symmetry

MDPI

Article

Efficient Hierarchical Identity-Based Encryption System for Internet of Things Infrastructure

Lifeng Guo [1], Jing Wang [1] and Wei-Chuen Yau [2,*]

[1] School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China
[2] School of Electrical and Computer Engineering, Xiamen University Malaysia, Jalan Sunsuria, Bandar Sunsuria, Sepang 43900, Selangor Darul Ehsan, Malaysia
* Correspondence: wcyau@xmu.edu.my; Tel.: +60-3-8800-6803

Received: 19 June 2019; Accepted: 8 July 2019; Published: 13 July 2019



Abstract: Security is a main concern for the Internet of Things (IoT) infrastructure as large volumes of data are collected and processed in the systems. Due to the limited resources of interconnected sensors and devices in the IoT systems, efficiency is one of the key considerations when deploying security solutions (e.g., symmetric/asymmetric encryption, authentication, etc.) in IoT. In this paper, we present an efficient Hierarchical Identity-Based Encryption (HIBE) system with short parameters for protecting data confidentiality in distributed IoT infrastructure. Our proposed HIBE system has the public parameters, private key, and ciphertext, each consisting of a constant number of group elements. We prove the full security of the HIBE system in the standard model using the dual system encryption technique. We also implement the proposed scheme and compare the performance with the original Lewko–Waters HIBE. To the best of our knowledge, our construction is the first HIBE system that achieves both full security in the standard model and short parameters in terms of the public parameters, private key, and ciphertext.

Keywords: cryptography; data security; Internet of Things; hierarchical identity-based encryption; dual system; standard model

1. Introduction

Encryption is a security solution used for data confidentiality protection. There are two types of encryption systems, namely symmetric and asymmetric. A symmetric encryption system uses the same key for encryption and decryption. In an asymmetric encryption (also known as public-key encryption) system, a public key of the data recipient is used for encryption, and the private key is used for decryption. In 1984, Shamir [1] proposed the idea of identity-based cryptography where the identity (ID) of a user is used as a public key, and a third party, called the Private Key Generator (PKG), is responsible for generating a private key for the user. This approach simplifies the problem of managing digital certificates in traditional public key systems. It was only in 2001 that Boneh and Franklin [2] constructed the first Identity-Based Encryption (IBE) scheme from the Weil pairing with chosen ciphertext security in the random oracle model. Many IBE schemes [3–5] were then proposed and proven secure in the standard model. Horwitz and Lynn [6] introduced the notion of Hierarchical Identity-Based Encryption (HIBE), which gives more flexibility for users to share and manage sensitive data. In an HIBE scheme, private keys at a lower level of PKG can be issued by a higher level of PKG. This reduces the burden for a single PKG setting of IBE.

One of the main design issues for the HIBE system is to obtain an efficient scheme with short parameters (e.g., public parameters, private key, ciphertext). It is desirable to design an HIBE scheme with parameters that are independent of the maximum hierarchy depth or the depth of user identity. Otherwise, it is not practical to be implemented in the application due to high computation and

communication cost. In terms of security, it is desirable to have an HIBE system with full security as compared to selective-identity security. Many previous HIBE schemes have been designed in the direction of achieving the above properties. However, most previous realizations of HIBE systems [7] only provided the above features partially, i.e., achieving constant-size ciphertext (resp. private key), but the private key (resp. ciphertext) still depends on the hierarchy depth.

1.1. Applications

The hierarchical structure and the key delegation property of HIBE make it an ideal security solution for protecting data confidentiality in the emerging computing and network environments, such as cloud storage, smart home systems, mobile networks, and the Internet of Things (IoT) [8–12]. The advantage of an HIBE system is that it provides the scalability of IBE system in a distributed network environment by reducing the workload of a root PKG. This can be achieved by delegating lower-level sub-PKGs for key generation.

Consider the applications in IoT where many sensors and devices are interconnected. HIBE can be deployed in the IoT system such as the unit IoT and Ubiquitous IoT (U2IoT) architecture [13,14] to support secure data storage and exchange. In the U2IoT architecture, multiple interconnected sensors or devices form a unit IoT. A local IoT and industrial IoT comprise multiple unit IoT within a region or an industry, respectively. The local IoT and industrial IoT in a country are integrated by a national IoT to form the ubiquitous IoT. Data collected and exchanged by each IoT system in this architecture are managed by a Unit Data Center (UDC), Local Data Center (LDC), Industry Data Center (IDC), and National Data Center (NDC), respectively.

The root PKG of HIBE can be deployed in NDC, which delegates LDC and IDC as the sub-PKGs. The delegations follow the paths in the hierarchical structure as shown in Figure 1, where the lowest level is formed by the sensors. The NDC is responsible for generating the public parameters of the whole system. It also generates the private keys used by LDC and IDC, respectively. Subsequently, the task of key generation can be delegated to LDC and IDC for generating private keys of the UDC under their management. The private keys used by the sensors at the lowest level can be obtained from their respective UDC. Due to the constraint of the computation capability and power supply of the sensors in the IoT system, it is desirable to implement an efficient and secure HIBE system with short public parameters, private key, or ciphertext.

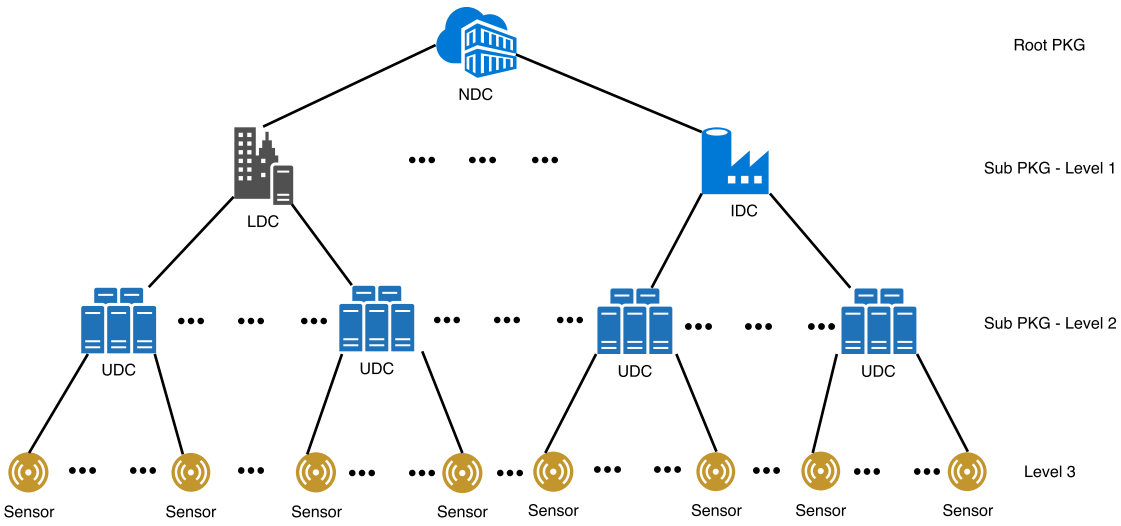

Figure 1. Hierarchical Identity-Based Encryption (HIBE) application scenario. NDC, National Data Center; LDC, Local Data Center; IDC, Industry Data Center; UDC, Unit Data Center; PKG, Private Key Generator.

1.2. Our Contributions

We present an efficient HIBE system with a constant size of public parameters, private key, and ciphertext based on the composite order group. In particular, both the private key and ciphertext of our proposed HIBE system contain only three group elements, respectively. Our improved HIBE system is constructed by tweaking the HIBE system of Lewko and Waters [7]. The proposed HIBE construction is more efficient than [7] in terms of the runtime for key generation and encryption, as well as the size of public parameters and the private key. We apply the dual system encryption technique to prove the full security under three static assumptions. We believe that our system is the first one that achieves both full security and a constant size of public parameters, private key, and ciphertext.

1.3. Related Work

Horwitz and Lynn [6] first presented the concept of HIBE. In 2002, Gentry and Silverberg [15] constructed the first HIBE scheme, which was proven secure in the random oracle model. An efficient selective-ID secure HIBE without the random oracle was proposed by Boneh and Boyen [16]. Since then, some fully-secure HIBE schemes [17,18] without the random oracle were presented. The security of the above schemes was proven using a partitioning strategy [19], which divided the identity space into identities that were used in the key query phase and challenge phase, respectively. IBE systems that were proven secure using this technique generally had large public parameters, which make them impractical. The partitioning approach is also inadequate to use for proving the security of HIBE systems.

These limitations were highlighted by Waters [19] and solved with his dual-system encryption approach. Waters realized a fully-secure HIBE system with linear-sized ciphertext based on the proof technique in [19]. Lewko and Waters [7] proposed a new technique for dual-system encryption in composite order bilinear groups. They constructed the HIBE system with constant-size ciphertexts without tags. Angelo, Vincenzo, and Giuseppe [20] modified the HIBE construction of [7] to achieve anonymity. Some other recent fully-secure HIBE systems, which were proven using the dual-system encryption technique, include [21,22]. Chen and Wee [21] proposed a compact HIBE scheme in prime-order groups, but the scheme had a linear-sized private key and ciphertext. In [22], a fully-secure and anonymous HIBE scheme with short ciphertexts in prime order (asymmetric) bilinear groups was proposed. However, the size of public parameters and the private key of their HIBE scheme grew linearly with the hierarchy depth.

Park and Lee [23] constructed the first anonymous HIBE with constant-size ciphertext over prime-order groups, which was proven secure without random oracles. The anonymous HIBE scheme proposed in [24] had a constant size of the private key and ciphertext. However, we note that their HIBE construction was only secure against selective-identity-chosen plaintext attack. Hu et al. [25] also proposed an HIBE system that achieved a constant-size ciphertext and private key without random oracle, but the size of the public parameters depended on the hierarchy depth. Since the proposed Revocable Hierarchical Identity-Based Encryption (RHIBE) by Seo and Emura [26], there have been many studies on the construction of more efficient RHIBE in terms of shorter public parameters, private key, update key, or ciphertext [27–32].

1.4. Paper Organization

Definitions of the HIBE system and complexity assumptions are briefly reviewed in Section 2. Our improved HIBE scheme is described in Section 3. The security proof is shown in Section 4. Section 5 compares the performance of our construction with the HIBE system in [7]. Section 6 gives the conclusions and an open problem of our improved scheme.

2. Preliminaries

2.1. Bilinear Groups

Let q_1, q_2, q_3 be distinct primes and $\mathbb{G}$ and $\mathbb{G}_T$ be cyclic groups of order $N = q_1q_2q_3$. To define composite order bilinear groups [33], we use a group generating algorithm $\mathcal{G}$, which takes a security parameter k as input and outputs a tuple $(N = q_1q_2q_3, \mathbb{G}, \mathbb{G}_T, e)$, where $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a map with the following properties:

1. (Bilinear) $\forall g, h \in \mathbb{G}, a, b \in \mathbb{Z}_N, e(g^a, h^b) = e(g, h)^{ab}$.
2. (Non-degenerate) $\exists g \in \mathbb{G}$ such that $e(g, g)$ has order N in $\mathbb{G}_T$.

Here, we let $\mathbb{G}_{q_1}$, $\mathbb{G}_{q_2}$, and $\mathbb{G}_{q_3}$ denote the subgroups of order q_1, q_2, and q_3 in $\mathbb{G}$, respectively. Given $h_i \in \mathbb{G}_{q_i}$ and $h_j \in \mathbb{G}_{q_j}$ for $i \neq j$, we have $e(h_i, h_j) = 1$ (i.e., the identity element in $\mathbb{G}_T$). This is known as the orthogonality property, and it will be used in our construction.

2.2. Complexity Assumptions

We use the complexity assumptions introduced in [7] to prove the security of our HIBE system. In the assumptions below, we let $\mathbb{G}_{q_1q_2}$ denote the subgroup of order q_1q_2 in $\mathbb{G}$ and $\mathbb{G}_{q_1q_3}$ denote the subgroup of order q_1q_3 in $\mathbb{G}$.

Assumption 1. *The following distribution is defined based on a group generator $\mathcal{G}$:*

$$(N = q_1q_2q_3, \mathbb{G}, \mathbb{G}_T, e) \longleftarrow \mathcal{G},$$

$$g_1 \longleftarrow \mathbb{G}_{q_1}, U_3 \longleftarrow \mathbb{G}_{q_3},$$
$$D = (N, \mathbb{G}, \mathbb{G}_T, e, g_1, U_3),$$
$$T_1 \longleftarrow \mathbb{G}_{q_1q_2}, T_2 \longleftarrow \mathbb{G}_{q_1}.$$

For an adversary $\mathcal{B}$, we define the advantage as follows:

$$\mathbf{Adv}_{\mathcal{B}}^{A1}(k) = |Pr[\mathcal{B}(D, T_1) = 1] - Pr[\mathcal{B}(D, T_2) = 1]|.$$

Definition 1. *We say that a group generator $\mathcal{G}$ satisfies Assumption 1 if for all polynomial time adversaries $\mathcal{B}$, we have that $\mathbf{Adv}_{\mathcal{B}}^{A1}(k)$ is negligible.*

Assumption 2. *The following distribution is defined based on a group generator $\mathcal{G}$:*

$$(N = q_1q_2q_3, \mathbb{G}, \mathbb{G}_T, e) \longleftarrow \mathcal{G},$$

$$g_1, U_1 \longleftarrow \mathbb{G}_{q_1}, U_2, V_2 \longleftarrow \mathbb{G}_{q_2}, U_3, V_3 \longleftarrow G_{q_3},$$
$$D = (N, \mathbb{G}, \mathbb{G}_T, e, g_1, U_1U_2, U_3, V_2V_3),$$
$$T_1 \longleftarrow \mathbb{G}, T_2 \longleftarrow \mathbb{G}_{q_1q_3}.$$

For an adversary $\mathcal{B}$, we define the advantage as follows:

$$\mathbf{Adv}_{\mathcal{B}}^{A2}(k) = |Pr[\mathcal{B}(D, T_1) = 1] - Pr[\mathcal{B}(D, T_2) = 1]|.$$

Definition 2. *We say that a group generator $\mathcal{G}$ satisfies Assumption 2 if for all polynomial time adversaries $\mathcal{B}$, we have that $\mathbf{Adv}_{\mathcal{B}}^{A2}(k)$ is negligible.*

Assumption 3. *The following distribution is defined based on a group generator $\mathcal{G}$:*

$$(N = q_1q_2q_3, \mathbb{G}, \mathbb{G}_T, e) \longleftarrow \mathcal{G}, \alpha, s \longleftarrow \mathbb{Z}_N,$$

$$g_1 \longleftarrow \mathbb{G}_{q_1}, U_2, V_2, W_2 \longleftarrow \mathbb{G}_{q_2}, U_3 \longleftarrow \mathbb{G}_{q_3},$$

$$D = (N, \mathbb{G}, \mathbb{G}_T, e, g_1, g_1^{\alpha}U_2, U_3, g_1^{s}V_2, W_2),$$

$$T_1 = e(g_1, g_1)^{\alpha s}, T_2 \longleftarrow \mathbb{G}_T.$$

For an adversary $\mathcal{B}$, we define the advantage as follows:

$$\mathbf{Adv}_{\mathcal{B}}^{A3}(k) = |Pr[\mathcal{B}(D, T_1) = 1] - Pr[\mathcal{B}(D, T_2) = 1]|.$$

Definition 3. *We say that a group generator $\mathcal{G}$ satisfies Assumption 3 if for all polynomial time adversaries $\mathcal{B}$, we have that $\mathbf{Adv}_{\mathcal{B}}^{A3}(k)$ is negligible.*

2.3. Hierarchical Identity-Based Encryption

An HIBE scheme is defined as the following five algorithms:

- **GlobalSetup** (1^k): On input 1^k where k is a security parameter, it returns the public parameters PK and a master secret key MSK.
- **KeyGen** ($MSK, ID = (ID_1, \ldots, ID_j)$): On input MSK and an identity $ID = (ID_1, \ldots, ID_j)$ of depth j, it returns a private key SK of $ID = (ID_1, \ldots, ID_j)$.
- **Delegate**($PK, SK_{ID=(ID_1,\ldots,ID_j)}, ID_{j+1}$): On input PK, a private key for $ID = (ID_1, \ldots, ID_j)$, and an identity ID_{j+1}, it returns a private key for ID=($ID_1, \ldots, ID_{j+1}$).
- **Encrypt** ($PK, M, (ID_1, \ldots, ID_j)$): On input message M, PK, and identity $(ID_1, \ldots, ID_j)$, it returns a ciphertext C.
- **Decrypt**($SK_{ID=(ID_1,\ldots,ID_j)}, C$): On input $SK_{ID=(ID_1,\ldots,ID_j)}$ and C, it returns the message M.

2.4. Security Definition

The security of HIBE [34] is defined in term of the following game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Setup: The challenger $\mathcal{C}$ runs the **GlobalSetup** algorithm and obtains the public parameters. $\mathcal{C}$ also maintains a set S for private keys it has created. Initially, $S = \emptyset$. $\mathcal{C}$ gives the public parameters to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ issues the following queries:

- **Create:** The identity vector $ID = (ID_1, \ldots, ID_j)$ of depth j is given to $\mathcal{C}$ by $\mathcal{A}$. $\mathcal{C}$ runs the **KeyGen** algorithm to generate the key for this identity vector. The key is then added in the set S. A reference of this key is returned to $\mathcal{A}$.
- **Delegate:** $\mathcal{A}$ specifies a private key SK_{ID} in S and gives an identity ID_{j+1} to $\mathcal{C}$. $\mathcal{C}$ runs the **Delegate** algorithm to generate a new private key for ID=($ID_1, \ldots, ID_{j+1}$) and adds this key to S. It returns a reference of this key to $\mathcal{A}$.
- **Reveal:** $\mathcal{A}$ specifies an element of the set S. $\mathcal{C}$ gives this private key to $\mathcal{A}$ and removes it from S. At this point, $\mathcal{A}$ no longer needs to make delegation queries for this private key, as it can run the **Delegate** algorithm by itself.

Challenge: $\mathcal{A}$ gives two equal-length messages M_0 and M_1, as well as a challenge identity ID^* to $\mathcal{C}$. The restriction is that no revealed identity in Phase 1 is a prefix of this challenge identity. $\mathcal{C}$ flips a coin $\beta \in \{0, 1\}$ and encrypts M_β under ID^*. It gives the resulting ciphertext C^* to $\mathcal{A}$.

Phase 2: This phase is identical to Phase 1 with the restriction that any revealed identity must not be a prefix of the challenge identity ID^*.

Guess. $\mathcal{A}$ outputs a guess $\beta' \in \{0,1\}$.
If $\beta = \beta'$, then $\mathcal{A}$ wins the game. We define the advantage of an adversary $\mathcal{A}$ in this game to be:

$$\mathbf{Adv}_{\mathcal{A}}^{HIBE}(k) = Pr[\beta' = \beta] - \frac{1}{2}.$$

Definition 4. *A hierarchical identity-based encryption scheme is secure if all polynomial time adversaries have at most a negligible advantage in the above security game.*

3. Our Improved HIBE System

Our improved HIBE system achieves short parameters with a constant size of ciphertext, private key, and public parameter. The idea of the HIBE construction is to tweak the original Lewko and Waters HIBE system [7] by reducing the parameters without sacrificing its security. We find out that the aim of adding $u_i \in \mathbb{G}_{q_1}$ to the original HIBE [7] is to make the key random in key delegation. We replace $u_i \in \mathbb{G}_{q_1}$ with $u_1 \in \mathbb{G}_{q_1}$, and the randomness is still the same as the original HIBE where the key K_1, K_2, E has a random $r' \in \mathbb{Z}_N$ in the exponent. Therefore, even when we take out $u_i \in \mathbb{G}_{q_1}$, the key remains fully randomized, and thus, our improved HIBE system also achieves full security.

3.1. Construction

Suppose $\mathbb{G}$ and $\mathbb{G}_T$ denote bilinear groups of order $N = q_1q_2q_3$, where q_1, q_2, q_3 are distinct primes. Let $\mathbb{G}_{q_i}$ be the subgroup of order q_i in $\mathbb{G}$ and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be the bilinear map. The HIBE system is constructed as follows.

- **GlobalSetup** (1^k): Let k be the security parameter, $g_1, h_1, u_1 \in \mathbb{G}_{q_1}, U_3 \in \mathbb{G}_{q_3}$, and $\alpha \in \mathbb{Z}_N$. The public parameters PK and master secret key MSK are generated as:

$$PK = \{N, g_1, h_1, u_1, U_3, e(g_1, g_1)^{\alpha}\}, MSK = \alpha.$$

- **KeyGen** $(MSK, ID = (ID_1, \ldots, ID_j))$: The key generation algorithm first selects a random $r \in \mathbb{Z}_N$ and random elements R_3, R_3', R_3'' of $\mathbb{G}_{q_3}$. It then generates the private key for an identity $ID = (ID_1, \ldots, ID_j)$ of depth j by computing:

$$K_1 = g_1^r R_3, K_2 = g_1^{\alpha}(u_1^{ID_1+\ldots+ID_j} h_1)^r R_3', E = u_1^r R_3''.$$

  It outputs the private key as $SK = (K_1, K_2, E)$.
- **Delegate**$(PK, SK_{ID=(ID_1,\ldots,ID_j)}, ID_{j+1})$: Given a private key $SK = (K_1', K_2', E')$ for $(ID_1, \ldots, ID_j)$, a new key for $(ID_1, \ldots, ID_{j+1})$ is created as follows. The delegation algorithm selects a random $r' \in \mathbb{Z}_N$ and random elements $\tilde{R}_3, \tilde{R}_3', \tilde{R}_3'' \in \mathbb{G}_{q_3}$. The new key is computed as: $K_1 = K_1' g_1^{r'} \tilde{R}_3$, $K_2 = K_2'(u_1^{ID_1+\ldots+ID_j} h_1)^{r'} (E')^{ID_{j+1}} u_1^{r' ID_{j+1}} \tilde{R}_3'$, $E = E' u_1^{r'} \tilde{R}_3''$. It outputs the private key for $ID = (ID_1, \ldots, ID_j, ID_{j+1})$ as $SK = (K_1, K_2, E)$. We fully rerandomize this new key, i.e., the new key is only related to the values $ID_1, \ldots, ID_j$ of the previous key.
- **Encrypt** $(PK, M, (ID_1, \ldots, ID_j))$: To encrypt a message M under the public key $(ID_1, \ldots, ID_j)$, the encryption algorithm selects a random $s \in \mathbb{Z}_N$. It computes:

$$C_0 = M \cdot e(g_1, g_1)^{\alpha s}, C_1 = (u_1^{ID_1+\ldots+ID_j} h_1)^s, C_2 = g_1^s.$$

- **Decrypt**$(SK_{ID=(ID_1,\ldots,ID_j)}, C)$: Given an identity $(ID_1, \ldots, ID_j)$ and a ciphertext $C = (C_0, C_1, C_2)$, it decrypts the ciphertext with the private key $SK_{ID} = (K_1, K_2, E)$ by first computing:

$$\begin{aligned} A &= \frac{e(K_2, C_2)}{e(K_1, C_1)} \\ &= \frac{e(g_1^{\alpha}(u_1^{ID_1+\ldots+ID_j}h_1)^r R_3', g_1^s)}{e(g_1^r R_3, (u_1^{ID_1+\ldots+ID_j}h_1)^s)} \\ &= \frac{e(g_1, g_1)^{\alpha s} \cdot e(u_1^{ID_1+\ldots+ID_j}h_1, g_1)^{rs} \cdot e(R_3', g_1^s)}{e(g_1, u_1^{ID_1+\ldots+ID_j}h_1)^{rs} \cdot e(R_3, (u_1^{ID_1+\ldots+ID_j}h_1)^s)} \\ &= e(g_1, g_1)^{\alpha s} \end{aligned}$$

Based on the orthogonality property, we have $e(R_3', g_1^s) = 1$ and $e(R_3, (u_1^{ID_1+\ldots+ID_j}h_1)^s) = 1$. The message is then recovered by computing:

$$C_0 / A = M$$

3.2. Semi-Functional Algorithms

The security of our HIBE system is proven by using the dual system encryption [7,19] methodology. With this proof technique, it is required to define an additional ciphertext and key, namely a semi-functional ciphertext and key. We only use them in the security proof of the HIBE system, but not in the real system.

We denote g_2 as a generator of $\mathbb{G}_{q_2}$. The semi-functional key is created based on the normal private key (K_1', K_2', E') that was generated by the key generation algorithm. We then select exponents $\gamma, z_k, z \in_R \mathbb{Z}_N$ and generate the semi-functional key as:

$$K_1 = K_1' \cdot g_2^{\gamma}, K_2 = K_2' \cdot g_2^{\gamma z_k}, E = E' \cdot g_2^{\gamma z}.$$

The semi-functional ciphertext is created based on the normal ciphertext (C_0', C_1', C_2') generated by the encryption algorithm. We then select exponents $x, z_c \in_R \mathbb{Z}_N$ and generate the semi-functional ciphertext as:

$$C_0 = C_0', C_1 = C_1' \cdot g_2^{x z_c}, C_2 = C_2' \cdot g_2^{x}.$$

The decryption of a semi-functional ciphertext under a semi-functional key is computed by multiplying the blinding factor with $e(g_2, g_2)^{x\gamma(z_k - z_c)}$. We will obtain a correct decryption if $z_c = z_k$.

4. Security Analysis

The security proof of our HIBE system is shown with a sequence of games from $\mathbf{Game_{Real}}$–$\mathbf{Game_{Final}}$ that are played between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. We will show that no polynomial-time adversary $\mathcal{A}$ can distinguish one game from the next under a complexity assumption.

$\mathbf{Game_{Real}}$: The first game is the usual security game used for defining HIBE security. In this game, normal private keys and the challenge ciphertext are used between $\mathcal{A}$ and $\mathcal{C}$.

$\mathbf{Game_{Real'}}$: This game is the same as $\mathbf{Game_{Real}}$ with the exception that all key queries will be answered by fresh calls to the key generation algorithm (the challenger $\mathcal{C}$ will not be asked to delegate keys in a particular way).

$\mathbf{Game_{Restricted}}$: This game is almost identical to $\mathbf{Game_{Real'}}$ except that the adversary $\mathcal{A}$ is restricted from making key queries for identities, which are prefixes of the challenge identity modulo q_2.

$\mathbf{Game}_k$: $\mathbf{Game}_k$ is similar to $\mathbf{Game_{Restricted}}$, except the following changes: 1. The first k keys are semi-functional for k from 0–w, where w denotes the number of key queries made by $\mathcal{A}$. The rest of

the keys are normal. 2. The ciphertext given to $\mathcal{A}$ is semi-functional. In $\mathbf{Game}_0$, all of the keys are normal. and a semi-functional challenge ciphertext is given to $\mathcal{A}$. In $\mathbf{Game}_w$, all the keys and challenge ciphertext given to $\mathcal{A}$ are semi-functional.

$\mathbf{Game_{Final}}$**:** This game is the same as $\mathbf{Game}_w$, except that the challenger gives a semi-functional encryption of a random message to $\mathcal{A}$ as the challenge ciphertext. The random message is independent of the messages provided by $\mathcal{A}$.

Lemma 1. *For any adversary* $\mathcal{A}$, $\mathbf{Adv}_{\mathcal{A}}^{Game_{Real}} = \mathbf{Adv}_{\mathcal{A}}^{Game_{Real'}}$.

Proof. Keys that are returned to $\mathcal{A}$'s queries are identically distributed whether they are generated by the **Delegate** algorithm from a previous key or from a fresh call to the **KeyGen** algorithm. $\mathcal{A}$'s view in $\mathbf{Game_{Real'}}$ is identical to its view in $\mathbf{Game_{Real}}$. □

Lemma 2. *Suppose there exists an adversary* $\mathcal{A}$ *such that* $\mathbf{Adv}_{\mathcal{A}}^{Game_{Real'}} - \mathbf{Adv}_{\mathcal{A}}^{Game_{Restricted}} = \epsilon$. *Then, we can build an algorithm* $\mathcal{B}$ *with advantage* $\mathbf{Adv}_{\mathcal{B}}^{A2} \geq \frac{\epsilon}{2}$ *in breaking Assumption 2.*

Proof. Suppose $\mathcal{A}$ has a probability ϵ of producing identities ID and ID^* such that $ID \neq ID^*$ modulo N and q_2 divides $ID - ID^*$. $\mathcal{B}$ can then determine a nontrivial factor of N by computing $a = gcd(ID - ID^*, N)$. Let $b = \frac{N}{a}$. We consider the following three cases:

1. one of a, b is q_1, and the other is $q_2 q_3$;
2. one of a, b is q_2, and the other is $q_1 q_3$;
3. one of a, b is q_3, and the other is $q_1 q_2$.

Given $D = (N, \mathbb{G}, \mathbb{G}_T, e, g_1, U_1U_2, U_3, V_2V_3)$ and T, where $T \in \mathbb{G}$ or $T \in \mathbb{G}_{q_1 q_3}$, $\mathcal{B}$ can simulate the security games with $\mathcal{A}$ and determine which of the above cases has occurred as follows:

- Case 1: $\mathcal{B}$ first tests whether $(V_2V_3)^a = 1$ or $(V_2V_3)^b = 1$. If either one of these equalities holds, then Case 1 occurs. $\mathcal{B}$ subsequently tests whether $e(T^a, U_1U_2) = 1$ (we assume without loss of generality that $a = q_1$ and $b = q_2 q_3$). If the equality holds, $\mathcal{B}$ determines that $T \in \mathbb{G}_{q_1 q_3}$. Otherwise, $T \in \mathbb{G}$.
- Case 2: $\mathcal{B}$ first tests whether $(U_1U_2)^a = 1$ or $(U_1U_2)^b = 1$. If neither of these holds and the test for Case 1 fails, then Case 2 occurs. Next, $\mathcal{B}$ can determine which of a, b is equal to $q_1 q_3$ by testing which of g_1^a, g_1^b is the identity. Without loss of generality, we assume that $a = q_2$ and $b = q_1 q_3$. $\mathcal{B}$ subsequently tests whether $T^b = 1$. If the equality holds, $\mathcal{B}$ determines that $T \in \mathbb{G}_{q_1 q_3}$. Otherwise, $T \in \mathbb{G}$.
- Case 3: If the tests for Cases 1 and 2 fail, then Case 3 occurs. $\mathcal{B}$ can then determine which of a, b is equal to q_3 by testing which of U_3^a, U_3^b is the identity. We assume without loss of generality that $a = q_3$. $\mathcal{B}$ subsequently tests whether $e(T^a, V_2V_3) = 1$. If the equality holds, $\mathcal{B}$ determines that $T \in \mathbb{G}_{q_1 q_3}$. Otherwise, $T \in \mathbb{G}$.

□

Lemma 3. *Suppose there exists an adversary* $\mathcal{A}$ *such that* $\mathbf{Adv}_{\mathcal{A}}^{Game_{Restricted}} - \mathbf{Adv}_{\mathcal{A}}^{Game_0} = \epsilon$. *Then, we can build an algorithm* $\mathcal{B}$ *with advantage* $\mathbf{Adv}_{\mathcal{B}}^{A1} = \epsilon$ *in breaking Assumption 1.*

Proof. Given $D = (N, \mathbb{G}, \mathbb{G}_T, e, g_1, U_3)$ and T, where $T \in \mathbb{G}_{q_1 q_2}$ or $T \in \mathbb{G}_{q_1}$, $\mathbf{Game_{Restricted}}$ or $\mathbf{Game}_0$ is simulated as follows. $\mathcal{B}$ first selects random exponents $\alpha, a, b \in \mathbb{Z}_N$. It then sets $g_1 = g_1, u_1 = g_1^a, h_1 = g_1^b$. $\mathcal{B}$ keeps the master secret key $MSK = \alpha$. It sends the public parameters $PK = \{N, g_1, h_1, u_1, e(g_1, g_1)^\alpha\}$ to $\mathcal{A}$.

$\mathcal{A}$'s queries for the private key of identity $ID = (ID_1, \cdots, ID_j)$ are answered by $\mathcal{B}$ as follows: $\mathcal{B}$ selects random exponents r, t, v and $w \in \mathbb{Z}_N$ and sets:

$$K_1 = g_1^r U_3^t, K_2 = g_1^\alpha (u_1^{ID_1 + \ldots + ID_j} h_1)^r U_3^w, E = u_1^r U_3^v$$

At some point, $\mathcal{B}$ receives a challenge identity $ID^* = (ID_1^*, \cdots, ID_j^*)$ and two messages M_0, M_1 from $\mathcal{A}$. $\mathcal{B}$ chooses a random $\beta \in \{0,1\}$ and returns the challenge ciphertext as follows:

$$C_0 = M_\beta \cdot e(T, g_1)^\alpha, C_1 = T^{a(ID_1^* + \ldots + ID_j^*) + b}, C_2 = T.$$

If $T \in \mathbb{G}_{q_1 q_2}$, the simulated semi-functional ciphertexts C_0, C_1, C_2 could match the primary semi-functional ciphertexts as follows:

$$C_0 = M_\beta \cdot e(g_1^s g_2^x, g_1)^\alpha = M_\beta \cdot e(g_1, g_1)^{\alpha s}$$

$$\begin{aligned} C_1 &= T^{(a(ID_1^* + \ldots + ID_j^*) + b)} \\ &= (g_1^s g_2^x)^{(a(ID_1^* + \cdots + ID_j^*) + b)} \\ &= g_1^{s(a(ID_1^* + \cdots + ID_j^*) + b)} \; g_2^{x(a(ID_1^* + \cdots + ID_j^*) + b)} \\ &= (u_1^{(ID_1^* + \cdots + ID_j^*)} h_1)^s g_2^{x z_c} \end{aligned}$$

$$C_2 = g_1^s g_2^x$$

The ciphertext (C_0, C_1, C_2) is semi-functional with T implicitly set as $g_1^s g_2^x$ and $z_c = a(ID_1^* + \ldots + ID_j^*) + b$ in $\mathbf{Game}_0$. This is properly distributed as $z_c \mod q_2$ is not correlated with $a \mod q_1$ and $b \mod q_1$. If $T \in \mathbb{G}_{q_1}$, then the ciphertext (C_0, C_1, C_2) is normal in $\mathbf{Game}_{\mathbf{Restricted}}$ with T implicitly set as g_1^s. Therefore, these possibilities for T can be distinguished by $\mathcal{B}$ using the output of $\mathcal{A}$. □

Lemma 4. *Suppose there exists an adversary $\mathcal{A}$ such that $\mathbf{Adv}_{\mathcal{A}}^{Game_{k-1}} - \mathbf{Adv}_{\mathcal{A}}^{Game_k} = \epsilon$. Then, we can build an algorithm $\mathcal{B}$ with advantage $\mathbf{Adv}_{\mathcal{B}}^{A2} = \epsilon$ in breaking Assumption 2.*

Proof. Given $D = (N, \mathbb{G}, \mathbb{G}_T, e, g_1, U_1 U_2, U_3, V_2 V_3)$ and T, where $T \in \mathbb{G}$ or $T \in \mathbb{G}_{q_1 q_3}$, $\mathbf{Game}_{k-1}$ or $\mathbf{Game}_k$ is simulated as follows. $\mathcal{B}$ selects random exponents $\alpha, a, b \in \mathbb{Z}_N$ and sets $g_1 = g_1, u_1 = g_1^a, h_1 = g_1^b$. $\mathcal{B}$ sends the public parameters $PK = \{N, g_1, h_1, u_1, e(g_1, g_1)^\alpha\}$ to $\mathcal{A}$ and keeps the master secret key $MSK = \alpha$.

$\mathcal{B}$ answers $\mathcal{A}$'s i^{th} key query for identity $(ID_1, \cdots, ID_j)$ as follows:

- For $i < k$, $\mathcal{B}$ selects random exponents $r, z, t, v \in \mathbb{Z}_N$ first and then creates a semi-functional key as: $K_1 = g_1^r (V_2 V_3)^t$, $K_2 = g_1^\alpha (u_1^{ID_1 + \ldots + ID_j} h_1)^r (V_2 V_3)^z$, $E = u_1^r (V_2 V_3)^v$.

  The semi-functional key is properly distributed with $g_2^\gamma = V_2^t$. We note that values of t and zmodulo q_1 and modulo q_3 are uncorrelated by the Chinese remainder theorem.
- For $i > k$, $\mathcal{B}$ runs the **KeyGen** algorithm to generate a normal key.
- For $i = k$, $\mathcal{B}$ sets $z_k = a(ID_1 + \cdots + ID_j) + b$ first and then creates the following key by choosing random exponents $w_k, w \in \mathbb{Z}_N$:

  $$K_1 = T, K_2 = g_1^\alpha T^{z_k} U_3^{w_k}, E = T^a U_3^w.$$

  If $T \in \mathbb{G}_{q_1 q_3}$, it means that T is implicitly set as $g_1^r R_3$ and the key is normal. If $T \in \mathbb{G}$, it means that T is implicitly set as $g_1^r g_2^\gamma R_3$ and we have a semi-functional key.

At some point, $\mathcal{B}$ receives a challenge identity $ID^* = (ID_1^*, \cdots, ID_j^*)$ and two messages M_0, M_1 from $\mathcal{A}$. $\mathcal{B}$ chooses a random $\beta \in \{0,1\}$ and returns the challenge ciphertext as follows: $C_0 = M_\beta \cdot e(U_1 U_2, g_1)^\alpha$, $C_1 = (U_1 U_2)^{a(ID_1^* + \ldots + ID_j^*) + b}$, $C_2 = U_1 U_2$.

This means that $g_1^s = U_1$ and $z_c = a(ID_1^* + \ldots + ID_j^*) + b$ are implicitly set in the challenge ciphertext. Since ID_k is not a prefix of the challenge $ID^* \mod q_2$, z_k and z_c are independent and randomly distributed.

If $\mathcal{B}$ generates a semi-functional ciphertext for ID_k and attempts to test whether the kth key is semi-functional by trying the decryption, then we will have $z_c = z_k$, which makes the decryption always work regardless.

If $T \in \mathbb{G}_{q_1 q_3}$, then $\mathcal{B}$ has properly simulated $\mathbf{Game}_{k-1}$. If $T \in \mathbb{G}$, then $\mathcal{B}$ has properly simulated $\mathbf{Game}_k$. Therefore, these possibilities for T can be distinguished by $\mathcal{B}$ using the output of $\mathcal{A}$. □

Lemma 5. *Suppose there exists an adversary $\mathcal{A}$ such that $\mathbf{Adv}_{\mathcal{A}}^{Game_w} - \mathbf{Adv}_{\mathcal{A}}^{Game_{Final}} = \epsilon$. Then, we can build an algorithm $\mathcal{B}$ with advantage $\mathbf{Adv}_{\mathcal{B}}^{A3} = \epsilon$ in breaking Assumption 3.*

Proof. Given $D = (N, \mathbb{G}, \mathbb{G}_T, e, g_1, g_1^{\alpha}, U_2, U_3, g_1^s V_2, W_2)$ and T, where $T = e(g_1, g_1)^{\alpha s}$ or T is a random element of $\mathbb{G}_T$, $\mathcal{B}$ simulates $\mathbf{Game}_w$ or $\mathbf{Game_{Final}}$ with $\mathcal{A}$ as follows. It chooses random exponents $a, b \in \mathbb{Z}_N$ and sets $g_1 = g_1, u_1 = g_1^a, h_1 = g_1^b, e(g_1, g_1)^{\alpha} = e(g_1^{\alpha} U_2, g_1)$. $\mathcal{B}$ sends the public parameters $PK = \{N, g_1, h_1, u_1, e(g_1, g_1)^{\alpha}\}$ to $\mathcal{A}$.

$\mathcal{A}$'s queries for the private key of identity $ID = (ID_1, \cdots, ID_j)$ are answered by $\mathcal{B}$ as follows: $\mathcal{B}$ randomly selects exponents $c, r, t, w, z \in \mathbb{Z}_N$, and sets a semi-functional key as: $K_1 = g_1^r W_2 U_3^t$, $K_2 = g_1^{\alpha} (u_1^{ID_1 + \ldots + ID_j} h_1)^r W_2^{a(ID_1 + \ldots + ID_j) + b} U_3^w$, $E = u_1^r W_2^z U_3^c$.

This means that it implicitly sets $W_2 = g_2^{\gamma}, z_k = a(ID_1 + \ldots + ID_j) + b$. At some point, $\mathcal{B}$ receives a challenge identity $ID^* = (ID_1^*, \cdots, ID_j^*)$ and two messages M_0, M_1 from $\mathcal{A}$. $\mathcal{B}$ chooses a random $\beta \in \{0, 1\}$ and returns the challenge ciphertext as follows:

$$C_0 = M_\beta T, C_1 = (g_1^s V_2)^{a(ID_1^* + \ldots + ID_j^*) + b}, C_2 = g_1^s V_2.$$

This implicitly sets $z_c = a(ID_1^* + \ldots + ID_j^*) + b$. We note that $u_1 = g_1^a$ and $h_1 = g_1^b$ are elements of $\mathbb{G}_{q_1}$, so when a and b are randomly chosen from $\mathbb{Z}_N$, the values of $a \mod q_1$ and $b \mod q_1$ and the value $z_c = a(ID_1^* + \ldots + ID_j^*) + b \mod q_2$ are random and independent.

If $T = e(g_1, g_1)^{\alpha s}$, then we have a properly-distributed semi-functional ciphertext (C_0, C_1, C_2) under the message M_β in $\mathbf{Game}_w$. If T is a random element of $\mathbb{G}_T$, then we have a semi-functional ciphertext (C_0, C_1, C_2) under a random message in $\mathbf{Game_{Final}}$. Therefore, these possibilities for T can be distinguished by $\mathcal{B}$ using the output of $\mathcal{A}$. □

Theorem 1. *If Assumptions 1–3 hold, then the proposed HIBE system is secure.*

Proof. If Assumptions 1–3 hold, then we have shown by the previous lemmas that the real security game is indistinguishable from $\mathbf{Game_{Final}}$, in which the value of $\mathcal{B}$ is information-theoretically hidden from the adversary. Hence, the adversary can obtain no advantage in breaking the HIBE system. □

5. Performance Comparison

We provide a performance comparison of our improved HIBE system with other HIBE systems proposed in [7,19,22,25]. The efficiency comparison is summarized in Table 1. Let n be the maximum depth of the HIBE and d be the depth of identity vector. "PK size", "SK size", and "CT size" denote the length of public parameters, private key, and ciphertext, respectively. We use P, E, and E_T to represent the computational cost of a bilinear pairing, a $\mathbb{G}$-exponentiation, and a $\mathbb{G}_T$-exponentiation operation respectively for the **KeyGen**, **Encrypt**, and **Decrypt** algorithms.

The public parameters size of [19] contains $(2n + 13)$ group elements. The private key size of [19] is $O(d)$. The public parameters and private key of [7] contain $(n + 3)$ and $(n - d + 2)$ group elements, respectively. The public parameters size of [25] is $(2bn + 3)|\mathbb{G}| + 1|\mathbb{Z}_p|$, where b is the bit length of an identity. However, the private key size of [25] is $O(1)$. In [22], the public parameters size is $O(n)$ and the private key contains $6(n - d) + 12$ group elements. In contrast, our improved HIBE system only requires constant-size public parameters and private key (i.e., independent of the depth of the hierarchy). In particular, public parameters and private key in our HIBE system contain only four

and three group elements, respectively. All the HIBE systems in this comparison have constant-size ciphertext except [19] with $O(d)$ of ciphertext size.

Table 1. Efficiency comparison between existing and our improved HIBE system.

Scheme	PK size	SK size	CT size	KeyGen	Encrypt	Decrypt
[19]	$O(n)$	$O(d)$	$O(d)$	$(4d+11)E$	$(3d+11)E+1E_T$	$(2d+7)P$
[7]	$O(n)$	$O(n-d)$	$O(1)$	$(n+3)E$	$(d+2)E+1E_T$	$2P$
[25]	$O(bn)$	$O(1)$	$O(1)$	$(2bd+6)E$	$(2bd+3)E+2E_T$	$2P+1E_T$
[22]	$O(n)$	$O(n-d)$	$O(1)$	$8n-6d+17$	$(3d+4)E+1E_T$	$6P$
Ours	$O(1)$	$O(1)$	$O(1)$	$5E$	$3E+1E_T$	$2P$

All of the three computational costs for [19] grow linearly with the depth of the identity vector. The computational costs of the **Decrypt** algorithm for [7,22,25] and our improved HIBE systems are all independent of the hierarchy depth. However, our improved HIBE system achieves better computation efficiency for the **KeyGen** and **Encrypt** algorithms as they are independent of the depth of the hierarchy and identity vector, respectively.

We implemented our improved HIBE system on a Windows 10 PC with a CPU of Intel Core i7-7700 (3.60 GHz) and memory of 16GB. We ran the **KeyGen** and **Encrypt** algorithms for different hierarchy depths ($n = 1, 5, 10, ..., 50$) and recorded the run time as shown in Tables 2 and 3, respectively. As shown in Figures 2 and 3, the run times of the **KeyGen** and **Encrypt** algorithms for our improved HIBE system were much faster than the original HIBE system in [7].

Table 2. The private key generation time of [7] and our improved HIBE system.

Depth	[7] (s)	Ours (s)
1	38.851	2.444
5	184.653	12.122
10	348.413	23.886
15	493.993	36.009
20	616.044	47.865
25	734.034	59.715
30	834.744	72.646
35	886.534	85.998
40	975.188	98.309
45	1012.288	108.427
50	1028.230	125.729

Table 3. The encryption time of [7] and our improved HIBE system.

Depth	[7] (s)	Ours (s)
1	0.887	0.857
5	0.949	0.840
10	1.051	0.825
15	1.132	0.839
20	1.289	0.831
25	1.427	0.828
30	1.517	0.841
35	1.536	0.852
40	1.635	0.840
45	1.755	0.842
50	1.817	0.882

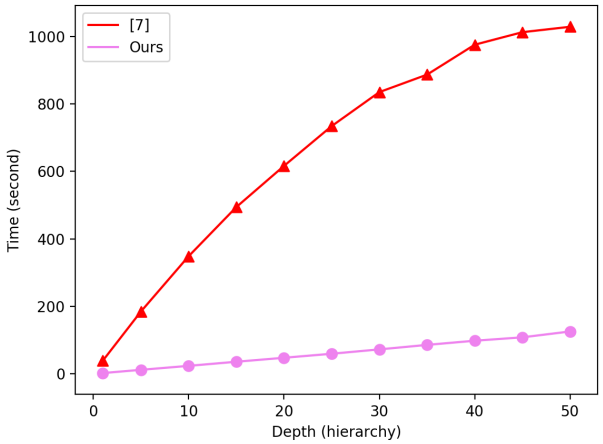

Figure 2. Run time of the **KeyGen** algorithm for [7] and our improved HIBE system.

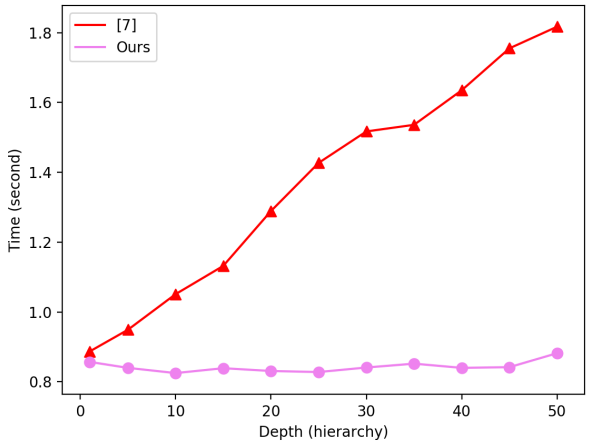

Figure 3. Run time of the **Encrypt** algorithm for [7] and our improved HIBE system.

6. Conclusions and Future Work

Deploying efficient data protection mechanisms are the main challenges for a complex and heterogeneous Internet of Things infrastructure. We proposed an efficient HIBE system with short parameters where the public parameters, private key, and ciphertext are all independent of the hierarchy or identity depth. We proved the security of the improved system using the dual-system encryption technique. This is the only HIBE system simultaneously achieving full security in the standard model and providing $O(1)$-sized public parameters, private key, and ciphertext. Our improved HIBE construction is based on composite order groups. In general, an HIBE construction in prime order groups can achieve similar security levels with a smaller size of group elements. This may result in shorter ciphertexts, and the decryption is more efficient as compared to an HIBE system constructed over composite order groups [23]. Therefore, it would be interesting to explore the construction of an HIBE system based on prime order groups that can achieve all these properties simultaneously.

Author Contributions: Writing, original draft preparation, L.G. and J.W.; writing, review and editing, W.-C.Y.

Funding: Lifeng Guo was supported by National Science Foundation of China (Grant No. 61202365), Project for Returned Overseas of Shanxi Province (2015-015), Natural Science Foundation of Shanxi Province, China (Grant No. 201701D12052), and National Natural Science Foundation of China (Grant No. 61872226).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; nor in the decision to publish the results.

References

1. Shamir, A. Identity-Based Cryptosystems and Signature Schemes. In *Advances in Cryptology*; Springer: Berlin/Heidelberg, Germany, 1984; pp. 47–53.
2. Boneh, D.; Franklin, M. Identity-Based Encryption from the Weil Pairing. In *Advances in Cryptology-CRYPTO 2001*; Springer: Berlin/Heidelberg, Germany, 2001; pp. 213–229.
3. Boneh, D.; Boyen, X. Secure Identity Based Encryption Without Random Oracles. In *Advances in Cryptology—CRYPTO 2004*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 443–459.
4. Gentry, C. Practical Identity-Based Encryption Without Random Oracles. In *Advances in Cryptology—EUROCRYPT 2006*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 445–464.
5. Waters, B. Efficient Identity-Based Encryption Without Random Oracles. In *Advances in Cryptology—EUROCRYPT 2005*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 1140-127.
6. Horwitz, J.; Lynn, B. Towards Hierarchical Identity-Based Encryption. In *Advances in Cryptology—EUROCRYPT 2002*; Springer: Berlin/Heidelberg, 2002; pp. 466–481.
7. Lewko, A.; Waters, B. New Techniques for Dual System Encryption and Fully Secure HIBE with Short Ciphertexts. In *Theory of Cryptography—TCC2010*; Springer: Berlin/Heidelberg, 2010; pp. 455–479.
8. Daniel, R.M.; Rajsingh, E.B.; Silas, S. Analysis of Hierarchical Identity Based Encryption Schemes and Its Applicability to Computing Environments. *J. Inf. Secur. Appl.* **2017**, *36*, 20–31. [CrossRef]
9. Li, Y.; Wang, Y.; Zhang, Y. SecHome: A Secure Large-Scale Smart Home System Using Hierarchical Identity Based Encryption. In *Information and Communications Security*; Springer: Cham, Switzerland, 2018; pp. 339–351.
10. Sha, K.; Wei, W.; Yang, T.A.; Wang, Z.; Shi, W. On Security Challenges and Open Issues in Internet of Things. *Future Gener. Comput. Syst.* **2018**, *83*, 326–337. [CrossRef]
11. Trnka, M.; Cerny, T.; Stickney, N. Survey of Authentication and Authorization for the Internet of Things. *Secur. Commun. Netw.* **2018**, *2018*, 4351603. [CrossRef]
12. Yu, F.R.; Tang, H.; Mason, P.C.; Wang, F. A Hierarchical Identity Based Key Management Scheme in Tactical Mobile Ad Hoc Networks. *IEEE Trans. Netw. Serv. Manag.* **2010**, *7*, 258–267. [CrossRef]
13. Ning, H.; Liu, H.; Yang, L.T. Aggregated-Proof Based Hierarchical Authentication Scheme for the Internet of Things. *IEEE Trans. Parallel Distrib. Syst.* **2015**, *26*, 657–667. [CrossRef]
14. Yang, L.T.; Liu, H.; Ning, H. Cyberentity Security in the Internet of Things. *Computer* **2013**, *46*, 46–53.
15. Gentry, C.; Silverberg, A. Hierarchical ID-Based Cryptography. In *Advances in Cryptology—ASIACRYPT 2002*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 548–566.
16. Boneh, D.; Boyen, X. Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In *Advances in Cryptology—EUROCRYPT 2004*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 223–238.
17. Chatterjee, S.; Sarkar, P. HIBE With Short Public Parameters Without Random Oracle. In *Advances in Cryptology—ASIACRYPT 2006*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 145–160.
18. Sarkar, P.; Chatterjee, S. Construction of a Hybrid HIBE Protocol Secure against Adaptive Attacks: without Random Oracle. In *First International Conference on Provable Security—ProvSec 2007*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 51–67.
19. Waters, B. Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In *Advances in Cryptology—CRYPTO 2009*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 619–636.
20. De Caro, A.; Iovino, V.; Persiano, G. Fully Secure Anonymous HIBE and Secret-Key Anonymous IBE with Short Ciphertexts. In *Pairing-Based Cryptography—Pairing 2010*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 347–366.
21. Chen, J.; Wee, H. Dual System Groups and its Applications—Compact HIBE and More. 2014. Available online: https://eprint.iacr.org/2014/265.pdf (accessed on 18 June 2019)
22. Lee, K.; Park, J.H.; Lee. D.H. Anonymous HIBE with Short Ciphertexts: Full Security in Prime Order Groups. *Designs Codes Cryptogr.* **2015**, *74*, 395–425. [CrossRef]

23. Park, J.H.; Lee, D.H. Anonymous HIBE: Compact Construction Over Prime-Order Groups. *IEEE Trans. Inf. Theory* **2013**, *59*, 2531–2541. [CrossRef]
24. Zhang, L.; Mu, Y.; Wu, Q. Compact Anonymous Hierarchical Identity-Based Encryption with Constant Size Private Keys. *Comput. J.*, **2016**, *59*, 452–461. [CrossRef]
25. Hu, X.; Wang, J.; Xu, H.; Yang, Y. Constant Size Ciphertext and Private Key HIBE without Random Oracles. *J. Inf. Sci. Eng.* **2014**, *30*, 333–345.
26. Seo, J.H.; Emura, K. Revocable Hierarchical Identity-based Encryption. *Theor. Comput. Sci.* **2014**, *542*, 44-–62. [CrossRef]
27. Jia, H.; Chen, Y.; Lan, J.; Huang, K.; Wang, J. Efficient Revocable Hierarchical Identity-Based Encryption using Cryptographic Accumulators. *Int. J. Inf. Secur.* **2018**, *17*, 477–490. [CrossRef]
28. Lee, K.; Park, S. Revocable Hierarchical Identity-Based Encryption with Shorter Private Keys and Update Keys. *Designs Codes Cryptogr.* **2018**, *86*, 2407–2440. [CrossRef]
29. Park, S.; Lee, D.H.; Lee, K. Revocable Hierarchical Identity-Based Encryption from Multilinear Maps. *arXiv* **2016**, arXiv:1610.07948.
30. Wang, C.; Li, Y.; Jiang, S.; Wu, J. An Efficient Adaptive-ID Secure Revocable Hierarchical Identity-Based Encryption Scheme. In Proceedings of the International Conference on Smart Computing and Communication (SmartCom 2016), Shenzhen, China, 17–19 December 2019; pp. 506–515.
31. Xing, Q.; Wang, B.; Wang, X.; Chen, P. Unbounded Revocable Hierarchical Identity-Based Encryption with Adaptive-ID Security. In Proceedings of the IEEE 18th International Conference on High Performance Computing and Communications, Sydney, Austrilia, 12–14 Decmber 2016; pp. 430–437.
32. Xing, Q.; Wang, B.; Wang, X.; Tao, J. Unbounded and Revocable Hierarchical Identity-Based Encryption with Adaptive Security, Decryption Key Exposure Resistant, and Short Public Parameters. *PLoS ONE* **2018**, *13*, e0195204. [CrossRef] [PubMed]
33. Boneh, D.; Goh, E.-J.; Nissim, K. Evaluating 2-DNF Formulas on Ciphertexts. In *Theory of Cryptography—TCC2005*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 325–341.
34. Shi, E.; Waters, B. Delegating Capabilities in Predicate Encryption Systems. In *Automata, Languages and Programming—ICALP2008*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 560–578.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).