IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON ARTIFICIAL INTELLIGENCE IN CYBERSECURITY

Received June 29, 2019, accepted July 22, 2019, date of publication July 25, 2019, date of current version August 9, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2931056

Certificateless Deniable Authenticated Encryption for Location-Based Privacy Protection

GUANHUA CHEN, JIANYANG ZHAO, YING JIN, QUANYIN ZHU, CHUNHUA JIN, JINSONG SHAN, AND HUI ZONG

Faculty of Computer and Software Engineering, Huaiyin Institute of Technology, Huai'an 233003, China

Corresponding authors: Guanhua Chen (jschenguanhua@126.com) and Jianyang Zhao (jszhaojy@163.com)

This work was supported in part by the Natural Science Foundation of Jiangsu Province under Grant BK20161302, in part by the Electric Power Company Technology Project of Jiangsu Province under Grant J2017123, in part by the Natural Science Foundation of Huai'an under Grant HAB201837, and in part by the Jiangsu Provincial Government Scholarship Council.

ABSTRACT Deniable authenticated encryption (DAE) is a cryptographic primitive that supports data confidentiality with deniable authentication in an efficient manner. The DAE plays a significant role in location-based service systems for privacy protection. In this paper, we construct a certificateless DAE (CLDAE) scheme. The CLDAE is based on certificateless cryptosystems (CLCs), which avoids the need to manage public key certificates in public key infrastructure (PKI)-based cryptosystems and key escrow problems in identity-based cryptosystems (IBCs). Our design utilizes hybrid methods: tag-key encapsulation mechanism (TKEM) and data encapsulation mechanism (DEM). This technique is more suitable for location-based applications. We show how to construct a CLDAE scheme utilizing a certificateless deniable authenticated tag-KEM (CLDATK) and a DEM. We also design a CLDATK scheme and provide formal security proof using the random oracle model (ROM). We conduct a comprehensive performance analysis, which shows that CLDAE is highly efficient in terms of communication overhead. We also provide an application of the CLDAE for a location-based service (LBS) system.

INDEX TERMS Deniable authenticated encryption (DAE), certificateless cryptography, random oracle model, location-based services (LBSs).

I. INTRODUCTION

With the rapid expansion in mobile social networks, smart devices, and localization techniques, location-based service (LBS) have become an indispensable part of daily life duo to the fact that they provide users with various types of services related to location [1]–[3].

To use an LBS, users submit their location-based requirements (a nearest point of interest (POI), such as the nearest hospital, gas station or movie theatre) to a location-based service provider (LBSP); Then the LBSP returns the location response to the users. On the basis of this, the LBSP can deduce from private information about the user, such as commute routes, daily activity trajectories, and social connections. For example, when a user requests an LBS for hospital, the LBSP could predict that the user may have a health problem.

While users enjoy the tremendous convenience of LBSs, using these services exposes private information and risks disclosure [4]–[12]. An adversary colluding with the LBSP can obtain sensitive user information, which threatens user privacy. For example, according to the revealed user data, the malicious adversary may abuse them to track, rob or make sure no one in the house to steal.

Therefore, protecting location-based privacy disclosure is a huge challenge. Generally, digital signature can realize authentication. However, it also has another property: non-repudiation. It means that any third party can verify the validity of the signature. Hence, traditional digital signature can not guarantee the user's location privacy. Deniable authentication [13] can solve this problem. It is such an authentication: it allows the LBSP to confirm the source of a submitted location-based query but without the ability to provide that source to any third party. So deniable authentication can ensure the location privacy of users.

A. RELATED WORKS

Three related notions are introduced, CLC, hybrid encryption, and deniable authentication.

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Quan.

Based on deniable authentication, Zeng *et al.* [14] designed a scheme for LBSs. However, their scheme is based on PKI, which requires blinding the public key certificate and the user's public key. This increases the computational cost of the user. To avoid utilizing public key certificates, an IBC was designed by Shamir [15]. In an IBC, a user's public key can be obtained from identifying information such as telephone numbers and identification numbers. Its private key is generated by a private key generator (PKG), who is known to be a trusted third party. Nevertheless, because the PKG is capable of acquiring the private key of every entity, the IBC has a key escrow problem. To solve the key escrow problem in IBC, CLC was proposed [16]. In a CLC, a partial private key associated with a user's identity is produced by a key generator center (KGC). Then, the user's complete private key is created by integrating its partial private key with a secret value selected by the user. This approach solves the key escrow problem because the KGC is no longer capable of obtaining the secret key, which means that it cannot calculate the complete private key.

The most appropriate way to encrypt large messages is to utilize hybrid encryption, which consists of two parts: a key encapsulation mechanism (KEM) and a data encapsulation mechanism (DEM). The KEM utilizes public key techniques to encrypt a session key, while the DEM employs the session key to encrypt the actual data. Cramer and Shoup [17] constructed and analyzed the first hybrid encryption scheme. Abe *et al.* [18] designed a hybrid tag/KEM-DEM encryption framework. The proposed scheme takes a tag as input, making it simpler and improving its security. Subsequently, some other KEM-DEM hybrid encryption schemes [19]–[24] have been designed. Thus, their construction allows for a modular design and provides a clear separation of these two parts. Sakai and Hanaoka [25] designed a hybrid encryption method with tag using a non-interactive proof that can encrypt arbitrary-length plaintexts. Baek *et al.* [26] designed a stateful KEM-DEM encryption scheme that recycles some random parameters by holding a state to reduce the current random value. This approach is beneficial because it reduces the number of required computations.

Deniable authentication (DA) possesses two main characteristics: (1) an authorized receiver with the ability to determine the source of a given message; (2) the authorized receiver is not capable of providing the source of the given message to a third party. Thus, DA is suitable for privacy scenarios such as LBSs and ad hoc networks [27], [28]. Jin *et al.* proposed some CLDA schemes [29]–[32]; however, these schemes cannot achieve confidentiality. To ensure the privacy-preservation of the transmitted message, some deniable authenticated encryption (DAE) schemes have been constructed. Li *et al.* [33] presented a DAE scheme that simultaneously achieves confidentiality, integrity and deniable authentication. However, their scheme works in a PKI environment; consequently, it has the public key management problem. To solve this problem, Li *et al.* [34] constructed a tag/KEM-DEM hybrid DAE scheme and showed it to be highly efficient through a comprehensive analysis. Nevertheless, their scheme has the key escrow problem, which means that the PKG is capable of knowing all the entities' private keys. Subsequently, Ahene *et al.* [35] constructed a CLDAE scheme that avoided utilizing public key certificates, but it assumed the participation of a fully trusted PKG.

B. MOTIVATION AND CONTRIBUTION

Motivated by the approaches described above, we construct a CLDAE scheme. Our design employs tag-KEM and DEM hybrid encryption methods, which are advantageous in real-world scenarios. The concrete CLDAE construction process is shown in our scheme. We construct a CLDATK scheme and give a formal security proof in the random oracle model (ROM). We also apply our CLDAE scheme to an LBS system.

C. ORGANIZATION

The following is the arrangement of this paper. In Section 2, we present a formal model for CLDAE, and a formal model for CLDATK is described in Section 3. In Section 4, we describes the construction process for a CLDAE on the basis of a CLDATK and a DEM and design a concrete CLDATK scheme in Section 5. We discuss the results of a comprehensive performance analysis in Section 6. In Section 7, we provide an application of CLDAE to an LBS system. In Section 8, we provide the conclusions.

II. FORMAL MODEL FOR CLDAE

The security notions for CLDAE are given in this section.

A. SYNTAX

A CLDAE scheme comprises the following six algorithms:

Setup: Given k (a security parameter), the KGC produces the *params* and a master private/public key pair (s, P_{pub}). Other algorithms do not need to include the *params* because they are public.

Partial-Private-Key-Extract: Given s and ID (a user's identity), the KGC calculates a partial private key D_{ID}.

Set-Secret-Key: Given ID, a user outputs x_{ID} (a secret key) and PK_{ID} (the corresponding public key).

Set-Private-Key: Given D_{ID} and x_{ID}, a user constructs SK_{ID} (its full private key).

Deniable-Authenticated-Encrypt (DAE): Given a sender's identity ID_s, public key PK_{ID_s}, full private key SK_{ID_s}, a receiver's identity ID_r, public key PK_{ID_r}, and a message m, the sender produces a ciphertext σ.

Deniable-Authenticated-Decrypt (DAD): Given a sender's identity ID_s, public key PK_{ID_s}, a receiver's identity ID_r, public key PK_{ID_r}, full private key SK_{ID_r}, and a ciphertext σ, the receiver returns a message m or a failure symbol $\perp$.

If $\sigma = DAE(m, ID_s, PK_{ID_s}, SK_{ID_s}, ID_r, PK_{ID_r})$, then $m = DAD(\sigma, ID_s, PK_{ID_s}, ID_r, PK_{ID_r}, SK_{ID_r})$.

B. SECURITY NOTIONS

Our construction must achieve the desirable security requirements below:

- Confidentiality: any independent third party other than the entities involved cannot acquire any valuable advice related to the plaintext of a ciphertext;
- Deniable authentication: the receiver creates a deniable transcript that is probabilistically indistinguishable from the sender.

Type I and Type II adversaries exist in [16], [36], [37]. Type I is an attacker imitating a user but does not know the master private key of the KGC. It is capable of replacing users' public keys. Type II is an attacker who is a KGC can obtain the master private key. However, it is not capable of replacing users' public keys.

For confidentiality, the standard security concept used in our construction is indistinguishability against adaptive chosen ciphertext attacks (IND-CCA2). In this study, there are two games "IND-CCA2-I" and "IND-CCA2-II". It is assumed that the two games are played between Type I/II adversaries with their challengers as follows.

"IND-CCA2-I" game:

Setup: $\mathcal{C}$ executes *Setup* algorithm, transmits the *params* to $\mathcal{F}_{\mathcal{I}}$ and keeps s secret.

Phase 1: $\mathcal{F}_{\mathcal{I}}$ executes the following queries.

- Partial private key (PPK) queries: $\mathcal{F}_{\mathcal{I}}$ elects an identity ID. $\mathcal{C}$ executes the PPK algorithm and transmits its PPK D_{ID} to $\mathcal{F}_{\mathcal{I}}$.
- Private key (PVK) queries: $\mathcal{F}_{\mathcal{I}}$ elects an identity ID. $\mathcal{C}$ first calculates the PPK D_{ID} and the secret key (sk) x_{ID}. Then, it transmits $SK_{ID} = (x_{ID}, D_{ID})$ to $\mathcal{F}_{\mathcal{I}}$. Note that if a public key corresponding to an identity has been replaced, $\mathcal{F}_{\mathcal{I}}$ cannot query the identity because $\mathcal{C}$ is unconscious of the secret key.
- Request public key (RPK) queries: $\mathcal{F}_{\mathcal{I}}$ elects an identity ID. $\mathcal{C}$ checks whether an item $(ID_i, PK_{ID_i}, x_{ID_i})$ is in the list L_k. If yes, then $\mathcal{C}$ outputs PK_{ID_i}. Otherwise, $\mathcal{C}$ produces an item, adds it into L_k and outputs the corresponding public key.
- Public key replacement (PKR) queries: $\mathcal{F}_{\mathcal{I}}$ may select a new value to replace the original public key.
- DAE queries: $\mathcal{F}_{\mathcal{I}}$ selects a message m and two identities ID_s, ID_r. $\mathcal{C}$ obtains SK_s by implementing the PVK algorithm. Then, it transmits the result of DAE $(m, ID_r, PK_r, ID_s, PK_s, SK_s)$ to $\mathcal{F}_{\mathcal{I}}$.
- DAD queries: $\mathcal{F}_{\mathcal{I}}$ selects ID_s, ID_r, and a ciphertext σ. $\mathcal{C}$ obtains SK_r by implementing the PVK algorithm. Then, it transmits the result of DAD $(\sigma, ID_s, PK_s, ID_r, PK_r, SK_r)$ to $\mathcal{F}_{\mathcal{I}}$ (the result is $\perp$ if σ is not valid).

Challenge: $\mathcal{F}_{\mathcal{I}}$ determines when Phase 1 is over. Then, $\mathcal{F}_{\mathcal{I}}$ returns two challenged identities (ID_s^*, ID_r^*) and two messages (m_0, m_1) with equal-length. In phase 1, it is not capable of requesting the private key of identity ID_r^*, and ID_r^* is an identity that the PK has not replaced and the PPK has not requested. $\mathcal{C}$ elects $b \in \{0, 1\}$, calculates $\sigma^* = \text{DAE}(m_b, ID_r^*, PK_r^*, ID_s^*, PK_s^*, SK_s^*)$ and returns σ^* to $\mathcal{F}_{\mathcal{I}}$.

Phase 2: $\mathcal{F}_{\mathcal{I}}$ makes queries as in Phase 1. In this phase, it is not capable of requesting the private key of identity ID_r^*, and ID_r^* is an identity that the PK has not replaced and the PPK has not requested—just as in phase 1. Additionally, $\mathcal{F}_{\mathcal{I}}$ cannot execute a DAD query on $(\sigma^*, ID_s^*, ID_r^*)$.

Guess: $\mathcal{F}_{\mathcal{I}}$ outputs a guess b'. If $b' = b$, it wins the game. $\mathcal{F}_{\mathcal{I}}$'s advantage is

$$Adv_{CLDAE}^{IND-CCA2-I}(\mathcal{F}_{\mathcal{I}}) = \left|2Pr[b' = b] - 1\right|,$$

in which $Pr[b' = b]$ represents the probability.

"IND-CCA2-II" game:

Setup: $\mathcal{C}$ executes *Setup* algorithm and transmits the *params* and s to $\mathcal{F}_{\mathcal{II}}$.

Phase 1: $\mathcal{F}_{\mathcal{II}}$ also executes the following queries as the adversary $\mathcal{F}_{\mathcal{I}}$. However, in this case, PPK queries do not need to be requested because $\mathcal{F}_{\mathcal{II}}$ is capable of calculating those by itself.

- Private key queries: These queries are the same as those in the IND-CCA2-I game.
- Request public key queries: These queries are the same as those in the IND-CCA2-I game.
- DAE queries: These queries are the same as those in the IND-CCA2-I game.
- DAD queries: These queries are the same as those in the IND-CCA2-I game.

Challenge: $\mathcal{F}_{\mathcal{II}}$ determines when Phase 1 is over. Then, $\mathcal{F}_{\mathcal{II}}$ outputs two challenged identities (ID_s^*, ID_r^*) and two messages (m_0, m_1) with equal-length. In phase 1, it is not capable of requesting the PVK of identity ID_r^*. $\mathcal{C}$ elects a bit $b \in \{0, 1\}$, calculates $\sigma^* = \text{DAE}(m_b, ID_r^*, PK_r^*, ID_s^*, PK_s^*, SK_s^*)$ and returns σ^* to $\mathcal{F}_{\mathcal{II}}$.

Phase 2: $\mathcal{F}_{\mathcal{II}}$ requests queries as in Phase 1. In this phase, it is not capable of requesting the PVK of identity ID_r^*. Additionally, $\mathcal{F}_{\mathcal{II}}$ cannot execute a DAD query on $(\sigma^*, ID_s^*, ID_r^*)$.

Guess: $\mathcal{F}_{\mathcal{II}}$ outputs a guess b'. If $b' = b$, it wins the game. $\mathcal{F}_{\mathcal{II}}$'s advantage is

$$Adv_{CLDAE}^{IND-CCA2-II}(\mathcal{F}_{\mathcal{II}}) = \left|2Pr[b' = b] - 1\right|,$$

in which $Pr[b' = b]$ represents the probability.

Definition 1: A CLDAE scheme is IND-CCA2-i ($i \in \{I, II\}$) secure if there is no PPT (probabilistic polynomial time) adversary $\mathcal{F}_{\mathcal{I}}$ (or $\mathcal{F}_{\mathcal{II}}$), who wins "IND-CCA2-i" game with non-negligible advantage.

In the IND-CCA2-i game, we suppose that the adversary can execute a PVK query on identity ID_s^*, thus ensuring the scheme's forward security. That is, even if the sender's private key is compromised, confidentiality is preserved.

For deniable authentication, the security concept used in our construction is deniable authentication against adaptive chosen message attacks (DA-CMA). Here, there are two games "DA-CMA-I" and "DA-CMA-II". It is assumed that these two games are played between Type I/II adversaries with their challengers as follows.

"DA-CMA-I" game:

Setup: The description is the same as that in the IND-CCA2-I game.

Attack: The description is the same as that in the IND-CCA2-I game.

Forgery: $\mathcal{F}_{\mathcal{I}}$ creates an item $(m^*, \sigma^*, ID_s^*, ID_r^*)$. It cannot request the partial private key or replace the public key or the private key of the identity ID_s^*. Additionally, σ^* is not the result of the DAE query on (m^*, ID_s^*, ID_r^*) in the attack stage. If the result of DAD $(\sigma^*, ID_s^*, PK_s^*, ID_r^*, PK_r^*, SK_r^*)$ is valid, $\mathcal{F}_{\mathcal{I}}$ wins the game.

The advantage of $\mathcal{F}_{\mathcal{I}}$ is defined as the probability that it will win.

"DA-CMA-II" game:

Setup: The description is the same as that in the IND-CCA2-I game.

Attack: The description is the same as that in the IND-CCA2-I game.

Forgery: $\mathcal{F}_{\mathcal{II}}$ creates an item $(m^*, \sigma^*, ID_s^*, ID_r^*)$. It cannot request the identity ID_s^*'s PVK. Additionally, σ^* is not the result of the DAE query on (m^*, ID_s^*, ID_r^*) in the attack stage. If the result of DAD $(\sigma^*, ID_s^*, PK_s^*, ID_r^*, PK_r^*, SK_r^*)$ is valid, $\mathcal{F}_{\mathcal{II}}$ wins the game.

The advantage of $\mathcal{F}_{\mathcal{II}}$ is defined as the probability that it will win.

Definition 2: A CLDAE scheme is DA-CMA-i ($i \in \{I, II\}$) secure if there is no PPT adversary $\mathcal{F}_{\mathcal{I}}$ (or $\mathcal{F}_{\mathcal{II}}$) that wins the "DA-CMA-i" game with a non-negligible advantage.

From the above definition, we can see that the adversary is not allowed to perform a private key query on the identity ID_r^*, which is essential for achieving deniability. The sender and the receiver can create an indistinguishable transcript.

C. DATA ENCAPSULATION MECHANISM (DEM)

There are two algorithms in a DEM.

- Enc: This algorithm takes a system parameter k, a message m, a key K, and returns a ciphertext c. We can denote as $c \leftarrow Enc(K, m)$.
- Dec: This algorithm takes a ciphertext c, a key K, and returns a message m or $\perp$, which implies c is not valid.

For a DEM, the security concept used in our construction is indistinguishability against passive attackers (IND-PA). Here, the game, is played between an adversary and its challenger, and we describe it as follows.

"IND-PA" game:

Setup: $\mathcal{A}$ submits (m_0, m_1), which are of equal length.

Challenge: $\mathcal{C}$ randomly elects $K, \beta \in \{0, 1\}$, and transmits $c^* = Enc(K, m_\beta)$, which is a challenged ciphertext to $\mathcal{A}$.

Guess: $\mathcal{A}$ returns a guess β'. If $\beta' = \beta$, it wins the game.

The advantage of $\mathcal{A}$ is

$$Adv_{DEM}^{IND-PA}(\mathcal{A}) = \left|2Pr[\beta' = \beta] - 1\right|,$$

in which $Pr[\beta' = \beta]$ represents the probability.

Definition 3: A DEM is DA-CPA secure if no PPT adversary $\mathcal{A}$ exists that wins the above game with a non-negligible advantage.

III. CLDATK

The following security notions for CLDATK are given in this section.

A. SYNTAX

A generic CLDATK comprises seven algorithms as follows:

Setup: The description is the same as that for CLDAE in Section 2.

Partial-Private-Key-Extract: The description is the same as that for CLDAE in Section 2.

Set-Secret-Key: The description is the same as that for CLDAE in Section 2.

Set-Private-Key: The description is the same as that for CLDAE in Section 2.

Sym: Given a sender's ID_s, PK_{ID_s}, SK_{ID_s}, a receiver's ID_r, and PK_{ID_r}, the sender generates a symmetric key K and the internal state information ω.

Encap: Given an arbitrary tag τ and the internal state information ω, the sender produces an encapsulation φ.

Decap: Given φ, τ, ID_s, PK_{ID_s}, ID_r, PK_{ID_r} and SK_{ID_r}, the receiver produces K or $\perp$, which implies that the encapsulation is invalid.

If $(K, \omega) = \text{Sym}(SK_{ID_s}, ID_s, PK_{ID_s}, ID_r, PK_{ID_r})$ and $\varphi = \text{Encap}(\omega, \tau)$, then $K = \text{Decap}(\varphi, \tau, ID_s, PK_{ID_s}, ID_r, PK_{ID_r}, SK_{ID_r})$.

B. SECURITY NOTIONS

A CLDATK must meet the conditions of deniable authentication and confidentiality. Here, we provide security notions for CLDATK. For confidentiality, we consider two games: "IND-CCA2-i" for $i \in \{0, 1\}$. It is assumed that the following two games are played between Type I/II adversaries and their challengers.

"IND-CCA2-I" game:

Setup: $\mathcal{C}$ executes *Setup* algorithm, transmits *params* to $\mathcal{F}_{\mathcal{I}}$, and keeps s to itself.

Phase 1: $\mathcal{F}_{\mathcal{I}}$ executes the following queries.

- PPK queries: The description is the same as that in the IND-CCA2-I game of CLDAE in Section 2.
- PVK queries: The description is the same as that in the IND-CCA2-I game of CLDAE in Section 2.
- RPK queries: The description is the same as that in the IND-CCA2-I game of CLDAE in Section 2.
- PKR queries: The description is the same as that in the IND-CCA2-I game of CLDAE in Section 2.
- Generate symmetric key queries: $\mathcal{F}_{\mathcal{I}}$ selects two identities ID_s, ID_r. $\mathcal{C}$ obtains the sender's SK_{ID_s} by implementing the PVK algorithm. It executes $(K, \omega) = \text{Sym}(SK_{ID_s}, ID_s, PK_{ID_s}, ID_r, PK_{ID_r})$, saves the value ω (the adversary considers that this value is hidden and overrides the previous value), and transmits the key K to $\mathcal{F}_{\mathcal{I}}$.
- Encapsulation queries: $\mathcal{F}_{\mathcal{I}}$ elects a tag τ. When there is no matching ω, $\mathcal{C}$ returns $\perp$. Otherwise, $\mathcal{C}$ removes the stored value ω and outputs $\varphi = Encap(\omega, \tau)$
- Decapsulation queries: $\mathcal{F}_{\mathcal{I}}$ selects an encapsulation φ, a τ, and two identities ID_s, ID_r. $\mathcal{C}$ generates the receiver's SK_r by implementing the PVK algorithm. It transmits the result of Decap(φ, τ, ID_s, PK_s, ID_r, PK_r, SK_r) to $\mathcal{F}_{\mathcal{I}}$ (if φ is invalid, the result is $\perp$).

Challenge: $\mathcal{F}_{\mathcal{I}}$ determines when Phase 1 is over. Then, $\mathcal{F}_{\mathcal{I}}$ outputs two challenged identities, ID_s^*, ID_r^*. In phase 1, it is not capable of requesting the partial private key, replacing the public key, and private key of identity ID_r^*. $\mathcal{C}$ executes $(K_1, \omega^*) = \text{Sym}(SK_{ID_s^*}, ID_s^*, PK_{ID_s^*}, ID_r^*, PK_{ID_r^*})$, randomly selects K_0, $b \in \{0, 1\}$, and transmits K_b to $\mathcal{F}_{\mathcal{I}}$. $\mathcal{F}_{\mathcal{I}}$ will request the same queries as before when it receives K_b. Then, $\mathcal{F}_{\mathcal{I}}$ produces a tag τ^*. $\mathcal{C}$ calculates $\varphi^* = \text{Encap}(\omega^*, \tau^*)$ as a challenged encapsulation and transmits it to $\mathcal{F}_{\mathcal{I}}$.

Phase 2: $\mathcal{F}_{\mathcal{I}}$ requests queries as in Phase 1. In this phase, it is not capable of requesting the partial private key, replacing the public key, and the private key of identity ID_r^* as in phase 1. Additionally, $\mathcal{F}_{\mathcal{I}}$ cannot execute a decapsulation query on $(\varphi^*, K_b, ID_s^*, ID_r^*)$.

Guess: $\mathcal{F}_{\mathcal{I}}$ outputs a guess b'. If $b' = b$, it wins the game. The advantage of $\mathcal{F}_{\mathcal{I}}$ is

$$Adv_{CLDATK}^{IND-CCA2-I}(\mathcal{F}_{\mathcal{I}}) = \left|2Pr[b' = b] - 1\right|,$$

in which $Pr[b' = b]$ denotes the probability.

"IND-CCA2-II" game:

Setup: $\mathcal{C}$ executes *Setup* algorithm and transmits *params* and s to $\mathcal{F}_{\mathcal{II}}$.

Phase 1: $\mathcal{F}_{\mathcal{II}}$ also performs the following queries as does the adversary $\mathcal{F}_{\mathcal{I}}$. However, it does not need to request PPK queries because $\mathcal{F}_{\mathcal{II}}$ is capable of performing that calculation itself.

- PVK queries: The description is the same as that of CLDATK's IND-CCA2-I game.
- RPK queries: The description is the same as that of CLDATK's IND-CCA2-I game.
- GSK queries: The description is the same as that of CLDATK's IND-CCA2-I game.
- Encapsulation queries:: The description is the same as that of CLDATK's IND-CCA2-I game.
- Decapsulation queries: The description is the same as that of CLDATK's IND-CCA2-I game.

Challenge: $\mathcal{F}_{\mathcal{II}}$ determines when Phase 1 is over. Then, $\mathcal{F}_{\mathcal{II}}$ outputs (ID_s^*, ID_r^*), which are two challenged identities. In phase 1, it cannot request the private key of identity ID_r^*. $\mathcal{C}$ executes $(K_1, \omega^*) = \text{Sym}(SK_{ID_s^*}, ID_s^*, PK_{ID_s^*}, ID_r^*, PK_{ID_r^*})$, randomly selects K_0, $b \in \{0, 1\}$, and transmits K_b to $\mathcal{F}_{\mathcal{II}}$. $\mathcal{F}_{\mathcal{II}}$ will request the same queries as before when it receives K_b. Then, $\mathcal{F}_{\mathcal{II}}$ produces a tag τ^*. $\mathcal{C}$ calculates $\varphi^* = \text{Encap}(\omega^*, \tau^*)$ as a challenged encapsulation and transmits it to $\mathcal{F}_{\mathcal{II}}$.

Phase 2: $\mathcal{F}_{\mathcal{II}}$ requests queries just as in Phase 1. In this phase, it is not capable of requesting the private key of identity ID_r^*. Additionally, $\mathcal{F}_{\mathcal{II}}$ cannot execute a decapsulation query on $(\varphi^*, K_b, ID_s^*, ID_r^*)$.

Guess: $\mathcal{F}_{\mathcal{II}}$ outputs a guess b'. If $b' = b$, it wins the game. The advantage of $\mathcal{F}_{\mathcal{II}}$ is

$$Adv_{CLDATK}^{IND-CCA2-II}(\mathcal{F}_{\mathcal{II}}) = \left|2Pr[b' = b] - 1\right|,$$

where $Pr[b' = b]$ denotes the probability.

Definition 4: A CLDATK scheme is IND-CCA2-i ($i \in \{I, II\}$) secure if there is no PPT adversary $\mathcal{F}_{\mathcal{I}}$ (or $\mathcal{F}_{\mathcal{II}}$) that wins the "IND-CCA2-i" game with a non-negligible advantage.

In the IND-CCA2-i game, we assume that the adversary can execute a PVK query on identity ID_s^*, which ensures the scheme's forward security. In other words, even if the sender's private key is compromised, confidentiality is preserved.

For deniable authentication, the security concept used in our construction is deniable authentication against adaptive chosen message attacks (DA-CMA). Here, there are two games, "DA-CMA-i"($i \in \{I, II\}$). These two games are played between Type I/II adversaries and their challengers and described as follows.

"DA-CMA-I" game:

Setup: The description is the same as that of CLDATK's IND-CCA2-I game.

Attack: The description is the same as that of CLDATK's IND-CCA2-I game.

Forgery: $\mathcal{F}_{\mathcal{I}}$ creates an item $(\tau^*, \varphi^*, ID_s^*, ID_r^*)$. It is not capable of requesting the partial private key, replacement public key, and private key of the identity ID_s^*. Additionally, in the attack stage, φ^* is not the result of a key encapsulation query on (τ^*, ID_s^*, ID_r^*). If the result of $\text{Decap}(\varphi^*, ID_s^*, PK_s^*, ID_r^*, PK_r^*, SK_r^*)$ is valid, $\mathcal{F}_{\mathcal{I}}$ wins the game.

The advantage of $\mathcal{F}_{\mathcal{I}}$ is defined as the probability that it wins.

"DA-CMA-II" game:

Setup: The same as in CLDATK's IND-CCA2-II game.

Attack: The same as in CLDATK's IND-CCA2-II game.

Forgery: $\mathcal{F}_{\mathcal{II}}$ creates an item $(\tau^*, \varphi^*, ID_s^*, ID_r^*)$. It is not capable of requesting the identity ID_s^*'s private key. Additionally, φ^* is not the result of a key encapsulation query on (τ^*, ID_s^*, ID_r^*) in the attack stage. If the result of $\text{Decap}(\varphi^*, ID_s^*, PK_s^*, ID_r^*, PK_r^*, SK_r^*)$ is valid, $\mathcal{F}_{\mathcal{II}}$ wins the game.

The advantage of $\mathcal{F}_{\mathcal{II}}$ is defined as the probability that it wins.

Definition 5: A CLDATK scheme is said to be DA-CMA-i($i \in \{I, II\}$) secure if no PPT adversary $\mathcal{F}_{\mathcal{I}}$ (or $\mathcal{F}_{\mathcal{II}}$) exists that wins the "DA-CMA-i" game ($i \in \{I, II\}$) with a non-negligible advantage.

In the above definition, there is a restriction that $\mathcal{F}_{\mathcal{I}}/\mathcal{F}_{\mathcal{I}}$ is not allowed to perform the PVK query on ID_r^*, which is essential for realizing deniability. The sender and the receiver can create an indistinguishable transcript.

IV. A HYBRID CLDAE

A hybrid CLDAE scheme consists of a CLDATK and a DEM. Fig. 1 shows the description. Here, the DEM returns the ciphertext, which is a tag. Such a design simplifies the scheme description and has better generic security advantages. Theorems 1 and 2 give the result of our design.

Theorem 1: We suppose that a hybrid CLDAE scheme consists of a CLDATK and a DEM. If the CLDATK and the DEM are IND-CCA2 secure and IND-CPA secure, respectively,

CLDAE.Setup: Inputting a security parameter k:
1. $(params, s)$=**CLDATK.Setup**(k)
2. Return the system parameters $params$ and the master private key s

CLDAE.Partial-Private–key-Extract: Inputting the master private key s and an identity $ID \in \{0,1\}^*$:
1. D_{ID}=**CLDATK.Partial-Private–key-Extract** (s, ID)
2. Return the partial private key D_{ID}

CLDAE.Set-secret–key: Inputting an identity $ID \in \{0,1\}^*$:
1. (x_{ID}, PK_{ID})= **CLDATK. Set-secret–key**(ID)
2. Return the secret key x_{ID} and the public key PK_{ID}

CLDAE.Set-private–key: Inputting the partial private key D_{ID} and the secret key x_{ID}
1. S_{ID}= **CLDATK.Set-private–key**(D_{ID}, x_{ID})
2. Return the private key S_{ID}

CLDAE.Deniable-Authenticated-Encrypt: Inputting a message $m \in \{0,1\}^*$, the sender's private key S_{ID_s}, identity ID_s and public key PK_{ID_s}, the receiver's identity ID_r and public key PK_{ID_r}:
1. (K, ω)=**CLDATK.Sym**$(S_{ID_s}, ID_s, PK_{ID_s}, ID_r, PK_{ID_r})$
2. c=**DEM.Enc**(K, m)
3. σ=**CLDATK.Encap**(ω, c)
4. Return the ciphertext $\varphi=(\sigma, c)$

CLDAE.Deniable-Authenticated-Decrypt: Inputting a ciphertext σ, the sender's identity ID_s and public key PK_{ID_s}, the receiver's private key S_{ID_r}, identity ID_r and public key PK_{ID_r}:
1. K=**CLDATK.Decap**$(\sigma, c, ID_s, PK_{ID_s}, S_{ID_r}, ID_r, PK_{ID_r})$
2. If $K = \perp$, then return $\perp$ and stop
3. m=**DEM.Dec**(K, c)
4. Return the message m

FIGURE 1. CLDAE construction.

then CLDAE is IND-CCA2 secure. Specifically, we obtain

$$Adv_{CLDAE}^{IND-CCA2-i}(\mathcal{F}) = Adv_{CLDATK}^{IND-CCA2-i}(\mathcal{C}_1) + Adv_{DEM}^{IND-PA}(\mathcal{C}_2), i \in \{I, II\}$$

Proof: Refer to Appendix 1.

Theorem 2: We suppose that a CLDAE consists of a CLDATK and a DEM. If the CLDATK is DA-CMA secure, the CLDAE is DA-CMA secure. Specifically, we obtain

$$Adv_{CLDAE}^{DA-CMA-i}(\mathcal{F}) \leq Adv_{CLDATK}^{DA-CMA-i}(\mathcal{C}), \quad i \in \{I, II\}$$

Proof: See Appendix 2.

V. A CLDATK SCHEME

Seven algorithms exist to describe our proposed scheme. Here, we set a tag as a ciphertext returned by the DEM; the goal is to simplify the description and produce better generic security advantages.

A. BASIC KNOWLEDGE

In this section, we introduce the basic properties of bilinear pairings, the decisonal bilinear Diffie-Hellman problem (DBDHP), the computational Diffie-Hellman problem (CDHP) and the bilinear Diffie-Hellman problem (BDHP).

Let G_1, G_2 be an additive group and a multiplicative group, respectively. G_1 is generated by P, and G_1 and G_2 have the same prime order q. A bilinear pairing is a map $e : G_1 \times G_1 \rightarrow G_2$ with the following properties:

1) Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in G_1, a, b \in \mathbb{Z}_q^*$.
2) Non-degeneracy: There exists $P, Q \in G_1$ such that $e(P, Q) \neq 1$.
3) Computability: There is an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in G_1$

The admissible maps of this type are the modified Weil pairing and the Tate pairing ([38]–[41] provide more information). The security of this scheme lies in the difficulty of solving the problems below.

Definition 1 (Decisional Bilinear Diffie-Hellman Problem (DBDHP)): According to the aforementioned basic definition of bilinear pairings, the DBDHP is to determine $\theta = e(P, P)^{abc}$ given (P, aP, bP, cP) with $a, b, c \in \mathbb{Z}_q^*$ and an element $\theta \in \mathbb{Z}_q^*$.

Definition 2 (Computational Diffie-Hellman Problem (CDHP)): According to the aforementioned basic definition of bilinear pairings, the CDHP is to compute abP given (P, aP, bP) with $a, b \in \mathbb{Z}_q^*$.

Definition 3 (Bilinear Diffie-Hellman Problem (BDHP)): According to the aforementioned basic definition of bilinear pairings, the BDHP is to compute $e(P, P)^{abc}$ given (P, aP, bP, cP) with $a, b, c \in \mathbb{Z}_q^*$.

B. OUR SCHEME

Setup: Given G_1, G_2, P, and e as in Subsection A, Section V. Let n be the key length of a DEM and k be a security parameter $(q \geq 2^k)$, H_1, H_2, H_3 are three cryptographic hash functions, where H_1: $\{0,1\}^* \rightarrow G_1$, H_2: $\{0,1\}^* \times G_1 \times G_2 \rightarrow \{0,1\}^n$ and H_3: $\{0,1\}^* \times G_1 \times G_2 \rightarrow \mathbb{Z}_q^*$. The KGC randomly selects a master key $s \in \mathbb{Z}_q^*$ and calculates $P_{pub} = sP$. The system parameters $params$ are $(G_1, G_2, e, q, n, k, P, P_{pub}, H_1, H_2, H_3)$ and a master private key s.

Partial private key Extract: Given the master key s and a user's identity $ID \in \{0,1\}^*$, the KGC calculates its partial private key $D_{ID} = sQ_{ID}$, where $Q_{ID} = H_1(ID)$.

Set secret key: Given a user's identity ID, the user selects $x_{ID} \in \mathbb{Z}_q^*$ as its secret value and sets the public key $PK_{ID} = x_{ID}P$.

Set full private key: Given its partial private key D_{ID} and its secret value x_{ID}, the user outputs its full private key $SK_{ID} = (x_{ID}, D_{ID})$.

Sym: Given a sender's identity ID_s, public key PK_{ID_s}, private key S_{ID_s}, a receiver's identity ID_r and public key PK_{ID_r}, the algorithm is executed as shown in the steps below.

1) Elect $r \in \mathbb{Z}_q^*$.
2) Calculate $T = e(P_{pub}, Q_{ID_r})^r$.
3) Calculate $K = H_2(T, ID_s, ID_r, PK_{ID_s}, PK_{ID_r})$.
4) Calculate K and $\omega = (r, T, SK_{ID_s}, ID_s, ID_r, PK_{ID_s}, PK_{ID_r})$.

Encap: Given the state information ω and an arbitrary tag τ, the following algorithm is executed.

1) Calculate $h = H_3(\tau, T, PK_{ID_s}, PK_{ID_r}, x_s PK_{ID_r})$.
2) Calculate $V = hD_{ID_s} + rP_{pub}$.
3) Calculate $W = e(V, Q_{ID_r})$.
4) Calculate $S = hQ_{ID_s}$.
5) Calculate $\sigma = (W, S)$.

Decap: Given an encapsulation σ, a tag τ, a sender's identity ID_s, public key PK_{ID_s}, a receiver's private key S_{ID_r}, identity ID_r and public key PK_{ID_r}, the algorithm is executed as follows.

1) Calculate $T = W/e(S, D_{ID_r})$.
2) Calculate $h = H_3(\tau, T, PK_{ID_s}, PK_{ID_r}, x_r PK_{ID_s})$.
3) If $S = hQ_{ID_s}$, output $K = H_2(T, ID_s, ID_r, PK_{ID_s}, PK_{ID_r})$, otherwise output the symbol $\perp$.

C. SECURITY

Theorems 3 and 4 provide the security results for the CLDATK.

Theorem 3: In the ROM, under the DBDH and CDH assumptions, the above CLDATK is IND-CCA2 secure.

Proof: See Appendix 3.

Theorem 4: In the ROM, under the BDH and CDH assumptions, the above CLDATK is DA-CMA secure.

Proof: See Appendix 4.

TABLE 1. Performance comparison.

Schemes	Computational cost			Ciphertext size	Security	
	PM	BP	EP		DA-CMA	IND-CCA2
JXZXL [29]	5	2	0	$\|G_1\| + \|G_2\| + 2\|m\|$	×	√
AJL [35]	5	4	1	$\|G_1\| + \|G_2\| + \|m\|$	√	√
Ours	6	3	1	$\|G_1\| + \|G_2\| + \|m\|$	√	√

VI. PERFORMANCE

Next, we construct a detailed performance analysis of our design with the existing schemes [29], [35] listed in Table 1. We adopt an elliptic curve $y^2 = x^3 + x \bmod p$ and assume that $|G_1| = 513$ bits, $|G_2| = 1024$ bits, $|m| = 160$ bits, $|q| = 160$ bits, $|p| = 512$ bits, and hash value $= 160$ bits. We denote the point multiplication in G_1 by PM, the exponentiation calculation in G_2 by EC, and the pairing calculation in G_2 by PC. The XOR, hash function, and addition calculations are omitted because their computation speeds are sufficiently fast as to be negligible. The size of x is $|x|$. √ means that this scheme has the property, while × means that this scheme does not have the property. As shown in Table 1, [35] and our scheme can achieve IND-CCA2 and DA-CMA security simultaneously, but [29] only achieves DA-CMA.

We conducted an experiment on the PBC library. As needed, we set the library's embedding degree to 2. The experiment was executed on an Intel Pentium(R) Dual-Core processor running at 2.69 GHz, with 2,048 MB of RAM (2,007.04 MB available). On this machine, a PM requires 15.927 ms using an ECC with q of 160 bits. A PC and an EC require 26.68 ms and 3.126 ms, respectively. Reference [29] (hereafter denoted as JXZXL) takes 132.995 ms, [35] (hereafter called AJL) takes 189.481 ms, and our scheme takes 178.728 ms. Fig. 2 shows the computational cost for JXZXL [29], AJL [35], and our scheme. From Fig. 2, the computational cost of our scheme is greater than that of [29], but less than that of [35]. If standard compression techniques are used, the length of the elements in groups G_1 and G_2 are 65 bytes and 128 bytes. The ciphertext size for JXZXL, AJL and our scheme are $|G_1| + |G_2| + 2|m| = 65 + 128 + |m|/4$ bytes $= 193 + |m|/4$ bytes, $|G_1| + |G_2| + |m| = 65 + 128 + |m|/8$ bytes$= 193 + |m|/$ bytes, and $|G_1| + |G_2| + |m| = 65 + 128 + |m|/8$ bytes$= 193 + |m|/$ bytes.

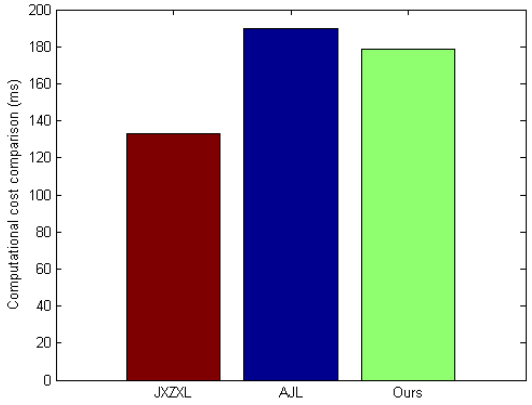


FIGURE 2. Computational cost comparison.

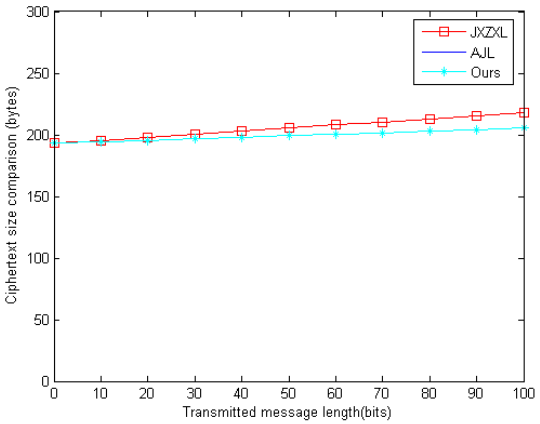


FIGURE 3. Ciphertext size comparison.

Fig. 3 shows the ciphertext size for JXZXL, AJL and our scheme. It can be seen from Fig. 3 that our scheme is smaller than that of AJL and the same as that of JXZXL.

VII. APPLICATION

We designed a certificateless location-based services (CLLBS) scheme utilizing the proposed CLDAE scheme that contains four phases: initialization, registration, authentication and expiration. Fig. 4 shows the concrete scheme.

A. INITIALIZATION PHASE

In this phase, the KGC executes Setup algorithm. Each entity is allotted an identity ID_i, $PK_{ID_i} = x_{ID_i} P$ and $SK_{ID_i} = (x_{ID_i}, D_{ID_i})$.

B. REGISTRATION PHASE

This phase requires user registration. It first transmits ID_i and PK_{ID_i} to the KGC, who checks whether ID_i is valid. If yes, the KGC produces a PPK $D_{ID_i} = sH_1(ID_i)$. Otherwise, the KGC refuses to register. After receiving D_{ID_i}, the user executes Set-Secret-Key and Set-Private-Key algorithms to acquire $PK_{ID_i} = x_{ID_i} P$ and $SK_{ID_i} = (x_{ID_i}, D_{ID_i})$.

C. AUTHENTICATION PHASE

A user ID_s wants to transmit a ciphertext to a service provider (SP) ID_r. The user produces m and executes a

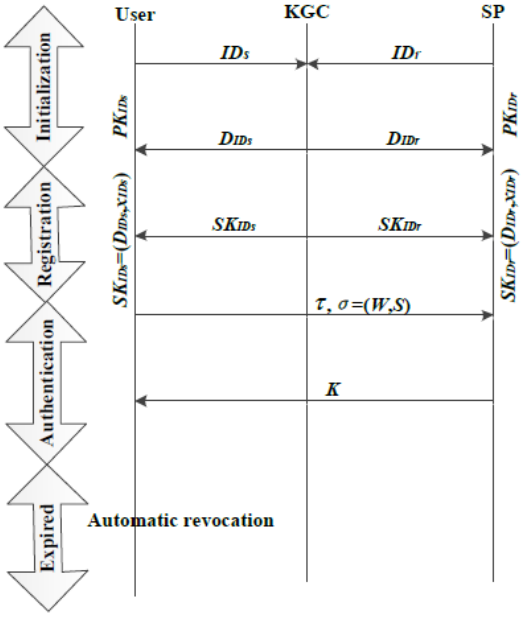


FIGURE 4. A CLLBS scheme.

DAE algorithm to produce an encapsulation ciphertext $\sigma = (W, S)$. To avoid replay attacks, we form a new encrypted message by concatenating m with a timestamp t. The user then transmits the ciphertext σ to the SP. The SP calculates the symmetric key $K = H_2(T, ID_s, ID_r, PK_{ID_s}, PK_{ID_r})$ and decrypts the message $m = DEM.Dec(K, c)$. To achieve confidentiality, the symmetric key is $K = H_2(T, ID_s, ID_r, PK_{ID_s}, PK_{ID_r})$ is kept by the user and the SP.

D. EXPIRED PHASE

Registration revocation occurs automatically because the date T expires. For example, if T is "06-30-2019", the user can access the LBS only before June 30, 2019. This means that the public/private keys of the user automatically expire after June 30, 2019. The SP stores an identity revocation list to guarantee the validity of the users.

VIII. CONCLUSION

In this paper, we proposed a hybrid CLDAE scheme in which confidentiality and deniable authentication occur in a single logical step. The design is based on a CLDATK and a DEM. We construct a CLDATK scheme that suffers neither from the public key certificate management problem nor the key escrow problem, and we provide formal proof in the ROM. A comprehensive performance analysis demonstrates that this construction is secure and effective. Compared with the two compared schemes, our design is more suitable for LBS applications.

APPENDIX

A. PROOF OF THEOREM 1

Proof. The following is our proof strategy. The modified attack games $Game_0$, $Game_1$, $Game_2$ is defined in [42], [43]. The distinct of games is how the environment answers $\mathcal{F}$'s queries.

$\mathcal{C}$ submits the challenged ciphretext $\sigma^* = (\psi^*, c^*)$ to $\mathcal{F}$ which can be obtained by encrypting either m_0 or m_1 on the basis of b using K^*, which is also utilized to decrypt the decapsulation ψ^* with the identities ID_s and ID_r chosen by $\mathcal{F}$. In $Game_i$ $(i = 0, 1, 2)$, we suppose that S_i is the event $\delta' = \delta$. δ is selected by the challenged oracle of $\mathcal{F}$. δ' is returned by $\mathcal{F}$. The probability depends on $\mathcal{F}$'s random selection and those of $\mathcal{F}$'s oracle.

The following lemma from [44] is utilized.

Lemma 1: Let E, E', and F be events defined on a probability space such that $Pr[E \wedge \neg F] = Pr[E' \wedge \neg F]$. Then, we have $\left|Pr[E] - Pr[E']\right| \leq Pr[F]$.

$Game_0$: We run the suitable key extraction algorithms to simulate $\mathcal{F}$'s view. Then we use the generated key to answer $\mathcal{F}$'s queries. Therefore, $\mathcal{F}$'s view is the same as it is in a real attack. So we get

$$\left|Pr[S_0] - \frac{1}{2}\right| = \frac{1}{2} Adv_{CLDAE}^{IND-CCA2-i}(\mathcal{F}), \quad i \in \{I, II\}$$

$Game_1$: In this game, we merely amend how the DAD oracle answers $\mathcal{F}$'s queries. After the invocation of the challenged DAE oracle, (φ, c) and the identities (ID_s, ID_r) are input to the DAD oracle. If $ID_s = ID_s^*$, $ID_r = ID_r^*$, $\varphi = \varphi^*$, and under the circumstance of a Type I adversary, the public keys have not been replaced, then the DAD oracle does not utilize the genuine DAD process, instead of utilizing the key K^* to decrypt c and outputs the result to $\mathcal{F}$.

Such a change has no effect on $\mathcal{F}$ and so

$$Pr[S_1] = Pr[S_0].$$

Lemma 2: A ppt algorithm $\mathcal{C}_1$'s running time is the same as that of $\mathcal{F}$, such that

$$|Pr[S_2] - Pr[S_1]| = Adv_{CLDATK}^{IND-CCA2-i}(\mathcal{C}_1), \quad i \in \{I, II\}.$$

Proof: The following proof shows how to structure an adversary $\mathcal{C}_1$ of the CLDATK to resist the IND-CCA2-i($i \in \{I, II\}$) attack.

The following game is played between $\mathcal{C}_1$ and $\mathcal{F}$.

- *Setup* $\mathcal{C}_1$ transmits the *param* to $\mathcal{F}$.

- *Phase* 1 When $\mathcal{F}$ executes a PPK query, PVK query, and PKR query on identity ID, $\mathcal{C}_1$ executes these queries to its own oracles and transmits the respond to $\mathcal{F}$. Given m, ID_i and ID_j, when $\mathcal{F}$ executes an encryption query, $\mathcal{C}_1$ does the following steps.

1) Execute GSK query on (ID_i, ID_j) to its own GSK query oracle to get K.
2) Calculate $c = DEM.Enc(K, m)$.
3) Execute a KE query on c to its own KE oracle to get φ.
4) Output $\sigma = (\varphi, c)$.

Given $\sigma = (\varphi, c)$, ID_i and ID_j, when $\mathcal{F}$ executes a decryption query, $\mathcal{C}_1$ does the steps below.

1) Execute a KD query on (φ, c, ID_i, ID_j) to get its own KD oracle to get K.
2) If $K = \perp$, stop.

3) Calculate $m = DEM.Dec(K, c)$ and output m.

- *Challenge* $\mathcal{F}$ produces challenged identities (ID_i, ID_j) and messages (m_0, m_1). $\mathcal{C}_1$ does the following steps.

1) Send (ID_i, ID_j) to its challenger to get K_β for $\beta \in \{0, 1\}$.
2) Select $\delta \in \{0, 1\}$.
3) Calculate $c^* = DEM.Enc(K_\delta, m_\delta)$.
4) Send c^* to $\mathcal{C}_1$ to get φ^*.
5) Output $\sigma^* = (\varphi^*, c^*)$ to $\mathcal{F}$.

- *Phase* 2 $\mathcal{F}$ does queries as in phase 1. When $\mathcal{F}$ has made its challenged encapsulation oracle, in order to answer $\mathcal{F}$'s decapsulation query with identities (ID_s, ID_r) and $\sigma = (\varphi, c)$, $\mathcal{C}_1$ does the following steps.

- If $(ID_s, ID_r, \varphi) = (ID_s^*, ID_r^*, \varphi^*)$, then the same process will be utilized before $\mathcal{F}$ submits to its challenged encapsulation oracle.
- Under the circumstance of a Type I adversary of a CLDATK scheme, if $(ID_s, ID_r, \varphi) = (ID_s^*, ID_r^*, \varphi^*)$ and the public keys have been replaced, then $\mathcal{C}_1$ responds to $\mathcal{F}$'s decapsulation query for $(ID_s^*, ID_r^*, \varphi^*, c^*)$ to get K. $\mathcal{C}_1$ utilizes K to decrypt c and transmits the answer to $\mathcal{F}$.
- Or else, $\mathcal{C}_1$ utilizes K_b to decrypt c and transmits the answer to $\mathcal{F}$.

- *Guess* $\mathcal{F}$ returns δ'. If $\delta' = \delta$, $\mathcal{C}_1$ returns $b' = 1$ which means K_b is a real key; otherwise it returns $b' = 0$ which means K_b is a random key.

When K_b is the real key, $\mathcal{F}$ is executed as it in $Game_1$. It implies that

$$Pr[S_1] = Pr[\delta' = \delta \mid b = 1] = Pr[b' = 1 \mid b = 1].$$

When K_b is the random key, $\mathcal{F}$ is executed as it in $Game_2$. It implies that

$$Pr[S_1] = Pr[\delta' = \delta \mid b = 0] = Pr[b' = 1 \mid b = 0].$$

From the security definition of CLDATK, we get

$$\begin{aligned} Adv_{CLDATK}^{IND-CCA2-i}(\mathcal{C}_1) &= \left|2Pr[b' = b] - 1\right| \\ &= \left|Pr[b' = 1 \mid b = 1] - Pr[b' = 1 \mid b = 0]\right|. \end{aligned}$$

Lemma 3: A ppt algorithm $\mathcal{C}_2$'s running time is the same as that of $\mathcal{F}$, such that

$$\left|Pr[S_2] - \frac{1}{2}\right| = \frac{1}{2} Adv_{DEM}^{IND-PA}(\mathcal{C}_2).$$

Proof: The following proof shows how to construct an adversary $\mathcal{C}_2$ of the CLDATK to resist the IND-PA attack. $\mathcal{F}$ is executed as it would be executed in game $Game_2$. Before $\mathcal{F}$ asks its challenged DAE query, we execute the corresponding CLDATK algorithms to answer $\mathcal{F}$'s queries. When $\mathcal{F}$ asks its challenged DAE oracle with identities (ID_s^*, ID_r^*), and messages (m_0, m_1), we just transfer (m_0, m_1) to $\mathcal{C}_2$'s challenged encryption oracle to get c^*. Then we request a GSK query to get K^* and ask an encapsulation query to obtain φ^*. We send (φ^*, c^*) to $\mathcal{F}$ and throw away K^*. We utilize the same procedure to answer $\mathcal{F}$'s queries as before apart from that if it asks a DAD query on $(ID_s^*, ID_r^*, \varphi^*, c)$ for some c. In this case of two cases:

- If a Type I adversary $\mathcal{F}$'s public keys have been replaced, then $\mathcal{C}_2$ decapsulates $(ID_s^*, ID_r^*, \varphi^*, c)$ by the secret key to get K, decrypts c and sends it to $\mathcal{F}$.
- Or else, we ask the decryption oracle of $\mathcal{C}_2$ with c and send the result to $\mathcal{F}$.

$Pr[S_2]$ is the probability that $\mathcal{C}_2$ accurately identifies the hidden bits of its challenged encryption oracle because $\mathcal{C}_2$ returns whatever $\mathcal{F}$ returns.

B. PROOF OF THEOREM 2

Proof: $\mathcal{F}$ is an adversary that attacks the CLDAE scheme with advantage $Adv_{CLDAE}^{DA-CMA-i}(\mathcal{F})$, for $i \in \{I, II\}$. An algorithm $\mathcal{C}$ that attacks the DA-CMA-i for the CLDATK with advantage at least $Adv_{CLDAE}^{DA-CMA-i}(\mathcal{F})$. We answers $\mathcal{F}$'s queries as follows.

- *Setup:* $\mathcal{C}$ transmits *param* to $\mathcal{F}$.

- *Attack:* When $\mathcal{F}$ executes a PPK query, PVK query, and PKR query on identity ID, $\mathcal{C}$ executes these queries to its own oracles and transmits the response to $\mathcal{F}$. Given m, ID_i and ID_j, when $\mathcal{F}$ executes a DAE query, $\mathcal{C}$ does the steps generate symmetric key query, encapsulation query and decapsulation query as $\mathcal{C}_1$ does in Lemma 2.

- *Fogery:* $\mathcal{F}$ returns $(m^*, \sigma^*, ID_s^*, ID_r^*)$, in which $\sigma^* = (\varphi^*, c^*)$. $\mathcal{C}$ returns $(\tau^*, \varphi^*, ID_s^*, ID_r^*)$, in which $\tau^* = c^*$.

Visibly, this is a perfect proof. If $\mathcal{F}$ wins the DA-CMA-i $i \in \{I, II\}$ game for CLDAE, $\mathcal{C}$ has the same advantage to win the DA-CMA-i game for CLDATK.

C. PROOF OF THEOREM 3

Proof: The proposed CLDATK is IND-CCA2-i ($i \in \{I, II\}$) secure in the ROM under the DBDH and CDH assumptions.

Proof: This theorem follows from Lemmas 4 and 5.

Lemma 4: In the ROM, under the DBDH assumption, a PPT adversary $\mathcal{F}_\mathcal{I}$ has a non-negligible advantage ϵ_{datk} winning the IND-CCA2-I game when executing in a time t and making q_{H_i} queries to H_i ($i = 1, 2, 3$), q_{par} PPK queries, q_{pk} PK queries, q_{gsk} GSK queries, q_{ke} KE queries, and q_{kd} KD queries, then $\mathcal{C}$ settles the DBDH problem with a probability

$$\epsilon_{datk} \geq \frac{\epsilon - q_d/2^{k-1}}{2q_{H_1}}$$

within $t' \leq t + O(q_{gsk} + q_{ke} + q_{kd})t_p$, in which t_p is one paring computation.

Proof: $\mathcal{C}$ gets an input (P, aP, bP, cP) and makes an attempt to calculate $e(P, P)^{abc}$. $\mathcal{C}$ plays $\mathcal{F}_\mathcal{I}$'s challenger and executes $\mathcal{F}_\mathcal{I}$ as a subroutine. $\mathcal{C}$ will answer $\mathcal{F}_\mathcal{I}$'s queries on H_1, H_2 and H_3. These answers are randomly produced, $\mathcal{C}$ maintains three lists L_1, L_2 and L_3 to save the answers. We make the assumptions as follows.

1) When $\mathcal{F}_\mathcal{I}$ requests the PPKE, SPK, GSK, KE and KD queries on identity ID, It must first request H_{ID}.
2) The result of a KE query will not be utilized in a KD query.

- *Setup* $\mathcal{F}_\mathcal{I}$ receives the system parameters with $P_{pub} = cP$ from $\mathcal{C}$. Here $\mathcal{C}$ does not know c.
- *Phase 1* $\mathcal{F}_\mathcal{I}$ executes the following queries.

- H_1 *queries* $\mathcal{C}$ randomly elects $\gamma \in \{1, 2, \ldots, q_{H_1}\}$. $\mathcal{F}_{\mathcal{I}}$ requests H_1 queries on its choice identities. At the γ-th query, $\mathcal{C}$ responds by $H_1(ID_\gamma) = bP$. At the i-th query with $i \neq \gamma$, $\mathcal{C}$ elects $w_i \in Z_q^*$, adds (ID_i, w_i) in the list L_1 and responds $H_1(ID_i) = w_iP$.
- H_2, H_3 *queries* $\mathcal{F}_{\mathcal{I}}$ requests its choice hash queries, $\mathcal{C}$ checks whether their lists include the corresponding items. If it is, $\mathcal{F}_{\mathcal{I}}$ will receive the already response; otherwise, a random value will be chosen. The request and response are added in the corresponding list.
- *PPKE queries* $\mathcal{F}_{\mathcal{I}}$ requests PPKE queries on identity ID_i. If $ID_i = ID_\gamma$, $\mathcal{C}$ fails. Otherwise, list L_1 must include (ID_i, w_i) (this means that $\mathcal{C}$ has responded $H_1(ID_i) = w_iP$.) $\mathcal{C}$ calculates the private key $cH_1(ID_i) = w_icP = w_iP_{pub}$ and transmits it to $\mathcal{F}_{\mathcal{I}}$.
- *Request public key queries* $\mathcal{F}_{\mathcal{I}}$ requests RPK queries on identity ID_i. If list L_k has contained the item $(ID_i, PK_{ID_i}, x_{ID_i})$, the stored public key PK_{ID_i} is returned. Otherwise, $\mathcal{C}$ produces a new item, outputs the public key and inserts them in list L_k.
- *Set private key queries* $\mathcal{F}_{\mathcal{I}}$ requests SPK queries on identity ID_i. If $ID_i = ID_\gamma$, $\mathcal{C}$ fails. Otherwise, $\mathcal{C}$ looks for an item $(ID_i, PK_{ID_i}, x_{ID_i})$ in L_k. If no, $\mathcal{C}$ will produce a new key pair and outputs $S_{ID_i} = (x_{ID_i}, w_iP_{pub})$
- *Replace public key queries* $\mathcal{F}_{\mathcal{I}}$ asks RPK queries on (ID_i, PK_{ID_l}). $\mathcal{C}$ adds/renews the item $(ID_i, PK_{ID_l}, \perp)$ to the list L_k.
- *Generation symmetric key queries* $\mathcal{F}_{\mathcal{I}}$ asks generation symmetric key queries on identities (ID_i, ID_j). If $ID_i \neq ID_\gamma$, $\mathcal{C}$ gets the private key S_{ID_i} by executing an SPK query. $\mathcal{C}$ then executes $(K, \omega) = Sym(S_{ID_s}, ID_s, ID_r, PK_{ID_s}, PK_{ID_r})$ as well as transmits K to $\mathcal{F}_{\mathcal{I}}$. $\mathcal{C}$ keeps ω and covers any previous value. If $ID_i = ID_\gamma$, but $ID_j \neq ID_\gamma$, $\mathcal{C}$ elects $x \in Z_q^*$, calculates $T = e(P_{pub}, Q_{ID_j})^x$. $\mathcal{C}$ then does H_2 queries to get $K = H_2(z)$ and transmits K to $\mathcal{F}_{\mathcal{I}}$. Here, the ω is (x, T, ID_i, ID_j).
- *Key encapsulation queries* $\mathcal{F}_{\mathcal{I}}$ generates τ. $\mathcal{C}$ checks if value ω is already exist. If not, $\mathcal{C}$ fails. Otherwise, $\mathcal{C}$ does the process as follows. If $ID_i \neq ID_\gamma$, $\mathcal{C}$ executes $\sigma = Encap(\omega, \tau)$ and transmits σ to $\mathcal{F}_{\mathcal{I}}$. If $ID_i = ID_\gamma$, but $ID_j \neq ID_\gamma$, $\mathcal{C}$ can get the private key S_{ID_j} by executing the SPK query. $\mathcal{C}$ executes H_3 queries to obtain $h = H_3(\tau, T)$, then calculate $S = hQ_{ID_i}$ and $W = Te(S, S_{ID_j})$. Finally, $\mathcal{C}$ transmits $\sigma = (S, W)$ to $\mathcal{F}_{\mathcal{I}}$.
- *Key decapsulation queries* $\mathcal{F}_{\mathcal{I}}$ submits a tag τ, an encapsulation σ, the identities (ID_i, ID_j) of a sender and a receiver. If $ID_j = ID_\gamma$, then (σ, τ) is not valid. If $\mathcal{F}_{\mathcal{I}}$ requests $H_3(T, \tau)$, in which $T = W/e(S, S_{ID_j})$, and $\mathcal{C}$ responds h which meets $S = hQ_{ID_i}$, it will fail. From $\mathcal{F}_{\mathcal{I}}$'s point of view, $\sigma = (S, W)$ is valid. The probability is at most $1/2^k$. If $ID_j \neq ID_\gamma$, $\mathcal{C}$ can get the private key S_{ID_j} by executing the SPK query. Then it transmits the result of $Decap(\sigma, \tau, ID_i, S_{ID_j})$ to $\mathcal{F}_{\mathcal{I}}$.
- *Challenge:* $\mathcal{F}_{\mathcal{I}}$ determines when phase 1 ends. It creates challenged identities (ID_s, ID_r). If $\mathcal{F}_{\mathcal{I}}$ has requested private key query on ID_γ, $\mathcal{C}$ fails. If $\mathcal{F}_{\mathcal{I}}$ does not elect $ID_r = ID_\gamma$ as its target identity, it also fails too. $\mathcal{C}$ elects $W^* \in G_2$, sets $S^* = aP$ and calculates $T^* = W^*/\theta$ (θ is the DBDH problem's candidate). Then $\mathcal{C}$ executes H_2 query to search $K_1 = H_2(T^*)$. $\mathcal{C}$ randomly elects K_0, $\beta \in \{0, 1\}$, and transmits K_β to $\mathcal{F}_{\mathcal{I}}$. Then $\mathcal{F}_{\mathcal{I}}$ transmits a tag τ^* to $\mathcal{C}$. Subsequently, $\mathcal{C}$ sends $\sigma^* = (S^*, W^*)$ to $\mathcal{F}_{\mathcal{I}}$.
- *Phase 2:* $\mathcal{F}_{\mathcal{I}}$ adaptively requests queries again as in phase 1 except it is not capable of asking a PPK query on ID_r and a KD query on (σ^*, τ^*) to get the private key.
- *Guess:* $\mathcal{F}_{\mathcal{I}}$ returns a bit β' for $(K_\beta, \omega^*) = Sym(S_{ID_s}, ID_s, PK_{ID_s}, ID_r, PK_{ID_r})$ and $\sigma^* = Encap(\omega^*, \tau^*)$ hold. If $\beta' = \beta$, $\mathcal{C}$ outputs 1 means $\theta = e(P, P)^{abc}$; Otherwise, $\mathcal{C}$ outputs 0 means $\theta \neq e(P, P)^{abc}$.

Now $\mathcal{C}$'s successful probability is estimated. $\mathcal{C}$ will fail if one of the following separate events is provided:

- E_1 : $\mathcal{F}_{\mathcal{I}}$ does not elect ID_γ as the receiver's identity in the challenge phase.
- E_2 : $\mathcal{F}_{\mathcal{I}}$ has asked a partial private key query on ID_γ.
- E_3 : $\mathcal{C}$ terminates in a decapsulation query since it rejects a valid encapsulation.

It is known that $\Pr[\neg E_1] = 1/q_{H_1}$, and $\Pr[E_3] \leq q_{kd}/2^k$. Additionally, $\neg E_1$ means $\neg E_2$.

Because

$$\begin{aligned} p_1 &= \Pr[\beta' = \beta \mid (K_\beta, \omega^*) \\ &= \mathrm{Sym}(S_{ID_s}, ID_s, PK_{ID_s}, ID_r, PK_{ID_r})] \end{aligned}$$

and

$$\sigma^* = Encap(\omega^*, \tau^*) = \frac{\epsilon + 1}{2} - \frac{q_{kd}}{2^k}$$

and

$$p_0 = \Pr[\beta' = i \mid \theta \in_R G_2] = \frac{1}{2} \quad \text{for } i = 0, 1,$$

We get

$$\begin{aligned} \mathrm{Adv}(\mathcal{C}) &= \frac{|p_1 - p_0|}{q_{H_1}} \\ &= (\frac{\epsilon + 1}{2} - \frac{q_{kd}}{2^k} - \frac{1}{2})(\frac{1}{q_{H_1}}) = \frac{\epsilon - q_{kd}/2^{k-1}}{2q_{H_1}} \end{aligned}$$

$\mathcal{C}$'s computation time is $O(q_{gsk} + q_{ke} + q_{kd})$ which denotes pairing computation in the GSK queries, KE queries and KD queries.

Lemma 5: In the ROM, under the CDH assumption, a PPT adversary $\mathcal{F}_{\mathcal{II}}$ has a non-negligible advantage ϵ_{datk} winning

the IND-CCA2-II game when executing in a time t and making q_{H_i} queries to H_i ($i = 1, 2, 3$), q_{pk} PK queries, q_{gsk} GSK queries, q_{ke} KE queries, and q_{kd} KD queries, then $\mathcal{C}$ tackles the CDH problem with

$$Adv^{IND_C CA2_I I}_{CLDAE}(\mathcal{A}) \leq q_T Adv_{CDH}(\mathcal{C}),$$

where $q_{H_1} = q_{pk} + q_{ke} + q_{kd} + 2$

Proof: $\mathcal{C}$ gets an input (P, aP, bP) and makes an attempt to calculate abP. $\mathcal{C}$ plays $\mathcal{F}_{\mathcal{II}}$'s challenger and executes $\mathcal{F}_{\mathcal{II}}$ as a subroutine in the IND-CCA2-II game for CLDATK. $\mathcal{C}$ will respond $\mathcal{F}_{\mathcal{II}}$'s queries on H_1, H_2 and H_3. These responses are randomly produced, $\mathcal{C}$ maintains three lists L_1, L_2 and L_3 to save the responses. We make the assumptions as follows.

1) When $\mathcal{F}_{\mathcal{II}}$ requests the SPK, GSK, KE and KE queries on identity ID, It must first request H_{ID}.
2) The result of a KE query will not be utilized in a KD query.

- *Setup:* $\mathcal{C}$ computes the system parameters with $P_{pub} = sP$ and transmits params and s to $\mathcal{F}_{\mathcal{II}}$. $\mathcal{C}$ selects two values $\gamma, \delta \in \{1, 2, \ldots, q_T\}$, and responds the following queries.
- *Phase 1:* $\mathcal{F}_{\mathcal{II}}$ executes queries as follows.
  - H_1, H_2, H_3 *queries:* $\mathcal{F}_{\mathcal{II}}$ requests these hash queries on its choice. $\mathcal{C}$ checks whether their lists include the corresponding items. If it is, the same response is sent to $\mathcal{F}_{\mathcal{II}}$; otherwise, a random value is chosen. The request and respond are added in the corresponding list.
  - *Request public key queries:* $\mathcal{F}_{\mathcal{II}}$ requests request public key queries on identity ID_i. If list L_k has contained the item (ID_i, PK_{ID_i}, x_{ID_i}), the stored public key PK_{ID_i} is output. Otherwise, $\mathcal{C}$ elects a value $r_i \in Z_q^*$. At the γ-th query, $\mathcal{C}$ responds by $PK_{ID_\gamma} = r_i aP$. At the δ-th query, $\mathcal{C}$ responds by $PK_{ID_\delta} = r_i bP$. At the $i \neq \gamma, \delta$-th query, $\mathcal{C}$ responds by $PK_{ID_i} = r_i P$ (here $r_i = x_{ID_i}$) and adds the item (ID_i, r_i, PK_{ID_i}) to L_k.
  - *Set private key queries:* $\mathcal{F}_{\mathcal{II}}$ requests SPK queries on identity ID_i. If $ID_i = ID_\delta, ID_\gamma$, $\mathcal{C}$ fails. Otherwise, $\mathcal{C}$ looks for an item (ID_i, PK_{ID_i}, x_{ID_i}) in L_k. If no, $\mathcal{C}$ will produce a new key pair and output $S_{ID_i} = (x_{ID_i}, w_i P_{pub})$.
  - *Generation symmetric key queries:* $\mathcal{F}_{\mathcal{II}}$ asks generation symmetric key queries on identities (ID_i, ID_j). If $ID_i \neq ID_\delta, ID_\gamma$, $\mathcal{C}$ gets the private key S_{ID_i} by executing an SPK query. Then $\mathcal{C}$ executes $(K, \omega) = Sym(S_{ID_i}, ID_i, ID_j, PK_{ID_i}, PK_{ID_j})$ and transmits K to $\mathcal{F}_{\mathcal{II}}$. $\mathcal{C}$ keeps ω and covers any previous value. If $ID_i = ID_\delta$ or $ID_i = ID_\gamma$, but $ID_j \neq ID_\delta, ID_\gamma$, $\mathcal{C}$ elects $x \in Z_q^*$, calculates $T = e(P_{pub}, Q_{ID_j})^x$. $\mathcal{C}$ then does H_2 queries to get $K = H_2(T)$ and transmits K to $\mathcal{F}_{\mathcal{II}}$. Here, the ω is (x, T, ID_i, ID_j).
  - *Key encapsulation queries:* $\mathcal{F}_{\mathcal{II}}$ generates τ. $\mathcal{C}$ checks whether value ω is already exist. If not, $\mathcal{C}$ fails. Otherwise, $\mathcal{C}$ does the process as follows. If $ID_i \neq ID_\delta, ID_\gamma$, $\mathcal{C}$ executes $\sigma = Encap(\omega, \tau)$ and transmits σ to $\mathcal{F}_{\mathcal{II}}$. If $ID_i = ID_\delta$ or $ID_i = ID_\gamma$, but $ID_j \neq ID_\delta, ID_\gamma$, $\mathcal{C}$ can get the private key S_{ID_j} by executing the SPK queries. $\mathcal{C}$ executes H_3 queries to obtain $h = H_3(\tau, T, PK_{ID_i}, PK_{ID_j}, R)$, then calculate $S = hQ_{ID_j}$ and $W = Te(S, S_{ID_j})$. If $ID_i = ID_\delta$ and $ID_j = ID_\gamma$, or $ID_i = ID_\gamma$ and $ID_j = ID_\delta$, $\mathcal{C}$ elects h from Z_q^*, sets $V = shQ_{ID_i}$, $W = e(V, Q_{ID_j})$, $S = hQ_{ID_i}$. Finally, $\mathcal{C}$ transmits $\sigma = (S, W)$ to $\mathcal{F}_{\mathcal{II}}$.
  - *Key decapsulation queries:* $\mathcal{F}_{\mathcal{II}}$ submits a tag τ, an encapsulation σ, the identities (ID_i, ID_j) of a sender and a receiver. If $ID_j = ID_\delta, ID_\gamma$, then (σ, τ) is not valid. If $\mathcal{F}_{\mathcal{II}}$ previously requests $H_3(\tau, T, PK_{ID_i}, PK_{ID_j}, R)$, in which $T = W/e(S, S_{ID_j})$, and $\mathcal{C}$ responds h which meets $S = hQ_{ID_i}$, it will fail. From $\mathcal{F}_{\mathcal{II}}$'s point of view, $\sigma = (S, W)$ is valid. If $ID_j \neq ID_\delta, ID_\gamma$, $\mathcal{C}$ can get S_{ID_j} by executing the SPK algorithm. $\mathcal{C}$ then transmits the result of $Decap(\sigma, \tau, ID_i, S_{ID_j})$ to $\mathcal{F}_{\mathcal{II}}$.
- *Challenge:* $\mathcal{F}_{\mathcal{II}}$ determines when phase 1 ends. It creates challenged identities (ID_i, ID_j). If $\mathcal{F}_{\mathcal{II}}$ has requested private key query on ID_δ, ID_γ, $\mathcal{C}$ will fail. If $\mathcal{F}_{\mathcal{II}}$ does not elect $ID_r = ID_\gamma$ as its target identity, it also fails too. $\mathcal{F}_{\mathcal{II}}$ returns a valid ciphertext $\sigma^* = (S^*, W^*)$ under the identities ID_δ, ID_γ. $\mathcal{F}_{\mathcal{II}}$ is conscious of σ^* is valid for (SK_{ID_i}, Q_{ID_j}) only if it inquires the hash value $H_3(\tau, T, PK_{ID_i}, PK_{ID_j}, R)$. On this occasion, the result of the CDH problem would be inserted in the list. Then $\mathcal{C}$ seeks the list for items of the form $(\tau, T, PK_{ID_i}, PK_{ID_j}, R)$. For each of them, $\mathcal{C}$ checks if $e(r_i^2 P; R) = e(riaP; ribP)$. If the equation is satisfied, $\mathcal{C}$ stops and outputs $R = abP$ as the result. If no such item exists, C fails and stops.
- *Phase 2:* $\mathcal{F}_{\mathcal{II}}$ adaptively requests queries again as in phase 1 except it is not capable of asking an SPK query on ID_j and a KD query on (σ^*, τ^*) to get the private key.
- *Guess:* $\mathcal{F}_{\mathcal{II}}$ returns a bit β' for $(K_\beta, \omega^*) = Sym(S_{ID_i}, ID_i, PK_{ID_i}, ID_j, PK_{ID_j})$ and $\sigma^* = Encap(\omega^*, \tau^*)$ hold. If $\beta' = \beta$, $\mathcal{C}$ outputs 1 means $R = abP$; Otherwise, $\mathcal{C}$ outputs 0 means $R \neq abP$.

From $\mathcal{F}_{\mathcal{II}}$'s view, the identities ID_δ, ID_γ are independent. The list L_1 for H_1 queries has at most q_{H_1} items. The probability of outputting ID_δ, ID_γ is $2/q_T$. If $\mathcal{F}_{\mathcal{II}}$ does not request the tuple $(\tau, T, PK_{ID_i}, PK_{ID_j}, R)$, it will not have any advantage. Otherwise, $\mathcal{C}$ will win the game because of the H_2 simulation.

Lemma 6: In the ROM, under the BDH assumption, a PPT adversary $\mathcal{F}_{\mathcal{I}}$ has a non-negligible advantage $\epsilon_{datk} \geq 5(q_{ke} + 1)(q_{ke} + q_{H_3})q_{H_1}/(2^k - 1))$ winning the DA-CMA-I game when executing in a time t and making q_{H_i} queries to H_i ($i = 1, 2, 3$), q_{par} PPK queries, q_{pk} PK queries, q_{gsk} GSK queries, q_{ke} KE queries, and q_{kd} KD queries, then $\mathcal{C}$ settles the BDH problem in expected time $t \leq 60343 q_{H_3} q_{H_1} 2^k / \epsilon_{datk}(2^k - 1)$.

Proof: To wield the forking algorithm [44], we have to prove how our design is applicable for the signature scheme described in [44]. In CLDATK imitate steps, the sender's private key fails (implying that the master private key fails). In this case, a method is needed to settle the BDH problem.

First, observe that the CLDATK of our design meets the requested three-phase honest-verifier zero-knowledge identification protocol, in which $\sigma_1 = T$ is the commitment, $h = H_3(\tau, T, PK_{ID_i}, PK_{ID_j}, R)$ is the hash value, and $\sigma_2 = W$ is the answer.

Second, we give a concrete imitate step and show a method of settling the BDH problem. Upon inputting (P, aP, bP, cP) of the BDH problem, $\mathcal{C}$ is needed to calculate $h = e(P, P)^{abc}$. $\mathcal{C}$ executes $\mathcal{F}_{\mathcal{I}}$ as a subroutine. $\mathcal{F}_{\mathcal{I}}$ consults $\mathcal{C}$ to answer H_1, H_2, and H_3 and $\mathcal{C}$ holds L_1, L_2, and L_3 to save the randomly generated responses. We describe this following process.

- *Setup:* $\mathcal{C}$ computes params with $P_{pub} = cP$ and sends P_{pub} with c to $\mathcal{F}_{\mathcal{I}}$.
- *Attack:* $\mathcal{C}$ responds $\mathcal{F}_{\mathcal{I}}$'s queries on the basis of the approach in Lemma 4 in addition to elect two different random values $\alpha, \beta \in \{1, 2, \ldots, q_{H_1}\}$ beforehand. At the α-th H_1 query, $\mathcal{C}$ responds by $H_1(ID_\alpha) = aP$. At the β-th H_1 query, $\mathcal{C}$ responds by $H_1(ID_\alpha) = bP$. If we assume $ID_\gamma = \{ID_\alpha, ID_\beta\}$, the process of simulation is the same as Lemma 4.
- *Fogery:* $\mathcal{F}_{\mathcal{I}}$ returns a tuple $(\sigma^*, \tau^*, ID_\alpha, ID_\beta)$, in which $\sigma^* = (S^*, W^*)$. We coalesce (ID_α, ID_β) and τ into a "generalized " forged tag (ID_γ, τ^*) in order to hide the identity-based aspect of the DA-CMA attack, and simulate the setting of an identity-less adaptive-CMA existential forgery. If $\mathcal{F}_{\mathcal{I}}$ is an efficient forger, then we are able to structure a Las Vegas machine $\mathcal{F}_{\mathcal{I}}{}'$ which returns $((ID_\gamma, \tau^*), h^*, \sigma^*)$ and $((ID_{\gamma,\tau^*}), \bar{h}^*, \bar{\sigma}^*)$ with $h^* \neq \bar{h}^*$ and the same commitment T^*. To settle the BDH problem based on the machine $\mathcal{F}_{\mathcal{I}}{}'$, we structure a machine $\mathcal{C}'$ as follows.
  1) $\mathcal{C}'$ executes $\mathcal{F}_{\mathcal{I}}{}'$ to receive two distinct signatures $((ID_\gamma, \tau^*), h^*, \sigma^*)$ and $((ID_\gamma, \tau^*), \bar{h}^*, \bar{\sigma}^*)$.
  2) $\mathcal{C}'$ calculates $e(P, P)^{abc}$ as $(W^*/\bar{W}^*)^{1/(h^*-\bar{h}^*)}$.

From the forking lemma [44] and the lemma on the relationship between given-identity and chosen-identity attack [45], [46], if $\mathcal{F}_{\mathcal{I}}$ succeeds in time t with probability $\epsilon_{datk} \geq 5(q_{ke}+1)(q_{ke}+q_{H_3})q_{H_1}/(2^k-1))$, then $\mathcal{C}'$ settles the BDH problem in expected time $t \leq 60343 q_{H_3} q_{H_1} 2^k / \epsilon_{datk}(2^k-1)$. Note that there is a change in the coefficient because the simulation elects two distinct values beforehand.

Lemma 7: In the ROM, under the CDH assumption, a PPT adversary $\mathcal{F}_{\mathcal{II}}$ has a non-negligible advantage ϵ_{datk} winning the DA-CMA-II game when executing in a time t and making q_{H_i} queries to H_i $(i = 1, 2, 3)$, q_{pvk} private key(PVK) queries, q_{pk} PK queries, q_{gsk} GSK queries, q_{ke} KE queries, and q_{kd} KD queries, then $\mathcal{C}$ settles the CDH problem with probability $\epsilon > (\epsilon_{datk} - (2/q_{pvk} + q_{ke}(q_{ke}+q_{H_3}+2)/2^k)$, within time $t' \leq t + (q_{gsk} + 2q_{ke} + q_{kd} + 2q_{H_3})t_e$, in which t_e means the time to calculate one pairing.

Proof: $\mathcal{C}$ gets an input (P, aP, bP) and makes an attempt to calculate abP. $\mathcal{C}$ plays $\mathcal{F}_{\mathcal{II}}$'s challenger and executes $\mathcal{F}_{\mathcal{II}}$ as a subroutine. $\mathcal{C}$ will answer $\mathcal{F}_{\mathcal{II}}$'s queries on H_1, H_2 and H_3. These responses are randomly produced, and $\mathcal{C}$ maintains three lists L_1, L_2 and L_3 to save the responses. Notice that both $\mathcal{C}$ and $\mathcal{F}_{\mathcal{II}}$ can calculate the partial private key $D_{ID_i} = sH_1(ID_i)$. We make the assumptions as follows.

1) When $\mathcal{F}_{\mathcal{II}}$ requests the SPK, GSK, KE and KD queries on identity ID, It must first request H_{ID}.
2) The result of a KE query will not be utilized in a KD query.

$\mathcal{C}$ responds $\mathcal{F}_{\mathcal{II}}$'s various queries on the basis of the method in Lemma 5.

Eventually, $\mathcal{F}_{\mathcal{II}}$ returns a valid ciphertext $\sigma^* = (S^*, W^*)$ from identities ID_γ, ID_δ. $\mathcal{F}_{\mathcal{II}}$ is not aware of σ^* is not a valid σ^* for (SK_{ID_i}, Q_{ID_j}) unless it makes for the hash value $H_3(\tau; T; r_i aP; r_i bP; r_i^2 abP)$. In this situation, the CDH problem's solution would be inserted in L_3. Then $\mathcal{C}$ seeks L_3 for tuples $(\tau; T; r_i aP; r_i bP; R)$. For each of them, $\mathcal{C}$ checks whether $e(r_i^2 P; R) = e(r_i aP; r_i bP)$. If yes, $\mathcal{C}$ stops and returns $R = abP$. If no, $\mathcal{C}$ fails.

Now we assess the failure probability of $\mathcal{C}$. $\mathcal{C}$ fails if $\mathcal{F}_{\mathcal{II}}$ requests the PVK queries on ID_γ, ID_δ. The failure probability is exactly $2/q_{pvk}$. The failure probability is at most $q_{ke}(q_{ke}+q_{H_3})/2^k$ for a conflict on H_3 in a key encapsulation query. The probability of rejecting a valid ciphertext is at most $2/2^k$. The bound on $\mathcal{C}$'s calculation time is a fact that each symmetric key query needs one pairing calculation, each key encapsulation query needs at most 2 pairing evaluations, each key decapsulation query needs one pairing evaluation. The solution's extraction from L_3 needs at most $2q_{H_3}$ pairings evaluations.

REFERENCES

[1] S. Zhang, G. Wang, and Q. Liu, "A dual privacy preserving scheme in continuous location-based services," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 4191–4200, Oct. 2018.

[2] I. Memon, Q. A. Arain, H. Memon, and F. A. Mangi, "Efficient user based authentication protocol for location based services discovery over road networks," *Wireless Pers. Commun.*, vol. 95, no. 4, pp. 3713–3732, Aug. 2017.

[3] S. C. Soma, T. Hashem, M. A. Cheema, and S. Samrose, "Trip planning queries with location privacy in spatial databases," *World Wide Web*, vol. 20, no. 2, pp. 205–236, Mar. 2017.

[4] W. Quan, Y. Liu, H. Zhang, and S. Yu, "Enhancing crowd collaborations for software defined vehicular networks," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 80–86, Aug. 2017.

[5] H. Zhang, W. Quan, H.-C. Chao, and C. Qiao, "Smart identifier network: A collaborative architecture for the future Internet," *IEEE Netw.*, vol. 30, no. 3, pp. 46–51, May/Jun. 2016.

[6] N. Cheng, F. Lyu, W. Quan, C. Zhou, H. He, W. Shi, and X. Shen, "Space/aerial-assisted computing offloading for IoT applications: A learning-based approach," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1117–1129, May 2019.

[7] M. Wernke, P. Skvortsov, F. Dürr, and K. Rothermel, "A classification of location privacy attacks and approaches," *Pers. Ubiquitous Comput.*, vol. 18, no. 1, pp. 163–175, 2014.

[8] I. Memon, I. Hussain, R. Akhtar, and G. Chen, "Enhanced privacy and authentication: An efficient and secure anonymous communication for location based service using asymmetric cryptography scheme," *Wireless Pers. Commun.*, vol. 84, no. 2, pp. 1487–1508, 2015.

[9] P. Asuquo, H. Cruickshank, J. Morley, C. P. A. Ogah, A. Lei, W. Hathal, S. Bao, and Z. Sun, "Security and privacy in location-based services for vehicular and mobile communications: An overview, challenges, and countermeasures," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4778–4802, Dec. 2018.

[10] N. Cheng, W. Xu, W. Shi, Y. Zhou, N. Lu, H. Zhou, and X. Shen, "Air-ground integrated mobile edge networks: Architecture, challenges, and opportunities," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 26–32, Aug. 2018.

[11] Y. Zhang, C. Xu, H. Li, K. Yang, J. Zhou, and X. Lin, "HealthDep: An efficient and secure deduplication scheme for cloud-assisted eHealth systems," *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 4101–4112, Sep. 2018.

[12] G. Liu, W. Quan, N. Cheng, H. Zhang, and S. Yu, "Efficient DDoS attacks mitigation for stateful forwarding in Internet of Things," *J. Netw. Comput. Appl.*, vol. 130, pp. 1–13, Mar. 2019.

[13] Y. Aumann and M. O. Rabin, "Authentication, enhanced security and error correcting codes," in *Advances in Cryptology—CRYPTO*, H. Krawczyk, Ed. Santa Barbara, CA, USA, 1998: Springer, pp. 299–303.

[14] S. Zeng, S. Tan, Y. Chen, M. He, M. Xia, and X. Li, "Privacy-preserving location-based service based on deniable authentication," in *Proc. IEEE/ACM 9th Int. Conf. Utility Cloud Comput. (UCC)*, C. Jiang, Ed. Shanghai, China, Dec. 2016, pp. 276–281.

[15] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology*, G. R. Blakley and D. Chaum, Ed. Santa Barbara, CA, USA: Springer, 1985, pp. 47–53.

[16] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Advances in Cryptology—ASIACRYPT*, C.-S. Laih, Ed. Taipei, Taiwan: Springer, 2003, pp. 452–473.

[17] R. Cramer and V. Shoup, "Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack," *SIAM J. Comput.*, vol. 33, no. 1, pp. 167–226, May 2003.

[18] M. Abe, R. Gennaro, and K. Kurosawa, "Tag-KEM/DEM: A new framework for hybrid encryption," *J. Cryptol.*, vol. 21, p. 1, pp. 97–130, Jan. 2008.

[19] K. Y. Choi, J. Cho, J. Y. Hwang, and T. Kwon, "Constructing efficient PAKE protocols from identity-based KEM/DEM," in *Proc. 16th Int. Workshop Inf. Secur. Appl. (WISA)*, H. Kim and D. Choi, Eds. Jeju Island, South Korea: IEEE Press, 2015, pp. 411–422.

[20] K. Emura, A. Kanaoka, S. Ohta, and T. Takahashi, "A KEM/DEM-based construction for secure and anonymous communication," in *Proc. IEEE 39th Annu. Comput. Softw. Appl. Conf., COMPSAC Workshops*, S. I. Ahamed, Ed. Taichung, Taiwan: IEEE Press, Jul. 2015, pp. 1–5.

[21] Y. Ishida, J. Shikata, and Y. Watanabe, "CCA-secure revocable identity-based encryption schemes with decryption key exposure resistance," *Int. J. Appl. Cryptogr.*, vol. 3, no. 3, pp. 288–311, Aug. 2017.

[22] X. Wu, Y. Han, M. Zhang, and S. Zhu, "Parallel long messages encryption scheme based on certificateless cryptosystem for big data," in *Proc. Int. Conf. Inf. Secur. Cryptogr.*, X. Chen, Ed. Xi'an, China: Springer, 2017, pp. 211–222.

[23] Y. Miao, X. Liu, R. H. Deng, H. Wu, H. Li, J. Li, and D. Wu, "Hybrid keyword-field search with efficient key management for industrial Internet of Things," *IEEE Trans. Ind. Inf.*, vol. 15, no. 6, pp. 3206–3217, Jun. 2019.

[24] X. Zhang, Y. Tang, H. Wang, C. Xu, Y. Miao, and H. Cheng, "Lattice-based proxy-oriented identity-based encryption with keyword search for cloud storage," *Inf. Sci.*, vol. 494, pp. 193–207, Aug. 2019.

[25] Y. Sakai and G. Hanaoka, "A remark on an identity-based encryption scheme with non-interactive opening," in *Proc. Int. Symp. Inf. Theory Appl. (ISITA)*, Singapore, Oct. 2018, pp. 703–706.

[26] J. Baek, W. Susilo, K. Salah, J. S. Ha, E. Damiani, and I. You, "Stateful public-key encryption: A security solution for resource-constrained environment," in *Proc. Cyber Secur., Princ., Techn., Appl.*, Singapore, 2019, pp. 1–22.

[27] F. Li, P. Xiong, and C. Jin, "Identity-based deniable authentication for ad hoc networks," *Computing*, vol. 96, no. 9, pp. 843–853, 2014.

[28] S. Zeng, Y. Chen, S. Tan, and M. He, "Concurrently deniable ring authentication and its application to LBS in VANETs," *Peer-to-Peer Netw. Appl.*, vol. 10, no. 4, pp. 844–856, Jul. 2017.

[29] C. Jin, C. Xu, F. Li, and X. Zhang, "A novel certificateless deniable authentication protocol," *Int. J. Comput. Appl.*, vol. 37, nos. 3–4, pp. 181–192, May 2015.

[30] C. Jin, C. Xu, X. Zhang, and F. Li, "An efficient certificateless deniable authentication protocol without pairings," *Int. J. Electron. Secur. Digit. Forensis*, vol. 7, no. 2, pp. 179–196, Jan. 2015.

[31] C. Jin, G. Chen, C. Yu, and J. Zhao, "Heterogeneous deniable authentication for E-voting systems," in *Proc. Int. Conf. Frontiers Cyber Secur.*, F. Li, Ed. Singapore: Springer, 2018, pp. 41–54.

[32] C. Jin, G. Chen, C. Yu, J. Zhao, Y. Jin, and J. Shan, "Heterogeneous deniable authentication and its application to e-voting systems," *J. Inf. Secur. Appl.*, vol. 47, pp. 104–111, Aug. 2019.

[33] F. Li, D. Zhong, and T. Takagi, "Efficient deniably authenticated encryption and its application to E-mail," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 11, pp. 2477–2486, Nov. 2016.

[34] F. Li, Z. Zheng, and C. Jin, "Identity-based deniable authenticated encryption and its application to e-mail system," *Telecommun. Syst.*, vol. 62, no. 4, pp. 625–639, Aug. 2016.

[35] E. Ahene, C. Jin, and F. Li, "Certificateless deniably authenticated encryption and its application to e-voting system," *Telecommun. Syst.*, vol. 70, no. 3, pp. 417–434, Mar. 2019.

[36] F. Li, M. Shirase, and T. Takagi, "Certificateless hybrid signcryption," in *Proc. Int. Conf. Inf. Secur. Pract. Exper.*, F. Bao, Ed. Xi'an, China: Springer, 2009, pp. 112–123.

[37] Y. Zhang, C. Xu, X. Lin, X. S. Shen, "Blockchain-based public integrity verification for cloud storage against procrastinating auditors," *IEEE Trans. Cloud Comput.*, to be published. doi: 10.1109/TCC.2019.2908400.

[38] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," *SIAM J. Comput.*, vol. 32, no. 3, pp. 586–615, 2003.

[39] Y. Zhang, C. Xu, J. Ni, H. Li, and X. Shen, "Blockchain-assisted public-key encryption with keyword search against keyword guessing attacks for cloud storage," *IEEE Trans. Cloud Comput.*, to be published. doi: 10.1109/TCC.2019.2923222.

[40] Y. Miao, X. Liu, K.-K. R. Choo, R. H. Deng, J. Li, H. Li, and J. Ma, "Privacy-preserving attribute-based keyword search in shared multi-owner setting," *IEEE Trans. Dependable Secure Comput.*, to be published. doi: 10.1109/TDSC.2019.2897675.

[41] X. Zhang, H. Wang, and C. Xu, "Identity-based key-exposure resilient cloud storage public auditing scheme from lattices," *Inf. Sci.*, vol. 472, pp. 223–234, Jan. 2019.

[42] Y. Zhang, C. Xu, X. Liang, H. Li, Y. Mu, and X. Zhang, "Efficient public verification of data integrity for cloud storage systems from indistinguishability obfuscation," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 3, pp. 676–688, Mar. 2017.

[43] X. Zhang, C. Xu, H. Wang, Y. Zhang, and S. Wang, "FS-PEKS: Lattice-based forward secure public-key encryption with keyword search for cloud-assisted industrial Internet of Things," *IEEE Trans. Depend. Sec. Comput.*, to be published. doi: 10.1109/TDSC.2019.2914117.

[44] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *J. Cryptol.*, vol. 13, no. 3, pp. 361–396, 2000.

[45] J. C. Choon and J. H. Cheon, "An identity-based signature from gap Diffie–Hellman groups," in *Public Key Cryptography—PKC*, Y. G. Desmedt, Ed. Miami, FL, USA: Springer, 2003, pp. 18–30.

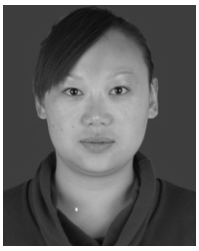
[46] Y. Miao, X. Liu, K. R. Choo, R. H. Deng, H. Wu, and H. Li, "Fair and dynamic data sharing framework in cloud-assisted Internet of everything," *IEEE Internet Things J.*, to be published. doi: 10.1109/JIOT.2019.2915123.

GUANHUA CHEN received the B.S. degree from the Xi'an University of Technology, Xi'an, China, in 2006. He is currently pursuing the master's degree with the Faculty of Computer and Software Engineering, Huaiyin Institute of Technology, Huai'an, China. His current research interests include cryptography and network security.

JIANYANG ZHAO received the M.S. degree in power electronic engineering and the Ph.D. degree in test measurement technique and instrument from the Nanjing University of Aeronautics and Astronautics. He is currently a Professor with the Huaiyin Institute of Technology. His current research interests include power quality monitoring and analysis, transient analysis, and power system equipment modeling and diagnoses.

YING JIN received the B.S. and M.S. degrees from Jiangsu University, in 1990 and 1993, respectively. She is currently a Professor with the Faculty of Computer and Software Engineering, Huaiyin Institute of Technology, Huai'an, China. Her current research interests include deep learning, embedded systems, and data analysis.

QUANYIN ZHU received the B.S. degree from the University of Electronic Science and Technology of China, Chengdu, Sichuan, China, in 1990. He was a Visiting Scholar with the Department of Automatic Control, Southeast University, Nanjing, Jiangsu, China, from 2000 to 2001, and an International Visiting Scholar with the School of Computing, Engineering and Information Science, Northumbria University, Newcastle, U.K., in 2007. He is currently a Professor with the Faculty of Computer and Software Engineering, Huaiyin Institute of Technology, Huaian, Jiangsu, China. His current research interests include data mining, intelligent information processing, and software engineering. He is the author or the coauthor of more than 30 papers published in international journals and conferences. He received five China invention patents. He is also a Senior Reviewer of the *Journal of Huaiyin Institute of Technology*, AICIT, and *Resources Policy*.



CHUNHUA JIN received the B.S. degree from Northwestern Polytechnical University, Xi'an, China, in 2007, the M.S. degree from Xidian University, Xi'an, China, in 2011, and the Ph.D. degree in cryptography from the University of Electronic Science and Technology of China, Chengdu, China, in 2016. She is currently a Teacher with the Faculty of Computer and Software Engineering, Huaiyin Institute of Technology, Huai'an, China. Her curent research interests include cryptography and network security.

JINSONG SHAN received the M.S. degree from Guizhou University, in 2006, and the Ph.D. degree from the PLA University of Science and Technology, in 2017. He is currently a Lecturer with the Huaiyin Institute of Technology, China. His current research interests include information retrieval, random algorithm, and machine learning.

HUI ZONG received the M.S. degree from the Nanjing University of Science and Technology, in 2011. She is currently a Senior Laboratory Engineer with the Huaiyin Institute of Technology, China. Her current research interests include date mining and parallel computing.