to ℓ-level IKE for arbitrary $\ell \ge 1$.

Definition 4 (γ-uniformity [23]) Let $\mathsf{HIBE} = (\mathsf{PGen}_{\mathsf{HIBE}}, \mathsf{Gen}^i_{\mathsf{HIBE}}\ (1 \le i \le t), \mathsf{Enc}_{\mathsf{HIBE}}, \mathsf{Dec}_{\mathsf{HIBE}})$ be t-level HIBE. For given $D^{t-1}.D^{t-2}.\cdots.D^0$, x, y and z, define $\gamma(D^{t-1}.D^{t-2}.\cdots.D^0, x, y, z) = \Pr[r \leftarrow_R \mathcal{COIN} : z = \mathsf{Enc}_{\mathsf{HIBE}}(D^{t-1}.D^{t-2}.\cdots.D^0, x, y; r)]$ where $\mathcal{COIN}$ is the internal coin-flipping space for $\mathsf{Enc}_{\mathsf{HIBE}}$. We say that HIBE is *γ-uniform* if $\gamma(D^{t-1}.D^{t-2}.\cdots.D^0, x, y, z) \le \gamma$ for any $D^{t-1}.D^{t-2}.\cdots.D^0$, x, y and z.

Theorem 1 *The above scheme is a KE-CCA secure 2-level IKE in the random oracle model, assuming that* HIBE_h $(1 \le h \le 3)$ *are* $\mathsf{IND\text{-}HID\text{-}CPA}$ *HIBEs. More precisely, suppose there is an adversary A who can break the above scheme with probability $1/2 + \epsilon_A$ with run time at most t_A. Suppose A makes at most q_{KG}, q_{KI}, q_{D}, q_{H_1}, q_{H_2}, q_{H_3} queries to* KG, KI, D, *H_1, H_2, H_3, respectively. Then, there is another adversary B who can break at least one of* HIBE_h $(1 \le h \le 3)$ *in the sense of* $\mathsf{IND\text{-}HID\text{-}CPA}$ *with probability $1/2 + \epsilon_B$ and running time t_B where*

$$\begin{aligned} \epsilon_B &\ge \frac{1}{3}\epsilon_A - \frac{1}{3}\frac{q_{H_1} + q_{H_2} + q_{H_3}}{2^{k_1}} - \frac{1}{6}q_{\mathsf{D}}\gamma_{max}, \\ t_B &\le t_A + (2q_{\mathsf{KG}} + 5q_{\mathsf{KI}})\tau_{GEN} + q_{H_1}q_{H_2}q_{H_3}(3\tau_{ENC} + O(k)), \end{aligned}$$

assuming that $\gamma_{max} = \max(\gamma_1, \gamma_2, \gamma_3)$, HIBE_i *is γ_i-uniform, and running time of* $\mathsf{Gen}^i_{\mathsf{HIBE}_h}$ *and* $\mathsf{Enc}_{\mathsf{HIBE_h}}$ *are at most τ_{GEN} and τ_{ENC}, respectively, for any h and i.*

Proof. See Appendix B. □

Random Oracle. If we want to eliminate random oracle, multiple encryption technique in [13] can be extended instead of the one we used of [30] to construct a KE-CCA secure IKE, assuming that underlying HIBEs are all $\mathsf{IND\text{-}HID\text{-}CCA}$ in the standard model, e.g. [11, 4, 5, 6, 29], while the above construction using [30] requires only $\mathsf{IND\text{-}HID\text{-}CPA}$ HIBEs. Furthermore, by applying a similar method to our proposed scheme, we can construct another KE-CCA secure IKE from HIBE with only one-wayness under chosen plaintext attacks.

[1] For simplicity, we assume for all HIBE_h, spaces of coin-flipping and messages to be $\mathcal{COIN}$ and $\{0,1\}^{n+k_1}$, respectively.

Strongly Secure Hierarchical "Standard" Key-Insulated Encryption. By extending the technique used in the above, we can construct a generic construction of a strongly secure key-insulated encryption [14] from a chosen plaintext secure IBE and a chosen plaintext secure standard public key encryption. This method can also be applied to the Cocks IBE [10] to construct a strongly secure key-insulated encryption. (The Boneh-Franklin IBE based scheme was proposed earlier in [9]).

In the following, we give general idea of the generic construction of strongly secure key-insulated encryption: Let $\mathsf{PKE} := (\mathsf{Gen}_{\mathsf{PKE}}, \mathsf{Enc}_{\mathsf{PKE}}, \mathsf{Dec}_{\mathsf{PKE}})$ be a semantically secure public key encryption scheme, where $\mathsf{Gen}_{\mathsf{PKE}}, \mathsf{Enc}_{\mathsf{PKE}}, \mathsf{Dec}_{\mathsf{PKE}}$ are algorithms for key generation, encryption and decryption, respectively, and $\mathsf{IBE} := (\mathsf{PGen}_{\mathsf{IBE}}, \mathsf{Gen}_{\mathsf{IBE}}, \mathsf{Enc}_{\mathsf{IBE}}, \mathsf{Dec}_{\mathsf{IBE}})$ be an IND-ID-CPA identity-based encryption scheme [8] (i.e. IND-HID-CPA for $t = 1$), where $\mathsf{PGen}_{\mathsf{IBE}}, \mathsf{Gen}_{\mathsf{IBE}}, \mathsf{Enc}_{\mathsf{IBE}}, \mathsf{Dec}_{\mathsf{IBE}}$ are algorithms for public-parameter generation, user-secret generation, encryption and decryption, respectively (note that IBE is equivalent to 1-level HIBE). Next, the user computes $\mathsf{Gen}_{\mathsf{PKE}}(1^k) = (dk, ek)$ and $\mathsf{PGen}_{\mathsf{IBE}}(1^k) = (s, p)$ for a security parameter k, and publicizes (ek, p). User keeps dk and stores s inside his PD. In order to renew his decryption key at time period t, PD computes $\mathsf{Gen}_{\mathsf{IBE}}(t, s, p) = s_t$ and sends the output value to the user. This value is used to update the decryption key, (dk, s_t) at time t. When encrypting a message m for time period t, $\overline{m}_1$, $\overline{m}_2$, r_1 and r_2, such that $\overline{m}_1 + \overline{m}_2 = m$, are picked uniformly at random, and $\mathsf{Enc}_{\mathsf{PKE}}(\overline{m}_1||r_1, ek; H_1(m, \overline{m}_1, r_1, r_2)) = c_1$ and $\mathsf{Enc}_{\mathsf{IBE}}(\overline{m}_2||r_2, t, p; H_2(m, \overline{m}_2, r_1, r_2)) = c_2$ are computed where H_1 and H_2 are random oracles. Finally, a ciphertext, (c_1, c_2) is generated. It is obvious that m can be recovered from (c_1, c_2) with decryption key (dk, s_t). In addition, chosen ciphertext attacks is prevented for the following cases as well: (1) exposure of unlimited number of decryption keys for any time periods except for t, (2) exposure of s. This is the first generic construction ever been built of a strongly secure key-insulated encryption from IBE and standard public key encryption in the random oracle model. Security proof is simialar to Theorem 1. Moreover, by using a similar method used in the previous subsection, we can extend the above scheme to be hierarchical as well. This is also the first hierarchical construction of a strongly secure key-insulated encryption.

5 Efficient Construction from Bilinear Mapping

Basic Idea. In the previous section, we showed a construction of KE-CCA secure IKE using HIBE as a black-box. Here, we propose a construction of KE-CCA secure IKE by directly extending Gentry-Silverberg HIBE (GS-HIBE) [25] (see also Appendix C) and Fujisaki-Okamoto conversion [22, 23]. The basic difference between the two construction is: for the specific construction, h-level HIBE for $1 \leq h \leq 3$ are integrated using the homomorphic property of pairing, while the generic construction is based on multiple encryption [30]. The specific construction we describe in this section is more efficient than the generic one. Note that since it is based on a very specific assumption, i.e. BDH assumption, it may lack flexibility in designing new construction in terms of security.

Construction. From bilinear mapping, a 2-level IKE $\mathsf{IKE} = (\mathsf{PGen}_{\mathsf{IKE}}, \mathsf{Gen}_{\mathsf{IKE}}, \Delta\text{-}\mathsf{Gen}^i_{\mathsf{IKE}}, \mathsf{Upd}^i_{\mathsf{IKE}}$ $(i = 1, 2), \mathsf{Enc}_{\mathsf{IKE}}, \mathsf{Dec}_{\mathsf{IKE}})$ can be constructed as shown in Fig. 3. Here, we give supplementary explanation of the Fig. 3 and give discussion on our specific construction in more details.

From bilinear mapping, a 2-level IKE $\mathsf{IKE} = (\mathsf{PGen}_{\mathsf{IKE}}, \mathsf{Gen}_{\mathsf{IKE}}, \Delta\text{-}\mathsf{Gen}^i_{\mathsf{IKE}}, \mathsf{Upd}^i_{\mathsf{IKE}}$ $(i = 1, 2), \mathsf{Enc}_{\mathsf{IKE}}, \mathsf{Dec}_{\mathsf{IKE}})$ can be constructed as follows.

$\mathsf{PGen}_{\mathsf{IKE}}$ generates two cyclic groups G_1 and G_2 of prime order q and an efficiently computable mapping $\hat{e} : G_1 \times G_1 \rightarrow G_2$ such that $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ for all $P, Q \in G_1$ and any positive integers a, b. This does not send all pairs in $G_1 \times G_1$ to the identity in G_2. Also, $\mathsf{PGen}_{\mathsf{IKE}}$ chooses cryptographic hash functions $H_1 : \{0,1\}^* \rightarrow G_1$, $H_2 : G_2 \rightarrow \{0,1\}^{n+k_1}$ and $H_3 : \{0,1\}^n \times \{0,1\}^{k_1} \rightarrow Z_q$, where n denotes the size of the message space. The security analysis will view H_1, H_2, H_3 as random oracles. It further

```
$\mathsf{PGen}_{\mathsf{IKE}}(1^k)$:
    set up $G_1$, $G_2$, $\hat{e}$, $P \in G_1$
    $s_1^0, s_2^1, s_3^2 \in_R Z_q$, $Q := \sum_{h=1}^{3} s_h^{h-1} P$
    choose $H_1$, $H_2$, $H_3$
return $s := (s_1^0, s_2^1, s_3^2)$
    $p := (G_1, G_2, \hat{e}, P, Q, H_1, H_2, H_3)$
```

```
$\mathsf{Gen}_{\mathsf{IKE}}(s, p, U)$:
    $P_U := H_1(U) \in G_1$
    $S_h^{h-1} := s_h^{h-1} P_U$, $1 \leq h \leq 3$
    $d_0^0 := (S_1^0, (\cdot,\cdot), (\cdot,\cdot,\cdot))$
    $d_0^1 := (S_2^1, (\cdot,\cdot))$, $d_0^2 := S_3^2$
return $(d_0^0, d_0^1, d_0^2)$
```

```
$\Delta\text{-}\mathsf{Gen}_{\mathsf{IKE}}^1(d_t^1, p, U, \mathtt{time})$:
    parse $d_t^1 = (S_2^1, (S_3^1, Q_3^1))$, $s_2^0, s_3^0, \in_R Z_q$
    $P_{t_0} := H_1(U.T_1(\mathtt{time}).T_0(\mathtt{time}))$
    $\hat{S}_h^0 := S_h^1 + s_h^0 P_{t_0}$, $\hat{Q}_h^0 := s_h^0 P$, $h = 2, 3$
return $\delta_{T_0(\mathtt{time})}^0 := ((\hat{S}_2^0, \hat{Q}_2^0), (\hat{S}_3^0, \hat{Q}_3^0, Q_3^1))$
```

```
$\Delta\text{-}\mathsf{Gen}_{\mathsf{IKE}}^2(d_0^2, p, U, \mathtt{time})$:
    parse $d_0^2 = S_3^2$, $s_3^1 \in_R Z_q$
    $P_{t_1} := H_1(U.T_1(\mathtt{time}))$
    $\hat{S}_3^1 := S_3^2 + s_3^1 P_{t_1}$, $\hat{Q}_h^1 := s_3^1 P$
return $\delta_{T_1(\mathtt{time})}^1 := (\hat{S}_3^1, \hat{Q}_3^1)$
```

```
$\mathsf{Upd}_{\mathsf{IKE}}^1(d_t^0, p, \delta_{T_0(\mathtt{time})}^0)$:
    parse $d_t^0 = (S_1^0, (S_2^0, Q_2^0), (S_3^0, Q_3^0, Q_3^1))$
    parse $\delta_{T_0(\mathtt{time})}^0 = ((\hat{S}_2^0, \hat{Q}_2^0), (\hat{S}_3^0, \hat{Q}_3^0, \hat{Q}_3^1))$
return $d_t^0 := (S_1^0, (\hat{S}_2^0, \hat{Q}_2^0), (\hat{S}_3^0, \hat{Q}_3^0, \hat{Q}_3^1))$
```

```
$\mathsf{Upd}_{\mathsf{IKE}}^2(d_t^1, p, \delta_{T_1(\mathtt{time})}^1)$:
    parse $d_t^1 = (S_2^1, (S_3^1, Q_3^1))$
    parse $\delta_{T_1(\mathtt{time})}^1 = (\hat{S}_3^1, \hat{Q}_3^1)$
return $d_{T_1(\mathtt{time})}^1 := (S_2^1, (\hat{S}_3^1, \hat{Q}_3^1))$
```

```
$\mathsf{Enc}_{\mathsf{IKE}}(m, U, p, \mathtt{time})$:
    $\mu \in_R \{0,1\}^n$, $r := H_3(\mu, m)$, $g := \hat{e}(Q, P_U) \in G_2$
    $P_U := H_1(U)$, $P_{t_1} := H_1(U.T_1(\mathtt{time}))$
    $P_{t_0} := H_1(U.T_1(\mathtt{time}).T_0(\mathtt{time}))$
    $c := \langle rP, rP_{t_1}, rP_{t_0}, (m||\mu) \oplus H_2(g^r) \rangle$
return $\langle c, \mathtt{time} \rangle$
```

```
$\mathsf{Dec}_{\mathsf{IKE}}(\langle c', \mathtt{time} \rangle, d_t^0, p)$:
    parse $c' = \langle V, V_{t_1}, V_{t_0}, W \rangle$
    parse $d_t^0 = (S_1^0, (S_2^0, Q_2^0), (S_3^0, Q_3^0, Q_3^1))$
    $(m'||\mu') := W \oplus H_2(\frac{\hat{e}(S_1^0 + S_2^0 + S_3^0, V)}{\hat{e}(Q_2^0 + Q_3^0, V_{t_0}) \hat{e}(Q_3^1, V_{t_1})})$
    validity check by re-encryption
return $m'$
```

Figure 3: KE-CCA Secure IKE from Bilinear Mapping.

generates master key s and its corresponding public paramter Q. $\mathsf{Gen}_{\mathsf{IKE}}$, $\Delta\text{-}\mathsf{Gen}_{\mathsf{IKE}}^i$ and $\mathsf{Upd}_{\mathsf{IKE}}^i$ $(i = 1, 2)$ are the same as the generic construction based on [25] as underlying HIBE. Based on the homomorphic property of pairing, $\mathsf{Enc}_{\mathsf{IKE}}$ and $\mathsf{Dec}_{\mathsf{IKE}}$ integrates three HIBE encryptions into one. Although hasn't been mentioned in Fig. 3, in order to protect active attacks, $\mathsf{Dec}_{\mathsf{IKE}}$ outputs $\perp$ and halts if (i) $t \neq T_0(\mathtt{time})$, (ii) $(V, V_{t_1}, V_{t_0}, W) \notin {G_1}^3 \times \{0,1\}^{n+k_1}$, or, (iii) re-encryption of m' for U, time and μ' is not identical to $\langle c', \mathsf{time} \rangle$.

Theorem 2 *The above scheme is a KE-CCA secure 2-level IKE in the random oracle model, assuming computational BDH (CBDH) problem [7, 8] is hard to solve. More precisely, suppose there is an adversary A who breaks the above scheme with probability $1/2 + \epsilon_A$ with run time at most t_A. Also, we suppose A makes at most q_{KG}, q_{KI}, q_{D}, $q_{H_1}, q_{H_2}, q_{H_3}$ queries to KG, KI, D, H_1, H_2, H_3, respectively. Then, there is another adversary who can solve the CBDH problem with probability ϵ_{cbdh} and running time t_{cbdh} where*

$$\begin{aligned} \epsilon_{cbdh} &\geq \frac{6}{e^3 q_{H_2}(3 + q_{\mathsf{KG}} + q_{\mathsf{KI}})^3} \cdot (\epsilon_A - \frac{q_{H_3}}{2^{k_1}} - \frac{q_{\mathsf{D}}}{2q}), \\ t_{cbdh} &\leq O(t_A + (3q_{\mathsf{KG}} + 5q_{\mathsf{KI}})\tau_{EXP} + q_{H_1}^3 q_{H_2} q_{H_3}(\tau_{\hat{e}} + O(k))), \end{aligned}$$

assuming time for exponentiation over G_1 is at most τ_{EXP}, and time for pairing computation is at most $\tau_{\hat{e}}$.

Proof. See Appendix D. □

Efficiency. In a pairing based scheme, the dominant factor that decides its total computation cost is the number of pairing computation carried out. For the above construction of KE-CCA secure IKE from bilinear mapping, only one and three pairing computations are required for encryption and decryption, respectively. On the other hand, for the generic construction (shown in the previous section) that uses [25] as the underlying HIBE, the numbers of pairing computation for encryption and decryption are three and six, respectively. Hence, in terms of computational cost, specific construction surpasses the generic construction based on [25] in efficiency. This result can be generalized for ℓ-level IKE for any $\ell > 1$ as shown in Table 1.

6 Generic HIBE from Any IBE

From our discussion so far, we can see that HIBE serves an important role as a building block of various cryptographic schemes, including the ones that we have proposed. In this section, we show a generic construction of HIBE from arbitrary IBE that also provides a (partial) solution to an open problem of HIBE. We can, for example, bring the Cocks IBE [10] to construct a HIBE, also implying that, hereafter, a new construction of IBE is proposed, it is automatically convertible to construct a HIBE. For the security definition, we introduce partial collusion resistance (i.e. $\mathsf{IND}\text{-}w\mathsf{HID}\text{-}\mathsf{CCA}$) [26] instead of full collusion resistance (i.e. $\mathsf{IND}\text{-}\mathsf{HID}\text{-}\mathsf{CCA}$) [25]. The security definition is relaxed but our contribution is significant as this is the first generic construction of HIBE constructed from arbitrary IBE. In this section, for simplicity, we show a construction of a 2-level HIBE, but it can also be extended to construct t-level HIBE for $t > 2$ very easily.

Security Definition. Our construction of HIBE proposed here is based on the security definition of [26]. Particularly, for our 2-level construction of HIBE, it is collusion free for the users (in the lower domain), but has polynomial-sized collusion threshold w for the sub-PKGs (in the higher domain), where $w = O(\mathsf{poly}(k))$ and k is the security parameter.

Cover Free Family. The scheme shown here utilizes cover free family (CFF) [18] similarly seen in the generic construction of key-insulated encryption [14], although, reminding that, the method used in [14] only addresses chosen plaintext security, and it cannot be applied straightforwardly to construct a chosen ciphertext secure HIBE.

Definition 5 (CFF) Let $L := \{\ell_1, \ell_2, \cdots, \ell_u\}$ and $F = \{F_1, \cdots, F_v\}$ be a family of subsets of L. We call (L, F) an (u, v, w)-*cover free family* (CFF) if for all $F_i \in F$, $F_i \not\subset F_{j_1} \cup \cdots \cup F_{j_w}$ for any $F_{j_k}(\neq F_i) \in F$, $k \in \{1, ..., w\}$.

It should be noted that there exist nontrivial constructions of CFF with $u = O(w^2 \log v)$ and $\#F_i = O(w \log v)$ $(1 \leq i \leq v)$. In the following, we assume $\#F_1 = \#F_2 = \cdots = \#F_v = \hat{u}$ for some $\hat{u}$ and $\#\{F_i | \ell_j \in F_i \in F\} \geq [v\hat{u}/u]$ for all $\ell_j \in L$. Concrete methods for generating CFF are given in [19].

Construction. Fig. 4 shows a generic construction of a chosen ciphertext secure 2-level HIBE with partial collusion resistance that can be built from an arbitrary $\mathsf{IND}\text{-}\mathsf{ID}\text{-}\mathsf{CPA}$ IBE using CFF. Here, we

Table 1: Numbers of pairing computations in the pairing based scheme and the generic scheme based on [25].

	encryption	decryption
pairing based scheme	1	$\ell + 1$
generic scheme	$\ell + 1$	$\frac{(\ell+1)(\ell+2)}{2}$

```
$\mathsf{PGen}_{\mathsf{HIBE}}(1^k)$:
    generate $(u,v,w)$-CFF $(L,F)$,    $(s_i,p_i) \leftarrow \mathsf{PGen}_{\mathsf{IBE}}(1^k)$, $1 \le i \le u$
    choose $H : \{0,1\}^* \rightarrow F$ and $H_i : \{0,1\}^{2n+\hat{u}k_1} \rightarrow \mathcal{COIN}$, $1 \le i \le u$
return $s := \{s_i\}_{1 \le i \le u}$ and $p := (H, \{p_i, H_i\}_{1 \le i \le u})$
```

```
$\mathsf{Gen}^1_{\mathsf{HIBE}}(D^1, s, p)$:
    parse $s = \{s_i\}_{1 \le i \le u}$ and $p = (H, \{p_i, H_i\}_{1 \le i \le u})$
    $F_{D^1} := H(D^1) \in F$
return $s_{D^1} := \{s_i\}_{i \in F_{D^1}}$
```

```
$\mathsf{Gen}^0_{\mathsf{HIBE}}(D^1.D^0, s_{D^1}, p)$:
    parse $s_{D^1} = \{s_i\}_{i \in F_{D^1}}$ and $p = (H, \{p_i, H_i\}_{1 \le i \le u})$
    $s_{i,D^1.D^0} \leftarrow \mathsf{Gen}_{\mathsf{IBE}}(D^1.D^0, s_i, p_i)$, $i \in F_{D^1}$
return $s_{D^1.D^0} := \{s_{i,D^1.D^0}\}_{i \in F_{D^1}}$
```

```
$\mathsf{Enc}_{\mathsf{HIBE}}(m, D^0.D^1, p)$:
    parse $p = (H, \{p_i, H_i\}_{1 \le i \le u})$
    $F_{D^1} := H(D^1) \in F$
    $\overline{m}_i \in_R \{0,1\}^n$, $i \in F_{D^1}$ such that $\oplus_{i \in F_{D^1}} \overline{m}_i = m$
    $r_i \in_R \{0,1\}^{k_1}$, $i \in F_{D^1}$
    $c_i \leftarrow \mathsf{Enc}_{\mathsf{IBE}}(\overline{m}_i || r_i, D^0.D^1, p_i; H_i(m, \overline{m}_i, R))$, $i \in F_{D^1}$
return $c := \{c_i\}_{i \in F_{D^1}}$
```

```
$\mathsf{Dec}_{\mathsf{HIBE}}(c', s_{D^1.D^0}, p)$:
    parse $s_{D^1} = \{s_i\}_{i \in F_{D^1}}$
    parse $p = (H, \{p_i, H_i\}_{1 \le i \le u})$
    $(\overline{m}'_i || r'_i) \leftarrow \mathsf{Dec}_{\mathsf{IBE}}(c'_i, s_{i,D^1.D^0}, p_i)$, $i \in F_{D^1}$
    $m' := \oplus_{i \in F_{D^1}} \overline{m}'_i$
    validity check by re-encryption
return $m'$
```

Figure 4: Generic Construction of Partially Collusion Resistant HIBE.

give supplementary explanation of the Fig. 4 and give discussion on our generic construction of HIBE in more details.

Let $\mathsf{IBE} = (\mathsf{PGen}_{\mathsf{IBE}}, \mathsf{Gen}_{\mathsf{IBE}}, \mathsf{Enc}_{\mathsf{IBE}}, \mathsf{Dec}_{\mathsf{IBE}})$ be standard IBE (i.e. 1-level HIBE). Then, 2-level HIBE $\mathsf{HIBE} = (\mathsf{PGen}_{\mathsf{HIBE}}, \mathsf{Gen}^i_{\mathsf{HIBE}}\ (i = 1,2), \mathsf{Enc}_{\mathsf{HIBE}}, \mathsf{Dec}_{\mathsf{HIBE}})$ can be constructed as follows.

$\mathsf{PGen}_{\mathsf{HIBE}}$ generates (u,v,w)-CFF (L,F) and u pairs of master key and public parameter of IBE where $L = \{1, \cdots, u\}$, $u = O(\mathsf{poly}(k))$, $v = O(\mathsf{exp}(k))$ and $w = O(\mathsf{poly}(k))$. For hash functions, n denotes the size of a message of HIBE, and $\mathcal{COIN}$ represents the internal coin-flipping space of $\mathsf{Enc}_{\mathsf{IBE}}$, assuming that $n + k_1$ is the size of a message in IBE. The security analysis will view H and H_i $(1 \le i \le u)$ as random oracles. $\mathsf{Gen}^1_{\mathsf{HIBE}}$ picks master keys corresponding to F_{D^1}. $\mathsf{Gen}^0_{\mathsf{HIBE}}$ generates IBE decryption keys by using $s_{D^1} = \{s_i\}_{i \in F_{D^1}}$. $\mathsf{Enc}_{\mathsf{HIBE}}$ encrypts m with encryption algorithms which correspond to F_{D^1} where R is concatenation of all r_i arranged in increasing order of i for $i \in F_{D^1}$. $\mathsf{Dec}_{\mathsf{HIBE}}$ decrypts all c'_i for $i \in F_{D^1}$. Then, it re-encrypts m' with $\overline{m}'_i$ and r'_i. Unless the encryption result is identical to c', $\mathsf{Dec}_{\mathsf{HIBE}}$ outputs $\perp$, otherwise, outputs m'.

Theorem 3 *The above scheme is* IND-$w\mathsf{HID}$-CCA *in the random oracle model, with a restriction that an adversary is allowed to query sub-PKGs' keys at most w times, assuming that* IBE *is* IND-ID-CPA. *More precisely, suppose there is an adversary A who breaks the above scheme with probability $1/2 + \epsilon_A$ with run time at most t_A. Also, suppose A makes at most q_{KG}, q_{D}, q_H, q_{H_i} queries to* KG, D, H, H_i $(1 \le i \le u)$, *respectively. Then, by letting $q_{all} := \sum_{1 \le i \le u} q_{H_i}$ and $q_{max} := \max_{\{i_1,\cdots,i_{\hat{u}}\} \subseteq \{1,\cdots,u\}}(\prod_{i \in \{i_1,\cdots,i_{\hat{u}}\}} q_{H_i})$, there is another adversary B who can break* IBE *in the sense of* IND-ID-CPA *with probability $1/2 + \epsilon_B$ and running time t_B where*

$$\begin{aligned} \epsilon_B &\ge \frac{\hat{u}}{u^2}(\epsilon_A - \frac{q_{all}}{2^{k_1}} - \frac{\gamma q_{\mathsf{D}}}{2}), \\ t_B &\le t_A + q_{\mathsf{KG}}\hat{u}\tau_{GEN} + \hat{u}\tau_{ENC} + q_H q_{max}(\tau_{ENC} + O(k)), \end{aligned}$$

assuming that IBE *is γ-uniform, and running time of* $\mathsf{Gen}_{\mathsf{IBE}}$ *and* $\mathsf{Enc}_{\mathsf{IBE}}$ *is at most τ_{GEN} and τ_{ENC}, respectively.*

Proof. See Appendix E. □

Extending to KE-CCA Secure IKE. When using the above HIBE for our generic construction of IKE, the resultant IKE guarantees security for an adversary who has limited access to helper keys but still has unlimited access for the number of times he can query the decryption keys.

We can also construct a KE-CCA secure IKE (with a similar restriction) directly from an arbitrary IBE. Next, we give an example. For reader's conveniences, we show a method to construct a KE-CCA secure 1-level IKE from a chosen plaintext secure IBE. Notation that will follow are the same as the notation that we used in our proposed HIBE. First, for a given security parameter k, compute $(s_i, p_i) = \mathsf{PGen}_{\mathsf{IBE}}(1^k)$ and $s_{i,U} = \mathsf{Gen}_{\mathsf{IBE}}(U, s_i, p_i)$ for $0 \leq i \leq u$. Then, $\{s_{i,U}\}_{1 \leq i \leq u}$ is stored in U's PD while s_0 is given to U as his initial decryption key. To encrypt m for U and $\texttt{time}$, $\overline{m}_i$ are picked from $\{0,1\}^n$ for all $i \in F'_{T_0(\texttt{time})} := H(U.T_0(\texttt{time})) \cup \{0\}$, such that $\oplus_{i \in F'_{T_0(\texttt{time})}} \overline{m}_i = m$. Also, r_i are picked from $\{0,1\}^{k_1}$ for all $i \in F'_{T_0(\texttt{time})}$. Then, run $\mathsf{Enc}_{\mathsf{IBE}}(\overline{m}_i || r_i, U, p_i; H_i(m, \overline{m}_i, R)) = \overline{c}_i$ for all $i \in F'_{T_0(\texttt{time})}$, where R denotes concatenation of all r_i for $i \in F'_{T_0(\texttt{time})}$ in increasing order of i. Finally, output $c := \{\overline{c}_i\}_{i \in F'_{T_0(\texttt{time})}}$. It is obvious that the decryption key $\{s_{i,U}\}_{i \in F'_{T_0(\texttt{time})}}$ for $\texttt{time}$ can be derived from the initially distributed keys. Also, KE-CCA security is guaranteed in this scheme. Formal security proof will appear in the full version of this paper.

Chosen Plaintext Secure Construction. Our proposed HIBE uses the method devised to "securely combine" multiple IBEs to achieve chosen ciphertext security. If chosen plaintext security is only what you are looking for, you may not want to use this method, instead, a straightforward multiple encryption of IBE is more suited. Take notice that even if the underlying IBEs are IND-ID-CCA, still, straightforward multiple encryption will not be good enough to construct a chosen ciphertext secure HIBE since there exist a very effective attack that makes it completely insecure.

HIBE from a Weaker IBE. Similarly to our generic construction of KE-CCA secure IKE, a slight modification of the above scheme can enable construction of a IND-wHID-CCA HIBE from IBE with *one-wayness* under chosen plaintext attacks.

Acknowledgemnt

We would like to thank Steven Galbraith, Craig Gentry and Phil MacKenzie for their helpful comments and discussions.

References

[1] S.S. Al-Riyami and K.G. Paterson, "Certificateless public key cryptography," Proc. of Asiacrypt'03, LNCS 2894, Springer-Verlag, pp.452-473, 2003.

[2] J. Baek and Y. Zheng, "Identity-based threshold decryption," Proc. of PKC'04, LNCS 2947, Springer-Verlag, pp.262-276, 2004.

[3] M. Bellare, A. Desai, E. Jokipii and P. Rogaway, "A concrete security treatment of symmetric encryption," Proc. of 38th IEEE Symposium on Foundations of Computer Science (FOCS), pp.394-403, 1997.

[4] D. Boneh and X. Boyen, "Efficient selective-ID secure identity-based encryption without random oracles," Proc. of Eurocrypt'04, LNCS 3027, Springer-Verlag, pp.223-238, 2004.

[5] D. Boneh and X. Boyen, "Secure identity based encryption without random oracles," Proc. of Crypto'04, LNCS 3152, Springer-Verlag, pp.443-459, 2004.

[6] D. Boneh, X. Boyen and E.-J. Goh, "Hierarchical identity based encryption with constant size ciphertext." Proc. of Eurocyrpt'05, to appear.

[7] D. Boneh and M. Franklin, “Identity-based encryption from the Weil pairing,” Proc. of Crypto’01, LNCS 2139, Springer-Verlag, pp.213-229, 2001.

[8] D. Boneh and M. Franklin, “Identity-based encryption from the Weil pairing,” SIAM J. of Computing, vol. 32, no. 3, pp.586-615, 2003 (full version of [7]).

[9] M. Bellare and A. Palacio, “Protecting against key exposure: strongly key-insulated encryption with optimal threshold,” available at `http://eprint.iacr.org/2002/064/` .

[10] C. Cocks, “An identity based encryption scheme based on quadratic residues,” Proc. of IMA Int. Conf. 2001, Coding and Cryptography, LNCS 2260, Springer-Verlag, pp. 360-363, 2001.

[11] R. Canetti, S. Halevi and J. Katz, “A forward secure public key encryption scheme,” Proc. of Eurocrypt’03, LNCS 2656, Springer-Verlag, pp.255-271, 2003.

[12] R. Canetti, S. Halevi and J. Katz, “Chosen-ciphertext security from identity-based encryption,” Proc. of Eurocrypt’04, LNCS 3027, Springer-Verlag, pp.207-222, 2004.

[13] Y. Dodis and J. Katz, “Chosen-ciphertext security of multiple encryption,” Proc. of TCC’05, LNCS 3378, Springer-Verlag, pp.188-209, 2005.

[14] Y. Dodis, J. Katz, S. Xu and M. Yung, “Key-insulated public key cryptosystems,” Proc. of Eurocrypt’02, LNCS 2332, Springer-Verlag, pp.65-82, 2002.

[15] Y. Dodis, M. Franklin, J. Katz, A. Miyaji and M. Yung, “Intrusion-resilient public-key encryption,” Proc. of CT-RSA’03, LNCS 2612, Springer-Verlag, pp.19-32, 2003.

[16] Y. Dodis, M. Franklin, J. Katz, A. Miyaji and M. Yung, “A generic construction for intrusion-resilient public-key encryption,” Proc. of CT-RSA’04, LNCS 2964, Springer-Verlag, pp.81-98, 2004.

[17] Y. Dodis and M. Yung, “Exposure-resilience for free: the hierarchical ID-based encryption case,” Proc. IEEE Security in Storage Workshop 2002, pp.45-52, 2002.

[18] P. Erdös, P. Frankl and Z. Furedi, “Families of finite sets in which no sets is covered by the union of two others,” J. of Combin. Theory Ser. A 33, pp.158-166, 1982.

[19] P. Erdös, P. Frankl and Z. Furedi, “Families of finite sets in which no sets is covered by the union of r others,” Israel Journal of Math., 51, pp.79-89, 1985.

[20] U. Feige, A. Fiat and A. Shamir, “Zero-knowledge proofs of identity,” J. of Cryptology, 1, 2, pp.77-94, 1988.

[21] A. Fiat and A. Shamir, “How to prove yourself: practical solutions to identification and signature problems,” Proc. of Crypto’86, LNCS 263, Springer-Verlag, pp.186-194, 1986.

[22] E. Fujisaki and T. Okamoto, “How to enhance the security of public-key encryption at minimum cost,” Proc. of PKC’99, LNCS 1560, Springer-Verlag, pp.53-68, 1999.

[23] E. Fujisaki and T. Okamoto, “Secure integration of asymmetric and symmetric encryption schemes,” Proc. of Crypto’99, LNCS 1666, Springer-Verlag, pp.537-554, 1999.

[24] C. Gentry, “Certificate-based encryption and the certificate revocation problem,” Proc. of Eurocrypt’03, LNCS 2656, Springer-Verlag, pp.272-293, 2003.

[25] C. Gentry and A. Silverberg, “Hierarchical ID-based cryptography,” Proc. of Asiacrypt’02, LNCS 2501, Springer-Verlag, pp.548-566, 2002.

[26] J. Horwitz and B. Lynn, “Toward hierarchical identity-based encryption,” Proc. of Eurocrypt’02, LNCS 2332, Springer-Verlag, pp.466-481, 2002.

[27] A. Shamir, “Identity-based cryptosystems and signature schemes,” Proc. of Crypto’84, LNCS 196, Springer-Verlag, pp.47-53, 1985.

[28] S. Shinozaki, T. Itoh, A. Fujioka and S. Tsujii, “Provably secure key-updating schemes in identity-based systems,” Proc. of Eurocrypt’90, LNCS 473, Springer-Verlag, pp.16-30, 1990.

[29] B. Waters, “Efficient identity based encryption without random oracles,” Proc. of Eurocrypt’05, to appear.

[30] R. Zhang, G. Hanaoka, J. Shikata and H. Imai, "On the security of multiple encryption or CCA-security + CCA-security = CCA-security?" Proc. of PKC'04, LNCS 2947, Springer-Verlag, pp.360-374, 2004.

[31] Amendment 1 to ITU-T Recommendation X.509-ISO/IEC 95 94-8: 1995, *Information Technology - Open Systems Interconnection - The Directory: Authentication Framework.*

Appendix A: Formal Security Definitions for HIBE

Here, we give a formal security definition of hierarchical identity-based encryption (HIBE). The definition runs parallel with [25] and [26] which is the hierarchical extension of Boneh and Flanklin's IBE [7, 8].

Regarding chosen ciphertext attacks, we address the following three types of oracles: First, is a *key generation oracle* KG which on input $D^{t-1}.D^{t-2}.\cdots.D^{i}$ returns $D^{t-1}.D^{t-2}.\cdots.D^{i}$'s secret $s_{D^{t-1}.D^{t-2}.\cdots.D^{i}}$ for $0 \leq i \leq t-1$. Next, is a *left-or-right encryption oracle* LR which for a given user $D^{*,t-1}.D^{*,t-2}.\cdots.D^{*,0}$ and equal length messages m_0, m_1, returns a *challenge ciphertext* $c := \mathsf{Enc}_{\mathsf{HIBE}}(D^{*,t-1}.D^{*,t-2}.\cdots.D^{*,0}, m_b, p)$ where $b \in \{0,1\}$. This models an encryption request of an adversary who can pick a victim's identity and a message pair of his choice. Finally, the adversary is allowed access to a *decryption oracle* D, which on input $D^{t-1}.D^{t-2}.\cdots.D^{0}$ and a ciphertext c, returns a decryption result of c using $s_{D^{t-1}.D^{t-2}.\cdots.D^{0}}$. This one models the chosen ciphertext attack. Also, if you are considering only chosen plaintext attacks, any access to D is prohibited while accesses to KG and LR remain permitted.

The adversary may query the three oracles adaptively in any order he wants, subject to the restriction that he makes only one query to the left-or-right oracle. Let $D^{*,t-1}.D^{*,t-2}.\cdots.D^{*,0}$ be the user's identifier of this query and let c^* denote the challenge ciphertext returned by the left-or-right oracle in response to this query. The adversary succeeds by guessing the value b. A HIBE is considered secure, if any probabilistic polynomial time adversary has success probability negligibly close to 1/2.

Definition 6 Let $\mathsf{HIBE} = (\mathsf{PGen}_{\mathsf{HIBE}}, \mathsf{Gen}^{i}_{\mathsf{HIBE}}\ (1 \leq i \leq t), \mathsf{Enc}_{\mathsf{HIBE}}, \mathsf{Dec}_{\mathsf{HIBE}})$ be a hierarchical identity-based encryption scheme. Define adversary A's succeeding probability in the above chosen ciphertext attack game as:

$$\mathsf{Succ}_{A,\mathsf{HIBE}} := \Pr[(s,p) \leftarrow \mathsf{PGen}_{\mathsf{HIBE}}(1^k); b \in_R \{0,1\}; b' \leftarrow A^{\mathsf{KG}(\cdot,s,p),\mathsf{LR}(\cdot,\cdot,\cdot,s,p),\mathsf{D}(\cdot,\cdot,s,p)} : b' = b],$$

where any element in $\{(D^{*,t-1}.D^{*,t-2}.\cdots.D^{*,i} : 0 \leq i \leq t-1)\}$ is never asked to KG and A is not allowed to query $\mathsf{D}(D^{*,t-1}.D^{*,t-2}.\cdots.D^{*,0}, c^*, s, p)$ if c is returned by LR. Then, HIBE is

- <u>IND-HID-CCA</u> if for any probabilistic polynomial time adversary A, $|\mathsf{Succ}_{A,\mathsf{HIBE}} - 1/2|$ is negligible (particularly, we call IND-ID-CCA if $t = 1$),
- <u>IND-HID-CPA</u> if for any probabilistic polynomial time adversary A who is not allowed to submit any query to D at all, $|\mathsf{Succ}_{A,\mathsf{HIBE}} - 1/2|$ is negligible (particularly, we call IND-ID-CPA if $t = 1$),
- <u>IND-wHID-CCA</u> if for any probabilistic polynomial time adversary A who is allowed to submit queries to KG at most w times for given layers in the hierarchy, $|\mathsf{Succ}_{A,\mathsf{HIBE}} - 1/2|$ is negligible (A is also allowed to submit unlimited number of queries to KG for at least one layer),
- <u>IND-wHID-CPA</u> if for any probabilistic polynomial time adversary A who is allowed to submit queries to KG at most w times for given layers in the hierarchy, but no query to D is permitted, $|\mathsf{Succ}_{A,\mathsf{HIBE}} - 1/2|$ is negligible (A is also allowed to submit unlimited number of queries to KG for at least one layer).

We next give concrete examples for the above IND-wHID-CCA and IND-wHID-CPA. Suppose we have a 2-level HIBE which includes a root-PKG layer, a sub-PKG layer and a user layer. The sub-PKG layer is set as the special layer in which the number of queries from the adversary is bounded. In the IND-wHID-CCA (or IND-wHID-CPA) setting, an adversary is allowed to ask the sub-PKGs' keys for at most w times while allowing unlimited number of user's decryption keys to be exposed. In addition to KG, the adversary is allowed access to D also when considering the IND-wHID-CCA setting.

Appendix B: Proof of Theorem 1

Here, we prove KE-CCA security for our generic construction. We construct an adversary B who can break at least one of underlying HIBEs in the sense of IND-HID-CPA by using another adversary A who is able to break KE-CCA security of the proposed IKE.

For given public parameters p_h $(1 \leq h \leq 3)$ which corresponds to HIBE_h, respectively, B chooses $i' \in \{0, 1, 2\}$ and computes $\mathsf{PGen}_{\mathsf{HIBE}_h}(1^k) = (s'_h, p'_h)$ for $1 \leq h \leq 3,\ h \neq i' + 1$. Also, B sets (p_1, p'_2, p'_3), (p'_1, p_2, p'_3) and (p'_1, p'_2, p_3) for $i' = 0, 1$ and 2, respectively, as (part of) public parameter of IKE and sends it to A. On A's requests for the oracles, B answers to them following the next simulation:

Simulation of LR. For an LR oracle query $U^*, \mathtt{time}^*, m_0, m_1$ from A, B simulates IKE's LR oracle as follows. First, B sets $a = i' + 1$. For all h $(1 \leq h \leq 3,\ h \neq a)$, B picks $\overline{m}_h \in_R \{0, 1\}^n$ and $r_h \in_R \{0, 1\}^{k_1}$ such that $\oplus_{1 \leq h \leq 3,\ h \neq a} \overline{m}_h = \alpha$ for $\alpha \in_R \{0, 1\}^n$. Also, B sets $\overline{m}_{a,0} = m_0 \oplus \alpha$ and $\overline{m}_{a,1} = m_1 \oplus \alpha$. Then, B picks $r_{a,j} \in_R \{0, 1\}^{k_1}$ for $j = 0, 1$, and sets $U_1^* = U^*$, $U_2^* = U^*.T_0(\mathtt{time}^*)$ and $U_3^* = U^*.T_1(\mathtt{time}^*).T_0(\mathtt{time}^*)$. Also, B sends U_a^*, $(\overline{m}_{a,0}||r_{a,0})$, $(\overline{m}_{a,1}||r_{a,1})$ to B's own LR oracle which corresponds to HIBE_a, and the oracle returns challenge ciphertext c_a^*. Next, B encrypts $(\overline{m}_h||r_h)$ by the encryption algorithm of HIBE_h with p'_h and U_h^*, and produces challenge ciphertexts c_h^* for $1 \leq h \leq 3,\ h \neq a$. Finally, B returns $\langle (c_1^*, c_2^*, c_3^*), \mathtt{time}^* \rangle$ to A. Note that B's goal is to distinguish the underlying plaintext of c_a^*.

Simulation of H_h. For H_h $(1 \leq h \leq 3)$ oracle queries, B returns random values if the query has not been asked before, otherwise B returns the same value as before. If a H_h query is identical to $(m_{b'}, \overline{m}_h, \omega_1, \omega_2, \omega_3)$ such that $\omega_a = r_{a,b'}$ and $\omega_h = r_h$ $(1 \leq h \leq 3,\ h \neq a)$ for some $b' \in \{0, 1\}$ (here, $\overline{m}_a$ means $\overline{m}_{a,b'}$), B outputs $\langle b', a \rangle$ and halts.

Simulation of KG. It is clear that for any of the KG queries, B can answer it perfectly by asking B's own KG oracles. More precisely, on A's request for a KG oracle query $U (\neq U^*)$, B can ask U to B's KG oracle corresponding to HIBE_a, as well as run user-secret generation algorithms of HIBE_h with master key s'_h for $1 \leq h \leq 3,\ h \neq a$. Then, B produces d_0^i for $0 \leq i \leq 2$ by using these results and return (d_0^0, d_0^1, d_0^2).

Simulation of KI. Interestingly, answers to A's KI oracle query can be perfectly simulated by B when i' is the "special level" (see Def. 2) chosen by A. Namely, B can perfectly answer any KI oracle query by using B's own KG oracles which corresponds to HIBE_a and master keys s'_h $(1 \leq h \leq 3,\ h \neq a)$ which correspond to HIBE_h. It should be noticed that the simulation is perfect even if $U = U^*$.

Simulation of D. On A's D query for U and $\langle c, \mathsf{time} \rangle$, B searches for the combinations of A's previous queries made to H_1, H_2, H_3 such that each of the combinations consists of the next three queries ψ_1, ψ_2, ψ_3, where for $1 \leq i \leq 3$, query ψ_i is asked to H_i and ψ_i forms $(m, \overline{m}_i, r_1, r_2, r_3)$ for some n-bit strings m, $\overline{m}_i$ and k_1-bit strings r_1, r_2, r_3 such that $\oplus_{1 \leq i \leq 3} \overline{m}_i = m$ (note that m, r_1, r_2 and r_3 are common for all ψ_1, ψ_2 and ψ_3). If there exists such a combination whose corresponding ciphertext (for U and $\mathtt{time}$) is identical to $\langle c, \mathsf{time} \rangle$, then B returns m. Otherwise, B returns $\perp$.

When A outputs b', B also outputs $\langle b', a \rangle$ as an answer for the IND-HID-CPA game for HIBE_a.

Now, we estimate B's succeeding probability. Simulations of LR, H_h $(1 \le h \le 3)$, and KG are perfect. Simulation of KI fails only when i' is not the special level chosen by A. Therefore, if we let $1/2 + \epsilon_A$ be the succeeding probability of A, then B's secceeding probability can be estimated to be $1/2 + \epsilon_B$ where

$$\epsilon_B \ge \frac{1}{3}(\frac{1}{2} + \epsilon_A - \Pr[H\text{-}Ask]) \cdot \Pr[\neg \mathsf{D}\text{-}Fail] + \frac{2}{3} \cdot \frac{1}{2} - \frac{1}{2},$$

where H-Ask denotes an event that $(m_{\overline{b}}, \overline{m}_h, \omega_1, \omega_2, \omega_3)$ such that $\omega_a = r_{a,\overline{b}}$ and $\omega_j = r_j$ $(j \neq a)$ is asked to H_h for some h, and D-$Fail$ denotes an event that B rejects a D query which should not be rejected.

Since it is informtion-theoretically impossible to find $r_{a,\overline{b}}$, we have $\Pr[H\text{-}Ask] \le 1-(1-1/2^{k_1})^{q_{H_1}+q_{H_2}+q_{H_3}}$, where q_{H_i} $(1 \le i \le 3)$ are the numbers of queries made to H_i. Simulation of D fails only when A submits a ciphertext which should not be rejected, but its corresponding H_i oracle query is not asked. Therefore, $\Pr[\neg\mathsf{D}\text{-}Fail] \ge (1 - \gamma_{max})^{q_{\mathsf{D}}}$, where q_{D} is the number of queries for D, $\gamma_{max} = \max(\gamma_1, \gamma_2, \gamma_3)$, assuming that HIBE_i is γ_i-uniform.

Hence, we have

$$\begin{aligned}\epsilon_B &\ge \frac{1}{3}(\frac{1}{2} + \epsilon_A - (1 - (1 - \frac{1}{2^{k_1}})^{q_{H_1}+q_{H_2}+q_{H_3}}))(1 - \gamma_{max})^{q_{\mathsf{D}}} + \frac{2}{3} \cdot \frac{1}{2} - \frac{1}{2} \\ &\ge \frac{1}{3}\epsilon_A - \frac{1}{3}\frac{q_{H_1} + q_{H_2} + q_{H_3}}{2^{k_1}} - \frac{1}{6}q_{\mathsf{D}}\gamma_{max}.\end{aligned}$$

Also, letting t_A be A's running time, B's running time can be estimated to be t_B where

$$t_B \le t_A + (2q_{\mathsf{KG}} + 5q_{\mathsf{KI}})\tau_{GEN} + q_{H_1} \cdot q_{H_2} \cdot q_{H_3}(3\tau_{ENC} + O(k)),$$

assuming that the number of queries made to KG and KI is q_{KI} and q_{KI}, respectively, and running time of $\mathsf{Gen}^i_{\mathsf{HIBE}_h}$ and $\mathsf{Enc}_{\mathsf{HIBE_h}}$ are at most τ_{GEN} and τ_{ENC}, respectively, for any h and i. Therefore, ϵ_A is negligible if ϵ_B, $1/2^{k_1}$ and γ_{max} are all negligible, and hence, our proposed generic construction of IKE is KE-CCA secure. □

Appendix C: Gentry-Silverberg HIBE [25]

Here, we give a brief review of Gentry-Silverberg HIBE (GS-HBIE) [25]. For simplicity, we consider for the depth of hierarchy being two, i.e. $t = 2$. On input 1^k, a root-PKG set up two cyclic groups G_1 and G_2 of prime order q, and also an efficiently computable mapping $\hat{e} : G_1 \times G_1 \rightarrow G_2$ such that $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ for all $P, Q \in G_1$ and any positive integers a, b. (This does not send all pairs in $G_1 \times G_1$ to the identity in G_2.) The root-PKG chooses an arbitrary generator $P \in G_1$, picks $s \in_R Z_q$, calculates $Q^{gs} := sP$ and sets cryptographic hash functions $H^{gs}_1 : \{0,1\}^* \rightarrow G_1$ and $H^{gs}_2 : G_2 \rightarrow \{0,1\}^{n^{gs}}$, where n^{gs} denotes the size of the message space. Next, the root-PKG keeps master key s and sets the public parameter $p^{gs} := (G_1, G_2, \hat{e}, P, Q^{gs}, H^{gs}_1, H^{gs}_2)$. For a sub-PKG D^1, the root-PKG computes $H^{gs}_1(D^1) = P_{D^1} \in G_1$ and $S_{D^1} := sP_{D^1}$, and gives S_{D^1} to D^1. For a user $D^1.D^0$, D^1 picks $s' \in_R Z_q$ and computes $S_{D^1.D^0} := S_{D^1} + s'P_{D^1.D^0}$, $Q' := s'P$ where $P_{D^1.D^0} := H_1(D^1.D^0)$, and gives $(S_{D^1.D^0}, Q')$ to $D^1.D^0$. When encrypting $m \in \{0,1\}^{n^{gs}}$ for $D^1.D^0$, a sender computes $c := \langle rP, rP_{D^1.D^0}, m \oplus H^{gs}_2(g^r)\rangle$, where $g := \hat{e}(Q, P_{D^1}) \in G_2$ and $r \in_R Z_q$. On receiving $c' = \langle V, V', W\rangle$, $D^1.D^0$ calculates $W \oplus H^{gs}_2(\hat{e}(S_{D^1.D^0}, V)\hat{e}(Q', V')^{-1}) = m$.

Theorem 4 ([8],[25]) *GS-HIBE is* $\mathsf{IND\text{-}HID\text{-}CPA}$ *in the random oracle model assuming that the CBDH problem [7, 8] is hard to solve. Concretely, suppose there is an* $\mathsf{IND\text{-}HID\text{-}CCA}$ *adversary A who can break the above scheme in the sense of* $\mathsf{IND\text{-}HID\text{-}CPA}$ *with probability $1/2 + \epsilon_A$ and runs in time at most t_A. Also, suppose A makes at most q_{KG} queries to* KG *and $q_{H^{gs}_2}$ queries to H^{gs}_2, then there exists another* *adversary B that solves the BDH problem underlying the HIBE with probability of at least ϵ_B and running time t_B, where*

$$\begin{aligned} \epsilon_B &\geq \frac{2\epsilon_A}{q_{H_2^{gs}}}\left(\frac{t}{e(t+q_{\mathsf{KG}})}\right)^t, \\ t_B &= O(t_A), \end{aligned}$$

where e is the base of the natural logarithm and t is the depth of hierarchy.

For more details in IND-HID-CCA security of the GS-HIBE, see [25]. Note that IND-HID-CPA security is sufficient to prove the security of our pairing-based IKE.

Appendix D: Proof of Theorem 2

Here, we prove KE-CCA security of our pairing-based construction under the CBDH assumption [7, 8]. For this, it is sufficient to construct an adversary which can break GS-HIBE [25] by using another adversary which can break our proposed scheme since the security of GS-HIBE is proven under the CBDH assumption.

More specifically, we consutruct an adversary B which for given three public parameters of 1-level, 2-level and 3-level GS-HIBEs, breaks one of them in the sense of IND-HID-CPA by using an adversary A which is able to break KE-CCA security of the proposed scheme. Note that assuming B's advantage is ϵ_B, success probablity ϵ_{cbdh} of solving the CBDH problem becomes

$$\epsilon_{cbdh} \geq \frac{1}{3} \cdot \frac{2\epsilon_B}{q_{H_2^{3gs}}}\left(\frac{3}{e(3+q_{\mathsf{KG}^{3gs}})}\right)^3,$$

where $q_{\mathsf{KG}^{3gs}}$ and $q_{H_2^{3gs}}$ are the total number of queries to the key generation oracles and H_2^{gs} oracles for the three GS-HIBEs.

For given three GS-HIBE public parameters, B chooses $a \in_R \{1,2,3\}$ and picks a-level GS-HIBE public parameter from the given public parameters. For simplicity, we assume $a = 3$. Proofs for $a = 1$ and 2 can be done in a similar manner. Let this GS-HIBE public parameter denote $p^{gs} := (G_1, G_2, \hat{e}, P, Q^{gs}, H_1^{gs}, H_2^{gs})$ (see Appendix C). Then, B chooses $s_1, s_2 \in_R Z_q$. Next, B sets $Q := Q^{gs}$ and gives $p := (G_1, G_2, \hat{e}, P, Q, H_1, H_2, H_3)$ to A as an IKE public parameter, where H_i $(1 \leq i \leq 3)$ are random oracles. Also, we assume $n^{gs} = n + k_1$.

On A's requests for the oracles, B answers to them by the following simulation:

Simulation of LR. For an LR oracle query $U^*, \mathtt{time}^*, m_0, m_1$ from A, B simulates IKE's LR oracle as follows. First, B sets $U_1^* = U^*$, $U_2^* = U^*.T_1(\mathtt{time}^*)$ and $U_3^* = U^*.T_1(\mathtt{time}^*).T_0(\mathtt{time}^*)$, and picks $\mu_0, \mu_1 \in_R \{0,1\}^{k_1}$. Then, B sends U_3^*, $(m_0||\mu_0)$, $(m_1||\mu_1)$ to B's own LR oracle of the 3-level GS-HIBE. B's LR oracle flips a coin $b \in_R \{0,1\}$ and returns a challenge ciphertext $c^{gs} := \langle \mathbf{V}, W \rangle$ where $\mathbf{V} = (rP, rH_1^{gs}(U_2^*), rH_1^{gs}(U_3^*))$ and $W = (m_b||\mu_b) \oplus H_2^{gs}(\hat{e}(H_1^{gs}(U_1^*), rQ^{gs}))$. Finally, B sends a challenge ciphertext $c^* := c^{gs}$ to A.

Simulation of H_i. For H_1 and H_2 oracle queries, B submits the same queries to his H_1^{gs} and H_2^{gs} oracles and returns their answers, respectively. For H_3 oracle queries, B returns random values if the query has not been asked before, otherwise, B returns the same value as before. If a H_3 query is identical to $(m_{b'}||\mu_{b'})$ for some $b' \in \{0,1\}$, B outputs b' and halts. B stores the asked querys and their answers.

Simulation of KG and KI. It is clear that for any KG query, B can perfectly answer to it by asking B's own KG oracle. More precisely, on A's request for a KG oracle query $U(\neq U^*)$, B asks U's key of the 3-level GS-HIBE to B's own KG oracle, and sets $d_0^2 := S_U^{gs} - (s_1+s_2)H_1(U)$, where S_U^{gs} is the answer from B's KG oracle. Also, B computes $d_0^{h-1} = s_h H_1(U)$ for $h = 1, 2$. Then, B returns (d_0^0, d_0^1, d_0^2). Similarly, KI can also be prefectly simulated when the "special level" chosen by A is 2 (see Def. 2).

Simulation of D. On A's D query for U and $\langle c, \mathtt{time}\rangle$, B searches for the combinations of A's previous queries for H_1, H_2, H_3 such that each of the combinations consists of the next five queries $\psi_{1,1}, \psi_{1,2}, \psi_{1,3}, \psi_2, \psi_3$, where queries $\psi_{1,1}, \psi_{1,2}, \psi_{1,3}$ have been asked to H_1 before, and $\psi_{1,1}, \psi_{1,2}, \psi_{1,3}$ form $H_1(U)$, $H_1(U.T_1(\mathtt{time}))$ and $H_1(U.T_1(\mathtt{time}).T_0(\mathtt{time}))$, respectively, for some U and $\mathtt{time}$. Also, queries ψ_2 and ψ_3 have been asked to H_2 and H_3, respectively, before and ψ_2 and ψ_3 form $\hat{e}(Q, \psi_{1,1})^{H_3(\psi_3)}$ and (μ, m), respectively, for some μ and m. If there exists such a combination whose corresponding ciphertext is identical to $\langle c, \mathtt{time}\rangle$, B returns m. Otherwise, B returns $\perp$.

When A outputs b', B also outputs b' as the answer of $\mathsf{IND\text{-}HID\text{-}CCA}$ game of the 3-level GS-HIBE.

Now, we estimate B's succeeding probability. Simulations of LR, H_h $(1 \leq h \leq 3)$, and KG are perfect. Simulation of KI fails only when 2 is not the "special level" chosen by A. Therefore, if we let $1/2 + \epsilon_A$ be the succeeding probability of A, then B's secceeding probability can be estimated to be $1/2 + \epsilon_B$ where

$$\epsilon_B \geq \frac{1}{3}(\frac{1}{2} + \epsilon_A - \Pr[H_3\text{-}Ask]) \cdot \Pr[\neg\mathsf{D}\text{-}Fail] + \frac{2}{3}\cdot\frac{1}{2} - \frac{1}{2},$$

where H_3-Ask denotes an event that $(m_{\overline{b}}|\mu_{\overline{b}})$ is asked to H_3, and D-$Fail$ denotes an event that B rejects a D query which should not be rejected.

Since it is informtion-theoretically impossible to find $\mu_{\overline{b}}$, we have $\Pr[H_3\text{-}Ask] \leq 1 - (1 - 1/2^{k_1})^{q_{H_3}}$, where q_{H_3} is the numbers of queries made to H_3. Simulation of D fails only when A submits a ciphertext which should not be rejected, but its corresponding H_3 oracle query is not asked. Therefore, $\Pr[\neg\mathsf{D}\text{-}Fail] \geq (1 - 1/q)^{q_\mathsf{D}}$, where q_D is the number of queries for D.

Hence, we have

$$\begin{aligned}\epsilon_B &\geq \frac{1}{3}(\frac{1}{2} + \epsilon_A - (1 - (1 - \frac{1}{2^{k_1}})^{q_{H_3}}))(1 - \frac{1}{q})^{q_\mathsf{D}} + \frac{2}{3}\cdot\frac{1}{2} - \frac{1}{2}\\ &\geq \frac{1}{3}\epsilon_A - \frac{1}{3}\frac{q_{H_3}}{2^{k_1}} - \frac{q_\mathsf{D}}{6q}.\end{aligned}$$

Consequently, letting success probablity of solving the CBDH problem denote ϵ_{cbdh}, we have

$$\begin{aligned}\epsilon_{cbdh} &\geq \frac{1}{3}\cdot\frac{2}{q_{H_2}}(\frac{3}{e(3 + q_\mathsf{KG} + q_\mathsf{KI})})^3 \cdot (\frac{1}{3}\epsilon_A - \frac{1}{3}\frac{q_{H_3}}{2^{k_1}} - \frac{q_\mathsf{D}}{6q})\\ &\geq \frac{6}{e^3 q_{H_2}(3 + q_\mathsf{KG} + q_\mathsf{KI})^3}\cdot(\epsilon_A - \frac{q_{H_3}}{2^{k_1}} - \frac{q_\mathsf{D}}{2q}).\end{aligned}$$

Also, if letting t_A be A's running time, then B's running time is estimated to be t_B where

$$t_B \leq t_A + (3q_\mathsf{KG} + 5q_\mathsf{KI})\tau_{EXP} + q_{H_1}^3 q_{H_2} q_{H_3}(\tau_{\hat{e}} + O(k)),$$

assuming the number of queries made to KG and KI are q_KG and q_KI, respectively, time for exponentiation over G_1 is at most τ_{EXP}, and time for pairing computation is at most $\tau_{\hat{e}}$. Therefore, ϵ_A is negligible if ϵ_{cbdh}, $1/q$ and $1/2^{k_1}$ are all negligible, and hence, our pairing-based construction of IKE is KE-CCA secure. □

Appendix E: Proof of Theorem 3

Here, we construct an adversary B who can break the underlying IBE in the sense of $\mathsf{IND\text{-}ID\text{-}CPA}$ by using another adversary A who can break our proposed 2-level HIBE.

B For a given public parameter p of IBE, B sets $p_u := p$ and generates (u, v, w)-cover free family (L, F). Also, B computes $\mathsf{PGen}_{\mathsf{IBE}}(1^k) = (s_i, p_i)$ for $1 \leq i \leq u - 1$, sets $(p_1, \cdots, p_u)$ as (part of) public parameter of HIBE and sends it to A. On A's requests for the oracles, B answers to them by the following simulation:

Simulation of LR. For an LR oracle query $D^{*,1}.D^{*,0}, m_0, m_1$ from A, B simulates HIBE's LR oracle as follows. B asks $D^{*,1}$ to H oracle and computes $\mathsf{Enc}_{\mathsf{IBE}}(\overline{m}_i || r_i, D^{*,1}.D^{*,0}, p_i) = c_i^*$ for $i \in H(D^{*,1}) \backslash \{u\}$, where $r_i \in_R \{0,1\}^{k_1}$ and $\overline{m}_i \in_R \{0,1\}^n$ such that $\oplus_{i \in H(D^{*,1})} \overline{m}_i = \alpha$, where $\alpha \in_R \{0,1\}^n$. Then, B picks $r_{u,0}, r_{u,1} \in_R \{0,1\}^{k_1}$ and submits $D^{*,1}.D^{*,0}$, $(m_0 \oplus \alpha || r_{u_0})$ and $(m_1 \oplus \alpha || r_{u,1})$ to B's own LR oracle to obtain c_u^*. Finally, B sends c_i^* for all $i \in H(D^{*,1})$ to A. Note that B can break IBE only when $u \in H(D^{*,1})$, and therefore, we assume $u \in H(D^{*,1})$ in the rest of this proof.

Simulation of H and H_i. For H and H_i $(1 \leq i \leq u)$ oracle queries, B returns a random value if the query has not been asked before, otherwise, B returns the same value as before. If a H_i query is identical to $(m_{b'}, \overline{m}_i, R_{b'})$ such that $R_{b'}$ is concatenation of all r_i arranged in increasing order of i for $i \in H(D^{*,1})$, where $r_u := r_{u,b'}$ and $\overline{m}_u := m_{b'} \oplus \alpha$, for some $b' \in \{0,1\}$, B outputs b' and halts.

Simulation of KG. KG can be simulated as follows. On A's request for KG oracle query $D^1.D^0$, B answers $s_{D^1.D^0}$ by computing $\mathsf{Gen}_{\mathsf{IBE}}(D^1.D^0, s_i, p_i) = s_{i,D^1.D^0}$ for all $i \in F_{D^1} \backslash \{u\}$ and querying $D^1.D^0$ to B's own KG oracle to obtain $s_{u,D^1.D^0}$. While, for A's request for KG oracle query D^1, B answers $s_{D^1} := \{s_i\}_{i \in F_{D^1}}$ if $u \notin F_{D^1}$, otherwise, B outputs random b' and halts. Such a simulation fails when A asks D^1 such that $u \in F_{D^1}$. It should be noted that from the nature of (u, v, w)-CFF, A cannot obtain at least one of master keys of underlying IBEs, assuming that A is allowed to submit at most w queries to KG.

Simulation of D. On A's D query c and $D^1.D^0$, B searches for the combinations of A's previous queries for $H_1, \cdots, H_u$ such that each of the combinations consists of $\hat{u}$ queries ψ_i for all $i \in H(D^1)$, query ψ_i has been asked to H_i and that ψ_i forms $(m, \overline{m}_i, R)$ for some n-bit strings m, $\overline{m}_i$ and $\hat{u}k_1$-bit string R, such that $\oplus_{j \in H(D^1)} \overline{m}_j = m$ (note that m and R are common to all of these queries). Then, B splits R into k_1-bit strings r_i for $i \in H(D^1)$ such that R is a concatenation of all r_i arranged in increasing order of i for $i \in H(D^1)$. If there exists such a combination of queries whose corresponding ciphertext (for $D^1.D^0$) is identical to c, then B returns m. Otherwise, B returns $\perp$.

If A outputs b', then B also outputs b' as an answer for the $\mathsf{IND\text{-}ID\text{-}CPA}$ game for IBE.

Now, we estimate B's succeeding probability. Simulations of LR, H_i $(1 \leq i \leq u)$ and H are perfect. Therefore, if we let $1/2 + \epsilon_A$ be the succeeding probability of A, then B's secceeding probability can be estimated to be $1/2 + \epsilon_B$ where

$$\begin{aligned} \epsilon_B \quad \geq \quad & \Pr[\mathit{Embed}] \cdot \Pr[\neg \mathsf{KG}\text{-}\mathit{Fail}] \cdot (\frac{1}{2} + \epsilon_A - \Pr[H\text{-}\mathit{Ask}]) \cdot \Pr[\neg \mathsf{D}\text{-}\mathit{Fail}] \\ & + (1 - \Pr[\mathit{Embed}] \cdot \Pr[\neg \mathsf{KG}\text{-}\mathit{Fail}]) \cdot \frac{1}{2} - \frac{1}{2}, \end{aligned}$$

where Embed denotes an event that $u \in H(D^{*,1})$, $\mathsf{KG}\text{-}\mathit{Fail}$ denotes an event that A asks D^1 such that $u \in H(D^1)$, $H\text{-}\mathit{Ask}$ denotes an event that $(m_{\overline{b}}, \overline{m}_i, R_{\overline{b}})$ is asked to H_i for some i, and $\mathsf{D}\text{-}\mathit{Fail}$ denotes an event that B rejects a D query which should not be rejected.

It is clear that $\Pr[Embed] \geq \#\{F_i | u \in F_i \in F\}/\#F = \hat{u}/u$. Also, $\Pr[\neg\mathsf{KG}\text{-}Fail] \geq 1/u$ since from the nature of (u, v, w)-CFF, there exists at least one underlying IBE whose master key has not been exposed to A. Furthermore, since it is informtion theoretically impossible to find $r_{u,\bar{b}}$, we have $\Pr[H\text{-}Ask] \leq 1 - (1 - 1/2^{k_1})^{q_{all}}$, where $q_{all} := \sum_{1 \leq i \leq u} q_{H_i}$ and q_{H_i} is the number of queries to H_i $(1 \leq i \leq u)$. Finally, simulation of D fails only when A submits a ciphertext which should not be rejected, but its corresponding H_i oracle query is not asked. Therefore, $\Pr[\neg\mathsf{D}\text{-}Fail] \geq (1-\gamma)^{q_{\mathsf{D}}}$ where q_{D} is the number of queries to D, assuming that IBE is γ-uniform.

Hence, we have

$$\begin{aligned} \epsilon_B &\geq \frac{\hat{u}}{u}\left(\frac{1}{2} + \epsilon_A - \left(1 - \left(1 - \frac{1}{2^{k_1}}\right)^{q_{all}}\right)\right) \cdot \frac{1}{u} \cdot (1-\gamma)^{q_{\mathsf{D}}} + \left(1 - \frac{\hat{u}}{u} \cdot \frac{1}{u}\right)\frac{1}{2} - \frac{1}{2} \\ &\geq \frac{\hat{u}}{u^2}\left(\epsilon_A - \frac{q_{all}}{2^{k_1}} - \frac{\gamma q_{\mathsf{D}}}{2}\right) \end{aligned}$$

Also, letting t_A be A's running time, B's running time is estimated to be t_B where

$$t_B \leq t_A + q_{\mathsf{KG}}\hat{u}\tau_{GEN} + \hat{u}\tau_{ENC} + q_H \cdot q_{max}(\tau_{ENC} + O(k)),$$

assuming that $q_{max} := \max_{\{i_1,\cdots,i_{\hat{u}}\} \subseteq \{1,\cdots,u\}}(\prod_{i \in \{i_1,\cdots,i_{\hat{u}}\}} q_{H_i})$ and the number of queries made to KG and H are q_{KG} and q_H, respectively, and running time of $\mathsf{Gen}_{\mathsf{IBE}}$ and $\mathsf{Enc}_{\mathsf{IBE}}$ is at most τ_{GEN} and τ_{ENC}, respectively.

Hence, ϵ_A is negligible if ϵ_B, $1/2^{k_1}$ and γ are all negligible, and therefore, our proposed generic construction of HIBE is $\mathsf{IND}\text{-}w\mathsf{HID}\text{-}\mathsf{CCA}$ with a restriction that an adversary is not allowed to ask KG for more than w times. □