sensors

MDPI

Article

Identity-Based Encryption with Filtered Equality Test for Smart City Applications

Yang Ming * and Erxiu Wang

School of Information Engineering, Chang'an University, Xi'an 710064, China
* Correspondence: yangming@chd.edu.cn; Tel.: +86-136-0911-6306

Received: 22 May 2019; Accepted: 9 July 2019; Published: 10 July 2019

Abstract: With the growth of the urban population, the rapid development of smart cities has become the focus of urban regional development. Smart medical care is an indispensable part of smart city construction, which promotes the development of the medical industry. However, the security of data and timely service are the current problems faced by intelligent medical systems. Based on the public key encryption with filtered equality test and identity-based cryptography, an identity-based encryption with the filtered equality test (IBE-FET) is proposed for smart healthcare, in which a data receiver can use the private key and the message set to generate a warrant and send it to the cloud server. A cloud server can verify the equality between ciphertexts without decryption and check whether the encrypted message belongs to the same message set. Furthermore, the security analysis shows that the proposed scheme satisfies one-way security against the chosen identity and ciphertext attack in the random oracle model under the computational bilinear Diffie-Hellman assumption. The performance comparison shows that the scheme is feasible and practical in real life.

Keywords: smart healthcare; identity-based encryption; filtered equality test; random oracle model

1. Introduction

The concept of the smart city (SC) [1] emerges in the context in which the current global power supply and consumption trends are socially, environmentally and economically unsustainable. It refers to an urban transformation which, with the use of the latest information and communications technologies (ICT), improves cities' efficiency. Currently, more and more people live in cities and every person uses more than five devices to access the Internet. Thus, the various embedded devices are integrated with urban infrastructure to optimize daily life of citizens.

Recently, with the rapid development of the Internet of Things (IoT) [2] and ICT, the applications of the smart city [3] are on the rise, which can enhance the life quality of citizens. Representative smart city applications are given in Figure 1, which benefit the city and people in a variety of aspects: economy, education, healthcare, and living. Meanwhile, the smart city has a new, complete level of effectiveness, sustainability and efficiency.

The main goal of the smart city is to greatly improve quality of life. Nevertheless, the security and privacy problems are of great importance to the users in the smart city [4–6]. Progress in the IoT and cloud computing technology is driving the development of smart systems to support and improve healthcare system. However, the current healthcare system is faced with a series of challenges in providing low cost health care services. Besides, it is difficult for patients in some areas to obtain a timely healthcare services due to poor medical conditions. As a result, smart healthcare [7,8] has emerged recently as the key component of a new generation healthcare network. The so-called smart healthcare is to improve the efficiency of biomedical systems and healthcare infrastructures through various entities and technologies, including smart sensors, wearable devices, ICT and more [9].

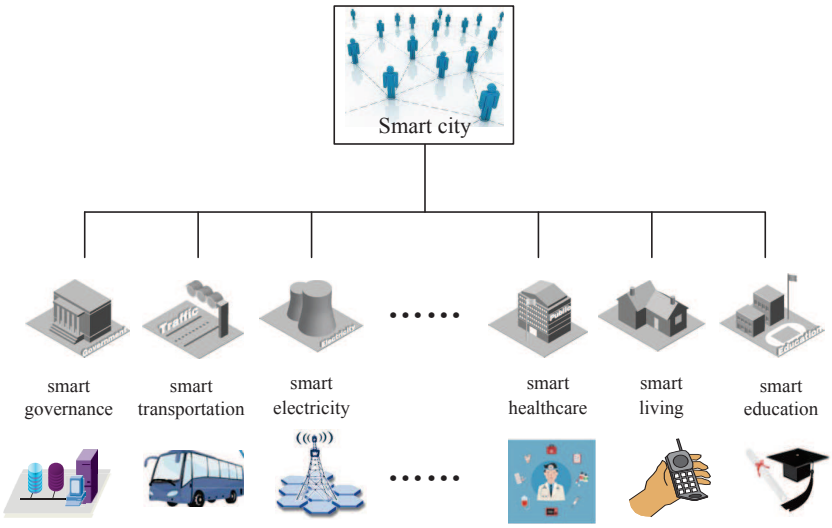

Figure 1. Representative smart city applications.

In the smart healthcare system, patients are paying more and more attention to the security of private information. Zhang et al. [10–13] has done in-depth research and proposed privacy-preserving access control systems by adopting attribute-based encryption techniques to improve the security of smart healthcare. However, the techniques are complex and unfeasible in practice. To save storage space and protect the user's privacy, the sensitive information must be stored in the untrusted healthcare cloud servers in an encrypted form. However, given some ciphertexts, no one can distinguish the relationships among the ciphertexts without decryption. Searchable encryption (SE) [14–16] is a practical and promising solution to this problem. To provide the capability for searching in the ciphertexts, the public key encryption with keyword search (PKE-KS) schemes [17–22] were proposed, which is one practical implementation of SE. However, the PKE-KS schemes have one weakness that the ciphertexts are generated by the same public keys and therefore it is not applicable to some scenarios. To solve this problem, the public key encryption with equality test (PKE-ET) schemes [23–31] were put forward, which allowed equality tests made on the ciphertexts by different public keys as well as the same public keys. To alleviate the storage cost of certificates, identity-based encryption with equality test (IBE-ET) schemes [32,33] were proposed. Along with research, to make fine-grained authorization more flexible and inspired by the idea of attribute-based encryption, the attribute-based encryption with equality test (ABE-ET) schemes [34–37] were presented.

To provide more flexible equality testing to satisfy different requirements, Huang et al. presented the public key encryption with filtered equality test (PKE-FET) schemes [38,39], in which only a few selected message sets can be equality tested. An authorized user can determine not only whether two ciphertexts contain the same plaintext (without decryption) but also whether the plaintext belongs to the message set.

In this paper, we integrate the identity-based cryptography [40] into PKE-FET to propose a new concept of identity-based encryption with the filtered equality test (IBE-FET) for smart healthcare. A practical application scenario using IBE-FET is shown in Figure 2.

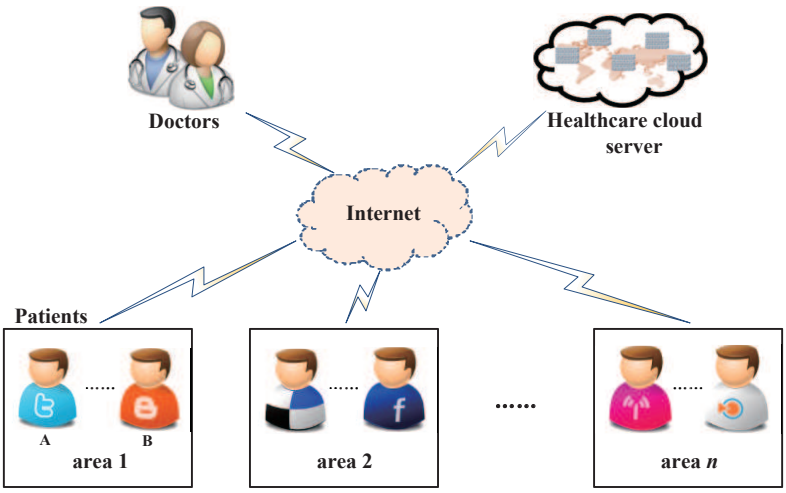

Figure 2. A practical application scenario of identity-based encryption with filtered equality test (IBE-FET).

In the smart healthcare system, there are three parties: doctors, the healthcare cloud server (HCS) and patients, where the patients are distributed in different areas. To ensure the privacy of patients, the sensitive data is encrypted during transmission. It is desired that the healthcare providers optimize the distribution of family doctors, and thus they need to search for the encrypted information. With the assumption that patients A and B with the same symptoms belong to area 1, A encrypts his privacy information (symptom and area) under the identity ID_A and the doctor's identity ID_D, and transmits the tuple $\{ID_A, \text{IBE-FET}(ID_D, ID_A, \text{symptom, area 1})\}$ to HCS. Additionally, A generates a warrant w_A and transmits to HCS. B transmits $\{ID_B, \text{IBE-FET}(ID_D, ID_B, \text{symptom, area 1})\}$ and w_B to HCS in the same way. Upon obtaining these data, the HCS could determine and search whether A and B are distributed in the same areas and have the same symptom. However, there is no knowledge what the real areas and symptom are. Then, the HCS sends the search result to the patients A and B, respectively, which allows them to share their medical experience with each other. Most important of all, the HCS can investigate the cause of the disease and arrange family doctors reasonably to improve the efficiency of healthcare. The above scenario can be extended to multi-user scenarios. For instance, more patients can get the warrant and send it to the HCS along with the requests and obtain feedback, indicating whether there are any patients belonging to the same area who have the same symptom features.

Besides, the IBE-FET scheme can also be applied to the smart grid system [41,42], which contains electricity suppliers, a power system cloud server and users. To protect the privacy and enhance the power quality of users, the privacy information (e.g., power consumers and location) is generally transmitted in encrypted form. Based on IBE-FET, the power system cloud server can determine and search whether there are any users belonging to the same area that have the same feature (e.g., power flow and peak loading). Then, they send the search result to the electricity suppliers for improvement of the power distribution and optimization of the power flow.

1.1. Our Contributions

This paper proposes an identity-based encryption with the filtered equality test (IBE-FET). The main contributions are summarized as follows:

- Based on secret sharing and bilinear pairing, an IBE-FET scheme is proposed, which does not use the certificate verification to solve the problems of certificate management.
- The security analysis indicates that the IBE-FET scheme is one-way secure against the chosen identity and ciphertext attack (OW-ID-CCA) based on the computational bilinear Diffie-Hellman assumption in the random oracle model.

- The performance analysis shows that the IBE-FET scheme achieves the function of a filtered equality test and a higher efficiency in terms of communication cost than the related scheme [39], and therefore the proposed scheme is more suitable for smart healthcare systems.

1.2. Organization

The organization of this paper is as follows: We will briefly discuss related work in Section 2 and review some preliminaries in Section 3; in Section 4, we introduce the framework of IBE-FET; a concrete IBE-FET scheme is put forward in Section 5; Section 6 proposes a formal security proof; comparison and performance evaluations are described in Section 7; and Section 8 concludes this paper.

2. Related Works

The concept of public key encryption with the keyword search (PKE-KS) was first put forward by Boneh et al. [17]. In PKE-KS, each user can use their private key to generate a token for a keyword and send the token to the tester. Upon receiving the token, the tester can determine the equality of ciphertexts. Then, some interesting extension schemes [18–22] were proposed to satisfy various requirements.

PKE-KS aims at testing the keyword's equality using a given trapdoor. However, it is not suitable for an equality test on ciphertexts by different public keys. In order to solve this problem, Yang et al. [23] proposed public key encryption with the equality test (PKE-ET). The so-called "equality test (ET)" refers to an authorized user who can verify the equality of two ciphertexts encrypted by different public keys, while the decryption keeps unavailable. However, in the PKE-ET scheme, anyone has the ability to execute the equality test without any authorization. As a fundamental security service, the authorization mechanism becomes increasingly important in modern smart system. The hierarchical key assignment techniques [43–46] were presented, which can provide fine-grained authentication and access control for the user. In order to mitigate the potential vulnerabilities and protect the user's privacy, Tang et al. [24] integrated the fine-grained authorization mechanism into PKE-ET. In this scheme, two users require cooperation to generate the token by running the authorization algorithm and send this token to the tester, with the tester authorized to verify the equality between the ciphertexts. In addition, Tang et al. [25] introduced the concept of coarse-grained authorization scheme, in this system, every user independently generates the token by running the authorization algorithm and sends it to the tester, who executes the equality test from their ciphertexts. In 2012, Tang [26] expanded [24] to a two-proxy agents setting, where two proxies require cooperation to perform the equality test. Lu et al. [27] introduced a stronger security model for PKE-ET to meet the different demands. In 2015, the public key encryption with the delegated equality test scheme (PKE-DET) was proposed by Ma et al. [28] and in this scheme every user can generate the delegation token independently for the cloud server. Different from PKE-DET, Huang et al. [29] introduced an efficient public key encryption with the authorized equality test (PKE-AET), a provision of two kinds of warrants (recipient warrants and ciphertext warrants) and allowance of the authorized users to use warrants to execute the equality test on two ciphertexts encrypted by different public keys. To satisfy various requirements, the public key encryption supporting equality test and flexible authorization (PKE-ET-FA) was proposed by Ma et al. [30]. In this scheme, four types of authorization were presented to strengthen the user privacy protection. However, it is inefficient due to using bilinear pairings. In 2016, Lin et al. [31] proposed an efficient PKE-ET-FA scheme without using bilinear pairing, which was more suitable for practice. In order to solve the certificate management problem, the identity-based encryption with equality test (IBE-ET) [32,33] was presented. To determine the equality of two ciphertexts encrypted under different access policies, the attribute-based encryption with equality test schemes (ABE-ET) [34–37] were put forward.

For making the equality test more flexible, based on bilinear pairing and secret sharing, Huang et al. [38,39] proposed the public key encryption with the filtered equality test (PKE-FET). In these schemes, the receiver selects n messages as a set Ω, and then the receiver can use a private key

and Ω to generate the warrant w and sends this warrant to someone, who can execute the equality test without decryption.

The PKE-FET scheme needs certification authority to ensure the authenticity of public keys; however, it is worth noting that the problems of certificate management arise. Accordingly, inspired by the concept of identity-based cryptography [40,47,48], we presented an identity-based encryption with the filtered equality test scheme (IBE-FET), simplifying the certificate management of PKE-FET.

3. Preliminaries

This section introduces some preliminaries, including bilinear pairing, secret sharing and security assumption.

3.1. Bilinear Pairing

Let $\mathbb{G}_1$, $\mathbb{G}_T$ be two cyclic groups of prime order q, and g is a generator of $\mathbb{G}_1$. $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$ is a bilinear pairing if the following three properties hold:

- **Bilinearity**: For all $u, v \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_q^*$, where $e(u^a, v^b) = e(u, v)^{ab}$.
- **Non-degeneracy**: $e(g, g) \neq 1$.
- **Computability**: It is an efficient algorithm to compute $e(u, v)$ for all $u, v \in \mathbb{G}_1$.

3.2. Secret Sharing

The idea of secret sharing is introduced in [49], with a secret value k assigned to n users. A trusted party holds k and randomly picks $t-1$ numbers $r_1, r_2, \cdots, r_{t-1}$ form t points on a 2-dimensional plane, which are $\{(0, k), (1, r_1), \cdots, (t-1, r_{t-1})\}$. According to these points, there is only one polynomial function ψ with $t-1$ degree determined. Then, the trusted party computes the points $(i, \psi(i))$ for user $i \in [t, n]$, in which all the points satisfy $y_i = \psi(i)$. By distributing these points, it formalizes a t-out-of-n secret sharing scheme. Therefore, as for any t or more than t users, it can reconstruct the polynomial function ψ and obtain the secret value k by computing $k = \psi(0)$, but if less than t users, it cannot rebuild the secret value k.

3.3. Assumption

Computational Bilinear Diffie-Hellman (CBDH) Problem: Let g be the generator of $\mathbb{G}_1$ and $a, b, c \in \mathbb{Z}_q^*$ be chosen at randomly. Given a tuple $(g, g^a, g^b, g^c) \in \mathbb{G}_1$, the task of CBDH problem is to compute $e(g, g)^{abc} \in \mathbb{G}_T$.

The probability of the algorithm $\mathcal{A}$ in solving the CBDH problem is defined as

$$Adv_{\mathcal{A}}^{CBDH} = \Pr[\mathcal{A}(g, g^a, g^b, g^c) = e(g, g)^{abc}] \leqslant \varepsilon.$$

Computational Bilinear Diffie-Hellman (CBDH) Assumption: The CBDH assumption holds if for any polynomial-time algorithm $\mathcal{A}$ solves the CBDH problem with the negligible probability.

4. Framework of IBE-FET

The system model, syntax and security model are described in the following sections.

4.1. System Model

The system model of IBE-FET includes four parts: private key generator (PKG), sender (patient), receiver (doctor) and the cloud server, as illustrated in Figure 3. All ciphertexts are generated by the senders under the receiver's identity and stored in the cloud server. The PKG's task is to generate the private keys for the users (senders and receivers) secretly. To compare the ciphertexts, the receiver generates the corresponding warrant using its private key and the message set, sending it to the cloud server; wherein the warrant denotes the trapdoor of authentication. As a result, with the

warrant, the cloud server is able to verify the equality between the ciphertexts without decryption and check whether the message belongs to the message set. The work of each part is described in more details below:

- **PKG**: It is responsible for generating the master key msk and the private key sk_{ID}, and then keeps msk by itself and sends sk_{ID} to the sender and receiver through a secure way.
- **Sender (patient)**: The sender encrypts their private date under the receiver's identity ID_R to generate the ciphertext C and stores it in the cloud server.
- **Receiver (doctor)**: Upon receiving the private key sk_{ID_R} from PKG, the receiver generates the warrant w and sends it to the cloud server. It is noted that the receiver can use the private key to decrypt the ciphertext at any time.
- **Cloud server**: With the warrant, the cloud server is in charge of executing the filtered equality test and returns a query result.

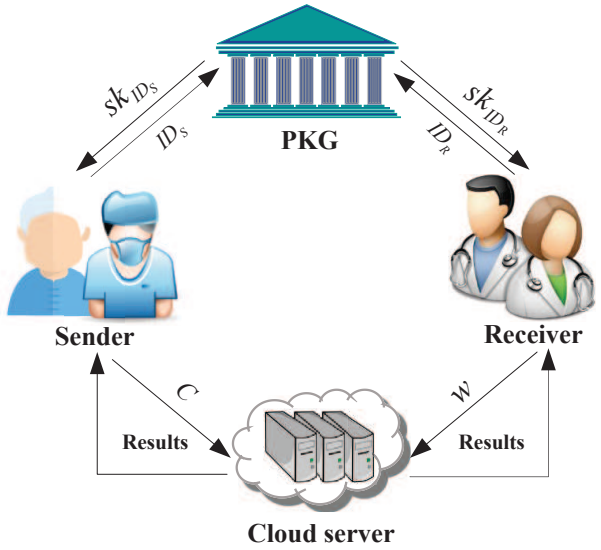

Figure 3. System model for IBE-FET.

The detail data flow of the filtered equality test (FET) is described in Figure 4.

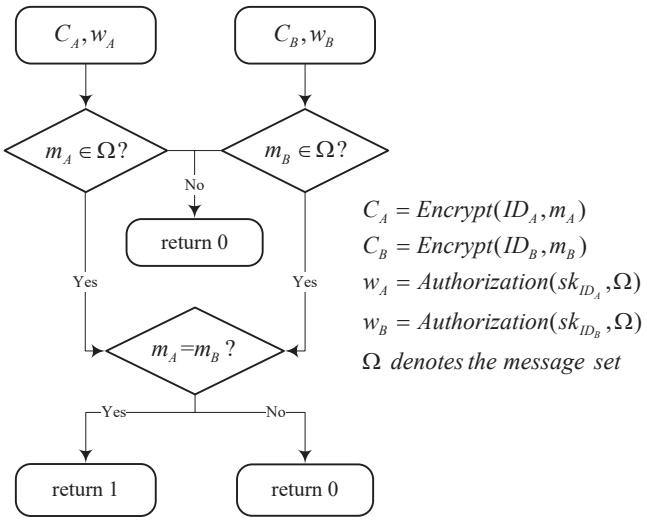

Figure 4. Flow chart of FET.

4.2. Syntax

The IBE-FET scheme consists of the following six algorithms: setup, extract, encrypt, decrypt, authorization and filtered equality test. Let Δ denote message space and $\Omega \subseteq \Delta$ denote the message set.

Setup: Taking a security parameter k as input, this algorithm outputs the master key msk and the system parameters PP.

Extract: Taking the master key msk and the identity ID as input, this algorithm outputs the private key sk_{ID}.

Encrypt: Taking the system parameters PP, the plaintext $m \in \Delta$ and the identity ID as input, this algorithm outputs the ciphertext C.

Decrypt: Taking the system parameters PP, the ciphertext C and the private key sk_{ID} as input, this algorithm outputs the corresponding plaintext m.

Authorization: Taking the system parameters PP, the identity ID, the private key sk_{ID} and the message set Ω as input, this algorithm outputs the warrant w_{ID}.

Filtered equality test: Taking the system parameters PP, the ciphertexts C_A and C_B, the warrants w_{ID_A} and w_{ID_B} as input, this algorithm returns 1 if $m_A \in \Omega$, $m_B \in \Omega$ and $m_A = m_B$. Otherwise, it returns 0.

For the property of consistency, the following conditions must be satisfied.

Correctness: When sk_{ID} is generated by the **Extract** algorithm given ID, then, for all $m \in \Delta$, $\Pr[\text{Decrypt}(\text{Encrypt}(ID, m), sk_{ID}) = m] = 1$.

Perfect consistency: When w_{ID_A} and w_{ID_B} are generated by the **Authorization** algorithm given ID_A, ID_B and Ω, then, for all $m_A \in \Omega$, $m_B \in \Omega$ and $m_A = m_B$, the filtered equality test algorithm must return 1.

Computational soundness: When w_{ID_A} and w_{ID_B} are generated by the **Authorization** algorithm given ID_A, ID_B and Ω, then, for all $m_A \in \Omega$, $m_B \in \Omega$ and $m_A \neq m_B$, the probability that the filtered equality test algorithm returns 1 is negligible.

4.3. Security Model

The security of IBE-FET needs to satisfy one-way security against the chosen identity and ciphertext attack (OW-ID-CCA), which is defined by an interactive game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Setup: $\mathcal{C}$ generates the master key msk and the system parameters $PP_{IBE-FET}$ by running the **Setup** algorithm. Then $\mathcal{C}$ sends $PP_{IBE-FET}$ to $\mathcal{A}$ and keeps msk by itself.

Phase 1: $\mathcal{A}$ makes the following queries for polynomial number of times.

- **Hash** H **queries**: $\mathcal{A}$ submits a query, then $\mathcal{C}$ returns a random value to $\mathcal{A}$.
- **Private key queries**: $\mathcal{A}$ submits the identity ID_j to $\mathcal{C}$, then $\mathcal{C}$ runs the **Extract** algorithm and returns the private key sk_{ID_j} to $\mathcal{A}$.
- **Decryption queries**: $\mathcal{A}$ submits the identity ID_j and the ciphertext C_j to $\mathcal{C}$, then $\mathcal{C}$ runs the **Extract** algorithm to obtain sk_{ID_j} and runs the **Decrypt** algorithm to return the plaintext m_j to $\mathcal{A}$.
- **Authorization queries**: $\mathcal{A}$ submits the identity ID_j and the message set Ω_j to $\mathcal{C}$, then $\mathcal{C}$ runs the **Extract** algorithm to obtain sk_{ID_j} and runs the **Authorization** algorithm to return the warrant w_{ID_j} to $\mathcal{A}$.

Challenge: $\mathcal{A}$ submits a challenge identity ID^* to $\mathcal{C}$, where ID^* does not appear in private key queries in **Phase 1**. $\mathcal{C}$ randomly chooses a plaintext $m^* \in \Delta$ and sets C^* be the challenge ciphertext. Finally, $\mathcal{C}$ sends C^* to $\mathcal{A}$.

Phase 2: Similar to **Phase 1**.

- **Hash** H **queries**: $\mathcal{C}$ responds as in **Phase 1**.
- **Private key queries**: If $ID_j \neq ID^*$, $\mathcal{C}$ responds as in **Phase 1**. Otherwise, $\mathcal{C}$ returns $\perp$.
- **Decryption queries**: If $(ID_j, C_j) \neq (ID^*, C^*)$, $\mathcal{C}$ responds as in **Phase 1**. Otherwise, $\mathcal{C}$ returns $\perp$.

- **Authorization queries**: $\mathcal{C}$ responds as in **Phase 1**.

Guess: $\mathcal{A}$ outputs a guess m' and wins the above game if $m' = m^*$.
The advantage of $\mathcal{A}$ winning the above game is defined as

$$Adv_{IBE-FET,\mathcal{A}}^{OW-ID-CCA} = \Pr[m' = m^*].$$

Definition 1. *The IBE-FET scheme is OW-ID-CCA security if for any adversaries $\mathcal{A}$, $Adv_{IBE-FET,\mathcal{A}}^{OW-ID-CCA}$ is negligible.*

Next, the security of the public key encryption (PKE) scheme (which will be mentioned later) needs to satisfy one-way security against the chosen ciphertext attack (OW-CCA), which is defined by an interactive game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Setup: $\mathcal{C}$ generates the private key sk and the system parameters PP_{PKE} by running the **Setup** algorithm. Then $\mathcal{C}$ sends PP_{PKE} to $\mathcal{A}$ and keeps sk by itself.

Phase 1: $\mathcal{A}$ makes the following queries for polynomial number of times.

- **Hash** H **queries**: $\mathcal{A}$ submits a query, then $\mathcal{C}$ returns a random value to $\mathcal{A}$.
- **Decryption queries**: $\mathcal{A}$ submits the ciphertext C_i to $\mathcal{C}$, then $\mathcal{C}$ runs the **Decrypt** algorithm and returns the plaintext m_i to $\mathcal{A}$.

Challenge: $\mathcal{C}$ randomly chooses a challenge plaintext $m^* \in \Delta$ and runs the **Encrypt** algorithm to obtain the challenge ciphertext C^*. Finally, $\mathcal{C}$ sends C^* to $\mathcal{A}$.

Phase 2: Similar to **Phase 1**.

- **Hash** H **queries**: $\mathcal{C}$ responds as in **Phase 1**.
- **Decryption queries**: If $C_i \neq C^*$, $\mathcal{C}$ responds as in **Phase 1**. Otherwise, $\mathcal{C}$ returns $\perp$.

Guess: $\mathcal{A}$ outputs a guess m' and wins the above game if $m' = m^*$.
The advantage of $\mathcal{A}$ wining the above game is defined as

$$Adv_{PKE,\mathcal{A}}^{OW-CCA} = \Pr[m' = m^*].$$

Definition 2. *The PKE scheme is OW-CCA security if, for any adversaries $\mathcal{A}$, $Adv_{PKE,\mathcal{A}}^{OW-CCA}$ is negligible.*

5. The Proposed Scheme

In this section, a detailed construction of IBE-FET is proposed.

- **Setup**: Given a security parameter k, the PKG executes as follows:

  (1) Chooses a bilinear pairing: $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$, where $\mathbb{G}_1$ and $\mathbb{G}_T$ are two cyclic groups with prime order q, g is a generator of $\mathbb{G}_1$.
  (2) Randomly picks $u, s_0, s_1, \cdots, s_n \in \mathbb{Z}_q^*$ and computes $U = g^u$, $S_0 = g^{s_0}$, $S_1 = g^{s_1}$, $\cdots$, $S_n = g^{s_n}$.
  (3) Chooses four one-way hash functions $H_1 : \{0,1\}^* \to \mathbb{G}_1$, $H_2 : \{0,1\}^{l_1} \to \mathbb{Z}_q^*$, $H_3 : \mathbb{G}_T \to \{0,1\}^{l_1+l_2}$, $H_4 : \{0,1\}^{l_1} \to \mathbb{G}_T$, where l_1 is the length of the message and l_2 is the length of $\mathbb{Z}_q^*$.

  The system parameters are $PP_{IBE-FET} = \{e, q, \mathbb{G}_1, \mathbb{G}_T, g, U, S_0, S_1, \cdots, S_n, H_1, H_2, H_3, H_4\}$ and the master key are $msk = \{u, s_0, s_1, \cdots, s_n\}$.
- **Extract**: Given the identity ID and the master key $u, s_0, s_1, \cdots, s_n$, PKG computes $h_{ID} = H_1(ID)$ and the private key $sk_{ID} = \{h_{ID}^u, h_{ID}^{s_0}, h_{ID}^{s_1}, \cdots, h_{ID}^{s_n}\}$.
- **Encrypt**: Given the message m and the identity ID, the sender executes as follows:

(1) Randomly chooses $r, t \in \mathbb{Z}_q^*$.

(2) Computes $h_{ID} = H_1(ID)$, $h = H_2(m)$, $S = \prod_{i=0}^{n} S_i^{rh^i}$,

$C_1 = \{C_{1,0} = g^r, C_{1,1} = g^{rh}, \cdots, C_{1,n} = g^{rh^n}\}$,

$C_2 = g^t$,

$C_3 = (m||r) \oplus H_3(e(h_{ID}, U)^t)$,

$C_4 = e(h_{ID}, S) \cdot H_4(m)$.

The ciphertext is $C = \{C_1, C_2, C_3, C_4\}$, where $C_1 = (C_{1,0}, C_{1,1}, \cdots, C_{1,n})$.

- **Decrypt**: Given the ciphertext C and the private key sk_{ID}, the receiver executes as follows:

(1) Computes $C_3 \oplus H_3(e(h_{ID}^u, C_2)) = m||r$ and $h = H_2(m)$.

(2) Verifies

$$C_{1,i} = g^{rh^i} \text{ and } C_4 = \prod_{i=0}^{n} e(h_{ID}^{s_i}, C_{1,i}) \cdot H_4(m)$$

for all $i \in [0, n]$. If holds, it outputs m. Otherwise, it outputs $\perp$.

- **Authorization**: Given the message set $\Omega = \{m_1, m_2, \cdots, m_n\}$ and the private key $sk_{ID} = \{h_{ID}^{s_0}, h_{ID}^{s_1}, \cdots, h_{ID}^{s_n}\}$, the receiver performs the following steps:

(1) Computes a n-degree polynomial function $f(x) = \prod_{i=1}^{n} (x - H_2(m_i)) = \sum_{i=0}^{n} a_i x^i$ and obtains the coefficient $a_0, a_1, \cdots, a_n$.

(2) Computes $w_{ID,i} = h_{ID}^{s_i} \cdot h_{ID}^{a_i}$ for all $i \in [0, n]$ and sends the warrant $w_{ID} = \{w_{ID,0}, w_{ID,1}, \cdots, w_{ID,n}\}$ to the cloud server.

- **Filtered equality test**: Given two ciphertexts $C_A = \{C_{A,1} = (C_{A,1,0}, C_{A,1,1}, \cdots, C_{A,1,n}), C_{A,2}, C_{A,3}, C_{A,4}\}$ and $C_B = \{C_{B,1} = (C_{B,1,0}, C_{B,1,1}, \cdots, C_{B,1,n}), C_{B,2}, C_{B,3}, C_{B,4}\}$, two warrants $w_{ID_A} = \{w_{ID_A,0}, w_{ID_A,1}, \cdots, w_{ID_A,n}\}$ and $w_{ID_B} = \{w_{ID_B,0}, w_{ID_B,1}, \cdots, w_{ID_B,n}\}$, the cloud server executes as follows:

(1) Computes $z_A = \frac{C_{A,4}}{\prod_{i=0}^{n} e(C_{A,1,i}, w_{ID_A,i})}$ and $z_B = \frac{C_{B,4}}{\prod_{i=0}^{n} e(C_{B,1,i}, w_{ID_B,i})}$.

(2) Checks whether $z_A = z_B$ or not. It outputs 1 if $z_A = z_B$, which means $m_A \in \Omega$, $m_B \in \Omega$ and $m_A = m_B$. Otherwise, it outputs 0.

Correctness: The decryption algorithm computes

$$\begin{aligned} & C_3 \oplus H_3(e(h_{ID}^u, C_2)) \\ &= (m||r) \oplus H_3(e(h_{ID}, U)^t) \oplus H_3(e(h_{ID}^u, g^t)) \\ &= (m||r) \oplus H_3(e(h_{ID}, g^u)^t) \oplus H_3(e(h_{ID}^u, g^t)) \\ &= m||r \end{aligned}$$

Then, let $h = H_2(m)$, it checks both $C_{1,i} = g^{rh^i}$ and $C_4 = \prod_{i=0}^{n} e(h_{ID}^{s_i}, C_{1,i}) \cdot H_4(m) = \prod_{i=0}^{n} e(h_{ID}^{s_i}, g^{rh^i}) \cdot H_4(m) = e(h_{ID}, g)^{r \sum_{i=0}^{n} s_i h^i} \cdot H_4(m) = e(h_{ID}, S) \cdot H_4(m)$ for all $i \in [0, n]$. It is straightforward that the correctness holds along with the decryption algorithm.

Perfect consistency: On input (C_A, w_{ID_A}) and (C_B, w_{ID_B}), the filtered equality test algorithm obtains z_A by computing

$$z_A = \frac{C_{A,4}}{\prod_{i=0}^{n} e(C_{A,1,i}, w_{ID_A,i})} = \frac{\prod_{i=0}^{n} e(h_{ID_A}, S_i)^{rh^i} \cdot H_4(m_A)}{\prod_{i=0}^{n} e(g^{rh^i}, h_{ID_A}^{(s_i+a_i)})}$$
$$= \frac{e(h_{ID_A}, g)^{r\sum_{i=0}^{n} s_i h^i} \cdot H_4(m_A)}{e(g, h_{ID_A})^{r\sum_{i=0}^{n}(s_i h^i + a_i h^i)}} = \frac{e(h_{ID_A}, g)^{r\sum_{i=0}^{n} s_i h^i} \cdot H_4(m_A)}{e(g, h_{ID_A})^{r\sum_{i=0}^{n} s_i h^i + rf(H_2(m_A))}}.$$

If $m_A \in \Omega$, we have $f(H_2(m_A)) = \sum_{i=0}^{n} a_i H_2(m_A)^i = 0$, therefore $z_A = \frac{e(h_{ID_A}, g)^{r\sum_{i=0}^{n} s_i h^i} \cdot H_4(m_A)}{e(g, h_{ID_A})^{r\sum_{i=0}^{n} s_i h^i}}$ $= H_4(m_A)$. Similarly, if $m_B \in \Omega$, we can obtain $z_B = H_4(m_B)$. If $m_A = m_B$, then $z_A = z_B$. The filtered equality test algorithm outputs 1.

Computational soundness: For any $m_A \in \Omega$ and $m_B \in \Omega$, by the inference of consistency, z_A and z_B will be computed as $z_A = H_4(m_A)$ and $z_B = H_4(m_B)$, respectively. If $m_A \neq m_B$, then $z_A \neq z_B$, this is because $H_4(m)$ is a collision resistant function. Hence the probability that the filtered equality test algorithm returns 1 is negligible. The computational soundness holds.

6. Security Proof

In this section, based on CBDH assumption, the proposed IBE-FET scheme is proved to be OW-ID-CCA security in the random oracle model. The detail of security proof is shown in Figure 5. Using the same method [32,33,40], we prove the security of the proposed scheme in two steps. We first show that an OW-ID-CCA attack on IBE-FET can be converted to an OW-CCA attack on PKE, then, we show that PKE is OW-CCA secure if the DBDH assumption holds.

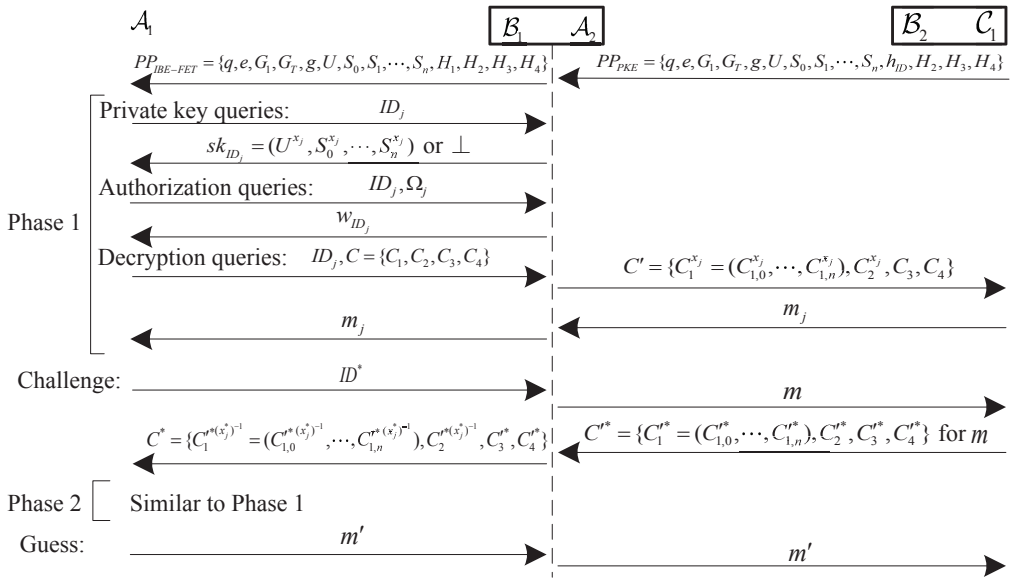

Figure 5. The security proof of IBE-FET.

Theorem 1. *Supposing there is an OW-ID-CCA adversary $\mathcal{A}$ that is able to break the proposed scheme with a non-negligible probability ε, then there exists an algorithm $\mathcal{B}$ that solves the CBDH problem with the probability at least $\varepsilon' = \frac{\varepsilon}{e(q_{sk}+q_{aut}+q_d+1)(q_{H_3}+1)} - \frac{q_{H_3} \cdot q_d}{2^{l_1+l_2}(q_{H_3}+1)}$, where q_{sk} is the number of the private key queries, q_{aut} is*

the number of the authorization queries, q_d is the number of the decryption queries and q_{H_3} is the number of H_3 queries, l_1 is the length of the message and l_2 is the length of $\mathbb{Z}_q^*$.

Proof. Theorem 1 is proved based on the following Theorem 2 and Theorem 3. □

To prove Theorem 1, we must convert the OW-ID-CCA attack on an IBE-FET scheme to an OW-CCA attack on a PKE scheme. A related PKE scheme is described below.

- **Setup**: Given a security parameter k, the system executes as follows:

  (1) Chooses a bilinear pairing: $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$, where $\mathbb{G}_1$ and $\mathbb{G}_T$ are two cyclic groups with prime order q, g is a generator of $\mathbb{G}_1$.
  (2) Randomly picks $h_{ID} \in \mathbb{G}_1$, $u, s_0, s_1, \cdots, s_n \in \mathbb{Z}_q^*$ and computes $U = g^u$, $S_0 = g^{s_0}$, $S_1 = g^{s_1}$, $\cdots$, $S_n = g^{s_n}$.
  (3) Chooses three one-way hash functions: $H_2 : \{0,1\}^{l_1} \to \mathbb{Z}_q^*$, $H_3 : \mathbb{G}_T \to \{0,1\}^{l_1+l_2}$, $H_4 : \{0,1\}^{l_1} \to \mathbb{G}_T$, where l_1 is the length of the message and l_2 is the length of $\mathbb{Z}_q^*$.

  The system parameters are $PP_{PKE} = \{e, \mathbb{G}_1, \mathbb{G}_T, q, g, U, S_0, S_1, \cdots, S_n, h_{ID}, H_2, H_3, H_4\}$ and the pravate key are $sk_{ID} = \{h_{ID}^u, h_{ID}^{s_0}, h_{ID}^{s_1}, \cdots, h_{ID}^{s_n}\}$.
- **Encrypt**: Given the message m, the sender executes as follows:

  (1) Randomly chooses $r, t \in \mathbb{Z}_q^*$.
  (2) Computes $h = H_2(m)$, $S = \prod_{i=0}^{n} S_i^{rh^i}$, $C_1 = \{C_{1,0} = g^r, C_{1,1} = g^{rh}, \cdots, C_{1,n} = g^{rh^n}\}$, $C_2 = g^t$, $C_3 = (m||r) \oplus H_3(e(h_{ID}, U)^t)$, $C_4 = e(h_{ID}, S) \cdot H_4(m)$.

  The ciphertext is $C = \{C_1, C_2, C_3, C_4\}$, where $C_1 = (C_{1,0}, C_{1,1}, \cdots, C_{1,n})$.
- **Decrypt**: Given the ciphertexts C and the private key sk_{ID}, the receiver works as follows:

  (1) Computes $C_3 \oplus H_3(e(h_{ID}^u, C_2)) = m||r$ and $h = H_2(m)$.
  (2) Verifies

$$C_{1,i} = g^{rh^i} \text{ and } C_4 = \prod_{i=0}^{n} e(h_{ID}^{s_i}, C_{1,i}) \cdot H_4(m)$$

  for all $i \in [0, n]$. If holds, it outputs m. Otherwise, it outputs $\perp$.

Theorem 2. *Supposing there is an OW-ID-CCA adversary $\mathcal{A}_1$ that is able to break the proposed IBE-FET scheme with a non-negligible probability ε_1, then there exists an OW-CCA adversary $\mathcal{B}_1$ that can break the PKE scheme with the probability at least $\varepsilon_1' = \frac{\varepsilon_1}{e(q_{sk}+q_{aut}+q_d+1)}$, where q_{sk} is the number of the private key queries, q_{aut} is the number of the authorization queries and q_d is the number of the decryption queries.*

Proof. In order to convert an OW-ID-CCA attack on IBE-FET to an OW-CCA attack on PKE, we can construct a simulator $\mathcal{C}_1$ to execute the game between $\mathcal{A}_1$ and $\mathcal{B}_1$. □

Initialization: $\mathcal{C}_1$ runs the Setup algorithm of PKE and returns the system parameters $PP_{PKE} = \{q, e, \mathbb{G}_1, \mathbb{G}_T, g, U, S_0, S_1, \cdots, S_n, h_{ID}, H_2, H_3, H_4\}$ to $\mathcal{B}_1$. $\mathcal{A}_1$ interacts with $\mathcal{B}_1$ as follows.

Setup: $\mathcal{B}_1$ chooses a hash function H_1 and returns the system parameters $PP_{IBE-FET} = \{q, e, \mathbb{G}_1, \mathbb{G}_T, g, U, S_0, S_1, \cdots, S_n, H_1, H_2, H_3, H_4\}$ to $\mathcal{A}_1$. For the quickly respond and consistency, $\mathcal{B}_1$ maintains an initially empty list H_1^{list} of tuples $(ID_j, h_{1,j}, x_j, c_j)$.

Phase 1: $\mathcal{A}_1$ makes the following queries.

- **Hash H_1 queries**: $\mathcal{A}_1$ submits a query on ID_j, $\mathcal{B}_1$ checks the list H_1^{list} and performs as below:

- If H_1^{list} contains $(ID_j, h_{1,j}, x_j, c_j)$, $\mathcal{B}_1$ responds with previous value $h_{1,j}$ to $\mathcal{A}_1$.
- If H_1^{list} doesn't contain $(ID_j, h_{1,j}, x_j, c_j)$, based on the Coron's technology [50], $\mathcal{B}_1$ tosses a coin $c_j \in \{0, 1\}$ that yield 0 with probability δ and 1 with probability $1-\delta$. $\mathcal{B}_1$ randomly chooses $x_j \in \mathbb{Z}_q^*$. If $c_j = 0$, $\mathcal{B}_1$ computes $h_{1,j} = g^{x_j}$. If $c_j = 1$, $\mathcal{B}_1$ computes $h_{1,j} = h_{ID}^{x_j}$. Finally, $\mathcal{B}_1$ adds the tuple $(ID_j, h_{1,j}, x_j, c_j)$ to the list H_1^{list} and returns $h_{1,j}$ to $\mathcal{A}_1$.

- **Private key queries**: $\mathcal{A}_1$ submits a private key query on ID_j, $\mathcal{B}_1$ makes the hash H_1 query on ID_j to obtain the corresponding tuple $(ID_j, h_{1,j}, x_j, c_j)$.

  - If $c_j = 0$, $\mathcal{B}_1$ returns $sk_{ID_j} = \{U^{x_j}, S_0^{x_j}, S_1^{x_j}, \cdots, S_n^{x_j}\}$ to $\mathcal{A}_1$.
  - If $c_j = 1$, $\mathcal{B}_1$ returns $\perp$.

- **Decryption queries**: $\mathcal{A}_1$ submits a decryption query on ID_j and $C = \{C_1, C_2, C_3, C_4\}$, $\mathcal{B}_1$ makes the hash H_1 query on ID_j to obtain the corresponding tuple $(ID_j, h_{1,j}, x_j, c_j)$.

  - If $c_j = 0$, $\mathcal{B}_1$ obtains $sk_{ID_j} = \{U^{x_j}, S_0^{x_j}, \cdots, S_n^{x_j}\}$ and decrypts C using sk_{ID_j}.
  - If $c_j = 1$, $\mathcal{B}_1$ obtains $h_{1,j} = h_{ID}^{x_j}$ and computes $sk_{ID_j} = \{(h_{ID}^{x_j})^u, (h_{ID}^{x_j})^{s_0}, (h_{ID}^{x_j})^{s_1}, \cdots, (h_{ID}^{x_j})^{s_n}\}$. Then $\mathcal{B}_1$ sets $C' = \{C_1^{x_j} = (C_{1,0}^{x_j}, C_{1,1}^{x_j}, \cdots, C_{1,n}^{x_j}), C_2^{x_j}, C_3, C_4\}$. Note that the IBE-FET decryption of C using $sk_{ID_j} = \{(h_{ID}^{x_j})^u, (h_{ID}^{x_j})^{s_0}, (h_{ID}^{x_j})^{s_1}, \cdots, (h_{ID}^{x_j})^{s_n}\}$ is the same as the PKE decryption of C' using $sk_{ID_j} = \{h_{ID}^u, h_{ID}^{s_0}, h_{ID}^{s_1}, \cdots, h_{ID}^{s_n}\}$ because $e((h_{ID}^{x_j})^u, C_2) = e(h_{ID}^u, C_2^{x_j})$ and $e((h_{ID}^{x_j})^{s_i}, C_{1,i}) = e(h_{ID}^{s_i}, C_{1,i}^{x_j})$ for any $i \in [0, n]$. $\mathcal{B}_1$ makes the decryption query on C' to $\mathcal{C}_1$ and returns the response of $\mathcal{C}_1$ to $\mathcal{A}_1$.

- **Authorization queries**: $\mathcal{A}_1$ submits an authorization query on ID_j and the message set Ω_j, $\mathcal{B}_1$ makes the private key query on ID_j to obtain sk_{ID_j}. Then $\mathcal{B}_1$ runs the authorization algorithm and returns the warrant w_{ID_j} to $\mathcal{A}_1$.

Challenge: $\mathcal{A}_1$ chooses the challenge identity ID^* and returns it to $\mathcal{B}_1$. Here, ID^* does not appear in the private key queries of Phase 1. Then $\mathcal{B}_1$ makes the hash H_1 query on ID^* to get the tuple $(ID^*, h_{1,j}^*, x_j^*, c_j^*)$ and executes as follows:

- If $c_j^* = 0$, $\mathcal{B}_1$ returns $\perp$.
- If $c_j^* = 1$, $\mathcal{C}_1$ randomly chooses m^* and returns a PKE challenge ciphertext $C'^* = \{C_1'^* = (C_{1,0}'^*, C_{1,1}'^*, \cdots, C_{1,n}'^*), C_2'^*, C_3'^*, C_4'^*\}$ on m^* to $\mathcal{B}_1$. Then $\mathcal{B}_1$ returns $C^* = \{C_1'^{*(x_j^*)^{-1}} = (C_{1,0}'^{*\,(x_j^*)^{-1}}, C_{1,1}'^{*\,(x_j^*)^{-1}} \cdots, C_{1,n}'^{*\,(x_j^*)^{-1}}), C_2'^{*(x_j^*)^{-1}}, C_3'^*, C_4'^*\}$ to $\mathcal{A}_1$.

Phase 2: $\mathcal{A}_1$ makes queries as done in Phase 1.

- **Private key queries**: If $ID^* \neq ID_j$, $\mathcal{B}_1$ responds as in Phase 1. Otherwise, $\mathcal{B}_1$ returns $\perp$.
- **Decryption queries**: If $(ID^*, C^*) \neq (ID_j, C_j)$, $\mathcal{B}_1$ responds as in Phase 1. Otherwise, $\mathcal{B}_1$ returns $\perp$.
- **Authorization queries**: $\mathcal{B}_1$ responds as in Phase 1.

Guess: $\mathcal{A}_1$ outputs a guess m' for m^*. $\mathcal{B}_1$ outputs a guess m' for m^*.

We define the following three events:

- ζ_1 : $\mathcal{B}_1$ aborts in the private key query during Phase 1 or Phase 2.
- ζ_2 : $\mathcal{B}_1$ aborts in the challenge phase.
- ζ_3 : $\mathcal{B}_1$ aborts in the decryption query in Phase 2.

Thus, we have

$$\Pr[\neg\zeta_1 \wedge \neg\zeta_2 \wedge \neg\zeta_3] \geqslant (1-\delta)\delta^{(q_{sk}+q_{aut}+q_d)}.$$

Clearly, $(1-\delta)\delta^{(q_{sk}+q_{aut}+q_d)}$ can obtain the maximized when $\delta = 1 - \frac{1}{(q_{sk}+q_{aut}+q_d+1)}$. The probability that $\mathcal{B}_1$ does not abort is at least $\frac{1}{(q_{sk}+q_{aut}+q_d+1)}$. Therefore, the advantage of $\mathcal{B}_1$ is at least $\frac{\varepsilon_1}{e(q_{sk}+q_{aut}+q_d+1)}$.

Theorem 3. *Supposing there is an OW-CCA adversary $\mathcal{A}_2$ that is able to break the PKE scheme with a non-negligible probability ε_2, then there exists an algorithm $\mathcal{B}_2$ that solves the CBDH problem with the probability at least $\varepsilon_2' = \frac{\varepsilon_2}{q_{H_3}+1} - \frac{q_{H_3}\cdot q_d}{(q_{H_3}+1)\cdot 2^{l_1+l_2}}$, where q_{H_3} is the number of H_3 queries and q_d is the number of the decryption queries, l_1 is the length of the message and l_2 is the length of $\mathbb{Z}_q^*$.*

Proof. Let $\varepsilon_2 = Adv_{PKE,\mathcal{A}_2}^{OW-CCA}$ represent the advantage of $\mathcal{A}_2$ in the OW-CCA security game. According to schemes [23–31], this theorem is proved by performing a series of games. Let Q_i denote the event that $m' = m^*$ in Game i $(i = 0, 1, 2)$. We define the Game 0 to be the real security game against the adversary in Definition 2. Then, we can modify the last game in an indistinguishable way to obtain the next game. The adversary has no advantage unconditionally in last game, thus he can make the queries many times, then the event will happen in the next game. Since each game is indistinguishable from the next, to prove the real security game, we can show that the probability of an event is negligible if the DBDH assumption holds. The detailed process is shown as follows. □

Game 0:

1. Initial phase: $\mathcal{B}_2$ generates $u, s_0, s_1, \cdots, s_n \in \mathbb{Z}_q^*$ and $h_{ID} \in \mathbb{G}_1$ by running the Setup algorithm, then computes $U = g^u$, $S_0 = g^{s_0}, S_1 = g^{s_1}, \cdots, S_n = g^{s_n}$. Finally, $\mathcal{B}_2$ returns the system parameters $PP_{PKE} = \{q, e, \mathbb{G}_1, \mathbb{G}_T, g, U, S_0, S_1, \cdots, S_n, h_{ID}, H_2, H_3, H_4\}$ to $\mathcal{A}_2$. For the quickly respond and consistency, $\mathcal{B}_2$ maintains an initially empty list H_3^{list} of tuples $(\Phi_i, h_{3,i})$.

2. Query phase: $\mathcal{B}_2$ works as follows:

- **Hash H_3 queries**: $\mathcal{A}_2$ makes a hash H_3 query on Φ_i, $\mathcal{B}_2$ checks the list H_3^{list} and performs as follows.
  - If H_3^{list} includes $(\Phi_i, h_{3,i})$, $\mathcal{B}_2$ returns $h_{3,i}$ to $\mathcal{A}_2$.
  - If H_3^{list} doesn't include $(\Phi_i, h_{3,i})$, $\mathcal{B}_2$ selects a random sting $h_{3,i} \in \{0,1\}^{l_1+l_2}$ and returns $h_{3,i}$ to $\mathcal{A}_2$.
- **Decryption queries**: $\mathcal{A}_2$ makes a decryption query on C, $\mathcal{B}_2$ returns m to $\mathcal{A}_2$ by running the decryption algorithm using the private key.

3. Challenge phase: For any m^*, $\mathcal{B}_2$ randomly chooses $r, t \in \mathbb{Z}_q^*$ and computes $h = H_2(m^*)$, $S = \prod_{i=0}^{n} S_i^{rh^i}$ and defines the challenge ciphertexts

$$C^* = \{C_1^* = (C_{1,0}^*, C_{1,1}^*, \cdots, C_{1,n}^*), C_2^*, C_3^*, C_4^*\}$$

as follows:

$C_1^* = \{C_{1,0}^* = g^r, C_{1,1}^* = g^{rh}, \cdots, C_{1,n}^* = g^{rh^n}\}$,
$C_2^* = g^t$,
$C_3^* = (m^*||r) \oplus H_3(e(h_{ID}, U)^t)$,
$C_4^* = e(h_{ID}, S) \cdot H_4(m^*)$.

4. Output phase: $\mathcal{A}_2$ outputs a guess m' for m^*.

Thus, the advantage of $\mathcal{A}_2$ winning in Game 0 is

$$Adv_{PKE,\mathcal{A}_2}^{OW-CCA} = Pr[Q_0]. \tag{1}$$

Game 1:

1. Initial phase: $\mathcal{B}_2$ responds as in Game 0.
2. Query phase: $\mathcal{B}_2$ works as follows:

- **Hash** H_3 **queries**: $\mathcal{A}_2$ makes a hash H_3 query on Φ_i, $\mathcal{B}_2$ checks the list H_3^{list} and performs as follows.
  - If H_3^{list} includes $(\Phi_i, h_{3,i})$. When $\Phi_i = e(h_{ID}, U)^t$, $\mathcal{B}_2$ defines $\omega_1^* = H_3(e(h_{ID}, U)^t)$ as $h_{3,i}$ and returns ω_1^* to $\mathcal{A}_2$; otherwise, $\mathcal{B}_2$ returns $h_{3,i}$ to $\mathcal{A}_2$.
  - If H_3^{list} doesn't include $(\Phi_i, h_{3,i})$, $\mathcal{B}_2$ selects a random sting $h_{3,i} \in \{0,1\}^{l_1+l_2}$ and returns $h_{3,i}$ to $\mathcal{A}_2$.
- **Decryption queries**: $\mathcal{B}_2$ responds a decryption query as in Game 0.

3. Challenge phase: For any m^*, $\mathcal{B}_2$ randomly chooses $r, t \in \mathbb{Z}_q^*$, $\omega_1^* \in \{0,1\}^{l_1+l_2}$ and computes $h = H_2(m^*)$, $S = \prod_{i=0}^{n} S_i^{rh^i}$ and defines the challenge ciphertexts

$$C^* = \{C_1^* = (C_{1,0}^*, C_{1,1}^*, \cdots, C_{1,n}^*), C_2^*, C_3^*, C_4^*\}$$

as follows:
$C_1^* = \{C_{1,0}^* = g^r, C_{1,1}^* = g^{rh}, \cdots, C_{1,n}^* = g^{rh^n}\}$,
$C_2^* = g^t$,
$C_3^* = (m^*||r) \oplus \omega_1^*$,
$C_4^* = e(h_{ID}, S) \cdot H_4(m^*)$.
4. Update phase: $\mathcal{B}_2$ adds the tuple $(e(h_{ID}, U)^t, \omega_1^*)$ to the list H_3^{list}.
5. Output phase: $\mathcal{A}_2$ outputs a guess m' for m^*.

Compared to Game 0, the value of H_3 is replaced by a random value ω_1^* in Game 1. According to the random oracle model, the advantage of $\mathcal{A}_2$ winning in Game 1 is identical to Game 0. Thus

$$Adv_{PKE,\mathcal{A}_2}^{OW-CCA} = Pr[Q_0] = Pr[Q_1]. \quad (2)$$

Game 2:

1. Initial phase: $\mathcal{B}_2$ responds as in Game 1.
2. Query phase: $\mathcal{B}_2$ works as follows:

- **Hash** H_3 **queries**: $\mathcal{A}_2$ makes a hash H_3 query on Φ_i, $\mathcal{B}_2$ checks the list H_3^{list} and performs as follows.
  - If H_3^{list} includes $(\Phi_i, h_{3,i})$. When $\Phi_i = e(h_{ID}, U)^t$, $\mathcal{B}_2$ returns $\perp$. Define this event as E_1; otherwise, $\mathcal{B}_2$ returns $h_{3,i}$ to $\mathcal{A}_2$.
  - If H_3^{list} does not include $(\Phi_i, h_{3,i})$, $\mathcal{B}_2$ selects a random sting $h_{3,i} \in \{0,1\}^{l_1+l_2}$ and returns $h_{3,i}$ to $\mathcal{A}_2$.
- **Decryption queries**: $\mathcal{A}_2$ makes a decryption query on C. If C is equal to the challenge ciphertext C^* except C_3, $\mathcal{B}_2$ returns $\perp$. Otherwise, $\mathcal{B}_2$ responds as in Game 1.

3. Challenge phase: For any m^*, $\mathcal{B}_2$ randomly chooses $r, t \in \mathbb{Z}_q^*$, $\omega_2^* \in \{0,1\}^{l_1+l_2}$ and computes $h = H_2(m^*)$, $S = \prod_{i=0}^{n} S_i^{rh^i}$ and defines the challenge ciphertexts

$$C^* = \{C_1^* = (C_{1,0}^*, C_{1,1}^*, \cdots, C_{1,n}^*), C_2^*, C_3^*, C_4^*\}$$

as follows:

$C_1^* = \{C_{1,0}^* = g^r, C_{1,1}^* = g^{rh}, \cdots, C_{1,n}^* = g^{rh^n}\}$,
$C_2^* = g^t$,
$C_3^* = \omega_2^*$,
$C_4^* = e(h_{ID}, S) \cdot H_4(m^*)$.

4. Update phase: $\mathcal{B}_2$ adds the tuple $(e(h_{ID}, U)^t, \omega_2^* \oplus (m^*||r))$ to the list H_3^{list}.

5. Output phase: $\mathcal{A}_2$ outputs a guess m' for m^*.

Compared to Game 1, the value of C_3^* is replaced by a random value ω_2^* in Game 2. According to the random oracle model, if the event E_1 does not occur, Game 2 is the same as Game 1. Therefore

$$|\Pr[Q_2]| - |\Pr[Q_1]| \leqslant \Pr[E_1]. \tag{3}$$

Now, we proof the event E_1 occurs with negligible probability

$$\Pr[E_1] \leqslant Adv_{P_1}^{CBDH} \cdot q_{H_3} + \frac{q_d \cdot q_{H_3}}{2^{l_1+l_2}}. \tag{4}$$

Claim 1. *Event E_1 occurs with negligible probability $\Pr[E_1]$ in Game 2 if the CBDH problem is intractable.*

Proof. Assume the event E_1 occurs in Game 2 with a non-negligible probability $\Pr[E_1]$, we can construct an algorithm P_1 that can compute $e(g,g)^{xyz}$ with a non-negligible probability when receiving a random CBDH problem instance (g, g^x, g^y, g^z). □

P_1 randomly selects $r, s_0, s_1, \cdots, s_n \in \mathbb{Z}_q^*$, $m^* \in \Delta$, $v_1^* \in \{0,1\}^{l_1+l_2}$ and computes $h = H_2(m^*)$. The system parameters are $\{h_{ID} = g^x, U = g^y, S_0 = g^{s_0}, S_1 = g^{s_1}, \cdots, S_n = g^{s_n}, S = \prod_{i=0}^{n} S_i^{rh^i} = g^{r\sum_{i=0}^{n} s_i h^i}\}$. Then, P_1 calculates $C_1^* = \{C_{1,0}^* = g^r, C_{1,1}^* = g^{rh}, \cdots, C_{1,n}^* = g^{rh^n}\}$, $C_2^* = g^z$, $C_3^* = v_1^*$ and $C_4^* = e(h_{ID}, S) \cdot H_4(m^*)$ as the challenge ciphertexts and adds $(\perp, v_1^* \oplus (m^*||r))$ into the list H_3^{list}. Finally, P_1 returns $PP_{PKE} = \{q, e, \mathbb{G}_1, \mathbb{G}_T, g, U, S_0, S_1, \cdots, S_n, S, h_{ID}, H_2, H_3, H_4\}$ and the challenge ciphertexts $C^* = \{C_1^*, C_2^*, C_3^*, C_4^*\}$ to $\mathcal{A}_2$. $\mathcal{A}_2$ makes the following queries:

- **Hash H_3 queries**: P_1 responds as in Game 2.
- **Decryption queries**: $\mathcal{A}_2$ makes a decryption query on C. If $C_1 = C_1^*, C_2 = C_2^*, C_3 \neq C_3^*, C_4 = C_4^*$, P_1 returns $\perp$. Otherwise, P_1 searches the list H_3^{list} to get $h_{3,i}$ and computes $m^*||r = h_{3,i} \oplus C_3^*$, $h = H_2(m^*)$. If $C_{1,i}^* = g^{rh^i}$ and $C_4^* = \prod_{i=0}^{n} e(h_{ID}, C_{1,i}^*)^{s_i} \cdot H_4(m^*)$ are hold for all $i \in [0,n]$, P_1 returns m^* to $\mathcal{A}_2$.

If the following two cases holds, P_1 can solve the CBDH problem:

1. $\mathcal{A}_2$ has never made a hash H_3 query on $e(h_{ID}, C_2)^y$ before a decryption query on $C = \{C_1, C_2, C_3, C_4\}$. In this case, P_1 returns $\perp$. If C is a valid ciphertext, it means $\mathcal{A}_2$ guesses the value of $h_{3,i}$ correctly. Thus the probability is $\frac{1}{2^{l_1+l_2}}$.
2. The event E_1 occurs in the hash H_3 queries. It means that the list H_3^{list} includes the tuple $(e(h_{ID}, C_2)^y, \perp)$. The probability is $\frac{\Pr[E_1]}{q_{H_3}}$.

Let X_1 to be event that the ciphertext is valid when P_1 returns $\perp$ in the case 1. Then we have

$$\Pr[X_1] \leqslant \frac{q_d}{2^{l_1+l_2}}. \tag{5}$$

Let X_2 to be event in case 2 that P_1 obtains $e(g,g)^{xyz}$ as a solution of the CBDH problem. If X_1 does not occur and $(e(h_{ID}, C_2)^y, \perp)$ appears in the list H_3^{list} with the probability at least $\Pr[E_1]$. So

$$\Pr[X_2 \,|\neg\, X_1\,] = \frac{\Pr[E_1]}{q_{H_3}}. \tag{6}$$

Then

$$\begin{aligned}\Pr[X_2] &= \Pr[X_2\,|X_1\,]\Pr[X_1] + \Pr[X_2\,|\neg X_1\,]\Pr[\neg X_1] \\ &\geqslant \Pr[X_2\,|\neg X_1\,]\Pr[\neg X_1] \\ &= \Pr[X_2\,|\neg X_1\,](1-\Pr[X_1]) \\ &= \Pr[X_2\,|\neg X_1\,] - \Pr[X_2\,|\neg X_1\,]\Pr[X_1] \\ &\geqslant \Pr[X_2\,|\neg X_1\,] - \Pr[X_1] \\ &= \frac{\Pr[X_1]}{q_{H_3}} - \frac{q_d}{2^{l_1+l_2}}.\end{aligned}$$

So, we obtain

$$Adv_{P_1}^{CBDH} \geqslant \tfrac{\Pr[E_1]}{q_{H_3}} - \tfrac{q_d}{2^{l_1+l_2}}. \tag{7}$$

According to the assumption, if $\Pr[E_1]$ is non-negligible, the advantage $Adv_{P_1}^{CBDH}$ is non-negligible. The proof of Claim 1 is completed.

Claim 2. *Event Q_2 occurs with negligible probability $\Pr[Q_2]$ in Game 2 if the CBDH problem is intractable.*

Proof. Assume the event Q_2 occurs in Game 2 with a non-negligible probability $\Pr[Q_2]$, we can construct an algorithm P_2 that can compute $e(g,g)^{xyz}$ with a non-negligible probability when receiving a random CBDH problem instance (g, g^x, g^y, g^z). □

P_2 randomly selects $t, s_1, s_2, \cdots, s_n \in \mathbb{Z}_q^*$, $\nu_1^* \in \{0,1\}^{l_1+l_2}$, $\nu_2^* \in \mathbb{G}_T$, $m^* \in \Delta$ and computes $h = H_2(m^*)$. The system parameters are $\{h_{ID} = g^x, S_0 = g^y, S_1 = g^{s_1}, S_2 = g^{s_2}, \cdots, S_n = g^{s_n}\}$. Then, P_2 calculates $C_1^* = \{C_{1,0}^* = g^z, C_{1,1}^* = g^{zh}, C_{1,2}^* = g^{zh^2}, \cdots, C_{1,n}^* = g^{zh^n}, C_2^* = g^t, C_3^* = \nu_1^*$ and $C_4^* = \nu_2^* \cdot H_4(m^*)$ as the challenge ciphertexts and adds $(\perp, \nu_1^* \oplus (m^*||r))$ into the list H_3^{list}. And P_2 returns $PP_{PKE} = \{q, e, \mathbb{G}_1, \mathbb{G}_T, g, U, S_0, S_1, S_2, \cdots, S_n, h_{ID}, H_2, H_3, H_4\}$ and the challenge ciphertexts $C^* = \{C_1^*, C_2^*, C_3^*, C_4^*\}$ to $\mathcal{A}_2$.

$\mathcal{A}_2$ interacts with P_2 as Game 2.

Finally, P_2 obtains $e(g,g)^{xyz}$ by computing

$$e(h_{ID}, C_{1,0}^*)^y = \frac{C_4^*}{H_4(m^*) \cdot \prod_{i=1}^{n} e(h_{ID}, C_{1,i}^*)^{s_i}}.$$

Therefore, we have

$$\Pr[Q_2] \leqslant Adv_{P_2}^{CBDH}. \tag{8}$$

According to the assumption, if $\Pr[Q_2]$ is non-negligible, the advantage $Adv_{P_2}^{CBDH}$ is non-negligible. The proof of Claim 2 is completed.

Owing to the Equations (1)–(8), we can claim that

$$\begin{aligned}Adv_{PKE,\mathcal{A}_2}^{OW-CCA} &= \Pr[Q_0] \\ &= \Pr[Q_1] \\ &\leqslant \Pr[Q_2] + Adv^{CBDH} \cdot q_{H_3} + \frac{q_{H_3} \cdot q_d}{2^{l_1+l_2}} \\ &\leqslant (q_{H_3}+1) \cdot Adv^{CBDH} + \frac{q_{H_3} \cdot q_d}{2^{l_1+l_2}}.\end{aligned}$$

So, Theorem 3 has been proved.

According to Theorem 2 and Theorem 3, we can show that the proposed IBE-FET scheme satisfies OW-ID-CCA security. Assume an OW-ID-CCA adversary $\mathcal{A}$ is able to against IBE-FET with the

probability ε, then there the algorithm $\mathcal{B}$ can solve the CBDH problem with the probability at least $\varepsilon' = \frac{\varepsilon}{e(q_{dk}+q_{Aut}+q_d+1)(q_{H_3}+1)} - \frac{q_{H_3} \cdot q_d}{(2^{l_1+l_2})(q_{H_3}+1)}$.

7. Comparison and Performance Evaluation

In this section, we present the comparisons between the proposed IBE-FET scheme and the existing related schemes [23–25,30,32,33,39].

7.1. Comparison

The comparison for the proposed IBE-FET scheme and the related schemes [23–25,30,32,33,39] is given in Table 1. Let ET be the quality test, FET be the filtered quality test, ID be the identity-based and ROM be the random oracle model. Let ✓ denote "satisfy" and ✗ denote "not satisfy".

Table 1. Comparison.

Schemes	ET	FET	ID	ROM	Security	Assumption
[23]	✓	✗	✗	✓	OW-CCA	CDH
[24]	✓	✗	✗	✓	OW-CCA,IND-CCA	CDH,DDH
[25]	✓	✗	✗	✓	OW-CCA,IND-CCA	CDH
[30]	✓	✗	✗	✓	OW-CCA,IND-CCA	CONF,CDH
[32]	✓	✗	✓	✓	OW-ID-CCA	CDH
[33]	✓	✗	✓	✓	OW-ID-CCA	CBDH
[39]	✓	✓	✗	✗	IND-CCA	SXDH
The proposed scheme	✓	✓	✓	✓	OW-ID-CCA	CBDH

From Table 1, it is clearly observed that scheme [39] and the proposed scheme support the filtered equality test while other schemes only provide the equality test. Schemes [32,33] and the proposed scheme adopt the identity-based cryptography which can avoid the certificate management problem, while other schemes adopt public key cryptography. With regard to security, all schemes are provably secure based on basic assumptions in the random oracle except scheme [39]. However, none of the schemes [23–25,30,32,33,39] could satisfy both the properties of the filtered equality test and of the identity-based one, only our scheme can do it.

7.2. Computation Cost

For computation complexity estimation, the time cost for performing the cryptographic operations is defined as follows. Let T_E and T_P denote the time of a scale multiplication operation and a bilinear pairing operation, respectively. The time of a map-to-point hash function operation is denoted as T_H. Other lightweight operations (point addition, one way hash function operation) are not taken into account.

To offer the security level of 80-bit, we adopt the symmetric bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$, here $\mathbb{G}_1$ is the cyclic group generated by a generator g with the order q on a super singular elliptic curve $E : y^2 = x^3 + x \bmod p$ with embedding degree 2. p is 512-bit prime number and q is 160-bit Solinas prime number, which satisfy $q \cdot 12 \cdot r = p + 1$. Using the MIRACL Crypto SDK [51], the running time of the cryptographic operations are quantified. The experiment is run on an Intel Core i5-4590, 3.3GHz CPU, 8 gigabytes memory with Windows 7 environment. Table 2 lists the average execution times of cryptographic operations T_E, T_P, and T_H.

Table 2. Execution time of cryptographic operation.

Cryptographic Operation	Execution Time
Scalar multiplication T_E	3.7770
Bilinear pairing T_P	9.0791
Map-to-point hash function T_H	9.7052

Based on the experimental results, the computation cost of the proposed IBE-FET scheme and the related schemes [23–25,30,32,33,39] are summarized in Table 3.

Table 3. Computation costs.

Schemes	Encryption	Decryption	Authorization	Equality Test
[23]	$3T_E$	$3T_E$	$\perp$	$2T_P$
[24]	$4T_E$	$2T_E$	$3T_E$	$4T_P$
[25]	$5T_E$	$2T_E$	$\perp$	$4T_P$
[30]	$6T_E$	$5T_E$	$\perp$	$2T_E + 2T_P$
[32]	$6T_E + 2T_P + 2T_H$	$2T_E + 2T_P + 1T_H$	$1T_E$	$4T_P + 2T_H$
[33]	$2T_E + 1T_H$	$2T_P + 1T_H$	$1T_H$	$2T_E + 4T_P + 2T_H$
[39]	$(n+4)T_E + 1T_H$	$(n+3)T_E + 1T_P + 1T_H$	$(n+1)T_E$	$(n+1)T_P$
The proposed scheme	$(n+3)T_E + 2T_H + 2T_P$	$(n+1)T_E + 1T_H + (n+2)T_P$	$(n+1)T_E$	$(n+1)T_P$

In the encryption phase, the proposed scheme needs to execute $n+3$ scalar multiplication operations, two bilinear pairing operations and two map-to-point hash operations; therefore, the total encryption time is $(n+3)T_E + 2T_P + 2T_H = 3.7770n + 48.8996$ ms. In the decryption phase, the proposed scheme needs to execute $n+1$ scalar multiplication operations, $n+2$ bilinear pairing operations and one map-to-point hash operation; therefore, the total decryption time is $(n+1)T_E + (n+2)T_P + 1T_H = 12.8561n + 31.6404$ ms. In the authorization phase, the proposed scheme needs to execute $n+1$ scalar multiplication operations; therefore, the total authorization time is $(n+1)T_E = 3.7770n + 3.7770$ ms. In the test phase, the proposed scheme needs to execute $n+1$ bilinear pairing operations; therefore, the total test time is $(n+1)T_P = 9.0791n + 9.0791$ ms. From Table 3, we can arrive at the fact that the computational cost of the proposed scheme is higher than those of other schemes [23–25,30,32,33,39] in both encryption and decryption phases. In terms of authorization and test phases, the proposed scheme has the same computational cost as scheme [39], which is more than those of other schemes [23–25,30,32,33,39].

Figure 6 describes the relationship between the computational cost of the proposed scheme and the number of message n. As shown in Figure 6, the total computational cost increases linearly with the number of message in all phases. The computational cost is equal to 67.7496, 95.9209, 22.6270 and 54.4746 ms when $n = 5$, that is equal to 162.2096, 417.3234, 117.0870 and 281.4521 ms when $n = 30$, in encryption, decryption, authorization, and equation test phase of the proposed scheme, respectively. Based on the above analysis, the computational cost of the proposed scheme is feasible.

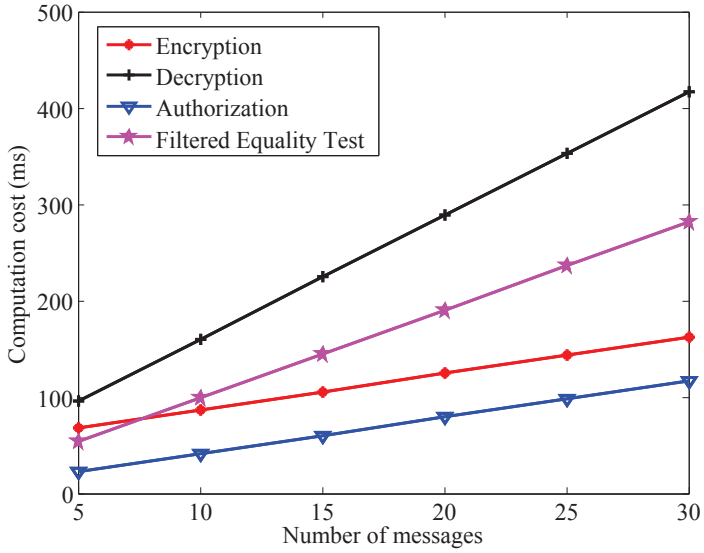

Figure 6. Computational cost with different number of messages.

7.3. Communication Cost

We compare the communication cost of the proposed IBE-FET and those of the related schemes [23–25,30,32,33,39] in this section. The communication cost is represented by the size of message transmitted. The sender transmits the ciphertext to the cloud server for storing and a warrant is transmitted from the receiver to the cloud server in order to perform the filter equality test. Therefore, the communication cost is generated as a result of the communication between the sender and the cloud server and between the receiver and the cloud server. Let $|PK|$, $|CT|$, $|WT|$ denote the sizes of the public key, ciphertext and warrant, respectively. Let $|\mathbb{G}_1|$ be the length of the element in group $\mathbb{G}_1$, $|\mathbb{G}_T|$ be the length of the element in group $\mathbb{G}_T$, $|\mathbb{Z}_q|$ be the element's length of $\mathbb{Z}_q$. Since the size of q is 512 bits (64 bytes), therefore the sizes of the elements in group $\mathbb{G}_1$ and $\mathbb{G}_T$ are 512 bits (64 bytes) and 3072 bits (384 bytes) respectively. The length of $\mathbb{Z}_q$ is 512 bits (64 bytes).

Based on the above analysis, in the proposed scheme, the ciphertext $C = \{C_1 = (C_{1,0}, C_{1,1}, \cdots, C_{1,n}), C_2, C_3, C_4\}$ is sent from the sender to the cloud server, where $C_{1,i} \in \mathbb{G}_1$, $C_2 \in \mathbb{G}_1$, $C_3 \in \mathbb{G}_T$, $C_4 \in \mathbb{Z}_q$. Therefore, the communication cost is $(n+2)|\mathbb{G}_1| + |\mathbb{G}_T| + |\mathbb{Z}_q| = 64n + 576$ *bytes*. The warrant $w_{ID} = \{w_{ID,0}, w_{ID,1}, \cdots, w_{ID,n}\}$ is sent from the receiver to the cloud server, where $w_{ID,i} \in \mathbb{G}_1$. Therefore, the communication cost is $(n+1)|\mathbb{G}_1| = 64n + 64$ *bytes*. The results of the comparison are listed in Table 4.

Table 4. Communication costs.

Schemes	$\|PK\|$	$\|CT\|$	$\|WT\|$
[23]	$1\|\mathbb{G}_1\| = 64$ bytes	$3\|\mathbb{G}_1\| + 1\|\mathbb{Z}_q\| = 256$ bytes	$\perp$
[24]	$2\|\mathbb{G}_1\| = 128$ bytes	$3\|\mathbb{G}_1\| + 1\|\mathbb{Z}_q\| = 256$ bytes	$3\|G_1\| = 192$ bytes
[25]	$2\|\mathbb{G}_1\| = 128$ bytes	$3\|\mathbb{G}_1\| + 1\|\mathbb{Z}_q\| = 256$ bytes	$1\|Z_q\| = 64$ bytes
[30]	$2\|\mathbb{G}_1\| = 128$ bytes	$5\|\mathbb{G}_1\| + 1\|\mathbb{Z}_q\| = 384$ bytes	$\perp$
[32]	$2\|\mathbb{G}_1\| = 128$ bytes	$5\|\mathbb{G}_1\| + 1\|\mathbb{Z}_q\| = 384$ bytes	$1\|\mathbb{G}_1\| = 64$ bytes
[33]	$2\|\mathbb{G}_1\| = 128$ bytes	$2\|\mathbb{G}_1\| + 2\|\mathbb{Z}_q\| = 256$ bytes	$1\|\mathbb{G}_1\| = 64$ bytes
[39]	$(n+2)\|\mathbb{G}_1\| + 1\|\mathbb{G}_T\|$ $= 64n + 512$ bytes	$(n+2)\|\mathbb{G}_1\| + 1\|\mathbb{G}_T\|$ $= 64n + 576$ bytes	$(n+1)\|\mathbb{G}_1\|$ $= 64n + 64$ bytes
The proposed scheme	$(n+2)\|\mathbb{G}_1\|$ $= 64n + 128$ bytes	$(n+2)\|\mathbb{G}_1\| + 1\|\mathbb{G}_T\| + 1\|\mathbb{Z}_q\|$ $= 64n + 576$ bytes	$(n+1)\|\mathbb{G}_1\|$ $= 64n + 64$ bytes

From Table 4, we can see that the communication cost of schemes [23–25,30,32,33,39] is a fixed value, while that of the proposed scheme and scheme [39] increases linearly with the number of message n. From the above analysis, we find that when the message n is constant, the public key's size of the proposed scheme is smaller than those of scheme [39]. As for the size of ciphertext and warrant, the communication cost of the proposed scheme is equal to that of scheme [39]. Thus, the communication cost of the proposed IBE-FET scheme is lower than that of scheme [39].

8. Conclusions

In this paper, based on bilinear pairing and secret sharing, we have presented an identity-based encryption with the filtered equality test (IBE-FET) scheme. The security analysis demonstrated that the proposed IBE-FET is OW-ID-CCA secure under the CBDH assumptions in the random oracle model. The performance evaluation and comparison indicate that the proposed IBE-FET achieves greater functionality than most previous schemes and adopts identity-based cryptography which avoids the certificate management issue effectively. In addition, the total computational cost increases linearly with the number of message n in all phases. Besides, in terms of communication cost, the proposed scheme is efficient. Therefore, the proposed IBE-FET scheme is more practical.

Author Contributions: Y.M. and E.W. conceived of the work, designed the concrete scheme and wrote the paper.

Acknowledgments: This work was supported in part by the Natural Science Foundation of Shaanxi Province under Grant 2018JM6081, in part by the Project of Science and Technology of Xi'an City under Grant

2017088CG/RC051(CADX002), and in part by the Fundamental Research Funds for the Central Universities, CHD, under Grant 300102249204.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nam, T.; Pardo, T.A. Conceptualizing smart city with dimensions of technology, people, and institutions. In Proceedings of the 12th Annual International Digital Government Research Conference on Digital Government Innovation in Challenging Times, College Park, MD, USA, 12–15 June 2011; pp. 282–291.
2. Yu, Y.; Li, Y.; Tian, J. Blockchain-based solutions to security and privacy issues in the Internet of Things. *IEEE Wirel. Commun.* **2018**, *25*, 12–18. [CrossRef]
3. Su, K.; Jie, L.; Hongbo, F. Smart city and the applications. In Proceedings of the International Conference on Electronics, Communications and Control (ICECC), Ningbo, China, 9–11 September 2011; pp. 1028–1031.
4. Ferraz, F.S.; Ferraz, C.A.G. Smart city security issues: Depicting information security issues in the role of an urban environment. In Proceedings of the 7th International Conference on Utility and Cloud Computing (UCC), London, UK, 8–11 December 2014; pp. 842–847.
5. Zheng, D.; Wu, A.; Zhang, Y.; Zhao, Q. Efficient and privacy-preserving medical data sharing in Internet of Things with limited computing power. *IEEE Access* **2018**, *6*, 28019–28027. [CrossRef]
6. Zhang, Y.; Yang, M.; Zheng, D.; Lang, P.; Wu, A.; Chen, C. Efficient and secure big data storage system with leakage resilience in cloud computing. *Soft Comput.* **2018**, *22*, 7763–7772. [CrossRef]
7. Catarinucci, L.; De Donno, D.; Mainetti, L. An IoT-aware architecture for smart healthcare systems. *IEEE Internet Things J.* **2015**, *2*, 515–526. [CrossRef]
8. Demirkan, H. A smart healthcare systems framework. *IT Prof.* **2013**, *15*, 38–45. [CrossRef]
9. Acampora, G.; Cook, D.J.; Rashidi, P. A survey on ambient intelligence in healthcare. *Proc. IEEE* **2013**, *101*, 2470–2494. [CrossRef] [PubMed]
10. Zhang, Y.; Zheng, D.; Deng, R.H. Security and privacy in smart health: Efficient policy-hiding attribute-based access control. *IEEE Internet Things J.* **2018**, *5*, 2130–2145. [CrossRef]
11. Zhang, Y.; Lang, P.; Zheng, D.; Yang, M.; Guo, R. A secure and privacy-aware smart health system with secret key leakage resilience. *Secur. Commun. Netw.* **2018**, *2018*, 1–13. [CrossRef]
12. Zhang, Y.; Deng, R.H.; Han, G. Secure smart health with privacy-aware aggregate authentication and access control in Internet of Things. *J. Netw. Comput. Appl.* **2018**, *123*, 89–100. [CrossRef]
13. Zhang, Y.; Zheng, D.; Guo, R.; Lan, Q. Fine-grained access control systems suitable for resource-constrained users in cloud computing. *Comput. Inf.* **2018**, *37*, 327–348. [CrossRef]
14. Abdalla, M.; Bellare, M.; Catalano, D. Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. In Proceedings of the Advances in Cryptology-Crypto'05, Santa Barbara, CA, USA, 14–18 August 2005; pp. 205–222.
15. Bellare, M.; Boldyreva, A.; O'Neill, A. Deterministic and efficiently searchable encryption. In Proceedings of the Advances in Cryptology-Crypto'07, Santa Barbara, CA, USA, 19–23 August 2007; pp. 535–552.
16. Fuhr, T.; Paillier, P. Decryptable searchable encryption. In Proceedings of the International Conference on Provable Security, Wollongong, Australia, 1–2 November 2007; pp. 228–236.
17. Boneh, D.; Di Crescenzo, G.; Ostrovsky, R. Public key encryption with keyword search. In Proceedings of the Advances in Cryptology-Crypto'04, Interlaken, Switzerland, 2–6 May 2004; pp. 506–522.
18. Yau, W.C.; Heng, S.H.; Goi, B.M. Off-line keyword guessing attacks on recent public key encryption with keyword search schemes. In Proceedings of the International Conference on Autonomic and Trusted Computing (ATC), Oslo, Norway, 23–25 June 2008; pp. 100–105.
19. Ibraimi, L.; Nikova, S.; Hartel, P. Public-key encryption with delegated search. In Proceedings of the International Conference on Applied Cryptography and Network Security (ACNS), Nerja, Spain, 7–10 June 2011; pp. 532–549.
20. Fang, L.; Susilo, W.; Ge, C. Public key encryption with keyword search secure against keyword guessing attacks without random oracle. *Inform. Sci.* **2013**, *238*, 221–241. [CrossRef]
21. Baek, J.; Safavi-Naini, R.; Susilo, W. Public key encryption with keyword search revisited. In Proceedings of the International Conference on Computational Science and Its Applications (ICCSA), Perugia, Italy, 30 June–3 July 2008; pp. 1249–1259.

22. Chen, R.; Mu, Y.; Yang, G. A new general framework for secure public key encryption with keyword search. In Proceedings of the Australasian Conference on Information Security and Privacy (ACISP), Brisbane, QLD, Australia, 29 June–1 July 2015; pp. 59–76.
23. Yang, G.; Tan, C.H.; Huang, Q. Probabilistic public key encryption with equality test. In Proceedings of the Cryptographers' Track at the RSA Conference (CT-RSA), San Francisco, CA, USA, 1–5 March 2010; pp. 119–131.
24. Tang, Q. Towards public key encryption scheme supporting equality test with fine-grained authorization. In Proceedings of the Australasian Conference on Information Security and Privacy (ACISP), Melbourne, VIC, Australia, 11–13 July 2011; pp. 389–406.
25. Tang, Q. Public key encryption supporting plaintext equality test and user-specified authorization. *Secur. Commun. Netw.* **2012**, *5*, 1351–1362. [CrossRef]
26. Tang, Q. Public key encryption schemes supporting equality test with authorization of different granularity. *Int. J. Appl. Cryptogr.* **2012**, *2*, 304–321. [CrossRef]
27. Lu, Y.; Zhang, R.; Lin, D. Stronger security model for public-key encryption with equality test. In Proceedings of the International Conference on Pairing-Based Cryptography, Cologne, Germany, 16–18 May 2012; pp. 65–82.
28. Ma, S.; Zhang, M.; Huang, Q. Public key encryption with delegated equality test in a multi-user setting. *Comput. J.* **2014**, *58*, 986–1002. [CrossRef]
29. Huang, K.; Tso, R.; Chen, Y.C. Pke-aet: Public key encryption with authorized equality test. *Comput. J.* **2015**, *58*, 2686–2697. [CrossRef]
30. Ma, S.; Huang, Q.; Zhang, M. Efficient public key encryption with equality test supporting flexible authorization. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 458–470. [CrossRef]
31. Lin, X.J.; Qu, H.; Zhang, X. Public Key Encryption Supporting Equality Test and Flexible Authorization Without Bilinear Pairings. Cryptology ePrint Archive. 2016. Available online: http://eprint.iacr.org/2016/277 (accessed on 1 July 2019).
32. Ma, S. Identity-based encryption with outsourced equality test in cloud computing. *Inf. Sci.* **2016**, *328*, 389–402. [CrossRef]
33. Wu, L.; Zhang, Y.; Choo, K.R. Efficient and secure identity-based encryption scheme with equality test in cloud computing. *Future Gener. Comput. Syst.* **2017**, *73*, 22–31. [CrossRef]
34. Zhu, H.; Wang, L.; Ahmad, H.; Niu, X. Key-policy attribute-based encryption with equality test in cloud computing. *IEEE Access* **2017**, *5*, 20428–20439. [CrossRef]
35. Wang, Q.; Peng, L.; Xiong, H.; Sun, J. Ciphertext-policy attribute-based encryption with delegated equality test in cloud computing. *IEEE Access* **2017**, *6*, 760–771. [CrossRef]
36. Liao, Y.; Chen, H.; Li, F.; Jiang, S.; Zhou, S.; Mohammed, R. Insecurity of a key-policy attribute based encryption scheme with equality test. *IEEE Access* **2018**, *6*, 10189–10196. [CrossRef]
37. Sun, J.; Bao, Y.; Nie, X.; Xiong, H. Attribute-hiding predicate encryption with equality test in cloud computing. *IEEE Access* **2018**, *6*, 31621–31629. [CrossRef]
38. Huang, K.; Chen, Y.C.; Tso, R. Semantic secure public key encryption with filtered equality test pke-fet. In Proceedings of the 12th International Joint Conference on E-Business and Telecommunications (ICETE), Colmar, France, 20–22 July 2015; pp. 327–334.
39. Huang, K.; Tso, R.; Chen, Y.C. Somewhat semantic secure public key encryption with filtered-equality-test in the standard model and its extension to searchable encryption. *J. Comput. Syst. Sci.* **2017**, *89*, 400–409. [CrossRef]
40. Boneh, D.; Franklin, M. Identity-based encryption from the Weil pairing. In Proceedings of the Advances in Cryptology-Crypto'01, Santa Barbara, CA, USA, 19–23 August 2001; pp. 213–229.
41. Amin, S.M.; Wollenberg, B.F. Toward a smart grid: Power delivery for the 21st century. *IEEE Power Energy Mag.* **2005**, *3*, 34–41. [CrossRef]
42. Heydt, G.T. The next generation of power distribution systems. *IEEE Trans. Smart Grid.* **2010**, *1*, 225–235. [CrossRef]
43. Alderman, J.; Farley, N.; Crampton, J. Tree-based cryptographic access control. In Proceedings of the 22nd European Symposium on Research in Computer Security (ESORICS), Oslo, Norway, 11–15 September 2017; pp. 47–64.

44. Alderman, J.; Crampton, J.; Farley, N. A framework for the cryptographic enforcement of information flow policies. In Proceedings of the 22nd ACM on Symposium on Access Control Models and Technologies (SACMAT), Indianapolis, IN, USA, 21–23 June 2017; pp. 143–154.
45. Castiglione, A.; De Santis, A.; Masucci, B. Supporting dynamic updates in storage clouds with the Akl-Taylor scheme. *Inf. Sci.* **2017**, *387*, 56–74. [CrossRef]
46. Castiglione, A.; De Santis, A.; Masucci, B. Key indistinguishability versus strong key indistinguishability for hierarchical key assignment schemes. *IEEE Trans. Dependable Secur. Comput.* **2016**, *13*, 451–460. [CrossRef]
47. Yu, Y.; Ho Au, M.; Ateniese, G.; Huang, X.; Susilo, W.; Dai, Y.; Min, G. Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage. *IEEE Trans. Inf. Forensics Sec.* **2017**, *12*, 767–778. [CrossRef]
48. Li, Y.; Yu, Y.; Susilo, W.; Min, G.; Ni, J.; Choo, R. Fuzzy identity-based data integrity auditing for reliable cloud storage systems. *IEEE Trans. Dependable Secur. Comput.* **2019**, 16, 72–83. [CrossRef]
49. Shamir, A. How to share a secret. *Commun. ACM* **1979**, *22*, 612–613. [CrossRef]
50. Coron, J.S. On the exact security of full domain hash. In Proceedings of the Advances in Cryptology-Crypto'00, Santa Barbara, CA, USA, 20–24 August 2000; pp. 229–235.
51. Ltd S.S. Multi Precision Integer and Rational Arithmetic Cryptographic Library (MIRACL). 2019. Available online: http://www.certivox.com/miracl/ (accessed on 1 July 2019).