Efficient and Provable Secure Anonymous Hierarchical Identity-based Broadcast Encryption (HIBBE) Scheme without Random Oracle

Mohammmad Hassan Ameri, Javad Mohajeri, Mahmoud Salmasizadeh

Abstract—Hierarchical identity-based broadcast encryption (HIBBE) organizes the users in a tree-like structure in which they can delegate the decryption ability to their subordinates. In addition, the trusted third party (TTP) can reduce its burden because the users' secret keys can be generated in a distributed mechanism by users' supervisors. HIBBE enables encrypting a message for any arbitrary set of receivers, and only the chosen users and their supervisors are able to decrypt. To preserving the anonymity of the intended receivers, in this paper, for the first time, we propose an anonymous HIBBE scheme. The proposed scheme is constructed based on composite order bilinear maps. We formally define the anonymity against chosen identity vector set and chosen plaintext attack (Anon-CIVS-CPA), and prove that the proposed scheme provides this property. Performance evaluation shows the practical and deployable aspects of our proposed scheme. With the advantage of HIBBE, we enable hierarchical identity-based signature (HIBS) schemes to sign a message for a set of designated verifiers. This resulted in proposing a generic construction for the novel notion of hierarchical identity-based multi-designated verifiable signature (HIB-MDVS). We formally define HIB-MDVS's security against existential forgery under chosen message attack (EF-CMA), prove that the resulting HIB-MDVS is unforgeable, and finally show that it provides the anonymity of the intended verifiers.

Index Terms—Broadcast encryption, Hierarchical identity-based encryption, Identity-based multi designated verifier signature, Provable security, Pairing-based cryptography

1 Introduction

Nowadays, we are cognizant of the inevitable role of network-centric world, its applications and services which has given rise to several security concerns related to managing access control and policies. Access control management ensures that only the intended or authorized users are allowed to have access to certain resources and services. Broadcast encryption (BE) is such scheme which efficiently establishes an access control policy on the encrypted data. BE is known as a strong tool for encrypting a message for an arbitrary set of users, and only the intended users can decrypt the broadcasted ciphertext. The concept of BE was introduced by Fiat and Naor, in 1993 [1], and it was pursued by many other researchers to enhance its security and improve its efficiency. For example, Sakai et al. [2] realize BE in the identity-based setting and proposed a scheme which is secure in the random oracle model. As the applications of BE, we can imply to its functionality in access control in encrypted file systems, satellite TV subscription services, and DVD content protection [3] and [4].

Identity-based BE (IBBE) is a public key cryptographic primitive in which the users are associated to a unique identity and the sender uses the identities of the intended receivers as their public key to encrypt the message. Then, each designated receiver uses its secret key to decrypt the broadcasted ciphertext. A trusted third party (TTP) is employed in the system as a private key generator (PKG) to generate and distribute the secret keys of the users associated with their own identities by means of a master secret key.

Typically, the users are organized in the networks with hierarchical structures and a hierarchy is specified by the fact that the users of higher level of hierarchy may have more access rights than their subordinates. With the benefit of hierarchical structure of networks, PKG can reduce its burden of generating the users' secret keys in a distributed mechanism and by delegating the key generation rights to the users whit higher level of hierarchy. Recently some schemes have been proposed to address the key management problem in hierarchical structures ([5], [6], [7], [8], [9] and [10]).

Hierarchical identity-based BE (HIBBE) generalizes IBBE by distributing the secret key generation process between the users which are organized in a network with hierarchical structure. The users receives their secret keys from their ancestors who are supported by a secret key delegation mechanism. In 2014, Liu et al. [11] defined the security model of HIBBE in a formal way and they proposed a concrete construction for HIBBE with constant size ciphertext. After that, in 2015, Liu et al. [12] proposed a practical HIBBE scheme which is semantically secure against chosen ciphertext attack (CCA).

Anonymity (or privacy-preserving) in the BE scheme is a matter of concern and urgently desired to be considered in the cryptographic protocols. For example, if the using BE scheme in satellite TV subscription services does not provide the anonymity of the receivers, then both the authorized and unauthorized users can know who has paid subscription to a certain channel which means that the privacy of the receives' has been violated. Anonymity in HIBBE means that the ciphertext leaks no information of the intended receivers' identities. The first fully anonymous identity-based BE (IBBE) proposed by Ren et al. [13], and they prove that their scheme is secure in the standard model without random oracle. Fully anonymity means

- *M. H. Ameri, J. Mohajeri and M. Salmasizadeh are with the Electronics Research Institute of Sharif University of Technology, Tehran, Iran.*
E-mail: ameri_mohammadhasan@ee.sharif.edu, {mohajer, salmasi}@sharif.edu

that even the insider users can not obtain the identity of other receivers [13]. To the best of our knowledge, there have not been proposed any anonymous HIBBE scheme. Motivated by this scenario, it is desirable to propose an anonymous HIBBE scheme to preserve the users' privacy. In this paper, we for the first time propose an anonymous HIBBE scheme which is designed based on composite order bilinear maps. We define the anonymity of HIBBE against chosen identity vector sets and chosen plaintext attack (Anon-CIVS-CPA) in a formal way and show that the proposed scheme is secure against Anon-CIVS-CPA.

Recently, it has been shown a great interest in delegating the verification rights of the generated signatures to the set of designated verifiers and just the intended ones can verify the validity of the signature [14]. This primitives are known as the multi-designated verifiers signatures [15]. In this paper, we introduce the novel notion of hierarchical identity-based multi-designated verifier signature (HIB-MDVS) and define its unforgeability in a formal way. As one of the application of our proposed anonymous HIBBE, we present a generic construction of HIB-MDVS using HIBBE as the one of the employed building blocks. With the advantage of anonymous HIBBE, the resulting HIB-MDVS preserve the anonymity of the designated verifiers.

1.1 Our contributions

The contribution of the paper is five-fold and is summarized as follows:

(1)We introduce the novel notion of anonymous HIBBE and formally define anonymity of the HIBBE against chosen identity vector sets and chosen plaintext attack (Anon-CIVS-CPA) and proposed an anonymous construction based on composite order bilinear map.

(2) We formally prove that the security of our proposed scheme of anonymous HIBBE is reduced to the 4 assumptions which are introduced in [16]. For this aim, we first define a sequence of games and by means of the hybrid argument we prove that our scheme provides Anon-CIVS-CPA.

(3) We present the performance analysis of the proposed scheme of anonymous HIBBE. The numerical results show that the resulting construction is practical and consumes reasonable time for the required computations.

(4) In this paper, we have found that by using HIBBE scheme we can enable hierarchical identity-based signature (HIBS) schemes [17] to sign a message for a set of designated verifiers. This culminated by introducing the novel notion of HIB-MDVS. We also formally define HIB-MDVS's security against existential forgery under chosen message attack (EF-CMA).

(5)Finally, we propose a generic construction of HIB-MDVS in a modular structure based on HIBBE and hierarchical identity-based signature (HIBS) as the building blocks. We will prove that the unforgeability of the proposed scheme is reduced to the unforgeability of the using HIBS. We will also show that by employing the anonymous HIBBE scheme the designated verifiers' anonymity will be provided. The performance evaluation shows the practical aspects of the resulting HIB-MDVS.

1.2 Organization

The rest of this paper is organized as follows. Section II reviews the using notations and cryptographic primitives and complexity assumptions. Section III is dedicated to formally define the anonymity of HIBBE and present a concrete construction for anonymous HIBBE. Section IV evaluate the performance of the proposed scheme and presents the experimental results. Section V analyses the security of the proposed HIBBE. Section VI introduces the concept of HIB-MDVS and defines its security in a formal way. Section VII propose the generic construction of HIB-MDVS and Section VIII proves its security. The performance evaluation of the resulting concrete construction of HIB-MDVS extracted from the generic structure is presented in Section IX. Finally, we conclude the paper in Section X.

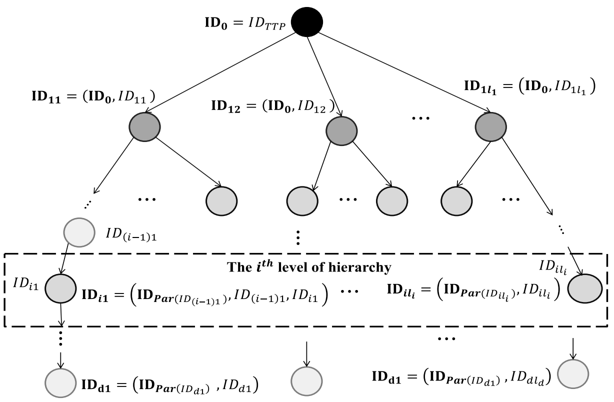
$\mathbf{ID_0} = ID_{TTP}$
$\mathbf{ID_{11}} = (\mathbf{ID_0}, ID_{11})$
$\mathbf{ID_{12}} = (\mathbf{ID_0}, ID_{12})$
$\mathbf{ID_{1l_1}} = (\mathbf{ID_0}, ID_{1l_1})$
$ID_{(i-1)1}$
The i^{th} level of hierarchy
ID_{i1}
$\mathbf{ID_{i1}} = \left(\mathbf{ID}_{Par(ID_{(i-1)1})}, ID_{(i-1)1}, ID_{i1}\right)$
ID_{il_i}
$\mathbf{ID_{il_i}} = \left(\mathbf{ID}_{Par(ID_{il_i})}, ID_{il_i}\right)$
$\mathbf{ID_{d1}} = \left(\mathbf{ID}_{Par(ID_{d1})}, ID_{d1}\right)$
$\mathbf{ID_{d1}} = \left(\mathbf{ID}_{Par(ID_{d1})}, ID_{dl_d}\right)$

Fig. 1. Hierarchical model of the network. ID_{TTP} expresses the identity of TTP and ID_{ij} denotes the identity of jth user in the ith level of hierarchy. The ith level contains l_i users, therefore we have $1 \leq j \leq l_i$. $\mathbf{ID}_{ij} = (\mathbf{ID}_{Par(ID_{ij})}, ID_{ij})$ denotes the identity vector of the user whose identity is ID_{ij}.

2 Preliminaries

2.1 Notations

In a network with a hierarchical structure like a tree, each user is associated with an identity and an identity vector. The existing identities are indexed with a number according to the tree-like hierarchical structure of the network. The identity vector of each user contains two parts: the first part is identity vector of its parents and the second part is the user's identity. Consider a network with d level of hierarchies where in the d'th level, $l_{d'}$ users exist. The identity of the $j_{d'}$th user in the level d', is denoted as $ID_{d'j_{d'}}$ and its corresponding identity vector is $\mathbf{ID}_{d'j_{d'}} = (\mathbf{ID}_{Par(ID_{d'j_{d'}})}, ID_{d'j_{d'}}) = (ID_{1j_1}, \ldots, ID_{(d'-1)j_{d'-1}}, ID_{d'j_{d'}})$. The notation $Par(\mathbf{ID}_{d'j_{d'}}) = \{(ID_{1j_1}, \ldots, ID_{d''j_{d''}}) : d'' \leq d'\}$ denotes the parent of $\mathbf{ID}_{d'j_{d'}}$.

We denote $||\mathbf{ID}|| = d'$ as the depth of hierarchy of $\mathbf{ID}$ and $S_{\mathbf{ID}}$ as the identity set associating with $\mathbf{ID}$. Figure 1 shows a network with a hierarchical structure and the users' identity vectors. Denote d as the maximum value of d' which expresses the number of level of hierarchy in the network. $\mathbf{V}$ is defined as a set of identity vectors. The notation $Par(\mathbf{V}) = \bigcup_{\mathbf{ID} \in \mathbf{V}} Par(\mathbf{ID})$ denotes the parent set of $\mathbf{V}$ and similarly $S_{\mathbf{V}}$ is the identity set associating to $\mathbf{V}$. We also denote $||V|| = \max\{d'_i : \mathbf{ID}_i \in \mathbf{V}\}$ as the maximum depth in V. Table 1 is a list of notations which are commonly used in this paper.

For example, according to Figure 2, if $\mathbf{V} = \{\mathbf{ID}_{22}, \mathbf{ID}_{31}, \mathbf{ID}_{33}\}$ such that $\mathbf{ID}_{33} = (\mathbf{ID}_{23}, ID_9)$= (ID_2, ID_5, ID_9), $\mathbf{ID}_{31} = (ID_1, ID_3, , ID_7)$, and

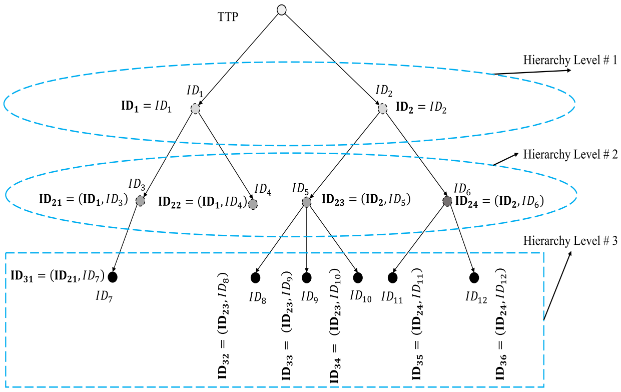
TTP
Hierarchy Level # 1
Hierarchy Level # 2
Hierarchy Level # 3
ID_1
$\mathbf{ID_1} = ID_1$
ID_2
$\mathbf{ID_2} = ID_2$
ID_3
$\mathbf{ID_{21}} = (\mathbf{ID_1}, ID_3)$
ID_4
$\mathbf{ID_{22}} = (\mathbf{ID_1}, ID_4)$
ID_5
$\mathbf{ID_{23}} = (\mathbf{ID_2}, ID_5)$
ID_6
$\mathbf{ID_{24}} = (\mathbf{ID_2}, ID_6)$
$\mathbf{ID_{31}} = (\mathbf{ID_{21}}, ID_7)$
ID_7
$\mathbf{ID_{32}} = (\mathbf{ID_{23}}, ID_8)$
ID_8
$\mathbf{ID_{33}} = (\mathbf{ID_{23}}, ID_9)$
ID_9
$\mathbf{ID_{34}} = (\mathbf{ID_{23}}, ID_{10})$
ID_{10}
ID_{11}
$\mathbf{ID_{35}} = (\mathbf{ID_{24}}, ID_{11})$
ID_{12}
$\mathbf{ID_{36}} = (\mathbf{ID_{24}}, ID_{12})$

Fig. 2. The example of a network with 12 users who are organized in 3 levels of hierarchy.

TABLE 1
Frequently Used Notations

Symbols	Description
$\mathbf{V}$	Identity vectors set, denotes as $\mathbf{V} = \{\mathbf{ID}_1, \ldots, \mathbf{ID}_{\lvert V \rvert}\}$
$Par(\mathbf{ID})$	The parents of identity vector **ID**
$\lvert\lvert \mathbf{ID} \rvert\rvert$	The depth of identity vector **ID**
$\lvert\lvert \mathbf{v} \rvert\rvert$	the depth of identity vector set **V**
$S_{\mathbf{ID}}$	Identity set associating to **ID**
$S_{\mathbf{V}}$	Identity set associating to **V**
$a \in_R S$	Randomly selection of a from set S
$\lvert S \rvert$	Denotes the cardinality of set S
$Out \leftarrow \mathsf{Alg}(In)$	Denotes that algorithm Alg outputs Out on input In
$A := B$	Allocating the value of B to A

$\mathbf{ID}_{22} = (ID_1, ID_4)$, then $S_{\mathbf{ID}_{33}} = \{ID_2, ID_5, ID_9\}$, $S_{\mathbf{V}} = \{ID_1, ID_2, ID_3, ID_4, ID_5, ID_7, ID_9\}$, and $||V|| = 3$.

The main goal of using hierarchical structures is to divide the burden of TTP in generating the users' secret key among the users. Therefore, the users with the higher level of hierarchy can generate the secret keys of their children. In addition, the users can delegate their decryption rights to their subordinates.

2.2 Composite order bilinear map

Composite order bilinear map for the first time was used in the cryptographic construction proposed in [18]. We define the group generator $\mathcal{G}$ as the algorithm which is run to output the description of the composite order bilinear map. Therefor, $\mathcal{G}(1^\lambda)$ takes as input the security parameter λ and outputs the tuple $\sum = (N = p_1p_2p_3p_4, G, G_T, e)$ where $N = p_1p_2p_3p_4$ is the order of cyclic groups G and G_T where $p_i, i = 1, \ldots, 4$ are four distinct prime numbers and $e : G \times G \to G_T$ is a composite order bilinear map with the following properties:

1) **Bilinearity**: $\forall g, h \in G$ and for all $a, b \in \mathbb{Z}_N^*, e(g^a, h^b) = e(g, h)^{ab}$
2) **Non-degeneracy**: $\exists g \in G$ such that the order of $e(g, g)$ is N in G_T

Let G_{p_i} where $i \in \{1, 2, 3, 4\}$ is a subgroup of G with order of p_i. If $f \in G_{P_i}$ and $h \in G_{p_j}$ are two elements of different subgroups (i.e., $i \neq j$), then $e(f, h) = 1$ and it is called the *orthogonality property* [16]. We note that this important feature is used in the proposed construction of anonymous HIBBE. In the following, we prove that the orthogonally property is always holding [19]. Let g be the generator of G and $f \in G_{p_1}$ and $h \in G_{p_2}$. Therefor, $g^{p_2p_3p_4}$ and $g^{p_1p_3p_4}$ are the generators of G_{p_1} and G_{p_2} respectively and we have $G_{p_1} =< g^{p_2p_3p_4} >$ and $G_{p_2} =< g^{p_1p_3p_4} >$. Hence, for some a_1, a_2, $f = (g^{p_2p_3p_4})^{a_1}$ and $h = (g^{p_1p_3p_4})^{a_2}$. Then:

$$e(f, h) = e((g^{p_2p_3p_4})^{a_1}, (g^{p_1p_3p_4})^{a_2}) = e(g^{a_1p_3p_4}, g^{a_2})^{p_1p_2p_3p_4} = 1 \tag{1}$$

In what follows, we review some complexity assumptions which are used to prove the security of the proposed scheme.

In the assumptions below, G_{abc} denotes a subgroup of order abc in G where $a, b, c \in \{1, p_1, p_2, p_3, p_4\}$.

2.2.1 *Assumption 1*

Let $\mathcal{G}(1^\lambda)$ be the group generator which is run to compute $\sum = (N = p_1p_2p_3p_4, G, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$. Suppose that the distribution D is set as follows:

$$g_1, A_1 \leftarrow G_{p_1}, A_2, B_2 \leftarrow G_{p_2}, g_3 \leftarrow G_{p_3}, g_4, B_4 \leftarrow G_{p_4}$$
$$D := (\textstyle\sum, g_1, g_3, g_4, A_1A_2, B_2B_4)$$

Suppose that the PPT algorithm $\mathcal{A}$ is given D and tries to distinguish between $T_1 \leftarrow G_{p_1p_2p_4}$ and $T_2 \leftarrow G_{p_1p_4}$. In this case, the advantage of $\mathcal{A}$ is defined as follows:

$$\begin{aligned} Adv_{A1}^{\mathcal{A}} = |\Pr[&\mathcal{A}(D, T_1 \leftarrow G_{p_1p_2p_4}) = 1] \\ &- \Pr[\mathcal{A}(D, T_2 \leftarrow G_{p_1p_4}) = 1]| \end{aligned} \tag{2}$$

Definition 1. Assumption 1 holds if for all PPT algorithms $\mathcal{A}$, $Adv_{A1}^{\mathcal{A}}$ is a negligible function in security parameter λ [16].

2.2.2 *Assumption 2*

Let $\mathcal{G}(1^\lambda)$ be the group generator which is run to compute $\sum = (N = p_1p_2p_3p_4, G, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$. Suppose that the distribution D is set as follows:

$$g_1, A_1 \leftarrow G_{p_1}, A_2, B_2 \leftarrow G_{p_2}, g_3, B_3 \leftarrow G_{p_3}, g_4 \leftarrow G_{p_4}$$
$$D := (\textstyle\sum, g_1, g_3, g_4, A_1A_2, B_2B_3)$$

Suppose that the PPT algorithm $\mathcal{A}$ is given D and tries to distinguish between $T_1 \leftarrow G_{p_1p_2p_3}$ and $T_2 \leftarrow G_{p_1p_3}$. In this case, the advantage of $\mathcal{A}$ is defined as follows:

$$\begin{aligned} Adv_{A2}^{\mathcal{A}} = |\Pr[&\mathcal{A}(D, T_1 \leftarrow G_{p_1p_2p_3}) = 1] \\ &- \Pr[\mathcal{A}(D, T_2 \leftarrow G_{p_1p_3}) = 1]| \end{aligned} \tag{3}$$

Definition 2. Assumption 2 holds if for all PPT algorithms $\mathcal{A}$, $Adv_{A2}^{\mathcal{A}}$ is a negligible function in security parameter λ.

2.2.3 *Assumption 3*

Let $\mathcal{G}(1^\lambda)$ be the group generator which is run to compute $\sum = (N = p_1p_2p_3p_4, G, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$. Suppose that the distribution D is set as follows:

$$\alpha, s, r \leftarrow \mathbb{Z}_N, g_1 \leftarrow G_{p_1}, g_2, A_2, B_2 \leftarrow G_{p_2}, g_3 \leftarrow G_{p_3},$$
$$g_4 \leftarrow G_{p_4}; D := (\textstyle\sum, g_1, g_2, g_3, g_4, g_1^{\alpha}A_2, g_1^{s}B_2, g_2^{r}, A_2^{r})$$

Suppose that the PPT algorithm $\mathcal{A}$ is given D and tries to distinguish between $T_1 = e(g_1, g_1)^{\alpha s}$ and $T_2 \leftarrow G_T$. In this case, the advantage of $\mathcal{A}$ is defined as follows:

$$Adv_{A3}^{\mathcal{A}} = |\Pr[\mathcal{A}(D, T_1 = e(g_1, g_1)^{\alpha s}) = 1] - \Pr[\mathcal{A}(D, T_2 \leftarrow G_T) = 1]| \quad (4)$$

Definition 3. Assumption 3 holds if for all PPT algorithms $\mathcal{A}$, $Adv_{A3}^{\mathcal{A}}$ is a negligible function in security parameter λ [16].

2.2.4 Assumption 4

Let $\mathcal{G}(1^\lambda)$ be the group generator which is run to compute $\sum = (N = p_1p_2p_3p_4, G, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$. Suppose that the distribution D is set as follows:

$$\hat{r}, s \leftarrow \mathbb{Z}_N, g_1, U, A_1 \leftarrow G_{p_1}, g_2, A_2, B_2, D_2, F_2 \leftarrow G_{p_2},$$
$$g_3 \leftarrow G_{p_3}, g_4, A_4, B_4, D_4 \leftarrow G_{p_4}, A_{24}, B_{24}, D_{24} \leftarrow G_{p_2p_4}$$
$$D :=$$
$$(\sum, g_1, g_2, g_3, g_4, U, U^s A_{24}, U^{\hat{r}}, A_1A_4, A_1^{\hat{r}}A_2, g_1^{\hat{r}}B_2, g_1^s B_{24})$$

Suppose that the PPT algorithm $\mathcal{A}$ is given D and tries to distinguish between $T_1 = A_1^s D_{24}$ and $T_2 \leftarrow G_{p_1p_2p_4}$. In this case, the advantage of $\mathcal{A}$ is defined as follows:

$$Adv_{A4}^{\mathcal{A}} = |\Pr[\mathcal{A}(D, T_1 = A_1^s D_{24}) = 1] - \Pr[\mathcal{A}(D, T_2 \leftarrow G_{p_1p_2p_4}) = 1]| \quad (5)$$

Definition 4. Assumption 4 holds if for all PPT algorithms $\mathcal{A}$, $Adv_{A4}^{\mathcal{A}}$ is a negligible function in security parameter λ [16].

3 Hierarchical Identity Based Broadcast Encryption (HIBBE)

Typically, each HIBBE scheme contains four polynomial time algorithms: Setup, Extract, BroadEnc, and BroadDec. These four algorithms are presented in the following.

- $(PP, MSK) \leftarrow \mathsf{Setup}(\lambda)$: This algorithm takes λ as input and generates MSK and PP.
- $SK_{\mathbf{ID}} \leftarrow \mathsf{Extract}(SK_{Par(\mathbf{ID})}, \mathbf{ID}, PP)$: The inputs of this algorithm are $SK_{Par(\mathbf{ID})}$ as the parent's secret key of $\mathbf{ID}$, the identity vector $\mathbf{ID}$, and PP. It outputs the secret key $SK_{\mathbf{ID}}$. Note that in the hierarchical model, TTP is in the top level of hierarchy with identity vector $\mathbf{ID}_0 = ID_{TTA}$. So, its secret key is denoted as $SK_0 = MSK$.
- $C \leftarrow \mathsf{BroadEnc}(PP, M, \mathbf{V})$: This algorithm takes PP, the message $M \in \mathcal{M}$, and the identity vector set $\mathbf{V}$ as inputs, encrypts M and outputs the resulting ciphertext C.
- $M := \mathsf{BroadDec}(PP, C, SK_{\mathbf{ID}})$: This algorithm takes as input PP, C, and the secret key $SK_{\mathbf{ID}}$. If $\mathbf{ID} \in Par(\mathbf{V})$, then the output of this algorithm is the decryption of C, i.e. M.

3.1 Anonymous HIBBE

In the following, we define anonymity of the HIBBE against the chosen identity vector set and chosen plaintext attacks (Anon-CIVS-CPA). In an anonymous HIBBE scheme, the adversary can not distinguish between the ciphertexts of a message which are encrypted using different identity vector sets of its choice. Consequently, he/she does not infer any information about the identities of designated receivers. As an application of anonymous HIBBE, it should be noted that this primitive is employed as a building block in the proposed generic construction of VABKS [20].

The security game of Anon-CIVS-CPA consist of five steps: **Setup**, **Phase 1**, **Challenge**, **Phase 2** and **Guess** which are defined as follows.

In this game, $\mathcal{A}$ tries to learn some information about the identity of the receivers from the challenge ciphertext. he/she is allowed to access the secret key extraction oracle adaptively to obtain the users' secret keys of its choice. Following we formally define Anon-CIVS-CPA security of HIBBE, through the game which holds between the challenger, $\mathcal{C}$, and $\mathcal{A}$.

Setup: $\mathcal{C}$ runs the algorithm, $(MSK, PP) \leftarrow \mathsf{Setup}(\lambda)$, and sends PP to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ is allowed to access to the extract oracle on the identity vectors of its choice.

- $SK_i \leftarrow \mathcal{O}_{\mathsf{Extract}}(\mathbf{ID}_i)$: First of all, before $\mathcal{A}$ starts to query, $\mathcal{C}$ selects the identity vector list, $L_{\mathbf{ID}}$, which is initially empty. In this oracle, $\mathcal{A}$ adaptively issues $\mathbf{ID}_i$ to $\mathcal{C}$. Then $\mathcal{C}$ generates the secret key SK_i and sends it to $\mathcal{A}$. Then $\mathcal{C}$ adds the queried $\mathbf{ID}_i$ to the list of identity vectors, $L_{\mathbf{ID}}$.

Challenge: $\mathcal{A}$ selects the two identity vector sets $\mathbf{V}_0$, $\mathbf{V}_1$, where $||\mathbf{V}_0|| = ||\mathbf{V}_1||$ and $L_{\mathbf{ID}} \bigcap \{\bigcup_{i=0,1} Par(V_i)\} = \emptyset$. $\mathcal{A}$ also selects the message, $M \in \mathcal{M}$, and sends $(\mathbf{V}_0, \mathbf{V}_1, M)$ to $\mathcal{C}$. Then, $\mathcal{C}$ selects the random bit $b \in_R \{0, 1\}$ and encrypts the message M by $\mathbf{V}_b$. Then $\mathcal{C}$ sends $C_b \leftarrow \mathsf{BroadEnc}(PP, M, \mathbf{V}_b)$ to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ continues querying $\mathcal{O}_{\mathsf{Extract}}(\mathbf{ID})$ to receive the secret key of the identity vector, $\mathbf{ID}$, such that $\mathbf{ID} \notin Par(\mathbf{V}_0) \cup Par(\mathbf{V}_1)$. This condition should be hold to prevent the adversary from trivially guessing the value of b.

Guess: Finally, adversary $\mathcal{A}$ outputs $b' \in \{0, 1\}$ as a guess for the value of b, and wins the game, if $b = b'$.

We define the advantage of the PPT adversary $\mathcal{A}$ in attacking the anonymity of HIBBE system with security parameter λ as follows:

$$Adv_{\mathcal{A},HIBBE}^{Anon-CIVS-CPA}(\lambda) = |\Pr[\mathcal{A}(\lambda) = b' : b' = b] - \frac{1}{2}| \quad (6)$$

Definition 5 (Anon-CIVS-CPA). A HIBBE is Anon-CIVS-CPA secure against PPT adversary $\mathcal{A}$ if its advantage is a negligible function:

$$Adv_{\mathcal{A},HIBBE}^{Anon-CIVS-CPA}(\lambda) \leq \mathsf{negl}(\lambda) \quad (7)$$

3.2 Concrete construction of anonymous HIBBE

With the inspiration of the proposed anonymous HIBBE in [16], in this section, we propose a concrete scheme for anonymous HIBBE. Actually, we transfer the HIBE scheme of [16] to anonymous HIBBE with the idea of broadcasting which is used in the anonymous identity-based broadcast encryption scheme presented in [13].

Setup ($(\mathsf{msk}, \mathsf{pp}) \leftarrow \mathsf{Setup}(\lambda)$): Suppose that $l = l_\lambda$ denotes the maximum depth of the hierarchy, and $\mathcal{I} = (N, G, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$ the composite order bilinear map. This algorithm selects the random elements $Y_1, X_1, u_i, \dots, u_l \in_R G_{p_1}, Y_3 \in G_{p_3}, X_4, Y_4 \in_R G_{p_4}$ and $\alpha \in_R \mathbb{Z}_N$. It also selects the hash function $H_{14} : \boldsymbol{\mathcal{ID}} \to \mathbb{Z}_N$ where $\boldsymbol{\mathcal{ID}}$ is the universal set of all identity vectors. We denote the public parameters by the tuple $\mathsf{pp} := (N, Y_1, Y_3, Y_4, t = X_1X_4, u_1, \dots, u_l, A = e(Y_1, Y_1)^\alpha, H_{14})$ and the master secret key is the tuple $\mathsf{msk} := (X_1, \alpha)$.

Key generation ($\mathsf{sk}_{\mathbf{ID}_i} \leftarrow \mathsf{Extract}(\mathsf{sk}_{ID_i}, \mathbf{ID}_i = (\mathbf{ID}_{i-1}, ID_i), \mathsf{pp})$): We consider the key extraction algorithm according to the two scenarios: 1) $\mathsf{sk}_{ID_i} = \mathsf{msk}$ and the TTP itself generates the secret keys of the users, and 2) $\mathsf{sk}_{ID_i} \neq \mathsf{msk}$ where we called the second scenario delegation phase and the users in the higher level of hierarchy generates the secret keys of their subordinates using their delegation rights.

First scenario: Given the identity vector $\mathbf{ID}_i = (ID_1, \dots, ID_i)$, TTP chooses random numbers $r_1, r_2 \in_R \mathbb{Z}_N$, and for $t = 1, 2$, $R_{t,1}, R_{t,2}, R_{t,i+1}, \dots, R_{t,l} \in_R G_{p_3}$. The secret key is tuple $\mathsf{sk}_{\mathbf{ID}_i} = (\mathsf{K}_{t,1}, \mathsf{K}_{t,2}, \mathsf{E}_{t,i+1}, \dots, \mathsf{E}_{t,l})$ which is computed as follows:

$$\begin{aligned}
\mathsf{K}_{1,1} &= Y_1^{r_1} R_{1,1}, \mathsf{K}_{1,2} = Y_1^{\alpha}\left(u_1^{ID_1} \dots u_i^{ID_i} X_1\right)^{r_1} R_{1,2} \\
\mathsf{E}_{1,i+1} &= u_{i+1}^{r_1} R_{1,i+1}, \dots, E_{1,l} = u_l^{r_1} R_{1,l} \\
\mathsf{K}_{2,1} &= Y_1^{r_2} R_{2,1}, \mathsf{K}_{2,1} = \left(u_1^{ID_1} \dots u_i^{ID_i} X_1\right)^{r_2} R_{2,2} \\
\mathsf{E}_{2,i+1} &= u_{i+1}^{r_2} R_{2,i+1}, \dots, \mathsf{E}_{2,l} = u_l^{r_2} R_{2,l}
\end{aligned}$$

Delegation phase: Suppose that the parent's secret key of $\mathbf{ID}_i$ is $\mathsf{sk}_{\mathbf{ID}_{i-1}} = (\mathsf{K}'_{t,1}, \mathsf{K}'_{t,2}, \mathsf{E}'_{t,i}, \dots, \mathsf{E}'_{t,l} = u_l^{r_1} R_{t,l})$ and $\mathbf{ID}_{i-1} = (ID_1, \dots, ID_{i-1})$. In this phase of algorithm, the secret key of user with identity vector $\mathbf{ID}_i$ will be generated as follows. First, it choose the random numbers $\tilde{r}_1, \tilde{r}_2 \in_R \mathbb{Z}_N$ and, for $t \in \{1, 2\}$, $R_{t,1}, R_{t,2}, R_{t,i+1}, \dots, R_{t,l} \in_R G_{p_3}$.

$$\begin{aligned}
\mathsf{K}_{1,1} &= \mathsf{K}'_{1,1}(\mathsf{K}'_{2,1})^{\tilde{r}_1} R_{1,1} \\
\mathsf{K}_{1,2} &= \mathsf{K}'_{1,2}(\mathsf{K}'_{2,2})^{\tilde{r}_1} (\mathsf{E}'_{1,i})^{ID_i} (\mathsf{E}'_{2,i})^{\tilde{r}_1 ID_i} R_{1,2} \\
\mathsf{E}_{1,i+1} &= \mathsf{E}'_{1,i+1}(\mathsf{E}'_{2,i+1})^{\tilde{r}_1} R_{1,i+1}, \dots, \mathsf{E}_{1,l} = \mathsf{E}'_{1,l}(\mathsf{E}'_{2,l})^{\tilde{r}_1} R_{1,l} \\
\mathsf{K}_{2,1} &= (\mathsf{K}'_{2,1})^{\tilde{r}_2} R_{2,1}, \mathsf{K}_{2,2} = (\mathsf{K}'_{2,2})^{\tilde{r}_2} (\mathsf{E}'_{2,i})^{\tilde{r}_2 ID_i} R_{2,2} \\
\mathsf{E}_{2,i+1} &= (\mathsf{E}'_{2,i+1})^{\tilde{r}_2} R_{2,i+1}, \dots, \mathsf{E}_{2,l} = \mathsf{E}'_{2,l}(\mathsf{E}'_{2,l})^{\tilde{r}_2} R_{2,l}
\end{aligned}$$

Note that, the private keys contain two parts, one part is used for decryption of the ciphertext (i.e., $(\mathsf{K}_{1,1}, \mathsf{K}_{1,2}, \mathsf{E}_{1,i+1}, \dots, \mathsf{E}_{1,l})$) and the second part is used for the delegation part (i.e., $(\mathsf{K}_{2,1}, \mathsf{K}_{2,2}, \mathsf{E}_{2,i+1}, \dots, \mathsf{E}_{2,l})$).

Encryption ($\mathsf{cph} \leftarrow \mathsf{BroadEnc}(\mathsf{pp}, M, \mathbf{V} = \{\mathbf{ID}_1, \dots, \mathbf{ID}_L\})$): Suppose that for h from 1 to L, $\mathbf{ID}_h = (ID_{h,1}, \dots, ID_{h,i_h})$. This algorithm selects random numbers $s \in_R \mathbb{Z}_N, Z, Z' \in_R G_{p_4}$ and computes the ciphertext cph for the message $M \in G_T$ as follows:

$$\begin{aligned}
& C_0 = M.A^s \\
& \forall h = 1, \dots, L : \left\{ C'_{1,h} = \left(u_1^{ID_{h,1}} \dots u_{i_h}^{ID_{t,i_h}} t\right)^s Z \right. \\
& x_h = H_{14}(\mathbf{ID}_h), f_h(x) = \prod_{j \neq h} \frac{x - x_j}{x_h - x_j} = \sum_{j=1}^{L-1} a_{hj} x^j, \\
& so\ \ f_h(x_h) = 1 \,\&\, f_h(x_j) = 0 : h \neq j \\
& \left. C_{1,h} = \prod_{j=1}^{L} C'^{a_{jh}}_{1,j} \right\} \\
& C_1 = (C_{1,1}, \dots, C_{1,L}), C_2 = (Y_1)^s Z' \\
& \mathsf{cph} := (C_0, C_1, C_2)
\end{aligned} \tag{8}$$

Decryption ($M := \mathsf{BroadDec}(\mathsf{pp}, \mathsf{cph}, \mathsf{sk}_{\mathbf{ID}})$): If the identity vector $\mathbf{ID} \in \mathbf{V}$, then its associated user can decrypt the ciphertext. So this algorithm first compute $x^* = H_{14}(\mathbf{ID})$ and acts as follows:

$$\begin{aligned}
C^* &= \prod_{j=1}^{L} C_{1,j}^{(x^*)^{j-1}} \\
&= C_{11}^{(x^*)^0} \times C_{12}^{(x^*)^1} \times \dots \times C_{1L}^{(x^*)^{L-1}} \\
&= (\prod_{j=1}^{L} C'_{1j}{}^{a_{j1}}) \times \dots \times (\prod_{j=1}^{L} C'_{1j}{}^{a_{jL}})^{(x^*)^{L-1}} \\
&= \left(C'_{11}{}^{a_{11}} \times \dots \times C'_{11}{}^{a_{1L}(x^*)^{L-1}}\right) \times \dots \\
&\dots \times \left(C'_{1L}{}^{a_{L1}} \times \dots \times C'_{1L}{}^{a_{LL}(x^*)^{L-1}}\right) \\
&= C'_{11}{}^{\sum_{j=1}^{L} a_{1j}(x^*)^{j-1}} \times \dots \times C'_{1L}{}^{\sum_{j=1}^{L} a_{Lj}(x^*)^{j-1}} \\
&= C'_{11}{}^{f_1(x^*)} \times \dots \times C'_{1L}{}^{f_L(x^*)}
\end{aligned}$$

Therefore, if the identity vector $\mathbf{ID} \in \mathbf{V}$, then there exists one element in $\{1, \dots L\}$ like j in such a way that $x^* = x_j$, $f_j(x^*) = 1$ and $f_h(x^*) = 0$ for all $1 \leq h \leq L$ such that $h \neq j$. Consequently $C^* = C'_{1j}$ and the receiver can compute the original message as follows:

$$\begin{aligned}
C_0 \frac{e(\mathsf{K}_{1,1}, C^*)}{e(\mathsf{K}_{1,2}, C_2)} &= Me(Y_1, Y_1)^{\alpha s} \frac{e(\mathsf{K}_{1,1}, C^*)}{e(\mathsf{K}_{1,2}, C_2)} \\
&= \frac{Me(Y_1, Y_1)^{\alpha s} e(Y_1, (u_1^{ID_{j,1}} \dots u_{i_j}^{ID_{j,i_j}}))^{r_1 s}}{e(Y_1, Y_1)^{\alpha s} e(Y_1, (u_1^{ID_{j,1}} \dots u_{i_j}^{ID_{j,i_j}}))^{r_1 s}} \\
&= M
\end{aligned} \tag{9}$$

4 Security analysis of the proposed construction of anonymous HIBBE

4.1 The proof overview

It is shown that the proposed scheme of HIBE in [16] is anonymous. The authors to show that their scheme is anonymous they defined two additional structures: *semi-functional ciphertext* and *semi-functional keys*. Anonymity of our proposed scheme can be proved in a similar way to the security proof presented to show that the HIBE scheme of [16] is anonymous. To this aim we should define the semi-functional ciphertext as follows. Denote that the semi-functional key are defined exactly the same as [16] without any change.

Semi-functional Ciphertext: Suppose that the tuple $(C'_0, \{C'_{1,1}, \dots, C'_{1,L}\}, C'_2)$ be the ciphertext of the normal encryption algorithm. Let that g_2 denote a generator of group G_{p_2}. The exponents $x, z_1, \dots, z_L \in_R \mathbb{Z}_N$ are chosen uniformly at random. The semi-functional ciphertext is computed as follows:

$$C_0 = C'_0, \{C_{1,t} = C'_{1,t} g_2^{xz_t} : \forall 1 \le t \le L\}, C_2 = C'_2 g_2^{x} \quad (10)$$

Semi-functional Secret Key: Let $(\mathsf{K}'_{t,1}, \mathsf{K}'_{t,2}, \mathsf{E}'_{t,i+1}, \dots, \mathsf{E}'_{t,l})$ be the normal secret key which is generated in the extraction algorithm. The exponents $z, \gamma, z_k \in \mathbb{Z}_N$ and, for $t \in \{1, 2\}$ the random exponent $z_{t,i+1}, \dots, z_{t,l} \in \mathbb{Z}_N$ are chosen uniformly at random. The semi-functional secret key is computed as follows:

$$\mathsf{K}_{1,1} = \mathsf{K}'_{1,1} g_2^{\gamma}, \mathsf{K}_{1,2} = \mathsf{K}'_{1,2} g_2^{\gamma z_k}, \{\mathsf{E}_{1,t} = \mathsf{E}'_{1,t} g_2^{\gamma z_{1,t}}\}_{t=i+1}^{l} \quad (11)$$

$$\mathsf{K}_{2,1} = \mathsf{K}'_{2,1} g_2^{z\gamma}, \mathsf{K}_{2,2} = \mathsf{K}'_{2,2} g_2^{z\gamma z_k}, \{\mathsf{E}_{2,t} = \mathsf{E}'_{2,t} g_2^{z\gamma z_{2,t}}\}_{t=i+1}^{l} \quad (12)$$

We note that by using the first and second sub-keys of the semi-functional secret key associated to the authorized user with identity vector **ID**, the decryption of the semi-functional ciphertexts are respectively $Me(g_2, g_2)^{x\gamma(f_z(x^*)-z_k)}$ and $Me(g_2, g_2)^{xz\gamma(f_z(x^*)-z_k)}$ where $f_z(x) = z_1 + z_2 x + \dots + z_L x^{L-1}$ and $x^* = H_{14}(\mathbf{ID})$. If $f_z(x^*) = z_k$ then the decryption will still work.

We denote that the prove of anonymity of the proposed HIBBE relays on Assumptions 1, 2, 3, and 4. Note that the PPT adversary $\mathcal{A}$ is allowed to make q queries for the secret keys and according to the hybrid argument, based on the following sequence of $q + 5$ games between $\mathcal{A}$ and the challenger $\mathcal{C}$, we prove that the anonymity of our scheme is reduced to these three assumptions.

$\mathsf{Game}_{\mathsf{Real}}$: is exactly the same as the real game which defines the anonymity of HIBBE.

$\mathsf{Game}_{\mathsf{Real}'}$: is the same as the real game except that all the queries for the secret keys are responded by the first scenario of the key generation algorithm ($\mathcal{C}$ will not be asked to respond the queries by the Delegation phase of the key generation algorithm).

$\mathsf{Game}_{\mathsf{Restricted}}$: is the same as $\mathsf{Game}_{\mathsf{Real}'}$ except that $\mathcal{A}$ can not query the secret keys associated to the identities which are prefix of one of the challenge identities module p_2.

Game_k: for $0 \le k \le q$, Game_k is similar to $\mathsf{Game}_{\mathsf{Restricted}}$ except that the given challenge ciphertext to $\mathcal{A}$ is semi-functional and the first k secret keys are generated in the semi functional format and the rest $q - k$ queried secret keys are normal.

$\mathsf{Game}_{\mathsf{Final}_0}$: is exactly similar to Game_q and the only different is that C_0 of the challenge semi functional ciphertext is independent from the messages selected by $\mathcal{A}$. Therefor in this game C_0 is chosen uniformly at random from G_T.

$\mathsf{Game}_{\mathsf{Final}_1}$: is exactly similar to $\mathsf{Game}_{\mathsf{Final}_0}$ and the only difference is that in the semi-functional ciphertext, for t from 1 to l, $C_{1,t}$ are chosen from $G_{p_1 p_2 p_4}$ uniformly at random. Therefore, the semi-functional ciphertext is independent from the challenge identities which are chosen by $\mathcal{A}$ in the challenge phase and the advantage of all the adversaries is 0.

4.2 Indistinguishability of $\mathsf{Game}_{\mathsf{real}}$ and $\mathsf{Game}_{\mathsf{real}'}$

Lemma 1. For any adversary $\mathcal{A}$, $Adv^{\mathcal{A}}_{\mathsf{Game}_{\mathsf{real}}}$ is equal to $Adv^{\mathcal{A}}_{\mathsf{Game}_{\mathsf{real}'}}$.

Proof. As in the both scenarios of the key generation algorithm, i.e., **first scenario** and **Delegation phase**, the secret keys are distributed identically; therefore, in the adversary's view, there is no different between these security games.

4.3 Indistinguishability of $\mathsf{Game}_{\mathsf{real}'}$ and $\mathsf{Game}_{\mathsf{Restricted}}$

Lemma 2. Consider that there exists a PPT adversary $\mathcal{A}$ such that $Adv^{\mathcal{A}}_{\mathsf{Game}_{\mathsf{real}'}} - Adv^{\mathcal{A}}_{\mathsf{Game}_{\mathsf{Restricted}}} = \epsilon$. Then there exists a PPT adversary $\mathcal{B}$ whose advantage in breaking Assumption 1 is at least $\frac{\epsilon}{3}$.

Proof. Suppose that there exists an adversary $\mathcal{A}$ who can find the identity vector $\mathbf{ID} = (ID_1, \dots, ID_k)$ which is the prefix of one of the elements of the challenge identity vectors sets, like $\mathbf{ID}^* = (ID_1^*, \dots, ID_j^*)$, module p_2 with the probability of ϵ. Therefore, there exists i such that $ID_i \neq ID_i^*$ module N and p_2 divides $ID_i - ID_i^*$. As a result $a = \gcd(ID_i - ID_i^*, N)$ is a non-trivial factor of N. Note that p_2 divides a. We set $b = \frac{N}{a}$ and the following three cases are all possible states and at least one of them with probability at least $\frac{\epsilon}{3}$ occurs.

1. $\mathsf{ord}(Y_1) \mid b$.
2. $\mathsf{ord}(Y_1) \nmid b \; and \; \mathsf{ord}(Y_4) \mid b$.
3. $\mathsf{ord}(Y_1) \nmid b \; and \; \mathsf{ord}(Y_4) \nmid b \; and \; \mathsf{ord}(Y_3) \mid b$.

Without loss of generality we consider case 1 which has the probability at least $\frac{\epsilon}{3}$. Then we construct the PPT adversary $\mathcal{B}$ who breaks Assumption 1. $\mathcal{B}$ is given the tuple $D := (\sum, g_1, g_3, g_4, A_1A_2, B_2B_4)$ and T. Then it runs Setup to computes PP. In this case, $\mathcal{B}$ sets $Y_1 = g_1$, $Y_3 = g_3$, and $Y_4 = g_4$. As $\mathcal{B}$ runs the setup algorithm, it know the master secret key MSK related to PP. Therefor, by means of MSK, $\mathcal{B}$ answers all $\mathcal{A}$'s queries. At the end of the game, $\mathcal{B}$ for all **ID**s which are the identity vectors that have been queried by $\mathcal{A}$, $\mathcal{B}$ computes $a = \gcd(ID_i - ID_i^*)$ for all $\mathbf{ID}^*$s are included in the two challenge identity vector sets. If $e((A_1A_2)^a, B_2B_4)$ is the identity element of G_T, then it tests whether $e(T^b, A_1A_2)$ is an identity element of G_T or not. If the response of the second test is positive, then $\mathcal{B}$ promulgates that $T \in G_{p_1p_4}$; otherwise, it promulgates that $T \in G_{p_1p_2p_4}$. As a result, $\mathcal{B}$ can is able to break Assumption 1 with the probability at least $\frac{\epsilon}{3}$. As we suppose that all PPT adversaries breaks Assumption 1 with a negligible advantage, we can conclude that $\frac{\epsilon}{3}$ is a negligible functions and consequently ϵ is a negligible function. So $\mathsf{Game}_{\mathsf{real}'}$ and $\mathsf{Game}_{\mathsf{Restricted}}$ are indistinguishable.

4.4 Indistinguishability of $\mathsf{Game}_{\mathsf{Restricted}}$ and Game_0

Lemma 3. If there exists a PPT adversary $\mathcal{A}$ such that $Adv^{\mathcal{A}}_{\mathsf{Game}_{\mathsf{Restricted}}} - Adv^{\mathcal{A}}_{\mathsf{Game}_0} = \epsilon$, then there exists a PPT adversary $\mathcal{B}$ whose advantage in breaking Assumption 1 is ϵ.

Proof.$\mathcal{B}$ is given $D := (\sum, g_1, g_3, g_4, A_1A_2, B_2B_4)$ and T. $\mathcal{B}$ simulates $\mathsf{Game}_{\mathsf{Restricted}}$ or Game_0 according to the value of T. So, if $T \in G_{p_1p_4}$ then it simulates $\mathsf{Game}_{\mathsf{Restricted}}$ and if $T \in G_{p_1p_2p_4}$ it simulates Game_0 for $\mathcal{A}$.

Then $\mathcal{B}$ runs the setup algorithm to generate MSK and PP as follows. It first selects the exponents $\alpha, a_1, \ldots, a_l, b, c \in_R \mathbb{Z}_N$ uniformly at random and then sets $Y_1 = g_1$, $Y_3 = g_4$, $Y_4 = g_3, X_4 = Y_4^c$, $X_1 = Y_1^b$ and for i from 1 to l, $u_i = Y_1^{a_i}$. Then, $\mathcal{B}$ sets the public parameter $PP := (N, Y_1, Y_3, Y_4, t = X_1 X_4, u_1, \ldots, u_l, \Omega = e(Y_1, Y_1)^\alpha)$ and sends it to $\mathcal{A}$. $\mathcal{B}$ also sets the $MSK = (X_1, \alpha)$ as the master secret key which is related to PP and as it knows the master secret key, it can answer all of $\mathcal{A}$'s queries for the secret keys of the intended identity vectors.

As it mentioned in the challenge phase of Anon-CIV-CPA, $\mathcal{A}$ outputs the message M and two challenge identity vector sets $\mathbf{V}_t = \{\mathbf{ID}^*_{1t}, \ldots, \mathbf{ID}^*_{Lt}\}$, $t \in \{0, 1\}$. Then, $\mathcal{B}$ flips a random coin $\beta \in_R \{0, 1\}$ and encrypts M using $\mathbf{V}_\beta$ as follows:

Suppose that for h from 1 to L, $\mathbf{ID}^*_{h\beta} = (ID_{\beta h,1}, \ldots, ID_{\beta h,i_h})$. This algorithm selects random numbers $s \in_R \mathbb{Z}_N, Z, Z' \in_R G_{p_4}$ and computes the ciphertext cph for the message $M \in G_T$ as follows:

$$\begin{aligned}
&C_0 = M.e(T, Y_1)^\alpha \\
&\forall h = 1, \ldots, L : \left\{ \begin{aligned} &C'_{1,h} = T^{a_1 ID_{\beta h,1} + \cdots + a_{i_h} ID_{\beta h,i_h} + b} \\
&x_h = H_{14}(\mathbf{ID}_{ht}), f_h(x) = \prod_{j \neq h} \frac{x - x_j}{x_h - x_j} = \sum_{j=1}^{L-1} a_{hj} x^j, \\
&so \ \ f_h(x_h) = 1 \,\&\, f_h(x_j) = 0 : h \neq j \\
&C_{1,h} = \prod_{j=1}^{L} C'^{a_{jh}}_{1,j} \end{aligned} \right\} \\
&C_1 = (C_{1,1}, \ldots, C_{1,L}), C_2 = T \\
&\mathsf{cph} := (C_0, C_1, C_2) \qquad (13)
\end{aligned}$$

The proof will become completed with the following two cases. If $T \in G_{p_1 p_3}$, then we can show $T = Y_1^{s_1} Y_4^{s_3}$. It can be seen that in this case, (C_0, C_1, C_2) is a normal ciphertext with randomness $s = s_1$, $Z_h = Y_4^{s_3(a_1 ID_{\beta h,1} + \cdots + a_{i_h} ID_{\beta h,i_h} + b)}$ and $Z' = Y_4^{s_3}$. If $T \in G_{p_1 p_2 p_3}$, then we can show $T = Y_1^{s_1} g_2^{s_2} Y_4^{s_3}$. It can be seen that in this case, (C_0, C_1, C_2) is a semi-functional ciphertext with randomness $s = s_1, Z_h = Y_4^{s_3(a_1 ID_{\beta h,1} + \cdots + a_{i_h} ID_{\beta h,i_h} + b)}$, $x = s_2$ and this ciphertext implicitly sets $z_1, \ldots, z_L$ to random values.

4.5 Indistinguishability of Game_{k-1} and Game_k

Lemma 4. If there exists a PPT adversary $\mathcal{A}$ such that $Adv^{\mathcal{A}}_{\mathsf{Game}_{k-1}} - Adv^{\mathcal{A}}_{\mathsf{Game}_k} = \epsilon$, then there exists a PPT adversary $\mathcal{B}$ whose advantage in breaking Assumption 2 is ϵ.

Proof. $\mathcal{B}$ is given $D := (\sum, g_1, g_3, g_4, A_1 A_2, B_2 B_3)$ and T. According to the value of T, $\mathcal{B}$ simulates Game_{k-1} or Game_k. We will show that, if $T \in G_{p_1 p_3}$ then it simulates Game_{k-1} and if $T \in G_{p_1 p_2 p_3}$ it simulates Game_k for $\mathcal{A}$.

Then $\mathcal{B}$ runs the setup algorithm to generate MSK and PP as follows. It first selects the exponents $\alpha, a_1, \ldots, a_l, b, c \in_R \mathbb{Z}_N$ uniformly at random and then sets $Y_1 = g_1$, $Y_3 = g_3$, $Y_4 = g_4, X_4 = Y_4^c$, $X_1 = Y_1^b$ and for i from 1 to l, $u_i = Y_1^{a_i}$. Then, $\mathcal{B}$ sets $PP := (N, Y_1, Y_3, Y_4, t = X_1 X_4, u_1, \ldots, u_l, \Omega = e(Y_1, Y_1)^\alpha)$ as the public parameter and sends it to $\mathcal{A}$. $\mathcal{B}$ also sets $MSK = (X_1, \alpha)$ as the master secret key which is related to PP. $\mathcal{B}$ answers the i-th secret key query for identity vector ($\mathbf{ID}_i = (ID_{i,1}, \ldots, ID_{i,j})$). $\mathcal{B}$ generates semi-functional secret keys in the first $k - 1$ queries by choosing the $r_1, r_2, f, z, \omega \in_R \mathbb{Z}_N$ and, for $t \in \{1, 2\}$, $\omega_{t,2}, \omega_{t,j+1}, \ldots, \omega_{t,l} \in_R \mathbb{Z}_N$ uniformly at random. Then $\mathcal{B}$ acts as follows:

$$\begin{aligned}
&\mathsf{K}_{1,1} = Y_1^{r_1} (B_2 B_3)^f \\
&\mathsf{K}_{1,2} = Y_1^\alpha (B_2 B_3)^\omega \left(u_1^{ID_{i,1}} \ldots u_j^{ID_{i,j}} X_1 \right)^{r_1} Y_3^{\omega_{1,2}} \\
&\mathsf{E}_{1,j+1} = u_{j+1}^{r_1} (B_2 B_3)^{\omega_{1,j+1}}, \ldots, E_{1,l} = u_l^{r_1} (B_2 B_3)^{\omega_{1,l}} \\
&\mathsf{K}_{2,1} = Y_1^{r_2} (B_2 B_3)^{zf} \\
&\mathsf{K}_{2,2} = (B_2 B_3)^\omega \left(u_1^{ID_{i,1}} \ldots u_j^{ID_{i,j}} X_1 \right)^{r_2} Y_3^{\omega_{2,2}} \\
&\mathsf{E}_{2,j+1} = u_{j+1}^{r_2} (B_2 B_3)^{\omega_{2,j+1}}, \ldots, \mathsf{E}_{2,l} = u_l^{r_2} (B_2 B_3)^{\omega_{2,l}}
\end{aligned}$$

We can show that $B_2 = g_2^\Gamma$, it can be easily seen that the generated secret key is semi-functional because $\gamma = \Gamma f$ and $\gamma z_k = \Gamma \omega$.

For i from $k + 1$ to q, $\mathcal{B}$ runs the key generation algorithm to generate normal secret keys by using $MSK = (X_1, \alpha)$.

Suppose that for the k-th query the identity vector is $\mathbf{ID}_k = (ID_{k,1}, \ldots, ID_{k,j})$. To answer this query, $\mathcal{B}$ sets $z_k = a_1 ID_{k,1} + \cdots + a_j ID_{k,j} + b$, choses $r'_2 \in_R \mathbb{Z}_N$ and, for $t \in \{1, 2\}$, $\omega_{i,2}, \omega_{i,j+1}, \ldots, \omega_{i,l} \in_R \mathbb{N}$ uniformly at random. Then it sets the k-th secret key as follows:

$$\begin{aligned}
&K_{1,1} = T, K_{1,2} = Y_1^\alpha T^{z_k} Y_3^{\omega_{1,2}}, \{E_{1,m} = T^{a_m} Y_3^{\omega_{1,m}}\}_{m=j+1}^{l} \\
&K_{2,1} = T^{r'_2}, K_{2,2} = T^{r'_2 z_k} T^{z_k} Y_3^{\omega_{2,2}} \\
&\{E_{2,m} = T^{r'_2 a_m} Y_3^{\omega_{2,m}}\}_{m=j+1}^{l}
\end{aligned}$$

If $T \in G_{p_1 p_3}$, then we can write it as $Y_1^{r'_1} Y_3^{r_3}$ and the constructed secret key is normal with $r_1 = r'_1$, $r_2 = r'_1 r'_2$. If $T \in G_{p_1 p_2 p_3}$, then we can write it as $Y_1^{r'_1} g_2^{s_2} Y_3^{r_3}$ and the resulting secret key is semi-functional with $r_1 = r'_1$, $r_2 = r'_1 r'_2$, $\gamma = s_2$ and $z = r'_2$.

In the challenge phase, $\mathcal{A}$ outputs the two challenge messages and identity vector sets $(M, \mathbf{V}_t = \{\mathbf{ID}^*_{1t}, \ldots, \mathbf{ID}^*_{Lt}\})$, $t \in \{0, 1\}$. Then, $\mathcal{B}$ flips a random coin $\beta \in_R \{0, 1\}$ and selects $z, z' \in_R \mathbb{Z}_N$ and encrypts M using $\mathbf{V}_\beta$ as follows:

$$\begin{aligned}
&C_0 = M.e(A_1 A_2, Y_1)^\alpha \\
&\forall h = 1, \ldots, L : \left\{ \begin{aligned} &C'_{1,h} = (A_1 A_2)^{a_1 ID_{\beta h,1} + \cdots + a_{i_h} ID_{\beta h,i_h} + b} Y_4^z \\
&x_h = H_{14}(\mathbf{ID}_{ht}), f_h(x) = \prod_{j \neq h} \frac{x - x_j}{x_h - x_j} = \sum_{j=1}^{L-1} a_{hj} x^j, \\
&so \ \ f_h(x_h) = 1 \,\&\, f_h(x_j) = 0 : h \neq j \\
&C_{1,h} = \prod_{j=1}^{L} C'^{a_{jh}}_{1,j} \end{aligned} \right\} \\
&C_1 = (C_{1,1}, \ldots, C_{1,L}), C_2 = A_1 A_2 Y_4^{z'} \\
&\mathsf{cph} := (C_0, C_1, C_2) \qquad (14)
\end{aligned}$$

It can be seen that, $\mathcal{B}$ sets $Y_1^s = A_1$ and $z_1, \ldots, z_L$ are implicitly set at random. Since it is supposed that $\mathbf{ID}_k$ is not congruent to $\mathbf{ID}^*_{h\beta}$ modulo p_2, for all h from 1 to L, then we conclude that z_k and $z_1, \ldots, z_L$ are independent and they have been distributed in randomly. Therefor, it can be seen that if $T \in G_{p_1p_2p_3}$, then $\mathcal{B}$ simulates Game_{k-1} and if $T \in G_{p_1p_2p_3}$, then $\mathcal{B}$ simulates Game_k properly.

4.6 Indistinguishability of Game_q and $\mathsf{Game}_{\mathsf{Final}_0}$

Lemma 5. If there exists a PPT adversary $\mathcal{A}$ such that $Adv^{\mathcal{A}}_{\mathsf{Game}_q} - Adv^{\mathcal{A}}_{\mathsf{Game}_{\mathsf{Final}_0}} = \epsilon$, then there exists a PPT adversary $\mathcal{B}$ whose advantage in breaking Assumption 3 is ϵ.

Proof. $\mathcal{B}$ is given $D := (\sum, g_1, g_2, g_3, g_4, g_1^{\alpha} A_2, g_1^s B_2, g_2^r, A_2^r)$ and T. According to the value of T, $\mathcal{B}$ simulates Game_q or $\mathsf{Game}_{\mathsf{Final}_0}$. We will show that, if $T = e(g_1, g_1)^{\alpha s}$ then it simulates Game_q and if $T \in_R G_T$ it simulates $\mathsf{Game}_{\mathsf{Final}_0}$ for $\mathcal{A}$.

Then $\mathcal{B}$ runs the setup algorithm to generate MSK and PP as follows. It first selects the exponents $\alpha, a_1, \dots, a_l, b, c \in_R \mathbb{Z}_N$ uniformly at random and then sets $Y_1 = g_1$, $Y_3 = g_3$, $Y_4 = g_4, X_4 = Y_4^c$, $X_1 = Y_1^b$ and for i from 1 to l, $u_i = Y_1^{a_i}$. Then, $\mathcal{B}$ sets $PP := (N, Y_1, Y_3, Y_4, t = X_1X_4, u_1, \dots, u_l, \Omega = e(g_1^{\alpha} A_2, Y_1) = e(Y_1, Y_1)^{\alpha})$ as the public parameter and sends it to $\mathcal{A}$. $\mathcal{B}$ also sets $MSK = (X_1, \alpha)$ as the master secret key which is related to PP. $\mathcal{B}$ answers the i-th secret key query for identity vector $(\mathbf{ID}_i = (ID_{i,1}, \dots, ID_{i,j}))$. $\mathcal{B}$ generates semi-functional secret keys to answer all queries by choosing the $r_1, r_2, z, z', \omega \in_R \mathbb{Z}_N$ and, for $t \in \{1, 2\}$, $z_{t,j+1}, \dots, z_{t,l}, \omega_{t,1}, \omega_{t,2}, \omega_{t,j+1}, \dots, \omega_{t,l} \in_R \mathbb{Z}_N$ uniformly at random. Then $\mathcal{B}$ acts as follows:

$$\mathsf{K}_{1,1} = Y_1^{r_1} g_2^{z} Y_3^{\omega_{1,1}}$$
$$\mathsf{K}_{1,2} = (g_1^{\alpha} A_2) g_2^{z'} \left(u_1^{ID_{i,1}} \dots u_j^{ID_{i,j}} X_1\right)^{r_1} Y_3^{\omega_{1,2}}$$
$$\mathsf{E}_{1,j+1} = u_{j+1}^{r_1} g_2^{z_{1,j+1}} Y_3^{\omega_{1,j+1}}, \dots, E_{1,l} = u_l^{r_1} g_2^{z_{1,l}} Y_3^{\omega_{1,l}}$$
$$\mathsf{K}_{2,1} = Y_1^{r_2} (g_2^{r})^{z} Y_3^{\omega_{2,1}}$$
$$\mathsf{K}_{2,2} = A_2^{r} (g_2^{r})^{z'} \left(u_1^{ID_{i,1}} \dots u_j^{ID_{i,j}} X_1\right)^{r_2} Y_3^{\omega_{2,2}}$$
$$\mathsf{E}_{2,j+1} = u_{j+1}^{r_2} g_2^{z_{2,j+1}} Y_3^{\omega_{2,j+1}}, \dots, \mathsf{E}_{2,l} = u_l^{r_2} g_2^{z_{2,l}} Y_3^{\omega_{2,l}}$$

In the challenge phase, $\mathcal{A}$ outputs the two challenge messages and identity vector sets $(M, \mathbf{V}_t = \{\mathbf{ID}^*_{1t}, \dots, \mathbf{ID}^*_{Lt}\})$, $t \in \{0, 1\}$. Then, $\mathcal{B}$ flips a random coin $\beta \in_R \{0, 1\}$ and selects $z, z' \in_R \mathbb{Z}_N$ and encrypts M_{β} using $\mathbf{V}_{\beta}$ as follows:

$$C_0 = M_{\beta}.T$$
$$\forall h = 1, \dots, L : \left\{ \begin{aligned} & C'_{1,h} = (g_1^s B_2)^{a_1 ID_{\beta h,1} + \dots + a_{i_h} ID_{\beta h, i_h} + b} Y_4^{z} \\ & x_h = H_{14}(\mathbf{ID}_{ht}), f_h(x) = \prod_{j \neq h} \frac{x - x_j}{x_h - x_j} = \sum_{j=1}^{L-1} a_{hj} x^j, \\ & so \;\; f_h(x_h) = 1 \,\&\, f_h(x_j) = 0 : h \neq j \\ & C_{1,h} = \prod_{j=1}^{L} C'^{a_{jh}}_{1,j} \end{aligned} \right\}$$
$$C_1 = (C_{1,1}, \dots, C_{1,L}), C_2 = g_1^s B_2 Y_4^{z'}$$
$$\mathsf{cph} := (C_0, C_1, C_2) \tag{15}$$

Since $a_1, \dots, a_l$ are chosen randomly, therefore, their values modulo p_1 and p_2 are random and independent. Therefore the resulting ciphertext implicitly sets $z_1, \dots, z_L$ uniformly at random and the resulting ciphertext is semi-functional. Finally, the proof is finished by observing that, if $T = e(g_1, g_1)^{\alpha s}$, then the resulting semi-functional ciphertext is the encryption of M, and if T be a random element of G_T, then the resulting semi-functional ciphertext is the encryption of a random message which is independent to the challenge message selected by $\mathcal{A}$.

In the end of the game, $\mathcal{A}$ outputs β'. $\mathcal{B}$ checks weather $\beta = \beta'$ or not. If this equality holds, then $\mathcal{B}$ realizes that $T = e(g_1, g_1)^{\alpha s}$; otherwise, $\mathcal{B}$ realizes that T is a random element in G_T. Therefore, the advantage of $\mathcal{B}$ to break Assumption 3 is equal to the difference of $\mathcal{A}$'s advantage in Game_q and $\mathsf{Game}_{\mathsf{Final}_0}$. So we conclude that these two games are indistinguishable.

4.7 Indistinguishability of $\mathsf{Game}_{\mathsf{Final}_0}$ and $\mathsf{Game}_{\mathsf{Final}_1}$

Lemma 6. If there exists a PPT adversary $\mathcal{A}$ such that $Adv^{\mathcal{A}}_{\mathsf{Game}_{\mathsf{Final}_0}} - Adv^{\mathcal{A}}_{\mathsf{Game}_{\mathsf{Final}_1}} = \epsilon$, then there exists a PPT adversary $\mathcal{B}$ whose advantage in breaking Assumption 4 is ϵ.

Proof. First of all, we notice that if there exists an adversary $\mathcal{A}'$ who distinguishes between the encryption of a message using two challenge identity vector sets $\mathbf{V}_0$ and $\mathbf{V}_1$ of its choice, then we can construct $\mathcal{A}$ who distinguishes between the encryption of a message with the identity vector set of its choice and a random identity vector set. In the rest of proof, we suppose that games are been simulating for $\mathcal{A}$.

$\mathcal{B}$ is given $(\sum, g_1, g_2, g_3, g_4, U, U^s A_{24}, U^{\hat{r}}, A_1A_4, A_1^{\hat{r}} A_2, g_1^{\hat{r}} B_2, g_1^s B_{24})$ and T. According to the value of T, $\mathcal{B}$ simulates $\mathsf{Game}_{\mathsf{Final}_1}$ or $\mathsf{Game}_{\mathsf{Final}_0}$. We will show that, if $T = A_1^s D_{24}$ then it simulates $\mathsf{Game}_{\mathsf{Final}_0}$ and if $T \in_R G_{p_1p_2p_4}$ it simulates $\mathsf{Game}_{\mathsf{Final}_1}$ for $\mathcal{A}$.

Then $\mathcal{B}$ runs the setup algorithm to generate MSK and PP as follows. It first selects the exponents $\alpha, a_1, \dots, a_l, b, c \in_R \mathbb{Z}_N$ uniformly at random and then sets $Y_1 = g_1$, $Y_3 = g_3$, $Y_4 = g_4, t = A_1A_4$, and for i from 1 to l, $u_i = U^{a_i}$. Then, $\mathcal{B}$ sets $PP := (N, Y_1, Y_3, Y_4, t, u_1, \dots, u_l, \Omega = e(Y_1, Y_1)^{\alpha})$ as the public parameter and sends it to $\mathcal{A}$. $\mathcal{B}$ answers the i-th secret key query for identity vector $(\mathbf{ID}_i = (ID_{i,1}, \dots, ID_{i,j}))$. $\mathcal{B}$ generates semi-functional secret keys to answer all queries by choosing the $r'_1, r'_2 \in_R \mathbb{Z}_N$ and, for $t \in \{1, 2\}$, $z_{t,j+1}, \dots, z_{t,l}, \omega_{t,1}, \omega_{t,2}, \omega_{t,j+1}, \dots, \omega_{t,l} \in_R \mathbb{Z}_N$ uniformly at random. Then $\mathcal{B}$ acts as follows:

$$\mathsf{K}_{1,1} = (g_1^{\hat{r}} B_2)^{r'_1} Y_3^{\omega_{1,1}}$$
$$\mathsf{K}_{1,2} = Y_1^{\alpha} \left(\left(U^{\hat{r}}\right)^{a_1 ID_{i,1} + \dots + a_j ID_{i,j}} (A_1^{\hat{r}} A_2) \right)^{r'_1} Y_3^{\omega_{1,2}}$$
$$\mathsf{E}_{1,j+1} = (U^{\hat{r}})^{r'_1 a_{j+1}} Y_2^{z_{1,j+1}} Y_3^{\omega_{1,j+1}}$$
$$\dots, E_{1,l} = (U^{\hat{r}})^{r'_1 a_l} Y_2^{z_{1,l}} Y_3^{\omega_{1,l}}$$
$$\mathsf{K}_{2,1} = (g_1^{\hat{r}} B_2)^{r'_2} Y_3^{\omega_{2,1}}$$
$$\mathsf{K}_{2,2} = \left(\left(U^{\hat{r}}\right)^{a_1 ID_{i,1} + \dots + a_j ID_{i,j}} (A_1^{\hat{r}} A_2) \right)^{r'_2} Y_3^{\omega_{2,2}}$$
$$\mathsf{E}_{2,j+1} = (U^{\hat{r}})^{r'_2 a_{j+1}} Y_2^{z_{2,j+1}} Y_3^{\omega_{2,j+1}}$$
$$\dots, E_{2,l} = (U^{\hat{r}})^{r'_2 a_l} Y_2^{z_{2,l}} Y_3^{\omega_{2,l}}$$

It can be shown that in the resulting secret keys $r_1 = \hat{r}r'_1$ and $r_2 = \hat{r}r'_2$. In the challenge phase, $\mathcal{A}$ outputs the challenge message and identity vector sets $(M, \mathbf{V}_t = \{\mathbf{ID}^*_{1t}, \dots, \mathbf{ID}^*_{Lt}\})$, $t \in \{0, 1\}$. Then, $\mathcal{B}$ selects $C_0 \in G_T$ and computes the challenge ciphertext as follows:

$$\begin{aligned}
&C_0\\
&\forall h = 1, \dots, L : \left\{ C'_{1,h} = T(U^s A_{24})^{a_1 ID_{\beta h,1} + \dots + a_{i_h} ID_{\beta h, i_h}} \right.\\
&x_h = H_{14}(\mathbf{ID}_{ht}), f_h(x) = \prod_{j \neq h} \frac{x - x_j}{x_h - x_j} = \sum_{j=1}^{L-1} a_{hj} x^j,\\
&so \;\; f_h(x_h) = 1 \;\&\; f_h(x_j) = 0 : h \neq j\\
&\left. C_{1,h} = \prod_{j=1}^{L} C'^{a_{jh}}_{1,j} \right\}\\
&C_1 = (C_{1,1}, \dots, C_{1,L}), C_2 = g_1^s B_{24}\\
&\mathsf{cph} := (C_0, C_1, C_2)
\end{aligned} \tag{16}$$

If $T = A_1^s D_{24}$, then the adversary generates a semi-functional ciphertext of a random message using the identity vector set $\mathbf{V}_\beta$. If T is a random element of the group, $G_{p_1 p_2 p_4}$, therefore C_1 and C_2 are random elements of $G_{p_1 p_2 p_4}$ and the result ciphertext is semi-functional of a random message using an implicitly randomly chosen identity vector set. Hence, $\mathcal{B}$ is able to distinguish between the two possible states of T by means of $\mathcal{A}$. As we suppose that Assumption 4 holds, we conclude that the games $\mathsf{Game}_{\mathsf{Final}_0}$ and $\mathsf{Game}_{\mathsf{Final}_1}$ are indistinguishable.

4.8 The advantage in $\mathsf{Game}_{\mathsf{Final}_1}$

Theorem 1. If Assumptions 1, 2, 3, and 4, simultaneously hold, then the proposed HIBBE scheme is secure.

Proof. According to the previous lemmata, we have proved that if the mentioned assumptions hold, then the real security game is indistinguishable from $\mathsf{Game}_{\mathsf{Final}_1}$. The advantage of adversary in $\mathsf{Game}_{\mathsf{Final}_1}$ is zero because the value of β is information-theoretically hidden from attacker. Therefor, we conclude that the advantage of attacker in breaking the Anonymous HIBBE is a negligible function.

5 Performance evaluation of the proposed anonymous HIBBE

In this section, we will present the performance analysis over the proposed anonymous HIBBE and HIB-MDVS schemes. To this end, the computation, communication and storage overheads are computed. We denote the modular multiplication, exponentiation over $\mathbb{G}_N$ and $\mathbb{G}_T$, and pairing by Mul_N, exp_N, Mul_T, exp_T and Pair_N respectively.

Without loss of generality, for simplicity to analyze the performance of the proposed scheme, we assume that all identities in the set $\mathbf{V}$ has the same depth.

As the first scenario of key generation algorithm can be done in an off-line procedure, we just compute the computation overhead of delegation phase. After that, we compute the computation overheads of the encryption and decryption algorithms in terms of hierarchy depth of $\mathbf{V}$, i.e. $d_m = ||\mathbf{V}||$ and the number of intended receivers, i.e, L. Then , we compute the communication and storage overheads.

5.1 The delegation phase

As mentioned before, the maximum levels of hierarchy is l. Suppose that, the user in the $i-1$-th level generates the secret keys of its subordinates in the i-th level. According to the delegation phase of the key generation algorithm, the computation overhead of the delegation phase in term of l and i is equal to $(7 + 2l - 2i)exp_N + (9 + 2l - 2i)Mul_N$ (Table 4).

5.2 Encryption algorithm

As computing the coefficients of $f_h(x)$ are independent of computation of $C'_{1,h}$, for each identity vector, we can compute the coefficients of the functions corresponding to each identity vector separately and store them. Therefore without computing $C'_{1,h}$ we can compute $C_{1,h}$ by using the stored coefficients as follows:

$$\begin{aligned}
C'_{1,h} &= \left(u_1{}^{ID_{h,1}} \dots u_{d_m}{}^{ID_{h,d_m}} t\right)^s Z\\
C_{1,h} &= \prod_{j=1}^{L} C'_{1,h}{}^{a_{jh}}\\
&= C'_{1,1}{}^{a_{1h}} \times \dots \times C'_{1,L}{}^{a_{Lh}}\\
&= \left(\left(u_1{}^{ID_{1,1}} \dots u_{d_m}{}^{ID_{1,d_m}} t\right)^s Z\right)^{a_{1h}} \times \dots\\
&\dots \times \left(\left(u_1{}^{ID_{L,1}} \dots u_{d_m}{}^{ID_{L,d_m}} t\right)^s Z\right)^{a_{Lh}}\\
&= \left(u_1{}^{s(\sum_{j=1}^{L} ID_{j,1} a_{jh})}\right) \times \dots\\
&\dots \times \left(u_{d_m}{}^{s(\sum_{j=1}^{L} ID_{j,d_m} a_{jh})}\right) \times t^{s(\sum_{j=1}^{L} a_{jh})} \times Z^{\sum_{j=1}^{L} a_{jh}}
\end{aligned}$$

In this case, we can see that for computing each $C_{1,h}$ we need $d_m + 2$ exponentiations and $d_m + 1$ multiplications. In addition, to compute the whole ciphertext we need one pairing, one multiplication, and one exponentiation. Consequently, the total computation overhead of the encryption algorithms is $exp_T + (L(d_m + 2) + 1)exp_N + (L(d_m + 1) + 1)Mul_N$

5.3 Decryption algorithm

For decrypting the message, L exponentiations and $L - 1$ multiplication is required to compute C^*. In addition, 2 paring computations and multiplications in $\mathbb{G}_T$ is required to extract the message from the ciphertext. Therefore, the computation overhead of decryption part is $2\mathsf{Pair}_N + Lexp_N + (L-1)Mul_N + 2Mul_T$.

5.4 Communication overhead:

In our proposed scheme the communication overhead is equal to the ciphertext size. To send the encrypted message, we need $\log_2(|N|)$, $L(\log_2(|p_1 p_4|))$, and $\log_2(|p_1 p_4|)$ bits because of C_0, C_1, and C_2 respectively. So, the total communication overhead is $(L + 1)\log_2(|p_1 p_4|) + \log_2(|N|)$.

5.5 Storage overhead:

The storage overhead of each user is mainly related to the size of the secret key. The size of the secret key in the ith level of hierarchy is $(2(l - i) + 4)\log_2(|p_1 p_3|)$ where l is the total level of hierarchy in the network.

TABLE 2
Basic information, The information of the using platform and type of the composite order bilinear maps [22].

Ellipse curve type	Type A1
Ellipse curve	$y^2 = x^3 + x$
Symmetry or not	Symmetry
Order	$N = p_1 p_2 p_3 p_4$
Security level	$\log p_i = 192$
Platform	Personal computer
CPU series	Intel core i5-2400
RAM	4GB
Operate system	Windows 8
JDK version	JDK 1.6
jPBC version	2.0.0

TABLE 3
Benchmark. The time execution of the basic algorithms [22].

Operation	Time (ms)
Pairing: Pair_N	90
Element exponentiation in G_N: exp_N	105
Element exponentiation in G_T: exp_T	9
Element multiplication in G_N: Mul_N	0.1
Element multiplication in G_T: Mul_T	0.01

5.6 Implementation

The execution time of the algorithms Pair_N, exp_N, exp_T, Mul_N, and Mul_T is measured by calling the modules of jPBC library [21] in Java. As the proposed anonymous HIBBE scheme is designed based on composite order bilinear pairing, jPBC library is used which introduces interfaces for implementing composite order bilinear maps. In [22], the time executions of the mentioned algorithms is computed and we illustrate these results in Table 3. The information related to the platform which is used in [22] and their implementations are illustrated in Table 2. According to Table 3, the execution time of running encryption and decryption algorithms of our proposed HIBBE scheme when $d_m = 3$ for $L = \{5, 10, 15, 20, 25\}$ is illustrated in Table 5.

6 Hierarchical Identity-Based MDVS (HIB-MDVS)

In this section, for the first time, we will introduce the concept of HIB-MDVS where the generation of users' secret keys is in a hierarchical setting. Therefore, the users higher in the hierarchy can generate a secret key for the children of their domain. We introduce the notion of HIB-MDVS and define its unforgeability in a formal way.

A HIB-MDVS scheme consists of four polynomial-time algorithms: Setup, Extract, MDVSig and MDVSVrfy. These four algorithms are as follows:

- $(PP, MSK) \leftarrow \mathsf{Setp}(\lambda)$: This algorithm takes λ as input to generates PP and MSK.
- $USK_{\mathbf{ID}} \leftarrow \mathsf{Extract}(USK_{Par(\mathbf{ID})}, \mathbf{ID})$: This algorithm takes **ID** and its parent secret key $USK_{Par(\mathbf{ID})}$ as inputs to generate $USK_{\mathbf{ID}}$. As mentioned before, $MSK = USK_0$ is the secret key of the TTP.
- $\sigma \leftarrow \mathsf{MDVSig}(PP, USK_{\mathbf{ID}_s}, M, \mathbf{V})$: In this algorithm, the signer with the identity vector $\mathbf{ID}_s$, generates the signature σ for $M \in \mathcal{M}$ and only the users whose identity vectors are in **V** can verify it.
- $\{0, 1\} := \mathsf{MDVSVrfy}(PP, \mathbf{ID}_s, USK_{\mathbf{ID}}, \sigma)$: The user with identity vector **ID** runs this deterministic algorithm. This algorithm takes PP, $\mathbf{ID}_s$, σ, and the user secret key $USK_{\mathbf{ID}}$ as the inputs and tests whether σ is a valid signature. If $\mathbf{ID} \in Par(\mathbf{V})$, then this user can check the validity of σ. This algorithm returns 1 if, (1) $\mathbf{ID} \in Par(\mathbf{V})$ and (2) signature σ is a valid signature; otherwise, returns 0.

6.1 Security model of HIB-MDVS

In the following, we will define existential forgery against adaptively chosen message attack (EF-CMA) for HIB-MDVS according to the following game which is held between $\mathcal{C}$ and $\mathcal{A}$.

Setup: $\mathcal{C}$ runs the algorithm $(MSK, PP) \leftarrow \mathsf{Setup}(\lambda)$ to generate PP and MSK. $\mathcal{C}$ selects $\mathbf{ID}_s$ of the signer and generates the secret key $SK_{\mathbf{ID}_s}$. Then it sends $\mathbf{ID}_s$, PP and λ to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ is allowed to query adaptively to the following oracles for polynomially many times. $\mathcal{C}$ keeps the messages list L_M which is initially empty.

- $USK_{\mathbf{ID}} \leftarrow \mathcal{O}_{\mathsf{Extract}}(\mathbf{ID})$: This oracle gets the identity vector **ID** and generates the secret key $USK_{\mathbf{ID}} \leftarrow \mathsf{Extract}(USK_{Par(\mathbf{ID})}, \mathbf{ID})$ and sends it to $\mathcal{A}$. Note that if $\mathbf{ID} \in Par(\mathbf{ID}_s)$, this oracle stops to answer.
- $\sigma_i \leftarrow \mathcal{O}_{\mathsf{Sign}}(M_i, \mathbf{V}_i)$: This oracle receives M_i from $\mathcal{A}$ and computes the valid signature $\sigma_i \leftarrow \mathsf{MDVSig}(PP, USK_{\mathbf{ID}_s}, M_i, \mathbf{V}_i)$ and sends it to $\mathcal{A}$. This oracle also adds M_i to L_M for each query.

Guess: $\mathcal{A}$ outputs the tuple $(\sigma_F, M_F, \mathbf{V}_F)$ where M_F was never queried before ($M_F \notin L_M$). $\mathcal{A}$ wins the game if σ_F is a valid HIB-MDVS signature. The advantage of $\mathcal{A}$ to win this game is defined according to Equation (17).

$$\begin{aligned} Adv^{EF-CMA}_{HIB-MDVS,\mathcal{A}} &= \Pr[\mathcal{A}(\lambda) = (M_F, \sigma_F) : M_F \notin L_M \\ &\& \forall \mathbf{ID} \in \mathbf{V}_F : 1 := \mathsf{MDVSVrfy}(PP, \mathbf{ID}_s, USK_{\mathbf{ID}}, \sigma_F)] \end{aligned} \tag{17}$$

Definition 6 (Security of HIB-MDVS against EF-CMA). The HIB-MDVS scheme is unforgeable if the advantage of any PPT adversary $\mathcal{A}$ to win in the EF-CMA game is a negligible function as follows:

$$Adv^{EF-CMA}_{HIB-MDVS,\mathcal{A}} \leq \mathsf{negl}(\lambda) \tag{18}$$

7 Generic construction of HIB-MDVS

This section presents a generic construction of HIB-MDVS based on anonymous HIBBE and an unforgeable hierarchical identity-based signature (HIBS) [17] as the building blocks.

The basic idea of the proposed scheme is that the signer encrypts a random key like k by the HIBBE for the legitimate verifiers. Then the signer signs the hashed value of the concatenation of message and the key k, i.e., $H(k||m)$ and broadcast the ciphertext of the key k, the resulting signature and the message. For the verification process, the legitimate user decrypt the ciphertext part of the receiving signature and then runs the verification algorithm of the mentioned signature on the message $m' = H(k||m)$.

We not that the most important advantage of using anonymous HIBBE to construct a HIB-MDVS signature is that we can

TABLE 4
The complexity analysis of the proposed anonymous HIBBE

Criteria		Our proposed scheme
Computation Overhead	**Delegation Phase**	$(7+2l-2i)exp_N + (9+2l-2i)Mul_N$
	BroadEnc	$exp_T + (L(d_m+2)+1)exp_N + (L(d_m+1)+1)Mul_N$
	BroadDec	$2\mathsf{Pair}_N + Lexp_N + (L-1)Mul_N + 2Mul_T$
Communication Overhead		$(L+1)\log_2(\vert p_1p_4\vert) + \log_2(\vert N\vert)$
Storage Overhead		$(2(l-i)+4)\log_2(\vert p_1p_3\vert)$

TABLE 5
The performance of our proposed anonymous HIBBE, for $d_m = 3$ and different values of the number of intended receiver, L.

Algorithm	$L = 5$(s)	$L = 10$(s)	$L = 15$(s)	$L = 20$(s)	$L = 25$(s)
BroadEnc	2.741	4.369	7.995	10.622	13.249
BroadDec	0.705	1.231	1.756	2.282	2.807

provide the verifiers privacy by hiding the information related to their identity. Therefore, it is computationally hard for an adversary to infer some information about the identities of the verifiers. We will discuss about the anonymity of HIB-MDVS in Remark 1.

7.1 The proposed generic construction

Suppose that the algorithm Π_1 : $(\mathsf{Setup}_1, \mathsf{Extract}_1, \mathsf{BroadEnc}, \mathsf{BroadDec})$ is a HIBBE scheme and the algorithm Π_2 : $(\mathsf{Setup}_2, \mathsf{Extract}_1, \mathsf{Sign}, \mathsf{Vrfy})$ ([17]) is an unforgeable HIBS scheme. The details of the scheme is presented as follows:

Setup $((\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(\lambda))$: To compute the public parameters pp and the master secret key msk this algorithms runs $(\mathsf{pp}_1, \mathsf{msk}_1) \leftarrow \mathsf{Setup}_1(\lambda)$ and $(\mathsf{pp}_2, \mathsf{msk}_2) \leftarrow \mathsf{Setup}_2(\lambda)$ and sets the pair of public parameter and master secret key as $(\mathsf{pp}, \mathsf{msk}) := \Big((\mathsf{pp}_1, \mathsf{pp}_2), (\mathsf{msk}_1, \mathsf{msk}_2)\Big)$.

Key generation $(\mathsf{sk}_{\mathbf{ID}} \leftarrow \mathsf{Extract}(\mathsf{sk}_{par(\mathbf{ID})}, \mathbf{ID}))$: The identity vector $\mathbf{ID} = (ID_1, \ldots, ID_i)$ is given to the extraction algorithm and the secret key of the parent of the identity vector $\mathbf{ID}$ is $\mathsf{sk}_{\mathbf{ID}} = (\mathsf{sk}_{1,par(\mathbf{ID})}, \mathsf{sk}_{2,Par(\mathbf{ID})})$ where the first part is related to the HIBBE and the second part is related to the HIBS. This algorithm runs the extraction algorithms $\mathsf{sk}_{1,\mathbf{ID}} \leftarrow \mathsf{Extract}_1(\mathsf{sk}_{1,Par(\mathbf{ID})}, \mathbf{ID})$ and $\mathsf{sk}_{2,\mathbf{ID}} \leftarrow \mathsf{Extract}_2(\mathsf{sk}_{2,par(\mathbf{ID})}, \mathbf{ID})$ and sets the secret key associated to the identity vector $\mathbf{ID}$ equal to $\mathsf{sk}_{\mathbf{ID}} := (\mathsf{sk}_{1,\mathbf{ID}}, \mathsf{sk}_{2,\mathbf{ID}})$.

Sign $(\sigma \leftarrow \mathsf{MDVSig}(\mathsf{pp}, sk_{\mathbf{ID}_s}, m, \mathbf{V}))$: The identity vector set $\mathbf{V} = \{\mathbf{ID}_1, \ldots, \mathbf{ID}_L\}$ is given. The signer first chooses the random string $k \in_R \mathcal{M}$ and runs the encryption algorithm $C_k \leftarrow \mathsf{BroadEnc}(\mathsf{pp}_1, k, \mathbf{V})$. Then it compute the hashed value of message and the random string k, i.e., $h = H(k||m)$ and signs h by running the sign algorithm $\sigma' \leftarrow \mathsf{Sign}(h, \mathsf{sk}_{\mathbf{ID}_s})$. The resulting signature is $\sigma := (m, C_k, \sigma')$.

Verify $(\{0,1\} := \mathsf{MDVSVrfy}(\mathsf{pp}, \mathbf{ID}_s, \mathsf{sk}_{\mathbf{ID}}, \sigma))$: The verifier with the identity vector $\mathbf{ID}$ decrypts the ciphertext C_k by running the algorithm $k' := \mathsf{BroadDec}(C_k, \mathsf{sk}_{1,\mathbf{ID}}, \mathsf{pp}_1)$ where the secret key is $\mathsf{sk}_{\mathbf{ID}} = (\mathsf{sk}_{1,\mathbf{ID}}, \mathsf{sk}_{2,\mathbf{ID}})$. If $\mathbf{ID} \in \mathbf{V}$, then $k' = k$. After that, it computes $h' = H(k'||m)$ and runs the verification algorithm of HIBS, i.e., $\{0,1\} := \mathsf{Vrfy}(\sigma', h', \mathbf{ID}_s, \mathsf{pp}_2)$.

8 Security of the proposed HIB-MDVS

In this section we review the security of HIBS against existential forgery for selective ID, adaptive chosen-message-and-identity attack (EF-sID-CMIA) [17], discuss about the anonymity of the proposed HIB-MDVS scheme in Remark 1, and then prove that the proposed generic construction of HIB-MDVS is secure against the EF-CMA according to Theorem 2.

8.1 Security of HIBS against EF-sID-CMIA

The formal definition of EF-sID-CMIA is presented as follows [17]:

Init: $\mathcal{A}$ outputs the challenge identity vector $\mathbf{ID}^*$ and sends it to $\mathcal{C}$.

Setup: $\mathcal{C}$ runs $(\mathsf{pp}_2, \mathsf{msk}_2) \leftarrow \mathsf{Setup}_2(\lambda)$ and sends pp_2 to $\mathcal{A}$.

Attack: $\mathcal{A}$ can query the following oracles:

- **Extract:** $\mathcal{A}$ chooses $\mathbf{ID}$ and sends it to $\mathcal{C}$. If $\mathbf{ID} \notin Par(\mathbf{ID}^*)$ then $\mathcal{C}$ returns $\mathsf{sk} \leftarrow \mathsf{Extract}_2(\mathsf{msk}_2, \mathbf{ID})$ to $\mathcal{A}$.
- **Sign:** $\mathcal{A}$ sends $(\mathbf{ID}, m)$ to $\mathcal{C}$ and receives the valid signature $\sigma \leftarrow \mathsf{Sign}(m, \mathsf{sk}_{\mathbf{ID}})$ form $\mathcal{C}$.

Forgery: $\mathcal{A}$ outputs the tuple (σ^*, m^*). $\mathcal{A}$ wins the game if $1 := \mathsf{Vrfy}(\sigma^*, m^*, \mathbf{ID}^*, \mathsf{pp}_2)$.

If the probability $\Pr[\mathcal{A}(\lambda) = (\sigma^*, m^*) : 1 := \mathsf{Vrfy}(\sigma^*, m^*, \mathbf{ID}^*, \mathsf{pp}_2)]$ is a negligible function in terms of λ, then HIBS is secure against EF-sID-CMIA.

8.2 Security proof

Remark 1 (Anonymity of the proposed HIB-MDVS scheme). As we use an anonymous HIBBE scheme, we can conclude that all the information related to the verifiers' identities are hidden. We have formally proved that the output ciphertext of encryption algorithm of HIBBE does not leakage any information about the receivers' identities. As a result, the identities of the verifiers of the proposed signature sty anonymous. Note that, also the authorized receivers can not infer any information about the other intended verifiers. This guaranties the fully anonymity of the proposed HIB-MDVS scheme.

Theorem 2 (unforgeability of HIB-MDVS). If the using HIBS scheme in the proposed generic construction of HIB-MDVS is secure against EF-sID-CMIA, then the resulting HIB-MDVS scheme is secure against the EF-CMA.

We suppose that the HIB-MDVS scheme is not secure and $\mathcal{A}$ can win in the EF-CMA game with non-negligible advantage $\epsilon(\lambda)$. Then we design the adversary $\mathcal{B}$ who wins the EF-sID-CMIA game with non- negligible probability.

Setup: $\mathcal{A}$ selects $\mathbf{ID}_s$ and sends it to $\mathcal{B}$. $\mathcal{B}$ sets $\mathbf{ID}^* = \mathbf{ID}_s$ and sends $\mathbf{ID}^*$ to $\mathcal{C}$. Then $\mathcal{C}$ runs $(\mathsf{pp}_2, \mathsf{msk}_2) \leftarrow \mathsf{Setup}_2(\lambda)$ and sends pp_2 to $\mathcal{B}$. $\mathcal{B}$ runs $(\mathsf{pp}_1, \mathsf{msk}_1) \leftarrow \mathsf{Setup}_1(\lambda)$ and sends $\mathsf{pp} := (\mathsf{pp}_1, \mathsf{pp}_2)$ to $\mathcal{A}$.

Phase 1: $\mathcal{B}$ simulates Phase 1 for $\mathcal{A}$ by responding to its queries.

- $\mathcal{O}_{\mathsf{Extract}}(\mathbf{ID})$: $\mathcal{B}$ receives $\mathbf{ID}$ and sends it to $\mathcal{C}$. $\mathcal{C}$ returns $\mathcal{B}$ the secret key $\mathsf{sk}_2 \leftarrow \mathsf{Extract}_2(\mathsf{msk}_2, \mathbf{ID})$. Then $\mathcal{B}$ runs $\mathsf{sk}_1 \leftarrow \mathsf{Extract}_1(\mathsf{msk}_1, \mathbf{ID})$ and returns $\mathcal{A}$ the resulting secret key, i.e., $\mathsf{sk} := (\mathsf{sk}_1, \mathsf{sk}_2)$.
- $\mathcal{O}_{\mathsf{Sign}}(m_i, \mathbf{V}_i)$: $\mathcal{B}$ receives the tuple $(m_i, \mathbf{V}_i)$ from $\mathcal{A}$. $\mathcal{B}$ selects $k_i \in_R \mathcal{M}$ and computes $C_{k_i} \leftarrow \mathsf{BroadEnc}(\mathsf{pp}_1, k_i, \mathbf{V}_i)$ and $h_i = H(k_i || m_i)$. Then it sends $(h_i, \mathbf{ID}^*)$ to $\mathcal{C}$ and receives $\sigma'_i \leftarrow \mathsf{Sign}(h_i, \mathsf{sk}_{2,\mathbf{ID}^*})$. Then $\mathcal{B}$ sets $\sigma_i := (m_i, C_{k_i}, \sigma'_i)$ and sends it to $\mathcal{A}$.

Guess: $\mathcal{A}$ outputs the tuple $(\sigma_F = (m_F, C_{k_F}, \sigma'_F), m_F, \mathbf{V}_F)$ and $\mathcal{B}$ receives it. Then $\mathcal{B}$ selects an arbitrary $\mathbf{ID}_F \in \mathbf{V}_F$ and computes the secret key $\mathsf{sk}_{1,\mathbf{ID}_F} \leftarrow \mathsf{Extract}_1(\mathsf{msk}_1, \mathbf{ID}_F)$. Then it computes $k_F := \mathsf{BroadDec}(C_{k_F}, \mathsf{sk}_{1,\mathbf{ID}_F}, \mathsf{pp}_1)$. Then $\mathcal{B}$ computes $h_F = H(k_F || m_F)$ and then sends the tuple $(\sigma^* := \sigma'_F, m^* = h_F)$ to $\mathcal{C}$ as a forgery for the HIBS scheme.

So the probability that $\mathcal{B}$ wins in the EF-sID-CMIA security game is the same as the advantage of $\mathcal{A}$ to win in the EF-CMA game, i.e., $\epsilon(\lambda)$. As is a non-negligible function, we conclude that $\mathcal{B}$ wins with a non-negligible probability. This contradicts with the assumption that the using HIBS scheme is secure against EF-sID-CMIA. So the proposed generic construction of HIB-MDVS is secure against EF-CMA.

9 Complexity analysis of the proposed generic construction of HIB-MDVS

In this section, we compute the asymptotic complexity of the resulting HIB-MDVS which is constructed by replacing our proposed HIBBE and the HIBS (presented in [17]) in our generic construction of HIB-MDVS. Table 6 illustrates the asymptotic complexity of the prosed concrete construction of HIB-MDVS. Note that the proposed HIBS scheme in [17] is designed based on prime order bilinear pairing. We denote $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ as the bilinear pairing with prime order p. Therefore, in Table 6, Mul_p, exp_p respectively denote the multiplication and exponentiation over the cyclic group $\mathbb{G}_1$ of prime order p and Pair_p denotes the pairing operation over $\mathbb{G}_2$.

According to Table 6 of [23], the time execution of the required operation in the prime order bilinear maps are much lower than the composite order. Therefore, the execution time of the HIB-MDVS's algorithms can be computed regardless of the prime order operations. So, to compute the execution time, we can use the results of Table 3.

10 Conclusion

In this paper, a new cryptographic notion which is called HIB-MDVS was introduced and its security definition was defined based on EF-CMA security game and a generic construction of HIB-MDVS was proposed based on HIBS and HIBBE schemes. In addition, a formal definition of HIBBE, i.e., anonymity against chosen identity vector set and chosen message attack, was presented and a concrete construction of anonymous HIBBE schemes was suggested. The proposed construction is designed based on composite order bilinear maps. We proved that the anonymity of the proposed scheme is reduced to the three assumptions which are assumed to be hard in the composite order bilinear maps. Performance evaluation shows that our proposed anonymous HIBBE scheme is practical.

References

[1] A. Fiat and M. Naor, "Broadcast encryption," in *Annual International Cryptology Conference.* Springer, 1993, pp. 480–491.

[2] R. Sakai and J. Furukawa, "Identity-based broadcast encryption." *IACR Cryptology ePrint Archive*, vol. 2007, p. 217, 2007.

[3] X. Zhao, "Amendment to trace and revoke systems with short ciphertexts." *International Journal of Network Security*, vol. 14, no. 5, pp. 251–256, 2012.

[4] B. Malek and A. Miri, "Adaptively secure broadcast encryption with short ciphertexts." *International Journal of Network Security*, vol. 14, no. 2, pp. 71–79, 2012.

[5] S. G. Akl and P. D. Taylor, "Cryptographic solution to a problem of access control in a hierarchy," *ACM Transactions on Computer Systems (TOCS)*, vol. 1, no. 3, pp. 239–248, 1983.

[6] M. J. Atallah, M. Blanton, N. Fazio, and K. B. Frikken, "Dynamic and efficient key management for access hierarchies," *ACM Transactions on Information and System Security (TISSEC)*, vol. 12, no. 3, p. 18, 2009.

[7] A. Castiglione, A. D. Santis, and B. Masucci, "Key indistinguishability vs.strong key indistinguishability for hierarchical key assignment schemes," *IEEE Transactions on Dependable and Secure Computing*, vol. PP, no. 99, pp. 1–1, 2015.

[8] M. Cafaro, R. Civino, and B. Masucci, "On the equivalence of two security notions for hierarchical key assignment schemes in the unconditional setting," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 4, pp. 485–490, 2015.

[9] E. Bertino, N. Shang, and S. S. Wagstaff Jr, "An efficient time-bound hierarchical key management scheme for secure broadcasting," *IEEE transactions on dependable and secure computing*, vol. 5, no. 2, pp. 65–70, 2008.

[10] A. Castiglione, A. De Santis, B. Masucci, F. Palmieri, A. Castiglione, J. Li, and X. Huang, "Hierarchical and shared access control," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 4, pp. 850–865, 2016.

[11] W. Liu, J. Liu, Q. Wu, and B. Qin, "Hierarchical identity-based broadcast encryption," in *Information Security and Privacy - 19th Australasian Conference, ACISP 2014, Wollongong, NSW, Australia, July 7-9, 2014. Proceedings.* Springer, 2014, pp. 242–257.

[12] W. Liu, J. Liu, Q. Wu, B. Qin, and Y. Li, "Practical chosen-ciphertext secure hierarchical identity-based broadcast encryption," *International Journal of Information Security*, pp. 1–16, 2015.

[13] Y. Ren, Z. Niu, and X. Zhang, "Fully anonymous identity-based broadcast encryption without random oracles," *International Journal of Network Security*, vol. 16, no. 4, pp. 256–264, 2014.

[14] J. Zhou, X. Lin, X. Dong, and Z. Cao, "Psmpa: Patient self-controllable and multi-level privacy-preserving cooperative authentication in distributedm-healthcare cloud computing system," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 6, pp. 1693–1703, 2015.

[15] S. S. M. Chow, "Identity-based strong multi-designated verifiers signatures," in *Public Key Infrastructure, Third European PKI Workshop: Theory and Practice, EuroPKI 2006, Turin, Italy, June 19-20, 2006, Proceedings.* Springer, 2006, pp. 257–259.

[16] A. D. Caro, V. Iovino, and G. Persiano, "Fully secure anonymous HIBE and secret-key anonymous IBE with short ciphertexts," in *Pairing-Based Cryptography - Pairing 2010 - 4th International Conference, Yamanaka Hot Spring, Japan, December 2010. Proceedings*, 2010, pp. 347–366.

[17] S. S. Chow, L. C. Hui, S. M. Yiu, and K. Chow, "Secure hierarchical identity based signature and its application," in *Information and Communications Security.* Springer, 2004, pp. 480–494.

[18] D. Boneh, E. Goh, and K. Nissim, "Evaluating 2-dnf formulas on ciphertexts," in *Theory of Cryptography, Second Theory of Cryptography Conference, TCC 2005, Cambridge, MA, USA, February 10-12, 2005, Proceedings*, 2005, pp. 325–341.

[19] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques.* Springer, 2010, pp. 62–91.

TABLE 6
The complexity analysis of the proposed concrete construction of HIB-MDVS. In this table, k denotes the level of hierarchy of the signer ($||\mathbf{ID}_s|| = k$). We note that the storage overhead is considered as the secret key size of the user in the i-th level of hierarchy. To achieve the required security level we should select $\log_2 p_i = \log_2 p$.

Criteria		Our proposed scheme
Computation Overhead	**Extraction**	$\big((7+2l-2i)exp_N + (9+2l-2i)Mul_N\big) + \big(2Mul_p + 3exp_p\big)$
	Sign	$\big(exp_T + (L(d_m+2)+1)exp_N + (L(d_m+1)+1)Mul_N\big) + \big((i+2)exp_p\big)$
	Verify	$\big(2\mathsf{Pair}_N + Lexp_N + (L-1)Mul_N + 2Mul_T\big) + \big((k+1)Mul_p + (k+1)exp_p + k\mathsf{Pair}_p\big)$
Communication Overhead		$(L+1)\log_2(p_1p_4) + \log_2(N) + (k+2)\log_2 p$
Storage Overhead		$(2(l-i)+4)\log_2(p_1p_3) + i\log_2 p$

[20] M. H. Ameri, M. R. Assar, J. Mohajeri, and M. Salmasizadeh, "A generic construction for verifiable attribute-based keyword search schemes," 2015. [Online]. Available: http://eprint.iacr.org/2015/915

[21] A. De Caro and V. Iovino, "jpbc: Java pairing based cryptography," in *Computers and Communications (ISCC), 2011 IEEE Symposium on.* IEEE, 2011, pp. 850–855.

[22] C. Hu, R. Yang, P. Liu, Z. Yu, Y. Zhou, and Q. Xu, "Public-key encryption with keyword search secure against continual memory attacks," *Security and Communication Networks*, 2016.

[23] A. Guillevic, "Comparing the pairing efficiency over composite-order and prime-order elliptic curves," in *International Conference on Applied Cryptography and Network Security.* Springer, 2013, pp. 357–372.