Identity-Based Encryption with Non-Interactive Key Update

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

December 2, 2004

Abstract

In this paper, we discuss non-interactive updating of private keys in identity-based encryption (IBE). IBE is a public key cryptosystem where a public key is an arbitrary string. Key revocation in IBE, in practice, is unavoidable and also a problem that cannot be bypassed. Our main contribution of this paper is to propose a novel constructions of IBE in which the private key is renewed without having to make any changes to its public key, i.e. user's identity. We achieve this by extending the hierarchical IBE (HIBE). Regarding security, in addition to chosen ciphertext attack, we address semantic security for a very strong attack environment which models all possible types of key exposures in the random oracle model. Straightforward extension of the HIBE, however, is completely insecure for such an attack model. Moreover, we show a method of constructing (partially collusion resistant) HIBE from arbitrary IBE in the random oracle model. By the combination of this method and the technique used in the above scheme we can construct an IBE with non-interactive key update from 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) [25], where any public key is determined as an arbitrary string, e.g. user's name, e-mail address, etc. Identity-based system can simplify certificate management in public key infrastructures.

In this paper, we address a crucial but heretofore undiscussed issue; non-interactive updating of user's private key in IBE. Revocation and renewal of private keys is a problem that cannot be bypassed in practice. In conventional public key schemes, certification revocation list (CRL) is generally utilized to minimize the damage due to key compromisation, and here, users become aware of the voided key by referring to the CRLs, and abort it, if necessary. In IBE, however, some caution must be taken when straightforward implementation of CRL is carried out, as it may be inefficient to do so, since invalidating a user's identity-based key and terminating the link between the user identity and the public key implies loosing one of principal advantages of IBE. For example, one of suitable application of IBE is of a mobile phone network system, in which case, phone number represents the identity unique to each user. It will be both simple and convenient for the mobile phone users to be able to communicate with each other by using their phone numbers only. The problem however arises when the user needs to renew his key. It is necessary for him to be able to change his private key without changing his phone number and this will be the main subject of our discussion.

[*] Network Management Development Department, NTT DoCoMo, Inc. yamamotoyumi@nttdocomo.co.jp

[†] Institute of Industrial Science, the University of Tokyo. hanaoka@imailab.iis.u-tokyo.ac.jp

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

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

Our Results. Our main contribution of this paper is to study renewal of private keys in IBE. We begin our discussion by looking into the difficulty in constructing an IBE with the most useful and essential property as key revocation based on the conventional model of IBE. Then, we re-construct the model of IBE and show a new generalized method that can efficiently renew the private keys in IBE. Based on this model, we show a construction of IBE with non-interactive key update that *lets the user update his key on his own without the help of a central authority, and most importantly, without changing his identity.* In this scheme, similar to [13], we assume a private device which is different from the main hardware where the actual decryption is carried out. Private device is not connected to the network and assumes only a small storage and computation capacity. In stead, a private device stores a secret *helper key* which assists the user to update his decryption key at each and fixed time period and only at times he needs his private key to be renewed. All secret operations are still 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 [13]. Here, we mean "strongly" by system guaranteeing security even when the private device is physically compromised. Our scheme is different from [13] in that the private device can be divided into multiple levels forming a hierarchical structure, and hence, its security is improved.

In brief, our proposed scheme can be said as an extension of hierarchical identity-based encryption schemes (HIBE) [24, 23]. Straightforward extension of a HIBE, however, will be completely vulnerable in our attack model. In this paper, we propose two different secure constructions of IBE with non-interactive key update. One is a generic construction built from HIBE. More precisely, we bring an arbitrary (chosen plaintext secure) HIBE to construct a chosen ciphertext secure IBE with non-interactive key update. Also, the underlying assumption of such scheme is 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 updating. Second, we show a specific construction of IBE with non-interactive key update, and its efficiency is improved compared to the above generic construction. In addition to the proposal of these two constructions, we also show a technique that enables us to construct a (partially collusion resistant) HIBE from an arbitrary IBE. Moreover, this result can be further applied to our generic construction of IBE with non-interactive key update to convert an arbitrary IBE to have key-updating property as well. Note that we mean "partial collusion resistant" in that we argue based on the security definition in [24] and not in [23]. Security of our schemes is proved in the random oracle model.

Applications: Mobile Phone Scenario. Requiring a "private device" may seem inconvenient at first glance, however, in practice, it may not be the case. Let's go back to the example (see **Background.**) of a mobile phone system. If you are a mobile phone user, then it is your routine job to re-charge your battery once every now and then. Now, assume a battery charger that "intelligently" functions also as the private device. Such a device can provide a convenient way of renewing the private key as well as re-charging the battery at the same time. Security of this system is also guaranteed even if you lose your battery charger. As you can see, this is already a practical and attractive application of IBE, but its security can be further improved by constructing a hierarchical version of this scheme. We assume the user's private devices to be structured hierarchically into two levels. We let the first level private device, the battery charger, be the one that does the actual decryption key updating, and second level private device does the updating of the helper key of the first level private device. For example, decryption key can be updated every day while the helper key of the battery charger every 2 or 3 months. This way, second level private device can be kept somewhere safer while keeping the battery charger in places more handy. Furthermore, our schemes even guarantee security against an adversary who obtains any of the private devices including the one in the second level.

Related Works. Issue of revocability of private keys in identity-based schemes was initially discussed by Shinozaki, Itoh, Fujioka and Tsujii [26], however, it required prior communication for revocation and did not show advantage over conventional public key schemes which also required prior interaction between the user and the certificate authority. Furthermore, their scheme was specific to Fiat-Shamir identification scheme [19, 20] and could not generally be applied to identity-based schemes. Recently, Baek and Zheng [2] proposed a threshold decryption method for IBE which prevents the keys from getting exposed rather than dealing with cases after key exposure has occured. In [16], Dodis and Yung proposed an interesting method for refreshment of private keys in HIBE, and their scheme can efficiently deal with the problem of *gradual* key exposure in which a secret key is assumed to be slowly compromised over time.

Boneh and Franklin in their paper ([6], Section 1.1.1) showed the first generalized method for key revocation for identity-based encryption schemes. In their scheme, a privileged Private Key Generator (PKG) generates each user's private 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 which may be 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 identity-based schemes (i.e. low costs in communication and computation). Further, it needs to work out a way to establish secure channel between the PKG and the user. For instance, it needs to compensate for additional and requires complicated transactions if the secret information required to setup a secure channel is exposed. Moreover, forward security must also be considered. Hence, it is not desirable to require frequent communication via secure channel with the PKG in identity-based schemes.

On the other hand, as a solution to key exposure and revocation problem in conventional public key systems, Dodis, Katz, Xu and Yung [13] proposed a scheme called *key-insulated encryption*. Their scheme assumes a private device in which a *helper key* is stored. The helper key assists the user to renew his decryption key by generating the secret information needed to update the decryption key. Here, the public key is fixed. In [14, 15], Dodis, Franklin, Katz, Miyaji and Yung further improved [13] with forward secrecy as an additional property. Notice that being able to renew the private key without having to make changes to the corresponding public key as seen in the key-insulated encryption scheme, is the very technique, desired in IBE. Although, a possible harmonization of the advantages of these two schemes, or an identity-based version of a (strongly secure) key-insulated encryption scheme, has never been constructed. There also has never been a construction built of a hierarchical version of key-insulated encryption where the private device 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, [22, 1], but they are all looked from the point of view of conventional public key infrastructure.

We mentioned earlier that our scheme can be seen as an extension of HIBE [24, 23, 4]. HIBE is a powerful cryptographic tool and plays an important role and also forms basis of various cryptographic techniques, e.g. [10]. Moreover, the only methods known to construct HIBE [24, 23, 4] are ones that require specific assumptions in elliptic curve cryptography, e.g. Bilinear Diffie-Hellman (BDH), as the underlying assumption, and therefore, lacks flexibility in the selection of underlying assumption. (Note that for an identity-based encryption scheme, there is also a construction based on quadratic residuosity problem [9].) As you can realize, building a novel HIBE construction is hard and is one of important research topics in this area, especially, an open problem for a generic construction of HIBE based on an arbitrary IBE.

2 Model and Definitions

Overview of the Model. Before we get into discussing the construction of our model of IBE with non-interactive key update, recall we said earlier that it was impossible to construct an IBE with useful and essential property as key revocation based on the model of conventional IBE. To be more specific, in a conventional IBE that only uses the public parameter distributed at system set up phase and the user's identity to encrypt a message, it is impossible for the user to *immediately* revoke and renew his private key *only* at time he loses his key without loosing the advantage of IBE in terms of communication cost. Hence, there is a need to build a new IBE model for an IBE to have key-updating property.

As already mentioned, the model of IBE with key renewal shown in [6] requires a secure channel between the user and the PKG that needs to be available at all times, moreover, PKG in their model needs to renew the users' private keys at fixed and also frequent time intervals. This model is simple and generally practical for some types of applications, however, there are other cases where frequent communications via constantly available secure channel between the user and the PKG is neither preferred nor available.

In our new model of IBE with non-interactive key update, we introduce a *private device* which stores a helper key used to renew the decryption key. This model allows the decryption keys to be renewed at regular time intervals without having to require any kind of interactions between other entities. Furthermore, we consider a hierarchical construction of our model by letting the helper key stored in each level of the hierarchy to be renewed using the helper key of a level higher. (See **Applications: Mobile Phone Scenario** in Sec. 1.) Our model is, in fact, regarded as both a hierarchical and also an identity-based extension of a key-insulated encryption [13]. Similarly to [13], we address *random-access key updating*, namely, it allows one-step renewal of current private key to any of the private keys of any time period (even the past keys). Such function lets any ciphertext of any time period to be decrypted at any time.

Model. We assume the user's private devices 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 private device. The actual decryption takes place at the user's terminal, the 0-level private device, in which 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 $i = 2, \cdots, \ell$. So the data generated by the first-level helper key is used to renew the user's decryption key. For simplicity, we consider the specific case $\ell = 2$, where the first and second level private devices correspond to the battery charger and the device that updates the battery charger's helper key, respectively, in the mobile phone scenario. (Note that our scheme can easily be generalized for arbitrary $\ell \geq 1$.) Also, let $T_0(\cdot)$ and $T_1(\cdot)$ be functions which map a *time* to the corresponding time periods used for decryption keys and first-level helper keys, respectively. For example, in this scenario, $T_0(\texttt{2004/Aug./26th/17}:\texttt{00}) = \texttt{2004/Aug./26th}$ and $T_1(\texttt{2004/Aug./26th/17}:\texttt{00}) = \texttt{2004/Jul.-Sep.}$, assuming that user's decryption key is updated every day, and the first-level helper key is updated every 3 months. In addition, let $T_2(\cdot)$ be a function such that for all $\texttt{time}$, $T_2(\texttt{time}) = 0$. In our model, at time $\texttt{time}$, a user can update his decryption key only if his first-level helper key is valid for time period $T_1(\texttt{time})$. The first-level helper key can be updated at any time.

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

$\underline{\mathsf{PGen}_{\mathsf{IKE}}.}$ 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.

$\underline{\mathsf{Gen}_{\mathsf{IKE}}.}$ The *user-secret generation algorithm* $\mathsf{Gen}_{\mathsf{IKE}}$ takes as input s, p and a user's identity U, and outputs U's initial private keys (d^0_0, d^1_0, d^2_0), where d^0_0 is U's initial decryption key, and d^i_0 $(i = 1, 2)$ are stored in U's i-th level private device as the initial i-th helper key.

$\underline{\Delta\text{-}\mathsf{Gen}^i_{\mathsf{IKE}}.}$ A helper key stored in the first (resp. second) level private device and the $\Delta\text{-}\mathsf{Gen}^1_{\mathsf{IKE}}$ (resp. $\Delta\text{-}\mathsf{Gen}^2_{\mathsf{IKE}}$) are used to generate the data required to renew the decryption key (resp. a first-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 and $\mathtt{time}$ as inputs, and outputs key-update information $\delta^{i-1}_{T_{i-1}(\mathtt{time})}$ only if $t = T_i(\mathtt{time})$.

$\underline{\mathsf{Upd}^i_{\mathsf{IKE}}}$. U's decryption key (resp. U's first-level helper key), key-update information $\delta^0_{T_0(\mathtt{time})}$ (resp. $\delta^1_{T_1(\mathtt{time})}$) and $\mathsf{Upd}^1_{\mathsf{IKE}}$ (resp. $\mathsf{Upd}^2_{\mathsf{IKE}}$) are used to generate U's decryption key (resp. U's first-level helper key) for $\mathtt{time}$. More specifically, for $i = 1, 2$, the *key-update information generation algorithm* $\mathsf{Upd}^i_{\mathsf{IKE}}$ takes d^{i-1}_t, p and $\delta^{i-1}_{T_{i-1}(\mathtt{time})}$ for any t, and outputs a new key $d^{i-1}_{T_{i-1}(\mathtt{time})}$ for time period $T_{i-1}(\mathtt{time})$.

$\underline{\mathsf{Enc}_{\mathsf{IKE}}}$. The *encryption algorithm* $\mathsf{Enc}_{\mathsf{IKE}}$ takes inputs m, U, p and $\mathtt{time}$, where m is a plaintext, U is the user identity and $\mathtt{time}$ indicates the time at which m is encrypted, and outputs ciphertext $\langle c, \mathtt{time}\rangle$.

$\underline{\mathsf{Dec}_{\mathsf{IKE}}}$. The *decryption algorithm* $\mathsf{Dec}_{\mathsf{IKE}}$ takes $\langle c, \mathtt{time}\rangle$, d^0_t and p as inputs, and outputs m or $\perp$ where $\perp$ indicates failure. $\mathsf{Dec}_{\mathsf{IKE}}$ correctly recovers the plaintext only if $t = T_0(\mathtt{time})$.

Security Definition. The security of an IKE is based on an assumption that it is difficult for an adversary to illegally obtain all of the victim user's keys which are managed in different manners. That is, since the victim's private devices in which the helper keys are stored are not connected to the network, the adversary needs to physically steal each private device to obtain the key. It also becomes harder to obtain the helper keys of the private devices as the level goes higher. Even if we assume that the adversary may not succeed in obtaining all of the keys simultaneously, we still need to consider the case of partial robbery (even the helper key in the highest level can be stolen). Therefore, in our attack model, in addition to the standard IND-ID-CCA setting [6, 7], an adversary can access even to the victim's decryption keys and helper keys except for those that can trivially let the adversary guess what the target decryption key is from the definition of IKE. Next, we give some examples of key exposures which we mean by our security definition.

<u>Examples of Key Exposures.</u> We consider a 2-level IKE with a user's second-level helper key which is never updated, and a first-level helper key and a decryption key which are renewed every three months and a day, respectively. Then, any ciphertext for 2004/Dec./31st should not be decrypted by dishonest means even for the following cases:

1. Exposure of the victim's first-level helper keys for 2004/Jan.-Mar., $\cdots$, 2004/Jul.-Sep. and decryption keys for 2004/Jan./1st, $\cdots$, 2004/Dec./30th
2. Exposure of the victim's second-level helper key and decryption keys for 2004/Jan./1st, $\cdots$, 2004/Dec./30th
3. Exposure of the victim's second-level helper key and first-level helper keys for 2004/Jan.-Mar., $\cdots$, 2004/Oct.-Dec.

It should be noticed that in case of the exposure of the victim's first-level helper key for 2004/Oct.-Dec. and decryption key for 2004/Dec./30th, the decryption key for 2004/Dec./31st can easily be computed by 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, an adversary is allowed to have 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 private keys (d^0_0, d^1_0, d^2_0). 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 a *challenge ciphertext* $c :=$ $\mathsf{Enc}_{\mathsf{IKE}}(m_b, U, p, \mathsf{time})$ where $b \in_R \{0, 1\}$. This models encryption requests from the 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 its corresponding decryption key d^0_t such that $t = T_0(\mathtt{time})$. 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 them, we introduce a *key issue oracle* $\mathsf{KI}(\cdot, \cdot, \cdot, s, p)$ which on input i, U and $\mathtt{time}$, returns d^i_t, where $t = T_i(\mathtt{time})$. This models partial exposure of honest user's keys including 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 let him compute 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 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 any key of any user to KI 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, $|\mathsf{Succ}_{A,\mathsf{IKE}} - 1/2|$ is negligible. (Note that the "special level" means level of an uncompromised private device of U^*.)

Exposure of Key-Updating Information. If the security of the IKE is examined in a closer manner, exposure of key-update information should also be addressed. However, if $\delta^i_{T_i(\mathtt{time})}$ can be computed from $d^i_{T_i(\mathtt{time})}$ and d^i_t for any $\mathtt{time}$ and t, then, exposure of key-update information can be simulated by the use of KI. Hence, the above security definition is sufficient even against exposure of the key-update information if this property holds. All of our constructions satisfy this property.

3 Insecurity of Straightforward IKE from HIBE

As already mentioned, there is some likeness between HIBE and our IKE, however, we'd like to note again that it is difficult to straightforwardly construct a KE-CCA secure IKE from a HIBE. In this section, we clarify the relation between HIBE and IKE.

Brief Review of HIBE. HIBE is a technique which distributes the workload of the role of PKG in IBE in the issuing of user private keys which is considered to be a burdensome task by organizing the PKGs in a hierarchical tree structure. Here, we give the definition of HIBE and its security. This definition runs parallel with [23] which is the hierarchical extension of Boneh and Flanklin's [6, 7]. 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 the depth of the hierarchy. The user's ancestors in the hierarchy tree are the root-PKG and users/sub-PKGs whose identities are $\{(D^{t-1}.D^{t-2}.\cdots.D^i : 0 \le i \le t-1)\}$.

Definition 3 (HIBE) A *t-level hierarchical identity-based encryption (HIBE)* HIBE consists of $3 + t$ algorithms: $\mathsf{HIBE} = (\mathsf{PGen}_{\mathsf{HIBE}}, \mathsf{Gen}^i_{\mathsf{HIBE}}\ (1 \le i \le t), \mathsf{Enc}_{\mathsf{HIBE}}, \mathsf{Dec}_{\mathsf{HIBE}})$ which 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* $\mathsf{Gen}^t_{\mathsf{HIBE}}$ takes as input D^{t-1}, s and p, and outputs D^{t-1}'s

key $s_{D^{t-1}}$. Similarly, for $2 \leq i \leq t$, $\mathsf{Gen}_{\mathsf{HIBE}}^{t-i+1}$ takes as input $D^{t-1}.D^{t-2}.\cdots.D^{t-i}$, $s_{D^{t-1}.D^{t-2}.\dots.D^{t-i+1}}$ and p, and outputs $D^{t-1}.D^{t-2}.\cdots.D^{t-i}$'s key $s_{D^{t-1}.D^{t-2}.\dots.D^{t-i}}$. Here, for $1 \leq i \leq t-1$, $s_{D^{t-1}.D^{t-2}.\dots.D^{t-i}}$ is the sub-master key which enables $D^{t-1}.D^{t-2}.\cdots.D^{t-i}$ to generate his descendant's keys, and $s_{D^{t-1}.D^{t-2}.\dots.D^{0}}$ 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 as input m, $D^{t-1}.D^{t-2}.\cdots.D^{0}$ and p, where m is a plaintext, $D^{t-1}.D^{t-2}.\cdots.D^{0}$ is the receiver's identity, and outputs a ciphertext c.

$\underline{\mathsf{Dec}_{\mathsf{HIBE}}.}$ The *decryption algorithm* $\mathsf{Dec}_{\mathsf{HIBE}}$ takes as input c, $s_{D^{t-1}.D^{t-2}.\dots.D^{0}}$ and p, and outputs m or $\perp$ which means failure. $\mathsf{Dec}_{\mathsf{HIBE}}$ recovers the plaintext only if c has correctly been encrypted 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 them to a *left-or-right encryption* oracle LR which returns a ciphertext of m_b such that $b \in_R \{0,1\}$ for a target user. The adversary may also have access to a *decryption oracle* D which returns decryption results of any ciphertext except for the challenge ciphertext from LR, and a *key generation oracle* KG which exposes any entity's key except for the target's and its ancestors'. A HIBE is *secure* if any adversary can correctly guess the value of b with a probability at most $1/2 + neg$ such that neg is negligible. Especially, (secure) HIBE is $\mathsf{IND\text{-}HID\text{-}CCA}$ (resp. $\mathsf{IND\text{-}HID\text{-}CPA}$) if unlimited access to D and KG (resp. only KG) is allowed for the adversary [23]. Also, (secure) HIBE is $\mathsf{IND}\text{-}w\mathsf{HID\text{-}CCA}$ (resp. $\mathsf{IND}\text{-}w\mathsf{HID\text{-}CPA}$) if unlimited access (resp. no access) to D is allowed, while the number of queries to KG is bounded as follows [24]. For at least one level of 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.

Insecurity of HIBE as IKE. 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 `time` are set as $d_0^2 := \mathsf{Gen}_{\mathsf{HIBE}}^3(U,s,p)$ and $d_{T_i(\mathtt{time})}^i := \mathsf{Gen}_{\mathsf{HIBE}}^{i+1}(T_i(\mathtt{time}), d_{T_{i+1}(\mathtt{time})}^{i+1}, p)$ 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 [24] as well.

Above method described, that of a straightforward construction of IKE from HIBE, at first glance, may seem secure, but it is, in fact, not. (In other words, this construction is not KE-CCA secure.) For example, this construction does not provide security against 2. and 3. of the <u>Examples of Key Exposures.</u> in the previous section. Namely, if the first-level private device (e.g. the battery charger) is stolen at `2004/Oct./1st/0:00`, then confidentiality of all the ciphertexts which are generated during the period `2004/Oct.-Dec.` will all be lost. Moreover, exposure of the second-level helper key implies completely compromising the security for any time period just because of this one key. Hence, it is not secure.

4 Generic Construction

Basic Idea. As mentioned in the previous section, a straightforward construction from a HIBE is a problem as it looses its security when a user's private device is compromised. The trick behind our generic construction is to bring three distinct HIBEs as composites to construct an IKE. In our proposed generic construction of an IKE, each of HIBE plays a part in guaranteeing the security against different types of key exposures, and even if a private device is compromised, system remains secure.

Namely, a careful consideration and secure integration of the embedded HIBEs is indispensable in constructing a KE-CCA secure IKE. In order to achieve this, here, we extend *multiple encryption* technique proposed in [27] to achieve KE-CCA security. However, it should be noticed that the original [27] scheme is applied only to standard public key encryption, and that it cannot be straightforwardly adapted to our proposed scheme.

Construction. Here, we show a generic construction of KE-CCA secure IKE from any HIBE that only has *chosen plaintext security*, i.e. $\mathsf{IND\text{-}HID\text{-}\underline{CPA}}$ (See Appendix A).

Generic Construction of KE-CCA secure IKE

Set up h-level HIBE $\mathsf{HIBE}_h = (\mathsf{PGen}_{\mathsf{HIBE}_h}, \mathsf{Gen}^i_{\mathsf{HIBE}_h}\ (1 \le i \le h), \mathsf{Enc}_{\mathsf{HIBE}_h}, \mathsf{Dec}_{\mathsf{HIBE}_h})$ for $1 \le h \le 3$. Then, 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.

$\underline{\mathsf{PGen}_{\mathsf{IKE}}}$: For $1 \le h \le 3$, run $\mathsf{PGen}_{\mathsf{HIBE}_h}(1^k) = (s_h, p_h)$, and set cryptographic hash functions $H_h : \{0,1\}^{2n+3k_1} \to \mathcal{COIN}$, where n denotes the size of a message of IKE, and $\mathcal{COIN}$ 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 (for simplicity, we assume for all HIBE_h $(1 \le h \le 3)$, spaces of internal coin-flipping and messages are $\mathcal{COIN}$ and $\{0,1\}^{n+k_1}$, respectively). The security analysis will view H_h $(1 \le h \le 3)$ as random oracles. Then, output $s := (s_1, s_2, s_3)$ and $p := (p_1, p_2, p_3, H_1, H_2, H_3)$.

$\underline{\mathsf{Gen}_{\mathsf{IKE}}}$: For input s, p and U, parse $s = (s_1, s_2, s_3)$ and $p = (p_1, p_2, p_3, H_1, H_2, H_3)$, and for $1 \le h \le 3$, run $\mathsf{Gen}^h_{\mathsf{HIBE}_h}(U, s_h, p_h) = s_{h,U}$. Then, set $d^0_0 = (s_{1,U}, 0, 0)$, $d^1_0 = (s_{2,U}, 0)$ and $d^2_0 = s_{3,U}$, and output U's initial keys (d^0_0, d^1_0, d^2_0).

$\underline{\Delta\text{-}\mathsf{Gen}^i_{\mathsf{IKE}}}$: For input d^i_t, p and $\mathtt{time}'$, parse $d^i_t = (\sigma_{i+1}, \cdots, \sigma_3)$, and run $\mathsf{Gen}^i_{\mathsf{HIBE}_h}(T_{i-1}(\mathtt{time}'), \sigma_h, p_h) = \sigma'_h$ for $i + 1 \le h \le 3$. Then, output $\delta^{i-1}_{T_{i-1}(\mathtt{time}')} := (\sigma'_{i+1}, \cdots, \sigma'_3)$.[1]

$\underline{\mathsf{Upd}^i_{\mathsf{IKE}}}$: For input d^{i-1}_t, p and $\delta^{i-1}_{T_{i-1}(\mathtt{time}')}$, parse $d^{i-1}_t = (\sigma_i, \cdots, \sigma_3)$ and $\delta^{i-1}_{T_{i-1}(\mathtt{time}')} = (\sigma'_{i+1}, \cdots, \sigma'_3)$, and output $d^{i-1}_{T_{i-1}(\mathtt{time}')} := (\sigma_i, \sigma'_{i+1}, \cdots, \sigma'_3)$.

$\underline{\mathsf{Enc}_{\mathsf{IKE}}}$: For input m, U, p and time, pick $\overline{m}_1, \overline{m}_2, \overline{m}_3 \in \{0,1\}^n$ uniformly at random such that $\oplus_{1 \le i \le 3} \overline{m}_i = m$. Also, pick $r_1, r_2, r_3 \in_R \{0,1\}^{k_1}$. Then, by letting $R_h := H_h(m, \overline{m}_h, r_1, r_2, r_3)$, for $1 \le h \le 3$ compute

$$c_h = \mathsf{Enc}_{\mathsf{HIBE}_h}(\overline{m}_h || r_h, V_h, p_h; R_h),$$

where $V_1 := U$, $V_2 := U.T_0(\mathtt{time})$ and $V_3 := U.T_1(\mathtt{time}).T_0(\mathtt{time})$. ("$;R$" denotes internal coin-flipping with randomness R.) Finally, set $c = (c_1, c_2, c_3)$, and output $\langle c, \mathtt{time} \rangle$ as ciphertext.

$\underline{\mathsf{Dec}_{\mathsf{IKE}}}$: For input $\langle c', \mathtt{time} \rangle$, d^0_t and p, output $\perp$ if $t \ne T_0(\mathtt{time})$. Else, parse $c' = (c'_1, c'_2, c'_3)$ and $d^0_t = (\sigma_1, \sigma_2, \sigma_3)$. Next, compute

$$\mathsf{Dec}_{\mathsf{HIBE}_h}(c'_h, \sigma_h, p_h) = (\overline{m}'_h || r'_h)$$

for $1 \le h \le 3$, and compute $\oplus_{1 \le h \le 3} \overline{m}'_h = m'$. Then, by letting $R'_h := H_h(m', \overline{m}'_h, r'_1, r'_2, r'_3))$, compute $\mathsf{Enc}_{\mathsf{HIBE}_h}(\overline{m}'_h || r'_h, V_h, p_h; R'_h) = \nu_h$ 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 [21]) 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 a 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 breaks the above scheme with probability $1/2 + \epsilon_A$, and A runs in 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*

[1] Namely, $d^1_t = (\sigma_2, \sigma_3)$ and $\delta^0_{T_0(\mathtt{time}')} = (\sigma'_2, \sigma'_3)$ for $i = 1$, and $d^2_t = \sigma_3$ and $\delta^1_{T_1(\mathtt{time}')} = \sigma'_3$ for $i = 2$.

another adversary B who can break at least one of HIBE_h $(1 \le h \le 3)$ *in the sense of* IND-HID-CPA *with probability* $1/2 + \epsilon_B$ *and running time* t_B *where*

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

assuming that $\gamma_{max} = \max(\gamma_1, \gamma_2, \gamma_3)$*,* HIBE_i *is* γ_i*-uniform, and running time of* $\mathsf{Gen}^i_{\mathsf{HIBE}_h}$ *is at most* τ_{GEN} *for any* h *and* i *such that* $1 \le h \le 3$ *and* $1 \le i \le h$.

Proof. See Appendix B. □

Random Oracle. If we want to eliminate random oracle, multiple encryption technique in [12] can be extended instead of the one used in [27] to construct a KE-CCA secure IKE, assuming that underlying HIBEs are all IND-HID-CCA in the standard model, e.g. [11, 4, 5], while the above construction using [27] requires only IND-HID-CPA HIBEs. Furthermore, by applying a similar method to the our proposed scheme, we can construct yet another KE-CCA secure IKE from HIBE that only need to have one-wayness under chosen plaintext attacks. All of these constructions will be shown in the full version of this paper.

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 [13] 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 [9] to construct a strongly secure key-insulated encryption. (The Boneh-Franklin IBE based scheme was proposed earlier in [8]).

In the following, we give a 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 [7] (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 in his private device. For the renewal of his private key at time period t, his private device computes $\mathsf{Gen}_{\mathsf{IBE}}(t, s, p) = s_t$ and sends the value obtained back to the user. He will then use this value (key-update information) to update/generate his decryption key, (dk, s_t) at time t. When encrypting a message m for time period t, then $\overline{m}_1$, $\overline{m}_2$, 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, any chosen ciphertext attacks can be prevented even for the following cases: (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 of this scheme will appear in the full version of this paper (proof technique is similar 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 will also be the first hierarchical construction of a strongly secure key-insulated encryption.

5 Efficient Construction from Bilinear Mapping

Construction. In the previous section, we showed a construction of a KE-CCA secure IKE using HIBE as a black-box. Here, we propose a construction of a KE-CCA secure IKE by directly extending Gentry-Silverberg HIBE [23] (see also Appendix C) and Fujisaki-Okamoto conversion [21]. This method is more efficient than the one we have shown in the previous section, however, it is based on a very specific assumption, i.e. BDH assumption, and may lack flexibility in designing new construction in terms of security.

KE-CCA secure IKE from Bilinear Mapping

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.

$\underline{\mathsf{PGen}_{\mathsf{IKE}}}$: On input 1^k, set up two cyclic groups G_1 and G_2 of prime order q and an efficiently computable mapping $\hat{e} : G_1 \times G_1 \to G_2$ such that $\hat{e}(aQ, bR) = \hat{e}(Q, R)^{ab}$ for all $Q, R \in G_1$ and any positive integers a, b. (This does not send all pairs in $G_1 \times G_1$ to the identity in G_2.) Choose an arbitrary generator $P \in G_1$, and pick $s^{h-1}_h \in_R Z_q$ for $1 \leq h \leq 3$. Then, calculate $Q := \sum_{1 \leq h \leq 3} s^{h-1}_h P$. Also, set cryptographic hash functions $H_1 : \{0, 1\}^* \to G_1$, $H_2 : G_2 \to \{0, 1\}^n$, $H_3 : \{0, 1\}^n \times \{0, 1\}^n \to Z_q$ and $H_4 : \{0, 1\}^n \to \{0, 1\}^n$, where n denotes the size of the message space. The security analysis will view $H_1, \cdots, H_4$ as random oracles. Then, output master key $s := (s^0_1, s^1_2, s^2_3)$ and public parameter $p := (G_1, G_2, \hat{e}, P, Q, H_1, H_2, H_3, H_4)$.

$\underline{\mathsf{Gen}_{\mathsf{IKE}}}$: On input s, p and U, compute $H_1(U) = P_U \in G_1$, and $S^{h-1}_h := s^{h-1}_h P_U$ for $1 \leq h \leq 3$. Then, output U's initial keys (d^0_0, d^1_0, d^2_0), where $d^0_0 := (S^0_1, (0, 0), (0, 0, 0))$, $d^1_0 := (S^1_2, (0, 0))$ and $d^2_0 := S^2_3$.

$\underline{\Delta\text{-}\mathsf{Gen}^{(i)}_{\mathsf{IKE}}}$: For $i = 2$, on input d^i_t, p and $\mathtt{time}'$, parse $d^2_0 = S^2_3$. Next, pick $s^1_3 \in_R Z_q$, and compute $\hat{S}^1_3 := S^2_3 + s^1_3 P_{t_1}$, $\hat{Q}^1_h := s^1_3 P$, where $P_{t_1} := H_1(U.T_1(\mathtt{time}'))$. Then, output $\delta^1_{T_1(\mathtt{time}')} := (\hat{S}^1_3, \hat{Q}^1_3)$. For $i = 1$, on input d^i_t, p and $\mathtt{time}'$, output $\perp$ if $t \neq T_1(\mathtt{time}')$. Else, parse $d^1_t = (S^1_2, (S^1_3, Q^1_3))$. Next, for $h = 2, 3$, pick $s^0_h \in_R Z_q$, and compute $\hat{S}^0_h := S^1_h + s^0_h P_{t_0}$, $\hat{Q}^0_h := s^0_h P$, where $P_{t_0} := H_1(U.T_1(\mathtt{time}').T_0(\mathtt{time}'))$. Also, set $\hat{Q}^1_3 = Q^1_3$. Then, output $\delta^0_{T_0(\mathtt{time}')} := ((\hat{S}^0_2, \hat{Q}^0_2), (\hat{S}^0_3, \hat{Q}^0_3, \hat{Q}^1_3))$.

$\underline{\mathsf{Upd}^{(i)}_{\mathsf{IKE}}}$: For $i = 2$, on input d^1_t, p and $\delta^1_{T_1(\mathtt{time}')}$, parse $d^1_t = (S^1_2, (S^1_3, Q^1_3))$ and $\delta^1_{T_1(\mathtt{time}')} = (\hat{S}^1_3, \hat{Q}^1_3)$, and output $d^1_{T_1(\mathtt{time}')} = (S^1_2, (\hat{S}^1_3, \hat{Q}^1_3))$. For $i = 1$, on input d^0_t, p and $\delta^0_{T_0(\mathtt{time}')}$, parse $d^0_t = (S^0_1, (S^0_2, Q^0_2), (S^0_3, Q^0_3, Q^1_3))$ and $\delta^0_{T_0(\mathtt{time}')} = ((\hat{S}^0_2, \hat{Q}^0_2), (\hat{S}^0_3, \hat{Q}^0_3, \hat{Q}^1_3))$, and output $d^0_t = (S^0_1, (\hat{S}^0_2, \hat{Q}^0_2), (\hat{S}^0_3, \hat{Q}^0_3, \hat{Q}^1_3))$.[2]

$\underline{\mathsf{Enc}_{\mathsf{IKE}}}$: On input m, U, p and $\mathtt{time}$, pick $\mu \in_R \{0, 1\}^n$ and set $r := H_3(\mu, m)$. Then, compute

$$c := \langle rP, rP_{t_1}, rP_{t_0}, \mu \oplus H_2(g^r), m \oplus H_4(\mu)) \rangle,$$

where $g := \hat{e}(Q, P_U) \in G_2$, $P_U := H_1(U)$, $P_{t_1} := H_1(U.T_1(\mathtt{time}))$ and $P_{t_0} := H_1(U.T_1(\mathtt{time}).T_0(\mathtt{time}))$.

$\underline{\mathsf{Dec}_{\mathsf{IKE}}}$: On input $\langle c', \mathtt{time} \rangle$ and $d^0_{U,t}$, output $\perp$ if $t \neq T_0(\mathtt{time})$. Else, parse $c' = \langle V, V_{t_1}, V_{t_0}, W, \Gamma \rangle$ and $d^0_{U,t} = (S^0_1, (S^0_2, Q^0_2), (S^0_3, Q^0_3, Q^1_3))$ and output $\perp$ if $(V, V_{t_1}, V_{t_0}, W, \Gamma) \notin {G_1}^5$. Else, calculate

$$W \oplus H_2\left(\frac{\hat{e}(S^0_1 + S^0_2 + S^0_3, V)}{\hat{e}(Q^0_2 + Q^0_3, V_{t_0}) \hat{e}(Q^1_3, V_{t_1})}\right) = \mu',$$

and $\Gamma \oplus H_4(\mu') = m'$. Next, re-encrypt m' (for U and time) by using μ' as the internal coin-flipping. If the result of re-encryption is identical to $\langle c', \mathsf{time} \rangle$, output m', otherwise output $\perp$.

Theorem 2 *The above scheme is a KE-CCA secure 2-level IKE in the random oracle model, assuming that BDH problem [6, 7] is hard to solve. More precisely, suppose there is an adversary A who breaks the above scheme with probability $1/2 + \epsilon_A$, and A runs in time at most t_A. Also, suppose A makes at most* q_{KG}, q_{KI}, q_{D}, $q_{H_1}, \cdots, q_{H_4}$ *queries to* KG, KI, D, $H_1, \cdots, H_4$, *respectively. Then, there is another adversary*

[2] Note that $(S^1_3, Q^1_3) = (0, 0)$, $(S^0_2, Q^0_2) = (0, 0)$ and $(S^0_3, Q^0_3, Q^1_3) = (0, 0, 0)$ for $t = 0$.

B who can break Gentry-Silverberg HIBE [23], which is proven to be secure under BDH assumption in the random oracle model in the sense of IND-HID-CPA *(see Theorem 4 in Appendix C) with probability* $1/2 + \epsilon_B$ *and running time* t_B *where*

$$\begin{aligned} \epsilon_B &\geq \frac{1}{3}\epsilon_A - \frac{1}{6}(\frac{q_{H_2} + q_{\mathsf{D}}}{q} + \frac{q_{H_3} + q_{H_4}}{2^n}), \\ t_B &\leq t_A + (2q_{H_1} + 2q_{\mathsf{KG}} + 5q_{\mathsf{KI}})\tau_{EXP} + q_{H_1}\tau_{\mathsf{poly}} + O((\log_2 q)q_{H_2} + n(2q_{H_3} + q_{H_4})), \end{aligned}$$

assuming that time for computing xP *for an integer* x *is at most* τ_{EXP}*, and* $\tau_{\mathsf{poly}} = O(\mathsf{poly}(k))$*.*

Proof. See Appendix D. □

Efficiency. In a pairing based scheme, the number of pairing computation done is the dominant factor that decides its total computation cost. For the above construction of a 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 a generic construction (shown in the previous section) that uses [23] as the underlying HIBE, the numbers of pairing computation for encryption and decryption are three and six, respectively. Therefore, in terms of computational cost, the above specific construction surpasses the generic construction based on [23]. This result can be generalized for ℓ-level IKE for any $\ell > 1$ as shown in Table 1.

Extending Message Space. In the above scheme, instead of using one-time pad $\Gamma = m \oplus H_4(\mu)$, we can also utilize semantically secure symmetric encryption by using $H_4(\mu)$ as the encryption key [21].

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 propose a generic construction of HIBE from arbitrary IBE that also provides a (partial) solution to an open problem of HIBE. With such a construction, for example, we can bring the Cocks IBE [9] to construct a HIBE as well. This also implies that, hereafter, a new construction of IBE is proposed, automatically, it is convertible to a HIBE. For the security definition of the construction of HIBE, we introduce partial collusion resistance (i.e. IND-wHID-CCA) [24] instead of full collusion resistance (i.e. IND-HID-CCA) [23]. The security definition is more relaxed, but still, there has still never been a 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 $t > 2$.

Security Definition. Our construction of HIBE which will be proposed here is based on the security definition of [24]. 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.

Table 1: Numbers of pairing computations in the pairing based scheme and the generic scheme based on [23].

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

Cover Free Family. The scheme shown here utilizes cover free family (CFF) [17] similarly seen in the generic construction of key-insulated encryption [13], although, reminding that, the method used in [13] 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 $|F_i| = O(w \log v)$ $(1 \leq i \leq v)$, where $|F_i|$ denotes the number of elements in F_i. 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 [18].

Construction. Now we show a generic construction of a chosen ciphertext secure 2-level HIBE with partial collusion resistance built from an arbitrary chosen plaintext secure IBE (i.e. $\mathsf{IND\text{-}ID\text{-}\underline{CPA}}$) using CFF.

Generic Construction of Partially Collusion Resistant HIBE

Let $\mathsf{IBE} = (\mathsf{PGen}_{\mathsf{IBE}}, \mathsf{Gen}_{\mathsf{IBE}}, \mathsf{Enc}_{\mathsf{IBE}}, \mathsf{Dec}_{\mathsf{IBE}})$ be a standard IBE (i.e. 1-level HIBE). Then, a 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.
$\underline{\mathsf{PGen}_{\mathsf{HIBE}}}$: On input 1^k, set up u copies of IBE. For $1 \leq i \leq u$, compute $(s_i, p_i) = \mathsf{PGen}_{\mathsf{IBE}}(1^k)$, and generate (u, v, w)-CFF (L, F), such that $L = \{1, \cdots, u\}$, $u = O(\mathsf{poly}(k))$, $v = O(\mathsf{exp}(k))$ and $w = O(\mathsf{poly}(k))$. Then, choose cryptographic hash functions $H_i : \{0,1\}^{2n+\hat{u}k_1} \rightarrow \mathcal{COIN}$ for $1 \leq i \leq u$, where 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. Also, choose a cryptographic hash function $H : \{0,1\}^* \rightarrow F$. Finally, output $s := (s_1, \cdots, s_u)$ and $p := (p_1, \cdots, p_u, H_1, \cdots, H_u, H)$. The security analysis will view H_i $(1 \leq i \leq u)$ and H as random oracles.
$\underline{\mathsf{Gen}^1_{\mathsf{HIBE}}}$: On input D^1, s and p, where D^1 is a 1-level sub-PKG, parse $s = (s_1, \cdots, s_u)$ and $p = (p_1, \cdots, p_u, H_1, \cdots, H_u, H)$ and compute $H(D^1) = F_{D^1} \in F$. Then, output D^1's key $s_{D^1} := \{s_i | i \in F_{D^1}\}$.
$\underline{\mathsf{Gen}^0_{\mathsf{HIBE}}}$: On input $D^1.D^0$, s_{D^1} and p, where $D^1.D^0$ is a user under sub-PKG D^1, for all $i \in F_{D^1}$, run $\mathsf{Gen}_{\mathsf{IBE}}(D^1.D^0, s_i, p_i) = s_{i,D^1.D^0}$ and output D^1's key $s_{D^1.D^0} := \{s_{i,D^1.D^0} | i \in F_{D^1}\}$.
$\underline{\mathsf{Enc}_{\mathsf{HIBE}}}$: On input m, $D^0.D^1$ and p, pick $\overline{m}_i \in_R \{0,1\}^n$ for all $i \in F_{D^1}$ such that $\oplus_{i \in F_{D^1}} \overline{m}_i = m$. Also, pick $r_i \in_R \{0,1\}^{k_1}$ for all $i \in F_{D^1}$. Let R be concatenation of all r_i arranged in increasing order of i for $i \in F_{D^1}$. Then, compute

$$c_i = \mathsf{Enc}_{\mathsf{IBE}}(\overline{m}_i || r_i, D^0.D^1, p_i; H_i(m, \overline{m}_i, R))$$

for all $i \in F_{D^1}$. Then, output $c := \{c_i | i \in F_{D^1}\}$.
$\underline{\mathsf{Dec}_{\mathsf{HIBE}}}$: On input $c' (= \{c'_i | i \in F_{D^1}\})$, $s_{D^1.D^0} (= \{s_{i,D^1.D^0} | i \in F_{D^1}\})$ and p, for all $i \in F_{D^1}$, compute

$$\mathsf{Dec}_{\mathsf{IBE}}(c'_i, s_{i,D^1.D^0}, p_i) = (\overline{m}'_i || r'_i).$$

Let R' be concatenation of all r'_i arranged in increasing order of i for $i \in F_{D^1}$, and m' be $\oplus_{i \in F_{D^1}} \overline{m}'_i$. Next, run $\mathsf{Enc}_{\mathsf{IBE}}(\overline{m}'_i || r'_i, D^0.D^1, p_i; H_i(m', \overline{m}'_i, R)) = \nu_i$ for all $i \in F_{D^1}$. Unless $\nu_i = c'_i$ for all $i \in F_{D^1}$, output $\perp$, otherwise, output m'.

Theorem 3 *The above scheme is* $\mathsf{IND\text{-}}w\mathsf{HID\text{-}CCA}$ *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* $\mathsf{IND\text{-}ID\text{-}CPA}$. *More* *precisely, suppose there is an adversary A who breaks the above scheme with probability $1/2 + \epsilon_A$, and A runs in time at most t_A. Also, suppose A makes at most q_{KG}, q_{D}, q_{H_i} queries to KG, D, H_i $(1 \leq i \leq u)$, respectively. Then, by letting $q_{max} := \max_{D^1} \sum_{i \in H(D^1)} q_{H_i}$, 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*

$$\begin{aligned} \epsilon_B &\geq \frac{\hat{u}}{u^2}\epsilon_A - \frac{\hat{u}}{2u^2}(\frac{q_{max}}{2^{k_1}} + q_{\mathsf{D}}\gamma), \\ t_B &\leq t_A + q_{\mathsf{KG}}\hat{u}\tau_{GEN} + \hat{u}\tau_{ENC} + O((2n + \hat{u}k_1)(\sum_{1 \leq i \leq u} q_{H_i})), \end{aligned}$$

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. Here, 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 private device while s_0 is given to U as his initial decryption key. To encrypt m for U and `time`, $\overline{m}_i$ are picked from $\{0,1\}^n$ for all $i \in F'_{T_0(\mathtt{time})} := H(U.T_0(\mathtt{time})) \cup \{0\}$, such that $\oplus_{i \in F'_{T_0(\mathtt{time})}} \overline{m}_i = m$. Also, r_i are picked from $\{0,1\}^{k_1}$ for all $i \in F'_{T_0(\mathtt{time})}$. 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 $i \in F'_{T_0(\mathtt{time})}$, where R denotes concatenation of all r_i for $i \in F'_{T_0(\mathtt{time})}$ in increasing order of i. Finally, output $c := \{\overline{c}_i | i \in F'_{T_0(\mathtt{time})}\}$. It is obvious that the decryption key $\{s_{i,U} | i \in F'_{T_0(\mathtt{time})}\}$ for `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 IBEs may be suited. Take notice that even if the underlying IBEs are IND-ID-CCA, still, straightforward multiple encryption is not 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.

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.???-???, 2004.

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

[7] 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 [6]).

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

[9] 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.

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

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

[12] Y. Dodis and J. Katz, rump session talk, Crypto’03, 2003.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

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

[20] 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.

[21] 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.

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

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

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

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

[26] 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.

[27] 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.

[28] 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 [23] and [24] which is the hierarchical extension of Boneh and Flanklin's IBE [6, 7].

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 $0 \le i \le t-1$. 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 $\mathsf{HIBE} = (\mathsf{PGen}_{\mathsf{HIBE}}, \mathsf{Gen}^{i}_{\mathsf{HIBE}}\ (1 \le i \le t), \mathsf{Enc}_{\mathsf{HIBE}}, \mathsf{Dec}_{\mathsf{HIBE}})$ 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 $\{(D^{*,t-1}.D^{*,t-2}.\cdots.D^{*,i} : 0 \le i \le t-1)\}$ 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 $\mathsf{IND}\text{-}w\mathsf{HID}\text{-}\mathsf{CCA}$ and $\mathsf{IND}\text{-}w\mathsf{HID}\text{-}\mathsf{CPA}$. 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 $\mathsf{IND}\text{-}w\mathsf{HID}\text{-}\mathsf{CCA}$ (or $\mathsf{IND}\text{-}w\mathsf{HID}\text{-}\mathsf{CPA}$) 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 $\mathsf{IND}\text{-}w\mathsf{HID}\text{-}\mathsf{CCA}$ 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 $\mathsf{IND}\text{-}\mathsf{HID}\text{-}\mathsf{CPA}$ 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 $\overline{m}_h \in_R \{0,1\}^n$ such that $\oplus_{1 \leq h \leq 3,\ h \neq a} \overline{m}_h = 0$. Also, B sets $\overline{m}_{a,0} = m_0$ and $\overline{m}_{a,1} = m_1$. Then, B picks $r_{h,j} \in_R \{0,1\}^{k_1}$ for $1 \leq h \leq 3,\ j = 0, 1$, and sets $V_1 = U^*$, $V_2 = U^*.T_0(\mathtt{time}^*)$ and $V_3 = U^*.T_1(\mathtt{time}^*).T_0(\mathtt{time}^*)$. Also, B sends V_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 flips a coin $b \in_R \{0,1\}$ and encrypts $(\overline{m}_{h,b}||r_{h,b})$ by the encryption algorithm of HIBE_h with p'_h and V_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_i. For H_i $(1 \leq i \leq 3)$ oracle queries, B returns random values if the query has not been asked before, otherwise B returns the same value as before.

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^i_0 for $0 \leq i \leq 2$ by using these results and return (d^0_0, d^1_0, d^2_0).

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. More precisely, on A's request for a KI oracle query i, U and $\mathtt{time}$, B calculates

- keys for U, $U.T_0(\mathtt{time})$ and $U.T_1(\mathtt{time}).T_0(\mathtt{time})$ which correspond to HIBE_h for $1 \leq h \leq 3$, respectively, by either asking B's KG oracle for HIBE_a or by running the key generation algorithms of HIBE_h with master keys $s'_1, \cdots, s'_3$ for $1 \leq h \leq 3,\ h \neq a$ if $i = 0$,
- keys for U and $U.T_1(\mathtt{time})$ which correspond to HIBE_h for $2 \leq h \leq 3$, respectively, by either asking B's KG oracle for HIBE_a (only when $a = 2$ or 3) or by running the key generation algorithms of HIBE_h with master keys s'_h for $h = 2, 3,\ h \neq a$ if $i = 1$,
- U's key which correspond to HIBE_3, by either asking B's KG oracle which corresponds to HIBE_a (only when $a = 3$) or by running the key generation algorithms of HIBE_3 with master key s'_3 if $i = 2$.

By using these keys, B produces $d^i_{T_i(\texttt{time})}$ and returns it to A. It should be noticed that the above 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 $1 \le i \le 3$, 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 $\oplus_{1\le i\le 3}\overline{m}_i = m$ (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 $\texttt{time}$) 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 $\mathsf{IND\text{-}HID\text{-}CPA}$ game for HIBE_a.

Now, we estimate B's succeeding probability. Simulations of H_i $(1 \le i \le 3)$ and KG are both perfect. Simulation of LR fails only when B asks an H_i query which corresponds to the challenge ciphertext. Therefore, the succeeding probability of the simulation becomes at least $(1 - 1/2^{k_1})^{q_{H_1}+q_{H_2}+q_{H_3}}$, where q_{H_i} $(1 \le i \le 3)$ are the numbers of queries made to H_i. Simulation of KI fails only when i' is not the special level chosen by A. 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, the succeeding probability of the simulation becomes at least $(1-\gamma_{max})^{q_{\mathsf{D}}}$, 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. 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

$$\begin{aligned}\epsilon_B &\ge \frac{1}{3}\Big(\frac{1}{2}+\epsilon_A\Big)\Big(1-\frac{1}{2^{k_1}}\Big)^{q_{H_1}+q_{H_2}+q_{H_3}}(1-\gamma_{max})^{q_{\mathsf{D}}} + \frac{2}{3}\cdot\frac{1}{2} - \frac{1}{2}\\ &\simeq \frac{1}{3}\epsilon_{A_2} - \frac{1}{6}\Big(\frac{q_{H_1}+q_{H_2}+q_{H_3}}{2^{k_1}} + q_{\mathsf{D}}\gamma_{max}\Big).\end{aligned}$$

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

$$t_B \le t_A + (2q_{\mathsf{KG}} + 5q_{\mathsf{KI}})\tau_{GEN} + O((2n+3k_1)(q_{H_1}+q_{H_2}+q_{H_3})),$$

assuming that the number of queries made to KG and KI is q_{KI} and q_{KI}, respectively, and running time of $\mathsf{Gen}^i_{\mathsf{HIBE}_h}$ is at most τ_{GEN} for any h and i such that $1 \le h \le 3$ and $1 \le i \le h$. 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 [23]

Here, we give a brief review of Gentry-Silverberg HIBE [23]. 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 $\hat{e} : G_1 \times G_1 \to G_2$ such that $\hat{e}(aQ, bR) = \hat{e}(Q, R)^{ab}$ for all $Q, R \in G_1$ 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 $Q^{\mathsf{HIBE}} := sP$ and sets cryptographic hash functions $H_1^{\mathsf{HIBE}} : \{0,1\}^* \to G_1$, $H_2^{\mathsf{HIBE}} : G_2 \to \{0,1\}^n$, $H_3^{\mathsf{HIBE}} : \{0,1\}^n \times \{0,1\}^n \to Z_q$ and $H_4^{\mathsf{HIBE}} : \{0,1\}^n \to \{0,1\}^n$, where n denotes the size of the message space. Next, the root-PKG keeps master key s and sets the public parameter $p^{\mathsf{HIBE}} := (G_1, G_2, \hat{e}, P, Q^{\mathsf{HIBE}}, H_1^{\mathsf{HIBE}}, H_2^{\mathsf{HIBE}}, H_3^{\mathsf{HIBE}}, H_4^{\mathsf{HIBE}})$. For a sub-PKG D^1, the PKG computes $H_1^{\mathsf{HIBE}}(D^1) = P_{D^1} \in G_1$ 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 $P_{D^1.D^0} := H_1(D^1.D^0)$. When encrypting $m \in \{0,1\}^n$ for $D^1.D^0$, a sender picks $\mu \in_R \{0,1\}^n$, sets $r := H_3^{\mathsf{HIBE}}(\mu, m)$ and computes $c := \langle rP, rP_{D^1.D^0}, \mu \oplus H_2^{\mathsf{HIBE}}(g^r), m \oplus H_4^{\mathsf{HIBE}}(\mu)\rangle$, where $g := \hat{e}(Q, P_{D^1}) \in G_2$. On receiving $c' = \langle V, V', W, \Gamma\rangle$, $D^1.D^0$ calculates $W \oplus H_2^{\mathsf{HIBE}}(\hat{e}(S_{D^1.D^0}, V)\hat{e}(Q', V')^{-1}) = \mu'$, and $\Gamma \oplus H_4^{\mathsf{HIBE}}(\mu') = m'$.

Next, $D^1.D^0$ re-encrypts m' (for $D^1.D^0$) by using μ' as the internal coin-flipping. If the result of re-encryption is identical to $\langle c', \mathsf{time}\rangle$, $D^1.D^0$ outputs m', otherwise outputs $\perp$.

Theorem 4 ([7],[23]) *Gentry-Silverberg HIBE is* IND-HID-CPA *in the random oracle model assuming that BDH problem [6, 7] is hard to solve. Concretely, suppose there is an* IND-HID-CCA *adversary A who can break the above scheme in the sense of* IND-HID-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 $q_{\mathsf{H_i}}$ queries to H_i for $2 \le i \le 4$, then there exists an algorithm B that solves the BDH problem underlying the HIBE with probability of at least $1/2+\epsilon_B$ and running time t_B, where*

$$\begin{aligned}\epsilon_B &\ge 2\epsilon_A\left(\frac{t}{e(t+q_{\mathsf{KG}})q_{H_2}}\right)^t - \frac{q_{H_3}+q_{H_4}}{2^n},\\ t_B &= O(t_A),\end{aligned}$$

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 Gentry-Silverberg HIBE, see [23]. 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. We construct an adversary B who can break Gentry-Silverberg HIBE [23] in the sense of IND-HID-$\underline{\text{CPA}}$ by using another adversary A which is able to break KE-CCA security of the proposed scheme. Note that the security of [23] is proven under BDH assumption [6], and hence, existence of B implies that our scheme is also secure under the same assumption (in the random oracle model).

For given $p^{\mathsf{HIBE}} := (G_1, G_2, \hat{e}, P, Q^{\mathsf{HIBE}}, H_1^{\mathsf{HIBE}}, H_2^{\mathsf{HIBE}}, H_3^{\mathsf{HIBE}}, H_4^{\mathsf{HIBE}})$ as public parameter of Gentry-Silverberg HIBE (see Appendix C), B chooses $a \in \{1,2,3\}$ and $s'_h \in_R Z_q$ $(1 \le h \le 3,\ h \ne a)$. Then, B sets $Q := Q^{\mathsf{HIBE}} + \sum_{1\le h\le 3,\ h\ne a} s'_h P$ and gives $p := (G_1, G_2, \hat{e}, P, Q, H_1, H_2, H_3, H_4)$ to A as an IKE public parameter, where H_i $(1 \le i \le 4)$ are random oracles. 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 $V_1 = U^*$, $V_2 = U^*.T_0(\mathtt{time}^*)$ and $V_3 = U^*.T_1(\mathtt{time}^*).T_0(\mathtt{time}^*)$, and sends V_a, m_0, m_1 to B's own LR oracle. (Here, the target HIBE works as an a-level HIBE.[3]) B's LR oracle flips a coin $b \in_R \{0,1\}$, picks $\mu \in_R \{0,1\}^n$ and returns $c^*_{\mathsf{HIBE}} := \langle \mathcal{P}, W, \Gamma\rangle$ where $r = H_3^{\mathsf{HIBE}}(\mu, m)$, $W = \mu \oplus H_2^{\mathsf{HIBE}}(g^r)$, $\Gamma = m \oplus H_4^{\mathsf{HIBE}}(\mu)$, and $\mathcal{P} = rP$ if $a = 1$, $\mathcal{P} = (rP, rH_1^{\mathsf{HIBE}}(U^*.T_1(\mathtt{time}^*)))$ if $a = 2$ and $\mathcal{P} = (rP, rH_1^{\mathsf{HIBE}}(U^*.T_1(\mathtt{time}^*)), rH_1^{\mathsf{HIBE}}(U^*.T_1(\mathtt{time}^*).T_0(\mathtt{time}^*)))$ if $a = 3$. Then,

- if $a = 1$, B picks $r_1, r_2 \in_R Z_q$ and stores $H_1(U^*.T_1(\mathtt{time}^*)) = r_1P$ and $H_1(U^*.T_1(\mathtt{time}^*).T_0(\mathtt{time}^*)) = r_2P$ in his own H_1 oracle list,
- if $a = 2$, B stores $H_1(U^*.T_0(\mathtt{time}^*)) = H_1^{\mathsf{HIBE}}(U^*.T_1(\mathtt{time}^*))$. Also, B picks $r_2 \in_R Z_q$ and stores $H_1(U^*.T_1(\mathtt{time}^*).T_0(\mathtt{time}^*)) = r_2P$ in his own H_1 oracle list,
- if $a = 3$, B stores $H_1(U^*.T_0(\mathtt{time}^*)) = H_1^{\mathsf{HIBE}}(U^*.T_1(\mathtt{time}^*))$ and also $H_1(U^*.T_1(\mathtt{time}^*).T_0(\mathtt{time}^*)) = H_1^{\mathsf{HIBE}}(U^*.T_1(\mathtt{time}^*).T_0(\mathtt{time}^*))$ in his own H_1 oracle list.

Finally, B sends a challenge ciphertext $c^* := \langle \mathcal{P}', W, \Gamma\rangle$ to A, where

[3] The depth of hierarchy in Gentry-Silverberg HIBE can be flexibly determined after setting up the system.

- if $a = 1$, $\mathcal{P}' = (rP, r_1 \cdot rP, r_2 \cdot rP)$,
- if $a = 2$, $\mathcal{P}' = (rP, rH_1^{\mathsf{HIBE}}(U^*.T_1(\mathtt{time}^*)), r_2 \cdot rP)$,
- if $a = 3$, $\mathcal{P}' = (rP, rH_1^{\mathsf{HIBE}}(U^*.T_1(\mathtt{time}^*)), rH_1^{\mathsf{HIBE}}(U^*.T_1(\mathtt{time}^*).T_0(\mathtt{time}^*)))$.

Simulation of H_i. For H_1' oracle queries, B submits the same queries to his H_1^{HIBE} oracle except for the next

- if $a = 1$, and the query forms $x.T_0(y)$ or $x.T_1(y).T_0(z)$ for any x, y, z, then B picks $r_1, r_2 \in_R Z_q$ and stores $H_1(x.T_1(y)) = r_1P$ or $H_1(x.T_1(y).T_0(z)) = r_2P$ in his own H_1 oracle list, respectively,
- if $a = 2$, and the query forms $x.T_b(y)$ for any x, y and $b \in \{0, 1\}$, then B submits $x.T_{b\oplus 1}(y)$ to H_1^{HIBE} oracle, and if $a = 2$, and the query forms $x.T_1(y).T_0(z)$ for any x, y, z, then B picks $r_2 \in_R Z_q$s and stores $H_1(x.T_1(y).T_0(z)) = r_2P$ in his own H_1' oracle list,
- if $a = 3$, and the query forms $x.T_b(y)$ for any x, y and $b \in \{0, 1\}$, then B submits $x.T_{b\oplus 1}(y)$ to H_1^{HIBE} oracle.

For H_i $(2 \le i \le 4)$ oracle queries, B returns random values if the query has not been asked before, otherwise, B returns the same value as before.

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 for the target HIBE to B's own KG oracle, and sets this key as d_0^{a-1}. Also, B computes $d_0^{h-1} = s_h' H_1(U)$ for $1 \le h \le 3,\ h \neq a$. Then, B returns (d_0^0, d_0^1, d_0^2). Similarly, KI can also be prefectly simulated when $a - 1$ is the "special level" chosen by A (see Def. 2).

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 for H_1, H_2, H_3, H_4 such that each of the combinations consists of the next six queries $\psi_{1,1}, \psi_{1,2}, \psi_{1,3}, \psi_2, \psi_3, \psi_4$, 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 $H_1(x)$, $H_1(x.T_0(y))$ and $H_1(x.T_1(y).T_0(z))$, respectively, for some x, y, z. Also, queries ψ_2, ψ_3, ψ_4 have been asked to H_2, H_3, H_4, respectively, before and ψ_2, ψ_3, ψ_4 form $\hat{e}(Q, \psi_{1,1})^{\psi_3}, (\mu, m), \mu$, respectively, for some μ and m. If there exists such a combination whose corresponding ciphertext (for U and $\mathtt{time}$) is identical to $\langle c, \mathtt{time}\rangle$, B returns m. Otherwise, B returns $\perp$.

When A outputs b', B searches for an H_2 query κ such that

$$c^*_{\mathsf{HIBE}} = \langle \mathcal{P}', \mu' \oplus H_2(\kappa \hat{e}(\sum_{1\le h\le 3,\ h\neq a} s_h' H_1(U), r'P)^{-1}), m' \oplus H_4(\mu')\rangle,$$

where

$$\mu' = W \oplus H_2(\kappa \hat{e}(\sum_{1\le h\le 3,\ h\neq a} s_h' H_1(U), rP)^{-1}),\ m' = \Gamma \oplus H_4(\mu'),\ r' = H_3(\mu', m'),$$

and $\mathcal{P}' = r'P$ if $a = 1$, $\mathcal{P}' = (r'P, r'h_1)$ if $a = 2$ and $\mathcal{P}' = (r'P, r'h_1, r'h_2)$ if $a = 3$ where $h_1 := H_1^{\mathsf{HIBE}}(U^*.T_1(\mathtt{time}^*))$ and $h_2 := H_1^{\mathsf{HIBE}}(U^*.T_1(\mathtt{time}^*).T_0(\mathtt{time}^*))$. If there exists such κ and also if m' is identical to $m_{b'}$ for $b' \in \{0, 1\}$, B then outputs b'.

Now, we estimate B's succeeding probability. Simulations of H_i $(1 \le i \le 3)$ and KG are perfect. Simulation of LR fails only when B asks H_i, a query that corresponds to the challenge ciphertext, then, the succeeding probability of the simulation becomes at least $(1-1/q)^{q_{H_2}}(1-1/2^n)^{q_{H_3}}(1-1/2^n)^{q_{H_4}}$, where q_{H_i} $(2 \leq i \leq 4)$ are the numbers of queries made to H_i. Simulation of KI fails only if $a-1$ is not the special level chosen by A. 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, the succeeding probability of the simulation becomes at least $(1-1/q)^{q_{\mathsf{D}}}$ where q_{D} is the number of queries made to D. If we let $1/2+\epsilon_A$ be the succeeding probability of A, then B's secceeding probability is now estimated to be $1/2+\epsilon_B$ where

$$\begin{aligned}\epsilon_B &\geq \frac{1}{3}(\frac{1}{2}+\epsilon_A)(1-1/q)^{q_{H_2}}(1-1/2^n)^{q_{H_3}}(1-1/2^n)^{q_{H_4}}(1-1/q)^{q_{\mathsf{D}}}+\frac{2}{3}\cdot\frac{1}{2}-\frac{1}{2}\\ &\simeq \frac{1}{3}\epsilon_A-\frac{1}{6}(\frac{q_{H_2}+q_{\mathsf{D}}}{q}+\frac{q_{H_3}+q_{H_4}}{2^n}).\end{aligned}$$

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

$$t_B \leq t_A+(2q_{H_1}+2q_{\mathsf{KG}}+5q_{\mathsf{KI}})\tau_{EXP}+q_{H_1}\tau_{\mathsf{poly}}+O((\log_2 q)q_{H_2}+n(2q_{H_3}+q_{H_4})),$$

assuming the number of queries made to KG and KI are q_{KG} and q_{KI}, respectively, and time for computing xP for an integer x is at most τ_{EXP}, and $\tau_{\mathsf{poly}}=O(\mathsf{poly}(k))$. Therefore, ϵ_A is negligible if ϵ_B, $1/q$ and $1/2^n$ 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.

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 $\oplus_{i\in H(D^1)}\overline{m}_i=0$. Then, B picks $r_u \in_R \{0,1\}^{k_1}$ and submits $D^1.D^0$, $(m_0||r_u)$ and $(m_1||r_u)$ to B's own LR oracle to obtain c_u. Finally, B sends c_i for all $i \in H(D^1)$ to A.

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.

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 $s_{D^1.D^0}=\{s_i|i \in F_{D^1}\}$ 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 (including B's target IBE), 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 k_1-bit strings r_h for all $h \in H(D^1)$, and also, $\oplus_{h\in H(D^1)}\overline{m}_h=m$, where R is a concatenation of all r_h arranged in increasing order of h for $h \in H(D^1)$ (note that m and r_h for all $h \in H(D^1)$ are common to all of these queries). 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 H_i $(1 \le i \le u)$ and H are perfect. Simulation of LR fails when B asks $(m_b, \overline{m}_i, R)$ to H_i for some $i \in H(D^1)$, where R is the concatenation of all r_i arranged in increasing order of i for $i \in H(D^1)$. Consequently, the succeeding probability of the simulation becomes at least $(1-1/2^{k_1})^{q_{max}}$, where $q_{max} := \max_{D^1} \sum_{i \in H(D^1)} q_{H_i}$, and q_{H_i} $(i \in H(D^1)$ are the numbers of queries made to H_i. Simulation of KG fails only if A asks D^1 such that $u \in F_{D^1}$. From the nature of (u, v, w)-CFF, there exists at least one underlying IBE whose master key has not been exposed to A. This also means that the succeeding probability of simulation of KG is at least $1/u$. 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, the succeeding probability of the simulation becomes at least $(1-\gamma)^{q_\mathsf{D}}$ where q_D is the number of queries for D, assuming that IBE is γ-uniform. If we let $1/2 + \epsilon_A$ be the succeeding probability of A, then B's succeeding probability is estimated to be $1/2 + \epsilon_B$ where

$$\begin{aligned}
\epsilon_B &\ge \frac{\#\{F_i | u \in F_i \in F\}}{\#F}(\epsilon_A + \frac{1}{2})\frac{1}{u}(1 - 1/2^{k_1})^{q_{max}}(1-\gamma)^{q_\mathsf{D}} \\
&\quad + (1 - \frac{\#\{F_i | u \in F_i \in F\}}{\#F}\frac{1}{u})\frac{1}{2} - \frac{1}{2} \\
&\simeq \frac{\hat{u}}{u}(\epsilon_A + \frac{1}{2})\frac{1}{u}(1 - \frac{q_{max}}{2^{k_1}})(1 - q_\mathsf{D}\gamma) + (1 - \frac{\hat{u}}{u^2})\frac{1}{2} - \frac{1}{2} \\
&\simeq \frac{\hat{u}}{u^2}\epsilon_A - \frac{\hat{u}}{2u^2}(\frac{q_{max}}{2^{k_1}} + q_\mathsf{D}\gamma)
\end{aligned}$$

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

$$t_B \le t_A + q_{\mathsf{KG}}\hat{u}\tau_{GEN} + \hat{u}\tau_{ENC} + O((2n + \hat{u}k_1)(\sum_{1 \le i \le u} q_{H_i})),$$

assuming that the number of queries made to KG is q_{KG}, and running time of $\mathsf{Gen_{IBE}}$ and $\mathsf{Enc_{IBE}}$ 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 IND-$w\mathsf{HID\text{-}CCA}$ with a restriction that an adversary is not allowed to ask KG for more than w times. □