4-2020

Identity-based encryption transformation for flexible sharing of encrypted data in public cloud

Robert H. DENG
Singapore Management University, robertdeng@smu.edu.sg

Zheng QIN

Qianhong WU

Zhenyu GUAN

Robert H. DENG

See next page for additional authors

Citation

DENG, Robert H.; QIN, Zheng; WU, Qianhong; GUAN, Zhenyu; DENG, Robert H.; WANG, Yujue; and ZHOU, Yunya. Identity-based encryption transformation for flexible sharing of encrypted data in public cloud. (2020). *IEEE Transactions on Information Forensics and Security*. 15, 3168-3180.
Available at: https://ink.library.smu.edu.sg/sis_research/7129

Identity-Based Encryption Transformation for Flexible Sharing of Encrypted Data in Public Cloud

Hua Deng, Zheng Qin*, *Member, IEEE,* Qianhong Wu*, *Member, IEEE,* Zhenyu Guan, *Member, IEEE,* Robert H. Deng, *Fellow, IEEE,* Yujue Wang, and Yunya Zhou

Abstract—With the rapid development of cloud computing, an increasing number of individuals and organizations are sharing data in the public cloud. To protect the privacy of data stored in the cloud, a data owner usually encrypts his data in such a way that certain designated data users can decrypt the data. This raises a serious problem when the encrypted data needs to be shared to more people beyond those initially designated by the data owner. To address this problem, we introduce and formalize an identity-based encryption transformation (IBET) model by seamlessly integrating two well-established encryption mechanisms, namely identity-based encryption (IBE) and identity-based broadcast encryption (IBBE). In IBET, data users are identified and authorized for data access based on their recognizable identities, which avoids complicated certificate management in usual secure distributed systems. More importantly, IBET provides a transformation mechanism that converts an IBE ciphertext into an IBBE ciphertext so that a new group of users not specified during the IBE encryption can access the underlying data. We design a concrete IBET scheme based on bilinear groups and prove its security against powerful attacks. Thorough theoretical and experimental analyses demonstrate the high efficiency and practicability of the proposed scheme.

Index Terms—Cloud computing; Data sharing; Data privacy; Access control; Cryptographic encryption.

1 Introduction

Cloud computing provides powerful and flexible storage services for individuals and organizations [1]. It brings about lots of benefits of sharing data with geographically dispersed data users, and significantly reduces local burden of storage management and maintenance. However, the concerns on data security and privacy are becoming one of the major obstacles impeding more widespread usage of cloud storage [2], since data owners lose physical control on their data after data are outsourced to cloud servers maintained by a cloud services provider (CSP). Data owners may worry about whether their sensitive data have been accessed by unauthorized users or malicious CSP.

Cryptographic encryptions are widely suggested as standard approaches to protect the security and privacy of data outsourced to clouds [3]. With encryption mechanisms, data owners first encrypt their data and then outsource to cloud servers. Then the data in clouds are stored in ciphertext format and can only be accessed by the users having matching decryption keys. In a public cloud storage system, where different data owners may employ different encryption mechanisms according to their own data sharing requirements, it is often that a data owner wants to share his data with only *one* user and thus encrypts the data to generate a particular ciphertext that can only be decrypted by the specific user. However, as data sharing requirement changes, the same data owner would like to share his data with *more* users, which, therefore, requires to transform the ciphertext format so that multiple users can decrypt.

There are many scenarios in which the ciphertext transformation mentioned above is highly desirable. Consider a group of medical insurance agents draft a health insurance plan for a client. To do so, each agent needs to collect the client's personal information (e.g., electronic health records, occupations data, financial reports) from various data sources such as hospitals, employers, tax departments. The required data may be stored in remote cloud servers and especially, may be encrypted under different encryption mechanisms. To allow the agents to read and make use of the required data, a naive way is to let each agent acquire the corresponding decryption keys from the authorities who manage respective data. However, this would pose great concerns on data privacy. The authorities would ask a natural question: *"If I give my decryption key to the agents, how to assure that all the agents would not leak the decryption key or use the decryption key to access other clients' stored data?"*

This paper attempts to solve such problem technically so that the authorities can transform the ciphertexts from one

* *Z. Qin and Q. Wu are the corresponding authors.*

- *H. Deng and Z. Qin are with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. E-mail: {hdeng;zqin}@hnu.edu.cn*
- *Q. Wu and Z. Guan are with the School of Cyber Science and Technology, Beihang University, Beijing, China. E-mail: qhwu@xidian.edu.cn; guanzhenyu@buaa.edu.cn*
- *R. H. Deng is with the School of Information Systems, Singapore Management University, Singapore. E-mail: robertdeng@smu.edu.sg*
- *Y. Wang is with Guangxi Key Laboratory of Cryptography and Information Security, School of Computer Science and Information Security, Guilin University of Electronic Technology. E-mail: yjwang@guet.edu.cn*
- *Y. Zhou is with the State Grid Hunan Maintenance Company, Changsha, China. E-mail: zhouyy12@hn.sgcc.com.cn*

encryption system to another, without handing over their decryption keys. In particular, we consider an encryption transformation mechanism that connects two types of well-established encryption systems, i.e., identity-based encryption (IBE) and identity-based broadcast encryption (IBBE). We take electronic health records sharing as a motivation of our work.

Suppose a patient is equipped with implantable or wearable medical sensors to collect personal physiological records. These records are aggregated at a mobile device and then uploaded to a remote server. To protect personal privacy, the patient may encrypt his health records by some encryption mechanism, e.g., IBE, so that only his doctor can read the health records and then make proper diagnosis. At some point, the doctor finds a complicated situation about the patient's health and consequently, decides to consult a group of experts from different hospitals. For full understanding of the patient's health condition, the experts first need to read the health records (see Fig. 1). Since the records are encrypted previously, the experts are impossible to directly read the data. Meanwhile, the encryption method taken by the patient and the corresponding decryption key are unknown to the experts. This results in a dilemma for the experts: *"How could we read the patient's health records in order to provide our treatment advices?"*

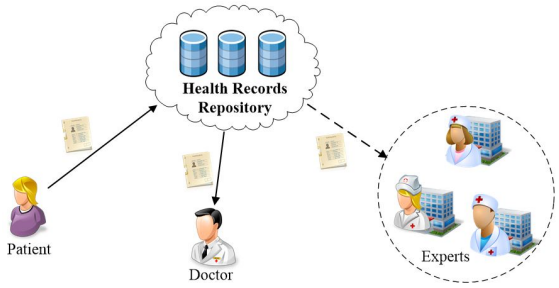

Fig. 1. Electronic Health Records Sharing with More Doctors

A trivial solution would be that the doctor first decrypts all the encrypted records and then sends out the data in plaintext (not encrypted) format to each expert. This, however, may be impractical for the doctor since a considerable computation and communication costs may be caused due to the massive health data uploaded everyday. More importantly, there is a risk of privacy disclosure by sending data in plaintext format.

There exists a cryptographic tool called proxy re-encryption (PRE) that would be of help here. PRE can transform the doctor's ciphertext into a ciphertext that can be decrypted by one expert. Then, for n experts, PRE needs to run n times repeatedly for transferring the patient's health data to all experts, which is inefficient. We observe that IBBE achieves a useful encryption mechanism that allows multiple users to simultaneously decrypt a ciphertext. Thus, we ask: *"Can we find an efficient way to transform the encrypted data in IBE ciphertext format into an IBBE ciphertext so that multiple users can decrypt at the same time?"*

1.1 Our contributions

In this paper, we try to answer the above question by studying encryption transformation between two different encryption systems. For the first time, we propose a novel notion called identity-based encryption transformation (IBET). We also define the notion (including algorithm definition and security model) of IBET. Then we design a concrete IBET scheme in bilinear groups, which provides the following attractive features.

- **Identity-based data storage**. Data owner can securely outsource their data to a remote cloud server which is not fully trusted. The data are encrypted and stored in the server in IBE/IBBE ciphertext format so that only the users authorized by the data owners can access them. All users, including data owners and data consumers, are recognized with their unique identities, which avoids the usage of complicated public-key certificates.
- **Cross-domain encryption transformation**. Our IBET scheme achieves a cross-domain encryption transformation which can be viewed as a bridge connecting IBE and IBBE. In particular, a data owner (or an authorized data consumer) can transform the data stored in IBE ciphertext format into the data in IBBE ciphertext format, so that a set of users specified by the data owner (or the authorized data consumer) can simultaneously access the data.
- **Strong security guarantee**. Our IBET scheme achieves a strong security in the sense that: 1) it can deter any unauthorized access to the data stored in the cloud server; 2) it can prevent leakage of some private information (e.g., private key) about the one who authorizes to transform encrypted data; 3) the transformation would not reveal any useful information about the sensitive data.

We also conduct a series of experiments on our IBET scheme and make comparisons with some related schemes. The results show that the IBET scheme achieves a high performance in transforming the encrypted data, without incurring any significant computation costs to cloud clients or cloud servers.

Applications. Our IBET scheme can be applied to many real-world data sharing applications. First of all, the example of health records sharing described previously is an appropriate area where our IBET can be applied. Cloud-based encrypted email forwarding is another possible application. Imagine that several companies deploy their email systems on cloud servers. IBET can be used as a gateway to transform an encrypted email destined to an employee in one company into an encrypted email what can be received and decrypted by multiple employees in different companies. Vehicular ad-hoc network is also a potential application for IBET. When a car receives an encrypted report about front car condition or accident ahead and would like further to broadcast the situation to rear vehicles, IBET can be used to directly transform the encrypted report into a broadcast ciphertext that allows multiple receivers to decrypt. Last but not least, in a mobile office environment, IBET may be utilized as a mobile application to securely share business data with a company director via a public cloud, and then transform the encrypted business data (if requested) so that the whole management team can access.

1.2 Related Work

Outsourced data protection. Cryptographic encryption methods have been extensively used to secure data outsourced to clouds. Traditional public-key encryption methods are applied to achieve user-centric access control on outsourced data [4], [5]. Identity-based encryption (IBE) [6] is a promising cryptographic tool which eliminates trusted certificates for all users. Wei *et al.* [7] exploited IBE to secure data sharing in mobile computing environments. He *et al.* [8] employed IBE to construct a handshake scheme in healthcare social network to secure data exchanged in patients. Identity-based broadcast encryption (IBBE) [9] extends IBE to support multi-receiver encryption in the sense that a user encrypts a message once for multiple intended receivers. In light of such useful feature, Deng *et al.* [10] utilized IBBE in cloud storage systems to allow multiple authorized visitors to access the same outsourced file. To revoke some recipients from the initial receiver set of the IBBE ciphertext, a number of revocable IBBE schemes are proposed [11], [12], [13], [14].

Inter-domain Transformation. Blaze *et al.* [15] first introduced the concept of proxy re-encryption to handle ciphertext transformation within an encryption system. With this PRE, a user can transform a ciphertext generated under Alice's public key into a ciphertext under Bob's public key. Ateniese *et al.* [16] classified PRE into different categories: bidirectional and unidirectional PRE, single-hop and multi-hop PRE, interactive and non-interactive PRE. Many efforts have been made to improve efficiency and security of PRE and most of them focus on unidirectional PRE. Libert and Vergnaud [17] presented the first unidirectional PRE scheme. Cao *et al.* [18] proposed the autonomous path PRE scheme to enable a user to designate a path of preferred authorized visitors to his outsourced data. Guo *et al.* [19] introduced accountability into unidirectional PRE to identify the proxy which abuses its re-encryption keys.

By combining PRE and IBE, Green and Ateniese [20] proposed the first identity-based PRE (IBPRE), which is an extension of PRE in identity-based settings. Chu and Tzeng [21] presented an IBPRE scheme with short ciphertexts and decryption keys, while it is vulnerable to collusion attack, i.e., the coalition of the proxy server and the authorized users could compromise the secret information about data owners. Liang *et al.* [22] overcome this security issue by proposing the cloud-based revocable IBPRE scheme. This scheme requires the interaction between data owners and a key generator authority for each transformation, which may result an efficiency problem. Xu *et al.* [23] proposed an IBBE-based PRE scheme by introducing IBBE into PRE. Apart from IBPRE, there are other extensions of PRE, such as attribute-based PRE [24], [25], time-based PRE [26], function-based PRE [27], etc. However, these PRE schemes mainly provides ciphertext transformation in the same encryption system, that is, ciphertexts cannot be converted into another format.

Cross-domain transformation. There are a few schemes achieving cross-domain encryption transformation. Matsuo [28] linked the traditional public-key encryption and identity-based encryption by allowing to transform a ciphertext of public key systems into a ciphertext of IBE systems. Mizuno and Doi [29] also proposed a unidirectional PRE scheme that transforms ciphertexts of an attribute-based encryption system into ciphertexts of an IBE system, while requiring users to interact with each other and store additional information for transformation. Recently, Jiang *et al.* [30] proposed a cross-domain encryption switching scheme that connects traditional public-key encryption and identity-based encryption, while it requires cryptographic certificates for all the users in the public-key encryption system. This paper aims at addressing cross-domain transformation in identity-based settings; thus saves the cost for certificate management. Moreover, this paper provides encryption transformation from (one-receiver) IBE system to (multi-receiver) IBBE system so that one's data can be shared with more users even though the data have already been encrypted.

1.3 Paper Organization

The rest of the paper is organized as follows. We describe the IBET system architecture, threat model and security goals in Section 2. The framework and security definition of the IBET system are formalized in Section 3. We present a concrete IBET scheme in Section 4. The security and performance analyses are given in Section 5 and Section 6, respectively. Finally, Section 7 concludes the paper.

2 System Model

2.1 System Architecture

The architecture of our IBET system is shown in Fig. 2. An IBET system consists of four types of entities, that is, data owners, data consumers, registry authority (RA) and cloud service provider (CSP). Generally, data owners and data consumers are both cloud clients. RA is a trusted party that is responsible for setting up system, responding to registration requests and issuing public parameters for file outsourcing. CSP has two major tasks: 1) providing storage services for clients to store outsourced files; 2) providing computation services for clients to transform stored files. In real world, an enterprise or an organization can buy the storage and computation services provided by CSP, and the IT center of the enterprise or the organization plays the role of RA. In this way, all the (registered) employees can make use of storage and computation services.

Data owners can outsource data to CSP. Specifically, to protect data privacy, data owners can employ IBE encryption mechanism to process data and then outsource the resulting files (data in ciphertext format) to CSP. Suppose that a file is the result of IBE encryption for some data (thus the data can be accessed by only one data consumer). If the corresponding data owner further wants to share the data with more data consumers, he generates an authorization token and sends it to CSP; then CSP can transform the file in IBE ciphertext format into a file in IBBE ciphertext format so that all designated data consumers can decrypt and then access the underlying data. In this way, for the data previously encrypted by IBE and originally accessible to only one data consumer, the data owner can authorize more data consumers to access it.

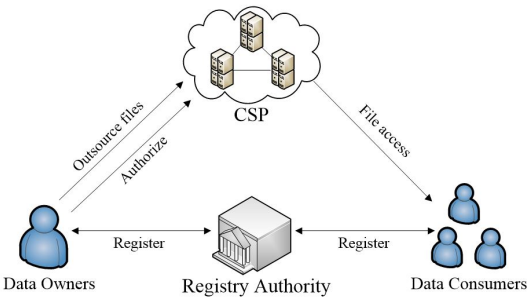

Fig. 2. System Architecture

2.2 Threat Model and Security Goals

An IBET system confronts three types of active attacks. First, cloud clients may impersonate data owners or authorized data consumers to try to access outsourced data, e.g., an employee pretends to be his colleague by using the colleague's device to access CSP. Second, malicious CSP or hackers intruding in cloud servers may search and steal owners' data. Third, CSP may abuse the authorization tokens of data owners to transform encrypted data that are out of the scope of authorization. Considering these realistic attacks, we require that a secure IBET system should at least satisfy the following security goals.

- Data security protection: If data have been encrypted before outsourced, then only the clients holding correct decryption keys can access (these client are also called authorized clients). The encrypted data are unreadable to CSP or unauthorized clients (those having no correct decryption keys).
- Controllable transformation: Only the files specified by the data owner in the authorization token can be transformed by CSP. CSP and other clients cannot cooperatively deduce a valid authorization token in order to transform unspecified files, nor detect sensitive information about the data encrypted in unspecified files.

3 Definitions

3.1 Framework of IBET System

Formally, an IBET system consist of six polynomial-time computable algorithms, that is, Setup, Register, Encrypt, Authorize, Transform, and Decrypt.

- Setup$(1^\lambda, m) \rightarrow (PP, MSK)$: The system setup algorithm, run by RA, takes as input a security parameter λ and the allowed maximal number m of data consumers authorized to access the same data. It outputs the public parameter PP for the system and the master secret key MSK for RA itself.
- Register$(PP, MSK, ID) \rightarrow SK_{ID}$: The registration algorithm, run by RA, takes as input the public parameter PP, the master secret key MSK and an identity $ID \in \{0,1\}^*$. It outputs a private key SK_{ID}.
- Encrypt$(PP, M, ID) \rightarrow CT_{ID}$: The encryption algorithm, run by a data owner, takes as input the public parameter PP, the message M to be encrypted and an identity ID. It outputs an IBE ciphertext CT_{ID}.
- Authorize$(PP, SK_{ID}, S) \rightarrow TK_{ID\rightarrow S}$: The authorization algorithm, run by a data owner with identity ID, takes as input the data owner's private key SK_{ID}, the public parameter PP and the set S of identities of data consumers. It outputs an authorization token $TK_{ID\rightarrow S}$.
- Transform$(PP, TK_{ID\rightarrow S}, CT_{ID}) \rightarrow CT_S$: The transformation algorithm, run by CSP, takes as input the authorization token $TK_{ID\rightarrow S}$, the public parameter PP and the IBE ciphertext CT_{ID}. It outputs a transformed (IBBE) ciphertext CT_S.
- Decrypt$(PP, CT_{ID}/CT_S, SK_{ID'}) \rightarrow M/\perp$: The decryption algorithm, run by a data consumer ID', takes as input the public parameter PP, a private key $SK_{ID'}$ and a ciphertext CT_{ID} or CT_S. For CT_{ID}, it outputs the message M if $ID = ID'$ and a false symbol $\perp$ otherwise; for CT_S, it outputs the message M if $ID' \in S$ and a false symbol $\perp$ otherwise.

A secure IBET scheme should be *sound*, that is, if each entity honestly follows the scheme, then any failure would not happen during the scheme running. Formally, for any $(PP, MSK) \leftarrow$ Setup$(1^\lambda, m)$, the following conditions must be satisfied:

- For any IBE ciphertext $CT_{ID} \leftarrow$ Encrypt(PP, M, ID) and any private key $SK_{ID'} \leftarrow$ Register(PP, ID', MSK), if $ID = ID'$, then the decryption algorithm Decrypt$(PP, CT_{ID}, SK_{ID'})$ always outputs the plaintext M.
- For any transformed ciphertext $CT_S \leftarrow$ Transform$(PP, TK_{ID\rightarrow S}, CT_{ID})$, where $TK_{ID\rightarrow S} \leftarrow$ Authorize(PP, SK_{ID}, S) and $CT_{ID} \leftarrow$ Encrypt(PP, M, ID), and any private key $SK_{ID'} \leftarrow$ Register(PP, MSK, ID'), if $ID' \in S$, the decryption algorithm Decrypt$(PP, CT_S, SK_{ID'})$ always outputs the plaintext M.

The first condition is straightforward. It means that any encrypted message in IBE ciphertext format can only be decrypted by the intended data consumer. The second one is somewhat sophisticated. Its main idea is to define that any properly transformed ciphertext (from an IBE ciphertext) can be correctly decrypted by all intended data consumers. Thus, we must define what is a properly transformed ciphertext and who are the intended data consumers able to decrypt the ciphertext.

For a transformed ciphertext, the second condition defines that this ciphertext is properly transformed from the original IBE ciphertext, if the authorization token used in the transformation was created by the user who is capable of decrypting the original ciphertext. Also, the second condition defines that a transformed ciphertexts can be decrypted by the data consumers whose identities are indicated in the authorization token.

3.2 Formal Security Definitions

We present formal security definitions to capture the *indistinguishability of ciphertexts against selective identity and chosen-plaintext attack* (IND-sID-CPA) launched by unauthorized clients and curious CSP, and the *leakage-resistance of private keys against collusion attack* (LR-CA) launched by

authorized clients and CSP. For the former, we prevent an adversary, which is not given a valid private key for decryption, from gaining access to the data encrypted in IBE or IBBE ciphertext. For the latter, we prevent an adversary, which could collude with authorized clients and CSP by having their private keys and authorization tokens, respectively, from recovering the private keys that were used to generate the authorization tokens. We note that if the private key of a data owner is compromised, then all the owner's data stored in CSP are revealed to the adversary.

We first consider the case where unauthorized clients or malicious CSP try to access the data encrypted in IBE ciphertext or transformed (IBBE) ciphertext. Let $\mathcal{A}$ be a probabilistic polynomial-time adversary, which plays the following game with the challenger $\mathcal{C}$ and tries to distinguish two encrypted messages.

Setup: The adversary $\mathcal{A}$ chooses a target identity ID^* and sends it to the challenger. With security parameter λ and the maximum number m of authorized data consumers, the challenger $\mathcal{C}$ runs the Setup algorithm to generate system public parameters PP and master secret keys MSK. It gives PP to $\mathcal{A}$ and keeps MSK secret.

Phase 1: The adversary $\mathcal{A}$ can adaptively issue the following queries to the challenger:

- *User registration*(ID_i). $\mathcal{A}$ can ask for a private key for any user with identity ID_i. In response, the challenger $\mathcal{C}$ runs the registration algorithm and returns the output private key SK_{ID_i} to $\mathcal{A}$.
- *Authorization*($ID_i \rightarrow S_i$). In each query, $\mathcal{A}$ can ask for an authorization token by submitting an identity ID_i and a set S_i of identities. In response, if ID_i has not been queried for a private key, the challenger $\mathcal{C}$ first generates SK_{ID_i}. Then, $\mathcal{C}$ runs the authorization algorithm $\mathsf{Authorize}(PP, SK_{ID_i}, S_i)$ to generate an authorization token $TK_{ID_i \rightarrow S_i}$ and returns it to $\mathcal{A}$.

Challenge: When deciding that Phase 1 is over, the adversary $\mathcal{A}$ submits two equal-length messages M_0 and M_1. The restrictions for $\mathcal{A}$ are that (1) it has never queried private key for ID^*; and (2) for any S_i and any $ID_i \in S_i$, it has queried at most one of two queries: a private key query on ID_i and an authorization token query on (ID^*, S_i). The challenger $\mathcal{C}$ then flips a coin $b \in \{0, 1\}$, encrypts M_b under ID^* and returns the ciphertext CT_{ID^*} to $\mathcal{A}$.

Phase 2: The same as Phase 1 with the restrictions described in challenge phase.

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

The advantage of $\mathcal{A}$ in this game is defined as

$$Adv_{\mathcal{A}}^{IND-sID-CPA} = |\Pr[b = b'] - 1/2|.$$

Definition 1. *An IBET system is IND-sID-CPA secure if all probabilistic polynomial-time adversaries $\mathcal{A}$ have at most a negligible advantage in the above game.*

In the challenge phase of the above game, the two restrictions are to prevent the adversary from winning the game in a trivial way. Specifically, if the adversary has a private key for ID^*, then it can correctly decrypt the challenge ciphertext CT_{ID^*} and always output $b' = b$. If the adversary has a private key for $ID_i \in S_i$ and an authorization token for ID^* and S_i, it can first convert CT_{ID^*} into CT_{S_i} and then use the private key of ID_i to decrypt CT_{S_i}. In this way, the adversary can also always output $b' = b$.

We note that Definition 1 covers the security against unauthorized access attack to the data stored in both IBE and IBBE ciphertext formats. The adversary, which is challenged with an IBE ciphertext CT_{ID^*}, can query authorization token $TK_{ID^* \rightarrow S}$ and then apply $TK_{ID^* \rightarrow S}$ to transform CT_{ID^*} into a transformed IBBE ciphertext CT_S. This means that in the IBET system, unauthorized clients and CSP have access to both IBE and IBBE ciphertexts. Definition 1 says that a secure IBET scheme can resist unauthorized access to the data encrypted in any ciphertext.

We proceed to define the LR-CA security of an IBET system against authorized clients and malicious CSP.

Setup: With security parameter λ and the maximum number m of authorized data consumers, the challenger $\mathcal{C}$ runs the Setup algorithm to generate system public parameters PP and master secret keys MSK. It gives PP to $\mathcal{A}$ and keeps MSK secret.

Queries: The adversary $\mathcal{A}$ submits an identity ID^* and adaptively issues the following queries to the challenger as in Definition 1:

- *User registration*(ID_i). The same as Definition 1.
- *Authorization*($ID^* \rightarrow S_i$). The same as Definition 1.

Challenge: At last, the adversary $\mathcal{A}$ outputs a private key SK' with regard to ID^*. We say that $\mathcal{A}$ wins the game if the following conditions hold.

- $\mathcal{A}$ has never queried the private key for ID^*.
- $SK' = SK_{ID^*}$.

The advantage of $\mathcal{A}$ in this game is defined as

$$Adv_{\mathcal{A}}^{LR-CA} = \Pr[SK' = SK_{ID^*}].$$

Definition 2. *An IBET system is LR-CA secure if all probabilistic polynomial-time adversaries $\mathcal{A}$ have at most a negligible advantage in the above game.*

4 An IBET Scheme

4.1 An Overview

It is challenging to achieve the mechanism that transforms a file allowing just one authorized visitor, into another file that allows multiple ones. At first sight, it seems that the original authorized visitor could employ IBBE to encrypt his private key for all the intended receivers, so that each one of them can obtain the private key and then decrypt the file just as the authorized visitor does. This, however, exposure of the authorized visitor's private key would lead to an unwanted access to outsourced data.

To achieve the encryption transformation while maintaining the secrecy of private keys, we introduce a *privacy-preserving authorization* method to the construction of IBET. Specifically, when generating an authorization token, the data owner blinds his private key by a random factor; CSP uses the authorization token to transform a file and obtains a transformed file that is the result of the plaintext blinded by the random factor. Only the authorized data consumers can obtain the random factor from the transformed file and then

recover the plaintext. In this way, the data owner's private key is well protected.

From a technical point of view, we follow Boneh and Boyen's identity-based encryption scheme [31] in our construction but compress the public parameters by reducing one element. We also employ Delerablée's identity-based broadcast encryption scheme [9] to achieve the multi-receiver functionality. The authorization token is generated by applying once the IBBE encryption and the transformed file is in Delerablée's IBBE-type ciphertext format.

4.2 Construction

In this section, we present our IBET construction built on bilinear groups. Table 1 summarizes the notations throughout the paper.

Suppose $\mathbb{G}$ and $\mathbb{G}_T$ are two (multiplicative) cyclic groups of prime order p. A bilinear map $e(\cdot,\cdot)$ is a map $\mathbb{G}\times\mathbb{G}\rightarrow\mathbb{G}_T$ which has the following properties: 1) *Bilinearity*: for all $g,h\in\mathbb{G}$ and all $a,b\in\mathbb{Z}_p$, $e(g^a,h^b)=e(g^b,h^a)=e(g,h)^{ab}$; 2) *Non-degeneracy*: $e(g,h)\neq 1$. We say that $\mathbb{G}$ is a bilinear group if the group operations in $\mathbb{G}$ and the bilinear map $e:\mathbb{G}\times\mathbb{G}\rightarrow\mathbb{G}_T$ can be efficiently computed.

Our IBET scheme will rely on the following complexity assumptions.

General Decisional Diffie-Hellman Exponent (GDDHE) assumption [9]. Suppose $\mathbb{G}$ is a cyclic group of prime order p and $g_0,h_0\in\mathbb{G}$. Let P and Q be two co-prime polynomials with pairwise distinct roots, of respective orders q and k. The GDDHE assumption says that given $(g_0,g_0^{\alpha},...,g_0^{\alpha^{q-1}},g_0^{\alpha P(\alpha)},g_0^{s\alpha P(\alpha)})\in\mathbb{G}^{q+2}$, $(h_0,h_0^{\alpha},...,h_0^{\alpha^{2k}},h_0^{sQ(\alpha)})\in\mathbb{G}^{2k+2}$ and $T\in\mathbb{G}_T$, the probability of any probabilistic polynomial-time (PPT) algorithm $\mathcal{A}$ in deciding whether T is equal to $e(g_0,h_0)^{sP(\alpha)}$ or is a random value of $\mathbb{G}_T$ is negligible.

A variation of q-SDH assumption [32]. We give a natural variation of the q-Strong-Diffie-Hellman (q-SDH) assumption. Suppose $\mathbb{G}$ is a cyclic group of prime order p. The variation of q-SDH assumption states that, given a tuple of elements $(g,g^x,g^{x^2},...,g^{x^q})\in\mathbb{G}^{q+1}$ and a *fixed* value $c\in\mathbb{Z}_p$, the probability of any PPT algorithm $\mathcal{A}$ in computing $g^{1/(x+c)}$ is negligible. We note that the value c is fixed in this version of q-SDH assumption, while, by contrast, it is freely chosen in the standard q-SDH assumption [32]. Boneh *et al.* has already discussed this variation of q-SDH assumption in [33], but for completeness we still give a proof (in Appendix A) that this variation of q-SDH assumption holds in any group where the q-SDH assumption holds.

4.2.1 System Setup

The trusted party RA generates two cyclic groups $\mathbb{G}$ and $\mathbb{G}_T$ of prime order $p>3$ and a bilinear map $e:\mathbb{G}\times\mathbb{G}\rightarrow\mathbb{G}_T$. RA chooses a random generator $g\in\mathbb{G}$ and random values $\alpha\in\mathbb{Z}_p^*$ and $h,u\in\mathbb{G}$. Then it computes $g_1=g^{\alpha}$ and $u^{\alpha},h^{\alpha},h^{\alpha^2},...,h^{\alpha^m}\in\mathbb{G}$, where m is set as the maximum size of the set of data consumers who can access the same data. RA also selects two cryptographic hash functions $H_0:\{0,1\}^*\rightarrow\mathbb{Z}_p^*$ and $H_1:\mathbb{G}_T\rightarrow\mathbb{G}$. The hash function H_0 can be implemented by applying standard hash functions such as SHA-2 and the hash function H_1 can be realized by using the MapToPoint encoding function [6]. Specifically, given the underlying elliptic curve (e.g., $y^2=x^3+1$ over $\mathbb{F}_q$, where $q=\ell p-1$ and p does not divide ℓ) of $\mathbb{G}$, for an input $X\in\mathbb{G}_T$, first use a hash function $G:\{0,1\}^*\rightarrow\mathbb{F}_q$ to map X to an element $y_0\in\mathbb{F}_q$ and then compute $x_0\in\mathbb{F}_q$ such that $Q=(x_0,y_0)$ is a point on the elliptic curve. Then take $Y=Q^{\ell}\in\mathbb{G}$ of order p as the output of $H_1(X)$. More details about MapToPoint can be found in [6]. The system public parameters and master secret key are defined as

$$PP=\left(g_1,u,u^{\alpha},h,h^{\alpha},h^{\alpha^2},\cdots,h^{\alpha^m},e(g,h),H_0(\cdot),H_1(\cdot)\right)$$

and $MSK=(g,\alpha)$.

TABLE 1
Notations

Symbol	Meaning
$\mathbb{G},\mathbb{G}_T$	Cyclic groups with bilinear map $e:\mathbb{G}\times\mathbb{G}\rightarrow\mathbb{G}_T$
p	The large prime order of groups $\mathbb{G}$ and $\mathbb{G}_T$
g	A generator of $\mathbb{G}$
PP	The system public parameters
MSK	The system master secret key
H_0,H_1	Two cryptographic hash functions
ID	An identity of a user, e.g., an email address
S	A set of different identities, i.e., $S=\{ID_i\}$
SK_{ID}	A private key for the user with identity ID
CT_{ID}	An IBE ciphertext in an original file
CT_S	An IBBE ciphertext in a transformed file
s,t,r	Random values in $\mathbb{Z}_p^*$
u,h	Random values in $\mathbb{G}$
m	the maximum number of data consumers who can access the same data
n	the number of data consumers specified by a data owner

4.2.2 User Registration

In this procedure, a user asks RA for joining in the system. RA first checks the validation of the requestor. If the user passes, RA generates an authorized credential (e.g., a private key). Suppose that the requesting user is associated with an identity ID. RA uses its master secret key and the hash function H_0 to compute

$$SK_{ID}=g^{\frac{1}{\alpha+H_0(ID)}}.$$

Then RA gives SK_{ID} to the user through a secure channel.

4.2.3 File Creation

When using the storage service provided by CSP to store data, the data owners encrypt their data and outsource the resulting files to CSP. The files are stored in ciphertext format and can only be accessed by authorized data consumers. In practice, *key encapsulation* is a typical technique to reduce the costs of encryption. In such technique, a data owner first encrypts his data via a symmetric encryption mechanism (e.g., AES) and then encrypts the symmetric encryption key with the asymmetric encryption. The performance of the asymmetric encryption is thus independent of the data size. Our IBET scheme also follows this technique. A data owner first picks a random symmetric key $M\in\mathbb{G}_T$ and uses it to encrypt the data to be outsourced to CSP. Then the data owner employs IBE encryption mechanism to encrypt M. According to different data sharing requirements, there are two cases where data owners encrypt M.

- *Case 1:* Some data should be accessed by only one user. For example, a mobile user encrypts his private photos to be stored in clouds and wants just himself to be able to access. In such case, the user (data owner) chooses a random value $s \in \mathbb{Z}_p^*$ and computes

$$C_0 = M \cdot e(g,h)^s,\ C_1 = h^{s(\alpha + H_0(ID))}.$$

  Then $CT_{ID} = (C_0, C_1)$ is the ciphertext for M, where ID is the identity of the intended data consumer.
- *Case 2:* Some data would be shared with multiple users but the identities of these users cannot be determined beforehand. For instance, a patient feels that his health records may be diagnosed by different doctors, but for now, he can just determine one doctor. In this case, the patient (data owner) chooses a random value $s \in \mathbb{Z}_p^*$ and computes

$$C_0 = M \cdot e(g,h)^s,\ C_1 = h^{s(\alpha + H_0(ID))},$$
$$C_2 = u^{s(\alpha + H_0(ID))}.$$

  Then $CT_{ID} = (C_0, C_1, C_2)$ is the ciphertext for M.

The ciphertext of case 2 includes one more component than that of case 1. This component is crucial for transformation. Thus, only the files created in case 2 can be transformed.

Finally, CT_{ID} and the encryption of data under M form the file outsourced to CSP.

4.2.4 Authorization

When a data owner (or an authorized data consumer) finds that additional users should be authorized to access the data encrypted in an outsourced file, the data owner authorizes CSP to transform the file so that all the intended users can access the data. To do so, the data owner generates an authorization token as follows. Suppose that $S = \{ID_i\}_{i=1}^{n \le m}$ is the set of the identities of all intended data consumers. The data owner chooses random values $t, r \in \mathbb{Z}_p^*$ and computes

$$d_1 = g_1^{-t}, \quad d_2 = h^{t \prod_{i=1}^{n} (\alpha + H_0(ID_i))}$$
$$d_3 = H_1(e(g,h)^t) \cdot h^r,\ \ d_4 = SK_{ID} \cdot u^{-r},$$

where SK_{ID} is the private key of the data owner. The authorization token is set as $TK_{ID \to S} = (d_1, d_2, d_3, d_4)$. Then the data owner sends $TK_{ID \to S}$ to CSP.

4.2.5 File Transformation

Receiving the data owner's authorization token, CSP starts to transform the specified file. In fact, CSP just needs to transform the IBE ciphertext (precisely, case 2 ciphertext) about the symmetric key of the file. The other part of the file, i.e., encryption of data under the symmetric key, remains unchanged (see Fig. 3). Given the authorization token $TK_{ID \to S} = (d_1, d_2, d_3, d_4)$ and the IBE ciphertext $CT_{ID} = (C_0, C_1, C_2)$, CSP transforms CT_{ID} to be $CT_S = (c_1, c_2, c_3, c_4, c_5)$ where $c_1 = d_1, c_2 = d_2, c_3 = d_3, c_4 = C_2$ and

$$\begin{aligned} c_5 &= C_0 / e(C_1, d_4) \\ &= M \cdot e(g,h)^s / e\left(h^{s(\alpha + H_0(ID))}, g^{\frac{1}{\alpha + H_0(ID)}} \cdot u^{-r}\right) \\ &= M \cdot e\left(h^{s(\alpha + H_0(ID))}, u^r\right). \end{aligned}$$

This transformed ciphertext CT_S is an IBBE-type ciphertext. Then, ciphertext CT_S and the (unchanged) encryption of data form a transformed file in CSP.

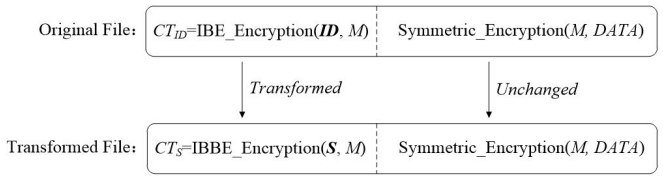

Fig. 3. File Transformation

4.2.6 File Access

There are two kinds of files in the system, i.e., the original files and the transformed files. The access about these two kinds of files are described as follows.

- *Original files:* An original file contains an IBE ciphertext of a symmetric key. For an IBE ciphertext $CT_{ID} = (C_0, C_1)$ (case 1 ciphertext) or $CT_{ID} = (C_0, C_1, C_2)$ (case 2 ciphertext) that is associated with identity ID, the data consumer with the same identity ID uses C_0 and C_1 to compute: $M = C_0 / e(SK_{ID}, C_1)$. Then the data owner uses the symmetric key M to finally recover the data.
- *Transformed files:* A transformed file contain an IBBE ciphertext that is converted from an original IBE ciphertext. For an IBBE ciphertext $CT_S = (c_1, c_2, c_3, c_4, c_5)$ associated with the identity set S, a data consumer with identity $ID_i \in S$ can compute

$$B = \left(e\left(c_1, h^{\Delta_{i,S}(\alpha)}\right) \cdot e\left(SK_{ID_i}, c_2\right)\right)^{\frac{1}{\prod_{j=1, j \neq i}^{n} H_0(ID_j)}}$$

  with $\Delta_{i,S}(\alpha) =$

$$\frac{1}{\alpha} \cdot \left(\prod_{j=1, j \neq i}^{n} (\alpha + H_0(ID_j)) - \prod_{j=1, j \neq i}^{n} H_0(ID_j) \right),$$

  and $h^r = c_3 / H_1(B)$. Then obtain $M = c_5 / e(h^r, c_4)$. Using symmetric key M, the data owner can finally recover the data.

5 Soundness and Security

In this section, we show that our IBET scheme is sound and enjoys the IND-sID-CPA and LR-CA security.

Theorem 1. *For any valid IBE ciphertext included in an original file, a data consumer having the correct private key always decrypts the ciphertext successfully. For any valid IBBE ciphertext included in a transformed file, all the data consumers having correct private keys always decrypt the ciphertext successfully.*

Proof: For a valid IBE ciphertext $CT_{ID} = (C_0, C_1)$ (we only consider case 1 ciphertext since only components C_0 and C_1 are used here), the data consumer who has the private key SK_{ID} can compute

$$\begin{aligned} M &= C_0 / e(SK_{ID}, C_1) \\ &= C_0 / e\left(g^{\frac{1}{\alpha + H_0(ID)}}, h^{s(\alpha + H_0(ID))}\right) \\ &= M \cdot e(g,h)^s / e(g,h)^s. \end{aligned}$$

For a valid IBBE ciphertext CT_S that is correctly transformed from an IBE ciphertext, any data consumer with identity ID_i included in S can first compute

$$\begin{aligned}B' &= \left(e\left(c_1, h^{\Delta_{i,S}(\alpha)}\right) \cdot e\left(SK_{ID_i}, c_2\right)\right) \\ &= e\left(g^{-\alpha t}, h^{\Delta_{i,S}(\alpha)}\right) \cdot e\left(g^{\frac{1}{\alpha+H_0(ID_i)}}, h^{t\prod_{j=1}^{n}(\alpha+H_0(ID_j))}\right) \\ &= e\left(g,h\right)^{-t\cdot\left(\prod_{j=1,j\neq i}^{n}(\alpha+H_0(ID_j))-\prod_{j=1,j\neq i}^{n}H_0(ID_j)\right)} \\ &\quad \cdot e\left(g,h\right)^{t\prod_{j=1,j\neq i}^{n}(\alpha+H_0(ID_j))} \\ &= e\left(g,h\right)^{t\prod_{j=1,j\neq i}^{n}H_0(ID_j)}.\end{aligned}$$

Then compute $B = B'^{\frac{1}{\prod_{j=1,j\neq i}^{n}H_0(ID_j)}} = e(g,h)^t$.

By having B, we have that

$$h^r = c_3/H_1(B) = H_1(e(g,h)^t)\cdot h^r/H_1(e(g,h)^t).$$

By having h^r, we have that

$$\begin{aligned}M &= c_5/e(h^r, c_4) \\ &= M\cdot e(h^{s(\alpha+H_0(ID))}, u^r)/e(h^r, u^{s(\alpha+H_0(ID))}).\end{aligned}$$

□

Theorem 2. *Suppose that the GDDHE assumption holds in bilinear groups. The proposed IBET scheme is IND-sID-CPA secure against adaptive impersonation and unauthorized access attacks. Specifically, neither CSP nor any client having no correct private keys can access any data encrypted in any outsourced file of his choice.*

Proof: Given in Appendix B.

□

Theorem 3. *Suppose that the variation of the q-SDH assumption holds in bilinear groups. The proposed IBET scheme offers LR-CA security for data owners against CSP and authorized clients. Specifically, for any authorization token with regard to any set S of identities, neither CSP nor any client specified in the set S can recover the private key of the data owner who generated the authorization token, and in this way access all the data of the owner.*

Proof: Suppose an adversary $\mathcal{A}$ can recover the private key of a data owner (ID^*) with probability ϵ, then we build an algorithm $\mathcal{B}$ to break the variation of the q-SDH assumption with probability ϵ. In the following, $\mathcal{B}$ perfectly simulates the challenger interacting with $\mathcal{A}$.

Setup: Given an instance $(g_0, g_0^{\alpha}, g_0^{\alpha^2}, ..., g_0^{\alpha^q}, c) \in \mathbb{G}^{q+1}\times\mathbb{Z}_p$ of the variation of q-SDH assumption, $\mathcal{B}$'s goal is to compute $g_0^{1/(\alpha+c)}$. $\mathcal{B}$ defines a univariate polynomial $f(x) = \prod_{i=1}^{q-1}(x+w_i)$ with randomly chosen values $\{w_i \in \mathbb{Z}_p^*\}$. Expand f and write $f(x) = \sum_{i=0}^{q-1} y_i x^i$, where $y_0, ..., y_{q-1} \in \mathbb{Z}_p$ are the coefficients of polynomial f. $\mathcal{B}$ chooses a random $\theta \in \mathbb{Z}_p$ and computes $g = \prod_{i=0}^{q-1}(g_0^{\alpha^i})^{y_i\theta} \in \mathbb{G}$ and $g^{\alpha} = \prod_{i=0}^{q-1}(g_0^{\alpha^{i+1}})^{y_i\theta} \in \mathbb{G}$, hence $g = g_0^{\theta f(\alpha)}$ and $g_1 = g_0^{\theta f(\alpha)\alpha}$. $\mathcal{B}$ also randomly chooses $\delta, \eta \in \mathbb{Z}_p$ and computes $u = g_0^{\delta}$, $u^{\alpha} = g_0^{\alpha\delta}$ and $h = g_0^{(\alpha+c)\eta}$. Hence, $e(g,h) = e(g_0^{\theta f(\alpha)}, g_0^{(\alpha+c)\eta})$. By deciding an integer $m < q$, $\mathcal{B}$ computes $h^{\alpha^i} = (g_0^{(\alpha+c)\eta})^{\alpha^i} = g_0^{\eta\alpha^{i+1}} \cdot g_0^{c\eta\alpha^i}$ for each $i \in [m]$. At last, $\mathcal{B}$ outputs the public parameters $PP = (g_1, u, u^{\alpha}, h, h^{\alpha}, h^{\alpha^2}, ..., h^{\alpha^m}, e(g,h))$. Note that $\mathcal{B}$ does not know the master secret key α.

Algorithm $\mathcal{B}$ models the hash functions $H_0 : \{0,1\}^* \to \mathbb{Z}_p^*$ and $H_1 : \mathbb{G}_T \to \mathbb{G}$ as random oracles. It starts by establishing a table T_0 of tuples (ID_i, w_i, SK_{ID_i}) and a table T_1 of tuples $(X_i \in \mathbb{G}_T, Y_i \in \mathbb{G})$. Table T_0 contains at the beginning $\{(*, w_i, *)\}_{i=1}^{q-1}$ and table T_1 is initialized to be empty. The number of identities that are queried to H_0 is less than $q - q_K$, with q_K the number of private key queries. For a hash query of H_0 on ID_i, if ID_i exists in T_0, return w_i; otherwise, choose an unused value w_i, return $H_0(ID_i) = w_i$ and record $(ID_i, w_i, *)$ on T_0. For a query of H_1 on $X_i \in \mathbb{G}_T$, if X_i exists in T_1, return Y_i; otherwise, return a random element $Y_i \in \mathbb{G}$ and record (X_i, Y_i) on T_1.

Queries: The adversary $\mathcal{A}$ submits an identity ID^* and adaptively issues the following queries to $\mathcal{B}$:

- *User registration*(ID_i). The adversary $\mathcal{A}$ submits $ID_i \neq ID^*$ for requesting private key for ID_i. If $\mathcal{A}$ has already queried private key for ID_i, $\mathcal{B}$ returns the corresponding private key SK_{ID_i} in T_0. Otherwise, if $\mathcal{A}$ has already queried the hash value for ID_i, $\mathcal{B}$ uses the corresponding w_i to compute the private key as follows. First, define $f_i(x) = f(x)/(x+w_i) = \prod_{j=1,j\neq i}^{q-1}(x+w_j)$. As before, we expand f_i and write $f_i(x) = \sum_{j=0}^{q-2} z_j x^j$ while calculating its coefficients. Then compute $SK_{ID_i} = \prod_{j=0}^{q-2}(g_0^{\alpha^j})^{z_j\theta}$, hence $SK_{ID_i} = g_0^{\theta f_i(\alpha)} = g^{1/(\alpha+H_0(ID_i))}$. Last, return SK_{ID_i} and complete table T_0 with SK_{ID_i} for ID_i. If neither the private key for ID_i nor the hash value for ID_i have already been queried, $\mathcal{B}$ sets $H_0(ID_i) = w_i$, computes SK_{ID_i} exactly as above and completes the table T_0 for ID_i.
- *Authorization*($ID^* \to S_i$). The adversary requests an authorization token from ID^* to $S_i = \{ID_j\}_{j=1}^{n}$. As response, $\mathcal{B}$ should give out $TK_{ID^*\to S_i}$. To do so, $\mathcal{B}$ defines $H_0(ID^*) = c$ and records $(ID^*, c, *)$ in T_0. Note that $\mathcal{B}$ cannot compute the private key $SK_{ID^*} = g_0^{\theta f(\alpha)/(\alpha+c)}$ since it does not have knowledge about α. Thus, $\mathcal{B}$ randomly chooses $t, r' \in \mathbb{Z}_p^*$ and computes $TK_{ID^*\to S_i} = (d_1, d_2, d_3, d_4)$, where $d_1 = g_1^{-t}, d_2 = h^{t\prod_{j=1}^{n}(\alpha+H_0(ID_j))}, d_3 = H_1(e(g,h)^t)\cdot(g_0^{\frac{\theta\eta}{\delta}f(\alpha)}\cdot h^{r'}), d_4 = u^{-r'}$. We note that $TK_{ID^*\to S_i}$ is a properly-distributed authorization token in $\mathcal{A}$'s view. To see this point, we define a random value $r = \frac{\theta}{\delta}\cdot\frac{f(\alpha)}{\alpha+c} + r' \in \mathbb{Z}_p$, and obtain

  $$h^r = (g_0^{(\alpha+c)\eta})^{\frac{\theta}{\delta}\cdot\frac{f(\alpha)}{\alpha+c}+r'} = g_0^{\frac{\theta\eta}{\delta}f(\alpha)}\cdot h^{r'}$$

  and

  $$\begin{aligned}SK_{ID^*}u^{-r} &= SK_{ID^*}\cdot g_0^{\delta(-\frac{\theta}{\delta}\frac{f(\alpha)}{(\alpha+c)}-r')} \\ &= SK_{ID^*}\cdot SK_{ID^*}^{-1}\cdot g_0^{-\delta r'} = u^{-r'}.\end{aligned}$$

  Then we have that $d_3 = H_1(e(g,h)^t)\cdot h^r$ and $d_4 = SK_{ID^*}u^{-r}$. Therefore $TK_{ID^*\to S_i} = (d_1, d_2, d_3, d_4)$ is a properly-distributed authorization token.

Challenge: $\mathcal{A}$ outputs a valid private key SK' for ID^*. Since a valid private key SK' for ID^* entails $e(h^{\alpha}\cdot h^c, SK') = e(g,h)$ where $c = H_0(ID^*)$, we deduce that $e(h^{\alpha+c}, SK') = e(g,h)$ and therefore

$$SK' = g^{\frac{1}{(\alpha+c)}} = g_0^{\theta\frac{f(\alpha)}{(\alpha+c)}}.$$

Similarly with the proof of Lemma 9 in [32], we use long division to compute the ratio $f(\alpha)/(\alpha+c)$ that appears in the exponent. Using long division, we rewrite the polynomial f as $f(x) = (x+c)\lambda(x) + \zeta$ for some easily computable polynomial $\lambda(x) = \sum_{i=0}^{q-2} \lambda_i x^i$ and constant $\zeta \in \mathbb{Z}_p$. Then the ratio $f(x)/(x+c)$ can be written as $f(x)/(x+c) = \sum_{i=0}^{q-2} \lambda_i x^i + \zeta/(x+c)$. Thus the private key SK' can be expressed as

$$SK' = g_0^{\theta\left(\sum_{i=0}^{q-2} \lambda_i \alpha^i + \frac{\zeta}{(\alpha+c)}\right)}.$$

Note that $\zeta \neq 0$ since $(x+c)$ does not divide $f(x)$. Then $\mathcal{B}$ can compute

$$\begin{aligned} D &= \left((SK')^{\theta^{-1}} \cdot \prod_{i=0}^{q-2} (g_0^{\alpha^i})^{-\lambda_i} \right)^{\zeta^{-1}} \\ &= \left(g_0^{\sum_{i=0}^{q-2} \lambda_i \alpha^i} \cdot g_0^{-\sum_{i=0}^{q-2} \lambda_i \alpha^i} \cdot g_0^{\frac{\zeta}{(\alpha+c)}} \right)^{\zeta^{-1}} \\ &= g_0^{\frac{1}{(\alpha+c)}}. \end{aligned}$$

$\mathcal{B}$ returns D as the required solution to the given instance of the variation of q-SDH assumption. Therefore, if there exists a PPT adversary that breaks the LR-CA security of the IBET scheme with advantage ϵ, then we can break the variation of q-SDH assumption with advantage ϵ.

□

6 Performance Evaluation

6.1 Theoretical Analysis

We summarize the computation overhead of every algorithm at each entity side in Table 2. We mainly consider the most expensive cryptographic operations, i.e., exponentiations and bilinear maps. In the table, we let t_e and t_p denote the evaluation time of an exponentiation operations in $\mathbb{G}$ and a bilinear pairing, respectively.

The computation cost of setup algorithm taken by RA is linear in the allowed maximum number m of the data consumers who can access the same data. In registration phase, RA only needs to perform one exponentiation in $\mathbb{G}$ to produce a private key. The Encrypt algorithm offers two ways for a data owner to secure data. If the data owner wants only one user (e.g., just himself) to access outsourced data, he generates the case 1 ciphertext, which takes him two exponentiations; if he would like to share data with more users in the future, he can generate the case 2 ciphertext, by taking just four exponentiations. When the data owner decides the identities of the users who can access his data, the owner takes the cost linear with the number n of these users to create an authorization token. The Transform algorithms takes CSP one pairing to transform a file. For an original file, the Decrypt algorithm takes a data consumer one bilinear pairing to decrypt; for a transformed file, the cost of the Decrypt algorithm is linear with the total number of authorized data consumers.

Table 3 further compares our IBET scheme with other related schemes, in terms of storage costs of client and CSP server and computation cost of token generation over bilinear groups, as well as some useful functionalities. In the table, $|\mathbb{Z}_p^*|, |\mathbb{G}|, |\mathbb{G}_T|$ denote the length of a value in $\mathbb{Z}_p^*, \mathbb{G}$ and $\mathbb{G}_T$, respectively. Matsuo's and Jiang *et al.*'s schemes support a cross-domain transformation that converts files generated in public-key encryption (PKE) system into files generated in IBE system, but their schemes require a user of PKE to store public parameters (public keys) with the size linear to the number (N) of intended data consumers who can access his data. This efficiency drawback is overcome by identity-based encryption mechanism which has been achieved in Xu *et al.*'s and ours schemes. In comparison with Xu *et al.*'s scheme, ours requires less public parameters to be stored at client side and achieves identity-based cross-domain transformation feature. This feature erases the limitation of transforming only in one encryption system. Moreover, it enables users to first choose efficient identity-based encryption mechanisms to protect data, and then transform the encrypted data (if they like) so that users from a different (IBBE) encryption system can access.

TABLE 2
Computation complexity of each algorithm in the IBET scheme

Algorithms	Computations	Entity
Setup	$(m+2)t_e + 1t_p$	RA
Register	$1t_e$	RA
Encrypt	$2t_e$ (case 1) or $4t_e$ (case 2)	Data owner
Authorize	$(n+4)t_e$	Data owner
Transform	$1t_p$	CSP
Decrypt	IBE: $1t_p$	Data consumer
	IBBE: $(n-1)\cdot t_e + 3t_p$	

6.2 Experimental Analysis

We conducted a series of experiments to evaluate the performance of the IBET scheme. The bilinear cryptographic operations are implemented by using the Standford PBC library (http://crypto.stanford.edu/pbc/). The elliptic curve is of Type A ($y^2 = x^3 + x$) so that p is a 160-bits prime and the size of an element of $\mathbb{G}$ is 256 bits. The details of hardware and software environments of our experiments are summarized in Table 4.

In the experiments, we first evaluated the performance of file creation and (original) file access. In particular, we implemented the efficient BB04 IBE scheme [31] to compare its performance with ours in terms of file creation and file access. We also followed the idea of key encapsulation in the experiments for backwards compatibility. That is, we first used symmetric keys of 256-bits AES to encrypt real data (about 1 KB), and then encrypted the symmetric keys via IBE. The file access process thus involves two steps: first obtaining the symmetric keys and then using the keys to recover data.

Table 5 shows that BB04 IBE and our IBET schemes require roughly the same computation times in file creation (about 60 msec) and file access (about 50msec) processes, respectively. This means that although our IBET scheme introduces the encryption transformation mechanism, the most frequently used file creation and file access processes have not been affected. Cloud clients will feel little difference in using our IBET scheme or an ordinary IBE scheme to outsource and access files.

TABLE 3
Comparison with related works in bilinear groups

Schemes	Costs at Client side			Costs at CSP side		①	②	③
	Public parameters storage	Private key storage	Token computation	Original file storage	Transformed file storage			
Matsuo[28]	PKE: $3N\|\mathbb{Z}_p^*\|$ IBE: $4\|\mathbb{G}\|+1\|\mathbb{G}_T\|$	PKE: $3\|\mathbb{Z}_p^*\|$ IBE: $2\|\mathbb{G}\|$	$1t_e$	$3\|\mathbb{G}\|+1\|\mathbb{G}_T\|$	$2\|\mathbb{G}\|+1\|\mathbb{G}_T\|$	√	×	×
Jiang *et al.*[30]	PKE: $7N\|\mathbb{Z}_p^*\|$ IBE: $5\|\mathbb{G}\|+1\|\mathbb{G}_T\|$	PKE: $2\|\mathbb{Z}_p^*\|$ IBE: $2\|\mathbb{Z}_p^*\|+1\|\mathbb{G}\|$	$6t_e+1t_p$	$4\|\mathbb{G}\|+3\|\mathbb{G}_T\|$	$2\|\mathbb{G}\|+2\|\mathbb{G}_T\|$	√	√	×
Xu *et al.*[23]	$(3m+2)\|\mathbb{G}\|+1\|\mathbb{G}_T\|$	$1\|\mathbb{G}\|$	$(n+5)t_e$	$3\|\mathbb{G}\|+1\|\mathbb{G}_T\|$	$4\|\mathbb{G}\|+1\|\mathbb{G}_T\|$	×	√	√
Our IBET	$(m+1)\|\mathbb{G}\|+1\|\mathbb{G}_T\|$	$1\|\mathbb{G}\|$	$(n+4)t_e$	$2\|\mathbb{G}\|+1\|\mathbb{G}_T\|$	$4\|\mathbb{G}\|+1\|\mathbb{G}_T\|$	√	√	√

Notations: ① means cross-domain transformation; ② means non-interactive transformation; ③ means identity-based setting.

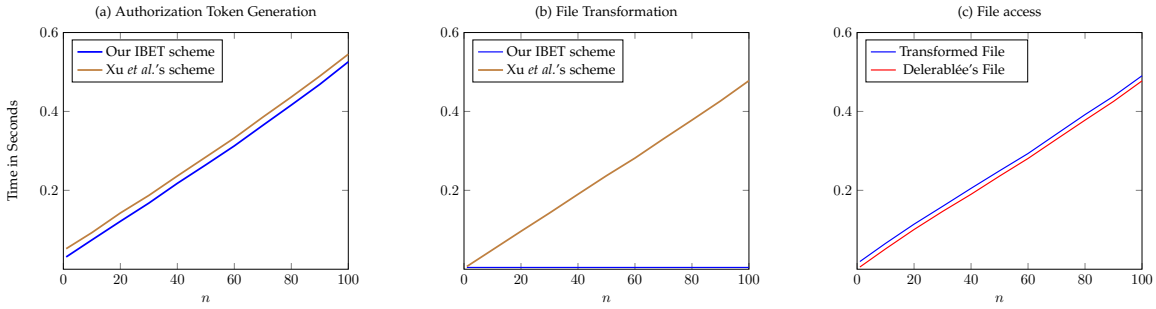

Fig. 4. Execution time of Authorization token generation, File transformation and (transformed) File access

TABLE 4
Experiment environments

Environment		Details
Hardware	CPU	Intel(R) Core(TM) i5-5200U @ 2.20GHz
	Memory	8GB
Software	Operating System	Microsoft Windows 10
	Compiler	Microsoft Visual C++ 6.0
	Program Library	Pairing Based Cryptography

TABLE 5
Execution time of File creation and (original) File access

Schemes	File Creation	File Access
BB04 IBE	56.411 ms	54.112 ms
Our IBET	60.786 ms	48.097 ms

Our IBET scheme provides the encryption transformation mechanism for data owners to transform original files into files in Delerablée's IBBE-type ciphertext format. Xu *et al.*'s scheme achieves a similar mechanism that transforms an IBBE-ciphertext-format file into another IBBE-ciphertext-format file. Thus, we additionally implemented Xu *et al.*'s scheme and Delerablée's IBBE scheme to conduct the experiments for authorization token generation, file transformation and (transformed) file access processes.

Fig. 4 (a) shows that the times cost by the generation of an authorization token of our IBET and Xu *et al.*'s schemes both grow linearly with the number (n) of authorized data owners. More precisely, our IBET scheme consumes a little (0.02112s) less time in this process. Fig. 4 (b) compares the times cost by the file transformation of our IBET and Xu *et al.*'s schemes. It can be seen that the time consumed in Xu *et al.*'s is linear with the number of authorized data consumers; while in ours the consumed time is constant (about 0.00458s). This is because that Xu *et al.*'s scheme actually transforms an IBBE ciphertext into another IBBE ciphertext, while ours transforms an IBE ciphertext. Fig. 4 (c) compares the times cost by the transformed file access of our IBET scheme and file access of Delerablée's scheme. It reveals that the time consumed in our transformed file access grows linearly with the number of authorized users, but is very close to that cost by the file access of Delerablée's IBBE scheme. This is a very attracting feature in that authorized users take almost no extra computations to access a transformed file; in other words, for the users who deploy IBBE scheme locally, there is no difference of time cost in accessing a transformed file or a file generated under ordinary IBBE encryption mechanism.

7 Conclusion

In this paper we studied how to securely and efficiently transform encrypted data in clouds. To address this issue, we proposed an identity-based encryption transformation (IBET) model, which connects the well-studied IBE and IBBE systems. IBET allows data owners to secure outsourced data with identity-based access control, which eliminates complicated cryptographic certificates for all users. Moreover, IBET provides a transformation mechanism for data owners to authorize cloud service provider (CSP) to transform a file in IBE-ciphertext formant into a file in IBBE-ciphertext format, so that a set of authorized users can access the underlying data. We proposed a concrete IBET scheme that is secure against powerful attacks. Thorough experimental analyses demonstrate the efficiency and practicability of the scheme.

ACKNOWLEDGMENTS

This paper is supported by the National Key Research and Development Program of China through projects 2017YFB0902900, 2017YFB0802500 and 2019QY(Y)0602; by the National Natural Science Foundation of China through projects 61902123, 61972058, 61932011, 61972019, 61872130, 61862012, 61772191, 61702028, 61772538, 61672083 and 91646203; by the China Postdoctoral Science Foundation through project 2019M662769; by the Science and Technology Key Projects of Hunan Province through projects 2015TP1004, 2016JC2012 and 2018TP1009; by the Science and Technology Key Projects of Changsha City through projects kq1801008 and kq1804008; by the National Cryptography Development Fund through project MMJJ20170106; by the Aeronautical Science Foundation of China through project 2017ZC51038; by the Foundation of Science and Technology on Information Assurance Laboratory through project 61421120305162112006; by Singapore National Research Foundation grant NRF2018NCR-NSOE004-0001 and AXA Research Fund; and by the Guangxi Natural Science Foundation under grant 2018GXNSFAA281232.

APPENDIX A
PROOF OF THE VARIATION OF q-SDH ASSUMPTION

We show that as long as the standard q-SDH assumption [32] holds in group $\mathbb{G}$, then the variation of q-SDH assumption also holds in $\mathbb{G}$. We firs review the standard q-SDH assumption. As shown in [32], the standard q-SDH assumption states that given a $(q+1)$-tuple of elements $(g, g^x, g^{x^2}, ..., g^{x^q}) \in \mathbb{G}^{q+1}$, there exists no PPT algorithm that outputs a pair $(c, g^{1/(x+c)})$ for a freely chosen value $c \in \mathbb{Z}_p \backslash \{-x\}$. Suppose that there exists a PPT algorithm $\mathcal{A}$ breaking the variation of q-SDH assumption with advantage ϵ, that is, on input $(g, g^x, g^{x^2}, ..., g^{x^q}, c) \in \mathbb{G}^{q+1} \times \mathbb{Z}_p$, $\mathcal{A}$ outputs $g^{1/(x+c)}$ with probability ϵ. Then we construct an algorithm $\mathcal{B}$ that breaks the standard q-SDH assumption with the advantage $\epsilon \cdot (1 - 1/p)$. $\mathcal{B}$ is given a q-SDH instance $(g, g^x, g^{x^2}, ..., g^{x^q})$ and its goal is to compute $(c, g^{1/(x+c)})$ for a freely chosen $c \in \mathbb{Z}_p \backslash \{-x\}$. To leverage $\mathcal{A}$, algorithm $\mathcal{B}$ picks a random $c \in \mathbb{Z}_p$ and forwards the tuple $(g, g^x, g^{x^2}, ..., g^{x^q}, c)$ as an instance of the variation of q-SDH to $\mathcal{A}$. $\mathcal{A}$ outputs $g^{1/(x+c)} \in \mathbb{G}$ as its solution. Then $\mathcal{B}$ outputs $(c, g^{1/(x+c)})$ correspondingly. We note that $\mathcal{B}$ successfully breaks the standard q-SDH assumption except for the case $c = -x$. The probability for $c = -x$ is $1/p$. Therefore, if $\mathcal{A}$ breaks the variation of q-SDH assumption with advantage ϵ, then $\mathcal{B}$ can break the standard q-SDH assumption [32] with advantage $\epsilon \cdot (1 - 1/p)$.

APPENDIX B
PROOF OF THEOREM 2

In the IBET scheme, a file is composed by a ciphertext of an asymmetric encryption (IBE or IBBE) and a ciphertext of a symmetric encryption (e.g, AES). Many symmetric encryptions have been well studied and extensively used in practice. We assume that the symmetric encryption can also well protect the data privacy in our scheme. Thus, we focus on proving the security of the asymmetric encryption.

Assume that an adversary $\mathcal{A}$ can access the symmetric key encrypted in a file with probability ϵ without using the correct key. We show that we can leverage $\mathcal{A}$'s output to break the GDDHE assumption with probability at most ϵ. To do so, we construct an algorithm $\mathcal{B}$ which simulates a challenger interacting with $\mathcal{A}$.

Algorithm $\mathcal{B}$ is given an instance of GDDHE: $(g_0, g_0^{\alpha}, ..., g_0^{\alpha^{q-1}}, g_0^{\alpha P(\alpha)}, g_0^{s\alpha P(\alpha)})$ and $(h_0, h_0^{\alpha}, ..., h_0^{\alpha^{2k}}, h_0^{sQ(\alpha)})$, as well as $T \in \mathbb{G}_T$ which is equal to $e(g_0, h_0)^{sP(\alpha)}$ or a random element of $\mathbb{G}_T$, where g_0, h_0 are generators of cyclic group $\mathbb{G}$, and $P(x) = \prod_{i=1}^{q}(x + w_i)$ and $Q(x) = \prod_{i=q+1}^{q+k}(x + w_i)$ are two polynomials with distinct roots picked from $\mathbb{Z}_p^*$.

Setup: The adversary $\mathcal{A}$ chooses a target identity ID^* and sends it to the challenger. The algorithm $\mathcal{B}$ sets $m = k$ as the allowed maximum number of data consumers who can access the same data. $\mathcal{B}$ then sets

$$g_1 = g_0^{\alpha P(\alpha)}, \quad h = h_0^{\prod_{i=q+2}^{q+k}(\alpha + w_i)},$$

$$h^{\alpha} = h_0^{\alpha \prod_{i=q+2}^{q+k}(\alpha + w_i)}, \ \ ..., \ \ h^{\alpha^k} = h_0^{\alpha^k \prod_{i=q+2}^{q+k}(\alpha + w_i)},$$

$$e(g, h) = e\left(g_0, h_0\right)^{P(\alpha) \prod_{i=q+2}^{q+k}(\alpha + w_i)}.$$

$\mathcal{B}$ chooses a random value $\gamma \in \mathbb{Z}_p^*$ and computes $u = h^{\gamma}, u^{\alpha} = h^{\alpha\gamma}$. To model hash functions H_0 and H_1, $\mathcal{B}$ establishes two tables T_0 and T_1 just the same as in the proof of Theorem 3. Then $\mathcal{B}$ defines the following public parameters PP and sends it to $\mathcal{A}$: $PP = \left(g_1, u, u^{\alpha}, h, h^{\alpha}, h^{\alpha^2}, \cdots, h^{\alpha^k}, e(g, h), H_0(\cdot), H_1(\cdot)\right)$. Note that $\mathcal{B}$ does not know the master secret key $MSK = (g = g_0^{P(\alpha)}, \alpha)$.

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

- *User registration*(ID_i). $\mathcal{A}$ queries the private key for identity $ID_i \neq ID^*$. If $\mathcal{A}$ has already queried the private key for ID_i, $\mathcal{B}$ returns the corresponding private key SK_{ID_i} in T_0. Otherwise, if $\mathcal{A}$ has already queried the hash value for ID_i, $\mathcal{B}$ uses the corresponding w_i to compute the private key

  $$SK_{ID_i} = g^{\frac{1}{\alpha + H_0(ID_i)}} = g_0^{\frac{P(\alpha)}{(\alpha + w_i)}}.$$

  Then $\mathcal{B}$ returns SK_{ID_i} to $\mathcal{A}$ and completes the table T_0 with SK_{ID_i} for ID_i. If neither the private key for ID_i nor the hash value for ID_i have already been queried, $\mathcal{B}$ sets $H_0(ID_i) = w_i$, computes SK_{ID_i} exactly as above and completes the table T_0 for ID_i.
- *Authorization*($ID_i \rightarrow S_i$). $\mathcal{B}$ initiates a table $L = (ID_i, S_i, TK_{ID_i \rightarrow S_i})$ to store related information about authorization tokens. For an authorization token query from ID_i to $S_i = \{ID_j\}_{j=1}^{n<k}$, if $(ID_i, S_i, TK_{ID_i \rightarrow S_i})$ already exists in table L, $\mathcal{B}$ returns $TK_{ID_i \rightarrow S_i}$ to $\mathcal{A}$. Otherwise, $\mathcal{B}$ first calls the registration algorithm to produce a private key for ID_i and then uses it to create authorization token $TK_{ID_i \rightarrow S_i}$. In doing so, if $ID_i \neq ID^*$, $\mathcal{B}$ computes the private key SK_{ID_i} as above; if $ID_i = ID^*$, however, SK_{ID^*} cannot be created due to the absence of $(\alpha + w_{q+1})$ in $P(\alpha)$. Therefore, we elaborate the responses of $\mathcal{B}$ into two cases:

- $ID_i \neq ID^*$: $\mathcal{B}$ first creates the private key SK_{ID_i} just as in user registration query. Then $\mathcal{B}$ picks random values $t, r \in \mathbb{Z}_p^*$ and computes $d_1 = g_1^{-t}, d_2 = h^t \prod_{j=1}^{n}(\alpha + H_0(ID_j))$, $d_3 = H_1(e(g,h)^t)h^r, d_4 = SK_{ID_i} \cdot u^{-r}$. Then $\mathcal{B}$ gives the authorization token $TK_{ID_i \to S_i} = (d_1, d_2, d_3, d_4)$ to $\mathcal{A}$ and records $(ID_i, S, TK_{ID_i \to S_i})$ in table L.
- $ID_i = ID^*$: In this case, $\mathcal{B}$ outputs a random authorization token instead of a well-formed one. It chooses random elements $d_4 \in \mathbb{G}$, $t, r \in \mathbb{Z}_p^*$ and computes $d_1 = g_1^{-t}, d_2 = h^t \prod_{j=1}^{n}(\alpha + H_0(ID_j))$, $d_3 = H_1(e(g,h)^t)h^r$. Then $\mathcal{B}$ gives $TK_{ID_i \to S_i} = (d_1, d_2, d_3, d_4)$ to $\mathcal{A}$ and records $(ID_i, S, TK_{ID_i \to S_i})$ in table L.

Challenge: When deciding Phase 1 is over, the adversary $\mathcal{A}$ outputs two messages M_0 and M_1. If there are a record $(ID^*, S_i, TK_{ID^* \to S_i})$ in table L and a record (ID_i, SK_{ID_i}) for any $ID_i \in S_i$ in table T_0, algorithm $\mathcal{B}$ aborts. Otherwise, $\mathcal{B}$ randomly chooses a bit $b \in \{0,1\}$ and computes

$$C_0 = M_b \cdot T^{\prod_{i=q+2}^{q+k} w_i} \cdot e(g_0^{s\alpha P(\alpha)}, h_0^{\Delta(\alpha)}),$$

$$C_1 = h_0^{sQ(\alpha)}, \;\; C_2 = C_1^{\gamma},$$

where $\Delta(\alpha) = \frac{1}{\alpha}\left(\prod_{i=q+2}^{q+k}(\alpha + w_i) - \prod_{i=q+2}^{q+k} w_i\right)$.

One can verify that

$$C_1 = h_0^{s \prod_{i=q+2}^{q+k}(\alpha + w_i) \cdot (\alpha + w_{q+1})} = h^{s(\alpha + H_0(ID^*))},$$

$$C_2 = \left(h^{s(\alpha + H_0(ID^*))}\right)^{\gamma} = u^{s(\alpha + H_0(ID^*))}.$$

Note that if $T = e(g_0, h_0)^{sP(\alpha)}$, then $C_0 = M_b e(g,h)^s$.

Then $\mathcal{B}$ gives $CT_{ID} = (C_0, C_1, C_2)$ to $\mathcal{A}$ as a challenge.

Phase 2: The same as Phase 1 with the restrictions described in Definition 1.

Guess: The adversary $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$ and $\mathcal{B}$ outputs the same bit.

We claim that the probability of $\mathcal{A}$ in distinguishing a well-formed authorization token from a random one is negligible. In the case $ID_i = ID^*$, $\mathcal{B}$ chooses a random $d_4 \in \mathbb{G}$. For such d_4, there must exist a value $r' \in \mathbb{Z}_p^*$ such that $d_4 = SK_{ID^*} \cdot u^{-r'}$. Then the random authorization token can be written as $TK_{ID^* \to S_i} = (g_1^{-t}, h^t \prod_{j=1}^{n}(\alpha + H_0(ID_j)), H_1(e(g,h)^t) \cdot h^r, SK_{ID^*} \cdot u^{-r'})$. On the other hand, the well-formed authorization token is of the form $(g_1^{-t'}, h^{t'} \prod_{j=1}^{n}(\alpha + H_0(ID_j)), H_1(e(g,h)^{t'}) \cdot h^{r'}, SK_{ID^*} \cdot u^{-r'})$. Then, distinguishing the random token from a well-formed one is identical to distinguishing the part $(g_1^{-t}, h^t \prod_{j=1}^{n}(\alpha + H_0(ID_j)), H_1(e(g,h)^t) \cdot h^r)$ from the part $(g_1^{-t'}, h^{t'} \prod_{j=1}^{n}(\alpha + H_0(ID_j)), H_1(e(g,h)^{t'}) \cdot h^{r'})$. Indeed, these two parts are the Delerablée's IBBE encryptions of h^r and $h^{r'}$, respectively. Since the probability of $\mathcal{A}$ in distinguishing Delerablée's IBBE ciphertexts has been proved to be negligible by Theorem 1 in [9], the probability of $\mathcal{A}$ in distinguishing a well-formed authorization token from a random one is negligible too. Therefore, if the adversary $\mathcal{A}$ can break the IND-sID-CPA security of the IBET scheme with probability ϵ, then we can use $\mathcal{A}$ to break the GDDHE assumption with probability at most ϵ, which completes the proof of Theorem 2.

REFERENCES

[1] D. Song, E. Shi, I. Fischer, and U. Shankar, "Cloud data protection for the masses," *Computer*, vol. 45, no. 1, pp. 39–45, 2012.

[2] J. Yu, K. Ren, and C. Wang, "Enabling cloud storage auditing with verifiable outsourcing of key updates," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1362–1375, 2016.

[3] H. Yin, Z. Qin, J. Zhang, L. Ou, and K. Li, "Achieving secure, universal, and fine-grained query results verification for secure search scheme over encrypted cloud data," *IEEE Transactions on Cloud Computing*, 2017.

[4] K. Li, W. Zhang, C. Yang, and N. Yu, "Security analysis on one-to-many order preserving encryption-based cloud data search," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1918–1926, 2015.

[5] R. Zhang, R. Xue, and L. Liu, "Searchable encryption for healthcare clouds: a survey," *IEEE Transactions on Services Computing*, vol. 11, no. 6, pp. 978–996, 2018.

[6] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *SIAM Journal on Computing*, vol. 32, no. 3, pp. 586–615, 2003.

[7] J. Wei, W. Liu, and X. Hu, "Secure data sharing in cloud computing using revocable-storage identity-based encryption," *IEEE Transactions on Cloud Computing*, 2016.

[8] D. He, N. Kumar, H. Wang, L. Wang, K.-K. R. Choo, and A. Vinel, "A provably-secure cross-domain handshake scheme with symptoms-matching for mobile healthcare social network," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 633–645, 2018.

[9] C. Delerablée, "Identity-based broadcast encryption with constant size ciphertexts and private keys," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2007, pp. 200–215.

[10] H. Deng, Q. Wu, B. Qin, W. Susilo, J. Liu, and W. Shi, "Asymmetric cross-cryptosystem re-encryption applicable to efficient and secure mobile access to outsourced data," in *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*. ACM, 2015, pp. 393–404.

[11] J. Lai, Y. Mu, F. Guo, W. Susilo, and R. Chen, "Anonymous identity-based broadcast encryption with revocation for file sharing," in *Australasian Conference on Information Security and Privacy*. Springer, 2016, pp. 223–239.

[12] J. Lai, Y. Mu, F. Guo, and R. Chen, "Fully privacy-preserving id-based broadcast encryption with authorization," *The Computer Journal*, vol. 60, no. 12, pp. 1809–1821, 2017.

[13] W. Susilo, R. Chen, F. Guo, G. Yang, Y. Mu, and Y.-W. Chow, "Recipient revocable identity-based broadcast encryption: how to revoke some recipients in ibbe without knowledge of the plaintext," in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*. ACM, 2016, pp. 201–210.

[14] J. Lai, Y. Mu, F. Guo, W. Susilo, and R. Chen, "Fully privacy-preserving and revocable id-based broadcast encryption for data access control in smart city," *Personal and Ubiquitous Computing*, vol. 21, no. 5, pp. 855–868, 2017.

[15] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in *EUROCRYPT 1998*. Springer Berlin Heidelberg, 1998, pp. 127–144.

[16] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *Information and System Security (TISSEC), ACM Transactions on*, vol. 9, no. 1, pp. 1–30, 2006.

[17] B. Libert and D. Vergnaud, "Unidirectional chosen-ciphertext secure proxy re-encryption," in *PKC 2008*. Springer Berlin Heidelberg, 2008, pp. 360–379.

[18] Z. Cao, H. Wang, and Y. Zhao, "Ap-pre: Autonomous path proxy re-encryption and its application," *IEEE Transactions on Dependable and Secure Computing*, 2017.

[19] H. Guo, Z. Zhang, J. Xu, N. An, and X. Lan, "Accountable proxy re-encryption for secure data sharing," *IEEE Transactions on Dependable and Secure Computing*, 2018.

[20] M. Green and G. Ateniese, "Identity-based proxy re-encryption," in *ACNS 2007*. Springer Berlin Heidelberg, 2007, pp. 288–306.

[21] C. K. Chu and W. G. Tzeng, "Identity-based proxy re-encryption without random oracles," in *ISC 2007*. Springer Berlin Heidelberg, 2007, pp. 189–202.

[22] K. Liang, J. K. Liu, D. S. Wong, and W. Susilo, "An efficient cloud-based revocable identity-based proxy re-encryption scheme for public clouds data sharing," in *European Symposium on Research in Computer Security*. Springer, 2014, pp. 257–272.
[23] P. Xu, T. Jiao, Q. Wu, W. Wang, and H. Jin, "Conditional identity-based broadcast proxy re-encryption and its application to cloud email," *IEEE Transactions on Computers*, vol. 65, no. 1, pp. 66–79, 2016.
[24] K. Liang, M. H. Au, J. K. Liu, W. Susilo, D. S. Wong, G. Yang, Y. Yu, and A. Yang, "A secure and efficient ciphertext-policy attribute-based proxy re-encryption for cloud data sharing," *Future Generation Computer Systems*, vol. 52, pp. 95–108, 2015.
[25] C. Ge, W. Susilo, L. Fang, J. Wang, and Y. Shi, "A cca-secure key-policy attribute-based proxy re-encryption in the adaptive corruption model for dropbox data sharing system," *Designs, Codes and Cryptography*, pp. 1–17, 2018.
[26] Y. Yang and M. Ma, "Conjunctive keyword search with designated tester and timing enabled proxy re-encryption function for e-health clouds," *IEEE Transactions Information Forensics and Security*, vol. 11, no. 4, pp. 746–759, 2016.
[27] K. Liang, M. H. Au, J. K. Liu, W. Susilo, D. S. Wong, G. Yang, T. V. X. Phuong, and Q. Xie, "A dfa-based functional proxy re-encryption scheme for secure public cloud data sharing," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 10, pp. 1667–1680, 2014.
[28] T. Matsuo, "Proxy re-encryption systems for identity-based encryption," in *Pairing 2007*. Springer Berlin Heidelberg, 2007, pp. 247–267.
[29] T. Mizuno and H. Doi, "Hybrid proxy re-encryption scheme for attribute-based encryption," in *International Conference on Information Security and Cryptology*. Springer, 2009, pp. 288–302.
[30] P. Jiang, J. Ning, K. Liang, C. Dong, J. Chen, and Z. Cao, "Encryption switching service: Securely switch your encrypted data to another format," *IEEE Transactions Services Computing*, 2018.
[31] D. Boneh and X. Boyen, "Efficient selective-id secure identity-based encryption without random oracles," in *EUROCRYPT 2004*. Springer Berlin Heidelberg, 2004, pp. 223–238.
[32] ——, "Short signatures without random oracles and the sdh assumption in bilinear groups," *Journal of Cryptology*, vol. 21, no. 2, pp. 149–177, 2008.
[33] D. Boneh, X. Boyen, and E. J. Goh, "Hierarchical identity based encryption with constant size ciphertext," in *EUROCRYPT 2005*. Springer Berlin Heidelberg, 2005, pp. 440–456.

Hua Deng received his MS degree in cryptography from Southwest Jiaotong University, China, in 2010 and Ph.D. degree in information security from Wuhan University, China, in 2015. Now he is an associate research fellow at the College of Computer Science and Electronic Engineering, Hunan University, China. His research interests include applied cryptography, data security and privacy, cloud security.

Zheng Qin received the Ph.D. degree in computer software and theory from Chongqing University, China, in 2001. From 2010 to 2011, he served as a visiting scholar with the Department of Computer Science, Michigan University. He is currently a professor with the College of Computer Science and Electronic Engineering, Hunan University, where he also serves as the Vice Dean. He also serves as the Director of the Hunan Key Laboratory of Big Data Research and Application. His main interests include network and data security, machine learning, and applied cryptography.

Qianhong Wu received the M.Sc. degree in applied mathematics from Sichuan University, Sichuan, China, and the Ph.D. degree in cryptography from Xidian University, Xi'an, China, in 2001 and 2004, respectively. Since then, he has been an Associate Research Fellow with the University of Wollongong, Wollongong, Australia, an Associate Professor with Wuhan University, Wuhan, China, and a Senior Researcher with the Universitat Rovira i Virgili, Tarragona, Catalonia. His research interests include cryptography, information security and privacy, and ad hoc network security. He has been a main researcher or project holder/coholder for more than ten Chinese-, Australian-, and Spanish-funded projects. He has authored over 60 publications and served on the program committees of several international conferences on information security and privacy.

Zhenyu Guan received his Ph.D. degree in electronic engineering from Imperial College London, UK, in 2013. Since then, he has joined Beihang University (Beijing) as a lecturer. He is a member of IEEE and IEICE. His current research interests include cryptography engineering, security of IoT, blockchain

Robert H. Deng is AXA Chair Professor of Cybersecurity, Director of the Secure Mobile Centre, and Deputy Dean for Faculty & Research, School of Information Systems, Singapore Management University (SMU). His research interests are in the areas of data security and privacy, network security, and system security. He received the Outstanding University Researcher Award from National University of Singapore, Lee Kuan Yew Fellowship for Research Excellence from SMU, and Asia-Pacific Information Security Leadership Achievements Community Service Star from International Information Systems Security Certification Consortium. He serves/served on many editorial boards and conference committees, including the editorial boards of ACM Transactions on Privacy and Security, IEEE Security & Privacy, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Information Forensics and Security, Journal of Computer Science and Technology, and Steering Committee Chair of the ACM Asia Conference on Computer and Communications Security. He is a Fellow of IEEE and Fellow of Academy of Engineering Singapore.

Yujue Wang received the Ph.D. degrees from Wuhan University, Wuhan, China, and City University of Hong Kong, Hong Kong, under the joint Ph.D. program, in 2015. He was a Research Fellow with the School of Information Systems, Singapore Management University. He is currently with the School of Computer Science and Information Security, Guilin University of Electronic Technology, China. His research interests include applied cryptography, database security and cloud computing security

Yunya Zhou received her BS degree in electronic information engineering from the College of Information Science and Engineering, Ocean University of China, China, in 2013 and MS degree in information security from the School of Electronic and Information Engineering, Beihang University, China, in 2016. Now she is a security engineer at State Grid Hunan Maintenance Company. Her research interests include applied cryptography and cyber-security.