Hindawi
Security and Communication Networks
Volume 2022, Article ID 4905763, 12 pages
https://doi.org/10.1155/2022/4905763

Research Article

Dual-Server Identity-Based Encryption with Authorized Equality Test for IoT Data in Clouds

Meng Zhao[1] and Yong Ding[1,2]

[1]*Guangxi Key Laboratory of Cryptography and Information Security, School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China*
[2]*Cyberspace Security Research Center, Pengcheng Laboratory, Shenzhen, China*

Correspondence should be addressed to Yong Ding; stone_dingy@126.com

Received 9 May 2022; Revised 20 August 2022; Accepted 6 September 2022; Published 11 October 2022

Academic Editor: AnMin Fu

The massive amounts of data collected by Internet of things (IoT) devices can be stored in clouds to solve the problem of the low storage capacity of IoT terminals. However, the privacy and security of outsourced IoT data may be compromised on the cloud side. Traditional cryptographic technologies can protect data privacy but require the user to retrieve the data for decryption and further processing, which would bring vast amounts of bandwidth and computation burden to users. This paper proposes a dual-server identity-based encryption scheme supporting authorized ciphertext equality test (DS-IBE-AET), where two noncolluding servers with authorizations from users can collaboratively carry out an equality test on outsourced IoT ciphertexts without decrypting the data. DS-IBE-AET can resist offline keyword guessing attacks confronted by existing encryption schemes with equality test in the single server model. Security analysis demonstrates that the proposed DS-IBE-AET scheme offers unforgeability for private keys of users and servers and confidentiality protection for outsourced IoT data and authentication tokens. The performance analysis indicates the practicality of our DS-IBE-AET construction for securing outsourced IoT data in clouds.

1. Introduction

With the advancement of cloud computing, various types of user data produced in the Internet of things, Internet of vehicles, smart grid and other applications can be maintained in the cloud to reduce the local storage costs. In order to protect data privacy on cloud servers, the most common and effective method is to encrypt data, then upload encrypted data to servers. Traditional data encryption technologies can ensure data confidentiality; however, they would make encrypted data unsearchable and incomparable [1, 2]. Thus, users have to retrieve the data from remote cloud servers, then decrypt it for processing. This will not be able to take advantage of the powerful computing resources of the cloud server and bring huge computing overhead to users.

To solve this problem, Boneh et al. [3] proposed a public key encryption scheme with keyword search (PEKS), where the keyword is encrypted and outsourced along with the encrypted message so that it can be compared with the encrypted trapdoor for realizing privacy-preserving search over the outsourced data. Particularly, the search process relies on the equality test between the encrypted keyword and trapdoor. In 2010, Yang et al. [4] presented a public key encryption scheme withequality test (PKEET), which allowed the cloud server to check whether two ciphertexts had the same plaintext without decryption. Here, these ciphertexts may be generated by different users with different public keys. Since then, many variants supporting ciphertext equality tests with different functions and characteristics have been introduced [5–8].

However, most of these schemes supporting equality tests on outsourced ciphertexts are proposed in the single server model, which cannot resist offline keyword guessing attacks. That is, the cloud server is able to generate ciphertext for any message in the message space in the public key setting, then after being authorized, it can perform the

equality test procedure with outsourced ciphertexts. In this way, the cloud server would find all the ciphertexts that encrypted the chosen message through the equality test procedure. Therefore, the confidentiality of these outsourced ciphertexts is compromised. To address this issue, Zhao et al. [9] proposed a public key encryption scheme with authorized equality test in the dual-server model, where the outsourced data is only stored at the primary server and two servers would not collude to launch attacks against user data. However, since their scheme is designed for as public key setting, they confront complex certificate management problems. Also, Wu et al. [10] designed an identity-based scheme supporting equality test in a dual-server model, while the privacy of the authentication token was not considered.

1.1. Our Contributions. In this paper, we propose a dual-server identity-based encryption scheme supporting authorized equality tests on outsourced IoT ciphertexts (DS-IBE-AET). As in the dual-server model of [9, 10], the front server and back server would not launch collusion attacks to compromise the confidentiality of outsourced IoT data, where these data are only kept at the front server. The equality test procedures can be executed in sequence only after both servers have obtained user authorizations, which can also be conducted in a multiuser setting with the authorizations from different users. The back server can only get internal test results from the front server, which makes it impossible to deduce the information from user data.

In our DS-IBE-AET construction, the encrypted authorization tokens for two servers are in the same format, which should be decrypted using the respective secret key for performing equality test procedures. Compared to [9, 10], our DS-IBE-AET construction designed for the identity-based setting avoids the burden of certificate management. Security analysis demonstrates that the proposed DS-IBE-AET construction guarantees the unforgeability of users' and servers' private keys, the privacy of outsourced data against two servers, as well as the privacy of authorization tokens. The performance analysis indicates that the proposed DS-IBE-AET construction is practical in IoT-related applications.

1.2. Related Works. Public key encryption with equality test is closely related to PEKS. Boneh et al. [3] introduced PEKS to allow the e-mail gateway to test whether the e-mail contained some special keywords, where the gateway did not need to decrypt emails. The main idea behind PKES is to test the equality of the encrypted keywords and trapdoor. PKEET was first introduced by Yang et al. [4], which allows any entity to perform an equality test on two ciphertexts to determine whether they were generated by the same plaintext, where the ciphertexts may be produced with different public keys. The ciphertext equality test technology has been extensively used in different scenarios, for example, privacy-preserving equi-join in relational databases [7, 11], secure deduplication on cloud data [12, 13], implicit authentication [14], and privacy-preserving road condition monitoring [15].

Since the outsourced ciphertexts can be publicly compared in Yang et al.'s PKEET [4], many solutions supporting authentication mechanisms have been developed. Tang introduced the AoN-PKEET [16] and FG-PKEET [17] to realize coarse-grained and fine-grained authorization for ciphertext equality tests, respectively. Wang et al. [18] presented a public key signcryption scheme with designated equality test to secure messaging services. Lee et al. [19] presented a generic PKEET construction by employing a two-level hierarchical identity-based encryption scheme, a strongly unforgeable one-time signature scheme, and a cryptographic hash function, whose security can be proved in the standard model. Attribute-based and proxy encryption schemes supporting authorized equality testing on ciphertexts had been studied in [20, 21], respectively. Compared with our DS-IBE-AET construction in the dual-server model, these schemes were designed in a public key setting, which faced the complex certificate management problem and cannot resist offline keyword guessing attacks.

Many identity-based encryption schemes (IBE) with ciphertext equality tests (IBEET) have been presented, which can mitigate the complexity of public key certificate management in PKEET. Ma [22] first introduced IBEET by combining PKEET and IBE. Wu et al. [23] put forward a novel IBEET scheme, in which users are divided into different groups, and only the users in the same group can generate ciphertexts with the shared secret token. In [24], Lee et al. noted that Wu et al.'s scheme [23] cannot resist insider attack and presented an improved IBEET construction. Alornyo et al. [25] constructed an IBEET from witness-based encryption technology to resist insider attacks, which offered weak indistinguishability under chosen ciphertext attacks in the random oracle model. Ling et al. [26] introduced group IBEET, where only the group administrator was able to issue authorization tokens to the tester. Compared with our DS-IBE-AET construction in the dual-server model, these schemes engage only a single cloud server, which cannot resist offline keyword guessing attacks.

The dual-server model has been generally employed in designing secure and privacy-preserving systems for resisting keyword guessing attacks launched by cloud servers, where two semitrusted servers would not collude with each other [27]. In [28], Tang introduced an amended FG-PKEET scheme in the two-proxy setting, where the equality test procedure had to be interactively carried out by two proxies. In this way, the ciphertexts can be protected against offline message recovery attacks. Wu et al. [10] proposed a dual-server identity-based encryption with equality test for mobile health social networks, in which two servers with authentication tokens can collaborate to complete the ciphertext equality test, while the privacy of the authentication token was not considered. Recently, Zhao et al. [9] proposed a public key encryption construction supporting authorized equality test on outsourced IoT data in a non-colluding dual-server model. Compared with [9], our DS-IBE-AET construction is developed in an identity-based setting, which can avoid the complex certificate management problem.

1.3. Paper Organization. The remainder of this paper is structured as follows. Section 2 reviews the preliminaries. The system model, security requirements, and system framework are introduced in Section 3. A concrete DS-IBE-AET scheme is presented in Section 4, while security and performance are analyzed in Section 5. Finally, the paper is concluded in Section 6.

2. Preliminaries

2.1. Bilinear Groups. Suppose $\mathbb{G} = <g>$ and $\mathbb{G}_T$ are two cyclic groups of prime order q. The mapping $\hat{e}$: $\mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_T$ is a bilinear pairing if the following conditions are satisfied:

2.1.1. Bilinearity. For any $g_1, g_2 \in_R \mathbb{G}$ and $\alpha, \beta \in_R \mathbb{Z}_q^*$,

$$\hat{e}\left(g_1^{\alpha}, g_2^{\beta}\right) = \hat{e}\left(g_1, g_2\right)^{\alpha\beta}. \tag{1}$$

2.1.2. Non-Degeneracy. There exists $g_1, g_2 \in \mathbb{G}$ such that,

$$\hat{e}\left(g_1, g_2\right) \neq 1. \tag{2}$$

2.1.3. Efficiency. For $g_1, g_2 \in_R \mathbb{G}$, there exists an efficient algorithm to compute $\hat{e}(g_1, g_2)$.

2.2. Complexity Assumptions. The security of our DS-IBE-AET construction relies on the following complexity assumptions.

CDH assumption. Suppose $\mathbb{G} = <g>$ is a cyclic group of prime order q. Given a tuple (g, g^a, g^b) where $a, b \in_R \mathbb{Z}_q^*$, no probabilistic polynomial-time algorithm $\mathcal{A}$ can compute g^{ab} with nonnegligible probability.

CBDH assumption. Suppose $\mathbb{G} = <g>$ and $\mathbb{G}_T$ are two cyclic groups of prime order q and satisfy bilinear pairing $\hat{e}$: $\mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_T$. Given a tuple (g, g^a, g^b, g^c) where $a, b, c \in_R \mathbb{Z}_q^*$, no probabilistic polynomial-time algorithm $\mathcal{A}$ can compute $\hat{e}(g, g)^{abc}$ with nonnegligible probability.

3. System Model and Security Requirements

3.1. System Model. As shown in Figure 1, in a DS-IBE-AET system, there are three types of entities, namely, a key generation center (KGC), users, and servers. KGC is an honest entity that is responsible for initializing the DS-IBE-AET system by producing the master private key and public parameters. It also issues the private keys for all users , the front server S_f and back server S_b according to their identities, respectively.

In the DS-IBE-AET system, both the data sender and the data recipient are system users. The data sender encrypts the data using the identity of the data recipient and the system public parameters, and the generated ciphertexts are only sent to the front server S_f for storage. The data recipient is able to retrieve ciphertexts from the front server S_f and run the decryption procedure using his/her private key. Also, a data recipient is able to authorize the front server S_f and back server S_b to perform equality test on his/her ciphertexts without decryption. The authorization tokens are encrypted using the identities of two servers, so that they can only be decrypted by the two servers.

The front server S_f has huge storage resources for maintaining user data in ciphertext format. Both the front server S_f and back server S_b have powerful computing capabilities for collaboratively performing equality tests on user ciphertexts after being authorized. With the authorizations from users, the front server S_f is able to generate internal results of equality tests on ciphertexts, which are then sent to the back server S_b to further confirm whether two ciphertexts encrypt the same message. The authorization tokens only allow two servers to collaboratively perform equality tests on users' ciphertexts.

3.2. Security Requirements. In the DS-IBE-AET system, the front server and back server would not launch collusion attacks to compromise the privacy of user ciphertexts. A secure DS-IBE-AET system must meet the following security conditions:

3.2.1. Unforgeability of User Private Key. The private key generated by KGC for user cannot be forged by any entity.

3.2.2. Unforgeability of Server Private Key. The private keys generated by KGC for the front server S_f and back server S_b cannot be forged by any entity.

3.2.3. Data Privacy against the front Server. The front server S_f cannot deduce the private information of users from the stored ciphertexts before and after being authorized by users to perform equality tests.

3.2.4. Data Privacy against the Back Server. After obtaining the users' authorizations for performing the equality test, the back server S_b cannot deduce the private information of users from the received internal results.

3.2.5. Privacy Protection on Authentication Token. The authentication tokens generated for the front server S_f and back server S_b can only be decrypted by themselves, respectively.

3.3. System Framework. A DS-IBE-AET scheme consists of nine polynomial-time procedures, namely, Setup, UKeyExt, SKeyExt, Encrypt, Decrypt, Authen, DecAuth, $EqTest_f$, and $EqTest_b$.

3.3.1. Setup. On input of the security parameter δ, the system setup procedure, which is run by KGC, generates the system master private key mpk and system public parameters param. We denote $(\text{mpk}, \text{param}) \leftarrow \text{Setup}(1^{\delta})$. Note that param is the implicit input for the following eight procedures.

3.3.2. UKeyExt. On input of the master private key mpk and a user identity ID_i, the user key extraction procedure, which is run by KGC, generates a private key usk_i for user ID_i. We denote $usk_i \leftarrow U\text{KeyExt}(mpk, ID_i)$.

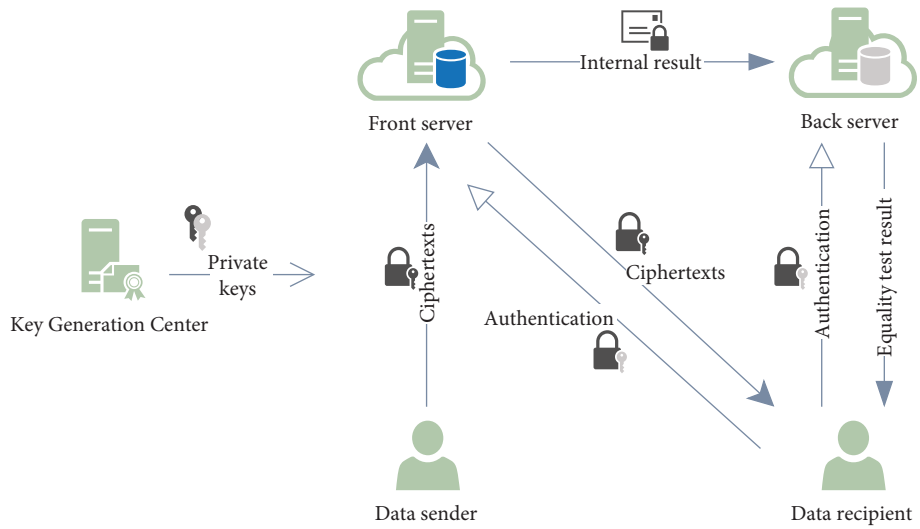

Figure 1: A system model of DS-IBE-AET.

3.3.3. SKeyExt. On input of the master private key mpk and the identity S_f of the front server (resp. S_b of the back server), the server key extraction procedure, which is run by KGC, generates a private key ssk_f for the front server S_f (resp. ssk_b for the back server S_b). We denote $ssk_f/ssk_b \leftarrow \text{SKeyExt}(mpk, S_f/S_b)$.

3.3.4. Encrypt. On input of the identity ID_i of the data recipient and a message m, the data encryption procedure, which is performed by the data sender, generates a ciphertext c and sends it to the front server S_f. We denote $c \leftarrow \text{Encrypt}(ID_i, m)$.

3.3.5. Decrypt. On input of the private key usk_i of the data recipient ID_i and a ciphertext c, the data decryption procedure, which is performed by data recipient, outputs a plaintext m or $\perp$ that signifies an error in decryption. We denote $m/\perp \leftarrow \text{Decrypt}(usk_i, c)$.

3.3.6. Authen. On input of the private key usk_i of user ID_i and the identities (S_f, S_b) of the front server and back server, the authentication token generation procedure, which is carried out by the user ID_i, generates ciphertext authentication tokens $\hat{t}_{i,f}$ and $\hat{t}_{i,b}$ for two servers. Note that the tokens $\hat{t}_{i,f}$ and $\hat{t}_{i,b}$ are sent to the front server S_f and back server S_b, respectively. We denote $(\hat{t}_{i,f}, \hat{t}_{i,b}) \leftarrow \text{Authen}(usk_i, S_f, S_b)$.

3.3.7. DecAuth. On input of the private key ssk_f of the front server S_f (resp. ssk_b of the back server S_b) and a ciphertext authentication token $\hat{t}_{i,f}$ (resp. $\hat{t}_{i,b}$), the authentication decryption procedure, which is performed by the front server S_f (resp. the back server S_b), outputs a plaintext authentication token $\tau_{i,f}$ (resp. $\tau_{i,b}$) or $\perp$ that signifies an error in decryption. We denote $\tau_{i,f}/\perp \leftarrow \text{DecAuth}(ssk_f, \hat{t}_{i,f})$ for the front server S_f and $\tau_{i,b}/\perp \leftarrow \text{DecAuth}(ssk_b, \hat{t}_{i,b})$ for the back server S_b.

3.3.8. $EqTest_f$. On input of the plaintext authentication tokens $\tau_{i,f}$ and $\tau_{j,f}$ of two users ID_i and ID_j, respectively, and their ciphertexts c and c', the front equality test procedure, which is performed by the front server S_f, outputs an internal result Γ and sends it to the back server S_b. We denote $\Gamma \leftarrow \text{EqTest}_f(\tau_{i,f}, \tau_{j,f}, c, c')$.

3.3.9. $EqTest_b$. On input of the plaintext authentication tokens $\tau_{i,b}$ and $\tau_{j,b}$ of two users ID_i and ID_j, respectively, and an internal result Γ, the back equality test procedure, which is performed by the back server S_b, outputs 1 if c and c' encrypt the same message or 0 otherwise. We denote $1/0 \leftarrow \text{EqTest}_b(\tau_{i,b}, \tau_{j,b}, \Gamma)$.

A DS-IBE-AET construction must be sound in the sense that if the procedures are performed honestly, the following conditions hold:

(i) The private key extracted by KGC for some users can be validated by such a user.

(ii) The private key extracted by KGC for each server can be validated by such a server.

(iii) The ciphertext generated by the data encryption procedure can be decrypted by the data decryption procedure.

(iv) The ciphertext authentication token generated by the authentication token generation procedure can be decrypted by the authentication decryption procedure.

(v) For any two ciphertexts that encrypt the same message, which may belong to different users, the front and back equality test procedures can collaboratively output 1.

(vi) For any two ciphertexts that encrypt different messages, which may belong to different users, the front and back equality test procedures collaboratively output 0 with overwhelming probability.

Definition 1. (Soundness): A DS-IBE-AET construction is sound if, for any security parameter δ, any master private key and public parameters $(mpk, \text{param}) \leftarrow \text{Setup}(1^{\delta})$, any private keys $usk_i \leftarrow U\text{KeyExt}(mpk, ID_i)$, $usk_j \leftarrow U\text{KeyExt}(mpk, ID_j)$ of two users ID_i and ID_j, any private key of the front server $ssk_f \leftarrow S\text{KeyExt}(mpk, S_f)$, and any private key of the back server $ssk_b \leftarrow S\text{KeyExt}(mpk, S_b)$, the following conditions are satisfied:

(i) The private key usk_i can be verified as valid in the verification step by the user ID_i.

(ii) The private key ssk_f can be verified as valid in the verification step by the front server S_f, and the private key ssk_b can be verified as valid in the verification step by the back server S_b.

(iii) For any message m, $\text{Decrypt}(usk_i, \text{Encrypt}(ID_i, m)) = m$.

(iv) $\text{DecAuth}(ssk_f, \widehat{t}_{i,f}) = \tau_{i,f}$ and $\text{DecAuth}(ssk_b, \widehat{t}_{i,b}) = \tau_{i,b}$, where $(\widehat{t}_{i,f}, \widehat{t}_{i,b}) \leftarrow \text{Authen}(usk_i, S_f, S_b)$.

(v) For any two messages m, m' such that $c \leftarrow \text{Encrypt}(ID_i, m)$ and $c' \leftarrow \text{Encrypt}(ID_i, m')$, if $m = m'$, then $\text{EqTest}_b(\tau_{i,b}, \tau_{j,b}, \Gamma) = 1$, otherwise $\Pr[\text{EqTest}_b(\tau_{i,b}, \tau_{j,b}, \Gamma) = 0] \geq 1 - \varepsilon(\cdot)$, where $\Gamma \leftarrow \text{EqTest}_f(\tau_{i,f}, \tau_{j,f}, c, c')$, $\tau_{i,f} = \text{DecAuth}(ssk_f, \widehat{t}_{i,f})$, $\tau_{i,b} = \text{DecAuth}(ssk_b, \widehat{t}_{i,b})$, $\tau_{j,f} = \text{DecAuth}(ssk_f, \widehat{t}_{j,f})$, $\tau_{j,b} = \text{DecAuth}(ssk_b, \widehat{t}_{j,b})$, $(\widehat{t}_{i,f}, \widehat{t}_{i,b}) \leftarrow \text{Authen}(usk_i, S_f, S_b)$, $(\widehat{t}_{j,f}, \widehat{t}_{j,b}) \leftarrow \text{Authen}(usk_j, S_f, S_b)$, and $\varepsilon(\cdot)$ represents a negligible function.

4. Concrete DS-IBE-AET Construction

This section presents a concrete DS-IBE-AET construction in bilinear groups, where a running process is shown in Figure 2, and the frequently used symbols are summarized in Table 1.

4.1. System Setup. Given a security parameter δ, KGC chooses cyclic groups $\mathbb{G} = \langle g \rangle$ and $\mathbb{G}_T$ satisfying bilinear mapping $\widehat{e}: \mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_T$, where groups $\mathbb{G}$ and $\mathbb{G}_T$ have prime order q. KGC selects four cryptographic hash functions $H_1: \{0,1\}^* \longrightarrow \mathbb{G}$, $H_2: \{0,1\}^{\xi_m} \longrightarrow \mathbb{G}$, $H_3: \mathbb{G}_T \longrightarrow \mathbb{G}$ and $H_4: \mathbb{G}_T \times \mathbb{G} \longrightarrow \{0,1\}^{\xi_{\mathbb{G}} + \log_q}$, where $\xi_{\mathbb{G}}$ and ξ_m, respectively, denote the element size in group $\mathbb{G}$ and message space. Also, KGC picks three random elements $d_1, d_2, d_3 \in \mathbb{Z}_q^*$ and computes the following:

$$\begin{aligned} V_1 &= g^{d_1}, \\ V_2 &= g^{d_2}, \\ V_3 &= g^{d_3}. \end{aligned} \tag{3}$$

At last, KGC keeps the master private key $mpk = (d_1, d_2, d_3)$ secret and publishes the public parameter $\text{param} = (\delta, \mathbb{G}, \mathbb{G}_T, q, \widehat{e}, g, H_1, H_2, H_3, H_4, V_1, V_2, V_3)$.

4.2. User Key Extraction. Given the identity of user ID_i, KGC generates the private key $usk_i = (usk_{i,1}, usk_{i,2}, usk_{i,3})$ as follows:

$$\begin{aligned} usk_{i,1} &= H_1(ID_i)^{d_1}, \\ usk_{i,2} &= H_1(ID_i)^{d_2}, \\ usk_{i,3} &= H_1(ID_i)^{d_3}. \end{aligned} \tag{4}$$

The private key usk_i is sent to the user ID_i via secure channel. Note that the user ID_i is able to validate usk_i as follows:

$$\widehat{e}(usk_{i,1}, g) = \widehat{e}(H_1(ID_i), V_1), \tag{5}$$

$$\widehat{e}(usk_{i,2}, g) = \widehat{e}(H_1(ID_i), V_2), \tag{6}$$

$$\widehat{e}(usk_{i,3}, g) = \widehat{e}(H_1(ID_i), V_3). \tag{7}$$

4.3. Server Key Extraction. Given the identity of the front server S_f, KGC generates the private key as follows:

$$ssk_f = H_1(S_f)^{d_1}, \tag{8}$$

which is sent to the front server S_f via secure channel. Note that the front server S_f is able to validate ssk_f as follows:

$$\widehat{e}(ssk_f, g) = \widehat{e}(H_1(S_f), V_1). \tag{9}$$

Similarly, KGC can generate the private key for the back server S_b as follows:

$$ssk_b = H_1(S_b)^{d_1}, \tag{10}$$

and the back server S_b is able to validate ssk_b as follows:

$$\widehat{e}(ssk_b, g) = \widehat{e}(H_1(S_b), V_1). \tag{11}$$

4.4. Data Encryption. For a message $m \in \{0,1\}^{\xi_m}$, the sender randomly picks an element $\alpha \in \mathbb{Z}_q^*$, and computes the ciphertext $c = (c_1, c_2, c_3)$, where

$$\begin{aligned} c_1 &= g^{\alpha}, \\ c_2 &= H_2(m) \cdot H_3(\widehat{e}(H_1(ID_i)), V_1)^{\alpha} \\ &\quad \cdot H_3(\widehat{e}(H_1(ID_i)), V_2)^{\alpha}, \\ c_3 &= (m \parallel \alpha) \oplus H_4(\widehat{e}(H_1(ID_i)), V_3)^{\alpha} \parallel H_2(m)). \end{aligned} \tag{12}$$

The ciphertext c is sent to the front server S_f.

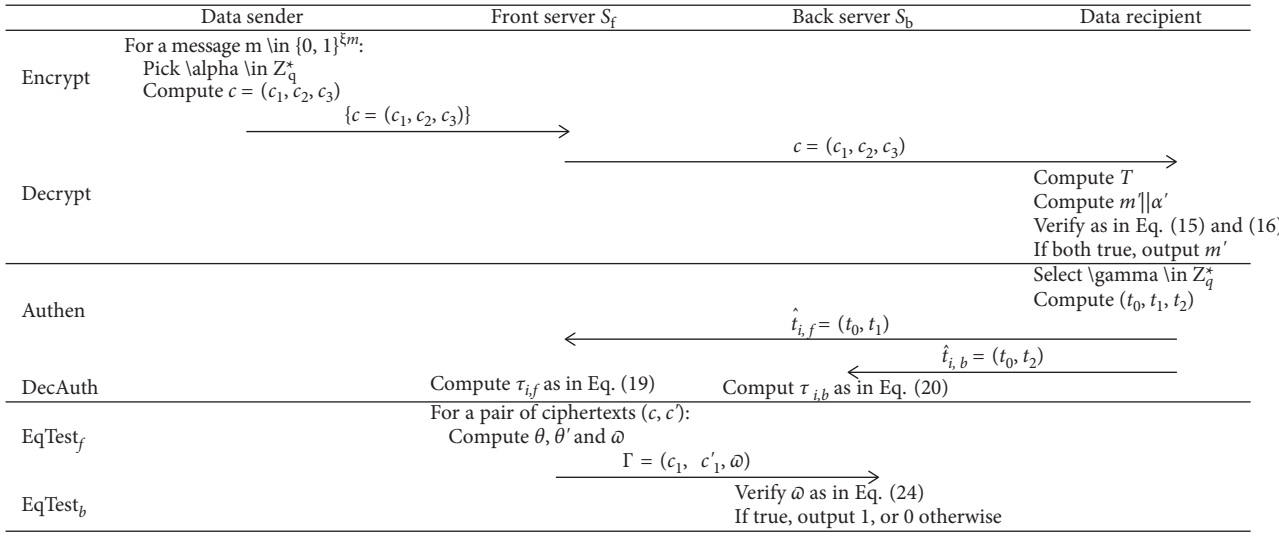

Figure 2: A running process of the proposed DS-IBE-AET construction.

Table 1: Notations.

Symbol	Meaning
δ	Security parameter
$\mathbb{G}$, $\mathbb{G}_T$	Cyclic groups of prime order q satisfying bilinear pairing $\hat{e}$: $\mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_T$
H_1, H_2, H_3, H_4	Cryptographic hash functions
g	A generator of $\mathbb{G}$
$mpk = (d_1, d_2, d_3)$	The master private key
$param$	The public parameter
$usk_i = (usk_{i,1}, usk_{i,2}, usk_{i,3})$	Private key of user ID_i
ssk_f, ssk_b	Private keys of S_f and S_b
$c = (c_1, c_2, c_3)$	Ciphertext of message m
$\hat{t}_{i,f} = (t_0, t_1)$, $\hat{t}_{i,b} = (t_0, t_2)$	Ciphertext authentication tokens of user ID_i for S_f and S_b
$\tau_{i,f'}\tau_{i,b}$	Plaintext authentication tokens of user ID_i for S_f and S_b
$\Gamma = (c_1, c'_1, \varpi)$	Internal test result of equality test

4.5. Data Decryption. For ciphertext $c = (c_1, c_2, c_3)$, the user ID_i decrypts it with the private key usk_i as follows. The user ID_i computes the following:

$$T = \frac{c_2}{H_3\left(\hat{e}\left(usk_{i,1}, c_1\right)\right) \cdot H_3\left(\hat{e}\left(usk_{i,2}, c_1\right)\right)}, \tag{13}$$

$$m\prime \parallel \alpha' \leftarrow c_3 \oplus H_4\left(\hat{e}\left(usk_{i,3}, c_1\right) \parallel T\right). \tag{14}$$

Next, the user ID_i checks whether both of the following equalities hold:

$$c_1 = g^{a\prime}, \tag{15}$$

$$T = H_2(m\prime). \tag{16}$$

If so, $m\prime$ is outputted, otherwise $\perp$ is outputted.

4.6. Authorization. The user ID_i randomly picks an element $\gamma \in \mathbb{Z}_q^*$ and computes the following:

$$t_0 = g^{\gamma} t_1 = usk_{i,1} \cdot H_3\left(\hat{e}\left(H_1\left(S_f\right)\right), V_1\right)^{\gamma}, \tag{17}$$

$$t_1 = usk_{i,1} \cdot H_3\left(\hat{e}\left(H_1\left(S_f\right)\right), V_1\right)^{\gamma}. \tag{18}$$

Then, the encrypted authorization tokens $\hat{t}_{i,f} = (t_0, t_1)$ and $\hat{t}_{i,b} = (t_0, t_2)$ are sent to the front server S_f and to the back server S_b, respectively.

4.7. Token Decryption. Given the encrypted authorization token $\hat{t}_{i,f}$, the front server S_f computes the following equation:

$$\tau_{i,f} = \frac{t_1}{H_3\left(\hat{e}\left(ssk_f, t_0\right)\right)}. \tag{19}$$

The back server S_b can decrypt the token $usk_{i,b}$ in the similar way as follows:

$$\tau_{i,b} = \frac{t_2}{H_3\left(\hat{e}\left(ssk_b, t_0\right)\right)}, \tag{20}$$

then, these two servers are able to validate the recovered tokens as in (5) and (6), respectively.

4.8. Front Server Ciphertext Test. For the ciphertexts $c = (c_1, c_2, c_3)$ and $c\prime = (c_1', c_2', c_3')$ of two users ID_i and ID_j, respectively, the front server S_f generates the internal test result $\Gamma = (c_1, c_1', \varpi)$ with their respective tokens $\tau_{i,f}$ and $\tau_{j,f}$. The front server S_f computes the following equation:

$$\theta = \frac{c_2}{H_3\left(\hat{e}\left(\tau_{i,f}, c_1\right)\right)}, \tag{21}$$

$$\theta\prime = \frac{c_2'}{H_3\left(\hat{e}\left(\tau_{j,f}, c_1'\right)\right)}, \tag{22}$$

then, it computes

$$\varpi = \frac{\theta}{\theta\prime}, \tag{23}$$

the internal test result $\Gamma = (c_1, c_1', \varpi)$ is sent to the back server S_b.

4.9. Back Server Ciphertext Test. For the internal test result $\Gamma = (c_1, c_1', \varpi)$ on the ciphertexts of users ID_i and ID_j, the back server S_b checks the following equality with the received tokens $\tau_{i,b}$ and $\tau_{j,b}$:

$$\varpi = \frac{H_3\left(\hat{e}\left(\tau_{i,b}, c_1\right)\right)}{H_3\left(\hat{e}\left(\tau_{j,b}, c_1'\right)\right)}, \tag{24}$$

if it holds, then "1" is outputted, which means the two ciphertexts c and c' of users ID_i and ID_j encrypt the same message; otherwise "0" is outputted, which means different messages are encrypted in two ciphertexts.

Theorem 1. *The proposed DS-IBE-AET construction in bilinear groups is sound.*

Proof. 1 First, for the first element $usk_{i,1}$ in the private key usk_i of user ID_i, the equality in (1) holds as follows:

$$\hat{e}\left(usk_{i,1}, g\right) = \hat{e}\left(H_1\left(ID_i\right)^{d_1}, g\right) = \hat{e}\left(H_1\left(ID_i\right), V_1\right). \tag{25}$$

The equalities in (6) and (7) for the other two elements $usk_{i,2}$ and $usk_{i,3}$ can be verified in the similar way.

Second, for the private key ssk_f for the front server S_f, the equality in (9) holds as follows:

$$\hat{e}\left(ssk_f, g\right) = \hat{e}\left(H_1\left(ID_i\right)^{d_1}, g\right) = \hat{e}\left(H_1\left(ID_i\right), V_1\right). \tag{26}$$

The equality in (11) for the back server S_b can be verified in the similar way.

Third, for the correctness of decryption on user ciphertexts, since

$$\begin{aligned} T &= \frac{c_2}{H_3\left(\hat{e}\left(usk_{i,1}, c_1\right)\right) \cdot H_3\left(\hat{e}\left(usk_{i,2}, c_1\right)\right)} \\ &= \frac{H_2(m) \cdot H_3\left(\hat{e}\left(H_1\left(ID_i\right), V_1\right)^{\alpha} \cdot H_3\left(\hat{e}\left(H_1\left(ID_i\right), V_2\right)^{\alpha}\right)\right.}{H_3\left(\hat{e}\left(H_1\left(ID_i\right)^{d_1}, g^{\alpha}\right)\right) \cdot H_3\left(\hat{e}\left(H_1\left(ID_i\right)^{d_2}, g^{\alpha}\right)\right)} \\ &= \frac{H_2(m) \cdot H_3\left(\hat{e}\left(H_1\left(ID_i\right), V_1\right)^{\alpha} \cdot \hat{e}\left(H_1\left(ID_i\right), V_2\right)^{\alpha}\right.}{H_3\left(\hat{e}\left(H_1\left(ID_i\right)^{d_1}, g^{\alpha}\right)\right) \cdot H_3\left(\hat{e}\left(H_1\left(ID_i\right)^{d_2}, g^{\alpha}\right)\right)} \\ &= H_2(m). \end{aligned} \tag{27}$$

we have

$$\begin{aligned} m'\|\alpha' &= c_3 \oplus H_4\left(\hat{e}\left(usk_{i,3}, c_1\right)\|T\right) \\ &= (m\|\alpha) \oplus H_4\left(\hat{e}\left(H_1\left(ID_i\right), V_3\right)^{\alpha}\right. \\ &\left.\|H_2(m)\right) \oplus H_4\left(\hat{e}\left(H_1\left(ID_i\right)^{d_3}, g^{\alpha}\right)\|H_2(m)\right)b \\ &= (m\|\alpha) \oplus H_4\left(\hat{e}\left(H_1\left(ID_i\right), g^{d_3}\right)^{\alpha}\|H_2(m)\right) \\ &\oplus H_4\left(\hat{e}\left(H_1\left(ID_i\right)^{d_3}, g^{\alpha}\right)\|H_2(m)\right) \\ &= (m\|\alpha), \end{aligned} \tag{28}$$

thus, the equalities (15) and (16) hold, which means the message m can be successfully decrypted.

Four, for the authorization token decryption, it can be seen that

$$\begin{aligned} \tau_{i,f} &= \frac{t_1}{H_3\left(\hat{e}\left(ssk_f, t_0\right)\right)} \\ &= \frac{usk_{i,1} \cdot H_3\left(\hat{e}\left(H_1\left(S_f, V_1\right)^{\gamma}\right)\right.}{H_3\left(\hat{e}\left(H_1\left(S_f\right)^{d_1}\right), g^{\gamma}\right)} \\ &= \frac{usk_{i,1} \cdot H_3\left(\hat{e}\left(H_1\left(S_f\right), g^{d_1}\right)^{\gamma}\right)}{H_3\hat{e}\left(H_1\left(S_f\right)^{d_1}, g^{\gamma}\right)} \\ &= usk_{i,1}. \end{aligned} \tag{29}$$

$$\begin{aligned} \tau_{i,b} &= \frac{t_2}{H_3\left(\hat{e}\left(ssk_b, t_0\right)\right)} \\ &= \frac{usk_{i,b} \cdot H_3\left(\hat{e}\left(H_1\left(S_b\right), V_1\right)^{\gamma}\right)}{H_3\hat{e}\left(H_1\left(S_b\right)^{d_1}, g^{\gamma}\right)} \\ &= \frac{usk_{i,b} \cdot H_3\left(\hat{e}\left(H_1\left(S_b\right), g^{d_1}\right)^{\gamma}\right)}{H_3\hat{e}\left(H_1\left(S_b\right)^{d_1}, g^{\gamma}\right)} \\ &= usk_{i,2}. \end{aligned}$$

Thus, the tokens for the front server S_f and back server S_b can be correctly decrypted as in (19) and (20).

Five, for an authorized equality test on ciphertexts, since

$$\begin{aligned}\theta &= \frac{c_3}{H_3\left(\hat{e}\left(\tau_{i,1}, c_1\right)\right)}\\ &= \frac{H_2(m)\cdot H_3\left(\hat{e}\left(H_1(ID_i), V_1\right)^{\alpha}\right)\cdot H_3\left(\hat{e}\left(H_1(ID_i), V_2\right)^{\alpha}\right)}{H_3\hat{e}\left(\left(H_1(ID_i)^{d_1}, g^{\alpha}\right)\right)}\\ &= \frac{H_2(m)\cdot H_3\left(\hat{e}\left(H_1(ID_i), g^{d_1}\right)^{\alpha}\right)\cdot H_3\left(\hat{e}\left(H_1(ID_i), V_2\right)^{\alpha}\right)}{H_3\hat{e}\left(\left(H_1(ID_i)^{d_1}, g^{\alpha}\right)\right)}\\ &= H_2(m)\cdot H_3\left(\hat{e}\left(H_1(ID_i), V_2\right)^{\alpha}\right),\\ \theta\prime &= \frac{c_2'}{H_3\left(\hat{e}\left(\tau_{j,1}, c_1'\right)\right)},\\ &= \frac{H_2(m\prime)\cdot H_3\left(\hat{e}\left(H_1\left(ID_j\right), V_1\right)^{\alpha\prime}\right)\cdot H_3\left(\hat{e}\left(H_1\left(ID_j\right), V_2\right)^{\alpha\prime}\right)}{H_3\hat{e}\left(\left(H_1\left(ID_j\right)^{d_1}, g^{\alpha\prime}\right)\right)}\\ &= \frac{H_2(m\prime)\cdot H_3\left(\hat{e}\left(H_1\left(ID_j\right), g^{d_1}\right)^{\alpha\prime}\right)\cdot H_3\left(\hat{e}\left(H_1\left(ID_j\right), V_2\right)^{\alpha\prime}\right)}{H_3\left(\hat{e}\left(H_1\left(ID_j\right)^{d_1}, g^{\alpha\prime}\right)\right) = H_2(m\prime)\cdot H_3\left(\hat{e}\left(H_1\left(ID_j\right), V_2\right)^{\alpha\prime}\right).}\end{aligned} \tag{30}$$

we have

$$\begin{aligned}\omega &= \frac{\theta}{\theta'}\\ &= \frac{H_2(m)\cdot H_3\left(\hat{e}\left(H_1(ID_i), V_2\right)^{\alpha}\right)}{H_2(m')\cdot H_3\left(\hat{e}\left(H_1\left(ID_j\right), V_2\right)^{\alpha'}\right)}\\ &= \frac{H_2(m)\cdot H_3\left(\hat{e}\left(H_1(ID_i), g^{d_2}\right)^{\alpha}\right)}{H_2(m')\cdot H_3\left(\hat{e}\left(H_1\left(ID_j\right), g^{d_2}\right)^{\alpha\prime}\right)}\\ &= \frac{H_2(m)\cdot H_3\left(\hat{e}\left(H_1(ID_i)^{d_2}, g^{\alpha}\right)\right)}{H_2(m')\cdot H_3\left(\hat{e}\left(H_1\left(ID_j\right)^{d_2}, g^{\alpha'}\right)\right)}\\ &= \frac{H_2(m)\cdot H_3\left(\hat{e}\left(usk_{i,2}, c_1\right)\right)}{H_2(m')\cdot H_3\left(\hat{e}\left(usk_{j,2}, c_1'\right)\right)}\\ &\overset{m=m'}{\Leftrightarrow} \frac{H_3\left(\hat{e}\left(usk_{i,2}, c_1\right)\right)}{H_3\left(\hat{e}\left(usk_{j,2}, c_1'\right)\right)}.\end{aligned} \tag{31}$$

It can be seen that if the messages m and m' are the same, then the equality in (24) holds.

Therefore, the proposed DS-IBE-AET construction in bilinear groups is sound. □

5. Analysis and Comparison

5.1. Security Analysis

Theorem 2. *The proposed DS-IBE-AET construction in the dual-server model can guarantee the unforgeability of the private keys of users.*

Proof. 2 As shown in Sections 4.2, the private keys of users are generated by KGC using their private keys (d_1, d_2, d_3). Particularly, each element of the private key is a signature on the user's identity with the short signature scheme of Boneh et al [29]. Therefore, according to the security result that the BLS signature scheme is secure against existential forgery under adaptive chosen-message attacks in the random oracle model assuming the CDH assumption holds [29], Theorem 3.2, the proposed DS-IBE-AET scheme can protect the unforgeability of private keys of users. □

Theorem 3. *The proposed DS-IBE-AET construction in the dual-server model can guarantee the unforgeability of the private keys of both servers.*

Proof. 3 Similar to the analysis for Theorem 2, the private keys of both servers are generated by KGC using their private keys d_1 by employing the short signature scheme of Boneh et al [29]. Thus, according to the security result of [29], Theorem 3.2, the proposed DS-IBE-AET scheme can protect the unforgeability of the private keys of two servers. □

Theorem 4. *The proposed DS-IBE-AET construction in the dual-server model can guarantee the privacy of outsourced data against the front server.*

Proof. 4 As shown in Section 4.4, the ciphertext in the proposed DS-IBE-AET scheme has a similar form as Lee et al.'s PKE-AET scheme [30]. Note that their scheme is designed in generic cyclic groups, and our DS-IBE-AET scheme is developed in bilinear groups in an identity-based setting. Moreover, the pair (c_1, c_2) can be seen as an extension of the ciphertext in Boneh and Franklin's basic IBE scheme [31], Section 4. For the second component c_2 of our ciphertext, two public parameters V_1 and V_2 are used, which would be used to enable both the front server and back server to collaboratively perform equality tests on ciphertexts; while only one public key g^{α} is used in producing c_2 in Lee et al.'s PKE-AET scheme [30], since their scheme is considered in the single server model. Therefore, before the front server gets authorized, the proof for the privacy of outsourced data follows [30], Theorem 4.1 and [31], Theorem 4.1, that is, the proposed DS-IBE-AET scheme offers indistinguishability against chosen ciphertext and chosen identity attacks (IND-ID-CCA security) for the front server under the CDH and CBDH assumptions. When the front server is authorized, it would get the authorization token $\tau_{i,1}$ for performing an equality test on ciphertexts. Thus, the

proposed DS-IBE-AET scheme offers one-wayness security against chosen ciphertext attacks and chosen identity attacks [31, 32] under the CDH and CBDH assumptions. □

Theorem 5. *The proposed DS-IBE-AET construction in the dual-server model can guarantee the privacy of outsourced data against the back server.*

Proof. 5 In the proposed DS-IBE-AET scheme, the outsourced ciphertexts are only stored at the front server. When collaboratively performing equality tests on ciphertexts, only the intermediate result $\Gamma = (c_1, c_1, \varpi)$ is given to the back server by the front server. Note that ϖ is computed from θ and θ'. As shown in equations (12) and (13), θ and θ' have a similar form of c_2 in ciphertext of Lee et al.'s scheme [30], but in an identity-based setting [31]. That is, the pairs (c_1, γ) and $(c_1', \gamma\prime)$ have a similar form of (c_1, c_2) in Lee et al.'s scheme [30] and Boneh and Franklin's basic IBE scheme [31], Section 4. Thus, the proof is similar to that in [30], Theorem 4.1, and [31], Theorem 4.1, that is, the proposed DS-IBE-AET scheme is IND-ID-CCA secure against the back server under the CDH and CBDH assumptions. □

Theorem 6. *The proposed DS-IBE-AET construction in the dual-server model can guarantee the privacy of an authentication token.*

Proof. 6 The ciphertext authentication token in the proposed DS-IBE-AET construction is generated in a similar way as the ciphertexts in Boneh and Franklin's basic IBE scheme [31], Section 4. The difference is that t_0 is used to construct two encrypted authorization tokens $\widehat{t}_{i,f} = (t_0, t_1)$ and $\widehat{t}_{i,b} = (t_0, t_2)$ for two servers, respectively. Thus, the proof is similar to that in [31], Theorem 4.1, that is, the authentication token in the proposed DS-IBE-AET construction enjoys indistinguishability against chosen plaintext and chosen identity attacks under the CBDH assumption. □

5.2. Performance Analysis. In this section, we analyze the efficiency of our DS-IBE-AET construction in each procedure and compare with Zhao et al.'s construction [9] in terms of resource-intensive operations such as exponentiation, bilinear pairing, and the map-to-point hash function. As shown in Table 2, let ℓ_E denote the evaluation cost of an exponentiation in group $\mathbb{G}$, ℓ_P represent the evaluation cost of a bilinear pairing $\widehat{e}(\cdot,\cdot)$ and ℓ_H signify a map-to-point hash function, respectively.

Since our DS-IBE-AET construction is developed in an identity-based setting, the private keys of users and servers are generated by the trusted KGC, where the computational cost of generating a private key for a user is 3 times the cost for a server. Users and servers only need to verify the correctness of the issued private keys, respectively. Note that these private keys should be delivered via a secure channel, thus the verification process can be omitted by the respective users and servers. While in Zhao et al.'s construction [9], the private keys are produced by respective user and server, which take 3 and 2 exponentiation operations on the bilinear group $\mathbb{G}$, respectively.

To facilitate the analysis of the data encryption procedure, the exponentiation operation in group $\mathbb{G}_T$ in both schemes is converted to first computing the exponentiation operation in group $\mathbb{G}$ and then performing the bilinear pairing operation $\widehat{e}(\cdot,\cdot)$, which can enable intermediate calculated parameters to be reused and reduce computing costs. Hence, the Encrypt procedure of our DS-IBE-AET scheme takes two less exponentiation operations than that in Zhao et al.'s construction [9] when encrypting a message. Since our DS-IBE-AET scheme is designed in an identity-based setting, it takes one more bilinear pairing and map-to-point hash function evaluation than [9]. For decrypting a ciphertext, although our DS-IBE-AET scheme takes one more bilinear pairing operation than Zhao et al.'s construction [9], it only requires one exponentiation operation, whereas the latter needs to carry out 4 exponentiation operations.

In the authentication phase, our DS-IBE-AET scheme allows the user to generate different tokens for two servers. Note that these tokens have the same form and share one element t_0. Thus, the computing cost for generating t_0 can be shared by two tokens. Also, the exponentiation operation of V_1^{γ} can be reused in producing both t_1 and t_2.

As shown in (17) and (18), the generation of t_1 and t_2, respectively, requires two time-consuming map-to-point hash operations. While in Zhao et al.'s construction [9], two servers shall be authenticated with the same token; that is, the servers would recover the same token with the only difference that their respective private keys would be used during decryption. Moreover, since the authentication token is in fact an element of the user's private key, it can be validated according to the relationship with the corresponding public key.

After being authorized, the front server and back server are able to cooperatively carry out equality tests on outsourced ciphertexts. On both server sides, our DS-IBE-AET scheme is much more efficient than Zhao et al.'s construction [9], where no exponentiation operations are required in our DS-IBE-AET scheme. Specifically, to perform an equality test on one pair of ciphertexts, both servers in Zhao et al.'s construction [9] should take 4 more exponentiation operations than those in our DS-IBE-AET scheme, which is due to the fact that the private keys of these servers should be used in their respective procedures. It can be seen that the computing costs for the equality test in both schemes are linear with the number of compared ciphertexts.

Moreover, we evaluate the performance of our DS-IBE-AET scheme and compare it with Zhao et al.'s construction [9], where the experimental execution times of cryptographic operations in [33] are used. The experiments of [33] were carried out on a platform with a Windows 7 operating system, an Intel I7-4700@3.40 GHz CPU, and 4 GB of memory, where the MIRACL Cryptographic SDK [34] was run with $\log q = 512$. The exact execution times of three resource-intensive cryptographic operations are shown in Table 3.

Table 2: Comparison of computing costs.

Procedure		Our DS-IBE-AET construction	Zhao et al.s construction [9]
UKeyExt	KGC	$3\ \ell_E + 3\ \ell_H$	—
	User	$6\ \ell_P + 3\ \ell_H$	$3\ \ell_E$
SKeyExt	KGC	$1\ \ell_E + 1\ \ell_H$	—
	Server	$2\ \ell_P + 1\ \ell_H$	$2\ \ell_E$
Encrypt		$2\ \ell_E + 3\ \ell_P + 4\ \ell_H$	$4\ \ell_E + 2\ \ell_P + 3\ \ell_H$
Decrypt		$1\ \ell_E + 3\ \ell_P + 3\ \ell_H$	$4\ \ell_E + 2\ \ell_P + 3\ \ell_H$
Authen		$2\ \ell_E + 2\ \ell_P + 2\ \ell_H$	$2\ \ell_E + 1\ \ell_P$
DecAuth	Decryption	$1\ \ell_P + 1\ \ell_H$	$1\ \ell_E + 1\ \ell_P$
	Verification	$2\ \ell_P + 1\ \ell_H$	$1\ \ell_E$
EqTest_f		$2\ \ell_P + 2\ \ell_H$	$4\ \ell_E + 2\ \ell_P + 2\ \ell_H$
EqTest_b		$2\ \ell_P + 2\ \ell_H$	$4\ \ell_E + 2\ \ell_P + 2\ \ell_H$

Table 3: Execution time of cryptographic operations.

Cryptographic operation	Computing time (ms)
Bilinear pairing	4.211
Exponentiation in group $\mathbb{G}$	1.709
Map-to-point hash function	4.406

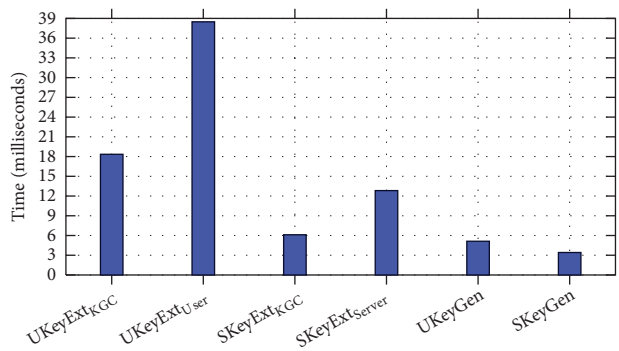

Figure 3: Performance of private key extraction procedures in both schemes.

The performance of private key extraction procedures of our DS-IBE-AET scheme and the key generation procedure of Zhao et al.'s construction [9] are depicted in Figure 3. It can be seen that in our DS-IBE-AET scheme, the verification procedures at both the user and server sides take more time than KGC. Note that the private keys for users and servers only need to be extracted once; the computational costs for them are affordable. In Zhao et al.'s construction [9], users and servers can generate private keys for themselves in less than 6 milliseconds, and there is no verification procedure.

The performance of other procedures is depicted in Figure 4, where the case for each procedure to be executed once is considered for both schemes. To encrypt a message, the proposed DS-IBE-AET scheme will take 5 milliseconds more than Zhao et al.'s construction [9], while our data decryption procedure is more efficient. Since the encryption of the authentication token in our scheme is designed in an identity-based setting, it would take more time to encrypt, decrypt, and validate the token than that in Zhao et al.'s construction [9]. For collaboratively performing equality test on two ciphertexts, both the front and back servers roughly take 7 milliseconds less than Zhao et al.'s construction [9].

The comparison on communication costs between our DS-IBE-AET construction and Zhao et al.'s scheme [9] is shown in Table 4, in terms of the sizes of the user private key, server private key, ciphertext, authorization token, and internal equality test results. Since our DS-IBE-AET construction is designed in an identity-based setting, it requires KGC to issue the private keys for users and servers. As shown in Table 4, each user's private key in our scheme contains three elements in group $\mathbb{G}$ and each server's private key contains only one element in group $\mathbb{G}$. While for the scheme from Zhao et al. [9], it was developed in a public key setting and the private keys can be respectively generated by each user and server. However, it is well-known that the corresponding public keys for the users and servers should be maintained through the public key infrastructure.

For the ciphertext corresponding to a message, both our DS-IBE-AET construction and Zhao et al.'s scheme [9] are composed of three elements of the same size and enjoy the same communication costs. For the authorization phase, our

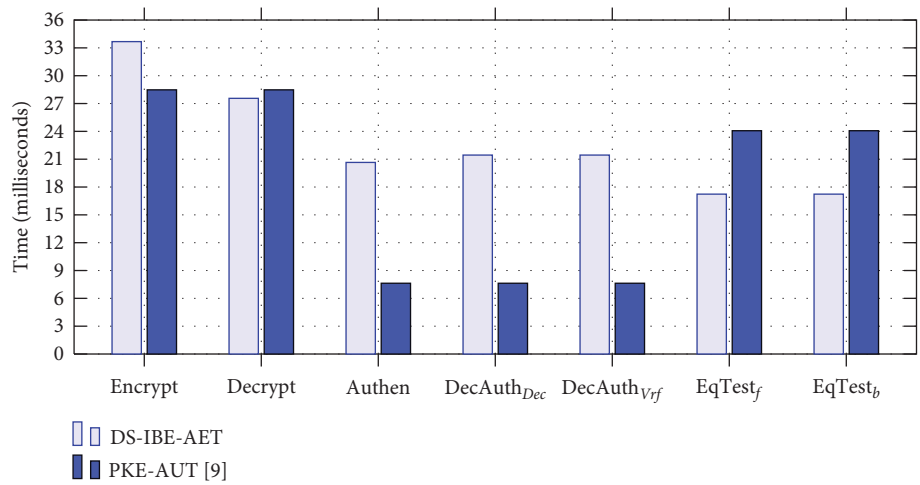

Figure 4: Performance of each procedure in our DS-IBE-AET scheme.

Table 4: Comparison of communication costs.

	Our DS-IBE-AET scheme	Zhao et al.'s scheme [9]
User private key	$3\ \xi_{\mathbb{G}}$	—
Server private key	$\xi_{\mathbb{G}}$	—
Ciphertext	$2\ \xi_{\mathbb{G}} + \xi_m + \log q$	$2\ \xi_{\mathbb{G}} + \xi_m + \log q$
Authorization	$4\ \xi_{\mathbb{G}}$	$2\ \xi_{\mathbb{G}} + 2\ \log\ q$
Internal test result	$3\ \xi_{\mathbb{G}}$	$3\ \xi_{\mathbb{G}}$

DS-IBE-AET construction sends different authorization tokens of the same size of $2\xi_{\mathbb{G}}$ to each server. Whereas in Zhao et al.'s scheme [9], the two servers would receive an identical authorization token containing one element in group $\mathbb{G}$ and one element in $\mathbb{Z}_q$. In the equality test phase, both schemes require the front server to deliver the internal test result to the back server, which comprises three elements in group $\mathbb{G}$ for comparing a pair of ciphertexts.

6. Conclusion

This paper proposed an identity-based encryption with authorized equality test on ciphertexts in a dual-sever setting (DS-IBE-AET), which addressed the complicated certificate management problem in existing proposals supporting equality test on ciphertexts in a public key setting. Particularly, the proposed DS-IBE-AET construction can resist keyword guessing attacks on outsourced ciphertexts that are only stored on the front server side. Only after obtaining the authentication from users would the front server and back server be able to collaboratively perform equality tests on the ciphertexts of these users, where the front server generates an internal test result for further confirmation by the back server. Security analysis demonstrated that the presented DS-IBE-AET scheme can protect the privacy of outsourced ciphertexts and authentication tokens, and performance analysis showed the practicality of our DS-IBE-AET scheme.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This article was supported by the Guangxi Natural Science Foundation under grants 2019GXNSFFA245015 and 2019GXNSFGA245004, the National Natural Science Foundation of China under projects 62162017 and 62172119, and the Peng Cheng Laboratory Project of Guangdong Province PCL2021A09, PCL2022A02, and PCL2021A03.

References

[1] X. Fu, X. Nie, and F. Li, "Large universe attribute based access control with efficient decryption in cloud storage system," *Journal of Systems and Software*, vol. 135, pp. 157–164, 2018.

[2] X. Fu, Y. Ding, and H. Li, "A survey of lattice based expressive attribute based encryption," *Computer Science Review*, vol. 43, Article ID 100438, 2022.

[3] D. Boneh, G. Di Crescenzo, R. Persiano, and G. Persiano, "Public key encryption with keyword search," in *Advances in Cryptology - EUROCRYPT 2004*, C. Cachin and J. L. Camenisch, Eds., Springer, Heidelberg, Germany, pp. 506–522, 2004.

[4] G. Yang, C. H. Tan, and D. S. Wong, "Probabilistic public key encryption with equality test," *Topics in Cryptology - CT-RSA 2010*, Springer-Verlag, in *Proceedings of the 2010 International Conference on Topics in Cryptology, CT-RSA'10*, pp. 119–131, June 2010.

[5] L. Wu, Y. Zhang, K. Choo, and D. He, "Efficient and secure identity-based encryption scheme with equality test in cloud

computing," *Future Generation Computer Systems*, vol. 73, pp. 22–31, 2017.

[6] H. Li, Q. Huang, S. Ma, and W. Susilo, "Authorized equality test on identity-based ciphertexts for secret data sharing via cloud storage," *IEEE Access*, vol. 7, pp. 25409–25421, Article ID 25409, 2019.

[7] Y. Wang and H H. Pang, "Probabilistic public key encryption for controlled equijoin in relational databases," *The Computer Journal*, vol. 60, no. 4, pp. 600–612, 2017.

[8] M. Ramadan, Y. Liao, F. Li, and H. Abdalla, "IBEET-RSA: identity-based encryption with equality test over RSA for wireless body area networks," *Mobile Networks and Applications*, vol. 25, no. 1, pp. 223–233, 2020.

[9] M. Zhao, Y. Ding, S. Tang, and H. Wang, "Public key encryption with authorized equality test on outsourced ciphertexts for cloud-assisted iot in dual server model," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–10, Article ID 4462134, 2022.

[10] L. Wu, Y. Zhang, and D. He, "Dual server identity-based encryption with equality test for cloud computing," *Journal of Computer Research and Development*, vol. 54, no. 10, pp. 2232–2243, 2017.

[11] P. Hweehwa and X. Ding, "Privacy-preserving ad-hoc equijoin on outsourced data," *ACM Transactions on Database Systems*, vol. 39, no. 3, pp. 1–40, October 2014.

[12] H. Cui, R. H. Deng, and G. Wu, "Attribute-based storage supporting secure deduplication of encrypted data in cloud," *IEEE Transactions on Big Data*, vol. 5, no. 3, pp. 330–342, 2019.

[13] Y. Zheng, M. Wang, Y. Li, and V. Athanasios, "Vasilakos. "Encrypted data management with deduplication in cloud computing"," *IEEE Cloud Computing*, vol. 3, no. 2, pp. 28–35, 2016.

[14] Y. Wang, H H. Pang, and H. Robert, "Cca secure encryption supporting authorized equality test on ciphertexts in standard model and its applications," *Information Sciences*, vol. 414, pp. 289–305, 2017.

[15] Y. Wang and Y. Ding, "Privacy-preserving cloud-based road condition monitoring with source authentication in vanets," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 7, pp. 1779–1790, 2019.

[16] Q. Tang, "Public key encryption supporting plaintext equality test and user-specified authorization," *Security and Communication Networks*, vol. 5, no. 12, pp. 1351–1362, 2012.

[17] Q. Tang, "Towards public key encryption scheme supporting equality test with fine-grained authorization," in *Proceedings of the 16th Australasian Conference on Information Security and Privacy ACISP'11*, pp. 389–406, Springer-Verlag, Heidelberg, Germany, June 2011.

[18] Y. Wang, H H. Pang, H. D. Robert, D. Yong, W. Qianhong, and Q. Bo, "Securing messaging services through efficient signcryption with designated equality test," *Information Sciences*, vol. 490, pp. 146–165, 2019.

[19] H. T. Lee, San Ling, J. H. Seo, H. Wang, and Y Y. Taek, "Public key encryption with equality test in the standard model," *Information Sciences*, vol. 516, pp. 89–108, 2020.

[20] C. Li, Q. Shen, and Z. Xie, "Large universe CCA2 CP-abe with equality and validity test in the standard model," *The Computer Journal*, vol. 64, no. 4, pp. 509–533, 07 2020.

[21] Y. Wang, H H. Pang, H. D. Robert, D. Yong, W. Qianhong, and Q. Bo, "Secure server-aided data sharing clique with attestation," *Information Sciences*, vol. 522, pp. 80–98, 2020.

[22] S. Ma, "Identity-based encryption with outsourced equality test in cloud computing," *Information Sciences*, vol. 328, pp. 389–402, 2016.

[23] T. Wu, S. Ma, Yi Mu, and S. Zeng, "Id-based encryption with equality test against insider attack," in *Australasian Conference on Information Security and Privacy*, pp. 168–183, Springer International Publishing, New York, NY, USA, 2017.

[24] H. T. Lee, H. Wang, and K. Zhang, "Security analysis and modification of id-based encryption with equality test from ACISP 2017,"vol. 10946, pp. 780–786, in *Proceedings of the Information Security and Privacy - 23rd Australasian Conference, ACISP 2018*, vol. 10946, Springer, Wollongong, NSW, Australia, July 2018.

[25] S. Alornyo, A. Edward Mensah, and O A. Abraham, "Identity-based public key cryptographic primitive with delegated equality test against insider attack in cloud computing," *International Journal on Network Security*, vol. 22, no. 5, pp. 743–751, 2020.

[26] Y. Ling, S. Ma, and Q. Huang, "Efficient group ID-based encryption with equality test against insider attack," *The Computer Journal*, vol. 64, no. 4, 674 pages, 661.

[27] R. Chen, Yi Mu, G. Yang, F. Guo, and X. Wang, "Dual-server public-key encryption with keyword search for secure cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 4, pp. 789–798, 2016.

[28] Q. Tang, "Public key encryption schemes supporting equality test with authorisation of different granu- larity," *International Journal of Applied Cryptography*, vol. 2, no. 4, pp. 304–321, 2012.

[29] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," *Journal of Cryptology*, vol. 17, pp. 297–319, 2004.

[30] H. T. Lee, San Ling, J. H. Seo, and H. Wang, "CCA2 attack and modification of Huang et al.'s public key encryption with authorized equality test," *The Computer Journal*, vol. 59, no. 11, pp. 1689–1694, 2016.

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

[32] S. Ma, M. Zhang, Q. Huang, and Bo Yang, "Public key encryption with delegated equality test in a multi-user setting," *The Computer Journal*, vol. 58, no. 4, pp. 986–1002, 04 2014.

[33] D. He, S. Zeadally, B. Xu, and X. Huang, "An efficient identity-based conditional privacy- preserving authentication scheme for vehicular ad hoc networks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2681–2691, 2015.

[34] S. D. K. Miracl Cryptographic, "Multiprecision integer and rational arithmetic cryptographic library," 2019, https://github.com/miracl/miracl.