Chinese Journal of Electronics
Vol.32, No.3, May 2023

A Novel Construction of Updatable Identity-Based Hash Proof System and Its Applications

QIAO Zirui[1], ZHOU Yanwei[1,2,3,5], YANG Bo[1], ZHANG Wenzheng[3], and ZHANG Mingwu[4]

(1. *School of Computer Science, Shaanxi Normal University, Xi'an 710119, China*)

(2. *State Key Laboratory of Cryptology, Beijing 100878, China*)

(3. *Science and Technology on Communication Security Laboratory, Chengdu 610041, China*)

(4. *School of Computere, Hubei University of Technology, Wuhan 430068, China*)

(5. *Henan Key Laboratory of Network Cryptography Technology, Zhengzhou 450001, China*)

Abstract — In the previous works, to further provide the continuous leakage resilience for the identity-based encryption scheme, a new cryptography primitive, called updatable identity-based hash proof system (U-IB-HPS), was proposed. However, most of the existing constructions have some deficiencies, they either do not have perfect key update function or the corresponding security with tight reduction relies on a non-static complexity assumption. To address the above problems, a new construction of U-IB-HPS is created, and the corresponding security of our system is proved based on the static complexity assumption. Also, the corresponding comparisons and analysis of performances show that our proposal not only achieves the perfect key update function and the anonymity, but also has the tight security reduction. In additional, our proposal achieves the same computational efficiency as other previous systems. To further illustrate the practical function of U-IB-HPS, a generic method of non-interactive data authorization protocol with continuous leakage resilience is designed by employing U-IB-HPS as an underlying tool, which can provide continuous leakage-resilient data authorization function for the cloud computing. Hence, the application field of U-IB-HPS is further extended through our study.

Key words — Updatable identity-based hash proof system, Anonymity, Key update, Leakage resilience.

I. Introduction

In the traditional security model of cryptographic primitives, such as chosen-plaintext attacks (CPA) security and chosen-ciphertext attacks (CCA) security, we usually assume that any adversary can only obtain the corresponding input and output of the cryptographic algorithm, and any leakage information on the internal secret states (such as private key) cannot be captured by the adversary. However, in the actual applications, any adversary can obtain some valuable information on the internal secret states through various leakage attacks, and the traditional security of cryptography primitives will be invalid in the leakage setting. Therefore, cryptographic primitives with leakage resilience are required to meet the security requirements of the actual applications [1]. In the past fews years, the leakage-resilient cryptography has become one of the research hotspots [2], [3], and several cryptographic primitives with leakage resilience were proposed, such as identity-based encryption (IBE) [4]–[6], secret sharing [7], [8], and authenticated key exchange [9].

For the research of leakage-resilient IBE scheme [10]–[12], a common underlying tool, called identity-based hash proof system (IB-HPS), was used. That is, IB-HPS and randomness extractor were employed together to create leakage-resilient IBE. However, in the above proposals, the maximum amount of the additional leakage on the internal secret information can not exceed a parameter set by the cryptography primitive, which is a fixed value, and smaller than the user's private key length. In other words, if the actual leakage length is larger than the system leakage parameter, then the corresponding cryptographic primitive with bounded leakage resilience will not keep its claimed provable security. Hence, in order to further improve

Manuscript Received Nov. 7, 2016; Accepted Mar. 23, 2017. This work was supported by the National Natural Science Foundation of China (62272287, U2001205), the Sichuan Science and Technology Program (2020JDJQ0076), and the Research Funds of Henan Key Laboratory of Network Cryptography Technology (LNCT2021-A04).

 DOI:10.23919/cje.2022.00.203

the leakage resistance of the IBE scheme, it is necessary to provide it with stronger leakage resistance.

In the actual applications, the continuous leakage attacks can be performed, and the adversary can obtain the leakage information with arbitrary length. To address this issue, the continuous leakage resilience is desirable for cryptography primitives [13]. That is, the ability of cryptographic primitives to resist leakage attack can be improved by changing the bounded leakage-resilient ability to continuous leakage resilience. In [11], to generate an IBE scheme that resists continuous leakage attacks, a new basic technology, called updatable identity-based hash proof system (U-IB-HPS), was proposed, which can be used as a basic technology to create the IBE scheme with continuous leakage resilience. Compared with IB-HPS, the above new primitive has an additional algorithm, which is used to refresh user's private key. In other words, the above algorithm outputs a new private key that is indistinguishable, for any adversary, from the original private key. However, the instantiation of IB-HPS is better achieved than that of U-IB-HPS, because, in a U-IB-HPS, two challenges need to be addressed: 1) how to keep the valid/invalid ciphertexts indistinguishability when the user's private key is updated; 2) how to achieve the smoothness property when the user's private key is updated. In order to solve the above challenges, Zhou *et al.* [11] propose the concrete construction of U-IB-HPS by employing the orthogonality of two vectors, in which, the user's private key is written as a vector, and the key update algorithm uses random vectors to update it.

Related works The definition of IB-HPS was proposed in [12], and a CPA secure leakage-resilient IBE scheme can be generated from an IB-HPS and an average-case strong randomness extractor. Based on the above conclusions, Chow *et al.* [10] proposed three concrete constructions of IB-HPS from the classic method used to create IBE scheme. However, for the third IB-HPS of [10], in the formal security proof, the challenge identity will be guessed by the simulator, which is correct with probability $1/L$, where L is the maximum number of identities submitted by the adversary in the security game. That is, for the security proof of the above system, the ability of adversary used to attack the security of IB-HPS cannot be completely translated into the challenger's ability to solve the corresponding hard problem, so there is a partial loss of the adversary's ability. Also, the formal proof for the second instance was omitted in [10]. Furthermore, if the queried identity is equal to the master secret key, then the corresponding IB-HPS designed by Alwen *et al.* [12] will abort in the key generation algorithm.

Based on the conclusions of [13], we have that if the user's private key can be refreshed without changing the functionality and the public parameter of cryptography scheme, then a cryptography scheme with bounded leakage resilience can resist the continuous leakage attacks through key update operation. That is, the function of cryptography scheme is changeless even if the user's private key is refreshed many times.

Naturally, we are able to achieve the design goals of the IBE scheme with continuous leakage resilience by providing key update functionality for the underlying IB-HPS. Hence, the notion of U-IB-HPS was proposed with four concrete instantiations in [11]. To reduce the length of user's private key of [11], a novel instantiation of U-IB-HPS was built in [14], which can be used to construct anonymous IBE schemes. However, in the key update operation of [14], we have to choose a random value from a set $\mathbf{ker}^*(tk_{id}^1, tk_{id}^2)$ (where this value can make two non-orthogonal vectors tk_{id}^1 and tk_{id}^2 are orthogonal) to refresh private key $sk_{id} = (sk_{id,1}, sk_{id,2}, sk_{id,3})$. However, the set $\mathbf{ker}^*(tk_{id}^1, tk_{id}^2)$ only consists of the multiples of prime p (p is the prime order of group), in this case, an adversary can obtain the original element $sk_{id,3}$ of private key from the updatable element $sk'_{id,3}$ if $sk'_{id,3}$ was captured by the adversary. Also, a user needs to store the update trapdoor, which increases the storage burden for users and decreases the storage efficiency of the above scheme.

Although, the concrete construction of U-IB-HPS proposed in [14] is novel, it is not perfect, in which, an adversary can recover the original key element from the updated one when the updated private key is completely captured. This is because the randomness used to update private key comes from a set which consists of the multiples of prime p. In other words, if the adversary can capture the third element of the updated private key, he/she can recover the original element corresponding to it. Although the exposure of the third element will not affect the security of the corresponding scheme, there may still exist some potential security risks. To sum up, in [14], the update algorithm only updates two elements $sk_{id,1}$ and $sk_{id,2}$ of the user's private key $sk_{id} = (sk_{id,1}, sk_{id,2}, sk_{id,3})$, which is flawed for a perfect update operation. An improved construction of IB-HPS was designed in [15], and the length of system public parameters is long, resulting in low storage efficiency. Furthermore, several constructions of IB-HPS based on lattices were created in [16], [17].

As discussed above, we have that U-IB-HPS is a practical foundation tool for creating identity-based cryptographic primitives with continuous leakage resilience, such as IBE, hybrid encryption, and key exchange. However, the previous constructions of U-IB-HPS either did not have the perfect key update opera-

tion or were created based on a non-static complexity assumptions (the security is dependent on the number of queries made by the adversary), for example, q-type assumption. Therefore, it is a hot research issue to construct U-IB-HPS with perfect key updating function based on static security assumption.

Our motivations U-IB-HPS is a useful tool, and the proposal with better performance can be used to create many cryptographic schemes with continuous leakage resilience. However, for the previous concrete constructs of U-IB-HPS, some elements of the private key are only updated, do not have perfect key update capabilities, and some constructs are created based on a non-static security assumptions. From the practical application, U-IB-HPS not only updates all elements of the private key, but also obtains the corresponding security from the static security assumption. Therefore, in this paper, we will focus on the study of key update operation, after that, a novel concrete construction of U-IB-HPS is designed to achieve the better performance, also, the security of our proposal is proved based on the hardness of classic static security assumption.

U-IB-HPS can be employed as a key encapsulation mechanism to create a session key in network communication. Hence, to further improve usability, we will add the corresponding security properties for U-IB-HPS according to the real requirements of the actual application. For example, the network users do not want their identity information to be obtained by unauthorized users, so it is necessary to protect users' privacy information (such as identity and location) in the network environment. Hence, anonymity of identity is not only an important security requirement in the Internet, but also may influence user's behaviors. The anonymity of identity is a necessary security property of network protocol, and we will provide anonymity function for U-IB-HPS, in this case, the corresponding protocol created based on this tool will naturally inherit anonymity. Therefore, in this paper, we provide a general method to design anonymous security protocols.

Furthermore, with the development of network communication technology, more and more data is stored in the cloud servers. In this case, the data authorization is a key technology to achieve secure data access control. In Fig.1, to achieve secure data storage, the data owner usually encrypts the data and stores it in the cloud server. However, to ensure the availability of data, the data owner and the data user usually need to create a shared session key. That is, a session key used to encrypt the shared data will be generated between the data owner and the data user through interactive exchange. However, the above traditional method is inefficient and not practical for mobile terminals, because the key exchange process will consume a lot of computing power of mobile devices (in fact, the computing resources of mobile terminals are limited). To obtain an efficient data authorization method, we will study the non-interactive (one-round) data authorization protocol that is resistant to leakage attacks, has high computing efficiency, and is suitable for employed in mobile devices. In other words, our conclusions will expand the application of U-IB-HPS.

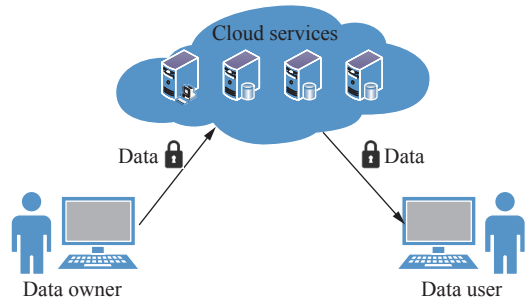

Fig. 1. Data storage in the cloud.

Our contributions In order to further obtain a U-IB-HPS with better performances over the bilinear groups, a concrete construction of U-IB-HPS is proposed in this paper, the security of our proposed system is proved in the selective identity security model, also, our proposal has several advantages over the existing related constructions of U-IB-HPS, such as anonymity and perfect key update. To sum up, our contributions are described as follows:

1) We create an instantiation of U-IB-HPS with perfect key update operation, in which, for any adversary, there is no connection between the updated private key and the original private key. Also, the security of our proposal is proved, in the selective idetity security model, from the static complexity assumption, i.e., decisional bilinear Diffie-Hellman (DBDH) assumption, where the security is not dependent on the number of query submitted by the adversary.

2) Compared with the previous works, our system not only realizes the perfect key update function, but also has the anonymity. More importantly, the tight security reduction is achieved in our construction, that is, in the security proof, the adversary's ability to attack the security of U-IB-HPS is completely transformed into a solution for the hardness of DBDH problem.

3) To further obtain secure data authorization with leakage resilience in cloud computing, a generic method of non-interactive leakage-resilient data authorization protocol is created from U-IB-HPS, which can efficiently realize the secure storage of data in cloud computing.

We have to stress that, our proposal is constructed based on the static complexity assumptions (the se-

curity does not depends on the number of queries submitted by the adversary) under the selective identity security model. Naturally, the reader will have the following question:

Can we construct a U-IB-HPS based on a static assumption in the adaptive security model?

Generally speaking, IBE and (U-)IB-HPS have the same construction method, so we firstly analyze the generation methods of IBE scheme, after that, discusses the corresponding hard prolem in designing IBE scheme.

Waters IBE scheme [18] has adaptive security from the static complexity assumption, however, for the challenge identity, the corresponding private key cannot be generated by the challenger, and the above scheme cannot obtain the tight security proof. Without loss of generality, the corresponding method employed to construct IBE scheme can also be used to create IB-HPS/U-IB-HPS, however, the above method of constructing Waters IBE scheme [18] cannot be used to create IB-HPS/U-IB-HPS, because, in the above method, the challenger cannot generate the challenge identity's private key, however, from the viewpoint of the challenger of IB-HPS/U-IB-HPS, he/she needs to respond the key generation query for any identity including the challenge identity. In other words, if we can design an adaptively secure IB-HPS/U-IB-HPS based on the classic static assumption, then we also can creat an IBE scheme with tightly adaptive security from the same complexity assumption. However, how to create a concrete construction of the above IBE scheme is a known open hard problem in identity-based cryptography. Naturally, we can definitely answer that the above question is also an open hard problem. Therefore, for U-IB-HPS, the adaptive security over the static assumption can only be obtained in the selective identity security model, because the corresponding private key of the challenge identity can be created in this model. Hence, in this paper, the corresponding security of our system will be proved in the above model.

II. Preliminaries

Let $\kappa \in \mathbb{N}$ be the security parameter. We use $s \leftarrow_R S$ to denote that s is uniformly at random picked from S. Also, $\mathsf{negl}(\kappa)$ denotes the negligible value over κ.

Lower case letters (e.g., $\vec{a}$) is used to denote vectors. If any two vectors $\vec{a} = (a_1, \ldots, a_n)$ and $\vec{b} = (b_1, \ldots, b_n)$ are orthogonal, then we have $\langle \vec{a}, \vec{b} \rangle = \sum_{i=1}^{n} a_i b_i = 0$.

For any $t \in \mathbb{Z}_q^*$ and $\vec{a}$, $\vec{b}$, $\vec{c} \in (\mathbb{Z}_q)^n$, we have $(g^{\vec{a}})^{\vec{b}} = g^{\langle \vec{a}, \vec{b} \rangle}$ and $g^{t\langle \vec{a}, \vec{b} \rangle} = g^{\langle t\vec{a}, \vec{b} \rangle} = g^{\langle \vec{a}, t\vec{b} \rangle}$.

Furthermore, in our construction, let $\mathcal{SK}$ be the user's private key space, $\mathcal{C}$ the ciphertext space, $\mathcal{ID}$ the user's identity space, and $\mathcal{K}$ the encapsulated-key space.

1. Complexity assumption

Definition 1 (Decisional bilinear Diffie-Hellman, DBDH) For any two random tuples $T_1 = (g, g^a, g^b, g^c, e(g,g)^{abc})$ and $T_0 = (g, g^a, g^b, g^c, e(g,g)^{\omega})$ (where $a, b, c, \omega \leftarrow_R \mathbb{Z}_q^*$), if DBDH assumption holds, then the corresponding advantage $\mathsf{Adv}_{\mathcal{G},\mathcal{A}}^{DBDH}(\kappa) = |\Pr[\mathcal{A}(T_1) = 1] - \Pr[\mathcal{A}(T_0) = 1]|$ of any probabilistic polynomial time adversary $\mathcal{A}$ is negligible. That is, from the viewpoint of $\mathcal{A}$, T_1 and T_0 are indistinguishable.

2. IB-HPS and U-IB-HPS

An IB-HPS consists Setup, KeyGen, Encap, Encap^* and Decap five algorithms, where, the setup algorithm Setup is used to generate system public parameter mpk and master secret key msk, and the key generation algorithm KeyGen is employed to create private key with the corresponding identity as input. Furthermore, two encapsulation algorithms Encap and Encap^* output valid and invalid encapsulation ciphertext, and the decapsulation algorithm Decap returns the decapsulated result for the inputted ciphertext. Notice that, for any input ciphertext, Decap will output a decapsulation result even if the corresponding input is an invalid encapsulated ciphertext. The formal definition and the security properties of IB-HPS were described in [10], [12], and an IB-HPS meets correctness, universality, smoothness and valid/invalid ciphertext indistinguishability, etc. We refer the reader to [10], [12] for details.

In U-IB-HPS, user's private key can be refreshed by an additional algorithm Update, which outputs a new private key. Also, the original key and the updated key are indistinguishable from the viewpoint of any probabilistic polynomial time adversary. More importantly, the function and the public parameters of U-IB-HPS are unchanged even if the user's private key has been updated several times. Compared with IB-HPS, U-IB-HPS adds an algorithm Update to achieve key update function. For a U-IB-HPS, the corresponding formal definition and security properties were introduced in [11], [14]. We refer the reader to [11], [14] for details. In addition to satisfying the above security properties, such as correctness, universality, smoothness, and valid/invalid ciphertext indistinguishability, U-IB-HPS also needs to satisfy re-randomization and invariance of update. The differences and connections between IB-HPS and U-IB-HPS are shown in Fig.2.

As described above, U-IB-HPS is proposed from IB-HPS by adding an additional key update algorithm. However, two additional security properties such as re-randomization and invariance of update need to be considered in U-IB-HPS. From the application, U-IB-HPS can be used to construct a continuous leakage resilient

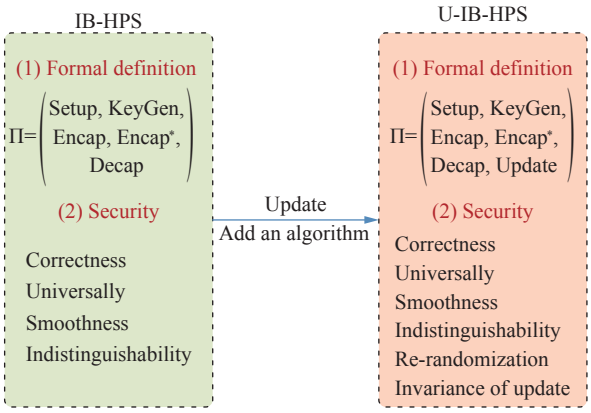

Fig. 2. The differences and connections between IB-HPS and U-IB-HPS.

IBE scheme, while IB-HPS only achieves resistance to bounded leakage attacks. Specially, U-IB-HPS can periodically update the private key, so that the leakage information of the past private key is not effective for the updated one. Therefore, the adversary needs to collect the leakage information about the new private key. Therefore, the key update algorithm can update the private key without changing the public information, and can assist the corresponding IBE scheme to achieve continuous leakage resilience.

3. Anonymity

In this part, we take IB-HPS as an example to discuss its anonymity, and the anonymity of U-IB-HPS is similar.

In IB-HPS, the anonymity of identity denotes that the corresponding invalid ciphertexts corresponding to two identities are indistinguishable, even if the adversary can obtain the complete private key of any identity (including the above two challenge identities). In fact, in IB-HPS and U-IB-HPS, the anonymity of identity also shows that the corresponding invalid ciphertext is indistinguishable. Hence, for an IB-HPS $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encap}, \mathsf{Encap}^*, \mathsf{Decap})$, the anonymity of identity can be defined through the following experiment $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{Anonymity}}(\kappa)$, where $\mathcal{A}$ is an adversary that employed to break the anonymity of IB-HPS.

$\mathsf{Exp}_{\mathcal{A}}^{\mathsf{Anonymity}}(\kappa)$:

1) $(mpk, msk) \leftarrow \mathsf{Setup}(1^\kappa)$;

2) $(id_0^*, id_1^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KeyGen}}(\cdot)}(mpk)$;

3) $C_0^* \leftarrow \mathsf{Encap}^*(id_0^*)$ and $C_1^* \leftarrow \mathsf{Encap}^*(id_1^*)$;

4) $v' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KeyGen}}(\cdot)}(mpk, C_v^*)$, where $v \leftarrow \{0,1\}$;

5) If $v' = v$, then the adversary $\mathcal{A}$ wins.

Let $\mathcal{O}_{\mathsf{KeyGen}}(\cdot)$ be the key generation oracle, for an identity id_i, the corresponding private key sk_{id_i} can be obtained by $\mathcal{A}$ through querying $\mathcal{O}_{\mathsf{KeyGen}}(\cdot)$ with id_i as input. Notice that, $\mathcal{A}$ can obtain the private key of any identity including id_0^* and id_1^*. Furthermore, the anonymity discussed in this paper refers to the anonymity of identity.

Therefore, the advantage of probabilistic polynomial time adversary $\mathcal{A}$ is defined as $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Anonymity}}(\kappa) = \left| \Pr[v' = v] - \frac{1}{2} \right|$.

Definition 2 For any probabilistic polynomial time adversary $\mathcal{A}$, in the above experiment $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{Anonymity}}(\kappa)$, if the advantage $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Anonymity}}(\kappa)$ is negligible, then the corresponding IB-HPS has anonymity.

Notice that, for IB-HPS, the anonymity means that the invalid encapsulation ciphertext does not reveal any information on the corresponding identity.

III. Anonymous IB-HPS

To further show our underlying key technology, we firstly design a new instantiation of IB-HPS, which has several advantages over previous systems. After that, we will create a novel construction of U-IB-HPS from this basic IB-HPS in the next section.

1. Constructions

Our basic construction of the IB-HPS $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encap}, \mathsf{Encap}^*, \mathsf{Decap})$ consists of the following five algorithms:

1) $(mpk, msk) \leftarrow \mathsf{Setup}(1^\kappa)$

Run group generation algorithm $\mathcal{G}(1^\kappa)$ to obtain a tuple $\mathbb{G} = (q, G, G_T, e(\cdot,\cdot), g)$, where G is a group of order large prime q, g is a generator of G, and $e : G \times G \to G_T$ is an efficiently computable bilinear map.

Choose $\alpha \leftarrow_R \mathbb{Z}_q^*$ and $g_2, g_3, u, h \leftarrow_R G$, after that, compute $g_1 = g^\alpha$.

Set $mpk = (\mathbb{G}, g_1, g_2, g_3, u, h)$ and $msk = \alpha$, where mpk is also the common input of the following algorithms.

2) $sk_{id} \leftarrow \mathsf{KeyGen}(id, msk)$

Choose $r, t \leftarrow_R \mathbb{Z}_q^*$, and compute $d_1 = g_2^{\alpha} g_3^{\alpha id} \left(u^{id} h\right)^{-r}$, $d_2 = g^r$ and $d_3 = -t$.

Output $sk_{id} = (d_1, d_2, d_3)$.

3) $(C, k) \leftarrow \mathsf{Encap}(id)$ Choose $\beta \leftarrow_R \mathbb{Z}_q^*$, and compute $c_1 = g^\beta$, $c_2 = \left(u^{id} h\right)^\beta$ and $c_3 = e(g_1, g_2)^\beta$.

Output the valid encapsulation ciphertext $C = (c_1, c_2, c_3)$ and the encapsulated-key $k = e(g_1, g_3^{id})^\beta$.

Specially, $e(g_1, g_2)$ and $e(g_1, g_3)$ can be included in mpk, in this case, the encapsulation algorithm does not require any pairing computation.

4) $C \leftarrow \mathsf{Encap}^*(id)$

Choose $\beta, \beta^* \leftarrow_R \mathbb{Z}_q^*$ such that $\beta \neq \beta^*$, and compute $c_1 = g^\beta$, $c_2 = \left(u^{id} h\right)^\beta$ and $c_3 = e(g_1, g_2)^{\beta^*}$.

Output $C = (c_1, c_2, c_3)$ as the invalid encapsulation ciphertext.

5) $k' \leftarrow \mathsf{Decap}(sk_{id}, C)$

Output $k' = e(c_1, d_1)e(c_2, d_2)c_3^{d_3}$ as the corresponding decapsulated result.

2. Correctness

The following equation shows that our system has perfect correctness.

$$\begin{aligned} k' &= e(c_1, d_1)e(c_2, d_2)c_3^{d_3} \\ &= e(g^{\beta}, g_2^{\alpha t} g_3^{\alpha id}(u^{id}h)^{-r})e((u^{id}h)^{\beta}, g^r)e(g_1, g_2)^{-\beta t} \\ &= e(g^{\beta}, g_2^{\alpha})^t e(g^{\beta}, g_3^{\alpha id})e(g^{\beta}, (u^{id}h)^{-r})e((u^{id}h)^{\beta}, g^r) \\ &\quad \times e(g_1, g_2)^{-\beta t} \\ &= e(g_1, g_3^{id})^{\beta} \end{aligned}$$

where $g_1 = g^{\alpha}$.

According to the above equation, the decapsulated result k' outputted by the decapsulation algorithm Decap with valid ciphertext as input is consistent with the encapsulated-key k generated by the valid encapsulation algorithm Encap.

3. Security

In this part, the corresponding security (such as smoothness and universality) of our system will be shown. To make it easy to distinguish, in this paper, the invalid encapsulation ciphertext will be written as $C^* = (c_1^*, c_2^*, c_3^*)$.

1) Smoothness

In IB-HPS, the smoothness means that the result of decapsulation algorithm for an invalid encapsulation ciphertext is a uniform random value over the encapsulated-key space $\mathcal{K}$.

Let id be an identity over identity space $\mathcal{ID}$, then, for any $sk_{id} = (d_1, d_2, d_3) = (g_2^{\alpha t} g_3^{\alpha id}(u^{id}h)^{-r}, g^r, -t)$ and $C^* = (c_1^*, c_2^*, c_3^*) = (g^{\beta}, (u^{id}h)^{\beta}, e(g_1, g_2)^{\beta^*})$ generated by KeyGen and Encap with id as input, we have

$$\begin{aligned} k' &= \mathsf{Decap}(sk_{id}, C^*) \\ &= e(c_1^*, d_1)e(c_2^*, d_2)(c_3^*)^{d_3} \\ &= e(g^{\beta}, g_2^{\alpha t} g_3^{\alpha id}(u^{id}h)^{-r})e((u^{id}h)^{\beta}, g^r)e(g_1, g_2)^{-\beta^* t} \\ &= e(g^{\beta}, g_2^{\alpha})^t e(g^{\beta}, g_3^{\alpha id})e(g^{\beta}, (u^{id}h)^{-r})e((u^{id}h)^{\beta}, g^r) \\ &\quad \times e(g_1, g_2)^{-\beta^* t} \\ &= e(g_1, g_3^{id})^{\beta} e(g_1, g_2)^{t(\beta - \beta^*)} \end{aligned}$$

where $t, \beta^*, \beta \leftarrow_R \mathbb{Z}_q^*$ and $\beta^* \neq \beta$. Hence, for any C^*, the decapsulation result k' is a random value over the encapsulation key space $\mathcal{K} = G_T$, because t, β^*, β are randomly chosen from $\mathbb{Z}_q^*$.

2) Universality

In IB-HPS, the universality means that the different private keys corresponding to the same identity can obtain different decapsulation results when decapsulating an invalid encapsulation ciphertext.

For any invalid encapsulation ciphertext C^*, based on the smoothness, we have $k' = e(g_1, g_3^{id})^{\beta} e(g_1, g_2)^{t(\beta - \beta^*)}$, where t is a random number inherited from the corresponding private key $sk_{id} = (d_1, d_2, d_3) = \left(g_2^{\alpha t} g_3^{\alpha id}(u^{id}h)^{-r}, g^r, -t\right)$. Therefore, for any two private key $sk_{id} \neq sk'_{id}$ of the same identity (it means that $t \neq t'$), we have $\mathsf{Decap}(C^*, sk_{id}) \neq \mathsf{Decap}(C^*, sk'_{id})$.

Notice that, the key generation is a randomization algorithm, and the different private keys for the same identity are created with various random values.

3) Anonymity

In IB-HPS, the anonymity means that invalid encapsulation ciphertexts generated by different identities are indistinguishable, even if the adversary can obtain the private key of the corresponding identities.

For any $id_1, id_2 \in \mathcal{ID}$, $id_1 \neq id_2$, as well as for their private keys $sk_{id_1} = (d_1^1, d_2^1, d_3^1) = (g_2^{\alpha t_1} g_3^{\alpha id_1}(u^{id_1}h)^{-r_1}, g^{r_1}, -t_1)$ and $sk_{id_2} = (d_1^2, d_2^2, d_3^2) = (g_2^{\alpha t_2} g_3^{\alpha id_2}(u^{id_2}h)^{-r_2}, g^{r_2}, -t_2)$, where

$$sk_{id_1} = \mathsf{KeyGen}(msk, id_1),\ sk_{id_2} = \mathsf{KeyGen}(msk, id_2)$$

Let $C_1^* = (c_1^{1^*}, c_2^{1^*}, c_3^{1^*}) = \left(g^{\beta_1}, (u^{id_1}h)^{\beta_1}, e(g_1, g_2)^{\beta_1^*}\right)$ and $C_2^* = (c_1^{2^*}, c_2^{2^*}, c_3^{2^*}) = \left(g^{\beta_2}, (u^{id_2}h)^{\beta_2}, e(g_1, g_2)^{\beta_2^*}\right)$ be two invalid encapsulation ciphertexts by running Encap^* with id_1 and id_2 as input, i.e., $C_1^* = \mathsf{Encap}^*(id_1)$, $C_2^* = \mathsf{Encap}^*(id_2)$.

Then, we can obtain

$$\begin{aligned} k'_1 &= \mathsf{Decap}(C_1^*, sk_{id_1}) = e(g_1, g_3^{id_1})^{\beta_1} e(g_1, g_2)^{t_1(\beta_1 - \beta_1^*)} \\ k'_2 &= \mathsf{Decap}(C_2^*, sk_{id_2}) = e(g_1, g_3^{id_2})^{\beta_2} e(g_1, g_2)^{t_2(\beta_2 - \beta_2^*)} \end{aligned}$$

where $\beta_1, \beta_2, \beta_1^*, \beta_2^*, t_1, t_2 \leftarrow_R \mathbb{Z}_q^*$ and $t_1 \neq t_2$, $\beta_1 \neq \beta_2$, $\beta_1^* \neq \beta_2^*$.

As discussed above, (C_1, k'_1) and (C_2, k'_2) are two independent random tuples over $\mathcal{C} \times \mathcal{K}$. Therefore, for any adversary, the corresponding invalid encapsulation ciphertexts generated by two identities are indistinguishable, even if the complete private keys of the above two identities are captured by the adversary.

4) Indistinguishability of valid/invalid ciphertext

In IB-HPS, the indistinguishability of valid/invalid ciphertext means that the valid encapsulation ciphertext and invalid encapsulation ciphertext corresponding to an identity are indistinguishable, even if the adversary can capture the corresponding private key of the above identity.

Now, we will prove the indistinguishability of valid/invalid ciphertext of our proposal through the following theorem.

Theorem 1 If there exists a probabilistic polynomial time adversary $\mathcal{A}$ who can break the indistinguishability of valid/invalid ciphertext of our basic IB-HPS Π, then there exist a simulator $\mathcal{S}$ who can solve

DBDH assumption.

Proof Before the game starts, a challenge tuple (g, g^a, g^b, g^c, T_v) and the corresponding public tuple $\mathbb{G} = (q, G, G_T, e(\cdot,\cdot), g)$ will be received by $\mathcal{S}$ from the corresponding challenger of DBDH assumption, where $T_v = e(g,g)^{abc}$ or $T_v \leftarrow G_T$ (in this case, T_v can be written as $T_v = e(g,g)^{abc^*}$ and $c^* \neq c$). Before the system is built, an identity id^* will be obtained by $\mathcal{S}$ from $\mathcal{A}$, which is employed as challenge identity.

The interaction between $\mathcal{S}$ and $\mathcal{A}$ is described below:

– **Setup** $\mathcal{S}$ sets $g_1 = g^a$ (implicitly set $msk = a$) and $g_2 = g^b$. After that, chooses $m, n, z \leftarrow_R \mathbb{Z}_q^*$, and computes $g_3 = g^m$, $u = g_1^z$ and $h = u^{-id^*} g^n$.

It sends $mpk = (\mathbb{G}, g_1, g_2, g_3, u, h)$ to $\mathcal{A}$, where the simulation game is indistinguishable from the actual construction because the above parameters are random values from the view point of $\mathcal{A}$.

– **Test stage 1** The complete private keys for any identity id (including id^*) will be generated by $\mathcal{S}$.

i) For $id \neq id^*$, $\mathcal{S}$ chooses $t, r \leftarrow_R \mathbb{Z}_q^*$, and outputs

$$sk_{id} = \left(g_1^{mid} g_2^{\frac{-nt}{z(id-id^*)}} (u^{id} h)^{-r}, g^r g_2^{\frac{t}{z(id-id^*)}}, -t\right)$$

Let $r' = r + \frac{bt}{z(id-id^*)}$, where r' is a random value over $\mathbb{Z}_q^*$, because $r, t \leftarrow_R \mathbb{Z}_q^*$. Hence, we have

$$\begin{aligned}
&g_1^{mid} g_2^{\frac{-nt}{z(id-id^*)}} (u^{id} h)^{-r} \\
&= g_3^{aid} g^{\frac{-nbt}{z(id-id^*)}} (u^{id} h)^{-r' + \frac{bt}{z(id-id^*)}} \\
&= g_3^{aid} g^{\frac{-nbt}{z(id-id^*)}} (g_1^{z(id-id^*)} g^n)^{\frac{bt}{z(id-id^*)}} (u^{id} h)^{-r'} \\
&= g_2^{at} g_3^{aid} (u^{id} h)^{-r'} \\
&g^r g_2^{\frac{t}{z(id-id^*)}} = g^r g^{\frac{bt}{z(id-id^*)}} = g^{r + \frac{bt}{z(id-id^*)}} = g^{r'}
\end{aligned}$$

where $u^{id} h = u^{id-id^*} g^n = g_1^{z(id-id^*)} g^n$. Based on the above argument, we have that sk_{id} is a valid private key.

ii) For $id = id^*$, $\mathcal{S}$ chooses $t^*, r^* \leftarrow_R \mathbb{Z}_q^*$, and outputs

$$sk_{id^*} = \left(g_1^{mid^*} g_2^{\frac{-nt^*}{q}} \left(u^{id^*} h\right)^{-r^*}, g^{r^*} g_2^{\frac{t^*}{q}}, -t^*\right)$$

Let $\hat{r}^* = r^* + \frac{bt^*}{q}$, where $\hat{r}^*$ is a random value over $\mathbb{Z}_q^*$ because $r^*, t^* \leftarrow_R \mathbb{Z}_q^*$. Hence, we have

$$\begin{aligned}
&g_1^{mid^*} g_2^{\frac{-nt^*}{q}} \left(u^{id^*} h\right)^{-r^*} \\
&= g_3^{a \cdot id^*} g^{\frac{-nbt^*}{q}} \left(u^{id^*} h\right)^{-\hat{r}^* + \frac{bt^*}{q}} \\
&= g_3^{a \cdot id^*} g^{\frac{-nbt^*}{q}} \left(g_1^q g^n\right)^{\frac{bt^*}{q}} \left(u^{id^*} h\right)^{-\hat{r}^*} \\
&= g_2^{at^*} g_3^{a \cdot id} \left(u^{id^*} h\right)^{-\hat{r}^*} \\
&g^{r^*} g_2^{\frac{t^*}{q}} = g^{r^*} g^{\frac{bt^*}{q}} = g^{r^* + \frac{bt^*}{q}} = g^{\hat{r}^*}
\end{aligned}$$

where $u^{id^*} h = g^n$ and $g_1^q = 1$. Based on the above argument, we have that sk_{id^*} is also a valid private key.

– **Challenge stage** $\mathcal{S}$ returns $C' = (c_1', c_2', c_3') = (g^c, g^{cn}, T_v)$ as the challenge ciphertext to $\mathcal{A}$. Notice that, if $T_v = e(g,g)^{abc}$, then C' can be written as $C' = (c_1', c_2', c_3') = (g^c, (u^{id^*} h)^c, (g_1, g_2)^c)$ (where $(u^{id^*} h) = g^n$), and C' is a valid ciphertext. Otherwise, $T_v = e(g,g)^{abc^*}$, and C' can be written as $C' = (c_1', c_2', c_3') = (g^c, (u^{id^*} h)^c, (g_1, g_2)^{c^*})$ (where $c \neq c^*$), and C' is an invalid ciphertext.

– **Test stage 2** At this stage, $\mathcal{A}$ can also obtain the corresponding private key of any identity from $\mathcal{S}$. Specially, in both test stages, $\mathcal{S}$'s response to the same identity by using the same private key.

– **Output** Finally, $\mathcal{A}$ returns the guess v', and $\mathcal{S}$ outputs v' received from $\mathcal{A}$.

Therefore, we can obtain that, in our construction Π, if $\mathcal{A}$ can break the valid/invalid indistinguishability with a obvious advantage $\mathsf{Adv}_{\mathcal{A},\Pi}^{\text{VI-IND}}(\kappa)$, then the corresponding simulator $\mathcal{S}$ can output a correct guess of DBDH assumption with a non-negligible advantage $\mathsf{Adv}_{\mathcal{S}}^{DBDH}(\kappa) \approx \mathsf{Adv}_{\mathcal{A},\Pi}^{\text{VI-IND}}(\kappa)$.

Specially, in the simulation game, $\mathcal{S}$ does not abort. Thus, our security proof has tight reduction. That is, we completely transform the ability of $\mathcal{A}$ to attack the indistinguishability of valid/invalid encapsulation ciphertext into the ability of $\mathcal{S}$ to solve DBDH hard problem.

IV. A New Construction of Anonymous U-IB-HPS

In this part, a new construction of U-IB-HPS is generated from the above basic IB-HPS, and the key update function is achieved by running additional algorithm Update'.

1. Construction

Our new construction of U-IB-HPS $\Pi' = (\mathsf{Setup}', \mathsf{KeyGen}', \mathsf{Update}', \mathsf{Encap}', \mathsf{Encap}'^*, \mathsf{Decap}')$ consists of the following six algorithms.

1) $(mpk, msk) \leftarrow \mathsf{Setup}'(1^\kappa)$

Run $\mathbb{G} = (q, G, G_T, e(\cdot,\cdot), g) \leftarrow \mathcal{G}(1^\kappa)$. Choose $\alpha \leftarrow_R \mathbb{Z}_q^*$ and $g_3, u, h \leftarrow_R G$, and set $g_1 = g^\alpha$.

Let $n \in \mathbb{N}$ be the length of vector. Choose $\vec{a} \leftarrow_R (\mathbb{Z}_q)^n$, and compute $g_2 = g^{\vec{a}}$. To hide $\vec{a}$, choose $\eta \leftarrow_R \mathbb{Z}_q^*$, and compute $\vec{\beta} = \eta \vec{a}$. That is, we use random number η to hide the vector $\vec{a}$.

Set $mpk = \langle \mathbb{G}, g_1, g_2, g_3, u, h, \vec{\beta}, n \rangle$ as the public parameters and $msk = \alpha$ as the master secret key.

2) $sk_{id} \leftarrow \mathsf{KeyGen}'(id, msk)$

Choose $r \leftarrow_R \mathbb{Z}_q^*$, $\vec{t} \leftarrow_R (\mathbb{Z}_q)^n$ and $\langle \vec{\beta}, \vec{t} \rangle = 0$ (it means that $\langle \vec{a}, \vec{t} \rangle = 0$), compute

$$d_1 = g_2^{-\alpha\vec{t}} g_3^{\alpha id}(u^{id}h)^{-r},\ \ d_2 = g^r,\ \ d_3 = \vec{t}$$

Output $sk_{id} = (d_1, d_2, d_3)$.

3) $sk_{id}^j \leftarrow \mathsf{Update}'(sk_{id}^{j-1})$

Choose $r_j \leftarrow_R \mathbb{Z}_q^*$, $\vec{t}_j \leftarrow_R (\mathbb{Z}_q)^n$ and $\langle\vec{\beta}, \vec{t}_j\rangle = 0$ (it means that $\langle\vec{a}, \vec{t}_j\rangle = 0$), compute

$$d_1^j = d_1^{j-1}(u^{id}h)^{-r_j},\ \ d_2^j = d_2^{j-1} g^{r_j},$$
$$d_3^j = d_3^{j-1} + \vec{t}_j$$

Thus, for any $j \in \mathbb{N}$, we have

$$d_1^j = g_2^{-\alpha\vec{t}} g_3^{\alpha id}(u^{id}h)^{-(r+\sum_{i=1}^j r_i)},$$
$$d_2^j = g^{r+\sum_{i=1}^j r_i},\quad d_3^j = \vec{t} + \sum_{i=1}^j \vec{t}_j$$

Output $sk_{id}^j = (d_1^j, d_2^j, d_3^j)$.

In the above algorithm, all of elements of updated private key are random from the view point of adversary. Hence, our proposal has perfect key update function.

4) $(C, k) \leftarrow \mathsf{Encap}'(id)$

Choose $\gamma \leftarrow_R \mathbb{Z}_q^*$, and compute

$$c_1 = g^\gamma,\ \ c_2 = (u^{id}h)^\gamma,\ \ c_3 = e(g_1, g_2)^\gamma$$

Output $C = (c_1, c_2, c_3)$ and $k = e(g_1, g_3^{id})^\gamma$.

5) $C \leftarrow \mathsf{Encap}'^*(id)$

Choose $\gamma, \gamma^* \leftarrow_R \mathbb{Z}_q^*$ and $\gamma \neq \gamma^*$, compute

$$c_1^* = g^\gamma,\ \ c_2^* = (u^{id}h)^\gamma,\ \ c_3^* = e(g_1, g_2)^{\gamma^*}$$

Output $C^* = (c_1^*, c_2^*, c_3^*)$.

6) $k \leftarrow \mathsf{Decap}'(sk_{id}, C)$

Compute $k' = e(c_1, d_1)e(c_2, d_2)c_3^{d_3}$, and output k as the decapsulation result.

2. Correctness

The following equations shows that our new construction has perfect correctness.

For the original private key $sk_{id} = (d_1, d_2, d_3)$, we can obtain

$$\begin{aligned}
k &= \mathsf{Decap}'(sk_{id}, C)\\
&= e(c_1, d_1)e(c_2, d_2)c_3^{d_3}\\
&= e(g^\gamma, g_2^{-\alpha\vec{t}} g_3^{\alpha id}(u^{id}h)^{-r})e((u^{id}h)^\gamma, g^r)e(g_1, g_2)^{\gamma\vec{t}}\\
&= e(g^\gamma, g_2^{-\alpha\vec{t}})e(g^\gamma, g_3^{\alpha\cdot id})e(g^\gamma, (u^{id}h)^{-r})e((u^{id}h)^\gamma, g^r)\\
&\quad \times e(g_1, g_2)^{\gamma\vec{t}}\\
&= e(g_1, g_3^{id})^\gamma
\end{aligned}$$

And for the updated private key $sk_{id}^j = (d_1^j, d_2^j, d_3^j)$, we have

$$\begin{aligned}
k &= \mathsf{Decap}'(sk_{id}^j, C)\\
&= e(c_1, d_1^j)e(c_2, d_2^j)c_3^{d_3^j}\\
&= e(g^\gamma, g_2^{-\alpha\vec{t}} g_3^{\alpha id}(u^{id}h)^{-(r+\sum_{i=1}^j r_i)})\\
&\quad \times e((u^{id}h)^\gamma, g^{r+\sum_{i=1}^j r_i})e(g_1, g_2)^{\gamma(\vec{t}+\sum_{i=1}^j \vec{t}_i)}\\
&= e(g^\gamma, g_2^{-\alpha\vec{t}})e(g^\gamma, g_3^{\alpha\cdot id})e(g_1, g_2)^{\gamma\vec{t}}\\
&\quad \times e(g_1, g_2)^{\gamma\vec{t}_1} \ldots e(g_1, g_2)^{\gamma\vec{t}_j}\\
&= e(g_1, g_3^{id})^\gamma
\end{aligned}$$

Notice that, for any $i = 1, 2, \ldots, j$, we can obtain

$$e(g_1, g_2)^{\gamma\vec{t}_i} = e(g_1, g^{\vec{a}})^{\gamma\vec{t}_i} = e(g_1, g)^{\gamma<\vec{a}, \vec{t}_i>} = 1$$

where $<\vec{a}, \vec{t}_i> = 0$.

3. Security

1) Smoothness

Let id be an identity over the identity space, then, for any $sk_{id} \leftarrow \mathsf{KeyGen}(msk, id)$ and $C^* \leftarrow \mathsf{Encap}^*(id)$, we have

$$\begin{aligned}
k' &= \mathsf{Decap}'(sk_{id}, C^*)\\
&= e(c_1^*, d_1)e(c_2^*, d_2)(c_3^*)^{d_3}\\
&= e(g^\gamma, g_2^{-\alpha\vec{t}} g_3^{\alpha id}(u^{id}h)^{-r})e((u^{id}h)^\gamma, g^r)e(g_1, g_2)^{\gamma^*\vec{t}}\\
&= e(g^\gamma, g_2^{-\alpha\vec{t}})e(g^\gamma, g_3^{\alpha id})e(g^\gamma, (u^{id}h)^{-r})e((u^{id}h)^\gamma, g^r)\\
&\quad \times e(g_1, g_2)^{\gamma^*\vec{t}}\\
&= e(g_1, g_3^{id})^\gamma e(g_1, g_2)^{(\gamma^*-\gamma)\vec{t}}
\end{aligned}$$

where $\vec{t} \leftarrow_R (\mathbb{Z}_q)^n$ and $\gamma^*, \gamma \leftarrow_R \mathbb{Z}_q^*$ such that $\gamma^* \neq \gamma$.

Similarly, for any updated private key sk_{id}^j, we can obtain

$$\begin{aligned}
\hat{k}' &= \mathsf{Decap}(sk_{id}^j, C^*)\\
&= e(c_1^*, d_1^j)e(c_2^*, d_2^j)(c_3^*)^{d_3^j}\\
&= e(g^\gamma, g_2^{-\alpha\vec{t}} g_3^{\alpha id}(u^{id}h)^{-(r+\sum_{i=1}^j r_i)})\\
&\quad \times e((u^{id}h)^\gamma, g^{r+\sum_{i=1}^j r_i})e(g_1, g_2)^{\gamma^*(\vec{t}+\sum_{i=1}^j \vec{t}_i)}\\
&= e(g_1, g_3^{id})^\gamma e(g_1, g_2)^{(\gamma^*-\gamma)\vec{t}}
\end{aligned}$$

where, for any $i = 1, 2, \ldots, j$, we have

$$e(g_1, g_2)^{\gamma\vec{t}_i} = e(g_1, g^{\vec{a}})^{\gamma\vec{t}_i} = e(g_1, g)^{\gamma<\vec{a}, \vec{t}_i>} = 1$$

Therefore, the decapsulation results k' and $\hat{k}'$ are two random values over the encapsulated-key space $\mathcal{K} = G_T$.

2) Universality

According to the smoothness, for any identity id and the corresponding invalid encapsulation ciphertext

C^* (where $C^* = \mathsf{Encap}(id)$), we have

$$k = e(g_1, g_3^{id})^{\gamma} e(g_1, g_2)^{(\gamma^* - \gamma)\vec{t}}$$

where $\vec{t}$ is inherited from the user's private key sk_{id}, and each private key has the different vector $\vec{t}$. Thus, for any $sk_{id} \neq sk'_{id}$, we have

$$\mathsf{Decap}(C^*, sk_{id}) \neq \mathsf{Decap}(C^*, sk'_{id})$$

where $sk_{id} = \mathsf{KeyGen}(msk, id)$, $sk'_{id} = \mathsf{KeyGen}(msk, id)$.

For an U-IB-HPS, Zhou *et al.* [11] showed that, the different keys for the same identity must be generated by employing KeyGen, because the updated private key has the same underlying randomness as the original key. Hence, for an identity, the different keys must be generated from different underlying random values with the key generation algorithm.

3) Anonymity

From the description of anonymity of our basic IB-HPS Π, we can obtain our new system Π' is also anonymous.

4) Valid/invalid ciphertext indistinguishability

Similarly, we will prove the indistinguishability of valid/invalid ciphertext of our proposal Π' through the following theorem.

Theorem 2 If there exists a probabilistic polynomial time adversary $\mathcal{A}$ who can break the indistinguishability of valid/invalid ciphertext of our basic U-IB-HPS Π', then there exist a simulator $\mathcal{S}$ who can solve DBDH assumption.

Proof Similarly, a challenge tuple (g, g^a, g^b, g^c, T_v) and the corresponding public tuple $\mathbb{G} = (q, G, G_T, e(\cdot,\cdot), g)$ will be received by $\mathcal{S}$, where $T_v = e(g,g)^{abc}$ or $T_v = e(g,g)^{abc^*}$ $(c^* \neq c)$. Before the system is built, an identity id^* will be obtained by $\mathcal{S}$ from $\mathcal{A}$, which is employed as challenge identity. Hence, the distinguishability game is shown as follows.

– **Setup** In this stage, $\mathcal{S}$ does the following operations.

Chooses $\vec{a} \leftarrow_R (\mathbb{Z}_q)^n$, sets $g_1 = g^a$ (implicitly set $msk = a$) and $g_2 = (g^b)^{\vec{a}}$.

Chooses $x, y, z, \eta \leftarrow_R \mathbb{Z}_q^*$, after that, computes $g_3 = g^x$, $u = g_1^z$, $h = u^{-id^*} g^y$ and $\vec{\beta} = \eta \vec{a}$.

Finally, sends $mpk = (q, G, G_T, e(\cdot), g, g_1, g_2, g_3, h, \vec{\beta}, n)$ as the public parameter to $\mathcal{A}$.

– **Test stage 1** The complete private keys for any identity id (including id^*) will be generated by $\mathcal{S}$.

i) For any $id \neq id^*$, $\mathcal{S}$ chooses $r \leftarrow_R \mathbb{Z}_q^*$, $\vec{t} \leftarrow_R (\mathbb{Z}_q)^n$, and outputs $sk_{id} = \left(g_1^{xid} g_2^{\frac{-y<\vec{a},\vec{t}>}{z(id-id^*)}} (u^{id}h)^{-r}, g^r g_2^{\frac{<\vec{a},\vec{t}>}{z(id-id^*)}}, \vec{t}\right)$.

Let $r' = r + \frac{b<\vec{a},\vec{t}>}{z(id-id^*)}$, and r' is a random value over $\mathbb{Z}_q^*$ because $r \leftarrow_R \mathbb{Z}_q^*$ and $\vec{t} \leftarrow_R (\mathbb{Z}_q)^n$. Then, we have

$$\begin{aligned}
&g_1^{xid} g_2^{\frac{-y<\vec{a},\vec{t}>}{z(id-id^*)}} (u^{id}h)^{-r} \\
&= g_3^{a \cdot id} g^{\frac{-by<\vec{a},\vec{t}>}{z(id-id^*)}} (u^{id}h)^{-r' + \frac{b<\vec{a},\vec{t}>}{z(id-id^*)}} \\
&= g_3^{a \cdot id} g^{\frac{-by<\vec{a},\vec{t}>}{z(id-id^*)}} (g^{az(id-id^*)} g^y)^{\frac{b<\vec{a},\vec{t}>}{z(id-id^*)}} (u^{id}h)^{-r'} \\
&= g^{ab<\vec{a},\vec{t}>} g_3^{aid} (u^{id}h)^{-r'} \\
&= g_2^{a\vec{t}} g_3^{aid} (u^{id}h)^{-r'}
\end{aligned}$$

$$g^r g_2^{\frac{<\vec{a},\vec{t}>}{z(id-id^*)}} = g^r g^{\frac{b<\vec{a},\vec{t}>}{z(id-id^*)}} = g^{r + \frac{b<\vec{a},\vec{t}>}{z(id-id^*)}} = g^{r'}$$

Based on the above argument, we have that sk_{id_i} is a correct private key.

ii) For id^*, $\mathcal{S}$ chooses $r^* \leftarrow_R \mathbb{Z}_q^*$ and $\vec{t}^* \leftarrow_R (\mathbb{Z}_q)^n$, after that, outputs $sk_{id^*} = \left(g_1^{x \cdot id^*} g^{\frac{-y<\vec{a},\vec{t}^*>}{q}} (g_1^{id^*}h)^{-r^*}, g^{r^*} g_2^{\frac{<\vec{a},\vec{t}^*>}{q}}, \vec{t}^*\right)$.

Let $\hat{r}^* = r^* + \frac{b<\vec{a},\vec{t}^*>}{q}$, where $\hat{r}^*$ is a random value over $\mathbb{Z}_q^*$ because $r^* \leftarrow_R \mathbb{Z}_q^*$ and $\vec{t}^* \leftarrow_R (\mathbb{Z}_q)^n$. Then, we have

$$\begin{aligned}
&g_1^{xid^*} g^{\frac{-y<\vec{a},\vec{t}^*>}{q}} (u^{id^*}h)^{-r^*} \\
&= g_3^{aid^*} g^{\frac{-yb<\vec{a},\vec{t}^*>}{q}} (u^{id^*}h)^{-\hat{r}^* + \frac{b<\vec{a},\vec{t}^*>}{q}} \\
&= g_3^{aid^*} g^{\frac{-yb<\vec{a},\vec{t}^*>}{q}} (g_1^q g^y)^{\frac{b<\vec{a},\vec{t}^*>}{q}} (u^{id^*}h)^{-\hat{r}^*} \\
&= g^{ab<\vec{a},\vec{t}^*>} g_3^{aid^*} (u^{id^*}h)^{-\hat{r}^*} \\
&= g_2^{a\vec{t}^*} g_3^{aid^*} (u^{id^*}h)^{-\hat{r}^*}
\end{aligned}$$

$$g^{r^*} g_2^{\frac{<\vec{a},\vec{t}^*>}{q}} = g^{r^*} g^{\frac{b<\vec{a},\vec{t}^*>}{q}} = g^{r^* + \frac{b<\vec{a},\vec{t}^*>}{q}} = g^{\hat{r}^*}$$

Based on the above argument, we have that sk_{id^*} is also a correct private key.

– **Challenge stage** $\mathcal{S}$ returns the challenge ciphertext $C' = (c'_1, c'_2, c'_3) = (g^c, g^{cn}, T_v^{\vec{a}})$ to $\mathcal{A}$. Notice that, if $T_v = e(g,g)^{abc}$, then C' can be written as $C' = (c'_1, c'_2, c'_3) = (g^c, (u^{id^*}h)^c, (g_1, g_2)^c)$, and C' is a valid ciphertext. Otherwise, $T_v = e(g,g)^{abc^*}$, and C' can be written as $C' = (c'_1, c'_2, c'_3) = (g^c, (u^{id^*}h)^c, (g_1, g_2)^{c^*}$. Hence, C' is an invalid ciphertext.

– **Test stage 2** At this stage, $\mathcal{A}$ can also obtain the corresponding private key of any identity from $\mathcal{S}$.

– **Output** Finally, $\mathcal{A}$ outputs the guess v', and $\mathcal{S}$ returns v' received from $\mathcal{A}$.

From Theorem 1, we can obtain that if the security of our proposal Π' can be broken by the adversary $\mathcal{A}$ with an obvious advantage $\mathsf{Adv}^{\text{VI-IND}}_{\mathcal{A},\Pi'}(\kappa)$, then the corresponding simulator $\mathcal{S}$ can solve DBDH assumption with non-negligible advantage $\mathsf{Adv}^{DBDH}_{\mathcal{S}}(\kappa) \approx \mathsf{Adv}^{\text{VI-IND}}_{\mathcal{A},\Pi'}(\kappa)$.

5) Re-randomization property

For any $r_j, r_{j-1} \leftarrow_R \mathbb{Z}_q^*$ and $\vec{t}_j, \vec{t}_{j-1} \leftarrow_R (\mathbb{Z}_q)^n$, sk^j_{id}

and sk_{id}^{j-1} are two independent values over the private key space $\mathcal{SK}$. Also, the update key sk_{id}^{j} is a new private key from the view point of the adversary. Hence, the re-randomization property is achieved in the above our proposal Π'.

6) Invariance of update

For any identity id, we can obtain that $\mathsf{Decap}(C^*, sk_{id}) = \mathsf{Decap}(C^*, sk'_{id})$, where $C^* = \mathsf{Encap}^*(id)$, $sk_{id} = \mathsf{KeyGen}(id, msk)$ and $sk'_{id} = \mathsf{Update}(sk_{id})$. Hence, the invariance of update is also obtained in the above our proposal Π'.

4. Performance analysis

Now, we compare the performance of our constructs Π and Π' with that of the previous works [10], [11], [12], [14], [19], [20], and the comparison results are shown in Table 1 and Table 2, where Table 1 is the comparison of performance, and Table 2 is the comparison of computational efficiency. Specially, their instantiations were created in [10] and [19], and denoted by Scheme [10]-1, 2, and 3, and Scheme [19]-1, 2, and 3, respectively. In addition, two instantiations were proposed in [11] and denoted by Scheme [11]-1 and [11]-2.

Table 1. Comparison results of performance

Properties	Scheme [10]-1	Scheme [10]-2	Scheme [10]-3	Scheme [11]-1	Scheme [11]-2	Scheme [12]
$\mathcal{L}_{sk}$	$2\|G\| + 1\|q\|$	$2\|G\| + 1\|q\|$	$2\|G_N\| + 1\|N\|$	$2\|G\| + 3n\|q\|$	$2\|G\| + 3n\|q\|$	$1\|G\| + 1\|q\|$
$\mathcal{L}_C$	$2\|G\| + 1\|G_T\|$	$2\|G\| + 1\|G_T\|$	$3\|G_N\|$	$2\|G\| + 1\|G_T\|$	$2\|G\| + 1\|G_T\|$	$1\|G\| + 1\|G_T\|$
$\mathcal{K}$	G_T	G_T	G_N	G_T	G_T	G_T
Tight reduction	×	○	×	×	×	✓
Anonymity	×	×	×	×	×	×
Perfect update	×	×	×	✓	✓	×
Properties	Scheme [14]	Scheme [19]-1	Scheme [19]-2	Scheme [19]-3	Scheme [20]	Our system Π'
$\mathcal{L}_{sk}$	$2\|G\| + 1\|q\|$	$2\|G\| + 1\|q\|$	$1\|G\| + 1\|q\|$	$1\|G\| + 1\|q\|$	$1\|G\| + 1\|q\|$	$2\|G\| + n\|q\|$
$\mathcal{L}_C$	$2\|G\| + 1\|G_T\|$	$2\|G\| + 1\|G_T\|$	$1\|G\| + 1\|G_T\|$	$1\|G\| + 1\|G_T\|$	$1\|G\| + 1\|G_T\|$	$2\|G\| + 1\|G_T\|$
$\mathcal{K}$	G_T	G_T	G_T	G_T	G_T	G_T
Tight reduction	×	○	×	×	✓	✓
Anonymity	✓	✓	✓	✓	×	✓
Perfect update	×	×	×	×	×	✓

Table 2. Comparison of computational efficiency

Scheme	KeyGen	Encap	Decap
System [10]-1	$4T_e$	$4T_e$	$2T_b + 1T_e$
System [10]-2	$(4 + l_{id})T_e$	$(4 + l_{id})T_e$	$2T_b + 1T_e$
System [10]-3	$4T_e$	$4T_e$	$2T_b + 1T_e$
System [11]-1	$4T_e$	$4T_e$	$2T_b + 1T_e$
System [11]-2	$6T_e$	$4T_e$	$2T_b + 1T_e$
System [14]	$4T_e$	$4T_e$	$2T_b + 1T_e$
Our system Π'	$4T_e$	$4T_e$	$2T_b + 1T_e$

In Table 1, "○" denotes the corresponding security properties are not discussed. "✓" denotes the corresponding security properties are satisfied, "×" denotes that the security property is not satisfied. Furthermore, we use $\mathcal{L}_C$ and $\mathcal{L}_{sk}$ to denote the corresponding length of ciphertext and private key, respectively. For the size, we use $|G|$, $|G_T|$, $|G_N|$ to denote the number of bits for the representation of elements in G, G_T and G_N, where N is a composite number. In Table 2, we use T_b to denote the bilinear pairing operation, and use T_e to denote the exponent operation over group. Furthermore, l_{id} denotes the length of identity and n denotes the length of vector.

From Table 1, we find that, the tight security proof cannot be achieved in Scheme [10]-3, because the challenger needs to guess a random identity as the challenge identity in the security proof. Similarly, tight security reduction cannot be obtained in Scheme [10]-1 and Scheme [19]-2. However, Scheme [12], Scheme [19]-3 and Scheme [20] achieved the tight security reduction from the non-static q-ABDHE assumption. In addition, Scheme [10]-2 and Scheme [19]-1 do not discuss the corresponding security, and the formal proofs for proposed scheme were omited. Also, the anonymity is not discussed in [10], [11], [12]. Our proposals Π and Π' are anonymous and updatable, and can achieve tight reduction from the static security assumption. Although, Scheme [14] is anonymous, the previous analysis showed that this system does not have the perfect key update function.

To further compare the computational efficiency, the execution times of related cryptographic operation are collected from a personal computer by taking the mean of 30 consecutive executions with various inputs, which is collocated with Intel(R) Core i5-4200H CPU@ 2.8 GHz, 2 GB RAM, PBC library (PBC-0.5.14) and 64-bit Ubuntu 18.04 operating system, and the results of the execution times are $T_b = 1.365$ (ms) and $T_e = 1.112$ (ms).

For the key generation, the encapsulation and the decasulation algorithms of U-IB-HPS, the comparison of

computational costs between our system and the previous constructions [10], [11], [14] is shown in Fig.3. Furthermore, some previous systems [12], [19], [20] have high computational efficiency, however, the corresponding schemes created in [12], [20] do not obtain the anonymity and cannot achieve the continuous leakage resilience. Also, the scheme designed in [12] cannot obtain a tight security protocol without perfect key update function.

From Table 1, Table 2 and Fig.3, we can have that, our proposal has better performance while maintaining high computational efficiency. More specifically, for the design of cryptographic primitives, the performance is achieved through corresponding calculation. Our proposal maintains the same computational efficiency as the existing constructions of U-IB-HPS while having more security properties, which means that our proposal has higher computational efficiency, because our construction achieves more security performance with the same computation cost.

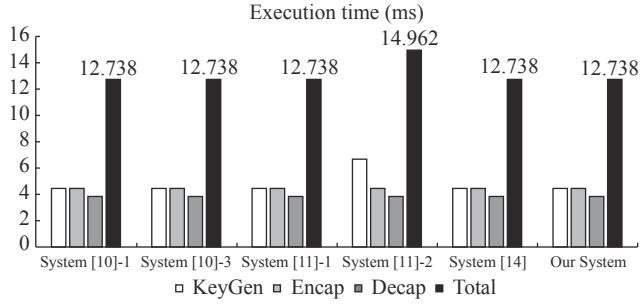

Fig. 3. The comparison of computational efficiency between our construction and the previous systems.

V. Non-interactive Leakage-Resilient Data Authorization

In [14], the generic construction of multiple cryptographic primitives with leakage resilience is proposed from U-IB-HPS. In this section, we will explore other applications of U-IB-HPS.

With the development of cloud computing, the people always share data through the Internet and store a large amount of shared data by employing cloud servers. However, how to achieve efficient data access authorization will be a challenge problem. Although, the traditional key exchange technology can generate a shared session key between data owner and data user, which increases the computation costs of parties and makes the computation efficiency of the corresponding schemes is low. Also, the above scheme cannot achieve the leakage resilience. That is, if an adversary can obtain a certain amount of leakage on the private key, then the above traditional protocol cannot keep their claimed security. Hence, the above method is not the best solution for data sharing authorization in the cloud computing, and it will also be faced to leakage attacks.

To achieve leakage-resilient cloud data security sharing, in this section, a non-interactive (one-round) leakage-resilient data authorization protocol is created by employing U-IB-HPS and average-case strong randomness extractor (we refer the reader to [3], [4] for details), which is described in Fig.4. In our protocol, the cloud server consists of two parts, one is key generation center (KGC), which is used to generate the system environment, and the other is database, which is used to store the ciphertext data. Furthermore, KGC is used to perform the registration of user, and is also employed to create the corresponding private keys for users.

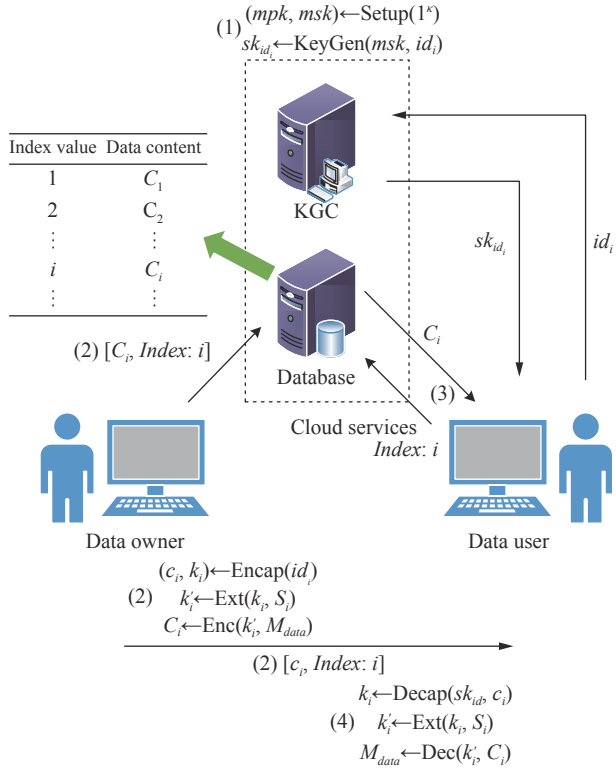

Fig. 4. Leakage-resilient non-interactive data authorization protocol.

Let $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Update}, \mathsf{Encap}, \mathsf{Encap}^*, \mathsf{Decap})$ be a U-IB-HPS with the encapsulated-key space $\{0,1\}^{l_1}$, let $\mathsf{Ext} : \{0,1\}^{l_1} \times \{0,1\}^{l_s} \to \{0,1\}^{l_2}$ be the average-case strong randomness extractor, and $(\mathsf{Enc}, \mathsf{Dec})$ be the symmetric cryptographic scheme with the key space $\{0,1\}^{l_2}$. The detailed information of our protocol is shown as follows.

1) KGC creates the system environment by running $(mpk, msk) \leftarrow \mathsf{Setup}(1^\kappa)$, and mpk will be published. Furthermore, the data user with identity id applies for registration with KGC and obtains the corresponding private key sk_{id} generated by KGC, where

$sk_{id} \leftarrow \mathsf{KeyGen}(msk, id)$.

2) Through the following computations, the data owner generates authorization messages $(c_i, S_i, Index\colon i)$ for the data user with the identity id_i, and stores the ciphertext information $(C_i, Index\colon i)$ of the shared data M_{data} in the cloud database.

- Choose $S_i \leftarrow_R \{0,1\}^{l_s}$, and compute

$$(c_i, k_i) \leftarrow \mathsf{Encap}(id_i) \text{ and } k_i' = \mathsf{Ext}(k_i, S_i)$$

where S_i is a random seed.

- Compute $C_i \leftarrow \mathsf{Enc}(k_i', M_{data})$.
- The ciphertext $(C_i, Index\colon i)$ of M_{data} is stored by the data owner in the cloud server, where i is an index, which is convenient for users to retrieve ciphertext.
- Finally, the data owner sends $(c_i, S_i, Index\colon i)$ to data user that expects authorization.

3) When receiving the messages $(c_i, S_i, Index\colon i)$ of the data owner, the data user downloads the corresponding ciphertext data C_i from the cloud database based on the index label $Index : i$, after that, the data user can obtain the shared data M_{data} with his/her own private key through the following computations.

- Compute $k_i \leftarrow \mathsf{Decap}(sk_{id_i}, c_i)$ and$k_i' = \mathsf{Ext}(k_i, S_i)$.
- Compute $M_{data} \leftarrow \mathsf{Dec}(k_i', C_i)$.
- Finally, the data user runs $sk'_{id_i} \leftarrow \mathsf{Update}(sk_{id_i})$, and sets sk'_{id_i} as the new private key. In this case, the additional leakage on the sk_{id_i} is invalid for sk'_{id_i}. Therefore, the data user can resist the new leakage attacks, so the above protocol provides continuous leakage resilience.

We have to stress that, in our protocol, the leakage resilience can be obtained from the underlying tools U-IB-HPS and randomness extractor, in which, U-IB-HPS generates an input that meets the requirements for the randomness extractor, and an encapsulated key is generated by extractor that remains random in the leakage setting. That is, as long as the length of the leakage information obtained by the adversary is less than the leakage parameter, the key used to encrypt data is still uniformly random.

Therefore, our protocol can achieve the corresponding security properties based on the correctness and security of the underlying U-IB-HPS. Furthermore, the above non-interactive data authorization protocol also has continuous leakage resilience, and for any round leakage parameter $\lambda \leq l_1 - l_2 - \omega(\kappa)$, our protocol can keep their claimed security.

As discussed above, U-IB-HPS has a wide range of applications, it generates an encapsulation key, which can be used as the symmetric key of symmetric cryptographic primitives. Therefore, U-IB-HPS is an important basic tool to construct identity-based hybrid encryption scheme, which can inherit the efficient management mode of public key in public key cryptography. At the same time, it can take into account the advantages of fast encryption and decryption of symmetric cryptography. Hence, U-IB-HPS can be employed as a key encapsulation mechanism in actual application to achieve secure data transmission, for example, Internet of things [21], [22], the public cloud [23], [24], edge computing [25], industrial Internet of things [26]–[28], etc.

VI. Conclusions

The previous constructions of U-IB-HPS either obtain tight security reduction based on the non-static complexity assumptions, or do not have perfect key update. To further adress the above problems, a new concrete construction of U-IB-HPS is generated in this paper, which has provable security in the selective identity security model based on the hardness of DBDH assumption. Compared with the previous works [10]–[12], [14], [19], [20], our proposal achieves tight security reduction from the static security assumption, and our system is anonymous. Furthermore, based on the U-IB-HPS and average case strong randomness extractor, a non-interactive (one round) leakage-resilient data authorization protocol is proposed. Our work has enriched the application of U-IB-HPS.

According to our discussion, we have that it is a challenging problem to construct an IB-HPS or U-IB-HPS with adaptive security under a static assumption. In order to solve the above problems, we will try to construct a generic construction of IB-HPS based on known cryptographic primitives such as IBE.

References

[1] M. Naor and G. Segev, "Public-key cryptosystems resilient to key leakage," *SIAM Journal on Computing*, vol.41, no.4, pp.772–814, 2012.

[2] S. L. Liu, J. Weng, and Y. L. Zhao, "Efficient public key cryptosystem resilient to key leakage chosen ciphertext attacks," in *Proceedings of the Cryptographers' Track at the RSA Conference*, San Francisco, CA, USA, pp.84–100, 2013.

[3] A. Labao and H. Adorna, "A CCA-PKE secure-cryptosystem resilient to randomness reset and secret-key leakage," *Cryptography*, vol.6, no.1, article no.2, 2022.

[4] R. Nishimaki and T. Yamakawa, "Leakage-resilient identity-based encryption in bounded retrieval model with nearly optimal leakage-ratio," in *Proceedings of the 22nd IACR International Workshop on Public Key Cryptography*, Beijing, China, pp.466–495, 2019.

[5] C. L. Cai, X. R. Qin, T. H. Yuen, *et al.*, "Tight leakage-resilient identity-based encryption under multi-challenge setting," in *Proceedings of 2022 ACM on Asia Conference on Computer and Communications Security*, Nagasaki, Japan, pp.42–53, 2022.

[6] J. G. Li, M. L. Teng, Y. C. Zhang, *et al.*, "A leakage-resilient CCA-secure identity-based encryption scheme," *The Computer Journal*, vol.59, no.7, pp.1066–1075, 2016.

[7] F. Benhamouda, A. Degwekar, Y. Ishai, *et al.*, "On the loc-

al leakage resilience of linear secret sharing schemes," *Journal of Cryptology*, vol.34, no.2, article no.10, 2021.
[8] H. K. Maji, H. H. Nguyen, A. Paskin-Cherniavsky, *et al.*, "Leakage-resilience of the Shamir secret-sharing scheme against physical-bit leakages," in *Proceedings of the 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Zagreb, Croatia, pp.344–374, 2021.
[9] J. Alawatugoda and T. Okamoto, "Standard model leakage-resilient authenticated key exchange using inner-product extractors," *Designs, Codes and Cryptography*, vol.90, no.4, pp.1059–1079, 2022.
[10] S. S. M. Chow, Y. Dodis, Y. Rouselakis, *et al.*, "Practical leakage-resilient identity-based encryption from simple assumptions," in *Proceedings of the 17th ACM Conference on Computer and Communications Security*, Chicago, IL, USA, pp.152–161, 2010.
[11] Y. W. Zhou, B. Yang, and Y. Mu, "The generic construction of continuous leakage-resilient identity-based cryptosystems," *Theoretical Computer Science*, vol.772, pp.1–45, 2019.
[12] J. Alwen, Y. Dodis, M. Naor, *et al.*, "Public-key encryption in the bounded-retrieval model," in *Proceedings of the 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, French Riviera, pp.113–134, 2010.
[13] Y. Dodis, K. Haralambiev, A. López-Alt, *et al.*, "Cryptography against continuous memory attacks," in *Proceedings of the 51th Annual Symposium on Foundations of Computer Science*, Las Vegas, NV, USA, pp.511–520, 2010.
[14] Y. W. Zhou, B. Yang, Z. Xia, *et al.*, "Anonymous and updatable identity-based hash proof system," *IEEE Systems Journal*, vol.13, no.3, pp.2818–2829, 2019.
[15] Y. W. Zhou, B. Yang, T. Wang, *et al.*, "Novel updatable identity-based hash proof system and its applications," *Theoretical Computer Science*, vol.804, pp.1–28, 2020.
[16] Q. Q. Lai, B. Yang, Y. Yu, *et al.*, "Updatable identity-based hash proof system based on lattices and its application to leakage-resilient public-key encryption schemes," *Journal of Computer Science and Technology*, vol.33, no.6, pp.1243–1260, 2018.
[17] Q. Q. Lai, B. Yang, Z. Xia, *et al.*, "Novel identity-based hash proof system with compact master public key from lattices in the standard model," *International Journal of Foundations of Computer Science*, vol.30, no.4, pp.589–606, 2019.
[18] B. Waters, "Efficient identity-based encryption without random oracles," in *Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Aarhus, Denmark, pp.114–127, 2005.
[19] Y. Chen, Z. Y. Zhang, D. D. Lin, *et al.*, "Anonymous identity-based hash proof system and its applications," in *Proceedings of the 6th International Conference on Provable Security*, Chengdu, China, pp.26–28, 2012.
[20] B. Wang, "Leakage-resilient message authentication code scheme based on hidden identity weak hash proof system," *IET Information Security*, vol.10, no.4, pp.173–179, 2016.
[21] Y. R. Liu, J. Yu, J. X. Fan, *et al.*, "Achieving privacy-preserving DSSE for intelligent IoT healthcare system," *IEEE Transactions on Industrial Informatics*, vol.18, no.3, pp.2010–2020, 2022.
[22] J. G. Li, Y. Chen, J. G. Han, *et al.*, "Decentralized attribute-based server-aid signature in the internet of things," *IEEE Internet of Things Journal*, vol.9, no.6, pp.4573–4583, 2022.
[23] J. H. Wei, X. F. Chen, X. Y. Huang, *et al.*, "RS-HABE: revocable-storage and hierarchical attribute-based access scheme for secure sharing of e-health records in public cloud," *IEEE Transactions on Dependable and Secure Computing*, vol.18, no.5, pp.2301–2315, 2021.
[24] L. Zhou, A. M. Fu, G. M. Yang, *et al.*, "Efficient certificateless multi-copy integrity auditing scheme supporting data dynamics," *IEEE Transactions on Dependable and Secure Computing*, vol.19, no.2, pp.1118–1132, 2022.
[25] W. Y. Zheng, B. Chen, and D. B. He, "An adaptive access control scheme based on trust degrees for edge computing," *Computer Standards & Interfaces*, vol.82, article no.103640, 2022.
[26] Z. R. Qiao, Q. L. Yang, Y. W. Zhou, *et al.*, "Improved secure transaction scheme with certificateless cryptographic primitives for IoT-based mobile payments," *IEEE Systems Journal*, vol.16, no.2, pp.1842–1850, 2022.
[27] Z. R. Qiao, Y. W. Zhou, B. Yang, *et al.*, "Secure and efficient certificate-based proxy signature schemes for industrial internet of things," *IEEE Systems Journal*, vol.16, no.3, pp.4719–4730, 2022.
[28] M. W. Zhang, W. X. Song, and J. X. Zhang, "A secure clinical diagnosis with privacy-preserving multiclass support vector machine in clouds," *IEEE Systems Journal*, vol.16, no.1, pp.67–78, 2022.

QIAO Zirui was born in 1985. She received the B.E. degree in computer software and theory from Shaanxi Normal University. She is currently working toward the Ph.D. degree with the School of Computer Science of Shaanxi Normal University. Her research interests include information security and cryptography.
(Email: qzr_snnu@163.com)

ZHOU Yanwei (corresponding author) was born in 1986. He received the Ph.D. degree in computer software and theory from the Shaanxi Normal University, Xi'an, China, in 2018. He is currently an Associate Professor with the Shaanxi Normal University, China. His research interests include cryptography.
(Email: zyw@snnu.edu.cn)

YANG Bo was born in 1963. He received the Ph.D. degree in cryptography from Xidian University. He is a Professor of Shaanxi Normal University. His research interests include information security and cryptography.
(Email: byang@snnu.edu.cn)

ZHANG Wenzheng was born in 1965. He received the B.E. degree in computer systems organization from Southwest Communication Research Institute. He is a Professor of China Electronics Technology Group Corporation. His research interests include cryptography.

ZHANG Mingwu was born in 1970. He received the Ph.D. degree from the South China Agricultural University, Guangzhou, China, in 2009. He is currently a Professor with the Hubei University of Technology. His research interests include cryptography.