IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 24, 2019, accepted May 12, 2019, date of publication May 16, 2019, date of current version June 7, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2917326

Efficient Certificateless Public Key Cryptography With Equality Test for Internet of Vehicles

RASHAD ELHABOB, YANAN ZHAO, IVA SELLA, AND HU XIONG
School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China

Corresponding author: Hu Xiong (xionghu.uestc@gmail.com)

This work was supported in part by the 13th Five-Year Plan of National Cryptography Development Fund for Cryptographic Theory of China under Grant MMJJ20170204, in part by the Fundamental Research Funds for the Central Universities under Grant ZYGX2016J091, the Guangxi Colleges and Universities Key Laboratory of Cloud Computing and Complex Systems, and in part by the Natural Science Foundation of China under Grant U1401257, Grant 61472064, and Grant 61602096.

ABSTRACT The fast progression of the Internet of Vehicles (IoV) has resulted in a large number of vehicles connecting to networks. This leads to massive growth in the data collected from vehicles via IoV. Fortunately, cloud computing provides a vast range of services such as operating systems, hardware, software, and resources. Therefore, the massive amount of data cumulated through IoV can be outsourced to the cloud. However, considering the untrusted nature of the cloud, the cumulated data must be encrypted before it is outsourced to the cloud server. Unfortunately, this ensures to difficulty while searching the data. To address this challenge, an efficient certificateless public key cryptography with equality test (CL-PKC-ET) is presented in this paper. In this scheme, the authorized cloud server has the permission to execute the equality test on encrypted data and retrieve the result without knowing any relevant information about the ciphertext. Our CL-PKC-ET scheme is demonstrated under the Bilinear Diffie-Hellman assumption in the random oracle model. Ultimately, we compare the CL-PKE-ET with a state-of-art scheme and the performance evaluation indicates that our scheme accomplishes 96.40%, 32.08%, and 43.98% reduction in computation costs during the encryption, decryption, and test stages, respectively. Therefore, we assert that our scheme is ideal for deployment in both the cloud and IoV environments.

INDEX TERMS Internet of Vehicles (IoV), cloud, certificateless, equality test.

I. INTRODUCTION

The paramount growth of Internet of Things (IoT) technologies such as wireless sensor networks, purpose to define the future for humans. The interconnected devices in IoT [2] allow for information flow by making use of the wireless network technology. One of the major areas that IoT has revolutionized is the smart-transport sector which has seen the mobilization of the Internet of Vehicles (IoV) [9]. The IoV can be viewed as an extension of IoT as it achieves the consolidation of management in the transportation area [12]. The IoV is a network that comprises vehicles that are IoT-enabled through the integration of information within the network. Such information comprises of the location of vehicles, their momentum, and even the routes they use. Such information is collected by the use of sensors and devices located within the vehicles [13].

The associate editor coordinating the review of this manuscript and approving it for publication was SK Hafizul Islam.

As communication and computation technologies continue to evolve, growth in the number of vehicles being linked to the IoT has turned the IoV into an interactive network of importance. Vehicles can therefore communicate with each other in an instant. This instant communication means vehicles can receive notifications from their sensors on things like braking, lane changing or even negotiating corners. With such notifications, there is efficiency in managing heavy traffic and the conveyance of passengers is smooth and safe [22]. The continuous generation of notifications from various sensors means a large amount of data accrued from various places and is of different attributes. All this large amounts of data and information include private details like the real-time location of a vehicle and data associated with the driving state information and traffic safety [34]. The growth in vehicles connecting to the Internet, has resulted in the collection of vast amounts of data across the vehicle and application platforms. However, given the nodes restrained resources in IoV [12], cloud computing [30] is the perfect solution, as it is able to

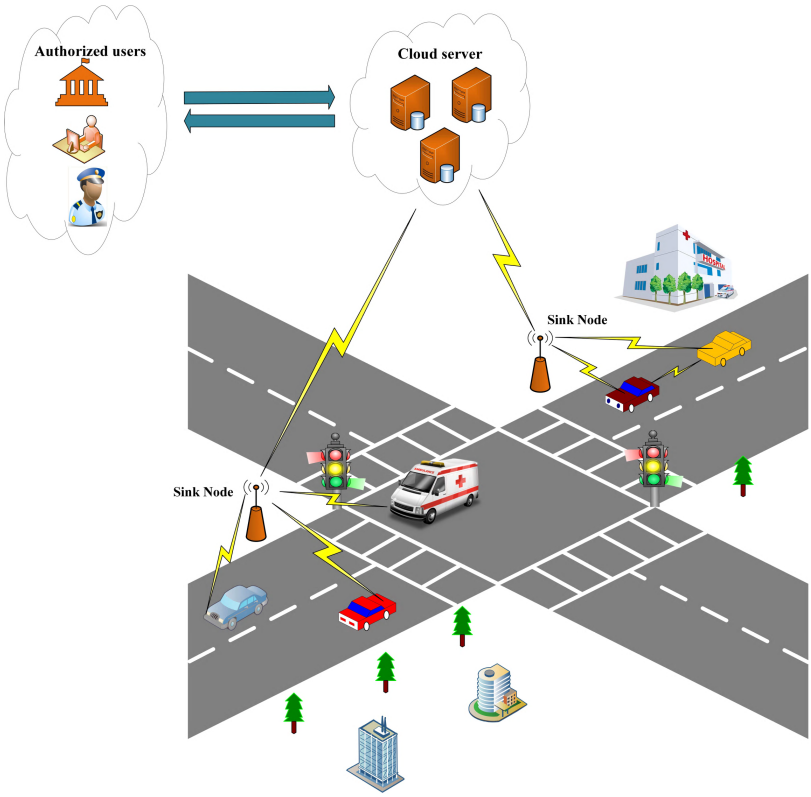


FIGURE 1. A typical scenario for IoV.

outsource almost unlimited storage resources and resource processing. Therefore, the IoV nodes can outsource the large amounts of data to the cloud server. However, the process of outsourcing this data to the cloud servers leads to users losing control over their data. This is because the data could be changed, erased, modified or rather completely damaged beyond recovery. For that reason, it is imperative to see to it that the reliability, integrity, and confidentiality of the data is maintained [8]. To ensure that these qualities are upheld, it is important to encrypt the data before it is outsourced to the cloud server. Consequently, the typical scenario of IoV is showed in FIGURE 1. In this scenario, the aim is to ensure the data cumulated by vehicle nodes which is relayed to the cloud server by sink nodes, is easily accessible from anywhere. The collected data is encrypted by the vehicle nodes before outsourcing to the cloud server. Hence, only the authorized user can access the data from anywhere. For further clarification, we consider a scenario where a police traffic department wants to know all the vehicles that are in a particular location at a given time. Knowing that, all the data stored in the cloud server for these vehicles is encrypted. Thus, the police traffic department must send a query to the cloud server that includes the specific time and location, which in turn will search the stored data and then return the information of vehicles that were present in the specified time and location. Finally, the police traffic department can decrypt the search result received from the cloud server because this information is encrypted using the public key of the police traffic department.

Encryption also ensures that the cloud service providers involved are not accessing users' data without consent or accidentally exposing their data. Later on, the authorized users may want to access this encrypted data and information and therefore they have to search through it. This whole process would definitely incur high costs during computation and communication. It would also consume a lot of time and when considered in real-time application it is unrealistic. Boneh *et al.* [4] proposed a notion of public key encryption with keyword search (PKE-KS) to assist users in retrieving their information effectively. However, for this PKE-KS scheme, the cloud server only compares keywords and trapdoors that have been encrypted with the same public keys. Unfortunately, due to the IoV possessing a heterogeneous nature, this scheme becomes unsuitable for searching the cloud. Therefore, to deal with this problem, Yang *et al.* [33] proposed a notion of public key encryption with equality test (PKE-ET). Here, the search functionality can be performed between a pair of ciphertexts that are encrypted with the same public key and also with different public keys.

The PKE-ET scheme managed to achieve the search functionality requirements within the cloud server. However, this scheme is still constructed on the basis of the public key infrastructure (PKI). This means each user's public key must

be authenticated by a certificate, specifically a digital signature that is generated by a certificate authority (CA). Ma [18] further proposed a notion of identity-based encryption with equality test (IBE-ET). This scheme is constructed from the identity-based cryptosystem (IBC). In this scheme, the certificates that were found in the PKI system were eliminated as a result of setting the user's identity as their public key. Nonetheless, this scheme faces difficulties prompted by the key escrow. This happens when decryption keys are allocated in the care of third parties known as private key generators (PKG). In this case, the PKG can access the users' encrypted data at any given time. Al-Riyami *et al*. [1] therefore presented a notion of the certificateless public key cryptosystem (CL-PKC), where users would own private keys that are in two parts. The first part is created by the key generation center (KGC), while a second part is created by the user. Here, the KGC only has partial access to the private key of a user to prevent it from accessing the user's data. Qu *et al*. [23] further presented a new scheme where he incorporated the notion of PKE-ET and CL-PKC which is known as certificateless public key cryptosystem supporting equivalence test (CL-PKC-ET). However, Qu *et al*.'s scheme is inefficient, especially in the IoV environment relative to the limited resources in the sensor nodes, which encrypt data and transmit it to the cloud server.

A. CONTRIBUTION

To deal with the problems mentioned above, our paper presents an efficient certificateless public key cryptography with equality test (CL-PKC-ET) for the IoV environment. We summarize this paper's contributions as follows.

1) By incorporating the notion of PKE-ET and CL-PKC, the CL-PKC-ET scheme is proposed in this paper. In this scheme, the authorized cloud server is allowed to execute the equality test on encrypted data and then retrieves the result without any knowledge of the relevant information about the ciphertext.
2) We also propose the concrete construction of the CL-PKC-ET scheme. We further prove the security of our suggested scheme under the Bilinear Diffie-Hellman assumption in the random oracle model.
3) Eventually, our CL-PKC-ET is compared with the state-of-art scheme. The performance evaluation demonstrates that our suggested scheme is ideal in the IoV environment.

We organize the rest of this paper as follows. The definition of CL-PKC-ET such as preliminaries, the system model, CL-PKC-ET's framework as well as the security model are illustrated in Section II. The concrete construction is illustrated in Section III, while the security analysis of our proposed scheme is discussed in Section IV. In Section V, the performance analysis of our suggested scheme is presented. Eventually, the conclusion of our paper is given in Section VI.

B. RELATED WORKS

1) BACKGROUND OF IoV

The fact that more vehicles are connecting to IoV networks everyday, means there is a huge amount of data that is collected through the nodes of these vehicles. Therefore, the security and privacy of this huge amounts of data is important. Mershad and Artail [21] suggested a security scheme for the information and messages interchanged between the users and roadside units. Unfortunately, IoV was not quite scalable hence scalability was still an issue. Wu *et al*. [29] proposed a system which would efficiently balance the public safety and vehicle privacy that ensures the reliability of messages. Wang *et al*. [27] however presented a mechanism that would guarantee security for the privacy-preserving communication with convenient cryptographic primitives within the vehicle-to-grid networks. Additionally, Cardenas *et al*. [5] and Xu *et al*. [31] developed a mechanism for the security and privacy of the big data sector. Furthermore, Liu *et al*. [16] suggested a scheme based on key exchange to ensure the scheduling of big data is secure. Moreover, Li *et al*. in [15] and [14] proposed a notion that focused on the security models that would be able to resolve the issue on authentication and privacy issues in corresponding areas. Recently, Guo *et al*. [10] proposed a secure mechanism for big data collection in a large scale IoV scheme. In Guo *et al*.'s scheme, there is an improvement in the security and privacy of data collected in IoV. However, all the above-mentioned schemes have been concerned with methods of data collection, storage confidentially and privacy but have not addressed how we can search on the data cumulated in the cloud server.

2) PKE-ET

To address this issue in the IoV era, public key encryption with equality test (PKE-ET) [33] was first proposed by Yang *et al*. The PKE-ET scheme was designed to provide a platform where any user could compare a pair of ciphertexts and examine whether the encryptions are of the same message, even though the ciphertexts are generated by different public keys. However, Tang proposed a fine-grained authorization on the public key encryption with equality test in order to reduce the vulnerabilities within PKE-ET. This scheme was referred to as fine-grained authorization public key encryption with equality test (FG-PKE-ET) [24]. In this scheme, two users have the right to run an authorization algorithm to generate a token that would be given to a semi-trusted proxy. The proxy would then execute an equality test between the users' ciphertexts. Moreover, Tang further introduced a two-proxy setting scheme [25], where the two proxies function together to perform the equality test. In addition, Tang proposed a more refined scheme referred to as all-or-nothing PKE-ET (AON-PKE-ET) [26]. In this scheme, the equality test is performed in a coarser granularity manner. This means when a proxy receives a token, it can execute an equivalence test between a given user and any other user. In addition, Ma *et al*. proposed a public key encryption

scheme with delegated equality test (PKE-DET) [20], where only a delegated party is able to perform an equivalence test in a practical multi-user setting. This means the designated server can test if the ciphertexts contain the same message even without decrypting them. However, a non-delegated server should not be allowed to deduce any substantial information from the encrypted data while the equality test is being performed. Huang *et al*. further proposed a scheme that authorized the equality test performed known as public key encryption with authorized equality test (PKE-AET) [11]. In this scheme, users have the mandate to perform an equivalence test between two ciphertexts or two specified ciphertexts. Ma *et al*. further improved the PKE-AET by suggesting a scheme that provided flexibility in authorization referred to as public key encryption with equality test incorporating flexible authorization (PKE-ET-FA) [19]. In this scheme, four authorization policies are simultaneously implemented in order to enhance privacy. Besides, the venture into the 5G networks in relation to the PKE-ET, propagated *Xu et al*. to propose a scheme that supports three types of authorization that occur simultaneously [32]. In addition, the scheme verifies the results it receives from an untrusted server, and the user can further evaluate whether the cloud has correctly performed the authorized equality test. In spite of that, the schemes mentioned above are still constructed on the basis of public key infrastructure (PKI). This means each user requires their public key authenticated by a certificate, specifically a digital signature that is generated by the CA.

3) IBE-ET

To deal with issues concerning the management of certificates, and some problems related with them, Ma proposed an identity-based encryption with equality Test (IBE-ET) [18]. The IBE-ET scheme incorporated attributes of both public key encryption and identity-based encryption schemes. Recently, Wu *et al*. [28] presented an efficient IBE-ET scheme, which decreases the need for time-consuming HashToPoint function in Ma's scheme. Nonetheless, the IBE-ET schemes face difficulties that are brought about by the key escrow problem.

4) CL-PKC-ET

To solve the key escrow problem, Qu *et al*. [23] presented a new scheme where he incorporated the notion of PKE-ET and CL-PKC, referred to as certificateless public key cryptography with equality test (CL-PKC-ET). However, Qu *et al*.'s scheme is inefficient, especially in the IoV environment relative to the limited resources in the sensor nodes, which encrypt data and transmit it to the cloud server.

II. DEFINITIONS

A. PRELIMINARIES

1) BILINEAR MAP

Let $\mathbb{G}_1$ be a cyclic additive group and $\mathbb{G}_2$ be a cyclic multiplicative group with the same prime order q. Let P be a generator of $\mathbb{G}_1$. A bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ should satisfy the following conditions:

1) Bilinearity: For any two random points $X, Y \in_R \mathbb{G}_1$ and any two random numbers $a, b \in_R \mathbb{Z}_p^*$, $e(X^a, Y^b) = e(X, Y)^{ab}$.
2) Non-degeneracy: If P is a generator of group $\mathbb{G}_1$, $e(P, P) \neq 1_{\mathbb{G}_2}$.
3) Computability: For any two random points $X, Y \in_R \mathbb{G}_1$, an efficient algorithm is required to compute $e(X, Y)$.

2) MODIFIED BILINEAR DIFFIE-HELLMAN INVERSION (mBDHI) PROBLEM

Given two groups $\mathbb{G}_1$, $\mathbb{G}_2$ with the same prime order q. Let $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be an admissible bilinear map and let P be a generator of $\mathbb{G}_1$. The mBDHI problem in $< \mathbb{G}_1, \mathbb{G}_2, e >$ is to compute $e(P, P)^{\frac{1}{\alpha+\mu}}$ given $< P, \alpha P, \mu >$, where $\alpha, \mu \in_R \mathbb{Z}_p^*$ [6].

B. SYSTEM MODEL

FIGURE 2 illustrates the CL-PKC-ET's system model. The units of this system work as follows.

1) *Registration:* In this phase, the IoV nodes and the authorized users send their identities to the key generation center (KGC). The KGC then returns the partial private key to the IoV nodes and authorized users. Then, the IoV node and authorized user can generate the full-private key.
2) *Setup:* In this phase, the IoV node encrypts the collected data and outsources it to the cloud server. Additionally, the IoV node generates the trapdoor by using its private key and conveys it to the cloud server. In addition, the authorized user uses his/her secret key to create trapdoor and sends it to the cloud.
3) *Query:* In this phase, when an authorized user needs to return the data stored on the cloud server, he/she sends a query (keywords) to the cloud server. The keywords in the query depend on what the authorized user needs. For example, if the authorized user wants to retrieve encrypted data related to a specific vehicle, the query must contain the vehicle ID and other keywords related to the query such as location, time, etc.
4) *Search:* In this phase, the cloud server is delegated to perform the equality test after receiving the trapdoor from the IoV nodes and authorized users. The cloud server then conveys the result to the authorized users. Note that, only the authorized user can perform the decryption of the result.

C. FRAMEWORK OF CL-PKC-ET

A certificateless public key cryptography scheme that entails equality test is described using the following algorithms:

1) **Setup:** The **KGC** runs this algorithm. It uses k as input for the security parameter and outputs the public parameters params and a master secret key msk.

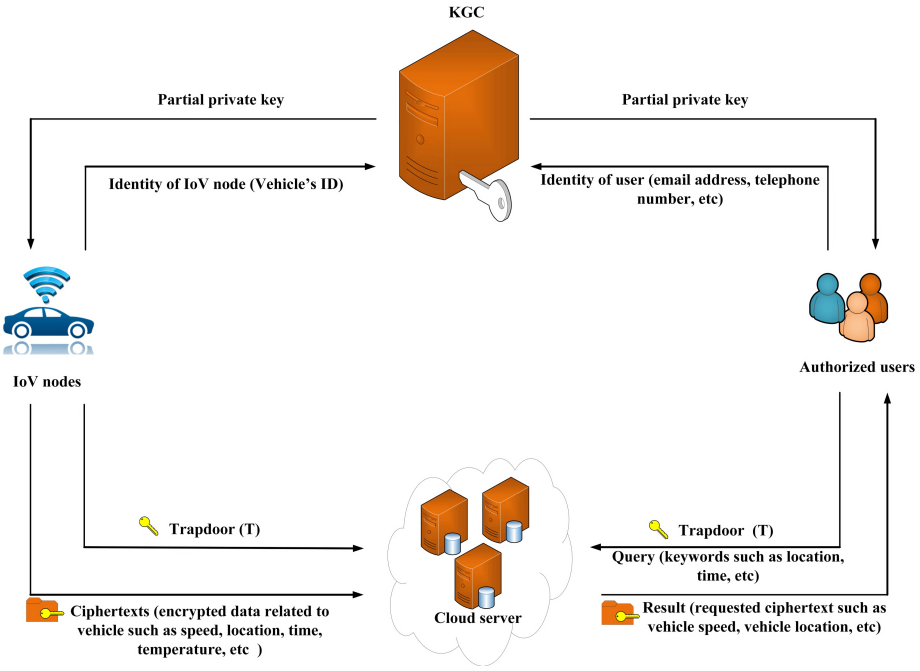


FIGURE 2. System model of CL-PKC-ET.

2) **Partial-Private-Key-Extraction:** The **KGC** also runs this algorithm. It uses as input params, msk, and a public identity of a user $ID \in \{0, 1\}^*$. It returns a partial secret key D_{ID}.
3) **User-Key-Generation:** The user runs this algorithm. It uses as inputs params and ID. Then, it returns the user's secret value x_{ID} and a public key PK_{ID}.
4) **Private-Key-Generation:** The user runs this algorithm. It uses as inputs params, D_{ID}, and x_{ID}. It returns a complete private key S.
5) **Trapdoor:** The user runs this algorithm. It uses as input S_{ID} , and returns a trapdoor T_{ID}.
6) **Encrypt:** This algorithm uses inputs params, a message M, and public key PK_{ID} as inputs. It returns a ciphertext C.
7) **Decrypt:** This algorithm uses params, a ciphertext C, and private key S_{ID} as inputs. It returns the plaintext M. Otherwise, it returns the symbol $\perp$.
8) **Test:** The cloud server runs this algorithm. It uses as inputs a ciphertext C_A and a trapdoor $T_{ID,A}$ of the User A. Furthermore, it uses as inputs a ciphertext C_B and a trapdoor $T_{ID,B}$ of the User B. Then, the **Test** algorithm returns 1 if $C_A = C_B$. Otherwise, it returns 0.

D. SECURITY MODEL

According to [23], for the security of CL-PKC-ET four types of adversaries are considered. The four types of adversaries are described as follows:

1) **Type-1 adversary:** This adversary $\mathcal{A}_1$ cannot acquire the master secret key. However, $\mathcal{A}_1$ can solicit and replace public keys with values it chooses by itself. With this kind of adversary, one-way chosen ciphertext attack (OW-CCA) security for CL-PKC-ET is defined.
2) **Type-2 adversary:** This adversary $\mathcal{A}_2$ can acquire the master secret key. However, $\mathcal{A}_2$ cannot replace public keys of any users. With this type of adversary, OW-CCA security for CL-PKC-ET is defined.
3) **Type-3 adversary:** This adversary $\mathcal{A}_3$ cannot acquire the master secret key. However, $\mathcal{A}_3$ may request and change public keys with values of its choosing. With this kind of adversary, indistinguishability chosen ciphertext attack (IND-CCA) security for CL-PKC-ET is defined.
4) **Type-4 adversary:** This adversary $\mathcal{A}_4$ can acquire the master secret key. However, $\mathcal{A}_4$ cannot replace public keys of any users. With this kind of adversary, IND-CCA security for CL-PKC-ET is considered.

1) THE SECURITY OF OW-CCA FOR TYPE-1 ADVERSARY

The OW-CCA security against **Type-1 adversary** in CL-PKC-ET is shown as follows.

Game 1: Given that $\mathcal{A}_1$ is a **Type-1 adversary**. The game between $\mathcal{A}_1$ and the challenger is depicted as follows:

1) **Setup:** This algorithm is run by the challenger to generate the public parameters params and master secret key msk. Afterwards, the challenger gives params to $\mathcal{A}_1$ and keeps msk secret. Furthermore, the challenger runs the **Partial-Private-Key-Extraction**, **User-Key-Generation**, and **Set-Private-Key** algorithms to create n private keys S_i associated with ID_i, where $(1 \leq i \leq n)$. Finally, the challenger gives all ID_i to $\mathcal{A}_1$.

2) **Phase 1:** $\mathcal{A}_1$ can issue the following queries.
   - **Hash queries:** $\mathcal{A}_1$ requests all hash random oracles, then the challenger responds to $\mathcal{A}_1$ with the corresponding value.
   - **Partial-Private-Key-Extraction-queries** $\langle ID_i \rangle$**:** The challenger runs **Partial-Private-Key-Extraction** algorithm to generate D_i and then sends it to $\mathcal{A}_1$.
   - **Set-Private-key-queries** $\langle ID_i \rangle$**:** The challenger runs **Set-Private-Key** algorithm to generate S_i and then sends it to $\mathcal{A}_1$.
   - **Public-key-queries** $\langle ID_i \rangle$**:** The **User-Key-Generation algorithm is run by the challenger to generate** PK_{ID_i} **and then convey** it to $\mathcal{A}_1$.
   - **Replace-public-key-queries** $\left\langle ID_i, PK'_{ID_i} \right\rangle$**:** The challenger replaces the public key PK_{ID_i} of the corresponding user with PK'_{ID_i}.
   - **Decryption-queries** $\langle ID_i, C_i \rangle$**:** The challenger executes the algorithm **Decrypt**(C_i, S_i), where S_i is the secret key associated with ID_i. Eventually, the challenger gives M_i to $\mathcal{A}_1$.
   - **Trapdoor-queries:** The challenger runs the **Trapdoor** algorithm to generate T_{ID_i} and then conveys it to $\mathcal{A}_1$.
3) **Challenge:** The challenger randomly picks the plaintext $M \in \{0,1\}^*$ and computes $C^* =$ **Encrypt**(ID_{ch}, M). Eventually, the challenger gives C^* to $\mathcal{A}_1$ as its challenge ciphertext.
4) **Phase 2:** The challenger responds to $\mathcal{A}_1$ in the same way as in **Phase 1** on the grounds that:
   - ID_{ch} is not queried in the **Set-Private-key-queries**.
   - If the public key associated with ID_{ch} is replaced, the ID_{ch} should not be queried in the **Partial-Private-Key-Extraction-queries**.
   - If the public key of the user is replaced, the corresponding identity ID_i should not be queried in the **Set-Private-key-queries**.
   - (ID_{ch}, C^*) is not queried in the **Decryption-queries**.
5) **Guess:** $\mathcal{A}_1$ outputs M', and wins if $M' = M$. The advantage of $\mathcal{A}_1$ in the game above is defined as follows:

$$Adv^{OW-CCA}_{CL-PKC-ET,\mathcal{A}_1}(k) = Pr[M' = M].$$

Definition 1: The CL-PKC-ET scheme is OW-CCA secure if for all OW-CCA adversary $\mathcal{A}_1$,its advantage $Adv^{OW-CCA}_{CL-PKC-ET,\mathcal{A}_1}(k)$ is negligible given the security parameter k.

2) THE SECURITY OF OW-CCA FOR TYPE-2 ADVERSARY

The OW-CCA security against **Type-2 adversary** in CL-PKC-ET is shown as follows:

Game 2: Given that $\mathcal{A}_2$ is a **Type-2 adversary**. The game between $\mathcal{A}_2$ and the challenger is illustrated as follows:

1) **Setup:** This algorithm is run by the challenger to generate the public parameters params and master secret key msk. Afterwards, the challenger sends params and the msk to $\mathcal{A}_2$. Furthermore, the challenger runs the **Partial-Private-Key-Extraction**, **User-Key-Generation**, and **Set-Private-Key** algorithms to create n private keys S_i associated with ID_i, where $(1 \leq i \leq n)$. Finally, the challenger gives all ID_i to $\mathcal{A}_2$.
2) **Phase 1:** $\mathcal{A}_2$ issues queries as in **Game 1**, except the **Partial-Private-Key-Extraction-queries** and the **Replace-public-key-queries** are not allowed to issue in this game.
3) **Challenge:** The challenger randomly picks the plaintext $M \in \{0,1\}^*$ and computes $C^* =$ **Encrypt**(ID_{ch}, M). Eventually, the challenger gives C^* to $\mathcal{A}_2$ as its challenge ciphertext.
4) **Phase 2:** The challenger responds to $\mathcal{A}_2$ in the same way as in **Phase 1** on grounds that:
   - ID_{ch} is not queried in the **Set-Private-key-queries**.
   - (ID_{ch}, C^*) is not queried in the **Decryption-queries**.
5) **Guess:** $\mathcal{A}_2$ outputs M', and wins if $M' = M$. Therefore, $\mathcal{A}_2$'s advantage in the game is:

$$Adv^{OW-CCA}_{CL-PKC-ET,\mathcal{A}_2}(k) = Pr[M' = M].$$

Definition 2: The CL-PKC-ET scheme is OW-CCA secure if for all OW-CCA adversary $\mathcal{A}_2$, its advantage $Adv^{OW-CCA}_{CL-PKC-ET,\mathcal{A}_2}(\lambda)$ is negligible given the security parameter k.

3) THE SECURITY OF IND-CCA FOR TYPE-3 ADVERSARY

The IND-CCA security against **Type-3 adversary** in CL-PKC-ET is shown as follows:

Game 3: Given that $\mathcal{A}_3$ is a **Type-3 adversary**. The game between $\mathcal{A}_3$ and the challenger is depicted as follows:

1) **Setup:** This algorithm is run by the challenger to generate the public parameters params and master secret key msk. Afterwards, the challenger gives params to $\mathcal{A}_3$ and keeps msk secret. Furthermore, the challenger runs the **Partial-Private-Key-Extraction**, **User-Key-Generation**, and **Set-Private-Key** algorithms to create n private keys S_i associated with ID_i, where $(1 \leq i \leq n)$. Finally, the challenger gives all ID_i to $\mathcal{A}_3$.
2) **Phase 1:** $\mathcal{A}_3$ issues queries as in **Game 1**.
3) **Challenge:** $\mathcal{A}_3$ gives challenger two plaintexts of equal-length $M_0, M_1 \in \{0,1\}^*$. The challenger then randomly selects a bit $\xi \in \{0,1\}$ and computes $C_\xi =$ **Encrypt**(M_ξ, ID_{ch}). The challenger then sends C_ξ to $\mathcal{A}_3$ as the challenge ciphertext.
4) **Phase 2:** The challenger responds to $\mathcal{A}_3$ in a similar way as in **Phase 1** on grounds that:
   - ID_{ch} is not queried in the **Set-Private-key-queries**.
   - If the public key associated with ID_{ch} is replaced, the ID_{ch} is not queried in the **Partial-Private-Key-Extraction-queries**.

- If the public key of the user is replaced, the corresponding identity ID_i should not be queried in the **Set-Private-key-queries**.
- (ID_{ch}, C^*) should not be queried in the **Decryption-queries**.
- ID_{ch} should not be queried in the **Trapdoor-queries**.

5) **Guess:** $\mathcal{A}_3$ outputs $\xi' \in \{0, 1\}$, and wins if $\xi' = \xi$. The advantage of $\mathcal{A}_3$ in the game above is depicted as follows:

$$Adv^{IND-CCA}_{CL-PKC-ET,\mathcal{A}_3}(k) = |Pr[\xi' = \xi] - \frac{1}{2}|.$$

Definition 3: The CL-PKC-ET scheme is IND-CCA secure if for all IND-CCA adversary $\mathcal{A}_3$, its advantage $Adv^{OW-CCA}_{CL-PKC-ET,\mathcal{A}_3}(k)$ is negligible given the security parameter k.

4) THE SECURITY OF IND-CCA FOR TYPE-4 ADVERSARY

The IND-CCA security against **Type-4 adversary** in CL-PKC-ET is shown as follows:

Game 4: Suppose that $\mathcal{A}_4$ be a **Type-4 adversary**. The game between $\mathcal{A}_4$ and the challenger is depicted as follows:

1) **Setup:** This algorithm is run by the challenger to generate the public parameters params and master secret key msk. Afterwards, challenger sends params and msk to $\mathcal{A}_4$. Furthermore, the challenger runs the **Partial-Private-Key-Extraction**, **User-Key-Generation**, and **Set-Private-Key** algorithms to create n private keys S_i associated with ID_i, where $(1 \leq i \leq n)$. Finally, the challenger gives all ID_i to $\mathcal{A}_4$.
2) **Phase 1:** $\mathcal{A}_4$ issues queries as in **Game 1**, except the **Partial secret key queries** and the **Replace public key queries** are not allowed to issue in this game.
3) **Challenge:** $\mathcal{A}_4$ gives the challenger a pair of plaintexts of equal-length $M_0, M_1 \in \{0, 1\}^*$. The challenger selects a random bit $\xi \in \{0, 1\}$ and computes $C_\xi =$ **Encrypt**(M_ξ, ID_{ch}). After that, the challenger sends C_ξ to $\mathcal{A}_4$ as the challenge ciphertext.
4) **Phase 2:** The challenger responds to $\mathcal{A}_4$ in a similar way as in **Phase 1** on grounds that:
   - ID_{ch} is not queried in the **Set-Private-key-queries**.
   - (ID_{ch}, C^*) is not queried in the **Decryption-queries**.
   - ID_{ch} is not queried in the **Trapdoor-queries**.
5) **Guess:** $\mathcal{A}_4$ outputs $\xi' \in \{0, 1\}$, and wins if $\xi' = \xi$. The advantage of $\mathcal{A}_4$ in the game above is depicted as follows:

$$Adv^{IND-CCA}_{CL-PKC-ET,\mathcal{A}_4}(k) = |Pr[\xi' = \xi] - \frac{1}{2}|.$$

Definition 4: The CL-PKC-ET scheme is IND-CCA secure if for all IND-CCA adversary $\mathcal{A}_4$, its advantage $Adv^{OW-CCA}_{CL-PKC-ET,\mathcal{A}_4}(k)$ is negligible given the security parameter k.

III. CONCRETE CONSTRUCTION

In this segment, the concrete construction of CL-PKC-ET scheme is proposed as follows.

1) **Setup:** Given a secure parameter k, the **Setup** algorithm proceeds as follows:
   - It generates the pairing parameters: two groups $\mathbb{G}_1$, $\mathbb{G}_2$ of prime order q, and an admissible bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. The algorithm then chooses a random generator $P \in \mathbb{G}_1$.
   - It then chooses cryptographic hash functions: $H_1 : \{0, 1\}^* \to \mathbb{Z}_p^*, H_2 : \mathbb{G}_1 \to \mathbb{Z}_p^*, H_3 : \mathbb{G}_2 \to \{0, 1\}^n$, where n is the number of bits of the message to be sent, and $H_4 : \mathbb{G}_2 \to \mathbb{Z}_p^*$.
   - Randomly choose $(s_1, s_2) \in \mathbb{Z}_p^*$, then set $P_{pub} = s_1 P$ and $P'_{pub} = s_2 P$. The public parameters are $\langle p, \mathbb{G}_1, \mathbb{G}_2, e, g, P_{Pub}, P'_{Pub}, H_1, H_2, H_3 \rangle$, where $g = e(P, P)$.
2) **Partial-Private-Key-Extraction:** When the **KGC** receives the user identity ID, the **KGC** returns the partial private key

$$D_{ID} = (D_1, D_2) = \left(\frac{1}{h_{ID} + s_1} P, \frac{1}{h_{ID} + s_2} P\right),$$

where $h_{ID} = H_1(ID)$.
3) **User-Key-Generation:** The algorithm uses as input params and D_{ID}. It chooses the secret value $x_{ID} \in \mathbb{Z}_p^*$ and sets the public key

$$PK_{ID} = (PK_1, PK_2) = (x_{ID}(h_{ID}P + P_{pub}), x_{ID}(h_{ID}P + P'_{pub})).$$

4) **Set-Private-Key:** The algorithm uses as inputs params, D_{ID}, and x_{ID}. It returns the full private key

$$S = (S_1, S_2) = \left(\frac{1}{x_{ID} + h_1} D_1, \frac{1}{x_{ID} + h_2} D_2\right),$$

where $h_1 = H_2(PK_1)$, $h_2 = H_2(PK_2)$.
5) **Trapdoor:** This algorithm takes as input a private key S and outputs a trapdoor

$$T_{ID} = S_2 = \frac{1}{x_{ID} + h_2} D_2.$$

6) **Encrypt:** Given a message $M \in \{0, 1\}^*$, the identity ID, and the public key $PK_{ID} = (PK_1, PK_2)$ as inputs, the algorithm works as follows:
   - Select $(r_1, r_2) \in_R \mathbb{Z}_p^*$.
   - Calculate $C_1 = r_1(PK_1 + h_1(h_{ID}P + P_{pub}))$.
   - Calculate $C_2 = r_2(PK_2 + h_2(h_{ID}P + P'_{pub}))$.
   - Calculate $C_3 = (M||r_2) \oplus H_3(g^{r_1})$.
   - Calculate $C_4 = (m \cdot r_2) \cdot H_4(g^{r_2})$, where $m = H_1(M)$.
7) **Decrypt:** Given a ciphertext $C = (C_1, C_2, C_3, C_4)$ and a private key $S_1 = \frac{1}{x_{ID}+h_1} D_1$ as inputs. It returns the plaintext M by working as follows.

- Compute $M||r_2 = C_3 \oplus H_3(e(C_1, S_1))$.

$$\begin{aligned} M||r_2 &= C_3 \oplus H_3(e(C_1, S_1)). \\ &= C_3 \oplus H_3(e(r_1(PK_1 + h_1(h_{ID}P + P_{pub})) , \\ &\quad \frac{1}{x_{ID} + h_1} D_1)). \\ &= C_3 \oplus H_3(e(r_1(x_{ID}(h_{ID}P + P_{pub}) \\ &\quad + h_1(h_{ID}P + P_{pub})), \frac{1}{x_{ID}+h_1} \cdot \frac{1}{h_{ID}+s_1} P)). \\ &= C_3 \oplus H_3(e(r_1(h_{ID}P + P_{pub})(x_{ID} + h_1) , \\ &\quad \frac{1}{x_{ID} + h_1} \cdot \frac{1}{h_{ID} + s_1} P)), \\ &= C_3 \oplus H_3(e(r_1\, P(h_{ID} + s_1) , \frac{1}{h_{ID} + s_1} P)), \\ &= C_3 \oplus H_3(e(r_1\, P , P)), \\ &= C_3 \oplus H_3(g^{r_1}). \\ &= M||r_2. \end{aligned}$$

- Verify if $C_2 = r_2(PK_2 + h_2(h_{ID}P + P'_{pub})$ and $\frac{C_4}{(H_1(M)\cdot r_2)} = H_4(e(C_2, S_2))$.
- If both verifications pass, return M. Otherwise, return the symbol $\perp$.

8) **Test**$(C_i, T_{ID,i}, C_j, T_{ID,i})$: Let U_i and U_j be two users of a system. Let $C_i = (C_{i,1}, C_{i,2}, C_{i,3}, C_{i,4})$ and $C_j = (C_{j,1}, C_{j,2}, C_{j,3}, C_{j,4})$ be the ciphertexts of U_i and U_j, respectively. The **Test** algorithm works as follows:

$$\begin{aligned} Q_i &= e(T_{ID,i} , C_{i,2}), \\ &= e(S_{i,2} , r_{i,2}(PK_{i,2} + h_{2_i}(h_{ID_i}P + P'_{pub}))), \\ &= e(\frac{1}{x_{ID_i}+h_{2_i}} D_{i,2} , r_2(PK_{i,2}+h_{2_i}(h_{ID_i}P+P'_{pub}))), \\ &= e(\frac{1}{x_{ID_i}+h_{2_i}} D_{i,2} , r_{i,2}(x_{ID_i}(h_{ID_i}P+P'_{pub}) \\ &\quad + h_{2_i}(h_{ID_i}P + P'_{pub}))), \\ &= e(\frac{1}{x_{ID_i}+h_{2_i}} D_{i,2}, r_{i,2}(h_{ID_i}P+P'_{pub})(x_{ID_i}+h_{2_i}))), \\ &= e(\frac{1}{h_{ID_i} + s_2} P , r_{i,2}(h_{ID_i}P + P'_{pub})), \\ &= e(P , r_{i,2}P), \\ &= g^{r_{i,2}}. \\ R_i &= \frac{C_{i,4}}{H_4(Q_i)}, \\ &= m_i \cdot r_{i,2}. \end{aligned}$$

From the above calculations, we have $Q_j = g^{r_{j,2}}$ and $R_j = m_j \cdot r_{j,2}$. The **Test** algorithm returns 1 if $Q_i^{R_j} = Q_j^{R_i}$. Otherwise, it returns 0.

IV. SECURITY ANALYSIS

Theorem 1: Supposing the mBDHI assumption is intractable. The proposed CL-PKC-ET scheme is OW-CCA secure under the random oracle model.

To prove **Theorem 1**, we use two lemmas as follows:

Lemma 1: Suppose that $\mathcal{A}_1$ breaks the CL-PKC-ET scheme with advantage ε_1. Assume that $\mathcal{A}_1$ makes q_e extract partial private key queries, q_{ex} extract private key queries, q_d decryption queries, q_t trapdoor query, q_{H_1} hash queries to H_1, and q_{H_3} hash queries to H_3. Then, we constructed an algorithm $\mathcal{B}_1$ to solve the mBDHI problem with advantage:

$$\varepsilon_1 \geq \frac{\varepsilon}{q_{H_3} + 2q_{H_4}} \left(1 - \frac{q_d}{2^k}\right)^{q_e+q_{ex}+q_t}$$

Proof:

1) We demonstrate how an algorithm $\mathcal{B}_1$ can solve a random given instance $(P, \alpha P, \mu)$ of the mBDHI problem by reacting with $\mathcal{A}_1$ as subroutine, where $\alpha, \mu \in \mathbb{Z}_p^*$. Algorithm $\mathcal{B}_1$ runs the **Setup** and picks a master secret key $s_1, s_2 \in \mathbb{Z}_p^*$ randomly, then set $P_{pub} = s_1 P$, $P'_{pub} = s_2 P$, and $g = e(P, P)$. Also, $\mathcal{B}_1$ picks a random challenge identity $ID^* \in \{0, 1\}^*$. Finally, $\mathcal{B}_1$ gives the public parameters *params* and ID^* to $\mathcal{A}_1$ how then makes the following queries.
2) **Phase 1:**
   - H_1**-queries:** $\mathcal{B}_1$ prepares a list of tuples $\langle ID_i, \beta_i \rangle$ denoted by L^{H_1}. Upon receiving the ID_i, $\mathcal{B}_1$ works as follows:
     - If ID_i already exists in tuples $\langle ID_i, \beta_i \rangle$ in L^{H_1}, $\mathcal{B}_1$ returns β_i to $\mathcal{A}_1$.
     - Else, $\mathcal{B}_1$ chooses $\beta \in \mathbb{Z}_p^*$ randomly. Then, $\mathcal{B}_1$ adds $\langle ID_i, \beta \rangle$ into L^{H_1}, and returns β to $\mathcal{A}_1$
   - H_2**-queries:** $\mathcal{B}_1$ prepares a list of tuples $\langle PK_{ID_i}, z_i \rangle$ denoted by L^{H_2}. Upon receiving the PK_{ID_i}, $\mathcal{B}_1$ works as follows:
     - If $PK_{ID_i} = (PK_{ID_{i,1}}, PK_{ID_{i,2}})$ already exists in tuples $\langle PK_{ID_i}, z_i \rangle$ in L^{H_2}, $\mathcal{B}_1$ returns $z_i = (z_{i,1}, z_{i,2})$ to $\mathcal{A}_1$.
     - Else, $\mathcal{B}_1$ checks if $PK^*_{ID_i} = (PK^*_{i,1}, PK^*_{i,2}) = (\alpha(\beta P + P_{pub}), \alpha(\beta P + P'_{pub}))$, it returns $\mu_i = (\mu_{i,1}, \mu_{i,2})$ and adds L^{H_2} with $\langle PK_{ID_i}, \mu \rangle$.
     - Otherwise, $\mathcal{B}_1$ chooses $z = (z_1, z_2) \in \mathbb{Z}_p^*$ randomly. Then, $\mathcal{B}_1$ adds $\langle PK_{ID_i}, z \rangle$ into L^{H_2}, and returns z to $\mathcal{A}_1$
   - H_3**-queries:** $\mathcal{B}_1$ prepares a list of tuples $\langle v_i, w_i \rangle$ denoted by L^{H_3}. When $\mathcal{A}_1$ makes query to H_3-**queries** with $v_i \in \{0, 1\}^n$, $\mathcal{B}_1$ works as follows:
     - If v_i already exists in tuples $\langle v_i, w_i \rangle$ in L^{H_3}, $\mathcal{B}_1$ returns w_i.
     - Else, $\mathcal{B}_1$ chooses $w \in \{0, 1\}^n$ randomly. Then, $\mathcal{B}_1$ adds $\langle v_i, w \rangle$ into L^{H_3}, and returns w.
   - H_4**-queries:** $\mathcal{B}_1$ prepares a list of tuples $\langle l_i, u_i \rangle$ denoted by L^{H_4}. When $\mathcal{A}_1$ makes query to H_4-**queries** with $l_i \in \mathbb{Z}_p^*$, $\mathcal{B}_1$ works as follows:
     - If l_i already exists in tuples $\langle l_i, u_i \rangle$ in L^{H_4}, $\mathcal{B}_1$ returns u_i.
     - Else, $\mathcal{B}_1$ chooses $u \in \mathbb{Z}_p^*$ randomly. Then, $\mathcal{B}_1$ adds $\langle l_i, u \rangle$ into L^{H_4}, and returns u.

– In addition, $\mathcal{B}_1$ emulates random oracle on its own to obtain $w_i = H_3(g^{r_{i,1}}) \in \{0,1\}^n$ and calculates $C_{i,3} = (M||r_{i,2}) \oplus w_i$ and $\delta_i = \psi \cdot e(P,P)^{u_i}$, where $\psi = g^{r_{i,1}}$. Eventually, $\mathcal{B}_1$ adds the tuple $\langle l_i, u_i, C_{i,3}, \delta_i \rangle$ to the L^{H_4} list.

- **Partial-Private-Key-Extraction-queries:** Algorithm $\mathcal{B}_1$ prepared a list denoted by L^{PPK}. This list consists a tuples $\langle ID_i, D_{ID_i} \rangle$. When $\mathcal{A}_1$ requests the **Partial-Private-Key-Extraction-queries**, algorithm $\mathcal{B}_1$ executes the H_1**-queries** to get $\langle ID_i, \beta_i \rangle$ and then works as follows:
  – If $ID_i \neq ID^*$, $\mathcal{B}_1$ calculates $D_{ID_i} = (D_{i,1}, D_{i,2}) = (\frac{1}{\beta_i + s_1} P, \frac{1}{\beta_i + s_2} P)$ and returns D_{ID_i} to $\mathcal{A}_1$, and then updates L^{PPK} by adding $\langle ID_i, D_{ID_i} \rangle$ tuples.
  – Else, $\mathcal{B}_1$ aborts and stops.
- **Public-Key-queries:** Algorithm $\mathcal{B}_1$ prepared a list denoted by L^{PK}. This list consists a tuples $\langle ID_i, x_i, PK_{ID_i} \rangle$. When $\mathcal{A}_1$ requests the Public-Key-query, algorithm $\mathcal{B}_1$ reacts as follows:
  – If $ID_i = ID^*$, $\mathcal{B}_1$ calculates the public key as $PK^*_{ID_i} = (PK^*_{i,1}, PK^*_{i,2}) = (\beta\alpha P + s_1\alpha P), (\beta\alpha P + s_2\alpha P)$
  – If PK_{ID_i} already exists in tuples $\langle ID_i, x_i, PK_{ID_i} \rangle$ in L^{PK}, $\mathcal{B}_1$ returns PK_{ID_i} to $\mathcal{A}_1$.
  – Else, $\mathcal{B}_1$ chooses $x_i \in \mathbb{Z}_p^*$ randomly, and calculates $PK_{ID_i} = (PK_{i,1}, PK_{i,2}) = (x_i(\beta_i P + P_{pub}), x_i(\beta_i P + P'_{pub}))$.
  – Finally, $\mathcal{B}_1$ adds $\langle ID_i, x_i, PK_{ID_i} \rangle$ into L^{PK}, and returns PK_{ID_i} to $\mathcal{A}_1$.
- **Replace-Public-Key-queries:** $\mathcal{A}_1$ replaces the public key of any user with random value.
- **Set-Private-Key-queries:** If $ID_i = ID^*$, $\mathcal{B}_1$ aborts. Else, $\mathcal{B}_1$ works as follows:
  – If the tuples $\langle ID_i, D_{ID_i} \rangle$ of L^{PPK} and the tuples of $\langle ID_i, x_i, PK_{ID_i} \rangle$ of L^{PK} already exist, $\mathcal{B}_1$ assigns $S_i = (S_{i,1}, S_{i,2}) = \left(\frac{1}{x_i + z_{i,1}} D_{i,1}, \frac{1}{x_i + z_{i,2}} D_{i,2}\right)$.
  – Else, $\mathcal{B}_1$ generates a new private key information and returns $S_{ID_i} = (S_{i,1}, S_{i,2}) = \left(\frac{1}{x_i + z_{i,}} D_{i,1}, \frac{1}{x_i + z_{i,2}} D_{i,2}\right)$ to $\mathcal{A}_1$.
- **Trapdoor-queries:** For this query $\mathcal{B}_1$ responds to the $\mathcal{A}_1$ with $S_{i,2} = \frac{1}{x_i + z_{i,2}} D_{i,2}$.
- **Decryption-queries:** Let $C_i = (C_{i,1}, C_{i,2}, C_{i,3}, C_{i,4})$. To get the private key S_i associated with ID_i, $\mathcal{B}_1$ performs a **Set-Private-Key-queries**. $\mathcal{B}_1$ searches the list L^{H_4} for the inputs of the form $\langle l_i, u_i, C_{i,3}, \delta_i \rangle$ indexed by $i \in \{1, ..., q_{H_4}\}$. If this input is not there, $\mathcal{B}_1$ aborts and stops. Else, $\mathcal{B}_1$ verifies if

$$C_{i,2} = r_{i,2}(PK_{i,2} + z_{i,2}(\beta_i P + P'_{pub}) \quad and$$

$$\frac{C_{i,4}}{(H_1(M) \cdot r_{i,2})} = H_4(e(C_{i,2}, S_{i,2})).$$

If both verifications pass, $\mathcal{B}_1$ returns M. Otherwise, $\mathcal{B}_1$ aborts and rejects ciphertext. For all queries, the probability of rejected valid ciphertext is not more than $\frac{q_d}{2^k}$.

3) **Challenge:** $\mathcal{A}_1$ returns identity ID_{ch} on which it wishes to be challenged. The algorithm $\mathcal{B}_1$ chooses $M^* \in \{0,1\}^*$, $\lambda_1, \lambda_2 \in \mathbb{Z}_p^*$, $C_3^* \in \{0,1\}^*$ and $C_4 \in \mathbb{Z}_p^*$ randomly and then calculates $C_1^* = \lambda_1 P$ and $C_2^* = \lambda_2 P$. Finally, $\mathcal{B}_1$ sends C^* to $\mathcal{A}_1$ as a challenge ciphertext. $\mathcal{A}_1$ can not get the plaintext unless it queries H_3 or H_4 on $e(C_1^*, S_1)$.
4) **Phase 2:** In this phase, the response of the challenger to $\mathcal{A}_1$ is similar of that one obtained in **Phase 1**. The following grounds are considered.
   - ID_{ch} is not queried in the **Set-Private-Key-queries**.
   - If the public key associated with ID_{ch} is replaced, the ID_{ch} is not queried in the **Partial-Private-Key-Extraction-queries**.
   - If the public key of the user is replaced, the corresponding identity ID_i is not queried in the **Set-Private-Key-queries**.
   - (ID_{ch}, C^*) should not be queried in the **Decryption-queries**.
5) **Guess:** $\mathcal{A}_1$ returns M' for M^*. $\mathcal{B}_1$ picks a random input $\langle v_i, w_i \rangle \in_R L^{H_3}$ or $\langle l_i, u_i, C_{i,3}, \delta_i \rangle \in_R L^{H_4}$. Considering the L^{H_3} includes no more than $q_{H_3} + q_{H_4}$ inputs, the chosen input will include the correct item $\psi = e(C_1^*, S_1)$ with the probability $1/(q_{H_3} + 2q_{H_4})$. In fact, the mBDHI problem can be recognized by the following equation

$$\psi = e(C_1^*, S_1) = e(\lambda_1 P, \frac{1}{\alpha + \mu_1} \cdot \frac{1}{\beta + s_1} P).$$

According to the proof in [3], we have

$$e(P,P)^{\frac{1}{\alpha+\mu_1}} = \psi^{\frac{\beta+s_1}{\lambda_1}}.$$

The advantages of $\mathcal{B}_1$ is ε_1 in solving the q-IBDH problem is as follows:

1) H_1**-queries**, H_2**-queries**, H_3**-queries**, and H_4**-queries** are respond to the $\mathcal{A}_1$ as in the real attack. Therefore, any response is answered with a random value.
2) All replies to **Decryption-queries** is valid except when E (reject a valid ciphertext) happens. The algorithm $\mathcal{B}_1$ is not broken if E does not happen. Then, we have:

$$Pr[\neg abort] = Pr[\neg E]$$

Since $Pr[E] \leq \frac{q_d}{2^k}$, Thus, we have

$$Pr[\neg abort] \geq \left(1 - \frac{q_d}{2^k}\right)^{q_e + q_{ex} + q_t}$$

Also, the algorithm $\mathcal{B}_1$ picks the right item from L^{H_3} or L^{H_4} with the probability $1/(q_{H_3} + 2q_{H_4})$. Thus, we have

$$\varepsilon_1 \geq \frac{\varepsilon}{q_{H_3} + 2q_{H_4}} \left(1 - \frac{q_d}{2^k}\right)^{q_e + q_{ex} + q_t}$$

The proof of **Lemma 1** is finished.

Lemma 2: Suppose that $\mathcal{A}_2$ breaks the CL-PKC-ET scheme with advantage ε_2. Assume that $\mathcal{A}_2$ makes q_{ex} extract private key query, q_t trapdoor query, q_{H_1} hash query to H_1, and q_{H_3} hash query to H_3. Then, we constructed an algorithm $\mathcal{B}_2$ to resolve the q-IBDH problem with advantage:

$$\varepsilon_2 \geq \frac{\varepsilon}{q_{H_3} + 2q_{H_4}} \left(1 - \frac{q_d}{2^k}\right)^{q_{ex}+q_t}$$

Proof:

1) We demonstrate how an algorithm $\mathcal{B}_2$ can solve a random given instance $(P, \alpha P, \mu)$ of the mBDHI problem by reacting with $\mathcal{A}_2$ as subroutine, where $\alpha, \mu \in \mathbb{Z}_p^*$. Algorithm $\mathcal{B}_2$ runs the **Setup** and picks a master secret key $s_1, s_2 \in \mathbb{Z}_p^*$ randomly, then set $P_{pub} = s_1 P$, $P'_{pub} = s_2 P$, and $g = e(P, P)$. Also, $\mathcal{B}_2$ picks a random challenge identity $ID^* \in \{0, 1\}^*$. Finally, $\mathcal{B}_2$ gives the public parameters params and msk to $\mathcal{A}_2$ how then makes the following queries.
2) **Phase 1**: In this phase, the response of $\mathcal{B}_2$ to $\mathcal{A}_2$ is similar of that one obtained in **Phase 1** in proof of **Lemma 1**, except the **Partial-secret key queries** and the **Replace-public-key-queries** are not allowed to issue in this phase.
3) **Challenge:** $\mathcal{A}_2$ returns identity ID_{ch} on which it wishes to be challenged. The algorithm $\mathcal{B}_2$ chooses $M^* \in \{0, 1\}^*$, $\lambda_1, \lambda_2 \in \mathbb{Z}_p^*$, $C_3^* \in \{0, 1\}^*$ and $C_4 \in \mathbb{Z}_p^*$ randomly and then calculates $C_1^* = \lambda_1 P$ and $C_2^* = \lambda_2 P$. Finally, $\mathcal{B}_2$ sends C^* to $\mathcal{A}_2$ as a challenge ciphertext. $\mathcal{A}_2$ can not get the plaintext unless it queries H_3 or H_4 on $e(C_1^*, S_1)$.
4) **Phase 2:** In this phase, the response of the challenger to $\mathcal{A}_2$ is similar of that one obtained in **Phase 1** on grounds that:
   - ID_{ch} is not queried in the **Set-Private-Key-queries**.
   - (ID_{ch}, C^*) is not queried in the **Decryption-queries**.
5) **Guess:** $\mathcal{A}_2$ returns M' for M^*. $\mathcal{B}_2$ picks a random input $\langle v_i, w_i \rangle \in_R L^{H_3}$ or $\langle l_i, u_i, C_{i,3}, \delta_i \rangle \in_R L^{H_4}$. Considering the L^{H_3} includes no more than $q_{H_3} + q_{H_4}$ inputs, the chosen input will include the correct item $\psi = e(C_1^*, S_1)$ with the probability $1/(q_{H_3} + 2q_{H_4})$. In fact, the mBDHI problem can be recognized by the following equation

$$\psi = e(C_1^*, S_1) = e(\lambda_1 P, \frac{1}{\alpha + \mu_1} \cdot \frac{1}{\beta + s_1} P).$$

According to the proof in [3], we have

$$e(P, P)^{\frac{1}{\alpha+\mu_1}} = \psi^{\frac{\beta+s_1}{\lambda_1}}.$$

The advantages of $\mathcal{B}_2$ is ε_2 in solving the mBDHI problem is as follows:

1) H_1**-queries**, H_2**-queries**, H_3**-queries**, and H_4**-queries** are respond to the $\mathcal{A}_2$ as in the real attack. Therefore, any response is answered with a random value.
2) All replies to **Decryption-queries** is valid except when E (reject a valid ciphertext) happens. The algorithm $\mathcal{B}_2$ is not broken if E does not happen. Then, we have:

$$Pr[\neg abort] = Pr[\neg E]$$

Since $Pr[E] \leq \frac{q_d}{2^k}$, Thus, we have

$$Pr[\neg abort] \geq \left(1 - \frac{q_d}{2^k}\right)^{q_{ex}+q_t}$$

Also, the algorithm $\mathcal{B}_2$ picks the right item from L^{H_3} or L^{H_4} with the probability $1/(q_{H_3} + 2q_{H_4})$. Thus, we have

$$\varepsilon_2 \geq \frac{\varepsilon}{q_{H_3} + 2q_{H_4}} \left(1 - \frac{q_d}{2^k}\right)^{q_{ex}+q_t}$$

The proof of **Lemma 2** is finished.

Theorem 2: Supposing the mBDHI assumption is intractable. The proposed CL-PKC-ET scheme is IND-CCA secure under the random oracle model.

To prove **Theorem 2**, we use two lemmas as follows:

Lemma 3: Suppose that $\mathcal{A}_3$ breaks the CL-PKC-ET scheme with advantage ε_3. Assume that $\mathcal{A}_3$ makes q_e extract partial private key queries, q_{ex} extract private key queries, q_d decryption queries, q_t trapdoor queries, q_{H_1} hash queries to H_1, and q_{H_3} hash queries to H_3. Then, we constructed an algorithm $\mathcal{B}_3$ to solve the mBDHI problem with advantage:

$$\varepsilon_3 \geq \frac{\varepsilon}{q_{H_3} + 2q_{H_4}} \left(1 - \frac{q_d}{2^k}\right)^{q_e+q_{ex}+q_t}$$

Proof:

1) We demonstrate how an algorithm $\mathcal{B}_3$ can solve a random given instance $(P, \alpha P, \mu)$ of the mBDHI problem by reacting with $\mathcal{A}_3$ as subroutine, where $\alpha, \mu \in \mathbb{Z}_p^*$. Algorithm $\mathcal{B}_3$ runs the **Setup** and picks a master secret key $s_1, s_2 \in \mathbb{Z}_p^*$ randomly, then set $P_{pub} = s_1 P$, $P'_{pub} = s_2 P$, and $g = e(P, P)$. Also, $\mathcal{B}_3$ picks a random challenge identity $ID^* \in \{0, 1\}^*$. Finally, $\mathcal{B}_3$ gives the public parameters params and ID^* to $\mathcal{A}_3$ how then makes the following queries.
2) **Phase 1**: This phase is similar to the **Phase 1** in the proof of **Lemma 1**.
3) **Challenge:** $\mathcal{A}_3$ outputs two plaintext $M_0, M_1 \in_R \{0, 1\}^*$ and the identity ID_{ch} on which it wishes to be challenged. The algorithm $\mathcal{B}_3$ chooses $\xi \in_R \{0, 1\}$, and randomly chooses $\lambda_1, \lambda_2 \in \mathbb{Z}_p^*$, $C_3^* \in \{0, 1\}^*$ and $C_4^* \in \mathbb{Z}_p^*$ and calculates $C_1^* = \lambda_1 P$ and $C_2^* = \lambda_2 P$. Finally, $\mathcal{B}_3$ sends C_ξ^* to $\mathcal{A}_3$ as a challenge ciphertext. $\mathcal{A}_3$ can not get the plaintext unless it queries H_3 or H_4 on $e(C_1^*, S_1)$.
4) **Phase 2:** In this phase, the response of the challenger to $\mathcal{A}_3$ is similar of that one obtained in **Phase 1**. The following grounds are considered.
   - ID_{ch} is not queried in the **Set-Private-Key-queries**.
   - If the public key associated with ID_{ch} is replaced, the ID_{ch} is not queried in the **Partial-Private-Key-Extraction-queries**.

- If the public key of the user is replaced, the corresponding identity ID_i is not queried in the **Set-Private-Key-queries**.
- (ID_{ch}, C^*) should not be queried in the **Decryption-queries**.
- ID_{ch} should not be queried in the **Trapdoor-queries**.

5) **Guess:** $\mathcal{A}_3$ returns ξ' for ξ. $\mathcal{B}_3$ picks a random input $\langle v_i, w_i\rangle \in_R L^{H_3}$ or $\langle l_i, u_i, C_{i,3}, \delta_i\rangle \in_R L^{H_4}$. Considering the L^{H_3} includes no more than $q_{H_3}+q_{H_4}$ inputs, the chosen input will include the correct item $\psi = e(C_1^*, S_1)$ with the probability $1/(q_{H_3}+2q_{H_4})$. In fact, the mBDHI problem can be recognized by the following equation

$$\psi = e(C_1^*, S_1) = e(\lambda_1 P, \frac{1}{\alpha+\mu_1}\cdot\frac{1}{\beta+s_1}P).$$

According to the proof in [3], we have

$$e(P,P)^{\frac{1}{\alpha+\mu_1}} = \psi^{\frac{\beta+s_1}{\lambda_1}}.$$

The advantages of $\mathcal{B}_3$ is ε_3 in solving the mBDHI problem is as follows:

1) H_1**-queries**, H_2**-queries**, H_3**-queries**, and H_4**-queries** are respond to the $\mathcal{A}_3$ as in the real attack. Therefore, any response is answered with a random value.
2) All replies to **Decryption-queries** is valid except when E (reject a valid ciphertext) happens. The algorithm $\mathcal{B}_3$ is not broken if E does not happen. Then, we have:

$$Pr[\neg abort] = Pr[\neg E]$$

Since $Pr[E] \le \frac{q_d}{2^k}$, Thus, we have

$$Pr[\neg abort] \ge \left(1-\frac{q_d}{2^k}\right)^{q_e+q_{ex}+q_t}$$

Also, the algorithm $\mathcal{B}_3$ picks the right item from L^{H_3} or L^{H_4} with the probability $1/(q_{H_3}+2q_{H_4})$. Thus, we have

$$\varepsilon_3 \ge \frac{\varepsilon}{q_{H_3}+2q_{H_4}}\left(1-\frac{q_d}{2^k}\right)^{q_e+q_{ex}+q_t}$$

The proof of **Lemma 3** is finished.

Lemma 4: Suppose that $\mathcal{A}_4$ breaks the CL-PKC-ET scheme with advantage ε_4. Assume that $\mathcal{A}_2$ makes q_{ex} extract private key query, q_t trapdoor query, q_{H_1} hash query to H_1, and q_{H_3} hash query to H_3. Then, we constructed an algorithm $\mathcal{B}_4$ to resolve the mBDHI problem with advantage:

$$\varepsilon_4 \ge \frac{\varepsilon}{q_{H_3}+2q_{H_4}}\left(1-\frac{q_d}{2^k}\right)^{q_{ex}+q_t}$$

Proof:

1) We demonstrate how an algorithm $\mathcal{B}_4$ can solve a random given instance $(P, \alpha P, \mu)$ of the mBDHI problem by reacting with $\mathcal{A}_4$ as subroutine, where $\alpha, \mu \in \mathbb{Z}_p^*$. Algorithm $\mathcal{B}_4$ runs the **Setup** and picks a master secret key $s_1, s_2 \in \mathbb{Z}_p^*$ randomly, then set $P_{pub} = s_1P$, $P'_{pub} = s_2P$, and $g = e(P,P)$. Also, $\mathcal{B}_4$ picks a random challenge identity $ID^* \in \{0,1\}^*$. Finally, $\mathcal{B}_4$ gives the public parameters *params* and *msk* to $\mathcal{A}_4$ how then makes the following queries.

TABLE 1. Symbols and running times (ms).

Symbols	Characterization	Times
T_{e_1}	The exponentiation operation in $\mathbb{G}_1$	5.378
T_{e_2}	The exponentiation operation in $\mathbb{G}_2$	1.369
T_p	The pairing operation	11.370
T_h	The hash function	0.0006
T_m	The point multiplication operation	0.0282

2) **Phase 1**: In this phase, the response of $\mathcal{B}_4$ to $\mathcal{A}_4$ is similar of that one obtained in **Phase 1** in proof of **Lemma 3**, except the **Partial-Private-Key-Extraction-queries** and the **Replace-Public-Key-queries** are not allowed to issue in this phase.
3) **Challenge:** $\mathcal{A}_4$ outputs two plaintext $M_0, M_1 \in_R \{0,1\}^*$ and the identity ID_{ch} on which it wishes to be challenged. The algorithm $\mathcal{B}_4$ chooses $\xi \in_R \{0,1\}$, and randomly chooses $\lambda_1, \lambda_2 \in \mathbb{Z}_p^*$, $C_3^* \in \{0,1\}^*$ and $C_4^* \in \mathbb{Z}_p^*$ and calculates $C_1^* = \lambda_1 P$ and $C_2^* = \lambda_2 P$. Finally, $\mathcal{B}_4$ sends C_ξ^* to $\mathcal{A}_4$ as a challenge ciphertext. $\mathcal{A}_4$ can not get the plaintext unless it queries H_3 or H_4 on $e(C_1^*, S_1)$.
4) **Phase 2:** In this phase, the response of the challenger to $\mathcal{A}_4$ is similar of that one obtained in **Phase 1** on grounds that:
   - ID_{ch} is not queried in the **Set-Private-Key-queries**.
   - (ID_{ch}, C^*) is not queried in the **Decryption-queries**.
   - ID_{ch} is not queried in the **Trapdoor-queries**.
5) **Guess:** $\mathcal{A}_4$ returns ξ' for ξ. $\mathcal{B}_4$ picks a random input $\langle v_i, w_i\rangle \in_R L^{H_3}$ or $\langle l_i, u_i, C_{i,3}, \delta_i\rangle \in_R L^{H_4}$. Considering the L^{H_3} includes no more than $q_{H_3}+q_{H_4}$ inputs, the chosen input will include the correct item $\psi = e(C_1^*, S_1)$ with the probability $1/(q_{H_3}+2q_{H_4})$. In fact, the mBDHI problem can be recognized by the following equation

$$\psi = e(C_1^*, S_1) = e(\lambda_1 P, \frac{1}{\alpha+\mu_1}\cdot\frac{1}{\beta+s_1}P).$$

According to the proof in [3], we have

$$e(P,P)^{\frac{1}{\alpha+\mu_1}} = \psi^{\frac{\beta+s_1}{\lambda_1}}.$$

The advantages of $\mathcal{B}_4$ is ε_4 in solving the mBDHI problem is as follows:

1) H_1**-queries**, H_2**-queries**, H_3**-queries**, and H_4**-queries** are respond to the $\mathcal{A}_4$ as in the real attack. Therefore, any response is answered with a random value.
2) All replies to **Decryption-queries** is valid except when E happens (reject a valid ciphertext). The algorithm $\mathcal{B}_4$ is not broken if E does not happen. Then, we have:

$$Pr[\neg abort] = Pr[\neg E]$$

Since $Pr[E] \le \frac{q_d}{2^k}$, Thus, we have

$$Pr[\neg abort] \ge \left(1-\frac{q_d}{2^k}\right)^{q_{ex}+q_t}$$

TABLE 2. Comparison of computation cost.

Scheme	Encryption	Decryption	Test
Qu *et al.* [23]	$4T_p$+$6T_{e_1}$+$5T_h$+T_m=77.779	$2T_p$+$2T_{e_1}$+$4T_h$+T_m=33.527	$4T_p$+$2T_h$=45.481
Our scheme	$2T_{e_2}$+$5T_h$+$2T_m$=2.797	$2T_p$+$5T_h$+T_m=22.771	$2T_p$+$2T_{e_2}$+$2T_h$=25.479

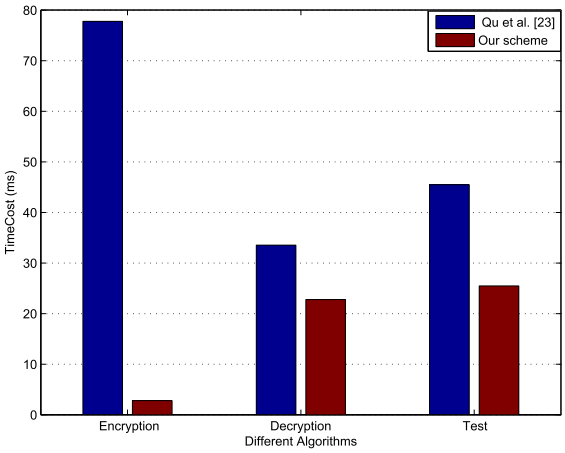


FIGURE 3. Comparison of computation cost.

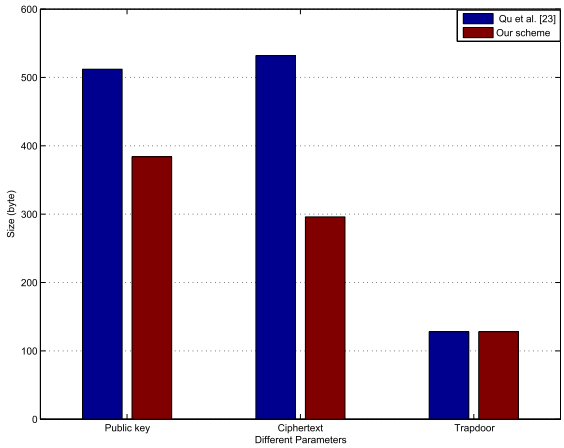


FIGURE 4. Comparison of communication cost.

TABLE 3. Comparison of communication cost.

Scheme	$\|PK\|$	$\|CT\|$	$\|TD\|$
Qu *et al.* [23]	$4\|\mathbb{G}_1\|$	$4\|\mathbb{G}_1\|$+$\|\mathbb{Z}_p\|$	$\|\mathbb{G}_1\|$
Our scheme	$3\|\mathbb{G}_1\|$	$2\|\mathbb{G}_1\|$+$2\|\mathbb{Z}_p\|$	$\|\mathbb{G}_1\|$

Also, the algorithm $\mathcal{B}_4$ picks the right item from L^{H_3} or L^{H_4} with the probability $1/(q_{H_3} + 2q_{H_4})$. Thus, we have

$$\varepsilon_4 \geq \frac{\varepsilon}{q_{H_3} + 2q_{H_4}} \left(1 - \frac{q_d}{2^k}\right)^{q_{ex}+q_t}$$

The proof of **Lemma 4** is finished.

V. PERFORMANCE ANALYSIS

When we consider computation and communication costs, the performance analysis of our suggested scheme and that of Qu *et al.*'s scheme [23] are discussed under this segment.

A. COMPUTATION COST

In Table 1, the running times and symbols are presented. According to Pairing-Based Cryptography (PBC) library [17], the performance analysis are carried out on a personal computer (windows 10 pro operating system with Intel(R) Core(TM) i7-7700 CPU @3.60 GHz @3.60 GHz and 8GB RAM) using C++. In our experiment, we made use of Type A pairings constructed from the curve $z^2 = x^3 + x$ over a finite field $\mathbb{F}_p$. In addition, we realized 128 bits of RSA security levels [7]. According to Table 2 and FIGURE 3, the computation cost based on our suggested scheme is decreased by 96.40%, 32.08%, and 43.98% in Encryption, Decryption, and Test stages respectively, when compared with Qu *et al.*'s scheme.

B. COMMUNICATION COST

Suppose that the $|PK|$, $|CT|$, and $|TD|$ represented the size of public key, the size of ciphertext, and the size of trapdoor, respectively. Assume that the size of each element in $\mathbb{Z}_p^*$, $\mathbb{G}_1$, and $\mathbb{G}_2$ are 20 bytes, 128 bytes, and 128 bytes, respectively. Let $|\mathbb{G}_1|$, $|\mathbb{G}_2|$, and $|\mathbb{Z}_p|$ denote the size of a point in $\mathbb{G}_1$, $\mathbb{G}_2$, and the bit length in $\mathbb{Z}_p^*$, respectively. According to Table 3 and Figure 4, our suggested scheme has less communication cost as compared to Qu *et al.*'s scheme.

VI. CONCLUSION

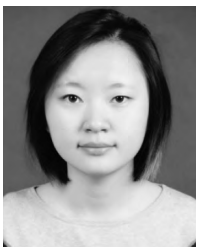
The scheme we present in this paper can be implemented to provide an efficient search on the encrypted data accumulated within the cloud. This efficient CL-PKC-ET scheme takes the feature of the equality test that can perform the search between two keywords encrypted under the different public key as well as the same public key. In addition, the CL-PKC-ET scheme is constructed under the certificateless cryptosystem (CLC). Thus, it is solved the problems of certificate management which appeared with the equality test schemes constructed under the PKI cryptosystem and the key escrow problem which appeared with the equality test schemes constructed under the identity-based cryptosystem (IBC). Furthermore, we demonstrate the security of our scheme based on the OW-CCA and IND-CCA in the random oracle model assuming the mBDHI problem is intractable. We further evaluated our scheme regarding the computation and communication costs with the equality test scheme constructed under the CLC. Based on this comparison made with the scheme in [23], our scheme accomplishes 96.40%, 32.08%, and 43.98% reduction in computation costs during the encryption, decryption and test stages, respectively. Therefore, we assert that this efficient CL-PKC-ET scheme is more preferable for deployment in both cloud and IoV environments. The future work consists of constructing CL-PKC-ET scheme without using the bilinear pairings.

REFERENCES

[1] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Asiacrypt*, vol. 2894. Berlin, Germany: Springer, 2003, pp. 452–473.

[2] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.
[3] D. Boneh and X. Boyen, "Short signatures without random oracles," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, May 2004, pp. 56–73.
[4] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, May 2004, pp. 506–522.
[5] A. A. Cárdenas, P. K. Manadhata, and S. P. Rajan, "Big data analytics for security," *IEEE Security Privacy*, vol. 11, no. 6, pp. 74–76, Nov./Dec. 2013.
[6] L. Chen and Z. Cheng, "Security proof of Sakai-Kasahara's identity-based encryption scheme," in *Proc. IMA Int. Conf. Cryptogr. Coding*, Dec. 2005, pp. 442–459.
[7] J. Daemen and V. Rijmen, *The Design of Rijndael: AES-the Advanced Encryption Standard*. Berlin, Germany: Springer, 2013.
[8] C. Esposito, A. Castiglione, B. Martini, and K.-K. R. Choo, "Cloud manufacturing: Security, privacy, and forensic concerns," *IEEE Cloud Comput.*, vol. 3, no. 4, pp. 16–22, Jul./Aug. 2016.
[9] J. A. Guerrero-Ibanez, S. Zeadally, and J. Contreras-Castillo, "Integration challenges of intelligent transportation systems with connected vehicle, cloud computing, and Internet of Things technologies," *IEEE Wireless Commun.*, vol. 22, no. 6, pp. 122–128, Dec. 2015.
[10] L. Guo, M. Dong, K. Ota, Q. Li, T. Ye, J. Wu, and J. Li, "A secure mechanism for big data collection in large scale Internet of vehicle," *IEEE Internet Things J.*, vol. 4, no. 2, pp. 601–610, Apr. 2017.
[11] K. Huang, R. Tso, Y.-C. Chen, S. M. M. Rahman, A. Almogren, and A. Alamri, "PKE-AET: Public key encryption with authorized equality test," *Comput. J.*, vol. 58, no. 10, pp. 2686–2697, Apr. 2015.
[12] O. Kaiwartya, A. H. Abdullah, Y. Cao, A. Altameem, M. Prasad, C.-T. Lin, and X. Liu, "Internet of vehicles: Motivation, layered architecture, network model, challenges, and future aspects," *IEEE Access*, vol. 4, pp. 5356–5373, 2016.
[13] N. Kumar, N. Chilamkurti, and J. J. P. C. Rodrigues, "Bayesian coalition game as-a-service for content distribution in Internet of vehicles," *IEEE Internet Things J.*, vol. 1, no. 6, pp. 544–555, Dec. 2014.
[14] H. Li, X. Lin, H. Yang, X. Liang, R. Lu, and X. Shen, "EPPDR: An efficient privacy-preserving demand response scheme with adaptive key evolution in smart grid," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 8, pp. 2053–2064, Aug. 2014.
[15] H. Li, Y. Yang, T. H. Luan, X. Liang, L. Zhou, and X. S. Shen, "Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data," *IEEE Trans. Dependable Secure Comput.*, vol. 13, no. 3, pp. 312–325, May/Jun. 2016.
[16] C. Liu, X. Zhang, C. Liu, Y. Yang, R. Ranjan, D. Georgakopoulos, and J. Chen, "An iterative hierarchical key exchange scheme for secure scheduling of big data applications in cloud computing," in *Proc. 12th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Jul. 2013, pp. 9–16.
[17] B. Lynn *et al.* (2006). *The Pairing-Based Cryptography Library*. Accessed: Mar. 27, 2013. [Online]. Available: http://crypto.stanford.edu/pbc/
[18] S. Ma, "Identity-based encryption with outsourced equality test in cloud computing," *Inf. Sci.*, vol. 328, pp. 389–402, Jan. 2016.
[19] S. Ma, Q. Huang, M. Zhang, and B. Yang, "Efficient public key encryption with equality test supporting flexible authorization," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 3, pp. 458–470, Mar. 2015.
[20] S. Ma, M. Zhang, Q. Huang, and B. Yang, "Public key encryption with delegated equality test in a multi-user setting," *Comput. J.*, vol. 58, no. 4, pp. 986–1002, Apr. 2015.
[21] K. Mershad and H. Artail, "A framework for secure and efficient data acquisition in vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 62, no. 2, pp. 536–551, Feb. 2013.
[22] Z. Ning, X. Hu, Z. Chen, M. Zhou, B. Hu, J. Cheng, and M. S. Obaidat, "A cooperative quality-aware service access system for social Internet of vehicles," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2506–2517, Aug. 2018.
[23] H. Qu, Z. Yan, X.-J. Lin, Q. Zhang, and L. Sun, "Certificateless public key encryption with equality test," *Inf. Sci.*, vol. 462, pp. 76–92, Sep. 2018.
[24] Q. Tang, "Towards public key encryption scheme supporting equality test with fine-grained authorization," in *Proc. Australas. Conf. Inf. Secur. Privacy*, Jul. 2011, pp. 389–406.
[25] Q. Tang, "Public key encryption schemes supporting equality test with authorisation of different granularity," *Int. J. Appl. Cryptogr.*, vol. 2, no. 4, pp. 304–321, 2012.
[26] Q. Tang, "Public key encryption supporting plaintext equality test and user-specified authorization," *Secur. Commun. Netw.*, vol. 5, no. 12, pp. 1351–1362, 2012.
[27] H. Wang, B. Qin, Q. Wu, L. Xu, and J. Domingo-Ferrer, "TPP: Traceable privacy-preserving communication and precise reward for vehicle-to-grid networks in smart grids," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 11, pp. 2340–2351, Nov. 2015.
[28] L. Wu, Y. Zhang, K.-K. R. Choo, and D. He, "Efficient and secure identity-based encryption scheme with equality test in cloud computing," *Future Gener. Comput. Syst.*, vol. 73, pp. 22–31, Aug. 2017.
[29] Q. Wu, J. Domingo-Ferrer, and U. Gonzalez-Nicolas, "Balanced trustworthiness, safety, and privacy in vehicle-to-vehicle communications," *IEEE Trans. Veh. Technol.*, vol. 59, no. 2, pp. 559–573, Feb. 2010.
[30] Z. Xiao and Y. Xiao, "Security and privacy in cloud computing," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 2, pp. 843–859, 2nd Quart, 2013.
[31] L. Xu, C. Jiang, J. Wang, J. Yuan, and Y. Ren, "Information security in big data: Privacy and data mining," *IEEE Access*, vol. 2, pp. 1149–1176, 2014.
[32] Y. Xu, M. Wang, H. Zhong, J. Cui, L. Liu, and V. N. Franqueira, "Verifiable public key encryption scheme with equality test in 5G networks," *IEEE Access*, vol. 5, pp. 12702–12713, 2017.
[33] G. Yang, C. H. Tan, Q. Huang, and D. S. Wong, "Probabilistic public key encryption with equality test," in *Proc. Cryptographers' Track At RSA Conf.*, Mar. 2010, pp. 119–131.
[34] Y. Zhou, S. Chen, Y. Zhou, M. Chen, and Q. Xiao, "Privacy-preserving multi-point traffic volume measurement through vehicle-to-infrastructure communications," *IEEE Trans. Veh. Technol.*, vol. 64, no. 12, pp. 5619–5630, Dec. 2015.

RASHAD ELHABOB received the B.S. degree from the Faculty of Computer Science and Information Technology, University of Karary, Khartoum, Sudan, in 2010, and the master's degree from the Faculty of Mathematical Science, University of Khartoum, in 2014. He is currently pursuing the Ph.D. degree with the School of Software Engineering, University of Electronic Science and Technology of China, Chengdu, China. His current research interests include cryptography and network security.



YANAN ZHAO received the B.S. degree from the Jiangxi University of Science and Technology, in 2017. She is currently pursuing the M.S. degree with the School of Information and Software Engineering, University of Electronic Science and Technology of China. Her research interest includes identity-based public key cryptography.

IVA SELLA received the B.Tech. (IT) degree from the Technical University of Kenya, in 2014. She is currently pursuing the M.S. degree with the School of Information and Software Engineering, University of Electronic Science and Technology of China. Her research interest includes certificateless public key cryptography.

HU XIONG received the Ph.D. degree from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2009, where he is currently a Full Professor. His research interests include public key cryptography and networks security.